New Method and Analysis of Proximity Trajectory-Only Learned Dynamics for Small Body Gravity Fields

Thesis by

Daniel Neamati

In Partial Fulfillment of the Requirements for the
Degree of
Bachelor of Science in Mechanical Engineering

Caltech

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2021
Defended 27 May 2021

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my family that has supported me immensely. I especially want to thank them for their support through the last four virtual terms. Next, I would like to thank Prof. Chung for more than just his role as my thesis advisor. I am grateful to Prof. Chung for making this senior thesis opportunity available, guiding me through developing my proposal, and evaluating my research approach along the way. Outside of the senior thesis domain, I wanted to thank Prof. Chung for helping me navigate the graduate school application process and supporting my application. I would also like to thank Yashwanth, my graduate student mentor. I truly could not have done this thesis without his mentorship. Whether recommending papers for me to read, reviewing my drafts, or helping me talk through roadblocks, Yashwanth has definitively made me a better researcher. Additionally, I would like to thank Nikhil Ranganathan for his insightful suggestions on tackling this research problem.

Moreover, the endeavor of conducting a senior thesis has been so much more than lines of Python code. I would like to thank Dr. Voong at the Hixon Writing Center for building my skills as a scientific storyteller and technical communicator. Dr. Voong's fantastic feedback to improve my presentations and develop a much more cohesive thesis empowered me with a new framework to assess scientific writing. In a similar vein, I am thankful that Regina (who also conducted a senior thesis this year) and I became a peer review team. I appreciate the comradery in successfully navigating the senior thesis process together.

There are also many people removed from the immediate research that helped me make this thesis a reality. I would like to thank Holly, Martha, and Sonya for all the help scheduling meetings and presentations in a virtual world with different timezones. I would like to thank my past research mentors, including Prof. Zachary Manchester, Brian Jackson, Kevin Tracy, Pavlina Karafillis, Tom Green, John Gallon, Chris Sercel, Dr. Eric Roberts, Dr. Manuel Bedrossian, Dr. Scott Perl, Dr. Chris Lindensmith, Dr. Erik Fischer, Dr. German Martinez, Prof. Nilton Rennó, Dr. Scott Hall, Dr. Ethan Dale, Eric Viges, and Dean Gallimore.

I am additionally grateful to my mentors who supported me throughout my Caltech journey and have become new role models in my life. I especially want to thank Dr. Jenn Weaver at the Center for Teaching, Learning, and Outreach (CTLO),

Prof. Melany Hunt (my option advisor), Prof. Bethany Ehlmann (my minor option advisor), Dr. Michael Mello, Prof. Chiara Daraio, Prof. Domniki Asimaki, Prof. Cindy Weinstein, Dr. Lindsey Malcom-Piquex, and the Caltech Deans.

Lastly, I would like to more directly thank the following people who make up my thesis committee and advising team. Thank you so much!

Prof. Soon-Jo Chung

Bren Professor of Aerospace and Control and Dynamical Systems,
NASA Jet Propulsion Laboratory Research Scientist,
Thesis Advisor

Yashwanth Kumar Nakka

Ph.D. Candidate in GALCIT,
Graduate Student Mentor

Prof. Austin Minnich

Professor of Mechanical Engineering and Applied Physics,
Chair of Undergraduate Thesis Committee

Prof. Melany Hunt

Dotty and Dick Hayman Professor of Mechanical Engineering,
Undergraduate Mechanical Engineering Option Advisor

Prof. Bethany Ehlmann

Professor of Planetary Science,
Associate Director of the Keck Institute for Space Studies,
Undergraduate Planetary Science Minor Advisor

ABSTRACT

Recent missions to small bodies in the past decade (e.g., *Rosetta*, *Hayabusa 2*, and *OSIRIS-REx*) have reshaped our understanding of small bodies and inspired new, more-capable future missions. Despite the high demand for more missions, large uncertainties in small body properties make missions challenging. Recent work in stochastic optimal control can ensure safety in the face of uncertainty in state, constraints, and dynamics. These stochastic optimal controllers require a model of the underlying dynamics, which is difficult for proximity maneuvers and landing around small bodies. Shape models and finite element-like models are the state-of-the-art for high-fidelity gravity models, but they are computationally expensive and do not readily incorporate onboard data. No gravity model yet exists that can use short-horizon position and acceleration data from recent trajectories onboard in safety-critical autonomous proximity maneuvers and landing. Therefore, we propose a new trajectory-only learning-based method to develop a gravity model. We consider three learning frameworks: Gaussian Process Models, Neural Networks, and Physics-Informed Neural Networks. For each framework, we assess the benefits, computational costs, and limitations of the framework. We found that the Gaussian Process Model generally outperforms the other frameworks in cases of moderate uncertainty. As the uncertainty declines or the data is sufficiently filtered, Neural Networks with spectral normalization provide more accurate gravity models and are computationally cheaper to evaluate. Lastly, we reflect on the methods in this thesis and recommend possible problem reformulations for future research.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Number *Page*

LIST OF TABLES

NOMENCLATURE

Brillouin sphere. The minimum sphere containing the entire small body within the reference coordinate frame.

Differentiation. The process of separating dense material (namely iron) towards the core of a body and lighter material (namely silicates and ices) towards the surface.

Finite element methods. A technique in mechanical engineering used most commonly in fluids and solids research. A paralleled technique in gravity research separates a small body into many smaller elements with point mass like properties.

Gaussian Process Model. A class of machine learning strategies that builds upon stochastic processes to form a distribution over functions. In this thesis, we narrow the Gaussian Process Model to exact inference with a radial basis function kernel.

Gravity model. An approximation of the gravity field within the relevant region around a solar system body.

Hill sphere. The sphere whereby the gravitation influence of the secondary (such as an asteroid) begins to overtake the gravitation influence of the primary (such as the sun).

Learning. A short-hand for “machine learning," which refers to algorithmic strategies to make inference from training data.

Lipschitz continuity. A condition on the smoothness of a function to avoid abrupt shifts. Lipschitz continuity is especially valuable in differential equations. Within robotics and learned dynamics research, Lipschitz continuity is important to ensure safe motion for an autonomous vehicle.

Neural Network. A versatile, popular, and broad class of machine learning strategies that feature layered nodes of variable size. In this thesis, we narrow the Neural Network to a feedforward Neural Network with 3 to 5 layers and rectified linear unit activation functions.

Physics-Informed Neural Network. A type of Neural Network with a modified cost function to ensure that the network predictions comply with a given physical law. For smoothness, we use hyperbolic tangent functions over rectified linear unit activation functions.

Planetary defense. The monitoring and research efforts to understand possible small body impacts with Earth.

Polyhedron shape model. A mesh-like description of an object as a set of vertices, edges, and faces. This mesh-like shape model can fully describe a gravity model if the object is of constant density.

Proximity maneuvers. Spacecraft motion planning and correction near the surface of a solar system body.

Resurfacing. The large-scale process whereby a solar system body's surface melts, subducts under other crustal material, or otherwise altered (such as via volcanism or heavy bombardment).

Small bodies. A large class of solar system objects (including asteroids, comets, and some moons) where their self-gravity is not large enough to produce a spherical shape.

Spectral normalization. A technique to constrain the weights in a Neural Network to ensure a Lipschitz continuous output.

Spherical harmonics expansion. A series expansion that is convenient to solving Laplace's equation (a partial differential equation) in spherical coordinates.

Trajectory-Only. Building training data from a single trajectory rather than a randomized collection of points.

MATHEMATICAL NOMENCLATURE

a	=	Semi-major axis (m)
$\vec{a}$	=	Acceleration vector (m s^{-2})
$\tilde{a}$	=	Normalized acceleration (see Appendix C)
$a_{r,\theta,\phi,x,y,z}$	=	Acceleration in the r, θ, ϕ, x, y, or z directions, respectively (m s^{-2})
$A \cos \alpha$	=	Projected area (m^2)
c	=	Speed of Light ($2.997\,92 \times 10^5$ km s^{-1})
C	=	Integrating constant
C_n^m	=	(n, m)-th Cosine Tesseral Coefficient
e	=	Eccentricity
$\mathbb{E}$	=	Expectation Operator
$\vec{F}$	=	Force (N)
$\mathcal{F}$	=	Safe set of states (set over m and m s^{-1})
$\vec{g}$	=	Gravitational acceleration field (m s^{-2})
g	=	True dynamics (m s^{-2})
$\hat{g}$	=	Learned dynamics (m s^{-2})
γ	=	Scaling parameter between the n-th harmonic an Hill sphere
G	=	Gravitational Constant (6.6743×10^{-11} m^3 kg^{-1} s^{-2})
G_{SC}	=	Solar Constant (1.36 kW m^{-2} at Earth)
J, J^*	=	The cost function and optimized cost value, respectively
$\tilde{J}_n$	=	Dimensionless n-th Zonal Coefficient
K	=	Shared acceleration coefficient (defined in Appendix A)
K_L	=	Lipschitz constant
m	=	Mass of the Secondary (kg)
m_s	=	Satellite Mass (kg)
M	=	Mass of the Primary (kg)
μ	=	Central Body's Gravitational Parameter (Gm or GM)
n	=	A natural number denoting the term in the spherical harmonics expansion
$P_n^0 = P_n$	=	The n-th degree Legendre P Polynomial
P_n^m	=	The n-th degree m-order Associated Legendre P Polynomial for $m \neq 0$
Pr	=	Probability of a specified event
ϕ	=	Azimuthal Angle (rad)
r	=	Radius, $\hat{r}$ for unit vector, $\vec{r}$ for radius vector (m)
$\tilde{r}$	=	Normalized radius vector (see Appendix C)

r_H	=	Hill Radius (AU or m)
ρ	=	Density ($\mathrm{g\,cm^{-3}}$)
R	=	Normalizing average radius of the body (m)
S_n^m	=	(n, m)-th Cosine Tesseral Coefficient
$t_{0,f}$	=	Starting and final times, respectively (s)
$\vec{T}$	=	Control Thrust (N)
τ	=	Orbital period (s)
θ	=	Polar Angle from the equator (rad)
Θ	=	Neural Network weights
u	=	Specific Energy ($\mathrm{m^2\,s^{-2}}$)
$u_{n,H}$	=	Specific Energy of the n-th harmonic or Hill sphere, respectively ($\mathrm{m^2\,s^{-2}}$)
$\bar{u}$	=	Mean control trajectory (N)
U	=	Gravitational Potential (J)
$\mathcal{U}$	=	Set of mechanically feasible controls (set over N)
V	=	Volume ($\mathrm{m^3}$)
$v_{x,y,z}$	=	Velocity in the x, y, or z Cartesian directions, respectively ($\mathrm{m\,s^{-1}}$)
$\tilde{v}$	=	Normalized velocity (see Appendix C)
x, y, z	=	Position in the Cartesian directions (m)

Chapter 1

MOTIVATION AND BACKGROUND

1.1 Small Body Exploration for Science, Economics, and Humanity

Small bodies, such as asteroids and comets, hold resources, opportunities, and challenges that larger solar system objects do not share. Without a large size, these objects do not have the self-gravity to maintain a spherical shape. Combining observation from Earth and spacecraft, NASA has developed a database of known small bodies (Davis et al., 2021). Examples of these small bodies are included in Figure 1.1. Knowing that these small bodies exist is not enough. For scientific, economic, and humanitarian reasons, we seek to visit, land, and sample small bodies.

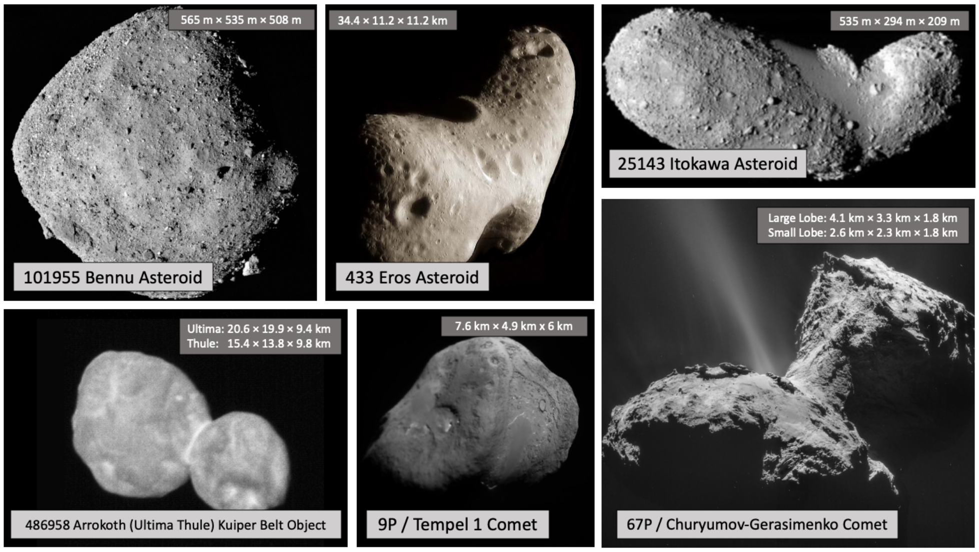

565 m × 535 m × 508 m
101955 Bennu Asteroid
34.4 × 11.2 × 11.2 km
433 Eros Asteroid
535 m × 294 m × 209 m
25143 Itokawa Asteroid
Ultima: 20.6 × 19.9 × 9.4 km
Thule: 15.4 × 13.8 × 9.8 km
486958 Arrokoth (Ultima Thule) Kuiper Belt Object
7.6 km × 4.9 km x 6 km
9P / Tempel 1 Comet
Large Lobe: 4.1 km × 3.3 km × 1.8 km
Small Lobe: 2.6 km × 2.3 km × 1.8 km
67P / Churyumov-Gerasimenko Comet

Figure 1.1: Examples of solar system small bodies illustrating the breadth of geometries including near-spherical (101955 Bennu and 9P / Tempel 1), elongated (433 Eros and 25143 Itokawa), and binary (486958 Arrokoth and 67P/Churyumov–Gerasimenko). From left to right, the *OSIRIS-REx* mission visited the carbonaceous Near-Earth asteroid 101955 Bennu, the *NEAR Shoemaker* mission visited the stony asteroid 433 Eros, the *Hayabusa* mission visited the carbonaceous Near-Earth asteroid 25143 Itokawa, the *New Horizons* mission visited the trans-Neptunian object 486958 Arrokoth (also known as Ultima Thule), the *Deep Impact* mission visited the comet 9P/Tempel, and the *Rosetta* mission visited the comet 67P/Churyumov–Gerasimenko. Images from NASA Goddard/UArizona, NASA JPL/JHUAPL, JAXA, NASA JPL/University of Maryland, NASA/JHUAPL/SRI, and ESA. Data from NASA JPL and public domain.

Scientific Motivations for Small Body Exploration

Planetary science and astrophysics research actively seeks to understand the early solar system, the formation of planetary systems, and the transport of material around the solar system. However, Earth's resurfacing mechanisms have destroyed nearly all of the first billion years of Earth's sedimentary geologic record. On Earth, the oldest fragments of early Earth date back to 4.4 billion years ago but are only tiny Zicron grains (Cavosie, Valley, and Wilde, 2007). The oldest known sedimentary deposits date to 3.7 billion years ago. While these sedimentary deposits are much larger than the tiny Zircon grains, heavy metamorphism has tarnished the interpretability of these samples (Myers, 2001). Other terrestrial bodies suffered similar fates. Volcanic processes constantly overturn Venus's crust yielding an extraordinarily young surface (Basilevsky and Head, 1998). Mercury might seem more promising since its surface is less active, but the immense heat from the accretion process means that it too underwent resurfacing. Our best data indicates that the oldest portions of Mercury's surface are 4 billion years old (Marchi et al., 2013). The gas and ice giants (e.g., Jupiter, Saturn, Uranus, and Neptune) don't have a surface at all. The solar system moons are also often young. Triton and Europa have notably young surfaces and the mechanism to renew their surface is not yet understood (Schenk and Zahnle, 2007; Kattenhorn and Prockter, 2014). While interesting in their own respects, these large planet-scale and moon-scale bodies, unfortunately, do not let us probe questions about the early solar system, its formation, or its transport mechanisms.

On the other hand, small bodies have not experienced this level of resurfacing (Patterson, 1956; Amelin et al., 2002). Therefore, small bodies are unique time capsules into the early solar system. To form a holistic picture of the solar system, we need a diverse set of small body samples across asteroid classes, Kuiper belt objects, and long-period comets. With such an inventory, we can probe pressing scientific topics including the origin of prebiotic material, the transport of interplanetary or interstellar material, the mineralogy of the early solar system, and the mechanisms for Earth's planetary growth.

Economic Motivations for Small Body Exploration

Small bodies are economically valuable for extracting rare materials and enabling long-duration outer solar system missions. Whether platinum for catalytic converters, tellurium for cell phones, precious metals for jewelry, or lanthanides for precision spectroscopy targets, many industries eagerly seek access to rare metals that one can readily access on small bodies (Lotzof, 2020; Kargel, 1994). Most dense metals migrate inwards during planet formation (Caro, 2011; Weiss and Elkins-Tanton, 2013), rendering these metals largely inaccessible. With a smaller size and without large-scale material differentiation (i.e., the separation of dense material into the body's core), rare metals are easily accessible on small bodies.

Moreover, the asteroid belt provides a convenient, resource-rich pit stop for outer planet missions, including access to water and hydrocarbons. With a smaller gravity well, launching off a small body takes much less fuel than larger bodies. Moreover, the water and hydrocarbons available on asteroids provide convenient fuel sources for pit stops between the inner terrestrial planets and the outer giant planets (Rapp, 2018). These asteroid-based refueling stations may be both lucrative and enabling technologies for outer solar system exploration.

Humanitarian Motivations for Small Body Exploration

Small bodies can also be destructive. NASA and international partners (i.e., the International Academy of Astronautics) have allocated notable funds to planetary defense: the field that monitors and investigates possible small body impacts with Earth. Small body impacts (most commonly Near-Earth asteroids) can incur substantial costs in infrastructure damage and human casualties (Tricia Talbert, 2018), although mass extinction level impacts are extraordinarily rare and to not seem to pose an immediate danger to Earth. At current, small body outgassing and interplanetary transport is an active area of research that influences our ability to predict long-term asteroid trajectories (Bottke et al., 2006). Measurements from Earth are difficult and of low resolution compared to the high-fidelity data from small body missions, such as *Rosetta* (2004-2016), *Hayabusa* (2003-2005 and 2014-2020), and *OSIRIS-REx* (2016-2023). Therefore, for our sake as a population, we seek to understand small bodies in-depth: their scientific properties, their short and long-term processes, their astrodynamics, and their dangers.

1.2 Meeting the Challenges: Proximity Maneuvers and Landing

Each motivating field (science, economics, and humanitarian) requires spacecraft missions that orbit near the small body surface or lander missions to sample the small body. We need to unpack the technologies necessary to fulfill the motivating goals and determine how this thesis can address the needs of spacecraft missions to small bodies. Figure 1.2 serves as a map for placing this thesis in the broader field of proximity maneuvers and landing on small bodies.

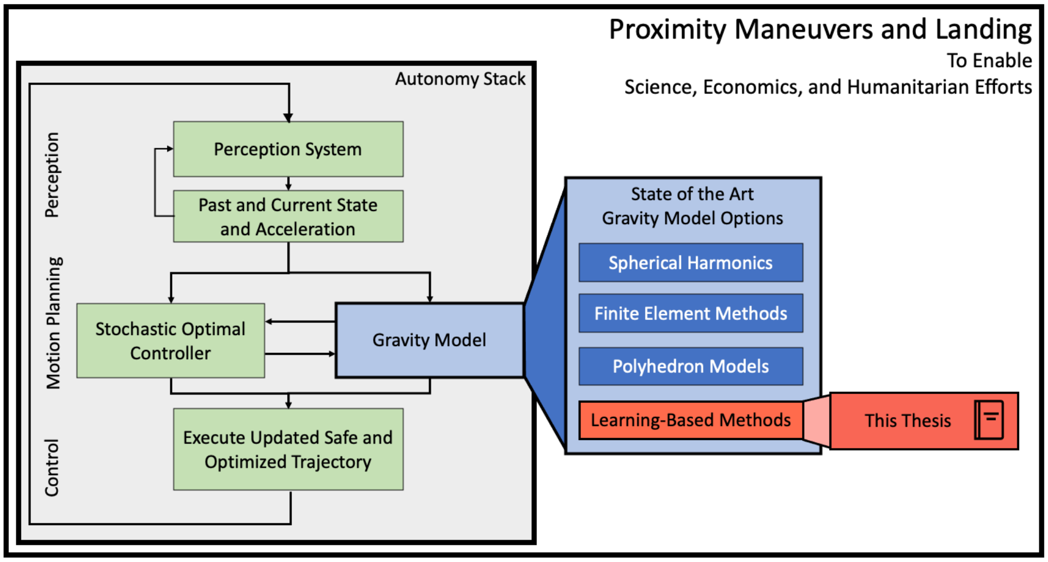

Proximity Maneuvers and Landing
To Enable
Science, Economics, and Humanitarian Efforts
Autonomy Stack
Perception
Motion Planning
Control
Perception System
Past and Current State and Acceleration
Stochastic Optimal Controller
Gravity Model
Execute Updated Safe and Optimized Trajectory
State of the Art Gravity Model Options
Spherical Harmonics
Finite Element Methods
Polyhedron Models
Learning-Based Methods
This Thesis

Figure 1.2: Concept schematic of how this thesis fits into the broader context of autonomous proximity maneuvers and landing. In the background section, we will start from the outer box with an overview of proximity maneuvers. Then, we will discuss the autonomy stack and narrow in on the Information-Cost Stochastic Nonlinear Optimal Control, which works well with learning-based methods. Lastly, we will expand on the difficulty of gravity models and demonstrate the contribution of this thesis.

Since these small bodies are far from Earth, the light time delay during critical near-surface maneuvers prohibits teleoperated operations. Increasingly, spacecraft must operate autonomously and safely. The failure of the *Philae* Lander in 2014 (as part of the *Rosetta* mission to the comet 67P/Churyumov-Gerasimenko) highlights the need for novel active, robust, and safe autonomous landing techniques for small bodies (ESA, 2015).

Spacecraft and robots alike face the difficult control challenge of optimally maneuvering from an initial position with an initial velocity to a target position at a target velocity. We can split this challenge into three broad categories (1) Perception or Navigation, (2) Motion Planning or Guidance, and (3) Control. The perception or navigation system (hereafter called the perception system) is responsible for determining the spacecraft's position and velocity (grouped under a state vector in $\mathbb{R}^6$). The motion planning or guidance system (hereafter called the motion planning system) is responsible for using the estimated system state to determine the desired trajectory. The control system controls onboard subsystems, such as thrusters and reaction wheels, and is responsible for executing the desired path.

Safety requires that the spacecraft is able to prepare a crash-free trajectory. Optimality requires that we find the best crash-free trajectory. In physical systems, uncertainty (including uncertainty in spacecraft state and uncertainty about the small body) makes the safe and optimal autonomy challenge especially difficult. Recent work has formulated spacecraft and robotic motion planning with chance constraints in stochastic optimal control to handle uncertainty within the autonomy stack (Oguri and Jay W McMahon, 2020; Nakka and Chung, 2019; Nakka, Liu, et al., 2020; Nakka, Hönig, et al., 2021).

The recent Information-cost Stochastic Nonlinear Optimal Control (Info-SNOC) framework presented in (Nakka, Liu, et al., 2020) provides an especially valuable subset of stochastic optimal control. In particular, Info-SNOC is a sequential convex optimization algorithm that balances simultaneously exploring the gravity field and maintaining a collision-free, physically-feasible trajectory. First, the learning framework (such as a Neural Network) within Info-SNOC is responsible for learning any unmodeled dynamics terms using data from the spacecraft perception system. We label the data-driven approximation of the dynamics as the learned dynamics. Second, the stochastic optimal control framework takes in three critical inputs: (1) the learned dynamics, (2) the exploration objectives (i.e., preference to help the learning framework) and performance objectives (i.e., preference to reduce control magnitudes), and (3) the safety-critical chance constraints. With these three inputs, Info-SNOC produces an optimized, safe trajectory that balances the need to obtain information for the learning framework with the control limits of the spacecraft. Low-level, high-frequency controllers (such as a combination of short-horizon model predictive controllers and PID controllers) are responsible for keeping the spacecraft on the optimized trajectory. To further the Info-SNOC framework as an enabling strategy for spacecraft motion planning around small bodies, we need to understand and model the uncertainties in the learned dynamics.

1.3 Dangers of Uncertainty in the Dynamics

Of the possible uncertainties, uncertainty in the dynamics is particularly worrisome for spacecraft. Flying with an inaccurate dynamics model incurs substantial risks for proximity maneuvers and landing, including spacecraft collision with the small body. For proximity maneuvers, the highest magnitude term in the dynamics is the force of gravity from the small body on the spacecraft. Equation 1.1 describes the most general form of the gravitational potential ($U : \mathbb{R}^3 \to \mathbb{R}$) (Takahashi, 2013).

$$U = G \int_V \frac{\rho(\vec{r}')}{|\vec{r} - \vec{r}'|} dv \tag{1.1}$$

Where G is the gravitational constant, V is the integrating volume, $\rho(\vec{r}')$ is the density distribution, $\vec{r}$ is the radius vector of the spacecraft, and $\vec{r}'$ is the radius vector of the mass element. With appropriate mathematical steps, the potential obeys Poisson's equation as per equation 1.2. For empty space, $\rho \to 0$ to form Laplace's equation (Robert A. Werner and Daniel J. Scheeres, 1997).

$$\nabla^2 U = -\nabla \cdot \vec{g} = 4\pi G \rho \tag{1.2}$$

Where $\nabla^2 = \nabla \cdot \nabla$ is the second partial derivative of the spatial variables (i.e., the Laplace operator), $\nabla \cdot$ is the divergence with respect to the spatial variables, and $\vec{g} : \mathbb{R}^3 \to \mathbb{R}^3$ is the gravitational field (i.e., the force per mass of the spacecraft at a specified location). The critical insight of writing the gravitational field in this way is to recognize that gravity forms a *conservative field*. The force on an object due to gravity is only a function of position: the gravitational field does not depend on velocity, and it is path independent (i.e., it does not matter how the spacecraft reached that position). Despite the deceptively simple expression in equation 1.2, the density distribution is highly uncertain for small bodies.

The gravity of the small body is not the only force on the spacecraft. Additional contributions include solar radiation pressure, solar gravity, and small body outgassing. These forces largely depend on how far the spacecraft is from the sun and the small body of interest. The solar radiation pressure and small body outgassing are not conservative forces. Often, the solar radiation pressure can be treated as a bias term that pushes the spacecraft away from the sun. This is a time-varying process that depends on the sun's location relative to the spacecraft and the projected area of the spacecraft onto the solar radiation front. As per (Bate, Mueller, and White, 1971), the force (F_p) of solar radiation pressure is given by equation 1.3,

$$F_p = \frac{G_{SC}}{c} A \cos \alpha \tag{1.3}$$

where G_{SC} is the solar constant, c is the speed of light, and $A \cos \alpha$ is the projected area (that varies with time). (Bottke et al., 2006; Daniel J Scheeres, 2016; Yu, 2016) cover these perturbations in more detail.

With so many sources of uncertainty (i.e., the density distribution, position relative to the density distribution, the outgassing of the small body), how can we build a collision-free trajectory? Fundamentally, we need a way to estimate the dynamics using data from the perception system (namely data that includes images, LIDAR, RADAR, and accelerometer measurements), as noted in Figure 1.2. Since the gravity field is the most prominent part of the dynamics, we will focus on estimating this field. As in (Takahashi, 2013; Jones, 2010; Daniel J Scheeres, 2016), from here on, we label the estimated gravity field as the gravity model. To enable safe and autonomous spacecraft motion planning, the stochastic optimal controller (such as Info-SNOC) requires an efficient approximate estimate of the gravity field with the relevant positions that the trajectory may visit. The efficiency of the gravity model reflects (1) a sufficiently accurate model and (2) a model that the optimizer can evaluate sufficiently quickly. With the gravity model and the uncertainty information built into the stochastic optimal control framework, we can combat the dangers of uncertainties in the dynamics.

1.4 Core Problem: Developing an Efficient Gravity Model for Spacecraft Control

What makes the gravitational field around small bodies (such as those in Figure 1.1) so different than larger bodies? Unfortunately, the principal properties of preserving the early solar system history and containing valuable resources that make small bodies worthwhile mission targets also yield a complex gravity field. To better understand the difficulty of small body gravity models, we need to contrast them with large body gravity models.

Gravity Models for Large Bodies yield High Accuracy with Little Information

For large planet-scale or moon-scale bodies, efficient gravity models are readily available in the extensive literature for motion planning and long-term trajectory design between and around large bodies (Lo, 2002; Bate, Mueller, and White, 1971). The spherical shape of the solar system bodies means that the first-order point mass contribution dominates the dynamics. In many cases, the point mass term alone sufficiently accurately approximates approach and proximity orbits. To account for the body's oblateness or asymmetry across the equator, only a few additional spherical harmonics terms (i.e., J_2 and J_3, respectively) capture a significant amount of the orbit perturbations for proximity maneuvers.

Gravity Models for Small Bodies require Plentiful Information to Achieve Reasonable Accuracy

For small bodies, determining the gravity model is significantly more challenging than large planet-scale or moon-scale bodies. Without the large size to differentiate their interior nor enable significant resurfacing, small bodies retain their history of planetesimal collisions in the protoplanetary disk (Patterson, 1956; Amelin et al., 2002). The chaotic history of collisions means that the small bodies are often not circular or even elliptical. They have a broad range in densities from 1 gram per cubic centimeter (stony asteroids like 10 Hygiea) to 4 g/cc (stony-iron asteroids like 16 Psyche) to 8 g/cc (pure iron) (Walsh, 2018; Weiss and Elkins-Tanton, 2013). But, most critically, the density distribution need not be uniform as seen in pallasite meteorites or the asteroid 101955 Bennu (Walsh, 2018; D. J. Scheeres et al., 2020). With a small size, modest changes in the interior density distribution of the asteroid necessarily notably change the gravitational field close to the body.

Without a significant number of missions to a diverse set of small bodies, the planetary science and aerospace communities do not have empirical evidence for how small body density distributions and gravitational fields vary throughout the solar system, across asteroid spectral classes, or within asteroid families (Weiss and Elkins-Tanton, 2013). Investigating ways of determining the small body's gravity field and handling the estimated model in an efficient framework remains an active field of research (Oguri and Jay W McMahon, 2020; Daniel J. Scheeres and Marzari, 2003; B. Hockman, Reid, et al., 2017; B. Hockman and Pavone, 2017; B. J. Hockman et al., 2017).

A Cyclic Problem

The crux of the issue of developing a gravity model is its cyclic nature; it is a chicken-and-egg problem. We need a high-fidelity gravity model to plan a safe trajectory, but we simultaneously need to visit the asteroid on a safe trajectory to estimate the high-fidelity gravity model (Figure 1.3).

As with many cyclic problems, one can simultaneously tackle both sides of the cycle (safe trajectory planning and gravity model development) to address the chicken-and-egg complication. Current methods combine visual imagery and RADAR from Earth with visual imagery, LIDAR, and RADAR from the spacecraft (Daniel J Scheeres, 2016). Most commonly, engineers on Earth combine the data from different angles with stereo-photoclinometry (D. J. Scheeres et al., 2020). We

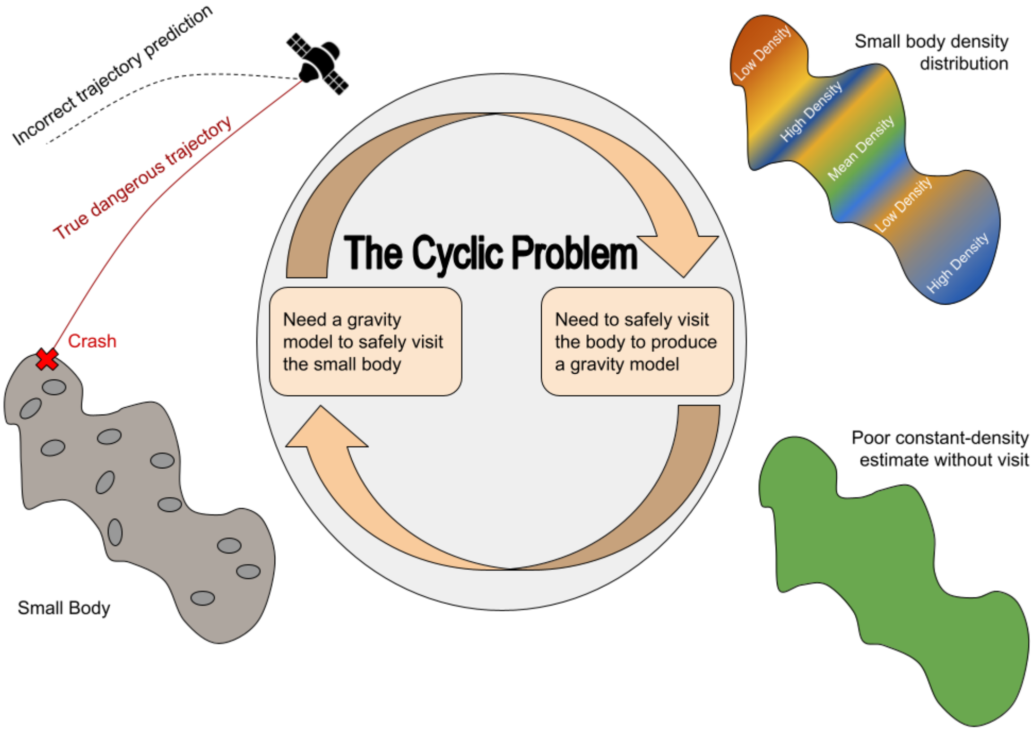

Incorrect trajectory prediction
True dangerous trajectory
Crash
Small Body
The Cyclic Problem
Need a gravity model to safely visit the small body
Need to safely visit the body to produce a gravity model
Small body density distribution
Low Density
High Density
Mean Density
Low Density
High Density
Poor constant-density estimate without visit

Figure 1.3: Graphic of the cyclic problem. Fundamentally, we need a gravity model to safely visit the small body, but we concurrently need to safely visit the small body to produce a gravity model. Consider the case where we fail to have an accurate gravity model. Then the spacecraft may believe it is on a safe trajectory when its true trajectory results in collision. On the flip side, consider the case where we fail to visit the small body. Then, we can only make coarse approximations of the gravity field (such as a constant-density assumption), which may not be accurate.

start with a point mass model to generate a safe approach trajectory far from the small body's surface. In orbit, we use the spacecraft's perception data to refine the gravity model. However, the Earth-in-the-loop strategy has three problems: (1) the perception data downlink (i.e., high-resolution images, LIDAR, RADAR, and accelerometer measurements) costs considerable communications strain, (2) the latency from the asteroid belt and beyond to Earth is on the order of minutes to hours, and (3) the state-of-the-art gravity shape models (such as those in blue in Figure 1.2) are computationally expensive to use and increasingly expensive with a more refined model.

1.5 Contributions of the Thesis: New Methods in Trajectory-Only Learned Dynamics

This thesis will leverage recent developments in learning frameworks (such as Neural Networks) to learn the gravity dynamics during flight (Nakka, Hönig, et al., 2021; Gao and Liao, 2019; Song, Cheng, and Gong, 2019; Cheng et al., 2020). Specifically, we seek to explore **the extent to which learning frameworks can effectively approximate the gravity field dynamics with data from recent trajectories.**

As the spacecraft gets closer to the small body, the magnitude of the perturbation increases, and it becomes easier to estimate the gravity field from the spacecraft's onboard accelerometers. As the gravity model improves, the spacecraft can transition to proximity maneuvers closer to the asteroid while maintaining spacecraft safety. The onboard learning strategy combats the problems above: (1) the perception data does not need to be downlinked to Earth since the learning framework can be run on board, (2) the perception to motion planning latency is small since the algorithm is run on board, and (3) the internal learning framework handles the gravity model at a constant evaluation complexity.

The Info-SNOC algorithm opens the door for learning-based methods in spacecraft motion planning. The most far-reaching learning architecture would be to learn the dynamics *only* with data from prior trajectories since this would enable safe travel to even the most unknown objects. So in this thesis, we want to explore this uncommon strategy in contrast to (Martin and Schaub, 2020; Song, Cheng, and Gong, 2019; Cheng et al., 2020; Peng and Bai, 2021) that sample points densely near the body regardless of preceding trajectories.

We split the thesis into five research objectives (Chapter 2). In chapters 3-5, we present trade-offs, rationales, and discussion to contextualize this research within physical systems. Moreover, the physics engine developed for this thesis is versatile and can be readily applied to future learned gravity model work. In chapters 6-8, we present the results of the three learning frameworks that we developed and examined. From the lessons learned in this thesis, we present a thorough list of recommendations to future researchers that also seek to combine complementary research across aerospace engineering, planetary science, and robotics (Chapter 9. Lastly, we close the thesis with a brief conclusion in Chapter 10.

Chapter 2

OBJECTIVES

Our overarching goal is to **investigate the extent to which learning frameworks effectively approximate the gravity field dynamics with data from recent trajectories**. From this goal, we distill five objectives and provide the rationale below.

1. **Develop a computationally efficient physics engine to build training data and test learning frameworks. Ensure that this framework can readily adapt to future learning frameworks and architectures**

   The first objective provides the critical software back-end for the last three objectives and enables future research. A versatile integrator takes either true dynamics for data generation or learned dynamics for evaluation. The physics engine simulates the spacecraft dynamics in orbit of the small body and the predicted trajectory from the learned dynamics. This objective is addressed in chapter 3.

2. **Analyze the trajectory design method in realizing a particular science mission criterion**

   In the second objective, the aim is to fit the research into the bigger picture of small body exploration. Specifically, the recent OSIRIS-REx mission to the asteroid 101955 Bennu serves as a timely and realistic benchmark mission to contextualize the thesis (B. Williams et al., 2018; Berry et al., 2013). With this context, we gauge where learned dynamics are most valuable in the orbits. Moreover, the mission context provides requirements that the learning-based methods from objectives 3-5 would have on the overall spacecraft system. This objective is addressed in chapter 4.

3. **Compare and contrast a set of multi-task multi-variate Gaussian Process Model learning frameworks**

   Gaussian Process Models would be particularly well suited to stochastic optimal control, conditional on computational efficiency and accuracy. The stochastic part of the controller relies on covariance information of the spacecraft state and in the learned dynamics. The mathematical overview is detailed in chapter 5. The Gaussian Process Model natively provides this covariance information (Rasmussen and C. K. I. Williams, 2006). Moreover, work by (Gao and Liao, 2019) suggests that Gaussian Process Regression may be suited for gravity field estimation. Hence, we start with the Gaussian Process Model as the first of three learning frameworks. This objective is addressed in chapter 6.

4. **Compare and contrast a Neural Network framework to the multi-task multi-variate Gaussian Process Model framework**

   To improve the computational efficiency, we then consider a Neural Network in the fourth objective. The trade-off for quicker evaluation is that the Neural Network does not natively provide covariance information. This objective is addressed in chapter 7.

5. **Compare and contrast a Physics-Informed Neural Network to the previous frameworks**

   The Neural Network in objective 4 goes hand-in-hand with the Physics-Informed Neural Network in objective 5. The physics-informed variant additionally encodes the gravity field's conservative field properties (Eq. 1.2) into the architecture. This objective is addressed in chapter 8.

With these five objectives, we achieve the overarching goal “investigate the extent to which learning frameworks effectively approximate the gravity field dynamics with data from recent trajectories." Objective 1 provides the back-end support for objectives 3-5 and a testing framework, objective 2 provides a reference to evaluate the efficiency performance, and the three different approaches in objectives 3-5 provide some breadth in learning frameworks.

Chapter 3

PHYSICS ENGINE FOR TRAJECTORY-ONLY LEARNED DYNAMICS

To develop any learning-based model, we require training data. Some robotics applications can access training data directly from field-based work. For example, one might drive a car for many thousands of miles to develop an inventory of high-fidelity sensor data, on-board driving data, and performance evaluation data. However, with such a limited number of missions to small bodies, as of yet, such data is not yet accessible. Instead, we want to simulate what a spacecraft might experience when in orbit around a small body.

3.1 Physics Engine Architecture

The physics engine has two key parts: the integrator and the dynamics model. We use a standard 4-th order Runge-Kutta integrator throughout all cases, though this is easily interchanged via the Python SciPy library or equivalently in other languages (such as DifferentialEquations.jl in Julia). On the other hand, the dynamics model is not easily available off the shelf. We have two types of dynamics models: (1) the reference or true dynamics model and (2) the learned dynamics model. Since we are concerned with isolating the effect of gravity, these dynamics models are equivalently gravity models.

Under the hood, the physics engine operates as follows. First, we build the training dataset (Figure 3.1, left). Given an initial state (x_0, v_0), a trajectory length (t), and the true gravity model (represented as a spherical harmonics expansion for this thesis), a Runge-Kutta integrator solves the vector ordinary differential equation initial value problem to produce a true propagated trajectory. Second, we provide the propagated true trajectory as a training set to the relevant learning framework (red dotted arrow in Figure 3.1). Namely, the learning framework has an $\mathbb{R}^3$ position input for an $\mathbb{R}^3$ acceleration output, just like the true gravity model. The timestamp and the velocity are not provided since the gravity field is a conservative field as noted in Section 1.3. Third, we replace the true gravity model with the learned gravity model (right). Running the Runge-Kutta integrator again, we produce a predicted propagated trajectory.

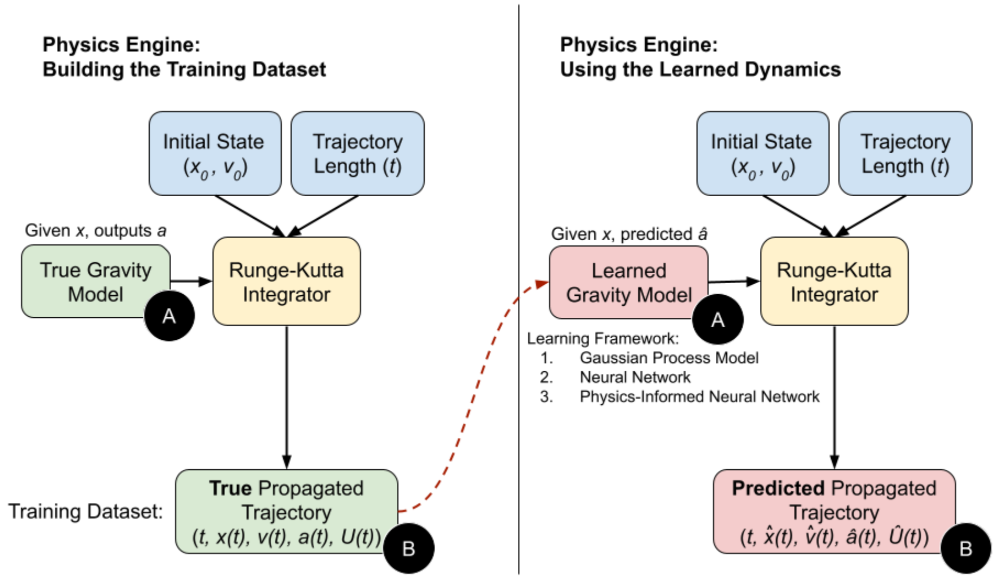

Physics Engine:
Building the Training Dataset
Initial State (x_0, v_0)
Trajectory Length (t)
Given x, outputs a
True Gravity Model
A
Runge-Kutta Integrator
Training Dataset:
True Propagated Trajectory
$(t, x(t), v(t), a(t), U(t))$
B
Physics Engine:
Using the Learned Dynamics
Initial State (x_0, v_0)
Trajectory Length (t)
Given x, predicted $\hat{a}$
Learned Gravity Model
A
Runge-Kutta Integrator
Learning Framework:
1. Gaussian Process Model
2. Neural Network
3. Physics-Informed Neural Network
Predicted Propagated Trajectory
$(t, \hat{x}(t), \hat{v}(t), \hat{a}(t), \hat{U}(t))$
B

Figure 3.1: Physics Engine Code Base Architecture. We use the true gravity model to build the training trajectory dataset (left side and green boxes). After training the learned model (dashed red arrow), we arrive at the trained learned gravity model (red). We then use the learned gravity model to build the predicted trajectory dataset (right side and red boxes). The test point between pointwise direct model outputs are labeled "A" and the test point between trajectory outputs are labeled "B"

To compare the true and learned models, we have two test points (labelled A and B in Figure 3.1). At A, we conduct a pointwise comparison between the models. Namely, at a given position x, we compare the output acceleration of the true gravity model $a(x)$ to the predicted acceleration of the learned gravity model $\hat{a}(x)$. At B, we conduct a trajectory propagation comparison between the models. Namely, at a given initial state and trajectory length, we propagate the trajectory forward. We then compare the true propagated trajectory (namely $x(t)$) to the predicted propagated trajectory (namely $\hat{x}(t)$).

3.2 Gravity Model Trade-Offs

The core of the physics engine is a small body gravity field to simulate the dynamics of the spacecraft. As noted in section 1.4, accurately modeling a small body's gravitational field is an active area of research in planetary science and aerospace engineering (B. Hockman and Pavone, 2017; B. Hockman, Reid, et al., 2017; Daniel J. Scheeres and Marzari, 2003; D J Scheeres, Mcmahon, et al., 2019; D J Scheeres, J. McMahon, et al., 2019; Takahashi, 2013). For the physics engine backend, we need an established non-learning gravity model to produce the trajectory

Table 3.1: Trade-off table for gravity model choices

Gravity Model	Model Variant	Computation Complexity	Availability	Realism
Spherical Harmonics	Zonal Only	$O(m)$	At small m	At large m
Spherical Harmonics	Zonal and Tesseral	$O(m)$	Not Often	At moderate m
Polyhedron	Constant Density	$O(n^2)$, $O(v)$	Readily	At moderate n
Polyhedron	Multi-Density	$O(n^2)$, $O(v)$	Not Often	At moderate n
Finite Elements	Cubic or Spherical	$O(n^3)$	Manual Conversion	At moderate n
Finite Elements	Tetrahedral	$O(n^3)$	Manual Conversion	At moderate n

The green boxes indicate the best case in a given column. The orange boxes are the medium case and the red boxes are worst case conditions. Unfortunately, the gravity models that are the most realistic are generally not readily available.

data for the learning-based methods. We need high versatility (i.e., the ability to quickly fabricate different models to test the learning framework performance), quick evaluation to produce training data, and parameter availability to compare to real world objects, where applicable. Three primary methods include Spherical Harmonics Expansions, Polyhedron Models, and Finite Element Approximations. The trade-offs are summarized in Table 3.1 and explained in the paragraphs below.

As noted in Section 1.4, for large bodies, the spherical harmonics expansion is the method of choice. The spherical harmonics provide high accuracy with only a few terms. Moreover, since the expansion yields only a single additional expression per additional term (m), this method has a linear and generally low computational complexity. For small bodies, higher-order terms in the expansion are still prominent and should be included. While the computational complexity is maintained (since it is still a linear increase in the number of steps per number of included terms), the accuracy is less reliable for small bodies, especially for irregularly shaped bodies (B. Hockman and Pavone, 2017) or binary bodies (Daniel J Scheeres, 2016). If we orient the Cartesian axes along the rotation axis of the small body and the body is sufficiently symmetric (such as the asteroid 101955 Bennu), the zonal spherical harmonic terms are generally the most prominent in the expansion. But, for completely asymmetric asteroids or tumbling asteroids, the zonal terms are not enough. We would have to include the tesseral and sectoral terms. This distinction is further discussed in Appendix A. Often in the literature, only the zonal terms are estimated, as in (D. J. Scheeres et al., 2020). In some cases, the tesseral terms are also estimated, albeit often from shape models rather than direct measurements, as in (Takahashi, 2013). Lastly, the spherical harmonics expansion is only valid outside of the circumscribing sphere (also known as the Brillouin sphere, see Chapter 4 for more details). Inside of the sphere, the series does not converge (Robert A. Werner and Daniel J. Scheeres, 1997; Martin and Schaub, 2020). In summary, the zonal spherical harmonics expansion has linear computational complexity, is available for select asteroids, and requires many terms (about 8-12) for realism.

For small body scientific research, many in the field have turned to polyhedron models. These shape models are the community's highest fidelity data products for small body shapes and approximate gravity fields (namely, using a constant density assumption). NASA's Planetary Data System Small Body Node hosts a large “Shape Models of Asteroids, Comets, and Satellites" database including high-resolution shape models from spacecraft missions and lower-resolution shape models of Near-Earth Asteroids from ground-based radio science (Davis et al., 2021). Often, stereophotoclinometry with optical and radar imagery is the primary method for generating the shape model from spacecraft flybys or orbits as in (D. J. Scheeres et al., 2020). The shape model is not, however, a density distribution model. As mentioned in Section 1.4, the missing piece of the density distribution is critical to understanding the gravity field. In the literature, the polyhedron model is often used under a constant density assumption which is too strong of an assumption in most cases (Section 1.4). When the assumption breaks, this leads to a biased estimator of the true gravity field. Recent work in (Takahashi, 2013), explores extending the shape model to a multi-density model. While this method improves the gravity model overall, it has large errors at the density boundaries, is difficult to estimate, and not readily available.

In addition to issues in modeling the density distribution, both polyhedral model versions (i.e., the constant density and multi-density versions) have a large computational load. For every added vertex, we add multiple edges and multiple faces. These all increase linearly with the number of points. Specifically, for v vertices, we have $2(v-2)$ faces and $3(v-2)$ edges (Daniel J Scheeres, 2016). But, to evaluate the potential, we need to calculate the position difference from edge to spacecraft ($\vec{x}_e$) and face to spececraft ($\vec{x}_f$). Thereafter, we determine the normal components of each face and form a matrix product. This quickly increases the computation time because we have to evaluate the distance ($\vec{x}_e$ and $\vec{x}_f$) from scratch each time. Plus, we then need to reevaluate the matrix product and series. The primary calculation is in equation 3.1. The full mathematical formalism and derivation can be found in (Robert A. Werner and Daniel J. Scheeres, 1997) and summarized in (Daniel J Scheeres, 2016).

$$U(\vec{x}) = \frac{G\rho}{2}\left(\sum_{e\ \in\ \text{edges}} \vec{x}_e \cdot \mathbf{E}_e \cdot \vec{x}_e\, L_e - \sum_{f\ \in\ \text{faces}} \vec{x}_f \cdot \mathbf{F}_f \cdot \vec{x}_f\, \omega_f\right) \tag{3.1}$$

where G is the gravitational constant, ρ is the density, L_e scales the term by the potential of a wire with the relevant dimensions, and ω_f scales the term by the potential of a face with the relevant dimensions. The matrices $\mathbf{E}_e$ and $\mathbf{F}_f$ express the orientation of the edge e or face f, respectively.

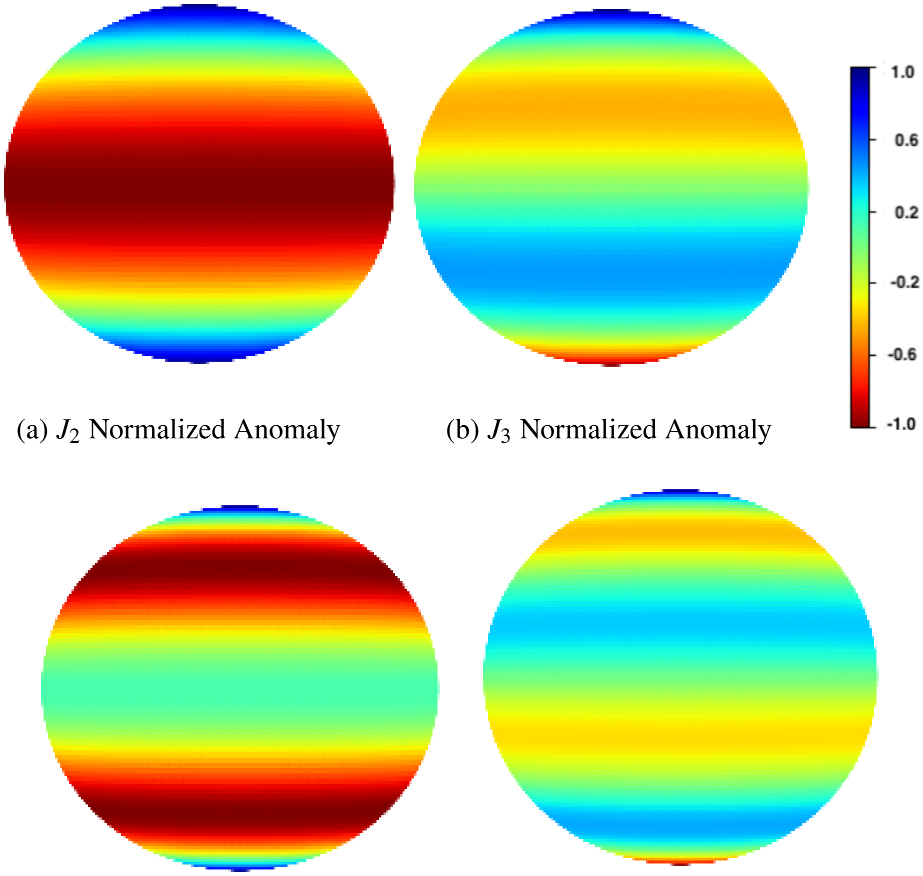
Producing a model with three times the resolution requires nine times the number of faces (e.g., starting with a square, subdividing the sides in 3 yields 9 new squares). In the multi-density model, we have an additional metric of depth d into the body. Then, producing a multi-density model with three times the resolution requires $9d$ times the number of faces. Hence, we get an overall $O(v)$ number of evaluations and an $O(n^2)$ increase in the number of faces per resolution parameter n. With thousands of points for high fidelity models, this method becomes impractical for trajectory planning. To mitigate this problem, the gravity field is often sampled at prescribed intervals and interpolated during planning (B. Hockman and Pavone, 2017). In summary, the shape model method is a promising method for high versatility gravity models, but the large computational load is a serious problem for onboard processing on real missions.

The last technique is a finite element method (FEM) or Riemann Sum approach for gravitational contexts as in (Park, Robert A Werner, and Bhaskaran, 2010). FEM is a well-studied field within mechanical engineering and is promising in terms of realism. However, at current, without an instrument to determine the density of the asteroid interior, the Riemann Sum method or FEM-type method can only achieve an accuracy as good as the polyhedron methods. Moreover, voxel-based or tetrahedral-based asteroid models are not available and would require a manual conversion from a polyhedron model. Lastly, a naive Riemann Sum approach would require the cube of the number of voxels where we want to refine the model. This baseline approach is prohibitively computational complex.

Hence, for the purposes of this thesis, the zonal spherical harmonics expansion is the right fit (Table 3.1). First, the zonal spherical harmonics expansion is the most versatile option. We can quickly fabricate new models by changing only a few parameters (namely, we update m constants). In contrast, to fabricate a new polyhedron model, we would need to update $v \sim 10^3$ vertices. To fabricate a new FEM model, we would need even larger updates. Second, since the number of harmonics terms ($m \sim 10$) for the spherical harmonics expansion is much smaller than the number of vertices ($v \sim 10^3$) of the polyhedral model (and v is much smaller than the voxels needed for a FEM model), the spherical harmonics expansion is much more computationally efficient. Moreover, to produce a more detailed spherical harmonics model, we simply add an additional harmonics term (i.e., the complexity scales as $O(m)$). This outperforms the polyhedron and finite element methods which must produce a finer mesh for a more detailed model. In the worst case, this scales as $O(n^2)$ for the polyhedron models and $O(n^3)$ for the finite element methods. The spherical harmonics is even more computationally efficient with more detailed models. Though, we also must keep in mind that we only get the accuracy up to the point where the expansion loses validity: the coefficients are negligible, the orbit is inside the Brillouin circumscribing sphere, or we need missing tesseral or sectoral terms (see Chapter 4 for more details). Lastly, we have available parameters for the zonal harmonics expansion, but not the full zonal, tesseral, and sectoral expansion. Therefore, the zonal spherical harmonics expansion is the right fit for the physics back-end of the project: highly versatile, computationally favorable, sufficiently realistic, and decently available.

3.3 Zonal Spherical Harmonics

Following the trade-offs above, this thesis will use the zonal spherical harmonics expansion as the back-end gravity model in the physics engine. To use the expansion with the integrator, we require a mathematical description of the zonal spherical harmonics dynamics.



(a) J_2 Normalized Anomaly

(b) J_3 Normalized Anomaly

(c) J_4 Normalized Anomaly

(d) J_5 Normalized Anomaly

Figure 3.2: Side view of normalized J_n term anomaly. For each subfigure, the values are rescaled between -1 and 1 to illustrate the most contrast (see shared legend in the top right). Values near -1 (deep red) correspond to a negative anomaly: low-density regions or lack of material. Values near 1 (deep blue) correspond to a positive anomaly: high-density regions or excess material. The even harmonics (such as J_2 and J_4) are symmetric about the equator. The odd harmonics (such as J_3 and J_4) are symmetric in absolute value but flipped about the equator. For example, J_3 shows a negative anomaly band at mid latitudes in the north (orange region) and a positive anomaly band of the same magnitude in the south (blue region). Similarly, the poles are the same direction (namely positive anomaly) in the even harmonics (deep blue regions) but are the opposite direction (namely positive anomaly in the north and negative anomaly in the south) in the odd harmonics (deep blue and deep red regions, respectively).

For mathematical convenience, the expansion is split into the point mass term and the zonal harmonics terms. The point mass term is presented in equation 3.2.

$$\vec{a} = \frac{\vec{F}_{grav,\ point}}{m_s} = \frac{\mu}{r^2}(-\hat{r}) = \frac{-\mu\vec{r}}{r^3} \tag{3.2}$$

where $F_{grav,\ point}$ is the force of gravity of the point mass, m_s is the satellite mass, μ is the central body's Gravitational Parameter (i.e., GM where G is the Gravitational Constant and M is the mass of the central body), r is the radius magnitude, $\hat{r}$ is the radius unit vector, and $\vec{r}$ is the radius vector. The zonal harmonics terms are presented in equations 3.3 - 3.5.

$$a_x = \frac{Kx}{|r|^2 - z^2}\left[|r|P_n\left(\frac{z}{|r|}\right) - zP_{n+1}\left(\frac{z}{|r|}\right)\right] \tag{3.3}$$

$$a_y = a_x\frac{y}{x} \tag{3.4}$$

$$a_z = K\ P_{n+1}\left(\frac{z}{|r|}\right) \tag{3.5}$$

where K is given by equation 3.6, (x, y, z) are the Cartesian coordinates, and P_n is the n-th degree Legendre P Polynomial.

$$K = \mu\tilde{J}_n(n+1)R^n r^{-(n+2)} \tag{3.6}$$

where $\tilde{J}_n$ is the dimensionless n-th Zonal Coefficient and R^n is the normalizing radius of the central body. The even spherical harmonics terms (e.g., J_2) are symmetric about the equator whereas the odd harmonics (e.g., J_3) terms describe asymmetric bulges (Figure 3.2). The full derivation of the acceleration from the zonal spherical harmonics terms is presented in Appendix A. These coefficient terms are recovered via multiple techniques, though regression and filtering techniques are common in the mission post-processing (Jones, 2010).

3.4 Physics Engine Equations of Motion

Equipped with the point mass term in equation 3.2 and the zonal terms in equations 3.3 - 3.5, we can write the spacecraft's equations of motion.

$$\sum\vec{F} = m_s\vec{a} \tag{3.7}$$

$$\frac{-\mu m_s\vec{r}}{|r|^3} + m_s\sum_{n=2}^{N_z}\left[a_{x,n,\text{zonal}}\hat{x} + a_{y,n,\text{zonal}}\hat{y} + a_{z,n,\text{zonal}}\hat{z}\right] + \vec{T} = m_s\vec{a} \tag{3.8}$$

$$\frac{-\mu\vec{r}}{|r|^3} + \sum_{n=2}^{N_z}\left[a_{x,n,\text{zonal}}\hat{x} + a_{y,n,\text{zonal}}\hat{y} + a_{z,n,\text{zonal}}\hat{z}\right] + \frac{\vec{T}}{m_s} = \vec{a} \tag{3.9}$$

Splitting up the terms in the series by Cartesian direction yields equation 3.11.

$$\begin{aligned}\vec{a} =& \frac{-\mu x}{|r|^3}\hat{x} + \frac{-\mu y}{|r|^3}\hat{y} + \frac{-\mu z}{|r|^3}\hat{z} \\ &+ \sum_{n=2}^{N_z}\left[a_{x,n,\text{zonal}}\hat{x}\right] + \sum_{n=2}^{N_z}\left[a_{y,n,\text{zonal}}\hat{y}\right] + \sum_{n=2}^{N_z}\left[a_{z,n,\text{zonal}}\hat{z}\right] \\ &+ \frac{\vec{T}}{m_s} \end{aligned} \tag{3.10}$$

$$\begin{aligned}\vec{a} =& \left(\frac{-\mu x}{|r|^3} + \sum_{n=2}^{N_z}\left[a_{x,n,\text{zonal}}\right] + \frac{T_x}{m_s}\right)\hat{x} \\ &+ \left(\frac{-\mu y}{|r|^3} + \sum_{n=2}^{N_z}\left[a_{y,n,\text{zonal}}\right] + \frac{T_y}{m_s}\right)\hat{y} \\ &+ \left(\frac{-\mu z}{|r|^3} + \sum_{n=2}^{N_z}\left[a_{z,n,\text{zonal}}\right] + \frac{T_z}{m_s}\right)\hat{z} \end{aligned} \tag{3.11}$$

Hence, we convert this second-order vector ordinary differential equation into a system of first-order differential equations with the state vector as the stack of position and velocity.

$$\frac{d}{dt}\begin{bmatrix} x \\ y \\ z \\ v_x \\ v_y \\ v_z \end{bmatrix} = \underbrace{\begin{bmatrix} v_x \\ v_y \\ v_z \\ \frac{-\mu x}{|r|^3} + \sum_{n=2}^{N_z}\left[a_{x,n,\text{zonal}}\right] \\ \frac{-\mu y}{|r|^3} + \sum_{n=2}^{N_z}\left[a_{y,n,\text{zonal}}\right] \\ \frac{-\mu z}{|r|^3} + \sum_{n=2}^{N_z}\left[a_{z,n,\text{zonal}}\right] \end{bmatrix}}_{\text{state contribution}} + \underbrace{\frac{1}{m_s}\begin{bmatrix} 0 \\ 0 \\ 0 \\ T_x \\ T_y \\ T_z \end{bmatrix}}_{\text{control contribution}} \tag{3.12}$$

While this is a highly nonlinear dynamics equation, the Runge-Kutta integrator can still efficiently and accurately solve the initial value problem.

Chapter 4

SMALL BODIES THROUGH THE LENS OF ASTEROID 101955 BENNU

The recent *OSIRIS-REx* mission visited the asteroid 101955 Bennu and will return samples to Earth in 2023. In terms of proximity maneuvers and sample collection, 101955 Bennu is a great target. As a carbonaceous asteroid, we can probe scientific questions for one of the largest asteroid spectral groups. On a near-Earth orbit, we design lower fuel trajectories. As a nearly rotationally and equatorially symmetric small body, we can accurately apply the zonal spherical harmonics expansion (D. J. Scheeres et al., 2020).

In chapter 3, we showed that a zonal spherical harmonics expansion optimized our design for the back-end reference gravity model. With the *OSIRIS-REx* mission to 101955 Bennu, we have the necessary J_2 to J_5 parameters to build a fifth-order expansion of the gravity model. But, as noted in chapter 3, the spherical harmonics expansion is only valid between the Hill sphere on the upper end and the Brillouin sphere on the lower end. Outside of this region, the spherical harmonics can be used, but the accuracy will suffer (Robert A. Werner and Daniel J. Scheeres, 1997; Takahashi, 2013; Takahashi and D. J. Scheeres, 2014). In this chapter, we seek to more formally investigate these bounds in the limits of validity and demonstrate their implications for small body missions, especially from the perspective of the physics engine and trajectory-only learned dynamics.

4.1 Upper Bound on Validity: The Hill Sphere

The Hill sphere represents the sphere whereby the gravitation influence of the secondary (e.g., an asteroid) begins to overtake the gravitation influence of the primary (e.g., the sun). The radius that parameterizes this sphere is given in equation 4.1.

$$r_H \approx a(1-e)\sqrt[3]{\frac{m}{3M}} \tag{4.1}$$

where a and e are the semi-major axis and eccentricity, respectively, of the secondary's orbit with respect to its orbit around the primary. Following, m is the mass of the secondary, and M is the mass of the primary. The Hill sphere is an upper bound since other perturbations, such as solar radiation pressure will further decrease the upper bound radius where the secondary's gravitational influence exceeds that of the primary.

For reference, we have plotted the Hill sphere radii for different solar system bodies using data from Wolfram Alpha (Figure 4.1). (Note that the Sun's mass is 1.989×10^{30} kg). Most notably, 4 Vesta has a mass about 10^9 times larger than 101955 Bennu with a much smaller eccentricity. As such, 101955 Bennu's Hill radius is very small. A linear regression on the presented data yields a slope estimate of 0.3288 ± 0.0155 in log-log space (p-value of 2.6×10^{-8}). At essentially a third, this makes the power law in equation 4.1 where the mass m varies much more than the semimajor axis a or the eccentricity e. Namely, Earth has $a(1 - e) = 0.98$ AU, 67P/Churyumov–Gerasimenko has $a(1 - e) = 1.24$ AU, and 101955 Bennu has $a(1 - e) = 0.90$ AU.

4.2 Lower Bound on Validity: The Brillouin sphere

The Brillouin sphere represents a mathematical shift whereas the Hill sphere represents a physical shift between the gravitational dominance of the sun and the secondary (e.g., Earth or 101955 Bennu). The spherical harmonics expansion is fundamentally a convenient series expansion to solve Laplace's equation (Eq. 1.2). The spherical harmonics expansion is only valid when the position of the spacecraft is outside the smallest sphere that fully encloses the primary in the reference coordinate frame. In the small body literature, this smallest sphere is the Brillouin sphere. Once we cross into this sphere, the series does not converge because it is not meant to converge in this region. Instead, we could use a different series expansion, such as in (Takahashi, 2013).

Objects that have smaller Hill spheres also have smaller Brillouin spheres. Quantum mechanics predominantly governs the density of materials and concentrates densities between 1 gram per cubic centimeter and 10 grams per cubic centimeter, especially in a planetary science context. While Earth's Brillouin sphere radius (its equatorial radius) is 6738 km, 101955 Bennu's Brillouin sphere radius (also its equatorial radius) is a mere 0.28-0.29 km.

In contrast to real life, we can form arbitrarily denser objects in simulation to lower the Brillouin sphere radius and ensure convergence. The physics engine from this thesis does not include the object shape or size. As such, the physics engine is completely unaware of the small bodies Brillouin sphere except for the purposes of normalizing the units (see Appendix C). Thus, the spherical harmonics expansion

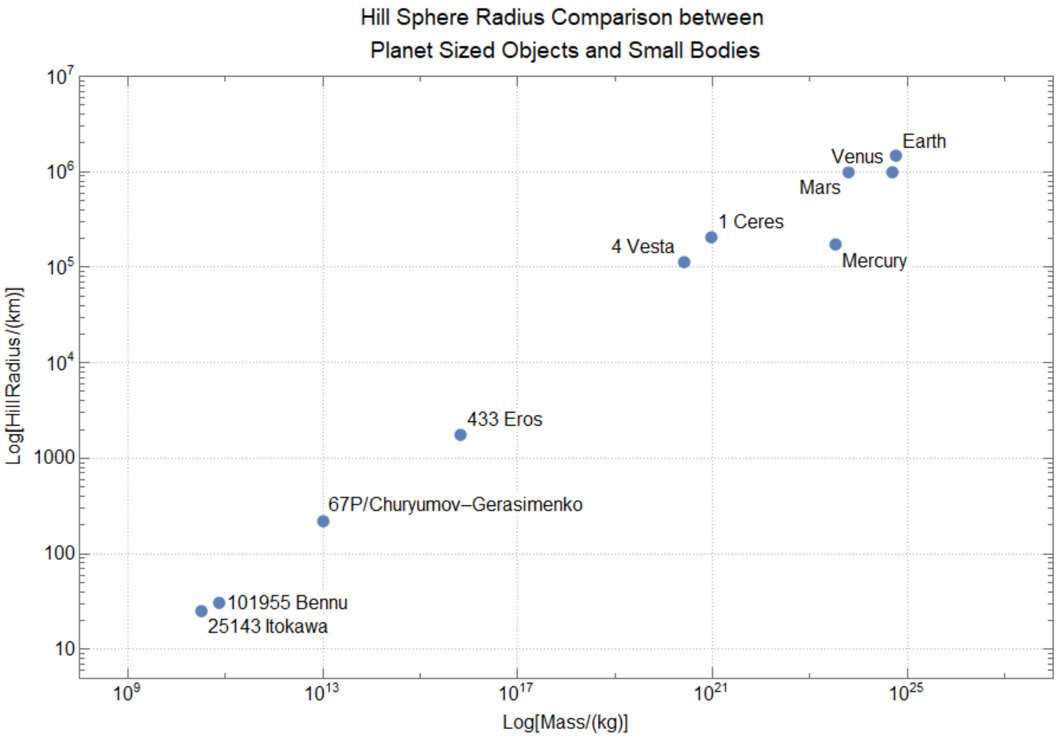


Figure 4.1: Comparison of Hill sphere radii between larger planet sized objects and small bodies. As fairly large terrestrial bodies, Venus, Earth, and Mars are separated from the other objects. Mercury, the last terrestrial planet is separated from the rest with a similar Hill Radius to 1 Ceres (a dwarf planet and the largest asteroid belt object). 4 Vesta is the second largest asteroid belt object. 433 Eros, 101955 Bennu, and 25143 Itokama are 3 small body targets from past missions. 67P/Churyumov–Gerasimenko is a Jupiter family comet. The other small bodies from Figure 1.1 are not included since their masses are not well known. The apparent power-law relation is presented in equation 4.1.

in the physics engine is defined everywhere except the origin.

4.3 Zonal Harmonic Dominance within the Region of Validity

Within the region of validity, the point term dominates even near the surface. At the Hill sphere radius, the potential is at its smallest (within the region of validity). Since the terms of the spherical harmonics expansion drop off as $\frac{1}{r^{n+1}}$ (Appendix A), the point mass is the only meaningful contribution. As we get closer to the body, the higher-order harmonics become more relevant. We can more intuitively compare the contribution of each term in the spherical harmonics expansion to the specific potential at the Hill sphere radius. In particular, the contribution of the n-th term in the series is given in equation 4.2.

$$u_n(r,\theta) = \frac{-\tilde{J}_n \mu R^n P_n^0(\sin(\theta))}{r^{n+1}} \tag{4.2}$$

The potential at the Hill sphere is from the point mass term at the Hill Radius (Eq. 4.3).

$$u_H = \frac{-\mu}{r_H} \tag{4.3}$$

With the point mass potential at the Hill sphere, we can form a dimensionless ratio between the n-th term in the series (u_n) and the Hill sphere (u_H), as in equation 4.4.

$$\frac{u_n}{u_H} = \frac{\tilde{J}_n R^n r_H P_n^0(\sin(\theta))}{r^{n+1}} \tag{4.4}$$

To visually represent this relation, we form a polar plot showing a contour where the n-th term in the series (u_n) is a specified small fraction the Hill sphere (u_H). Namely, $\frac{u_n}{u_h} = \gamma$.

$$\frac{u_n}{u_H} = \gamma = \frac{\tilde{J}_n R^n r_H P_n^0(\sin(\theta))}{r^{n+1}} \tag{4.5}$$

$$r^{n+1} = \frac{\tilde{J}_n R^n r_H P_n^0(\sin(\theta))}{\gamma} \tag{4.6}$$

$$r = \left(\frac{\tilde{J}_n R^n r_H P_n^0(\sin(\theta))}{\gamma}\right)^{\frac{1}{n+1}} \tag{4.7}$$

We can now plugin values for 101955 Bennu given in Table 4.1 with data from (D. J. Scheeres et al., 2020). More detailed plots of the spherical harmonics and the relevant drop-off in the n-th component are also detailed in (D. J. Scheeres et al., 2020).

Plotting the radius at which a given n-th harmonic is 10% of the potential energy of the point mass term at the Hill sphere ($|\gamma| = 10\%$) yields Figure 4.2. The absolute value accounts for the flip between positive and negative values (Figure 3.2). Matching (Daniel J Scheeres, 2016), the J_2 term far exceeds the impact of the other terms in the harmonics expansion. Even then, the J_2 term is only 10 % of the Hill sphere potential when it is with merely 3 times the bounding radius. At a distance $r = 3R$, the point mass term is 35 times (3500%) the Hill sphere potential for the asteroid 101955 Bennu.

Culminating from these results and analysis, setting up a learning framework that learns both the point mass term and the spherical harmonics terms is quite different

Table 4.1: 101955 Bennu gravity field and size parameters

Property	Value
Brillouin sphere radius (R)	290 m
Gravitational parameter (μ)	4.89 $\mathrm{m}^3\,\mathrm{s}^{-2}$
$\tilde{J}_2$	1.93×10^{-2}
$\tilde{J}_3$	-1.22×10^{-3}
$\tilde{J}_4$	-6.50×10^{-3}
$\tilde{J}_5$	6.73×10^{-3}

The Brillouin sphere radius bounds the asteroid. In combination with the gravitational parameter (μ), we use the Brillouin sphere radius to normalize each of the spherical harmonics terms (see Appendix C). Table is adapted from (D. J. Scheeres et al., 2020).

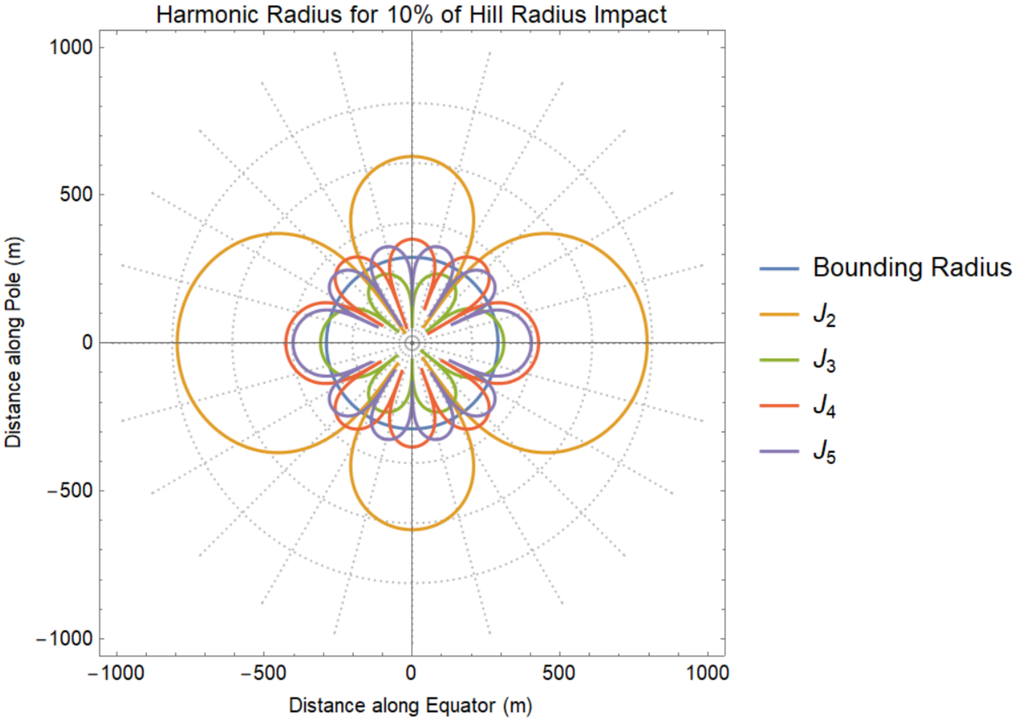


Figure 4.2: Radius as a function of co-latitude θ where the specific potential of the n-th harmonic term is 10% of the specific potential at the Hill Radius. Within each contour, the impact of the term increases as $|\gamma|^{n+1}$. Recall that the spherical harmonics are not valid within the bounding radius. For Bennu and similar asteroids, the harmonics are only appreciable in close proximity to the surface.

than learning only one of the two. If the spacecraft's orbit is outside a radius two times the small body's Brillouin sphere radius, it will be very difficult to learn or estimate the higher-order terms. Moreover, for the stochastic optimal control problem, the uncertainty in the potential and acceleration is much higher closer to the surface. Therefore, to improve the gravity model to enable landing at the surface we would likely require higher eccentricity orbits. By lowering the perigee, we can sequentially learn the gravity field in proximity to the small body without committing to a low circular orbit that renders the entire orbit in a high uncertainty zone.

Chapter 5

METHODS FOR TRAJECTORY-ONLY LEARNED DYNAMICS WITHIN STOCHASTIC OPTIMAL CONTROL

To set the context for the learning-based methods to estimate the gravity field, we start with identifying the gravity model within the stochastic optimal control framework. As presented in (Nakka, Liu, et al., 2020), the stochastic optimal control framework minimizes the expectation of a cost function over the state and control (Eq. 5.1) subject to a set of constraints. The primary constraint is the dynamics constraint (Eq. 5.2), which ensures that the equation is dynamically feasible (i.e., it obeys the relevant laws of physics). The next constraint (Eq. 5.3) describes a chance constraint with respect to the state of the system along the trajectory. We include the controls constraint equation 5.4 to check that the controls are mechanically feasible (i.e., the controller is not asking the spacecraft to exceed its mechanical thrust or pointing capabilities). The last constraint (Eq. 5.5) fixes the boundaries of the trajectory (i.e., the trajectory starts at the spacecraft's current state and ends at the destination state).

$$\begin{aligned}
J^* = \min_{x(t),\, \bar{u}(t)} \quad & \mathbb{E}\left[\int_{t_0}^{t_f} J(x(t),\bar{u}(t))dt + J_f(x(t),\bar{u}(t))\right] && (5.1)\\
\text{subject to} \quad & \dot{x}(t) = f(x(t),\bar{u}(t)) + \hat{g}(x(t),\bar{u}(t)) && (5.2)\\
& \Pr(x(t) \in \mathcal{F}) \geq 1-\epsilon, \quad \forall t \in \left[t_0, t_f\right] && (5.3)\\
& \bar{u}(t) \in \mathcal{U} \quad \forall t \in \left[t_0, t_f\right] && (5.4)\\
& x\left(t_0\right) = x_0 \quad \mathbb{E}\left(x\left(t_f\right)\right) = \mu_{x_f} && (5.5)
\end{aligned}$$

where $x(t)$ and $\bar{u}(t)$ are the state and control trajectories, respectively. J^* is the optimal cost, J is a function that penalizes the state and control trajectories based on a user-defined metric (e.g., distance away from a baseline trajectory), and J_f functions similarly to J, but places an additional cost at the endpoint. In the Info-SNOC framework, we specifically choose a cost function where we can tune between an information-seeking trajectory and a fuel optimal cost. The expectation operator $\mathbb{E}$ is linear and positive. $\dot{x}(t)$ is the time derivative of the state trajectory, which we express as two functions f and $\hat{g}$. The function f is the known part of the dynamics and the function $\hat{g}$ is a learned approximation of g. Essentially, in reality, we have $\dot{x} = f(x,u) + g(x,u)$, but g is unknown to the controller, so our best approximation is $\dot{x} = f(x, u) + \hat{g}(x, u)$. The expression $\Pr(x(t) \in \mathcal{F})$ expresses the probability that the state $x(t)$ is within a safe set $\mathcal{F}$. The user chooses ϵ small enough to express confidence about the safety of the trajectory. The safe set for obstacle avoidance (e.g., not crashing into the small body) can be described as a disjunctive convex set as in (Blackmore, Ono, and B. C. Williams, 2011) to enable computational efficiency. Lastly, the set $\mathcal{U}$ is the set of mechanically feasible controls. The set includes restrictions on thrust angle, thrust magnitude, and pointing constraints). Similar to collision constraints, we can losslessly convexify most thrust constraints as in (Behçet and Blackmore, 2011).

For this thesis, the most important part is the dynamics constraint (Eq. 5.2). For spacecraft control, the gravity model is not known beforehand. Hence, the true gravity field is g in the aforementioned framework with the gravity model as $\hat{g}$. The conceptual split between g and $\hat{g}$ is directly incorporated into the physics engine architecture (Chapter 3, especially Figure 3.1). Info-SNOC's ability to handle learned dynamics within the stochastic optimal control framework means that we can choose a learning-based approximation framework for the gravity model.

5.1 Methodological Consistencies across Learning Frameworks

Chiefly, we must develop an overall learning framework architecture that matches the top-level goal of trajectory-only learning. From the learning framework perspective, we start with a training data set from past trajectories (for more details on data generation and the physics engine, please see Chapter 3). With the training data, the learning framework develops a model that approximates the underlying dynamics that created the training data. For this thesis, we are isolating the gravity model and treating it as completely unknown. Therefore, we do not have any “known" part of the dynamics (mathematically $f(x(t), \bar{u}(t)) = 0$). In reality, we would include terms such as average orientation-dependent solar radiation pressure. Moreover, we are assuming that we are on a coasting orbit and are not applying any thrust (mathematically $g(x(t), \bar{u}(t)) \rightarrow g(x(t))$ and $\hat{g}(x(t), \bar{u}(t)) \rightarrow \hat{g}(x(t))$). This leaves us with

$$\dot{x} = g(x) \tag{5.6}$$

In contrast, we can use the trained model to propagate the orbit with

$$\dot{x} = \hat{g}(x) \tag{5.7}$$

as would be done within the Info-SNOC framework. Third, we evaluate the learning frameworks with two methods: (1) the accuracy of the trained dynamics model evaluated in the acceleration space and (2) the accuracy of the propagated trajectory evaluated in the position space. We do not need to explicitly test the generalizability of the learned dynamics outside the region near the trained trajectories since the error build-up in the propagated trajectories moves the gravity model evaluation outside the trained region, though this can be done in future work via a test trajectory formed from a different initial condition than the training trajectories.

5.2 Methodological Differences across Learning Frameworks

The Gaussian Process Model is advantageous in its ability to handle uncertainty and noise in the system. Moreover, when we tried the Gaussian Process Model with low noise (or no noise) data, we discovered that the GPyTorch Gaussian Process Model solver meets numerical errors. It is currently hypothesized that this low error causes a singular or near singular matrix in the solver. The result is that the optimizer's update step is inaccurate, and the model performance suffers.

Instead of low noise data, we mimic online spacecraft applications which will have uncertainty. Specifically, most spacecraft will have larger uncertainty in state than acceleration since the state is predicted (as in (Bhaskaran et al., 2011; Lorenz et al., 2017)) while the acceleration is measured. Hence, we introduce noise on the state as $a = a(x+\epsilon)$, where a is the acceleration, x is the position, and ϵ is Gaussian error. Alternatively, we could consider a hypothetical where the noise on the acceleration is larger than the state. Then, the noise could be added to the output as $a = a(x) + \epsilon$, which is more common in learning applications. Lastly, we could add both. In all cases, the error is exogenous to have accurate regressions.

In contrast, as implemented in this thesis, the Neural Network and the Physics-Informed Neural Network do not provide uncertainty information and are not especially robust to noise. For these cases, we substantially decrease the noise on the trajectory to reflect that the Neural Network and the Physics-Informed Neural Network are not intended for these noisy cases.

The next big difference is the input and output of the Neural Networks. The Gaussian Process Model and Neural Network must learn the acceleration $\vec{a}$ as a function of position $\vec{x}$ (i.e., $\hat{g} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$) as:

$$\vec{a} = \vec{a}(\vec{x}) \tag{5.8}$$

$$\vec{a} = \begin{bmatrix} a_x(x, y, z) \\ a_y(x, y, z) \\ a_z(x, y, z \end{bmatrix} \tag{5.9}$$

In contrast, the Physics-Informed Neural Network learns the potential $U = U(\vec{x})$ (Martin and Schaub, 2020). This strategy then changes the cost function to represent the underlying conservative relationship of the gravity field (Eq. 5.10).

$$\vec{a} = -\nabla U \tag{5.10}$$

Unlike the previous learning architectures where the system operates as $f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$, the PINN framework is $f : \mathbb{R}^3 \rightarrow \mathbb{R}$. Then, to recover the accelerations ($\vec{a}$), (Martin and Schaub, 2020) use the auto-differentiability of the network. The resulting cost function (J) is the mean-squared error loss between the measured accelerations ($\vec{a}_1$) and the auto-differentiated potential (∇U) as per equation 5.11.

$$J(\Theta) = \frac{1}{N_f} \sum_{i=1}^{N_f} |\vec{a}_i - \nabla \hat{U}(x_i|\Theta)|^2 \tag{5.11}$$

The weights (Θ) are updated with gradient descent. The key advantage is that the conservative relationship of the gravity field explicitly links together the components of the acceleration vector (namely $a_x = \ddot{x}$, $a_y = \ddot{y}$, and $a_z = \ddot{z}$, in Cartesian coordinates). To augment the cost function of equation 5.11 with the typical mean-squared error loss, we would have the cost function given by equation 5.12

$$J(\Theta) = \frac{1}{N_f} \sum_{i=1}^{N_f} |\hat{U}(x_i|\Theta) - \hat{U}(x_i|\Theta)|^2 \tag{5.12}$$

This cost function requires knowledge of the potential function U beforehand. Currently, there is no sensor to directly measure the potential (U). However, it is possible to estimate U from the acceleration using the Gradient Theorem (Eq. 5.13).

$$\int_{\vec{r}_0}^{\vec{r}_1} \vec{a}(\vec{r}) \cdot d\vec{r} = \int_{\vec{r}_0}^{\vec{r}_1} \nabla U(\vec{r}) \cdot d\vec{r} = U(\vec{r}_1) - U(\vec{r}_0) \tag{5.13}$$

Since the spacecraft navigation system estimates $\vec{a}(\vec{r})$ over the trajectory, it is reasonable that the navigation system could estimate $U(\vec{r}(t))$. Fortunately, since the acceleration is the deliverable to the planning system, the reference point $U(\vec{r}_0)$ can be set arbitrarily. Mathematically, we estimate $U(\vec{r}(t))$ via equation 5.14.

$$U(\vec{r}(t)) = C + U(\vec{r}_0) + \int_{\vec{r}_0}^{\vec{r}(t)} \vec{a}(\vec{r}) \cdot d\vec{r} \tag{5.14}$$

The PINN learns $\hat{U}$ to approximate U, as given in equation 5.14. When the PINN evaluates the acceleration via $\vec{a} = -\nabla U$ (Eq. 5.10), the constant offsets (C and $U(\vec{r}_0)$) vanish.

The last difference between the frameworks is continuity. The Gaussian Process Model (as implemented) produces a smooth function. However, the Neural Network and Physics-Informed Neural Networks are not guaranteed to produce smooth functions. We tackle the continuity condition head-on with Lipschitz continuity and spectral normalization.

Lipschitz continuity applies to higher dimensional spaces and reduces to epsilon-delta continuity for real-valued functions. Namely, for two inputs $\vec{a}$ and $\vec{b}$, the difference in the output $f(\vec{a})$ and $f(\vec{b})$ is bounded as per equation 5.15.

$$||f(\vec{a}) - f(\vec{b})|| \leq K_L||\vec{a} - \vec{b}|| \tag{5.15}$$

The minimum constant K_L to satisfy the inequality is the Lipschitz constant (Cosgrove, 2018). If the learned dynamics can satisfy the Lipschitz continuity condition, a numerical integrator (e.g., a Runge-Kutta scheme as employed in the physics engine per Figure 3.1) can more quickly and accurately propagate the dynamics forward. Moreover, Lipschitz continuity is critical to ensuring safety guarantees in the trajectory planning and model predictive control phases.

A typical Neural Network will not meet the Lipschitz continuity condition in general. However, recent work across machine learning applications has introduced spectral normalization (Cosgrove, 2018). This technique controls the weights of a given layer during the training process to ensure that the Neural Network and, equivalently, the Physics-Informed Neural Network meet the Lipschitz continuity condition.

Chapter 6

GAUSSIAN PROCESS MODEL FRAMEWORKS

Well suited to handle noise and uncertainty, Gaussian Process Model frameworks are promising solutions for learned dynamics within stochastic optimal control. In building a new method for trajectory-only learned dynamics, the Gaussian Process Model will take data from a noisy sampling of a past trajectory and learn a gravity model. The uncertainty in the spacecraft state is often larger than the uncertainty in the acceleration (as in (Bhaskaran et al., 2011; Lorenz et al., 2017)). Hence, we introduce noise on the state as $a = a(x + \epsilon)$, where a is the acceleration, x is the position, and ϵ is Gaussian error (Figure 6.1). We interrogate how well the Gaussian Process Model can predict the acceleration a.

6.1 Base Orbits

As detailed in Appendix C, the learning frameworks (including the Gaussian Process Model frameworks) train on a dimensionless differential equation. We select the random initial point for the integrator from a user-specified Keplerian orbital element range (the initial point is based on the Keplerian osculating orbit, but the orbit itself is not Keplerian due to the higher-order terms). For our primary batch of base orbits, we select $\tilde{a} \in [1.5, 2.0]$ for the semi-major axis, $e \in [0.2, 0.8]$ for the eccentricity, and the full defined range for the inclination, the longitude of the ascending node, the argument of periapsis, and the true anomaly. The orbital element ranges are selected such that the orbit is close enough to the body that the higher-order terms in the expansion (e.g., J_4 and J_5) play a noticeable role (See Chapter 4 and specifically Figure 4.2). Moreover, a higher eccentricity allows the spacecraft to probe the gravity model closer to the surface and better match expected operations. Four orbits are chosen for training and an additional orbit is chosen for testing (Fig. 6.2). The orbits are propagated for four Keplerian periods and each orbit has roughly ten thousand data points each in $\mathbb{R}^{10}$ (for time $\mathbb{R}$, position $\mathbb{R}^3$, velocity $\mathbb{R}^3$, and acceleration $\mathbb{R}^3$).

6.2 Gaussian Process Model Training and Evaluation

The model trains on the aforementioned data as $a = a(x + \epsilon)$ (Section 5.2). As is common for Gaussian Process Models, we use an exact marginal log-likelihood

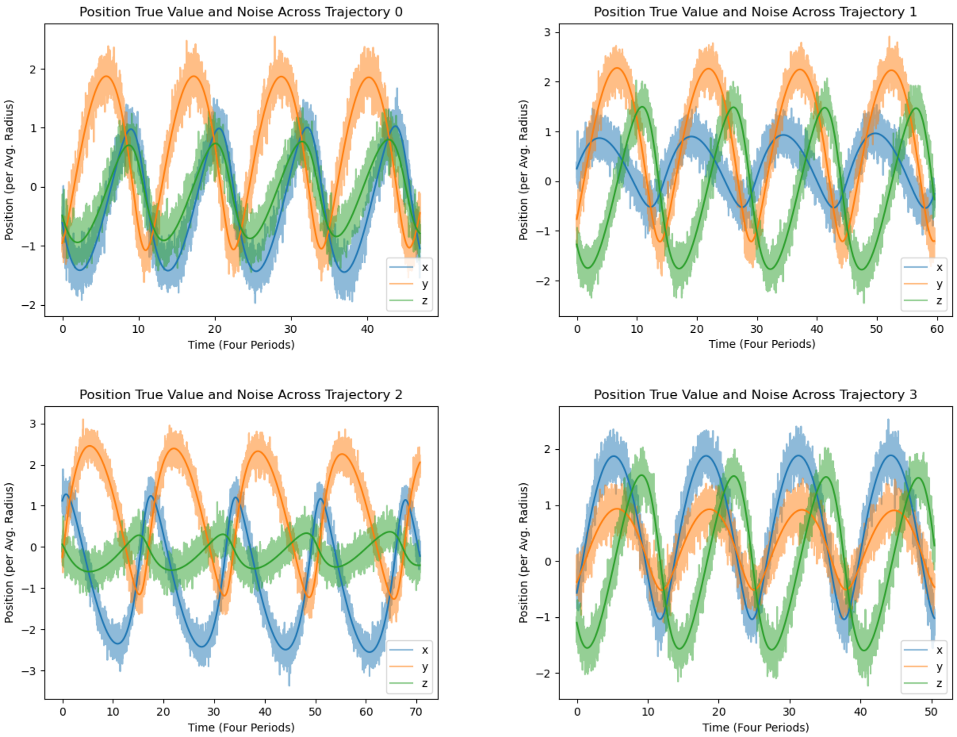


Figure 6.1: A comparison of true position trajectory (solid line) and noisy training position trajectory (translucent line). Each trajectory is split into Cartesian coordinates x, y, and z, and displayed for all four periods. Note that the true trajectories are displayed in three dimensions in Figure 6.2.

loss function and a radial basis function kernel (Rasmussen and C. K. I. Williams, 2006). For all of the training orbits, the loss quickly falls before rising and settling in 100 epochs (Fig. 6.3). Oftentimes, when numerical issues arise the loss increases. However, the loss increase is not sufficient to note a numerical issue. Namely, the relationship is one-directional.

To evaluate the quality of learned dynamics most directly, the acceleration prediction is compared to the true acceleration along the orbit. More precisely, we take the true trajectory (without noise) and feed the position at each discretized step through the learned dynamics. We then compare the output acceleration from the learned model to the true acceleration at each point along the orbit.

To visualize the acceleration change over the course of the trajectory, the acceleration vectors for the true accelerations and predicted accelerations are plotted in three-dimensional space (Fig. 6.4 and 6.5). Namely, the Gaussian Process Model is functioning as $\mathbb{R}^3 \to \mathbb{R}^3$, but the trajectory maps time ($\mathbb{R}$) to position ($\mathbb{R}^3$). Hence,

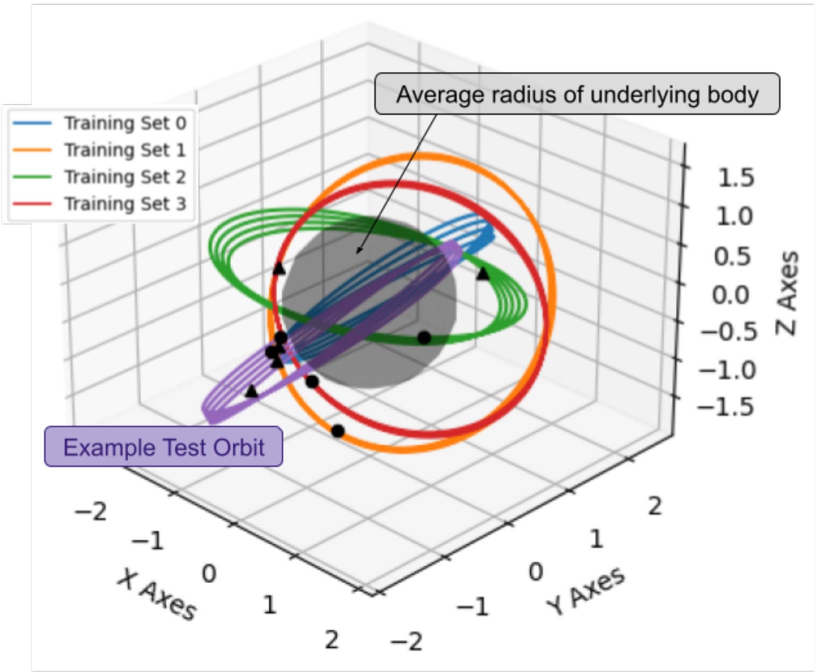


Figure 6.2: Three-dimensional plot of the training (blue, orange, green, and red) and testing (purple) trajectories in dimensionless position space. The trajectories are propagated for four Keplerian periods. The average radius of the underlying body is shown in a translucent black. None of the orbits intersect this sphere of average radius. The black circles mark the initial positions of the trajectories. The black triangles mark the final positions of the trajectories. Due to the non-uniform gravity field, the orbits change shape. As such, the initial and final positions do not overlap perfectly.

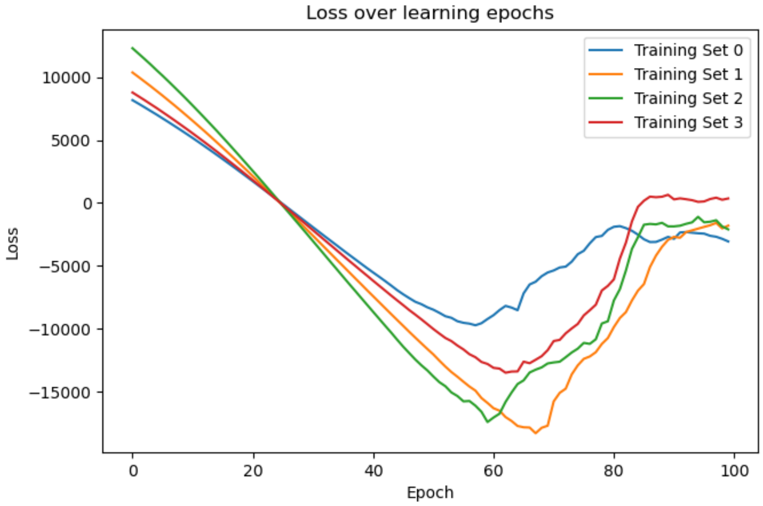
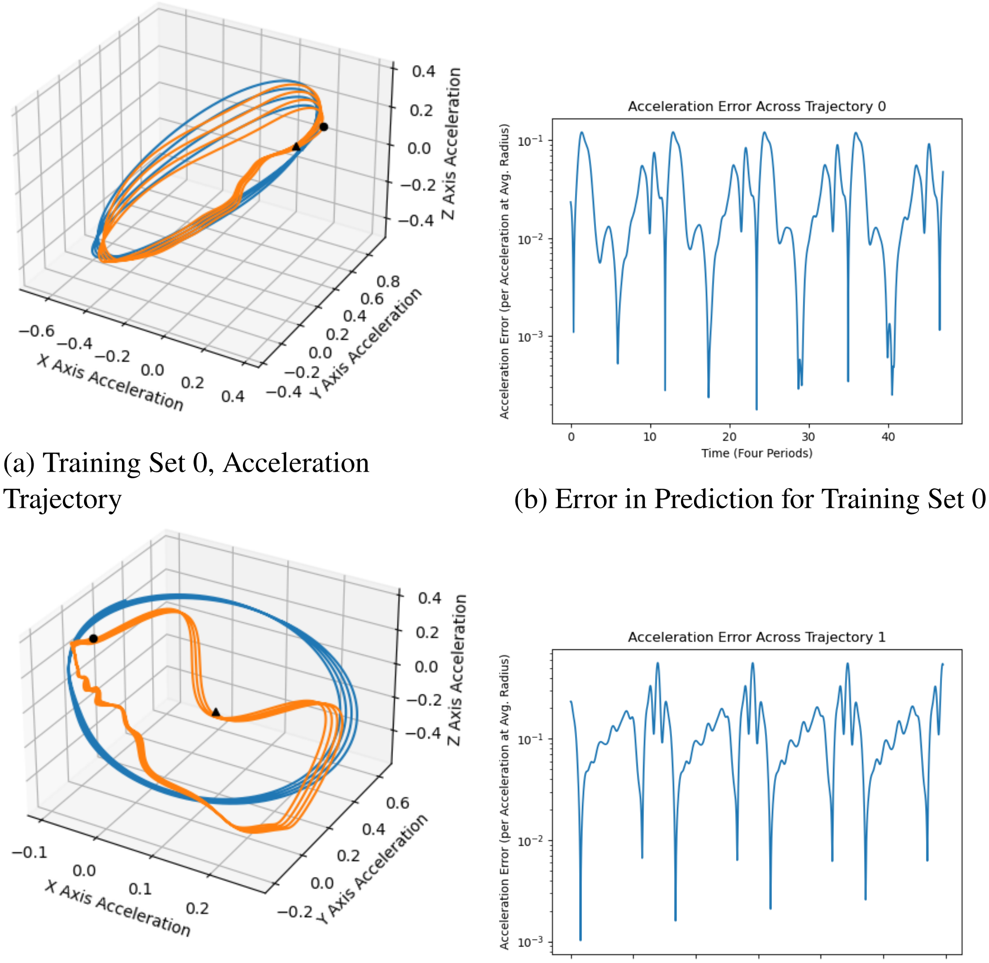


Figure 6.3: Loss curve over the training process. The colors of the curves correspond to the orbit colors in Figure 6.2.

in the acceleration trajectory plots, we are parametrically plotting $\vec{a}(\vec{x}(t)) : \mathbb{R} \to \mathbb{R}^3$ rather than $\vec{a}(\vec{x}) : \mathbb{R}^3 \to \mathbb{R}^3$ since the latter has too many dimensions for plotting. Since these learned dynamics are determined in dimensionless units (see Appendix C), the plots are generated in dimensionless units. To convert back to SI units, we scale all lengths by R^2/μ (hence the shape is identical). The error between the true and predicted acceleration is shown in Figures 6.4 and 6.5. This corresponds to test point A in the physics engine (see Chapter 3, specifically Figure 3.1).



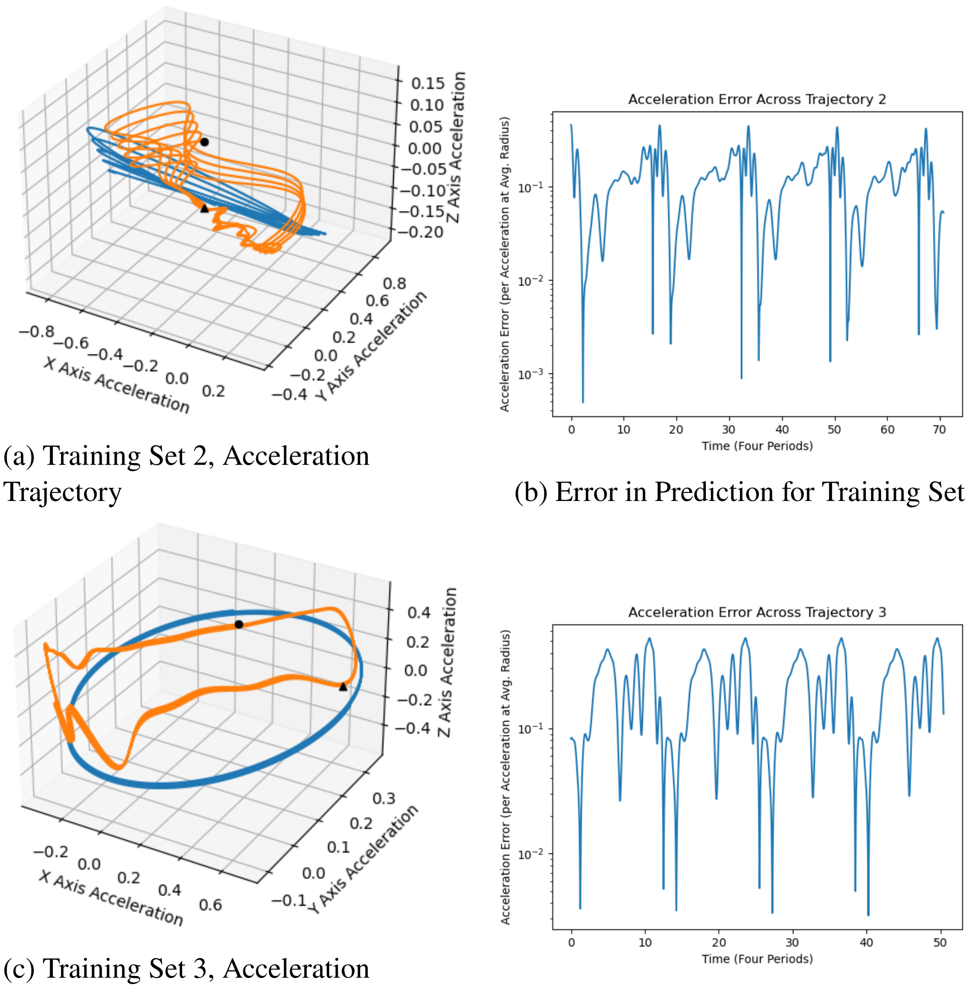
(a) Training Set 0, Acceleration Trajectory

(b) Error in Prediction for Training Set 0

(c) Training Set 1, Acceleration Trajectory

(d) Error in Prediction for Training Set 1

Figure 6.4: Acceleration trajectories and respective errors for the first two of the four training orbits closer to the body. For (a) and (c), the blue curve is the true acceleration along the trajectory and the orange curve is the predicted acceleration. Subfigures (b) and (d) correspond to the error at each timestep calculated as a pointwise l^2-Norm between the curves in (a) and (c), respectively. The black circles mark the initial positions of the trajectories. The black triangles mark the final positions of the trajectories. Matching with Figure 6.2, (a) and (b) correspond to training set 0 (the blue orbit in Figure 6.2). Likewise, (c) and (d) correspond to training set 1 (the orange curve in Figure 6.2).



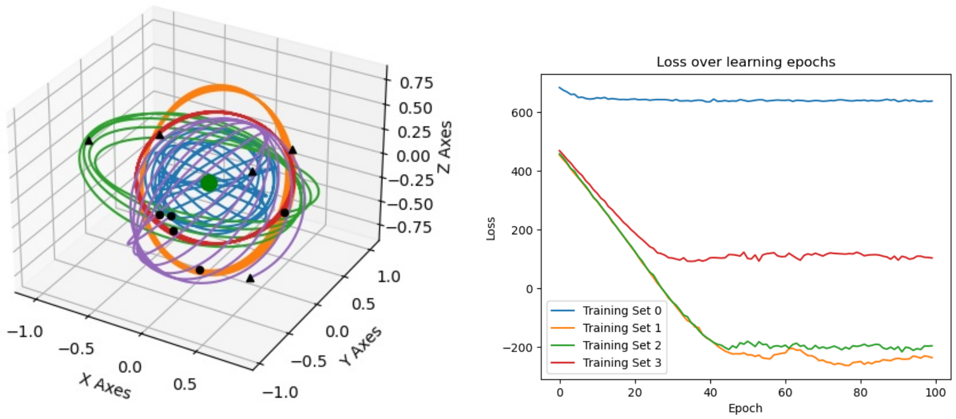
(a) Training Set 2, Acceleration Trajectory

(b) Error in Prediction for Training Set 2

(c) Training Set 3, Acceleration Trajectory

(d) Error in Prediction for Training Set 3

Figure 6.5: Acceleration trajectories and respective errors for the last two of the four training orbits closer to the body. For (a) and (c), the blue curve is the true acceleration along the trajectory and the orange curve is the predicted acceleration. Subfigures (b) and (d) correspond to the error at each timestep calculated as a pointwise l^2−Norm between the curves in (a) and (c), respectively. The black circles mark the initial positions of the trajectories. The black triangles mark the final positions of the trajectories. Matching with Figure 6.2, (a) and (b) correspond to training set 2 (the green orbit in Figure 6.2). Likewise, (c) and (d) correspond to training set 3 (the red curve in Figure 6.2).



(a) Base Orbits

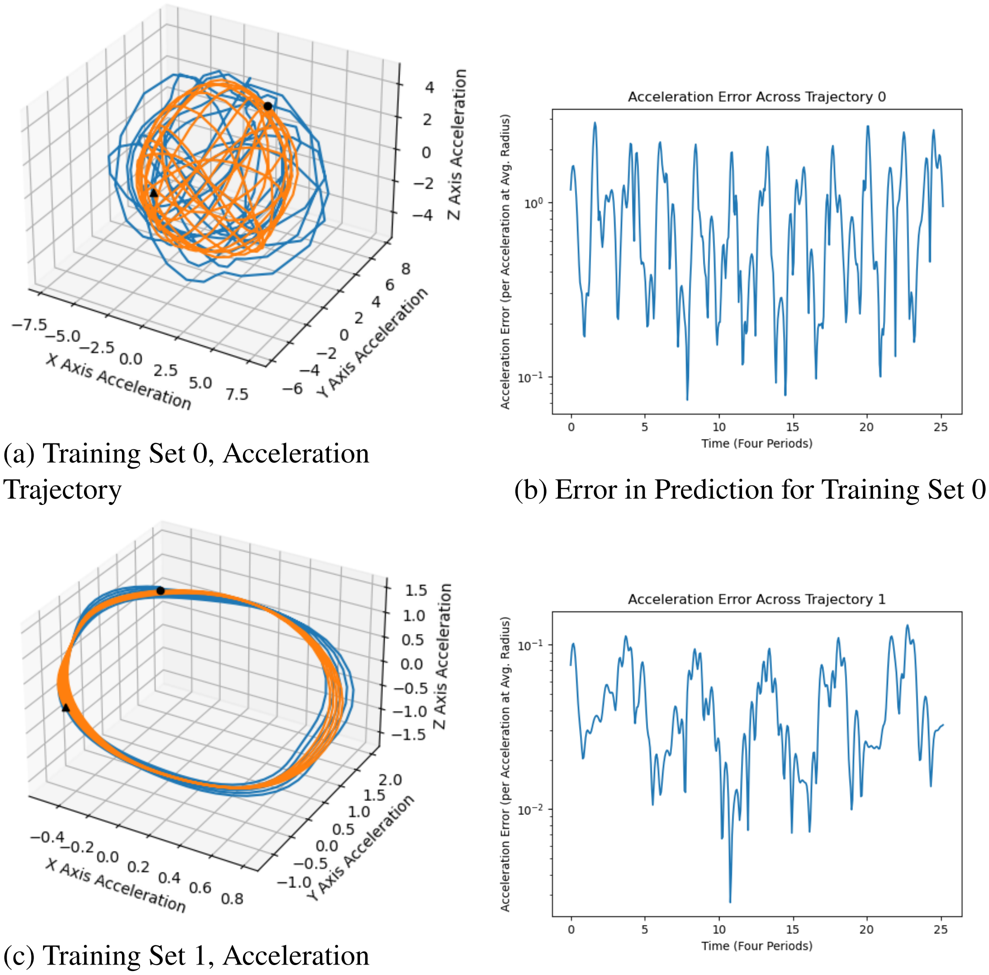
(b) Loss

Figure 6.6: The base orbits (a) include four training orbits (blue, orange, green, and red) and one test orbit (purple). The black circles mark the initial positions of the trajectories. The black triangles mark the final positions of the trajectories. The green dot represents the origin. The loss (b) drops and settles. The colors of the orbits match the colors in the loss plot. The blue orbit does not converge after 100 epochs. Notice that the blue orbit is highly perturbed, making it difficult for the Gaussian Process Model to understand the dynamics and generalize.

Comparison and Contrast to training with Less Data, Less Noise, and Closer to the Body

The pipeline is repeated with less data, less noise, and closer to the small body. With less data, the model trains and evaluates more quickly. Closer to the small body, the orbits are more perturbed. Mirroring the analysis above, new training orbits are generated and trained. In this case, all of the orbits are propagated for the same time length yielding four thousand data points (each in $\mathbb{R}^3$). The loss now drops significantly and settles without rising except for the most perturbed orbit (Fig. 6.6).

Following the same evaluation as before, the acceleration trajectory displays the comparison between the true acceleration and the predicted acceleration along the trajectory (Fig. 6.7 and 6.8). This again corresponds to test point A in the physics engine (see Chapter 3, specifically Figure 3.1). Notice that the predictions below are notably better than those shown in Figures 6.4 and 6.5. This can be predominantly attributed to the reduced noise and better convergence within 100 epochs. This comparison is discussed in Table 6.1, where we include additional combinations of data volume and noise levels.



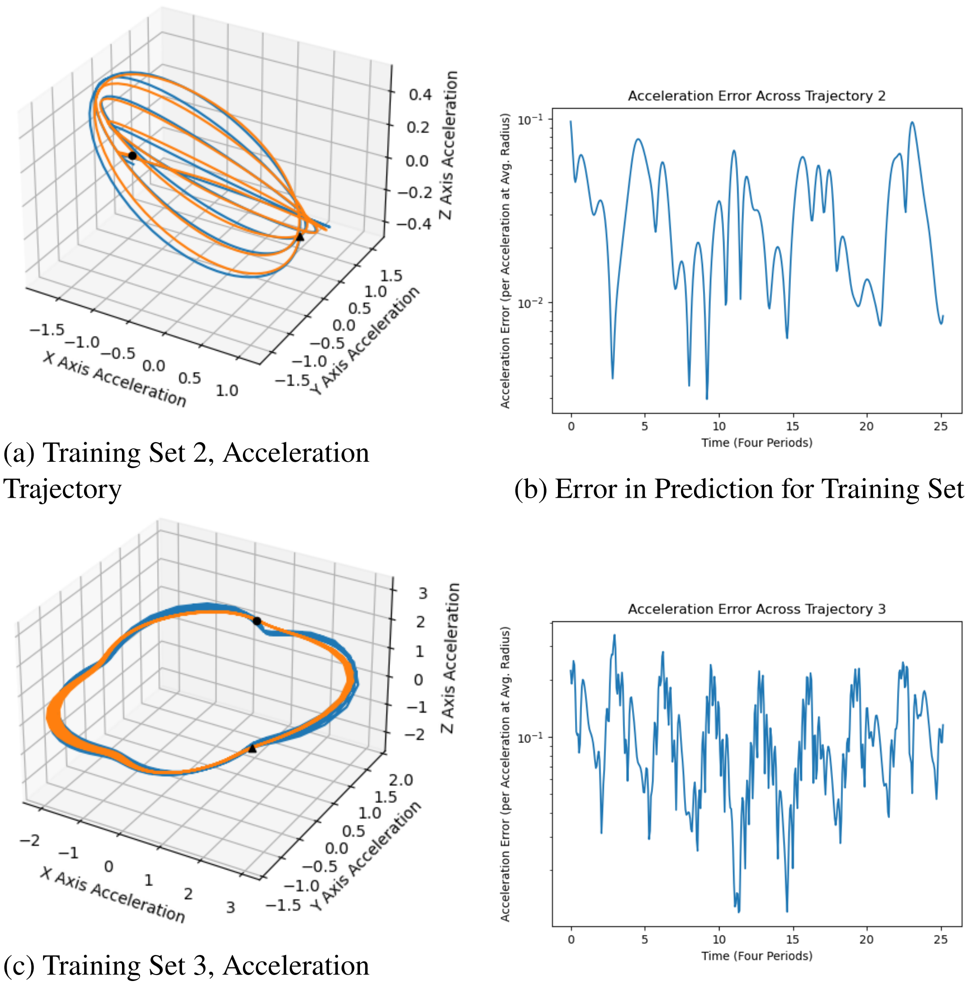
(a) Training Set 0, Acceleration Trajectory

(b) Error in Prediction for Training Set 0

(c) Training Set 1, Acceleration Trajectory

(d) Error in Prediction for Training Set 1

Figure 6.7: Acceleration trajectories and respective errors for the first two of the four training orbits closer to the body. For (a) and (c), the blue curve is the true acceleration along the trajectory and the orange curve is the predicted acceleration. Subfigures (b) and (d) correspond to the error at each timestep calculated as a pointwise l^2-Norm between the curves in (a) and (c), respectively. The black circles mark the initial positions of the trajectories. The black triangles mark the final positions of the trajectories. Matching with Figure 6.6, (a) and (b) correspond to training set 0 (the blue orbit in Figure 6.6). Likewise, (c) and (d) correspond to training set 1 (the orange curve in Figure 6.6).



(a) Training Set 2, Acceleration Trajectory

(b) Error in Prediction for Training Set 2

(c) Training Set 3, Acceleration Trajectory

(d) Error in Prediction for Training Set 3

Figure 6.8: Acceleration trajectories and respective errors for the last two of the four training orbits closer to the body. For (a) and (c), the blue curve is the true acceleration along the trajectory and the orange curve is the predicted acceleration. Subfigures (b) and (d) correspond to the error at each timestep calculated as a pointwise l^2−Norm between the curves in (a) and (c), respectively. The black circles mark the initial positions of the trajectories. The black triangles mark the final positions of the trajectories. Matching with Figure 6.6, (a) and (b) correspond to training set 2 (the green orbit in Figure 6.6). Likewise, (c) and (d) correspond to training set 3 (the red curve in Figure 6.6).

Trajectory Propagation with Learned Dynamics

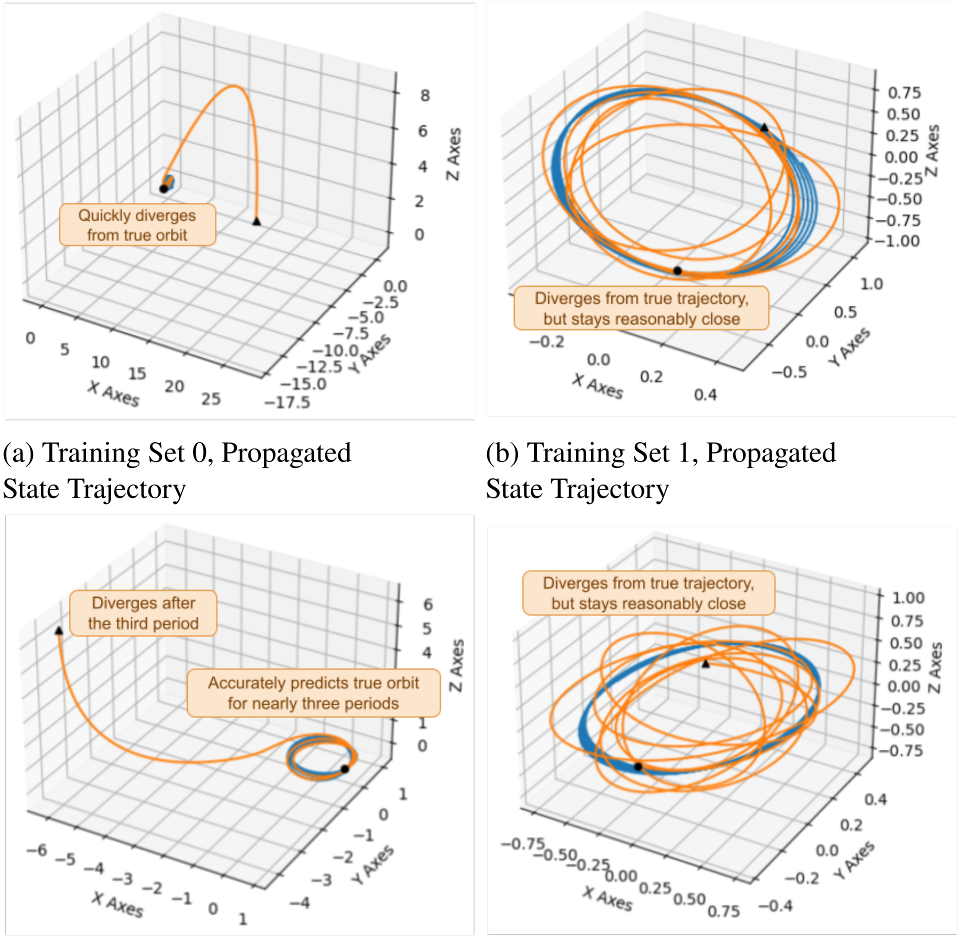
Lastly, to test out the learned dynamics in context, the trajectories (in state) are propagated forward in time from the initial condition of each respective training set. To compare against the full true trajectories shown in Figure 6.6, the dynamics are propagated for the same duration as the reference trajectories. This corresponds to test point B in the physics engine (see Chapter 3, specifically Figure 3.1).

This long horizon propagation is outside the normal planning length and represents the extreme of the use-case for the learned dynamics. In a short horizon controller, the trajectory is sufficiently close to the true trajectory to maintain orbit and glean reasonable predictions. Recalling back to the zonal harmonic dominance within the region of validity (see Chapter 4), the point mass term should dominate even close to the body. While the Gaussian Process Model is fairly accurate in reflecting the point mass term near the training data (i.e., rough Keplerian shapes), the model completely loses this insight outside of the training data domain. Without a mechanism to regularize this insight, the model quickly diverges.

Optimal Gaussian Process Model Parameters

Table 6.1 describes the synthesized trade-offs after testing the GP dynamics under multiple settings. Above, the first setting is the “Moderate Noise" for “Large Data Volume" and yields inaccurate predictions. The second setting is the “Low Noise" for “Moderate Data Volume" and yields fairly good predictions.

Note that one can truncate or sample the input trajectory data to adjust the data volume during flight. On the other hand, the noise level is not adjustable. Instead, the fidelity of the navigation system sets the noise level. Sensor fusion from onboard accelerometers and terrain relative navigation techniques may provide low noise estimates of the state and acceleration.



Quickly diverges from true orbit

Diverges from true trajectory, but stays reasonably close

Diverges after the third period

Accurately predicts true orbit for nearly three periods

Diverges from true trajectory, but stays reasonably close

(a) Training Set 0, Propagated State Trajectory

(b) Training Set 1, Propagated State Trajectory

(c) Training Set 2, Propagated State Trajectory

(d) Training Set 3, Propagated State Trajectory

Figure 6.9: Propagated state trajectory showing the position along the trajectory in $\mathbb{R}^3$. The blue curve is the true position along the trajectory and the orange curve is the predicted position. The black circles mark the initial positions of the trajectories where both trajectories begin. The black triangles mark the final positions of the trajectories. Matching with Figure 6.6, the top left subfigure (a) corresponds to training set 0 (blue curve in Figure 6.6). Likewise, the top right subfigure (b) is training set 1 (orange curve), the bottom left subfigure (c) is training set 2 (green curve), and the bottom right subfigure (d) is training set 3 (red curve).

Table 6.1: Trade-off table for Gaussian Process Model settings

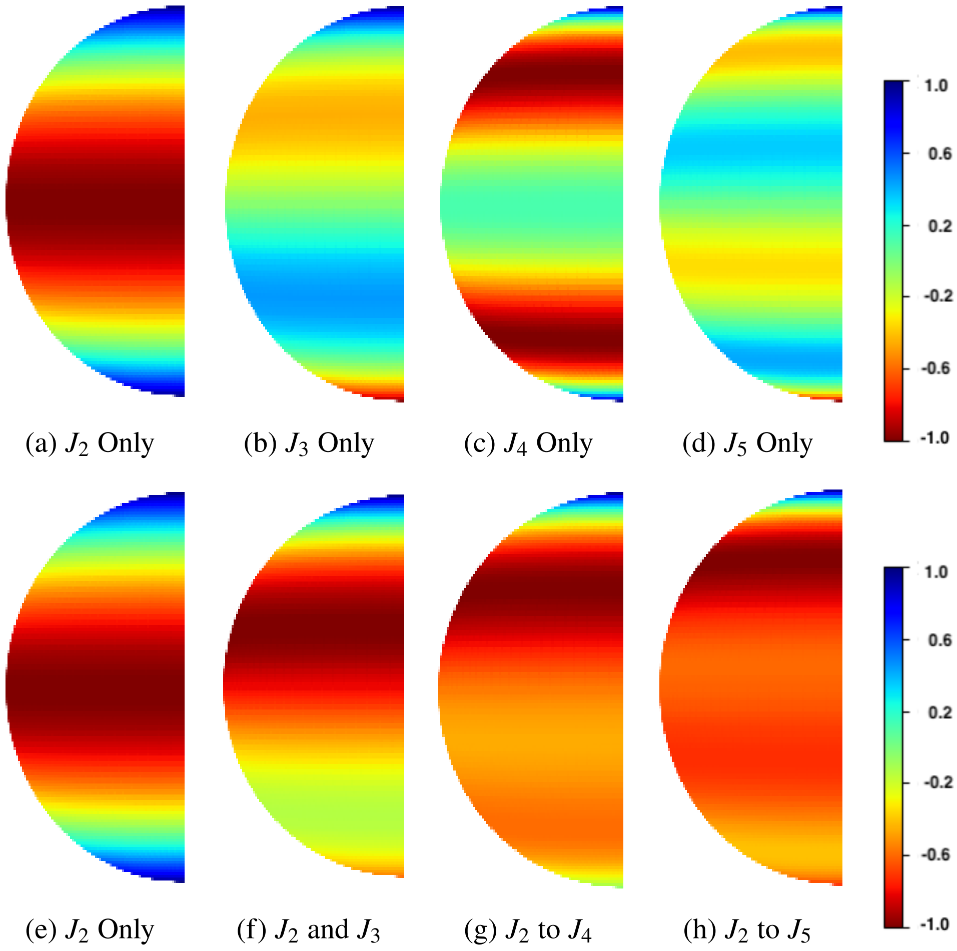
Noise	Low Data Volume	Moderate Data Volume	Large Data Volume
	(~ 10^2 data points)	(~ 10^3 data points)	(~ 10^4 data points)
Low Noise	Modest Predictions Very low computation time	Good Predictions Fair computation time	Numerical limitations
Moderate Noise	Inaccurate Predictions Very low computation time	Reasonable Predictions Fair computation time	Inaccurate Predictions High computation time

Comparison of Gaussian Process Model performance across data volume (i.e., the number of data points in the training data) and noise level (i.e., the noise added to the true trajectory training data. The green box (moderate data volumes at low noise) corresponds to the best trade-off given the results in this thesis. Such a case is likely best suited to short horizon control rather than long horizon planning. Yellow boxes correspond to settings that are okay, but not as good. Orange boxes correspond to settings that are not recommended. The red box corresponds to limitations such as computational load that would be detrimental to spacecraft operations. A deeper look at this trade-off is left to future work and would more precisely define the precision and computation time contours.

6.3 Comparison of Gaussian Process Model Across Different Harmonics

Next, we test the robustness of the Gaussian Process Model to different terms in the zonal harmonics. If the Gaussian Process Model was brittle to higher order terms, then we would need much higher term expansions in future work. On the other hand, if the Gaussian Process Model was adaptable to higher order terms, then a short expansion would be sufficient to test the learning performance. To do this, we repeat the analysis above (moderate data volumes and low noise) with different gravity models.

In the first experiment, consider a body where the point mass term and a single spherical harmonic (e.g., only the point mass term and the J_3 term) perfectly represent the body's gravity field (Fig. 6.10). As before, the GP must learn the dynamics



(a) J_2 Only

(b) J_3 Only

(c) J_4 Only

(d) J_5 Only

(e) J_2 Only

(f) J_2 and J_3

(g) J_2 to J_4

(h) J_2 to J_5

Figure 6.10: Side view of normalized J_n term anomalies as quarter spheres for the first test case in the top row (a-d) and second test case in the bottom row (e-h). For each subfigure, the values are rescaled between -1 and 1 to illustrate the most contrast (see shared legend in the top right). Values near -1 (deep red) correspond to a negative anomaly: low-density regions or lack of material. Values near 1 (deep blue) correspond to a positive anomaly: high-density regions or excess material. The first row is a quarter sphere version of Figure 3.2. To construct the second row, we add plots from the top row from left to right. For example, subfigure (g) expresses J_2 to J_4; so we sum figures (a), (b), and (c).

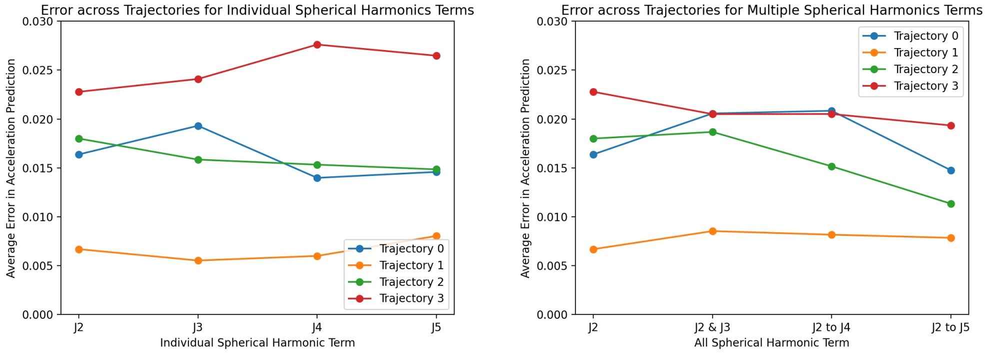
and predict the accelerations. Across the first four zonal spherical harmonics terms (J_2 to J_5), the resulting error is roughly the same (Fig. 6.11a). Hence, for the same coefficient value, the GP has roughly equivalent success at approximating zonal spherical harmonic terms.

In the second experiment, consider a body where the point mass term and a small set of spherical harmonic perfectly represent the body's gravity field. For example, the first gravity model sums the point mass and J_2 terms. The second case sums the point mass, J_2, and J_3 terms. The third and fourth cases follow this pattern to include J_4 and J_5. Every added term has the same coefficient value (sign and magnitude). Again, the GP must learn the dynamics and predict the accelerations. Across the four gravity models, the resulting error is roughly the same or slightly decreases when more terms are added (Fig. 6.11b). Hence, for the same coefficient value, the GP has roughly equivalent success at approximating each of the four gravity models. Since the additional terms are all of the same sign and magnitude, they smooth out the equatorial region and have a large difference between the equator and pole. Since trajectory 2 is more equatorial, this smoothening in the latter cases is advantageous. In contrast, trajectory 1 is more polar, so the model with the point mass term and the first four zonal spherical harmonics terms (J_2 through J_5) becomes more difficult to learn.

Combining the insight from both experiments, we conclude that the base gravity model is representative of other gravity models, especially zonal spherical harmonics models.

6.4 Remaining Gap after Gaussian Process Model Trade-Offs

An ideal learning framework would improve for lower noise and more data points (towards the top right of Table 6.1). Meanwhile, this framework would train and evaluate quickly. While the Gaussian Process Model framework improves for lower noise (up to a limit where numerical difficulties arise when no noise is present), it cannot support large data volumes (though this finding goes against (Gao and Liao, 2019)). As the data volumes increase, the training and evaluation time becomes prohibitively long. For example, the large data volume case can take the greater part of a day to train and evaluate the dynamics. The moderate data volume case takes several minutes (about 10 minutes) to train and evaluate the dynamics. While the moderate data volume case is more reasonable, it is not sufficient for on-board situations. Lastly, for long-term planning, the model must generalize to unexplored



(a) Varying gravity model based on individual spherical harmonics terms

(b) Varying gravity model with increasing number of spherical harmonics terms

Figure 6.11: Error across trajectories for the two experiments with different gravity models. In the first experiment (a), the GP learns four gravity models based on the point mass term and a single additional spherical harmonic term. In the second experiment (b), the GP learns four gravity models based on the point mass term and an increasing number of spherical harmonics. In both experiments, the variation between different trajectories (i.e., different initial conditions) is larger than the variation across gravity models. Moreover, the error is not normally distributed. Some regions of a trajectory have much higher error than others in a “patchy" sense rather than a unimodal spread (for example, see Figures 6.4, 6.5, 6.7, and 6.8).

states. However, the Gaussian Process Model fails to generalize well beyond the input trajectory.

Therefore, the Gaussian Process Model is promising for noisy or highly uncertain situations and short-term planning. However, both the lack of generalization and the long computation time notably hinder the Gaussian Process Model performance for long-term planning, especially on-board the spacecraft.

Chapter 7

NEURAL NETWORK FRAMEWORKS

The primary motivation for the Neural Network is to ensure fast evaluation times after the network has trained on large data volumes. While the Gaussian Process Model evaluation time is dependent on the number of data points, the Neural Network evaluation time is constant with respect to the number of data points. Underneath the hood, the Gaussian Process Model uses the data matrix directly in the evaluation step. On the other hand, the Neural Network simply computes the weights and the activation functions (namely the rectified linear unit, ReLU) during the evaluation. The sizes of the weights are fixed beforehand in the Neural Network architecture.

7.1 Neural Network Training and Evaluation

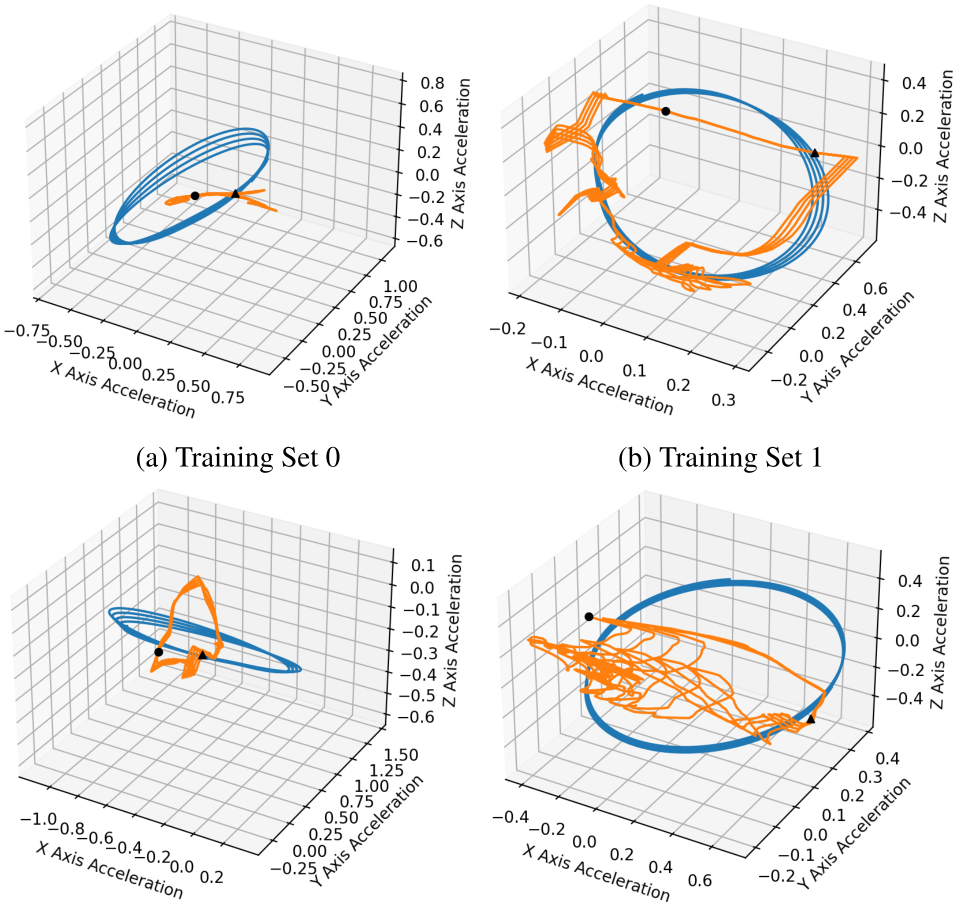
To more clearly visualize the impact of noise and spectral normalization, we train the model under three test cases (Table 7.1). For all of the test cases, we use the same orbits as Figure 6.2.

The first test case is approximating noisy data without spectral normalization (Figure 7.1). The predicted accelerations do not accurately match the true accelerations. Moreover, the predicted accelerations are not continuous (i.e., a small change in

Table 7.1: Neural Network training cases to illustrate the underlying limitations and the impact of spectral normalization

Test Case	Noisy Data	Spectral Normalization
1	Yes (Noisy)	No (No spectral normalization)
2	Yes (Noisy)	Yes (Spectral normalization included)
3	No (No Noise)	Yes (Spectral normalization included)

We have three test cases where we “switch" the noisy data and spectral normalization on or off. Test case two would be the ideal case where we handle noise and continuity simultaneously. Test case one serves as a comparison without the spectral normalization. Test case three serves as a comparison without the noise. The last combination (no noise and no spectral normalization) is not included since omitting spectral normalization would drop the Lipschitz continuity guarantee without any noticeable improvements.



(a) Training Set 0

(b) Training Set 1

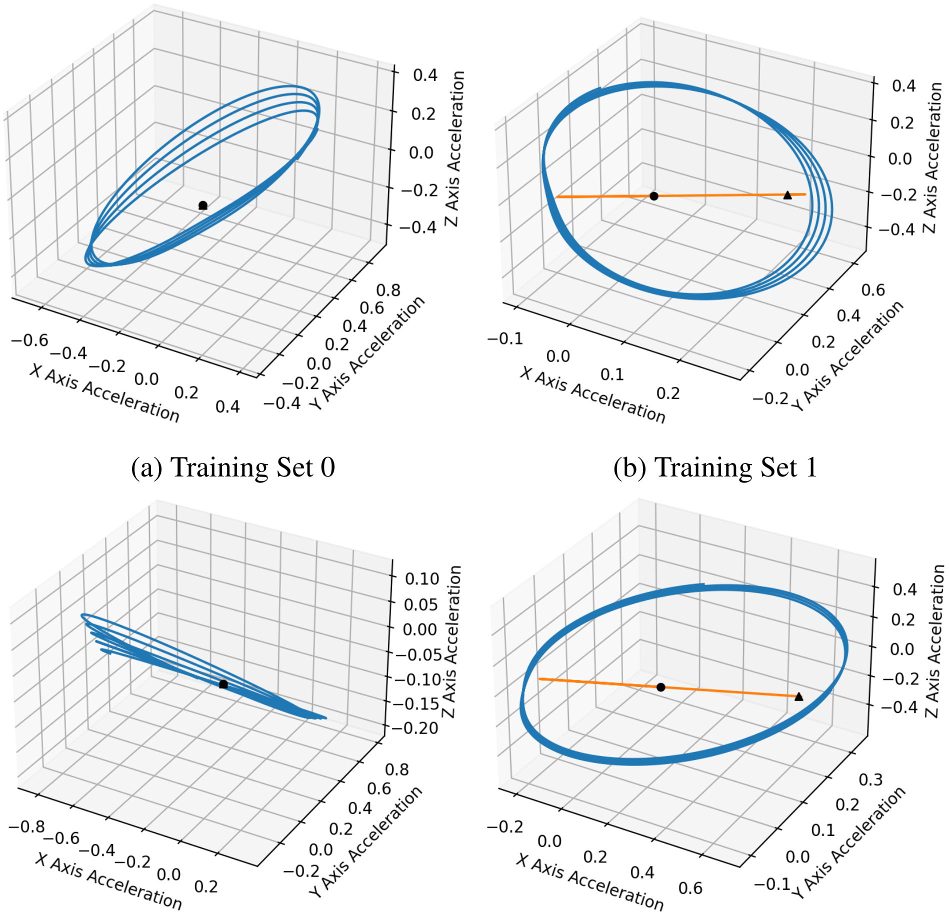
(c) Training Set 2

(d) Training Set 3

Figure 7.1: Test Case 1: Neural Network training for noisy data without the spectral normalization. The Neural Network predicted accelerations (orange) do not accurately approximate the true accelerations (blue). Since the spectral normalization is not included, the output does not maintain Lipschitz continuity.

input position $\vec{x}$ yields a large change in acceleration). This matches the theoretical note that a Neural Network output is not guaranteed to be Lipschitz continuous.

The second test case is approximating noisy data with spectral normalization (Figure 7.2). The predicted accelerations are worse in this test case. The Neural Network almost entirely predicts zero acceleration. This matches the theoretical note that spectral normalization is forcing a Lipschitz continuous function. As an example consider an input position $\vec{x}_1$. The true dynamics would have $\vec{a}(\vec{x}_1)$, which is continuous. But, with noise, the training set has $\vec{a}(\vec{x}_1) + \vec{\epsilon}_1$, where $\vec{\epsilon}_1$ is the error. Then, a small change in position (i.e., $\vec{x}_2 = \vec{x}_1 + \delta$) yields $\vec{a}(\vec{x}_2) + \vec{\epsilon}_2 \not\approx \vec{a}(\vec{x}_1 + \delta) + \vec{\epsilon}_1$. That is, we lose continuity since the noise is jagged.



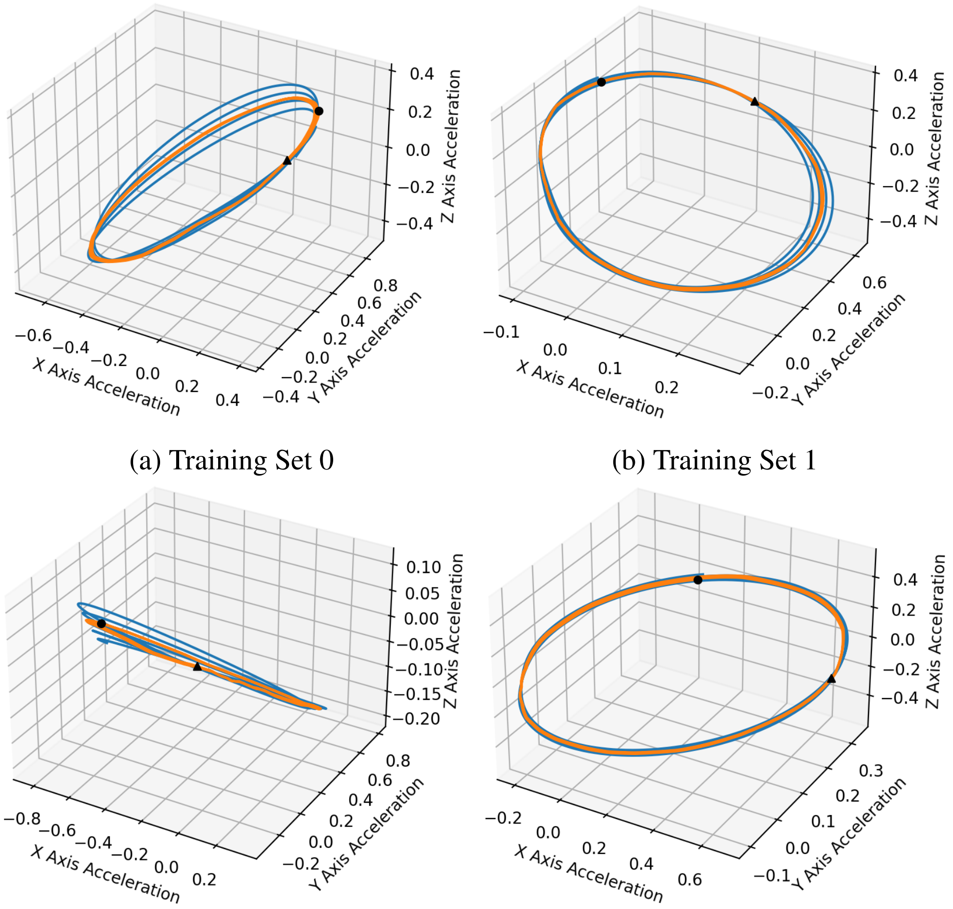
(a) Training Set 0

(b) Training Set 1

(c) Training Set 2

(d) Training Set 3

Figure 7.2: Test Case 2: Neural Network training for noisy data with the spectral normalization. Since the noise is large (e.g., Figure 6.1), the inputs and outputs themselves do not maintain Lipschitz continuity. Nevertheless, spectral normalization tries to maintain Lipschitz continuity. The Neural Network thereby predicts accelerations (shown in orange) that are either uniformly zero (training set 0 and 2) or fall on a line (training set 1 and 3). The predicted accelerations (orange) do not accurately approximate the true accelerations (blue). The key takeaway is that the inputs need to be sufficiently smooth (e.g., filtered or more precise) for the spectral normalization technique to make sense.



(a) Training Set 0

(b) Training Set 1

(c) Training Set 2

(d) Training Set 3

Figure 7.3: Test Case 3: Neural Network training for filtered data with the spectral normalization. With the reduced noise, the Neural Network provides a fairly good approximation (orange) of the true accelerations (blue). Even without the Physics-Informed learning (next section), the Neural Network with spectral normalization is able to approximate the dynamics about as good as the best test cases of the Gaussian Process Model. As with the Gaussian Process Model, the Neural Network does not capture the full dynamics.

The third test case is approximating filter data with spectral normalization (Figure 7.3). The filtered data has no noise and is smooth. Here, we get a much better result. The Neural Network produces a continuous function (as the spectral normalization enforces) that moderately matches the true data.

7.2 Comparison and Contrast between the Neural Network and Gaussian Process Models

We can provide a contrast between the first set of the Gaussian Process Models (i.e., Figures 6.4 and 6.5) and the Neural Networks (Figures 7.1-7.3). Namely, we used the same orbits for training the first set of Gaussian Process Models as the Neural Networks. However, we must remember two caveats: (1) the Gaussian Process Model provides useful covariance data for stochastic optimal control that the Neural Network is not built to provide, and (2) the first set of Gaussian Process Models trained on noisy data whereas only the first two test cases of the Neural Network trained on noisy data.

With this in mind, the Gaussian Process Model significantly outperforms the Neural Network in cases where there is noise. Namely, in Figures 6.4 and 6.5, the Gaussian Process Model's predicted acceleration does not always match the true acceleration, and the error is fairly large. However, the first two cases of the Neural Networks produced highly unfavorable results. In both of these cases, deploying such as Neural Network on a spacecraft would be disastrous since the results would be inaccurate and the Neural Network would fail to learn the underlying dynamics even with more data. Therefore, the Gaussian Process Model is the preferred option for situations with large noise.

If we enter a regime where the noise is very low, the conclusion flips: the Neural Network mostly outperforms the Gaussian Process Model. In Figure 7.3, the Neural Network's predicted accelerations nicely fall within the range of true accelerations and are comfortably continuous (test 3). As such, the Neural Network produces a more accurate model. However, the Neural Network in test 3 preferentially produces acceleration trajectories that do not capture the full variability in the dynamics. This is most clearly observed in subfigure (a) of Figure 7.3. In contrast, the Gaussian Process Model *does* seem to capture more of the variability, while not as accurate in general (see subfigure (a) of Figure 6.4 and subfigure (a) of Figure 6.5).

The error statistics from the Neural Network framework is much higher than that predicted in (Martin and Schaub, 2020; Cheng et al., 2020; Song, Cheng, and Gong, 2019). To a large extent, we can attribute the poorer performance to a difference in methods. Effectively, prior works formulate the problem as an interpolation problem whereas we have created an extrapolation problem. A neural network will necessarily preform better in the interpolation problem since it has optimized its approximation on the training domain. Please see Chapter 9 for more details.

To a lesser extent, we can attribute the poorer performance to a difference in data volumes. Specifically, prior work uses 10^2 to 10^4 times as many data points. In this analysis, we kept the moderate data volumes to ensure a fair comparison with the Gaussian Process Models. By switching the approach and including higher data volumes, we believe that the Neural Network will be able to achieve much more accurate results.

Chapter 8

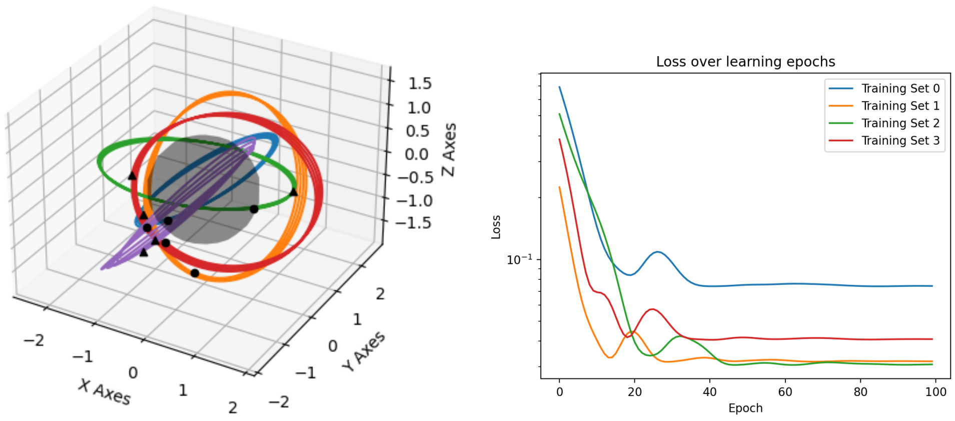
PHYSICS-INFORMED NEURAL NETWORK FRAMEWORKS

Following the work in (Martin and Schaub, 2020), the Physics-Informed Neural Network framework builds in the conservative nature of gravitational field directly into the learning framework. From the Gaussian Process Model section, we observed how the predicted propagated trajectories diverge due to the inability for the model to generalize the point mass term dominance outside of the training data. Since the conservative property is built into the Physics-Informed Neural Network, we can help regularize the framework output to combat the divergence observed with the Gaussian Process Model. From the Neural Network section, we demonstrated how the low noise case with spectral normalization can provide smooth and fairly accurate results. Since a Physics-Informed Neural Network is fundamentally a subset of the larger family of Neural Networks, Physics-Informed Neural Network can also benefit from lower noise, large data volumes, and spectral normalization.

8.1 Physics-Informed Neural Network Training and Evaluation

Physics-Informed Neural Networks similar to those in (Martin and Schaub, 2020) focus on learning the potential U rather than the acceleration $\vec{a}$. Hence, to evaluate the Physics-Informed Neural Network, we critically need to evaluate the fit of $U(\vec{x})$. Whereas before we had acceleration trajectory plots (since the learning framework output was $\vec{a} : \mathbb{R}^3 \to \mathbb{R}^3$), we now have a one-dimensional plot (since the learning framework output is $U : \mathbb{R}^3 \to \mathbb{R}$). First, the base orbit and loss are shown in Figure 8.1. The loss seems to converge more quickly than the Gaussian Process Model and Neural Network cases. The comparison of the true potential to the predicted acceleration along with the associated error is shown in Figure 8.2. Note that the Physics-Informed Neural Network also includes spectral normalization and we assume filtered data (i.e., no noise). In this sense, test 3 of the Neural Network is the most similar framework.

Unfortunately, the Physics-Informed Neural Network (as implemented) does regress the potential function well enough to produce meaningful accelerations. Namely, when we calculate the gradients with autograd in pytorch using $\vec{a} = -\nabla U$ (as in Eq. 5.10), we then arrive at the acceleration plots in Figure 8.3 where the predicted accelerations do not closely match the true accelerations.



(a) Base Orbits

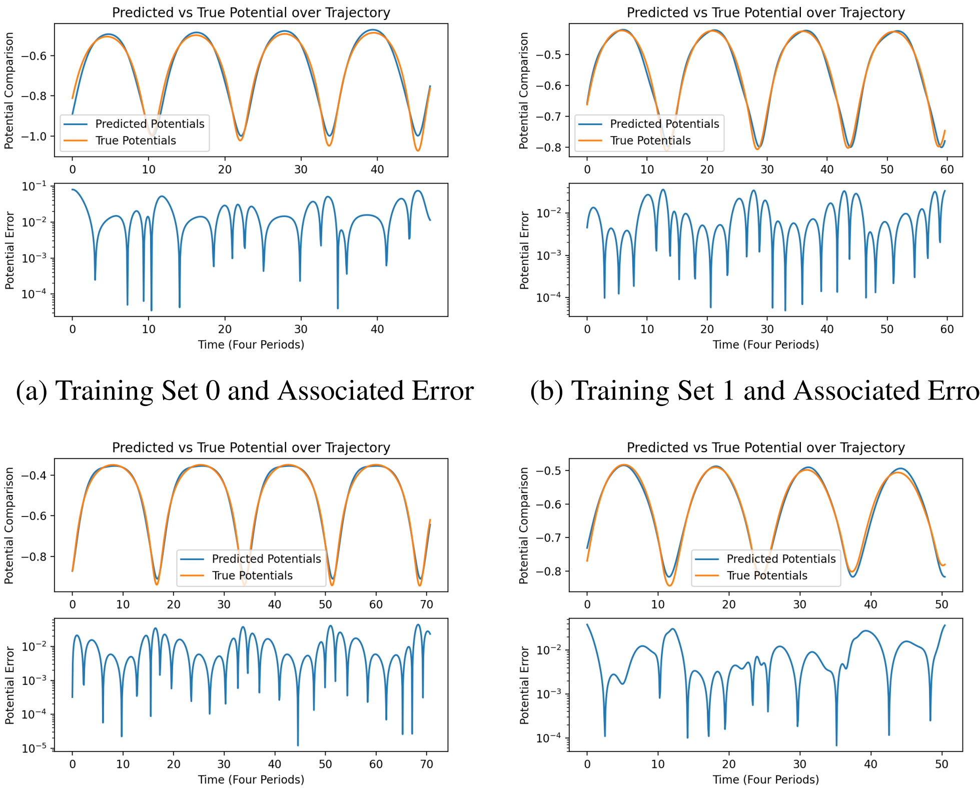
(b) Loss

Figure 8.1: Similar to Figure 6.3, the base orbits (a) include four training orbits (blue, orange, green, and red) and one test orbit (purple). The black circles mark the initial positions of the trajectories where both trajectories begin. The black triangles mark the final positions of the trajectories. The translucent black sphere dot represents the small body and is centered on the origin. The loss (b) drops and settles. The colors of the orbits match the colors in the loss plot. The green and orange trajectories converge to a similar loss. The red trajectory converges to a loss that is close to the green and orange trajectory, but not as low. The blue orbit is closest to the body. While it seems to converge on its own, the loss value is higher than the other orbits.

8.2 Comparison and Contrast between the Physics-Informed Neural Network and Other Learning Frameworks

Overall, the Gaussian Process Model and the Neural Networks outperform the Physics-Informed Neural Network. In particular, the Physics-Informed Neural Network's predicted accelerations (Figure 8.3) do much more poorly than the most similar method: test case 3 of the Neural Network (Figure 7.3). It also does more poorly than the Gaussian Process Model (Figures 6.4 and 6.5). An interesting comparison note is that the hyperbolic tangent function (i.e., tanh) produced smoother accelerations than the rectified linear unit (i.e., ReLU) in the Physics-Informed Neural Network. We did not observe this disparity between activation functions trend in the Neural Network case. Note that all the Neural Network plots (Figures 7.1-7.3) show results with a rectified linear unit activation function.

With a look at all three frameworks (Gaussian Process Model, Neural Network, and Physics-Informed Neural Network), we can determine which framework is best for which application (assuming a Trajectory-Only approach). If we are in a situation



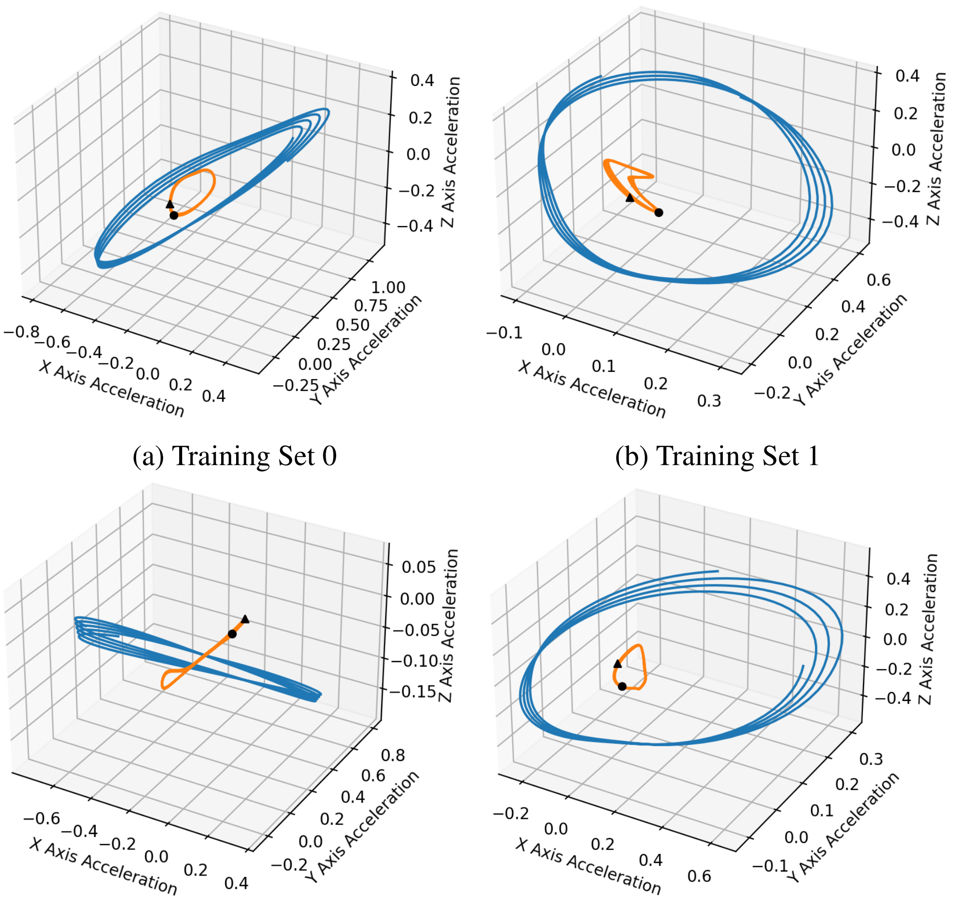
(a) Training Set 0 and Associated Error

(b) Training Set 1 and Associated Error

(c) Training Set 2 and Associated Error

(d) Training Set 3 and Associated Error

Figure 8.2: Potential trajectories and respective errors for all four training orbits closer to the body. For the upper plots in each subfigure, the blue curve is the true acceleration along the trajectory and the orange curve is the predicted acceleration. The lower plots in each subfigure are the error at each timestep calculated as a pointwise l^2−Norm between the curves in the upper part. Matching with Figure 8.1, subfigure (a) corresponds to training set 0 (the blue orbit in Figure 8.1). Likewise, subfigure (b) corresponds to training set 1 (the orange orbit in Figure 8.1), subfigure (c) corresponds to training set 2 (the green orbit), and subfigure (d) corresponds to training set 3 (the red orbit).



(a) Training Set 0

(b) Training Set 1

(c) Training Set 2

(d) Training Set 3

Figure 8.3: Acceleration trajectories from Physics-Informed Neural Network learning framework. The black circles mark the initial positions of the trajectories where both trajectories begin. The black triangles mark the final positions of the trajectories. The numbering matches Figures 8.1 and 8.2.

with high uncertainty or high noise, then the Gaussian Process Model is the best of the three frameworks. It natively handles the error and can provide error estimates for the stochastic optimal controller. If we are able to smooth the data or reduce the noise, then the Neural Network framework becomes the best option. This could be especially useful for a longer term trajectory planner or for using many trajectories in the learning step. Lastly, the Physics-Informed Neural Network is promising based on (Martin and Schaub, 2020), but it does not work well with the Trajectory-Only strategy from this thesis. Specifically, the gradient step is quite sensitive to the switch from an extrapolation problem (as in this thesis) to an interpolation problem (as in (Martin and Schaub, 2020)). Please see chapter 9 for more information. All-in-all, the Physics-Informed Neural Network has not shown great results as implemented in this thesis, but it is promising from a theoretical standpoint.

Chapter 9

RECOMMENDATIONS FOR FUTURE LEARNED GRAVITY MODELS RESEARCH

We believe that we can learn from the lessons in this thesis to produce a more fruitful problem formulation in the future. In the current trajectory-only set-up, the learning framework is fundamentally faced with an extrapolation problem (Figure 9.1). In this thesis, we developed a trajectory-only perspective on the training data. In particular, for all the cases, we use four periods worth of data (hence, we have four intersections onto the plane or four blue points in Figure 9.1 left panel). The trajectories only sample a very narrow region of positions. Consider if we probe a new position near the trajectories or even the same input position, but without the noise (shown as the red X).The point is not actually within the training domain. So, we have to extrapolate to determine the acceleration at the new point (i.e., the red X is not in the light blue band). Without more structure to the problem, we should not expect the learning framework to generalize in this way. With this logic, we have two options (1) reformulate the problem as an interpolation or (2) embed more structure into the Neural Network.

To reformulate the problem as an interpolation, we would more closely follow work from (Gao and Liao, 2019; Song, Cheng, and Gong, 2019; Cheng et al., 2020; Martin and Schaub, 2020). Namely, we start with a base model such as a constant density polyhedron model that we would generate on approach via stereophotoclinometry. This is shown as the middle panel in Figure 9.1 where the green points denote samplings of the offline constant density polyhedron model. If we probe a new position (i.e., the red X), the new point is well within the training domain (i.e., the red X is within the light green region). These points may be biased, but the learning problem is now an interpolation problem. We would then use a learning framework to more efficiently represent the model such as a Gaussian Process Model as in (Gao and Liao, 2019), a Neural Network as in (Song, Cheng, and Gong, 2019; Cheng et al., 2020), or a Physics-Informed Neural Network as in (Martin and Schaub, 2020). Suppose we mix the novel strategy from this thesis with the dense interpolation that is used more frequently in the literature (right panel of Figure 9.1). The novelty is to incorporate the trajectory data into the data set with higher preference than the base model. We start with the offline, possible biased baseline (green points). As we orbit

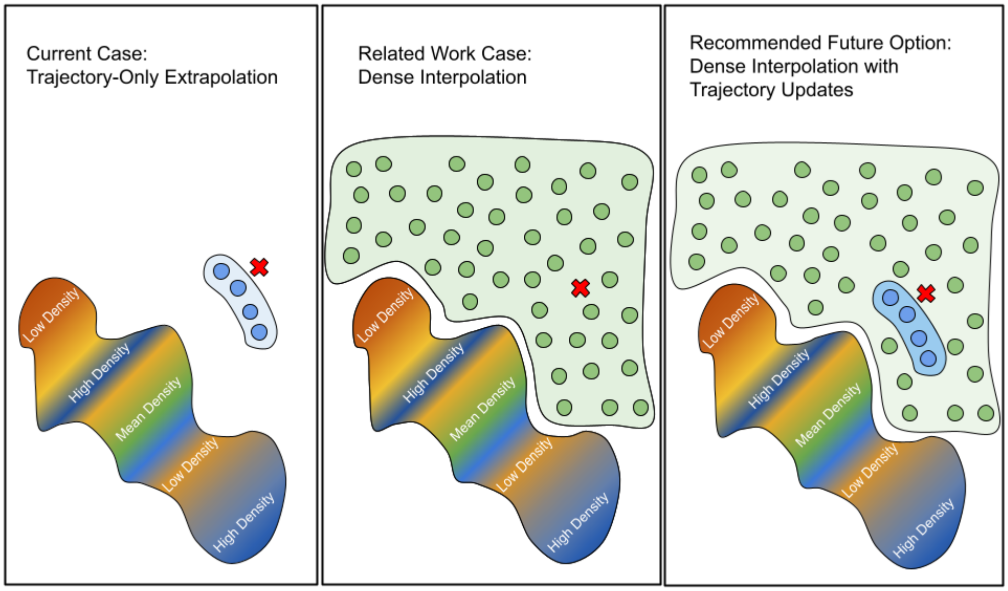


Figure 9.1: Graphical representation between the trajectory-only extrapolation case (left panel), dense interpolation case (middle panel), and recommendation for future work (right panel). For visualization, we display a two-dimensional schematic. This is conceptually equivalent to placing a dot every time a trajectory crosses a specified plane (as is done in a Poincaré Section). In all the panels, we illustrate the underlying density distribution of the small body as a color gradient where blue corresponds to high density regions and red corresponds to low density regions (as in Figure 1.3).

the small body we obtain lower uncertainty and more accurate data (blue points). We can update the model with the more accurate points to reach a compromise: (1) we have interpolation throughout and (2) the model reduces the bias with more orbits. In the former consideration, we have reformulated the trajectory-only problem as an interpolation problem since the learning framework always probes a point that is interior to the training regime (the union of green and blue shaded regions). For a trajectory with sufficient coverage, the learning framework provides an unbiased estimate of the gravity field since it is using the measured acceleration data from the perception system (recall that the constant density polyhedron is a biased estimator as noted in Section 3.2) and the perception data is given higher priority. To the author’s best knowledge, this would be a novel problem formulation.

We could also reformulate the problem with more structure. This can take many flavors. One option would be to split the problem into multiple parts. One estimator might estimate the total mass of the body (and hence the point mass term), another might estimate the moment of inertia (and hence the J_2 term), and then the final estimator would only have to estimate the higher-order effects.

A different option would be to build a data-driven adapted polyhedron model. That is, we start at a constant density polyhedron. Since most small bodies do not have a constant density, there will bias between the predictions from the constant density polyhedron and the accelerometer measurements (as noted in Section 3.2). The learning framework then has to produce the nearest constant density mesh that accurately represents the gravity field. This method is advantageous since the result is interpretable and the trained model can generalize well to different small bodies. To interpret the result, we would compare the shape polyhedron to the gravity model polyhedron. When the gravity model polyhedron is interior to the shape polyhedron, we have a negativity gravitational anomaly (i.e., the material is less dense than expected, so we need a smaller volume of material to achieve the measured field for the same density). When the gravity model polyhedron is exterior to the shape polyhedron, we have a positive gravitational anomaly (i.e., the material is denser than expected). Lastly, a perfect fit would mean that the small body is indeed of constant density. The trained model can also generalize well across small bodies since the learning framework is determining how best to change a mesh subject to new data as opposed to learning the gravity field of a single body. The main issue with this method is that it does not overcome the computational complexity issues of the polyhedral model.

A third option would focus on the entire trajectory propagation rather than estimating the dynamics. For example, we could build a Physics-Informed Neural Network that encodes the conservation of kinetic and potential energy rather than the concept that the gravity field is a conservative field. In such a case, we would pass both position and velocity to the Physics-Informed Neural Network and its task would be to determine the position and velocity at a later timestamp.

For each of these reformulation strategies, we can still take lessons from this thesis. First, a spacecraft integrator with a spherical harmonics gravity provides a highly versatile, computationally favorable, sufficiently realistic, and decently available physics engine back-end to develop training data (Table 3.1). Second, low noise, moderate data volume Gaussian Process Models are suitably accurate, robust, and informative gravity models for short-horizon planning (Table 6.1). Moreover, Gaussian Process Models can conveniently fit within the stochastic optimal control structure since they provide covariance information, produce smooth functions, and evaluate points within a reasonable time (for low to moderate data volumes). Third, for low noise or low uncertainty cases, Neural Networks with spectral normalization can produce accurate and continuous models (Figure 7.3). Therefore, there is significant algorithmic design space to develop new learning-based small-body gravity models that can smoothly integrate into stochastic optimal control learning frameworks.

Chapter 10

CONCLUDING REMARKS

Our overarching goal was to investigate the extent to which learning frameworks effectively approximate the gravity field dynamics with data from recent trajectories. Overall, the Gaussian Process Model provides the most attractive learning framework for cases of high uncertainty, as we would expect for small body missions (Figure 6.4-6.9). However, numerical limitations and significant computational time for large data volumes remain challenges for incorporating the Gaussian Process Model framework on spacecraft, especially at low noise. If the noise can be effectively filtered to produce a smooth signal, the Neural Network approach with spectral normalization appears promising. For the Neural Network, we can include larger datasets and maintain low evaluation times compared to the Gaussian Process Model. Although the Physics-Informed Neural Network appears theoretically promising, the problem formulation in this thesis is not able to reach the performance that one would theoretically expect.

All-in-all, we introduced a new method in trajectory-only learned dynamics that adds to our ability to conduct proximity maneuvers around small bodies. To tackle the dangers of uncertainty in the gravity dynamics, we formulated our learning-based method with the intention of combining it with stochastic optimal control (namely Information-Cost Stochastic Nonlinear Optimal Control). From the lessons learned in this thesis, we developed an inventory of ideas and recommendations to guide future research. Whether from scientific, economic, or humanitarian motives, we believe that this thesis provides a valuable contribution in reaching our goal of safe landings and missions to small bodies in our solar system.

BIBLIOGRAPHY

Amelin, Yuri et al. (Sept. 2002). "Lead Isotopic Ages of Chondrules and Calcium-Aluminum-Rich Inclusions". In: *Science* 297.5587, pp. 1678–1683. DOI: 10.1126/science.1073950. URL: https://science.sciencemag.org/content/297/5587/1678.

Basilevsky, Alexander T. and James W. Head (Apr. 1998). "The geologic history of Venus: A stratigraphic view". In: *Journal of Geophysical Research E: Planets* 103.3334, pp. 8531–8544. ISSN: 01480227. DOI: 10.1029/98je00487.

Bate, Roger, Donald D. Mueller, and Jerry E. White (1971). *Fundamentals of Astrodynamics*. 1st ed. New York: Dover Publications, Inc., pp. 1–470. ISBN: 0486600610.

Behçet, Açıkmeşe and Lars Blackmore (Feb. 2011). "Lossless convexification of a class of optimal control problems with non-convex control constraints". In: *Automatica* 47.2, pp. 341–347. ISSN: 00051098. DOI: 10.1016/j.automatica.2010.10.037.

Berry, Kevin et al. (Feb. 2013). "OSIRIS-REx Touch-And-Go (TAG) Mission Design and Analysis AAS 13-095". In: *36th Annual AAS Guidance and Control Conference*. Breckenridge, CO: AAS/AIAA.

Bhaskaran, Shyam et al. (July 2011). "Small Body Landings Using Autonomous Onboard Optical Navigation". In: *The Journal of the Astronautical Sciences* 58.3, pp. 409–427.

Blackmore, Lars, Masahiro Ono, and Brian C Williams (2011). "Chance-constrained optimal path planning with obstacles". In: *IEEE Transactions on Robotics* 27.6, pp. 1080–1094. ISSN: 15523098. DOI: 10.1109/TRO.2011.2161160.

Bottke, William F. et al. (2006). "The Yarkovsky and YORP effects: Implications for asteroid dynamics". In: *Annual Review of Earth and Planetary Sciences* 34, pp. 157–191. ISSN: 00846597. DOI: 10.1146/annurev.earth.34.031405.125154.

Caro, Guillaume (May 2011). "Early silicate earth differentiation". In: *Annual Review of Earth and Planetary Sciences* 39, pp. 31–58. ISSN: 00846597. DOI: 10.1146/annurev-earth-040610-133400.

Cavosie, Aaron J., John W. Valley, and Simon A. Wilde (2007). "The Oldest Terrestrial Mineral Record: A Review of 4400 to 4000 Ma Detrital Zircons from Jack Hills, Western Australia". In: *Developments in Precambrian Geology*. Vol. 15. Elsevier. Chap. 2.5, pp. 91–111. ISBN: 9780444528100. DOI: 10.1016/S0166-2635(07)15025-8.

Cheng, Lin et al. (May 2020). “Real-Time Optimal Control for Irregular Asteroid Landings Using Deep Neural Networks”. In: *Acta Astronautica* 170, pp. 66–79. DOI: 10.1016/j.actaastro.2019.11.039. URL: http://arxiv.org/abs/1901.02210%20http://dx.doi.org/10.1016/j.actaastro.2019.11.039.

Cosgrove, Christian (Jan. 2018). *Spectral Normalization Explained*. URL: https://christiancosgrove.com/blog/2018/01/04/spectral-normalization-explained.html.

Davis, Don et al. (2021). *Planetary Data System (PDS) Asteroid/Dust Subnode*. Tucson, AZ. URL: https://sbn.psi.edu/pds/.

ESA (Dec. 2015). *The Rosetta lander*. URL: http://www.esa.int/Science_Exploration/Space_Science/Rosetta/The_Rosetta_lander.

Gao, Ai and Wentao Liao (Apr. 2019). “Efficient gravity field modeling method for small bodies based on Gaussian process regression”. In: *Acta Astronautica* 157, pp. 73–91. ISSN: 00945765. DOI: 10.1016/j.actaastro.2018.12.020.

Hockman, Benjamin and Marco Pavone (Dec. 2017). “Stochastic Motion Planning for Hopping Rovers on Small Solar System Bodies”. In: *Robotics Research*. Ed. by Nancy M. Amato et al. 18th ISRR Symposium. Vol. 10. Puerto Varas, Chile: Springer International Publishing, pp. 877–893. DOI: 10.1007/978-3-030-28619-4{_}60.

Hockman, Benjamin, Robert G Reid, et al. (Oct. 2017). “Experimental Methods for Mobility and Surface Operations of Microgravity Robots”. In: *2016 International Symposium on Experimental Robotics*. Ed. by Dana Kulić et al. 1st ed. Tokyo, Japan: Springer, Cham, pp. 752–763. DOI: 10.1007/978-3-319-50115-4{_}65. URL: http://youtu.be/bDmoqjNQAu8.

Hockman, Benjamin J. et al. (Jan. 2017). “Design, Control, and Experimentation of Internally-Actuated Rovers for the Exploration of Low-gravity Planetary Bodies”. In: *Journal of Field Robotics* 34.1, pp. 5–24. ISSN: 15564967. DOI: 10.1002/rob.21656.

Jones, Brandon Allan (2010). “Efficient Models for the Evaluation and Estimation of the Gravity Field”. PhD thesis. University of Colorado, Boulder. URL: http://scholar.colorado.edu/asen_gradetds.

Kargel, Jeffrey S (Oct. 1994). “Metalliferous asteroids metals as potential sources of precious”. In: *Journal of Geophysical Research* 99.10, pp. 129–141.

Kattenhorn, Simon A. and Louise M. Prockter (Oct. 2014). “Evidence for subduction in the ice shell of Europa”. In: *Nature Geoscience* 7.10, pp. 762–767. ISSN: 17520908. DOI: 10.1038/NGEO2245.

Lo, Martin W (Mar. 2002). “The InterPlanetary Superhighway and the Origins Program”. In: *IEEE Aerospace Conference*. Big Sky, MT: IEEE, pp. 7–3542. URL: 10.1109/AERO.2002.1035332.

Lorenz, David A. et al. (June 2017). "Lessons Learned from OSIRIS-REx Autonomous Navigation Using Natural Feature Tracking". In: *IEEE Aerospace Conference Proceedings*. Big Sky, MT: IEEE. ISBN: 9781509016136. DOI: 10.1109/AERO.2017.7943684.

Lotzof, Kerry (Oct. 2020). *Your mobile phone is powered by precious metals and minerals*. London. URL: https://www.nhm.ac.uk/discover/your-mobile-phone-is-powered-by-precious-metals-and-minerals.html.

Marchi, Simone et al. (2013). "Global resurfacing of Mercury 4.0-4.1 billion years ago by heavy bombardment and volcanism". In: *Nature* 499.7456, pp. 59–61. ISSN: 00280836. DOI: 10.1038/nature12280.

Martin, John R. and Hanspeter Schaub (Feb. 2020). "Applications of Physics-informed Neural Networks for Gravity Field Modeling AAS 21-393". In: *AAS/AIAA Space Flight Mechanics Meeting*. Charlotte, NC: AAS/AIAA.

Myers, John S (Jan. 2001). "Protoliths of the 3.8-3.7 Ga Isua greenstone belt, West Greenland". In: *Precambrian Research* 105.2-4, pp. 129–141. URL: www.elsevier.com/locate/precamres.

Nakka, Yashwanth Kumar and Soon Jo Chung (2019). "Trajectory Optimization for Chance-Constrained Nonlinear Stochastic Systems". In: *Proceedings of the IEEE Conference on Decision and Control*. Vol. 2019-Decem, pp. 3811–3818. ISBN: 9781728113982. DOI: 10.1109/CDC40024.2019.9028893.

Nakka, Yashwanth Kumar, Wolfgang Hönig, et al. (2021). "Information-Based Guidance and Control Architecture for Multi-Spacecraft On-Orbit Inspection". In: DOI: 10.2514/6.2021-1103. URL: http://arc.aiaa.org.

Nakka, Yashwanth Kumar, Anqi Liu, et al. (2020). "Chance-Constrained Trajectory Optimization for Safe Exploration and Learning of Nonlinear Systems". In: URL: http://arxiv.org/abs/2005.04374.

Oguri, Kenshiro and Jay W McMahon (Feb. 2020). "Autonomous Guidance for Robust Achievement of Science Observations around Small Bodies". In: *AAS Guidance, Navigation, and Control Conference*. Breckenridge, CO. URL: https://www.researchgate.net/publication/339137418.

Park, Ryan S, Robert A Werner, and Shyam Bhaskaran (Jan. 2010). "Estimating Small-Body Gravity Field from Shape Model and Navigation Data". In: *Journal of Guidance, Control, and Dynamics* 33.1.

Patterson, Claire (Oct. 1956). "Age of meteorites and the earth". In: *Geochimica et Cosmochimica Acta* 10.4, pp. 230–237.

Peng, Hao and Xiaoli Bai (Apr. 2021). "Fusion of a machine learning approach and classical orbit predictions". In: *Acta Astronautica*. ISSN: 00945765. DOI: 10.1016/j.actaastro.2021.04.017. URL: https://www.sciencedirect.com/science/article/pii/S0094576521001636.

Rapp, Donald (2018). “Refueling Spacecraft in LEO Using Propellants Derived from the Moon or Asteroids”. In: *Use of Extraterrestrial Resources for Human Space Missions to Moon or Mars*. Cham: Springer International Publishing, pp. 163–176. ISBN: 978-3-319-72694-6. DOI: `10.1007/978-3-319-72694-6{\_}4`. URL: `https://doi.org/10.1007/978-3-319-72694-6_4`.

Rasmussen, Carl E. and Christopher K I Williams (2006). *Gaussian Processes for Machine Learning*. Ed. by Thomas Diettrich et al. Cambridge, MA: MIT Press. ISBN: 026218253X. URL: `www.GaussianProcess.org/gpml`.

Scheeres, D J, J W Mcmahon, et al. (2019). “Comparing the Estimated Dynamical Environments and Mass Distributions of Bennu and Ryugu”. In: *Lunar and Planetary Science Conference: Asteroid Science*.

Scheeres, D J, JW McMahon, et al. (2019). “The Gravity and Global Geophysical Environment of (101955) Bennu”. In: *50th Lunar and Planetary Science Conference 2019 (LPI Contrib. No. 2132)*.

Scheeres, D. J. et al. (Oct. 2020). “Heterogeneous mass distribution of the rubble-pile asteroid (101955) Bennu”. In: *Science advances* 6.41, eabc3350. ISSN: 23752548. DOI: `10.1126/sciadv.abc3350`. URL: `http://advances.sciencemag.org/`.

Scheeres, Daniel J (June 2016). *Orbital Motion in Strongly Perturbed Environments: Applications to Asteroid, Comet and Planetary Satellite Orbiters*. 3rd ed. Germany: Springer Berlin Heidelberg.

Scheeres, Daniel J. and Francesco Marzari (Jan. 2003). “Spacecraft dynamics in the vicinity of a comet”. In: *Journal of the Astronautical Sciences* 50.1, pp. 35–52. ISSN: 00219142. DOI: `10.1007/bf03546329`.

Schenk, Paul M. and Kevin Zahnle (Dec. 2007). “On the negligible surface age of Triton”. In: *Icarus* 192.1, pp. 135–149. ISSN: 00191035. DOI: `10.1016/j.icarus.2007.07.004`.

Song, Yu, Lin Cheng, and Shengping Gong (Jan. 2019). “Fast Estimation of Gravitational Field of Irregular Asteroids Based on Deep Neural Network and Its Application”. In: *29th AAS/AIAA Space Flight*. Ka’anapali, Maui, HI: AAS/AIAA. URL: `https://www.researchgate.net/publication/331523846`.

Takahashi, Yu (2013). “Gravity Field Characterization around Small Bodies”. PhD thesis. Boulder: University of Colorado at Boulder. URL: `https://scholar.colorado.edu/concern/graduate_thesis_or_dissertations/fq977v06t`.

Takahashi, Yu and D. J. Scheeres (May 2014). “Morphology driven density distribution estimation for small bodies”. In: *Icarus* 233, pp. 179–193. ISSN: 00191035. DOI: `10.1016/j.icarus.2014.02.004`.

Tricia Talbert (Feb. 2018). *Five Years after the Chelyabinsk Meteor: NASA Leads Efforts in Planetary Defense*. Washington DC. URL: www.nasa.gov/feature/five-years-after-the-chelyabinsk-meteor-nasa-leads-efforts-in-planetary-defense.

Walsh, Kevin J (July 2018). "Rubble Pile Asteroids". In: *Annual Review of Astronomy and Astrophysics* 56, pp. 593–624. DOI: 10.1146/annurev-astro-081817. URL: https://doi.org/10.1146/annurev-astro-081817-.

Weiss, Benjamin P. and Linda T. Elkins-Tanton (May 2013). "Differentiated planetesimals and the parent bodies of chondrites". In: *Annual Review of Earth and Planetary Sciences* 41, pp. 529–560. ISSN: 00846597. DOI: 10.1146/annurev-earth-040610-133520.

Werner, Robert A. and Daniel J. Scheeres (1997). "Exterior gravitation of a polyhedron derived and compared with harmonic and mascon gravitation representations of asteroid 4769 Castalia". In: *Celestial Mechanics and Dynamical Astronomy* 65.3, pp. 313–344. ISSN: 09232958. DOI: 10.1007/bf00053511.

Williams, B. et al. (June 2018). "OSIRIS-REx Flight Dynamics and Navigation Design". In: *Space Science Reviews* 214.69. ISSN: 15729672. DOI: 10.1007/s11214-018-0501-x.

Yu, Yang (2016). "Orbital Dynamics in the Gravitational Field of Small Bodies". PhD thesis. Beijing, China: Tsinghua University. URL: http://www.springer.com/series/8790.

Appendix A

DERIVATION OF ACCELERATIONS FROM ZONAL SPHERICAL HARMONICS

A.1 Gravitational Potential and Preliminaries

The specific energy potential characterizes the geopotential of a body per equation A.1.

$$u = \underbrace{\frac{-\mu}{r}}_{\text{Point Mass}} + \underbrace{\sum_{n=2}^{N_z} \frac{J_n P_n^0(\sin(\theta))}{r^{n+1}}}_{\text{Zonal Terms}} + \underbrace{\sum_{n=2}^{N_t} \sum_{m=1}^{n} \frac{P_n^m(\sin(\theta))(C_n^m \cos(m\phi) + S_n^m \sin(m\phi)}{r^{n+1}}}_{\text{Tesseral Terms}} \tag{A.1}$$

The form of the non-dimensional constants are detailed below (Bate, Mueller, and White, 1971):

$$\tilde{J}_n = \frac{-J_n}{\mu R^n}, \qquad \tilde{C}_n^m = \frac{-C_n^m}{\mu R^n}, \qquad \text{and} \qquad \tilde{S}_n^m = \frac{-S_n^m}{\mu R^n}$$

Note that μ (the central body's Gravitational Parameter) has units of $\mathrm{m^3\,s^{-2}}$ and u has units of $\mathrm{m^2\,s^{-2}}$ which is a energy per mass. To recover the energy (U), $U = um_s$ can be used where m_s is the spacecraft mass. Generally, the point mass contribution is the largest and the easiest to measure. For many asteroids, this point mass contribution can be estimated from Earth. For spherical bodies or bodies that are nearly symmetric about the equator, the zonal terms are the next most prominent. Often in the literature, the zonal terms are reported and not the tesseral terms (such as in (D. J. Scheeres et al., 2020)). Investigation into the best representation of the gravity model of a small body is an active research area and is beyond the scope of this thesis (Section 3.2). Since the focus of this thesis is more on the planning and control problem, the zonal terms are sufficiently available and realistic to verify the results of this thesis.

While the potential is conveniently written in the form of equation A.1, it is not sufficient for the integrator (if a symplectic integrator was used, one might leave the equation as is). Instead, it is necessary to (1) derive the acceleration from the potential and (2) convert from spherical coordinates to Cartesian coordinates.

A.2 Acceleration from Gravitational Potential

Note that just as force is the negative gradient of the potential energy, acceleration is the negative gradient of the specific potential energy by Newton's second law. To get the acceleration, the derivative of specific potential energy is taken with respect to (r, θ, ϕ). Namely $\vec{a} = -\nabla u$ yields:

$$a_r = -\left(\frac{\partial u}{\partial r}\right) \tag{A.2}$$

$$a_\theta = \frac{-1}{r}\left(\frac{\partial u}{\partial \theta}\right) \tag{A.3}$$

$$a_\phi = \frac{-1}{r\sin\theta}\left(\frac{\partial u}{\partial \phi}\right) \tag{A.4}$$

Notice that without the tesseral terms, u is not a function of ϕ. Moreover, the point mass term simply yields the traditional equation for the gravity of a point mass

$$\vec{a} = \frac{\vec{F}_{grav}}{m_s} = \frac{\mu}{r^2}(-\hat{r}) = \frac{-\mu\vec{r}}{r^3} \tag{A.5}$$

This now leaves the zonal terms. The radial term is simple and yields

$$a_r = -[\mu\tilde{J}_n P_n^0(\sin\theta)]R^n\left(\frac{d}{dr}\left(r^{-(n+1)}\right)\right) \tag{A.6}$$

$$a_r = -[\mu\tilde{J}_n P_n^0(\sin\theta)]R^n\left(-(n+1)r^{-(n+2)}\right) \tag{A.7}$$

The more difficult derivative is in θ which includes the Legendre P polynomials. To simplify the derivations we need two recurrence relations of the Legendre P polynomials (the second is known as Bonnet's recurrence formula):

$$\frac{dP_{n+1}(x)}{dx} = (n+1)P_n(x) + x\frac{dP_n(x)}{dx} \tag{A.8}$$

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x) \tag{A.9}$$

Expanding, this is rewritten as the expression below (see proof in Appendix B)

$$\frac{dP_n(x)}{dx} = \frac{(n+1)P_{n+1}(x) - (n+1)xP_n(x)}{x^2-1} \tag{A.10}$$

Using chain rule and equation A.10 yields:

$$a_\theta = -\left(\frac{1}{r}\right)\left[\frac{\mu\tilde{J}_n}{r}\left(\frac{R}{r}\right)^n\right]\left(\frac{d}{d\theta}P_n(\sin\theta)\right) \tag{A.11}$$

$$a_\theta = -\left(\frac{1}{r}\right)\left[\frac{\mu\tilde{J}_n}{r}\left(\frac{R}{r}\right)^n\right]\left((n+1)\sec(\theta)[-P_{n+1}(\sin\theta) + (\sin\theta)P_n(\sin\theta)]\right) \tag{A.12}$$

To simplify the expressions, notice that there is a common term:

$$K = \mu \tilde{J}_n (n+1) R^n r^{-(n+2)} \tag{A.13}$$

Using K, the final results for the zonal term in spherical coordinates are

$$a_r = K\, P_n(\sin\theta) \tag{A.14}$$

$$a_\theta = K \sec(\theta)[P_{n+1}(\sin\theta) - (\sin\theta) P_n(\sin\theta)] \tag{A.15}$$

$$a_\phi = 0 \tag{A.16}$$

A.3 Cartesian Conversion

The variable transform defined below converts back to Cartesian coordinates, where the $\hat{\phi}$ contribution is implicitly assumed to be zero since the equation does not vary in the $\hat{\phi}$ direction:

$$\hat{x} = \hat{r}\cos(\theta)\cos(\phi) - \hat{\theta}\cos(\phi)\sin(\theta) \tag{A.17}$$

$$\hat{y} = \hat{r}\cos(\theta)\sin(\phi) - \hat{\theta}\sin(\phi)\sin(\theta) \tag{A.18}$$

$$\hat{z} = \hat{r}\sin(\theta) + \hat{\theta}\cos(\theta) \tag{A.19}$$

Next, based on a common coordinate system with z pointing towards the north pole, the below trigonometric identities provide the Cartesian expressions needed for the coordinate transformation.

$$\sin(\theta) = \frac{z}{|r|} \tag{A.20}$$

$$\cos(\theta) = \frac{\sqrt{x^2+y^2}}{|r|} \tag{A.21}$$

$$\tan(\theta) = \frac{z}{\sqrt{x^2+y^2}} = \frac{z\sqrt{x^2+y^2}}{|r|^2 - z^2} \tag{A.22}$$

$$\sin(\phi) = \frac{x}{\sqrt{x^2+y^2}} \tag{A.23}$$

$$\cos(\phi) = \frac{y}{\sqrt{x^2+y^2}} \tag{A.24}$$

After further algebra transforming equations A.14 - A.16 via equations A.17 - A.19

and simplifying with equations A.20 - A.24, the final expressions are

$$a_x = \frac{Kx}{|r|^2 - z^2}\left[|r|P_n\left(\frac{z}{|r|}\right) - zP_{n+1}\left(\frac{z}{|r|}\right)\right] \tag{A.25}$$

$$a_y = a_x\frac{y}{x} \tag{A.26}$$

$$a_z = K\ P_{n+1}\left(\frac{z}{|r|}\right) \tag{A.27}$$

Notice the persistent singularity in equations A.25 - A.27 at the poles (i.e., $x = y = 0$ and $r \longrightarrow z$). Either by evaluating the limit or by symmetry about ϕ, we have $a_x = 0$, $a_y = 0$. and $a_z = K$, where $P_n(1) = 1$ simplifies a_z, by definition of the Legendre polynomial.

Appendix B

LEGENDRE POLYNOMIAL RECURRENCE RELATION DERIVATION

We want to prove the following:

$$\frac{dP_n(x)}{dx} = \frac{(n+1)P_{n+1}(x) - (n+1)xP_n(x)}{x^2-1}$$

Using only the below two recurrence relations:

$$\frac{dP_{n+1}(x)}{dx} = (n+1)P_n(x) + x\frac{dP_n(x)}{dx} \tag{B.1}$$

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x) \tag{B.2}$$

Proof. First, we differentiate Bonnet's recurrence formula,

$$\frac{d}{dx}\left((n+1)P_{n+1}(x)\right) = \left((2n+1)xP_n(x) - nP_{n-1}(x)\right) \tag{B.3}$$

$$(n+1)\frac{dP_{n+1}(x)}{dx} = (2n+1)P_n(x) + (2n+1)x\frac{dP_n(x)}{dx} - n\frac{dP_{n-1}(x)}{dx} \tag{B.4}$$

Applying the first recurrence relation and simplifying the left-hand side yields

$$(n+1)\left((n+1)P_n(x) + x\frac{dP_n(x)}{dx}\right) = (n^2+2n+1)P_n(x) + (n+1)x\frac{dP_n(x)}{dx} \tag{B.5}$$

Which simplifies the whole expression from equation B.4 as

$$n^2P_n(x) - nx\frac{dP_n(x)}{dx} = -n\frac{dP_{n-1}(x)}{dx} \tag{B.6}$$

$$nP_n(x) - x\frac{dP_n(x)}{dx} = -\frac{dP_{n-1}(x)}{dx} \tag{B.7}$$

Now we write the first recurrence relation in terms of $\frac{dP_n(x)}{dx}$

$$\frac{dP_n(x)}{dx} = \frac{1}{x}\frac{dP_{n+1}(x)}{dx} - \frac{n+1}{x}P_n(x) \tag{B.8}$$

Offsetting the indices by one yields:

$$\frac{dP_{n-1}(x)}{dx} = \frac{1}{x}\frac{dP_n(x)}{dx} - \frac{n}{x}P_{n-1}(x) \tag{B.9}$$

Substituting equation B.9 back into equation B.7, yields:

$$nP_n(x) - x\frac{dP_n(x)}{dx} = -\left[\frac{1}{x}\frac{dP_n(x)}{dx} - \frac{n}{x}P_{n-1}(x)\right] \tag{B.10}$$

$$\left(x - \frac{1}{x}\right)\frac{dP_n(x)}{dx} = nP_n(x) - \frac{n}{x}P_{n-1}(x) \tag{B.11}$$

$$\left(\frac{x^2 - 1}{x}\right)\frac{dP_n(x)}{dx} = nP_n(x) - \frac{n}{x}P_{n-1}(x) \tag{B.12}$$

$$\frac{dP_n(x)}{dx} = \frac{nxP_n(x) - nP_{n-1}(x)}{x^2 - 1} \tag{B.13}$$

Our last intermediate step is to reorder Bonnet's relation:

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x) \tag{B.14}$$

$$(n+1)P_{n+1}(x) - (n+1)xP_n(x) = nxP_n(x) - nP_{n-1}(x) \tag{B.15}$$

Combining equation B.13 with equation B.15 yields the desired result.

$$\frac{dP_n(x)}{dx} = \frac{(n+1)P_{n+1}(x) - (n+1)xP_n(x)}{x^2 - 1} \tag{B.16}$$

□

Appendix C

DIMENSIONAL ANALYSIS: BUCKINGHAM PI THEOREM

C.1 Overview

Recall that the Gaussian Process Model must estimate the acceleration (a) as a function of position (r). Note that the true dynamics are only a function of position (not velocity or time) since we are assuming that solar radiation and solar gravitational perturbations are considered separate from this framework. To avoid matrix ill-conditioning under the hood of the Gaussian Process Model, the function output (a) should be on the same order as the input (r) and near unity. As a result, it is convenient to solve dimensionless equations and train the Gaussian Process Model with dimensionless quantities $\tilde{a}$ and $\tilde{r}$.

C.2 Dimensionless Expressions in the Lowest Order System

Using Buckingham Pi Theorem, we seek to identify the dimensionless groups of the dynamics. We consider the set-up where $a \propto f(G, M, r)$ where G is the Gravitational constant, M is the mass of the primary, and r is the position. The dimension matrix is hence:

	a	G	M	r
Mass	0	-1	1	0
Length	1	3	0	1
Time	-2	-2	0	0

The null space is the vector $(1, -1, -1, 2)$, which corresponds to:

$$1 \propto \frac{ar^2}{GM} \tag{C.1}$$

or more naturally,

$$a \propto \frac{GM}{r^2} \tag{C.2}$$

as is expected for the point mass term in the gravitational expansion. A variant of this normalization will be used to make the differential equation dimensionless in the next section. It is critical to note that this normalization is immediate from Buckingham Pi theorem. Similarly, we can consider the instantaneous Keplerian period τ_{kepl} with the set-up $\tau_{kepl} \propto f(G, M, r)$. The dimension matrix is hence:

	τ_{kepl}	G	M	r
Mass	0	-1	1	0
Length	0	3	0	1
Time	1	-2	0	0

The null space is the vector $(-2, -1, -1, 3)$, which corresponds to:

$$1 \propto \frac{r^3}{\tau_{kepl}^2 GM} \tag{C.3}$$

or more naturally

$$\tau_{kepl} \propto \sqrt{\frac{r^3}{GM}} \tag{C.4}$$

as is expected for the Keplerian period. From the full Keplerian derivation, the proportionality constant is 2π.

C.3 Dimensionless Differential Equation

Recall from the methods section, that we have a dynamical system described by:

$$\begin{aligned}\vec{a} = &\left(\frac{-\mu x}{|r|^3} + \sum_{n=2}^{N_z}\left[a_{x,n,\text{zonal}}\right] + \frac{T_x}{m_s}\right)\hat{x}\\ &+ \left(\frac{-\mu y}{|r|^3} + \sum_{n=2}^{N_z}\left[a_{y,n,\text{zonal}}\right] + \frac{T_y}{m_s}\right)\hat{y}\\ &+ \left(\frac{-\mu z}{|r|^3} + \sum_{n=2}^{N_z}\left[a_{z,n,\text{zonal}}\right] + \frac{T_z}{m_s}\right)\hat{z}\end{aligned} \tag{C.5}$$

Without loss of generality, we can focus on the $\hat{x}$ direction. Specifically:

$$\frac{-\mu x}{|r|^3} + \sum_{n=2}^{N_z}\left[a_{x,n,\text{zonal}}\right] + \frac{T_x}{m_s} = a_x \tag{C.6}$$

where

$$a_{x,n,\text{zonal}} = \frac{Kx}{|r|^2 - z^2}\left[|r|P_n\left(\frac{z}{|r|}\right) - zP_{n+1}\left(\frac{z}{|r|}\right)\right] \tag{C.7}$$

and K is given by Eq. C.8

$$K = \mu\tilde{J}_n(n+1)R^n r^{-(n+2)} \tag{C.8}$$

where (x, y, z) are the Cartesian coordinates, and P_n is the n-th degree Legendre P Polynomial. This formulation of K motivates three key insights. First, the parameter

$\mu = GM$ (i.e., the gravitation parameter of the primary) appears in every term of the dynamics, except for the control thrust. Second, expressing the position in units of the average radius (R) of the body allows us to eliminate R throughout and simplify the terms of the spherical harmonics expansion. Third, notice that M and R are intrinsic properties of the primary body. Even if they are not well constrained prior to flight, fixing these values and using them for normalization provides a necessary method for comparing gravity models across bodies, prevents numerical ill-conditioning, and more naturally captures the scale of the problem.

Applying the normalization to $\mu = GM$, we arrive at

$$\frac{-x}{|r|^3} + \sum_{n=2}^{N_z}\left[\frac{\tilde{J}_n(n+1)R^n r^{-(n+2)}x}{|r|^2 - z^2}\left[|r|P_n\left(\frac{z}{|r|}\right) - zP_{n+1}\left(\frac{z}{|r|}\right)\right]\right] + \frac{T_x}{\mu m_s} = \frac{a_x}{\mu} = \frac{a_x}{GM} \tag{C.9}$$

Next we apply the normalization of $\tilde{r} = \frac{r}{R}$ (or equivalently $r = \tilde{r}R$). Recognizing R is constant (namely so that $|R| = R$), we arrive at:

$$\begin{aligned}\frac{a_x}{\mu} =& \frac{T_x}{\mu m_s} - \frac{\tilde{x}R}{|\tilde{r}R|^3}\\ &+ \sum_{n=2}^{N_z}\left[\frac{\tilde{J}_n(n+1)R^n(\tilde{r}R)^{-(n+2)}\tilde{x}R}{|\tilde{r}R|^2 - (\tilde{z}R)^2}\left[|\tilde{r}R|\; P_n\left(\frac{\tilde{z}R}{|\tilde{r}R|}\right) - \tilde{z}R\; P_{n+1}\left(\frac{\tilde{z}R}{|\tilde{r}R|}\right)\right]\right]\end{aligned} \tag{C.10}$$

$$\begin{aligned}\frac{a_x}{\mu} =& \frac{T_x}{\mu m_s} - \frac{\tilde{x}}{R^2|\tilde{r}|^3}\\ &+ \sum_{n=2}^{N_z}\left[\frac{\tilde{J}_n(n+1)R^{-2}\tilde{r}^{\;-(n+2)}\tilde{x}R}{R^2\left(|\tilde{r}|^2 - (\tilde{z})^2\right)}\left[|\tilde{r}|R\; P_n\left(\frac{\tilde{z}}{|\tilde{r}|}\right) - \tilde{z}R\; P_{n+1}\left(\frac{\tilde{z}}{|\tilde{r}|}\right)\right]\right]\end{aligned} \tag{C.11}$$

$$\begin{aligned}\frac{a_x}{\mu} =& \frac{T_x}{\mu m_s} - \frac{\tilde{x}}{R^2|\tilde{r}|^3}\\ &+ \frac{1}{R^2}\sum_{n=2}^{N_z}\left[\frac{\tilde{J}_n(n+1)\tilde{r}^{\;-(n+2)}x}{|\tilde{r}|^2 - (\tilde{z})^2}\left[|\tilde{r}|\; P_n\left(\frac{\tilde{z}}{|\tilde{r}|}\right) - \tilde{z}\; P_{n+1}\left(\frac{\tilde{z}}{|\tilde{r}|}\right)\right]\right]\end{aligned} \tag{C.12}$$

$$\begin{aligned}\tilde{a}_x = \frac{a_x R^2}{\mu} =& \frac{T_x R^2}{\mu m_s} - \frac{\tilde{x}}{|\tilde{r}|^3}\\ &+ \sum_{n=2}^{N_z}\left[\frac{\tilde{J}_n(n+1)\tilde{r}^{\;-(n+2)}x}{|\tilde{r}|^2 - (\tilde{z})^2}\left[|\tilde{r}|\; P_n\left(\frac{\tilde{z}}{|\tilde{r}|}\right) - \tilde{z}\; P_{n+1}\left(\frac{\tilde{z}}{|\tilde{r}|}\right)\right]\right]\end{aligned} \tag{C.13}$$

notice that the form $\tilde{a}_x = \frac{a_x R^2}{\mu}$ matches the form of equation C.1 where we take $r \longrightarrow R$ as the normalizing constant (we cannot scale to r since it is a variable). This intuitively means that we scale the acceleration by the point mass acceleration at the average radius R. Since the parameter K is shared across each of a_x, a_y, and a_z, we label (for convenience in computation), $\tilde{K}$ defined via equation C.15.

$$\tilde{K} = \frac{KR^2}{\mu} = \tilde{J}_n(n+1)R^{(n+2)}r^{-(n+2)} \tag{C.14}$$

$$\tilde{K} = \tilde{J}_n(n+1)\left(\frac{R}{r}\right)^{n+2} = \tilde{J}_n(n+1)\tilde{r}^{\ -(n+2)} \tag{C.15}$$

Then, the dimensionless thrust is $\tilde{T}_x = \frac{T_x R^2}{\mu}$ to yield to yield

$$\tilde{a}_x = \frac{\tilde{T}_x}{m_s} - \frac{\tilde{x}}{|\tilde{r}|^3} + \sum_{n=2}^{N_z}\left[\tilde{a}_{x,n,\text{zonal}}\right] \tag{C.16}$$

Where $\tilde{a}_{x,n,\text{zonal}}$ is simplified via equation C.15 to

$$\tilde{a}_{x,n,\text{zonal}} = \frac{\tilde{K}\tilde{x}}{|\tilde{r}|^2 - \tilde{z}^2}\left[|\tilde{r}|P_n\left(\frac{\tilde{z}}{|\tilde{r}|}\right) - \tilde{z}P_{n+1}\left(\frac{\tilde{z}}{|\tilde{r}|}\right)\right] \tag{C.17}$$

Notice that $\tilde{r}$ is usually between 1 (at the average radius) and 100 (roughly the Hill sphere limit, depending on mass of the object). Then, the velocities are generally between 0.01 and 10. The accelerations are generally between 0.001 and 10. Hence, we prevent ill-conditioning since the position, velocity, and acceleration data are of similar magnitude and near unity.

C.4 Summary and Key Takeaways

The following conversions move between dimensionless quantities and the otherwise base scientific units (Table C.1).

Table C.1: Conversion table between dimensionless quantities (each has a tilde) and quantities with dimensions

Quantity	Expression	Explanation
Acceleration	$\tilde{a} = a\frac{R^2}{\mu}$	The acceleration felt at the spacecraft per the acceleration at the average radius of a point mass body.
Velocity	$\tilde{v} = v\sqrt{\frac{R}{\mu}}$	The spacecraft velocity per the velocity of circular orbit about a point mass at the average radius of the body. Equivalently, the normalized position per unit of normalized time.
Position	$\tilde{r} = \frac{r}{R}$	The spacecraft position per average radius of the body.
Period (Keplerian)	$\tilde{\tau}_{kepl} = \tau\left(2\pi\sqrt{\frac{\mu}{r^3}}\right)$	The instantaneous orbital period per period of a Keplerian orbit about a point mass. Note the equation is written in terms of r as a variable, rather than R which is a constant.
Time	$\tilde{t} = t\sqrt{\frac{\mu}{R^3}}$	The time elapsed per the time to complete one radian of a circular Keplerian orbit at the average radius of a point mass body. ($\tilde{\tau}_{kepl}/2\pi$ for $r \longrightarrow R$)