Springer Series in Astrophysics and Cosmology

Zhengyu Song · Dangjun Zhao · Stephan Theil *Editors*

Autonomous Trajectory Planning and Guidance Control for Launch Vehicles

Springer

Springer Series in Astrophysics and Cosmology

The series covers all areas of astrophysics and cosmology, including theory, observations, and instrumentation. It publishes monographs and edited volumes. All books are authored or edited by leading experts in the field and are primarily intended for researchers and graduate students.

Zhengyu Song · Dangjun Zhao · Stephan Theil
Editors

Autonomous Trajectory Planning and Guidance Control for Launch Vehicles

Springer

Editors
Zhengyu Song
China Academy of Launch Vehicle Technology
Beijing, China

Dangjun Zhao
Central South University
Changsha, China

Stephan Theil
German Aerospace Center
Bremen, Germany

ISSN 2731-734X ISSN 2731-7358 (electronic)
Springer Series in Astrophysics and Cosmology
ISBN 978-981-99-0612-3 ISBN 978-981-99-0613-0 (eBook)
https://doi.org/10.1007/978-981-99-0613-0

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Foreword

Aerospace technology is one of the most challenging high-tech fields in the world today, which has a strong driving effect on lots of industrial technologies. In recent years, the far-sighted space projects such as the development and utilization of space resources, human lunar explorations, and large-scale deep space explorations have become worldwide research hotspots. The scale of access to space is growing fast, and the distance to explore space is extending. In all these activities, the launch vehicles, as the only approach for human beings to go out of the earth today, have attracted more and more attention. Many space agencies and private enterprises are developing new-generation launchers to meet the increasing demands of responsive, massive, and low-cost transportation to and from space.

Auto pilot was first applied in rockets. However, the traditional design concept of its control system is highly dependent on the prescribed scenarios, which is relatively inadaptable to unexpected conditions and unknown environments. An autonomous guidance control for launch vehicles is very challenging, where a mature solution is desperately welcomed to the international aerospace industries. On the other hand, the artificial intelligence (AI) technologies have made great progresses. For instance, auto pilot is no longer a scientific fantasy for automobiles, and it will enter the life of thousands of families in the near future according to best estimations, leading to hundreds of millions of application cases. Compared with this trend, the application of autonomous flight technologies in launch vehicles is comparatively cautious and conservative.

In the changing era with the rapid development of science and technology, the aerospace industry is also trying to improve the autonomy and intelligence of space vehicles. As one of the outstanding representatives of these efforts, the authors bring forward the studies of the autonomous dynamic trajectory optimization control for launch vehicles, aiming to tackle unexpected situations intelligently and improve flight autonomy, adaptability, and fault tolerance. These studies focus on the needs of the real world, striving to bridge the technical gap between academic researches and engineering applications. It should be cautious when bringing the bold ideas and innovative theories into practical applications, thus successful verifications with demonstration and real flight vehicles, or high-fidelity simulation systems are detailed

discussed in these studies, which have great academic and practical values, and presented to the readers a broad vision of auto pilot in space and a comprehensive understanding of the cutting-edge technologies for the ascent flight and powered landing.

Beijing, China Weimin Bao
June 2022

Preface

Since the beginning of the twenty-first century, the space launch activities worldwide have grown vigorously. In China, the number of launches in 2021 reached 55, setting a record for that year, and is expected to exceed 60 in 2022. Commercial launches also account for the booming trend, for example, SpaceX alone has nearly maintained once-a-week launch cadence in 2022 till now.

New rockets appear on the stage one by one. China's new generation launch vehicles, i.e., LM-5, LM-7, and LM-8 rockets, have initiated their launch services, and at the same time, the next-generation manned launcher is being developed. In Russia, some versions of Angara series have completed their maiden flights. Other launchers, such as the H3 (Japan), Ariane 6 (ESA), and Vulcan (USA), are about to make their debuts in the near future. Moreover, many rockets developed by private companies have continued to progress forward to cope with fierce competitions.

Reusable launch vehicles (RLVs) have again attracted widespread attention, and this is mainly contributed to the successful practice from SpaceX, which demonstrated that the vertical takeoff and vertical landing (VTVL) is a feasible solution for reusability. In addition to the VTVL solution, other schemes, which adopt vertical takeoff and horizontal landing (VTHL), horizontal takeoff and horizontal landing (HTHL), as well as an upcoming field of the global ultra-high express transportation system, have been in the best interest of various space agencies and companies.

In the meanwhile, the risk of space launches still exists and is gradually increasing in line with the rapidly rising launch activities and commercial rockets. In the history of space launches, the propulsion and the control systems are the two main contributors to the launch failures. However, with the development of information technologies, the increase of the functional density of hardware products, the application of redundant or fault-tolerant solutions, and the improvement of the testability of avionics, the launch losses caused by control systems exhibit a downward trend, and the failures induced by propulsion systems become the focus of attention. For instance, the Long March launchers suffered multiple losses caused by the propulsion systems from 2013 to 2020. Although no irreparable disasters occurred such as explosions, the control systems which are deliberated-designed for the prescribed scenarios feel helpless to save the missions under these failure conditions.

Based on the above background, Dr. Song Zhengyu and other scholars organized a research team to develop some new autonomous guidance technologies, which potentially can fulfill flight missions in case of no fatal faults. For the ascent phase, the thrust drop is a common unfatal fault. While for the descent phase of a reusable launch vehicle, the unpredictable and uncontrollable initial descending conditions as well as the wind disturbances and atmospheric uncertainties can be taken as some type of faults for the prescribed descending trajectory. Autonomous trajectory re-planning during the flight is a feasible and necessary method to handle the above faults. Consequently, the dynamic optimization methods for online trajectory control of launch vehicles are intensively studied to satisfy the requirements of autonomous flight control under faults and various uncertain conditions. The authors try to address the real-time computational and convergent problems in the implementation of online optimization and control via several methods.

The research team was participated by 15 scholars from China, Germany, and other countries. This book is a summary of the researches of the team, reflecting the latest progress of relevant projects and academic researches of the autonomous guidance method. The monograph is strongly applications-oriented and exhibits a strong link with real-world space activities and great value to the aerospace industry. There are seven chapters included in this book, and each is a complete report concerning a topic. The book reflects the contents of four aspects. The first chapter summarizes the development of launch vehicles, especially mentions the demand and influences of reusable, intelligent, and autonomous technologies on the performance of launch vehicles after entering the twenty-first century. Chapters 2–4 mainly focus on the flight in the ascent phase, in which the autonomous guidance is mainly reflected in the online planning after the failures occur. Chapters 5 and 6 mainly discuss the powered descent guidance technologies. Aerodynamic uncertainty has a significant impact on the ascent/landing guidance control, thus, the estimation of aerodynamic parameters is discussed in the last Chap. 7, which is helpful to improve flight autonomy.

Beijing, China
June 2022

Zhengyu Song

Acknowledgements

The inspiring and fascinating topics covered in this book could not have been completed to this level of details without the timely and invaluable efforts of the co-authors who not only contributed their time and knowledge, but also provided material as well as their technical expertise to this book. I would also like to thank all participants of book for their helpful suggestions and tips.

In addition, many thanks to Prof. Zhao Dangjun, Dr. Wang Cong and Mr. Zhang Zhiwei, who have worked hard and patiently to organize this manuscript.

This study was also supported by the development plans of Chinese launch vehicles LM-2F, LM-7, LM-8, and the National Natural Science Foundation of China under Grant 52232014 and 61773341.

Introduction

This book presents a number of advanced autonomous dynamic trajectory optimal control technologies for launch vehicles. Improving flight autonomy has gained broad-ranging concerns, so as to responsively and autonomously handle the unexpected events or failures. It was reported that the enhanced Proton M launch system adopted artificial intelligence elements to improve the in-flight fail-safe performance level,[1] while many NASA scientists and engineers prefer to discuss machine learning and autonomy rather than artificial intelligence.[2] In China, the China Academy of Launch Vehicle Technology has drafted a roadmap of smart rockets, and in Europe, the Future Launchers Preparatory Program is currently investigating on-board real-time trajectory guidance optimization technologies for future reusable launchers.[3]

For ascent flight, if particular cases occur (e.g., the failures of engine out or thrust drop), the prescribed shift conditions between the flight phases are no longer satisfied. The remaining carrying capacity could be fully utilized through autonomous re-planning, to maximize the capability to finish the original mission, or release the payload to an optimal parking orbit, thereby avoiding the total loss brought by crashing to the ground. Under these situations, the space missions face the need for "end-to-end" global optimizations compared to traditional serial sequential optimizations. For the recovery of the launchers, the initial conditions of the powered vertical landing are hard to be prescribed in advance, and the landing is confronted with wind disturbances and atmospheric uncertainties. Thus, an on-board dynamic trajectory planning is definitely needed for the pinpoint soft landing, which is also a key technology for landing on any celestial body with an atmosphere. With the evolving of highly efficient algorithms and embedded hardware and software products, the

[1] Enhanced Proton M Launch System—New Features. SPACENEWS, October 4, 2021. https://spacenews.com/enhanced-proton-m-launch-system-new-features/.

[2] Beyond HAL: How artificial intelligence is changing space systems. August 15, 2017. https://spacenews.com/beyond-hal-how-artificial-intelligence-is-changing-space-systems/.

[3] Unveiling vehicles and technologies for future space transportation. https://www.esa.int/Enabling_Support/Space_Transportation/Future_space_transportation/Unveiling_vehicles_and_technologies_for_future_space_transportation.

model-based optimization and control would replace the traditional methods, which not only deal with unconstrained or simply constrained tasks, but also tackle complex constraints to address the new challenges.

This book is organized into seven chapters.

Chapter 1: This chapter reviews the development of launch vehicle technologies first, from the initial development phase, the space shuttle phase, and the commercial service phase to the comprehensive performance improvement phase, where the historical and the emergent launchers are briefly introduced. Then, the current development of launch vehicle technologies is summarized, showing an upcoming "space transportation revolution era" with distinguished features such as reusable, autonomous, or intelligent. Finally, facing the increasing demands of space activities, the future technical challenges of reusable launch vehicles and intelligent and autonomous technologies, including but not limited to the guidance control, are discussed.

Chapter 2: This chapter starts with the definition and explanation of the autonomous guidance method (AGM) in the ascent phase. Two technical systems of AGMs for Long March launch vehicles are discussed. One is the closed-loop guidance method for a prescribed target orbit, and its basic theory, fundamental and enhanced algorithms are introduced in order. The second is the joint optimization of the target orbit and the flight path, mainly applied under the thrust drop fault, in which three solutions are discussed, including the state-triggered-indices-based method for the continuous powered phases, segmented rescue optimizations crossing a coasting phase, and multiple graded optimizations. These studies will further improve the autonomy and fault adaptability of the ascent flight.

Chapter 3: This chapter focuses on the accent guidance problem under the thrust drop fault based on an improved generalized quasi-spectral model predictive static programming (IGS-MPSP) method, in which a scale factor is first introduced for the time interval as the additional variable to adjust the terminal time, then a new sensitive relation for the final time is established. The proposed IGS-MPSP method can well deal with the pool initial guess for searching the appropriate final time of the accent guidance problem. The numerical comparison studies reveal that, for the accent guidance problem with thrust drop faults, the IGS-MPSP method presents similar results to that of SOCP-based methods, but with more computational efficiency.

Chapter 4: This chapter proposes a Birkhoff-polynomial-based pseudospectral method for optimal control of a class of nonlinear cascaded second-order systems, which are widely used to formulate the motion of space vehicles. The convex version of the original problem is derived, then both the first- and second-order Birkhoff pseudospectral methods in the manner of Chebyshev and Legendre polynomials are used to transcribe the resulting convex problem. The major advantages that lie in the condition number, the scale of the transcribed problem, and the computational cost consumption are dramatically reduced while maintaining accuracy and efficiency. These advantages are well demonstrated by a simple cart problem with an analytical solution and a rescue orbit searching problem for a launch vehicle with thrust drop failures.

Chapter 5: This chapter presents an autonomous descent guidance algorithm which is able to deal with both the aerodynamic descent and the powered landing phases of a reusable rocket. The sequential convex optimization is applied to a Cartesian representation of the equations of motion. The contributions are a more systematic exploitation and separation of convex and non-convex terms to minimize the computational cost, the inclusion of highly nonlinear terms represented by aerodynamic accelerations, a complete problem reformulation based on the Euler angle rates, an improved transcription based on a generalized *hp* pseudospectral method, and a dedicated formulation of the aerodynamic guidance problem. The numerical results reveal the proposed approach is a valid candidate solution to solve the entire descent phase in real time.

Chapter 6: This chapter investigates novel strategies efornhancing the adaptability and autonomy of decision-making in powered descent of launch vehicles. Two guidance algorithms for specific sceneries are presented. One is the multi-point powered descent algorithm for planetary soft landings, where the best landing site is identified quickly from many candidates and the associated fuel optimal trajectory is generated with sensitivity information. The second is the trajectory optimization algorithm for emergency landings on highways, which reconstructs the trajectory to make decision between the pad and highway landings in terms of the degree of trajectory deviations.

Chapter 7: This chapter aims to estimate the aerodynamic parameters of launch vehicles based on the measurements of distance, height, and velocity. For the existence of inestimability, not all the aerodynamic parameters can be estimated. Different from the previous approaches, the prediction accuracy of the model with estimated values is focused on, not the accuracy of the estimation for the aerodynamic parameters themselves. The aerodynamic parameters are considered as time-dependent parameters and are approximated with the piece-wise linear function. Statistic criterion-based aerodynamic parameter estimation is considered for improving the accuracy of the model prediction. Finally, the numerical experiments are demonstrated.

The presented technologies have been tested and validated on real-world launchers, highly representative benchmarks, and simulation systems, which would be viable candidates for future space transportation and recovery systems.

Zhengyu Song
Research Fellow, China Academy of
Launch Vehicle Technology
Adjunct Professor
College of Control Science and
Engineering
Zhejiang University
Hangzhou, People's Republic of China
zycalt12@sina.com

Contents

1 Review, Prospect and Technical Challenge of Launch Vehicle 1
Xiaowei Wang, Feng Zhang, Dongsheng Hu, Rong Chen, and Zhengyu Song

2 Autonomous Guidance Control for Ascent Flight 33
Zhengyu Song, Cong Wang, and Yong He

3 Ascent Predictive Guidance for Thrust Drop Fault of Launch Vehicles Using Improved GS-MPSP 75
Xiaodong Yan and Cong Zhou

4 Birkhoff Pseudospectral Method and Convex Programming for Trajectory Optimization 99
Dangjun Zhao, Zhiwei Zhang, and Mingzhen Gui

5 Autonomous Descent Guidance via Sequential Pseudospectral Convex Programming 129
Marco Sagliano, David Seelbinder, and Stephan Theil

6 Simultaneous Trajectory Optimization for Adaptive Powered Descent 177
Zhenyu Wei, Lin Ma, Kexin Wang, and Zhijiang Shao

7 Aerodynamic Parameter Estimation for Launch Vehicles 201
Jian Jia, Weifeng Chen, and Zixuan Wang

Conclusions 215

Chapter 1
Review, Prospect and Technical Challenge of Launch Vehicle

Xiaowei Wang, Feng Zhang, Dongsheng Hu, Rong Chen, and Zhengyu Song

1.1 Review on Development of Launch Vehicle

Following six decades of development, the technology of launch vehicle has progressed with respect to the dual action of demand traction and technical promotion, providing increasingly valuable high-tech services for society. Currently, the development of launch vehicle is progressing with respect to stronger capabilities, higher reliability, lowering costs, flexibility, and user convenience [1–3]. Retrospectively, the global development history of the launch vehicle technology can be roughly categorized into the following four stages with distinct characteristics of the decades (Fig. 1.1).

(1) Initial Development Stage (1950–1970s): This stage primarily solved the problems of zero-to-one, and furnished the basic demands of access to space.

(2) Space Shuttle Stage (1970–1990s): In this stage, the reliability and carrying capacity of launch vehicle have tremendously improved to meet the diverse launch demands, and concomitantly, the early phase of reusable technology developed.

X. Wang · F. Zhang · D. Hu · R. Chen
China Academy of Launch Vehicle Technology, Beijing, People's Republic of China
e-mail: wangxwbuaa@163.com

F. Zhang
e-mail: jimmyzf2004@126.com

D. Hu
e-mail: hudongsheng82@126.com

R. Chen
e-mail: ronda_coco@163.com

Z. Song (✉)
China Academy of Launch Vehicle Technology, China Aerospace Science and Technology Corporation, 100076 Beijing, People's Republic of China
e-mail: zycalt12@sina.com

Z. Song et al. (eds.), *Autonomous Trajectory Planning and Guidance Control for Launch Vehicles*, Springer Series in Astrophysics and Cosmology,
https://doi.org/10.1007/978-981-99-0613-0_1

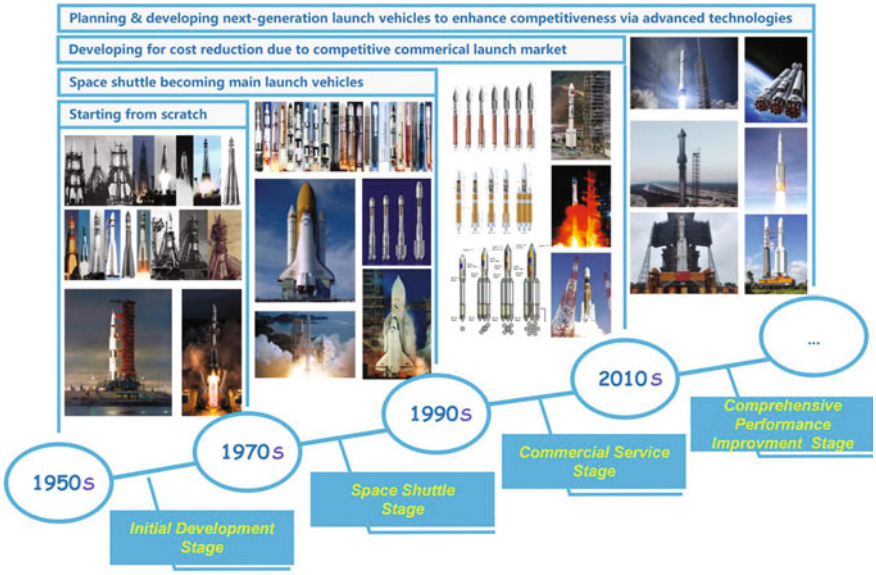


Fig. 1.1 Development stage of launch vehicle

(3) Commercial Service Stage (1990–2010s): International commercial launches flourished during this stage. To capture the international commercial launch market, the design and development of the launch vehicle weighed in the cost factors and more diversified mission adaptability demands.

(4) Comprehensive Performance Improvement Stage (2010–now): Owing to the stiff competition and incessant technical innovation, the launch vehicles during this stage have an improved comprehensive performance through the appropriation of the principles of modularization, serialization, and combination. By employing the advanced technologies such as the reusability and artificial intelligence, the comprehensive performance of the launch vehicle is fully enhanced. Therefore, the constantly emerging novel launch vehicles augment their competitiveness in the launch market [2, 3].

1.1.1 Initial Development Stage (1950–1970s)

The emergence of the launch vehicle, on a global scale, began in the mid-1950s, and its development has been primarily based on the ballistic missile technology. During this stage, the United States and the Soviet Union occupied the leading position in the space competition. They solved the zero-to-one problem and developed several launch vehicle series with complex configurations.

The early American launch vehicles primarily include Juno, Thor, Delta, Atlas, Titan, and Vanguard. The first five models have been developed on the basis of the Jupiter medium-range missile, the Thor medium-range ballistic missile, the Atlas intercontinental ballistic missile, and the Titan intercontinental ballistic missile. Further, they were equipped with different upper stages such as Agena and Centaur, thus forming various different launch vehicle series. Concomitantly, based on the demands of the Apollo moon landing, the United States has developed the Saturn series on the basis of the newly developed high-thrust engines and large-diameter rocket bodies, including three types of Saturn I, Saturn IB, and Saturn V. During 1969–1972, a total of seven manned moon missions were carried out.

The Saturn V launch vehicle adopts a three-stage configuration. The first stage employs five high-thrust F-1 liquid oxygen and kerosene engines, and the second and third stages use five and one J-2 liquid hydrogen and liquid oxygen engines, respectively. The full length is 110.6 m, the maximum diameter 10.06 m, the take-off mass 2946 t, the carrying capacity in low-earth orbit 120 t, and the carrying capacity in the earth-moon transfer orbit 50 t. Among them, F-1 is the liquid launch vehicle engine with the highest single-nozzle thrust in history, and its ground thrust has been 6806 kN.

Since 1957, on the basis of strategic ballistic missiles, the Soviet Union has successfully developed Sputnik, Luna, Vostok, Voskhod, Soyuz, Molniya, Kosmos, Tsyklon, and other launch vehicles. It has also launched a large number of satellites, and manned or unmanned spaceships, space stations, moon probes, and Mars and Venus probes, and other spacecrafts, thus creating multiple world records in human spaceflight. The first six models belong to the R-7 launch vehicle family, which is a series of launch vehicles derived from the first intercontinental ballistic missile, viz., the Soviet R-7. The Sputnik is the first launch vehicle for the first artificial earth satellite launched by the Soviet Union. Its basic stage is the most launched one, for any launch vehicle in the world, hitherto. It consists of a core stage bundled with four liquid boosters using liquid oxygen and kerosene propellant, and is equipped with different second and upper stages, thus, forming a huge family of launch vehicle. To compete with the United States in the human lunar mission, the Soviet Union developed the N-1 heavy-lift launch vehicle, though all of the four launches failed. Furthermore, the Soviet Union also developed the Proton series, which employ the conventional propellants to launch large spacecrafts such as the Proton Satellite and the space station cabins.

Meanwhile, in Europe, France, and the United Kingdom began to develop their own launch vehicles in 1960. France developed the Diamant series on the basis of the sounding rockets and missiles. The United Kingdom developed the Black Arrow and Blue Streak launch vehicles. Concurrently, several European countries established the European Launcher Development Organization and developed the Europa series of launch vehicles.

Japan developed the L-series and M-series of launch vehicles based on the sounding rocket technology, and launched its first artificial earth satellite with the L-4S launch vehicle, in February 1970.

Meanwhile, China developed the Long March (LM) 1 and 2 launch vehicles. In April 1970, the LM-1 launch vehicle successfully launched the first artificial satellite of China, i.e., Dongfanghong-1, thus, laying a solid foundation for the space industry of China.

1.1.2 Space Shuttle Stage (1970–1990s)

Since 1970s, the reusable launch vehicle technology has been favored. Both the United States and the Soviet Union have focused on the development of reusable aerospace transportation system. The hazardous implementation of the program has also affected the development of expendable launch vehicles. They primarily improved the carrying capacity and reliability, besides promoting the advances in technology.

By 1972, the United States concentrated on the development of partially reusable space shuttles, expecting a significant reduction in space launch costs through the reuse of vehicles, to eventually replace the expendable launch vehicles. The space shuttle consists of an external storage tank, a solid booster and an orbiter, among which the solid booster and orbiter can be recycled and reused. The space shuttle first flew in 1981. In 1982, the U.S. government announced that it would replace the expendable launch vehicles such as the Atlas, Delta, and Titan, with the space shuttle to launch all U.S. payloads. The painful lessons from the Space Shuttle Challenger accident in 1986, persuaded the U.S. government to resume the use of expendable launch vehicles, thus establishing a vehicle team consisting of space shuttles and expendable launch vehicles. Prior 1982 and post 1986, the United States upgraded and improved the Atlas, Delta, and Titan launch vehicles by improving the engine performance, bundling solid boosters, lengthening the storage tank, and replacing the upper stage. Further, they formed the series of Atlas G, Atlas H, Atlas I and Atlas II, Delta 1000, Delta2000, Delta3000 and Delta II, Titan III/IV and Titan IV series, and other launch vehicles to enhance the carrying capacity comparable to the space shuttle. Among them, the Delta II launch vehicle became the primary medium-lift launch vehicle of the United States during 1990s.

The United States produced a total of six space shuttles. However, owing to the consideration of the aging and safety of the space shuttles, besides the high cost, complex operation, and long operating cycle, they finally have decided to announce the decommission of the Space Shuttle after the final flight in 2011.

By December 1971, the Soviet Union began research and development on the reusable space transportation systems. Further, by exploiting their own advantages in the liquid launch vehicle engines, they began developing the Energia heavy-lift launch vehicle by 1976. The launch vehicle has a full length of 58.7m, a diameter of 7.75m, a takeoff mass of about 2220t, a takeoff thrust of about 3616t, and a carrying capacity in low-earth orbit of about 100t. The primary task is to transport the reusable orbiter, i.e., Buran Space Shuttle, and launch large payloads into the low-earth space. The Buran Space Shuttle made only one unmanned flight. Concurrently, the Soviet

Union carried out a standardized upgrade of the Soyuz launch vehicle to form the Soyuz U and U2 launch vehicles, with improved carrying performance. Further, they developed the Zenit 2 launch vehicle using the technology of the Energia launch vehicle engine. It is mainly used in the domestic military satellites and has rapid response capabilities. However, subsequent to 1990s, the Zenit series have gradually been utilized for international commercial space launches.

Since 1973, Europe began to develop Ariane launch vehicles based on the Europa and Diamant launch vehicles. By the early 1990s, the Ariane launch vehicles have steadily developed from Ariane 1 to Ariane 2, Ariane 3, and Ariane 4 by improving the performance, lengthening the propellant storage tank, bundling solid or liquid boosters, and increasing the number of bundled boosters. The carrying capacity has been quadrupled, and the payload adaptability was also greatly improved.

To master the liquid launch vehicle technology for launching large satellites, and obtain medium, high, and geosynchronous orbit satellite launch capabilities, Japan adapted the Thor-Delta launch vehicle technology from the United States in the 1970s, and hence developed the N series launch vehicles. Further in the 1980s, they developed the H-I launch vehicles with a higher payload capacity.

Meanwhile, China primarily developed the LM-2C and LM-3 launch vehicles. Among them, the LM-2C has been improved on the basis of the LM-2, which in turn, enhanced the carrying capacity and reliability to a certain extent. LM-3 is a three-stage launch vehicle developed on the basis of the LM-2C launch vehicle. Its three stages use liquid hydrogen and oxygen cryogenic propellant, and are primarily used to launch geosynchronous orbit satellites.

India began to develop the four-stage solid launch vehicle, Satellite Launch Vehicle 3 (SLV-3), on the basis of the sounding rockets in 1973, and successfully sent the Rohini Satellite into low-earth orbit in 1980, and then developed Augmented Satellite Launch Vehicle (ASLV) on the basis of SLV-3.

1.1.3 Commercial Service Stage (1990–2010s)

By 1990s, the launch vehicle technology entered a new stage of development. Competition in the international commercial launch market has intensified, and the launch demands for high-mass communication satellite have increased. Most nations have begun the development and improvement programs for low-cost launch vehicles.

The U.S. Air Force began to implement the development program of Evolved Expendable Launch Vehicle (EELV) in 1994. The goal has been to reduce launch costs, improve the reliability, and the capture of international commercial launch market while meeting the domestic launch demands. The program finally gave birth to the series of Atlas V and Delta IV launch vehicles, and both of which have fully inherited the advantages of the previous models. Further, they adopted modular design ideas, and advanced power and control technology, to reduce the launch costs and improve the carrying capacity, reliability and operability, besides replacing the original launch vehicle model at the beginning of the 21st century. The primary

common modules of the Atlas V series include the common core stage, common centaur upper stage, solid binding booster, and payload fairing. The common core stage uses one RD-180 engine, which can be categorized into 400 and 500 series according to different combinations. They can meet the launch demands of various medium-lift payloads, besides taking into account the launch requirements of the U.S. military's payloads. The Delta IV series employs a large-diameter common core (5.08 m) bundled with different types of boosters, and achieves multiple carrying capacities through modular combination, including Delta 4 medium-lift, modified Delta IV medium-lift and Delta IV Heavy. The common core stage employs the newly developed low-cost RS-68 engine, and both the common core stage and the second stage use liquid hydrogen (oxygen) propellant.

Following the disintegration of the Soviet Union in 1991, Russia made improvements over the Proton and Soyuz launch vehicles, forming multiple launch vehicles such as Proton M, Soyuz ST, and Soyuz 2. Further, they equipped them with Breeze M and Fregat upper stages, and debuted them into the commercial launch market. Russia has also altered its decommissioned and reduced strategic ballistic missiles into launch vehicles such as Start, Rockot, Dnepr, Volna, Strela, and Shtil, and introduced them into the commercial launch market. Furthermore, Russia began to develop environmentally friendly, non-toxic, and advanced-performance Angara series, based on the idea of Generalization, Serialization, and Combination, by 1994. By this step they hoped to replace the major existing launch vehicles of Russia with Angara series. Nevertheless, only the Angara 1.2 and 5 configurations have performed a total of three launch missions, hitherto.

Europe has developed the Ariane 5 in accordance with the demands of the commercial launch market, besides the development and utilization of low-earth orbit. It is the world's first high-thrust launch vehicle designed with a large-diameter and less-stage scheme, and has been continuously improved with respect to the performance of the launch vehicle. Ariane 5 series including Ariane 5G, 5ES, 5ECA, 5ECB, and other models have been formed, which can perform single-satellite-in-one-launch and multiple-satellites-in-one-launch missions, and gradually become the driving force in the international commercial launch market. Concurrently, Europe started the development of the Vega small launch vehicle as a supplement to the Ariane 5 and Soyuz, for launching small governmental and commercial payloads. Vega first flew in 2012. Meanwhile, to fill the gap in the medium-lift carrying capacity, the Russian Soyuz ST has been introduced.

The Ariane 5 series are all in the two-stage configuration. The core stage is equipped with one Vulcan liquid oxygen and hydrogen engine, bundled with two solid boosters, and the carrying capacity for geosynchronous transfer orbit (GTO) can reach 6.9–10.5 t.

Japan has developed the H-IIA series and H-IIB based on the H-II launch vehicle through modular design. This is to meet the diversified launch demands and enhance its competitiveness in the space launch market, and improve the reliability of launch vehicle. The H-IIA series are all in the two-stage configuration bundled with different numbers of boosters, including HII-A202, H-IIA2022, H-IIA2024, and H-IIA204. The maximum carrying capacity for GTO is 5.7 t. The first stage is powered by one

LE-7A liquid oxygen & liquid hydrogen engine, and the second stage is powered by one LE-5B liquid oxygen and hydrogen engine. The H-IIB launch vehicle has an increased diameter in the first stage, viz., from 4 m to 5.3 m on the basis of H-IIA, and installs two LE-7A engines in the first stage, besides the carrying capacity for GTO reaching 8 t.

China has introduced the developed LM-2E, LM-2C/SM, and LM-3A series of launch vehicle into the international commercial launch market, and successfully developed the LM-2F launch vehicle. LM-2F is a two-stage launch vehicle bundled with four boosters, based on LM-2E and developed in accordance with the mission requirements of human spaceflight. The reliability index reaches 0.97 and the safety index reaches 0.997. The first flight was on November 20, 1999, and it successfully launched the Shenzhou-1 test spacecraft. The first successful human launch was on October 15, 2003.

During this stage, India has successively developed a Polar Satellite Launch Vehicle (PSLV) and a Geosynchronous Satellite Launch Vehicle (GSLV), which are capable of launching medium, large, and geosynchronous orbit satellites.

1.1.4 Comprehensive Performance Improvement Stage (2010s–Now)

Since 2010, the United States, Russia, Europe, and Japan, based on the long-term development goals, are actualizing the upgrades through the layout in developing the next generation of launch vehicles. They are based on the development principles of modularization, serialization, and combination, besides benefiting from the improvement of economy. Further, they have utilized the advanced technology such as the reusability and artificial intelligence, to enhance the comprehensive market competitiveness of the launch vehicle [2, 3].

Aided by the efficient management and technical innovation, SpaceX has pursued research and application on vertical take-off and landing technology, and gradually occupied more than half of the international space commercial launch market with marked price advantages for Falcon 9 and Falcon heavy. Its primary focus is on the development of the Superheavy-Starship transportation system. Superheavy-Starship aims to actualize the future airline-flight-mode space transportation. It is a common vehicle that can perform services such as global rapid transportation, space shuttle transportation, earth-moon transfer transportation, lunar landing and ascent, forecasting Mars exploration transportation. It has been selected by NASA as the lunar lander in the Artemis program. Affected by multiple-factors such as the risk of outsourcing Russian engine supply and the return of American manufacturing, ULA initiated developing a Vulcan launch vehicle with certain intelligence. It has achieved different carrying capacities by bundling multiple solid boosters and upgrading upper stage performance, and is intended to significantly reduce the launch costs, compete with SpaceX, and eventually replaced the EELV launch vehicle. Furthermore, Blue

Origin is also developing the New Glenn launch vehicle during the same period. Concurrently, United States is fast developing the Space Launch System (SLS) heavy-lift launch vehicle to support its Artemis program and manned deep space exploration strategy [4].

To enhance the competitiveness, Russia has proposed Soyuz 5 and 6 launch vehicles. Both of them adopt a simple two-stage serial configuration and employ the first stage of liquid oxygen and kerosene, and a second stage with liquid oxygen and kerosene, or liquid hydrogen and liquid oxygen. They will replace the Soyuz 2 series of launch vehicles of similar capabilities in the future. Furthermore, the first stage of the two launch vehicles can also be utilized as the core stage or the booster module of the Russian heavy-lift launch vehicle. As a medium-lift launch vehicle, the Soyuz 5 and 6 have the characteristics of a simple configuration and strong modularization (generalization). Technically, they have the potential to be the primary force of the next-generation medium-lift launch vehicles of Russia. Russia has also proposed the Amur reusable launch vehicle scheme that employs the liquid oxygen and methane propellant, and the first stage can be recycled and reused vertically.

Europe and Japan have proposed new launch vehicle updating programs, viz., Ariane 6 and H-III, respectively, in response to the stiffening competition in the international commercial launch market. Both of the programs factor in cost reduction as the first priority, conduct development in response to the market demand, and avoid pursuing exclusive technical advancement. The Ariane 6 inherits most of the design basis and mature technology of the Ariane 5, and will replace the Ariane 5 and Soyuz ST in future [5]. Meanwhile, with the rapid development of small satellite market, some commerical aerospace companies develope various small reusable rocket, such as Spanish launch startup PLD Space's Miura 1 reusable suborbital rocket and British startup Orbex' Prime launch vehicle, holding low-cost and rapid-launch features.

H-III launch vehicle of Japan fully inherits the mature technology of the H-IIA/IIB, while focusing on improving the design in reducing costs and improving reliability. The first stage uses two to three LE-9 liquid oxygen and hydrogen engines. Both the second stage and the solid booster have been improved with respect to the H-2A/2B. Different numbers of boosters are bundled to achieve different carrying capacities. The carrying capacity for GTO can reach 7t. Concurrently, Japan has developed the Epsilon solid launch vehicle based on the concept of intelligent measurement, launch, and control to significantly reduce the number of ground testers and the launch costs [6].

Aiming the international advanced level and the demands of launch vehicle upgrading, China has successfully developed a new generation of launch vehicles based on 120 t liquid oxygen and kerosene engines and 50 t liquid oxygen and hydrogen engines, covering the launch mission demands of low, medium, and high orbit spacecrafts. The new-generation launch vehicles of China is a series of vehicles according to the design concept of Generalization, Serialization, and Combination, founded on "one series, two engines, and three modules", and aiming at the reality and urgent demands of the space development of China. This includes LM-5/5B built based on 5m diameter modules, LM-6, LM-7/7A, and LM-8 built based on 3.35 m diameter modules. The maximum carrying capacity in low-earth orbit reaches 25 t.

The maximum carrying capacity for GTO is 14 t. These launch vehicles first flew successively in 2015-2021. Among them, the LM-5 launched the Tianwen-1 probe and the Chang'e-5 probe into space, and achieved the first Mars landing &patrol and lunar sampling & return of China. Further, the LM-5B launched the China Space Station modules to the low-earth orbit. Furthermore, China is currently making technical improvements to launch vehicles based on the demands of low-cost and high-reliability, and preparing for future upgrading, besides concurrently developing the reusable technology and super heavy-lift launch vehicles.

1.2 Development Prospect of Launch Vehicle

With the development of human society, the space field has become an important territory for human survival and development, particularly in the 21st century. The demand for various space missions is fast increasing, the space activities become more and more frequent, and the big space era is coming.

(1) Mankind's dependence on space is increasing on a daily basis, and space technology is playing an important role.

Since the 1960s, hundreds of space science and exploration missions have been carried out around the world. Human footprints have spread across the eight planets of the solar system, and scientific cognition has engendered new breakthroughs. With respect to satellite applications and services, the communication satellite system has steadily evolved, and the accuracy of the navigation satellites has grown manifold. Further, the public service capabilities of remote sensing satellites have been enhanced, thus providing humans with highly accurate monitoring and warning of weather, environment, and disasters. The economic scale of the space industry is burgeoning with impending expansions.

Predictably, the space field will be an important territory for human survival and development in future, whereas the cislunar space serves an outpost for human exploration of space. The space industry, owing to its expansion, will also take the lead in entering a new era of cislunar economy.

(2) The current world space missions have developed to a new stage of large-scale access to space.

Recently, the demands for global ubiquitous network access and Internet of Things connection services have been on the rise, and the development of low-orbit communication have accentuated it. Nearly tens of thousands of giant communication constellation projects represented by Starlink, OneWeb constellation, and Hongyan constellation have emerged in sequence, and the development of low-orbit Internet, satellite Internet of things, and other fields will accelerate. Concurrently, with the development of space technology, and the demand for a large-scale space infrastructure represented by space power stations and space factories, besides human lunar exploration and construction of lunar bases, have become increasingly strong. Space missions have eventually developed to a new stage represented by large-scale Inter-

Table 1.1 Forecast of scale demand for access to space in 2045

Space activities	2045/t
Communication/navigation/remote sensing	3000
Space science exploration and experiment	4000
On-orbit service and maintenance	4000
Space tourism and global rapid transportation	40000
Space key infrastructure	5000
Deep space exploration	2000
Space energy	40000
Resource exploration and utilization	10000
Space-based warning	2000
Space safety	4000
Space medicine/agriculture/manufacture	5000
Space environmental monitoring and warning	1000
Others	5000
Subtotal	125000

net constellations, large-scale space resource development and utilization, large-scale lunar exploration, and large-scale deep space exploration. According to the course of the global space industry, and based on the current foundation, it is estimated that by 2045, the global annual scale demand for access to space will exceed 100,000 tons, as listed in the Table 1.1.

(3) Commercial spaceflight is advancing rapidly, and the space industry is showing a new direction of development.

Owing to the relentless development of space technology and the increasing scale of the space industry, commercial spaceflight such as commercial launch vehicles, low-orbit Internet constellations, and commercial remote sensing have been rapidly promoted. The space industry is displaying a new course of development, and will eventually expand to the fields of space tourism, global airline-flight-mode rapid transportation, space resource development, energy utilization, on-orbit manufacturing, medicine, and health in the future.

With the continuous development of space technology, the scale of future space missions will become colossal, and the scope of space applications and services will become extensive, besides a rampant commercialization of the space industry. The space industry will be more integrated with human society, economy, and livelihood of people. To encounter the demands of the large-scale access to space in the future, the Space Transportation Revolution needs to be implemented. A Revolution Era of Space Transportation is impending, which has the following basic characteristics [3].

- Manifold increase of carrying capacity;
- Substantial Cost reduction;
- Enhancement of Reliability and safety;
- Profound effect on society and life;
- Intelligent delivery vehicles;
- Airline-flight-mode operation and management;
- Fast launch, convenient, and flexible;
- Form a large-scale industry.

The new demand also brings many challenges. It is necessary to adopt the latest scientific and technological achievements to the maximum in the existing ways of access to space. Concurrently applying and cultivating the reusable technology and intelligent autonomous technology, to meet the future demand for large-scale and low-cost access to space.

1.3 Current Development Status of Launch Vehicle Reusable Technology

From the perspective of the history of launch vehicles, expendable launch vehicles are still the mainstream method for nations to enter space. However, the era of the large-scale and low-cost access to space, particularly with the successive proposal of the airline-flight-mode space transportation system and the mission concept of global arrival within an hour, leads to the rapid development of reusable technology.

Reusable launch vehicles can greatly reduce the cost of access to space through multiple uses and cost sharing, which is the course of development for vehicles in the future.

According to the current development situation, reusable space transportation systems can be divided into two categories, viz., axisymmetric configuration and lifting body configuration.

1.3.1 Reusable Space Transportation System in Axisymmetric Configuration

The reusable transportation system based on the traditional axisymmetric configuration primarily recycles and reuses its stages. Therefore, the stage recycling is a major issue. According to the recycling method of the stage, it can be categorized into the parachute reusable launch vehicles and the vertical take-off and vertical landing (VTVL) reusable launch vehicles. Furthermore, fairing recycling has also begun to be practiced and applied.

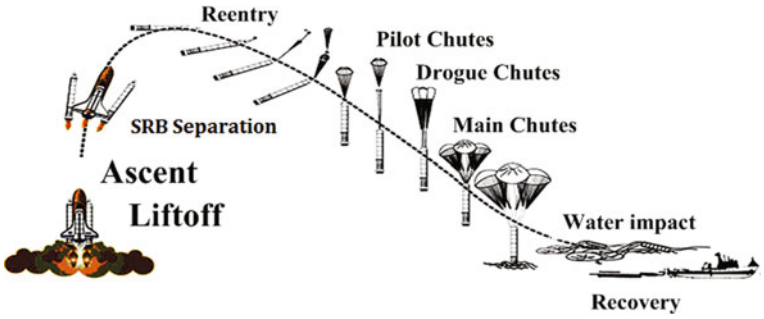


Fig. 1.2 Space shuttle SRB recovery process

1.3.1.1 Parachute Recovery Reusable Launch Vehicle

For parachute recovery reusable launch vehicles, following the completion of the mission of the launch vehicle, the parachute is deployed to decelerate the vehicle during the return, to a speed of tens of meters per second, and finally, to fall on land or at sea for recycle and reuse. The examples are Kistler's K-1 launch vehicle, NASA's space shuttle booster, and ULA's Vulcan launch vehicle.

(1) Parachute recovery of the space shuttle booster at sea
The recovery of the solid rocket booster (SRB) of the space shuttle uses a large group parachute. The SRB unit integrates the ascent, reentry, and recovery subsystems. The integrated booster subsystem includes the thrust vector control, auxiliary power unit, avionics, pyrotechnic signal, range safety system, parachute, thermal protection, and water recovery system. The technical difficulties of SRB include the subsystem integration, thermal environment and harsh load environment (including falling water impact). Multiple subsystems have been improved to meet the reusable requirements. Each booster deploys three main parachutes to slow down and finally land on the sea. The SRB recovery process is shown in Fig. 1.2 [7].

(2) Parachute recovery of K-1
Kistler's K-1 launch vehicle program started in 1993, expecting to reduce the launch cost of the launch vehicle through reusability, for commercial launches. K-1 is a two-stage fully reusable vehicle, and the first stage adopts a recovery scheme based on parachute and cushion airbag. Following the release of the payload, the second stage reentry and return to the launch site also employ the recovery scheme based on parachute and cushion airbag.

Owing to the development of the two-stage fully reusable vehicle being hazardous, and the unsure project funding, the development plan of K-1 is fluctuating. Despite certain tests and verifications, it has not been introduced for practical applications. However, the vast majority of development tests on recovery and landing system have been completed, and the feasibility of the recovery approach based on group parachute and airbag has been verified through the demonstration and verification test on the aircraft. The K-1 recovery process is shown in Fig. 1.3 [8, 9].

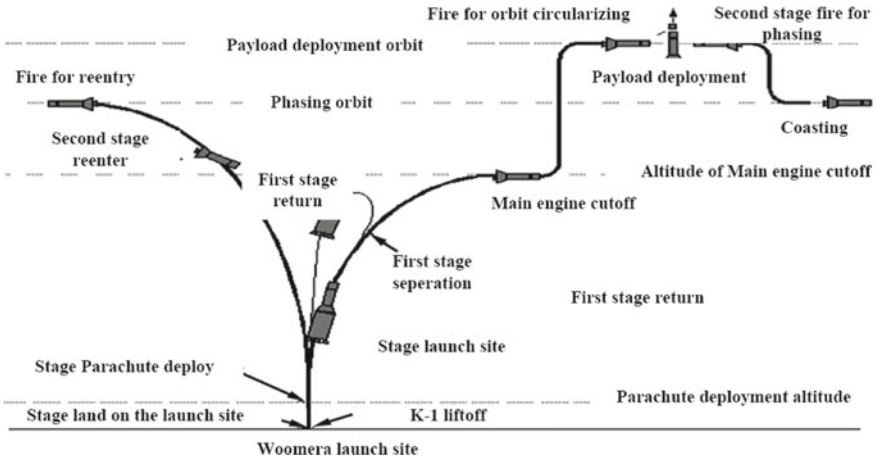


Fig. 1.3 The K-1 recovery process

(3) Vulcan aerial recovery

In April 2015, ULA announced the Vulcan program of the heavy-lift launch vehicle. The Vulcan will use the Sensible Modular Autonomous Return Technology (SMART) technology to achieve the recovery and reuse of the stage.

Subsequent to the separation of the first and second stages, the first stage engine of the launch vehicle will be separated from the first stage and re-enter the atmosphere under the protection of an inflatable heat shield. It will be decelerated by the parachute, and finally recovered in the air by a helicopter. ULA stated that the cost of the propulsion system accounts for 65% of the total cost of the first stage, and the recycling of the first stage engine will reduce the cost of the first stage propulsion system by 90%. ULA claimed that the SMART recovery project is only the beginning of the launch vehicle recovery program of the company. In future, ULA will also recover other launch vehicle components to further reduce the launch costs.

(4) Fairing recovery of Falcon 9

The fairing generally adopts an extremely light and thin carbon fiber sandwich structure, which mandates precise manufacturing and testing before application. The production cost accounts for about 10% of the total cost of a single launch. Generally, after the launch vehicle reaches the Karman Line at a height of 100 km, the rocket becomes unaffected by the atmosphere, and the fairing automatically separates and falls. If the fairing can be recovered for re-launch, it will further reduce the cost of manufacturing and launch.

SpaceX leads the research and practice of the fairing recovery and in May 2017, for the first time, actualized the controlled fairing splashing down in the ocean, which employed ships for salvage and recovery. Since the internal structure does not account for the corrosion resistance requirements, the fairing recovered after splashing down in the ocean must be processed before reuse, which increases the cost and difficulty

Fig. 1.4 The launch vehicle whose first stage is for parachute recovery

of recycling. To handle this issue, SpaceX uses modified vessels for the direct capture and recovery to prevent the fairing from contact with seawater. On June 25, 2019, after the deployment of the equipment of cushion nets on ships, the net capture of fairing in the air was realized for the first time. This denotes that SpaceX has successfully verified the two methods of fairing recovery. However, owing to the relatively low probability of the net capture of the fairing in the air in actual flight missions, the scheme for net capture in the air has been cancelled.

(5) The aerial recovery of the first stage of the Rocket Lab's Electron launch vehicle On August 7, 2019, Rocket Lab, co-sponsored by New Zealand and the United States, proposed a program for the first stage recovery in the air for its Electron small launch vehicle.

Electron is a two-stage launch vehicle with a length of 17 m, a diameter of 1.2 m, a take-off mass of 10.5 t, and a carrying capacity in 500 km sun-synchronous orbit (SSO) of 150 kg. The first stage of the launch vehicle employs nine Rutherford engines, and the second stage employs one vacuum Rutherford engine. The Electron utilizes advanced carbon fiber composite materials to design a high-strength and light-weight flight structure. The first stage structure is light in weight and suitable for the helicopter hooking recovery in the air. Following the separation of the first stage, the main engine stops decelerating, or deploys inflatable airbags to decelerate. Finally, a two-stage parachute is deployed to decelerate. The main parachute is a parafoil, and the helicopter is employed for the aerial hooking recovery in the air.

(6) Parachute recovery technology in China
In terms of the parachute recovery of launch vehicles, China has completed the study on the parachute recovery scheme and the airdrop test and verification for the liquid oxygen and methane first stage.

The launch vehicle whose first stage is for parachute recovery is a small two-stage liquid launch vehicle with a total length of about 29 m. The overall layout is shown in Fig. 1.4. The take-off mass is about 100 tons. When launched from the Jiuquan launch site, the carrying capacity for 700 km SSO exceeds 650 kg.

To reduce the launch cost of the launch vehicle, the first stage is recovered and reused. The recovery landing system adopts the scheme of two stabilizing parachutes and two decelerating parachutes, besides three main parachutes. This decelerates the separated first stage, and employs the airbag system arranged at the front and rear ends of the first stage to cushion the landing process, thus, reducing the impact load in the landing process and the damage to the vehicle body caused by the impact load. The cushion airbag is a combination of inner and outer airbags. The outer airbag cushions the landing process of the first stage. The inner airbag is encapsulated in

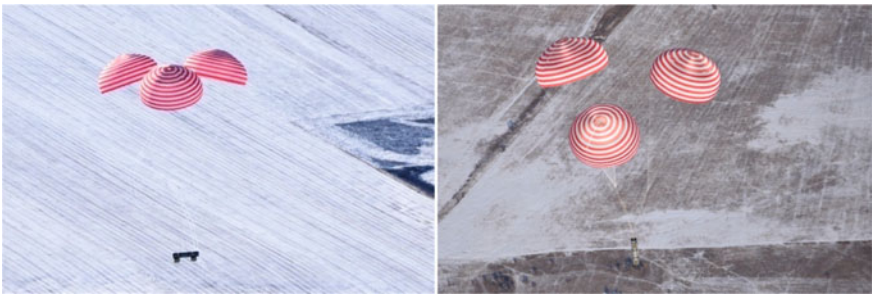


Fig. 1.5 The airdrop test for the parachute recovery of first stage

the outer airbag, and functions as a support to the first stage of the launch vehicle following the landing, so that the first stage does not directly impact the ground.

China has also completed the airdrop test for a group parachute recovery of the launch vehicle stage, and has developed the scale model for airdrop test of the first stage of launch vehicle and the scale test prototype of the group parachute system. The airdrop test uses a helicopter as the airdrop platform, and carries and releases the scale prototype of the group parachute and cushion airbag system, and the scale model of the first stage. In the test, the decelerating parachute and the main parachute have been deployed in sequence, and hence, reached a steady falling speed. Finally, the scale model of the first stage has been brought to the ground. The airbag deployment and landing cushion have been normal, and the stage model was successfully recovered (Fig. 1.5).

1.3.1.2 VTVL Reusable Launch Vehicle

For VTVL reusable launch vehicle, following the completion of launch mission, it restarts the main engine to decelerate the stage during the return process, and finally employs the landing legs to accurately land on the predetermined position, for e.g., SpaceX's Falcon 9 and Blue Origin's New Glenn.

The earliest use of the vertical return recovery was the Delta clipper scheme proposed by McDonnell Douglas in the 1990s. It was designed to utilize the VTVL technology to achieve single-stage-to-orbit and reuse. Delta clipper has conducted a total of twelve flight tests, with a maximum flight altitude of 3155 m, and verified the technologies of the VTVL, fast flyback, and simplified ground support [10].

The first stage of the Falcon 9 adopts the VTVL scheme, and the first stage will return to the launch site from the sub-orbit after the stage separation, or land vertically down range. To develop the reusable Falcon 9, SpaceX has developed a diversified and progressively-developing verification plan of the reusable technology. The recovery operations for the first stage of Falcon 9 have carried out a large number of experimental verifications, and has successfully carried out dozens of offshore platform and land recovery. A single first stage module has performed up to eleven

launch missions (by December 2021). Concurrently, on the basis of the Falcon 9, two core stages have been added to form the Falcon Heavy. All three core stages can be landed vertically and reused.

SpaceX's Superheavy-Starship system applies the VTVL reusable technology for the design of the heavy-lift launch vehicles, and it is expected to become the world's first two-stage fully reusable heavy-lift launch vehicle. Since the launch vehicle has been proposed in 2016, its scheme has been iteratively optimized. Following the transition of the Big Falcon Rocket (BFR), the combined scheme of Superheavy and Starship was finally determined in 2018.

While continuously optimizing the scheme, SpaceX has adopted a model of prototype verification iteration and dual-line research and development (R&D), to achieve a rapid verification and optimization of the related technologies. Since 2019, it has experienced tests such as the Starhopper series, MK series, and SN series, and achieved a 10-km high-altitude flight test in the SN15 test. Currently, it is advancing its work towards the direct flying into the orbit.

Furthermore, Blue Origin has also proposed the New Glenn series of launch vehicle which adopts the vertical return recovery of the first stage.

The New Glenn has two-stage and three-stage configurations. The diameter of the launch vehicle is about 7m. The first stage is equipped with seven BE-4 liquid oxygen and methane engines with a thrust of 17000 kN. Further, the first stage is separated in the ascending section and returns vertically to land on the offshore platform for recovery. Unlike the Falcon 9, which uses a grid fin, the recovery process employs an aerodynamic control surface to implement the aerodynamic control. Moreover, Blue Origin's suborbital human launch vehicle of the New Shepard, has successfully completed multiple recovery and reuse.

DLR, CNES, and JAXA are jointly developing the CALLISTO aircraft based on their experience in the vertical landing technology. All the three organizations will use the aircraft to verify their own guidance and control algorithms. The maximum flight altitude and the speed can reach 40 km and Mach 2, respectively. On this basis, the European Space Agency (ESA) will develop the Themis launch vehicle. As the milestone verification project of the European reusable technology, it is categorized into two processes, viz., single engine and three engines. Its maximum speed can reach Mach 6–8, and it can simulate the whole process of the dynamic deceleration and aerodynamic deceleration when the launch vehicle enters the dense atmosphere at high speed. The proven key reusable technology will be applied to the Ariane Next series of launch vehicles.

In 2020, the Roscosmos officially released its new generation of commercial reusable launch vehicle Amur. The Amur launch vehicle has a height of 55 m, a takeoff mass 360 t, and a diameter 4.1 m. It uses liquid oxygen and methane propellant with a two-stage configuration. The first stage of the launch vehicle takes off and lands vertically, and is equipped with five RD-0169A engines with a sea level thrust of 100 t. The second stage of the launch vehicle is expendable, and uses a vacuum version of the RD-0169V engine, same as the first stage, with a vacuum thrust of about 110 t. The first stage of the launch vehicle is initially designed for 10 times reuse, and the long-term design goal is 100 times.

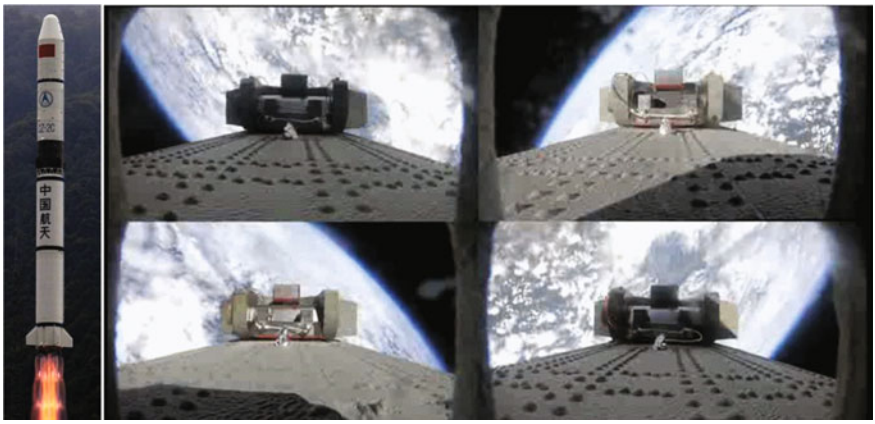


Fig. 1.6 Flight test for the gridfin based landing zone control (LM-2C)

China is also actively carrying out research and application of VTVL key technologies. Currently, it has broken through the manufacturing and control technology of titanium alloy integral casting grid fin, and carried out a flight test and verification of the landing zone control for the first stage of a LM-2C in 2019 (Fig. 1.6).

The LM-8 has carried out research on key reusable technologies with the goal of vertical recovery [11]. To verify the key technology, powered by the UAV turbojet engine, a small aircraft Peacock has been designed, and the verification of the guidance and control algorithm has been carried out. The composition of the Peacock aircraft is displayed in Fig. 1.7.

The slenderness and thrust-to-weight ratios of the Peacock aircraft simulate the parameters of the LM-8. All electronic equipment adopts commercial off the shelf (COTS), and the flying altitude is controlled near 400 m. This verifies the adaptability of the convex optimization algorithm in the embedded computing environment.

Based on the second stage of the LM-7, a VTVL flight test has been designed. It is powered by liquid oxygen and kerosene engine, and the flight altitude is controlled at 2 km. The aircraft operates four engines during the ascent phase. Two of the engines are turned off at the highest point, and the remaining two continue delivering the power for reverse thrust. Furthermore, to verify the guidance and control algorithms, the following technologies of the aircraft will be tested, such as the landing cushion mechanisms, relative navigation, and rapid post-processing of liquid oxygen and kerosene engines after landing. Among them, the landing cushion mechanism adopts a triangular overall scheme, which has multiple functions of folding and retracting, landing cushion, deployment deceleration, and locking and bearing. Its composition is displayed in Fig. 1.8.

Since the existing Long March launch vehicles in service do not employ multiple engines in parallel, the guidance and control algorithm [12] under the condition of limited throttling capability, which causes high thrust-to-weight ratio of the landing stage (thrust-to-weight ratio greater than 2.0) has been studied. Under the premise

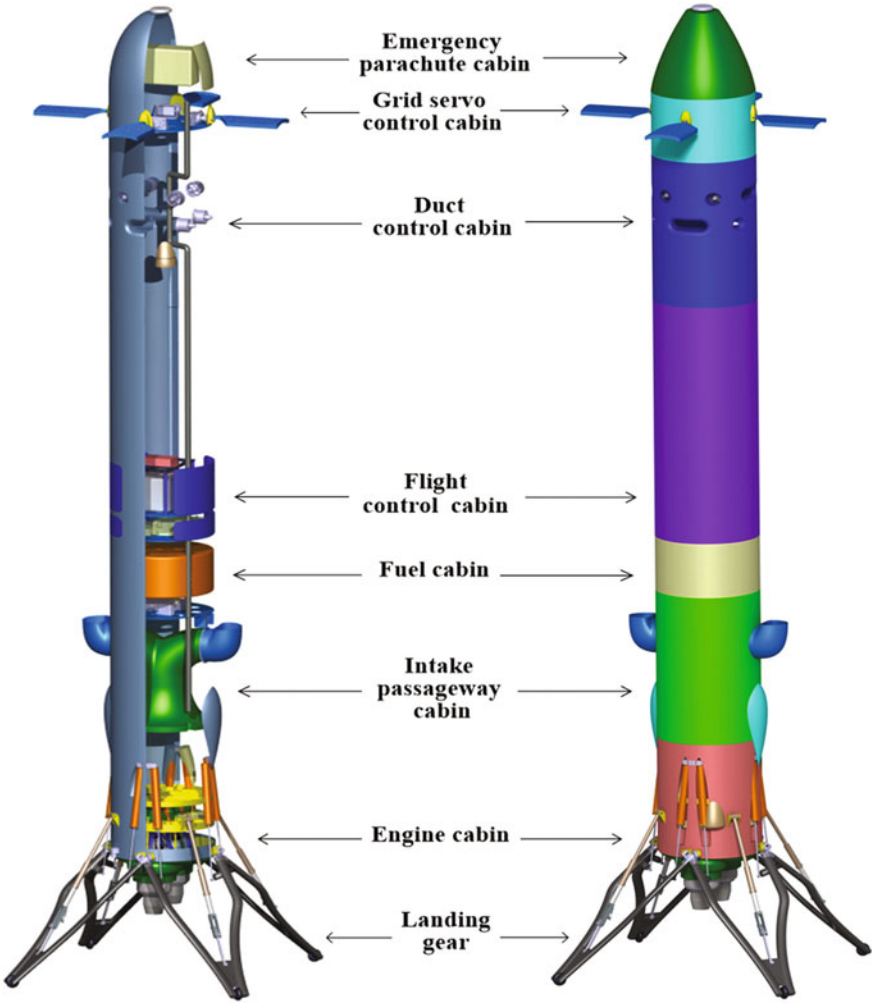


Fig. 1.7 The Peacock Testbed for the verifications of GNC algorithm

of unchanging engine configuration, the foundation has been laid for the upgrading of the in-service launch vehicles to reusable launch vehicle.

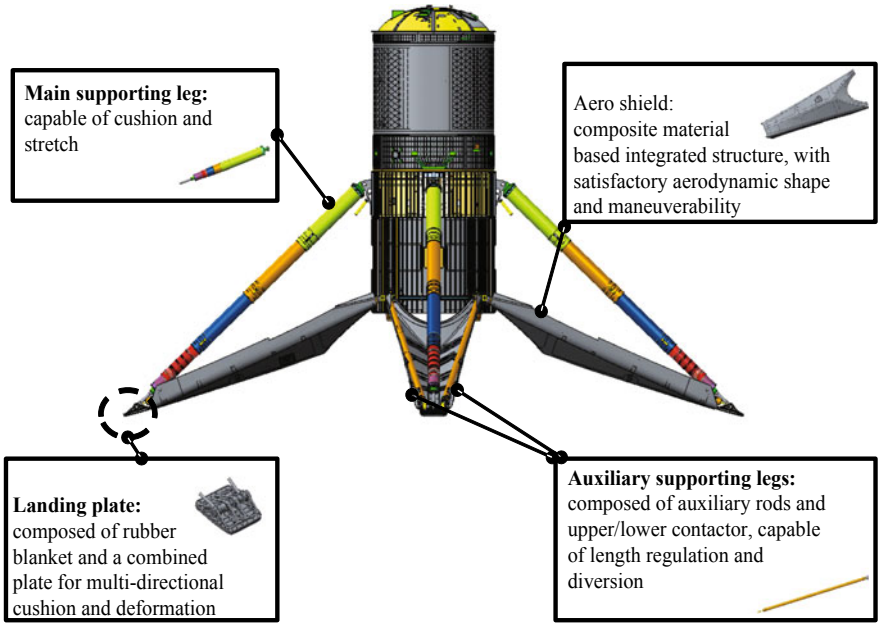


Fig. 1.8 Supporting legs of the Peacock aircraft

1.3.2 Reusable Space Transportation System in Lifting-body Configuration

1.3.2.1 Rocket Powered Lifting-Body Reusable Launch Vehicle

(1) Reusable first stage

Reusable first stage refers to a vehicle with autonomously controllable returns, lands horizontally, and is reused after transporting the payload from the surface of the earth to the suborbital space. It is the first stage of a two-stage orbiting space transportation system. Typical launch vehicles include the Reusable Booster System (RBS) and XS-1 from the United States, the Baikal from Russia and the RLV-TD from India.

The RBS is a reusable vehicle development program launched by the U.S. Air Force to realize the operational responsive space (ORS) capability. The program is directed by the U.S. Air Force Research Laboratory (AFRL). The U.S. Air Force has demonstrated that the RBS concept is a reliable method to meet its future demand of rapid access to space [11].

The RBS system uses a vertical launch mode. The first stage and the upper stage fly to the separation point and then return to the launch site, thus landing on the runway like an airplane.

The U.S. Air Force is contemplating the use of rocket-powered boosting to implement the return of the first stage. Following the separation of the rocket-powered

booster from the upper stage, it performs a return maneuver first, and then returns to the launch site through an unpowered reentry gliding flight before landing.

The turnover cycle of the RBS is 8 hours, of which 2 hours are used for launch preparation, and 6 hours are used for inspection and maintenance of the airport runway after landing. The RBS engine can be reused 10 times, and the booster can be reused 100 times. If there are eight flights each year, then its launch cost will be reduced by about 50% [13].

In 2013, the U.S. Defense Advanced Research Projects Agency (DARPA) relayed the U.S. Air Force to continue the research and testing of reusable booster system. Thus, they launched a new Experimental Space Aircraft (XS-1) Program, headed by the leading industrial sector in pre-research, for a continuous improvement of key technologies. Further, they laid a technical foundation for the future development of practical vehicles [14]. However, owing to economic factors, technical difficulties, and low carrying efficiency, it was finally canceled in early 2020.

The Baikal Program of Russia began by the end of 1998. The Baikal is a winged flyback booster that can fly back to the launch site and land like an airplane, automatically. It can be reused 100 times. Certain structural and propulsion components of the Baikal's common core stage booster have adopted mature technologies, including deployable wings, an all-moving tail, and an auxiliary turbojet engine.

(2) Reusable orbiter stage

Reusable orbiter stage refers to the reusable launch vehicle that is transported by the first stage vehicle to the sub-orbit, and then separates and powers its own main engine to transport the payload to the required orbit. Further, it can stay in orbit for a long period and perform various orbital service mission, on-demand return, and horizontal landing. Typical vehicles include the Space Shuttle Orbiter of the United States and the Buran Space Shuttle Orbiter of the Soviet Union.

- Space shuttle orbiter

The Space Shuttle Orbiter of the United States first flew in 1981, and was later decommissioned in 2011. According to the design requirements, each orbiter can be reused 100 times, and each time, a maximum of 29.5 t payload can be transported to the low-earth orbit, besides returning a payload of 14.5 t to the ground. The orbiter can carry 3 to 7 people and stay in the orbit for 7 to 30 days to perform tasks such as rendezvous, docking, parking, crew/cargo transportation, space testing, satellite launch, overhaul, and recovery. Hitherto, the six space shuttle orbiters of NASA have carried out a total of 135 missions.

- Buran space shuttle orbiter

By 1978, the Soviet Union began to develop the Buran space shuttle orbiter, which was successfully launched in November 1988.

The Buran space shuttle orbiter is 36 m long and 16 m high, with a triangular wingspan of 24 m, a fuselage diameter of 5.6 m, and a take-off mass of 105 t (14.3 times that of the mass of Soyuz manned spaceship), and a landing weight of 82 t. It was launched by using the newly developed Energia heavy-lift launch vehicle at that time.

Fig. 1.9 Launching satellites by aerospace craft

The Soviet Union produced a total of two Buran test aircrafts. On November 15, 1988, the Energia sent the unmanned Buran space shuttle orbiter into a predetermined orbit of 250 km. The Buran automatically orbited the earth twice. Subsequent to 3 hours in orbit, it returned to the ground on the same day, as scheduled, and landed at an airport 12 km away from the launch site. The first flight of the Buran was successful. The flight test of another test aircraft was canceled owing to economic influence.

On July 16, 2021, the vehicle of China for sub-orbit reusable flight test and verification took off at the Jiuquan Satellite Launch Center. Following the completion of the flight according to the set procedure, it horizontally landed at the Alxa Right Banner Airport in a smooth manner. The first flight was a complete success. The suborbital reusable vehicle can be employed as a stage of the lifting-body rocket-powered reusable space transportation system.

1.3.2.2 Combined-Cycle Powered Reusable Launch Vehicle

The combined-cycle powered reusable launch vehicle draws the main power from the combined-cycle powered engine, and adopts the method of horizontal take-off and horizontal landing (HTHL). According to the operating range of the engine, the vehicle can be employed as the first or the second stage.

The combined-cycle powered reusable launch vehicle has the potential to achieve single-stage-to-orbit. It can transport payloads from the ground to the required low-earth orbit, return on demand, and land horizontally. It is a method of implementation for a completely reusable space transportation system.

The SABRE aerospace project is a two-stage-to-orbit combined-cycle powered reusable launch vehicle project that uses the SABRE engine to launch satellites.

Currently, the project has received investment from Boeing, Rolls-Royce, and BAE Systems, and completed the assembly of the first pre-cooler prototype (HTX)and related auxiliary equipment, at the newly built TF2 test station in Colorado, USA (Fig. 1.9).

Skylon is a new generation of single-stage-to-orbit aerospace vehicle being developed by REL in collaboration with other European research institutions [16].

The overall design of the first Skylon (Skylon C1) has been completed. The total take-off mass is about 275 t. It has a slender fuselage design, with the fuselage length nearly 85 m and a wingspan 25 m, with a built-in propellant storage and load capsules. Two SABRE engines have been installed symmetrically at the tip, which has the ability to transport a payload of 12 m in length, 4.6 m in diameter, and 12 t in weight into a 300 km orbit.

1.4 Development Status of Launch Vehicle Intelligent Autonomous Technology

With the development of space technology, high-density launch of vehicles has become normal, and reusability has become a development trend, besides ever changing mission requirements. Higher demands are proposed for the performance such as reliability, convenience, maintainability, mission upgradability, on-orbit deployment, and mission planning, which necessitates the development of intelligent autonomous technology for the space transportation systems.

(1) Demand for high-density launch and quick response
In future, launch vehicle will encounter continuous high-density launch missions, so as to promote the construction of space infrastructure, and to meet the booming demand of commercial spaceflight. Internet constellation programs continue to advance. It mandates the shortening of the launch preparation time and improving the launch reliability, which raises stringent requirements for the launch vehicle testing, and launch capabilities.

(2) Demand for reusable and airline-flight-mode transportation
Owing to the development of aerospace technology and the demand for large-scale development of earth-moon space, the reusable space transportation system will eventually shift towards airline-flight-mode. This requires strong autonomous inspection, maintenance, and flight control capabilities, and the intelligent autonomous technology can provide strong support.

(3) Demand for improvement of the adaptation capability to multiple missions and flight faults

Owing to the increasing demand for access to space, besides space utilization missions in the future, the space transportation systems shall have intelligent and autonomous flight control and on-orbit deployment capabilities. Further, they should meet the requirements of the flight failure response capabilities, and the high-precision and diversified transfer of spacecraft. Currently, the failures owing to various reasons are generally coped with or adapted to, by means of redundancy and backup. The means are relatively simple and the system cost is relatively high, besides the types of failures to be dealt with being relatively limited. With the expansion of

the system scale, the mission complexity of the launch vehicle continues to increase, and higher requirements are placed on the flight failure handling capabilities of the launch vehicle.

Owing to the above-mentioned demands, it has acquired the ability to autonomously perceive its own state and the external environment, besides the continuous improvement in autonomous response and learning. This is achieved through the application of intelligent autonomous technology to the transportation system, during the launch and flight. It can independently complete the launch preparations and deal with uncertainties. During the flight, based on the results of intelligent perception of itself and the external environment, it has the ability to adapt and respond to the environmental changes and failures, independently, to achieve efficient and reliable access to space.

Currently, certain space faring nations in the world have taken the lead in conducting research and related practices for the autonomous flight control of the propulsion systems and actuator failures, besides intelligent autonomous technology applications such as the autonomous operation control technology.

1.4.1 Propulsion System Fault Identification and Mission Reconstruction

In the domestic and foreign space history, failures of the launch vehicle are more frequent owing to the failure of propulsion system. According to statistics, by 1970s, the United States had launched thousands of medium and long-range missiles, and launch vehicles, of which about 50% of the flight failures were due to propulsion system failures. Between 1990 and 2015, 64 foreign launch vehicles failed due to propulsion system failures, accounting for 51% of all launch failures.

Through intelligent autonomous technology, it can enhance the ability of the launch vehicle to actively adapt to propulsion failures and autonomous decision-making, and either to continue or degrade for completing tasks.

The space faring nations, viz., United States and Russia, have incorporated typical failure modes into the design and verification process, and made their main launch vehicles have a certain degree of failure adaptability by planning the mission capacity margins or adopting propulsion redundancy in the configuration selection.

(1) United States

United States was the first country to carry out research on the fault identification and diagnosis technology. It carried out research and application of the launch vehicle fault adaptability in the 1960s. The well-known engine manufacturer Rocketdyne has made statistics on the failures of seven types of engines (MA-3, MA-5, RS-27, F1, H1, J-2, SSME) during the development process. A total of 85,000 failures have been counted out of the 2500 engines that have conducted 1,000 flights, and the failure records were evaluated, screened, classified, and summarized into sixteen failure modes for the engine failure analysis and prediction. The Saturn series have

adopted “path adaptive guidance” that includes flight path optimization. During the launch of Apollo 6 in 1968, two engines of Saturn V were shut down, owing to the failure in the second stage. However, the control system stabilized the launch vehicle and prolonged the engine working time to propel the launch vehicle into the orbit, normally. In 1985, the safety system on the Challenger space shuttle shut down the malfunctioning No. 1 engine in due time, which did not have a fatal impact on the launch. Falcon 9 ground take-off allows one engine failure, and allows two engine failures after flying for a period of time. In October 2012 and March 2020, the Falcon 9 had two engine failures during the flight, and both of instances have been overcome, and the payload was successfully transported to the orbit through reconfiguration.

(2) Russia
Russia also has advanced technical experience in the launch vehicle diagnosis and health management technologies. Typical systems include the health monitoring and life assessment system that is developed for the high-power liquid launch vehicle engine (RD-170), and the orbiter real-time automatic monitoring and prediction system developed for the Buran Space Shuttle. Furthermore, the N1 launch vehicle has the capability to complete subsequent tasks with the remaining engines even after the failure of two engines during the flight.

1.4.2 Fault Identification and Control Reconfiguration of Actuator

The China’s LM-3B has applied the fault diagnosis and reconstruction algorithm of the attitude control thrusters. For the failure of attitude control thrusters, the information of the control system inertial measurement device is utilized. Furthermore, based on the priori motion information generated by the thruster control, the corresponding relationship of the command and expected angle of acceleration are employed to make a logical analysis and judgment, and hence, to complete the nozzle failure identification. The control reconfiguration is executed according to the identification result, and the control strategy is adjusted in real time. The effectiveness of the technology had been verified during the launch mission of the LM-3B in July 2020, and the onboard test verification was successfully completed. Further, the application flight was successfully realized in the subsequent missions. On December 22, 2020, China’s new-generation LM-8had successfully achieved its first flight. This flight has added the ability of online flight failure recognition during coast phase, and can autonomously perform attitude control reconfiguration under specific failure conditions (Fig. 1.10).

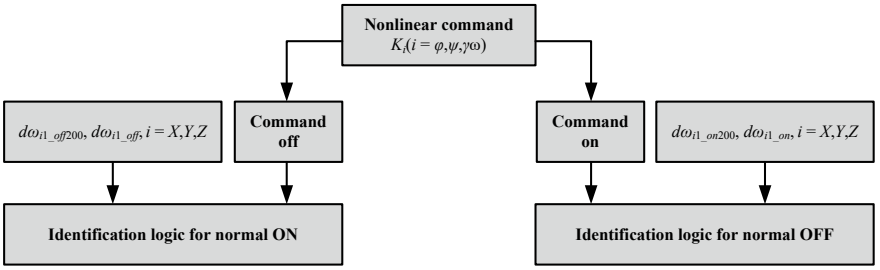


Fig. 1.10 The principles for fault diagnosis for attitude thrusters

1.4.3 Autonomous Control Technology

For the pre-launch ground test and load operation of the launch vehicles, the autonomous operation control technology can actualize the failure prediction and diagnosis, anomaly detection, fault detection, and isolation of the system and components before the launch. Different from the conventional ground process of launch vehicles, the technology can be applied to launch vehicle engine testing, ground loading, and future in-situ resource utilization of the moon or Mars surface. This can reduce the operation and maintenance costs, improve the effectiveness of the system, expand the surface operation and maintenance tasks for lunar and Mars missions, and minimize the tasks with manned operations (such as operations at dangerous location and teleoperations).

The autonomous operation of the launch vehicle can be achieved by integrating the technologies with respect to the health management, command and control, computing systems, decision-making, control software, intelligent components, and devices. Among them, the integrated system health management technology integrates the data, information, and knowledge distributed in each unit of the system for anomaly detection, fault diagnosis, and trend prediction. Further, it provides the users with the integrated perception of the status of important units in the system for the decision-making of the user [17].

The first stage of Falcon 9R has the ability to repeatedly execute multiple missions. Its health detection and diagnosis system monitor the health of the reusable launch vehicle body and engine during the full life cycle. In-depth research has been conducted on the detection content, intelligent detection methods, detection procedures and maintenance methods. Further, the reuse intelligent evaluation technologies, relevant in the post recovery of the launch vehicle body, have been developed, so that the reusable launch vehicle stage can complete the detection and maintenance, and restore the ability to launch in a relatively short time.

To apply autonomous operation technology to ground propellant loading, NASA implemented the research program on Integrated Ground Operations Demonstration Units (IGODU), which is intended for the advancement of processing and command control technologies related to the cryogenic operations, and reduction of the

complexity of the operating and launch facilities, thereby reducing launch costs [18]. Further, NASA has built a Simulated Propellant Loading System (SPLS). Taking into account the discrepancies between the program, and the final autonomous loading and operation, NASA launched the Autonomous Propellant Loading (APL) program [19] to verify the autonomous parallel loading, health monitoring, failure detection, and autonomous emergency handling technologies of multiple storage tanks and multiple media. Compared with the demonstration and verification of SPLS, the APL program expects to increase the maturity of the autonomous loading from level IV to V. Further, it employs the real cryogenic propellant and software in ground support equipment (GSE) or its integration to improve the model and expand capabilities to support more complex operations.

Based on the improvement in the maturity of the autonomous loading technology, the Atlas V has been left unattended from the start of –7.5 h core stage propellant loading to ignition and launch. The Delta IV has also been unattended from the –5.5 h propellant loading before the launch. The Falcon 9 has realized the unattended front end of the launch from the commencement of the loading.

In terms of the launch control applications, generally the launch cycle of the expendable launch vehicle is more than two weeks, and the launch service period is exceedingly long. Meanwhile, there are more personnel involved in the launch, and the demand for human resources is greater. In contrast, the development team of the Japan's Epsilon has applied autonomous operation and control technology for the launch control process of the launch vehicle. This step has simplified the launch system and pre-launch process of the launch vehicles, shortened the launch vehicle's launch preparation time to only 6 days, and reduced the human resource requirements. The launch vehicle adopts novel communication architecture and is connected to the ground support facilities through a high-speed network, making the launch operation safer and simpler. It became the first launch vehicle of Japan that can be controlled from outside the restricted area of the launch site [20].

1.5 Future Development Technical Challenge of Reusable and Intelligent Autonomous Technologies

1.5.1 Technical Challenge of Reusable Technology

The reusable technology has become an important technical development for launch vehicle. The arrival of the era of large-scale and low-cost access to space has also brought more technical challenges against the development of reuse technologies.

For the reusable transportation system in the axisymmetric configuration, the technical challenges of the parachute recovery and VTVL methods are as follows.

(1) The parachute recovery method has a low carrying capacity loss and relatively high technological maturity. However, this method mandates the design of a large group parachute system. The deployment area of the parachute system can reach

hundreds to thousands of square meters. The flexibility of the parachute system has a great impact to the recovery process. Restricted by the initial conditions of recovery, it may also be necessary to open the parachute at supersonic speeds. If the land-based parachute recovery is adopted, the large-scale cushioning airbags need to be designed. If helicopters are employed for the parachute recovery in air, it is necessary to ensure the stability of the capture. Therefore, the main technical challenges include the following technology of large-scale group parachute, large-scale cushioning airbag, supersonic parachute opening, fairing parachute recovery, and aerial capture of the recycled body.

(2) With the development of the technology of the throttling engine and advanced control theory, the VTVL methods have been investigated in the past decade. The Falcon 9 of the SpaceX has achieved dozens of VTVL missions. Europe, Russia, China, and other countries have put forward verification plan for the reuse of the VTVL technology. For the VTVL methods, the flight profile of the vehicle during the return process is complicated. Further, the system interference caused by the unknown shear wind and structural deviation, unmodeled dynamics, such as elasticity/aerodynamic coupling in the vehicle body, and liquid sloshing due to large-scale attitude adjustment, bring great challenges to the high-precision landing of the recovery stage. Furthermore, the landing support mechanism is also the key to ensuring the vertical and stable recovery, and there are also technical challenges in the mechanism design and reliable deployment. Therefore, the primary technical challenges of the VTVL method include the trajectory design and optimization technology, return high-precision guidance and control technology, and landing support technology.

The technical challenges of the rocket powered and the combined-cycle powered reusable transportation system in the lifting-body configuration are given as follows.

(1) Although the United States, Russia, Europe, and China have carried out different degrees of flight demonstration verification for the rocket-powered lifting-body reusable transportation system, even the United States has put the space shuttle into service as the main space transport vehicle from 1981 to 2011. However, in future, tremendous challenges are in waiting owing to the quest for affordable, technologically advanced, and feasible transportation system solutions. The solutions are inclusive of the overall design and optimization technology, aerodynamic layout, aerodynamic characteristics design technology, heat-resistant materials and structure technology, reuse maintenance and operation technology, high-precision integrated guidance, navigation, and control technology for on-orbit flight and reentry.

(2) For the combined-cycle powered reusable transportation system, the technical foundation is relatively weak and the design is complex. The primary focus is on the design and development of the combined-cycle powered engine. Moreover, under the combined-cycle powered mode, different factors such as the engine performance, structural load, and aerodynamic heat have multi-dimensional constraints on the design parameters such as the flight dynamic pressure, overload, and attack angle change. These parameters that will result in the overall scheme design faces the problem of multi-factor coupling and narrow feasible region. Furthermore, the combined-cycle powered flight area is large and the environment changes drastically. To exploit

the advantages of the combined-cycle powered propulsion performance, timely and stable switching of propulsion forms is particularly important, and the scheme design is complicated. Therefore, the primary challenges of the combined-cycle powered propulsion mode include the integrated design and of the flight body and propulsion in wide area, and the integrated control of the flight body and propulsion.

1.5.2 Technical Challenge of Intelligent Autonomous Technology

The emergence of artificial intelligence technology has created novel opportunities for the development of space technology. Compared with the unmanned driving of other modes of transport, the flight of the launch vehicle is unmanned from the beginning, but it is an automatic flight in a relatively definite design scenario, and thus its autonomy and intelligence are insufficient. Owing to the increasing frequency of space launches, incidents of various abnormalities have also increased significantly. If intelligent autonomous technology can achieve results in response to emergencies, it will promote the development of commercial spaceflight and a new economic ecology.

Many NASA scientists and engineers prefer to talk about machine learning and autonomy rather than artificial intelligence [21]. Currently, there are a few reliable applications of AI in the real-time flight of vehicles. However, it is more prominent in data mining, image recognition, failure diagnosis, and other fields, and the required real-time computing capability far exceeds that of embedded computing platforms in aircrafts. This phenomenon has also improved. According to reports, the enhanced Proton M has used the AI technology. The in-flight fail-safe performance level has been improved by introducing artificial intelligence elements. The launch vehicle control system automatically identifies and offsets potential failures related to propelling units, such as the steering actuators and engines. The mission objective is thus being achieved [22]. It can be seen that the AI technology is primarily expected to handle non-nominal working conditions.

AI technology can be utilized during the entire life cycle of launch vehicle including research, development, operation, and support. Further, the AI technology has several other areas of application, which can comprehensively build an intelligent transportation ecosystem including research and verification, production and manufacturing, testing and launch, flight, and evaluation. For example, explore the digital development process of the launch vehicles to improve the design efficiency; carry out more realistic virtual demonstration and verification tests of combination of virtuality and reality to break the bottleneck of the difference between the space and the earth, and the difficulty of physical testing; realize the autonomy of ground operations, reduce manpower requirements and improve rapid response capabilities to meet the challenges of normalization of multiple launches; realize the information sharing of the rocket-ground equipment through high-speed networks, and utilize

data mining, intelligent diagnosis, and other technologies to lay foundation for failure detection, isolation, system reconfiguration and improvement of the reliability of flight; develop intelligent manufacturing technology to improve the production efficiency and product qualification rate and support the development of space economy.

However, realizing more intelligent flights through AI technology has always been the biggest challenge for the intelligent autonomous technology of launch vehicles. From the perspective of control, the different characteristics of the launch vehicles, civil aviation aircrafts, and automobiles, in terms of route planning, guidance and control [23]. On the computing platforms with limited computing capability and power consumption and with less data support, and under very strict real-time requirements, it takes more arduous efforts to realize the intelligent or autonomous flight.

1.6 Conclusions

The mankind pace of space exploration has never stopped, and recently the demand has increased substantially. The era of space economy is coming. As the only tool of access to space, the launch vehicle has ushered in new development opportunities and challenges under the dual action of the demand traction and technology promotion. Currently, space missions show diversified development demands of large-scale, low-cost, high-reliability, and airline-flight-mode, which impose additional stringent requirements on the balance among performance, cost and reliability, mission adaptability, and rapid response capabilities of the launch vehicle.

Reusable and intelligent autonomous technologies have branded a distinctive time imprint for the development of the current vehicles. Falcon9 of SpaceX has been reused by VTVL, and has been put into commercial operation. Various new launch vehicles featuring vertical take-off, horizontal, or parachute landing are also under steady development. The reusable launch vehicles will provide technical and economic feasibility for large-scale access to space, making future space activities more diversified and frequent. This requires the vehicle to not only meet one-time or specific target tasks, but also adapt to changing demands and new environments that have never been explored. This is precisely the field where the application of intelligent autonomous technology is expected.

Realization of the intelligent or autonomous flight of the vehicle mandates the collaborative optimization of multiple disciplines. Particularly, the control system will play an important role as the nerve center of the vehicle. Owing to the development of information technology, the performance of the control system has been significantly improved. Therefore, the expectations from the control system are retrained to complete various predetermined tasks according to the content planned in advance, besides the ability to autonomously deal with various emergencies encountered in flight, maximum reduction of the dependence on the ground measurement and control personnel, reduced pre-launch preparations, and improved mission response and

survivability. Transition from the automatic control to autonomous control will be an important step towards smart launch vehicles.

References

1. W.M. Bao, X.W. Wang, Develop high reliable and low-cost technology of access to space embrace new space economy era. Aerosp. China **20**(4), 23–30 (2019)
2. X.J. Wang, Future development of china aerospace transportation system. Missil. Space Veh. **1**, 1–6 (2021)
3. Y. Lu, R. Giuseppe, X.W. Wang, F. Zhang, *Strategy of Large-Scale and Low-Cost Access to Space in the Future* (China Astronautic Publishing House, Beijing, 2021)
4. Joan A. Singer, Jerry R. Cook, etc. NASA space launch system operations strategy, in *63rd International Astronautical Congress* (Naples, Italy, 2012)
5. M. Chaize, G. Collange, etc. Ariane 6 launcher system development status, in *69th International Astronautical Congress* (Bremen, Germany, 2018)
6. Y. Morita, T. Imoto, etc. Development status of Japan's epsilon solid rocket launcher and its evolution, in *62nd International Astronautical Congress* (Cape Town, SA, 2011)
7. Z.H. Zhao, Space shuttle booster lightweight recovery system. Spacecraft Recov. Remote Sens. **18**(4), 3–8 (1997)
8. M. Robert, T. Anthony, A status report on the development of the kistler aerospace k-1 reusable launch vehicle. AIAA 2001-2069
9. L.A. Ledsinger, J.R. Olds, Optimized solutions for kistler k-1 branching trajectories using multidisciplinary design optimization techniques. J. Spacecraft Rockets **39**(3), 420–429 (2001)
10. Z.H. Gao, P.Z. Zhang, Y. Lu, et al., Analysis of vertical landing technique in reusable launch vehicle. J. Astronaut. **37**(1) (2016)
11. Z.Y. Song, Y.T. Wu, S.S. Xu et al., LM-8: the pioneer of long march rocket series on the innovations of commercialization and intelligence. J. Deep Space Explorat. **8**(1), 3–16 (2021). ((Chinese))
12. Z.Y. Song, C. Wang, Powered soft landing guidance method for launchers with non-cluster configured engines. Acta Astronaut. **189**, 379–390 (2021)
13. A.F. Dissel, B.C. Glass, S.B. Voorhees, et al., Benefits of integrated system health management for increased operational responsiveness of reusable booster system launch vehicles, in *AIAA Space 2010 Conference and Exposition* (California, 2010). AIAA 2010-8671
14. DARPA. Broad agency announcement experimental spaceplane (XS-1) tactical technology office (TTO), DARPA-BAA-14-01, 2013
15. H. Lane, "Wings In Orbit", scientific and engineering legacies of the space shuttle (1971–2010), NASA/SP-2010-3409, pp. 86–87
16. M. Hampsell, Progress on Skylon and Sabre, in *Beijing: 64th International Astronautical Congress*. IAC-13,D2.4,6x19609 (2013)
17. F. Figueroa, K. Melcher, Integrated systems health management for intelligent systems, in *AIAA, Advances in Intelligent and Autonomous Aerospace systems, Chapter 6*, pp. 173–200
18. M. Acuna, Cryogenics testbed laboratory flange baseline configuration. NASA, KSC-2013-090
19. M. Walker, W.E. Walker, F. Figueroa, Enabling autonomous propellant loading: providing situational awareness through model based reasoning, in *62nd Machinery Failure Prevention Technology (MFPT) and International Instrumentation Symposium, Dayton* (OH, United States, 2016), pp. 24–26
20. Y. Morita, T. Imoto, S. Tokudome, et al., The first flight of Japan's Epsilon launch vehicle, in *64th International Astronautical Congress*. IAC-13-D2.1.2 (Beijing, China, 2013)
21. How artificial intelligent is changing space system, SpaceNews, Aug. 15, 2017
22. Enhanced Proton M launch system - New features, SpaceNews, Oct. 4, 2021

23. Z.Y. Song, Q.H. Gong, C. Wang et al., Review and progress of the autonomous guidance method for Long March launch vehicle ascent flight. Sci. Sin. Inform. **51**, 1587–1608 (2021). ((Chinese))

Chapter 2
Autonomous Guidance Control for Ascent Flight

Zhengyu Song, Cong Wang, and Yong He

2.1 Introduction

The purpose of the guidance control is to release a payload into a prescribed target orbit (PTO) accurately. The parameters that determine an orbit are called orbital elements (OEs), which include the semi-major axis a, the eccentricity e, the argument of perigee angle ω, the inclination angle i, and the longitude of ascending node (LAN) or the right ascension of ascending node (RAAN) Ω, where a and e can be converted to the perigee height h_p and the apogee height h_a. Thus, the guidance mission of a launcher is a typical optimal control problem with multi-terminal constraints, which requires complex iterative calculations. Considering various constraints in practical applications, such as the accuracy of inertial navigation systems and the performances of embedded computing devices (speed and storage capacity), guidance methods need to balance the mission requirements, hardware resources, and algorithm complexity. A variety of guidance methods has been developed with distinct era characteristics.

Z. Song (✉)
Research fellow, China Academy of Launch Vehicle Technology, China Aerospace Science and Technology Corporation, 100076 Beijing, People's Republic of China
e-mail: zycalt12@sina.com

C. Wang · Y. He
Senior Engineer, Beijing Aerospace Automatic Control Institute, 100854 Beijing, People's Republic of China
e-mail: wangcong0404@126.com

Y. He
e-mail: heyong2004@163.com

Z. Song et al. (eds.), *Autonomous Trajectory Planning and Guidance Control for Launch Vehicles*, Springer Series in Astrophysics and Cosmology,
https://doi.org/10.1007/978-981-99-0613-0_2

2.1.1 Traditional Guidance Methods

(1) Guidance methods in early stage

The early guidance methods of the ascent phase for launchers in various countries were open-loop guidance (OLG) methods [1–3]. In these solutions, an off-line trajectory to the PTO was planned in advance, including the time-varying position, velocity, and thrust directions (guidance commands). After liftoff, the guidance commands in the corresponding flight phases were interpolated based on the trajectory by taking the time, velocity, or altitude as the independent variable. In general, the command interpolated over time consumes more fuel than that over the velocity. OLGs usually transform the OEs into terminal velocity and position constraints at the prescribed injection point, and they perform well to meet the load limit requirements when flying in the atmosphere using a wind biasing trajectory based on the wind field of the launch day [4].

The perturbation guidance method (PGM) was developed to further improve the injection accuracy, and its guide coefficients were designed offline based on the flight profiles and the most concerned OEs. The state variables of the velocity and position were fed back into the guidance loop, then their deviations to the nominal values were calculated, and the guidance commands were initiated therefrom to guide the launcher to fly as close to the nominal trajectory as possible [5–8]. PGMs have been applied to launchers since the 1950s, when the guide control could only be conducted first on a simple computing device (which cannot be regarded as a computer). The algorithm was very simple during the early stage, and all the complex operations, such as the calculation of the gravity of Earth, could not be completed onboard. For example, the PGM of the Long March launch vehicles (LMLVs) proposed by Cui et al. generated the guide commands onboard based on interpolation tables [7]. Combined with the perturbation cutoff equations, a better separation control accuracy was obtained [8–10]. Considering that the precision of the inertial devices was also low at that time, the PGM had been applied for a long time.

With the development of avionics, the computational bottlenecks on the guidance methods have been greatly eliminated, and PGMs have also further developed. For instance, the simplified apparent velocity accumulation has evolved into explicit navigation calculations. The influence of the second-order term of gravity has also been involved in the algorithms, and more OEs, not just those most concerned ones, can be satisfied by segmented or weighted guide controls.

Compared with OLGs, PGMs calculate guide commands online, exhibiting a certain degree of autonomy. However, if the control deviations increase, the hypotheses of the first-order linearization in the design of guide coefficients cannot hold, which greatly degrades the performance of the PGMs.

(2) Explicit guidance methods

Explicit guidance methods calculate guidance commands in real-time based on the explicit expressions of control functions, which are generally applied when the rocket enters into the vacuum environment. Owing to the release of the wind loads on the vehicle's structure in a vacuum, an optimal guidance command can be derived by

closed-loop guidance (CLG) algorithms. The OEs of the PTO can be directly taken as the terminal constraints, which greatly improves the adaptability and injection accuracy [11]. Typical analytical CLG methods include the iterative guidance mode (IGM) [11–16] for rockets, the powered explicit guidance (PEG) [17–19] for the space shuttles, and the optimal guidance (OPGUID) [20, 21] based on variational methods.

The IGM could iteratively calculate the required velocity and position increments to the PTO, and then plan the optimal flight path [13, 14]. The earlier the IGM is called, the more complex the algorithm is, because the flight profile covers more flight phases, but the stronger the fault adaptability becomes. The development in IGM boosted the progress of the rendezvous and docking (RVD) missions in China's manned spaceflight project, where the IGM across two continuous powered phases was first used in the LM-2F/Y8 mission in November 2011. After this, a prediction-correction IGM was adopted for LM-7, which achieved high injection accuracies under high thrust conditions without terminal velocity correction systems. In September 2020, the IGM with a terminal attitude constraint was first utilized in the LM-2F/T3 mission, and the OEs and terminal attitudes were well controlled simultaneously without a reaction control system (RCS) [15]. The IGM across the coasting phase was first used in the maiden flight of LM-8 in December 2020.

The abort or termination requirements during ascent were required for the space shuttles, allowing them to return safely or enter into a pre-set parking orbit if one main engine failed. Thus, a semi-analytical prediction-correction algorithm, i.e., PEG, was proposed. It was a kind of linear tangent guidance, assuming that the thrust direction vector satisfies the linear tangent control laws from the point of view of fuel minimization, and then the guidance law was derived based on the variational method. The covariates were solved based on the required velocity increments, corrected by estimating the velocity deviations at the shutdown time [19]. The number of OEs that must be satisfied can be selected to meet the different mission needs, and the scenario of returning after failures is also considered by the PEG.

OPGUID had been regarded as a backup for the IGM and PEG from the era of the Saturn rockets to the space shuttles. It could meet all the necessary optimality conditions, including the Euler-Lagrange equations, but its maturity is still considered to need improvement.

IGM and PEG exhibit basically the same performances in terms of adaptability, robustness, target performance, and flexibility when confronted with engine faults, while the advantages of the OPGUID lie in its fewer assumptions [21]. In addition, an early release IGM (ERIGM) was studied to be applied before entering the vacuum [1]. The restrictions of the rocket structure caused by the aerodynamic loads when passing through high-dynamic-pressure or high-wind regimes are taken as the constraint of the output commands. Although the assumption of ERIGM's analytical expression in a vacuum is not satisfied in the atmosphere, the simulation analysis shows that compared with the combination of OLG + IGM, the ERIGM can improve the launch performance while maintaining a smaller dispersion of the terminal states.

2.1.2 Autonomous Guidance Methods

Autonomous guidance methods (AGMs) in and out of the atmosphere have become a popular research topic since 2016, aiming to improve flight autonomy under various scenarios. AGM is a kind of method to dynamically generate the current guidance command based on real-time trajectory planning, while satisfying the process and terminal constraints of all the following flight phases. It does not rely on but also does not exclude the reference trajectory planned off-line, and it can deal with complex, time-varying, and nonlinear constraints onboard, exhibiting strong adaptability and robustness.

2.1.2.1 Related Studies

The analytical CLGs based on the optimal control theory under vacuum conditions, i.e., IGM, PEG, and OPGUID, can also be regarded as the first generation of AGMs. With the upgrading of the onboard computing power, the online trajectory planning methods, which mainly include indirect [22–24] and direct methods [25–27], were developed and backed by numerical calculations. If their planning period meets the real-time requirements, they may replace the existing analytical methods; if not, a combination of "on-line planning + tracking guidance" can be adopted. Currently, many new concepts related to guidance control have been proposed, reflecting some aspects of the features of AGMs:

(1) Computational guidance
In 2016, the Journal of Guidance, Control, and Dynamics published a special issue of Computational Guidance and Control [28] and pointed out that the control laws and controllers with fixed structures in traditional guidance and control will be replaced by algorithms, which are different from other branches of computational engineering and science. The commands of computational guidance would be model- or data-based, and there is no need for in-advance planning, gain adjustment, or a large amount of off-line design for the nominal state. Thus, computational guidance can be regarded as a special and possibly the main solution strategy for AGMs.

(2) Model-based real-time optimization
At the 2016 Aerospace Conference, it was suggested that model-based real-time optimization is the main direction of future research. It can deal with complex constraints [29, 30], and overcome the shortcomings of traditional real-time optimal control methods which only handle unconstrained or simply constrained problems Therefore, model-based optimization can be regarded as the main research area of AGMs.

(3) Autonomous mission planning
This concept was first found in NASA's project "Flight Autonomy", which can be regarded as a higher-level representation of AGMs. If the mission planning can be conducted in real-time, its effect is equivalent to that of AGMs. Four elements of

autonomous mission planning [31], such as automatic operators, intelligent initial guesses, powerful optimization software, and technologies supporting online real-time operations, are also the key factors of AGMs.

(4) Adaptive guidance
The adaptive guidance and mission planning were proposed in the roadmap of aerospace intelligent systems drafted by the AIAA Intelligent Systems Technical Committee [32]. These technologies would learn and optimize the system behavior, optimize the aerodynamics and performances of propulsion systems, and solve challenging problems such as the real-time aerodynamic control optimization, convergence of onboard optimization, computational efficiency, adaptive control with sensor constraints, and the security cost of over-optimization. Adaptive guidance reveals the main features and requirements of AGMs.

(5) Adaptive optimal guidance
It was reported by Russian scholars at the 2016 International Astronautical Congress [33] that the three lost global navigation satellite system (GLONASS) satellites launched on December 5, 2010 might have been saved if adaptive optimal guidance had been applied. This method adopts model-based numerical optimization, making full use of the upper stage's carrying capacity to compensate for the performance degradation of the former stages and re-planning the flight path to the PTO. This solution was similar to the "End to End" (E2E) space mission planning architecture [34], which refers to a multi-stage simultaneous optimization from launching to the final destination. From this point of view, adaptive optimal guidance or E2E planning emphasizes the global optimization feature of AGMs.

All the above-mentioned technologies aim to deploy the payloads to the PTO, even facing unexpected conditions. The potential assumption is that the PTO is reachable, which is reasonable under normal conditions or with enough margin of performance left when a failure occurs.

(6) Fault-tolerant guidance (FTG)
If the PTO is unreachable under failure conditions, the propellant will be exhausted during the flight to it, and the terminal velocity and position may not ensure a parking orbit, causing the rocket/satellite to crash into the ground. The FTG is proposed to reconstruct the mission under this situation [35] to avoid a complete loss, because the well studied fault-tolerant control cannot overcome the effect of gravity to enter into an orbit. Thus, FTG represents an important application scenario of AGMs.

The failures of thrust drop do occur in space launches, but if the PTO is still reachable, either IGM or PEG is capable of re-planning the flight path and releasing the payloads to it. For the Space Launch System (SLS) of NASA, the mission abort design was studied when the target was no longer reachable, but the strategy was based on the off-line simulation and loaded into the onboard computer in advance. It was reported that the Artemis I 1 flight software would pre-upload nine alternative targets. However, only autonomous rescue measures can fully use the remaining carrying capabilities of the launcher to save a mission or avoid crashing down to the ground.

2.1.2.2 Features of AGMs

AGMs are significantly different from OLGs and have a wider range than the traditional CLGs, and could satisfy the demands of research institutions and experts to improve flight autonomy.

AGMs are not synonymous with the trajectory optimization. Trajectory optimization is usually planned off-line, and the result is used as the reference for the tracking control of the launcher. However, the trajectory is generally designed according to the nominal state and cannot predict all disturbances and uncertainties inflight, then the actual flight path usually deviates from the prescribed trajectory. Moreover, trajectory optimization is time consuming, but this is tolerable and the real-time feature is not a key factor for off-line planning. Numerical computing is applied in the trajectory optimization which can consider as many constraints and variables as possible. Even if the planning is not convergent, it can be stopped by human interventions or reset by a new initial guess.

In recent years, trajectory optimization has gradually been adopted for onboard applications, such as the online trajectory optimization, where the constraints and variables it deals with are tailored to facilitate onboard processings, and the dynamic trajectory planning, representing the on-line, real-time, and iterative planning. The former plans the trajectory once or several times followed by the tracking control. The latter is mostly equivalent to AGMs if the frequency of the dynamic planning is almost the same as that of the guidance control. However, only a PTO and no mission reconstruction is considered by these trajectory optimization technologies.

AGMs have the following four distinct features:

(1) Online. This sets a high demand for the real-time performance of the algorithms and onboard computers. Although a prescribed flight path is no longer required, it can be used as the initial guess to accelerate the online computing.
(2) Dynamic. Planning is scheduled in each guidance cycle, sometimes called the iterative optimization. The shorter the period of planning is, the stronger the adaptability to uncertainties and interferences becomes. Only the current command of the newly planned results is used for the real control. This process is repeated in the next guidance cycle.
(3) Global. Each planning process obtains a whole flight path from the initial states, e.g., the current velocity, position, and mass, to the terminal constraints. This is quite different from rolling optimizations, where only a short time period of dynamics is involved.
(4) Reconstructing. When the PTO is impossible to be reached (often caused by propulsion system faults), it can reconstruct the flight profile or mission target to match the remaining carrying capacity, so as to save the mission or avoid irretrievable disasters as far as possible.

The challenges faced by rockets for the autonomous guidance are also different from those faced by automobiles and civil aircrafts.By taking into account the practices of automobile industries, mission planning and guidance methods are no longer strictly distinguished.

(1) For automobiles, the planning problem is rules or process based. They basically drive in planar motions under relatively certain environments and operations on road networks with determined risk levels. When an emergency issue occurs, there is not more than one minute to switch from a cruising state to a safe state, such as resting on the roadside and waiting for conditions to improve. They have many control modes, such as forward motion, backward motion, steering, acceleration, deceleration, pausing, and restarting. The difficulties lie in the time-varying dynamic interactions with other vehicles, pedestrians, traffic conditions, and road markings, as well as driving rules.
(2) Civil aircrafts usually fly in prescribed routes at fixed altitudes except for takeoff and landing. They also have high control abilities, such as the forward motion, upward motion, downward motion, steering, acceleration, and deceleration. The main challenges lie in handling uncertainties autonomously, including the local climate phenomena, variable weather, sudden surges, and out-of-service airports due to delays. They are sensitive to climate conditions, such as the airflow, gales, and thunderstorms. Emergency treatment is definitely needed considering the available fuel, alternative airports and their altitudes, runway lengths, and slope constraints, extended flights in harsh terrain (such as mountains or wilderness), and possible survival concerns.
(3) Launchers are usually insensitive to abrupt climate changes, because they fly across the atmosphere very quickly, and there are no dynamic constraints similar to that of civil vehicles. The planning problem is strongly nonlinear due to the gravitational and OE constraints. In fault conditions, there is no prescribed parking orbit similar to alternative airports or roadsides, and finding an optimal rescue orbit onboard is very challenging. The control modes are limited, and no descent or stop inflight is available. The amplitude of the thrust is usually fixed, and only the thrust direction can be adjusted. Unlike automobiles or aircrafts, there are no database supports or high-performance computing platforms onboard, even if a database is available.

The problems concerning various vehicles are different in terms of the guidance control or mission planning. Except for the abrupt changes of the climate and dynamic constraints, the AGMs of a launcher face more challenges. This is partly due to the stronger nonlinearity in the optimization problem mainly induced by the gravitational force, the terminal constraints, and a wider range of mass changes, while the other reasons mainly lie in the weaker computing power of the on-board computer (OBC) compared with those on automobiles or aircrafts.

AGMs can relax the pressure of attitude control. If the control deviations exist in each guidance cycle, the accumulated errors can be eliminated in line with a newly planned trajectory in the next cycle, and the influence of errors is retained within a very short planning period. Thus, AGMs improve the robustness of a launcher to disturbances or uncertainties.

It would become difficult to obtain analytical solutions with the increased number of variables that need to be determined, however, an analytical solution after simplifying the problem is a preferable initial guess for AGMs. Even so, AGMs can-

not solve too many constraints without affecting the computational efficiency, and hundreds of milliseconds of the solving time are the maximum tolerance. Further, divergence should be avoided in AGMs, but if it does occur, some countermeasures must be scheduled in advance.

2.1.3 Summary

The history of ascent guidance methods applied in LMLVs is introduced here to sum up the above discussion. The three stages are explained in Fig. 2.1, reflecting the changes of the mission requirements and the evolution of the guidance methods. The latter two stages in Fig. 2.1 represent two typical applications of AGMs:

(1) Closed-loop guidance methods for determined target orbit
When the target orbit is determined (i.e., the target matches the carrying capacity), traditional or enhanced analytical methods can play a very good role in the extra-atmospheric flight. For example, although the IGM shows weakness in large arc flight profiles owing to its simplified mean gravitational field assumption, this can be solved by the segmented processing if a coasting phase is inserted. An enhanced PEG algorithm is also being developed by the SLS to adapt to extended flight times.

When flying in the atmosphere, the landing restrictions of the launcher's jettisons, such as the separated boosters and fairing, should be seriously considered. Then, tracking control is still a relatively safe method under this condition to ensure that the debris falls within a predictable area. Sun and Lu proposed the homotopy method to deal with the atmospheric density onboard for the ascent guidance control, and it showed a certain adaptability to the main engine thrust loss [22]. However, the constraints of the landing area were not considered.

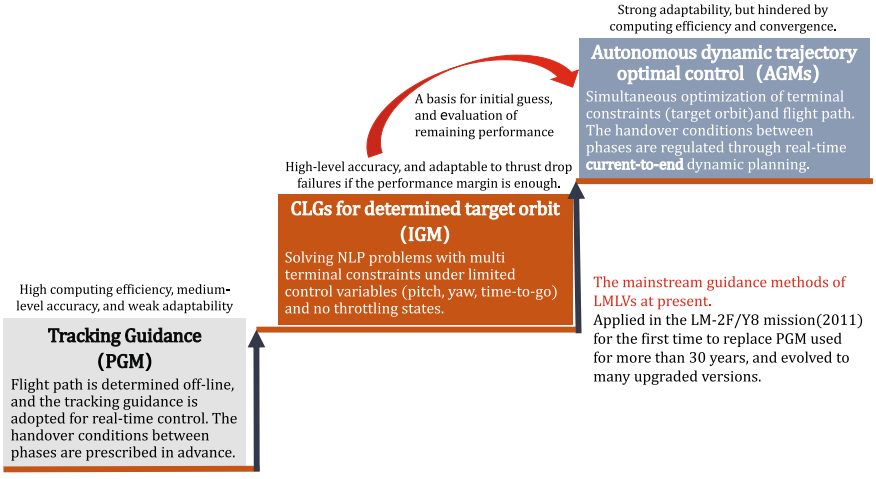


Fig. 2.1 Evolution of ascent guidance methods for Long March rockets

(2) Simultaneous optimization of target orbit and flight path
This only occurs when the PTO is no longer reachable and a new target orbit should be determined. Onboard joint optimization of the new orbit and flight path is very difficult, so a special failure mode, called "engine out," is considered in the SLS, and a similar approach was also used by the space shuttle. NASA expects a successful mission even if one engine cuts off prematurely, so a sufficient performance margin is definitely needed. When facing severe failures, the SLS can make decisions based on the pre-uploaded alternatives, as introduced in sections above. In recent years, the studies of the autonomous dynamic trajectory optimization under typical failure modes have been initiated for LMLVs and obtained positive results.

At present, few studies on ascent AGMs have been publicly published when considering the needs of mission re-constructions.

2.2 Motion Models of Launchers

2.2.1 Motion Models

The differential translational motion is usually described in the launch inertial coordinate system (LICS) and is shown as follows:

$$\begin{aligned} \dot{\boldsymbol{r}} &= \boldsymbol{V}, \\ \dot{\boldsymbol{V}} &= \frac{\boldsymbol{F}_T+\boldsymbol{F}_R+\boldsymbol{F}_A+\boldsymbol{F}_s+\boldsymbol{F}_e+\boldsymbol{F}_D+\boldsymbol{G}}{m}, \\ \dot{m} &= -\frac{(\|\boldsymbol{F}_T\|-S_e(P_e-P_a))}{I_{sp}g_0}, \end{aligned} \tag{2.1}$$

where $\boldsymbol{r}$ is the position vector, $\boldsymbol{V}$ is the velocity, $\boldsymbol{G}$ is the gravity. $\boldsymbol{F}$ is the other forces acting on the vehicle, and the subscripts T, R, A, s, e, and D represent the engine control, RCS, aerodynamic, sloshing, elastic, and interference torques, respectively. I_{sp} is the specific impulse of the engine, S_e is the cross-sectional area of the nozzle, P_e is the atmospheric pressure in the design state, P_a is the external atmospheric pressure inflight, m is the mass, and g_0 is the gravitational acceleration of the sea level. The origin of the LICS is the launch point, the x-axis points in the launch direction in the horizontal plane of the launch site, and the y-axis points to the sky along the connecting line between the earth center and the launch point. The z-axis satisfies the right-hand rule.

The aerodynamic force is related to the shape of the launcher and the dynamic pressure inflight, which can be expressed as

$$\boldsymbol{F}_A = qS_A\boldsymbol{C}_A. \tag{2.2}$$

where q is the dynamic pressure, S_A is the reference area, $\boldsymbol{C}_A$ is the aerodynamic coefficient related to the altitude, Mach number, angle of attack, and sideslip angle.

The differential rotational motion is usually described in the vehicle's body coordinate system (BCS). The origin of the BCS is located at the center of mass of the launcher, the x_1-axis points to the head along the body axis, the y_1-axis is perpendicular to the longitudinal symmetry plane of the launcher and points upward, and the z_1-axis satisfies the right-hand rule.

The following equations reflect the influence of torques on the angular velocity of the launcher:

$$\dot{\boldsymbol{\omega}}\boldsymbol{J} = \boldsymbol{M}_T + \boldsymbol{M}_R + \boldsymbol{M}_A + \boldsymbol{M}_s + \boldsymbol{M}_e + \boldsymbol{M}_D - \boldsymbol{\omega} \times \boldsymbol{J}\boldsymbol{\omega}. \tag{2.3}$$

where $\boldsymbol{\omega}$ is the angular velocity rotating around the axial direction, $\boldsymbol{J}$ is the moment of inertia, and $\boldsymbol{M}$ is the torque acting on the rocket.

$$\boldsymbol{M}_A = \left(qS_A l_A \boldsymbol{C}_d^{\omega}\boldsymbol{\omega}/\|\boldsymbol{V}\| + qS_A l_A \boldsymbol{C}_d\right), \tag{2.4}$$

where l_A is the reference length, $\boldsymbol{C}_d$ is the aerodynamic torque, and $\boldsymbol{C}_d^{\omega}$ is the damping coefficient.

$$\boldsymbol{M}_e = \sum_i (b_{1i}^{\varphi}\dot{q}_i + b_{2i}^{\varphi} q_i), \tag{2.5}$$

where q_i is the i-th order elastic generalized coordinate, b_{1i}^{φ} and b_{2i}^{φ} are the i-th order elastic additional moment coefficients.

The propellant sloshing moment can be divided into three parts: normal, transverse, and axial. For example, the propellant normal sloshing moment (M_{s3}) is

$$M_{s3} = \sum_p (b_{4p}^{\omega_3}\Delta\ddot{y}_p - b_{5p}^{\omega_3}\Delta y_p), \tag{2.6}$$

where Δy_p is the p-th order longitudinal sloshing displacement, $b_{4p}^{\omega_3}$ and $b_{5p}^{\omega_3}$ are the interaction coefficients of the sloshing moment and sloshing centroid in the pitch channel.

The modeling of the elastic vibration is established according to the finite element method [36]:

$$\begin{aligned}
&\ddot{q}_i + 2\zeta_i\varpi_i\dot{q}_i + \varpi_i^2 q_i = \\
&D_{1i}^{\omega_{z1}}\omega_{z1} + D_{2i}^{\omega_{z1}}\alpha_3 + D_{3i}^{\omega_{z1}}\delta_{\omega_{z1}} + D_{3i}^{''\omega_{z1}}\ddot{\delta}_{\omega_{z1}} \\
&+\sum_p (G_{ip}^{''\omega_{z1}}\Delta\ddot{y}_p + G_{ip}^{\omega_{z1}}\Delta y_p) + \sum_j (R_{ij}^{'\omega_{z1}}\dot{q}_j + R_{ij}^{\omega_{z1}} q_j) \\
&+D_{1i}^{\omega_{y1}}\omega_{y1} + D_{2i}^{\omega_{y1}}\alpha_2 + D_{3i}^{\omega_{y1}}\delta_{\omega_{y1}} + D_{3i}^{''\omega_{y1}}\ddot{\delta}_{\omega_{y1}} \\
&+\sum_p (G_{ip}^{''\omega_{y1}}\Delta\ddot{z}_p + G_{ip}^{\omega_{y1}}\Delta z_p) + \sum_j (R_{ij}^{'\omega_{y1}}\dot{q}_j + R_{ij}^{\omega_{y1}} q_j) \\
&+D_{1i}^{\omega_{x1}}\omega_{y1} + D_{2i}^{\omega_{y1}}\alpha_1 + D_{3i}^{\omega_{x1}}\delta_{\omega_{x1}} + D_{3i}^{''\omega_{x1}}\ddot{\delta}_{\omega_{x1}} \\
&+\bar{Q}_{xi} + \bar{Q}_{yi} + \bar{Q}_{zi},
\end{aligned} \tag{2.7}$$

where ξ_i is the i-order elastic damping, $\varpi_i$is the i-order elastic frequency. Δz_p is the p-th order transverse sloshing displacement. $\delta_{\omega_{x1}}$, $\delta_{\omega_{y1}}$, $\delta_{\omega_{z1}}$ are the engine swing angles, $G_{ip}^{''\omega_{z1}}$, $G_{ip}^{\omega_{z1}}$, $G_{ip}^{''\omega_{y1}}$, $G_{ip}^{\omega_{y1}}$ are the coupling coefficients of the p-th order sloshing to the i-th order elasticity, $R_{ij}^{'\omega_{z1}}$, $R_{ij}^{\omega_{z1}}$, $R_{ij}^{'\omega_{y1}}$, $R_{ij}^{\omega_{y1}}$ are the coupling coefficients of the j-th order to the i-th order elasticity, $\bar{Q}_{xi}$, $\bar{Q}_{yi}$, $\bar{Q}_{zi}$ are the elastic generalized disturbances.

The motion equation describing normal sloshing is

$$\begin{aligned}&\Delta\ddot{y}_p + 2\zeta_{hp}\Omega_p\Delta\dot{y}_p + \Omega_p^2\Delta y_p = \\ &-E_1\Delta\dot{\theta} + E_2\Delta\varphi + E_3\Delta\alpha - E_{pz}\Delta\ddot{\varphi} + \sum_i (E''_{ip}\ddot{q}_{iy} + E_{ip}q_{iy}),\end{aligned} \tag{2.8}$$

where E''_{ip}, E_{ip} are the elastic hinge coupling coefficients, ζ_{hp} is the i-th order sloshing damping, Ω_p is the p-th order sloshing frequency.

2.2.2 Constraints and Objectives

(1) Initial state constraints

The takeoff time t_0 is defined as the initial time. The initial position at t_0 is the location of the launch point, the initial velocity is that of the launch point generated by the earth's rotation, and the liftoff mass is the initial mass. The rocket flies vertically off the launch pad, and the initial state constraints can be expressed as

$$[\boldsymbol{r}, \boldsymbol{V}, m](t_0) = [\boldsymbol{r}_0, \boldsymbol{V}_0, m_0], \ \varphi = 90^\circ, \ \psi = 0^\circ, \tag{2.9}$$

where ϕ, ψ are the pitch and yaw angles, respectively.

(2) Process constraints

When flying in the atmosphere, the following constraints should be met to ensure structural safety:

$$|q\alpha| \le q\alpha_{\max}, \quad N \le N_{\max}, \quad q < q_{\max}, \tag{2.10}$$

where N is the overload, α is the angle of attack (AOA). The subscripts max and min represent the maximum and minimum values of the corresponding constraints, respectively.

Limited by the control ability of the actuators, the following constraints are prescribed:

$$|\delta| \le \delta_{\max}, \quad MH \le MH_{\max}, \tag{2.11}$$

where δ is the engine swing angle, and MH is the hinge moment. To ensure the attitude stability, the following constraints are imposed:

$$\begin{aligned} &\left|\omega_{\varphi}\right|, \left|\omega_{\psi}\right| \leq \omega_{\max},\ \varphi_{\min} \leq \varphi \leq \varphi_{\max},\ \psi_{\min} \leq \psi \leq \psi_{\max}, \\ &\boldsymbol{u}_T(t) = [0, 1, 0]^T,\ t \in [t_0, t_1], \end{aligned} \tag{2.12}$$

where ω_{ϕ}, ω_{ψ} are the corresponding angular velocities, $\boldsymbol{u}_T$ is the thrust direction.

The rocket should keep rising vertically for a short time (t_1) after takeoff, that is, $\boldsymbol{u}_T$ is perpendicular to the horizontal plane.

According to the engine configurations, the mass differential equation given by Eq. (2.1) is revised to the following equation, where the equivalent thrust and specific impulse of the k-stage engines are denoted by the superscript k:

$$\dot{m}^k = -\frac{\left(\left\|\boldsymbol{F}_T^k\right\| - S_e^k\left(P_e^k - P_e^k\right)\right)}{I_{sp}^k g_0}. \tag{2.13}$$

For the multi-stage launcher, the states of the velocity, position, and attitude between the stages are continuous. During stage separations, the mass constraints are included:

$$\left[\boldsymbol{r}_0,\ \boldsymbol{V}_0,\ \varphi_0,\ \psi_0\right]^k = \left[\boldsymbol{r}_f,\ \boldsymbol{V}_f,\ \varphi_f,\ \psi_f\right]^{k-1},\ m_0^k = m_f^{k-1} - m_s^{k-1}, \tag{2.14}$$

where m_s is the separation mass, and the subscript f represents the terminal state of each stage.

(3) Terminal constraints

When the payload is released from the launcher, it would enter into an orbit, which is determined by the terminal states of the payload, the gravitational force, and other perturbation forces. If only considering the gravitational effect, the OEs can be calculated based on V_x, V_y, V_z, and x, y, z in the LICS.

First, we have the following equation:

$$\begin{aligned} &x_r = x + R_{0x},\ y_r = y + R_{0y},\ z_r = z + R_{0z}, \\ &r = \sqrt{x_r^2 + y_r^2 + z_r^2},\ V = \sqrt{V_x^2 + V_y^2 + V_z^2}, \end{aligned} \tag{2.15}$$

where V_x, V_y, V_z are the velocity components in LICS, and x, y, z are the position components in LICS, R_{0x}, R_{0y}, R_{0z} are the geocentric vector components to the launch point.

Then, the OEs are

$$a = \frac{r}{2-\upsilon},\ \upsilon = \frac{rV^2}{\mu}, \tag{2.16}$$

$$e = \sqrt{1 - (2-\upsilon)\,\upsilon\cos^2\gamma},\ \gamma = \arcsin\frac{V_x x_r + V_y y_r + V_z z_r}{Vr}, \tag{2.17}$$

$$h_p = r_p - R_e,\ r_p = a(1-e),\ h_a = r_a - R_e,\ r_a = a(1-e), \tag{2.18}$$

where R_e is the radius of the Earth; r_p, r_a are the distances from the center of the Earth to perigee and apogee, respectively; μ is the gravitational constant.

$$T = 2\pi\sqrt{\frac{a^3}{\mu}}, \tag{2.19}$$

where T is orbital period.

The parameters characterizing the orbital direction are i, Ω, and ω:

$$i = \arccos\frac{h_z}{h},\ \begin{bmatrix} h_x \\ h_y \\ h_z \end{bmatrix} = \begin{bmatrix} yV_z - zV_y \\ zV_x - xV_z \\ xV_y - yV_x \end{bmatrix},\ h = \sqrt{h_x^2 + h_y^2 + h_z^2}, \tag{2.20}$$

where $[h_x, h_y, h_z]^T$ is the vector product of velocity and position.

$$\sin\Omega = \frac{h_x}{\sqrt{h_x^2+h_y^2}},\ \cos\Omega = -\frac{h_y}{\sqrt{h_x^2+h_y^2}}, \tag{2.21}$$

where $\Omega \in [0, 2\pi]$, and the quadrant is determined according to the symbols of $\sin\Omega$ and $\cos\Omega$. Furthermore,

$$w = u - f, \tag{2.22}$$

where f is the true anomaly, u is the angular argument to the ascending node and calculated by the following equation:

$$u = \begin{cases} \arccos\frac{x_r\cos\Omega + y_r\sin\Omega}{r}, & (z_r \geq 0) \\ 2\pi - \arccos\frac{x_r\cos\Omega + y_r\sin\Omega}{r}, & (z_r < 0) \end{cases}, \tag{2.23}$$

f is used to characterize the position of the payload in orbit and is expressed as follows:

$$f = \begin{cases} \arccos\frac{a(1-e^2)-r}{er}, & (\gamma \geq 0) \\ 2\pi - \arccos\frac{a(1-e^2)-r}{er}, & (\gamma < 0) \end{cases}. \tag{2.24}$$

The terminal velocity and position are transformed into constraints of the OEs, as well as a terminal mass constraint, shown as follows:

$$\begin{aligned} &\left\| [a_s,\ e_s,\ i_s,\ \Omega_s,\ w_s,\ f_s]^T - [a,\ e,\ i,\ \Omega,\ w,\ f]^T(t_s) \right\| \leq \Delta Orbit, \\ &m(t_s) \leq m_{allow}, \end{aligned} \tag{2.25}$$

where t_s is the terminal time and the subscript s represents the nominal state at the departure time. $\Delta Orbit$ is the maximum allowance of the six orbital elements, m_{allow} is the minimum allowable mass at the end of the ascent phase.

(4) Objectives

The objective of the ascent guidance method of a launcher can be expressed as the weighted sum of maximizing the residual mass at the payload departure time while minimizing the terminal state deviations:

$$\min J = -m\left(t_s\right) + \lambda_{orbit} \left\| [a_s, e_s, i_s, \Omega_s, w_s, f_s]^T - [a, e, i, \Omega, w, f]^T \left(t_s\right) \right\|, \tag{2.26}$$

where λ_{orbit} is the weight of the terminal state deviations.

In addition to ensuring a stable flight, the peak value of the hinge torque and the peak power consumption of the servomechanisms need to be minimized:

$$\min J = \lambda_{MH} |MH| + \lambda_{sv} \int_0^{t_f} |MH \times \omega_{sv}| dt. \tag{2.27}$$

where $\lambda_{MH}, \lambda_{sv}$ are the weight coefficients, ω_{sv} is the angular velocity of servomechanisms.

This section constructs a complete motion model of a launch vehicle. It should be pointed out that the tracking guidance is still used when flying in the atmosphere, so the variables related to aerodynamics are interpolated according to the parameters of the nominal trajectory. The exo-atmospheric guidance methods are the focuses of the following discussion, and the rotational equations are not included in the AGMs due to the assumption that the attitude control can track the guidance commands well.

2.3 Exo-Atmospheric Analytical Guidance Methods

2.3.1 Basic Closed-Loop Guidance Method for Long March Launch Vehicles (LMLVs)

- The process of CLG

A closed-loop guidance method for LMLVs in a vacuum environment is summarized as follows.

Step 1: Release the expectation of the fixed-point injection, and take five OEs directly as terminal constraints.

Step 2: Find the most matching entry point according to the current state of the launcher.

The state includes the velocity, position, mass, specific impulse, and mass flow rate. The time-to-go and entry point are solved iteratively based on the above parameters, and the entry point is updated in each guidance cycle.

Step 3: Construct an optimization problem of the current guidance cycle in the orbital coordinate system (OCS).

The main features of this optimization problem are as follows:

1. The objective function is to minimize the fuel consumption;
2. The OE constraints are transformed into the state variables after an optimal entry point is found;
3. The terminal constraints in the OCS are further simplified, and only the velocity along the $o\xi$ axis and the position along the $o\eta$ axis are non-zero.

4. Make the following conversion:

$$\dot{W}_{x1}(t) = \frac{P_{x1}}{m(t)} = \frac{I_{sp} \cdot \dot{m}}{m_0 - \dot{m}t} = \frac{I_{sp}}{\tau - t}, \tag{2.28}$$

where P_{x1} is the axial thrust, m_0 is the initial mass, and $\dot{m}$ is the mass-flow rate. In this way, $\dot{W}_{x1}(t)$ is related to I_{sp} and the mass flow ratio $\tau = m_0/\dot{m}$, rather than the parameters that are difficult to measure in real time, such as P_{x1}, m_0, and $\dot{m}$. It should be noted that τ can be determined by $\dot{W}_{x1}(0)$, then the analytical expression of $\dot{W}_{x1}(t)$ can be obtained.

5. Design an analytic expression to represent the thrust direction, i.e., the pitch and yaw commands.
   Many simulations have shown that the optimal thrust direction in a vacuum environment can be approximated as a linear function of time, as follows:

$$\begin{cases} \varphi_{cx}(t) = \tilde{\varphi} + (-k_1 + k_2 \cdot t) \\ \psi_{cx}(t) = \tilde{\psi} + (-k_3 + k_4 \cdot t) \end{cases}. \tag{2.29}$$

6. Solve for the unknown variables in Eq. (2.29) based on the terminal velocity and position constraints, then obtain the guidance command of the current cycle for the real-time control.

Step 4: Repeat Steps 2 and 3 during each guidance cycle until the cutoff equations are met to shut down the engines.

The CLG is also known as the iterative guidance mode (IGM). The above processing assumes that the PTO lies within the performance scope of the launcher, so we can always find a matching entry point on the PTO, and the optimization problem is converted to the planning of fixed-point terminal constraints in each guidance cycle. In the following cycle, the entry point shall be updated again.

The CLG has the following advantages over the PGMs:

1. High injection accuracy. It predicts and regulates the entry point responsively, mostly matching the states of the launcher and guaranteeing that all OEs are met. The initial states, terminal constraints, and performance indices rather than a reference trajectory are considered in the real-time planning, allowing deviations from the prescribed flight path to counter interferences. On the contrary, PGMs can only satisfy few constraints or synthesized objectives, flying nearby the nominal trajectory.
2. Robust to thrust variations. This is due to its sensitivity to the change of the thrust, and the flight path would be re-planned in line with the variations.
3. Responsive to the target orbit adjustment. If the target is re-scheduled just before liftoff, only the new OEs need to be uploaded to the OBC, avoiding the hard work of the reference trajectory preparation and guide coefficients tuning.

- The solutions of the guidance law variables

Seven parameters in the guidance law of Eq. (2.29), i.e., $\tilde{\phi}$, $\tilde{\psi}$, k_1, $\sim$, k_4, and t, need to be solved. Note that t represents the time-to-go, also denoted as T_k. The CLG

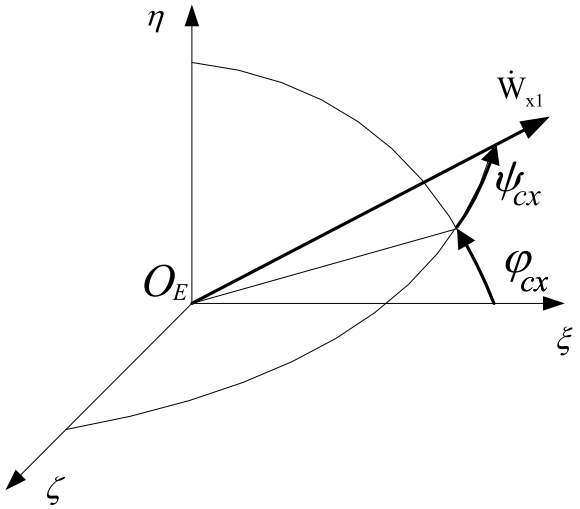


Fig. 2.2 Definition of thrust vector direction

is adopted when the rocket flies out of the atmosphere, so only the thrust and gravity are considered as the external forces, and the aerodynamic drag is omitted, which makes the following analytical solution possible.

The OCS is labeled as $O_E - \xi\eta\zeta$, where $O_E\eta$ points from the center of the earth to the injection point, $\xi O_E \eta$ denotes the orbital plan, and the three coordinate axes follow the right-hand rule, as shown in Fig. 2.2.

Consider that the OEs are set as the terminal constraints and the flight has been out of the atmosphere, the planning problem described in Sect. 2.2 is revised in the OCS as follows:

$$\text{Objective}: J = \int_0^{T_k} d\tilde{t} = T_k, \tag{2.30}$$

$$\begin{aligned} &Dynamics: \dot{\boldsymbol{X}} = \boldsymbol{f}(\boldsymbol{X}, \boldsymbol{u}, \tilde{t}), \\ &or \begin{cases} \dot{V}_\xi(\tilde{t}) = \dot{W}_{x1}(\tilde{t}) \cdot \cos\varphi^*(\tilde{t}) \cdot \cos\psi^*(\tilde{t}) + g_\xi(\tilde{t}) \\ \dot{V}_\eta(\tilde{t}) = \dot{W}_{x1}(\tilde{t}) \cdot \sin\varphi^*(\tilde{t}) \cdot \cos\psi^*(\tilde{t}) + g_\eta(\tilde{t}) \\ \dot{V}_\zeta(\tilde{t}) = -\dot{W}_{x1}(\tilde{t}) \cdot \sin\psi^*(\tilde{t}) + g_\zeta(\tilde{t}) \\ \dot{\xi}(\tilde{t}) = V_\xi(\tilde{t}) \\ \dot{\eta}(\tilde{t}) = V_\eta(\tilde{t}) \\ \dot{\zeta}(\tilde{t}) = V_\zeta(\tilde{t}) \end{cases}, \end{aligned} \tag{2.31}$$

Constraints:

$$\boldsymbol{X} = \left[V_{\xi 0} \;\; V_{\eta 0} \;\; V_{\zeta 0} \;\; \xi_0 \;\; \eta_0 \;\; \zeta_0 \right]^T, \tag{2.32}$$

$$N_1\left(\boldsymbol{X}\left(T_k\right), T_k\right) = 0. \tag{2.33}$$

The performance index, Eq. (2.30), is fuel-efficient and can also be expressed as the shortest time for the liquid launcher considering a basically constant thrust and mass flow rate. Equation (2.32) is the initial condition, and Eq. (2.33) represents the terminal constraints, i.e., the target orbital elements.

The control variables are the directions of the thrust vectors, which can be expressed by the Euler angles, ϕ_{cx} and ψ_{cx}, as follows:

$$\boldsymbol{u} = [\phi_{cx}(t), \psi_{cx}(t)]. \tag{2.34}$$

Hypothesis 1: A uniform gravitational field is introduced to simplify the state equations, i.e., the gravity is expressed by the average of the gravity of the current and entry point (the selection of the entry point will be introduced later):

$$\begin{cases} g_\xi(\tilde{t}) = \bar{g}_\xi \\ g_\eta(\tilde{t}) = \bar{g}_\eta \\ g_\zeta(\tilde{t}) = \bar{g}_\zeta \end{cases}. \tag{2.35}$$

The following Hamiltonian function is established:

$$\begin{aligned} H = 1 + \boldsymbol{\lambda}^t \boldsymbol{f} = 1 + \lambda_{V\xi}(\dot{W}_{x1} \cos\phi^* \cos\psi^* + \bar{g}_\xi) \\ + \lambda_{V\eta}(\dot{W}_{x1} \sin\phi^* \cos\psi^* + \bar{g}_\eta) + \lambda_{V\zeta}(-\dot{W}_{x1} \sin\psi^* \\ + \bar{g}_\zeta) + \lambda_\xi V_\xi + \lambda_\eta V_\eta + \lambda_\zeta V_\zeta. \end{aligned} \tag{2.36}$$

To maximize the Hamiltonian function Eq. (2.36), the following conditions should be met:

$$\frac{\partial H}{\partial \phi^*} = \dot{W}_{x1} \cos\psi^* \left(-\lambda_{V\xi} \sin\phi^* + \lambda_{V\eta} \cos\phi^*\right) = 0, \tag{2.37}$$

$$\frac{\partial H}{\partial \psi^*} = \dot{W}_{x1} \left(-\lambda_{V\xi} \cos\phi^* \sin\psi^* - \lambda_{V\eta} \sin\phi^* \sin\psi^* - \lambda_{V\zeta} \cos\psi^*\right) = 0. \tag{2.38}$$

By solving the above equations, we obtain the following equations:

$$\phi^* = \arctan \frac{\lambda_{V\eta}}{\lambda_{V\xi}}, \tag{2.39}$$

$$\psi^* = -\arctan \frac{\lambda_{V\zeta}}{\lambda_{V\xi}} \cos\phi^*. \tag{2.40}$$

The adjoint equations are as follows:

$$\begin{aligned} &\dot{\lambda}_{V\xi} = -\frac{\partial H}{\partial V_\xi} = -\lambda_\xi,\ \dot{\lambda}_{V\eta} = -\frac{\partial H}{\partial V_\mu} = -\lambda_\eta,\ \dot{\lambda}_{V\zeta} = -\frac{\partial H}{\partial V_\zeta} = -\lambda_\zeta, \\ &\dot{\lambda}_\xi = -\frac{\partial H}{\partial \xi} = 0,\ \dot{\lambda}_\eta = -\frac{\partial H}{\partial \eta} = 0,\ \dot{\lambda}_\zeta = -\frac{\partial H}{\partial \zeta} = 0. \end{aligned} \tag{2.41}$$

The following solution is derived from Eq. (2.41):

$$\begin{aligned}&\lambda_{V\xi} = \lambda_{V\xi 0} - \lambda_\xi \tilde{t},\ \lambda_{V\eta} = \lambda_{V\eta 0} - \lambda_\eta \tilde{t},\ \lambda_{V\zeta} = \lambda_{V\zeta 0} - \lambda_\zeta \tilde{t},\\ &\lambda_\xi = \lambda_{\xi 0},\ \lambda_\eta = \lambda_{\eta 0},\ \lambda_\zeta = \lambda_{\zeta 0}.\end{aligned} \tag{2.42}$$

(1) First, only the velocity constraints are considered.
If only the terminal velocity constraints are considered and the terminal position constraints are relaxed, then

$$\lambda_\xi = \lambda_\eta = \lambda_\zeta = 0. \tag{2.43}$$

Substituting Eqs. (2.43) and (2.42) into Eqs. (2.39) and (2.40), we obtain the optimal solutions of the control variables:

$$\phi^* = \arctan \frac{\lambda_{V\eta 0}}{\lambda_{V\xi 0}} = \tilde{\phi}, \tag{2.44}$$

$$\psi^* = -\arctan \frac{\lambda_{V\zeta 0}}{\lambda_{V\xi 0}} \cos\tilde{\phi} = \tilde{\psi}. \tag{2.45}$$

Thus, an important conclusion is drawn: the optimal control variables are constant if only the velocity constraints are taken into account. To determine this constant, we substitute Eqs. (2.44) and (2.45) into the first three terms of state equations given by Eq. (2.31):

$$\begin{cases} \dot{V}_\xi(\tilde{t}) = \dot{W}_{x1}(\tilde{t}) \cdot \cos\tilde{\varphi} \cdot \cos\tilde{\psi} + \bar{g}_\xi \\ \dot{V}_\eta(\tilde{t}) = \dot{W}_{x1}(\tilde{t}) \cdot \sin\tilde{\varphi} \cdot \cos\tilde{\psi} + \bar{g}_\eta \\ \dot{V}_\zeta(\tilde{t}) = -\dot{W}_{x1}(\tilde{t}) \cdot \sin\tilde{\psi} + \bar{g}_\zeta \end{cases}. \tag{2.46}$$

Assuming that the time-to-go, T_k, and the entry point are known, so the terminal velocity and position constraints are determined, then $\tilde{\phi}$ and $\tilde{\psi}$ can be obtained by integration:

$$\begin{cases} V_{\xi k} - V_{\xi 0} = L \cdot \cos\tilde{\phi} \cdot \cos\tilde{\psi} + \bar{g}_\xi \cdot T_k \\ V_{\eta k} - V_{\eta 0} = L \cdot \sin\tilde{\phi} \cdot \cos\tilde{\psi} + \bar{g}_\eta \cdot T_k \\ V_{\zeta k} - V_{\zeta 0} = -L \cdot \sin\tilde{\psi} + \bar{g}_\zeta \cdot T_k \end{cases}, \tag{2.47}$$

where $L = \int_0^{T_k} \dot{W}_{x1} d\tilde{t}$. Then,

$$\tilde{\phi} = \arctan \frac{V_{\eta k} - V_{\eta 0} - \bar{g}_\eta \cdot T_k}{V_{\xi k} - V_{\xi 0} - \bar{g}_\xi \cdot T_k}, \tag{2.48}$$

$$\tilde{\psi} = \arcsin \frac{-V_{\zeta k} + V_{\zeta 0} + \bar{g}_\zeta \cdot T_k}{L}. \tag{2.49}$$

$$L = \int_0^{T_k} \dot{W}_{x1} d\tilde{t} = \int_0^{T_k} \frac{I_{sp}}{\tau - \tilde{t}} d\tilde{t} = I_{sp} \cdot \ln \frac{\tau}{\tau - T_k}. \tag{2.50}$$

Hypothesis 2: For the guidance law of Eq. (2.29), $\tilde{\phi}$ and $\tilde{\psi}$ are used to ensure the terminal velocity conditions, and they are determined by Eqs. (2.48) and (2.49), respectively. The expressions $(-k_1 + k_2 \cdot \tilde{t})$ and $(-k_3 + k_4 \cdot \tilde{t})$ could be used to meet the terminal position conditions.
(2) Second, the position constraints are also included.
Thus the expressions of Eqs. (2.44), (2.45) are updated as follows:

$$\phi^* = \tilde{\phi} - k_1 + k_2 t, \ \psi^* = \tilde{\psi} - k_3 + k_4 t \tag{2.51}$$

Based on the known entry point, the terminal conditions can be transformed into the following:

$$\boldsymbol{X}(T_k) = \begin{bmatrix} V_{\xi k} \\ V_{\eta k} \\ V_{\zeta k} \\ \xi_k \\ \eta_k \\ \zeta_k \end{bmatrix} = \begin{bmatrix} V_k \cdot \cos\theta_k \\ V_k \cdot \sin\theta_k \\ 0 \\ 0 \\ R_k \\ 0 \end{bmatrix}, \tag{2.52}$$

where V_k is the required injection velocity, θ_k is the terminal velocity inclination, and R_k is the terminal geocentric vector diameter.

The state equation is converted to the following form:

$$\begin{cases} \ddot{\xi}(\tilde{t}) = \dot{W}_{x1}(\tilde{t}) \cdot \cos\varphi^*(\tilde{t}) \cdot \cos\psi^*(\tilde{t}) + \bar{g}_\xi \\ \ddot{\eta}(\tilde{t}) = \dot{W}_{x1}(\tilde{t}) \cdot \sin\varphi^*(\tilde{t}) \cdot \cos\psi^*(\tilde{t}) + \bar{g}_\eta \\ \ddot{\zeta}(\tilde{t}) = -\dot{W}_{x1}(\tilde{t}) \cdot \sin\psi^*(\tilde{t}) + \bar{g}_\zeta \end{cases}. \tag{2.53}$$

If $(-k_1 + k_2 \cdot \tilde{t})$ and $(-k_3 + k_4 \cdot \tilde{t})$ are small quantities, then

$$\begin{aligned} &\cos k_i \approx 1, \\ &\sin k_i \approx k_i, \\ &\cos\phi^* = \cos\tilde{\phi} + k_1 \sin\tilde{\phi} - k_2\tilde{t}\sin\tilde{\phi}, \\ &\sin\phi^* = \sin\tilde{\phi} - k_1 \cos\tilde{\phi} + k_2\tilde{t}\cos\tilde{\phi}, \\ &\cos\psi^* = \cos\tilde{\psi} + k_3 \sin\tilde{\psi} - k_4\tilde{t}\sin\tilde{\psi}, \\ &\sin\psi^* = \sin\tilde{\psi} - k_3 \cos\tilde{\psi} + k_4\tilde{t}\cos\tilde{\psi}. \end{aligned} \tag{2.54}$$

The expressions $(-k_1 + k_2 \cdot \tilde{t})$ and $(-k_3 + k_4 \cdot \tilde{t})$ are required not to have apparent effects on the terminal velocity, so

$$\int_0^{t_k} (-k_1 + k_2 \cdot t) dt = 0, \tag{2.55}$$

$$\int_0^{t_k} (-k_3 + k_4 \cdot t) dt = 0. \tag{2.56}$$

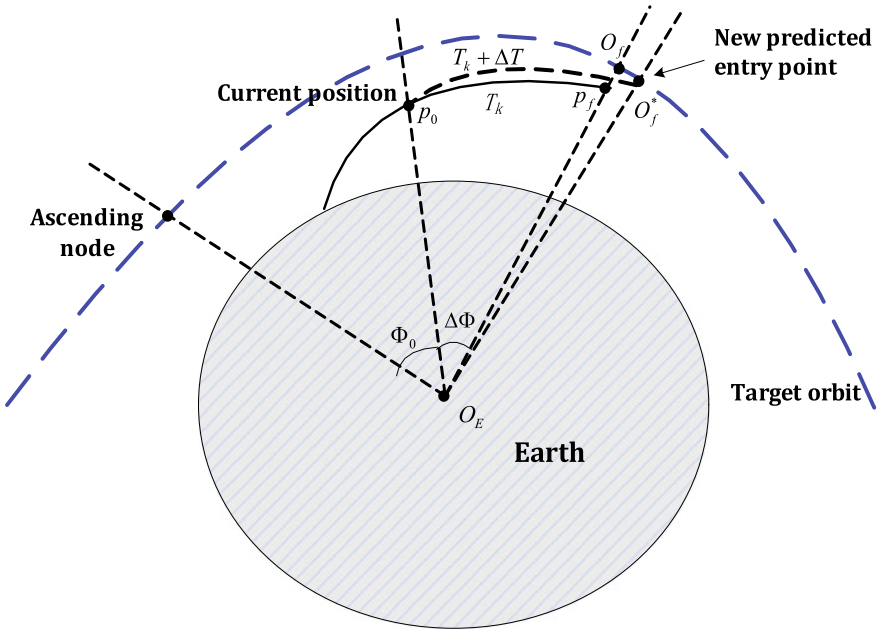


Fig. 2.3 Solving of optimal entry point

The longitudinal position is mainly determined by T_k. Thus the transverse and normal position constraints are:

$$\eta_k = \eta + v_\eta \cdot t_k + \int_0^{t_k} \int_0^t \ddot{\eta}(\tilde{t}) dt dt, \tag{2.57}$$

$$\zeta_k = \zeta + v_\zeta \cdot t_k + \int_0^{t_k} \int_0^t \ddot{\zeta}(\tilde{t}) dt dt. \tag{2.58}$$

The four equations, i.e., Eqs. (2.55)–(2.58), are used to solve $k_1 \sim k_4$. The integral processes are not complicated that the derivation is omitted, and interested readers can refer to [37].

(3) Finally, the time-to-go and the entry point are solved based on the geocentric angle.

T_k, the latest time-to-go, would be solved with the optimal entry point concurrently. This process is illustrated in Fig. 2.3.

P_0 is the current position of the launcher. The terminal position P_f can be predicted according to the CLG command planned in the current cycle. According to the geocentric angle between P_f and the ascending node $\Phi_0 + \Delta\Phi$, the position O_f on the target orbit can be determined with the same geocentric angle. If the velocities of P_f and O_f are the same, O_f is then regarded as the latest entry point, and T_k needs no compensation; otherwise, a correction time Δt should be found to ensure the new predicted terminal position after $T_k + \Delta t$ is located in the PTO, as shown as O_f^* in Fig. 2.3. O_f^* is also considered to be a new entry point.

According to the above analysis, T_k is updated by the following process:

$$v_0 + L(T_k) + \tilde{g}(T_k + \Delta t) + \frac{\partial L}{\partial t} \cdot \Delta t = f_v(\mathbf{S}) + \frac{\partial f_v(\mathbf{S})}{\partial t} \cdot \Delta t, \tag{2.59}$$

$$T_k \leftarrow T_k + \Delta t, \tag{2.60}$$

where v_0 is the current velocity of the rocket, $L(T_k)$ denotes the apparent velocity increment, $\tilde{g}(T_k + \Delta t)$ represents the gravitational effects on the velocity, f_v denotes the velocity of position $\boldsymbol{S}$ on the PTO, and $\boldsymbol{S}$ has the same geocentric angle as the predicted terminal position of the launcher.

After T_k converges to a stable value during the iteration process, the orbit entry point, O_f^*, is also determined. Thus the terminal velocity and position are known, which are used by Eqs. (2.35), (2.48), (2.49), (2.57), (2.58).

At this point, all the variables in Eq. (2.29) have been solved, and the guidance law is then updated and applied for current control. The above solving process is carried out iteratively in each guidance cycle.

In the above treatment, some approximates are made, which would produce deviations. However, as the rocket approaches the entry point, the accuracy of the above processing is also continuously improved. If the flight arc is long, the gravitational effect can be processed in segments, or a high-order approximation can be substituted.

The CLG is very sensitive to thrust variations, including thrust drops. Thus, it has a certain fault-tolerance ability by adjusting the flight path in time. When performing trajectory planning at fault time t_d and taking t_d as the start time of the following flight, we obtain

$$\dot{W}_{x1}(t_d) = \frac{I_{sp}}{\tau(t_d)}, \text{ i.e., } \tau(t_d) = \frac{I_{sp}}{\dot{W}_{x1}(t_d)}. \tag{2.61}$$

That is, $\tau(t_d)$ is updated during each planning cycle according to the apparent acceleration $\dot{W}_{x1}(t_d)$ measured by the IMU. Then, the dropped thrust is reflected in the apparent acceleration, which causes $\tau(t_d)$ to increase. Under the assumption that I_{sp} and T_k remain unchanged, L decreases according to Eq. (2.50), so Δt increases therefrom according to Eq. (2.59), and then T_k increases. This means that the flight time is extended and the orbit entry point is re-determined.

An example of LM-7 is given to illustrate IGM's adaptability to thrust variations. The engines of its second stage start up with a thrust of 150 kN, and then are tuned to 180 kN within 7 s. Four failure modes are considered, and the guidance commands (the pitch Euler angles) are compared with those under nominal conditions, as shown in Fig. 2.4, where the guidance commands are all re-planned after the failure occurs. However, the premise of the fault tolerance is that there is sufficient remaining performance to reach the PTO, and the measures if the performance is greatly degraded are discussed in Sect. 2.4.

Nowadays, many enhanced algorithms have evolved from this basic method. The improvements are mainly concentrated in Step 3 of Sect. 2.3.1, i.e., revising the form

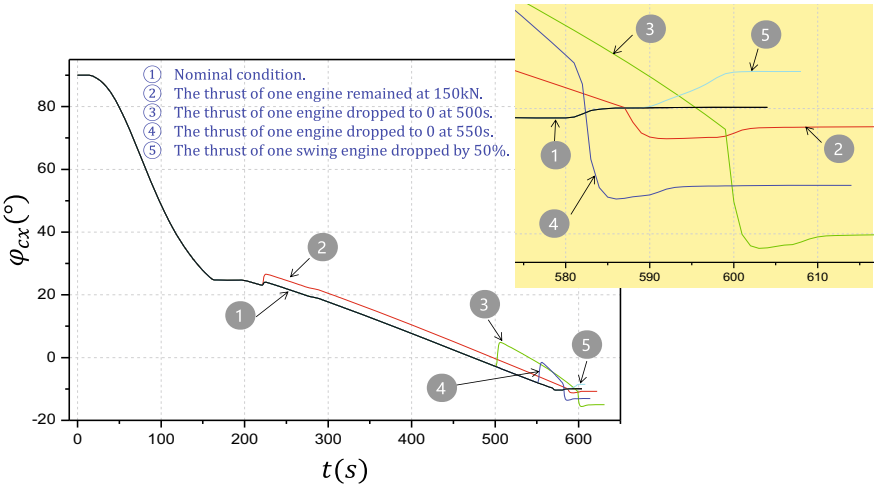


Fig. 2.4 Adaptability of CLG to thrust reduction

of the thrust directions or correcting the terminal velocity and position constraints. These upgraded versions of CLGs are discussed in the following sections.

2.3.2 Evolutions of the Closed-Loop Guidance Methods

The evolution methods provide acceptable suboptimal solutions under more complicated scenarios or constraints, which are proved to be feasible in real flights.

2.3.2.1 IGM Across Different Flight Phases

The above discussion only considers single powered flight phase. However, the IGM is not always applied in the last stage. The earlier the IGM is introduced, the more robust it is to faults.

During planning, the accelerations of the next stages need to be integrated to obtain the terminal states, so the algorithm's complexity is closely related to the number of stages or segments, and each additional segment will require additional calculations and control branches in the software. However, the number of segments is also closely related to the change of the thrust-to-weight ratio, which is determined by the trajectory characteristics. Therefore, the moment when the CLG is introduced for the real-time control should be thoroughly studied.

Segmentation for continuous powered phases is applied in the LM-2F/Y8 mission. The CLG was introduced after the fairing was jettisoned. The second stage of the LM-2F was operating at that moment, which was equipped with five engines, i.e.,

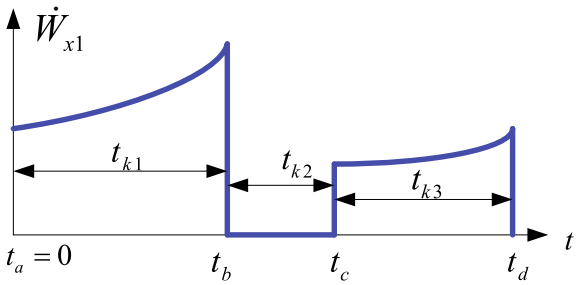


Fig. 2.5 Thrust variation with a coasting phase

one high-thrust main engine and four low-thrust swing engines. Thus, the flying was divided into two segments: all five engines were working, or only four swing engines were working after the main engine was shut down. During the working phase of the five engines, the equivalent impulse for the second flight segment is set based on the theoretical values of four swing engines; when the main engine shut down, it was updated based on the apparent acceleration measured by the IMU. Thus, the state equations were no longer continuous when facing different powered phases. An example including two powered phases (two burns) and a coasting phase in between is shown in Fig. 2.5.

The time-to-go has three components:

$$t_k = t_{k1} + t_{k2} + t_{k3}. \tag{2.62}$$

The corresponding apparent acceleration is as follows:

$$\dot{W}_{x1}(t) = \begin{cases} \frac{I_{sp1}}{\tau_1 - t} & t_a \le t < t_b \\ 0 & t_b \le t < t_c \\ \frac{I_{sp3}}{\tilde{\tau}_3 - t} & t_c \le t \le t_d \end{cases}. \tag{2.63}$$

As discussed above, $\tilde{\tau}_3$ is set as a theoretical value during the iterative computing of the first burn and then updated by the real flight data when the second burn initiates. Thus, the closed-loop guidance across different flight phases are implemented by replacing Eq. (2.28) with Eq. (2.63).

2.3.2.2 IGMs with Terminal Attitude Constraints

The guidance command is realized by adjusting the attitude of the launcher. To meet the terminal velocity and position constraints simultaneously at the orbit entry point, the guidance law requires the attitude to be tuned to a certain state to satisfy the thrust vector requirements. Furthermore, the actual flight path would deviate from the planned trajectory because of the existence of interferences and model

uncertainties. These lead to large dispersions between the real and nominal attitude when the payload is released, and the degree of dispersions is mainly related to the magnitude of the interferences and the flight accelerations.

The missions with terminal attitude constraints are not uncommon, and usually a reaction control system (RCS) is configured to regulate the attitude after the orbit injection. However, if the guidance method could satisfy the constraints, the RCS could be omitted to simplify the launcher, improve the reliability, and reduce the cost. This demands that the guidance method satisfies OEs and attitude constraints simultaneously only through the thrust vector control of the pitch, yaw, and cutoff sequences.

An upgraded quadratic time-to-go function expressing the thrust direction is proposed to meet the terminal attitude constraints, i.e., Eq. (2.29) is modified as follows:

$$\begin{cases} \varphi_{cx}(t) = \tilde{\varphi} + \left(-k_1 + k_2 \cdot t + k_5 \cdot t^2\right) \\ \psi_{cx}(t) = \tilde{\psi} + \left(-k_3 + k_4 \cdot t + k_6 \cdot t^2\right) \end{cases}. \tag{2.64}$$

The two new variables, k_5 and k_6, can be obtained based on the terminal pitch and yaw attitude constraints. The solving of these new variables can be found in Ref. [15], and the method is applied in the LM-2F/ T3 mission.

However, the terminal attitude cannot be set arbitrarily in this way. If the angle between the thrust and the terminal velocity direction is too large, the assumption that k_1-k_6 are small does not hold. This is the premise for deriving an analytical guidance law by simplifying the trigonometric functions. In the next section, another prediction and correction algorithm is proposed and compared to tackle the same problem.

2.3.3 *Prediction-Correction Iterative Guidance Method*

To avoid the singularity in the solving of the IGM when approaching the cutoff moment, the iterative calculation is terminated in advance before the engine shuts down. The variables of the guidance law then remain unchanged for the follow-up control. The errors arising therefrom should be compensated for, which is the initial purpose of the prediction-correction IGM. Its process is described as follows [38]:
(1) Based on Eq. (2.29), calculate the pitch and yaw commands when the IGM is terminated, $\phi(t_{f0})$ and $\psi(t_{f0})$, respectively, where $t_{f0} = t_f - \Delta t$, t_f represents terminal time, and t_{f0} is the moment when the IGM is terminated.
(2) Construct the apparent acceleration model $\dot{W}_{x1}(t)$:

$$\dot{W}_{x1}(t) = \frac{I_{sp}}{\tau(t) - t}, \tag{2.65}$$

where $\tau(t) = m(t_{f0})/\dot{m}(t)$, and t takes t_{f0} as the starting point:

$$\dot{W}_{x1}(t_{f0}) = \frac{I_{sp}}{\tau(t) - 0}. \tag{2.66}$$

Then,

$$\dot{W}_{x1}(t) = \frac{I_{sp}}{\frac{I_{sp}}{\dot{W}_{x1}(t_{f0})} - t}. \tag{2.67}$$

(3) Calculate the increases in the apparent velocity (or velocity) and position from t_{f0} to t_f, and then obtain the predicted terminal state $\mathbf{X}_k^*(t_f)$ under this condition, which can be expressed as a function of the following variables:

$$\mathbf{X}_k^*(t_f) = f_{\text{pts}}(\mathbf{X}_k(t_{f0}), \phi(t_{f0}), \psi(t_{f0}), \dot{W}_{x1}(t)). \tag{2.68}$$

where f_{pts} denotes the function to calculate the terminal state.

The thrust vector remains the same after t_{f0}.

(4) Calculate the compensation to the terminal constraints $\Delta\mathbf{X}_k(t_f)$ as follows:

$$\Delta\mathbf{X}_k(t_f) = \mathbf{X}_k(t_f) - \mathbf{X}_k^*(t_f). \tag{2.69}$$

(5) Update the terminal constraints $\mathbf{X}_k(t_f)$:

$$\mathbf{X}_k(t_f) \leftarrow \mathbf{X}_k(t_f) + \Delta\mathbf{X}_k(t_f). \tag{2.70}$$

The terminal constraints, which include the velocity and position constraints, were formerly determined by the entry point, however Eq. (2.70) renews them based on error predictions, while the time-to-go remains the same. Then, the variables of guidance law is re-calculated based on Eq. (2.70) in the current guidance cycle. Although these updated terminal constraints are not strictly optimal, the simulations show that the resulting errors are acceptable.

For different application scenarios, Eq. (2.69) has various updates. Note that $\Delta\mathbf{X}_k(t_f)$ represents the terminal constraint compensation induced by various factors during the period from t_{f0} (or other moments we are interested in) to t_f. These compensations are due to the systematic errors caused by conditions such as the in-advance termination of the IGM, the deviations caused by the tracking control or the cutoff thrust, and the other processing such as the attitude regulation.

2.3.3.1 Direct Injection Under High Thrust

Under high-thrust conditions before a payload is released to an orbit, the same attitude tracking error would result in large lateral or normal velocity deviations, and the disturbances and the uncertainties of the cutoff thrust also increase. All the above effects are adverse to the entry accuracy, and this is what LM-7 faces when launching cargo spacecrafts.

In general, a terminal velocity correction system (TVCS) could be installed to reduce the velocity errors after the main engines were shut down, but this auxiliary system would increase the complexity and cost of the launcher, reduce the carrying capacity and reliability. The final solution of LM-7 is to shut down two fixed engines in the last stage ahead of the cutoff of two other swing engines to reduce the overload before entering into an orbit. The time interval between these two cutoffs should not be too long to affect the launcher's performance, but a short interval would lead to a rapid time-varying thrust due to the coupling of the cutoff thrusts of four engines, and prominent variations in the guidance commands because the position constraints are very sensitive to the thrust. This leads to the increases in the attitude tracking errors and the injection deviations. To tackle this dilemma, the position constraints of the CLG are relaxed just before the cutoff of two fixed engines, letting the guidance commands quickly enter into a stable state. Although this strategy would produces systematic position errors, but a relatively accurate prediction of the terminal position is realized due to the stable guidance commands and high tracking accuracies, thus a prediction-correction scheme could be adopted before the shutdown of the two fixed engines.

The state vector at time t_f, $\mathbf{X}_k^{(2)}(t_f)$, which indicates the scenario that only two swing engines operate while the other two fixed engines shut down, could be predicted through the motion equation of the last stage in the vacuum regime. It should be pointed out that after two fixed engines shut down, t_f would not be updated again.

Similarly, $\mathbf{X}_k(t_f)$ represents the state vector predicted by the CLG if the four engines shut down simultaneously in the end. The deviation is solved as follows:

$$\Delta\mathbf{X}_k(t_f) = \mathbf{X}_k(t_f) - \mathbf{X}_k^{(2)}(t_f). \tag{2.71}$$

The above deviation is introduced into the IGM terminal constraint:

$$\mathbf{X}_k(t_f) \leftarrow \mathbf{X}_k(t_f) + \Delta\mathbf{X}_k(t_f). \tag{2.72}$$

The control variables of the IGM would not update when the two fixed engines shut down and remain unchanged until the end of the flight. More detailed discussion can refer to Ref. [13].

2.3.3.2 Error Correction of Terminal Velocity

Even if the TVCS is configured, the process of velocity corrections is general open-loop. The entry accuracy mostly depends on whether the state vectors at the cutoff time are consistent with the theoretical conditions. Owing to the influences of various disturbances and deviations (such as thrust deviations), it is most probable that the terminal state differs from the expected value. Thus, the open-loop velocity correction based on the prescribed command sequences will lead to non-negligible velocity errors, which is disadvantageous to the orbit entry accuracy. This problem can also

be solved by the prediction-correction strategy, but at this time, only the terminal velocity would be corrected.

After the engines shut down, the velocity increments caused by the cutoff thrust can be predicted as follows in the target OCS:

$$\begin{cases} \Delta W_{xcf} = \int_{t_f}^{t_{cut}} \dot{W}_{cf}(t) \cos\phi_f^* \cos\psi_f^* dt \\ \Delta W_{ycf} = \int_{t_f}^{t_{cut}} \dot{W}_{cf}(t) \sin\phi_f^* \cos\psi_f^* dt \\ \Delta W_{zcf} = \int_{t_f}^{t_{cut}} \dot{W}_{cf}(t) \sin\psi_f^* dt \end{cases}. \tag{2.73}$$

where φ_f^*, ψ_f^* are the real pitch and yaw control commands at the cutoff moment, $\dot{W}_{cf}$ is the apparent acceleration of the cutoff thrust, t_{cut} is the moment when the cutoff thrust ends, ΔW_{xcf}, ΔW_{ycf}, ΔW_{zcf} are the apparent velocity increments induced by the cutoff thrust under φ_f^*, ψ_f^* commands.

Similarly, the nominal apparent velocity increments based on the prescribed pitch and yaw angles can also be obtained as $[\Delta W_{xf}, \Delta W_{yf}, \Delta W_{zf}]^T$, and then,

$$\Delta \mathbf{X}_k(t_f) = [\Delta W_{xf} - \Delta W_{xcf}, \Delta W_{yf} - \Delta W_{ycf}, \Delta W_{zf} - \Delta W_{zcf}, 0, 0, 0]^T. \tag{2.74}$$

This scheme plays a major role in the LM-8/Y1 mission. The deviation between the terminal pitch command of the IGM and the nominal condition was 10.5°, and if no measures were adopted, the velocity errors caused by cutoff thrust would exceed the TVCS's correction ability, resulting in a timed shutdown of the TVCS (its working time was scheduled as 40 s). Thus, the velocity errors would not be fully compensated for, thereby affecting the injection accuracy. Benefiting from the above algorithm, the timed shutdown of the TVCS was avoided, and the accuracy of the semi-major axis was ensured and greatly improved.

2.3.3.3 Handling of Terminal Attitude Constraints

Based on the prediction-correction strategy, a new approach different from that in Sect. 2.3.2.2 is discussed to handle terminal attitude constraints. IGM is terminated at time t_{f0}, which is close to the terminal time t_f, and then an attitude adjustment phase is introduced to regulate the thrust vector from the current values $\phi(t_{f0})$, $\psi(t_{f0})$ to the expected terminal states ϕ_f, ψ_f. Taking the pitch channel as an example, the expression of the attitude adjustment is as follows:

$$\phi(t) = \begin{cases} \frac{2(\phi_f - \phi(t_{f0}))}{(t_{f1} - t_{f0})^2}(t - t_{f0})^2 + \phi_{f0}, t_{f0} \le t \le \frac{t_{f1}+t_{f0}}{2} \\ -\frac{2(\phi_f - \phi(t_{f0}))}{(t_{f1} - t_{f0})^2}(t - t_{f1})^2 + \phi_f, \frac{t_{f1}+t_{f0}}{2} < t \le t_{f1} \\ \phi_f, t_{f1} < t \le t_f \end{cases}, \tag{2.75}$$

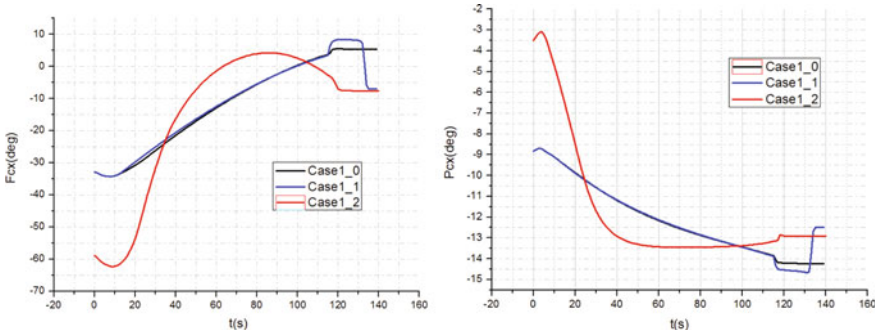


Fig. 2.6 Comparisons of Euler angles

where t_{f1} is the moment when the attitude adjustment ends, $\phi(t)$ represents the thrust vector at time t, and the expression $\psi(t)$ in the yaw channel is similar and not repeated.

The apparent acceleration during the attitude adjustment can be expressed as

$$\dot{W}_{x1}(t) = \frac{I_{sp}}{\tau - t}. \tag{2.76}$$

Based on Eqs. (2.75) and (2.76), we can determine the terminal state as follows:

$$\mathbf{X}_k^*(t_f) = f_{\text{pts}}(\mathbf{X}_k(t_0), \phi(t), \psi(t), \dot{W}_{x1}(t)). \tag{2.77}$$

The terminal compensation is

$$\Delta\mathbf{X}_k(t_f) = \mathbf{X}_k(t_f) - \mathbf{X}_k^*(t_f). \tag{2.78}$$

Three cases under different methods, i.e., the fundamental CLG (labeled as Case1_0), the method introduced in this section (labeled as Case1_1), and the method introduced in Sect. 2.3.2.2 (labeled as Case1_2), are compared in the Fig. 2.6, where F_{CX} and P_{CX} represent the pitch and yaw Euler angles, respectively.

Compared with the method introduced in Sect. 2.3.2.2, we can see an obvious attitude regulation process before entering into an orbit, and the guidance commands before the adjustment are more consistent with that of the fundamental method.

2.4 Joint Optimization of Target Orbit and Flight Path

For most launch failures caused by a thrust drop, the engines can continue to operate without an explosion. If the engines deteriorate to a very risky level indicated by the sensed data, active shutdown is preferable to ensure flight safety. No matter what

measures are taken, the carrying capacity will degrade. If still using the prescribed guidance law under this situation, whether an orbit is reachable depends on the remaining carrying capacity. Under severe failures, the propellant will be exhausted inflight, the terminal velocity and position will not be able meet the requirements of circling the Earth, and LV/payloads will crash to the ground. Thus, onboard decision making is required in the above situations to save missions.

The assumption for analytical CLGs is that we can always find an injection point on the PTO that matches the current flight states, and how to find the target orbit is not within the scope of guidance methods. However, this assumption does not hold if the PTO is beyond the performance capabilities of the rocket. A possible rescue orbit in which the rocket makes use of the remaining fuel should be found first, then the flight path should be planned or solved concurrently. A rescue orbit refers to a new target that is different from the PTO, where the payloads can enter as the starting point for the follow-up orbital transfer to avoid crashing. It has the same meaning as a parking orbit in most contexts, where the satellite can circle the Earth for many rounds. To consume as little fuel as possible by the payloads during the orbit transfer, an optimal rescue orbit becomes attractive. Under special conditions, a rescue orbit can also be a sub-orbit with a negative perigee height. The payload could not circle the Earth under this condition and should initiate the orbit transfer as soon as possible when released by the rocket.

It is difficult to find an analytical optimal rescue orbit, so a numerical method is usually adopted. This problem was first discussed in Refs. [40, 41]. In Ref. [40], the errors of different OEs were regulated through weights of element deviations in the objective. In Ref. [41], sequential optimization was conducted based on state-triggered-indices (STI), so as to gradually approach the optimal rescue solution. A convex optimization (COP) sub-problem was constructed, and its solution was taken as the initial value of the rescue planning problem. During the COP process, the geocentric angle of the injection point after the failure is estimated referring to the IGM process, then the COP sub-problem is transformed to the OCS to simplify the terminal constraints. This treatment greatly improves the calculation efficiency of the COP. Reference [39] proposed solutions for the maximum orbital radius optimization if the rescue orbit were confined to a circular orbit. However, none of the above methods can adapt to the flight scenario where a coasting phase is inserted. In Ref. [43], an autonomous mission reconfiguration algorithm considering the coasting phase was discussed to handle the typical failure modes that occur in real launchers, but the coasting orbit and the command sequences during coasting still refer to the prescribed planning results. This treatment can obtain a feasible solution when failures occur, but it may not make full use of the remaining performance. Thus the study of the multiple graded optimization (MGO) continues, while solving the MGO online is still very challenging.

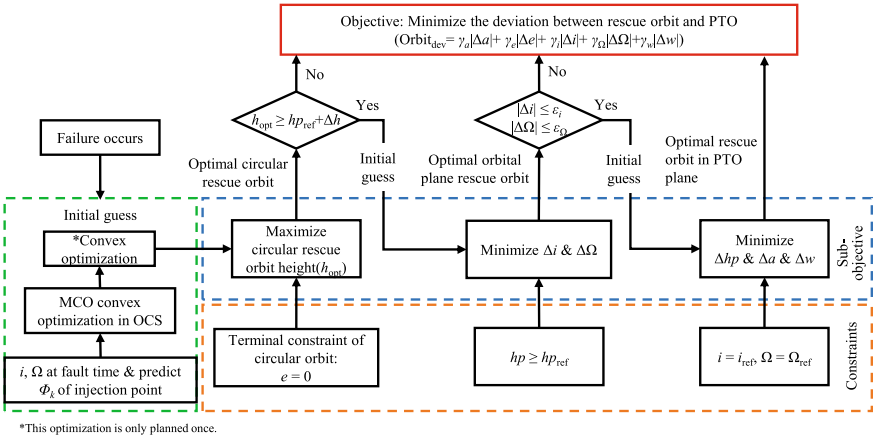


Fig. 2.7 Framework of STI method

2.4.1 State-Triggered-Indices (STI) Based Method for Continuous Powered Phases

The process of the STI-based optimization is explained in Fig. 2.7 [41]. The impacts of the deviations of the OEs on the fuel consumed to correct these errors are closely related to the orbital characteristics and the launcher's current state, which exhibit strong nonlinear features. Thus, the objective in Fig. 2.7 cannot be solved directly because of the concerns on the convergence or local optimal solutions, and it is transferred to three sub-problems.

(1) For the sake of safety, the orbital height should be ensured first after a failure occurs, so the first reaction is to find a maximum height circular orbit (MCO).

If the height is less than a safety value, it means the rocket can hardly stay in any orbit, the rescue is then abandoned. If the height meets the safe threshold but is less than the perigee height of the PTO, the circular orbit is then taken as the rescue orbit (optimal circular orbit, OCO). However, if the height is much higher, it indicates that there is a certain performance margin used to adjust other OE errors. Then, the next planning is triggered.

(2) The orbital inclination and LAN are regulated while ensuring the height of the perigee to obtain the optimal eclipse orbit (OEO).

The deviations of the inclination and LAN are eliminated as much as possible while keeping the perigee height of the rescue orbit around the required value. If the rescue orbit can be coplanar with the PTO, the following planning will be triggered again.

(3) The argument of the perigee, semi-major axis, and eccentricity are regulated while maintaining the perigee height and orbital planar elements to obtain the optimal rescue orbit (ORO).

During the optimization, the solution of the current sub-problem is taken as the initial guess of the next sub-problem. The initial guess can meet the equality constraints of all the motion equations, improving the convergence and efficiency of the numerical computation. However, to obtain a reasonable initial guess value of the first sub-problem, the nonlinear terminal constraints of the OEs are transformed to the OCS. This idea is inherited from the IGM, and the transformation matrix is as follows:

$$G_O = \begin{bmatrix} -\sin\Omega_0\cos i_0 & \cos\Omega_0\cos i_0 & \sin i_0 \\ \cos\Omega_0 & \sin\Omega_0 & 0 \\ -\sin\Omega_0\sin i_0 & \cos\Omega\sin i_0 & -\cos i_0 \end{bmatrix} \begin{bmatrix} \cos\Phi_k & -\sin\Phi_k & 0 \\ \sin\Phi_k & \cos\Phi_k & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (2.79)$$

where Φ_k is the geocentric angle between the orbit entry point and the ascending node (see Fig. 2.3).

In the OCS labeled as $O-\xi\eta\zeta$, the position components along the $O\xi$ and $O\zeta$ axes are 0, and the velocity components along the $O\eta$ and $O\zeta$ axes are 0. The terminal constraints are summarized as follows:

$$\xi_f = \zeta_f = 0,\ V_\eta = V_\zeta = 0,\ \mu = \eta_f V_{\xi f}^2. \quad (2.80)$$

Compared with the constraints in the LICS, i.e., Eqs. (2.16)–(2.23), (2.80) is greatly simplified, where Φ_k in Fig. 2.3 can be predicted as follows [40]:

$$\Phi_k = \Phi_0 + d\Phi, \quad (2.81)$$

where Φ_0 is the geocentric angle between the position of the launch vehicle at the current time and the ascending node, and $d\Phi$ is the geocentric angle in the current orbital plane during the remaining flight range.

$$d\Phi \approx \frac{d\Phi_{ref}}{\kappa}, \quad (2.82)$$

where κ is the percentage of nominal thrust after a failure occurs.

$$d\Phi_{ref} = \arccos\left(\frac{\mathbf{P}_{ref}\cdot\mathbf{P}_0}{\|\mathbf{P}_{ref}\|\cdot\|\mathbf{P}_0\|}\right). \quad (2.83)$$

An example is given below. For a PTO with $h_p = 200$ km and $h_a = 300$ km, the corresponding OEs are shown in Table 2.1. It is assumed that the thrust reduction is caused by the decrease in the mass flow rate at 118.2 s, and the remaining thrust is 77.94%. The results in Fig. 2.8 were obtained by the STI-based processing.

Traj_{ref} is the nominal flight path. At the fault moment, Φ_k is estimated as 171.05°, and $d\Phi_{est} = 15.6475°$. The OCS can be established when $i_0 = 41.27°$ and $\Omega_0 = 315.51°$ at the fault time. The result of the COP is shown by the blue line labeled Traj_{CVX}, which can be taken as the initial guess for the OCO by the adaptive collo-

Table 2.1 Orbital elements of prescribed target orbit

Orbital elements	a	e	i	w	Ω
Values	6628140 (m)	7.54×10^{-3}	42°	160°	315°

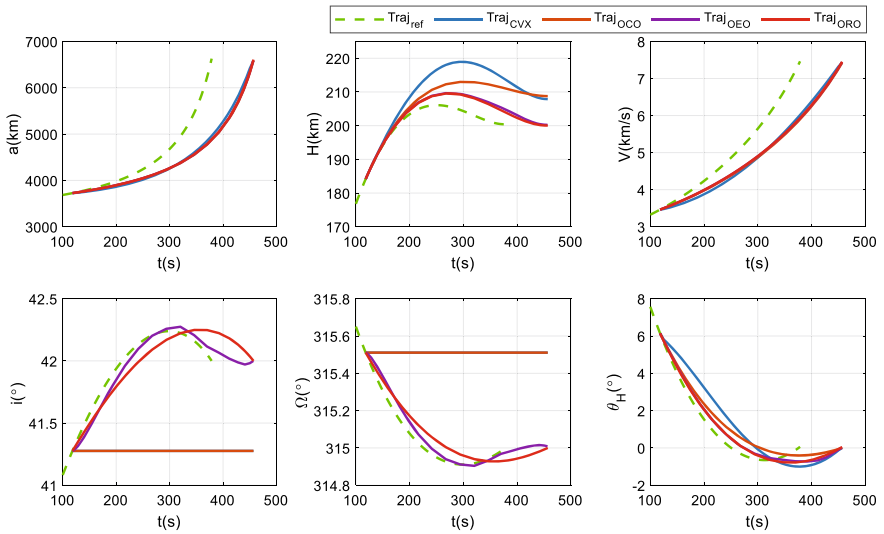


Fig. 2.8 Trajectories of rescue results

Table 2.2 Orbital elements of OCO, OEO and ORO

Orbital elements	a (km)	e	i (o)	Ω (o)	w (o)	h_p (km)	h_a (km)
PTO	6628.1	0.0075	42	315	160	200	300
COP	6586.0	0.0	41.28	315.51	–	207.9	207.9
OCO	6586.9	0.0	41.28	315.51	–	208.8	208.8
OEO	6586.0	0.0012	42	315.01	161.49	200.1	215.6
ORO	6607.6	0.0045	42	315	160.40	199.5	259.4

cation method. The OCO result is represented by Traj_{OCO}, and $d\Phi_{act} = 15.6496°$, showing that the deviation from the estimated value is 0.0021°.

Since the height of OCO is 208.8 km, which is greater than the h_p of the PTO, then the following planning is triggered, where $[\lambda_{hp}, \lambda_a, \lambda_i, \lambda_\Omega] = [10^{-3}, 10^{-3}, 1, 1]$. The result is shown as Traj_{OEO} with $h_p = 200.1$ km and $h_a = 215.6$ km.

By defining $\varepsilon_i = \varepsilon_\Omega = 0.05°$, Δi and $\Delta\Omega$ are both less than 0.05°, triggering the next planning. The result of the ORO is represented as Traj_{ORO}, and the OEs are summarized in Table 2.2.

Because the first planning of the OCO needs to solve the COP problem to obtain the initial guess, the calculation time accounts for more than 60% of the total planning time. By taking the OCO as the initial value, the solutions of the OEO and ORO problems can converge quickly.

More detailed discussion can be found in Ref. [41].

2.4.2 *Segmented Rescue Optimization Crossing Coasting Phase*

If a coasting phase is inserted into the optimization of the rescue orbit, the complexity of on-line planning will be further increased. Thus, a segmented planning strategy (SPS) is studied first, and the continuous solution is discussed in the next section. The SPS is similar to the IGM across the coasting phase, i.e., the coasting obit is taken as the target orbit of the first burn. Under nominal conditions, the IGM across the coasting arc does not lose optimality, because the terminal constraints of each flight phase are reachable. However, these constraints might not best match the remaining performance and guidance command sequences when a failure occurs, leading to the handover conditions between phases being unreachable.

However, the SPS relaxes the computational burden of online planning and demonstrates its effectiveness under the typical failure modes [42]. Under the background of launching satellites to the GTO using a two-stage rocket, its solutions are briefly explained as follows:

(1) Identify the fault mode first.
Three failure modes are considered. If the engine is going to explode, shut it down immediately, and let the subsequent stages make up for the performance loss of the premature cutoff. If an engine fails to start or shuts down by accident, restart it again if it has multiple burns. The restart may succeed or fail; even if it succeeds, it will make the engine unable to operate the following scheduled startup owing to the restriction on re-ignition times. However, the restarting scheme has the effect of postponing the fault moment and reducing the impact on the performance degradation. If only the thrust drops and there is no emergent risk, let the engine continue working. In the discussion in this section, it is assumed that there is no leakage, and all the remaining propellants can be utilized.
(2) Judge the flying regimes. If flying in the atmosphere and considering the landing area of the rocket debris, call the PGM for the tracing control until the fuel in the boosters is exhausted, then turn off the engines.
(3) If flying out of the atmosphere, evaluate the remaining performance by the ES-IGM algorithm [42].

If flying before coasting, first evaluate whether the prescribed coasting orbit is reachable; if flying after coasting, evaluate whether the PTO is reachable. The ES-IGM algorithm is based on the numerical integration and summarized in Ref. [42].

(4) Call the STI to optimize the new orbit and flight path if the prescribed orbit is beyond the remaining performance capabilities; otherwise, call the IGM for the guidance control.

The above algorithm is an approximate processing because the coasting phase is not optimized according to the fault state. If the coasting orbit is still reachable, the guidance command consequence is inherited during the coasting; if not, a new transfer orbit is planned by the STI method, and the triggering of the second burn is scheduled nearby the apogee of the new coasting orbit.

The discussion in Ref. [42] indicated that, if the PGM is adopted from the current point to the end or if the IGM is called only during the last burn, the LV/satellite may fall out of space under fault conditions with a high probability. In contrast, if the IGM is called as early as possible, the payload could be deployed into an orbit. This echoes the previous conclusions, the earlier the IGM is adopted inflight, the stronger the fault adaptability becomes. However, the IGM cannot guarantee a safe parking orbit, so the evaluation of the remaining performance is very important to support onboard decision making.

2.4.3 Multiple Graded Optimization

The STI method specifies the minimum orbital height as a safety constraint, for example, not less than 150 km. Thus, the satellite could circle the Earth and then carry out an orbital transfer at an appropriate point. If taking the payload as the final stage of a launcher, the flight process of the LV/payload can be jointly optimized, which is the meaning of the E2E optimization. At this time, we can relax the safety restrictions on the orbital height, even plan a sub-orbit (the perigee height is negative) to increase the apogee height, and make the orbital transfer responsively when flying to the apogee. E2E optimization can reduce the propellant consumed during the orbital transfer.

With the increase in commercial launches and constellations, multiple-satellite ridesharing launches are becoming more and more common. The purpose of the MGO is to separate some payloads in advance during the coasting phase while sending the remaining payloads to the PTO if the performance of the launcher is greatly reduced.

To clearly explain the MGO problem, the trajectory planning problems of the powered-coasting-powered profiles are summarized in Table 2.3. Offline numerical optimization is applied to analyze and compare the features of IGM, autonomous coasting reconstruction (ACRC), and MGO under thrust drop failures.

In Table 2.3, $\boldsymbol{F}_T$ is the nominal thrust, F_T^1 and F_T^2 are the nominal thrust magnitude of the 1st and the 2nd powered phase, respectively, and κ is the percentage of the remaining thrust to its nominal value; t_0 and t_f are defined as the fault time and the terminal time of the second stage, t_1 and t_2 are the engine cutoff time of the 1st powered phase and the start time of the 2nd powered phase, respectively; t_{c0}, t_{cf} are the initial and terminal times of the coasting phase; $t_{1\max}$ and t_{cool} are the

Table 2.3 Description of typical optimization problems

Method	IGM	ACRC	MGO
Objective	$J = -m\left(t_f\right)$		
Common constraints	$s.t.\ \ Dynamics:\ \dot{\boldsymbol{r}} = \boldsymbol{V},\ \dot{\boldsymbol{V}} = \frac{\kappa \boldsymbol{F}_T}{m} - \frac{\mu}{\|\boldsymbol{r}\|^3}\boldsymbol{r},\ \dot{m} = -\frac{\kappa\left\|\boldsymbol{F}_T\right\|}{I_{sp}g_0}$ $phase_1:\ \begin{cases}\left[\boldsymbol{r}_0,\ \boldsymbol{V}_0,\ m_0\right] = \left[\boldsymbol{r},\ \boldsymbol{V},\ m\right](t_0),\\ \|\boldsymbol{F}_T(t)\| = F_T^1,\ t \in \left[t_0, t_1\right],\ t_1 \le t_{1\max}\end{cases}$ $phase_c:\ \begin{cases}\left[\boldsymbol{r},\ \boldsymbol{V},\ m\right](t_{c0}) = \left[\boldsymbol{r},\ \boldsymbol{V},\ m\right](t_1),\\ \|\boldsymbol{F}_T(t)\| = 0,\ t \in \left[t_{c0}, t_{cf}\right],\end{cases}$ $phase_2:\ \begin{cases}\left[\boldsymbol{r},\ \boldsymbol{V},\ m\right](t_2) = \left[\boldsymbol{r},\ \boldsymbol{V},\ m\right](t_{cf}),\\ \|\boldsymbol{F}_T(t)\| = F_T^2,\ t \in \left[t_2, t_f\right],\\ \left[a_{ref},\ e_{ref},\ i_{ref},\ \Omega_{ref},\ w_{ref}\right]\\ = Fun_{orbit}\left(\boldsymbol{r}\left(t_f\right), \boldsymbol{V}\left(t_f\right)\right),\\ t_f - t_2 \ge t_{2\min},\end{cases}$		
Special constraints	$phase_1:$ $[a_c, e_c, i_c, \Omega_c, w_c]$ $= Fun\left(\boldsymbol{r}\left(t_1\right), \boldsymbol{V}\left(t_1\right)\right),$ $phase_c:$ $t_{cf} - t_{c0} = t_{coast},$ $phase_2:$ $m\left(t_f\right) \ge m_{\min}.$	$phase_c:$ $t_{cf} - t_1 \in [t_{cool}, t_{c\max}],$ $phase_2:$ $m\left(t_f\right) \ge m_{\min}.$	$phase_c:$ $\begin{cases}m\left(t_{c0}\right) = m\left(t_1\right) - m_{sep},\\ t_{cf} - t_1 \in [t_{cool}, t_{c\max}],\end{cases}$ $phase_2:$ $m\left(t_f\right) \ge m_{\min} - m_{sep}.$

maximum first burn time and the engine cooling time; $t_{c\max}$, $t_{2\min}$ are the maximum coast phase time and the minimum second burn time; t_{coast} is the standard coasting time, $m_{\min}$ and m_{sep} are the minimum mass of the rocket and the separation mass off the payloads.

For the IGM, the coasting OEs $[a_c, e_c, i_c, \Omega_c, w_c] = Fun\left(\boldsymbol{r}\left(t_1\right), \boldsymbol{V}\left(t_1\right)\right)$ are introduced as the terminal constraints in the first powered flight phase. After entering the coasting phase, a timed schedule is applied as the startup condition of the second powered flight phase. Then, the IGM is called again to fly to the PTO.

For the ACRC method, the planning of the powered-coasting-powered profiles is optimized simultaneously while taking all payloads as a whole, so the coasting orbit will be re-planned, and there are no fixed OEs as the constraints of the first burn. The coasting time is planned onboard only considering the cryogenic propellant management and the precooling time required to restart engines. The terminal mass constraint is the same as that of the IGM.

Compared with the ACRC, the MGO method considers the solution of departing parts of the payloads during the coasting. Thus, $m\left(t_{c0}\right) = m\left(t_1\right) - m_{sep}$, and the terminal mass constraint of the second burn is reduced accordingly.

- Fault adaptability analysis

The following analysis is based on a two-stage launcher, and the launch site and PTO parameters are shown in Tables 2.4 and 2.5.

Table 2.4 Main parameters

Rocket	Thrust (kN)	Specific impulse (m/s)	Structural mass (kg)	Propellant quality (kg)
Side booster	2400	2924	15000	141000
First stage	2400	2924	18500	153500
Second stage	328.8	4295	4350	16950
Fairing	–	–	2200	–

Table 2.5 Parameters of PTO and launch site

PTO				Launch site	
a (m)	e	i (∘)	Ω (∘)	Longitude (∘)	Latitude (∘)
7078140	0	50	94.5	110.95	19.61

We set t_{cool} as 60 s, $t_{c\max}$ as 850 s, and $t_{2\min}$ as 50 s. The adaptive collocation method is used to plan the nominal trajectory of the launcher off-line, as shown in Fig. 2.9. The superscript '1st' represents the flight state during the flight phase when the side and core boosters are working, and '2nd', '3rd', and '4th' represent the first burn, coasting, and second burn of the second stage, respectively. According to the optimization results, the performance of the rocket is 5840 kg without considering the orbital height constraints of the coasting orbit.

According to the nominal trajectory, t_{coast} in the IGM is defined as 528.5 s. It is assumed that the launcher carries 10 identical satellites, each weighing 584 kg. The failure time is introduced in the time interval of 200–350 s of the first burn, and the thrust after failure occurs is represented by a factor κ. The simulation results are shown in Fig. 2.10, where S3 represents the fault adaptation range of the IGM, S2 represents the range of the ACRC, which is more than that of the IGM, and S1 represents the range of the MGO, which is greater than that of the ACRC. Method 1 and Method 2 represent the lower limits of fault adaptation ranges of the IGM and ACRC, respectively, and Method 3-1 and Method 3-2 represent the lower limits of the MGO corresponding to departing 5 or 9 satellites during coasting, respectively.

For Method 1, the IGM can only endure the thrust dropping by 10% if the failure occurs at 200 s. With the delay of the failure, the dropping tolerance increases exponentially, and 33% of the total thrust can still send the payloads to the PTO if the failure occurs at 350 s. For Method 2, if the coasting could be re-planned online, the allowable dropping thrust could be extended to 64% at 200 s and 6% at 350 s. If the failure state of the thrust were deteriorated beyond the lower limit, for example, thrust dropping to 45% at 200 s, all the payloads could not enter into the PTO. However, for Method 3-1 under this condition, the MGO could send half of the payloads to the PTO by releasing the other half during coasting, avoiding the complete loss of the mission. The more payloads released during the coasting, the more severe thrust drop failure could be endured, but fewer payloads would be sent to the PTO.

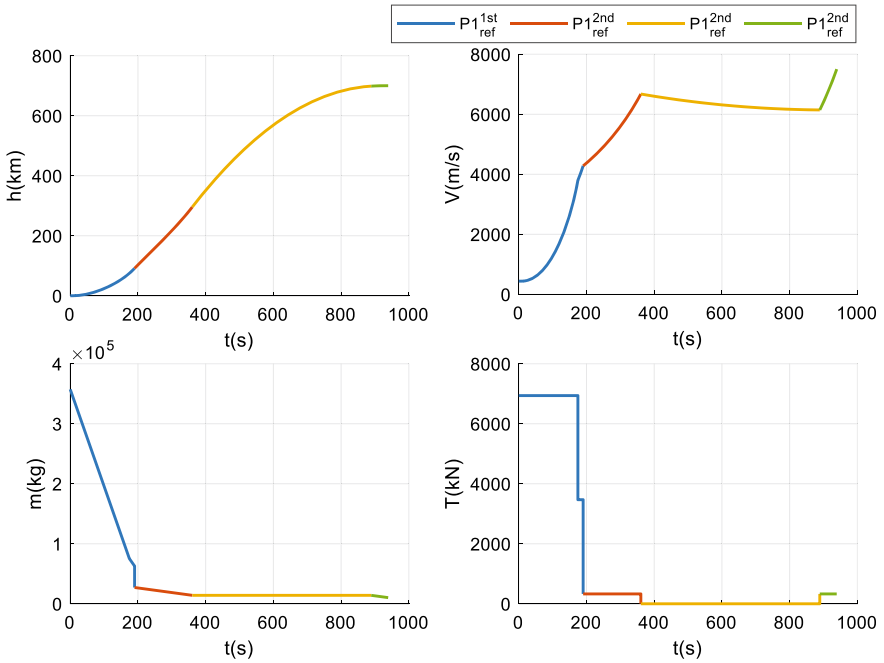


Fig. 2.9 Parameters of the nominal flight trajectory

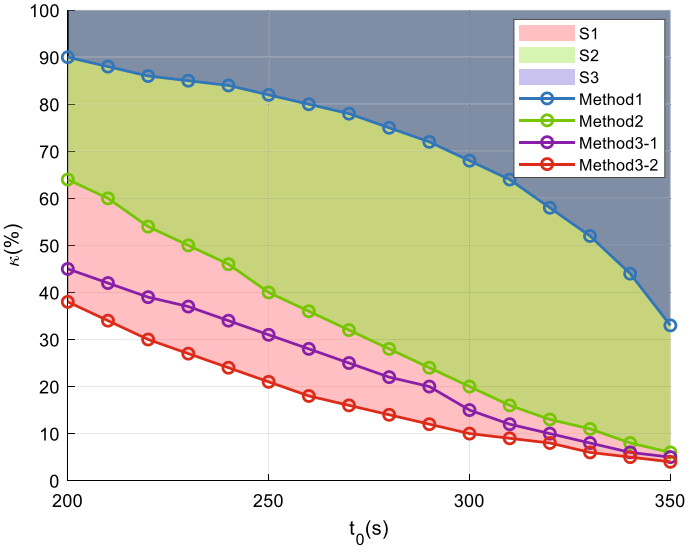


Fig. 2.10 Fault adaptation ranges

Compared with the IGM, the ACRC could adaptively adjust the coasting orbital inclination and LAN by extending the flight time of the first burn, regulate the shape of the coasting orbit, and elevate its perigee height to reduce the fuel consumption of the second burn. For the MGO method, the main mission of the second burn was to elevate the apogee height by boosting the speed, so the acceleration of the second burn could be improved by releasing parts of the payloads in advance.

- Case analysis

A test case is provided shown in Table 2.6.

If the minimum departure mass of 2625.5 kg could be determined onboard, five payloads should be separated in advance. With m_{sep} as 5×584 kg in the MGO, the optimization results are shown in Fig. 2.11.

The flight time of each phase by the MGO is shown in Table 2.7. The coasting OEs are shown in Table 2.8.

In the above analysis, the coasting orbit is optimized as a sub-orbit, and the satellites released during the coasting will inevitably crash to the ground. Another solution is to constrain a minimum safe perigee height of the coasting orbit, so the departed satellites could still circle the Earth and wait for rescue, but the number of satellites that could be put into the PTO would greatly decrease. No matter which solution was adopted, the MGO could avoid the complete loss of payloads for rideshare launches.

Table 2.6 Failure state of test case

t (s)	κ	x (km)	y (km)	z (km)	Vx (m/s)	Vy (m/s)	Vz (m/s)	m (kg)
250	31%	548.8	135.8	55.8	4863.0	718.4	81.9	25880.4

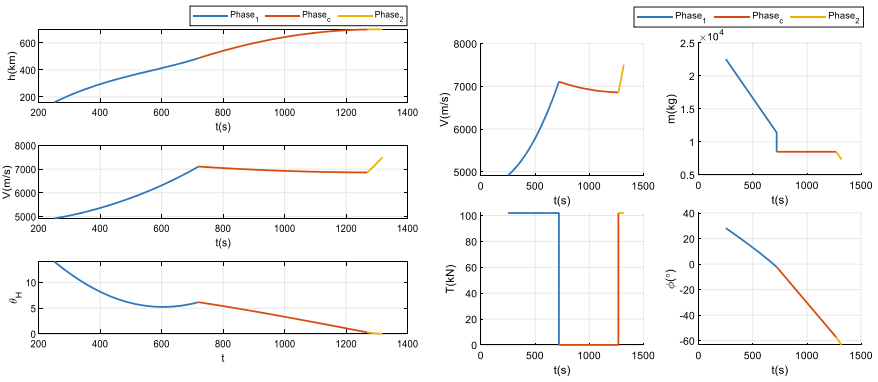


Fig. 2.11 MGO planning results

Table 2.7 Flight time

Cutoff of the first burn (s)	Startup of the second burn (s)	Cutoff of the second burn (s)
719.5659	1268.7850	1318.7850

Table 2.8 Coasting orbital elements

a (km)	e	i (∘)	Ω (∘)	w (∘)	h_p (km)	h_a (km)
6076.2	0.1648	50.0092	94.6643	–94.3177	–1303.5	699.7

However, the discussion in this section is based on off-line plannings, and onboard ACRC or MGO plannings are still challenging. A special issue is analyzed in Ref. [43], where the payload could still enter into the PTO by adjusting the coasting orbit.

2.5 Conclusions

Ascent guidance methods are a basic, fully studied, and seemly mature technology. The off-line planning and on-line tracking strategy were widely applied in the early stage and have achieved good results; they are even still used currently. Considering the wind load relief and the restrictions of the debris landing area, the OLG or tracking guidance is still playing a major role in endo-atmospheric flight.

CLGs perform well for exo-atmospheric guidance. No structural load constraints and atmospheric disturbances are considered. The optimization problem is then simplified to obtain analytical solutions based on optimal control theories. Compared with the tracking guidance, the CLGs are more adaptive to model uncertainties and interferences, and they are capable of satisfying multiple terminal constraints such as six OEs to obtain higher injection accuracies. If mild thrust drop failures occur, they can also be taken as disturbances handled by the CLGs.

It's assumed that model uncertainties and disturbances are bounded. If the faults' effect is far beyond the limits, the CLGs no longer work. Thus, the AGMs are attracting more interest. AGMs cannot ensure an entry into the PTO, because they cannot violate physical laws under severe failures, but they may reconstruct the mission to avoid the complete loss of the payloads.

In conclusion, the AGMs need to solve the following sub-problems: (1) onboard model identification or reconstruction, which mainly occurs in the case of abnormal conditions, such as loss of thrust; (2) evaluation of the remaining performance, which is to simplify the decision making: whether to use the CLGs to the PTO or to reconstruct the mission; and (3) determination of the optimization objectives, i.e., keeping the payload in an orbit, the end-to-end planning, or the graded optimization for rideshare launches.

Although there is no perfect or groundbreaking method to solve all the above problems, the collocation method with smart initial guesses provides a strategy for complex onboard planning. The convergence is not guaranteed, but it is better than doing nothing to let the LV/payload assembly fall from space. Any solution, although not optimal, is acceptable under these failure scenarios. The study of the AGMs to reconstruct the mission is just beginning, while the study of the analytical guidance is still important because it is often the first choice of the initial guess.

References

1. J.M. Hanson, M.W. Shrader, C.A. Cruzen, Ascent guidance comparisons. J. Astronaut. Sci. **43**(3), 307–326 (1995)
2. H.S. Chern, A open loop guidance architecture for navigationally robust on-orbit docking. NASA-CR-4687, August 1, 1995
3. A.W. Deaton, P.B. Kelley, Structural load reduction of the space shuttle booster orbiter configuration using a load relief guidance technique. NASA-TM-X-64738, April 5, 1973
4. N. Ahmad, E.J. Anzalone, A.S. Craig, G.A. Dukeman, Evolution and impact of saturn v on space launch system from a guidance, navigation, and mission analysis perspective, in *70th International Astronautical Congress* (Washington D.C., USA, 2019)
5. B. Kampos, Guidance, flight mechanics and trajectory optimization. Volume 9 - general perturbations theory. NASA-CR-1008, April 1, 1968
6. J.L. Speyer, B.S.A. Jarmark, Robust perturbation guidance for the advanced launch system, in *1989 American Control Conference* (1989), pp. 2489–2494
7. X.S. Cui, Guidance technique of cz-2e launch vehicle (in Chinese). Missil. Space Veh. **2**(6), 21–27 (1993)
8. W.B. Huang, Y.H. Zhang, S. Shi, W.H. Zhang, Research on the real-time calculation of perturbation guidance coefficients for the launch vehicle (in Chinese). J. Natl. Univ. Defense Technol. **35**(1), 19–23 (2013)
9. M.L. Tang, W. Qiu, Y. Wang, X.G. Zhang, Impact point control of first sub-stage of launch vehicle based on perturbation guidance (in Chinese). Missi. Space Veh. **4**, 68–71 (2017)
10. Z.K. Tian, S.C. Yang, D.L. Feng, Y.Z. Yao, Impact point prediction algorithm based on perturbation theory (in Chinese). Modern Defence Technol. **42**(3), 86–90 (2014)
11. I.E. Smith, General formulation of the iterative guidance mode. NASA TMX-53414 (1966)
12. H.J. Horn, The iterative guidance law for saturn, in *12th East Coast Conference on Aerospace and Navigational Electronics* (1965)
13. X.G. Lv, Z.Y. Song, Guidance methods of long-march launch vehicles (in Chinese). J. Astronaut. **38**(9), 895–902 (2017)
14. Z.Y. Song, H. Pan, C. Wang, Q.H. Gong, Development of flight control technology of long march launch vehicles (in Chinese). J. Astronaut. **41**(7), 868–879 (2020)
15. G.X. Shi, X.G. Lv, Q.H. Gong, Research on quadratic curve igm for multi-terminal constraints (in Chinese). Chinese Space Sci. Technol. **38**(2), 24–31 (2018)
16. W.G. Green, W.B. Tucker, Comparison of the atlas/centaur /surveyor/ and igm guidance concepts. NASA-TM-X-53674, November 22, 1964
17. W. Schleich, The space shuttle ascent guidance and control, in *Guidance and Control Conference* (San Diego, CA, U.S.A., 1982)
18. R.L. Mchenry, A.D. Long, B.F. Cockrell III., J.R. Thibodeau, T., J. Brand, Space shuttle ascent guidance, navigation, and control. J. Astronaut. Sci. **27**(1), 1–38 (1979)
19. P. Von der Porten, N. Ahmad, M. Hawkins, T. Fill, Powered explicit guidance modifications and enhancements for space launch system block-1 and block-1b vehicles, in *AAS GNC (Guidance, Navigation, and Control) Conference* (2018)
20. K.R. Brown, E.F. Harrold, G.W. Johnson, Rapid optimization of multiple-burn rocket flights, 8 march 1968 - 8 march 1969. NASA-CR-1430, September 1, 1969
21. P. Von der Porten, N. Ahmad, M. Hawkins, Closed loop guidance trade study for space launch system block-1b vehicle, in *AAS/AIAA Astrodynamics Specialist Conference* (2018)
22. H.S. Sun, *Closed-loop endo-atmospheric ascent guidance for reusable launch vehicle*. PhD thesis, Iowa State University (2005)
23. G. Dukeman, Atmospheric ascent guidance for rocket-powered launch vehicles, in *AIAA Guidance, Navigation, and Control Conference and Exhibit* (2002)
24. E. Bourgeois, O. Bokanowski, A. Désilles, H. Zidani, New improvements in the optimization of the launcher ascent trajectory through the hjb approach, in *7th European Conference for Aeronautics and Space Sciencesl (EUCASS)* (2017)

25. R.R. Bless. Time-domain finite elements in optimal control with application to launch-vehicle guidance. NASA-CR-4376, May 1, 1991
26. A. Vachon, A. Desbiens, E. Gagnon, C. Bérard, Launch ascent guidance by discrete multi-model predictive control. Acta Astronaut. **95**, 101–110 (2014)
27. A.J. Calise, D.H. Hodges, M.S. Leung, R.R. Bless, Optimal guidance law development for an advanced launch system. NASA-CR-192189, June 1, 1991
28. P. Lu, Introducing computational guidance and control. J. Guidance, Control, Dyn. **40**(2), 193 (2017)
29. R. Shotwell, J. Benito, A. Karp, J. Dankanich, Drivers, developments and options under consideration for a mars ascent vehicle, in *2016 IEEE Aerospace Conference* (2016)
30. J.A. Starek, E. Schmerling, G.D. Maher, B.W. Barbee, M. Pavone, Real-time, propellant-optimized spacecraft motion planning under clohessy-wiltshire-hill dynamics, in *2016 IEEE Aerospace Conference* (2016), pp 1–16
31. M.B. Goforth, Nasa-johnson space center engineering directorate overview and l-8 initiative. JSC-E-DAA-TN48720, November 13, 2017
32. T. Christopher, Y. Adnan, N. Nhan, (eds.), *Roadmap for Intelligent Systems in Aerospace* (AIAA - Intelligent Systems Technical Committee, 2016)
33. O.V. Yanova, B.G. Akobian, Launcher mission risk reduction due to the advanced adaptive guidance algorithms, in *the 67th International Astronautical Congress (IAC)* (Guadalajara, 2016)
34. R. Lugo, D. Litton, M. Qu, J. Shidner, R. Powell, A robust method to integrate end-to-end mission architecture optimization tools, in *2016 IEEE Aerospace Conference* (2016)
35. Z. Ali, H. David, C. Jérôme, E. Denis, G. Philippe, *Fault Diagnosis and Fault-Tolerant Control and Guidance for Aerospace Vehicles: From Theory to Application* (Springer, London, 2014)
36. Y.W. Xu, *Control System-(Part I) (in Chinese)* (China Aerospace Publishing House, 1989)
37. Z.Y. Song, *Design of High Reliability Launch Vehicle Control Systems (in Chinese)* (China Aerospace Publishing House, 2014)
38. Y. He, J. Wang, Z.Y. Song, J.X. Chai, H.F. Hu, Study and application of iterative guidance algorithm with adaptive prediction and compensation (in Chinese). J. Astronaut. **43**(6), 109–118 (2022)
39. Z.M. Hao, R. Zhang, Onboard real-time generation of launch vehicle abort orbits. J. Guid. Control. Dyn. **44**(8), 1541–1549 (2021)
40. Z.Y. Song, C. Wang, Q.H. Gong, Autonomous trajectory planning for launch vehicle under thrust drop failure (in Chinese). Sci. Sin. Inf. **49**(11), 1472–1487 (2019)
41. Z.Y. Song, C. Wang, Q.H. Gong, Joint dynamic optimization of the target orbit and flight trajectory of a launch vehicle based on state-triggered indices. Acta Astronaut. **174**, 82–93 (2020)
42. Z.Y. Song, Y. Liu, Y. He, C. Wang, Autonomous mission reconstruction during the ascending flight of launch vehicles under typical propulsion system failures. Chin. J. Aeronaut. **35**(6), 211–225 (2022)
43. C. Wang, Z.Y. Song, Powered-coast-powered guidance reconfiguration method of launch vehicle with thrust drop fault. Guidance, Navigat Control **02**(01), 2250003 (2022)

Chapter 3
Ascent Predictive Guidance for Thrust Drop Fault of Launch Vehicles Using Improved GS-MPSP

Xiaodong Yan and Cong Zhou

3.1 Introduction

Increasing complex space missions require launch vehicles to be with greater load-carrying capacity, better orbit injection accuracy and higher reliability. Such demands also cause the increased complexity of the vehicle, leading to a higher probability of fault, especially for the propulsion system. To remedy this issue, an advanced and robust ascent guidance capable of fault-tolerant is critical for the success of mission. Iterative guidance method [1] (IGM) and powered explicit guidance [2] (PEG) are two commonly used methods for the ascent phase of launch vehicles. These two guidance methods work well in the nominal condition and can adapt to many off-nominal conditions [3]. However, they lack of strong adaptive capacity, which cannot guarantee the reliability when the dynamic model or parameters change significantly. Alternatively, numerical approaches based on the optimal control theory may be the better choice. The existing algorithms can be divided into direct methods and indirect methods. Using the indirect methods, the guidance problem is transformed into Hamilton two-point boundary value problems [4] (TPBVP), but the solving process of this Hamilton two-point boundary value problem is complicated and highly sensitive to the initial guess. Using the direct method, the guidance problem is transformed into a nonlinear programming problem [5] (NLP). However, solving such problem is extremely computational intensive, which is difficult to meet the real-time requirement for online application.

In recent years, computational guidance has been proposed, which allows for onboard to generate guidance commands by the rapid trajectory optimization or

X. Yan (✉) · C. Zhou
School of Astronautics, Northwestern Polytechnical University,
710072 Xi'an, People's Republic of China
e-mail: yan804@nwpu.edu.cn

C. Zhou
e-mail: zhoucongnwpu@163.com

Z. Song et al. (eds.), *Autonomous Trajectory Planning and Guidance Control for Launch Vehicles*, Springer Series in Astrophysics and Cosmology,
https://doi.org/10.1007/978-981-99-0613-0_3

other numerical computation [6]. Convex optimization method is the most popular one in this field [7]. The advantage of convex optimization is that the convex problem can be reliably solved by the interior point method to gain global optimality in polynomial time, regardless of the initial guess. Owing to the high efficiency, the convex optimization methods have been applied into various guidance designs, such as entry guidance [8], landing guidance [9], as well as ascent guidance [10]. A Newton-Kantorovich pseudospectral convex programming method was presented in [10] to solve the ascent trajectory planning problem. The combination of Newton-Kantorovich and pseudospectral discrete furtherly improves the computational efficiency. Similarly, Li et al. [11] presented a pseudospectral based convex optimization approach to solve the ascent guidance problem in the presence of thrust drop fault. Song et al. [12] investigated online joint optimization of the target orbit and flight trajectory for launch vehicles under a propulsion system fault. Most recently, the optimal abort orbit problem is studied in [13], which employed the SOCP approach to solve the circular abort orbit with the maximum of the radius. Through these literature, the computational guidance and online programming methods for ascent problems have been studied extensively. However, most of them are still limited in terms of the computational efficiency.

Another promising approach for online application is model predictive static programming (MPSP) [14]. Owing to featuring an explicit closed-form solution and avoiding numerical complexities of optimal control theory, this method exhibits a higher computational efficiency over convex optimization methods. In our previous work, a new developed generalized quasi-spectral MPSP (GS-MPSP) has been proposed [15], which furtherly improves the computational speed by using spectral discretization and collocation method. This new algorithm also offers the advantage of smooth control and better usability, and hence holds great promise for online application.

In this chapter, based on the GS-MPSP philosophy, an ascent guidance for the thrust drop fault of the launch vehicle is presented. For the ascent guidance problem, the thrust drop fault may lead to the flight time substantially changing compared to the nominal trajectory, and a proper final time or orbit injection point is not easy to give out. In this case, the ability to search the final time in a large range is required for the guidance algorithm. The original GS-MPSP [15] is able to address the free terminal time problem by formulating a sensitive relation between the final time and final outputs, and taking the final time as the additional variable to be determined. However, this operation is quite rough and feasible just for slightly adjusting the terminal time when a proper initial guess is provided. Therefore, an improved GS-MPSP (IGS-MPSP) method is first proposed, which furtherly enhances the convergence robustness for the free final time in the presence of the poor initial guess by introducing a scale factor of time interval. Then, the ascent guidance is systematically formulated using this improved algorithm. Consequently, a numerical simulation for the case of injecting into a circular orbit is conducted to verify the effectiveness of the proposed method. The comparison with the SOCP based method is also carried out.

3.2 Generic Theory of the IGS-MPSP Method

The GS-MPSP method is proposed for efficiently solving a class of nonlinear terminal constraint problem. The considered nonlinear system dynamics is as follows:

$$\dot{\boldsymbol{X}}(t) = \boldsymbol{f}(\boldsymbol{X}, \boldsymbol{U}, t), t \in [t_0, t_f] \tag{3.1}$$

$$\boldsymbol{Y}(t) = \boldsymbol{h}(\boldsymbol{X}, t) \tag{3.2}$$

where $\boldsymbol{X} \in \boldsymbol{R}^n$, $\boldsymbol{U} \in \boldsymbol{R}^m$ and $\boldsymbol{Y} \in \boldsymbol{R}^p$ denote the state, control and output vectors, respectively. The purpose of this approach is to find suitable control history $\boldsymbol{U}(t)$ to ensure that the final system output $\boldsymbol{Y}_f(t_f)$ approaches the desired value $\boldsymbol{Y}_d$ with minimum control effort.

3.2.1 The Sensitivity Relation for Free-Terminal Time Continuous System

In this subsection, a sensitivity relation for the continuous system with the free -terminal time is derived. It is based on the sensitivity relation developed in Ref. [15, 16]. The brief introduction is presented in the following.

In the proposed method, it is considered that the terminal time is adjusted by uniformly scaling the length of time interval $[t_0, t_f]$. Thus, the updated terminal time can be written as:

$$t_f^{l+1} = t_f^l + \Delta\kappa \cdot (t_f^l - t_0) \tag{3.3}$$

where $\Delta\kappa \in R$ is the scale factor, and the superscript l and $l+1$ denotes the current step and the update step. The differential of Eq. (3.3) is given by

$$dt^{l+1} = dt^l + \Delta\kappa \cdot dt^l, t \in [t_0, t_f^l] \tag{3.4}$$

Note that in Eq. (3.4), the term $\Delta\kappa \cdot dt^l$ denotes the changed time length for each infinitesimal time interval, $dt, t \in [t_0, t_f^l]$. Next, the final output vector of the system (3.1) can be expressed as follow by introducing a weighting matrix $\boldsymbol{W}(t) \in \boldsymbol{R}^{p \times n}$

$$\boldsymbol{Y}(\boldsymbol{X}(t_f)) = \boldsymbol{Y}(\boldsymbol{X}(t_f)) + \int_{t_0}^{t_f} \left[\boldsymbol{W}(t) \cdot (\boldsymbol{f}(\boldsymbol{X}, \boldsymbol{U}, t) - \dot{\boldsymbol{X}})\right] dt \tag{3.5}$$

We then differentiate both sides of Eq. (3.5), it gives

$$d\boldsymbol{Y}(\boldsymbol{X}(t_f)) = \frac{\partial \boldsymbol{Y}(\boldsymbol{X}(t))}{\partial \boldsymbol{X}(t)} \cdot d\boldsymbol{X}(t_f) + \int_{t_0}^{t_f} \left[\boldsymbol{W}(t) \cdot \frac{\partial \boldsymbol{f}(\boldsymbol{X}, \boldsymbol{U}, t)}{\partial \boldsymbol{X}(t)} \cdot \delta \boldsymbol{X}(t) + \boldsymbol{W}(t) \cdot \frac{\partial \boldsymbol{f}(\boldsymbol{X}, \boldsymbol{U}, t)}{\partial \boldsymbol{U}(t)} \cdot \delta \boldsymbol{U}(t) - \boldsymbol{W}(t) \cdot d\dot{\boldsymbol{X}}(t) \right] dt \tag{3.6}$$

Note that in Eq. (3.6), $d\boldsymbol{X}(t)$ denotes the differential of $\boldsymbol{X}(t)$ taking into account the differential change of time interval, $\Delta\kappa \cdot dt$, and $\delta \boldsymbol{X}$ denotes the variation in $\boldsymbol{X}$ when the time interval is assumed to be fixed. They have the relationship as follow

$$d\boldsymbol{X}(t) = \delta \boldsymbol{X}(t) + \dot{\boldsymbol{X}}(t) \cdot \Delta\kappa \cdot dt \tag{3.7}$$

We conduct the first order differential of Eq.(3.7) respect to the time, it yields

$$d\dot{\boldsymbol{X}}(t) = \delta \dot{\boldsymbol{X}}(t) + \dot{\boldsymbol{X}}(t) \cdot \Delta\kappa \tag{3.8}$$

Substituting the $d\dot{\boldsymbol{X}}(t)$ for Eq. (3.8) into the term of the integrand on right side of Eq. (3.6) leads to

$$d\boldsymbol{Y}(\boldsymbol{X}(t_f)) = \frac{\partial \boldsymbol{Y}(\boldsymbol{X}(t))}{\partial \boldsymbol{X}(t)} \cdot d\boldsymbol{X}(t_f) + \int_{t_0}^{t_f} \left[\boldsymbol{W}(t) \cdot \frac{\partial \boldsymbol{f}(\boldsymbol{X}, \boldsymbol{U}, t)}{\partial \boldsymbol{X}(t)} \cdot \delta \boldsymbol{X}(t) + \boldsymbol{W}(t) \cdot \frac{\partial \boldsymbol{f}(\boldsymbol{X}, \boldsymbol{U}, t)}{\partial \boldsymbol{U}(t)} \cdot \delta \boldsymbol{U}(t) - \boldsymbol{W}(t) \cdot \dot{\boldsymbol{X}}(t) \cdot \Delta\kappa - \boldsymbol{W}(t) \cdot \delta\dot{\boldsymbol{X}}(t) \right] dt \tag{3.9}$$

By integrating the last term on the right side of Eq. (3.6), we obtain

$$\begin{aligned} d\boldsymbol{Y}(\boldsymbol{X}(t_f)) = {} & \frac{\partial \boldsymbol{Y}(\boldsymbol{X}(t_f))}{\partial \boldsymbol{X}(t_f)} \cdot \delta \boldsymbol{X}(t_f) - [\boldsymbol{W}(t) \cdot \delta \boldsymbol{X}(t)]_{t=t_f} + [\boldsymbol{W}(t) \cdot \delta \boldsymbol{X}(t)]_{t=t_0} \\ & + \int_{t_0}^{t_f} \left[\boldsymbol{W}(t) \cdot \frac{\partial \boldsymbol{f}(\boldsymbol{X}, \boldsymbol{U}, t)}{\partial \boldsymbol{X}(t)} \cdot \delta \boldsymbol{X}(t) + \dot{\boldsymbol{W}}(t) \cdot \delta \boldsymbol{X}(t) \right. \\ & \left. + \boldsymbol{W}(t) \cdot \frac{\partial \boldsymbol{f}(\boldsymbol{X}, \boldsymbol{U}, t)}{\partial \boldsymbol{U}(t)} \cdot \delta \boldsymbol{U}(t) - \boldsymbol{W}(t) \cdot \dot{\boldsymbol{X}}(t) \cdot \Delta\kappa \right] dt \end{aligned} \tag{3.10}$$

Equation (3.10) can be further rearranged as

$$d\boldsymbol{Y}(\boldsymbol{X}(t_f)) = \left(\frac{\partial \boldsymbol{Y}(\boldsymbol{X}(t_f))}{\partial \boldsymbol{X}(t_f)} - [\boldsymbol{W}(t)]_{t=t_f}\right) \cdot \delta \boldsymbol{X}(t_f) + [\boldsymbol{W}(t) \cdot \delta \boldsymbol{X}(t)]_{t=t_0} + \int_{t_0}^{t_f} \left[\left(\boldsymbol{W}(t) \cdot \frac{\partial \boldsymbol{f}(\boldsymbol{X}, \boldsymbol{U}, t)}{\partial \boldsymbol{X}(t)} + \dot{\boldsymbol{W}}(t)\right) \cdot \delta \boldsymbol{X}(t) + \boldsymbol{W}(t) \cdot \frac{\partial \boldsymbol{f}(\boldsymbol{X}, \boldsymbol{U}, t)}{\partial \boldsymbol{U}(t)} \cdot \delta \boldsymbol{U}(t) - \boldsymbol{W}(t) \cdot \dot{\boldsymbol{X}}(t) \cdot \Delta\kappa\right] dt \tag{3.11}$$

Here, it is necessary to choose $\boldsymbol{W}(t)$ so that eliminates the coefficients of $\delta \boldsymbol{X}(t)$ and $\delta \boldsymbol{X}(t_f)$ in Eq. (3.11), which leads to the following weighting matrix dynamics with the associated boundary condition at the final time t_f:

$$\dot{\boldsymbol{W}}(t) = -\boldsymbol{W}(t) \cdot \frac{\partial \boldsymbol{f}(\boldsymbol{X}, \boldsymbol{U}, t)}{\partial \boldsymbol{X}(t)} \tag{3.12}$$

$$\boldsymbol{W}(t_f) = \frac{\partial \boldsymbol{Y}(\boldsymbol{X}(t_f))}{\partial \boldsymbol{X}(t_f)} \tag{3.13}$$

In Eq. (3.11), it is straight to obtain $\delta \boldsymbol{X}(t_0) = 0$ with the fact that the initial conditions are specified. Then use this observation and the weighting matrix dynamic as presented in Eqs. (3.12) and (3.13), the Eq. (3.11) can be further simplified as

$$\begin{aligned} d\boldsymbol{Y}(\boldsymbol{X}(t_f)) &= \int_{t_0}^{t_f} \left[\boldsymbol{W}(t) \cdot \frac{\partial \boldsymbol{f}(\boldsymbol{X}, \boldsymbol{U}, t)}{\partial \boldsymbol{U}(t)} \cdot \delta \boldsymbol{U}(t)\right] dt + \int_{t_0}^{t_f} \left[-\boldsymbol{W}(t) \cdot \dot{\boldsymbol{X}}(t) \cdot \Delta\kappa\right] dt \\ &= \int_{t_0}^{t_f} [\boldsymbol{B}_s(t) \cdot \delta \boldsymbol{U}(t)]\, dt + \boldsymbol{B}_\kappa \cdot \Delta\kappa \end{aligned} \tag{3.14}$$

where

$$\boldsymbol{B}_s(t) \triangleq \boldsymbol{W}(t) \cdot \frac{\partial \boldsymbol{f}(\boldsymbol{X}, \boldsymbol{U}, t)}{\partial \boldsymbol{U}(t)} \tag{3.15}$$

$$\boldsymbol{B}_\kappa \triangleq -\int_{t_0}^{t_f} \left[\boldsymbol{W}(t) \cdot \dot{\boldsymbol{X}}(t)\right] dt \tag{3.16}$$

where $\boldsymbol{B}_s(t)$ is the sensitivity matrix that relates the error $\delta \boldsymbol{U}$ to $d\boldsymbol{Y}$ as per Ref. [16]. And $\boldsymbol{B}_\kappa$ can be interpreted as the sensitivity matrix that relates the scale factor of time internal, $\Delta\kappa$ to the final out error $d\boldsymbol{Y}$. Note that in here, the sensitive relation for terminal time is indirectly formulated by the scale factor of time internal, $\Delta\kappa$. Compared with the original way in GS-MPSP, this strategy is more accurate since

it uniformly scales each infinitesimal time interval on $[t_0, t_f]$, rather than roughly adjust the final time.

3.2.2 The Mathematical of IGS-MPSP Method

In IGS-MPSP, the control vector is represented by a weighted summation of some basic spectral functions to reduce the number of various to be optimized:

$$\boldsymbol{U}(t) = \sum_{i=1}^{N_p} \boldsymbol{C}_i P_i(t) \tag{3.17}$$

where $\boldsymbol{C}_j = [c_{1j}, c_{2j}, \ldots, c_{mj}]^T$ is the coefficient vector corresponding to the jth spectral function. N_p is the number of basic functions in the expression, and $\boldsymbol{P}_j(t)$ is the basic spectral function. The spectral functions can be selected as Legendre series, Chebyshev series, etc.

Next, the new developed relationship as presented in Eq. (3.14) is applied to derive the GS-MPSP method for the free-time problem. Since the control history $\boldsymbol{U}(t), t \in [t_0, t_f]$ has been represented by the spectral functions in Eq. (3.17), and a new scale factor of time internal, $\Delta\kappa$, is introduced to adjust the unspecified final time as in Eq. (3.3), both the coefficient vector $[\boldsymbol{C}_1, \boldsymbol{C}_2, \ldots, \boldsymbol{C}_{N_p}]$ and scale factor $\delta\kappa$ are selected as variable to be determined.

First, the variation of the control history during the time $t = [t_0, t_f]$ can be obtained from Eq. (3.17):

$$d\boldsymbol{U}(t) = \sum_{j=1}^{N_p} d\boldsymbol{C}_j P_j(t) \tag{3.18}$$

Substituting Eq. (3.18) into Eq. (3.14), it yields

$$\begin{aligned} d\boldsymbol{Y}_N &= \int_{t_0}^{t_f} \left[\boldsymbol{B}_s(t) \cdot \sum_{j=1}^{N_p} \delta\boldsymbol{C}_j P_j(t) \right] dt + \boldsymbol{B}_\kappa \cdot \Delta\kappa \\ &= \sum_{j=1}^{N_p} \boldsymbol{A}_j \cdot d\boldsymbol{C}_j + \boldsymbol{B}_\kappa \cdot \Delta\kappa \end{aligned} \tag{3.19}$$

in which

$$\boldsymbol{A}_j = \int_{t_0}^{t_f} \boldsymbol{B}_s(t) \cdot P_j(t) dt,\ j = 1, 2 \ldots N_p \tag{3.20}$$

In Eq. (3.19), $\boldsymbol{A}_j$ is the spectral sensitivity matrix as per Ref. [15], which relates the error of the coefficient of the jth spectral function, $d\boldsymbol{C}_j$, to the error of the output $d\boldsymbol{Y}_N$. Thus, a linear formula for the error of the final output and the error of each coefficient vector as well as the scale factor is obtained. Then the desired coefficients and scale factor can be worked out by formulating a static programming problem:

The update equation for coefficient vectors can be written as

$$\boldsymbol{C}_j^{l+1} = \boldsymbol{C}_j^l - d\boldsymbol{C}_j^l \tag{3.21}$$

where $\boldsymbol{C}_j^l$ denotes as the jth coefficient at the current iteration (represented by superscript l), and $\boldsymbol{C}_j^{l+1}$ denotes the updated coefficient in the next iteration (represented by superscript $l+1$). After substituting the expression of $d\boldsymbol{C}_j^l$ in Eq. (3.21) into Eq. (3.19), the error of final output can be written as

$$d\boldsymbol{Y}_N = \sum_{j=1}^{N_p} \boldsymbol{F}_j \delta \boldsymbol{C}_j^l = \sum_{j=1}^{N_p} F_j(\boldsymbol{C}_j^l - \boldsymbol{C}_j^{l+1}) = \boldsymbol{c}_\lambda - \sum_{j=1}^{N_p} \boldsymbol{A}_j \boldsymbol{C}_j^{l+1} + \boldsymbol{B}_\kappa \cdot \Delta\kappa \tag{3.22}$$

where

$$\boldsymbol{c}_\lambda = \sum_{j=1}^{N_p} \boldsymbol{A}_j \boldsymbol{C}_j^l \tag{3.23}$$

Equation (3.22) is apparently a linear equations set about $\boldsymbol{C}_j^{l+1}$ and $\Delta\kappa$, which contains $N_p \times m + 1$ unknowns and p equations where $N_p \times m + 1 > p$. Hence, Eq. (3.22) represents an under-constrained system and it is necessary to impose an appropriate performance index to facilitate a solution. In here, we consider to minimum the control effort, that is

$$J = \frac{1}{2} \int_{t_0}^{t_f^l} \left[\boldsymbol{U}^{l+1}(t)^T \boldsymbol{R}(t) \boldsymbol{U}^{l+1}(t) \right] \cdot dt + R_\kappa \Delta\kappa^2 \tag{3.24}$$

where $\boldsymbol{R}(t)$ and R_κ are the weight matrix for control and scale factor. Substituting Eq. (3.17) into Eq. (3.24), it yields

$$J = \frac{1}{2} \int_{t_0}^{t_f^l} \left[\left(\sum_{i=1}^{N_p} \boldsymbol{C}_i^{l+1} P_i(t) \right)^T \boldsymbol{R}(t) \left(\sum_{i=1}^{N_p} \boldsymbol{C}_i^{l+1} P_i(t) \right) \right] \cdot dt + R_\kappa \Delta\kappa^2 \tag{3.25}$$

Then, combined with the cost function (3.25) and the constraints given in Eq. (3.22), a static programming problem for the coefficient vector and scale vector can be formulated. The augmented cost function of this problem is given by

$$\bar{J} = J + \boldsymbol{\lambda}^T \left(d\boldsymbol{Y}_N - \boldsymbol{c}_\lambda + \sum_{j=1}^{N_p} \boldsymbol{A}_j \boldsymbol{C}_j^{l+1} - \boldsymbol{B}_\kappa \cdot \Delta\kappa \right) \tag{3.26}$$

where $\boldsymbol{\lambda} \in \boldsymbol{R}^p$ is Lagrange multiplier. The first-order optimality conditions are

$$\frac{\partial \bar{J}}{\partial d\boldsymbol{C}_j} = \sum_{i=1}^{N_p} \boldsymbol{R}_{ij} \cdot \boldsymbol{C}_j^{l+1} - \boldsymbol{A}_j{}^T \cdot \boldsymbol{\lambda} = \boldsymbol{0}, \quad j = 1, 2 \cdots N_p \tag{3.27}$$

$$\frac{\partial \bar{J}}{\partial \Delta\kappa} = R_\kappa \cdot \Delta\kappa - \boldsymbol{B}_\kappa{}^T \cdot \boldsymbol{\lambda} = \boldsymbol{0} \tag{3.28}$$

where

$$\boldsymbol{R}_{ij} = \int_{t_0}^{t_f^l} \left[P_i(t) \boldsymbol{R}(t) P_j(t) \right] \cdot dt \tag{3.29}$$

Thus, Eqs. (3.22), (3.27) and (3.28) make up a equation set about $\boldsymbol{C}_j^{l+1}$, $\boldsymbol{\lambda}$ and $\delta\kappa$, which can be written as a compact form as follow

$$\boldsymbol{D} \cdot \tilde{\boldsymbol{X}} = \boldsymbol{E} \tag{3.30}$$

where

$$\boldsymbol{D} = \begin{bmatrix} \boldsymbol{R}_{11} & \cdots & \boldsymbol{R}_{1N_p} & -\boldsymbol{A}_1{}^T & 0 \\ \vdots & \ddots & \vdots & \vdots & 0 \\ \boldsymbol{R}_{N_p 1} & \cdots & \boldsymbol{R}_{N_p N_p} & -\boldsymbol{A}_{N_p}{}^T & 0 \\ \boldsymbol{A}_1 & \cdots & \boldsymbol{A}_{N_p} 0 & -\boldsymbol{B}_\kappa & \\ 0 & \cdots & 0 & -\boldsymbol{B}_\kappa{}^T & R_\kappa \end{bmatrix}, \tilde{\boldsymbol{X}} = \begin{bmatrix} \boldsymbol{C}_1^{l+1} \\ \vdots \\ \boldsymbol{C}_{N_p}^{l+1} \\ \boldsymbol{\lambda} \\ \Delta\kappa \end{bmatrix}, \boldsymbol{E} = \begin{bmatrix} \boldsymbol{0} \\ \vdots \\ \boldsymbol{0} \\ \boldsymbol{c}_\lambda - d\boldsymbol{Y}_N \\ 0 \end{bmatrix} \tag{3.31}$$

The Eq. (3.30) contains $N_p \times m + p + 1$ unknowns (i.e., $\boldsymbol{C}_1^{l+1}, \boldsymbol{C}_2^{l+1} \cdots \boldsymbol{C}_{N_p}^{l+1}$, $\boldsymbol{\lambda}, \Delta\kappa$) and the same number of equations. Assuming that the matrix $\boldsymbol{D}$ is nonsingular, the unknown vector $\tilde{\boldsymbol{X}}$ can be solved by

$$\tilde{\boldsymbol{X}} = \boldsymbol{D}^{-1} \cdot \boldsymbol{E} \tag{3.32}$$

Consequently, the updated coefficients $\boldsymbol{C}_j^{l+1}$ and scale factor $\Delta\kappa$ are obtained from the solution of $\tilde{\boldsymbol{X}}$. Then substituting the $\Delta\kappa$ into Eq. (3.3), the update terminal time can be obtained by

$$t_f^{l+1} = t_f^l + \Delta\kappa \cdot (t_f^l - t_0) \tag{3.33}$$

And substitute $\boldsymbol{C}_j^{l+1}(j = 1, 2 \cdots N_p)$ into Eq. (3.17), the updated control history at time $t \in [t_0, t_f^{l+1}]$ is eventually given by

$$\boldsymbol{U}^{l+1}(t) = \sum_{i=1}^{N_p} \boldsymbol{C}_i^{l+1} P_i(t), t \in [t_0, t_f^{l+1}] \tag{3.34}$$

Remark 1 To implement the IGS-MPSP algorithm, the sensitivity matrix $\boldsymbol{A}_j$, $\boldsymbol{B}_\kappa$ and $\boldsymbol{R}_{ij}$ are necessary to be worked out in each iteration. The Gauss Quadrature Collocation method can be applied to efficiently compute such matrix and ensure the computational efficiency of this approach. The detailed procedure will be presented in the next subsection.

Remark 2 Compared with the original algorithm [15], the improved method introduces a scale factor of time internal to adjust the terminal time, and accordingly a sensitive relation for this factor is derived. This way improves the accuracy of sensitive relation for terminal time and hence is able to search the final time in a wide range when a poor initial guess is provided. That is, the convergence robustness for initial guess of final time is improved.

3.2.3 The Computation of Sensitive Matrix by Gauss Quadrature Collocation

In this subsection, the Gauss Quadrature Collocation method is applied to efficiently compute the sensitive matrix $\boldsymbol{A}_j$, $\boldsymbol{B}_\kappa$ and $\boldsymbol{R}_{ij}$. The detailed procedure is presented as follow.

For the convenience of solving, the physical time $t \in [t_0, t_f]$ is converted to the scale time $\tau \in [-1, 1]$ by the following relation:

$$t \equiv t(\tau, t_0, t_f) = \frac{t_f - t_0}{2}\tau + \frac{t_f + t_0}{2} \tag{3.35}$$

Next, the collocation method is used to solve the weighting dynamic equation as presented in Eqs. (3.12) and (3.13). First, we rewrite the matrix equation (3.12) as the following vector equation with the independent variable τ:

$$\begin{aligned} &\dot{\boldsymbol{W}}_k(\tau) = -\boldsymbol{W}_k(\tau) \cdot \boldsymbol{f}_x(\tau) \\ &k = 1, 2, \ldots, p \end{aligned} \tag{3.36}$$

where $\boldsymbol{W}_k(\tau)$ denotes the kth row vector of matrix $\boldsymbol{W}(t)$, and $\boldsymbol{f}_x(\tau)$ is defined by

$$\boldsymbol{f}_x(\tau) \triangleq \frac{\partial \boldsymbol{f}(\boldsymbol{X}, \boldsymbol{U}, t)}{\partial \boldsymbol{X}(t)} \cdot \frac{t_f - t_0}{2} \tag{3.37}$$

Then N Lagrange interpolating polynomials $L_i(\tau)(i = 1, 2, \ldots, N)$ are used to appropriate both sides of Eq. (3.36), by which the differential equation can be converted to a series of algeria equations at specified collocation points $\tau_i(i = 1, 2 \cdots N)$. In here, the Gauss-Lobatto type collocation is used, such as Legendre-Gauss-Lobatto (LGL), or Chebyshev-Gauss-Lobatto (CGL) series. Note that, in principle $\boldsymbol{W}_k(\tau)$ must satisfy Eq. (3.36) at all collocation points $\tau_i(i = 1, 2, \ldots, N)$. However, $\boldsymbol{W}_k$ is generally computed by integrating the matrix dynamics (3.36) backward from τ_N to τ_1 since the value at the final time $t_f(\tau_N)$ is known. This means $\boldsymbol{W}_k(\tau)|_{\tau=\tau_1}$ is the last integral step as well as the integration result. Therefore, $\boldsymbol{W}_k(\tau)|_{\tau=\tau_1}$ is not necessary to strictly satisfy the differential equation (3.36), and we just consider the $N-1$ collocation points $\tau_i(i = 1, 2, \ldots, N)$ for the according collocation equations. Consequently, the collocation equations are given in the compact form:

$$(\boldsymbol{D} \otimes \boldsymbol{I}_n) \cdot \Omega_k = -\boldsymbol{f} \cdot \Omega_k \tag{3.38}$$

where $\Omega_k = [\boldsymbol{W}_k(\tau_1), \boldsymbol{W}_k(\tau_2), \ldots \boldsymbol{W}_k(\tau_N)]^T$; $\boldsymbol{I}_n$ is an $n \times n$ identity matrix and $\boldsymbol{D} \otimes \boldsymbol{I}_n$ denotes the Kronecker product of $\boldsymbol{D}$ and $\boldsymbol{I}_n$; $\boldsymbol{D} \in \mathbb{R}^{(N-1)\times N}$ is known as the differential matrix. The matrix $\boldsymbol{D}$ and $\boldsymbol{f}$ are given by

$$\boldsymbol{D} = \begin{vmatrix} \dot{L}_1(\tau_2) & \dot{L}_2(\tau_2) & \cdots & \dot{L}_N(\tau_2) \\ \dot{L}_1(\tau_3) & \dot{L}_2(\tau_3) & \cdots & \dot{L}_N(\tau_3) \\ \vdots & \vdots & \ddots & \vdots \\ \dot{L}_1(\tau_N) & \dot{L}_2(\tau_N) & \cdots & \dot{L}_N(\tau_N) \end{vmatrix}, \boldsymbol{f} = \boldsymbol{diag} \begin{vmatrix} \boldsymbol{0}_{n(N-1)\times n} & & & \\ & \boldsymbol{f}_x^T(\tau_2) & & \\ & & \ddots & \\ & & & \boldsymbol{f}_x^T(\tau_N) \end{vmatrix} \tag{3.39}$$

Equation (3.38) can be further simplified by

$$\boldsymbol{A}\Omega_k = 0 \tag{3.40}$$

where $\boldsymbol{A} = \boldsymbol{f} + (\boldsymbol{D} \otimes \boldsymbol{I}_n) \in \boldsymbol{R}^{(N-1)n\times Nn}$. Equation (3.40) contains $(N-1)n$ linear equations and the same number of unknowns (that is $\boldsymbol{W}_k(\tau_i)(i = 1, 2, \ldots, N-1)$). Defining the unknown vector as $\boldsymbol{X}_k = \left[\boldsymbol{W}_k(\tau_1), \boldsymbol{W}_k(\tau_2), \ldots \boldsymbol{W}_k(\tau_{N-1})\right]^T$, it is easy to obtain $\Omega_{\mathbf{k}} = \left[\boldsymbol{X_k}^{\mathbf{T}} \ \boldsymbol{W_k}(\tau_{\mathbf{N}})\right]^{\mathbf{T}}$. Next $\boldsymbol{A}$ is rearranged as $\boldsymbol{A} = [\boldsymbol{A}_F, \boldsymbol{A}_N]$, where $\boldsymbol{A}_F$ and $\boldsymbol{A}_N$ denote the first $(N-1)n$ columns and the rest n columns of $\boldsymbol{A}$, respectively. Using these relations, the linear equations can be further expressed as

$$\boldsymbol{A}\Omega_k = [\boldsymbol{A}_F, \boldsymbol{A}_N] \cdot \begin{bmatrix} \boldsymbol{X}_k \\ \boldsymbol{W}_k(\tau_N)^T \end{bmatrix} = \boldsymbol{A}_F\boldsymbol{X}_k + \boldsymbol{A}_N\boldsymbol{W}_k(\tau_N)^T = 0 \tag{3.41}$$

Assuming that the matrix $\boldsymbol{A}_F$ is nonsingular, the $\boldsymbol{X}_k$ is eventually solved by

$$\boldsymbol{X}_k = -\boldsymbol{A}_F^{-1}\boldsymbol{A}_N \cdot \boldsymbol{W}_k(\tau_N)^T \tag{3.42}$$

The solution of $\boldsymbol{X}_k$ gives the value of kth row of matrix $\boldsymbol{W}(\tau)$ at the collocation points τ_k $(\tau_1, \tau_2, \ldots, \tau_{N-1})$. By repeating the above calculation procedure for each

row ($k = 1, 2, \cdots, p$), the matrix $\boldsymbol{W}(\tau)$ at all the collocation points $(\tau_1, \tau_2, \ldots, \tau_N)$ can be obtained.

Subsequently, the sensitivity matrix $\boldsymbol{B}_s(t)$ at collocation points can be calculated out according to Eq. (3.15):

$$\boldsymbol{B}_s(\tau_i) = \boldsymbol{W}(\tau_i) \cdot \left.\frac{\partial \boldsymbol{f}(\boldsymbol{X}, \boldsymbol{U}, t)}{\partial \boldsymbol{U}(t)}\right|_{t=\tau_i}, i = 1, 2, \ldots, N \tag{3.43}$$

Lastly, the principle of Gaussian quadrature [16] is applied to compute the sensitive matrix $\boldsymbol{A}_j$,$\boldsymbol{B}_\kappa$ and $\boldsymbol{R}_{ij}$:

$$\begin{aligned} &\boldsymbol{A}_j = \int_{t_0}^{t_f} \boldsymbol{B}_s(t) \cdot P_j(t) dt = \frac{t_f - t_0}{2} \sum_{i=1}^{N} \boldsymbol{B}_s(\tau_i) \cdot P_j(\tau_i) \cdot \eta_i \\ &j = 1, 2, \ldots, N_p \end{aligned} \tag{3.44}$$

$$\boldsymbol{B}_\kappa \triangleq -\int_{t_0}^{t_f} \left[\boldsymbol{W}(t) \cdot \dot{\boldsymbol{X}}(t)\right] = -\frac{t_f - t_0}{2} \sum_{i=1}^{N} \boldsymbol{W}(\tau_i) \cdot \dot{\boldsymbol{X}}(\tau_i) \cdot \eta_i \tag{3.45}$$

$$\begin{aligned} &\boldsymbol{R}_{ij} = \int_{t_0}^{t_f} \left[P_i(t) \boldsymbol{R}(t) P_j(t)\right] \cdot dt = \frac{t_f - t_0}{2} \sum_{k=1}^{N} P_i(\tau_k) \boldsymbol{R}(\tau_k) P_j(\tau_k) \cdot \eta_k \\ &i, j = 1, 2, \ldots, N_p \end{aligned} \tag{3.46}$$

where η_i is the weight coefficient of Gaussian quadrature corresponding to the collocation point τ_i. In this way, such sensitive matrix is obtained by a set of algebraic operation at very few collocation points.

Remark 3 In the calculation loop for each row of the matrix $\boldsymbol{W}$, the matrix $\boldsymbol{A}$ remains unchanged and just need to be computed once, since which only upon to the given collocated points $\tau_i (i = 1, 2, \ldots, N)$ and $\boldsymbol{f}_x(\tau_i)$. This feature effectively reduces the computational complexity.

Remark 4 Since the spectral sensitivity matrix is directly worked out by the Gaussian quadrature method in Eqs. (3.42)–(3.46) and avoids heavy computational consumption produced by the numerical integration of a series of matrix differential equations, the computational efficiency is improved significantly.

3.2.4 The Implementation Step of IGS-MPSP

The implementation procedure of this approach is provided in Algorithm 1. This method starts from the initial guess for spectral coefficients and terminal time. Then the final output errors and trajectory state are evaluated out. If the tolerance of the output errors is small enough, the desired control sequence is obtained. Otherwise, the corresponding sensitivity matrices are recalculated and the spectral coefficients as well as terminal time are updated. Then, the updated control history is generated, and the output errors are evaluated again. This iterative procedure is repeated until a specified criterion for the terminal output errors is met.

Algorithm 1: IGS-MPSP algorithm

Step 1: Initialize the initial guess $\boldsymbol{C}_j^0, t_f^0$, the stopping criterions δ_y, the number of spectral function N_p, the number of collocation points N

Step 2: For $k = 0, 1, 2, \ldots$

(2.1) Compute the control history $\boldsymbol{U}^k(t), t \in [t_0, t_f]$ by Eq.(3.17);
Integrate the system dynamic to obtain the trajectory state as well as the output error, $\|d\boldsymbol{Y}_N\|_\infty$;

(2.2) If $\|d\boldsymbol{Y}\|_\infty < \delta_y$
Output the current control history $\boldsymbol{U}^k(t)$ and break the iteration;
Otherwise continue the iteration.

(2.3) Compute the sensitive matrix $\boldsymbol{A}_j$, $\boldsymbol{B}_\kappa$ and $\boldsymbol{R}_{ij}$ according to Eq.(3.42) − Eq.(3.46);
Solve the linear equation (3.32) to obtain the updated solution;

(2.4) Update the terminal time by
$t_f^{l+1} = t_f^l + \Delta\kappa \cdot (t_f^l - t_0)$

(2.5) Obtain the update coefficient vector $\boldsymbol{C}_j^{k+1}$ from the updated solution $\tilde{\boldsymbol{X}}$;

Since the spectral coefficients have no physical meaning, it's not straightforward to assign an initial guess with appropriate values. Therefore, the least-squares algorithm is applied to obtain the initial guess of the spectral coefficients when an initial guess of control sequence is provided.

Denoting $\hat{\boldsymbol{C}}^0 = [\hat{\boldsymbol{C}}_1^0, \hat{\boldsymbol{C}}_2^0, \ldots, \hat{\boldsymbol{C}}_{N_p}^0]$ as the initial guess of the spectral coefficients, as per Eq. (3.17), the control vector represented by the guess $\hat{\boldsymbol{C}}^0$ at time step $t_k, k = 1, 2, \ldots, n$ can be written as

$$\hat{\boldsymbol{U}}_k^0 = \sum_{j=1}^{N_p} P_j(t_k)\hat{\boldsymbol{C}}_j^0, k = 1, 2, \ldots, n \tag{3.47}$$

In matrix form, Eq. (3.47) can be written as

$$\hat{\boldsymbol{U}}^0 = \hat{\boldsymbol{C}}^0 \boldsymbol{P} \tag{3.48}$$

where $\hat{\boldsymbol{U}}^0 = [\hat{\boldsymbol{U}}_1^0, \hat{\boldsymbol{U}}_2^0, \ldots, \hat{\boldsymbol{U}}_n^0]$, and

$$\boldsymbol{P} = \begin{vmatrix} P_1(t_1) & P_1(t_2) & \dots & P_1(t_n) \\ P_2(t_1) & P_2(t_2) & \dots & P_2(t_n) \\ \vdots & \vdots & \ddots & \vdots \\ P_{N_p}(t_1) & P_{N_p}(t_2) & \dots & P_{N_p}(t_n) \end{vmatrix} \tag{3.49}$$

If the initial control guess sequence $\boldsymbol{U}^0 = [\boldsymbol{U}_1^0, \boldsymbol{U}_2^0, \dots, \boldsymbol{U}_n^0]$ is given, the proper spectral function coefficients $\hat{\boldsymbol{C}}^0$ are to be found so as to minimize $|\hat{\boldsymbol{U}}^0 - \boldsymbol{U}^0|$. According to the principle of the least squares, the coefficients are estimated as

$$\hat{\boldsymbol{C}}^0 = \boldsymbol{U}^0 \boldsymbol{P}^T \cdot (\boldsymbol{P}\boldsymbol{P}^T)^{-1} \tag{3.50}$$

The initial guess of the spectral coefficients is obtained from Eq. (3.50).

3.3 The Ascent Predictive Guidance Under Thrust Drop Fault

In this subsection, the proposed method is employed to solve the ascent guidance problem of launch vehicle under thrust drop fault. The problem formula is firstly introduced, then the detailed procedure to address this problem is presented.

3.3.1 Problem Formulation

To be solved conveniently, a modified orbital inertial (MOI) coordinate system is firstly defined as follow. As shown in Fig. 3.1, P_F is the position of launch vehicle when the fault occurs; P_F' is the projection of P_F onto the injected orbital plane; P_f is the nominal injection point. Then, the origin of this modified orbital coordinate (MOC) is located at the center of the earth. The coordinate plane coincides with the injected orbital plane $OP_F'P_f$, in which the axis O_eY directing to the midpoint of the arc $P_F'P_f$ and the axis O_eX perpendicular to the O_eY. Lastly, the axis O_eZ is determined by the right-hand-thread rule.

Note that the MOI coordinate can be determined by the position of launch vehicle at the time that the fault occurs and the injected orbit information (the inclination i_f, longitude of ascending node Ω_f, and injection point for nominal trajectory). The relationship between the Modified orbital inertial (MOI) coordinate and the Earth-centered inertial (ECI) system is given by

$$\mathrm{X}^{MOI} = \mathrm{M}_{ECI}^{MOI}(\Omega_f, i_f, P'_F) \cdot \mathrm{X}^{ECI} \tag{3.51}$$

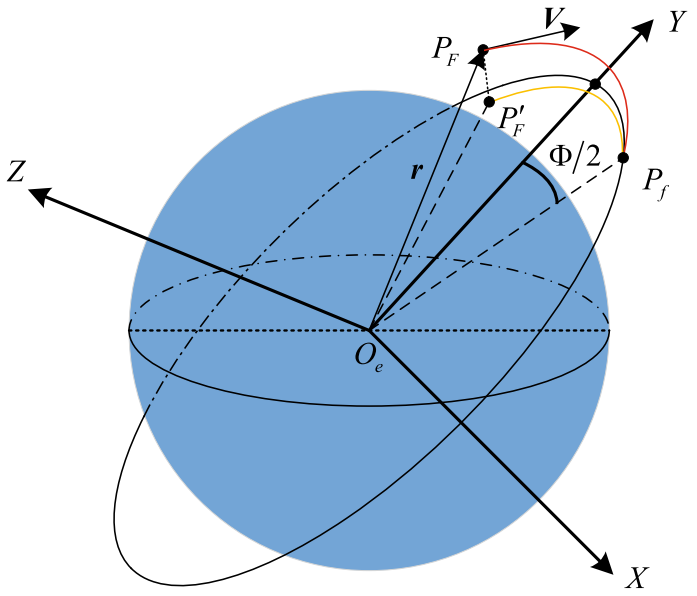


Fig. 3.1 Modified orbital inertial (MOI) coordinate system

where M_{ECI}^{MOI} denotes the transformation matrix from the ECI coordinate system to the MOI system.

It is considered the thrust drop fault occurs at the second stage of the launch vehicle. In this flight stage, the launch vehicle is assumed to fly out of the dense atmosphere and the aerodynamic forces can be ignored. Therefore, the three-dimensional point-mass dynamic equations of launch vehicles build in the (MOI) coordinate is given as follow:

$$\begin{cases} \dot{\boldsymbol{r}} = \boldsymbol{V} \\ \dot{\boldsymbol{V}} = T \cdot \boldsymbol{e}_T / m \ - \mu \boldsymbol{r} / r^3 \\ \dot{m} = -m_e \end{cases} \tag{3.52}$$

where $\boldsymbol{r} = [r_x, r_y, r_z]^T$ is the position vector in the MOI coordinate system; $\boldsymbol{V} = [V_x, V_y, V_z]^T$ is the inertial velocity vector; T is the thrust magnitude, which is considered to be constant; m is the mass of vehicle and m_e is the mass flow rate; $\boldsymbol{e}_T$ denotes the thrust direction vector, which is generally aligned with the body longitudinal axis of the vehicle and can be given by

$$\boldsymbol{e}_T = [\cos\varphi \cos\psi, \sin\varphi \cos\psi, -\sin\psi]^T \tag{3.53}$$

where φ and ψ are the pitch angle and yaw angle relative to the MOI coordinate system, respectively. The dynamic equations as presented in Eqs. (3.52) and (3.53) can be written as the compact form

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}) \tag{3.54}$$

where $\boldsymbol{x} = [r_x, r_y, r_z, V_x, V_y, V_z]^T$ is the state vector and $\boldsymbol{u} = [\varphi, \psi]^T$ is the control vector of the system.

Remark 5 Since the flight path angle and angle of attack of the launch vehicle is generally small in the second stage, the defined MOI and the according dynamic equation will ensure the pitch angle of the vehicle remain a small value. This can effectively reduce the nonlinearity of the controls as presented in Eq. (3.53) and improve the convergence of the algorithm. Moreover, such definition can simplify the terminal constraint to be introduced later.

3.3.2 Terminal Constraints

It is assumed that the thrust fault of the launch vehicle takes place at the initial time t_0, and the corresponding states are given by:

$$\boldsymbol{X}(t_0) = \boldsymbol{X}_0 \tag{3.55}$$

The final orbital injection time t_f is constrained by

$$t_f \leq t_{f,\max} \tag{3.56}$$

where $t_{f,\max}$ is the maximum burn time of the vehicle, which is determined by the remaining fuel and mass flow rate.

The terminal constraints of ascent guidance are determined by the orbital insertion conditions, which are generally provided by the semi-major axis a_f, eccentricity e_f, orbital inclination i_f, and longitude of ascending node Ω_f. In here, we consider to entry into a circular orbit ($e_f = 0$). Then, the first two conditions can be equivalently described by

$$\left\| \boldsymbol{r}(t_f) \right\| = r_f^* \tag{3.57}$$

$$\left\| \boldsymbol{V}(t_f) \right\| = V_f^* \tag{3.58}$$

$$\boldsymbol{r}^T(t_f)\boldsymbol{V}(t_f) = 0 \tag{3.59}$$

Moreover, in the modified orbital inertial (MOI) coordinate system, the final two orbital insertion conditions i_f and Ω_f are equivalent to make the final position vector component $r_z(t_f)$ and velocity vector component $V_z(t_f)$ to be zero. Therefore, the terminal constraints of this ascent guidance problem are defined by

$$h(\boldsymbol{x}(t_f)) = \begin{bmatrix} \|\boldsymbol{r}(t_f)\| = r_f^* \\ \|\boldsymbol{V}(t_f)\| = V_f^* \\ \boldsymbol{r}^T(t_f)\boldsymbol{V}(t_f) \\ r_z(t_f) \\ V_z(t_f) \end{bmatrix} = \mathbf{0} \tag{3.60}$$

Thus, the ascent guidance problem can be organized by

$\boldsymbol{P}^0$: find t_f, $\mathrm{u}(t)$, $t \in [t_0, t_f]$

subject to:

$$\mathrm{x}(t) = \mathrm{f}(\mathrm{x}(t),\mathrm{u}(t),t) \tag{3.61}$$

$$\mathrm{x}(t_0) = \mathrm{x}_0 \tag{3.62}$$

$$h(\mathrm{x}(t_f)) = 0 \tag{3.63}$$

$$t_f \leq t_{f,\max} \tag{3.64}$$

3.3.3 Solved by the IGS-MPSP

As introduced earlier, the proposed algorithm is able to solve the free-final time guidance problem. However, this algorithm cannot directly handle the inequality constraint as given in Eq. (3.64). Therefore, a numerical trick is additionally conducted to address this constraint.

First, the proposed method is employed to solve the problem $\boldsymbol{P}^0$ in which the constraint (3.64) is omitted. Then the obtained terminal time t_f is checked. If this value is smaller than the maximum $t_{f,\max}$, it means the solution is feasible and the obtained control history can be used as the renewed commands of the launch vehicle. Otherwise, it implies the solution is infeasible for this problem. That is, the launch vehicle cannot directly entry into the required orbit. In this situation, a new guidance strategy is needed, such as entering into a new parking orbit or transfer orbit. This case is beyond the scope of this work.

The detailed implementation steps are summarized as follow. As presented in Fig. 3.2, the guidance strategy is triggered by a fault detection. Then the current state t_0, x_0 is employed as the initial state of the proposed algorithm, and the terminal time and control of nominal trajectory is used as the initial guess. Obviously, these controls and terminal time guess cannot steer the launch vehicle to well meet the required terminal states in the presence of thrust drop fault. Thus, the proposed algorithm is iteratively conducted to obtain the updated terminal time and control history. If the updated terminal time is smaller than the maximum value, then the corresponding control history is directly used as the new guidance for the launch vehicle. Otherwise, it implies that the launch vehicle cannot directly inject into the original orbit.

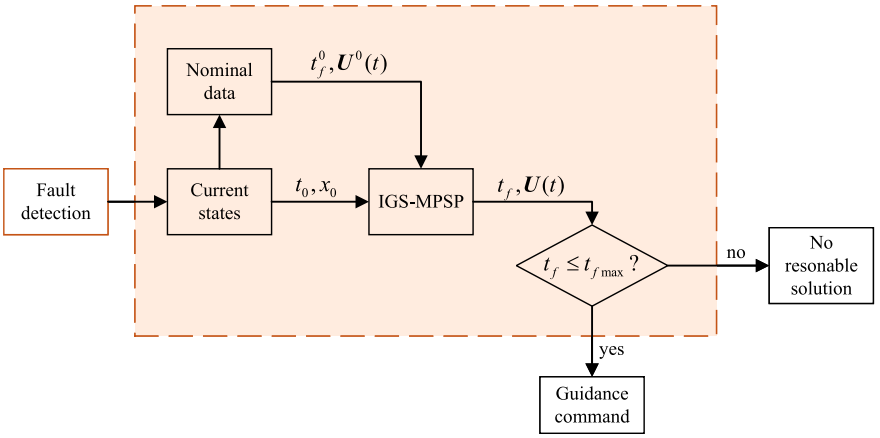


Fig. 3.2 Ascent guidance strategy for launch vehicle under thrust drop fault

3.4 Numerical Results

In this section, the numeric simulation is carried out to demonstrate the performance of the proposed method in term of accuracy and computational efficiency. It is considered that the fault of thrust drop occurs at the second stage flight phase of a launch vehicle. The nominal parameters of the launch vehicle are illustrated in Table 3.1. Both the initial conditions (at the fault occurring time) and target orbit's parameters are given in Table 3.2.

The proposed method is employed to solve such guidance problem when the thrust of the launch vehicle drops to 70% and 80% of the nominal value, respectively. In the algorithm implementation, the six order Legendre polynomials are used as the spectral function of control, and the Legendre-Gauss-Latto (LGL) points are selected as the collocation points in computing the spectral sensitivity matrix. In addition, the number of LGL nodes is taken as 15.

Additionally, a SOCP based method [11] is conducted in here as the comparison of the proposed method. This algorithm takes the minimum fuel consumption as the optimization object, and the classical Euler method is used to discrete the problems. The number of discretization nodes is set to be 50, which is determined by comprehensively considering the solving accuracy and efficiency.

Table 3.1 Nominal parameters of the launch vehicle

Parameter	Value	Unit
Initial mass m_0	27.769	ton
Dry mass m_{dry}	17	ton
Nominal thrust T	98	KN
Dry mass m_e	22.138	kg/s

Table 3.2 Initial conditions and target orbit

Parameter	Value	Unit
Initial position vector $\boldsymbol{r}_0$	[368502, 6508822, 7201]	m
Initial velocity vector $\boldsymbol{v}_0$	[6148.49, 99.79, −18.13]	m/s
Target orbit semi-major axis a_f	6578145	m
Target orbit eccentricity e_f	0	
Target orbit inclination i_f	0.5068	rad
Target orbit ascending node Ω_f	6.2368	rad

In simulations, the terminal time and control history of nominal trajectory are used as the initial guess of the proposed method and SOCP based method. Moreover, all numerical simulations are implemented in the MATLAB 2021a environment on a personal desktop (Intel i7-8750H, 3.2 GHz). The CVX [17] optimization toolbox with SDPT3 4.0 [18] is employed as the solver of the SOCP based method.

3.4.1 The Results by the Proposed Method

The proposed method reaches the required tolerance of terminal conditions by 8 iterations. Figure 3.3 depicts the altitude profiles of the trajectories with the thrust of 70% and 80% nominal value. Additionally, the nominal trajectory and the trajec-

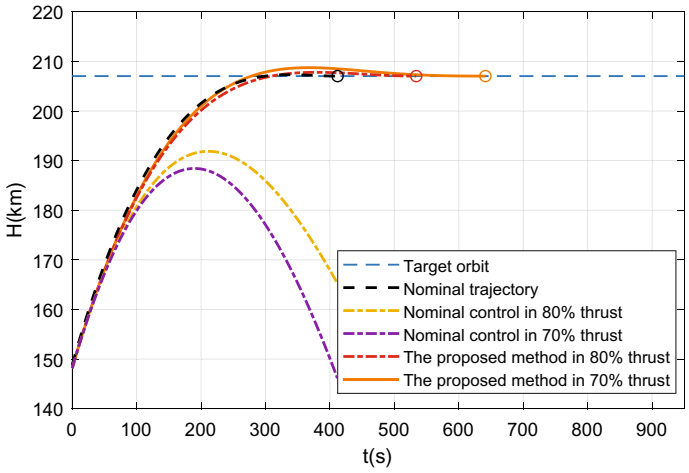
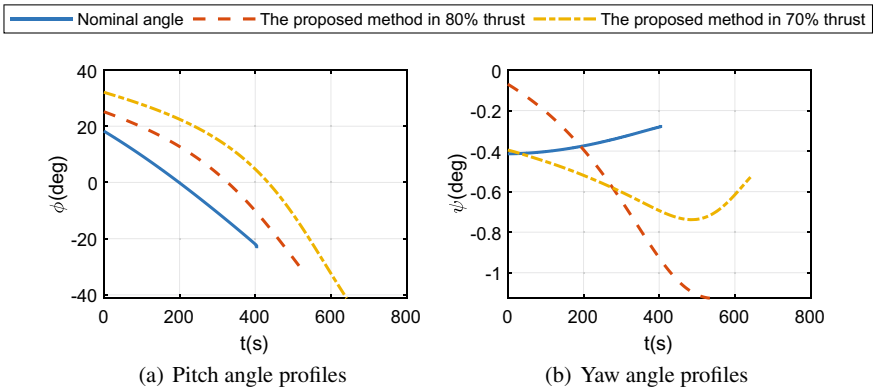


Fig. 3.3 Altitude profiles for nominal trajectory and the trajectories obtained by IGS-MPSP



(a) Pitch angle profiles (b) Yaw angle profiles

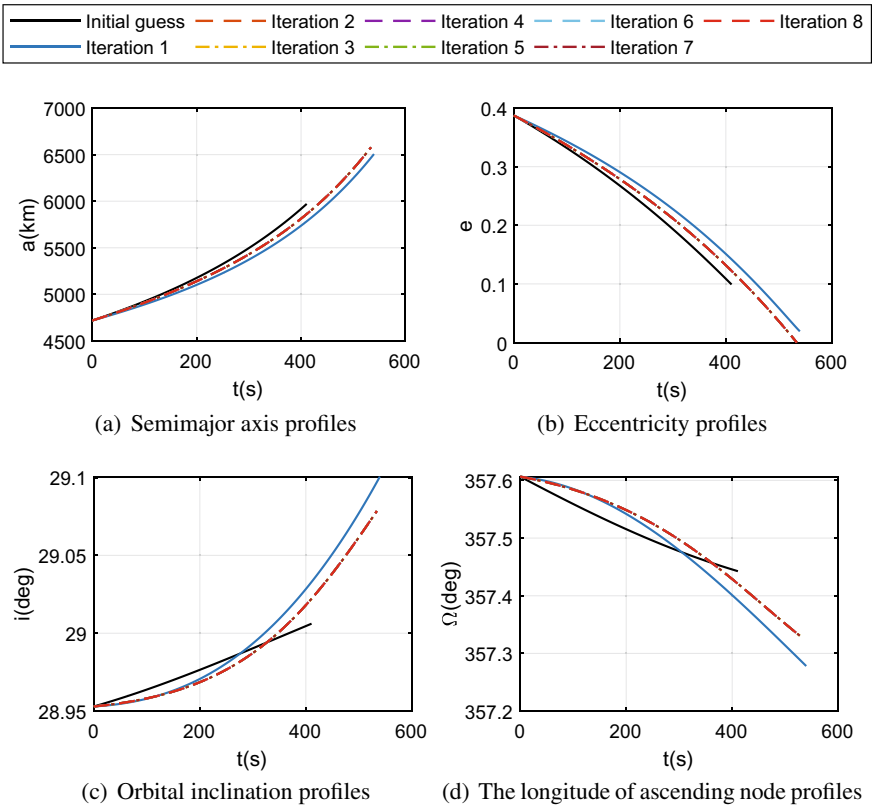
Fig. 3.4 Control profiles for the nominal trajectory

tories obtained by the nominal control in the case of thrust drop are also provided in Fig. 3.3. It can be seen that with the nominal control, the launch vehicle fails to enter into the target orbit under the thrust drop. The proposed method succeeds in regenerating the updated guidance commands (shown in Fig. 3.4) to steer the vehicle into the original target orbit. Specially, the trajectory states for each iteration of the proposed algorithm when the thrust is 80% of nominal value is presented in Fig. 3.5. It clearly reflects that the proposed method reaches the required orbit injection parameters by a few iterations, even if the relatively poor control and terminal time guess (nominal trajectory) are given. At the same time, the orbit injection time is considerably increased as compared to the nominal value, but still smaller than the maximum allowable value. These results demonstrate the effectiveness of the proposed method, which is able to re-plan the ascent trajectory under the thrust drop, and search the appropriate orbit injection time when a relatively accurate guess cannot not be given out.

3.4.2 Comparison with SOCP Method

Furthermore, the comparison between the proposed method and SOCP based method are provided in Table 3.3, Figs. 3.6 and 3.7.

The Table 3.3 illustrates the terminal mass and terminal time achieved by the proposed algorithm and SOCP based method, respectively. It can be noted that the results obtained by the proposed algorithm are very close to that produced by SOCP method. The deviations are less than 0.025%. This means the proposed method achieves a near optimality for the fuel consumption compared to the SOCP based method. Moreover, the control histories and trajectories for the proposed method and SOCP are depicted in Figs. 3.6 and 3.7. As it can be seen, the control profiles as well as the trajectories by the proposed method are similar to those of SOCP, no matter in



(a) Semimajor axis profiles (b) Eccentricity profiles

(c) Orbital inclination profiles (d) The longitude of ascending node profiles

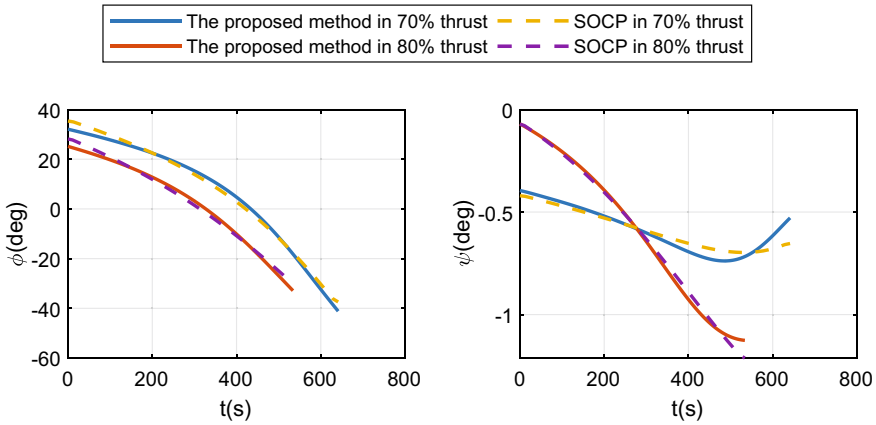
Fig. 3.5 The trajectory for each iteration in the 80% of nominal thrust

Table 3.3 The results of final time and final mass

	80% of nominal thrust		70% of nominal thrust	
	t_f (s)	m_f (kg)	t_f (s)	m_f (kg)
The proposed method	533.9370	18312.7483	641.2647	17831.5629
SOCP	533.6772	18317.3497	641.2007	17832.5554

the case of 70% of the nominal thrust or 80% of the nominal thrust. This demonstrates the proposed method produces approximate effect of SOCP based method to solve the ascent guidance problem with the thrust drop fault.

Lastly, the computational efficiency of the proposed method and SOCP method are comparatively investigated by conducting the same simulation case. Figure 3.8 and Table 3.4 present the CPU time consumed by two methods, in which the time elapses for one iteration and the total are shown. It can be seen that, the CPU time consumed by the proposed method is almost one-sixtieth or one-seventieth of that



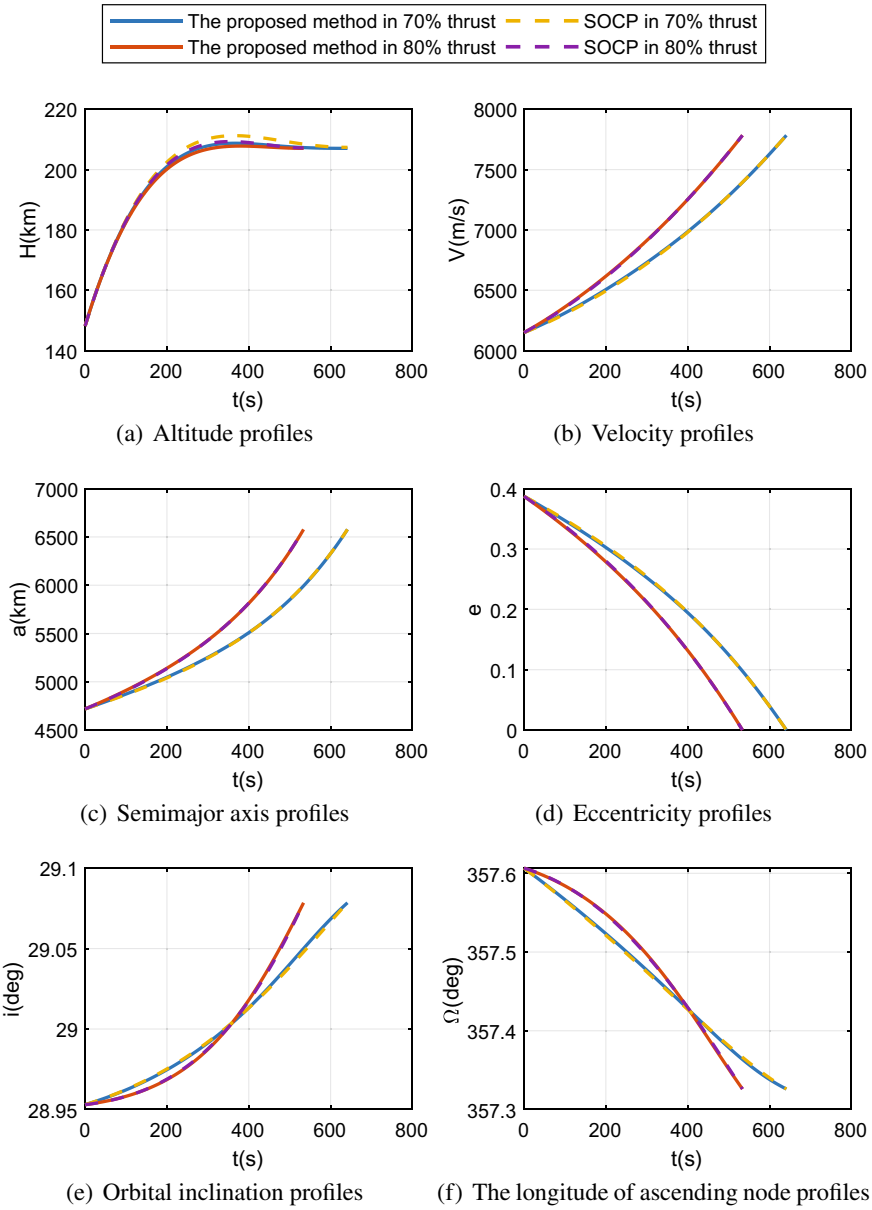
(a) Pitch angle profiles (b) Yaw angle profiles

Fig. 3.6 The Control profile for IGS-MPSP and SOCP method

by the SOCP method for one iteration and the total value. This result clearly demonstrates the superiority of the proposed method in the computational speed. Such highly computational efficiency is achieved by a series of careful design such as the spectral representation of control, sensitive matrix computation conducted by collocation method. Hence, the proposed method owns the great potential for online application.

3.5 Conclusion

In this chapter, a predictive ascent guidance based on the IGS-MPSP for the thrust drop fault of the launch vehicle is presented. Firstly, an IGS-MPSP method is derived. Compared with the original GS-MPSP method, this approach introduces a scale factor for the time internal as the additional variable to adjust the terminal time. Then, a new sensitive relation for the final time is established. Since the accuracy of the sensitive relation is improved, the approach owns the better performance to search the appropriate final time in the presence of the poor initial guess. Hence, it is more suitable for the ascent guidance problem. Secondly, the application of the proposed method for the ascent guidance problem under the thrust drop fault is detailed introduced. The numerical simulation for a typical case and comparison with the SOCP based method are carried out. The results indicate the effectiveness of the proposed method, which generate approximate results with the SOCP method but with considerably higher computing efficiency.



(a) Altitude profiles

(b) Velocity profiles

(c) Semimajor axis profiles

(d) Eccentricity profiles

(e) Orbital inclination profiles

(f) The longitude of ascending node profiles

Fig. 3.7 Trajectories by various methods

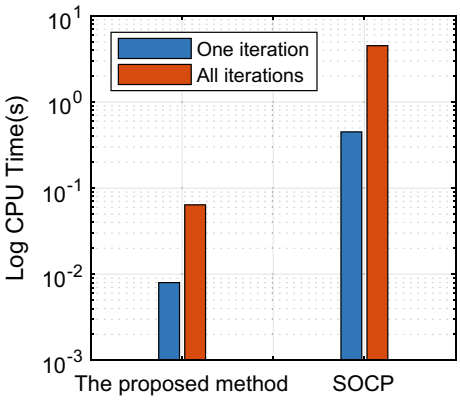


Fig. 3.8 CPU time consumed by various methods

Table 3.4 The CPU time consumed by various methods

Method	CPU time for one iteration (s)	Iterations number	Total CPU time(s)
SOCP	0.451	10	4.512
IGS-MPSP	0.008	8	0.064

References

1. I.E. Smith, General formulation of the iterative guidance mode, in *NASA TM X-53414* (1966)
2. R. Jaggers, An explicit solution to the exoatmospheric powered flight guidance and trajectory optimization problem for rocket propelled vehicles, in *Guidance and Control Conference* (1977), p. 1051
3. E.-J. Song, S. Cho, W.-R. Roh, A comparison of iterative explicit guidance algorithms for space launch vehicles. Adv. Space Res. **55**(1), 463–476 (2015)
4. K. Mall, M.J. Grant, E. Taheri, Uniform trigonometrization method for optimal control problems with control and state constraints. J. Spacecr. Rocket. **57**(5), 995–1007 (2020)
5. M.H. Gräßlin, J. Telaar, U.M. Schöttle, Ascent and reentry guidance concept based on NLP-methods. Acta Astronaut. **55**(3–9), 461–471 (2004)
6. P. Lu, Introducing computational guidance and control (2017)
7. X. Liu, L. Ping, B. Pan, Survey of convex optimization for aerospace applications. Astrodynamics **1**(1), 23–40 (2017)
8. X. Liu, Z. Shen, L. Ping, Entry trajectory optimization by second-order cone programming. J. Guid. Control Dyn. **39**(2), 227–241 (2016)
9. M. Szmuk, T.P. Reynolds, B. Açıkmeşe, Successive convexification for real-time six-degree-of-freedom powered descent guidance with state-triggered constraints. J. Guid. Control Dyn. **43**(8), 1399–1413 (2020)
10. X. Cheng, H. Li, R. Zhang, Efficient ascent trajectory optimization using convex models based on the Newton-Kantorovich/Pseudospectral approach. Aerosp. Sci. Technol. **66**, 140–151 (2017)
11. Y. Li, B. Pang, C. Wei, N. Cui, Y. Liu, Online trajectory optimization for power system fault of launch vehicles via convex programming. Aerosp. Sci. Technol. **98**, 105682 (2020)
12. Z. Song, C. Wang, Q. Gong, Joint dynamic optimization of the target orbit and flight trajectory of a launch vehicle based on state-triggered indices. Acta Astronaut. **174**, 82–93 (2020)

13. Z. Hao, R. Zhang, Onboard real-time generation of launch vehicle abort orbits. J. Guid. Control Dyn. **44**(8), 1541–1549 (2021)
14. R. Padhi, M. Kothari, Model predictive static programming: a computationally efficient technique for suboptimal control design. Int. J. Innov. Comput. Inf. Control **5**(2), 399–411 (2009)
15. C. Zhou, X. Yan, S. Tang, Generalized quasi-spectral model predictive static programming method using gaussian quadrature collocation. Aerosp. Sci. Technol. **106**, 106134 (2020)
16. A. Maity, H.B. Oza, R. Padhi, Generalized model predictive static programming and angle-constrained guidance of air-to-ground missiles. J. Guid. Control Dyn. **37**(6), 1897–1913 (2014)
17. M. Grant, S. Boyd, CVX: Matlab software for disciplined convex programming, version 2.1. (2016)
18. K.-C. Toh, M.J. Todd, R.H. Tütüncü, On the implementation and usage of SDPT3–a Matlab software package for semidefinite-quadratic-linear programming, version 4.0, in *Handbook on Semidefinite, Conic and Polynomial Optimization* (Springer, Berlin, 2012), pp. 715–754

Chapter 4
Birkhoff Pseudospectral Method and Convex Programming for Trajectory Optimization

Dangjun Zhao, Zhiwei Zhang, and Mingzhen Gui

4.1 Introduction

Trajectory optimization, an optimal control problem (OCP) in essence, is an important issue in many engineering applications including space missions, such as orbit insertion of launchers, orbit rescue, formation flying, etc. There exist two kinds of solving methods for OCP, i.e., indirect and direct methods. For some simple OCPs, using the indirect methods can result in analytic solutions, which are not easy to be obtained for complicated systems. Direct methods transcribe an OCPs into a finite-dimensional nonlinear programming (NLP) problem via discretizing the states and the controls at a set of mesh points, which should be carefully designed via compromising the computational burden and the solution accuracy. In general, the larger number of mesh points, the more accurate solution as well as the larger computational cost including CPU time and memory [1]. There are many numerical methods have been developed for the transcription of OCPs, and the most common method is by using Pseudospectral (PS) collocation scheme [2], which is an optimal choice of mesh points in the reason of well-established rules of approximation theory [3]. Actually, there have several mature optimal control toolkits based PS methods, such as DIDO [4], GPOPS [5]. The resulting NLP problem can be solved by the well-known algorithm packages, such as IPOPT [6] or SNOPT [7]. However, these algorithms cannot obtain a solution in polynomial-time, and the resulting solution is locally

D. Zhao (✉) · Z. Zhang · M. Gui
School of Automation in Central South University, Changsha, People's Republic of China
e-mail: zhao_dj@csu.edu.cn

Z. Zhang
e-mail: zw_zhang20@csu.edu.cn

M. Gui
e-mail: guimingzhen@csu.edu.cn

Z. Song et al. (eds.), *Autonomous Trajectory Planning and Guidance Control for Launch Vehicles*, Springer Series in Astrophysics and Cosmology,
https://doi.org/10.1007/978-981-99-0613-0_4

optimal. Moreover, a good initial guess solution should be provided for complicated problems.

Convex optimization method provides a polynomial-time complexity for solving convex programming problem, and has been wildly used for solving OCP, especially for complicated space missions [8–11]. In convex optimization framework, the original OCP is converted into a second-order conic programming (SOCP) problem by using some convexification techniques and discretization on uniformly distributed points. To achieve sufficiently accurate solution, one way is to choose a large number of discretized points, which leads to extra computational burden, and an-other way is employing PS method [12–14]. In [15], the various discretization methods, including the zeroth order hold (ZOH), the first order hold (FOH), the classical Runge–Kutta fourth order integration method, and global PS methods (Chebyshev-Gauss-Lobatto, CGL, Legendre-Gauss-Radau, LGR) are considered in the convex optimization framework for solving the planetary powered landing trajectory optimization problem, and the authors con-clude that PS methods can provide more consistent solutions, which are less sensitive to the number of mesh points.

It is to note that only the cases of mesh points are considered in [15], while in some certain applications, especially aerospace applications, to achieve more accuracy, the more mesh points are required [16]. However, purely increasing will lead to ill-conditioned NLP problem [17], which is hard to solve or cannot to be solved. In order to overcome such ill-conditioned phenomenon, some well-conditioned PS methods for ordinary differential equations (ODEs) have been proposed in [17, 18], in which, Birkhoff PS method stemmed from Birkhoff interpolation [19] is introduced to solve higher-order ODEs. In [16, 19], the first-order Chebyshev Birkhoff PS (CBPS) method is proposed to transcribe general OCP into NLP, and the advantages of CBPS over other PS methods, especially for large mesh grids, is demonstrated by its application in solving an orbital transfer problem.

In this chapter, we apply PS methods using Bikrhoff polynomials to the convex optimization framework for OCPs, particularly, the first order and second-order Birkhoff PS (BPS) methods with LGL and CGL collocation schemes are applied to transcribe a class of cascaded second-order systems, which may be convex or not. The first-order BPS method in convex optimization framework is similar to that in [16], hence we focus on the PS transcription for convexified OCP by using second-order Birkhoff polynomials. The main contributions of this paper lie in two aspects: (1) the unified matrix formulation of OCP by using BPS method within the convex optimization framework is proposed and validated; and (2) the computational performances and solution accuracies resulted from various PS schemes are extensively exploited thereby a useful conclusion that, using BPS method renders the remarkable drops in the condition number for the generated programming problem therefore lowering the computational cost.

The remainder of this chapter is organized as follows. The preliminaries of convex programming and PS method for general convexified optimal control problems are presented in Sect. 4.2, then the proposed well-conditioned PS method via Birkhoff polynomials within the convex optimization framework is detailed in Sect. 4.3. The

demonstrated examples of a simple cart problem as well as a rescue orbit searching problem for validating the effective-ness and efficiency of the proposed method are presented in Sect. 4.4 followed by the conclusions in Sect. 4.5.

4.2 Preliminaries of Convex Programming and PS Method for Optimal Control

In general, the free-final-time OCP can be converted into a fixed-final-time problem by normalizing the time into the domain of [0, 1], consequently, only the fixed-final-time problem is considered in this chapter. A typical OCP is given as the following problem G.

$$\text{Problem} G \min J = \int_{t_0}^{t_f} L(\mathbf{x}, \mathbf{u}) \mathrm{d}t \tag{4.1}$$

$$\text{s.t. } \dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{x}}_1 \\ \dot{\mathbf{x}}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{x}_2 \\ \mathbf{f}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{u}) \end{bmatrix} \triangleq \mathbf{F}(\mathbf{x}, \mathbf{u}) \tag{4.2}$$

$$\mathbf{P}(\mathbf{x}, \mathbf{u}) \leq \mathbf{0} \tag{4.3}$$

$$\mathbf{x}(t_0) = \mathbf{x}_0, \boldsymbol{\psi}(\mathbf{x}(t_f)) = \mathbf{0} \tag{4.4}$$

where the prescribed time instants $t_0 < t_f < \infty$, the state vector $\mathbf{x} = [\mathbf{x}_1^T, \mathbf{x}_2^T] \in \mathbb{R}^{2n}$, the control vector $\mathbf{u} \in \mathbb{R}^m$, $\mathbf{x}(\dot{\mathbf{t}}_0)$ and $\mathbf{x}(\dot{\mathbf{t}}_f)$ are specified with the boundary constraints (4.4), the nonlinear path constraints in (4.3) may be various form in terms of $\mathbf{x}$ and $\mathbf{u}$. The problem G is a kind of Lagrange problem, in which, the performance index (4.1) is the integral of Lagrange function [9] $L(\mathbf{x}, \mathbf{u})$ on time domain $[t_0, t_f]$. According to the optimal control theory [20], other types including Mayer, Bollza, and Quadratic problems can be converted into a Lagrange problem. The nonlinear dynamics in (4.2) is a typical cascaded second-order system, which is widely used to formulate the linear motion of a spacecraft, such as the Mars landing [8], launch ascent [21], etc.

4.2.1 Convex Programming Method for OCP

To solve the general problem **G** by convex programming method, the performance index (4.1) and the constraints (4.2)∼(4.4) should be the form of linear or second-order cone (SOC), thus the convexification techniques are required for converting the nonlinear or concave constraints to the corresponding convex formulation. The most important issue lies in the conversion process is to guarantee the solution of the converted problem is still that of the original problem.

For some typical problems with the form of G, which are relatively simple, one can introduce some auxiliary state and (or) control variables as well as some unique relaxation to obtain the convex versions of the original problems. Such procedure is called a lossless convexification technique, which can be found in [8, 9, 22, 23]. For general OCPs, successive convex programming (SCP) method is proposed for handling more general nonconvex constraints [10, 24–26]. The main idea of SCP is to solve a nonlinear and nonconvex problems via iteratively solving a series of local convex approximate problem using linearization on the solution of last iteration. Denote the solution of the kth iteration to the original problem G as $\left\{\mathbf{x}^k, \mathbf{u}^k\right\}$, thereby the following sequential convex version

$$\begin{aligned}
\text{Problem} SC: \min J = \int_{t_0}^{t_f} \left[L(\mathbf{z}^k) + \left[L_z(\mathbf{z}^k)\right]^T (\mathbf{z} - \mathbf{z}^k) \right] dt \\
\text{with} \quad \mathbf{z} = [\mathbf{x}^T, \mathbf{u}^T]^T \\
s.t. \quad \dot{\mathbf{x}}_1 = \mathbf{x_2} \\
\dot{\mathbf{x}}_2 = \mathbf{a}_1^k \mathbf{x}_1 + \mathbf{a}_2^k \mathbf{x}_2 + \mathbf{g}^k \mathbf{u} + \mathbf{c}^k \\
\mathbf{P}(\mathbf{z}^k) + \mathbf{P}_x^k (\mathbf{z} - \mathbf{z}^k) \le \mathbf{0} \\
\mathbf{x}(t_0) = \mathbf{x_0} \\
\boldsymbol{\psi}(\mathbf{x}_{t_f}^k) + \boldsymbol{\psi}_{x_{tf}}^k (\mathbf{x}(t_f) - \mathbf{x}_{t_f}^k) = 0 \\
|\mathbf{z} - \mathbf{z}^\mathbf{k}| \le \varepsilon
\end{aligned} \tag{4.5}$$

which is a first-order approximation of the problem G, and the symbols denote the following:

(1) The performance index $L(\mathbf{x}, \mathbf{u})$ is approximated by its Taylor first order expansion, and $L_z(\mathbf{z}^k) = \partial L / \partial \mathbf{z}|_{\mathbf{z}=\mathbf{z}^k}$. Similarly the path constraints (4.3) and the boundary constraints (4.4) are first-orderly approximated, and

$$\mathbf{P}_z^k = \partial \mathbf{P} / \partial \mathbf{z}|_{\mathbf{z}=\mathbf{z}^\mathbf{k}}, \ \psi_{x_{tf}}^k = \partial \psi / \partial \mathbf{x}_{\mathbf{t_f}}|_{\mathbf{x}_{\mathbf{t_f}}=\mathbf{x}_{\mathbf{t_f}}^\mathbf{k}}.$$

(2) The dynamic constraints (4.2) are linearized on the trajectory $\left\{\mathbf{x}^k, \mathbf{u}^k\right\}$, and the linearizing coefficients are given by

$$\begin{aligned}
&\mathbf{a}_1^k = \partial \mathbf{f} / \partial \mathbf{x}_1|_{\mathbf{x}=\mathbf{x}^k, \mathbf{u}=\mathbf{u}^k}, \ \mathbf{a}_2^k = \partial \mathbf{f} / \partial \mathbf{x}_2|_{\mathbf{x}=\mathbf{x}^k, \mathbf{u}=\mathbf{u}^k} \\
&\mathbf{g}^k = \partial \mathbf{f} / \partial \mathbf{u}|_{\mathbf{x}=\mathbf{x}^k, \mathbf{u}=\mathbf{u}^k} \\
&\mathbf{c}^k = \mathbf{f}(\mathbf{x}^k, \mathbf{u}^k) - \mathbf{a}_1^k \mathbf{x}_1^k - \mathbf{a}_2^k \mathbf{x}_1^k - \mathbf{g}^k \mathbf{u}^k
\end{aligned} \tag{4.6}$$

(3) The trust region constraints $|\mathbf{z} - \mathbf{z}^\mathbf{k}| \le \varepsilon$ are introduced in problem SC to ensure its solution is still that of original problem G, and the reason lies in that the first-order approximation is only reasonable within a small neighborhood around $\{\mathbf{x}^\mathbf{k}, \mathbf{u}^\mathbf{k}\}$.

For convex problem SC, one can use some appropriate discretizing methods, and let the states and controls on the discrete points be decision variables henceforth a convex programming problem (or an SOCP problem), which can be efficiently solved by the classical conjugate gradient method [13] or the popular primal-dual interior method, such as ECOS [27], Mosek [28].

Remark 1 Unlike the second-order approximations used in sequential quadratic programming (SQP) [29], the above procedure in the manner of first-order guarantees the approximations are convex and suitable be solved by efficient convex programming algorithms. In [10], the authors have presented a thoroughly theoretical analysis, and pointed out that, by introducing appropriate trust region constraints, the iterative solutions of the series problem SC will converge to that of the original problem G.

In this chapter, we concentrate on the PS method for problem SC, which contains various constraints and tedious bookkeeping, and may cause distractions. In fact, the emphasis of discretizing procedure is about the dynamic constraints, hence a distilled convex optimal control problem is given by

$$\text{Variables}: \mathbf{x} = [\mathbf{x}_1^T, \mathbf{x}_2^T]^T \in {}^{2n}, \mathbf{u} \in {}^{m}, \tau \in [-1, 1] \tag{4.7}$$

$$\text{Problem} C: \min J[\mathbf{x}(\cdot), \mathbf{u}(\cdot)] = E(\mathbf{x}(-1), \mathbf{x}(1)) \tag{4.8}$$

$$\begin{aligned} s.t. \quad \dot{\mathbf{x}}_1(\tau) &= s\mathbf{x}_2(\tau) \\ \dot{\mathbf{x}}_2(\tau) &= s\left[\mathbf{a}_1\mathbf{x}_1(\tau) + \mathbf{a}_2\mathbf{x}_2(\tau) + \mathbf{g}\mathbf{u}(\tau)\right] \end{aligned} \tag{4.9}$$

$$\mathbf{e}(\mathbf{x}(-1), \mathbf{x}(1)) = \mathbf{0} \tag{4.10}$$

where the index function $E(\cdot)$ and the boundary constraints $\mathbf{e}(\cdot)$ are reasonably supposed to be convex; the time mapping scale factor $s = dt/d\tau = (t_f - t_0)/2$, which is resulted from that, the time dependent variable $t \in [t_0, t_f]$ is converted into the PS time $\tau \in [-1, 1]$ by using the mapping $t = (t_f - t_0)\tau/2 + (t_f + t_0)/2$.

In problem C, the dynamic constraint (4.9) and boundary constraint (4.9) are considered, while the path constraints $\mathbf{P}(\cdot)$ are not considered for the convenience of subsequent discussions. It is a reasonable simplification, since the PS results generated for problem C can be easily recovered to problem SC as shown in [16, 30].

4.2.2 PS Method for Convex Optimal Control Problem

In PS optimal control techniques, any time signal $f(\cdot)$ can be expressed as a linear combination of a series of orthogonally Lagrange polynomials

$$f(\tau) = \sum_{i=0}^{N} f_i \mathcal{L}_i(\tau) \tag{4.11}$$

where the spectral coefficient $f_i = f(\tau_i); \pi^N := [\tau_0, \tau_1, \ldots, \tau_N]$ is the discrete point set such that $-1 \leq \tau_0 \leq \tau_1 \leq \cdots \leq \tau_{N-1} \leq \tau_N \leq 1$ and $\mathcal{L}_i \in P_N$ are the Lagrange orthogonally interpolating polynomials at the freedom of degree N, which satisfy the Kronecker relationship

$$\mathcal{L}_j(\tau_i) = \delta_{ij} \tag{4.12}$$

Differentiating (4.11) and evaluating $f(\tau)$ on the discrete points yields:

$$\dot{f}(\tau_i) = \sum_{j=0}^{N} f_j \dot{\mathcal{L}}_j(\tau_i) \triangleq \dot{\mathbf{f}} = \mathcal{D}\mathbf{f} \tag{4.13}$$

where the PS differentiation matrix (PSDM) $\mathcal{D} \in {}^{(N+1)\times(N+1)}$ is given by $d_{ij} = \dot{\mathcal{L}}_j(\tau_i)$ for $0 \leq i, j \leq N$; and $\mathbf{f} = [f_0, \ldots, f_N]^T$. It's to note that the higher-order differentiation matrix is given by [31]

$$\mathcal{D}^{(k)} = \mathcal{D}\mathcal{D}\ldots\mathcal{D} = \mathcal{D}^k \tag{4.14}$$

Particularly for second differential of $f(\tau)$, we have

$$\ddot{\mathbf{f}} = \mathcal{D}^{(2)}\mathbf{f} \tag{4.15}$$

The best choices for $\mathcal{L}_i$ are the Legendre and Chebyshev polynomials [30] henceforth the Legendre and Chebyshev PS methods. When mesh points defined by $\tau_i \in (-1, 1)$, the mesh points are Gauss points, while for Gauss-Labotto points, $\tau_i \in [-1, 1]$ contain the two boundary points, and for Gauss-Radau points, $\tau_i \in (-1, 1]$ or $\tau_i \in [-1, 1)$. Since the boundary points must be constrained in many OCPs, Legend-Gauss-Labotto (LGL) and Chebyshev-Gauss-Labotto (CGL) collocation schemes are usually used in PS methods [32].

The PS transcription by using (4.13) for the general problem G can be found in [5, 33]. Here two types of PS transcription with matrix formulations for the convex problem C are presented as follows.

(1) Consider the dynamic constraints in problem C as a first-order system, and let

$$\begin{aligned} \mathbf{X} &= [\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_N]^T = [\mathbf{x}(\tau_0), \mathbf{x}(\tau_1), \ldots, \mathbf{x}(\tau_N)]^T \in {}^{(N+1)\times 2n} \\ \mathbf{U} &= [\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_N]^T = [\mathbf{u}(\tau_0), \mathbf{u}(\tau_1), \ldots, \mathbf{u}(\tau_N)]^T \in {}^{(N+1)\times m} \end{aligned} \tag{4.16}$$

Using (4.13) yields

$$\mathcal{D}\mathbf{X} = \dot{\mathbf{X}} \tag{4.17}$$

For the convenience of representation, rewrite the dynamic constraint (4.9) in problem C as

$$\dot{\mathbf{x}}(\tau) = \begin{bmatrix} \dot{\mathbf{x}}_1(\tau) \\ \dot{\mathbf{x}}_2(\tau) \end{bmatrix} = s \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{a}_1 & \mathbf{a_2} \end{bmatrix} \mathbf{x}(\tau) + s \begin{bmatrix} \mathbf{0} \\ \mathbf{g} \end{bmatrix} \mathbf{u}(\tau) \triangleq \mathbf{F}(\mathbf{x}, \mathbf{u}) \tag{4.18}$$

Denote

$$\mathcal{F}(\mathbf{X}, \mathbf{U}) = [\mathbf{F}_0, \mathbf{F}_1, \ldots, \mathbf{F}_N]^T = [\mathbf{F}(\mathbf{x}(\tau_0), \mathbf{u}(\tau_0)), \mathbf{F}(\mathbf{x}(\tau_1), \mathbf{u}(\tau_1)), \ldots, \mathbf{F}(\mathbf{x}(\tau_N), \mathbf{u}(\tau_N))]^T \tag{4.19}$$

Thus, problem C can be transcribed as

$$\begin{aligned} &\text{Decision Variables} : \mathbf{X} =\in {}^{(N+1)\times 2n}, \mathbf{U} \in {}^{(N+1)\times m} \\ &\text{Problem } DPSC1 : \min J^N[\mathbf{X}, \mathbf{U}] := E(\mathbf{x}_0, \mathbf{x}_N) \\ &\quad s.t. \quad \mathcal{D}\mathbf{X} = \mathcal{F}(\mathbf{X}, \mathbf{U}) \\ &\qquad\quad \mathbf{e}(\mathbf{x}_0, \mathbf{x}_N) = \mathbf{0} \end{aligned} \tag{4.20}$$

(2) An alternative PS method for problem C is taking its dynamic constraints as a second-order systems:

$$\ddot{\mathbf{x}}_1(\tau) = s\dot{\mathbf{x}}_2(\tau) = s^2 \left[\mathbf{a}_1\mathbf{x}_1(\tau) + \mathbf{a}_2\dot{\mathbf{x}}_1(\tau) + \mathbf{gu}\right] \triangleq \mathbf{f}(\mathbf{x}_1(\tau), \dot{\mathbf{x}}_1(\tau), \mathbf{u}(\tau)) \tag{4.21}$$

and let

$$\mathbf{X}_1 = [\mathbf{x}_{10}, \mathbf{x}_{11}, \ldots, \mathbf{x}_{1N}]^T = [\mathbf{x}_1(\tau_0), \mathbf{x}_1(\tau_1), \ldots, \mathbf{x}_1(\tau_N)]^T \in {}^{(N+1)\times n} \tag{4.22}$$

$$\mathcal{F}(\mathbf{X}_1, \dot{\mathbf{X}}_1, \mathbf{U}) = [\mathbf{f}_0, \ldots, \mathbf{f}_N]^T = [\mathbf{f}(\mathbf{x}_1(\tau_0), \dot{\mathbf{x}}_1(\tau_0), \mathbf{u}(\tau_0)) \ldots, \mathbf{f}(\mathbf{x}_1(\tau_N), \dot{\mathbf{x}}_1(\tau_N), \mathbf{u}(\tau_N))]^T \tag{4.23}$$

thus, the system (4.18) can be transcribed as

$$\mathcal{D}\mathbf{X}_1 = \dot{\mathbf{X}}_1 = s\mathbf{X}_2 = s[\mathbf{x}_{20}, \mathbf{x}_{21}, \ldots, \mathbf{x}_{2N}]^T = s[\mathbf{x}_2(\tau_0), \mathbf{x}_2(\tau_1), \ldots, \mathbf{x}_2(\tau_N)]^T \tag{4.24}$$

and

$$\mathcal{D}^{(2)}\mathbf{X}_1 = \ddot{\mathbf{X}}_1 = \mathcal{F}(\mathbf{X}_1, \dot{\mathbf{X}}_1, \mathbf{U}) = \mathcal{F}(\mathbf{X}_1, \mathbf{X}_2, \mathbf{U}) \tag{4.25}$$

Combining (4.21), (4.25), problem C can be transcribed as the following second-order form:

$$\begin{aligned} &\text{Decision Variables} : \mathbf{X}_1 \in {}^{(N+1)\times n}, \mathbf{U} \in {}^{(N+1)\times m} \\ &\text{Temporal Variable} : \mathbf{X}_2 = \mathcal{D}\mathbf{X}_1/s \\ &\text{Problem } DPSC2 : \min J^N[\mathbf{X}, \mathbf{U}] := E(\mathbf{x}_0, \mathbf{x}_N), \mathbf{X} = [\mathbf{X}_1^T, \mathbf{X_2}^T]^T \\ &\quad s.t. \quad \mathcal{D}^{(2)}\mathbf{X}_1 = \mathcal{F}(\mathbf{X}_1, \mathbf{X}_2, \mathbf{U}) \\ &\qquad\quad \mathbf{e}(\mathbf{x}_0, \mathbf{x}_N) = \mathbf{0} \end{aligned} \tag{4.26}$$

where the temporal variable $\mathbf{X}_2$ is algebraically represented by the decision variable $\mathbf{X}_1$.

It is obviously that the number of decision variables in problem *DPSC2* is only half of that in problem *DPSC2*, and this will reduce the scale of convexified problem and alleviate the requirement of computational memory. It is to note that such reduction of decision variables will make the transcribed problem denser, which will undermine the solving efficiency for some convex optimization solver. The PSDM $\mathcal{D}$ and $\mathcal{D}^{(2)}$ usually include large round-off error which is resulted from the large condition number [34], particularly, the condition numbers of the generated problems *DPSC1* and *DPSC2*, irrespectively, dramatically grow in the manner of $O(N^3)$ and $O(N^4)$ (shown in Fig. in the subsequent of this chapter), where N is the number of mesh grids. The overlarge condition number will lead to an ill conditioned problem *DPSC1* and *DPSC2*, which are difficult to solve since the numerically unstable phenomenon accompanying with the large condition number. Hence, the mesh grid number should be limited to alleviate numerical difficulties, meanwhile, the limited mesh grid number potentially undermines the pursuit of more accurate solution.

Actually, in order to reduce the condition number, there have many studies (see [17] and references therein) to find an appropriate preconditioner $\mathbf{M}$ for PS method with the forms of (4.20) and (4.26), such that the discretized dynamic constrains can be written as

$$\mathbf{M}\mathcal{D}\mathbf{X} = \mathbf{M}\mathcal{F}(\mathbf{X}, \mathbf{U}) \tag{4.27}$$

for (4.20), or

$$\mathbf{M}\mathcal{D}^{(2)}\mathbf{X} = \mathbf{M}\mathcal{F}(\mathbf{X}_1, \mathbf{X}_2\mathbf{U}) \tag{4.28}$$

for (4.26). The function of the matrix $\mathbf{M}$ is to reduce the condition number of the matrix equations resulted by $\mathbf{M}\mathcal{D}$ or $\mathbf{M}\mathcal{D}^{(2)}$. However, given a non-diagonal matrix $\mathbf{M}$, the sparsity of the right hand of (4.27) and (4.28) will decrease therefore increasing the computational burden of the resulted mathematical programming problem [16]. An alternative method is show in the next section, in which, the use of Birkhoff interpolating polynomials [18] provides a new PS method by which solutions can be obtained over thousands of mesh points in a stable manner.

4.3 Well-Conditioned Second-Order Birkhoff PS Method

4.3.1 Birkhoff Interpolation at GL Points

Let $f(\tau)$ be a second-order continuously differentiable function on the interval $\tau \in [-1, 1]$, then the second-order Birkhoff interpolation polynomial $f^N(\tau)$ is given as

$$f^N(\tau) := f(\tau_0)B_0(x) + \sum_{i=1}^{N-1} \ddot{f}(\tau_i)B_i(\tau) + f(\tau_N)B_N(\tau), \tau \in [-1, 1] \tag{4.29}$$

where $\pi^N = [\tau_0, \tau_1, \ldots, \tau_N]$ with $\tau_0 = -1$ and $\tau_N = 1$ is a GL point grid, and $B_i (i = 0, \ldots, N)$, the counterpart of the Lagrange basis polynomials $\{\mathcal{L}\}_{i=0}^{N}$, are the Birkhoff interpolation basis polynomials of order N or less, meanwhile, the following interpolation condition on mesh points must be satisfied:

$$\ddot{f}^N(\tau_i) = \ddot{f}(\tau_i) \text{ for } 1 \le i \le N-1 \text{ and } f^N(\pm 1) = f(\pm 1) \tag{4.30}$$

If the interpolation (4.29) exists, then B_i satisfy the following [18]

$$\begin{aligned} &B_0(\tau_0) = 1, B_N(\tau_0) = 0, B_j(\tau_0) = 0, j = 1, \ldots, N-1 \\ &B_0(\tau_N) = 0, B_N(\tau_N) = 1, B_j(\tau_N) = 0, j = 1, \ldots, N-1 \\ &\ddot{B}_0(\tau_i) = 0, \ddot{B}_N(\tau_i) = 0, \ddot{B}_j(\tau_i) = \delta_{ij} i = 1, \ldots, N-1; j = 1, \ldots, N-1 \end{aligned} \tag{4.31}$$

According to Theorem 3.1 proposed in [18], B_i can be defined in terms of the Lagrange basis polynomials $\{\tilde{\mathcal{L}}_i\}_{i=1}^{N-1}$ of degree $N-2$, therefore

$$\begin{aligned} &B_0(\tau) = \tfrac{1-\tau}{2} \\ &B_i(\tau) = \tfrac{1+\tau}{2} \int_{-1}^{1} (\tau - 1)\tilde{\mathcal{L}}_i(\tau) d\tau + \int_{-1}^{\tau} (\tau - t)\tilde{\mathcal{L}}_i(t) dt, j = 1, \ldots, N-1 \\ &B_N(\tau) = \tfrac{1+\tau}{2}. \end{aligned} \tag{4.32}$$

Evaluate the Birkhoff basis interpolation polynomials on the given GL points π^N, and let $b_{ij} = B_j(\tau_i)$, thus the second-order Birkhoff PS integration matrix (BPSIM) as

$$\begin{aligned} \mathcal{B} &= [b_{ij}]|_{0 \le i,j \le N} \in {}^{(N+1)\times(N+1)} \\ \mathcal{B}_{in} &= [b_{ij}]|_{1 \le i,j \le N-1} \in {}^{(N-1)\times(N-1)} \end{aligned} \tag{4.33}$$

Using the matrix defined in (4.33), rewrite (4.29) on, we henceforth have:

$$\mathcal{B}\tilde{\mathbf{f}}^{(2)} = \mathbf{f} \tag{4.34}$$

where $\tilde{\mathbf{f}}^{(2)} = [f(\tau_0), \ddot{f}(\tau_1), \ldots, \ddot{f}(\tau_{N-1}), f(\tau_N)]^T$, $\mathbf{f} = [f(\tau_0), f(\tau_1), \ldots, f(\tau_{N-1}), f(\tau_N)]^T$. In light of (4.13), differentiating two sides of (4.34) with respect to PS time yields

$$\mathcal{D}\mathcal{B}\tilde{\mathbf{f}}^{(2)} = \mathcal{B}^{(1)}\tilde{\mathbf{f}}^{(2)} = \mathcal{D}\mathbf{f} = \dot{\mathbf{f}} \tag{4.35}$$

and, second-order differentiating has

$$\mathcal{D}^{(2)}\mathcal{B}\tilde{\mathbf{f}}^{(2)} = \mathcal{B}^{(2)}\tilde{\mathbf{f}}^{(2)} = \mathcal{D}^{(2)}\mathbf{f} = \ddot{\mathbf{f}} \tag{4.36}$$

The above equations can be generalized as [18]

$$\mathcal{B}^{(k)} = \mathcal{D}^{(k)}\mathcal{B} = \mathcal{D}^k\mathcal{B} = \mathcal{D}\mathcal{B}^{(k-1)}, k \ge 1 \tag{4.37}$$

Recall (4.15), and replace the first row and the last row of $\mathcal{D}^{(2)}$ by $\mathbf{e}_1 = [1, 0, \ldots, 0]$ and $\mathbf{e}_1 = [1, 0, \ldots, 0]$, respectively, therefore a matrix denoted by $\tilde{\mathcal{D}}^{(2)}$. Then there has

$$\widetilde{\mathcal{D}}^2\mathbf{f} = \tilde{\mathbf{f}}^2 \tag{4.38}$$

From (4.34) and (4.38), it is obvious that $\widetilde{\mathcal{D}}^2\mathbf{f} = \widetilde{\mathcal{D}}^2\mathcal{B}\tilde{\mathbf{f}}^{(2)} = \tilde{\mathbf{f}}^{(2}$, consequently, we have

$$\widetilde{\mathcal{D}}^2\mathcal{B} = \mathbf{I}_{N+1}, \mathcal{D}_{in}^2\mathcal{B}_{in} = \mathbf{I}_{N-1} \tag{4.39}$$

where $\mathcal{D}_{in}^{(2)}$ is a submatrix of $\mathcal{D}^{(2)}$, i.e, $\mathcal{D}_{in}^2 = [d_{i,j}]_{1\le i,j\le N-1}$; and $\mathbf{I^n}$ is an identity matrix.

According to (4.39), we can compute $\mathcal{B}$ by the inverse of $\widetilde{\mathcal{D}}^2$. Unfortunately, the condition number of $\widetilde{\mathcal{D}}^2$ will dramatically increase for the overlarge mesh points, and this will lead to the loss of accuracy for the matrix inverse computation. Consequently, it is necessary to formulate the Birkhoff integration matrix in accordance with (4.32), and the details for computing $\mathcal{B}$ can be found in [18].

Besides the second-order Birkhoff interpolation polynomial given by (4.29), one can easily obtain the first-order Birkhoff polynomial and the corresponding first-order BPSIM. The applications of first-order BPSIM for the OCP are demonstrated in [18, 35], in which, the OCP is transcribed into an NLP problem by first-order Birkhoff PS method. The interesting readers can found the details about first-order BPSIM in [35].

4.3.2 Preconditioned Birkhoff PS Method

It is possible to facilitate solving problem *DPSC2* by using Birkhoff PSIM as preconditioners. Let the mesh grid $\pi^N = [\tau_0, \tau_{in}^N, \tau_N]$ where π_{in}^N is the set of inner points except boundary points τ_0 and τ_N, and partition the algebraic equation $\mathcal{D}^{(2)}\mathbf{X}_1 = \mathcal{F}(\mathbf{X}_1, \mathbf{X}_2, \mathbf{U})$ in (4.26) as

$$\begin{bmatrix} d_{0,0}^{(2)} & \mathbf{d}_{0,1:N-1}^{(2)} & d_{0,N}^{(2)} \\ \mathbf{d}_{1:N-1,0}^{(2)} & \mathcal{D}_{in} & \mathbf{d}_{1:N-1,N}^{(2)} \\ d_{N,0}^{(2)} & \mathbf{d}_{N,1:N-1}^{(2)} & d_{N,N}^{(2)} \end{bmatrix} \begin{bmatrix} \mathbf{X}_{10} \\ \mathbf{X}_{1in} \\ \mathbf{X}_{1N} \end{bmatrix} = \begin{bmatrix} \mathcal{F}_0(\mathbf{X}_1, \mathbf{X}_2, \mathbf{U}) \\ \mathcal{F}_{in}(\mathbf{X}_1, \mathbf{X}_2, \mathbf{U}) \\ \mathcal{F}_N(\mathbf{X}_1, \mathbf{X}_2, \mathbf{U}) \end{bmatrix} \tag{4.40}$$

where $d_{i,j}^{(2)}$ is the entry of the second PSDM $\mathcal{D}^{(2)}$. We henceforth have

$$\mathcal{D}_{in}\mathbf{X}_{in} = \mathcal{F}_{in}(\mathbf{X}_1, \mathbf{X}_2, \mathbf{U}) - \mathbf{d}_{1:N-1,0}^{(2)}\mathbf{X}_{10} - \mathbf{d}_{1:N-1,N}^{(2)}\mathbf{X}_{1N} \tag{4.41}$$

Left-multiplying both sides of (4.41) by $\mathcal{B}^{in}$ and using (4.39) yield

$$\mathbf{X}_{in} = \mathcal{B}_{in}\mathcal{F}_{in}(\mathbf{X}_1, \mathbf{X}_2, \mathbf{U}) - \mathcal{B}_{in}\mathbf{d}_{1:N-1,0}^{(2)}\mathbf{X}_{10} - \mathcal{B}_{in}\mathbf{d}_{1:N-1,N}^{(2)}\mathbf{X}_{1N} \tag{4.42}$$

Meanwhile, the boundary points at τ_0 and τ_N are constrained by

$$\mathcal{D}_0^{(2)}\mathbf{X}_1 = \mathcal{F}(\mathbf{X}_{10}, \mathbf{X}_{20}, \mathbf{U}_0), \mathcal{D}_N^{(2)}\mathbf{X}_1 = \mathcal{F}(\mathbf{X}_{1N}, \mathbf{X}_{2N}, \mathbf{U}_N) \tag{4.43}$$

where a matrix with subscript i represents the ith row vector of the matrix.

Based on (4.41) (4.43), the preconditioned problem *DPSC2* is given as follows

$$\begin{aligned}
&\text{DecisionVariables} : \mathbf{X}_1 \in {}^{(N+1)\times n}, \mathbf{U} \in {}^{(N+1)\times m}\\
&\text{TemporalVariable} : \mathbf{X}_2 = \mathcal{D}\mathbf{X}_1/s\\
&\text{Problem } P-DPSC2 : \min J^N[\mathbf{X}, \mathbf{U}] := E(\mathbf{x}_0, \mathbf{x}_N), \mathbf{X} = [\mathbf{X}_1^T, {\mathbf{X}_\mathbf{2}}^T]^T\\
&\qquad s.t.\ \ \mathcal{D}_0^{(2)}\mathbf{X}_1 = \mathcal{F}(\mathbf{X}_{10}, \mathbf{X}_{20}, \mathbf{U}_0),\\
&\qquad\quad \mathbf{X}_{in} = \mathcal{B}_{in}\mathcal{F}_{in}(\mathbf{X}_1, \mathbf{X}_2, \mathbf{U}) - \mathcal{B}_{in}\mathbf{d}_{1:N-1,0}^{(2)}\mathbf{X}_{10} - \mathcal{B}_{in}\mathbf{d}_{1:N-1,N}^{(2)}\mathbf{X}_{1N}\\
&\qquad\quad \mathcal{D}_N^{(2)}\mathbf{X}_1 = \mathcal{F}(\mathbf{X}_{1N}, \mathbf{X}_{2N}, \mathbf{U}_N)\\
&\qquad\quad \mathbf{e}(\mathbf{x}_0, \mathbf{x}_N) = \mathbf{0}
\end{aligned} \tag{4.44}$$

4.3.3 Birkhoff PS Method for Convex Optimal Control

In order to apply the Birkhoff interpolates of the previous subsection to problem *C*. Denote the unknown optimization variables over the mesh grid π^N by

$$\mathbf{V}^N = [\mathbf{x}_{10}, \mathbf{x}_{11}, \ldots, \mathbf{x}_{1N}]^T = [\mathbf{x}_1(\tau_0), \ddot{\mathbf{x}}_1(\tau_1), \ldots, \ddot{\mathbf{x}}_{1(N-1)}(\tau_{N-1}), \mathbf{x}_1(\tau_N)]^T \in \mathbb{R}^{(N+1)\times n} \tag{4.45}$$

correspondingly, over the inner points τ_{in}^N we have

$$\mathbf{V}_{in}^N = [\mathbf{x}_{11}, \ldots, \mathbf{x}_{1(N-1)}]^T = [\ddot{\mathbf{x}}_1(\tau_1), \ldots, \ddot{\mathbf{x}}_{1(N-1)}(\tau_{N-1})]^T \in \mathbb{R}^{(N+1)\times n} \tag{4.46}$$

According to (4.34) (4.37) we have

$$\mathbf{X}_1 = \begin{bmatrix} \mathbf{X}_{10} \\ \mathbf{X}_{1in} \\ \mathbf{X}_{1N} \end{bmatrix} = \mathcal{B}\mathbf{V} = \mathcal{B}\begin{bmatrix} \mathbf{V}_0 \\ \mathbf{V}_{in} \\ \mathbf{V}_N \end{bmatrix} \tag{4.47}$$

$$\mathcal{B}^{(1)}\mathbf{V} = \mathcal{D}\mathcal{B}\mathbf{V} = \mathcal{D}\mathbf{X}_1 = \dot{\mathbf{X}}_1 = s\mathbf{X}_2 \tag{4.48}$$

$$\mathcal{B}^{(2)}\mathbf{V} = \mathcal{D}^2\mathcal{B}\mathbf{V} = \mathcal{D}^2\mathbf{X}_\mathbf{1} = \ddot{\mathbf{X}}_1 \tag{4.49}$$

where $\mathcal{B}_0$ and $\mathcal{B}_N$, respectively, are the first and last row of $\mathcal{B}$. Thus, for the dynamic constraints (4.21) we have

$$\mathcal{B}_{0.}^{(2)}\mathbf{V} = \mathcal{F}(\mathbf{X}_{10}, \dot{\mathbf{X}}_{10}, \mathbf{U}_0) \tag{4.50}$$

$$\mathbf{V}_{in} = \mathcal{F}_{in}(\mathbf{X}_{1in}, \dot{\mathbf{X}}_{1in}, \mathbf{U}_{in}) \tag{4.51}$$

$$\mathcal{B}_{N.}^{(2)}\mathbf{V} = \mathcal{F}(\mathbf{X}_{1N}, \dot{\mathbf{X}}_{1N}, \mathbf{U}_{N}) \tag{4.52}$$

The Birkhoff equality constraints (4.50) and (4.52) impose the differential equation (4.21) at the boundary point τ_0 and τ_N. The following proposition presents the equivalency between the above Birkhoff equality constraints and the differential equation via the Lagrange condition.

Proposition *Let* $\mathbf{X}_{1j} = \mathbf{x}_{1j}^T = \mathbf{x}_1^T(\tau_j)$*, At* $\tau = \tau_0$ *and* $\tau = \tau_N$*, the Lagrange and Birkhoff interpolants respectively satisfy the conditions*

$$\sum_{j=0}^{N} \mathbf{X}_{1j}\, \ddot{\mathcal{L}}(\tau_0) = \mathcal{B}_{0.}^{(2)}\mathbf{V} \tag{4.53}$$

and

$$\sum_{j=0}^{N} \mathbf{X}_{1j}\, \ddot{\mathcal{L}}(\tau_N) = \mathcal{B}_{N.}^{(2)}\mathbf{V} \tag{4.54}$$

Proof Recall the Birkhoff interpolation polynomial (4.29) and the Lagrange interpolants, we have

$$\begin{aligned} \mathbf{X}_1(\tau) &= \sum_{j=0}^{N} \mathbf{X}_{1j}\mathcal{L}(\tau) = \mathbf{X}_{10}B_0(\tau) + \sum_{j=1}^{N-1} \mathbf{V}_i B_i(\tau) + \mathbf{X}_{1N}B_N(\tau) \\ &= \mathbf{V}_1 B_0(\tau) + \sum_{j=1}^{N-1} \mathbf{V}_i B_i(\tau) + \mathbf{V}_N B_N(\tau) \end{aligned} \tag{4.55}$$

where $\mathbf{V}_i$ denotes the ith row of $\mathbf{V}$. Differentiating (4.55) twice with respect to PS time τ yields

$$\begin{aligned} \sum_{j=0}^{N} \mathbf{X}_{1j}\ddot{\mathcal{L}}(\tau) &= \mathbf{V}_0\ddot{B}_0(\tau) + \sum_{j=1}^{N-1} \mathbf{V}_i \ddot{B}_i(\tau) + \mathbf{V}_N \ddot{B}_N(\tau) \\ &= \left[\ddot{B}_0(\tau), \ddot{B}_1(\tau), \ldots, \ddot{B}_N(\tau)\right]\mathbf{V} \end{aligned} \tag{4.56}$$

Evaluating (4.56) at $\tau = \tau_0$ we henceforth have

$$\sum_{j=0}^{N} \mathbf{X}_{1j}\ddot{\mathcal{L}}(\tau_0) = \left[\ddot{B}_0(\tau_0), \ddot{B}_1(\tau_0), \ldots, \ddot{B}_N(\tau_0)\right]\mathbf{V} \tag{4.57}$$

It is obvious that the first row of BPSIM is the evaluating value of $B_j(\tau)(j = 0, \ldots, N)$ at $\tau = \tau_0$ in light of (4.33), further, the first row of $\mathcal{B}^{(2)}$ denoted by $\mathcal{B}_{0.}^{(2)} = \left[\ddot{B}_0(\tau_0), \ddot{B}_1(\tau_0), \ldots, \ddot{B}_N(\tau_0)\right]$. Thus (4.53) holds, and it is similar to (4.54). □

Collecting the above relevant equation, the problem C can be transcribed as the follows

$$\begin{aligned}
&\text{DecisionVariables} : \mathbf{V} \in {}^{(N+1)\times n}, \mathbf{U} \in {}^{(N+1)\times m} \\
&\text{TemporalVariables} : \mathbf{X_1} = \mathcal{B}\mathbf{V}, \mathbf{X}_2 = \mathcal{B}^{(1)}\mathbf{V} \\
&\text{Problem}\, BPSC2 : \min J^N[\mathbf{X}, \mathbf{U}] := E(\mathbf{x}_0, \mathbf{x}_N), \mathbf{X} = [\mathbf{X}_1^T, \mathbf{X_2}^T]^T \\
&\qquad s.t. \quad \mathcal{B}_{0.}^{(2)}\mathbf{V} = \mathcal{F}(\mathbf{X}_{10}, \dot{\mathbf{X}}_{10}, \mathbf{U}_0) \\
&\qquad\quad \mathbf{V}_{in} = \mathcal{F}_{in}(\mathbf{X}_{1in}, \dot{\mathbf{X}}_{1in}, \mathbf{U}_{in}) \\
&\qquad\quad \mathcal{B}_{N.}^{(2)}\mathbf{V} = \mathcal{F}(\mathbf{X}_{1N}, \dot{\mathbf{X}}_{1N}, \mathbf{U}_N) \\
&\qquad\quad \mathbf{e}(\mathbf{x}_0, \mathbf{x}_N) = \mathbf{0}.
\end{aligned} \tag{4.58}$$

4.4 Application Examples

To demonstrate the effectiveness of the proposed method for OCPs by using Birkhoff PS method and convex programming algorithms, two application examples are considered in this section. One is a simple cart problem with boundary position and velocity constraints, which is convex in nature. Another is an online rescue trajectory optimization problem for launch ascent to orbital insertion, which is concave and complicated, and SCP method by using successive linearization is required.

The following numerical results are obtained on a laptop computer with an Intel Core i7-1065G7 CPU @ 1.3GHz and 32G RAM. YALMIP [36] and ECOS [27, 37] are used for modeling and solving problems. YALMIP is a MATLAB toolbox for the rapid modeling for mathematical optimization problems, while ECOS is a light embedded toolkit based on the primal-dual interior-point algorithm. ECOS is designed to solve the following standard SOCP problem.

$$\begin{aligned}
&\min \mathbf{c}^T\mathbf{x} \\
&s.t. \mathcal{A}\mathbf{x} = \mathbf{b} \\
&\mathcal{G}\mathbf{x} + \mathbf{s} = \mathbf{h}, \mathbf{s} \in \mathcal{K}
\end{aligned} \tag{4.59}$$

where $\mathbf{x}$ are decision variables to be optimized, s are the slack variables, and $\mathcal{K}$ are conic constraints. The previous problems of DPSC2, P-DPSC2, and BPSC2 are modeled by using YALMIP therefore a programming problem with the form of (4.59). The matrix $\mathcal{A}$ in equality constraints in (4.59) strongly depends on the collocation scheme, and the condition number of $\mathcal{A}$ will dramatically impacts on the efficiency of solving algorithms, although many special skills are proposed in ECOS to handle such phenomenon. Hence in the following numerical studies, the condition number of $\mathcal{A}$ is taken as an index to evaluate the performance of different PS schemes including the first-order and second-order methods.

In the remainder of this paper, three second-order methods refer to DPSC2 in (4.26), P-DPSC2 in (4.44) and BPSC2 in (4.58). While three first-order methods, respectively, are problem DPSC1 in (4.20), the preconditioned version denoted by P-DPSC1, and BPSC1 transcribed by using the first-order Birkhoff PS method. Note

the formulations of P-DPSC1 and BPSC1 resulted from the first-order Birkhoff PS method are not presented in this paper, actually they are similar to P-DPSC2 and BPSC2, and the interested readers can find the details in [16]. Moreover, to demonstrate the effects of different collocation schemes, CGL and LGL collocation schemes are considered. The computation of BPSIM relating to Chebyshev and Legendre polynomials are given in [18].

4.4.1 Simple Cart Problem

An OCP of a simple cart system with fixed final time is given as follows [38]

$$\begin{aligned} &\min J = \int_0^{t_f} u^2 dt \\ &s.t. \dot{x}_1 = x_2, \dot{x}_2 = -x_2 + u \\ &x_1(0) = 0, x_2(0) = 0 \\ &a x_1(t_f) + b x_2(t_f) - c = 0 \end{aligned} \tag{4.60}$$

where the parameters $a = 1.0, b = -2.694528, c = -1.155356$, the final time $t_f = 2$. Using the indirect method for (4.60) results the analytical solutions given as

$$\begin{aligned} u(t) &= \tfrac{1}{4}e^t - \tfrac{1}{2} \\ x_1(t) &= -\tfrac{3}{8}e^{-t} + \tfrac{1}{8}e^t - \tfrac{1}{2}t + \tfrac{1}{4} \\ x_2(t) &= \tfrac{3}{8}e^{-t} + \tfrac{1}{8}e^t - \tfrac{1}{2} \end{aligned} \tag{4.61}$$

with the optimal performance index $J_{ana} = 0.577678$.

The problem (4.60) is convex in nature, hence the aforementioned six PS methods with convex programming can be directly used, and the convexifying operations are not required. Figure 4.1 presents the analytical solution as well as that of BPSC2 with LGL mesh grid. The interpolating curves based on the optimized values on mesh points via Lagrange polynomials at the freedom of degree almost identify with the analytical solution.

The comparison results are demonstrated by Tables 4.1 and 4.2, in which, six PS methods (DPSC1, P-DPSC1, BPSC1, DPSC2, P-DPSC2 and BPSC2) are implemented through CGL and LGL mesh grids ($N = 5, 20, 60, 120$), respectively. Four different indices are listed as follows.

(1) Cond.$\mathcal{A}$ refers to the condition number of the matrix in equality constraint of (4.59).

(2) The integration error e in two tables is defined by

$$e = \|\mathbf{x} - \mathbf{x}_{ana}\|_\infty \tag{4.62}$$

where $\mathbf{x}_{ana} = [x_{1ana}, x_{2ana}]$ are given by (4.61), $\mathbf{x} = [x_1, x_2]^T$ are the integration solution of simple cart system via 4-order Runge–Kuta algorithm with the control

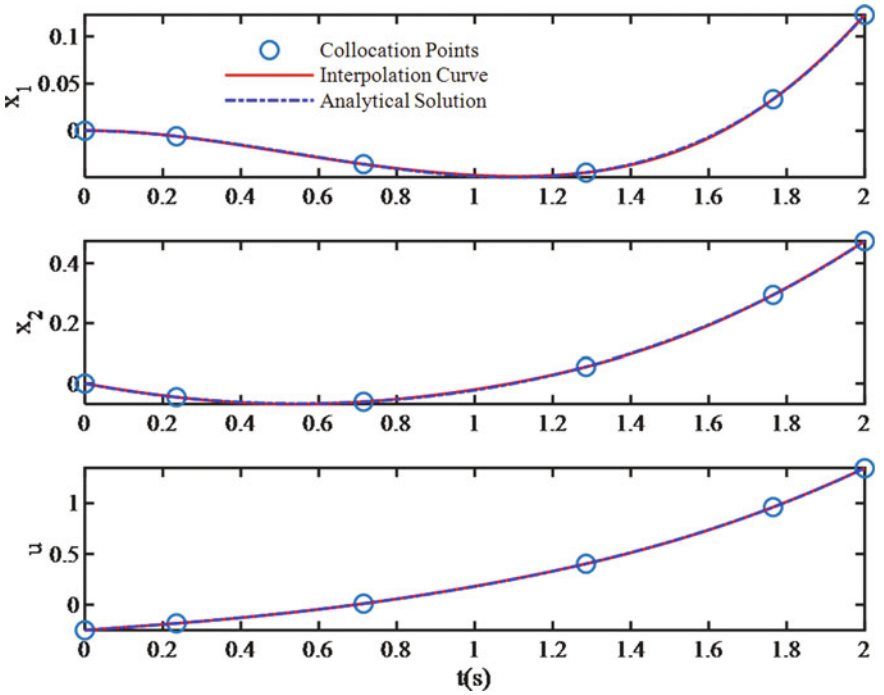


Fig. 4.1 Optimal solution via BPSC2 with LGL collocation scheme and the analytical solution (mesh grid $N = 5$)

input $u(t)$ which is interpolated from the optimized control variables u^N on mesh nodes, and the step of Runge-Kuta integration is $t_s = 0.001s$.

(3) $\Delta J = |J^N - J_{ana}|$, where J_N is resulted from optimized algorithms on different mesh grid N.

(4) Solve time in millisecond refer to the time consumed by ECOS.

Observe from Tables 4.1 and 4.2 that the proposed second-order BPSC methods provide a more stable performance than other methods while increasing the number of mesh grids N. The crucial reason lies that when the mesh grids increase, the condition number of $\mathcal{A}$ in DPSC2 will dramatically grows like $O(N^4)$, and that in BPSC2 behaves like $O(N)$. The growth of condition number of $\mathcal{A}$ resulted from DPSC1, DPSC2, BPSC1, and BPSC2 is clearly illustrated in Fig. 4.2 from which, CGL collocation scheme is slightly advantageous over than LGL. Meanwhile, the solver time consumptions of all mentioned methods provided in Tables 4.1 and 4.2 strongly relate to the condition number. It is also can be observed that when $N \geq 60$, DPSC2 and P-DPSC2 cannot provide a reasonable solution since the numerical instability aroused by the overlarge condition number. It can be seen that, preconditioned method using BPSIM according to the method in [16], and the main reason lies in the handling of boundary constraints. In [17], the authors proposed another preconditioning method, however, the method concentrates on solving general ODEs with

Table 4.1 Comparison of condition numbers, integration error, performance index error and solve time for first-order PS methods

N	DPSC1				P-DPSC1				BPSC1			
	Cond.$\mathcal{A}$	Integration error e	ΔJ_i	Solve time(ms)	Cond.$\mathcal{A}$	Integration error e	ΔJ_i	Solve time(ms)	Cond.$\mathcal{A}$	Integration error e	ΔJ_i	Solve time(ms)
Chebyshev												
5	127.32	1.10E-03	7.91E-05	3.11	382.34	1.10E-03	7.91E-05	2.88	6.75E+01	1.10E-03	7.91E-05	2.41
20	6.37E+03	8.13E-04	3.04E-07	3.71	1.51E+05	7.99E-04	3.04E-07	2.2	3.69E+02	7.99E-04	3.04E-07	3.26
60	1.69E+05	7.92E-04	5.28E-06	11.8	2.07E+07	8.00E-04	3.26E-07	14.49	1.87E+03	7.99E-04	3.04E-07	11.65
120	1.35E+06	4.86E-04	1.45E-04	236.94	4.66E+08	8.41E-04	1.89E-07	116.3	5.28E+03	7.99E-04	3.04E-07	88.42
Legendre												
5	1.45E+02	8.45E-04	6.69E-07	2.32	3.24E+02	8.45E-04	6.68E-07	2.85	8.53E+01	8.45E-04	6.68E-07	2.61
20	6.78E+03	7.99E-04	3.04E-07	2.93	6.25E+04	7.99E-04	3.04E-07	2.86	1.01E+03	7.99E-04	3.04E-07	2.34
60	1.75E+05	7.99E-04	3.05E-07	24	4.80E+06	7.99E-04	3.05E-07	13.5	8.89E+03	7.99E-04	3.04E-07	9.13
120	1.38E+06	8.19E-04	1.96E-06	331.98	7.58E+07	8.02E-04	4.38E-07	65.97	3.54E+04	7.99E-04	3.04E-07	49.56

Table 4.2 Comparison of condition numbers, integration error, performance index error and solve time for second-order PS methods

N	DPSC1				P-DPSC1				BPSC1			
	Cond.$\mathcal{A}$	Integration error e	ΔJ_i	Solve time(ms)	Cond.$\mathcal{A}$	Integration error e	ΔJ_i	Solve time(ms)	Cond.$\mathcal{A}$	Integration error e	ΔJ_i	Solve time(ms)
Chebyshev												
5	3.86E+02	5.04E-04	7.91E-05	3.92	4.89E+03	0.001	7.91E-05	1.87	9.29	0.001	7.91E-05	3.36
20	1.35E+05	8.13E-04	2.58E-05	8.47	1.07E+08	8.29E-04	4.75E-05	8.99	26.90	7.99E-04	3.04E-07	1.91
60	1.82E+07	**0.5737**	**0.5767**	30.77	3.92E+11	**0.5306**	**0.5704**	57.23	74.64	7.99E-04	4.04E-07	4.24
120	4.09E+08	**0.5972**	**0.5777**	362.02	7.05E+13	**0.5985**	**0.5777**	499.75	1.46E+02	7.99E-04	3.04E-07	18.47
Legendre												
5	3.92E+02	8.44E-04	6.68E-06	2.36	4.12E+03	8.45E-04	6.68E-07	3.00	11.30	8.45E-04	6.69E-07	1.63
20	1.19E+05	8.01E-04	5.80E-06	25.73	4.75E+07	8.10E-04	2.10E-05	2.09	61.26	7.99E-04	3.04E-07	2.34
60	1.49E+07	**0.54**	**0.5728**	99.99	9.40E+10	**0.49**	**0.5613**	164.4	2.91E+02	7.99E-04	3.04E-07	4.23
120	3.29E+08	**0.60**	**0.5777**	360.12	1.17E+13	**0.61**	**0.5774**	516.22	8.03E+02	7.99E-04	4.04E-07	23.52

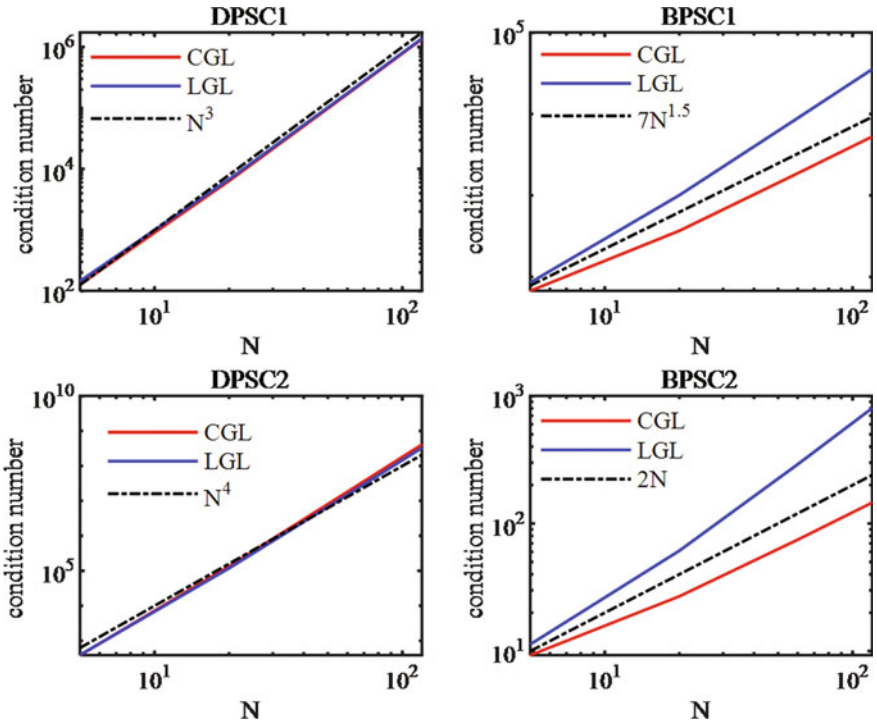


Fig. 4.2 Condition numbers of $\mathcal{A}$ resulted from DPSC1, DPSC2, BPSC1 and BPSC2

boundary conditions, the control input cannot been considered. Moreover, comparing BPSC1 and BPSC2, the latter method demonstrates more efficiency while the increase of mesh grid number.

4.4.2 Rescue Orbit Searching Problem

The second illustrative problem is about launch insertion while thrust failure. When a thrust drop failure occurs during a launch mission, may the prescribed target orbit be reachable or not? This question can be addressed by searching a maximum-height circular orbit (MCO) in the orbital plane formed at the time of failure, and this is validated by [21], in which MCO is obtained by solving the following optimal control problem.

$$\text{ProblemMCO} : \min J = -h_f \tag{4.63}$$

$$\begin{aligned} s.t. \quad & \dot{\mathbf{r}} = \mathbf{v} \\ & \dot{\mathbf{v}} = -\tfrac{1}{r^3}\mathbf{r} + \tfrac{\kappa T}{m}\mathbf{i_b}, \quad \|\mathbf{i}_b\| = 1 \end{aligned} \tag{4.64}$$

$$\dot{m} = -\frac{\kappa T}{I_{sp} g_0} \tag{4.65}$$

$$e(t_f) = 0 \Leftrightarrow \begin{cases} \mathbf{r}(t_f)^T \mathbf{v}(t_f) = 0 \\ r(t_f) v^2(t_f) - \mu = 0 \end{cases} \tag{4.66}$$

$$i(t_f) = i_{t_f} \Leftrightarrow \begin{cases} \mathbf{r}(t_f)^T \mathbf{i}_{h\text{ref}} = 0 \\ \mathbf{v}(t_f)^T \mathbf{i}_{h\text{ref}} = 0 \end{cases} \tag{4.67}$$

$$\mathbf{r}(t_0) = \mathbf{r}_0, \mathbf{v}(t_0) = \mathbf{v}_0 \tag{4.68}$$

where $\mathbf{r}$ and $\mathbf{v}$ are the position and velocity vectors of the launch vehicle in the earth centered inertial (ECI) coordinate, respectively, normalized by the Earth radius R_E and $\sqrt{R_E g_E}$, where g_E is the gravitational acceleration at sea level, meanwhile their magnitudes defined by $r = ||\mathbf{r}||$, $v = ||\mathbf{v}||$; the dependent variable t is normalized by $\sqrt{R_E/g_E}$; the thrust direction is given by $\mathbf{i}_b$; thrust failure factor κ refers to the percentage of the failure thrust compared to the nominal thrust T normalized by $m_0 g_E$, where m_0 is the initial mass of vehicle; the mass m is scaled by m_0, and the rocket engine's specific impulse $I_s p$ in seconds is scaled by $\sqrt{R_E/g_E}$; g_0 represents normalized gravitational acceleration by g_E;. The equations (4.66) and (4.67), respectively, impose constraints on the eccentricity e and the inclination angle i on the final time t_f. The reference normalized orbital angular moment is provided by

$$\mathbf{i}_{h\text{ref}} = [\sin\Omega_{ref} \sin i_{ref}, -\cos\Omega_{ref} \sin i_{ref}, \cos i_{ref}]^T \tag{4.69}$$

where Ω_{ref} and i_{ref} are the longitude of the ascending node (LAN) and the inclination angle, which are predefined by the nominal trajectory without any failure.

To solve problem MCO via convex programming method, an initial trajectory is necessary for linearizing and convexifying the non-convex constraints. However, a good initial trajectory, which can make solving procedure definitely converges, is hard to be obtained. In [22], through estimating the geocentric angle of injection point, a temporary orbital coordinate system (OCS) was established, then the maximum height searching was conducted in OCS, in which, the problem MCO is confined in 2-dimensional space. The solution obtained in OCS was converted into original inertial coordinate system, and was taken as the initial trajectory of problem MCO.

In this paper, a new procedure for solving problem MCO is presented. Note that the specific impulse I_{sp} remains whatever unchanged, and the thrust magnitude is assumed being a known constant after the thrust drop. Hence the mass rate $\dot{m}$ as well as the remaining flight time $t_{go} = m_{prop}/\dot{m}$ can be calculated after thrust failure occurring, where m_{prop} is the known remaining propellent. As a result, the mass variation equation in (4.65) can be replaced by an algebraic equation

$$m(t) = m_0 - (t - t_0)\dot{m}, t_0 \le t \le t_0 + t_{go} \tag{4.70}$$

Correspondingly, the acceleration magnitude in any time instant is given by

$$A(t) = \frac{\kappa T}{m(t)} \tag{4.71}$$

Hence the dynamic constraints in (4.65) can be omitted in the subsequent optimization procedure, and the term of T/m in (4.64) can be replaced by A in (4.71).

The proposed iteratively solving procedure via SC algorithm is as follows.

Step 1. In the first iteration $k = 0$, let the radius of vehicle $r^0(t)$ be a straight line from the vehicle's current radius $r(t_0)$ and the prescribed radius $r(t_f)$ of the nominal trajectory. Irrespective of the constraints (4.66), we henceforth have the following simplified convex version of problem MCO.

$$\begin{aligned}
&\text{Subproblem0 :}\\
&\min J = -x_{of}\\
&s.t. \dot{\mathbf{r}} = \mathbf{v},\ \dot{\mathbf{v}} = -\mathbf{r}/(r^0)^3 + A(t)\mathbf{i}_b\\
&\mathbf{r}(t_f)^T\mathbf{i}_{h'} = 0,\ \mathbf{v}(t_f)^T\mathbf{i}_{h'} = 0\\
&\mathbf{r}(t_0) = \mathbf{r}_0,\ \mathbf{v}(t_0) = \mathbf{v}_0\\
&\|\mathbf{i}_b\| \le 1
\end{aligned} \tag{4.72}$$

Here, the index performance

$$x_{of} = [1, 0, 0]\,\mathbf{T}_{ECI}^{O'}\mathbf{r}(t_f) \tag{4.73}$$

represents the position along the x-axis in a temporary OCS, which is defined by the LAN Ω' and the inclination angle i' when the thrust failure occurs, as well as the prescribed argument of the perigee ω_{ref}, thereby

$$\mathbf{T}_{ECI}^{O'} = \begin{bmatrix} \cos\Omega'\cos\omega_{ref} - \sin\Omega'\sin\omega_{ref}\cos i' & \sin\Omega'\sin\omega_{ref} + \cos\Omega'\sin\omega_{ref}\cos i' & \sin\omega_{ref}\sin i' \\ -\cos\Omega'\sin\omega_{ref} - \sin\Omega'\cos\omega_{ref}\cos i' & -\sin\Omega'\sin\omega_{ref} + \cos\Omega'\cos\omega_{ref}\cos i' & \cos\omega_{ref}\sin i' \\ \sin\Omega'\sin i' & -\cos\Omega'\sin i' & \cos i' \end{bmatrix} \tag{4.74}$$

Note that Ω' and i' can be calculated by the current position $\mathbf{r}_0$ and velocity $\mathbf{v}_0$ of the vehicle. Correspondingly, the desired terminal position and the velocity vectors are constrained by $\mathbf{i}_{h'}$ defined by

$$\mathbf{i}_{h'} = [\sin\Omega'\sin i', -\cos\Omega'\sin', \cos i']^T \tag{4.75}$$

Problem (4.72) can be solved by SOCP solver after discretization, thereby the solution denoted by $\{\mathbf{r}^1, \mathbf{v}^1, \mathbf{i}_b^1\}$

Step 2. When $k \ge 1$, the constraints (4.64) and (4.66) in problem MCO can be linearized on the solution of last iteration $\{\mathbf{r}^k, \mathbf{v}^k, \mathbf{i}_b^k\}$ thereby the convex subproblem 1 as follows

$$
\begin{aligned}
&\text{Subproblem1}: \\
&\min J = -\left(r_f^k\right)^2 - 2r_f^k\left(\mathbf{r}_f^k\right)^T(\mathbf{r}_f - \mathbf{r}_f^k) \Leftarrow J = -\left\|\mathbf{r}_f\right\|^2 \\
&s.t.\quad \dot{\mathbf{r}} = \mathbf{v}, \dot{\mathbf{v}} = -\mathbf{r}/(r^k)^3 + A(t)\mathbf{i}_b, r^k(t) = ||\mathbf{r}^k(t)|| \\
&\mathbf{h}(\mathbf{r}_f^k\mathbf{v}_f^k) + \left.\frac{\partial \mathbf{h}}{\partial \mathbf{r}_f}\right|_{\mathbf{r}_f=\mathbf{r}_f^k,\mathbf{v}_f=\mathbf{v}_f^k}(\mathbf{r}_f - \mathbf{r}_f^k) + \left.\frac{\partial \mathbf{h}}{\partial \mathbf{v}_f}\right|_{\mathbf{r}_f=\mathbf{r}_f^k,\mathbf{v}_f=\mathbf{v}_f^k}(\mathbf{v}_f - \mathbf{v}_f^k) = \mathbf{0} \\
&\mathbf{r}(t_f)^T\mathbf{i}_{h\text{ref}} = 0, \mathbf{v}(t_f)^T\mathbf{i}_{h\text{ref}} = 0 \\
&\mathbf{r}(t_0) = \mathbf{r}_0, \mathbf{v}(t_0) = \mathbf{v}_0 \\
&\|\mathbf{i}_b\| \le 1
\end{aligned}
\tag{4.76}
$$

where the square of final radius, equivalent to the maximum height, is taken as the optimization objective, i.e. $J = -\left\|\mathbf{r}_f\right\|^2$. In accordance with the convexification techniques provided by [10], the performance index J and the nonlinear equality constraints in (4.66), denoted by $\mathbf{h}(\mathbf{r}_f, \mathbf{v}_f) = 0$ in (4.76), are linearized on $\{\mathbf{r}_f^k, \mathbf{v}_f^k\}$ which represent the final position and velocity vectors of the kth iterative solution. Solving problem (4.76) yields the solution $\{\mathbf{r}^{k+1}, \mathbf{v}^{k+1}, \mathbf{i}_b^{k+1}\}$.

Step3. Compute $error = ||r^{k+1}(t) - r^k(t)||$. If $error \ge \delta$, where δ is a prescribed small positive number, then set $k = k + 1$, and go to step 2; otherwise, terminate the

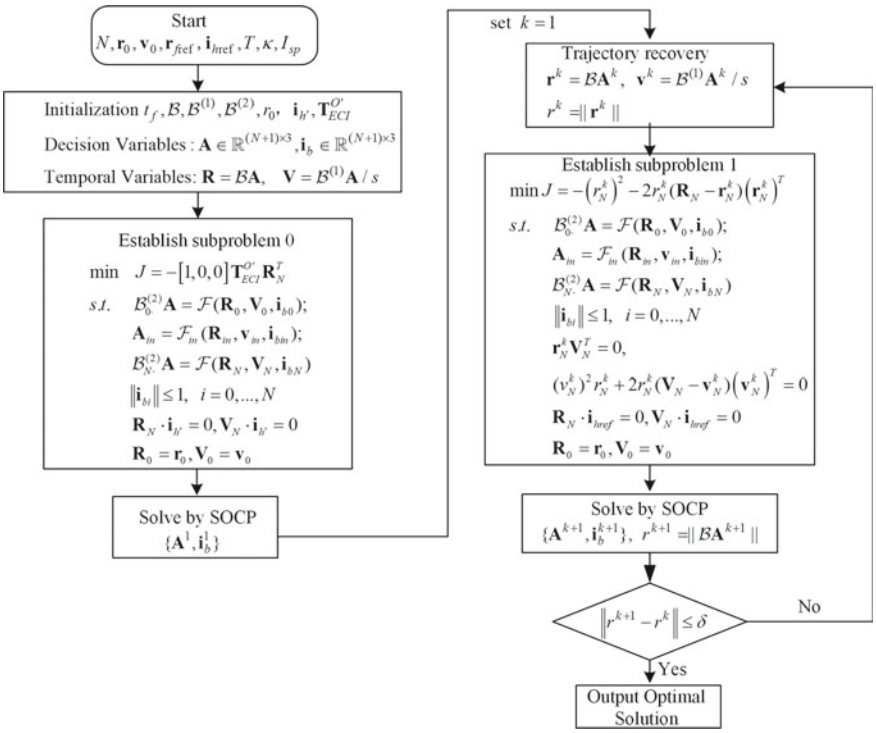


Fig. 4.3 Flow chart for solving problem MCO via BPSC2

iteration, and the optimal solution is found to be $\{\mathbf{r}^{k+1}, \mathbf{v}^{k+1}, \mathbf{i}_b^{k+1}\}$., thus the maximum circular orbital height is given by $h_{opt} = ||\mathbf{r}_f^{k+1}|| - R_E$.

The above solving procedures can be summarized as the flow chart shown in Fig. 4.3, in which, the BPSC2 method proposed in (4.58) is used to transcribe the subproblem 0 and subproblem 1. In accordance with the flow chart, one can easily use YALMIP to model and solve problem MCO.

Remark 2 In Step 1, since it is not easy to provide a three dimensional trajectory for initializing problem (4.72), only simply linear radius profile of the vehicle from the current position to the prescribed orbital insertion point is used for establishing problem (4.72). On this scenario, $\mathbf{r}_f$ and $\mathbf{v}_f$ on the final time cannot be initialized, hence the constraints (4.66), which enforce the orbital eccentricity be zero, are not considered. Further, the constraints (4.67), which confine the final orbit in the nominal orbital plane, are relaxed to confine the final states of vehicle in the current orbital plane, and this is effective demonstrated by the latter numerical results.

Remark 3 In Step 2, problem (4.76) is formulated by a very rough trajectory provided the first iteration in Step 1, in which, the terminal states $\mathbf{r}_f^1$ and $\mathbf{v}_f^1$ are hard to satisfy the constraints in (4.66), this results in that, directly linearizing (4.66) on $\mathbf{r}_f^1$ and $\mathbf{v}_f^1$ is not reasonable. Actually, the numerical studies reveal such linearizing manner cannot guarantee the convergence. Hence, the constraints (4.66) are relaxed as

$$\left(\mathbf{r}_f^k\right)^T \mathbf{v}(t_f) = 0 r_f^k v^2(t_f) - \mu = 0 \tag{4.77}$$

which enforce the final velocity vector $\mathbf{v}(t_f)$ be perpendicular to the final position vector $\mathbf{r}_f^k$ provided by the last iteration solution, meanwhile, their magnitudes satisfy with the requirement of circular orbit. Such relaxations are reasonable in the above solving procedure, because in Step 3, when the iterative termination criterion is satisfied, we have $\mathbf{r}_f^{k+1} \to \mathbf{r}_f^k$, which means the relaxed constraints in (4.77) almost are equivalent to those in (4.66). Obviously, the degree of almost equivalence depends on the iterative termination condition δ, which is set as 10_{-6} in the following numerical studies.

According to the above procedure, the proposed DPSC1, BPSC1, and BPSC2 are used to solve problem MCO, and the aforementioned preconditioned methods (P-DPSC and P-DPSC2) are not considered since their poor performance demonstrated in the last subsection. Moreover, GPOPS is adopted to obtain a baseline solution for problem MCO.

Remark 4 The thrust direction $\mathbf{i}_b$ in problem MCO is constrained by $\|\mathbf{i}_b\| = 1$, which is non-convex and relaxed to a conic constraint $\|\mathbf{i}_b\| \geq 1$ in (4.72) and (4.76). Such relaxation is similar to that in [8, 25], and it can be proved that the optimal solution obtained by solving (4.76) will satisfy $\left\|\mathbf{i}_b^*(t)\right\| = 1, t \in [t_0, t_f]$ as shown in Fig. 4.4.

In the subsequent numerical studies, the vehicle parameters are same to those in the appendix A of [21]. Without loss of generality, the thrust drop occurs in the start

of the second stage of vehicle. When the thrust failure factor, the optimized height and velocity trajectories by using GPOPS and BPSC2 are shown in Fig. 4.4, which reveals the results of two method are almost same. The results of DPSC1 and BPSC1 is similar to that of BPSC2, hence they are not presented in Fig. 4.4. DPSC2 cannot converge for any mesh grid, and DPSC1 converges only when mesh grid. It is to note that in DPSC1, the trust region constraints should be carefully selected, otherwise it cannot converge at all. Meanwhile, the trust region constraints can be discarded in BPSC1 and BPSC2, which are still converged. Figures 4.5 and 4.6, irrespectively, present the iterative procedures of DPSC1 and BPSC2, and it can be seen that BPSC2 is more efficient and accurate than DPSC1.

Table 4.3 presents the performance comparison between GPOPS, BPSC1 and BPSC2 for different thrust failure factor κ. CGL collocation scheme with $N = 40$ is adopted in Birkhoff PS methods. The resulted trajectory profiles by using BPSC2 for different κ and the nominal trajectory without any failure are presented in Fig.

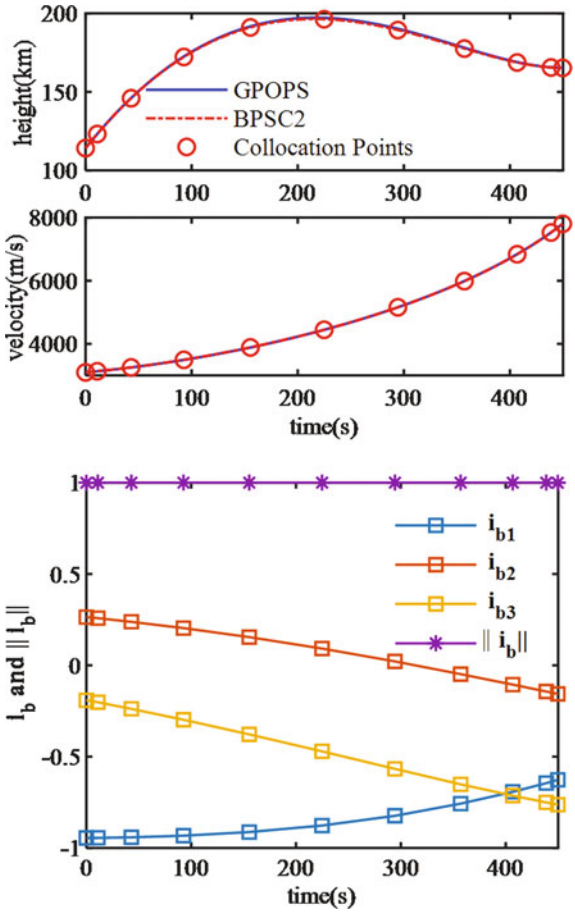


Fig. 4.4 Optimal height, velocity and control profiles by using GPOPS and BPSC2 ($N = 10, \kappa = 0.85$)

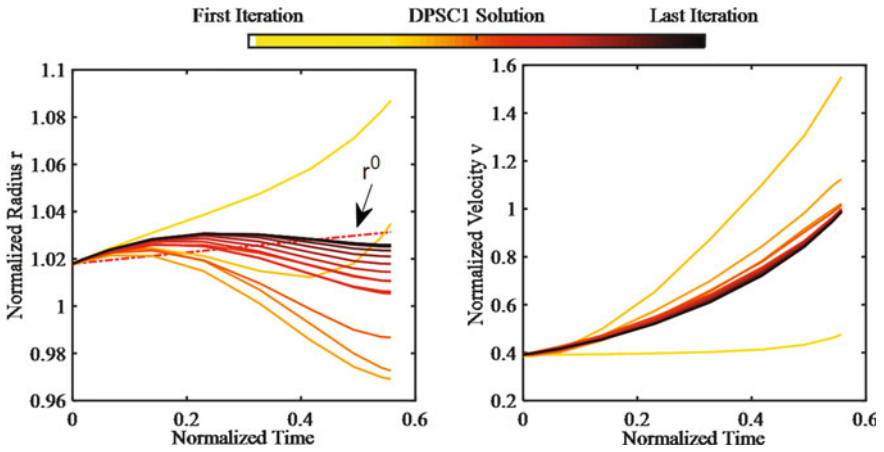


Fig. 4.5 Iterative procedure of DPSC1 ($N = 10$, $\kappa = 0.85$, 17 iterations, maximum height 165.0785 km, eccentricity 1.9445e-4)

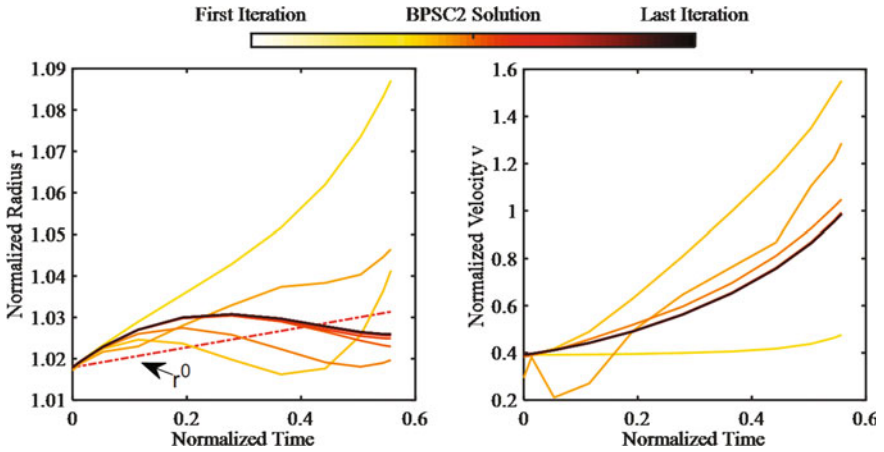


Fig. 4.6 Iterative procedure of DPSC1 ($N = 10$, $\kappa = 0.85$, 17 iterations, maximum height 165.2726 km, eccentricity 7.0068e-07)

4.7. Observing Table 4.3, for each scenario, the index of maximum height and the eccentricity provided by GPOPS are the best among three methods. The reason lies in that, GPOPS directly solves the original problem MCO, while BPSC1 and BPSC2 solve a series of convexified and relaxed problems. Particularly, the constraints on eccentricity of MCO are relaxed as (4.77) in our PS methods. However, BPSC1 and BPSC2 is more efficient than GPOPS, meanwhile, the optimized orbits with very small eccentricities are almost circular, and such performance is acceptable in practical engineering.

Table 4.3 Comparison of GPOPS, BPSC1, and BPSC2 for different thrust failure factor

κ	GPOPS			BPSC1			BPSC2		
	Maximum height(km)	Eccentricity	Solve time(s)	Maximum height(km)	Eccentricity	Solve time(s)	Maximum height(km)	Eccentricity	Solve time(s)
1.00	236.4511	2.29E-13	9.3881	236.3851	8.02E-07	0.574	236.3848	1.49E-06	0.262
0.95	222.1213	1.96E-14	3.301	222.0533	3.52E-07	0.605	222.0533	4.21E-07	0.321
0.90	200.1612	2.32E-13	2.9653	200.0895	6.97E-07	0.549	200.0895	4.55E-07	0.286
0.85	165.3954	1.02E-14	10.0759	165.3075	5.76E-07	0.799	165.3082	4.97E-07	0.288
0.80	99.0316	8.60E-15	15.721	98.7585	3.41E-07	1.18	98.7581	4.91E-07	0.313

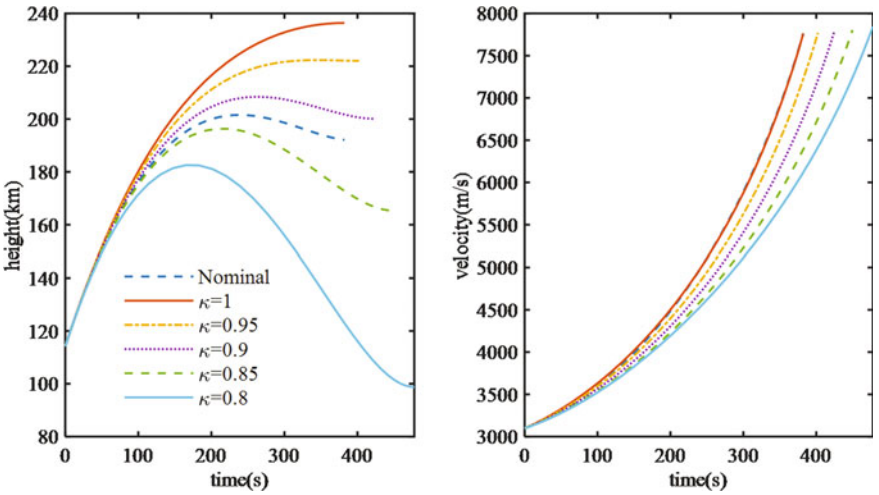


Fig. 4.7 Height and velocity profiles via BPSC2 for different thrust failure factor and the nominal trajectory

Further, the performances of BPSC1 and BPSC2 with different collocation schemes and different mesh grids are listed in Table 4.4. In general, the computational time consuming of BPSC2 is less than that of BPSC1 since the less decision variables used in BPSC2 thereby smaller scale of the transcribed SOCP problem. The growth of condition number of generated by CGL and LGL with respect to the mesh grid N is similar to Fig. 4.2, and CGL scheme is better than LGL to some degree.

Table 4.4 Comparison of BPSC1 and BPSC2 for different mesh grid

N	BPSC1					BPSC2				
	Cond.$\mathcal{A}$	Maximum height(km)	Eccentricity	Iteration number	Solve time(s)	Cond.$\mathcal{A}$	Maximum height(km)	Eccentricity	Iteration number	Solve time(s)
Chebyshev										
10	1.18E+03	165.2726	8.08E-07	16	0.032	2.65E+03	165.2726	7.01E-07	14	0.016
30	6.03E+03	165.3075	6.78E-07	15	0.157	8.03E+03	165.3069	4.85E-07	18	0.237
60	1.71E+04	165.3078	3.71E-07	16	0.998	1.61E+03	165.3077	4.98E-07	19	0.395
120	4.83E+04	165.3078	2.01E-07	14	6.379	3.23E+04	165.3077	2.07E-07	18	2.653
Legendre										
10	2.27E+03	165.215	8.04E-07	15	0.016	4.58E+03	165.2149	8.53E-07	17	0.015
30	1.97E+04	165.3072	8.29E-07	15	0.203	2.27E+04	165.3072	7.38E-07	14	0.083
60	7.81E+04	165.3078	4.55E-07	16	0.971	6.35E+04	165.3078	4.41E-07	19	0.419
120	3.11E+05	165.3079	2.23E-07	16	6.398	1.78E+05	165.3075	4.74E-07	16	2.612

4.5 Conclusions

In this chapter, Birkhoff polynomial based PS method is introduced to solve OCPs via convex programming algorithms. The constrained optimal control problem for a type of general second-order cascaded system is mainly considered. The matrix formulation of PS method with BPSIM under the convex programming framework for general OCPs is proposed, meanwhile, the solving procedure is generalized. Comparing to other PS methods using PSDM, the Birkhoff PS method renders a well-conditioned programming problem since the unique characteristics of BPSIM, which will alleviate the computational burden for the programming solvers. This advantage is remarkably demonstrated in the transcription of cascaded second-order systems, in which, the condition number of DPSC2 behaves like while that of BPSC2 like. Moreover, this allows us using more mesh points to improve accuracy of solution. From the view of application, BPSC and BPSC2 with LGL as well as CGL collocation scheme present similar performance while, however, for complicated problem, BPSC2 renders smaller scale of generated problem than that of BPSC1 while, consequently less computational time consuming. In general, for the typical cascaded second order, BPSC2 method is the best choice among the mentioned PS methods, and it can be potentially used in online trajectory optimization and guidance for ascent and recovery of launch vehicles.

References

1. J. Zhao, S. Li, Adaptive mesh refinement method for solving optimal control problems using interpolation error analysis and improved data compression. J. Frankl. Inst. **357**, 1603–1627 (2020)
2. M. Sagliano, Generalized hp pseudospectral-convex programming for powered descent and landing. J. Guid. Control Dyn. **42**, 1562–1570 (2019)
3. L.N. Trefethen, *Approximation Theory and Approximation Practice, Extended Edition*, *Society for Industrial and Applied Mathematics* (2019)
4. I.M. Ross, F. Fahroo, Pseudospectral knotting methods for solving nonsmooth optimal control problems. J. Guid. Control Dyn. **27**, 397–405 (2004)
5. G.T. Huntington, Advancement and analysis of a gauss pseudospectral transcription for optimal control problems, in *Massachusetts Institute of Technology* (2007)
6. A. Wächter, L.T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math. Program. **106**, 25–57 (2006)
7. P.E. Gill, W. Murray, M.A. Saunders, SNOPT: an SQP algorithm for large-scale constrained optimization. SIAM Rev. **47**, 99–131 (2005)
8. B. Acikmese, S.R. Ploen, Convex programming approach to powered descent guidance for Mars landing. J. Guid. Control Dyn. **30**, 1353–1366 (2007)
9. B. Açıkmeşe, J.M. Carson, L. Blackmore, Lossless convexification of nonconvex control bound and pointing constraints of the soft landing optimal control problem. IEEE Trans. Control Syst. Technol. **21**, 2104–2113 (2013)

10. X. Liu, P. Lu, Solving nonconvex optimal control problems by convex optimization. J. Guid. Control Dyn. **37**, 750–765 (2014)
11. D.-J. Zhao, Z.-Y. Song, Reentry trajectory optimization with waypoint and no-fly zone constraints using multiphase convex programming. ACTA Astronaut. **137**, 60–69 (2017)
12. C.-M. Yu, D.-J. Zhao, Y. Yang, Efficient convex optimization of reentry trajectory via the Chebyshev pseudospectral method. Int. J. Aerosp. Eng. **2019**, 1–9 (2019)
13. Y. Li, W. Chen, H. Zhou, L. Yang, Conjugate gradient method with pseudospectral collocation scheme for optimal rocket landing guidance. Aerosp. Sci. Technol. 104 (2020)
14. G. Tang, F. Jiang, J. Li, Fuel-optimal low-thrust trajectory optimization using indirect method and successive convex programming. IEEE Trans. Aerosp. Electron. Syst. **54**, 2053–2066 (2018)
15. D. Malyuta, T. Reynolds, M. Szmuk, M. Mesbahi, B. Acikmese, J.M. Carson, Discretization performance and accuracy analysis for the rocket powered descent guidance problem (2019)
16. N. Koeppen, I.M. Ross, L.C. Wilcox, R.J. Proulx, Fast mesh refinement in pseudospectral optimal control. J. Guid. Control Dyn. **42**, 711–722 (2019)
17. C. McCoid, M.R. Trummer, Preconditioning of spectral methods via Birkhoff interpolation. Numer. Algorithms **79**, 555–573 (2017)
18. L.-L. Wang, M.D. Samson, X. Zhao, A well-conditioned collocation method using a pseudospectral integration matrix. SIAM J. Sci. Comput. **36**, A907–A929 (2014)
19. I.M. Ross, R.J. Proulx, Further results on fast Birkhoff pseudospectral optimal control programming. J. Guid. Control Dyn. **42**, 2086–2092 (2019)
20. D.G. Hull, *Optimal Control Theory for Applications* (Springer, New York, 2003)
21. Z. Song, C. Wang, Q. Gong, Joint dynamic optimization of the target orbit and flight trajectory of a launch vehicle based on state-triggered indices. Acta Astronaut. **174**, 82–93 (2020)
22. L. Blackmore, B. Açikmeşe, D.P. Scharf, Minimum-landing-error powered-descent guidance for mars landing using convex optimization. J. Guid. Control Dyn. **33**, 1161–1171 (2010)
23. Y. Li, B. Pang, C. Wei, N. Cui, Y. Liu, Online trajectory optimization for power system fault of launch vehicles via convex programming. Aerosp. Sci. Technol. **98**, 105682 (2020)
24. D. Morgan, S.-J. Chung, F.Y. Hadaegh, Model predictive control of swarms of spacecraft using sequential convex programming. J. Guid. Control Dyn. **37**, 1725–1740 (2014)
25. X. Liu, Z. Shen, P. Lu, Entry trajectory optimization by second-order cone programming. J. Guid. Control. Dyn. (Artic. Adv.) **39**, 227–241 (2016)
26. M. Szmuk, T.P. Reynolds, B. Açıkmeşe, Successive convexification for real-time six-degree-of-freedom powered descent guidance with state-triggered constraints. J. Guid. Control Dyn. **43**, 1399–1413 (2020)
27. A. Domahidi, E. Chu, S. Boyd, ECOS: an SOCP solver for embedded systems, in *European Control Conference, Zurich Switzerland* (2013)
28. E.D. Andersen, C. Roos, T. Terlaky, On implementing a primal-dual interior-point method for conic quadratic optimization. Math. Program. **95**, 249–277 (2003)
29. P.T. Boggs, J.W. Tolle, Sequential quadratic programming. Acta Numer. **4**, 1–51 (1995)
30. F. Fahroo, I.M. Ross, Advances in pseudospectral methods for optimal control, in *AIAA Guidance, Navigation and Control Conference and Exhibit* (2008)
31. J. Shen, T. Tang, L.L. Wang, *Spectral Methods: Algorithms Analysis and Applications* (Springer, Berlin, 2011)
32. X. Guo, M. Zhu, Direct trajectory optimization based on a mapped Chebyshev pseudospectral method. Chin. J. Aeronaut **26**, 401–412 (2013)
33. D. Benson, A Gauss pseudospectral transcription for optimal control, in *Massachusetts Institute of Technology* (2005)
34. R. Baltensperger, J.P. Berrut, The errors in calculating the pseudospectral differentiation matrices for Chebyshev-Gauss-Lobatto points. Comput. Math. Appl. **37**, 41–48 (1999)
35. Y.G. Shi, *Theory of Birkhoff interpolation* (Nova Science Publishers, New York, 2003)
36. J. Löfberg, YALMIP: a toolbox for modeling and optimization in MATLAB2004, in *Proceedings of the CACSD Conference,Taipei Taiwan* (2004)

37. A. Domahidi, Methods and tools for embedded optimization and control, in*ETH ZURICH* (2013)
38. B.A. Conway, K.M. Larson, Collocation versus differential inclusion in direct optimization. J. Guid. Control Dyn. **21**, 780–786 (1998)

Chapter 5
Autonomous Descent Guidance via Sequential Pseudospectral Convex Programming

Marco Sagliano, David Seelbinder, and Stephan Theil

5.1 Introduction

The last ten years have been disruptive for rocket technology. We are witnessing a paradigm shift which has its focus on reusability, a dream pursued since the beginning of the Space Shuttle program [8], but that only now we are able to fully see as weekly-based, operative technology. This is mainly the result of SpaceX efforts. The company led by Elon Musk paved the way for a deep reshaping of the conception of rockets, mainly with their Falcon 9 program, able, at the moment that this chapter is getting written, to successfully complete its 100th landing [25]. The concurrent development of the even more ambitious Starship program [26], together with the efforts of other players, such as Rocket Lab with its Neutron [9] and Blue Origin with the New Glenn rocket [16] confirms that the disruption we are experiencing is irreversible, and needs to be embraced rather than feared. With this spirit agencies and intergovernmental institutions are updating their plans to keep the pace of the private sector.

In this complex scenario the German Aerospace Center (DLR), the Japan Aerospace Exploration Agency (JAXA), and the French National Centre for Space Studies (CNES) decided to join their resources and know-how in a trilateral agreement aiming at developing and demonstrating the technologies that will be needed for future reusable launch vehicles. The agreement led to the CALLISTO project (Cooperative Action Leading to Launcher Innovation in Stage Toss back Operations) [7], whose demonstrator is currently in development. Its objective is to develop and improve all the critical technologies that are required for making reusable launch

M. Sagliano (✉)
Senior GNC Researcher, German Aerospace Center, Bremen, Germany
e-mail: Marco.Sagliano@dlr.de

D. Seelbinder
Guidance and Control Group Leader, German Aerospace Center, Bremen, Germany
e-mail: David.Seelbinder@dlr.de

S. Theil
GNC Department Head, German Aerospace Center, Bremen, Germany
e-mail: Stephan.Theil@dlr.de

Z. Song et al. (eds.), *Autonomous Trajectory Planning and Guidance Control for Launch Vehicles*, Springer Series in Astrophysics and Cosmology,
https://doi.org/10.1007/978-981-99-0613-0_5

systems operative at industrial level in the next decade. The CALLISTO project will culminate in a series of flights that will performed from the Kourou Space Center (KSC), in French Guiana.

To maximize the know-how return of each partner it was decided to have two parallel lines of development for the Guidance and Control (G&C) subsystem. One will be developed by CNES, whereas DLR and JAXA decided to strengthen their efforts and develop a unique, fully integrated G&C solution [23]. Since the focus is to demonstrate reusability technologies for an end-to-end scenario the mission profile consists of multiple flight phases, corresponding to different aerodynamic configurations of the vehicle and different actuation capabilities. Specifically, we identify four main phases of flight, which correspond to different G&C strategies: the ascent phase, the boostback maneuver, the aerodynamic phase, and the powered descent and landing phase. Consequently, several algorithms are required to cope with each of the phases to be able to successfully and autonomously complete such an ambitious mission.

This chapter focuses on the guidance strategy applicable to the last two phases, namely the aerodynamic descent phase and the powered landing phase. As a matter of fact it is well-known that the non-powered, aerodynamically guided phase is critical for the error management in terms of position and velocity [3]. Moreover, large uncertainties due to both the atmosphere and the aerodynamic properties of the vehicle affect the resulting trajectory. Lastly, there will be errors coming from previous segments of flight that the G&C subsystem has to compensate for. All these aspects make the aerodynamic guidance a challenging problem, which requires the capability to generate valid solutions rapidly, in reliable way, and in case significant off-nominal conditions are experienced during the mission, these need be taken into account. On the other hand the powered landing phase requires high accuracy and a perfect coordination of thrust, position, velocity and attitude to meet the strict requirements allowing for a safe and accurate touchdown, the so-called *pinpoint landing* [4].

Given the aforementioned reasons, the problem of generating valid guidance solutions in the frame of Entry, Descent, and Landing (EDL) has gained great attention, and multiple research groups and companies have worked on the subject. In many different solutions the key technology is represented by the use of Convex Optimization [6], a sub-branch of Numerical Optimization characterized by several intriguing properties, including the guarantee to find a solution if there exists one, a limited dependency on initial guesses, and the computation rapidness, due to state-of-the-art interior point primal-dual solvers [2]. In the specific frame of EDL large attention was dedicated to the application of Second-Order Conic Programming (SOCP), a specific subset of convex optimization, in which all the inequality constraints are formulated in linear or conic form.

Among the proposed methodologies a break-through was represented by the formulation of the entry problem in the energy domain [11]. In this case the non-convex constraints were transformed into upper and lower bounds on the altitude, by expressing the speed as a function of the energy. Moreover, to overcome the non-convexity intrinsically associated with the bank angle σ, two new controls, defined as the sine

and the cosine of the bank angle, were adopted. The substitution was then made valid by ensuring that the identity $\sin\sigma^2 + \cos\sigma^2 = 1$ was satisfied. Moreover, this formulation overcomes the difficulties of having free-final time, since the final value of energy is automatically obtained by the corresponding final altitude and speed. With the idea of retrieving the benefits of the Space Shuttle Entry Guidance, and generalizing it through the use of convex-optimization technologies, a drag-energy approach based on the application of pseudospectral methods was proposed [22]. In this case a valid drag-energy profile was computed by reformulating the problem in terms of inverse of drag acceleration, and the solution was mapped against longitudinal states to obtain a complete guidance solution. An interesting approach was also formulated by Wang and Grant by exploiting second-order conic programming [29]. In this case the problem was directly transcribed in the time domain by using a direct linearization approach of the nonlinear equations underlying the problem. Wang and Lu further improved the method [30] by means of line-search and trust region techniques that were introduced to speed up the convergence process. The previous approaches were mainly applied to VTHL (Vertical Take-off, Horizontal Landing) vehicles.

For what regards VTVL (Vertical Take-off, Vertical Landing) rockets, the landing phase was extensively treated in the last years, starting from the seminal work of Acikmese et al. [1], and this research area is still very active now, with multiple applications of Convex Optimization [4], Successive Convex Optimization [27, 28], and Pseudospectral Convex Optimization in its standard and generalized forms [18, 20, 21]. Moreover, a first, successful attempt to combine aerodynamic and propulsive control was also proposed by Xinfu Liu [10], where the problem was reduced to two dimensions, and a new set of variables, needed to convexify the subproblem, was introduced. Yang and Liu also proposed to use altitude as independent variable to be able to deal with free-final time powered descent problems [31], through the corresponding manipulation of the equations of motion.

As pointed out by the authors in this last work the methods to deal with non-convex approach through convex techniques can be mainly divided into *Direct Linearization Approach* and *Nonlinearity-Kept and Linearization Approach*. In the former the equations of motion, as well as the constraints and the cost function are directly obtained by linearizing the problem around the solution found at the previous iteration. The validity of the approach is ensured by an ad-hoc choice of the static trust regions radii. For this class of methods a drawback can be a slow convergence, and further strategies might be needed to improve the quality of the process [30]. Note that in these approaches the authors introduce some variable transformations aiming at convexifying the problem while keeping nonlinear features of the original formulation. The difficulties in this case arise because of the peculiarity of the transformations needed (which strongly depend on the nature of the problem), the assumptions required to ensure that some simplifications and transformations are valid, and the difficulty to generalize the method (for example it can be hard to extend to the 3-D case the transformations obtained for the in-plane scenario).

In this work we propose a third approach, meant as middle-ground between the two aforementioned techniques. There are four main novelties associated with this work.

First, we endorse an ad-hoc formulation of the equations of motion which minimizes the presence of non-convex terms. For example, by using Cartesian representation for position and velocity we avoid the trigonometric terms appearing in the equations of motion for longitude and latitude, which are typically used for aerodynamic entry trajectories [15]. Second, we introduce a different parametrization of controls, based on Euler-angle rates defined with respect to the target-centered Altitude-Crossrange-Downrange directions, in addition to the thrust-rate already adopted in literature. This choice allows to maximize the presence of linear terms in our differential equations while having the controls appearing in affine form, a property which simplifies the convergence process, as pointed out by Liu et al. [12]. Moreover the proposed approach gives us the chance to explicitly limit control rates as well, leading to smooth solutions, and therefore to trajectories which can be more easily tracked by the attitude controller.

The third novelty relies on entirely convoying the nonlinearities into the terms appearing in the differential equations representing the accelerations for the aerodynamic phase and the mass rate. We consequently apply numerical linearization only to these terms, obtaining therefore an hybrid computation of the matrix representing the equation of motion in linearized form.

The fourth aspect we want to emphasize is the use of a systematic transcription based on generalized hp pseudospectral methods, already adopted for the powered landing problem [20], but here extended to deal with the aerodynamic phase problem too. This choice benefits from the properties of pseudospectral methods [19], such as the quasi-exponential (or *spectral*) convergence, and the easiness of implementation. Moreover, its mapping between time and pseudotime is conveniently exploited to formulate the free-final time problem, leading to an increased capability of handling initial dispersions, since we don't need any accurate a-priori knowledge or estimate of the flight time for off-nominal cases.

A last remark concerns the non-convex accelerations, with special focus on the aerodynamic terms: it is in fact worth mentioning that in related works some simplifications (e.g., constant drag coefficient) are typically made, given their relative importance. However, in this work we focus on conditions which are as close as possible to what the vehicle in a real scenario will experience. Therefore we reject these simplifications typically used for this class of methods, (e.g., constant gravity, negligible lift, or constant drag coefficient). Instead, we use a full-blown aerodynamic database, which includes lift, drag, side-force, as well as aerodynamic torques, which depend on the 3-D attitude of the vehicle as well as on the Mach number [14]. These assumptions are required given the centrality of the aerodynamic accelerations for this type of problems.

Numerical results are shown for a CALLISTO-class rocket. We extend our recent results [21] by also analyzing off-nominal conditions for both the flight phases. The chapter is organized as follows: in Sect. 5.2 the mission and the vehicle are briefly described, while Sect. 5.3 focuses on the problem formulation in continuous form. Sections 5.4 and 5.5 describe the convexified approach and its corresponding pseudospectral transcription, respectively, while numerical results are described in Sect. 5.6. Finally, we draw some conclusions in Sect. 5.7.

5.2 Mission and Vehicle

In this section we provide an overview of the mission scenario as well as of the vehicle.

5.2.1 Vehicle and Mission Overview

As mentioned in the introduction the vehicle considered in this work is a 40-kN class rocket. The rocket thrust can be throttled between 40 and 110% of its nominal maximum value, and the engine is mounted on a gimballed system able to provide pitch and yaw control capability during the powered phases, namely ascent and landing. Roll control is ensured by a set of eight reaction control system (RCS) thrusters, mounted on top of the rocket. During the unpowered phases we can rely on a set of four steerable fins mounted on top. They are able to provide complete aerodynamic control, ensuring full control throughout the entire mission. An impression of the rocket used as example for this work is depicted in Fig. 5.1. The mission considered is a Return To Launch Site (RTLS) Scenario. This means that the rocket will fly back and perform the landing onto a platform that is very close to the launch site, as visible in Fig. 5.1. A series of flights will take place at the Guiana Space Center, the European Spaceport in French Guiana. This flight campaign will give indications on the level of refurbishment required between two consecutive flights performed with the same vehicle, while providing first-hand data to all the partners to further enhance the knowledge of reusable technologies and some of the related critical technologies, especially in terms of Guidance, Navigation and Control domain.

5.2.2 Rocket Modeling

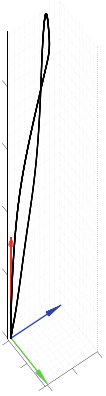
While mass and center of mass (CoM) are considered constant during the aerodynamic descent, during the powered phase the vehicle experiences a significant variation of both these variables. This effect is accounted for by storing the CoM as a lookup-table depending on the current mass. Mass variations coming from RCS are in this context neglected. Therefore, we can express this dependency as

$$CoM = CoM(m) \tag{5.1}$$

The axis-symmetry of the vehicle has been exploited to compute aerodynamic coefficients as function of an horizontal and a vertical angle of attack α_1 and α_2, as illustrated in Fig. 5.2. The aerodynamic force coefficients with respect to the body axes

$$\mathbf{C}^{aero}_{BODY} = [C^{aero}_{BODY,x}\ C^{aero}_{BODY,y}\ C^{aero}_{BODY,z}]^T$$

(a) CALLISTO experimental vehicle



(b) Return-to-Launch-Site mission profile

Fig. 5.1 Mission and vehicle overview: **a** CALLISTO rocket, **b** Reference mission profile

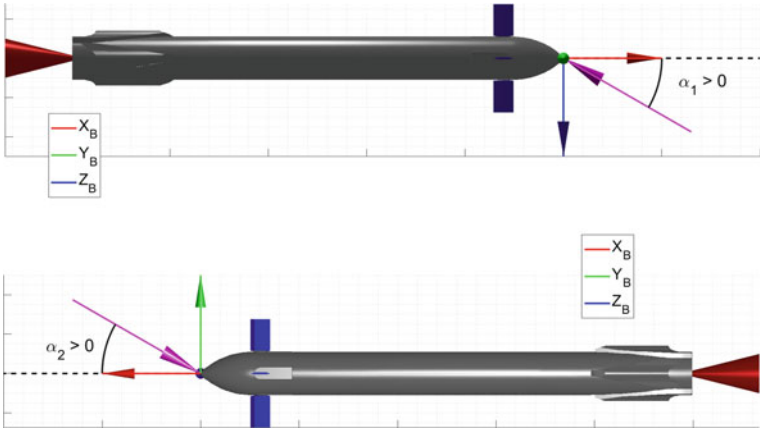


Fig. 5.2 Vertical and horizontal angles of attack

are provided as multidimensional look-up tables, which depend on Mach number M, and angles of attack α_1 and α_2, and on the four fin deflections δ_1,δ_2,δ_3,δ_4.

$$\mathbf{C}^{aero}_{BODY} = \begin{bmatrix} C^{aero}_{BODY,x}(M, \alpha_1, \alpha_2, \delta_1, \delta_2, \delta_3, \delta_4) \\ C^{aero}_{BODY,y}(M, \alpha_1, \alpha_2, \delta_1, \delta_2, \delta_3, \delta_4) \\ C^{aero}_{BODY,z}(M, \alpha_1, \alpha_2, \delta_1, \delta_2, \delta_3, \delta_4) \end{bmatrix} \tag{5.2}$$

From these coefficients the aerodynamic force can be computed as

$$\mathbf{F}^{aero}_{BODY,i} = \tfrac{1}{2}\rho V^2 S \mathbf{C}^{aero}_{BODY,i}\ , \quad i = x, y, z \tag{5.3}$$

with the reference surface S equal to 0.95 m. The term ρ represents the atmospheric density, which depends on altitude, whereas V is the speed of the vehicle with respect to the air. For what regards the atmospheric density we employ a model coming from experimental measures, and provided as look-up table, where the geodetic altitude is the independent variable.

$$\rho = \rho(h)$$

This choice confirms once more that no strong simplifications (i.e., exponential atmospheric profile) are required with the proposed method.

Remark 1: Note that, although this solution is inherently based on a 3-DOF model we are interested to generate solutions which we define *6-DOF capable*, which means the generated trajectory has to provide an attitude which can be trimmed by the aerodynamic fins. This aspect is currently included in the present work, as it will be shown in Sect. 5.6.

5.3 Problem Formulation

In this section we describe in detail the problem formulation for both the aerodynamic descent and the powered landing phases of the flight.

Aerodynamic Descent

A. Equations of Motion

We describe the aerodynamic guidance problem in a target-centered Downrange-Crossrange-Altitude (DCA) reference frame, as depicted in Fig. 5.1. This reference frame can be thought of as a Up-East-North local reference frame rotated around the x-axis to align the z-axis with the plane containing most of the trajectory. We cannot rely on thrust during this phase, and therefore only the aerodynamic forces

can be used to control the vehicle. These forces are a direct function of the speed and the relative attitude of the body axes with respect to the airflow. Therefore the attitude is implicitly the main way to control the rocket, modeled in this work as a 3-DOF point mass. The desired attitude will then define the reference signals to be tracked by the attitude controller, which is realized by using the fins to generate the desired aerodynamic torques. The translational dynamics equations can be described as follows. Note that from now on the reference frame indication is omitted for brevity.

$$\begin{aligned} \dot{\mathbf{r}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= \mathbf{a}^{grav} + \mathbf{a}^{aero} - 2\omega \times \mathbf{v} - \omega \times (\omega \times \mathbf{r}) \\ \dot{\theta} &= u_\theta \\ \dot{\psi} &= u_\psi \end{aligned} \tag{5.4}$$

The terms $\mathbf{r}$ and $\mathbf{v}$ are the position and the velocity of the CoM of the vehicle expressed in the DCA reference frame. We include non-inertial terms due to the rotation of the Earth ω, whereas θ and ψ are the pitch and yaw angle of the rocket with respect to the target-centered Downrange-Crossrange-Altitude reference frame. We assume that the roll angle is kept constant during the descent to maximize the decoupling between pitch and yaw axes. Note that the controls we effectively use are the pitch rate u_θ and the yaw rate u_ψ. This choice is twofold beneficial: first, it decouples the control matrix from the states, as we will see. Second, it allows to impose explicit bounds on the control rates, making the solution smoother. The sources of non-convexity are in this case two: the gravity, which is a non-linear function of position, and the aerodynamic accelerations, which depend nonlinearly on altitude, velocity, and attitude angles. For what regards the gravity we can assume a central-body model,

$$\mathbf{a}^{grav} = -\mu_\oplus \frac{\mathbf{r} + \mathbf{r}_T}{\|\mathbf{r} + \mathbf{r}_T\|_2^3} \tag{5.5}$$

with $\mu_\oplus$ representing the gravitational parameter of the Earth, while $\mathbf{r}_T$ is the position vector of the target site with respect to the center of the Earth. Note that with this formulation more accurate models, like the one based on the World Geodetic System 84, could be adopted. However, this more advanced modeling is kept for future development, and a simpler choice was here preferred. For what regards the aerodynamic accelerations they represent a nonlinear combination of the states. In fact, the aerodynamic accelerations are, for the rocket under analysis, expressed in body-reference frame as

$$\mathbf{a}_{BODY}^{aero} = \frac{1}{m}\mathbf{F}_{BODY}^{aero}(\alpha_1, \alpha_2, M, q_{dyn}, \delta_1, \ldots, \delta_4) \tag{5.6}$$

where α_1 and α_2 are the vertical and horizontal angle of attack (introduced in Fig. 5.2 and used to exploit the axis-symmetry of the vehicle) M represents the Mach number, while q_{dyn} is the dynamic pressure, function of the altitude (through the atmospheric density ρ) and the speed V.

$$q_{dyn} = \frac{1}{2}\rho V^2 \tag{5.7}$$

Since we are interested to express the aerodynamic accelerations in DCA coordinates it is necessary to transform the outcome of Eq. (5.6) as

$$\mathbf{a}^{aero}_{DCA} = \mathbf{R}^{DCA}_{BODY} \cdot \mathbf{a}^{aero}_{BODY} \tag{5.8}$$

where $\mathbf{R}^{DCA}_{BODY}$ represents the rotation matrix from $BODY$ to DCA. This matrix is composed by two different contributions.

$$\mathbf{R}^{DCA}_{BODY} = \mathbf{R}^{DCA}_{UEN} \cdot \mathbf{R}^{UEN}_{BODY} \tag{5.9}$$

with UEN representing the target-fixed Up-East-North reference frame. Note that the first term of the right hand side of Eq. (5.9) is only function of the target position $\mathbf{r}_T$ and of the angle χ identifying the trajectory plane. Both are constant,

$$\mathbf{R}^{DCA}_{UEN} = \mathbf{R}^{DCA}_{UEN}(\mathbf{r}_T, \chi) \tag{5.10}$$

while the second contribution is a direct function of the attitude of the body. In fact, we can write

$$\mathbf{R}^{UEN}_{BODY} = \mathbf{R}^{UEN}_{BODY}(\theta, \psi) \tag{5.11}$$

and this relationship embeds part of the nonlinearities requiring linearization. Note that the derivation of the aerodynamic accelerations in DCA is not only necessary for the formulation of the equations of motion, but is also useful because it gives us an indication of the dependencies to be considered when the linearized equations of motion are derived.

B. Boundary Conditions

The problem will have fixed initial and final conditions, coming from previous and successive phases of flight,

$$\begin{bmatrix} \mathbf{r}_{DCA}(t_0) \\ \mathbf{v}_{DCA}(t_0) \\ \theta_{DCA}(t_0) \\ \psi_{DCA}(t_0) \\ \mathbf{r}_{DCA}(t_F) \\ \mathbf{v}_{DCA}(t_F) \\ \theta_{DCA}(t_F) \\ \psi_{DCA}(t_F) \end{bmatrix} = \begin{bmatrix} \mathbf{r}_{DCA,0} \\ \mathbf{v}_{DCA,0} \\ \theta_{DCA,0} \\ \psi_{DCA,0} \\ \mathbf{r}_{DCA,F} \\ \mathbf{v}_{DCA,F} \\ \theta_{DCA,F} \\ \psi_{DCA,F} \end{bmatrix} \tag{5.12}$$

as we want, at the end of the aerodynamic descent (occurring at the free final time t_F) to be in conditions of correctly switching to the powered phase in optimal conditions for the pinpoint landing.

Remark 2: Note that the Euler angles adopted throughout this work are built on the Up-East-North convention, and not on the traditional North-East-Down. This choice is motivated by the need to avoid the classical singularity of the pitch angle at 90°, which is what would happen for a vertical descending vehicle. With the adopted convention we ensure to be away from the singularity, as the vertical descent is associated with a pitch angle $\theta = 0°$.

C. Constraints

For the aerodynamic phase no nonlinear constraints were taken into account in this specific example, even though it is possible to do it by directly linearizing them ([21, 29]).

D. States and controls bounds

Finally, we want to have a meaningful upper bound and lower bound on the states,

$$\begin{bmatrix} r_{xL} \\ r_{yL} \\ r_{zL} \\ v_{xL} \\ v_{yL} \\ v_{zL} \\ \theta_L \\ \psi_L \end{bmatrix} \leq \begin{bmatrix} r_x \\ r_y \\ r_z \\ v_x \\ v_y \\ v_z \\ \theta \\ \psi \end{bmatrix} \leq \begin{bmatrix} r_{xU} \\ r_{yU} \\ r_{zU} \\ v_{xU} \\ v_{yU} \\ v_{zU} \\ \theta_U \\ \psi_U \end{bmatrix} \tag{5.13}$$

which derive mainly from flight-safety studies. Moreover, to limit the closed-loop bandwidth associated with the attitude controller, and generate a smooth solution, upper and lower bounds are also assigned to the rates of pitch and yaw angles, with $u_{\theta,max} = u_{\psi,max} = 10°/\text{s}$.

$$\begin{bmatrix} -u_{\theta,max} \\ -u_{\psi,max} \end{bmatrix} \leq \begin{bmatrix} u_\theta \\ u_\psi \end{bmatrix} \leq \begin{bmatrix} u_{\theta,max} \\ u_{\psi,max} \end{bmatrix} \tag{5.14}$$

E. Cost function

Finally, for this problem we are interested to minimize the control activity, therefore we simply express the cost function as

$$J = \int_{t_0}^{t_F} \left[u_\theta^2(t) + u_\psi^2(t) \right] dt \tag{5.15}$$

where Eq. (5.15) will be weighted by a user-defined positive weight w_u, whose value is formally irrelevant here, but becomes important in the construction of an

augmented cost function that takes also other effects into account, as it will be shown in Sects. 5.4 and 5.5. The problem to be solved is therefore the following: we aim at minimizing Eq. (5.15) with the system subject to the differential equations defined in Eq. (5.4). The solution has to satisfy the boundary conditions given by Eq. (5.12), as well as states and control box constraints defined according to Eqs. (5.13) and (5.14).

Powered Landing

A. Equations of motion

We can extend the previous formulation to the powered landing problem. Several elegant formulations have been proposed over the years to deal with this problem. Here we aim at including the presence of realistic effects, which in order of relevance are (1) the thrust-aerodynamic forces interaction, (2) the minimization of aerodynamic torques that could prevent the 6-DOF feasibility of the trajectory, (3) the motion of the center of mass while descending, (4) the effect of the pressure on the effective thrust generated, and 5) other effects, like non-constant gravitational acceleration and non-inertial forces due to the rotation of the Earth. While in the previous subsection we were using the aerodynamic forces as means of control in this case their effect is combined with the force exerted by the engine to dominate the motion of the rocket. The corresponding model is a 3-DOF point having variable mass, and its evolution is described by the following set of equations:

$$\begin{aligned}
\dot{\mathbf{r}} &= \mathbf{v} \\
\dot{\mathbf{v}} &= \mathbf{a}^{thr} + \mathbf{a}^{grav} + \mathbf{a}^{aero} - 2\omega \times \mathbf{v} - \omega \times (\omega \times \mathbf{r}) \\
\dot{m} &= -\tfrac{T_{vac}}{I_{sp} g_0} \\
\dot{\phi} &= u_\phi \\
\dot{\theta} &= u_\theta \\
\dot{\psi} &= u_\psi \\
\dot{T}_{atmo} &= u_T
\end{aligned} \tag{5.16}$$

Note the presence of the roll angle ϕ, the mass m and the atmospheric thrust T_{atmo}, with the last two terms linked to the vacuum thrust T_{vac} through the equation

$$T_{atmo} = T_{vac} - A_{nz} p \tag{5.17}$$

with A_{nz} indicating the nozzle area, p the atmospheric pressure, and T_{vac} the thrust generated in vacuum. The controls are for this scenario the Euler angle rates u_ϕ, u_θ, u_ψ. Moreover, to limit the instantaneous change of thrust, not compatible with the physical rocket engine, we include the thrust rate u_T. This choice gives the chance to decouple the control matrix from the states as done for the aerodynamic descent, and

at the same time to obtain solution physically realizable by the rocket's actuators. For this problem there are four different sources of non-convexity: in addition to the aforementioned gravity and aerodynamic accelerations we have now the acceleration caused by the thrust $\mathbf{a}_{thr} = \mathbf{T}_{atmo}/m$. Moreover, there is an exponential-like dependence on the massflow from the altitude through the pressure in virtue of Eq. (5.17). The aerodynamic accelerations can be computed again by invoking Eq. (5.6) and we can convert them into their DCA representation exactly as done through Eqs. (5.8)–(5.10). The only difference is represented by the modification of Eq. (5.11), which, in virtue of the dependence on the roll angle ϕ is re-written as follows.

$$\mathbf{R}_{BODY}^{UEN} = \mathbf{R}_{BODY}^{UEN}(\phi, \theta, \psi) \tag{5.18}$$

B. Boundary Conditions

The boundary conditions described in Eqs. (5.12) are still valid. We augment them with some further conditions coming from the new variables included in the problem as follows.

$$\begin{bmatrix} m(t_0) \\ \phi_{DCA}(t_0) \\ T(t_0) \\ \phi_{DCA}(t_F) \end{bmatrix} = \begin{bmatrix} m_0 \\ \phi_{DCA,0} \\ T_0 \\ \phi_{DCA,F} \end{bmatrix} \tag{5.19}$$

Note that we omitted the final value of mass and thrust, as they are determined by the algorithm. Moreover, we include final conditions for the attitude to ensure that the vehicle lands with its x-body axis being normal to the local horizontal plane.

C. Constraints

Three types of constraints are included here: first, we introduce the classical glideslope constraint to enforce the vehicle to impose a controlled ratio between reduction of horizontal and vertical distance with respect to the landing spot.

$$\frac{r_A}{\left\|\mathbf{r}_{D,C}\right\|} \geq \tan \gamma_{gs} \tag{5.20}$$

with glideslope angle equal to 70°. To further enforce a vertical motion towards the end of the pinpoint landing sequence it is imposed that in the last segment of the trajectory, approximately corresponding to the last 5 s of flight, both side-components of position and velocity are bounded, and specifically

$$\begin{matrix} \left\|\mathbf{r}_{D,C}(t \leq t_F - t^*)\right\| \leq r_{D,C,max} \\ \left\|\mathbf{v}_{D,C}(t \leq t_F - t^*)\right\| \leq v_{D,C,max} \end{matrix} \tag{5.21}$$

with t^* equal to 5 s, $r_{D,C,max}$ equal to 1 m, and $v_{D,C,max}$ defined as 0.1 m/s.

Note that all these constraints can be modeled as second-order conic constraints, and therefore do not require linearization.

D. States and controls bounds

While Eqs. (5.13)–(5.14) still hold for the powered phase too, we augment them according to the problem definition by adding the following box constraints for the states,

$$\begin{bmatrix} m_L \\ \phi_L \\ T_L \end{bmatrix} \leq \begin{bmatrix} m \\ \phi \\ T \end{bmatrix} \leq \begin{bmatrix} m_u \\ \phi_U \\ T_U \end{bmatrix} \tag{5.22}$$

and the controls

$$\begin{bmatrix} -u_{\phi,max} \\ -u_{T,max} \end{bmatrix} \leq \begin{bmatrix} u_\phi \\ u_T \end{bmatrix} \leq \begin{bmatrix} u_{\phi,max} \\ u_{T,max} \end{bmatrix} \tag{5.23}$$

with $u_{\theta,max} = u_{\psi,max} = 5°/s$, while the roll rate is limited to $0.1°/s$. Finally, a maximum throttle rate $u_{T,max}$ here normalized, is included in the formulation. Finally, note that with the current formulation the thrust vector inclination is not constrained, as we are relying on a 3-DoF formulation. However, given the constraint on the final body axes the thrust tilt angle is implicitly constrained to be within $\pm\delta_{TVC,max}$ with respect to the local vertical axes at touchdown, where $\delta_{TVC,max}$ is the maximum thrust gimbal angle.

E. Cost function

The cost function we build for this problem is made of different contributions: first, we are interested to maximize the final mass of the vehicle, which corresponds to the minimization of the fuel required to perform the landing maneuver. Moreover, we introduce a penalization of the control rates to ensure that the solution we obtain is smooth enough.

$$J = -w_m m(t_F) + w_u \int_{t_0}^{t_F} \left(\mathbf{u}^T \cdot \mathbf{R} \cdot \mathbf{u}\right) dt \tag{5.24}$$

The vector $\mathbf{u}$ embeds all the four controls included in the formulation through $\mathbf{R}$, defined as a unitary diagonal matrix, while the terms w_m and w_u measure the relative importance of the two terms in the optimization process, with the former not excessively larger than the latter. This choice is motivated by the fact that while we are interested to optimize the fuel consumption, we also want to discourage through the presence of the term w_u large variations of the control rates. In fact, larger control variations could lead to hectic control profiles, which might be slightly more efficient from the fuel-consumption perspective, but less safe. We have therefore completely defined the problem to be solved: we aim at minimizing Eq. (5.24) with the system

subject to the differential equations defined in Eq. (5.16). The solution has to satisfy the boundary conditions given by Eqs. (5.12), (5.19) and (5.13)–(5.14), (5.22)–(5.23) as well as the constraints of Eqs. (5.20) and (5.21).

5.4 Convex Formulation

In this section we will transform the two continuous problems described in Sect. 5.3 into a sequence of convex problems, to be solved iteratively.

Aerodynamic Descent

A. Equations of motion
For what regards the equations of motion during the aerodynamic phase we can decompose the system described in Eq. (5.4) in a convex part, and a non-convex part. Defined the state vector as

$$\mathbf{x} = \left[\mathbf{r}_{DCA} \ \mathbf{v}_{DCA} \ \theta \ \psi \right]^T \tag{5.25}$$

we can write the equations of motion as

$$\dot{\mathbf{x}} = \mathbf{f}_{nc}(\mathbf{x}) + \mathbf{f}_c(\mathbf{x}) + \mathbf{B}\mathbf{u} + \mathbf{C}\nu \tag{5.26}$$

with

$$\mathbf{u} = \left[u_\theta \ u_\psi \right]^T \tag{5.27}$$

representing the physical controls used to manipulate the attitude of the vehicle, and consequently, the aerodynamic forces generated, while the vector $\nu \in \mathbb{R}^{n_s}$, defined as

$$\mathbf{v} = \left[\nu_{r_x} \ \nu_{r_y} \ \nu_{r_z} \ \nu_{v_x} \ \nu_{v_y} \ \nu_{v_z} \ \nu_\theta \ \nu_\psi \right]^T \tag{5.28}$$

represents the virtual controls, required to avoid artificial infeasibility [28]. The matrix $\mathbf{C}$ is a design parameter to decide which and how many virtual controls will be used to help the convergence process. For this work the matrix is defined as

$$\mathbf{C} = \begin{bmatrix} \mathbf{I}_{6\times 6} & \mathbf{O}_{6\times 2} \\ \mathbf{O}_{2\times 6} & \mathbf{O}_{2\times 2} \end{bmatrix} \tag{5.29}$$

which implies that virtual controls are only applied to the translational states, acting as synthetic accelerations and velocities affecting the differential equations of

v and **r**, respectively. This choice is due to the nature of the problem, given that including virtual controls affecting the attitude states might not provide any physical improvement to the convergence process.

For what regards the convex terms, they are represented by

$$\mathbf{f}_c(\mathbf{x}) = \begin{bmatrix} \mathbf{v}_{DCA} \\ -2\omega \times \mathbf{v}_{DCA} - \omega \times (\omega \times \mathbf{r}_{DCA}) \\ \mathbf{O}_{2\times 1} \end{bmatrix} = \mathbf{A}_c \cdot \mathbf{x} \tag{5.30}$$

with

$$\mathbf{A}_c \triangleq \begin{bmatrix} \mathbf{O}_{[3\times 3]} & \mathbf{I}_{[3\times 3]} & \mathbf{O}_{[3\times 2]} \\ -\omega \times (\omega\times) & -2\omega\times & \mathbf{O}_{[3\times 2]} \\ \mathbf{O}_{[3\times 3]} & \mathbf{O}_{[3\times 3]} & \mathbf{O}_{[3\times 2]} \end{bmatrix} \tag{5.31}$$

The matrix $\mathbf{A}_c$ only contains constant terms, and is therefore computed only once during the initialization of the algorithm.

The non-convex term can be convoyed into the fourth, fifth and sixth elements of the vector

$$\mathbf{f}_{nc}(\mathbf{x}) = \begin{bmatrix} \mathbf{O}_{3\times 1} \\ \mathbf{a}_{DCA}^{grav} + \mathbf{a}_{DCA}^{aero} \\ \mathbf{O}_{2\times 1} \end{bmatrix} \tag{5.32}$$

and this contribution represents the only term that requires linearization. Finally, the control matrix **B** is

$$\mathbf{B} = \begin{bmatrix} \mathbf{O}_{6\times 2} \\ \mathbf{I}_{2\times 2} \end{bmatrix} \tag{5.33}$$

We can see that the system is affine in control, which is a very important property of the problem to be iteratively solved by using sequential convex programming, as demonstrated by Liu et al. [12]. Moreover, the structure chosen to represent the problem suggests us that we can apply a partial linearization and perform sequential convex programming by exploiting the distinction between convex and non-convex terms. On this purpose, suppose we have solved the problem k times, with $k = 0, \ldots, k_{max}$. The solution with $k = 0$ can either be a propagation of dummy controls, or a linear interpolation between initial and final states and controls. To solve the $(k+1)$th sub-problem we linearize the nonlinear terms of equations of motion around the sub-solution k. The subscript k will indicate the terms computed by using the corresponding kth solution. We can therefore rewrite the system described in Eq. (5.26) as

$$\dot{\mathbf{x}} = \mathbf{A}_c\mathbf{x} + \mathbf{A}_k\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{C}\nu + \mathbf{G}_k \tag{5.34}$$

where

$$\mathbf{A}_k \triangleq \left.\frac{\partial \mathbf{f}_{nc}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_k} \tag{5.35}$$

and

$$\mathbf{G}_k \triangleq \mathbf{f}_{nc}(\mathbf{x}_k) - \mathbf{A}_k \mathbf{x}_k \tag{5.36}$$

Note that, as highlighted by multiple authors [28, 30] it is necessary that the new solution does not largely differ from the previous one. This condition is needed to ensure that the nonlinear behavior of the system is well captured by the first two terms of the Taylor expansions underlying the linearization. To have a meaningful linearization process trust-region constraints are adopted. The way to implement trust-region constraints has been widely treated in literature in multiple forms. Some researchers prefer to express the trust-region radius as a user-defined vector [10]. This approach has the advantage to reduce the size of the problem, since the trust-region size is an input to the subproblem to be solved, rather than a variable to be optimized. Other relevant works include update rules for shrinking or enlarging their size depending on some metrics measuring the validity of the linearization at each iteration [5, 13]. Finally, a further approach consists in introducing dynamic upper bounds for the trust region as part of the subproblem formulation [28]. In a similar fashion to this last approach we introduce trust-region upper bounds on the difference between the new solution and the previous iteration, used to build the current subproblem to be solved.

$$\left\| \tilde{\mathbf{X}}(t) - \tilde{\mathbf{X}}_k(t) \right\| \leq \zeta(t) \tag{5.37}$$

with

$$\tilde{\mathbf{X}} \triangleq \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{bmatrix}, \quad \tilde{\mathbf{X}}_k \triangleq \begin{bmatrix} \mathbf{x}_k(t) \\ \mathbf{u}_k(t) \end{bmatrix} \tag{5.38}$$

and ζ representing an upper bound that limits the excursion between two consecutive iterations, to be penalized as well through a corresponding slack variable

$$\| \zeta \|_2 \leq s_\zeta, \quad s_\zeta \in \mathbb{R} \tag{5.39}$$

Note that the problem must be scaled in order to have the construction of the norm in Eq. (5.37) to be a legitimate operation.

B. Constraints

As mentioned in Sect. 5.3 no nonlinear constraints have been considered in the formulation of the aerodynamic guidance problem. However other constraints need to be included: specifically, we define upper and lower bounds on states and controls corresponding to Eqs. (5.13)–(5.14) as pointwise linear inequalities. Moreover, we need to impose constraints on virtual controls. Since they are only used to avoid *artificial obstructions*, it is required to reduce them to a negligible value along the convergence process to ensure that the computed trajectory is a physical solution to our problem. For this reason, as already proposed in literature ([28]) every virtual control vector is bounded by a corresponding slack variable η_v

$$\|\nu\|_2 \leq \eta_\nu, \quad \eta_\nu \in \mathbb{R}^{np} \tag{5.40}$$

and to ensure that the virtual controls are minimized over the process, we include a further slack variable as upper bound for the norm of η_ν:

$$\|\eta_\nu\|_2 \leq s_\eta \tag{5.41}$$

with the term s_η included in the cost function, and scaled by a positive value w_η.

C. Augmented Cost function
To include the penalization of trust regions and virtual controls in the formulation the augmented cost function for the subproblem is defined as

$$J_{aug}(\mathbf{x}, \mathbf{u}, \nu, \zeta, s_\eta, s_\zeta) = w_u J + w_\eta \cdot s_\eta + w_\zeta \cdot s_\zeta \tag{5.42}$$

with the weights w_η and w_ζ measuring the relative importance of the penalization of virtual control and trust region with respect to the true cost function J defined in Eq. (5.15), which is weighted by w_u. In this work w_η is assumed equal to 10^4 for both the descent and the landing phases, while w_ζ is equal to 10^{-1} for the aerodynamic phase, and to 1 for the powered segment. This choice gives full priority to the reduction of the virtual controls, and poses as secondary objective the shrinkage of the trust-region radii. A unitary value of w_u is associated with the original cost function. In conclusions during the aerodynamic phase we are interested to optimize at each iteration Eq. (5.42), subject to Eqs. (5.34) while ensuring proper penalization of both the trust region size through Eqs. (5.37) and (5.39), and a shrinkage of virtual controls through Eqs. (5.40) and (5.41).

Powered Landing

A. Equations of motion
By extending the logic of the previous section, we expand the state vector for the powered landing phase in the following manner,

$$\mathbf{x} = \left[\mathbf{r}_{DCA} \ \mathbf{v}_{DCA} \ m \ \phi \ \theta \ \psi \ T \right]^T \tag{5.43}$$

with the corresponding dynamics that in vector-form remains the same as Eq. (5.26) but where the control vector is now the following.

$$\mathbf{u} = \left[u_\phi \ u_\theta \ u_\psi \ u_T \right]^T \tag{5.44}$$

The virtual control vector $\nu \in \mathbb{R}^{n_s}$, is in this case defined as

$$\mathbf{v} = \left[\nu_{r_x} \ \nu_{r_y} \ \nu_{r_z} \ \nu_{v_x} \ \nu_{v_y} \ \nu_{v_z} \ \nu_m \ \nu_\phi \ \nu_\theta \ \nu_\psi \ \nu_T \right]^T \tag{5.45}$$

and the matrix $\mathbf{C}$ defined in Eq. (5.29) is augmented accordingly,

$$\mathbf{C} = \begin{bmatrix} \mathbf{I}_{6\times 6} & \mathbf{O}_{6\times 5} \\ \mathbf{O}_{5\times 6} & \mathbf{O}_{5\times 5} \end{bmatrix} \tag{5.46}$$

such that also in this case virtual controls affect the translational motion only. The convex terms are common to those defined during the aerodynamic descent, modified only to take the different size of the state vector into account.

$$\mathbf{f}_c(\mathbf{x}) = \begin{bmatrix} \mathbf{v}_{DCA} \\ -2\omega \times \mathbf{v}_{DCA} - \omega \times (\omega \times \mathbf{r}_{DCA}) \\ \mathbf{O}_{5\times 1} \end{bmatrix} = \mathbf{A}_c \cdot \mathbf{x} \tag{5.47}$$

with

$$\mathbf{A}_c \triangleq \begin{bmatrix} \mathbf{O}_{[3\times 3]} & \mathbf{I}_{[3\times 3]} & \mathbf{O}_{[3\times 5]} \\ -\omega \times (\omega\times) & -2\omega\times & \mathbf{O}_{[3\times 5]} \\ \mathbf{O}_{[5\times 3]} & \mathbf{O}_{[5\times 3]} & \mathbf{O}_{[5\times 5]} \end{bmatrix} \tag{5.48}$$

The non-convex terms are grouped into the fourth, fifth, sixth and seventh differential equations coming from Eq. (5.16).

$$\mathbf{f}_{nc}(\mathbf{x}) = \begin{bmatrix} \mathbf{O}_{3\times 1} \\ \mathbf{a}_{DCA}^{thr} + \mathbf{a}_{DCA}^{grav} + \mathbf{a}_{DCA}^{aero} \\ -\frac{T_{vac}}{I_{sp} g_0} \\ \mathbf{O}_{4\times 1} \end{bmatrix} \tag{5.49}$$

As previously done, numerical linearization is applied to these terms only. Finally, the control matrix $\mathbf{B}$ is

$$\mathbf{B} = \begin{bmatrix} \mathbf{O}_{7\times 4} \\ \mathbf{I}_{4\times 4} \end{bmatrix} \tag{5.50}$$

The procedure is therefore exactly the same as the one highlighted in the previous section. We apply to this augmented formulation Eqs. (5.34)–(5.39), (5.40), (5.41) at each iteration. The only differences reside in the size of states and controls, and in the corresponding non-convex contributions. Moreover, we have different constraints and cost function, described in the next subsections.

B. Constraints
As aforementioned the constraints included in this work, and described by Eqs. (5.20) and (5.21) can be exactly implemented as Second-order conic constraints. For the glide-slope constraint we impose

$$\left\|\mathbf{A}_{gs}\mathbf{r}+\mathbf{b}_{gs}\right\| \leq \mathbf{c}_{gs}\mathbf{r}+d_{gs} \tag{5.51}$$

with

$$\begin{aligned}\mathbf{A}_{gs}\begin{bmatrix}0&1&0\\0&0&1\end{bmatrix},\quad \mathbf{b}_{gs}=\begin{bmatrix}0\\0\end{bmatrix},\\ \mathbf{c}_{gs}=\begin{bmatrix}1&0&0\end{bmatrix},\quad d_{gs}=0\end{aligned} \tag{5.52}$$

For the limitations of horizontal position and velocity at the end of the landing phase, we can derive similar expressions:

$$\begin{aligned}\|\mathbf{A}_r\mathbf{r}+\mathbf{b}_r\| \leq \mathbf{c}_r\mathbf{r}+d_r\\ \|\mathbf{A}_v\mathbf{v}+\mathbf{b}_v\| \leq \mathbf{c}_v\mathbf{v}+d_v\end{aligned} \tag{5.53}$$

where the corresponding matrices are defined as

$$\begin{aligned}\mathbf{A}_r=\mathbf{A}_v=\begin{bmatrix}0&1&0\\0&0&1\end{bmatrix},\quad \mathbf{b}_r=\mathbf{b}_v=\begin{bmatrix}0\\0\end{bmatrix},\\ \mathbf{c}_r=\mathbf{c}_v=\begin{bmatrix}0&0&0\end{bmatrix},\quad d_r=1,\quad d_v=0.1\end{aligned} \tag{5.54}$$

C. Augmented Cost function
The augmented cost is formally the same as Eq. (5.42). However some practical terms will differ due to the application peculiarities since the true cost J, defined respectively by Eqs. (5.15) and (5.24) are clearly distinct. Finally, w_m and w_u are in this case equal to 100 and 10, respectively. To summarize the landing convexified problem we want to minimize at each iteration Eq. (5.42). The solution must satisfy Eqs. (5.34), (5.20), and (5.21) while ensuring proper penalization of both the trust region size through Eqs. (5.37) and (5.39), and of the virtual controls through Eqs. (5.40) and (5.41). Note that since no 6-DoF dynamics is considered in this work no explicit penalization of roll torque commands are included. However, the limitation imposed on the roll rate is such that it will be possible for the attitude control system to track it. In the next section we will transcribe the problem through the use of hp generalized pseudospectral methods.

5.5 Sequential Pseudospectral Convex Programming

The transcription proposed here is conceptually common to both the aerodynamic descent and the powered landing problems. Consequently, we will focus on the aspects common to both first, emphasizing later the differences between the two spe-

cific formulations, especially in terms of constraints, cost function, and initialization strategy. The transcription we propose adopts an hp generalized pseudospectral transcription based on the use of flipped Legendre-Gauss-Radau (fLGR) method. This method is a valid alternative to the more traditional Euler and trapezoidal discretization transcriptions, given its higher accuracy. Moreover, its main drawback, i.e., a typically larger CPU time, can be mitigated by proper choice of h and p [20]. In the remainder of this section we will identify the steps of transcription according to the Sequential Pseudospectral Convex Programming (SPCP) method here proposed.

5.5.1 Discretization

Motivated by the good results obtained in our previous works [18, 20] we extend the methodology to the problem formulated in Sect. 5.4. Specifically, we propose to use n segments, and in each of them perform a local collocation using $p+1$ nodes coming from the p roots of the corresponding fLGR polynomial, defined in the domain $(-1, \quad 1]$, and initial non-collocated node at $\tau = -1$. A visualization of the domain is visible in Fig. 5.3.

Note that since the domain is broken into segments, some linking conditions connecting them are needed. They will explicitly be defined in this section, and form, together with the equations of motion and the boundary conditions the set of linear equations underlying the transcription.

Another benefit is associated with the possibility to have an open final-time formulation of the guidance problem. In fact, to come up with a free final-time discretization some researchers prefer to reformulate the problem by using a different independent variable, known to be monotonically changing, and with known initial and final values [31]. A different approach was the use of a stretching term $\hat{\sigma}$ in the equations of motion [27]. This term can be in fact re-interpreted as very close to the typical mapping between physical time and pseudospectral time k_t, defined as

$$k_t \triangleq \frac{t_F - t_0}{2n} \tag{5.55}$$

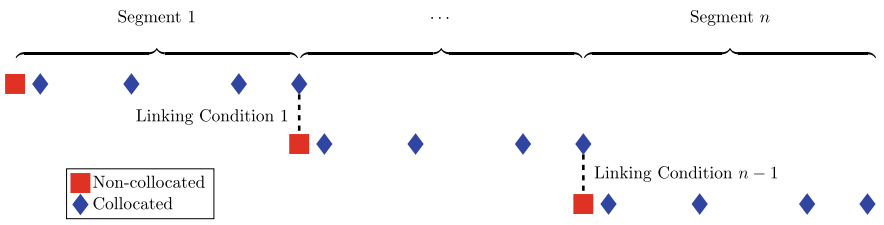


Fig. 5.3 Domain of the hp flipped Legendre-Gauss-Radau method

with t_0 and t_F initial and final times of the physical problem to be solved, and n the number of phases to be used in the hp framework we are proposing. This observation helps in rewriting the transcription in presence of free final time as follows. We assume to have the state vector $\mathbf{x} \in \mathbb{R}^{n_s}$, the control vector $\mathbf{u} \in \mathbb{R}^{n_c}$, the virtual control vector $\nu \in \mathbb{R}^{n_v}$, and a total of $n(p+1)$ discrete time steps, corresponding to the n segments and the $p+1$ discrete points in each of the segments. Computed the $p+1$ nodes τ_i corresponding to a single segment and defined between -1 and 1 we can, for every couple t_0, and t_F, identify the single ith discrete timestep associated with the jth segment as

$$t_i^j = t_0 + \frac{t_F - t_0}{n}\left(j - \frac{1}{2}\right) + \frac{t_F - t_0}{2n}\tau_i, \quad j = 1, \ldots, n-1 \tag{5.56}$$

Defined the discrete time domain, we can introduce the augmented discrete decision vector $\mathbf{X} \in \mathbb{R}^{n_{var}}$ as

$$\mathbf{X} = \left[\, \mathbf{x}_0^1 \; \mathbf{u}_0^1 \; \nu_0^1 \cdots \mathbf{x}_p^n \; \mathbf{u}_p^n \; \nu_p^n \; \eta_0^1 \cdots \eta_p^n \; \zeta_0^1 \cdots \zeta_p^n \; \mu_0^1 \cdots \mu_p^n \; s_\eta \; s_\zeta \; t_F \,\right]^T \tag{5.57}$$

The first $[n_s + n_c + n_v]\,[n(p+1)]$ elements correspond to states, control and virtual controls, respectively. The set of data associated with $\eta_0^1,\ldots,\eta_p^n$ represents the upper bounds on virtual controls, while the variables identified as $\zeta_0^1,\ldots,\zeta_p^n$ constrain the size of the pointwise trust regions. The variables $\mu_0^1,\ldots,\mu_p^n$ are associated with the cost function. We can see the presence of the slack variables s_η and s_ζ, penalizing virtual controls and trust regions, as they appear in Eq. (5.42). Finally, since the problem has open final time the variable t_F appears as last element of the augmented vector, by assuming, without compromising any possibility of general application of the proposed method, that $t_0 = 0$. This assumption is always applicable by simply shifting the time vector by the initial time of the aerodynamic descent or the powered landing sequence.

5.5.2 Dynamics

Let us consider the dynamics of our system as convexified in Eq. (5.34). By adopting the time mapping of Eq. (5.55) we can rewrite it as follows:

$$\dot{\mathbf{x}} = k_t \left[\mathbf{A}_c\mathbf{x} + \mathbf{A}_k\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{C}\nu + \mathbf{G}_k\right] \tag{5.58}$$

where the equation now describes the evolution of the states with respect to a new independent variable τ, defined between –1 and 1, and chosen because it represents the domain of definition of Legendre-Gauss-Radau polynomials [17, 19]. We can use this expression to derive linear system of equations representing Eq. (5.58) in discrete form.

By keeping in mind that only $\mathbf{A}_k$ and $\mathbf{G}_k$ change at each iteration we can perform an expansion of the right-hand side of Eq. (5.58) with respect to the variables $\mathbf{x}$, $\mathbf{u}$, ν, and t_F. Let us define the following quantities.

$$\begin{aligned}
\tilde{\mathbf{A}} &\triangleq k_t \left[\mathbf{A}_k + \mathbf{A}_c\right] \\
\tilde{\mathbf{B}} &\triangleq k_t \mathbf{B} \\
\tilde{\mathbf{C}} &\triangleq k_t \mathbf{C} \\
\tilde{\mathbf{E}} &\triangleq \tfrac{1}{2n} \left[\mathbf{f}_{nc}(\mathbf{x}_k) + \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{C}\nu_k\right] \\
\tilde{\mathbf{G}} &\triangleq k_t \mathbf{G}_k - \tfrac{t_{F,k}}{2n} \left[\mathbf{f}_{nc}(\mathbf{x}_k) + \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{C}\nu_k\right]
\end{aligned} \tag{5.59}$$

It is straightforward to verify that the equations of motion can be expressed as

$$\dot{\mathbf{x}} = \tilde{\mathbf{A}}\mathbf{x} + \tilde{\mathbf{B}}\mathbf{u} + \tilde{\mathbf{C}}\nu + \tilde{\mathbf{E}}t_F + \tilde{\mathbf{G}} \tag{5.60}$$

This expression needs to be tailored for the specific domain of choice. In this work we choose to apply the hp-methods as a series of n equally spaced segments, and in each of them we can collocate the differential equations using $p + 1$ nodes. This choice is motivated by the fact that we are interested to real-time-capable methods, and therefore we do not focus on refinement methods which iteratively adapt the size and the distribution of the meshes.

The final step is the inclusion of the pseudospectral differential operator. Note that the derivative of the state in the discrete points $\mathbf{x}_i, i = 1, \ldots, n$ can be approximated by a matrix $\mathbf{D}$ in the form

$$\dot{\mathbf{x}} \cong \mathbf{D} \cdot \mathbf{x} \tag{5.61}$$

Equation (5.61) tells us that the derivative in one of the discrete points of the domain can be approximated by a linear combination of the values that the variable $\mathbf{x}$ assumes over all the discretized points through the coefficients provided by the columns of $\mathbf{D}$.

We can exploit this property to finally build the linear matrix representing the equations of motion as follow: defined $\mathbf{D}_{i,\ldots}$ as the ith row of $\mathbf{D}$ we have that for each segment $j \in [1, \ldots, n]$ and node $i \in [0, \ldots, p]$

$$\mathbf{D}_{i,\ldots} \cdot \mathbf{x}^j_{0,\ldots,p} - \tilde{\mathbf{A}}\mathbf{x}^j_i - \tilde{\mathbf{B}}\mathbf{u}^j_i - \tilde{\mathbf{C}}\nu^j_i - \tilde{\mathbf{E}}t_F = \tilde{\mathbf{G}} \tag{5.62}$$

which in matrix form can be assembled as

$$\mathbf{A}_{EoM}\mathbf{X} = \mathbf{b}_{EoM} \tag{5.63}$$

5.5.3 Boundary Conditions

We can augment the previous linear system of Eq. (5.63) by including hard constraints to comply with initial and final states definitions. The corresponding matrix will simply by

$$\mathbf{A}_{bc}\mathbf{X} = \mathbf{b}_{bc} \tag{5.64}$$

corresponding to

$$\mathbf{A}_{bc} = \begin{bmatrix} I_{n_s} & O_{n_s\times(n_t)(n)(p-2)} & O_{n_s\times n_s} & O_{n_s\times(n_{var}-[(n_t)(n)(p-2)+2(n_s)])} \\ O_{n_s\times n_s} & O_{n_s\times(n_t)(n)(p-2)} & I_{n_s} & O_{n_s\times(n_{var}-[(n_t)(n)(p-2)+2(n_s)])} \end{bmatrix} \tag{5.65}$$

where I_{n_x} and $O_{n_y\times n_z}$ are the identity matrix and the zero matrix of size n_x and $n_y \times n_z$, respectively, while $n_t = n_s + n_c + n_v$. The vector $\mathbf{b}_{bc}$ is intuitively defined as

$$\mathbf{b}_{bc} = \begin{bmatrix} \mathbf{x}(t_0) \\ \mathbf{x}(t_F) \end{bmatrix} \tag{5.66}$$

In case some initial and / or final conditions are left free the corresponding rows in Eqs. (5.65) and (5.66) can simply be deleted.

5.5.4 Linking Conditions

The discretization introduced in Fig. 5.3 requires some extra constraints known as *linking conditions* or *linkage conditions*, needed to enforce continuity of the states and controls defined on the edge of the segments. These conditions are represented by equality constraints in the form

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{u} \\ \nu \end{bmatrix}_p^j = \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \\ \nu \end{bmatrix}_0^{j+1}, \quad j \in [1, n-1] \tag{5.67}$$

It is immediate to see that these conditions can be built by assigning identity matrices of consistent dimension to the proper indices of a matrix $\mathbf{A}_{\mathbf{lc}}$ with the corresponding vector $\mathbf{b}_{\mathbf{lc}} \triangleq O_{n_t\times 1}$.

The overall system of differential equations is therefore given by

$$\mathbf{A}_{eq}\mathbf{X} = \mathbf{b}_{eq} \tag{5.68}$$

with

$$\mathbf{A}_{eq} = \begin{bmatrix} \mathbf{A}_{EoM} \\ \mathbf{A}_{bc} \\ \mathbf{A}_{lc} \end{bmatrix}, \quad \mathbf{b}_{eq} = \begin{bmatrix} \mathbf{b}_{EoM} \\ \mathbf{b}_{bc} \\ \mathbf{b}_{lc} \end{bmatrix} \tag{5.69}$$

5.5.5 Cost

For the cost function by recovering Eq. (5.42), and remembering the quadrature expression associated with the hp fLGR method an the continuous expression we can rewrite it as

$$J_{aug} = -w_m m_p^n + w_u \frac{t_F - t_0}{2n} \mathbf{w}_{fLGR}^T \mu + w_v \cdot s_v + w_\zeta \cdot s_\zeta + w_{t_F} s_t \tag{5.70}$$

where the weights w_{fLGR} are dictated by the fLGR theory [17], and w_m equal to 0 if we refer to the aerodynamic descent, or larger than 0 if referred to the powered landing case. Finally, note that a penalization on the final time variation is included through a corresponding slack variable s_t. This variable acts as upper bound on the variation of the final time with respect to the previous one, and is also modeled as conic constraint. Its construction is trivial and skipped to avoid excess of redundancy in the equations.

Remark 3: Note that since we are dealing with open final time problems a formal linearization of the second term in Eq. (5.70), bilinear in the variables μ_i^j and t_F, is needed. For easiness of implementation this linearization is not carried out, and approximated by $\frac{t_{F,k}-t_0}{2n} w_{fLGR}^T \mu$. This approximation is valid as long as the condition $(t_F - t_{F,k})\mathbf{w}_{fLGR}^T \mu_k \ll (t_F - t_0)\mathbf{w}_{fLGR}^T \mu$ is satisfied. This aspect is omitted here for brevity, but numerically verified during the simulations, and therefore the approximation is valid.

The role of the variables μ is to act as upper bound for the true elements appearing in the cost function of Eq. (5.15). Given the selected cost function it is immediate to observe that it can be cast in a second-order cone constraint as follows.

$$\left\| \begin{matrix} u_\theta \\ u_\psi \end{matrix} \right\|_{2,i}^j \leq \mu_i^j, \quad \begin{matrix} i \in [0, p] \\ j \in [1, n] \end{matrix} \tag{5.71}$$

The transcription is completed by applying point-wise the second-order conic constraints representing the upper bound on virtual controls, i.e.,

$$\left\| \nu_i^j \right\|_2 \leq \eta_{\nu,i}^j, \quad \begin{matrix} i \in [0, p] \\ j \in [1, n] \end{matrix} \tag{5.72}$$

and on the trust region.

$$\left\| \tilde{\mathbf{X}} - \tilde{\mathbf{X}}_k \right\|_{2,i}^j \leq \zeta_i^j \tag{5.73}$$

$$\| \zeta \|_{2,i}^j \leq s_{\zeta,i}^j, \quad s_\zeta \in \mathbb{R}^{np} \tag{5.74}$$

5.5.6 Constraints—Powered Landing

In addition to Eq. (5.71) (opportunely augmented to consider the different number of controls) the constraints of Eqs. (5.20) and (5.21) are also modeled as second-order conic constraints. Therefore, they can simply be applied to each discrete node representing position and velocity.

$$\begin{aligned} \left\| \mathbf{A}_{gs}\mathbf{r}_i^j + \mathbf{b}_{gs} \right\|_2 &\leq \mathbf{c}_{gs}\mathbf{r}_i^j + d_{gs} \\ \left\| \mathbf{A}_r\mathbf{r}_i^j + \mathbf{b}_r \right\|_2 &\leq \mathbf{c}_r\mathbf{r}_i^j + d_r, \\ \left\| \mathbf{A}_v\mathbf{v}_i^j + \mathbf{b}_v \right\|_2 &\leq \mathbf{c}_v\mathbf{v}_i^j + d_v \end{aligned} \qquad \begin{aligned} &i \in [0,\, p] \\ &j \in [1, n] \end{aligned} \tag{5.75}$$

5.5.7 Initialization

For the aerodynamic descent the initial guess for the states is built by using linear interpolation between the desired initial and final states, while the controls were kept equal to 0. For the landing phase a more sophisticated strategy is adopted. A scheme illustrating the SPCP algorithm is depicted in Fig. 5.4. Inspired by the idea of Simplicio et al. [24] we adopt an educated guess, obtained by solving the problem with the method described in [20]. Given the reference scenario the simplified problem is solved with hard constraints for initial and final conditions, as well as for the time of flight, assumed to be fixed. In this initialization solution neither aerodynamic effects, nor control rates are considered, and the gravity is assumed to be constant. Then, the obtained solution is converted into a format compatible with the SPCP transcription illustrated in this section, and utilized as k^{th} solution, with $k = 0$ to start the sequential pseudospectral convex optimization procedure.

The solution obtained contains the thrust vector in Cartesian coordinates, which are converted into the corresponding Euler angle representation by assuming that the x-body axis coincides with the thrust vector. This is a valid assumption since we are dealing with a 3-DOF model. The control rates are then obtained by numerically differentiating the Euler angles and the thrust magnitude profile, completing the information required to build the very first solution according to the format described by Eqs. (5.43) and (5.44). It is then possible to start the sequential pseudospectral convex optimization, and at the end of each iteration the algorithm checks whether convergence has been reached. In negative case the current solution is used to build the new subproblem and iterate the procedure. In the opposite case the procedure is concluded. To validate our solution we perform a feedforward propagation of the full nonlinear equations of motion driven by the computed controls, and compare the obtained solution with the optimized one, which is the outcome of the proposed algorithm.

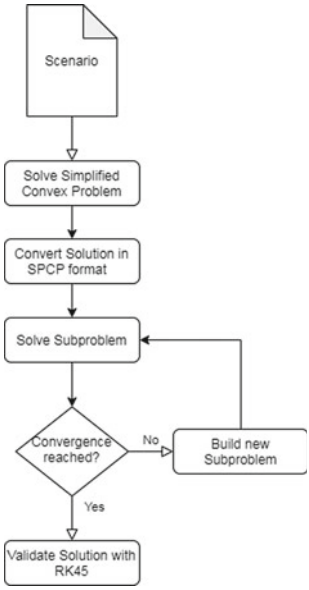


Fig. 5.4 Sequential Pseudospectral Convex Programming (SPCP) scheme

Remark 4: Note that this further step is not considered to be part of the on-board guidance strategy, but it is only meant as validation tool to measure the effectiveness and the accuracy of the proposed algorithm.

5.5.8 Convergence Criterion

Once the algorithm is initialized the process of generating subsolutions is repeated until convergence is reached. In literature many authors meaningfully use as criterion the difference between subsolutions meant in vector or scalar form [28, 29]. To take into account both the convergence of subsolutions and the cost function we use as stopping criterion the difference between consecutive values of the augmented functions, defined in Eq. (5.70). Although this choice plays no big differences in practical terms, since very close subsolutions will also lead to similar cost functions, it represents a way to account also for the variations of other parts of the algorithms, such as virtual controls and trust region radii, which do not belong to the set of physical variables of the problem. The stopping criterion can be therefore expressed as

$$\delta J_{aug} \triangleq \left| J_{aug}^{k} - J_{aug}^{k-1} \right| \leq \epsilon \tag{5.76}$$

with ϵ chosen equal to $5 \cdot 10^{-5}$ for the aerodynamic phase and to $2 \cdot 10^{-4}$ for the powered landing segment. The choice is dictated by empirical experience showing that given the shorter duration of the powered phase a slightly less stringent criterion

is sufficient to reach very good accuracy while limiting the number of iterations, and therefore speeding up the total execution time of the algorithm.

5.6 Numerical Results

This section illustrates results obtained with the proposed method for both types of scenarios. First, we will describe the aerodynamic descent scenario, followed by the powered landing results.

5.6.1 Aerodynamic Descent—Nominal

For the aerodynamic descent initial and final conditions are described in Table 5.1. Note that all the positions have been scaled with respect to the initial altitude, the scaled gravity is equal to 1, and all the other variables have been scaled consistently with these two assumptions. Results are shown in Figs. 5.5 through 5.11. By looking at the states (Fig. 5.5) we see that the solution shows a smooth behavior while satisfying initial and final conditions. The same holds for the attitude (Fig. 5.6a), where the specific upper and lower bounds for pitch and yaw are met, and the attitude rates, correctly bounded between –10 and 10° (Fig. 5.6b). The overall trajectory is depicted in Fig. 5.7, where the body axes (in RGB convention) depict the corresponding attitude while performing the aerodynamic descent. The associated aerodynamic behavior is visible in Fig. 5.8a, showing the normalized aerodynamic forces in body axes. Note that to further enhance the 6-DOF feasibility of the solution the aerodynamic forces are computed by dynamically trimming the vehicle. To verify the correct behavior of the solution the resulting aerodynamic torques with respect to the center of mass are stored and observed. The normalized aerodynamic torque residuals are depicted in Fig. 5.8b, and are close to the zero-machine. The trimming is realized by opportunely

Table 5.1 Aerodynamic descent—initial and final conditions

Initial state	Value	Final state	Value
$r_A(t_0)$	1.0000	$r_A(t_f)$	0.0714
$r_C(t_0)$	−0.0038	$r_C(t_f)$	0.0006
$r_D(t_0)$	0.0669	$r_D(t_f)$	0.0158
$v_A(t_0)$	−1.1056	$v_A(t_f)$	−0.4509
$v_C(t_0)$	0.0014	$v_C(t_f)$	0.0026
$v_D(t_0)$	−0.0254	$v_D(t_f)$	−0.0297
$\theta(t_0)$	0.0172	$\theta(t_f)$	−0.0391
$\psi(t_0)$	−0.0030	$\psi(t_f)$	−0.0037

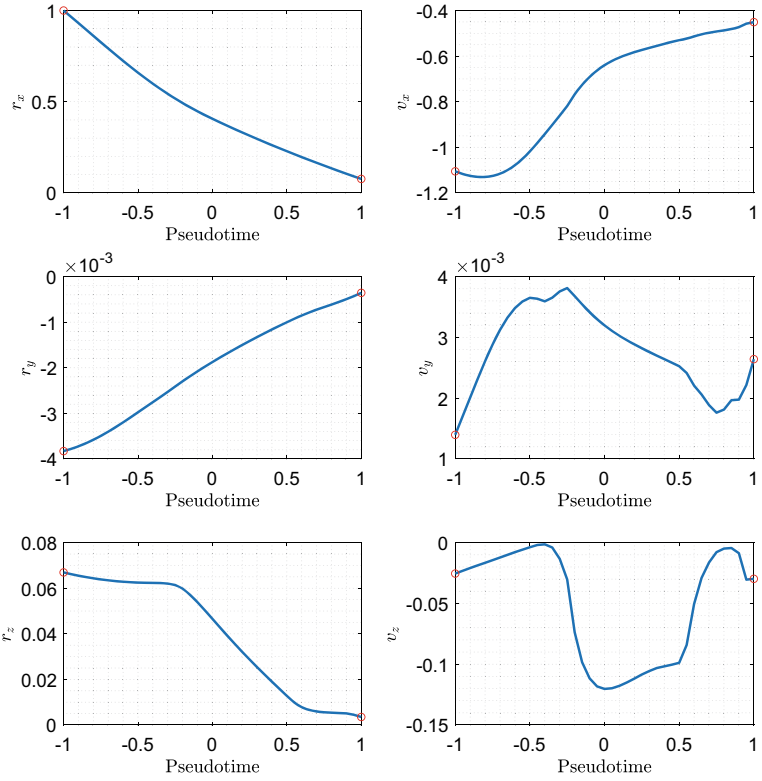
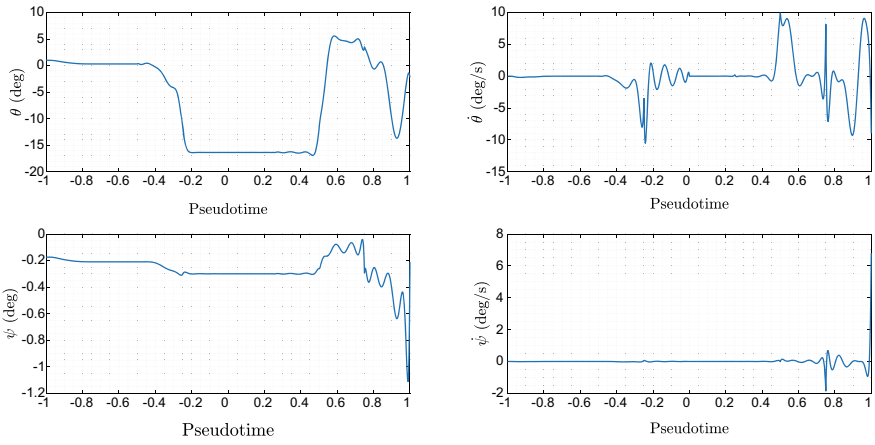


Fig. 5.5 Aerodynamic guidance solution—translational states

deflecting the four fins, which ensure that the aero-torque disturbances are nullified while satisfying the maximum allowed fin deflections (shown in normalized coordinates in Fig. 5.8c).

For what regards the accuracy of the solution a full propagation of the trajectory by using the full set of nonlinear equations is performed, and depicted in Fig. 5.9a, with the mismatch between the two profiles visible in Fig. 5.9b. The two solutions agree very well, with a maximum scaled error in the order of $4 \cdot 10^{-4}$ for the position and $2 \cdot 10^{-3}$ for the velocity. In full scale these results correspond to meter-error for the position, and less than 0.5 m/s for the velocity. The convergence behavior is depicted in Figs. 5.10a through 5.10f. The first thing to observe is the behavior of the cost and the augmented cost, shown in Fig. 5.10a and b. At the beginning of the process larger variations between solutions are experienced. From iteration 4 to the end the algorithm converges to a specific solution, and therefore the upper bounds on trust regions and virtual control fade away. As a consequence the two profiles converge to very similar values. The reduction of virtual controls and trust regions can be seen in Fig. 5.10c and d. We can see that starting from the second iteration the method does not really rely on virtual controls, meaning that the actual controls are sufficient to



(a) Attitude states (b) Attitude rates

Fig. 5.6 Aerodynamic guidance solution: **a** Attitude states, and **b** Attitude rates

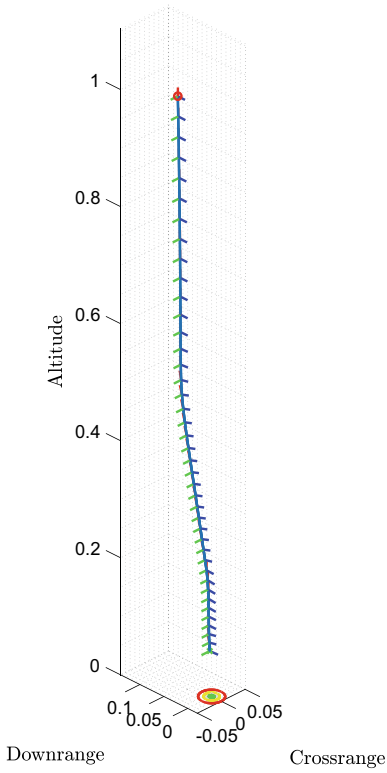
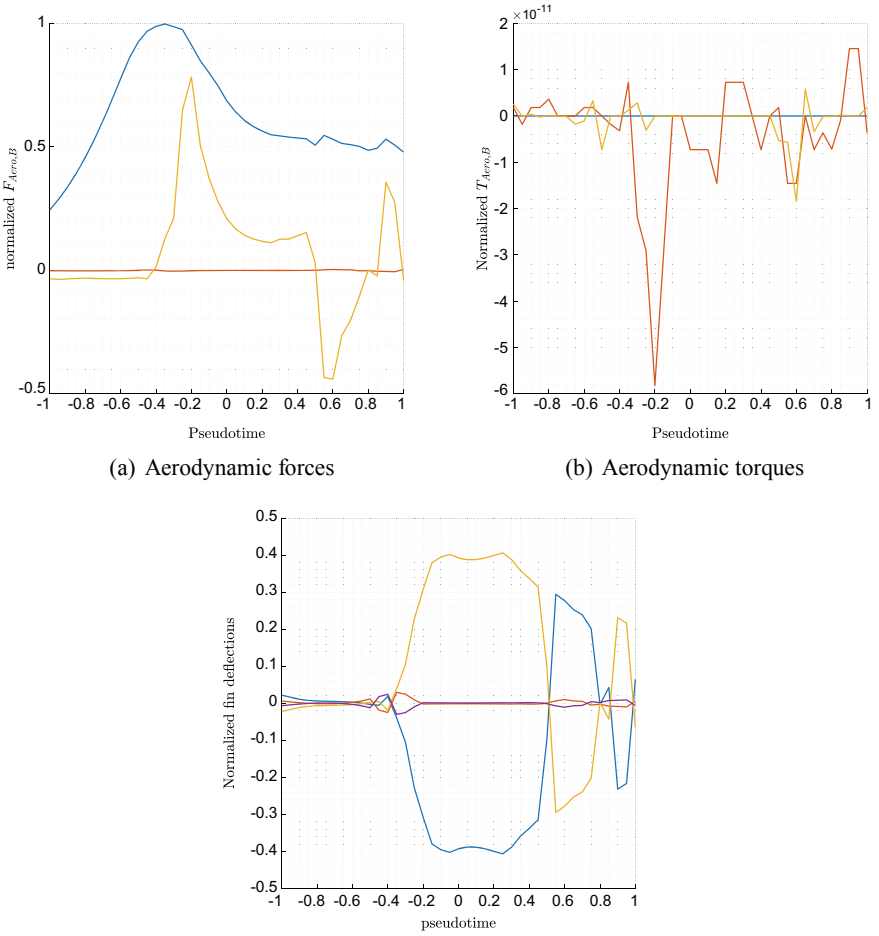


Fig. 5.7 Aerodynamic guidance solution—trajectory: the body axes are depicted in red (X), green (Y), and blue (Z)



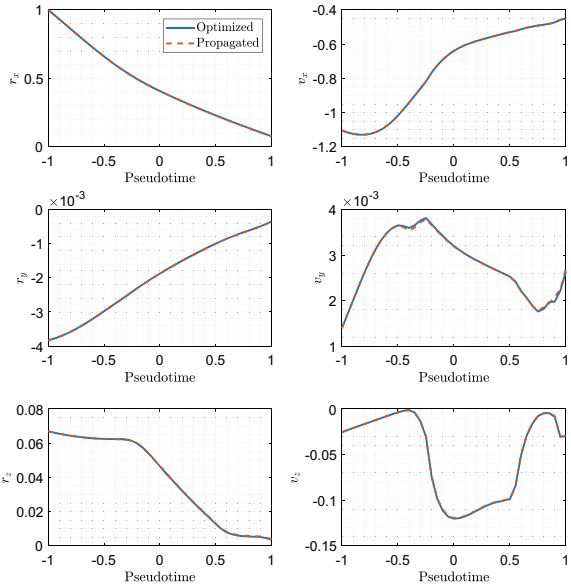
(a) Aerodynamic forces

(b) Aerodynamic torques

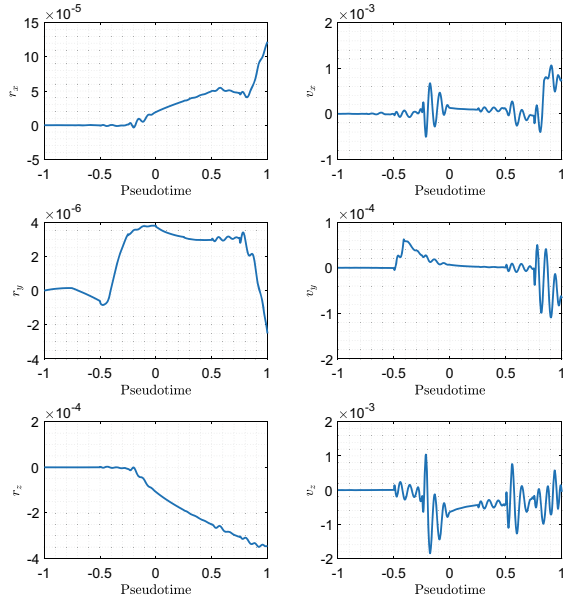
(c) Fin deflections

Fig. 5.8 Aerodynamic guidance solution: **a** Aerodynamic forces, **b** Aerodynamic torques, **c** Fin deflections

solve the convex subproblems. The trust regions upper bound becomes smaller than 10^{-3} starting from the 7^{th} iteration, and no sensitive variations of the augmented cost function and of the solution are observed. Finally, we can observe that the final time variations between consecutive solutions rapidly decreases too (Fig. 5.10e). Note that as further test the initialized final time is given as the converged final time + 10 s to observe whether the algorithm was able to come back to the optimal value. This behavior is confirmed by looking at the first iteration, where a variation of about 9 s is observed, followed by smaller variations along the successive iterations. The last plot on the bottom right (Fig. 5.10f) shows the decisions of keeping or rejecting the subsolution along the iterations. A subsolution is rejected in two cases: first, when the

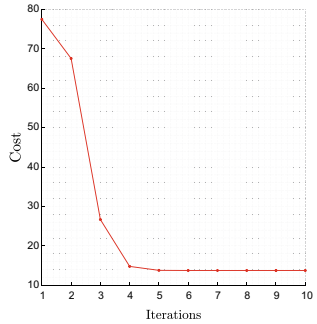


(a) Translational states

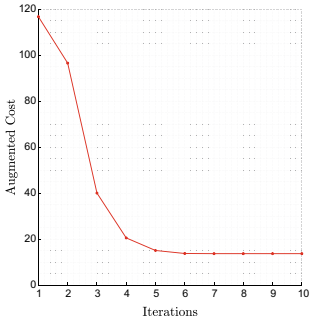


(b) State errors

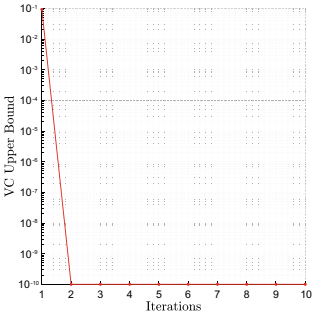
Fig. 5.9 Nominal solution validation: **a** comparison of states, and **b** difference between optimized and propagated states



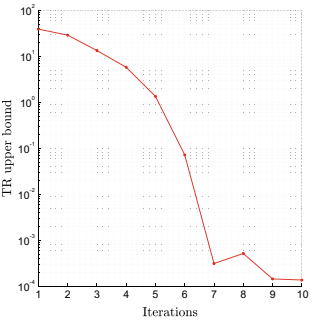
(a) Cost function



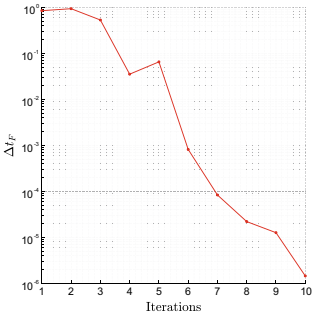
(b) Augmented cost function



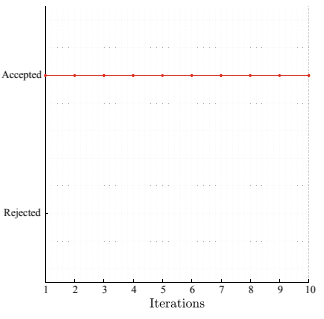
(c) Virtual control upper bound



(d) Trust region upper bound



(e) t_F variations



(f) Acceptance / Rejection sequence

Fig. 5.10 Convergence behavior: **a** Cost function, **b** Augmented cost function, **c** virtual control upper bound, **d** trust region upper bound, **e** final time variations, and **f** acceptance/rejection decision

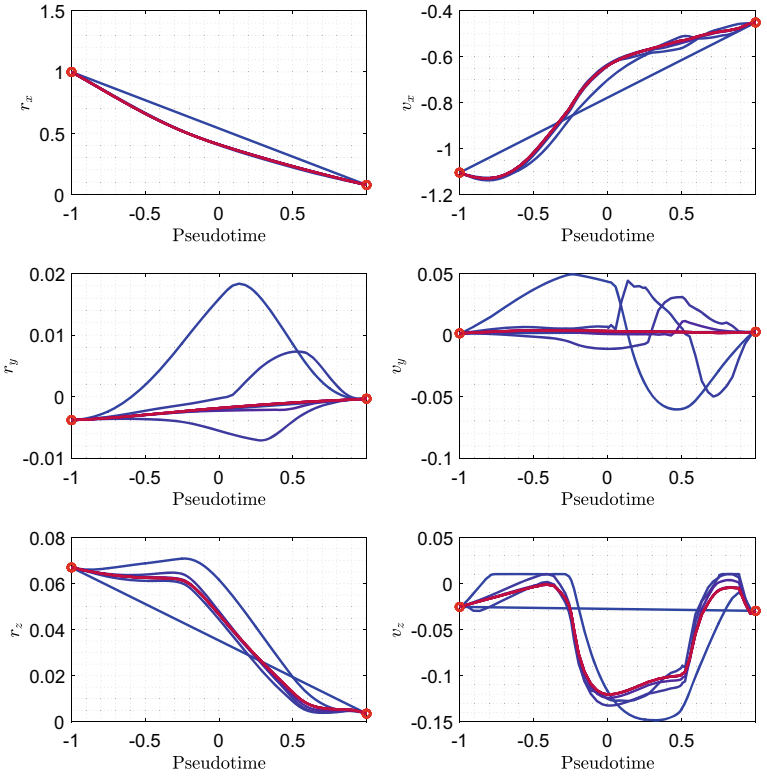


Fig. 5.11 Convergence behavior: evolution of translational states

SOCP solver returns an infeasibility status, or if the maximum number of iterations is reached without finding a valid solution, with the iteration limit for the SOCP solver set equal to 100. This issue did not occur in the results shown here, as confirmed by Fig. 5.10f.

To give an intuitive idea of the convergence process we plot the several subsolutions obtained over the iterations in Fig. 5.11, showing the translational states. The process is initialized with a trivial linear interpolation between initial and final desired states (in blue). After three iterations the solution is already resembling the final one (in red), that is only refined in the remaining iterations. This is a consequence of having variable trust region upper boundaries, which allow larger variations at the beginning if needed, and are dynamically reduced, making, iteration after iteration, the linearized dynamics more and more able to capture the behavior of the nonlinear differential equations underlying the problem.

5.6.2 *Aerodynamic Descent—Dispersed Cases*

As further test we simulated 25 dispersed cases associated with different initial conditions in terms of position and velocity for both the aerodynamic and the powered landing phases. Note that these cases are purely demonstrative and of course not representative of a full Monte-Carlo campaign. However, they confirm the capability of the algorithm to generate valid solutions over a much larger set of conditions than the one given by the nominal scenario. Specifically, since it is assumed that the aerodynamic guidance algorithm is triggered at a specific altitude, errors in terms of crossrange and downrange components have been considered for what regards the position. These errors are equal to 200 m, whereas all the three components of the velocity are perturbed up to ±15 m/s. All the perturbations are uniformly distributed. Figures 5.12 and 5.13 show the resulting trajectories, together with the translational and the rotational states. Moreover, the Runge-Kutta validation of the obtained solutions are depicted in Fig. 5.14a and b. All the trajectories converged to the prescribed interface conditions while satisfying the constraints. Moreover, from

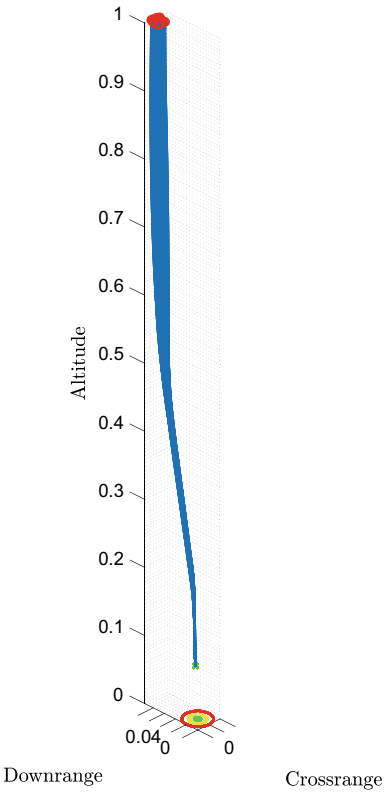
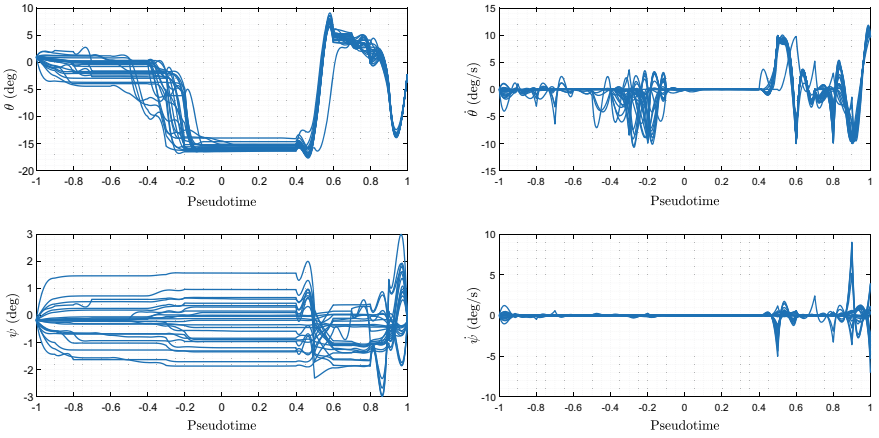


Fig. 5.12 Aerodynamic guidance solution—dispersed trajectories



(a) Attitude states (b) Attitude rates

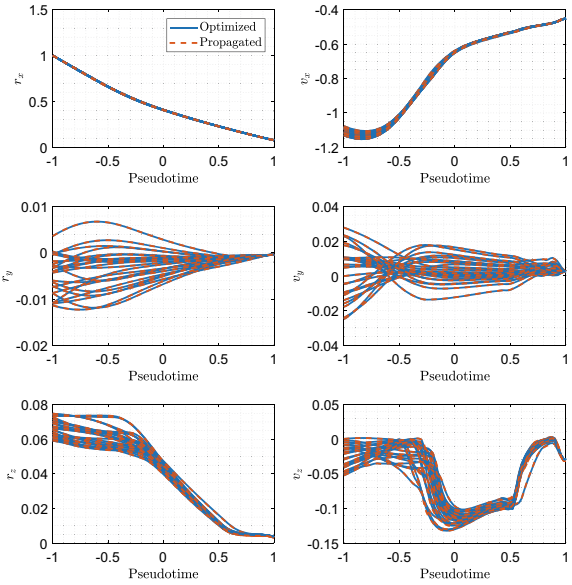
Fig. 5.13 Aerodynamic guidance solution: **a** Attitude states, and **b** Attitude rates

the validation of the solutions (Fig. 5.14a–b) we can see that all of them fully satisfy the equations of motion, with a consistently small error between propagated and optimized states.

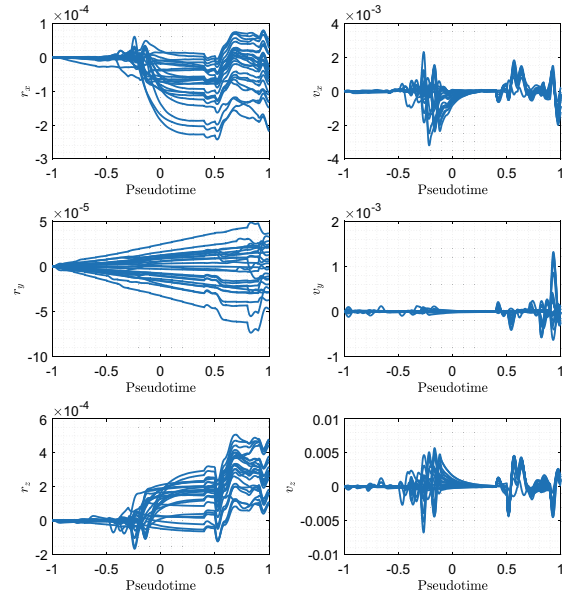
5.6.3 Powered Landing—Nominal

The prescribed initial and final conditions for the powered landing scenario we are dealing with are described in Table 5.2. The weights w_{vc} and w_{tr} are equal to 500 and 10. We keep the same penalization of final-time variations as for the aerodynamic descent. States, and controls are depicted in Figs. 5.15 and 5.16. Besides a smooth solution also in this case with initial and final boundaries fully satisfied we can see that in the last phase of landing the horizontal components of position and velocity are correctly constrained too, and so is the glideslope constraint (here omitted for brevity). Moreover, the attitude rates always lie in the prescribed boundaries, and the vehicle shows a vertical attitude when landing.

The throttle profile, with its corresponding throttle rate and mass profiles are shown in Fig. 5.17. All of them are within the prescribed limits. The attitude is visible also in Fig. 5.18, while the corresponding aerodynamic forces and torques are depicted in Fig. 5.19a and b. Note that the forces are computed also in this case by taking the attitude controllability into account, such that the aerodynamic torque is constantly minimized (Fig. 5.19c). Specifically, only 15% of the fin maximum deflections is sufficient to remove the undesired aerodynamic torque that would from having fin deflections equal to 0°. Since the effectiveness of fins decreases with the dynamic



(a) Translational states



(b) State errors

Fig. 5.14 Dispersed solutions validation: **a** Comparison of states, and **b** Difference between optimized and propagated states

Table 5.2 Powered landing—initial and final conditions

Initial state	Value	Final state	Value
$r_A(t_0)$	1.0000	$r_A(t_f)$	0.0002
$r_C(t_0)$	−0.0049	$r_C(t_f)$	0.0000
$r_D(t_0)$	0.053	$r_D(t_f)$	0.0000
$v_A(t_0)$	−1.4778	$v_A(t_f)$	−0.0064
$v_C(t_0)$	0.0087	$v_C(t_f)$	0.0000
$v_D(t_0)$	−0.1511	$v_D(t_f)$	0.0000
$m(t_0)$	1.0000	$m(t_F)$	–
$\phi(t_0)$	3.1415	$\phi(t_f)$	3.1415
$\theta(t_0)$	0.0126	$\theta(t_f)$	0.0000
$\psi(t_0)$	−0.0073	$\psi(t_f)$	0.0000
$T(t_0)$	1.100	$T(t_F)$	–

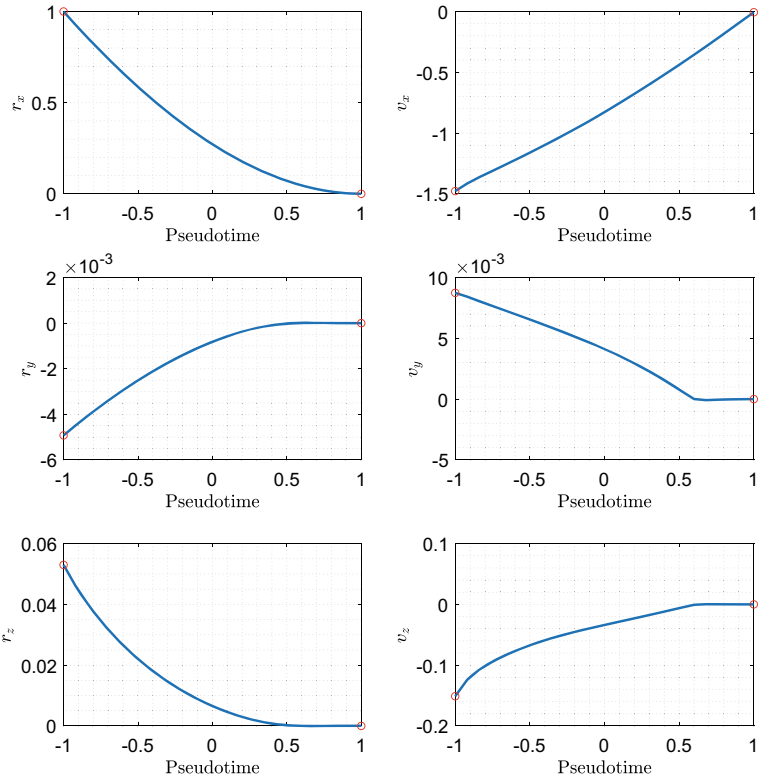
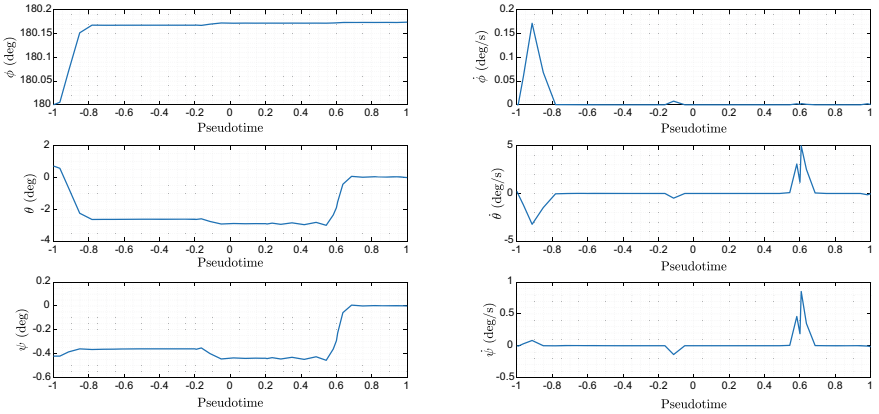


Fig. 5.15 Landing guidance solution—translational states



(a) Attitude States (b) Attitude Rates

Fig. 5.16 Landing guidance solution: **a** Attitude states, and **b** Attitude rates

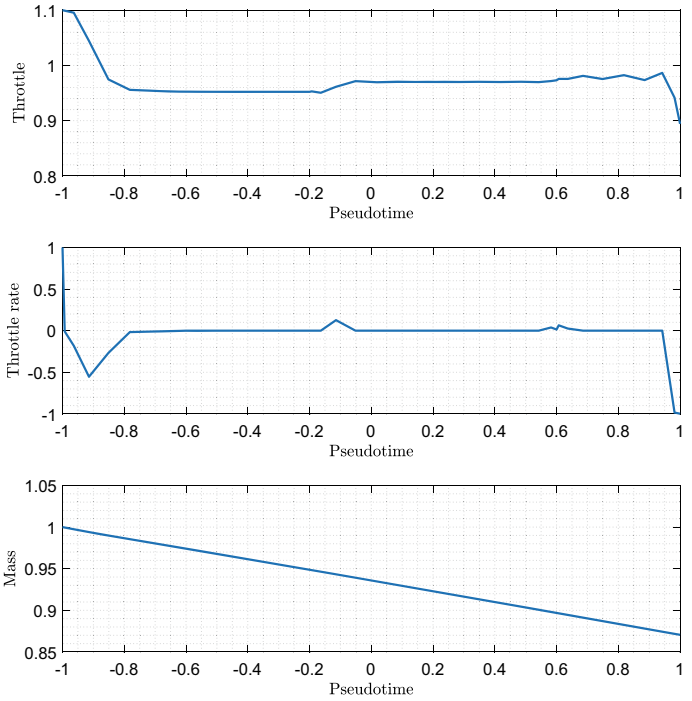


Fig. 5.17 Landing guidance solution: throttle, throttle rate, and normalized mass

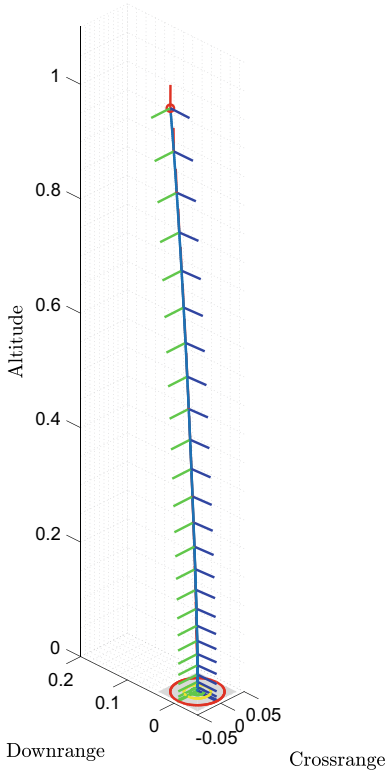
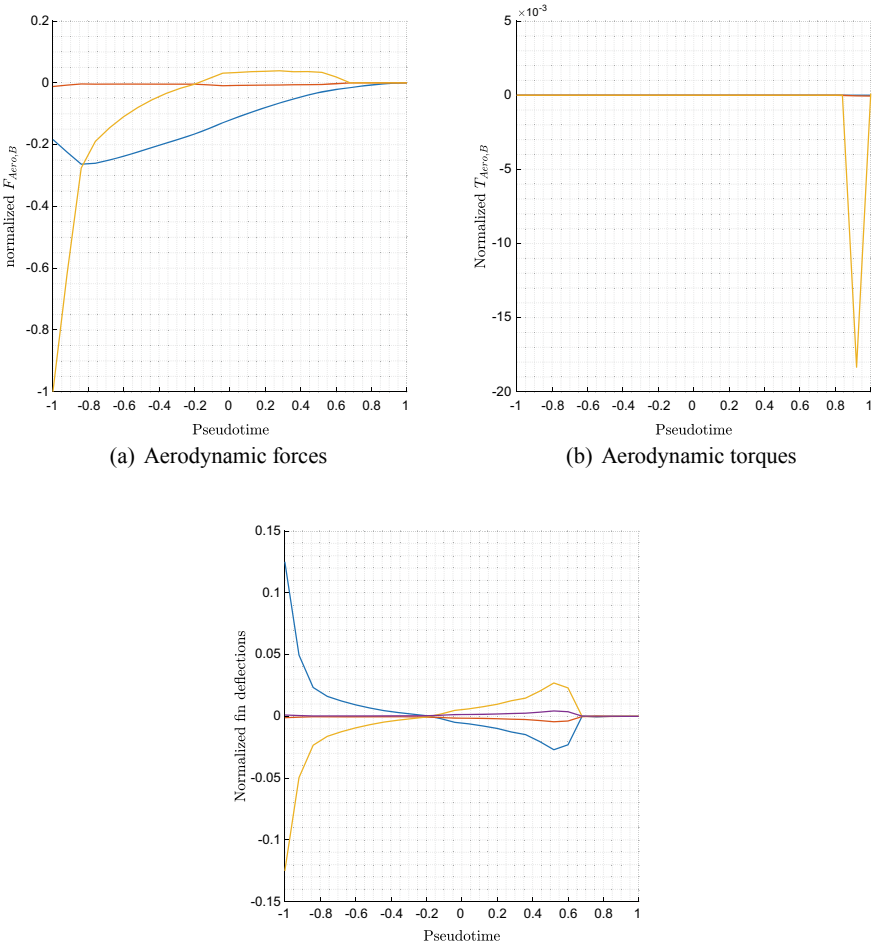


Fig. 5.18 Landing guidance solution—trajectory x-body is in red, y-body in green, z-body in blue

pressure, at the pseudotime equal to approximately 0.65 the fins are disabled and the TVC can continue to control the attitude until touchdown occurs. Finally, as depicted at the end of the scheme of Fig. 5.4, a validation through Runge-Kutta 45 is performed to verify that the obtained solution satisfies the full nonlinear equations of motion. The results are visible in Fig. 5.20a and b. Note that the solution perfectly matches the propagated one, with an error that in full scale is in the order of 0.02 m for the position components, and below 0.1 m/s for the velocity components. We can have a look at the convergence properties of the algorithm: (Fig. 5.21a–f). First, by looking at Fig. 5.21a we can see that no big changes are observed in the original cost function. This means that the mass consumption remains approximately the same, whereas the algorithm focuses on the refinement of the trajectory. This is confirmed by the augmented cost in Fig. 5.21b, where we can see that after iteration 2 no big variations occur anymore. The upper bounds on virtual controls (Fig. 5.21c) is constantly equal to 10^{-10}, meaning that the algorithm is always able to obtain a solution without actively leveraging the use of virtual controls for this specific scenario. Good convergence properties are visible also from the variation of the upper bounds on trust regions (Fig. 5.21d), which at the end of the iterative process is in the order of $2 \cdot 10^{-4}$. The same behavior is visible for the variations of t_F, shown



(a) Aerodynamic forces

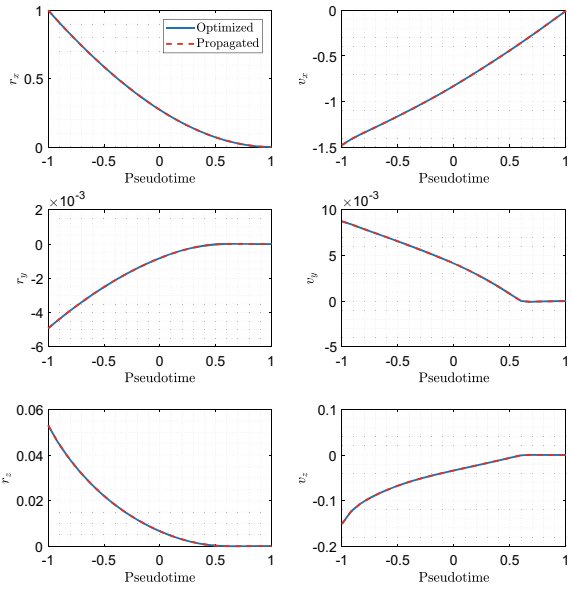
(b) Aerodynamic torques

(c) Fin deflections

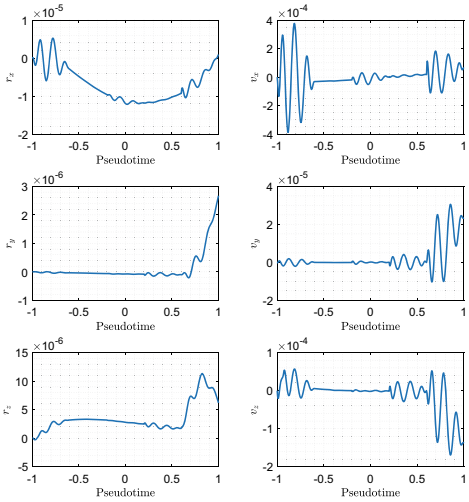
Fig. 5.19 Landing guidance solution: **a** Aerodynamic forces, **b** Aerodynamic torques, **c** Fin deflections

in Fig. 5.21e. After iteration 2 the variations on the final time are always smaller than 0.1 s, and become negligible after iteration 4.

The convergence process in terms of acceptance/rejection is depicted in Fig. 5.21f. All the solutions are accepted, confirming that the proposed approach shows good feasibility. Finally, the convergence behavior can be also seen in Fig. 5.22, where the colormap moves from blue to red as the number of iterations goes from the first to the last iteration. All the states quickly converge to the final solution. This figure shows that the initialization strategy correctly captures most of the behavior, simplifying the work of the SPCP algorithm. Note however, that whenever this is not the case, the SPCP was nevertheless able to correct the violations, as previously shown in [21].

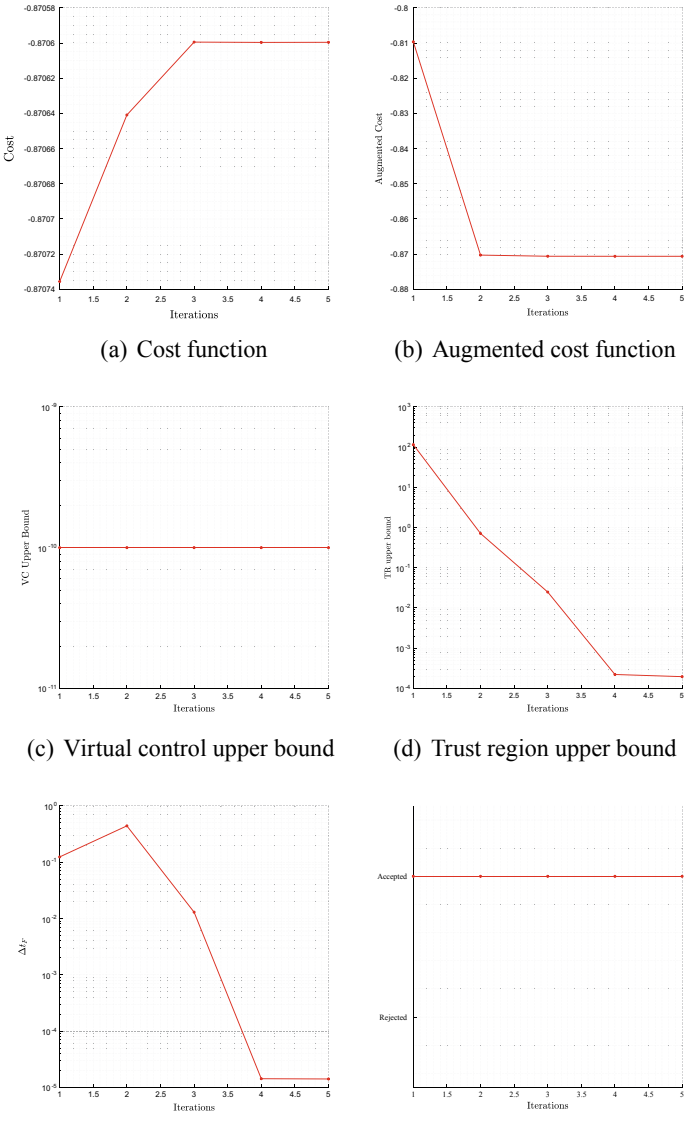


(a) Translational states



(b) State errors

Fig. 5.20 Nominal solution validation: **a** Comparison of states, and **b** Difference between optimized and propagated states



(a) Cost function (b) Augmented cost function

(c) Virtual control upper bound (d) Trust region upper bound

(e) t_F variation (f) Rejection / Acceptance

Fig. 5.21 Convergence behavior: **a** Cost function, **b** Augmented cost function, **c** Virtual control upper bound, **d** Trust region upper bound, **e** t_F variations, and **f** Acceptance/rejection policy

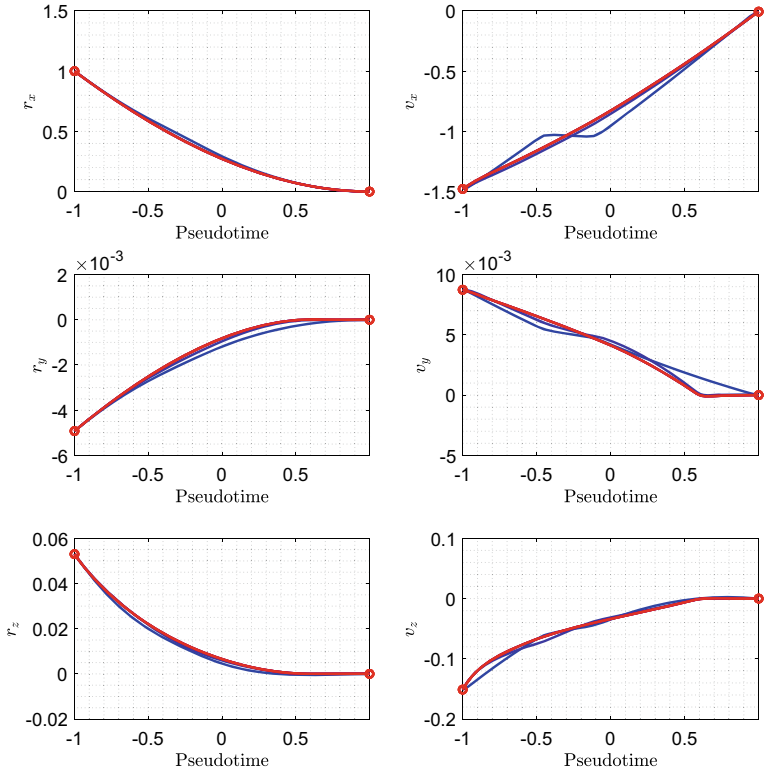


Fig. 5.22 Convergence behavior—translational states

5.6.4 Powered Landing—Dispersed Cases

Also in this case we did a preliminary analysis of the algorithm in presence of dispersions on the initial conditions in terms of position and velocity, as well as for the attitude. The errors on crossrange and downrange position are uniformly dispersed up to $\pm$ 50 m, while for the three velocity components the error is up to $\pm$5 m/s. Moreover, up to 2.5° error is added to the initial pitch and yaw angles. Figures 5.23 and 5.24 show the resulting trajectories, together with the translational and the rotational states. Moreover, the Runge-Kutta validation of the obtained solutions are depicted in Fig. 5.25a and b.

Also in this case all the trajectories fullfill the requirements (Fig. 5.23) and all the states and control limitations are satisfied (Fig. 5.24). Finally also in this case from the point of view of the accuracy of the solution we obtain a consistent ensemble of trajectories that accurately capture the nonlinear behavior of the system (Fig. 5.25a), and show a maximum error of approximately $7 \cdot 10^{-4}$ in position, and $4 \cdot 10^{-4}$ in velocity.

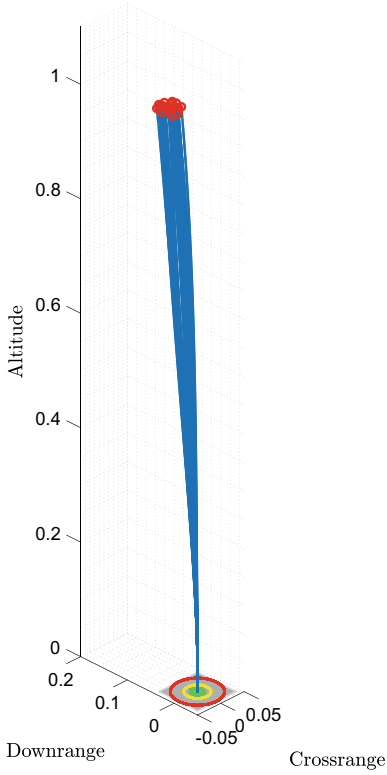
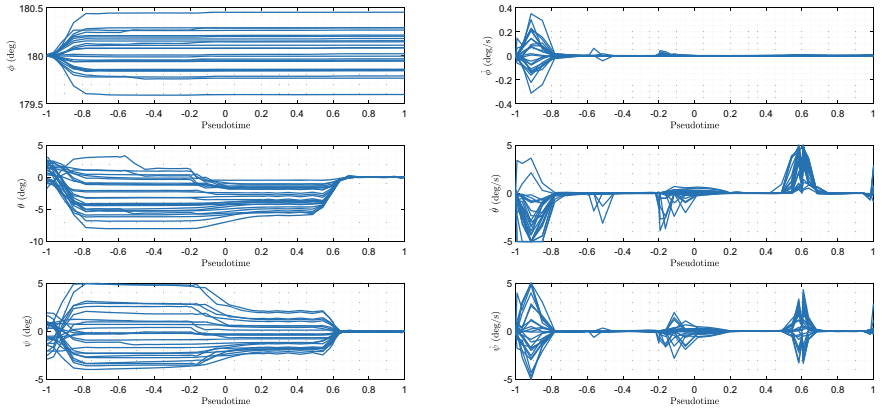


Fig. 5.23 Powered landing guidance solution—dispersed trajectories

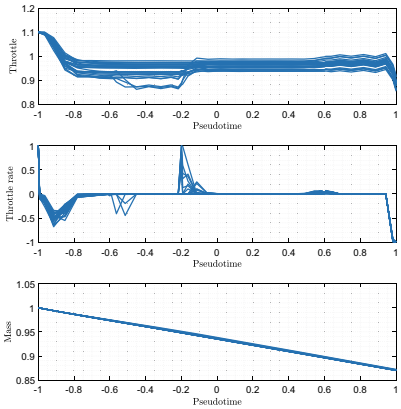
5.7 Conclusions

In this chapter we proposed an approach able to deal with both the aerodynamic descent and the powered landing phases of a reusable rocket. In the former case the control means are represented by the attitude of the vehicle with respect to the airspeed, which induces the aerodynamic forces that effectively drive the motion of the rocket during the unpowered descent, while in the latter these contributions are effects to be coupled with the thrust force, which is the main control means during the landing phase.

The approach exploited the distinction between convex and non-convex terms to come up with a convex sub-problem in which the need of numerical linearization is reduced to the sole non-convex terms. The reformulation of the problem in terms of rates of Euler angles allowed to express the system in affine form, with the corresponding benefits in terms of convergence behavior. Moreover, a transcription based on hp pseudospectral methods allowed on one side to improve the accuracy of the obtained solution, and on the other side to naturally formulate the free-final time version of the problem.



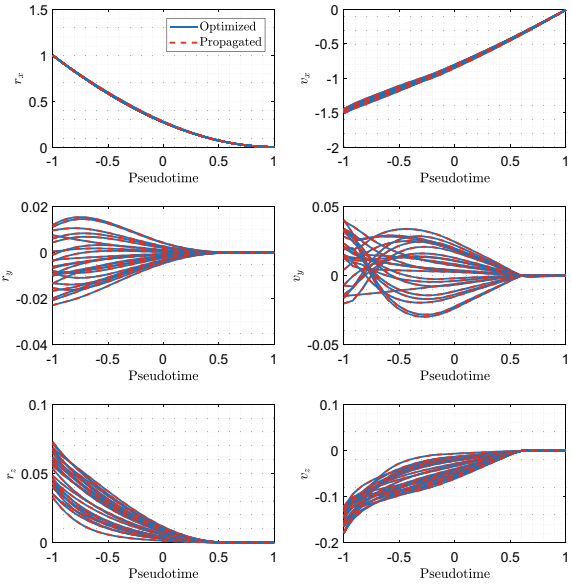
(a) Attitude states (b) Attitude rates



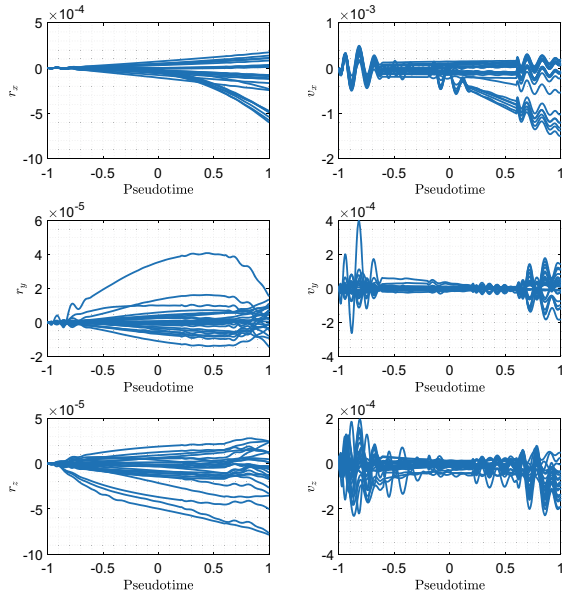
(c) Throttle, throttle rate and mass

Fig. 5.24 Powered landing guidance solution: **a** Attitude states, and **b** Attitude rates

Numerical results for nominal and dispersed condition confirm that the proposed approach is a viable method to quickly solve both the aerodynamic and landing guidance problems while providing at the same time a very accurate solution, with errors in the meter-range for what regards position, and less than 0.2 m/s in terms of velocity. The method is therefore a candidate technology to cover the entire descent guidance problem associated with the use of reusable rockets.



(a) Translational states



(b) State errors

Fig. 5.25 Dispersed solutions validation: **a** Comparison of states, and **b** Difference between optimized and propagated states

References

1. B. Acikmese, S.R. Ploen, Convex programming approach to powered descent guidance for mars landing. J. Guid. Control Dyn. **30**(5), 1353–1366 (2007)
2. A. Ben Tal, A. Nemirovski, *Modern Lectures on Convex Optimization*. Society for Industrial and Applied Mathematics (2001)
3. L. Blackmore, Autonomous precision landing of space rockets, in *National Academy of Engineering: The Bridge on Frontiers of Engineering*, vol. 4, issue 46 (2016), pp. 15–20
4. L. Blackmore, B. Acikmese, D.P. Scharf, Minimum-landing-error powered-descent guidance for mars landing using convex optimization. J. Guid. Control Dyn. **33**(4), 1161–1171 (2010). (July)
5. R. Bonalli, A. Cauligi, A. Bylard, M. Pavone, GuSTO: guaranteed sequential trajectory optimization via sequential convex programming, in *2019 International Conference on Robotics and Automation (ICRA)* (IEEE, 2019)
6. S. Boyd, L. Vandenberghe, *Convex Optimization* (Cambridge University Press, 2004)
7. E. Dumont, S. Ishimoto, P. Tatiossian, J. Klevanski, B. Reimann, T. Ecker, L. Witte, J. Riehmer, M. Sagliano, S. Giagkozoglou, I. Petkov, W. Rotärmel, R.G. Schwarz, D. Seelbinder, M. Markgraf, J. Sommer, D. Pfau, H. Martens, Callisto: a demonstrator for reusable launcher key technologies (2019)
8. E. Keith, The cost of reusability (space transportation launch vehicles), in *31st Joint Propulsion Conference and Exhibit* (American Institute of Aeronautics and Astronautics, 1995)
9. R. Lab, Neutron—the mega constellation launcher (2021)
10. X. Liu, Fuel-optimal rocket landing with aerodynamic controls. J. Guid. Control Dyn. 1–13(2018)
11. X. Liu, Z. Shen, L. Ping, Entry trajectory optimization by second-order cone programming. J. Guid. Control Dyn. **39**(2), 227–241 (2015)
12. X. Liu, Z. Shen, L. Ping, Entry trajectory optimization by second-order cone programming. J. Guid. Control Dyn. **39**(2), 227–241 (2016)
13. Y. Mao, M. Szmuk, B. Acikmese, Successive convexification of non-convex optimal control problems and its convergence properties, in *2016 IEEE 55th Conference on Decision and Control (CDC)* (IEEE, 2016)
14. A. Marwege, J. Riehmer, J. Klevanski, A. Guelhan, T. Ecker, B. Reimann, E. Dumont, First wind tunnel data of CALLISTO—reusable VTVL launcher first stage demonstrator (2019)
15. E. Mooij, *The Motion of a Vehicle in a Planetary Atmosphere* (Delft University Press, Delft, 1997)
16. B. Origin, New Glenn—our next (really) big step—an orbital reusable launch vehicle that will build the road to space (2021)
17. M. Sagliano, *Development of a Novel Algorithm for High Performance Reentry Guidance*. Ph.D. thesis (2016)
18. M. Sagliano, Pseudospectral convex optimization for powered descent and landing. J. Guid. Control Dyn. (2018)
19. M. Sagliano, S. Theil, M. Bergsma, V. D'Onofrio, L. Whittle, G. Viavattene, On the Radau pseudospectral method: theoretical and implementation advances. CEAS Space J. (2017)
20. M. Sagliano, Generalized hp pseudospectral-convex programming for powered descent and landing. J. Guid. Control Dyn. 1–9 (2019)
21. M. Sagliano, A. Heidecker, J.M. Hernández, S. Farì, M. Schlotterer, S. Woicke, D. Seelbinder, E. Dumont, Onboard guidance for reusable rockets: aerodynamic descent and powered landing, in *AIAA Scitech 2021 Forum* (American Institute of Aeronautics and Astronautics, 2021)
22. M. Sagliano, E. Mooij, Optimal drag-energy entry guidance via pseudospectral convex optimization **117**, 106946 (2021)
23. M. Sagliano, T. Tsukamoto, J.A. Maces-Hernandez, D. Seelbinder, S. Ishimoto, E. Dumont, Guidance and control strategy for the CALLISTO flight experiment, in *8th European Conference for Aeronautics and Aerospace Sciences (EUCASS)* (2019)

24. P. Simplício, A. Marcos, S. Bennani, Guidance of reusable launchers: improving descent and landing performance. J. Guid. Control Dyn. **42**(10), 2206–2219 (2019). (Oct)
25. spacenews.com. Falcon 9 launches cargo dragon, lands 100th booster (2021)
26. SpaceX. Starship sn15 flight test (2021)
27. M. Szmuk, B. Acikmese, Successive convexification for 6-DoF mars rocket powered landing with free-final-time, in *2018 AIAA Guidance, Navigation, and Control Conference. American Institute of Aeronautics and Astronautics* (2018)
28. M. Szmuk, U. Eren, B. Acikmese, Successive convexification for mars 6-DoF powered descent landing guidance, in *AIAA SciTech Forum* (American Institute of Aeronautics and Astronautics, 2017)
29. Z. Wang, M.J. Grant, Constrained trajectory optimization for planetary entry via sequential convex programming. J. Guid. Control Dyn. **40**(10), 2603–2615 (2017). (Oct)
30. Z. Wang, Y. Lu, Improved sequential convex programming algorithms for entry trajectory optimization. J. Spacecr. Rockets 1–14 (2020)
31. R. Yang, X. Liu, Fuel-optimal powered descent guidance with free final-time and path constraints. Acta Astronaut. **172**, 70–81 (2020). (Jul)

Chapter 6
Simultaneous Trajectory Optimization for Adaptive Powered Descent

Zhenyu Wei, Lin Ma, Kexin Wang, and Zhijiang Shao

6.1 Introduction

The powered descent phase is the terminal phase of the planetary descent and landing mission, where the required direction and magnitude of the thrust engine are determined to steer the spacecraft to meet the pre-determined targeting condition at the end, such as touchdown at the designated landing site with the required velocity [12]. Methods of powered descent guidance originated from the Apollo era and have been further promoted by later Mars landing missions. Typically, acceleration schemes, such as gravity turning and polynomial guidance algorithms [6, 19], were adopted to design and calculate an analytical thrust profile for soft landing. Such methods can hardly achieve precise landing, and its ability to perform in real time for avoiding constraints violation is limited. Besides, fuel optimality is usually ignored in the acceleration scheme. To overcome these shortcomings, powered descent methods with higher landing accuracy based on numerical optimization are proposed [18], including indirect methods by solving a two-point boundary value problem, direct methods based on solving nonlinear programming (NLP) or convex optimization (CO) problems [2, 4, 7]. Powered descent methods based on numerical

Z. Wei · L. Ma · K. Wang · Z. Shao (✉)
College of Control Science and Engineering in Zhejiang University, Hangzhou, People's Republic of China
e-mail: szj@zju.edu.cn

Z. Wei
e-mail: weizy@zju.edu.cn

L. Ma
e-mail: Ima@zju.edu.cn

K. Wang
e-mail: kxwang@zju.edu.cn

Z. Song et al. (eds.), *Autonomous Trajectory Planning and Guidance Control for Launch Vehicles*, Springer Series in Astrophysics and Cosmology,
https://doi.org/10.1007/978-981-99-0613-0_6

optimization essentially solve a trajectory optimization problem, which guarantees the feasibility of physical constraints, as well as fuel optimality.

Future planetary exploration missions will require spacecrafts to land in high-value scientific areas such as craters and boulder fields [5]. When encountering unexpected events or subsystem failure, the spacecraft needs to deflect the pre-determined landing trajectory to an alternative site [8]. The powered descent trajectory must be reconstructed during flight, posing new challenges to the guidance algorithm [15]. Benefited from computer hardware improvement, onboard computational guidance and control based on numerical optimization have been gradually becoming an important direction for powered descent methods [11]. This new trend enables a spacecraft to autonomously change its landing site and generate powered descent trajectories online.

This chapter explores the adaptive powered descent algorithms in two scenarios. The first scenario discusses adaptive powered descent in planetary exploration missions. Since the planetary exploration missions are usually carried out on the planet with a thin atmosphere, a three degree-of-freedom (DOF) translational model is formulated disregarding aerodynamic forces. With optimal sensitivity method, fuel consumption of multiple candidate landing sites is accurately estimated, which facilitates a rapid decision of optimal landing site. In the second scenario, a model of six DOF is established to depict dynamics of a reusable rocket because the dense earth atmosphere causes significant aerodynamic torques. In order to realize emergency landing under significant deviation from the nominal trajectory, a highway landing trajectory optimization problem is established. Through successive convexification, trajectory recovery can be implemented despite the large initial uncertainties. These two trajectory optimization algorithms are validated through simulations.

The remainder of this chapter is organized as follows. The multi-point powered descent algorithm for fuel optimality is presented in Sect. 6.2. The highway powered descent algorithm is detailed in Sect. 6.3. Finally, Sect. 6.4 gives conclusions.

6.2 Multi-point Powered Descent Based on Optimal Sensitivity

As a result of increased focus on the exploration of planets in the solar system, planetary powered descent is gaining renewed interest. The trajectory must be planned onboard because the state of the lander at the start of powered descent cannot be adequately predicted beforehand due to atmospheric uncertainties and/or the limits of deep space navigation [16, 17]. The multi-point powered descent algorithm aims to make a real-time decision that recognizes among a list of potential landing sites the one with a trajectory of least fuel consumption leading to it.

6.2.1 Problem Formulation

Since powered descent phase of a planetary landing mission starts at a low altitude relative to planet's radius, the uniform gravity assumption is appropriate. Other forces such as aerodynamic forces due to winds are neglected in optimal trajectory design and they are treated as disturbances [4]. a lumped mass rigid body model of the lander is used, where the translational dynamics are decoupled from rotational dynamics. This is a common assumption used in practice mainly because bandwidth of the attitude control authority is typically far higher than that of the translational one [1].

The translational dynamics of the lander are expressed in a surface fixed frame of reference illustrated in Fig. 6.1 as follows:

$$\begin{aligned} \dot{\boldsymbol{r}}(t) &= \boldsymbol{v}(t) \\ \dot{\boldsymbol{v}}(t) &= \boldsymbol{g} + \boldsymbol{T}(t)/m(t) \\ \dot{m}(t) &= -\alpha \left\| \boldsymbol{T}(t) \right\| \end{aligned} \tag{6.1}$$

where $\boldsymbol{r} = \begin{bmatrix} x & y & z \end{bmatrix}^{\mathrm{T}}$ is the position vector; $\boldsymbol{v} = \begin{bmatrix} v_x & v_y & v_z \end{bmatrix}^{\mathrm{T}}$ is the velocity vector; $\boldsymbol{g}$ is the constant gravitational acceleration vector of the planet; $\boldsymbol{T} = \begin{bmatrix} T_x & T_y & T_z \end{bmatrix}^{\mathrm{T}}$ is the thrust vector; m is the lander mass; and α is a positive constant describing the fuel consumption rate.

The magnitude of $\boldsymbol{T}$ is bounded as:

$$T_{\min} \leq \left\| \boldsymbol{T}(t) \right\| \leq T_{\max} \tag{6.2}$$

Onboard sensors for terrain-relative navigation generally require specific viewing orientations, which imposes a constraint on the lander orientation [1]. This may imply that the thrust direction should not deviate more than γ degrees from the positive x direction. This type of constraint can easily be expressed as follows:

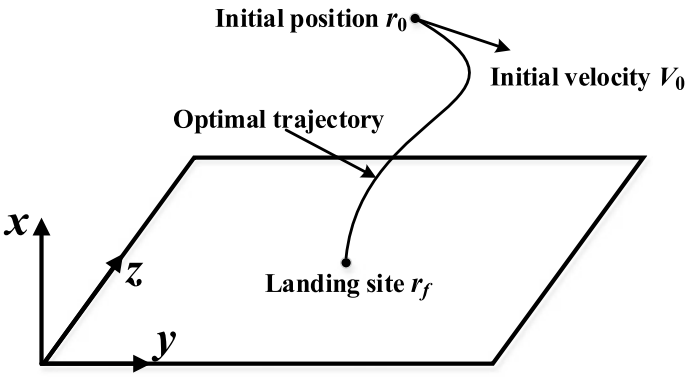


Fig. 6.1 Coordinate of powered descent process

$$\begin{aligned} &\hat{\boldsymbol{n}}^{\mathrm{T}}\boldsymbol{T}(t) \geq \|\boldsymbol{T}(t)\| \cos\gamma \\ &\hat{\boldsymbol{n}}^{\mathrm{T}} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \end{aligned} \tag{6.3}$$

where $\cos\gamma$ and $\hat{\boldsymbol{n}}$ describe the cone that the thrust vector should point into. Also, it is required that the trajectory does not go below the surface during the maneuver, namely:

$$x(t) \geq 0 \tag{6.4}$$

The mass of the fuel consumed cannot be more than the total mass of the fuel, hence the constraint is given by:

$$m_0 - m_{fuel} \leq m(t) \leq m_0 \tag{6.5}$$

where m_{fuel} is total mass of the fuel, and m_0 is the initial mass of the lander.

The initial and final position and velocity, and initial mass are specified as follows:

$$\begin{aligned} &m(0) = m_0 \\ &\boldsymbol{r}(0) = \boldsymbol{r}_0, \boldsymbol{r}(t_f) = \boldsymbol{r}_f \\ &\boldsymbol{v}(0) = \boldsymbol{v}_0, \boldsymbol{v}(t_f) = \boldsymbol{v}_f \end{aligned} \tag{6.6}$$

Therefore, we obtain the problem of powered descent as follows:

$$\begin{aligned} &\max_{T} m(t_f) \\ &s.t. \text{ Dynamics} : (6.1) \\ &\quad \text{PathConstraints} : (6.2) - (6.5) \\ &\quad \text{BoundaryConstraints} : (6.6) \end{aligned} \tag{6.7}$$

As shown in (6.7), the problem of single-point powered descent is established. For the problem of multi-point powered descent, it is straightforward to calculate the optimal trajectory point by point, then select the fuel-optimal landing site. However, the point-by-point approach is quite time-consuming, especially in the case of many candidate landing sites. Therefore, an effective and efficient multi-point powered descent algorithm is necessary.

6.2.2 Optimal Sensitivity

The finite-element collocation approach is chosen to discretize the state and control variables of a dynamic optimization problem as (6.7), leading to a large-scale NLP problem. We prefer Radau collocation points because they allow constraints to be set at the end of each element and to stabilize the system more efficiently if high index differential-algebraic equations are present. Without loss of generality, the following slacked NLP problem is obtained after discretization [3, 9]:

$$\begin{aligned} &\min_{\boldsymbol{x}} f(\boldsymbol{x};\boldsymbol{p}) \\ &s.t.\, \boldsymbol{c}(\boldsymbol{x};\boldsymbol{p}) = 0 \\ &\quad \boldsymbol{x} \geq 0 \end{aligned} \tag{6.8}$$

where the variable vector $\boldsymbol{x} \in \mathbb{R}^{n_x}$ consists of the discretized state, control and slack variables in formulation (6.7); The parameter vector $\boldsymbol{p} \in \mathbb{R}^{n_p}$ refers to the vector of perturbation parameters that corresponds to the position vector of candidate landing site; and $\boldsymbol{c}(\boldsymbol{x};\boldsymbol{p}) : \mathbb{R}^{n_x+n_p} \to \mathbb{R}^m$ are equality constraints.

The interior point method is used to solve the above NLP problem. The inequality constraints are transformed into a barrier term and added to the objective function. The following sequence of barrier subproblems is solved with $\mu \to 0$ [21]:

$$\begin{aligned} &\min_{\boldsymbol{x}} f(\boldsymbol{x};\boldsymbol{p}) - \mu \textstyle\sum_{i=1}^{n_x} \ln(x_i) \\ &s.t.\, \boldsymbol{c}(\boldsymbol{x};\boldsymbol{p}) = 0 \end{aligned} \tag{6.9}$$

The KKT conditions are defined as:

$$\begin{aligned} &\nabla_x L(\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{v};\boldsymbol{p}) = \nabla_x f(\boldsymbol{x};\boldsymbol{p}) + \nabla_x \boldsymbol{c}(\boldsymbol{x};\boldsymbol{p})\boldsymbol{\lambda} - \boldsymbol{v} = 0 \\ &\boldsymbol{c}(\boldsymbol{x};\boldsymbol{p}) = 0 \\ &x^{(i)} v^{(i)} = \mu \end{aligned} \tag{6.10}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^{n_\lambda}$ and $\boldsymbol{v} \in \mathbb{R}^{n_x}$ are vectors of the Lagrangian multipliers of the equality constraints and the bounds respectively; the superscript i denotes the ith component of a vector.

Under proper assumptions, we obtain the following theorem:

Theorem 6.1 ([14]) *Consider problem (6.8) with $f(\boldsymbol{x};\boldsymbol{p})$ and $\boldsymbol{c}(\boldsymbol{x};\boldsymbol{p})$ at least twice differentiable in $\boldsymbol{x}$. Let $\boldsymbol{x}^*$ be a local constrained minimizer of problem (6.8). The linear independence constraint qualification, strict complementarity, and strict second-order sufficient conditions hold at $\boldsymbol{x}^*$. Then, for barrier problem (6.9) with $\boldsymbol{p} = \boldsymbol{p}_0$ and $\mu \to 0$, a unique, continuous differentiable vector function $\boldsymbol{x}(\mu;\boldsymbol{p}_0)$ exists; the minimizer of problem (6.9) exists for $\mu > 0$ in a neighborhood of $\mu = 0$; $\lim_{\mu\to 0^+} \boldsymbol{x}(\mu;\boldsymbol{p}_0) = \boldsymbol{x}^*$; $\left\|\boldsymbol{x}(\mu;\boldsymbol{p}_0) - \boldsymbol{x}^*\right\| = O(\mu)$.*

Theorem 6.2 ([14]) *For barrier problem (6.9), assume that $f(\boldsymbol{x};\boldsymbol{p})$ and $\boldsymbol{c}(\boldsymbol{x};\boldsymbol{p})$ are k times differentiable in $\boldsymbol{p}$ and k+1 times differentiable in $\boldsymbol{x}$, and that the assumption of $\boldsymbol{x}^*$ in Theorem 6.1 hold. Then, for the solution of problem (6.9) with a small positive μ:*

1. *$\boldsymbol{x}(\mu;\boldsymbol{p}_0)$ is an isolated minimizer and the associated barrier multipliers and dual variables are unique.*
2. *For some $\boldsymbol{p}$ in a neighborhood of $\boldsymbol{p}_0$, a k times differentiable function exists as follows:*

$$\boldsymbol{s}(\mu;\boldsymbol{p})^{\mathrm{T}} = \left[\boldsymbol{x}(\mu;\boldsymbol{p})^{\mathrm{T}}\ \boldsymbol{\lambda}(\mu;\boldsymbol{p})^{\mathrm{T}}\ \boldsymbol{v}(\mu;\boldsymbol{p})^{\mathrm{T}}\right] \tag{6.11}$$

which corresponds to a locally unique minimum for problem (6.9) with the mul-

tiplier vector $\boldsymbol{\lambda}$ *and dual variable vector* $\boldsymbol{v}$.

3. *Solution* $\boldsymbol{s}(0;\, \boldsymbol{p}_0) \equiv \lim_{\mu\to 0,\, \boldsymbol{p}\to \boldsymbol{p}_0} \boldsymbol{s}(\mu;\, \boldsymbol{p}) = \boldsymbol{s}^*$.

Theorem 6.1 indicates that nearby solutions of problem (6.9) provide useful information for bounding properties for problem (6.8) for small positive values of μ. Theorem 6.2 considers the barrier formulation and sensitivity of its solution with respect to changes in values of $\boldsymbol{p}$. Based on Theorems 6.1 and 6.2, the implicit function theorem can be applied to formulation (6.10) at $\boldsymbol{p}_0$. We define the quantities:

$$
\begin{aligned}
S_s(\boldsymbol{s}(\mu;\, \boldsymbol{p}_0)) &= \begin{bmatrix} \nabla_{xx} L(\boldsymbol{s}(\mu;\, \boldsymbol{p}_0)) & \nabla_x \boldsymbol{c}(\boldsymbol{x}(\mu;\, \boldsymbol{p}_0)) & -\boldsymbol{I}_{nx} \\ \nabla_x \boldsymbol{c}(\boldsymbol{x}(\mu;\, \boldsymbol{p}_0))^{\mathrm{T}} & 0 & 0 \\ V(\mu;\, \boldsymbol{p}_0) & 0 & \boldsymbol{X}(\mu;\, \boldsymbol{p}_0) \end{bmatrix} \\
S_p(\boldsymbol{s}(\mu;\, \boldsymbol{p}_0)) &= \begin{bmatrix} \nabla_{xp} L(\boldsymbol{s}(\mu;\, \boldsymbol{p}_0)) \\ \nabla_p \boldsymbol{c}(\boldsymbol{x}(\mu;\, \boldsymbol{p}_0)) \\ 0 \end{bmatrix} \\
S_\mu(\boldsymbol{s}(\mu;\, \boldsymbol{p}_0)) &= \begin{bmatrix} 0 \\ 0 \\ -\boldsymbol{I}_{nx} \end{bmatrix}
\end{aligned}
\tag{6.12}
$$

where $\boldsymbol{X} = diag\{\boldsymbol{x}\}$, $\boldsymbol{V} = diag\{\boldsymbol{v}\}$, and $\boldsymbol{I}$ is the identity matrix. $S_s(\boldsymbol{s}(\mu;\, \boldsymbol{p}_0))$ and $S_p(\boldsymbol{s}(\mu;\, \boldsymbol{p}_0))$ are solved numerically. If the assumptions of Theorem 6.1 hold, $S_s(\boldsymbol{s}(\mu;\, \boldsymbol{p}_0))$ is non-singular, and the sensitivity $d\boldsymbol{s}(\mu;\, \boldsymbol{p}_0)^{\mathrm{T}}/d\boldsymbol{p}$ can be calculated from:

$$
\frac{d\boldsymbol{s}(\mu;\, \boldsymbol{p}_0)^{\mathrm{T}}}{d\boldsymbol{p}} = -S_s(\boldsymbol{s}(\mu;\, \boldsymbol{p}_0))^{-1} S_p(\boldsymbol{s}(\mu;\, \boldsymbol{p}_0)) \tag{6.13}
$$

For small value of μ and $\|\boldsymbol{p} - \boldsymbol{p}_0\|$, expand $\boldsymbol{s}(\mu;\, \boldsymbol{p})$ at $(\mu;\, \boldsymbol{p}_0)$:

$$
\boldsymbol{s}(\mu;\, \boldsymbol{p}) = \boldsymbol{s}(\mu;\, \boldsymbol{p}_0) - S_s(\boldsymbol{s}(\mu;\, \boldsymbol{p}_0))^{-1} S_p(\boldsymbol{s}(\mu;\, \boldsymbol{p}_0))(\boldsymbol{p} - \boldsymbol{p}_0) + o(\boldsymbol{p} - \boldsymbol{p}_0) \tag{6.14}
$$

We can also have:

$$
\begin{aligned}
\boldsymbol{s}(0;\, \boldsymbol{p}) = \boldsymbol{s}(\mu;\, \boldsymbol{p}_0) - S_s(\boldsymbol{s}(\mu;\, \boldsymbol{p}_0))^{-1} [S_p(\boldsymbol{s}(\mu;\, \boldsymbol{p}_0))(\boldsymbol{p} - \boldsymbol{p}_0) + S_\mu(\boldsymbol{s}(\mu;\, \boldsymbol{p}_0))\mu] \\
+ o(\boldsymbol{p} - \boldsymbol{p}_0) + o(\mu)
\end{aligned}
\tag{6.15}
$$

6.2.3 Multi-point Guidance Algorithm

The main idea of the multi-point powered descent algorithm is to utilize the fuel optimal trajectory from the initial state of the lander to the centroids of the candidate landing sites to estimate the fuel optimal trajectories from the same initial state of the lander to each candidate landing site based on sensitivity analysis. In the sensitivity analysis, the nominal parameter vector $\boldsymbol{p}_0$ is the position of the centroid, and each perturbed parameter vector $\boldsymbol{p}$ is the position of one candidate landing site.

In terms of the optimal sensitivity theory above, the overall multi-point powered descent algorithm is given as follows:

Step1: Partition all candidate landing sites into clusters based on the modified K-means clustering method [13]. (Note that the following steps show the guidance algorithm implemented in one cluster. The proposed algorithm should be implemented for each cluster in practice.)

Step2: Obtain the fuel-optimal trajectory $\boldsymbol{s}_{optimal}(0;\, \boldsymbol{p}_0)$ from the initial position of the lander (x_0, y_0, z_0) to the centroid $\boldsymbol{p}_0$.

Step3: Estimate trajectories $\boldsymbol{s}_{estimated}(0;\, \boldsymbol{p})$ from the initial position of the lander (x_0, y_0, z_0) to the candidate landing sites $\boldsymbol{p}$ whose centroid is $\boldsymbol{p}_0$ based on sensitivity analysis as follows:

$$\boldsymbol{s}_{estimated}(0;\, \boldsymbol{p}) = \boldsymbol{s}_{optimal}(0;\, \boldsymbol{p}_0) - S_s(\boldsymbol{s}(0;\, \boldsymbol{p}_0))^{-1} S_p(\boldsymbol{s}(0;\, \boldsymbol{p}_0))(\boldsymbol{p} - \boldsymbol{p}_0) \quad (6.16)$$

Step4: Select the optimal landing site $\boldsymbol{p}^* = (x_{f^*}, y_{f^*}, z_{f^*})$ that consumes the least fuel based on Step 3.

Step5: Calculate the fuel-optimal trajectory from the initial position of the lander (x_0, y_0, z_0) to the optimal landing site $(x_{f^*}, y_{f^*}, z_{f^*})$, using $\boldsymbol{s}_{estimated}(0;\, \boldsymbol{p}^*)$ as the initial guess.

Step6: Output the optimal results. (For a couple of clusters, compare the best landing sites from all the clusters before outputting the best one.)

Theorem 6.3 *Given the accuracy of sensitivity analysis, the proposed multi-point powered descent algorithm has minimum total estimation error with modified K-means clustering method.*

Proof The optimal trajectory from the initial state of the lander to the defined centroid C_j, $j = 1, \ldots, K$ is $\boldsymbol{s}_{optimal}(0;\, \boldsymbol{p}_0)$, and the trajectories to each candidate landing site can be estimated by (6.16). The total estimation error is written as follows:

$$
\begin{aligned}
Error &= \sum_{i=1}^{N} Error_i \\
&= \sum_{j=1}^{K} \sum_{i \in Cluster(j)} \boldsymbol{s}_{optimal}(0;\, \boldsymbol{p}_i) - \boldsymbol{s}_{estimated}(0;\, \boldsymbol{p}_i) \\
&= \sum_{j=1}^{K} \textstyle\sum_{i \in Cluster(j)} \boldsymbol{s}_{optimal}(0;\, \boldsymbol{p}_i) - \{\boldsymbol{s}_{estimated}(0;\, \boldsymbol{p}_{j0}) \\
&\quad - S_s(\boldsymbol{s}(0;\, \boldsymbol{p}_{j0}))^{-1} S_p(\boldsymbol{s}(0;\, \boldsymbol{p}_{j0}))(\boldsymbol{p} - \boldsymbol{p}_{j0})\} \\
&= \sum_{j=1}^{K} \sum_{i \in Cluster(j)} o\Big(\left\| \boldsymbol{p}_i - \boldsymbol{p}_{j0} \right\| \Big)
\end{aligned}
\tag{6.17}
$$

To minimize the total estimation error, we have:

$$
\min\ Error = \min \sum_{j=1}^{K} \sum_{i \in Cluster(j)} o(\left\| \boldsymbol{p}_i - \boldsymbol{p}_{j0} \right\|) \tag{6.18}
$$

which is equivalent to

$$
\min_{\boldsymbol{p}_{j0}} \sum_{j=1}^{K} \sum_{i \in Cluster(j)} o(\left\| \boldsymbol{p}_i - \boldsymbol{p}_{j0} \right\|) \tag{6.19}
$$

where the optimal $\boldsymbol{p}_{j0}$, $j = 1, \ldots, K$ are consistent with the definition of the centroids of the modified K-means clustering method.

6.2.4 Simulation Results

In Scenario 1, the initial position of the lander in the figures is (2500, 150, 150)m, and the initial velocity of the lander is $(-10, -50, 10)$m/s. One cluster is obtained based on the modified K-means clustering method, and the position of the centroid of these 36 candidate landing sites is (0.5, 0, 0)m. In the following figures, A denotes the initial position of the lander; D_1 to D_{36} denote the candidate landing sites from No. 1 to No. 36; C refers to the centroid of these candidate landing sites.

Figure 6.2 shows the results obtained by the proposed multi-point powered descent algorithm. First, the fuel-optimal trajectory from A to the centroid C is obtained as shown in Fig. 6.2a. Then, the estimated trajectories expressed by the red lines in Fig. 6.2b and c from A to each candidate landing site are obtained based on the sensitivity analysis. Fig. 6.2c is the vertical view of Fig. 6.2b. The fuel-optimal trajectory from these estimated trajectories is selected from the estimation results. In this scenario, the estimated trajectory from A to D_1 is fuel-optimal, thus D_1 is the best landing

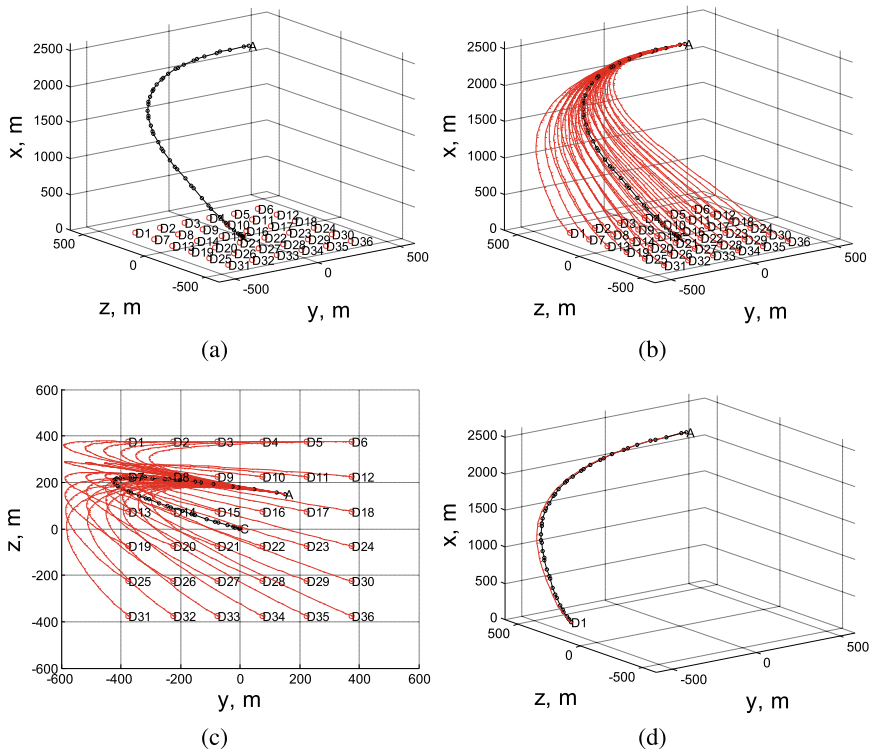


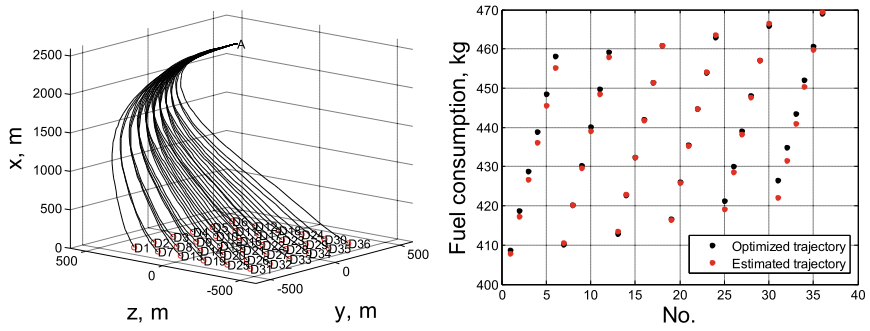
Fig. 6.2 Results by the proposed multi-point powered descent algorithm (Scenario 1)

site. Finally, the trajectory from A to D_1 is optimized based on the initial value given by the estimated trajectory and is shown in Fig. 6.2d.

Figure 6.3a shows the optimized trajectories from the initial position A to each candidate landing site. The trajectories are optimized separately for destinations ranging from D_1 to D_{36}. Then, the best trajectory that consumes the least fuel is selected from the resulting 36 trajectories. Figure 6.3b shows the results of fuel consumption of the optimized trajectories in Fig. 6.3a and the estimated trajectories in Fig. 6.2b. It is obvious that the sensitivity resulted fuel consumption approximates the results by actual optimization very well, and D_1 is the truly the best landing site, which verifies the validity of the proposed algorithm.

The performance of the proposed approach compared with that of the general approach (point by point) for Scenario 1 is presented in Table 6.1 This table includes the number of optimization runs (#nor), total CPU time taken in solving the problem (#Total CPU), and CPU time for sensitivity calculation (#Sen CPU). Apparently, the proposed approach is much better for real-time applications.

The sub-figures in Fig. 6.4 show the optimal state and control profiles from the initial position of the lander to the best landing site D_1. Figure 6.2d shows the



(a) (b)

Fig. 6.3 Comparison with actual optimization results (Scenario 1)

Table 6.1 Performance comparison of Scenario 1

Approach	#nor	#Total CPU, s	#Sen CPU, s
General approach	36	17.4	0
Proposed approach	2	2.5	0.2

three-dimensional position of the lander with respect to time. The lander finally lands at the specified landing site (0.5, −375, 375)m. Figure 6.4a gives the velocity vector of the lander with respect to time, showing that the velocity vector eventually equals zero when the lander lands on the specified position of the planetary surface. Figure 6.4b shows the thrust magnitude profile with respect to time. The thrust magnitude changes between the maximum and minimum. It can be seen that the optimal thrust control consists of a period of minimum thrust after the mission starts, then a period of maximum thrust until touchdown. As shown in Fig. 6.4c, the thrust pointing satisfies the corresponding requirements.

In Scenario 2, the initial position of the lander is (2500, −500, 500)m, and the initial velocity of the lander is (−10, 40, −40) m/s. Based on the modified K-means clustering method, three clusters are obtained, and the positions of the centroids are (0.5, −186.875, 440)m denoted by C_1, (0.5, −1068.75, −725)m denoted by C_2, and (0.5, 1126.25, 630)m denoted by C_3, respectively. In the following figures, A denotes the initial position of the lander; D_1 to D_{48} denote the candidate landing sites from No. 1 to No. 48. The centroid of D_1 to D_{24} is C_1; the centroid of D_{25} to D_{36} is C_2; and the centroid of D_{37} to D_{48} is C_3.

Figure 6.5 shows results obtained by the proposed multi-point algorithm. First, the fuel-optimal trajectories from A to the centroids C_1, C_2, and C_3 are respectively obtained as shown in Fig. 6.5a. Then, the estimated trajectories expressed by the red lines in Fig. 6.5b and c from A to each candidate landing site are obtained based on the sensitivity analysis. Figure 6.5c is the vertical view of Fig. 6.5b. The trajectories to sites from D_1 to D_{24} are estimated by the optimal trajectory from A to C_1; the

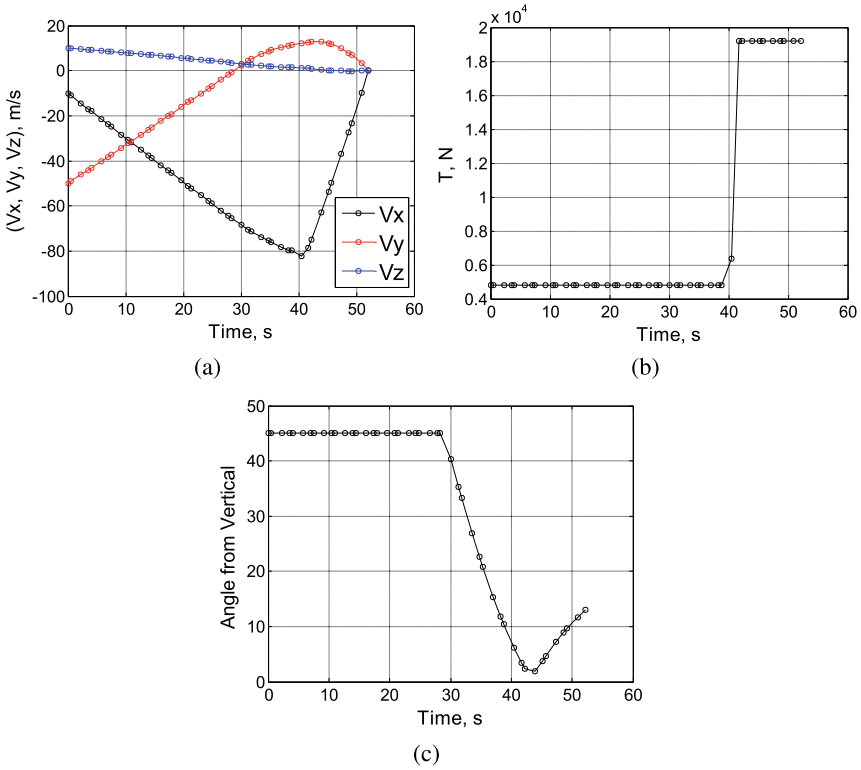


Fig. 6.4 Optimal state and control profiles (Scenario1)

trajectories to sites from D_{25} to D_{36} are estimated by the optimal trajectory from A to C_2; and the trajectories to sites from D_{37} to D_{48} are estimated by the optimal trajectory from A to C_3. The fuel-optimal trajectory is selected from these estimated trajectories. In this scenario, the estimated trajectory from A to D_{25} is fuel-optimal, thus D_{25} is the best landing site. Finally, the trajectory from A to D_{25} is optimized based on the initial guess given by the estimated trajectory and is shown in Fig. 6.5d.

Figure 6.6a shows the optimized trajectories from the initial position A to each candidate landing site. The trajectories are optimized for destination D_1 to D_{48} point by point. Then, the best trajectory that consumes the least fuel is selected from the resulting 48 trajectories. Figure 6.6b shows the results of fuel consumption of the optimized trajectories in Fig. 6.6a and the estimated trajectories in Fig. 6.5b. Again, fuel consumption resulted by the two algorithms are nearly the same, and D_{25} is fuel optimal. Performance comparison of these two algorithms in scenario 2 is presented in Table 6.2. In this case of multiple clusters, the proposed algorithm is still suitable for online applications.

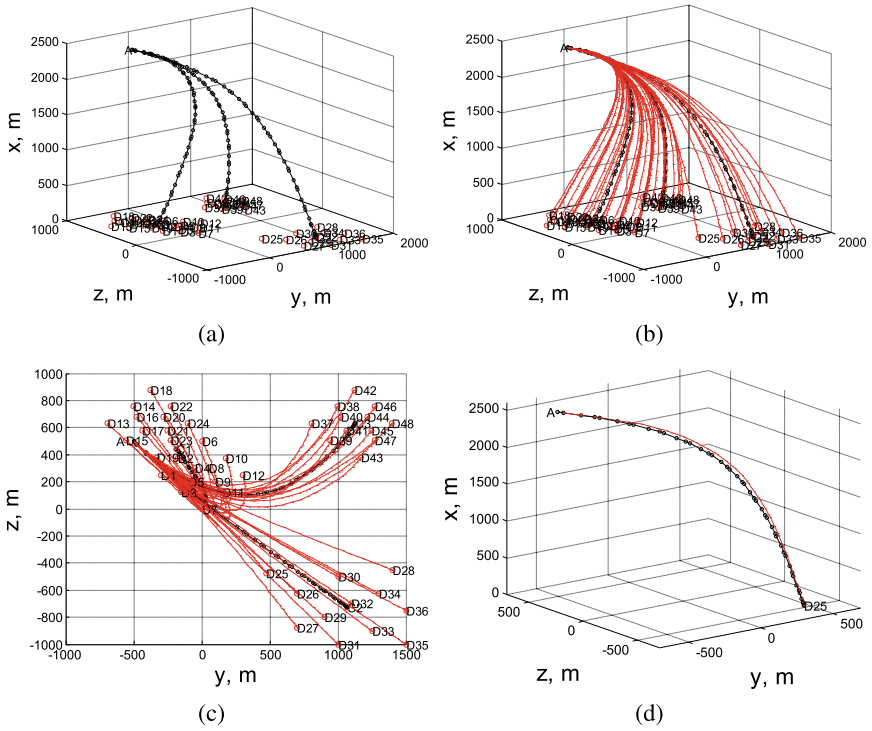


Fig. 6.5 Results by the proposed multi-point powered descent algorithm (Scenario 2)

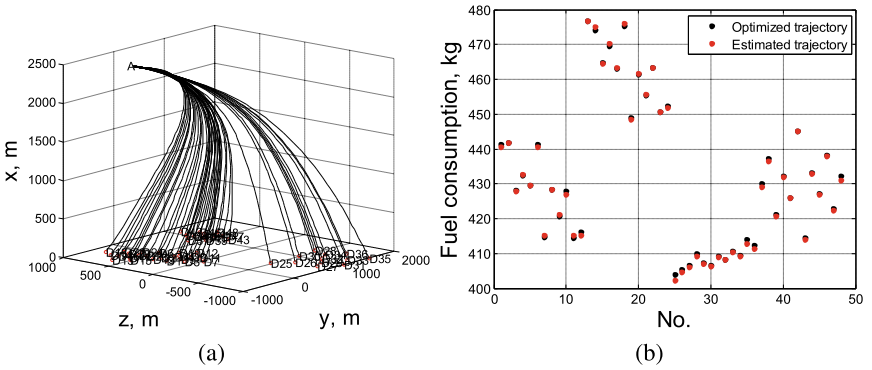


Fig. 6.6 Comparison with actual optimization results (Scenario 2)

Table 6.2 Performance comparison of Scenario 2

Approach	#nor	#Total CPU, s	#Sen CPU, s
General approach	48	21.9	0
Proposed approach	4	4.5	0.3

6.3 Highway Powered Descent Based on Successive Convexification

In the whole flight, the powered landing phase is the last opportunity for a reusable rocket to correct the trajectory deviation [10]. However, this is not even possible when difficulties, such as abrupt environment changes, a system failure, etc., appear at this stage. In order to deal with such emergencies, we attempt to take advantage of the highways near the pre-specified landing zone, trying to land on some highway instead. The adaptive powered descent algorithm in this section is devised to reconstruct trajectories for highway landing autonomously, to help landing in extreme conditions.

6.3.1 Problem Formulation

Due to the dense atmosphere, the aerodynamic forces are not negligible and have to be incorporated in the dynamic formulation. Moreover, the rotational dynamics are applied to describe attitude changes, preventing a rocket from tip-over.

We define the quaternion $\boldsymbol{q}_{B\leftarrow I}$ to describe the attitude of ground coordinate relative to body coordinate:

$$\boldsymbol{q}_{B\leftarrow I} = \begin{bmatrix} \cos(\theta/2) \\ \sin(\theta/2)\hat{\mathbf{e}} \end{bmatrix} \tag{6.20}$$

where $\hat{\mathbf{e}}$ is the rotation axis of ground coordinate relative to body coordinate, θ is the rotation angle.

The six degree-of-freedom model of a reusable rocket can be expressed as:

$$\begin{aligned} \dot{m} &= -\tfrac{1}{I_{sp}g_0}\|\boldsymbol{T}_B\| \\ \dot{\boldsymbol{r}}_I &= \boldsymbol{v}_I \\ \dot{\boldsymbol{v}}_I &= \tfrac{1}{m}(\boldsymbol{T}_I + \boldsymbol{D}_I) + \boldsymbol{g} \\ \dot{\boldsymbol{q}}_{B\leftarrow I} &= \tfrac{1}{2}[\boldsymbol{\omega}_B\otimes]\,\boldsymbol{q}_{B\leftarrow I} \\ \mathbf{J}_B\dot{\boldsymbol{\omega}}_B &= \boldsymbol{M}_B - [\boldsymbol{\omega}_B\times]\,\mathbf{J}_B\boldsymbol{\omega}_B \end{aligned} \tag{6.21}$$

where m is the reusable rocket mass, I_{sp} is the vacuum-specific-impulse, $\boldsymbol{g}$ is the gravity vector; the position $\boldsymbol{r}_I$, velocity $\boldsymbol{v}_I$, thrust force $\boldsymbol{T}_I$ and drag force $\boldsymbol{D}_I$ are all expressed in terms of the ground coordinate; the angular velocity $\boldsymbol{\omega}_B$, rocket's moment of inertia $\mathbf{J}_B$, torque $\boldsymbol{M}_B$ are expressed in terms of the body coordinate. The transformation between coordinates is realized by direction cosine matrix $\boldsymbol{C}_{B\leftarrow I}$ and $\boldsymbol{C}_{I\leftarrow B}$, which are detailed in [22]. For any vector $\boldsymbol{\xi} \in \mathbb{R}^3$, we define matrixes $[\boldsymbol{\xi}\otimes]$ and $[\boldsymbol{\xi}\times]$ as in [20].

Assume the drag force $\boldsymbol{D}_I$ is proportional to the square of the velocity $\boldsymbol{v}_I$, and its direction is opposite to that of $\boldsymbol{v}_I$. Then, $\boldsymbol{D}_I$ is expressed as:

$$\boldsymbol{D}_I = 0.5 C_D S_{ref} \rho \left\| \boldsymbol{v}_I \right\| \boldsymbol{v}_I \tag{6.22}$$

where C_D is the drag coefficient, S_{ref} is the aerodynamic reference area, ρ is the atmosphere density.

Assume the aerodynamic center remains at the same position, the torque $\boldsymbol{M}_B$ is expressed as:

$$\boldsymbol{M}_B = [\mathbf{r}_P \times] \boldsymbol{D}_B + [\mathbf{r}_T \times] \boldsymbol{T}_B \tag{6.23}$$

where $\mathbf{r}_P$ is the position vector of drag force, $\mathbf{r}_T$ is the position vector of thrust force.

The fuel of the reusable rocket is limited:

$$m_{dry} \leq m(t) \tag{6.24}$$

The tilt angle of the reusable rocket is expressed as the angle between the x-axes of body coordinate and that of the ground coordinate. During the landing phase, the tilt angle is limited as:

$$\hat{n}_3^{\mathrm{T}} \boldsymbol{C}_{I \leftarrow B} \hat{n}_3 = 1 - 2(q_1^2 + q_2^2) \geq \cos \theta_{\max} \tag{6.25}$$

To ensure landing safety, the maximum angular rate is limited as:

$$\left\| \boldsymbol{\omega}_B \right\| \leq \omega_{\max} \tag{6.26}$$

As illustrated in Fig. 6.7, the trajectory is constrained in an area restricted by two inclined planes to prevent subsurface flight. The glide angle relative to the horizon plane is constrained:

$$\begin{aligned} & r_{I,z}(t) - r_{f,z} \geq \sin \theta_{gs} \sqrt{(d_i)^2 + (r_{I,z}(t) - r_{f,z})^2} \\ & d_i = \frac{a_i r_{I,x}(t) + b_i r_{I,y}(t) + c_i}{\sqrt{(a_i)^2 + (b_i)^2}} \end{aligned} \tag{6.27}$$

The magnitude of the thrust vector is bounded in the interval $[T_{\min}, T_{\max}]$, and the gimble angle of the engine is limited in $[0, \delta_{\max}]$:

$$T_{\min} \leq \left\| \boldsymbol{T}_B(t) \right\| \leq T_{\max} \tag{6.28}$$

$$\cos \delta_{\max} \left\| \boldsymbol{T}_B(t) \right\| \leq T_{B,z}(t) \tag{6.29}$$

The boundary states of the reusable rocket are constrained as:

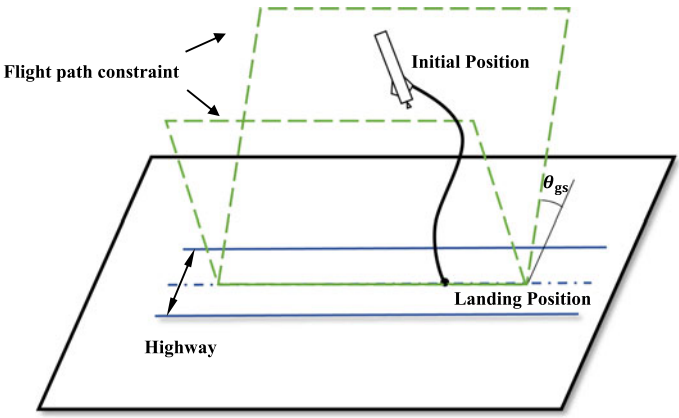


Fig. 6.7 Illustration of highway landing

$$\begin{cases} m(0) = m_{wet}, \boldsymbol{r}_I(0) = \boldsymbol{r}_i, \boldsymbol{v}_I(0) = \boldsymbol{v}_i \\ \boldsymbol{q}_{B\leftarrow I}(0) = \boldsymbol{q}_{B\leftarrow I,i}, \boldsymbol{\omega}_B(0) = \boldsymbol{\omega}_i \end{cases} \tag{6.30}$$

$$\begin{cases} a_i r_{I,x}(t_f) + b_i r_{I,y}(t_f) + c_i = 0 \\ x_{\min} \le r_{I,x}(t_f) \le x_{\max}, y_{\min} \le r_{I,y}(t_f) \le y_{\max} \\ r_{I,z}(t_f) = r_{f,z}, \boldsymbol{v}_I(t_f) = \boldsymbol{v}_f \\ \boldsymbol{q}_{B\leftarrow I}(t_f) = \boldsymbol{q}_{B\leftarrow I,f}, \boldsymbol{\omega}_B(t_f) = \boldsymbol{\omega}_f \end{cases} \tag{6.31}$$

where $a_i r_{I,x}(t_f) + b_i r_{I,y}(t_f) + c_i$ denotes the ith highway in the ground coordinate, which is available for the emergency landing of the reusable rocket. To describe the length of the highway, the x-axis interval $[x_{\min}, x_{\max}]$ and y-axis interval $[y_{\min}, y_{\max}]$ is defined.

Therefore, we obtain the problem of highway landing as follows:

$$\begin{aligned} &\max_{\boldsymbol{T}_B, t_f} t_f \\ &s.t. \text{ Dynamics} : (6.21) \\ &\quad \text{PathConstraints} : (6.24) - (6.29) \\ &\quad \text{BoundaryConstraints} : (6.30) - (6.31) \end{aligned} \tag{6.32}$$

The above problem is a non-convex optimization problem minimizing the flight time. The non-convexity lies in the nonlinear dynamics and the lower bound of the thrust magnitude. Once transformed into the convex optimization problem, this problem can be quickly solved by the interior-point method to obtain a trajectory that lands the reusable rocket on the highway.

6.3.2 Successive Convexification

We define the state variable and the control variable as:

$$\boldsymbol{x} = \left[m \ \boldsymbol{r}_I^{\mathrm{T}} \ \boldsymbol{v}_I^{\mathrm{T}} \ \boldsymbol{q}_{B\leftarrow I}^{\mathrm{T}} \ \boldsymbol{\omega}_B^{\mathrm{T}}\right]^{\mathrm{T}} \tag{6.33}$$

$$\boldsymbol{u} = \left[T_{B,x} \ T_{B,y} \ T_{B,z}\right]^{\mathrm{T}} \tag{6.34}$$

Normalized trajectory time is utilized to convert the problem into a fixed-final-time one:

$$\dot{\boldsymbol{x}}(\tau) \triangleq \frac{d}{d\tau}\boldsymbol{x}(\tau) = \sigma \boldsymbol{f}(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau)) \tag{6.35}$$

where $\boldsymbol{f}(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau))$ denotes the differential equation of kinematics and dynamics, $\tau \in [0, 1]$ is the normalized time, $\sigma = dt/d\tau$ denotes the time dilation coefficient.

Assume that a reference trajectory comprised of $\hat{\boldsymbol{x}}(\tau)$, $\hat{\boldsymbol{u}}(\tau)$, $\hat{\sigma}$ is given, then we can approximate the nonlinear dynamics with first-order Tayler series:

$$\begin{aligned}
\boldsymbol{x}'(\tau) &= \boldsymbol{A}(\tau)\boldsymbol{x}(\tau) + \boldsymbol{B}(\tau)\boldsymbol{u}(\tau) + \boldsymbol{C}(\tau) + \boldsymbol{D}(\tau) \\
\boldsymbol{A}(\tau) &= \hat{\sigma} \cdot \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau))}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}(\tau), \hat{\boldsymbol{u}}(\tau)} \\
\boldsymbol{B}(\tau) &= \hat{\sigma} \cdot \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau))}{\partial \boldsymbol{u}}\right|_{\hat{\boldsymbol{x}}(\tau), \hat{\boldsymbol{u}}(\tau)} \\
\boldsymbol{C}(\tau) &= -\boldsymbol{A}(\tau)\hat{\boldsymbol{x}}(\tau) - \boldsymbol{B}(\tau)\hat{\boldsymbol{u}}(\tau) \\
\boldsymbol{D}(\tau) &= \hat{\sigma}\boldsymbol{f}(\hat{\boldsymbol{x}}(\tau), \hat{\boldsymbol{u}}(\tau))
\end{aligned} \tag{6.36}$$

Similarly, the lower bound of the thrust magnitude can be approximated as:

$$T_{\min} \leq \frac{\hat{\boldsymbol{u}}^{\mathrm{T}}(\tau)}{\left\|\hat{\boldsymbol{u}}(\tau)\right\|}\boldsymbol{u}(\tau), \ \left\|\hat{\boldsymbol{u}}(\tau)\right\| \leq T_{\max} \tag{6.37}$$

Then the linearized problem is casted into a finite-dimensional convex optimization problem using fourth-order Runge–Kutta discretization. Without loss of generality, this can be described by the following second-order cone programming (SOCP) problem:

$$\begin{aligned}
&\min_{x} \ \boldsymbol{f}^{\mathrm{T}}\boldsymbol{x} \\
&s.t. \ \|\boldsymbol{A}_i\boldsymbol{x} + \boldsymbol{b}_i\| \leq \boldsymbol{c}_i^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{d}_i \\
&\quad \boldsymbol{F}\boldsymbol{x} = \boldsymbol{g}
\end{aligned} \tag{6.38}$$

The above SOCP problem approximates the original problem (6.32) near the reference trajectory. According to successive convexification method [20], a sequence

of SOCP subproblems can be solved to approach the optimal solution of (6.32). The procedure is as follows:

Step1: Initialize $\boldsymbol{x}_k^0$, $\boldsymbol{u}_k^0$ and σ_k^0 using (6.39). Set iteration number j = 0.

$$\begin{aligned} \boldsymbol{x}_k^0 &= \tfrac{K-1-k}{K-1}\boldsymbol{x}_i + \tfrac{k}{K-1}\boldsymbol{x}_f \\ \boldsymbol{u}_k^0 &= \left(\tfrac{K-1-k}{K-1}m_{wet} + \tfrac{k}{K-1}m_{dry}\right)\boldsymbol{g}_0 \\ \sigma^0 &= t_{guess} \end{aligned} \tag{6.39}$$

Step2: Set $\boldsymbol{x}_k^j$, $\boldsymbol{u}_k^j$ and σ_k^j as the reference trajectory, transform the original problem (6.32) using linearization and Runge-Kutta discretization.

Step3: Set iteration number j = j + 1, combine the trust-region and virtual control methods to solve the discretized problem (6.38). Update $\boldsymbol{x}_k^j$, $\boldsymbol{u}_k^j$ and σ_k^j.

Step4: Check the termination condition using (6.40). If the condition is satisfied, continue. Otherwise, go to step2.

$$\left(\left\|V^j\right\|_1 \le v_{tol} \ and \ \left\|\Delta^j\right\|_2 \le \Delta_{tol} \ and \ \Delta_\sigma \le \Delta_{\sigma,tol}\right) \ or \ (j > N_{\max}) \tag{6.40}$$

Step5: Output $\boldsymbol{x}_k^j$, $\boldsymbol{u}_k^j$ and σ_k^j as the solution of problem (6.32).

6.3.3 Highway Guidance Algorithm

The terminal status of a reusable rocket is limited in several aspects: (1) it must land on the pad or the highway, any landing position error that exceeds the safety tolerance may damage the rocket or the landing area; (2) it must impact the ground with small velocity to avoid damage in its structure; (3) it must touch the ground near vertically, in that large attitude angle may lead to tip-over. Therefore, we choose the landing position error Δr_f, terminal velocity v_f, and terminal attitude angle θ_f as landing performance indexes:

$$\begin{cases} \Delta r_f = \sqrt{(r_{f,x} - r_{p,x})^2 + (r_{f,y} - r_{p,y})^2} \\ v_f = \sqrt{v_{f,x}^2 + v_{f,y}^2 + v_{f,z}^2} \\ \theta_f = \arccos(1 - 2(q_1^2 + q_2^2)) \end{cases} \tag{6.41}$$

To get the landing performance index, linear interpolation is employed to get a continuous form of the control variable $\tilde{\boldsymbol{u}}$, then the trajectory is calculated under the open-loop control of $\tilde{\boldsymbol{u}}$. At last, we can get perform indexes using (6.41).

The proposed trajectory reconstruction algorithm considers both traditional pad landing and highway landing. When large deviation from the designed trajectory is detected, it calculates a pad landing trajectory at first. If the trajectory satisfies

landing performance tolerance, it proceeds to perform pad landing. Otherwise, it chooses an available highway to land on. The detailed procedure is as follows:

Step1: Calculate the pad landing trajectory;

Step2: Evaluate the pad landing performance. If the evaluation indexes satisfy (6.42), go to Step 8; Otherwise, continue.

$$\begin{cases} padcase : \Delta r_f \le r_{p,\max} \ and \ v_f \le v_{f,\max} \ and \ \theta_f \le \theta_{f,\max} \\ highwaycase : \Delta r_f \le r_{h,\max} \ and \ v_f \le v_{f,\max} \ and \ \theta_f \le \theta_{f,\max} \end{cases} \tag{6.42}$$

Step3: Get the number of available highways $N_{highway}$. Set k = 1.

Step4: Calculate the landing trajectory of kth highway.

Step5: Evaluate the landing performance of kth highway. If the indexes satisfy (6.42), go to Step 8; Otherwise, continue.

Step6: Calculate the comprehensive index E_k as follows. If $k < N_{highway}$, k = k +1, go to Step4; Otherwise, continue.

$$\begin{aligned} E_i &= \tau_1 g\left(\Delta r_f\right) + \tau_2 g\left(v_f\right) + \tau_3 g\left(\theta_f\right) \\ g\left(x\right) &= \frac{1}{1+e^{k(1-x/x_e)}} \end{aligned} \tag{6.43}$$

where τ_1, τ_2 and τ_3 are weight coefficients. $g\left(x\right)$ is the normalization function.

Step7: Choose the highway with the smallest E_i.

Step8: Output the trajectory result.

6.3.4 Simulation Results

The effectiveness of highway landing under large trajectory deviations is demonstrated first. We assume that there exists only one highway ranging from (−5000,0,0) to (5000,0,0) along the x-axis direction, and the landing pad lies in (0,0,0). The initial position of the reusable rocket deviates from (−656,54,4154) to (−3280,270,4154).

The nominal trajectory is denoted as trajectory1. With the given large deviation, the trajectories of the pad landing and the highway lading are denoted as trajectory2 and trajectory3, respectively.

The 3-D trajectory results under open-loop control are depicted in Fig. 6.8. Both trajectory1 and trajectory2 attempt to land the rocket on the pad, but trajectory2 violates the velocity and the attitude limit which may lead to tilt-over. Trajectory3 succeeds in landing the reusable rocket on position (−2679.54,0,0) of the highway. The flight time costs of the three trajectories are 25.25 s, 52.89 s, and 25.26 s. It can be seen from Fig. 6.9 that trajectory1 and trajectory3 share similar thrust level curve, while trajectory2 spends more thrust to land the reusable rocket. Therefore, in this case highway landing provides a better solution than pad landing under large trajectory deviation.

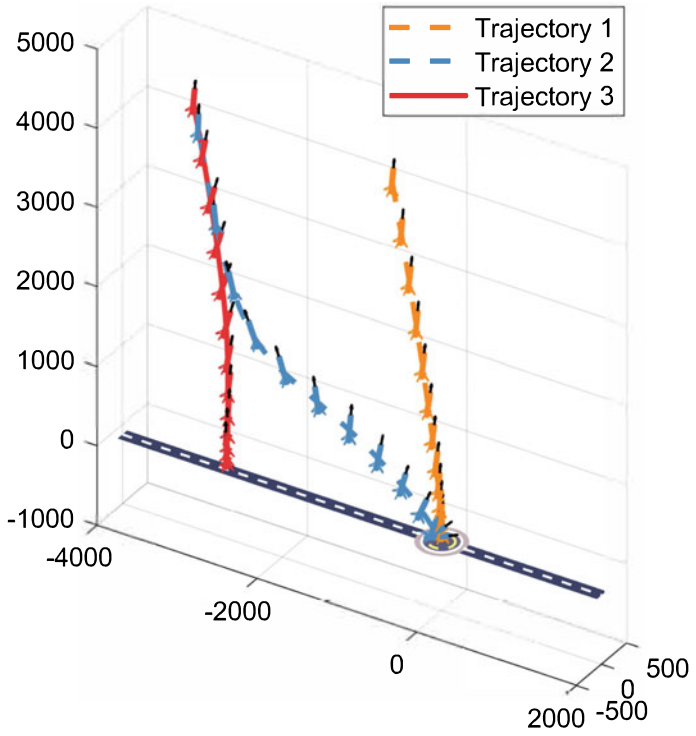


Fig. 6.8 3-D trajectory results

To validate optimality of the landing performance of trajectory3, among all the potential trajectories leading to different ends on the highway, we evenly choose 121 points locating from (−5000,0,0) to (5000,0,0) on the highway to compute landing trajectory. As shown in Fig. 6.10, the performance index of trajectory3 is marked in red, and the index of other trajectories are marked in green. It is observed that the landing position error of trajectory3 is within the highway landing tolerance, which guarantees an accurate landing. Also, the deviation value of trajectory3 is smaller than 86.8% (105/121) of other trajectories. The terminal velocity and the terminal attitude of trajectory3 outperform 96.7% (117/121) and 95.9% (116/121) of other trajectories, respectively. To summarize, the solution trajectory of the highway landing optimization problem has a better landing performance than most of the other trajectories.

The robustness of the trajectory reconstruction algorithm is verified by implementing 500-run Monte Carlo simulations. We assume that there exist three highways surrounding the landing pad. The terminal altitude of the rocket is set 30m above the ground, which tolerates the distance deviation along the altitude. The initial state parameters used are listed in Table 6.3.

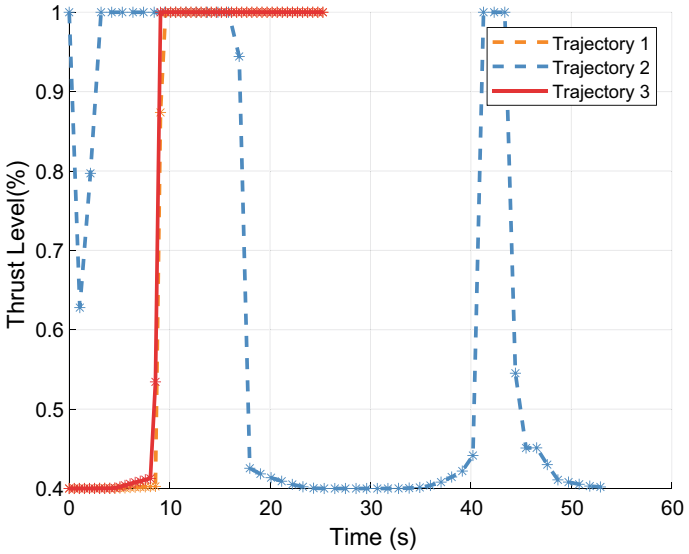


Fig. 6.9 Thrust profile

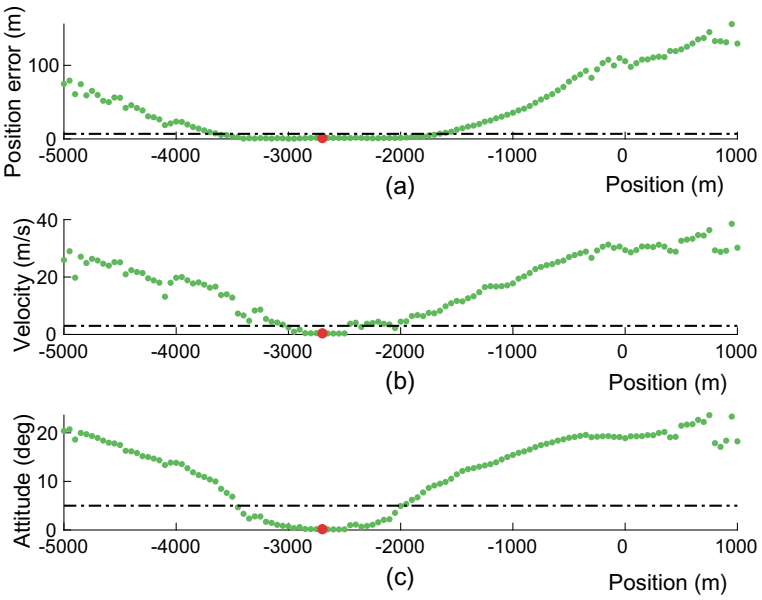


Fig. 6.10 Comparison results between the landing point and other points

Table 6.3 Monte Carlo simulation parameters

State variable	Initial value	Range
$\boldsymbol{r}_i$	$(-656, 54, 4154)$m	$(2000, 2000, 0)$m
$\boldsymbol{v}_i$	$(-63, -5, -310)$ m/s	$(10, 10, 10)$ m/s
$\boldsymbol{\theta}_{B\leftarrow I,i}$	$(0, -5, 0)^\circ$	$(5, 5, 5)^\circ$

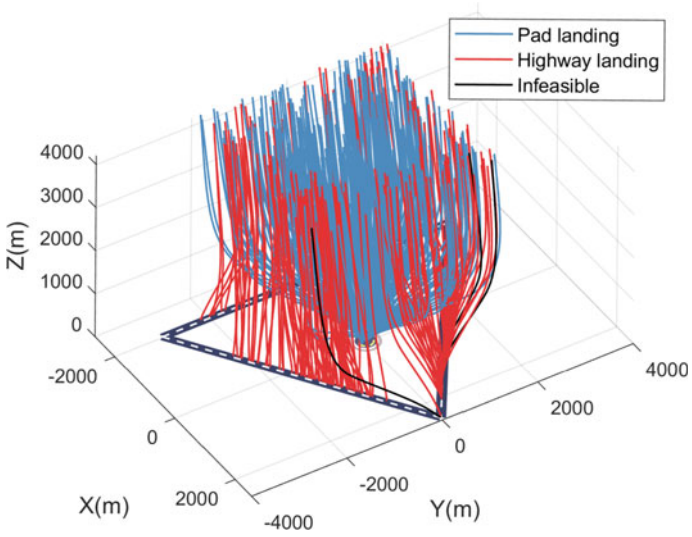


Fig. 6.11 3-D trajectory results

Figure 6.11 shows trajectory optimization results of the Monte Carlo simulations. The trajectories of pad landing, highway landing, and failure are marked in blue, red and black color, respectively. As the figure reveals, when the initial state deviation is small, pad landing is probably executed. If the state deviation exceeds the tolerable interval, highway landing is performed. Also, Fig. 6.11 shows that the reusable rocket tends to choose the nearest highway to land. Only in several extreme conditions, the trajectory reconstruction algorithm fails to find a feasible trajectory. Compare the proposed algorithm with the standard one, the success rate rises from 65% (326/500) to 99.4% (497/500). In conclusion, the trajectory reconstruction algorithm reduces dependence of generating a practical landing trajectory on small initial state perturbations, and improves the robustness of trajectory recovery by adding highway landing choices and solving the optimization problem in a sequential manner.

Figure 6.12 depicts fuel consumption comparison. Generally, highway landing trajectories use less fuel than pad landing trajectories, which can save more fuel for the guidance and control system to implement real-time adjustments.

The landing performance results are illustrated in Fig. 6.13. The threshold of pad landing is indicated by the dash-dotted line, and the threshold of highway landing

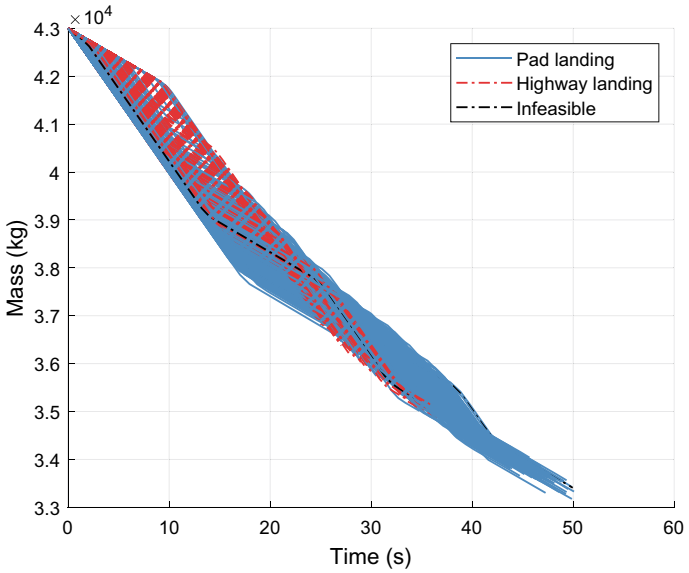


Fig. 6.12 Fuel consumption profiles

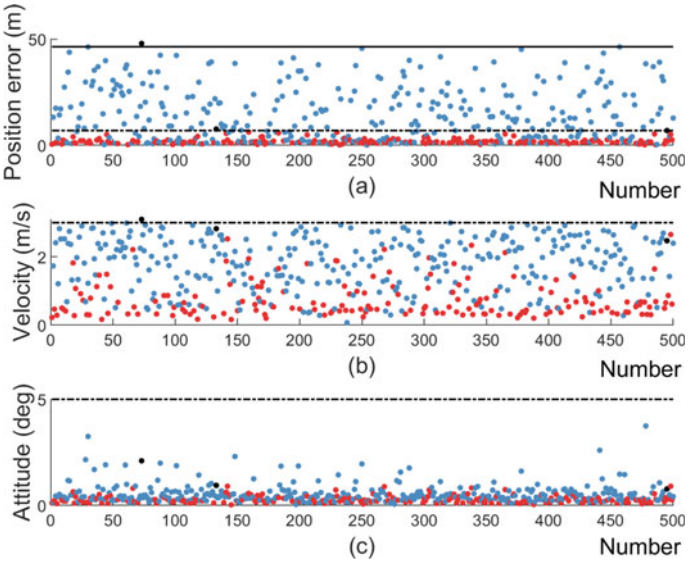


Fig. 6.13 Landing performances comparison

is denoted by the solid line. It is noted that 99.4% (497/500) of trajectories meet the performance requirements in landing position error, 99.8% (499/500) in terminal velocity, and 100% (500/500) in terminal attitude. In the failed cases, the maximum error in the landing position error exceeds 10.3% (0.71m), and the maximum error in terminal velocity exceeds 3.2% (0.096 m/s), of the threshold. Hence, the position error is the main factor affecting success.

6.4 Conclusions

This chapter develops algorithms and strategies to improve adaptive and autonomous decision making in powered decent of spacecrafts.

Firstly, a multi-point powered descent algorithm for planetary soft landing is presented. Here the best landing site is identified from many candidate landing sites efficiently by optimal sensitivity analysis. The proposed guidance algorithm establishes a relation between landing site selection and fast trajectory generation based on optimal sensitivity. The theoretical analysis shows that the proposed algorithm has minimum total estimation error. Numerical results indicate that the algorithm is capable of online applications.

Another powered descent algorithm is proposed for landing in large deviation from the nominal trajectory. Here highway landing trajectory optimization problem is proposed to handle the significant trajectory deviation and is solved by successive convexification technique. A trajectory reconstruction algorithm is devised to make decision between pad and highway landing in terms of the degree of trajectory deviation. The results of the Monte Carlo simulations show that the algorithm improves trajectory optimization under large uncertainties in the initial states.

References

1. B. Açıkmeşe, J.M. Carson, L. Blackmore, Lossless convexification of nonconvex control bound and pointing constraints of the soft landing optimal control problem. IEEE Trans. Control. Syst. Technol. **21**(6), 2104–2113 (2013)
2. B. Acikmese, S.R. Ploen, Convex programming approach to powered descent guidance for mars landing. J. Guid. Control Dyn. **30**(5), 1353–1366 (2007)
3. L.T. Biegler, An overview of simultaneous strategies for dynamic optimization. Chem. Eng. Process.: Process. Intensif. **46**(11), 1043–1053 (2007)
4. L. Blackmore, B. Açikmeşe, D.P. Scharf, Minimum-landing-error powered-descent guidance for mars landing using convex optimization. J. Guid. Control Dyn. **33**(4), 1161–1171 (2010)
5. H. Chu, L. Ma, K. Wang, Z. Shao, Z. Song, Trajectory optimization for lunar soft landing with complex constraints. Adv. Space Res. **60**(9), 2060–2076 (2017)
6. M.A. Connor, Gravity turn trajectories through the atmosphere. J. Spacecr. Rocket. **3**(8), 1308–1311 (1966)
7. F. Fahroo, I.M. Ross, Direct trajectory optimization by a chebyshev pseudospectral method. J. Guid. Control Dyn. **25**(1), 160–166 (2002)

8. X. Jiang, S. Li, T. Tao, Innovative hazard detection and avoidance strategy for autonomous safe planetary landing. Acta Astronaut. **126**, 66–76 (2016)
9. S. Kameswaran, L.T. Biegler, Simultaneous dynamic optimization strategies: recent advances and challenges. Comput. Chem. Eng. **30**(10–12), 1560–1575 (2006)
10. X. Liu, Fuel-optimal rocket landing with aerodynamic controls. J. Guid. Control Dyn. **42**(1), 65–77 (2019)
11. P. Lu, Introducing computational guidance and control (2017)
12. L. Ping, Theory of fractional-polynomial powered descent guidance. J. Guid. Control Dyn. **43**(3), 398–409 (2020)
13. L. Ma, K. Wang, Z. Xu, Z. Shao, Z. Song, L.T. Biegler, Multi-point powered descent guidance based on optimal sensitivity. Aerosp. Sci. Technol. **86**, 465–477 (2019)
14. H. Pirnay, R. López-Negrete, L.T. Biegler, Optimal sensitivity based on IPOPT. Math. Program. Comput. **4**(4), 307–331 (2012)
15. M. Sagliano, E. Mooij, S. Theil, Onboard trajectory generation for entry vehicles via adaptive multivariate pseudospectral interpolation. J. Guid. Control Dyn. **40**(2), 466–476 (2016)
16. D.P. Scharf, B. Açıkmeşe, D. Dueri, J. Benito, J. Casoliva, Implementation and experimental demonstration of onboard powered-descent guidance. J. Guid. Control Dyn. **40**(2), 213–229 (2017)
17. D.P. Scharf, S.R. Ploen, B.A. Acikmese, Interpolation-enhanced powered descent guidance for onboard nominal, off-nominal, and multi-x scenarios, in *AIAA Guidance, Navigation, and Control Conference* (2015), p. 0850
18. Yu. Song, X. Miao, S. Gong, Adaptive powered descent guidance based on multi-phase pseudospectral convex optimization. Acta Astronaut. **180**, 386–397 (2021)
19. R. Sostaric, J. Rea, Powered descent guidance methods for the moon and mars, in *AIAA Guidance, Navigation, and Control Conference and Exhibit* (2005), p. 6287
20. M. Szmuk, B. Acikmese, Successive convexification for 6-dof mars rocket powered landing with free-final-time, in *AIAA Guidance. Navigation, and Control Conference* (2018), p. 0617
21. A. Wächter, L.T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math. Program. **106**(1), 25–57 (2006)
22. Y. Yang, *Spacecraft Modeling, Attitude Determination, and Control Quaternion-based Approach: Quaternion-Based Approach* (CRC Press, 2019)

Chapter 7
Aerodynamic Parameter Estimation for Launch Vehicles

Jian Jia, Weifeng Chen, and Zixuan Wang

7.1 Introduction

7.1.1 What is Aerodynamic Parameter Estimation

Aerodynamic force plays an important role in the flight of space launch vehicles. Therefore, obtaining accurate aerodynamic characteristics is the basis and prerequisite for establishing an aerodynamic model and designing a vehicle with excellent characteristics.

Aerodynamic parameter identification is to establish the aerodynamic mathematical model reflecting the flight state of the vehicle and identify the coefficients based on the input and the measured output. Normally, the approaches for obtaining aerodynamic characteristics are theoretical calculation, wind tunnel test and the flight experiment. The method of theoretical calculation is based on the development of computer technology and aerodynamics, which is important for the initial design stage. The wind tunnel test is the basic method of aerodynamics research in recent years, which can establish the experimental database of aerodynamic model and provide data support for the establishment and development of aerodynamic model in the real application. The flight test is the closest way to the real flight mode, which can get the first-hand flight data and provide technical support for the aerodynamic

J. Jia · W. Chen (✉) · Z. Wang
College of Information Engineering in Zhejiang University of Technology, Hangzhou, People's Republic of China
e-mail: wfchen@zjut.edu.cn

J. Jia
e-mail: 2112103178@zjut.edu.cn

Z. Wang
e-mail: wzxkevin2022@163.com

Z. Song et al. (eds.), *Autonomous Trajectory Planning and Guidance Control for Launch Vehicles*, Springer Series in Astrophysics and Cosmology,
https://doi.org/10.1007/978-981-99-0613-0_7

model. However, these methods have certain defects. The theoretical calculation is hampered by limited theoretical knowledge and computing power of machine. The approximation error between the mathematical model and the real model cannot be eliminated. The accuracy of the established aerodynamic model is poor. The wind tunnel test uses the scaled-down model of real vehicles. The experiment results may be influenced by the tunnel size, structure and so on. Hence, the generated flight data may differ from the real flight data. The flight experiment is the closest to the real flight state. However, the cost of the launch experiment is quite high, and the aerodynamic parameter can only be indirectly estimated based on the motion measurement of the vehicle.

The process of aerodynamic parameter identification is a systematic process, including four aspects: [1]

(1) Experimental design. Through the experiment to obtain a sufficient amount of information and a sufficient number of experimental data.

(2) Aerodynamic model determination. A suitable mathematical model is constructed according to certain experience and guidelines.

(3) Aerodynamic parameter identification process. Determine a set of model parameter values based on the experimental data and the established model. Then, the numerical results calculated by the model can best fit the test data.

(4) Model check. Through the operation to check whether the established model conforms to the flight dynamics model of the vehicle.

There are many factors that affect the accuracy of aerodynamic parameter identification, such as sensor accuracy, wind speed influence and control system design. Also, there are many difficulties in the estimation algorithm design, in addition to the parameter estimation criterion. The most widely used methods for aerodynamic parameter identification are the least squares method, the Kalman filter, and the maximum likelihood estimation method.

7.1.2 Approaches for Aerodynamic Parameters Estimation

In system identification, least squares method is one of the most basic estimation methods that can be used for static systems as well as dynamic systems. In practical application, measurement data are often given in time order. In order to reduce the computation cost and the memory requirement, recursive least squares is usually used. The basic recursive least squares method has the following advantages:

(1) Unknown parameters can be easily found and the sum of the squares of the errors between the predicted and the actual measurements can be minimized.

(2) It gives the best parameter fit in a statistical sense when the measurement noise follows the Gaussian distribution and each measurement is independent.

However, the basic recursive least squares method has some drawbacks and limitations in dealing with the complicated system:

(1) It requires the calculation of inverse of matrix, which may not exist.

(2) If the number of samples is relatively large, the cost for computing the inverse matrix is large.

(3) It cannot be directly used for nonlinear system.

There are many works for identifying aerodynamic parameters by using least squares method. Zanette et al. [2] proposed a new calculation tool named RealSysId, which used the recursive orthogonal least square method for aerodynamic parameter identification. The recursive orthogonalization process obtained the matrices Q and R through the Givens rotation method. In this process, it was not necessary to store all the data matrices, which reduced the calculation amount of the algorithm and improved the real-time performance of system identification. Yang et al. [3] proposed a least-squares algorithm to identify the aerodynamic parameters of the projectile. The pathological problems existing in the traditional least squares method were effectively solved by dividing the whole launching process into three dynamic processes and establishing the aerodynamic model separately. Tang et al. [4] proposed a numerically robust least squares algorithm based on vector orthogonal polynomials. This method used matrix scores to describe the model. Expanding the numerator and denominator polynomial matrices on the basis of vector orthogonality, a very suitable numerical substitution was found. This method overcame the numerical problem of the least squares estimator. Guibert et al. [5] proposed an aerodynamic parameter identification method using constrained least squares. It was used in identifying the lift coefficients of the aircraft based on a new segmented polynomial model. The method improved computational accuracy compared to similar segmented models.

Kalman filtering is a recursive filtering method derived from the principle of linear unbiased minimum variance estimation. Chowdhary et al. [6] conducted an analysis and research on the unscented Kalman filter method used for aerodynamic parameter identification. This method could handle nonlinear systems and reduce approximation error of linearization. At the same time, it could speed up system convergence and improve the reliability of system estimation. When the system had considerable complexity, this method was better than the extended Kalman filter method in performance. Although it could be a good alternative to the extended Kalman filtering method, it would significantly increase the amount of calculations in the system. The method should be further explored and improved. Li et al. [7] used the aerodynamic parameter identification method of unscented Kalman filtering to verify that nonlinear models could be approximated by linear models. The state of the system was expanded by adding the estimated parameters. The application of this method proved that if the accuracy loss was ignored, the linear model could be used for the control design of the aircraft, and the control performance was well. This method could effectively improve the convergence speed and robustness of the system. Compared with the method proposed by Chowdhary et al., the method proposed by Li et al. used an augmented state unscented Kalman filter method, it was an improvement on the former method. This method proved that not only the nonlinear model could be used for system identification, but also the simplified linear model could be, and they had similar identification results. It could simplify the system modeling process, reduce costs, and shorten the design cycle. An adaptive unscented Kalman filtering method for aerodynamic parameters identification was proposed by Majeed

et al. [8]. It utilized two parallel unscented Kalman filters (UKF). The master UKF estimated the states and parameters using the noise covariance obtained by the slave UKF, while the slave UKF estimated the noise covariance using the innovations generated by the master UKF. The system could identify the system with unknown noise and determine the vehicle parameters in the uncertain environment where the noise characteristics change rapidly. However, the computational cost for this kind of two-level approach may be large and it may not be suitable for online application. Ding et al. [9] proposed a Bayesian adaptive unscented Kalman filter for aerodynamic parameter identification. This method combined Bayesian inference method and unscented Kalman filter method to jointly estimate the covariance coefficient of unknown noise. In the process of using the unscented Kalman filter, the Gauss-Newton method was used to maximize the posterior likelihood function of the unknown parameter estimation. This method had good filtering characteristics, high estimation accuracy and calculation efficiency. Compared with the adaptive unscented Kalman filter method proposed by Majeed et al., the Bayesian adaptive unscented Kalman filter method proposed by Ding et al. could reduce the calculation cost of the system and improve the calculation efficiency of the system. This method had better real-time performance and a wider range of applications. Zhang et al. [10] proposed a real-time estimation algorithm for unsteady aerodynamics. The Kalman filter was used in the algorithm based on the constant acceleration model to identify the unsteady terms in the aerodynamic parameter identification model. The non-Gaussian noise was dealt with by introducing the maximum correlation entropy criterion. The fast time-varying characteristics of the system could be effectively tracked and the accuracy of aerodynamic parameter estimation could be improved in this real-time estimation method. Although the Kalman filter method has been widely used in aerodynamic parameter identification, there are several disadvantages. The first is that the state model and noise statistical properties of the system must be known precisely, which is difficult in practice. When inaccurate or incorrect system models or noise statistical properties are used, the filtering can make the state estimation error large. Second, the computation cost of Jacobi matrices is heavy for high nonlinear system. Finally, the convergence of Kalman filter for nonlinear system should be improved.

In addition to the least squares and Kalman filter, there are many other kind of approaches for aerodynamic parameters estimation. In order to study the effect of icing on the tail of the aircraft on the flight status, Xu et al. [11] used the maximum likelihood estimation method. According to the flight data of the aircraft, the aerodynamic parameters of the normal model and the two interference models are identified. The identification results proved that the maximum likelihood estimation method was effective and also the lift and drag characteristics would be greatly affected by the icing on the tail of the aircraft. It provided a reference for the model improvement of the aircraft. Kumar et al. [12] estimated the aerodynamic parameters of the vehicle by analyzing flight data using traditional methods and neural network-based methods. The state of the system model should be estimated before system identification by the traditional estimation methods, while the neural network-based method had the advantage that a priori mathematical model was not necessary. Both methods had the similar estimation accuracy. Wang et al. [13] proposed an RBF

neural network that automatically added hidden nodes to identify the aerodynamic parameters of the reentry vehicle. The initial weights were optimized during neural network training, the identification accuracy was improved and the training time was shortened. At the same time, satisfactory identification results could be obtained by using this method to identify the parameters in the severely nonlinear state equation. Morelli [14] proposed a method for real-time estimation of aerodynamic parameters using flight data. More accurate modeling results could be provided without measuring the airflow angle of the aircraft. The airflow angle was reconstructed in the time domain model, after removing the constant deviation and drift from the time domain data, and then Fourier transform was applied. Real-time parameter estimation was performed in the frequency domain. In this identification method, the flight cost of the aircraft was effectively reduced, and the stability and safety of the aircraft were improved. An aerodynamic parameter identification method using traditional linear theory of nonlinear programming was proposed by Burchett [15]. After the linear theory solution was reformulated, the aerodynamic parameters and initial conditions were easier to be distinguished. The aerodynamic parameters and angular rate derivatives were obtained in a closed form. The derivatives were used to improve the gradient-based parameter estimation. Simulation results showed that the estimation accuracy of the system parameters could be effectively improved. Bagherzadeh et al. [16] proposed a global nonlinear aerodynamic parameter identification method. The amplitude and frequency models were decomposed and analyzed through an empirical mode decomposition algorithm. The non-linear behavior of the system could be effectively predicted and analyzed. The stability of the system also could be improved. In order to eliminate the error caused by the linearization of the model, an output error method proposed by Tu et al. [17] was used to identify the aerodynamic parameters of the nonlinear model. The nonlinear model with measurement noise was simulated and analyzed, and the influence of model error on the accuracy of the identification result was reduced. Dou et al. [18] proposed a method with a combination of maximum expectation (EM) and extended Kalman filtering (EKF) methods for aerodynamic parameter identification of the return segment of a reusable vehicle. The EKF was used to identify the system, which could effectively filter out the noise and estimate the unknown aerodynamic parameters, and the EM was used to estimate the a priori statistical distribution of measurement noise and process noise in the EKF process, which could reduce the influence of measurement and process noise on the system.

In this work, the aerodynamic parameter estimation for launch vehicles is considered based on simultaneous approach and maximum likelihood principle. Also, the prediction accuracy of the model with estimated values is focused on, not accuracy of the estimation for the aerodynamic parameters themselves.

7.2 Statistic Criterion Based Aerodynamic Parameter Estimation

The rocket is assumed to be a symmetric cylindrical shape without considering the side slip angle. In the velocity coordinate, the motion of the rocket can be described as follows, [19]

$$\begin{cases} \frac{dx}{dt} = \frac{R_e}{R_e+h} v\cos\theta \\ \frac{dh}{dt} = v\sin\theta \\ \frac{dv}{dt} = \frac{P_e\cos\alpha}{m} - \frac{\rho v^2 S_m C_D}{2m} - g\sin\theta \\ \frac{d\theta}{dt} = \frac{P_e\sin\alpha}{mv} + \frac{\rho v^2 S_m C_L}{2mv} - \frac{g\cos\theta}{v} + \frac{v\cos\theta}{R_e+h} \end{cases} \tag{7.1}$$

where v is the speed, P_e is the axial thrust, S_m is the rocket reference area, θ is the flight path angle, R_e is the radius of the earth, h is the flight altitude, m is the mass of the rocket, x is the distance of the rocket. The attack angle α is a control variable. The considered time span is $[0,\ t_f]$. The atmosphere density ρ is calculated based on Yang's model [20]. The drag coefficient C_D and lift coefficient C_L are considered as time dependent parameters. Simultaneous approach [21] is applied for aerodynamic parameter estimation and the motion model of rocket should be discretized. Formulation (7.1) can be converted into algebraic equations by approximating the state and control variables using a family of polynomials on finite elements $[t_{i-1},\ t_i]$, which satisfied

$$t_0 < t_1 < \cdots < t_N = t_f \tag{7.2}$$

where N is the number of finite elements. Lagrange interpolation polynomials are used as [21]

$$\begin{cases} x \approx x^{(K)} = \sum_{j=0}^{K} l_j(\tau)\, x_{ij},\ h \approx h^{(K)} = \sum_{j=0}^{K} l_j(\tau)\, h_{ij} \\ v \approx v^{(K)} = \sum_{j=0}^{K} l_j(\tau)\, v_{ij},\ \theta \approx \theta^{(K)} = \sum_{j=0}^{K} l_j(\tau)\, \theta_{ij} \\ \alpha \approx \alpha^{(K)} = \sum_{j=1}^{K} \bar{l}_j(\tau)\, \alpha_{ij} \\ l_j(\tau) = \sum_{k=0,\neq j}^{K} \frac{(\tau-\tau_k)}{(\tau_j-\tau_k)},\ \bar{l}_j(\tau) = \sum_{k=1,\neq j}^{K} \frac{(\tau-\tau_k)}{(\tau_j-\tau_k)} \end{cases} \tag{7.3}$$

where K is the number of collocation points. The drag coefficient C_D and lift coefficient C_L are approximated by piecewise linear function

$$\begin{cases} C_D \approx C_D^{(m-1)} + \dfrac{\left(C_D^{(m)} - C_D^{(m-1)}\right)}{\left(t_{i_m^D} - t_{i_{m-1}^D}\right)} \left(t - t_{i_{m-1}^D}\right), m = 1, \ldots, N_D \\ C_L \approx C_L^{(m-1)} + \dfrac{\left(C_L^{(m)} - C_L^{(m-1)}\right)}{\left(t_{i_m^L} - t_{i_{m-1}^L}\right)} \left(t - t_{i_{m-1}^L}\right), m = 1, \ldots, N_L \end{cases} \tag{7.4}$$

N_D and N_L are the number of segments for C_D and C_L. i_m^D and i_m^L are elements from $1...N$. It is assumed that the measured outputs are distance x, height h and velocity v. Based on Eqs. (7.1)–(7.4), the formulation for aerodynamic parameter estimation can be established. For simplifying the following discussion, the resulting nonlinear programming for aerodynamic parameter estimation is represented as follows,

$$\begin{aligned} &\min\ J = \tfrac{1}{2}(\mathbf{y} - \mathbf{y}^{(meas)})^T \mathbf{C}_y^{-1} (\mathbf{y} - \mathbf{y}^{(meas)}) \\ &s.t. \begin{cases} \mathbf{c}(\boldsymbol{\gamma}, \mathbf{p}) = 0 \\ \mathbf{y} = \mathbf{h}(\boldsymbol{\gamma}) \end{cases} \end{aligned} \tag{7.5}$$

where $\boldsymbol{\gamma} \in \boldsymbol{R}^{ng}$ represents the vector of distance x, height h, velocity v, flight path angle θ at all the discretization points. $\mathbf{c} : \mathbf{R}^{ng} \rightarrow \mathbf{R}^{ng}$ represents the discretization of (7.1) based on Eqs. (7.2)–(7.4) and $\mathbf{h} : \mathbf{R}^{ng} \rightarrow \mathbf{R}^{ny}$ represents the relationship between system output and state variables. $\mathbf{y} \in \mathbf{R}^{ny}$ represents the vector of distance x, height h, velocity v at all the sampling points. $\mathbf{y}^{(meas)} \in \mathbf{R}^{ny}$ represents the measurements vector corresponding to $\mathbf{y}$. $\mathbf{C}_y$ is a diagonal matrix, and each diagonal element is the variance of the measurement noise corresponding to $\mathbf{y}$. The Jacobian of $\mathbf{c}$ with respect to $\boldsymbol{\gamma}$ is a square matrix. For all physically meaningful parameters $\mathbf{p} = [{C_D}^{(1)}, {C_D}^{(2)}, \ldots, {C_D}^{(ND)}, {C_L}^{(1)}, {C_L}^{(2)}, \ldots, {C_L}^{(NL)}]^T$, the Jacobian is usually assumed to be non-singular, and it satisfies the linear independence constraint qualification (LICQ). The reduced Hessian of the Lagrange function for (7.5), denoted by $\mathbf{H}r$ which is calculated as [22]

$$\mathbf{Z} = \begin{bmatrix} \mathrm{I} \\ -\begin{bmatrix} \nabla_\gamma \mathbf{c}^T & \mathbf{0} \\ \nabla_\gamma \mathbf{h}^T & -\mathbf{I} \end{bmatrix}^{-1} \begin{bmatrix} \nabla_\gamma \mathbf{c}^T \\ \nabla_\gamma \mathbf{h}^T \end{bmatrix} \end{bmatrix}, \mathbf{H_r} = \left(\mathbf{Z}^T \begin{bmatrix} \mathbf{0}\,\mathbf{0} & \mathbf{0} \\ \mathbf{0}\,\mathbf{0} & \mathbf{0} \\ \mathbf{0}\,\mathbf{0} & \mathbf{C_y^{-1}} \end{bmatrix} \mathbf{Z} \right) \tag{7.6}$$

In this work, reduced Hessian based transformed parameter strategy [22] is used for eliminating the correlation among the aerodynamic parameters. The reduced Hessian $\mathbf{H}r$ is decomposed as

$$\mathbf{H_r} = \mathbf{V} \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_{N_D+N_L} \end{bmatrix} \mathbf{V}^T \tag{7.7}$$

where $\mathbf{V}$ is the orthogonal matrix and the $\lambda_1, \lambda_2, \ldots, \lambda_{ND+NL}$ with descending order are eigenvalues. The aerodynamic parameters $\mathbf{p}$ can be transformed as

$$\mathbf{q} = \mathbf{V}^T \mathbf{p} \tag{7.8}$$

where $\mathbf{q}$ is the vector of transformed parameters. Different from the mean squared error criterion [23], the number of estimated transformed parameters is determined based on the modified E-optimal design criterion [24] and $\mathbf{q}$ is divided into $[\mathbf{q_1}^T, \mathbf{q_2}^T]^T$, where only $\mathbf{q}_1$ are estimated. Usually, the unestimated parameters $\mathbf{q}_2$ are fixed at the nominal values $\mathbf{q_2}^{(trial)}$ and the following problem is solved,

$$\begin{aligned} &\min\ J = \tfrac{1}{2}(\mathbf{y} - \mathbf{y}^{(meas)})^T \mathbf{C}_y^{-1} (\mathbf{y} - \mathbf{y}^{(meas)}) \\ &s.t. \begin{cases} \mathbf{c}(\boldsymbol{\gamma}, \mathbf{p}) = 0 \\ \mathbf{y} = \mathbf{h}(\boldsymbol{\gamma}) \\ \mathbf{q} = [\mathbf{q}_1^T, \mathbf{q}_2^T]^T = \mathbf{V}^T \mathbf{p} \\ \mathbf{q}_2 = \mathbf{q}_2^{(trial)} \end{cases} \end{aligned} \tag{7.9}$$

However, unreasonable nominal values for the fixed parameters significantly affect the prediction of the model. Statistic criterion based approach is used for determining the nominal values for unestimated parameters and the statistic criterion is designed as [22]

$$\begin{aligned} &\eta = |skewness_1 - 0| + |kurtosis_1 - 3| \\ &+|skewness_1 - skewness_2| + |kurtosis_1 - kurtosis_2| \\ &skewness_1 = \frac{\frac{1}{ny}\sum_{i=1}^{ny} (\varepsilon_i - 0)^3}{\sigma^3}, skewness_2 = \frac{z_3}{z_2^{1.5}} \\ &kurtosis_1 = \frac{\frac{1}{ny}\sum_{i=1}^{ny} (\varepsilon_i - 0)^4}{\sigma^4}, kurtosis_2 = \frac{z_4}{z_2^2} \\ &\varepsilon_i = \mathbf{y}_i(\boldsymbol{\alpha}(\boldsymbol{\alpha}_2^{(trial)})) - \mathbf{y}_i^{(m)}, \bar{\varepsilon} = \frac{1}{ny}\sum_{i=1}^{ny} \varepsilon_i \\ &z_2 = \frac{1}{ny}\sum_{i=1}^{ny} (\varepsilon_i - \bar{\varepsilon})^2, z_3 = \frac{1}{ny}\sum_{i=1}^{ny} (\varepsilon_i - \bar{\varepsilon})^3, z_4 = \frac{1}{ny}\sum_{i=1}^{ny} (\varepsilon_i - \bar{\varepsilon})^4 \end{aligned} \tag{7.10}$$

where the subscript i in (7.10) represents the i-th component of the vector $\mathbf{y}$, ny is the dimension of $\mathbf{y}$ and the measurement noise follows Gauss probability distribution $N(0, \sigma^2)$. The statistic criterion based approach [22] combined with the modified E-optimal design criterion for estimating the aerodynamic parameters are described as

Step 1: Let $\mathbf{p}^{(trial)} = \mathbf{p}^{(0)}$ and initialize n_iter.

Step 2: Calculate the reduced Hessian matrix at $\mathbf{p}^{(trial)}$ and the orthogonal matrix $\mathbf{V}$.

Step 3: Calculate the number of estimated parameters np based on the modified E-optimal design criterion and the transformed parameter vector $\mathbf{q}$ is partitioned into $\mathbf{q}_1$ and $\mathbf{q}_2$ according to np.

Step 4: Let $iter = 0$ and $\eta^{(opt)} = Infinity$.

Step 5: Obtain $\mathbf{p}^{(trial)}$ by sampling based on the given range and the distribution of $\mathbf{p}$ and then $\mathbf{q}^{(trial)} = [\mathbf{q}_1^{(trial)T}, \mathbf{q}_2^{(trial)T}]^T$ is equal to $\mathbf{V}^T\mathbf{p}^{(trial)}$.

Step 6: Fix $\mathbf{q}_2$ at $\mathbf{q}_2{}^{(trial)}$, and estimate the selected parameters $\mathbf{q}_1$ by solving the NLP problem (7.9).

Step 7: Calculate the criterion η based on Eq. (7.10). If $\eta^{(opt)} > \eta$, go to Step 8, otherwise go to Step 9.

Step 8: Let $\eta^{(opt)} = \eta$, $\mathbf{q}_2^{(opt)} = \mathbf{q}_2^{(trial)}$, $iter = 0$.

Step 9: Let $iter = iter + 1$.

Step 10: If $iter ==$ n_iter, print the solution $\mathbf{q}$ corresponding to $\eta^{(opt)}$ and stop, otherwise return to Step 5.

7.3 Numerical Results

Only the ascending stage of rocket launching is considered and the total flying time is 66 s. The measurement sampling time is 10 ms and the standard deviations of the measurement error for x, h and v are 1m, 1m and 1m/s respectively. For simplicity, the true values for drag coefficient C_D and lift coefficient C_L are assumed to be 0.5 and 0.1 respectively. Although the aerodynamic parameters are set to be constant here, they can be handled as time dependent. N_D is set to 6, i.e. the drag coefficient is approximated by 6-segment linear function. N_L is set to 1 and $C_L^{(0)} = C_L^{(1)}$ is required. Hence, there are 8 parameters should be estimated. The initial guesses for drag coefficient and lift coefficient are ${C_D}^{(0)} = 0.1$, ${C_D}^{(1)} = 0.6$, ${C_D}^{(2)} = 0.1$, ${C_D}^{(3)} = 0.6$, ${C_D}^{(4)} = 0.1$, ${C_D}^{(5)} = 0.6$, ${C_D}^{(6)} = 0.1$, ${C_L}^{(0)} = 0.12$. The threshold for the modified E-optimal design criterion is set to 3000. The procedure for determining C_D and C_L is terminated if the criterion η has not been updated n_iter (=2000) times from the last update. The lower and upper bound for C_D and C_L is set to 1.0×10^{-4} and 1.0 respectively.

Based on the modified E-optimal design criterion, $np = 5$ is obtained. Namely, there are 5 estimated parameters and 3 parameters are fixed. The results corresponding to the initial guess $\mathbf{p}^{(0)}$ (with fixed $\mathbf{q}_2^{(0)}$) are shown in the Table 7.1 with index 0. Since there are random sampling involved in the algorithm, 10 experiments are performed. The estimated values for C_D and C_L corresponding to the 10 experiments are shown in the Table 7.1 with index from 1 to 10.

The average values for ${C_D}^{(0)} \sim {C_D}^{(6)}$ and $C_L^{(0)}$ in ten runs of experiments are 0.0184218, 0.625591, 0.463646, 0.493976, 0.508575, 0.474668, 0.549004 and 0.393578 respectively. Compared with the true values, the relative errors of the average estimated values for $C_D^{(0)}$, $C_D^{(1)}$ and $C_L^{(0)}$ are over 20%, while others are less than 10%.

There are total 6600 samplings for each state variable. The average errors of the state variables x, h and v corresponding to each group of estimated values are shown in the Table 7.2.

Table 7.1 The estimation results corresponding to the initial guess $\mathbf{p}^{(0)}$ and ten runs of experiments

Index	$C_D^{(0)}$	$C_D^{(1)}$	$C_D^{(2)}$	$C_D^{(3)}$	$C_D^{(4)}$	$C_D^{(5)}$	$C_D^{(6)}$	$C_L^{(0)}$
0	0.215860	0.963712	0.187435	0.664765	0.410800	0.576117	0.317641	0.12
1	0.00973538	0.704978	0.390345	0.550248	0.470303	0.519544	0.440065	0.354477
2	9.99902×10^{-5}	0.606472	0.482575	0.478713	0.518989	0.461943	0.579400	0.500331
3	0.0472689	0.693603	0.395671	0.547289	0.471071	0.517612	0.441830	0.387841
4	9.99904×10^{-5}	0.589686	0.495180	0.463465	0.518278	0.443390	0.587525	0.767355
5	9.99900×10^{-5}	0.577351	0.500931	0.460219	0.514317	0.440913	0.576756	0.848617
6	9.99901×10^{-5}	0.577637	0.507894	0.466159	0.534681	0.458859	0.613621	0.177960
7	0.0165830	0.718111	0.381058	0.557412	0.470089	0.525892	0.438070	0.0953417
8	9.99925×10^{-5}	0.607618	0.484226	0.476169	0.522774	0.459112	0.591079	0.471854
9	9.99929×10^{-5}	0.583925	0.503101	0.467718	0.532258	0.458087	0.609689	0.289643
10	0.0001	0.596529	0.495474	0.472371	0.532993	0.461328	0.612009	0.0423575

Table 7.2 The average errors of the state variables x, h and v corresponding to each group of estimated values

Index	Average error for $x(m)$	Average error for $h(m)$	Average error for $v(m/s)$	$\eta^{(opt)}$
0	0.0309	0.497	0.152	1.179876
1	0.00968	0.155	0.0474	0.045993
2	0.0135	0.144	0.0607	0.041314
3	0.00926	0.148	0.0451	0.046015
4	0.0163	0.165	0.0739	0.039483
5	0.0154	0.155	0.0698	0.039509
6	0.0171	0.171	0.0781	0.043525
7	0.0107	0.172	0.0525	0.046095
8	0.0152	0.158	0.0678	0.040707
9	0.0169	0.170	0.0767	0.039435
10	0.0169	0.172	0.0764	0.040215

From Table 7.2, we can see that any group (from index 1 to 10) of estimated values obtained by the algorithm is much better than those of the results corresponding to the initial guess $\mathbf{p}^{(0)}$ (with fixed $\mathbf{q}_2^{(0)}$). The mean values for "average error for x", "average error for h" and "average error for v" in ten runs of experiments are 0.014094, 0.161 and 0.06484, which are better than the results corresponding to the initial guess $\mathbf{p}^{(0)}$ (0.0309, 0.497, 0.152).

The 9th group of estimated values correspond to the least statistic criterion $\eta^{(opt)}$. Although it is not the best group of estimated values for C_D and C_L, the prediction accuracy of the model based on the aerodynamic parameters from the initial guess $\mathbf{p}^{(0)}$ (with fixed $\mathbf{q}_2^{(0)}$) and the 9th group in Table 7.1 are demonstrated with perturbations on the initial state of x, h and v (the true initial states are 0, 0 and 0 respectively).

Table 7.3 The prediction accuracy of the model based on the aerodynamic parameters from the initial guess $\mathbf{p}^{(0)}$ (with fixed $\mathbf{q}_2^{(0)}$) and the 9th group in Table 7.1

Index	Initial state			Average error with $\mathbf{p}^{(0)}$			Average error with 9th group of estimated values		
	$x(m)$	$h(m)$	$v(m/s)$	$x(m)$	$h(m)$	$v(m/s)$	$x(m)$	$h(m)$	$v(m/s)$
1	−0.529922	1.51898	−0.593886	0.659221	0.491745	0.152231	0.181038	0.170222	0.0772265
2	−0.747719	0.596029	−0.842127	0.121801	0.489039	0.150362	0.031816	0.171807	0.0760482
3	1.2561	1.09157	0.219278	3.82984	1.79644	0.247193	1.18389	0.487766	0.139285
4	−1.77971	1.15713	−0.810567	0.14702	0.489202	0.150398	0.0386173	0.171686	0.0760595
5	−0.255352	1.10633	−0.794001	0.162489	0.489304	0.150428	0.0428183	0.171625	0.0760722
6	0.386493	0.211519	1.74936	0.0233694	0.532853	0.151829	0.00855238	0.167072	0.0757704
7	−0.796123	−0.858599	−0.481118	2.1745	0.69549	0.173995	0.625694	0.20479	0.0915449
8	1.00884	1.84864	0.493863	0.660709	0.510168	0.152752	0.208574	0.168341	0.0771112
9	0.599498	1.15546	1.1738	0.0666171	0.518643	0.151473	0.0228759	0.167337	0.0758297
10	−1.20365	−0.548289	2.32498	0.0111777	0.549556	0.152206	0.00433586	0.167592	0.0757173

The randomly generated initial states (following Gauss distribution $N(0, 1)$) and the corresponding results with the initial guess $\mathbf{p}^{(0)}$ (with fixed $\mathbf{q}_2^{(0)}$) and the 9th group in Table 7.1 are shown in the Table 7.3.

From Table 7.3, we can see that the performance of the aerodynamic parameters from the 9th group in Table 7.1 is still superior to the initial guess $\mathbf{p}^{(0)}$ (with fixed $\mathbf{q}_2^{(0)}$) even if the initial states are perturbed.

7.4 Conclusions

It is difficult to estimate all the aerodynamic parameters of launch vehicles based on the measurements of the distance, height and velocity. The traditional approach is to fix the unestimated variables on the nominal values before estimation, however the selection of the nominal values have significant impact on the prediction accuracy of the model. In this work, the modified E-optimal design criterion and the statistic criterion based approach is used for estimating the aerodynamic parameters. The numerical results show that the prediction accuracy can be remarkably improved even when the initial states are perturbed.

References

1. J.B. Hutchinson, The application of the "method of maximum likelihood" to the estimation of linkage. Genetics **14**(6), 519 (1929)
2. J.V. Zanette, F.A. Almeida, RealSysId: a software tool for real-time aircraft model structure selection and parameter estimation. Aerosp. Sci. Technol. **54**(4), 302–311 (2016)
3. Y. Yang, J. Zhao, J. Liu, K. Wang, G. Wang, Q. Wang, Engineering algorithm of missile parameter identification based on least square method. J. Proj. Rocket. Missiles Guid. **38**(4), 77–80 (2018)
4. W. Tang, Z. Shi, J. Chen, Aircraft flutter modal parameter identification using a numerically robust least-squares estimator in frequency domain. Chin. J. Aeronaut. **21**(6), 550–558 (2008)
5. V. Guibert, J.P. Condomines, M. Brunot, M. Bronz, Piecewise polynomial model identification using constrained least squares for UAS stall. IFAC Pap. **54**(7), 493–498 (2021)
6. G. Chowdhary, R. Jategaonkar, Aerodynamic parameter estimation from flight data applying extended and unscented Kalman filter. Aerosp. Sci. Technol. **14**(2), 106–117 (2009)
7. M. Li, L. Liu, S.M. Veres, Comparison of linear and nonlinear aerodynamic parameter estimation approaches for an unmanned aerial vehicle using unscented Kalman filter. Aircr. Eng. Aerosp. Technol. **20**(3), 39–44 (2011)
8. M. Majeed, I.N. Kar, Aerodynamic parameter estimation using adaptive unscented Kalman filter. Aircr. Eng. Aerosp. Technol. **85**(4), 267–279 (2013)
9. D. Ding, K. He, W. Qian, A Bayesian adaptive unscented Kalman filter for aircraft parameter and noise estimation. J. Sens. 1–11 (2021)
10. W. Zhang, J. Zhu, Online identification of aerodynamics with fast time-varying features using Kalman filter. IET Control Theory: Appl. **15**(2), 272–280 (2021)
11. Z. Xu, Y. Cao, M. Zhao, Parameter identification of tailplane iced aircraft based on maximum likelihood method. Mech. Mater. **192**, 57–62 (2012)
12. R. Kumar, A.K. Ghosh, Estimation of lateral-directional aerodynamic derivatives from flight data using conventional and neural based methods. Aeronaut. J. **118**(1210), 1453–1479 (2014)

13. X. Wang, Application of a RBF neural network in aerodynamic parameter identification of a reentry body. Missile Space Launch Technol. **6**, 5–8 (2002)
14. E.A. Morelli, Real-time aerodynamic parameter estimation without air flow angle measurements. J. Aircr. **49**(4), 1064–1074 (2012)
15. B.T. Burchett, Aerodynamic parameter identification for symmetric projectiles: an improved gradient based method. Aerosp. Sci. Technol. **30**(1), 119–127 (2013)
16. S. Bagherzadeh, M. Sabzeparvar, M. Karrari, Nonlinear aerodynamic model identification using empirical mode decomposition. Aerosp. Sci. Technol. **229**(9), 1588–1605 (2014)
17. H. Tu, L. Liu, Aerodynamic parameter identification of UAV based on output-error method. Mech. Mater. **568–570**, 1012–1015 (2014)
18. L. Dou, M. Du, X. Zhang, Y. Wang, Aerodynamic parameter identification of the RLV reentry process based on the EM-EKF algorithm. J. Tianjin Univ. (Sci. Technol.) **52**(12), 1285–1292 (2019)
19. J. Kang, S. Zhang, C. Hu, Application of aerodynamic parameter online identification in rocket ascent guidance. J. Harbin Eng. Univ. **41**(7), 1052–1058 (2020)
20. B. Yang, Formulization of standard atmospheric parameters. J. Astronaut. **1**, 83–86 (1983)
21. S. Kameswaran, L.T. Biegler, Simultaneous dynamic optimization strategies: recent advances and challenges. Comput.: Chem. Eng. **30**(10–12), 1560–1575 (2006)
22. W. Chen, B. Wang, L.T. Biegler, Parameter estimation with improved model prediction for over-parametrized nonlinear systems. Comput.: Chem. Eng. **157**, 107601 (2022)
23. B. Kim, J.H. Lee, Parameter subset selection and biased estimation for a class of ill-conditioned estimation problems. J. Process Control **81**, 65–75 (2019)
24. K.A.P. McLean, K.B. McAuley, Mathematical modelling of chemical processes-obtaining the best model predictions and parameter estimates using identifiability and estimability procedures. Can. J. Chem. Eng. **90**(2), 351–366 (2012)

Conclusions

Zhengyu Song[1]

In the end, we would like to briefly discuss some future challenges and opportunities.

• **Endo-atmospheric closed-loop guidance**

There have been no well-solved and generalized methods or algorithms for closed-loop guidance (CLG) when flying in the atmosphere, where the tracking guidance control has dominated the applications from past to present. The reasons for this phenomenon are contributed to two classical problems: the constraints for structural load tolerance and the falling area limitations of a rocket's jettisons.

Aerodynamic loads will do harm to the structural integrity, which has been recognized since the beginning of the rocket development. However, after so many years of extensive studies, load relief is still an active topic. Three kinds of methods have been applied. First, the wind biasing trajectory design is to plan the flight path offline that meets the angle of attack constraint, while the tracking guidance control is conducted inflight. However, considering the accuracy and uncertainties of the wind field data, usually a strategy with under compensations is prioritized to avoid over compensations. Second, the load relief control regards the force and torque generated by the wind as interferences, but this treatment will have an impact on flight stability and tracking control accuracy. Thus the abovementioned two methods cannot completely relieve the wind loads. Load relief guidance can regulate the flight path in rea-time with angle-of-attack feedback, and the onboard wind profile is modeled either as a low-order polynomial curve fit or as a look-up table. However, this kind of instantaneous control is difficult to predict the effects of the wind compensations on the orbit entry accuracy, and no closed-loop updating are reflected in the current

[1] Researcher, China Academy of Launch Vehicle Technology; also, Adjunct Professor, College of Control Science and Engineering, Zhejiang University, 310027 Hangzhou, People's Republic of China; zycalt12@sina.com

Z. Song et al. (eds.), *Autonomous Trajectory Planning and Guidance Control for Launch Vehicles*, Springer Series in Astrophysics and Cosmology,
https://doi.org/10.1007/978-981-99-0613-0

and follow-up guidance commands, which means the deviations from the nominal trajectory induced by the load relief could only be corrected after flying out of the atmosphere.

When the falling area of the debris is concerned, the complexity of the planning problem is further increased. The falling process of debris is uncontrolled and difficult to be represented by a dynamic or parameterized model, so the falling point cannot be estimated responsively and the constraint cannot be incorporated into the CLG's feedback mechanism by parameter adjustments, resulting in the violation of the landing area constraint thereafter. However, a large number of launch activities have shown that the falling point is mainly affected by the initial velocity and position of the separated body, and the impacts of the aerodynamics can be treated as errors. Thus a new approach may be feasible if a model with the statistical characteristics of the falling point could be established to describe the maximum error.

• **Exo-atmospheric autonomous guidance**

CLG methods outside the atmosphere have also been widely discussed, which are all based on the premise that the target orbit is known, that is, the guidance methods do not care about whether the target orbit is beyond a launcher's performance. Those seemly not so "intuitive" terminal constraints of the orbital elements are transformed into the velocity and position constraints of the current optimal entry point, representing a classical two-point boundary value problem. With no aerodynamic disturbances and the relaxation of debris falling area constraints, an analytical exo-atmospheric guidance law is possible.

If the prescribed target orbit is beyond a launcher's performance due to the non-fatal faults such as thrust dropping, the terminal boundary conditions become unknown and need to be optimized simultaneously with the flight path. It is difficult to ensure a convergent and optimal solution even if this problem is solved offline. At this time, finding a solution is more important than finding an optimal one, so the strategies to the typical failures could be pre-set and uploaded to the onboard computer before liftoff, and the responses inflight are made based on the prescribed targets. However, the autonomous guidance methods (AGMs) are more adaptable to the thrust variations than in-advance offline planning, but at present only a few technical breakthroughs have been made, which exhibit a culture that doing something is better than nothing to do to wait helplessly as the failure occurs.

There is an assumption for the discussions of the AGMs in this book, i.e., all remaining propellant could be utilized, but this is not the situation if the thrust dropping is caused by the propellant leakage, ablation, or reduction of the pump efficiency. The latter two fault modes will lead to engine explosion, so timely shutdown is a preferable measure. The leakage would results in the variations of the mass-flow-rate and the remaining propellant not be fully usable. Moreover, the engine specific impulse, mass-flow-rate, and mixing ratio of the propellant would all be affected under failure conditions and have different influences on the AGMs. Thus, a clear understanding of the failure modes of the engines is definitely required to conduct mission reconstructions.

• **End-to-end trajectory optimization**

At present, the sequential optimization, not end-to-end (E2E) planning, is applied for the ascent flight or recovery. Strictly speaking, it is the perfect state of E2E if taking the payloads as the final stage of the rocket for a joint optimization. The sequential optimization needs to consider the handover conditions between each flight phase, which are rarely regulated adaptively even a CLG is adopted within this phase. This leads to the suboptimal solution from the point of view of the entire flight profile.

The same is true of the rocket recovery. The powered landing guidance methods are mostly studied, whose premise is that the initial condition of the powered descent is within the feasible region, or in other words, the handover conditions between the aerodynamic decelerating phase and the powered descent phase should be well controlled. However, the control means during the aerodynamic decelerating are relatively weak (generally relying on RCS and grid wings) and the handover conditions are hard to be accurately controlled. Thus, another requirement is arisen that the feasible region of the starting point of back-propelling should be greatly extended to tolerate large deviations, which demands a deep throttling capability to maintain a low thrust-to-weight ratio.

E2E optimization only considers the initial and terminal conditions, while the handover conditions can be regulated autonomously as needed. This is certainly an appealing vision, but online planning is time-consuming, computationally intensive, and convergent not ensured. Considering that the powered vertical recovery is at the expense of the payload carrying capacity, the driving force for E2E optimization from takeoff, stage separation, to landing, is still urgent.

Numerical computing may be the best choice for E2E optimization when an analytical solution is impossible. With the performance improving of solvers, hardware and software products, the numerical computing can also acquire satisfied results if a good initial guess is provided, although how to obtain this initial value is not only a scientific problem but also an art. The analytical solution of a simplified problem by relaxing some constraints is the uppermost initial guess.

Despite growing enthusiasm for the artificial intelligent (AI) technologies, they have not been able to play impressive roles in spaceflight except for fault diagnosis, pattern recognition and other classification applications. After sufficient training, AI presents "reflex" features to the outer world, which are difficult to be interpreted, to explain the reasoning mechanism, or to predict the impact on follow-up operations. These make the AI technologies face a trust issue when applied in influential projects. However, AI technologies can be regarded as a dependable way to generate an initial guess. Even if the solution provided by the AI technologies is a feasible one, it seems more credible if this solution is re-processed as the initial value.

The studies introduced in this book lay the foundation to promote autonomy for space vehicles. For any innovative solutions, if they do not show obvious advantages over the existing approaches, or the need to them has not been clearly established, or the assessments needed to match their risk tolerance have not been conducted, there

is a considerable and understandable reservation in committing oneself to these complicated schemes where millions of dollar vehicles are concerned. Thus, a rideshare demonstration flight is welcome to assess the appropriateness and readiness of these studies and build confidence.