Collision-Free Spacecraft Trajectory Design in Multi-Body Systems

by

Renee Lynn Spear

B.S., Embry-Riddle Aeronautical University, 2019

M.S., University of Colorado Boulder, 2021

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Aerospace Engineering Sciences

2025

Committee Members:

Prof. Natasha Bosanac, Chair

Dr. Gregory Whiffen

Prof. Daniel Scheeres

Prof. Morteza Lahijanian

Prof. Emiliano Dall'Anese

Spear, Renee Lynn (Ph.D., Aerospace Engineering Sciences)

Collision-Free Spacecraft Trajectory Design in

Multi-Body Systems

Thesis directed by Prof. Natasha Bosanac

With the incorporation of cislunar space into operational territory, agile spacecraft will play a key role in successful cislunar operations by maneuvering and replanning trajectories in response to hazardous situations. Heritage methods for this task in geocentric operations are insufficient in chaotic regimes, such as cislunar space, due to the increased complexity and sensitivity of the environment. Furthermore, humans-in-the-loop typically design, and redesign if necessary, collision-free spacecraft paths for hazard avoidance and space domain regulation. Hence, the development of new approaches to safe spacecraft trajectory design is critical to future robust and agile cislunar operations while reducing the burden on humans-in-the-loop.

This research addresses these challenges by formulating collision-free trajectory design for cislunar spacecraft operations as a path planning problem with dynamic obstacles. First, clustering is used to generate geometrically-distinct obstacle corridors from trajectories emanating from local initial state sets in the Circular Restricted Three-Body Problem (CR3BP). Then, traditional path planning techniques from robotics are leveraged to create a hierarchical tree-based initial guess construction framework for rapidly generating geometrically-distinct spacecraft trajectories in the CR3BP. A feasible set of trajectories is recovered for each initial guess using a nested continuation procedure with multiple-shooting for differential corrections. Finally, the spacecraft trajectories are analyzed for collisions with dynamic obstacle corridors using a two-level, cell-based collision checking process. These methods are demonstrated by constructing an array of geometrically-distinct, collision-free transfers between periodic orbits in the Earth-Moon CR3BP. This research enables the construction of digestible summaries of potential future spacecraft motions as well as automatic and rapid generation of collision-free spacecraft trajectories.

Dedication

To my loving and supportive Mom (Wendy), Dad (Gregg), and brother (Brandon).

Thank you for giving me a world of opportunities.

"A man's heart plans his way,

But the LORD directs his steps."

Proverbs 16:9 NKJV

Acknowledgements

I am grateful to have had an incredible support system all these years. To my family: thank you for your immeasurable support over the last decade of higher education. This research would not have been possible without your consistent love and support. Whether near or far, you were always lifting me up. To each person who has been part of my life over the last 10 years: thank you for shaping me into the person I am today. In particular, I would like to thank the Garlow, Kirby, Baioumy, Huffer, Fiske, and Deuel families for standing alongside me in every season.

To my advisor, Dr. Natasha Bosanac: thank you for your time, dedication, and guidance throughout the last several years. You pushed me to think critically; construct innovative solutions; have confidence in my work; and grow into the researcher I am today. I also want to thank my committee members for their invaluable feedback. It is an honor to have had each of you on my committee. Furthermore, I would like to thank all past and present members of the Bosanac Group: Ian, Jack, Stefano, Thomas, Kristen, Giuliana, Sai, Cole, Max, Austin, and Miguel. Thank you for making my time at CU special and contributing to my professional and personal growth. I would like to give a special thank-you to my dear friend, Giuliana, for walking with me through this degree. The last four years would not have been nearly as much fun without you.

Finally, I want to thank the National Defense Science and Engineering Graduate (NDSEG) Fellowship Program for supporting this research via a fellowship award under contract FA9550-21-F-0003. Thank you, as well, to my NDSEG mentor, Dr. Tristan Griffith, for your feedback, time, mentoring, and excellent questions every time we discussed this work.

Contents

Chapter

1 Introduction **1**

1.1 Motivation 1

1.2 Previous Work 2

1.2.1 Spacecraft Trajectory Design in Multi-Body Systems 2

1.2.2 Motion Planning in Robotics 4

1.2.3 Motion Planning Applied to Spacecraft Trajectory Design 4

1.2.4 Clustering Applied to Spacecraft Trajectories 5

1.3 Dissertation Overview 6

1.3.1 Organization 6

1.3.2 Contributions 8

2 Dynamical Model **11**

2.1 Equations of Motion 11

2.2 Jacobi Constant 20

2.3 Equilibrium Points 21

2.4 Zero Velocity Surfaces 24

2.5 Periodic Orbits 25

3 Correcting Trajectories **27**

3.1 Corrections via Multiple-Shooting 27

3.1.1 Periodic Orbits . . . 28
3.1.2 Impulsive Maneuver-Enabled Trajectories . . . 32
3.2 Natural Parameter Continuation for Maneuver Reduction . . . 34

4 Clustering Techniques **37**
4.1 Overview of Clustering Algorithms . . . 37
4.2 Hierarchical Density-Based Spatial Clustering of Applications with Noise . . . 38

5 Collision-Free Path Planning **43**
5.1 Tree-Based Path Planners . . . 44
5.2 Graph Search Techniques . . . 46
5.3 Cell Decomposition . . . 48
5.4 Collision Checking Techniques . . . 48

6 Construct a Digestible Motion Type Summary **50**
6.1 Generating Input Data . . . 51
6.2 Step 1: Propagate Continuous Trajectories . . . 53
6.3 Step 2: Summarize Trajectories . . . 54
6.4 Step 3: Construct Initial Summary of Motion Types . . . 57
6.5 Step 4: Update Summary with Additional Samples . . . 60
6.6 Step 5: Append Summary with Finer Clusters . . . 64
6.7 Summary of Governing Parameters and Decisions . . . 66

7 Tree-Based Initial Guess Construction **68**
7.1 Phase 1: Grow a Forest of Trees . . . 69
7.1.1 Sampling Tree Roots . . . 69
7.1.2 Growing Trees . . . 72
7.2 Phase 2: Forest-Guided Initial Guess Construction . . . 77
7.2.1 Locating Connections Between Trees . . . 77

7.2.2 Identifying Connected Tree Sequences 78
7.2.3 Searching the Tree Sequences for Initial Guesses 83
7.2.4 Summary of Governing Parameters and Decisions 84
7.3 Assessing Geometric Similarity 85
7.4 Recovering a Continuous Path 88

8 Generate Collision-Free Trajectories **91**
8.1 Dynamic Obstacle Maps 92
8.2 Corridor Avoidance 93
8.3 Summary of Governing Parameters and Decisions 99

9 Application: Collision-Free Spacecraft Transfers in the Earth-Moon CR3BP **101**
9.1 Spacecraft Motion Summaries from Local Initial State Sets 101
9.1.1 Summarized Motions Near L_1 Halo Orbits 102
9.1.2 Summarized Motions Near a Distant Prograde Orbit 105
9.2 Planar Transfers Between L_1 and L_2 Lyapunov Orbits 106
9.2.1 Single-Energy-Level Transfers 106
9.2.2 Multi-Energy-Level Transfers 116
9.3 Spatial Transfers Between L_1 and L_2 Northern Halo Orbits 131

10 Concluding Remarks **151**
10.1 Recommendations for Future Work 153

Bibliography **156**

Tables

Table

2.1 Locations of equilibrium points in the Earth-Moon CR3BP with $\mu \approx 1.215058 \times 10^{-2}$. 24

6.1 Parameters and decisions governing the data-driven categorization process. 67

7.1 Parameters and decisions governing the hierarchical tree-based planning process. . . 86

8.1 Parameters and decisions governing the collision checking process. 100

9.1 Reference states for each selected scenario demonstrating the data-driven categorization process. 102

9.2 Orbit parameters and boundary conditions for each planar transfer design scenario. . 106

9.3 Summary of transfer characteristics connecting planar L_1 and L_2 Lyapunov orbits at $C_J = 3.1556$. 113

9.4 Summary of transfer characteristics connecting a planar L_1 Lyapunov orbit at $C_J = 3.1556$ and a planar L_2 Lyapunov orbit at $C_J = 3.1459$. 118

9.5 Summary of transfer characteristics connecting L_1 and L_2 northern halo orbits at $C_J = 3.1468$. 132

Figures

Figure

2.1 Conceptual depiction of the inertial frame $\hat{X}\hat{Y}\hat{Z}$. 12
2.2 Conceptual depiction of the rotating frame $\hat{x}\hat{y}\hat{z}$ relative to the inertial frame $\hat{X}\hat{Y}\hat{Z}$. . 17
2.3 Equilibrium point locations in the Earth-Moon CR3BP with respect to the two primaries: Earth (E) and Moon (M). 23
2.4 Zero velocity curves for Jacobi constants equal to a) 3.31, b) 3.18, c) 3.13, and d) 3.01 in the Earth-Moon CR3BP. 26

3.1 Conceptualization of the multiple-shooting corrections method using four arcs to discretize a periodic orbit. 28
3.2 Correcting a northern L_1 halo orbit in the Earth-Moon CR3BP using multiple-shooting. 31

4.1 Conceptualization of core distances between points a and b. 39
4.2 Conceptualization of clustering a small dataset with HDBSCAN. 41
4.3 Conceptualization of a noise point (gray) grouped into a cluster (orange) via border point association. 42

5.1 Conceptual RRT generation in a linear environment: a) randomly sample (gold) the environment and extend the tree rooted at the initial state (red) towards the random sample; b) repeatedly sample the environment and extend the tree from the nearest node until c) a path is recovered. 45

6.1 High-level overview of the data-driven categorization process. 51
6.2 A 200-member subset of planar trajectories propagated for 17.3 days from the sampled initial conditions in the Earth-Moon CR3BP. 54
6.3 a) Locations of local maximum curvature along a trajectory. b) The locations of p states spaced equally in arclength along the trajectory. 56
6.4 Overview of the trajectory summarization process in Step 2. 57
6.5 Overview of the process for generating an initial motion type summary in Step 3. . . 58
6.6 Initial summary of trajectories generated from a local point cloud of initial states along the selected L_1 Lyapunov orbit in the Earth-Moon CR3BP. 60
6.7 Velocity components of initial conditions a) after initial clustering and b) with 536 new samples (black circles) after inter-cluster sampling. 61
6.8 Conceptual depiction of a set of planar initial conditions uniformly sampled in a square grid and colored according to cluster assignment. 62
6.9 Conceptual depiction of inter-cluster sampling between cluster boundaries where black circles denote new samples. 63
6.10 Overview of the inter-cluster sampling process in Step 4. 64
6.11 Final summary of trajectories generated for up to 17.3 days from a local point cloud of initial states along the selected L_1 Lyapunov orbit. 65

7.1 Overview of the process for sampling tree roots in Phase 1. 70
7.2 Tree roots for building a forest to support construction of a planar path between L_1 and L_2 Lyapunov orbits at $C_J = 3.1556$: filled circles denote the positions of the roots whereas arrows denote the direction of the velocity vectors at each root. 71
7.3 Overview of the process for growing the i-th tree in a forest in Phase 1. 73
7.4 A subset of five trees, each depicted with a different color, from the forest grown to support planar path construction between L_1 and L_2 Lyapunov orbits at $C_J = 3.1556$ and satisfy the dynamics in the CR3BP. 76

7.5 Overview of the process for identifying connections between trees in Phase 2. 78

7.6 a) Two trees connected by five sets of nodes, shown in red. b) The five connections between the blue and black trees. 79

7.7 Conceptual depiction of a tree sequence graph. 79

7.8 An example of a seven-tree sequence returned from the k-smoothest paths search algorithm that connects the trees rooted at the boundary conditions. 83

7.9 A coarse, zero-revolution initial guess connecting boundary conditions on L_1 and L_2 Lyapunov orbits at $C_J = 3.1556$ with a cumulative velocity discontinuity equal to 687.90 m/s and estimated time of flight equal to 26.12 days. A heteroclinic connection at $C_J = 3.1520$ is provided in gray for comparison. 85

7.10 Four geometrically-distinct initial guesses connecting boundary conditions along L_1 and L_2 Lyapunov orbits at $C_J = 3.1556$. 88

7.11 The evolution of a zero-revolution transfer connecting boundary conditions on L_1 and L_2 Lyapunov orbits at $C_J = 3.1556$: initial guess (gray), geometry-focused solution (purple) with Δv equal to 36.88 m/s and TOF equal to 28.03 days, and energy-focused solution (blue) with Δv equal to 29.78 m/s and TOF equal to 28.20 days. 89

8.1 Overview of the two-level, cell-based corridor avoidance process for a single dynamic obstacle map. 94

8.2 A continuous spacecraft path (blue) that connects L_1 and L_2 Lyapunov orbits is analyzed for collisions with a dynamic obstacle corridor (copper) that originates near the vicinity of a distant prograde orbit. 96

8.3 Coarse collision maps 1 - 15 constructed for a dynamic obstacle corridor observed at $t_{obs} = 12.11$ days near a distant prograde orbit at $C_J = 3.1700$ in the Earth-Moon CR3BP. 97

8.4 Coarse collision maps 16 - 27 constructed for a dynamic obstacle corridor observed at $t_{obs} = 12.11$ days near a distant prograde orbit at $C_J = 3.1700$ in the Earth-Moon CR3BP. 98

8.5 Overlap in position and time between a dynamic obstacle corridor and spacecraft trajectory is identified for $t \in [12.11, 14.11]$ days. 99

8.6 An example of a fine collision checking map with a collision occurring at $t_{coll} = 13.29$ days. 100

9.1 Fifteen types of 17.3-day trajectories that begin near the selected reference state along a L_1 northern halo orbit at $C_J = 3.1486$ in the Earth-Moon CR3BP. 103

9.2 Three types of 17.3-day trajectories that begin near perilune along the selected L_1 near rectilinear halo orbit at $C_J = 2.9947$ in the Earth-Moon CR3BP. 104

9.3 Distinct types of 26.05-day trajectories beginning near a reference state between perilune and apolune along a DPO at $C_J = 3.1700$ in the Earth-Moon CR3BP. . . . 105

9.4 Initial guesses 1-20 with distinct geometries for spacecraft trajectories connecting boundary conditions along L_1 and L_2 Lyapunov orbits at $C_J = 3.1556$ in the Earth-Moon CR3BP. 107

9.5 Initial guesses 21-40 with distinct geometries for spacecraft trajectories connecting boundary conditions along L_1 and L_2 Lyapunov orbits at $C_J = 3.1556$ in the Earth-Moon CR3BP. 108

9.6 Initial guesses 41-60 with distinct geometries for spacecraft trajectories connecting boundary conditions along L_1 and L_2 Lyapunov orbits at $C_J = 3.1556$ in the Earth-Moon CR3BP. 109

9.7 Initial guesses 61-80 with distinct geometries for spacecraft trajectories connecting boundary conditions along L_1 and L_2 Lyapunov orbits at $C_J = 3.1556$ in the Earth-Moon CR3BP. 110

9.8 Initial guesses 81-100 with distinct geometries for spacecraft trajectories connecting boundary conditions along L_1 and L_2 Lyapunov orbits at $C_J = 3.1556$ in the Earth-Moon CR3BP. 111

9.9 The evolution of a single-revolution transfer connecting boundary conditions on L_1 and L_2 Lyapunov orbits at $C_J = 3.1556$: initial guess (gray), geometry-focused solution (purple) with Δv equal to 386.32 m/s and TOF equal to 25.66 days, and energy-focused solution (blue) with Δv equal to 5.85 m/s and TOF equal to 26.85 days. 113

9.10 The evolution of a geometrically-distinct single-revolution transfer connecting boundary conditions on L_1 and L_2 Lyapunov orbits at $C_J = 3.1556$: initial guess (gray), geometry-focused solution (purple) with Δv equal to 443.55 m/s and TOF equal to 42.86 days, and energy-focused solution (blue) with Δv equal to 6.77 m/s and TOF equal to 38.21 days. 114

9.11 The evolution of a geometrically-distinct transfer connecting boundary conditions on L_1 and L_2 Lyapunov orbits at $C_J = 3.1556$: initial guess (gray), geometry-focused solution (purple) with Δv equal to 542.50 m/s and TOF equal to 34.12 days, and energy-focused solution (blue) with Δv equal to 19.14 m/s and TOF equal to 30.34 days. 115

9.12 The evolution of a single-revolution transfer with high curvature that connects boundary conditions on L_1 and L_2 Lyapunov orbits at $C_J = 3.1556$: initial guess (gray), geometry-focused solution (purple) with Δv equal to 308.37 m/s and TOF equal to 28.50 days, and energy-focused solution (blue) with Δv equal to 125.94 m/s and TOF equal to 26.88 days. 116

9.13 Evolution of the Jacobi constant along the geometry- (G) and energy-focused (E) solutions for transfers connecting L_1 and L_2 Lyapunov orbits at $C_J = 3.1556$, marked with a dashed red line, in the Earth-Moon CR3BP. 117

9.14 The state vectors of the tree roots span the desired range of Jacobi constant values, $C_J = [3.1459, 3.1556]$. 117

9.15 Initial guesses 1-20 with distinct geometries for spacecraft trajectories connecting boundary conditions along an L_1 Lyapunov orbit at $C_J = 3.1556$ and L_2 Lyapunov orbit at $C_J = 3.1459$ in the Earth-Moon CR3BP. 119

9.16 Initial guesses 21-40 with distinct geometries for spacecraft trajectories connecting boundary conditions along an L_1 Lyapunov orbit at $C_J = 3.1556$ and L_2 Lyapunov orbit at $C_J = 3.1459$ in the Earth-Moon CR3BP. 120

9.17 Initial guesses 41-60 with distinct geometries for spacecraft trajectories connecting boundary conditions along an L_1 Lyapunov orbit at $C_J = 3.1556$ and L_2 Lyapunov orbit at $C_J = 3.1459$ in the Earth-Moon CR3BP. 121

9.18 Initial guesses 61-80 with distinct geometries for spacecraft trajectories connecting boundary conditions along an L_1 Lyapunov orbit at $C_J = 3.1556$ and L_2 Lyapunov orbit at $C_J = 3.1459$ in the Earth-Moon CR3BP. 122

9.19 Initial guesses 81-100 with distinct geometries for spacecraft trajectories connecting boundary conditions along an L_1 Lyapunov orbit at $C_J = 3.1556$ and L_2 Lyapunov orbit at $C_J = 3.1459$ in the Earth-Moon CR3BP. 123

9.20 The evolution of a zero-revolution transfer that connects boundary conditions on L_1 and L_2 Lyapunov orbits at different C_J: initial guess (gray), geometry-focused solution (purple) with Δv equal to 252.60 m/s and TOF equal to 23.08 days, and energy-focused solution (blue) with Δv equal to 14.04 m/s and TOF equal to 26.49 days. 124

9.21 The evolution of a single-revolution transfer with a high apolune that connects boundary conditions on L_1 and L_2 Lyapunov orbits at different C_J: initial guess (gray), geometry-focused solution (purple) with Δv equal to 312.98 m/s and TOF equal to 26.92 days, and energy-focused solution (blue) with Δv equal to 21.78 m/s and TOF equal to 28.02 days. 125

9.22 The evolution of a low-curvature, single-revolution transfer that connects boundary conditions on L_1 and L_2 Lyapunov orbits at different C_J: initial guess (gray), geometry-focused solution (purple) with Δv equal to 315.31 m/s and TOF equal to 26.81 days, and energy-focused solution (blue) with Δv equal to 24.66 m/s and TOF equal to 26.07 days. 126

9.23 The evolution of a geometrically-distinct transfer that connects boundary conditions on L_1 and L_2 Lyapunov orbits at different C_J: initial guess (gray), geometry-focused solution (purple) with Δv equal to 313.01 m/s and TOF equal to 33.12 days, and energy-focused solution (blue) with Δv equal to 150.62 m/s and TOF equal to 34.43 days. 127

9.24 Evolution of the Jacobi constant along the geometry- (G) and energy-focused (E) solutions for transfers connecting an L_1 Lyapunov orbit at $C_J = 3.1556$, marked with a dashed red line, and an L_2 Lyapunov orbit at $C_J = 3.1459$, marked with a dashed blue line, in the Earth-Moon CR3BP. 128

9.25 The geometry-focused spacecraft transfer recovered from initial guess 1 collides with an obstacle corridor at approximately $t_{coll} = 9.60$ days. 128

9.26 The geometry-focused spacecraft transfer recovered from initial guess 1 collides with an obstacle corridor at approximately $t_{coll} = 18.11$ days. 129

9.27 The geometry-focused spacecraft transfer recovered from initial guess 72 collides with an obstacle corridor at approximately $t_{coll} = 12.01$ days. 130

9.28 The geometry-focused spacecraft transfer recovered from initial guess 1 collides with an obstacle corridor at approximately $t_{coll} = 14.26$ days. 130

9.29 The geometry-focused spacecraft transfer recovered from initial guess 3 collides with an obstacle corridor at approximately $t_{coll} = 14.01$ days. 131

9.30 Initial guesses 1-4 for geometrically-distinct spacecraft trajectories connecting boundary conditions along L_1 and L_2 northern halo orbits at $C_J = 3.1486$ in the Earth-Moon CR3BP. 133

9.31 Initial guesses 5-8 for geometrically-distinct spacecraft trajectories connecting boundary conditions along L_1 and L_2 northern halo orbits at $C_J = 3.1486$ in the Earth-Moon CR3BP. 134

9.32 Initial guesses 9-12 for geometrically-distinct spacecraft trajectories connecting boundary conditions along L_1 and L_2 northern halo orbits at $C_J = 3.1486$ in the Earth-Moon CR3BP. 135

9.33 Initial guesses 13-16 for geometrically-distinct spacecraft trajectories connecting boundary conditions along L_1 and L_2 northern halo orbits at $C_J = 3.1486$ in the Earth-Moon CR3BP. 136

9.34 Initial guesses 17-20 for geometrically-distinct spacecraft trajectories connecting boundary conditions along L_1 and L_2 northern halo orbits at $C_J = 3.1486$ in the Earth-Moon CR3BP. 137

9.35 Initial guesses 21-24 for geometrically-distinct spacecraft trajectories connecting boundary conditions along L_1 and L_2 northern halo orbits at $C_J = 3.1486$ in the Earth-Moon CR3BP. 138

9.36 Initial guesses 25-28 for geometrically-distinct spacecraft trajectories connecting boundary conditions along L_1 and L_2 northern halo orbits at $C_J = 3.1486$ in the Earth-Moon CR3BP. 139

9.37 Initial guesses 29-32 for geometrically-distinct spacecraft trajectories connecting boundary conditions along L_1 and L_2 northern halo orbits at $C_J = 3.1486$ in the Earth-Moon CR3BP. 140

9.38 Initial guesses 33-36 for geometrically-distinct spacecraft trajectories connecting boundary conditions along L_1 and L_2 northern halo orbits at $C_J = 3.1486$ in the Earth-Moon CR3BP. 141

9.39 Initial guesses 37-40 for geometrically-distinct spacecraft trajectories connecting boundary conditions along L_1 and L_2 northern halo orbits at $C_J = 3.1486$ in the Earth-Moon CR3BP. 142

9.40 Initial guesses 41-44 for geometrically-distinct spacecraft trajectories connecting boundary conditions along L_1 and L_2 northern halo orbits at $C_J = 3.1486$ in the Earth-Moon CR3BP. 143

9.41 Initial guesses 45-48 for geometrically-distinct spacecraft trajectories connecting boundary conditions along L_1 and L_2 northern halo orbits at $C_J = 3.1486$ in the Earth-Moon CR3BP. 144

9.42 Initial guesses 49-50 for geometrically-distinct spacecraft trajectories connecting boundary conditions along L_1 and L_2 northern halo orbits at $C_J = 3.1486$ in the Earth-Moon CR3BP. 145

9.43 a) Spatial representation of initial guess 2. b) Spatial representations of the geometry-focused (purple) and energy-focused (blue) transfers. c) Initial guess 2 projected onto the xz-plane. d) Geometry- and energy-focused transfers projected onto the xz-plane. The geometry-focused transfer possesses a Δv equal to 692.91 m/s with a TOF equal to 22.46 days while the energy-focused solution has a Δv equal to 184.29 m/s and TOF equal to 24.67 days. 146

9.44 a) Spatial representation of initial guess 34. b) Spatial representations of the geometry-focused (purple) and energy-focused (blue) transfers. c) Initial guess 34 projected onto the xz-plane. d) Geometry- and energy-focused transfers projected onto the xz-plane. The geometry-focused path is associated with a Δv equal to 569.18 m/s and TOF equal to 28.76 days. The energy-focused transfer requires a total maneuver magnitude of 241.53 m/s and a flight time of 30.42 days. 147

9.45 a) Spatial representation of initial guess 16. b) Spatial representations of the geometry-focused (purple) and energy-focused (blue) transfers. c) Initial guess 16 projected onto the xz-plane. d) Geometry- and energy-focused transfers projected onto the xz-plane. The solution in purple has a total maneuver magnitude equal to 782.31 m/s with a TOF equal to 17.64 days while the path in blue possesses a total Δv equal to 382.33 m/s with a TOF equal to 24.82 days. 148

9.46 a) Spatial representation of initial guess 18. b) Spatial representations of the geometry-focused (purple) and energy-focused (blue) transfers. c) Initial guess 18 projected onto the xz-plane. d) Geometry- and energy-focused transfers projected onto the xz-plane. The geometry-focused spacecraft trajectory in purple possesses a higher Δv, equal to 891.93 m/s, than the energy-focused path whose Δv is equal to 188.31 m/s. The times of flight are 24.31 days and 25.31 days, respectively. 149

9.47 a) Spatial representation of initial guess 11. b) Spatial representations of the geometry-focused (purple) and energy-focused (blue) transfers. c) Initial guess 11 projected onto the xz-plane. d) Geometry- and energy-focused transfers projected onto the xz-plane. The geometry-focused spacecraft trajectory possesses a Δv equal to 585.16 m/s while the energy-focused has a Δv equal to 93.24 m/s. The flight times are 19.46 days and 27.11 days, respectively. 150

9.48 Evolution of the Jacobi constant along the geometry- (G) and energy-focused (E) solutions for transfers connecting L_1 and L_2 northern halo orbits at $C_J = 3.1486$, marked with a dashed red line, in the Earth-Moon CR3BP. 150

Chapter 1

Introduction

To support an expanded presence in cislunar space, agile spacecraft will need to maneuver and replan their paths in response to potential hazards. However, rapidly designing a complex trajectory between two boundary conditions in a chaotic multi-body gravitational environment can be a challenging, computationally intensive problem. Typically, initial guesses for the path and maneuvers are constructed and then corrected to potentially produce a continuous and feasible trajectory. This process not only requires repetition to produce a set of options, but also relies on a trajectory designer for both low-level analysis and high-level decision making. Furthermore, the need to avoid collisions with other moving objects increases the complexity of the design problem. The research presented in this dissertation addresses the challenges of path design and collision avoidance in chaotic dynamical systems through the development of a new approach to rapid, collision-free spacecraft trajectory design via adaptations of techniques from data mining and motion planning.

1.1 Motivation

In recent years, cislunar exploration has reignited with the promise of new and exciting opportunities for space operations. For example, NASA's recent Artemis campaign aims to explore the Moon for scientific discovery, technological advancement, and insight into living and working on bodies other than Earth [80]. Four Artemis missions are currently scheduled to support this endeavor. Artemis I launched in 2022 as an uncrewed flight test around the Moon [80]. Artemis

II and III are forecasted to launch in 2026 and 2027, respectively, with Artemis IV ensuing. The currently planned Artemis IV mission includes a lunar space station, known as Gateway, that will support human missions to Mars. To reduce the risks associated with the Gateway mission, NASA collaborated with Advanced Space, the Terran Orbital Corporation, Stellar Exploration Inc., and Rocket Lab to develop, operate, build, and launch the Cislunar Autonomous Positioning System Technology Operations and Navigation Experiment (CAPSTONE) spacecraft [81]. Launched in 2022, CAPSTONE verified the characteristics of a near rectilinear halo orbit and validated innovative cislunar autonomous navigation software [81].

As cislunar space becomes increasingly populated, its safety and sustainability must be considered. This will not only protect the future of the cislunar environment, but also space assets as they travel within or through the region to their destinations. Enabling safe travel in this regime will rely on developing a new approach to rapid trajectory design in cislunar space.

1.2 Previous Work

1.2.1 Spacecraft Trajectory Design in Multi-Body Systems

Spacecraft motion in the restricted problem has been studied for many years. In celestial mechanics, the connection of physical causes with observable reality began with Isaac Newton's theory of gravitation, which was published in 1687 [43]. Leonhard Euler built on Newton's work by using a rotating coordinate system to formulate the restricted problem in 1772 and recover its collinear equilibrium points [43]. In the same year, Joseph-Louis Lagrange proved the existence of the triangular equilibrium points and reduced the formulation of the problem from order 18 to order 7 [43]. The Jacobi constant, named after Carl Gustav Jacob Jacobi, was discovered in 1836 and led to the introduction of zero velocity surfaces and allowable regions of motion in 1838 by George Hill [43, 117]. Many researchers followed Hill, including Henri Poincaré who published a memoir on the three-body problem in 1890 [43]. Poincaré's work focused on the qualitative aspects of celestial mechanics, leading to the development of dynamical systems and chaos theory [43].

His first paper, published in 1884, further contributed to the field by providing periodic solutions to the three-body problem [43]. Study of this dynamical system did not end with Poincaré, but was followed with additional investigation by other researchers including George David Birkoff who advanced the qualitative dynamical methods applied to the restricted problem [43].

More recent studies of the restricted problem provided further foundation for the expansion of its solution space. For example, Victor Szebehely published a comprehensive book in 1967 on the study of motion in the circular restricted three-body problem (CR3BP), including investigations of its numerical solutions such as periodic orbits [117]. These fundamental solutions govern natural motion in the CR3BP and provide important insight into the existence of natural and low-cost transfers between periodic orbits within a multi-body system. Computation and analysis of periodic orbits for trajectory design in the restricted problem has been studied by a vast number of researchers. A few examples include Howell's [51] computation of libration point orbit families in the Sun-Earth and Earth-Moon systems; Grebow's [22] application of orbits for studying the lunar south pole; Vaquero [110] and Gupta's [60] studies of resonant orbits for trajectory design and cislunar accessibility in complex environments; Folta et al.'s [18] creation of an orbital catalog; Koh et al.'s [23] use of cell mapping to analyze orbits near Europa and Enceladus; Anderson et al.'s [90] exploration of search methods for locating unstable resonant orbits; and Lara and Russell's [61] investigation of the influence of the mass ratio in the restricted problem on periodic orbits.

Natural and low-cost transfer design between three-body orbits has also been studied by numerous researchers. For example, Lo and Ross studied the natural InterPlanetary Superhighway, a network of passageways that provide low-energy transport throughout the Solar System [62, 63]. Additionally, Davis et al. [52] leveraged invariant manifolds to design optimal transfers between unstable periodic orbits at different energy levels. Furthermore, Haapala and Howell [6] studied and cataloged heteroclinic, homoclinic, and low-cost transfers between lunar libration point orbits in the Earth-Moon CR3BP; Koon et al. [121] analyzed heteroclinic connections between periodic orbits and resonance transitions; and Parker et al. [49] investigated the use of invariant manifolds for designing low-energy trajectories and orbital chains. While this is not a comprehensive list of

the innovative trajectory design techniques used to construct natural and low-cost transfers, these techniques often involve a human-in-the-loop.

Typically, manual analysis is performed in a lower-fidelity model before transitioning to an ephemeris model to support trajectory design in multi-body systems [70]. The CR3BP is commonly used to approximate a spacecraft's motion near two larger bodies. General arcs and fundamental solutions in the CR3BP, such as periodic orbits and invariant manifolds, are manually examined and assembled with maneuver profiles to construct an initial guess. These initial guesses are transitioned into higher fidelity models and corrected [70]. Since a human-in-the-loop is typically involved in this process, trajectory design becomes time-consuming and complex in some scenarios, raising operational costs and complexity. The current state-of-the-art, combined with current and future goals of cislunar space exploration, demonstrate the need for rapid and automatic design of collision-free trajectories in multi-body systems that limit reliance on operators.

1.2.2 Motion Planning in Robotics

Motion planning, or path planning, techniques may be leveraged to generate collision-free paths for an object traveling through a cluttered environment. This concept has widely been used in the field of robotics, but has also been applied to a variety of other path planning problems including interactive virtual environments and game design [64], camera movement [24], and uncrewed aerial vehicles [96, 97]. A common collision-free path planning process consists of two steps: (1) creating an environment map including current and predicted states of the vehicle and hazards; and (2) searching the map for collision-free paths between initial and target locations [97].

1.2.3 Motion Planning Applied to Spacecraft Trajectory Design

Within the last few decades, the application of motion planning techniques for spacecraft trajectory design has become increasingly prominent. Rapid spacecraft trajectory design, in particular, has been one area of interest within the astrodynamics community where these techniques have been applied. For example, Bruchko and Bosanac used probabilistic roadmap and graph

search techniques for rapid trajectory design in multi-body systems [53]. Smith and Bosanac [112] as well as Miceli and Bosanac [32] employed motion primitives and graphs to summarize the solution space and rapidly design geometrically-distinct trajectories. Das-Stuart et al. combined graph search and reinforcement learning techniques to autonomously construct initial guesses from natural arcs within regions accessible to the spacecraft [5]. Additionally, the A* search algorithm was employed by Parrish to find near-optimal paths for continuous-thrust trajectory optimization [77].

Recently, tree-based planners have been applied to spacecraft motion planning. For example, Danielson and Kloeppel used a tree-based search algorithm combined with an invariant-set motion planner for safe, rapid planning of spacecraft attitudes [12]. Tree based-planners have also been used in safe, relative spacecraft motion planning by Starek et al. [40] and Deka and McMahon [107]. Together, these works have demonstrated the value of tree-based planners in spacecraft motion planning.

1.2.4 Clustering Applied to Spacecraft Trajectories

Clustering algorithms, which group similar data and separate dissimilar data from within a dataset, have previously been used to group solutions for a variety of astrodynamics applications [45]. Early examples include Hadjighasem et al. using spectral clustering for detecting Lagrangian vortices within nonlinear dynamical systems [8]; Nakhjiri and Villac using k-means clustering for locating bounded motions near a distant retrograde orbit on a Poincaré map [79]; and Foslien et al. using data mining to identify anomalous conditions on the International Space Station's gimbal system [54]. More recently, Bosanac as well as Bonasera and Bosanac have developed a distributed clustering framework to summarize the geometries of a wide array of trajectories in a multi-body system [71, 73, 99]. Smith and Bosanac also used clustering to summarize arcs along families of periodic orbits and hyperbolic invariant manifolds that supply building blocks for trajectory design in a multi-body system [111]. Finally, Miceli et al. used clustering to extract low lunar frozen orbits from trajectories generated in a high-fidelity lunar gravity model [34]. These examples have

demonstrated the value of clustering for automatically extracting information from datasets in astrodynamics.

1.3 Dissertation Overview

To address the challenges of spacecraft trajectory design and collision avoidance in chaotic dynamical systems, the goal of this research is to develop a novel approach for rapidly and automatically constructing collision-free spacecraft trajectories within a multi-body system by combining trajectory design techniques from astrodynamics with path planning and data mining methods. To support this proof of concept, the potential paths of a hazardous object are generated from a local initial state set and summarized. The resulting motion types serve as dynamic obstacles for collision avoidance. Then, a hierarchical tree-based planning method is used to construct initial guesses for spacecraft trajectories between two boundary conditions. Finally, the initial guesses are corrected and analyzed for potential collisions to recover collision-free, continuous spacecraft trajectories in the Earth-Moon CR3BP.

1.3.1 Organization

This dissertation is divided into the following chapters:

- Chapter 2: The dynamical model for the circular restricted three-body problem is described with its equations of motion derived. Fundamental concepts are also included in this chapter, including a description and derivation of the Jacobi constant, explanation of the Lagrange points, and definition of zero velocity surfaces. A brief discussion of periodic orbits closes the chapter.

- Chapter 3: The numerical methods used to compute periodic orbits as well as correct initial guesses are discussed in this chapter. These methods include multiple-shooting for corrections and natural parameter continuation.

- Chapter 4: This chapter presents a high-level overview of clustering as well as a description

of Hierarchical Density-Based Spatial Clustering of Applications with Noise, which is used in this dissertation.

- Chapter 5: Existing collision-free path planning techniques are described in this chapter. This includes a discussion of tree-based path planners, graph-search techniques, environment maps, and collision checking techniques found in robotics motion planning.

- Chapter 6: This chapter focuses on the technical approach used to generate and summarize spacecraft trajectories generated from a local initial state set. The steps of the clustering approach are detailed in this chapter and demonstrated for a local initial state set along a planar L_1 Lyapunov orbit in the Earth-Moon CR3BP. The resulting summary supplies a description of possible future motions of a potentially hazardous object via a set of dynamic obstacle corridors.

- Chapter 7: The hierarchical tree-based approach to rapid initial guess construction for spacecraft trajectories is discussed in this chapter. Then, initial guesses are corrected to recover continuous spacecraft trajectories. This process is demonstrated by designing a planar spacecraft trajectory connecting boundary conditions along L_1 and L_2 Lyapunov orbits in the Earth-Moon CR3BP.

- Chapter 8: The two-level, cell-based collision checking process is discussed in this chapter. This approach is demonstrated by analyzing the planar spacecraft trajectories generated in Chapter 7 for collisions with dynamic obstacle corridors of other observed spacecraft.

- Chapter 9: The tree-based initial guess construction process enables rapid generation of spacecraft trajectories in multi-body systems. This process is demonstrated for planar and spatial transfers at a single energy level, as well as planar transfers spanning multiple energy levels. To demonstrate the utility of the initial guesses, a subset are selected to recover continuous, collision-free paths.

- Chapter 10: The work completed in this dissertation is summarized in this chapter. Fur-

thermore, recommendations are presented for future investigations and applications of this new, tree-based framework for constructing collision-free spacecraft trajectories in a multi-body gravitational system.

1.3.2 Contributions

This research focuses on developing a proof of concept for collision-free spacecraft trajectory design in multi-body systems using tree-based path planning methods. The contributions of this research include:

(1) An automated method for summarizing predictions of object motion, as a function of geometry and time, resulting from local initial state sets along periodic orbits.

(2) A hierarchical tree-based approach for automatic and rapid design of an array of geometrically-diverse initial guesses for spacecraft trajectories.

(3) A two-level, cell-based approach to collision detection for identifying collision-free spacecraft trajectories in a multi-body system.

These contributions support collision-free spacecraft trajectory design in chaotic dynamical systems and may eventually enable the development of a safe and sustainable cislunar environment.

The automated method for summarizing object motion offers an original contribution to characterizing trajectories that emanate from a local region of phase space but exhibit diverse paths through a chaotic system. This challenge has been encountered in propagating uncertainty along spacecraft trajectories. Many uncertainty propagation analyses focus on trajectories that closely resemble a single nominal path. However, recent approaches have addressed calculating probability distributions that describe non-Gaussian variations around a reference path or approximating distinct modes governing the evolution of a diverse set of trajectories using Gaussian mixture models [118], radial basis functions [85], and the probabilistic admissible region [58]. In contrast, the method presented in this dissertation focuses on using clustering to group a diverse set of paths that emanate from a local region of phase space by their geometry. This approach can discover

geometrically distinct groups without assuming a function approximation form or incorporating trajectory characteristics while, at the same time, avoiding the propagation of additional parameters along the trajectories. Furthermore, unlike previous applications of clustering to spacecraft trajectories [34, 73, 99], the local proximity of the initial conditions of trajectories generated in this dissertation reduces both the geometric diversity across the dataset and separation between geometrically different trajectories. These characteristics present an interesting new application for using clustering to discover distinct motion types. In contrast to these works, this dissertation also uses a slightly different trajectory summarization and description approach as well as an additional resampling step. Supplying a digestible summary of object motion types may be useful for manual analysis or decision making, autonomous path planning, analyzing the impact of state uncertainty on spacecraft motion, and collision detection.

The second major contribution of this dissertation is a proof of concept for a novel tree-based approach to automatic initial guess construction for spacecraft trajectories in a complex, multi-body gravitational system. A challenge in automatic spacecraft trajectory design is defining a single objective function that promotes recovery of a wide array of geometrically-distinct paths. For instance, encoding an objective function may include reducing the distance traversed between the boundary conditions, specifying the number of revolutions about a primary body, and limiting discontinuity in velocity space while satisfying the dynamics of a specified dynamical model. In the field of robotics, sampling-based planning algorithms, such as probabilistic roadmaps and rapidly-exploring random trees (RRTs), have been adapted to solve motion queries in kinodynamic environments where kinematic and dynamic constraints must be considered [28, 89, 103]. Additionally, Bruchko and Bosanac [53] successfully applied sampling-based kinodynamic planning to initial guess construction of spacecraft trajectories via probabilistic roadmaps. This dissertation explores the use of another probabilistic planner, rapidly-exploring random trees, to efficiently extract a wide array of geometrically-distinct initial guesses without requiring solutions to boundary value problems. Rather, RRTs step forward or backward in time with randomized samples to rapidly explore a high-dimensional solution space. Furthermore, a forest of trees is leveraged in this work

for a hierarchical approach to automatic initial guess construction that overcomes the challenge of encoding an objective function, which is encountered when using only one or two goal-biased RRTs. Rapidly and autonomously constructing initial guesses for spacecraft trajectories with distinct itineraries may eventually be useful for reducing the burden on an operator or supporting onboard redesign of spacecraft trajectories.

The final contribution of this dissertation is a two-level approach to collision detection for identifying collision-free spacecraft trajectories in a multi-body system. This two-level approach reduces the computation time of collision checking while retaining sufficient accuracy to identify collisions without requiring parameterization of the dynamic obstacle corridors. Furthermore, the collision checking process supports rapid assessment of the generated trajectory options for collisions with other observed spacecraft. This collision-checking method is useful for understanding the effects of transfer departure times, potential future spacecraft motions, and transfer geometries on available spacecraft paths.

Chapter 2

Dynamical Model

The circular restricted three-body problem supplies an approximate dynamical model of a multi-body system. The CR3BP governs the motion of a body (P_3) of negligible mass under the gravitational influence of two massive bodies, also referred to as primaries [120]. The larger primary body is denoted as P_1 while the smaller primary body is denoted as P_2; both primaries are modeled as point masses with constant mass. Because the third body is assumed to possess a negligible mass compared to the masses of P_1 and P_2, it does not influence the orbits of the two primaries; therefore, the primaries are assumed to travel on conics about their barycenter [117]. One additional assumption is introduced to remove time-dependency: the two primaries are assumed to follow circular orbits. As a consequence of these assumptions, this dynamical model is a simplification of the more general three-body problem. Nonetheless, it is a useful tool that provides insight into the motion of a third body in a multi-body gravitational environment. This chapter was originally published in Spear's masters thesis [91]. Furthermore, the derivations presented in Sections 2.1 and 2.3 follow the derivations presented by Bosanac [72].

2.1 Equations of Motion

One approach to deriving the equations of motion begins with the definition and analysis of the potential function for the third body. First, each body P_i is located in an inertial coordinate frame, with axes $\hat{X}\hat{Y}\hat{Z}$, and an inertially-fixed origin, O. Note that $(\tilde{\ })$ denotes a dimensional quantity. In this frame, the state vectors of the three bodies are written as

$$\tilde{\bar{\bar{X}}}_1 = \left[\tilde{X}_1, \tilde{Y}_1, \tilde{Z}_1, \tilde{X}_1', \tilde{Y}_1', \tilde{Z}_1'\right] \tag{2.1}$$

$$\tilde{\bar{\bar{X}}}_2 = \left[\tilde{X}_2, \tilde{Y}_2, \tilde{Z}_2, \tilde{X}_2', \tilde{Y}_2', \tilde{Z}_2'\right] \tag{2.2}$$

$$\tilde{\bar{\bar{X}}}_3 = \left[\tilde{X}, \tilde{Y}, \tilde{Z}, \tilde{X}', \tilde{Y}', \tilde{Z}'\right] \tag{2.3}$$

noting that $()'$ indicates a time derivative with respect to an observer in the inertial frame. Using these definitions, the scalar potential of the third body due to the gravity of the other two point masses within this system is written as

$$\tilde{U}_3 = \frac{\tilde{G}\tilde{M}_1}{\tilde{R}_{13}} + \frac{\tilde{G}\tilde{M}_2}{\tilde{R}_{23}} \tag{2.4}$$

where $\tilde{G}$ is the dimensional, universal gravitational constant, $\tilde{R}_{13}$ is the distance between the third body and larger primary, and $\tilde{R}_{23}$ is the distance between the third body and smaller primary [117]. These distances are conceptually depicted in Figure 2.1 and equal to

$$\tilde{R}_{13} = \sqrt{\left(\tilde{X} - \tilde{X}_1\right)^2 + \left(\tilde{Y} - \tilde{Y}_1\right)^2 + \left(\tilde{Z} - \tilde{Z}_1\right)^2} \tag{2.5}$$

$$\tilde{R}_{23} = \sqrt{\left(\tilde{X} - \tilde{X}_2\right)^2 + \left(\tilde{Y} - \tilde{Y}_2\right)^2 + \left(\tilde{Z} - \tilde{Z}_2\right)^2} \tag{2.6}$$

Given the scalar potential function for the third body in Equation 2.4, the equations of motion are derived relative to an inertial observer by using the vector derivative of this scalar potential

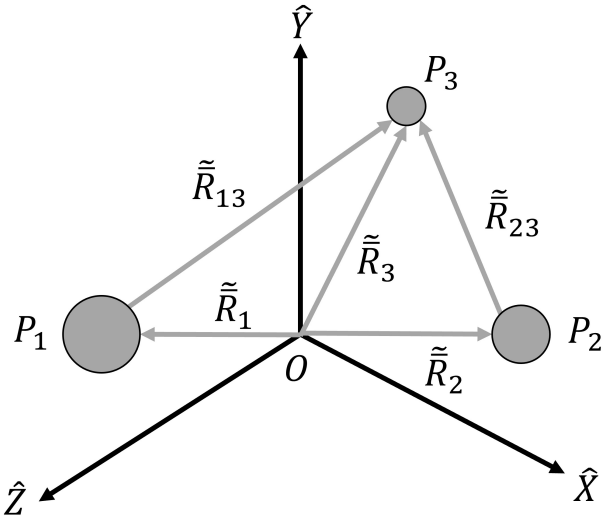

Figure 2.1: Conceptual depiction of the inertial frame $\hat{X}\hat{Y}\hat{Z}$.

function. Its partial derivatives are used to calculate the acceleration components of the potential function acting on the body via

$$\bar{F}_i = \bar{\nabla}_i \tilde{U}_i = \bar{\tilde{R}}_i'' \tag{2.7}$$

where the ∇ operator indicates, generally, the following for a three-dimensional vector:

$$\bar{\nabla}_3() = \frac{\partial}{\partial X_3}()\hat{X} + \frac{\partial}{\partial Y_3}()\hat{Y} + \frac{\partial}{\partial Z_3}()\hat{Z} \tag{2.8}$$

The vector derivative of the scalar potential function is then written as

$$\bar{\nabla}\tilde{U}_3 = \bar{\tilde{R}}_3'' = \tilde{X}''\hat{X} + \tilde{Y}''\hat{Y} + \tilde{Z}''\hat{Z} = \frac{\partial \tilde{U}_3}{\partial \tilde{X}}\hat{X} + \frac{\partial \tilde{U}_3}{\partial \tilde{Y}}\hat{Y} + \frac{\partial \tilde{U}_3}{\partial \tilde{Z}}\hat{Z} \tag{2.9}$$

As an example, the partial derivative of the scalar potential with respect to the dimensional X-coordinate is expanded to equal

$$\frac{\partial \tilde{U}_3}{\partial \tilde{X}} = \frac{\partial}{\partial \tilde{X}}\left(\frac{\tilde{G}\tilde{M}_1}{\tilde{R}_{13}} + \frac{\tilde{G}\tilde{M}_2}{\tilde{R}_{23}}\right) = -\frac{\tilde{G}\tilde{M}_1\left(\tilde{X} - \tilde{X}_1\right)}{\tilde{R}_{13}^3} - \frac{\tilde{G}\tilde{M}_2\left(\tilde{X} - \tilde{X}_2\right)}{\tilde{R}_{23}^3} \tag{2.10}$$

Similarly, the remaining partial derivatives are written as

$$\frac{\partial \tilde{U}_3}{\partial \tilde{Y}} = -\frac{\tilde{G}\tilde{M}_1\left(\tilde{Y} - \tilde{Y}_1\right)}{\tilde{R}_{13}^3} - \frac{\tilde{G}\tilde{M}_2\left(\tilde{Y} - \tilde{Y}_2\right)}{\tilde{R}_{23}^3} \tag{2.11}$$

$$\frac{\partial \tilde{U}_3}{\partial \tilde{Z}} = -\frac{\tilde{G}\tilde{M}_1\left(\tilde{Z} - \tilde{Z}_1\right)}{\tilde{R}_{13}^3} - \frac{\tilde{G}\tilde{M}_2\left(\tilde{Z} - \tilde{Z}_2\right)}{\tilde{R}_{23}^3} \tag{2.12}$$

When Equations 2.10 - 2.12 are substituted into Equation 2.7, the dimensional equations of motion in an inertial frame for the third body are equal to

$$\tilde{X}'' = -\frac{\tilde{G}\tilde{M}_1\left(\tilde{X} - \tilde{X}_1\right)}{\tilde{R}_{13}^3} - \frac{\tilde{G}\tilde{M}_2\left(\tilde{X} - \tilde{X}_2\right)}{\tilde{R}_{23}^3} \tag{2.13}$$

$$\tilde{Y}'' = -\frac{\tilde{G}\tilde{M}_1\left(\tilde{Y} - \tilde{Y}_1\right)}{\tilde{R}_{13}^3} - \frac{\tilde{G}\tilde{M}_2\left(\tilde{Y} - \tilde{Y}_2\right)}{\tilde{R}_{23}^3} \tag{2.14}$$

$$\tilde{Z}'' = -\frac{\tilde{G}\tilde{M}_1\left(\tilde{Z} - \tilde{Z}_1\right)}{\tilde{R}_{13}^3} - \frac{\tilde{G}\tilde{M}_2\left(\tilde{Z} - \tilde{Z}_2\right)}{\tilde{R}_{23}^3} \tag{2.15}$$

These three second-order, scalar differential equations for a general three body problem may be converted into six first-order, scalar differential equations and used for numerical integration. However, the positions of the primaries are required for every time step within integration. If similar

equations of motions are written for both primary bodies, there are 18 scalar, first-order differential equations (six for each of the three bodies). Because there are only 10 constants of motion for a three body system, this problem does not admit an analytical solution [35]. Furthermore, the dynamics are time dependent. Therefore, the assumptions introduced earlier in this chapter are used to simplify the model.

A normalization scheme is commonly used to nondimensionalize length, time, and mass quantities within the CR3BP. Dimensional and nondimensional quantities are related by characteristic quantities associated with length, mass, and time. Normalization is used because these parameters may have distinct orders of magnitude. Furthermore, normalization supports comparison between systems of similar relative masses. To normalize length quantities, the characteristic quantity l^* is defined using the assumed constant distance between the two primaries. Mathematically,

$$l^* = \tilde{D}_1 + \tilde{D}_2 \tag{2.16}$$

where $\tilde{D}_i$ is the dimensional distance of body P_i from the origin O, assumed to be located at the barycenter of the primary system. The characteristic quantity for mass, m^*, is used to normalize mass and is set equal to the combined mass of the system's primaries as

$$m^* = \tilde{M}_1 + \tilde{M}_2 \tag{2.17}$$

where $\tilde{M}_i$ is the dimensional mass of body P_i. This characteristic quantity is also used to define the mass ratio of the system, μ, as

$$\mu = M_2 = \frac{\tilde{M}_2}{m^*} \tag{2.18}$$

where M_2 is the nondimensional mass of the smaller primary. In the Earth-Moon system, $\mu \approx 1.215058\times10^{-2}$. The nondimensional mass of the first primary is also written in terms of the mass ratio or mass characteristic quantity as

$$M_1 = 1 - \mu = \frac{\tilde{M}_1}{m^*} \tag{2.19}$$

where M_1 is the nondimensional mass of the first primary. Finally, the characteristic time quantity

is set equal to

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

The gravitational constant, $\tilde{G}$, may also be converted to the following nondimensional parameter using the three characteristic quantities defined in Equations 2.16, 2.17, and 2.20:

$$G^* = \frac{\tilde{G}m^* (t^*)^2}{(l^*)^3} = 1 \tag{2.21}$$

In the Earth-Moon CR3BP, $l^* = 384{,}400$ km, $m^* = 6.047683 \times 10^{24}$ kg, and $t^* = 375{,}126.4166$ seconds [16, 117].

These characteristic quantities are used to write nondimensional equations of motion for the CR3BP in the inertial frame. As an example, the position components and distances between primaries are nondimensionalized as

$$X = \frac{\tilde{X}}{l^*} \qquad Y = \frac{\tilde{Y}}{l^*} \qquad Z = \frac{\tilde{Z}}{l^*} \tag{2.22}$$

$$R_{13} = \frac{\tilde{R}_{13}}{l^*} \qquad R_{23} = \frac{\tilde{R}_{23}}{l^*} \tag{2.23}$$

The nondimensionalized distances between the third body and each of the primaries are then described mathematically via the following expressions:

$$R_{13} = \sqrt{(X - X_1)^2 + (Y - Y_1)^2 + (Z - Z_1)^2} \tag{2.24}$$

$$R_{23} = \sqrt{(X - X_2)^2 + (Y - Y_2)^2 + (Z - Z_2)^2} \tag{2.25}$$

where the nondimensionalized state vectors are defined as

$$\bar{X}_1 = [X_1, Y_1, Z_1, X_1', Y_1', Z_1'] \tag{2.26}$$

$$\bar{X}_2 = [X_2, Y_2, Z_2, X_2', Y_2', Z_2'] \tag{2.27}$$

$$\bar{X}_3 = [X, Y, Z, X', Y', Z'] \tag{2.28}$$

With these definitions, the nondimensional equations of motion can be derived.

As an example of deriving the nondimensional equations of motion, the X-component of acceleration is derived starting from Equation 2.13. First, the aforementioned characteristic quantities are substituted into Equation 2.13 such that

$$\frac{l^*}{(t^*)^2}X'' = -\frac{\tilde{G}\,(1-\mu)\,m^* l^*\,(X-X_1)}{(l^*)^3\,R_{13}^3} - \frac{\tilde{G}\mu m^* l^*\,(X-X_2)}{(l^*)^3\,R_{23}^3} \tag{2.29}$$

and simplifies to

$$X'' = \frac{\tilde{G}\,(t^*)^2\,m^*}{(l^*)^3}\left[-\frac{(1-\mu)\,(X-X_1)}{R_{13}^3} - \frac{\mu\,(X-X_2)}{R_{23}^3}\right] \tag{2.30}$$

Rearranging and substituting the nondimensional gravitational constant, G^*, in Equation 2.21 for the dimensional gravitational constant, $\tilde{G}$, gives

$$X'' = \frac{\frac{(l^*)^3}{m^*(t^*)^2}\,(t^*)^2\,m^*}{(l^*)^3}\left[-\frac{(1-\mu)\,(X-X_1)}{R_{13}^3} - \frac{\mu\,(X-X_2)}{R_{23}^3}\right] \tag{2.31}$$

This expression may be further simplified to its final nondimensional form, provided in Equation 2.32. Substituting the nondimensional state components into the second-order, scalar differential equations that govern the motion of the CR3BP in the inertial frame allows for the expression of all three equations of motion in nondimensional form as

$$X'' = -\frac{(1-\mu)\,(X-X_1)}{R_{13}^3} - \frac{\mu\,(X-X_2)}{R_{23}^3} \tag{2.32}$$

$$Y'' = -\frac{(1-\mu)\,(Y-Y_1)}{R_{13}^3} - \frac{\mu\,(Y-Y_2)}{R_{23}^3} \tag{2.33}$$

$$Z'' = -\frac{(1-\mu)\,(Z-Z_1)}{R_{13}^3} - \frac{\mu\,(Z-Z_2)}{R_{23}^3} \tag{2.34}$$

However, these equations are still time-dependent with respect to the locations of the primaries.

The assumption that the primaries travel on circular orbits about their mutual barycenter is used to eliminate the time-dependency of the equations of motion. To incorporate this assumption, the X- and Y-components of their positions are equal to

$$X_1 = -\mu\cos(t) \qquad Y_1 = -\mu\sin(t) \tag{2.35}$$

$$X_2 = (1-\mu)\cos(t) \qquad Y_2 = (1-\mu)\sin(t) \tag{2.36}$$

where t is time. Since the primaries lie in the xy-plane and rotate at a constant rate of one in the normalized system, the Z-coordinates have a one-to-one mapping: $Z = z$ [69]. Substituting these coordinates for the primaries written in the inertial frame and relative to the barycenter, the equations of motion become

$$X'' = -\frac{(1-\mu)\,(X+\mu\cos(t))}{R_{13}^3} - \frac{\mu\,(X-(1-\mu)\cos(t))}{R_{23}^3} \tag{2.37}$$

$$Y'' = -\frac{(1-\mu)\,(Y+\mu\sin(t))}{R_{13}^3} - \frac{\mu\,(Y-(1-\mu)\sin(t))}{R_{23}^3} \tag{2.38}$$

$$Z'' = -\frac{(1-\mu)\,(Z)}{R_{13}^3} - \frac{\mu\,(Z)}{R_{23}^3} \tag{2.39}$$

While the location of the primaries, P_1 and P_2, are no longer required for all time, time dependency complicates the analysis of the solution space [117].

To eliminate a dependency on time, the equations of motion are formulated in a frame that rotates with the primaries. Figure 2.2 conceptually displays the configuration of the CR3BP in the rotating frame. In this frame, P_1 and P_2 are always fixed along the $\hat{x}$ axis, the $\hat{z}$ axis is aligned with the orbital angular momentum vector of the primary system, and $\hat{y}$ lies in the orbital plane of the two primaries, completing the right-handed triad [117]. Here, B denotes the barycenter while $\hat{x}\hat{y}\hat{z}$ denotes the axes of the rotating frame. The angle between the vector from the barycenter to P_2 and $\hat{X}$ axis is defined by η.

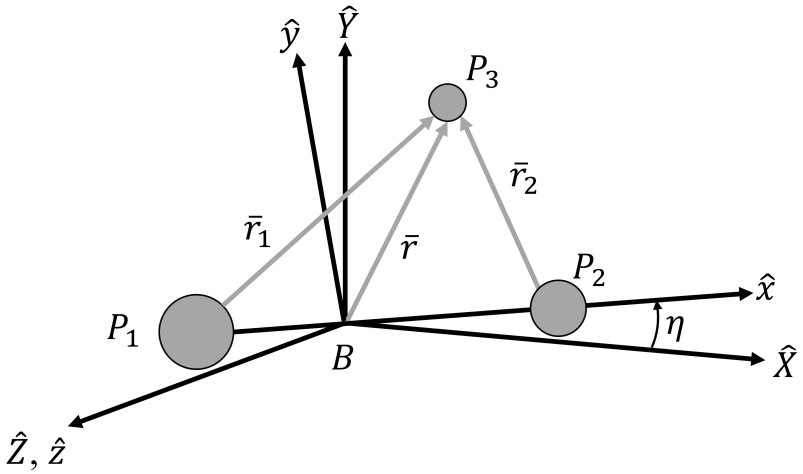

Figure 2.2: Conceptual depiction of the rotating frame $\hat{x}\hat{y}\hat{z}$ relative to the inertial frame $\hat{X}\hat{Y}\hat{Z}$.

Given this definition of the rotating frame, a direction cosine matrix is formulated for transformations from the inertial to rotating frames. The relationship between the position coordinates in these two frames is expressed in matrix form as

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos(\eta) & \sin(\eta) & 0 \\ -\sin(\eta) & \cos(\eta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{2.40}$$

The angle, η, between the vector from the barycenter to P_2 and $\hat{X}$-axis may be expressed in terms of time since $\eta = nt$. However, the mean motion, n, is unity and $\eta = t$. This transformation is used to rewrite the equations of motion in the rotating frame in combination with the relationship and locations of the primaries in the rotating frame. The state vector of P_3 with respect to the barycenter is written as

$$\bar{x} = [x, y, z, \dot{x}, \dot{y}, \dot{z}]^T \tag{2.41}$$

Applying the rotation in Equation 2.40 with $\eta = t$, P_1 is located at $(-\mu, 0, 0)$ while P_2 is located at $(1-\mu, 0, 0)$ in the rotating frame. Therefore, the position vectors of the third body with respect to the primaries are expressed mathematically as

$$\bar{r}_1 = (x+\mu)\,\hat{x} + y\hat{y} + z\hat{z} \tag{2.42}$$

$$\bar{r}_2 = (x-1+\mu)\,\hat{x} + y\hat{y} + z\hat{z} \tag{2.43}$$

With this information, Equations 2.32 - 2.34 can be transformed into the rotating frame.

To rewrite the equations of motion in the rotating frame, the acceleration of the third body must be rewritten in the inertial frame using an observer that is also located in that frame. To begin this derivation, note that the acceleration may be written as

$$\bar{r}'' = {}^{I}\frac{d\bar{v}}{dt} = {}^{I}\frac{d}{dt}\left(\frac{d\bar{r}}{dt}\right) \tag{2.44}$$

where $(\,)''$ indicates an observer fixed in the inertial frame. First,

$${}^{I}\frac{d\bar{r}}{dt} = {}^{R}\frac{d\bar{r}}{dt} + {}^{I}\bar{\omega}^{R} \times \bar{r} = \dot{\hat{r}} + (\hat{z}) \times (x\hat{x} + y\hat{y} + z\hat{z}) \tag{2.45}$$

where $\dot{()}$ is a derivative with respect to an observer in the rotating frame and $\bar{\omega}$ is the angular velocity vector. This expression simplifies using transport theorem [38] to

$$ {}^{I}\frac{d\bar{r}}{dt} = (\dot{x} - y)\,\hat{x} + (\dot{y} + x)\,\hat{y} + \dot{z}\hat{z} \tag{2.46} $$

Taking the derivative of this expression gives

$$ {}^{I}\frac{d}{dt}\left(\frac{d\bar{r}}{dt}\right) = (\dot{x} - y)\,\hat{x} + (\dot{y} + x)\,\hat{y} + \dot{z}\hat{z} + \hat{z} \times [(\dot{x} - y)\,\hat{x} + (\dot{y} + x)\,\hat{y} + \dot{z}\hat{z}] \tag{2.47} $$

which results in

$$ \bar{r}'' = {}^{I}\frac{d}{dt}\left(\frac{d\bar{r}}{dt}\right) = (\ddot{x} - 2\dot{y} - x)\,\hat{x} + (\ddot{y} + 2\dot{x} - y)\,\hat{y} + \ddot{z}\hat{z} \tag{2.48} $$

This expression represents the acceleration of the third body in the rotating frame.

The acceleration of the third body with respect to an observer in the rotating frame and an expression for the locations of the bodies are used to formulate the nondimensional equations of motion in the rotating frame. These equations of motion are equal to

$$ \ddot{x} = 2\dot{y} + x - \frac{(1-\mu)\,(x+\mu)}{r_1^3} - \frac{\mu\,(x-1+\mu)}{r_2^3} \tag{2.49} $$

$$ \ddot{y} = -2\dot{x} + y - \frac{(1-\mu)\,y}{r_1^3} - \frac{\mu y}{r_2^3} \tag{2.50} $$

$$ \ddot{z} = -\frac{(1-\mu)\,z}{r_1^3} - \frac{\mu z}{r_2^3} \tag{2.51} $$

where the nondimensional distances of the third body, P_3, with respect to each of the primaries, P_1 and P_2, are equal to

$$ r_1 = \sqrt{(x+\mu)^2 + y^2 + z^2} \tag{2.52} $$

$$ r_2 = \sqrt{(x-1+\mu)^2 + y^2 + z^2} \tag{2.53} $$

Solutions to the CR3BP may be generated via numerical integration given an initial state and after these three second-order differential equations have been converted into six first-order differential equations.

2.2 Jacobi Constant

An integral of motion, known as the Jacobi constant, exists with this formulation of the CR3BP in the rotating frame. This constant is inversely related to a spacecraft's energy: a decrease in the Jacobi constant indicates there is an increase in energy [35, 120]. As a result, this integral is often used to determine regions of motion for the third body [16]. One approach to calculating this quantity leverages the pseudo-potential function [117], which is defined as

$$U^* = \frac{1}{2}\left(x^2 + y^2\right) + \frac{1-\mu}{r_1} + \frac{\mu}{r_2} \tag{2.54}$$

where the first term is an additional term as a result of the pseudo-potential written in the rotating frame (compared to the inertial definition in Equation 2.4); the second term represents the gravitational interaction between the third body and larger primary, P_1; and the third term represents the gravitational interaction between the third body and smaller primary, P_2. This pseudo-potential function is selected to ensure that the equations of motion may be written using the partial derivatives of the pseudo-potential function. Mathematically,

$$\ddot{x} = 2\dot{y} + \frac{\partial U^*}{\partial x} \qquad \ddot{y} = -2\dot{x} + \frac{\partial U^*}{\partial y} \qquad \ddot{z} = \frac{\partial U^*}{\partial z} \tag{2.55}$$

This relationship, along with the acceleration and velocity vectors in the rotating frame, is used to derive the integral of motion that will define the Jacobi constant. Taking the dot product of the acceleration and velocity vectors and substituting the equations of motion produces the following expression:

$$\ddot{x}\dot{x} + \ddot{y}\dot{y} + \ddot{z}\dot{z} = \left(2\dot{y} + \frac{\partial U^*}{\partial x}\right)\dot{x} + \left(-2\dot{x} + \frac{\partial U^*}{\partial y}\right)\dot{y} + \left(\frac{\partial U^*}{\partial z}\right)\dot{z} = \frac{\partial U^*}{\partial x}\dot{x} + \frac{\partial U^*}{\partial y}\dot{y} + \frac{\partial U^*}{\partial z}\dot{z} \tag{2.56}$$

In the CR3BP, the pseudo-potential function does not explicitly depend on time; therefore, its partial derivative with respect to time is zero. Then, the total time derivative of the pseudo-potential function is written as

$$\frac{dU^*}{dt} = \frac{\partial U^*}{\partial x}\frac{dx}{dt} + \frac{\partial U^*}{\partial y}\frac{dy}{dt} + \frac{\partial U^*}{\partial z}\frac{dz}{dt} \tag{2.57}$$

where t is time. Substituting this expression into Equation 2.56 produces the following:

$$\frac{dU^*}{dt} = \ddot{x}\dot{x} + \ddot{y}\dot{y} + \ddot{z}\dot{z} = \frac{1}{2}\frac{d}{dt}\left(\dot{x}^2 + \dot{y}^2 + \dot{z}^2\right) \tag{2.58}$$

Recall that $\dot{x}^2 + \dot{y}^2 + \dot{z}^2 = v^2$ where v is the velocity magnitude. Taking the integral of this expression is mathematically written as

$$\int \frac{1}{2}\frac{d}{dt}\left(v^2\right)\, dt = \int \frac{dU^*}{dt}\, dt \tag{2.59}$$

which produces the following expression:

$$v^2 = 2U^* + C_J \tag{2.60}$$

where C_J is the Jacobi constant. Rearranging this expression, the Jacobi constant is defined as

$$C_J = 2U^* - v^2 = \left(x^2 + y^2\right) + \frac{2\left(1-\mu\right)}{r_1} + \frac{2\mu}{r_2} - \dot{x}^2 - \dot{y}^2 - \dot{z}^2 \tag{2.61}$$

This quantity remains constant along any natural solution in the CR3BP and may be used to identify regions of motion within a system at a given energy level [116].

2.3 Equilibrium Points

Equilibrium points, also referred to as Lagrange or libration points, are stationary solutions to the CR3BP in the rotating frame. At these points, both velocity and acceleration are zero; therefore, the third body could be placed at one of these points and remain at that location indefinitely when its motion is modeled with the dynamics of the CR3BP [35]. There are five equilibrium points in this dynamical model which lie in the plane of the primaries: L_1, L_2, L_3, L_4, and L_5.

The locations of the equilibrium points are determined by examining their relationship with the equations of motion. Since velocity and acceleration are equal to zero, the following is true:

$$\frac{\partial U^*}{\partial x} = x - \frac{(1-\mu)(x+\mu)}{r_1^3} - \frac{\mu(x-1+\mu)}{r_2^3} = 0 \tag{2.62}$$

$$\frac{\partial U^*}{\partial y} = y - \frac{(1-\mu)y}{r_1^3} - \frac{\mu y}{r_2^3} = 0 \tag{2.63}$$

$$\frac{\partial U^*}{\partial z} = \frac{-(1-\mu)z}{r_1^3} - \frac{\mu z}{r_2^3} = 0 \tag{2.64}$$

With all equilibrium points in the plane of the primaries, $z = 0$ which satisfies Equation 2.64. Equation 2.63 has two possible solutions.

First, the collinear equilibrium points correspond to locations where $y = 0$. Then, Equation 2.62 has three conditions under which it can be satisfied:

$$(x+\mu) > 0 \qquad (x-1+\mu) > 0 \tag{2.65}$$

$$(x+\mu) < 0 \qquad (x-1+\mu) < 0 \tag{2.66}$$

$$(x+\mu) > 0 \qquad (x-1+\mu) < 0 \tag{2.67}$$

Numerical root finding, such as with Newton's method, can be used with an initial guess to locate the x-coordinate of the collinear equilibrium points, which lie at $y = 0$. Three regions are defined to select these initial guesses:

(1) To the left of P_1: $x < -\mu$

(2) Between P_1 and P_2: $-\mu < x < 1-\mu$

(3) To the right of P_2: $x > 1-\mu$

As depicted in Figure 2.3, the location of L_3 lies in region (1), L_2 in region (3), and L_1 in region (2).

The second possible solution to Equation 2.63 is when $y \neq 0$, which gives the locations of the triangular equilibrium points. These solutions lie at the vertices of two equilateral triangles formed with the two primaries, P_1 and P_2. Three significant locations that aid in the recovery of the locations of these equilibrium points are the following:

(i) $r_1 = 1, r_2 = 0$ when $(x, y, z) = (1-\mu, 0, 0)$

(ii) $r_1 = 0, r_2 = 1$ when $(x, y, z) = (-\mu, 0, 0)$

(iii) $r_1 = \mu, r_2 = 1-\mu$ when $(x, y, z) = (0, 0, 0)$

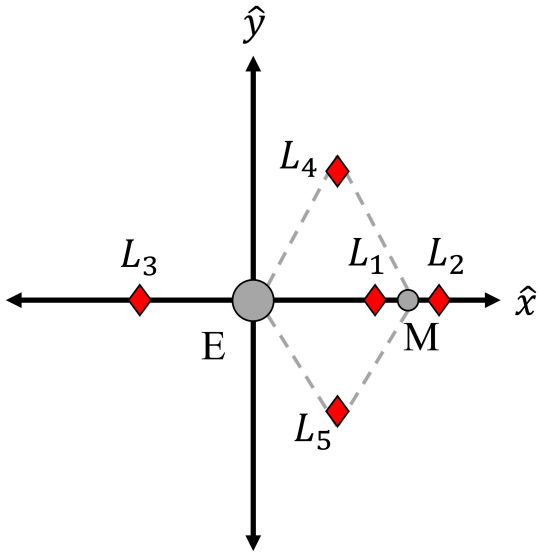

Figure 2.3: Equilibrium point locations in the Earth-Moon CR3BP with respect to the two primaries: Earth (E) and Moon (M).

Furthermore, the distance of P_3 from the origin can be written as a function of r_1 and r_2 as

$$l^2 = ar_1^2 + br_2^2 + q \tag{2.68}$$

Solving for a, b, and q in Equation 2.68 allows the pseudo-potential function to be rewritten in terms of r_1 and r_2 and used recover the locations of the triangular equilibrium points. Using locations (i) and (ii) along with Equation 2.68, q may be expressed via the following two equations:

$$q = \mu^2 - b \qquad q = (1-\mu^2) - a \tag{2.69}$$

These equations are combined to provide an expression for a, written as

$$a = 1 - 2\mu + b \tag{2.70}$$

Then, these expressions, location (iii), and Equation 2.68 are leveraged to find $b = \mu$. Substituting the expressions for a, b, and q into Equation 2.68 gives the following:

$$x^2 + y^2 = (1-\mu)r_1^2 + \mu r_2^2 + \mu(\mu - 1) \tag{2.71}$$

The pseudo-potential function from Equation 2.54 may then be rewritten as

$$U^* = \frac{1}{2}((1-\mu)r_1^2 + \mu r_2^2 + \mu(\mu-1)) + \frac{1-\mu}{r_1} - \frac{\mu}{r_2} \tag{2.72}$$

Recall that the problem of locating equilibrium points in U^* can be posed as

$$\frac{\partial U^*}{\partial x} = \frac{\partial U^*}{\partial y} = 0 \tag{2.73}$$

which is equivalent to

$$\frac{\partial U^*}{\partial r_1} = \frac{\partial U^*}{\partial r_2} = 0 \tag{2.74}$$

The partials of Equation 2.72 with respect to r_1 and r_2 are satisfied when $r_1 = r_2 = 1$. Given these values for the distances of the third body with respect to each of these primaries, Equation 2.52 - 2.53 are solved to find the $x-$ and $y-$coordinates of the triangular equilibrium points: $x_{L_{4,5}} = \frac{1}{2} - \mu$, $y_{L_{4,5}} = \pm\frac{\sqrt{3}}{2}$. The locations of the five equilibrium points in the Earth-Moon CR3BP are provided in Table 2.1. The set of equilibrium points is one of four fundamental solutions to the CR3BP. Analyzing motion near them can lead to the recovery of other solutions [117].

2.4 Zero Velocity Surfaces

Zero velocity surfaces (ZVS) separate regions of allowable and forbidden motion for the third body at a specific value of the Jacobi constant. A ZVS is composed of infinite states that possess a zero velocity. These surfaces are calculated by solving Equation 2.75 given a specific value of the Jacobi constant with zero velocity. Mathematically, this is expressed as

$$C_J = \left(x^2 + y^2\right) + \frac{2\left(1-\mu\right)}{r_1} + \frac{2\mu}{r_2} \tag{2.75}$$

where the components of the state vectors $(x, y, z, 0, 0, 0)$ are solutions to this expression that define the three-dimensional surface.

Table 2.1: Locations of equilibrium points in the Earth-Moon CR3BP with $\mu \approx 1.215058\times10^{-2}$.

Equilibrium Point	**Location: [x, y, z]**
L_1	[0.836915132366261, 0, 0]
L_2	[1.15568216029081, 0, 0]
L_3	[-1.00506264525194, 0, 0]
L_4	[0.487849415730458, 0.866025403784439, 0]
L_5	[0.487849415730458, -0.866025403784439, 0]

When the three-dimensional surface intersects with the plane of the primaries, the intersection forms a two-dimensional curve known as a zero velocity curve (ZVC). These curves, also known as Hill Curves, were first introduced by George Hill and provide insight into the allowable motion of a third body between regions around the two primaries in the CR3BP [117]. In Figure 2.4, four sets of ZVCs are generated in the Earth-Moon CR3BP to demonstrate flow between regions at different energy levels. The black lines denote the ZVCs while the dark-gray and white sections show regions of forbidden and allowable motion, respectively. The value of C_J in Figure 2.4a) is greater than the Jacobi constant of L_1, meaning motion can only remain in the vicinity of one of the primaries; a third body cannot travel between regions. The value of C_J is decreased to below the Jacobi constant at L_1 and above the Jacobi constant at L_2 in Figure 2.4b), permitting travel around the Moon, through the L_1 gateway, and in the vicinity of the Earth. In Figure 2.4c), the value of C_J is below that of L_2 to allow motion through the L_2 gateway to the exterior region of the system. Finally, an example of a ZVC at a Jacobi constant that permits third body travel around the Moon, Earth, and through the L_1, L_2, and L_3 gateways is displayed in Figure 2.4d).

2.5 Periodic Orbits

Periodic orbits are a fundamental solution to the CR3BP that are periodic after a specific time interval in the rotating frame. There are an infinite number of these trajectories existing in continuous families throughout the CR3BP at a specific value of μ [109]. A periodic orbit is defined by its orbital period and a non-unique fixed state, $\bar{x}_{PO}$; across these continuous families the initial conditions and orbit periods vary. Well-known periodic orbit families in the CR3BP include libration point and primary-centered orbits [21, 66].

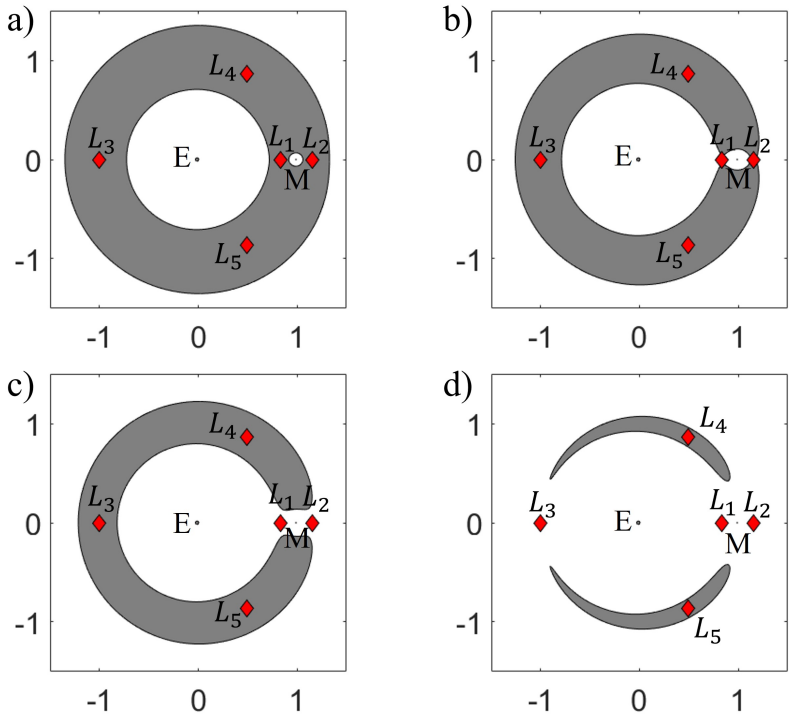

Figure 2.4: Zero velocity curves for Jacobi constants equal to a) 3.31, b) 3.18, c) 3.13, and d) 3.01 in the Earth-Moon CR3BP.

Chapter 3

Correcting Trajectories

Numerical methods are essential to generating solutions in the CR3BP. Corrections schemes may be used to correct initial guesses to recover continuous paths. Then, continuation methods can be applied to these solutions to recover additional members of a set of trajectories. This chapter provides an overview of the numerical methods used to generate periodic orbits, correct initial guesses for maneuver-enabled paths connecting boundary conditions located along periodic orbits in the Earth-Moon CR3BP, and recover a feasible set of trajectories with desired characteristics. The information presented in Section 3.1.1 was originally published in Spear's master's thesis [91].

3.1 Corrections via Multiple-Shooting

A differential corrections method known as multiple-shooting is used to recover continuous trajectories such as periodic orbits and transfers. Multiple-shooting schemes discretize a trajectory into multiple segments, or arcs, then adjust each segment until the path is continuous to within a user-defined tolerance and satisfies any additional constraints [19]. These schemes are typically formulated using free-variable and constraint vectors that are iteratively adjusted to achieve continuity. In this work, the free-variable vector contains descriptions of each arc via its initial state and the corresponding integration time. The initial state along each arc is numerically integrated in the CR3BP using a Runge-Kutta 8th/9th order method in C++ via the GNU Scientific Library [1] with relative and absolute tolerances of 10^{-13} and 10^{-14}, respectively. The constraint vector is constructed according to the desired application. In each iteration, the error in the constraints is

calculated and driven to zero simultaneously to within a user-defined tolerance by adjusting the free variables.

Figure 3.1 provides an example of using four discontinuous arcs to recover a periodic orbit. The initial and final states of each arc are denoted with white and black circles, respectively. These arcs are then corrected to recover a nearby solution, i.e. the periodic orbit plotted in blue. Generally, the free-variable vector is comprised of parameters that may be updated to meet the constraints of the problem [120]. For the maneuver-enabled paths constructed in this dissertation, the constraint vector enforces position continuity between neighboring arcs while full state continuity is enforced for periodic orbit corrections. The free-variable and constraint vector formulations differ for each of these scenarios and are discussed in detail in the following subsections.

3.1.1 Periodic Orbits

A periodic orbit is computed in this dissertation using multiple-shooting. This multiple-shooting formulation is the same as that described by Bosanac [70] for recovering continuous periodic orbits and assumes that the arcs possess equivalent integration times. The free-variable vector,

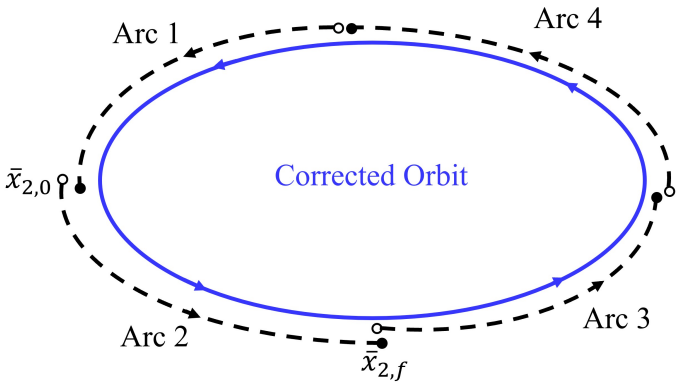

Figure 3.1: Conceptualization of the multiple-shooting corrections method using four arcs to discretize a periodic orbit.

$\bar{V}$, is a $(6n_{arc}+1)\times 1$ dimensional vector constructed as follows:

$$\bar{V}=\begin{bmatrix}\bar{x}_{1,0}\\ \bar{x}_{2,0}\\ \bar{x}_{3,0}\\ \vdots\\ \bar{x}_{n_{arc},0}\\ \Delta t\end{bmatrix} \tag{3.1}$$

where $\bar{x}_{i,0}$ is the initial state of arc i and Δt is the common integration time along each of the n_{arc} arcs. The constraint vector, $\bar{F}(\bar{V})$, provided in Equation 3.2 then reflects state continuity constraints between neighboring arcs. The final constraint, which is between the end of arc n_{arc} and the start of arc 1, only provides continuity in five of the six state elements as the full state constraint is not required for periodicity when the remainder of the trajectory is continuous [70]. This $(6n_{arc}\times 1)$ constraint vector is written as

$$\bar{F}\left(\bar{V}\right)=\begin{bmatrix}\bar{x}_{1,f}-\bar{x}_{2,0}\\ \bar{x}_{2,f}-\bar{x}_{3,0}\\ \vdots\\ \bar{x}_{n-1,f}-\bar{x}_{n_{arc},0}\\ x_{n_{arc},f}-x_{1,0}\\ y_{n_{arc},f}-y_{1,0}\\ z_{n_{arc},f}-z_{1,0}\\ \dot{x}_{n_{arc},f}-\dot{x}_{1,0}\\ \dot{z}_{n_{arc},f}-\dot{z}_{1,0}\\ y_{1,0}\end{bmatrix} \tag{3.2}$$

where $\bar{x}_{i,f}$ is the state vector at the end of arc i. The final row in the constraint vector constrains the y-position component of the state vector at the beginning of the first arc to lie at $y=0$. This constraint vector may also be augmented with a Jacobi constant constraint to calculate an orbit of

a similar Jacobi constant given a nearby orbit. This constraint may take the form of the absolute value of the difference between a desired Jacobi constant and an orbit's current Jacobi constant.

The Jacobian matrix is used to update the free-variable vector at every iteration of the corrections method. The derivative of the constraint vector in Equation 3.2 with respect to the free-variable vector in Equation 3.1 is used to determine the Jacobian, $\boldsymbol{D\bar{F}(\bar{V})}$, as

$$\boldsymbol{D\bar{F}}\left(\bar{\boldsymbol{V}}\right) = \frac{\partial \bar{F}\left(\bar{V}\right)}{\partial \bar{V}} = \begin{bmatrix} \boldsymbol{\Phi}_1 & -\mathbf{I}_{6\times6} & \mathbf{0}_{6\times6} & \dots & \mathbf{0}_{6\times6} & \dot{\bar{x}}_{1,f} \\ \mathbf{0}_{6\times6} & \boldsymbol{\Phi}_2 & -\mathbf{I}_{6\times6} & \dots & \mathbf{0}_{6\times6} & \dot{\bar{x}}_{2,f} \\ \vdots & & & & \vdots & \\ -\mathbf{H}_{5\times6} & \mathbf{0}_{5\times6} & \mathbf{0}_{5\times6} & \dots & \boldsymbol{\phi}_{n_{arc}} & \dot{\bar{X}}_{n_{arc},f} \\ [0\ 1\ 0\ 0\ 0\ 0] & \mathbf{0}_{1,6} & \mathbf{0}_{1,6} & \dots & \mathbf{0}_{1,6} & 0 \end{bmatrix} \tag{3.3}$$

where $\boldsymbol{\Phi}_i$ is the state transition matrix (STM) of arc i, $\mathbf{I}_{6\times6}$ is a 6×6 identity matrix, $\dot{\bar{x}}_{i,f}$ is the derivative of the state vector at the end of arc i, $\mathbf{0}_{i\times j}$ is the $i \times j$ zero matrix, $\mathbf{H}_{5\times6}$ is the identity matrix with the fifth row removed, $\boldsymbol{\phi}_{n_{arc}}$ is the state transition matrix of arc i with the fifth row removed, and $\dot{\bar{X}}_{n_{arc},f}$ is the derivative of the final state vector of arc n_{arc} with the fifth row removed. The STM linearly maps the variation in a trajectory relative to a reference trajectory from an initial time, t_0, to a final time, t_f [120]. Mathematically,

$$\boldsymbol{\Phi}\left(t_f, t_0\right) = \frac{\partial \bar{x}\left(t_f\right)}{\partial \bar{x}\left(t_0\right)} = \begin{bmatrix} \frac{\partial x_f}{\partial x_0} & \frac{\partial x_f}{\partial y_0} & \frac{\partial x_f}{\partial z_0} & \frac{\partial x_f}{\partial \dot{x}_0} & \frac{\partial x_f}{\partial \dot{y}_0} & \frac{\partial x_f}{\partial \dot{z}_0} \\ \frac{\partial y_f}{\partial x_0} & \frac{\partial y_f}{\partial y_0} & \frac{\partial y_f}{\partial z_0} & \frac{\partial y_f}{\partial \dot{x}_0} & \frac{\partial y_f}{\partial \dot{y}_0} & \frac{\partial y_f}{\partial \dot{z}_0} \\ \frac{\partial z_f}{\partial x_0} & \frac{\partial z_f}{\partial y_0} & \frac{\partial z_f}{\partial z_0} & \frac{\partial z_f}{\partial \dot{x}_0} & \frac{\partial z_f}{\partial \dot{y}_0} & \frac{\partial z_f}{\partial \dot{z}_0} \\ \frac{\partial \dot{x}_f}{\partial x_0} & \frac{\partial \dot{x}_f}{\partial y_0} & \frac{\partial \dot{x}_f}{\partial z_0} & \frac{\partial \dot{x}_f}{\partial \dot{x}_0} & \frac{\partial \dot{x}_f}{\partial \dot{y}_0} & \frac{\partial \dot{x}_f}{\partial \dot{z}_0} \\ \frac{\partial \dot{y}_f}{\partial x_0} & \frac{\partial \dot{y}_f}{\partial y_0} & \frac{\partial \dot{y}_f}{\partial z_0} & \frac{\partial \dot{y}_f}{\partial \dot{x}_0} & \frac{\partial \dot{y}_f}{\partial \dot{y}_0} & \frac{\partial \dot{y}_f}{\partial \dot{z}_0} \\ \frac{\partial \dot{z}_f}{\partial x_0} & \frac{\partial \dot{z}_f}{\partial y_0} & \frac{\partial \dot{z}_f}{\partial z_0} & \frac{\partial \dot{z}_f}{\partial \dot{x}_0} & \frac{\partial \dot{z}_f}{\partial \dot{y}_0} & \frac{\partial \dot{z}_f}{\partial \dot{z}_0} \end{bmatrix} \tag{3.4}$$

where the subscript f indicates a variable calculated at time t_f and a subscripted *0* indicates a variable evaluated at time t_0.

An iterative method is then used to compute a periodic orbit that has been discretized into n_{arc} arcs from an initial guess for the free variable vector, $\bar{V}_0$. Using the free-variable vector at

iteration i, i.e., $\bar{V}_i$, the free-variable vector at the next iteration, $\bar{V}_{i+1}$, is calculated using Newton's method. For a rectangular Jacobian matrix, the update equation is the following:

$$\bar{V}_{i+1} = \bar{V}_i - \boldsymbol{D\bar{F}}\left(\bar{\boldsymbol{V}}_{\boldsymbol{i}}\right)^T \left[\boldsymbol{D\bar{F}}\left(\bar{\boldsymbol{V}}_{\boldsymbol{i}}\right)\boldsymbol{D\bar{F}}\left(\bar{\boldsymbol{V}}_{\boldsymbol{i}}\right)^T\right]^{-1} \bar{F}\left(\bar{V}_i\right) \tag{3.5}$$

where $\boldsymbol{D\bar{F}}\left(\bar{\boldsymbol{V}}_{\boldsymbol{i}}\right)$ is the Jacobian evaluated at the current iteration and $\bar{F}\left(\bar{V}_i\right)$ is the constraint vector evaluated at the current iteration. This vector is updated until the magnitude of the constraint vector is equal to zero within a tolerance of 10^{-13}. Note that Newton's method is an iterative scheme to approximate a root of a function by taking its first derivative; therefore, the initial guess must be close to the function's true root for this method to converge quadratically [27].

This multiple-shooting method is demonstrated on a northern L_1 halo orbit at $C_J = 3.1567$ in the Earth-Moon CR3BP. Figure 3.2 displays the initial guess for the periodic orbit with a black, dashed line while the corrected orbit is shown with a solid blue line. The directions of motion are denoted with arrows, L_1 is displayed using a red diamond, and the Moon is depicted as a gray sphere scaled to possess a radius of 1,738.0 km [16]. The initial guess is constructed by perturbing a periodic orbit near the bifurcation with the L_1 Lyapunov orbit family. Then, the initial guess is discretized into four arcs and a solution is recovered after five

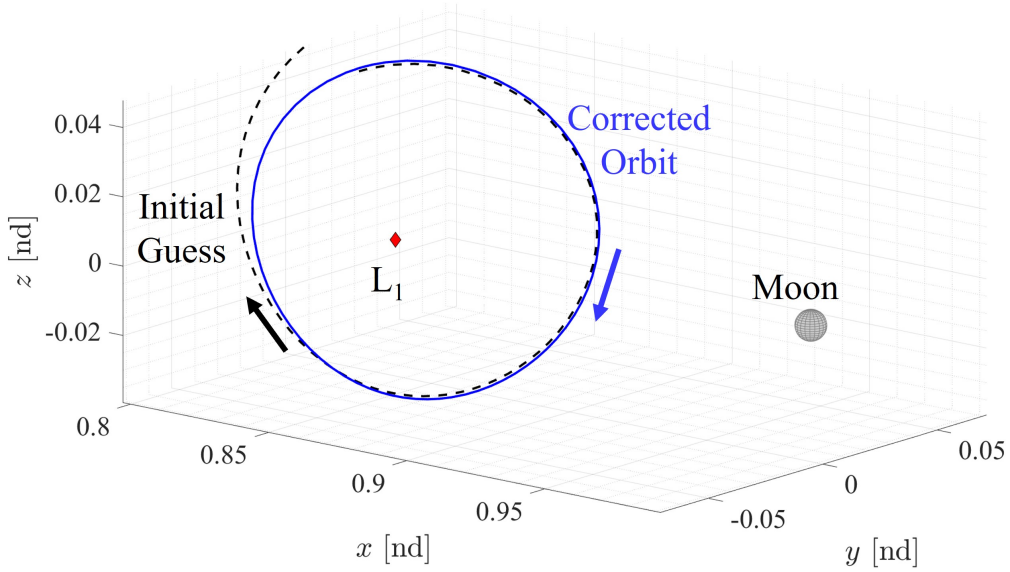

Figure 3.2: Correcting a northern L_1 halo orbit in the Earth-Moon CR3BP using multiple-shooting.

iterations of the shooting scheme. The corrected orbit is described by a fixed state $\bar{x}_{PO} = [0.823725874812321, 0, 0.0464081352286445, 0, 0.155839702089999, 0]^T$ nondimensional units and a period equal to 11.97 days. The periodicity of the corrected orbit is confirmed by calculating the difference between the initial and final states: $\bar{x}_{1,0} - \bar{x}_{4,f}$. If the error in each state component is less than 10^{-13}, it is considered periodic, which is true for this L_1 orbit.

3.1.2 Impulsive Maneuver-Enabled Trajectories

Multiple-shooting is also used to correct impulsive maneuver-enabled transfers within the Earth-Moon CR3BP. The initial guess for the transfer is discretized into a total of n_{arc} arcs and corrected. This leads to the following definition of the $(7n_{arc} \times 1)$ free-variable vector:

$$\bar{V} = \begin{bmatrix} \bar{x}_{1,0} \\ \Delta t_1 \\ \bar{x}_{2,0} \\ \Delta t_2 \\ \vdots \\ \bar{x}_{n_{arc},0} \\ \Delta t_{n_{arc}} \end{bmatrix} \tag{3.6}$$

where $\bar{x}_{i,0}$ is the initial state of arc i and Δt_i is the corresponding integration time for arc i.

The constraint vector is formulated to enforce state continuity between arcs as well as any additional constraints. In this work, n_{man} impulsive maneuvers are allowed between the end of specified arcs and the start of the neighboring arcs, requiring that only position continuity is enforced between these neighboring arcs. Additionally, full state continuity is required at the boundary conditions: $\bar{x}_{des1}$ at t_0 at the start of the transfer and $\bar{x}_{des2}$ at t_f at the end of the

transfer. Thus, the $(6(n_{arc} - n_{man}) + 3n_{man}) \times 1)$ constraint vector is defined as

$$\bar{F}\left(\bar{V}\right) = \begin{bmatrix} \bar{x}_{1,0} - \bar{x}_{des1} \\ \bar{x}_{1,f} - \bar{x}_{2,0} \\ \vdots \\ \bar{r}_{i-1,f} - \bar{r}_{i,0} \\ \vdots \\ \bar{x}_{n_{arc}-1,f} - \bar{x}_{n_{arc},0} \\ \bar{x}_{n_{arc},f} - \bar{x}_{des2} \end{bmatrix} \tag{3.7}$$

where $\bar{r}_{i,0}$ is the initial position of arc i and $\bar{r}_{i,f}$ is the final position of arc i. This constraint vector may also be augmented with additional constraints as needed.

A Jacobian is required to update the free-variable vector. In this formulation, the Jacobian is a $((6(n_{arc} - n_{man}) + 3n_{man}) \times 7n_{arc})$ matrix equal to

$$\boldsymbol{D\bar{F}}\left(\bar{\boldsymbol{V}}\right) = \begin{bmatrix} \mathbf{I}_{6\times6} & \mathbf{0}_{6\times1} & \dots & \dots & \dots & \dots & \dots & \mathbf{0}_{6\times6} & \mathbf{0}_{6\times1} \\ \boldsymbol{\Phi}_1 & \dot{\bar{x}}_{1,f} & -\mathbf{I}_{6\times6} & \dots & \dots & \dots & \dots & \mathbf{0}_{6\times6} & \mathbf{0}_{6\times1} \\ \mathbf{0}_{6\times6} & \mathbf{0}_{6\times1} & \boldsymbol{\Phi}_2 & \dot{\bar{x}}_{2,f} & -\mathbf{I}_{6\times6} & \dots & \dots & \mathbf{0}_{6\times6} & \mathbf{0}_{6\times1} \\ \vdots & & & & \vdots & & & & \vdots \\ \mathbf{0}_{3\times6} & \mathbf{0}_{3\times1} & \dots & \boldsymbol{\Phi}_{i,r} & \dot{\bar{r}}_{i,f} & -\mathbf{I}_{3\times6} & \dots & \mathbf{0}_{3\times6} & \mathbf{0}_{3\times1} \\ \vdots & & & & \vdots & & & & \vdots \\ \mathbf{0}_{6\times6} & \mathbf{0}_{6\times1} & \dots & \dots & \dots & \boldsymbol{\Phi}_{n_{arc}-1} & \dot{\bar{x}}_{n_{arc}-1,f} & -\mathbf{I}_{6\times6} & \mathbf{0}_{6\times1} \\ \mathbf{0}_{6\times6} & \mathbf{0}_{6\times1} & \dots & \dots & \dots & \dots & \dots & \boldsymbol{\Phi}_{n_{arc}} & \dot{\bar{x}}_{n_{arc},f} \end{bmatrix} \tag{3.8}$$

where $\boldsymbol{\Phi}_{i,r}$ is a 3×6 matrix from the portion of the STM covering only the partials of $\bar{x}_f$ with respect to position components and $\dot{\bar{r}}_f$ is the derivative of the position vector at the end of the arc. The update equation for this formulation of multiple-shooting is the same as Equation 3.5. With this formulation of multiple-shooting, an initial guess for a maneuver-enabled path connecting boundary conditions along periodic orbits is corrected to produce a nearby continuous path, if one exists.

3.2 Natural Parameter Continuation for Maneuver Reduction

Natural parameter continuation enables computation of additional members of a set of trajectories, such as periodic orbits or transfers, once a single member has been recovered [27]. Nearby members can be located by stepping along a specified natural parameter such as the orbital period, an initial state component, or the Jacobi constant. In this dissertation, natural parameter continuation is used to reduce the maneuver requirements of a transfer connecting periodic orbits while retaining the geometry of the initial guess.

A procedure to reduce the maneuvers along each generated transfer is modeled after the optimization-based approach developed by Smith and Bosanac to recover a trajectory that preserves the geometry of an initial guess while reducing the maneuver requirements [112]. Smith and Bosanac [112] presented a multi-objective cost function that is defined using two terms: 1) the difference between the position vector at specified states along the current guess for a path and a reference path and 2) the sum of squares of the magnitude of impulsive maneuvers. Although the second term does not minimize the total Δv, it supports better convergence towards a "minimum energy" solution [78]. This multi-objective cost function is defined as [112]

$$J = w_{geo}((\bar{r}_{curr} - \bar{r}_{ref})^T(\bar{r}_{curr} - \bar{r}_{ref})) + w_{man}\left(\sum_{j=1}^{q} \Delta v_j^2\right) \tag{3.9}$$

where w_{geo} is the weight of the term that prioritizes reducing the geometric difference between the current guess and the reference trajectory, w_{man} is the weight of the term that prioritizes reducing the maneuver requirements, $\bar{r}_{curr}$ is a $3n_s \times 1$ vector containing the selected n_s states along the trajectory at the current iteration, $\bar{r}_{ref}$ is a $3n_s \times 1$ vector containing the selected n_s states along a reference path, and Δv_j is the jth maneuver along the transfer. At a single combination of w_{man} and w_{geo}, Smith and Bosanac computed a solution using local, constrained optimization [112]. Then, continuation was used to gradually adjust the weights of this multi-objective cost function from $[w_{geo}, w_{man}] = [0.95, 0.05]$ to $[w_{geo}, w_{man}] = [0.05, 0.95]$ and solve the constrained optimization problem at each step.

In this dissertation, natural parameter continuation is used to reduce the maneuver require-

ments of a spacecraft transfer between two periodic orbits while retaining the geometry of the initial guess [27]. This continuation scheme is formulated using multiple-shooting augmented with a multi-objective constraint. Accordingly, the free-variable vector is defined using Equation 3.6. Then, the constraint vector from Equation 3.7 is augmented with the multi-objective cost function from Equation 3.9, formulated as a constraint, to produce the following vector:

$$\bar{F}_{aug}\left(\bar{V}\right) = \begin{bmatrix} \bar{F}\left(\bar{V}\right) \\ J - J_{des} \end{bmatrix} \tag{3.10}$$

where J_{des} is the desired value of J, given a specified combination of w_{geo} and w_{man}. When evaluating J, the reference trajectory is defined as the first trajectory corrected from the initial guess. The Jacobian in Equation 3.8 is augmented with an additional row reflecting the partial derivative of the augmented row of the constraint vector. This row in the Jacobian is written as

$$\begin{bmatrix} \bar{A}_1 & \bar{B}_2 & \dots & \dots & \dots & \dots & \dots & \bar{B}_{n_{arc}-1} & \bar{A}_{n_{arc}} \end{bmatrix} \tag{3.11}$$

where

$$\bar{A}_1 = \bar{A}_{n_{arc}} = 2w_{geo}\left(\bar{r}_{curr} - \bar{r}_{ref}\right) \tag{3.12}$$

Then, for all $j \in [2, n_{arc} - 1]$,

$$\bar{B}_j = 2w_{man} \begin{bmatrix} \left(\Delta v_{\dot{x},j+1}\right)\Phi_{4,1} + \left(\Delta v_{\dot{y},j+1}\right)\Phi_{5,1} + \left(\Delta v_{\dot{z},j+1}\right)\Phi_{6,1} \\ \left(\Delta v_{\dot{x},j+1}\right)\Phi_{4,2} + \left(\Delta v_{\dot{y},j+1}\right)\Phi_{5,2} + \left(\Delta v_{\dot{z},j+1}\right)\Phi_{6,2} \\ \left(\Delta v_{\dot{x},j+1}\right)\Phi_{4,3} + \left(\Delta v_{\dot{y},j+1}\right)\Phi_{5,3} + \left(\Delta v_{\dot{z},j+1}\right)\Phi_{6,3} \\ -\Delta v_{\dot{x},j-1} + \left(\Delta v_{\dot{x},j+1}\right)\Phi_{4,4} + \left(\Delta v_{\dot{y},j+1}\right)\Phi_{5,4} + \left(\Delta v_{\dot{z},j+1}\right)\Phi_{6,4} \\ -\Delta v_{\dot{y},j-1} + \left(\Delta v_{\dot{x},j+1}\right)\Phi_{4,5} + \left(\Delta v_{\dot{y},j+1}\right)\Phi_{5,5} + \left(\Delta v_{\dot{z},j+1}\right)\Phi_{6,5} \\ -\Delta v_{\dot{z},j-1} + \left(\Delta v_{\dot{x},j+1}\right)\Phi_{4,6} + \left(\Delta v_{\dot{y},j+1}\right)\Phi_{5,6} + \left(\Delta v_{\dot{z},j+1}\right)\Phi_{6,6} \\ \Delta v_{\dot{x},j+1}(\dot{x}) + \Delta v_{\dot{y},j+1}(\dot{y}) + \Delta v_{\dot{z},j+1}(\dot{z}) \end{bmatrix}^T + \begin{bmatrix} \bar{A}_j \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}^T \tag{3.13}$$

where, when a maneuver is present,

$$\Delta v_{\dot{x},j+1} = \dot{x}_{j,f} - \dot{x}_{j+1,0} \qquad \Delta v_{\dot{x},j-1} = \dot{x}_{j-1,f} - \dot{x}_{j,0} \tag{3.14}$$

$$\Delta v_{\dot{y},j+1} = \dot{y}_{j,f} - \dot{y}_{j+1,0} \qquad \Delta v_{\dot{y},j-1} = \dot{y}_{j-1,f} - \dot{y}_{j,0} \tag{3.15}$$

$$\Delta v_{\dot{z},j+1} = \dot{z}_{j,f} - \dot{z}_{j+1,0} \qquad \Delta v_{\dot{z},j-1} = \dot{z}_{j-1,f} - \dot{z}_{j,0} \tag{3.16}$$

The update equation remains the same as Equation 3.5.

The continuation procedure is formulated in two, nested layers. The inner layer gradually lowers the value of J_{des} for a specified combination of w_{geo} and w_{man}, solving the augmented multiple-shooting problem at each step. This inner continuation layer focuses on lowering the value of the multi-objective cost function defined in Equation 3.9; this step could be replaced by a suitable local, constrained optimization procedure in future work. The value of J_{des} is gradually reduced at each continuation step by using an adaptive step size, ΔJ_{des}. Initially, this step size is equal to

$$\Delta J_{des} = 10^{(\lfloor \log_{10}(J_{des,est}) \rfloor - 1)} \tag{3.17}$$

where J_0 is the initial value of J for a specified combination of w_{geo} and w_{man}. If the multiple shooting scheme returns a solution, J_{des} is decreased by ΔJ_{des}; otherwise, J_{des} is increased by ΔJ_{des} and the step size is reduced by half. Continuation proceeds until the step size is two orders of magnitude lower than its initial value, i.e., until $\Delta J_{des} \leq 10^{(\lfloor \log_{10}(J_0) \rfloor - 2)}$. Then, the outer layer gradually adjusts the weights in the multi-objective cost function defined in Equation 3.9 from $[w_{geo}, w_{man}] = [0.95, 0.05]$ to $[w_{geo}, w_{man}] = [0, 1]$ by increments of 0.05. Through this approach, the multi-objective cost function is gradually adjusted from prioritizing geometric resemblance to the initial guess to prioritizing reducing the sum of squares of the maneuver magnitudes.

Two trajectories in the feasible set of solutions generated during continuation are used for analysis in this dissertation. The first solution is labeled as geometry-focused, prioritizing its geometric resemblance to the initial guess, and is equal to the trajectory with the lowest value of J for $[w_{geo}, w_{man}] = [0.95, 0.05]$. The second trajectory is labeled as an energy-focused solution and is equal to the trajectory with the lowest value of J for $[w_{geo}, w_{man}] = [0, 1]$ and prioritizes reducing the sum of squares of the maneuver magnitudes. The use of the word "focused", as opposed to "optimal", in each label is consistent with the use of continuation, rather than optimization.

Chapter 4

Clustering Techniques

Sets of nonlinear trajectories may be challenging to manually categorize in the absence of an analytical solution or require automated analysis when reducing the dependency on a human-in-the-loop. In these cases, data mining techniques may be useful for automatically extracting patterns and other information [45]. One technique, clustering, enables extraction of a set of clusters that group similar data and separate dissimilar data [45]. These clusters can supply an automated and digestible summary of a larger dataset that aids subsequent analysis. This chapter presents an overview of the clustering techniques used in this dissertation. Note: This chapter expands on the information presented in an article by Spear and Bosanac in the Journal of Guidance, Control, and Dynamics by AIAA [92].

4.1 Overview of Clustering Algorithms

Clustering is a data mining technique used to group similar members of a dataset and separate dissimilar members [11]. Consider Q members of a dataset to be clustered. Each member is described by an S-dimensional feature vector, $\bar{f}_i$ for $i \in [1, Q]$, that is tailored to the desired application. Then, each pair of members are compared by calculating the distance between their feature vectors using a selected distance measure [45]. These distances are used by a clustering algorithm, along with any governing parameters, to construct groups of sufficiently similar members of the dataset. These clusters can be valuable for manual analysis or decision-making as they offer a digestible summary of the dataset.

Various clustering algorithms exist, with the most common approaches classified as partitioning, model-based, hierarchical, or density-based methods [45]. Partitioning algorithms directly divide the data points into k groups where k is a user-defined parameter [45]. Alternatively, model-based clustering schemes cluster the data to match a model that is typically a statistical distribution [11, 45]. Hierarchical methods use a dendrogram representation of possible clusters before grouping [45]. Density-based methods identify clusters as dense regions separated by sparse regions within the dataset [45]. Selection of a suitable algorithm is influenced by the characteristics of the data and the application; this decision, along with the values of any governing parameters, may also influence the recovered clusters.

Trajectory clustering, specifically, focuses on discovering patterns or groups of possible motions for a moving object [11]. There are various approaches to summarizing the trajectories to be clustered, including 1) using a time series representation, which may offer a high-fidelity representation of a trajectory but suffer from the curse of dimensionality; or 2) constructing a model of the trajectory, which may reduce the dimension of the description but suffer from inaccuracy when applied to a wide variety of paths [11, 114]. These finite-dimensional trajectory descriptions are then input to a selected clustering algorithm to extract groups of sufficiently similar paths.

4.2 Hierarchical Density-Based Spatial Clustering of Applications with Noise

This dissertation uses Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN), a clustering algorithm developed by Campello et al. as a hierarchical extension of the well-known Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm [88]. By combining density-based and hierarchical clustering concepts, HDBSCAN supports the identification of clusters as dense groupings of data in the selected S-dimensional feature space while also allowing clusters to possess distinct shapes and densities [87, 88]. Furthermore, this approach does not require a priori knowledge of the number of clusters [87, 88]. Due to these properties, this algorithm has previously been used to group spacecraft trajectories by their geometry in the work performed by Bosanac [71, 73], Bonasera and Bosanac [99], and Miceli et al.

[34].

HDBSCAN first assesses the similarity between the Q members of a dataset using density information. First, a core distance, d_{core}, is calculated for each member as the distance to its $N_{minCore}$-nearest neighbor; thus, d_{core} reflects the density of the dataset near the member. The core distances of members a and b are used to calculate their mutual reachability distance as $d_{mreach}(a,b) = \max\{d_{core}(a), d_{core}(b), d(a,b)\}$ where $d(a,b)$ indicates a selected distance measure [88]. In this work, the Euclidean distance metric is used. This transformation to a mutual reachability distance space further separates members within high-density regions from those in sparse regions. Furthermore, $N_{minCore}$ serves as a governing parameter that defines the size of the neighborhood used to capture density information. Figure 4.1 presents a conceptualization of the core distances, reproduced from McInnes et al. [56], for points a and b and $N_{minCore} = 7$.

With the mutual reachability distance metric, HDBSCAN builds a minimum spanning tree to locate dense regions within the dataset. The minimum spanning tree summarizes a weighted graph representation of the mutual reachability distances between the members of the dataset [88]. In this graph, each node represents a member of the dataset while edges between nodes are weighted using the mutual reachability distance of those points. The minimum spanning tree of this graph is built by locating the lowest-cost edge in the graph that connects the tree to a node that is not yet in

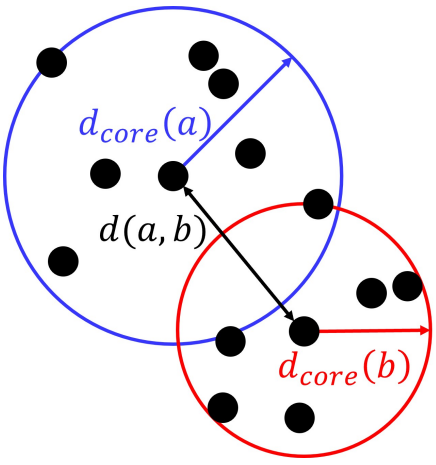

Figure 4.1: Conceptualization of core distances between points a and b.

the tree. Then, this edge and node is added to the minimum spanning tree. Building the minimum spanning tree reduces the computational complexity of understanding the hierarchical relationship of the dataset, which is an integral part of HDBSCAN. Then, an agglomerative approach is applied to this minimum spanning tree to create a hierarchy of possible clusters as a function of the mutual reachability distance via a dendrogram.

HDBSCAN leverages this dendrogram to locate the final clusters that meet size and stability requirements. First, the dendrogram is condensed into a smaller cluster tree that preserves parent clusters or allows the parents to be further divided into child clusters. The criteria for this split is determined by $N_{minClust}$, which specifies the minimum number of members in a single cluster [88]. At each division of the clusters in the dendrogram, the sizes of the clusters at the split are assessed. If the cluster size is smaller than $N_{minClust}$, the parent cluster is retained and the other members are regarded as leaving the cluster. If the cluster size is greater than or equal to $N_{minClust}$, then the child clusters are retained in the dendrogram.

The stability of each cluster is assessed to extract the final groups from the condensed dendrogram. Stability is assessed by analyzing the value of the mutual reachability distance at a cluster's birth λ_{birth}, or where it begins in the tree, and death λ_{death}, or where it reached the minimum mutual reachability of the tree or split into child clusters [56]. An additional parameter, λ_l, is defined as the value at which a point left a cluster, which happens between its birth and death. The stability, Λ, can then be computed as [56]

$$\Lambda = \sum_{l \in cluster} (\lambda_l - \lambda_{birth}) \tag{4.1}$$

All leaf nodes are selected as clusters. Then, using a bottom-up approach, the stability of the child clusters is summed. If this summation is larger than the parent cluster, the cluster stability is set to be the sum of the child stabilities. If the cluster's stability is larger, that cluster is selected for output and its descendants are ignored.

Once the root of the tree is reached, the selected clusters are returned from HDBSCAN as a summary of the dataset. Malzer and Baum present a modification of the HDBSCAN algorithm

that merges clusters separated by a distance below a threshold ϵ [13]; this modification mitigates the possibility of recovering an excessive number of small clusters that correspond to locally dense regions within a large group of sufficiently similar members. Any members not grouped are classified as noise points and may represent outliers in the data or insufficiently sampled regions. Figure 4.2 depicts a conceptualization, reproduced from McInnes et al. [56], of grouping a small dataset with HDBSCAN. Three clusters (blue, pink, and orange) along with three noise points (gray) are recovered after applying HDBSCAN to the black dataset on the left. HDBSCAN is accessed in Python using the *hdbscan* [56] package which requires a computational effort of $O(N\log(N))$ and is predominantly governed by three parameters: $N_{minCore}$, $N_{minClust}$, and ϵ [57].

Border point association is one refinement technique that is used after a set of clusters is generated to label noise points that lie in the local neighborhood of a cluster member. This post-processing step is suggested by Campello et al. in an early paper on HDBSCAN [87]. Specifically, any noise points that lie within the $N_{minCore}$-neighborhood of a labeled member in the mutual reachability distance space are considered border points of their closest cluster [87]. Because these members of the dataset lie at the border of a cluster, they are sufficiently similar to the labeled cluster members but were originally designated as noise points because their local neighborhoods in the feature vector space are sparser. Figure 4.3 presents a conceptual example, following the clustering conceptualization in Figure 4.2, of a noise point (gray) grouped into the orange cluster as a result of border point association. The $N_{minCore}$-neighborhood is displayed with a dashed orange circle for $N_{minCore} = 8$.

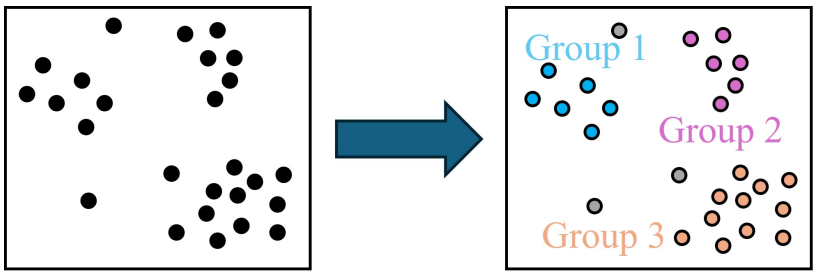

Figure 4.2: Conceptualization of clustering a small dataset with HDBSCAN.

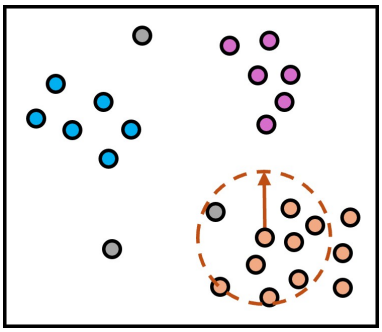

Figure 4.3: Conceptualization of a noise point (gray) grouped into a cluster (orange) via border point association.

Chapter 5

Collision-Free Path Planning

A variety of approaches have been developed to solve path planning problems. These include navigation planners such as potential functions, exact and approximate representations of the environment with cell decomposition, and sampling-based algorithms [36]. Many of these approaches were originally developed for path planning in linear dynamical systems. However, recent improvements have supported path planning in more complex environments and with uncertainty or dynamic and geometric constraints [36, 103].

Sampling-based planners employ sampling schemes to explore an environment, then connect those samples to obtain solutions to a motion query [36]. Two traditional categories of approaches are single-query and multi-query methods. Single-query methods construct solutions to new path queries with no knowledge of previous solutions whereas multi-query methods leverage information from previous path queries to construct new solutions [115]. Well-known single-query sampling-based planners include tree-based algorithms such as rapidly-exploring random trees whereas probabilistic roadmaps are a popular option for multi-query planners. These methods are also examples of probabilistically complete planners: the probability that the planner will fail to find a path, if one exists, approaches zero as additional samples are drawn [36, 101].

Environment maps represent the collision-free space where an object can safely travel. These obstacle maps may be constructed using representation techniques such as cell decomposition or Voronoi diagrams, and used for collision detection during or after the path construction process [36, 97, 105]. In these maps, dynamic obstacles are often modeled using repulsive forces, convex

shapes, or point masses [36]. Shape primitive skeletons use geometric primitives to represent an environment's topology and are a popular option for modeling static obstacles [59].

This dissertation research leverages path planning methods and environment maps to identify collision-free spacecraft trajectories. This chapter presents an overview of the techniques used in this work to automatically and rapidly construct initial guesses and identify collisions. These methods include tree-based path planners, graph search algorithms, cell decomposition, and collision checking techniques.

5.1 Tree-Based Path Planners

In the field of motion planning, tree-based planners are a popular sampling-based approach for automatically and rapidly constructing a path within an environment. This approach consists of growing a tree, represented by a set of nodes and edges, from a desired initial state to a target state via random sampling of the configuration space. A path is recovered once the tree reaches the target, to within a specified tolerance.

RRTs are randomized data structures that were originally developed to overcome challenges with other path planning techniques in high-dimensional environments for nonholonomic systems [101]. For example, potential fields rely on selecting a good heuristic potential function and probabilistic roadmaps require connecting pairs of nearby configurations, or states, for graph construction. These tasks can be challenging or computationally intensive when the environment is high dimensional or highly constrained. RRTs possess low computational complexity in high-dimensional or complex environments because they do not require point-to-point connections, solving many boundary value problems, or picking a heuristic function [101]. Instead, RRTs iteratively expand toward randomly-selected points [101].

Path recovery using an RRT consists of growing a tree from a desired initial state via random sampling of the configuration space. A conceptual example of using an RRT to construct a path between two desired configurations is shown in Figure 5.1 where the initial and target configurations are denoted with red and blue circles, respectively, and obstacles are represented with dark gray

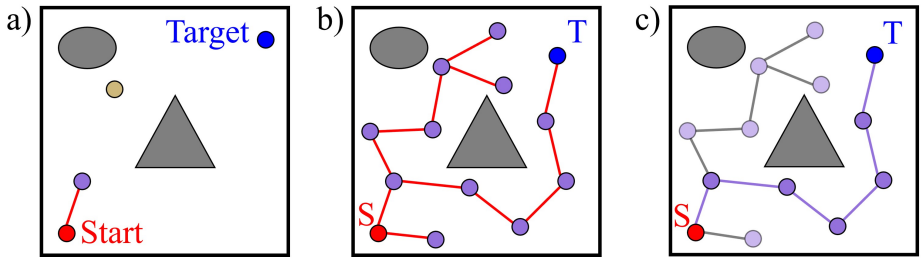

Figure 5.1: Conceptual RRT generation in a linear environment: a) randomly sample (gold) the environment and extend the tree rooted at the initial state (red) towards the random sample; b) repeatedly sample the environment and extend the tree from the nearest node until c) a path is recovered.

shapes. First, a random sample is drawn within the configuration space, denoted by the gold circle in Figure 5.1a). To extend the tree, the nearest node in the tree to the random sample is selected. This node represents a state in the environment. Then, the direction of the velocity vector is aligned with the direction to the random sample, and the state is propagated. This step adds a new branch and node to the tree, shown by a red line and purple circle in Figure 5.1a). The nearest node is often identified in the configuration space, but could also be found in the velocity or full phase space. The process is repeated to explore the environment, as portrayed in Figure 5.1b), until a termination condition, such as reaching a goal. By traversing the tree, a path is recovered, as displayed in Figure 5.1c) in purple.

Variations on the RRT algorithm have also supported path planning in complex motion planning problems. Bidirectional RRTs grow trees from each of the boundary conditions [36, 37, 104]; expansive-spaces trees rely on probabilistic sampling to ensure sufficient coverage in regions of low density [36]; and stable-sparse RRTs reduce computational time by retaining only a sparse set of nodes and edges [122]. Multipartite [67] and time-based RRTs [10] have been used for robot path planning in dynamic systems: multipartite RRTs leverage tree branches from previous motion queries and bias in the sampling distribution whereas time-based RRTs track the time and state at each node. Furthermore, multi-RRT algorithms employ a forest, composed of multiple trees, to recover paths and have been used in robotics to visit target locations while minimizing travel cost

[46], exploit local structure for exploring narrow passageways [108], and balance exploration and exploitation to rapidly find a feasible path [124]. Multi-RRT algorithms have also been used in uncrewed aerial vehicle path planning for application to indoor rescue scenarios [42].

5.2 Graph Search Techniques

Motion planning strategies often rely on the use of directed graphs. A directed graph is a collections of vertices, representing objects, and directed edges, representing the relationships between objects in a specific order [36, 93]; edge weights can be used to define the cost of traversing each edge. Traditionally, the root of a graph is a special vertex that possesses no parent vertices. A parent has vertices accessible to it called children, and a leaf is a vertex that has no children [36]. The number of edges that are incident into a vertex from its parents defines its in-degree whereas the number of edges that are incident from a vertex to its children defines its out-degree [93]. Characterizing vertices by in- and out-degrees supplies insight into the traversability of the directed graph.

Graph search algorithms may be applied to a graph to recover a sequence of vertices connected by edges; in path planning, this sequence may produce a path through the environment [36]. In a specified application, selection of a search algorithm depends on the desired search metrics, available computational effort, and types of solutions sought. Classical breadth-first searches guarantee that the first solution uses the smallest number of vertices and edges within the graph whereas depth-first searches find longer paths earlier in the search [102]. It may be desirable to optimize path length or other search metrics, including safety, time, number of vertices to explore, diversity, or computational efficiency. Heuristic-informed graph search algorithms, such as A* and Yen's, may return optimal paths with respect to the search metric. However, selection of a suitable heuristic can be a challenging task in some applications and a poorly-defined heuristic may result in longer search times and suboptimal paths [36].

A* is a best-first graph search algorithm that identifies the path that minimizes the sum of the edge weights between a selected pair of start and goal vertices [83]. This search algorithm uses

a queue to store and rank incomplete paths that are constructed by adding neighboring vertices to the current sequence of vertices. At each iteration, A* identifies the path to expand from the queue as the path with the lowest cost. The current vertex is then the most recently added vertex of the selected path. Next, the algorithm searches for all neighbors of the current vertex and computes a cost for each neighbor in the graph. The cost of the neighboring vertex $(i+1)$ is defined as $c_{i,i+1} = g_{i,i+1} + h_{i+1,targ}$, where $g_{i,i+1}$ is the cost to traverse the graph from the current vertex i to the neighbor vertex and $h_{i+1,targ}$ is the heuristic or estimated cost to traverse the graph from the neighbor vertex to the target vertex. Although problem-dependent, the heuristic must be defined such that $h_{j,targ} \leq h^*_{j,targ}$, where h^* is the exact value of the heuristic or true cost from vertex j to the target vertex, to ensure that the algorithm returns the overall lowest cost solution [83]. The incomplete paths obtained by adding each neighbor to the current path are stored in the queue with their cost. This process is repeated until the path with the lowest cost in the queue reaches the target vertex [83]. When the heuristic is selected as zero, A* reduces to the well-known Dijkstra's algorithm [102].

Yen's algorithm identifies k diverse paths with the lowest cost between a pair of start and goal vertices, originally using Dijkstra's as the single-best path algorithm [29]. First, Dijkstra's algorithm is employed to find the optimal path through the graph. This solution is placed in a list that stores the k diverse solutions to the motion query, labeled as list $\mathcal{A}$. Then, the graph and the best path are iteratively modified by removing edges between the vertices comprising the path. At each iteration, the root path is defined as the vertex sequence from the first vertex to the vertex associated with the removed edge. If the best path is composed of N vertices, Yen's algorithm will create $N-1$ root paths and subgraphs, i.e., graphs where one edge between vertices i and $(i+1)$ of the path has been removed. Then, the selected search algorithm is used to recover the best path from the last vertex of the root path to the target vertex in each subgraph. The best paths from each of these subgraphs are saved in a temporary list, labeled as list $\mathcal{B}$, and sorted by their cost. The path with the lowest cost from list $\mathcal{B}$ is moved to list $\mathcal{A}$, and the process is repeated with the lowest cost path moved to list $\mathcal{A}$ until k paths exist in list $\mathcal{A}$ [50]. Although Yen's algorithm is not

necessarily computationally efficient, it supports the proof of concept used within this work.

5.3 Cell Decomposition

Previous researchers have used cell decomposition, which relies on partitioning the configuration space into cells, to find a preliminary collision-free path that guides RRT construction across different disciplines [3, 4, 31, 44, 82, 97, 106]. Although a variety of decomposition strategies exist, they can generally be categorized into exact or approximate methods. Exact cell decomposition decomposes the obstacle-free configuration space into a set of non-overlapping regions whose union exactly describes the obstacle-free region [41]. Additionally, the geometry of the obstacle-free configuration space may guide selection of the cell shape to ensure the environment is accurately represented [41]. Conversely, approximate cell decomposition conservatively approximates the obstacle-free space using cells with simple pre-specified shapes, such as squares or rectangles [41]. These shapes are standardized to reduce sensitivity to numerically-approximate computations and decompose the space by iteratively applying the same computation [41]. Approximate methods allow the user to control the minimum cell resolution and are typically easier to implement than exact methods [41].

5.4 Collision Checking Techniques

Dynamic obstacle maps have been used with collision detection algorithms to identify potential collisions. A challenge with dynamic obstacles is accurately tracking their motion as a function of time, which increases the complexity of collision detection compared to static environments. A variety of approaches to collision detection exist throughout the literature including conservative advancement techniques [47], shape primitive skeletons [59], continuous collision detection [105, 119], lazy collision detection [9, 36, 86], or cell-based collision detection [102, 123].

This work uses cell-based and lazy collision checking techniques to identify potential collisions between spacecraft transfers and dynamic obstacle corridors. Specifically, dynamic obstacle maps conservatively represent the obstacle corridors that are associated with a local initial state

set where cells containing an obstacle corridor are designated as potential collision cells. Then, lazy collision detection is employed to analyze a spacecraft trajectory for collisions with obstacle corridors after the trajectory is generated. A benefit to this approach, as opposed to incorporating obstacle information into the path planning process, is a reduction in the computational complexity of collision checking by performing checks only when necessary [36]. However, not all solutions recovered during path planning may be feasible.

Chapter 6

Construct a Digestible Motion Type Summary

A challenge that is emerging in space situational awareness and space traffic management is analyzing the possible future motions of an object in the presence of uncertainty within cislunar space. In a chaotic system, uncertainty in the initial state may result in a variety of possible paths over sufficiently long time intervals. The level of uncertainty associated with a state estimate may also increase when off-nominal conditions are present, the objects are difficult to observe and track over time, and when characterizing an unknown object [2]. As the initial state estimate, level of uncertainty, and model parameters all vary, the array of possible future motions can evolve in geometry and itinerary as the paths pass through distinct regions of the Earth-Moon system. In these motivating examples, it may be valuable to extract a clear, digestible summary of the distinct motion types that originate from a local region of the phase space.

This chapter presents a clustering-based approach for automatically categorizing a diverse array of trajectories generated from a local point cloud of initial states that begin near a reference state. Each of the generated clusters supplies a 'motion type' that captures a set of trajectories with similar geometric and temporal variations. As a result, the collection of clusters supplies a digestible summary of the set of trajectories. The data-driven categorization process consists of five steps that are depicted in the flowchart in Figure 6.1 and described in detail in each section of this chapter. Furthermore, each step of the technical approach is demonstrated for planar trajectories generated from a set of initial states in the vicinity of a reference state along an L_1 Lyapunov orbit in the Earth-Moon CR3BP. The computational times for each step are determined using a

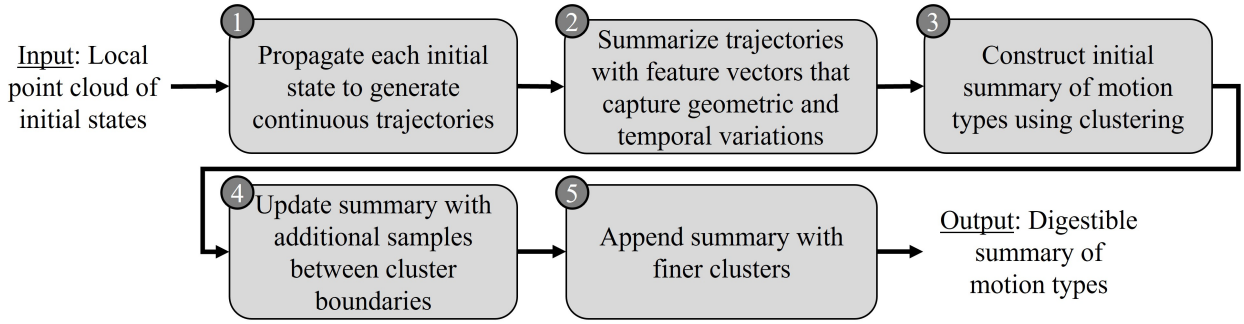

Figure 6.1: High-level overview of the data-driven categorization process.

computer with an AMD Ryzen 5 5600G with Radeon Graphics 4.20 GHz processor. Note: This work has been published in an article by Spear and Bosanac in the Journal of Guidance, Control, and Dynamics by AIAA [92].

6.1 Generating Input Data

The data-driven categorization process takes an input of a set of initial states that lie in the local neighborhood of a reference state. Throughout this chapter, this set is labeled as a local point cloud of initial states, encompassing a hyperellipsoid in six-dimensional space that is centered on the selected reference state. This local point cloud could physically represent a snapshot of the state space where an object is reasonably expected to exist at a selected epoch within, for example, 1) a 3σ uncertainty region around a state estimate or 2) a pre-specified corridor for targeting a reference state. These motivating examples often involve associating probabilities to each initial state. However, this work relies on sampling initial conditions within the boundary of the local point cloud to identify the dominant geometries of the associated trajectories regardless of their likelihood of occurrence. Future work that applies this data-driven categorization process to trajectory prediction problems could incorporate probabilities during sampling, categorization, or after generating a digestible summary of possible motions.

Deterministic, geometric, or hybrid sampling schemes may be employed to sufficiently sample the state space within the local neighborhood of a reference state [20, 25]. For a uniform,

dense distribution of samples, suitable approaches may include Halton point, spiral, or grid-based methods. Halton points are a low-discrepancy point set generated by a quasi-random number sequence known as the Halton sequence [55]. This set is guaranteed to be well distributed over all dimensions and maintain low-discrepancy as samples are added to the sequence [65]. Spiral sampling with the Golden ratio ensures a near-uniform, space-filling pattern with a faster computation time than low-discrepancy point sets based on the van der Corput sequence, such as Halton points [7, 14]. Geometric grid-based sampling methods fill the desired space by sampling at fixed intervals uniformly distributed in every dimension [20]. Square, uniform grid-based sampling is one of the simplest methods that may be used to sufficiently sample a region [20].

Each of these methods possesses a challenge with respect to its application in this work. While Halton points fill the state space with low non-uniformity, they may leave areas under-sampled, particularly in higher dimensions as the samples can exhibit regular patterns [55]. Additionally, variations on Halton point sampling may be less uniform than other sampling schemes [20]. Spiral sampling has been successfully applied in two- and three-dimensional scenarios [7, 14, 84, 94, 95]; however, in the spatial case, samples are taken from the surface of three-dimensional objects. The challenge for the application of spiral sampling in this research is to uniformly distribute the samples within a multi-dimensional object. Meanwhile, grid-based approaches can guarantee densely-packed sampling in two- and three-dimensions, but suffer from sparse region coverage in higher-dimensions [20].

The scheme selected to sample initial states within the local point cloud in this work balances supplying a sufficiently dense set of samples with reducing the total number of samples for computational feasibility. Although uniformly sampling each of the six dimensions of the phase space within a specified region around a reference state is conceptually straightforward, this approach was observed to suffer from sparse coverage when generating a reasonable number of samples [20]. To reduce the number of samples while sufficiently covering the local region of the phase space, position and velocity vectors are first independently sampled using a grid-based approach that supplies dense and uniform sampling only in their associated three-dimensional subspaces. Samples are

defined using a square, uniformly-spaced grid and only those n_{IC} position vectors or n_{IC} velocity vectors that lie within the specified boundaries of the local point cloud are retained. Then, the pseudorandom number generator in MATLAB's *randperm* function is used to identify two distinct sequences of unique integers between 1 and n_{IC} [68]. The i-th integer in each list identifies the specific position and velocity vectors that are combined to form the i-th of n_{IC} six-dimensional state vectors.

To demonstrate the sampling approach used in this chapter, consider a local point cloud of initial states that exist near a single reference state along an L_1 Lyapunov orbit with a period of 12.27 days and $C_J = 3.1542$ in the Earth-Moon CR3BP. This nondimensional reference state is specified in the rotating frame as $\bar{x}_{PO} = [0.816988444235, 0, 0, 0, 0.195756600373, 0]^T$, occurring at the leftmost crossing of the x-axis. Planar states that lie within 10.5 km in position components and 10.5 m/s in velocity components of the reference state are sampled using the method described in the previous paragraph. This local point cloud size corresponds to the 3σ uncertainties estimated by Bradley et al. [75] for cislunar navigation using optical (angles only) measurements. Although Fedeler et al. [100] note that uncertainty levels may vary for different types of motions, the proof of concept in this dissertation uses the same local point cloud size for each of the selected scenarios. In this L_1 Lyapunov scenario, 1,006 states are sampled, which typically requires on the order of 10^{-2} seconds on the specified computer. Across this set of initial states, the Jacobi constant varies on the order of 10^{-3} nondimensional units.

6.2 Step 1: Propagate Continuous Trajectories

Each of the input initial states is propagated in a selected dynamical model for a specified duration. In this research, all initial conditions are propagated forward in time for at least 17.3 days in the Earth-Moon CR3BP; this duration exceeds the periods of the selected orbits and allows distinct geometries to emerge, if they exist. However, propagation is terminated early upon impact with a spherical approximation of either of the primaries, defined using their equatorial radii of 1,738.0 km and 6,378.1363 km for the Moon and Earth, respectively [16]. Following this procedure,

Figure 6.2 displays a 200-member subset of the continuous trajectories generated from the 1,006 initial conditions used in this scenario with the L_1 Lyapunov orbit shown in blue. Generating the full set of continuous trajectories, $\mathcal{T}$, typically requires on the order of 0.1 seconds in C++ on the specified computer. These initial conditions occur near the red circle with the Moon and Earth appearing as gray circles scaled to possess the radius of the respective body, and red diamonds locating L_1 and L_2. Figure 6.2 demonstrates that paths either impact the Moon, remain in the vicinity of the Moon, or depart through one of the L_1 or L_2 gateways, producing a variety of distinct geometries.

6.3 Step 2: Summarize Trajectories

To enable categorization via clustering, each continuous trajectory is discretized into a sequence of p states at equal intervals in arclength. When numerically integrating a state from t_1 to t_2, the arclength, s_{arcL}, is defined as the distance traversed along the trajectory [113]. In this work, this quantity is calculated along a trajectory in the rotating frame as

$$s_{arcL} = \int_{t_1}^{t_2} |\bar{v}|\, dt \tag{6.1}$$

where $\bar{v} = [\dot{x}, \dot{y}, \dot{z}]^T$ is the nondimensional velocity vector. Then, p states, including the final state but excluding the initial state, are sampled at intervals of s_{arcL}/p along the arclength of the trajectory. The initial state is excluded because all initial states lie within a local neighborhood and,

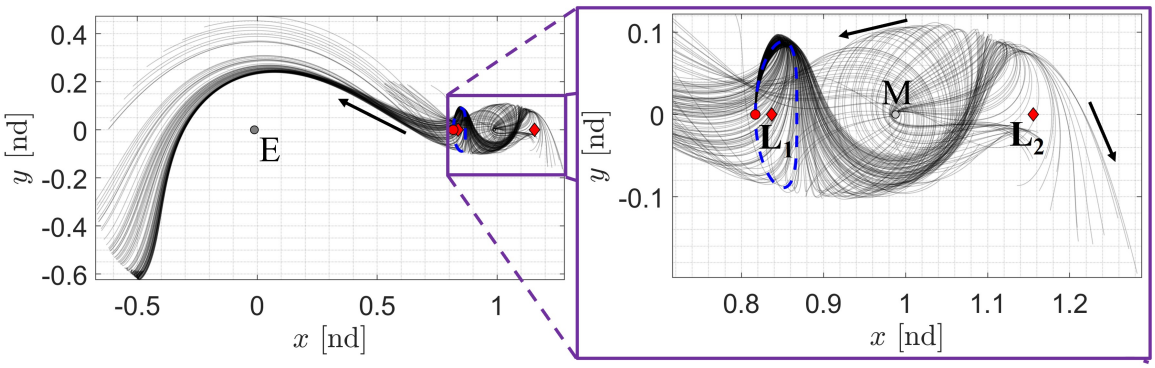

Figure 6.2: A 200-member subset of planar trajectories propagated for 17.3 days from the sampled initial conditions in the Earth-Moon CR3BP.

therefore, do not meaningfully contribute to identifying geometric differences between trajectories. Although there are various possible approaches for discretizing a continuous path, this work employs an equal arclength distribution to 1) limit the influence of variations in the speed along a trajectory, and 2) ensure sample locations smoothly vary as the geometry gradually evolves. However, a limitation of this approach is that the samples may not be placed in geometrically meaningful locations, such as apses, or sufficiently capture smaller revolutions along a trajectory that traverses a large distance when p is too small.

To help mitigate this limitation, the number of states, p, used to discretize each trajectory is selected heuristically using a curve-based approach. The curvature, $\kappa(\bar{x})$, reflects the deviation from a straight line at a state $\bar{x}$ along a trajectory. At a single state, this quantity is defined as [113]

$$\kappa(\bar{x}) = \frac{\sqrt{(\ddot{z}\dot{y} - \ddot{y}\dot{z})^2 + (\ddot{x}\dot{z} - \ddot{z}\dot{x})^2 + (\ddot{y}\dot{x} - \ddot{x}\dot{y})^2}}{(\dot{x}^2 + \dot{y}^2 + \dot{z}^2)^{3/2}} \tag{6.2}$$

Local maxima in the curvature supply geometrically meaningful locations along a trajectory and typically occur near periapses and apoapses measured relative to primaries or equilibrium points. However, unlike apses, maxima in the curvature do not require specification of a reference point as one or more trajectories exhibit turning points in distinct regions of the Earth-Moon system. To identify the number of local maxima in the curvature along a single trajectory, p_{traj}, the curvature is calculated at each state along the trajectory in the rotating frame via Equation 6.2 and then the local peaks in the time history of $\kappa(\bar{x})$ are identified. The largest number of local maxima in the curvature along each trajectory in the dataset defines the value of p_{max}. Then, each trajectory in the dataset is sampled using $p = 2(p_{max} + 1)$ states that are equally-distributed in arclength. This value of p is selected to ensure there are at least twice as many samples as the number of significant geometric locations, including the final state, along the trajectory. As a result, this value of p has been observed to sufficiently summarize all trajectories in the datasets examined in this work. Furthermore, each trajectory in the dataset is summarized by the same number of samples, reducing the complexity of clustering.

To demonstrate the trajectory discretization process, consider a single, planar trajectory generated in Step 1. Figure 6.3a) displays this trajectory along with blue stars at the locations of maximum curvature and a black star at the final state. Repeating this calculation for each of the 1,006 trajectories generated for 17.3 days in Step 1 produces $p_{max} = 7$, consistent with the number of curvature maxima that occur along the trajectory in Figure 6.3a). This trajectory is then discretized into $p = 16$ states that are spaced equally in arclength. Figure 6.3b) shows the resulting discretization with the sampled states, indicated by black stars, sufficiently capturing the shape of the path within the configuration space.

The p sampled states are used to form two finite-dimensional feature vectors that, together, capture the spatiotemporal evolution of a trajectory. Although position-based feature vectors are a viable option for encoding the spatial evolution of a trajectory, different regions of a multi-body system possess distinct sensitivities to the same distance between two sampled position vectors. This challenge can result in lower quality clusters generated from sequences of position vectors, including excessive grouping, incorrect grouping, or excessive differentiation. Furthermore, it is difficult to intuitively and automatically select suitable tolerances for two trajectories to be considered sufficiently similar. To address this challenge, Bosanac has demonstrated using velocity unit vectors to cluster spacecraft trajectories by their shape [74]. Applying this concept to this work, all trajectories begin from a single local neighborhood in the phase space. Therefore, two trajectories with

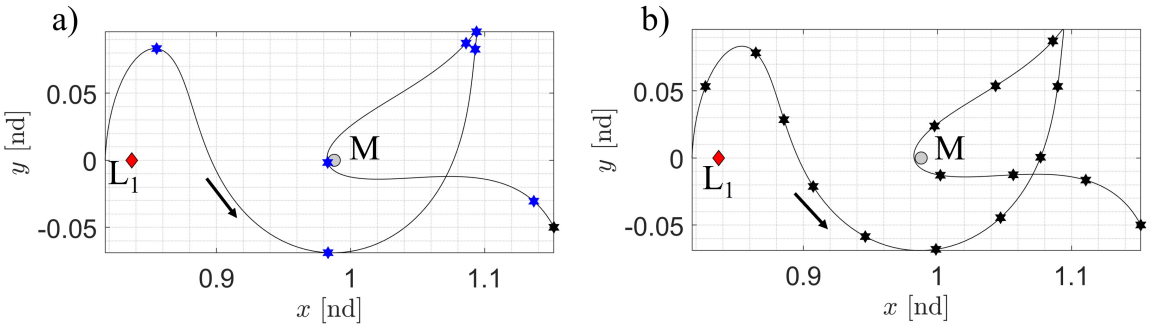

Figure 6.3: a) Locations of local maximum curvature along a trajectory. b) The locations of p states spaced equally in arclength along the trajectory.

a similar shape and time evolution are expected to follow similar paths through the configuration space. Accordingly, the following two separate feature vectors are used in this dissertation: 1) $\bar{f}_{\hat{v}}$ captures the shape of the path via the velocity direction at sampled states and 2) $\bar{f}_{\Delta\tau}$ captures the temporal evolution via the elapsed time between subsequent states. Mathematically, these feature vectors are calculated as

$$\bar{f}_{\hat{v}} = \left[\hat{\bar{x}}_1, \hat{\bar{y}}_1, \hat{\bar{z}}_1, \cdots, \hat{\bar{x}}_p, \hat{\bar{y}}_p, \hat{\bar{z}}_p\right] \qquad \bar{f}_{\Delta\tau} = [\Delta\tau_1, \cdots, \Delta\tau_p] \tag{6.3}$$

where $\hat{\bar{x}}_i, \hat{\bar{y}}_i, \hat{\bar{z}}_i$ are the components of the velocity unit vector at sampled state i in the rotating frame. In addition, $\Delta\tau_i$ is the nondimensional time elapsed from the previous state, $\bar{x}_{i-1}$, for $i > 1$ or the initial time, $\tau_0 = 0$, when $i = 1$, to the current state, $\bar{x}_i$. Through these definitions, $\bar{f}_{\hat{v}}$ is a $3p$-dimensional vector for spatial motion whereas $\bar{f}_{\Delta\tau}$ is a p-dimensional vector. For planar trajectories, the z-component of the velocity unit vector at each state is removed from $\bar{f}_{\hat{v}}$ to avoid including redundant information. In the selected scenario, summarizing the full set of trajectories typically requires around four seconds on the specified computer with $O(N)$ complexity for each step. The process for summarizing trajectories is outlined in the flowchart provided in Figure 6.4.

6.4 Step 3: Construct Initial Summary of Motion Types

With a selected set of input parameters, HDBSCAN is used to automatically group trajectories by their spatiotemporal variations, producing an initial summary of motion types. This clustering process is summarized in the flowchart provided in Figure 6.5. First, trajectories are

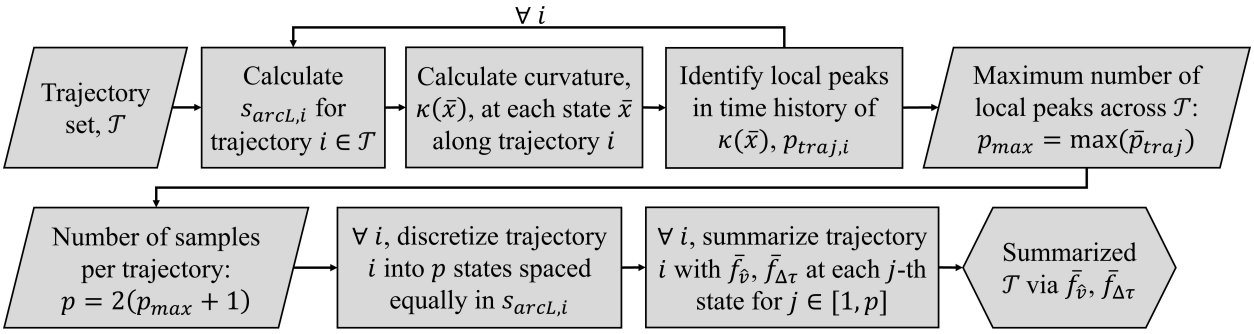

Figure 6.4: Overview of the trajectory summarization process in Step 2.

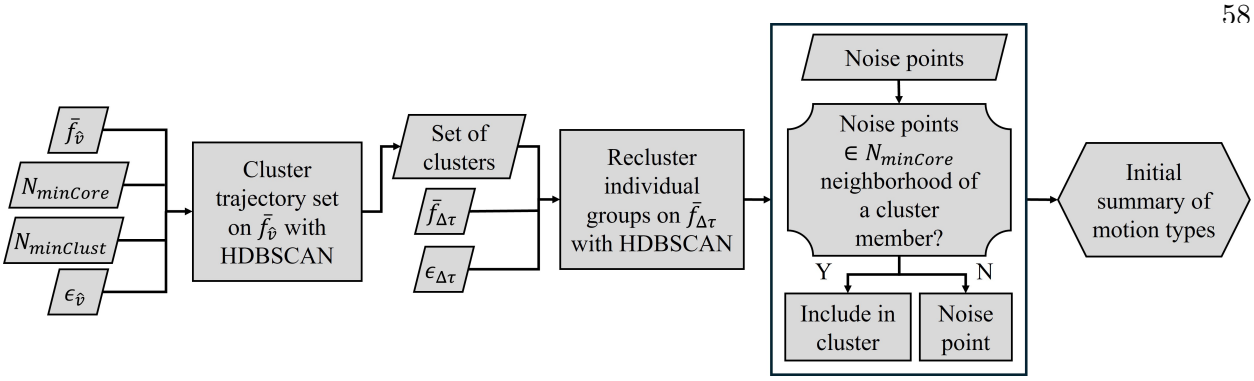

Figure 6.5: Overview of the process for generating an initial motion type summary in Step 3.

clustered using their velocity direction feature vectors, $\bar{f}_{\hat{v}}$, to discover groups of geometrically similar trajectories. Although the trajectories are generated from a set of initial states that lie within a local region of the phase space, their geometric variations over sufficiently long time horizons are expected to result in dense groupings of the data within the higher-dimensional feature space associated with $\bar{f}_{\hat{v}}$. Each dense grouping corresponds to a cluster that should be identified by HDBSCAN. Then, each cluster is refined by clustering only its members via their time-based feature vectors $\bar{f}_{\Delta\tau}$. This cluster refinement step supports outlier removal and allows groups to be further separated, as needed, based on additional temporal variations. After this initial two-step clustering process, each trajectory is either assigned to a group or labeled as noise. Then, border point association, as outlined in Chapter 4, is performed to label noise points that lie in the local neighborhood of a cluster member.

Although $\bar{f}_{\hat{v}}$ and $\bar{f}_{\Delta\tau}$ could be weighted and combined to construct a single $4p$-dimensional feature vector, simultaneously clustering a dataset with distinct features can result in a higher likelihood of inaccurate groupings. Specifically, the same difference between two weighted but distinct feature types could result in two trajectories being assigned to the same cluster despite substantial differences in their geometric and/or temporal variations. However, independently using the two feature vectors has been observed in this work to improve the cluster accuracy while also eliminating the need to strategically weight the distinct features.

The parameters governing HDBSCAN are manually defined to be consistent across all sce-

narios in this dissertation. However, automated and adaptive parameter selection is a valuable avenue of future work. Manual selection of the minimum cluster size and neighborhood size begins with setting $N_{minCore} - 1 = N_{minClust}$ to measure density across the same-sized neighborhood as the minimum acceptable cluster size. Then, a small value of $N_{minClust}$ is selected to prioritize local density estimates and variations between nearby members of the data. In addition, the Euclidean distance is selected to assess the difference between two feature vectors to support fast clustering. Finally, the minimum threshold for cluster differentiation in the velocity-based feature space is heuristically selected as $\epsilon_{\hat{v}} = 2\sqrt{p}\sin\left(\frac{\alpha}{2}\right)$. This heuristic, as presented by Bosanac, reflects the Euclidean distance between a set of p unit vectors separated by an angle equal to α [74]. For cluster refinement in the time-based feature space, $\epsilon_{\Delta\tau} = N_{minCore}\max(d_{mc}, \epsilon_{thresh})$ where d_{mc} is the maximum core distance from each trajectory to its nearest neighbor and ϵ_{thresh} is a user-selected small minimum threshold. Similar to the heuristic presented by Bosanac, this quantity estimates a reasonable $N_{minCore}$-neighborhood radius that adapts to the trajectories within the cluster to be refined and is not biased by significant outliers [74].

Applying this clustering process to the trajectories generated from the local point cloud of initial states near the selected reference state along a 12.27-day L_1 Lyapunov orbit results in successful separation of spatiotemporally distinct trajectories. The 1,006 trajectories are grouped into 13 clusters using $N_{minCore} = 4$ and $N_{minClust} = 5$ for clustering in both $\bar{f}_{\hat{v}}$ and $\bar{f}_{\Delta\tau}$, $\epsilon_{\hat{v}}$ is calculated using $\alpha = 5°$, and $\epsilon_{\Delta\tau}$ is calculated using $\epsilon_{thresh} = 10^{-3}\sqrt{p}$. On the specified computer, this step requires on the order of one second to complete. The initial summary is presented in Figure 6.6 with each subfigure displaying one cluster of trajectories in the Earth-Moon rotating frame. The L_1 Lyapunov orbit is depicted with a black dashed line in cluster 11. The horizontal and vertical axes represent the x- and y-position coordinates, respectively, in the Earth-Moon rotating frame in nondimensional units. In addition, the final state of each trajectory is denoted with a blue dot whereas color indicates the elapsed time along each trajectory. Black arrows denote the direction of motion. Twelve trajectories are categorized by HDBSCAN as noise points and are not depicted in this summary. This noise rate, approximately 1.2%, is considered acceptable because these

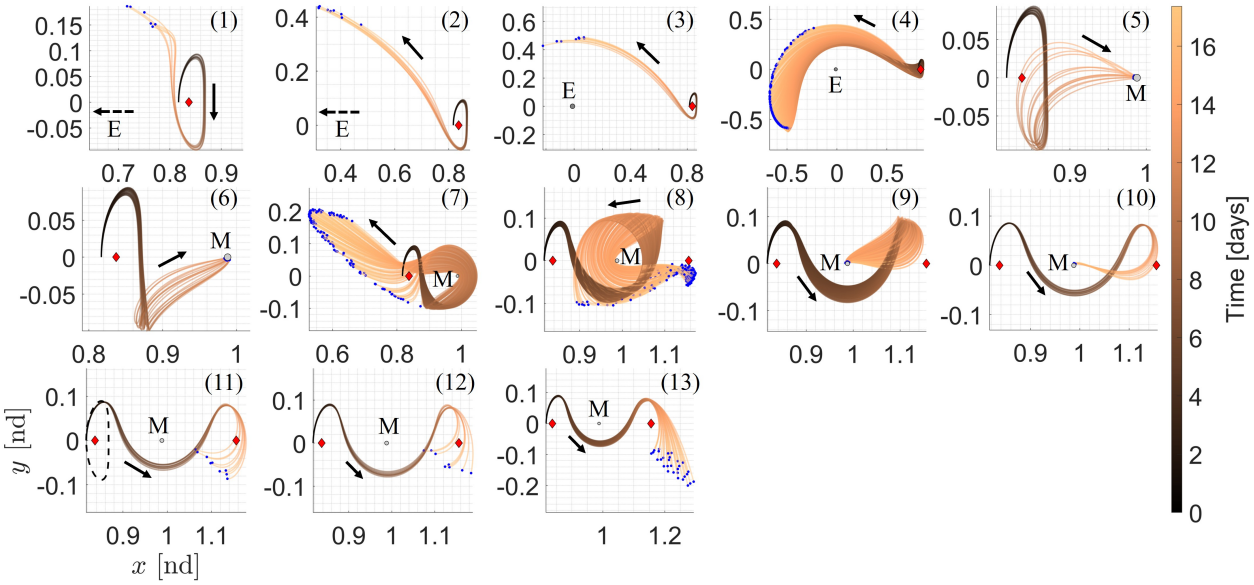

Figure 6.6: Initial summary of trajectories generated from a local point cloud of initial states along the selected L_1 Lyapunov orbit in the Earth-Moon CR3BP.

noise points either 1) resemble existing clusters but do not lie within the $N_{minCore}$-neighborhood of cluster members, or 2) do not exist near enough trajectories with similar variations in shape and time to create a separate cluster of at least $N_{minClust}$ members.

Figure 6.6 reveals that each cluster contains trajectories that are geometrically similar. For instance, clusters 1-4 capture trajectories that briefly resemble three-quarters of an L_1 Lyapunov orbit before departing for the vicinity of the Earth. Clusters 7-8, however, contain trajectories that resemble half an L_1 Lyapunov orbit before revolving around the Moon once with a low perilune. These trajectories are separated into two clusters because there is a distinct split in the velocity direction along the trajectories.

6.5 Step 4: Update Summary with Additional Samples

Clustering algorithms aim to separate a dataset into groups based on the selected features. In some cases, the dataset may be insufficiently dense in the feature vector space between clusters or along their boundaries. In these scenarios, cluster refinement may increase the accuracy of cluster boundaries and member assignment by identifying sufficiently similar members or increasing the

density of the dataset in sparse regions.

To increase sampling near cluster boundaries or in sparse regions between clusters, additional initial states are sampled from within the local point cloud and used to update the summary produced in the previous step. To contextualize the formulation of this process, spatiotemporal variations between the trajectories in distinct clusters from Step 3 are observed to be correlated to variations in the velocity components of the initial state. To demonstrate this observation, Figure 6.7a) displays the velocity components of each initial condition colored by the assigned cluster with white diamonds locating points designated as noise by HDBSCAN. Examining the connection between the clustering results and the structure of the local neighborhood of the reference trajectory is an interesting avenue of future work. Nevertheless, in this example, the majority of noise points lie between cluster boundaries. Therefore, expanding the trajectory set with additional samples between these boundaries may aid in grouping noise points along with increasing the accuracy of cluster boundaries.

The inter-cluster sampling scheme is inspired by a Voronoi diagram which divides an E-dimensional space into cells such that each cell consists of one point and cell boundaries lie at the midpoint between two points [30]. In this work, every velocity vector from the original initial state

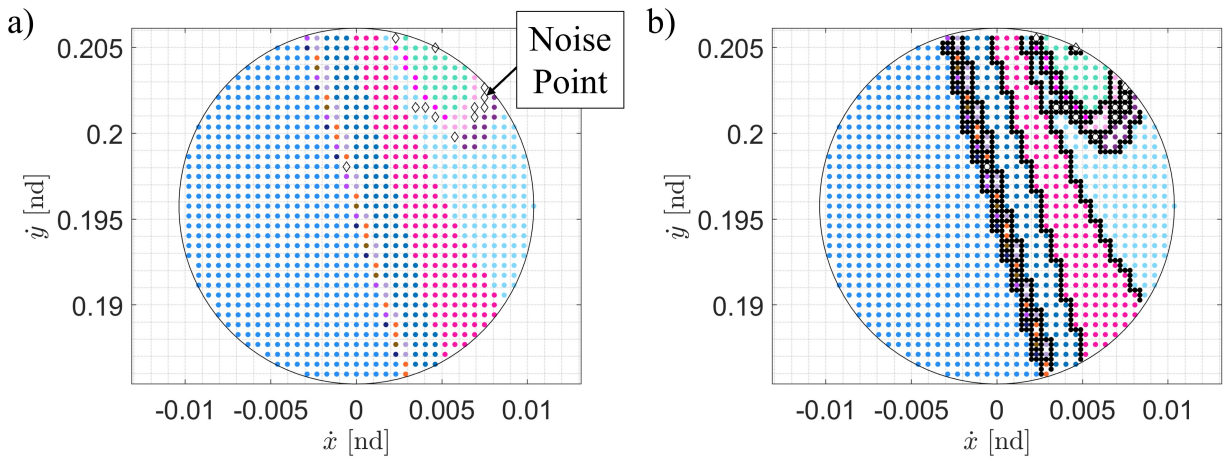

Figure 6.7: Velocity components of initial conditions a) after initial clustering and b) with 536 new samples (black circles) after inter-cluster sampling.

set is treated as if it lies in its own cell in velocity space. Then, the cluster labels for adjacent cells are compared to determine if a sample is needed between them. This approach enables the sampling of additional initial states without relying on explicit boundary construction.

Adjacent samples are located based on Euclidean distances between initial conditions. In general, for one- to three-dimensional points, the distance, d_{pts}, between any velocity vector, $\bar{v}_{pt,j}$, and its adjacent neighbors, $\bar{v}_{pt,i}$, in a square unit grid is $d_{pts} = ||\bar{v}_{pt,j} - \bar{v}_{pt,i}|| < 2u_{grid}$ where u_{grid} is the grid size. Therefore, the neighbors adjacent to each velocity sample are identified by calculating the distance between each j-th velocity sample and all other samples. If $d_{pts} < 2u_{grid}$, the point is an adjacent neighbor. Figure 6.8 conceptually depicts a set of 16 planar initial conditions sampled in a uniform grid with spacing u_{grid} in position and velocity space. The cells for each of the sampled points are denoted with dashed, light gray lines. The initial conditions are color-coded by cluster assignment with noise points shown in gray. Structure is present in velocity space, but less apparent in position space.

Cluster labels are compared for the trajectories generated from the velocity vectors of two initial states that lie in adjacent cells. If the cluster labels are different, a new velocity vector is sampled at the midpoint between the original two velocity vectors. To complete the construction of a six-dimensional state vector, the position vector of this new initial state is recovered by placing a sample at the midpoint between the position components corresponding to the two original velocity samples. Note that new samples are also placed between noise points since the initial conditions

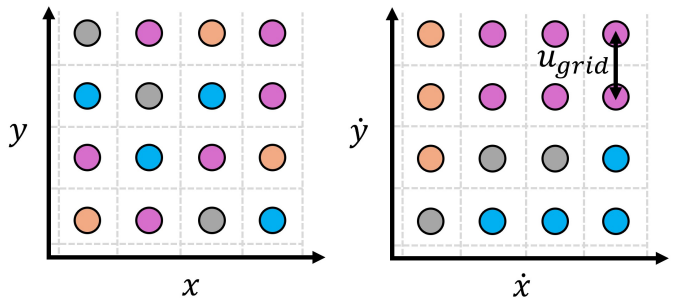

Figure 6.8: Conceptual depiction of a set of planar initial conditions uniformly sampled in a square grid and colored according to cluster assignment.

corresponding to noise point trajectories are not members of a group. A conceptual example of this sampling scheme, including samples between adjacent noise points, is presented in Figure 6.9. The 19 new planar samples are shown with black circles, where one is colored red for clarity as an example of a paired set of vectors in position and velocity space that compose a new six-dimensional state vector. The full set of initial conditions now contains 35 state vectors (16 from the original sampling and 19 from this refinement step). This inter-cluster sampling scheme is applied to the L_1 Lyapunov orbit scenario used throughout this chapter. This produces an additional 536 planar initial conditions which fall between cluster boundaries, shown in black in Figure 6.7b). The inter-cluster sampling process is summarized in this section in the flowchart provided in Figure 6.10.

The trajectories associated with the expanded set of initial conditions are clustered with HDBSCAN to construct a new summary. First, the new trajectories are summarized with the method presented in Step 2. Then, all trajectories are clustered using the same method and governing parameters presented in Step 3 followed by border point association. The result is an updated set of clusters and noise points.

This step sometimes results in some of the clusters from the initial summary being joined as two or more separated groups gain sufficient new samples between them to identify a continuous variation between members of the two groups. As an example, consider the expanded dataset generated from initial states near the leftmost x-axis crossing of the 12.27-day L_1 Lyapunov orbit in the Earth-Moon CR3BP. In this case, clusters 9-13 of Figure 6.6 are joined to produce a single

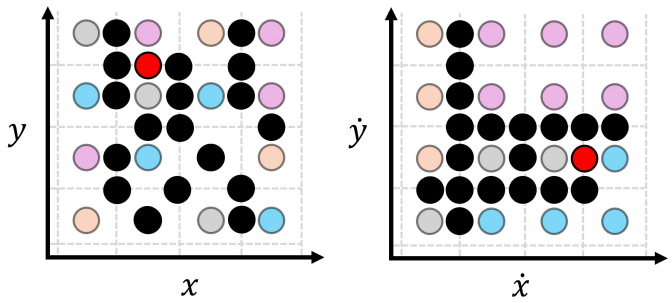

Figure 6.9: Conceptual depiction of inter-cluster sampling between cluster boundaries where black circles denote new samples.

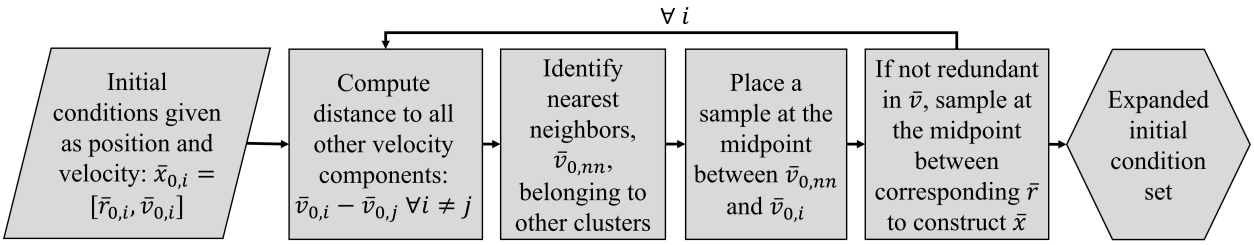

Figure 6.10: Overview of the inter-cluster sampling process in Step 4.

group of trajectories that briefly resemble the L_1 Lyapunov orbit for approximately one quarter of a revolution before impacting the Moon, performing a close approach to the Moon with a low perilune, or departing through the L_2 gateway. Although the trajectories terminate in different regions, their shape continuously varies within the cluster and over this time horizon. Alternative sampling schemes and feature vector definitions may enable HDBSCAN to separate this large cluster of trajectories as their distribution in the higher-dimensional feature vector space evolves, thereby producing a more detailed summary with a higher number of smaller clusters. For this scenario, this step requires around eight seconds on the specified computer with a computational complexity of $O(N^2)$.

6.6 Step 5: Append Summary with Finer Clusters

The final step of the data-driven categorization process involves grouping the remaining noise points to capture finer clusters that may consist of fewer members. Specifically, there may be some trajectories that are designated as noise, but are similar to fewer than $(N_{minClust} - 1)$ members. Accordingly, this final step leverages the clustering method presented in Step 3 to identify smaller groups of noise points that correspond to insufficiently sampled or low-density regions in the larger dataset. In this case, the HDBSCAN input parameters are manually selected with smaller values of $N_{minCore}$ and $N_{minClust}$ to construct locally-dense groups with $\epsilon_{\Delta\tau} = \epsilon_{\hat{v}} = 0$ due to the small number of trajectories in this dataset. Then, these new, smaller clusters are appended to the summary of the expanded dataset produced in Step 4. After this step, some trajectories may still

be designated as noise.

Applying this final step to the remaining unlabeled trajectories after Step 4 in the 12.27-day L_1 Lyapunov orbit scenario in the Earth-Moon CR3BP allows the discovery of finer groupings from amongst the unlabeled trajectories. Selecting $N_{minCore} = 1$ and $N_{minClust} = 2$ results in eight small clusters that are appended to the cluster summary generated in Step 4, producing a total of 21 clusters; selecting $N_{minCore} = 1$ removes the notion of density when calculating the mutual reachability distance. This step produces a larger number of more detailed clusters from the reduced dataset. Using the same representation as Figure 6.6, Figure 6.11 displays the 21 motion types that capture the array of geometries exhibited by 1,542 trajectories as well as the remaining six trajectories that are designated noise. This final step in the data-driven process typically requires on the order of 10^{-2} seconds on the specified computer. The total runtime for the example used throughout this chapter is 25.25 seconds.

Analyzing Figure 6.11 reveals that each cluster successfully captures trajectories with a similar geometry whereas distinct clusters separate trajectories with distinct geometries. For instance, clusters 20-21 produced in Step 5 are noticeably different from the trajectories in the larger clus-

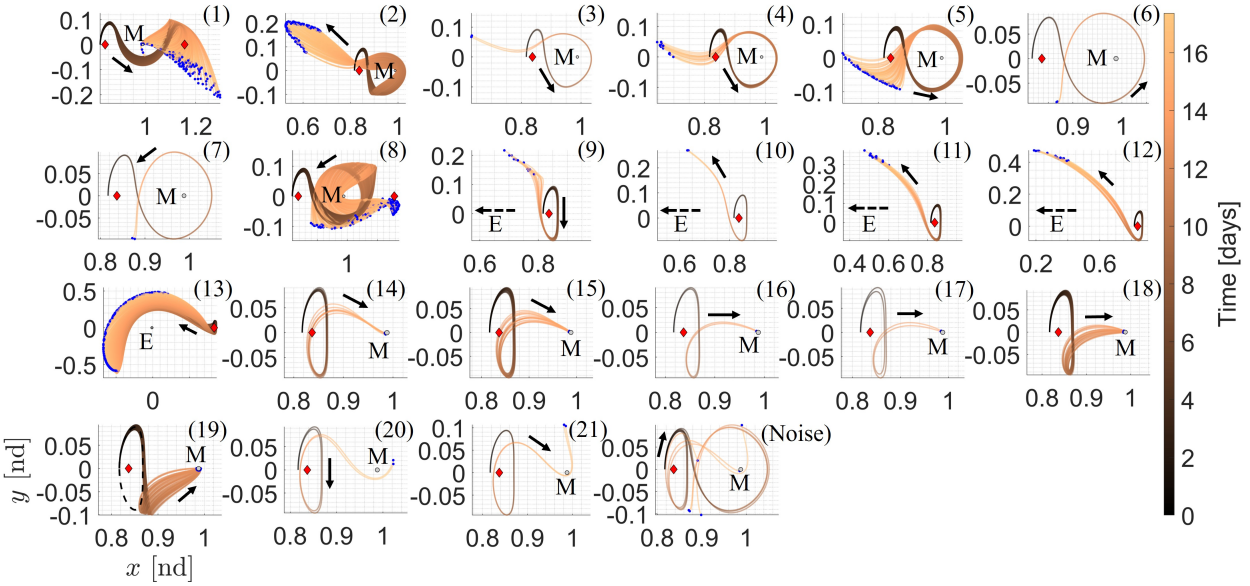

Figure 6.11: Final summary of trajectories generated for up to 17.3 days from a local point cloud of initial states along the selected L_1 Lyapunov orbit.

ters. Other small clusters added in this step resemble existing larger clusters but either possess a sufficiently different shape or temporal evolution, or exist in a more localized region of high density. The remaining six trajectories that are designated as noise resemble the members of clusters 5, 8, 14, and 21 in the summary but do not lie in sufficiently dense regions to be joined with those groups. Accordingly, the presented data-driven process automatically summarizes the generated set of trajectories via detailed groups from localized regions of high density. This summary can extract key motion types, reduce the complexity of manually analyzing a set of trajectories, or potentially enable automation in support of path planning or orbit prediction tasks.

6.7 Summary of Governing Parameters and Decisions

A set of parameters and decisions are required to govern the data-driven categorization of a set of trajectories that begin from a local region of the phase space. Table 6.1 summarizes these parameters and decisions and lists their influence on the resulting summary. Alternative parameters or decisions can be made throughout the trajectory summarization and clustering process to better fit a desired application or increase the level of differentiation between trajectories.

Although the CR3BP is used to demonstrate the proof of concept in this research, it is expected that the clustering-based method presented in this chapter would be useful for categorizing trajectories generated in higher-fidelity models of a three-body system with little modification. For this proof of concept, the CR3BP offers a sufficiently diverse and complex solution space while supporting fast trajectory generation without reliance on additional parameters, such as an initial epoch or spacecraft properties, required by higher fidelity models. Furthermore, the presented approach does not include assumptions based on the selected system or dynamical model fidelity.

Table 6.1: Parameters and decisions governing the data-driven categorization process.

Parameter/Decision	**Step(s)**	**Influence on Categorization Process**
Step size between initial conditions	0	Influences the number of initial condition samples.
Cloud boundaries	0	Determines the size of the region of initial states centered on the reference state.
Integration time	1	Allows distinct geometries to emerge, if they exist. Can increase the required number of samples for sufficient representation of solution geometry.
Discrete sample selection	2	Defines placement of discrete samples along continuous trajectories. Influences fidelity and characteristics of the solution geometry that are captured by the selected states.
Number of discrete states, p	2	Influences fidelity of discrete representation of trajectories and dimension of feature vector.
Feature selection	2	Influences similarity assessment across all trajectories within the set and characteristics that will be reflected in the resulting clusters.
Clustering algorithm	3, 5	Influences the recovered clusters.
$N_{minCore}$	3, 5	Governs core distance calculation that is used to estimate density across the dataset.
$N_{minClust}$	3, 5	Specifies minimum number of members in a single cluster.
ϵ	3, 5	Defines minimum desired resolution for separating two types of trajectories.

Chapter 7

Tree-Based Initial Guess Construction

With the incorporation of cislunar space into operational territory, agile spacecraft will play a key role in successful cislunar operations by maneuvering and replanning trajectories in response to hazardous situations. However, rapidly designing a complex trajectory in a multi-body gravitational system often relies on an operator for low-level analysis and high-level decision making. Typically, the trajectory designer must construct initial guesses for the path and maneuvers and then correct the initial guesses to produce continuous, feasible trajectories. Furthermore, the chaotic dynamics of a multi-body gravitational system can make rapid trajectory design a challenging and computationally intensive problem.

To address the challenge of rapid and automatic initial guess construction in multi-body gravitational environments, this chapter details the hierarchical tree-based planning approach for automatically constructing initial guesses for spacecraft trajectories between two boundary conditions. The design process consists of two phases: 1) growing a forest of trees that each locally explore the phase space and 2) using the forest to generate initial guesses. To demonstrate the utility of the tree-based framework, initial guesses are corrected to recover continuous trajectories. Each step of the approach is demonstrated in this chapter for a planar transfer between L_1 and L_2 Lyapunov orbits at a Jacobi constant equal to 3.1556 in the Earth-Moon CR3BP. A wider variety of results appears in Chapter 9. Additionally, computational times are provided for each step using a computer with an AMD Ryzen 5 5600G with Radeon Graphics 4.20 GHz processor.

7.1 Phase 1: Grow a Forest of Trees

Inspired by the use of multi-tree planning algorithms in robotics [46, 67, 108, 124], the first phase of the tree-based planning approach to initial guess construction involves building a forest of trees to discretely summarize a segment of the continuous solution space. Individually, each tree is constructed as a directed graph that summarizes motion within a local region of the phase space. A forest, composed of multiple trees, collectively explores a broader region of the phase space. Because sequences of trees are used in this work to generate complex and geometrically diverse global paths, the use of a forest of trees, as opposed to one or two RRTs, mitigates the risk of recovering short but maneuver-inefficient paths between two boundary conditions. Furthermore, this approach supports constructing complex paths that reach a target state without the use of predefined waypoints or complex objective functions. Building the forest of trees consists of two steps: i) sampling the tree roots and ii) growing the trees with branches that satisfy the dynamics in the CR3BP.

7.1.1 Sampling Tree Roots

The tree roots are located at position vectors that are sampled from a uniform grid in the configuration space. Inspired by the Kinodynamic Motion Planning by Interior-Exterior Cell Exploration [39] planner, the environment is represented by a square-cell grid with a specified cell height u_{root}. Then, the position vectors of samples are located at unique cell vertices to uniformly distribute samples within the configuration space of the environment, excluding its boundary and the regions within the zero velocity curves. Starting from each boundary condition, samples are also placed along each i-th periodic orbit at intervals of equal arclength, $s_{nat,i}$, calculated as

$$s_{nat,i} = \frac{s_{arcL,i}}{\lfloor s_{arcL,i}/s_{branch} \rfloor} \tag{7.1}$$

where $s_{arcL,i}$ is the total arclength of the i-th periodic orbit and s_{branch} is the arclength of tree branches that is defined in the next step of Phase 1, as described in Section 7.1.2. This root placement strategy supports generating transfers that flow away from or towards the periodic

orbits that define the boundary conditions [53].

For each of the sampled position vectors that do not lie along a periodic orbit, m velocity unit vectors are selected. In a planar environment, m unit vectors are uniformly distributed in the xy-plane in the range $[0°, 360°)$ to possess an equal angular separation. In the spatial environment, the near-uniform distribution of unit velocity vectors is achieved by first leveraging spiral sampling with the Golden ratio to place points on the surface of a unit sphere [14]. Then, a unit velocity vector originates at the center of the sphere and terminates at a point on the sphere's surface.

Each unit vector is multiplied by the speed associated with a specific value of the Jacobi constant. For transfers connecting periodic orbits at different energy levels, the Jacobi constant is randomly selected from the interval $[C_{J1}, C_{J2}]$ where C_{J1} and C_{J2} are defined as the Jacobi constants of the initial and final periodic orbits, respectively. Each value of the Jacobi constant is selected using MATLAB's *rand* function that draws random scalar values from the uniform distribution in the interval $(0, 1)$ [68]. For transfers spanning a single energy level, the Jacobi constant is equal to the desired Jacobi constant of the selected scenario.

Each unique combination of position and velocity vectors defines the tree root. For roots that do not lie along a periodic orbit, m tree roots are located at the same position but with distinct velocity vectors. However, if the root lies along a periodic orbit, the sampled velocity vector is pointed in the direction of natural motion to avoid abrupt departures or arrivals. This process for sampling the set of tree roots, $\mathcal{R}$, is summarized by the flowchart in Figure 7.1. Each tree root defines the first node in the tree.

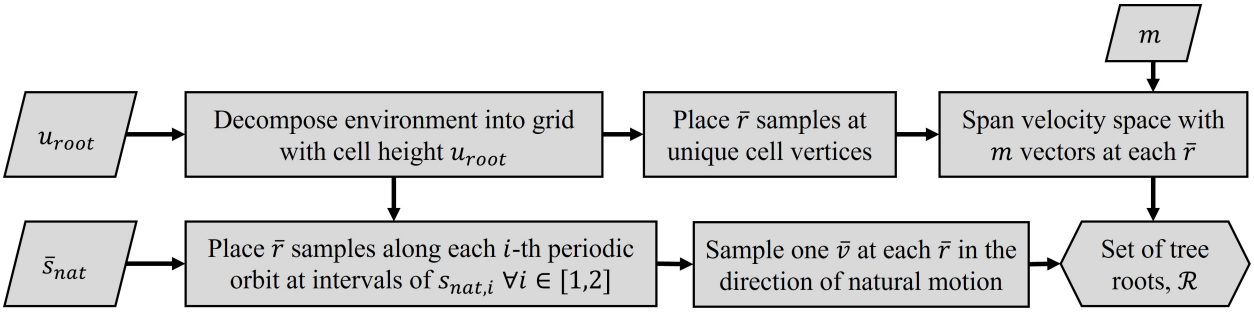

Figure 7.1: Overview of the process for sampling tree roots in Phase 1.

To demonstrate this sampling approach, consider the construction of an initial guess for a planar transfer between a 12.24-day L_1 Lyapunov orbit and a 14.79-day L_2 Lyapunov orbit at a Jacobi constant equal to 3.1556 in the Earth-Moon CR3BP. Figure 7.2 depicts a planar environment where the Moon appears as a gray circle scaled to possess a radius of 1,738.0 km [16], red diamonds locate L_1 and L_2, and the dark gray regions denote the forbidden regions at $C_J = 3.1556$. The depicted region of the system is decomposed into cells with $u_{root} = 0.025$ nondimensional units. Then, $m = 8$ velocity vectors are defined at each position vector such that each root possesses an equal angular separation of 45°. In addition, seven roots, including the boundary conditions, are evenly distributed in arclength along each periodic orbit. These values of the governing parameters produce a total of 1,030 roots distributed throughout the environment. In Figure 7.2, the position of each root on the grid is shown with a gray circle. For clarity, the roots along each periodic orbit are shown with red except for those located at the boundary conditions, which are displayed in purple. Blue arrows correspond to velocity vector directions associated with positions sampled on the grid

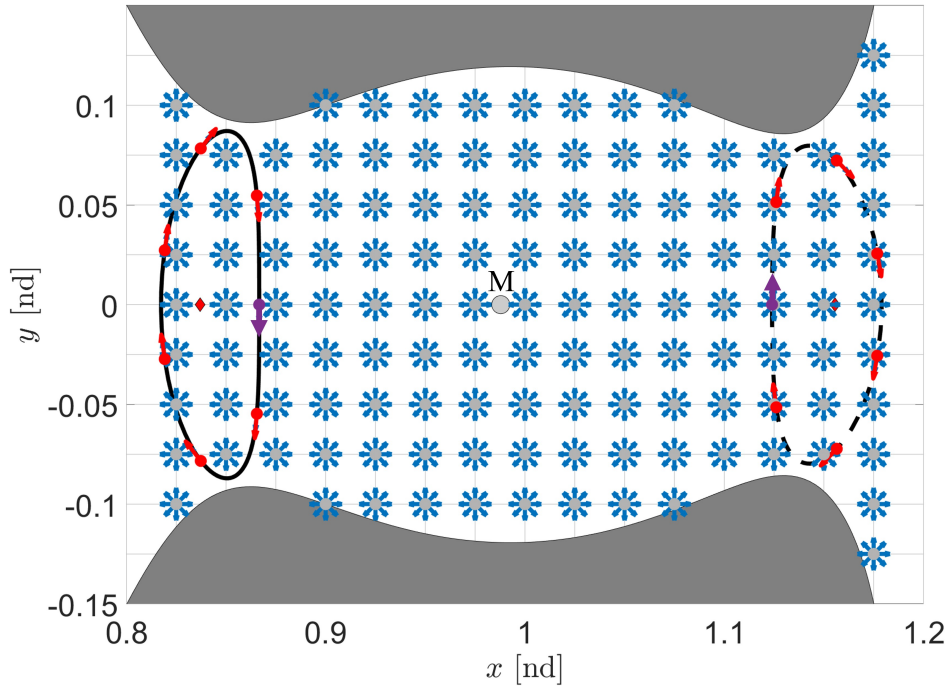

Figure 7.2: Tree roots for building a forest to support construction of a planar path between L_1 and L_2 Lyapunov orbits at $C_J = 3.1556$: filled circles denote the positions of the roots whereas arrows denote the direction of the velocity vectors at each root.

whereas red or purple arrows show the velocity vector directions for positions sampled along the periodic orbits. In this scenario, the initial state is located along the L_1 Lyapunov orbit at $\bar{x}_{init} = [0.866634949946303, 0, 0, 0, -0.210056789639986, 0]^T$ nondimensional units and the target state is located along the L_2 Lyapunov orbit at $\bar{x}_{targ} = [1.12398465047742, 0, 0, 0, 0.158922217869289, 0]^T$ nondimensional units. Sampling these roots takes on the order of 10^{-1} seconds on the specified computer.

7.1.2 Growing Trees

Grown from a root $\bar{x}_{root}$, each tree randomly explores a local region of the environment while governed by the dynamics in the CR3BP using a variation of the traditional RRT algorithm. In this work, a tree is iteratively extended by randomly selecting a growth node from the existing nodes in the tree. The velocity vector at the selected node is randomly modified multiple times to produce a set of new states with the same position vector as the growth nodes. These new states are propagated to generate small arcs in the dynamical environment, producing edges, or branches, in the tree that connect the growth node to a set of new nodes at the final states of the arcs. At each step, the pseudorandom number generator in MATLAB's *randi* [68] function is used to randomly select the value of governing parameters. The process for growing the i-th tree, τ_i, in the forest, $\mathcal{F}$, is summarized in the flowchart in Figure 7.3 and described in detail throughout the next few paragraphs.

The number of branches to extend from each growth node is randomly sampled. On the first iteration of growing the tree, the tree root is the growth node and the number of branches, n_{branch}, is sampled in $(0, K]$ where K is a user-defined constant less than or equal to the desired number of nodes in a tree, n_τ. At least one branch must grow to initiate tree growth; therefore, zero is omitted on the first iteration. For all successive iterations, n_{branch} is resampled in $[0, K]$ to mimic the traditional RRT algorithm built to promote efficient exploration of the environment.

The initial states of the n_{branch} branches emanating from a selected growth node possess the same position vector but unique velocity vectors. First, angles are randomly selected from within a

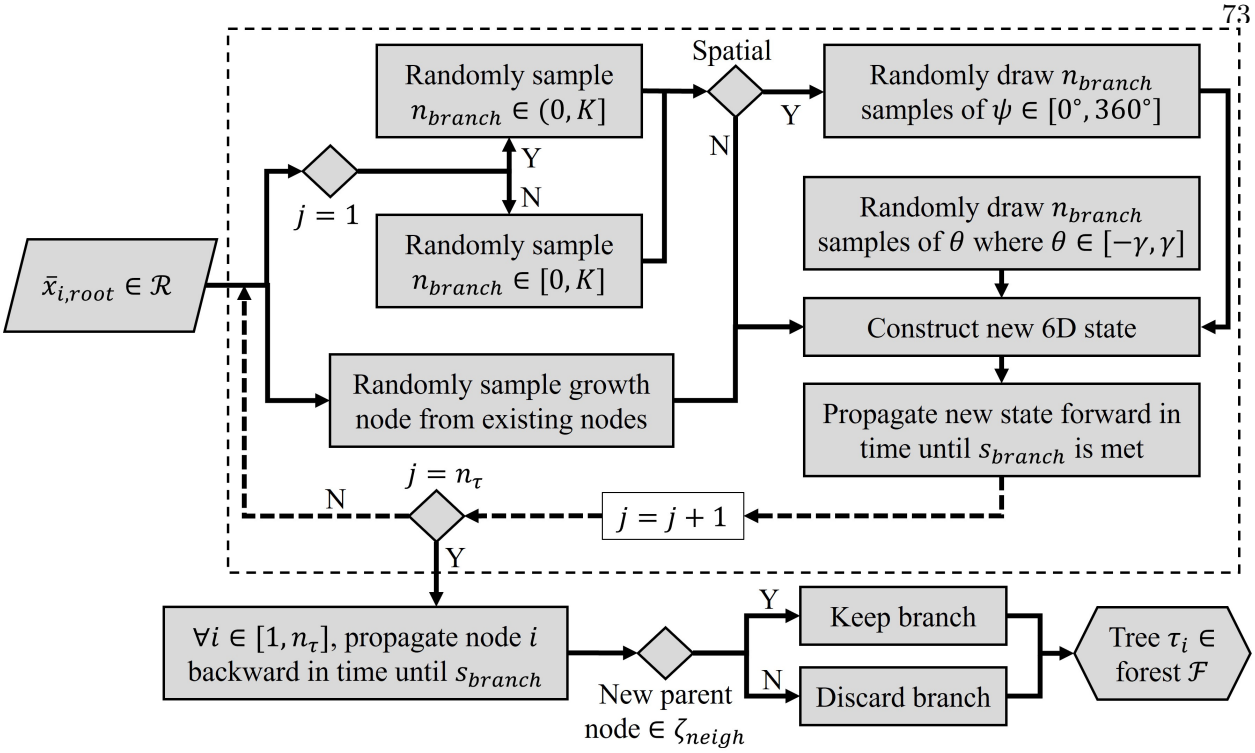

Figure 7.3: Overview of the process for growing the i-th tree in a forest in Phase 1.

specified range. Then, rotation matrices are used to calculate the new velocity unit vector that lies within a specified maximum angle from the velocity vector of the growth node. This unit vector is multiplied by the speed that would produce the same value of the Jacobi constant as all other nodes in the current tree. Combining the position vector of the selected growth node with the new velocity vector constructs an initial state for each of the n_{branch} branches. This approach to forest construction enables each tree to explore its local environment by balancing random exploration with limiting abrupt changes in velocity direction. These characteristics promote efficient tree growth and have been observed to support identifying high-quality initial guesses for maneuver-efficient paths using the forest.

In a planar transfer scenario, n_{branch} angles are sampled. These angles, $\bar{\theta} = [\theta_1, ..., \theta_{n_{branch}}]$, are randomly sampled within the range $[-\gamma, \gamma]$ where $\gamma = \frac{360°}{2m}$. Then, the velocity unit vector of each of the n_{branch} branches emanating from the growth node is calculated via a rotation by each

angle in $\bar{\theta}$ within the xy-plane. Mathematically, the i-th velocity vector, $\bar{v}_{rand,i}$, is calculated as

$$\bar{v}_{rand,i} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \bar{v}_{grow} \tag{7.2}$$

where $\bar{v}_{grow}$ is the growth node's 3×1 velocity vector.

For a spatial scenario, two sets of rotation angles are sampled and defined using a set of axes $\hat{i}\hat{j}\hat{k}$ local to each velocity vector. The $\hat{k}$-axis lies along the cone's axis of symmetry and the $\hat{i}\hat{j}$-plane coincides with the base of the cone of a constant radius. Then, $\bar{\theta}$ is sampled as defined in the planar case and a second set of angles, $\bar{\psi} = [\psi_1, ..., , \psi_{n_{branch}}] \in [0°, 360°]$, is also randomly sampled. Using these angles and a unit velocity vector aligned with the $\hat{k}$-axis, $\hat{v}_{local}$, the i-th randomly sampled velocity vector is calculated as

$$\bar{v}_{rand,i} = v_{CJ} \boldsymbol{R}_{\hat{z}}(\xi) \boldsymbol{R}_{\hat{y}}(90 - \beta) \boldsymbol{R}_{\hat{k}}(\psi_i) \boldsymbol{R}_{\hat{j}}(\theta_i) \hat{v}_{local} \tag{7.3}$$

If $\bar{v}_{0,g} = [v_{x0,g}, v_{y0,g}, v_{z0,g}]^T$ is the velocity vector at the growth node in the tree, a matrix $\boldsymbol{R}$ rotates a three-dimensional point about a specified axis, $\xi = \tan^{-1}(v_{y0,g}/v_{x0,g})$, $\beta = \tan^{-1}(v_{z0,g}/v_{x0,g})$, and v_{CJ} is the speed that produces the same value of the Jacobi constant as all other nodes in the current tree.

A spatial environment requires packing cones into a sphere to span the velocity space; therefore, the definition of γ provided for the planar scenario is not used. Rather, spherical coding, or the packing of points on a sphere, must be employed [15, 76]. This work uses the putative optimal spherical code by Kottwitz [15] for a three-dimensional sphere to determine that $m = 21$ such that $\gamma \approx 22.5°$ for consistency with the planar scenario.

Each new initial state is used to generate an arc that produces a new child node that is connected to the growth node via a branch. Each initial state is propagated forward in time in the CR3BP for a specified constant arclength s_{branch} using a Runge-Kutta 8th/9th order method in C++ via the GNU Scientific Library [1] with relative and absolute tolerances of 10^{-13} and 10^{-14}, respectively. However, propagation terminates early if the branch impacts a primary or crosses a

predefined boundary. As noted by Bruchko and Bosanac [53] when applying kinodynamic sampling to initial guess construction, using the arclength as opposed to the integration time for branch definition reduces the influence of dynamical sensitivity on the number and length of branches throughout the phase space. The branch length is selected to be sufficiently long to capture natural dynamics yet short enough to reasonably allow connections between branches. For each root that lies along a periodic orbit, one natural arc is also generated for an arclength of $s_{nat,i}$ to promote connectivity between trees along the i-th periodic orbit, enabling up to one full revolution about the orbits at the beginning and end of the transfer. The state at the end of each branch is a new child node.

Directional branches are initially added via forward propagation until a tree obtains a predefined number of nodes, n_τ. To promote dynamical flow and exploration, additional branches are then added by propagating nodes backward in time that possess at least one out-degree. If the parent node of the new branch lies within a neighborhood, ζ_{neigh}, in configuration space of the existing parents of the back-propagated node, the branch is considered redundant and not added to the tree. Trees rooted along the final periodic orbit are grown using the opposite process: branches are added via backward then forward propagation to promote flow onto the final periodic orbit. Trees rooted at the initial and target states do not have additional branches added via backward and forward propagation, respectively.

Roots with position vectors that are sufficiently close to the Moon and a velocity vector in the direction of the Moon are removed if they do not produce valid branches within a specified number of growth iterations. During tree growth, if a branch impacts the Moon, the branch is discarded. After n_τ failed propagation attempts, the root and its associated tree are removed from the forest.

To demonstrate the tree generation process, consider the L_1 to L_2 Lyapunov orbit transfer scenario used throughout this chapter. Figure 7.4 displays five trees in blue, magenta, black, red, and purple constructed with $n_\tau = 50$ nodes, $s_{branch} = 2u_{root}$ nondimensional units, $\gamma = 22.5°$, $K = 5$ nodes, and $\zeta_{neigh} = 5 \times 10^{-3}$ nondimensional units. The roots of each tree are located

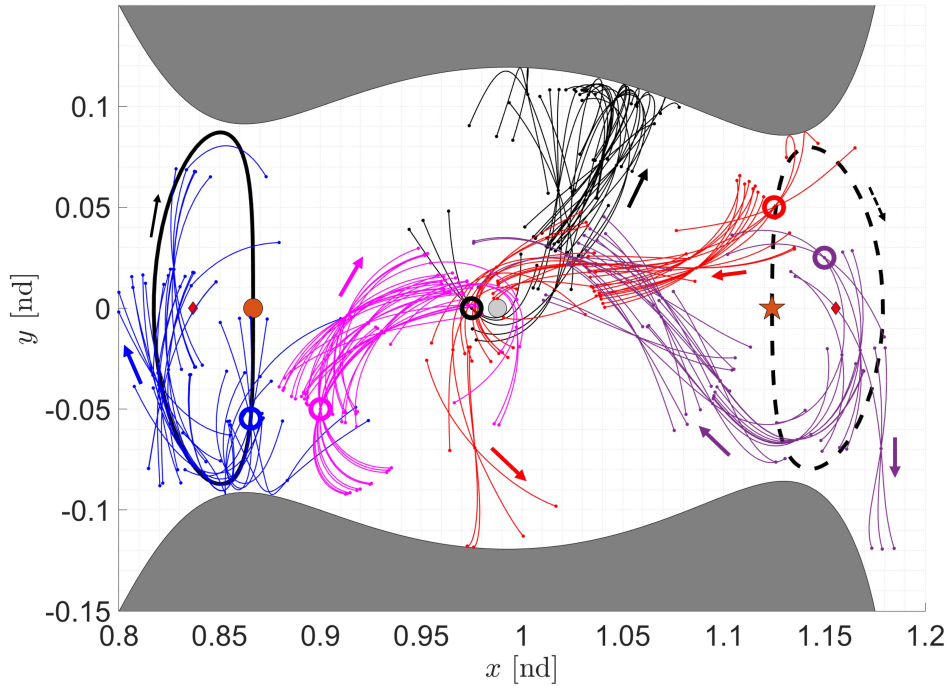

Figure 7.4: A subset of five trees, each depicted with a different color, from the forest grown to support planar path construction between L_1 and L_2 Lyapunov orbits at $C_J = 3.1556$ and satisfy the dynamics in the CR3BP.

at the centers of the bold circles and arrows denote the direction of motion. Additionally, the initial and target orbits are displayed with solid and dashed thick black lines, respectively, while the orange circle and star locate the initial and target states. In the selected scenario, four tree roots are removed due to n_τ propagation attempts through the Moon; therefore, the final forest consists of 1,026 trees grown from all but four of the roots in Figure 7.2. On the specified computer, constructing the forest for this scenario typically requires about thirty seconds when computed in parallel with four cores. The trees depicted in Figure 7.4 smoothly capture the flow in the CR3BP while exploring local regions of the phase space. In the full forest, seven additional trees are located at the position vectors associated with each of the depicted roots to span the velocity space at these locations. This approach to tree growth enables generation of a diverse set of trees that explore the phase space yet limit abrupt changes in velocity direction within each tree. Moreover, the trees rooted on the grid may be reused when reversing or changing boundary conditions.

7.2 Phase 2: Forest-Guided Initial Guess Construction

In the second phase of the hierarchical tree-based planning approach to trajectory generation, a path between the boundary conditions is constructed using the forest. A path is constructed as a sequence of nodes and branches, sampled from one or more trees. This phase exploits the exploration conducted in the first phase to construct an initial guess by following three steps: i) locating connections between trees in the forest; ii) identifying connected tree sequences; and iii) using the tree sequences to generate initial guesses that connect a set of boundary conditions. This approach is implemented using two hierarchical graph structures to reduce the computational burden of discovering paths of various geometries and eliminate the need to encode an objective function that guides path generation between boundary conditions.

7.2.1 Locating Connections Between Trees

To support searching the forest for a path, connections between trees are identified. In this work, pairs of nodes that belong to distinct trees are connected if all three of the following conditions are satisfied:

(1) Their position vectors lie within a specified distance ζ_{tol}. The discontinuity, ζ_{tol}, is selected to support identifying nodes that are sufficiently close in the configuration space to promote a successful corrections process while supporting recovery of a path in the forest that connects the boundary conditions.

(2) The angle between their velocity vectors is less than or equal to a specified threshold, selected as 2γ, i.e., the diameter of the velocity cone used to grow the trees in Phase 1. This criterion avoids abrupt changes in velocity direction in the resulting initial guesses without over-constraining the connections.

(3) The second node in the pair possesses at least one out-degree, indicating that the associated branch extends forward in time to support traversing the forest.

At this step, each tree may be connected to multiple trees at one or more locations. To improve the computational efficiency of this step, a pair of trees is analyzed to identify connections if they 1) overlap in the configuration space or 2) are within a distance of ζ_{tol} at their closest point. This entire process is summarized in the flowchart in Figure 7.5. Locating the connections in the forest for the example used throughout this section typically requires about 3.5 minutes on the specified computer when computed in parallel with four cores.

To demonstrate the identification of tree connections, consider a pair of trees that exist in the forest generated in Phase 1 for the selected planar example. Figure 7.6a) presents an example of two trees with five connections, denoted with red nodes, where $\zeta_{tol} = 5 \times 10^{-3}$ nondimensional units. In the zoomed-in view in Figure 7.6b), the connected nodes are indicated by dashed red arrows that capture directionality of the connections, i.e., from branches along the blue tree to branches along the black tree.

7.2.2 Identifying Connected Tree Sequences

To reduce the computational burden of searching for paths through a graph of the connected trees, a hierarchical approach is employed. First, a tree sequence graph is constructed to generate a set of tree sequences. Then, the tree sequences are used to construct a subgraph that is searched to generate one or more paths.

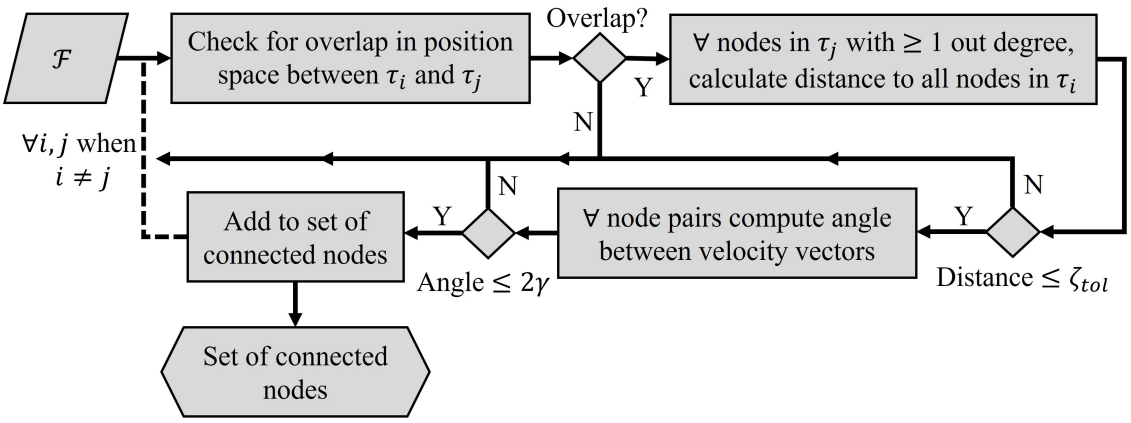

Figure 7.5: Overview of the process for identifying connections between trees in Phase 2.

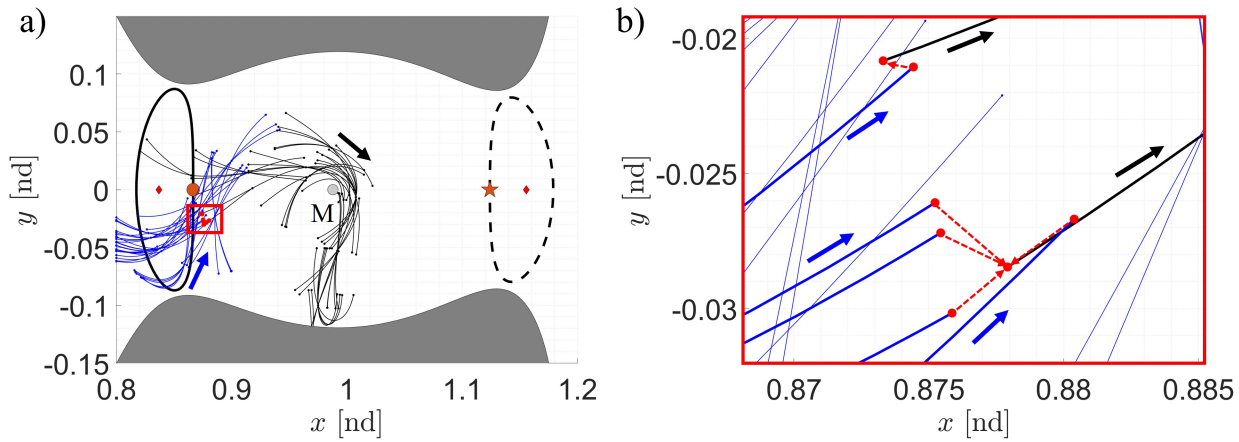

Figure 7.6: a) Two trees connected by five sets of nodes, shown in red. b) The five connections between the blue and black trees.

A tree sequence graph is constructed as a weighted, directed graph. Specifically, each vertex of the graph represents an entire tree and the edges represent connections between trees. Figure 7.7 conceptually depicts a tree sequence graph where τ_s and τ_f are the trees rooted at the initial and target states, respectively, and seven trees are rooted along the periodic orbits. The trees along the initial and final orbits are always connected, as shown with solid arrows, in a direction that is consistent with the flow along the orbit. These trees may also connect to other trees in the forest in any direction, indicated by dashed, bidirectional arrows.

Each edge weight is equal to the minimum value of the relative velocity between all pairs of

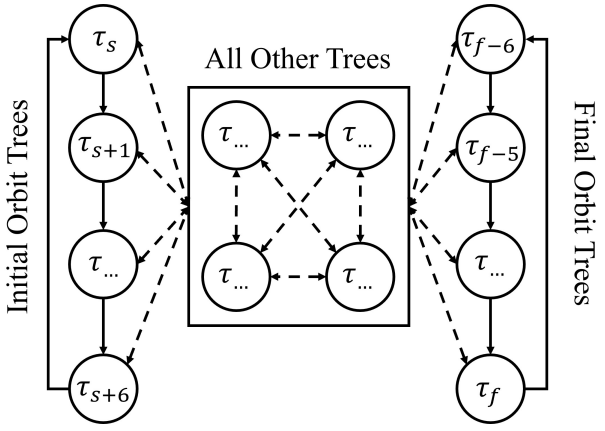

Figure 7.7: Conceptual depiction of a tree sequence graph.

connected nodes within the associated two trees. This edge weight, $w_{j,l,seq}$, between nodes j and l is calculated as

$$w_{j,l,seq} = \min \left(\frac{\|\bar{v}_{i,out} - \bar{v}_{i,in}\|}{\|\bar{v}_{i,out}\|} \right) \forall i \tag{7.4}$$

where $\bar{v}_{i,out}$ and $\bar{v}_{i,in}$ are the velocities of the i-th connected pair of outgoing and incoming nodes, respectively, from trees j and l. Consistent with the approach used by Bruchko and Bosanac when applying kinodynamic sampling to initial guess construction [53], using this weight encourages the recovery of smooth paths while avoiding biasing a search algorithm toward recovering solutions that maintain a lower speed, e.g., by avoiding close passes to the Moon.

The high-level tree sequence graph is searched to produce a sequence of connected trees from the initial state to the target state. The first tree sequence is identified to minimize the cumulative edge weight, i.e., an estimate of the total smoothness at the connections between trees. Additional tree sequences that produce the next $k-1$ lowest cumulative edge weights are also identified to support the recovery of a variety of geometrically-distinct initial guesses. Although a variety of graph search algorithms exist, this research leverages A* [83] and Yen's algorithm [50].

To recover tree sequences that produce paths connecting the boundary conditions in this work, information regarding the traversability of the trees comprising a sequence is incorporated into the graph search. The information contained in the tree sequence graph does not provide insight into whether a path exists between connected trees, only if trees are connected. Applying a graph search algorithm to the high-level tree sequence graph may lead to tree sequences that, when searched, do not produce paths connecting the boundary conditions. For example, consider if a current tree τ_i is connected to a previous tree τ_{i-1} through its j-th branch, b_{j,τ_i}, and to the next tree τ_{i+1} via its a-th branch, b_{a,τ_i}. A path connecting $\tau_{i-1} - \tau_i - \tau_{i+1}$ exists only if b_{a,τ_i} is reachable from b_{j,τ_i}. However, the trees grown in Phase 1 are not fully traversable, meaning that not all branches can be reached from any other branch within the same tree. Therefore, a path might not exist within the tree sequence $\tau_{i-1} - \tau_i - \tau_{i+1}$.

The tree sequence graph is searched via a modification of Yen's algorithm that is labeled in

this dissertation as a k-smoothest search algorithm. The goal of this search algorithm is to generate the k smoothest, yet diverse, sequences of traversable trees from which an array of geometrically diverse paths can be extracted. This k-smoothest search algorithm, inspired by the search algorithm presented by Miceli and Bosanac [33], uses both A* and Yen's algorithms [50, 83].

7.2.2.1 k-Smoothest Search Algorithm

The k-smoothest search algorithm presented in this section, implemented by Giuliana E. Miceli, addresses the challenge of directly applying the A* and Yen's search algorithms to the computation of tree sequences. The new algorithm produces a variety of traversable tree sequences from which an array of geometrically-diverse paths can be extracted. The algorithm first searches for one sequence that connects the trees rooted at the boundary conditions, and then searches for multiple diverse sequences connecting these trees. This process produces k diverse sequences of trees that can be searched to obtain initial guesses for spacecraft paths connecting the boundary conditions.

First, the k-smoothest algorithm employs a variation of A* to recover the smoothest, traversable tree sequence from a graph. The pseudocode for this approach is presented in Algorithm 1. Note that the queue size and tree sequence length may be limited to improve the memory requirements of the search algorithm and reduce computational time. The cost associated with every neighboring vertex representing a tree is computed as $c(\tau_i, \tau_{i+1}) = g(\tau_i, \tau_{i+1}) + h(\tau_{i+1}, \tau_f)$, where $g(\tau_i, \tau_{i+1})$ is the cost to traverse the graph from the current tree τ_i to neighboring tree τ_{i+1} and the heuristic, $h(\tau_{i+1}, \tau_f)$, is defined as the true cost to traverse the graph from the neighboring tree to the tree rooted at the target state, τ_f. For each incomplete path, the cost associated with it is $c(\tau_s) = g(\tau_s) + h(\tau_{i+1}, \tau_f)$, where $g(\tau_s)$ is the sum of the edge weights from the start vertex, τ_s, to the current vertex τ_i. At each search iteration, reachable neighboring trees are identified to ensure the sequence is traversable from the current tree to the neighboring tree in line 7 of Algorithm 1. This path is identified by analyzing nodes within each tree. The indices saved in the queue are used to identify the node of the current tree, τ_i, in the set of nodes, N_{IN}, that connect to the previous

Algorithm 1 Modified A* Search Algorithm for Tree Sequence Graph

Initialize a queue at the start tree. Each queue entry contains:
- A sequence of trees (an incomplete sequence),
- The current tree (last tree in the sequence),
- $c(\tau_s)$ and $g(\tau_s)$,
- Indices of the incoming tree nodes, N_{IN}, for the current tree.

while current tree $\neq$ target tree **do**
    Sort the queue in increasing order of $c(\tau_s)$.
    Select the sequence with the lowest $c(\tau_s)$.
    Identify all neighboring, connected trees to the current tree.
    Remove neighboring trees used in the incomplete sequence to avoid cycles.
    Identify the reachable neighboring trees.
    Calculate $c(\tau_s)$ for each reachable tree.
    Rank the reachable trees by increasing cost, $c(\tau_s)$.
    for n_{neigh} reachable trees **do**
        Create a new sequence by appending the reachable tree to the current sequence and computing $g(\tau_s)$ as the sum of the current sequence cost with the cost to the reachable tree.
        Add the new sequence to the queue with its associated values of $g(\tau_s)$, $c(\tau_s)$, and N_{IN} for the current tree.
    end for
end while

tree, τ_{i-1}. Then, the nodes of the current tree that connect to the next tree, τ_{i+1}, are defined as N_{OUT}. Within the current tree, these node sets are evaluated to determine if N_{OUT} are reachable from N_{IN}. If at least one node in N_{OUT} is reachable, then the index of the reachable node in N_{OUT} is saved in the queue and the neighboring tree is retained for the next evaluation. Otherwise, the tree is discarded as a neighbor.

After the smoothest tree sequence has been identified, the graph is searched for additional $k-1$ tree sequences. This process is implemented using Yen's algorithm [50] with A* [83]. A modification is included in this version of Yen's algorithm, which removes the use of a root path from the search over the subgraphs, promoting the recovery of more diverse tree sequences. The search of k-smoothest paths terminates once the number of complete sequences between list $\mathcal{A}$ and list $\mathcal{B}$ is greater than or equal to k. Leveraging the sequences obtained from list $\mathcal{A}$ and list $\mathcal{B}$ allows for faster recovery of k sequences while retaining diversity in the solution set, albeit at the expense of recovering suboptimal solutions.

To demonstrate the process of generating tree sequences, consider the planar L_1 to L_2 Lyapunov orbit transfer scenario presented within this chapter. Applying this graph search algorithm to the high-level tree sequence graph in this example, composed of 81,608 edges and 1,026 vertices, produces $k = 100$ smooth, diverse tree sequences in 48.5 minutes using the specified computer and parallel computing with four cores. As an example, consider the 68th best tree sequence, composed of seven trees, shown in Figure 7.8 with each tree plotted in its entirety.

7.2.3 Searching the Tree Sequences for Initial Guesses

The final step of Phase 2 focuses on recovering an initial guess for a trajectory from a tree sequence. For each tree sequence, a localized path graph is constructed. The vertices are defined as the nodes of every tree in the sequence. Edges are added between vertices associated with 1) nodes within a single tree that are connected by a branch and 2) nodes that are connected across distinct trees, as described in Section 7.2.1. This graph consists of weighted and directed edges

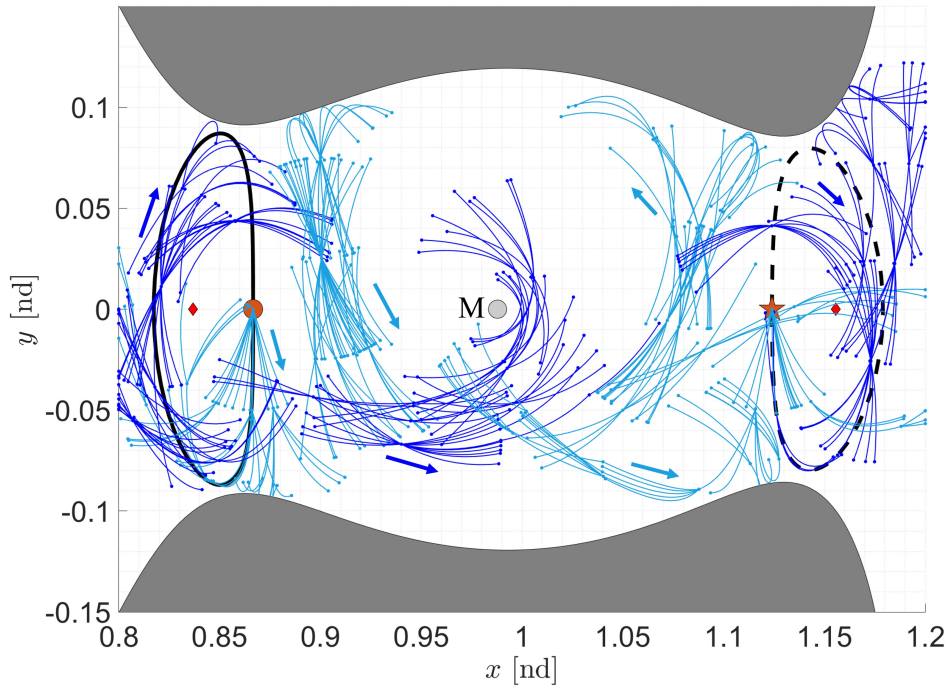

Figure 7.8: An example of a seven-tree sequence returned from the k-smoothest paths search algorithm that connects the trees rooted at the boundary conditions.

where the edge weight, $w_{j,l,path}$, between vertices j and l is defined similar to Equation 7.4 as

$$w_{j,l,path} = \frac{\|\bar{v}_j - \bar{v}_l\|}{\|\bar{v}_j\|} \tag{7.5}$$

These subgraphs are less complex than the high-level tree sequence graph. Therefore, Dijkstra's algorithm [29], or A* with no heuristic, is used to identify the shortest path in the graph. This path is composed of a sequence of vertices and edges that is converted to a sequence of nodes and branches to supply an initial guess for a trajectory. Constructing and searching one subgraph on the specified computer requires, on average, 0.3 seconds.

To demonstrate the process of generating an initial guess from a tree sequence, consider the example of connecting boundary conditions along L_1 and L_2 Lyapunov orbits in the Earth-Moon CR3BP. The tree sequence depicted in Figure 7.8 is used to construct a localized path graph that is composed of 493 vertices and 504 edges. This graph is searched to produce a coarse initial guess as displayed in Figure 7.9; in this new figure, the original tree sequence appears in the background with transparency. The last tree in the sequence does not leverage any branches; only a single state connects to the target state. Furthermore, this initial guess resembles the geometry of a known heteroclinic connection, shown in gray, at a nearby value of the Jacobi constant, $C_J = 3.1520$ [26]. This path is characterized by a cumulative velocity discontinuity equal to 687.90 m/s and a time of flight (TOF) equal to 26.12 days. Note, however, that because this initial guess is coarsely generated and discontinuous, the cumulative velocity discontinuity is not necessarily reflective of the sum of any impulsive maneuvers. Nevertheless, the recovery of this initial guess demonstrates that the tree-based initial guess construction method works as intended: to enable automatic generation of an initial guess that connects boundary conditions along periodic orbits while following the dynamics of the CR3BP.

7.2.4 Summary of Governing Parameters and Decisions

The hierarchical tree-based procedure for automatically generating initial guesses for spacecraft trajectories is governed by a set of parameters and decisions. Table 7.1 summarizes these

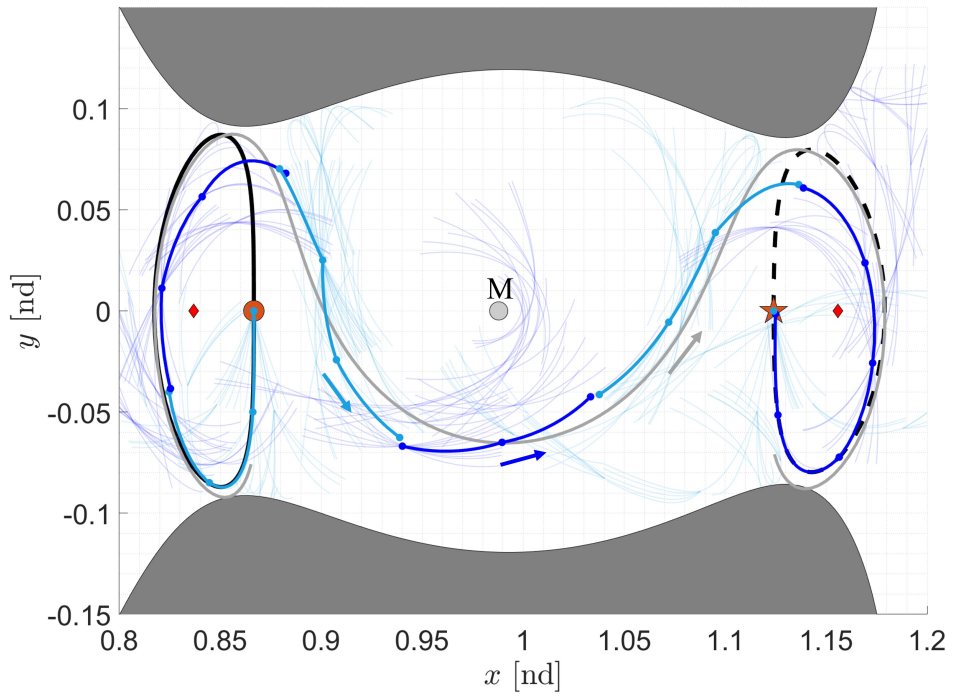

Figure 7.9: A coarse, zero-revolution initial guess connecting boundary conditions on L_1 and L_2 Lyapunov orbits at $C_J = 3.1556$ with a cumulative velocity discontinuity equal to 687.90 m/s and estimated time of flight equal to 26.12 days. A heteroclinic connection at $C_J = 3.1520$ is provided in gray for comparison.

governing parameters and decisions. Alternative selections for these parameters or decisions may be used across different applications to increase the flexibility of path recovery or alter the computational requirements.

7.3 Assessing Geometric Similarity

In this work, multiple geometrically-distinct initial guesses are generated from distinct tree sequences. To numerically assess the geometric similarity across initial guesses, a path similarity assessment is performed. First, the initial guesses are categorized by their total absolute curvature. Then, initial guesses within each curvature group are assessed for geometric similarity. This numerical assessment supports the observation, via visual inspection, that distinct tree sequences produce geometrically-distinct initial guesses.

Initial guesses generated from distinct tree sequences are categorized by their total absolute

Table 7.1: Parameters and decisions governing the hierarchical tree-based planning process.

Governing Parameter or Decision	**Influence on Computation and/or Results**	**Selected Value**
Grid size, u_{root}	Influences the number of trees in the forest.	0.025
Number of velocity vectors, m	Determines the size of the velocity cone, γ, and influences the number of trees in the forest. Smaller velocity cones produce smoother initial guesses.	Planar: 8, Spatial: 21
Maximum value for tree branch growth, K	Defines the maximum number of tree branches that can be added in a single growth iteration. Influences the span of the environment explored per tree.	5
Branch arclength, s_{branch}	Influences the span of the environment explored per tree, potential connectivity between trees, graph size, and number of arcs in each initial guess.	$2u_{root}$
Number of nodes in a tree, n_τ	Defines tree size, influences the span of the environment explored per tree, and localized path graph size.	50
Neighborhood for redundant branches, ζ_{neigh}	Defines tolerance for removing redundant tree branches, influences time required to search the local path graph.	5×10^{-3}
Position connection tolerance, ζ_{tol}	Defines tolerance in the configuration space for identifying connections between tree branches, influences the discontinuity in initial guesses and flexibility of path recovery.	5×10^{-3}
Velocity connection tolerance	Defines tolerance in the velocity space for identifying connections between tree branches, influences the discontinuity in initial guesses and flexibility of path recovery.	2γ
Number of smoothest paths, k	Defines the number of diverse tree sequences to extract from the graph.	Planar: 100, Spatial: 50
Number of neighbors for exploration, n_{neigh}	Determines the number of neighboring, reachable trees explored at each iteration of the k-smoothest graph search before arriving to the target vertex and influences the required computational time.	20
Graph edge weights	Influences the paths generated from each graph.	Equations 7.4, 7.5
Maximum path length	Sets the maximum length of a tree sequence returned from the graph search, influences computational time.	Planar: 20, Spatial: 100
Maximum queue size in A*	Influences memory requirements of search algorithm.	500

curvature, κ_{tot}, to simplify the analysis of the solution space. The total absolute curvature is defined as [113]

$$\kappa_{tot} = (t_0, t_f) = \int_{t_0}^{t_f} \kappa \, ds_{arcL} = \int_{t_0}^{t_f} \kappa \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2} \, dt \tag{7.6}$$

to capture the number of revolutions that the trajectory completes around an evolving center of curvature where κ is defined in Equation 6.2. To calculate the total absolute curvature of a discontinuous initial guess, κ_{tot} is calculated along each branch and then summed. Then, paths are sorted into approximate groups by intervals of 2π in κ_{tot}, approximating the number of revolutions along the path without the need for visual inspection.

The initial guesses within each curvature group are then compared to determine if the initial guesses possess sufficiently different geometries. A path similarity assessment is applied to a set of paths within the same curvature group and governed by the maximum angular difference between a pair of velocity unit vectors along two paths. Dynamic time warping (DTW) is used to compare time series data, particularly in the field of data mining [98, 114]. DTW may be used to align two time series of unequal length by warping the data to align features on a common time axis [68, 114]. Therefore, DTW is used to create paths of equal length by warping the position coordinates at each node using MATLAB's *dtw* function [68]. Then, the angle between the i-th pair of velocity unit vectors along a pair of warped initial guesses is calculated as $\Omega_i = \cos^{-1}(\hat{v}_{i,1} \cdot \hat{v}_{i,2})$. If the maximum angle between one pair of initial guesses, Ω_{max}, is greater than 4γ, the coarse paths are considered geometrically diverse. When this path similarity assessment is applied to the 100 initial guesses generated for the L_1 to L_2 Lyapunov orbit transfer design scenario used throughout this chapter, all initial guesses pass the similarity assessment.

Four additional coarse initial guesses are selected to demonstrate the geometric diversity across a set of initial guesses for spacecraft trajectories. Figure 7.10 presents the initial guesses with the fifth (dark blue), eighth (red), 18th (maroon), and 34th (light blue) smallest cumulative relative velocity discontinuities while performing a single revolution about the Moon between the selected L_1 and L_2 Lyapunov orbits. Despite occasionally sharing arcs, these initial guesses possess

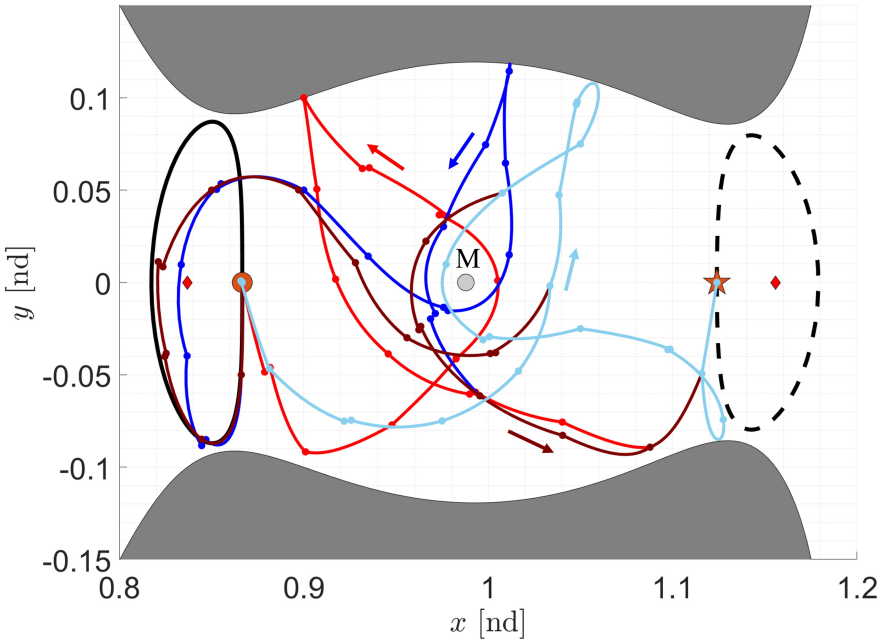

Figure 7.10: Four geometrically-distinct initial guesses connecting boundary conditions along L_1 and L_2 Lyapunov orbits at $C_J = 3.1556$.

distinct geometries in the lunar region and/or along the departure and arrival segments.

7.4 Recovering a Continuous Path

Continuous spacecraft trajectories are recovered from initial guesses using multiple-shooting for differential correction. First, impulsive maneuvers are manually placed at 1) the connections to the initial and final periodic orbits and 2) discontinuities between branches that exist near maxima in the curvature κ that tend to correspond to locations near apses relative to primary bodies or libration points. Because maneuver placement is often influenced by the application, mission constraints, or needs of a user, the selected strategy supports the proof of concept in this research; however, exploring alternative strategies or optimal maneuver placement is an interesting avenue for future work. To support successful recovery of a continuous transfer, one revolution about each i-th periodic orbit, consisting of natural arcs of arclength $s_{nat,i}$, is then appended to the beginning and end of the initial guess. Next, the initial guess is corrected to preserve its geometry while recovering a continuous trajectory; during corrections, the maneuvers shift.

Continuation is used to gradually prioritize reducing the total maneuver requirements over maintaining geometric resemblance to the initial guess, often substantially lowering the cumulative velocity discontinuities at locations where maneuvers occur. As opposed to optimization, continuation is used for maneuver reduction in this proof of concept due to its computational speed and generation of solutions that remain close to the last known solution after each continuation step. This approach produces a solution that may be close to, but not guaranteed to be, optimal; the use of local, constrained optimization is an interesting avenue for future work.

The initial guess in Figure 7.9 produces one solution that prioritizes geometric resemblance to the initial guess and one solution that prioritizes reducing the maneuver requirements. These transfers are presented in Figure 7.11: the initial guess is plotted in gray, the geometry-focused solution is displayed in purple with a total $\Delta v = 36.88$ m/s and a flight time of 28.03 days, and the energy-focused solution is depicted in blue with a total $\Delta v = 29.78$ m/s and a flight time equal

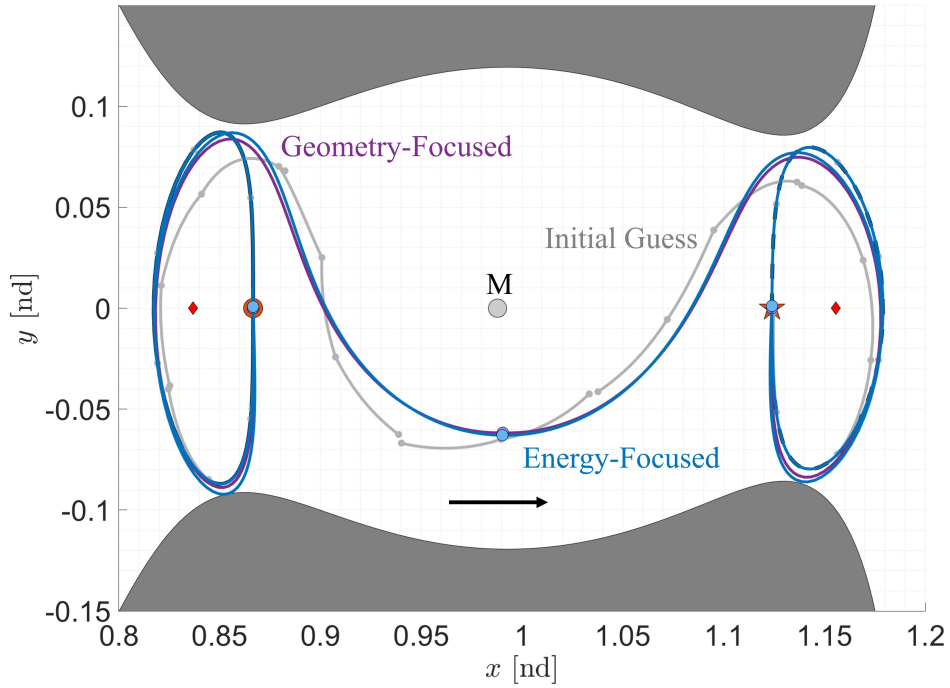

Figure 7.11: The evolution of a zero-revolution transfer connecting boundary conditions on L_1 and L_2 Lyapunov orbits at $C_J = 3.1556$: initial guess (gray), geometry-focused solution (purple) with Δv equal to 36.88 m/s and TOF equal to 28.03 days, and energy-focused solution (blue) with Δv equal to 29.78 m/s and TOF equal to 28.20 days.

to 28.20 days. The three impulsive maneuver locations are depicted by filled circles, colored to match their associated trajectory. For the corrected solutions, the time of flight of a trajectory is calculated as the time elapsed between the first and final maneuver locations, signifying departure from or arrival to the boundary conditions.

The solutions depicted in Figure 7.11 demonstrate the successful recovery of smooth, continuous spacecraft trajectories using continuation and possess geometries resembling heteroclinic connections at nearby energy levels. This example also demonstrates the value of the coarsely-designed initial guesses to rapidly generate meaningful, continuous transfers with substantially lower velocity discontinuity while retaining the geometry of the initial guess. An in-depth presentation and discussion of the continuous transfers recovered for the L_1 to L_1 Lyapunov orbit transfer scenario used throughout this chapter is presented in Chapter 9.

Chapter 8

Generate Collision-Free Trajectories

This chapter details a two-level, cell-based approach to collision identification to address the challenge of ensuring spacecraft trajectories are collision-free. In this work, a dynamic obstacle is defined as a corridor composed of continuous, geometrically-similar trajectories. These corridors are generated from a set of trajectories that begin within a local initial state set surrounding the estimated state of an object at its observation epoch. Note, however, that this initial state set does not currently encode probability information. Then, the data-driven categorization method presented in Chapter 6 is employed to categorize the trajectories associated with the initial state set by their spatiotemporal variations. The result is a set of obstacle corridors that satisfy the following characteristics: within each corridor, trajectories possess a similar geometry whereas trajectories in distinct corridors possess distinct geometries. A collision is defined to exist when a spacecraft path overlaps with at least one dynamic obstacle corridor in position and time. The goal of collision-checking in this research is to identify the spacecraft paths that avoid these corridors. The collision-checking approach is demonstrated for a planar transfer between L_1 and L_2 Lyapunov orbits at a Jacobi constant equal to 3.1556 in the Earth-Moon CR3BP. A wider variety of results appears in Chapter 9. Additionally, computational times are provided for each step using a computer with an AMD Ryzen 5 5600G with Radeon Graphics 4.20 GHz processor.

8.1 Dynamic Obstacle Maps

This work employs approximate cell decomposition to create dynamic obstacle maps for corridor avoidance. Each trajectory in a corridor is represented by a finite set of discrete states. Then, the environment is decomposed into a set of cells where the smallest cells conservatively cover the obstacle corridors.

The key components to creating a dynamic obstacle map include cell size and shape. In this work, the cells are defined as squares or cubes that, at the smallest resolution, possess a constant height of u_{min} where a collision margin of $\frac{u_{min}}{2}$ is guaranteed for any state along an object path that lies in the center of a cell. In this work, $u_{min} = 21$ km is selected such that the collision margin is equal 10.5 km; to support a proof of concept, this value is informed by a spacecraft's 3σ position uncertainty in cislunar space [17]. This design decision was informed by the Multimission Automated Deepspace Conjunction Assessment Process (MADCAP) developed by NASA's Jet Propulsion Laboratory for predicting and avoiding close approaches of space objects and conducting conjunction risk assessments [17]. According to MADCAP, spacecraft are predicted to be too close if they are within the trajectory's 3σ uncertainty estimates [17]. However, a constant cell size that is informed by a spacecraft's uncertainty does not reflect the growth of the uncertainty over time or variation based on location in cislunar space. A valuable avenue of future work includes constructing maps with a variable cell size to reflect this growth.

To guarantee a collision tolerance of $\frac{u_{min}}{2}$, buffer cells are added to the decomposition. These buffer cells are the same size and shape as the cells that approximate the dynamic obstacles and offer an additional layer around the cells covering the obstacles. This buffer is useful when two subsequent discrete states along a trajectory contained within an obstacle corridor lie just outside a cell but the continuous path connecting them passes through the cell. This cell is designated as a buffer cell, and is no longer mislabeled as collision-free.

To adequately represent each potential collision path, reduce the number of mislabeled cells, and decompose the environment without leaving any path segment in the obstacle corridor uncov-

ered by cells, the potential collision paths must be sufficiently summarized with states. A challenge with discretely representing the continuous obstacle corridors is selecting the number of discrete states to be small enough such that the corridors are sufficiently mapped with cells during decomposition while large enough to maintain computational efficiency [48]. In this work, geometric insight is used to determine that discretizing a potential collision path into states spaced equally in arclengths of s_{min} will sufficiently represent each path when s_{min} is calculated as

$$s_{min} = u_{min} - \delta_{min} \tag{8.1}$$

where u_{min} is the cell height and $\delta_{min} > (\sqrt{2}-1)u_{min}$. This arclength discretization also guarantees that any mislabeled cells, or cells containing segments of paths without discrete states, will contain path segments of lengths less than s_{min}. A limitation of this method is that the desired cell height is not guaranteed to be the exact size of the cells at the true minimum resolution; this resolution is dictated by the size of the environment. As long as the actual smallest cell size is greater than the desired minimum cell size, the guarantees described above remain true.

To reduce computational complexity and simplify the collision checking process, the dynamic obstacle corridors are decomposed piecewise in time for user-defined intervals. In this work, each map spans approximately two days. Additionally, the maps overlap in time to increase the accuracy of trajectory representation in each time interval since the continuous trajectories are coarsely represented with discrete states. Furthermore, this multi-map approach eliminates the need to associate multiple times with a single cell when a trajectory overlaps with itself and is represented for all time via a single map.

8.2 Corridor Avoidance

Corridor avoidance is performed in this work using a two-level collision detection algorithm in conjunction with dynamic obstacle maps. First, inspired by cell-based collision detection methods, a set of coarse dynamic obstacle maps is generated for specified time intervals. Then, lazy collision checking is used to identify a coarse collision region. If an overlap exists in position between the

obstacle corridor and the spacecraft trajectory, a fine collision detection step is employed. This two-level approach reduces the computation time of collision checking while retaining sufficient accuracy to identify collisions without requiring parameterization of the dynamic obstacle corridors. The two-level collision checking process for a single dynamic obstacle map is summarized in the flowchart in Figure 8.1 and described in more detail in the following paragraphs.

Coarse collision detection identifies a coarse region of the environment where the spacecraft path and obstacle corridor overlap in position space. The coarse obstacle maps are created with cells that are 100 times larger than the desired cell size, $u_{coarse} = 100u_{min}$. Accordingly, the trajectories in the obstacle corridor are represented with discrete states spaced equally in arclengths of s_{coarse}. Mathematically,

$$s_{coarse} = u_{coarse} - \delta_{coarse} \tag{8.2}$$

This coarse approximation reduces the computational requirements of the collision checking process,

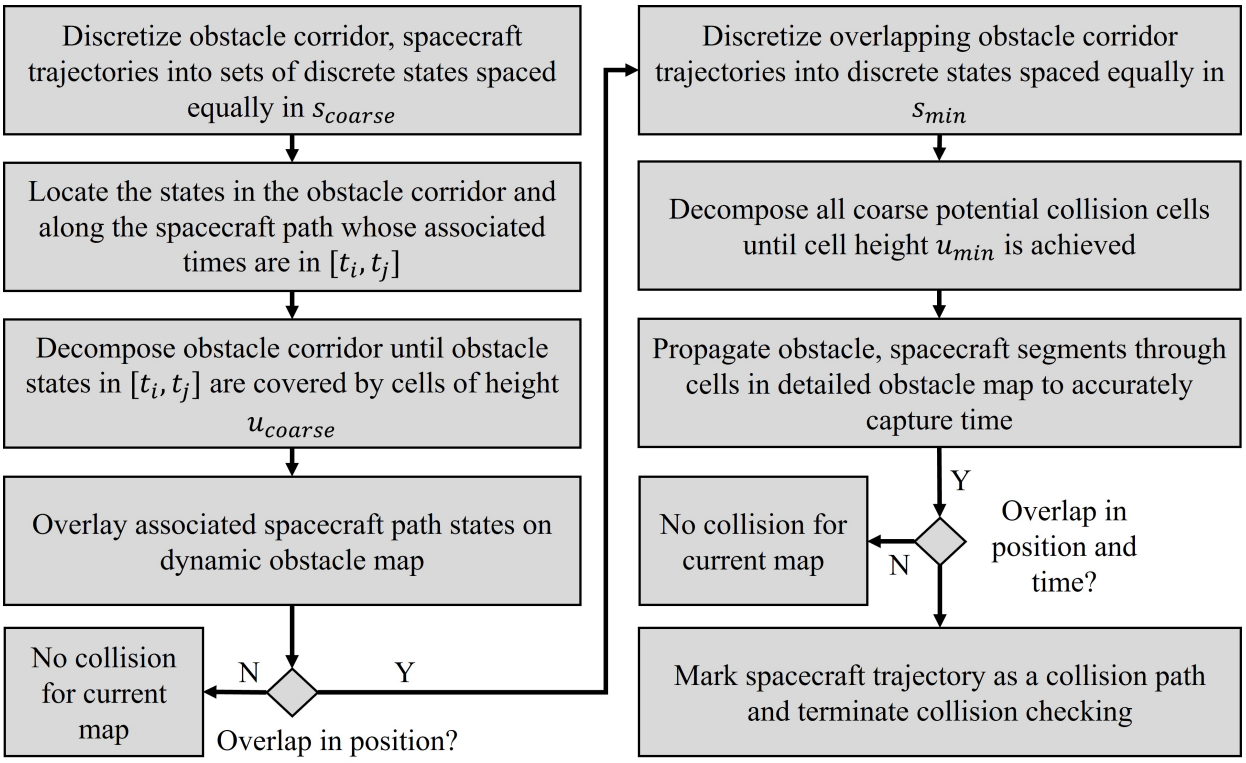

Figure 8.1: Overview of the two-level, cell-based corridor avoidance process for a single dynamic obstacle map.

particularly when no overlap is present between the spacecraft's trajectory and the obstacle corridor for a given time interval. The spacecraft path is represented using the same discretization method for the trajectories in the obstacle corridor to ensure sufficient representation of the continuous path when used in conjunction with a dynamic obstacle map.

To provide an example of constructing coarse collision maps piecewise in time, consider the energy-focused spacecraft trajectory recovered in Chapter 7 that connects the L_1 and L_2 Lyapunov orbits. Then, assume the state of an observed object at an epoch t_{obj} is located within a local neighborhood centered on $\bar{x}_{obj}(t_{obj}) = [1.02454653948, 0.0541769443650, 0, -0.380916147888, -0.110095105627, 0]^T$ nondimensional units along a distant prograde orbit (DPO) at $C_J = 3.1700$ in the Earth-Moon CR3BP. The local neighborhood possesses a radius of 10.5 km in position and 10.5 m/s in velocity. The potential future motions of this object have been grouped by their spatiotemporal variations using the data-driven categorization process presented in Chapter 6. The summary of the object's motion is presented in Figure 9.3 and supplies 18 unique dynamic obstacles.

Consider a scenario where the spacecraft is expected to begin its transfer 12.11 days before the object's observation epoch, t_{obj}. In this scenario, the 16th obstacle corridor is used to demonstrate the collision checking process. An overview of the selected scenario is provided in Figure 8.2 where the color gradients denote the elapsed time along each path, filled black circles locate maneuvers, arrows provide the directions of motion, a filled pink triangle denotes the location along the continuous spacecraft path at t_{obj}, and a filled pink square locates the location of the dynamic obstacle along its corridor at t_{obj}. Figures 8.3 - 8.4 display the 27 coarse obstacle maps constructed for this scenario where each subfigure presents a coarse collision map for the specified time interval, red circles denote the discrete states representing the obstacle corridor over the associated time interval, and square cells compose the obstacle map.

To determine if a potential collision exists, the discrete states representing a spacecraft path are added to a dynamic obstacle map. If at least one cell in the map contains one or more discrete states representing the spacecraft trajectory for the map's specified time interval, the path is marked as a potential collision trajectory. Then, the potential collision cells are coarsely bounded to identify

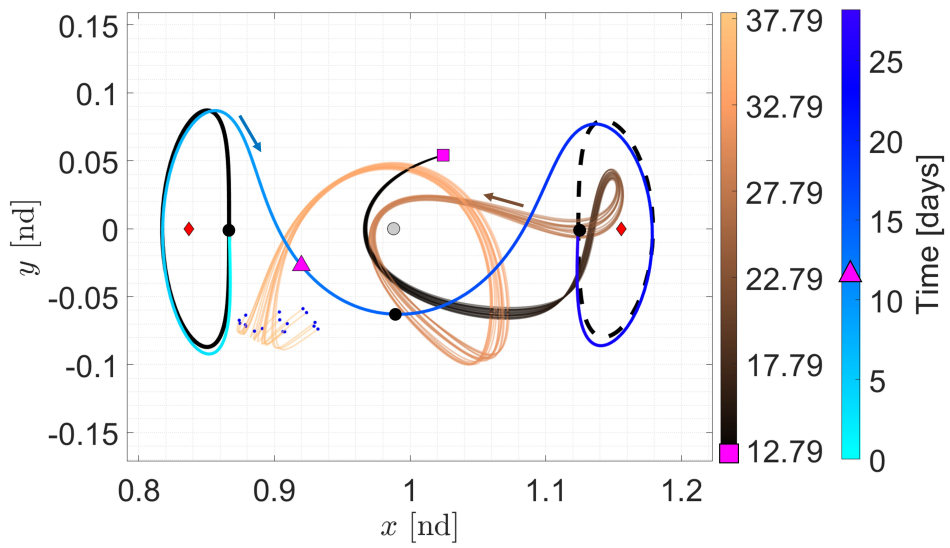

Figure 8.2: A continuous spacecraft path (blue) that connects L_1 and L_2 Lyapunov orbits is analyzed for collisions with a dynamic obstacle corridor (copper) that originates near the vicinity of a distant prograde orbit.

a potential collision region. Coarse collision checking is repeated for all segments corresponding to the predefined time ranges until either a collision is identified during fine collision checking or all segments have been analyzed for collisions.

Following the example outlined in this chapter, overlaying the spacecraft trajectory on the second coarse map in Figure 8.3 reveals an overlap in position and time. This overlap is depicted in Figure 8.5a) where arrows denote the directions of motion, the spacecraft trajectory is plotted in blue, and the trajectories within the obstacle corridor are colored with a dark brown to copper gradient to reflect the elapsed time. Figure 8.5b) magnifies the region bounded in teal in Figure 8.5a) and the black rectangle highlights the coarsely bounded collision region.

If a coarse collision region is identified, fine collision checking is employed. First, the trajectories in the obstacle corridor are discretized into arcs that each possess an arclength of s_{min}. Then, a detailed map is constructed by decomposing cells in the coarse collision region in parallel until the desired cell height, u_{min}, is met. All trajectories in the obstacle corridor that pass through a coarse cell are propagated forward in time while capturing locations where the trajectories pass through cells of height u_{min}. This propagation procedure is repeated for the segment of the spacecraft

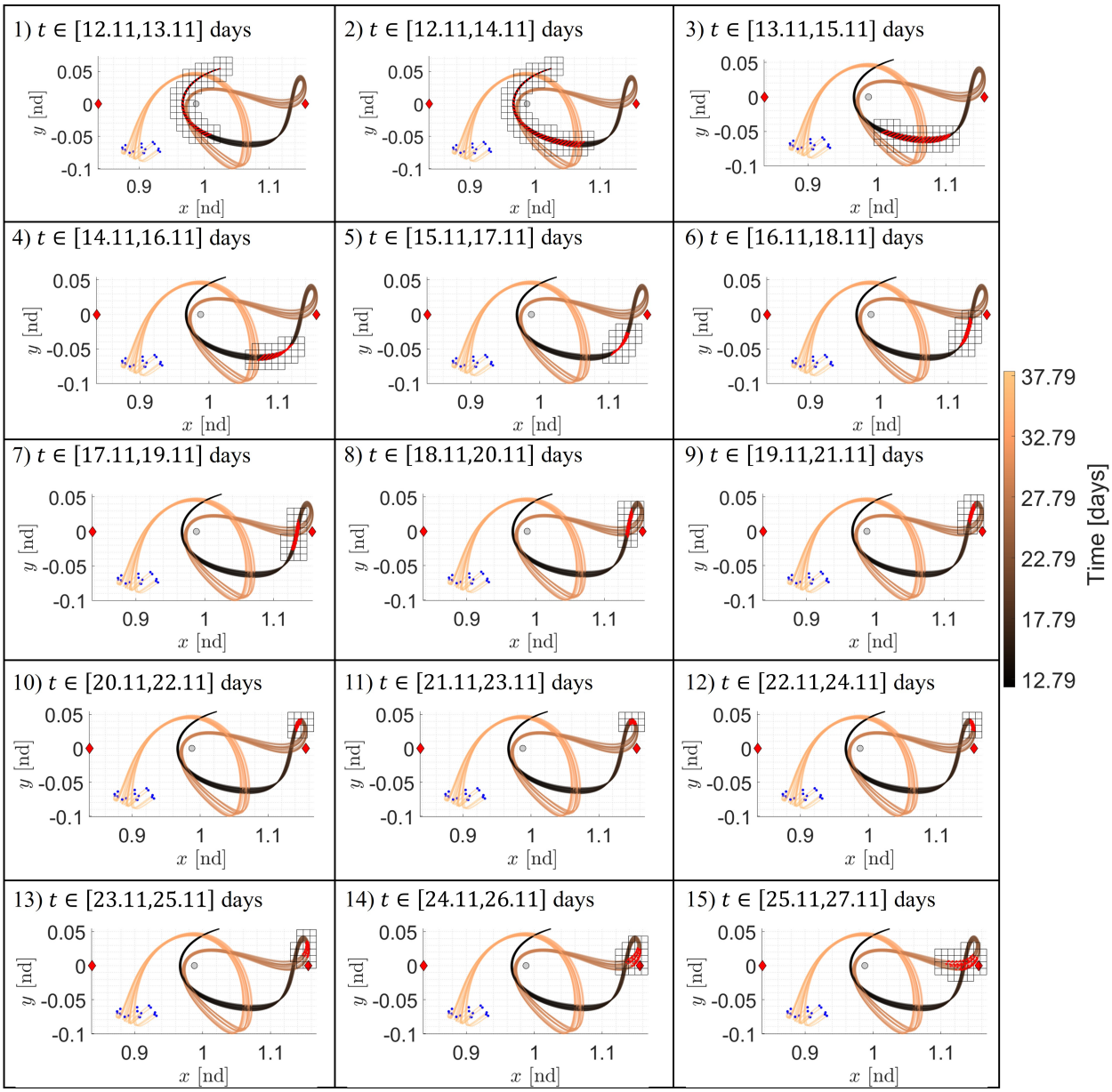

Figure 8.3: Coarse collision maps 1 - 15 constructed for a dynamic obstacle corridor observed at $t_{obs} = 12.11$ days near a distant prograde orbit at $C_J = 3.1700$ in the Earth-Moon CR3BP.

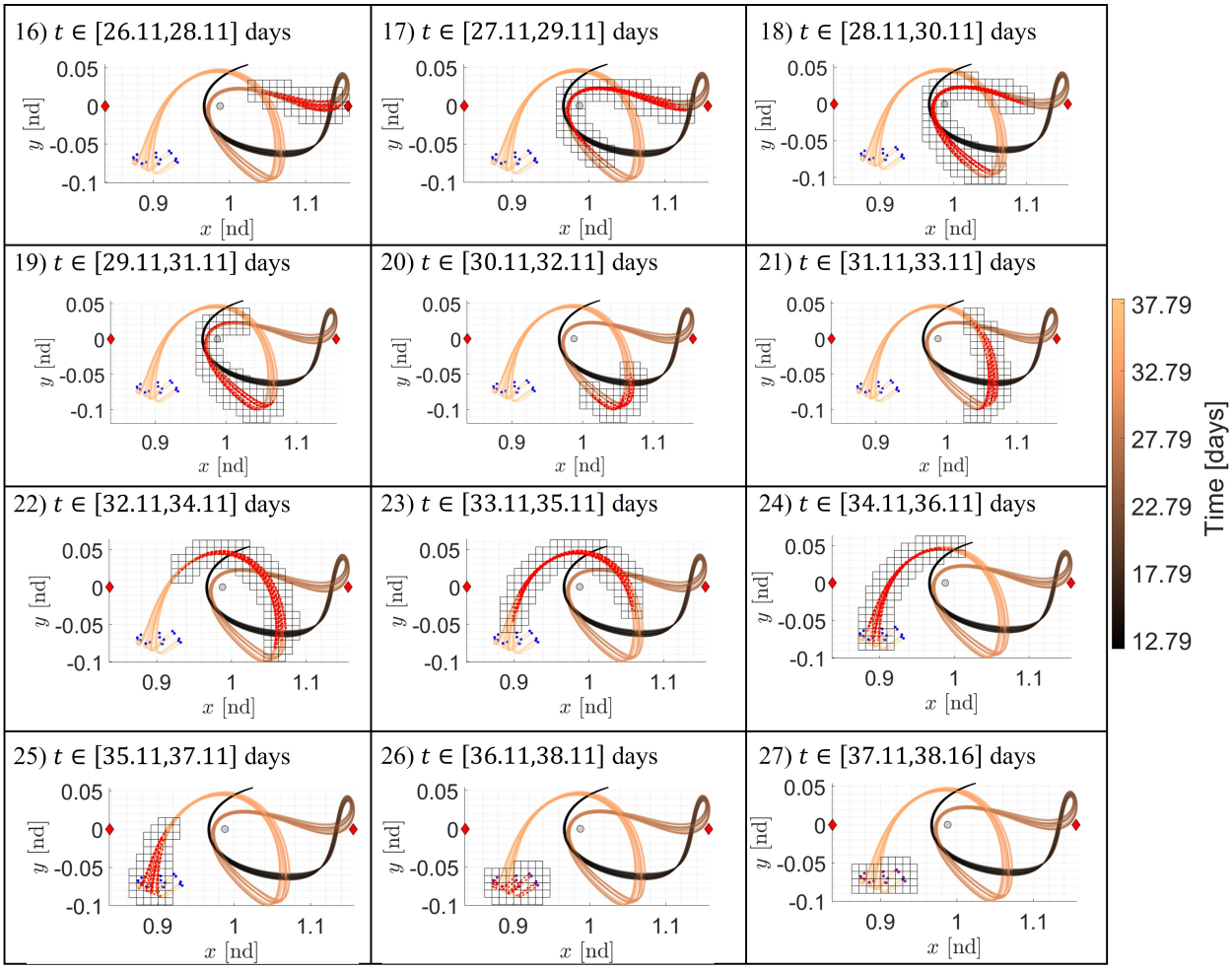

Figure 8.4: Coarse collision maps 16 - 27 constructed for a dynamic obstacle corridor observed at $t_{obs} = 12.11$ days near a distant prograde orbit at $C_J = 3.1700$ in the Earth-Moon CR3BP.

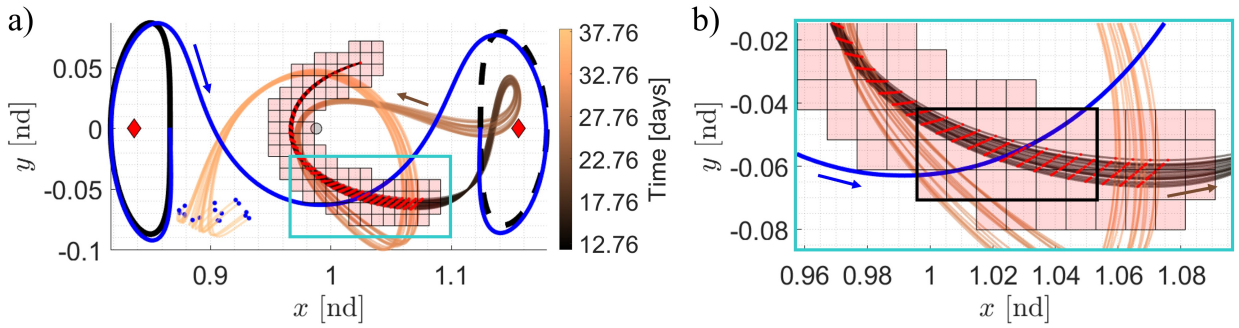

Figure 8.5: Overlap in position and time between a dynamic obstacle corridor and spacecraft trajectory is identified for $t \in [12.11, 14.11]$ days.

trajectory that corresponds to the map's time interval. This representation of the obstacle corridor and spacecraft path supports accurate comparisons of position and time without the computational effort of checking all cells in the coarse obstacle map.

Collisions occur if the spacecraft path overlaps with the obstacle corridor in position and time in at least one cell in a detailed obstacle map. First, the positions are analyzed to determine if overlap is present. Then, the time ranges for each object are compared. The time ranges are composed of the object's entrance and exit times, from propagation, of the potential collision cell. Collision detection terminates in this step upon first occurrence of a collision. If no collision is identified, fine collision detection is repeated for all cells in the coarse bounding region until either a collision is identified or all cells have been checked. In the scenario used throughout this chapter, a collision is identified at a collision time, t_{coll}, equal to 13.29 days and is visualized in Figure 8.6 where the filled yellow circles locate the discrete states representing the spacecraft path.

8.3 Summary of Governing Parameters and Decisions

To govern the automatic generation of a collision-free spacecraft path, a set of parameters and decisions is required. Table 8.1 presents the governing parameters and decisions for the two-level, cell-based collision checking method. Alternative selections for parameters or decisions may be used to alter the computational requirements and representation of objects in the environment.

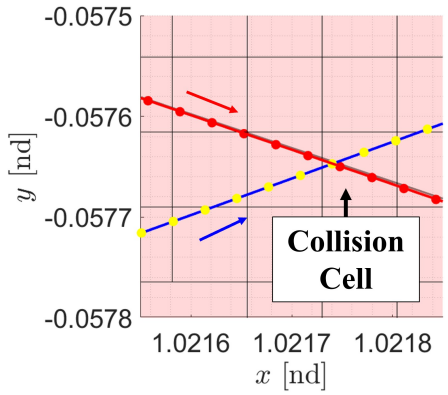

Figure 8.6: An example of a fine collision checking map with a collision occurring at $t_{coll} = 13.29$ days.

Table 8.1: Parameters and decisions governing the collision checking process.

Governing Parameter/Decision	**Influence on Computation and Results**	**Selected Value**
Cell height, u_{coarse} and u_{min}	Influences the computational effort for corridor avoidance, accuracy of recovering collisions.	$u_{min} = 21$ km, $u_{coarse} = 100u_{min}$
Cell shape	Influences sensitivity to numerically-approximate computations, complexity of cell decomposition, ability to control the minimum cell resolution.	Square or cube of constant height
Discrete trajectory representation, s_{coarse} and s_{min}	Equally spaced in intervals of arclength as a function of the cell height to minimize insufficient representation of continuous paths.	Equations 8.1 - 8.2
Obstacle map time interval	Reduces collision checking complexity.	2 days

Chapter 9

Application: Collision-Free Spacecraft Transfers in the Earth-Moon CR3BP

This chapter demonstrates the data-driven categorization, tree-based planning, and corridor avoidance methods presented in Chapters 6-8 for selected scenarios in the Earth-Moon CR3BP. First, trajectories that emanate from a local region of the phase space are categorized by their spatiotemporal variations and summarized using a digestible motion summary. The motion summaries consist of dynamic obstacle corridors where every trajectory in a corridor possesses similar geometries and trajectories across corridors possess distinct geometries. Then, an array of geometrically-distinct initial guesses are generated using a hierarchical tree-based planning approach. A selected set of initial guesses are corrected to demonstrate the utility of the tree-based framework. Finally, one set of continuous spacecraft trajectories are analyzed for collisions with dynamic obstacles. While these approaches are applied in the Earth-Moon CR3BP in this dissertation, it is expected that they may be used with higher-fidelity models of other three-body systems with minimal required modifications.

9.1 Spacecraft Motion Summaries from Local Initial State Sets

The data-driven categorization process described Chapter 6 is used to summarize sets of trajectories generated from local point clouds of initial conditions near three reference states, in addition to the example provided in Chapter 6, in the Earth-Moon CR3BP. These reference states are selected to produce sets of trajectories with different characteristics or itineraries to support exploring both the generalizability and limitations of the presented data-driven categorization pro-

cess. For all scenarios, the same HDBSCAN input parameters used for the L_1 Lyapunov orbit scenario in Chapter 6 are used: $N_{minCore} = 4$, $N_{minClust} = 5$, $\alpha = 5°$, and $\epsilon_{thresh} = 10^{-3}\sqrt{p}$ in Step 4; and $N_{minCore} = 1$, $N_{minClust} = 2$, $\epsilon = 0$ in Step 5. Table 9.1 summarizes the reference states in each scenario.

Table 9.1: Reference states for each selected scenario demonstrating the data-driven categorization process.

Reference Orbit	C_J	$\bar{x}_{PO} = [\bar{r}_{PO}, \bar{v}_{PO}]^T$ **[nd]**
11.99-day L_1 northern halo	3.1486	$\bar{r}_{PO}$ = [0.824125682194, 0, 0.0566946270474] $\bar{v}_{PO}$ = [0, 0.167128773665, 0]
7.962-day L_1 NRHO	2.9947	$\bar{r}_{PO}$ = [0.988454510548, -0.00114952066778, 0.00705658766736] $\bar{v}_{PO}$ = [-0.0150237631700, -1.82248741510, -0.148294929894]
10.92-day DPO	3.1700	$\bar{r}_{PO}$ = [1.02454653948, 0.0541769443650, 0] $\bar{v}_{PO}$ = [-0.380916147888, -0.110095105627, 0]

9.1.1 Summarized Motions Near L_1 Halo Orbits

9.1.1.1 L_1 Northern Halo Orbit

For this case, a reference state is selected at the leftmost crossing of the xz-plane along the L_1 northern halo orbit depicted with a dashed black curve in cluster 7 of Figure 9.1. The local point cloud of initial states are centered on this reference state. The associated trajectories are generated for 17.3 days and described by $p = 12$ states. The data-driven categorization process identifies 15 clusters composed of 1,677 trajectories as displayed in Figure 9.1. This figure uses a similar configuration as Figure 6.11: the horizontal and vertical axes represent the x- and y-position coordinates, respectively, in the Earth-Moon rotating frame in nondimensional units; the final state of each trajectory is denoted with a blue dot whereas color indicates the elapsed time along each trajectory; and black arrows denote the direction of motion. However, in this spatial scenario, each subfigure shows an individual cluster projected onto the $xy-$, $xz-$, and $yz-$planes for clarity. There are seven noise points that do not appear in the summary.

Figure 9.1 reveals that solutions either impact the Moon, remain near the reference orbit, revolve around the Moon with different geometries, or depart to the Earth vicinity. These key

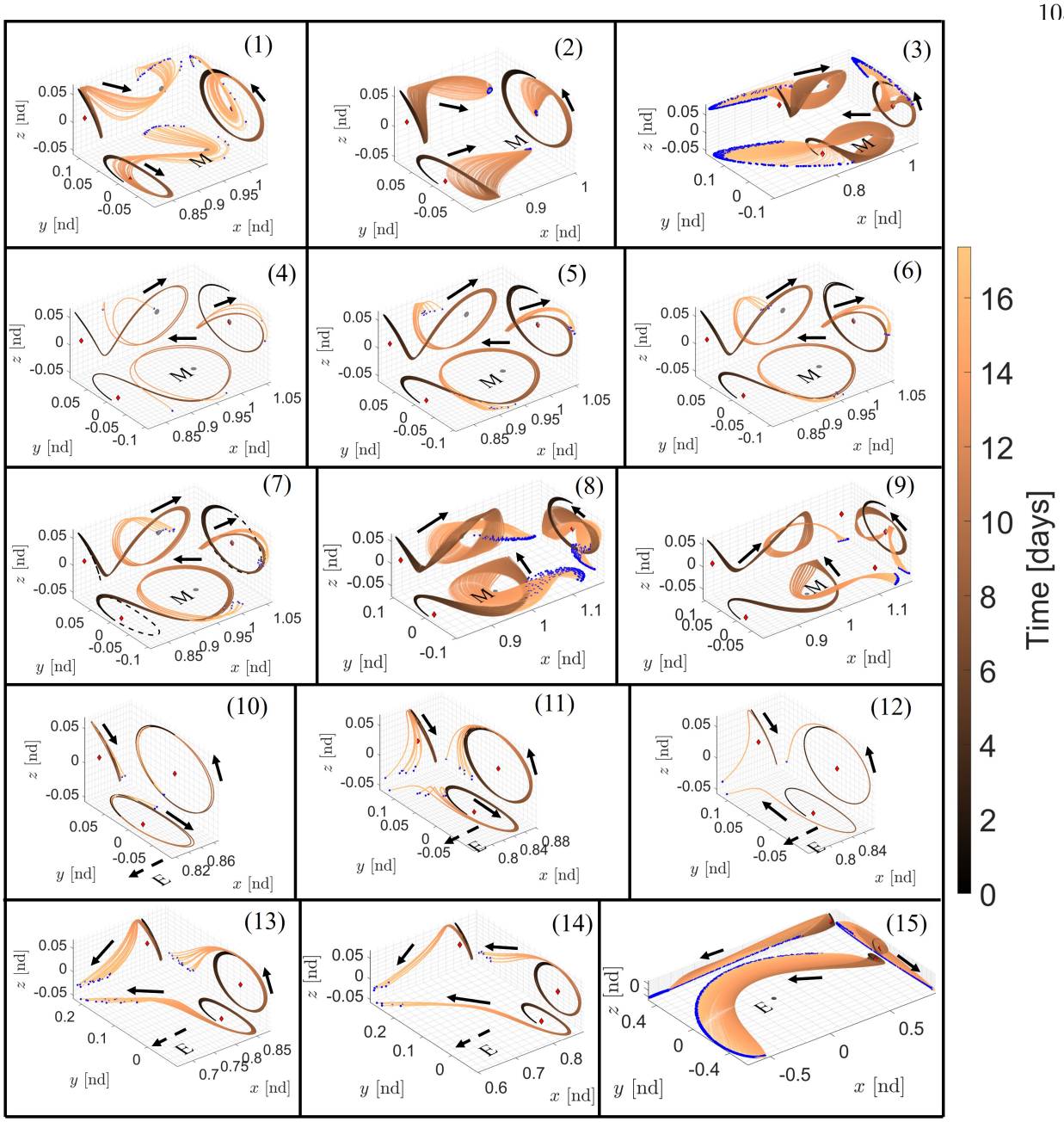

Figure 9.1: Fifteen types of 17.3-day trajectories that begin near the selected reference state along a L_1 northern halo orbit at $C_J = 3.1486$ in the Earth-Moon CR3BP.

motion types are separated across distinct clusters. In some cases, such as cluster 3, a single motion type captures a continuous evolution of a large set of trajectories over the selected time interval. However, propagating these trajectories for a longer time may result in additional geometric differentiation and, potentially, multiple clusters. Alternatively, the trajectories in clusters

4-7 perform one revolution about the Moon. However, due to the selected small values of $N_{minCore}$ and $N_{minClust}$ and slight variations in shape, time, and region of termination, these trajectories are separated into distinct clusters.

9.1.1.2 L_1 Near Rectilinear Halo Orbit

For the selected L_1 near rectilinear halo orbit (NRHO) in the Earth-Moon CR3BP, the reference state is placed near perilune. Then, trajectories are generated from the local point cloud of initial states near this reference state, propagated for 17.3 days and sampled using $p = 10$ states spaced equally in arclength. Applying the data-driven categorization process to the resulting 3,498 trajectories produces a summary with three distinct motion types and zero ungrouped trajectories. These clusters, along with the selected NRHO, are plotted in Figure 9.2 using the same configuration and spatial representation as Figure 9.1. In this figure, the second and third clusters separate motions that impact the Moon from those that quickly depart the vicinity of the NRHO with substantially different geometries. The first cluster in Figure 9.2 captures trajectories that remain close to the NRHO for the majority of the 17.3 days before finally departing. On either side of the cluster, the trajectories end in different regions of the solution space: either in the Moon vicinity or the Earth vicinity. However, across the cluster, the trajectories continuously evolve between these two behaviors, consistent with the use of a velocity-based feature vector and governing parameters

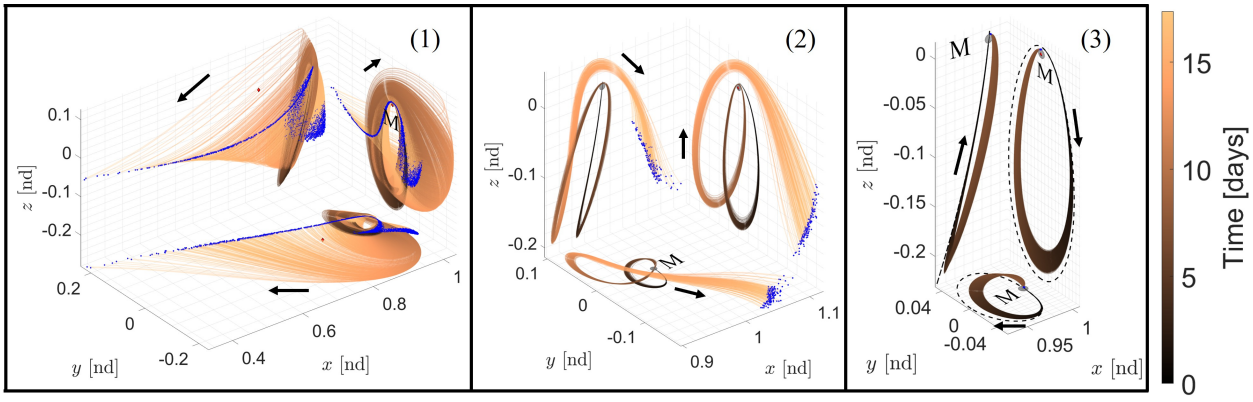

Figure 9.2: Three types of 17.3-day trajectories that begin near perilune along the selected L_1 near rectilinear halo orbit at $C_J = 2.9947$ in the Earth-Moon CR3BP.

that focus on local assessments of similarity.

9.1.2 Summarized Motions Near a Distant Prograde Orbit

Trajectories are propagated from a local point cloud of initial states near a reference state that lies between perilune and apolune along a DPO at a Jacobi constant of 3.1700 in the Earth-Moon CR3BP. In this example, 1,506 trajectories are generated for 26.05 days to allow distinct geometries to emerge and described by $p = 24$ states. All but nine of the trajectories are summarized in the 18 clusters depicted in Figure 9.3, with the same representation as Figure 6.11. Across the 18 clusters in Figure 9.3, trajectories in each cluster possess a similar geometry and distinct geometries appear in different clusters. For instance, clusters 1-2 capture trajectories that resemble the reference DPO. Clusters 4-9, however, summarize trajectories that depart through the L_2 gateway. These trajectories are likely separated into different clusters due to the evolution of the sampled states as they reach distinct locations in the exterior region. Cluster 3 then captures a transition between these two itineraries. Next, clusters 10-11 capture trajectories that perform a few revolutions around the Moon with two different geometries. Finally, clusters 12-18 contain trajectories with

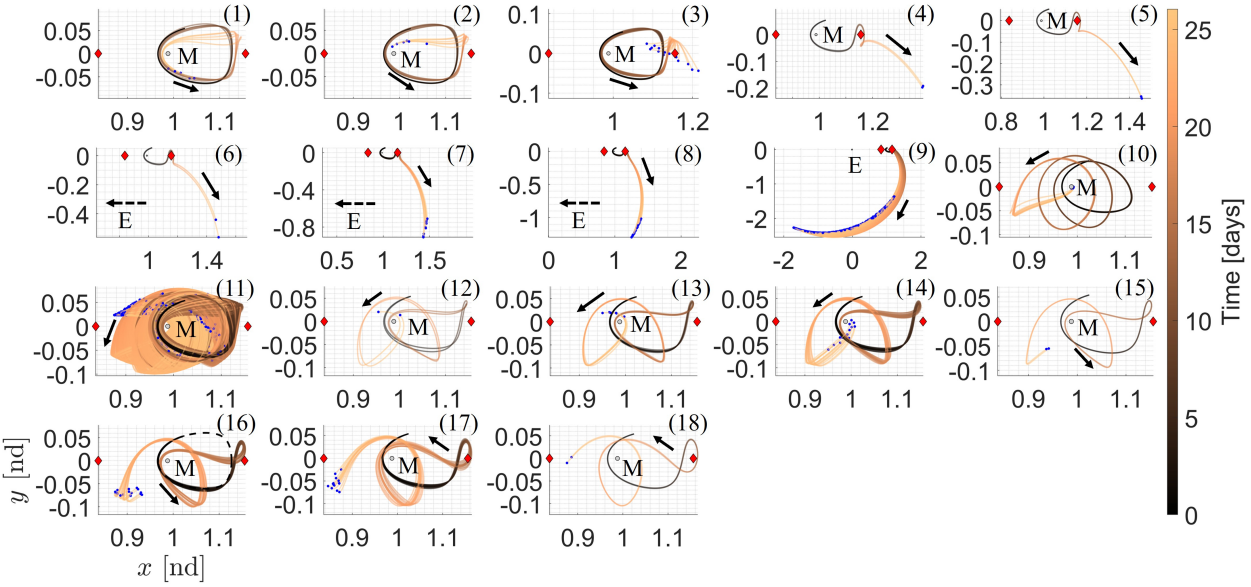

Figure 9.3: Distinct types of 26.05-day trajectories beginning near a reference state between perilune and apolune along a DPO at $C_J = 3.1700$ in the Earth-Moon CR3BP.

a similar geometry that are distributed across multiple clusters due to localized variations in the 1) path near the first apolune, 2) second perilune passage, and 3) location of the final state. This additional geometric differentiation supplies a more detailed summary of the key motion types.

9.2 Planar Transfers Between L_1 and L_2 Lyapunov Orbits

To demonstrate the use of the hierarchical tree-based approach for automatic and rapid collision-free trajectory design described in Chapter 7, two planar design scenarios in the Earth-Moon CR3BP are explored in this section. Table 9.2 summarizes the orbital parameters and boundary conditions used in each scenario. These scenarios are selected to demonstrate that 1) geometrically distinct initial guesses for spacecraft trajectories can be automatically generated and 2) collisions with dynamic obstacles can be identified, when they exist. The values of the governing parameters used in these scenarios are listed in Table 7.1. Furthermore, to demonstrate the utility of tree-based initial guess construction, four initial guesses are selected in each case for correction. Finally, the continuous trajectories recovered in the second scenario are assessed for collisions.

Table 9.2: Orbit parameters and boundary conditions for each planar transfer design scenario.

Case	**Ref. Orbits**	**C_J**	**Boundary Conditions [nd]**
1	12.24-day L_1 Lyapunov	3.1556	$\bar{x}_{init} = [0.866634949946303, 0, 0, 0, -0.210056789639986, 0]^T$
	14.79-day L_2 Lyapunov	3.1556	$\bar{x}_{targ} = [1.12398465047742, 0, 0, 0, 0.158922217869289, 0]^T$
2	12.24-day L_1 Lyapunov	3.1556	$\bar{x}_{init} = [0.866634949946303, 0, 0, 0, -0.210056789639986, 0]^T$
	14.90-day L_2 Lyapunov	3.1459	$\bar{x}_{targ} = [1.11440879949729, 0, 0, 0, 0.204364798565404, 0]^T$

9.2.1 Single-Energy-Level Transfers

This example continues the transfer design scenario between L_1 and L_2 Lyapunov orbits at $C_J = 3.1556$ as presented in Chapter 7. Specifically, the same forest as in Section 7.1.2 is searched to produce 100 initial guesses. Figures 9.4 - 9.8 showcase the wide array of geometries present across

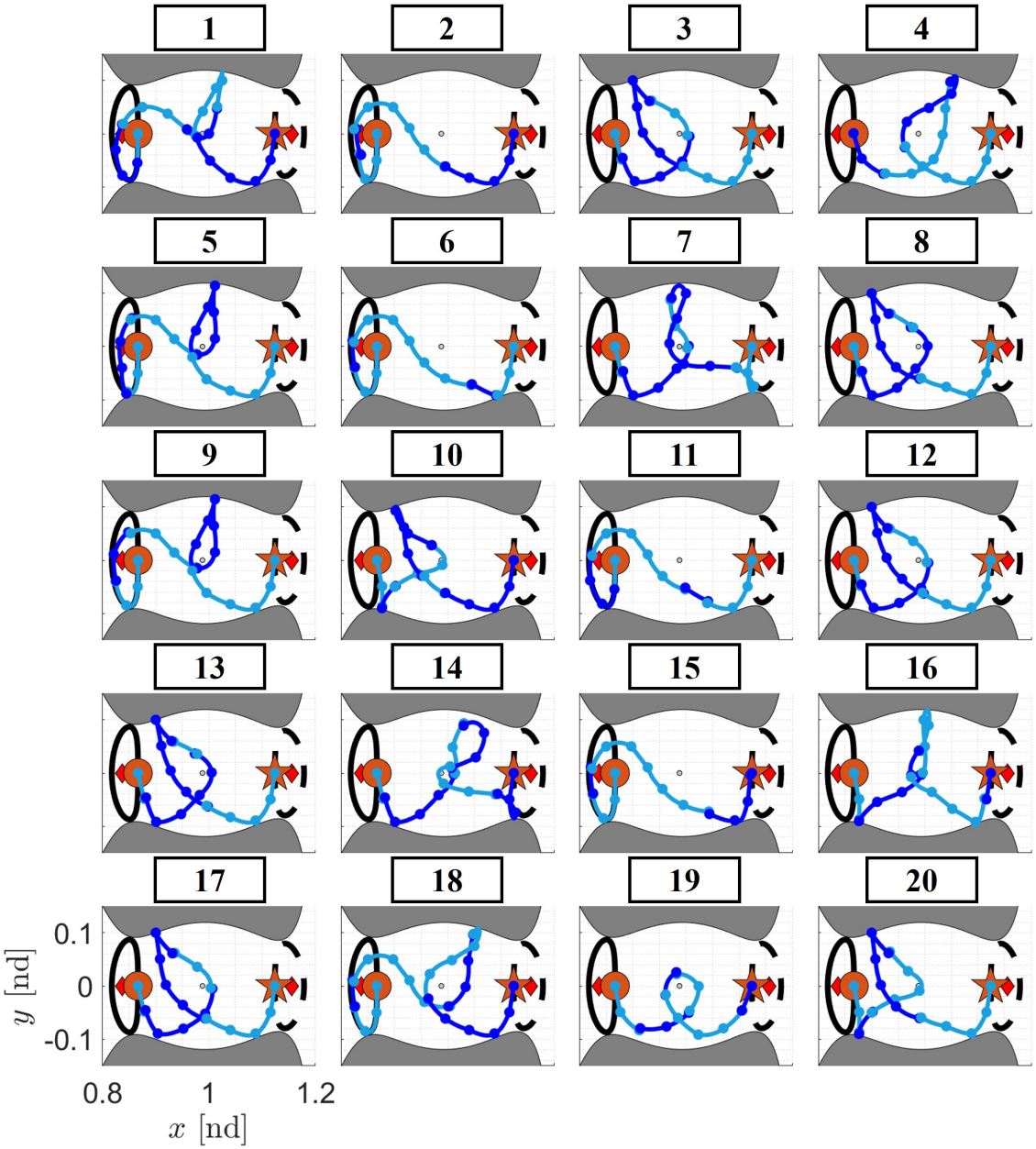

Figure 9.4: Initial guesses 1-20 with distinct geometries for spacecraft trajectories connecting boundary conditions along L_1 and L_2 Lyapunov orbits at $C_J = 3.1556$ in the Earth-Moon CR3BP.

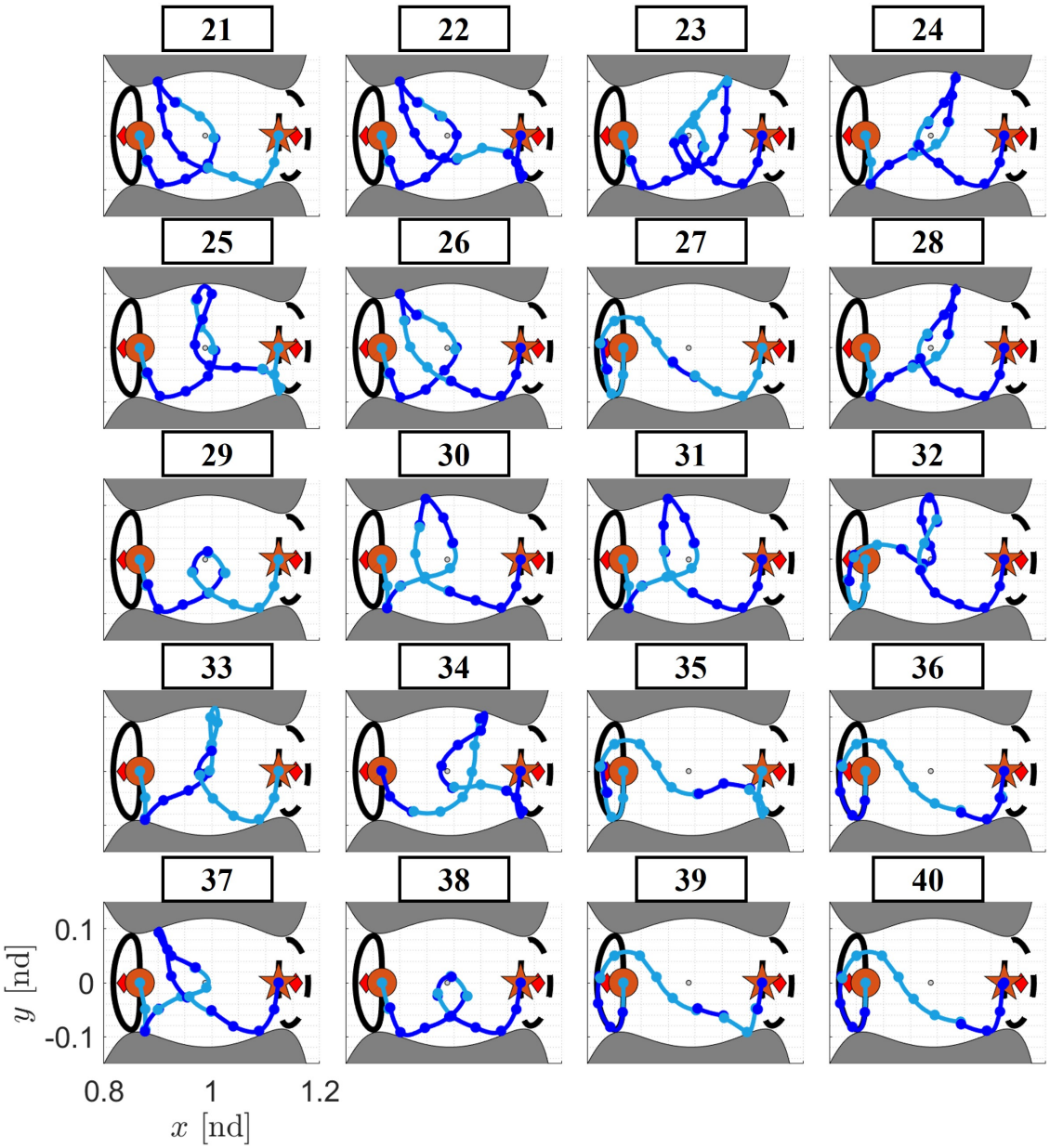

Figure 9.5: Initial guesses 21-40 with distinct geometries for spacecraft trajectories connecting boundary conditions along L_1 and L_2 Lyapunov orbits at $C_J = 3.1556$ in the Earth-Moon CR3BP.

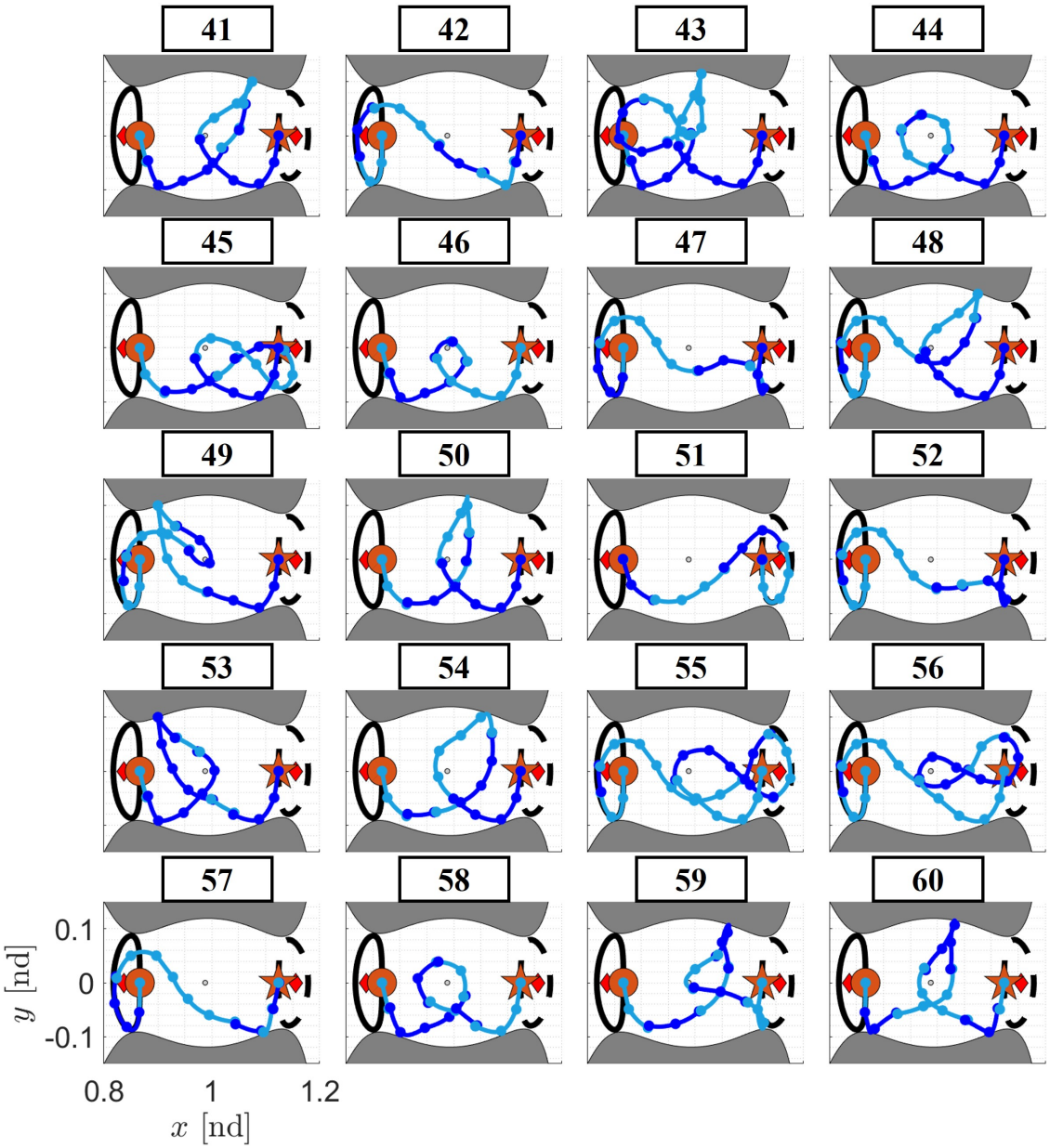

Figure 9.6: Initial guesses 41-60 with distinct geometries for spacecraft trajectories connecting boundary conditions along L_1 and L_2 Lyapunov orbits at $C_J = 3.1556$ in the Earth-Moon CR3BP.

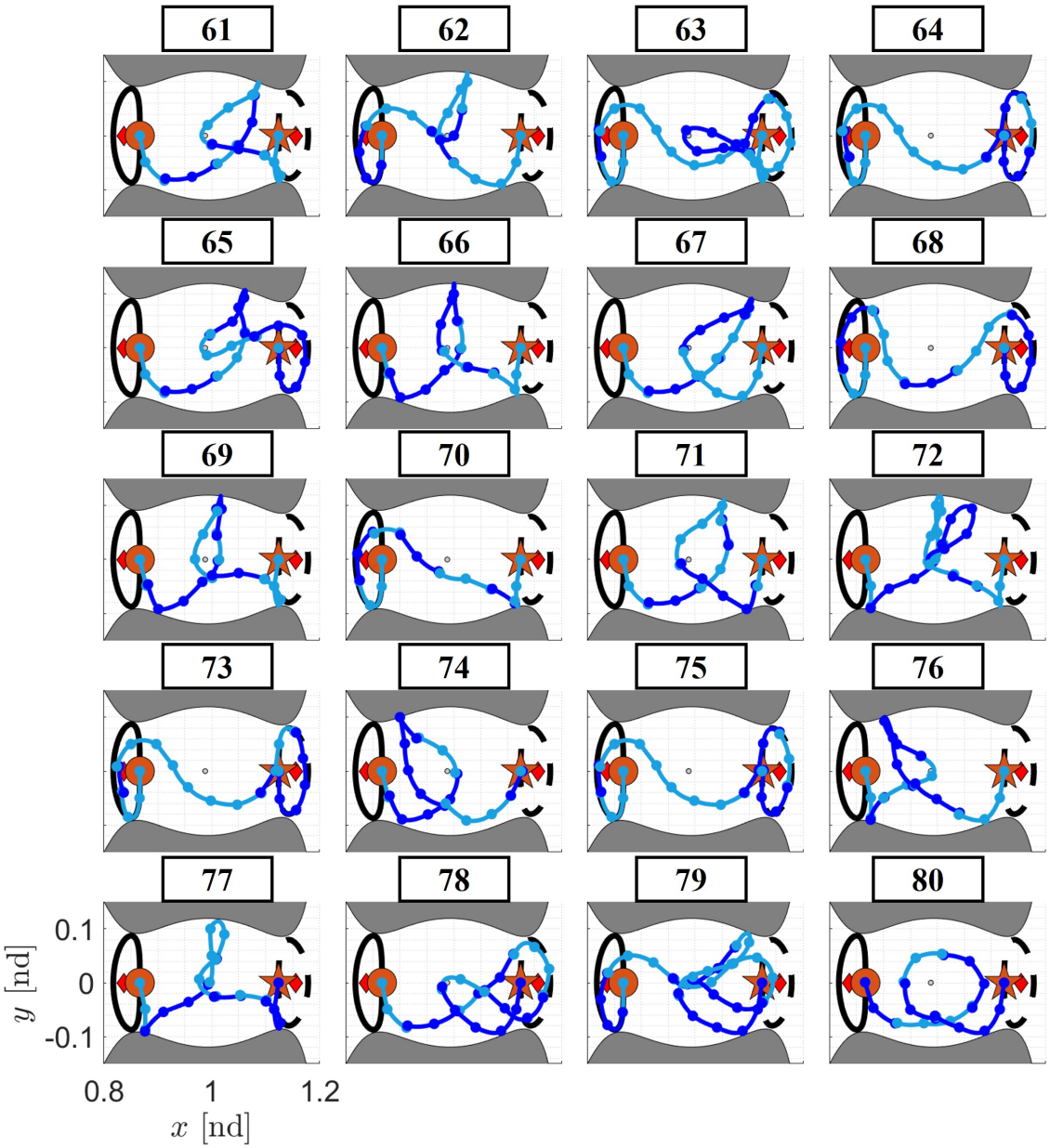

Figure 9.7: Initial guesses 61-80 with distinct geometries for spacecraft trajectories connecting boundary conditions along L_1 and L_2 Lyapunov orbits at $C_J = 3.1556$ in the Earth-Moon CR3BP.

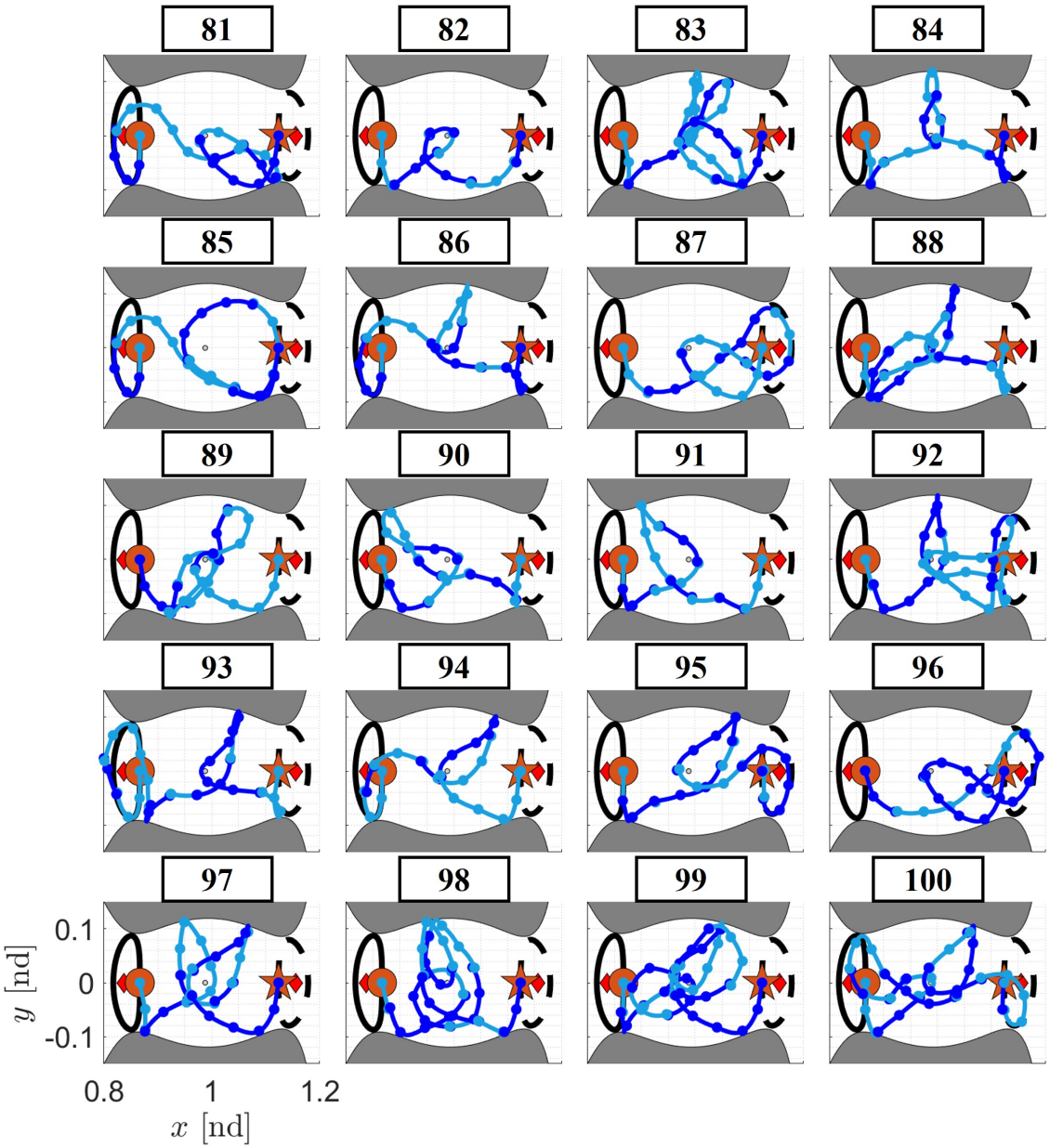

Figure 9.8: Initial guesses 81-100 with distinct geometries for spacecraft trajectories connecting boundary conditions along L_1 and L_2 Lyapunov orbits at $C_J = 3.1556$ in the Earth-Moon CR3BP.

this set of initial guesses. In each subfigure, the horizontal and vertical axes are the x- and y-axes of the Earth-Moon rotating frame as labeled in the bottom left subfigure using nondimensional units; all subfigures possess the same axis limits in x and y. In addition, the Moon is depicted as a gray circle scaled to possess its radius; red diamonds denote L_1 and L_2; the gray curves denote the ZVCs at $C_J = 3.1556$; the orange circle and star locate the initial and target states, respectively; the initial and final orbits are shown with solid and dashed black lines, respectively; and the initial guesses are colored to alternate between light and dark blue to differentiate between branches leveraged from different trees. Each initial guess is recovered from a unique tree sequence composed of four to 15 trees and uniquely numbered from lowest to highest values of the cumulative relative velocity discontinuity.

A wide array of geometries are present across these 100 initial guesses. Each initial guess possesses start and end segments that do not abruptly depart from or arrive onto the Lyapunov orbits. Furthermore, initial guesses may complete almost one revolution about one or both periodic orbits as demonstrated by initial guesses 1, 32, 56, 64, and 93; directly depart from or arrive onto the initial and final orbits, as seen in initial guesses 19, 50, and 82; or perform a high-curvature turn near the ZVCs, exhibited by initial guesses 16, 31, 47, and 84. Additional geometric variety is present in the vicinity of the Moon as initial guesses possess up to three revolutions about the Moon or partial revolutions in the vicinity of the periodic orbits before returning to the Moon's vicinity. This geometric diversity in the initial guesses may be useful for supplying a variety of trajectory options to a designer, ground support, or onboard decision-making algorithm.

Four coarse initial guesses are corrected to study their geometric evolution. Each initial guess is associated with two continuous solutions: an energy-focused trajectory that prioritizes reducing the maneuver requirements and a geometry-focused trajectory that prioritizes geometric resemblance to the initial guess. The characteristics of the continuous trajectories associated with the selected initial guesses are summarized in Table 9.3. Since these transfers span a single energy level, the Jacobi constant could be constrained during correction. However, this constraint may not allow recovery of continuous transfers that geometrically resemble the initial guess.

Table 9.3: Summary of transfer characteristics connecting planar L_1 and L_2 Lyapunov orbits at $C_J = 3.1556$.

Case	Fig. #	Initial Guess		ID # - Priority	Corrected Path	
		Δv [m/s]	TOF [days]		Δv [m/s]	TOF [days]
1	9.9	1,528.54	13.88	80 - Geometry	386.32	25.66
				80 - Energy	5.85	26.85
	9.10	1,349.93	28.52	85 - Geometry	443.55	42.86
				85 - Energy	6.77	38.21
	9.11	1,171.75	22.60	87 - Geometry	542.50	34.12
				87 - Energy	19.14	30.34
	9.12	1,602.38	21.15	65 - Geometry	308.37	28.50
				65 - Energy	125.94	26.88

Two of the selected initial guesses perform geometrically-distinct, low-curvature revolutions about the Moon but correct to similar energy-focused trajectories. The coarse initial guesses with the 80th and 85th lowest values of the relative velocity discontinuity are depicted in Figures 9.9 - 9.10 along with their corresponding transfers. The configuration of Figures 9.9 - 9.10 is the

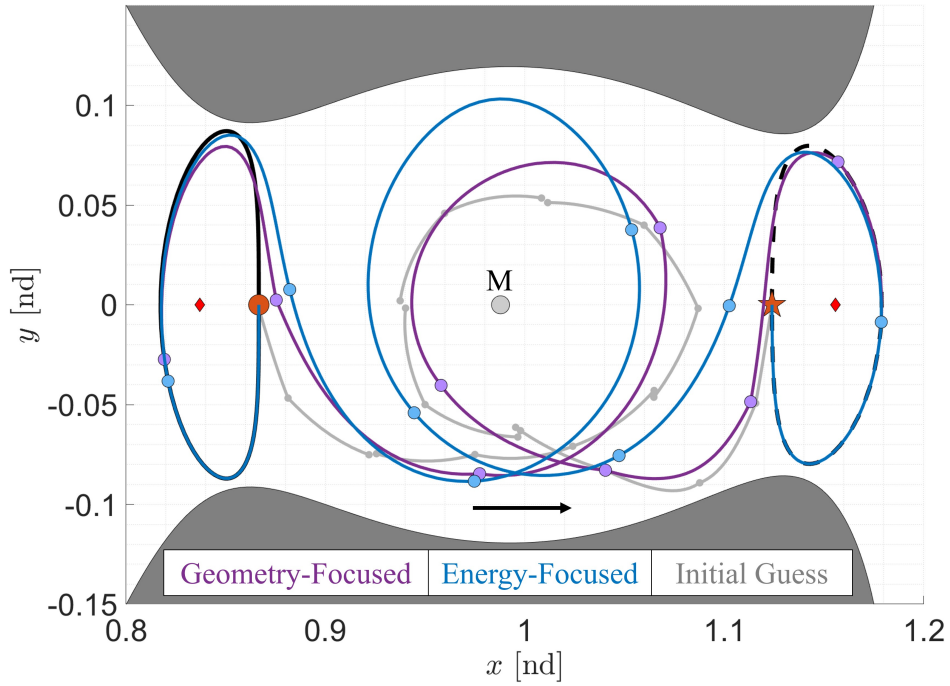

Figure 9.9: The evolution of a single-revolution transfer connecting boundary conditions on L_1 and L_2 Lyapunov orbits at $C_J = 3.1556$: initial guess (gray), geometry-focused solution (purple) with Δv equal to 386.32 m/s and TOF equal to 25.66 days, and energy-focused solution (blue) with Δv equal to 5.85 m/s and TOF equal to 26.85 days.

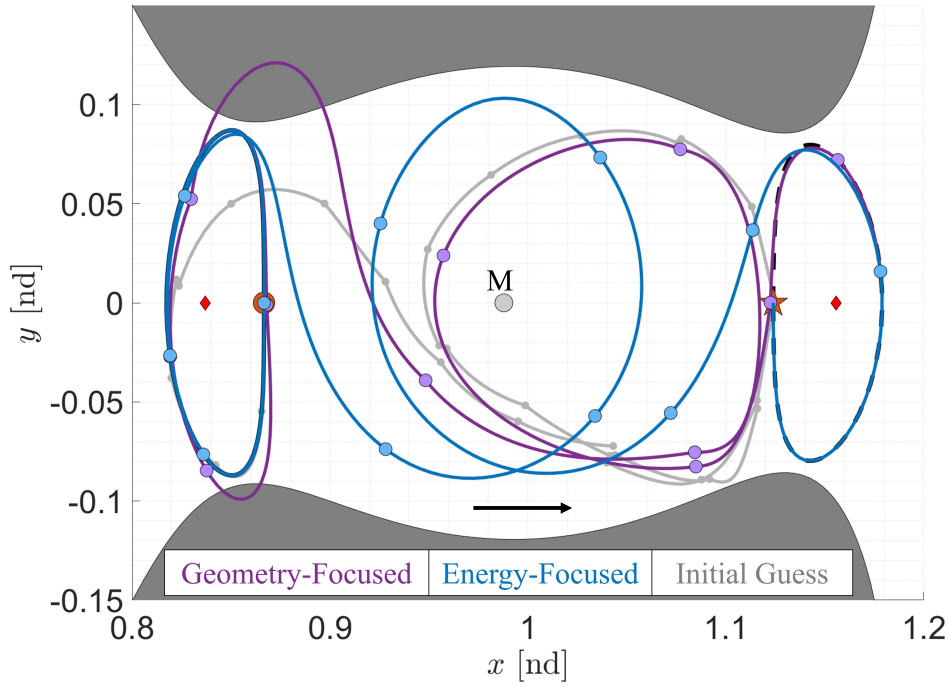

Figure 9.10: The evolution of a geometrically-distinct single-revolution transfer connecting boundary conditions on L_1 and L_2 Lyapunov orbits at $C_J = 3.1556$: initial guess (gray), geometry-focused solution (purple) with Δv equal to 443.55 m/s and TOF equal to 42.86 days, and energy-focused solution (blue) with Δv equal to 6.77 m/s and TOF equal to 38.21 days.

same as the configuration of Figure 7.11: the initial guess is plotted in gray, the geometry-focused solution is plotted in purple, and the energy-focused solution is displayed in blue. Additionally, the ZVCs are plotted at $C_J = 3.1556$. Eight and 10 maneuvers are manually placed along these initial guesses, respectively. The corrections process produces a family of smooth transfers that resemble the initial guesses. The lunar revolution exhibited by the geometry-focused solution in Figure 9.10 resembles orbits found in the distant prograde orbit family. As energy is prioritized during corrections, the total maneuver magnitude is significantly reduced in both scenarios, as listed in Table 9.3. The geometry-focused solution in Figure 9.9 possesses a $\Delta v = 386.32$ m/s and is reduced to 5.85 m/s. Similarly, the Δv of the solutions in Figure 9.10 is reduced from 443.55 m/s to 6.77 m/s. Both single-revolution initial guesses converge to an energy-focused solution that exhibits the geometry of a known a heteroclinic connection at a nearby Jacobi constant equal to 3.1564 [26].

The next set of initial guesses selected for correction exhibit high-curvature revolutions about the Moon. The initial guesses and their continuous solutions are displayed in Figures 9.11 - 9.12 and possess the 87th and 65th lowest values of relative velocity discontinuity. Nine maneuvers are placed along both initial guesses. The geometry of the initial guess in Figure 9.11 significantly evolves as the total maneuver magnitude reduces from 542.50 m/s to 19.14 m/s, as listed in Table 9.3. This significant reduction in maneuver requirements may be a result of the existence of a known single-revolution heteroclinic connection at a nearby value of the Jacobi constant [26]. The solutions in Figure 9.12 exhibit less geometric variance as the maneuver magnitude is prioritized during correction, despite a decrease of 182.43 m/s in the Δv.

The variation in Jacobi constant is analyzed across each of the eight continuous trajectories that connect boundary conditions between L_1 and L_2 Lyapunov orbits at $C_J = 3.1556$. At the start of each arc along each transfer, the Jacobi constant is calculated and plotted in Figure 9.13.

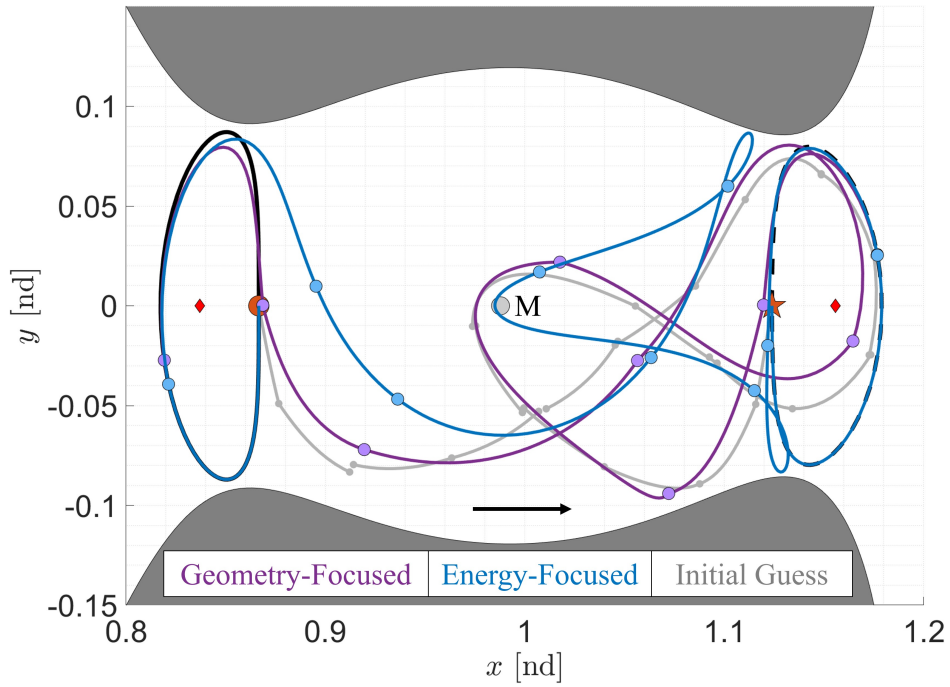

Figure 9.11: The evolution of a geometrically-distinct transfer connecting boundary conditions on L_1 and L_2 Lyapunov orbits at $C_J = 3.1556$: initial guess (gray), geometry-focused solution (purple) with Δv equal to 542.50 m/s and TOF equal to 34.12 days, and energy-focused solution (blue) with Δv equal to 19.14 m/s and TOF equal to 30.34 days.

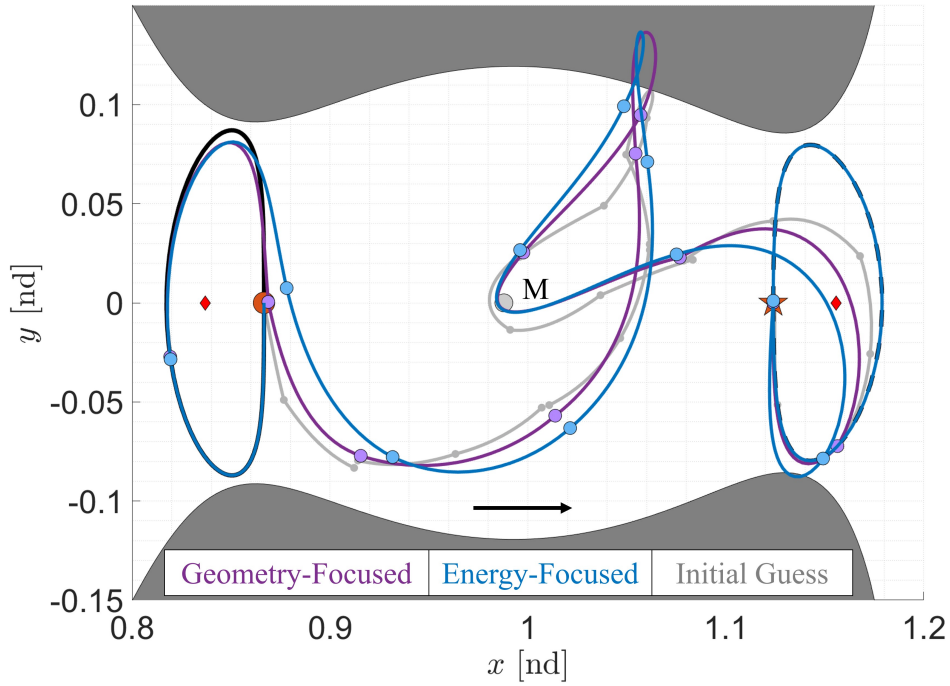

Figure 9.12: The evolution of a single-revolution transfer with high curvature that connects boundary conditions on L_1 and L_2 Lyapunov orbits at $C_J = 3.1556$: initial guess (gray), geometry-focused solution (purple) with Δv equal to 308.37 m/s and TOF equal to 28.50 days, and energy-focused solution (blue) with Δv equal to 125.94 m/s and TOF equal to 26.88 days.

The discrete values of the Jacobi constant are located with black circles while the desired value of the Jacobi constant, $C_J = 3.1556$, is marked by a red dashed line. Additionally, each subfigure is labeled with the trajectory identification number and priority type: geometry-focused (G) and energy-focused (E). As expected, the energy-focused transfers not only possess less overall variation in C_J, but also, generally, exhibit less drastic changes in C_J between arcs along the transfers than the geometry-focused trajectories. Visualizing the change in the Jacobi constant is particularly interesting for the transfers corresponding to initial guesses 65 and 85 as one or both of their corrected solutions pass through the ZVCs due to changes in C_J, as seen in Figures 9.10 and 9.12 .

9.2.2 Multi-Energy-Level Transfers

Consider a transfer design scenario between L_1 and L_2 Lyapunov orbits at different C_J. A forest is generated using the governing parameters listed in Table 7.1 where the tree roots

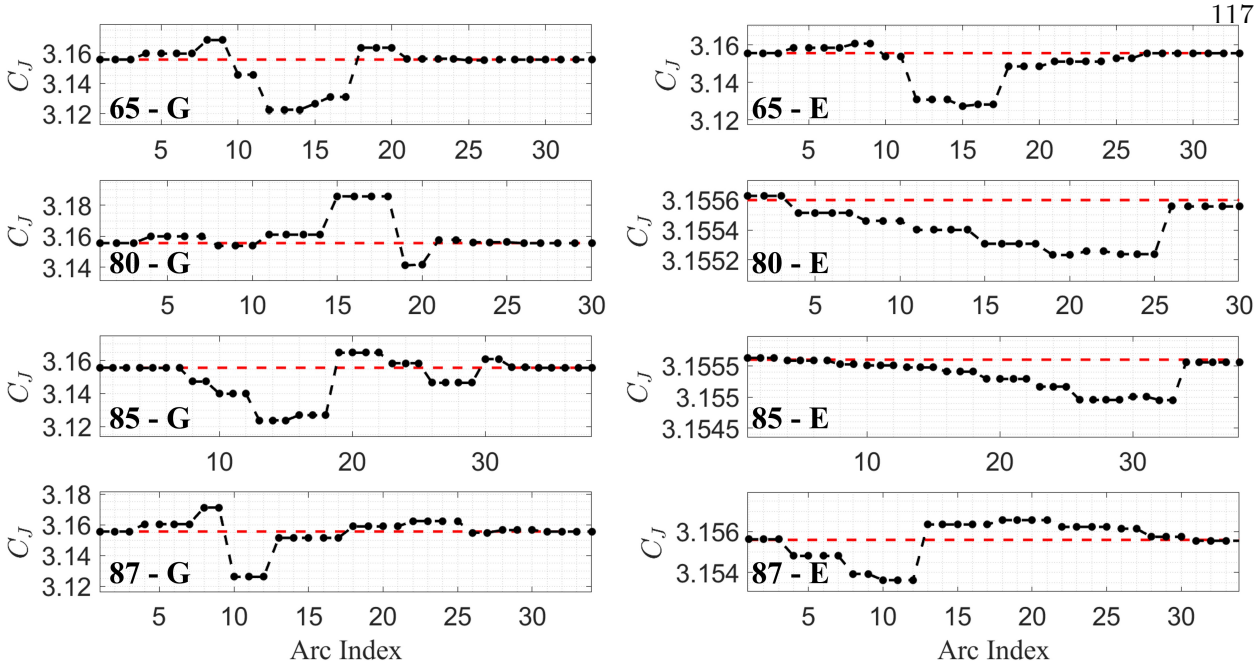

Figure 9.13: Evolution of the Jacobi constant along the geometry- (G) and energy-focused (E) solutions for transfers connecting L_1 and L_2 Lyapunov orbits at $C_J = 3.1556$, marked with a dashed red line, in the Earth-Moon CR3BP.

span the desired range of C_J. Figure 9.14 displays the value of the Jacobi constant at each tree root using a blue asterisk; tree roots are labeled with unique tree identification numbers (Tree ID). The minimum, mean, and maximum values of the Jacobi constant are also indicated by red horizontal lines. This figure demonstrates that the states at the tree roots span the desired range, $C_J = [3.1459, 3.1556]$, between the Jacobi constant values of the initial and final orbits. Note that tree roots with the same position vector but distinct velocity vectors correspond to states at unique values of C_J. In this scenario, the forest is composed of 1,190 trees.

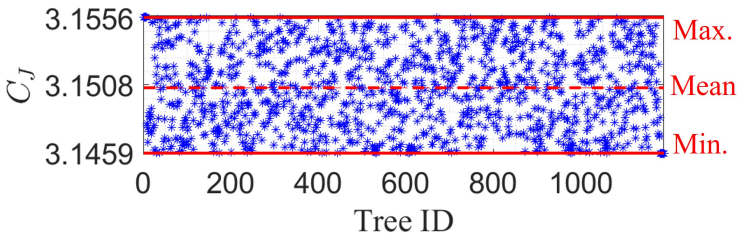

Figure 9.14: The state vectors of the tree roots span the desired range of Jacobi constant values, $C_J = [3.1459, 3.1556]$.

One hundred discontinuous initial guesses are recovered for this scenario. The set of geometrically-distinct initial guesses are displayed in Figures 9.15 - 9.19 with the same representation as Figures 9.4 - 9.8 except branches that compose the initial guesses are colored to reflect the associated value of the Jacobi constant. These initial guesses exhibit similar geometries to the initial guesses spanning a single value of C_J; this may suggest a continuous geometric evolution within this solution space across different energy levels. The characteristics for the corrected transfer scenarios in this section are summarized in Table 9.4.

The initial guesses with the lowest and third-lowest values of the relative velocity discontinuity are corrected to study their geometric evolution. Figures 9.20 - 9.21 depict these initial guesses and their corresponding transfers with the same configuration as Figure 9.9. In each figure, the ZVCs are plotted at $C_J = 3.1459$ even though the Jacobi constant evolves along the transfer. Seven impulsive maneuvers are manually placed along initial guess 1 while 10 maneuvers are placed along initial guess 3. When corrected, geometry- and energy-focused paths are recovered from families of smooth transfers that resemble the initial guesses and are plotted in Figures 9.20 - 9.21. While heteroclinic connections do not exist for transfers connecting periodic orbits at different energy levels, the geometry of the energy-focused solution in Figure 9.20 resembles the geometry of a known zero-revolution heteroclinic connection [26]. Similar geometry is exhibited by the continuous transfers in Figure 7.11. Furthermore, the solutions provided in Figure 9.21 possess resemblance to

Table 9.4: Summary of transfer characteristics connecting a planar L_1 Lyapunov orbit at $C_J = 3.1556$ and a planar L_2 Lyapunov orbit at $C_J = 3.1459$.

Case	**Fig. #**	**Initial Guess**		**ID # - Priority**	**Corrected Path**	
		Δv [m/s]	**TOF [days]**		**Δv [m/s]**	**TOF [days]**
2	9.20	1,165.01	20.66	1 - Geometry	252.60	23.08
				1 - Energy	14.04	26.49
	9.21	923.18	25.94	3 - Geometry	312.98	26.92
				3 - Energy	21.78	28.02
	9.22	883.45	15.90	62 - Geometry	315.31	26.81
				62 - Energy	24.66	26.07
	9.23	1,540.14	26.47	72 - Geometry	313.01	33.12
				72 - Energy	150.62	34.43

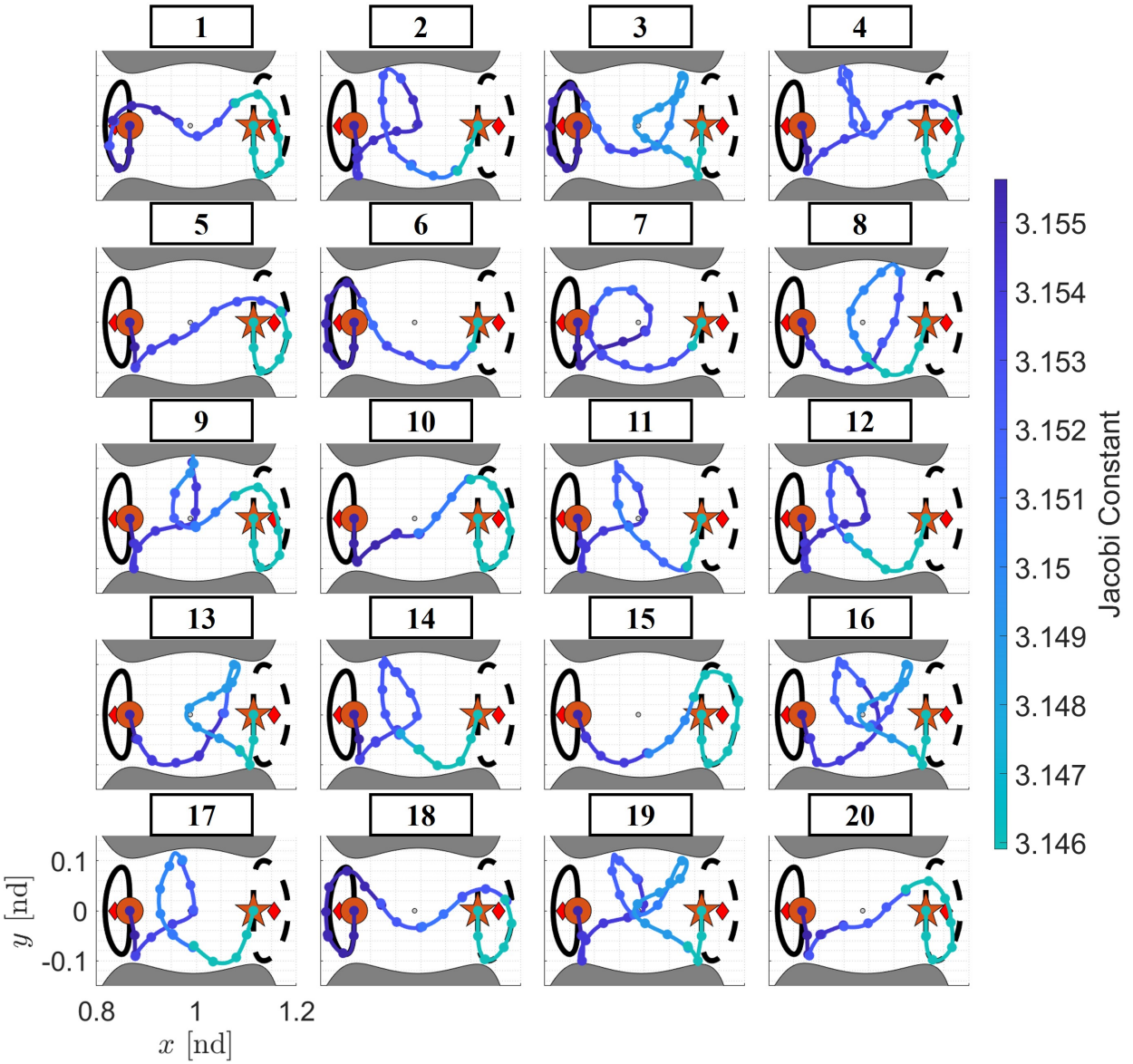

Figure 9.15: Initial guesses 1-20 with distinct geometries for spacecraft trajectories connecting boundary conditions along an L_1 Lyapunov orbit at $C_J = 3.1556$ and L_2 Lyapunov orbit at $C_J = 3.1459$ in the Earth-Moon CR3BP.

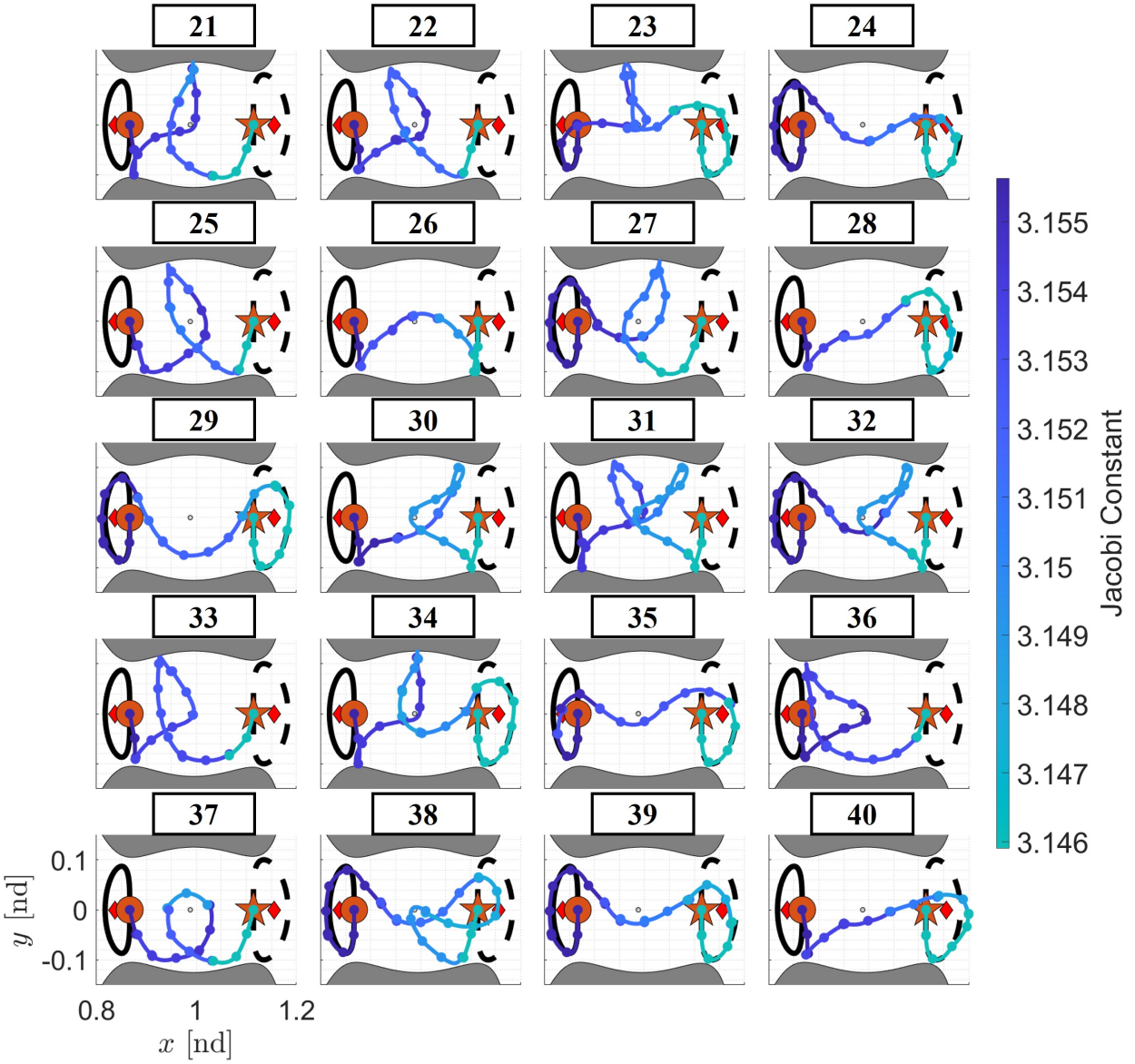

Figure 9.16: Initial guesses 21-40 with distinct geometries for spacecraft trajectories connecting boundary conditions along an L_1 Lyapunov orbit at $C_J = 3.1556$ and L_2 Lyapunov orbit at $C_J = 3.1459$ in the Earth-Moon CR3BP.

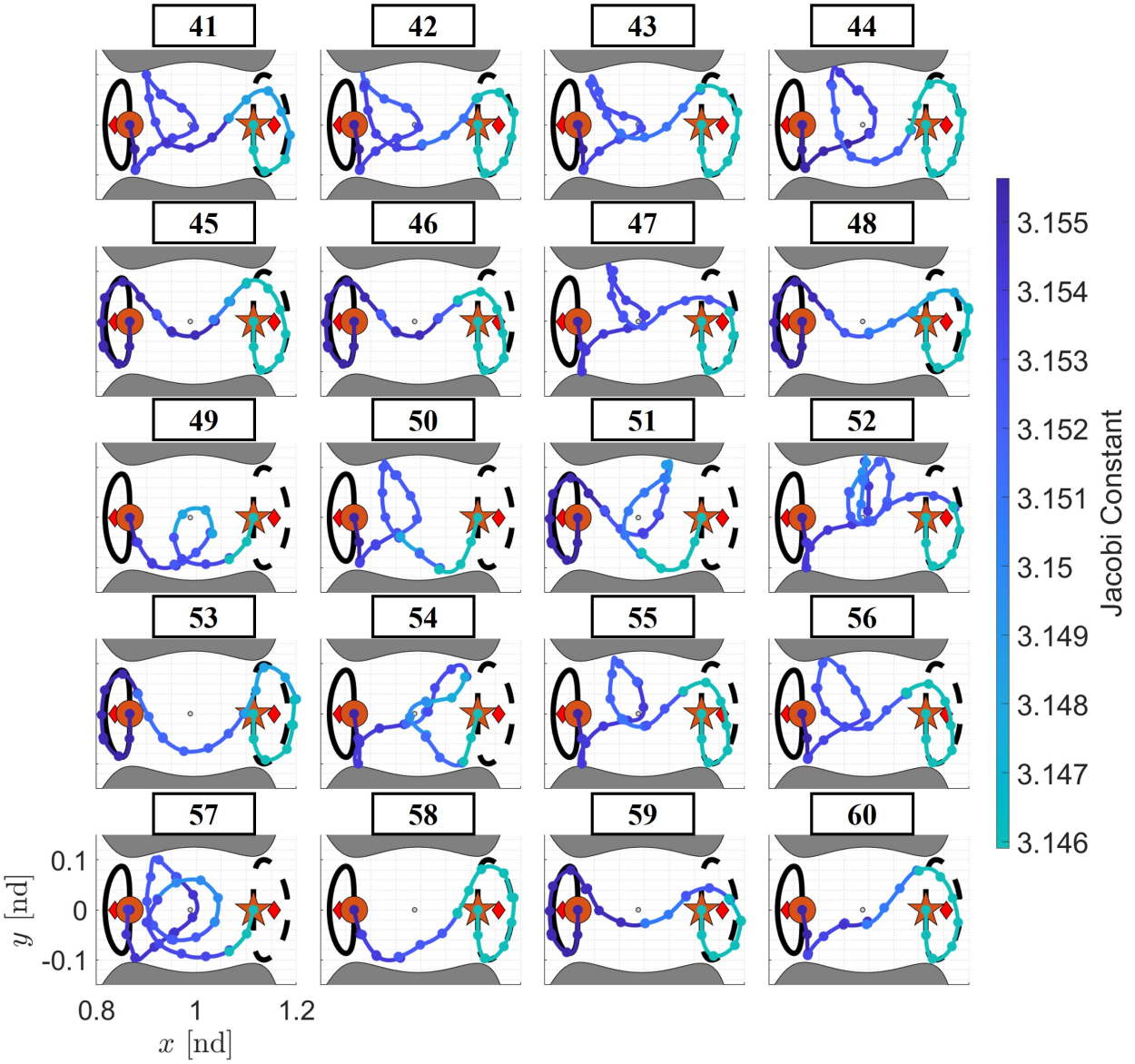

Figure 9.17: Initial guesses 41-60 with distinct geometries for spacecraft trajectories connecting boundary conditions along an L_1 Lyapunov orbit at $C_J = 3.1556$ and L_2 Lyapunov orbit at $C_J = 3.1459$ in the Earth-Moon CR3BP.

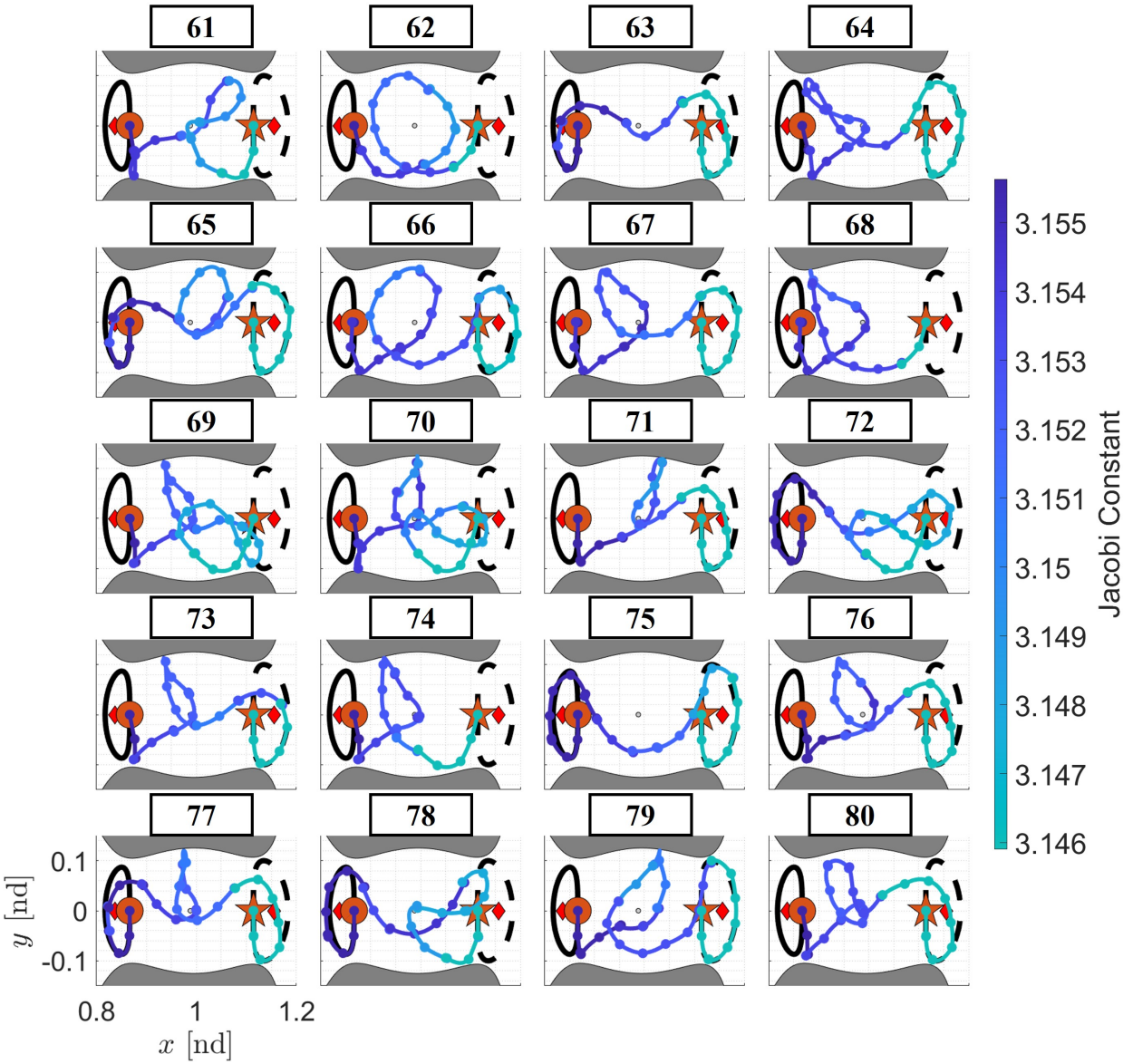

Figure 9.18: Initial guesses 61-80 with distinct geometries for spacecraft trajectories connecting boundary conditions along an L_1 Lyapunov orbit at $C_J = 3.1556$ and L_2 Lyapunov orbit at $C_J = 3.1459$ in the Earth-Moon CR3BP.

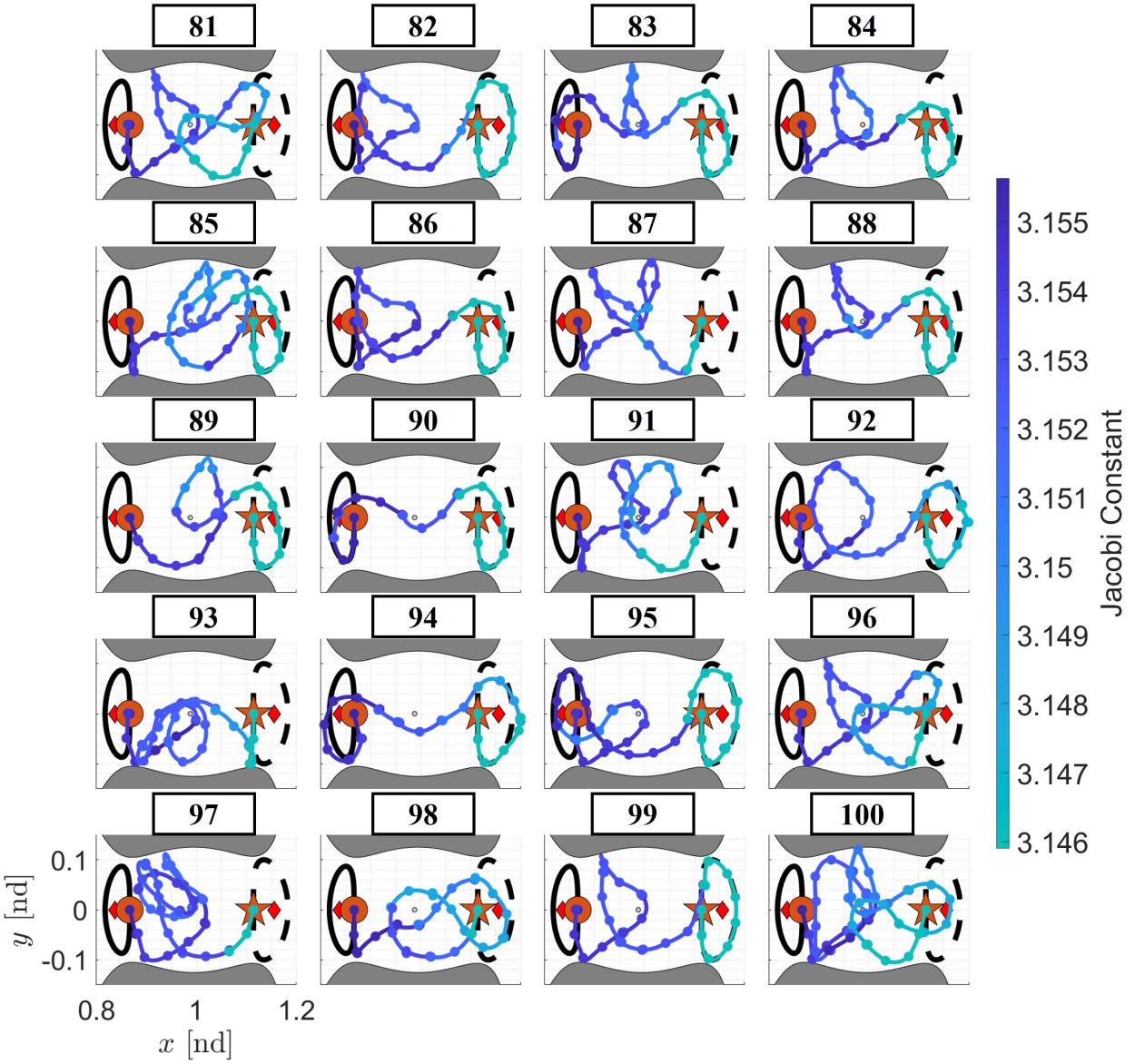

Figure 9.19: Initial guesses 81-100 with distinct geometries for spacecraft trajectories connecting boundary conditions along an L_1 Lyapunov orbit at $C_J = 3.1556$ and L_2 Lyapunov orbit at $C_J = 3.1459$ in the Earth-Moon CR3BP.

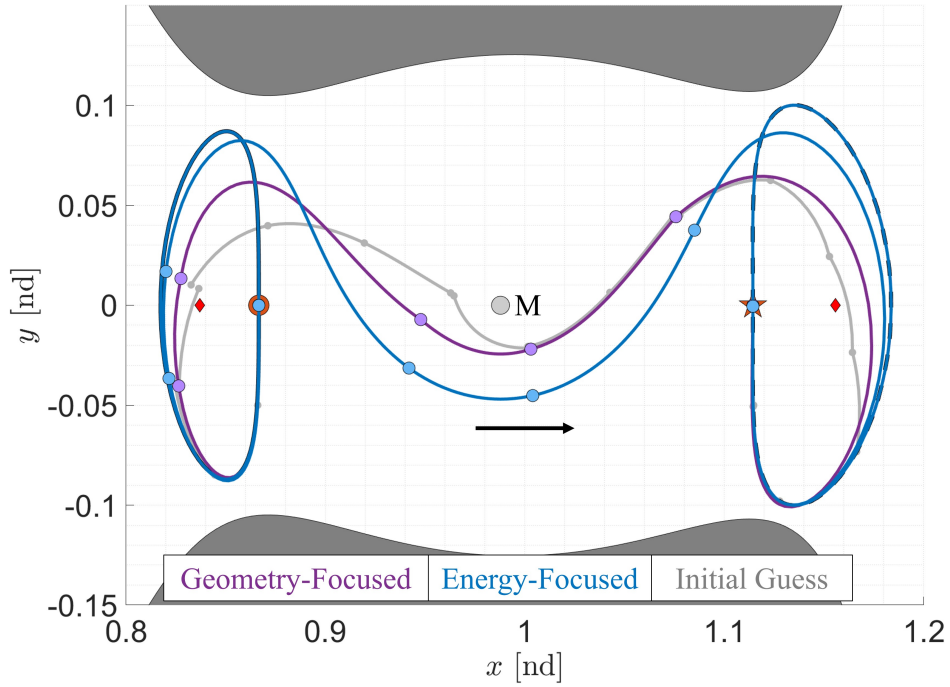

Figure 9.20: The evolution of a zero-revolution transfer that connects boundary conditions on L_1 and L_2 Lyapunov orbits at different C_J: initial guess (gray), geometry-focused solution (purple) with Δv equal to 252.60 m/s and TOF equal to 23.08 days, and energy-focused solution (blue) with Δv equal to 14.04 m/s and TOF equal to 26.49 days.

the energy-efficient solution in Figure 9.11. Identifying similar geometries in continuous trajectories at different energy levels further suggests that geometries in this solution space exist and evolve across multiple energy levels.

Additionally, the initial guesses with the 62nd and 72nd lowest values of the relative velocity discontinuity are corrected for analysis. The coarse initial guesses and corresponding transfers are depicted in Figures 9.22 - 9.23. For both examples, 11 maneuvers are manually placed along the initial guess. In Figure 9.22, there is little difference in the path followed by the geometry- to energy-focused transfers, even as the total maneuver magnitude reduces from 315.31 m/s to 24.66 m/s, as listed in Table 9.4. This characteristic may be a result of the geometric similarity of the initial guess to a known single-revolution heteroclinic connection at a nearby, single value of the Jacobi constant [26]. In Figure 9.23, the geometry-focused transfer closely resembles the initial guess. However, the energy-focused trajectory noticeably differs from the initial guess near apolune

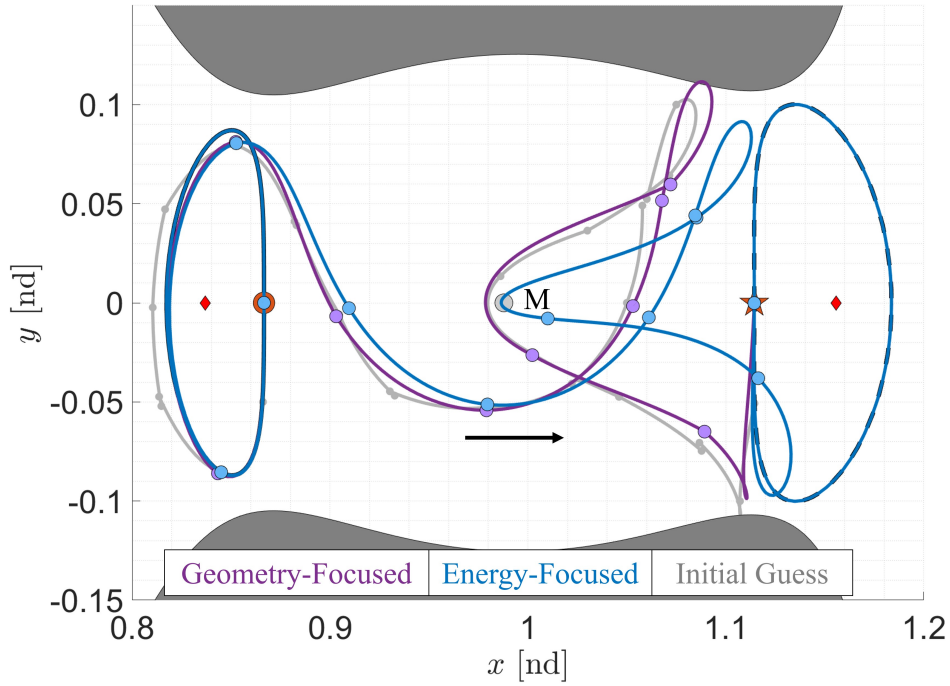

Figure 9.21: The evolution of a single-revolution transfer with a high apolune that connects boundary conditions on L_1 and L_2 Lyapunov orbits at different C_J: initial guess (gray), geometry-focused solution (purple) with Δv equal to 312.98 m/s and TOF equal to 26.92 days, and energy-focused solution (blue) with Δv equal to 21.78 m/s and TOF equal to 28.02 days.

as the total maneuver magnitude reduces by over half, as listed in Table 9.4. However, the high-level geometry of this transfer is retained through corrections and continuation.

Overall, transfers spanning multiple energy levels retain the geometry of the initial guess throughout the corrections and continuation process more often than transfers spanning one energy level. Unlike environments at a single value of the Jacobi constant, speed changes are captured in the transfers presented in these multi-energy scenarios, which may aid in preserving geometry. Furthermore, the continuous span of the desired range of the Jacobi constant in the forest promotes smaller changes in speed across trees compared to constructing multiple layers of the forest at selected intervals of the Jacobi constant.

The variation in Jacobi constant is analyzed across each of the continuous transfers that connect boundary conditions between the selected L_1 and L_2 Lyapunov orbits spanning multiple energy levels. The variation in Jacobi constant across each trajectory is shown in Figure 9.24 with

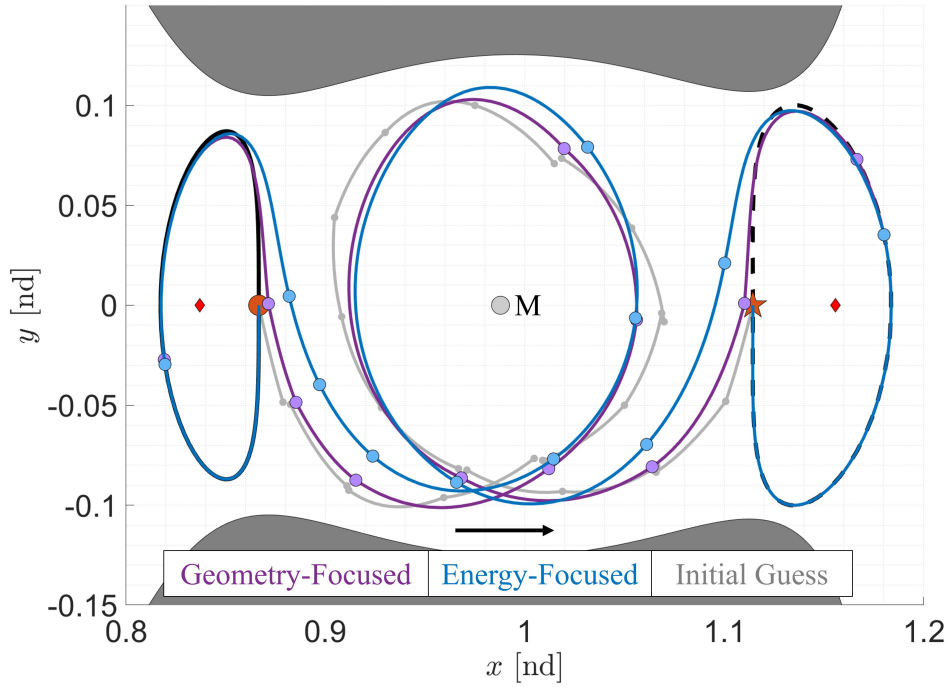

Figure 9.22: The evolution of a low-curvature, single-revolution transfer that connects boundary conditions on L_1 and L_2 Lyapunov orbits at different C_J: initial guess (gray), geometry-focused solution (purple) with Δv equal to 315.31 m/s and TOF equal to 26.81 days, and energy-focused solution (blue) with Δv equal to 24.66 m/s and TOF equal to 26.07 days.

the same configuration as Figure 9.13. However, in this figure $C_J = 3.1556$ is marked with a red dashed line while $C_J = 3.1459$ is denoted with a blue dashed line. Similar to the single-energy scenario presented in Section 9.2.1, the energy-focused transfers possess less variation in the Jacobi constant than the geometry-focused transfers. In particular, the energy-focused transfers recovered from initial guesses 1, 3, and 62 remain between or near the Jacobi constants of the selected L_1 and L_2 Lyapunov orbits. The energy-focused transfer recovered from initial guess 72, on the other hand, departs this range. This departure occurs near arc 30, which corresponds to the region of the trajectory that crosses the ZVCs plotted at $C_J = 3.1459$ in Figure 9.23.

9.2.2.1 Identification of Collision Trajectories

Consistent with the example in Chapter 8, the same 18 dynamic obstacles that are plotted in Figure 9.3 of Section 9.1 are compared to the continuous transfers in Figures 9.20 - 9.23. In

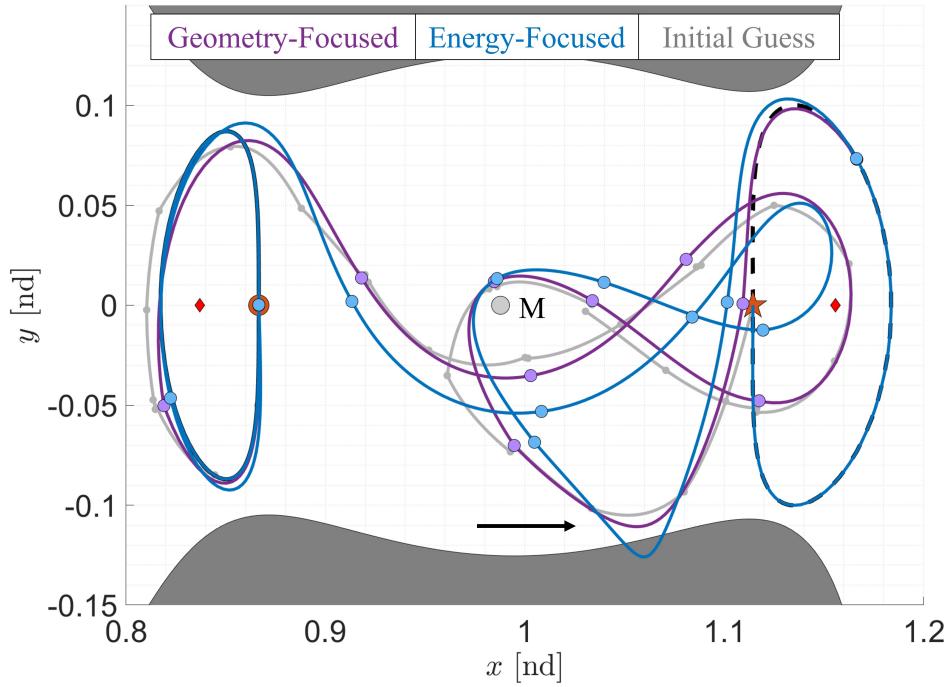

Figure 9.23: The evolution of a geometrically-distinct transfer that connects boundary conditions on L_1 and L_2 Lyapunov orbits at different C_J: initial guess (gray), geometry-focused solution (purple) with Δv equal to 313.01 m/s and TOF equal to 33.12 days, and energy-focused solution (blue) with Δv equal to 150.62 m/s and TOF equal to 34.43 days.

this example, the energy-focused transfer in Figure 9.22 collides with two obstacle corridors if a spacecraft leverages these transfers 9.57 hours or 13.82 hours, respectively, before t_{obj}, the epoch associated with the possible initial states of the observed object near the DPO. These collision paths are depicted in Figures 9.25 - 9.26 where the color gradients denote the elapsed time along each path, filled black circles locate maneuvers, arrows provide the directions of motion, a filled pink triangle denotes the location along the continuous spacecraft path at t_{obj}, and a filled pink square locates the location of the dynamic obstacle along its corridor at t_{obj}. In this example, only the energy-focused trajectory collides with the dynamic obstacles at the red cross marker; therefore, the spacecraft may be able to avoid these collisions while still following a specific type of transfer by using the geometry-focused solution. However, if the total Δv along this geometry-focused transfer is too large, a transfer generated from another initial guess, such as those presented in Figures 9.15 - 9.19, could be analyzed to assess whether it offers an alternative, collision-free option.

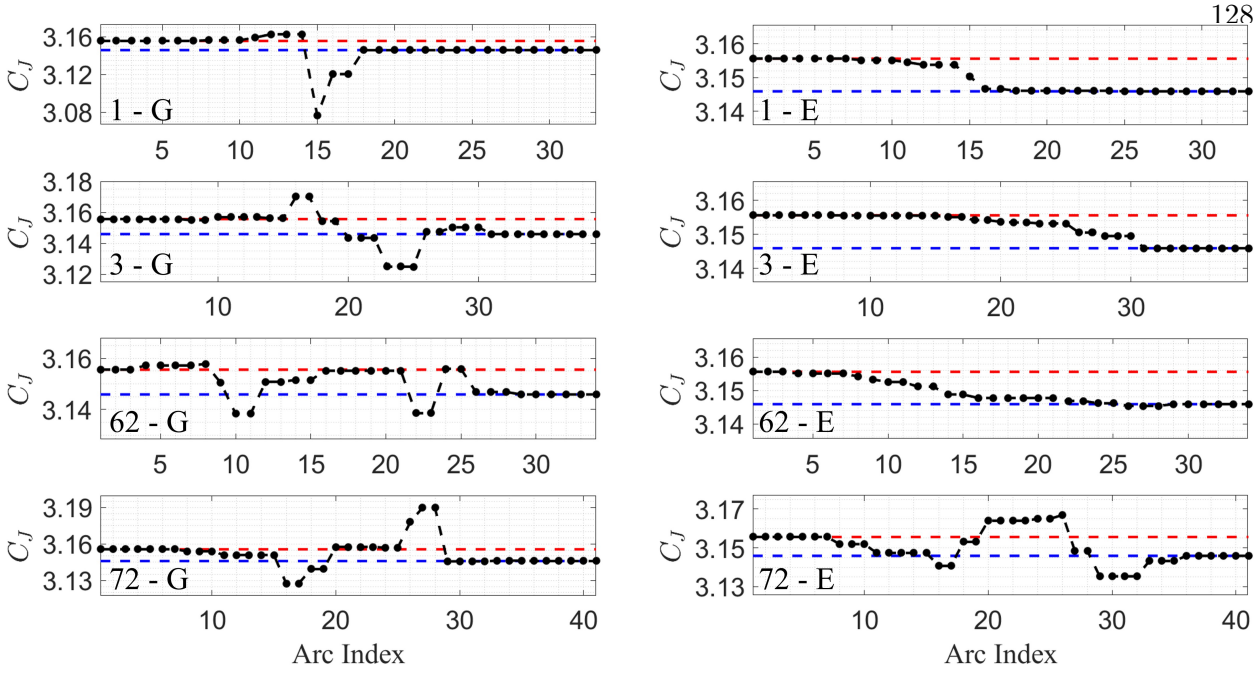

Figure 9.24: Evolution of the Jacobi constant along the geometry- (G) and energy-focused (E) solutions for transfers connecting an L_1 Lyapunov orbit at $C_J = 3.1556$, marked with a dashed red line, and an L_2 Lyapunov orbit at $C_J = 3.1459$, marked with a dashed blue line, in the Earth-Moon CR3BP.

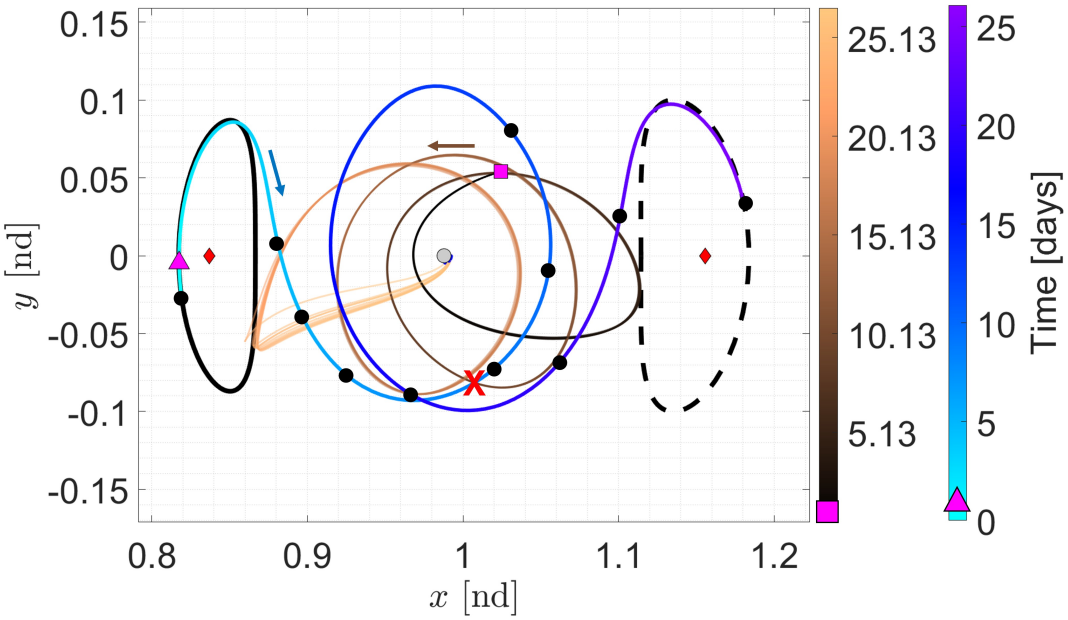

Figure 9.25: The geometry-focused spacecraft transfer recovered from initial guess 1 collides with an obstacle corridor at approximately $t_{coll} = 9.60$ days.

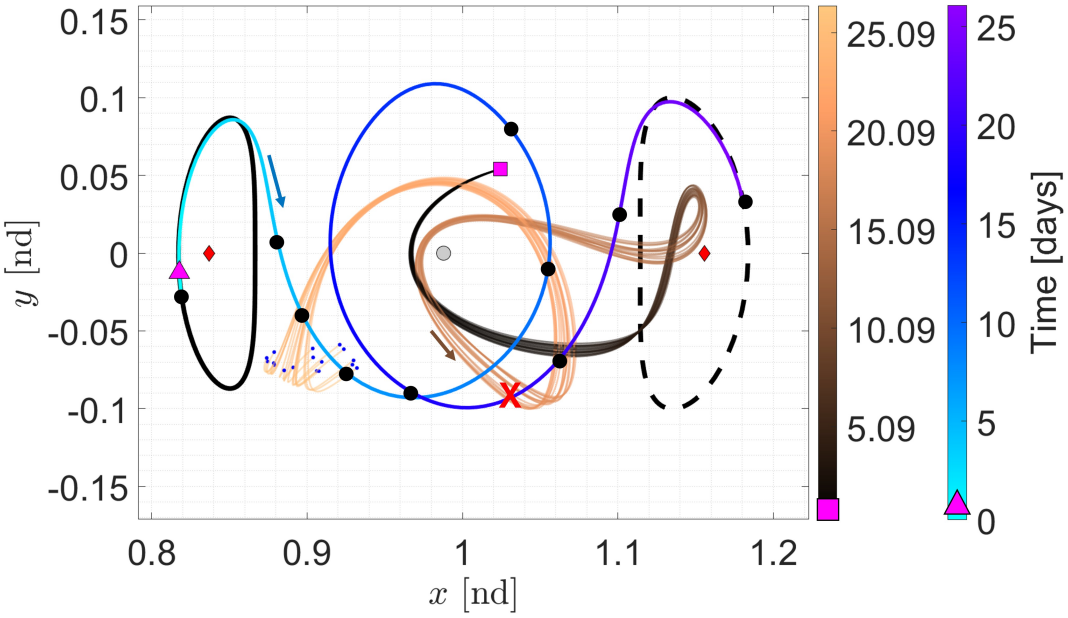

Figure 9.26: The geometry-focused spacecraft transfer recovered from initial guess 1 collides with an obstacle corridor at approximately $t_{coll} = 18.11$ days.

Consider a second scenario where the spacecraft is scheduled to start its transfer 10.98 hours after t_{obj}. The geometry-focused trajectory in Figure 9.23 is identified as a collision trajectory if the observed object follows the motion corridor shown in Figure 9.27 where the collision is denoted with a red cross marker at $t_{coll} = 12.01$ days. By adjusting its departure time, this spacecraft could follow this transfer while avoiding this collision. Otherwise, the energy-focused solution could be leveraged to follow a similar path yet avoid the collision, with a substantially lower total Δv.

A second obstacle is observed in the vicinity of the same DPO 2.29 days later. Therefore, collision checking is again employed to determine the safety of a spacecraft traveling between Lyapunov orbits at different energy levels in the Earth-Moon CR3BP. A spacecraft following the energy-focused trajectory presented in Figure 9.20 would collide with the obstacle 11.51 days later if the obstacle is within the obstacle corridor presented in Figure 9.28. To avoid the collision, alternative itineraries could be considered.

In a final scenario, a spacecraft is scheduled to leverage the geometry-focused transfer in Figure 9.21 8.28 hours before the observation of a spacecraft in the vicinity of L_1. The esti-

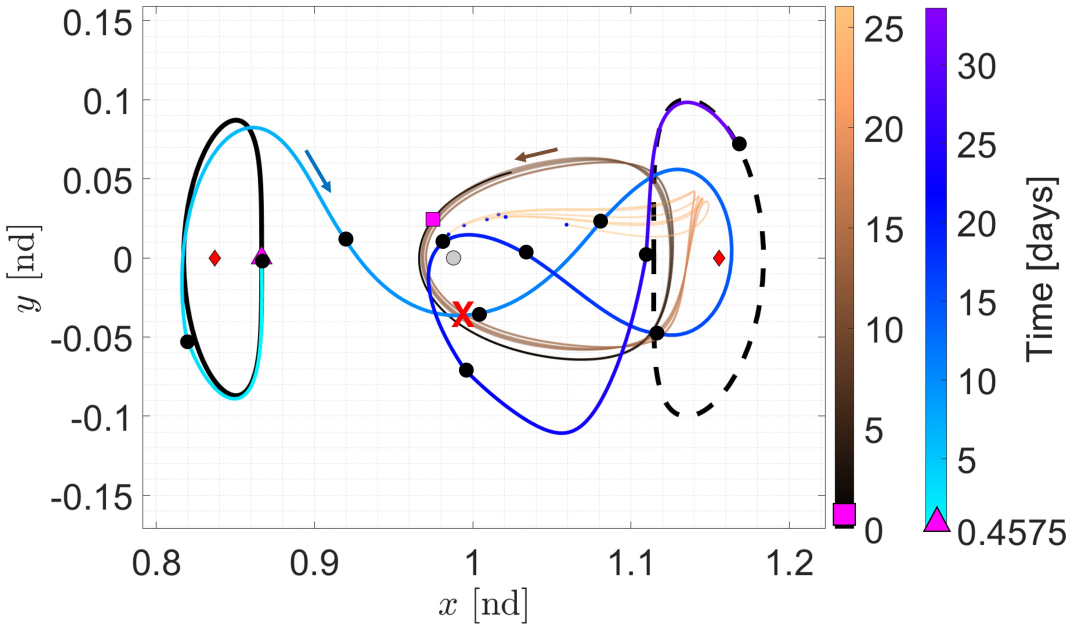

Figure 9.27: The geometry-focused spacecraft transfer recovered from initial guess 72 collides with an obstacle corridor at approximately $t_{coll} = 12.01$ days.

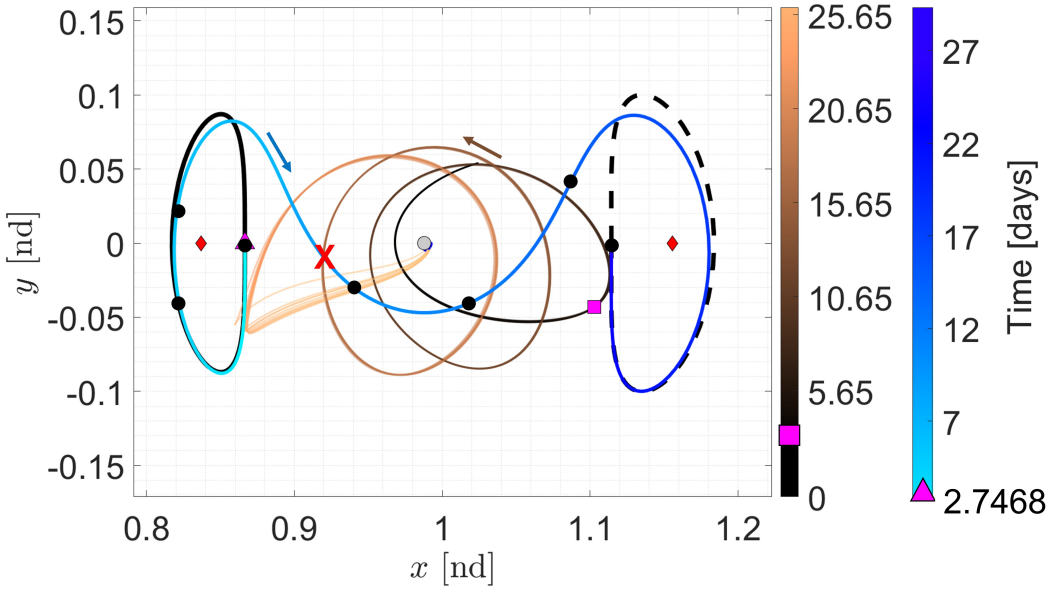

Figure 9.28: The geometry-focused spacecraft transfer recovered from initial guess 1 collides with an obstacle corridor at approximately $t_{coll} = 14.26$ days.

mate of the object's state at its observation epoch, t_{obj}, is $\bar{x}_{obj}(t_{obj}) = [0.816988444235, 0, 0, 0, 0.195756600373, 0]^T$ nondimensional units. The dynamic obstacle corridors associated with this observed spacecraft are provided in Figure 6.11. Upon analysis for collisions with the desired transfer, a collision is detected at $t_{coll} = 14.01$ days past the start of the transfer. Figure 9.29 displays the collision scenario. The collision can be avoided by adjusting the departure time of the spacecraft along this itinerary or evaluating an alternative trajectory to determine if it would provide a collision-free path.

9.3 Spatial Transfers Between L_1 and L_2 Northern Halo Orbits

Spatial transfers connecting periodic orbits in the Earth-Moon CR3BP are designed using the hierarchical tree-based approach. Consider designing a transfer between an 11.99-day L_1 northern halo orbit and a 14.80-day L_2 northern halo orbit at $C_J = 3.1486$ with boundary conditions located at $\bar{x}_{init} = [0.86270116846605, -0.0636380951699078, -0.0225545843958819, -0.0352699550755518, -0.101522330019963, 0.103361518534065]^T$ nondimensional units and $\bar{x}_{targ} = [1.11739052000141,$

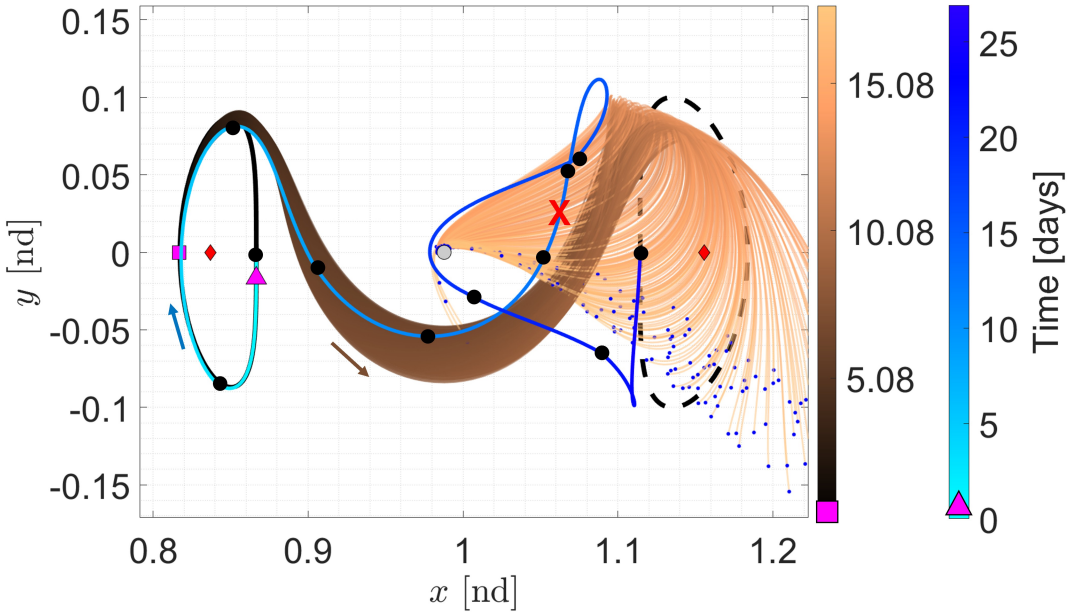

Figure 9.29: The geometry-focused spacecraft transfer recovered from initial guess 3 collides with an obstacle corridor at approximately $t_{coll} = 14.01$ days.

$0.00636863101120599, -0.0201820985356166, 0.000516716132122006, 0.184154882350836, 0.0042384$
$4301013162]^T$ nondimensional units. In this scenario, the forest consists of 19,041 trees. Fifty initial guesses are recovered from unique tree sequences, composed of six to 32 trees from the forest. The initial guesses provided in Figures 9.30 - 9.38 are recovered from sequences consisting of six to eight trees while Figures 9.39 - 9.42 contain discontinuous paths found in longer tree sequences. The left column in each figure provides a spatial view of the initial guesses while the right column presents a projection of the initial guesses onto the xz-plane to aid visualization. In each subfigure, the initial and target states are represented by an orange circle and star, respectively, and the branches from distinct trees alternate between light and dark blue colors in sequential order. These discontinuous paths smoothly depart from and arrive onto the halo orbits, which is a result of the velocity cone constraint used to build and connect trees. Furthermore, each path exhibits a different geometry. The revolutions vary by shape, such as possessing high apolunes in initial guess 4 or exhibiting lower curvature revolutions in initial guess 36; high versus low inclination as seen in the xz-plane for initial guesses 3, 16, and 22; and rotation or twisting, such as the difference between initial guesses 14, 31, and 33. Initial guesses 2, 11, 16, 18, and 34 are used to generate continuous transfers. The characteristics of these initial guesses and their associated transfers are summarized in Table 9.5.

Table 9.5: Summary of transfer characteristics connecting L_1 and L_2 northern halo orbits at $C_J = 3.1468$.

Fig. #	**Initial Guess**		**ID # - Priority**	**Corrected Path**	
	Δv [m/s]	**TOF [days]**		**Δv [m/s]**	**TOF [days]**
9.43	1,730.57	19.92	2 - Geometry	692.91	22.46
			2 - Energy	184.29	24.67
9.44	2,536.01	22.49	34 - Geometry	569.18	28.76
			34 - Energy	241.53	30.42
9.45	2,718.44	15.41	16 - Geometry	782.31	17.64
			16 - Energy	382.33	24.82
9.46	2,716.95	23.25	18 - Geometry	891.93	24.31
			18 - Energy	188.31	25.31
9.47	1,875.17	20.51	11 - Geometry	585.16	19.46
			11 - Energy	93.24	27.11

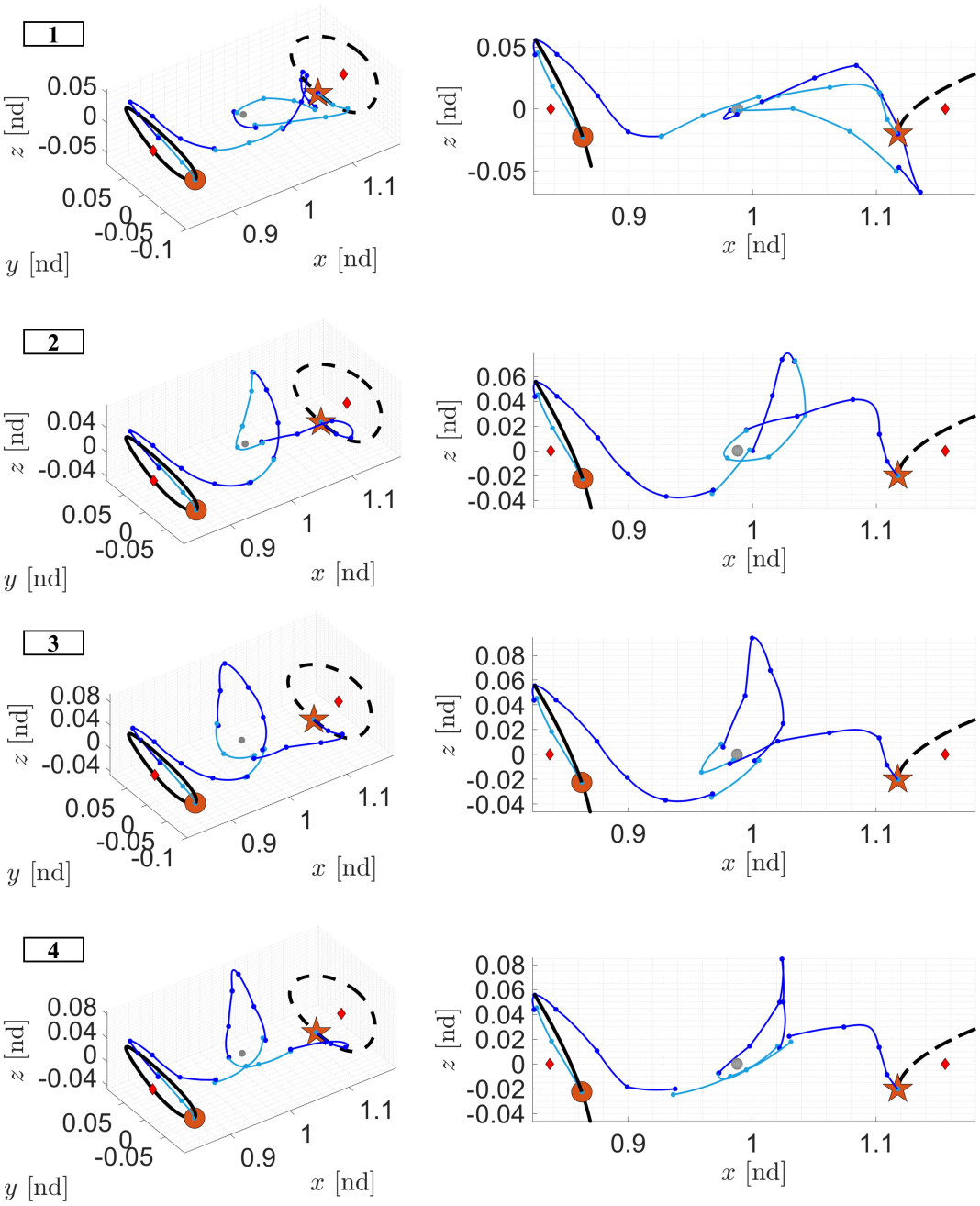

Figure 9.30: Initial guesses 1-4 for geometrically-distinct spacecraft trajectories connecting boundary conditions along L_1 and L_2 northern halo orbits at $C_J = 3.1486$ in the Earth-Moon CR3BP.

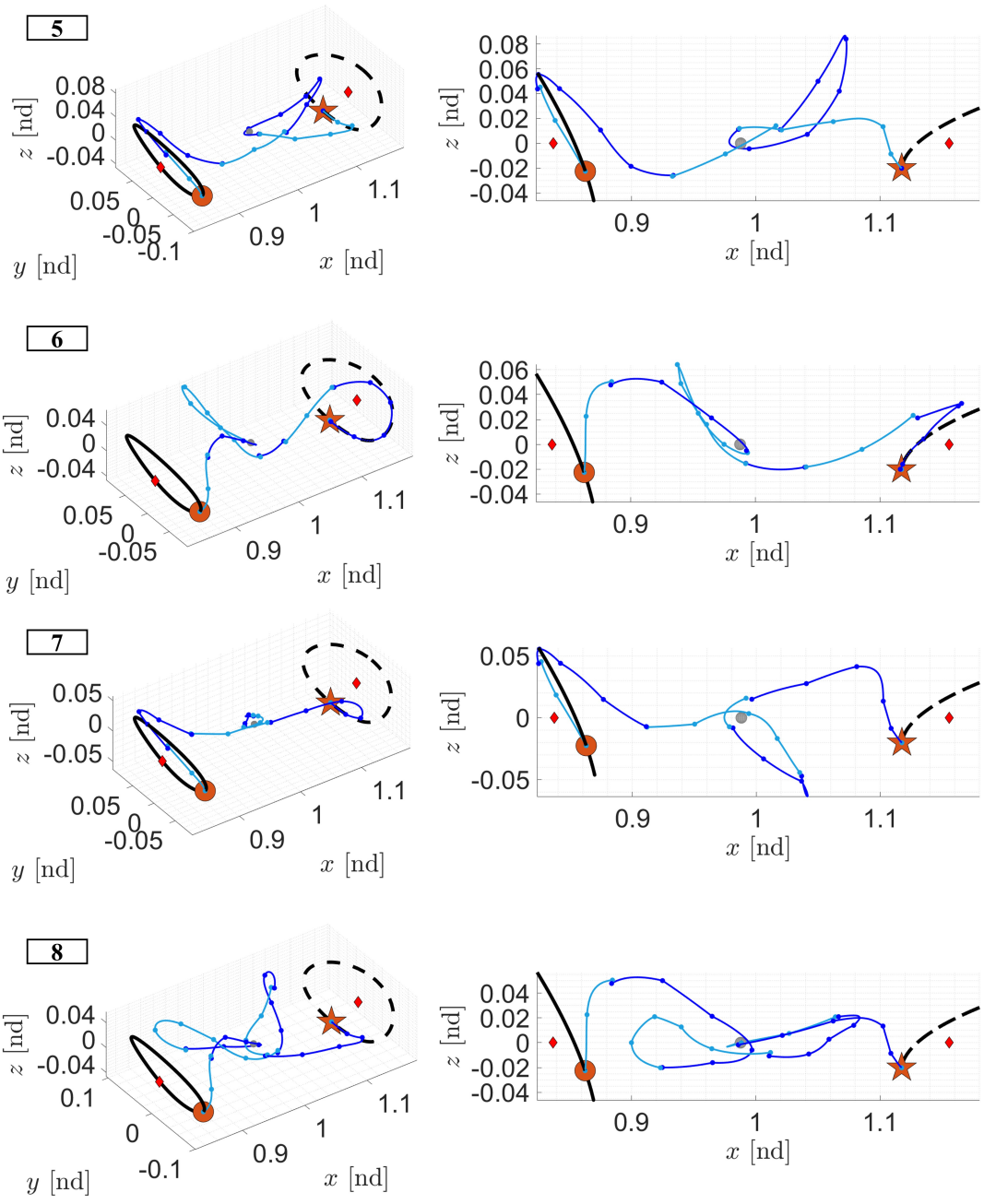

Figure 9.31: Initial guesses 5-8 for geometrically-distinct spacecraft trajectories connecting boundary conditions along L_1 and L_2 northern halo orbits at $C_J = 3.1486$ in the Earth-Moon CR3BP.

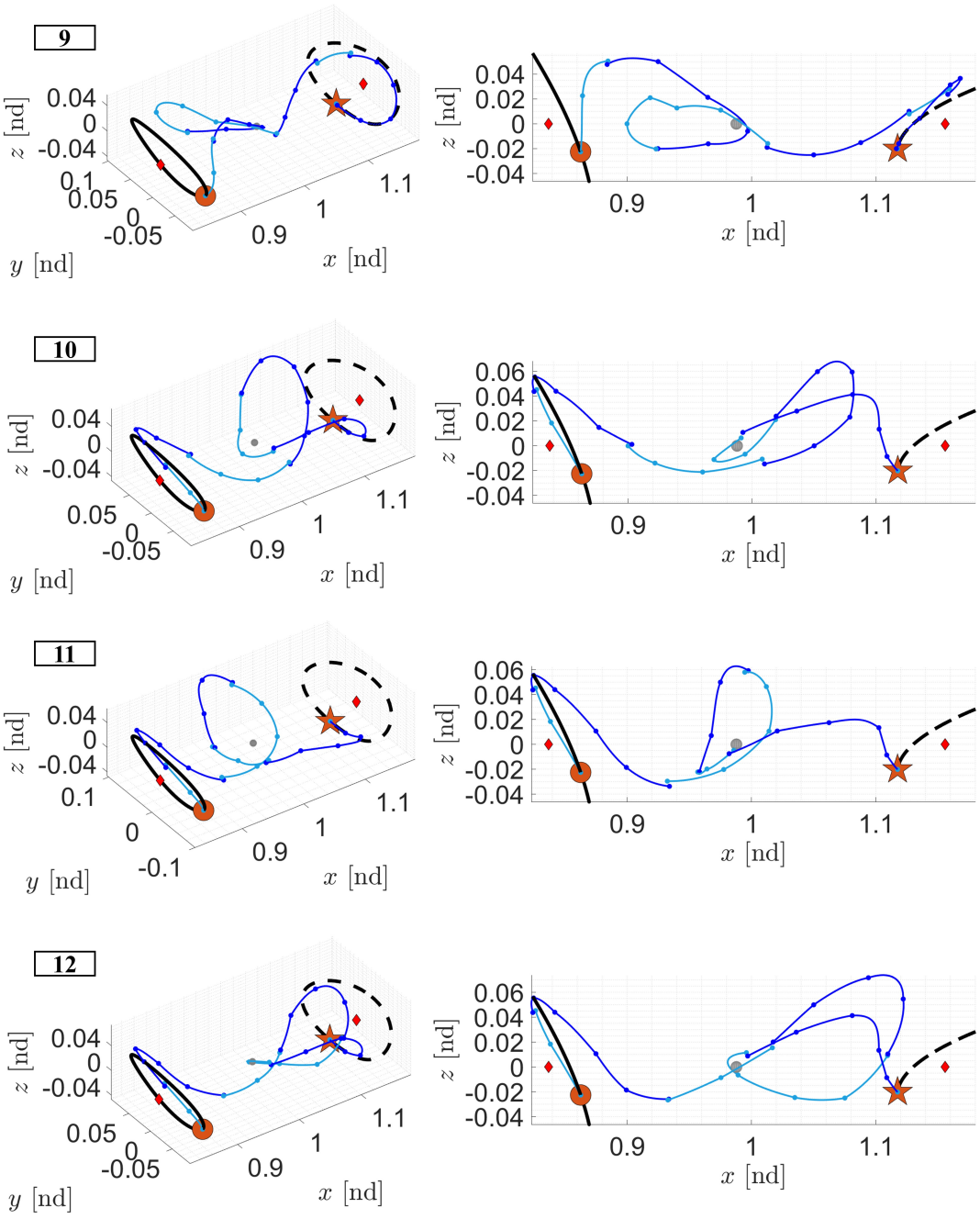

Figure 9.32: Initial guesses 9-12 for geometrically-distinct spacecraft trajectories connecting boundary conditions along L_1 and L_2 northern halo orbits at $C_J = 3.1486$ in the Earth-Moon CR3BP.

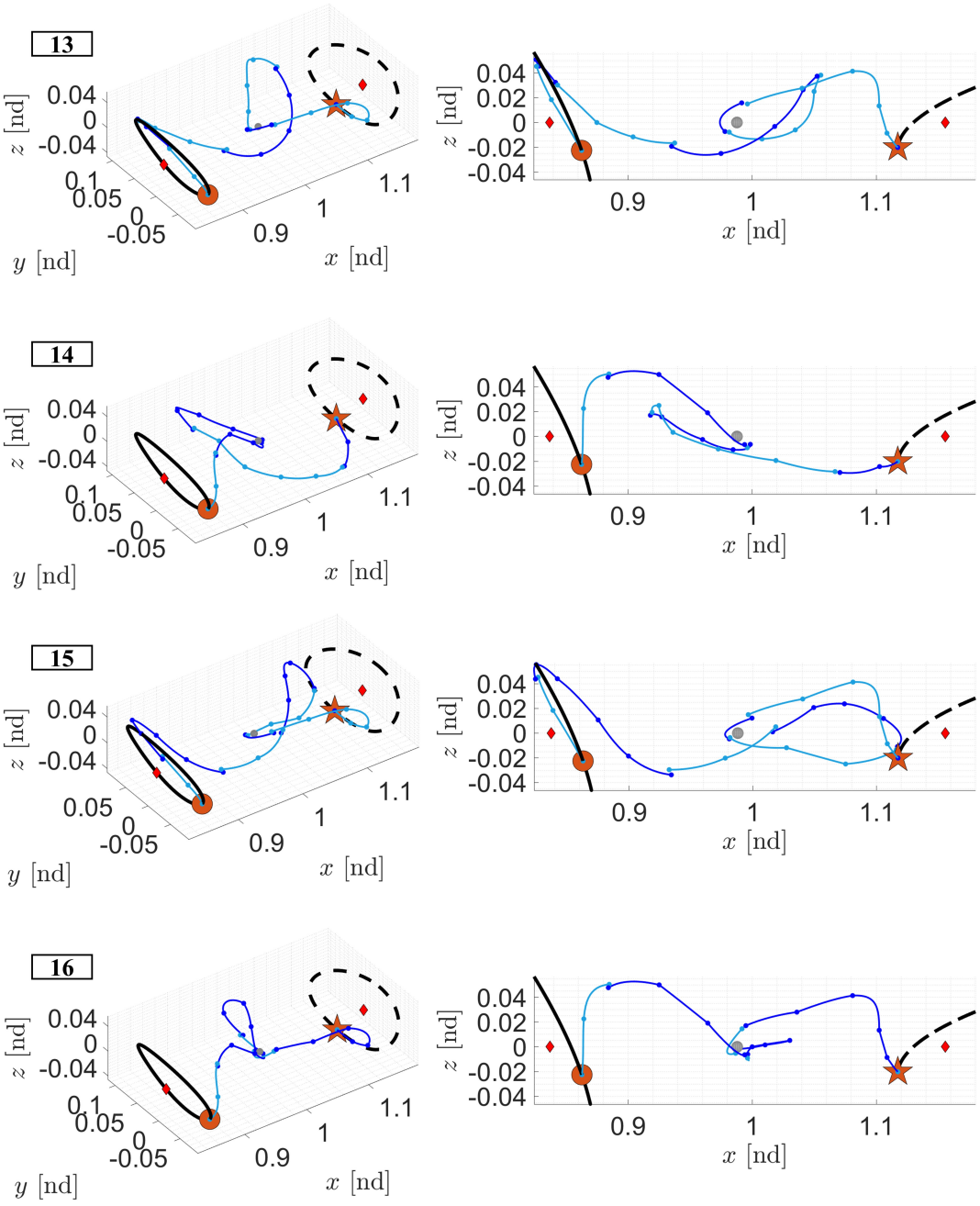

Figure 9.33: Initial guesses 13-16 for geometrically-distinct spacecraft trajectories connecting boundary conditions along L_1 and L_2 northern halo orbits at $C_J = 3.1486$ in the Earth-Moon CR3BP.

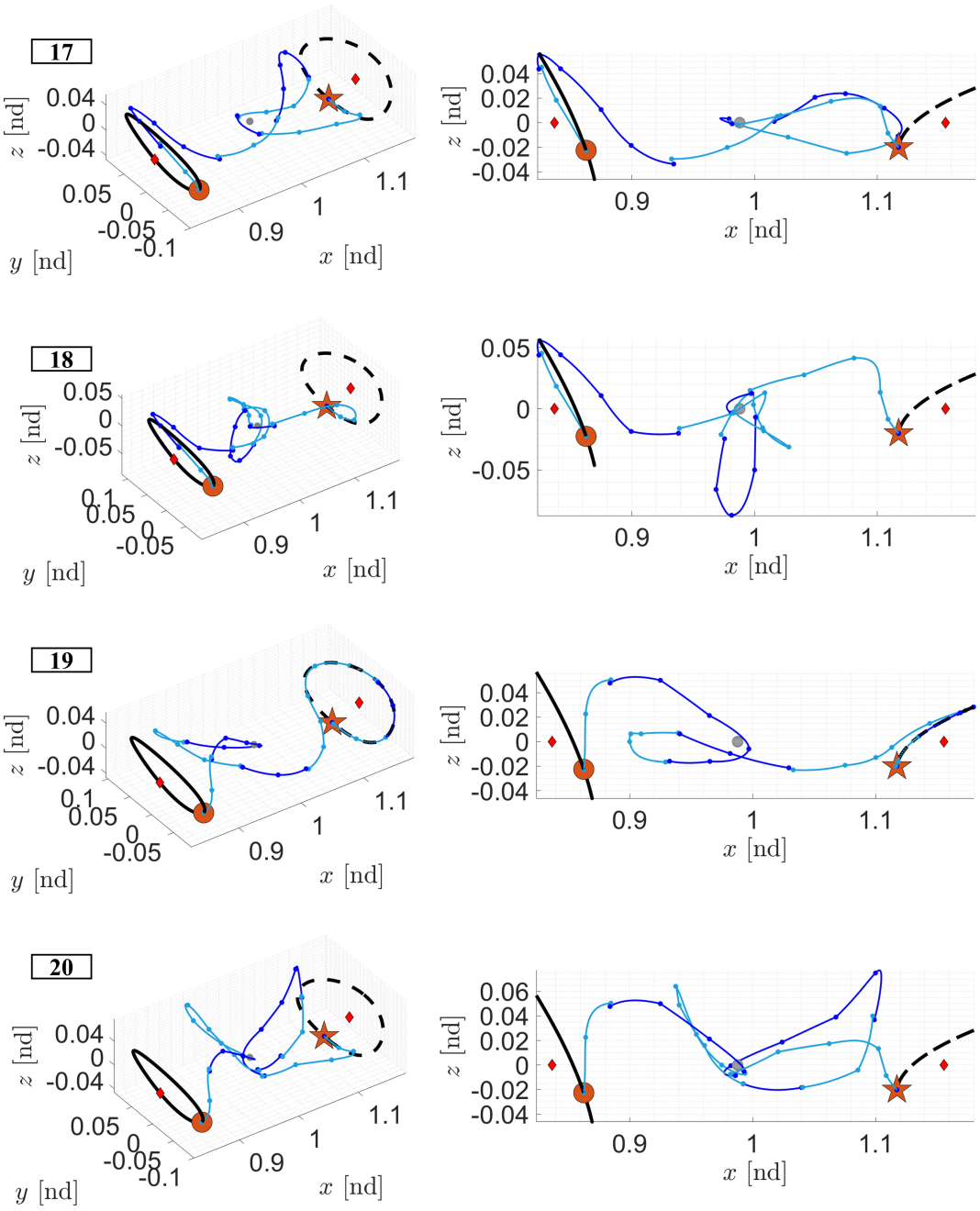

Figure 9.34: Initial guesses 17-20 for geometrically-distinct spacecraft trajectories connecting boundary conditions along L_1 and L_2 northern halo orbits at $C_J = 3.1486$ in the Earth-Moon CR3BP.

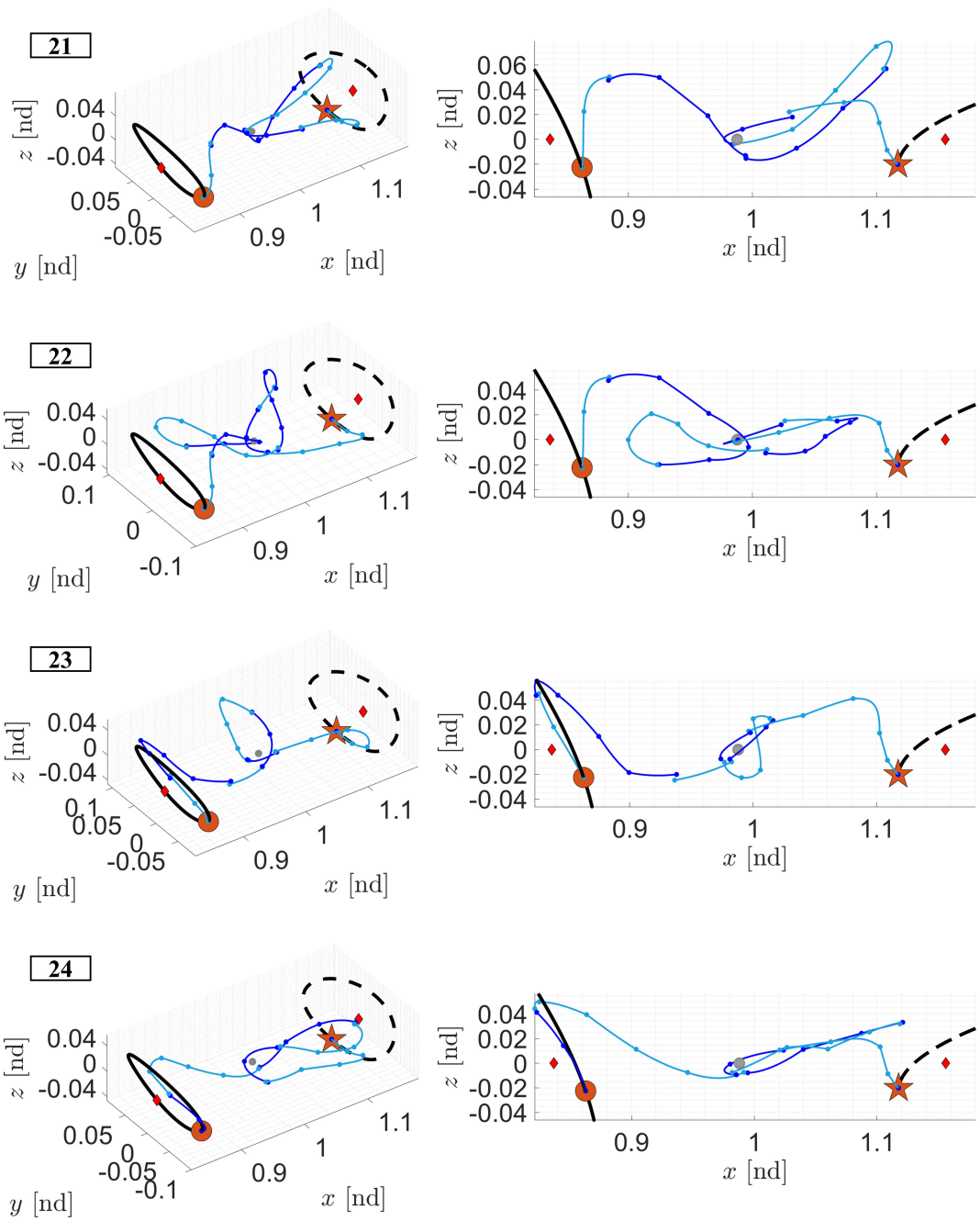

Figure 9.35: Initial guesses 21-24 for geometrically-distinct spacecraft trajectories connecting boundary conditions along L_1 and L_2 northern halo orbits at $C_J = 3.1486$ in the Earth-Moon CR3BP.

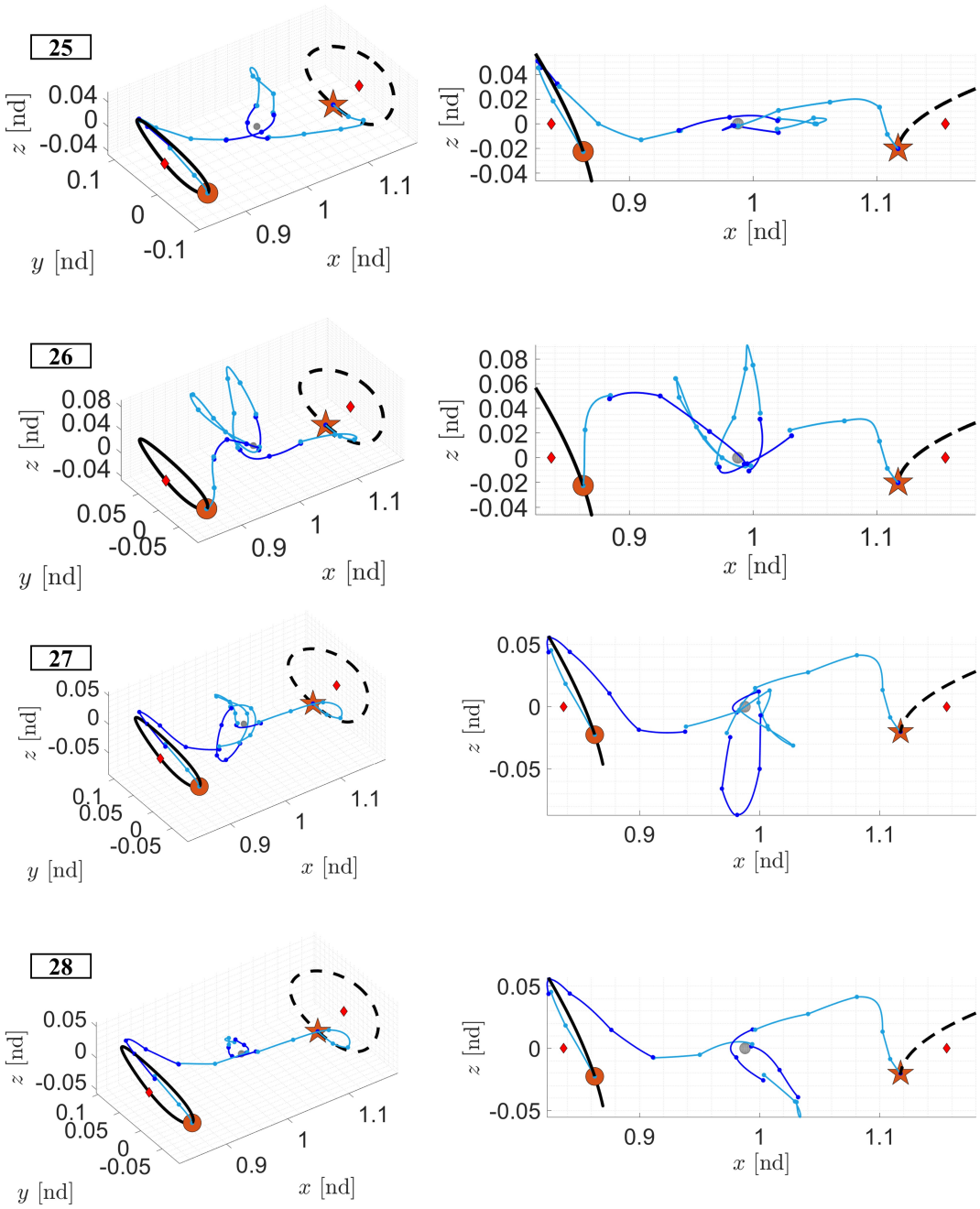

Figure 9.36: Initial guesses 25-28 for geometrically-distinct spacecraft trajectories connecting boundary conditions along L_1 and L_2 northern halo orbits at $C_J = 3.1486$ in the Earth-Moon CR3BP.

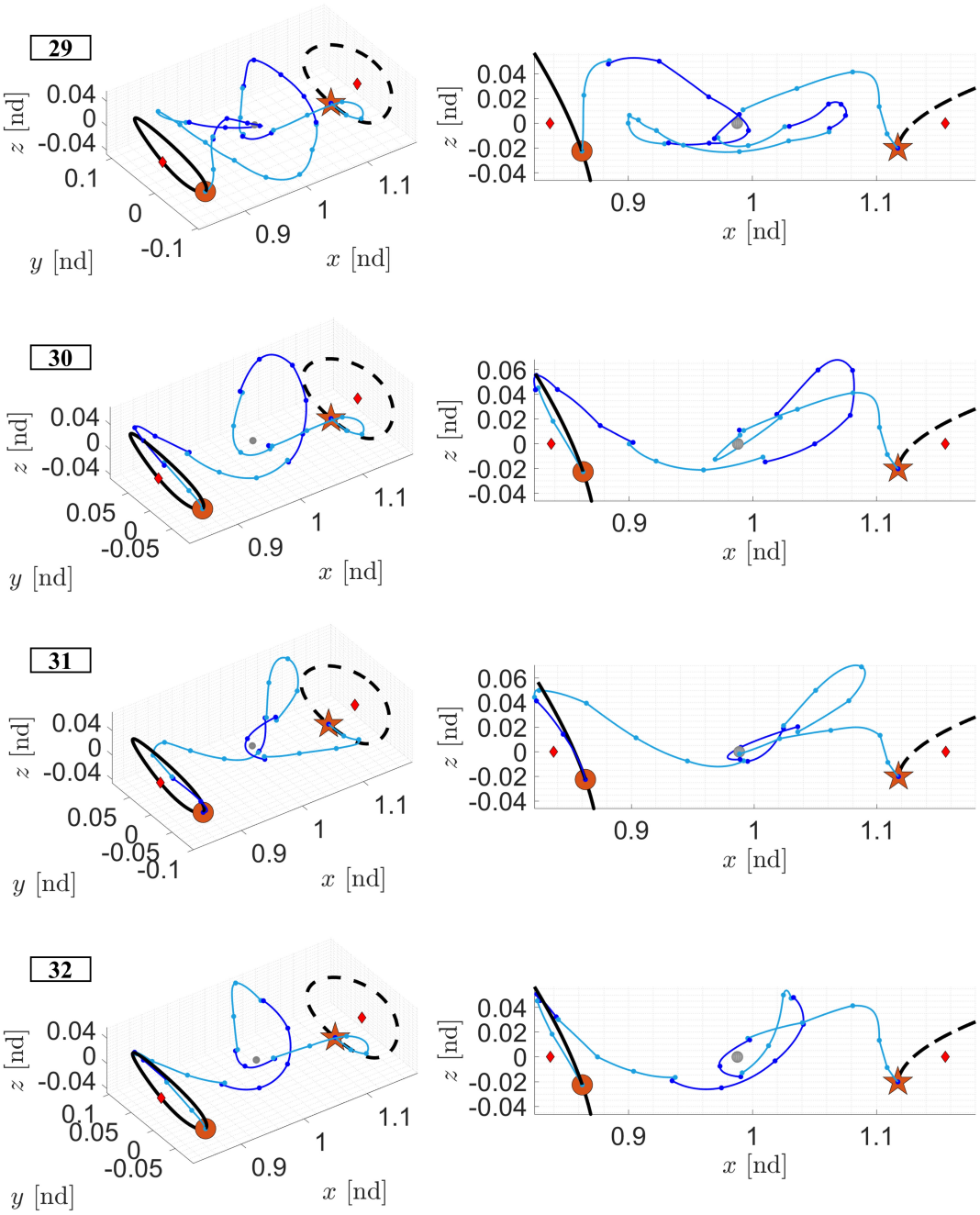

Figure 9.37: Initial guesses 29-32 for geometrically-distinct spacecraft trajectories connecting boundary conditions along L_1 and L_2 northern halo orbits at $C_J = 3.1486$ in the Earth-Moon CR3BP.

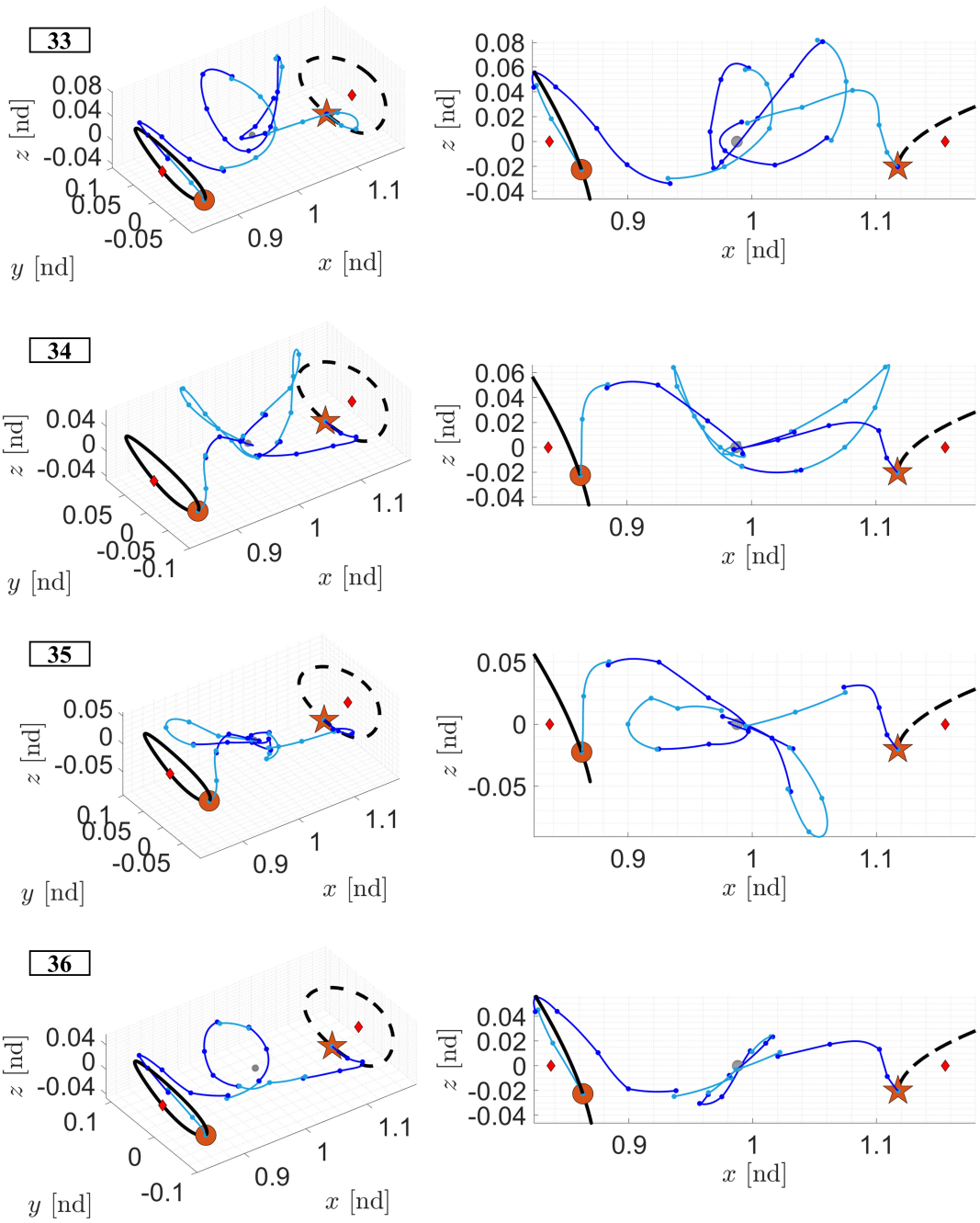

Figure 9.38: Initial guesses 33-36 for geometrically-distinct spacecraft trajectories connecting boundary conditions along L_1 and L_2 northern halo orbits at $C_J = 3.1486$ in the Earth-Moon CR3BP.

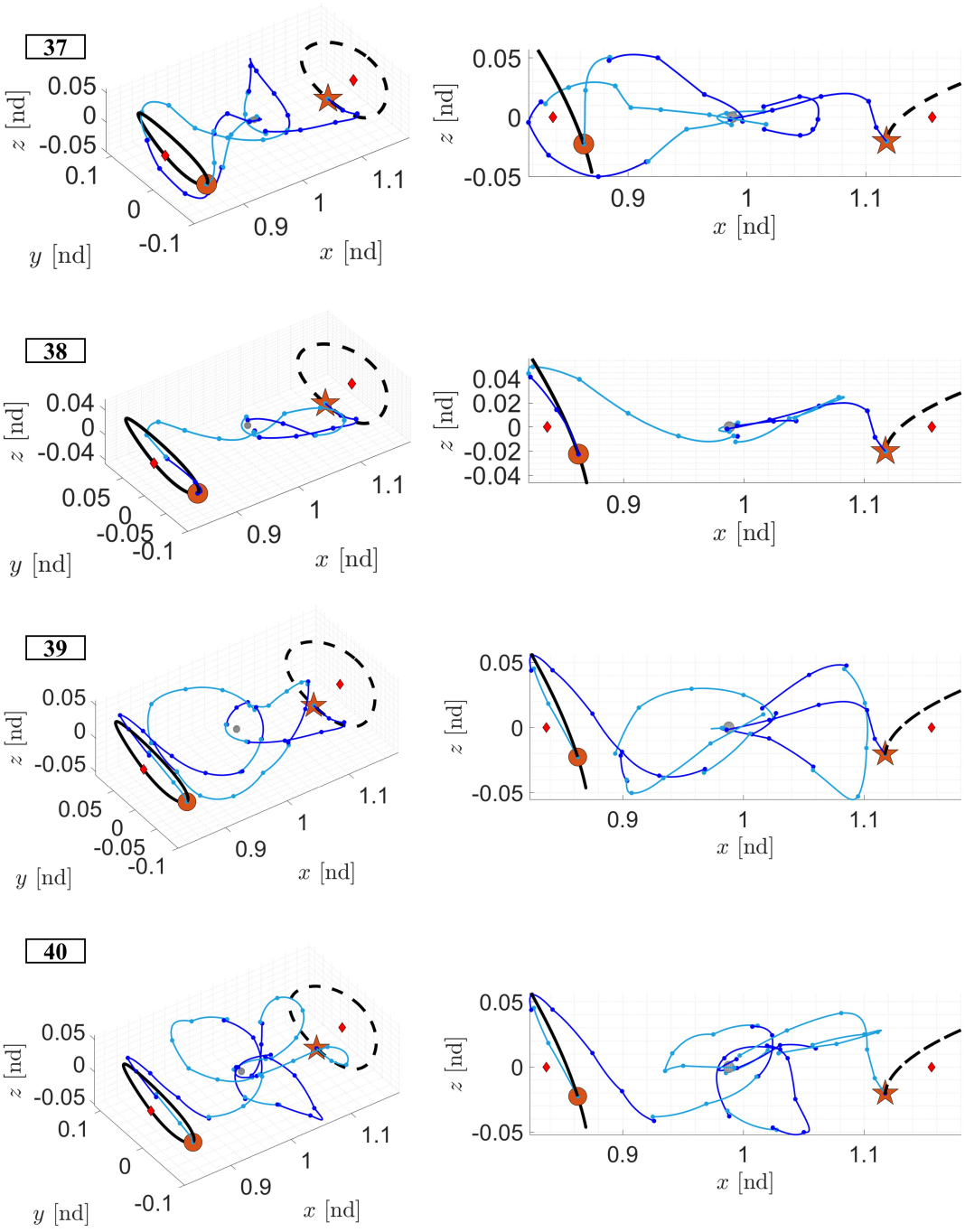

Figure 9.39: Initial guesses 37-40 for geometrically-distinct spacecraft trajectories connecting boundary conditions along L_1 and L_2 northern halo orbits at $C_J = 3.1486$ in the Earth-Moon CR3BP.

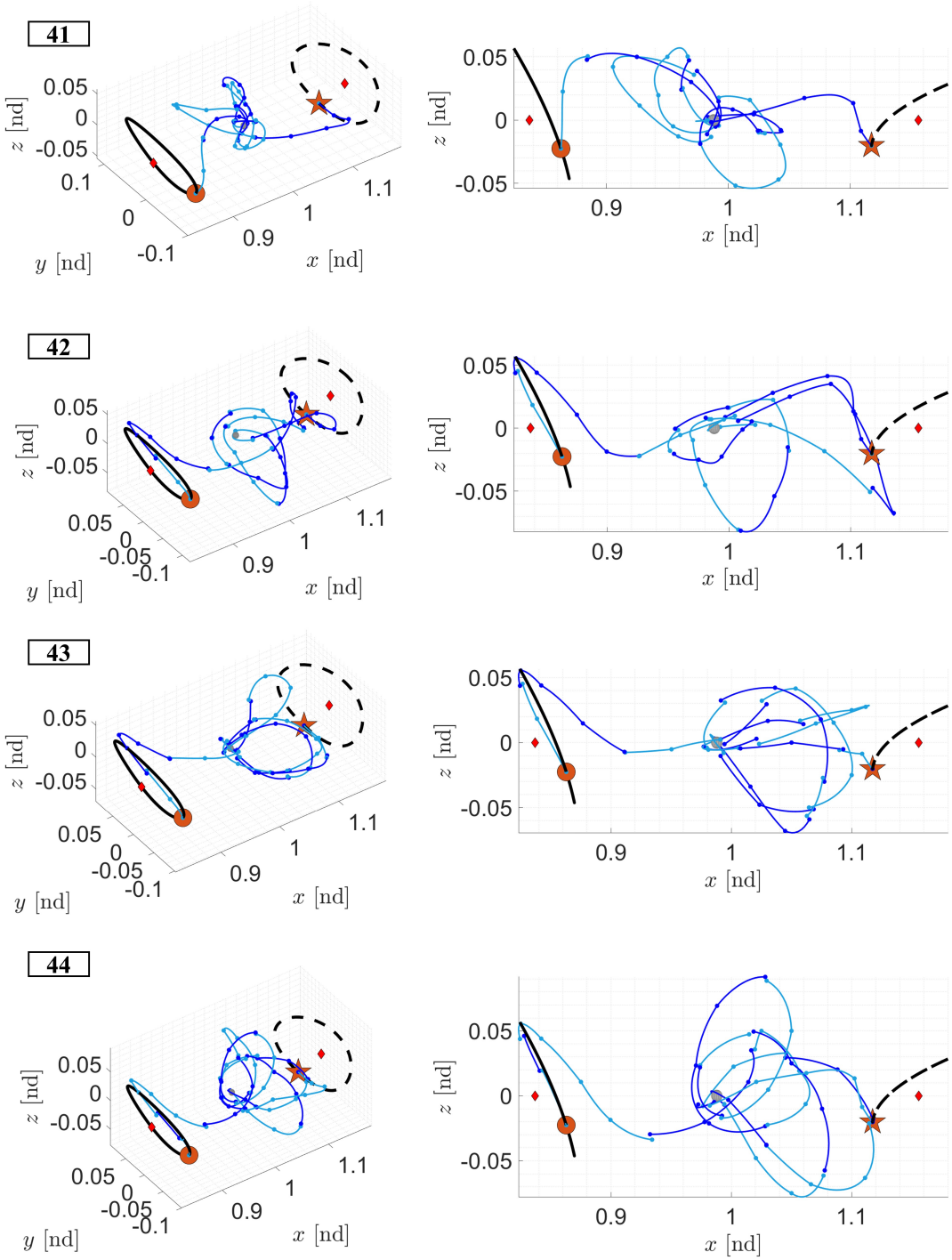

Figure 9.40: Initial guesses 41-44 for geometrically-distinct spacecraft trajectories connecting boundary conditions along L_1 and L_2 northern halo orbits at $C_J = 3.1486$ in the Earth-Moon CR3BP.

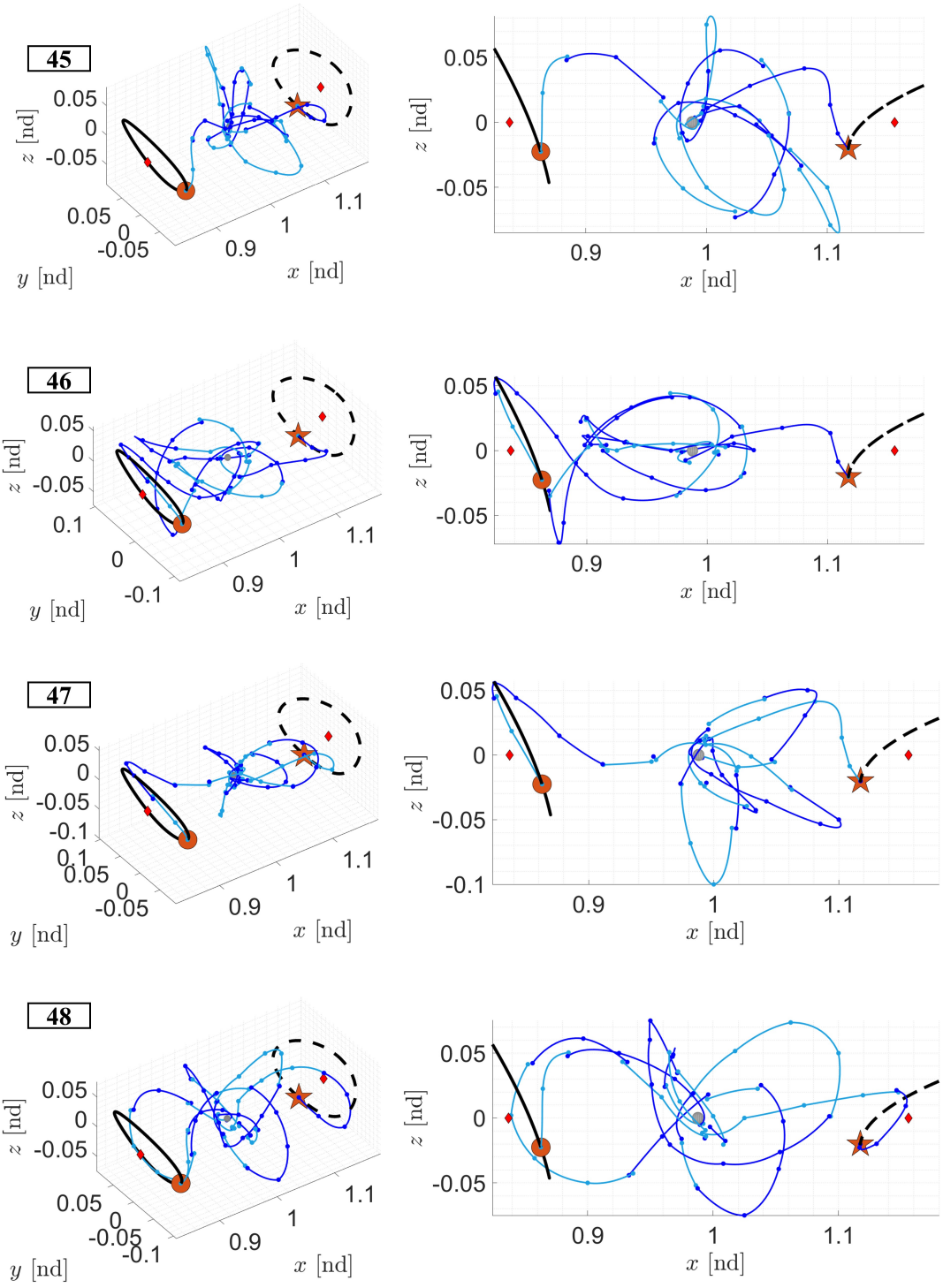

Figure 9.41: Initial guesses 45-48 for geometrically-distinct spacecraft trajectories connecting boundary conditions along L_1 and L_2 northern halo orbits at $C_J = 3.1486$ in the Earth-Moon CR3BP.

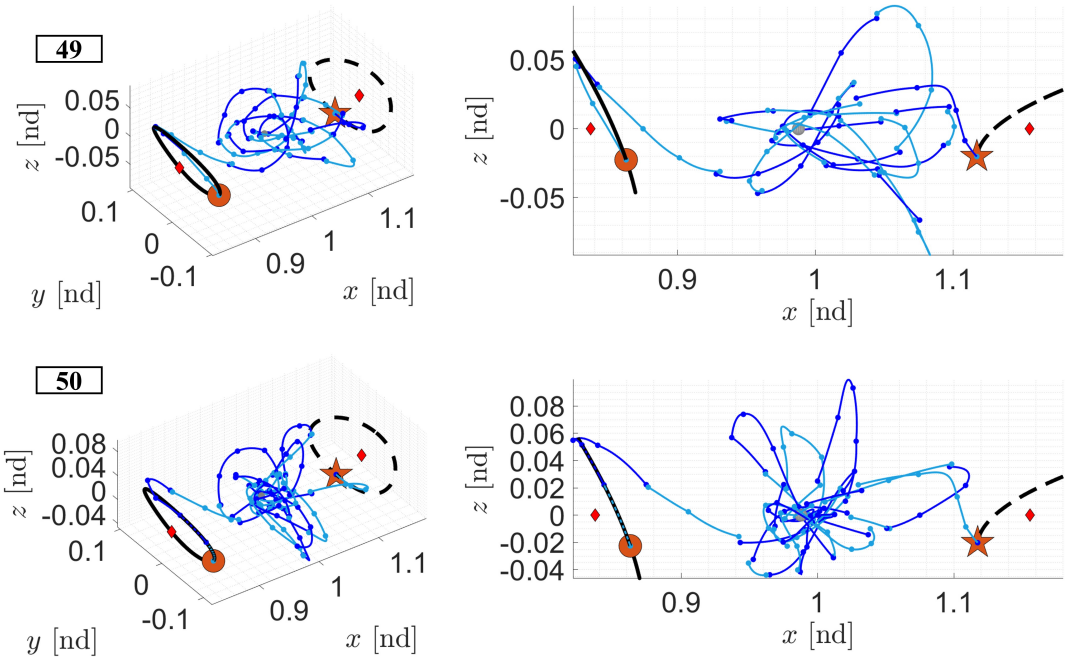

Figure 9.42: Initial guesses 49-50 for geometrically-distinct spacecraft trajectories connecting boundary conditions along L_1 and L_2 northern halo orbits at $C_J = 3.1486$ in the Earth-Moon CR3BP.

The orientation of initial guess 2 with respect to the Moon evolves during corrections. The corresponding transfers, with 10 impulsive maneuvers, are provided in the rightmost subfigures of Figure 9.43 while the initial guess is plotted in the leftmost subfigures. Both a spatial view and projection onto the xz-plane are presented in Figure 9.43 to aid visualization. The geometry-focused trajectory possesses a similar rotation and inclination to the initial guess; however, the inclination is lowered as the total maneuver magnitude is prioritized. Overall, the high-level geometry of the initial guess is preserved.

The transfers generated from initial guess 34 leverage nine impulsive maneuvers and exhibit little variance in their geometry. In fact, Figure 9.44 shows that only minor differences appear in the departure segment from the L_1 northern halo orbit and the inclination of the revolutions with respect to the Moon. As reducing control energy is prioritized, a greater portion of the transfer

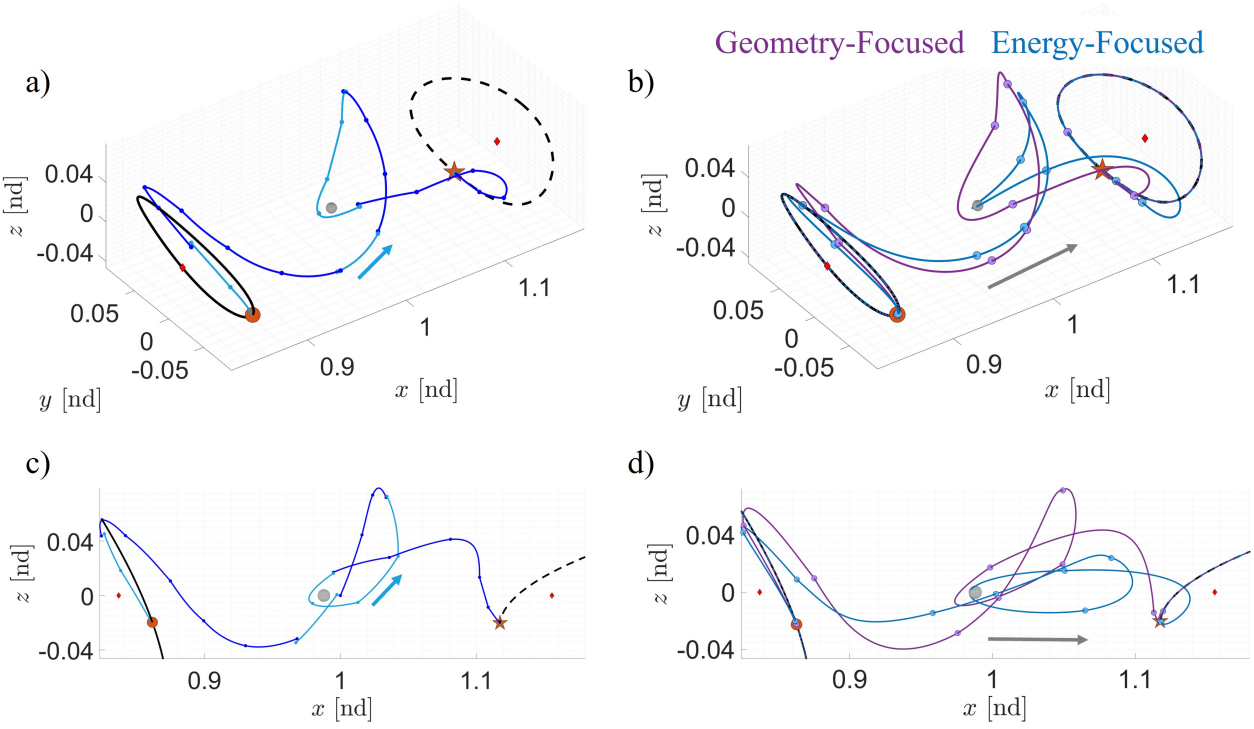

Figure 9.43: a) Spatial representation of initial guess 2. b) Spatial representations of the geometry-focused (purple) and energy-focused (blue) transfers. c) Initial guess 2 projected onto the xz-plane. d) Geometry- and energy-focused transfers projected onto the xz-plane. The geometry-focused transfer possesses a Δv equal to 692.91 m/s with a TOF equal to 22.46 days while the energy-focused solution has a Δv equal to 184.29 m/s and TOF equal to 24.67 days.

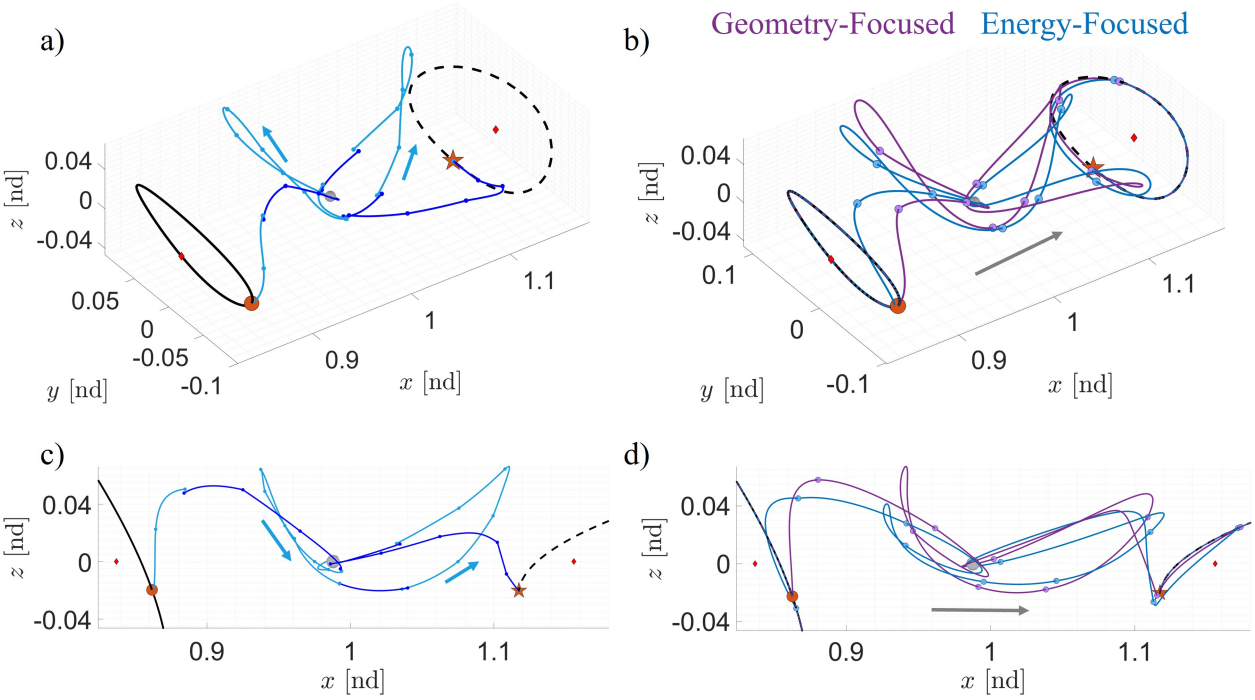

Figure 9.44: a) Spatial representation of initial guess 34. b) Spatial representations of the geometry-focused (purple) and energy-focused (blue) transfers. c) Initial guess 34 projected onto the xz-plane. d) Geometry- and energy-focused transfers projected onto the xz-plane. The geometry-focused path is associated with a Δv equal to 569.18 m/s and TOF equal to 28.76 days. The energy-focused transfer requires a total maneuver magnitude of 241.53 m/s and a flight time of 30.42 days.

remains in the vicinity of the L_1 northern halo orbit during departure and the inclination of the revolutions lowers with respect to the Moon.

Initial guess 16 exhibits a twisted, high-curvature revolution about the Moon. Figure 9.45 displays the initial guess and corrected paths, which leverage eight impulsive maneuvers. The geometry-focused trajectory possesses minimal variance with respect to the initial guess. However, the reduction in total maneuver magnitude between the geometry- and energy-focused trajectories results in a significant variation in size and orientation of the revolution about the Moon. This reduction is equal to 399.98 m/s, which is calculated using the Δv listed in Table 9.5.

The transfers associated with initial guess 18 possess significant geometric evolution from the initial guess. At a high level, the geometry of the initial guess is retained in both transfers shown in Figure 9.46. However, the geometry of the energy-focused solution varies significantly from that

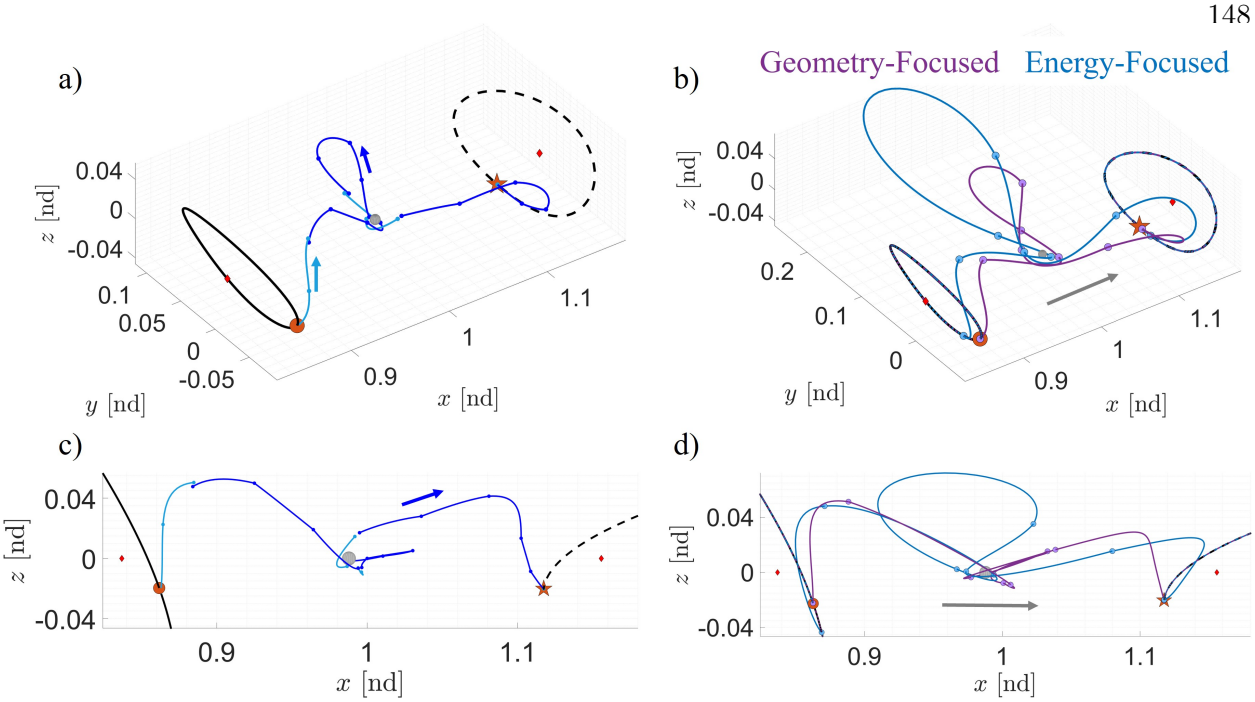

Figure 9.45: a) Spatial representation of initial guess 16. b) Spatial representations of the geometry-focused (purple) and energy-focused (blue) transfers. c) Initial guess 16 projected onto the xz-plane. d) Geometry- and energy-focused transfers projected onto the xz-plane. The solution in purple has a total maneuver magnitude equal to 782.31 m/s with a TOF equal to 17.64 days while the path in blue possesses a total Δv equal to 382.33 m/s with a TOF equal to 24.82 days.

of the initial guess: the revolutions about the Moon are more gradual and oriented at a different inclination with respect to the Moon. Nine maneuvers are manually placed along both solutions.

Significant geometric evolution also occurs between initial guess 11 and its corrected trajectories. Compared to the initial guess, the geometry-focused trajectory is rotated and twisted about the Moon and possesses a higher apolune as seen in Figure 9.47. The geometry of the trajectory continues to evolve as Δv is reduced to recover an energy-focused transfer, which possesses a lower inclination and lower curvature revolution about the Moon than the geometry-focused trajectory.

The variation in Jacobi constant is analyzed across each of the ten continuous trajectories that connect boundary conditions between L_1 and L_2 northern halo orbits at $C_J = 3.1486$ in the Earth-Moon CR3BP. The variation in Jacobi constant across each trajectory is provided in Figure 9.48 with the same configuration as Figure 9.13 where $C_J = 3.1486$ is denoted with a red dashed line. Unlike previous scenarios, not all energy-focused spatial transfers exhibit less variation in

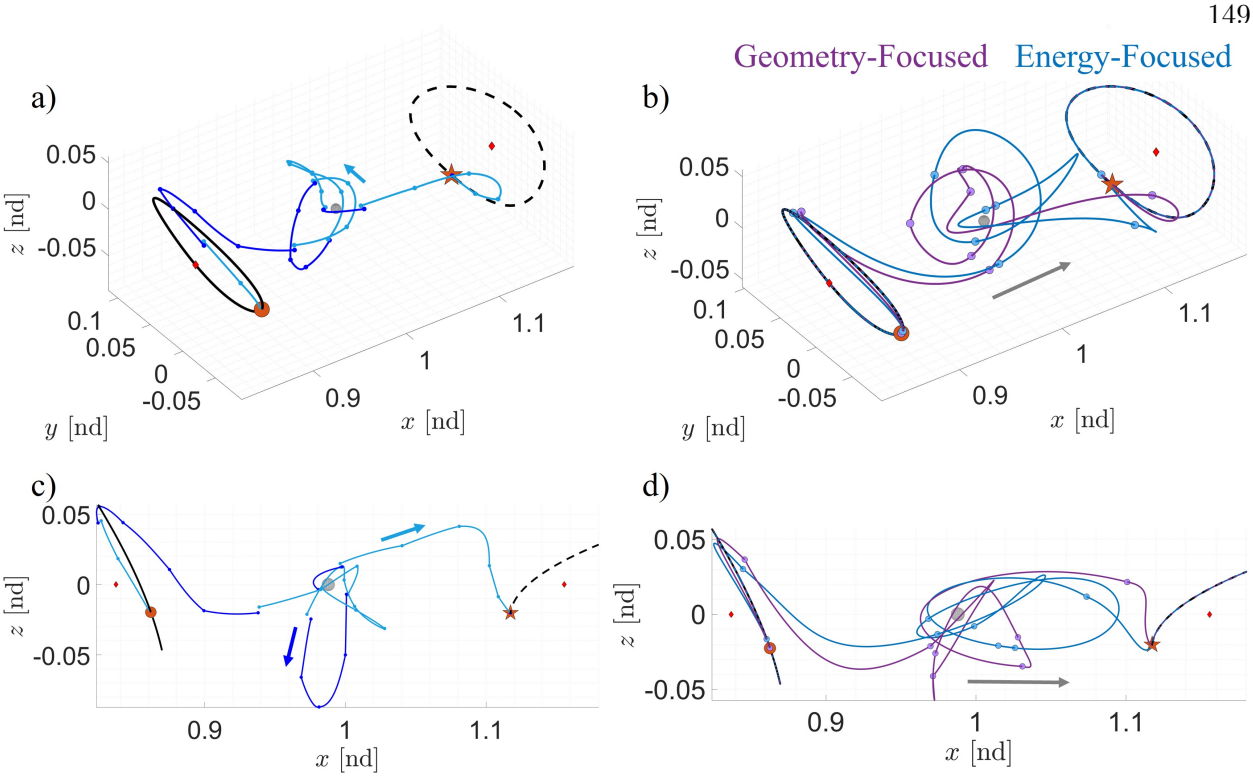

Figure 9.46: a) Spatial representation of initial guess 18. b) Spatial representations of the geometry-focused (purple) and energy-focused (blue) transfers. c) Initial guess 18 projected onto the xz-plane. d) Geometry- and energy-focused transfers projected onto the xz-plane. The geometry-focused spacecraft trajectory in purple possesses a higher Δv, equal to 891.93 m/s, than the energy-focused path whose Δv is equal to 188.31 m/s. The times of flight are 24.31 days and 25.31 days, respectively.

the Jacobi constant than the associated geometry-focused transfers. The energy-focused transfer recovered from initial guess 16, for instance, possesses an overall lower Δv than the corresponding geometry-focused transfer but spans a wider range of Jacobi constants. The geometry-focused transfer spans the range $C_J = [3.0580, 3.1515]$ while the energy-focused transfer spans the range $C_J = [3.0210, 3.1623]$. This is accompanied by a significant change in geometry, as seen in Figure 9.45 to reduce the total maneuver magnitude.

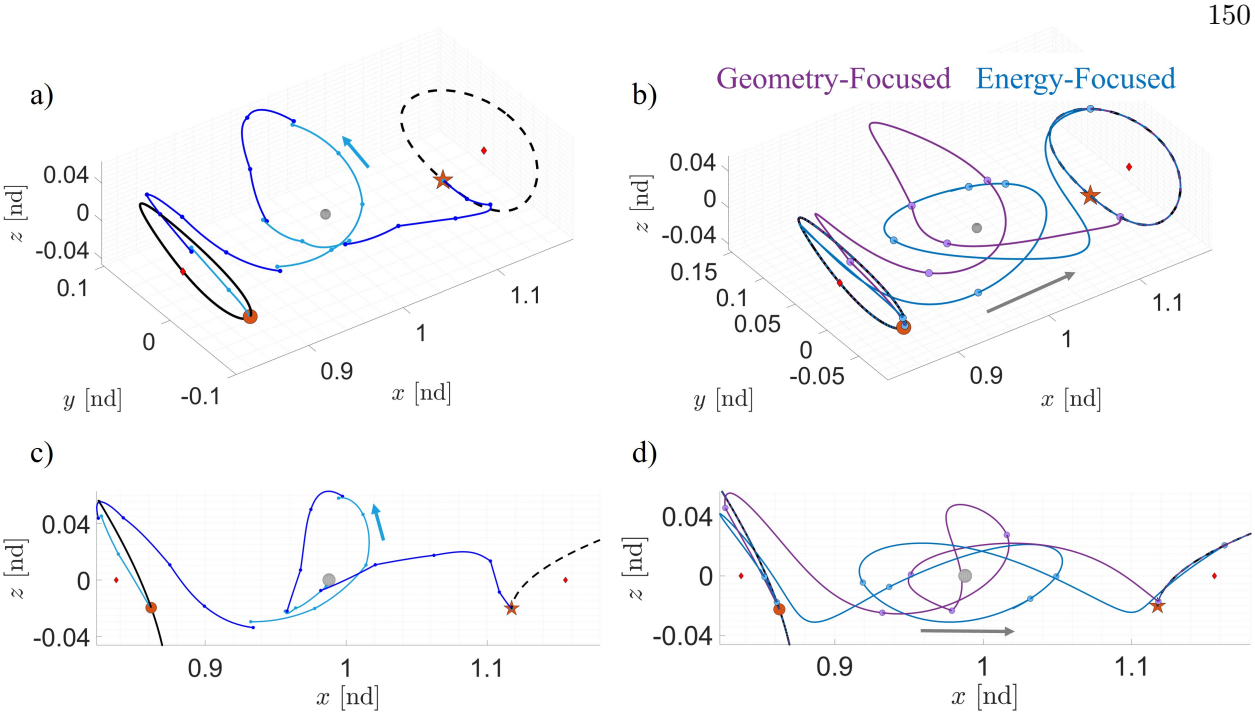

Figure 9.47: a) Spatial representation of initial guess 11. b) Spatial representations of the geometry-focused (purple) and energy-focused (blue) transfers. c) Initial guess 11 projected onto the xz-plane. d) Geometry- and energy-focused transfers projected onto the xz-plane. The geometry-focused spacecraft trajectory possesses a Δv equal to 585.16 m/s while the energy-focused has a Δv equal to 93.24 m/s. The flight times are 19.46 days and 27.11 days, respectively.

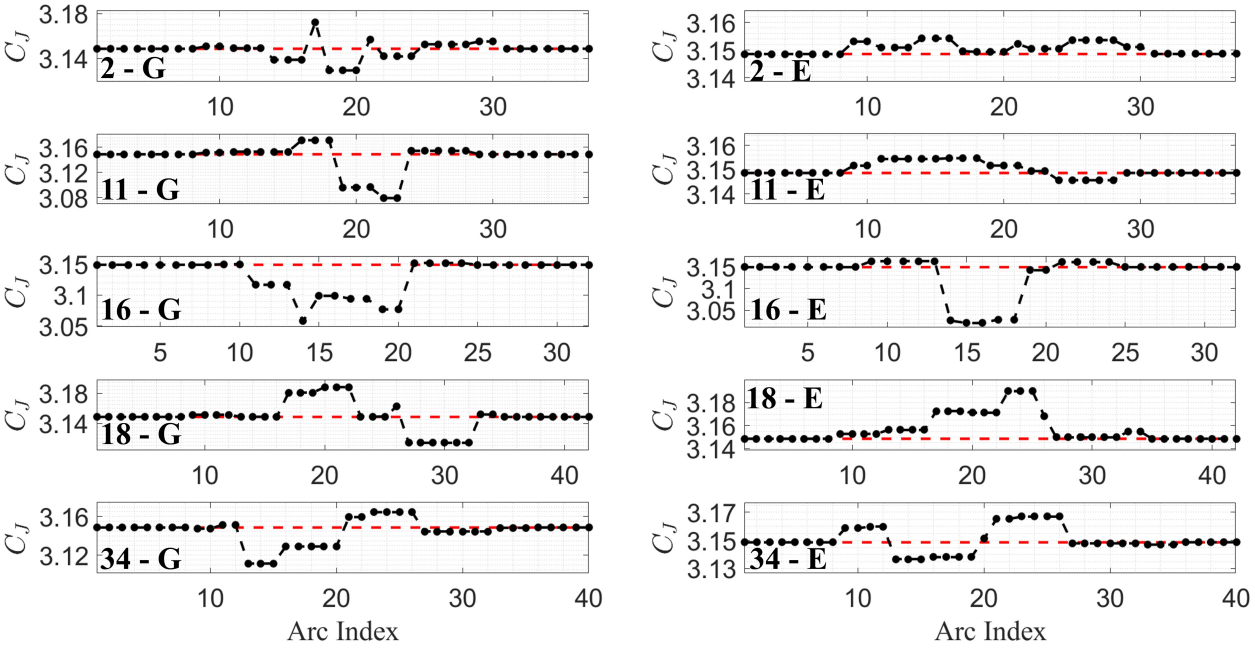

Figure 9.48: Evolution of the Jacobi constant along the geometry- (G) and energy-focused (E) solutions for transfers connecting L_1 and L_2 northern halo orbits at $C_J = 3.1486$, marked with a dashed red line, in the Earth-Moon CR3BP.

Chapter 10

Concluding Remarks

This dissertation presents a proof of concept for a new approach to collision-free spacecraft trajectory design in multi-body systems that leverages a combination of techniques from astrodynamics, data mining, and robotics. To summarize potential future spacecraft motions, dynamic obstacle corridors are constructed from local point clouds of initial states along periodic orbits. Then, a hierarchical tree-based planning approach is used to rapidly and autonomously construct arrays of geometrically-distinct initial guesses for spacecraft trajectories. To demonstrate the utility of the tree-based framework, correction and continuation schemes are employed to recover continuous trajectories. Then, corrected trajectories are analyzed for collisions with the dynamic obstacle corridors. This proof of concept is demonstrated for collision-free spacecraft trajectory design in the Earth-Moon CR3BP.

A clustering-based framework was developed to summarize the types of possible motions generated from a local point cloud of initial conditions by their geometric and temporal variations. In this proof of concept, initial states are propagated forward in time in the Earth-Moon CR3BP to generate the associated continuous trajectories. Each trajectory is then discretized into a set of states that are evenly distributed as a function of arclength. Finite-dimensional feature vectors are constructed to summarize each continuous trajectory using the velocity direction at each sample and elapsed time between samples. Then, HDBSCAN is used to cluster these feature vectors. Finally, these clusters are updated with additional information to increase their accuracy and ensure less prevalent geometries are captured, if they exist. The result is a set of clusters that

supply a digestible summary of the array of possible motion corridors emanating from a local point cloud of initial states. Through application to four scenarios in the Earth-Moon CR3BP, this data-driven categorization approach was demonstrated to automatically generate groups of trajectories with distinct spatiotemporal evolutions, when these differences exist. In each case, these clusters summarized over 1,000 trajectories via several key motion types. The automatic generation of these summaries may eventually reduce the burden on an analyst or support automation when identifying potential spacecraft motions for space situational awareness or collision-free path planning.

To address the challenges of rapid and automatic spacecraft trajectory design in chaotic dynamical environments, a new, tree-based framework was presented for constructing collision-free spacecraft trajectories in a multi-body gravitational system. First, initial guesses construction consists of two phases: 1) growing a forest of trees that captures dynamical flow throughout the environment and 2) using the forest to guide generation of an initial guess for a path connecting two boundary conditions. Leveraging a forest of trees enables rapid generation of initial guesses by not requiring solutions to boundary value problems and employing a hierarchical structure. Through application to two scenarios, this approach is demonstrated to automatically recover geometrically-distinct initial guesses between sets of boundary conditions along periodic orbits in the Earth-Moon CR3BP. The initial guesses exhibit distinct geometries and produce continuous transfers that retain their individual geometries. Then, a lazy, cell-based collision detection method is developed to address challenges with efficiently and robustly recovering collision-free trajectories when using discrete representations of continuous paths. Collision detection is performed for spacecraft trajectories when in the proximity of dynamic obstacle corridors that could, for example, represent possible motions of another spacecraft. If a transfer collides with one of these corridors, its departure time may be altered or a different initial guess may be leveraged to avoid the collision. This automated method for initial guess construction and collision identification may reduce reliance on a human operator during trajectory design, eventually support onboard redesign of spacecraft trajectories, and enable a safe and sustainable cislunar environment.

10.1 Recommendations for Future Work

The proof of concept for the collision-free spacecraft trajectory design method presented in this dissertation would benefit from further improvement and application to additional trajectory design scenarios. The list of future work that warrants investigation includes the following:

- Incorporating probabilities associated with trajectories sampled from a local initial state set would expand the applicability of this work to trajectory prediction problems. These probabilities could be incorporated during generation of the input data, categorization, or after the motion summary has been created. Furthermore, associating a likelihood of occurrence with each motion type in a motion summary may reduce the computational time required for collision checking to limit the analysis to the most probable motion types.

- The current data-driven categorization method relies on manual selection of the governing parameters for HDBSCAN. Creating an automated and adaptive method for parameter selection would reduce the reliance on a human analyst who must possess, or develop, an intuition for selecting these parameters.

- Examining the relationship, if one exists, between the motion corridors recovered using the data-driven categorization process and the structure of the local neighborhood of the reference trajectory could supply insight useful for motion prediction.

- The clustering-based framework for summarizing trajectory motion successfully categorized trajectory sets by their spatiotemporal variations. However, some limitations emerged, such as one large cluster including trajectories that continuously evolve across the set to exhibit multiple geometries or one motion type being split into multiple smaller clusters due to variations in the final state impacting sample placement. Alternative feature vectors, sampling schemes, and clustering algorithms may address these limitations in future work.

- Future application of the data-driven categorization process to a wider variety of dynamical models, celestial systems, and initial state sets may support characterizing the quality of

the generated summaries and assessing the generalizability of a clustering-based framework.

- The hierarchical tree-based planning approach leveraged a constant branch length during forest construction in this dissertation. An interesting future investigation may be to grow trees in the forest with variable branch lengths. Investigating this parameter further may lead to the discovery of a relationship between branch length, the connectivity of the forest, and tree exploration of the local phase space.

- Investigating alternative maneuver placement strategies or optimal maneuver placement is recommended. The number of maneuvers used, along with their location, dictate the success of the correction and continuation processes and affect the total Δv of each converged solution. Furthermore, an automatic maneuver placement strategy would further reduce the reliance an operator when correcting initial guesses.

- Continuation is leveraged in this work to recover continuous spacecraft trajectories that either prioritize resembling the geometry of the initial guess or reducing the maneuver requirements. A valuable avenue of future work is to implement an optimization scheme to recover optimal trajectories with respect to propellant usage while retaining the geometry of the initial guess. Furthermore, incorporating obstacles into the optimization process may aid in the recovery of collision-free spacecraft trajectories.

- Dynamic obstacle maps used for corridor avoidance are constructed using approximate cell decomposition. These maps are constructed to negate issues with discrete representations of the obstacle and spacecraft trajectories. Three valuable avenues of future work include 1) representing the obstacle corridor with an exact decomposition of the environment, 2) constructing maps with a variable cell size to reflect variation in the uncertainty as a function of time or location, or 3) parameterizing obstacle corridors in lieu of using dynamic obstacle maps.

- This research may also be useful for non-collision avoidance problems. For example, regions

or states that are not scientifically interesting for a particular mission may be modeled as "hazards" and avoided using the methodology presented in this dissertation. Therefore, a valuable future investigation is to apply this research to a variety of avoidance scenarios.

- An interesting avenue of future work is to apply the tree-based initial guess construction and collision checking methods to increasingly complex transfer design scenarios and higher-fidelity models. Although these methods are demonstrated in the CR3BP for this proof of concept, it is expected that they can be applied to other higher-fidelity dynamical models of three-body systems with minimal modification. This investigation would determine the generalizability of these approaches and provide insight into whether the methods are capable of designing trajectories in systems where the solution space is less understood. Furthermore, building upon these methods to support trajectory replanning, for example, would be a valuable extension of this work.

Bibliography

[1] GSL - GNU Scientific Library. `https://www.gnu.org/software/gsl/`, 2009. Last accessed 10 May 2025.

[2] Continuing Kepler's Quest: Assessing Air Force Space Command's Astrodynamics Standards. The National Academies Press, 2012.

[3] A. Abbadi and V. Prenosil. Safe Path Planning Using Cell Decomposition Approximation. In International Conference on Distance Learning, Simulation, and Communication, Brno, Czeck Republic, May 2015.

[4] A. Abbadi, R. Matousek, P. Osmera, and L. Knispel. Spatial Guidance to RRT Planner Using Cell-Decomposition Algorithm. In 20th International Conference on Soft Computing, Brno, Czech Republic, June 2014.

[5] A. Das-Stuart, K. C. Howell, and D. C. Folta. Rapid Trajectory Design in Complex Environments Enabled by Reinforcement Learning and Graph Search Strategies. Acta Astronautica, 171:172–195, 2020. DOI: https://doi.org/10.1016/j.actaastro.2019.04.037.

[6] A. F. Haapala and K. C. Howell. A Framework for Constructing Transfers Linking Periodic Libration Point Orbits in the Spatial Circular Restricted Three-Body Problem. International Journal of Bifurcation and Chaos, 26(5), 2016. DOI: https://doi.org/10.1142/S0218127416300135.

[7] A. González. Measurement of Areas on a Sphere Using Fibonacci and Latitude–Longitude Lattices. Mathematical Geosciences, 42:49–64, 2010. DOI: https://doi.org/10.1007/s11004-009-9257-x.

[8] A. Hadjighasem, D. Karrasch, H. Teramoto, and G. Haller. A Spectral Clustering Approach to Lagrangian Vortex Detection. Physics Review E, 93(6):63–83, 2016. DOI: https://doi.org/10.1103/PhysRevE.93.063107.

[9] A. Mandalika, S. Choudhury, O. Salzman, and S. Srinivasa. Generalized Lazy Search for Robot Motion Planning: Interleaving Search and Edge Evaluation via Event-based Toggles. In International Conference on Automated Planning and Scheduling (ICAPS), July 2019.

[10] A. Sintov and A. Shapiro. Time-Based RRT Algorithm for Rendezvous Planning of Two Dynamic Systems. In 2014 IEEE International Conference on Robotics and Automation, Hong Kong, China, 2014.

[11] C. C. Aggarwal and C. K. Reddy. Data Clustering: Algorithms and Applications. CRC Press, 2014.

[12] C. Danielson and J. Kloeppel. Rapid Construction of Safe Search-Trees for Spacecraft Attitude Planning. In 2023 American Control Conference (ACC), San Diego, CA, May 2023.

[13] C. Malzer and M. Baum. A Hybrid Approach To Hierarchical Density-based Cluster Selection. In 2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, Karlsruhe, Germany, 2020. DOI: https://doi.org/10.1109/MFI49285.2020.9235263.

[14] C. Schretter, L. Kobbelt, and P.-O. Dehaye. Golden Ratio Sequences for Low-Discrepancy Sampling. ACM Journal of Graphics Tools, 16(2):95–104, 2012. DOI: https://doi.org/10.1080/2165347X.2012.679555.

[15] D. A. Kottwitz. The Densest Packing of Equal Circles on a Sphere. Acta Crystallographica Section A, 47(3):158–165, May 1991. DOI: https://doi.org/10.1107/S0108767390011370.

[16] D. A. Vallado. Fundamentals of Astrodynamics and Applications. Microcosm Press and Springer, 4th edition, 2013.

[17] D. Berry. Conjunction Assessment Process For Lunar Orbiters. Technical report, Jet Propulsion Laboratory, 2022.

[18] D. C. Folta, N. Bosanac, D. Guzzetti, and K. C. Howell. An Earth-Moon System Trajectory Design Reference Catalog. Acta Astronautica, 110:341–353, 2015. DOI: https://doi.org/10.1016/j.actaastro.2014.07.037.

[19] D. D. Morrison, J. D. Riley, and J. F. Zancanaro. Multiple Shooting Method for Two-Point Boundary Value Problems. Communications of the ACM, 5(12):613–614, 1962. DOI: https://doi.org/10.1145/355580.369128.

[20] D. G. Loyola-R, M. Pedergnana, and S. G. García. Smart Sampling and Incremental Function Learning for Very Large High Dimensional Data. Neural Networks, 78:75–87, 2015. DOI: https://doi.org/10.1016/j.neunet.2015.09.001.

[21] D. Guzzetti, N. Bosanac, A. Haapala, and K. C. Howell. Rapid Trajectory Design in the Earth-Moon Ephemeris System via an Interactive Catalog of Periodic and Quasi-Periodic Oribts. Acta Astronautica, 126:439–455, 2016. DOI: https://doi.org/10.1016/j.actaastro.2016.06.029.

[22] D. J. Grebow. Generating Periodic Orbits in the Circular Restricted Three-Body Problem with Applications to Lunar South Pole Coverage. Master's Thesis, Purdue University, West Lafayette, Indiana, May 2006.

[23] D. Koh, R. L. Anderson, and I. Bermejo-Moreno. Cell-Mapping Orbit Search for Mission Design at Ocean Worlds Using Parallel Computing. The Journal of the Astronautical Sciences, 68:172–196, 2021. DOI: https://doi.org/10.1007/s40295-021-00251-6.

[24] D. Nieuwenhuisen and M. H. Overmars. Motion Planning for Camera Movements. In IEEE International Conference on Robotics and Automation, New Orleans, LA, April 2004.

[25] D. P. Hardin, T. Michaels, and E. B. Saff. A Comparison of Popular Point Configurations on S^2. Dolomites Research Notes on Approximation, 9:16–49, 2016. DOI: https://doi.org/10.48550/arXiv.1607.04590.

[26] E. Canalias and J. J. Masdemont. Homoclinic and Heteroclinic Transfer Trajectories between Lyapunov Orbits in the Sun-Earth and Earth-Moon Systems. Discrete and Continuous Dynamical Systems, 14(2):261–279, 2005. DOI: https://doi.org/10.3934/dcds.2006.14.261.

[27] E. L. Allgower and K. Georg. Numerical Continuation Methods: An Introduction. Springer Berlin, Heidelberg, 2011. DOI: https://doi.org/10.1007/978-3-642-61257-2.

[28] E. Schmerling and M. Pavone. Encyclopedia of Robotics, chapter Kinodynamic Planning. Springer, 2019.

[29] E. W. Dijkstra. A Note on Two Problems in Connexion with Graphs. Numerische Mathematik, 1:269–271, 1959. DOI: https://doi.org/10.1007/BF01386390.

[30] F. Aurenhammer. Voronoi Diagrams - A Survey of a Fundamental Geometric Data Structure. ACM Computing Surveys, 23(3):345–405, 1991. DOI: https://doi.org/10.1145/116873.116880.

[31] F. Lingelbach. Path Planning Using Probabilistic Cell Decomposition. In IEEE International Conference on Robotics and Automation, April 2004.

[32] G. E. Miceli and N. Bosanac. Designing Neptunian System Tours via a Motion Primitive Approach,. In AAS/AIAA Space Flight Mechanics Meeting, Kaua'i, HI, 2025.

[33] G. E. Miceli and N. Bosanac. Designing Neptunian System Tours via a Motion Primitive Approach. In AAS/AIAA Space Flight Mechanics Meeting, Kaua'i, HI, January 2025.

[34] G. E. Miceli, N. Bosanac, M. A. Mesarch, D. C. Folta, and R. L. Mesarch. Clustering Approach to Identifying Low Lunar Frozen Orbits in a High-Fidelity Model. In 2023 AAS/AIAA Astrodynamics Specialist Conference, Big Sky, MT, August 2023.

[35] G. S. Krishnaswami and H. Senapati. An Introduction to the Classical Three-Body Problem: From Periodic Solutions to Instabilities and Chaos. Resonance, 24:87–114, 2019. DOI: https://doi.org/10.1007/s12045-019-0760-1.

[36] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, and S. Thrun. Principles of Robot Motion: Theory, Algorithms, and Implementations. The MIT Press, Cambridge, MA, 2005.

[37] H. Liu, X. Zhang, J. Wen, R. Wang, and X. Chen. Goal-Biased Bidirectional RRT based on Curve-Smoothing. International Federation of Automatic Control, 52(24):255 – 260, 2019. DOI: https://doi.org/10.1016/j.ifacol.2019.12.417.

[38] H. Schaub and J. L. Junkins. Analytical Mechanics of Space Systems. AIAA Education Series, 2018.

[39] I. A. Şucan and L. E. Kavraki. A Sampling-Based Tree Planner for Systems With Complex Dynamics. IEEE Transactions on Robotics, 28(1):116–131, 2011. DOI: https://doi.org/10.1109/TRO.2011.2160466.

[40] J. A. Starek, E. Schmerling, G. D. Maher, B. W. Barbee, and M. Pavone. Fast, Safe, Propellant-Efficient Spacecraft Motion Planning Under Clohessy–Wiltshire–Hill Dynamics. Journal of Guidance, Control and Dynamics, 40(2):418–438, 2017. DOI: https://doi.org/10.2514/1.G001913.

[41] J.-C. Latombe. Robot Motion Planning. Springer, 1991.

[42] J. Chen, H. Li, and Q. Zhou. RRT-Forest: An Efficient Multi-UAV Cooperative Path Planning Method for Indoor Rescue. In 2023 8th International Conference on Advanced Robotics and Mechatronics, Sanya, China, 2023.

[43] J. E. Barrow-Green. Poincaré and the Three Body Problem. PhD Dissertation, The Open University, 1993.

[44] J. Guitton, J.-L. Farges, and R. Chatila. Cell-RRT: Decomposing the Environment for Better Plan. In The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, October 2009.

[45] J. Han, M. Kamber, and J. Pei. Data Mining Concepts and Techniques. Morgan Kaufmann Publishers, Waltham, MA, 3 edition, 2012.

[46] J. Janoš, V. Vonásek, and R. Pěnička. Multi-Goal Path Planning Using Multiple Random Trees. IEEE Robotics and Automation Letters, 6(2):4201–4208, 2021. DOI: https://doi.org/10.1109/LRA.2021.3068679.

[47] J. Pan, L. Zhang, and D. Manocha. Collision-Free and Curvature-Continuous Path Smoothing in Cluttered Environments, volume 17. MIT Press, 2012.

[48] J. Pan, L. Zhang, and D. Manocha. Collision-Free and Smooth Trajectory Computation in Cluttered Environments. The International Journal of Robotics Research, 31(10):1155–1175, 2012. DOI: https://doi.org/10.1177/0278364912453186.

[49] J. S. Parker, K. E. Davis, and G. H. Born. Chaining Periodic Three-Body Orbits in the Earth-Moon System. Acta Astronautica, 67(5-6):623–638, 2010. DOI: https://doi.org/10.1016/j.actaastro.2010.04.003.

[50] J. Y. Yen. An Algorithm for Finding Shortest Routes from All Source Nodes to a Given Destination in General Networks. Quarterly of Applied Mathematics, 27:526–530, 1970. DOI: https://doi.org/10.1090/qam/253822.

[51] K. C. Howell. Families of Orbits in the Vicinity of the Collinear Libration Points. The Journal of Astronautical Sciences, 49(1):107 – 125, 2001. DOI: https://doi.org/10.1007/BF03546339.

[52] K. E. Davis, R. L. Anderson, D. J. Scheeres, and G. H. Born. Optimal Transfers Between Unstable Periodic Orbits Using Invariant Manifolds. Celestial Mechanics and Dynamical Astronomy, 109:241–264, 2011. DOI: https://doi.org/10.1007/s10569-010-9327-x.

[53] K. L. Bruchko and N. Bosanac. Rapid Trajectory Design in Multi-Body System using Sampling-Based Kinodynamic Planning. In AAS/AIAA Astrodynamics Specialist Conference, Big Sky, MT, August 2023.

[54] K. Z. Haigh, W. Foslien, and V. Guralnik. Data Mining for Space Applications. In Space OPS 2004 Conference, Montreal, Quebec, May 2004.

[55] L. Kocis and W. J. Whiten. Computational Investigations of Low-Discrepancy Sequences. ACM Transactions on Mathematical Software, 23(2):266–294, 1997. DOI: https://doi.org/10.1145/264029.264064.

[56] L. McInnes, J. Healy, and S. Astels. The hdbscan Clustering Library. `https://hdbscan.readthedocs.io/en/latest/index.html`, 2016. Last accessed May 2025.

[57] L. McInnes, J. Healy, and S. Astels. hdbscan: Hierarchical Density Based Clustering. Journal of Open Source Software, 2(11), 2017. DOI: https://doi.org/10.21105/joss.00205.

[58] M. Bolden, I. Hussein, H. Borowski, R. See, and E. Griggs. Probabilistic Initial Orbit Determination and Object Tracking in Cislunar Space Using Optical Sensors. In Advanced Maui Optical and Space Surveillance Technologies Conference, Maui, HI, 2022.

[59] M. Ghosh, S. Thomas, and N. M. Amato. Fast Collision Detection for Motion Planning Using Shape Primitive Skeletons. In Algorithmic Foundations of Robotics XIII, pages 36–51. Springer International Publishing, 2020. DOI: https://doi.org/10.1007/978-3-030-44051-0_3.

[60] M. Gupta. Navigating Chaos: Resonant Orbits for Sustaining Cislunar Operations. PhD Dissertation, Purdue University, West Lafayette, Indiana, May 2024.

[61] M. Lara and R. P. Russell. On the Family "G" of the Restricted Three-Body Problem. Monografías de la Real Academia de Ciencias de Zaragoza, 30:51–66, 2006.

[62] M. Lo and S. Ross. SURFing the Solar System: Invariant Manifolds and the Dynamics of the Solar System. Technical Report IOM 312/97, Jet Propulsion Laboratory, 1997.

[63] M. Lo and S. Ross. The Lunar L1 Gateway: Portal to the Stars and Beyond. In AIAA Space 2001 Conference, Albuquerque, NM, August 2001.

[64] M. Overmars, I. Karamouzas, and R. Geraerts. Flexible Path Planning Using Corridor Maps. In Algorithms - ESA 2008 Lecture Notes in Computer Science, 2008.

[65] M. Pharr, W. Jakob, and G. Humphreys. Sampling and Reconstruction. Online, 3rd edition, 2004.

[66] M. Vaquero and J. Senent. Poincare: A Multi-Body, Multi-System Trajectory Design Tool. In 7th International Conference on Astrodynamics Tools and Techniques, Oberpfaffenhofen, Germany, November 2018.

[67] M. Zucker, J. Kuffner, and M. Branicky. Multipartite RRTs for Rapid Replanning in Dynamic Environments. In Proceedings 2007 IEEE International Conference on Robotics and Automation, Rome, Italy, April 2007.

[68] MathWorks. MATLAB. https://www.mathworks.com/products/matlab.html, 2025. Last accessed May 2025.

[69] N. Bosanac. Exploring the Influence of a Three-Body Interaction Added to the Gravitational Potential Function in the Circular Restricted Three-Body Problem: A Numerical Frequency Analysis. Master's Thesis, Purdue University, West Lafayette, Indiana, December 2012.

[70] N. Bosanac. Leveraging Natural Dynamical Structures to Explore Multi-Body Systems. PhD Dissertation, Purdue University, West Lafayette, Indiana, August 2016.

[71] N. Bosanac. Data Mining Approach to Poincaré Maps in Multi-Body Trajectory Design. Journal of Guidance, Control, and Dynamics, 43(6):1190–1200, 2020. DOI: https://doi.org/10.2514/1.G004857.

[72] N. Bosanac. Lecture Notes in Advanced Astrodynamics, August 2020.

[73] N. Bosanac. Data-Driven Summary of Natural Spacecraft Trajectories in the Earth-Moon System. In 2023 AAS/AIAA Astrodynamics Specialist Conference, Big Sky, MT, August 2023.

[74] N. Bosanac. Data-Driven Summary of Motion in an Ephemeris Model of Cislunar Space. In 2025 AAS/AIAA Space Flight Mechanics Meeting, Kaua'i, HI, January 2025.

[75] N. Bradley, Z. Olikara, S. Bhaskaran, and B. Young. Cislunar Navigation Accuracy Using Optical Observations of Natural and Artificial Targets. Journal of Spacecraft and Rockets, 57(4):777–792, 2020. DOI: https://doi.org/10.2514/1.A34694.

[76] N. J. A. Sloane, R. H. Hardin, W. D. Smith, and Others. Tables of Spherical Codes. Published electronically at NeilSloane.com/packings/.

[77] N. L. Parrish. A* Pathfinding for Continuous-Thrust Trajectory Optimization. In AAS 37th Annual Guidance and Control Conference, Breckenridge, CO, 2014.

[78] N. L. Parrish and D. J. Scheeres. Low-Thrust Trajectory Optimization with Simplified SQP Algorithm. In AAS/AIAA Astrodynamics Specialist Conference, Stevenson, WA, August 2017.

[79] N. Nakhjiri and B. Villac. Automated Stable Region Generation, Detection, and Representation for Applications to Mission Design. Journal of Celestial Mechanics and Dynamical Astronomy, 123:63–83, 2015. DOI: https://doi.org/10.1007/s10569-015-9629-0.

[80] NASA. Artemis. https://www.nasa.gov/humans-in-space/artemis/missions. Last accessed 4 January 2025.

[81] NASA. What is CAPSTONE? https://www.nasa.gov/smallspacecraft/capstone/. Last accessed 4 January 2025.

[82] Noborio, H., Naniwa, T., Arimoto, S. A Quadtree-Based Path-Planning Algorithm for a Mobile Robot. Journal of Robotic Systems, 7(4):555–574, 1990. DOI: https://doi.org/10.1002/rob.4620070404.

[83] P. E. Hart, N. J. Nilsson, and B. Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Transactions on Systems Science and Cybernetics, 4(2):100–107, 1968. DOI: https://doi.org/10.1109/TSSC.1968.300136.

[84] P. Prusinkiewicz and A. Lindenmayer. The Algorithmic Beauty of Plants. Springer-Verlag, New York, 1990.

[85] P. Sivasankar and T. Elgohary. Cislunar Orbital Uncertainty Propagation Through the Application of Radial Basis Function Approximation. In AIAA SciTech 2024 Forum, Orlando, FL, 2024. DOI: https://doi.org/10.2514/6.2024-0202.

[86] R. Gayle, K. R. Klingler, and P. G. Xavier. Lazy Reconfigurable Forest (LRF) - An Approach for Motion Planning with Multiple Tasks in Dynamic Environments. In 2007 IEE International Conference on Robotics and Automation, Roma, Italy, April 2007.

[87] R. J. G. B. Campello, D. Moulavi, A. Zimek, and J. Sander. Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection. ACM Transactions on Knowledge Discovery from Data, 10(1):1–51, 2015. DOI: https://doi.org/10.1145/2733381.

[88] R. J. G. B. Campello, D. Moulavi, and J. Sander. Density-Based Clustering Based on Hierarchical Density Estimates, page 160–172. Springer, 2013.

[89] R. Kindel, D. Hsu, J.-C. Latombe, and S. Rock. Kinodynamic Motion Planning Amidst Moving Obstacles. In Proceedings of the 2000 IEEE International Conference on Robotics and Automation, San Francisco, CA, April 2000.

[90] R. L. Anderson, S. Campagnola, and G. Lantoine. Broad Search for Unstable Resonant Orbits in the Planar Circular Restricted Three-Body Problem. Celestial Mechanics and Dynamical Astronomy, 124:177–199, 2016. DOI: https://doi.org/10.1007/s10569-015-9659-7.

[91] R. L. Spear. Planar Heteroclinic Connections in the Neptune-Triton Circular Restricted Three Body Problem. Master's Thesis, Univeresity of Colorado Boulder, Boulder, Colorado, August 2021.

[92] R. L. Spear and N. Bosanac. Clustering Spacecraft Trajectories Generated from a Local Initial State Set. Journal of Guidance, Control, and Dynamics, 2025. DOI: https://doi.org/10.2514/1.G008695.

[93] R. P. Grimaldi. Discrete and Combinatorial Mathematics: An Applied Introduction, chapter 11. Addison-Wesley Publishing Company, 3rd edition, 1994.

[94] R. Swinbank and R. J. Purser. Fibonacci Grids: A Novel Approach to Global Modelling. Quarterly Journal of the Royal Meteorological Society, 132(619):1769–1793, 2006. DOI: https://doi.org/10.1256/qj.05.227.

[95] R. W. Chan, E. A. Ramsay, C. H. Cunningham, and D. B. Plewes. Temporal Stability of Adaptive 3D Radial MRI Using Multidimensional

Golden Means. Magnetic Resonance in Medicine, 61:354–363, 2009. DOI: https://doi.org/10.1002/mrm.21837.

[96] S. A. Bortoff. Path Planning for UAVs. In Proceedings of the American Control Conference, Chicago, IL, June 2000.

[97] S. Aggarwal and N. Kumar. Path Planning Techniques for Unmanned Aerial Vehicles: A Review, Solutions, and Challenges. Journal of Computer Communications, 149:270–299, 2020. DOI: https://doi.org/10.1016/j.comcom.2019.10.014.

[98] S. Aghabozorgi, A. S. Shirkhorshidi, and T. Y. Wah. Time Series Clustering - A Decade Review. Information Systems, 53:16–38, 2015. DOI: https://doi.org/10.1016/j.is.2015.04.007.

[99] S. Bonasera and N. Bosanac. Applying Data Mining Techniques to Higher-Dimensional Poincaré maps in the Circular Restricted Three-Body Problem. Journal of Celestial Mechanics and Dynamical Astronomy, 133(51), 2021. DOI: https://doi.org/10.1007/s10569-021-10047-3.

[100] S. Fedeler, M. Holzinger, and W. Whitacre. Sensor Tasking in the Cislunar Regime Using Monte Carlo Tree Search. Advances in Space Research, 70(3):792–811, 2022. DOI: https://doi.org/10.1016/j.asr.2022.05.003.

[101] S. M. LaValle. Rapidly-Exploring Random Trees: A New Tool for Path Planning. The Annual Research Report, 1998.

[102] S. M. LaValle. Planning Algorithms. Cambridge University Press, 2006.

[103] S. M. LaValle and J. J. Kuffner. Randomized Kinodynamic Planning. In Proceedings of the 1999 IEEE International Conference on Robotics and Automation, Detroit, MI, May 1999.

[104] S. M. LaValle and J. J. Kuffner. Algorithmic and Computational Robotics: New Directions, chapter 23. CRC Press, 2001.

[105] S. Redon, Y. J. Kim, M. C. Lin, and D. Manocha. Fast Continuous Collision Detection for Articulated Models. Journal of Computing and Information Science in Engineering, 5:126–137, 2005. DOI: https://doi.org/10.1115/1.1884133.

[106] Samaniego, F., Sanchis, J., García-Nieto, S., Simarro, R. Recursive Rewarding Modified Adaptive Cell Decomposition (RR-MACD): A Dynamic Path Planning Algorithm for UAVs. Electron., 8(3), 2019. DOI: https://doi.org/10.3390/electronics8030306.

[107] T. Deka and J. McMahon. Efficient Astrodynamics-Informed Kinodynamic Motion Planning for Relative Spacecraft Motion. Advances in Space Research, 73(11):5426–5445, 2024. DOI: https://doi.org/10.1016/j.asr.2024.02.019.

[108] T. Lai and F. Ramos. Adaptively Exploits Local Structure with Generalised Multi-Trees Motion Planning. IEEE Robotics and Automation Letters, 7(2):1111 – 1117, 2022. DOI: https://doi.org/10.1109/LRA.2021.3132985.

[109] T. M. Vaquero Escribano. Poincaré Sections and Resonant Orbits in the Restricted Three-Body Problem. Master's Thesis, Purdue University, West Lafayette, Indiana, May 2010.

[110] T. M. Vaquero Escribano. Spacecraft Transfer Trajectory Design Exploiting Resonant Orbits in Multi-Body Environments. PhD Dissertation, Purdue University, West Lafayette, Indiana, August 2013.

[111] T. R. Smith and N. Bosanac. Constructing Motion Primitive Sets to Summarize Periodic Orbit Families and Hyperbolic Invariant Manifolds in a Multi-Body System. Journal of Celestial Mechanics and Dynamical Astronomy, 134(7), 2022. DOI: https://doi.org/10.1007/s10569-022-10063-x.

[112] T. R. Smith and N. Bosanac. Motion Primitive Approach to Spacecraft Trajectory Design in a Multi-Body Sytem. The Journal of the Astronautical Sciences, 70(34):1–47, 2023. DOI: https://doi.org/10.1007/s40295-023-00395-7.

[113] T. Shifrin. Differential Geometry: A First Course in Curves and Surfaces. University of Georgia, 2023.

[114] T. W. Liao. Clustering of Time Series Data - A Survey. Pattern Recognition, 38(11):1857 – 1874, 2005. DOI: https://doi.org/10.1016/j.patcog.2005.01.025.

[115] T.-Y. Li and Y.-C. Shie. An Incremental Learning Approach to Motion Planning with Roadmap Management. In In Proceedings of the IEEE International Conference on Robotics and Automation, Washington, DC, May 2002.

[116] V. G. Szebehely. Zero Velocity Curves and Orbits in the Restricted Problem of Three Bodies. The Astronomical Journal, 38(3):147 – 151, 1963. DOI: https://doi.org/10.1086/108931.

[117] V. Szebehely. Theory of Orbits: The Restricted Problem of Three Bodies. Academic Press, 1967. DOI: https://doi.org/10.1016/B978-0-12-395732-0.X5001-6.

[118] V. Vittaldev, R. P. Russell, and R. Linares. Spacecraft Uncertainty Propagation using Gaussian Mixture Models and Polynomial Chaos Expansions. Journal of Guidance, Control, and Dynamics, 39(9):2163–2169, 2016. DOI: https://doi.org/10.2514/1.G001571.

[119] W. Merkt, V. Ivan, and S. Vijayakumar. Continuous-Time Collision Avoidance for Trajectory Optimization in Dynamic Environments. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 7248–7255, Macau, China, 2019. DOI: https://doi.org/10.1109/IROS40897.2019.8967641.

[120] W. S. Koon, M. W. Lo, J. E. Marsden, and S. D. Ross. Dynamical Systems, the Three Body Problem and Space Mission Design. Marsden Books, 2006.

[121] W. S. Koon, M. W. Lo, J. E. Marsden, and S. D. Ross. Heteroclinic Connections Between Periodic Orbits and Resonance Transitions in Celestial Mechanics. Chaos: An Interdisciplinary Journal of Nonlinear Science, 10(2):427–469, 2020. DOI: https://doi.org/10.1063/1.166509.

[122] Y. Li, Z. Littlefield, and K. Bekris. Algorithmic Foundations of Robotics XI, chapter 16, pages 263–282. Springer, 2015.

[123] Z. Du, Y. Wen, C. Xiao, L. Huang, C. Zhou, and F. Zhang. Trajectory-Cell Based Method for the Unmanned Surface Vehicle Motion Planning. Applied Ocean Research, 86:207–221, 2019. DOI: https://doi.org/10.1016/j.apor.2019.02.005.

[124] Zhirui Sun, Jiankun Wang, and Max Q.-H. Meng. Multi-Tree Guided Efficient Robot Motion Planning. Procedia Computer Science, 209:31–39, 2022. DOI: https://doi.org/10.1016/j.procs.2022.10.096.