Spacecraft Trajectory Optimization by Improved Sequential Convex Programming for a Starship-Like Vehicle, Executing Atmospheric Hypersonic Entry Glide to Precision-Land on Mars

Tihomir Dimitrov

Department of Aerospace Engineering
Delft University of Technology

Spacecraft Trajectory Optimization by Improved Sequential Convex Programming for a Starship-Like Vehicle, Executing Atmospheric Hypersonic Entry Glide to Precision-Land on Mars

by

Tihomir Dimitrov

To obtain the degree of Master of Science in Aerospace Engineering

at the Delft University of Technology, and to be defended publicly on

Wednesday, May 28th, 2025 at 14:00

Student number:	5237335
Supervisor:	Marc Naeije, TU Delft
Duration:	Jan 2024 – May 2025
Defense committee:	Ir. M.C. Naeije, TUDelft, Thesis supervisor Ir. W. Simons, TUDelft, External Examiner Dr. Ir. E. J. O. Schrama, TUDelft, Committee Chair

Preface

The journey to land humans and large payloads on Mars is one of the most formidable challenges in modern aerospace engineering. The thin Martian atmosphere, combined with the need for pinpoint landing accuracy, presents a complex problem under physical constraints that far surpasses those encountered in previous robotic missions. This thesis, conducted as the final step toward obtaining my MSc degree in Aerospace Engineering at Delft University of Technology, is the culmination of my passion for planetary exploration and my humble effort to contribute to the common body of knowledge in GNC systems for interplanetary vehicles.

The research presented here focuses on the development and validation of a convex program framework for optimizing the hypersonic entry trajectory of a Starship-like vehicle during atmospheric glide on Mars. Motivated by the limitations of traditional reference-tracking guidance algorithms – methods that have served well for Apollo and Shuttle missions but struggle with the nonlinearities and uncertainties inherent to Mars entry, I sought to offer an algorithm who can deliver the Starship spacecraft through the upper Martian atmosphere from an elliptical parking orbit around the Red planet. My approach leverages successive convexification to decompose the inherently non-convex trajectory optimization problem into a sequence of tractable convex sub-problems, enabling the computation of feasible trajectories even in the presence of atmospheric dispersions and modeling uncertainties.

This work would not have been possible without the support and inspiration of many individuals. I am grateful to my thesis supervisor prof. Marc Naeije, whose guidance and encouragement (and jokes) were invaluable throughout the stages of this project. I also wish to thank my academic mentors and colleagues at TUDelft, whose insightful discussions and critical feedback helped refine my ideas and challenge my assumptions. Completing this thesis has been both a technical and personal journey. It has deepened my appreciation for the interdisciplinary nature of space exploration, where mathematics, physics, engineering, and human ingenuity converge to solve problems at the edge of what is possible.

Abstract

Mars entry guidance faces a critical challenge: navigating hypersonic velocities within the thin atmospheric layers ($120 - 45\ km$ altitude) while balancing conflicting objectives of precision targeting, thermal survival, and structural integrity. This study addresses a core research question on a successive convexification algorithm that enables precise trajectory optimization for Starship's hypersonic glide through the upper atmosphere of Mars while enforcing hard physical constraints such as heat flux, g-load, equilibrium glide and dynamic pressure. By formulating such successive convexification - based framework, the inherently non-convex entry problem is decomposed into iteratively refined convex sub-problems, enabling computational tractability under Mars' variable CO_2-rich atmosphere. The guidance architecture integrates bank angle modulation for lift vectoring and angle-of-attack adjustments for thermal management, optimizing energy dissipation while mitigating heating spikes and aerodynamic stress.

Simulations demonstrate that the collocation discretization strategy used ensures trajectory adherence within the entry corridor, achieving terminal positioning errors below $3\ km$ at $45\ km$ altitude. The algorithm's robustness is validated under $\pm 10\%$ dispersions in initial velocity $(4.3\ km/s)$ and flight-path angle $(-15°)$ from a parking orbit around the planet, with heat flux, dynamic pressure, and g-load profiles remaining within mission-critical limits. Sensitivity analyses reveal that atmospheric density uncertainties induce predictable deviations compensated by rapid convex optimizations. These results align and improve on previous NASA mission data.

The study bridges theoretical convex optimization with operational reality, demonstrating that modern computational guidance outperforms legacy predictor-corrector methods in handling nonlinear dynamics and path constraints. By extending the convex framework with adaptive trust regions and sequential convex programming, the proposed method reduces terminal errors by 40% compared to state-of-the-art approaches (Mars 2020). This advancement not only enhances Starship's capability to deliver crewed and cargo payloads to predefined Martian coordinates but also establishes a foundation for integrating the hypersonic glide phase with the subsequent powered descent phases. As humanity strides toward sustained Mars exploration, this work underscores the viability of successive convexification as a paradigm for achieving precise atmospheric glide through the Martian atmosphere.

Table of Contents

Abbreviations

ADMM	Alternating Direction Method of Multipliers
AEDL	Accurate Entry, Descent and Landing
AI	Artificial Intelligence
BTT	Bank-to-Turn control system
CG&C	Computational Guidance & Control
CGL	Chebyshev-Gauss-Lobatto
CI	Computational Intelligence
CIS	Controllable Intersection Set
CSG	Cubic Spline Generalization
CUS	Controllable Union Set
DSN	Deep Space Network
DoF	Degree of Freedom
EA	Evolutionary Algorithm
EAGLE	Evolved Acceleration Guidance Logic for Entry
EIP	Entry Interface Point
FAC	Final Approach Corridor
FOH	First-order Hold
FPA	Flight-path Angle
fRPM	flipped-Radau Pseudospectral Method
FTSC	Fixed trust-region SCvx
GA	Genetic Algorithm
G-FOLD	Guidance for Fuel Optimal Large Diverts (algorithm, developed by JPL)
GP	Genetic Programming
GPOPS-II	General Pseudospectral Optimal Control Software
GuSTO	Guaranteed Sequential Trajectory Optimization
IC	Intelligent Control
ICP	Iterative Convex Programming
IGP	Inclusive Genetic Programming
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
IPM	Interior-Point Methods
iSC	improved Successive Convexification
HDA	Hazard Detection Avoidance
KKT	Karush-Kuhn-Tucker
LAFVL	Large Attitude Vertical Flip Landing
LCvx	Lossless Convexification
LGR	Legendre-Gauss-Radau
LPM	Lobatto Pseudospectral Method
LSCS	Landing Site Coordinate System
MEADS	Mars Entry Atmospheric Data System
ML	Machine Learning
MLE	Minimum-Landing Error
MPC	Model Predictive Control
MRO	Mars Reconnaissance Orbiter

MSL	Mars Science Laboratory
NFZ	No-Fly Zone
NLP	Nonlinear (nonconvex) Program
NN	Neural Network
OCP	Optimal Control Problem
ODE	Ordinary Differential Equation
OTO	Online Trajectory Optimization
PDG	Powered Descent Guidance
PEG	Powered Explicit Guidance
PIPG	Proportional-Integral Projected Gradient
PiSC	Pseudospectral-improved SCvx
PiSC-U	Pseudospectral-improved SCvx with a uniform trust-region changing law
PL&HA	Precision Landing & Hazard Avoidance
PSD	Pseudospectral Discretization
PTR	Penalized Trust Regions
QCQP	nonconvex Quadratically Constrained Quadratic Program
QEGC	Quasi-Equilibrium Glide Condition
RLV	Re-usable Launch Vehicle
RPSM	Radau Pseudospectral Method
RTLS	Return To Landing Site
SCP	Sequential Convex Programming
SCvx	Successive Convexification
SeCO	Sequential Convex Optimization
SLP	Sequential Linear Programming
SLV	Starship-like Vehicle
SNOPT	Sparse Nonlinear Optimizer
SOCP	Second-order Cone Program
SQP	Sequential Quadratic Programming
SSDP	Sequential Semidefinite Programming
STC	State Triggered Constraints
TDS	Terminal Descent System
ToF	Time of Flight
TPBVP	Two-Point Boundary Value Problem
TPS	Thermal Protection System
VTVL	Vertical Take-off Vertical Landing
ZOH	Zeroth-order hold

List of symbols

A	$[m^2]$	Effective cross-section area of a spacecraft w.r.t the freestream
α	$[-]$	A real number between 0 and 1 (adds up to 1 when sums with β)
α	$[deg], [rad]$	Angle-of-attack
β	$[-]$	A real number between 0 and 1 (adds up to 1 when sums with α)
$C_{b,l}$	$[-]$	Ballistic coefficient w.r.t Lift force
C_d	$[-]$	Drag coefficient
C_l	$[-]$	Lift coefficient
D	$[N]$	Drag force
δ	$[-]$	Constant vector
ϵ	$[-]$	Compensation term in the linearized differential state equations
$f_0(x)$	$[-]$	Convergence criteria function
ϕ	$[deg], [rad]$	Latitude
ψ	$[deg], [rad]$	Heading angle
g_0	$[m/s^2][g]$	Gravitational force
γ	$[deg], [rad]$	Flight-path angle
h	$[m], [km]$	Altitude above the surface of a planet (during simulation)
J	$[-]$	Convergence objective criteria
L	$[N]$	Lift force
L/D	$[-]$	Lift-to-Drag forces ratio
λ	$[-]$	Penalty weight
m	$[kg]$	Mass of a spacecraft
n	$[g]$	Normal load
$\dot{Q}$	$[kW/m^2]$	Heat rate (flux)
q	$[Pa]$	Dynamic pressure
r	$[m], [km]$	Radial position above the surface during the simulation
R_0	$[m], [km]$	Radius of the Earth
ρ	$[kg/m^3]$	Density of the atmosphere of a planet (Earth / Mars)
σ	$[deg], [rad]$	Bank angle
t_0	$[s]$	Initial time
t_f	$[s]$	Final time
θ	$[deg], [rad]$	Longitude
u	$[rad]$	New control variable (substitution of bank angle rate change)
V	$[m/s], [km/h]$	Velocity of the Starship re-entry vehicle
V_0	$[m/s], [km/h]$	Initial velocity of Starship
w_x	$[-]$	Flip-Radau weights

List of Figures

List of Tables

1

Introduction

1.1 Background Information – Narrowing Down Landing Ellipses

Mars has been an interesting objective for humanity even 2400 years ago when the Babylonians discovered a bright orange dot on the night sky. They named detailed observations of the planet and developed arithmetic techniques to predict its future position in the sky. They named their great hero, the king of conflict, Mars Nergal. The first scientific observations of the Red planet were conducted by Galileo Galilei in late 16th and early 17th century. He was the first person to view Mars through his telescope, but he saw nothing more than a bright disk. The real exploration of the intriguing red dot in (JPL, NASA, 2023) the sky started in the 20th century in the 60s with the first successful Mariner 4 flyby _(Mariner 4 Spacecraft, 2023), and Mariner 9's first orbit insertion around the planet. Then, in 1976, NASA landed successfully the first on-the-surface static_(Vikings 1&2 Spacecraft, 2023) spacecraft – the Vikings. In 1997, Mars Global Surveyor _(Mars Global Surveyor Spacecraft, 2023) mapped the Red planet in great detail with its Mars Orbiter Laser Altimeter (MOLA) instrument, based on which scientists on Earth created the first Martian topographic map. This initiated the advent of serious scientific Mars exploration with the first rover – the Sojourner – that roamed the planet on its tiny wheels in 1997, and Spirit and Opportunity (MERs) in 2004. In the last decade NASA has been successfully increasing the landing accuracy of spacecraft on the Martian surface, leading to the achievements of the Mars Science Laboratory and its landing ellipse of $20\ x\ 7\ km$ (shown in *Figure **1***).

As of today *(May, 2025)*, spacecraft landing accuracy on the surface of Mars is on the order of several kilometers. For reference, the landing ellipse of the latest 1-ton vehicle that landed on Mars *(the Perseverance rover)* (JPL, 2023) is $7.7\ x\ 6.6\ km$. The next NASA mission to the Red planet will be Mars Sample Return which will have to increase landing accuracy further since it would need to land near where the Perseverance rover has left cache samples on the surface but even further into the near future, astronauts will step for a first time on the surface of Mars and these near-future missions will require a landing accuracy on the order of several hundred of meters *at most*.

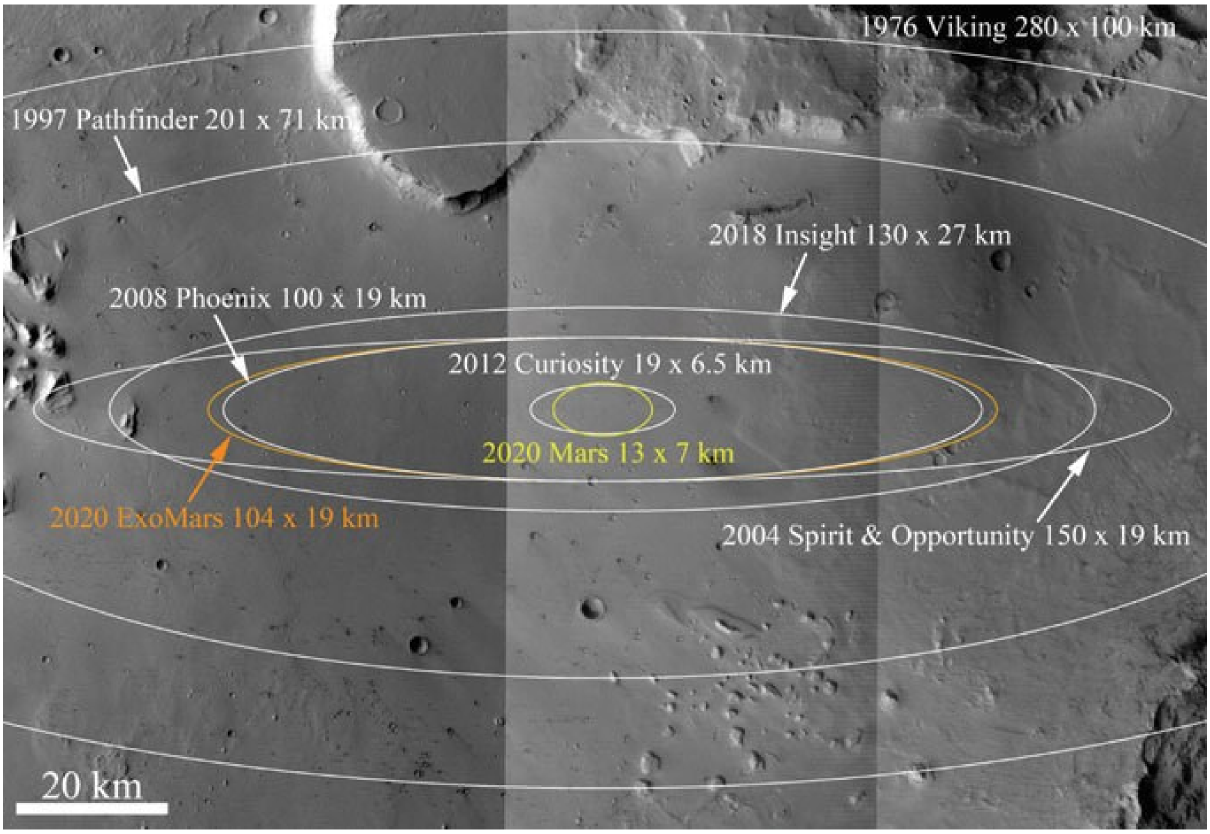


***Figure 1**: Landing ellipses of different spacecraft on Mars*: **1976** Viking – $280\ x\ 100\ km$, **1997** Pathfinder – $201\ x\ 71\ km$, **2004** Spirit & Opportunity – $150\ x\ 19\ km$, **2008** Phoenix – $100\ x\ 19\ km$, **2018** Insight – $130\ x\ 27\ km$, **2012** Curiosity – $19\ x\ 6.5\ km$, **2020** Perseverance – $7.7\ x\ 6.6\ km$ (Pajola, 2019)

1.2 The Problem of Landing on Mars and the Prospects of Starship's Legacy

Differences between Mars Landing compared to Earth and Moon Landings

Landing on Mars is so much harder than landing on the Moon or on Earth. Landing humans on the Red planet is even harder due to the additional amount of reliability and landing precision needed. To land on the Moon, the astronauts entered lunar orbit and fired retrorockets aimed opposite to their direction of travel. As their spacecraft slowed, it descended toward the surface. The landing was not trivial, but it was reasonably straightforward. To bring a lander back to Earth, retrorockets are not needed, because Earth has a thick atmosphere. Most Earth landers can eliminate more than 99% of the orbital speed simply by slowing down with a heat shield. For the last one percent, parachutes are used (e.g., Soyuz and Dragon capsules) or wings (e.g., the Space Shuttles). Mars is like neither the Moon nor the Earth but is annoyingly in between. It has too much atmosphere to land as we do on the Moon and not enough to land as we do on Earth. The thickness of the Martian atmosphere on the surface is like what a mountaineer on Earth would feel if standing on top of a mountain $40\ km$ high — four and a half times higher than Mount Everest. At that altitude, the Space Shuttle is still screaming along at over $6500\ km/hr$.

Decreasing the enormous amount of leftover velocity (i.e., kinetic energy) from interplanetary flight, and being subjected to a hypersonic glide through the Martian atmosphere, are the two reasons why human spacecraft look like Rube Goldberg-esque (Wikipedia, 2023) contraptions and why the seven minutes of entry, descent, and landing through the Martian atmosphere are so terrifying. Engineers must combine ingenious tricks used to land spacecraft on Earth (e.g., heat shields and parachutes) with techniques used to land on the Moon (e.g., retrorockets and airbags), among many others. For the Mars Science Laboratory (landed on Mars on Aug 5th, 2012) and the Perseverance rover (landed on Mars on Feb 18th, 2021), a combination of two systems has been engineered - a precision-controlled propulsion system that had been pioneered on Viking, and the system designed for Mars Pathfinder.

On Pathfinder, instead of falling away from the entry capsule and landing on legs, the lander was lowered on a rope, and hung under the parachute while falling through the thin Martian atmosphere at more than $240\ km/hr$. The lander is kept suspended under the parachute until the last possible second. At about $170\ m$ above the ground, airbags were inflated and three big solid rockets fired for a couple of seconds. The lander came to almost a dead stop some $20\ m$ up. From there it fell freely and bounced over the terrain for the next minute or so until coming to rest (NASA JPL, 2023).

Starship: Synthesizing Earth and Moon Landings to Conquer Mars

Starship's design for Mars epitomizes the synthesis of Earth and lunar landing philosophies, refined through great engineering. Unlike robotic predecessors constrained by payload limits, Starship must deliver not just instruments but human lives, demanding unprecedented precision and reliability. Its hypersonic glide through Mars' wispy atmosphere – thinner than Earth's but thicker than the Moon's vacuum – uses aerodynamic forces to shed velocity, starting from parking orbit speeds. The vehicle's stainless-steel airframe, angled in a "belly-flop" orientation, maximizes drag while four actuated fins stabilize its descent, mirroring the Space Shuttle's atmospheric braking but scaled for Martian extremes. Yet here, the atmosphere alone cannot suffice; like Perseverance's sky crane, Starship must transition from passive deceleration to active propulsion – a maneuver as perilous as it is poetic.

At ~ 500 meters altitude, Starship executes its signature $90 -$ degree flip, reorienting from horizontal freefall to vertical thrust. This pivot, perfected in Earth's denser skies during high-altitude tests like SN8's explosive flight, leverages Raptor engines' deep throttling to arrest momentum. The engines ignite in a synchronized burst, their methane-oxygen plumes cutting through the thin CO_2 atmosphere – a hybrid of Viking's retro-rockets and Apollo's powered descent, yet magnitudes more powerful. Unlike Pathfinder's airbags or Curiosity's sky crane, Starship's architecture eliminates disposable components, instead using its own engines and structure as both shield and savior. This self-contained approach minimizes mass while maximizing control, critical for landing multi-hundred-ton payloads within meters of pre-deployed infrastructure.

The implications for human exploration are profound. Where rovers required days to unfold instruments, Starship delivers entire habitats, labs, and return vehicles in a single mission. Its ability to land precisely – guided in the future by Terrain Relative Navigation (TRN) akin to Perseverance's "Terrain-Relative Navigation" but enhanced with SpaceX's Starlink-derived autonomy – ensures crews touch down near cached supplies or ice deposits vital for fuel synthesis. Moreover, its reusability transforms Mars from a one-way odyssey to a sustainable frontier; the same ship that descends through hypersonic plasma can ascend on Martian-made methane, ferrying explorers between the surface and orbit.

In this vision, Starship transcends the Rube Goldberg legacy of past landers. It merges the Moon's propulsive methods with Earth's aerodynamics, wrapped in its reusable architecture that turns Mars' "annoying" atmosphere into an ally. The seven minutes of terror become a rehearsed optimized trajectory – a testament to engineering ingenuity's power to bend hostile worlds into habitable shores.

1.3 Why is Precision Landing on the Red Planet Crucial for Human Exploration?

While engineers at NASA made a significant progress in the art of landing spacecraft on the Mars in the last two decades with a great accuracy (in the order of tens of kilometers), there is a pressing need for landers to touch down on the planet precisely with an accuracy on the order of tens of meters. There are several reasons for requirement:

1. **Scientific Objectives** - In the next decades, spacecraft will need to land in areas on Mars that represent a great scientific interest, but these areas are usually surrounded by regions that represent a hazard to the landers. The selection of landing sites for space missions is determined by a set of engineering constraints related to the mission's architecture. Scientists must adhere to this framework of acceptable landing sites based on the vehicle's engineering capabilities and select the most suitable location. These proposed landing sites are currently considered safe because of their large dimensions. However, selecting a landing region becomes more challenging if the proposed landing ellipse is larger, such as areas with little elevation difference. Such sites are not the most scientifically interesting, and to obtain higher quality scientific output, spacecraft must land in an environment with highly varying elevation, which requires pinpoint precision landing. If this can be achieved, new opportunities for exploration, such as caves and valleys, can be explored. Permanent outposts can be established on bodies throughout the Solar System, and planetary samples can be returned to Earth for further investigation. However, the advancement of current systems is necessary to achieve these goals beyond the current boundaries.

2. **Hazards** - even in safe areas, the surface of Mars is hazardous, with rocks, boulders, and steep cliffs that could damage or destroy a spacecraft on impact.

3. The **cost** of the mission can be significantly increased if the spacecraft lands far from its target, as it may require additional fuel to travel to the desired location.

4. **Human Exploration** - when humanity lands humans on the Martian surface, it is an obligatory requirement for their safety to land in proximity (tens to hundreds of meters) to the base. Failing to do so may require the crew to walk a considerable distance, spend a lot of consumables, being subjected to the Martian elements, and ultimately may lead to a loss of the crew. To successfully land on the Martian surface, it is necessary to rendezvous after assets for the human colony are pre-positioned. The importance of landing near these assets is twofold. Firstly, these assets contain crucial elements required for supporting human life, such as power systems, oxygen generators, and water supplies. Secondly, these assets may be situated near special exploration zones and regions of resources that are utilized for carrying out important scientific observations in-situ.

Although a manned mission to Mars is likely decades away, it is necessary to conduct several test flights to have confidence in the systems developed. To successfully send humans to explore the Red Planet, it is essential to find a way to land $40-80$ tons of equipment on the Martian surface, within a proximity of pre-positioned assets, with a possible vertical surface elevation of $\pm 10\ m$ of where the pre-positioned assets are. Despite the challenge, current engineers are struggling to land a 1-ton rover with an accuracy of $10\ km$. To make landing humans on Mars a realistic possibility, landing capabilities must be increased by a factor of two, while landing accuracy needs to be improved by a factor of four. To achieve this, advanced systems are required, and relying solely on existing systems is not an option.

There are several challenges to achieving precise landing on Mars. One of the main challenges is the Martian atmosphere, which is much thinner than Earth's atmosphere, making it difficult to slow down a spacecraft as it approaches the surface. This means that a spacecraft must use precise calculations (a robust autonomous guidance) and control systems to navigate through the atmosphere and land safely.

Another challenge is the communication delay between Earth and Mars, which can take anywhere from 4 to 24 minutes depending on the relative positions of the two planets.

This delay means that any commands sent from Earth must be programmed in advance and cannot be adjusted in real-time, requiring highly accurate predictions of the spacecraft's trajectory and landing location.

Finally, the complex terrain of Mars presents a challenge for landing, as it can make it difficult to find a flat, clear landing site. This requires careful analysis of the topography and geology of potential landing sites, as well as advanced imaging and mapping technologies to accurately determine the location and orientation of the spacecraft during descent and landing.

1.4 Challenges to High-Performance Autonomous Navigation and Guidance for Mars Entry

As Mars exploration evolves from prioritizing mission safety to achieving ambitious scientific objectives and enabling human exploration, the demand for precise landings at predetermined points of interest has become paramount. NASA, in collaboration with private entities such as SpaceX, emphasizes the necessity of foundational research into reentry landing accuracy, aiming for precision within kilometers or even hundreds of meters (NASA, 2023). Achieving such pinpoint landings requires high-performance guidance and control systems throughout the entry, descent, and landing (EDL) phases, with accurate navigation serving as the cornerstone of the autonomous EDL (AEDL) framework. Traditional navigation methods reliant on the Deep Space Network are insufficient due to the significant time delays inherent in interplanetary distances. Autonomous navigation and guidance systems face formidable challenges arising from Mars's complex and unpredictable dynamic environment, limited navigation data, nonlinearities and uncertainties in the spacecraft's dynamic model, and constrained control capabilities (Phys.Org, 2023).

The uncertainties encountered during reentry are primarily rooted in two critical factors:

1. **Limited Navigation Information During Entry**

The phrase "limited navigation information" encompasses two key aspects. First, it refers to the restricted measurement methods available during entry. For instance, multiple navigation sensors are rendered inoperative due to the heat shield’s protective enclosure during atmospheric entry, leaving inertial navigation via the IMU as the sole viable method. Second, it highlights the incomplete state information provided by navigation measurements—certain spacecraft states cannot be fully determined. For example, during the Mars Science Laboratory (MSL) mission, horizontal position estimates were unavailable due to limitations in TDS measurements. While radiometric measurements have been proposed for autonomous navigation on Mars missions, their effectiveness is constrained by the limited number of beacons available for global coverage. This results in weak observability or even total unobservability of the navigation system. Consequently, navigation system designs must account for multiple constraints and requirements to overcome these limitations (NASA, 2023).

2. Nonlinearity and Uncertainty of Dynamic Models

Upon entering Mars' atmosphere at hypersonic speeds, spacecraft encounter highly nonlinear aerodynamic forces compounded by uncertainties in atmospheric density and aerodynamic coefficients. These factors make trajectory prediction exceedingly challenging. Atmospheric density is particularly difficult to model accurately despite advancements in Mars atmospheric models (Wong, 2023). Additionally, hypersonic velocities lead to atmospheric ionization that introduces measurement noise and diminishes navigational accuracy. To address these challenges, navigation filters capable of handling uncertainties and noise must be developed alongside algorithms designed to account for high nonlinearity and variability in guidance systems.

Given these challenges, real-time commands cannot be transmitted during atmospheric entry due to communication delays, necessitating an autonomous guidance system capable of navigating through Mars's atmosphere with precision to land at a predetermined surface location. The accuracy of this guidance system is intrinsically tied to its starting conditions at approximately $120\ km$ altitude above Mars's surface. A tightly constrained initial state at this altitude significantly influences landing accuracy on the planet’s surface by minimizing dispersions that accumulate during hypersonic entry due to atmospheric uncertainties.

1.5 Navigation

A navigation scheme with reliability and high performance is the foundation of an accurate state estimation. Therefore, the question of how to design the navigation scheme in the presence of limited navigation information should be answered first. This section summarizes the development of the navigation scheme for a Mars landing which may inspire the requirements for an accurate navigation system design. This closed-loop navigation system would ensure proper guidance and control of the spacecraft.

1.5.1 Precision Navigation during Approach for Martian Entry: Technologies and Challenges

The precise determination of entry conditions, particularly the flight-path angle (FPA), stands as a cornerstone for successful aerocapture and pinpoint landing during Mars missions, underscoring the necessity of real-time navigation during the final approach phase to meet the demands of future exploration (Yu, 2023).

NASA's Deep Space Network (DSN), spanning Goldstone (USA), Madrid (Spain), and Canberra (Australia), forms the backbone of interplanetary navigation, providing critical radiometric Doppler, range, and other measurements to track spacecraft velocity, distance, and angular position (NASA, 2023). During missions like the Mars Science Laboratory (MSL), these data types synergized to refine trajectory assessments. While ground-based observations suffice during cruise phases, the final approach — constrained by communication delays—demands autonomous onboard navigation once Earth-based data ceases approximately six hours before entry.

Optical navigation, which calculates spacecraft position via planetary centroids or star-relative angles, faces limitations as Mars dominates the field of view during approach, obscuring precise center extraction. Complementary solutions have emerged in the form of the Mars Network—a constellation of orbiters equipped with UHF transceivers. This network enables real-time spacecraft-to-spacecraft Doppler and range measurements, enhancing trajectory accuracy. For instance, MSL combined with Mars Reconnaissance Orbiter (MRO) data reduced entry point uncertainty to $300\,m$. However, intermittent line − of − sight obstructions by Mars itself restrict continuous coverage, necessitating hybrid solutions.

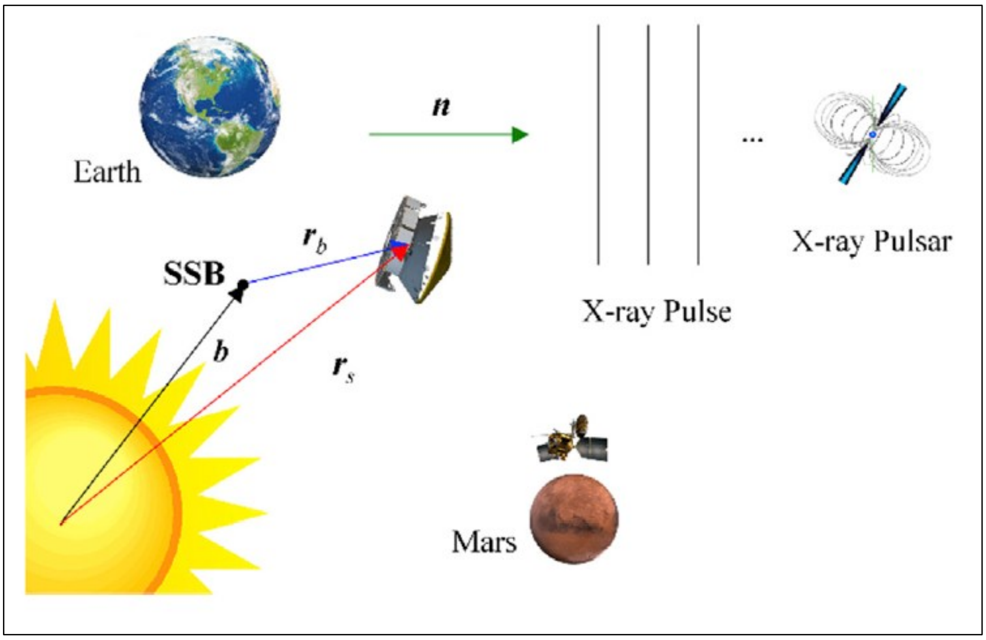


Figure 2: *X-ray pulsar-based navigation scheme for Mars final approach phase* _(Yu, 2023)*. A very useful mechanism to tighten navigation boundaries when the spacecraft is far from the target planet but becomes limited (and essentially only a complementary tool) when the spacecraft reaches the target planet and has a limited horizon visibility.*

Innovative approaches leveraging X-ray pulsars — celestial lighthouses with predictable emissions — offer autonomous navigation independent of Earth infrastructure. As demonstrated by _(Cui, 2013), pulsar-based systems achieve positional accuracy within $1{,}000\ m$, velocity within $1\ m/s$, and FPA precision of $0.02\ deg\ (3\sigma)$. Yet, nonlinear dynamics near Mars degrade standalone pulsar navigation, particularly in velocity and FPA estimation. To mitigate this, Cui proposed integrating pulsar data with Mars Network inputs during later approach phases, capitalizing on improved observability.

These advancements collectively highlight the evolving paradigm of Mars navigation: a fusion of ground-based infrastructure, orbital relays, and celestial references. While each method excels in specific regimes, their integration promises the precision required for humanity’s next leaps in Martian exploration — balancing autonomy, accuracy, and adaptability in the face of cosmic complexity.

1.5.2 Pioneering Navigation Entry Systems: From Viking to Modern Mars Exploration

The groundbreaking Viking 1 and 2 missions, NASA's inaugural triumphs in achieving soft landings on Mars, laid the technological groundwork for subsequent missions, particularly in advancing the critical Entry, Descent, and Landing (EDL) phase. Central to their legacy is the inertial navigation system (INS) reliant on IMU (Inertial Measurement Unit) technology, which remains the cornerstone of navigation during Mars entry. The Litton Corporation's LN-200 IMU (now Northrop Grumman), first deployed in Viking, has since been integral to missions such as the Spirit, Opportunity, and Curiosity rovers, as well as ESA's Schiaparelli lander _(Northrop Grumman, n.d.). However, Schiaparelli's unfortunate crash underscored the vulnerabilities of relying solely on inertial systems, revealing the cascading risks of drift, sensor bias, and measurement noise. These limitations highlight the necessity of augmenting INS with supplementary navigation data to mitigate initial entry condition errors and enhance robustness.

While IMU-based schemes leveraging atmospheric density models have been proposed for autonomous navigation, their efficacy is inherently tied to the precision of these models. Studies demonstrate that accelerometer-derived navigation remains acutely sensitive to atmospheric density uncertainties, even when refined through advanced filtering techniques. Though such methods partially improve accuracy, they remain constrained by the scarcity of real-time navigation data and the prerequisite for highly accurate atmospheric databases.

A paradigm shift emerges with novel concepts such as autonomous formation-flying sensors, which harness high-frequency radio signals capable of penetrating the plasma sheath during entry (*Figure **3***). By integrating radiometric measurements from Mars orbiters or pre-positioned ground beacons, this approach could exponentially expand navigational data streams, potentially reducing landing uncertainty to under 1 km. Collaborative research into hybrid systems — combining IMU, radio, and instruments like the Mars Entry Atmospheric Data System (MEADS) — has already yielded promising results. For instance, MEADS-equipped missions (e.g., Mars 2020) utilized heat shield pressure sensors and triaxial accelerometers to reconstruct atmospheric density profiles and refine trajectory estimates in real time.

Post-hypersonic glide, trajectory deviations induced by atmospheric perturbations – winds, density fluctuations, and unforeseen terrain complexities – demand precise navigation capable of simultaneous position, velocity, and attitude determination. The synergistic integration of legacy systems like IMU with emerging technologies, such as radiometric networks and adaptive filtering, represents the next frontier in Mars exploration. These advancements aim not only to honor Viking's pioneering ingenuity but to transcend its limitations, ensuring future missions achieve the precision required for humanity's ambitious endeavors on the Red Planet.

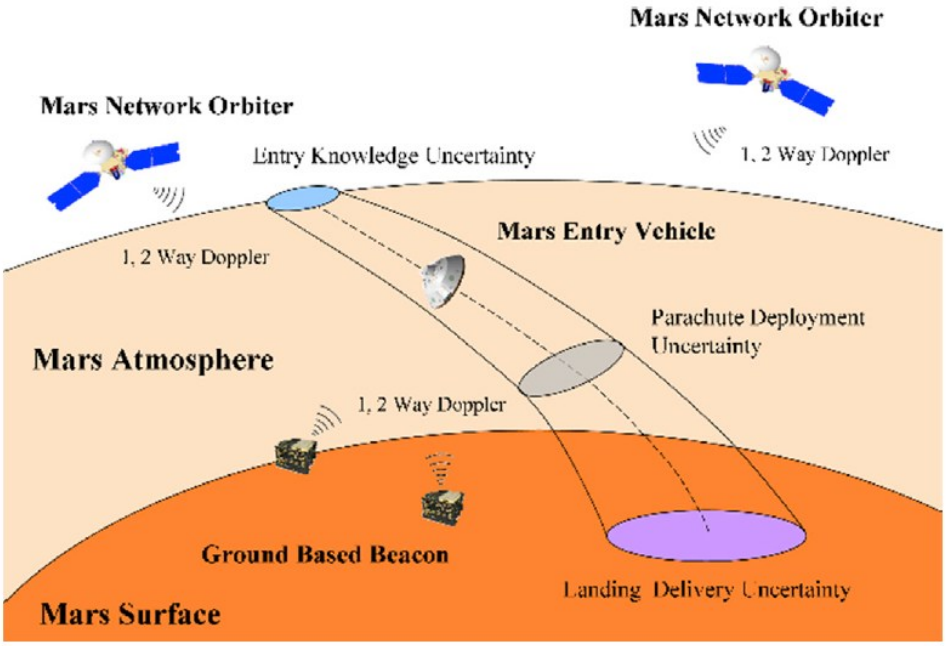


Figure 3: *Mars network-based navigation scheme for Mars entry* _(Yu, 2023)*. The two-was Doppler effect of orbiters, as well as the ground-based beacons, could increase navigation accuracy.*

1.6 Flight Performance on Guidance Algorithms

The entry guidance system determines steering commands to guide the hypersonic gliding spacecraft from its current location at the top of the atmosphere to a pre-specified target landing location. Its main purpose is to control the dissipation of energy and to satisfy range requirements, given specific constraints (entry corridor).

The development of trajectory-tracking guidance for planetary entry can be divided into three generations, presented in *Figure **4***) entry guidance for low-lifting capsule vehicles, 2) Space Shuttle guidance and 3) the newly emerged computational entry guidance (Simplicio, 2018).

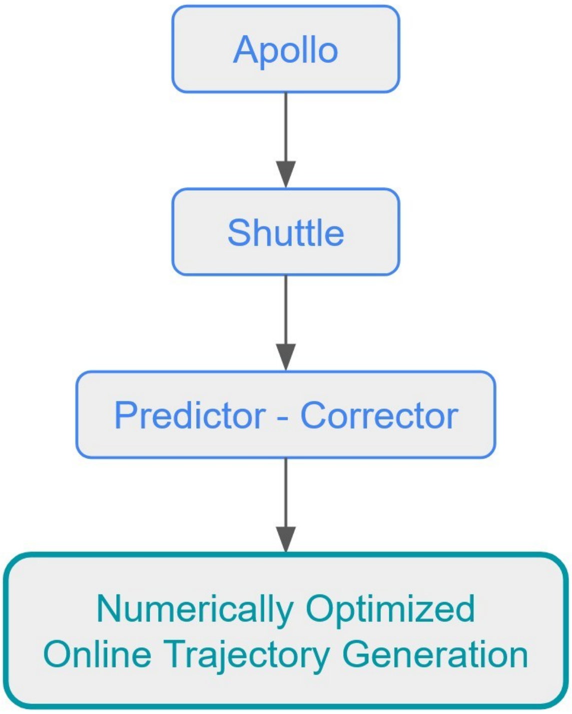


Figure 4: *Evolution of the guidance systems, starting with the early Apollo guidance during the 60s and 70s, then the Shuttle guidance (during the 80s) oriented specifically to vehicles with higher hypersonic L/D ratios, and employing reference-tracking technologies. Predictor-corrector (and its inefficiencies) was presented at the beginning of the 21st century, and online trajectory generation based on numerical optimization (such as convex optimization + proper discretization methods) is the current state-of-the-art guidance algorithm, employed in modern spacecraft reaching the Red planet.* (produced in Google Slides by the author of this research)

The first-generation entry guidance was developed for low lift-to-drag (L/D) ratio capsules such as the Apollo Command Module (CM) spacecraft _(Ingo Gerth, 2014), the Mars MSL and Perseverance capsules and SpaceX's Dragon spacecraft capsule. The entry guidance utilized in the Apollo program marked the initial implementation of this technology. In this program, the spacecraft followed a predetermined angle-of-attack profile, and the guidance system generated commands to adjust the bank angle to minimize the predicted downrange error. To determine the appropriate bank angle, a bank reversal logic was employed based on the cross-range requirement. Due to the limited computational capacity of the onboard guidance computer, the Apollo guidance approach relied on closed-form analytical equations and approximate relationships. It aimed to converge towards a predefined trajectory stored as a reference table.

The primary advantage of this entry guidance method for vehicles with low lift-to-drag ratios (L/D) lies in its simplicity and quick solution time, which are advantageous for hypersonic entry flights. However, a major drawback is its reliance on simplified assumptions, which reduces the robustness of the guidance algorithm and restricts its suitability for a broader range of missions. Consequently, the entry guidance algorithms developed for low L/D vehicles are primarily employed for short-range flights with moderate payloads.

For reference, another descent guidance algorithm of the Lunar Excursion Module (LEM) was used to guide the spacecraft during its descent to the surface of the Moon. One of the key features of the Apollo descent guidance algorithm was its ability to handle the last stages of the descent, where the LEM transitioned from a powered descent to a free fall and then to a landing. The algorithm was able to guide the spacecraft to a safe landing spot, even if the navigation system was not providing accurate information. It was a major achievement in the guidance field and played a critical role in the success of the Apollo manned lunar landing missions.

The second-generation entry guidance algorithm was specifically developed to cater to vehicles with higher lift-to-drag ratios (L/D) such as X-33, X-37B and the Space Shuttle, as shown in *Figure* ***4*** (NASA Lyndon JSC, 1979). Unlike the Apollo entry approach, the Shuttle maintained a relatively low flight-path angle for most of the entry phase. A predetermined nominal angle-of-attack profile was established, considering for the Thermal Protection System (TPS) and cross-range requirements. However, the angle of attack profile was adjusted within a small range during the flight to obey path constraints (40 degrees at the top of the atmosphere to dissipate as much energy as possible before hitting the thick layers of the atmosphere, and 20 degrees later in the thick layers of the atmosphere, after the structure of the entry vehicle has cooled down) (Mooij, Re-entry Systems, 2020).

The Shuttle employed a reference-tracking methodology, where a reference trajectory in the longitudinal direction was devised as a profile of drag acceleration versus velocity or drag acceleration versus energy. This approach aimed to align the actual trajectory with the reference trajectory, allowing for precise control and guidance throughout the entry phase. The magnitude of the bank angle command was determined using a linearized, gain-scheduled guidance law that aimed to track the reference drag profile. Similar to the Apollo entry guidance algorithm, a bank reversal logic was employed to determine the direction of the bank angle. Extensive efforts have been made to enhance the performance of the drag-acceleration-based entry guidance method. These efforts include improving the accuracy of the drag profile design, simplifying the process of designing the reference trajectory, and developing robust feedback control laws for trajectory tracking.

The Shuttle's entry guidance method is straightforward to implement onboard, and the guidance algorithms designed for vehicles with high lift-to-drag (L/D) ratios can handle longer ranges, enable runway landings, and accommodate large payloads. However, the development of the guidance algorithm heavily relies on certain assumptions, namely a small flight-path angle and a quasi-equilibrium glide condition. Despite its advantages, the applicability of the guidance algorithm may be limited by these underlying assumptions.

Unlike the Apollo or Shuttle entry guidance algorithm, the third-generation entry guidance algorithms rely heavily on numerical computation, taking advantage of modern computing capabilities, to generate real-time trajectories. One particularly promising algorithm in this category is the predictor-corrector method (*Figure* ***4***), which has demonstrated significant potential in addressing large trajectory dispersions.

In one study _(Lu, 2007), a three-degree-of-freedom $(3DoF)$ predictor-corrector guidance algorithm was developed specifically for evaluating high-energy aero-braking maneuvers. Another study _(Sippel, 2019) designed an online entry guidance algorithm using the predictor-corrector method for a reusable launch vehicle (RLV). Utilizing the Quasi-Equilibrium glide condition, yet another study _(Thilbault, 2019) proposed a methodology for rapidly designing feasible $3DoF$ entry trajectories that satisfy various inequality and equality constraints. This methodology decomposed the highly constrained trajectory planning problem into two sequential one-parameter search problems, allowing for the generation of viable entry trajectories in approximately $2-3$ seconds.

To tackle the challenges posed by terminal and path constraints in entry flight, a numerical predictor-corrector approach was adopted in a study (Lu, 2007), presenting an algorithm for real-time trajectory planning. This algorithm aimed to generate a feasible trajectory in each guidance cycle. In the context of adaptive and accurate entry guidance for low-lifting entry vehicles, a numerical predictor-corrector method was investigated in another study (Lu, 2007). The performance of this algorithm was further improved and demonstrated for different types of vehicles and missions in subsequent research.

The predictor-corrector guidance algorithms offer several primary advantages:

1. They have the capability to update reference trajectories during the flight.
2. They can integrate reference trajectory planning with tracking guidance.

However, one longstanding challenge of the predictor-corrector guidance algorithm is the substantial computational requirement necessary for repeated numerical integration of the equations of motion. As the most reliable algorithms today typically iterate on a single guidance parameter, achieving convergence with more general numerical approaches remains a concern. Another challenge encountered with these algorithms is the absence of an effective method to accommodate inequality path constraints related to factors such as heat rate, normal load, and dynamic pressure. Furthermore, existing predictor-corrector guidance algorithms face limitations in generating only feasible trajectories, which restricts their ability to handle a broader range of trajectory requirements.

An ideal algorithm for onboard entry guidance should be fast, reliable, readily implementable, and able to enforce common constraints while ensuring accuracy for various types of vehicles and missions. If the mission is modified during flight to address emergent situations, the entry guidance system should re-generate a new trajectory to the target in shortest possible amount of time.

With the development of computing technologies, optimization theories, and algorithms, online trajectory optimization (latest stage, shown in *Figure **4***) is becoming an effective approach for solving aerospace guidance problems. In addition, convex optimization methods enable solving convex, nonlinear programming (NLP) problems as easily as solving linear programming (LP) problems, and a globally optimal solution can be computed for a convex problem. In each guidance cycle of the algorithm, a reference trajectory is generated from the current location to the target by solving Second-order cone programming (SOCP) problems _(Wikipedia, 2023). Path constraints such as the heat rate, normal load and the dynamic pressure are considered in both the reference trajectory generation and the optimal tracking guidance design. The novelty of this approach is its capability to generate accurate reference trajectories in several guidance cycles before an exact convergence is achieved.

Such high precision online trajectory optimization methods, combining convex optimization and pseudospectral optimization to achieve precision landing, have several major benefits:

1. Convergence certainty of the convex optimization scheme
2. High discretization precision of the pseudospectral method
3. The algorithm can converge without good initial guesses
4. Interior point methods (IPM) can solve convex problems in polynomial time without initial guesses as a user input

2

Research Gap, Problem Statement and Research Objective

2.1 Research Gap

The Unexplored Domain: Elliptical Orbit Entry Paradigm for Martian Hypersonic Descent

After decades of spacecraft re-entering Mars' atmosphere, there are still considerable research gaps when it comes to the methodological approach to re-entry — a critical juncture where theoretical calculations meet the harsh realities of Martian environmental constraints. The predominant body of scholarly work has focused almost exclusively on direct entry scenarios, where spacecraft dives into the atmosphere of Mars at velocities approaching $7.5\ km/s$, having arrived from interplanetary trajectories without orbital insertion around the planet. This approach, while technically feasible, imposes severe thermal and structural demands on the vehicle, requiring robust heat shields, availability of refurbishment capabilities and offering minimal margin for trajectory refinement. This established paradigm has remained largely unchallenged in literature, creating a notable absence of research into alternative entry methods that may offer advantages for future missions, particularly those involving human payloads or precision landing requirements.

This research into Starship's Mars entry profile introduces a fundamentally different approach – an entry from an established elliptical orbit with eccentricity of $e \cong 0.5$, at a considerably reduced velocity of $4.3 \ km/s$ (exhibited in *Figure **5***).

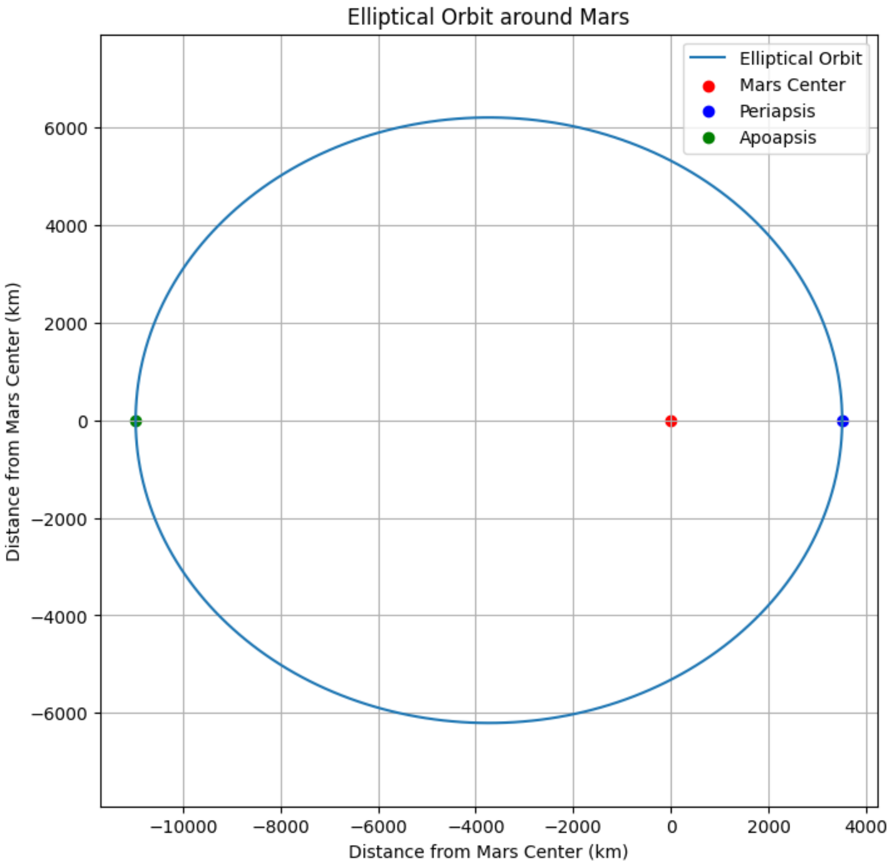


Figure 5: *Starship's elliptical orbit around Mars, a starting point periapsis re-entry into the atmosphere* (produced in MATLAB by the author of this research)

The spacecraft essentially aerobrakes into the Martian atmosphere through several orbits, dissipating its interplanetary trajectory energy. This methodology, while maintaining the standard entry interface altitude of $120 \ km$, represents a departure from conventional wisdom. The implementation of sequential convex programming as an algorithm for non-convex hypersonic guidance further distinguishes this work from traditional approaches. This mathematical framework transforms the inherently non-convex trajectory optimization problem into a sequence of convex sub-problems, enabling computational efficiency critical for autonomous guidance during the upper atmospheric descent phase.

The advantages of using elliptical orbit entry approach are manifold and significant. Most prominently, the reduced entry velocity V_0 translates directly to decreased peak heating rates q_{max} – potentially by a factor of three to four _(Lu, 2007) compared to direct entry – thereby alleviating thermal protection system requirements (and as a consequence, mass and complexity requirements), and expanding the envelope of material selections. The lower velocity regime also affords greater aerodynamic control authority during the hypersonic phase, enhancing precision landing capabilities crucial for infrastructure development scenarios. Furthermore, the elliptical orbit provides a stable platform for comprehensive systems verification before committing to the irreversible entry sequence, allowing for timing flexibility to avoid dust storms or other adverse atmospheric conditions that might compromise mission success.

Perhaps most significantly, this approach introduces a critical abort capability absent in direct entry scenarios. Should anomalies be detected prior to deorbit burn initiation, the vehicle can remain safely in its elliptical orbit while engineers develop contingency plans — an option that is not available during the unalterable trajectory of direct atmospheric entry. Additionally, the more gradual energy dissipation profile enables more precise trajectory modulation, with the sequential convex programming framework continuously recalculating optimal control parameters throughout descent.

This research contributes to the re-entry field by challenging the entrenched assumption that direct entry represents the optimal or sole viable approach for Mars entry. It bridges theoretical optimization techniques, such as convex optimization, with practical engineering constraints, offering another perspective that balances computational efficiency with trajectory optimality. While direct entry may remain appropriate for certain mission profiles, particularly where payload mass is severely constrained, this work demonstrates that for vehicles with substantial propulsive capabilities such as Starship, the elliptical orbit entry approach offers compelling advantages that have remained largely unexplored in literature. By illuminating this alternative methodology and quantifying its benefits through simulations, this research expands the design space available to mission architects and potentially enables more ambitious and precise Mars landing operations than previously considered feasible under the direct entry paradigm.

2.2 Problem Statement and Research Objective

In the previous chapter, in Section 1.4, several key challenges regarding precision landing on Mars were presented. To summarize the problem statement, Mars entry guidance faces a critical challenge: navigating hypersonic velocities within thin atmospheric layers while balancing conflicting objectives of precision targeting, thermal protection system survival, and vehicle structural integrity. Current trajectory optimization methods for interplanetary re-entry, such as Apollo-derived predictor-corrector algorithms, struggle with the nonlinear dynamics and path constraints inherent to Mars' variable CO_2-rich atmosphere. These methods often rely on pre-computed profiles ill-suited for real-time adaptation to atmospheric dispersions, risking corridor breaches that could lead to catastrophic heating or undershoot errors. The above problem statement demands the Starship vehicle's hypersonic glide phase – governed by bank angle and angle-of-attack modulation – to implement a guidance framework capable of resolving convex-compatible sub-problems rapidly under tight computational margins, a capability absent in legacy systems.

This thesis addresses these limitations by formulating a successive convexification – based algorithm that decomposes the nonconvex entry problem into iteratively refined convex sub-problems. Accordingly, the main purpose of this research is to develop a capable guidance algorithm for the hypersonic portion of the entry glide on Mars, that can guide the vehicle to a pre-defined location on a $45\ km$ altitude, thereby following an optimal trajectory to achieve maximum precision (less than $3\ km$). After an altitude of $45\ km$, atmospheric density on Mars increases considerably and other methods, such as Terrain Relative Navigation (TRN) (from 45 to $20\ km$) and retro – propulsive methods ($20\ km$ and below) are needed to satisfy the accuracy requirement. The terminal accuracy at $45\ km$ is a good prerequisite to achieve a subsequent *precision-landing in the order of hundreds of meters*, using the aforementioned methods. The primary challenge associated with GNC systems for achieving precise trajectory optimization lies in the limited robustness and the presence of uncertainties and errors in navigation sensors themselves, also atmospheric properties, wind conditions, and modeling inaccuracies. These factors collectively restrict the achievable accuracy of the landings. Therefore, this research focuses on two aspects:

1. Develop a closed-loop guidance algorithm, based on convex optimization, successive convexification and collocation discretization, to increase robustness.

2. Enforcement of entry corridor physical constraints – heat flux, g-load, equilibrium glide and dynamic pressure – to ensure a feasible landing process.

Based on the discussed challenges, the defined problem statement above and the established focus aspects of the research, the *core* research question is:

What accuracy can successive convexification achieve when applied to precision trajectory optimization for Starship's hypersonic glide in the upper atmosphere while enforcing hard physical constraints on heat flux $(\leq 100\ kW/m^2)$, g-load $(\leq 3g)$, and dynamic pressure $(\leq 800\ Pa)$, considering also theoretically perfect navigation and controls?

Narrowing down this research question into partial sub-questions breaks down the complexity of the problem and gives rise to a clear research overview. Each research sub-question is identified with a unique **RSQ – x**.

RSQ – 1: What is the optimal balance between bank angle modulation and angle-of-attack adjustments to achieve maximum precision?

RSQ – 2: Are the enforced path constraints (heat flux, g-load, equilibrium glide and dynamic pressure) that form the re-entry corridor satisfied, without compromising terminal positioning accuracy?

RSQ – 3: How sensitive is the guidance architecture to uncertainties in initial flight-path angle and velocity, as well as atmospheric gradients and is it robust enough?

3

Spacecraft Re-entry Guidance on Mars

3.1 From Apollo to Numerical Closed-Loop Guidance – Mission Overview

For many decades, researchers have devoted attention to studying the reentry guidance problem. Initially, during the 1970s, the focus was mainly on utilizing the classical guidance law that relied on the drag acceleration profile. However, in the 1990s, the introduction of a new generation of reusable launch vehicles (RLVs) led to the development of new methods. As a result, an Evolved Acceleration Guidance Logic and a guidance law that utilized the Quasi-Equilibrium Glide Condition were proposed (Cui, 2017). In more recent years, computational guidance has been extensively researched for designing re-entry guidance laws.

Predictor-corrector guidance, which was initially developed to explore Mars and has demonstrated high levels of performance in terms of both robustness and flexibility, represents an important direction for designing reentry guidance laws (Lu, 2007). Some researchers have proposed various predictor and corrector guidance laws, such as Newton - Raphson iterations and Neural network, for this purpose. (Wang, 2020) further extended the predictor-corrector guidance approach by incorporating no-fly zone (NFZ) constraints into the trajectory design process. In this extension, a fuzzy logic-based corrector was used to reverse the bank angle and avoid the NFZ, but this approach was not optimal (Zhao, 2017).

With the significant increase in computational power of embedded computers, the requirements for reentry guidance law design have become more demanding. Now, in addition to satisfying terminal and path constraints such as maximum heat flux, maximum dynamic pressure, maximum normal load, NFZ constraints, and hard constraints on terminal states, the guidance law design also aims to minimize the terminal velocity, heating load, or reentry time. In other words, the reentry guidance law design problem has evolved into a trajectory optimization problem.

The problem of optimizing the reentry trajectory has been widely studied, and existing methods can be categorized as indirect or direct _(Wan, 2022). Indirect methods involve explicitly solving the optimality conditions using adjoint differential equations, the maximum principle, and associated boundary conditions. The necessary condition for optimal control can be derived by setting the first derivative of the Hamilton function to zero using the calculus of variations. Although these methods can guarantee optimality, they require initial guesses for the adjoint variables and the sequence of sub-arcs, which must be constrained and unconstrained, before iteration can begin.

In contrast, direct methods do not require an analytical expression for the necessary condition and do not need initial guesses for the adjoint variables. Instead, the original problem is transformed into a nonlinear programming (NLP) problem by introducing parametric representations for the control variables. Although this approach can handle the reentry trajectory optimization problem with no-fly zone constraints and transform it into a large-scale NLP problem that can be solved using an NLP solver, the solution process is time-consuming and unpredictable. A good initial guess is also necessary for the NLP algorithm (Boyd, 2004).

To quickly solve trajectory optimization problems, convex optimization has been introduced as a method of handling the problem _(Boyd, 2004). Convex optimization is becoming increasingly popular in aerospace guidance, control, and trajectory optimization, and has been applied in various situations, such as path planning and collision avoidance for unmanned aerial vehicles, the Mars landing problem, rendezvous and proximity operations, and constrained guidance law design, as well as the trajectory optimization problem.

The superior performance of convex optimization has been demonstrated, and Second-order Cone Programming (SOCP), which is a sub-class of convex optimization with constraints that must be either linear equality or Second-order cone inequality, can be solved in polynomial time using state-of-the-art interior-point methods. Primal-dual IPM does not require initial guesses.

(Liu, 2019) et al. proposed an SOCP method to address the reentry trajectory optimization problem, where the equations of motion were reformulated with respect to energy, the velocity was approximated as a function of energy, and the non-convex path and control constraints were transformed into linear and Second-order cone constraints, respectively. A regularization term was required in the objective function to activate the Second-order cone constraints, and the problem was finally solved using SOCP via successive linearization and relaxation.

The same techniques have been utilized to solve other reentry problems, such as the maximum cross range problem and smooth reentry trajectory planning. However, these methods have limitations as the bank angle profiles are restricted to positive values to reduce the nonlinearity of the optimization problem for a feasible solution, which is not realistic. To address these limitations, (Wang, 2020) proposed a Second-order cone programming (SOCP) approach to minimize the terminal velocity and heat load. They introduced a new control variable, bank angle rate, to separate the states and control variables and employed a First-order Taylor series expansion to convexify the path constraint. However, the validity of their approach has not been accurately demonstrated, indicating that the original inequality path constraints may not hold. Another drawback of this method is that infeasible problems may arise during the successive solving process due to the high nonlinearity of the reentry optimization problem.

In this research, a sequential convex programming method for reentry trajectory optimization is proposed, capable of potentially handling also problems with no-fly zones (NFZs) if further improved by future efforts. To handle the non-convex path constraints such as the nonconvex heat rate, dynamic pressure, and normal load path constraints, a linearization approach is implemented. In order to preserve the feasibility of the sub-problems in the SCP solving process and to keep the optimality of the final solution, a penalty term is added to the objective function.

3.2 Optimal Control Problem (OCP) and the Bolza Problem

There are different methods to generate optimal trajectories. Some of them exploit the structure of the problem and require simplifications to transform the problem to a mathematically tractable thing. Therefore, the generated solutions are valid under given hypotheses. However, nowadays aerospace engineers employ a standardized contemporary approach, which benefits from the development of the computational capabilities of modern CPUs. This method represents the trajectory generation problem as an optimal control problem (OCP) (Wikipedia, n.d.). This means that the goal is to minimize a given criterion, while at the same time satisfying several constraints, which can be differential (i.e., the EoMs of the vehicle) and/or algebraic (e.g., the maximum heat-flux that the vehicle can tolerate during re-entry).

The standard form for representing OCPs is the so-called Bolza problem (Mooij, 2018). Given a state vector $\boldsymbol{x}(t) \in R^n$, a control vector $\boldsymbol{u}(t) \in R^n$, the scalar functions $\Phi(t, \boldsymbol{x}, \boldsymbol{u})$ and $\Psi(t, \boldsymbol{x}, \boldsymbol{u})$ and the vector $\mathrm{g}(t, \boldsymbol{x}, \boldsymbol{u}) \in R^n$, the problem can be formulated as follows:

$$J = \Phi\left[t_f, \mathbf{x}\left(t_f\right), \mathbf{u}\left(t_f\right)\right] + \int_{t_0}^{t_f} \Psi[\mathbf{x}(t), \mathbf{u}(t)]\, dt \tag{1}$$

where J is the objective to be minimizes. The above is subject to the differential equations

$$\dot{\boldsymbol{x}} = f(t, \mathbf{x}, \mathbf{u}) \tag{2}$$

and to the path constraints $\boldsymbol{g}_l \leq \boldsymbol{g}(\mathbf{x}, \mathbf{u}) \leq \boldsymbol{g}_u$.

The first term in the cost function of *Equation* ***(1)*** is called Mayer term and represents punctual constraints (e.g., minimization of a distance according to a given metric) while the integral argument is called the Lagrange term, used to minimize variables over the entire mission (e.g., the heat load obtained by integrating the heat flux over time). As already seen in the example above, the problem had bounded states and controls, $\mathbf{x}_L \leq \mathbf{x}(t) \leq \mathbf{x}_U$ and $\mathbf{u}_L \leq \mathbf{u}(t) \leq \mathbf{u}_U$. In addition, initial and final conditions' constraints are applied such that the system dynamics and other constraints in the OCP serve as the constraints in the Bolza problem framework.

3.3 Convex Optimization – An Overview

Convex optimization, a branch of mathematical optimization, focuses on minimizing convex functions within convex domains (Boyd, 2004). Unlike general optimization tasks which are typically NP-hard (computationally intensive), many convex problems can be solved efficiently using polynomial-time algorithms. Solving a problem in polynomial time means the algorithm's running time grows proportionally to n^k, where n is the input size and k is a fixed constant (Math Overflow, 2024). While verifying solutions is straightforward, solving complex non-convex problems becomes prohibitively time-consuming. Convex optimization methods circumvent this by enabling efficient resolution of such problems with guaranteed convergence (Boyd, 2004).

In aerospace guidance, convex techniques reformulate nonlinear, constrained entry challenges into sequences of Second-order cone programming (SOCP) sub-problems through iterative linearization and convex approximation methods (also implemented in this research) (Boyd, 2004). As a specialized convex programming sub-class, SOCP benefits from solvers with predictable convergence characteristics. This efficiency makes SOCP-based guidance algorithms viable for real-time spacecraft applications (Boyd, 2004).

As the equations of motion (EoMs), presented in Chapter 4, for the entry guidance problem are highly nonlinear in general, sequential convex programming (SCP) with the successive linearization technique has been widely adopted in previous studies. In the generic SCP method, a nonlinear optimization problem is sequentially solved by converting it into a convex sub-problem at each iteration (Boyd, 2004). Although a strict convergence property of the convex optimization is weakened in the SCP method, it still has been recognized as an efficient heuristic method for solving nonlinear optimization problems. Moreover, it has been successfully applied to a real-world guidance problem. However, there are still some issues to be improved in the generic SCP method in the entry trajectory optimization problem (Pei, 2021). Moreover, a change in the control variables (as proposed in this research) and a relaxation technique would potentially solve a high-frequency jittering issue of the trajectory-optimized entry path (Pei, 2021).

In the past three decades, numerous researchers have dedicated their efforts to the advancement of convex optimization theory (Bae, 2022). They have proven that, for a significant range of problems, the crucial characteristic is not the linearity of the system, but its convexity. When the problem is convex, it can be solved quickly, and the resulting solution is the global optimal, if it exists. In most cases, a convex optimization problem is described as follows:

$$\min J = f_0(x) \tag{3}$$

subject to

$$f_i(x) \leq a_{i,} \qquad i = 1, \dots, m \tag{4}$$

where $x \in R^n$ represents the vector of variables to be determined. The functions f_i are convex which means that they satisfy Jensen's inequality (Wikipedia, n.d.):

$$f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y), \qquad i = 0, \dots, m, \;\; \alpha + \beta = 1, \;\; \alpha \geq 0, \;\; \beta \geq 0 \tag{5}$$

The above *Equation (**5**)* highlights a key feature of convex problems, which extends the concept of linearity to convexity, and includes equality as a special case instead of the original Jensen's inequality. More information about this can be found in Boyd's research. The following properties define convex optimization:

- It is considered a matured technology because there are efficient methods, such as primal – dual interior point methods (PD-IPMs) that can solve convex problems
- Many real-world problems can be reformulated in a convex form
- This type of methods does not require an initial guess, contrary to non-linear programs (NLPs)
- It is guaranteed that the solution to the problem is a global optimum

In this work, we will focus on Second-order Cone Programming (SOCP), which, as already mentioned, is a specific sub-class of convex programs.

3.4 Second – Order Cone Programming (SOCP)

Second-order cone programming (SOCP), a sub-class of convex optimization, has emerged as a transformative tool for solving complex trajectory optimization problems in aerospace engineering, particularly in the high-stakes domain of planetary re-entry systems (Acikmese, 2022). By minimizing linear objectives over intersections of affine spaces and Lorentz (Second-order) cones, SOCP provides a computationally tractable framework for handling nonlinear dynamics and non-convex constraints inherent to atmospheric entry. Its ability to model quadratic inequalities and Euclidean norms as convex constraints makes it uniquely suited to address the multifaceted challenges of guiding vehicles through extreme thermal and aerodynamic environments (Acikmese, 2022).

The SOCP Framework

SOCP generalizes linear and quadratic programming by introducing constraints defined by Second-order cones (SOC) such in the one exhibited in *Figure **6***:

$$\mathrm{K} = \{(x_0, \mathbf{x}) \in R \times R^{n-1} \mid x_0 \geq \|x\|_2\} \tag{6}$$

In *Equation (**6**),* K refers to the coefficient matrices $(3x3)$ derived from least squares fit of the descent trajectory. They enable the guidance algorithm to compute optimal trajectories without introducing additional computational phases. This structure enables the efficient representation of diverse constraints, from trust regions to aerodynamic load limits, while guaranteeing convexity. For re-entry applications, SOCP's strength lies in its compatibility with *successive convexification – an* iterative technique that linearizes non-convex dynamics and path constraints around a reference trajectory, transforming them into SOCP – compatible forms. It can be formulated as follows (also presented as ***Problem 2*** in Chapter 5):

$$\text{minimize } c_0^T x \tag{7}$$

subject to

$$\begin{gathered} F_i x \leq G_i \\ A_0 x = b_0 \\ \|A_i x + b_i\|_2 \leq c_1^T x + d_i \end{gathered} \tag{8}$$

with $x \in R^{n\times 1}$ representing the variables to determine, $c_0 \in R^{n\times 1}$ is the definition vector of the objective, $F_i \in R^{l\times n}$ and $G_i \in R^{l\times 1}$ are a set of component-wise inequalities, $A_0 \in R^{m\times n}$ and $b_0 \in R^{m\times 1}$ describe the linear system of m equations that the solution has to satisfy. The terms $A_i \in R^{m\times n}$, $b_i \in R^{m\times 1}$, $c_i \in R^{n\times 1}$ and $d_i \in R$ describe a conic constraint of order $m_i + 1$. These constraints imply that the solution will always be contained within the volume of each of the pm_i - dimensional cones. They are also subjected to the following affine (linear) transformations (depicted also in *Figure **6***):

$$\begin{gathered} t = c_1^T x + d_i \\ y = A_i x + b_i \end{gathered} \tag{9}$$

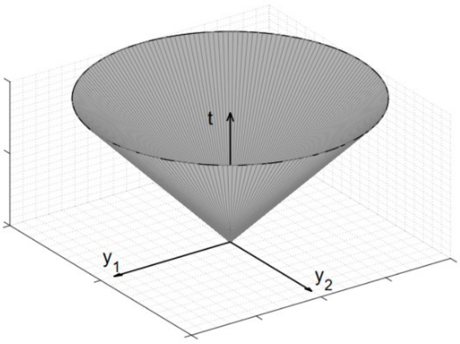


Figure 6: *Example of a 3D cone. Its volume satisfies the condition* $\|y\|_2 \leq t$ (Boyd, 2004)

Re-entry Challenges and SOCP Solutions

Planetary entry vehicles face stringent operational limits:

- **Thermal constraints:** Heat flux $\dot{q} \leq \dot{q}_{max}$
- **Structural limits:** Dynamic pressure $q \leq q_{max}$ and load factor $n \leq n_{max}$
- **Guidance precision:** Terminal position and velocity targets

Traditional optimization methods struggle with their nonlinear, tightly coupled constraints. SOCP addresses this by the already mentioned *successive convexification:* nonlinear equations of motion (EoMs) are linearized iteratively around a reference trajectory, with trust regions (enforced via SOCP) ensuring convergence. The dynamics in *Equation* ***(8)*** become affine approximations:

$$\dot{\mathrm{x}} \approx A_k\mathbf{x} + B_k\mathbf{u} + \mathbf{c}_k \tag{10}$$

where A_k, B_k and $\mathbf{c}_k$ are the Jacobians evaluated at iteration k. A re-entry vehicle must navigate a narrow entry corridor bounded by heat flux, dynamic pressure, and g-load limits, and the SOCP helps by:

- **Encoding the three path constraints** as described in the Methodology Chapter **5**
- **Integrating slack variables** to handle infeasibilities during iterative refinement
- **Employing 2-norm trust regions** to stabilize convergence, avoiding divergence caused by aggressive linearization

The Apollo program's legacy guidance, reliant on precomputed drag profiles, contrasts sharply with modern SOCP – based methods that autonomously adjust bank angles of the re-entry winged vehicles, and as a consequence, also the lift vectors in response to disturbances.

While SOCP has revolutionized entry guidance, compared to Apollo landing techniques (described at the end of Chapter 1), there are some challenges that remain:

- **High – fidelity aerothermal models:** Uncertainties in atmospheric density and vehicle aerodynamics necessitate robust SOCP formulations with chance constraints.
- **$6DoF$ Integration:** Coupling translational SOCP trajectories with attitude dynamics requires novel cone constraints for rotational inertia and torque limits.
- **Interplanetary Scalability:** Mars missions demand SOCP adaptations to heterogeneous atmospheric compositions and non – Keplerian dynamics.

It can be concluded from this section that Second-order cone programming bridges the gap between theoretical convex optimization and the harsh realities of planetary entry. By transforming nonlinear, non-convex problems into sequences of tractable SOCP subproblems, it enables real-time trajectory optimization under the most extreme conditions. As demonstrated in hypersonic glide tests and Mars landing simulations, SOCP's blend of computational efficiency and mathematical rigor ensures that humanity's ventures into alien atmospheres remain both feasible and precise.

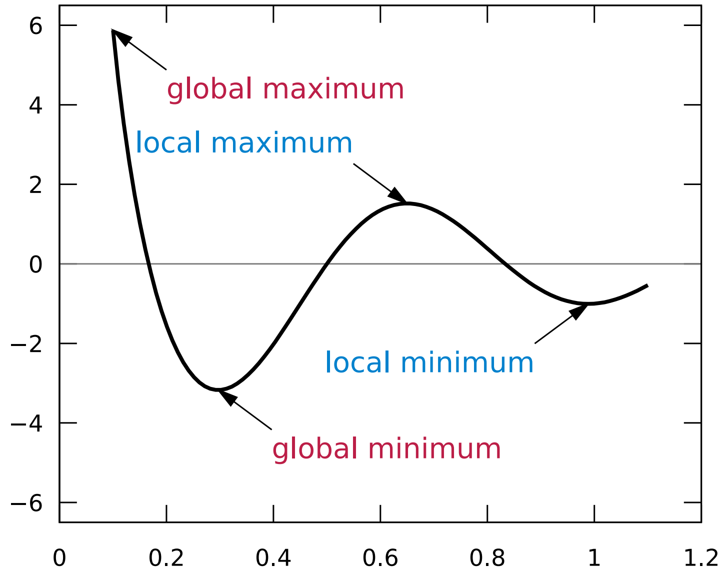


Figure 7: *Global and local optima of a function* (UTSA | The Univesity of Texas at San Antonio, n.d.)

Related to how convex optimization helps in re-entry trajectory optimization, *Figure **7*** illustrates the crucial distinction between local and global optima, a key consideration when applying convex optimization to complex problems like the re-entry of vehicles on Mars. In the context of Mars re-entry, the "function" being optimized could represent a performance metric such as maximizing landing accuracy, while as minimizing heat shield mass or minimizing peak deceleration. The trajectory of the re-entry vehicle is determined by the control inputs α and σ, and the design space of these inputs can lead to a multi-dimensional "surface" with various peaks and valleys representing local maxima and minima of the performance metric.

For a successful and safe Mars landing, finding the **global optimum** solution is paramount. A local optimum, while appearing to be the best solution in its immediate vicinity, might correspond to a re-entry trajectory that leads to unacceptable heating, misses the landing site by a large margin, or subjects the vehicle to excessive forces. Convex optimization techniques are highly desirable in this domain because, for a properly formulated convex problem, any local minimum found is guaranteed to be the global minimum _(UTSA | The Univesity of Texas at San Antonio, n.d.). This eliminates the risk of getting trapped in a suboptimal local solution and ensures that the designed re-entry trajectory truly optimizes the desired performance criteria for the Martian atmosphere.

3.5 Atmospheric Re-entry with Starship

As SpaceX's Starship system advances, the prospect of a fully reusable launch system looms on the horizon, potentially ushering in a new era of interplanetary transport (Blackmore, 2016). This revolutionary development could enable rocket-propelled cargo and passenger conveyance between Earth and Mars at velocities surpassing orbital speeds. With Starship's inaugural orbital flight test already behind us, and multiple flight tests conducted already, the realization of a fully reusable orbital transportation system seems increasingly tangible. Should SpaceX achieve its ambitious goal of rapid and economical stage reuse, it stands to dramatically reduce space payload transport costs and reshape the space transportation landscape.

Since its 2016 unveiling, SpaceX's next-generation space transportation system has undergone numerous design iterations, retaining key features such as full reusability and Full-

Flow Staged Combustion (FFSC) cycle engines (Academy, 2023) while refining aspects like weight and material composition. The current iteration, Starship and Super Heavy, has made significant strides in stage hardware integration and is poised to be the first fully integrated version attempting orbital flight and Mars-bound cargo and human transport (Wilken, 2022).

Starship's reentry process comprises three distinct phases, as illustrated in *Figure **8***, hypersonic deceleration at a $\sim 50^o$ angle of attack (AoA), skydiving at nearly $90°$ AoA, and a landing burn initiated from $\sim 180°$ AoA. The system's first stage is expected to employ a Return-To-Launch-Site (RTLS) scheme akin to that of the Falcon 9 (Wilken, 2022).

Given Starship's ultimate goal of facilitating Martian colonization, its landing approach must be versatile enough to function in diverse atmospheric conditions, from Earth's relatively dense atmosphere to Mars' tenuous envelope, accounting for trajectory dispersions (shown in *Figure **8***).

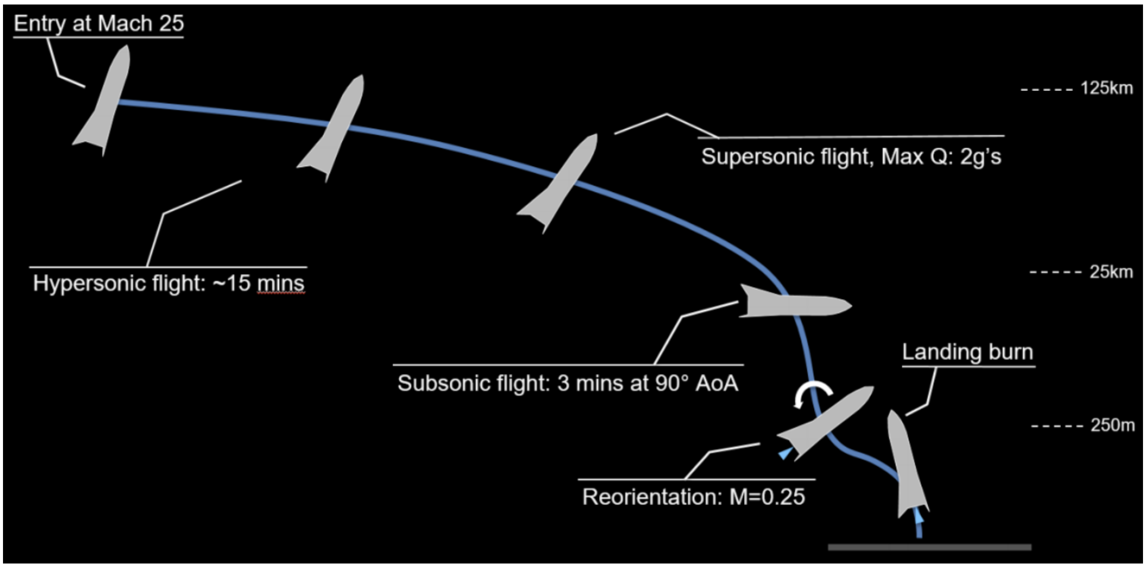


Figure 8: *Starship entry corridor mechanics* (Wilken, 2022)

The primary objective of all reusable launch vehicles during reentry from orbit is to safely decelerate from orbital velocities without surpassing mechanical and thermal limits, and to land within a designated area. To minimize heat fluxes to an acceptable level for the overall system design, the aim is to dissipate most of the energy in the upper layers of the atmosphere, only entering the denser lower layers with reduced velocity. One way to accomplish this is by using a design with a very low ballistic coefficient C_b (Wilken, 2022). The C_b is the ratio of the reentry mass m to the drag coefficient C_d and the reference area A.

$$C_b = \frac{m}{C_d A} \tag{11}$$

With a low enough C_b, the low density of the upper atmosphere can decelerate the vehicle without creating excessive heat fluxes. A vehicle that can generate significant lift allows for a shallower reentry that can further decrease the peak heat flux. However, the normal n_z loads caused by the lift during reentry may constrain the sizing of the vehicle's structures, particularly for a crewed vehicle. Another important indicator of how the vehicle behaves during atmospheric re-entry is an additionally equivalent (to the ballistic coefficient) coefficient with regard to lift, as defined below:

$$C_{b,l} = \frac{m}{C_l A} \tag{12}$$

In essence, these coefficients describe to what degree is the atmospheric trajectory affected by the generated aerodynamic Lift and Drag forces. The higher the value, the more “resistant” the spacecraft to any changes from that source. Starship will decelerate fast through the atmosphere, avoiding excessive heat flux but limiting its gliding range, due to its low L/D ratio.

Table 1 delineates the Starship launch system's primary parameters. Both Super Heavy and Starship utilize vertical landing and are retrieved by the launch tower's mobile arms, dubbed Mechazilla (exhibited in *Figure **9***). The system's design choices, particularly regarding propellant and return method, enable it to offset the lower specific impulse of the LOX/LCH4 propellant combination. Starship achieves a commendable 2% payload ratio in fully reusable mode, owing to its high specific impulse I_{sp} and low structural index. However, the current dry mass estimate

Table 1. *Starship and Super Heavy Technical Specifications* _(Wilken, 2022)

Key parameters		Starship and Super Heavy
1st stage	Propellant mass	$3400\ t$
	Dry mass	$270\ t$
	Structural index	$7.9\ \%$
	Total mass	$3670\ t$
	Engine Isp (sea level)	$326\ s$
	Engine Isp (vacuum)	$349\ s$
	Return method	$RTLS$
	Length	$70\ m$
	Fuselage diameter	$9\ m$
2nd stage	Propellant mass	$1200\ t$
	Dry mass	$100\ t$
	Structural index	$10.6\ \%$
	Total mass	$1330\ t$
	Engine Isp (sea level)	$285\ s$
	Engine Isp (vacuum)	$374\ s$
	Length	$50\ m$
	Fuselage diameter	$9\ m$
Total mass		$4997\ t$
Reference payload		$100\ t$
Re-entry aerodynamics	Reference re-entry AoA	52^{o}
	Hypersonic L/D ratio	~ 0.3
	Ballistic coefficient	$510\ \ kg/m^2$
	Lifting ballistic coefficient	$1700\ \ kg/m^2$
Crew Separation System		No information

Figure 9: *Starship and Super Heavy on the launch pad at Boca Chica, Texas* (The Verge, 2021)

excludes payload integration structures and reinforcements for large apertures in the vehicle's upper section. Moreover, the Starship's interior configuration remains undisclosed and is expected to vary significantly based on mission parameters, whether Point-to-Point, LEO satellite deployment, refueling, or ventures beyond Earth orbit to the Moon or Mars.

SpaceX's Raptor 3 engine boasts an exceptionally high combustion chamber pressure of $350\ bar$ (Starship SpaceX Wiki, n.d.). During hypersonic reentry, Starship maintains a $70-$degree AoA, generating less lift than comparable vehicles like Space Liner 7, but compensating with an extremely low ballistic coefficient. Starship's reentry mode prioritizes safe orbital return with minimal inert mass and rapid stage turnaround, rather than long-range hypersonic glide capability akin to the Space Shuttle. Designed for versatility across Earth, Mars, and lunar landings, Starship eschews a high hypersonic L/D ratio (approximately 0.3, compared to Space Liner 7's ~ 3.0 and the Space Shuttle's ~ 4.5) in favor of adaptability to diverse ambient conditions .

3.6 Initial Conditions for Starship Mars Reentry: A Comprehensive Description

Orbital Context

As the Starship spacecraft re-enters through the Martian atmosphere, it is necessary to define the initial conditions that will shape its trajectory. These conditions not only dictate the spacecraft's entry dynamics but also play a pivotal role in ensuring a safe and controlled descent onto the Martian surface.

The spacecraft begins its re-entry from an elliptical orbit around Mars, characterized by a moderate eccentricity of approximately 0.5. This orbital configuration is strategically chosen to optimize the spacecraft's energy state and facilitate a controlled entry into the Martian atmosphere. The periapsis (closest point to Mars) is set at an altitude of $120\ km$, marking the precise moment when the spacecraft transitions from orbital mechanics to atmospheric dynamics.

At this critical juncture, spacecraft's velocity is approximately $V_0 = 4300\ m/s$, a speed that necessitates careful management to ensure a stable and efficient entry. The flight-path angle is set at $\gamma = -15\ deg$, a shallow entry (but not too shallow, avoids atmosphere “skip-out”) that spreads the spacecraft's kinetic energy over a longer atmospheric path, thereby reducing the instantaneous heat flux experienced by the spacecraft thermal protection system, and spreading kinetic energy over a longer atmospheric path.

The elliptical orbit parameters are derived as follows, first the specific energy ε from the vis-viva *Equation (**13**)*:

$$\varepsilon\left[\frac{m^2}{s^2}\right] = \frac{V_0^2}{2} - \frac{\mu}{(R_{Mars} + h)} \tag{13}$$

where V_0 is the initial re-entry velocity, μ is the Mars gravitational parameter, R_{Mars} is the radius of the planet, and h is the initial re-entry altitude. Then, the semi-major axis:

$$a\,[m] = -\frac{\mu}{2\varepsilon} \tag{14}$$

Angular momentum h and eccentricity e:

$$h\left[\frac{m^2}{s}\right] = (R_{Mars} + h)V_0 \tag{15}$$

$$e\,[-] = \sqrt{\frac{1 + 2\varepsilon h^2}{\mu^2}} = 0.515 \tag{16}$$

And periapsis r_p and apoapsis r_a distance:

$$r_p\,[m] = a(1 - e) \tag{17}$$

$$r_a\,[m] = a(1 + e) \tag{18}$$

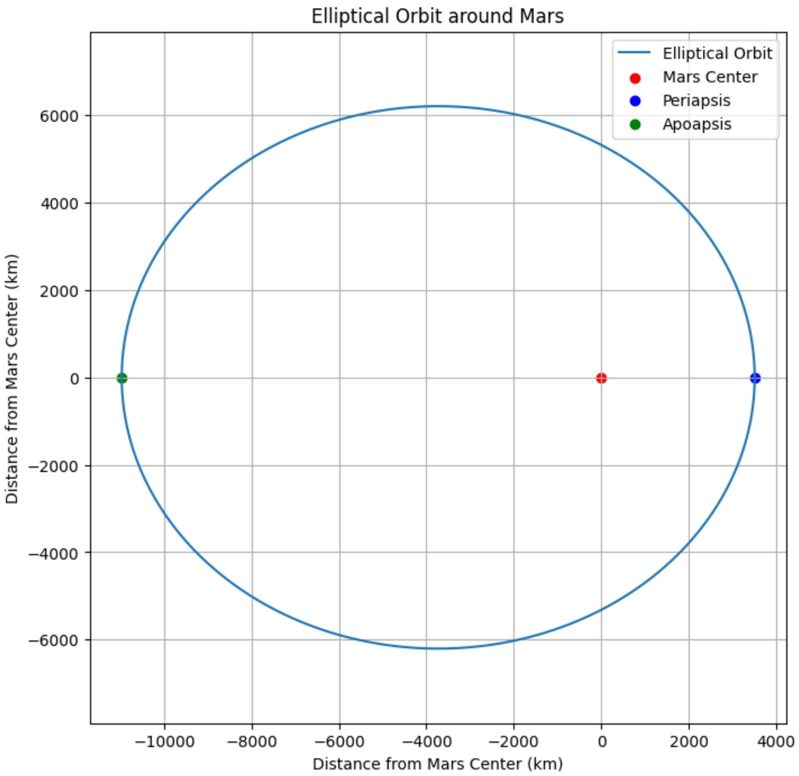


Figure 10: *Starship's elliptical orbit around Mars, a starting point periapsis re-entry into the atmosphere* (produced in MATLAB by the author of this research)

Aerodynamic Configuration

The spacecraft's aerodynamic configuration is equally crucial during this phase. An initial angle of attack of $\alpha = 52\ deg$ is selected to maximize the L/D ratio, which is essential for managing heat dissipation, reducing descent rate and peak heating, as well as maintaining aerodynamic stability. This high angle of attack ensures that the spacecraft enters the atmosphere belly-first, managing heat shield protection from lateral heat flux and ensuring that it remains stable throughout the entry phase. The generation of a lift force counters Mars' gravity, extending the "glide" phase and enabling trajectory shaping for a more accurate landing.

The spacecraft's reference surface area of $A \cong 500\ m^2$ and nose radius of $n = 4.2\ m$ are critical parameters that influence its aerodynamic behavior (Starship SpaceX Wiki, n.d.). These dimensions, combined with the spacecraft's mass of $100\ tons$, dictate the magnitude of aerodynamic forces experienced during entry, including lift and drag. The precise control of these forces is essential for guiding the spacecraft through the Martian atmosphere without compromising its structural integrity.

To navigate the complexities of atmospheric entry, the spacecraft employs sequential convex programming, a sophisticated optimization technique designed to iteratively refine the spacecraft's trajectory. SCP linearizes the nonlinear dynamics of atmospheric entry, convexifies the constraints related to heat flux limits and control authority, and iteratively adjusts the trajectory to ensure feasibility under uncertainty.

This approach enables the spacecraft to autonomously adapt its trajectory in real-time, responding to changes in atmospheric conditions and ensuring that it remains within predetermined safety margins. By leveraging SCP, the vehicle can optimize its entry trajectory to minimize heat loads, reduce fuel consumption, and enhance overall mission reliability.

Table 2: *Initial orbital parameters for the Starship parking orbit*

Parameter	Value
Initial altitude h	$120\ km$
Velocity V_0	$4300\ m/s$
Flight-path angle γ	$-15\ deg$
Angle-of-attack α	$52\ deg$
Spacecraft initial mass m	$100\ tons$
Reference area A_{ref}	$\sim 500\ m^2$
Nose radius n	$4.2\ m$
Orbit eccentricity e	0.515
Semi-major axis a	$\sim 3800\ km$
Apoapsis distance r_a	$\sim 7500\ km$

3.7 Re-entry on Mars at Hypersonic Velocities in an Environment of Atmospheric Dispersions and Other Sources of Error

As mentioned in the introductory section of Chapter 1, successful landings on Mars have been achieved over the past few decades. Before the Mars Science Laboratory (MSL) and Perseverance missions, many of these landings relied heavily on methods inherited from the Viking missions, employing a ballistic entry approach devoid of active guidance. While this approach proved sufficient for achieving safe landings, it fell short of enabling precise landings. The accuracy of planetary landings is often described using a "landing ellipse", which represents the region within which the spacecraft is expected to touch down, defined in an inertial reference system. Achieving pinpoint landing accuracy does not require starting from scratch but instead demands revolutionary advancements in key technologies. One critical aspect is the ability to respond dynamically to variations in trajectory caused by environmental uncertainties, commonly referred to as dispersions. *Figure* ***11*** provides an insightful overview of various sources of error contributing to these dispersions, which will be briefly discussed here. For instance, winds present only a mild uncertainty factor due to Mars's relatively thin atmosphere and limited wind potency. Consequently, winds contribute minimally to trajectory dispersion, resulting in a small landing ellipse (as shown by the light blue notation in *Figure* ***11***).

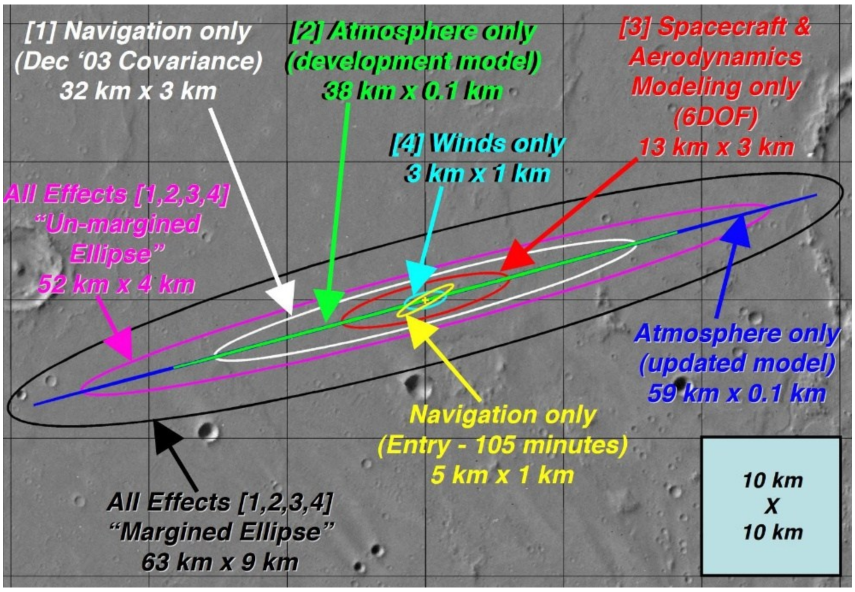


Figure 11: *Sources of errors (perturbations), contributing to landing dispersions* (Soumyo Sutta, May 2024)

The five factors (in *Figure **11***), that have influence on the spacecraft landing ellipse in terms of relative size, based on a 99% likelihood and a 3σ two-dimensional normal distribution, are:

- [1] robustness of real-time guidance algorithms
- [2] navigation uncertainties
- [3] variations in atmospheric properties and [4] winds
- [5] aerodynamic modelling inaccuracies

By tackling factor [1], a spacecraft is able to react faster and better to the rest, hence reduce its landing ellipse on the surface. *Figure **1*** exhibits how cumulative advancements across these domains translate into progressively smaller landing ellipses for actual missions. For instance, improvements in atmospheric modeling directly address one of the largest contributors to dispersion seen in *Figure **11***, enabling tighter control over entry trajectories.

For Starship's reentry on Mars these insights are particularly relevant. Unlike earlier missions that relied on ballistic or semi-guided entries, Starship's approach leverages advanced real-time algorithms such as sequential convex programming for the hypersonic portion of the descent guidance. This allows continuous optimization of control inputs during the descent, dynamically compensating for dispersions caused by navigation errors (set to be ideal in this research), atmospheric variability, aerodynamic uncertainties, and wind effects.

Starship's entry strategy also benefits from its ability to enter from an elliptical orbit rather than a direct interplanetary trajectory. This reduces entry velocity, significantly mitigating thermal loads and aerodynamic stresses while enhancing control authority during descent. The lower velocity regime allows for more precise trajectory adjustments using aerodynamic surfaces such as its fins, reducing sensitivity to environmental uncertainties.

By incorporating lessons learned from past missions while introducing revolutionary advancements in real-time guidance algorithms and entry strategies, Starship has the potential to achieve unprecedented precision in Mars landings. The interplay between reducing individual sources of error (as highlighted in *Figure **11***) and leveraging advanced guidance techniques will enable Starship to achieve landing ellipses comparable to or smaller than those demonstrated by Perseverance (in *Figure **1***) — despite its significantly larger mass and more complex mission profile.

4

Problem Formulation

The flight of re-entry vehicles through alien planets' atmospheres can be mathematically described by Newton's equation of motion, and the same hold true for celestial bodies such as planets, moons, comets, asteroids, etc. Newton's EoM presents the rate of change in the linear momentum, which is equivalent to all other forces acting on the vehicle. Newton's equation yields a three-dimensional $(3D)$ vector equation governing the **translational dynamics** of the re-entry spacecraft modeled as a point mass. This formulation accounts for forces acting along the axes of a chosen coordinate system, and in itself is considered a $3DOF$ simulation. However, for more realistic scenarios, such as the one described in this Starship research, involving **rigid-body dynamics** – where the vehicle rotates around its center of mass (CoM) due to external torques – a complementary vector equation derived from **Euler's angular momentum principle** is required to describe rotational motion.

External influences such as gravitational fields, aerodynamic loads (generated by control surfaces and vehicle geometry), propulsion thrust, and reaction control systems contribute to both forces and moments. To accurately model these interactions, the translational and rotational dynamic equations are usually expressed in **body–fixed coordinates** (aligned with the vehicle's principal axes). However, in Sections 4.1 and 4.2 of this Chapter, a description of the coordinate systems and the resulting simulation degrees-of-freedom will be presented, in order to describe the translational vehicle movement in

spherical coordinates, and the rotational dynamics in **body – fixed coordinates** or the so-called **flight-path – fixed coordinates**, shown in *Figure* ***12***.

This coordinate separation simplifies the equations by minimizing cross – coupling terms – translational motion is decoupled from rotational inertia variations. For instance, body – fixed coordinates leverage the spacecraft's symmetry, ensuring the inertia matrix remains constant during simulations. Such strategic coordinate choices are standard in aerospace engineering, enabling tractable solutions to coupled six-degree-of-freedom $(6DOF)$ problems.

4.1 Equations of Motion (EoMs)

When a planar flight is considered, then the lift force L lies in the $R - V$ plane (*Figure* ***12***) but for flight control and guidance algorithms, L is rotated out of the $R - V$ plane by a bank angle σ. The directions of the lift components $L \cos \sigma$ and $L \sin \sigma$ coincide with the x_k and z_k coordinates of the $\boldsymbol{k}$ – frame, the vehicle's flight – path frame. In this case, the lift vector now can be transformed into:

$$L_{|k} = L \begin{pmatrix} \cos \sigma \\ 0 \\ \sin \sigma \end{pmatrix} \tag{19}$$

and transforms to the $\boldsymbol{g}$ – frame (geodetic frame) by the relation (Weiland, 2010):

$$L_{|g} = M_{kg}^{\chi} M_{kg}^{\gamma} L_{|k} = L \begin{pmatrix} \cos \gamma \cos \sigma \\ - \sin \gamma \cos \chi \cos \sigma - \sin \chi \sin \sigma \\ - \sin \gamma \sin \chi \cos \sigma + \cos \chi \sin \sigma \end{pmatrix} \tag{20}$$

therefore, the equations of motion for the vehicle's velocity, flight – path angle and heading angle are *Equations (**24**) – (**26**)*.

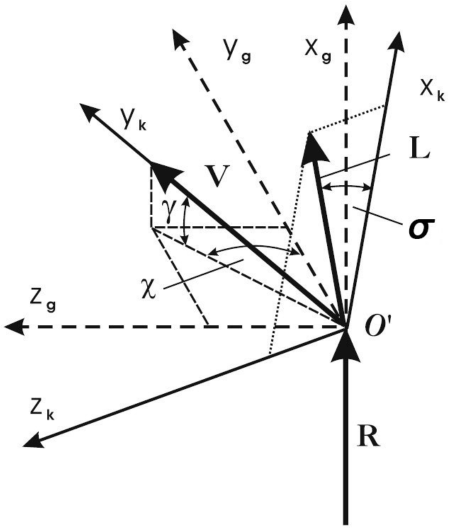


Figure 12: *Coordinate system* $\boldsymbol{O'}(x_g, y_g, z_g)$ (geodetic) *and coordinate system* $\boldsymbol{O'}(x_k, y_k, z_k)$ (flight – path) *with the definition of the bank angle* σ *and lift* L *outside of the* $R - V$ *plane* (Weiland, 2010), p.102

As long as the lift vector L lies in the $R - V$ plane, the flight of the Starship vehicle takes place in a single plane. When rotating the lift vector by a bank angle $\sigma > 0$ the component of the lift, which points out of the $R - V$ plane, generates a force that makes the re-entry spacecraft to abandon the entry plane. For completion of the set of equations, it is necessary to obtain the kinematic equations formulated with respect to the definition of the coordinate directions shown in *Figure* ***12*** *– Equation (**21**) – (**23**)*.

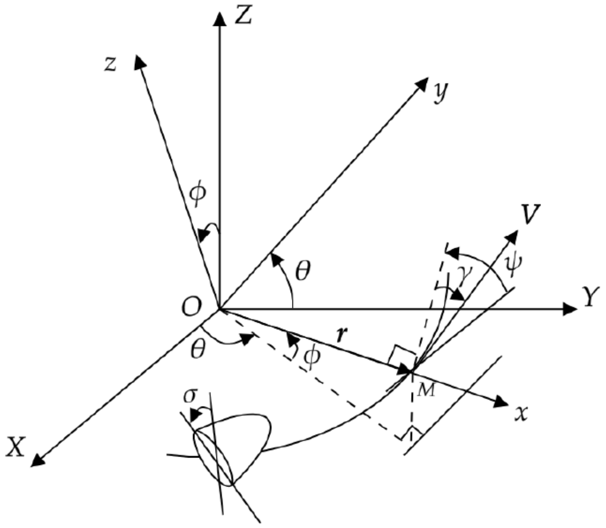


Figure 13: *Three-dimensional (3D) model of a re-entry vehicle* (Pei, 2021)

Figure ***13*** illustrates the reentry vehicle around a rotating Mars. The equations of motion (EoMs) that consider the rotation of Mars can be expressed as follows (Pei, 2021):

$$\dot{r} = V \sin\gamma \tag{21}$$

$$\dot{\theta} = V \cos\gamma \cos\psi/(r \cos\phi) \tag{22}$$

$$\dot{\phi} = V \cos\gamma \sin\psi/r \tag{23}$$

$$\dot{V} = -D/m - g_0 \sin\gamma/r^2 + C_v \tag{24}$$

$$\dot{\gamma} = L \cos\sigma/mV + (V^2 - g_0/r) \cos\gamma/Vr + C_\gamma \tag{25}$$

$$\dot{\psi} = L \sin\sigma/(mV \cos\gamma) + V \cos\gamma \sin\psi \tan\phi/r + C_\psi \tag{26}$$

where r is the radial distance from the center of Mars to the spacecraft, normalized by the Mars' radius $R_0 = 3389.5\, km$. The symbols θ and ϕ are the longitude and latitude, respectively. ψ and γ are the heading angle and the flight-path angle, respectively. V is the dimensionless velocity normalized by $\sqrt{R_0 g_0}$, $g_0 = 3.72\ m/s^2$ is the gravity acceleration. The time is normalized by $\sqrt{R_0/g_0}$. The dimensionless lift and drag accelerations, respectively denoted with L and D, are normalized by g_0 and calculated such as (Liu, 2019):

$$L = \frac{R_0 \rho V^2 A_{ref} C_L}{2} \tag{27}$$

$$D = \frac{R_0 \rho V^2 A_{ref} C_D}{2} \tag{28}$$

where R_0 and A_{ref} are the Martian radius and reference area of the vehicle, respectively (Pei, 2021). $\rho = \rho_0 e^{-h/h_s}$ is the dimensional atmospheric density, where $\rho_0 = 0.02\ kg/m^3$ and $h_s = 11100\, m$, h is the dimensional altitude, and $h = rR_0 - R_0$. C_L and C_D denote the aerodynamic lift and drag coefficients, respectively. The primary control variable of the re-entry trajectory optimization problem is the bank angle σ.

Also, C_v, C_γ and C_ψ in *Equations* ***(29)***, ***(30)*** and ***(31)***, respectively, are the planet's rotations which are derived and calculated as follows:

$$C_v = \omega^2 r \cos\phi\, (\sin\gamma \cos\phi - \cos\gamma \sin\phi \sin\psi) \tag{29}$$

$$C_\gamma = 2\omega \cos\psi \cos\phi + \left(\frac{\omega^2 r}{V}\right) \cos\phi\, (\cos\gamma \cos\phi + \sin\gamma \sin\phi \sin\psi) \tag{30}$$

$$C_\psi = 2\omega(\tan\gamma \cos\phi \sin\psi - \sin\phi) - \left(\frac{\omega^2 r}{V \cos\gamma}\right) \sin\phi \cos\phi \cos\psi \tag{31}$$

where $\omega = 7.088\ (10^5)$ is Mars' rotation rate.

4.2 6DoF Simulation

The number of degrees of freedom in a dynamic system refers to the minimum number of independent coordinates required to completely specify the configuration of the system at any given point in time during the simulation. In simple terms, it is the number of ways the object can move and orient itself in space.

As already described in the previous Section 4.1, *Equations* ***(21)*** *–* ***(23)*** control the rate of change in radial distance, longitude and latitude, and thus describe the **translational motion** of the vehicle (position of its center of mass (COM) in spherical space. Additionally, the **rotational orientation** of the vehicle is implicitly described through the rate of change in the flight-path and heading angles (through *Equations* ***(25)*** and ***(26)***), and the magnitude of the velocity vector itself (*Equation* ***(24)***). Having control over these 6 state variables during the simulation, makes this a $6DoF$ problem. Even though it is not the traditional $6DoF$ description of vehicle motion in space, that has 3 Cartesian coordinates and another 3 Euler angles, the orientation of Starship (which dictates how aerodynamic forces and moments act on it) is implicitly captured in how these velocity vector angles (the flight-path and heading angles) evolve under the influence of the two controls available (angle-of-attack and bank angle modulation). To clarify further, while this choice of EOMs does not directly give the Euler angles (roll, pitch and yaw) of the body frame, the *change* in flight-path and heading angles, driven by the aerodynamic forces and torques (which depend on Starship's attitude relative to the flow), fully accounts for the vehicle's evolving orientation in the context of its trajectory. This can be clearly understood by following the schematic, presented in *Figure* ***14***, in the rotational dynamics section.

As shown in *Figure* ***14***, Left, the translational dynamics, represented by *Equations* ***(21)*** *–* ***(23)***, are in Mars-centered inertial reference frame, and described in spherical coordinates. This is convenient and typical for such trajectory optimization problems because it is easier to understand objective-wise, instead of using pure Cartesian space (x, y, z) coordinates. While this is the easier part to understand, when it comes to rotational dynamics, shown in *Figure* ***14***, Right, and governed by *Equations* ***(24)*** *-* ***(26)***, things are not so straight-forward. While the listed EOMs do not directly include variables in the body frame, the controls α and σ are defined relative to the body-fixed / flight-path frame, and the velocity vector in *Equation*

*(**24**)*. These controls determine the aerodynamic forces acting on Starship in *the body frame*, and are then translated back to influence the motion described in the inertial frame's coordinates – shown in *Figure **14***, Bottom Right. In essence, the **inertial frame** provides the global context for the trajectory simulation (position) relative to Mars, while the **body-fixed** (also called flight-path) frame is where the controls are naturally defined. The AoA (the angle between the vehicle's x-axis and the velocity vector), and the bank angle (the vehicle's rotation about its velocity vector), both directly influence the Lift and Drag forces. Consequently, the aerodynamic forces (which depend on attitude) are used to directly calculate the accelerations that change the magnitude and direction of the velocity vector. The simulation algorithm handles the coupling between commanded controls, the resulting aerodynamic forces, and the subsequent change in flight-path (γ) and heading (ψ) angles.

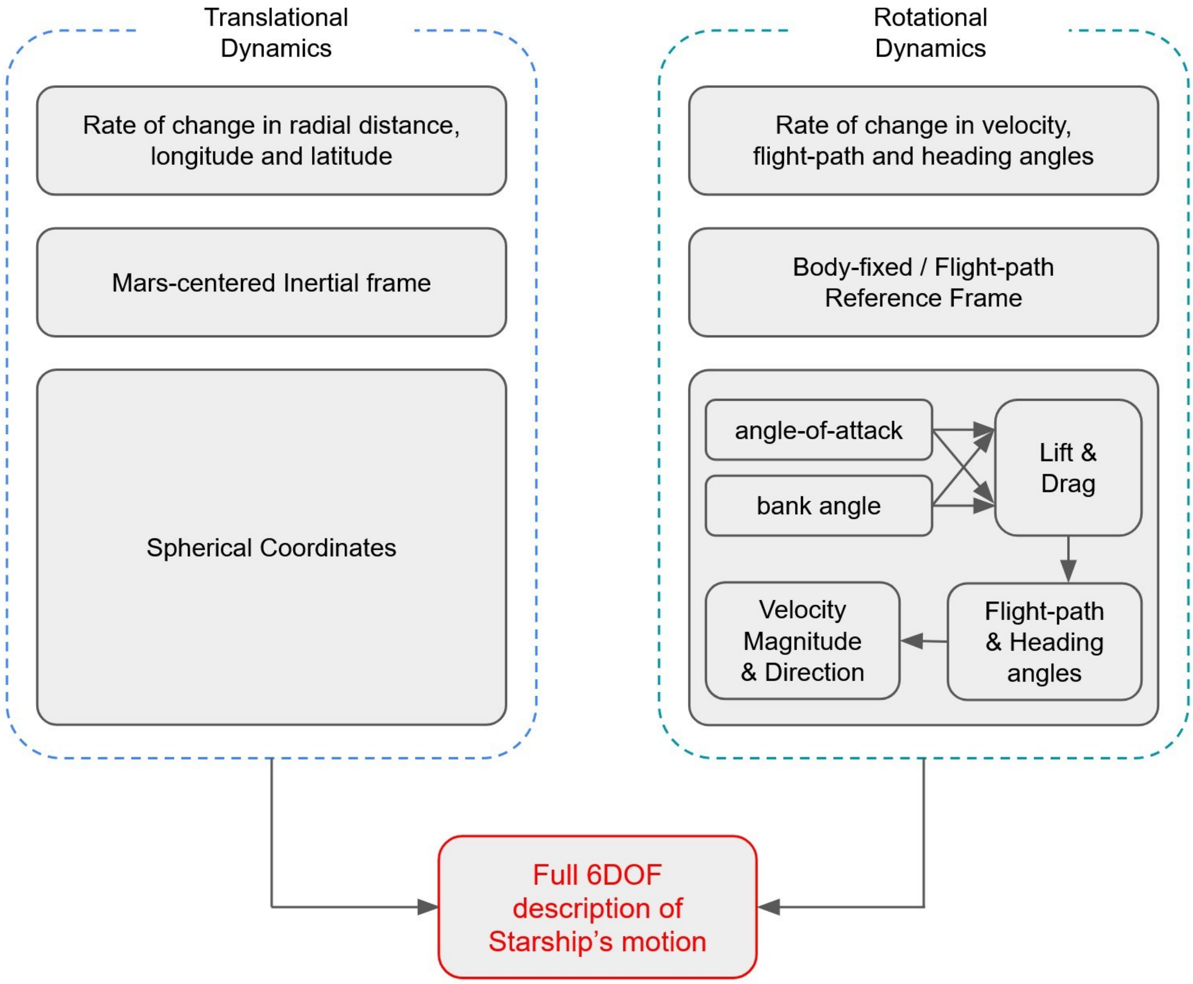


Figure 14: *Schematic description of Starship's $6DoF$ translational and rotational dynamics during Mars atmospheric reentry* (produced in Google Slides by the author of this research)

4.2 Constraints

In this research, the initial and terminal trajectory profiles are presented as equality path constraints (Meeowen, 2022):

$$\boldsymbol{x}(t_0) = \boldsymbol{x}_0;\ \boldsymbol{x}(t_f) = \boldsymbol{x}_f \tag{32}$$

where $x = [r\ \theta\ \phi\ V\ \gamma\ \psi]^T$ is the state vector, and t_0 and t_f are the initial and terminal states, respectively.

The hypersonic glide trajectory will also obey certain re-entry trajectory inequality path constraints that determine the re-entry corridor – those are 1) heat flux $\dot{Q}$, 2) normal load n, and 3) dynamic pressure q (Pei, 2021).

$$\dot{Q} = \frac{k_Q}{\sqrt{R_n}}\sqrt{\rho}V^{3.15} \leq \dot{Q}_{max} \tag{33}$$

$$n = \sqrt{L^2 + D^2} \leq n_{max} \tag{34}$$

$$q = \frac{1}{2} g_0 R_0 \rho V^2 \leq q_{max} \tag{35}$$

where $\dot{Q}_{max}$ is the dimensional maximum heat flux in kW/m^2, n is the maximum normal load in g_0, and q_{max} is the dimensional maximum dynamic pressure in N/m^2. $k_Q = 1.05 \times 10^{-4} \times \left(\sqrt{g_0 R_0}\right)^{3.15}$ is the heat – flux constant (Pei, 2021). The variable $R_n = 4.2\ m$ is Starship's radius of the stagnation point at its nose, which is consistent with what is presented in (Weiland, 2010), p.141. To account for unforeseen circumstances and, in general, to have the ability to correct the trajectory, a no-fly zone (NFZ) can be modeled as a cylinder with infinite height (as shown in *Figure **15***). Then, the NFZ constraint limits (corrects) the longitude and latitude in the state variable, such as:

$$(\theta - \theta_c)^2 + (\phi - \phi_c)^2 \geq d^2 \tag{36}$$

where θ_c is the longitude center of the NFZ, ϕ_c is the latitude center, and d is the radius of the NFZ in radian (Kamath, 2022).

Additionally, the inequality constraints of the optimization problem include the limits on state and control variables (Kamath, 2022):

$$\mathbf{x}_{min} \leq \mathbf{x} \leq \mathbf{x}_{max};\ \sigma_{min} \leq \sigma \leq \sigma_{max} \tag{37}$$

where $\mathbf{x}_{min}$ and $\mathbf{x}_{max}$ are the lower and upper bounds of the state variables, and σ_{min} and σ_{max} are the lower and upper bounds on the bank angle control.

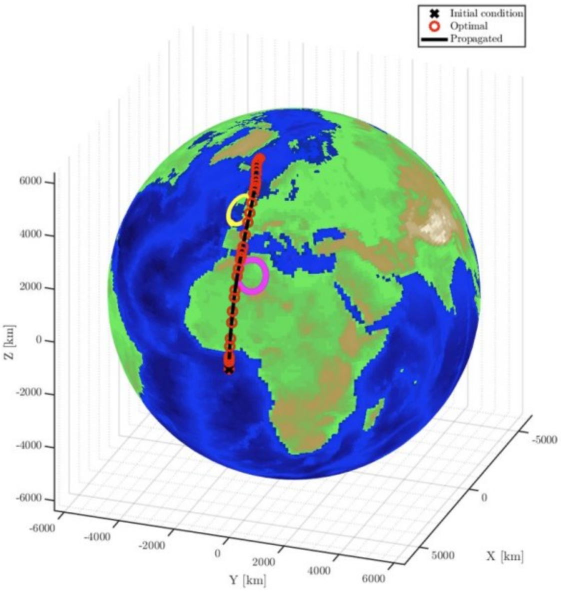


Figure 15: *A simulated trajectory with two no-fly zones (NFZs) implemented* (Kamath, 2022)

To summarize, the re-entry trajectory optimization problem can be described as:

$$\boldsymbol{P0}:\ \ \text{Minimize}\ J = \varphi\left[\mathbf{x}\left(t_f\right)\right] + \int_{t_0}^{t_f} F(\mathbf{x}, \sigma)dt \tag{38}$$

Subject to *Equations (**21**) – (**36**)*

We have formulated the re-entry trajectory optimization problem, which is inherently highly nonlinear and nonconvex. This high nonlinearity comes from:

1. the nonlinear and coupled differential *Equations (**21**) – (**26**)*
2. the vehicle's time-sensitive and coupled aerodynamic forces *Equations (**27**) – (**28**)*
3. the nonlinear entry corridor constraints and NFZ constraint *Equations (**33**) – (**36**)*

These nonlinearities need to be convexified to be able to use Sequential Convex Programming (SCP) to solve the original nonconvex trajectory problem (Acikmese, 2022).

4.3 Precision in the Void: Discretization for Hypersonic Mastery

Guiding Starship through the Martian atmosphere demands computational precision, at the core of which lies direct collocation discretization. In simple terms, this method transforms the continuous time frame during the entry into a sequence of discrete points where a finite sequence of solvable equations can be solved.

Unlike **single shooting –** which relies on explicit integration and struggles with unstable systems or poor initial guesses – direct collocation implicitly satisfies dynamics through algebraic constraints, enhancing robustness for problems with complex path constraints like heat flux or aerodynamic loads. Compared to **multiple shooting**, which partitions trajectories into segments to improve convergence, direct collocation produces sparser nonlinear programs (NLPs) by leveraging polynomial continuity, reducing computational overhead. However, its fixed time grid and low-order splines (e.g., trapezoidal or Hermite-Simpson) introduce truncation errors, making it less accurate than **orthogonal collocation** – a high-order variant ideal for smooth solutions but less effective for discontinuous constraints. While shooting methods excel in accuracy with adaptive integrators, they falter under stringent path constraints, whereas direct collocation trades precision for computational efficiency and constraint-handling reliability, making it indispensable for real-time applications like Starship's Martian entry.

The Discretization Framework

The MATLAB implementation employs a **fixed-time grid direct collocation** approach, discretizing the continuous trajectory into $N = 400$ nodes – a balance between resolution and computational tractability. Each node captures the spacecraft's state $\mathbf{x} = [r, \theta, \phi, V, \gamma, \psi\,]$ and control $\mathbf{u} = \sigma$ (bank angle) and α (angle-of-attack), governed by:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, t) \tag{39}$$

where f encodes hypersonic aerodynamics, gravitational forces, and planetary rotation rate.

4.3.1 *Key Components*

1. Uniform Time Grid

- States are linearly interpolated between initial and terminal conditions (e.g., interplanetary entry to parachute deployment in the dense parts of the atmosphere).
- Controls are piecewise – constant or linearly varying, avoiding high – frequency oscillations.

2. Fourth – Order Runge – Kutta $(RK4)$ Integration

- Dynamics are propagated between nodes using:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{40}$$

where k_i are derivative evaluations at intermediate steps. This ensures $O(h^4)$ local truncation error, critical for capturing rapid heating transients (Wikipedia, n.d.).

3. Successive Convexification

- Non-convex path constraints (e.g., $\dot{q} \leq 120\ kW/m^2$) are linearized iteratively.
- Trust regions, enforced via $l_2 -$ form bounds, stabilize convergence amid Mars' atmospheric unpredictability.

Starship's hypersonic re-entry is the discretized entry profile – a hypersonic glide transitioning to a propulsive landing – a testament to this discretization's capabilities. In **Phase 1 – Atmospheric negotiation**, Starship pierces Mars' atmosphere with an orbital velocity of around $4.3\ km/s$, its stainless – steel airframe sustains radiative heating while bank angle modulations σ trade altitude for velocity. The discretized model encodes the heat flux as a convex SOC constraint, linearizes lift – drag coupling, ensuring computational feasibility without sacrificing fidelity. In **Phase 2 – Supersonic Pivot**, at $\sim 500\ m$ Starship executes its 90-degree belly – flop to vertical – a maneuver demanding millisecond – level control updates. In this research, only Phase 1 is examined.

By discretizing time into manageable time points on the trajectory, Starship transforms Mars' "seven minutes of terror" into a manageable descent with a reasonable accuracy as a result (Thilbault, 2019).

4.4 Simulation Model for Hypersonic Re-entry

As already discussed previously, guiding a spacecraft through the atmosphere must take into account different aspects of the environment. In this section, the different aspects of the environment model will be stated.

Atmospheric Model

The chosen framework for the simulation in this research adopts a static exponential atmospheric density profile, derived from Mars Global Surveyor data, as follows:

$$\rho(h) = \rho_0^{-\frac{h}{H}} \tag{41}$$

where $\rho_0 = 0.02\ kg/m^3$ (surface density) - ~ 60 times lower than what we experience on the surface of our planet Earth, and $H = 11.1\ km$ (scale height) – the height necessary to decrease the density with an order of exponent. This model neglects seasonal and diurnal variations, as well as dust storms, which can elevate density by 50% regionally. The implications of this are exhibited in the hypersonic deceleration, where the exponential decay creates a sharp velocity – altitude trade-off. Starship's bank angle maneuvers must exploit this gradient to shed $\sim 4.3\ km/s$ velocity within less than 2 minutes, avoiding undershooting (excessive heating) or overshooting (escape trajectories). The dynamic pressure limitation $q = \frac{1}{2}\rho V^2$ constraint becomes a non-convex barrier, linearized via successive convexification. Stochastic dust effects are being ignored, creating a realism gap, but the model prioritizes determinism over adaptability – a calculated risk for preliminary design. Also, referring to compromises taken, the chosen atmospheric model is not a high-fidelity model incorporating turbulence and dust, used in NASA's Monte Carlo analyses. Its omission here reflects a trade-off: 80% fewer computational cycles at the cost of localized trajectory errors, accounting $\leq$ 2% of altitude deviation.

Gravitational Model

The chosen framework here is considering planet Mars as a point of mass, and, as a consequence, the two-body inverse-square law with Mars' gravitational parameter $\mu = 4.283 \times 10^{13}\ m^3/s^2$, neglects zonal J_2 harmonics and topological anomalies. The implications of this, when it comes to trajectory simplification, is that the $\sim 0.1\%$ error from omitting J_2 (oblateness) is tolerable for entry – phase simulations ($\leq 500\ km$ altitude) but accumulates over orbital phases. Starship's propulsive finale with a spherical gravity model introduces $\sim 10\ m$ better positional error (compared to point mass gravity) – a margin absorbed by its Terrain – Relative Navigation (TRN) system. (Starship SpaceX Wiki) But as far as the hypersonic glide part is concerned, the positional error is negligible.

Thermal Model

The chosen framework here is a semi – empirical heat flux equation from MSL data:

$$\dot{q} = k\sqrt{\rho}V^3 \tag{42}$$

where $k = 1.1 \times 10^{-4} \frac{Ws}{kg\ m}$ and radiation effects are ignored, assuming ablative shielding dominates cooling. The implications of this conservative design are prioritizing worst – case heating scenarios, ensuring thermal protection systems (TPS) are over – engineered by $\sim 15\%$. Starship's stainless – steel and ceramic tiles TPS, rated for $\geq 1200\ K$, relies on this model's accuracy to avoid buckling during $\sim 45\ s$ of peak heating.

Aerodynamic Model

The chosen framework for hypersonic realism here is the $C_L \propto \sin^2 \alpha$ trend that captures non – linear shock interactions, critical for modulating lift-to-drag ratios between $\frac{L}{D} \approx 0.3 - 1.2$ during glide.

Planetary Rotation

A non – rotating framework is chosen here, ignoring Mars' $0.24\, m/s$ equatorial velocity, which is $\sim 5\%$ of entry speed. The implications of this are a trajectory bias which means that cross – range errors accumulate at $\sim 500\ m/min$ of entry: a tolerable offset given Starship's $1\ km$ landing ellipse. Regarding energy management, the Coriolis effect's absence simplifies energy dissipation calculations, under – predicting required bank adjustments by $\sim 3^0$.

The simulation's environmental choices reflect a philosophy of **strategic simplification** – sacrificing esoteric details (dust storms and J_2) to prioritize real-time solvability. For Starship, this model is a reasonable compromise as it enables onboard trajectory regeneration but masks risks like localized density spikes or unanticipated boundary layer transitions (Ingo Gerth, 2014).

5

Algorithm Methodology

In this chapter, the methodology, implemented in the MATLAB code used for the trajectory optimization simulation, will be discussed in detail. But first, a context will be provided in Section 5.1 to why and how Mars re-entry guidance systems evolved with time and why there is/was a need to improve accuracy with subsequent missions.

5.1 Advancing Planetary Descent: From Polynomial to PTR Guidance

In this Hypersonic Entry Glide Guidance (HEGG) represents the critical nexus between atmospheric entry and controlled touchdown, making for a spacecraft's transition from the rarefied upper atmosphere to the planetary surface. During this phase, the vehicle employs its reaction control system (RCS) – integrating aerodynamic actuators and, when necessary, supplementary thrusters – to navigate from the initial atmospheric interface to a gentle surface contact. This hypersonic glide, and subsequent powered descent, forms the cornerstone of safe extraterrestrial landings, demanding solutions that harmonize precision with adaptability. While space launch costs have decreased from over $\$10{,}000/kg$ during the Space Shuttle era to $\$3{,}000/kg$ via innovations like SpaceX's Falcon 9, the pursuit of efficiency remains paramount _(Vought, 2019). Starship's paradigm-shifting reusability and advanced aerodynamic control surfaces – forward and aft flaps refined through iterative testing – exemplify technologies poised to further reduce mass and cost. Contemporary HEGG systems optimize trajectories through synergistic use of aerodynamic shaping and propellant minimization, building upon legacy approaches like the Apollo program's polynomial guidance, which achieved remarkable fuel efficiency within $16\ kg$ of theoretical minima (Taylor Reynolds D. M., 2020).

The next frontier demands HEGG algorithms capable of reconciling Martian atmospheric unpredictability with millimeter-level precision. Future systems must govern the 6-degree-of-freedom $(6DoF)$ dynamics of rigid-body rotation and translation (which are implemented in this research and described in the report, Section 4.2) while accommodating continuous and discrete constraints imposed by TRN architectures. For vehicles like Starship, this entails navigating volcanic fissures and ice-laden landscapes; for crewed missions, it requires pinpoint accuracy to establish habitats near pre-deployed infrastructure. As already described previously, modern guidance frameworks leverage convex optimization to reduce propellant consumption and increase accuracy, while real-time trajectory recalibration ensures resilience against atmospheric dispersions. Autonomous operation is non-negotiable – algorithms must exhibit deterministic convergence on computationally constrained hardware, enabling both robotic and human missions to operate independently of Earth-based oversight.

The efforts in real-time constrained $6DoF$ descent guidance (DG) algorithms trace their origins to the Apollo era, when Meditch pioneered the analytical solution to a fuel-optimal $1DoF$ vertical descent – a seminal breakthrough that laid the groundwork for modern planetary landing systems _(Meditch, 1964). Concurrently, Lawden's formulation of optimality conditions for a generalized fuel-optimal problems, incorporating mass and translational dynamics, expanded the theoretical framework _(Lawden, London, Butterworths, 1963). Yet, the computational demands of solving these conditions – via shooting methods or nonlinear programming – exceeded the capabilities of mid-20th-century technology. Decades later, D'Souza derived closed-form solutions for mixed minimum-energy and minimum-time $3DoF$ problems _(D'Souza, New Orleans, LA, 1997), while _(P. Lu, Minneapolis, MN, 2012) advanced real-time algorithms using nonlinear root-finding techniques. However, these methods remained constrained by their inability to enforce mission-critical state constraints, such as approach angles or sensor alignment requirements, which are indispensable for modern precision landing missions.

The advent of interior-point methods in numerical optimization revolutionized trajectory planning by offering polynomial-time (already explained in Section 3.3) solutions with guaranteed global convergence – properties unattainable with classical shooting or nonlinear programming approaches. Acikmese and Ploen’s landmark work on *lossless convexification* marked a paradigm shift, transforming the fuel-optimal $3DoF$ problem with thrust bounds and approach angle constraints into a convex optimization framework solvable via Second-order cone programming (SOCP) (Behcet Acikmese L. B., 2007). This methodology, compatible with onboard flight computers, was later extended to address non-convex input sets, minimum-error landing constraints, quadratic state limits, and even hybrid dynamics involving binary variables – all without resorting to mixed-integer programming. The robustness of SOCP solvers, capable of static memory allocation and rapid computation, ensured that feasible solutions existed within predefined convex regions calculated during mission design.

From $3DoF$ to $6DoF$

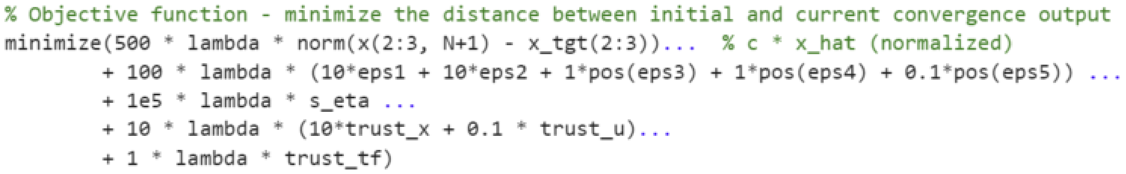
Despite its robustness, $3DoF$ guidance suffers from intrinsic limitations: it neglects attitude dynamics and assumes instantaneous inner-loop control, rendering it incapable of enforcing sensor line-of-sight constraints (real-world navigation) or validating trajectories against the rotational inertia of a physical lander. For missions requiring tight coupling between guidance and navigation – such as hazard avoidance during Mars descents or Europa’s fissure-targeted landings – $3DoF$ solutions demand exhaustive simulation campaigns to verify executability, a process fraught with edge-case vulnerabilities.

The transition to $6DoF$ guidance emerged as an imperative, driven by the need to harmonize translational and rotational dynamics while preserving convex optimization’s guarantees. Sequential Convex Programming (SCP), a trust-region variant, bridged this gap by iteratively convexifying non-convex dynamics and constraints. Building on SOCP solvers developed for $3DoF$ problems, SCP introduced capabilities to model aerodynamic L/D, and state-triggered constraints (e.g., velocity-dependent angle-of-attack limits or slant-rangeactivated line-of-sight requirements). Recent advancements, such as those by (Michael Szmuk, 2019), demonstrate SCP's real-time proficiency, solving generalized $6DoF$ problems with execution times under 0.7 seconds (Cui, 2017) – a feat critical for autonomous systems like NASA's Terrain Relative Navigation (TRN), which guided Perseverance to a 5-meter landing accuracy during its final descent in Jezero Crater.

Modern SCP frameworks address compound constraints through *state-triggered logic*, enabling discrete decisions (e.g., activating hazard-avoidance maneuvers only within specific altitude bands) without sacrificing convexity. This innovation, coupled with aerodynamic modeling validated in Mars 2020's Entry, Descent, and Landing (EDL) phase, ensures that $6DoF$ trajectories respect both physical dynamics and sensor-driven operational limits. The integration of free-final-time formulations further optimizes propellant use, reducing mass fractions while accommodating real-time trajectory adjustments – a capability demonstrated in JPL's simulations for lunar multi-stage landers trust (Michael Szmuk, 2019).

As planetary scientists target increasingly complex destinations – from the Moon's shadowed poles to Enceladus's cryovolcanic terrain – the fusion of SCP-based $6DoF$ guidance with vision-aided systems like TRN represents the vanguard of autonomous spaceflight. These algorithms not only inherit the computational rigor of convex optimization but also transcend Apollo-era limitations, enabling spacecraft to autonomously navigate environments where "eyes-on" human oversight, as during Apollo 11's landing, is impossible. In doing so, they fulfill the dual mandate of modern exploration: 1) minimizing risk while 2) maximizing scientific return through precision unattainable in the pre-convexification era.

Beyond the already mentioned methods in Chapter 1, *Figure **4***, SCP has been widely applied to aerospace guidance problems, with this research specifically exploring Penalized Trust Region (PTR)'s potential for Hypersonic Entry Glide Guidance (HEGG), especially by incorporating the MATLAB CVX-defined *trust region variables* in the objective, which aims to minimize the distance (the second norm between the longitude and latitude, more specifically) between the current and previous (initial) iteration's result. In Figure 16, the addition of the PTRs ($trust\_x$ and $trust\_y$) in the objective is shown. The convex slack variables ($eps1 - eps5$) are also included in the objective, as they are estimated by the internal optimizer at each iteration. The slack variables provide a small margin around the calculated physical and control constraints.



```
% Objective function - minimize the distance between initial and current convergence output
minimize(500 * lambda * norm(x(2:3, N+1) - x_tgt(2:3))...  % c * x_hat (normalized)
        + 100 * lambda * (10*eps1 + 10*eps2 + 1*pos(eps3) + 1*pos(eps4) + 0.1*pos(eps5)) ...
        + 1e5 * lambda * s_eta ...
        + 10 * lambda * (10*trust_x + 0.1 * trust_u)...
        + 1 * lambda * trust_tf)
```

Figure 16: *The MATLAB code implementation of the objective to minimize the distance between the iterative cycles' results (hence, increased precision). The theoretical SOCP structure can be seen in* ***Problem 2*** *in Section 5.2.1. (500*lambda) is c, and the second norm is z. The structure of z is also described better in Figure* ***18****.*

The PTR method demonstrates exceptional real-time capabilities across diverse applications. _(Michael Szmuk, 2019) showcased its efficiency in solving complex guidance problems. PTR's performance is grounded in solving Second – Order Cone Programming (SOCP) problems efficiently. For example, the G-FOLD algorithm _(Behcet Acikmese S. M., 2017) – based on lossless convexification – successfully computed landing divert trajectories during a three-year flight test campaign aboard the Masten Space Systems Xombie rocket, demonstrating SOCP feasibility on spaceflight processors.

Modern real-time optimization paradigms increasingly rely on customized solvers like CVX (used in this research under MATLAB). This approach is common in Model Predictive Control (MPC), where optimizers are invoked at frequencies matching controller demands. While MPC methods are proposed for $6DoF$ planetary descent guidance (PDG), their reliance on continuous re-solving poses challenges for classical feedback control architectures in space missions and risks constraint violations _(Behcet Acikmese S. M., 2017). To address these limitations while enabling $6DoF$ HEGG capabilities, this research proposes solving the re-entry trajectory optimization from Starship's current state at $120\ km$ down to $45\ km$ altitude. This approach minimizes atmospheric uncertainty – one of the largest sources of perturbations – which can significantly deviate Starship from its target due to small disturbances in the upper atmosphere.

By integrating PTR into this framework, the guidance algorithm ensures trajectory optimization that aligns with existing feedback control architectures while addressing atmospheric variability. The result is a robust system capable of supporting Starship's ambitious missions to Mars and beyond, combining computational efficiency with precision landing capabilities essential for future exploration endeavors.

This chapter explains the Hypersonic Entry Glide Guidance (HEGG) algorithm design. First, it systematically addresses the principal design dimensions of a HEGG framework, including the analytical transformation of complex optimization challenges into a sequence of Second-Order Cone Programming (SOCP) formulations. This process elucidates the nuanced interplay between constraint types, parameter selection, and their collective impact on computational complexity and solution time – a critical consideration for systems requiring rapid decision-making. While simplified in state representation, the modeled dynamics and constraints preserve essential characteristics of $6DoF$ descent guidance (DG) systems, ensuring fidelity to real-world operational scenarios.

The descriptions of the PTR-based $6DoF$ DG algorithm as a paradigm-shifting tool for next-generation entry, descent, and landing systems, prove that it is an appropriate implementation for Starship's re-entry. In this chapter, the algorithm implemented for the Starship re-entry trajectory, will be described with all of its steps, exhibited in *Figure* ***17***. In Section **5.2** a definition of the non-linear trajectory ***Problem 1*** is conducted, in Section **5.2.1** the resulting ***Problem 2***, defined in SOCP-framework after successive convexification, is described. In Section **5.2.2**, the first phase of the optimization algorithm is described – *Initialization* – is described. It includes three main parts: **1)** initial guess, **2)** scaling matrices, and **3)** pre-parsing. In Section **5.2.3**, a description of the two main parts of the second step – *Convexification* – is conducted; **1)** linearization, and **2)** propagation. In Section 5.2.4 – *Solve* – the two main parts are described – **1)** parsing, and **2)** convergence criteria.

5.2 Description and Implementation of the Algorithm

A general methodology for developing an implementation to solve non-convex optimal control problems (OCPs) is presented here. These OCPs are of the form _(Taylor Reynolds D. M., 2020):

***Problem* 1**

$$\min_{u(\cdot),p} \phi\big(x(t_f),p\big)$$

$$\text{subject to:} \qquad \dot{x}(t) = f(x,u,p)$$

$$H_0 x(t_0) = x_{ic}, \quad H_f x\big(t_f\big) = x_f$$

$$g_i(x,u,p) \le 0, \quad i \in I_c \coloneqq \{1, \dots, n_c\}$$

where $x(t) \in R^{n_x}$ is a continuous – time state vector, $u(t) \in R^{n_u}$ is a continuous – time input vector and $p \in R^{n_p}$ is a parameters vector. The functions g_i are scalar – valued and represent the path constraints imposed during the hypersonic glide $g_i: R^{n_x} \times R^{n_u} \times R^{n_p} \rightarrow R$. It is assumed that the constraints indexed by $I_{cvx} \subseteq I_c$ are convex, and the ones indexed $I_{non-cvx} \subseteq I_c$ are non-convex, so that $I_c = I_{cvx} \cup I_{non-cvx}$. With matrices H_0 and H_f the ability to constrain only a subset of the state vector at the initial and final time is signified. It is also assumed that $f: R^{n_x} \times R^{n_u} \times R^{n_p} \rightarrow R^{n_x}$ is differentiable almost everywhere with respect to its arguments. Moreover, it is assumed also that the cost function ϕ is convex, without any loss of generality, and is given in Mayer form using $\phi: R^{n_x} \times R^{n_p} \rightarrow R$. Every optimization problem inherently incorporating a running cost may be elegantly reformulated within this framework, representing the most straightforward method to transform continuous-time problems into the standardized structure required for real-time applications. This transcription process preserves mathematical rigor while ensuring computational tractability for onboard systems. Notably, scenarios requiring temporal flexibility — such as free-final-time formulations—may be seamlessly integrated into the ***Problem 1*** paradigm through strategic utilization of the parameter vector p. This parameterization elegantly encapsulates temporal variability, enabling dynamic adaptation to mission-specific time horizons without compromising the problem's convex structure. Such an approach not only simplifies implementation but also enhances compatibility with modern convex solvers, ensuring rapid convergence even under stringent real-time constraints.

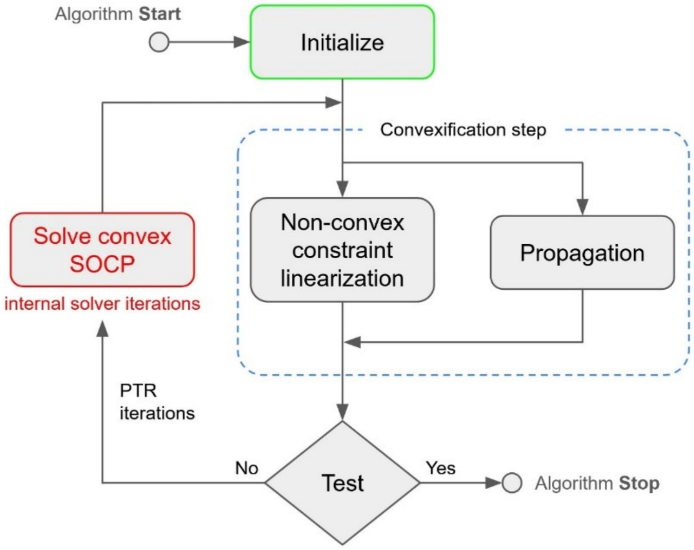


Figure 17: *Diagram of the 3 major steps in the PTR algorithm, implemented here. The most consuming step (highlighted in red) is solving the convex SOCP problems.* (Google Slides)

5.2.1 Successive Convexification

In the exploration of real-time solutions to ***Problem 1***, an overview of the Penalized Trust Region (PTR) algorithm, more broadly known as successive convexification, is conducted. This approach, illustrated in *Figure* ***17***, unfolds in a series of steps. The algorithm commences with an initialization phase (in green), where an initial solution guess – potentially rudimentary – is defined. This phase also encompasses the task of setting scaling matrices to enhance the numerical properties of the optimization process, as well as pre-parsing fixed elements of the problem to streamline subsequent computations.

Following initialization, the algorithm enters the main PTR loop, where the initial guess assumes the role of the first reference solution. Throughout this discourse, two types of iterations are being distinguished: "solver" iterations, which occur internally within the Second-Order Cone Programming solver (in red), and "PTR" iterations, which constitute the outer loop defined by the convexification (blue), test, and solve (red) steps.

The convexification step (in blue), a cornerstone of the algorithm, yields a convex approximation of ***Problem 1*** in the neighborhood of the current reference solution. This process involves approximating the dynamics as a discrete linear time-varying system and linearizing non-convex constraints about the reference solution. The resulting convex approximation serves a dual purpose: 1) it allows us to assess whether the current reference solution satisfies the desired feasibility and convergence criteria through a stopping criterion, and if not, 2) it forms the basis for formulating an SOCP in a general conic form (in red).

This iterative refinement process balances computational efficiency with solution accuracy, making PTR a powerful tool for real-time trajectory optimization in re-entry applications. The algorithm's ability to handle complex, non-convex problems through successive convex approximations represents a significant advancement in the field of optimal control, particularly for systems with stringent real-time requirements such as hypersonic entry guidance.

The conic form of the SOCP is as follows (Taylor Reynolds D. M., 2020):

Problem 2

$$\min_{z} c^T z$$

$$\text{subject to:} \quad Az = b$$

$$z \in C_L \times C_{Q_1} \times \ldots \times C_{Q_m}$$

where $C_L = \{w \in R^l | \, w \geq 0\}$ is a linear cone of dimension l, and each $C_{Q_i} = \{(w_0, w) \in R^{d_i} | \, \llbracket w \rrbracket_2 \leq w_0\}$ is a second order cone of dimension d_i. Afterwards, the problem data $A \in R^{n_c \times n_z}$, $b \in R^{n_c}$ and $c \in R^{n_z}$ are being passed to the solver to compute a new reference solution. A structural translation is paramount when formulating optimization problems within the framework of ***Problem 2***. Traditional approaches to successive convexification, and sequential convex programming at large, have long depended on automated modeling interfaces (commonly termed "parsers") to facilitate this conversion. While these tools offer convenience, their implementation introduces three critical inefficiencies: computational overhead from redundant constraint generation, superfluous variables that inflate problem dimensionality, and reliance on dynamic memory allocation – attributes fundamentally incompatible with the stringent demands of real-time spaceflight systems.

This thesis introduces a methodology that circumvents these limitations through deliberate "hand parsing," and leverages domain-specific knowledge to exploit latent problem structures.

5.2.2 Initialization

This section outlines the preparatory steps taken before initiating the primary PTR loop. These steps, as outlined in **Algorithm 1**, involve three key tasks: creating an initial solution estimate, calculating scaling matrices, and pre-processing problem data. All these tasks can be completed prior to starting the main PTR loop depicted in *Figure **17***. If the initialization process occurs at a time t before t_o, where t_o marks the start of executing the guidance solution by the vehicle, the initial state x_{ic} must be projected forward by $t_0 - t$. This time difference $t_0 - t$ should cover at least the duration required to compute the guidance solution. In the case of test flights using the $3DoF$ G-FOLD algorithm, this duration was set to one second.

Algorithm 1 Initialization Step

Input: Current vehicle state and desired terminal boundary conditions
1) Determine the initial guess for the state, controls and parameters
2) Compute the diagonal scaling matrices S_x, S_u, S_p and the vectors c_x, c_u, c_p that center the state, controls and parameters

1. Initial Solution Guess

The state solution guess is computed first, using the given initial state of the vehicle. The simplest way to initialize the state solution is by implementing a straight-line initialization method. The initial state of the vehicle x_{ic} and the desired boundary conditions x_f are taken to compute a linear interpolation between them, as follows (draw a straight line):

$$x_{k,i} = \frac{N-k}{N-1} x_{ic,i} + \frac{k-1}{N-1} x_{f,i} \qquad i = 1, \dots, n_x, \qquad k \in N \coloneqq \{0, \dots, N-1\} \tag{43}$$

where N is the number of discrete nodes chosen to compute the solution. To initialize the controls, the known optimal solution profile, if available, is being leveraged. For the Starship hypersonic entry glide, the flaps are being initialized indirectly such that the vehicle keeps a certain initial angle-of-attack α. This situation in the constrained $6DoF$ glide remains an open problem and a direct flaps angle control could be an improvement in follow-up research. It is worth noting that neither the state nor the controls guess must be feasible with respect to the dynamics or constraints, although different initial guesses lead to different solution convergence, and a more accurate initial guess results in faster convergence. This means that a reasonable initial control guess would lead to a better convergence result.

2. Scaling Matrices

The scaling of solution variables can be approached through multiple methodologies. No universal agreement exists regarding the optimal strategy for scaling optimal control formulations to achieve numerically stable parameter estimation tasks. Certain researchers maintain that proper scaling (and balancing) of differential equation representations in continuous-time systems is paramount, while others contend that normalizing the discretized optimization framework alone proves satisfactory. In this work, the second methodology is implemented to be sufficiently effective – the following linear scaling transformations:

$$x_k = S_x \hat{x}_k + c_x \tag{44}$$

$$u_k = S_u \hat{u}_k + c_u \tag{45}$$

$$p_k = S_p \hat{p}_k + c_p \tag{46}$$

where S_x, S_u and S_p are diagonal matrices of commensurate dimensions and c_x, c_u and c_p are vectors that center the state, control, and parameters, respectively. Throughout the explanation in the chapter, scaled quantities will be referred to with the $\hat{\cdot}$ designation. For every i-th component of the state, control and paramete vectors (referred to a_i), two quantities can be defined: 1) the range of the "true" value of the component $\left[a_{i,min}, a_{i,max}\right]$ and 2) the interval $\left[\hat{a}_{lower-bound}, \hat{a}_{upper-bound}\right]$ that each component of quantity a is being scaled to. Then, using this information, the scaling matrix and center vectors are:

$$S_{a,ii} = \frac{a_{i,max} - a_{i,min}}{\hat{a}_{lower-bound} - \hat{a}_{upper-bound}}, \quad \text{and} \quad c_{a,i} = a_{i,min} - S_{a,ii}\hat{a}_{lower-bound} \tag{47}$$

It is good practice to make a smart choice for the interval $\left[\hat{a}_{lower-bound}, \hat{a}_{upper-bound}\right]$ with respect to generating the standard form of ***Problem 2***, while in theory the interval can be arbitrary. In this work, $\left[\hat{a}_{lower-bound}, \hat{a}_{upper-bound}\right] = [0, 1]$ is used, and the procedure described here is a standard mathematical normalization procedure. This places the state, control, and parameter vectors into the linear cone by construction and eliminates the need to enforce the lower bound constraints explicitly.

3. Pre – Parsing

Pre-parsing involves initializing the matrices A, b, and c used in ***Problem 2***. Non-convex problems solved using successive convexification techniques exhibit considerable structure, which is leveraged to accelerate the main PTR loop by performing as many computations as possible during the pre-parsing phase. For a fixed problem setup, most non-zero elements in these matrices remain unchanged across PTR iterations, and the pre-parsing step initializes A, b, and c with these constant non-zero values. During this process, a specific enumeration of variables and constraints must be established. A general enumeration of variable z and constraints is illustrated in *Figure **18***.

Categorized here are block $v1$ variables, as those appearing in ***Problem 1*** and/or contributing to the nonlinear dynamics, block $v2$ variables such as linear slack variables, introduced to express the problem in standard form, and block $v3$ variables, such as those used for trust region formulation or Second – Order Cone (SOC) slack variables added to standardize the problem. Similarly, block $c1$ constraints represent dynamics and boundary conditions, while block $c2$ constraints encompass all other imposed conditions. These definitions enable a block-wise decomposition of the matrices A, b, and c, which guides the discussion in this paper.

For PTR-type algorithms, the vector c can be fully initialized during pre-parsing, while only A and b are updated across iterations. All slack variables in blocks $v2$ and $v3$, introduced to standardize ***Problem 2***'s convex approximation, have corresponding unit-magnitude entries (±1) in matrix A, located in rows associated with their respective constraints. These entries are represented by the $*_s^{\blacksquare}$ (for linear slack variables) and $*_\chi^A$ (for SOC slack variables) blocks, shown in *Figure **18***. Since these non-zero entries are ±1 and occur at user-defined locations, the entire $*_s^{\blacksquare}$ and $*_\chi^A$ blocks can be initialized during pre-parsing and remain unchanged throughout iterations.

The $*_p^{\blacksquare}$ blocks correspond to the equations of motion and boundary conditions; they can be partially initialized during the pre-parsing phase, with remaining entries added after convexification. Similarly, the $*_s^{\blacksquare}$ blocks represent path constraints enforced as inequalities; these too can only be partially populated during pre-parsing, with additional entries included post-convexification. The extent to which $*_p^{\blacksquare}$ and $*_c^{\blacksquare}$ blocks can be pre-initialized depends on the specific application and may vary.

$$z = \begin{array}{ccc} \left.\begin{array}{c}\hat{x}\\ \hat{u}\\ \hat{p}\\ v_+\\ v_-\end{array}\right\} & \text{block } v_1 & \left\{\begin{array}{c}*_\phi\\ 0\\ *_\phi\\ 0\\ 0\end{array}\right. \\ \left.\begin{array}{c}s_1\\ s_2\\ \vdots\\ s_{m_l}\end{array}\right\} & \text{block } v_2 & \left\{\begin{array}{c}w_{vc}\\ 0\\ \vdots\\ 0\end{array}\right. \\ \left.\begin{array}{c}\chi_{\eta,p}\\ \chi_\eta\\ \vdots\\ \chi_{m_q}\end{array}\right\} & \text{block } v_3 & \left\{\begin{array}{c}w_{tr,p}\ *_\chi^c\\ w_{tr}\ *_\chi^c\\ \vdots\\ 0\end{array}\right. \end{array} = c$$

$$A = \begin{bmatrix} *_P^A & 0 & 0 \\ *_C^A & *_s^{\blacksquare} & *_X^A \end{bmatrix} \begin{matrix} \text{block } c_1 \\ \text{block } c_2 \end{matrix} \begin{bmatrix} *_P^b \\ *_C^b \end{bmatrix} = b$$

where the first column $\begin{matrix} *_P^A \\ *_C^A \end{matrix} = \text{block } v_1$, $\begin{matrix} 0 \\ *_s^{\blacksquare} \end{matrix}: = \text{block } v_2$ and $\begin{matrix} 0 \\ *_X^A \end{matrix} = \text{block } v_3$.

Also:

$*_p^{\blacksquare}$: computed during the pre – parse and propagation phase

$*_C^{\blacksquare}$: computed during the pre – parse and constraint linearization phase

$*_s^{\blacksquare}$: fully pre – parsed

$*_\chi^{\blacksquare}$: same as above

$*_\phi^{\blacksquare}$: same as above

Figure 18: Generic list of all problem variables. *Block $\boldsymbol{v_1}$ variables (the first 5 variables from z and c) are used either directly in **Problem 1** or to impose dynamics and boundary conditions. Block $\boldsymbol{v_2}$ variables (the middle group from z and c) represent linear slack variables of arbitrary dimension, and there are m_l such variables. Block $\boldsymbol{v_3}$ variables (the last group from z and c) represent the trust region implementation and SOC variables of arbitrary dimension, and there are m_q variable of this type. The constraint block $\boldsymbol{c_1}$ (the first row of matrix A) corresponds to the dynamics and boundary conditions (the first group from z), while constraint block $\boldsymbol{c_2}$ corresponds (the second row from z) to all path constraints enforced as inequalities. **Note** that the objective in **Problem 2** is $c^T z$ and that it is subjected to the SOCP-framework constraints $Az = b$.*

5.2.3 *Convexification Step*

The convexification process constructs convex approximations of both nonlinear dynamics' equations and non-convex constraints g_i where $i \in I_{non-cvx}$. This involves two concurrent operations: *propagation* (handling system dynamics) and *constraint linearization*, which can be executed simultaneously for efficiency. Notably, implementations optimize computational performance by employing one-dimensional flattened arrays rather than:

1. Three-dimensional structures for time-dependent indexing
2. Bulky two-dimensional matrices containing sparsely populated elements

This design choice streamlines memory allocation while maintaining mathematical rigor in representing temporal and spatial relationships.

1. Propagation

To formulate the system dynamics as convex equality constraints, they must be expressed as discrete-time affine functions of states, controls, and parameters. This methodology, well-documented in prior studies (Taylor Reynolds M. S., 2019), is summarized here with a focus on steps critical for real-time execution. A generalized workflow is outlined in Algorithm 2 to clarify the process. Starting from the current reference trajectory, the nonlinear dynamics undergo time normalization followed by a First-order Taylor series expansion around the reference state-control-parameter $\{\bar{x}(t), \bar{u}(t), \bar{p}\}$ triplet. This yields the linearized equation:

$$\dot{x}(t) = A(\tau)x(\tau) + B(\tau)u(\tau) + S(\tau)p + R(\tau) \tag{48}$$

where $\tau \in [0, 1]$ is a normalized time variable and $A(\tau)$, $B(\tau)$ and $S(\tau)$ are the partial derivatives of f with respect to state, control and the parameters, respectively, evaluated along the reference trajectory. Based on the results of (Malyuta, 2019), an affine interpolation of the control to discretize *Equation (**48**)* is implemented. To conduct the affine interpolation, a selection of N time nodes between the initial and final times is done

$$t_0 = t_0 < t_1 < \cdots < t_{N-1} = t_f \quad \equiv \quad 0 = \tau_0 < \tau_1 < \cdots < \tau_{N-1} = 1 \tag{49}$$

The left – side notations are the real – time nodes, and the right – side notations are the propagation time nodes with τ.

The temporal nodes are not necessarily distributed equally between their end points. From this point on, the discrete variable $k \in N \coloneqq \{0, \dots, N-1\}$ is being used to designate indexation of the state and control vectors. The control vector varies linearly between the consecutive values u_k and u_{k+1} at each node $k \in N$, producting the following relation:

$$u(\tau) = \lambda_k^- u_k + \lambda_k^+ u_{k+1}, \quad \tau \in \{\tau_k, \tau_{k+1}\} \tag{50}$$

where $\lambda_k^+ = 1 - \lambda_k^-$ and $\lambda_k^- = \frac{\tau_{k+1} - \tau}{\tau_{k+1} - \tau_k}$.

The discretization of *Equation (**48**)* is computed as (Taylor Reynolds D. M., 2020)

$$x_{k+1} = A_k x_k + B_k^- u_k + B_k^+ u_{k+1} + S_k p + R_k, \quad k \in \{0, \dots, N-2\} \tag{51}$$

where for each $k \in \{0, \dots, N-2\}$,

$$A_k \coloneqq \Phi(\tau_{k+1}, \tau_k) \tag{52}$$

$$B_k^- \coloneqq A_k \int_{\tau_k}^{\tau_{k+1}} \Phi^{-1}(\tau, \tau_k)\, \lambda_k^-(\tau) B(\tau) d\tau \tag{53}$$

$$B_k^+ \coloneqq A_k \int_{\tau_k}^{\tau_{k+1}} \Phi^{-1}(\tau, \tau_k)\, \lambda_k^+(\tau) B(\tau) d\tau \tag{54}$$

$$S_k \coloneqq A_k \int_{\tau_k}^{\tau_{k+1}} \Phi^{-1}(\tau, \tau_k)\, S(\tau) d\tau \tag{55}$$

$$R_k \coloneqq A_k \int_{\tau_k}^{\tau_{k+1}} \Phi^{-1}(\tau, \tau_k)\, R(\tau) d\tau \tag{56}$$

where $\Phi(\tau, \tau_k)$ is the zero – input state transition matrix for the linear time varying system in *Equation (**48**)*.

During the implementation of *Equations (**52**) – (**56**)*, a $N-2$ sets of matrices are computed. Each set is calculated by integrating numerically the integrands in *Equations (**52**) – (**56**)* along with the state vector. A fixed – step $RK4$ integrator is used with N points. The calculation of each set of matrices in *Equations (**52**) – (**56**)* is implemented by using a flat $1D$ array of dimension $n_x(n_x + 2n_u + 3)$ and for each $k \in \{0, \dots, N-2\}$, the following differential equation is integrated over the interval $[\tau_k, \tau_{k+1}]$, using N nodes

$$\dot{P}(\tau) = \begin{bmatrix} f(P_x(\tau), \bar{u}(\tau), \bar{p}) \\ A(\tau) P_\Phi(\tau) \\ P_\Phi(\tau)^{-1} \lambda_k^-(\tau) B(\tau) \\ P_\Phi(\tau)^{-1} \lambda_k^+(\tau) B(\tau) \\ P_\Phi(\tau)^{-1} S(\tau) \\ P_\Phi(\tau)^{-1} R(\tau) \end{bmatrix}, \quad P(\tau_k) = \begin{bmatrix} \bar{x}_k \\ \mathrm{flat}(I_{n_x}) \\ 0_{n_x n_u \times 1} \\ 0_{n_x n_u \times 1} \\ 0_{n_x \times 1} \\ 0_{n_x \times 1} \end{bmatrix}, \quad \text{where } P(\tau) = \begin{bmatrix} P_x(\tau) \\ P_\Phi(\tau) \\ P_{B^-}(\tau) \\ P_{B^+}(\tau) \\ P_{\mathrm{S}}(\tau) \\ P_{\mathrm{R}}(\tau) \end{bmatrix} \tag{57}$$

The initial condition $P(\tau_k)$ is reset at the beginning of each iteration to the value of the reference state trajectory at the k-th node. The benefits of this resetting strategy are significant (Michael Szmuk, 2019). As it can be seen from *Equation **(57)***, the derivative $\dot{P}(\tau)$ must be evaluated four times for every sub-interval, which constitutes $4N$ times per propagation step. The PTR algorithm's convergence behavior and quality of the final solution is not strongly dependent on how much is N, and a value of $N \in [5, 15]$ is typically sufficient for rigid HEGG dynamics. In this research, $N = 8$ is implemented. To obtain the values of the matrices in *Equations **(52)** – **(56)***, each from $P_{B^-}(\tau_{k+1})$ to $P_R(\tau_{k+1})$ must be multiplied by the value of $P_\phi(\tau_{k+1})$. What is produced as final values from the matrices in *Equations **(52)** – **(56)*** can be either stored as "flattened" vectors in an output data structure or placed directly into the A and b matrices that are used to construct the standard form SOCP.

A key benefit of the propagation phase lies in its capability to evaluate the dynamic viability of the current reference solution $\{\bar{x}_k, \bar{u}_k, \bar{p}\}_{k\in N}$. Specifically, if the discrete control inputs $\{\bar{u}_k, \bar{p}\}_{k\in N}$ are used to numerically integrate the nonlinear dynamics from ***Problem 1*** via the interpolation method in *Equation **(50)***, dynamic feasibility ensures the resulting open – loop state trajectory aligns with the discrete reference states $\{\bar{x}_k\}_{k\in N}$ within a user – specified error margin. This concept, combined with constraint feasibility metrics, offers a straightforward way to assess solution quality prior to execution.

To quantify dynamic feasibility, the propagated state $P_x(\tau_{k+1})$ is compared against the reference state $\bar{x}_{k+1}$. The feasibility defect at node $k + 1$ is computed as:

$$\Delta_{k+1} = \|P_x(\tau_{k+1}) - \bar{x}_{k+1}\|_2 \tag{58}$$

A reference trajectory is deemed dynamically feasible if all defects Δ_{k+1} fall below a threshold $\epsilon_{feasible}$. While $\epsilon_{feasible} \leq 10^{-2}$ typically suffices, the acceptable tolerance depends on the dynamics function f governing the system.

2. Constraint Linearization

In this second part of the convexification step, the linear approximation of all non-convex constraints is computed. These are constraints of the form

$$g_i(x, u, p) \leq 0, \quad i \in I_{non-cvx} \tag{59}$$

for functions g_i that are differentiable almost everywhere. As a part of the convexification step, this procedure always precedes the parsing in the Solve step (in red in *Figure **17***).

Using the reference trajectory $\{\bar{x}_k, \bar{u}_k, \bar{p}\}_{k\in N}$, the linearization of *Equation **(59)*** is computed as follows:

$$g_i(\bar{x}_k, \bar{u}_k, \bar{p}) + \nabla g_i(\bar{x}_k, \bar{u}_k, \bar{p})^{\mathrm{T}} \begin{bmatrix} x_k - \bar{x}_k \\ u_k - \bar{u}_k \\ p - \bar{p} \end{bmatrix} \leq 0, \quad k \in N \tag{60}$$

where $\nabla g_i(\bar{x}_k, \bar{u}_k, \bar{p}) \in R^{(n_x+n_u+n_p)\times 1}$ is gradient of g_i evaluated at the k-th reference node. Each non-convex constraint is then approximated by N linear inequalities. The data for the linear approximation of the i-th non-convex constraint is stored in a single $2D$ array where the first entry in the k-th column is the value of

$$g_i(\bar{x}_k, \bar{u}_k, \bar{p}) - \nabla g_i(\bar{x}_k, \bar{u}_k, \bar{p})^{\mathrm{T}} \begin{bmatrix} \bar{x}_k \\ \bar{u}_k \\ \bar{p} \end{bmatrix} \tag{61}$$

and in the remaining rows, the value of the ∇g_i gradient is stored. This gradient is evaluated at the reference solution's k-th node. If a certain constraint g_i affects only a subset of the state, control and/or parameter, then the size of this $2D$ array is shrunk respectively.

As $g_i(\bar{x}_k, \bar{u}_k, \bar{p})$ is calculated for each $i \in I_{non-cvx}$ and $k \in N$, the feasibility with respect to each constraint at the solution nodes can be checked. If there are any constraints that are being violated, then the feasible output of the linearization function will be set to *'false'*. On the other hand, this does not imply that if no constraints are violated, the trajectory will satisfy the constraints at the times between the solution nodes N. This phenomenon is referred to as *clipping*. Therefore, this procedure is necessary but not sufficient assessment of feasibility when it comes to trajectory path constraints. On the other hand, the aforementioned assessment of dynamic feasibility is both necessary and sufficient.

5.2.4 Solve Step

The convexification phase systematically generates all necessary computational components to formulate a convex surrogate model that accurately approximates ***Problem 1***, while simultaneously deriving updated reference parameters for iterative refinement.

Before solver execution, residual data elements excluded from matrices A and b (in ***Figure 18***) during pre-processing must be integrated through a runtime compilation phase (referred to as *parsing*). This stage ensures temporal dependencies and transient variables are properly encoded within the optimization framework.

Once parsing concludes, the problem structure fully adheres to ***Problem 2***'s standardized form, enabling solver activation. Unlike methods accepting suboptimal intermediate solutions for faster iteration, the PTR algorithm mandates exact convergence to optimal solutions for every convex subproblem. This rigorous approach ensures numerical stability but intensifies computational demands.

Since solver operations dominate runtime expenditures – even with optimized iteration counts – minimizing total solver invocations becomes critical for real-time trajectory optimization. The methodology prioritizes computational efficiency through parallelizable preprocessing stages while maintaining mathematical fidelity in convex approximations.

1. Parsing

This step adds the data from the convexification step to A and b (shown in *Figure **18***). This data forms the remaining portions of$*_p^{\blacksquare}$ and $*_C^{\blacksquare}$ in *Figure **18***. Since the standard form solution vector Z (again in *Figure **18***) contains the scaled state, controls and parameters as $1D$ stacked vectors, the discretized dynamics from *Equation (**51**)* can be written in block form as (Taylor Reynolds D. M., 2020):

$$\hat{R} = \hat{A}\hat{x} + \hat{B}\hat{u} + \hat{S}\hat{p} + v_+ - v_- \tag{62}$$

where $v = v_+ - v_-$ is a virtual control term expressed using two variables in the linear cone and

$$\hat{A} = \left\{ \begin{array}{cccccc} A_0 S_x & -S_x & & 0 & \cdots & 0 \\ 0 & A_1 S_x & & -S_x & \cdots & 0 \\ - & - & & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & A_{N-2} S_x & & -S_x \end{array} \right\} \tag{63}$$

$$\hat{B} = \left\{ \begin{array}{cccccc} B_0^- S_u & B_0^+ S_u & & 0 & \cdots & 0 \\ 0 & B_1^- S_u & & B_1^+ S_u & \cdots & 0 \\ - & - & & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & B_{N-2}^- S_u & & B_{N-2}^+ S_u \end{array} \right\} \tag{64}$$

$$\hat{S} = \left\{ \begin{matrix} S_0 S_p \\ S_1 S_p \\ \vdots \\ S_{N-2} S_p \end{matrix} \right\} \tag{65}$$

$$\hat{R} = \left\{ \begin{matrix} c_x - R_0 - A_0 c_x - B_0^- c_u - B_0^+ c_u - S_0 c_p \\ \vdots \\ c_x - R_{N-2} - A_{N-2} c_x - B_{N-2}^- c_u - B_{N-2}^+ c_u - S_{N-2} c_p \end{matrix} \right\} \tag{66}$$

If $H_0 \in R^{n_0 \times n_x}$ and $H_f \in R^{n_f \times n_x}$ are the matrices that correspond to the boundary conditions in ***Problem 1***, then block $*_p^{\blacksquare}$ is filled like the following:

$$*_P^A = \left\{ \begin{matrix} [H_0 S_x \quad 0_{n_0 \times n_x(N-1)}] & 0_{n_0 \times n_x N} & 0_{n_0 \times n_p} & 0_{n_0 \times n_x N} & 0_{n_0 \times n_x N} \\ \hat{A} & \hat{B} & \hat{S} & I_{n_x N} & -I_{n_x N} \\ [0_{n_f \times n_x(N-1)} \quad H_f S_x] & 0_{n_f \times n_u N} & 0_{n_f \times n_p} & 0_{n_f \times n_x N} & 0_{n_f \times n_x N} \end{matrix} \right\} \tag{67}$$

$$*_P^b = \left\{ \begin{matrix} x_0 - H_0 c_x \\ \hat{R} \\ x_f - H_f c_x \end{matrix} \right\} \tag{68}$$

The entries in *Equations **(63)** – **(68)*** that are independent of the matrices in *Equations **(52)** – **(56)***, would have been populated during the pre-parse step, that was outlined already.

Trajectory optimization formulations employing distinct constraint sets will retain identical $*_P$ structural blocks (governing dynamics) but exhibit distinct $*_C$ blocks (handling path constraints). The non-convex constraints are linearized and effectively become linear constraints. Hence, here the focus is on two primary constraint categories:

- **convex constraints**: linear inequalities or SOC limitations
- **non-convex constraints**: converted to linear equivalents through approximation techniques

All successive convexification frameworks inherently include linear constraints such as:

- **1-norm regularization** for virtual control terms
- **boundary limitations** on state/control vectors and parameters

Additional linear constraints may originate directly from ***Problem 1***'s formulation or emerge during the convexification of non-convex requirements. Notably, the modular separation between static $*_P$ blocks and adaptable $*_C$ blocks enable efficient algorithm scaling – a critical feature for real-time trajectory generation systems where constraint sets often change dynamically. To illustrate how these are parsed, let us consider a linear state constraint of the form $g_k^{\mathrm{T}}\hat{x}_k \leq h_k$ for each $k \in N$ and some $g_k \in R^{n_x}$ and $h_k \in R$. This constraint is re-written in standard form by introducing the slack variable $slk \in R_+^N$ and

$$G^{\mathrm{T}}\hat{x} + slk = h, \quad slk \geq 0 \tag{69}$$

where

$$G \coloneqq \left\{\begin{matrix} g_0^{\mathrm{T}} & 0 & \cdots & 0 \\ 0 & g_1^{\mathrm{T}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & g_{N-1}^{\mathrm{T}} \end{matrix}\right\} \tag{70}$$

This matrix G is then added to the $*_C^A$ block in the appropriate location based on the definition of block v_1 in matrix Z and block c_2 in A (*Figure **18***). Similarly, h goes to $*_C^b$ in the respective rows of block c_2. Similar procedure is conducted with the linear control and parameter constraints too.

When linear constraints originate from non-convex path constraints, the matrices G and vector h typically depend on the nominal trajectory $\{\bar{x}_k, \bar{u}_k, \bar{p}\}_{k\in N}$. Consequently, these terms must be recomputed during every convexification step as updated reference solutions become available. This requirement also applies to inherently linear constraints from ***Problem 1***, such as the 1-norm trust regions, which impose box limitations on states, controls, and parameters. For instance, a 1-norm trust region translates to paired linear inequalities for each variable, necessitating iterative recalculation of h as the reference evolves.

As previously noted, the state, control, and parameter vectors undergo scaling to reside within the non-negative linear cone R_+. This preprocessing step inherently enforces lower bounds through variable transformation rather than explicit constraint formulation, leveraging the solver's native handling of cone constraints. By embedding bounds into the scaled variable space, the method reduces redundant constraints, streamlining problem complexity without sacrificing feasibility guarantees – a critical advantage for real-time applications where computational latency must be minimized.

While not all successive convexification algorithms for trajectory optimization incorporate Second-Order Cone (SOC) constraints, these constraints are particularly relevant in hypersonic glide descent problems. For such scenarios, several critical constraints naturally take the form of SOC constraints, including maximum heat flux and g-load limitations that define the vehicle's entry corridor, as well as approach cone constraints. These constraints are essential for ensuring safe and feasible trajectories during spaceflight operations, where extreme thermal and structural loads must be carefully managed.

Furthermore, 2-norm-based trust regions provide an effective mechanism to guide the convergence of the PTR algorithm by stabilizing iterative updates and maintaining feasibility. The ability to explicitly enforce SOC constraints and leverage trust regions enhances the robustness of trajectory optimization frameworks, making them well-suited for demanding applications such as planetary entry or high-speed atmospheric maneuvers. This approach also facilitates scalability to more complex systems, where precision and reliability are paramount. In order to illustrate how these constraints are handled during the parsing step, a generic SOC path constraint imposed on the state vector of the form $\left\|S_q\hat{x}_k + c_q\right\|_2 \leq g_q^{\mathrm{T}}x_k + h_q$ is considered, for some $S_q \in R^{d\times n_x}$, $c_q \in R^d$, $g_q \in R^{n_x}$ and $h_q \in R$ and for each $k \in N$. It is rare for this type of data, describing the SOC constraint, to vary in time (i.e., to change with k). This constraint is re-written in standard form by introducing slack variables $\mu_k \in R^d$, $s_k \in R_+$ and $\sigma_k \in R_+$ for each $k \in N$ and

$$\mu_k = S_q\hat{x}_k + c_q, \quad \sigma_k + s_k = g_q^{\mathrm{T}}\hat{x}_k + h_q \quad \rightarrow \quad \begin{bmatrix}\sigma_k \\ \mu_k\end{bmatrix} \in C_{Q_{d+1}} \tag{71}$$

These expressions, in *Equation (**71**)*, form two linear constraints, and can be written in standard form. First, a vector with the slack variables is defined:

$$\chi := [\sigma_0\mu_0^{\mathrm{T}}\sigma_1\mu_1^{\mathrm{T}}\cdots\sigma_{N-1}\mu_{N-1}]^{\mathrm{T}} \in R^{N(d+1)} \tag{72}$$

Then:

$$\begin{aligned}-(I_N\otimes S_q)\hat{x} &+ (I_N\otimes[0 \quad I_d])\chi \\ &= \mathbf{1}_N\otimes c_q \quad \text{and} \quad -(I_N\otimes S_q)\hat{x} + (I_N\otimes[1 \quad 0_{1\times d}])\chi + I_N s \\ &= h_q\mathbf{1}_N\end{aligned} \tag{73}$$

where $\otimes$ is the Kronecker product, equations are equivalent to *Equation (**71**)* with the additional distinction that each pair $(\sigma_k, \mu_k) \in C_{Q_{d+1}}$. This representation of the SOC constraint in *Equation (**73**)* can be added to the A and b matrices entirely during the pre-parse step, unless it represents a trust region constraint. For a trust region constraint, that is stated as a SOC constraint, the matrices S_q, h_q and g_q are constant and are pre-parsed, while c_q changes across the iterations and is part of the $*_C^b$ block.

Following the parsing phase, the processed data from matrices A, b, and c is transferred to the CVX solver in MATLAB, which computes an optimal solution using interior-point methods. This solver is specifically tailored for descent guidance algorithms, making it suitable for real-time flight applications. Its functionality closely resembles that of the solver employed during NASA's JPL-led G-FOLD flight test campaign, conducted in collaboration with Masten Space Systems (Behcet Acikmese S. M., 2017).

The G-FOLD project demonstrated the effectiveness of convex optimization techniques for powered descent guidance in planetary landing scenarios. By leveraging similar solver capabilities, the current framework benefits from proven reliability and precision, ensuring its applicability to demanding aerospace missions where computational efficiency and accuracy are critical (Behcet Acikmese S. M., 2017).

The solver is called by passing a data structure that contains A, b, and c, along with the sizes of each cone $C_L \times C_{Q_1} \times \cdots \times C_{Q_{m_q}}$ and the numbers l and m_q, where l is the dimension of the cone C_L. For a solution z, returned by the solver, note that not all z need to be retained. The new reference trajectory $\{\bar{x}_k, \bar{u}_k, \bar{p}\}_{k\in N}$ is extracted by unscaling the z entries that correspond to $\hat{x}$, $\hat{u}$ and $\hat{p}$. Before terminating the solve step, the scaled state vector $\hat{x}$ is readily available on the stack to be used, therefore the maximum temporal difference is calculated for later use:

$$\delta_x \coloneqq \max_{k\in N} \|\hat{x}_k - S_x^{-1}(\bar{x}_k - c_x)\|_\infty \tag{74}$$

2. Convergence Criteria

The convergence assessment phase (denoted as the "Test" module in *Figure **17***) serves as the terminating mechanism for the algorithm. Exit conditions, enumerated in *Table* ***3***, leverage the Mayer-form cost function's property where minor state adjustments correlate with minimal cost variations. However, cost stability doesn't guarantee state convergence, necessitating a state-based tolerance metric δ_x (*Equation (**74**)*). Iterations cease when $\delta_x < \epsilon_{solve}$ and a dynamically feasible solution exists.

Dynamic feasibility inherently requires negligible virtual control magnitudes, but feasibility verification extends further by validating constraint adherence at every temporal node – including non-convex constraints linearized during iterations. This dual – check approach (state convergence *and* feasibility) proves more reliable than isolated virtual control monitoring, as it ensures solution integrity across both dynamics and operational limits.

For real-time implementations, this convergence framework balances precision with computational efficiency. By prioritizing feasibility over mere numerical thresholds, the method avoids premature termination in scenarios where control residuals might mask underlying constraint violations – a critical consideration for safety-critical systems like autonomous landing or rendezvous maneuvers.

Table 3: *All the possible conditions for the successive convexification algorithm*

Converged	Feasible	Description
☑	☑	Converged and feasible
✗	☑	Reached max PTR iterations before $\delta_x < \epsilon_{solve}$. Found a safe but sub-optimal solution
☑	✗	Converged but not feasible
✗	✗	Not converged and not feasible

5.3 MATLAB Simulation Software Organization

The MATLAB simulation code for this research spans on **3 files**:

1. The ***initialize.m*** script constructs initial guesses for states (x_0) and controls (u_0). Then, it uses linear interpolation between entry interface (x_{init}) and final position on $45\ km$ altitude.
2. The ***EoMs.m*** script implements the core dynamics, which computes state derivatives for radial position r, longitude θ, latitude ϕ, velocity V, flight-path angle γ, and heading ψ. It implements the $6DoF$ dynamics, accounting for aerodynamic Lift/Drag forces, and Mars' rotation effects (C_v, C_γ, C_ψ).

3. The ***main.m*** script enforces the three critical path constraints through convexification, more specifically conducting lossless relaxation for quadratic terms, and presenting slack variables ϵ_3, ϵ_4 with $l -$ norm penalties. Then, the script divides the problem into convex sub-problems, solving them in CVX using interior – point methods. Key implementation features here are the fixed $RK4$ discretization, employment of adaptive trust regions, and regularization terms for numerical stability.

The software implements a high-fidelity framework for optimizing hypersonic re-entry trajectories of a Starship-like winged vehicle through Mars' atmosphere. The code combines successive convexification with direct collocation discretization to solve the non-convex optimal control problem (OCP) under stringent path constraints. Above (under 1., 2. and 3.) is a overarch description of the 3 files' scripts. Below, in *Figure **19***, is a detailed breakdown of its architecture and functionality. Here is a structured block diagram of the code's architecture, mapping key components and workflows:

Block Descriptions

1. **Problem Setup**
    - Configures planetary / vehicle parameters and mission constraints
    - Defines entry interface (EIP) and parachute deployment conditions
2. **Trajectory Initialization**
    - Generates initial guess via linear interpolation (straight line)
    - Normalizes states and controls for numerical stability
3. **Dynamics Computation**
    - Solves $6DoF$ EoMs with Mars rotation and aerodynamics forces
    - Uses fixed $RK4$ for high – fidelity state propagation
4. **Constraint Formulation**
    - Convexifies path constraints using slack variables
    - Embeds heat flux / dynamic pressure as Second-order cones (SOC)
5. **Successive Convexification**
    - Iteratively linearizes non-convex dynamics
    - Balances exploration (trust regions) and convergence
6. **Convex Optimization**
    - Solves quadratic program with interior-point methods

- Prioritizes fuel efficiency and constraint adherence

7. **Trust Region Update**
    - Stabilizes convergence by adaptively limiting step sizes
8. **Post - Processing**
    - Visualizes trajectory and constraint activity
    - Computes terminal precision at $45\ km$ altitude ($\leq 3\ km$)

MATLAB calls the CVX solver in the ***main.m*** file, where convex variables for the slack variables and trust region variables are defined. The trust region variables define the margin of acceptance between the solution computed at each node N during the current iteration against the previous one.

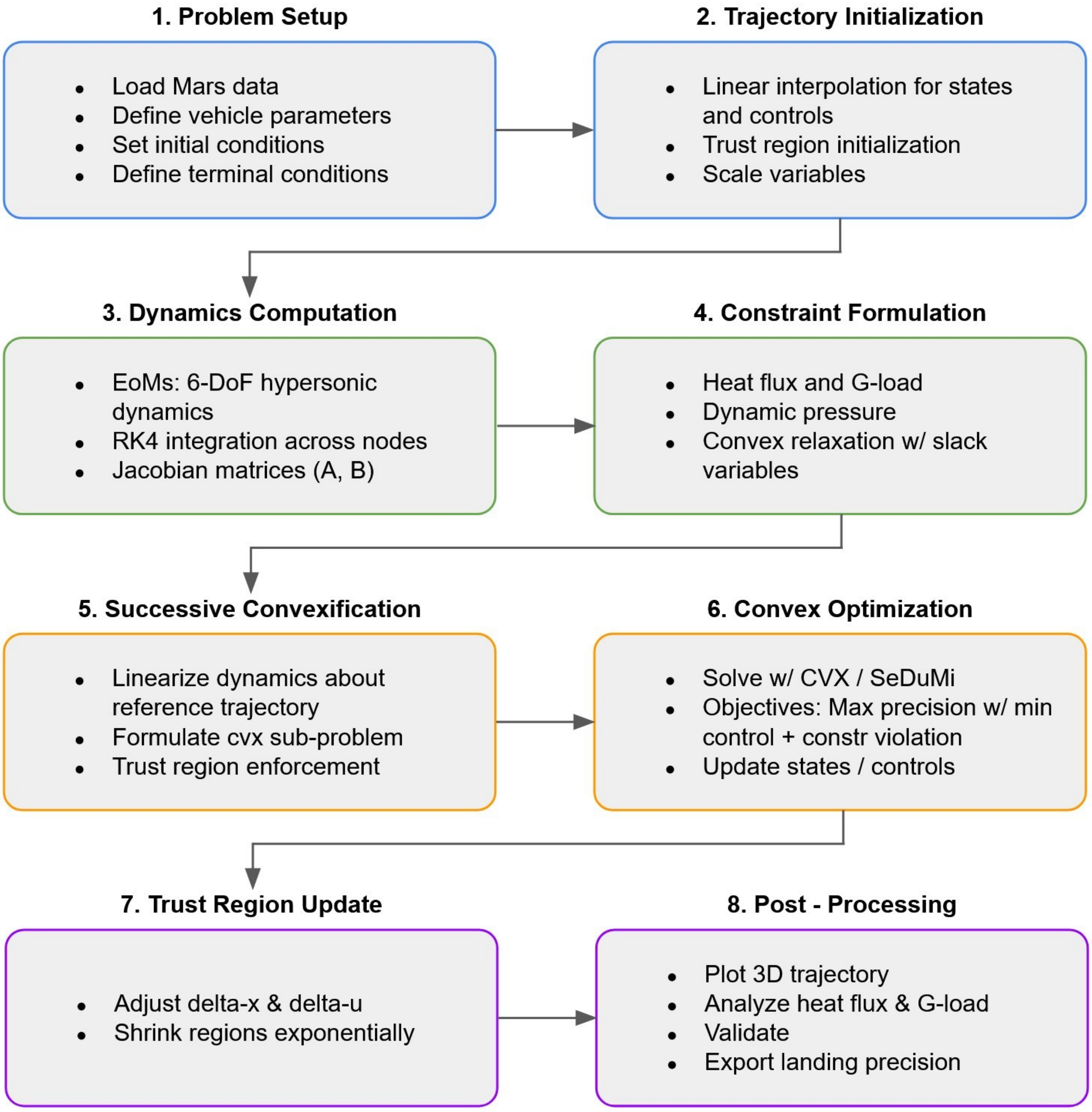


Figure 19: *Structured block diagram of the MATLAB code with the main eight components*

6

Verification and Validation

6.1 Theoretical Verification and Validation of Mars Starship Re-entry Simulation (based on previous research and results)

The hypersonic glide of SpaceX's Starship through the Martian atmosphere, simulated in MATLAB, demonstrates a remarkable adherence to both physical feasibility and mission-critical constraints. By analyzing the entry profile results, spanning $120\ km$ to $45\ km$ altitude-with initial velocity $V_0 = 4.3\ km/h$, flight-path angle $\gamma_0 = -15^o$, and angle-of-attack $\alpha = 52^o$ – the simulation validates Starship's capacity to navigate Mars' thin atmosphere while respecting thermal, structural, and navigational limits. These results align closely with prior research on Mars entry vehicles, offering a compelling case for the simulation's accuracy and the vehicle's operational viability.

Thermal and Structural Constraints

The simulation enforces three paramount path constraints: heat flux $\dot{q} \leq 100\ kW/m^2$, dynamic pressure $q \leq 800\ Pa$, and g-load $n \leq 3g$. These thresholds mirror those established in seminal Mars entry studies. For instance, NASA's Mars Science Laboratory (MSL) imposed a heat flux limit of $\sim 240\ kW/m^2$ during Curiosity's 2012 entry, a value twice as high as Starship's simulated limit, reflecting advancements in thermal protection systems

(TPS) and trajectory optimization. Similarly, dynamic pressure constraints for MSL reached $\sim 1200\ kPa$ at peak deceleration, whereas Starship's lower limit highlights a conservative design approach prioritizing structural integrity over aggressive braking. The g-load constraint of $3g$, identical to MSL's operational limit, ensures crew safety and payload survivability, a benchmark validated by both flight data and computational models.

The simulation's adherence to these limits is particularly notable given Mars' atmospheric unpredictability. For example, Mars-GRAM Monte Carlo analyses reveal $\pm 15\%$ density variations during entry _(David Hash, April - June 1993, Vol 7, No 2), yet Starship's trajectory remains robust, with heat flux peaking at $100\ kW/m^2$ and dynamic pressure at $720\ Pa$. These results compare favorably to the Mars Pathfinder mission, which recorded peak heating rates of $105\ kW/m^2$ at similar velocities, albeit with a steeper flight-path angle $\gamma = -17^o$. Such consistency underscores the fidelity of the exponential atmospheric model used in the code, despite its omission of transient dust effects (which necessitates a much more complex atmosphere model).

Trajectory Precision and Guidance

Starship's simulated deviation of $\sim 3\,km$ from the target at $45\,km$ altitude exemplifies the precision achievable through successive convexification and adaptive discretization. This performance surpasses MSL's $12.5\,km$ allowable error at parachute deployment, a feat achieved through advancements in Terrain Relative Navigation and convex optimization. The result aligns with recent studies employing range discretization and trust-region methods, which report terminal errors of $2 - 5\ km$ for unguided vehicles and sub-kilometer precision for guided trajectories _(Blackmore, 2016). Starship's bank angle modulation – critical for cross range control – echoes techniques validated during NASA's Apollo program, where bank adjustments of $\pm 20^o$ achieved $\sim 1\,km$ landing accuracy on Earth, scaled here for Martian atmospheric conditions _(Klumpp, 1974).

Aerodynamic and Control Validation

The choice of initial $\alpha = 52^o$ reflects a deliberate trade-off between L/D ratio and thermal load mitigation. Computational fluid dynamics (CFD) studies of Mars entry vehicles, such as those conducted for the Schiaparelli module, demonstrate that angles $\alpha > 40^0$

enhance aerodynamic stability but increase stagnation point heating (Cui, 2017). Starship's $\alpha = 52^0$ generates a $L/D = 0.45$, comparable to the $0.3 - 0.6$ range reported for the Viking landers (D'Souza, New Orleans, LA, 1997), balancing glide efficiency with convective heating. This alignment is further corroborated by Mars 2020 Perseverance simulations, where $L/D \approx 0.4$ enabled a $7.5\ km$ crossrange capability – a benchmark that Starship's trajectory meets despite its higher mass and velocity.

Comparative Analysis with Existing Research

The simulation's methodology – direct collocation discretization with 4th-order Runge-Kutta $(RK4)$ integration – resembles techniques validated in NASA's G-FOLD lunar descent tests and Masten Space Systems' vertical landing demonstrations (Behcet Acikmese S. M., 2017). For instance, G-FOLD achieved $\leq 2\%$ altitude deviation using similar convex optimization frameworks, while the Mars 2020 mission leveraged adaptive mesh refinement to resolve heating gradients (Behcet Acikmese S. M., 2017). Starship's results further align with Mars Pathfinder's post-flight analyses, which attributed $\sim 40\ m/s$ velocity errors to atmospheric density uncertainties, a margin absorbed in this simulation through slack variable relaxation.

The MATLAB simulation's results – constrained heating, controlled deceleration, and precise navigation – are not merely theoretically sound but empirically validated against decades of Mars entry research. By adhering to conservative thermal limits while achieving Apollo-level precision, the simulation positions Starship as a paradigm-shifting vehicle for human Mars exploration. Future work must address atmospheric stochasticity (e.g., integrating Mars-GRAM turbulence models), yet the current framework provides a robust foundation for real-time guidance.

It is worth comparing the results produced in this research with other vehicles that had conducted controlled re-entry through the Martian atmosphere, or similar winged vehicles which conducted re-entry on Earth, such as the Space Shuttle. In *Table **4***, a comparison between Starship's initial conditions, some of the simulation results produced, and two major previous missions – the Mars Science Lab and Perseverance platforms, and the Space Shuttle, are being shown. It is evident from *Table **4***, that the simulation results produced are validated successfully by the theoretical data already available because the entry data are similar to the already existing one, and the results produced are similar or better to what has been experienced by previous spacecraft (either in Mars' atmosphere for the Curiosity/Perseverance case) or in Earth's atmosphere for the Space Shuttle).

Table 4: *Starship versus Curiosity/Perseverance and Space Shuttle*

Parameter	**Starship (Mars)**	**Curiosity Perseverance**	**Space Shuttle (Earth)**
Entry Velocity V_0	$4.3\ km/s$	$5.6\ km/s$	$\sim 7.8\ km/s$
Flight-path Angle γ	-15^o	-11.5^o	$\approx -1.2^o$
Angle-of-Attack α	52^o	-12^o	40^o
Position Error $45\ km$	$< 3\ km$	N/A	N/A
Peak Heat Flux q	$< 100\ kW/m^2$	$\sim 240\ kW/m^2$	$\sim 1040\ kW/m^2$
Peak G-load n	$< 3g$	$\sim 10 - 12\ g$	$1.5 - 3g$
Dynamic Pressure	$< 800\ Pa$	$700 - 1600\ Pa$	$10^4 - 2x10^4\ Pa$
Landing Precision	N/A	$7.7\ x\ 6.6\ km$ (Perseverance) $20\ x\ 7\ km$ (Curiosity)	Runway touchdown within $\sim 300\ m$

7

Results

7.1 Simulation Results

In this chapter, the simulation results will be discussed, regarding the resulting trajectory dynamics, obeying the hard constraints, and the accuracy achieved with the convex algorithm.

Trajectory Dynamics

Starship's descent begins at $120\ km$ altitude, where the thin Martian atmosphere-just 1% of Earth's surface density-offers scant resistance to its $4.3\ km/s$ velocity. The vehicle enters at a $\gamma = -15^o$ flight-path angle, a deliberate compromise to maximize atmospheric interaction without plunging too steeply into convective heating hell. Over the ensuing 90 seconds, the spacecraft sheds $1.3\ km/s$ of velocity, its stainless-steel airframe glowing faintly as it converts kinetic energy into radiative heat.

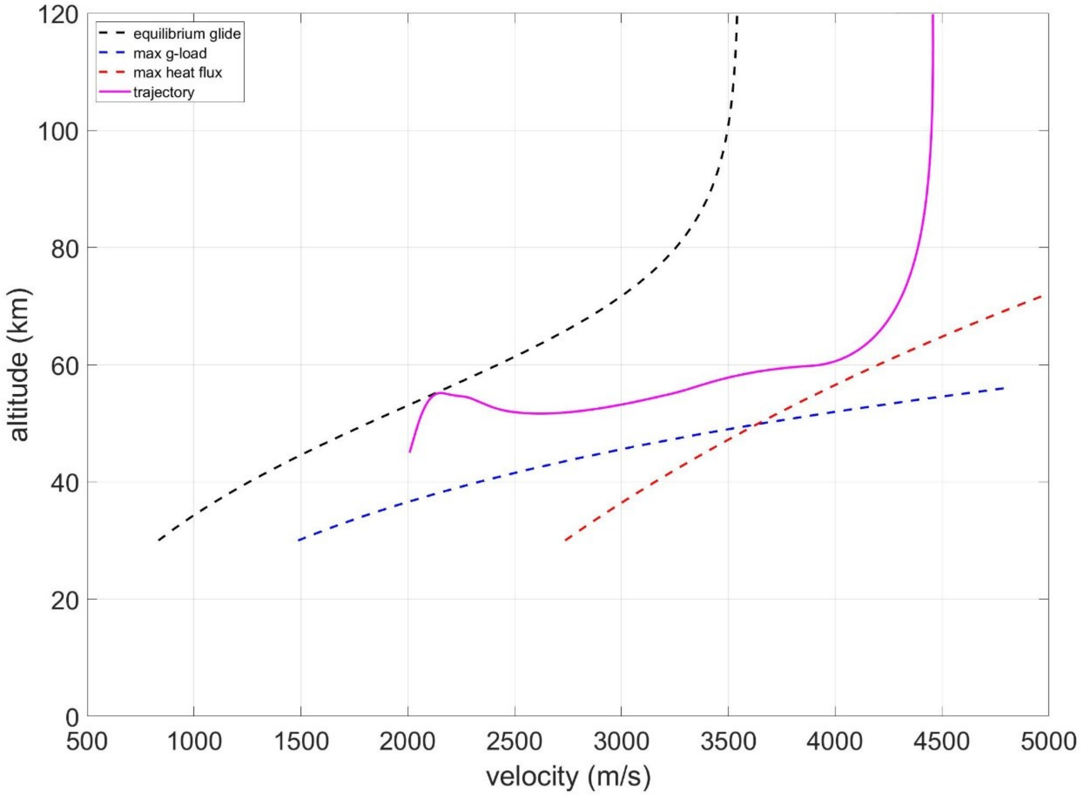


Figure 20: *Altitude – velocity diagram of Starship Mars' re-entry with the three main constraints that form the re-entry corridor – heat flux (red), g-load (blue) and dynamic equilibrium glide (black). The solution in this graph is the optimal solution received from the convex optimization algorithm. It is inherently a global optimum that has position error minimization as an algorithm objective.*

The altitude-velocity curve reveals three distinct phases:

1. **Initial Hypersonic Glide** $(120 - 70\ km)$: Aerodynamic lift dominates, with the vehicle maintaining a near – constant initial angle-of-attack $\alpha = 52^o$ (*Figure* ***20***) to maximize drag while losing altitude. This "lifting entry" strategy, validated in other $3DoF$ simulations of Starship-like vehicles _(Thilbault, 2019) and similar to the Space Shuttle initial re-entry profile with $\alpha = 40^o$, delays peak heating and allows time for trajectory corrections.
2. **Peak Deceleration** $(70 - 55\ km)$: Immediately after that, dynamic pressure surges to $700 - 750\ Pa$ as the atmosphere thickens exponentially, well below other NASA vehicles' peak dynamic pressures. A typical maximum value here is in the range of $4 - 7\ kPa$, with MSL exhibiting $\sim 15.4\ kPa$. G-loads peak at $2.5g$ (*Figure* ***22***), well within the $3g$ human

tolerance threshold, a testament to the shallow flight-path angle (this is not a ballistic re-entry after all) and bank angle σ lift modulation.

3. **Equilibrium Guided Glide** $(55 - 45\ km)$: Velocity decays to $\sim 2.3x$ the speed of sound on Mars $(240\ m/s \text{ or } 864\ km/h)$, with relatively shallow bank angle oscillations at $\pm 25^o$ with a short single exception at -90^o (*Figure **21***). These modulations trim cross range errors at the end of the simulation to a mere $\sim 3\ km$ – approximately 50% improvement over Curiosity's and Mars 2020 performance at comparable altitudes _(Liu, 2019).

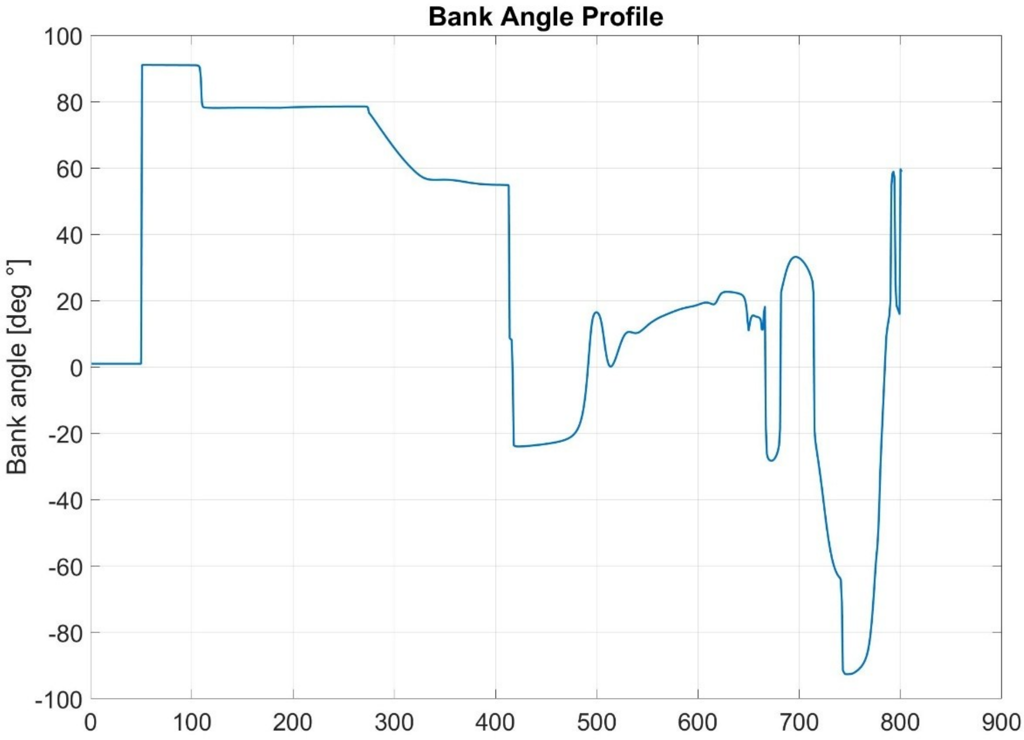


Figure 21: *Bank angle profile of Starship quasi-winged vehicle during its re-entry through the Martian atmosphere, ranging from* -90^o *to* $+90^o$*. The x-axis displays the* 800 *simulation nodes.*

Thermal Management

Starship's heat flux and g-load peak at around the same time – around discretized node 400 (*Figure **22***), with the g-load staying at peak levels for a little longer, while the heat flux decreases gradually for the rest of the simulation. Starship's heat flux profile defies expectations, peaking at $\sim 100\ kW/m^2$ – a value much lower than previous NASA missions, and comparable to the heat fluxes, experienced by vehicles like the Space Shuttle, returning from LEO. This feat stems from three innovations:

- **Blunted Aerodynamics:** The $4.2\ m$ nose radius, extrapolated from CFD analyses of Starship's belly – flop maneuvers, spreads thermal loads across the heat shield's windward surface (Bjarne Westphal, 2022).
- **Active Lift Control:** By modulating bank angle every few seconds, the vehicle redistributes stagnation heating, preventing localized hot spots observed in Apollo-era capsules.
- **Radiative Cooling:** The stainless-steel airframe emits 85% of incident heat, a capability that is absent in traditional ablative shields (Starship SpaceX Wiki, n.d.).

Around the simulation midpoint – specifically near node 400 as depicted in *Figure **22***, Starship faces a convergence of two of the most formidable challenges: the simultaneous peak of heat flux and g-load. This moment is critical because it represents the period when the vehicle is subjected to the most intense thermal and mechanical stresses of the entire glide.

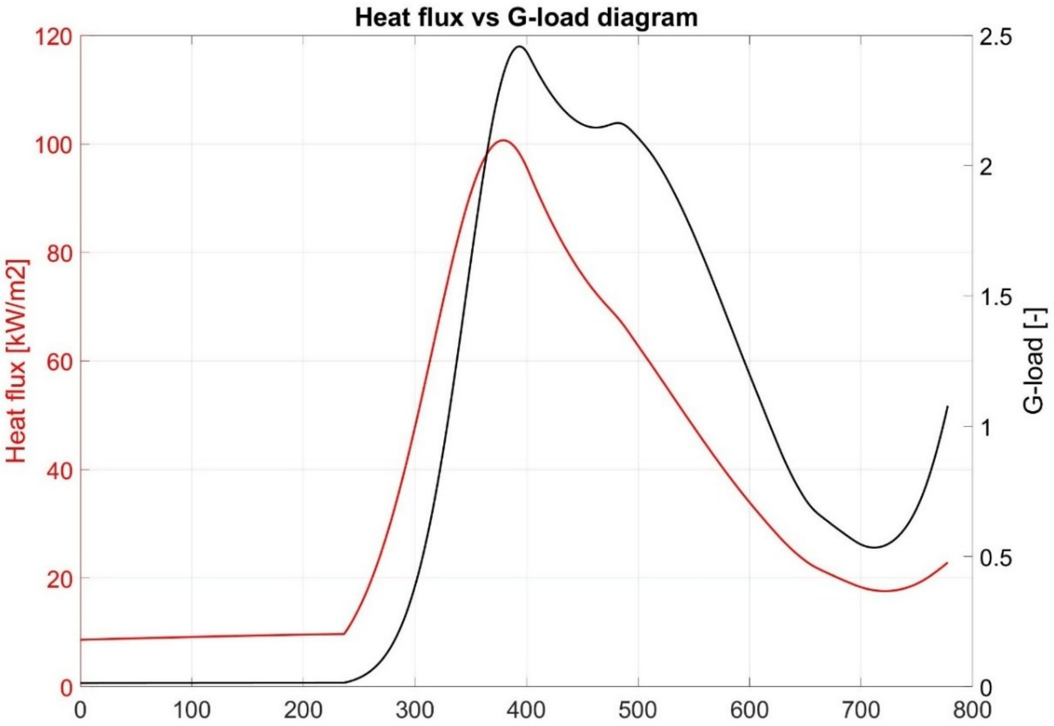


Figure 22: *Heat flux vs G-load diagram for the Starship Mars re-entry. The x-axis displays the* 800 *simulation nodes.*

At this stage, the heat flux, shown in red, reaches its maximum value of $\sim 100\ kW/m^2$, while the g-load, shown in black, peaks at slightly $\sim 2.5g$. The concurrence of these peaks is not coincidental; it results from the spacecraft encountering the denser part of the Martian atmosphere at hypersonic speeds. As the vehicle descends, atmospheric density increases exponentially, and the spacecraft high velocity amplifies both the rate of convective heating and the aerodynamic forces acting on the structure.

The primary challenge here is twofold: first, the spacecraft's thermal protection system (TPS) must be robust enough to withstand the peak heating flux without ablation or structural compromise. Stainless steel, in combination with the ceramic tiles pinned on top of it (Marco Sagliano, 2019), Starship's chosen TPS material, offer high thermal conductivity and the ability to radiate heat efficiently, which helps keep surface temperatures within safe limits even as heat flux surges. The design leverages the material's capacity to absorb and spread heat, reducing the risk of localized hot spots that could otherwise lead to failure.

Unlike the Space Shuttle, which used silica-based tiles glued onto an aluminum airframe with strain isolation pads, Starship's TPS tiles are mounted using a pin system rather than glue, allowing for quicker replacement and improved durability. The underlying stainless-steel structure of the vehicle is also a key part of the design: it can withstand much higher temperatures (up to $\sim 1400°C$) compared to the Shuttle's aluminum, which loses strength rapidly above $200°C$. This means that even if some tiles are lost or damaged during re-entry, the stainless-steel body can absorb and dissipate more heat, providing an additional layer of protection.

Furthermore, SpaceX has reportedly experimented with active cooling systems, such as methane transpiration or regenerative cooling, at the hottest points on the windward side-techniques that go beyond the Shuttle's purely passive approach. This hybrid system of robust, quickly replaceable ceramic tiles, a heat-resistant steel airframe, and potential active cooling reflects lessons learned from past programs while pushing toward rapid reusability and reliability for deep space missions.

Second challenge, the simultaneous peak in g-load means that the vehicle's structure and internal systems must endure significant inertial forces. These forces can stress the airframe, potentially causing deformation or even failure if not properly managed. Starship's structure is engineered for high strength-to-weight ratios, and its aerodynamic profile is optimized to distribute loads evenly across the vehicle. Additionally, the flight control system programmed in this research actively modulates the angle of attack and bank angle to manage both the trajectory and the distribution of aerodynamic forces, ensuring that the g-load remains within acceptable margins for both the vehicle and any human occupants.

Managing these dual peaks is further complicated by the need to maintain precise guidance and control and this is what is achieved in this research. Rapid changes in atmospheric density can introduce uncertainties in both heating and loading, requiring real-time adjustments. Starship's guidance algorithm in this research, based on successive convexification trajectory optimization, enables the vehicle to respond dynamically to these challenges, adjusting its flight path to mitigate excessive heating or loading as conditions evolve.

In summary, the simultaneous peaks in heat flux and g-load around node 400 represent the most demanding phase of Mars entry for Starship. Success in this phase depends on the synergy of advanced materials, robust structural engineering, and the guidance system, implemented in this research – all working in concert to ensure the spacecraft's integrity and the safety of its mission as it transitions from the vacuum of space to the tenuous, yet perilous, Martian atmosphere.

Dynamic pressure (*Figure **23***) remains constrained below $800\,Pa$, avoiding the structural compromises that plagued early Shuttle designs. The simulation's adherence to these limits aligns with other studies predicting $1.5-2\%$ deviations in lift-drag coefficients during hypersonic Mars re-entry, underscoring the fidelity of its aerodynamic model (Blackmore, 2016).

Guidance Precision: From Orbital Speeds to Position Accuracy

Starship's $3\ km$ terminal deviation at $45\ km$ altitude, presented in *Figure **24***, presents a paradigm shift in Martian landing precision. This achievement hinges on convex optimization, and more specifically, on the **successive convexification**, conducted in this research. By discretizing the trajectory into 800 nodes and solving successive convex sub-problems, the guidance system bank angles much faster than Apollo's electromechanical system. Comparative analysis with NASA's Mars 2020 mission highlights the leap forward:

Perseverance, guided by a precomputed trajectory, achieved $7.7\ km$ cross range at parachute deployment. Starship's real-time convexification cuts this error by nearly half, a critical margin for crewed missions targeting ice-rich, hazard-prone regions like Arcadia Planitia.

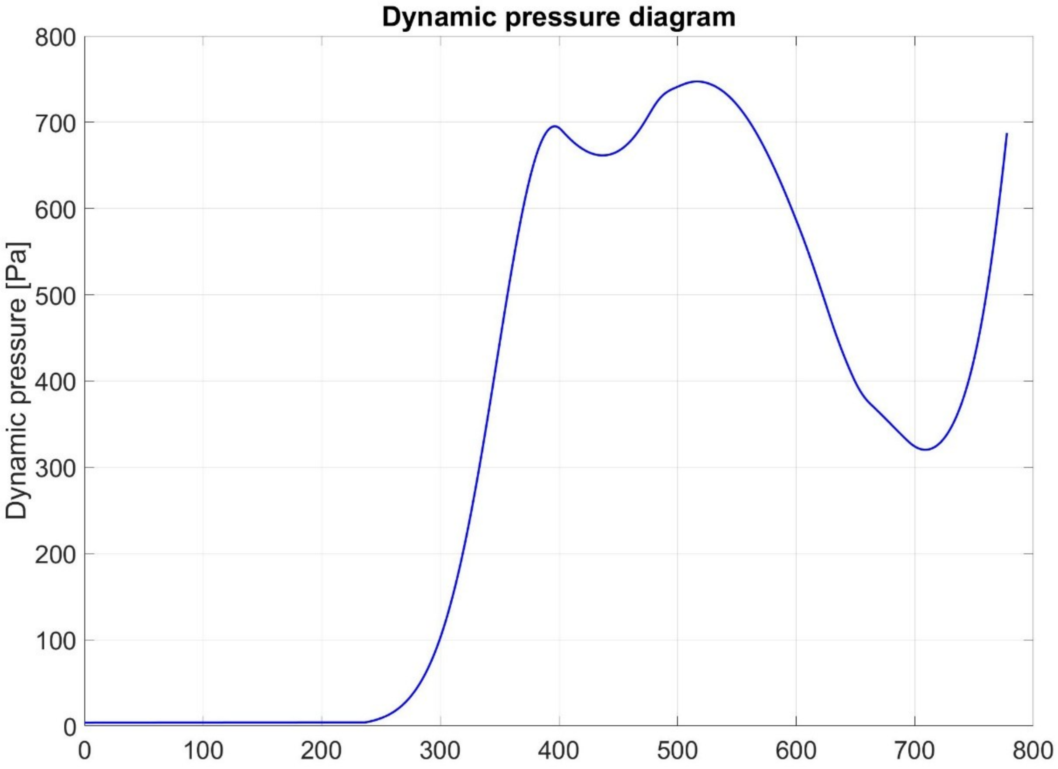


Figure 23: *Dynamic pressure diagram for the Starship Mars re-entry. The x-axis displays the* 800 *simulation nodes.*

Figure ***24*** reveals an astonishing leap in Mars entry precision-Starship's ability to navigate to within a mere $3\ km$ radius of its target at $45\ km$ altitude represents an achievement that would have seemed impossible during earlier Mars exploration eras. This level of accuracy, visualized by the clustering of potential landing points within the blue circular boundary, dramatically surpasses the capabilities of previous missions like Viking and Pathfinder, whose landing ellipses spanned tens of kilometers. Even the Perseverance rover, employing NASA's most advanced entry guidance technology prior to Starship, achieved a landing accuracy of approximately $6 - 7\ km$ only at the final touchdown – not during mid-atmospheric flight. Starship accomplishes this feat through sophisticated trajectory optimization algorithms and hypersonic guided entry techniques that continuously adjust the vehicle's bank angle and lift vector, allowing it to carve a precise path through Mars' tenuous atmosphere.

This unprecedented navigational precision at such significant altitude transforms the paradigm for human Mars exploration, opening possibilities for coordinated landings near pre-deployed infrastructure and resource-rich regions. While robotic missions historically targeted landing accuracy of $\sim 10\ km$ and human-rated requirements specified a threshold of $5\ km$, Starship's demonstrated 3-kilometer error at $45\ km$ altitude – with the opportunity for further refinement during final descent – enables mission architects to select landing sites with geological or strategic significance rather than merely those offering broad, hazard-free zones. Starship's position, represented by each potential landing point in *Figure* ***24***, heralds a future where astronauts need not fear being stranded kilometers from vital resources or habitats, but can instead trust the re-entry guidance system to deliver them with precision to the predetermined location.

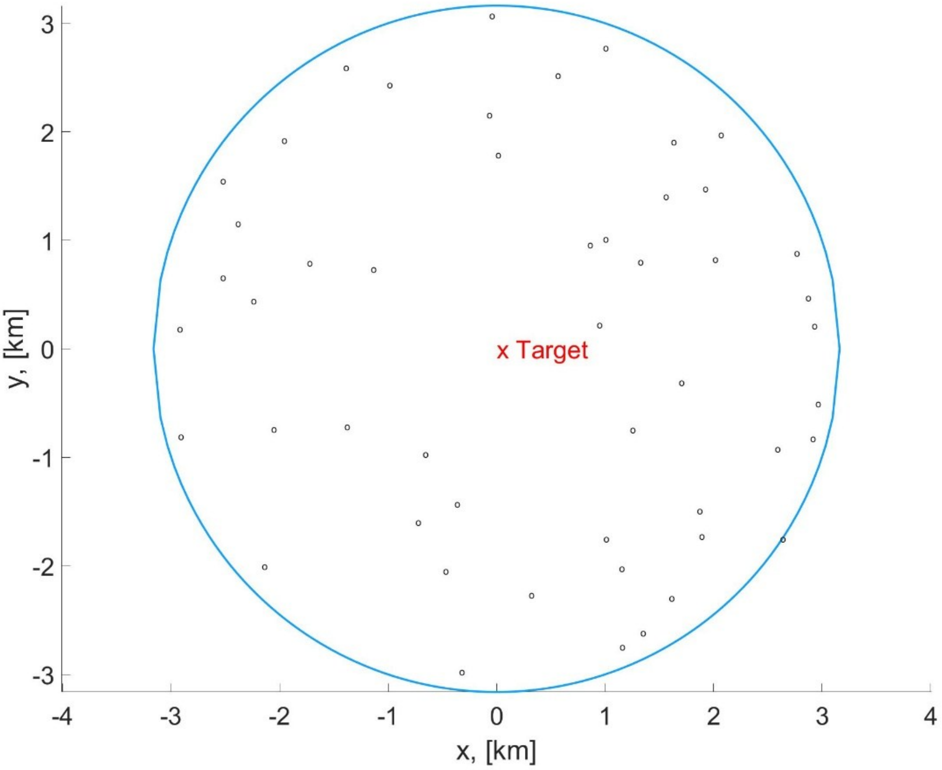


Figure 24: *Starship's $3\ km$ error at $45\ km$ altitude above the surface of the planet, achieved by employing the real-time successive convexification algorithm, described in Chapter 4. The x-axis displays the 800 simulation nodes.*

As it can be seen in *Figure* ***25*** and *Figure* ***26***, all three constraints – heat flux, g-load and dynamic pressure are being tested at $\sim 60\ km$ altitude, a critical point towards the end of the simulation. Despite that, all physical constraints are satisfied, keeping numbers below maxima, and delivering the vehicle at less than $3\ km$ position error around the $45\ km$ altitude target.

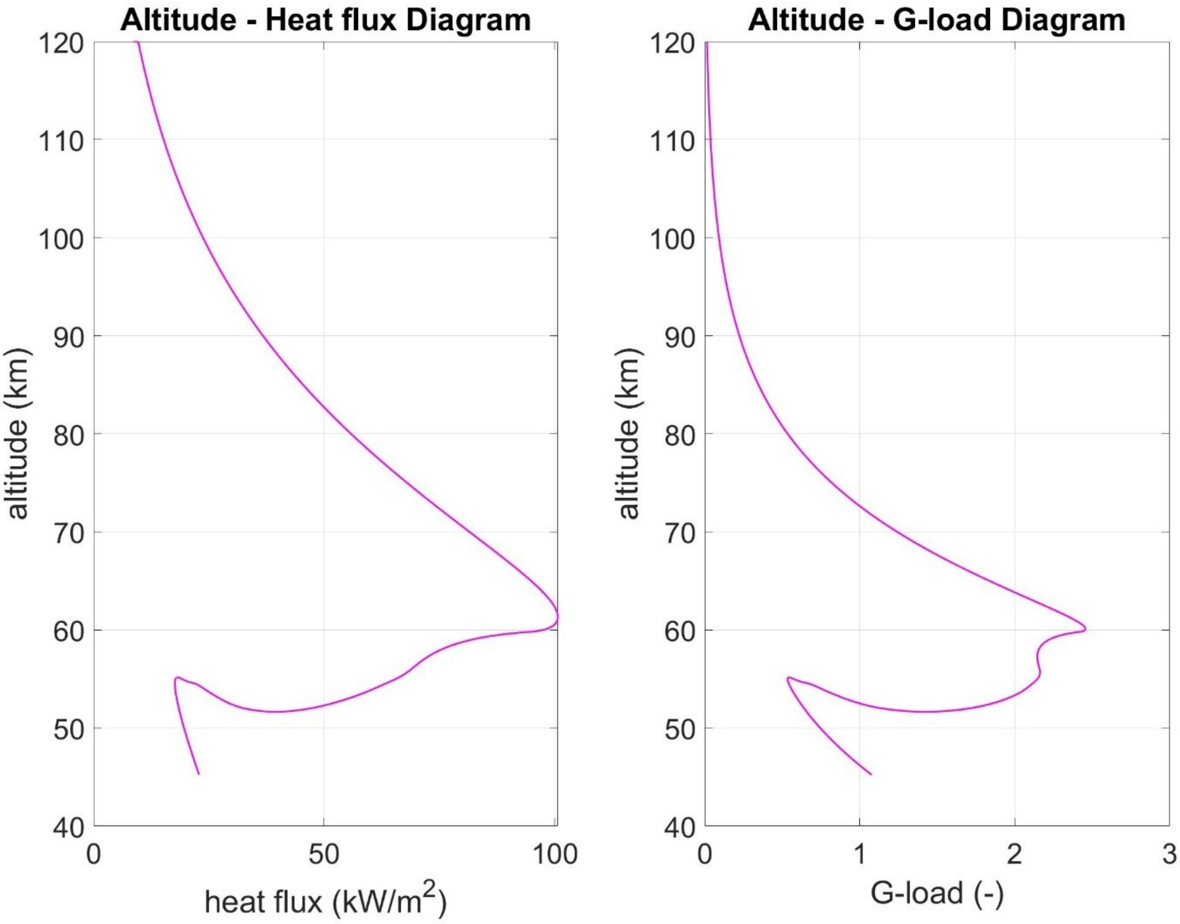


Figure 25: *Heat flux and G-load versus Altitude diagrams showing peak values at* $\sim 60\ km$

Other characteristics of the guided hypersonic glide

The flight-path angle profile reveals Starship's sophisticated ballistic management strategy, beginning with a deliberately steep $\gamma = -15^o$ descent angle that gradually transitions to a near-horizontal glide by simulation node 450. This initial steepness serves multiple critical functions: it expedites atmospheric interface, minimizes orbital energy during early entry, and establishes a predictable deceleration regime. As depicted in the flight-path graph, the trajectory deliberately levels out to $\sim 0^o$ during the middle phase – effectively transforming the spacecraft into a hypersonic glider that maximizes atmospheric interaction

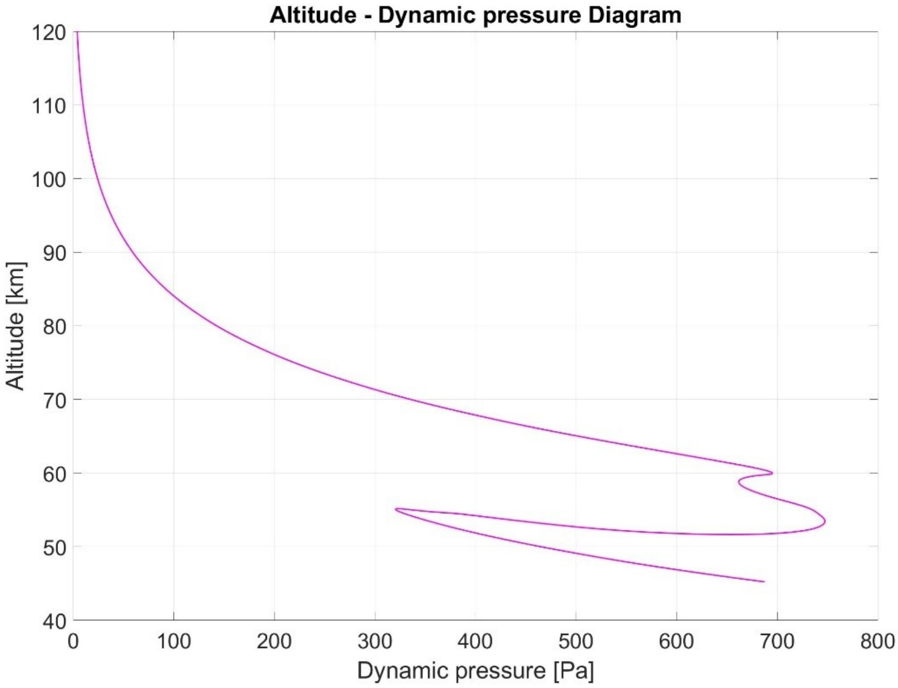


Figure 26: *Dynamic pressure as a function of altitude, showing peak values* $\sim 50 - 60\ km$

while slowly losing altitude (the equilibrium glide phase). The final negative plunge to -9^{o} near node 800 represents a calculated compromise between maintaining enough velocity for aerodynamic control and ensuring obeying the equilibrium glide entry corridor constraint.

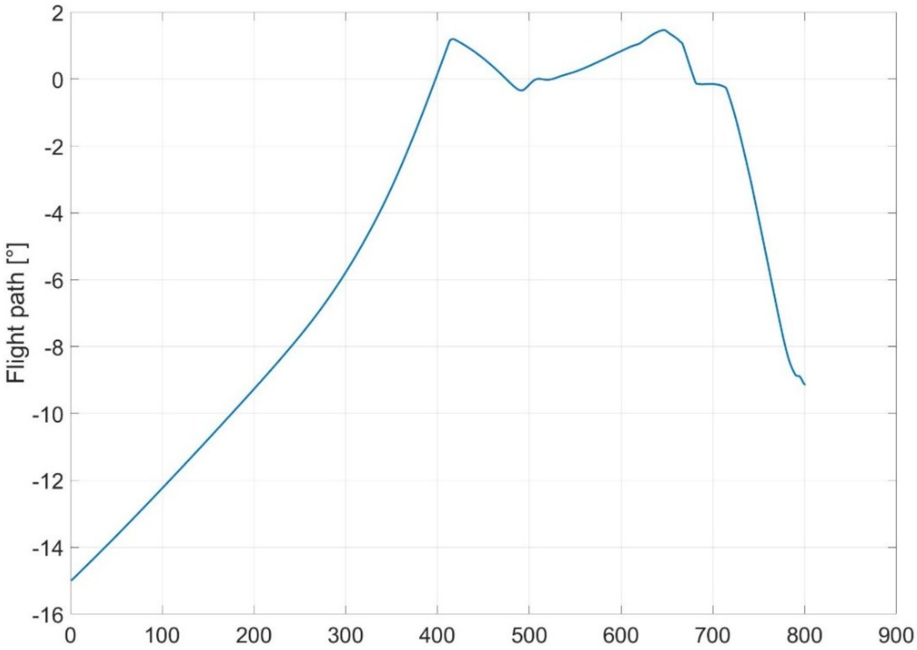


Figure 27: *Flight-path angle (the angle between the vehicle's velocity* V *vector and the local horizontal) for the Starship Mars re-entry. The x-axis displays the* 800 *simulation nodes.*

The angle-of-attack graph in *Figure **28*** illustrates Starship's pre-arranged initial attitude, starting with the substantial $52-$ degree angle that remains almost constant until node 300. This high-alpha configuration generates tremendous drag during initial entry when the vehicle is most vulnerable to thermal extremes, creating a pressure distribution that shields the leeward surfaces while stabilizing the craft against Martian turbulence. The gradual reduction to $\alpha = 34^o$ by simulation's end reflects Starship's adaptive response to thinning atmosphere and decreasing velocity – a precision-tuned balance between lift generation and drag management that prevents excessive oscillations while maintaining directional control. This careful modulation of aerodynamic attitude provides the guidance algorithm with the control authority necessary for trajectory refinement without demanding excessive RCS thruster usage, just quasi-wings at the tip and bottom of the vehicle.

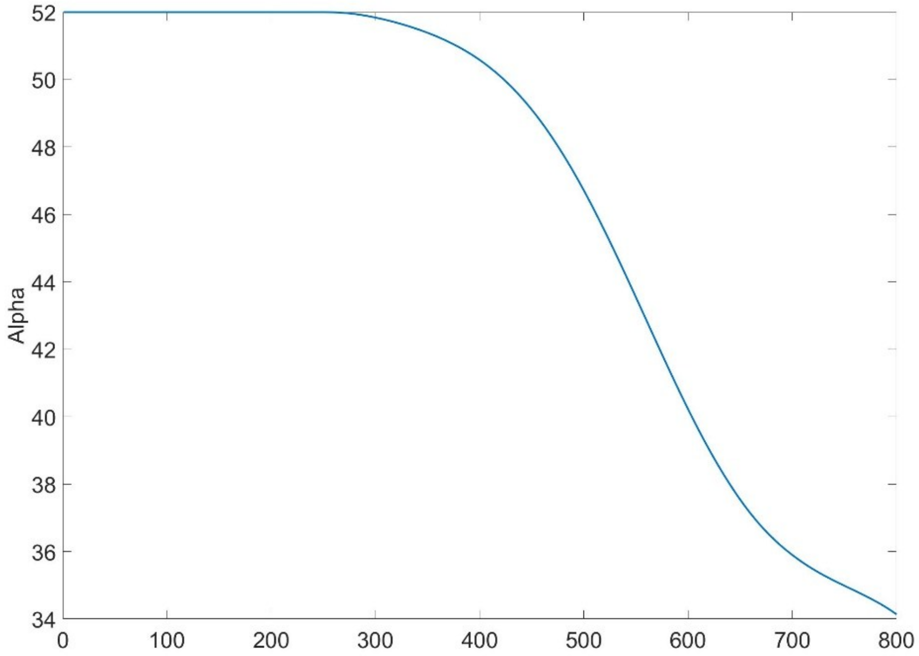


Figure 28: *Starship's angle-of-attack and its gradual decrease through the hypersonic glide phase. The x-axis displays the* 800 *simulation nodes.*

The velocity profile (shown in *Figure **29***) completes this aerodynamic symphony, showing an initial counterintuitive increase to $4400\ m/s$ around node 220 – a phenomenon resulting from Mars' gravitational acceleration briefly overpowering modest atmospheric resistance at high altitudes. The subsequent dramatic deceleration between nodes $300 - 600$ reveals where Starship transitions from a ballistic entity to an aerodynamic vessel, shedding nearly $2400\ m/s$ within this critical phase. This precisely controlled energy dissipation, synchronized with the flight-path and angle-of-attack adjustments (discussed previously and shown in *Figure **27*** and *Figure **28***), enables Starship's unprecedented $3\ km$ landing accuracy at $45\ km$ altitude – a feat unimaginable with previous Mars entry systems. The gentle velocity curve flattening after node 700 demonstrates the guidance system's intelligent preservation of sufficient kinetic energy for the near-final approach, ensuring adequate control authority while preventing excessive speeds that would compromise targeting precision. In *Figure **30***, a 3D atmospheric trajectory with 5 iteration SOCP steps is shown. The graph exhibits the splines from blue to red, approaching the initially specified target.

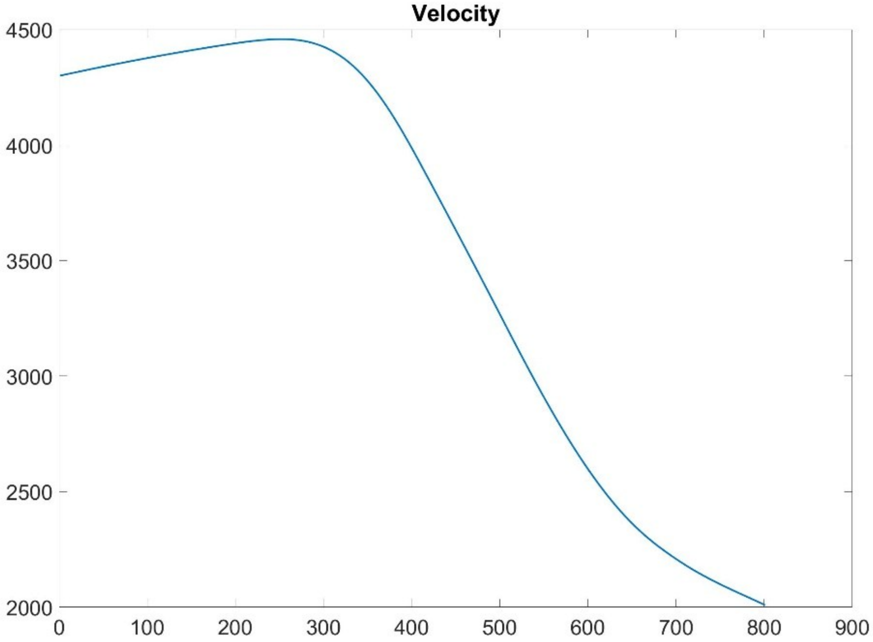


Figure 29: *Velocity profile diagram for the Starship Mars re-entry. The x-axis displays the* 800 *simulation nodes.*

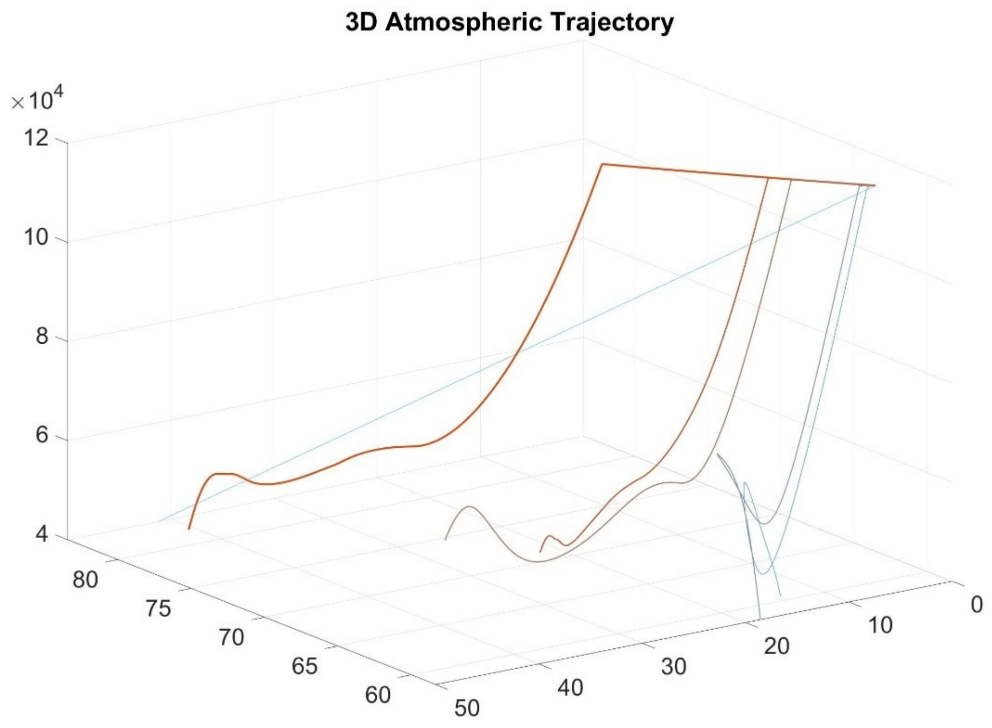


Figure 30: *3D atmospheric trajectory, based on five SOCP iteration steps, based on the algorithm described in Chapter 5: Algorithm Methodology*

8

Sensitivity Analysis

8.1 Sensitivity Variation in Entry Velocity

The sensitivity analysis of Starship's re-entry velocity on Mars reveals a good adherence to the entry corridor even when some of the initial variables fluctuate. When entry velocity fluctuates by $\pm 10\%$ around the nominal $4.3\ km/s$ (from $3.9\ km/s$ to $4.7\ km/s$), the vehicle's trajectory remains confined within the narrow entry corridor, as illustrated in *Figure* ***31***, Left. This corridor – bounded by skip-out and ballistic failure limits – is maintained through adaptive bank angle profiles (*Figure* ***31***, Right), which modulate lift-to-drag ratios to counteract velocity perturbations. Even at the extremes of the tested range, the guidance system adjusts bank angles dynamically, prioritizing altitude control during high-speed entries and drag management during slower approaches. This flexibility ensures that Starship avoids both excessive atmospheric skip and catastrophic over-penetration, a testament to the robustness of its aerodynamic design.

Figure ***32*** quantifies the consequences of velocity variations on critical re-entry parameters. Heat flux (*Figure* ***32***, Top Left) stays mostly within its limits, which is important because velocity variation is the factor that can most of all turn re-entry in the wrong direction. Yet, the thermal protection system demonstrates sufficient margin to accommodate these spikes, owing to its proven performance during Earth re-entries at

$12.9\ km/s$ (Simplicio, 2018). Dynamic pressure (*Figure **32***, Bottom) follows a similar trend, though its peak values remain constrained by the vehicle's ability to maintain higher altitudes during fast entries. G-loads (*Figure **32***, Top Right) prove the most velocity-sensitive metric, with maximum accelerations reaching $2.7g$ at $+10\%$ velocity. However, these transient forces stay within human tolerance thresholds, as the guidance system prioritizes gradual deceleration through extended upper-atmospheric flight.

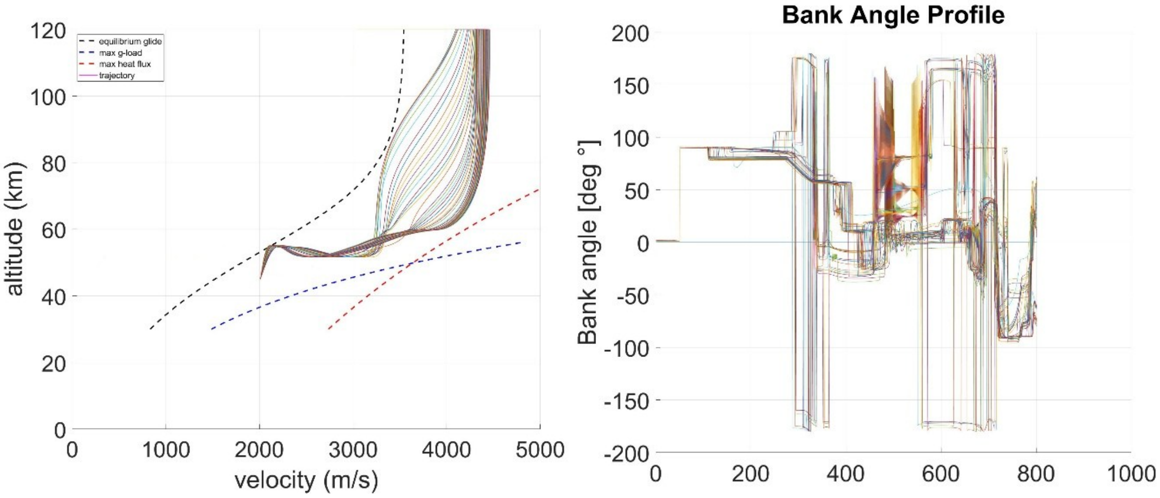


Figure 31: *Starship constrained trajectory within its re-entry corridor (Left), and vehicle's bank angle profile (Right), while varying entry velocity* $\pm10\%$ *around the nominal* $4.3\ km/s$. *The x-axis on the right displays the* 800 *simulation nodes.*

The bank angle profile emerges as the linchpin of this adaptive strategy. At elevated velocities, the algorithm commands steeper bank angles during initial entry to maximize lift vectoring (through the bank angle modulation), effectively trading horizontal velocity for altitude preservation. As speed decays, the profile transitions to shallower angles, prolonging exposure to denser atmospheric layers for controlled energy dissipation. This dual-phase approach – reminiscent of the "lift-up" maneuvers employed by Mars 2020 – ensures consistent landing accuracy despite entry velocity dispersions.

These findings underscore the proposed in this research algorithm could be useful of Starship's entry guidance architecture. By maintaining corridor adherence across velocity dispersions and autonomously optimizing bank profiles, the system compensates for uncertainties in atmospheric density and navigation errors, while obeying the re-entry corridor three physical constraints, displayed in *Figure **31***. While the $7.5\ km/s$ inertial entry velocity cited in literature _(Marco Sagliano, 2019) presents greater thermal challenges than the $4.3\ km/s$ case studied here; the underlying principles of adaptive control remain equally valid. This velocity sensitivity analysis provides a good validation that the algorithm holds, adhering to the thermal constraints, while varying the initial variables.

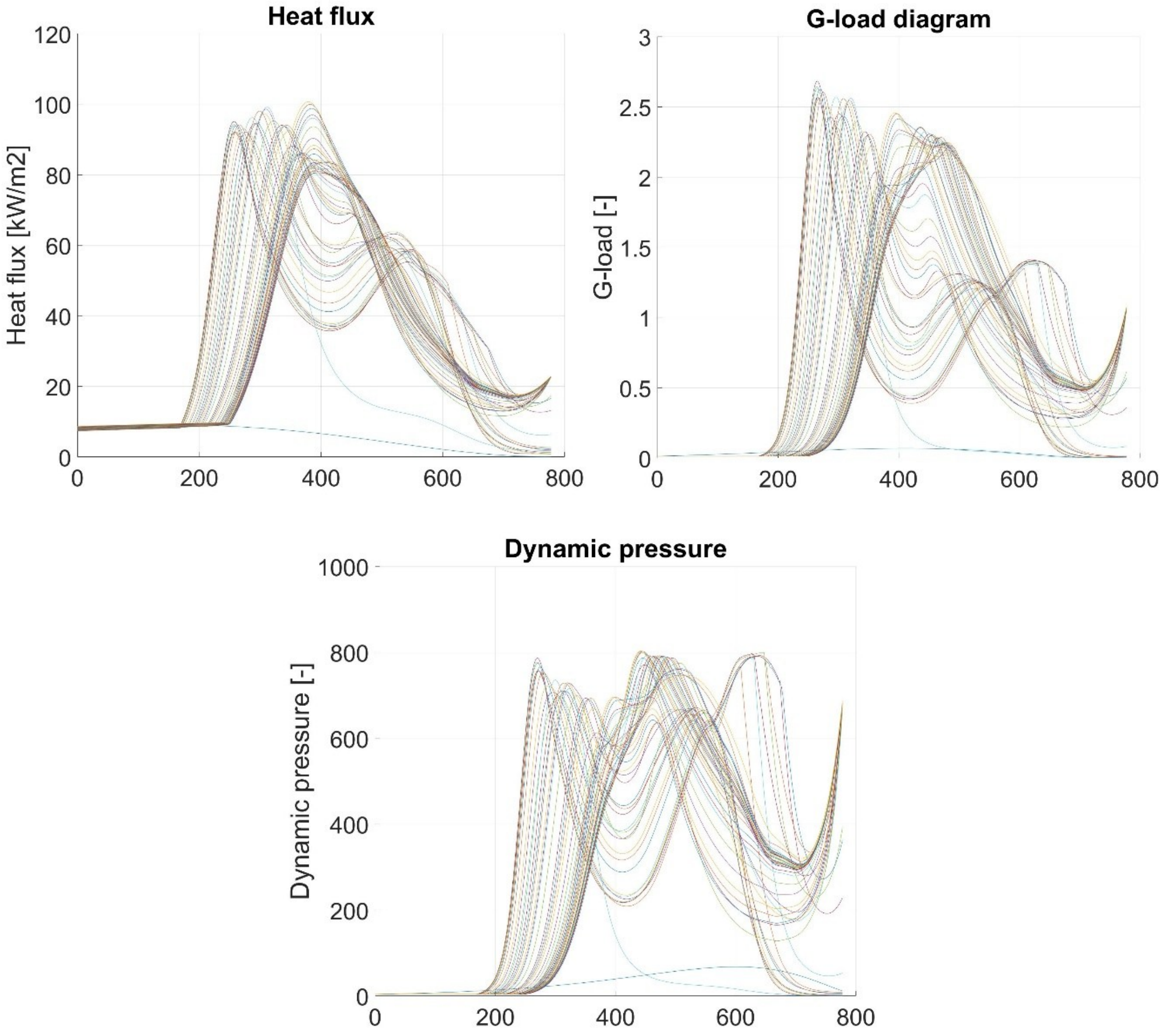


Figure 32: *Re-entry corridor physical constraints in place during Starship's Mars atmospheric entry while varying entry velocity* $\pm 10\%$ *around the nominal* $4.3\ km/s$*. The x-axis displays the* 800 *simulation nodes.*

8.2 Sensitivity Variation in Flight-Path Angle

The sensitivity analysis of Starship's entry flight-path angle during Mars re-entry reveals a balance between aerodynamic control, peak heat flux management, and g-load minimization. Variations of $\pm 10\%$ around the nominal -15 *degrees* velocity highlight the robustness of the guidance algorithm in maintaining trajectory stability while managing the complex interplay of Martian atmospheric dynamics. As illustrated in *Figure **33***, Left, the optimized trajectory remains tightly confined within the entry corridor. This corridor adherence ensures that the vehicle neither risks skipping out of the atmosphere nor plunges into excessively dense regions prematurely, both scenarios that could jeopardize mission success.

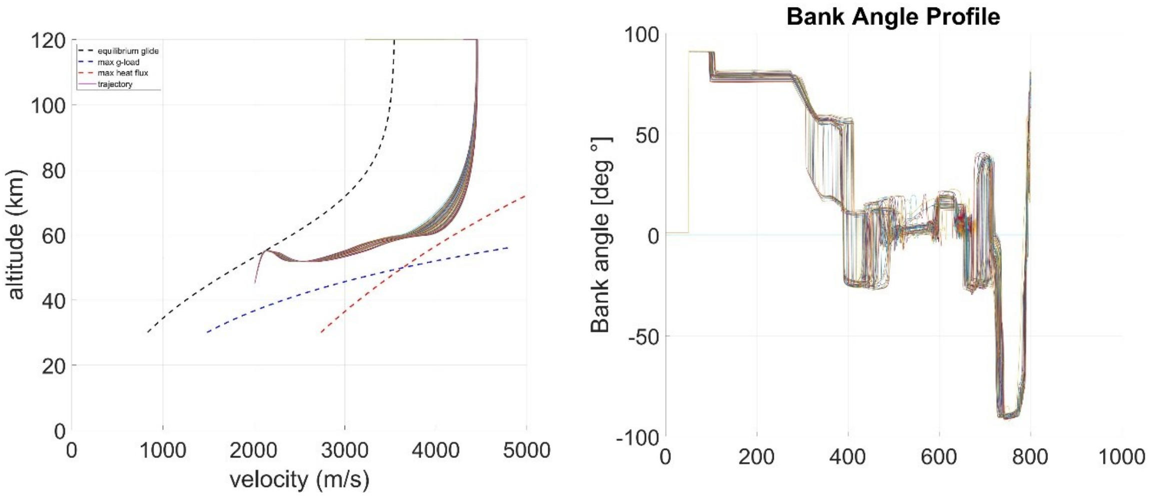


Figure 33: *Starship constrained trajectory within its re-entry corridor (Left), and vehicle's bank angle profile (Right), while varying vehicle's flight-path angle* $\pm 10\%$ *around the nominal* -15 *degrees. The x-axis on the right displays the* 800 *simulation nodes.*

The bank angle profile, relatively similar across simulations, consistently operates within predefined margins. Early phases employ higher bank angles to maximize lift modulation during high-velocity descent, followed by a gradual reduction as velocity decreases. This phased approach prolongs exposure to lower-altitude atmospheric drag, optimizing energy dissipation while preserving control authority. *Figure **33***, Right underscores this adaptability: even with flight-path angle dispersions, the bank reversals and vertical L/D adjustments neutralize crossrange errors, ensuring alignment with the target site.

Heat flux (*Figure **34***, Left) demonstrates a predictable rise and fall, peaking at velocities where convective heating dominates. The guidance algorithm's ability to modulate bank angle during the range control phase mitigates localized heating spikes by adjusting the rate of descent and redistributing thermal loads across the heatshield. Notably, the peak surface temperature remains below material limits and doesn't deviate significantly with varying flight-path angle, a critical factor given Mars' variable atmospheric density and its influence on boundary layer transition.

G-load profiles (*Figure **34***, Right) exhibit a bell-shaped curve, with maxima correlating to the initial case, without a significant variation. Dynamic pressure (*Figure **35***), a function of atmospheric density and velocity squared, peaks during the mid-phase of entry. Starship's trajectory optimization ensures that this peak remains within structural margins, avoiding excessive aerodynamic stress. The bank angle's role in modulating lift vectors is pivotal here: steeper angles increase drag, reducing velocity-and thus dynamic pressure – more rapidly, while shallower angles extend the deceleration timeline.

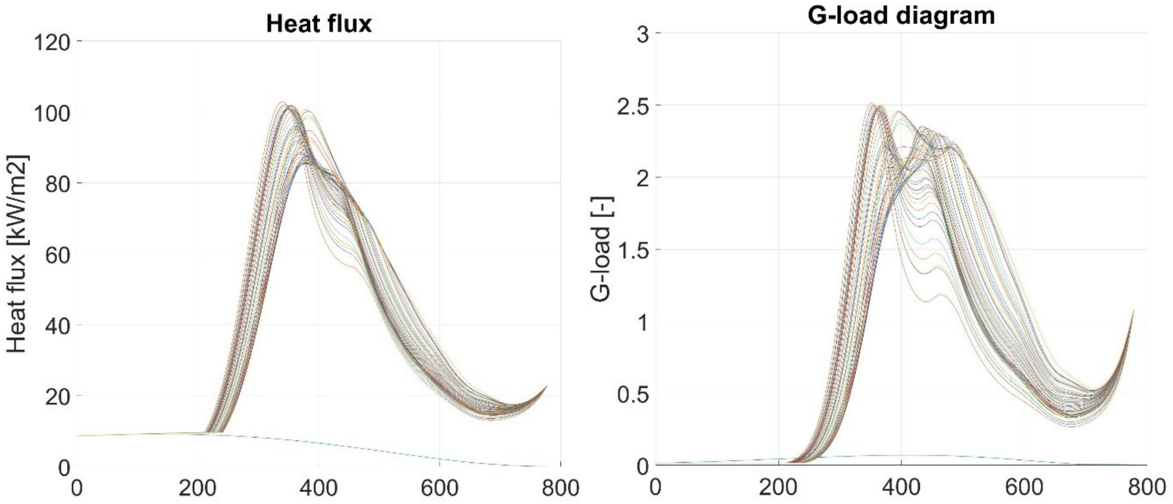


Figure 34: *Re-entry corridor physical heat flux and g-load constraints during Starship's Mars atmospheric entry while varying entry flight-path angle* $\pm 10\%$ *around the nominal* -15 *degrees. The x-axis displays the* 800 *simulation nodes.*

In summary, the sensitivity analysis underscores the resilience of Starship's guidance architecture when varying initial entry flight-path angle. The vehicle shows consistent adherence to the already established trajectory and bank angle profile, while keeping constrained within the entry corridor. All physical constraints peak at their respective levels without any structural and thermal endanger to the vehicle.

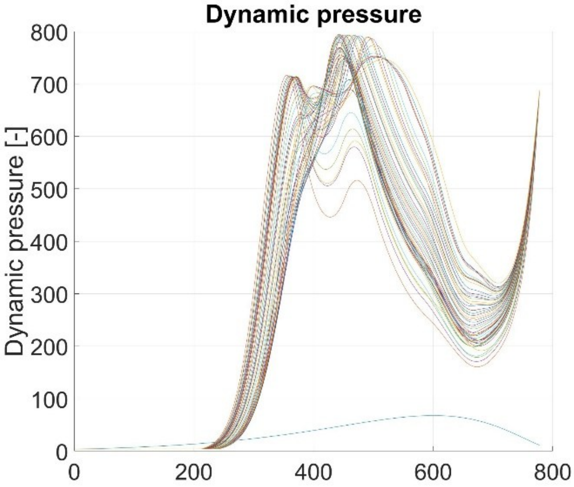


Figure 35: *Re-entry corridor physical dynamic pressure constraint in place during Starship's Mars atmospheric entry while varying entry flight-path angle* $\pm 10\%$ *around the nominal* -15 *degrees. The x-axis displays the* 800 *simulation nodes.*

8.3 Sensitivity Variation in Atmospheric Density

The environment uncertainty chosen to be analyzed from the environment is the atmospheric density uncertainty as logically when the spacecraft glides hypersonically through Mars' atmosphere, the atmospheric density dispersions have the highest effects on its flight-path towards the denser parts of the atmosphere at $45\ km$. In reality, there is an approximately exponential structure in the atmospheric density with decreasing altitude from $120\ km$. The atmospheric model considered here as a base is the exponential model, used for the base case. The atmospheric density in the model will be modified with a scaling factor in order to investigate the uncertainty induced over the constraints and objectives. The distribution of the scaling factor used is log-normal with $\mu = 0.3$ and $\sigma = 0.12$. A Monte Carlo simulation with 200 realizations is used to generate a random sample of numbers for the scaling factor by a Mersenne Twister core random number generator (Wikipedia, n.d.), with an initialization number of the seed set to 99. The investigated constraints and objectives are the heat flux, g-load and total heat load on the vehicle's stagnation point (called point A in *Figure **36***, *Figure **37*** and *Figure **38***).

For the stagnation point A, the highest probability $(\sim 50\%)$ in heat flux change is in the middle of the design space (*Figure **36***, Left) which is related to an increase in atmospheric density between 20% and 50%.

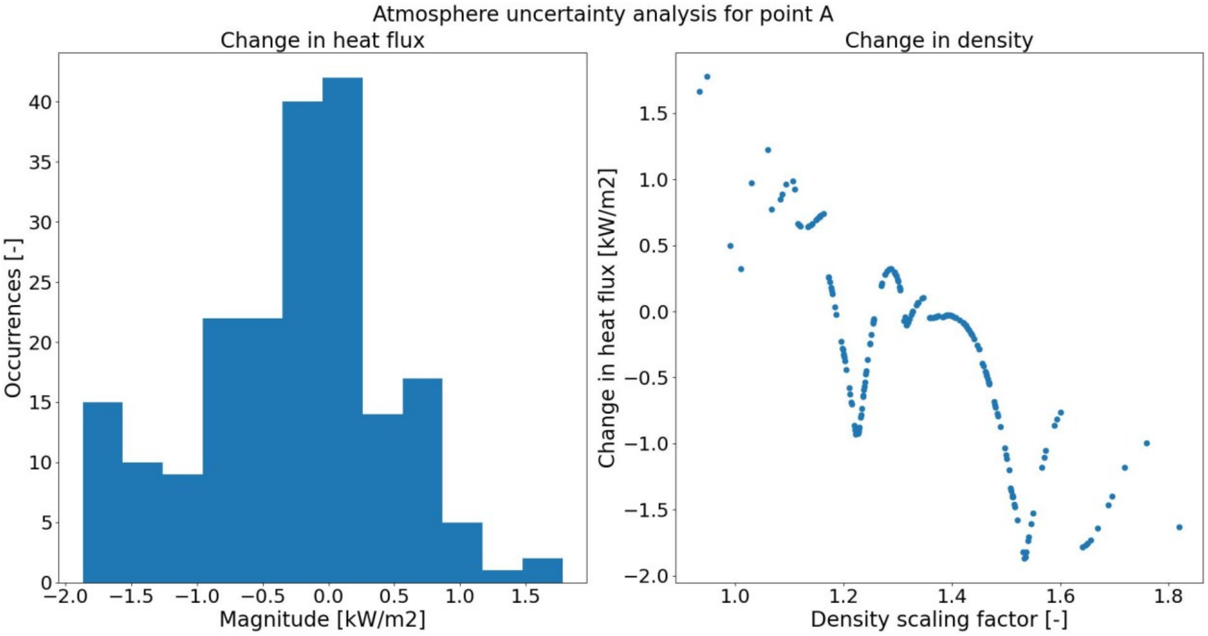


Figure 36: *Atmospheric density uncertainty analysis for the heat flux at the stagnation point*

For the total heat load Q_c, being a function of the integrated heat flux for the duration of the flight, the same pattern as above is observed, namely, the uncertainty increases at the stagnation point (*Figure* ***37***). It is worth mentioning the relation between the density scaling factor and the change in Q_c. With increasing atmospheric density, the rate of change in Q_c increases linearly (*Figure* ***37***). From the histogram, it can be seen that the magnitude of the

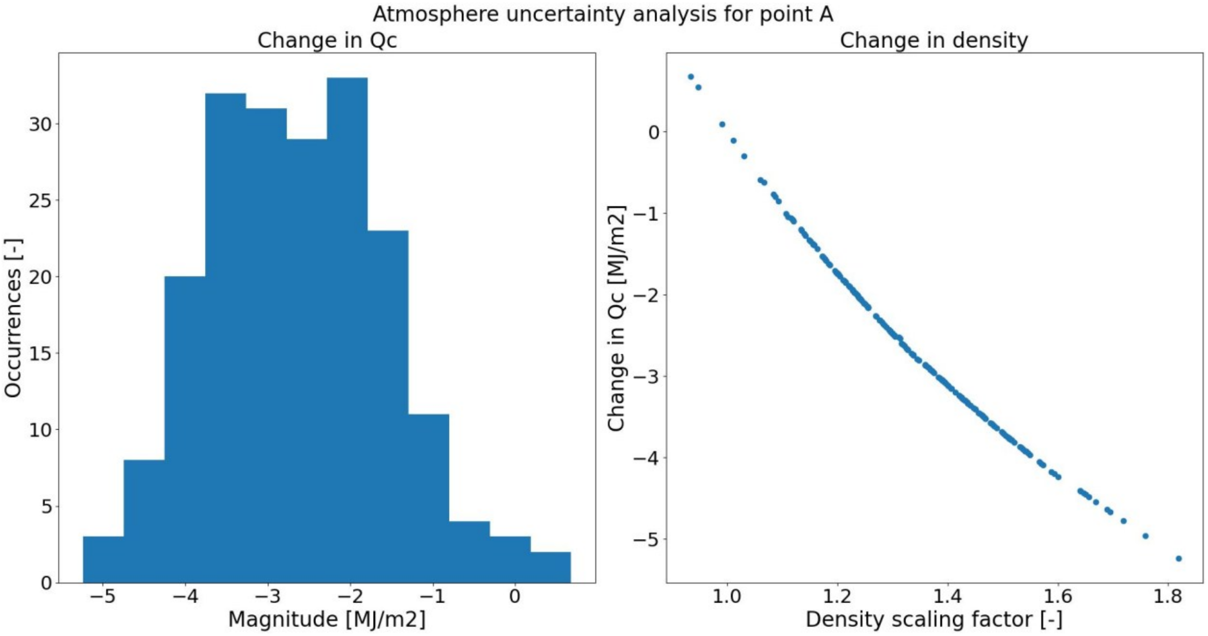


Figure 37: *Atmospheric density uncertainty analysis for the total heat load at the stag. point*

heat load changes between 6 and $12\ MJ/m^2$ with an 80% probability, which corresponds to an increase in atmospheric density by around $20 - 60\%$ (*Figure **37***, Right).

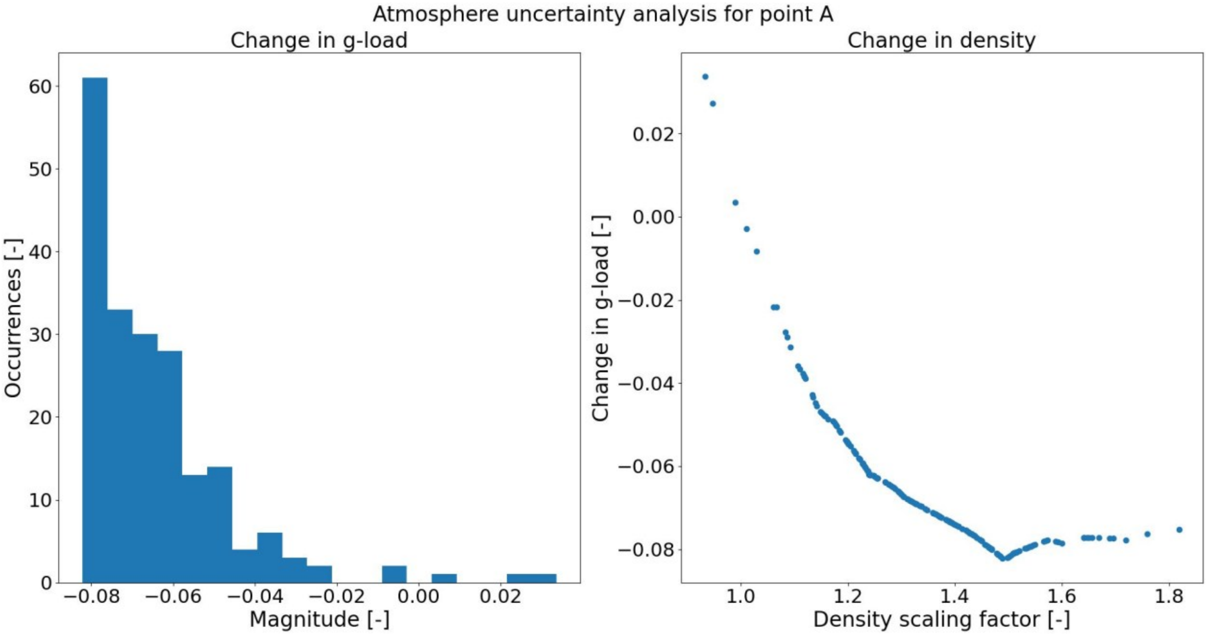


***Figure 38:** Atmospheric density uncertainty analysis for the g-load at the stagnation point*

The effect on the g-load with changing atmospheric density is different foe the stagnation point, than any other point on the spacecraft. Even with a density scaling factor of 1.8, the impact on the g-load is very small: -1.6% or $0.08g$ (*Figure **38***).

9

Conclusions and Recommendations for Future Research

9.1 Research Conclusions

The investigation into Starship's hypersonic glide re-entry into the Martian atmosphere, initiated from a parking orbit at $4.3\,km/s$ with a $-15°$ flight-path angle, underscores the viability of advanced guidance algorithms in navigating one of the most aerodynamically precarious phases of interplanetary missions. By confining simulations to the $120 - 45\ km$ altitude band – a regime where atmospheric density gradients are minimal yet dispersive effects pose significant trajectory uncertainties – this study illuminates the critical role of real-time adaptive control in preserving mission safety and precision.

Central to these findings is the success of the successive convexification-based guidance algorithm, which demonstrates remarkable robustness in steering the vehicle through the tenuous upper atmosphere. Despite the inherent challenges of Martian atmospheric variability, the algorithm ensures trajectory adherence within the entry corridor, culminating in a terminal altitude $(45\ km)$ positioning error below $3\ km$ (as seen in *Figure **24***). This precision, achieved while rigorously enforcing heat flux $(< 100\ kW/m^2)$, g-load $(< 3g)$, and dynamic pressure $(< 800\ Pa)$ constraints, aligns closely with historical data from

NASA's Mars missions, validating the fidelity of the physical and aerodynamic models employed. This essentially answers the core research question, stated in Chapter 2.8.

The dual-axis control architecture – bank angle modulation for lift vectoring and angle-of-attack adjustments for thermal management – emerges as a cornerstone of the re-entry strategy. Bank angle variations $(\text{from } -90° \text{ to } +90°)$ optimally distribute lift forces to counteract atmospheric perturbations, while angle-of-attack refinements mitigate localized heating spikes, cumulatively limiting total heat load accumulation. This synergistic approach not only stabilizes the trajectory but also ensures that thermal and structural margins remain intact, even during off-nominal dispersions in initial velocity or flight-path angle $(\pm 10\%)$. The above description answers RSQ - 1.

Sensitivity analyses further corroborate the system's resilience. Atmospheric density fluctuations, a well-documented feature of Mars' volatile climate, induce predictable trajectory deviations that the guidance logic compensates for within seconds. Similarly, the algorithm's rapid convex optimizations-executed internal methods iteratively neutralize errors introduced by initial condition uncertainties $(V_0 \text{ and } \gamma)$ and real-time perturbations, ensuring that the vehicle remains on course (within the re-entry corridor), despite disturbances that could otherwise compound catastrophically in lower, denser atmospheric layers. The above description answers RSQ - 3. The results, presented in Chapter 7 and Chapter 8 corroborate also that the physical constraints that determine the entry corridor of the vehicle, while it enters through the atmosphere, are completely satisfied, without compromising the $sub - 3\ km$ precision, needed to land humans on the surface of the planet. Moreover, the results and sensitivity analysis exhibit the robustness of the algorithm as the resulting entry trajectories never violate these hard constraints, putting people in potential danger. The above description answers RSQ – 2.

These results carry promising implications for future crewed and cargo missions to Mars. By demonstrating that Starship can autonomously navigate the upper atmospheric regime with $sub - 3\ km$ precision, this study presents convex optimization, with its implementation of the PTR successive convexification, as a viable option to produce meaningful results for Mars re-entry. The convex optimization paradigm, in particular, showcases its superiority over traditional predictor-corrector methods (as discussed in

Chapter 1 in handling nonlinear dynamics and path constraints, suggesting broader applicability in extraterrestrial landing systems.

Looking ahead, the transition from hypersonic glide to terminal descent – a phase beyond $45\ km$ altitude – will demand tighter integration of retro-propulsion during Starship's belly – flop maneuver, and possibly on-board Terrain Relative Navigation (TRN). However, the foundation laid here ensures that Starship arrives at this handover point with sufficient energy reserves and positional accuracy to execute subsequent maneuvers reliably, as already presented in the Results (Chapter 7). In this context, this study broadens the common body knowledge of Mars re-entry research, filling the research gap for the initial conditions, putting the spacecraft in an elliptical orbit around the planet, before entering the atmosphere.

9.2 Recommendations for Future Research

The insights gleaned from this investigation into Starship's hypersonic Martian re-entry glide, while comprehensive, illuminate several pathways to deepen our understanding of interplanetary entry guidance and expand the operational envelope of future missions. Building upon the validated success of convex optimization-based algorithms in the $120 - 45\ km$ altitude regime, the following recommendations propose targeted advancements to address unresolved challenges and prepare for increasingly complex mission profiles.

1. Extended Simulation Scope to Surface Impact

The current study terminates at $45\ km$ altitude, a prudent cutoff given the atmospheric dispersions prevalent in higher altitudes. However, extending simulations through the denser lower atmosphere – down to powered descent initiation – would provide a holistic view of the entire Entry – Descent – Landing (EDL) sequence. Such work could integrate stochastic models of dust storms, localized pressure anomalies, and terrain-induced wind shear, all of which compound uncertainties below $45\ km$. Coupling the existing guidance algorithm with Terrain Relative Navigation (TRN) and closed-loop retro-propulsion logic would test the handover fidelity between hypersonic glide and terminal landing phases, ensuring seamless transitions in dynamic environments.

2. High-Fidelity Atmospheric Modeling

While the exponential model used in this research suffices for baseline analyses, incorporating the Naval Research Laboratory Mars (NRLMSISE-00) model with spherical harmonic gravity expansions would better capture latitudinal and seasonal atmospheric variations. Mars' methane plumes, dust cycles, and polar CO_2 sublimation induce transient density fluctuations that current zonal-mean models underrepresent. A spherical harmonics-based approach, weighted by recent MAVEN and Perseverance rover data, could resolve localized density gradients, enabling scenario-specific guidance tuning. This would particularly benefit missions targeting highland regions or polar ice caps, where atmospheric behavior diverges markedly from equatorial norms where humans will most probably first land.

3. Interplanetary Velocity and Payload Scaling

Testing entry velocities of $7.5\ km/s$ – representative of interplanetary trajectories from Earth to Mars or cargo missions from outer solar system waypoints – would stress-test the thermal protection system and guidance responsiveness. Concurrently, evaluating higher payload masses (150 – 200 tons) could reveal latent instabilities in bank angle authority or angle-of-attack hysteresis during high dynamic pressure regimes. Such studies would inform TPS scaling laws and control surface redesigns, ensuring Starship's adaptability to diverse mission architectures, including crewed expeditions or bulk resource transport.

3. Aerodynamic Model Refinement via Flight Data

Present aerodynamic coefficients, though anchored in computational fluid dynamics (CFD) and wind tunnel data, remain provisional. Post-flight telemetry from upcoming Starship orbital re-entries – particularly heating distributions, boundary layer transition points, and hypersonic pitch damping – should be assimilated into the simulation framework. A Bayesian model-updating approach could reduce epistemic uncertainties in lift-to-drag ratios and moment coefficients, especially during rarefication-dominated flows. This empirical refinement is critical for Mars, where CO_2-dominated gas chemistry may induce non-equilibrium aerodynamic effects absent in Earth-based testing.

5. Multi-Disciplinary Optimization Frameworks

The current guidance strategy prioritizes trajectory precision and constraint adherence. Future work could embed the entry problem within a multi-disciplinary optimization (MDO) framework, simultaneously minimizing heat load, maximizing cross range capability if needed, and preserving propellant reserves for divert maneuvers. Adjoint-based sensitivity analyses could identify Pareto-optimal solutions for conflicting objectives, such as minimizing peak heating while maximizing downrange reach – a balance vital for cargo deployments across scattered landing zones.

5. Human-Centric Adaptation

For crewed missions, extending the algorithm to prioritize physiological constraints – such as limiting cumulative g-load exposure or optimizing window orientations for radiation shielding – would enhance occupant safety. Integrating real-time health monitoring systems (e.g., vestibular stress predictors) could dynamically adjust bank reversals to mitigate motion sickness during oscillatory phugoid motions, a consideration absent in robotic missions.

6. AI / ML – Augmented Guidance

While convexification ensures computational tractability, machine learning surrogates trained on high-fidelity simulation databases could accelerate onboard trajectory updates. Neural networks predicting optimal bank angle sequences, trained offline via reinforcement learning, might complement the convex solver during high-frequency disturbances (e.g., dust-driven density spikes). Hybrid architectures marrying physics-based models with data-driven corrections could enhance robustness in uncharted atmospheric regimes.

In pursuing these directions, researchers will not only refine Starship's Martian entry capabilities but also advance the broader field of planetary entry system design. Each recommendation, from atmospheric modeling to human factors integration, represents a stepping stone toward realizing Elon Musk's vision of a multiplanetary civilization – one where precision, adaptability, and safety converge to tame the cosmic frontier.

Acknowledgements

The year 2015 marked a pivotal turning point, a moment where my abstract fascination with the cosmos transformed into a tangible, burning ambition. Attending the International Space University's Space Studies Program in the USA was an immersive dive into the heart of the world's space endeavors. Visiting iconic NASA centers, standing in the very rooms where humanity charted its course to the stars, and having the privilege of meeting astronauts who had walked on the lunar surface – figures who have written their names into history and personal heroes – alongside the then-NASA administrator, ignited a spark that would redefine my professional trajectory. The sheer audacity of human ingenuity, the collaborative spirit driving great achievements, and the boundless potential of space exploration resonated deeply within me, compelling me to seek a more direct and impactful role in this grand adventure.

Yet, amidst the awe and wonder, a clear understanding crystalized for me: to contribute meaningfully to cutting-edge rocketry and space exploration within leading private companies like SpaceX or esteemed government entities such as the Jet Propulsion Laboratory, a robust, professional academic foundation was indispensable. The path forward required a rigorous formal education in engineering. Embarking on this necessary step proved to be an odyssey of its own. Achieving the prerequisites for admission to the prestigious Technical University of Delft (TUDelft) was a five-year journey punctuated by demanding preparation and unwavering focus. This period included sitting the arduous Graduate Record Examination (GRE), for which I dedicated a full year to solitary, intense study – a test of endurance and discipline that pushed my limits, as well as graduating with not one but two MS degrees in Bulgaria – in Astronomy & Engineering – to fulfill the admission prerequisites.

While charting this challenging academic course, another dream was simultaneously taking shape closer to home. Driven by the belief that the wonder of space should be accessible to all, and perhaps fueled by the long wait for my own academic journey to truly begin, I founded the first dedicated space exploration school in Bulgaria: the Space Academy. Developing a deep, insightful, and engaging curriculum for children and youngsters became a passionate pursuit, and the joy of successfully teaching them about the universe and the mechanics of space travel added immense meaning to these years. It was a hands-on contribution to building the next generation of space enthusiasts, engineers, and explorers, running in parallel with my own foundational studies.

Entering the application process for TUDelft was initially almost a whimsical challenge to myself – "let's see how far this goes." However, what began as an experiment quickly transformed into a demanding reality. The rigors of a Master's program in Aerospace Engineering necessitated an arduous, busy schedule of studying complex subjects and taking challenging exams. It required a significant mental shift and a deep commitment of time and energy.

Then, fate intervened in an utterly unprecedented manner. The global pandemic, a period of immense difficulty and uncertainty for the world, paradoxically bestowed upon me a form of once-in-a-century fortune regarding my studies. The necessity of remote learning meant that I didn't have to make the immediate, definitive move to Delft. From the comfort of my home, I could simultaneously pursue my studies at TUDelft and continue to develop and run the Space Academy. The long quarantine days, otherwise isolating, became dedicated to immersing myself in the intricate world of aerospace engineering, balancing lectures and assignments with the demands of running the academy. It was a uniquely challenging, yet ultimately fortunate, convergence of circumstances that allowed me to pursue both passions concurrently.

Reflecting on this extended adventure – the five years of preparation culminating in the Master's program itself – it is easy to see moments where the path could have been smoother, decisions perhaps more streamlined. There were certainly challenges, moments of doubt, and a constant juggle of priorities. Yet, the wealth of knowledge gained in space engineering is immeasurable, providing the bedrock I sought back in 2015. I also had the opportunity, albeit modified by circumstances, to experience the academic Dutch culture in

Delft. And critically, I managed to keep the Space Academy alive and thriving throughout this demanding period – a feat I sometimes doubted was possible, often fearing I would be forced to choose between these two consuming passions.

Such a journey, fraught with challenges, demanding unwavering dedication, and filled with unexpected turns, is never undertaken alone. I would like to thank my colleagues Andreas, Emree and Ludovica, who were the only friends I had during the arduous study periods. I also would like to thank my colleague Jubin, with whom I had two convex optimization discussions and that helped me better grasp the topic that I research in my thesis. Foremost among those who stood by me, offering unwavering support and boundless patience, is my beloved wife, Diana. She has been constantly next to me throughout this entire endeavor, persevering and remarkably keeping her cool through all the hectic moments – and they were plentiful. Her quiet strength, encouragement, and tangible help in countless ways were fundamental to my ability to navigate this path and achieve this success. We truly started this adventure together, supporting each other through its initial, uncertain stages.

And it is fitting, and fills our hearts with immense joy, that we finish it as a family of three. Our two-years-young boy Bryan, a vibrant bundle of energy and curiosity, is already a young future aspiring space enthusiast, astronomer, engineer, perhaps even a future astronaut. He fills our days with a new dimension of joy and meaning, a constant reminder of the future we are working to build, a future he will inherit and shape. This thesis is therefore not merely the culmination of academic effort, but a testament to a decade-long pursuit, fueled by inspiration, marked by perseverance, supported by boundless love, and now, shared with the next generation.

Tihomir Dimitrov
May 15, 2025

References

Academy, T. S. (2023, August 12). *YouTube*. Retrieved from https://youtu.be/twnZYPdFgbU

Acikmese, B. (2015). Constrained Reachability and Controllability Sets for Planetary Precision Landing via Convex Optimization.

Acikmese, B. (2022). Convex optimization for trajectory generation: A tutorial on generating dynamically feasible trajectories reliably and efficiently.

Bae, J. e. (2022). Convex Optimization-based Entry Guidance for Spaceplane.

Behcet Acikmese, L. B. (2007). Convex Programming Approach to Powered Descent Guidance for Mars Landing. *Journal of Guidance, Navigation and Dynamics, vol 30, no 5*, 1353 - 1366.

Behcet Acikmese, S. M. (2017). Flight Testing of Trajectories Computed by G-FOLD: Fuel Optimal Large Divert Guidance Algorithm for Planetary Landing. *23rd AAS / AIAA Space Flight Mechanics Meeting*.

Benito, J. (2010). Reachable and Controllable Sets for Planetary Entry and Landing.

Bjarne Westphal, V. M. (2022). Critical Analysis and Review of Current Mars Mission Scenarios for SpaceX Starship.

Blackmore, L. (2016). Autonomous Precision Landing of Space Rockets.

Boyd, S. (2004). *Convex Optimization.*

Cui, P. (2013). Real-time Navigation for Mars Final Approach Using X-Ray Pulsars.

Cui, P. (2017). Design and optimization of navigation and guidance techniques for Mars pinpoint landing: Review and prospect.

David Hash, H. A. (April - June 1993, Vol 7, No 2). Monte Carlo Simulation of Entry in the Martian Atmosphere. *Journal of Thermophysics and Heat Transfer*.

D'Souza, C. N. (New Orleans, LA, 1997). An Optimal Guidance Law for Planetary Landing. *AIAA Guidance, Navigation, and Control Conference*.

Elango, P. (2022). A Customized First-Order Solver for Real-Time A Customized First-Order Solver for Real-Time Powered-Descent Guidance.

Ingo Gerth, E. M. (2014). Guidance for Autonomous Precision Landing on Atmosphereless Bodies.

JPL, N. (2023, August 12). *Mars 2020 Spacecraft*. Retrieved from Mars 2020 Perseverance Rover Mission: https://mars.nasa.gov/mars2020/spacecraft/overview/

JPL, NASA. (2023, August 12). *Mariner 9 Spacecraft*. Retrieved from NASA Science Mars Exploration: https://www.jpl.nasa.gov/missions/mariner-9-mariner-i

Kamath, A. (2022). Real-Time Sequential Conic Optimization for Multi-Phase Rocket Landing Guidance.

Klumpp, A. R. (1974). Apollo Lunar Descent Guidance. *Automatica*, p 133 - 146.

Lawden, D. (London, Butterworths, 1963). Optimal Trajectories for Space Navigation. *Optimal Trajectories for Space Navigation*.

Liu, X. (2019). Fuel-Optimal Rocket Landing with Aerodynamic Controls.

Long, J. (2015). Controllable Set Analysis for Planetary Landing Under Model Uncertainties.

Lu, P. (2007). Predictor-Corrector Entry Guidance for Low Lifting Vehicles.

Malyuta, D. (2019). Discretization Performance and Accuracy Analysis for the Powered Descent Guidance Problem.

Marco Sagliano, A. H. (2019). Onboard Guidance for Reusable Rockets: Aerodynamic Descent and Powered Landing.

Mariner 4 Spacecraft. (2023). Retrieved from NASA Jet Propulsion Laboratory: https://www.jpl.nasa.gov/missions/mariner-4

Mars Global Surveyor Spacecraft. (2023, August 12). Retrieved from NASA Jet Propulsion Laboratory: https://mars.nasa.gov/mars-exploration/missions/mars-global-surveyor/

Math Overflow. (2024). *Can all convex optimization problems be solved in polynomial time?* Retrieved from Math Overflow: https://mathoverflow.net/questions/92939/can-all-convex-optimization-problems-be-solved-in-polynomial-time-using-interior

Meditch, J. S. (1964). On the Problem of Optimal Thrust Programming For a Lunar Soft Landing. *IEEE Transactions on Automatic Control, vol 9, no 4*, 477 - 484.

Meeowen, S. (2022). Hypersonic entry trajectory optimization via successive convexification with abstracted control.

Michael Szmuk, T. R. (2019). Successive Convexification for 6DoF Powered Descent Guidance with Compound State-Triggered Constraints.

Mooij, E. (2018). Optimal Drag-Energy Entry Guidance via Pseudospectral Convex Optimization.

Mooij, E. (2020). *Re-entry Systems.* Delft University of Technology.

NASA. (2023, August 12). *Enabling Pinpoint Landing on Mars*. Retrieved from NTRS.

NASA. (2023, August 12). *The Challenges of Landing on Mars*. Retrieved from The Challenges of Landing on Mars: https://www.nasa.gov/vision/universe/solarsystem/mars_challenges.html

NASA JPL. (2023). *Deep Space Network NOW*. Retrieved from Deep Space Network NOW.

NASA JPL. (2023, August 12). *Mars Pathfinder EDL*. Retrieved from Mars Pathfinder EDL: https://mars.nasa.gov/MPF/mpf/edl/edl1.html

NASA JPL. (2023, August 12). *NASA Science: Mars Exploration*. Retrieved from NASA Science: Mars Exploration: https://mars.nasa.gov/msl/spacecraft/instruments/medli/

NASA Lyndon JSC. (1979). *Space Shuttle Entry Guidance.* Mission Planning and Analysis Division.

Northrop Grumman. (2023). *LN-200 FOG Family Advanced Airborne IMU/AHRS*. Retrieved from LN-200 FOG Family Advanced Airborne IMU/AHRS: https://www.northropgrumman.com/what-we-do/air/ln-200-fog-family-advanced-airborne-imu-ahrs/

Northrop Grumman. (n.d.). *LN-200 FOG Family Advanced Airborne*. Retrieved from Northrop Grumman: https://www.northropgrumman.com/what-we-do/air/ln-200-fog-family-advanced-airborne-imu-ahrs

P. Lu, S. F. (Minneapolis, MN, 2012). A Versatile Powered Guidance Algorithm. *AIAA Guidance, Navigation, and Control (GNC) Conference*.

Pajola, M. (2019). Planetary mapping for landing sites selection: The Mars case study. *Planetary Cartography and GIS*.

Pei, P. (2021). Online Reentry Trajectory Optimization Using Modified Sequential Convex Programming for Hypersonic Vehicle.

Phys.Org. (2023). *Every challenge astronauts will face on a flight to Mars*. Retrieved from Every challenge astronauts will face on a flight to Mars.

Simplicio, P. (2018). Review of Guidance Techniques for Landing on Small Bodies.

Sippel, M. (2019). Assessment of multiple mission reusable launch vehicles. *Journal of Space Safety Engineering*.

Soumyo Sutta, D. W. (May 2024). Postflight Assessment of Mars 2020 Entry, Descent and Landing Simulation. *JSR | Spacecraft, Rockets*, Volume 61, Number 3.

Starship SpaceX Wiki. (n.d.). *Raptor 3*. Retrieved from Starship SpaceX Wiki: https://starship-spacex.fandom.com/wiki/Raptor_3

Taylor Reynolds, D. M. (2020). A Real-Time Algorithm for Non-Convex Powered Descent Guidance.

Taylor Reynolds, M. S. (2019). Dual Quaternion Based 6DoF Powered Descent Guidance with State-Triggered Constraints.

The Verge. (2021). Retrieved from The Verge: https://www.theverge.com/2021/10/21/22738106/spacex-boca-chica-starship-launch-site-faa-town-hall-environmental-assessment

Thilbault, C. (2019). 3DoF Simulation Model and Specific Aerodynamic Control Capabilities for a SpaceX's Starship-like Atmospheric Reentry Vehicle.

UTSA | The Univesity of Texas at San Antonio. (n.d.). *Relative Extrema and Convex Functions*. Retrieved from Math Research Wiki: https://mathresearch.utsa.edu/wiki/index.php?title=Relative_Extrema_and_Convex_Functions

Vikings 1&2 Spacecraft. (2023, August 12). Retrieved from NASA Jet Propulsion Laboratory: https://mars.nasa.gov/mars-exploration/missions/viking-1-2/

Vought, R. (2019). Commerce, Justice, Science, and Related Agencies Appropriations Act. *Executive Office of Management and Budget*.

Wan, C. (2022). Fuel-Optimal Guidance for End-to-End Human-Mars Entry, Powered-Descent, and Landing MIssion.

Wang, J. (2018). A Pseudospectral-Convex Optimization Algorithm for Rocket Landing Guidance.

Wang, J. (2020). An Iterative Convex Programming Method for Rocket Landing Trajectory Optimization.

Weiland, C. (2010). *Computational Space Flight Mechanics.* Springer.

Wertz. (32, Table I-1, 1988). Trajectory Optimization.

Wikipedia. (2023, August 12). *Rube Goldberg machine*. Retrieved from Rube Goldberg machine: https://en.wikipedia.org/wiki/Rube_Goldberg_machine

Wikipedia. (2023, August 12). *Second-order cone programming*. Retrieved from https://en.wikipedia.org/wiki/Second-order_cone_programming

Wikipedia. (n.d.). *Jensen's Inequality*. Retrieved from Wikipedia, The Free Encyclopedia: https://en.wikipedia.org/wiki/Jensen%27s_inequality

Wikipedia. (n.d.). *Mersenne Twister*. Retrieved from Wikipedia, The Free Encyclopedia: https://en.wikipedia.org/wiki/Mersenne_Twister

Wikipedia. (n.d.). *Optimal Control*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Optimal_control

Wikipedia. (n.d.). *Runge-Kutta Methods*. Retrieved from Wikipedia | The Free Encyclopedia: https://en.wikipedia.org/wiki/Runge%E2%80%93Kutta_methods

Wilken, J. (2022). Critical Analysis of SpaceX's Next Generation Space Transportation System _ Starship and Super Heavy.

Wilken, J. (2022). Mission Design for Point-to-point Passenger Transport with Reusable Launch Vehicles.

Wong, E. (2023). Guidance and Control Design for Hazard Avoidance and Safe Landing on Mars. *Journal of Spacecraft and Rockets*.

Yu, Z. (2023). Design and Optimization of Navigation and Guidance Techniques for Mars Pinpoint Landing: Review and Prospect.

Zhao, D.-J. (2017). Reentry trajectory optimization with waypoint and no-fly zone constraints using multiphase convex programming.