Title
Robust Thrust Vector Control for Precision Rocket-Landing

Author
Kamath, Abhinav

Publication Date
2021

Peer reviewed|Thesis/dissertation

Robust Thrust Vector Control for Precision Rocket-Landing

By

Abhinav Girish Kamath

Thesis

Submitted in partial satisfaction of the requirements for the degree of

Master of Science

in

Mechanical and Aerospace Engineering

in the

Office of Graduate Studies

of the

University of California

Davis

Approved:

Stephen K. Robinson, Chair

Francis F. Assadian

Xinfan Lin

Committee in Charge

2021

Contents

List of Figures

List of Tables

Abstract

Robust Thrust Vector Control for Precision Rocket-Landing

The objective of this thesis is to systematically develop the underlying theory behind and implementation of an integrated framework for analytical multibody dynamics modeling and closed-loop simulations with novel control strategies for the powered-descent and precision landing of rocket-powered space vehicles.

The thesis is organized as follows[1]: Chapter 1 provides an introduction to the rocket-landing problem and the motivation for developing new methods and algorithms to enable future planetary landing missions. Chapter 2 describes the implementation of a globally-optimal minimum-propellant powered-descent guidance (PDG) algorithm using lossless convexification and convex optimization. Chapter 3 explains the analytical formulation of the nonlinear equations of motion for a variable-mass multibody rocket system using the extended Kane's equations, and shows results from an open-loop simulation run with the optimal control commands obtained from guidance. Chapter 4 describes feedback control in detail, including a novel method for the design of internally stabilizing multivariable robust feedback controllers using Youla parameterization, along with its application to the underactuated lunar landing problem with feedback control only. Chapter 5 provides an algorithm for the design of internally stabilizing robust LPV controllers via Youla parameterization and applies it to the underactuated lunar landing scenario in a combined feedforward-feedback control architecture with propellant-optimal guidance, control allocation, and various actuator considerations. Chapter 6 concludes the thesis with key observations regarding the work done, the results obtained, the specific contributions, and potential directions for future research.

[1]Parts of this thesis were also presented in 34.

Acknowledgments

This thesis is a testament to all the support and encouragement I have been fortunate enough to receive over the years. To put in words the sheer extent of my gratitude and appreciation for all those who have directly and indirectly helped me in this endeavour is a challenging task; I shall try my best.

I am deeply indebted to my advisor and mentor, Professor Steve Robinson. To say that he has had a major impact on my life would be an understatement. Every positive step in my career will have his fingerprint all over it. I thank him for introducing me to the vast world of rocket science and spaceflight engineering, reigniting in me a sense of childlike wonder, excitement, and enthusiasm, and letting me explore the field of GNC. His space vehicle design class captivated me like no other, and I remain captivated to this day. Every single meeting of ours has been enlightening to me in one way or another—his all-encompassing knowledge, wisdom, unwavering care and support, grace, and humility never cease to amaze me, and continue to inspire me every single day.

I'd like to thank Professor Francis Assadian for the wealth of knowledge he has shared with me and his constant support and words of encouragement. Co-authoring our conference paper with him was an experience that I will always remember; especially all the brainstorming sessions we had and the numerous meetings during which he helped me figure out how to design complicated control systems in the most ingenious way, with his deep technical intuition and humility.

I'd like to extend my gratitude to Professor Xinfan Lin for graciously allowing me to audit his classes and agreeing to serve on my thesis committee. His clear explanation of seemingly complicated concepts helped me understand certain key ideas that directly aided in this work.

I'd also like to express my deepest appreciation to Professor Jason Moore, who not only introduced me to the theory of multibody dynamics, but also assisted me throughout the process of developing the computational rocket model, and with several other projects too. I am one of the lucky few who got to take his multibody dynamics class at UC Davis.

To all my mentors and colleagues at my internships, especially Reid Westwood, Daniel Edney, and Teo Flemming at skyTran, Riccardo Biasini, Steve Davis, and Ethan Head at The Boring Company, Matt Nehrenz and the entire GNC team at Reliable Robotics: the invaluable experience of working with and learning from you has given me a glimpse into simply what it takes to solve extremely difficult real-world challenges and problems that have the potential to change the world. Thank you for placing your trust in me, and giving me the opportunity to contribute to these important efforts toward an undeniably better future.

To Sherry Stukes and Dr. Abhi Jain at JPL: thank you for teaching me so much, being incredibly supportive, and most important of all, showing me that the sky is, in fact, not the limit.

I'd like to thank my uncle, Dr. Dinesh Prabhu, for being a huge inspiration to me and setting the bar for what a rocket scientist should aspire to be.

I also wish to express my utmost gratitude to my teacher, mentor, and well-wisher, Dr. Ganesh Rao, without whom I certainly wouldn't be here—I cannot thank you enough for all your support, encouragement, and your unshakeable belief in me.

To Professor Zhi Ding, whose reassuring words and faith in my decisions were indispensable to my advancement at a very crucial stage in my academic career: thank you very much. To Professor Nazli Gündeş: thank you for your kind words of support and encouragement. Thanks also to Kimia and Mason for being the amazing classmates and teammates that they were, and the friends that they are. I'd also like to thank Dr. Carlos Feres for being a caring guide and teacher.

I'd like to thank my teammates on the Space and Satellite Systems club, the UC Davis Table Tennis team, and team TESS—you showed me what groups of like-minded people can achieve through consistent effort and perseverance.

Being a part of OneLoop was one of the most rewarding experiences in this journey—thank you, OneLoop, for showing me what teams of dedicated and hard-working individuals can achieve even if the odds are stacked against them. I'd like to thank Kareem, Tomo, and everybody else that I had the opportunity to work with, learn from, and have good times with. Thanks also to Sofia Gizzi, for being our advisor and giving us important critical feedback throughout the competition year.

To Tammer and Jason: thank you for the being the best teammates I had the opportunity to collaborate with. I will never forget all the days and nights we spent working on Astrobee, and certainly not the nights we spent in the lab working on our assignments while live-streaming rocket launches on the big screen.

My special thanks go out to John, for being a humble mentor and friend, guiding me every step of the way, and agreeing to proof-read my work every single time without thinking twice, and Josh, for being the nicest roommate, teaching me about things beyond rocket science, and always making sure that I didn't overwork myself (and for all the cookies too). To my entire HRVIP team—you are simply the best.

I'd like to acknowledge all the ECE and MAE staff, TAs, and professors I had the good fortune of interacting with and learning from at UC Davis.

To Danylo: I am incredibly grateful for all your help, advice, and support—you motivate me and inspire me, and I am lucky to call you my friend. To Skye and Purna: your enthusiasm is infectious, and I cannot wait to start working with and learning from you. I particularly look forward to learning from and working with Professor Mehran Mesbahi as well—thank you for all your kind words of encouragement and support.

To Professor Behçet Açıkmeşe and Dr. Lars Blackmore: I hope you know the impact you have had and continue to have on this world. I cannot thank you enough for the revolution that you have set in motion, for it gives me something to look forward to and strive for every single day. I cannot wait to learn all there is to learn from you, and do important work to push the boundaries of what is possible under your guidance, Behçet.

To all my friends: thank you for always wishing the best for me. To my family: thank you for taking care of me and supporting me; thank you for bearing the brunt of the downs while letting me experience the ups and pursue my dreams.

Chapter 1

Introduction

With humans missions to the Moon and Mars on the horizon, the successful Mars landing missions in the recent past (Açıkmeşe et al., 2014; Chen et al., 2016; Nelessen et al., 2019; Prakash et al., 2008; San Martin, 2017; San Martin et al., 2013; Steltzner et al., 2006), and the almost routine landings of suborbital- and orbital-class reusable rocket boosters and prototypes in the last few years (Blackmore, 2016), there has been a renewed interest in planetary rocket-powered landing technologies, especially in the domain of guidance, navigation, and control (GNC). The recent announcements of the Human Landing System (HLS) and Commercial Lunar Payload Services (CLPS) contracts through NASA's Artemis lunar exploration program (Chavers et al., 2019, 2020; Smith et al., 2020), especially, have rekindled the spirit of lunar landing from the Apollo era—this comes nearly 50 years after humans last set foot on the Moon, only this time, with a stronger focus on enhanced safety, reliability, and precision, and plans for sustained operations on the lunar surface (Petersen et al., 2020). The onboard computing power that we have at our disposal today allows for the implementation of advanced guidance algorithms using real-time convex optimization (Açıkmeşe et al., 2013), for instance, and navigation methods such as map-relative localization (Johnson et al., 2016), that are also efficient, reliable, and repeatable. Such technologies have the potential to enable the exploration of previously inaccessible science-rich regions on celestial bodies, and also make human landing missions much safer.

One of the most critical aspects of a planetary landing mission is the powered-descent phase, which terminates with a soft touchdown on the surface. The Apollo lunar landing missions employed polynomial guidance for powered-descent and manual control for terminal-descent (Klumpp, 1974). Although all the attempted landings were successful, the descent trajectories were not propellant-optimal, and the dimensions of the landing dispersion ellipses were in the order of kilometers (Quaide and Oberbeck, 1969). Recent advances in technology have made way for a more robust, autonomous approach to precision planetary landing. One such technology is terrain relative navigation (TRN), which has been extensively researched and was successfully performed during the landing of Perseverance (Mars 2020) on the surface of Mars in February 2021 (Johnson et al., 2017). Precision landing reduces the size of the landing uncertainty ellipse considerably, bringing it down to a precision of meters, and is key to broadening the scope of planetary landing missions. It enables scientific exploration missions that require precise selection of landing sites (Blackmore, 2016). This could potentially involve a target landing zone in a region of interest that has an extremely small margin for error in terms of touchdown safety.

Recent breakthroughs in the formulation of real-time interior point method algorithms have led to the invention of powered-descent guidance (PDG) algorithms that are computationally efficient, making them desirable for real-time implementation (Dueri et al., 2017; Scharf et al., 2017). These algorithms rely on simplified dynamical models, however, and implementing them in an open-loop in a real scenario could lead to large trajectory tracking errors due to inaccuracies in the model and nonlinearities in the actual system that are not accounted for. Hence, it becomes necessary to adopt a feedback control design strategy that ensures that the closed-loop system is insensitive to these nonlinearities, while also being robust to system parameter uncertainties and external disturbances. Propellant-optimal guidance and TRN-augmented inertial navigation in conjunction with robust trajectory tracking control would enable a full-stack closed-loop guidance, navigation, and control design, as depicted in Figure 1.1, for autonomous precision landing for future human and robotic missions to the Moon and beyond.

Blackmore (2016) presents some of the challenges of trying to precisely land a rocket—to be able to efficiently solve what is arguably one of the most challenging engineering problems today, in order to expand mission capabilities while ensuring reliability and safety, extensive research on the subject is necessary. The objective of this thesis is to develop robust methods and tools that could potentially aid in the efforts toward that goal.

In this thesis, a multibody dynamics model of a variable-mass rocket system with a gimbaled main engine is developed, a propellant-optimal powered-descent guidance algorithm is implemented, and novel approaches to robust control design for multivariable and linear parameter-varying (LPV) systems are presented, along with their application to gimbaled thrust vector control (TVC) for the precision landing of rocket-powered space vehicles. The gimbaled TVC landing problem is often informally referred to as the *broomstick* problem, given that it is akin to trying to balance a broomstick with a finger. An underactuated planar lunar landing scenario is adopted as a case-study throughout this thesis to aid in the systematic development of the closed-loop simulation framework.

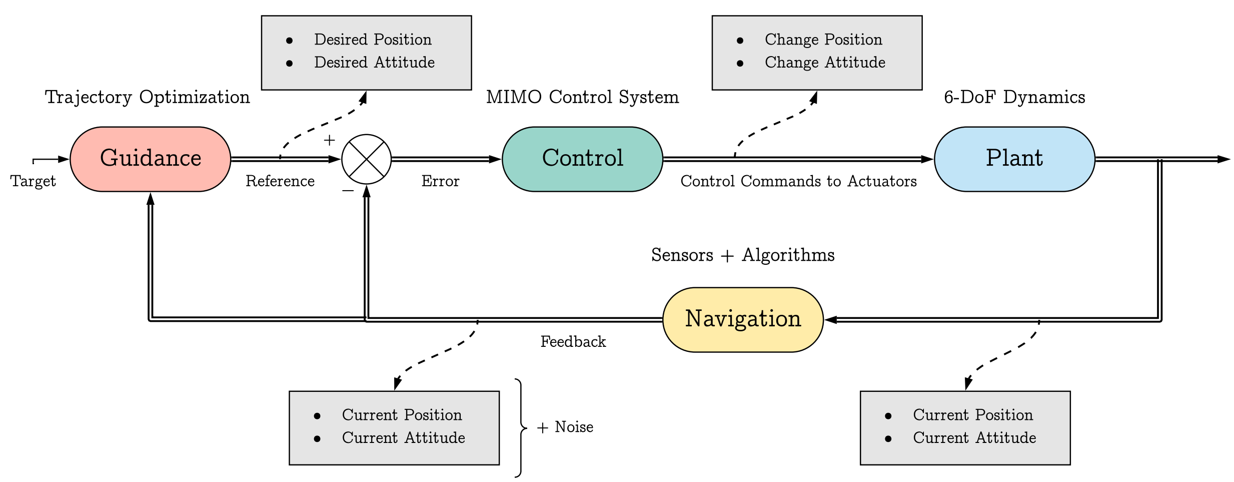

Figure 1.1: A high-level guidance, navigation, and control block diagram

Chapter 2

Guidance

2.1 Powered-Descent Guidance via Lossless Convexification

The purpose of guidance is two-fold—to generate reference state trajectories for the vehicle to follow, and to generate a series of feasible feedforward control commands to the actuators of the vehicle that ensure that it accurately follows the desired state trajectories without violating any of the imposed constraints. Guidance problems are typically posed as optimization problems, especially for complex regimes such as powered-descent and landing.

The powered-descent guidance (PDG) problem can be posed as an optimization problem subject to various constraints involving the dynamics of the landing spacecraft, thruster limitations, and physical constraints to ensure safe landing, with the amount of propellant consumed regarded as the 'cost' (Açıkmeşe and Ploen, 2007). Moreover, this constrained optimization problem can be structured as one of convex optimization, and by obeying a set of well-defined rules to ensure problem tractability, a globally optimal solution can be guaranteed (Boyd and Vandenberghe, 2004), the corresponding algorithm itself being amenable to real-time onboard implementation.

Optimization-based PDG is not only helpful in minimizing propellant consumption (which is a very important metric in spaceflight) or optimizing any other chosen objective for that matter, but also allows for making full use of the feasible flight envelope without the need for being too conservative (it enables aggressive divert maneuvers, for example), all while guaranteeing satisfaction of the imposed mission and physical constraints. Implementation of such algorithms can considerably expand the scope of rocket landing missions, and together with advances in navigation methods such as terrain-relative navigation (TRN), can enable precision landing on planetary bodies (with landing dispersion ellipses on the order of meters), including the Earth. Such capabilities can also significantly improve the safety of human landing missions.

Typically for planetary landing missions, after powered-descent has been initiated, the thrusters are shut down only at touchdown. The entirety of the powered-descent phase involves one continuous, throttled burn by the main engine—this strategy is adopted mainly as a safety consideration, given that a liquid-propellant rocket engine might shut down if throttled below a certain thrust limit and might not be able to reliably relight when required. However, the lower-bound on thrust that results from such a constraint turns out to be nonconvex. Additionally, requirements on its direction (thrust pointing) add nonconvex constraints to the trajectory optimization problem as well, and problem reformulation is required to make the problem tractable.

Convexification of these constraints and reformulation of the original problem have been proven to generate globally optimal solutions, given that feasible solutions exist. Convexification refers to the introduction of a slack variable to lift the nonconvex lower thrust bound and the nonconvex thrust pointing angle constraint into convex sets of feasible controls—more specifically, convex cones. Since a global optimum for the relaxed problem also guarantees a global optimum for the original problem—based on mathematical proofs involving Hamiltonian analysis, transversality conditions, and Pontryagin's maximum principle—this mathematical result is widely referred to as *lossless* convexification (Carson et al., 2011).

Post lossless convexification, the cost-function (propellant consumption) can be reformulated as the logarithm of the touchdown lander mass, with the optimization problem being maximization of the reformulated cost-function subject to the imposed constraints (Açıkmeşe and Blackmore, 2011).

The algorithm employed here follows the basis of the real-time implementable G-FOLD algorithm developed at JPL (Açıkmeşe et al., 2012). The problem formulation involves a piecewise-linear characterization of the control inputs (resulting in piecewise-cubic translation trajectories) and simplified and discretized equations of motion. It adopts a point-mass formulation, thus decoupling translation and *attitude* in guidance (the actual translational and attitude dynamics of the lander, however, are coupled and highly nonlinear—such modeling inaccuracies along with system parameter uncertainties, potential external disturbances, and sensor noise, necessitate feedback control on the actual vehicle). The relaxed convex minimum-propellant powered-descent guidance (PDG) problem, after a change of variables and discretization, is structured as a finite-dimensional convex optimization problem—in particular, a second-order cone program (SOCP) (Açıkmeşe et al., 2008). This formulation of the problem, the objective being maximization of the (logarithm of the) lander mass at touchdown, has been proven to produce globally optimal solutions when feasible solutions exist. The interested reader is referred to (Açıkmeşe et al., 2013) for a more complete description of the problem.

The Convex Propellant-Optimal PDG Problem

(Açıkmeşe, Scharf, Blackmore, and Wolf, 2008; Açıkmeşe, Casoliva, Carson, and Blackmore, 2012; Açıkmeşe, Carson, and Blackmore, 2013; Bhasin, 2016; Kos, Polsgrove, Sostaric, Braden, Sullivan, and Le, 2010; Malyuta, Reynolds, Szmuk, Lew, Bonalli, Pavone, and Açıkmeşe, 2021; Pinson and Lu, 2018)

$$\min_{u_0,..,u_N,\sigma_0,..,\sigma_N} \quad -z_N$$

$$\underset{\text{for } k=0,..,N}{\text{subject to}}$$

$$\left.\begin{aligned}
r_{k+1} &= r_k + \tfrac{\Delta t}{2}\left(v_k + v_{k+1}\right) - \tfrac{\Delta t^2}{12}\left(u_{k+1} - u_k\right)\\
v_{k+1} &= v_k + \tfrac{\Delta t}{2}\left(u_k + u_{k+1}\right) + g\Delta t\\
z_{k+1} &= z_k - \tfrac{\alpha\Delta t}{2}\left(\sigma_k + \sigma_{k+1}\right)
\end{aligned}\right\} \text{Dynamics}$$

$$\|u_k\| \leq \sigma_k \qquad \} \text{ Relaxation}$$

$$\left.\begin{aligned}
&z_{0,k} = \ln\left(m_{wet} - \alpha\rho_2 k\Delta t\right)\\
&\mu_{1,k} = \rho_1 e^{-z_{0,k}} \quad \mu_{2,k} = \rho_2 e^{-z_{0,k}}\\
&\mu_{1,k}\left[1 - \left(z_k - z_{0,k}\right) + \frac{\left(z_k - z_{0,k}\right)^2}{2}\right] \leq \sigma_k\\
&\qquad\qquad \leq \mu_{2,k}\left[1 - \left(z_k - z_{0,k}\right)\right]\\
&z_{0,k} \leq z_k \leq \ln\left(m_{\text{wet}} - \alpha\rho_1 k\Delta t\right)
\end{aligned}\right\} \text{Thrust and Mass Bounds}$$

$$\left.\begin{aligned}
&z_0 = \ln m_{wet} \quad z_N \geq \ln m_{dry} \quad N\Delta t = t_f\\
&r_0 = r_0 \quad r_N = r_f\\
&v_0 = v_0 \quad v_{N_x} = v_{f_x}
\end{aligned}\right\} \text{Boundary Conditions}$$

$$r_{k_x} \geq \tan(\theta_{GS})\left\|r_{k_{y,z}}\right\| \qquad \} \text{ Glide-Slope}$$

$$u_{k_x} \geq \cos(\theta_P)\sigma_k \quad u_{N_{y,z}} = 0 \qquad \} \text{ Thrust Pointing}$$

$$\left.\begin{aligned}
&v_{k_{y,z}} = 0 \quad \text{for } k = (N-i), ..., N;\ i \geq \frac{1}{\Delta t}\\
&v_{k_x} = v_{td} \quad \text{for } k = (N-i), ..., (N - \frac{1}{\Delta t})
\end{aligned}\right\} \text{Terminal-Descent}$$

r	Position
r_x, r_y, r_z	Altitude, crossrange, and downrange, respectively
v	Velocity
v_x, v_y, v_z	Rate-of-descent, lateral velocity, and longitudinal velocity, respectively
m_{wet}	Wet mass of the lander
m_{dry}	Dry mass of the lander
z	$\ln m;\ m \rightarrow$ instantaneous lander mass
g	Gravitational acceleration
u	$T_c/m;\ T_c \rightarrow$ commanded thrust vector
σ	$\Gamma/m;\ \Gamma \rightarrow$ slack variable $\mid \|T_c\| \leq \Gamma$
N	Number of temporal nodes
Δt	Temporal resolution
t_f	Time-of-flight
α	Thrust-specific fuel consumption (TSFC)
ρ_1, ρ_2	Lower and upper bounds on thrust, respectively
θ_{GS}	Minimum glide-slope angle from the ground plane
θ_P	Maximum thrust pointing angle from the vertical
$i\Delta t$	Duration of vertical-only terminal-descent
v_{td}	Constant rate-of-descent during vertical-only terminal-descent

In addition, the thrust pointing rate constraint can be *approximately* imposed by Equation 2.1 (Açıkmeşe and Ploen, 2005).

$$\left\| Q^{\frac{1}{2}} U_k \right\| \leq \frac{\lambda_{\min}(P^{\frac{1}{2}})}{\sqrt{2}} [\sigma(k\Delta t) + \sigma([k-1]\Delta t)] \tag{2.1}$$

where,

$$U_k = \begin{pmatrix} u(k\Delta t) \\ u([k-1]\Delta t) \end{pmatrix}, \qquad Q = \begin{pmatrix} I & -\frac{1}{2}I \\ -\frac{1}{2}I & I \end{pmatrix}, \qquad P = \begin{pmatrix} 1 & -\frac{\Omega}{2} \\ -\frac{\Omega}{2} & 1 \end{pmatrix},$$

$I \in \mathbb{R}^{3\times 3}$ is the identity matrix, and $\Omega = \cos(\omega\Delta t)$, ω being the maximum allowed angular rate.

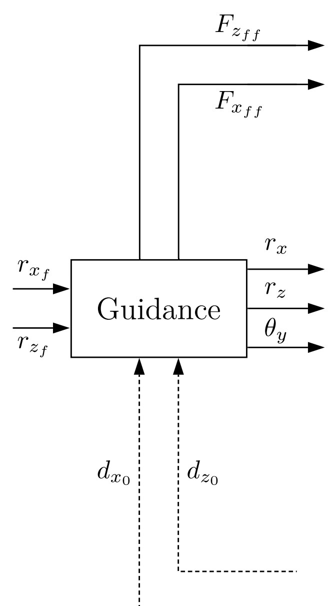

Figure 2.1: The guidance block (see Figure 5.4)

2.2 Trajectory Generation

An x - z planar lunar landing scenario is considered for the purposes of simulation and demonstration of the control design methodology adopted. The chosen parameter values and nominal boundary conditions for propellant-optimal guidance trajectory generation are listed in Table 2.1.

r_0	$[-500.0,\ 0.0,\ 400.0]$ m	Initial position
v_0	$[40.0,\ 0.0,\ -13.0]$ m/s	Initial velocity
r_f	$[0.0,\ 0.0,\ 0.0]$ m	Target landing position
v_f	$[0.0,\ 0.0,\ 0.0]$ m/s	Touchdown velocity
T_{max}	83000 N	Full-thrust magnitude
ρ_1	30% T_{max}	Lower thrust bound (minimum throttle)
ρ_2	80% T_{max}	Upper thrust bound (maximum throttle)
g	-1.625 m/s^2	Lunar gravitational acceleration
m_{wet}	25000 kg	Wet-mass of the lander (at r_0)
m_{dry}	10000 kg	Dry-mass of the lander
α	0.00022655325 s/m	Thrust-specific fuel consumption (TSFC)

Table 2.1: Guidance parameter values and boundary conditions

The main engine is allowed to throttle between 30% and 80% the full-thrust magnitude. The upper throttle limit is set to allow for a thrust margin between the braking-burn set throttle and the maximum engine thrust (Kos et al., 2010).

A linear search is used to determine the minimum feasible time-of-flight (62 seconds). Brent's algorithm (Brent, 2013) and the golden search technique (Bertsekas, 1997) are used to compute the propellant-optimal (global minimum) time-of-flight value (63.18 seconds), given that the minimum of propellant consumption is a unimodal function of the time-of-flight (Açıkmeşe et al., 2008). The trajectory is subject to a minimum glide-slope angle of 4 degrees from the ground plane—the glide-slope constraint ensures that the lander is at a safe distance from the surface at all points in the landing trajectory. It also enforces surface impact avoidance and does not allow the generation of a trajectory that involves subsurface flight (Carson et al., 2011).

The trajectory is also subject to a maximum thrust pointing angle of 50 degrees from the vertical—the thrust pointing constraint becomes especially useful when there are strict requirements on the orientation of onboard sensors for terrain relative navigation (Açıkmeşe et al., 2013). The thrust pointing rate constraint is approximately imposed (Equation 2.1) with a maximum allowed angular rate of 5 degrees/second. The maximum thrust pointing rate recorded in the simulation run was 3.86 degrees/second (it was observed, however, that for some more aggressive trajectories, the thrust pointing rate constraint was violated at the initiation of vertical-only terminal-descent).

Vertical-only terminal-descent constraints are imposed for the final 30 seconds of landing with a constant rate-of-descent of -1 m/s until the final temporal node. In a real landing scenario, these constraints would help mitigate undesirable fuel-slosh effects and tip-over on touchdown due to lateral motion (Kos et al., 2010). The projected amount of propellant consumed is 727.3 kg. The MOSEK [51] solver was used in tandem with CVXPY (Diamond and Boyd, 2016), a Python-embedded modeling language for convex optimization problems.

The resulting mass-depletion rate (proportional to the magnitude of thrust: $\dot{m} = -\alpha \left\| T_c \right\|$) and thrust pointing angle (main engine gimbal pitch) profiles are considered to be the open-loop control inputs for the planar lunar landing simulation with multivariable feedback control.

This PDG problem is solved to obtain reference state trajectories for the multivariable feedback control simulation described in Section 4.2. The optimal control inputs are not used in this closed-loop simulation, however. The optimal thrust profile demonstrates high-frequency *chatter* during the vertical-only terminal-descent phase, due to the trapezoidal-rule-based discretization scheme adopted for the dynamics. The frequency of this chatter is observed to increase with an increase in the number of temporal nodes. The phenomenon of chatter has been observed in the literature with the imposition of similar constraints (Liu, 2013; Liu and Lu, 2013; Liu et al., 2016; Szmuk et al., 2017). Although the guidance solutions obtained are feasible, the chatter is undesirable, especially when including the optimal thrust commands as feedforward control inputs in the simulation framework.

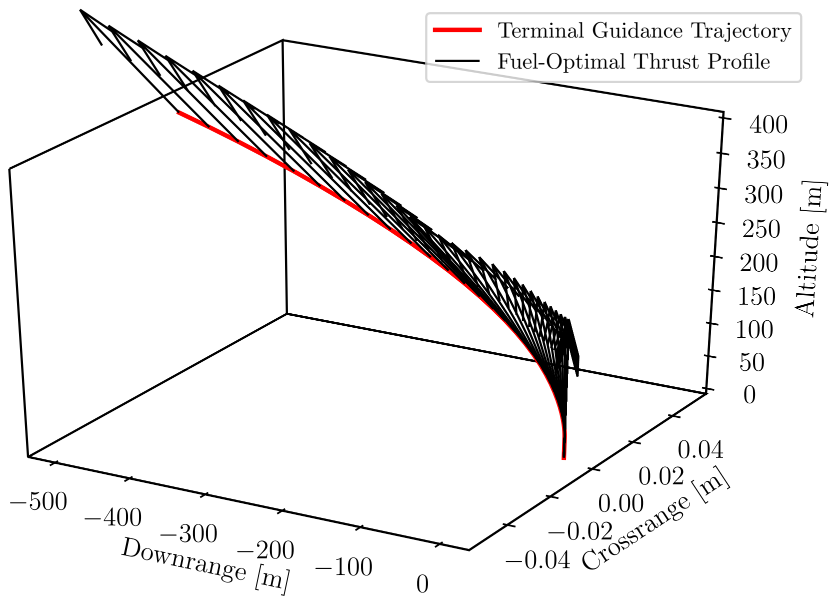

Figure 2.2: The reference powered-descent trajectory along with the optimal thrust profile generated using the guidance algorithm

In order to combat this behavior, the discretization scheme presented in (Açıkmeşe et al., 2013) is adopted. A continuous state-space realization of the dynamics is considered, with the position, velocity, and total mass regarded as the states. The system is discretized via the zero-order hold (ZOH) method with the sampling interval set to the temporal resolution of the numerical optimization problem (1 second). The optimization variables are appropriately scaled via affine transformations in order to obtain accurate results, and the cost-function is reformulated in terms of the acceleration magnitude (Açıkmeşe and Ploen, 2007; Malyuta et al., 2021). The (approximate) thrust pointing rate constraint is not imposed, in order to ensure that the solution is globally optimal. The resulting optimal control (thrust) solution is devoid of chatter. Figure 2.3 illustrates the lossless nature of convexification of the thrust lower-bound and pointing constraints via relaxation of the acceleration norm (the relaxed constraint holds with equality at optimality: $\|u\| = \sigma$).

The chosen parameter values and nominal boundary conditions for propellant-optimal guidance trajectory generation for this case are listed in Table 2.2. The maximum glide-slope angle, the maximum thrust pointing angle, and the terminal-descent parameters are left unchanged from the previous case. The propellant-optimal time-of-flight ($t_f^* = 79.0005$

seconds)[1] is directly found using Brent's algorithm. The projected amount of propellant consumed is 855.59 kg. The ECOS [25] solver was used along with CVXPY for this case.

r_0	$[-1000.0,\ 0.0,\ 900.0]$ m	Initial position
v_0	$[45.25,\ 0.0,\ -10.0]$ m/s	Initial velocity
r_f	$[0.0,\ 0.0,\ 0.0]$ m	Target landing position
v_f	$[0.0,\ 0.0,\ 0.0]$ m/s	Touchdown velocity
T_{max}	83000 N	Full-thrust magnitude
ρ_1	30% T_{max}	Lower thrust bound (minimum throttle)
ρ_2	80% T_{max}	Upper thrust bound (maximum throttle)
g	-1.625 m/s^2	Lunar gravitational acceleration
m_{wet}	25000 kg	Wet-mass of the lander (at r_0)
m_{dry}	10000 kg	Dry-mass of the lander
α	0.00022655325 s/m	Thrust-specific fuel consumption (TSFC)

Table 2.2: Guidance parameter values and boundary conditions (ZOH discretization case)

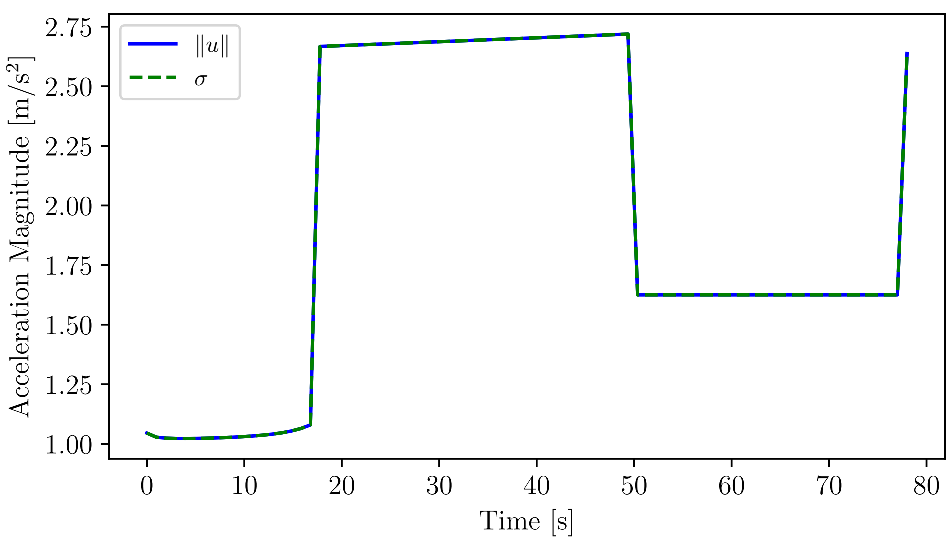

Figure 2.3: Illustration of the relaxed acceleration lower-bound holding with equality

[1]Since the temporal resolution for numerical optimization is chosen to be 1 second, however, the practical time-of-flight (in the closed-loop simulation described in Chapter 5) is set to $t_f = \lceil t_f^* \rceil = 80$ seconds.

Chapter 3

Dynamics

3.1 Extended Kane's Equations for Variable-Mass Systems

The modeling of dynamical systems has been studied extensively in the literature. Classical methods such as the Newton-Euler method, Lagrangian formulations, and Hamiltonian mechanics are popular, but can be cumbersome and computationally intensive for analytical multibody dynamics modeling. More recent methods such as the spatial operator algebra (SOA) (Jain, 2010; Rodriguez et al., 1991) and Kane's method (Kane and Levinson, 1985; Kane et al., 1983), are highly systematic approaches that become especially desirable in the modeling of complex systems. Here, a multibody dynamics modeling framework using Kane's method and the extended Kane's equations for variable-mass systems (Ge and Cheng, 1982) is presented.[1]

The lunar lander is modeled as a holonomic system. The body of the lander is considered to be a rigid cube with length l and uniform mass M, as shown in Figure 3.1. The main engine is located at point P, at which the generated thrust is applied. The inertial effects of the main engine on the overall lander dynamics are assumed to be negligible. The onboard propellant is assumed to be a particle with variable-mass, m, located midway between the body mass-center and P.

N is the Moon-fixed inertial reference frame, L is the reference frame attached to the

[1]Implementations of Kane's method for a holonomic system and the extended Kane's equations for a nonholonomic variable-mass system can be found in [35] and [36], respectively.

body of the lander, and E is the reference frame attached to the gimbaled main engine. Considering 6 generalized coordinates $\{d_x,\ d_y,\ d_z,\ q_1,\ q_2,\ q_3\}$, where $\{d_x,\ d_y,\ d_z\}$ represent the position of the mass-center of the lander with respect to a fixed point, O, in N, and $\{q_1,\ q_2,\ q_3\}$ represent the Euler rotation angles of L relative to N, their time-derivatives as the generalized speeds $\{\dot{d}_x : v_x,\ \dot{d}_y : v_y,\ \dot{d}_z : v_z,\ \dot{q}_1 : v_1,\ \dot{q}_2 : v_2,\ \dot{q}_3 : v_3\}$, and the propellant mass, m, the modeled system has 13 states. The lander is modeled as an underactuated system, with 6 degrees of freedom and 3 control inputs—the mass-depletion rate, $\dot{m}$, and the two main engine gimbal angles, e_1 and e_2. The thrust vector always acts along $\hat{e}_x$, and has a magnitude $-C\dot{m}$, where C, the exit-velocity of the ejected propellant, is a positive constant.

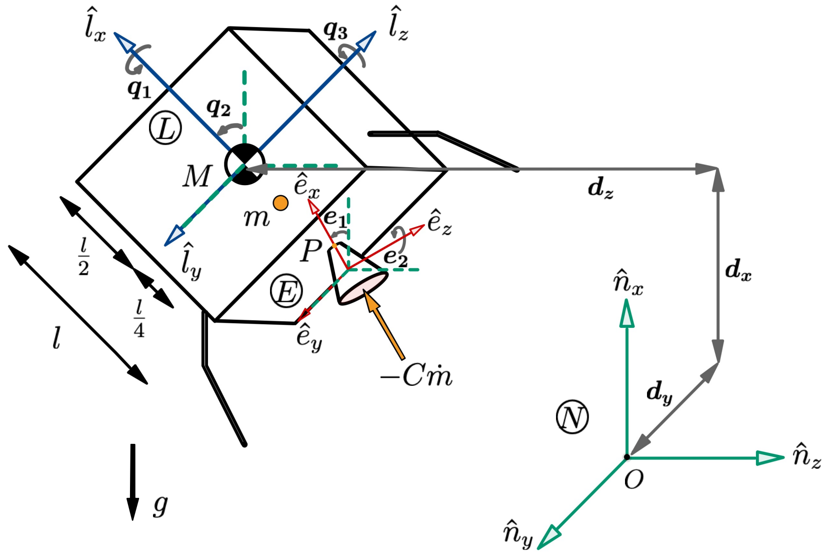

Figure 3.1: The modeled multibody lunar landing system with the generalized coordinates and the reference frames

The principal moments of inertia of the system can be parameterized by the mass terms to get the following closed-form equations: $I_{xx} = \frac{1}{6}Ml^2$; $I_{yy} = I_{zz} = \frac{1}{6}Ml^2 + m(\frac{l}{4})^2 \rightarrow$ note that I_{yy} and I_{zz} vary with propellant mass, m.

The extended Kane's equations for holonomic variable-mass systems (Ge and Cheng, 1982) take the form:

$$F_r + F_r^* + F_r^{**} = 0 \quad (r = 1, ..., 6) \tag{3.1}$$

where F_r $(r = 1, ..., 6)$, the generalized active forces, account for the resultant of all body and contact forces and torques, F_r^* $(r = 1, ..., 6)$, the generalized inertia forces, account for the inertia forces and torques, and $F_r^{**}(r = 1, ..., 6)$, the generalized thrusts (Banerjee, 2000; Ge and Cheng, 1982), account for the forces due to the time-derivative of mass.

The generalized forces acting at the mass-center of the rigid body L, and point P, from where the propellant (variable-mass particle) is being expelled, are defined as follows in the inertial reference frame, N:

$$
\begin{aligned}
F_r &\triangleq (V_r^L \cdot R_L) + (V_r^P \cdot R_P) \quad (r = 1, ..., 6) \\
F_r^* &\triangleq (V_r^L \cdot R_L^* + \omega_r^L \cdot T_L^*) + (V_r^P \cdot R_P^*) \quad (r = 1, ..., 6) \\
F_r^{**} &\triangleq V_r^P \cdot C\dot{m} \quad (r = 1, ..., 6)
\end{aligned}
\tag{3.2}
$$

where V_r^L $(r = 1, ..., 6)$ are the holonomic partial velocities of the mass-center of rigid body L, V_r^P $(r = 1, ..., 6)$ are the holonomic partial velocities of the point P, $\dot{m}$ is the time-derivative of m, and C is the exit-velocity of the variable-mass particle from P.

Here, the resultant of all contact and body forces acting on L (at the mass-center), R_L, the resultant of all contact and body forces acting at P, R_P, the inertia force acting on L (at the mass-center), R_L^*, the inertia force acting at P, R_P^*, and the inertia torque acting on L, T_L^*, are given by:

$$
\begin{aligned}
R_L &= -Mg\,\hat{n}_x \\
R_P &= -mg\,\hat{n}_x \\
R_L^* &= -Ma^L \\
R_P^* &= -ma^P \\
T_L^* &= -(\alpha^L \cdot I_L + \omega^L \times I_L \cdot \omega^L)
\end{aligned}
\tag{3.3}
$$

where, g is the lunar gravitational acceleration, a^L is the acceleration of the mass-center of L, a^P is the acceleration of point P, ω^L is the angular velocity of L, α^L is the angular acceleration of L, and I_L is the central inertia dyadic of L, all defined in the inertial reference frame, N.

The resulting equations of motion (3.1) can be expressed in the implicit form with the kinematics and dynamics combined (Meurer et al., 2017):

$$\mathbb{M}\dot{\mathbf{x}} = \mathbb{F} \tag{3.4}$$

where $\mathbb{M}$ is the mass matrix of the combined equations, $\mathbb{F}$ is the corresponding forcing vector, and $\dot{\mathbf{x}}$ is the derivative of the state vector that includes 13 states: the generalized coordinates, $\{d_x,\ d_y,\ d_z,\ q_1,\ q_2,\ q_3\}$, the generalized speeds, $\{v_x,\ v_y,\ v_z,\ v_1,\ v_2,\ v_3\}$, and the propellant mass, $\{m\}$. This system of equations can be linearized about a chosen operating point using Jacobian linearization to obtain an explicit first-order state-space representation of the system (Hampton et al., 2001), as shown in Equation 3.5, which will serve as the plant model for MIMO control system design.

$$\delta\dot{\mathbf{x}} = \mathbf{A}\delta\mathbf{x} + \mathbf{B}\delta\mathbf{u} \tag{3.5}$$

where $\mathbf{A}_{13\times 13}$ is the state matrix, $\mathbf{B}_{13\times 3}$ is the input matrix, and $\delta\mathbf{u}$ is the differential input (control) vector, where $\mathbf{u}$ includes the inputs, $\{\dot{m}, e_1, e_2\}$. As linearization is performed about a chosen operating point, the resulting linear state equations are described in terms of differential changes of $\mathbf{x}$, $\delta\mathbf{x}$, about that operating point.

3.2 Nonlinear Lander Model

The parameter values for the modeled lunar landing system, as shown in Table 3.1, are chosen to closely approximate the Altair lunar lander (Brown and Connolly, 2012).

l	9.35 m	Length of the lander (cube)
$M = m_{dry}$	10000 kg	Mass of the lander body
$m = m_{wet} - m_{dry}$	15000 kg	Initial propellant mass
$C = \frac{1}{\alpha}$	4413.973165 m/s	Exit-velocity of the propellant

Table 3.1: Parameter values of the nonlinear lander model

The first-order ordinary differential equations (ODEs) obtained from Equation 3.1 are integrated to obtain the evolution of the states. For the planar landing scenario, the control inputs from guidance are applied to the nonlinear model in an open-loop, and the mass-center of the lander body is tracked. The initial conditions of the lander are set to the values listed in Table 2.1. The resulting landing translation trajectory traced by the nonlinear model is shown in Figure 3.2, and the corresponding pitch trajectory is shown in Figure 3.3. As expected, the pitch of the lander body is unstable, due to the fact that no explicit attitude control torque commands are given to the lander.

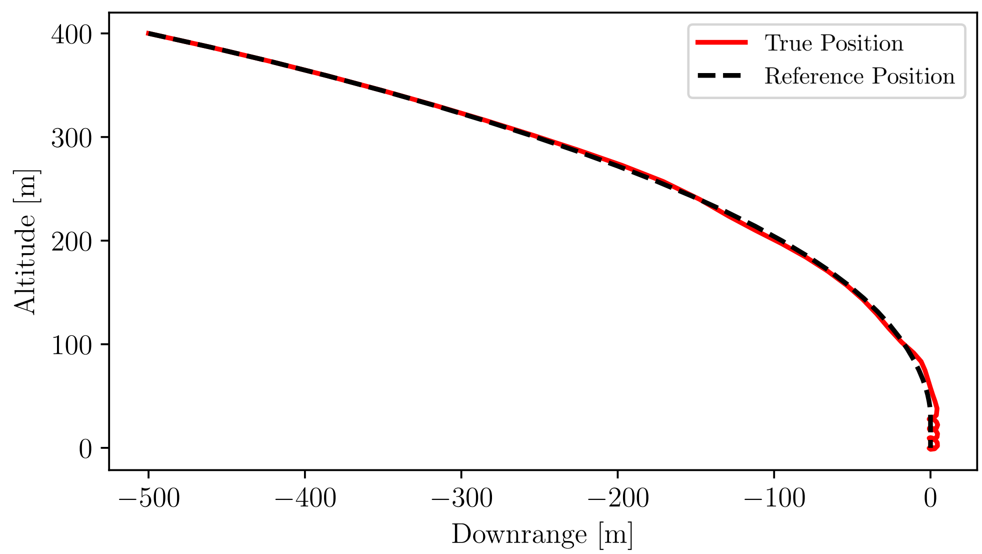

Figure 3.2: Translation trajectory traced by the body mass-center of the nonlinear lander model with open-loop control

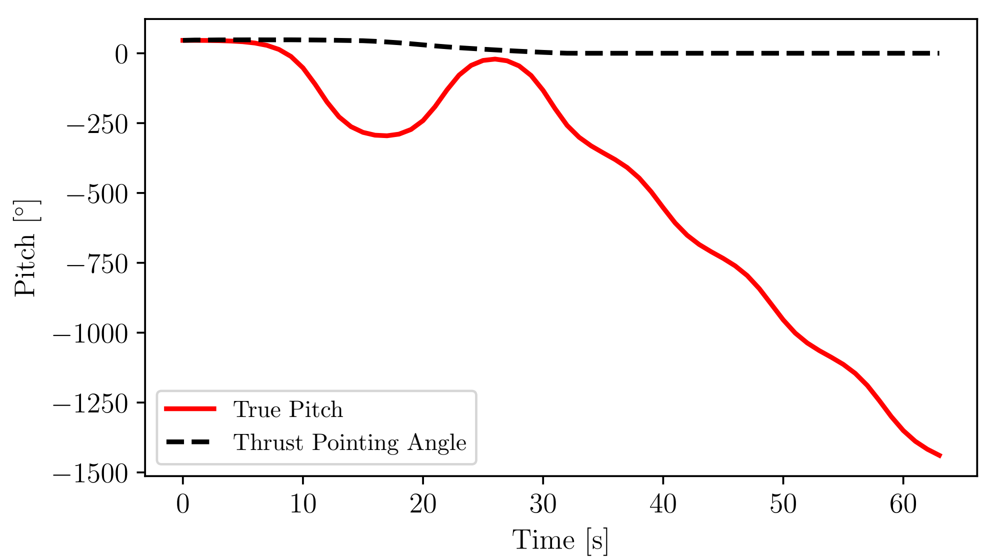

Figure 3.3: Pitch (unstable) trajectory followed by the body of the nonlinear lander model with open-loop control

Considering the 3-DoF (downrange, altitude, and pitch) planar landing scenario, a reduced-order model is obtained to include 7 states: $\{v_x, v_z, v_2, d_x, d_z, q_2, m\}$, 2 control inputs: $\{\dot{m}, e_1\}$, and 3 measured outputs: $\{d_x, d_z, q_2\}$. The resulting combined mass-matrix, state-derivative vector, and forcing vector (see Equation 3.4) are shown in Equations 3.6, 3.7, and 3.8, respectively.

$$
\mathbb{M} = \begin{pmatrix}
-M - m & 0 & -\frac{lm\sin(q_2)}{2} & 0 & 0 & 0 & -C\cos(e_1) \\
0 & -M - m & -\frac{lm\cos(q_2)}{2} & 0 & 0 & 0 & C\sin(e_1) \\
-\frac{lm\sin(q_2)}{2} & -\frac{lm\cos(q_2)}{2} & -\frac{l^2(8M+3m)}{48} - \frac{l^2 m}{4} & 0 & 0 & 0 & \frac{Cl\sin(e_1-q_2)}{2} \\
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix} \tag{3.6}
$$

$$
\dot{\mathbf{x}} = \begin{pmatrix}
\dot{v}_x \\
\dot{v}_z \\
\dot{v}_2 \\
\dot{d}_x \\
\dot{d}_z \\
\dot{q}_2 \\
\dot{m}
\end{pmatrix} \tag{3.7}
$$

$$
\mathbb{F} = \begin{pmatrix}
g(M+m) + \frac{lmv_2^2\cos(q_2)}{2} \\
-\frac{lmv_2^2\sin(q_2)}{2} \\
\frac{glm\sin(q_2)}{2} \\
v_x \\
v_z \\
v_2 \\
\dot{m}
\end{pmatrix} \tag{3.8}
$$

SymPy (Meurer et al., 2017), an open-source Python library for symbolic computation, was used to model the multibody dynamics of the lunar lander.

The nonlinear equations of motion are linearized about the ideal touchdown state (the final guidance state) to get a state-space realization of the system. The resulting transfer function matrix, G_p, of the linearized MIMO plant model is shown in Equation 3.9, where s

is the complex variable. The linear plant is a 3×2 transfer function matrix, with 2 inputs, $\{\dot{m},\, e_1\}$, and 3 outputs, $\{d_x, d_z, q_2\}$. The outputs and their derivatives are controllable.

$$G_p = \begin{pmatrix} -\frac{0.18185}{s^2} & 0 \\ 0 & -\frac{1.4667\,(s-1.082)\,(s+1.082)}{s^2\,(s-0.8626)\,(s+0.8626)} \\ 0 & -\frac{0.30654}{(s-0.8626)\,(s+0.8626)} \end{pmatrix} \tag{3.9}$$

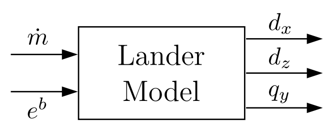

Figure 3.4: The nonlinear lander model block (see Figure 5.4)

Chapter 4

Control

Various methods for control design have been investigated and analyzed in the literature. Implementation of PID control has been successfully demonstrated in conjunction with the G-FOLD algorithm in real-time on the Masten Xombie lander testbed (Açıkmeşe et al., 2012). Recently, model predictive control (MPC) has gained popularity, given its seamless applicability with existing guidance algorithms (Lee and Mesbahi, 2017; Pascucci et al., 2015). Methods such as nonlinear model predictive control (NMPC) are capable of handling complex nonlinear system dynamics, but they come at the cost of being computational intensive. This is certainly true for online implementations, given that the algorithms require a solution to an optimization problem at every sampling instant to obtain the necessary control commands (Liu et al., 2012). Given the lack of robustness guarantees in methods such as PID control design, and the computational intensity of MPC, it becomes desirable to develop simple, real-time deployable, robust control systems that can handle complex nonlinear MIMO systems with coupled dynamics.

Many optimal control techniques such as the linear quadratic regulator (LQR) and linear quadratic Gaussian (LQG) control have also been widely investigated in the literature for both single-input single-output (SISO) and multiple-input multiple-output (MIMO) systems. Although these techniques can be used for the control of multivariable processes, methods such as LQG have been proven to lack robustness to system parameter uncertainties (Doyle, 1978). This fact was exemplified by the failures of LQG controller implementations on two separate occasions in 1975: the LQG controller on a Trident submarine caused it to unexpectedly surface in a rough sea simulation, and the LQG control system on the F-8C crusader aircraft led to unsatisfactory results. Although heuristic methods such as loop transfer recovery (LTR) can increase the robustness of LQG controllers, they come at the price of severely degrading the original LQG cost-function, and thus lead to non-optimal solutions (Noll).

Robust control becomes essential especially when human landing missions are considered (Orr and Shtessel, 2009). H_∞-optimization-based robust control has been successfully deployed on the Ariane 5 Evolution launch vehicle for the atmospheric flight phase, replacing the previously used LQG controller. In telecommunication satellites, H_∞-optimization-based robust control has been shown to reduce the propellant mass consumption by 10% during station-keeping maneuvers (Philippe et al.). Linear, constant- and varying-gain, multivariable feedback controllers can be designed offline and deployed for real-time control of systems, with high efficiency and low computational intensity, while also guaranteeing robustness to nonlinearities, system parameter uncertainties, external disturbances, and sensor noise. These designs can be extended to adapt to changing system parameters, thus making them adaptive and robust in nature (Lavretsky and Wise, 2013). The merit in adopting the Youla parameterization approach to robust control design has been been successfully demonstrated for automotive applications, especially for robust observer and estimation design (Assadian et al., 2018; Liu et al., 2019).

4.1 Youla Parameterization

Youla parameterization derives its name from the Youla parameter, $Y(s)$, which is defined as:

$$Y(s) \triangleq \frac{\hat{u}}{\hat{r}} = \frac{G_c}{1 + L(s)} \tag{4.1}$$

where $G_c(s)$ is the controller transfer function and $L(s)$ is the return ratio of the closed-loop shown in Figure 4.1. $\hat{u}$ is the controller output and $\hat{r}$ is the reference signal.

Additional transfer functions that are important in the Youla parameterization framework are the sensitivity transfer function, $S(s)$, and the complementary-sensitivity transfer function, $T(s)$, which are defined as follows:

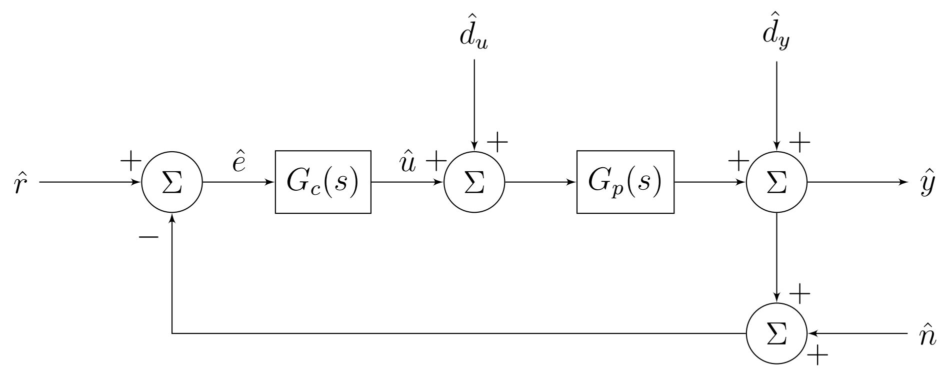

Figure 4.1: SISO unity-feedback loop with external disturbances and sensor noise (Assadian and Mallon, 2021)

$$S(s) \triangleq \frac{\hat{e}}{\hat{r}} = \frac{1}{1 + L(s)} \tag{4.2}$$

where $\hat{e}$ is the error signal, and

$$T(s) \triangleq \frac{\hat{y}}{\hat{r}} = \frac{L(s)}{1 + L(s)} \tag{4.3}$$

where $\hat{y}$ is the output signal.

The three transfer functions allow for direct shaping of closed-loop responses to all the inputs shown in Figure 4.1. The relationships are as follows:

$$\begin{pmatrix} \hat{e} \\ \hat{y} \\ \hat{u} \end{pmatrix} = \begin{pmatrix} S(s) & -S(s) & -S(s) & -G_p S(s) \\ T(s) & -T(s) & S(s) & G_p S(s) \\ Y(s) & -Y(s) & -Y(s) & -T(s) \end{pmatrix} \begin{pmatrix} \hat{r} \\ \hat{n} \\ \hat{d}_y \\ \hat{d}_u \end{pmatrix} \tag{4.4}$$

where $\hat{d}_y$ is the output disturbance and $\hat{d}_u$ is the controller output disturbance.

From these relationships, important points with respect to the physical aspect of systems can be noted about the three transfer functions—$S(s)$, $T(s)$, and $Y(s)$. $S(s)$ represents how sensitive the output of the feedback system is to output disturbances. $T(s)$ is the closed-loop transfer function and represents the behavior of the system to reference signals. $Y(s)$ represents the actuator output based on the reference signals.

These three transfer functions also provide information about internal stability. A system is internally stable if the following three conditions are met:

1. The Youla parameter, $Y(s)$, is BIBO stable.

2. $G_p(s)\, Y(s)$, or $T(s)$, is BIBO stable.

3. $G_p(s)\,(1 - G_p(s)Y(s))$, or $G_p(s)\, S(s)$, is BIBO stable.

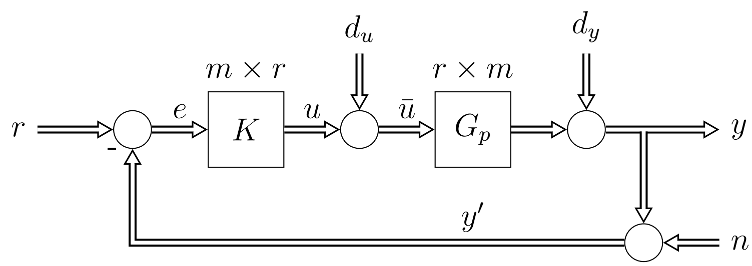

Figure 4.2: A MIMO feedback system [8]

The following transfer function matrix relationships can be derived for the MIMO system with return ratios $L_y = G_pG_c$ and $L_u = G_cG_p$ depicted in Figure 4.2:

- Output Complementary-Sensitivity, T_y (from r to y)

$$\begin{aligned} y &= \left(I + G_pG_c\right)^{-1} G_pG_c r \\ &= T_y r \\ \text{where,} \quad T_y &= (I + L_y)^{-1} L_y \end{aligned}$$

- The Youla Parameter, Y (from r to $\bar{u}$)

$$\begin{aligned} \bar{u} &= \left(I + G_cG_p\right)^{-1} G_c r \\ &= Y r \\ \text{where,} \quad Y &= (I + L_u)^{-1} G_c \end{aligned}$$

- Output Sensitivity, S_y (from d_y to y)

$$\begin{aligned} y &= (I + L_y)^{-1} d_y \\ &= \left(I + G_pG_c\right)^{-1} G_p d_u \\ &= S_y G_p d_u \\ \text{where,} \quad S_y &= (I + L_y)^{-1} \end{aligned}$$

- Input Sensitivity, S_u (from d_u to $\bar{u}$)

$$\begin{aligned}\bar{u} &= \left(I + G_c G_p\right)^{-1} d_u \\ &= S_u d_u \\ \text{where,} \quad S_u &= (I + L_u)^{-1}\end{aligned}$$

For the closed-loop system to be internally stable, these 4 independent transfer function matrices need to be bounded-input bounded-output (BIBO) stable. The transfer function matrix relationships can be consolidated into one equation as follows:

$$\begin{pmatrix} \bar{u} \\ y \end{pmatrix} = \begin{pmatrix} Y & S_u \\ T_y & S_y G_p \end{pmatrix} \begin{pmatrix} r \\ d_u \end{pmatrix} \tag{4.5}$$

Expressing Equation 4.5 in terms of the Youla parameter, Y:

$$\begin{pmatrix} \bar{u} \\ y \end{pmatrix} = \begin{pmatrix} Y & I - YG_p \\ G_p Y & G_p \left(I - YG_p\right) \end{pmatrix} \begin{pmatrix} r \\ d_u \end{pmatrix} \tag{4.6}$$

The closed-loop transfer function matrix in Equation 4.6 is completely parameterized by Y and G_p. Each entry of the matrix has to be BIBO stable in order to guarantee internal stability of the closed-loop system. Further, if G_p is stable, then the closed-loop is internally stable if and only if Y is stable. All internally stabilizing compensators can be parameterized by G_p and Y as follows:

$$G_c = (I - YG_p)^{-1} Y \tag{4.7}$$

A novel, systematic approach to design internally stabilizing, robust multivariable feedback controllers using Youla parameterization, developed by Dr. Francis F. Assadian at the University of California, Davis, is described in the following algorithm.

Algorithm 1 MIMO Robust Control Design via Youla Parameterization: *The FFA Method*

procedure MIMO(G_p) ▷ Plant TFM
  compute the SM-form of $G_{p_{(m \times n)}}$, $M_{p_{(m \times n)}}$ ▷ Smith-McMillan Form
  find unimodular U_L, $U_R \mid M_p = U_L G_p U_R$
  choose M_Y to shape $M_T \mid M_T = M_p M_Y$
  check: $Y = U_R M_Y U_L \rightarrow$ all entries of Y are proper TFs ▷ Youla TFM
  for $k = 1 : \min\{m, n\} \mid M_p(k, k) = G_k$ **do**
    SISO(G_k) ▷ Internal Stability
  end for
  compute $T_y = U_L^{-1} M_p M_Y U_L$ ▷ Complementary-Sensitivity TFM
  compute $S_y = I - T_y = U_L^{-1}(I - M_p M_Y) U_L$ ▷ Sensitivity TFM
  compute $G_c = U_R (I - M_Y M_p)^{-1} M_Y U_L$ ▷ Controller TFM
  check: $S_y G_p = U_L^{-1}(I - M_p M_Y) M_p U_R^{-1} \rightarrow$ meets requirement ▷ Loop-Shaping Check
  return G_c
end procedure
function SISO(G) ▷ SISO Interpolation Conditions
  if G has a $j\omega$-axis or RHP pole p of multiplicity a_p **then** ▷ Unstable Poles
    $S(p) = 0; T(p) = 1$
    for $j = 1 : a_p - 1$ **do**
      $\frac{d^j S(p)}{ds^j} = 0;\ \frac{d^j T(p)}{ds^j} = 0$
    end for
  end if
  if G has an NMP zero z of multiplicity a_z **then** ▷ Non-Minimum Phase Zeros
    $S(z) = 1; T(z) = 0$
    for $j = 1 : a_z - 1$ **do**
      $\frac{d^j S(z)}{ds^j} = 0;\ \frac{d^j T(z)}{ds^j} = 0$
    end for
  end if
end function

4.2 Multivariable Feedback Control

Following the procedure described in the control design algorithm, we find the Smith-McMillan form, M_p, of the linear plant, G_p, as shown in Equation 4.8. It is observed from the Smith-McMillan form that the plant is highly unstable, with 5 BIBO unstable poles and one stable pole. The corresponding left and right unimodular matrices, shown in Equations 4.9 and 4.10 respectively, are computed such that $M_p = U_L G_p U_R$. It can be verified that the determinants of the unimodular matrices are scalars.

$$M_p = \begin{pmatrix} \frac{1}{s^2\,(s-0.8626)\,(s+0.8626)} & 0 \\ 0 & \frac{1}{s^2} \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} G_1(s) & 0 \\ 0 & G_2(s) \\ 0 & 0 \end{pmatrix} \tag{4.8}$$

$$U_L = \begin{pmatrix} 0 & 0.5824 & -2.7865 \\ -5.499 & 0 & 0 \\ 0 & 0.8542\,s^2 & -4.0869\,(s-1.082)\,(s+1.082) \end{pmatrix} \tag{4.9}$$

$$U_R = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{4.10}$$

The entries of M_p (Equation 4.8) are treated as individual SISO (single-input single-output) *plants*: $G_1(s) = Mp_{11}$ and $G_2(s) = Mp_{22}$, both of which are BIBO unstable. The closed-loop (complementary-sensitivity) transfer functions, $T_1(s)$ and $T_2(s)$, of $G_1(s)$ and $G_2(s)$ respectively, are defined as follows: $T_1(s) = Y_1(s)G_1(s)$ and $T_2(s) = Y_2(s)G_2(s)$, where $Y_1(s)$ and $Y_2(s)$ are the Youla parameters that are to be designed so as to stabilize the entries of M_p and ensure that $T_1(s)$ and $T_2(s)$ meet the interpolation conditions for internal stability. The interpolation conditions are described as a function on line 15 in the algorithm.

The interpolation conditions for the unstable poles of G_1 are:

$$T_1(s)\Big|_{s=0.8626} = 1, \qquad T_1(s)\Big|_{s=0} = 1, \qquad \frac{dT_1(s)}{ds}\Big|_{s=0} = 0 \tag{4.11}$$

The designed Youla parameter that satisfies these conditions (4.11) is shown in Equation 4.12.

$$Y_1(s) = \frac{K\, s^2\,(s-0.8626)\,(s+0.8626)\,(\tau_{z_1}s+1)\,(\tau_{z_2}s+1)}{(s^2+2\zeta\omega_n s+\omega_n^2)\,\left(\tau_{p_1}s+1\right)\,\left(\tau_{p_2}s+1\right)\,\left(\tau_{p_3}s+1\right)^4} \tag{4.12}$$

The chosen parameter values for $Y_1(s)$ are listed in Table 4.1, where τ_{z_1}, τ_{z_2}, and τ_{p_1} are the solution variables chosen to satisfy the system of equations given by the interpolation conditions (4.11). The fourth-order pole with time-constant τ_{p_3} is included to ensure that $Y_1(s)$ and $T_1(s)$ are proper transfer functions, and also to lower the magnitude of the actuator effort (the frequency response of the Youla parameter) at high frequencies.

ω_n	3.75	Natural frequency of the second-order pole [rad/s]
K	14.0625	Gain $({\omega_n}^2)$
ζ	0.7071	Damping ratio $(1 \div \sqrt{2})$
τ_{z_1}	2.1198	Time-constant of the first zero [s] (solved for)
τ_{z_2}	1.8256	Time-constant of the second zero [s] (solved for)
τ_{p_1}	3.1416	Time-constant of the first pole [s] (solved for)
τ_{p_2}	0.0267	Time-constant of the second pole [s] $(1 \div 10\omega_n)$
τ_{p_3}	0.1	Time-constant of the third pole (fourth-order) [s]

Table 4.1: Parameter values of the Youla transfer function, $Y_1(s)$

The resulting stable closed-loop (complementary-sensitivity) transfer function, $T_1(s)$, is shown in Equation 4.13.

$$T_1(s) = \frac{K\,(\tau_{z_1}s+1)\,(\tau_{z_2}s+1)}{(s^2+2\zeta\omega_n s+\omega_n^2)\,\left(\tau_{p_1}s+1\right)\,\left(\tau_{p_2}s+1\right)\,\left(\tau_{p_3}s+1\right)^4} \tag{4.13}$$

In order to ensure good reference tracking, it is desirable to make all the closed-loop transfer functions associated with the entries of the Smith-McMillan form of the MIMO plant equal (for a MIMO plant with a square transfer function matrix, this would result in the decoupling of the MIMO closed-loop transfer function matrix). The chosen Youla parameter, $Y_2(s)$, that satisfies the interpolation conditions for $T_2(s)$ and ensures that $T_2(s)$ is equal to $T_1(s)$ is shown in Equation 4.14.

$$Y_2(s) = \frac{K\, s^2\,(\tau_{z_1}s+1)\,(\tau_{z_2}s+1)}{(s^2+2\zeta\omega_n s+\omega_n^2)\,\left(\tau_{p_1}s+1\right)\,\left(\tau_{p_2}s+1\right)\,\left(\tau_{p_3}s+1\right)^4} \tag{4.14}$$

Using the Youla parameters defined in Equations 4.12 and 4.14, the transfer function matrix, M_Y, is formulated, as shown in Equation 4.15.

$$M_Y = \begin{pmatrix} Y_1(s) & 0 & 0 \\ 0 & Y_2(s) & 0 \end{pmatrix} \tag{4.15}$$

The transfer function matrix with the individual closed-loop transfer functions, M_T, is shown in Equation 4.16, where $T_1(s) = T_2(s)$.

$$M_T = M_p M_Y = \begin{pmatrix} Y_1(s)G_1(s) & 0 & 0 \\ 0 & Y_2(s)G_2(s) & 0 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} T_1(s) & 0 & 0 \\ 0 & T_2(s) & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{4.16}$$

The singular values of M_Y, M_T, and M_S are shown in Figure 4.3, where $M_S = I - M_T, I \in \mathbb{R}^{3\times 3}$.

A summary of the closed-loop requirements for M_Y, M_T, and M_S is given in Table 4.2, where σ_{max} and σ_{all} indicate the maximum singular value and all singular values, respectively.

$\sigma_{all}(M_T) = 1$ at low frequencies	Reference (r) tracking
$\sigma_{max}(M_T) \ll 0$ at high frequencies	Sensor noise (n) rejection, robust stability
$\sigma_{max}(M_S) \ll 0$ at low frequencies	Disturbance (d_u, d_y) rejection
$\sigma_{max}(M_Y) \ll 0$ at high frequencies	Small control effort (u)

Table 4.2: Closed-loop requirements for M_Y, M_T, and M_S (see Figure 4.2)

The designed transfer function matrices, M_Y, M_T, and M_S, are pre- and post-multiplied by the unimodular matrices, as described in the algorithm, to obtain the stable MIMO

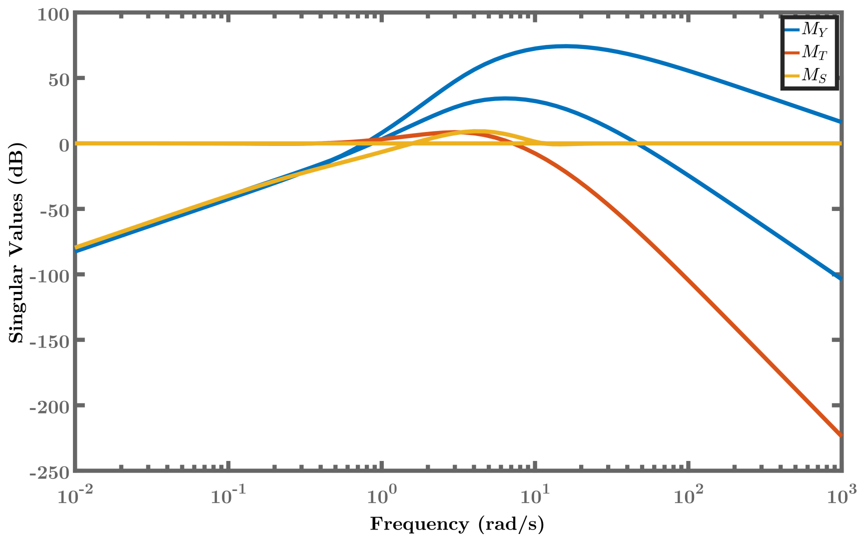

Figure 4.3: Singular values of M_Y, M_T, and M_S

Youla transfer function matrix, Y, the stable MIMO closed-loop (complementary-sensitivity) transfer function, T_y, the stable MIMO sensitivity transfer function, S_y, and the internally stabilizing robust MIMO controller, G_c. The singular values of Y, T_y, and S_y, are shown in Figure 4.4. The closed-loop requirements listed in Table 4.2 hold for the coupled transfer functions matrices as well.

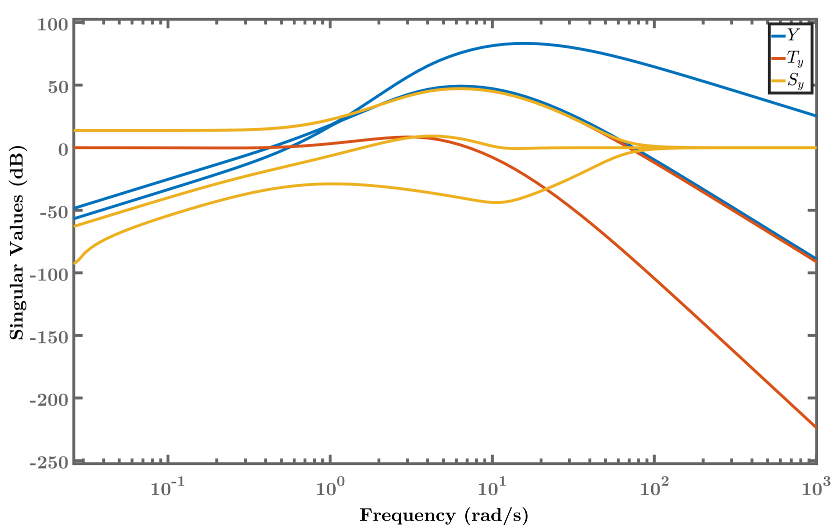

Figure 4.4: Singular values of Y, T_y, and S_y

It is observed that not all the closed-loop requirements are satisfied by the closed-loop system due to coupling (presence of non-diagonal terms in the transfer function matrices) introduced by the unimodular matrices. For instance, the maximum singular value of T_y has a magnitude of 13.8 dB at low frequencies, which is not ideal for reference tracking. Further, robustness analysis is performed on the closed-loop system by introducing uncertainty in the gains of the plant transfer function matrix, G_p. The singular values of G_p with 50% uncertainty in its gains ($G_{p,\,uncertain}$) are shown in Figure 4.5, as an example.

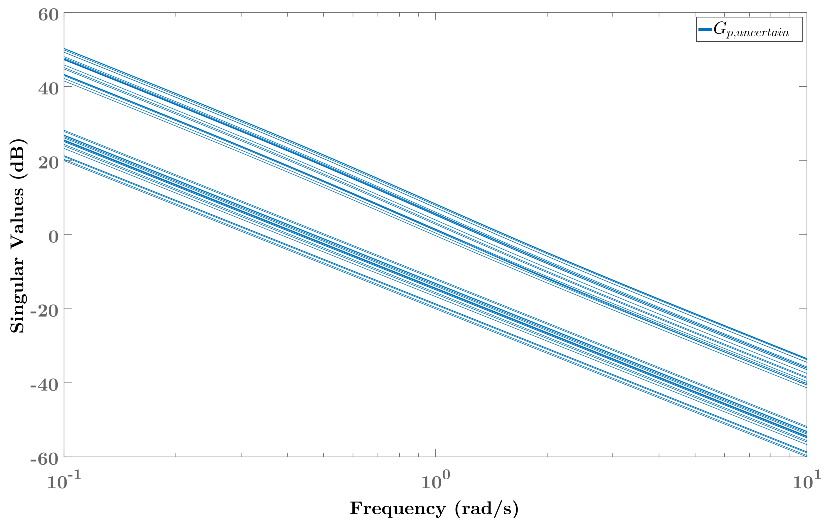

Figure 4.5: Singular values of G_p with 50% uncertainty in its gains

The resulting singular values of Y, T_y, and S_y, are shown in Figure 4.6. It is observed that the closed-loop system is robust to even 50% uncertainty in the plant transfer function matrix gains, especially at low frequencies.

MATLAB and Simulink were used for control design and robustness analysis.

The methodology adopted for control design enables complete stabilization of the closed-loop system, while also ensuring robustness to system parameter uncertainties, as shown in Figure 4.6. The linear model of the lander with the designed feedback controller tracks the reference altitude and downrange trajectories, as shown in Figure 4.7 and Figure 4.8, respectively. This closed-loop system, however, displays oscillatory behavior for pitch reference tracking, as shown in Figure 4.9. One of the reasons for this is that all the available tuning parameters are exhausted in meeting the interpolation conditions for internal stability of the closed-loop system. The existence of tight margins for parameter variations in terms of stability renders the tuning of the existing parameters challenging as well. Given the stability of the closed-loop system, however, the responses could be tuned by either outer-loop

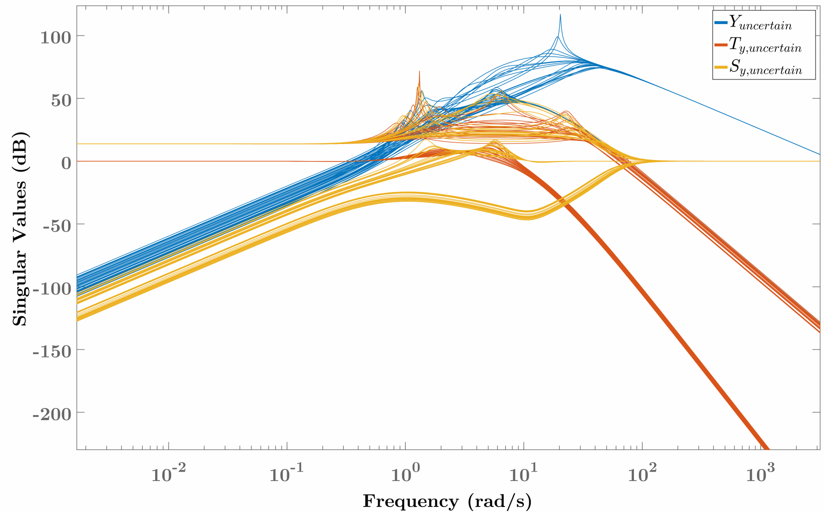

Figure 4.6: Singular values of Y, T_y, and S_y with 50% uncertainty in the gains of G_p

controllers or pre-filters (Alavi et al., 2005) or both. Another solution is to implement a combined feedforward-feedback control architecture with dynamic control allocation to handle underactuation, effectively decoupling the entire system and enabling robust trajectory tracking, as is described in Section 5.1.

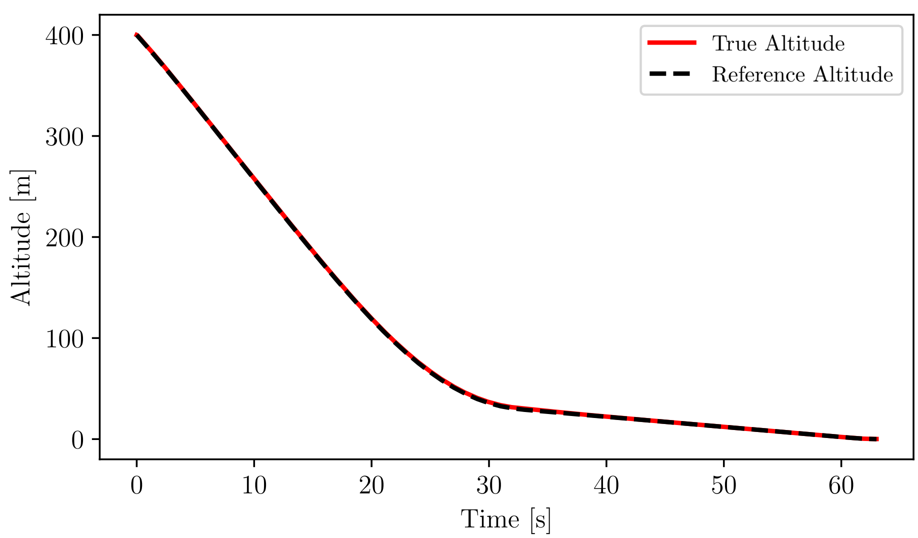

Figure 4.7: Tracking of the reference altitude trajectory by the linear model with feedback control

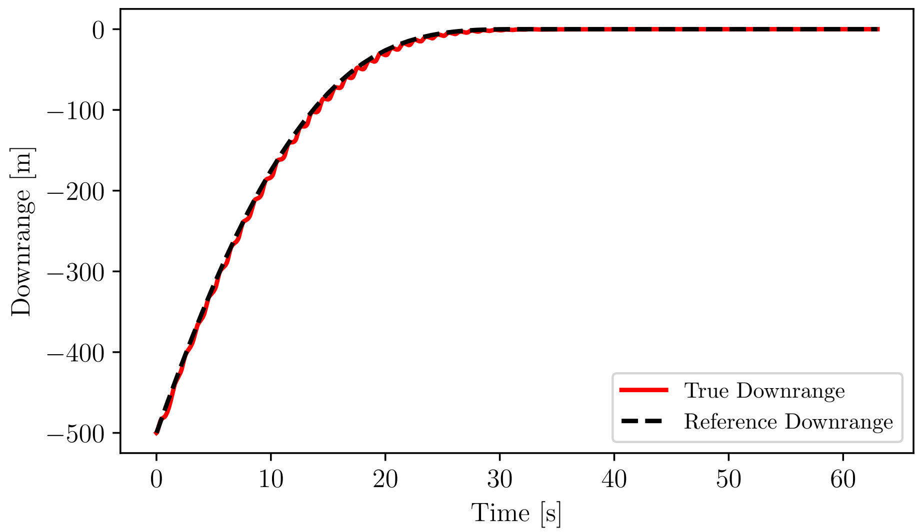

Figure 4.8: Tracking of the reference downrange trajectory by the linear model with feedback control

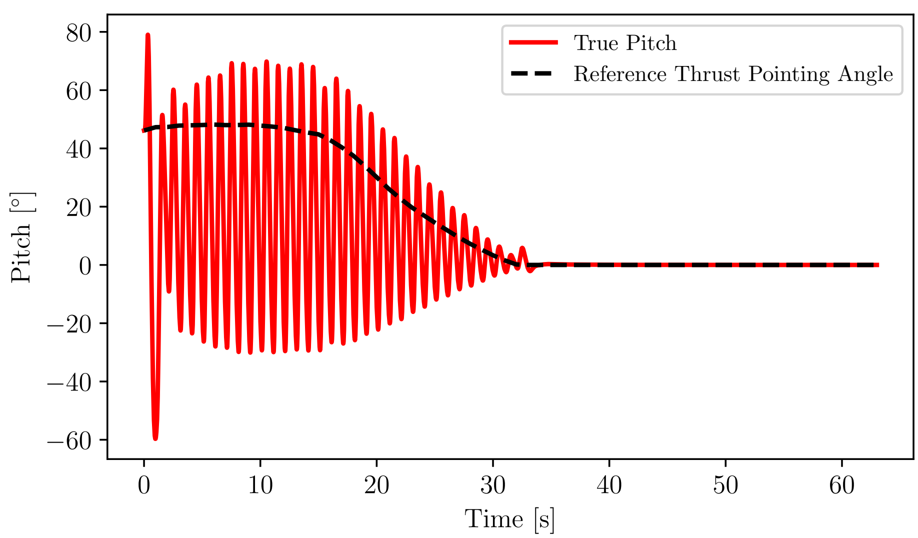

Figure 4.9: Tracking of the reference thrust pointing angle trajectory (proxy for reference pitch) by the linear model with feedback control

Chapter 5

Simulation

5.1 Feedforward-Feedback Control Architecture

A combined feedforward-feedback architecture with control allocation is implemented in order to address the shortcomings of the proposed multivariable control design described in 4.2. The feedforward signals are generated by the guidance algorithm adopted. Specifically, they include the thrust magnitudes along the inertial vertical and horizontal axes (for the planar landing scenario). These feedforward commands enable accurate translation-trajectory tracking, and tight adherence of the closed-loop trajectories to the constraints imposed in the powered-descent guidance algorithm adopted (2.1).

The feedback controllers that control the translation of the lander (altitude and downrange, for the planar landing scenario), are designed so as to null out the error between the desired trajectories and the actual trajectories of the lander. For attitude control, however, the adopted PDG algorithm does not generate any feedforward torque commands. Hence, the entirety of attitude control is handled in feedback. The thrust pointing angle (the angle of the feedforward thrust vector from the vertical) is used as a proxy for the reference attitude (pitch, for the planar landing scenario) trajectory. A feedback controller is then designed to generate torque commands to control the attitude of the lander. Thus, for the planar landing scenario, there are finally three control commands that are generated (vertical and horizontal thrust, and torque), but only two actuators (the mass-flow rate and the gimbal pitch angle) to provide the required control authority. Control allocation is implemented in order to perform this mapping of control commands in the inertial frame to actuator commands in the lander body frame.

Similar control strategies have been adopted in the context of powered-descent and landing in the literature. The G-FOLD algorithm was successfully tested in real-time onboard the Masten Xombie vehicle, with PID feedback controllers to close the loop (Açıkmeşe et al., 2013). More recently, the EmboRockETH model rocket was successfully flight-tested with MPC for position control, and PID controllers for attitude and attitude rate control (Spannagl et al., 2021). These formulations are tabulated in Table 5.1.

Here, the algorithm described in 2.1 is used to generate reference translation and attitude trajectories and feedforward control commands, and a control strategy using Youla parameterization-based linear parameter-varying (LPV) robust feedback controllers is implemented in simulation, along with the complete nonlinear model developed in Chapter 3. The simulation results obtained demonstrate feasibility of the closed-loop trajectories with respect to the imposed constraints, and potential for implementation in real-time, given the simplicity of the resulting controller structure.

Vehicle/Model	Guidance Formulation	Control Design Method
Masten Xombie [3]	Variable mass	PID
EmboRockETH [66]	Constant mass	MPC, PID
This work	Variable mass	Youla parameterization

Table 5.1: Control architectures for powered-descent with convex optimization-based 3-DoF guidance

5.1.1 Plant Model for Control Design

For the purpose of control design, a simpler model of the lander is considered (Equation 5.1). This model represents equations of motion that are completely decoupled in translation and rotation—it can be considered to be one of a variable-mass rigid body system, with thrust always passing through the CM (without gimbaling) and with independent attitude control.

$$F_x(t) = (m(u_m) + M)\,(a_x(t) + g) \tag{5.1a}$$

$$F_z(t) = (m(u_m) + M)\, a_z(t) \tag{5.1b}$$

$$\tau_y(t) = I_{yy}(m)\, \alpha_y(t) \tag{5.1c}$$

Here[1],

F_x, F_z are the thrust components along the inertial x and z axes

τ_y is the torque about the inertial (and body) y axis

a_x, a_z are the acceleration components along the inertial x and z axes

α_y is the angular acceleration about the inertial (and body) y axis

The remaining terms in Equations 5.1 are described in Section 3.1. Equations 5.1 can be reformulated in terms of the states $\{v_x,\ v_z,\ \omega_y,\ d_x,\ d_z,\ q_y,\ m\}$, as shown in Equations 5.2. These state equations will consitute the *decoupled* parameter-varying model for further developments[2].

$$\dot{v}_x = \frac{F_x}{M+m} - g \tag{5.2a}$$

$$\dot{v}_z = \frac{F_z}{M+m} \tag{5.2b}$$

$$\dot{\omega}_y = \frac{\tau_y}{I_{yy}} \tag{5.2c}$$

$$\dot{d}_x = v_x \tag{5.2d}$$

$$\dot{d}_z = v_z \tag{5.2e}$$

$$\dot{q}_y = \omega_y \tag{5.2f}$$

$$\dot{m} = -\frac{\|F_{xz}\|}{C} \tag{5.2g}$$

[1]The arguments for time (t), control (u_m), and propellant mass (m) are dropped henceforth for notational simplicity.

[2]For notational clarity, v_2 and q_2 (as described in Section 3.1) are replaced in this section by ω_y and q_y, respectively.

Additionally, the lever-arm (the distance between the main engine gimbal hinge point and the vehicle mass-center), l_{CM}, can be given by Equation 5.3.

$$l_{CM} = \frac{l(2M+m)}{4(M+m)} \tag{5.3}$$

The thrust component at the main engine gimbal hinge point, along the body z axis, is given by Equation 5.4.

$$F_z^b = \frac{\tau_y}{l_{CM}} \tag{5.4}$$

Therefore, the gimbal angle[3] in the body frame, e_b, is given by Equation 5.5.

$$e^b = -\arcsin\left(\frac{F_z^b}{\|F_{xz}\|}\right), \quad -1 \leq \frac{F_z^b}{\|F_{xz}\|} \leq 1 \tag{5.5}$$

5.1.2 Feedback Control Design

This section highlights the development of feedback controllers that enable accurate tracking of the altitude, downrange, and pitch trajectories generated by guidance. The *outputs* considered are altitude, downrange, and pitch. The *inputs* are the thrust component along the vertical axis, the thrust component along the longitudinal axis, and the torque about the lateral axis, all in the inertial frame.

In order to reveal the direct relationship between the inputs and outputs considered, Equations 5.2 can be rewritten in terms of the outputs only, as shown in Equations 5.6.

$$\ddot{d}_x = \frac{F_x}{M+m} - g \tag{5.6a}$$

$$\ddot{d}_z = \frac{F_z}{M+m} \tag{5.6b}$$

$$\ddot{q}_y = \frac{\tau_y}{I_{yy}} \tag{5.6c}$$

[3] The gimbal angle, e_b, is defined in the body frame and is different from e_1 (described in Section 3.1), which is defined in the inertial frame: $e_b = e_1 - q_2$

The gravity term in Equation 5.6a is considered to be an external disturbance, and is ignored in the following feedback control development. This assumption is made in order to obtain a linear representation of the system (Equation 5.6a is affine). It is to be noted that this assumption is fairly innocuous, given that the feedforward commands from guidance, which take gravity into account, possess most of the control authority with respect to altitude (and downrange) control; the purpose of feedback control for altitude (and downrange) trajectory-tracking is only for stabilization of the plant and the correction of errors that may be introduced due to model-mismatch, external disturbances, and sensor noise.

Ignoring gravity, taking the Laplace transform of Equations 5.2, and setting the initial conditions to zero, we get Equations 5.7.

$$s^2 D_x(s) = \left(\frac{1}{M+m}\right) \mathcal{F}_x(s) \tag{5.7a}$$

$$s^2 D_z(s) = \left(\frac{1}{M+m}\right) \mathcal{F}_z(s) \tag{5.7b}$$

$$s^2 Q_y(s) = \left(\frac{1}{I_{yy}}\right) \mathcal{T}_y(s) \tag{5.7c}$$

Therefore, the transfer functions of the plant for the purpose of feedback control design are given by Equations 5.8. These plant transfer functions are BIBO unstable, with repeated (double) poles at the origin of the s plane.

$$\frac{D_x(s)}{\mathcal{F}_x(s)} = \underbrace{\left(\frac{1}{M+m}\right)}_{\text{translation gain: } K_t(m)} \frac{1}{s^2} \tag{5.8a}$$

$$\frac{D_z(s)}{\mathcal{F}_z(s)} = \overbrace{\left(\frac{1}{M+m}\right)} \frac{1}{s^2} \tag{5.8b}$$

$$\frac{Q_y(s)}{\mathcal{T}_y(s)} = \underbrace{\left(\frac{1}{I_{yy}}\right)}_{\text{attitude gain: } K_a(m)} \frac{1}{s^2} \tag{5.8c}$$

Youla Parameterization-based Controller Design

As described in Subsection 4.2, Youla parameterization provides a systematic framework for the design of internally stabilizing controllers using algebraic manipulation (Assadian and Mallon, 2021; Blanchini et al., 2010; Kučera, 2011).

Feedback controllers are designed around the BIBO unstable plant transfer functions in Equations 5.8. The internally stabilizing controllers thus designed are linear parameter-varying (LPV)[4] in nature, due to the varying gains that are functions of the propellant mass. The design procedure has been described in the form of an algorithm (Algorithm 2), which includes the steps for an offline control design procedure (and is described as such for the sake of clarity)—the resulting controller structure and parameters other than the gain, $\frac{1}{K(m)}$, do not need to be modified or updated during flight. The closed-loop transfer function, $T(s)$, is time-invariant by design. The estimated propellant mass from the control commands (see lines 1 and 2 in Algorithm 3) are used to update the gain at each time-step.

The design parameters of the translation and attitude controllers for the lunar landing scenario are shown in Tables 5.2 and 5.3, respectively. The controller parameters $\{\omega_b, \zeta, \tau_p\}$ (prior to being scaled by the gains) were chosen via closed-loop testing and manual tuning. Given that the feedforward commands possess most of the translation control authority and are responsible for trajectory-tracking, and that the feedback controllers are for error-cleaning only, the translation controller responses are further scaled down to $\frac{10}{\sqrt{2}}\%$ their original values (such that the total feedback control thrust magnitude is scaled down to 10% the commanded feedback control thrust magnitude). In practice, the varying gain, $K(m)$, is set to unity during the design phase (Algorithm 2), and the control command is scaled in real-time by the actual gain that is based on the mass estimate.

[4]Although the propellant mass varies with time, it is dependent on a control input (mass-flow rate), the profile of which is not known a priori. Hence, the controllers are referred to as linear parameter-varying (LPV) systems as opposed to linear time-varying (LTV) systems.

Algorithm 2 LPV Control Design

Inputs: ω_b, ζ, $K(m)$, τ_p

1: $T^{\star}(s) = \dfrac{\texttt{wn}^2\,(\texttt{tz}\,s+1)}{(s^2+2\zeta\,\texttt{wn}\,s+\texttt{wn}^2)(\tau_p s+1)} \Bigg| \; T^{\star}(0)=0$ ▷ desired $T(s)$ | Alg. 1, line 17

2: $\texttt{eqn\_1} \longleftarrow \left.\dfrac{dT^{\star}(s)}{ds}\right|_{s=0} == 0$ ▷ Alg. 1, line 19

3: $\texttt{eqn\_2} \longleftarrow \left|T^{\star}(j\omega)\right|_{\omega=\omega_b} == \dfrac{1}{\sqrt{2}}$ ▷ closed-loop 3 dB bandwidth enforcement

4: $\texttt{wn, tz} \longleftarrow \texttt{solve(eqn\_1, eqn\_2)}$

5: **if** phase margin $\geq 60°$ **then** ▷ classical robustness

6: $\quad\omega_n,\ \tau_z = \texttt{wn, tz}$

7: **else**

8: $\quad$choose different τ_p and repeat (from line 1)

9: **end if**

$$G_p(s,m) = \frac{K(m)}{s^2}$$ ▷ LPV plant

10: $Y(s,m) = \dfrac{\omega_n^2\, s^2\,(\tau_z s+1)}{K(m)\,(s^2+2\zeta\omega_n s+\omega_n^2)(\tau_p s+1)}$ ▷ LPV Youla parameter

11: $T(s) = Y(s,m)\,G_p(s,m) = \dfrac{\omega_n^2\,(\tau_z s+1)}{(s^2+2\zeta\omega_n s+\omega_n^2)(\tau_p s+1)}$ ▷ complementary-sensitivity

12: $S(s) = 1 - T(s) = \dfrac{s^2\,(s\,\tau_p+2\zeta\omega_n\tau_p+1)}{(s^2+2\zeta\omega_n s+\omega_n^2)(\tau_p s+1)}$ ▷ sensitivity

13: $G_c(s,m) = \dfrac{Y(s,m)}{S(s)} = \dfrac{\omega_n^2\,(\tau_z s+1)}{K(m)\,(s\,\tau_p+2\zeta\omega_n\tau_p+1)}$ ▷ LPV controller

Return: $G_c(s,m)$

Chosen:		
ω_b	0.71	Closed-loop bandwidth [rad/s]
ζ	0.7071	Damping ratio $(1 \div \sqrt{2})$
$K(m)$	$K_t(m)$ (see Eq. 5.8)	Varying gain
τ_p	0.05	Time-constant of the first-order pole [s]
Solved for:		
ω_n	0.3415	Natural frequency of the second-order pole [rad/s]
τ_z	4.1913	Time-constant of the first-order zero [s]

Table 5.2: Translation control design parameters

Chosen:		
ω_b	3.55	Closed-loop bandwidth [rad/s]
ζ	0.7071	Damping ratio $(1 \div \sqrt{2})$
$K(m)$	$K_a(m)$ (see Eq. 5.8)	Varying gain
τ_p	0.01	Time-constant of the first-order pole [s]
Solved for:		
ω_n	1.7074	Natural frequency of the second-order pole [rad/s]
τ_z	0.8383	Time-constant of the first-order zero [s]

Table 5.3: Attitude control design parameters

Both the translation and attitude controllers result in a phase margin of 64.1 degrees. The singular value (Bode magnitude) plots of the Youla parameter, $Y(s)$, the closed-loop transfer function, $T(s)$, and the sensitivity transfer function, $S(s)$, for the translation and attitude controllers are shown in Figures 5.1 and 5.2 respectively. It can be observed that the frequency responses satisfy the closed-loop requirements listed in Table 4.2. Further, the maximum singular value of $Y(s)$ at high frequencies can be brought down by introducing additional poles in the transfer function as required. The resulting controllers are first-order transfer functions.

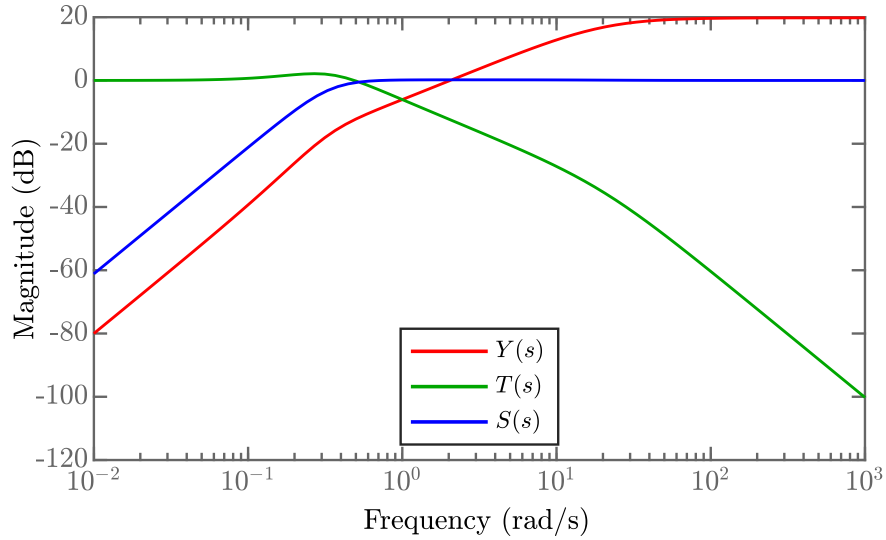

Figure 5.1: Translation loop (linear) frequency responses

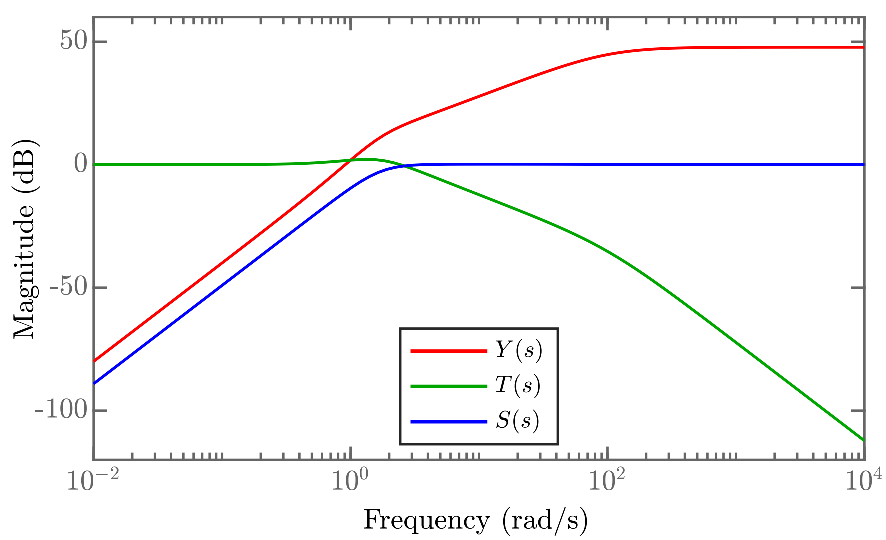

Figure 5.2: Attitude loop (linear) frequency responses

5.1.3 Control Allocation

Given that the actual system is underactuated, the three control commands $\{F_x, F_z, \tau_y\}$ need to be mapped to two actuator inputs $\{\dot{m}, e^b\}$, as shown in Figure 5.3 and described in Algorithm 3.

Figure 5.3: The control allocator block (see Figure 5.4)

Algorithm 3 Control Allocation

Inputs: F_x, F_z, τ_y

1: Compute the mass-flow rate, $\dot{m}$, from the thrust commands ▷ Equation 5.2g
2: Integrate the mass-flow rate to obtain the propellant mass, m
3: Compute the lever-arm (a function of the propellant mass) ▷ Equation 5.3
4: Compute the horizontal thrust component in the body frame, F_z^b ▷ Equation 5.4
5: Compute the gimbal angle in the body frame, e^b ▷ Equation 5.5

Return: $\dot{m}$, e^b

5.2 Closed-Loop Simulation

5.2.1 Framework

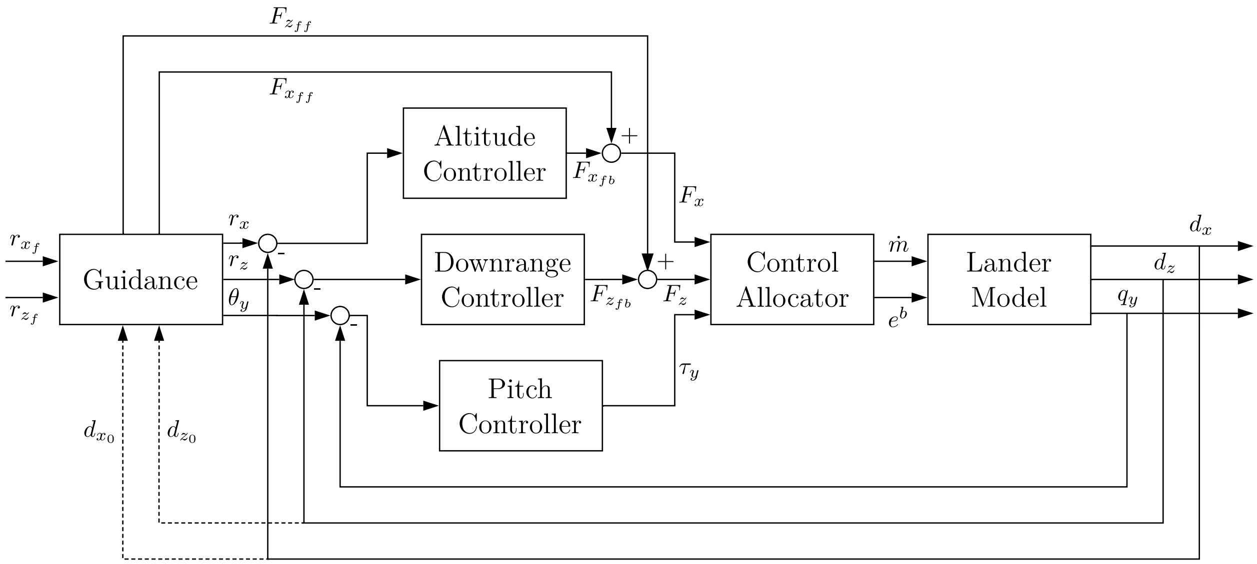

Figure 5.4: Powered-descent guidance and control architecture

The closed-loop planar powered-descent guidance and control simulation is set up as shown in Figure 5.4, along with the nonlinear model developed in Chapter 3.

The guidance block (Figure 2.1) is given the coordinates of the target landing site and the initial conditions of the lander. The resulting convex optimization problem solved, along with the imposed constraints and boundary conditions, is described in Section 2.2. The guidance output is a one-shot solution (in this case, the optimization problem is solved once at the beginning of the maneuver; in practice, however, if the need arises, the problem can be re-solved as required to obtain new trajectories on the fly, given its amenability to real-time implementation). The solution consists of two components: the reference translation trajectories for the lander to track, $\{r_x, r_z\}$, and the feedforward thrust commands, $\{F_{x_{ff}}, F_{z_{ff}}\}$. In addition, the thrust pointing angle from guidance, θ_y, is used as a proxy for the reference pitch trajectory.

The outputs considered are the altitude, d_x, the downrange, d_z, (translation) and the pitch angle of the lander, q_y (attitude). The errors between the outputs fed back and the reference trajectories are then passed through the feedback controllers, which generate the necessary feedback control commands, $\{F_{x_{fb}},\, F_{z_{fb}},\, \tau_y\}$. For translation, the corrective feedback control commands, $\{F_{x_{fb}},\, F_{z_{fb}}\}$, are then combined with the respective feedforward thrust commands from guidance, $\{F_{x_{ff}},\, F_{z_{ff}}\}$, to generate the closed-loop thrust commands, $\{F_x,\, F_z\}$. Since no feedforward torque commands are generated by guidance, the entirety of attitude control is handled by the attitude controller, which generates the necessary feedback torque command, τ_y. Since the lander is underactuated, a control allocator (Figure 5.3) is used to map the inertial control commands, $\{F_x,\, F_z,\, \tau_y\}$, to the actuator inputs, $\{\dot{m},\, e_b\}$, as described in Algorithm 3. The commands thus computed are passed on to the nonlinear lander model block (Figure 3.4). The closed-loop responses of the nonlinear lander model are then fed back, and this entire process is repeated at each simulation time-step.

5.2.2 Actuator Considerations

Since the constraints imposed in guidance apply to the feedforward control commands only, additional constraints need to be imposed in order to ensure that the closed-loop signals do not exceed their physical bounds. Moreover, given that the lander is inherently unstable, underactuated, and that it has only one throttlable gimbaled main engine for both stabilization and trajectory-tracking, additional constraints are required on the control commands to ensure satisfaction of the closed-loop requirements.

The guidance algorithm generates optimal solutions that constrain the thrust magnitude to throttle between 30% and 80% of the full-thrust magnitude, T_{max}. These throttle limits are intentionally set to more conservative values than the actual physical throttle limits in order to ensure that the feedback controllers are given enough control margin—the closed-loop thrust magnitude is constrained to throttle between 20% and 90% T_{max} (allowing for a 10% thrust margin on either end for feedback control). In order to bound the closed-loop thrust magnitude, each of the translation feedback control commands, $F_{x_{fb}}$ and $F_{z_{fb}}$, are restricted to $\pm\frac{10}{\sqrt{2}}\%\, T_{max}$, such that $\|F_{xz_{fb}}\| \leq 10\%\, T_{max}$. It is assumed that the thrust magnitude can throttle at a rate of 50% T_{max} per second, and therefore, the slew rate of the mass-flow rate actuator input $\dot{m}$ is set to $\frac{50\%\, T_{max}}{C}$ per second, where C is the propellant exit-velocity.

Additionally, given the control allocation scheme adopted, there exist fundamental bounds on what the magnitude of the torque command can be: from Equations 5.4 and 5.5, it follows that $|\tau_y| \leq \|F_{xz}\| \, l_{CM}$, where the equality, $|\tau_y| = \|F_{xz}\| \, l_{CM}$, corresponds to a gimbal angle of 90 degrees from the vertical axis in the body frame of the lander. However, the maximum gimbal angle for a realistic lander vehicle is much smaller. The maximum gimbal angle is set to 10 degrees from the vertical axis in the body frame, and in order to enforce that constraint, the torque command is saturated as follows: $|\tau_y| \leq \|F_{xz}\| \, l_{CM} \, \sin(10°)$. Further, a limit of 10 degrees/second is imposed on the gimbal rate.[5]

5.2.3 Results

The closed-loop simulation results demonstrate accurate reference trajectory-tracking and a sub-meter touchdown accuracy. Figures 5.5, 5.6, 5.7, 5.8, 5.9, 5.10, and 5.11 show the states of the nonlinear lander model during powered-descent. The vehicle touches down (reaches zero altitude) with a downrange error of 0.26 meters, and a zero pitch error. The vertical velocity, horizontal velocity, and body pitch rate at touchdown are -0.62 m/s, 0.11 m/s, and 0 degrees/s, respectively, which are well within the bounds for a nominal landing scenario. The total propellant consumed is 861.81 kg, which is only 6.62 kg more than the projected propellant consumption value from guidance. The lever-arm profile is shown in Figure 5.12.

The feedforward and closed-loop thrust commands, and the total thrust magnitude, along with the imposed bounds and the allowed control margins, are shown in Figures 5.14, 5.15, and 5.13. The torque profile is shown in Figure 5.16. The actuator inputs—the mass-flow rate and the gimbal angle, both of which demonstrate satisfaction of the imposed constraints—are shown in Figures 5.17 and 5.18 respectively. The gimbal rate profile is shown in Figure 5.19. Due to the control allocation scheme adopted, spikes are observed in the gimbal rate at the instants when the throttle setting is changed.

[5]Recent prototype lander tests have demonstrated feasibility of the chosen gimbaling constraints [49].

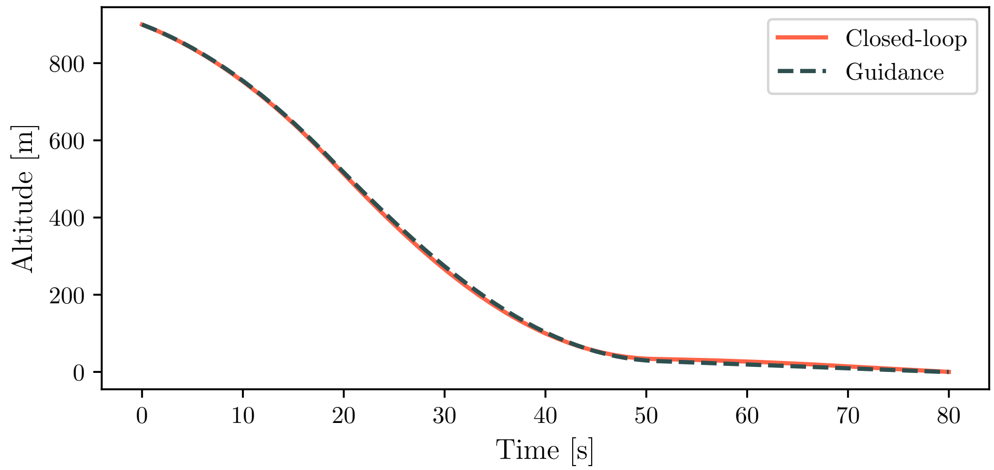

Figure 5.5: Altitude

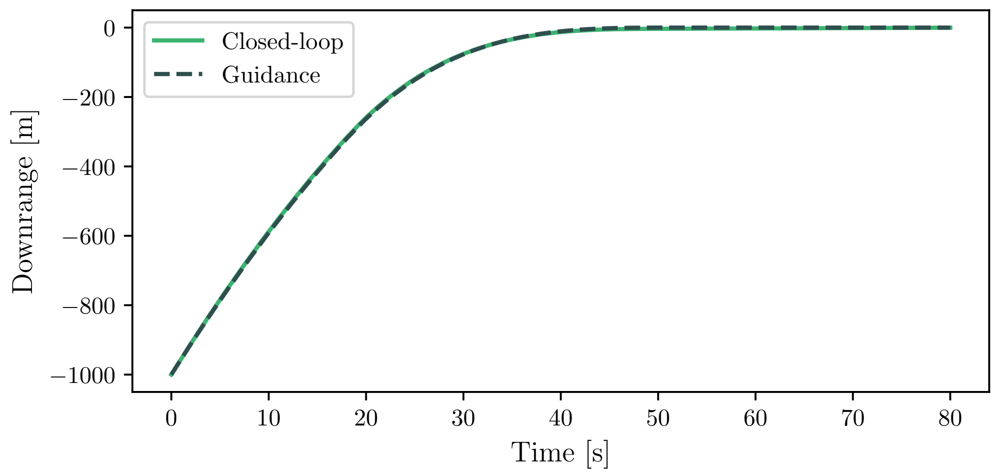

Figure 5.6: Downrange

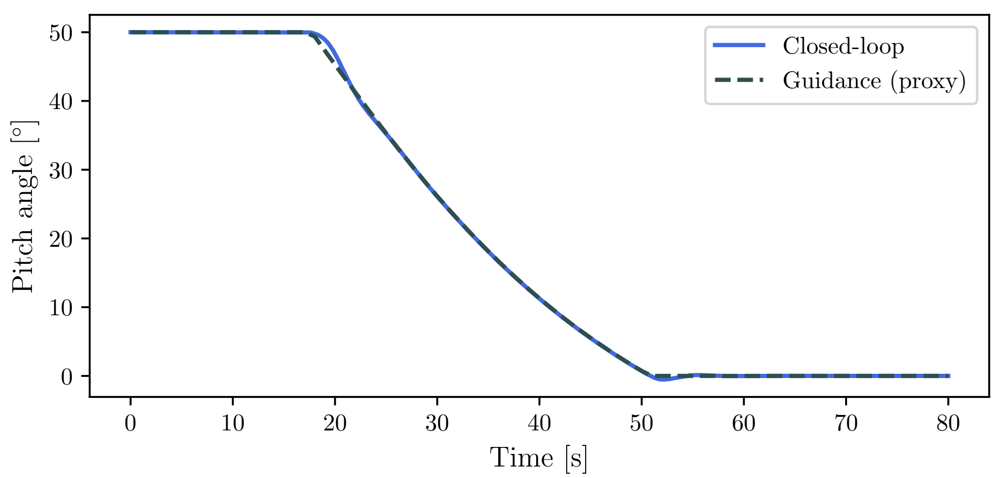

Figure 5.7: Pitch

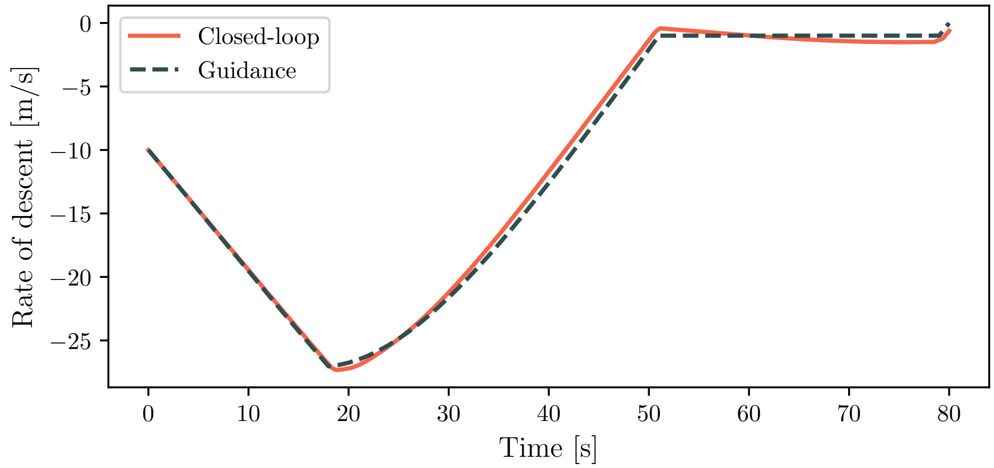

Figure 5.8: Rate-of-descent

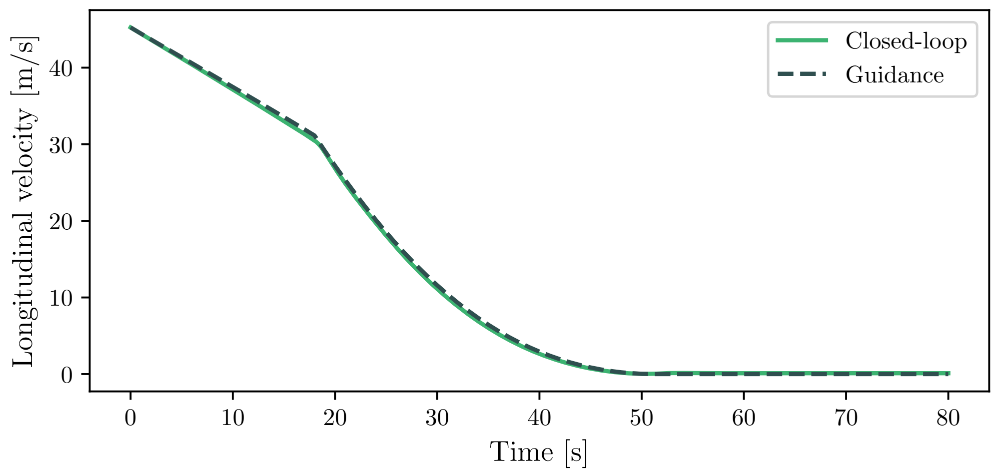

Figure 5.9: Longitudinal velocity

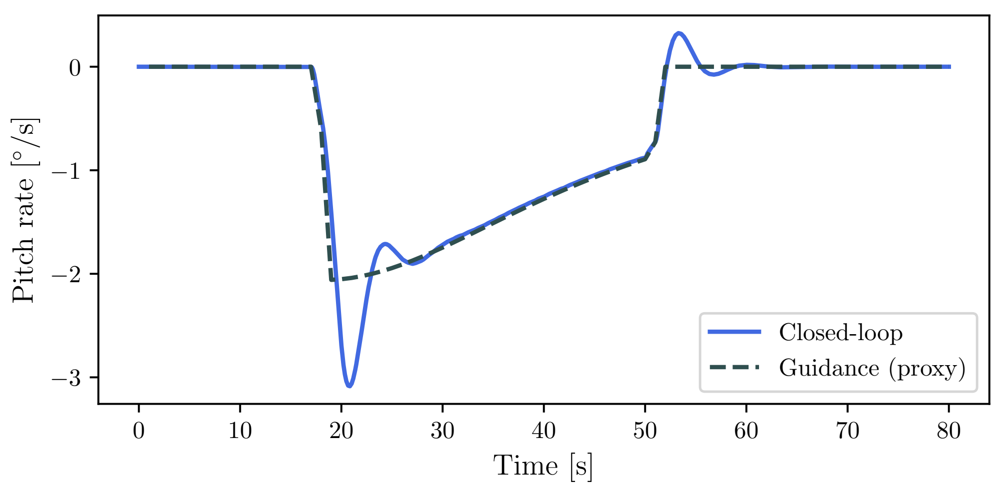

Figure 5.10: Pitch rate

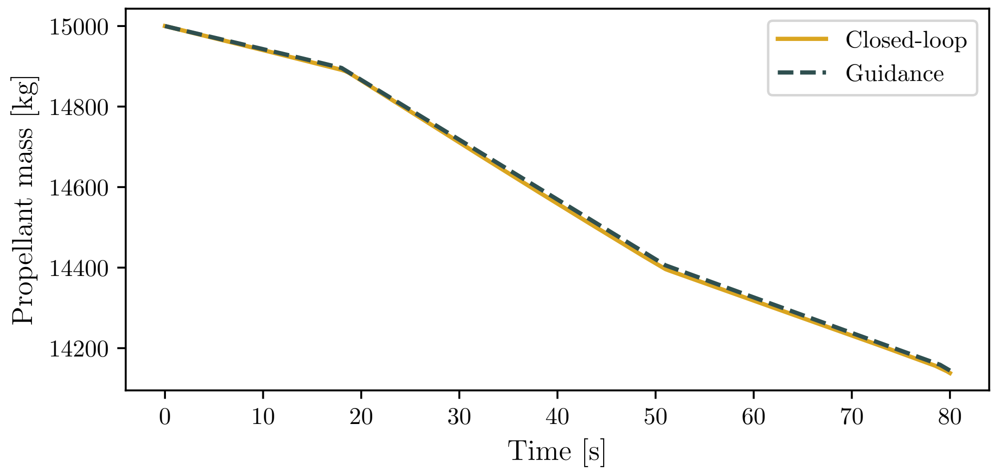

Figure 5.11: Propellant mass

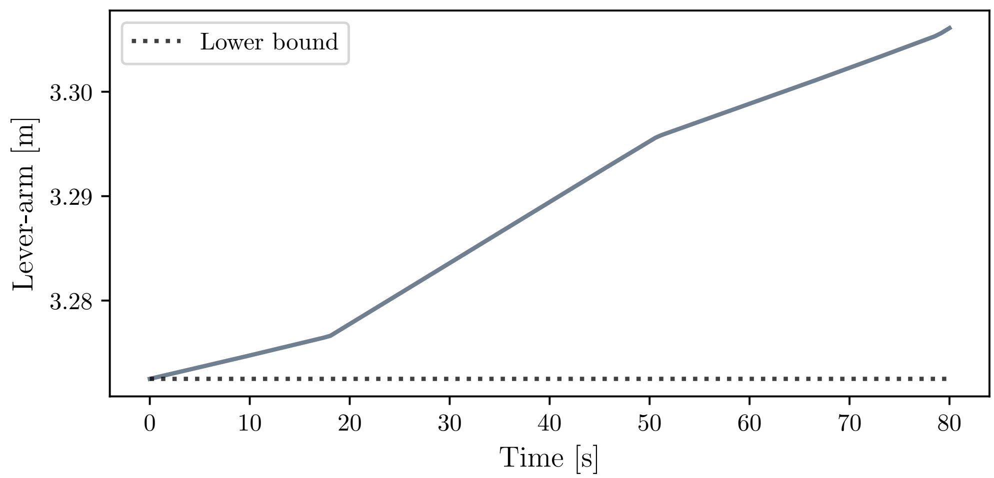

Figure 5.12: Lever-arm

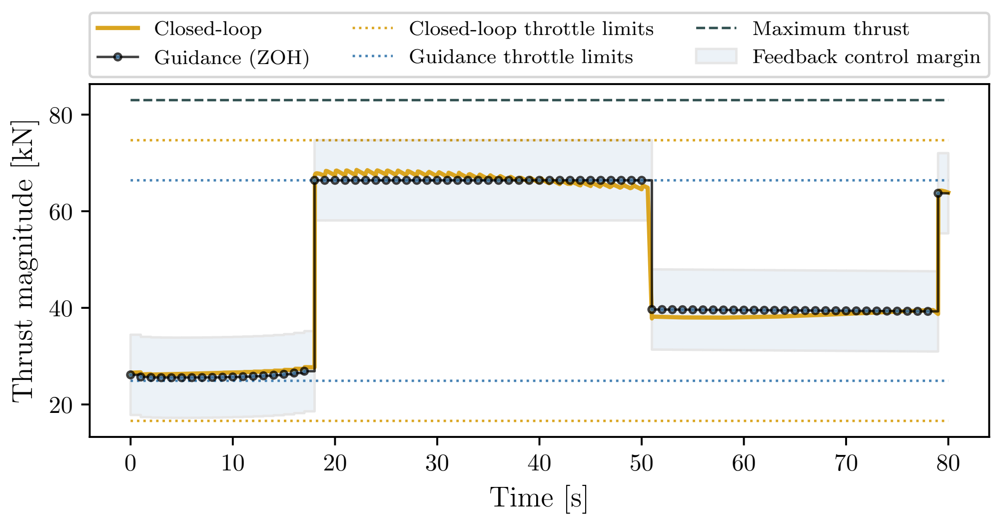

Figure 5.13: Thrust magnitude

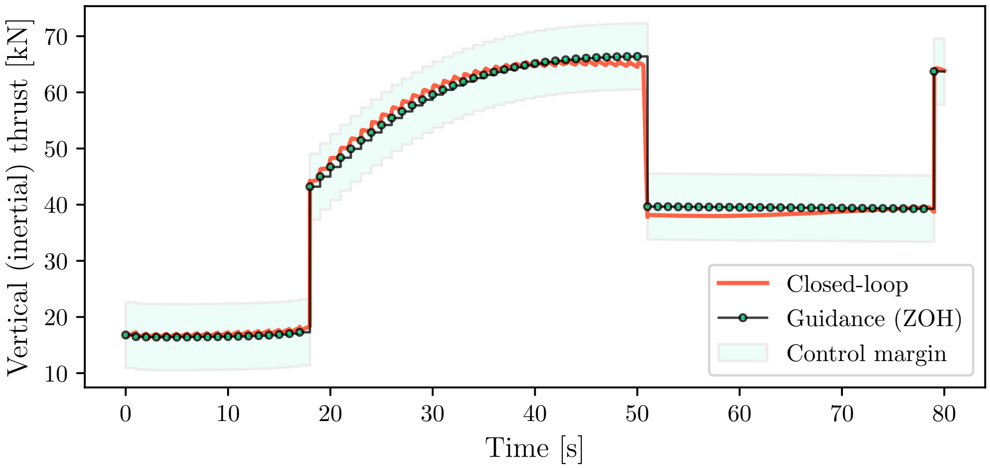

Figure 5.14: Vertical thrust

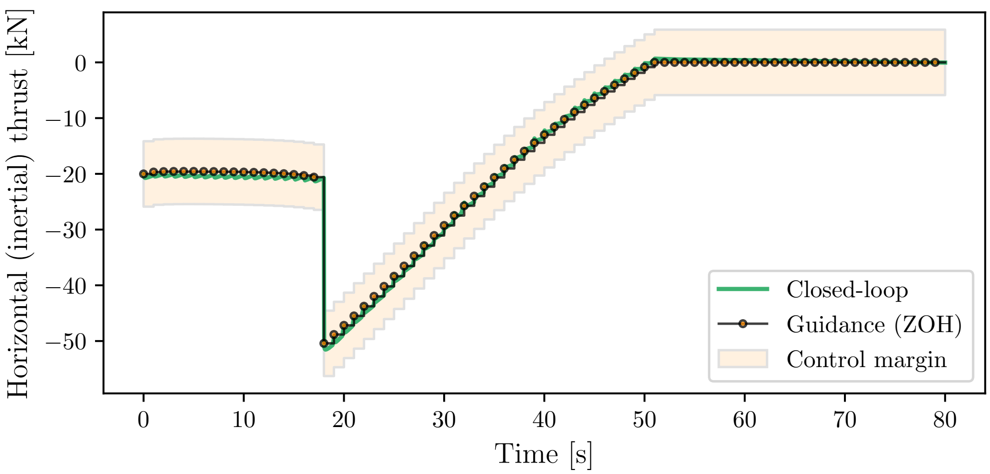

Figure 5.15: Horizontal thrust

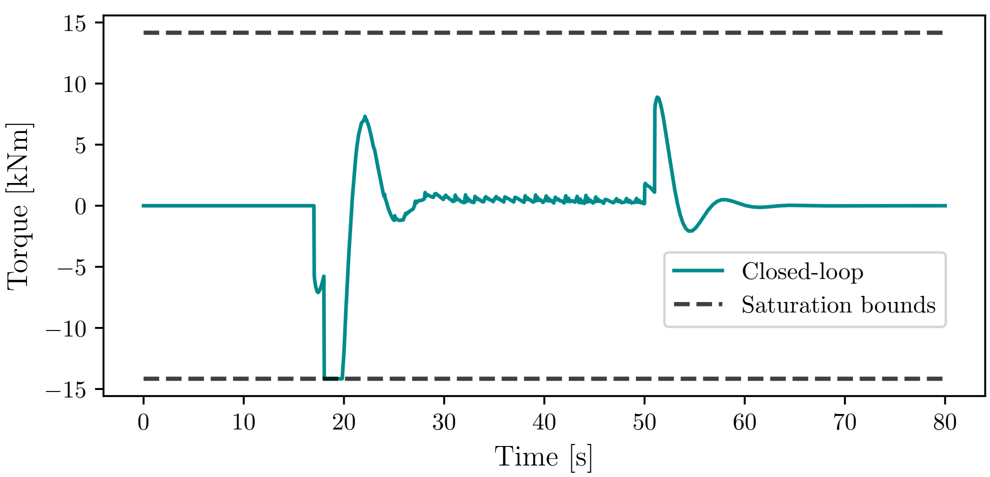

Figure 5.16: Torque

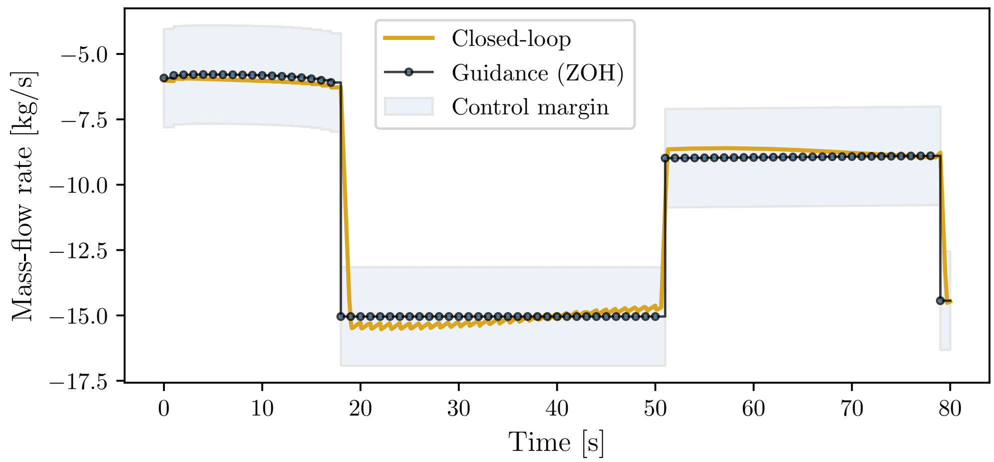

Figure 5.17: Mass-flow rate

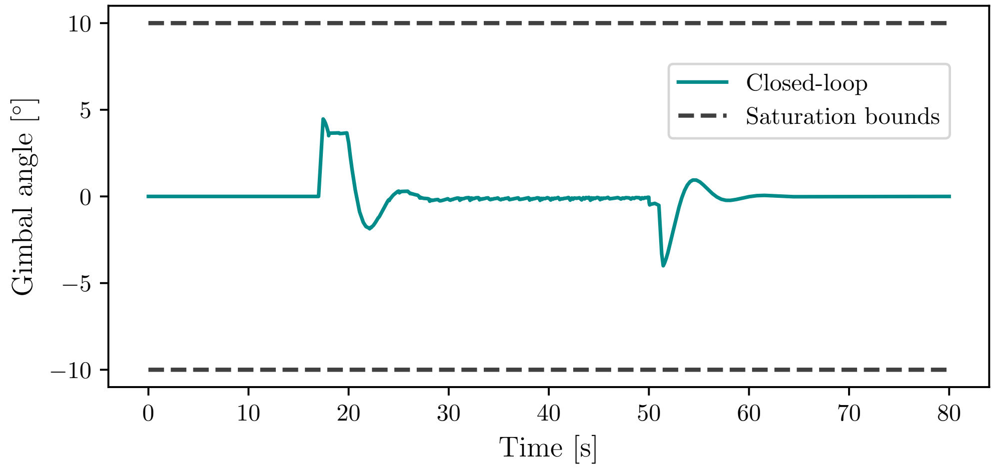

Figure 5.18: Gimbal angle

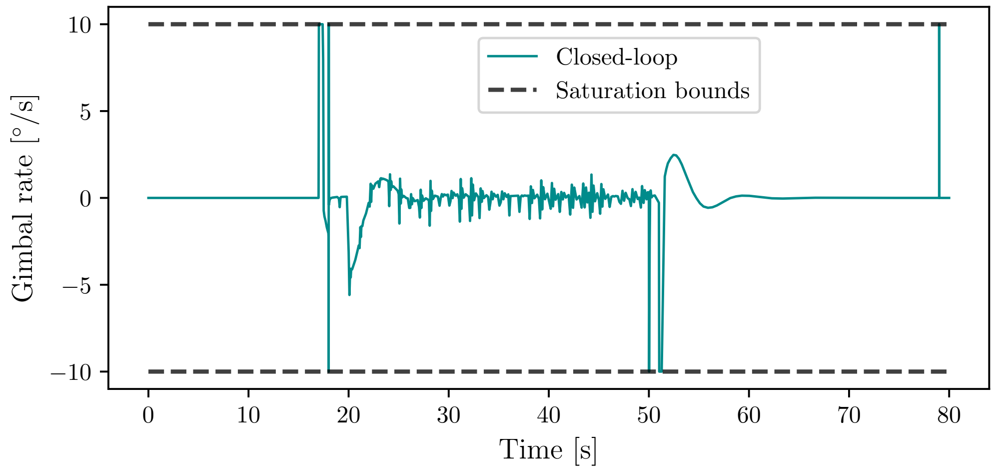

Figure 5.19: Gimbal rate

Chapter 6

Conclusions

6.1 Summary

A framework for analytical multibody dynamics modeling and closed-loop guidance and control simulations for autonomous precision rocket landing is introduced. Development of such an integrated framework with high-fidelity nonlinear dynamical models of variable-mass multibody space vehicles, state-of-the-art propellant-optimal guidance algorithms, and robust feedback control systems, would pave the way for efficient and reliable software-in-the-loop and hardware-in-the-loop powered-descent and precision landing simulations for both human and robotic missions, thus making this line of research very beneficial for future rocket-powered lander missions.

The extended Kane's equations for variable-mass systems are used to analytically model the nonlinear multibody dynamics of a planetary landing vehicle. Propellant-optimal guidance state trajectories are generated by employing lossless convexification in a convex optimization framework. The procedure to design internally stabilizing, multiple-input multiple-output (MIMO) robust feedback control systems for underactuated plants is described. The closed-loop system is shown to possess robustness to bounded uncertainties.

Further, a combined feedforward-feedback control architecture with control allocation and computationally efficient LPV feedback controllers is presented and validated by means of a closed-loop precision landing simulation of the nonlinear variable-mass multibody lander model.

6.2 Contributions

Guidance

- Inclusion of velocity-based vertical-only terminal-descent constraints to the lossless convexification-based 3-DoF powered-descent guidance (PDG) problem, with the relaxed acceleration lower-bound holding with equality (proof of optimality for the problem with the new constraints is yet to be provided).

Dynamics

- Implementation of extended Kane's equations—this method for analytically modeling variable-mass multibody systems is sparse in the literature (and to our knowledge, this work is the first instance of its application in the context of simulating rocket-powered landing vehicles [34]).

Control

- A novel method for robust, internally stabilizing, multivariable feedback control design (offline) for bounded-input bounded-output (BIBO) unstable, underactuated plants with output-only feedback, via Youla parameterization (the FFA method [34]).

- A novel method for the offline design of simple, robust, internally stabilizing, linear parameter-varying (LPV) controllers for BIBO unstable parameter-varying plants with output-only feedback, via Youla parameterization.

- Implementation and validation (in simulation) of a combined feedforward-feedback control architecture for planar, underactuated, constrained, closed-loop precision landing, that includes *partial* 2-DoF (x, z) guidance (translation-only) and complete 3-DoF control (altitude, downrange, and pitch) with only 2 actuator inputs (mass-flow rate and 1 gimbal angle)—which can be readily extended to 3-DoF (x, y, z) guidance with upto 5-DoF control (altitude, crossrange, downrange, pitch, and yaw) with only 3 actuator inputs (mass-flow rate and 2 gimbal angles); roll control can be provided by means of an additional actuator(s).

6.3 Discussion

The PDG algorithm that has been adopted is the state-of-the-art for 3-DoF rocket-landing in terms of computational performance guarantees and accessibility for future planetary landing missions, such as the proposed Artemis lunar landing missions and the Mars Sample Return mission.

The method used for variable-mass multibody dynamics modeling is highly systematic and computationally efficient, and can be extended to include fuel-slosh effects and bending modes for the development of high-fidelity computational multibody lander models.

The proposed novel method for LPV control design via Youla parameterization significantly simplifies the control design and implementation process, since it eliminates the need for the scheduling of gains via lookup tables, which is a common practice in PID controller implementations for LPV plants. The controller order can be increased systematically in order to meet any additional requirements without sacrificing internal stability and robustness. The method does not require full-state feedback unlike strategies such as LQR control, and has been shown to maintain robustness even with the inclusion of estimators/observers in the loop [7].

If implemented, the control strategy (with gimbaled thrust vector control) would result in fewer reaction control system (RCS) thruster firings and in turn, lower mass budgets [30] and increased payload capacities—and also prove beneficial in the event of failure of RCS thrusters (which are typically used on planetary landers for attitude control).

Once the controllers are appropriately tuned offline and deployed on the vehicle, they can be “universal”, in that they can be used (without the need for re-tuning) for a wide range of boundary conditions, given their robustness to system parameter uncertainties, model-mismatch, external disturbances, and sensor noise (robustness to time-delays can be achieved using this method as well), and thus have the potential to enable large divert maneuvers that are commanded mid-flight for scenarios such as hazard avoidance, emergency safe-landings, and ultra-precise landings in general.

6.4 Future Research

Future research directions could include proving optimality of the convex PDG problem with terminal-descent constraints, solving the 6-DoF PDG problem with lossless convexification (and the resulting convergence guarantees), building upon the existing simulation framework by modeling fuel-slosh effects and bending modes, developing a real-time framework including manual guidance capabilities for human subject testing, further exploring Youla parameterization for multivariable control and robust estimation design, and experimentally validating the proposed methods using a physical testbed, to name a few.

Bibliography

A. B. Açıkmeşe and S. Ploen. A powered descent guidance algorithm for Mars pinpoint landing. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, page 6288, 2005.

B. Açıkmeşe, D. Scharf, L. Blackmore, and A. Wolf. Enhancements on the convex programming based powered descent guidance algorithm for Mars landing. In *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, page 6426, 2008.

B. Açıkmeşe, M. Aung, J. Casoliva, S. Mohan, A. Johnson, D. Scharf, D. Masten, J. Scotkin, A. Wolf, and M. W. Regehr. Flight testing of trajectories computed by G-FOLD: Fuel optimal large divert guidance algorithm for planetary landing. In *AAS/AIAA spaceflight mechanics meeting*, 2013.

B. Açıkmeşe and L. Blackmore. Lossless convexification of a class of optimal control problems with non-convex control constraints. *Automatica*, 47(2):341–347, 2011.

B. Açıkmeşe, S. W. Sell, A. M. S. Martin, and J. J. Biesiadecki. Mars Science Laboratory flyaway guidance, navigation, and control system design. *Journal of Spacecraft and Rockets*, 51(4):1227–1236, 2014.

S. M. Alavi, A. K. Sedigh, and B. Labibi. Pre-filter design for tracking error specifications in mimo-qft. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 6466–6471. IEEE, 2005.

F. Assadian, A. K. Beckerman, and J. Velazquez Alcantar. Estimation design using youla parametrization with automotive applications. *Journal of Dynamic Systems, Measurement, and Control*, 140(8), 2018.

F. F. Assadian and K. R. Mallon. *Neoclassical Control: Control Systems and Estimation Design using the Youla Parameterization Technique.* Wiley, 2021.

B. Açıkmeşe and S. R. Ploen. Convex programming approach to powered descent guidance for Mars landing. *Journal of Guidance, Control, and Dynamics*, 30(5):1353–1366, 2007.

B. Açıkmeşe, J. Casoliva, J. Carson, and L. Blackmore. G-FOLD: A real-time implementable fuel optimal large divert guidance algorithm for planetary pinpoint landing. *LPI Contributions*, pages 4193–, 06 2012.

B. Açıkmeşe, J. M. Carson, and L. Blackmore. Lossless convexification of nonconvex control bound and pointing constraints of the soft landing optimal control problem. *IEEE Transactions on Control Systems Technology*, 21(6):2104–2113, 2013.

A. K. Banerjee. Dynamics of a variable-mass, flexible-body system. *Journal of Guidance, Control, and Dynamics*, 23(3):501–508, 2000.

D. P. Bertsekas. Nonlinear programming. *Journal of the Operational Research Society*, 48 (3):334–334, 1997.

N. S. Bhasin. Fuel-optimal spacecraft guidance for landing in planetary pits. Master's thesis, Carnegie Mellon University, Pittsburgh, PA, April 2016.

L. Blackmore. Autonomous precision landing of space rockets. In *in Frontiers of Engineering: Reports on Leading-Edge Engineering from the 2016 Symposium*, volume 46, pages 15–20, 2016.

F. Blanchini, D. Casagrande, S. Miani, and U. Viaro. Stable LPV realization of parametric transfer functions and its application to gain-scheduling control design. *IEEE Transactions on Automatic Control*, 55(10):2271–2281, 2010.

S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

R. P. Brent. *Algorithms for minimization without derivatives*. Courier Corporation, 2013.

K. K. Brown and J. F. Connolly. An Altair overview: Designing a lunar lander for 21st century human space exploration. 2012.

J. M. Carson, B. Açıkmeşe, and L. Blackmore. Lossless convexification of powered-descent guidance with non-convex thrust bound and pointing constraints. In *Proceedings of the 2011 American Control Conference*, pages 2651–2656. IEEE, 2011.

G. Chavers, N. Suzuki, M. Smith, L. Watson-Morgan, S. W. Clarke, W. C. Engelund, L. Aitchison, S. McEniry, L. Means, M. DeKlotz, et al. NASA's human lunar landing strategy. In *70th International Astronautical Congress*, 2019.

G. Chavers, L. Watson-Morgan, M. Smith, N. Suzuki, and T. Polsgrove. NASA's Human Landing System: The strategy for the 2024 mission and future sustainability. In *2020 IEEE Aerospace Conference*, pages 1–9. IEEE, 2020.

A. Chen, P. Brugarolas, E. Hines, A. Johnson, R. Otero, A. Stehura, G. Villar, and D. Way. Mars 2020 entry, descent, and landing system overview. 2016.

S. Diamond and S. Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

A. Domahidi, E. Chu, and S. Boyd. Ecos: An SOCP solver for embedded systems. In *2013 European Control Conference (ECC)*, pages 3071–3076. IEEE, 2013.

J. C. Doyle. Guaranteed margins for LQG regulators. *IEEE Transactions on automatic Control*, 23(4):756–757, 1978.

D. Dueri, B. Açıkmeşe, D. P. Scharf, and M. W. Harris. Customized real-time interior-point methods for onboard powered-descent guidance. *Journal of Guidance, Control, and Dynamics*, 40(2):197–212, 2017.

Z. M. Ge and Y. H. Cheng. Extended Kane's equations for nonholonomic variable mass system. 1982.

R. Hampton, G. Beech, N. Rao, J. Rupert, and Y. Kim. A "Kane's dynamics" model for the active rack isolation system. 2001.

D. G. Hoag. *Apollo navigation, guidance, and control systems: a progress report.* MIT Instrumentation Laboratory, 1969.

A. Jain. *Robot and multibody dynamics: analysis and algorithms.* Springer Science & Business Media, 2010.

A. Johnson, S. Aaron, J. Chang, Y. Cheng, J. Montgomery, S. Mohan, S. Schroeder, B. Tweddle, N. Trawny, and J. Zheng. The lander vision system for Mars 2020 entry descent and landing. 2017.

A. E. Johnson, Y. Cheng, J. Montgomery, N. Trawny, B. E. Tweddle, and J. Zheng. Design and analysis of map relative localization for access to hazardous landing sites on mars. In *AIAA Guidance, Navigation, and Control Conference*, page 0379, 2016.

A. G. Kamath, F. F. Assadian, and S. K. Robinson. Multivariable robust control for the powered-descent of a multibody lunar landing system. In *Artemis Program Special Session, 2020 AAS/AIAA Astrodynamics Specialist Conference*, volume 175, pages 1275–1293. Advances in the Astronautical Sciences, 2021.

A. G. Kamath. Astrobee: A holonomic free-flying space robot, PyDy documentation, 2020. https://pydy.readthedocs.io/en/latest/examples/astrobee.html.

A. G. Kamath. Modeling of a variable-mass nonholonomic gyrostatic rocket car using extended Kane's equations, PyDy documentation, 2021. https://pydy.readthedocs.io/en/latest/examples/rocket-car.html.

T. R. Kane and D. A. Levinson. *Dynamics, theory and applications.* McGraw Hill, 1985.

T. R. Kane, P. W. Likins, and D. A. Levinson. Spacecraft dynamics. *New York, McGraw-Hill Book Co, 1983, 445 p.*, 1983.

A. R. Klumpp. Apollo lunar descent guidance. *Automatica*, 10(2):133–146, 1974.

L. Kos, T. Polsgrove, R. Sostaric, E. Braden, J. Sullivan, and T. Le. Altair descent and ascent reference trajectory design and initial dispersion analyses. In *AIAA Guidance, Navigation, and Control Conference*, page 7720, 2010.

V. Kučera. A method to teach the parameterization of all stabilizing controllers. *IFAC Proceedings Volumes*, 44(1):6355–6360, 2011.

E. Lavretsky and K. A. Wise. Robust adaptive control. In *Robust and adaptive control*, pages 317–353. Springer, 2013.

U. Lee and M. Mesbahi. Constrained autonomous precision landing via dual quaternions and model predictive control. *Journal of Guidance, Control, and Dynamics*, 40(2):292–308, 2017.

C. Liu, W.-H. Chen, and J. Andrews. Tracking control of small-scale helicopters using explicit nonlinear mpc augmented with disturbance observers. *Control Engineering Practice*, 20 (3):258–268, 2012.

X. Liu. Autonomous trajectory planning by convex optimization. 2013.

X. Liu and P. Lu. Robust trajectory optimization for highly constrained rendezvous and proximity operations. In *AIAA Guidance, Navigation, and Control (GNC) Conference*, page 4720, 2013.

X. Liu, Z. Shen, and P. Lu. Entry trajectory optimization by second-order cone programming. *Journal of Guidance, Control, and Dynamics*, 39(2):227–241, 2016.

Z. Liu, F. Assadian, and K. Mallon. Vehicle yaw rate and sideslip estimations: A comparative analysis of siso and mimo youla controller output observer, linear and nonlinear kalman filters, and kinematic computation. *Journal of Mechanical and Aerospace Engineering*, 1 (1), 2019.

D. Malyuta, T. P. Reynolds, M. Szmuk, T. Lew, R. Bonalli, M. Pavone, and B. Açıkmeşe. Convex optimization-based trajectory generation. *IEEE Control Systems Magazine (work in progress)*, 2021.

A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. K. Moore, S. Singh, T. Rathnayake, S. Vig, B. E. Granger, R. P. Muller, F. Bonazzi, H. Gupta, S. Vats, F. Johansson, F. Pedregosa, M. J. Curry, A. R. Terrel, v. Roučka, A. Saboo, I. Fernando, S. Kulal, R. Cimrman, and A. Scopatz. Sympy: symbolic computing in python. *PeerJ Computer Science*, 3:e103, Jan. 2017. ISSN 2376-5992. doi: 10.7717/peerj-cs.103. URL https://doi.org/10.7717/peerj-cs.103.

MOSEK ApS. MOSEK optimizer API for Python 9.0.105, 2019. URL https://docs.mosek.com/9.0/pythonapi/index.html.

A. Nelessen, C. Sackier, I. Clark, P. Brugarolas, G. Villar, A. Chen, A. Stehura, R. Otero, E. Stilley, D. Way, et al. Mars 2020 entry, descent, and landing system overview. In *2019 IEEE Aerospace Conference*, pages 1–20. IEEE, 2019.

D. Noll. A generalization of the linear quadratic gaussian loop transfer recovery procedure (LQG/LTR).

J. Orr and Y. Shtessel. Robust lunar spacecraft autopilot design using high order sliding mode control. In *AIAA Guidance, Navigation, and Control Conference*, page 6182, 2009.

C. A. Pascucci, S. Bennani, and A. Bemporad. Model predictive control for powered descent guidance and control. In *2015 European Control Conference (ECC)*, pages 1388–1393. IEEE, 2015.

D. Petersen, J. Charvat, J. Somers, J. Pattarini, M. Stenger, M. Van Baalen, and S. Lee. Apollo to artemis: Mining 50-year old records to inform future human lunar landing systems. 2020.

C. Philippe, A. Annaswamy, G. Balas, J. Bals, S. Garg, A. Knoll, K. Krishnakumar, M. Maroni, R. Osterhuber, and Y. C. Yeh. Apollo as the catalyst for control technology.

R. M. Pinson and P. Lu. Trajectory design employing convex optimization for landing on irregularly shaped asteroids. *Journal of Guidance, Control, and Dynamics*, 41(6):1243–1256, 2018.

R. Prakash, P. D. Burkhart, A. Chen, K. A. Comeaux, C. S. Guernsey, D. M. Kipp, L. V. Lorenzoni, G. F. Mendeck, R. W. Powell, T. P. Rivellini, et al. Mars science laboratory entry, descent, and landing system overview. In *2008 IEEE Aerospace Conference*, pages 1–18. IEEE, 2008.

W. Quaide and V. Oberbeck. Geology of the apollo landing sites. *Earth-Science Reviews*, 5 (4):255–278, 1969.

G. Rodriguez, A. Jain, and K. Kreutz-Delgado. A spatial operator algebra for manipulator modeling and control. *The International Journal of Robotics Research*, 10(4):371–381, 1991.

A. M. San Martin. The challenges of landing Curiosity on Mars. 2017.

A. M. San Martin, S. W. Lee, and E. C. Wong. The development of the msl guidance, navigation, and control system for entry, descent, and landing. 2013.

D. P. Scharf, B. Açıkmeşe, D. Dueri, J. Benito, and J. Casoliva. Implementation and experimental demonstration of onboard powered-descent guidance. *Journal of Guidance, Control, and Dynamics*, 40(2):213–229, 2017.

M. Smith, D. Craig, N. Herrmann, E. Mahoney, J. Krezel, N. McIntyre, and K. Goodliff. The Artemis program: An overview of NASA's activities to return humans to the Moon. In *2020 IEEE Aerospace Conference*, pages 1–10. IEEE, 2020.

L. Spannagl, E. Hampp, A. Carron, J. Sieber, C. A. Pascucci, A. U. Zgraggen, A. Domahidi, and M. N. Zeilinger. Design, optimal guidance and control of a low-cost re-usable electric model rocket. *arXiv preprint arXiv:2103.04709*, 2021.

A. Steltzner, D. Kipp, A. Chen, D. Burkhart, C. Guernsey, G. Mendeck, R. Mitcheltree, R. Powell, T. Rivellini, M. San Martin, et al. Mars science laboratory entry, descent, and landing system. In *2006 IEEE Aerospace Conference*, pages 15–pp. IEEE, 2006.

M. Szmuk, U. Eren, and B. Açıkmeşe. Successive convexification for Mars 6-DoF powered descent landing guidance. In *AIAA Guidance, Navigation, and Control Conference*, page 1500, 2017.