

Fall 12-6-2023

## Development of a Constellation Simulator for a 5G/IoT Mission Planning System

Franco Criscola  
*Embry-Riddle Aeronautical University*

Follow this and additional works at: <https://commons.erau.edu/edt>



Part of the [Navigation, Guidance, Control and Dynamics Commons](#)

---

### Scholarly Commons Citation

Criscola, Franco, "Development of a Constellation Simulator for a 5G/IoT Mission Planning System" (2023). *Doctoral Dissertations and Master's Theses*. 785.  
<https://commons.erau.edu/edt/785>

This Thesis - Open Access is brought to you for free and open access by Scholarly Commons. It has been accepted for inclusion in Doctoral Dissertations and Master's Theses by an authorized administrator of Scholarly Commons. For more information, please contact [commons@erau.edu](mailto:commons@erau.edu).

By

A Thesis Submitted to the Faculty of Embry-Riddle Aeronautical University

In Partial Fulfillment of the Requirements for the Degree of

Master of Science in Aerospace Engineering

Embry-Riddle Aeronautical University

Daytona Beach, Florida

By

THESIS COMMITTEE



---

---

---

Graduate Program Coordinator,  
Dr. Hever Moncayo

---

Date

---

Dean of the College of Engineering,  
Dr. James W. Gregory

---

Date

---

Associate Provost of Academic Support,  
Dr. Kelly Austin

---

Date

## **ABSTRACT**

The advancement of 5G and Internet-of-Things technologies has presented new challenges for telecommunications providers. One of the challenges is integrating these technologies with present networks. A solution has been found in low-Earth orbit satellite constellations. On one hand, this method increases coverage and reduces costs, but on the other it raises new problems like how to efficiently manage large constellations of spacecraft. This thesis introduces the Constellation Management System, developed in collaboration with i2Cat foundation. This novel tool is composed of two modules: the simulator and the scheduler. The former propagates satellite motion and computes visibility events to various targets across the globe, while the latter utilizes this information to generate an optimized task plan for all nodes in the network. The simulator is the focus of this work. Testing conducted in this investigation using several scenarios demonstrates the simulator's accuracy in both propagation and window determination. A new Lambert's solver employing the Theory of Functional Connections is developed, using functional interpolation to integrate the perturbed dynamics and boundary conditions directly into the formulation. The TFC framework allows the development of a faster, more robust algorithm compared to current methods.

## **ACKNOWLEDGMENTS**

I would like to thank my family, friends, and all those who unknowingly helped me through my first two years of graduate school at Embry-Riddle Aeronautical University. I thank the members of the Space Trajectories and Applications Research (STAR) Laboratory for their constant help, advice, and shared experiences. I would like to express my deep gratitude to my advisor, Dr. David Canales, for his support, advice, and shared knowledge. I would not be where I am without your assistance. It was a privilege to be part of your group for the past 2 years. The opportunities you have provided me throughout this degree were undoubtedly invaluable to build the person I am today. I would also like to thank my committee, Dr. Troy Henderson, Dr. Eduardo Rojas, and Dr. Joan A. Ruiz-de-Azua, for taking the time to participate in my journey and giving me advice on how to improve on my work. I also thank Dr. Daniele Mortari for giving me the opportunity to work with him on an incredible ground-breaking project.

# TABLE OF CONTENTS

ABSTRACT . . . . .	1
ACKNOWLEDGMENTS . . . . .	i
TABLE OF CONTENTS . . . . .	ii
LIST OF FIGURES . . . . .	vi
LIST OF TABLES . . . . .	vii
NOMENCLATURE . . . . .	viii
1 Introduction . . . . .	1
1.1 Problem Statement . . . . .	3
1.1.1 Tool Requirements . . . . .	3
1.2 Research Goal . . . . .	5
1.3 Highlights . . . . .	5
1.4 Thesis Outline . . . . .	6
2 Background . . . . .	7
2.1 State of the Art of Constellation Management Tools . . . . .	7
2.1.1 Orbital Simulators . . . . .	9
2.2 Dynamical Models . . . . .	10
2.3 Two-Line Elements and the SGP4 Propagator . . . . .	11
2.4 Unscented Kalman Filter . . . . .	12
2.5 Lambert’s Problem . . . . .	15
2.5.1 Lambert’s Problem Using the Theory of Functional Connections . . . . .	16
2.5.2 Differential Corrections . . . . .	23

2.5.3	Lambert-Gauss Solver . . . . .	25
2.5.4	Universal Variable Algorithm . . . . .	26
2.6	Software and Libraries . . . . .	27
3	Lambert’s Problem Results . . . . .	28
3.1	Unperturbed Problems . . . . .	28
3.2	Validation for Unperturbed Problems . . . . .	31
3.2.1	Lambert-Gauss Analysis . . . . .	32
3.2.2	Method of Universal Variable . . . . .	32
3.2.3	Differential Corrections Validation . . . . .	33
3.3	Perturbed Problems & Validation . . . . .	36
3.3.1	Earth’s Oblateness . . . . .	37
3.3.2	Third-Body Perturbations . . . . .	39
3.3.3	Solar Radiation Pressure . . . . .	40
3.4	Polynomial Analysis . . . . .	43
4	Constellation Management System . . . . .	46
4.1	Orbital Simulator Structure . . . . .	48
4.1.1	Input Loading . . . . .	49
4.1.2	Windows of Coverage Computation . . . . .	50
4.1.3	Output Generation and Formatting . . . . .	52
4.2	Scheduler Structure . . . . .	52
5	Orbital Simulator Results . . . . .	54
5.1	Testing Scenarios . . . . .	54
5.1.1	Scenario 1 . . . . .	55

5.1.2	Scenario 2 . . . . .	55
5.1.3	Scenario 3 . . . . .	55
5.2	Simulator Results . . . . .	56
5.2.1	Scenario 1 Results . . . . .	57
5.2.2	Scenario 2 Results . . . . .	59
5.2.3	Scenario 3 Results . . . . .	61
5.3	Lunar Transfer Results . . . . .	62
5.4	Unscented Kalman Filter Results . . . . .	64
6	Conclusions and Future Work . . . . .	66
	REFERENCES . . . . .	67

## LIST OF FIGURES

<b>Figure</b>	<b>Page</b>
Figure 1.1 Non-terrestrial network structure. . . . .	2
Figure 2.1 State of the art of constellation management tools. . . . .	8
Figure 2.2 Two-line element format sample. . . . .	12
Figure 2.3 Error between a TLE and SGP4 propagated states for one and five days. . . . .	13
Figure 2.4 Geometry of the Lambert’s problem. . . . .	16
Figure 3.1 Three example orbits in LEO and MEO. . . . .	29
Figure 3.2 Variation in computation time as a function of arc angle. . . . .	30
Figure 3.3 Earth-Moon transfer arc. . . . .	30
Figure 3.4 Sun-centered Earth-Venus transfer arc. . . . .	31
Figure 3.5 Error plot showing the difference between TFC and ODE45. . . . .	31
Figure 3.6 Lambert-Gauss solver flexibility analysis. . . . .	33
Figure 3.7 Universal variable solver flexibility analysis. . . . .	34
Figure 3.8 Computation time of universal variable and TFC as a function of angle. . . . .	34
Figure 3.9 Difference between TFC and DC: LEO and Cislunar orbits. . . . .	35
Figure 3.10 Difference between TFC and DC for interplanetary transfer. . . . .	35
Figure 3.11 Time difference between DC and TFC as a function of ToF. . . . .	36
Figure 3.12 Time difference between DC and TFC as a function of transfer angle. . . . .	37
Figure 3.13 Time difference between DC and TFC as a function of cord length. . . . .	37
Figure 3.14 Error plots showing the norm of the vector difference between each state. . . . .	39
Figure 3.15 Multi-revolution solution trajectory and error plot. . . . .	40
Figure 3.16 Perturbed Lambert arc under third-body perturbation. . . . .	41
Figure 3.17 Error plots showing the norm of the vector difference between each state. . . . .	41
Figure 3.18 Perturbed Lambert arc under Solar radiation pressure. . . . .	42

<b>Figure</b>	<b>Page</b>
Figure 3.19 Error plots showing the norm of the vector difference between each state. . . . .	43
Figure 3.20 Degree- $\alpha$ plot at $r_f/r_0 = 1$ (8378 km) showing the expected trend. . . . .	44
Figure 3.21 Degree- $\alpha$ plot at $r_f/r_0 = 2.32$ (19410 km) showing failures to converge. . . . .	45
Figure 3.22 $\alpha$ -radius plot at degree 30 showing sections of concentrated errors. . . . .	45
Figure 4.1 Constellation management system architecture. . . . .	47
Figure 4.2 Constellation management system data flow. . . . .	48
Figure 4.3 Simulator workflow structure. . . . .	49
Figure 4.4 Florida GS example WoC. . . . .	50
Figure 4.5 Simulator data flow. . . . .	52
Figure 5.1 Scenario 1 map. . . . .	55
Figure 5.2 Scenario 2 map. . . . .	56
Figure 5.3 Scenario 3 map. . . . .	58
Figure 5.4 Position difference between estimates and measurements. . . . .	65

## LIST OF TABLES

<b>Table</b>	<b>Page</b>
Table 3.1 Orbital parameters of trajectories in Figure 3.1. . . . .	29
Table 4.1 Simulator inputs. . . . .	49
Table 5.1 Scenarios common parameters. . . . .	54
Table 5.2 Scenario 2 location coordinates. . . . .	56
Table 5.3 Scenario 2 location coordinates. . . . .	57
Table 5.4 Scenario 1: GS2 WoC for LEMUR-2-JENNIFERSONG. . . . .	58
Table 5.5 Scenario 1: UE101 WoC for LEMUR-2-JENNIFERSONG. . . . .	59
Table 5.6 Scenario 1: Spain Northeast WoC for LEMUR-2-JENNIFERSONG. . . . .	59
Table 5.7 Scenario 1: Sun-charging WoC for LEMUR-2-JENNIFERSONG. . . . .	59
Table 5.8 Scenario 2: GS11 WoC for LEMUR-2-SATCHMO. . . . .	60
Table 5.9 Scenario 2: UE102 WoC for LEMUR-2-SATCHMO. . . . .	60
Table 5.10 Scenario 2: USA East WoC for LEMUR-2-SATCHMO. . . . .	60
Table 5.11 Scenario 2: Sun-charging WoC for LEMUR-2-SATCHMO. . . . .	60
Table 5.12 Scenario 2: ISL WoC for LEMUR-2-SATCHMO. . . . .	61
Table 5.13 Scenario 3: GS3 WoC for satellite 1. . . . .	61
Table 5.14 Scenario 3: UE103 WoC for satellite 1. . . . .	61
Table 5.15 Scenario 3: USA West WoC for satellite 1. . . . .	62
Table 5.16 Scenario 3: Sun-charging WoC for satellite 1. . . . .	62
Table 5.17 Scenario 3: ISL WoC for satellite 1. . . . .	62
Table 5.18 Lunar probe Madrid DSN WoC. . . . .	63
Table 5.19 Lunar probe Goldstone DSN WoC. . . . .	63
Table 5.20 Lunar probe Canberra DSN WoC. . . . .	63

## NOMENCLATURE

$5G$	5 <sup>th</sup> Generation
$\mathbf{a}_{\text{pert}}$	Perturbing Acceleration
$\mathbf{r}$	Spacecraft Position Vector
$\mathbf{r}_{3b}$	Position of Third-Body Relative to Primary
$\mathbf{r}_{sc-3b}$	Position of Third-Body Relative to Spacecraft
$\mu$	Standard Gravitational Parameter of Primary
$\mu_{3b}$	Standard Gravitational Parameter of Third-Body
$\rho_a, \rho_s, \rho_d$	Relative Material Reflectivity Properties
$\hat{\mathbf{n}}$	Direction Normal to Surface Area
$\hat{\mathbf{r}}_s$	Normalized Position of the Spacecraft Relative to the Sun
$A$	Exposed Surface Area
$API$	Application Programming Interface
$APSI$	Advanced Planning and Scheduling Initiative
$ASPEN$	Automated Planning/Scheduling Environment
$BVP$	Boundary-Value Problem
$CH$	Construction Heuristics
$CMS$	Constellation Management System
$CN$	Core Network
$D - SpaCPlanS$	Distributed Spacecraft Coordination Planning and Scheduling

<i>DC</i>	Differential Corrections
<i>DSS</i>	Distributed Satellite System
<i>ECI</i>	Earth-Centered Inertial
<i>EO</i>	Earth Observation
<i>EUROPA</i>	Extensible Universal Remote Operations Planning Architecture
<i>FOV</i>	Field-of-View
<i>G</i>	Gravitational Constant
<i>GEO</i>	Geosynchronous Equatorial Orbit
<i>GMAT</i>	General Mission Analysis Tool
<i>GS</i>	Ground Station
<i>GSA</i>	Ground Station Aggregator
<i>GSL</i>	Ground-Satellite Link
<i>HEO</i>	High-Earth Orbit
<i>IoT</i>	Internet of Things
<i>ISL</i>	Inter-Satellite Link
$J_2$	Perturbation due to Earth's Oblateness
<i>LEO</i>	Low-Earth Orbit
<i>LS</i>	Local Search
<i>MEO</i>	Middle-Earth Orbit
<i>MNO</i>	Mobile Network Operator

<i>MP</i>	Mission Provider
<i>MPA</i>	Mission Provider Aggregator
<i>MPS</i>	Mission Planning System
<i>NTN</i>	Non-Terrestrial Network
$P_{\text{SRP}}$	Solar Radiation Pressure Coefficient
<i>PDF</i>	Probability Density Function
$R_{\text{eq}}$	Earth's Radius
<i>RAAN</i>	Right Ascension of the Ascending Node
<i>s/c</i>	Spacecraft
<i>SGP4</i>	Simplified General Perturbation 4
<i>SRP</i>	Solar Radiation Pressure
<i>STK</i>	Systems Tool Kit
<i>STM</i>	State Transition Matrix
<i>TA</i>	Target Area
<i>TLE</i>	Two-Line Element
<i>ToF</i>	Time of Flight
<i>UE</i>	User Equipment
<i>UEC</i>	User Equipment Connection
<i>UKF</i>	Unscented Kalman Filter
<i>WoC</i>	Window of Coverage

## 1 Introduction

Since the beginning of the space era post-World War 2, low-Earth orbit (LEO) has been at the frontier of exploration in the hands of government agencies (e.g. NASA, ESA, JAXA). However, in the past three decades, the cost and risk of launching spacecraft (s/c) have decreased, and huge interest in LEO has emerged from privately-owned corporations; interest in the form of telecommunications [1]. This influx is correlated to the numerous advancements in technology seen since the beginning of the 21<sup>st</sup> century, many of which have come from the space sector itself. More recently, the current Internet network has gained popularity amongst researchers due to its relatively stagnant growth. Ground networks have not caught up to the rapid development of advanced technologies [2]. The rise of 5<sup>th</sup> generation networks (5G) and the Internet of Things (IoT) has been, for lack of a better word, revolutionary, and has left the preexisting foundations of telecommunications behind, leaving companies in search of solutions [3]. Solutions that come in the form of Non-Terrestrial Networks (NTN) [2, 4].

Non-Terrestrial Networks integrate with current terrestrial systems to promote global connectivity with a three-dimensional architecture. A typical structure of an NTN is represented in Figure 1.1. Here, three primary connection types are displayed: User Equipment Connections (UEC) between user equipment and ground stations; Ground-Satellite Links (GSL) between ground station and satellites; and Inter-Satellite Links (ISL) between different satellites of the constellation. These NTN are composed of elements in varying altitudes, from airplanes and high-altitude balloons to LEO satellites and beyond [5]. They are particularly useful in high-traffic areas, such as densely populated cities, and on the opposite end of the spectrum where regions of the globe lack the infrastructure to communicate with the existing networks [3]. The main idea behind an NTN is for a user to connect to the closest node, which then passes on information to other nodes if necessary until the service or network provider is reached [3].

The subject of this investigation is a satellite-based NTN. These elements of the network are generally developed using common technologies like large geosynchronous equatorial orbit (GEO) satellites. However, this method creates high latency due to its far distance from users

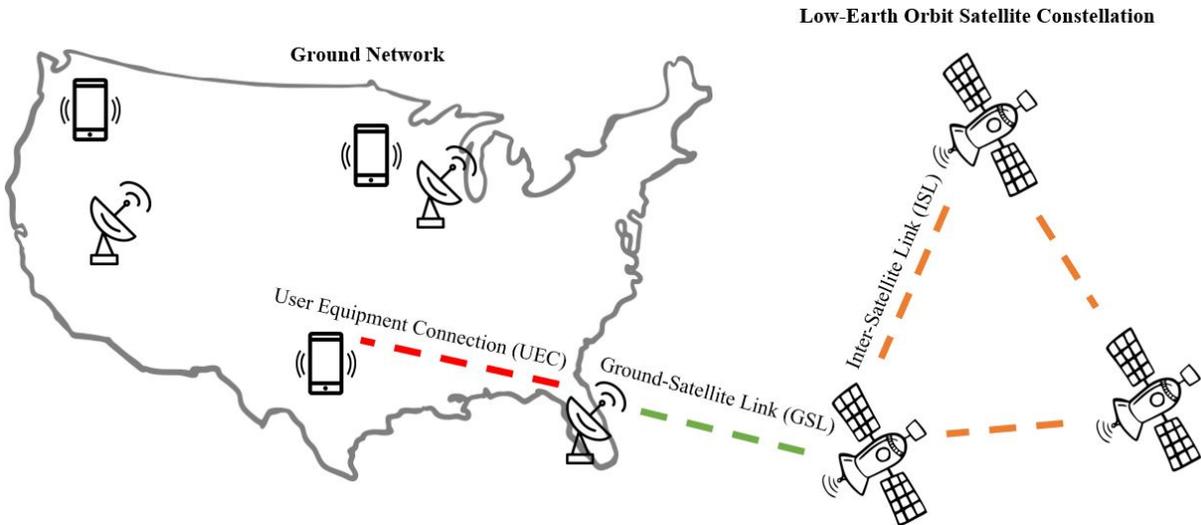


Figure 1.1 Non-terrestrial network structure.

and ground stations (GS), defeating the purpose of a high-speed IoT network, and creates high cost for the companies attempting to create such a network [6]. The altitude of the orbit may be decreased, which creates problems like “dead-zones”, but this could furthermore be fixed by increasing the number of satellites and creating an interconnected network of satellites called a Distributed Satellite System (DSS) [1, 7]. Moreover, increasing the number of satellites requires s/c mass and size to decrease to stay within project budget. The result is then a large constellation of small and cost-effective satellites in LEO interconnected by a network with a single objective, e.g. telecommunications. However, designing such complex systems causes previously unseen issues, such as inefficient constellation planning [8]. Increasing the size of the DSS and lowering the orbit means that the coverage time of satellites on particular areas, like GS and areas of interest, decreases substantially [9]. Thus, satellite operators lose the ability to manually command the s/c’s link to a GS. It is then necessary to automate this process by creating an optimized and generalized management tool, so that the constellation autonomously creates its own schedule in such a way that no satellite performs the same task simultaneously, every region of interest has connectivity to the DSS, and every constellation that uses the tool adapts it to fit its needs [9].

## **1.1 Problem Statement**

Several constellation management tools exist in the literature that are used to optimize task plans [10–14]. However, they lack a few key properties: easy customization; versatility for different applications (current state of the art is focused to a constellation type and mostly devoted to Earth observation [15, 16]), and easy to upgrade. Due to these characteristics, there is a gap that this investigation attempts to address. It is usually the case that conflicts arise between a scheduler and a chosen orbital simulation software. There are multiple orbit simulators that are currently used in the commercial and educational environments [17]. These are generally closed source, and hence require a purchase and training to use. Similarly, their general purpose characteristics create problems when trying to merge them with other software. The goal is thus to design a constellation management tool that works independently from any other software. This enables the simple combination of the two modules: the scheduler [18] and the simulator [19]. This work focuses on the latter; the simulator developed for the presented tool is designed independently as an algorithm to be built specifically for the constellation management.

### **1.1.1 Tool Requirements**

Developing an efficient task planner requires, among other things, full knowledge and prediction of the future state of the constellation. Although the environment of LEO is relatively predictable with current technologies, performing orbit determination and calculation of Windows of Coverage (WoC) becomes challenging when hundreds of satellites are present. Constellation operators have come up with constellation management planners that perform task assignments. These types of tools typically consist of two main components: an orbital simulator, whose role is to predict the future states of the constellation; and a task planner, which retrieves data from the simulator and generates an optimal task plan. With this goal and structure in mind, this thesis develops a set of categories used to assess the state of the art: automation, ease-of-use, and versatility.

Automation refers to the ability to complete its objectives with minimal human intervention. There are multiple aspects of the tool where automation is applied: from optimization algorithms for the task planner, to smooth data transfer between simulator and planner. Ideally, human intervention

should occur at the beginning of the process, during self-checking procedures, or to add new entities. The faster this process is performed, the smoother it is uploaded to the constellation for schedule updates.

Ease-of-use and automation are two distinct qualities that together serve to develop a powerful tool. Ease-of-use refers to the software's independence of complex, and typically licensed, external software, as well as the ability to easily update the constellation's entities. The goal is to ensure smooth communication between modules, and to allow operators to be easily trained in the algorithm. Consequently, by being able to simply add entities into the constellation (new satellites, locations, or areas), expansion of the constellation is possible at any given time.

Versatility is the tool's ability to adapt to any constellation size and application. In other words, the tool shall be completely generic. Multiple tools that have been previously developed serve one specific constellation or are designed for one specific use in mind, as will be shown in later sections. Although 5G-IoT networks are the main focus of this thesis, there are no barriers to the application of the tool to Earth observation constellations, or even extra-terrestrial systems like the Moon or Mars. In particular, the latter application is becoming more relevant as the settlement of the mentioned bodies is being planned [20].

To develop a tool that meets all of the above requirements, careful structuring of the system architecture is needed. The satellites must be propagated with high-fidelity models using the most accurate telemetry and orbit determination techniques. The WoC must be computed for each satellite and location, including GSL, ISL, UEC, and target area (TA) visibility. Additionally, in order to correctly determine if a location can truly be communicated with, the signal's link budget must be considered. This is a calculation of the signal power at the receiving end of communications. If this value is too low, then signal acquisition is not achieved. To determine the satellite's power and data storage, battery and memory states must be propagated, using consumption models that most closely resemble the expected usage. This includes incorporating Sun visibility in cases where Solar panels are present on the spacecraft. This entire process must be embedded "in the loop"; that is, running perpetually and synchronously so that data is accessed by the scheduler

at any time. Whenever the constellation entities are modified, which includes the upload of the most recent telemetry, the algorithm must automatically run again. The scheduler must be able to autonomously access and interpret the data via an application programming interface (API). Given the presented challenges and requirements, a tool that is structured promoting automation and, most importantly, easy upgradeability as needed, is key for efficient management of NTN.

## **1.2 Research Goal**

This investigation presents a constellation management tool for 5G/IoT purposes, denominated the Constellation Management System (CMS), which meets all the criteria discussed above. This tool has been developed in cooperation with i2Cat foundation [21]. The primary focus of this thesis is the simulator module of the CMS. This includes an algorithm that propagates motion of satellites and performs orbit determination to estimate the true location of the *s/c*. Additionally, it uses the resulting information to compute the WoC to different events as well as all locations provided. Necessary interfaces and automation procedures are also developed to merge and establish communication between simulator and scheduler.

## **1.3 Highlights**

The contributions made by this work are summarized as follows:

- A task planner specialized to 5G-IoT constellations is presented in the CMS.
- No similar planner of this kind is found in the literature, to the author's knowledge.
- A new perturbed Lambert's problem solver is developed using the theory of functional connections (TFC), found to be more efficient than other tested solvers.
- Results are generated for unperturbed and perturbed scenarios, validating results with other known solvers.
- Results from orbital simulations are presented using multiple scenarios, ranging in complexity.
- The CMS is adaptable enough to consider any optimal trajectory derived using TFC.

## 1.4 Thesis Outline

Chapter 2 presents the background information relevant to this thesis. First, the state-of-the-art of constellation management systems is included, considering typical simulators used for similar applications. Then, relevant dynamical concepts are expanded on, including the basics of the two-body problem, orbital perturbations, and Lambert's problem. Software used and developed throughout this thesis is noted. Chapter 3 analyzes the performance of the developed perturbed Lambert's solver. Unperturbed and perturbed solutions are compared to three algorithms: differential corrections (DC), a Lambert-Gauss solver, and a universal variable solution. Chapter 4 presents the structure of the CMS, expanding on the automation processes developed for more efficient task planning. Inputs and outputs for the simulator are specified. Chapter 5 delves into the results of the orbital simulator, showcasing scenarios with varying satellites and locations, as well as simulated orbits obtained using TFC. An unscented Kalman filter (UKF) is finally applied and the resulting WoC are compared to the original. Chapter 6 concludes this thesis by summarizing results and presenting potential applications of the work. Additionally, future work is discussed.

## 2 Background

This section introduces key background concepts that are used throughout this thesis. First, the state of the art of constellation management tools and simulators is presented. This is followed by a brief introduction to the two-body dynamics, including relevant perturbations around the Earth influence:  $J_2$  perturbation due to Earth's oblateness, third-body perturbations, and Solar radiation pressure (SRP). Two-line elements are discussed as a method to retrieve telemetry. The unscented Kalman filter is justified and introduced. Finally, Lambert's problem is formulated, including three state-of-the-art algorithms to solve it: differential corrections, the Lambert-Gauss method, and a solver that leverages the universal variable. Key software, libraries, and databases used are presented.

### 2.1 State of the Art of Constellation Management Tools

With the advent of satellite constellations in the late 20<sup>th</sup> and early 21<sup>st</sup> centuries, engineers and researchers have known of the giant logistical problem that large constellations create. This started with Iridium [22], a privately operated telecommunications constellation, but has been a constantly reoccurring challenge for operators because each constellation is different, with varying number of satellites and functionality. Therefore, constellation managers tend to design tools uniquely tailored to their own satellites [23]. In most cases, these organizations, federal or otherwise, have developed management tools for their specific constellations to plan and schedule tasks. Some examples include:

- The Automated Planning/Scheduling Environment (ASPEN) [10] created by NASA as an automated planning tool, designed to be for use of generic independent satellites for Earth observation (EO) purposes.
- The Advanced Planning and Scheduling Initiative (APSI) [11] established by ESA to provide a framework for future growth of AI technologies in scheduling.
- The Extensible Universal Remote Operations Planning Architecture (EUROPA) [12] developed by NASA as a manual planner and analyzer.

- The Distributed Spacecraft Coordination Planning and Scheduling (D-SpaCPlanS) [13] developed for the privately owned Iridium constellation, and a few others designed for EO satellites [24].
- EUTELSAT’s New Generation Constellation Control System, a privately owned manager designed for EUTELSAT’s constellation [25].
- Hifly, from GMV, designed using ESA’s SCS-2000 kernel for command and control [26].

The aforementioned tools, summarized in Figure 2.1 using the requirements described in Chapter 1, are usually exclusive for the organizations that developed them or for a specific constellation and functionalities. Few have the capability of being open to any entity interested in the tool. Thus, it appears that, although lots of research and investment have gone into the development of management and planning tools, none have gone to develop such tools standardized for telecommunications applications.

TOOL	Versatility	Automation	Ease-of-use
ASPEN		●	
APSI	●		●
EUROPA	●		
D-SpaCPlanS		●	
EUTELSAT		●	
Hifly		●	

Figure 2.1 State of the art of constellation management tools.

Many of the concepts described in the literature refer to increasing the automation and self-planning capabilities of EO satellites, using different approaches like mathematical programming [27] or graph theory models [28]. Additionally, most research automation and routing protocols are on-board the satellites themselves, which does not take into consideration scenarios where the requirements needed are not met such as no ISL, limited autonomy of the satellites or

store-and-forward approaches. There is little reference to commercial systems (MIDAS, Hifly, etc.) to cover the tasks of management and optimization of the constellation resources. This foreshadows the fact that the satellite constellation operators are either using in-house developed software, powerful commercial software such as Hifly, but tailored to their specific needs, and/or are still semi-manually scheduling their constellation (which would require a lot of manpower and a challenging scalability). Some operators are subcontracting this task to a specialized center, who again uses one of the aforementioned methods. Additionally, these traditional solutions for constellation management are focused on optimizing satellite by satellite operations, which is not compatible with a large constellation [27].

### **2.1.1 Orbital Simulators**

The dynamics present in the LEO environment are well-studied. Multiple models and techniques have been developed, as well as simulation software that consider these. Two of the most well-known are NASA's General Mission Analysis Tool (GMAT) and AGI's Systems Tool Kit (STK) [17]. Both software have satellite propagation capabilities with high-fidelity models, as well as computation of WoC for different events. They are both proven software that have been and still are used by private and public organizations across the world. Given the requirements for an effective constellation management tool readily accessible to smaller companies, these software are insufficient for this purpose. The simulator must be:

- Designed in a compatible programming language with the scheduler. This allows smooth data transfer between modules.
- An in-house algorithm. The use of third-party software presents barriers for the merging of two modules or acquiring licenses.
- Easy to use. Using external software generally requires licensing and/or substantial training to operate.
- Easy to upgrade. The simulator shall perform the functions needed by the task planner, including (but not limited to) orbital propagation and determination, computation of ISL and WoC considering field-of-view (FOV) and link budget, and data storage.

Given these requirements, it appears necessary to develop an in-house simulator for efficient development of optimized task plans.

## 2.2 Dynamical Models

The equations used throughout this thesis are commonly known as the two-body equations of motion. Expressed in the Earth-centered inertial (ECI) frame, these equations are expressed by the following ordinary differential equation:

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^3}\mathbf{r} \quad (1)$$

where  $\mathbf{r}$  is the vector from the primary body to the body of interest,  $\mu = Gm$  is the standard gravitational parameter of the primary, calculated using the gravitational constant  $G$  and the mass of the primary  $m$ . Perturbations are simply added to this differential equation by including an additional term as follows:

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^3}\mathbf{r} + \mathbf{a}_p \quad (2)$$

The perturbing acceleration  $\mathbf{a}_{\text{pert}}$  may include any perturbation. However, in this thesis, only three are considered relevant to the vicinity of the Earth-Moon system:

- **Earth's oblateness.** The  $J_2$  perturbation is expressed as [29]:

$$\mathbf{a}_{J_2} = -\frac{3 J_2 \mu R_{\text{eq}}^2}{2} \frac{1}{r^5} \begin{pmatrix} x [1 - 5(z/r)^2] \\ y [1 - 5(z/r)^2] \\ z [3 - 5(z/r)^2] \end{pmatrix} \quad (3)$$

where  $J_2 = 1.082629 \cdot 10^{-3}$  and  $R_{\text{eq}} = 6,378.137$  km is the Earth's equatorial radius.

- **Third-body.** The third-body perturbation is expressed as [30]:

$$\mathbf{a}_{3b} = \mu_{3b} \left( \frac{\mathbf{r}_{sc-3b}}{r_{sc-3b}^3} - \frac{\mathbf{r}_{3b}}{r_{3b}^3} \right) \quad (4)$$

where  $\mu_{3b}$  is the standard gravitational parameter of the third-body,  $\mathbf{r}_{sc-3b}$  is the vector of the third-body relative to the spacecraft, and  $\mathbf{r}_{3b}$  is the position vector of the third-body relative to the primary.

- **Solar radiation pressure.** The SRP perturbation is expressed as [31]:

$$\mathbf{a}_{\text{SRP}} = \frac{P_{\text{SRP}} A}{m} \left[ \rho_a (\hat{\mathbf{n}}^T \hat{\mathbf{r}}_s) \hat{\mathbf{r}}_s + 2\rho_s (\hat{\mathbf{n}}^T \hat{\mathbf{r}}_s)^2 \hat{\mathbf{n}} + \rho_d (\hat{\mathbf{n}}^T \hat{\mathbf{r}}_s) \left( \hat{\mathbf{r}}_s + \frac{2}{3} \hat{\mathbf{n}} \right) \right] \quad (5)$$

where  $\rho_a$ ,  $\rho_s$ , and  $\rho_d$  are the relative material reflectivity properties of the surface ( $\rho_a + \rho_s + \rho_d = 1$ ),  $A$  is the surface area exposed to the Sun,  $\hat{\mathbf{n}}$  the normal direction to the surface,  $\hat{\mathbf{r}}_s$  the Sun-to-satellite direction, and  $P_{\text{SRP}}$  is the Solar radiation pressure, given by  $4.57 \cdot 10^{-6}$  N/m<sup>2</sup>.

The three perturbations are included in the solution to the perturbed Lambert’s problem presented in this thesis.

### 2.3 Two-Line Elements and the SGP4 Propagator

In order to more accurately test the CMS tool, real satellite telemetry is preferred. A common way to store this data is through two-line element (TLE) sets [32]. A TLE is a format developed by NORAD that stores the mean orbital elements of a spacecraft. A sample TLE is shown in Figure 2.2, obtained on October 11<sup>th</sup>, 2023 at 03:05:15.931104 UTC. This format consists of the mean orbital elements at the time of the measurement, the julian date, satellite identifying information, as well as drag-related properties like the ballistic coefficient. Note that the “mean motion” and “revolution number” identifiers in a TLE will appear concatenated if the revolutions exceed 9999, as in the case of this sample. The TLE for a satellite is updated by NORAD when the difference between measurement and propagation is more than 5 km [32], which typically occurs every 8 to 12 hours. After a new TLE is generated, they are uploaded to the CelesTrak website [33]. In order to propagate a TLE, a Simplified General Perturbation 4 (SGP4) model is used in this work. This propagator is a perturbation-based statistical model that takes into account Earth’s oblateness, Moon’s gravitational pull, atmospheric drag, and other lesser perturbations [34]. It was developed

by NORAD to specifically propagate a TLE. However, if enough perturbations are introduced into the dynamical model then the dynamics are equivalent the SGP4 solution.

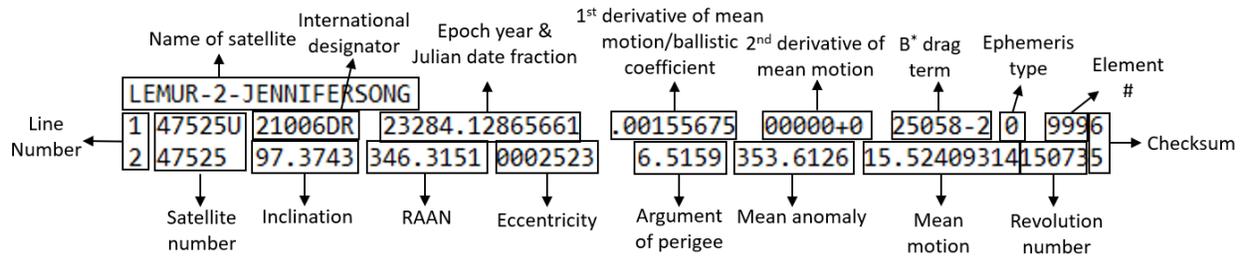
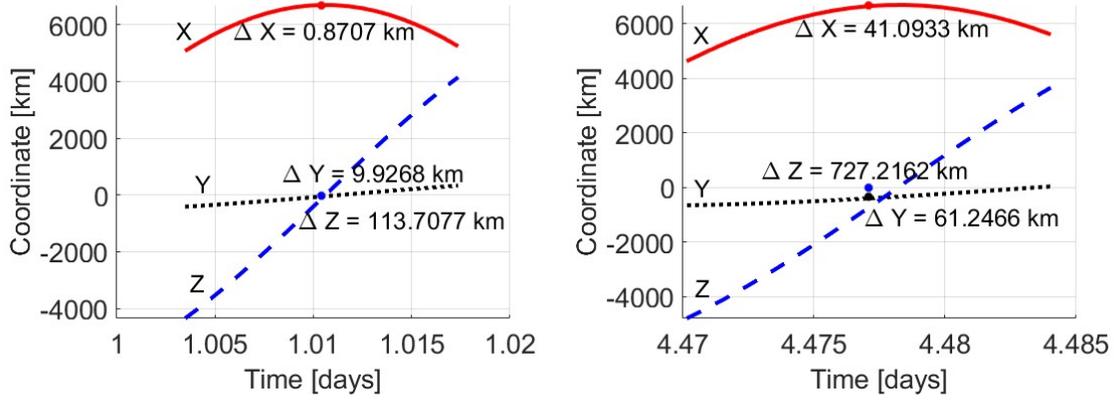


Figure 2.2 Two-line element format sample.

## 2.4 Unscented Kalman Filter

Literature regarding accuracy of a TLE over time shows that, on average, propagating a standard TLE with an SGP4 model yields an error of 1-2 km after 24 hours of propagation [32, 34, 35]. Figure 2.3 shows testing performed to confirm this error. A TLE obtained on September 17<sup>th</sup> is propagated for 5 days. Two additional TLE measurements are obtained after approximately one day (Figure 2.3a) and five days (Figure 2.3b), and then compared to the state. The error between the SGP4 propagated states and the measurements are shown on the plot, with each line representing an axis. The three  $\Delta$  values show the difference between the TLE (i.e., new measurement) and the propagation. Note that, due to the fact that TLE times are not entirely predictable, it is difficult to obtain measurements at exactly the time required, thus a measurement was only obtained after  $\sim 4.48$  days, rather than 5. For task scheduling purposes, it is convenient, if not necessary, to predict the state of the constellation after extended periods of time. From this figure, it is clear that the error present after even a single day is substantial for LEO.

Given these results, an orbit estimation technique is necessary to ensure accuracy of the propagated states. The most popular estimation technique is the Kalman filter and its variants. The unscented Kalman filter (UKF) corresponds to a statistical estimation technique used to address estimation problems based on probability density functions (PDF) [36]. Unlike an extended



(a) TLE propagated for 1 day.

(b) TLE propagated for 5 days.

Figure 2.3 Error between a TLE and SGP4 propagated states for one and five days.

Kalman filter, a UKF estimates the PDF of the system, enabling the determination of the most statistically probable state. This is achieved through the application of unscented transforms [36]. Such approach utilizes dynamical and measurement models to capture the system's dynamics and measurement characteristics, respectively:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}, \boldsymbol{\omega}, t) \quad \text{and} \quad \mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, \boldsymbol{\nu}_k) \quad (6)$$

where  $k$  represents the measurement number,  $\mathbf{x}_k$  is the  $k^{\text{th}}$  state,  $\mathbf{u}$  is the control input,  $\boldsymbol{\omega}$  is the process noise,  $t$  is the  $k^{\text{th}}$  time-step,  $\mathbf{y}_k$  is the measured state, and  $\boldsymbol{\nu}_k$  is the measurement noise. It is adequately assumed that the measurement model  $\mathbf{h}$  is entirely independent of the dynamical model  $\mathbf{f}$ . Note that the controller is beyond the scope of this research, i.e.,  $\mathbf{u} = 0$ . The algorithm's independence to the partial derivatives, i.e., the Jacobian, opens the door for applications around more complex dynamical problems where the full detailed model is not necessary [36]. Similarly, statistical models, such as an SGP4 propagator, are also applicable within a UKF [37]. With an SGP4 model, no information is known or needed about  $\mathbf{f}$ . The algorithm for a UKF is as follows:

1. Initialize estimate:

$$\hat{\mathbf{x}}_0 = E[\mathbf{x}_0] \quad \text{and} \quad P_0 = E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0^+)(\mathbf{x}_0 - \hat{\mathbf{x}}_0^+)^T] \quad (7)$$

where  $E$  is the expectation value and  $P_0$  is the initial covariance.

## 2. Propagation:

- Compute  $2N$  sigma-points  $\hat{\mathbf{x}}_{k-1}^{(i)} = \hat{\mathbf{x}}_{k-1}^+ + \tilde{\mathbf{x}}^{(i)}$ , where  $N$  is the state vector length. The sigma-points represent possible states within the covariance bubble of the current time-step. Each point is obtained from the  $i$ -th column of:

$$\tilde{\mathbf{x}}_k^{(i)} = \left(\sqrt{NP_{k-1}^+}\right)^T \quad i = 1, 2, \dots, N \quad (8)$$

$$\tilde{\mathbf{x}}_k^{(N+i)} = -\left(\sqrt{NP_{k-1}^+}\right)^T \quad i = 1, 2, \dots, N \quad (9)$$

Thus, a grouping of points, where each point represents one state, is obtained within the neighborhood of the current estimate. The matrix square root is obtained via Cholesky decomposition [36].

- Each one of these sigma-points is propagated using the dynamical model:

$$\hat{\mathbf{x}}_k^{(i)} = \mathbf{f}(\hat{\mathbf{x}}_{k-1}^{(i)}, \mathbf{u}_k, t_k) \quad (10)$$

- These  $2N$  propagated states are used to compute the new current estimate for the state:

$$\hat{\mathbf{x}}_k^- = \frac{1}{2N} \sum_{i=1}^{2N} \hat{\mathbf{x}}_k^{(i)} \quad (11)$$

Note that this estimate is simply the mean of all the sigma-points.

## 3. Update:

- Compute measurement using the measurement model for every sigma-point:

$$\hat{\mathbf{y}}_k^{(i)} = \mathbf{h}(\hat{\mathbf{x}}_k^{(i)}, t_k) \quad \text{and} \quad \hat{\mathbf{y}}_k = \frac{1}{2N} \sum_{i=1}^{2N} \hat{\mathbf{y}}_k^{(i)} \quad (12)$$

Note that the second equation is simply computing the mean between all these measurements. In this thesis, there is no measurement model. Rather, the state is being directly

measured. As such, all the sigma-points correspond to one same measurement.

- Compute  $y$ -covariance and  $x - y$  cross covariance using the sigma-points:

$$P_y = \frac{1}{2N} \sum_{i=1}^{2N} (\hat{\mathbf{y}}_k^{(i)} - \hat{\mathbf{y}}_k^-) (\hat{\mathbf{y}}_k^{(i)} - \hat{\mathbf{y}}_k^-)^T + R_k \quad (13)$$

where  $R_k$  is the measurement noise matrix, and

$$P_{xy} = \frac{1}{2N} \sum_{i=1}^{2N} (\hat{\mathbf{x}}^{(i)} - \hat{\mathbf{x}}_k^-) (\hat{\mathbf{y}}^{(i)} - \hat{\mathbf{y}}_k^-)^T \quad (14)$$

- Compute gain and update estimate:

$$K_k = P_{xy} P_y^{-1} \quad \hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k (\mathbf{y}_k - \hat{\mathbf{y}}_k) \quad P_k^+ = P_k^- + K_k P_y K_k^T \quad (15)$$

where  $P_k^-$  is the covariance of the state prior to propagation, and thus needs to be updated at each time-step.

It is clear from the algorithm above that a UKF is a computationally taxing algorithm, given that  $2N$  propagations are needed for each time-step. However, since a statistical propagator is used in this work, an extended Kalman filter is impossible unless a numerical Jacobian is used [37].

## 2.5 Lambert's Problem

Lambert's problem is a famous boundary-value problem (BVP) within the realm of orbital mechanics. It is defined by the initial and final position of the spacecraft, as well as the time of flight (ToF) between them. The formulation may also include additional perturbations outside of the two-body dynamics [29]. The problem was first defined by Heinrich Lambert (1728-1777): *Given the 2BP gravitational parameter ( $\mu$ ) of a planet, the time  $\Delta T$  required to accomplish a given transfer is a function of (a) the semi-major axis  $a$  of a conic trajectory joining the two position states, (b) the sum  $\mathbf{r}_0 + \mathbf{r}_f$  of the distances from the primary at the beginning and at the end of the transfer, and (c) the linear distance  $c$  between the two points (chord), i.e.:*

$$\sqrt{\mu} \Delta T = f(a, \mathbf{r}_0 + \mathbf{r}_f, c) \quad (16)$$

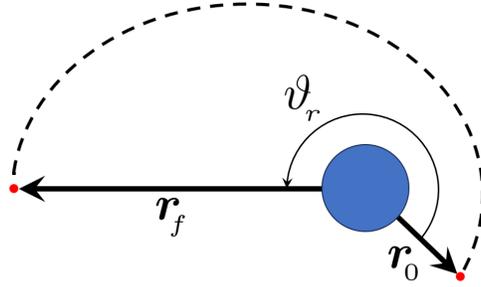


Figure 2.4 Geometry of the Lambert's problem.

In other words, Lambert's problem leverages the system's dynamics and the boundaries to obtain the trajectory of a spacecraft [38–40]. Figure 2.4 shows the geometry of the problem, where  $r_0$  and  $r_f$  represent the initial and final positions, respectively,  $\Delta T = t_f - t_0$ , and  $\vartheta_r$  is the angle covered by the arc connecting the boundaries, with degree of freedom being  $a$ .

Lambert's problem is used in multiple applications, such as orbit determination, orbit design and targeting, and orbital *rendezvous* [41]. No analytical solution to Lambert's problem is known, so authors have designed numerical algorithms that tend to trade robustness for efficiency, or *vice versa*. Multiple solvers exist, with authors grouping around the selected free parameters [42–49]. These include the semi-major axis [45, 50, 51], semi-latus rectum [47, 52], flight-path angle [48], or using Levi-Civita regularization [46]. An in-depth study of these algorithms shows that most of these lack robustness, the ability to consider perturbations, efficiency, or any combination of these qualities [41]. Thus, it is desirable to develop a Lambert's solver that possesses all of these characteristics. Four algorithms are developed and compared in this thesis: a solution leveraging TFC, a differential corrections (DC) scheme [38], the method of universal variables [29], and the Lambert-Gauss solver [30].

### 2.5.1 Lambert's Problem Using the Theory of Functional Connections

This thesis presents a novel framework that has been developed to solve the perturbed Lambert's problem leveraging TFC in collaboration with Dr. Mortari from Texas A&M. The formulation is

presented in this section, beginning with the definition of the Lambert's problem:

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^3} \mathbf{r} + \mathbf{a}_p(\mathbf{r}, \dot{\mathbf{r}}) \quad \text{subject to:} \quad \begin{cases} \mathbf{r}(0) = \mathbf{r}_0 \\ \mathbf{r}(\Delta T) = \mathbf{r}_f \end{cases} \quad (17)$$

where  $\mathbf{a}_p$  indicates a perturbing acceleration to the motion of a spacecraft,  $\mathbf{r}_0$  and  $\mathbf{r}_f$  represent the initial and final position vectors of the Lambert arc, and  $\Delta T$  denotes the ToF (Figure 2.4). In this study, the TFC [53–59] is applied to solve this problem by representing the position vector,  $\mathbf{r}(t)$ , with three variables (or functionals):  $p(t)$ ,  $\vartheta(t)$ , and  $h(t)$ , all within a new basis. These functionals define the position of the spacecraft in the Lambert arc as a function of time via:

$$\mathbf{r}(t) = p(t) [\hat{\mathbf{r}}_0 \cos \vartheta(t) + \hat{\mathbf{t}}_0 \sin \vartheta(t)] + h(t) \hat{\mathbf{h}}_0 \quad (18)$$

where  $\hat{\mathbf{r}}_0 = \frac{\mathbf{r}_0}{|\mathbf{r}_0|}$ ,  $\hat{\mathbf{r}}_f = \frac{\mathbf{r}_f}{|\mathbf{r}_f|}$ ,  $\hat{\mathbf{h}}_0 = \frac{\hat{\mathbf{r}}_0 \times \hat{\mathbf{r}}_f}{|\hat{\mathbf{r}}_0 \times \hat{\mathbf{r}}_f|}$ , and  $\hat{\mathbf{t}}_0 = \hat{\mathbf{h}}_0 \times \hat{\mathbf{r}}_0$ . In particular,  $[\hat{\mathbf{r}}_0, \hat{\mathbf{t}}_0, \hat{\mathbf{h}}_0]$  constitute the three directions of an orthogonal reference frame that is defined as long as the cross product  $\hat{\mathbf{r}}_0 \times \hat{\mathbf{r}}_f$  exists, i.e., a singularity occurs when  $\hat{\mathbf{r}}_0$  and  $\hat{\mathbf{r}}_f$  are parallel. The variable  $p(t)$  represents the projection of the radius vector on the  $[\hat{\mathbf{r}}_0, \hat{\mathbf{t}}_0]$  plane,  $\vartheta(t)$  is a parametric angle with no physical meaning (other than satisfying the bounds), and  $h(t)$  represents the orthogonal projection of the position vector with respect to the  $[\hat{\mathbf{r}}_0, \hat{\mathbf{t}}_0]$  plane. This formal expression to describe the evolution of the radius vector is particularly suitable if adopted by TFC. The first two derivatives of  $\mathbf{r}(t)$  are:

$$\dot{\mathbf{r}}(t) = [\dot{p} \cos \vartheta - p \dot{\vartheta} \sin \vartheta] \hat{\mathbf{r}}_0 + [\dot{p} \sin \vartheta + p \dot{\vartheta} \cos \vartheta] \hat{\mathbf{t}}_0 + \dot{h} \hat{\mathbf{h}}_0 \quad (19)$$

$$\begin{aligned} \ddot{\mathbf{r}}(t) = & [(\ddot{p} - p \dot{\vartheta}^2) \cos \vartheta - (2\dot{p} \dot{\vartheta} + p \ddot{\vartheta}) \sin \vartheta] \hat{\mathbf{r}}_0 + \\ & + [(\ddot{p} - p \dot{\vartheta}^2) \sin \vartheta + (2\dot{p} \dot{\vartheta} + p \ddot{\vartheta}) \cos \vartheta] \hat{\mathbf{t}}_0 + \ddot{h} \hat{\mathbf{h}}_0 \end{aligned} \quad (20)$$

Consequently, the position, velocity, and acceleration of the spacecraft are determined only with the three functionals along the entire Lambert arc. The boundary constraints of  $\vartheta(t)$  and  $h(t)$  are:

$$\begin{cases} \vartheta(0) = 0 \\ \vartheta(\Delta T) = \vartheta_r + 2k\pi \end{cases} \quad \text{and} \quad \begin{cases} h(0) = 0 \\ h(\Delta T) = 0 \end{cases} \quad (21)$$

where  $\hat{\mathbf{r}}_0^T \hat{\mathbf{r}}_f = \cos \vartheta_r$  and  $k$  is the number of revolutions (0 if single revolution). The boundary constraints of  $p(t)$  are:

$$p(0) = |\mathbf{r}_0| = r_0 \quad \text{and} \quad p(\Delta T) = |\mathbf{r}_f| = r_f \quad (22)$$

Additionally, the three functionals,  $p(t)$ ,  $\vartheta(t)$ , and  $h(t)$ , oscillate with the following approximate mean frequency:

$$\omega \approx \frac{2k\pi + \vartheta_r}{\Delta T} \quad (23)$$

These expected orbital oscillations are captured by including an oscillatory term in a vector of basis functions:

$$\mathbf{s}(z) = \begin{pmatrix} \mathbf{L}(z) \\ \cos(\omega t) \\ \sin(\omega t) \end{pmatrix}^T \quad (24)$$

where  $\mathbf{s}(z)$  is the basis functions vector, and the first two time derivatives are:

$$\dot{\mathbf{s}} = \begin{pmatrix} c \mathbf{L}(z)' \\ -\omega \sin(\omega t) \\ \omega \cos(\omega t) \end{pmatrix}^T \quad \text{and} \quad \ddot{\mathbf{s}} = \begin{pmatrix} c^2 \mathbf{L}(z)'' \\ -\omega^2 \cos(\omega t) \\ -\omega^2 \sin(\omega t) \end{pmatrix}^T \quad (25)$$

The vector  $\mathbf{L}(z)$  is a set of orthogonal polynomials (e.g., Legendre, Chebyshev, Laguerre). The example scenarios in this thesis are generated with Legendre polynomials. However, further analysis is performed on the polynomial type and degree using Gegenbauer polynomials. This kind of polynomial generalizes orthogonal polynomials to a single definition, where varying a particular parameter changes the representation. This will be further discussed later on in this thesis. The vector in Eq. (24) is used to define free functions,  $g_i(z)$ , which are in turn used to form the functionals. Necessarily, there must be three independent free functions to represent  $p(t)$ ,  $\vartheta(t)$ , and  $h(t)$ . These three functions,  $g_p(z)$ ,  $g_\vartheta(z)$ , and  $g_h(z)$ , are expressed as follows:

$$g_p(z) = \boldsymbol{\xi}_p^T \mathbf{s}(z), \quad g_\vartheta(z) = \boldsymbol{\xi}_\vartheta^T \mathbf{s}(z), \quad \text{and} \quad g_h(z) = \boldsymbol{\xi}_h^T \mathbf{s}(z). \quad (26)$$

and:

$$\begin{cases} \dot{g}_p = \boldsymbol{\xi}_p^T \dot{\mathbf{s}}, & \dot{g}_\vartheta = \boldsymbol{\xi}_\vartheta^T \dot{\mathbf{s}}, & \dot{g}_h = \boldsymbol{\xi}_h^T \dot{\mathbf{s}} \\ \ddot{g}_p = \boldsymbol{\xi}_p^T \ddot{\mathbf{s}}, & \ddot{g}_\vartheta = \boldsymbol{\xi}_\vartheta^T \ddot{\mathbf{s}}, & \ddot{g}_h = \boldsymbol{\xi}_h^T \ddot{\mathbf{s}} \end{cases} \quad (27)$$

The variables  $\boldsymbol{\xi}_p$ ,  $\boldsymbol{\xi}_\vartheta$ , and  $\boldsymbol{\xi}_h$  represent the three unknown coefficient vectors associated to  $p(t)$ ,  $\vartheta(t)$ , and  $h(t)$ . The variable  $z$  corresponds to the domain basis functions, represented in the form of collocation points due to the presence of the orthogonal polynomials  $\mathbf{L}(z)$ . A linear mapping is introduced between  $z$  and the time  $t$ :

$$z(t) = \frac{2}{\Delta T} t - 1 \in [-1, +1] \quad \rightarrow \quad t(z) = \frac{\Delta T}{2} (z + 1) \in [0, \Delta T] \quad (28)$$

from which a mapping constant  $c$  is determined as follows:

$$\dot{g} = \frac{dg}{dt} = \frac{dg}{dz} \cdot \frac{dz}{dt} = g' c = g' \frac{2}{\Delta T} \quad \text{and} \quad \frac{d^k g}{dt^k} = \frac{d^k g}{dz^k} c^k \quad (29)$$

This approach takes advantage of the three constrained expressions  $p(t)$ ,  $\vartheta(t)$ , and  $h(t)$  [53]. These are functionals that always satisfy the boundary constraints given in Eqs. (21) and (22). They represent the whole set of functions satisfying the boundary constraints (see Leake et al. [53] for

more specific detailed information). The bounds given in Eq. (21) and Eq. (22) allow TFC to derive the following expressions [53]:

$$\begin{cases} p(t, g_p) = g_p(z) + \frac{\Delta T - t}{\Delta T} [p_0 - g_p(-1)] + \frac{t}{\Delta T} [p(\Delta T) - g_p(+1)] \\ \vartheta(t, g_\vartheta) = g_\vartheta(z) - \frac{\Delta T - t}{\Delta T} g_\vartheta(-1) + \frac{t}{\Delta T} [\vartheta_r + 2k\pi - g_\vartheta(+1)] \\ h(t, g_h) = g_h(z) - \frac{\Delta T - t}{\Delta T} g_h(-1) - \frac{t}{\Delta T} g_h(+1) \end{cases} \quad (30)$$

whose time derivatives are:

$$\begin{cases} \dot{p}(t, g_p) = c g'_p(z) - \frac{1}{\Delta T} (p_0 - g_p(-1)) + \frac{1}{\Delta T} (p(\Delta T) - g_p(+1)) \\ \dot{\vartheta}(t, g_\vartheta) = c g'_\vartheta(z) + \frac{1}{\Delta T} g_\vartheta(-1) + \frac{1}{\Delta T} (\vartheta_r + 2k\pi - g_\vartheta(+1)) \\ \dot{h}(t, g_h) = c g'_h(z) + \frac{1}{\Delta T} g_h(-1) - \frac{1}{\Delta T} g_h(+1) \end{cases} \quad (31)$$

where  $g'_p(z)$ ,  $g'_\vartheta(z)$ , and  $g'_h(z)$ , indicate the first derivatives with respect to  $z$ . The second derivatives are:

$$\ddot{p}(t, g_p) = c^2 g''_p(z), \quad \ddot{\vartheta}(t, g_\vartheta) = c^2 g''_\vartheta(z), \quad \text{and} \quad \ddot{h}(t, g_h) = c^2 g''_h(z) \quad (32)$$

where  $\ddot{f} = \frac{d^2 f}{dt^2}$  and  $f'' = \frac{d^2 f}{dz^2}$ . It is important to remove the constant and linear terms in  $L(z)$

given that they have been already used to derive the constrained expressions in the above equations.

The set of Eqs. (30) constitute the TFC *constrained expressions* [53]. It is trivial to verify that these equations satisfy the boundary constraints for *any* expression of the free functions  $g_p(z)$ ,  $g_\vartheta(z)$ , and  $g_h(z)$ . Reference Leake et al. [53] describes the theory to derive the constrained expressions associated to a given set of constraints. These constraints are defined in terms of points, derivatives, integrals, limits, and any linear combination of them in uni-variate and multi-variate cases. The complete formulation has now been developed, and the goal becomes to solve for  $\xi_p$ ,  $\xi_\vartheta$ , and  $\xi_h$ .

Given that the problem consists of a larger number of equations than unknowns, it is solved via least-squares. The solution to the Lambert's problem is represented by the expression  $\mathcal{L}$  described

by  $\mathbf{r}(t)$  and its derivatives:

$$\mathcal{L} = \ddot{\mathbf{r}} + \frac{\mu}{r^3} \mathbf{r} - \mathbf{a}_p(\mathbf{r}, \dot{\mathbf{r}}) = \mathbf{0} \quad (33)$$

Using Eqs. (18) and Eqs. (30), as well as their derivatives, this problem is solved by estimating the unknown coefficients vectors,  $\boldsymbol{\xi}_p$ ,  $\boldsymbol{\xi}_\vartheta$ , and  $\boldsymbol{\xi}_h$ . Therefore, linearization around an estimated solution is obtained for every coordinate:

$$\mathbf{0} \approx \mathcal{L}_k + \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial \boldsymbol{\xi}_p} & \frac{\partial \mathcal{L}_x}{\partial \boldsymbol{\xi}_\vartheta} & \frac{\partial \mathcal{L}_x}{\partial \boldsymbol{\xi}_h} \end{bmatrix}_k \begin{Bmatrix} \boldsymbol{\xi}_p \\ \boldsymbol{\xi}_\vartheta \\ \boldsymbol{\xi}_h \end{Bmatrix}_k = \mathcal{L}_k + \mathcal{J}_k \boldsymbol{\xi}_k \quad (34)$$

from which the nonlinear least-squares solution is solved via an iterative process:

$$\boldsymbol{\xi}_{k+1} = \boldsymbol{\xi}_k - (\mathcal{J}_k^T \mathcal{J}_k)^{-1} \mathcal{J}_k^T \mathcal{L}_k \quad (35)$$

Note that this subscript  $k$  represents the current iteration (not to be confused with the number of revolutions),  $\boldsymbol{\xi}_k$  is a vector that includes  $\boldsymbol{\xi}_p$ ,  $\boldsymbol{\xi}_\vartheta$ , and  $\boldsymbol{\xi}_h$ , and  $\mathcal{J}_k$  is the Jacobian matrix of the system. Such a matrix requires the evaluation of the partial derivatives with respect to the three unknown vectors of coefficients. Starting with the partial derivatives of the functionals with respect to  $\boldsymbol{\xi}_p$ ,  $\boldsymbol{\xi}_\vartheta$ , and  $\boldsymbol{\xi}_h$ :

$$\frac{\partial p}{\partial \boldsymbol{\xi}_p} = \frac{\partial \vartheta}{\partial \boldsymbol{\xi}_\vartheta} = \frac{\partial h}{\partial \boldsymbol{\xi}_h} = \mathbf{s} - \frac{\Delta T - t}{\Delta T} \mathbf{s}_0 - \frac{t}{\Delta T} \mathbf{s}_f \quad (36)$$

$$\frac{\partial \dot{p}}{\partial \boldsymbol{\xi}_p} = \frac{\partial \dot{\vartheta}}{\partial \boldsymbol{\xi}_\vartheta} = \frac{\partial \dot{h}}{\partial \boldsymbol{\xi}_h} = \dot{\mathbf{s}} + \frac{\mathbf{s}_0}{\Delta T} - \frac{\mathbf{s}_f}{\Delta T} \quad (37)$$

$$\frac{\partial \ddot{p}}{\partial \boldsymbol{\xi}_p} = \frac{\partial \ddot{\vartheta}}{\partial \boldsymbol{\xi}_\vartheta} = \frac{\partial \ddot{h}}{\partial \boldsymbol{\xi}_h} = \ddot{\mathbf{s}} \quad (38)$$

which are used in the derivatives of  $r$  and  $\mathbf{r}$ :

$$\frac{\partial r}{\partial \xi_p} = \frac{p}{\sqrt{p^2 + h^2}} \frac{\partial p}{\partial \xi_p} \quad (39)$$

$$\frac{\partial r}{\partial \xi_h} = \frac{h}{\sqrt{p^2 + h^2}} \frac{\partial h}{\partial \xi_h} \quad (40)$$

$$\frac{\partial \mathbf{r}}{\partial \xi_p} = \frac{\partial p}{\partial \xi_p} (\hat{\mathbf{r}}_0 \cos \vartheta + \hat{\mathbf{t}}_0 \sin \vartheta) \quad (41)$$

$$\frac{\partial \mathbf{r}}{\partial \xi_\vartheta} = p \frac{\partial \vartheta}{\partial \xi_\vartheta} (-\hat{\mathbf{r}}_0 \sin \vartheta + \hat{\mathbf{t}}_0 \cos \vartheta) \quad (42)$$

$$\frac{\partial \mathbf{r}}{\partial \xi_h} = \frac{\partial h}{\partial \xi_h} \hat{\mathbf{h}}_0 \quad (43)$$

The partials of  $\dot{\mathbf{r}}$  and  $\ddot{\mathbf{r}}$  are

$$\frac{\partial \dot{\mathbf{r}}}{\partial \xi_p} = \left[ \frac{\partial \dot{p}}{\partial \xi_p} \cos \vartheta - \frac{\partial p}{\partial \xi_p} \dot{\vartheta} \sin \vartheta \right] \hat{\mathbf{r}}_0 + \left[ \frac{\partial \dot{p}}{\partial \xi_p} \sin \vartheta + \frac{\partial p}{\partial \xi_p} \dot{\vartheta} \cos \vartheta \right] \hat{\mathbf{t}}_0 \quad (44)$$

$$\begin{aligned} \frac{\partial \dot{\mathbf{r}}}{\partial \xi_\vartheta} &= \left[ -\dot{p} \frac{\partial \vartheta}{\partial \xi_\vartheta} \sin \vartheta - p \frac{\partial \dot{\vartheta}}{\partial \xi_\vartheta} \sin \vartheta - p \dot{\vartheta} \frac{\partial \vartheta}{\partial \xi_\vartheta} \cos \vartheta \right] \hat{\mathbf{r}}_0 + \\ &+ \left[ \dot{p} \frac{\partial \vartheta}{\partial \xi_\vartheta} \cos \vartheta + p \frac{\partial \dot{\vartheta}}{\partial \xi_\vartheta} \cos \vartheta - p \frac{\partial \vartheta}{\partial \xi_\vartheta} \dot{\vartheta} \sin \vartheta \right] \hat{\mathbf{t}}_0 \end{aligned} \quad (45)$$

$$\frac{\partial \dot{\mathbf{r}}}{\partial \xi_h} = \frac{\partial \dot{h}}{\partial \xi_h} \hat{\mathbf{h}}_0 \quad (46)$$

$$\begin{aligned} \frac{\partial \ddot{\mathbf{r}}}{\partial \xi_p} &= \left[ \left( \frac{\partial \ddot{p}}{\partial \xi_p} - \frac{\partial p}{\partial \xi_p} \dot{\vartheta}^2 \right) \cos \vartheta - \left( 2 \frac{\partial \dot{p}}{\partial \xi_p} \dot{\vartheta} + \frac{\partial p}{\partial \xi_p} \ddot{\vartheta} \right) \sin \vartheta \right] \hat{\mathbf{r}}_0 + \\ &+ \left[ \left( \frac{\partial \ddot{p}}{\partial \xi_p} - \frac{\partial p}{\partial \xi_p} \dot{\vartheta}^2 \right) \sin \vartheta + \left( 2 \frac{\partial \dot{p}}{\partial \xi_p} \dot{\vartheta} + \frac{\partial p}{\partial \xi_p} \ddot{\vartheta} \right) \cos \vartheta \right] \hat{\mathbf{t}}_0 \end{aligned} \quad (47)$$

$$\begin{aligned} \frac{\partial \ddot{\mathbf{r}}}{\partial \xi_\vartheta} &= \left[ -2p \dot{\vartheta} \frac{\partial \dot{\vartheta}}{\partial \xi_\vartheta} \cos \vartheta - \left( 2\dot{p} \frac{\partial \dot{\vartheta}}{\partial \xi_\vartheta} + p \frac{\partial \ddot{\vartheta}}{\partial \xi_\vartheta} \right) \sin \vartheta \right] \hat{\mathbf{r}}_0 + \\ &+ \left[ -2p \dot{\vartheta} \frac{\partial \dot{\vartheta}}{\partial \xi_\vartheta} \sin \vartheta + \left( 2\dot{p} \frac{\partial \dot{\vartheta}}{\partial \xi_\vartheta} + p \frac{\partial \ddot{\vartheta}}{\partial \xi_\vartheta} \right) \cos \vartheta \right] \hat{\mathbf{t}}_0 \\ &- \frac{\partial \vartheta}{\partial \xi_\vartheta} \left[ \left( \ddot{p} - p \dot{\vartheta}^2 \right) \sin \vartheta + \left( 2\dot{p} \dot{\vartheta} + p \ddot{\vartheta} \right) \cos \vartheta \right] \hat{\mathbf{r}}_0 + \\ &+ \frac{\partial \vartheta}{\partial \xi_\vartheta} \left[ \left( \ddot{p} - p \dot{\vartheta}^2 \right) \cos \vartheta - \left( 2\dot{p} \dot{\vartheta} + p \ddot{\vartheta} \right) \sin \vartheta \right] \hat{\mathbf{t}}_0 \end{aligned} \quad (48)$$

$$\frac{\partial \ddot{\mathbf{r}}}{\partial \xi_h} = \frac{\partial \ddot{h}}{\partial \xi_h} \hat{\mathbf{h}}_0 \quad (49)$$

The Jacobian is finally computed with the above formulas:

$$\frac{\partial \mathcal{L}}{\partial \xi_p} = \frac{\partial \ddot{\mathbf{r}}}{\partial \xi_p} - 3\mu \frac{\mathbf{r}}{r^4} \frac{\partial r}{\partial \xi_p} + \frac{\mu}{r^3} \frac{\partial \mathbf{r}}{\partial \xi_p} - \frac{\partial \mathbf{a}_p}{\partial \mathbf{r}} \frac{\partial \mathbf{r}}{\partial \xi_p} - \frac{\partial \mathbf{a}_p}{\partial \dot{\mathbf{r}}} \frac{\partial \dot{\mathbf{r}}}{\partial \xi_p} \quad (50a)$$

$$\frac{\partial \mathcal{L}}{\partial \xi_\vartheta} = \frac{\partial \ddot{\mathbf{r}}}{\partial \xi_\vartheta} + \frac{\mu}{r^3} \frac{\partial \mathbf{r}}{\partial \xi_\vartheta} - \frac{\partial \mathbf{a}_p}{\partial \mathbf{r}} \frac{\partial \mathbf{r}}{\partial \xi_\vartheta} - \frac{\partial \mathbf{a}_p}{\partial \dot{\mathbf{r}}} \frac{\partial \dot{\mathbf{r}}}{\partial \xi_\vartheta} \quad (50b)$$

$$\frac{\partial \mathcal{L}}{\partial \xi_h} = \frac{\partial \ddot{\mathbf{r}}}{\partial \xi_h} - 3\mu \frac{\mathbf{r}}{r^4} \frac{\partial r}{\partial \xi_h} + \frac{\mu}{r^3} \frac{\partial \mathbf{r}}{\partial \xi_h} - \frac{\partial \mathbf{a}_p}{\partial \mathbf{r}} \frac{\partial \mathbf{r}}{\partial \xi_h} - \frac{\partial \mathbf{a}_p}{\partial \dot{\mathbf{r}}} \frac{\partial \dot{\mathbf{r}}}{\partial \xi_h} \quad (50c)$$

Note that the last two terms in Eqs. (50a)- (50c) vary according to the perturbation type. Once the formulation is complete, the equations must be evaluated iteratively due to the indirect and direct dependence on  $\xi_p$ ,  $\xi_\vartheta$ , and  $\xi_h$ . Finally, the initial guess for  $\xi_{p0}$ ,  $\xi_{\vartheta0}$ , and  $\xi_{h0}$  to start the nonlinear iterative process is defined in two distinct ways:

1. If no prior knowledge is known about the particular problem, then  $\xi_{p0} = \xi_{\vartheta0} = \xi_{h0} = \mathbf{0}$ .

This means that the nonlinear iterative approach begins with an initial trajectory that linearly changes the values of  $p(t)$ ,  $\vartheta(t)$ , and  $h(t)$ , from their initial to their final values.

2. If prior knowledge to the problem is known, and the ToF is desired to be varied or perturbations are added, then the initial values of  $\xi_{p0}$ ,  $\xi_{\vartheta0}$ , and  $\xi_{h0}$  are obtained by best-fitting the initial known trajectory using the constrained expressions. This is done by obtaining the final  $\xi_i$  from the unperturbed problem.

Once an initial guess for  $\xi_p$ ,  $\xi_\vartheta$ , and  $\xi_h$  is provided, the nonlinear least-squares problem is solved. The formulation described above is more efficient and robust than other solvers, as will be demonstrated in this thesis. The two downsides of TFC, which are still present in other solvers, is the presence of a singularity when  $\vartheta_r$  is close to  $180^\circ$ , and the need for an initial guess to begin. However, this initial guess is simply defined as the zero-vector for unperturbed cases, making this process trivial.

## 2.5.2 Differential Corrections

A DC algorithm is a straightforward and generic scheme that consists of iteratively correcting an initial guess. This scheme is broad in application, meaning it is used for multiple different applications beyond Lambert's problem. It generally relies on a fairly precise initial guess for initial

state, leading to higher computation time. The algorithm begins by propagating an initial guess and initializing a constraint vector  $\mathcal{F}(\mathcal{Y})$ . This vector is defined as:

$$\mathcal{F}(\mathcal{Y}) = \left\{ X_a - X_t \quad Y_a - Y_t \quad Z_a - Z_t \right\}^T \quad (51)$$

where  $[X_a \ Y_a \ Z_a]$  represent the arrival coordinates and  $[X_t \ Y_t \ Z_t]$  are the target coordinates on arrival defined by  $r_f$ . The constraint vector is computed at each iteration until the Euclidean norm approaches a tolerance value. If the condition is not met, an equation derived from a Taylor series expansion is used to update the initial guess:

$$\mathcal{Y} = \mathcal{Y}_0 - \mathcal{J}(\mathcal{Y}_0)^{-1} \mathcal{F}(\mathcal{Y}) \quad (52)$$

where  $\mathcal{Y}$  is the updated initial velocity vector,  $\mathcal{Y}_0$  is the current velocity vector, and  $\mathcal{J}(\mathcal{Y}_0)$  is the Jacobian of the constraint vector relative to  $\mathcal{Y}$  evaluated at the final position. Note that the inverse of the Jacobian may be replaced by the pseudo-inverse if the matrix is not square. This Jacobian is easily retrieved from the state transition matrix (STM) propagated alongside the initial conditions. The STM is a measure of the sensitivity of the system to a changing initial condition, and it is propagated numerically alongside the state dynamics. The Jacobian is analytically defined as:

$$\mathcal{J}(\mathcal{Y}_0) = \begin{bmatrix} \frac{\partial X_a}{\partial \dot{X}_t} & \frac{\partial X_a}{\partial \dot{Y}_t} & \frac{\partial X_a}{\partial \dot{Z}_t} \\ \frac{\partial Y_a}{\partial \dot{X}_t} & \frac{\partial Y_a}{\partial \dot{Y}_t} & \frac{\partial Y_a}{\partial \dot{Z}_t} \\ \frac{\partial Z_a}{\partial \dot{X}_t} & \frac{\partial Z_a}{\partial \dot{Y}_t} & \frac{\partial Z_a}{\partial \dot{Z}_t} \end{bmatrix} \quad (53)$$

It is retrieved from the top-right 3x3 block matrix within the 6x6 STM. The process iterates once the updated guess is propagated. It is important to note that the closer the initial guess is to the true solution, the faster the convergence time. This concept will be explored during the testing phase in this thesis.

### 2.5.3 Lambert-Gauss Solver

This formulation was created by Lambert as an improvement to Gauss' original solution. It is a relatively simple approach, but possesses very little flexibility. It is slower than modern algorithms, and fails when the vectors  $r_0$  and  $r_f$  are too far apart. It has its foundations on Kepler's second law, which states that an object in orbit covers equal areas in equal times. The ratio between this area and the area of the triangle subtended by  $r_0$ ,  $r_f$ , and  $r$ , is expressed as  $y$  and is given by,

$$y^2 = \frac{\mu\Delta T^2 \sec^2(\theta_r/2)}{2r_0r_f \left( r_0 + r_f - 2\sqrt{r_0r_f} \cos(\theta_r/2) \cos(\Delta E/2) \right)} = \frac{m}{l + x_1} \quad (54)$$

where

$$l = \frac{r_0 + r_f}{4\sqrt{r_0r_f} \cos(\theta_r/2)} - \frac{1}{2} \quad m = \frac{\mu\Delta T^2}{8(r_0r_f)^{3/2} \cos^3(\theta_r/2)} \quad x_1 = \sin^2(\Delta E/4) \quad (55)$$

A temporary variable is then created to relate  $\Delta E$  and  $y$ , given by:

$$x_2 = \frac{\Delta E - \sin(\Delta E)}{\sin^3(\Delta E/2)} \quad (56)$$

However, in order to decrease computational time this expression is expanded via power series as a function of  $x_1$ , removing the need to solve for  $\Delta E$ . This expansion works only when  $\theta_r < 90^\circ$ .

The expression is determined as,

$$x_2 = \frac{4}{3} \left( 1 + \frac{6x_1}{5} + \frac{6(8)x_1^2}{5(7)} + \frac{6(8)(10)x_1^3}{5(7)(9)} + H.O.T \right) \quad (57)$$

With these variables,  $x_1$  is solved for in Eq. (54), and used with  $x_2$  in the update equation  $y = 1 + x_2(l + x_1)$ . The initial guess for  $y$  is typically chosen as 1, and the iteration loop proceeds until  $y$  converges. Variables  $l$  and  $m$  are constants for each scenario, and are thus outside of the loop.

The final  $y$  is used in the formula for the semi-parameter  $p$ :

$$p = \frac{r_0 r_f (1 - \cos(\theta_r))}{r_0 + r_f - 2\sqrt{r_0 r_f} \cos(\theta_r/2) \cos(\Delta E/2)} \quad (58)$$

where  $\cos(\Delta E/2) = 1 - 2x_1$ . The initial velocity is found via Lagrange coefficients:

$$f = 1 - \frac{r_f}{p} (1 - \cos(\theta_r)) \quad g = \frac{r_f r_0 \sin(\theta_r)}{\sqrt{\mu p}} \quad (59)$$

$$\dot{f} = \sqrt{\frac{1}{p}} \tan(\theta_r/2) \left( \frac{1 - \cos(\theta_r)}{p} - \frac{1}{r_f} - \frac{1}{r_0} \right) \quad \dot{g} = 1 - \frac{r_0}{p} (1 - \cos(\theta_r)) \quad (60)$$

where  $\dot{\mathbf{r}}_0 = (\mathbf{r}_f - f\mathbf{r}_0)/g$ .

#### 2.5.4 Universal Variable Algorithm

This solution is one of the more complicated formulations yet one of the fastest solvers [41]. The trade-off with this algorithm is its limitation to the two-body problem. See de la Torre Sangra and Fantino [41] for an in-depth analysis of this algorithm's flexibility. The algorithm is used in this investigation to validate the results obtained from the unperturbed TFC formulation, rather than for comparing efficiencies; this algorithm is semi-analytical nature and thus is inherently faster than the TFC solver. The formulation begins by replacing the universal anomaly  $\chi$  into the Lagrange coefficients equations:

$$f = 1 - \frac{\chi^2}{r_0} C(z) \quad g = \Delta T - \frac{1}{\sqrt{\mu}} \chi^3 S(z) \quad (61)$$

$$\dot{f} = \frac{\sqrt{\mu}}{r_0 r_f} \chi [z S(z) - 1] \quad \dot{g} = 1 - \frac{\chi^2}{r_f} C(z) \quad (62)$$

where  $\mu$  is the gravitational parameter of the two-body system,  $r_0$  is the magnitude of the initial position,  $r_f$  is the magnitude of the target position,  $z = \frac{\chi^2}{a}$ , and  $C(z)$  and  $S(z)$  are Stumpff functions. These functions are represented by an infinite series [29]. The formulation algebraically related three unknowns,  $\chi$ ,  $z$ , and  $h$  (implicit). The result is a non-linear function exclusively in terms of  $z$ , which is then solved using the Newton-Raphson method. This function is defined as:

$$F(z) = \left[ \frac{y(z)}{C(z)} \right]^{3/2} S(z) + A\sqrt{y(z)} - \sqrt{\mu}\Delta T \quad (63)$$

where  $y(z) = r_0 + r_f + A\frac{zS(z)-1}{\sqrt{C(z)}}$  and  $A = \sin\theta_r\sqrt{\frac{r_0r_f}{1-\cos\theta_r}}$ . The update equation is given by:

$$z_{k+1} = z_k - \frac{F(z_k)}{F'(z_k)} \quad (64)$$

Given an initial guess for  $z_0$ , the above equation is iteratively solved. The result is used to obtain the Lagrange coefficients via:

$$f = 1 - \frac{y(z)}{r_0} \quad g = A\sqrt{\frac{y(z)}{\mu}} \quad (65)$$

$$\dot{f} = \frac{\sqrt{\mu}}{r_0r_f}\sqrt{\frac{y(z)}{C(z)}}[zS(z) - 1] \quad \dot{g} = 1 - \frac{y(z)}{r_f} \quad (66)$$

A simple process then takes place to solve for the velocities at both ends of the Lambert arc using these coefficients.

## 2.6 Software and Libraries

The CMS and its modules are built in JAVA using the IntelliJ IDEA Ultimate application [60]. Using JAVA as the primary language allows for more efficient connection between modules thanks to the object-oriented interfaces. The library of functions *Orekit* [61] is used due to its abundance of functions useful to the development of the CMS. This includes calculation of WoC, ISL, SGP4 propagation, and many more. In order to store inputs and outputs to the CMS, two databases are used: InfluxDB [62] and MongoDB [63]. The specific usage of each is discussed later on in this thesis. InfluxDB is a time-series database that allows for the storage of time-based data (storing dates alongside relevant tags). MongoDB is an object-oriented database that is used for storing complex object structures. InfluxDB is used for storing the simulator's outputs, while MongoDB is used for storing the inputs. Further analysis of results is done on Matlab due to its ability to visually represent data in the form of plots and graphs.

### 3 Lambert’s Problem Results

To test the TFC-based Lambert solver, the results and simulations are split into two sections: unperturbed and perturbed problems. The results are validated by propagating the resulting initial velocity using Matlab’s ODE45, and then compared to the results obtained from other algorithms. All perturbations presented in Chapter 2 are then included in the algorithm. A polynomial analysis is finally conducted.

#### 3.1 Unperturbed Problems

This section introduces several test scenarios in LEO and middle-Earth orbit (MEO), with varying eccentricities and inclinations, as well as Cislunar and interplanetary transfers. Given that no perturbations are included in these testing scenarios, the orbital parameters of the resulting orbit are easily retrieved. The objective of this section is to demonstrate the algorithm’s capacity to generate solutions with various conditions and parameters.

Figure 3.1 shows three arcs generated via TFC, with different altitude, eccentricity, inclination, and right ascension of the ascending node (RAAN). Table 3.1 shows the orbital elements for each orbit. Additionally, Figure 3.2 shows the variation in computation time as a function of arc angle. This orbit is a transfer from MEO (2000 km altitude) to GEO, with varying angle and fixed ToF (2.5 hours). Ignoring the outliers, it is clear that the computation time increases as the angle increases. This is because TFC has a singularity at  $180^\circ$ . In fact, this simulation is limited to  $150^\circ$  due to divergence of the algorithm close to  $180^\circ$ . Two additional scenarios are generated in order to test the algorithm’s ability to find solutions at large distances and ToF, as well as around a different primary body. Figure 3.3 shows a GEO-Moon,  $130^\circ$ , 70-hour long transfer arc, and Figure 3.4 shows a Sun-centered Earth-Venus,  $120^\circ$ , 180-day long transfer arc. The solid-line arc in both plots represents the TFC-generated trajectory, while the remaining arcs represent the motion of the other objects. Note that the gravitational influence of third bodies is not considered in these scenarios.

Finally, to demonstrate that the generated trajectories obey the two-body dynamics of the system, Figure 3.5 shows an error plot between the lunar transfer of Figure 3.3 and the arc resulting from Matlab’s ODE45 propagator. The used time span is expanded using collocation points so that the

ranges between TFC and the propagator match. The initial velocity obtained using TFC is used in the propagator. The difference between both methods is small enough to be considered negligible. The results obtained in this section show that, although a singularity exists at angles close to  $180^\circ$ , TFC is robust and finds solutions of various orbit types. The results obtained are consistent with the system dynamics (Figure 3.5).

Table 3.1 Orbital parameters of trajectories in Figure 3.1.

Orbital Characteristic	Solid-Line Orbit	Dotted-Line Orbit	Black Orbit
Periapsis Altitude	500 km	500 km	3000 km
Eccentricity	0.5	0.5	0
Inclination	$0^\circ$	$165^\circ$	$30^\circ$
RAAN	$0^\circ$	$0^\circ$	$45^\circ$
Arc Angle	$120^\circ$	$150^\circ$	$90^\circ$

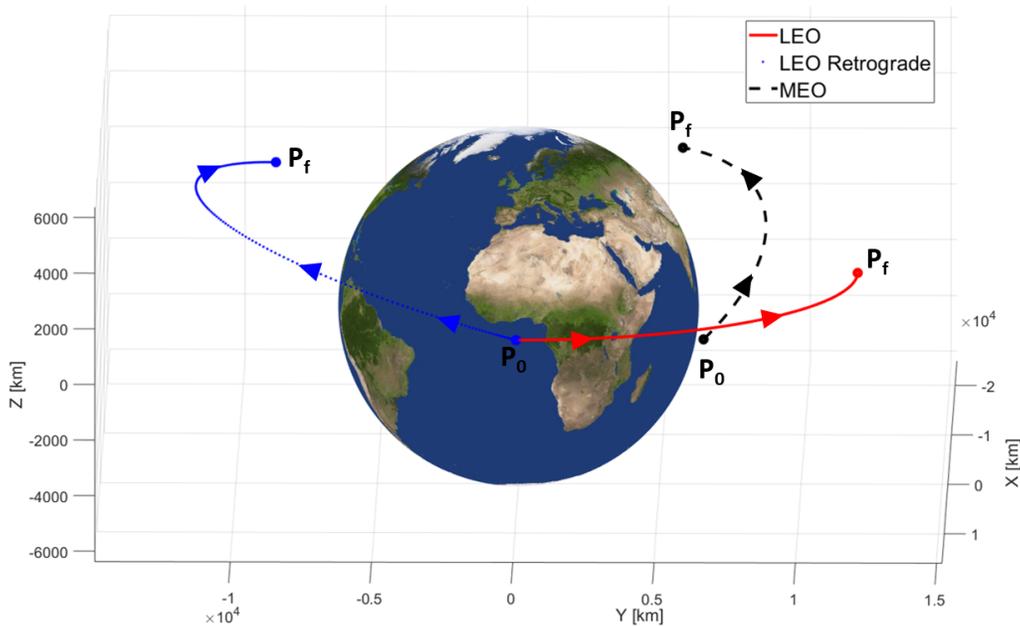


Figure 3.1 Three example orbits in LEO and MEO.

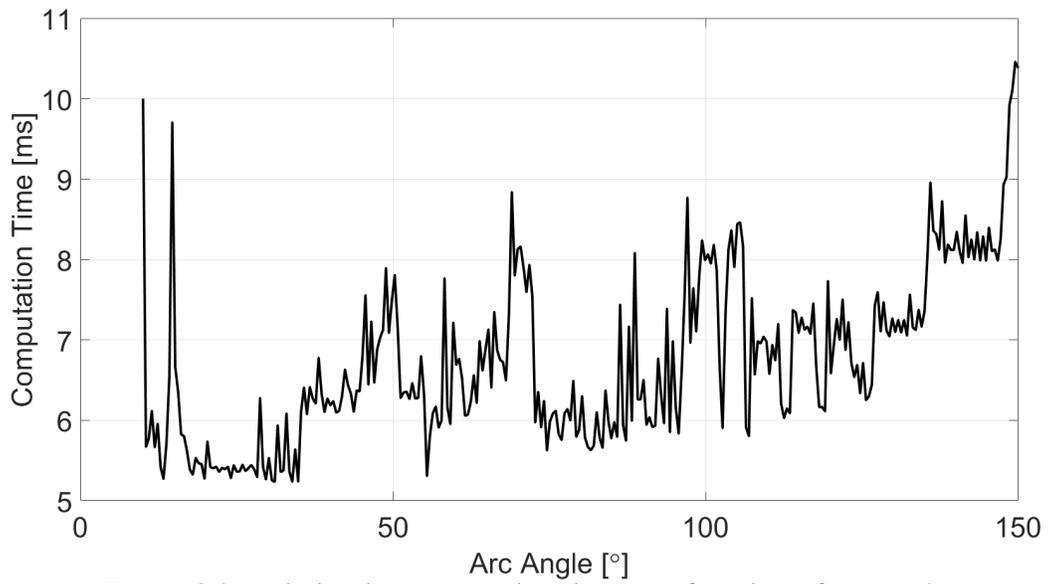


Figure 3.2 Variation in computation time as a function of arc angle.

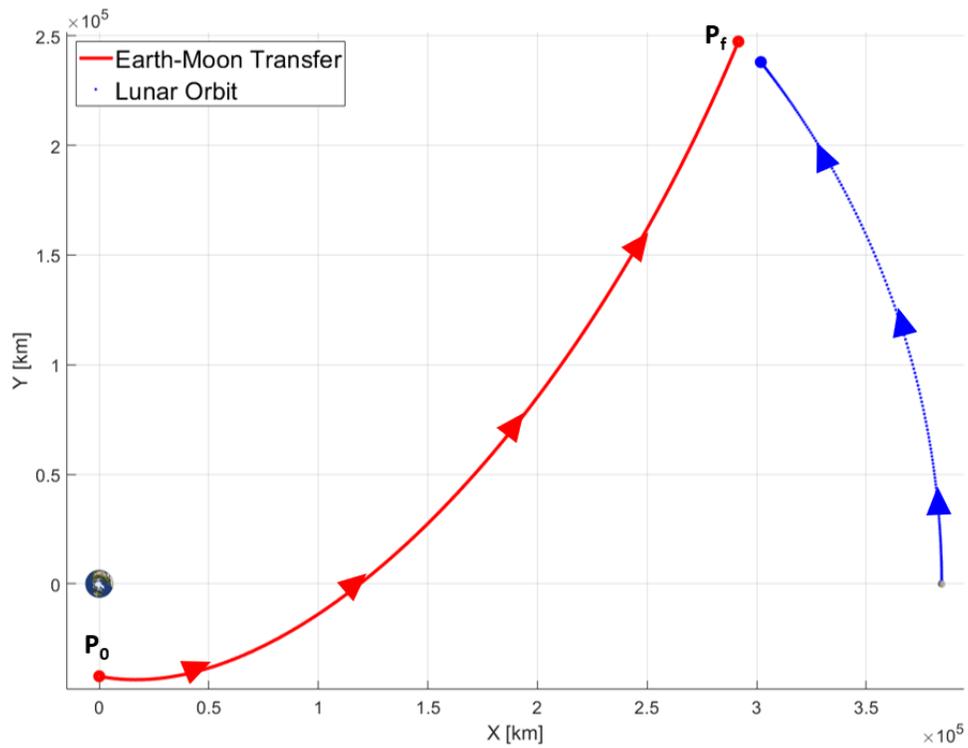


Figure 3.3 Earth-Moon transfer arc.

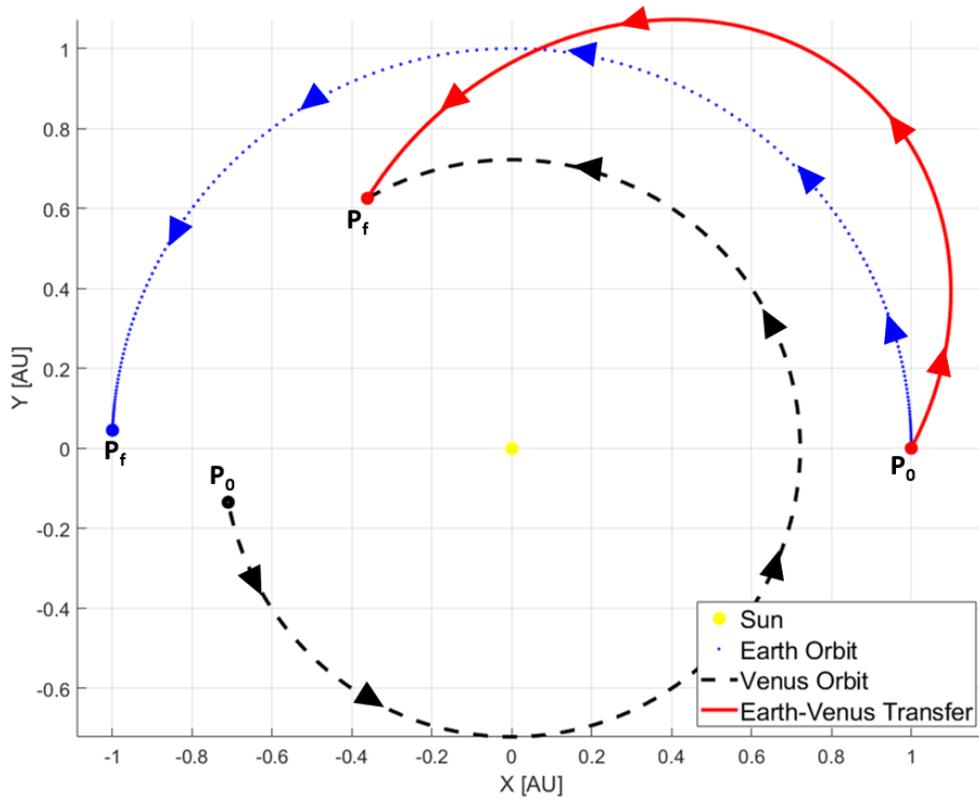


Figure 3.4 Sun-centered Earth-Venus transfer arc.

### 3.2 Validation for Unperturbed Problems

Three key algorithms in the literature are presented in this thesis to validate with the unperturbed solutions generated by the TFC-based algorithm. Testing reveals that the Lambert-Gauss solver

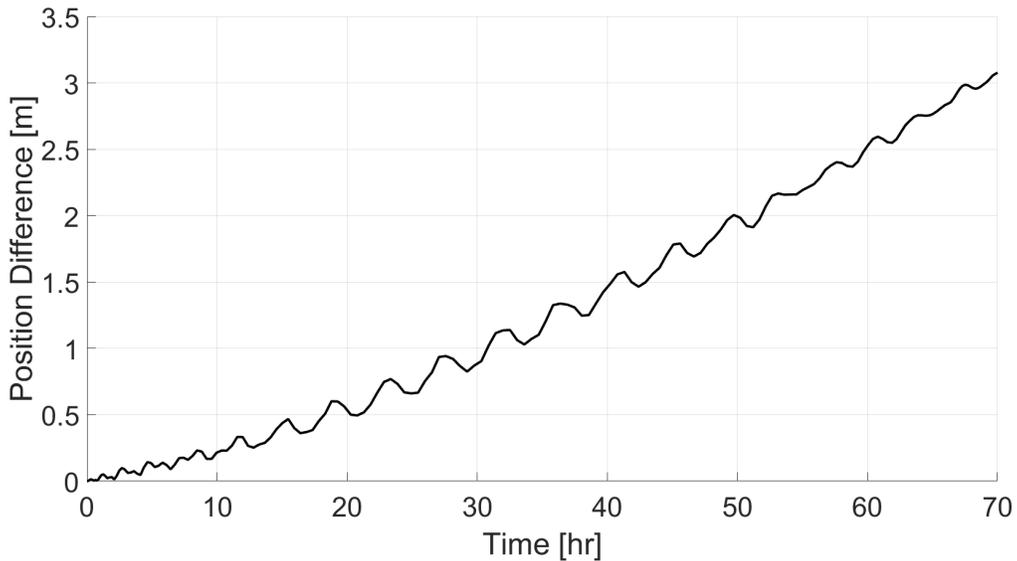


Figure 3.5 Error plot showing the difference between TFC and ODE45.

and the universal variables method fall short in comparison to TFC due to their restrictive nature. Comparative results highlight these algorithms' limited robustness, contrasting them with TFC and DC solutions.

### 3.2.1 Lambert-Gauss Analysis

The Lambert-Gauss solver is built upon Kepler's laws, thus being only comparable to the unperturbed TFC solver. Further analysis on the performance of the algorithm shows that this solver possesses two important limiting parameters: the transfer arc angle is limited to less than  $90^\circ$ , where angles approaching this value generate unreliable results, and the resultant orbit is limited to circular orbits. Note that convergence of  $y$  due to Eq. (54) is achieved regardless of these limitations, but the resulting initial velocity does not lead to  $r_f$  at the ToF. A simple test is performed on a 7078 km radius,  $30^\circ$  transfer arc. The ToF is varied from 8.23 min to 20.58 min, which corresponds to the ToF of a circular orbit ( $e=0$ ) to 2.5 times this ToF. The eccentricity for the resulting orbits is calculated from the ToF and boundary conditions, and plotted against the final position error (Figure 3.6). The minimum error of 18.5 km occurs when  $e$  is close to 0. As the eccentricity increases, the error increases exponentially. Given these strict limits, added to the fact that the algorithm only works within the two-body problem, no further comparison with the unperturbed TFC algorithm is done in this investigation.

### 3.2.2 Method of Universal Variable

The universal variable solver is another example of an algorithm built upon Kepler's laws. However, it is more robust than the Lambert-Gauss solver. The resultant orbit using this algorithm compared to the results for TFC shows virtually no difference between states at equal time steps. Figure 3.7 presents a surface plot showing multiple orbits, with the color bar representing the error between the target final position and the resultant final position from the solver. This plot is of an Earth-to-Moon transfer ([42,164; 0; 0] km to [-360,842; 131,336; 0] km), with an arc of  $150^\circ$  and a ToF varying between 0.5 (bottom-most orbit) and 4 days (top-most orbit). It is clear that, although the error does increase with ToF, it is still minimal within a meter. As expected, the solver performs better than the TFC algorithm within the tested unperturbed scenario (Figure 3.8).

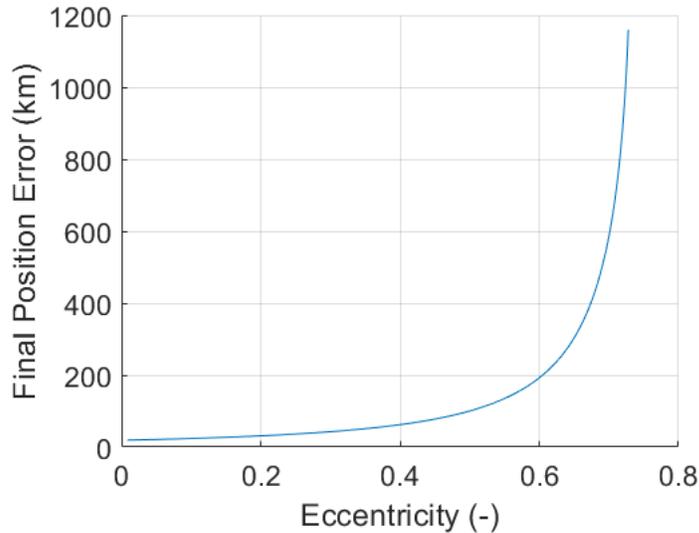


Figure 3.6 Lambert-Gauss solver flexibility analysis.

This comparison is done on a HEO-to-GEO (high-Earth orbit to geostationary orbit, [8,378; 0; 0] km to [42,164\*cos( $\theta$ ); 42,164\*sin( $\theta$ ); 0] km) transfer with varying  $\theta$  (transfer angle) at a fixed ToF (3 hours). A singularity for the TFC solver exists at angles close to 180°, so the variation in angle stops at 160°. Although the universal variable algorithm is highly efficient within the two-body environment, most real-life scenarios require perturbations. This solver may serve as a fast low-fidelity approximation, but it is unable to generate perturbed arcs.

### 3.2.3 Differential Corrections Validation

The DC algorithm is used to generate solutions for the prograde LEO arc (Figure 3.1), the Lunar transfer (Figure 3.3), and the Earth-Venus transfer (Figure 3.4). Figures 3.9 and 3.10 show error plots between the positions resulting from the DC and the TFC algorithms throughout the respective trajectories. There is a small position difference at the end of the arc. The TFC method guarantees that the boundaries are met. However, a key distinction between the two is that TFC has a much more lenient initial guess requirement. The DC method is initialized in this investigation using a Hohmann transfer as the initial guess. For the prograde LEO arc, the initial velocity guess is set to be in the  $y$ -direction. For the Lunar and Earth-Venus transfers, the direction is set to [1; 1; 0]/ $\sqrt{2}$  to achieve convergence. The procedure to obtain a good initial guess is generally more

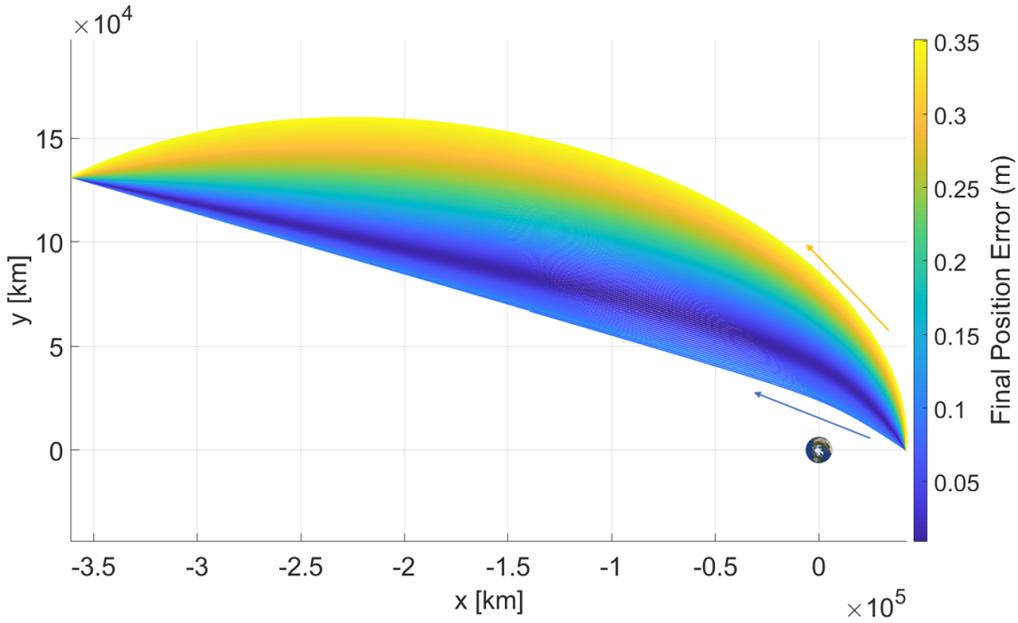


Figure 3.7 Universal variable solver flexibility analysis.

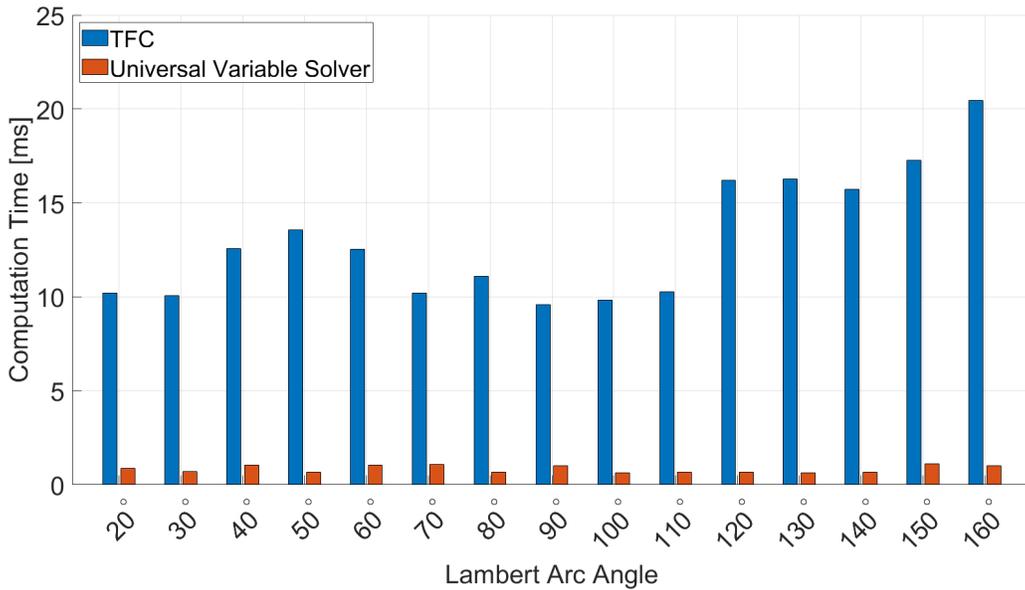


Figure 3.8 Computation time of universal variable and TFC as a function of angle.

complex than the approach taken in this thesis, but this is sufficient for the purposes of this project. From this perspective, TFC is clearly more efficient.

Three additional tests are conducted to determine parameters in which TFC is faster, slower, or

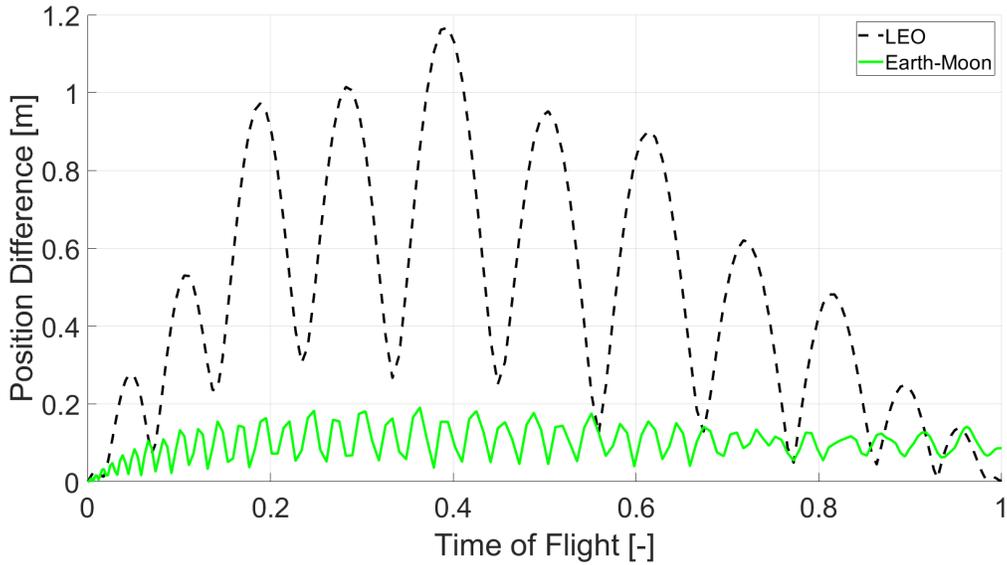


Figure 3.9 Difference between TFC and DC: LEO and Cislunar orbits.

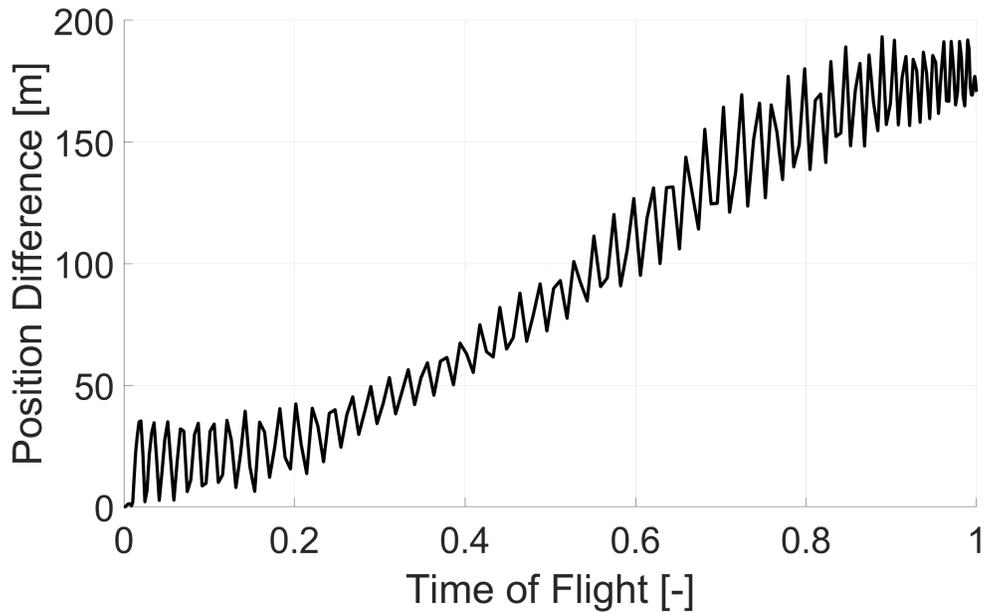


Figure 3.10 Difference between TFC and DC for interplanetary transfer.

equal to DC in terms of computation time. The first test is a MEO-GEO transfer with varying ToF (0.5-3.8 hours) while fixing the arc angle to  $120^\circ$  (Figure 3.11). The second test is a MEO-GEO

transfer with varying arc angle ( $10^\circ$ - $150^\circ$ ) while fixing the ToF to 2.5 hours (Figure 3.12). The third and final test varies the distance between initial and final positions (14,500-46,900 km) while fixing the ToF to 2.5 hours and the arc angle to  $120^\circ$  (Figure 3.13). The  $y$ -axis in all these plots represents the difference between TFC's computation time and DC's computation time, with negative values showing TFC as the faster algorithm. The DC algorithm in all cases is initialized using a Hohmann transfer as the initial guess. Although TFC is faster than DC in most cases, as both ToF and arc angle increase, DC's computation time decreases, with the exception of some outliers (Figures 3.11 and 3.12). Since DC is initialized using a Hohmann transfer, solutions approaching this initial guess converge faster. Figure 3.12 also highlights the divergence of TFC while approaching the singularity at  $180^\circ$ . Figure 3.13, on the other hand, raises a potential problem with TFC. As seen in the figure, there are large, smooth lines in the graph. These slopes represent the absence of data between two peaks, meaning the algorithm diverges at those cord lengths. Further investigation shows that varying the polynomial type and degree eliminates the gaps, and also decreases TFC's computation time. Nonetheless, given that no pattern is easily identified, the results from this figure are inconclusive.

### 3.3 Perturbed Problems & Validation

The TFC formulation presented in this thesis allows for the inclusion of any number of analytical, continuous, and differentiable perturbations. In order to include the perturbations, these must be

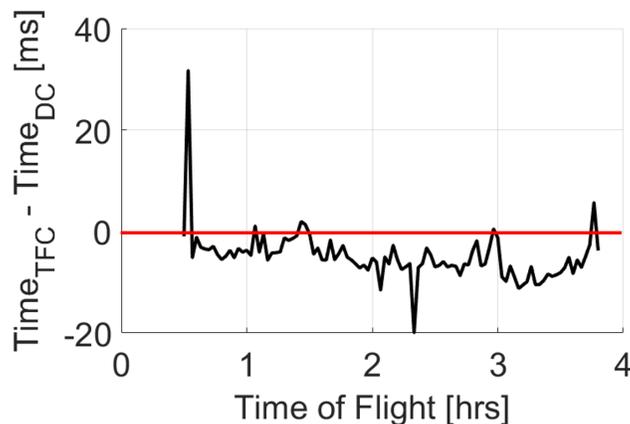


Figure 3.11 Time difference between DC and TFC as a function of ToF.

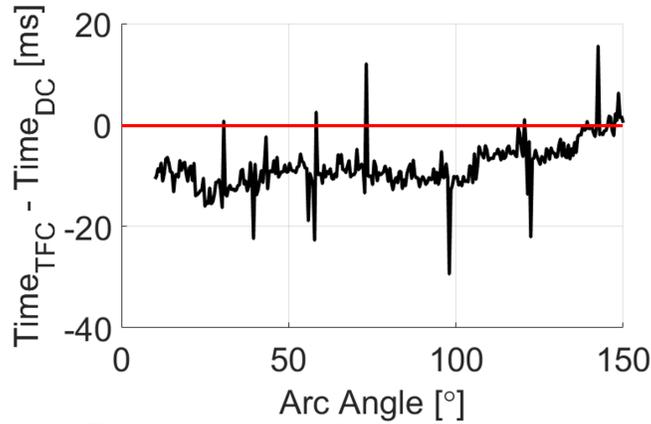


Figure 3.12 Time difference between DC and TFC as a function of transfer angle.

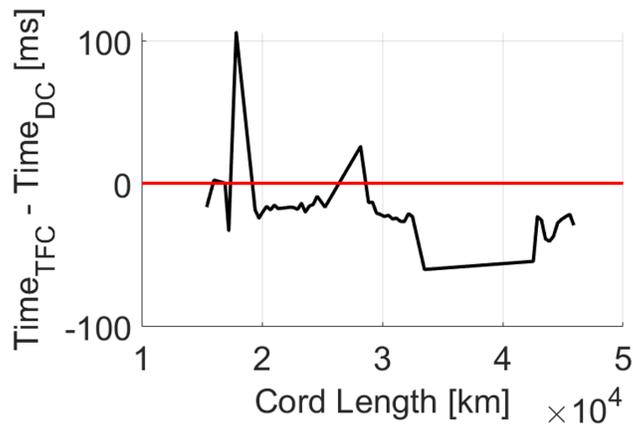


Figure 3.13 Time difference between DC and TFC as a function of cord length.

added to Eq. (33), as well as their partial derivatives to Eq. (50). This section introduces three scenarios where different perturbations are included. The perturbations are included separately, however they may be superimposed within the formulation. Two analyses are performed; first, the resultant perturbed orbit is compared to the unperturbed orbit generated via TFC, then the perturbed orbit is compared to a DC algorithm to determine TFC's ability to find an accurate solution. No computation time comparisons are performed in this section.

### 3.3.1 Earth's Oblateness

Orbits near Earth's surface are subject to the asymmetry of Earth's gravitational field. Although minor in small time scales, these effects stack over time, severely affecting the orbit. This pertur-

bation directly affects the RAAN and argument of periapsis depending on its inclination, altitude, and eccentricity [29]. This effect, commonly known as the  $J_2$  gravitational perturbation, is given by Eq. (3). The partial derivatives that constitute the Jacobian of this equation are:

$$\mathcal{J}_{J_2xx} = \frac{3J_2\mu r_{eq}^2}{2r^7} [5x^2(1 - 5(z/r)^2) - r^2(1 - 5(z/r)^2) - 10(xz/r)^2] \quad (67)$$

$$\mathcal{J}_{J_2xy} = \frac{15J_2\mu r_{eq}^2 xy}{2r^7} (1 - 7(z/r)^2) = \mathcal{J}_{J_2yx} \quad (68)$$

$$\mathcal{J}_{J_2xz} = \frac{15J_2\mu r_{eq}^2 xz}{2r^7} (3 - 7(z/r)^2) = \mathcal{J}_{J_2zx} \quad (69)$$

$$\mathcal{J}_{J_2yy} = \frac{3J_2\mu r_{eq}^2}{2r^7} [5y^2(1 - 5(z/r)^2) - r^2(1 - 5(z/r)^2) - 10(yz/r)^2] \quad (70)$$

$$\mathcal{J}_{J_2yz} = \frac{15J_2\mu r_{eq}^2 yz}{2r^7} (3 - 7(z/r)^2) = \mathcal{J}_{J_2zy} \quad (71)$$

$$\mathcal{J}_{J_2zz} = \frac{3J_2\mu r_{eq}^2}{2r^7} [5z^2(3 - 5(z/r)^2) - r^2(3 - 5(z/r)^2) - 10z^2(1 - (z/r)^2)] \quad (72)$$

A 150°, 77-min long arc is generated to test TFC's ability to find solutions under the effects of this perturbation. The semi-major axis of this orbit is 13,316 km, the eccentricity is 0.5, and the inclination is 50°. The choice of orbital parameters is meant to increase the effects of the perturbation. Although the magnitude of the perturbation is too small to visually notice in the duration of the transfer, Figure 3.14a shows that the perturbation exists. The  $y$ -axis corresponds to the difference between the perturbed and unperturbed solutions. Figure 3.14b shows the difference between TFC and DC. The error is minimal, proving TFC is capable of obtaining real solutions.

Due to the presence of oscillatory terms in Eq. (18), TFC is able to generate multi-revolution solutions. In unperturbed scenarios, this kind of solution is redundant, given that there is only one specific ToF that is able to find a solution. This is due to the periodic nature of the two-body problem. However, if a perturbation is added, the solutions become slightly more complex. Note that the ToF is still severely limited. In this investigation, multi-revolution solutions are found using the aforementioned  $J_2$  perturbation. To exaggerate the effects of this perturbing force, the  $J_2$  coefficient is increased by a factor of 10 (Figure 3.15). The figure is showing the difference between an unperturbed and a perturbed multi-revolution arcs. The periodicity of the orbit is clear

from the rise and fall of the error. In order to find a solution, the orthogonal polynomial used is of degree 80, in contrast to the degree 15 used in other scenarios.

### 3.3.2 Third-Body Perturbations

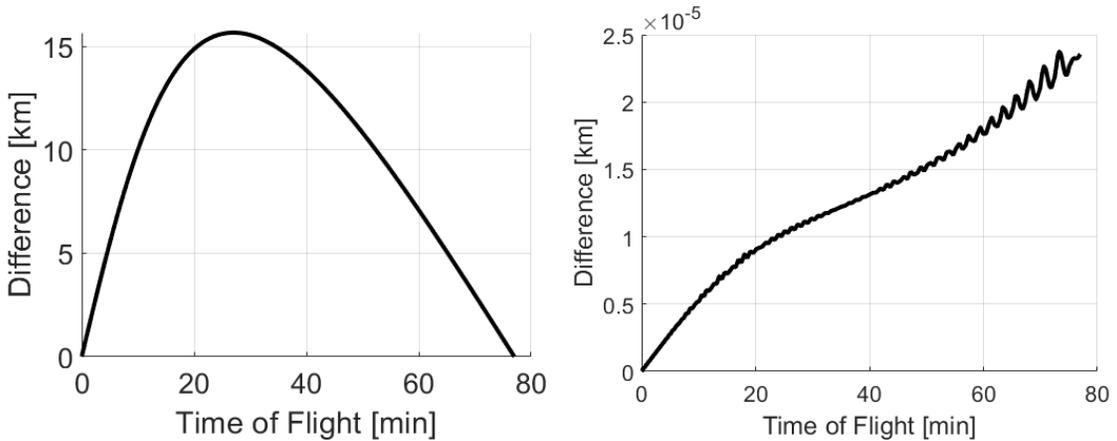
Solving Lambert's problem while considering the effects of the Moon's gravity is of particular importance as the Cislunar region becomes more populated [20]. The TFC framework allows for the inclusion of any number of perturbing bodies. In this investigation, the effects of the Moon are included (Eq. (4)). The partial derivatives of this perturbation are given by:

$$\mathcal{J}_{3BPxx} = \mu_{3b} \left( \frac{3x_{sc-3b}^2}{r_{sc-3b}^5} - \frac{1}{r_{sc-3b}^3} \right) \quad \mathcal{J}_{3BPxy} = \mu_{3b} \left( \frac{3x_{sc-3b}y_{sc-3b}}{r_{sc-3b}^5} \right) = \mathcal{J}_{3BPyx} \quad (73)$$

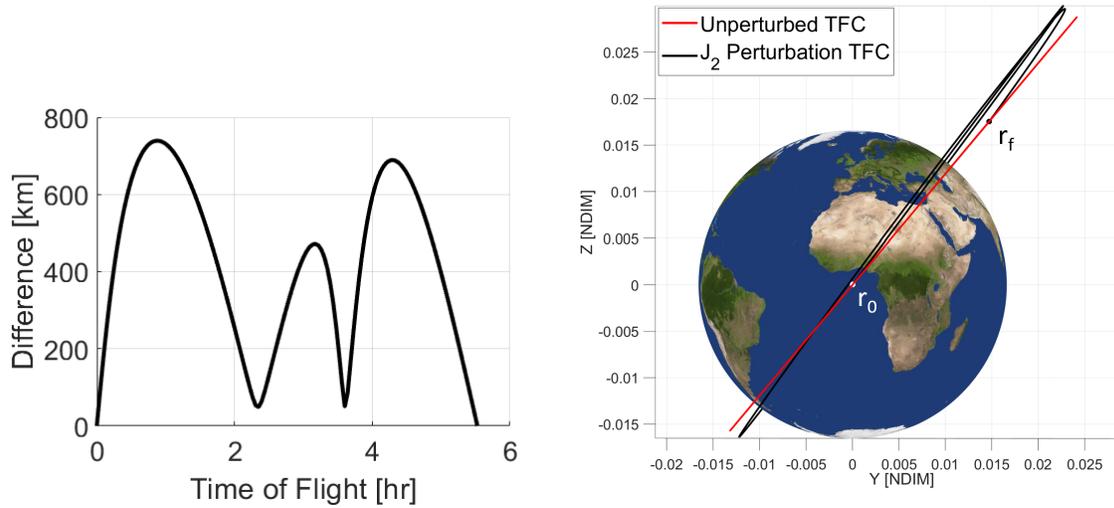
$$\mathcal{J}_{3BPyy} = \mu_{3b} \left( \frac{3y_{sc-3b}^2}{r_{sc-3b}^5} - \frac{1}{r_{sc-3b}^3} \right) \quad \mathcal{J}_{3BPxz} = \mu_{3b} \left( \frac{3x_{sc-3b}z_{sc-3b}}{r_{sc-3b}^5} \right) = \mathcal{J}_{3BPzx} \quad (74)$$

$$\mathcal{J}_{3BPzz} = \mu_{3b} \left( \frac{3z_{sc-3b}^2}{r_{sc-3b}^5} - \frac{1}{r_{sc-3b}^3} \right) \quad \mathcal{J}_{3BPzy} = \mu_{3b} \left( \frac{3y_{sc-3b}z_{sc-3b}}{r_{sc-3b}^5} \right) = \mathcal{J}_{3BPyz} \quad (75)$$

A 130° arc is generated to find a transfer from [0; -42164; 0] km to [291,644; 247,332; 0] km over a ToF of 70 hours, reaching a distance of approximately 13,000 km from the Moon. Note that the Moon must also be propagated, starting at [384,000; 0; 0] km. Figure 3.16 shows the results of the simulation, where the dotted line is the unperturbed arc, the solid line is the perturbed arc, and the right-most line is the orbit of the Moon. Figure 3.17a shows the difference between



(a)  $J_2$ -perturbed vs unperturbed Lambert arcs. (b) TFC vs differential corrections algorithms.  
 Figure 3.14 Error plots showing the norm of the vector difference between each state.



(a) Error plot between multi-revolution perturbed and unperturbed trajectories.

(b) Multi-revolution Lambert arc generated via TFC.

Figure 3.15 Multi-revolution solution trajectory and error plot.

the perturbed and unperturbed arcs. Figure 3.17b shows the difference between TFC and DC. As expected, the first plot shows a larger difference close to the Moon, as its gravitational influence increases. Most importantly, note that the boundary condition is still met at the end of the transfer. The second plot shows no significant difference between the TFC and DC solutions.

### 3.3.3 Solar Radiation Pressure

The final perturbation introduced in this investigation is the effects of SRP. Although the Sun's photons are massless, they are able to impart momentum to surfaces upon contact. The magnitude and direction of the change in momentum due to SRP is given by Eq. (5). The surface area of the chosen spacecraft is  $20,000 \text{ m}^2$ , the reflectivity coefficient  $\rho_s$  is 0.9, the absorption  $\rho_a$  and diffraction  $\rho_d$  coefficients are 0.1, the mass is 5 kg, and the surface area normal direction is held constant in the  $x$ -axis direction. The parameters are selected to exaggerate the effects of the SRP. The partial derivatives, simplified using the given parameters, are:

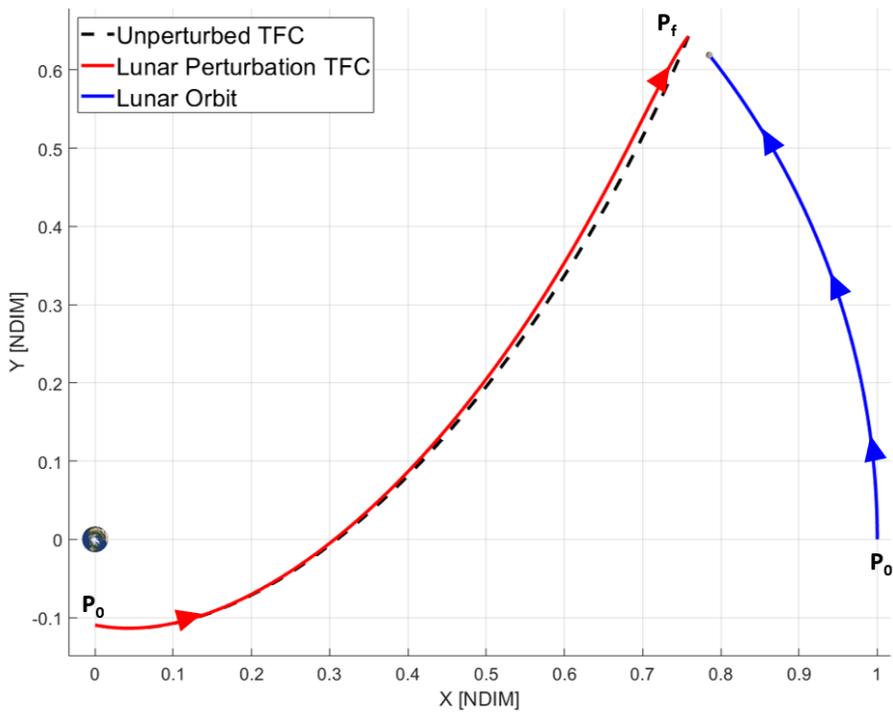
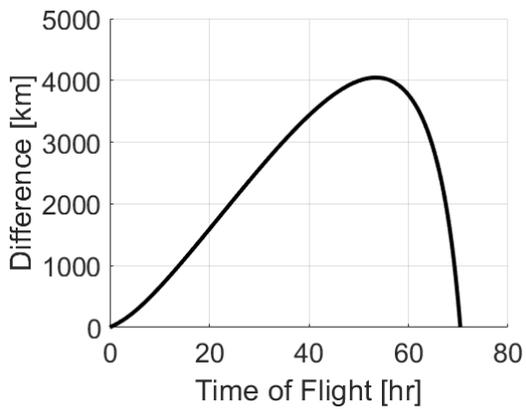
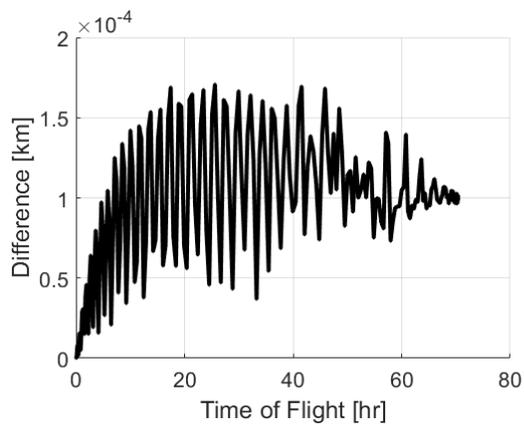


Figure 3.16 Perturbed Lambert arc under third-body perturbation.



(a) Third-body perturbed vs unperturbed Lambert arcs.



(b) TFC vs differential corrections algorithms.

Figure 3.17 Error plots showing the norm of the vector difference between each state.

$$\mathcal{J}_{SRPxx} = \frac{P_{SRPA}}{m} \left( \frac{2\rho_a x_s}{r_s^2} - \frac{2\rho_a x_s^3}{r_s^4} + \frac{4\rho_s x_s}{r_s^2} - \frac{4\rho_s x_s^3}{r_s^4} + \rho_d \left( \frac{x_s}{r_s^2} - \frac{2x_s^3}{r_s^4} + \frac{2}{3} \left( \frac{1}{r_s} - \frac{x_s^2}{r_s^3} \right) \right) \right) \quad (76)$$

$$\mathcal{J}_{SRPxy} = \frac{P_{SRPA}}{m} \left[ -\frac{2\rho_a x_s^2 y_s}{r_s^4} - \frac{4\rho_s y_s x_s^2}{r_s^4} - \frac{\rho_d x_s^2 y_s}{r_s^4} - \frac{\rho_d x_s y_s}{r_s^3} \left( \frac{2}{3} - \frac{x_s}{r_s} \right) \right] \quad (77)$$

$$\mathcal{J}_{SRPxz} = \frac{P_{SRPA}}{m} \left[ -\frac{2\rho_a x_s^2 z_s}{r_s^4} - \frac{4\rho_s z_s x_s^2}{r_s^4} - \frac{\rho_d x_s^2 z_s}{r_s^4} - \frac{\rho_d x_s z_s}{r_s^3} \left( \frac{2}{3} - \frac{x_s}{r_s} \right) \right] \quad (78)$$

$$\mathcal{J}_{SRPyx} = \frac{P_{SRPA}}{m} \left[ -\frac{2\rho_a y_s x_s^2}{r_s^4} - \frac{\rho_d y_s x_s^2}{r_s^4} + \frac{\rho_d y_s}{r_s} \left( \frac{1}{r_s} - \frac{x_s^2}{r_s^3} \right) \right] \quad (79)$$

$$\mathcal{J}_{SRPyy} = \frac{P_{SRPA}}{m} \left[ \frac{\rho_a x_s}{r_s^2} - \frac{2\rho_a x_s y_s^2}{r_s^4} + \frac{\rho_d x_s}{r_s} \left( \frac{1}{r_s} - \frac{y_s^2}{r_s^3} \right) - \frac{\rho_d x_s y_s^2}{r_s^4} \right] \quad (80)$$

$$\mathcal{J}_{SRPyz} = \frac{P_{SRPA}}{m} \left[ -\frac{2\rho_a x_s y_s z_s}{r_s^4} - \frac{2\rho_d x_s y_s z_s}{r_s^4} \right] = \mathcal{J}_{SRPzy} \quad (81)$$

$$\mathcal{J}_{SRPzx} = \frac{P_{SRPA}}{m} \left[ \frac{\rho_a z_s}{r_s^2} - \frac{2\rho_a x_s^2 z_s}{r_s^4} - \frac{\rho_d x_s^2 z_s}{r_s^4} + \frac{\rho_d z_s}{r_s} \left( \frac{1}{r_s} - \frac{x_s^2}{r_s^3} \right) \right] \quad (82)$$

$$\mathcal{J}_{SRPzz} = \frac{P_{SRPA}}{m} \left[ \frac{\rho_a x_s}{r_s^2} - \frac{2\rho_a x_s z_s^2}{r_s^4} + \frac{\rho_d x_s}{r_s} \left( \frac{1}{r_s} - \frac{z_s^2}{r_s^3} \right) - \frac{\rho_d x_s z_s^2}{r_s^4} \right] \quad (83)$$

The same Earth-Venus transfer from Figure 3.4 is used in this test. The effects of SRP on this spacecraft, although very minor, are clear from Figure 3.18. Figure 3.19a shows the magnitude of the effects, while Figure 3.19b validates the results using a DC algorithm.

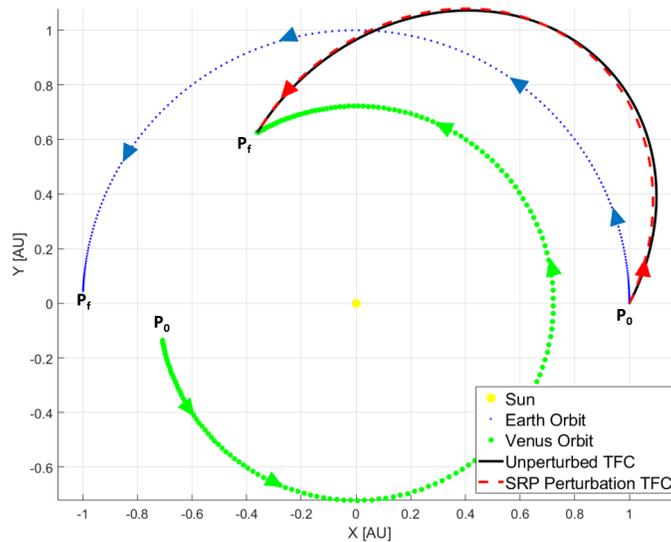
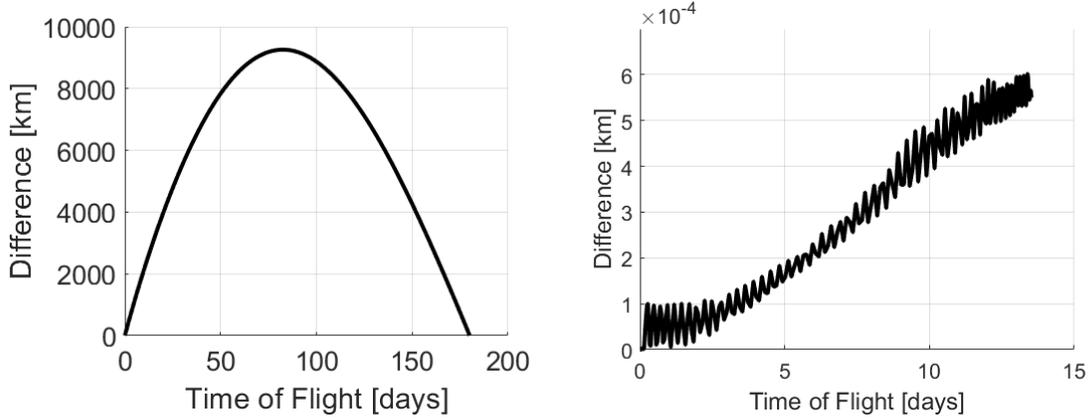


Figure 3.18 Perturbed Lambert arc under Solar radiation pressure.



(a) SRP vs unperturbed Lambert arcs.

(b) TFC vs differential corrections algorithms.

Figure 3.19 Error plots showing the norm of the vector difference between each state.

### 3.4 Polynomial Analysis

Certain scenarios tested in this investigation appear to converge differently due to a variety of reasons. Primarily, the magnitude of the initial and final positions appears to play a significant role. To solve this issue, most solutions are generated using non-dimensional units, where characteristic quantities vary depending on the context of the test. The reason for divergence lies in the degree and type of the polynomial chosen. For example, most testing scenarios around Earth are solved using non-dimensional units and a Legendre polynomial of degree 15. However, the Earth-Venus transfer, although non-dimensional as well, requires a degree of 150 to converge. This leads to an investigation into the effects of polynomial degree and type.

The Gegenbauer polynomial set is used to change polynomial type in a consistent manner. This set uses a selector variable,  $\alpha$ , to control the polynomial type independently of the degree in the following recurrence relation:

$$C_0^{(\alpha)}(x) = 1 \quad (84)$$

$$C_1^{(\alpha)}(x) = 2\alpha x \quad (85)$$

$$(n + 1)C_{n+1}^{(\alpha)}(x) = 2(n + \alpha)x C_n^{(\alpha)}(x) - (n + 2\alpha - 1)C_{n-1}^{(\alpha)}(x) \quad (86)$$

The Legendre polynomial corresponds to the Gegenbauer set with  $\alpha = 0.5$ ;  $\alpha$  may be incremented by a multiple of 0.5 to modify the polynomial type (minimum of  $\alpha = -0.5$ ). The subscript of each

constant  $C_n$  represents the degree of the polynomial; for a second-degree polynomial,  $n = 1$ .

To analyze the effects of varying the polynomial type and degree on convergence, a test similar to that shown in Figure 3.13 is conducted; the angle is fixed at  $120^\circ$  and the ToF is fixed at 2.5 hours, while the trajectory is varied by changing the final radius (8378 km - 42164 km). In addition, this test is conducted for multiple combinations of degree and  $\alpha$ ; the degree ranges from 20 to 40, and  $\alpha$  ranges from 0.5 to 10. (Figures 3.20, 3.21, 3.22). Note that the  $z$ -axis of Figure 3.22 measures the ratio of the final and initial radii, and the color bar measures the error of the trajectory. This error is calculated by propagating the output initial velocity from TFC with a two-body problem model (Eq. (2)), and then taking the difference between expected and propagated final position. The data set's maximum error is much higher than the color bar's maximum, which was chosen for clarity in certain trends.

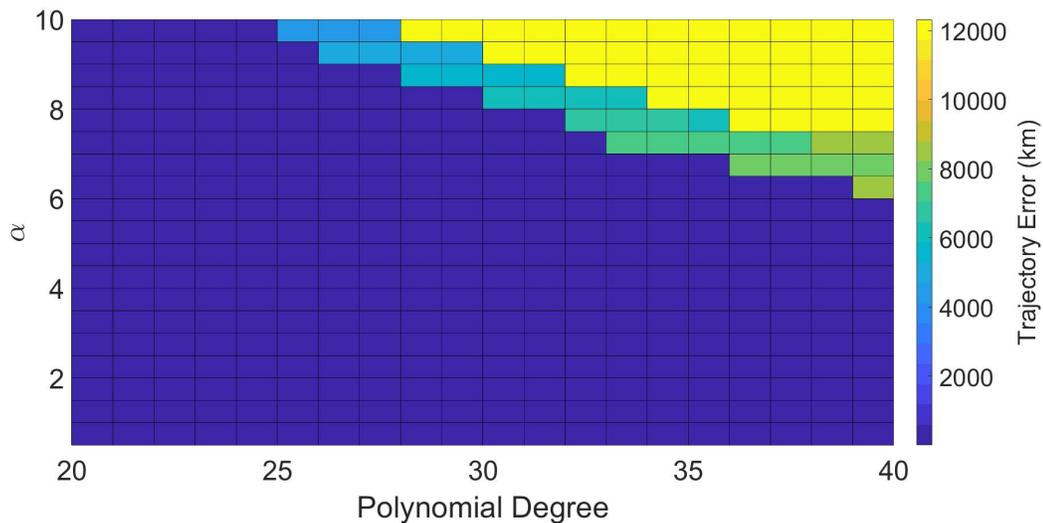


Figure 3.20 Degree- $\alpha$  plot at  $r_f/r_0 = 1$  (8378 km) showing the expected trend.

Figures 3.20 and 3.21 show expected behavior and errors, respectively; it is expected that most degree- $\alpha$  pairings would result in relatively low error, with the error increasing as the polynomial becomes overly complex (the convergence algorithm is limited to 200 iterations in this test). However, at certain settings, many pairings result in failures to converge, leading to extremely high

error. As Figure 3.22 shows, this behavior is erratic and unpredictable, but limited to certain bands. Additionally, certain degree- $\alpha$  pairings result in reduced quantities of errors, potentially avoiding them entirely. However, because the polynomial outputs are not the sole constraint determining failure or success in convergence, the problem may be solved entirely by other means.

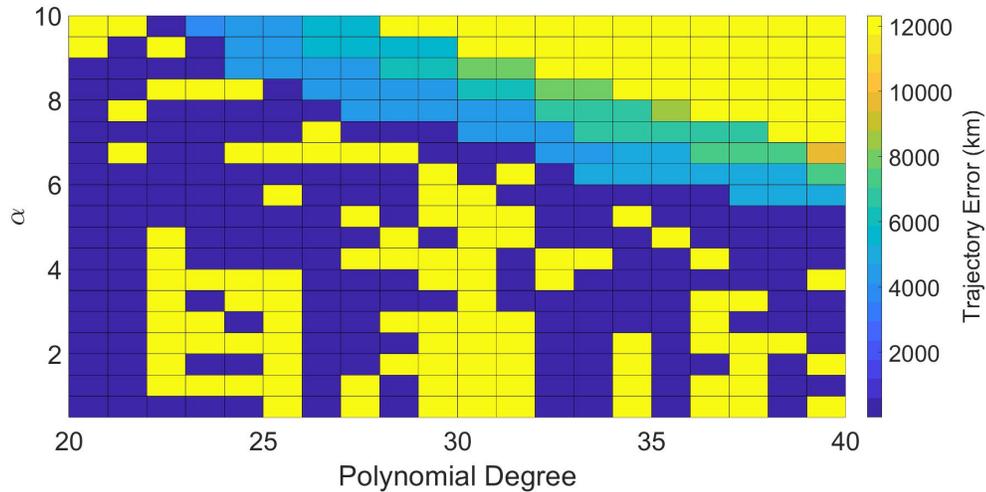


Figure 3.21 Degree- $\alpha$  plot at  $r_f/r_0 = 2.32$  (19410 km) showing failures to converge.

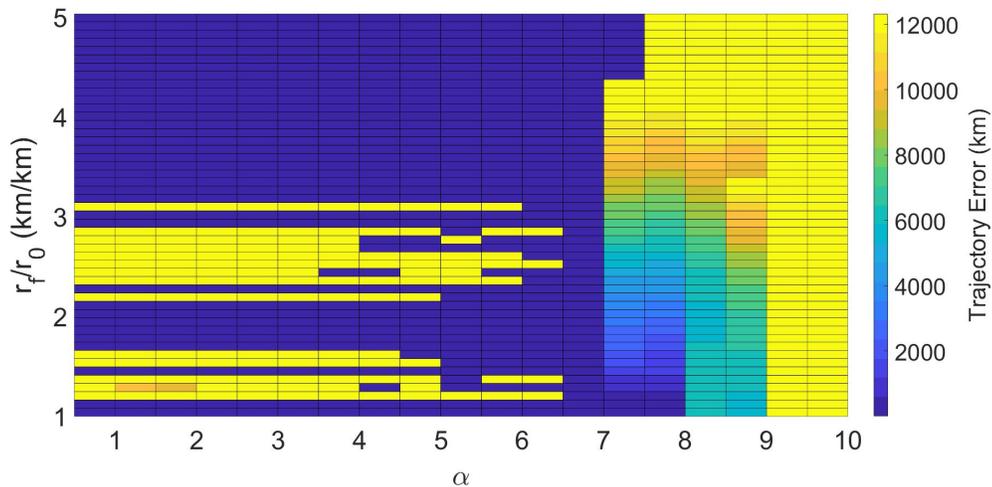


Figure 3.22  $\alpha$ -radius plot at degree 30 showing sections of concentrated errors.

## 4 Constellation Management System

This section presents the structure of the CMS, focusing on automation of the framework. Special focus is given to the structure of the simulator, as it is the goal of this thesis. The scheduler is discussed briefly to justify the structure of the simulator itself.

The CMS is designed to be fully automated, generating the most efficient and optimized task plan. It consists of a modular system with a simulator and a scheduler: the Mission Planning System (MPS). This system ensures efficient and automated data transfer. Figure 4.1 represents the structure and data flow of the CMS integrated into an NTN. The dotted rectangular boxes represent the three different data types that flow within the network: user, control, and operation. User and operation data are common in typical networks and flow from user equipment into the NTN and from the NTN into the spacecraft, respectively. The CMS requires a new type of data called control data. It is used by the CMS to input and output the necessary information from the constellation and users.

The large rectangular box represents the NTN. The Core Network (CN) is typical within telecommunications network and provides a path for the data to flow within the network. The CMS itself is composed of the MPS, the Mission Provider Aggregator (MPA), and the Ground Station Aggregator (GSA). The latter two correspond to interfaces that facilitate transfer of data from the respective nodes (Figure 4.1). The Mission Provider (MP) is the satellite operator, while the Mobile Network Operator (MNO) is the telecommunications provider. User equipment (UE) and GS are nodes that receive or output data into the network. When a satellite connects to a UE, user data flows between them. The satellite keeps this data (this is known as store-and-forward data transfer) until it is able to connect with a GS. At this instance, the user data is delivered to the GS and new user data is received by the satellite. Simultaneously, commands from the MP are transferred via operation data, as well as control data from the CMS. Within the ground network, the CMS is perpetually generating a new task plan and sending it to a GS in the form of control data via the GSA.

The MPS receives telemetry from the constellation and develops the most up-to-date task plan.

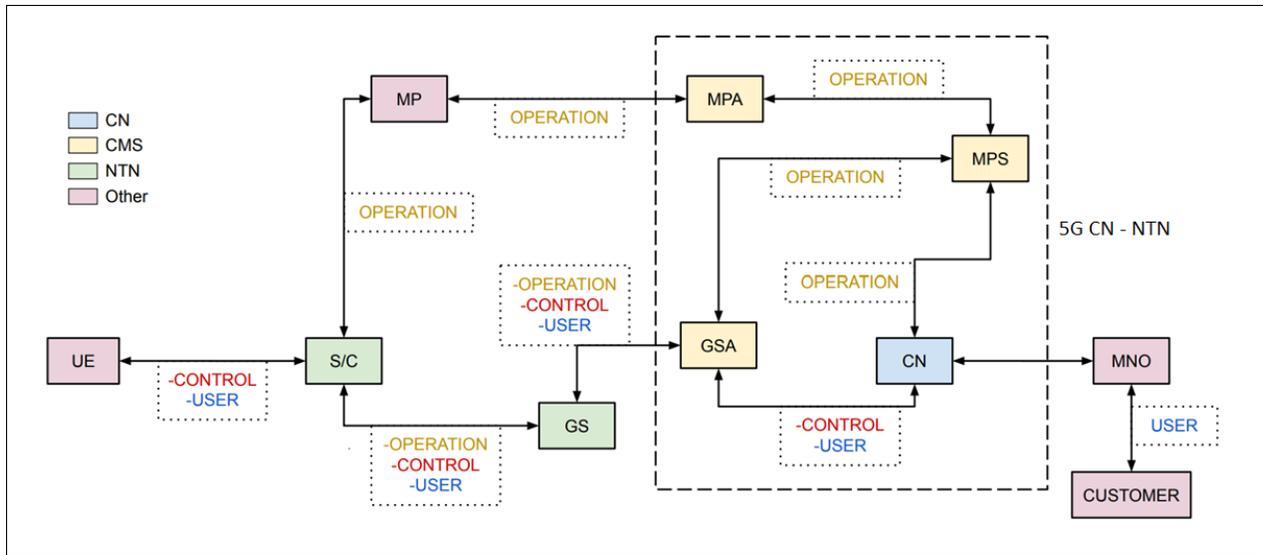


Figure 4.1 Constellation management system architecture.

The current development of the CMS considers connectivity to GS, UE, TA, ISL, and the Sun. The algorithm allows these locations to be replaced, thus being versatile to constellations of any size and type. A WoC is found when the FOV of a GS, UE, or a satellite is within the FOV of another satellite. The duration of Sun's WoC are used to determine an estimate for the energy generation via Solar panels. To increase the reliability of the generated task plan, link budget considerations are incorporated. Link budget refers to the signal power arriving at the receiving end of a connection after taking into account gains and losses. For example, although two satellites might be in view of one another, there may be too far to share data. Similarly, atmospheric losses may block the signal from arriving at a GS. This thesis considers link budget calculations only for ISL events, and only takes into account the distance between the satellites. The result is directly integrated into the duration of the WoC. Development of an accurate simulator is important to increase the reliability of the task plan generated by the scheduler. However, state-of-the-art schedulers and simulators perform similar functions. The key distinction is how the simulator is integrated into the CMS.

The structure and data flow of the CMS are designed with automation in mind. Most importantly, data flow between the simulator and scheduler occurs autonomously with little human involvement (Figure 4.2). The algorithm initiates by importing inputs from MongoDB. It reloads these inputs

in every iteration to accommodate any new constellation data that the user might add at any point. Once the information is loaded, the simulator generates the WoC specified by the user, notably GS, UE, TA, ISL, and Sun visibility events. The resulting WoC are transferred to the scheduler via an API. The scheduler then generates the optimal task plan and communicates it to the spacecraft via the CN. The entire CMS operates on an online server. The simulator runs on a schedule and is constantly uploading updated WoC to the InfluxDB database. Thus, the process is automated with little human involvement other than adding new constellation data. The remainder of this section details the structure and procedures included in the simulator, and briefly discusses the scheduler.

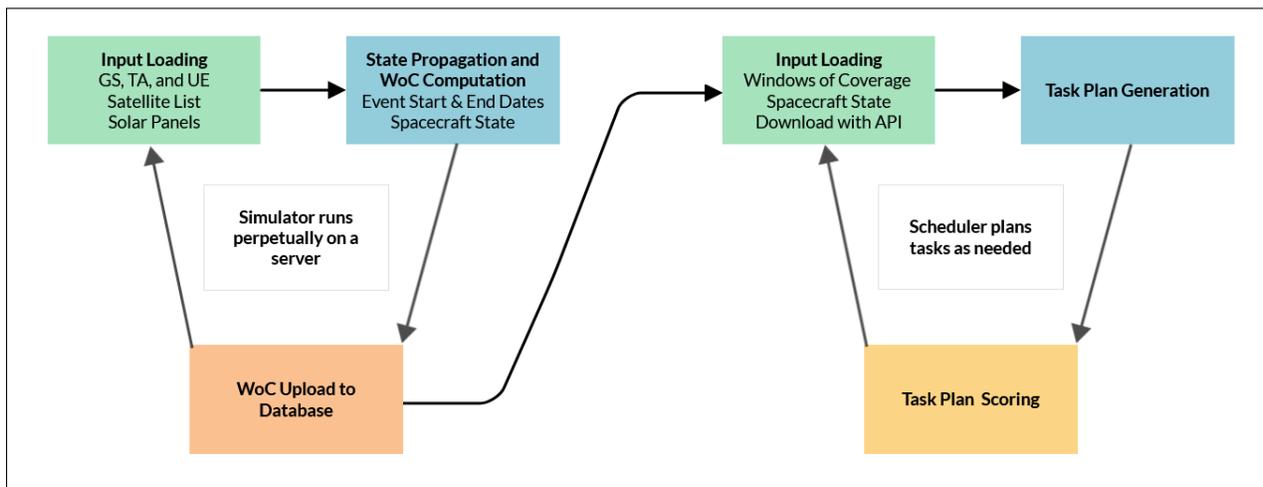


Figure 4.2 Constellation management system data flow.

#### 4.1 Orbital Simulator Structure

The MPS' simulator is designed in JAVA and uses *Orekit*, an open-source library. Utilizing JAVA enables the integration of the simulator with the scheduler, while *Orekit* offers functionalities for the simulator to facilitate mission analysis features. Distinctly segregating the simulator from the scheduler simplifies operations and permits automation; the simulator operates continuously as the scheduler formulates task plans. The simulator's workflow is divided into three major sections: input loading, WoC computation, and output generation and formatting (Figure 4.3). The system first loads the list of satellites, UE, GS, and TA coordinates, the satellites' battery life and charging

capabilities, and the satellites' FOV for processing. There are multiple ways for the simulator to propagate satellite data, including SGP4 and classical propagators. For this thesis, the SGP4 and 2-body propagators are used. If using real satellite data instead of orbital elements, the TLE is first retrieved from CelesTrak. Then, the simulator propagates the constellation to calculate crucial tasks, including the satellite's WoC, Sun charging, and eclipse conditions. The outputs are uploaded to a database, and optionally a XML or JSON file is saved useful for analysis. An API is available for the scheduler to retrieve WoC data for any chosen time period.

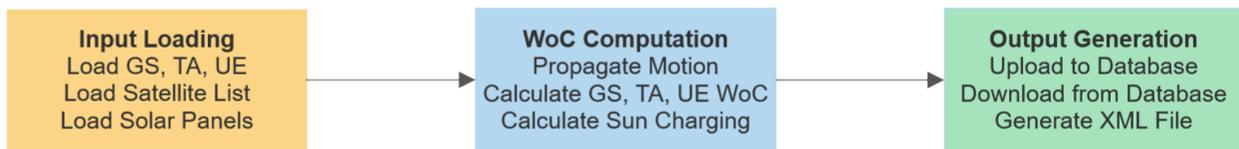


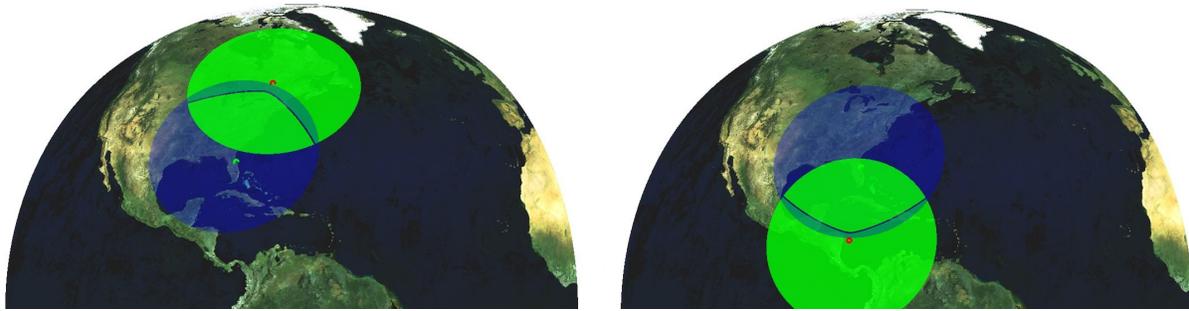
Figure 4.3 Simulator workflow structure.

#### 4.1.1 Input Loading

The simulator inputs are obtained from MongoDB via its API. The full list of inputs is reflected in Table 4.1. They are organized in discrete objects, which the simulator interprets and retrieves necessary information. A satellite object contains a name, power subsystem information, and FOV. Location objects, i.e., GS, UE, and TA, consist of latitude, longitude, and elevation angle information. The elevation angle is complimentary to the location's FOV. Automating this process allows for more flexibility in the system and reduces the need for user input.

Table 4.1 Simulator inputs.

Satellite List	TA List
GS List	UE List
GS/UE half-FOV	Antenna half-FOV
Spacecraft Attitude	Solar Panel Properties
Battery Voltage	Simulation Time



(a) Instant of WoC beginning.

(b) Instant of WoC end.

Figure 4.4 Florida GS example WoC.

#### 4.1.2 Windows of Coverage Computation

After loading the inputs, the simulator propagates the satellites and computes the WoC for the satellites in the constellation. The procedure is broken down further into two steps: propagation and event detection. First, all satellites are propagated and their data is retrieved. If real satellite data is used, the TLE is retrieved from Celestrak and it is propagated using SGP4. Otherwise, the orbital elements provided in the satellite object from MongoDB are propagated using a 2-body problem propagator. Event detectors are also added to the propagators. Events are defined by first contact, last contact, and the *s/c* state at each time-step during the event. Visibility between UE and GS is established when both the location and the satellite are within each other's FOV (Figure 4.4). To create a TA, several locations on the area's border are picked. These points form the vertices of an  $N$ -sided polygon, where  $N$  represents the number of cities. The larger the  $N$ , the smoother the edges of the TA become. When any portion of the TA is within the *s/c*'s FOV, an event is triggered. Inter-satellite link events occur when a body is no longer in between the two *s/c*. Including a link budget calculation reduces this event by adding a distance requirement. In reality, multiple factors are necessary for higher-fidelity calculations, including a variable FOV, signal losses, and line-of-sight direction. These considerations are not included in this thesis, but will be added in future iterations of the algorithm.

Sun visibility events are much simpler, as the only thing that has to be considered is the direct line-of-sight between the celestial body and the *s/c*. The orientation of the solar panel is considered

as well as the exact distance from the s/c to the Sun for higher fidelity calculations. The power per unit area at the location of the s/c is a function of the luminosity and distance to the Sun:

$$P = \frac{L}{4\pi|\mathbf{r}_S|^2} \quad (87)$$

where  $L$  correspond to the Sun's luminosity and is given by  $L = 3.828 \times 10^{26}$  W [64], and  $\mathbf{r}_S$  is the position vector of the Sun relative to the s/c. The variable  $P$  represents the power per unit area at the location of the satellite, but not all of this radiation is absorbed by the solar panel.

The final power that the s/c receives from the Sun depends on the angle  $\theta$  that relates the direction normal to the satellite's solar panel and  $\mathbf{r}_S$ . The total power absorbed,  $P_{abs}$ , is calculated as:

$$P_{abs} = PA\eta \cos \theta \quad (88)$$

where  $\eta$  is the quantum efficiency [65] and  $A$  is the area of the solar panel. The quantum efficiency is defined as the amount of photons converted into electrons within the system.

Since the duration of the event varies with each satellite, calculating the average charge is not accurate. Instead, the total period is broken down into small time steps,  $\delta t$  (smaller  $\delta t$  increases precision at the expense of increasing run-time). The algorithm uses the voltage of the on-board batteries to convert the energy received [65]:

$$Q = \sum \frac{LA\eta \cos \theta}{4\pi|\mathbf{r}_S|^2 V} \delta t \quad (89)$$

The total charge for each event,  $Q$ , is added over the duration of the event.

The WoC for all event types must be adjusted if they occur during propagation start or ending. That is, if the simulation begins while a WoC is underway, then the beginning of the WoC is marked with the start of propagation. Similarly, if the simulation ends while a WoC is underway, then the end of the WoC is marked with the end of propagation. This is done for all event types, including the Sun and ISL. In the case of ISL, an additional step involves verifying the distance between two

satellites during the WoC, adjusting the duration based on a specified minimum distance. This step constitutes a portion of a link budget. More detailed considerations for advanced link budget models exist, but are not included here.

### 4.1.3 Output Generation and Formatting

Finally, the simulator transmits the results to InfluxDB. The database is designed to function in real-time, significantly enhancing the scheduler's ability to perform data read and write operations. This efficiency is further supported by a JAVA library, which is not only readily accessible but also open-source. The integration of this library with the simulator is seamless, ensuring smooth and efficient operations. The WoC are stored until the scheduler's API retrieves them. The flow of data is summarized in Figure 4.5. Note that the process highlighted in this section is subject to change as the system requirements are updated over time. For instance, future work includes completely replacing InfluxDB in favor of MongoDB for both inputs and outputs.

## 4.2 Scheduler Structure

Given that the scheduler is not the focus of this thesis, only a short discussion of its structure is provided next. The scheduler is one of the key components of the CMS and the core of logic and intelligence of the whole system. It is also located within the MPS. It receives its inputs from the operator in the form of a task plan, and the aforementioned WoC from the simulator. It is divided into a scheduling engine which generates the task plan and a score calculation module given a set of constraints and criteria. It uses an incremental approach to increase performance.

The CMS supports a variety of optimization algorithms ranging from exhaustive search to more advanced AI algorithms. The implemented solver is composed of a construction heuristics (CH) algorithm to initialize the solution followed by a local search (LS) phase. The CH phase allows the solver to quickly obtain a good initial solution to start the LS phase, which improves the solution



Figure 4.5 Simulator data flow.

score until the termination criteria is met. The combination of these two phases results in the best compromise between quality of the solution and scalability [66]. The supported CH algorithms are “first fit decreasing” [67], “strongest fit” and “weakest fit” [68] among others. Additionally, the supported LS algorithms correspond to “tabu search” algorithms [69, 70], “late acceptance” [71], and “step counting hill climbing” [72]. The optimization algorithms base the move selection and optimization step for the next iteration on the solution score. There are different types of constraints in an optimization problem and, as such, the score calculation has to be able to model them. The constraints of the use case studied in this work are divided into three separate groups:

- *Feasibility Constraints*: These dictate whether or not a given solution is feasible. These include physical, geometrical and temporal constraints of the domain.
- *Business Constraints*: They represent the highest level of optimization criteria, usually tied to the business goals of the constellation operator. An example of such a constraint is to assign the maximum number of tasks to each satellite or to prioritize some task type like data download, following the concept of utilities [73].
- *Operation Constraints*: These represent the remaining optimization criteria once the business ones have been satisfied. They improve the overall operation of the constellation, maintaining high battery levels and low levels of on-board memory used.

This choice of constraints is sufficient to realistically model constellation scheduling requirements. These may be supplemented or changed in future prototypes.

## 5 Orbital Simulator Results

This section discusses various scenarios and their results for the orbital simulator introduced earlier. First, three primary scenarios demonstrate the simulator’s capabilities. The WOC results for GS, UE, and TA, are then presented and compared with those obtained from STK. Subsequently, the total duration of solar charging is computed, and eclipse durations are compared with STK data. Finally, the applicability of ISL is assessed in relevant scenarios.

Scenario 1 contains only one satellite, thus ISL are not possible. If link budget calculations are incorporated in the remaining scenarios, it is anticipated that the duration of ISL events will decrease; therefore, a comparison of these results is conducted. Additionally, an extra scenario is introduced to test the algorithm’s ability to determine the WoC for a trajectory obtained via TFC. such a trajectory corresponds to a Lunar transfer (Figure 3.3). Finally, the effects of including a UKF are analyzed.

### 5.1 Testing Scenarios

Telecommunications constellations typically contain dozens of satellites all around the world. To simulate results for a constellation like this, three scenarios are presented in this thesis, with varying number of locations and satellites. The parameters found in Table 5.1 are common to all scenarios and satellites. These parameters are determined from values used within LEO telecommunications satellites.

*Table 5.1* Scenarios common parameters.

<b>Input Type</b>	<b>Value</b>
GS half-FOV	70°
UE half-FOV	55°
Antenna half-FOV	60°
Spacecraft Attitude	Nadir Pointing
Panel Efficiency	25%
Panel Area	0.3 m <sup>2</sup>
Battery Voltage	24 V
Panel Normal Direction	(-1 0 0)

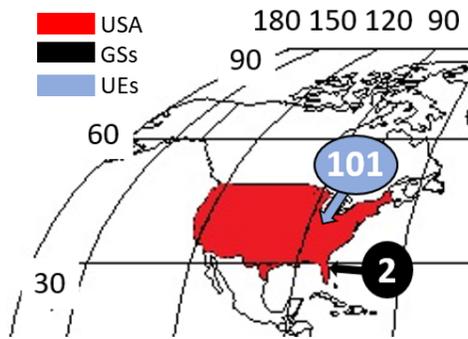


Figure 5.1 Scenario 1 map.

### 5.1.1 Scenario 1

The first scenario involves a simple example with one GS, one TA, one UE, and one satellite. Since no other satellites are present, no ISL are computed. Figure 5.1 shows the important locations in this scenario. Note that, in order to increase speed and efficiency of the algorithm, the TA are broken down into two sub-areas: USA-East and USA-West. Results are shown separately for each sub-area. The numbers represent the ID for the particular location and are common to all scenarios.

### 5.1.2 Scenario 2

The second scenario presented is more complex than scenario 1, consisting of four satellites, two GS, two TA, and ten UE. The ISL are possible in this scenario and are included in the results. Table 5.2 includes the coordinates of the locations, and Figure 5.2 shows them on a map including the TA.

### 5.1.3 Scenario 3

The final scenario substantially increases the complexity relative to scenario 2 (Table 5.3). This time, sixteen satellites are included, alongside five GS, thirteen UE, and two TA (Figure 5.3). A set of orbital elements are used to define initial conditions, rather than TLEs to demonstrate that the simulator works with different propagator models.

Table 5.2 Scenario 2 location coordinates.

Location	Latitude	Longitude	Half-FOV
GS2	29.0	-81.0	70°
GS11	69.65	18.96	70°
UE101	35.32	-91.14	55°
UE102	38.78	-106.41	55°
UE103	33.07	-110.97	55°
UE104	47.46	-116.18	55°
UE105	44.09	-74.67	55°
UE106	39.14	-96.11	55°
UE107	38.78	-4.34	55°
UE108	42.07	-7.23	55°
UE109	41.45	2.12	55°
UE110	39.78	2.89	55°

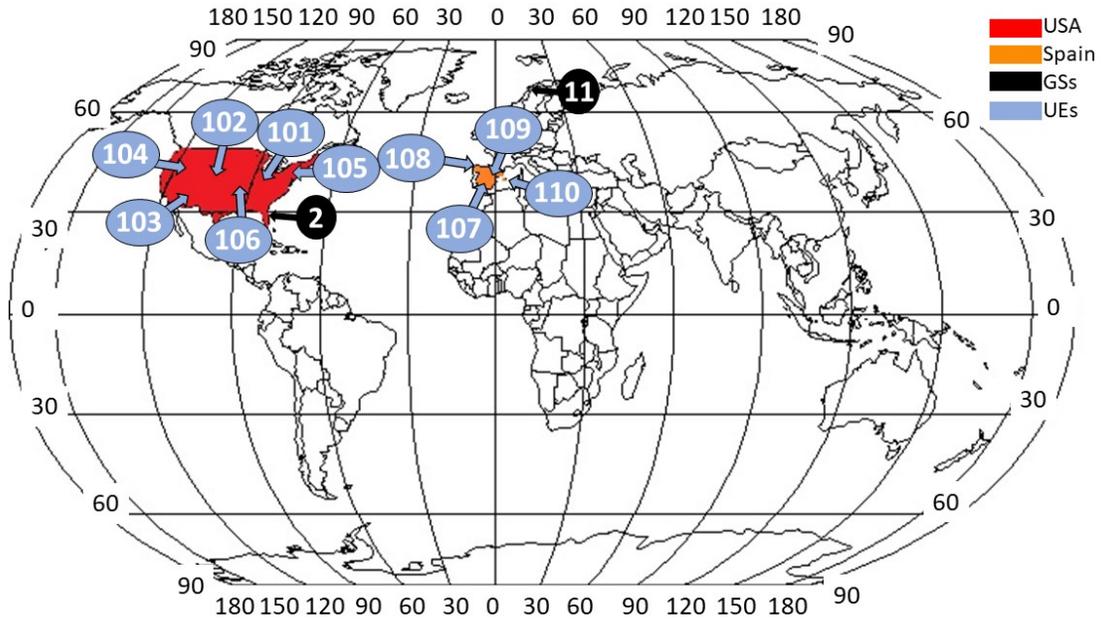


Figure 5.2 Scenario 2 map.

## 5.2 Simulator Results

The results in this section present the difference between the CMS' simulator and STK. Displaying the duration and start/end times of the WoC is not itself representative of the effectiveness of the simulator. For that reason, the results are shown as a  $\Delta T$  rather than as absolute time. Given

*Table 5.3 Scenario 2 location coordinates.*

<b>Location</b>	<b>Latitude</b>	<b>Longitude</b>	<b>Half-FOV</b>
GS1	68.33	-133.61	70°
GS3	-52.93	-70.85	70°
GS4	-33.95	18.43	70°
GS5	-33.75	151.18	70°
GS11	69.65	18.96	70°
UE101	35.32	-91.14	55°
UE102	38.78	-106.41	55°
UE103	33.07	-110.97	55°
UE104	47.46	-116.18	55°
UE105	44.09	-74.67	55°
UE106	39.14	-96.11	55°
UE107	38.78	-4.34	55°
UE108	42.07	-7.23	55°
UE109	41.45	2.12	55°
UE110	39.78	2.89	55°
UE111	-31.95	116.13	55°
UE112	23.50	54.79	55°
UE113	-8.70	-64.42	55°

that the results are consistent amongst the various locations, only results for one location of each type is shown. For Sun charging events, the same  $\Delta T$  value is shown, alongside the battery charge generated, computed with Eq. 89. Two values are shown for the ISL events, one considering link budget and one without, i.e., one only considers line-of-sight and the other the distance between the two spacecraft. The satellites chosen for these simulations belong to the LEMUR-2 telecommunications constellation. They are a LEO DSS, thus represent what a typical NTN looks like. A final scenario is used to represent the algorithm’s success at detecting events in a substantially different type of trajectory: a Lunar transfer.

### **5.2.1 Scenario 1 Results**

The satellite selected for scenario 1 is LEMUR-2-JENNIFERSONG. The TLE is retrieved from CelesTrak by using the following hyperlink: “<https://celestrak.org/NORAD/elements/gp.php?NAME=LEMUR-2-JENNIFERSONG&FORMAT=tle>”. This process is automated, such that inputting the satellite’s name into the simulator will yield the correct TLE without the need of

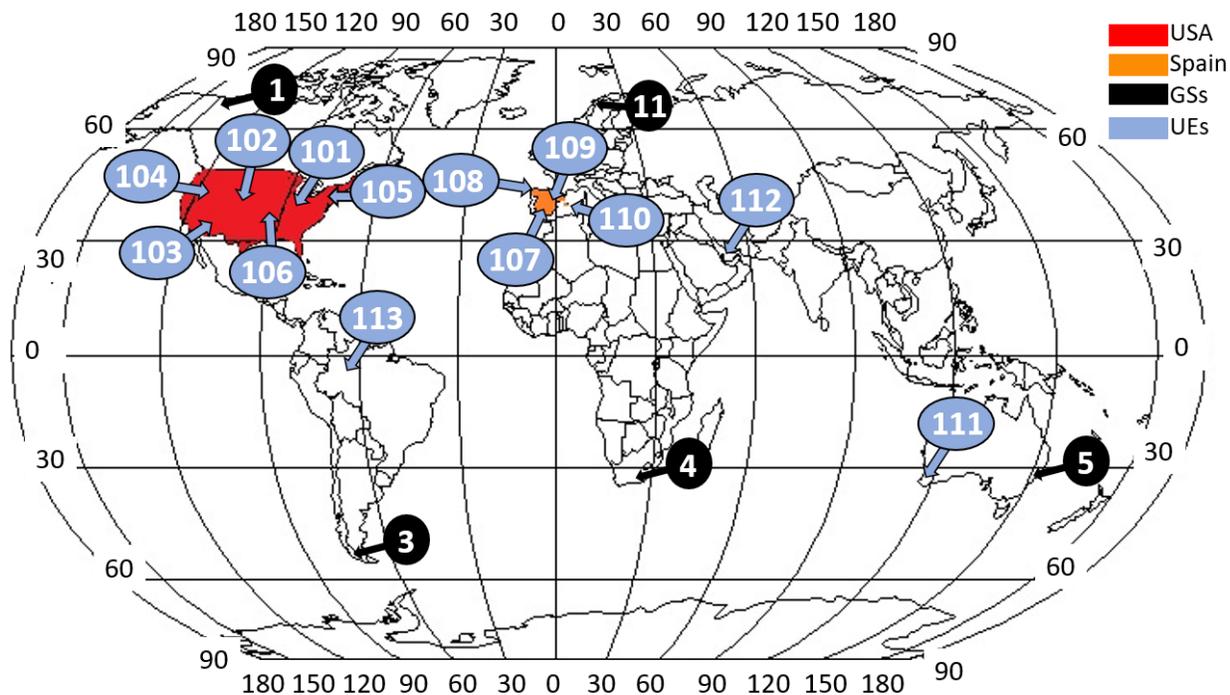


Figure 5.3 Scenario 3 map.

manually locating the telemetry. The date for the obtained TLE is 2023-10-19T04:18:18.878688Z, and the simulation time is 4 days. Tables 5.4- 5.6 show the results for GS, UE, and TA compared to STK for beginning and ending of the events. The results show that there is little difference between the results obtained from STK, and those obtained by the CMS simulator module. Table 5.7 shows similar results, including the generated charge for the WoC. Note that only a handful of events are chosen out of the total.

Table 5.4 Scenario 1: GS2 WoC for LEMUR-2-JENNIFERSONG.

Date	Event Start $\Delta T$	Event End $\Delta T$
2023-10-19	1.0548 s	0.6573 s
2023-10-21	0.6613 s	1.005 s
2023-10-21	0.990 s	0.674 s
2023-10-22	1.163 s	0.391 s
2023-10-23	0.623 s	1.078 s

Table 5.5 Scenario 1: UE101 WoC for LEMUR-2-JENNIFERSONG.

Date	Event Start $\Delta T$	Event End $\Delta T$
2023-10-20	0.011 s	0.004 s
2023-10-20	0.010 s	0.012 s
2023-10-22	0.010 s	0.005 s
2023-10-22	0.007 s	0.006 s

Table 5.6 Scenario 1: Spain Northeast WoC for LEMUR-2-JENNIFERSONG.

Date	Event Start $\Delta T$	Event End $\Delta T$
2023-10-19	1.904 s	1.908 s
2023-10-20	1.900 s	1.776 s
2023-10-21	1.882 s	1.861 s
2023-10-22	1.981 s	1.860 s
2023-10-23	1.249 s	2.035 s

Table 5.7 Scenario 1: Sun-charging WoC for LEMUR-2-JENNIFERSONG.

Date	Battery Charge	Event Start $\Delta T$	Event End $\Delta T$
2023-10-19	3.7697 A·hr	0.005 s	0.002 s
2023-10-20	3.7268 A·hr	0.007 s	0.001 s
2023-10-21	3.6948 A·hr	0.013 s	0.009 s
2023-10-22	3.6528 A·hr	0.002 s	0.003 s

### 5.2.2 Scenario 2 Results

Four satellites from the LEMUR-2 constellation are selected for this scenario: LEMUR-2-SATCHMO, LEMUR-2-SAM-AMELIA, LEMUR-2-OSCARLATOR, and LEMUR-2-HIMOMANDDAD. Although all satellites are used to compute the WoC, results are only shown for LEMUR-2-SATCHMO. The TLE obtained has the following epoch: 2023-10-20T04:13:06.295584Z. Tables 5.8- 5.11 show the results for GS, UE, TA, and the Sun-charging WoC. The results, although mostly similar to scenario 1, have higher difference than expected. This could be due to a variety of reasons, however the most likely explanation is that the tolerance used within the Orekit library varies with respect to that of STK. Regardless, the results are close enough as to validate the simulator. Table 5.12 shows the results for the obtained ISL, with each cell containing the start (top) and end (bottom) times. The dates shown correspond to the WoC obtained by observing

LEMUR-2-SAM-AMELIA. The limiting range applied to the link budget is set to 4300 km, as this is the approximate distance at which the satellites were in contact. The reason for this is to demonstrate how a WoC can change, although subtly at this range, if considering distance as well as line-of-sight.

*Table 5.8 Scenario 2: GS11 WoC for LEMUR-2-SATCHMO.*

<b>Date</b>	<b>Event Start <math>\Delta T</math></b>	<b>Event End <math>\Delta T</math></b>
2023-10-20	0.101 s	0.761 s
2023-10-21	0.167 s	0.677 s
2023-10-22	0.188 s	0.633 s
2023-10-23	0.341 s	0.614 s

*Table 5.9 Scenario 2: UE102 WoC for LEMUR-2-SATCHMO.*

<b>Date</b>	<b>Event Start <math>\Delta T</math></b>	<b>Event End <math>\Delta T</math></b>
2023-10-22	0.926 s	71.803 s
2023-10-23	0.867 s	65.264 s

*Table 5.10 Scenario 2: USA East WoC for LEMUR-2-SATCHMO.*

<b>Date</b>	<b>Event Start <math>\Delta T</math></b>	<b>Event End <math>\Delta T</math></b>
2023-10-20	2.677 s	123.94 s
2023-10-21	3.329 s	1.094 s
2023-10-22	179.838 s	215.811 s
2023-10-23	22.591 s	1.249 s

*Table 5.11 Scenario 2: Sun-charging WoC for LEMUR-2-SATCHMO.*

<b>Date</b>	<b>Battery Charge</b>	<b>Event Start <math>\Delta T</math></b>	<b>Event End <math>\Delta T</math></b>
2023-10-19	3.7697 A·hr	0.001 s	0.006 s
2023-10-20	3.7268 A·hr	0.010 s	0.005 s
2023-10-21	3.6948 A·hr	0.008 s	0.001 s
2023-10-22	3.6528 A·hr	0.003 s	0.015 s

*Table 5.12 Scenario 2: ISL WoC for LEMUR-2-SATCHMO.*

<b>Date</b>	<b>Line-of-Sight Only</b>	<b>Link Budget</b>
2023-10-20	06:38:41.080 07:08:54.955	06:39:41.080 07:07:54.955
2023-10-20	07:30:17.121 07:50:05.473	07:30:17.121 07:49:05.473
2023-10-20	08:21:00.036 08:30:56.028	08:24:00.036 08:27:56.028

### 5.2.3 Scenario 3 Results

In order to demonstrate that the simulator operates with a variety of dynamical models, this last scenario consists of sixteen satellites defined by their orbital elements. The four satellites are found in four circular and polar orbits, with four satellites in each orbital plane. The planes' RAAN changes by 90°, i.e., 0°, 90°, 180°, and 270°, and the satellites of each plane vary in argument of perigee also by 90°. The satellites are identified by integer numbers 1-16. The results (Tables 5.13-5.17) are consistent with the previous two scenarios. Obtained WoC are similar to STK as expected, however the difference in propagation is the most likely cause for the clear increase in the error, as seen, for example, in Table 5.13.

*Table 5.13 Scenario 3: GS3 WoC for satellite 1.*

<b>Date</b>	<b>Event Start <math>\Delta T</math></b>	<b>Event End <math>\Delta T</math></b>
2023-10-20	62.279 s	56.781 s
2023-10-21	80.505 s	78.893 s
2023-10-22	75.546 s	77.078 s
2023-10-23	64.185 s	62.281 s

*Table 5.14 Scenario 3: UE103 WoC for satellite 1.*

<b>Date</b>	<b>Event Start <math>\Delta T</math></b>	<b>Event End <math>\Delta T</math></b>
2023-10-21	88.697 s	87.505 s
2023-10-22	84.228 s	85.323 s
2023-10-23	82.274 s	81.167 s

*Table 5.15 Scenario 3: USA West WoC for satellite 1.*

<b>Date</b>	<b>Event Start <math>\Delta T</math></b>	<b>Event End <math>\Delta T</math></b>
2023-10-20	1.587 s	0.973 s
2023-10-21	1.565 s	1.300 s
2023-10-22	1.544 s	1.412 s
2023-10-23	1.673 s	1.418 s

*Table 5.16 Scenario 3: Sun-charging WoC for satellite 1.*

<b>Date</b>	<b>Battery Charge</b>	<b>Event Start <math>\Delta T</math></b>	<b>Event End <math>\Delta T</math></b>
2023-10-20	4.0168 A·hr	0.017 s	0.002 s
2023-10-20	4.0151 A·hr	0.007 s	0.002 s
2023-10-20	4.0134 A·hr	0.004 s	0.004 s
2023-10-20	4.0117 A·hr	0.022 s	0.009 s

*Table 5.17 Scenario 3: ISL WoC for satellite 1.*

<b>Date</b>	<b>Line-of-Sight Only</b>	<b>Link Budget</b>
2023-10-20	00:13:24.223	00:16:24.223
	00:32:15.221	00:29:15.221
2023-10-21	00:23:42.756	00:26:42.756
	00:42:33.754	00:39:33.754
2023-10-22	00:34:01.289	00:37:01.289
	00:52:52.286	00:49:52.286
2023-10-23	00:44:19.822	00:47:19.822
	01:03:10.819	01:00:10.819

### 5.3 Lunar Transfer Results

Figure 3.3 presented a Lunar transfer obtained by solving Lambert’s problem using TFC. This section obtains visibility events for a single satellite following this trajectory. In order to simulate a realistic scenario for such an orbit, three GS are used at the locations of NASA’s deep space network antennas: Madrid, Spain; Goldstone, USA; and Canberra, Australia. The elevation angle chosen for all GS is  $10^\circ$ . No UE or SA are used in this scenario. Given that a mission to the Moon would most likely possess Solar panels, Sun events are considered, and the panel is increased from  $0.3 m^2$  to  $56 m^2$ . This number matches the Orion capsule’s four  $14 m^2$  panels. Tables 5.18- 5.20 show the

results obtained for the aforementioned targets. Note that no comparison with STK is done, rather the start and end times gathered by the simulator are displayed. The WoC at these distances are primarily influenced by Earth’s rotation, not the spacecraft’s motion, as indicated by the roughly one-day separation of each visibility window. During the selected epoch, the spacecraft’s sunward trajectory ensures continuous solar contact over the five-day propagation period. The total charge generated during this event is 56,348 Ahr. Note that Lunar eclipses are not considered.

*Table 5.18 Lunar probe Madrid DSN WoC.*

<b>Date</b>	<b>Event Start Time</b>	<b>Event End Time</b>
2023-11-15	17:21:19.787	02:04:59.442
2023-11-16	18:31:43.562	02:58:33.038
2023-11-17	18:59:20.897	03:23:27.881
2023-11-18	19:15:31.919	03:38:47.674
2023-11-19	19:26:32.688	24:00:00.000

*Table 5.19 Lunar probe Goldstone DSN WoC.*

<b>Date</b>	<b>Event Start Time</b>	<b>Event End Time</b>
2023-11-15	00:00:00.000	07:15:13.315
2023-11-16	01:17:40.011	10:08:56.891
2023-11-17	02:06:26.467	10:48:32.829
2023-11-18	02:29:25.215	11:09:23.904
2023-11-19	02:43:40.821	11:22:52.214

*Table 5.20 Lunar probe Canberra DSN WoC.*

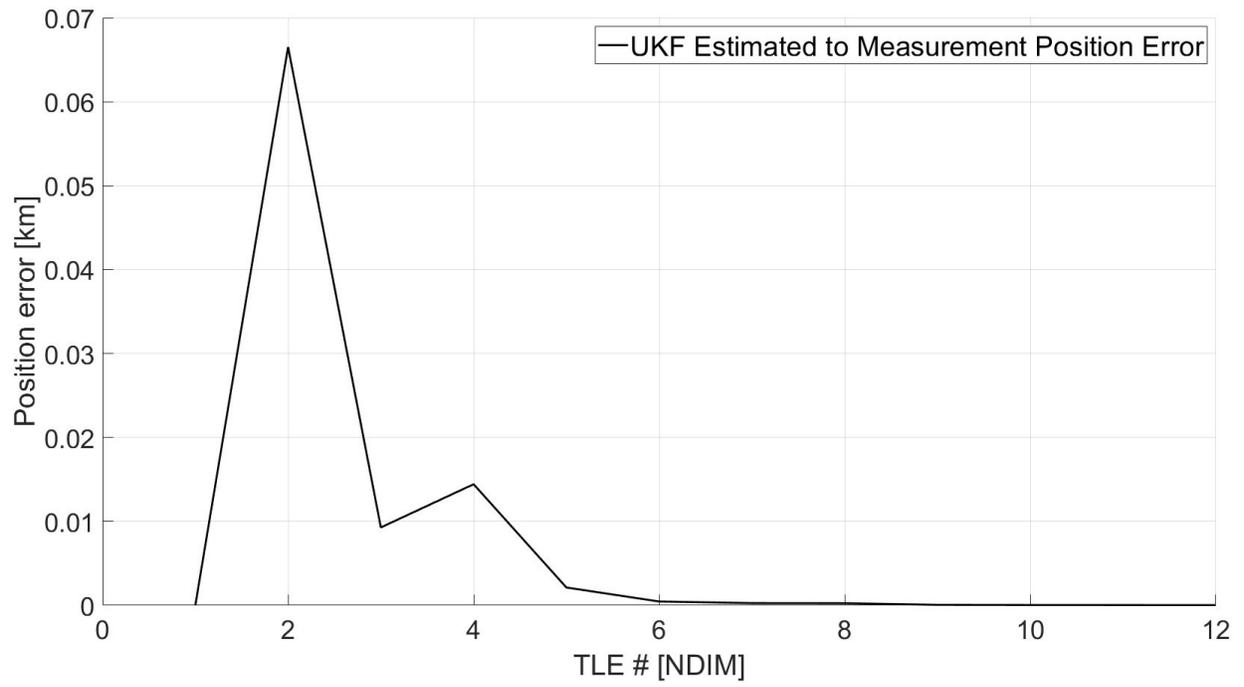
<b>Date</b>	<b>Event Start Time</b>	<b>Event End Time</b>
2023-11-15	02:35:44.316	16:01:52.068
2023-11-16	07:00:49.871	17:29:41.339
2023-11-17	07:40:03.165	18:01:27.270
2023-11-18	08:00:16.785	18:19:34.941
2023-11-19	08:13:16.342	18:31:43.890

## 5.4 Unscented Kalman Filter Results

Figure 2.3 demonstrates the need for properly determining the position of a satellite over a period of several days. A TLE is generally a noisy measurement, thus the uncertainty of each received telemetry is high. Similarly, propagating this data only increases the magnitude of this error. Given that the satellite constellations used for telecommunications are typically in LEO, the dynamics considered within the UKF are the two-body model perturbed by the Earth's oblateness (Eq. 3). Due to the previously presented preliminary analysis, the initial covariance  $P_0$  is defined as follows:

$$P_0 = \left[ \begin{array}{c|c} \sigma_p^2 I_{3 \times 3} & 0_{3 \times 3} \\ \hline 0_{3 \times 3} & \sigma_v^2 I_{3 \times 3} \end{array} \right] \quad (90)$$

where  $\sigma_p$  is the uncertainty in position of the measurements, set to 10 km,  $\sigma_v$  is the uncertainty in velocity, set to 1 km/s,  $0_{3 \times 3}$  is a zero matrix of given dimensions, and  $I_{3 \times 3}$  is an identity matrix with given dimensions. These values are determined from the results of Figure 2.3. The measurement noise matrix  $R_k$  from Eq. 13 is set to  $P_0$  due to the measurements being equal to the states themselves (no measurement model). Twelve TLE are retrieved from 2023-09-17T06:26:12.586272Z to 2023-09-22T18:02:40.903584Z, with a new TLE being obtained every six hours on average. The *Orekit* library is used to interpret the TLE and the filter is designed on Matlab. Figure 5.4 shows the error plot between the estimated position and each measurement. The  $x$ -axis is the measurement number, and the  $y$ -axis is the error in kilometers. As expected, the error spikes with the first measurement and decreases as more measurements are included. Based on the TLE used for this test, only eight measurements are necessary to reduce the error down to 1 meter.



*Figure 5.4* Position difference between estimates and measurements.

## 6 Conclusions and Future Work

The research in this thesis addresses a gap in constellation planning. The Constellation Management System (CMS) is an automated and versatile tool that computes the visibility events to any arbitrary location in the world and develops a schedule for tasks for all satellites in the constellation. It possesses an automated structure, where data flows unimpeded throughout the two modules: the simulator and the scheduler. Several API and databases are built into the system in order to more efficiently store and transfer data as inputs and outputs for the different modules. This thesis demonstrates that the simulator module accurately determines the Windows of Coverage (WoC) to the given locations relative to the state-of-the-art. In parallel to the CMS, the Lambert's problem is solved leveraging the Theory of Functional Connections (TFC), which is proven to be more robust, accurate, and efficient than other methods; in particular, differential corrections. Various unperturbed and perturbed scenarios are generated, considering third-body perturbations, perturbations due to Earth's oblateness, and Solar radiation pressure. Trajectories derived using TFC can be seamlessly integrated into the CMS to calculate the WoC for more intricate scenarios.

Several areas for future upgrades and enhancements in the CMS are identified. The current model includes only basic link budget calculations. Future models should account for more comprehensive factors, including atmospheric and antenna losses, variable field-of-view, and more. While the link budget is currently applied to inter-satellite links, it should also be extended to include connections with ground stations and user equipment for a more thorough analysis. Presently, two different databases, InfluxDB and MongoDB, are used for storing inputs and outputs. To maintain the structural integrity of objects like WoC, a transition to exclusively using MongoDB is planned. Additionally, a practical test of the CMS in an actual 5G-IoT mission is scheduled for 2024. This field test aims to enhance the tool's robustness by identifying and resolving bugs and optimizing performance. These enhancements will collectively elevate the CMS's capabilities, ensuring it remains a cutting-edge tool in the realm of satellite constellation planning.

## REFERENCES

- [1] Curzi, G., Modenini, D., and Tortora, P., “Large Constellations of Small Satellites: A Survey of Near Future Challenges and Missions,” *Aerospace*, Vol. 7, No. 9, 2020, pp. 133–134.
- [2] Giordani, M., and Zorzi, M., “Non-Terrestrial Networks in the 6G Era: Challenges and Opportunities,” *IEEE Network*, Vol. 35, 2021, pp. 244 – 251.
- [3] Ponce, E. M., “Development of an Event Simulator for Satellite Networks with 5G Applications,” Master’s thesis, Universidad Carlos III de Madrid, 2022.
- [4] Kellermann, T., Centelles, R. P., Camps-Mur, D., Ferrús, R., Guadalupi, M., and Augé, A. C., “Novel Architecture for Cellular IoT in Future Non-Terrestrial Networks: Store and Forward Adaptations for Enabling Discontinuous Feeder Link Operation,” *IEEE Access*, Vol. 10, 2022, pp. 68922–68936.
- [5] “TIP’s Non-Terrestrial Connectivity Solutions Project Group Relaunch: New Partnerships and New Scope,” <https://telecominfraproject.com/tips-non-terrestrial-connectivity-solutions-project-group-relaunch-new-partnerships-new-scope/>, 2021. Accessed: 2023-01-04.
- [6] de Azua, J. A. R., Calveras, A., and Camps, A., “Internet of Satellites (IoSat): Analysis of Network Models and Routing Protocol Requirements,” *IEEE Access*, Vol. 6, 2018, pp. 20390 – 20411.
- [7] Selva, D., Golkar, A., Korobova, O., Cruz, I. L., Collopy, P., and I. de Weck, O., “Distributed Earth Satellite Systems: What Is Needed to Move Forward?” *Journal of Aerospace Information Systems*, Vol. 14, No. 8, 2017, p. 412–438.
- [8] Lopez, C. A., “In Pursuit of Autonomous Distributed Satellite Systems,” Ph.D. thesis, Universitat Politècnica de Catalunya, 2019.
- [9] Rigo, C. A., Seman, L. O., Camponogara, E., Filho, E. M., and Bezerra, E. A., “Task Scheduling for Optimal Power Management and Quality-of-Service Assurance in CubeSats,” *Acta Astronautica*, Vol. 179, 2021, p. 550–560.
- [10] Yan, D., Fukunaga, A. S., Rabideau, G., and Chien, S., “ASPEN: A Framework for Automated Planning and Scheduling of Spacecraft Control and Operations,” *Jet Propulsion Laboratory*, 1997.
- [11] Steel, R., Niezette, M., Cesta, A., Fratini, S., Oddi, A., Cortellessa, G., Rasconi, R., Verfaillie, G., Pralet, C., Lavagna, M., Brambilla, A., Castellini, F., Donati, A., and Policella, N., “Advanced Planning and Scheduling Initiative- MrSpock Aims for Xmas,” *ESA Special Publication*, ESA Special Publication, Vol. 673, edited by H. Lacoste, 2009, p. 8.
- [12] Frank, J., and Jonsson, A., “Constraint-Based Attribute and Interval Planning,” *Constraints*, Vol. 8, 2003, pp. 339–364.

- [13] Richards, R. A., Houlette, R. T., and Mohammed, J. L., “Distributed Satellite Constellation Planning and Scheduling,” *FLAIRS Conference*, 2001, pp. 68–72.
- [14] Nies, G., Stenger, M., Krčál, J., Hermanns, H., Bisgaard, M., Gerhardt, D., Haverkort, B., Jongerden, M., Larsen, K. G., and Wognsen, E. R., “Mastering Operational Limitations of LEO Satellites – The GomX-3 Approach,” *Acta astronautica*, Vol. 151, 2018, pp. 726–735.
- [15] Li, G., Xing, L., and Chen, Y., “A Hybrid Online Scheduling Mechanism with Revision and Progressive Techniques for Autonomous Earth Observation Satellite,” *Acta astronautica*, Vol. 140, 2017, pp. 308–321.
- [16] Du, B., and Li, S., “A New Multi-Satellite Autonomous Mission Allocation and Planning Method,” *Acta astronautica*, Vol. 163, 2019, pp. 287–298.
- [17] Ruiz-De-Azua, J. A., “Contribution to the Development of Autonomous Satellite Communications Networks,” Ph.D. thesis, Universitat Politècnica de Catalunya, 2020.
- [18] Singla, A., Criscola, F., Ponce, E., Canales, D., Calveras, A., and Ruiz-De-Azua, J. A., “(Preprint) Towards 6G Non-Terrestrial Networks - An Autonomous Constellation Management Engine,” , 2023.
- [19] Criscola, F., Singla, A., Ponce, E., Calveras, A., Ruiz-De-Azua, J. A., and Canales, D., “(Preprint) Development of a Simulator for Coverage Planning of a 6G/IoT Constellation,” , 2023.
- [20] Baker-McEvelly, B., Doroba, S., Gilliam, A., Criscola, F., Canales, D., Frueh, C., and Henderson, T., “(Preprint) A Review on Hot-Spot Areas Within the Cislunar Region and Upon the Moon Surface, and Methods to Gather Passive Information from these Regions,” *33rd AAS/AIAA Space Flight Mechanics Meeting, Austin, 2023*.
- [21] “I2CAT Foundation,” <https://i2cat.net/>, 2023. Accessed: 2023-11-20.
- [22] Pratt, S. R., Raines, R. A., Fossa, C. E., and Temple, M. A., “An Operational and Performance Overview of the IRIDIUM Low Earth Orbit Satellite System,” *IEEE Communications Surveys & Tutorials*, Vol. 2, No. 2, 1999, pp. 2–10.
- [23] Wang, P., Reinelt, G., Gao, P., and Tan, Y., “A Model, a Heuristic and a Decision Support System to Solve the Scheduling Problem of an Earth Observing Satellite Constellation,” *Computers & Industrial Engineering*, Vol. 61, 2011, pp. 322–335.
- [24] Abramson, M., Carter, D., Kolitz, S., Ricard, M., and Scheidler, P., “Real-Time Optimized Earth Observation Autonomous Planning,” *NASA Earth Science Technology Conference, Pasadena, CA, 2002*.
- [25] Eichler, P., Teichwart, M., Schön, A., and Khanina, E., “Deployment of EUTELSAT’s New Generation Constellation Control System (Neo),” *Space OPS 2004 Conference*, 2004, p. 160.

- [26] Chamoun, J.-P., Risner, S., Beech, T., and Garcia, G., “Bridging ESA and NASA Worlds: Lessons Learned from the Integration of hifly/SCOS-2000 in NASA’s GMSEC,” *2006 IEEE Aerospace Conference*, IEEE, 2006.
- [27] Wang, S., Zhao, L., Cheng, J., Zhou, J., and Wang, Y., “Task Scheduling and Attitude Planning for Agile Earth Observation Satellite with Intensive Tasks,” *Aerospace Science and Technology*, Vol. 90, 2019, pp. 23–33.
- [28] Li, P., Li, J., Li, H., Zhang, S., and Yang, G., “Graph Based Task Scheduling Algorithm for Earth Observation Satellites,” *2018 IEEE Global Communications Conference (GLOBECOM)*, IEEE, 2018.
- [29] Curtis, H. D., *Orbital Mechanics for Engineering Students*, 4<sup>th</sup> ed., Springer, 2020.
- [30] Vallado, D. A., *Fundamentals of Astrodynamics and Applications*, Vol. 12, Springer, 2001.
- [31] Zardam, L., Farrés, A., and Puig, A., “High-fidelity Modeling and Visualizing of Solar Radiation Pressure: a Framework for High-Fidelity Analysis,” *UMBC Faculty Collection*, 2020.
- [32] Rossouw, N. C., “A GPS-Based On-Board Orbit Propagator for Low Earth-Orbiting Cube-Sats,” 2015.
- [33] “CelesTrak,” <https://celestrak.org/>, 2023. Accessed: 2023-11-20.
- [34] Baranov, I. V., “SGP4 Propagation Program Design and Validation,” *Canadian Space Agency*, 2009.
- [35] Ly, D., Lucken, R., and Giolito, D., “Correcting TLEs at Epoch: Application to the GPS Constellation,” *Journal of Space Safety Engineering*, Vol. 7, 2020, pp. 302–306.
- [36] Crassidis, J. L., and Junkins, J. L., *Optimal Estimation of Dynamic Systems*, 2<sup>nd</sup> ed., Champan & Hall, 2011.
- [37] Hamad, M., Elmageed, E., Yassin, A., and Esawi, M., “The Unscented Kalman Filter Applied to Satellite Orbit Determination (Using Only Publically Available Two-Line Element Sets),” 2016.
- [38] Canales, D., “Transfer Design Methodology Between Neighborhoods of Planetary Moons in the Circular Restricted Three-body Problem,” Ph.D. thesis, Purdue University, 2021.
- [39] Mortari, D., Criscola, F., and Canales, D., “(Preprint) Perturbed Lambert Problem using the Theory of Functional Connections,” *33rd AAS/AIAA Space Flight Mechanics Meeting, Austin, Texas*, 2023.
- [40] Criscola, F., Hudson, Z., Canales, D., and Mortari, D., “(Preprint) Solution of the Perturbed Lambert’s Problem using the Theory of Functional Connections,” *AAS/AIAA Astrodynamics*

*Specialist Meeting, Big Sky, Montana, 2023.*

- [41] de la Torre Sangra, D., and Fantino, E., “Review of Lambert’s Problem,” *25th International Symposium on Space Flight Dynamics*, 2019.
- [42] Klumpp, A. R., “Universal Lambert and Kepler Algorithms for Autonomous Rendezvous,” *Astrodynamics Conference*, 1990.
- [43] Peterson, G., Campbell, E., Hawkins, A., Balbás, J., Ivy, S., Merkurjev, E., Hall, T., Rodriguez, P., and Gustafson, E., “Relative Performance of Lambert Solvers 1: Zero-Revolution Methods,” *Advances in the Astronautical Sciences*, Vol. 136, 2010, pp. 1495–1509.
- [44] il Lee, S., Ahn, J., and Bang, J., “Dynamic Selection of Zero-Revolution Lambert Algorithms Using Performance Comparison Map,” *Aerospace Science and Technology*, Vol. 51, 2016, pp. 96–105. <https://doi.org/https://doi.org/10.1016/j.ast.2016.01.018>, URL <https://www.sciencedirect.com/science/article/pii/S1270963816300293>.
- [45] Wailliez, S. E., “On Lambert’s Problem and the Elliptic Time of Flight Equation: A Simple Semi-Analytical Inversion Method,” *Advances in Space Research*, Vol. 53, No. 5, 2014, pp. 890–898. <https://doi.org/https://doi.org/10.1016/j.asr.2013.12.033>, URL <https://www.sciencedirect.com/science/article/pii/S0273117713008302>.
- [46] De La Torre, D., Flores, R., and Fantino, E., “On the Solution of Lambert’s Problem by Regularization,” *Acta Astronautica*, Vol. 153, 2018, pp. 26–38. <https://doi.org/https://doi.org/10.1016/j.actaastro.2018.10.010>, URL <https://www.sciencedirect.com/science/article/pii/S0094576518306106>.
- [47] Avanzini, G., “A Simple Lambert Algorithm,” *Journal of Guidance Control and Dynamics*, Vol. 31, 2008, pp. 1587–1594. <https://doi.org/10.2514/1.36426>.
- [48] Arlulkar, P., and D., N., “Solution Based on Dynamical Approach for Multiple-Revolution Lambert Problem,” *Journal of Guidance Control and Dynamics*, Vol. 34, 2011, pp. 920–923. <https://doi.org/10.2514/1.51723>.
- [49] Pasicznic, J., Servadio, S., and Linares, R., “A Lambert’s Problem Solution via the Koopman Operator with Orthogonal Polynomials,” 2022.
- [50] Thorne, J. D., and Bain, R. D., “Series Reversion/Inversion of Lambert’s Time Function,” *Journal of the Astronautical Sciences*, Vol. 43, No. 3, 1995, pp. 277–287.
- [51] Thorne, J. D., “Convergence Behavior of Series Solutions of the Lambert Problem,” *Journal of Guidance, Control, and Dynamics*, , No. 8, 2014, pp. 1–6. <https://doi.org/10.2514/1.G000701>.
- [52] Boltz, F. W., “Second-Order P-Iterative Solution of the Lambert/Gauss Problem,” *Journal of the Astronautical Sciences*, Vol. 32, 1984, pp. 475–485.
- [53] Leake, C., Johnston, H., and Mortari, D., *The Theory of Functional Connections: A Functional*

*Interpolation. Framework with Applications*, Lulu, Morrisville NC, 2022.

- [54] Mortari, D., “The Theory of Connections: Connecting Points,” *Mathematics*, Vol. 5, No. 4, 2017, pp. 57–1–57–15. <https://doi.org/10.3390/math5040057>.
- [55] Mortari, D., “Least-Squares Solution of Linear Differential Equations,” *Mathematics*, Vol. 5, No. 4, 2017, pp. 48–1–48–18. <https://doi.org/10.3390/math5040048>.
- [56] Mortari, D., Johnston, H. R., and Smith, L., “High accuracy least-squares solutions of non-linear differential equations,” *Journal of Computational and Applied Mathematics*, Vol. 352, 2019, pp. 293–307. <https://doi.org/https://doi.org/10.1016/j.cam.2018.12.007>.
- [57] Mortari, D., and Leake, C. D., “The Multivariate Theory of Connections,” *Mathematics*, Vol. 7, No. 3, 2019, pp. 296–1–296–22. <https://doi.org/10.3390/math7030296>.
- [58] Johnston, H. R., “The Theory of Functional Connections: A Journey from Theory to Application,” Ph.D. thesis, Texas A&M University, August 2021.
- [59] Leake, C. D., “The Multivariate Theory of Functional Connections: An  $n$ -Dimensional Constraint Embedding Technique Applied to Partial Differential Equations,” Ph.D. thesis, Texas A&M University, August 2021.
- [60] “Intellij IDEA,” <https://www.jetbrains.com/idea/>, 2023. Accessed: 2023-10-11.
- [61] “OREKIT,” <https://orekit.org/>, 2008. Accessed: 2022-12-20.
- [62] “InfluxData,” <https://www.influxdata.com/>, 2013. Accessed: 2022-12-25.
- [63] “MongoDB,” <https://www.mongodb.com/>, 2023. Accessed: 2023-10-11.
- [64] Mamajek, E., Prsa, A., Torres, G., Harmanec, P., Asplund, M., Bennett, P., et al., “IAU 2015 Resolution B3 on Recommended Nominal Conversion Constants for Selected Solar and Planetary Properties,” *Solar System Astrophysics*, 2015.
- [65] Afif, A. R., Aprillia, B. S., and Priharti, W., “Design and Implementation of Battery Management System for Portable Solar Panel with Coulomb Counting Method,” *IOP Conference Series: Materials Science and Engineering*, Vol. 771, No. 1, 2020, p. 012005.
- [66] “OptaPlanner,” [https://docs.optaplanner.org/latest/optaplanner-docs/html\\_single/index.html](https://docs.optaplanner.org/latest/optaplanner-docs/html_single/index.html), 2006. Accessed: 2023-03-06.
- [67] Li, W. M., Mitchell, V., Nault, B. R., and Brind, D., “Operating Room Scheduling and Adaptive Control Using a Priority First Fit Decreasing Heuristic,” *Engineering Management Research*, Vol. 4, No. 2, 2015, pp. 46–53.
- [68] Santos, D., Fernandes, P., Cardoso, H. L., and Oliveira, E., “A Weighted Constraint Optimization Approach to the Nurse Scheduling Problem,” *2015 IEEE 18th International Conference*

*on Computational Science and Engineering*, 2015.

- [69] Glover, F., “Tabu Search—Part I,” *ORSA Journal on Computing*, Vol. 1, No. 3, 1989, pp. 190–206.
- [70] Glover, F., “Tabu Search—Part II,” *ORSA Journal on Computing*, Vol. 2, No. 1, 1990, pp. 4–32.
- [71] Burke, E. K., and Bykov, Y., “The Late Acceptance Hill-Climbing Heuristic,” *European Journal of Operational Research*, Vol. 258, No. 1, 2017, pp. 70–78.
- [72] Bykov, Y., and Petrovic, S., “A Step Counting Hill Climbing Algorithm applied to University Examination Timetabling,” *Journal of Scheduling*, Vol. 19, No. 4, 2016, pp. 479–492.
- [73] De Weck, O. L., Ross, A. M., and Rhodes, D. H., “Investigating Relationships and Semantic Sets Amongst System Lifecycle Properties (Ilities),” *Third International Engineering Systems Symposium*, 2012.