

# **Cislunar Trajectory Design Techniques in Virtual Reality**

by

Dhathri Harsha Somavarapu

A dissertation submitted to the Graduate Faculty of  
Auburn University  
in partial fulfillment of the  
requirements for the Degree of  
Doctor of Philosophy

Auburn, Alabama

August 05, 2023

Keywords: Cislunar Space, Trajectory Design, Virtual Reality

Copyright 2023 by Dhathri Harsha Somavarapu

Approved by

Davide Guzzetti, Chair, Assistant Professor of Aerospace Engineering  
Masatoshi Hirabayashi, Associate Professor of Aerospace Engineering  
Ehsan Taheri, Assistant Professor of Aerospace Engineering  
Hans-Werner Van-Wyk, Associate Professor of Mathematics  
Cheryl D. Seals, Professor of Computer Science

## Abstract

Mathematical models of various astrodynamical systems in the Solar system and identification of solutions within these models are crucial for the burgeoning space exploration efforts carried out by the government agencies and the private industry in the 21st century. Often times, these mathematical models are highly nonlinear and coupled. Closed-form analytical solutions to these systems seldom exist, thus forcing the human experts to use numerical methods to identify solutions. However, random initial-conditions or guesses to these numerical methods do not yield desired orbital motion solutions. And hence the identification of good initial-guesses became a problem in itself in this realm of engineering. There is a great need for novel techniques to identify good initial guesses to these powerful numerical methods. With this dissertation work, I leverage the virtual reality platform to help human operators derive good initial guesses for trajectory design problems in the cislunar space. The real three dimensional visualization of orbital motion in a given dynamical model on the VR platform is quite powerful for human operators to leverage their natural intuitive abilities to design trajectories.

Within this dissertation work, I present four new techniques on the VR platform to help the human operator identify potentially good initial guesses for the periodic orbit identification problem in the circular restricted three-body model (CR3BP) of orbital motion for the trajectory design in the cislunar space. The four techniques are validated with periodic orbit identification of four types of Lagrange point periodic orbits, Lyapunov, halo, axial and vertical. All four orbit types are successfully identified with the initial guesses derived using the four techniques.

I also implement an orbit-chain method application on the VR platform for low-thrust transfer trajectory identification in the cislunar circular restricted three-body model (CR3BP). The low-thrust transfer trajectory identification problem is much more complex because of the size of the search space of the problem. This problem requires finding dynamically feasible paths that are also optimal given some criteria. The orbit-chain method allows for taking advantage of naturally feasible paths of orbital motion in the initial guesses to help ease the

computational cost of finding optimal transfer trajectories. This VR based orbit-chain method is successfully validated with two example transfer trajectory identification problems.

Together, these VR based initial-guess derivation methods prove to be a feasible option for preliminary trajectory design in the cislunar space.

## Acknowledgments

This dissertation would not be possible without the support of numerous individuals. First and foremost, I am thankful for the unwavering support of my mother and sister for my endeavor to pursue doctoral studies this late in my life. They are and will continue to be the rocks of my life.

Secondly, I am forever thankful to my advisor, Dr. Davide Guzzetti, for a life altering experience in my doctoral studies. This doctoral journey turned out to be everything I had hoped for and much more in terms of learning, growing and making a dent in this world with my work because of his guidance and supervision. I am very thankful to have him in my corner in my professional life now and in the future.

Thirdly, I am thankful for the thoughtful guidance provided to me by my committee. I am very thankful for the two math classes I took from Dr. Hans Werner Van-Wyk and the continued discussions on collocation to this day.

The work from this dissertation is made possible in equal parts by the the support from the Auburn University Office of the Provost seed funding for AR/VR/XR initiatives and the NASA grant NNH20ZHA001O.

Lastly, I am thankful to the colleagues that I got to work with, and share pain and laughter with during my time at Auburn University. I am particularly thankful to Eirik Mulder for enlightening me on and providing support for the user-interactivity element development for my work. I am also thankful to Kanak Parmar for all the lively discussions we have had and continue to have at Auburn. Although my time in Auburn is coming to a close, I cherish our common time and experiences and I sure hope we run into each other again soon.

## Table of Contents

Abstract . . . . .	ii
Acknowledgments . . . . .	iv
List of Figures . . . . .	ix
List of Tables . . . . .	xii
1 Introduction . . . . .	1
1.1 Motivation . . . . .	1
1.2 Objectives . . . . .	2
1.3 Previous Contributions . . . . .	3
1.3.1 The Virtual Reality Platform for Trajectory Design . . . . .	3
1.3.2 The Cislunar Dynamics . . . . .	4
1.3.3 Cislunar Periodic Orbits . . . . .	4
1.3.4 Cislunar Orbital Transfers . . . . .	5
1.3.5 Differential Correction (DC) Methods for Trajectory Design . . . . .	7
1.3.6 Generalized Collocation for Trajectory Optimization . . . . .	8
1.4 Current Work . . . . .	9
2 The Cislunar Dynamical System . . . . .	11
2.1 The N-Body Model . . . . .	11
2.2 The Two-Body Model . . . . .	15
2.2.1 Constants of Two-Body Motion . . . . .	15

2.2.2	Two-Body Trajectories (Keplerian Motion) . . . . .	17
2.3	The Circular Restricted Three-Body Model . . . . .	18
2.3.1	The Equations of CR3BP Motion . . . . .	19
2.3.2	The Pseudo-Potential Function . . . . .	28
2.3.3	The Gravitational Equilibrium (Lagrange) Points . . . . .	28
2.3.4	The Jacobi Integral Constant and Zero-Velocity Surfaces . . . . .	31
2.3.5	Periodic Orbits in the CR3BP System . . . . .	35
2.3.6	Transformation of CR3BP Motion to an Inertial System . . . . .	35
3	Trajectory Design and Optimization Techniques in the Cislunar System . . . . .	38
3.1	The State Transition Matrix . . . . .	38
3.1.1	The State Transition Matrix for the CR3BP System . . . . .	40
3.1.2	The Monodromy Matrix and Orbit Stability . . . . .	41
3.2	Trajectory Targeting Methods . . . . .	42
3.2.1	The Multi-Variate Newton Differential Corrections . . . . .	42
3.2.2	The Single-Shooting Method . . . . .	44
3.2.3	The Multiple-Shooting Method . . . . .	47
3.3	Trajectory Optimization Methods . . . . .	51
3.3.1	The Generalized Collocation Method . . . . .	52
4	Trajectory Design and Optimization Techniques on the Virtual Reality Platform . . . . .	59
4.1	The Simulation of the Cislunar CR3BP System . . . . .	59
4.2	Periodic Orbit Identification Techniques . . . . .	60
4.2.1	Technique-I: Initial-Guess from a Non-Dynamical User-Drawing . . . . .	60
4.2.2	Technique-II: Initial-Guess from a Dynamical User-Drawing . . . . .	64
4.2.3	Technique-III: Initial-Guess from Patch-Points . . . . .	69

4.2.4	Technique-IV: Initial-Guess from the Simulation of Pure CR3BP Dynamical Motion . . . . .	74
4.3	The Numerical Algorithms for Periodic Orbit Identification . . . . .	79
4.3.1	The Cost Function for Periodic Orbit Identification . . . . .	79
4.3.2	Initial-Guesses for Technique-I . . . . .	79
4.3.3	Initial-Guesses for Technique-II . . . . .	82
4.3.4	Initial-Guesses for Technique-III . . . . .	83
4.3.5	Initial-Guesses for Technique-IV . . . . .	83
4.3.6	Implementation Details . . . . .	83
4.4	System Architectures for Periodic Orbit Identification Techniques . . . . .	83
4.5	Transfer Trajectory Identification Technique . . . . .	84
4.6	The Numerical Algorithm for Transfer Trajectory Identification . . . . .	89
4.7	System Architecture for the Transfer Trajectory Identification Technique . . . . .	91
5	Periodic Orbit Identification on the Virtual Reality Platform . . . . .	93
5.1	Technique-I Periodic Orbits . . . . .	93
5.2	Technique-II Periodic Orbits . . . . .	96
5.3	Technique-III Periodic Orbits . . . . .	99
5.4	Technique-IV Periodic Orbits . . . . .	102
5.5	Comparison of the Techniques . . . . .	105
6	Transfer Trajectory Identification on the Virtual Reality Platform . . . . .	107
6.1	Comparison of Results . . . . .	114
7	Conclusion . . . . .	117
7.1	Investigation Summary . . . . .	117
7.2	Periodic Orbit Identification Techniques . . . . .	117
7.3	Transfer Trajectory Identification Technique . . . . .	118

7.4	Future Work Recommendations . . . . .	119
	References . . . . .	120
	Appendices . . . . .	124
A	The VR Snapshots . . . . .	125
	A.1 Common Snapshots . . . . .	125
	A.2 Periodic Orbit Identification Snapshots . . . . .	128
	A.2.1 Snapshots for Technique-II . . . . .	128
	A.2.2 Snapshots for Technique-III . . . . .	130
	A.2.3 Snapshots for Technique-IV . . . . .	132
	A.3 Transfer Trajectory Identification Snapshots . . . . .	133

## List of Figures

2.1	An N-Body System in an Inertial Frame . . . . .	12
2.2	A Representation of Position Vectors with respect to a Central Body, $i$ , in an Inertial Frame . . . . .	13
2.3	A Model of the Restricted Three Body Motion . . . . .	20
2.4	The Gravitational Equilibrium (Lagrange) Points . . . . .	30
3.1	A Depiction of an Initial-Guess for the Multiple-Shooting Method . . . . .	48
3.2	A Depiction of a Solution for the Multiple-Shooting Method . . . . .	48
3.3	Node Placement Strategy for Collocation with 7th Degree Polynomial . . . . .	54
4.1	A Depiction of the Earth-Moon CR3BP System in VR . . . . .	60
4.2	VR User-Interaction Menu for Technique-I . . . . .	61
4.3	A Flow-Chart of Periodic Orbit Identification in Technique-I . . . . .	62
4.4	VR User-Interaction Menu for Technique-II . . . . .	65
4.5	A Flow-Chart of Periodic Orbit Identification in Technique-II . . . . .	66
4.6	VR User-Interaction Menu for Technique-III . . . . .	70
4.7	A Flow-Chart of Periodic Orbit Identification in Technique-III . . . . .	71
4.8	VR User-Interaction Menu for Technique-IV . . . . .	75
4.9	A Flow-Chart of Periodic Orbit Identification in Technique-IV . . . . .	76
4.10	Periodic Orbit Identification System Architecture with MATLAB Support . . . . .	84
4.11	Periodic Orbit Identification System Architecture with Python (CasADi) Support . . . . .	84
4.12	A Flow-Chart of Transfer Trajectory Identification in VR . . . . .	86
4.13	Transfer Trajectory Identification System Architecture with MATLAB Support . . . . .	92
5.1	Sample Collection Periodic Orbits Identified in Technique-I . . . . .	96

5.2	Sample Collection Periodic Orbits Identified in Technique-II . . . . .	99
5.3	Sample Collection Periodic Orbits Identified in Technique-III . . . . .	102
5.4	Sample Collection Periodic Orbits Identified in Technique-IV . . . . .	105
6.1	Initial-Guess Construction on the VR Platform for a DRO to $L_3$ Lyapunov Transfer . . . . .	108
6.2	The Solution for a DRO to $L_3$ Lyapunov Transfer . . . . .	109
6.3	Mass Consumption for a DRO to $L_3$ Lyapunov Transfer . . . . .	109
6.4	Thrust Profile for a DRO to $L_3$ Lyapunov Transfer . . . . .	110
6.5	Initial-Guess Construction on the VR Platform for an NRHO-to-DRO Exterior Transfer . . . . .	111
6.6	The Solution for an NRHO-to-DRO Exterior Transfer . . . . .	112
6.7	Mass Consumption for an NRHO-to-DRO Exterior Transfer . . . . .	113
6.8	Thrust Profile for an NRHO-to-DRO Exterior Transfer . . . . .	113
6.9	Pritchett's [19] Solution for a DRO to $L_3$ Lyapunov Orbit Transfer . . . . .	114
6.10	Pritchett's [19] Solution for an NRHO-to-DRO Exterior Transfer . . . . .	115
A.1	AOH - A Depiction of the Adjustment of Height of CR3BP System in VR . . .	125
A.2	AOE - A Depiction of the Adjustment of Energy of CR3BP System in VR . . .	125
A.3	AOTOF - A Depiction of the Adjustment of Time-of-Flight of CR3BP Look-Ahead Path in VR . . . . .	126
A.4	AOPP - A Depiction of the Adjustment of Patch-Points in the Initial-Guess in VR	126
A.5	SOP - A Depiction of the Selection Optimizer in VR . . . . .	126
A.6	ROP - A Depiction of the Running of Optimizer in VR . . . . .	127
A.7	WFS - A Depiction of the Waiting for Solution in VR . . . . .	127
A.8	ROS - A Depiction of the Review of Solution in VR . . . . .	127
A.9	AOS - A Depiction of the Archival of the Solution in VR . . . . .	128
A.10	SDRWG-II - A Depiction of the Starting of the Drawing in VR . . . . .	128
A.11	CDRWG-II - A Depiction of the Continuation of the Drawing in VR . . . . .	129

A.12 ALKHD - A Depiction of the Acceptance of the CR3BP Look-Ahead Path in VR129	
A.13 AIPPT - A Depiction of the First Two Patch-Point Placement . . . . .	130
A.14 MVAPPT - A Depiction of the Velocity Direction Vector Manipulation . . . . .	130
A.15 ANPPT - A Depiction of the Continued Patch-Point Placement . . . . .	131
A.16 CLKIG - A Depiction of the Capture of Initial Guess Action . . . . .	131
A.17 APPT - A Depiction of the Placement of Initial Position Point . . . . .	132
A.18 MVV - A Depiction of the Manipulation of the Velocity Vector . . . . .	132
A.19 A VR Orbital Transfer Scene with Energy Plots . . . . .	133
A.20 SOSOP - A Depiction of the Source Orbit Energy Plot . . . . .	133
A.21 SCRL-S - A Depiction of the Source Orbit Energy Plot Scrolling Action . . . . .	134
A.22 SOSO - A Depiction of the Selection of the Source Orbit . . . . .	134
A.23 SOIOP - A Depiction of the Intermediate Orbit Energy Plot . . . . .	134
A.24 SCRL-I - A Depiction of the Source Orbit Energy Plot Scrolling Action . . . . .	135
A.25 SOIO - A Depiction of the Selection off the Intermediate Orbit . . . . .	135
A.26 SOTOP - A Depiction of the Target Orbit Energy Plot . . . . .	135
A.27 SCRL-T - A Depiction of the Target Orbit Energy Plot Scrolling Action . . . . .	136
A.28 SOTO - A Depiction of the Selection of the Target Orbit . . . . .	136
A.29 SWTCLP - A Depiction of the Action to Switch to the Clipping Mode . . . . .	136
A.30 CLIP - A Depiction of the Orbit Clipping Action . . . . .	137
A.31 SWTOPT - A Depiction of the Action to Switch to Optimization Mode . . . . .	137
A.32 ROP - A Depiction of the Action to Run Optimizer . . . . .	137
A.33 WFS - A Depiction of the Action tO Wait for the Solution . . . . .	138
A.34 ROTs - A Depiction of the Review of the Solution . . . . .	138
A.35 AOTS - A Depiction of the Archival of the Solution . . . . .	138

## List of Tables

2.1	<i>L</i> <sub>1</sub> and <i>L</i> <sub>2</sub> Zero Velocity Curves and Surfaces . . . . .	33
2.2	<i>L</i> <sub>3</sub> and <i>L</i> <sub>4,5</sub> Zero Velocity Curves and Surfaces . . . . .	34
5.1	Solution Parameters for an <i>L</i> <sub>1</sub> Lyapunov Solution from Technique-I . . . . .	94
5.2	Solution Parameters for an <i>L</i> <sub>1</sub> Halo Solution from Technique-I . . . . .	94
5.3	Solution Parameters for an <i>L</i> <sub>1</sub> Axial Solution from Technique-I . . . . .	95
5.4	Solution Parameters for an <i>L</i> <sub>1</sub> Vertical Solution from Technique-I . . . . .	95
5.5	Solution Parameters for an <i>L</i> <sub>1</sub> Lyapunov Solution from Technique-II . . . . .	97
5.6	Solution Parameters for an <i>L</i> <sub>1</sub> Halo Solution from Technique-II . . . . .	97
5.7	Solution Parameters for an <i>L</i> <sub>1</sub> Axial Solution from Technique-II . . . . .	98
5.8	Solution Parameters for an <i>L</i> <sub>1</sub> Vertical Solution from Technique-II . . . . .	98
5.9	Solution Parameters for an <i>L</i> <sub>1</sub> Lyapunov Solution from Technique-III . . . . .	100
5.10	Solution Parameters for an <i>L</i> <sub>1</sub> Halo Solution from Technique-III . . . . .	100
5.11	Solution Parameters for an <i>L</i> <sub>1</sub> Axial Solution from Technique-III . . . . .	101
5.12	Solution Parameters for an <i>L</i> <sub>1</sub> Vertical Solution from Technique-III . . . . .	101
5.13	Solution Parameters for an <i>L</i> <sub>1</sub> Lyapunov Solution from Technique-IV . . . . .	103
5.14	Solution Parameters for an <i>L</i> <sub>1</sub> Halo Solution from Technique-IV . . . . .	103
5.15	Solution Parameters for an <i>L</i> <sub>1</sub> Axial Solution from Technique-IV . . . . .	104
5.16	Solution Parameters for an <i>L</i> <sub>1</sub> Vertical Solution from Technique-IV . . . . .	104
6.1	Solution Parameters for a DRO to <i>L</i> <sub>3</sub> Lyapunov Transfer . . . . .	110
6.2	Solution Parameters for an NRHO-to-DRO Exterior Transfer . . . . .	113

## Chapter 1

### Introduction

Over the past seventy five years, humans have made incredible strides in exploring the Solar system, starting with Sputnik to the more recent test flight of Orion (Artemis-I) to carry humans to the Moon. Both the government space programs (such as Artemis program from NASA), and the burgeoning private industry are attempting to build the space infrastructure to explore farther and farther away from the low-Earth orbit (LEO) and geostationary equatorial orbit (GEO) regions to the Moon and beyond, in search of natural resources (asteroid mining), water, and habitable moons of other planets.

#### 1.1 Motivation

To explore the Solar system, humanity needs a better understanding of the astrodynamical models for motion in these vast regions populated with myriad of planets, moons, and asteroids. Humanity has a good understanding of orbital motion near the Earth in the LEO and GEO regions, since all but a few handful space missions serving humanity are in these regions. This does not mean the prediction of orbital motion in these regions is any easier, far from it. However, prediction of orbital motion in the regions beyond LEO and GEO, even between Earth and the Moon (the cislunar region), is exponentially harder. The orbital motion in the LEO and GEO regions is typically modelled with the ideal two-body model combined with various perturbations, such as the atmospheric drag in LEO and solar radiation pressure in GEO. The two-body model is no longer sufficient to predict orbital motion in the cislunar region due to the significant gravitational effect of the Moon. The orbital motion in the cislunar region is typically modelled using an idealized system of motion called the “Circular Restricted Three-Body Problem (CR3BP)”. This mathematical model of orbital motion is highly nonlinear and coupled

and thus suffers from an initial-guess problem. In any nonlinear system, the initial-guess to the system drives the solutions and random initial-guesses do not typically yield desired motion. The solutions are highly unpredictable in these models. The work presented in this dissertation is motivated by the desire to design mechanisms to provide better initial-guesses to the nonlinear coupled mathematical models used to predict orbital motion in the complex cislunar region. Often times, the process of trajectory design and optimization in the complex astrodynamical systems is driven by human experts employing intuition and insight gained from years of experience in the field. Thus, the current work is also motivated to leverage human intuition and insight into the problem of orbital trajectory design and optimization process in the cislunar region.

## 1.2 Objectives

The orbital motion in the cislunar region is typically modelled with the CR3BP model. There are good foundations for understanding and predicting orbital motion in this model from significant contributions of various scholars. Various families of cislunar periodic orbits are identified in literature. However, the process of identifying initial-guesses that yield either desired periodic orbits or orbital transfer trajectories in the CR3BP model is not standardized. Human experts typically start with two-body orbits and convert them to the CR3BP model, or linear approximations of CR3BP motion and apply differential corrections to achieve desired orbital structures. There is no guarantee that this approach will yield required solutions. The process is often iterative and computationally intensive. There is a lot of room for improvements in terms of providing better initial-guess constructing mechanisms. The current work aims to improve the initial-guess construction process with the hypotheses,

1. Human intuition and insight in the trajectory design and optimization process provides better initial-guesses. More specifically, visualization of orbital motion structures in real three dimensions is highly insightful to the human user designing trajectories for a space mission.

2. The initial-guess construction process can be simplified and expedited in the real three dimensional space provided by a virtual reality (VR) platform. The initial-guess construction for both periodic orbits and transfer trajectories can be more effective on a virtual reality platform.
3. A coherent solution approach to the nonlinear cislunar CR3BP model can be constructed and demonstrated on a virtual reality platform. The approach calls for leveraging proven trajectory targeting and optimization techniques to identify solutions from good initial-guesses constructed on the virtual reality platform.

### 1.3 Previous Contributions

#### 1.3.1 The Virtual Reality Platform for Trajectory Design

The usage of a virtual reality platform for trajectory design and optimization is in nascent stages. The first known mention of the usage of modern virtual reality platform for trajectory design comes from Mathur [3] in 2015. Mathur developed the mechanisms to control parameters that dictate orbital motion in various NASA trajectory design tools such as GMAT [5] and Copernicus [4] using 3D desktop widgets built into these tools. However, Mathur did not extend the development of this concept to a virtual reality as of this writing based on a public literature search. In 2018, Stuart et al. [2] experimented with the virtual reality platform to help ease cognitive load for human users of various complex trajectory design phases and processes in the realm of space mission design. Stuart et al. demonstrated how the orbital motion changes for an interplanetary trajectory when certain control parameters for motion are changed in the virtual reality platform. However, the development of the concept did not proceed further as of this writing based on a public literature search. The VR platform makes visualization of motion easier than a 2D desktop computer. It is especially powerful for enforcing geometric constraints on motion and the work in this dissertation leverages the platform to make the initial-guess construction process simpler and more human-friendly.

### 1.3.2 The Cislunar Dynamics

Humanity's interest in astronomy and astrodynamics is ancient. Various calendars around the world have been built around the motion of the Sun, the Moon and other visible objects in the skies. The modern laws governing celestial motion were first postulated by Johann Kepler between 1609 and 1619, during which he declared the three laws of planetary motion from his mathematical fit to the observations of planets collected by Tycho Brahe [1]. With the publication of the seminal work in 1687, *the Principia*, Sir Isaac Newton ushered in an era of discoveries that expedited humanity's understanding of astrodynamics governing the motion of celestial objects in the Solar system and beyond [6]. Newton's universal law of gravitation initiated the long search for analytical solutions to the motion of celestial objects. In 1722, Leonard Euler reduced the complex N-Body motion to the "Circular Restricted Three-Body Problem (CR3BP)" [7]. With this reduction he also introduced an innovative notion of viewing the problem space from a non-inertial rotating frame. This new model allowed for a drastic increase in the understanding of motion in the cislunar region. In 1772, Joseph-Louis Lagrange discovered the two planar triangular gravitational equilibrium points, the same year that Euler discovered the three colinear gravitational equilibrium points in the CR3BP. In 1836, a single integral of motion for CR3BP was discovered by Carl Gustav Jacob Jacobi [8]. Now known as the Jacobi constant or integral, this led to another powerful discovery of allowable regions of motion called the Zero Velocity Surfaces, by Hill in 1878 [9]. In 1899, French scientist Henri Poincaré proved conclusively that no additional integrals of motion exist in CR3BP [10]. In 1881, the same Henri Poincaré devised a powerful tool for visualizing the complex solution space of CR3BP now called "the Poincaré map" [8]. However, to this day, the CR3BP problem has no known fully closed-form analytical solutions and hence human experts resort to numerical methods to explore and derive desired solutions in this problem space.

### 1.3.3 Cislunar Periodic Orbits

Periodic orbits are bounded solutions in the given dynamical model. Periodic orbits are an important set of solutions for CR3BP model, because they allow for predictable motion in the

system and are possible candidates for various space missions in the cislunar region. There are no universally standard set of procedures available in the field to derive desired periodic motion. In 1881, Henri Poincaré proved the existence infinite number of periodic orbits in the CR3BP model. Since then many scientists and researchers have attempted to identify families of periodic orbits in CR3BP. The dawn of the new computing era in the twentieth century ushered in the discovery of many families of periodic orbits. Grebow [11] describes the numerical methods that can identify a wide variety of related families of periodic orbits in an application for lunar south pole coverage. Guzzetti et al. [12] constructed a rapid interactive catalog of families of periodic and quasi-periodic orbits of CR3BP. Many experts in the academia and industry are currently in pursuit of new periodic orbit families. Zimovan-Spreen [13] from Purdue University most recently published period doubling and period multiplying families of halo periodic orbits. Despite these efforts, identifying the initial-state for a predictable periodic motion in CR3BP remains a difficult problem.

#### 1.3.4 Cislunar Orbital Transfers

Orbital transfers are often required to get to a desired intermediate or final destination in various space missions. There are two primary kinds of orbital transfers, high-thrust and low-thrust. Typically in high thrust transfers, the spacecraft thruster(s) are equipped with imparting a large near instantaneous change to the acceleration of the spacecraft. Hence, the high thrust orbital transfers can be modelled with instantaneous thrust maneuvers. Typically, human space missions are equipped with the high thrust thrusters and carry out high-thrust impulsive maneuvers. In contrast, in low-thrust transfers, the spacecraft thruster(s) can only impart sub-Newton thrust instantaneously while using very little fuel. As a result of this low-thrust capacity, more low-thrust maneuvers over a longer time are needed to achieve the same desired change in acceleration of the spacecraft as would from a high thrust capable thrusters. Typically low-thrust transfers are not suitable for human missions and are frequently used to transfer cargo to a destination. Low-thrust orbital transfers are more dynamically challenging because of the complex search space for solutions and are a focus of this dissertation.

In contrast to the theory of periodic orbits, the theory for low-thrust orbital transfers is not as developed. Typically low-thrust orbital transfers in CR3BP are identified solely using numerical methods. There are two different numerical approaches to the low-thrust orbital transfer problem, indirect and direct. The indirect method uses the calculus of variations approach and requires analytical derivatives for CR3BP equations of motion. The direct method discretizes the transfer trajectory into segments. In the collocation-based direct method the trajectory segments are represented using known orthogonal polynomials, while enforcing the CR3BP dynamics along the polynomial solutions. Shooting methods (Single and Multiple-Shooting) are also examples of direct methods. The direct method does not strictly need analytical derivatives for CR3BP equations of motion. In this dissertation, a direct method called “the generalized collocation method” is employed.

The orbital transfer problem is much harder to address than the identification of periodic orbits in CR3BP, mainly due to the difficulty in identifying good initial-guesses to the transfer solutions. In the identification orbital transfers, one has to first contend with identifying feasible paths for motion in the CR3BP dynamics, and then attempt to optimize either fuel-consumption or time-of-flight or both along these feasible paths. One effective method found in literature for constructing good initial-guesses to the transfer problem is “the orbit-chain method”. The orbit-chain method by definition combines a chain of feasible orbital paths in a given dynamical system to form a global path from a source to a destination. The patched-conic method [1] is an example of the orbit-chain method. The orbit-chain method in the CR3BP system is employed to stitch together periodic orbits and invariant manifolds (natural feasible paths emanating from and to a periodic orbit) to construct an orbital transfer [14]. Koon et al. [14] identified orbit-chains composed of heteroclinic cycles and homoclinic orbits that reveal global flow of motion in the CR3BP. Ross and Lo [15] describe the existence of a vast array of invariant manifold tubes emanating from three-body systems throughout the solar system that, when linked together, form an “Interplanetary Superhighway”. Lo and Parker [16] demonstrated a method to chain LPOs (Low-Prograde-Orbits) and their invariant manifolds to form new complex periodic orbits. Parker et al. [17] generalized this orbit-chain method for identifying feasible paths of orbital motion. The orbit-chain method used in the current work is borrowed from

Pritchett [19]. The orbit-chain method as described by Pritchett requires visual inspection of energy-versus-time-of-flight plots and the corresponding orbits by a human user to compose a good initial-guess to an orbital transfer solution. In the current work, a virtual reality platform is employed to ease the burden of visualization in the orbit-chain method on 2D desktops to the full 3D space of motion.

### 1.3.5 Differential Correction (DC) Methods for Trajectory Design

Differential Correction is a process that leverages the gradients of mathematical conditions/constraints for solutions with respect to the input variables to provide mathematically feasible solutions for a given system of differential equations. Differential Correction goes as far back as Newton's *Principia*. Often, in spacecraft trajectory design problem domain, one particular type of differential calculus problems arises, "the two-point boundary-value problem" in case of the periodic orbit and orbital transfer trajectory identification. The two-point boundary-value problems can be defined as a set of differential equations together with boundary conditions/constraints for solutions of the equations. Typically, trajectory design problems have constraints on starting and ending points of the transfer, which are modelled as the boundary conditions/constraints to the system of differential equations governing the orbital motion. Several methods exist in literature that leverage differential corrections, chief among them are called "the shooting methods". Shooting methods iteratively target boundary conditions until a desired tolerance is met for achieving such boundary conditions, while solving the system of differential equations with a set of initial conditions. The initial conditions in the iterative process are updated by making a correction to the previous initial conditions along the path of a slope that minimizes overall error for boundary conditions. Researchers Keller [20] and Osborne [21] presented shooting methods in which the solutions to two-point boundary value problems are produced by integrating appropriate initial value problems. They also presented limitations of shooting methods, as well as numerically stable treatments for these limitations in 1968 and 1969 respectively. Since then, shooting methods became a staple of solution methods for solving nonlinear systems of equations for which no known analytical solutions exist. The advent of faster computing technologies resulted in employment of the shooting methods

in solving complex spacecraft trajectory design problems. In this dissertation, a free-variable-constraint method presented by Pavlak [22] is utilized for trajectory design problems.

### 1.3.6 Generalized Collocation for Trajectory Optimization

Typically, the spacecraft trajectory design problems involve integrating complex nonlinear system of differential equations. However, constants of integration in multi-body orbital models, such as the cislunar model, are notoriously hard to come by, in fact there is only one constant of motion ever discovered in the CR3BP model, making it very hard to find solutions by employing direct integration of the system nonlinear differential equations of CR3BP model. Collocation is a technique where the unknown solution function of a given system of equations is approximated using a polynomial of degree,  $N$ . Collocation by definition is an implicit integration method, i.e., a future state of the system of equations is part of the integration/algebraic formula to solve for the unknown solution function, in addition to the past or current state. Implicit integration formulae are typically solved iteratively with numerical root finding methods, such as the Newton methods. The generalized collocation method [24] emphasizes the selection of a subset of roots of the basis orthogonal polynomial that is approximating the solution. This primary subset of roots are used to identify the coefficients of the said polynomial. The remaining subset of roots are used to test for conformity with the system of equations governing the solution space of the problem. In the generalized collocation [24] method, a Newton method is employed to find the coefficients of the orthogonal polynomial while reducing the conformity error based on the driving system of equations for the problem. Typically in literature [25, 18, 19], the solution to the polynomial coefficients are identified from a Nonlinear Programming Problem (NLP) formulated from the original problem. In this dissertation, I employed Grebow [25] and Pritchett's [18, 19] formulation of the generalized collocation method. The generalized collocation method is adapted to, 1) identify periodic orbit solutions, and 2) fuel-optimal low-thrust transfer trajectories.

## 1.4 Current Work

In this dissertation, novel techniques to derive initial-guesses for the solutions to periodic orbit and transfer trajectory identification problems in the cislunar dynamics are hypothesized and realized. This document presents four such novel techniques for periodic orbit identification developed leveraging the VR platform. The document also presents an implementation of the orbit-chain method [19] on the VR platform for transfer trajectory identification. The document is organized as follows,

- Chapter 2: The Cislunar Dynamical System.

The primary dynamical model employed in this work is presented in this chapter, with the setup starting from the two-body dynamics, leading up to the CR3BP dynamics.

- Chapter 3: Trajectory Design and Optimization Techniques in the Cislunar System.

The primary numerical methods for periodic orbit targeting and transfer trajectory optimization are presented in this chapter. The setup for this chapter follows a structure starting with fundamental tools needed for the numerical algorithms, followed by the shooting methods and the generalized collocation method.

- Chapter 4: Trajectory Design and Optimization Techniques on the Virtual Reality Platform.

The novel techniques devised in this work for facilitating good initial-guesses for periodic orbit and transfer trajectory identification techniques on the VR platform are described in this chapter.

- Chapter 5: Periodic Orbit Identification on the Virtual Reality Platform.

The verification results from all the periodic orbit techniques described in Chapter 4 are presented in this chapter.

- Chapter 6: Transfer Trajectory Identification on the Virtual Reality Platform.

The verification results for the transfer trajectory identification technique described in Chapter. 4 are presented in this chapter.

- Chapter 7: Conclusion.

A set of conclusions from the results of various techniques and future work recommendations are described in this chapter.

## Chapter 2

### The Cislunar Dynamical System

Mathematical models are critical for prediction of orbital motion. The mathematical description of the cislunar dynamical system begins with the mathematical model of the general N-Body model using Newtonian mechanics. The ideal two-body and restricted three-body models are then formulated with assumptions from certain known or assumed conditions of motion. Relevant particular solutions to these idealized models are presented along with some important characteristics and artifacts of orbital motion.

#### 2.1 The N-Body Model

One of the most general model of the motion of a celestial body is the N-Body model. Any celestial body in space experiences gravitational effects of the other celestial bodies in the system in which it resides. In his seminal publication, *Philosophiae Naturalis Principae Mathematica*, Sir Isaac Newton introduced the this Universal Law of Gravitation, stated as follows,

$$F_g = -\frac{\tilde{G}Mm}{r^3}\vec{r} \quad (2.1)$$

In this equation,  $F_g$  is the force experienced by body with mass  $m$  due to the gravitational pull of a body with mass  $M$ . And  $\vec{r}$  and  $r$  are the distance vector and the distance between the two bodies.  $\tilde{G}$  is the universal gravitational constant. Note that gravitational force acts in the opposite direction of motion of a celestial body, pulling it towards the body exerting the gravitational force. Inherent in equation 2.1 is the assumption that the both the bodies in the equation are spherically symmetric.

Figure 2.1 is showing a notional N-Body system in an inertial frame. The barycenter of the N bodies in this system is represented by the vector,  $\hat{O}$ , and also represents the origin of the inertial frame. Vectors  $\vec{r}_1$  through  $\vec{r}_n$  are the position vectors of the N bodies with respect to the barycenter. However, often we need to obtain the position vector of a spacecraft or another celestial body with respect to a central celestial body such as the Earth, Mars or the Moon. In this figure, the position vector  $\vec{r}_{ij}$  represents the position vector of body  $j$  with respect to a central body,  $i$ .

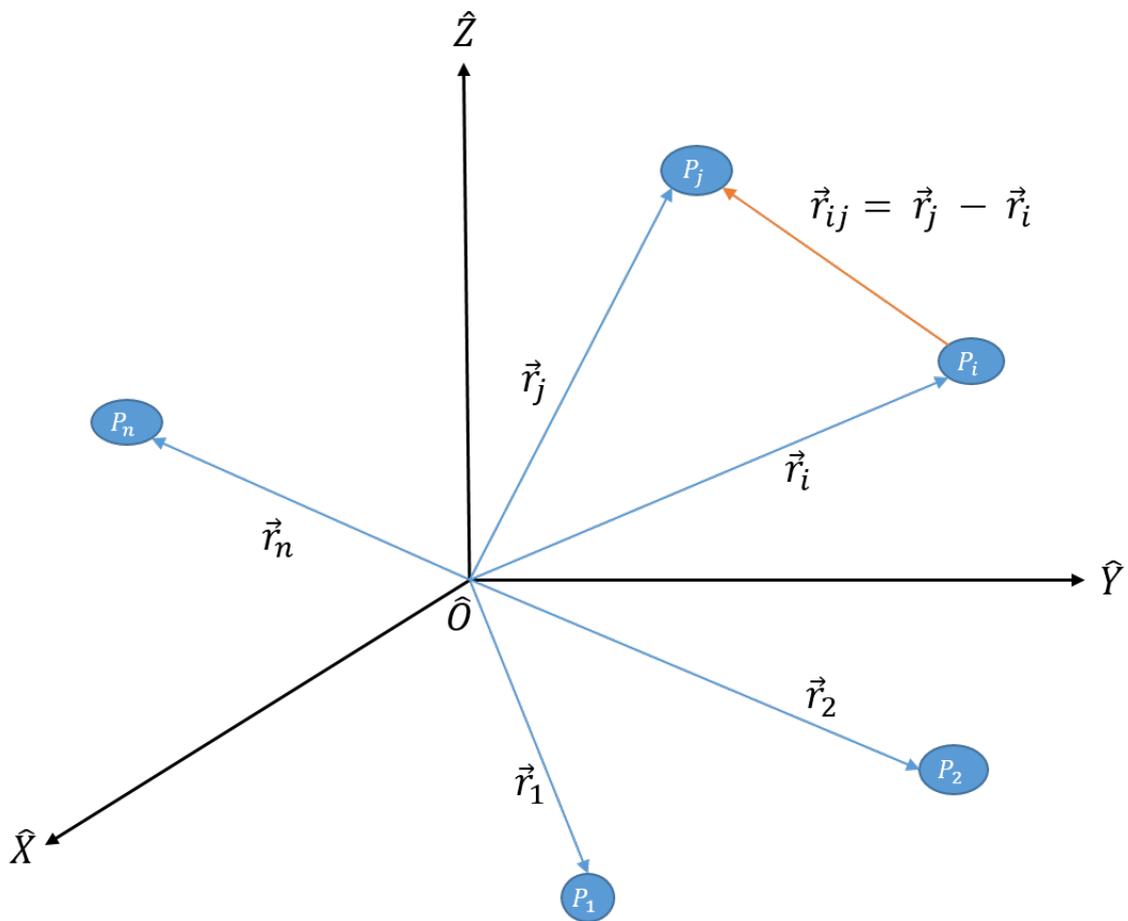


Figure 2.1: An N-Body System in an Inertial Frame

To derive the equations of motion for a spacecraft with respect to a central body,  $i$ , under the influence of various celestial bodies in the Solar system in an inertial frame, we first need to introduce the notations used to represent the position vectors of the spacecraft and various other celestial bodies with respect to that central body,  $i$ . Figure 2.2 is showing the position vectors of other celestial bodies and the spacecraft with respect to the central body,  $i$ . It is

also showing the position vectors of the spacecraft with respect to other celestial bodies in the N-Body system in an inertial frame.

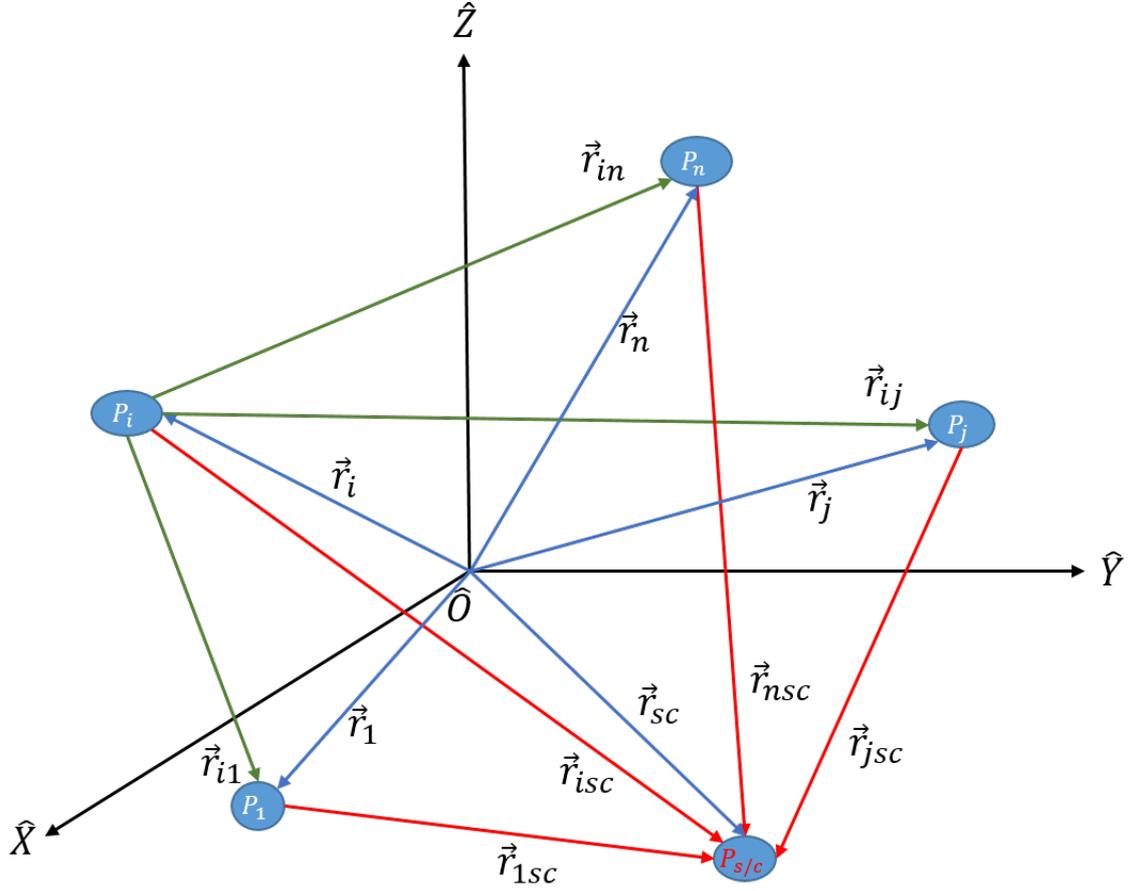


Figure 2.2: A Representation of Position Vectors with respect to a Central Body,  $i$ , in an Inertial Frame

Now we have the position vector of the spacecraft with respect to a central body,  $i$ , defined as,

$$\vec{r}_{isc} = \vec{r}_{sc} - \vec{r}_i \quad (2.2)$$

and the position vector any arbitrary third celestial body,  $j$ , with respect to a central body,  $i$ , defined as,

$$\vec{r}_{ij} = \vec{r}_j - \vec{r}_i \quad (2.3)$$

We also have the position vector of spacecraft with respect to any arbitrary third celestial body,  $j$ , defined as,

$$\vec{r}_{jsc} = \vec{r}_{sc} - \vec{r}_j \quad (2.4)$$

In an inertial frame, the acceleration of the spacecraft with respect to the central body,  $i$ , can be represented as,

$$\ddot{\vec{r}}_{isc} = \ddot{\vec{r}}_{sc} - \ddot{\vec{r}}_i \quad (2.5)$$

When the Newton's universal law of gravitation shown in equation (2.1) and Newton's second law of motion are applied to the motion of the celestial body,  $i$  in the inertial frame, we get,

$$m_i \ddot{\vec{r}}_i = -\tilde{G} m_i \sum_{j=1, j \neq i}^n \frac{m_j}{r_{ji}^3} \vec{r}_{ji}$$

So the acceleration vector for the celestial body,  $i$  in the inertial frame can be represented as,

$$\ddot{\vec{r}}_i = -\tilde{G} \sum_{j=1, j \neq i}^n \frac{m_j}{r_{ji}^3} \vec{r}_{ji} \quad (2.6)$$

Similarly, the acceleration vector for the spacecraft in the inertial frame can be represented as,

$$\ddot{\vec{r}}_{sc} = -\tilde{G} \sum_{j=1}^n \frac{m_j}{r_{jsc}^3} \vec{r}_{jsc} \quad (2.7)$$

Substituting equations (2.6) and (2.7) into equation (2.5), we get,

$$\ddot{\vec{r}}_{isc} = -\tilde{G} \sum_{j=1}^n \frac{m_j}{r_{jsc}^3} \vec{r}_{jsc} + \tilde{G} \sum_{j=1, j \neq i}^n \frac{m_j}{r_{ji}^3} \vec{r}_{ji}$$

When all the vectors are represented with respect to the central body,  $i$ , this equation can be rearranged as,

$$\ddot{\vec{r}}_{isc} = -\tilde{G} m_i \frac{\vec{r}_{isc}}{r_{isc}^3} + \tilde{G} m_j \sum_{j=1, j \neq i}^n \left( \frac{(\vec{r}_{ij} - \vec{r}_{isc})}{\|\vec{r}_{ij} - \vec{r}_{isc}\|^3} - \frac{\vec{r}_{ij}}{r_{ij}^3} \right) \quad (2.8)$$

Equation (2.8) does not have a known closed form analytical solution. Hence, there are various simplifications of this equation under some known conditions or assumptions for orbital motion. In the next two sections, two such simplified models are presented.

## 2.2 The Two-Body Model

The two-body model represents motion of a spacecraft with respect to only one single central body, such as the Earth, Mars or the Moon, ignoring gravitational effects and other perturbing effects of the remaining bodies in the Solar system. The Equation (2.8), when simplified to one central body,  $i$ , becomes,

$$\ddot{\vec{r}}_{isc} = -\tilde{G}m_i \frac{\vec{r}_{isc}}{r_{isc}^3} \quad (2.9)$$

Inherent in this representation of two body motion are two assumptions, 1) the central attracting body,  $i$ , is spherically symmetric, 2) There are no external or internal forces acting on the spacecraft other than the gravity of the central body,  $i$ .

### 2.2.1 Constants of Two-Body Motion

Equation (2.9) can be rewritten as follows,

$$\ddot{\vec{r}}_{isc} + \frac{\tilde{\mu}_i}{r_{isc}^3} \vec{r}_{isc} = 0 \quad (2.10)$$

Here,  $\tilde{\mu}_i = \tilde{G}m_i$ , represents the gravitational parameter of the central body,  $i$ . There are three constants of two-body motion that are conserved throughout orbital motion in this regime. They are the specific mechanical energy and the specific angular momentum and linear momentum of the center of mass of the two-body system.

## Specific Mechanical Energy: A Derivation [1]

Multiplying equation (2.10) by  $\dot{\vec{r}}_{isc}$  gives,

$$\dot{\vec{r}}_{isc} \cdot \ddot{\vec{r}}_{isc} + \dot{\vec{r}}_{isc} \cdot \frac{\tilde{\mu}_i}{r_{isc}^3} \vec{r}_{isc} = 0$$

Using the notation that  $\vec{v} = \dot{\vec{r}}_{isc}$  and the fact that in general for any vector the dot product is equal to a scalar, we get,

$$v\dot{v} + \frac{\tilde{\mu}_i}{r_{isc}^3} r_{isc} \dot{r}_{isc} = 0$$

$$v\dot{v} + \frac{\tilde{\mu}_i}{r_{isc}^2} \dot{r}_{isc} = 0$$

We can use the following mathematical facts to further simplify this equation,

$$\frac{d}{dt} \left( \frac{v^2}{2} \right) = v\dot{v}$$

$$\frac{d}{dt} \left( -\frac{\tilde{\mu}_i}{r_{isc}} \right) = \frac{\tilde{\mu}_i}{r_{isc}^2} \dot{r}_{isc}$$

The simplified equation is,

$$\frac{d}{dt} \left( \frac{v^2}{2} \right) + \frac{d}{dt} \left( -\frac{\tilde{\mu}_i}{r_{isc}} \right) = 0$$

This implies, that the terms in the brackets are constant, yielding the equation for a constant motion called the “Specific Mechanical Energy”,

$$\mathcal{E} = \frac{v_{isc}^2}{2} - \frac{\tilde{\mu}_i}{r_{isc}} \quad (2.11)$$

### Specific Angular Momentum: A Derivation [1]

Cross multiplying equation (2.10) by  $\vec{r}_{isc}$  yields,

$$\vec{r}_{isc} \times \ddot{\vec{r}}_{isc} + \vec{r}_{isc} \times \frac{\tilde{\mu}_i}{r_{isc}^3} \vec{r}_{isc} = 0$$

Since self vector product of any vector is the zero vector, the second term in this equation vanishes to yield,

$$\vec{r}_{isc} \times \ddot{\vec{r}}_{isc} = 0$$

This can be represented using vector calculus as,

$$\frac{d}{dt}(\vec{r}_{isc} \times \dot{\vec{r}}_{isc}) = 0$$

Hence the quantity in the bracket of this derivative is a constant, called the “Specific Angular Momentum”, represented as follows,

$$\vec{h}_{isc} = \vec{r}_{isc} \times \vec{v}_{isc} \quad (2.12)$$

#### 2.2.2 Two-Body Trajectories (Keplerian Motion)

The equation for a two-body trajectory is given as [1],

$$r = \frac{p}{1 + e \cos \nu} \quad (2.13)$$

In this equation,  $p$  is called the “semi-latus rectum”,  $e$  is the “eccentricity” and  $\nu$  is the angular position of the spacecraft along the trajectory called “true anomaly”. This two-body motion in general is also referred to as the “Keplerian motion” after the German astronomer Johannes Kepler who postulated the famous three laws for planetary motion. From this equation of a two-body trajectory, which incidentally also represents a conic section, we can conclude that,

1. Conic sections (circle, ellipse, parabola and hyperbola) are the only possible shapes and paths possible in equations governing the ideal two-body motion.

2. The primary focus of the conic section representing the motion of the spacecraft is at the center of the central body,  $i$ .
3. The specific mechanical energy (sum of potential and kinetic energies) of the spacecraft remains constant throughout conic section representing the motion of the spacecraft, albeit with and exchange between the two forms of energy throughout the conic section.
4. The orbital plane that contains the conic section representing the spacecraft motion remains fixed in an inertial frame centered on the center of the central body,  $i$ .
5. The specific angular momentum of the spacecraft about central body,  $i$ , remains constant throughout the conic section, albeit with position and velocity changing along with the flight path angle to keep the quantity constant throughout the conic section.

Other important definitions for the two-body motion are the expressions of specific mechanical energy, orbital time period and the mean motion of the spacecraft respectively in terms of the semi-major axis,  $a$ , of the conic section representing the motion of the spacecraft,

$$\mathcal{E} = -\frac{\tilde{\mu}}{2a} \quad (2.14)$$

$$T = 2\pi\sqrt{\frac{a^3}{\tilde{\mu}}} \quad (2.15)$$

$$N = \sqrt{\frac{\tilde{\mu}}{a^3}} \quad (2.16)$$

The two-body model is deemed adequate to represent motion of a spacecraft for low-Earth to up to geostationary equatorial orbital (GEO) distances. The model is insufficient to represent motion in the region beyond GEO, since the gravity of the Moon has significant impact on the motion of a spacecraft in this region.

### 2.3 The Circular Restricted Three-Body Model

The sphere-of-influence of the Moon on the motion of any spacecraft or celestial object in the Earth-Moon region is considered to be ten times the GEO distance (approximately 42165 km).

In this sphere-of-influence, the gravitational effect of the Moon makes the two-body model of orbital motion invalid. The motion in this region is not Keplerian, hence the orbits in this region are no longer the predictable conic sections. To model the motion in this region, we need to include the gravitational effect of the Moon in the orbital motion model. Although, the orbit of the Moon around the Earth is slightly eccentric, we can take an instantaneous snapshot of the Moon's position around the Earth and approximate a circle for Moon's motion with that instantaneous radius. The approach taken to model orbital motion in the Earth-Moon system is as follows,

1. Approximate the Moon's orbit around the Earth to be a circle with an instantaneous radius of the Moon around the Earth.
2. Assume that Earth and Moon are perfect spheres.
3. Assume that the body or spacecraft for which the orbital motion is modelled is infinitesimally small when compared to the Earth and the Moon. It means that this third body or spacecraft does not influence the motion of Earth or the Moon.

These assumptions allow us to model orbital motion in the Earth-Moon region with what is defined as the "Circular Restricted Three-Body Model/Problem (CR3BP)". The next few sections illustrate the derivation of equations governing this model and resulting important characteristics governing motion within this model.

### 2.3.1 The Equations of CR3BP Motion

Figure 2.3 is showing a model of the restricted three-body motion. There are two large primary bodies,  $P_1$ ,  $P_2$ , and a small third body,  $P_3$ , that could represent a spacecraft. The inertial coordinate system of the canonical N-Body model is represented by the capital letters,  $\hat{X}$  and  $\hat{Y}$ . There is a new coordinate system centered at the barycenter of the two large primary bodies, called the "rotating coordinate system". This system is represented by the small letters,  $\hat{x}$  and  $\hat{y}$  and is typically rotated in the inertial system by an angle of  $\theta$ . The x-axis of this rotating system is the line joining the two large primaries in the inertial system. The z-axis is aligned

with the angular momentum vector of the second primary's orbit around the first primary, and the y-axis completes the orthogonal triad of the system. The position vectors of the three bodies in the inertial system are represented by the vectors,  $\vec{D}_1$ ,  $\vec{D}_2$  and  $\vec{P}$  respectively. The position vectors of the third body with respect to the two large primaries are represented by the vectors,  $\vec{D}$  and  $\vec{R}$  respectively. Incidentally all these vectors are the same vectors in the rotating system as well, because of the definition of the rotating system.

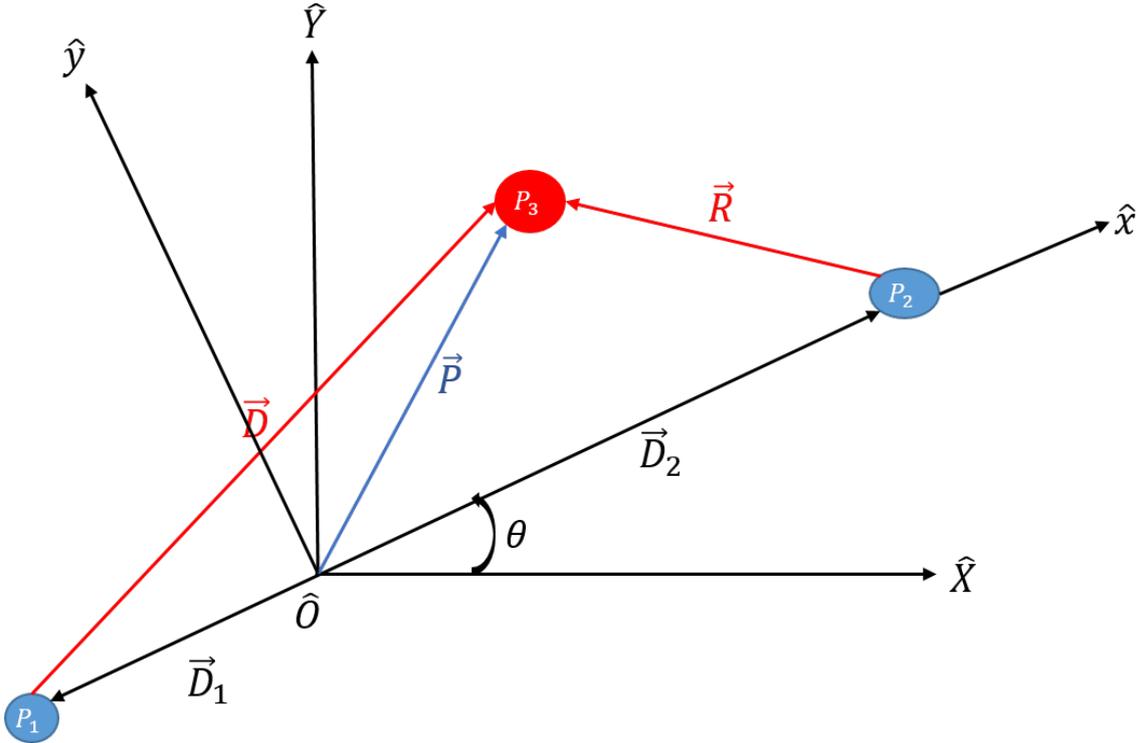


Figure 2.3: A Model of the Restricted Three Body Motion

The derivation of the equations of the CR3BP motion in this rotating frame is presented using the Newtonian mechanics. The Newtonian mechanics are valid only in an inertial frame/system. Using Newton's second law of motion and the universal gravitational law, we can express the motion of the third body in the inertial system as,

$$\begin{aligned}
 m_3^I \ddot{\vec{P}} &= -\frac{\tilde{G}m_1m_3}{D^3} \vec{D} - \frac{\tilde{G}m_2m_3}{R^3} \vec{R} \\
 {}_I \ddot{\vec{P}} &= -\frac{\tilde{G}m_1}{D^3} \vec{D} - \frac{\tilde{G}m_2}{R^3} \vec{R}
 \end{aligned} \tag{2.17}$$

The quantities in the equation (2.17) are highly varying in magnitude, i.e. they are not all in the same order of magnitude. To make the problem more tractable numerically, all the quantities in this equation are normalized using the following characteristic quantities. Several characteristic quantities are defined, one for each type of fundamental measurement encountered in the differential equations. Distance is nondimensionalized using the characteristic length,  $l^*$ , defined as the constant distance between the primaries.

$$l^* = D_1 + D_2 \quad (2.18)$$

where  $D_i$  is the scalar distance from the barycenter to the primary  $P_i$ . Likewise, the characteristic mass,  $m^*$ , is defined as the sum of the masses of  $P_1$  and  $P_2$ , i.e.,

$$m^* = m_1 + m_2 \quad (2.19)$$

The characteristic time is defined as follows:

$$t^* = \sqrt{\frac{l^{*3}}{\tilde{G}m^*}} \quad (2.20)$$

The characteristic time quantity is defined such that the nondimensional mean motion,  $n$ , of  $P_1$  and  $P_2$  is equal to unity. The nondimensional mean motion is defined as:

$$n = Nt^* = \sqrt{\frac{\tilde{G}m^*}{l^{*3}}} \sqrt{\frac{l^{*3}}{\tilde{G}m^*}} = 1 \quad (2.21)$$

where  $N$  is the dimensional mean motion of  $P_2$  in its Keplerian orbit around  $P_1$ . The period,  $P$ , of a circular orbit is related to mean motion as  $P = \frac{2\pi}{n}$ , thus the nondimensional period of the primaries in the CR3BP is  $2\pi$ .

With the characteristic quantities defined, following nondimensional parameters are defined and derived to simplify the equations of motion for CR3BP.

The nondimensional mass of  $P_2$  or mass parameter,  $\mu$ , is defined using the characteristic mass,

$$\mu = \frac{m_2}{m^*} \quad (2.22)$$

In addition to simplifying the equations of motion, the mass ratio,  $\mu$ , is useful to characterize a CR3BP system. The Earth-Moon system, the primary focus of this investigation, is represented in terms of a mass ratio  $\mu = 0.01215$ , which is relatively large compared to other CR3BP systems, for examples, Saturn-Titan where  $\mu = 0.000237$  and Sun-Jupiter with  $\mu = 0.000954$ . This difference in  $\mu$  implies that dynamic features that appear in the Earth-Moon system may not appear in lower mass ratio systems and vice versa.

Additionally, the nondimensional mass of  $P_1$  is defined as,

$$\begin{aligned} m_1 + m_2 = m^* &= m_1 + \mu m^* \\ \implies 1 - \mu &= \frac{m_1}{m^*} \end{aligned} \quad (2.23)$$

The nondimensional time,  $\tau$ , is defined as,

$$\tau = \frac{t}{t^*} \quad (2.24)$$

Here, any derivative with respect to dimensional time,  $t$  is converted to derivative with respect to  $\tau$  as follows:

$$dt = t^* d\tau \quad (2.25)$$

For example, the rotation rate of the rotating frame,  ${}^I W^R$  is nondimensionalized as follows:

$$\begin{aligned}
{}^I W^R &= \frac{d\theta}{dt} \hat{Z} = N \hat{Z} \\
{}^I W^R &= \frac{d\theta}{t^* d\tau} \hat{z} = N \hat{z} \\
{}^I W^R &= \frac{d\theta}{d\tau} \hat{z} = N t^* \hat{z} \\
{}^I W^R &= \frac{d\theta}{d\tau} \hat{z} = n \hat{z}
\end{aligned} \tag{2.26}$$

The goal is to derive simplified equations of motion for the CR3BP using nondimensional quantities from equation (2.17). Non-dimensional vectors in CR3BP are as follow:

$$\begin{aligned}
\vec{\rho} &= \frac{\vec{P}}{l^*} \\
\vec{r} &= \frac{\vec{R}}{l^*} \\
\vec{d} &= \frac{\vec{D}}{l^*}
\end{aligned} \tag{2.27}$$

### Kinematics of CR3BP Motion: A Derivation

The nondimensional position vector for the third body in CR3BP in the rotating frame components is defined as,

$$\vec{\rho} = x\hat{x} + y\hat{y} + z\hat{z} \quad (2.28)$$

We know the basic kinematic equation (BKE) as,

$${}^I\dot{\vec{\rho}} = \frac{{}^R d\vec{\rho}}{d\tau} + {}^I W^R \times \vec{\rho} \quad (2.29)$$

Using the BKE, we get the velocity vector in the inertial system as:

$$\begin{aligned} \frac{{}^R d\vec{\rho}}{d\tau} &= \dot{x}\hat{x} + \dot{y}\hat{y} + \dot{z}\hat{z} \\ {}^I\dot{\vec{\rho}} &= (\dot{x}\hat{x} + \dot{y}\hat{y} + \dot{z}\hat{z}) + [n\hat{z} \times (x\hat{x} + y\hat{y} + z\hat{z})] \\ {}^I\dot{\vec{\rho}} &= (\dot{x}\hat{x} + \dot{y}\hat{y} + \dot{z}\hat{z}) + [n\hat{z} \times x\hat{x} + n\hat{z} \times y\hat{y} + n\hat{z} \times z\hat{z}] \\ {}^I\dot{\vec{\rho}} &= (\dot{x}\hat{x} + \dot{y}\hat{y} + \dot{z}\hat{z}) + [nx\hat{y} - ny\hat{x} + nz(0)] \end{aligned}$$

In the rotating system coordinates:

$${}^I\dot{\vec{\rho}} = (\dot{x} - ny)\hat{x} + (\dot{y} + nx)\hat{y} + \dot{z}\hat{z} \quad (2.30)$$

### Kinematics of CR3BP Motion: Derivation Continued...

Once again using the BKE, the derivation for the acceleration vector in the inertial frame is as follows,

$$\begin{aligned}
 {}^I\ddot{\vec{\rho}} &= \frac{{}^R d^2 \vec{\rho}}{d\tau^2} + 2{}^I W^R \times \frac{{}^R d\vec{\rho}}{d\tau} + {}^I W^R \times {}^I W^R \times \vec{\rho} \\
 {}^I\ddot{\vec{\rho}} &= (\ddot{x}\hat{x} + \ddot{y}\hat{y} + \ddot{z}\hat{z}) + 2[n\hat{z} \times \dot{x}\hat{x} + n\hat{z} \times \dot{y}\hat{y} + n\hat{z} \times \dot{z}\hat{z}] \\
 &\quad + [n\hat{z} \times (n\hat{z} \times x\hat{x} + n\hat{z} \times y\hat{y} + n\hat{z} \times z\hat{z})] \\
 {}^I\ddot{\vec{\rho}} &= (\ddot{x}\hat{x} + \ddot{y}\hat{y} + \ddot{z}\hat{z}) + 2[n\dot{x}\hat{y} - n\dot{y}\hat{x} + n\dot{z}(0)] + [n\hat{z} \times (nx\hat{y} - ny\hat{x} + nz(0))] \\
 {}^I\ddot{\vec{\rho}} &= (\ddot{x}\hat{x} + \ddot{y}\hat{y} + \ddot{z}\hat{z}) + 2[n\dot{x}\hat{y} - n\dot{y}\hat{x}] + [-n^2 x\hat{x} - n^2 y\hat{y}]
 \end{aligned}$$

Thus the inertial acceleration vector in the rotating frame coordinates can be stated as:

$${}^I\ddot{\vec{\rho}} = (\ddot{x} - 2n\dot{y} - n^2 x)\hat{x} + (\ddot{y} + 2n\dot{x} - n^2 y)\hat{y} + \ddot{z}\hat{z} \quad (2.31)$$

## Kinetics of CR3BP Motion: A Derivation

$$\begin{aligned}
 {}^I\ddot{\vec{P}} &= -\frac{\tilde{G}m_1}{D^3}\vec{D} - \frac{\tilde{G}m_2}{R^3}\vec{R} \\
 \frac{{}^I d\vec{P}}{dt} &= \frac{l^* I d\vec{\rho}}{t^* d\tau} \\
 \frac{{}^I d^2\vec{P}}{dt^2} &= \frac{l^* I d^2\vec{\rho}}{t^{*2} d\tau^2} \\
 t^{*2} &= \frac{l^{*3}}{\tilde{G}m^*} \\
 \frac{l^* I d^2\rho}{t^{*2} d\tau^2} &= -\tilde{G}\left[\frac{(1-\mu)m^*}{(dl^*)^3}d\vec{l}^* + \frac{\mu m^*}{(rl^*)^3}\vec{r}l^*\right] \\
 \frac{l^* \tilde{G}m^*}{l^{*3}} I \ddot{\rho} &= -\frac{\tilde{G}m^* l^*}{l^{*3}} \left[\frac{(1-\mu)}{d^3}d\vec{d} + \frac{\mu}{r^3}\vec{r}\right]
 \end{aligned}$$

We get the following expression for the right hand side of equation (2.17) in nondimensional quantities as,

$${}^I\ddot{\rho} = -\frac{(1-\mu)}{d^3}d\vec{d} - \frac{\mu}{r^3}\vec{r} \quad (2.32)$$

Vector expressions for  $\vec{d}$  and  $\vec{r}$  in the rotating frame are,

$$\begin{aligned}
 \vec{d} &= (x + \mu)\hat{x} + y\hat{y} + z\hat{z} \\
 \vec{r} &= (x - 1 + \mu)\hat{x} + y\hat{y} + z\hat{z}
 \end{aligned} \quad (2.33)$$

Now, by substituting these into the equation (2.32), we get the following:

$$\begin{aligned}
 {}^I\ddot{\rho} &= -\frac{(1-\mu)}{d^3}[(x + \mu)\hat{x} + y\hat{y} + z\hat{z}] - \frac{\mu}{r^3}[(x - 1 + \mu)\hat{x} + y\hat{y} + z\hat{z}] \\
 {}^I\ddot{\rho} &= -\frac{(1-\mu)(x + \mu)\hat{x} + (1-\mu)y\hat{y} + (1-\mu)z\hat{z}}{d^3} - \frac{\mu(x - 1 + \mu)\hat{x} + \mu y\hat{y} + \mu z\hat{z}}{r^3}
 \end{aligned} \quad (2.34)$$

### Kinetics of CR3BP Motion: Derivation Continued...

Finally, we get the following equation for the right hand side in terms of 3 position components in the rotating frame as,

$${}^I\ddot{\rho} = \left[ -\frac{(1-\mu)(x+\mu)}{d^3} - \frac{\mu(x-1+\mu)}{r^3} \right] \hat{x} + \left[ -\frac{(1-\mu)y}{r^3} - \frac{\mu y}{r^3} \right] \hat{y} + \left[ -\frac{(1-\mu)z}{d^3} - \frac{\mu z}{r^3} \right] \hat{z} \quad (2.35)$$

Substituting equation (2.35) into equation (2.31) we get the following equations of motion for the CR3BP expressed in the rotating frame coordinates:

$$\begin{aligned} \ddot{x} - 2n\dot{y} - n^2x &= -\frac{(1-\mu)(x+\mu)}{d^3} - \frac{\mu(x-1+\mu)}{r^3} \\ \ddot{y} + 2n\dot{x} - n^2y &= -\frac{(1-\mu)y}{d^3} - \frac{\mu y}{r^3} \\ \ddot{z} &= -\frac{(1-\mu)z}{d^3} - \frac{\mu z}{r^3} \end{aligned} \quad (2.36)$$

Equations (2.36) are expressed in rotating coordinates and the scalar acceleration and velocity terms are evaluated relative to the rotating coordinate frame. The definitions of  $d$  and  $r$  in equations (2.33) indicate that their magnitudes reflect the relative distance from each primary to  $P_3$ .

The system of equations (2.36) are nonlinear and coupled. There are no known closed form analytical solutions for this system of equations. However, there are some solutions to this system of equations under some known conditions. Specifically, there are known gravitational equilibrium points in this system and there are known families of solutions of orbital motion about these gravitational equilibrium points. To predict motion in the CR3BP system, typically, the equilibrium point solutions are derived and are then converted to the inertial system solutions for further analysis to assess the suitability for a spacecraft mission. It is also important to note that the CR3BP system is time-independent, since time does not explicitly appear in the system of equations governing the motion. The CR3BP motion solution structures become time-dependent when converted to an inertial system.

### 2.3.2 The Pseudo-Potential Function

Typically in Newtonian mechanics, the acceleration component due to the potential energy is derived by taking the derivative of a gravitational potential function. In the CR3BP model, there is no such gravitational potential function inherently available and the rotating-frame of the CR3BP is not an inertial frame. So, as a substitute for a real gravitational potential function in the rotating-frame, we identify a pseudo-potential function, the derivative of which provides acceleration components based purely on the position components.

$$U^* = \frac{1-\mu}{d} + \frac{\mu}{r} + \frac{1}{2}n^2(x^2 + y^2) \quad (2.37)$$

The newly defined pseudo-potential function incorporates terms that accommodate the rotation of the coordinate frame. The equations of motion are then available in a more succinct form as follows,

$$\begin{aligned} \ddot{x} - 2n\dot{y} &= \frac{\partial U^*}{\partial x} \\ \ddot{y} + 2n\dot{x} &= \frac{\partial U^*}{\partial y} \\ \ddot{z} &= \frac{\partial U^*}{\partial z} \end{aligned} \quad (2.38)$$

### 2.3.3 The Gravitational Equilibrium (Lagrange) Points

The gravitational equilibrium points are by definition the solutions at which the system of differential equations of motion for CR3BP evaluate to zero. By definition, the velocity and acceleration of  $P_3$  in the rotating frame are zero at the equilibrium points. From equations (2.38), we can see that, at the equilibrium state, the gradient of the the pseudo-potential is equal to zero. This implies the following,

$$\begin{aligned}
\frac{\partial U^*}{\partial x_{eq}} &= 0 \\
\frac{\partial U^*}{\partial y_{eq}} &= 0 \\
\frac{\partial U^*}{\partial z_{eq}} &= 0
\end{aligned} \tag{2.39}$$

where  $x_{eq}$ ,  $y_{eq}$ , and  $z_{eq}$  are equilibrium position components of  $P_3$ . These equations evaluate to the following:

$$\frac{\partial U^*}{\partial x_{eq}} = -\frac{(1-\mu)(x_{eq} + \mu)}{d_{eq}^3} - \frac{\mu(x_{eq} + 1 - \mu)}{r_{eq}^3} + n^2 x_{eq} = 0 \tag{2.40}$$

$$\frac{\partial U^*}{\partial y_{eq}} = -\frac{(1-\mu)y_{eq}}{d_{eq}^3} - \frac{\mu y_{eq}}{r_{eq}^3} + n^2 y_{eq} = 0 \tag{2.41}$$

$$\frac{\partial U^*}{\partial z_{eq}} = -\frac{(1-\mu)z_{eq}}{d_{eq}^3} - \frac{\mu z_{eq}}{r_{eq}^3} = 0 \tag{2.42}$$

It's apparent from equation (2.42) that the equilibrium solutions are planar, since it is only satisfied when  $z_{eq} = 0$ . Similarly, equation (2.41) can be satisfied when  $y_{eq} = 0$ , which means that there are one or more equilibrium solutions along the x-axis of the rotating frame. To locate the colinear equilibrium solutions along the x-axis, substitute  $z_{eq} = y_{eq} = 0$  and  $n = 1$  into equation (2.40), which yields the following result,

$$\begin{aligned}
0 &= -\frac{(1-\mu)(x_{eq} + \mu)}{(x_{eq} + \mu)^3} - \frac{\mu(x_{eq} - 1 + \mu)}{(x_{eq} - 1 + \mu)^3} + x_{eq} \\
x_{eq} &= \frac{(1-\mu)(x_{eq} + \mu)}{(x_{eq} + \mu)^3} + \frac{\mu(x_{eq} - 1 + \mu)}{(x_{eq} - 1 + \mu)^3}
\end{aligned} \tag{2.43}$$

Equation (2.43) does not have a closed form analytical solution. However, it is a fifth order equation in terms of  $x_{eq}$  and can therefore have five solutions. Two of the five solutions have proven to be imaginary and are ignored. To find the rest of the 3 real solutions for  $x_{eq}$ , Figure 2.4 is used as a reference.

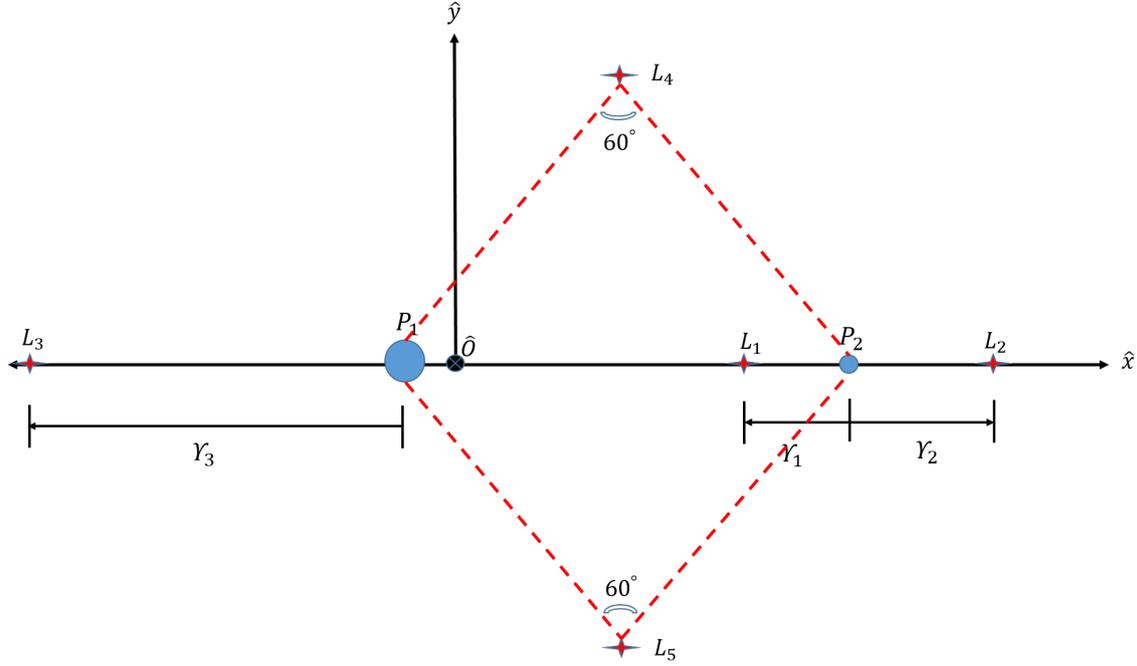


Figure 2.4: The Gravitational Equilibrium (Lagrange) Points

The points  $L_i$  in the figure are the equilibrium solutions, also known as the Lagrange points of CR3BP. To solve equation (2.43) for  $L_1$  and  $L_2$ , the distance of  $P_3$  from the smaller primary is used as  $\gamma_1$  and  $\gamma_2$  respectively. To solve for  $L_3$ , the distance of  $P_3$  from the larger primary,  $\gamma_3$  is used. So the possible values of  $x_{eq}$  are specified as follows:

$$\begin{aligned}
 x_1 &= 1 - \mu - \gamma_1 \\
 x_2 &= 1 - \mu + \gamma_2 \\
 x_3 &= -\mu - \gamma_3
 \end{aligned}
 \tag{2.44}$$

These equations, when substituted into equation (2.43), yield the following:

$$\begin{aligned}
1 - \mu - \gamma_1 &= \frac{(1 - \mu)}{(1 - \gamma_1)^2} + \frac{\mu}{\gamma_1^2} \\
1 - \mu + \gamma_2 &= \frac{(1 - \mu)}{(1 + \gamma_2)^2} + \frac{\mu}{\gamma_2^2} \\
-\mu - \gamma_3 &= \frac{(1 - \mu)}{\gamma_3^2} + \frac{\mu}{(1 + \gamma_3)^2}
\end{aligned} \tag{2.45}$$

The Newton-Raphson method can be used to solve each of the three equations above. Two additional non-colinear equilibrium solutions when  $y_{eq} \neq 0$ . When  $d = r$ , two possible values for the equilibrium solutions exist,

$$\begin{aligned}
x_{4,5} &= \frac{1}{2} - \mu \\
y_{4,5} &= \pm \frac{\sqrt{3}}{2}
\end{aligned} \tag{2.46}$$

$L_4$  and  $L_5$  are denoted as the ‘‘triangular’’ or ‘‘equilateral’’ Lagrange points.

### 2.3.4 The Jacobi Integral Constant and Zero-Velocity Surfaces

The system of equations (2.38) are nonlinear and coupled and there is no known closed form analytical solution. However, with the aid of the pseudo-potential function, one can derive at least one constant of CR3BP motion from the differential equations of motion. To derive such a constant of motion, the dot product of acceleration with velocity is taken.

$${}^R \rho'' \cdot {}^R \rho' = \dot{x}\ddot{x} + \dot{y}\ddot{y} + \dot{z}\ddot{z} = \frac{\partial U^*}{\partial x} \dot{x} + \frac{\partial U^*}{\partial y} \dot{y} + \frac{\partial U^*}{\partial z} \dot{z} \tag{2.47}$$

Here, ' and '' indicate derivatives in the rotating frame with respect to the nondimensional time  $\tau$ . The pseudo-potential,  $U^*$ , is autonomous and only a function of position, therefore, the right side of equation (2.47) equals the total scalar derivative  $\frac{dU^*}{d\tau}$ . Consequently, equation (2.47) is integrated resulting in,

$$\begin{aligned}\frac{1}{2}(\dot{x}^2 + \dot{y}^2 + \dot{z}^2) &= U^* + \textit{Integration constant} \\ \frac{1}{2}(\dot{x}^2 + \dot{y}^2 + \dot{z}^2) &= U^* - C\end{aligned}\tag{2.48}$$

where the integration constant  $C$  is defined with a negative sign by convention. Equation (2.48) is more succinctly expressed as,

$$v^2 = 2U^* - C\tag{2.49}$$

where  $v$  is the scalar Euclidean magnitude of the velocity of  $P_3$ ,  $v = \|\dot{\rho}^R\|_2$ , as viewed by a rotating observer. Equation (2.49) is denoted Jacobi's integral and the integration constant  $C$  labeled the Jacobi constant after the mathematician, Carl Gustav Jacob Jacobi. This constant represents an energy-like quantity in the CR3BP. The Jacobi constant has various uses, for example, approximating the energy change necessary for transfers and as a check on the accuracy of numerical integration. The Jacobi constant yields powerful insights into behavior within the CR3BP particularly when combined with particular solutions to the differential equations.

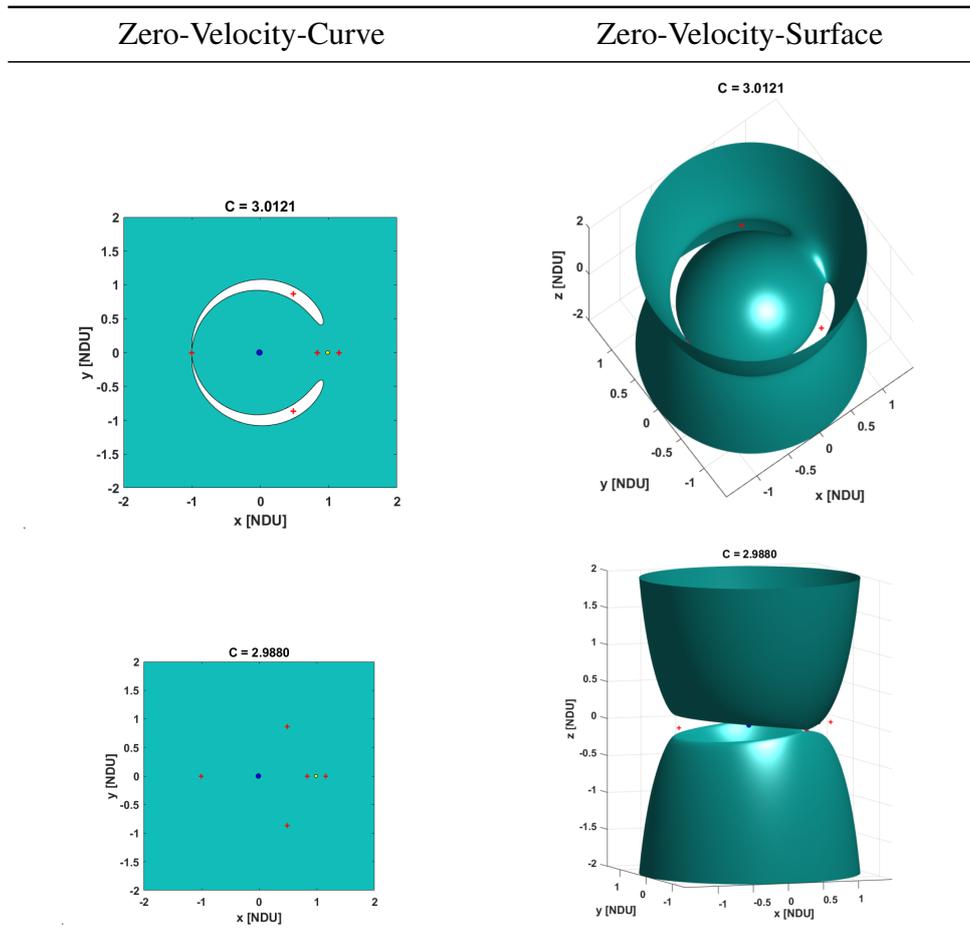
The Equation (2.49) provides powerful insight into natural motion dictated by the equations of motion (2.38) of CR3BP model. We can derive possible regions of motion depending on the Jacobi constant value. When we set velocity in this equation to zero (meaning that motion beyond this point is not feasible), and fix the Jacobi constant value to a desired value, we can derive regions where motion obeys that particular Jacobi constant value. To illustrate this concept, the zero velocity surfaces for the five known equilibrium points (Lagrange points) are shown in Figure tables 2.1 and 2.2.

In these figures, the Zero-Velocity-Curve is showing permissible motion in the color in the x-y plane, and the Zero-Velocity-Surface is showing the permissible motion in three dimensions. These two concepts are important for identification of periodic orbits in any CR3BP system.

Table 2.1:  $L_1$  and  $L_2$  Zero Velocity Curves and Surfaces

Zero-Velocity-Curve	Zero-Velocity-Surface

Table 2.2:  $L_3$  and  $L_{4,5}$  Zero Velocity Curves and Surfaces



### 2.3.5 Periodic Orbits in the CR3BP System

Periodic orbits are bounded solutions for a given dynamical system. The periodic orbits in the pure two-body model are the closed conic sections. There are no such standard solutions in the CR3BP model. The periodic orbits in CR3BP come about predominantly from known conditions, such as about the five known gravitational equilibrium points. In the CR3BP system, if a particular solution exists, there typically exists a family of such solutions. Most famous of such families of orbits are the Lagrange point orbits, such as the Lyapunov, and halo orbit families around  $L_1$  and  $L_2$ , axial and vertical orbit families around all five Lagrange points [30]. There are also Moon centered orbits such as the Direct Retrograde Orbits (DROs) and Low-Prograde Orbits (LPOs) and system encompassing Resonant orbits [30]. In this dissertation, Lagrange point orbit families are the primary focus, due to their popularity and ubiquity to establish the verification procedures for the techniques developed for orbit identification on the VR platform.

### 2.3.6 Transformation of CR3BP Motion to an Inertial System

The CR3BP model serves as a good model for uncovering motion structures such as the periodic orbits and manifolds for orbital transfers under the influence two primary bodies. However, to assess the inertial effects of eclipses (of the Sun, Earth, Moon etc), other perturbing effects such as the solar radiation pressure and nth body gravity, these motion structures of the CR3BP system are more conveniently transformed to an inertial system. Typically the transformation of motion structures from the CR3BP to an inertial system is performed with the aid of the instantaneous Keplerian orbital parameters of the second primary. For example a periodic orbit from the cislunar CR3BP to the Earth-centered J2000 inertial system is performed as follows,

1. Center the position coordinates of the CR3BP state ( $\vec{r}$ ) on Earth,

$$\vec{r}_{PC} = \begin{bmatrix} x_{PC} \\ y_{PC} \\ z_{PC} \end{bmatrix} \quad (2.50)$$

$$\vec{r}_{PC} = \vec{r} + \mu$$

2. Retrieve the dimensional position ( $\vec{R}$ ) and velocity ( $\vec{V}$ ) vectors of the Moon from JPL SPICE toolkit [26] and derive the instantaneous axes of the rotating frame as follows,

$$\begin{aligned} \hat{x} &= \frac{\vec{R}}{R} \\ \hat{y} &= \frac{\vec{R} \times \vec{V}}{\|\vec{R} \times \vec{V}\|} \\ \hat{z} &= \hat{x} \times \hat{y} \end{aligned} \quad (2.51)$$

3. Compose the position transformation matrix as follows,

$${}^I C^R = \begin{bmatrix} \hat{x} & \hat{y} & \hat{z} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \quad (2.52)$$

4. Compute the nondimensional inertial position vector,  $\vec{P}$ , using the position transformation matrix as follows,

$$\vec{P} = {}^I C^R \vec{r}_{PC} \quad (2.53)$$

5. If dimensional inertial position is desired, multiply the nondimensional inertial position vector with the magnitude of the instantaneous position vector,  $\vec{R}$ , of the second primary,

$$\tilde{\vec{P}} = \|\vec{R}\|_2 \vec{P} \quad (2.54)$$

### CR3BP-J2000 Transformation (Velocity): A Method [22]

1. The instantaneous angular velocity,  ${}^I W^R$ , of the rotating frame with respect to the J2000 frame is defined as follows from the Keplerian motion,

$$\begin{aligned} {}^I W^R &= \dot{\theta} \hat{z} \\ \dot{\theta} &= \frac{\vec{h}}{R^2} = \frac{\vec{R} \times \vec{V}}{R^2} \end{aligned} \quad (2.55)$$

2. Apply the Basic Kinematic Equation (BKE) to the Earth-centered position vector of CR3BP state to define the velocity vector,

$$\frac{{}^I d}{dt} \vec{r}_{PC} = \frac{{}^R d}{dt} \vec{r}_{PC} + {}^I W^R \times \vec{r}_{PC} \quad (2.56)$$

$$\frac{{}^I d}{dt} \vec{r}_{PC} = (\dot{x}_{PC} - \dot{\theta} y_{PC}) \hat{x} + (\dot{y}_{PC} + \dot{\theta} x_{PC}) \hat{y} + \dot{z}_{PC} \hat{z} \quad (2.57)$$

3. Compute the nondimensional inertial velocity,  $\vec{V}$ , as follows,

$$\vec{V} = \begin{bmatrix} \dot{X}_{PC} \\ \dot{Y}_{PC} \\ \dot{Z}_{PC} \end{bmatrix} = \begin{bmatrix} \dot{\theta} C_{12} & -\dot{\theta} C_{11} & 0 & C_{11} & C_{12} & C_{13} \\ \dot{\theta} C_{22} & -\dot{\theta} C_{21} & 0 & C_{21} & C_{22} & C_{23} \\ \dot{\theta} C_{32} & -\dot{\theta} C_{31} & 0 & C_{31} & C_{32} & C_{33} \end{bmatrix} \begin{bmatrix} x_{PC} \\ y_{PC} \\ z_{PC} \\ \dot{x}_{PC} \\ \dot{y}_{PC} \\ \dot{z}_{PC} \end{bmatrix} \quad (2.58)$$

4. If dimensional velocity is desired, compute the dimensional time and velocity as follows,

$$T = \sqrt{\frac{R^3}{\mu_{Earth} + \mu_{Moon}}} \quad (2.59)$$

$$\tilde{\vec{V}} = \frac{T}{R} \vec{V} \quad (2.60)$$

## Chapter 3

### Trajectory Design and Optimization Techniques in the Cislunar System

The system of differential equations (2.36) governing the CR3BP motion do not have known closed form analytical solutions. To identify periodic orbit and orbital transfer solutions in this system, numerical methods are used. The primary numerical method employed is the differential corrections method, which is gradient based. The numerical differential corrections methods such as the multi-variate Newton methods are highly susceptible to initial conditions. Arbitrary initial conditions do not typically yield desired motion. Hence the problems addressed are two-fold,

1. Employment of numerical methods to identify orbital motion solutions robustly in presence of bad initial conditions.
2. Techniques and methods to provide good initial conditions to the numerical methods.

In this chapter, foundations for these robust numerical methods and the methods themselves are discussed and presented.

#### 3.1 The State Transition Matrix

The state transition matrix is a linear artifact of a system of equations, which could either be a linear system or a nonlinear system. The state transition matrix allows for identifying solutions of a nonlinear system with respect to a known reference trajectory.

The state of the CR3BP motion is defined as,

$$\vec{s} = \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \quad (3.1)$$

The system of differential equations defining the CR3BP motion can be generalized as follows,

$$\dot{\vec{s}} = f(\tau, \vec{s}, \lambda) \quad (3.2)$$

Here,  $\tau$  represents the nondimensional time of the CR3BP, and  $\lambda$  represents the set of independent parameters such as  $\mu$  and  $n$  ( $= 1$  in CR3BP). Let some initial state,  $\vec{s}_0^*$  generate a valid reference trajectory. Let  $\delta\vec{s}$  be a small variation to this reference initial state. So we have the perturbed differential equation as,

$$\dot{\vec{s}} = \dot{\vec{s}}_0^* + \delta\dot{\vec{s}} = f(\tau, \vec{s}_0^* + \delta\vec{s}, \lambda)$$

When we take the Taylor series expansion of this equation and truncate the expansion to first order terms, we have,

$$\dot{\vec{s}}_0^* + \delta\dot{\vec{s}} = f(\tau, \vec{s}_0^*, \lambda) + \left. \frac{\partial f}{\partial \vec{s}} \right|_{\vec{s}_0^*} \delta\vec{s}$$

$$\delta\dot{\vec{s}} = \left. \frac{\partial f}{\partial \vec{s}} \right|_{\vec{s}_0^*} \delta\vec{s}$$

We have the following definitions from this equation,

$$A(\tau) = \left. \frac{\partial f}{\partial \vec{s}} \right|_{\vec{s}_0^*} \implies \text{Jacobian of the system} \quad (3.3)$$

$$\delta\dot{\vec{s}} = A(\tau)\delta\vec{s} \quad (3.4)$$

Equation (3.3) is called the Jacobian of the system represented by  $f(\tau, \vec{s}, \lambda)$ . The solution to Equation (3.4) in a linear sense can be written in terms of the matrix derivative of the state of the system at time  $\tau$  with respect to the state of the system at the initial time  $\tau_0$  as follows,

$$\delta \dot{\vec{s}} = \frac{\partial \vec{s}(\tau)}{\partial \vec{s}(\tau_0)} \delta \vec{s} \quad (3.5)$$

Since  $\vec{s}(\tau_0)$  and  $\tau$  are independent of each other, we can express the matrix derivative  $\frac{\partial \vec{s}(\tau)}{\partial \vec{s}(\tau_0)}$ , as follows,

$$\frac{d}{d\tau} \frac{\partial \vec{s}(\tau)}{\partial \vec{s}(\tau_0)} = \frac{d}{d\vec{s}(\tau_0)} \frac{\partial \vec{s}(\tau)}{\partial \tau} = \frac{d}{d\vec{s}(\tau_0)} \dot{\vec{s}}(\tau) \quad (3.6)$$

When the chain rule of mathematical differentiation is applied to this equation, we have,

$$\frac{d}{d\tau} \frac{\partial \vec{s}(\tau)}{\partial \vec{s}(\tau_0)} = \frac{d}{d\vec{s}(\tau_0)} \dot{\vec{s}}(\tau) = \frac{\partial f(\tau, \vec{s}, \lambda)}{\partial \vec{s}(\tau)} \frac{\partial \vec{s}(\tau)}{\partial \vec{s}(\tau_0)} = A(\tau) \frac{\partial \vec{s}(\tau)}{\partial \vec{s}(\tau_0)} \quad (3.7)$$

From this equation, we define the state transition matrix of the linear system as follows,

$$\Phi(\tau, \tau_0) = \frac{\partial \vec{s}(\tau)}{\partial \vec{s}(\tau_0)} \quad (3.8)$$

The equation governing evolution of the state transition matrix is as follows,

$$\dot{\Phi}(\tau, \tau_0) = A(\tau) \Phi(\tau, \tau_0) \quad (3.9)$$

### 3.1.1 The State Transition Matrix for the CR3BP System

The variations in the CR3BP system can be expressed as,

$$\begin{aligned} \delta \ddot{x} - 2n\delta \dot{y} &= \frac{\partial U^*}{\partial x \partial x} \delta x + \frac{\partial U^*}{\partial x \partial y} \delta y + \frac{\partial U^*}{\partial x \partial z} \delta z \\ \delta \dot{y} + 2n\delta \dot{x} &= \frac{\partial U^*}{\partial y \partial x} \delta x + \frac{\partial U^*}{\partial y \partial y} \delta y + \frac{\partial U^*}{\partial y \partial z} \delta z \\ \delta \ddot{z} &= \frac{\partial U^*}{\partial z \partial x} \delta x + \frac{\partial U^*}{\partial z \partial y} \delta y + \frac{\partial U^*}{\partial z \partial z} \delta z \end{aligned} \quad (3.10)$$

From equations (3.10) form the Jacobian matrix,  $A(\tau)$ , of the CR3BP system,

$$A(\tau) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{\partial U^*}{\partial x \partial x} & \frac{\partial U^*}{\partial x \partial y} & \frac{\partial U^*}{\partial x \partial z} & 0 & 2n & 0 \\ \frac{\partial U^*}{\partial y \partial x} & \frac{\partial U^*}{\partial y \partial y} & \frac{\partial U^*}{\partial y \partial z} & -2n & 0 & 0 \\ \frac{\partial U^*}{\partial z \partial x} & \frac{\partial U^*}{\partial z \partial y} & \frac{\partial U^*}{\partial z \partial z} & 0 & 0 & 0 \end{bmatrix} \quad (3.11)$$

The second partial derivatives of the pseudo-potential function in equation (3.11) are defined as,

$$\begin{aligned} \frac{\partial U^*}{\partial x \partial x} &= n^2 - \frac{1-\mu}{d^3} - \frac{\mu}{r^3} + \frac{3(1-\mu)(x+\mu)^2}{d^5} + \frac{3\mu(x-1+\mu)^2}{r^5} \\ \frac{\partial U^*}{\partial y \partial y} &= n^2 - \frac{1-\mu}{d^3} - \frac{\mu}{r^3} + \frac{3(1-\mu)y^2}{d^5} + \frac{3\mu y^2}{r^5} \\ \frac{\partial U^*}{\partial z \partial z} &= -\frac{1-\mu}{d^3} - \frac{\mu}{r^3} + \frac{3(1-\mu)z^2}{d^5} + \frac{3\mu z^2}{r^5} \\ \frac{\partial U^*}{\partial x \partial y} &= \frac{\partial U^*}{\partial y \partial x} = \frac{3(1-\mu)(x+\mu)y}{d^5} + \frac{3\mu(x-1+\mu)y}{r^5} \\ \frac{\partial U^*}{\partial x \partial z} &= \frac{\partial U^*}{\partial z \partial x} = \frac{3(1-\mu)(x+\mu)z}{d^5} + \frac{3\mu(x-1+\mu)z}{r^5} \\ \frac{\partial U^*}{\partial y \partial z} &= \frac{\partial U^*}{\partial z \partial y} = \frac{3(1-\mu)zy}{d^5} + \frac{3\mu zy}{r^5} \end{aligned} \quad (3.12)$$

### 3.1.2 The Monodromy Matrix and Orbit Stability

The monodromy matrix of a periodic orbit is defined as,

$$M = \Phi(\tau_0 + P, \tau_0) \quad (3.13)$$

In equation (3.13),  $P$  is the one single orbital period of a periodic orbit. Following are important properties of a monodromy matrix [27],

1. The Eigen values of the monodromy matrix always come in reciprocal pairs.
2. The determinant of the monodromy matrix is always unity.

3. The monodromy matrix is symplectic.

### 3.2 Trajectory Targeting Methods

The system of differential equations (2.36) governing CR3BP motion is nonlinear and coupled. Random initial conditions for this system does not yield desired motion. To derive desired motion, one can target the desired conditions of motion and apply differential corrections to the random initial conditions with respect to the desired conditions of motion. For periodic orbit identification, the target condition is the periodicity, i.e. the motion should get back to the initial state after a certain period of time called the orbital period. For orbital transfer trajectory identification, feasibility of the trajectory (according to CR3BP equations of motion) and reaching the target orbit are the target conditions.

#### 3.2.1 The Multi-Variate Newton Differential Corrections

In this method, a free variable vector, and a constraint vector are employed. The free variable vector is typically composed of the initial state guess variables, integration times for the state, expected epochs for boundaries of the initial state variables. The free variable vector could

contain slack variables in some cases. A minimal free variable vector is designated as,

$$\vec{X} = \begin{bmatrix} \vec{s}_1 \\ \vec{s}_2 \\ \cdot \\ \cdot \\ \cdot \\ \vec{s}_n \\ T_1 \\ T_2 \\ \cdot \\ \cdot \\ \cdot \\ T_{n-1} \end{bmatrix} \quad (3.14)$$

The constraint vector contains conditions of the desired motion. Typically these conditions are enforcing continuity at the trajectory segment boundaries, conditions on velocity and time-of-flight at a minimum. The vector can contain various other conditions such as enforcing a minimum distance from the primaries in the orbital motion. A minimal constraint vector is designated as,

$$\vec{F}(\vec{X}) = \begin{bmatrix} \vec{F}_1(\vec{X}) \\ \vec{F}_2(\vec{X}) \\ \cdot \\ \cdot \\ \cdot \\ \vec{F}_m(\vec{X}) \end{bmatrix} \quad (3.15)$$

Typically the dimension of the free variable vector is designated as  $n$  and that of the constraint variable vector is designated as  $m$ . The goal is to identify a desired initial conditions vector,  $\vec{X}^*$ , that drives the constraints vector to zero, i.e.,  $\vec{F}(\vec{X}^*) = 0$ . An iterative process based on Newton's method is employed to identify  $\vec{X}^*$ . The Newton's method involves expressing the

constraint functions vector,  $\vec{F}(\vec{X}^*) = 0$ , as Taylor series expansion, and then truncating the higher order terms,

$$F(\vec{X}) = F(\vec{X}_0) + \frac{\partial F(\vec{X}_0)}{\partial \vec{X}_0}(\vec{X} - \vec{X}_0) \quad (3.16)$$

In this equation,  $\vec{X}_0$  is the initial-guess for the solution free variable vector,  $\vec{X}^*$ .  $\frac{\partial F(\vec{X}_0)}{\partial \vec{X}_0}$  can be designated as  $DF(\vec{X}_0)$  for brevity in notation. For a solution,  $F(\vec{X}) = 0$ , hence, equation (3.16) can be generalized as,

$$\vec{X}_{i+1} = \vec{X}_i - DF(\vec{X}_i)^{-1}F(\vec{X}_i) \quad (3.17)$$

where  $\vec{X}_i$  is the current iteration of the free-variable vector,  $\vec{X}_{i+1}$  is the next iteration of the free-variable vector, and  $F(\vec{X}_i)$  is the value of the current constraint vector as evaluated after propagating the equations of motion from the initial condition  $\vec{X}_i$ . Since the current free variable vector and value of the current constraint vector are known,  $DF(\vec{X}_i)$  is also available. An iteration process is initiated by employing equation (3.17) as the update equation. Iterations continue until  $F(\vec{X}_{i+1}) = F(\vec{X}^*) = 0$  to some acceptable tolerance.

Equation (3.17) in it's basic form is only valid if the number of variables in the free variable vector,  $n$ , is equal to the number constraints in the constraint vector,  $m$ . However, if  $n > m$ , as is typical with spacecraft trajectory design problems, there exist infinitely many solutions to equation (3.17). To resolve ambiguity, one can employ a minimum norm solution to the initial-guess update problem as,

$$\vec{X}_{i+1} = \vec{X}_i - DF(\vec{X}_i)^T(DF(\vec{X}_i)DF(\vec{X}_i)^T)^{-1}F(\vec{X}_i) \quad (3.18)$$

The minimum norm solution to the initial-guess update problem allows for as close of a solution to the first initial-guess,  $\vec{X}_0$ , as possible.

### 3.2.2 The Single-Shooting Method

The single shooting method is the simplest of shooting methods. The single-shooting method utilizes a single integrated trajectory segment to solve a two-point boundary value problem.

To employ the free-variable-constraint method [22], we can start with an expression for the variation in state at final time from a variation at the initial time as,

$$\delta\vec{s}_F = \Phi(\tau_F, \tau_0)|_{\vec{s}^*} \delta\vec{s}_0 + \frac{\partial\vec{s}}{\partial\tau}|_{\vec{s}^*} \delta\tau \quad (3.19)$$

where,  $\delta\vec{s}_F$ , is the variation in the final state from the nominal/desired state. And,  $\delta\vec{s}_0$ , is the variation in the initial state from the current initial/reference state. Finally,  $\delta\tau$  is the variation in the time-of-flight from the nominal/reference value.

$$\delta\vec{s}_F = \begin{bmatrix} \delta x_F \\ \delta y_F \\ \delta z_F \\ \delta \dot{x}_F \\ \delta \dot{y}_F \\ \delta \dot{z}_F \\ \delta \tau_F \end{bmatrix} = \begin{bmatrix} x_F - x_{desired} \\ y_F - y_{desired} \\ z_F - z_{desired} \\ \dot{x}_F - \dot{x}_{desired} \\ \dot{y}_F - \dot{y}_{desired} \\ \dot{z}_F - \dot{z}_{desired} \end{bmatrix} \quad (3.20)$$

$$\delta\vec{s}_0 = \begin{bmatrix} \delta x_0 \\ \delta y_0 \\ \delta z_0 \\ \delta \dot{x}_0 \\ \delta \dot{y}_0 \\ \delta \dot{z}_0 \end{bmatrix} = \begin{bmatrix} x_0 - x^* \\ y_0 - y^* \\ z_0 - z^* \\ \dot{x}_0 - \dot{x}^* \\ \dot{y}_0 - \dot{y}^* \\ \dot{z}_0 - \dot{z}^* \end{bmatrix} \quad (3.21)$$

$$\delta\tau = \tau_F - \tau^* \quad (3.22)$$

Equation (3.19) can be rewritten as,

$$\begin{bmatrix} \delta x_F \\ \delta y_F \\ \delta z_F \\ \delta \dot{x}_F \\ \delta \dot{y}_F \\ \delta \dot{z}_F \end{bmatrix} = \begin{bmatrix} \frac{\partial x_F}{\partial x_0} & \frac{\partial x_F}{\partial y_0} & \frac{\partial x_F}{\partial z_0} & \frac{\partial x_F}{\partial \dot{x}_0} & \frac{\partial x_F}{\partial \dot{y}_0} & \frac{\partial x_F}{\partial \dot{z}_0} \\ \frac{\partial y_F}{\partial x_0} & \frac{\partial y_F}{\partial y_0} & \frac{\partial y_F}{\partial z_0} & \frac{\partial y_F}{\partial \dot{x}_0} & \frac{\partial y_F}{\partial \dot{y}_0} & \frac{\partial y_F}{\partial \dot{z}_0} \\ \frac{\partial z_F}{\partial x_0} & \frac{\partial z_F}{\partial y_0} & \frac{\partial z_F}{\partial z_0} & \frac{\partial z_F}{\partial \dot{x}_0} & \frac{\partial z_F}{\partial \dot{y}_0} & \frac{\partial z_F}{\partial \dot{z}_0} \\ \frac{\partial \dot{x}_F}{\partial x_0} & \frac{\partial \dot{x}_F}{\partial y_0} & \frac{\partial \dot{x}_F}{\partial z_0} & \frac{\partial \dot{x}_F}{\partial \dot{x}_0} & \frac{\partial \dot{x}_F}{\partial \dot{y}_0} & \frac{\partial \dot{x}_F}{\partial \dot{z}_0} \\ \frac{\partial \dot{y}_F}{\partial x_0} & \frac{\partial \dot{y}_F}{\partial y_0} & \frac{\partial \dot{y}_F}{\partial z_0} & \frac{\partial \dot{y}_F}{\partial \dot{x}_0} & \frac{\partial \dot{y}_F}{\partial \dot{y}_0} & \frac{\partial \dot{y}_F}{\partial \dot{z}_0} \\ \frac{\partial \dot{z}_F}{\partial x_0} & \frac{\partial \dot{z}_F}{\partial y_0} & \frac{\partial \dot{z}_F}{\partial z_0} & \frac{\partial \dot{z}_F}{\partial \dot{x}_0} & \frac{\partial \dot{z}_F}{\partial \dot{y}_0} & \frac{\partial \dot{z}_F}{\partial \dot{z}_0} \end{bmatrix} \begin{bmatrix} \delta x_0 \\ \delta y_0 \\ \delta z_0 \\ \delta \dot{x}_0 \\ \delta \dot{y}_0 \\ \delta \dot{z}_0 \end{bmatrix} + \begin{bmatrix} \dot{x}_F \\ \dot{y}_F \\ \dot{z}_F \\ \ddot{x}_F \\ \ddot{y}_F \\ \ddot{z}_F \end{bmatrix} \delta \tau \quad (3.23)$$

Equation (3.23) is the most general form differential corrections applied in the single-shooting method. An example targeting method using this scheme is now presented. In this example, a single-shooting method is used to identify Lyapunov orbits in the CR3BP model of orbital motion. Lyapunov orbits are planar periodic orbits that exist in the vicinity of the  $L_1$ ,  $L_2$  and  $L_3$  Lagrange points. These orbits exist only in the fundamental  $x$ - $y$  plane of motion of the secondary around the primary. Another characteristic of these orbits is that the velocity is perpendicular to the  $x$ -axis at the two  $x$ -axis crossings. The perpendicular crossing property is leveraged to identify these orbits with the single-shooting method. The perpendicular  $x$ -axis crossing condition enforces the following desired numerical conditions,

$$\vec{s}_{desired} = \begin{bmatrix} x_{desired} \\ y_{desired} \\ \dot{x}_{desired} \end{bmatrix} = \begin{bmatrix} x^* \\ 0 \\ 0 \end{bmatrix} \quad (3.24)$$

where,  $x^*$  is the desired  $x$  component of position at the perpendicular crossing of  $x$ -axis.  $\dot{x}$  is zero at the perpendicular crossing since motion is perpendicular to  $x$ -axis. Finally, since this condition is at the  $x$ -axis crossing, the  $y$  component of position is zero. Now for these conditions, equation (3.23) becomes,

$$\begin{bmatrix} \delta x_F \\ \delta y_F \\ \delta \dot{x}_F \end{bmatrix} = \begin{bmatrix} \frac{\partial x_F}{\partial x_0} & \frac{\partial x_F}{\partial y_0} \\ \frac{\partial y_F}{\partial x_0} & \frac{\partial y_F}{\partial y_0} \\ \frac{\partial \dot{x}_F}{\partial x_0} & \frac{\partial \dot{x}_F}{\partial y_0} \end{bmatrix} \begin{bmatrix} \delta \dot{x}_0 \\ \delta \dot{y}_0 \end{bmatrix} + \begin{bmatrix} \dot{x}_F \\ \dot{y}_F \\ \ddot{x}_F \end{bmatrix} \delta \tau \quad (3.25)$$

So the final differential correction equation for identifying Lyapunov orbits at the perpendicular  $x$ -axis crossing is:

$$\begin{bmatrix} \delta \dot{x}_0 \\ \delta \dot{y}_0 \\ \delta \tau \end{bmatrix} = \begin{bmatrix} \frac{\partial x_F}{\partial x_0} & \frac{\partial x_F}{\partial y_0} & \dot{x}_F \\ \frac{\partial y_F}{\partial x_0} & \frac{\partial y_F}{\partial y_0} & \dot{y}_F \\ \frac{\partial \dot{x}_F}{\partial x_0} & \frac{\partial \dot{x}_F}{\partial y_0} & \ddot{x}_F \end{bmatrix}^{-1} \begin{bmatrix} x_F - x^* \\ y_F \\ \dot{x}_F \end{bmatrix} \quad (3.26)$$

Typically in practice the Lyapunov orbits are identified by starting at a desired  $x$  position component and targeting the next  $x$ -axis crossing for the final state. Once the desired half of the orbit is corrected, by manipulating the  $x$  and  $y$  components of velocity and half of the time-of-flight of the full periodic orbit, the full orbit is obtained by numerically propagating for the remaining half of the orbit. Grebow [11] provided the single-shooting differential equations for identifying the Lyapunov, halo, axial and vertical Lagrange point orbits.

One common issue with the single-shooting method is that the nonlinearities of the CR3BP system are not captured well by the method, because of the linear approximations of equation (3.19). The problem is exacerbated when the time-interval over which the method is applied to the CR3BP system is increased. This problem can be addressed by a second kind of shooting methods called the multiple-shooting method described in the next section.

### 3.2.3 The Multiple-Shooting Method

The multiple-shooting method divides the interval over which a solution is sought into several smaller intervals, propagates the fully nonlinear equations of motion in each of the smaller intervals, and imposes path constraints along the segment interval extremes and the boundary conditions on the initial and final state. The method is effective for distributing the nonlinearity and numerical instability along the solution over the single shooting method. Figures 3.1 and 3.2 are showing a notional initial-guess and the corresponding solution for the multiple-shooting method. In Figure 3.1, the trajectory is divided into  $n - 1$  segments time-wise. Initially the trajectory is discontinuous, but eventually the trajectory becomes continuous within a certain tolerance upon an iterative application of a multi-variate Newton method called the free-variable-constraint method to the initial-guess.

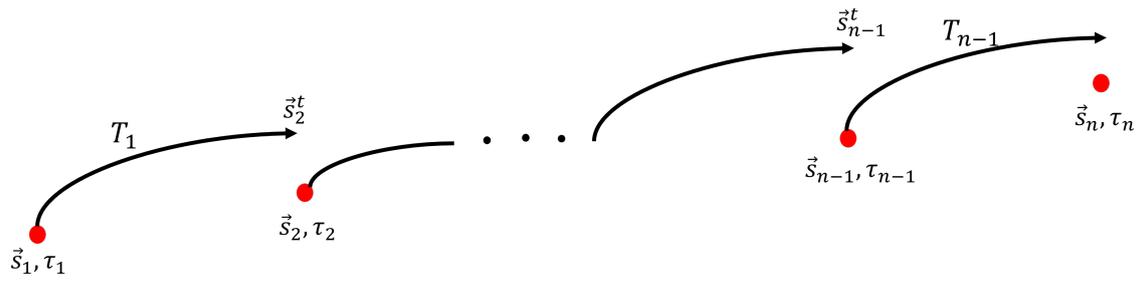


Figure 3.1: A Depiction of an Initial-Guess for the Multiple-Shooting Method

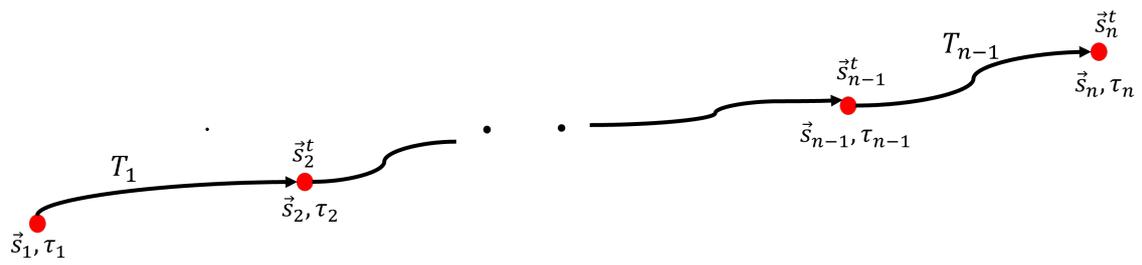


Figure 3.2: A Depiction of a Solution for the Multiple-Shooting Method

The general free-variable vector is designated as,

$$\vec{X} = \begin{bmatrix} \vec{s}_1 \\ \vec{s}_2 \\ \cdot \\ \cdot \\ \cdot \\ \vec{s}_n \\ T_1 \\ T_2 \\ \cdot \\ \cdot \\ T_{n-1} \\ \tau_1 \\ \tau_2 \\ \cdot \\ \cdot \\ \cdot \\ \tau_{n-1} \\ \tau_n \end{bmatrix} \quad (3.27)$$

where  $\vec{s}_i, i = 1, \dots, n$  is the initial state vector of segment  $i$ ,  $T_i, i = 1, \dots, n - 1$  is the time-of-flight in segment  $i$  and  $\tau_i, i = 1, \dots, n$  is the epoch of the initial state in segment  $i$ . The





methods are guaranteed to be locally optimal. However, the keys to indirect methods are the doubling of unknown variables to solve for (i.e. the costate variables) and the need for deriving analytical equations for costate from the equations of motion. The classical formulation of the indirect methods also does not allow for enforcement of path constraints for orbital motion, when necessary. Due to these limitations, an alternate approach of solving optimization problems, especially with the advent of computers. In the direct optimization method, the expected orbital path is discretized into multiple sub-segments, where each segment is implicitly integrated. The resulting sub-segments may not be continuous and hence the path constraints to ensure continuity along the segment boundaries and within the segments are enforced. The discretization process is called “direct transcription”. The direct transcription results in a vector of discretized input variables that need to satisfy the path constraints of orbital motion while also optimizing for fuel and/or time-of-flight. This problem formulation results in a nonlinear programming problem (NLP) that can be solved on a computer efficiently.

### 3.3.1 The Generalized Collocation Method

The generalized collocation method is a type of collocation method that can approximate the unknown solution function with any arbitrary degree orthogonal polynomial basis function. The goal of this method is to use arbitrary degree orthogonal polynomial functions to construct a nonlinear programming problem that can be solved using any NLP solver. In this algorithm, a candidate trajectory is divided into  $n$  arcs time-wise as,

$$\Pi : t_0 < t_1 < t_2 < \dots < t_n = t_f \quad (3.31)$$

and each  $i^{th}$  arc is approximated by an  $N^{th}$  degree polynomial as shown,

$$P_i(\tau) = C_i \times \begin{bmatrix} 1 & \tau & \tau^2 & \dots & \tau^N \end{bmatrix} \quad (3.32)$$

Here,  $\tau$  is the normalized time that spans the time interval  $[-1, 1]$  for all of the trajectory arcs and  $C_i$  are the coefficients of the polynomial approximation. Note that dimension of  $P_i(\tau)$  and hence that of  $C_i$  is  $l \times (N + 1)$ , where  $l$  is the number of state variables and  $N$  is the degree of

the polynomial. The conversion from dimensional time  $t$  to normalized time  $\tau$  in each arc,  $i$ , is given by,

$$\tau = \frac{2}{t_{i+1} - t_i}(t - t_i) - 1 \quad (3.33)$$

Because of time normalization, the dimensional time derivatives of state variables are computed from the normalized time derivatives of state variables as follows,

$$\dot{s}_i = \frac{\Delta t_i}{2} f(\tau_i, s_i) \quad (3.34)$$

where  $f$  includes the set of Equations (2.36). If there are  $l$  variables in the state describing the trajectory arc, then the polynomial  $P_i$  is an  $l$ -dimensional vector, and the matrix of coefficients  $C_i$  is of dimension  $l \times (N+1)$ . The points where states are collocated are selected corresponding to the roots of a particular representation of polynomials  $P_i(\tau)$  [25].

The version implemented in this dissertation uses the 7<sup>th</sup> degree Legendre-Gauss (LG) polynomials due to the simplicity of implementation while also preserving the accuracy of the solutions. The matrix of polynomial coefficients in each arc,  $C_i$ , is constructed using states at one or more root nodes on arc  $i$  for the polynomial. For odd degree polynomials, each arc is subdivided into  $\frac{N+1}{2}$  variable nodes and  $\frac{N-1}{2}$  defect points. The index of a variable node or defect point is indicated by the subscript  $j = 1, 2, \dots, N$ , and the variable nodes and defect points occur at odd and even numbered  $j$ , respectively. The purpose of dividing roots of the LG polynomial into odd and even nodes is to have the freedom to vary the variable nodes as dictated by the underlying algorithm while enforcing CR3BP dynamics at the defect points to ensure that the resulting polynomial is a good approximation of the solution for the CR3BP dynamics. Figure 3.3 shows the division of a polynomial segment into variable and defect points/nodes.

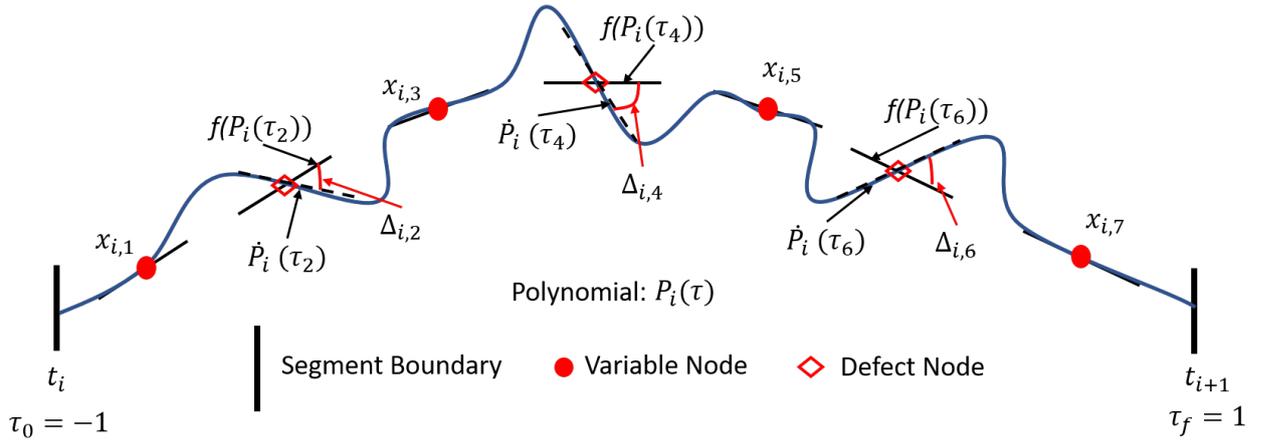


Figure 3.3: Node Placement Strategy for Collocation with 7th Degree Polynomial

A nonlinear programming problem (NLP) is constructed as follows,

1. Construct a nonlinear constraints function,
  - (a) Compute the matrix of coefficients,  $C_i$ , from the variable nodes in each arc.
  - (b) Using the matrix of coefficients,  $C_i$ , compute the states at the defect nodes.
  - (c) Compute the CR3BP dynamics violations at all the defect nodes.
  - (d) Compute the segment continuity violations.
  - (e) In case of periodic orbit identification, compute the periodicity violation.
  - (f) In case of transfer trajectory identification, compute the initial and final state violations for the transfer.
  - (g) Compose all the violations into an equality constraint vector.
2. Construct a cost function. This is problem specific. The desired cost functions are listed Sections 4.3 and 4.6 for the problems in this dissertation.
3. Define the NLP problem using the nonlinear constraints and cost function in any desired programming language NLP solver. In this dissertation, *fmincon* NLP solver in MATLAB and Python IPOPT are employed.
4. Define the initial-guess construction mechanisms for each problem addressed with the NLP problem. The initial-guess construction mechanisms for each of the problems in this dissertation are listed in Sections 4.3 and 4.6.

### Computation of the Coefficient Matrix, $C_i$

For each polynomial root node,  $j$ , along a given trajectory arc,  $i$ , the state and state derivatives are given as,

$$s_{i,j} = P_i(\tau_j) = P_{i,j}, \quad i = 1, 2, \dots, n \text{ and } j = 1, 2, \dots, N \quad (3.35)$$

where  $P_i$  is as defined in Equation (3.32), and

$$\dot{s}_{i,j} = \dot{P}_{i,j} = \frac{\Delta t_i}{2} f(\tau_j, s_{i,j}) \quad (3.36)$$

In Equation (3.36),  $f$  includes the set of Equations (2.36). Based on these equations, we can arrive at the following expression for variable nodes:

$$C_i[\tau \ \dot{\tau}] = \begin{bmatrix} s_{i,1} & s_{i,3} & s_{i,5} & s_{i,7} & \dot{s}_{i,1} & \dot{s}_{i,3} & \dot{s}_{i,5} & \dot{s}_{i,7} \end{bmatrix} \quad (3.37)$$

where  $\tau$  and  $\dot{\tau}$  represent matrices:

$$\tau = \begin{bmatrix} 1 & 1 & 1 & 1 \\ \tau_1 & \tau_3 & \tau_5 & \tau_7 \\ \tau_1^2 & \tau_3^2 & \tau_5^2 & \tau_7^2 \\ \tau_1^3 & \tau_3^3 & \tau_5^3 & \tau_7^3 \\ \tau_1^4 & \tau_3^4 & \tau_5^4 & \tau_7^4 \\ \tau_1^5 & \tau_3^5 & \tau_5^5 & \tau_7^5 \\ \tau_1^6 & \tau_3^6 & \tau_5^6 & \tau_7^6 \\ \tau_1^7 & \tau_3^7 & \tau_5^7 & \tau_7^7 \end{bmatrix} \quad (3.38)$$

and

$$\dot{\tau} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 2\tau_1 & 2\tau_3 & 2\tau_5 & 2\tau_7 \\ 3\tau_1^2 & 3\tau_3^2 & 3\tau_5^2 & 3\tau_7^2 \\ 4\tau_1^3 & 4\tau_3^3 & 4\tau_5^3 & 4\tau_7^3 \\ 5\tau_1^4 & 5\tau_3^4 & 5\tau_5^4 & 5\tau_7^4 \\ 6\tau_1^5 & 6\tau_3^5 & 6\tau_5^5 & 6\tau_7^5 \\ 7\tau_1^6 & 7\tau_3^6 & 7\tau_5^6 & 7\tau_7^6 \end{bmatrix} \quad (3.39)$$

The states at the variable nodes are supplied as part of the initial guess. The unknown coefficients are then computed by solving Equation (3.37) as follows:

$$C_i = \begin{bmatrix} s_{i,1} & s_{i,3} & s_{i,5} & s_{i,7} & \dot{s}_{i,1} & \dot{s}_{i,3} & \dot{s}_{i,5} & \dot{s}_{i,7} \end{bmatrix} A^{-1} \quad (3.40)$$

where  $A$  is given by  $[\tau \ \dot{\tau}]_{8 \times 8}$ .

### Computation of States at Defect Nodes

The normalized times at the location of the defect points are used to create the matrices  $B$  and  $D$  as follows:

$$B = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ -1 & \tau_2 & \tau_4 & \tau_6 & 1 \\ 1 & \tau_2^2 & \tau_4^2 & \tau_6^2 & 1 \\ -1 & \tau_2^3 & \tau_4^3 & \tau_6^3 & 1 \\ 1 & \tau_2^4 & \tau_4^4 & \tau_6^4 & 1 \\ -1 & \tau_2^5 & \tau_4^5 & \tau_6^5 & 1 \\ 1 & \tau_2^6 & \tau_4^6 & \tau_6^6 & 1 \\ -1 & \tau_2^7 & \tau_4^7 & \tau_6^7 & 1 \end{bmatrix} \quad (3.41)$$

and,

$$D = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 2\tau_2 & 2\tau_4 & 2\tau_6 \\ 3\tau_2^2 & 3\tau_4^2 & 3\tau_6^2 \\ 4\tau_2^3 & 4\tau_4^3 & 4\tau_6^3 \\ 5\tau_2^4 & 5\tau_4^4 & 5\tau_6^4 \\ 6\tau_2^5 & 6\tau_4^5 & 6\tau_6^5 \\ 7\tau_2^6 & 7\tau_4^6 & 7\tau_6^6 \end{bmatrix} \quad (3.42)$$

The first and last columns of B are included to compute the states at the boundary nodes along an arc because, in the LG node placement scheme, the boundary nodes of a segment are not used to construct the polynomial for that arc. It is convenient to compute the boundary node states as well so that they are available toward the evaluation of constraints for feasibility of the trajectory. The matrix B and the polynomial coefficients are then employed to construct the states at the defect points and boundary nodes as follows:

$$\begin{bmatrix} s_{i,0} & s_{i,2} & s_{i,4} & s_{i,6} & s_{i,f} \end{bmatrix} = C_i B \quad (3.43)$$

### Computation of Constraint Violations for Periodic Orbit Identification

The constraint matrix,  $F$ , is computed using the state at the defect and boundary points, the coefficients  $C_i$ , and matrix D as follows:

$$\vec{F} = \begin{bmatrix} s_{1,0} - s_{n,f} \\ s_{2,0} - s_{1,f} \\ \cdot \\ \cdot \\ \cdot \\ s_{n,0} - s_{n-1,f} \\ [C_1 D - \begin{bmatrix} \dot{s}_{1,2} & \dot{s}_{1,4} & \dot{s}_{1,6} \end{bmatrix}] \times W_1 \\ [C_2 D - \begin{bmatrix} \dot{s}_{2,2} & \dot{s}_{2,4} & \dot{s}_{2,6} \end{bmatrix}] \times W_2 \\ \cdot \\ \cdot \\ \cdot \\ [C_n D - \begin{bmatrix} \dot{s}_{n,2} & \dot{s}_{n,4} & \dot{s}_{n,6} \end{bmatrix}] \times W_n \end{bmatrix} \quad (3.44)$$

The first  $n$  constraints enforce continuity and periodicity on the trajectory. The second  $n$  constraints enforce dynamical feasibility (Equations (2.36)). The matrices  $W_i$  represent the quadrature weights of the LG polynomial roots.

## Chapter 4

### Trajectory Design and Optimization Techniques on the Virtual Reality Platform

One of the main pain points in trajectory design and optimization is the lack of universally standard and intuitive procedures for construction of initial-guesses to the numerical algorithms. The focus of this dissertation is two fold,

1. Provide procedures on the virtual reality platform to construct good initial-guesses to the numerical trajectory design and optimization algorithms.
2. Design and implement robust numerical algorithms to use the initial-guesses from the virtual reality platform to identify solutions.

#### 4.1 The Simulation of the Cislunar CR3BP System

The first of the two goals this dissertation is achieved entirely on a virtual reality platform. To achieve this goal, the cislunar CR3BP system is simulated on the VR platform in it's full 3D glory. The cislunar CR3BP system, with the Earth, the Moon, the five Lagrange points and the three coordinate axes of the rotating coordinate frame is simulated. The system is simulated as a Unity3D [28] C# application on a Windows-10 computer to be played out on a HTC VIVE Pro VR headset [29]. Figure 4.1 is showing the elements of the cislunar CR3BP system played out on the VR headset to the human user.

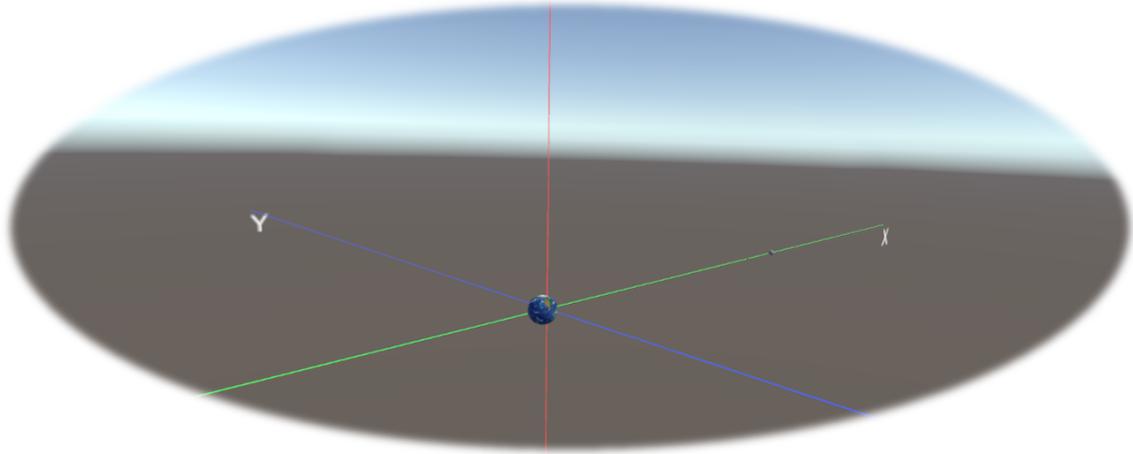


Figure 4.1: A Depiction of the Earth-Moon CR3BP System in VR

## 4.2 Periodic Orbit Identification Techniques

There are four different techniques developed in this dissertation for periodic orbit identification in the cislunar CR3BP system. For techniques II through IV, a CR3BP look-ahead trajectory is displayed to the human operator, and this trajectory is computed using a fixed step Runge-Kutta-4 integration method implemented in C# inside the Unity applications.

### 4.2.1 Technique-I: Initial-Guess from a Non-Dynamical User-Drawing

In this technique, a human user immersed in the cislunar CR3BP system is prompted to draw a periodic orbit from their imagination with the HTC VIVE controller. The VR application for this technique records only position points along the human drawing. The drawing is missing the velocity and time-of-flight part of the orbital state for the periodic orbit represented by the drawing. The challenge for the numerical trajectory optimization algorithm is to solve for the missing velocity and time-of-flight from the drawing. The goal of the corrector algorithms is to find a fit for the human drawing to a closest dynamically feasible CR3BP periodic orbit. The human drawings are known to have distortions from the intended periodic orbits represented by the drawings. For example, Figures ?? and ?? are illustrating the said distortions in the human drawings collected in the VR application. Figure 4.2 is showing the VR user interaction menu for Technique-I.



Figure 4.2: VR User-Interaction Menu for Technique-I

In the VR menu there are actions that a human user can take by pointing the laser pointer from the right controller to the appropriate menu item and pressing trigger, to generate the initial-guess in Technique-I. Figure 4.3 is illustrating a flow-chart for the screenplay of the VR application for Technique-I. The screenplay is described after this flow-chart.

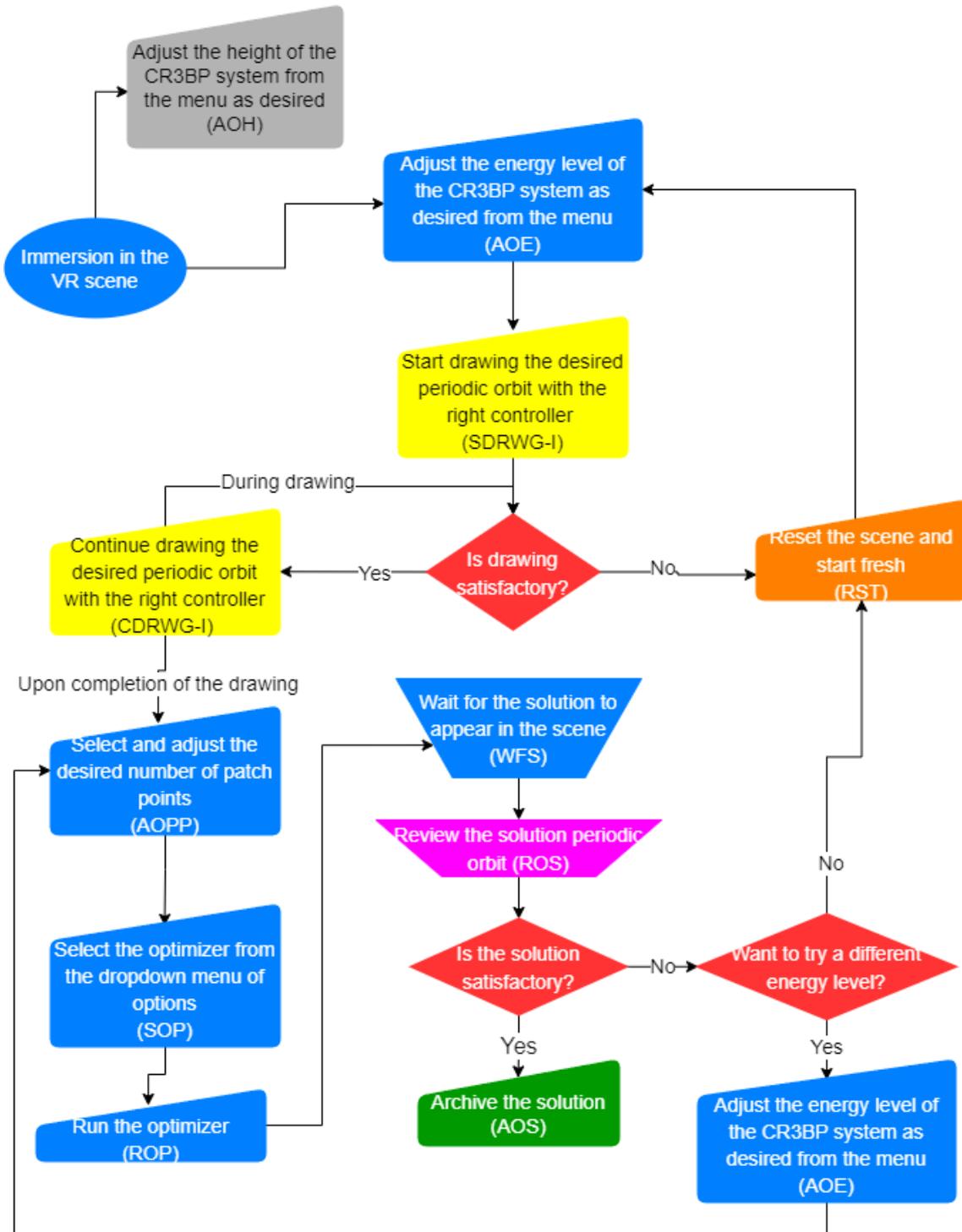


Figure 4.3: A Flow-Chart of Periodic Orbit Identification in Technique-I

## Identification of Periodic Orbits in Technique-I: A Screenplay

The screenplay for the VR scene for periodic identification in Technique-I is,

1. Mount the VR headset and enter the VR scene with the right controller turned on. Human user sees the scene in Figure 4.1.
2. Adjust the height of the system as desired from the system menu. The system menu can be accessed from the menu button of the right controller. Refer to Figure A.1 for a snapshot of this action.
3. Adjust the energy level of CR3BP system as desired from the system menu. Refer to Figure A.2 for a snapshot of this action.
4. Close the system menu to start drawing. The drawings can only be drawn when menu is not active in the scene.
5. Start drawing the periodic orbit of desired shape with the right controller trigger button. The drawing appears in the scene in yellow color. Refer to Figure A.10 for a snapshot of this action.
6. Check if the drawing so far is satisfactory for the shape of the orbit originally intended. If not, reset the scene and start over. If yes, continue drawing the orbit. Refer to Figure A.11 for a snapshot of this action.
7. While drawing the orbit, continuously keep verifying the drawing for originally intended shape.
8. Upon completion of the drawing, select and adjust the desired number of patch points along the initial-guess using the system menu. The number of patch points should capture the curvature of the initial-guess properly to find a good solution orbit. Refer to Figure A.4 for a snapshot of this action.

### Identification of Periodic Orbits in Technique-I: A Screenplay continued...

9. Next, select the desired optimizer from the drop-down menu of options for the optimizer in the system menu. It is okay to experiment with all the options for a single drawing, but each option can only be used one at a time. Refer to Figure A.5 for a snapshot of this action.
10. Next, run the optimizer from the system menu button dedicated to it. Refer to Figure A.6 for a snapshot of this action.
11. Now, wait for the optimizer to finish and the solution orbit to appear in the scene in color magenta. Refer to Figure A.7 for a snapshot of this action.
12. When the solution periodic orbit appears in the scene, review it for the quality of the solution with respect to the initial-guess, which also simultaneously appears in the scene. The quality of the solution can also be assessed based on the solution parameters that appear along with the scene in text form. Of particular importance is the constraint violation parameter. Typically, a value of at or below  $1e - 6$  is considered to be good. Refer to Figure A.8 for a snapshot of this action.
13. If the solution orbit is satisfactory, then archive the solution using the system menu option for it. Refer to Figure A.9 for a snapshot of this action.
14. If the solution is not satisfactory, but experimentation with different energy levels is desired, then reset the solution and adjust to the new energy level and return to step 8 and follow the steps from there. If no experimentation is desired, reset the scene and return to step 3.

#### 4.2.2 Technique-II: Initial-Guess from a Dynamical User-Drawing

In this technique, the human user is still prompted to draw a CR3BP periodic orbit. However, the user is also shown a look-ahead CR3BP trajectory that emanates from the user drawing. This look-ahead CR3BP trajectory is expected to help the user provide a better drawing. The

initial-guess in this technique is captured from sampling the first one or two points along the look-ahead CR3BP trajectory. Hence, the initial-guess constructed in this technique contains valid CR3BP position, velocity and time-of-flight information for a drawing. However, the CR3BP based initial-guess could still be discontinuous and may not represent a perfect CR3BP periodic orbit. The generalized collocation method is applicable here as well. In addition to the collocation method, multiple-shooting method can also be applied to identify periodic orbits from these initial-guesses. Figure 4.4 is showing the VR user-interaction menu for Technique-II.



Figure 4.4: VR User-Interaction Menu for Technique-II

In the VR menu there are actions that a human user can take by pointing the laser pointer from the right controller to the appropriate menu item and pressing trigger, to generate the initial-guess in Technique-II. Figure 4.5 is showing the flow-chart for this technique as implemented in VR, followed by the screenplay of the VR application for this technique.

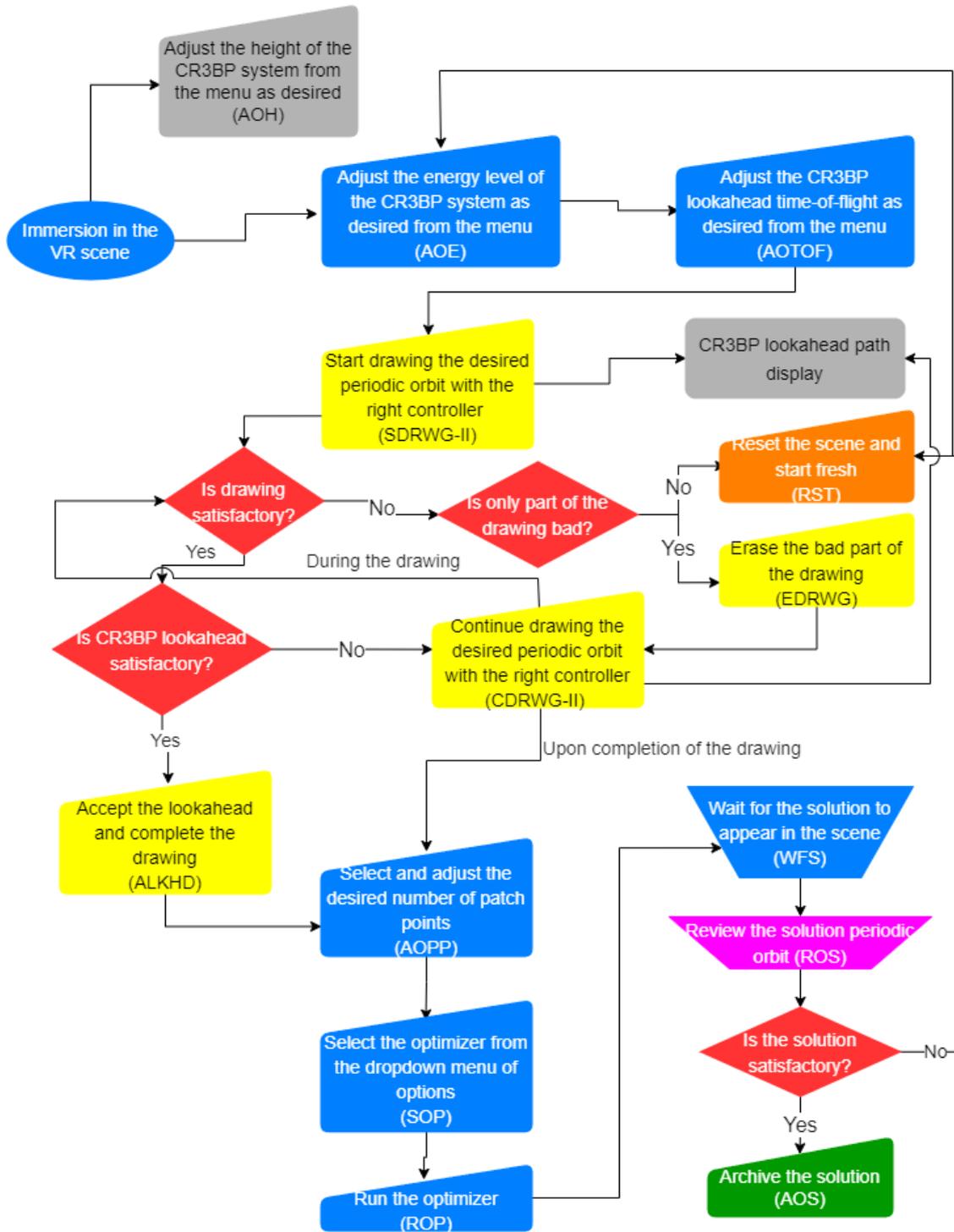


Figure 4.5: A Flow-Chart of Periodic Orbit Identification in Technique-II

## Identification of Periodic Orbits in Technique-II: A Screenplay

The screenplay for the VR scene for periodic identification in Technique-II is,

1. Mount the VR headset and enter the VR scene with the right controller turned on. Human user sees the scene in Figure 4.1.
2. Adjust the height of the system as desired from the system menu. The system menu can be accessed from the menu button of the right controller. Refer to Figure A.1 for a snapshot of this action.
3. Adjust the energy level of CR3BP system as desired from the system menu. This action is fixed per drawing/orbit, i.e., once the energy is fixed for the drawing, it cannot be adjusted again until the drawing is erased or reset. Refer to Figure A.2 for a snapshot of this action.
4. Adjust the CR3BP look-ahead time-of-flight as desired from the system menu. This action is not fixed per drawing/orbit. The value can be adjusted throughout the drawing process. Refer to Figure A.3 for a snapshot of this action.
5. Close the system menu to start drawing. The drawings can only be drawn when menu is not active in the scene.
6. Start drawing the periodic orbit of desired shape with the right controller trigger button. The drawing appears in the scene in yellow color. Refer to Figure A.10 for a snapshot of this action.
7. After a slight delay, the CR3BP look-ahead trajectory emanating from the drawing should appear in color white. Refer to Figure A.10 for a snapshot of this action.

## Identification of Periodic Orbits in Technique-II: A Screenplay continued....

8. Check if the drawing is satisfactory. If it's not, check if only part of the drawing is bad. If only part of the drawing is bad, then erase the bad part and continue drawing the desired orbit. If all of the drawing is bad, then reset the scene and start fresh from Steps 2 through 4. If the drawing so far is satisfactory, then verify if the CR3BP look-ahead trajectory is satisfactory. If yes, accept the look-ahead trajectory into the initial-guess. If no, continue drawing the desired periodic orbit with the right controller. Refer to Figures A.11 or A.12 for a snapshot of these actions.
9. Throughout the drawing process, the CR3BP look-ahead path appears along with the drawing. So keep verifying if the initial-guess can be completed with the look-ahead path or if the drawing is bad partly or wholly and acting accordingly.
10. Upon completion of the drawing, or acceptance of the look-ahead path to complete the drawing, select and adjust the desired number of patch points along the initial-guess using the system menu. The number of patch points should capture the curvature of the initial-guess properly to find a good solution orbit. Refer to Figure A.4 for a snapshot of this action.
11. Next, select the desired optimizer from the drop-down menu of options for the optimizer in the system menu. It is okay to experiment with all the options for a single drawing, but each option can only be used one at a time. Refer to Figure A.5 for a snapshot of this action.
12. Next, run the optimizer from the system menu button dedicated to it. Refer to Figure A.6 for a snapshot of this action.

### Identification of Periodic Orbits in Technique-II: A Screenplay continued....

13. Now, wait for the optimizer to finish and the solution orbit to appear in the scene in color magenta. Refer to Figure A.7 for a snapshot of this action.
14. When the solution periodic orbit appears in the scene, review it for the quality of the solution with respect to the initial-guess, which also simultaneously appears in the scene. The quality of the solution can also be assessed based on the solution parameters that appear along with the scene in text form. Of particular importance is the constraint violation parameter. Typically, a value of at or below  $1e - 6$  is considered to be good. Refer to Figure A.8 for a snapshot of this action.
15. If the solution orbit is satisfactory, then archive the solution using the system menu option for it. Refer to Figure A.9 for a snapshot of this action.
16. If the solution is not satisfactory, reset the scene and start from steps 2 through 4 again.

#### 4.2.3 Technique-III: Initial-Guess from Patch-Points

In this technique drawings are not employed, but instead the human user immersed in the cislunar CR3BP system is prompted to select position points along the intended periodic orbit. After the first two position patch points are placed by the user in the VR scene, the user is allowed to manipulate a lookahead CR3BP trajectory emanating from the first two patch points, by maneuvering the velocity direction vector so the CR3BP trajectory can match both the patch points along the way. The user can then continue to place patch points in the scene continuously to complete a periodic orbit initial-guess from the patch points. Figure 4.6 is showing the VR user interaction menu for Technique-III.



Figure 4.6: VR User-Interaction Menu for Technique-III

In the VR menu there are actions that a human user can take by pointing the laser pointer from the right controller to the appropriate menu item and pressing trigger, to generate the initial-guess in Technique-III. Figure 4.7 is showing the flow-chart for this technique. The screenplay for this technique as implemented in VR is shown immediately after the Figure 4.7.

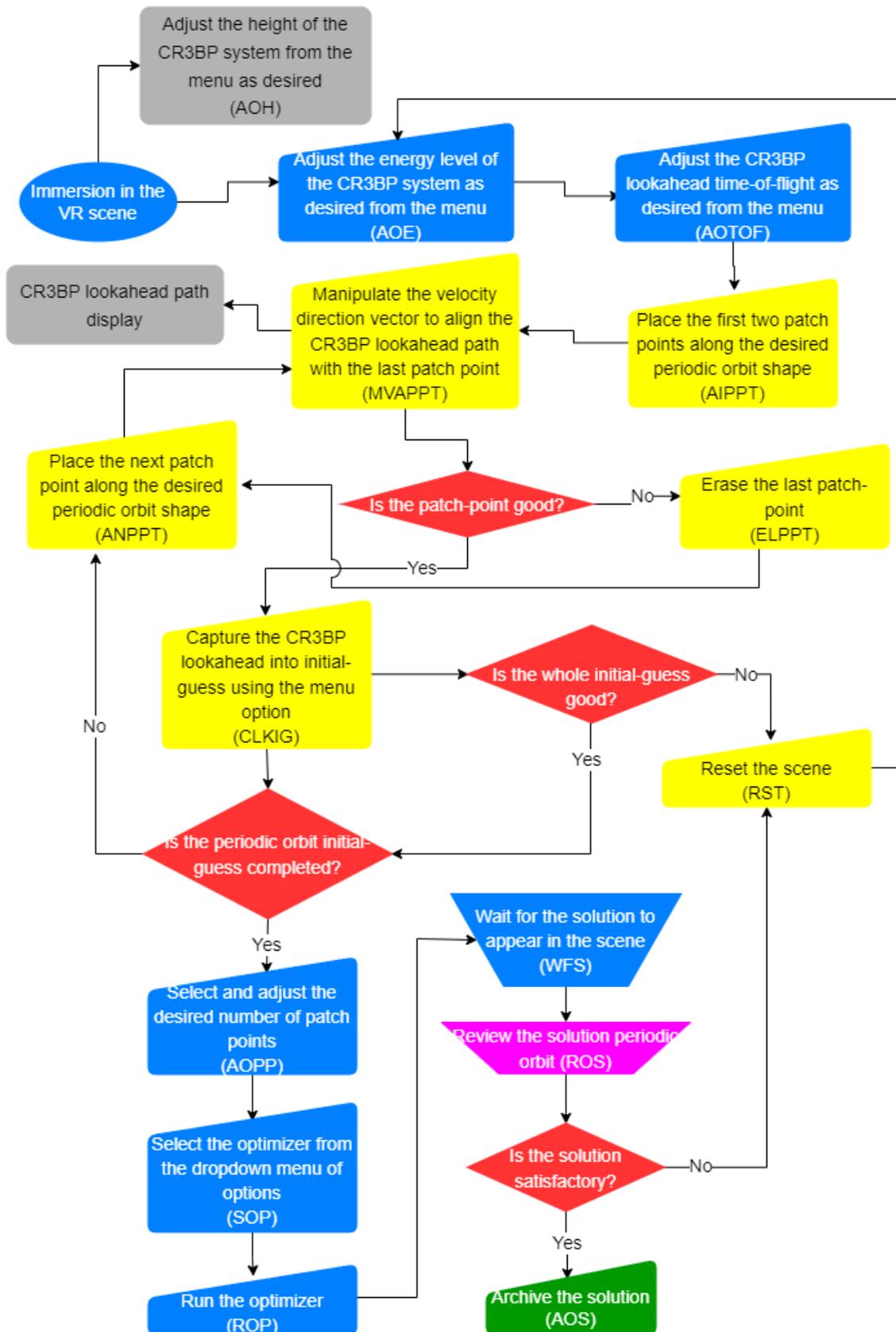


Figure 4.7: A Flow-Chart of Periodic Orbit Identification in Technique-III

### Identification of Periodic Orbits in Technique-III: A Screenplay

The screenplay for the VR scene for periodic identification in Technique-III is,

1. Mount the VR headset and enter the VR scene with the right controller turned on. Human user sees the scene in Figure 4.1.
2. Adjust the height of the system as desired from the system menu. The system menu can be accessed from the menu button of the right controller. Refer to Figure A.1 for a snapshot of this action.
3. Adjust the energy level of CR3BP system as desired from the system menu. This action is fixed per orbit, i.e., once the energy is fixed for the orbit, it cannot be adjusted again until the all the patch points are erased or reset. Refer to Figure A.2 for a snapshot of this action.
4. Adjust the CR3BP look-ahead time-of-flight as desired from the system menu. This action is not fixed per orbit. The value can be adjusted throughout the drawing process. Refer to Figure A.3 for a snapshot of this action.
5. Close the system menu to start placing the patch points. The patch points can only be added when menu is not active in the scene.
6. Start adding the patch points along the periodic orbit of desired shape with the right controller trigger button. The patch points appear in the scene in white color. Refer to Figure A.13 for a snapshot of this action.
7. After a slight delay, the CR3BP look-ahead trajectory emanating from the patch points should appear in color white. Refer to Figure A.13 for a snapshot of this action.

### Identification of Periodic Orbits in Technique-III: A Screenplay continued...

8. Now, manipulate the velocity direction vector with the right controller trigger button to align the CR3BP look-ahead path with the two initial patch points. The velocity direction vector has all 3 degrees of freedom. However, the velocity magnitude is fixed for the two patch points based on the energy level chosen and a forward finite difference method used to compute velocity magnitude. Refer to Figure A.14 for a snapshot of this action.
9. The patch points are considered good if the CR3BP path can be aligned with the patch points using velocity direction vector manipulation. If a patch point is not good, erase the patch point and continue to adding a new patch point. If the current patch point is good, capture the patch point into the initial guess for the intended periodic orbit. Refer to Figure A.16 for a depiction of this action in VR.
10. Next, verify if the whole initial guess is good or not. If the CR3BP path captured from all the patch points is resembling the intended orbit shape, then the whole initial guess is considered good. If it is not, reset the scene and start from the beginning with step 3.
11. If the whole initial guess is good, next verify if the initial guess construction is complete, if not continue to adding the next patch point and repeat the steps 8 through 10. Refer to Figure A.15 for a snapshot of this action.
12. Upon completion of the construction of the initial guess using patch points, select the desired optimizer from the drop-down menu of options for the optimizer in the system menu. It is okay to experiment with all the options for a single initial guess using the patch points, but each option can only be used one at a time. Refer to Figure A.5 for a snapshot of this action.
13. Next, run the optimizer from the system menu button dedicated to it. Refer to Figure A.6 for a snapshot of this action.

### Identification of Periodic Orbits in Technique-III: A Screenplay continued...

14. Now, wait for the optimizer to finish and the solution orbit to appear in the scene in color magenta. Refer to Figure A.7 for a snapshot of this action.
15. When the solution periodic orbit appears in the scene, review it for the quality of the solution with respect to the initial-guess, which also simultaneously appears in the scene. The quality of the solution can also be assessed based on the solution parameters that appear along with the scene in text form. Of particular importance is the constraint violation parameter. Typically, a value of at or below  $1e - 6$  is considered to be good. Refer to Figure A.8 for a snapshot of this action.
16. If the solution orbit is satisfactory, then archive the solution using the system menu option for it. Refer to Figure A.9 for a snapshot of this action.
17. If the solution is not satisfactory, reset the scene and start from step 3 again.

#### 4.2.4 Technique-IV: Initial-Guess from the Simulation of Pure CR3BP Dynamical Motion

In this technique, pure CR3BP dynamical motion is leveraged to construct initial-guesses. In the VR scene for this technique, the human user manipulates the initial position and velocity (magnitude and direction) as desired to get an approximate shape for the desired periodic orbit. Once an approximate shape is achieved in the CR3BP dynamics that resembles the desired periodic orbit, the human user can then proceed to choose the patch points along the CR3BP path and use an optimizer to correct for the exact desired orbit from the initial-guess. Figure 4.8 is showing the VR user interaction menu for Technique-IV.

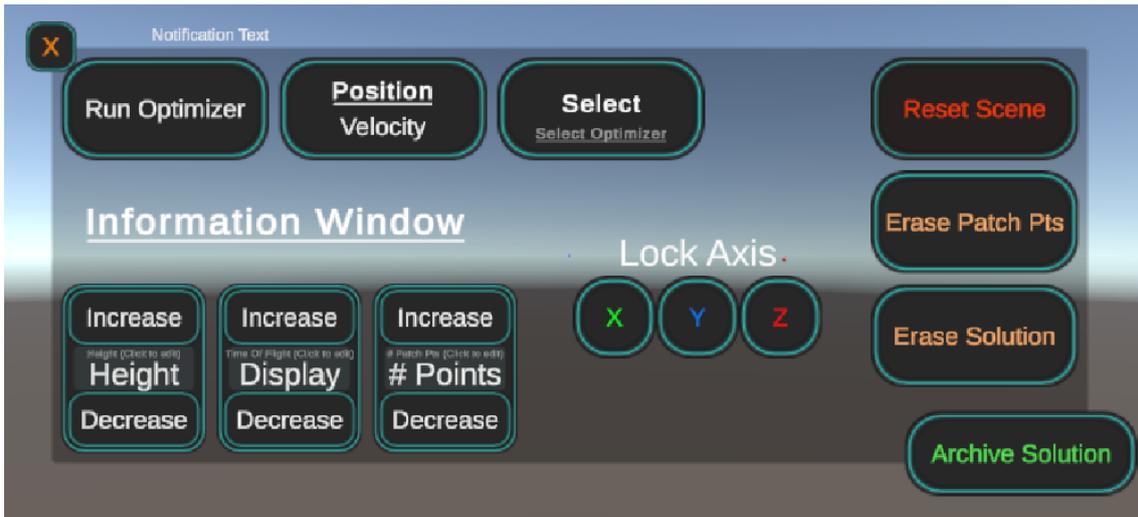


Figure 4.8: VR User-Interaction Menu for Technique-IV

In the VR menu there are actions that a human user can take by pointing the laser pointer from the right controller to the appropriate menu item and pressing trigger, to generate the initial-guess in Technique-IV. Figure 4.9 is illustrating the flow-chart for this technique. A screenplay for the technique follows immediately after.

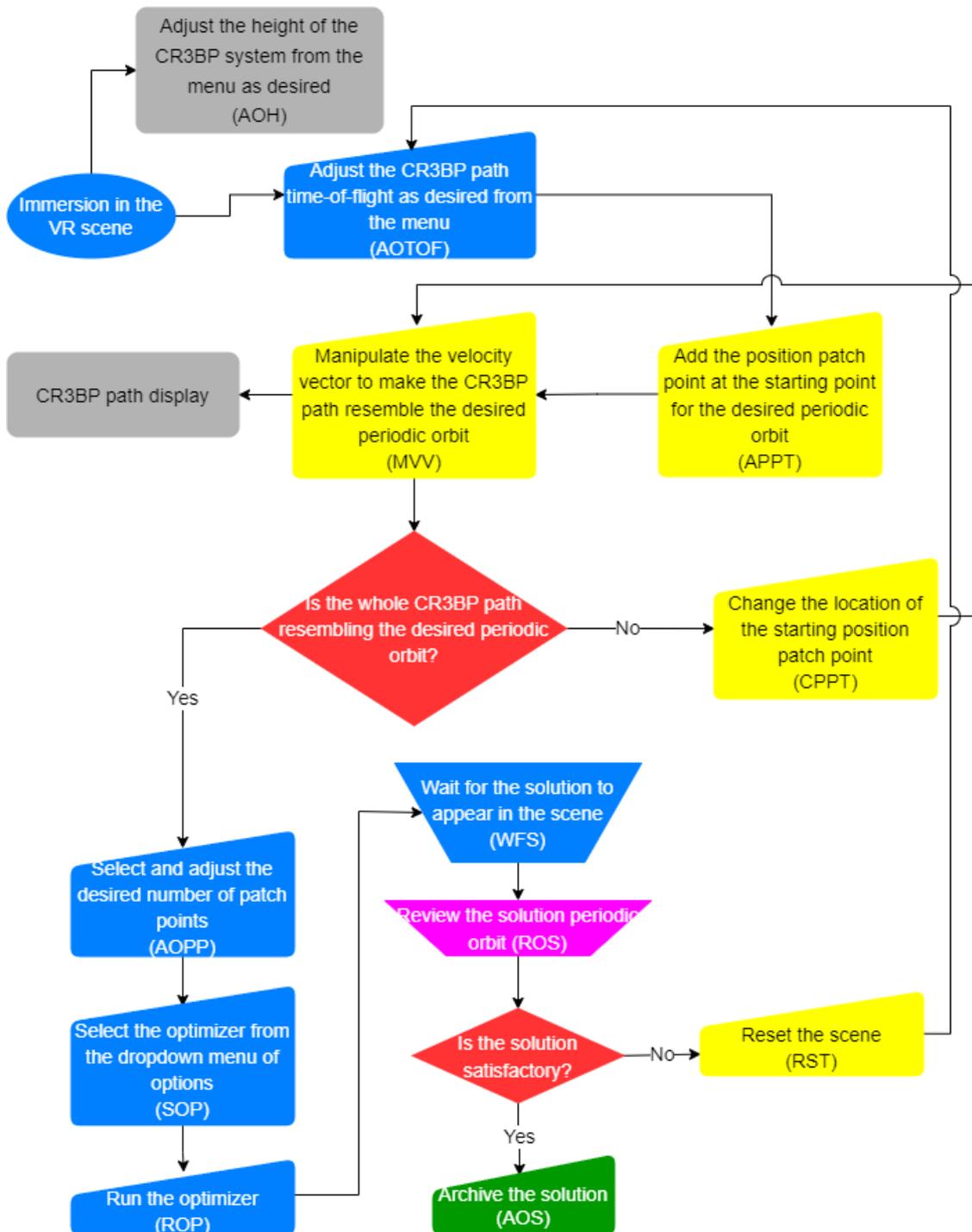


Figure 4.9: A Flow-Chart of Periodic Orbit Identification in Technique-IV

## Identification of Periodic Orbits in Technique-IV: A Screenplay

The screenplay for the VR scene for periodic identification in Technique-IV is,

1. Mount the VR headset and enter the VR scene with the right controller turned on. Human user sees the scene in Figure 4.1.
2. Adjust the height of the system as desired from the system menu. The system menu can be accessed from the menu button of the right controller. Refer to Figure A.1 for a snapshot of this action.
3. Adjust the CR3BP look-ahead time-of-flight as desired from the system menu. This action is not fixed per orbit. The value can be adjusted throughout the initial-guess construction process. Refer to Figure A.3 for a snapshot of this action.
4. Close the system menu to start manipulating initial position and velocity. The manipulation can only be done when menu is not active in the scene.
5. Start adding the initial position point with the right controller trigger button. The initial position appears in the scene in white color. Refer to Figure A.17 for a snapshot of this action.
6. After a slight delay, the CR3BP look-ahead trajectory emanating from the initial position and a random velocity appears in color white. Refer to Figure A.17 for a snapshot of this action.
7. Now, manipulate the velocity vector with the right controller trigger button to achieve the desired CR3BP path with the initial position point. The velocity vector has all 3 degrees of freedom for direction and length of the vector represents the magnitude. Refer to Figure A.18 for a depiction of this action in VR.

### Identification of Periodic Orbits in Technique-IV: A Screenplay continued...

8. Next, verify if the whole initial guess is good or not. If the CR3BP path captured from the initial position point and velocity is resembling the intended orbit shape, then the whole initial guess is considered good. If it is not, change the initial position point location and repeat step 7. If yes, proceed to next step.
9. Upon completion of the construction of the initial guess, select and adjust the desired number of patch points along the initial-guess using the system menu. The number of patch points should capture the curvature of the initial-guess properly to find a good solution orbit. Refer to Figure A.4 for a snapshot of this action.
10. Next, select the desired optimizer from the drop-down menu of options for the optimizer in the system menu. It is okay to experiment with all the options for a single initial guess, but each option can only be used one at a time. Refer to Figure A.5 for a snapshot of this action.
11. Next, run the optimizer from the system menu button dedicated to it. Refer to Figure A.6 for a snapshot of this action.
12. Now, wait for the optimizer to finish and the solution orbit to appear in the scene in color magenta. Refer to Figure A.7 for a snapshot of this action.
13. When the solution periodic orbit appears in the scene, review it for the quality of the solution with respect to the initial-guess, which also simultaneously appears in the scene. The quality of the solution can also be assessed based on the solution parameters that appear along with the scene in text form. Of particular importance is the constraint violation parameter. Typically, a value of at or below  $1e - 6$  is considered to be good. Refer to Figure A.8 for a snapshot of this action.
14. If the solution orbit is satisfactory, then archive the solution using the system menu option for it. Refer to Figure A.9 for a snapshot of this action.
15. If the solution is not satisfactory, reset the scene and start from step 3 again.

### 4.3 The Numerical Algorithms for Periodic Orbit Identification

There are two types of numerical optimization algorithms employed for periodic orbit identification. They are the multiple-shooting method (Section 3.2.3) and the generalized collocation method (Section 3.3.1). Here, I present the initial-guess construction methods and the cost function for periodic orbit identification VR schemes.

#### 4.3.1 The Cost Function for Periodic Orbit Identification

The selected cost function for the optimization is the average Euclidean distance of all patch-points along the discrete candidate solution trajectory to the closest points along the cost reference. In case of Technique-I, the cost reference is a collection of patch-points along the human drawing. In case of all the other techniques, the cost reference is the collection of initial set of CR3BP patch-points from the initial-guess. Formally, the cost function is defined as,

$$J = \frac{1}{n} \sum_{i=1}^n \sqrt{(x_i - x_{\text{ref}_i})^2 + (y_i - y_{\text{ref}_i})^2 + (z_i - z_{\text{ref}_i})^2} \quad (4.1)$$

In Equation (4.1),  $x_{\text{ref}}$ ,  $y_{\text{ref}}$  and  $z_{\text{ref}}$  are the components of closest position points along the cost reference.  $x_i$ ,  $y_i$ ,  $z_i$ , are the components of position points along the candidate optimal trajectory as calculated in the optimizer. Here,  $n$  is total number of patch-points along the candidate optimal trajectory. The goal of the optimizer is to minimize  $J$  subject to the dynamics specified in Equations (2.36). It is important to note that the cost function does not include any conditions on velocity and time-of-flight because these are not directly specified by the human user.

#### 4.3.2 Initial-Guesses for Technique-I

In this technique the variable-time collocation optimizer and multiple-shooting correctors are employed. The collocation optimizer takes  $\frac{(N+1)}{2}$  variable nodes per each arc in the input vector for the NLP solver, where  $N$  is the degree of the orthogonal polynomials employed. In this

dissertation, seventh degree Legendre-Gauss polynomials are employed. From the VR environment, the application only records positional data. A method is needed to generate a good initial guess for velocity at the variable nodes and the time-of-flight of each arc. Along with the positional data, having velocity and time-of-flight produces the complete set of information to initiate the drawing-to-orbit mapping process. Given that the trajectory energy measure for the CR3BP, called the Jacobi constant, complementing the position data-set, we employ finite-differences method to fill in the missing velocity and time-of-flight data in the initial guess. There are  $n - 1$  trajectory arcs chosen from the user-drawn trajectory for position data, where  $n$  is the number of patch points given as input in the VR menu. At arc patch-points  $s_i$  along the trajectory, a central difference scheme is used for computing the velocity direction ( $\frac{\vec{v}_i}{|\vec{v}_i|}$ ), and a forward difference scheme is used for computing the time-of-flight ( $\Delta t_i$ ). These are given by the Equations,

$$\frac{\vec{v}_i}{|\vec{v}_i|} = \frac{\vec{p}_{i+1} - \vec{p}_{i-1}}{|\vec{p}_{i+1} - \vec{p}_{i-1}|}, \text{ with } \vec{p} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (4.2)$$

$$|\vec{v}_i| = \sqrt{x_i^2 + y_i^2 + \frac{2(1-\mu)}{d_i} + \frac{2\mu}{r_i} - C} \quad (4.3)$$

$$\Delta t_i = \frac{|\vec{p}_{i+1} - \vec{p}_i|}{|\vec{v}_i|} \quad (4.4)$$

for all integer  $i$  values in  $[1, n - 1]$ . In these Equations, the subscript  $i$  denotes the identifier for the patch points. In Equation (4.3),  $\mu$  denotes the mass parameter,  $C$  denotes the Jacobi constant,  $d$  and  $r$  denote distances of the spacecraft from the larger and the smaller primaries respectively. The Jacobi constant is chosen by the human user in the VR menu. It's also worth noting that the central difference scheme is employed for the velocity direction guess, because a higher accuracy is needed in this case than for the time-of-flight where a forward difference scheme is sufficient. Once the initial guesses for velocity and time-of-flight are computed at the arc boundaries, the values of state at the variable nodes in each arc of the collocation input vector are computed with linear interpolation. The input vector for the variable time collocation

optimizer is,

$$\vec{X} = \begin{bmatrix} x_{1,1} \\ \cdot \\ \cdot \\ x_{1,\frac{N+1}{2}} \\ \cdot \\ \cdot \\ \cdot \\ x_{n,1} \\ \cdot \\ \cdot \\ \cdot \\ x_{n,\frac{N+1}{2}} \\ \cdot \\ \cdot \\ \cdot \\ \dot{z}_{n,\frac{N+1}{2}} \\ T_1 \\ \cdot \\ \cdot \\ T_{n-1} \end{bmatrix} \quad (4.5)$$

The input vector for the multiple-shooting variable time corrector is,

$$\vec{X} = \begin{bmatrix} \vec{s}_1 \\ \vec{s}_2 \\ \cdot \\ \cdot \\ \cdot \\ \vec{s}_{n-1} \\ \vec{s}_n \\ T_1 \\ \cdot \\ \cdot \\ T_{n-1} \end{bmatrix} \quad (4.6)$$

### 4.3.3 Initial-Guesses for Technique-II

For this technique, variable time collocation, fixed and variable time multiple-shooting algorithms are employed. The collocation optimizer takes  $\frac{(N+1)}{2}$  variable nodes per each arc in the input vector for the NLP solver, where  $N$  is the degree of the orthogonal polynomials employed. In this dissertation, the Legendre-Gauss polynomials are employed. The initial-guess input vector is constructed by dividing the entire CR3BP trajectory input into  $n$  patch points first. Second, the in each of the trajectory arcs outlined by the patch points, a free CR3BP trajectory is propagated to obtain the variable nodes in the input per each arc. The input vector is of the form given in Equation (4.5) for variable time collocation optimizer. The input vector for multiple-shooting fixed time corrector is,

$$\vec{X} = \begin{bmatrix} \vec{s}_1 \\ \vec{s}_2 \\ \cdot \\ \cdot \\ \cdot \\ \vec{s}_{n-1} \\ \vec{s}_n \\ T \end{bmatrix} \quad (4.7)$$

and is of the form for variable time multiple-shooting corrector,

$$\vec{X} = \begin{bmatrix} \vec{s}_1 \\ \vec{s}_2 \\ \cdot \\ \cdot \\ \cdot \\ \vec{s}_{n-1} \\ \vec{s}_n \\ T_1 \\ \cdot \\ \cdot \\ T_{n-1} \end{bmatrix} \quad (4.8)$$

#### 4.3.4 Initial-Guesses for Technique-III

For technique-III, variable-time collocation optimizer and fixed-time and variable-time multiple-shooting corrector algorithms are employed for identifying periodic orbits from the patch-point based initial guesses. The input vectors for all these three algorithms are as given in Equations (4.5), (4.7) and (4.8) respectively. The input vectors are constructed directly from the CR3BP state emanating from the patch-points given in the initial-guess in this technique from the VR platform.

#### 4.3.5 Initial-Guesses for Technique-IV

For technique-IV, fixed-time and variable-time multiple-shooting corrector algorithms are employed for identifying periodic orbits from the pure CR3BP state based initial guesses. The input vectors for the two algorithms are as given in Equations (4.7) and (4.8) respectively. The input vectors are constructed directly from the CR3BP state emanating from the initial position and velocity given in this technique from the VR platform.

#### 4.3.6 Implementation Details

The fixed time and variable time multiple-shooting and the variable time collocation algorithms are implemented in MATLAB, leveraging the *fmincon* tool with the *IPOPT* NLP solver. For multiple-shooting implementations, the MATLAB version is supplied with analytical derivatives for the Jacobian matrix. The variable time multiple-shooting and collocation algorithms are also implemented in Python, leveraging the *CasADi* [31] package. However, the Jacobian matrix is constructed automatically by the *CasADi* package using the automatic differentiation feature.

### 4.4 System Architectures for Periodic Orbit Identification Techniques

Figures 4.10 and 4.11 are showing the system architectures for the periodic orbit identification techniques. Note that the four periodic orbit identification techniques each have a single VR application that implements both these architectures together.

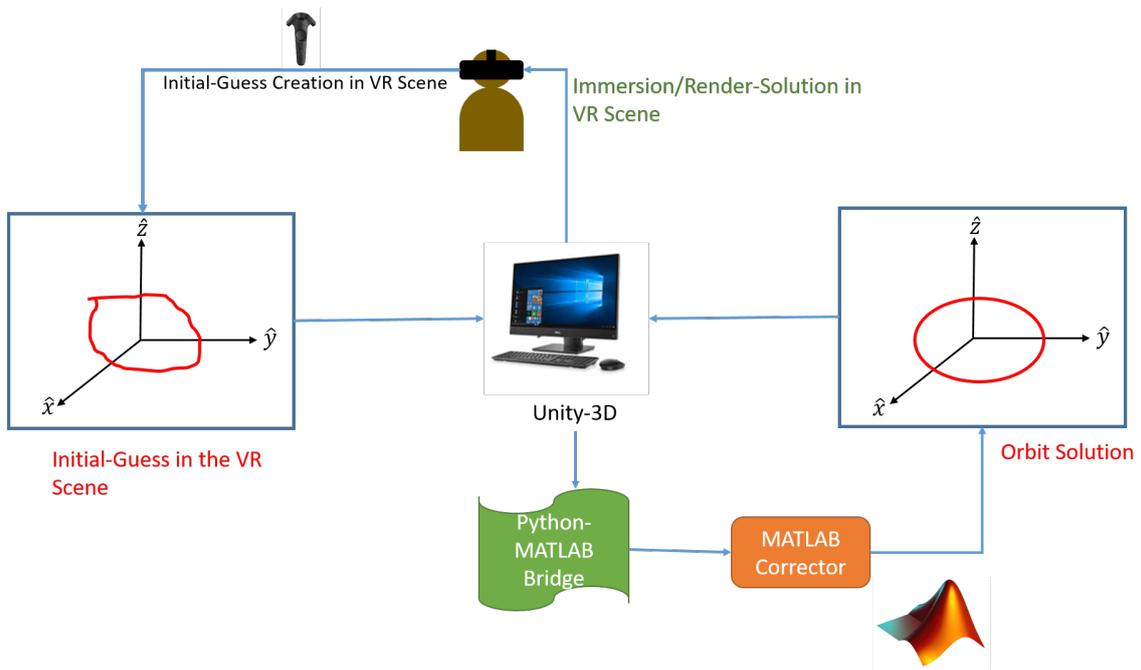


Figure 4.10: Periodic Orbit Identification System Architecture with MATLAB Support

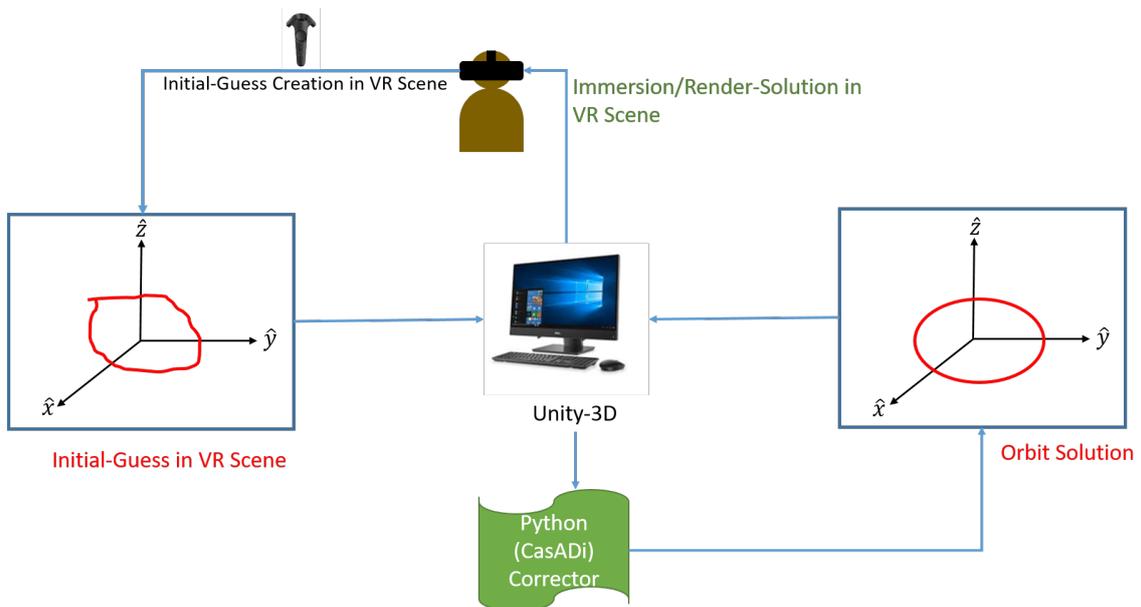


Figure 4.11: Periodic Orbit Identification System Architecture with Python (CasADi) Support

#### 4.5 Transfer Trajectory Identification Technique

For transfer trajectory identification, the orbit-chain method as described by Pritchett [19] is employed. In the orbit-chain method, there are six steps for composition of a good initial guess to transfer problem between two orbits. The six steps of this method are,

1. **Analyze:** In this step, the energy versus time-of-flight plots for various families of multi-body orbits are analyzed to identify intermediate orbits that gradually increase/decrease the energy of the spacecraft path from the source to the target orbit.
2. **Select:** In this step, the intermediate orbits are selected from the energy plots to stitch together to form an initial guess that can take the spacecraft from source energy level to the target energy level.
3. **Clip:** In this step, the intermediate orbits are clipped into parts that can together form a reasonable initial path for the orbital transfer.
4. **Subdivide:** In this step, the parts of the intermediate orbits are divided into a reasonable number of arcs for the direct collocation algorithm that processes the initial guess.
5. **Stack:** In this step, the initial guess input vector to the collocation optimizer is composed by stacking the arcs of the individual parts of the transfer trajectory initial guess from previous step.
6. **Define Control History:** In this final step, a reasonable guess for control history along the initial guess is made to finish the problem formulation for the collocation optimizer.

The first three steps from above list are implemented in the VR scene for the transfer trajectory identification. A human operator can inspect the energy versus time-of-flight plots of various families of cislunar CR3BP system and compose a reasonable initial guess for a transfer. While the operator is exploring the plots, the corresponding orbits from selections in the plots are also shown to the operator. This allows for the operator to interactively choose intermediate orbits that make sense in the physical space of the cislunar CR3BP system. The operator can then clip the chosen intermediate orbits at the appropriate positions in the simulated cislunar CR3BP system. Steps 4 through 6 are conducted offline in the attached collocation optimizer. The human operator can trigger the collocation optimizer from the VR controller. The result of the collocation optimizer is eventually displayed to the human operator in the VR scene. A flow-chart for a representative flow of actions in transfer trajectory identification VR scene are shown in Figure 4.12. A screenplay for this flow-chart is presented after this Figure.

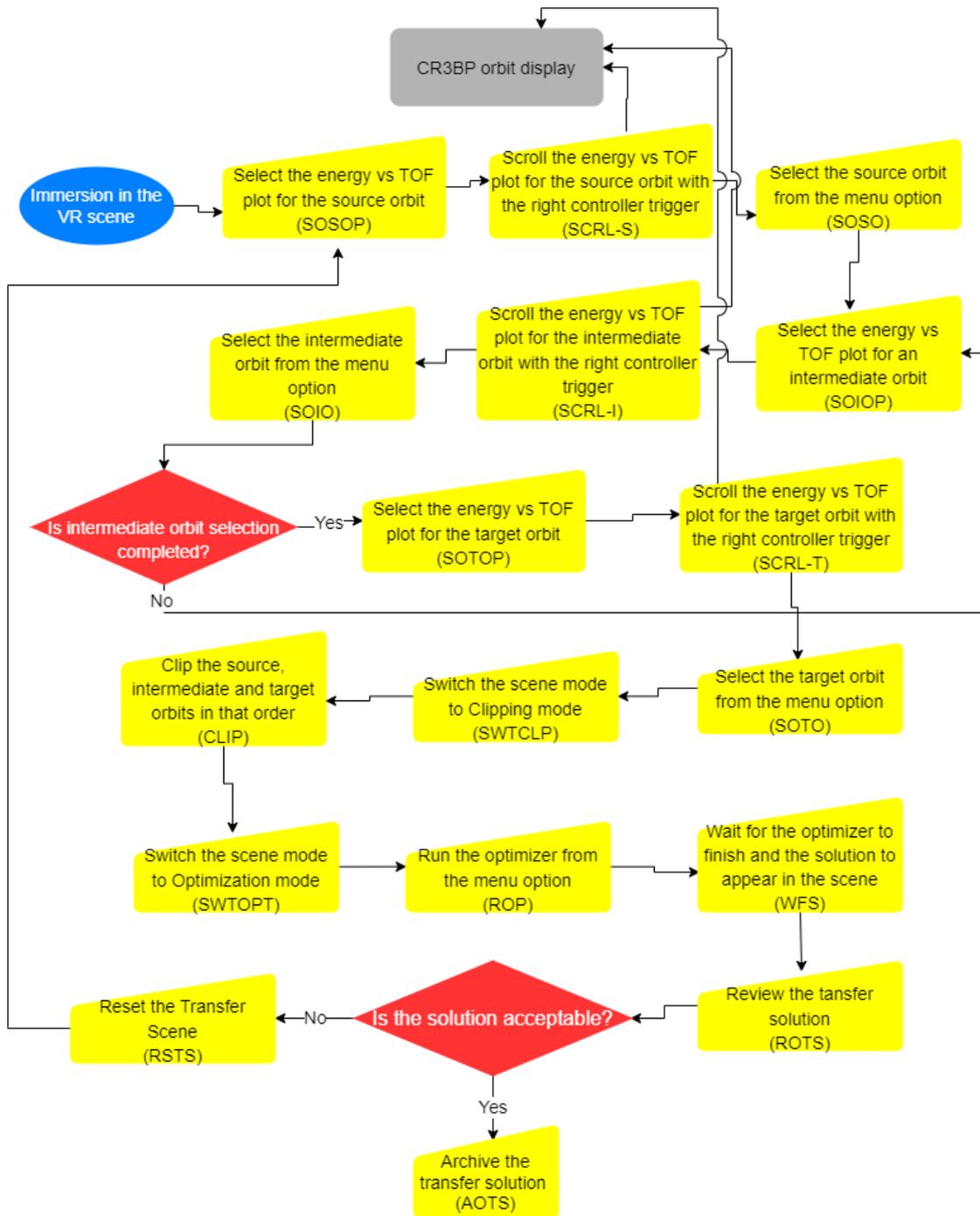


Figure 4.12: A Flow-Chart of Transfer Trajectory Identification in VR

## Identification of Low-Thrust CR3BP Transfer Trajectories: A Screenplay

The screenplay for the VR scene for periodic identification from pure CR3BP dynamical motion is,

1. Mount the VR headset and enter the VR scene with the right controller turned on. Human user sees the scene in Figure A.19.
2. Adjust the height of the system as desired from the system menu. The system menu can be accessed from the menu button of the right controller. Refer to Figure A.1 for a snapshot of this action.
3. From the list of energy-versus-time-of-flight plots, select the plot for the source orbit family, with the right controller trigger button. Refer to Figure A.20 for a depiction of this action in VR.
4. In this plot, scroll through the energy points and review the corresponding CR3BP orbits displayed in the scene. Refer to Figure A.21 for a depiction of this action in VR.
5. Select a desired source orbit from the menu option designated for orbit selection. Refer to Figure A.22 for a depiction of this action in VR.
6. Go back to the list of energy-versus-time-of-flight plots and then select the plot for the desired intermediate orbit family. Refer to Figure A.23 for a depiction of this action in VR.
7. In this intermediate family energy plot, scroll through the energy points and review the corresponding CR3BP orbits displayed in the scene. Refer to Figure A.24 for a depiction of this action in VR.
8. Select a desired intermediate orbit from the menu option designated for orbit selection. Make sure to select an orbit with either higher or lower energy depending on whether the source orbit energy is lower or higher than the intended target orbit energy level. Refer to Figure A.25 for a depiction of this action in VR.

## Identification of Low-Thrust CR3BP Transfer Trajectories: A Screenplay continued...

9. Verify whether all necessary intermediate orbits are selected. If not, repeat steps 6 through 8 as many times as necessary.
10. If all intermediate orbits are selected, proceed to selecting the energy-versus-time-of-flight plot for the family of orbits for the target orbit. Refer to Figure A.26 for a depiction of this action in VR.
11. In this plot, scroll through the energy points and review the corresponding CR3BP orbits displayed in the scene. Refer to Figure A.27 for a depiction of this action in VR.
12. Select a desired target orbit from the menu option designated for orbit selection. Refer to Figure A.28 for a depiction of this action in VR.
13. Now, switch the scene mode to clipping mode from the menu. In this mode, step 2 from the orbit-chain method is realized. In this mode, human operator selects the parts of the source, intermediate and target orbits that are part of the initial-guess for the transfer. To do so, the human operator clips each orbit with starting and ending points of the part that becomes part of the initial-guess. Refer to Figures A.29 and A.30 for a depiction of these actions in VR.
14. Upon completion of clipping step, switch the scene mode to optimization mode from the menu. Refer to Figure A.31 for a depiction of this action in VR.
15. Now, run the optimizer from the menu option designated for this action. Refer to Figure A.32 for a depiction of this action in VR.

## Identification of Low-Thrust CR3BP Transfer Trajectories: A Screenplay continued...

16. Wait for the final solution to appear in the scene. While the optimizer is running, the intermediate solutions are displayed to the human operator in the scene, so the progression of the solution towards the final solution can be seen. Refer to Figure A.7 for a depiction of this action in VR.
17. Upon completion of optimization and display of the final solution, review the transfer solution. To facilitate the review, a solution parameter panel is displayed to the human operator, along with the transfer trajectory path shown to the operator in the scene. Refer to Figure A.34 for a depiction of this action in VR.
18. If the solution is acceptable, archive the solution using the menu option and quit. Refer to Figure A.35 for a depiction of this action in VR.
19. If the solution is not acceptable, reset the scene from the menu option and start from step 3 again.

### 4.6 The Numerical Algorithm for Transfer Trajectory Identification

The generalized collocation algorithm described in Section 3.3.1 is able to treat inputs and constraints for periodic orbit identification only. To reuse the generalized collocation algorithm for the transfer trajectory identification (Collocation Optimization for Low-Thrust [COLT] as defined by Pritchett [19]), control variables are introduced in the input vector to this algorithm and the constraint formulation is modified to use the control variables from the input vector to

enforce control based constraints of the problem. The new input vector is given as,

$$\vec{X} = \begin{bmatrix} x_{1,1} \\ \cdot \\ \cdot \\ x_{1,\frac{N+1}{2}} \\ \cdot \\ \cdot \\ x_{n,1} \\ \cdot \\ \cdot \\ x_{n,\frac{N+1}{2}} \\ \cdot \\ \cdot \\ \dot{z}_{n,\frac{N+1}{2}} \\ m_{I,1,1} \\ \cdot \\ m_{I,n,\frac{N+1}{2}} \\ m_{F,1,1} \\ \cdot \\ m_{F,1,\frac{N+1}{2}} \\ u_{1,1,1} \\ \cdot \\ u_{1,n,\frac{N+1}{2}} \\ u_{2,1,1} \\ \cdot \\ u_{2,n,\frac{N+1}{2}} \\ u_{3,1,1} \\ \cdot \\ u_{3,n,\frac{N+1}{2}} \\ T_1 \\ \cdot \\ \cdot \\ T_{n-1} \end{bmatrix} \quad (4.9)$$

In Equation (4.9), variables  $m_I$  and  $m_F$  represent initial and final masses in each segment of the transfer trajectory in kilograms. Variables  $u_1$ ,  $u_2$  and  $u_3$  represent the  $x$ ,  $y$  and  $z$  components of the thrust vector  $u_i$  in each of the  $n-1$  transfer trajectory segments. There is a thrust component of acceleration enforced in the system dynamics as follows,

$$\begin{aligned}\ddot{x} &= 2n\dot{y} + n^2x - \frac{(1-\mu)(x+\mu)}{d^3} - \frac{\mu(x-1+\mu)}{r^3} + \frac{T}{m}u_1 \\ \ddot{y} &= -2n\dot{x} + n^2y - \frac{(1-\mu)y}{d^3} - \frac{\mu y}{r^3} + \frac{T}{m}u_2 \\ \ddot{z} &= -\frac{(1-\mu)z}{d^3} - \frac{\mu z}{r^3} + \frac{T}{m}u_3\end{aligned}\quad (4.10)$$

In addition the following constraints are enforced on mass consumption and thrust vector in each transfer trajectory segment,

$$\begin{aligned}\sigma_i &= \frac{m_{I,i} - m_{F,i}}{\dot{m}_{max}\Delta t_i} \\ 0 &\leq \sigma_i \leq 1 \\ \|u_i\| &= \sqrt{u_{1,i}^2 + u_{2,i}^2 + u_{3,i}^2} = 1\end{aligned}\quad (4.11)$$

In this dissertation the low-thrust transfer problem is solved with constant thrust engines with constant maximum mass flow rate,  $m_{max}$ . Constant thrust value used is 200 millinewtons.  $m_{max}$  computed assuming an engine Isp of 2000 seconds. The initial mass of the spacecraft is always assumed to be 1000 kilograms.

#### 4.7 System Architecture for the Transfer Trajectory Identification Technique

Figure 4.13 is showing the system architecture for the VR application built to implement the transfer trajectory identification technique. The MATLAB optimizer shown in this figure utilizes a Jacobian constructed using automatic differentiation feature provided by ADiGator [32] tool.

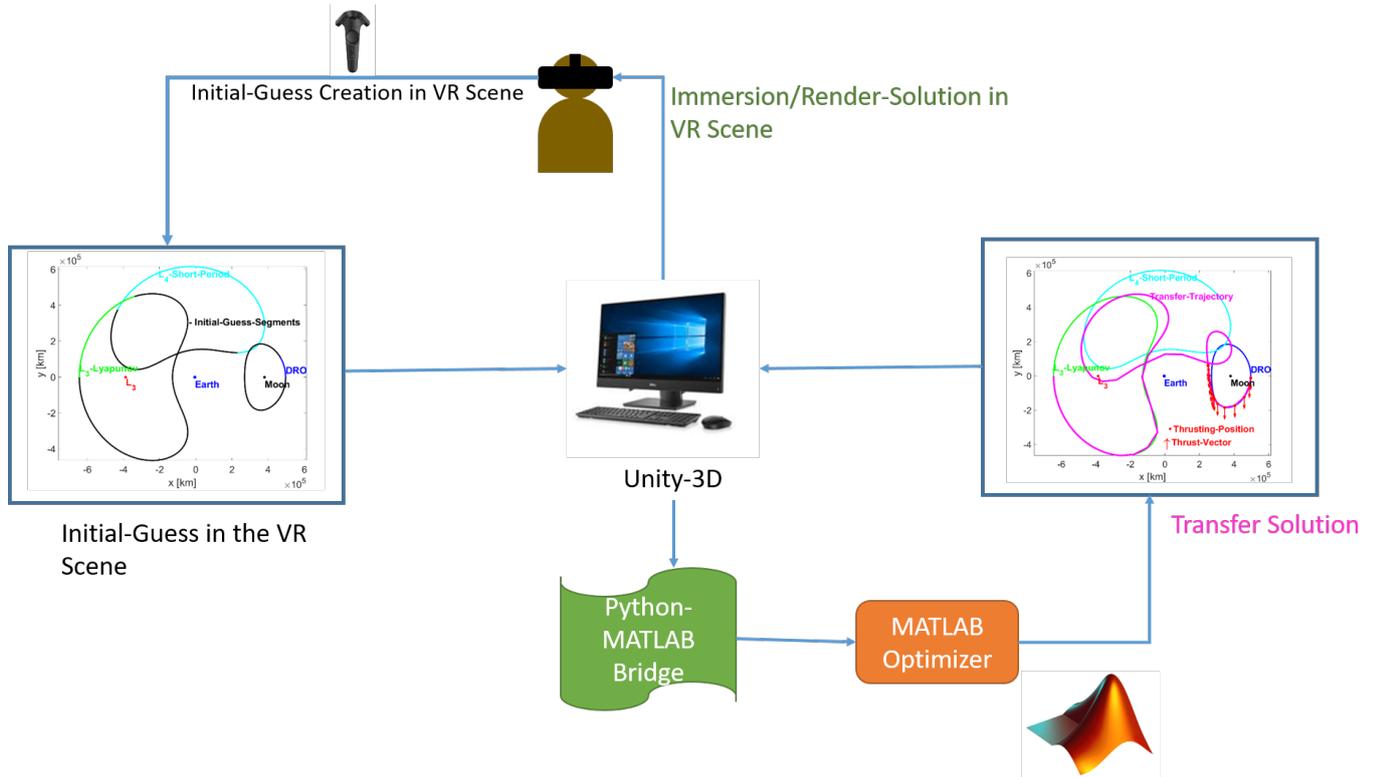


Figure 4.13: Transfer Trajectory Identification System Architecture with MATLAB Support

## Chapter 5

### Periodic Orbit Identification on the Virtual Reality Platform

This chapter presents the verification results for four periodic orbit identification techniques from Chapter 4. For the verification of the techniques, four types of  $L_1$  Lagrange point orbits, {Lyapunov, halo, axial, and vertical}, are identified. This is done so to establish a baseline for the validity of the periodic orbit identification techniques developed in VR. Note that even though results are only presented for  $L_1$  orbits, the techniques are capable of recovering any type of periodic orbit, if one exists. After all the results for the verification are presented, a comparison of the results from all four techniques is also presented.

#### 5.1 Technique-I Periodic Orbits

In the verification experiments, an energy level of 3 is used for all four orbit types, for the initial-guess generation for the corrector algorithms using the method described in Section 4.3.2. Figure 5.1 is showing a set of four of the common types of Lagrange point orbits identified using this technique. Tables 5.1 through 5.4 are showing the solution parameters for the orbit solutions in Figure 5.1.

Table 5.1: Solution Parameters for an  $L_1$  Lyapunov Solution from Technique-I

<b>Parameter</b>	<b>Value</b>
Arcs	11
Time-of-Flight [NTU]	3.9530031092839035
Jacobi-Constant [NDIM]	3.0205700269380142
Constraint-Violation [NDIM]	$2.1083972943537736e - 14$
Corrector-Type	Variable-Time-Collocation
Corrector-Platform	Python (CasADi)
NLP-Solver	IPOPT
Drawing-Energy [NDIM]	3
Execution-Time [Seconds]	2.1621222496032715

Table 5.2: Solution Parameters for an  $L_1$  Halo Solution from Technique-I

<b>Parameter</b>	<b>Value</b>
Arcs	10
Time-of-Flight [NTU]	2.78069602990958
Jacobi-Constant [NDIM]	3.0678526605049323
Constraint-Violation [NDIM]	$4.542315695888056e - 06$
Corrector-Type	Variable-Time-Multiple-Shooting
Corrector-Platform	Python (CasADi)
NLP-Solver	IPOPT
Drawing-Energy [NDIM]	3
Execution-Time [Seconds]	29.089023113250732

Table 5.3: Solution Parameters for an  $L_1$  Axial Solution from Technique-I

Parameter	Value
Arcs	12
Time-of-Flight [NTU]	3.9684431832861664
Jacobi-Constant [NDIM]	3.0163973684023855
Constraint-Violation [NDIM]	$8.964209003599562e - 14$
Corrector-Type	Variable-Time-Collocation
Corrector-Platform	Python (CasADi)
NLP-Solver	IPOPT
Drawing-Energy [NDIM]	3
Execution-Time [Seconds]	2.667895793914795

Table 5.4: Solution Parameters for an  $L_1$  Vertical Solution from Technique-I

Parameter	Value
Arcs	8
Time-of-Flight [NTU]	3.3565818626122734
Jacobi-Constant [NDIM]	3.0704190440173043
Constraint-Violation [NDIM]	$5.868312963342263e - 06$
Corrector-Type	Variable-Time-Collocation
Corrector-Platform	Python (CasADi)
NLP-Solver	IPOPT
Drawing-Energy [NDIM]	3
Execution-Time [Seconds]	3.953247547149658

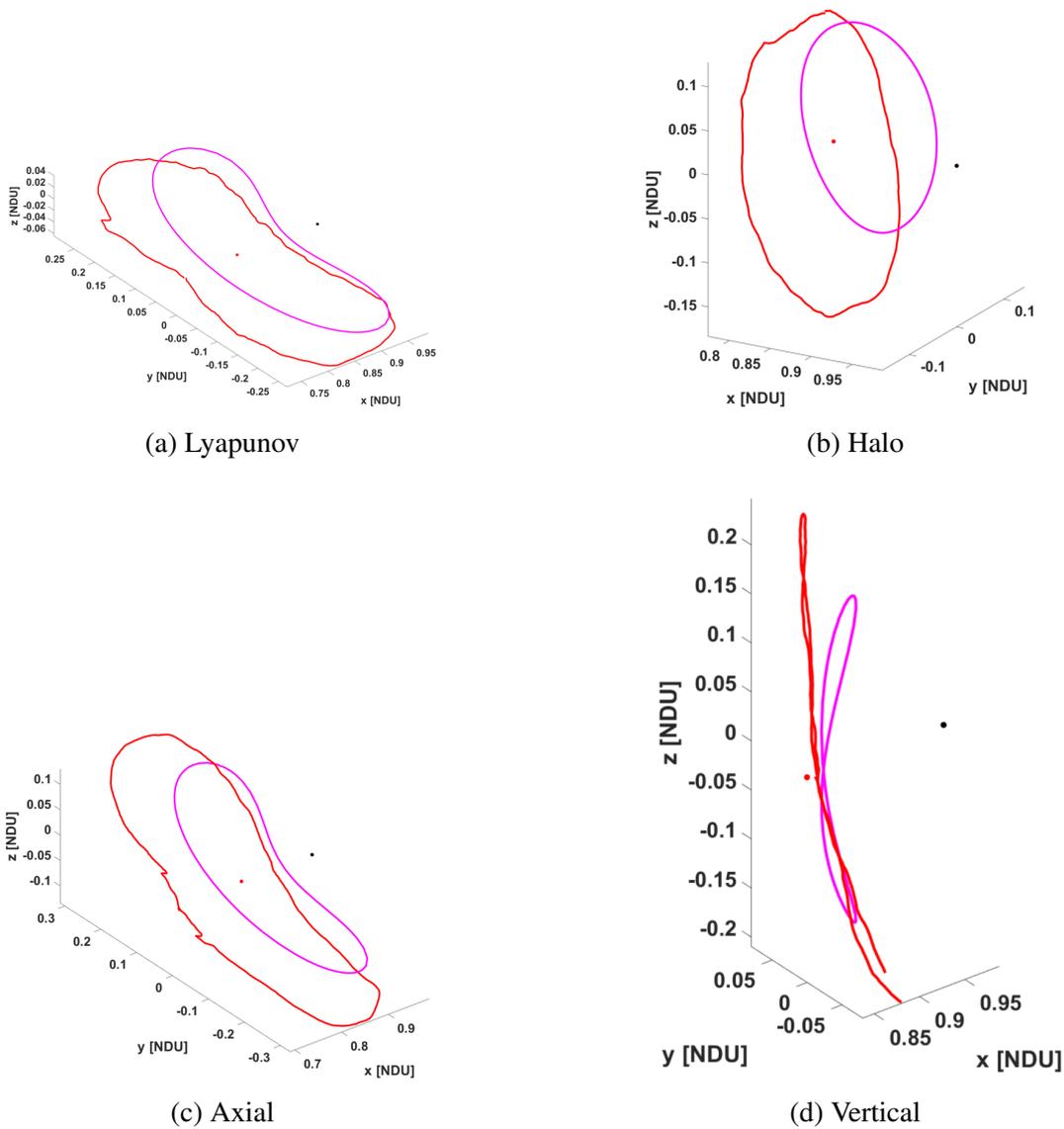


Figure 5.1: Sample Collection Periodic Orbits Identified in Technique-I

## 5.2 Technique-II Periodic Orbits

In the verification experiments, an energy level of 3 is used for all orbit type drawings. The initial-guess is constructed using the CR3BP state emanating from the drawing. In this technique, the drawing and the CR3BP-guess seem to overlap, but in actuality they do not, but are very close to each other in position space. However, the drawing does not have any valid velocity and epoch components associated with it, but the CR3BP-guess does, making it easier for the optimizer to find solutions.

Figure 5.2 is showing a set of four of the common types of Lagrange point orbits identified using this technique. Tables 5.5 through 5.8 are showing the solution parameters for the orbit solutions in Figure 5.2.

Table 5.5: Solution Parameters for an  $L_1$  Lyapunov Solution from Technique-II

Parameter	Value
Arcs	12
Time-of-Flight [NTU]	4.568494857060037
Jacobi-Constant [NDIM]	2.9891794430643097
Constraint-Violation [NDIM]	$2.9406406316202887e - 13$
Corrector-Type	Variable-Time-Collocation
Corrector-Platform	Python (CasADi)
NLP-Solver	IPOPT
Drawing-Energy [NDIM]	3
Execution-Time [Seconds]	3.0364768505096436

Table 5.6: Solution Parameters for an  $L_1$  Halo Solution from Technique-II

Parameter	Value
Arcs	8
Time-of-Flight [NTU]	2.751940538195498
Jacobi-Constant [NDIM]	3.045357990174661
Constraint-Violation [NDIM]	$8.139646118042634e - 07$
Corrector-Type	Variable-Time-Multiple-Shooting
Corrector-Platform	Python (CasADi)
NLP-Solver	IPOPT
Drawing-Energy [NDIM]	3
Execution-Time [Seconds]	28.4535710811615

Table 5.7: Solution Parameters for an  $L_1$  Axial Solution from Technique-II

Parameter	Value
Arcs	12
Time-of-Flight [NTU]	3.9862388626505454
Jacobi-Constant [NDIM]	3.0109441531197811
Constraint-Violation [NDIM]	$6.0127116847896948E - 7$
Corrector-Type	Variable-Time-Collocation
Corrector-Platform	MATLAB (fmincon)
NLP-Solver	IPOPT
Drawing-Energy [NDIM]	3
Execution-Time [Seconds]	6.5581

Table 5.8: Solution Parameters for an  $L_1$  Vertical Solution from Technique-II

Parameter	Value
Arcs	8
Time-of-Flight [NTU]	3.177359636047557
Jacobi-Constant [NDIM]	3.0979839861887917
Constraint-Violation [NDIM]	$8.203740390221706e - 06$
Corrector-Type	Variable-Time-Multiple-Shooting
Corrector-Platform	Python (CasADi)
NLP-Solver	IPOPT
Drawing-Energy [NDIM]	3
Execution-Time [Seconds]	29.990614414215088

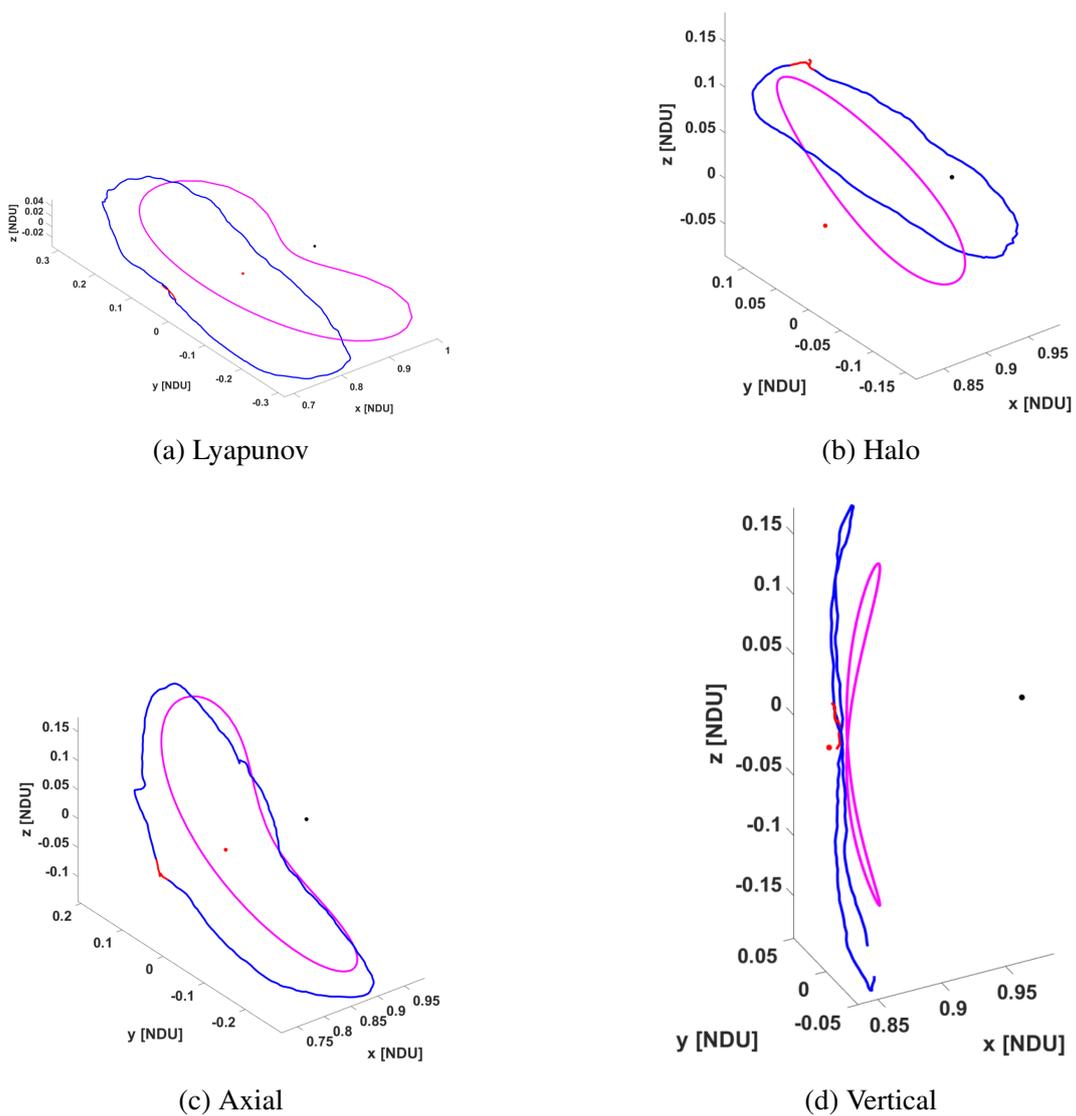


Figure 5.2: Sample Collection Periodic Orbits Identified in Technique-II

### 5.3 Technique-III Periodic Orbits

In the verification experiments for this technique, for Lyapunov and halo orbits an initial-guess energy of 3 is used. For axial and vertical orbits, an initial-guess energy of 2.8 is used, since these two types of orbits are hard to come by using this technique for initial-guess construction. Essentially, the CR3BP look-ahead trajectory turned out to be hard to produce matching an axial or vertical orbits.

Figure 5.3 is showing a set of four of the common types of Lagrange point orbits identified using this technique. Tables 5.9 through 5.12 are showing the solution parameters for the orbit solutions in Figure 5.3.

Table 5.9: Solution Parameters for an  $L_1$  Lyapunov Solution from Technique-III

Parameter	Value
Arcs	8
Time-of-Flight [NTU]	3.951551255222032
Jacobi-Constant [NDIM]	3.0209635460202797
Constraint-Violation [NDIM]	$1.647503211476864e - 05$
Corrector-Type	Variable-Time-Multiple-Shooting
Corrector-Platform	Python (CasADi)
NLP-Solver	IPOPT
Initial-Guess-Energy [NDIM]	3
Execution-Time [Seconds]	36.97302293777466

Table 5.10: Solution Parameters for an  $L_1$  Halo Solution from Technique-III

Parameter	Value
Arcs	6
Time-of-Flight [NTU]	2.7548273687696936
Jacobi-Constant [NDIM]	3.046827083821511
Constraint-Violation [NDIM]	0.00017639290069655105
Corrector-Type	Variable-Time-Collocation
Corrector-Platform	Python (CasADi)
NLP-Solver	IPOPT
Initial-Guess-Energy [NDIM]	3
Execution-Time [Seconds]	2.902064085006714

Table 5.11: Solution Parameters for an  $L_1$  Axial Solution from Technique-III

<b>Parameter</b>	<b>Value</b>
Arcs	8
Time-of-Flight [NTU]	3.983774820288912
Jacobi-Constant [NDIM]	3.012268589220998
Constraint-Violation [NDIM]	$2.5398364467679184e - 13$
Corrector-Type	Variable-Time-Collocation
Corrector-Platform	Python (CasADi)
NLP-Solver	IPOPT
Initial-Guess-Energy [NDIM]	2.8
Execution-Time [Seconds]	1.0503809452056885

Table 5.12: Solution Parameters for an  $L_1$  Vertical Solution from Technique-III

<b>Parameter</b>	<b>Value</b>
Arcs	8
Time-of-Flight [NTU]	3.6043309905671284
Jacobi-Constant [NDIM]	3.0389920834644744
Constraint-Violation [NDIM]	$1.4987946001563528e - 06$
Corrector-Type	Variable-Time-Collocation
Corrector-Platform	Python (CasADi)
NLP-Solver	IPOPT
Initial-Guess-Energy [NDIM]	2.8
Execution-Time [Seconds]	3.974919319152832

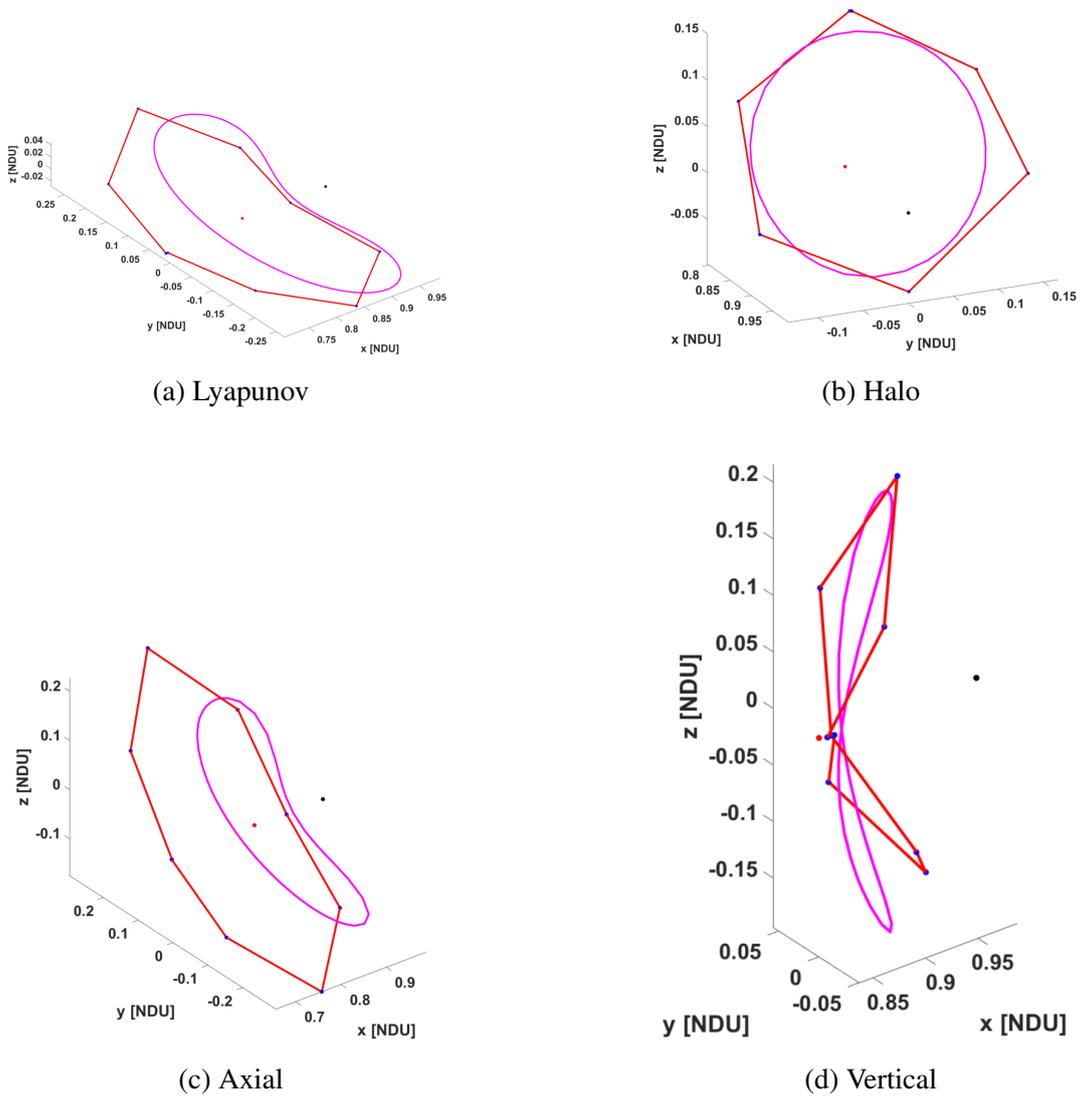


Figure 5.3: Sample Collection Periodic Orbits Identified in Technique-III

#### 5.4 Technique-IV Periodic Orbits

In this technique, the verification experiments do not need an input for the energy-level of the initial-guess being constructed. The energy-level is implied from the initial-position and velocity chosen for the initial-guess. Also, from these experiments, it became evident that providing a reasonable match to half of the typical orbit shape is enough to identify solutions in this method.

Figure 5.4 is showing a set of four of the common types of Lagrange point orbits identified using this technique. Tables 5.13 through 5.16 are showing the solution parameters for the orbit solutions in Figure 5.4.

Table 5.13: Solution Parameters for an  $L_1$  Lyapunov Solution from Technique-IV

<b>Parameter</b>	<b>Value</b>
Arcs	12
Time-of-Flight [NTU]	3.956193580122918
Jacobi-Constant [NDIM]	3.0198670702577557
Constraint-Violation [NDIM]	$1.781961565091783e - 13$
Corrector-Type	Variable-Time-Collocation
Corrector-Platform	Python (CasADi)
NLP-Solver	IPOPT
Execution-Time [Seconds]	0.5978765487670898

Table 5.14: Solution Parameters for an  $L_1$  Halo Solution from Technique-IV

<b>Parameter</b>	<b>Value</b>
Arcs	9
Time-of-Flight [NTU]	2.5802468264395753
Jacobi-Constant [NDIM]	3.010086672919797
Constraint-Violation [NDIM]	$5.659152423229619e - 06$
Corrector-Type	Variable-Time-Collocation
Corrector-Platform	Python (CasADi)
NLP-Solver	IPOPT
Execution-Time [Seconds]	4.3062803745269775

Table 5.15: Solution Parameters for an  $L_1$  Axial Solution from Technique-IV

Parameter	Value
Arcs	12
Time-of-Flight [NTU]	4.906360299206213
Jacobi-Constant [NDIM]	2.9454014561047486
Constraint-Violation [NDIM]	$1.1090985896426956e - 12$
Corrector-Type	Variable-Time-Collocation
Corrector-Platform	Python (CasADi)
NLP-Solver	IPOPT
Execution-Time [Seconds]	0.5252444744110107

Table 5.16: Solution Parameters for an  $L_1$  Vertical Solution from Technique-IV

Parameter	Value
Arcs	14
Time-of-Flight [NTU]	5.369606653563848
Jacobi-Constant [NDIM]	2.8406308069096884
Constraint-Violation [NDIM]	$1.8711825435906043e - 07$
Corrector-Type	Variable-Time-Collocation
Corrector-Platform	Python (CasADi)
NLP-Solver	IPOPT
Execution-Time [Seconds]	6.21152925491333

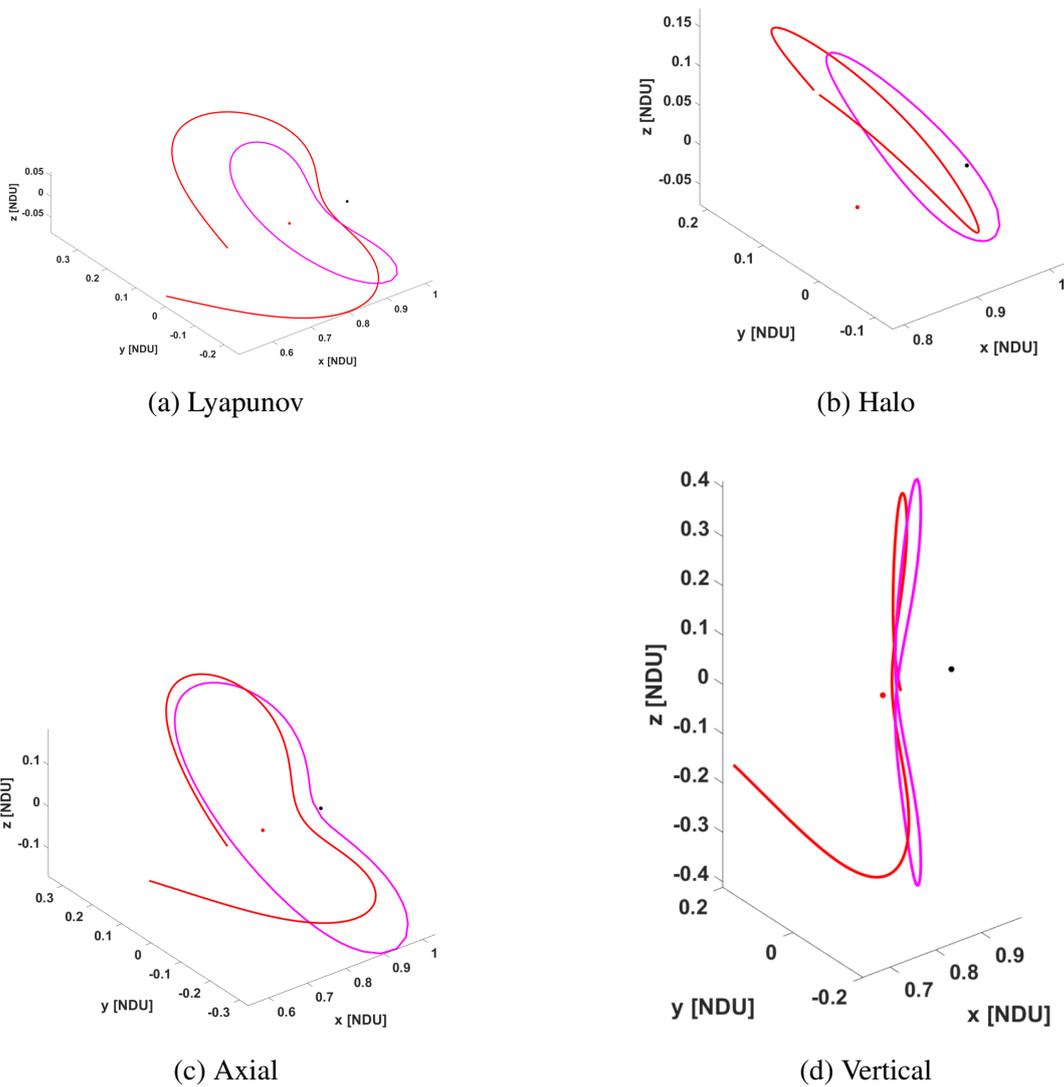


Figure 5.4: Sample Collection Periodic Orbits Identified in Technique-IV

## 5.5 Comparison of the Techniques

All four techniques are yielding periodic orbits from their respective initial-guess construction techniques. However there are some conditions under which one technique is better over the other. In technique-I, the user can freely draw a desired periodic orbit and then experiment with various different energy levels for the drawing to find a solution. However, some energy levels may not yield an initial guess from this method because the drawing may have crossed the zero-velocity-surface for those energy levels. Current VR application for this technique does not display the zero-velocity-surface for the user chosen energy level for the drawing, resulting in the user potentially drawing in the infeasible space of the system dictated by the energy level.

This same problem prevents certain drawings from becoming an initial-guess to solve for the periodic orbit in Technique-II as well. Recall that the CR3BP look-ahead trajectory emanating from the drawing is dependent upon the energy level specified by the user prior to the start of the drawing in this technique. If the CR3BP look-ahead trajectory is crossing the corresponding zero-velocity-surface for the input energy level, the initial-guess becomes infeasible. While this same issue of crossing zero-velocity-surfaces is not encountered in technique-III, it did turn out harder to produce initial-guesses for axial and vertical orbits in technique-III at certain energy levels beyond 2.9. Although a formal human subject experiment is not conducted as part of this dissertation, the users of technique-III did report additional difficulty and time-consumption in constructing initial-guesses in this technique. Technique-IV on the other-hand is reported to be the easiest to use and find solutions, since there are only initial-position and velocity to be manipulated by the user to construct an initial-guess. Moreover, in the technique an initial-guess resembling the desired orbit for only half a revolution is proven to yield a good solution as can be seen from results presented in Section 5.4. In all four of the techniques, the convergence rate is very high. A solution is almost always found using one or more of the corrector algorithms. Also, the convergence constraint violation of the order of  $-4$  or less is yielding good solutions. However, no correlation among the corrector algorithms could be found with these four techniques. In general, I have had to use all the corrector algorithms to produce results in these four techniques. Despite that, the collocation algorithm from Python (CasADi) implementation seems to do better in terms execution time among all the corrector algorithms.

## Chapter 6

### Transfer Trajectory Identification on the Virtual Reality Platform

This chapter presents the verification results for the transfer trajectory identification technique from Chapter 4. For the verification of the technique, two example transfers from Pritchett's dissertation are considered. In the first example, an implementation flow-chart 4.12 is considered with a sample transfer from a distant retrograde orbit (DRO) to an  $L_3$  Lyapunov orbit, with the aid of either  $L_4$  or  $L_5$  short-period orbits (SPOs). The choice of source, intermediate and target orbit families comes from one of Pritchett's verification examples for COLT [19].

An example transfer for this implementation is shown in Figures 6.1, 6.2, 6.3 and 6.4. In Figure 6.1, one can see the source, intermediate and target orbits along with the clipped parts of those orbits to compose the initial-guess for the transfer. In Figure 6.2, one can see the solution blue trajectory that starts from the chosen starting point on the source DRO and ends at the chosen end point on the  $L_3$  Lyapunov. Figure 6.3 is showing the mass consumption over the time-of-flight taken for the transfer. Figure 6.4 is showing the thrust profile for the transfer. Table 6.1 is listing the solution parameters from the COLT transfer algorithm.

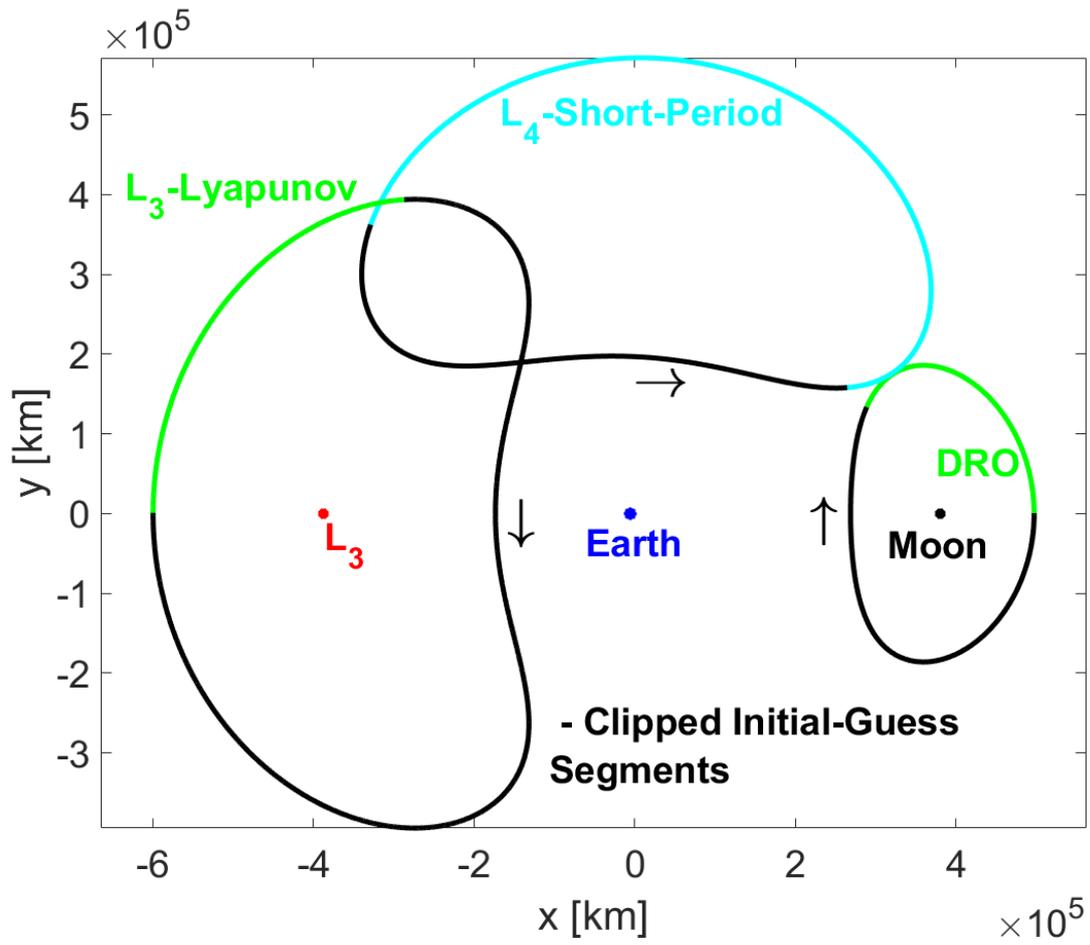


Figure 6.1: Initial-Guess Construction on the VR Platform for a DRO to  $L_3$  Lyapunov Transfer

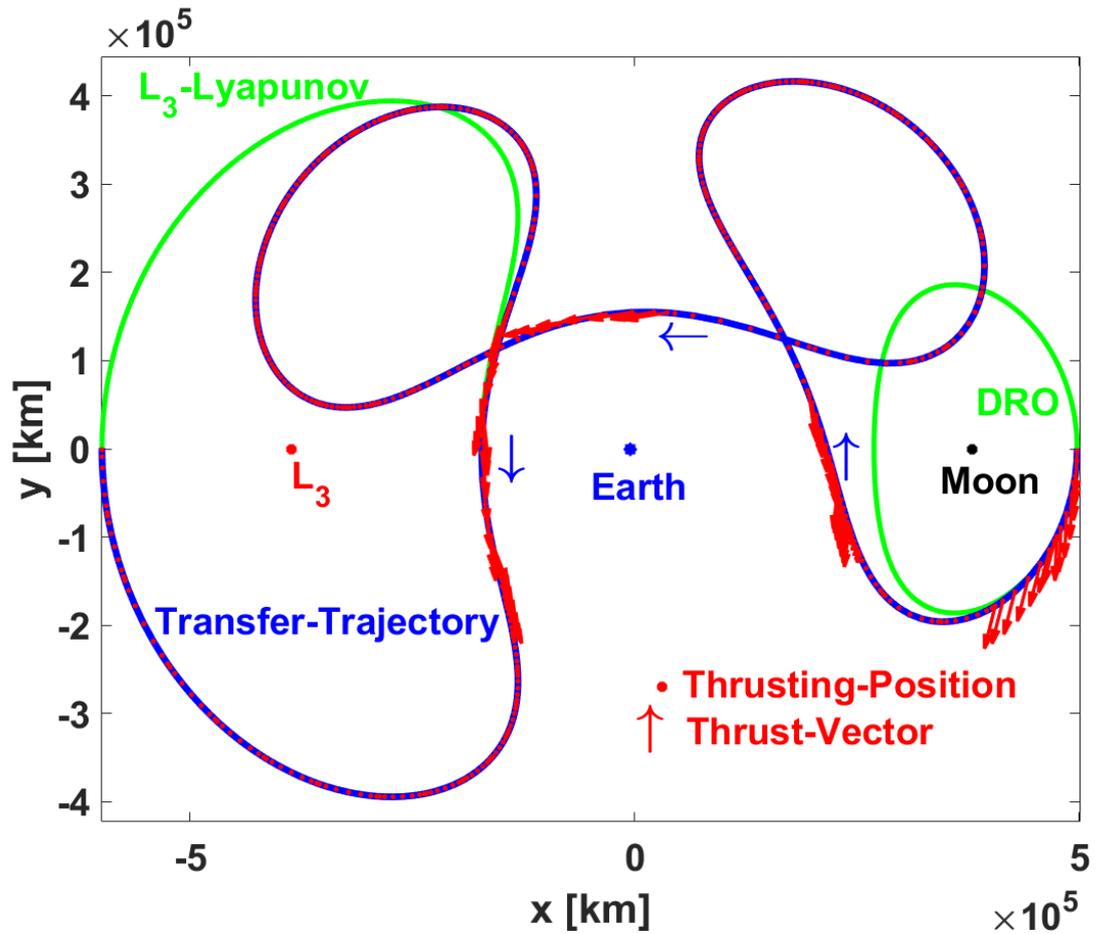


Figure 6.2: The Solution for a DRO to  $L_3$  Lyapunov Transfer

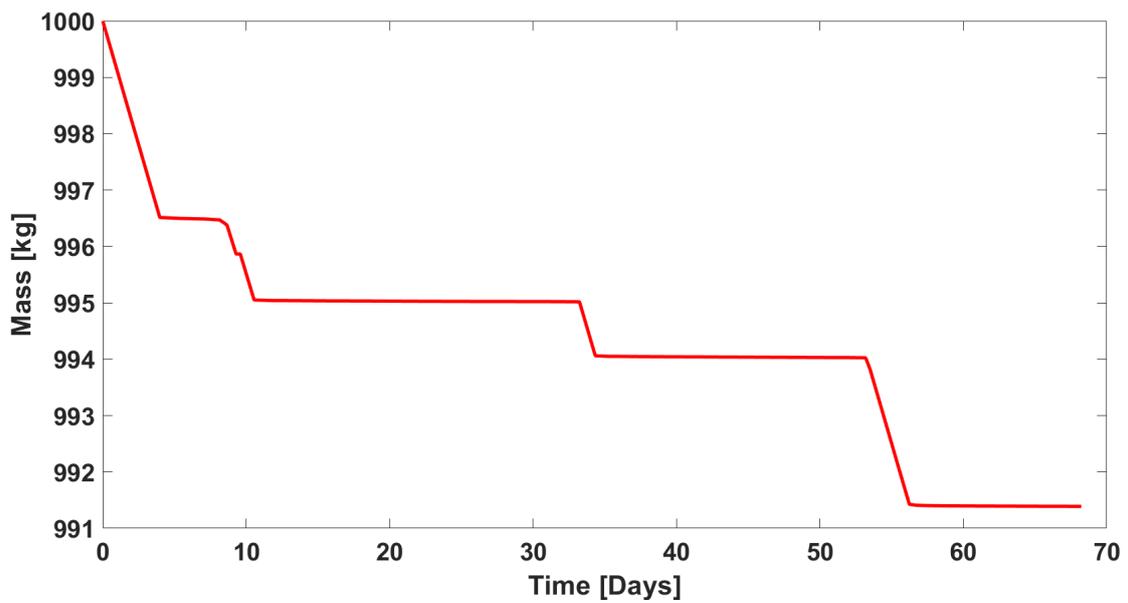


Figure 6.3: Mass Consumption for a DRO to  $L_3$  Lyapunov Transfer

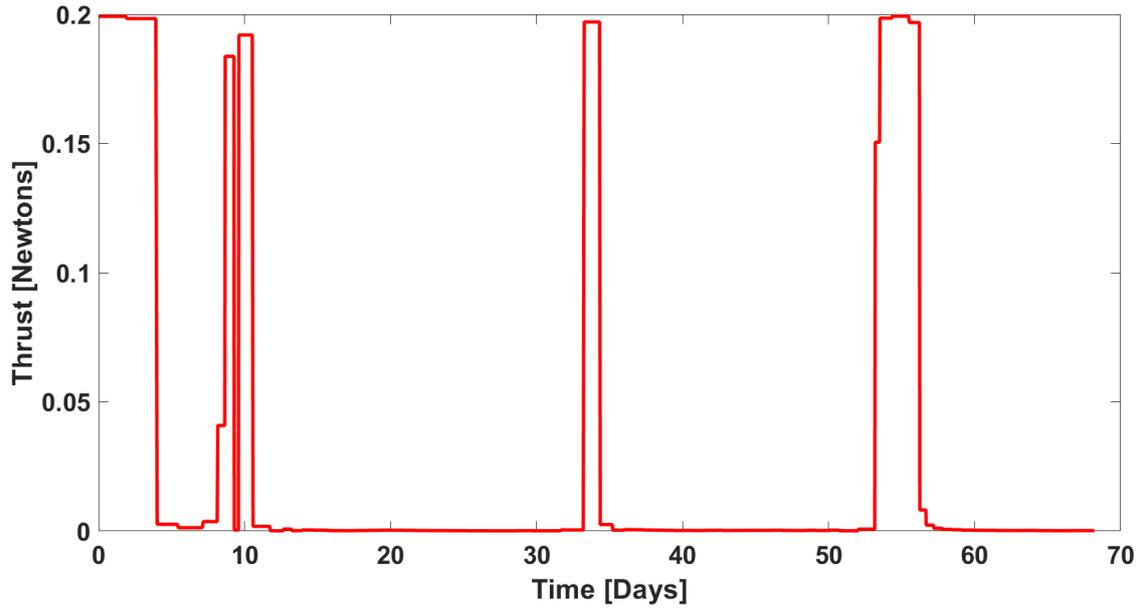


Figure 6.4: Thrust Profile for a DRO to  $L_3$  Lyapunov Transfer

Table 6.1: Solution Parameters for a DRO to  $L_3$  Lyapunov Transfer

Parameter	Value
Arcs	75
Time-of-Flight [days]	68.218365976461058
Mass Consumed [kg]	8.6112706695555516
Constraint-Violation	$9.414111e - 07$

In the second example, an implementation flow-chart 4.12 is considered with a sample transfer from a 9:2 Synodic resonance northern rectilinear halo orbit (NRHO) to a distant retrograde orbit (DRO), with the aid of 3 : 4 resonant orbit family. The choice of source, intermediate and target orbit families comes from one of Pritchett’s verification examples for COLT [19].

An example transfer for this implementation is shown in Figures 6.5, 6.6, 6.7 and 6.8. In Figure 6.5, one can see the source, intermediate and target orbits along with the clipped parts of those orbits to compose the initial-guess for the transfer. In Figure 6.6, one can see the solution blue trajectory that starts from the chosen starting point on the source NRHO and ends at the chosen end point on the target DRO. Figure 6.7 is showing the mass consumption over the

time-of-flight taken for the transfer. Figure 6.8 is showing the transfer thrust profile. Table 6.2 is listing the solution parameters from the COLT transfer algorithm.

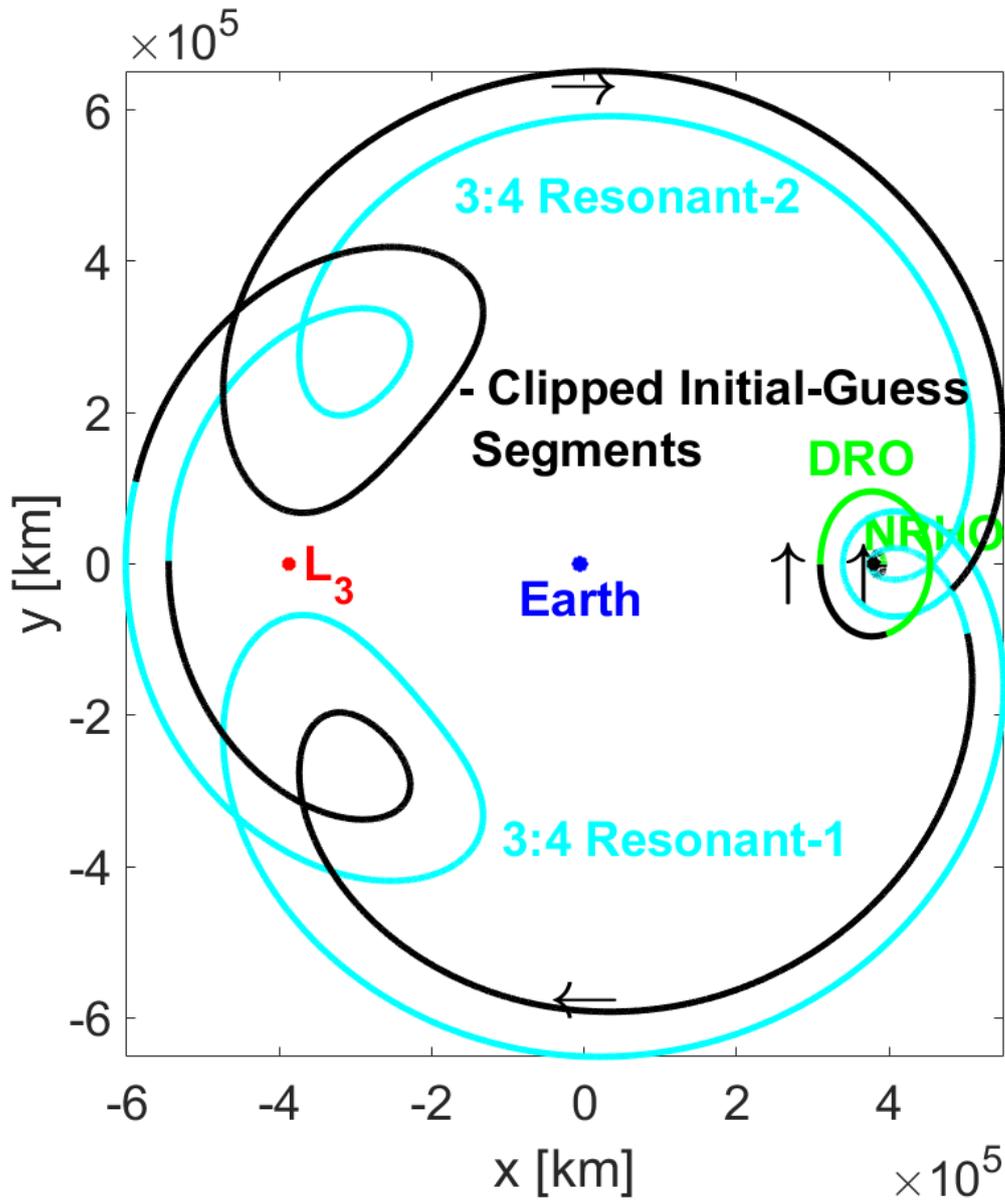


Figure 6.5: Initial-Guess Construction on the VR Platform for an NRHO-to-DRO Exterior Transfer

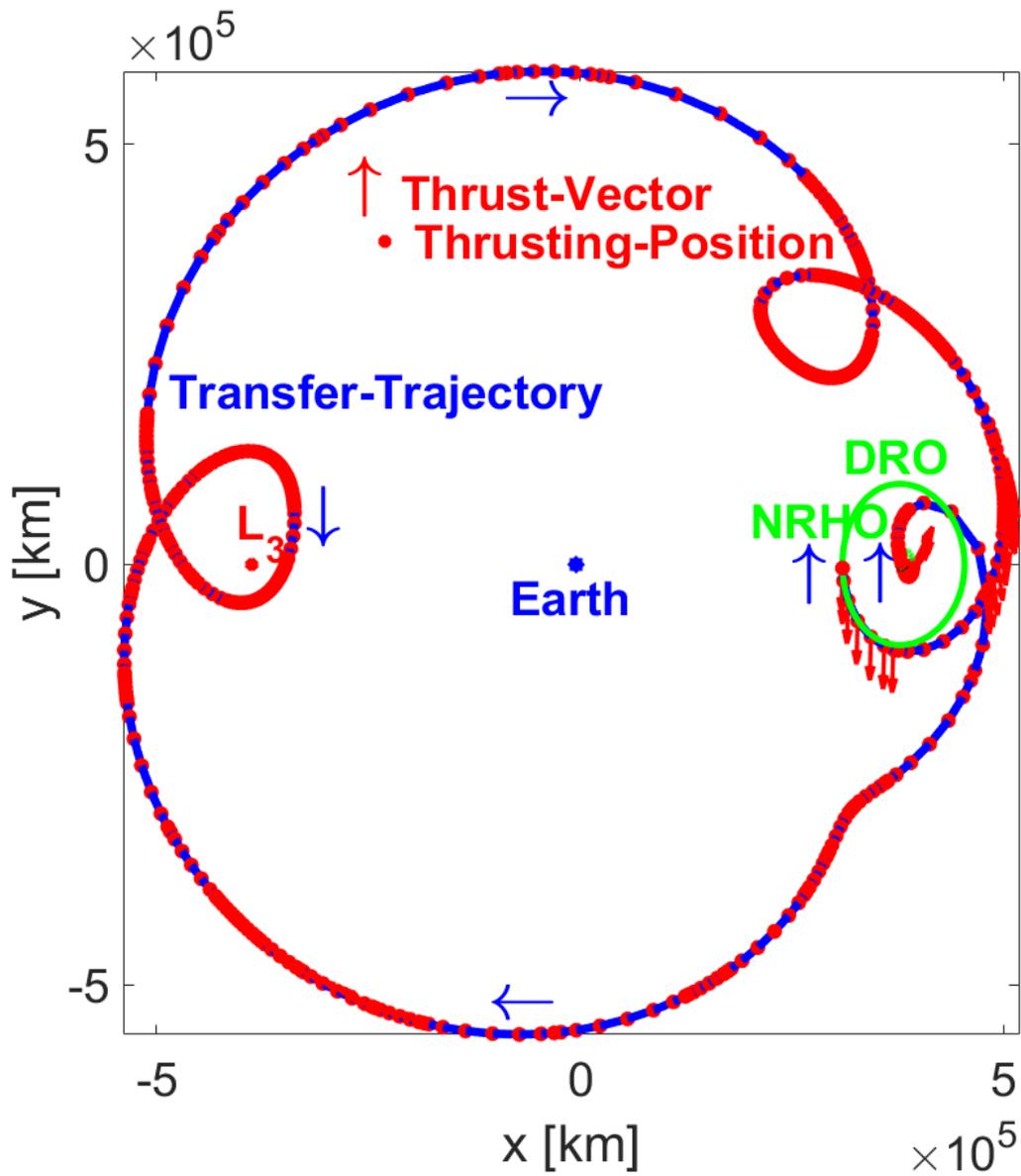


Figure 6.6: The Solution for an NRHO-to-DRO Exterior Transfer

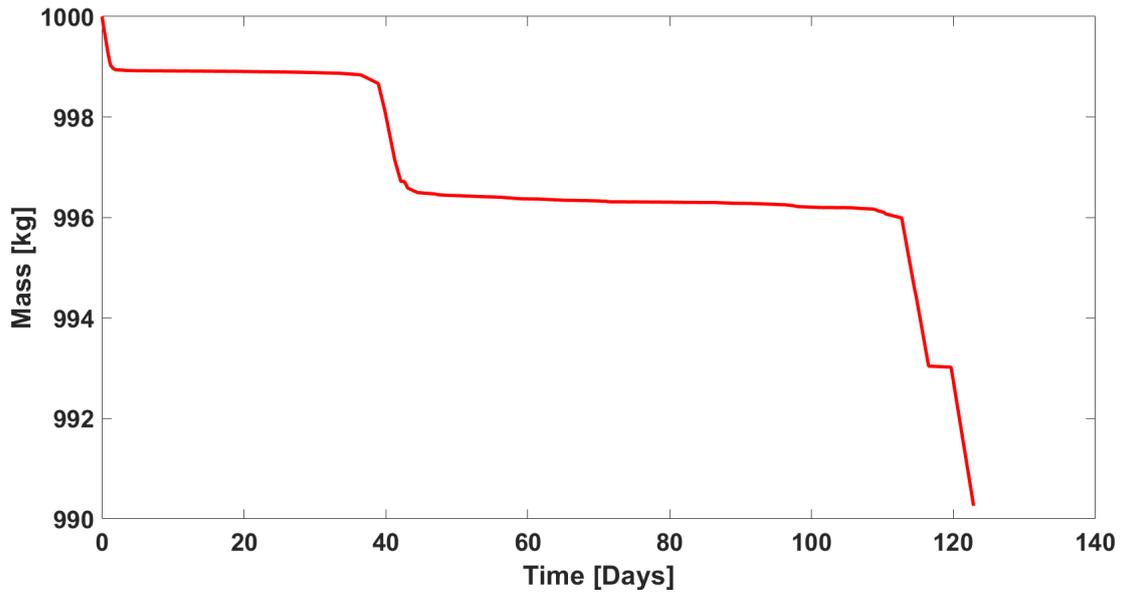


Figure 6.7: Mass Consumption for an NRHO-to-DRO Exterior Transfer

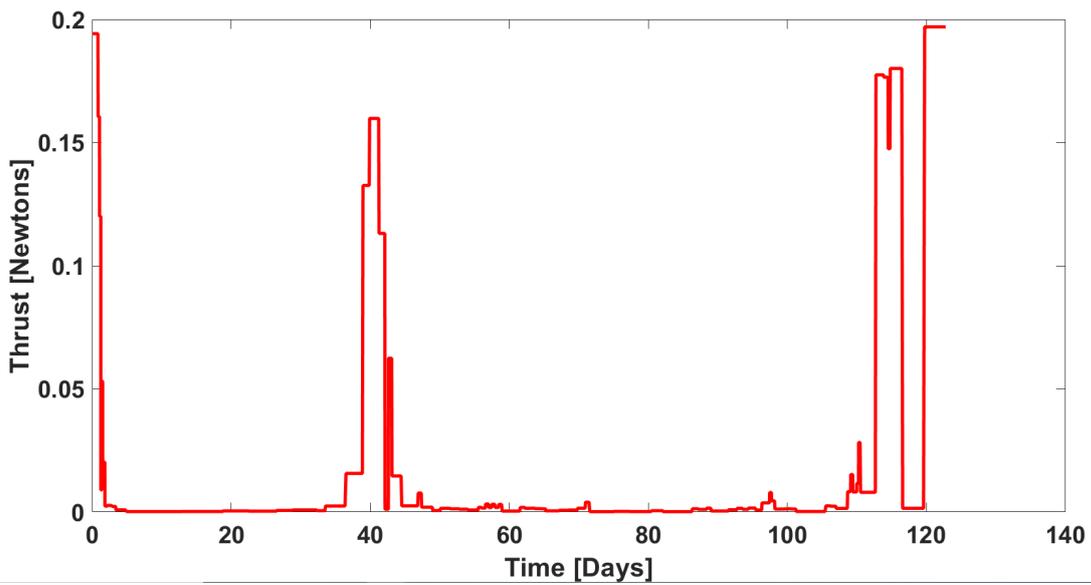


Figure 6.8: Thrust Profile for an NRHO-to-DRO Exterior Transfer

Table 6.2: Solution Parameters for an NRHO-to-DRO Exterior Transfer

Parameter	Value
Arcs	90
Time-of-Flight [days]	122.83050385584505
Mass Consumed [kg]	9.7394677774597085
Constraint-Violation	$9.051082e - 07$

## 6.1 Comparison of Results

The transfer trajectory solutions in Figures 6.2 and 6.6 with those of Pritchett's [19] are presented here. Pritchett's results in comparison are shown in Figures 6.9 and 6.10.

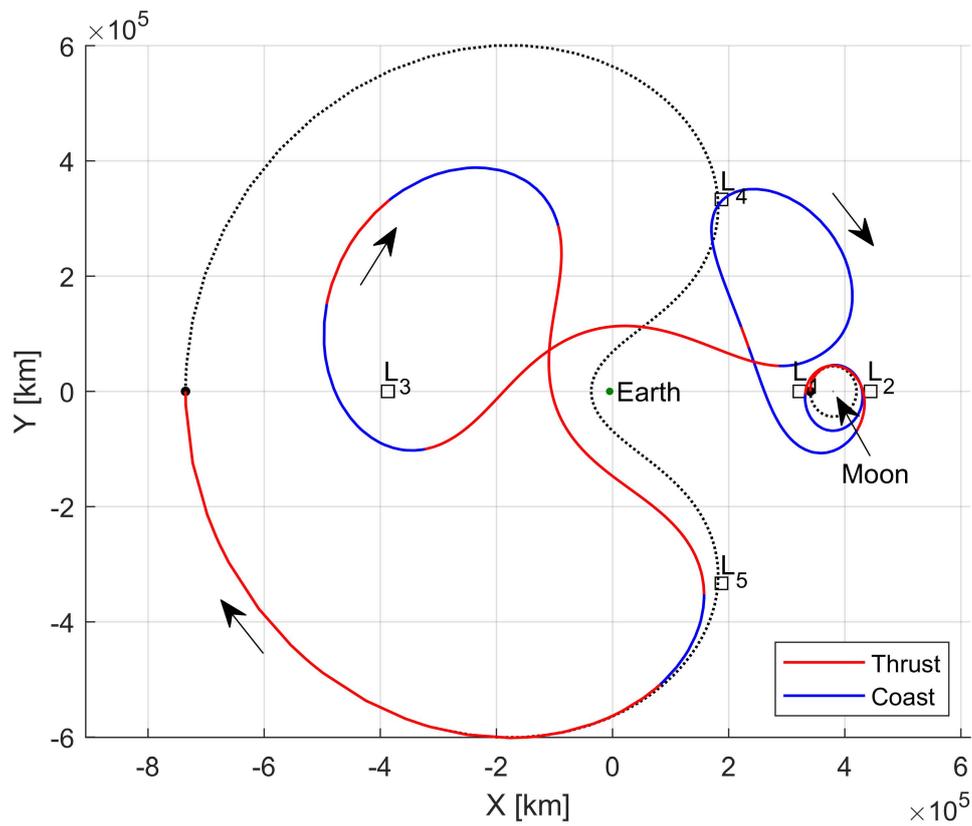


Figure 6.9: Pritchett's [19] Solution for a DRO to  $L_3$  Lyapunov Orbit Transfer

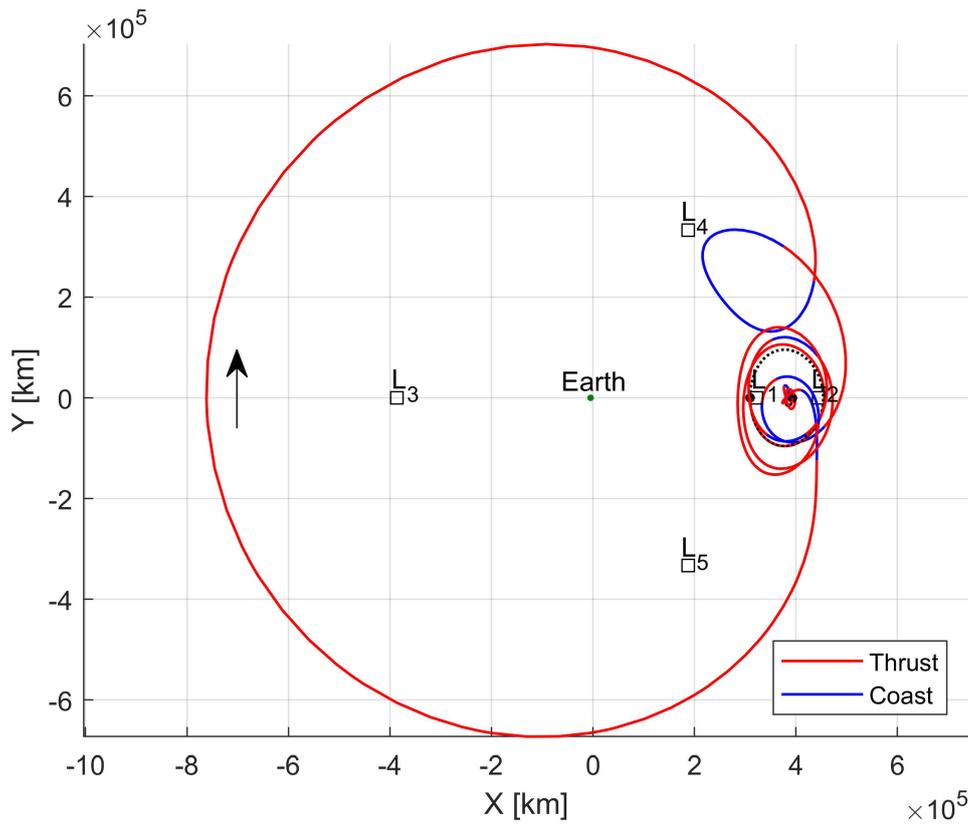


Figure 6.10: Pritchett's [19] Solution for an NRHO-to-DRO Exterior Transfer

For the first example of a DRO to  $L_3$  Lyapunov orbit transfer, the engine parameters and the spacecraft initial mass are kept the same as that of Pritchett's work. However, the source, intermediate and target orbits are not exactly the same. This deviation is taken to prove that the VR transfer technique is resilient to any arbitrary orbits. Despite this deviation, one see that in comparison, the general shape of the solution orbits between the two are similar. Pritchett reported that it took 79 days for this orbital transfer with 55 kg of fuel consumption. Whereas in my example, one can see from Table 6.1 the orbital transfer was completed in approximately 68.22 days and with a far less fuel consumption of approximately 8.61 kg.

For the second example of an NRHO to DRO exterior transfer, the engine parameters and initial mass are not kept the same as that of Pritchett's work. This is because Pritchett employs thrust continuation to arrive at his solution in this case and this process is deemed too slow for a VR application. However, unlike the first example, the source and target orbits are same as that of Pritchett's work. But the intermediate orbits are different than his work. As expected

the comparison of results illustrates that the optimal transfer trajectory structures are different owing to the different engine parameters, initial mass and lack of thrust continuation technique. Despite this dissimilarity, one can see in current optimal trajectory in blue in Figure 6.6 that the solution is reasonably good.

## Chapter 7

### Conclusion

The virtual-reality technology platform continues to evolve to make the use of the hardware more convenient. It is envisioned that the technology will eventually integrate seamlessly into human lives and activities. As it stands today, as is shown in the work from this dissertation, the real-3D aspect of virtual-reality technology opens up new avenues of solution methodologies in fields that are highly visual and demand high human cognitive load.

#### 7.1 Investigation Summary

Frameworks and VR applications for efficiently producing initial guesses for periodic orbit and orbital transfer trajectory identification are designed, developed and verified as part of this dissertation. The complex and chaotic cislunar CR3BP system demands good initial-guesses to the powerful numerical algorithms available to solve trajectory design and optimization problems in this realm. This dissertation provides new approaches to the construction of good initial-guesses leveraging the powerful third spatial dimension provided by the virtual reality platform.

#### 7.2 Periodic Orbit Identification Techniques

There are four techniques presented for the problem of periodic orbit identification. The first technique is free of all dynamical structures during the initial-guess construction process. And this makes the orbit identification algorithms execute for a longer time, however, there is more freedom to explore the search space of the problem. However, the initial-guess construction process is still subject to what dynamics allow for motion, and hence the technique could be

enhanced to let the human user see the zero-velocity-surfaces in the 3D space. Technique-II is an augmented version of Technique-I where the human user is provided some direction for the drawing via a CR3BP look-ahead trajectory emanating from the drawing. This technique provides a CR3BP conformant initial-guess for the numerical corrector algorithms to identify the matching periodic orbit faster, while still suffering from the same zero-velocity-surface crossing problem as that of Technique-I. Technique-III provides a more granular control over the initial-guess construction by allowing intermediate patch point selection for the initial-guess. This method is useful for when there are way points that one needs in the periodic orbit. Although this technique is also potentially subject to the zero-velocity-surface crossing problem theoretically, in practice no human user ran into this problem. Finally, technique-IV is a pure CR3BP dynamical approach to the good initial-guess construction problem. In this method, there are no restrictions on the energy level of the initial-guess. The energy level is purely dictated by the initial-position and velocity chosen by the human user. This technique proved to be the fastest in terms of initial-guess construction and periodic orbit identification. It also turned out that matching first half of the intended periodic orbit is sufficient in this technique. In practice all four techniques provided good periodic orbits from the initial-guesses, except for when running into the zero-velocity-surface crossing problem in the first three techniques.

### 7.3 Transfer Trajectory Identification Technique

There is one technique presented for the problem of orbital transfer trajectory identification. In this technique, the orbit-chain method is implemented to leverage the real 3D space of the virtual-reality platform, making the technology a good fit for the Analyze, Select and Clip steps of the method. The technique produced results that are comparable to baseline test results from Pritchett's dissertation, from where the orbit-chain method and test cases are borrowed. Compared to the traditional transfer trajectory identification methods, this VR based method seems to benefit from faster identification of the initial-guess.

#### 7.4 Future Work Recommendations

There is one obvious choice for future work for the periodic orbit identification techniques. The obvious choice is to improve them with the display of the zero-velocity-surfaces to the human user during the initial-guess construction process. There is also another obvious choice for the transfer trajectory identification technique is to leverage the manifolds in addition to the orbits for the initial-guess construction.

One not so obvious or perhaps a grandiose choice for future work is to build a system like GMAT or Copernicus in VR. Tools like GMAT and Copernicus allow for manipulation of the control parameters easily for the initial-guess construction but on the 2D desktops with some 3D augmentations to the initial-guess construction process. This 2D or the trying to operate in a 3D manner on a desktop is not highly intuitive for the human user. Making a general mission design tool like GMAT or Copernicus in VR has the potential to make the space trajectory design not only intuitive, but also fun and easily accessible.

## References

- [1] Bate, Roger R., Donald D. Mueller and Jerry E. White, “Fundamentals of Astrodynamics,” Dover Publications Inc, New York, 1971.
- [2] Stuart, Jeffrey, Basak Ramaswamy, Try Lam, Nathaniel Guy, Frank Laipert, Alex Menzies, Nicholas Bradley, Aprameya Mysore, and Nitin Arora. ”Do You See What I See? Interactive Visualization of Mission Design and Navigation,” *69th International Astronautical Congress*. Bremen, Germany, 2018.
- [3] Mathur, Ravishankar. “Visual Interactive Trajectory Design,” Emergent Space Technologies Inc., Greenbelt, MD, 2015.
- [4] Johnson, Greg, Sebastioan Munoz, and Jennifer Lehman. “Copernicus: A Generalized Trajectory Design and Optimization System.” UT Austin Engineering Communication ASE 333T, available online (2003).
- [5] Hughes, Steven P., Rizwan H. Qureshi, Steven D. Cooley, and Joel J. Parker. “Verification and Validation of the General Mission Analysis Tool (GMAT).” *In AIAA/AAS astrodynamics specialist conference*, p. 4151., San Diego, CA, 2014.
- [6] Newton, Isaac. “The Principia: Mathematical Principles of Natural Philosophy,” University of California Press, 1999, Translated by Bernard Cohen and Anne Whitman.
- [7] Barrow-Green, June. “Poincaré and the Three Body Problem.” *American Mathematical Society*, No. 11, 1997.
- [8] Szebehely, Victor. “Theory of Orbits: The Restricted Problem of Three Bodies.” Academic Press, Inc., New York, 1967.

- [9] Hill, George William. “Researches in the Lunar Theory.” *American Journal of Mathematics* 1, no. 1 (1878): 5-26.
- [10] Poincaré, Henri. “Les Méthodes Nouvelles de la Mécanique Céleste.” *Gauthier-Villars et fils*, Vol. 3, 1899.
- [11] Grebow, Daniel J. “Generating Periodic Orbits in the Circular Restricted Three-Body Problem with Applications to Lunar South Pole Coverage.” M.S. Thesis, Purdue University (2006): 1-165.
- [12] Guzzetti, Davide, Natasha Bosanac, Amanda Haapala, Kathleen C. Howell, and David C. Folta. “Rapid Trajectory Design in the Earth–Moon Ephemeris System via an Interactive Catalog of Periodic and Quasi-Periodic Orbits.” *Acta Astronautica* 126 (2016): 439-455.
- [13] Zimovan-Spreen, Emily. “Trajectory Design and Targeting For Applications to the Exploration Program in Cislunar Space.” Ph.D. Dissertation, Purdue University, 2021.
- [14] Koon, Wang Sang, Martin W. Lo, Jerrold E. Marsden, and Shane D. Ross. “Dynamical Systems, the Three-body Problem and Space Mission Design.” *Equadiff 99: International Conference on Differential Equations*, Berlin, Germany, August, 1999.
- [15] Ross, Shane, and Martin Lo. “The Lunar L1 Gateway-Portal to the Stars and Beyond.” *AIAA Space 2001 Conference and Exposition*, p. 4768., Albuquerque, NM, 2001.
- [16] Lo, Martin W., and Jeffrey S. Parker. “Chaining Simple Periodic Three Body Orbits.” *Advances in the Astronautical Sciences* 123 (2006): 2027-2042.
- [17] Parker, Jeffrey S., Kathryn E. Davis, and George H. Born. “Chaining Periodic Three-Body Orbits in the Earth–Moon system.” *Acta Astronautica* 67, no. 5-6 (2010): 623-638.
- [18] Pritchett, Robert E. “Numerical Methods for Low-Thrust Trajectory Optimization.” M.S. Thesis, Purdue University, 2016.
- [19] Pritchett, Robert E., “Strategies for Low-Thrust Transfer Design based on Direct Collocation Techniques,” Ph.D. Dissertation, Purdue University, 2020.

- [20] Herbert B. Keller, “Numerical Methods for Two-Point Boundary Value Problems,” Blaisdell, 1968.
- [21] M. R. Osborne, “On Shooting Methods for Boundary Value Problems,” *Journal of Mathematical Analysis and Applications*, vol. 27, no. 2, pp. 417–433, 1969.
- [22] Thomas A. Pavlak, “Trajectory Design and Orbit Maintenance Strategies in Multi-Body Dynamical Regimes,” Ph.D. Dissertation, Purdue University, 2013.
- [23] Kelly, Matthew. “An Introduction to Trajectory Optimization: How to do your own Direct Collocation.” *SIAM Review* 59, no. 4 (2017): 849-904.
- [24] Williams, Paul. “Hermite-Legendre-Gauss-Lobatto Direct Transcription in Trajectory Optimization.” *Journal of guidance, control, and dynamics* 32, no. 4 (2009): 1392-1395.
- [25] Grebow, Daniel J, and Thomas A Pavlak. “MColl: Monte Collocation Trajectory Design Tool - NASA Technical Reports Server (NTRS).” NASA. NASA, 2017. <https://ntrs.nasa.gov/citations/20190028461>.
- [26] Acton, Charles H. “Ancillary Data Services of NASA’s Navigation and Ancillary Information Facility.” *Planetary and Space Science* 44, no. 1 (1996): 65–70.
- [27] Bosanac, Natasha. “Leveraging Natural Dynamical Structures to Explore Multi-Body Systems.” Ph.D. dissertation, Purdue University, 2016.
- [28] Unity Technologies. “3D Gaming Engine.” Unity, 2005. <http://www.unity3d.com/>.
- [29] HTC VIVE. “Vive VR Headsets, Immersive Glasses & Equipment: United States.”, 2016. <https://www.vive.com/us/product/#pro%20series>.
- [30] Jet Propulsion Laboratory. “Three-Body Periodic Orbits.” NASA, 2021. [https://ssd.jpl.nasa.gov/tools/periodic\\_orbits.html#/periodic](https://ssd.jpl.nasa.gov/tools/periodic_orbits.html#/periodic).
- [31] Andersson, Joel A., Joris Gillis, Greg Horn, James B. Rawlings, and Moritz Diehl. “CasADi: A Software Framework for Nonlinear Optimization and Optimal Control.” *Mathematical Programming Computation* 11, no. 1 (2018): 1–36.

- [32] Weinstein, Matthew J., and Anil V. Rao. “Algorithm 984: ADiGator, a Toolbox for the Algorithmic Differentiation of Mathematical Functions in MATLAB Using Source Transformation via Operator Overloading.” *ACM Transactions on Mathematical Software* 44, no. 2 (2017): 1–25.
- [33] Somavarapu, Dhathri H., and Davide Guzzetti. “Toward Immersive Spacecraft Trajectory Design: Mapping User Drawings to Natural Periodic Orbits.” *Acta Astronautica* 184 (2021): 208-221.

## Appendices

## Appendix A

### The VR Snapshots

#### A.1 Common Snapshots

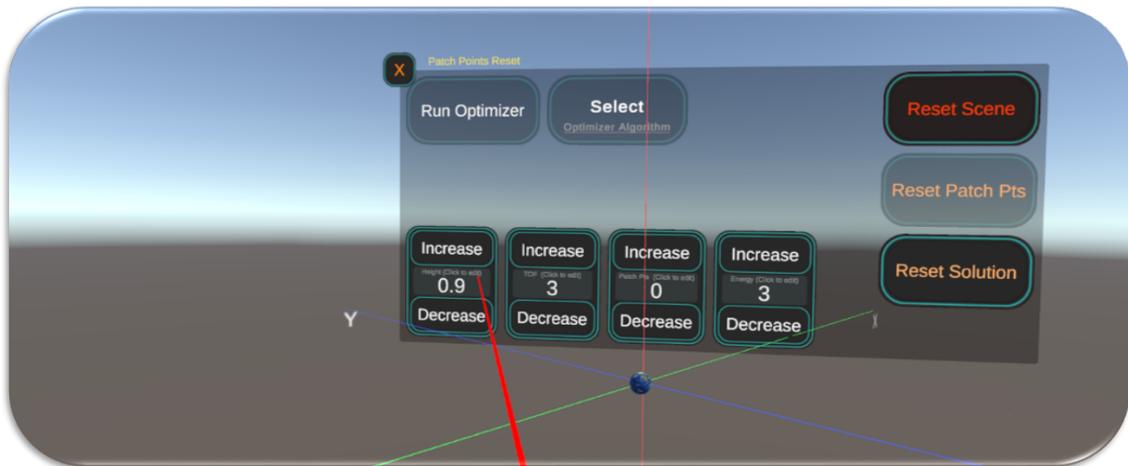


Figure A.1: AOH - A Depiction of the Adjustment of Height of CR3BP System in VR

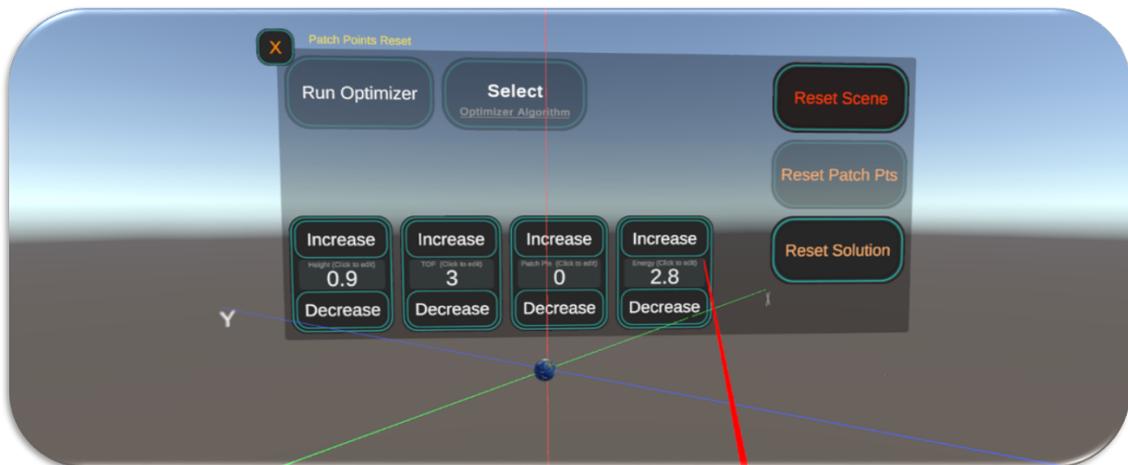


Figure A.2: AOE - A Depiction of the Adjustment of Energy of CR3BP System in VR

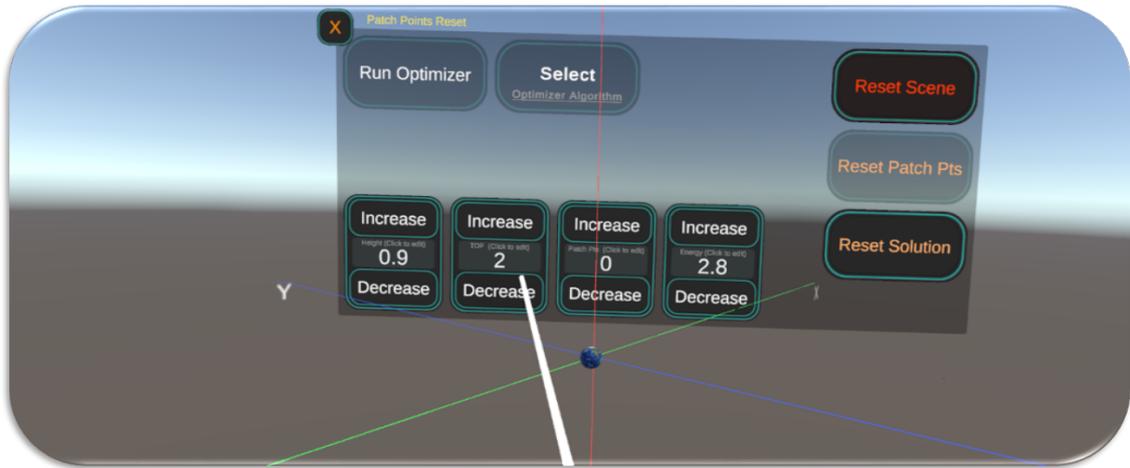


Figure A.3: AOTOF - A Depiction of the Adjustment of Time-of-Flight of CR3BP Look-Ahead Path in VR

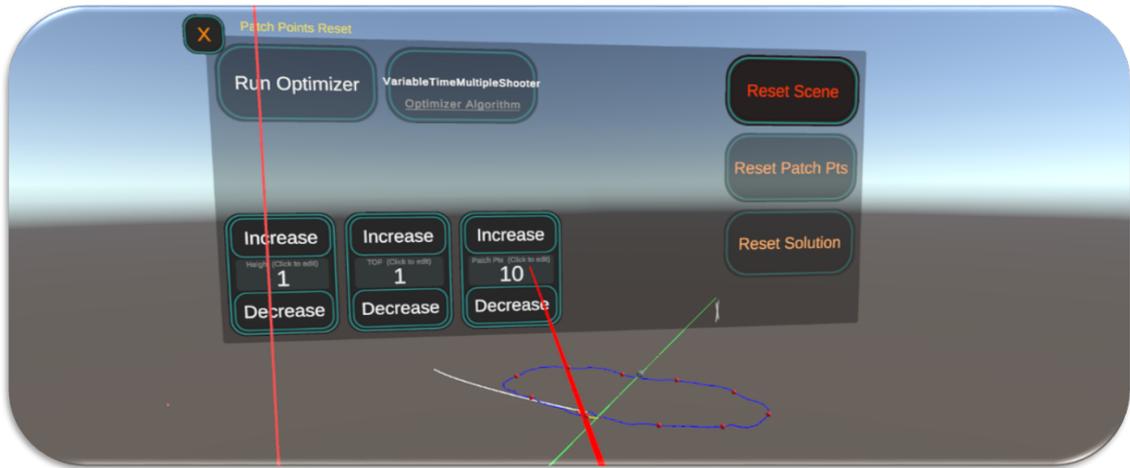


Figure A.4: AOPP - A Depiction of the Adjustment of Patch-Points in the Initial-Guess in VR

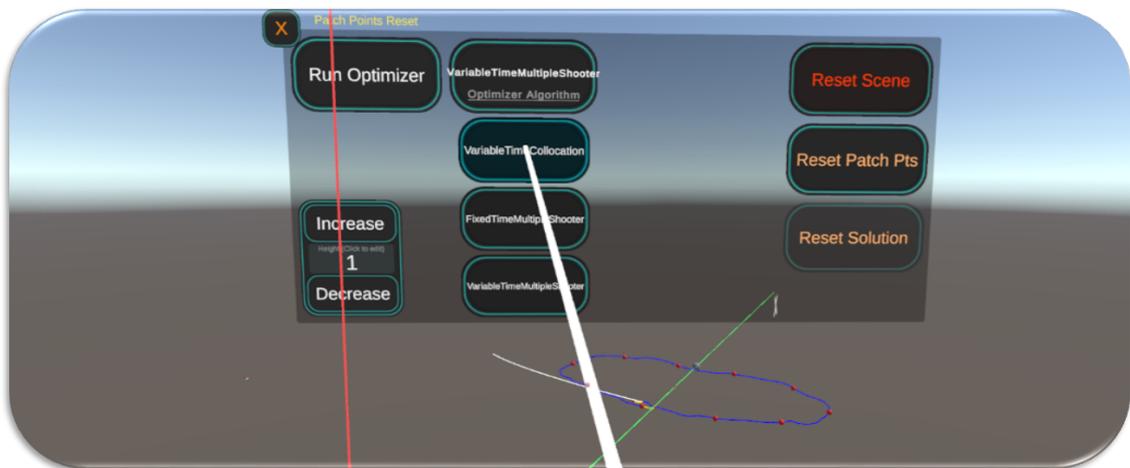


Figure A.5: SOP - A Depiction of the Selection Optimizer in VR

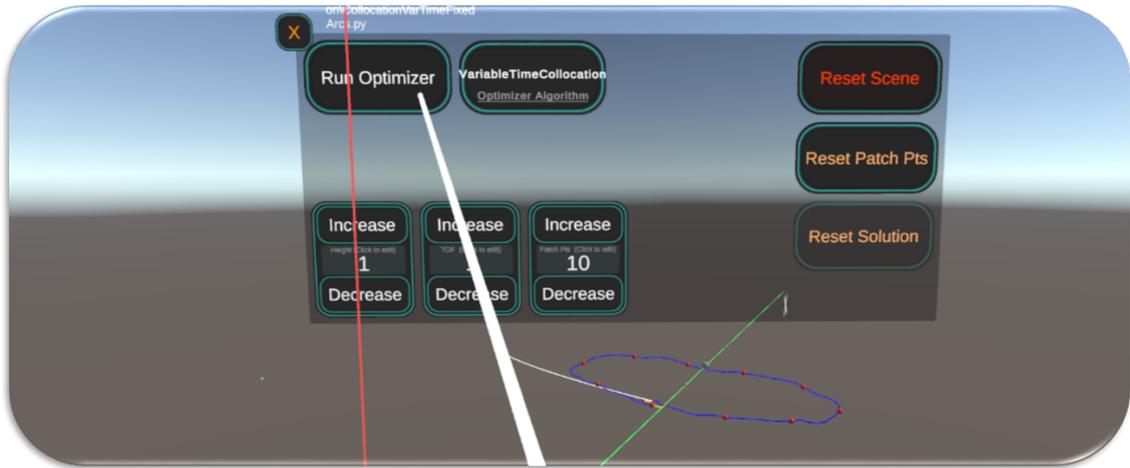


Figure A.6: ROP - A Depiction of the Running of Optimizer in VR

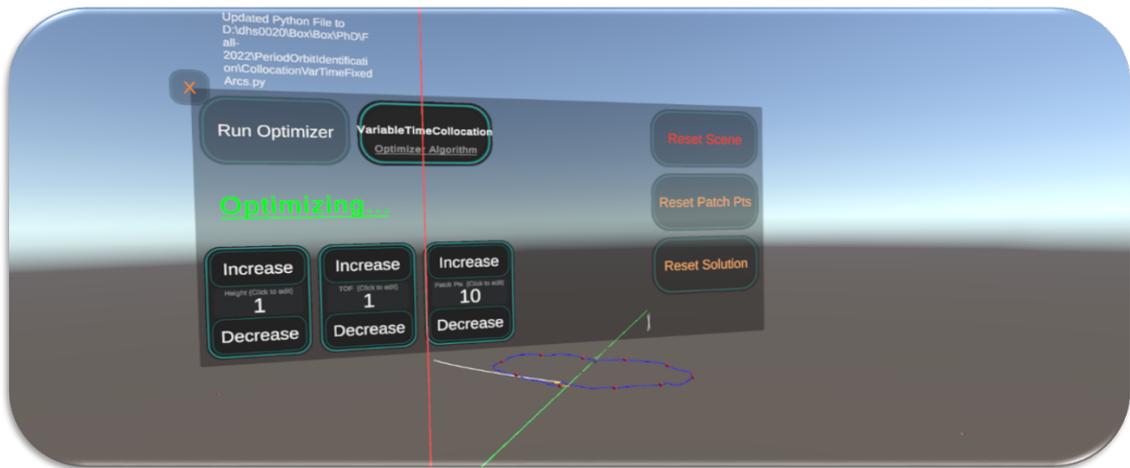


Figure A.7: WFS - A Depiction of the Waiting for Solution in VR

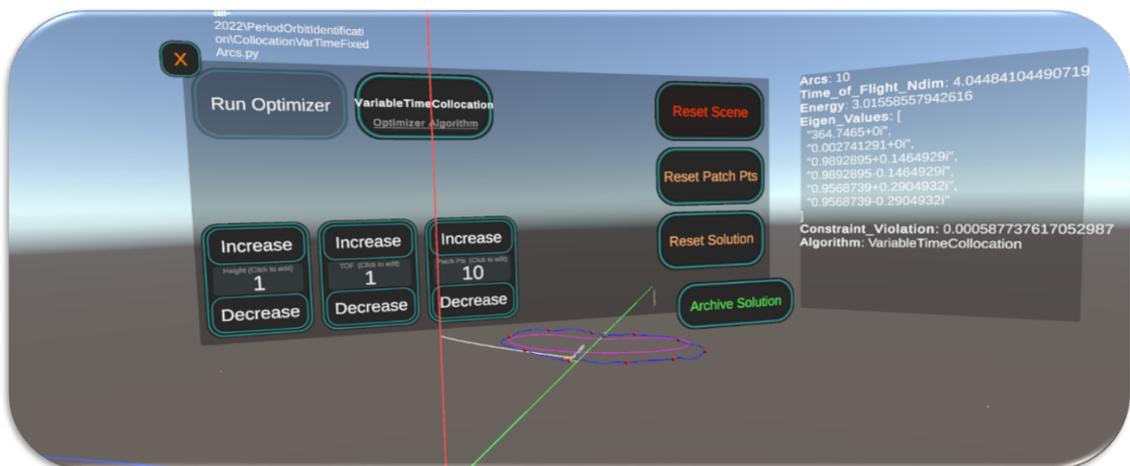


Figure A.8: ROS - A Depiction of the Review of Solution in VR

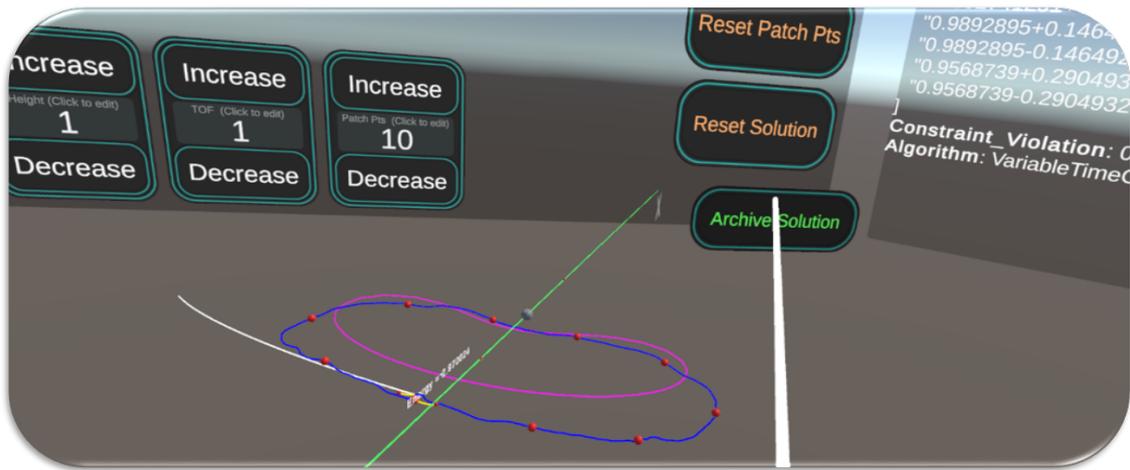


Figure A.9: AOS - A Depiction of the Archival of the Solution in VR

## A.2 Periodic Orbit Identification Snapshots

### A.2.1 Snapshots for Technique-II

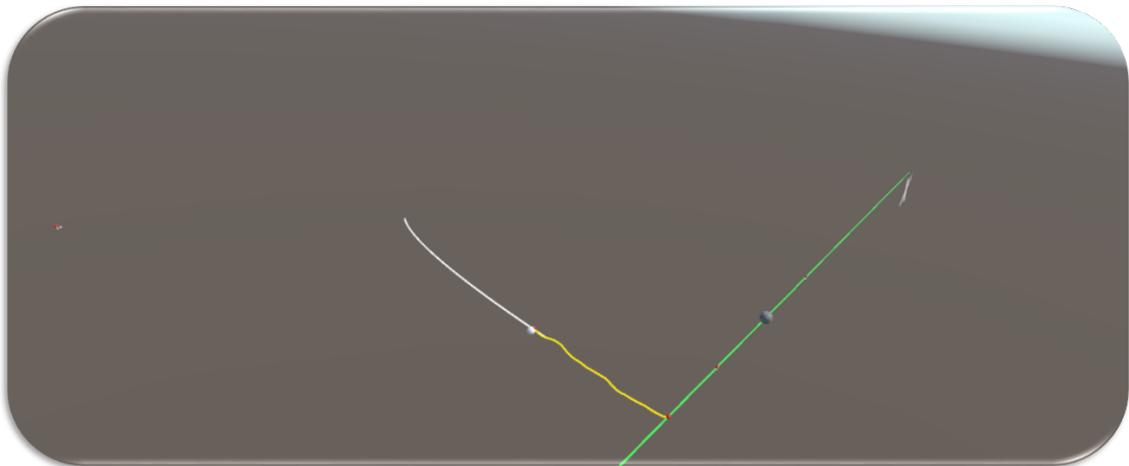


Figure A.10: SDRWG-II - A Depiction of the Starting of the Drawing in VR

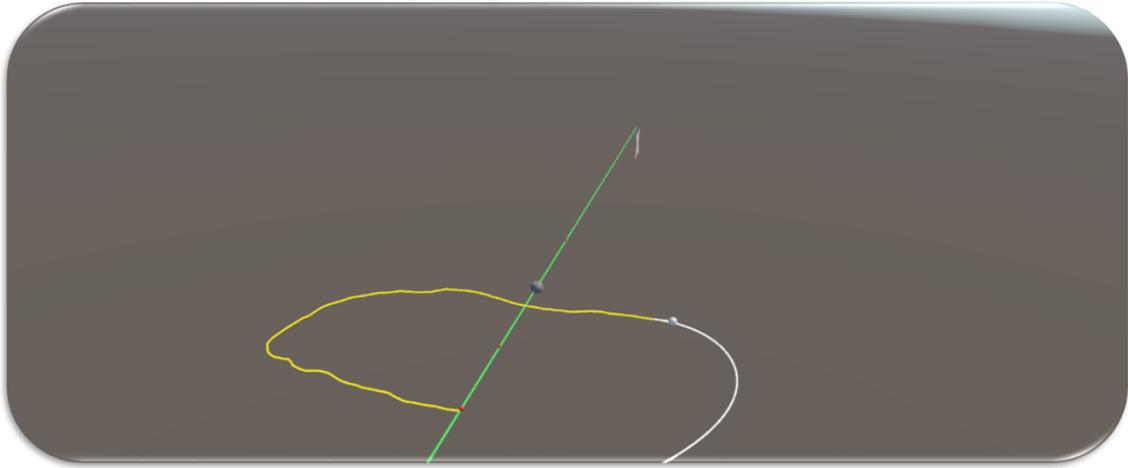


Figure A.11: CDRWG-II - A Depiction of the Continuation of the Drawing in VR

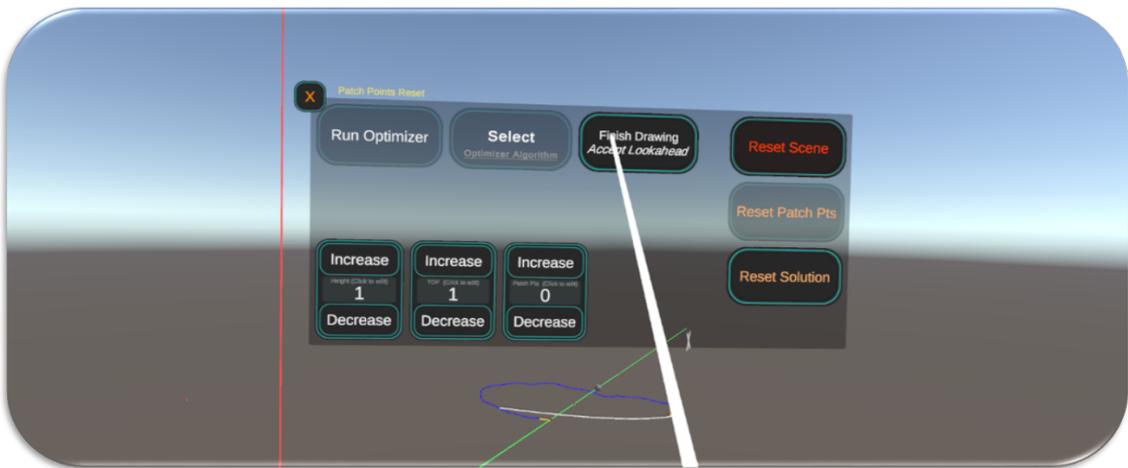


Figure A.12: ALKHD - A Depiction of the Acceptance of the CR3BP Look-Ahead Path in VR

### A.2.2 Snapshots for Technique-III

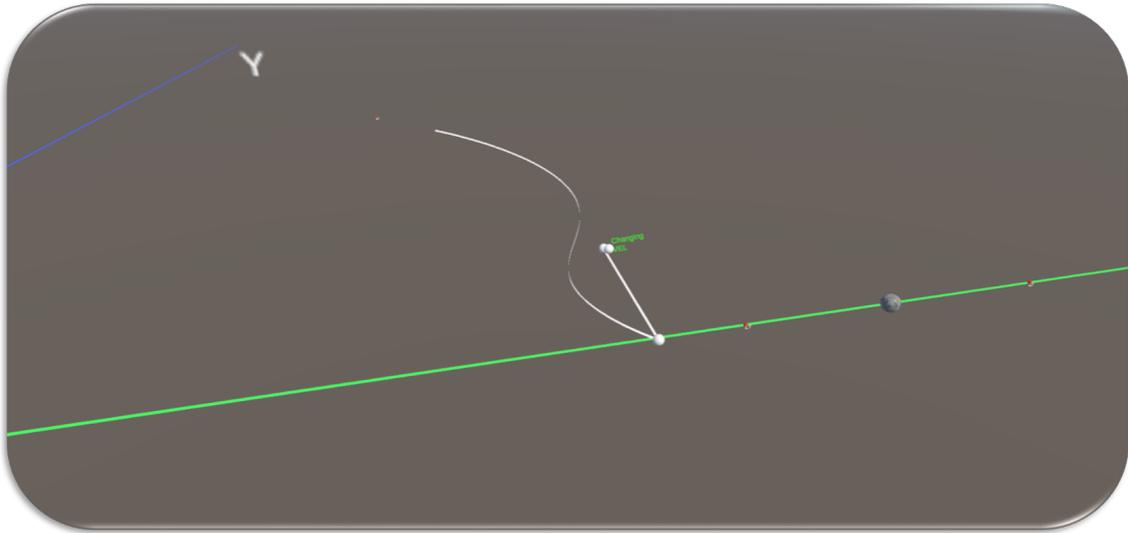


Figure A.13: AIPPT - A Depiction of the First Two Patch-Point Placement

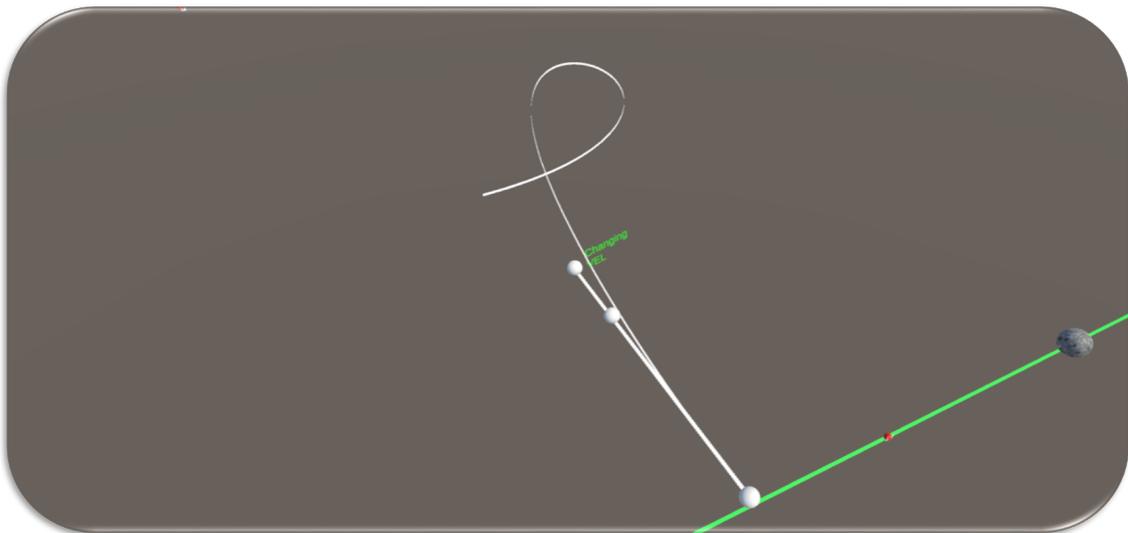


Figure A.14: MVAPPT - A Depiction of the Velocity Direction Vector Manipulation

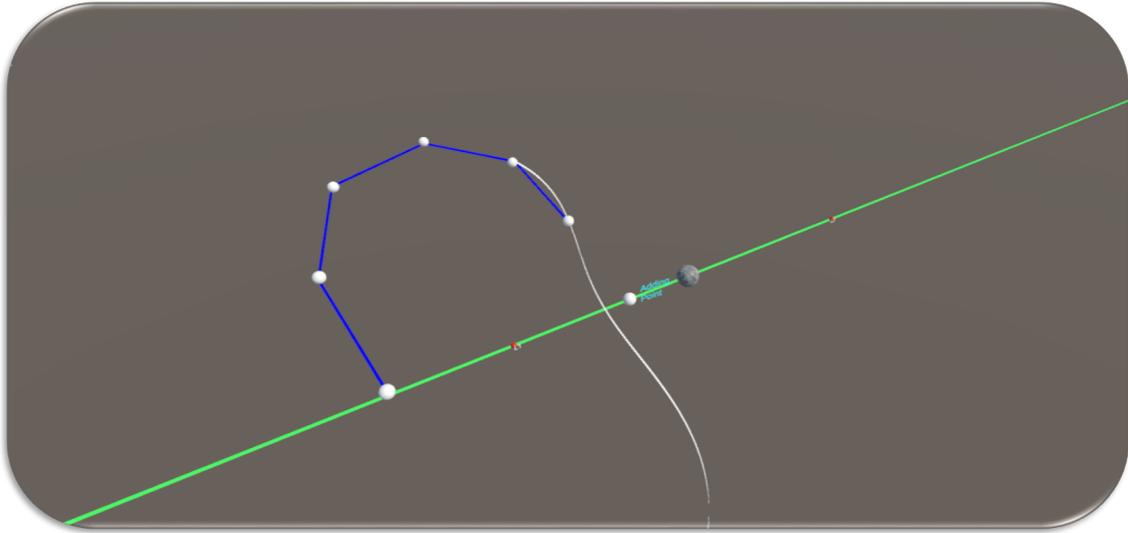


Figure A.15: ANPPT - A Depiction of the Continued Patch-Point Placement

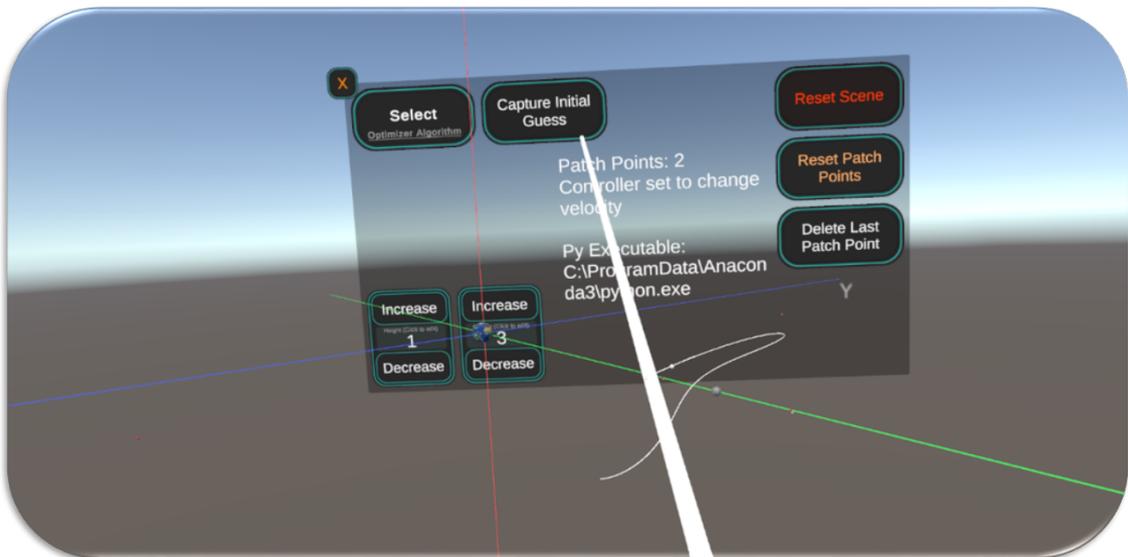


Figure A.16: CLKIG - A Depiction of the Capture of Initial Guess Action

### A.2.3 Snapshots for Technique-IV

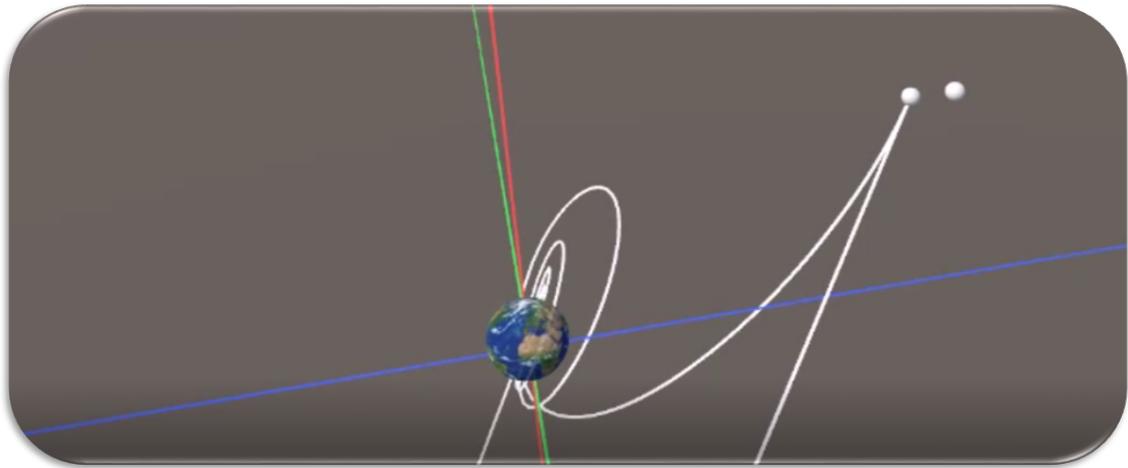


Figure A.17: APPT - A Depiction of the Placement of Initial Position Point

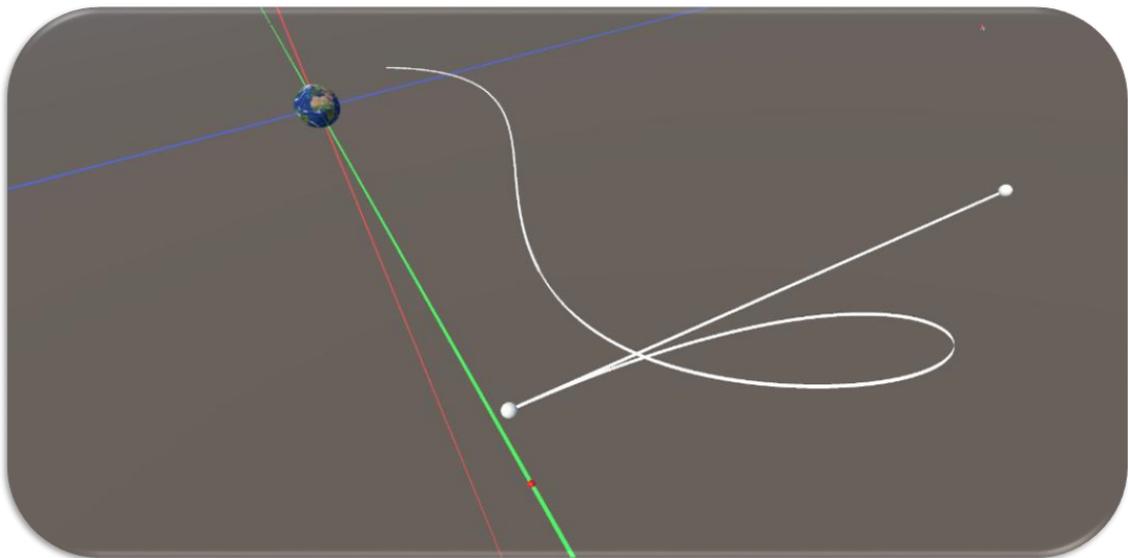


Figure A.18: MVV - A Depiction of the Manipulation of the Velocity Vector

### A.3 Transfer Trajectory Identification Snapshots

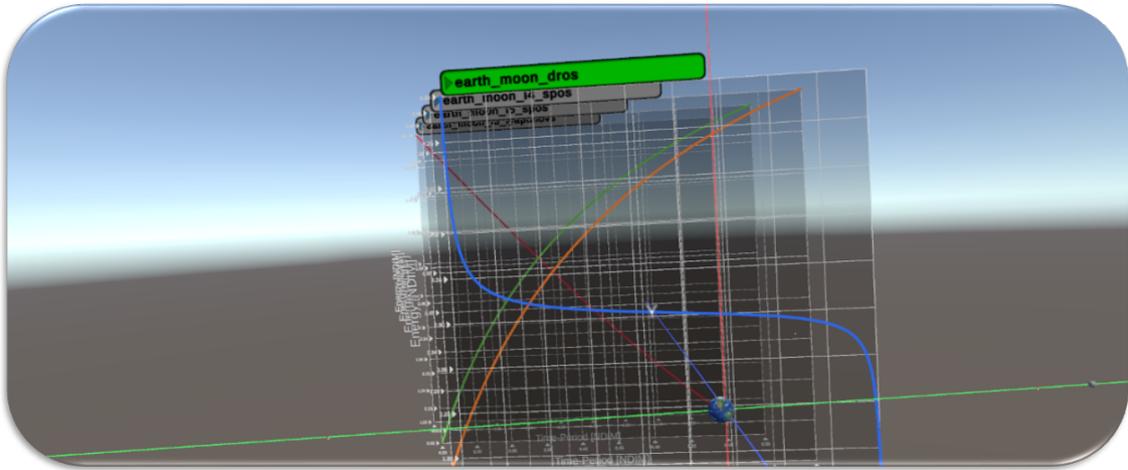


Figure A.19: A VR Orbital Transfer Scene with Energy Plots

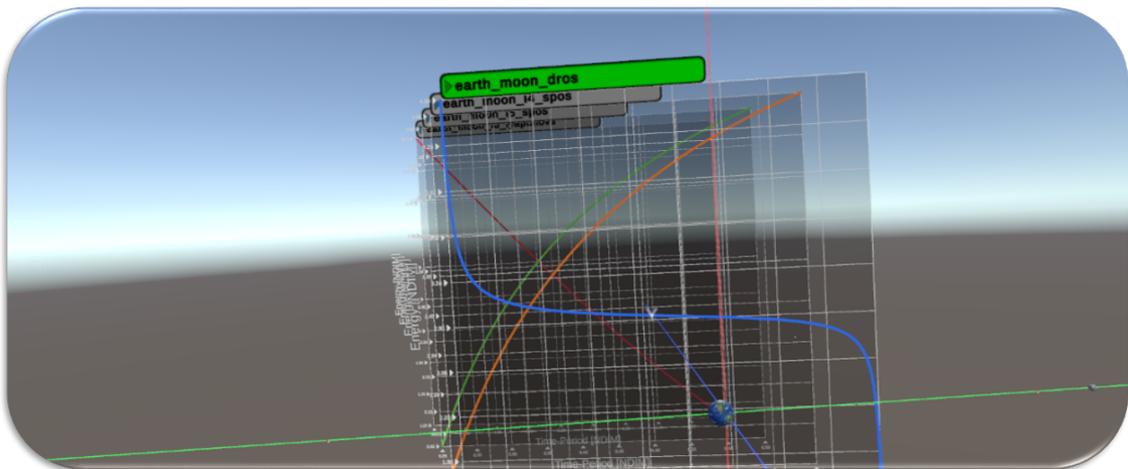


Figure A.20: SOSOP - A Depiction of the Source Orbit Energy Plot

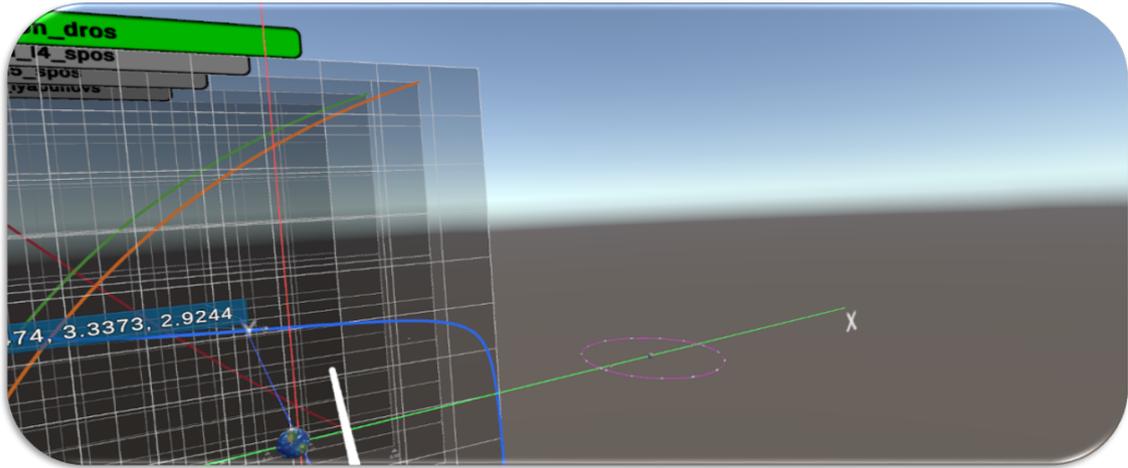


Figure A.21: SCRL-S - A Depiction of the Source Orbit Energy Plot Scrolling Action



Figure A.22: SOSO - A Depiction of the Selection of the Source Orbit

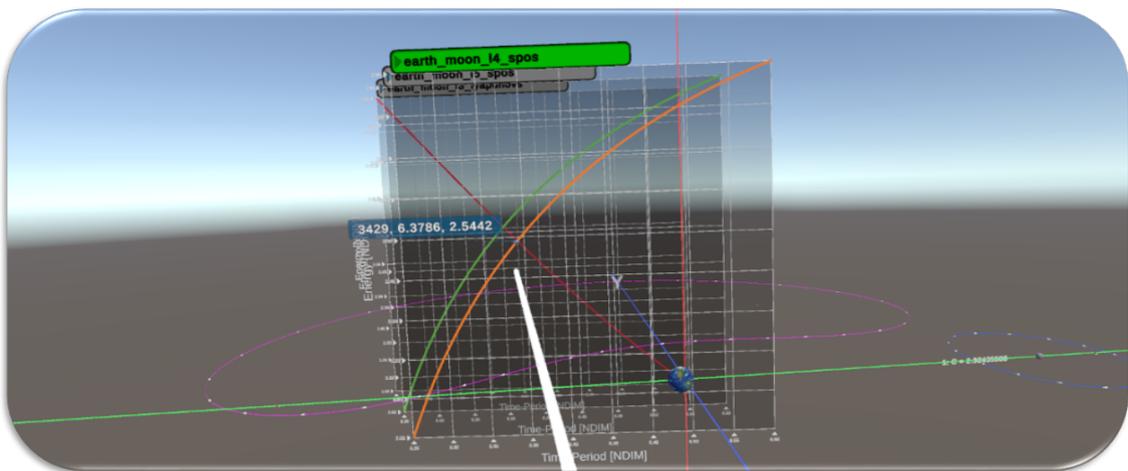


Figure A.23: SOIOP - A Depiction of the Intermediate Orbit Energy Plot

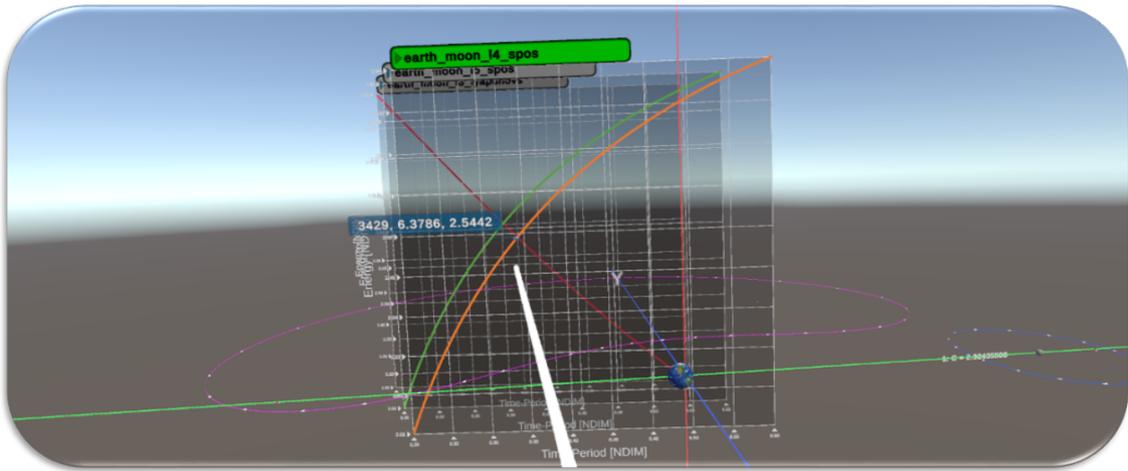


Figure A.24: SCRL-I - A Depiction of the Source Orbit Energy Plot Scrolling Action



Figure A.25: SOIO - A Depiction of the Selection off the Intermediate Orbit

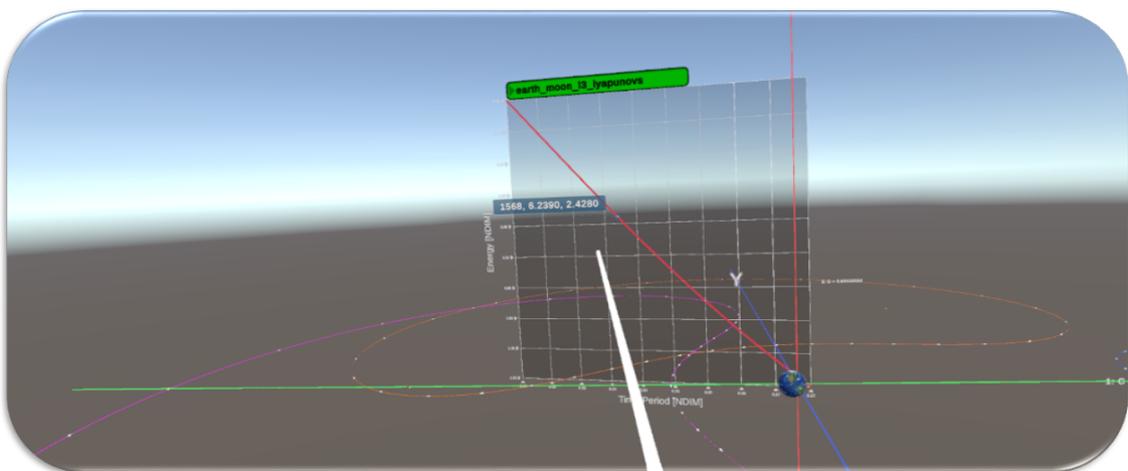


Figure A.26: SOTOP - A Depiction of the Target Orbit Energy Plot

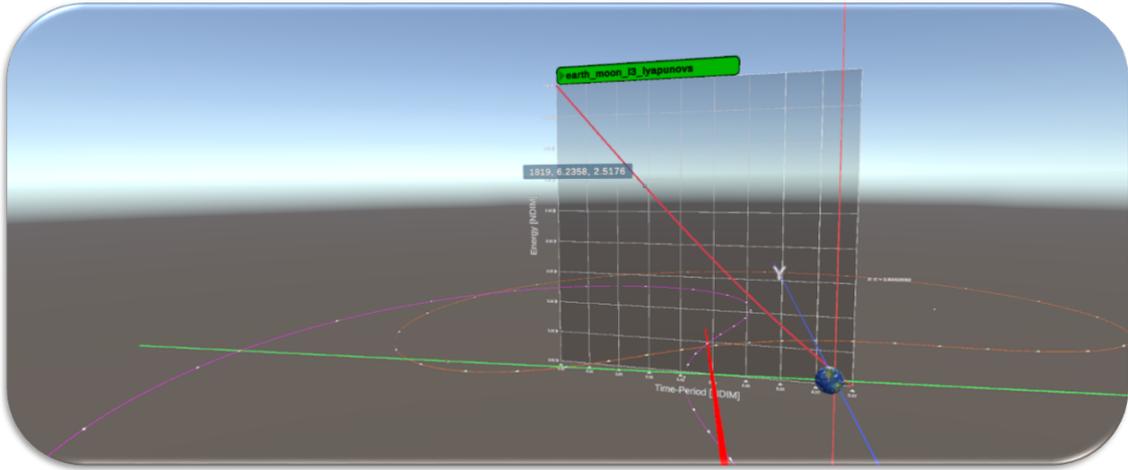


Figure A.27: SCRL-T - A Depiction of the Target Orbit Energy Plot Scrolling Action



Figure A.28: SOTO - A Depiction of the Selection of the Target Orbit

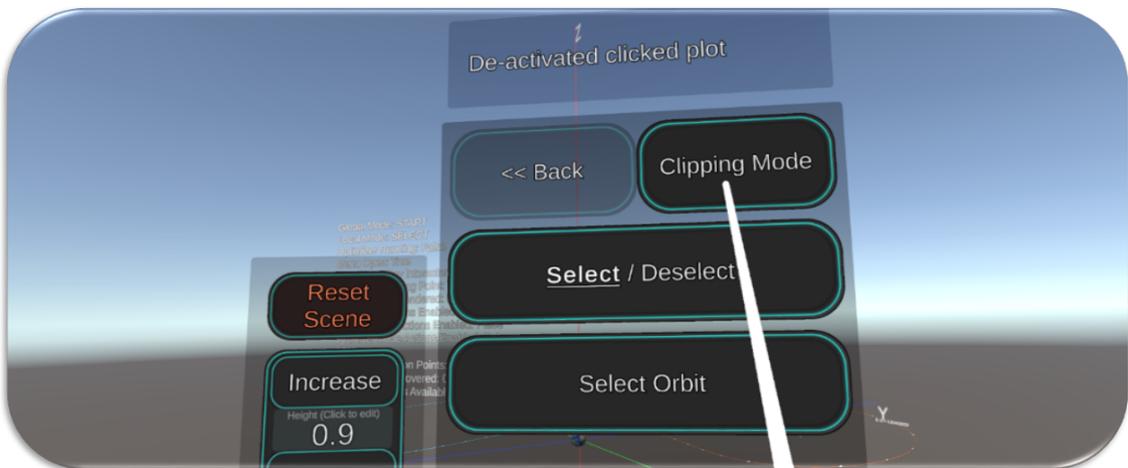


Figure A.29: SWTCLP - A Depiction of the Action to Switch to the Clipping Mode

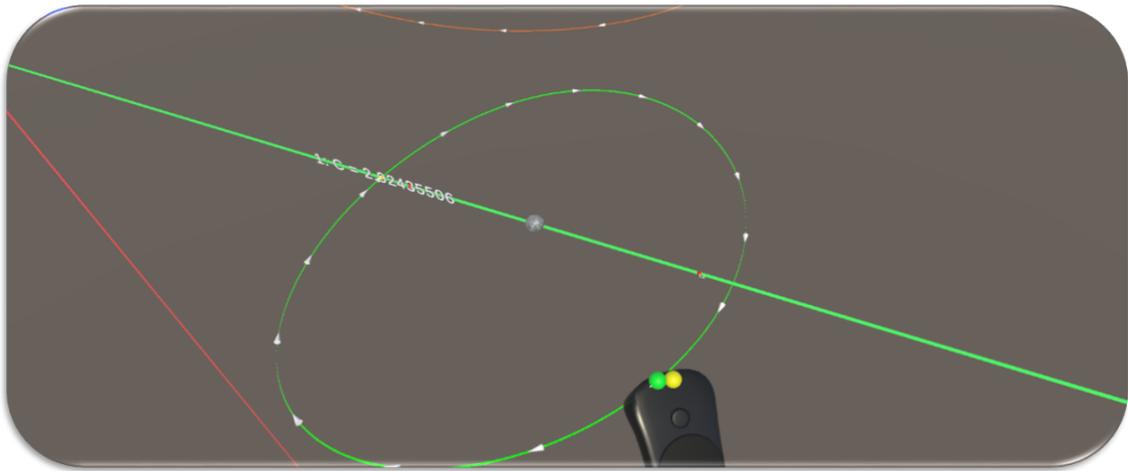


Figure A.30: CLIP - A Depiction of the Orbit Clipping Action



Figure A.31: SWTOPT - A Depiction of the Action to Switch to Optimization Mode

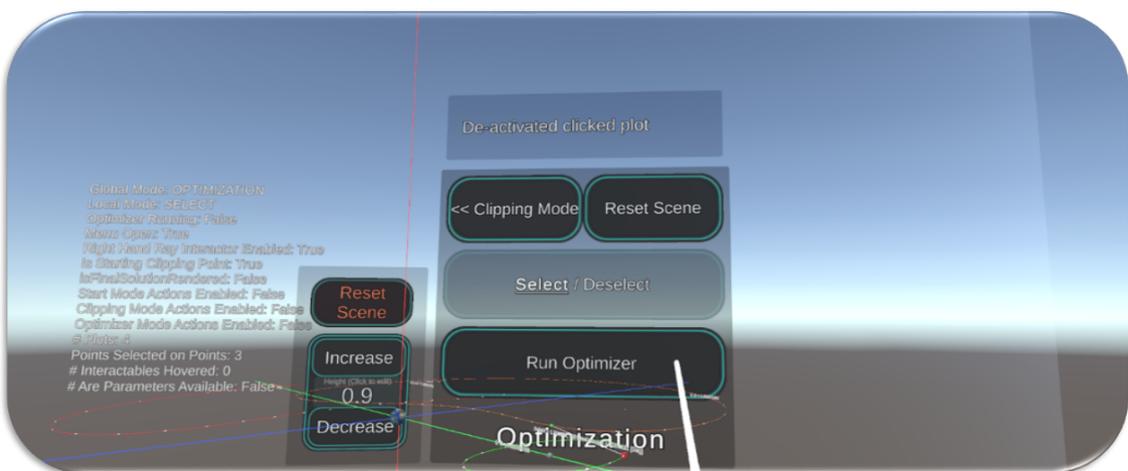


Figure A.32: ROP - A Depiction of the Action to Run Optimizer

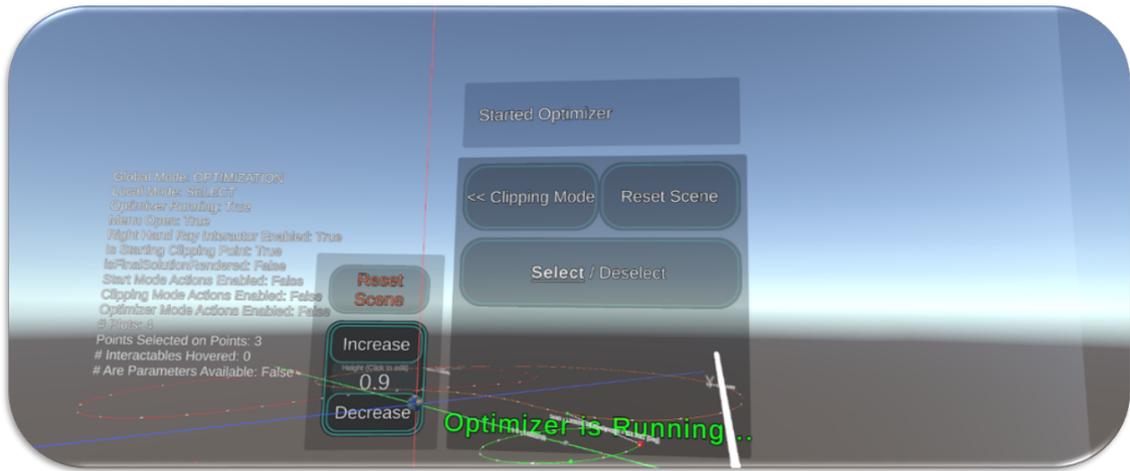


Figure A.33: WFS - A Depiction of the Action to Wait for the Solution

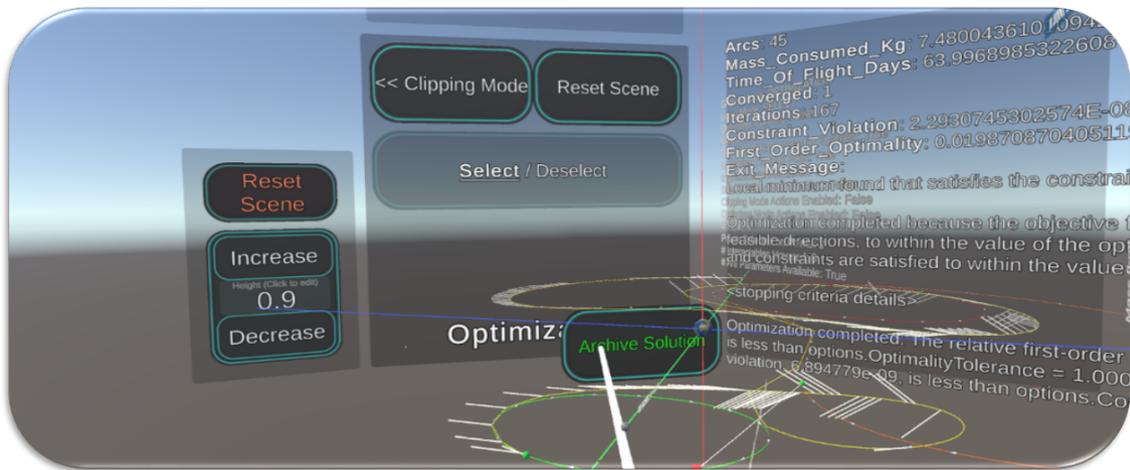


Figure A.34: ROTS - A Depiction of the Review of the Solution

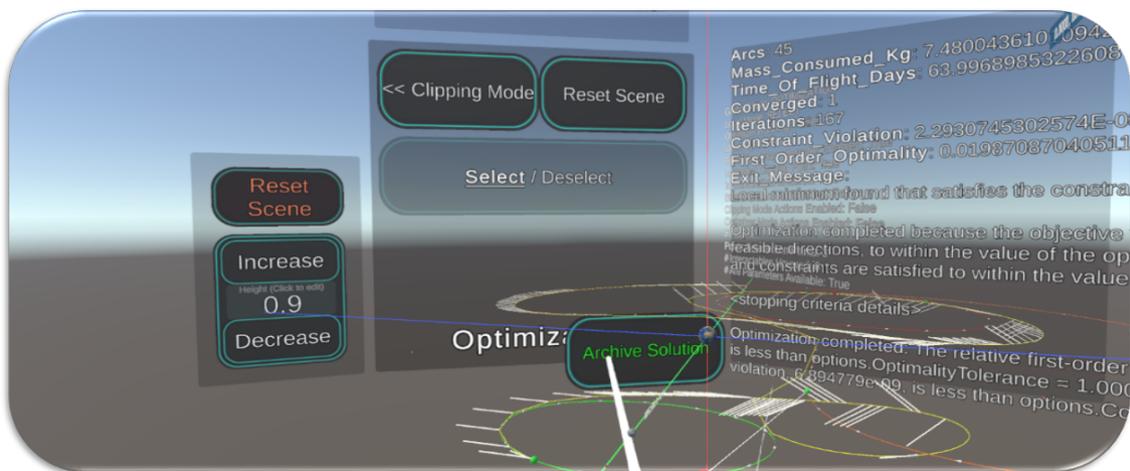


Figure A.35: AOTS - A Depiction of the Archival of the Solution