Incorporating Machine Learning into Trajectory Design Strategies in Multi-Body Systems

by

Stefano Bonasera

B.S., Aerospace Engineering, Politecnico di Milano, 2015

M.S., Space Engineering, Delft University of Technology, 2017

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

Ann and H.J. Smead Department of Aerospace Engineering Sciences

2022

Committee Members:

Prof. N. Bosanac, Chair

Prof. D.J. Scheeres

Prof. N. Ahmed

Dr. A. Farrés Basiana

Dr. S. Campagnola

Bonasera, Stefano (Ph.D., Aerospace Engineering Sciences)

Incorporating Machine Learning into

Trajectory Design Strategies in Multi-Body Systems

Thesis directed by Prof. N. Bosanac

Strategies for rapid trajectory design within multi-body systems typically focus on leveraging dynamical systems and traditional optimization theory for analysis and initial guess construction. These techniques often demand significant human effort in analyzing a large, complex, and high-dimensional solution space, as well as large computational resources. These requirements are cumbersome in time-critical scenarios where redesign is required and directly impacts the operational support and cost for missions or initial design tasks; they also limit the speed and capability of knowledge discovery tasks. However, recent advancements in machine learning have the potential to reduce the workload of a trajectory designer in analyzing and using information extracted from large and complex data. For example, techniques from unsupervised learning may aid the astrodynamicist in summarizing and understanding the solution space, while methods from reinforcement learning supports generating rapid decision-making models enabling a reduced involvement of the human operator. Thus, this investigation seeks to explore the incorporation of machine learning techniques into various steps of the trajectory design process in multi-body systems. Specifically, the presented work focuses on three fundamental analyses: 1) exploring the solution space with higher-dimensional Poincaré maps and unsupervised learning; 2) constructing natural spatial transfers between quasi-periodic trajectories with the assistance of manifold learning during initial guess construction; 3) designing impulsive maneuvers for station-keeping and orbit transfer via reinforcement learning strategies. The presented results highlight the beneficial impact of techniques from machine learning for trajectory design, offering a foundation for continued development.

A Ninnini

Acknowledgements

I would like to thank my advisor, Prof. Bosanac, for her extensive support and guidance throughout these four years. Thank you also for introducing me to many interesting research topics, and allowing my mind to experiment on a variety of tasks. I would also like to extend my gratitude to committee members, for their invaluable feedback and support.

Boulder allowed me to meet a lot of people, from all over the world. I have been lucky to have sit in numerous meetings with extremely talented people as Ian, Jack, Thomas, Kristen, Renee and Giuliana. Thank you for your time, your friendship, and all the technical feedback. A special thanks goes to Ian, Jack, and Thomas, for their demonstrated patience at the numerous hours spent in front of my beautiful blue grid plots, and for not laughing at my weird accent during our group meeting presentations. Thank you also to the people I met outside our lab, which are definitely too numerous to name just in a few pages. Thank you all for sharing with me a lot of adventures, and many fun moments between Boulder and Denver. I especially thank Marco, who joined Boulder with me at a similar time: I feel privileged to have known you, and I am extremely thankful for all the emotional support you gave me throughout these years. Also, I am extremely grateful for the emotional and technical support that you, Mattia, Paolo, and Cristian, have given me over the last four years. I am also grateful for NASA grants 80NSSC19K0222 and 80NSSC18K1536 and the Bahls Endowed Funds for supporting portions of this work.

Per ultimo, e più importante, vorrei ringraziare i miei genitori, per aver sempre supportato le mie decisioni in questi anni, nonostante mi abbiano portato ben lontano dalla Sicilia, e mia sorella per il tuo sostegno da quando sono nato. Grazie mille!

Contents

Chapter

1 Background and Motivation **1**

1.1 Understanding the Solution Space via Poincaré Maps 2

1.2 Constructing Spatial Transfers between Quasi-Periodic Trajectories 4

1.3 Designing Station-Keeping and Orbit Transfer Maneuvers 6

1.4 Organization of the Manuscript 8

1.5 Contributions 10

2 Dynamical Models **12**

2.1 Circular Restricted Three-Body Problem 12

2.1.1 Jacobi Constant 16

2.1.2 Equilibrium Points 17

2.1.3 Zero Velocity Surfaces 18

2.2 Elliptic Restricted Three-Body Problem 19

2.3 Point Mass Ephemeris Model 21

2.4 Coordinate Frame Transformations 23

2.4.1 Transforming from Rotating to Inertial Frame in the CR3BP 23

2.4.2 Transforming from Rotating to ICRF Inertial Frame in Ephemeris Model . . 24

3 Particular Solutions **27**

3.1 Periodic Orbits 28

3.2 Differential Correction 30
3.3 Continuation Methods 33
3.3.1 Natural-Parameter Continuation 34
3.3.2 Pseudo-Arclength Continuation 34
3.4 Periodic Orbits Near Orbital Resonances 36
3.5 Quasi-Periodic Trajectories 39
3.6 Hyperbolic Invariant Manifolds 44
3.7 Poincaré Map 46
3.8 Recovering Trajectories in Higher Fidelity Models 50

4 Machine Learning **54**
4.1 Unsupervised Learning 55
4.1.1 Clustering 55
4.1.2 Dimensionality Reduction 63
4.2 Reinforcement Learning 66
4.2.1 Feedforward Neural Networks 70
4.2.2 Value-Based, Policy-Gradient and Actor-Critic Methods 73
4.2.3 Trust Region and Proximal Policy Optimization 76
4.2.4 Transfer Learning 79

5 Unsupervised Learning for Higher-Dimensional Poincaré Maps **81**
5.1 Distributed Clustering for Spatial Poincaré Maps 82
5.1.1 Method Overview 82
5.1.2 Clustering Spatial Maps in the Sun-Earth CR3BP at $C_J = 3.00088$ 86
5.2 Clustering Maps across Different Dynamical Models 91
5.2.1 Method Overview 92
5.2.2 Cluster Persistence Across Dynamical Models 100
5.3 Examining Fundamental Transport Mechanisms 112

5.3.1 Projecting Hyperbolic Manifold Arcs onto the Global Clustering Result . . . 113

5.3.2 Analyzing Clusters Governed by Stable Manifolds near L_1 and L_2 116

5.4 Summary of Contributions 120

5.4.1 Scientific Contributions of the Approach 121

5.4.2 The Value of Unsupervised Learning for Higher-Dimensional Poincaré Maps . 122

5.4.3 Avenues for Future Work 122

6 Constructing Natural Transfers between Quasi-Periodic Trajectories **124**

6.1 Method Overview 125

6.1.1 Phase 1: Initial Guess Construction 126

6.1.2 Phase 2: Transfer Correction and Numerical Continuation 129

6.2 Recovering Natural Transfers between Prescribed Invariant 2-Tori near Resonances . 134

6.3 Generating Transfers Between Families of Invariant Tori near Resonances 142

6.4 Summary of Contributions 146

6.4.1 Scientific Contributions of the Presented Results 148

6.4.2 Benefits of Manifold Learning for Higher-Dimensional Poincaré Maps 149

6.4.3 Avenues for Future Work 149

7 Autonomous Maneuver Design with Reinforcement Learning **151**

7.1 Station-Keeping Maneuver Design in the Sun-Earth System 152

7.1.1 Scenario Overview 152

7.1.2 Reinforcement Learning Problem 154

7.1.3 Translating Station-Keeping Maneuver Design into an Optimization Problem 165

7.1.4 Station-Keeping Maneuver Design with Reinforcement Learning in an Ephemeris Model 167

7.1.5 Comparing Optimization and Reinforcement Learning for Station-Keeping Maneuver Design 172

7.2 Transfer between Prescribed Orbits in the Earth-Moon System 179

7.2.1 Scenario Overview 180

7.2.2 Translating Orbit Transfer into a Reinforcement Learning Problem 182

7.2.3 Generating Transfers between Prescribed Orbits 187

7.3 Transfer Design in a Family of Orbits in the Earth-Moon System 201

7.3.1 Scenario Overview 201

7.3.2 Translating the Scenario into a Reinforcement Learning Problem 203

7.3.3 Generating Transfers between Families of Orbits 206

7.4 Summary of Contributions 219

7.4.1 Scientific Contributions of the Presented Applications 221

7.4.2 Value of Reinforcement Learning for Autonomous Maneuvering in Multi-Body Systems 222

7.4.3 Ongoing Challenges in Applying Reinforcement Learning to Autonomous Maneuver Design 224

8 Concluding Remarks **226**

8.1 Summary of Results 226

8.2 Recommendations for Future Work 230

Bibliography **234**

Tables

Table

2.1 Characteristic quantities for the Sun-Earth and the Earth-Moon systems 14

5.1 HDBSCAN parameters and results for each partition of the dataset used to generate a prograde periapsis map. 87

5.2 Dynamical characteristics of Lyapunov and vertical orbits near L_1 and L_2 in the Sun-Earth CR3BP at $C_J = 3.00088$ 114

7.1 Selected hyperparameters for the training and the neural network (NN) structure. . 164

7.2 Characteristics of sequences of 8 maneuvers and generated trajectories constructed using policies trained with and without transfer learning (TL) 171

7.3 Validation of a policy for the baseline orbit transfer scenario between prescribed orbits. 191

7.4 Validation of a policy for the baseline orbit transfer scenario between members of a family of orbits. 209

Figures

Figure

2.1 Geometry of the circular restricted three-body problem (celestial bodies not to scale). 15

2.2 Location of the five Lagrange points and the primaries for the Earth-Moon CR3BP. 18

2.3 Zero-velocity curves for a set of Jacobi constants in the Earth-Moon system. 20

2.4 Zero-velocity surfaces for a set of Jacobi constants in the Earth-Moon system. 20

2.5 Geometry of the N_e-body problem (celestial bodies not to scale). 22

3.1 Example of Lyapunov and northern halo orbits in the Earth-Moon CR3BP. 35

3.2 Example of orbits near resonances in the Earth-Moon CR3BP. 39

3.3 Examples of quasi-periodic trajectories in the Sun-Earth CR3BP. 42

3.4 Examples of quasi-periodic trajectories near resonances in the Sun-Earth CR3BP. . . 42

3.5 Example trajectories from stable and unstable manifolds. 46

3.6 Conceptual visualization of a Poincaré mapping. 47

3.7 Example of a periapsis map in the Sun-Earth system. 50

4.1 Example of core distance used by HDBSCAN. 58

4.2 Visualization of the agent-environment interaction in reinforcement learning. 67

4.3 Conceptual visualization of a feedforward fully-connected neural network. 71

5.1 Poincaré map reflecting prograde periapses in the Sun-Earth CR3BP at a Jacobi constant $C_J = 3.00088$ and $\dot{z} = 0$ using three maps. 89

5.2 Poincaré map reflecting a subset of clusters of prograde periapses in the Sun-Earth CR3BP at a Jacobi constant $C_J = 3.00088$ and $\dot{z} = 0$ using multiple maps. 90

5.3 Representatives of the clusters labeled on the map in Fig. 5.2, plotted in the Sun-Earth rotating frame. 90

5.4 Poincaré map reflecting prograde periapses in the Sun-Earth system at a Jacobi constant $C_J = 3.00088$ and $\dot{z} = 0$ using the full dataset. 92

5.5 Clustered perigee maps near the Earth, constructed for the same set of initial conditions in: (a) the Sun-Earth CR3BP at $C_J = 3.00088$ and (b) the Sun-Earth ER3BP with $f_0 = \pi/2$. 96

5.6 Example of noise reassignment for a map in the ER3BP. 98

5.7 Clustered perigee maps for various values of the initial true anomaly in the Sun-Earth ER3BP. Clusters of similar solutions are correlated across maps. 103

5.8 Selected cluster representatives for maps generated in the ER3BP. 104

5.9 Clustered perigee maps for various values of initial epoch in the Sun-Earth point mass ephemeris model. Clusters of similar solutions are correlated across maps. . . 107

5.10 Selected cluster representatives for maps generated in a point mass ephemeris model. 108

5.11 Clustered perigee maps for the same initial conditions in the: CR3BP, ER3BP, and the Sun-Earth point mass ephemeris model. 110

5.12 UMAP projection of the global reduced dataset presented in Fig. 5.11. 111

5.13 Periodic orbits near L_1 and L_2 in the Sun-Earth CR3BP at $C_J = 3.00088$. 114

5.14 Examples of quasi-periodic trajectories in the Sun-Earth CR3BP. 115

5.15 Visual example of the association between stable manifold arcs and trajectories naturally escaping the Earth region before three revolutions 118

5.16 Visual example of the association between stable manifold arcs and trajectories experiencing temporary captured motion near the Earth before three revolutions . . . 119

6.1 Example orbits and invariant tori near the 3:2 and 1:2 resonances in the Earth-Moon CR3BP at $C_J = 2.73$. 135

6.2 Poincaré map of the intersections of the hyperbolic invariant manifolds with the $y = 0$ plane emanated from the invariant tori near the 1:2 and 3:2 resonances. 137

6.3 Projection of the crossing dataset onto a two-dimensional Euclidean space calculated by UMAP. 139

6.4 Selected natural transfers between invariant tori near the 1:2 and 3:2 resonances. . . 140

6.5 Crossings of the transfers overlaid on the Poincaré map of the hyperbolic invariant manifolds from the invariant tori near the 1:2 and 3:2 resonances. 141

6.6 UMAP projections of the crossings from the hyperbolic manifolds of two invariant tori near the 1:3 and the 3:1 resonances and constructed natural transfer. 142

6.7 UMAP projections of the crossings from the hyperbolic manifolds of two invariant tori near the 1:5 and the 2:3 resonances and constructed natural transfer. 142

6.8 Summary of the continuation scheme for natural transfers continued from Fig. 6.4(a). 144

6.9 Summary of the continuation scheme for natural transfers continued from Fig. 6.4(d). 147

7.1 Reference trajectories near the Sun-Earth L_2: southern halo orbit in the CR3BP and nearby quasi-halo bounded trajectory in a Sun-Earth-Moon ephemeris model. 153

7.2 Conceptual representation of the regions separating different reward formulations. . 156

7.3 Flowchart of the station-keeping point mass ephemeris RL environment. 159

7.4 Cost function and number of the Bayesian optimization run associated with the (a) actor and (b) critic width hyperparameter exploration. 164

7.5 Conceptual representation of the optimization scenario. 165

7.6 Comparison of training value between a policy trained in the point mass ephemeris model and a policy initially trained in the Sun-Earth CR3BP and later transferred to the point mass ephemeris model. 170

7.7 Station-keeping maneuvers in the point mass ephemeris model with regular momentum unloads. 173
7.8 Time history of the magnitude of the position (top) and velocity (bottom) of the controlled trajectory in Fig. 7.7, relative to the reference quasi-halo trajectory. . . . 173
7.9 Comparison between the RL policy and the optimization on a single-point solution. . 175
7.10 First set of histograms of a batch of 500 trajectories generated with the RL policy and the constrained optimization. 178
7.11 Second set of histograms of a batch of 500 trajectories generated with the RL policy and the constrained optimization. 180
7.12 Third set of histograms of a batch of 500 trajectories generated with the RL policy and the constrained optimization scheme. 181
7.13 Departing (blue) and arrival (red) orbit in the Earth-Moon system for the orbit transfer scenario in the (a) spatial- and (b) xz-perspective in configuration space. . 182
7.14 Conceptual representation of the orbit transfer scenario between prescribed orbits. . 184
7.15 Flowchart of the training process for the orbit transfer RL scenario. 187
7.16 Trend of values for policies trained with and without transfer learning. 190
7.17 Comparison between sampled trajectories in the Earth-Moon system, generated via policies trained with and without transfer learning. 192
7.18 Histograms of final position displacement and final velocity displacement. 193
7.19 Histograms of total maneuver magnitude and total reward. 194
7.20 Example of suboptimal trajectory generated in the Earth-Moon system and associated rewards and maneuver magnitudes. 196
7.21 Example scenario for a transfer between two southern halo orbits near L_2 in the Earth-Moon CR3BP 197
7.22 Average value per update for a transfer from a southern halo orbits near L_2 to a southern halo orbit near L_1 in the Earth-Moon CR3BP. 199

7.23 Sample transfer from a southern halo orbits near L_2 to a southern halo orbit near L_1 in the Earth-Moon CR3BP, generated by the converged policy. 199

7.24 Training process and sampled trajectory for an example scenario of a transfer between two spatial orbits near resonances in the Earth-Moon CR3BP. 200

7.25 Example of combinations of departing (blue) and arrival (red) orbits from the family of southern halo orbits near L_2 in the Earth-Moon CR3BP. 202

7.26 Example of internal region $\mathscr{D}$ for the family of L_2 southern halo orbits in the Earth-Moon system, depicted as a semi-transparent orange surface. 206

7.27 Flowchart for the orbit transfer RL environment between members of a family of orbits. 207

7.28 Average value per update for a transfer between members of a family of southern halo orbits near L_2 in the Earth-Moon CR3BP. 209

7.29 Histograms of final position displacement and final velocity displacement from the arrival orbits 211

7.30 Representation of the total maneuver magnitude for the batch of 5000 transfers generated in the Earth-Moon CR3BP. 212

7.31 Representation of (a) final position deviation and (b) final velocity deviation from the arrival orbits for the batch of 5000 transfers generated in the Earth-Moon CR3BP. 213

7.32 Examples of transfers between distinct combination of southern halo orbits near L_2 in the Earth-Moon system. 214

7.33 Average value per update for the orbit transfer scenario between members of a family of southern halo orbits near L_1 in the Earth-Moon system. 216

7.34 Examples of transfers between distinct combination of southern halo orbits near L_1 in the Earth-Moon system. 217

7.35 Average value per update for the orbit transfer scenario connecting members of the family of southern halo orbits near L_2 to members of the family of southern halo orbits near L_1 in the Earth-Moon system. 218

7.36 Examples of transfers connecting distinct combinations of southern halo orbits near L_2 and southern halo orbits near L_1 in the Earth-Moon system. 219

Chapter 1

Background and Motivation

Future plans of a large variety of space agencies and private companies involve increasing human presence well-beyond the Earth neighborhood. In this region, the gravitational influence of multiple planetary bodies governs the path followed by a spacecraft. Therefore, trajectories design in multi-body systems becomes a fundamental asset for enabling a plethora of future missions. The analysis and construction of end-to-end trajectories in multi-body systems requires a significant human involvement and computational resources. For example, investigating a solution space where a variety of arcs of interest can exist presents a great visualization challenge, demanding extensive human expertise and formulation of problem-dependent criteria to select an interesting arc. Likewise, designing sequences of maneuvers enabling an end-to-end trajectory often requires multiple iterations and large computational resources, sometimes not admissible during the final fast-paced phases of trajectory design. Techniques from machine learning can reduce the burden of a trajectory analyst, reducing the human intervention, and generating computationally lightweight solutions. For example, techniques from unsupervised learning can infer patterns in the solution space, and improve the visualization of higher-dimensional data, while methods from reinforcement learning can generate models between data, leading to a decreased involvement of a human operator. For these reasons, this investigation approaches different challenges of trajectory design in multi-body systems, trying to reduce computational efforts and human involvement during the different phases of end-to-end trajectory construction process.

Initially, the available design space is analyzed in a low-fidelity dynamical model to extract arcs that respect mission requirements. These arcs might be leveraged to generate transfers targeting specific solutions of interest, for example trajectories resembling periodic or quasi-periodic solutions. Eventually, these arcs are refined in higher-fidelity dynamical models, and maneuvers are included to generate end-to-end trajectories. Machine learning techniques are used in this dissertation to aid a trajectory analyst alongside the design process: unsupervised learning may reduce human burden in visualizing higher-dimensional solution spaces, manifold learning may aid an analyst to target existing transfers, while reinforcement learning may reduce the intervention of human operators for maneuver design. This section introduces previous contributions on these fundamental phases of trajectory design, together with an overview of techniques from machine learning that are relevant to this dissertation.

1.1 Understanding the Solution Space via Poincaré Maps

The early phases of the trajectory design process often involve the investigation of the underlying, complex solution space. Tools from dynamical systems theory, such as Poincaré maps, supply useful instruments for the exploration of the solution space, in both low- and higher-fidelity dynamical models [1–4]. A Poincaré map is generated by collecting the intersections a set of trajectories with a common surface of section. Examples of surface of sections include: hyperplanes; apses; hyperspheres about a central body [2, 5, 6]. A well-constructed Poincaré map reduces the dimensionality of the problem, therefore simplifying the visualization of the trajectory arcs. However, Poincaré maps may be difficult to visualize due to higher-dimensional crossings, potentially leading to data obscuration or loss of information. When the recorded intersections on a map are high-dimensional, such as when spatial trajectories are analyzed, a 2D representation does not completely describe a crossing. This problem has been previously analyzed through a variety of approaches. For example, Haapala leverages glyphs to incorporate more dimensions in a planar representation of higher-dimensional Poincaré map, and uses analytical separation criteria to discern the different trajectories [3]. Gomez et al. reduce the dimension of the data by including additional constraints [7]. These approaches address the problem of uniqueness in the 2D representation.

However, if the dataset is more complex, denser or retrieved from a complex dynamical model, rapid and informed analysis of higher-dimensional Poincaré maps can still represent a challenging task, especially when fast redesign analysis is needed. Furthermore, a trajectory designer is often interested in examining the available set of geometries in the solution space.

A recent alternative approach for visualization and analysis of a higher-dimensional Poincaré map is inspired by the field of Big Data, which uses unsupervised learning. This approach is designed to: autonomously differentiate trajectory arcs based on specified characteristics; avoid data obscuration by summarizing the dataset with a representative set of solutions; and avoid the incorporation of any constraints which limit the design space. Unsupervised learning techniques, such as clustering, have been applied to problems in astrodynamics in multi-body systems: Nakhijiri and Villac leverage clustering to extract dynamical features and stability constraints from phase space maps, while Villac, Anderson and Pini use clustering for autonomous grouping of ballistic orbits around small bodies [8, 9]. Smith and Bosanac also demonstrate the value of unsupervised learning in astrodynamics by leveraging different clustering techniques to extract motion primitives that summarize features of periodic orbits and arcs along the hyperbolic invariant manifolds in the Earth-Moon Circular Restricted Three-Body Problem (CR3BP) [10–12].

Motivated by these works, Bosanac introduces the use of clustering for the autonomous differentiation of a wide variety of trajectories captured by a general Poincaré map to construct a summary of the solution space, and aid in the visualization of multi-body systems [13]. In her work, trajectories in the planar Sun-Earth CR3BP are autonomously grouped based on geometric similarity, leveraging the Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) algorithm [14, 15]. Upon differentiation of the trajectory arcs in clusters, each group is associated with a unique best representative solution which summarizes the geometrical characteristics of the entire group [16]. HDBSCAN offers a robust and computationally efficient solution for autonomous trajectory differentiation, capable of dividing a dataset in groups with: no a priori knowledge of the number of clusters; clusters of any shape and density; and identification of noise. Bonasera and Bosanac build upon this work by extending this framework to higher-dimensional maps generated in complex dynamical models by using distributed clustering for the autonomous differentiation of large datasets of arcs [17, 18]. Through this approach, the trajectory designer can obtain a summary of the solution space without using analytical separation criteria, without applying constraints that reduce the design space, and overcoming the effect of data obscuration.

1.2 Constructing Spatial Transfers between Quasi-Periodic Trajectories

The dynamical mechanisms associated with the fundamental structures are often of interest to either understand natural transport or constructing trajectories with reduced propellant consumption. Examples such as ARTEMIS leverage the natural transport mechanisms granted by the hyperbolic invariant manifolds of various orbits to steer the spacecraft towards their final destination trajectories [19]. Some missions leverage trajectories in resonant motion with a secondary body. Examples include the Interstellar Boundary Explorer (IBEX) and the Transiting Exoplanet Survey Satellite (TESS), in a 3:1 and 2:1 resonant motion with the Moon, respectively, in the Earth-Moon system [20, 21]. Missions such as Cassini also use trajectories that leverage the connections between resonant motions and maneuvers to maximize scientific return [22]. The phenomenon of natural transport mechanisms between orbits near resonances, enabled by the hyperbolic invariant manifolds of these structures, is also well recognized in celestial mechanics: Jovian comets like Oterma, or the Kuiper Belt Objects (KBOs) have been observed or predicted to naturally transit between orbits near different resonances in their systems [5,23]. For these reasons, investigating the natural motion of a small body to ballistically transition between different resonances represents an interesting area of exploration in both the trajectory design and celestial mechanics communities.

The study of the natural transitions in multi-body systems has been primarily focused on finding connections between periodic orbits in low-fidelity dynamical models, leveraging tools from dynamical systems theory. For example, Koon et al. analyze the hyperbolic invariant manifolds and the associated intersections with a well-constructed Poincaré map to explain the natural transition of comet Oterma between two orbits near resonances in the planar Sun-Jupiter system [5]. Later, Vaquero [24] and Haapala and Howell [25] extend this investigation by studying natural transitions between spatial resonances. The solution space may be further expanded by considering natural bounded motion near these orbits, i.e. quasi-periodic trajectories.

Quasi-periodic trajectories foliate invariant tori near periodic orbits which exhibit a non-empty central manifold in low-fidelity dynamical models. Among the variety of approaches to compute invariant tori near periodic orbits [26], methods developed by Jorba [27], Gomez et al. [28], and Olikara and Scheeres [29, 30] supply computationally-feasible approaches to recovering families of tori and their associated hyperbolic invariant manifolds in low-fidelity dynamical models of multi-body systems. Olikara applies this algorithm to demonstrate the existence of natural transport between two tori in a prescribed region of the design space in the Earth-Moon CR3BP, demonstrating two different methods which: 1) fix a-priori the departure torus and recover the arrival structure after convergence in a specified region of the design space or 2) use continuation from a nearby planar connection between two periodic orbits. More recently, Henry and Scheeres leverage whiskers to find connections between prescribed tori in position space, later adjusting the trajectory via an impulsive maneuver to ensure full-state continuity [31]. These approaches successfully generate connections between invariant tori. However, a trajectory designer may be interested in finding natural connections between two predefined tori, or obtaining an overview of the array of transfers connecting families of tori. Previous contributions have focused on natural connections between two unstable orbits by investigating intersections between the invariant manifolds on a common surface of section [24, 25, 28]. However the higher-dimensionality of the design space of unstable tori and the associated larger dataset can hinder the process of identifying feasible connections.

Methods from machine learning may support the analyst in identifying natural transport mechanisms by reducing the complexity of the Poincaré map visualization process. In particular, techniques from manifold learning can help the trajectory designer by projecting a dataset, such as the crossings on a Poincaré map, onto a lower-dimensional space. Among the variety of existing manifold learning approaches, Uniform Manifold Approximation and Projection (UMAP) is a robust and computationally-efficient state-of-the-art solution for projecting highly nonlinear datasets onto low-dimensional spaces [32]. UMAP has been successfully used in various scientific domains, such as visualizing complex proteins in single cell biology [33], studying genetic structures in cohorts [34] and categorizing the origin of solar wind [35]. The lower-dimensional representation constructed by UMAP preserves the relative distances between points, while minimizing the topological distance between the original dataset and its projection. Leveraging manifold learning to reduce the visualization burden may help the trajectory designer to find natural connections between two spatial tori and understand the solution space. Moreover, the discovery of these transfers can support the analysis of small-bodies transitioning between resonances in the solar system.

1.3 Designing Station-Keeping and Orbit Transfer Maneuvers

Sequences of maneuvers are often implemented to target specific solutions of the design space or mitigate the effect of unmodeled dynamical perturbations that may divert the spacecraft from the designed path. Furthermore, the dynamical characteristics of a mission orbit may naturally steer the spacecraft away from its designed reference path when perturbations occur. Therefore, trajectory designers often leverage maneuvers to allow the spacecraft to remain bounded near the designed path. Examples of maneuvers that are regularly employed include station-keeping and orbit transfer maneuvers.

Station-keeping maneuvers are designed to allow the spacecraft to maintain bounded motion with respect to a desired orbit. Astrodynamicists often leverage tools from dynamical systems theory and optimization to generate station-keeping maneuvers in high-fidelity multi-body systems. For example, Pavlak and Howell analyzed the correlation between the direction of the locally optimal station-keeping maneuvers with the hyperbolic stable manifold near the underlying periodic orbit [36]. More recently, Farrés et al. [37] further investigate this observed alignment. These observations can be leveraged for the design of impulsive station-keeping maneuvers, as demonstrated by Bosanac et al.: in their work, initial guesses for the station-keeping maneuvers leverage the hyperbolic stable manifold of the orbit, and later refined via optimization techniques for the orbit maintenance of the Nancy Grace Roman Space Telescope [38].

Reconfiguration and orbit transfer maneuvers are used to guide a cluster or a single spacecraft, to a new relative geometric configuration or reference orbit. To date, two-body perturbed dynamics represent the major focus of reconfiguration maneuver design [39–41]. However, the chaotic dynamics of multi-body systems may potentially require new reconfiguration approaches for continuous and impulsive control [42–46]. Reconfiguration maneuvers in multi-body systems are a key component of operating formations: examples include the Laser Interferometer Space Antenna (LISA) [47], the Cluster II [48], the Magnetospheric Multiscale Mission (MMS) [49] and the future HelioSwarm missions [50]. The design of orbit transfer maneuvers, enabling single spacecraft to transit between different solutions in low-energy regimes has been extensively investigated leveraging dynamical system theory and optimization schemes [2, 51–53]. Natural transport mechanisms and families of orbits existing in low-fidelity multi-body systems can be used to generate transfers between prescribed solutions, along with continuous and impulsive control [53–55].

The design of station-keeping, reconfiguration, and orbit transfer maneuvers sometimes demands expertise and involvement of the trajectory designer due to the complex dynamical environment and the impact of uncertainties. This poses a high risk in time-critical scenarios where rapid redesign is required, corresponding to higher costs for operational support. Techniques from Reinforcement Learning (RL) have the potential for significantly reducing of analyst workload. An RL scenario involves training a policy which generates an action from the input observation, and an environment which governs the transition between two observation-action pairs. The policies are trained to maximize a user-defined quantity, the discounted cumulative reward. The trained policy can then be used in the maneuver design process: the trained policy can generate an action, for example a maneuver, upon receiving an input observation, for example the spacecraft state. Various implementations of RL algorithms have already demonstrated their benefits for a reduced human workload in multiple scientific domains [56–58].

In astrodynamics, several researchers have successfully explored state-of-the-art RL algorithms in chaotic multi-body systems for maneuver design in transfer scenarios [59–64], for relative motion around periodic orbits [65] and station-keeping [66–68]. For example, Bonasera et al.

demonstrate the use of a state-of-the-art RL algorithm for autonomous generation of sequences of station-keeping maneuvers in low- and higher-fidelity dynamical models [69]. Among the variety of RL methods, a member of the family of Proximal Policy Optimization algorithms, shortly referred to as PPO, has shown favorable convergence behavior for continuous maneuver generation for spacecraft in chaotic dynamical models [61–64, 67]. In these implementations, PPO is used to train a policy, which is represented by artificial neural networks. The trained networks, serving as universal function approximators, are employed to autonomously generate the required maneuvers and correct the spacecraft state. Leveraging PPO to train neural networks can be beneficial for station-keeping, reconfiguration and orbit transfer scenarios: the trained networks can reduce the workload from an astrodynamicist, and speed up the maneuver design process, especially in time-critical redesign situations.

1.4 Organization of the Manuscript

This dissertation focuses on applying machine learning techniques to different phases of the trajectory design process with the goal of aiding an astrodynamicist by simplifying information extraction of large design spaces, and constructing models for rapid trajectory and maneuver design. This document is structured as follows:

Chapter 2: dynamical models provide mathematical formulations to generate an approximation of the motion of a spacecraft. Two models of progressively higher fidelity are introduced. Initially, the circular restricted three-body problem, often used during the early stages of the trajectory design process, is formulated. Then, the higher fidelity point mass ephemeris model, leveraged to generate end-to-end arcs that fulfill mission requirements, is detailed. Eventually, coordinate frames rotation between rotating and inertial frames are presented.

Chapter 3: the low fidelity circular restricted three-body problem admits a variety of particular solutions that are often investigated during the early stages of the trajectory design process. These solutions, detailed in the chapter, include: periodic orbits, orbits near resonances, invariant tori, and hyperbolic manifolds. Then, numerical methods used to construct these solutions, and to transfer an arc from a low- to a higher-fidelity dynamical model, are presented. Eventually, Poincaré maps, a tool from dynamical system theory used to investigate the characteristics of the solution space, are introduced.

Chapter 4: techniques from machine learning that are used throughout this manuscript are detailed. First, unsupervised learning is highlighted: an overview of two algorithms from clustering and manifold learning, the Hierarchical Density-Based Spatial Clustering with Noise and the Uniform Manifold Approximation and Projection, is introduced. Then, reinforcement learning is presented, focusing on the family of proximal policy optimization algorithms and neural networks.

Chapter 5: techniques from unsupervised learning, comprising clustering, manifold learning, and distributed data mining, are leveraged to partition large high-dimensional datasets of trajectories on a Poincaré map generated in different dynamical models. The generated results are analyzed to demonstrate the benefit of unsupervised learning for an informed extraction of interesting trajectories arc, and establish correlation between groups of solutions across distinct models. Eventually, arcs along the stable hyperbolic manifolds of invariant tori are projected onto the generated result, visually demonstrating their governing nature and the additional benefits of a data-driven summary of the solution space.

Chapter 6: a flexible methodology to design natural transfer connecting invariant tori is presented. A technique from manifold learning is used to aid the identification of possible intersections between connecting arcs. The method is presented for a variety of transfers connecting quasi-periodic trajectories near resonances. Ultimately, the generated single point transfer is used to generate families of natural transfers with similar geometries that connect families of invariant tori.

Chapter 7: an algorithm from reinforcement learning is leveraged to construct models for autonomous generation of maneuver sequences. The algorithm is tested on three scenarios: a station-keeping maneuver framework for a spacecraft near a quasi-halo trajectory in a perturbed point mass ephemeris model in the Sun-Earth system; two orbit transfer scenarios between periodic orbits in the Earth-Moon circular restricted three-body problem. Transfer learning is used to reduce the required computational resources for training.

Chapter 8: the generated results are summarized, and an overview of future work recommendations is presented.

1.5 Contributions

Techniques from machine learning are used in this manuscript within different phases of the trajectory design process. Specifically, challenges associated with three phases of the design process are identified, and distinct methods from machine learning are applied to solve them. Therefore, the main contributions of the presented work include:

Unsupervised learning for higher-dimensional Poincaré maps

1.1) Used distributed data mining, clustering, and manifold learning to partition large datasets composed of high-dimensional trajectories into groups of arcs of similar geometrical features. The goal of this approach is to reduce the visualization burden of complex and high-dimensional design spaces.

1.2) Used a data-driven approach to correlate clusters of geometrically similar trajectories across Poincaré maps constructed in distinct dynamical models. Correlated clusters of distinct maps generated in higher-fidelity models at different values of the independent variable.

1.3) Associated subsets of the generated partitions of the solution space to arcs along the stable manifolds of invariant tori near L_1 and L_2 in the Sun-Earth circular restricted three-body problem.

Manifold learning for constructing natural transfers between invariant tori

2.1) Developed a strategy for generating and correcting an initial guess for a natural transfer between two prescribed invariant tori using a manifold learning algorithm that aids in the identification of arcs existing in high-dimensional and large design spaces.

2.2) Generated families of natural transfers with distinct geometries between invariant tori near resonances in the Earth-Moon circular restricted three-body problem.

Reinforcement learning for autonomous design of impulsive maneuver sequences

3.1) Formulated reinforcement learning environments for training models for autonomous station-keeping and orbit transfer maneuver design.

3.2) Demonstrated use of reinforcement learning to train a policy for autonomous design of sequences of impulsive maneuvers for station-keeping in a point mass ephemeris model. Compared the trained policy with station-keeping maneuver sequences obtained through constrained optimization.

3.3) Demonstrated use of reinforcement learning for constructing policies for autonomous design of sequences of impulsive maneuvers for orbit transfer between prescribed solutions and families of orbits in the Earth-Moon circular restricted three-body problem.

3.4) Demonstrated use of transfer learning to reduce the computational burden of training, and enabling the construction of policies with challenging environment formulations.

Chapter 2

Dynamical Models

Three dynamical models are leveraged in this work to investigate the trajectory solution space and allow the generation of arcs in multi-body systems: the circular restricted three-body problem (CR3BP), the elliptic restricted three-body problem, and a point mass ephemeris model. The first represents a framework where the motion of an assumed massless spacecraft is governed by the gravitational force of two main bodies, co-rotating in a circular motion. Although representing an approximation of the real dynamics governing the spacecraft motion, the CR3BP is often leveraged at the early stages of the trajectory design process to investigate the available solution space. Indeed, the dynamics described by the CR3BP requires significantly lower computational resources for trajectory propagation with respect to higher fidelity models. Higher fidelity dynamical models, like a point mass ephemeris model, more closely represent the real dynamics governing spacecraft motion, and are often used to refine trajectories obtained in the lower fidelity CR3BP. In this chapter, the dynamics of the circular restricted three-body problem is introduced in Sec. 2.1, followed by an overview of the elliptic restricted three-body problem in Sec. 2.2, and a discussion of a higher fidelity point mass ephemeris model in Sec. 2.3. Ultimately, Sec. 2.4 introduces a method to transition a spacecraft state between a rotational and an inertial frame.

2.1 Circular Restricted Three-Body Problem

The equations of motion for the circular restricted three-body problem are obtained from the application of the law of gravitation in a system of three bodies centered in the system barycenter

[70]. A test particle or spacecraft, P_3, is considered moving under the gravitational influence of two main bodies, P_1 and P_2, referred to as the primaries. The mass of each body is referred to as M_i. The position vector of the spacecraft in an inertial frame with axes $\hat{\boldsymbol{X}}\hat{\boldsymbol{Y}}\hat{\boldsymbol{Z}}$ and relative to an origin O is $\boldsymbol{R} = \boldsymbol{R}_3 = X\hat{\boldsymbol{X}} + Y\hat{\boldsymbol{Y}} + Z\hat{\boldsymbol{Z}}$, while the locations of the two primaries are $\boldsymbol{R}_1 = X_1\hat{\boldsymbol{X}} + Y_1\hat{\boldsymbol{Y}} + Z_1\hat{\boldsymbol{Z}}$ and $\boldsymbol{R}_2 = X_2\hat{\boldsymbol{X}} + Y_2\hat{\boldsymbol{Y}} + Z_2\hat{\boldsymbol{Z}}$. The relative position vector of body i with respect to body j is expressed as $\boldsymbol{R}_{ij} = (X_j - X_i)\,\hat{\boldsymbol{X}} + (Y_j - Y_i)\,\hat{\boldsymbol{Y}} + (Z_j - Z_i)\,\hat{\boldsymbol{Z}}$, with ℓ^2-norm $R_{ij} = \|\boldsymbol{R}_{ij}\|$. The equations governing the motion of P_3 with respect to an inertially fixed observer can therefore be obtained as:

$$\tilde{\boldsymbol{R}}'' = -\tilde{G} \sum_{j=1,2} \frac{\tilde{M}_j(\tilde{\boldsymbol{R}}_3 - \tilde{\boldsymbol{R}}_j)}{\tilde{R}_{j3}^3} \tag{2.1}$$

where $\tilde{G} = 6.6730 \times 10^{-20}$ km^3/(kg s^2) is the universal gravitational constant, $(\cdot)'$ represents the time derivative for an inertial observer, and the tilde denotes dimensional quantities. Equation 2.1, together with the dynamics describing the position and velocity of the primaries, constitute a system of 18 differential equations. Therefore, a total of 18 integrals of motion is necessary to solve for the motion of each considered body. However, only 10 constants of motion exist for the investigated system, generated from the conservation of linear momentum, angular momentum and energy. For this reason, an analytical expression explicitly describing the motion of P_3 cannot be formulated and Eq. (2.1) can be solved via numerical propagation.

A variety of simplifying assumptions can be introduced to construct a dynamical model that retains a good level of fidelity, allowing the generation of useful insights with a reduced computational cost. Initially, the mass of the spacecraft is considered negligible with respect to the two primaries. This constitutes a reasonable assumption for the cases treated in this manuscript, since the primaries are represented by massive bodies like by the Earth, the Sun and the Moon, while P_3 is assumed as a spacecraft. Since the gravitational influence exerted by body P_3 on the primaries is negligible, the barycenter of the system is located along the segment connecting the two primaries. It follows that the motion described by the primaries is represented by conics centered at the system barycenter. In the formulation of the dynamics described by the CR3BP, the primaries are assumed following a circular motion, therefore maintaining a constant distance. Moreover, the system barycenter is used as the origin of the inertial frame.

To favor comparison between systems with primaries of different masses and reduce ill-conditioning among state components, the dynamical equations of the CR3BP are often expressed with nondimensional quantities. Three characteristic quantities, associated with distance $l^\star$, mass $m^\star$ and time $t^\star$, are introduced for normalization as

$$l^\star = \tilde{R}_1 + \tilde{R}_2, \qquad m^\star = \tilde{M}_1 + \tilde{M}_2, \qquad t^\star = \sqrt{\frac{(\tilde{R}_1 + \tilde{R}_2)^3}{\tilde{G}(\tilde{M}_1 + \tilde{M}_2)}} \tag{2.2}$$

In the CR3BP, the characteristic distance equals the constant displacement between the two primaries, the characteristic mass represents the mass of the system and the characteristic time is computed to generate a normalized system with a unity mean motion. Also, a mass ratio μ is introduced as

$$\mu = \frac{\tilde{M}_2}{\tilde{M}_1 + \tilde{M}_2} \tag{2.3}$$

Examples of characteristic quantities for the Sun-Earth and the Earth-Moon systems are reported in Table 2.1.

System	$t^\star$ [s]	$l^\star$ [km]	μ
Sun-Earth	5.02264×10^6	149597870	3.00348×10^{-6}
Earth-Moon	3.75190×10^5	384400	0.01215

Table 2.1: Characteristic quantities and mass ratios for the Sun-Earth and the Earth-Moon systems.

The characteristic quantities, together with the mass ratio, support writing the nondimensional equations of motion of the CR3BP in an inertial frame centered at the system barycenter. However, the equations still manifest an explicit dependence from the nondimensional time. To express the equations of motion in an autonomous configuration, a rotating frame with axes $\hat{\boldsymbol{x}}\hat{\boldsymbol{y}}\hat{\boldsymbol{z}}$ and the same origin as the inertial frame is introduced. The spacecraft location in the inertial frame (X, Y, Z), is transformed with a rotation matrix into a nondimensional spacecraft location in the rotating frame (x, y, z) as:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{2.4}$$

where θ represents the angle between the $\hat{\boldsymbol{X}}$- and the $\hat{\boldsymbol{x}}$-axis, as represented in Fig. 2.1. The nondimensional locations of the primaries in the defined rotating frame are $(-\mu, 0, 0)$ and $(1 - \mu, 0, 0)$. The position vectors of the spacecraft from the primaries are $\boldsymbol{r}_1 = (x + \mu)\hat{\boldsymbol{x}}$ and $\boldsymbol{r}_2 = (x - 1 + \mu)\hat{\boldsymbol{x}}$.

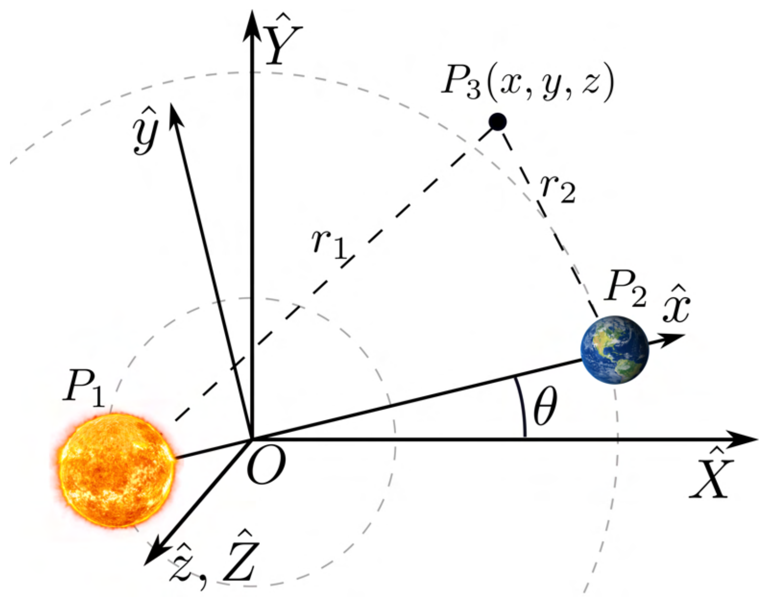

Figure 2.1: Geometry of the circular restricted three-body problem (celestial bodies not to scale).

The constructed rotation allows to express the nondimensional equations of motion governing the spacecraft in the rotating frame as:

$$\begin{cases} \ddot{x} = 2\dot{y} + x - \dfrac{(1-\mu)(x+\mu)}{r_1^3} - \dfrac{\mu(x-1+\mu)}{r_2^3} \\ \ddot{y} = -2\dot{x} + y - \dfrac{(1-\mu)y}{r_1^3} - \dfrac{\mu y}{r_2^3} \\ \ddot{z} = -\dfrac{(1-\mu)z}{r_1^3} - \dfrac{\mu z}{r_2^3} \end{cases} \tag{2.5}$$

where $\dot{(\cdot)}$ represent the time derivative for an observer in the rotating frame, while $r_1 = \|\boldsymbol{r}_1\|$ and $r_2 = \|\boldsymbol{r}_2\|$ are the distances of the spacecraft from the primaries. The system of ordinary differential equation expressed in Eq. (2.5) can be employed to propagate an initial spacecraft nondimensional state in the rotational frame $\boldsymbol{x} = (x, y, z, \dot{x}, \dot{y}, \dot{z})$ [70]. Ultimately, the dynamics expressed by Eq. (2.5) allows the existence of three symmetries:

1) Image of the xy-plane: if a trajectory $\boldsymbol{x}_1(t) = [x(t), y(t), z(t), \dot{x}(t), \dot{y}(t), \dot{z}(t)]$ exists, the mirrored trajectory with respect to the xy-plane, $\boldsymbol{x}_2(t) = [x(t), y(t), -z(t), \dot{x}(t), \dot{y}(t), -\dot{z}(t)]$, also satisfies Eq. (2.5).

2) Backward image of the xz-plane: if a trajectory $\boldsymbol{x}_1(t) = [x(t), y(t), z(t), \dot{x}(t), \dot{y}(t), \dot{z}(t)]$ exists, the backward trajectory, mirrored with respect to the xz-plane, $\boldsymbol{x}_2(t) = [x(-t), -y(-t), z(-t), \dot{x}(-t), -\dot{y}(-t), \dot{z}(-t)]$, also satisfies Eq. (2.5).

3) Backward image of the x-axis: if a trajectory $\boldsymbol{x}_1(t) = [x(t), y(t), z(t), \dot{x}(t), \dot{y}(t), \dot{z}(t)]$ exists, the backward trajectory $\boldsymbol{x}_2(t) = [x(-t), -y(-t), -z(-t), \dot{x}(-t), -\dot{y}(-t), -\dot{z}(-t)]$ mirrored with respect to the x-axis, also satisfies Eq. (2.5)

These known symmetries can be leveraged in trajectory design scenarios to extend the investigated solution space, without the actual computation of symmetric arcs [71].

2.1.1 Jacobi Constant

The system in Eq. (2.5) allows the existence of a constant of motion, called the Jacobi constant. To compute this constant of motion, a pseudo-potential function can be expressed as:

$$U(x, y, z) = \frac{x^2 + y^2}{2} + \frac{1-\mu}{r_1} + \frac{\mu}{r_2} \tag{2.6}$$

The system in Eq. (2.5) can be expressed in a compact notation using the pseudo-potential function:

$$\begin{cases} \ddot{x} - 2\dot{y} &= U_x, \\ \ddot{y} + 2\dot{x} &= U_y \\ \ddot{z} &= U_z \end{cases} \tag{2.7}$$

where the subscripts $(\cdot)_i$ indicates the partial derivative with respect to the generic component i, with $i \in \{x, y, z\}$. To obtain the expression of the Jacobi constant, the scalar product between the system in Eq. (2.7) and the velocity $(\dot{x}, \dot{y}, \dot{z})$ is computed, generating:

$$\ddot{x}\dot{x} + \ddot{y}\dot{y} + \ddot{z}\dot{z} = \dot{x}U_x + \dot{y}U_y + \dot{z}U_z \tag{2.8}$$

This equation can be integrated, obtaining:

$$\frac{v^2}{2} = \int \mathrm{d}U - \dot{U} \tag{2.9}$$

where the term $\dot{U}$ cancels since the pseudo-potential does not explicitly depends on the nondimensional time, and the magnitude of the velocity $v = \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}$. Additionally, Eq. (2.9) can be rearranged to generate the Jacobi constant as:

$$C_J = 2U - v^2 = x^2 + y^2 + \frac{2(1-\mu)}{r_1} + \frac{2\mu}{r_2} - \dot{x}^2 - \dot{y}^2 - \dot{z}^2 \tag{2.10}$$

Large and low values of Jacobi constant are associated with low and high energy, respectively. Moreover, the expression in Eq. (2.10) remains constant along a spacecraft trajectory propagated in a CR3BP model.

2.1.2 Equilibrium Points

The equations of motion for the CR3BP admits the existence of five equilibrium points in the rotating frame, called the Lagrangian points L_i, with $i \in \{1, 2, \dots, 5\}$ and defined as $\mathbb{L} = \{\boldsymbol{x} \in \mathbb{R}^6 |\, \dot{\boldsymbol{x}} = \boldsymbol{0}\}$. The location of these points can be derived from the system in Eq. (2.7), generating:

$$\begin{cases} U_x = x - \dfrac{(1-\mu)(x+\mu)}{r_1^3} - \dfrac{\mu(x-1+\mu)}{r_2^3} = 0 \\ U_y = y - \dfrac{(1-\mu)y}{r_1^3} - \dfrac{\mu y}{r_2^3} = 0 \\ U_z = -\dfrac{(1-\mu)z}{r_1^3} - \dfrac{\mu z}{r_2^3} = 0 \end{cases} \tag{2.11}$$

The third equation is satisfied when $z = 0$, therefore locating the Lagrangian points on the plane defined by the motion of the primaries. The first three Lagrangian points, also known as Euler-Lagrange points, or as collinear points, can be obtained by satisfying the second equation with

$y = 0$. Three regions exist on the line where an equilibrium point can be found, satisfying the system in Eq. (2.11), namely:

$$(x+\mu) > 0, (x-1+\mu) > 0, \qquad (x+\mu) > 0, (x-1+\mu) < 0, \qquad (x+\mu) < 0, (x-1+\mu) < 0 \quad (2.12)$$

The equibilibrium points in each of these three regions can be located for example with numerical root-solving algorithms to identify a value of x that satisfies $U_x = 0$. The location of the remaining two equilibrium points, also called triangular points, can be computed by setting $r_1 = r_2 = 1$ in Eq. (2.11). Figure 2.2 depicts the location of the equilibrium points for the Earth-Moon CR3BP.

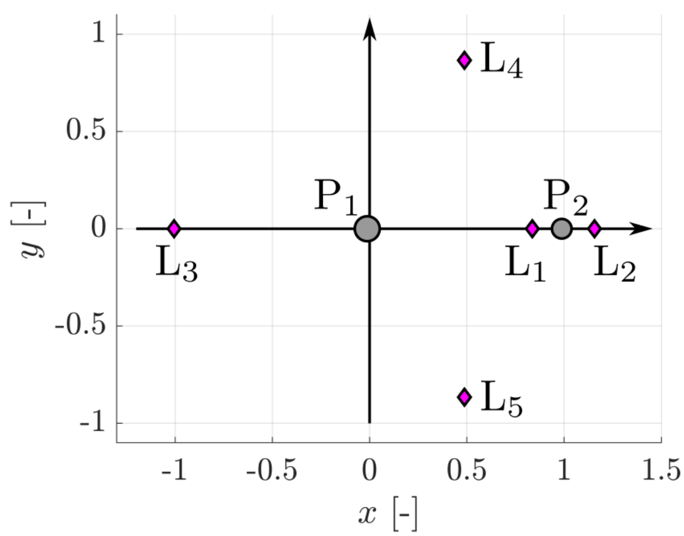

Figure 2.2: Location of the five Lagrange points and the primaries for the Earth-Moon CR3BP.

2.1.3 Zero Velocity Surfaces

At a single value of Jacobi constant, the available design space in the CR3BP can be divided into allowable and forbidden regions. Indeed, Eq. (2.10) allows to express the magnitude of the velocity as a function of the Jacobi constant and the pseudo-potential. Since imaginary values of velocity do not generate feasible spacecraft states, a surface called Zero Velocity Surface (ZVS) can be constructed as:

$$\left\{ (x, y, z) \in \mathbb{R}^3 \middle| x^2 + y^2 + \frac{2(1-\mu)}{r_1} + \frac{2\mu}{r_2} - C_J = 0 \right\} \tag{2.13}$$

This surface separates regions of motion where the velocity is real from regions where the velocity is imaginary. The spacecraft cannot enter regions where the velocity magnitude takes imaginary values. The intersection of the ZVS with the hyperplane $z = 0$ is often referred to as the Zero Velocity Curve (ZVC). Figures 2.3 and 2.4 present a visualization of the ZVC and ZVS at a variety of energy levels. In each subfigure, the two primaries and the Lagrangian points are depicted with grey circles and magenta diamonds, respectively. In particular, Fig. 2.3 presents four subfigures, reporting the ZVC in black and the forbidden region in cyan, in the Earth-Moon system. From Fig. 2.3 (a) to (d), the energy level is progressively increased, or alternatively the Jacobi constant is progressively decreased, starting from a value of $C_J > C_J(\mathrm{L}_1)$ and ending at a value $C_J < C_J(\mathrm{L}_{4/5})$. The same levels of energy are leveraged in Fig. 2.4 to depict the equivalent spatial representations of the ZVS, with semi-transparent blue markers. Figures 2.3 and2.4 allow to visualize the design space progressively available when increasing the energy level (decreasing C_J). Indeed, the motion is allowed in three disconnected regions when $C_J > C_J(\mathrm{L}_1)$, represented by: 1) a region near P_1; 2) a region surrounding P_2; 3) an external region. When $C_J(\mathrm{L}_1) < C_J < C_J(\mathrm{L}_2)$, the bottleneck at L_1 opens, connecting the regions around the primaries. Similarly, when $C_J(\mathrm{L}_2) < C_J < C_J(\mathrm{L}_3)$, the bottleneck at L_2 opens, allowing the spacecraft to ballistically transition between the interior and exterior region. When $C_J < C_J(\mathrm{L}_{4/5})$ the entire $z = 0$ hyperplane is contained in the allowable region, although the ZVS still limits the available design space when $z \neq 0$.

2.2 Elliptic Restricted Three-Body Problem

The Elliptic Restricted Three-Body Problem (ER3BP) is an approximated dynamical model that represents a higher-fidelity formulation of the CR3BP. In the ER3BP, two primaries are assumed following elliptical orbits about the system barycenter. Therefore, the distance between P_1 and P_2 is not constant as the primaries follow conics with a nonzero eccentricity. For example, the eccentricity is assumed equal to $e_P \approx 0.0167$ in the Sun-Earth system. As a result, the primaries only appear at fixed locations over time in a pulsating and rotating frame. Moreover, the true anomaly, f, of the primary system is used as an independent variable, rather than time, with $f = 0$

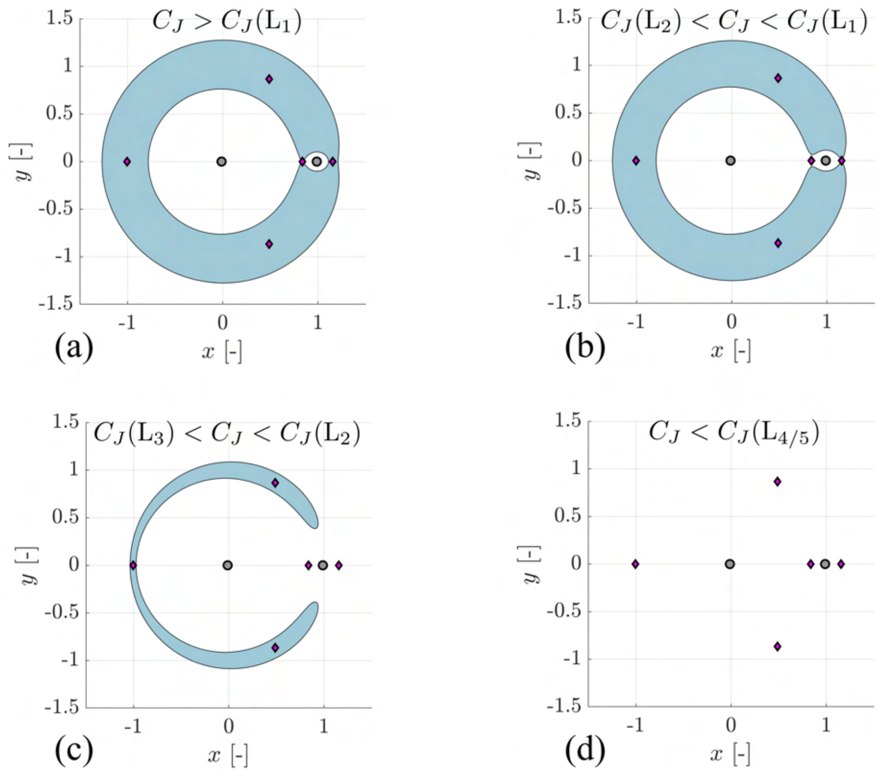

Figure 2.3: Zero-velocity curves for a set of Jacobi constants in the Earth-Moon system.

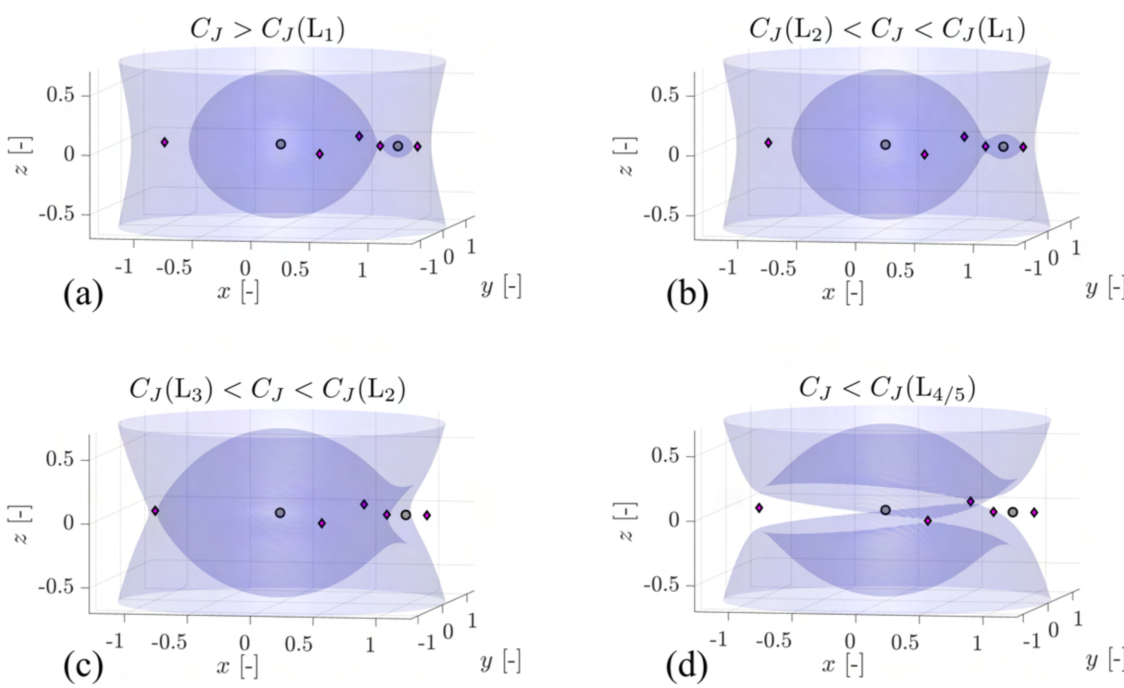

Figure 2.4: Zero-velocity surfaces for a set of Jacobi constants in the Earth-Moon system.

associated to an initial configuration with the primaries located at the periapsis [70]. Using this configuration, the nondimensional equations of motion for a spacecraft in the ER3BP are [70]:

$$\begin{cases} x^{\dagger\dagger} - 2y^{\dagger} = \omega_x \\ y^{\dagger\dagger} + 2x^{\dagger} = \omega_y \\ z^{\dagger\dagger} = \omega_z \\ t^{\dagger} = (1 - e_P^2)^{3/2}/(1 + e_P \cos(f))^2 \end{cases} \tag{2.14}$$

where the prime $(\cdot)^{\dagger}$ indicates in Eq. (2.14) a derivative with respect to the true anomaly f, and the pseudo-potential function is:

$$\omega(\boldsymbol{x}, f) = \frac{U(\boldsymbol{x}) - z^2 e_P \cos(f)/2}{1 + e_P \cos(f)} \tag{2.15}$$

Due to the explicit dependence of Eq. (2.14) on the time-like quantity f, an integral of motion does not exist [72].

2.3 Point Mass Ephemeris Model

A point mass ephemeris model is often used in trajectory design to generate a higher-fidelity representation of the dynamical model, and obtain a verification of the trajectory constructed with the restricted model. In the point mass ephemeris model used in this investigation, a spacecraft is assumed having a negligible mass with respect to the considered celestial bodies. The motion of the spacecraft is influenced by the gravitational attraction of N_e bodies, that are assumed to be spherically symmetric, therefore modeled as point masses. In an inertial orthogonal frame $(\hat{\boldsymbol{X}}, \hat{\boldsymbol{Y}}, \hat{\boldsymbol{Z}})$ with inertially-fixed origin O, the nondimensional spacecraft state is expressed as $\boldsymbol{X} = [\tilde{X}, \tilde{Y}, \tilde{Z}, \dot{\tilde{X}}, \dot{\tilde{Y}}, \dot{\tilde{Z}}]^T$. A conceptual visualization of the system configuration in the inertial frame, supporting the point mass ephemeris model description in this section, is presented in Fig. 2.5. It may be more convenient to express the spacecraft state and the associated dynamics with respect to a central body P_j. Recall that in this case the relative spacecraft state is expressed as $\tilde{\boldsymbol{R}}_{j3} = \left(\tilde{X} - \tilde{X}_j\right)\hat{\boldsymbol{X}} + \left(\tilde{Y} - \tilde{Y}_j\right)\hat{\boldsymbol{Y}} + \left(\tilde{Z} - \tilde{Z}_j\right)\hat{\boldsymbol{Z}}$, with ℓ^2-norm $\tilde{R}_{j3} = \|\tilde{\boldsymbol{R}}_{j3}\|$. A set of second-order differential equations governing the dynamics of P_3 in a P_j-centered J2000 inertial coordinate frame, relative to an inertial observer, takes the form [52, 73]:

$$\tilde{\boldsymbol{R}}''_{j,3} = \tilde{\boldsymbol{f}}_g + \tilde{\boldsymbol{f}}_p \tag{2.16}$$

where $\boldsymbol{f}_g$ represents the sum of the gravitational forces associated with the considered celestial bodies, assumed as point masses, while $\boldsymbol{f}_p$ represents the perturbative forces. If only gravitational forces are considered in the dynamical representation, Eq. (2.16) is expressed as:

$$\tilde{\boldsymbol{R}}''_{j,3} = -\frac{\tilde{G}\tilde{M}_j}{\tilde{R}^3_{j,3}}\tilde{\boldsymbol{R}}_{j,3} + \tilde{G}\sum_{\substack{i=1 \\ i\neq j,3}}^{N_e} \tilde{M}_i\left(\frac{\tilde{\boldsymbol{R}}_{3,i}}{\tilde{R}^3_{3,i}} - \frac{\tilde{\boldsymbol{R}}_{j,i}}{\tilde{R}^3_{j,i}}\right) \tag{2.17}$$

Equations of motion of the other $N_e - 1$ bodies are not included in the system of equations: the states of the considered celestial bodies are available via the JPL's DE421 ephemeris files [52, 74].

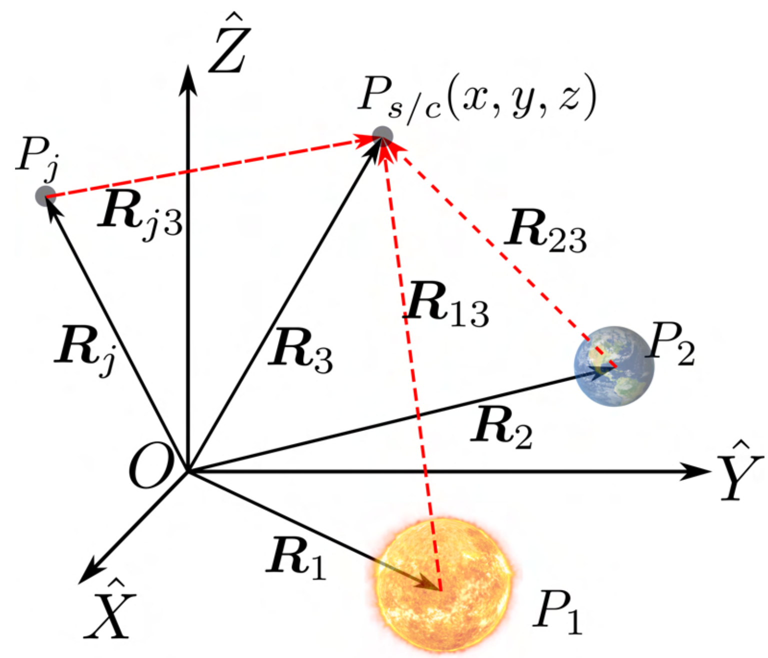

Figure 2.5: Geometry of the N_e-body problem (celestial bodies not to scale).

Additional perturbing forces, like solar radiation pressure (SRP) and spherical harmonics, can be straightforwardly incorporated into the right-hand side of Eq. (2.17). In this work, only SRP is included to augment the fidelity of the leveraged dynamical model. The dynamical impact of SRP can be included in the perturbative term $\boldsymbol{f}_p$, in the right-hand side of Eq. (2.16). In particular, the SRP acceleration is modeled as:

$$\boldsymbol{f}_p = \frac{kAS_0}{cM_3}\frac{R_0^2}{R_{\text{Sun},3}^3}\boldsymbol{R}_{\text{Sun},3} \tag{2.18}$$

where M_3 is the mass of the spacecraft, k is a constant reflectivity index, A is the cross-sectional area, $S_0 = 1.358 \times 10^3$ W/m^2 is the solar flux at $R_0 = 149597870$ km and $c = 299792.458$ m/s is the speed of light [38].

2.4 Coordinate Frame Transformations

In multi-body trajectory design, the investigated arc is often visualized in different coordinate frames to generate useful insights into the geometrical features of the designed trajectory. Moreover, coordinate frame transormations are employed when refining a trajectory obtained in the CR3BP into a higher-fidelity model centered at an arbitrary body P_j. For these reasons, the methodologies to transition a trajectory between rotating and inertial frames are highlighted in this section.

2.4.1 Transforming from Rotating to Inertial Frame in the CR3BP

Transforming a state expressed in the rotating coordinate frame into an inertial frame fixed at an arbitrary body P_j is accomplished with a rotation matrix. Indeed, by inverting Eq. (2.4), the spacecraft position in the inertial frame (X, Y, Z) can be retrieved from the spacecraft position expressed in the rotating frame (x, y, z) with a rotation of θ about the $\hat{\boldsymbol{Z}}$-axis as:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{2.19}$$

The transport theorem allows to construct the matrix for the full-state rotation. Indeed, the time derivative of the position vector as seen by an observer in the inertial frame can be expressed as a function of the time derivative of the position vector as seen by an observer in the rotating frame as:

$$\frac{{}^{I}\mathrm{d}\boldsymbol{r}}{\mathrm{d}t} = \frac{{}^{R}\mathrm{d}\boldsymbol{r}}{\mathrm{d}t} + \boldsymbol{\omega}_{IR} \times \boldsymbol{r} \tag{2.20}$$

where the prescripts ${}^{I}(\cdot)$ and ${}^{R}(\cdot)$ are associated with the observers in the inertial and rotating frames, and $\boldsymbol{\omega}_{IR} = \omega\hat{\boldsymbol{z}}$ is the rotation rate of the rotating frame with respect to the inertial one, with $\omega = 1$ following the assumption of the CR3BP. Equation 2.20 can be used to express the components of the velocity of the inertial frame as a function of the rotating state. The compact state transformation is formulated in matrix form as:

$$\begin{bmatrix} X \\ Y \\ Z \\ \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ -\sin(\theta) & -\cos(\theta) & 0 & \cos(\theta) & -\sin(\theta) & 0 \\ \cos(\theta) & -\sin(\theta) & 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \tag{2.21}$$

To transform a spacecraft state defined in an inertial frame into a spacecraft state expressed in the rotating frame, Eq. (2.21) is multiplied on the left-side by the inverse of the rotational matrix. Moreover, it is often convenient to express the spacecraft state in a rotating frame centered on the generic body P_j. For this reason, prior to applying the transformation in Eq. (2.21), the state is translated to be centered at the required central body P_j.

2.4.2 Transforming from Rotating to ICRF Inertial Frame in Ephemeris Model

A spacecraft state defined in the rotating frame in the CR3BP can be transformed into a spacecraft state expressed in a P_j-centered inertial frame that uses the axes of the ICRF with a similar procedure to what detailed in Sec. 2.4.1. Specifically, at any given time, or epoch, t, the spacecraft state in the nondimensional rotating frame is scaled in dimensional quantities using the instantaneous characteristic quantities. The latter are obtained using Eq. (2.2), although leveraging the instantaneous characteristic distance $l^{\star}(t)$. Positional quantities are scaled with $l^{\star}(t)$, while velocities by $l^{\star}(t)/t^{\star}(t)$. Moreover, the spacecraft state is translated to be either P_1-

or P_2-centered. To help readability, this section assumes the spacecraft state is rotated in a P_1-centered inertial frame using axes from the ICRF. However, minor modifications to the detailed procedure can be incorporated to rotate the spacecraft in a P_2-centered inertial frame with axes form the ICRF. The dimensional state in the rotating frame, centered at P_1 at epoch t, is defined as $\tilde{\boldsymbol{x}}_{1,3} \in \mathbb{R}^6$, while the dimensional P_1-centered state in the inertial frame is defined as $\tilde{\boldsymbol{X}}_{1,3} \in \mathbb{R}^6$. The explicit dependence of the instantaneous quantities computed in this sections is dropped to ease readability [52].

To retrieve the instantaneous rotating matrix at epoch t, the relative position and velocity vector of the P_2 with respect to P_1 are retrieved from an ephemeris file as $\tilde{\boldsymbol{R}}_{2,1}$ and $\tilde{\boldsymbol{V}}_{2,1}$ and are expressed using axes from the ICRF. Three unit vectors can be defined as [52]:

$$\hat{\tilde{\boldsymbol{x}}} = \frac{\tilde{\boldsymbol{R}}_{2,1}}{\tilde{R}_{2,1}}, \qquad \hat{\tilde{\boldsymbol{z}}} = \frac{\tilde{\boldsymbol{R}}_{2,1} \times \tilde{\boldsymbol{V}}_{2,1}}{\|\tilde{\boldsymbol{R}}_{2,1} \times \tilde{\boldsymbol{V}}_{2,1}\|}, \qquad \hat{\tilde{\boldsymbol{y}}} = \hat{\tilde{\boldsymbol{z}}} \times \hat{\tilde{\boldsymbol{x}}} \tag{2.22}$$

These three unit vectors, defined at t, are used to populate a first rotational matrix, allowing to express a dimensional position vector in the P_1-centered inertial frame as a function of a position vector defined in the rotating frame as:

$$\begin{bmatrix} \tilde{X}_{1,3} \\ \tilde{Y}_{1,3} \\ \tilde{Z}_{1,3} \end{bmatrix} = \begin{bmatrix} \hat{\tilde{\boldsymbol{x}}} & \hat{\tilde{\boldsymbol{y}}} & \hat{\tilde{\boldsymbol{z}}} \end{bmatrix} \begin{bmatrix} \tilde{x}_{1,3} \\ \tilde{y}_{1,3} \\ \tilde{z}_{1,3} \end{bmatrix} = \begin{bmatrix} \hat{\tilde{x}}_x & \hat{\tilde{y}}_x & \hat{\tilde{z}}_x \\ \hat{\tilde{x}}_y & \hat{\tilde{y}}_y & \hat{\tilde{z}}_y \\ \hat{\tilde{x}}_z & \hat{\tilde{y}}_z & \hat{\tilde{z}}_z \end{bmatrix} \begin{bmatrix} \tilde{x}_{1,3} \\ \tilde{y}_{1,3} \\ \tilde{z}_{1,3} \end{bmatrix} \tag{2.23}$$

The transport theorem in Eq. (2.20) is again used to compute the rotation matrix and obtain the velocity vector in the P_1-centered inertial frame, given the state defined in the rotating frame centered at P_1. When transitioning to an inertial frame in a point mass ephemeris model, the angular rate is not constant and equal to unity, therefore it must be computed at each epoch as:

$$\boldsymbol{\omega}_{IR} = \frac{\|\tilde{\boldsymbol{R}}_{2,1} \times \tilde{\boldsymbol{V}}_{2,1}\|}{\tilde{R}_{2,1}^2} \hat{\tilde{\boldsymbol{z}}} = \omega \hat{\tilde{\boldsymbol{z}}} \tag{2.24}$$

where it is noted how each quantity is instantaneously retrieved, given the current epoch t. Then, the full state transformation is:

$$\begin{bmatrix} \tilde{X}_{1,3} \\ \tilde{Y}_{1,3} \\ \tilde{Z}_{1,3} \\ \dot{\tilde{X}}_{1,3} \\ \dot{\tilde{Y}}_{1,3} \\ \dot{\tilde{Z}}_{1,3} \end{bmatrix} = \begin{bmatrix} \hat{\tilde{x}}_x & \hat{\tilde{y}}_x & \hat{\tilde{z}}_x & 0 & 0 & 0 \\ \hat{\tilde{x}}_y & \hat{\tilde{y}}_y & \hat{\tilde{z}}_y & 0 & 0 & 0 \\ \hat{\tilde{x}}_z & \hat{\tilde{y}}_z & \hat{\tilde{z}}_z & 0 & 0 & 0 \\ \omega\hat{\tilde{y}}_x & -\omega\hat{\tilde{x}}_x & 0 & \hat{\tilde{x}}_x & \hat{\tilde{y}}_x & \hat{\tilde{z}}_x \\ \omega\hat{\tilde{y}}_y & -\omega\hat{\tilde{x}}_y & 0 & \hat{\tilde{x}}_y & \hat{\tilde{y}}_y & \hat{\tilde{z}}_y \\ \omega\hat{\tilde{y}}_z & -\omega\hat{\tilde{x}}_z & 0 & \hat{\tilde{x}}_z & \hat{\tilde{y}}_z & \hat{\tilde{z}}_z \end{bmatrix} \begin{bmatrix} \tilde{x}_{1,3} \\ \tilde{y}_{1,3} \\ \tilde{z}_{1,3} \\ \dot{\tilde{x}}_{1,3} \\ \dot{\tilde{y}}_{1,3} \\ \dot{\tilde{z}}_{1,3} \end{bmatrix} \tag{2.25}$$

To summarize, the procedure to transform a state in the rotational frame at a generic epoch t into the inertial frame with axes from the ICRF, centered at P_1, can be divided into three phases:

1) Translate the rotational state from the barycenter of the P_1–P_2 system to P_1, and dimensionalize it with the instantaneous characteristic quantities.

2) Generate the instantaneous unit vectors in Eq. (2.22), and the instantaneous angular rate from Eq. (2.24).

3) Populate the rotational matrix in Eq. (2.25) and rotate the state.

To obtain a state in the rotating frame, centered at the P_1-P_2 barycenter, from a P_1-centered state defined in the inertial frame with axes from the ICRF, the previous steps are inverted and slighlty modified to generate the following sequence, repeated at each epoch [52]:

1) Generate the instantaneous unit vectors in Eq. (2.22), and the instantaneous angular rate from Eq. (2.24).

2) Populate the inverse of the rotational matrix in Eq. (2.25) and rotate the state to obtain the dimensional state, defined in the P_1-centered rotational frame (in the right-hand side of Eq. (2.25)).

3) Translate the state from the P_1-centered frame to the barycenter of the system, and scale it to generate a nondimensional representation.

Chapter 3

Particular Solutions

The dynamics of the circular restricted three-body problem allows the existence of a variety of particular solutions, including equilibrium points, periodic orbits, and quasi-periodic trajectories. With a fixed mass ratio μ, the equilibrium solutions appear at fixed locations in the rotating frame. Periodic and quasi-periodic solutions exist in families, some possessing unstable members that generate stable and unstable invariant manifolds. The particular solutions obtained by periodic orbits and quasi-periodic trajectories are studied, during the early phases of the trajectory design process, to generate insights in the available trajectory design space. Among the techniques astrodynamicists often leverage to investigate the solution space, Poincaré maps represent a widespread tool to study the characteristics of higher-dimensional dynamical flows. After generating these particular solutions in low-fidelity dynamics, trajectories are often corrected in higher-fidelity models to provide solutions retaining similar geometrical features. The refined trajectories are leveraged at the later stages of the trajectory design process to provide an end-to-end solution that fulfills the mission requirements. For all these reasons, this chapter provides an overview of the generation process and the salient dynamical characteristics of the particular solutions in the CR3BP. In particular, the characteristics of the periodic solutions and the numerical correction schemes adopted in this investigation to generate families or periodic orbits are presented in Secs. 3.1 and 3.2. Then, a particular family of periodic orbits existing in the CR3BP, the orbits near resonances, are detailed in Sec. 3.4. Quasi-periodic trajectories are outlined in Sec. 3.5, followed by an overview of the stable and unstable manifolds generated from unstable periodic and quasi-periodic solutions in Sec. 3.6.

The Poincaré maps are introduced in Sec. 3.7, before presenting an outline of the techniques used to transition a trajectory from a low- to a higher-fidelity dynamical model in Sec. 3.8.

3.1 Periodic Orbits

The autonomous CR3BP allows the existence of a variety of periodic orbits, distributed in continuous families. A trajectory $\boldsymbol{x}(t)$ represents a periodic orbit if:

$$\exists T \in \mathbb{R} : \boldsymbol{x}(t) = \boldsymbol{x}(t+T), \forall t \in \mathbb{R} \tag{3.1}$$

If T exists, an infinite set of periods can satisfy the Eq. (3.1), as nT, $\forall n \in \mathbb{N}^+$. However, only the condition $n = 1$ is leveraged to identify a $T-$periodic orbit in this investigation. In the CR3BP, the stability of the underlying solution largely affects the nearby dynamical flows, generating nearby structures that are often employed in trajectory design.

The stability of a periodic orbit in the CR3BP influences nearby trajectories. General insights concerning the stability of a periodic orbit in the CR3BP can be generated by observing the dynamical characteristics of the flow nearby the analyzed orbit. A generic periodic orbit in the CR3BP can be defined as a sequence of points in time $\boldsymbol{x}_R(t) \in \mathbb{R}^6$, with an initial condition $\boldsymbol{x}_R(t_0)$. From an initial perturbation $\delta\boldsymbol{x}(t_0) \neq \boldsymbol{0}$, a trajectory near the periodic orbit starting from $\boldsymbol{x}_R(t_0) + \delta\boldsymbol{x}(t_0)$ can be generated as $\boldsymbol{x}(t) = \boldsymbol{x}_R(t) + \delta\boldsymbol{x}(t)$. The equations of motion for the perturbed trajectory can be linearized with respect to the reference solution as:

$$\dot{\boldsymbol{x}}(t) = \dot{\boldsymbol{x}}_R(t) + \delta\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}_R(t_0) + \delta\boldsymbol{x}(t_0)) \tag{3.2}$$

where $\boldsymbol{f}(\cdot)$ is the vector field expressing the dynamics from Eq. (2.5). Equation 3.2 can be linearized in a neighborhood of the periodic orbit $\boldsymbol{x}_R(t)$ as:

$$\dot{\boldsymbol{x}}(t) \approx \boldsymbol{f}(\boldsymbol{x}_R(t_0)) + \left.\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}}\right|_{\boldsymbol{x}(t)} \delta\boldsymbol{x}(t) \tag{3.3}$$

This equation can be manipulated to express the dynamics of the perturbation from the reference:

$$\dot{\boldsymbol{x}}(t) - \boldsymbol{f}(\boldsymbol{x}_R(t_0)) = \delta\dot{\boldsymbol{x}}(t) \approx \left.\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}}\right|_{\boldsymbol{x}_R(t)} \delta\boldsymbol{x}(t) \tag{3.4}$$

with solution that takes the form:

$$\delta\boldsymbol{x}(t) = \boldsymbol{\Phi}(t, t_0)\delta\boldsymbol{x}(t_0) \tag{3.5}$$

Equation 3.5 introduces the state transition matrix (STM) as $\boldsymbol{\Phi}(t, t_0) \in \mathbb{R}^{6\times 6}$, which linearly maps initial state deviations $\delta\boldsymbol{x}(t_0)$ into future state deviations $\delta\boldsymbol{x}(t)$, assuming linearized motion. To retrieve the state transition matrix, Eqs. (3.4) and (3.5) are used to obtain:

$$\dot{\boldsymbol{\Phi}}(t, t_0) = \left.\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}}\right|_{\boldsymbol{x}_R(t)} \boldsymbol{\Phi}(t, t_0), \qquad \boldsymbol{\Phi}(t_0, t_0) = \mathbf{I}_{6\times 6} \tag{3.6}$$

where $\mathbf{I}_{6\times 6} \in \mathbb{R}^{6\times 6}$ is the identity matrix. Equation 3.6 is usually appended to the system of six differential equations in Eq. (2.5), defining a system of 42 ordinary differential equations. The appended system of equations is propagated to generate the sequence of spacecraft states and state transition matrix in time.

The stability of the underlying orbit can be assessed by leveraging Lyapunov theory. Indeed, given a reference solution $\boldsymbol{x}_R(t)$ (the periodic orbit) with an initial state $\boldsymbol{x}_R(t_0)$, a trajectory $\boldsymbol{x}(t)$ can be generated with a perturbed initial condition $\boldsymbol{x}(t_0) = \boldsymbol{x}_R(t_0) + \delta\boldsymbol{x}(t_0)$. The reference trajectory can be labeled as stable, unstable or asymptotically stable, according to:

- Stable, if $\exists\, \delta > 0 : \; \|\boldsymbol{x}(t) - \boldsymbol{x}_R(t)\| \leq \delta \;\forall t$.
- Unstable, if the nearby solution $\boldsymbol{x}(t)$ naturally departs from $\boldsymbol{x}_R(t)$.
- Asymptotically stable, if $\lim_{t\to\infty} \|\boldsymbol{x}(t) - \boldsymbol{x}_R(t)\| = 0$.

A particular form of the propagated STM is leveraged to retrieve information concerning the stability of the underlying orbit. The monodromy matrix, representing the STM propagated for one period $\boldsymbol{M} = \boldsymbol{\Phi}(t_0 + T, t_0)\,, \forall\, t_0 \in \mathbb{R}$, contains information on how an initial state deviation from the underlying periodic orbit is mapped by the linearized dynamical flow nearby the periodic orbit after one period. Moreover, the final perturbation of the state from the reference orbit after a number of periods $l \in \mathbb{N}$ can be decomposed leveraging the chain rule of the STM as [75]:

$$\delta\boldsymbol{x}(T^l) = \boldsymbol{\Phi}(t_0 + T^l, t_0)\delta\boldsymbol{x}(t_0) = \boldsymbol{M}^l\delta\boldsymbol{x}(t_0) \tag{3.7}$$

Therefore, $\boldsymbol{M}$ scales the initial perturbation from the reference solution to obtain the final deviation after one period. According to the Lyapunov stability definition, if the scaling effect of $\boldsymbol{M}$ is such that $\|\delta\boldsymbol{x}(T^l)\|$ remains contained within a small number δ, the reference orbit is stable. Analogously, if $\|\delta\boldsymbol{x}(T^l)\| \rightarrow 0$ the reference orbit is asymptotically stable, while it is unstable if $\|\delta\boldsymbol{x}(T^l)\|$ is unbounded. To investigate the scaling effect of $\boldsymbol{M}$ on the final deviation, Floquet theory is leveraged. The STM can be decomposed as a product of two T-periodic matrices and a diagonal matrix reporting the Floquet multipliers. For the monodromy matrix, the multiplicative T-periodic matrices are equivalent, therefore the Floquet multipliers are associated with the eigenvalues $\boldsymbol{M}$. For this reason, the stability characteristic of the flow nearby a periodic orbit is studied through the eigenvalues of the associated monodromy matrix [73, 75].

Due to the sympletic nature of the CR3BP, the $\boldsymbol{M}$ matrix has three distinct pair of eigenvalues λ. It can be demonstrated that one pair is represented by the trivial $\lambda_{1,2} = \pm 1$, associated with the motion along the direction of the periodic orbit. The remaining two nontrivial pairs of eigenvalues are therefore investigated to generate insights into the stability of the underlying periodic orbit. The nontrivial pairs can be represented as either reciprocal, where $\lambda_1 = 1/\lambda_2$, or as complex conjugate, where $\lambda_{1,2} = a \pm ib$, with $a, b \in \mathbb{R}$. If a nontrivial pair of eigenvalues with modulus larger than unity exists, stable and unstable modes near the periodic orbit are present, while if a nontrivial complex conjugate pair exists with modulus equal to unity, a nearby oscillatory mode exists. The stable and unstable modes are associated with nearby stable and unstable invariant manifolds, while the unitary complex conjugate pair is associated with nearby quasi-periodic trajectories [70, 73].

3.2 Differential Correction

A variety of techniques can be used to numerically recover a trajectory with specific characteristics [73, 75]. In particular, the combination of Netwon's method and multiple shooting is used in this manuscript to retrieve end-to-end trajectories that satisfy a predefined set of constraints.

Multiple shooting is a numerical method used for the solution of boundary value problems over large intervals. With multiple shooting, a defined interval is divided into smaller steps: an initial value problem is solved at each step, and matching conditions are enforced between contiguous steps. To implement multiple shooting, a free variable vector $\boldsymbol{V}_i \in \mathbb{R}^n$, and a constraint vector $\boldsymbol{F}(\boldsymbol{V}_i) \in \mathbb{R}^m$, are defined, with $n, m \in \mathbb{R}$. The free variable vector of a solution to the boundary value problem represents a zero of the constraint vector. However, an initial guess does not usually satisfy $\boldsymbol{F}(\boldsymbol{V}_i) = \boldsymbol{0}$. For this reason, Newton's method can be used to correct the free variable vector $\boldsymbol{V}_i$ at the i-th iteration to obtain an updated estimate of the free variable vector, $\boldsymbol{V}_{i+1}$. In particular, the constraint vector can be expanded via Taylor series in a neighborhood of the first guess solution as:

$$\boldsymbol{F}(\boldsymbol{V}_{i+1}) = \boldsymbol{F}(\boldsymbol{V}_i) + \boldsymbol{DF}(\boldsymbol{V}_i)(\boldsymbol{V}_{i+1} - \boldsymbol{V}_i) \tag{3.8}$$

where the Jacobian $\boldsymbol{DF}(\boldsymbol{V}_i)$ represents the derivative of the constraint vector with respect to the free variable vector, evaluated at the current free variable vector $\boldsymbol{V}_i$. Assuming the new solution satisfies $\boldsymbol{F}(\boldsymbol{V}_{i+1}) = \boldsymbol{0}$, a mathematical formulation for $\boldsymbol{V}_{i+1}$ can be obtained inverting Eq. (3.8) as:

$$\begin{cases} \boldsymbol{V}_{i+1} & = \boldsymbol{V}_i - \left[\boldsymbol{DF}(\boldsymbol{V}_i)\right]^{-1} \boldsymbol{F}(\boldsymbol{V}_i) & \text{if } n = m \\ \boldsymbol{V}_{i+1} & = \boldsymbol{V}_i - \boldsymbol{DF}(\boldsymbol{V}_i)^T \left[\boldsymbol{DF}(\boldsymbol{V}_i)\boldsymbol{DF}(\boldsymbol{V}_i)^T\right]^{-1} \boldsymbol{F}(\boldsymbol{V}_i) & \text{if } n \neq m \end{cases} \tag{3.9}$$

Newton's method provides an option for updating the free variable vector, with demonstrated quadratic convergence property when the initial guess is sufficiently near the zero of $\boldsymbol{F}(\cdot)$ [73]. However, due to the inherent errors in solving a dynamical system using numerical integration, a free variable vector is sufficiently close to a solution when $\|\boldsymbol{F}(\boldsymbol{V}_i)\|_L \leq \delta$, with $\delta \in \mathbb{R}$ a small positive number. The operator $\|\cdot\|_L$ represents the generic L-norm: for relatively small problems the ℓ^2-norm is used, while for the relatively large problems the ℓ^∞-norm is preferred in this investigation. This difference enables relatively large problems to numerically converge with small values of δ.

In a typical multiple shooting approach, the free variable vector is populated as a discrete representation of a trajectory of interest. Consider a periodic orbit $\boldsymbol{x}(t)$, discretized into a sequence of states, each defined as $\boldsymbol{x}(t_i)$. If the states are equally spaced in time, a unique propagation time is included in the free variable vector. All the discretized states are appended, together with the time separating each pair of states, to form the free variable vector as:

$$\boldsymbol{V} = [\boldsymbol{x}(t_0), \boldsymbol{x}(t_1), \ldots, \boldsymbol{x}(t_n), t_{\text{int}}]^T \in \mathbb{R}^{6n+1} \tag{3.10}$$

Then, each state is propagated for a propagation time t_{int} to encounter the subsequent state. When the multiple shooting framework is applied to retrieve a periodic orbit in the CR3BP, continuity is required between neighboring arcs. However, the Jacobi constant is preserved throughout a periodic orbit, forming an implicit constraint in this particular problem formulation. For this reason, a component of the state periodicity constraint can be removed, and a new constraint can be added to aid the convergence process. For example, the initial y-component can be fixed at 0, if the underlying periodic orbit crosses the x-axis. These conditions translate into a constraint vector taking the following form:

$$\boldsymbol{F}(\boldsymbol{V}) = \begin{bmatrix} \boldsymbol{x}(t_0; t_{\text{int}}) - \boldsymbol{x}(t_1) \\ \boldsymbol{x}(t_1; t_{\text{int}}) - \boldsymbol{x}(t_2) \\ \vdots \\ \boldsymbol{x}(t_{n-1}; t_{\text{int}}) - \boldsymbol{x}(t_n) \\ [\boldsymbol{x}(t_n; t_{\text{int}}) - \boldsymbol{x}(t_0)]\big|_{\boldsymbol{s}} \\ y(t_0) \end{bmatrix} \in \mathbb{R}^{6n} \tag{3.11}$$

where $\boldsymbol{x}(t_0; t_{\text{int}})$ represents the final state generated by propagating the initial state $\boldsymbol{x}(t_0)$ for a time t_{int}. The symbol $\cdot|_{\boldsymbol{s}}$ expresses that only the rows in the sequence $\boldsymbol{s} = \{1, 2, 3, 4, 6\}$ are used. To retrieve a periodic orbit in the CR3BP, the system in Eq. (2.5) is used to propagate the spacecraft states. The constraint vector is differentiated to populate the Jacobian matrix as:

$$\boldsymbol{DF}(\boldsymbol{V}_i) = \begin{bmatrix} \dfrac{\partial \boldsymbol{x}(t_0; t_{\text{int}})}{\partial \boldsymbol{x}(t_0)} & -\mathbf{I}_{6\times6} & \mathbf{0}_{6\times6} & \cdots & \mathbf{0}_{6\times6} & \dfrac{\partial \boldsymbol{x}(t_0; t_{\text{int}})}{\partial t_{\text{int}}} \\ \mathbf{0}_{6\times6} & \dfrac{\partial \boldsymbol{x}(t_1; t_{\text{int}})}{\partial \boldsymbol{x}(t_1)} & -\mathbf{I}_{6\times6} & \cdots & \mathbf{0}_{6\times6} & \dfrac{\partial \boldsymbol{x}(t_1; t_{\text{int}})}{\partial t_{\text{int}}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -\mathbf{I}_{6\times6}\big|_{\boldsymbol{s}} & \mathbf{0}_{5\times6} & \mathbf{0}_{5\times6} & \cdots & \dfrac{\partial \boldsymbol{x}(t_n; t_{\text{int}})}{\partial \boldsymbol{x}(t_n)}\bigg|_{\boldsymbol{s}} & \dfrac{\partial \boldsymbol{x}(t_n; t_{\text{int}})}{\partial t_{\text{int}}}\bigg|_{\boldsymbol{s}} \\ [0\ 1\ 0\ 0\ 0\ 0] & \mathbf{0}_{1\times6} & \mathbf{0}_{1\times6} & \cdots & \mathbf{0}_{1\times6} & 0 \end{bmatrix} \in \mathbb{R}^{6n\times6n+1} \tag{3.12}$$

where $\cdot|_{\boldsymbol{s}}$ symbolizes only the rows in the sequence $\boldsymbol{s} = \{1, 2, 3, 4, 6\}$ are preserved. Moreover, the derivative of the final states with respect to the initial ones can be expressed in terms of the STM using Eq. (3.5), while the derivatives of the propagated state with respect to the propagation time (last column of Eq. (3.12)) represent the derivative of the state at the final time of each arc as:

$$\frac{\partial \boldsymbol{x}(t_i; t_{\mathrm{int}})}{\partial \boldsymbol{x}(t_i)} = \boldsymbol{\Phi}(t_i + t_{\mathrm{int}}, t_i), \qquad \frac{\partial \boldsymbol{x}(t_i; t_{\mathrm{int}})}{\partial t_{\mathrm{int}}} = \dot{\boldsymbol{x}}(t_i; t_{\mathrm{int}}) \tag{3.13}$$

The number of arcs n used to discretize the trajectory encapsulates a trade-off between accuracy and computational efficiency. Indeed, relatively large values of n correspond to a larger dimensional problem that requires larger computational resources, although a large system reduces the numerical error intrinsic in the computation of the state transition matrices. In the limit case where $n = 1$, the algorithm reduces to single shooting [52]. Note that if the constraint vector and the Jacobian matrix defined in Eqs. (3.11) and (3.12) are used to retrieve a feasible orbit in the CR3BP, the second row in Eq. (3.9) is used, since the dimension of the free variable vector $\boldsymbol{V}$ and the constraint vector $\boldsymbol{F}(\boldsymbol{V})$ are different. For this reason, the algorithm can potentially compute a unique member of a 1-parameter family of solutions. To construct a specific solution with prescribed characteristics, the dimensions of the free variable vector and the constraint vector can be set as equal by introducing an additional constraint. This strategy is also leveraged in continuation methods to retrieve elements of the same family of solutions, as explained in Sec. 3.3.

3.3 Continuation Methods

After computing a single trajectory via a correction scheme, numerical continuation techniques can be used to retrieve different members from the associated family of solutions. A variety of numerical continuation techniques exist to retrieve elements from a family of solutions. Specifically, a more intuitive method called single-parameter continuation is first introduced in Sec. 3.3.1, followed by the more robust pseudo-arclength approach in Sec. 3.3.2.

3.3.1 Natural-Parameter Continuation

Natural-parameter continuation is often used in trajectory design to retrieve a finite set of members from a family, or set, of solutions. Prior to starting the continuation scheme, an initial solution is computed, using for example a combination of a multiple shooting scheme and Newton's method as presented in Sec. 3.2. To calculate another nearby solution via natural-parameter continuation, a new solution with a generic parameter $p(\boldsymbol{V}) \in \mathbb{R}$, is desired to possess a value $\bar{p}$, captured in the following constraint:

$$p(\boldsymbol{V}) - \bar{p} = 0 \tag{3.14}$$

When calculating a periodic orbit in the CR3BP, common choices for the parameter of interest are represented, but not limited to: coordinates x, y, z, $\dot{x}$, $\dot{y}$, $\dot{z}$; period of the orbit T; Jacobi constant of the orbit C_J; mass ratio μ. The vector $\boldsymbol{F}(\boldsymbol{V})$ is updated to accommodate this additional constraint. The differential correction scheme is then applied, leveraging a previously known solution $\boldsymbol{V}_0$ as an initial guess for a nearby solution. For convergence using Newton's method, $\boldsymbol{V}_0$ must be relatively close to the sought solution: this is generally achieved by using a relatively small perturbation δp. After converging to a second solution, the approach is repeated to generate a family of solutions, usually maintaining the sign of δp as constant throughout the continuation process. Although straightforward to implement and intuitive to understand, the natural-parameter continuation scheme might not be capable to fully extend the family at particular values of the selected parameter p, as for families of solutions that are non-monotonically increasing in p.

3.3.2 Pseudo-Arclength Continuation

Differently from single-parameter continuation, pseudo-arclength steps along a direction given by the tangent to the recovered family of trajectories. The tangent direction to the family is usually not aligned with any quantity of the family associated with a physical meaning [52]. With a previous solution $\boldsymbol{V}_{i-1} \in \mathbb{R}^{6n}$, the tangent to the family is computed by identifying the null space of the

Jacobian matrix evaluated at the previous solution as:

$$\boldsymbol{V}_{i-1}^{\star} = \mathrm{Null}(\boldsymbol{DF}(\boldsymbol{V}_{i-1})) \in \mathbb{R}^{6n} \tag{3.15}$$

with Jacobian matrix defined in Eq. (3.12) as $\boldsymbol{DF}(\boldsymbol{V}_{i-1}) \in \mathbb{R}^{6n \times 6n+1}$. Note that the tangent to the family of solution is parallel to the null space vector $\boldsymbol{V}_{i-1}^{\star}$. Then, a new constraint is included in the constraint vector $\boldsymbol{F}(\boldsymbol{V})$, requiring that the difference between the new solution $\boldsymbol{V}_i$ and $\boldsymbol{V}_{i-1}$ when projected onto the tangent direction equals a specified value, $\Delta s \in \mathbb{R}$. The new constraint takes the form:

$$(\boldsymbol{V}_i - \boldsymbol{V}_{i-1})^T \boldsymbol{V}_{i-1}^{\star} - \Delta s = 0 \tag{3.16}$$

where $(\boldsymbol{V}_i - \boldsymbol{V}_{i-1})$ represents the step taken in the solution space. The dot product between the step and the null space $\boldsymbol{V}_{i-1}^{\star}$ represents the component of the step in the solution space along the direction of the null space. With equal dimensions of $\boldsymbol{V}$ and $\boldsymbol{F}(\boldsymbol{V})$, Newton's method can be leveraged to iteratively compute a single solution that satisfies Eq. (3.12). In this investigation, pseudo-arclength is employed to generate families of periodic orbits in the CR3BP. Members of L_1 Lyapunov and L_2 halo families in the Earth-Moon CR3BP are depicted in Fig. 3.1. Pseudo-arclength continuation is also applied to compute members of families of quasi-periodic trajectories.

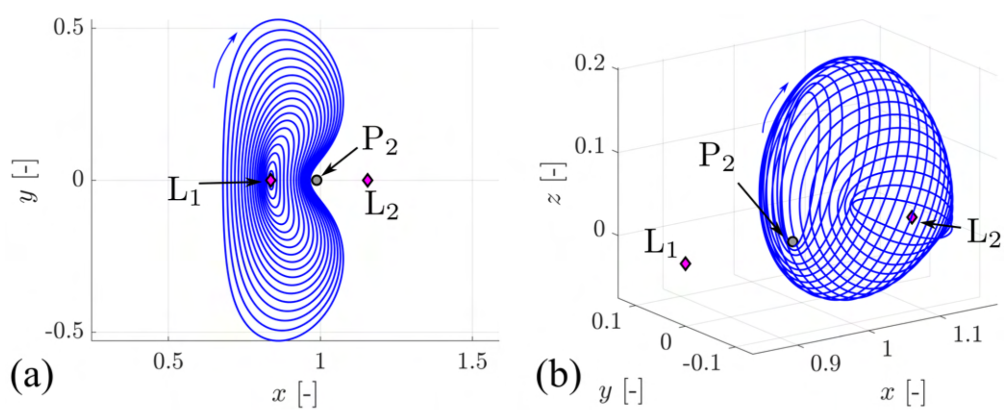

Figure 3.1: Example of periodic orbits in the families of (a) planar Lyapunov orbits near L_1 and (b) northern halo orbits near L_2, in the Earth-Moon CR3BP.

3.4 Periodic Orbits Near Orbital Resonances

Orbits near resonance represent a particular type of periodic orbits that is often leveraged in mission design in a variety of planetary systems [20–22, 76]. In particular, this investigation focuses on mean-motion or orbital resonance, shortly addressed as resonance throughout the presented investigation. The definition of a mean-motion orbital resonance is derived from the two-body dynamics. According to this definition, two massless bodies, B and C, are influenced in their motion only by the gravitational influence of a point-mass central body, A. Bodies B and C are in resonance if B completes exactly p orbits about A in the same time that C completes q revolutions about A, with $p, q \in \mathbb{N}^+$ [3, 24]. With the definition of mean motion $\tilde{n} = 2\pi/\tilde{T}$ from the two-body dynamical model, the resonance condition translates in a ratio between the periods of bodies B and C in their motion around A as:

$$\frac{p}{q} = \frac{\tilde{n}_p}{\tilde{n}_q} = \frac{2\pi/\tilde{T}_p}{2\pi/\tilde{T}_q} = \frac{\tilde{T}_q}{\tilde{T}_p} \tag{3.17}$$

where the $p : q$ resonance is classified as interior when $p > q$, and exterior when $p < q$. Often, body B can be modeled as a spacecraft, while body C can represent another body of the considered system: for example, for resonant orbits about the Earth, the Moon can represent body C.

In this investigation, the method highlighted by Vaquero is used to compute a resonant orbit [24]. According to this method, the spacecraft is assumed to be initialized at the periapsis or apoapsis of a planar resonant orbit around body A. The generic inertial state of the spacecraft with respect to body A is $\tilde{\boldsymbol{X}} = [\tilde{X}_0,\ \tilde{Y}_0,\ 0,\ \dot{\tilde{X}}_0,\ \dot{\tilde{Y}}_0,\ 0]$. Moreover, the initial state of the spacecraft is assumed to lie along the x-axis. Together with the constraint of the apsis initial location, the initial state of body B with respect to A is formulated as $\tilde{\boldsymbol{X}} = [\tilde{X}_0,\ 0,\ 0,\ 0,\ \dot{\tilde{Y}}_0,\ 0]$. To retrieve $\tilde{X}_0$ and $\dot{\tilde{y}}_0$, a set of input is identifies, comprising: the resonance integers, p and q; the mass parameter of body A, $\tilde{\mu}_A = \tilde{G}\tilde{M}_A$; the semi-major axis of body C in its motion around A, $\tilde{a}_C$; the eccentricity of the orbit of the spacecraft around body A, e_B; the initial true anomaly of the spacecraft in its orbit around body A, $\theta_B = k\pi$ with $k \in \mathbb{N}$. This set of input allows to retrieve the periods of the orbits followed by body C and the spacecraft around body A as:

$$\tilde{T}_q = 2\pi\sqrt{\frac{\tilde{a}_C^3}{\tilde{\mu}_A}}, \qquad \tilde{T}_p = \tilde{T}_q \frac{q}{p} \tag{3.18}$$

The period of body B around A, together with the input eccentricity and the initial true anomaly, are used to generate the inertial distance and velocity. Indeed, the semi-major axis of the orbit followed by the spacecraft is:

$$\tilde{a}_B = \sqrt[3]{\left(\tilde{\mu}_A \frac{\tilde{T}_p}{2\pi}\right)^2} \tag{3.19}$$

and the inertial position and velocity, corresponding to $\tilde{X}_0$ and $\dot{\tilde{Y}}_0$, are

$$\begin{aligned} \tilde{X}_0 &= \frac{\tilde{a}_B(1-e_B^2)}{1+e_B\cos(\theta_B)} \\ \dot{\tilde{Y}}_0 &= \sqrt{2\tilde{\mu}_A\left(\frac{1}{\tilde{X}_0}-\frac{1}{2\tilde{a}_B}\right)} \end{aligned} \tag{3.20}$$

To transition a resonant orbit retrieved in the two-body problem into a T-periodic orbit in the CR3BP, bodies A and C are assigned to P_1 and P_2, respectively. However, body C has a non-negligible mass in the CR3BP formulation, differently from the assumption of the two-body problem. For this reason, the ratio between the periods of the spacecraft and P_2 in their motion around P_1 in the CR3BP is only approximately equal to the $p:q$ ratio. Moreover, the resonant orbit computed in the two-body dynamics is not anymore a periodic orbit when transitioned to the higher fidelity CR3BP dynamical model. However, the original resonant orbit retrieved with the assumption of the two-body problem can be used as a first guess to obtain an orbit near resonance in the associated CR3BP. The nondimensional first guess, obtained in the inertial P_1-centered frame, is initially rotated in the P_1-P_2 rotating frame, centered at the P_1-P_2 barycenter. After rotation, the state produces an initial guess for a periodic orbit in the rotating frame.

A variety of approaches can be leveraged to obtain an orbit near resonance in the CR3BP, starting from a solution obtained in the two-body problem [24]. For this investigation, the continuous first guess trajectory is initially split in different nodes, using a multiple shooting technique to numerically correct for periodicity. However, converging to a periodic orbit in the CR3BP with similar geometrical features often represents a numerically challenging task. Therefore, if the first attempt to obtain the orbit near resonance in the analyzed system fails, a sequence of two continuation schemes is adopted to construct the orbit in the CR3BP. These two continuation schemes can be described as:

1) Continuation in the eccentricity e_B: although the required eccentricity for the spacecraft orbit in the two-body problem assumption is specified, a close approach to one of the primaries can significantly challenge the converge process. Therefore, the eccentricity is varied by defining the pericenter radius of the orbit followed by the spacecraft in its orbit about body A as:

$$\tilde{r}_p = \frac{\tilde{a}_C}{k_i} \qquad e_B = 1 - \frac{\tilde{r}_p}{\tilde{a}_B} \tag{3.21}$$

with $k_i = \{1.1, 1.2, \dots\}$ progressively increasing until computing a periodic orbit. This continuation scheme modifies the shape of the orbit in the two-body problem framework, leading to a different first guess used by the multiple shooting scheme.

2) Continuation in the mass ratio μ: the resonant orbit retrieved with the two-body problem assumption with the original eccentricity e_B is utilized as an initial solution to retrieve members of a family of orbits. In particular, natural-parameter continuation scheme is used, and the mass ratio μ is varied from 0 till the prescribed mass ratio of the analyzed CR3BP system.

This approach is leveraged to retrieve a variety of different orbits near selected resonances in the CR3BP. Figure 3.2 presents a tabular visual overview of the different geometrical features of 16 members from different families of planar orbits near resonance in the Earth-Moon CR3BP. Each row and column are associated with a specific value of p and q, respectively. In each frame, the Earth and the Moon are reported with gray circles, the Lagrangian points as magenta diamonds, and the periodic orbits in blue. When an orbit near resonance is retrieved in the analyzed CR3BP system, pseudo-arclength continuation is used to generate additional members of the same family of resonant orbits [77–79].

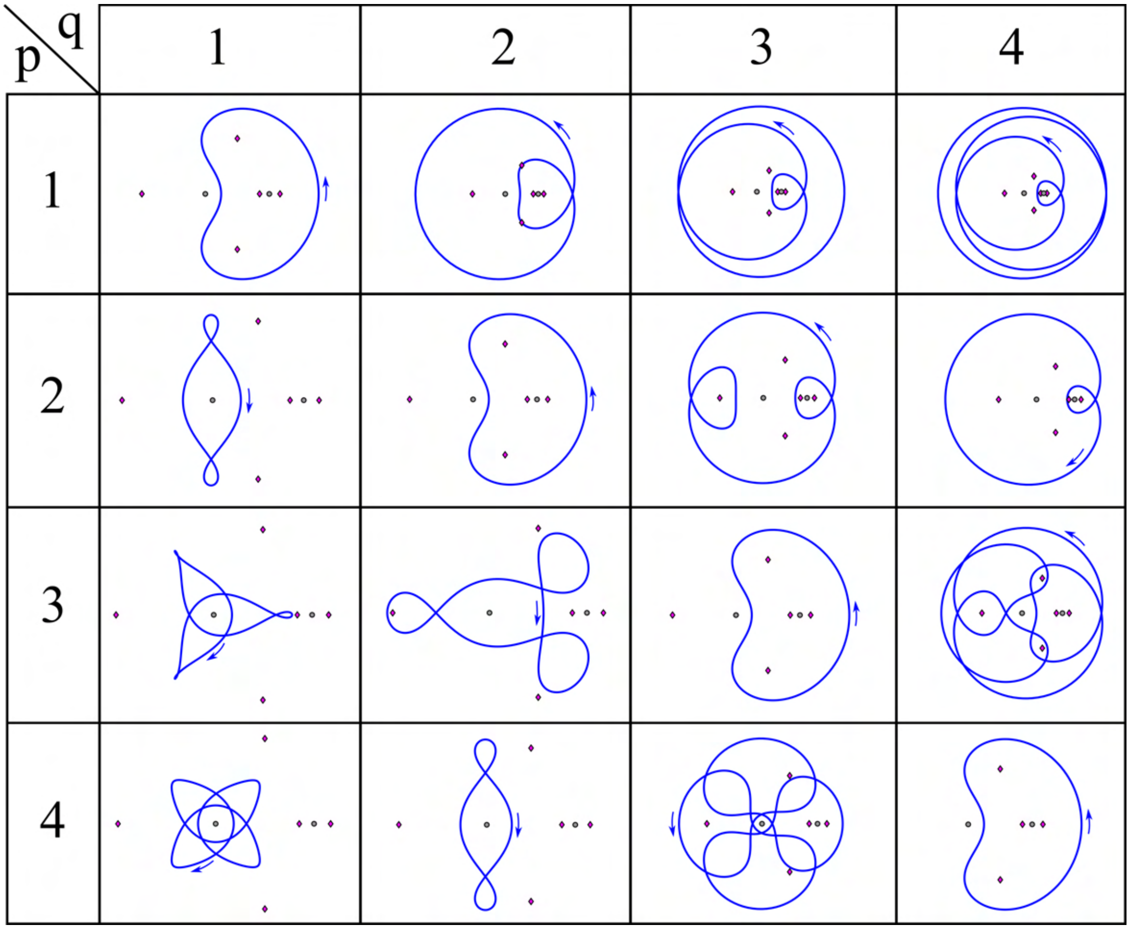

Figure 3.2: Representative members of 16 families of orbits near resonance in the Earth-Moon CR3BP. Rows and columns are associated with the p and q index, respectively.

3.5 Quasi-Periodic Trajectories

Quasi-periodic trajectories are bounded solutions that can exist in low-fidelity dynamical models nearby periodic orbits. In their bounded motion near a periodic orbit, quasi-periodic trajectories foliate the surface of invariant structures called n-tori, with $n \in \mathbb{N}^+$. The invariant tori are defined by n incommensurate frequencies. In particular, this investigation focuses on planar and spatial invariant tori governed by two fundamental frequencies: the 2-tori. Therefore, a point on a 2-torus is defined by two angular quantities $(\theta_1(t), \theta_2(t))$, associated with the longitudinal and transverse directions, with constant frequencies $\omega_1 = \dot{\theta}_1$, $\omega_2 = \dot{\theta}_2$. Moreover, quasi-periodic trajectories are only retrieved in the low-fidelity CR3BP throughout this investigation, although they can be obtained also in higher-fidelity dynamical models [80, 81]. Quasi-periodic trajectories exist nearby a periodic solution with a monodromy matrix possessing a pair of complex conjugate eigenvalues with unitary modulus.

Different strategies exist to compute an approximation of an invariant torus [26]. Specifically, the implementation developed by Jorba [27] and Gomez and Mondelo [28], later refined by Olikara and Scheeres [29], is used in this investigation. This method can generate accurate approximations of invariant tori with relatively small computational effort [26]. This strategy approximates an invariant torus at multiple locations by a discrete number of states along its invariant curves $v(\cdot)$. An invariant curve represents an equilibrium solution of the mapping under the flow governed by the dynamics of the CR3BP. Specifically, when the invariant curve is propagated by the dynamical model for a stroboscopic mapping time $T_S = 2\pi/\omega_1$, the same invariant curve is retrieved. However, each point along the invariant curve experiences a rotation: a state $\boldsymbol{x}(\theta_1, \theta_2)$ that initially lies on the invariant curve experiences a rotation along the curve by an angle $\rho = 2\pi\omega_2/\omega_1$ after being propagated for the stroboscopic mapping time. The invariance condition for the single point is mathematically written as:

$$R_{-\rho} v\left(\boldsymbol{x}(\theta_1, \theta_2)\right) - \boldsymbol{x}(\theta_1, \theta_2) = \boldsymbol{0} \in \mathbb{R}^6 \tag{3.22}$$

where $R_{-\rho}$ is a rotational operator. The invariant condition in Eq. (3.22) can be extended to a set of N_Q points along the same invariant curve, improving the torus approximation. The points used to approximate the same invariant curve are initialized with a constant relative angular distance $\delta\theta_2$ along the curve, and are approximated via a truncated Fourier series. As a result, Eq. (3.22) is extended to the set of N_Q points: the operator $R_{-\rho}$ is transformed into a combined sequence of matrices, $\boldsymbol{R}(-\rho)$, and the points approximating the invariant curve are aggregated into a matrix $\boldsymbol{U} \in \mathbb{R}^{N_Q \times 6}$. The numerical equivalent of the invariance condition for the set of N_Q points is expressed as:

$$\boldsymbol{S} = \mathrm{vec}(\boldsymbol{R}(-\rho) v\left(\boldsymbol{U}(\theta_1, \theta_2)\right) - \boldsymbol{U}(\theta_1, \theta_2)) = \boldsymbol{0} \in \mathbb{R}^{6N_Q} \tag{3.23}$$

where the condition is vectorized by the vec$(\cdot)$ operator.

To retrieve a 2-torus existing near a periodic orbit with a pair of complex conjugate eigenvalues with unitary modulus, a location $\boldsymbol{x}_0$ along the periodic orbit is defined, corresponding to a longitudinal angle $\theta_1 = 0$. The eigenvector $\boldsymbol{v}_C \in \mathbb{C}^6$ that is associated with the complex unitary eigenvalue $\lambda_C \in \mathbb{C}$ of the monodromy matrix starting at $\boldsymbol{x}_0$ is leveraged to generate an initial guess of the invariant curve near $\boldsymbol{x}_0$. Indeed, $\boldsymbol{v}_C$ defines N_Q points along an initial guess for the invariant curve as:

$$\boldsymbol{x}(\theta_1, \theta_{2,i}) = \boldsymbol{x}_0 + \epsilon(\text{Re}[\boldsymbol{v}_C]\cos(\theta_{2,i}) + \text{Im}[\boldsymbol{v}_C]\sin(\theta_{2,i})) \tag{3.24}$$

where $\epsilon \in \mathbb{R}$ is a small value, and an odd number of equally spaced values of the transverse angle θ_2 are used. Moreover, the period of the underlying periodic orbit is used as an approximation for the stroboscopic mapping time T_S, while the rotational angle is approximated as $\rho = \text{Re}[-i \ln \lambda_C]$ [75]. An initial guess is therefore constructed by appending: the approximated states along the invariant curve $\boldsymbol{x}(\theta_1, \theta_{2,i})$; the stroboscopic time T_S; and the rotational angle ρ. This initial guess can be corrected employing a differential correction scheme [30, 75]. When applying multiple shooting to correct for a invariant torus near a periodic orbit, a number of M_Q equally-spaced locations along θ_1 can be identified. From each of these locations, an approximated invariant curve is computed, leveraging Eq. (3.24). The states approximating the invariant curves at multiple instances along the torus are appended, together with the rotational angle ρ and a fraction of the stroboscopic mapping time T_S/M_Q, forming a free variable vector that is corrected for trajectory continuity and to satisfy the invariance condition [30, 75, 82].

A variety of constraints can also been considered to retrieve a torus with particular physical or geometrical characteristics. Indeed, with a set of N_Q points along each invariant curve, and a set of M_Q invariant curves along the original orbit, the free variable vector has dimension $\boldsymbol{X} \in \mathbb{R}^{6N_Q M_Q + 2}$. However, the constraint vector leveraged in this investigation uniquely solves for continuity and invariance condition, therefore representing $\boldsymbol{F}(\boldsymbol{X}) \in \mathbb{R}^{6M_Q N_Q}$. For this reason, two more conditions can be defined to retrieve a unique torus. Candidate adjunct conditions comprise: defining a fixed stroboscopic mapping time T_S; identifying a required rotational angle ρ; identifying an average fixed level of Jacobi constant for one invariant curve C_J. When the generated first guess converges to a torus, the latter can be used as a first solution to retrieve other members of the family. Indeed, thanks to the nature of the CR3BP and the dimensionality of the problem, 2-tori appear in 2-parameter families. To populate a family with different members, methodologies like pseudo-arclength can be leveraged [75]. This approach is applied throughout this investigation to retrieve elements of the families of QPTs near L_1 and L_2 in the Sun-Earth system, as depicted in Fig. 3.3(a-b), respectively, and to obtain members of QPTs foliating tori near resonances in the Earth-Moon system, as visualized in Fig. 3.4.

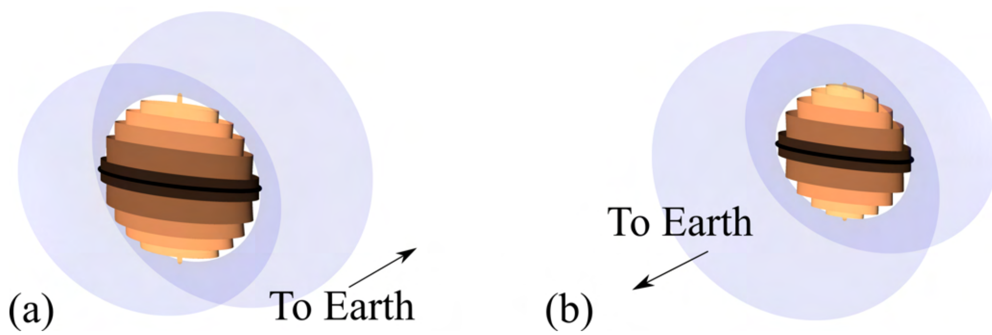

Figure 3.3: Representative members of families of QPTs near (a) L_1 and (b) L_2 at $C_J = 3.00088$ in the Sun-Earth CR3BP.

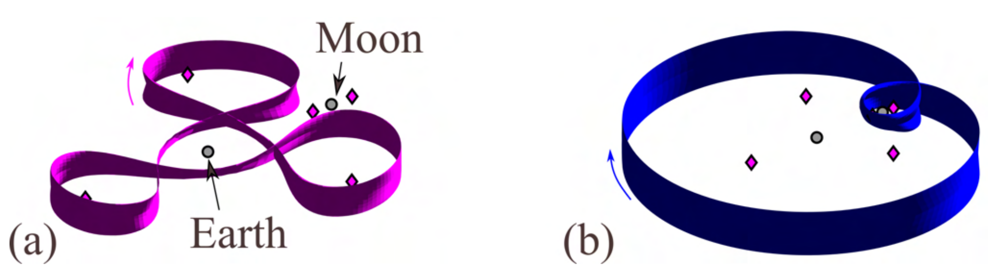

Figure 3.4: Representative of families of QPTs near the (a) 3:2 and (b) 1:2 resonances in the Earth-Moon system at $C_J = 2.73$.

Information concerning the dynamical flow near a 2-torus represents an important asset for mission design. To generate a 2-torus in the CR3BP, the invariance condition in Eq. (3.23) is leveraged to generate an approximation of the invariant curve. Then, the differential of the invariance condition $\boldsymbol{DS} \in \mathbb{R}^{6N_Q \times 6N_Q}$ is used to infer the stability characteristics of the analyzed torus. The differential is expressed as:

$$\boldsymbol{DS} = \left(\boldsymbol{R}(-\rho) \otimes \boldsymbol{I}_{6N_Q \times 6N_Q}\right) \hat{\boldsymbol{\Phi}}(t_0 + T_S, t_0) \;\in \mathbb{R}^{6N_Q \times 6N_Q} \tag{3.25}$$

where $\otimes$ is the Kronecker product operator, $\boldsymbol{I}_{6N_Q \times 6N_Q}$ is the identity matrix, and $\hat{\boldsymbol{\Phi}}(t_0 + T_S, t_0) = \text{diag}(\boldsymbol{\Phi}_1(t_0 + T_S, t_0), \boldsymbol{\Phi}_2(t_0 + T_S, t_0), \ldots, \boldsymbol{\Phi}_{N_Q}(t_0 + T_S, t_0))$ is a block diagonal STM of the N_Q points approximating the initial invariant curve [82]. However, the approximation of the converged invariant curve relies on the obtained transverse angles sequence $\bar{\theta}_2$, separating each point along the same invariant curve. The converged sequence significantly depends on the initial guess. Therefore, a different initial placement of the states approximating the invariant curve can likely generate a different converged solution. Likewise, a rotation by a fixed angle along the longitudinal direction of each state of the converged invariant curve can still satisfy Eq. (3.23). These different solutions might lead to distinct differentials of the invariance condition $\boldsymbol{DS}$, and therefore distinct spectral decomposition of $\boldsymbol{DS}$. However, Jorba demonstrates that tori in the CR3BP are reducible, therefore the differential of the invariance condition does not depend on the sequence of transverse angles $\bar{\theta}_2$ used to approximate the invariant curve [27, 75].

Since the 2-tori in the CR3BP are reducible, the eigenvalues of the differential of the invariance condition are placed in concentric circles on the Gauss plane [27, 75]. Each of these circles is associated with a radius R_i equal to the modulus of a specific eigenvalue [82]. Thanks to the symplectic nature of the CR3BP, and analogously to the periodic orbit scenario, pairs of circles with reciprocal radii exist: if a circle with radius R_i exists in the Gauss plane reporting the eigenvalues of the $\boldsymbol{DS}$, then another circle with radius $1/R_i$ is present on the same plane. Also, similarly to the periodic orbit case, the trivial circle with radius $R_i = 1$ is always present in the Gauss plane. Ultimately, if a collection of eigenvalues of DS is located along a radius with $R_i > 1$, the investigated invariant torus is unstable [27, 82].

3.6 Hyperbolic Invariant Manifolds

When the investigated periodic or quasi-periodic solution is unstable, structures called hyperbolic invariant manifolds exist, that are populated by trajectories that flow into or depart from the originating solution. When the solution under investigation is unstable, both a stable and an unstable invariant manifold exist. The stable (unstable) manifold represents a set of solutions $\boldsymbol{x}(t)$ in the available space that approach (departs) the originating solution as $t \to +\infty$.

To produce a subset of trajectories lying on the stable or unstable manifold of a periodic orbit, the spectral decomposition of the monodromy matrix is leveraged. In particular, to generate a trajectory within the stable manifold that approaches a location $\boldsymbol{x}_{PO}$ along the investigated periodic orbit, the stable eigenvector $\boldsymbol{\nu}_s$, associated with a stable eigenvalue $|\lambda_s| < 1$ of the monodromy matrix, is selected. With a small displacement $\epsilon > 0$, the initial state of the trajectory along the stable manifold is retrieved as:

$$\boldsymbol{x} = \boldsymbol{x}_{PO} \pm \epsilon \boldsymbol{\nu}_s \tag{3.26}$$

where the $\pm$ defines the direction of the manifold arc. To generate an arc $\boldsymbol{x}(t)$ that approaches the investigated periodic orbit for $t \to +\infty$, the initial state $\boldsymbol{x}$ is propagated backward in time. Similarly, to retrieve a trajectory lying on the unstable manifold of the investigated orbit, the unstable eigenvector $\boldsymbol{\nu}_u$, associated with an unstable eigenvalue $|\lambda_u| > 1$ of the monodromy matrix, is used. A small number $\epsilon > 0$ is then employed to perturb the state along the periodic orbit in the direction of $\boldsymbol{\nu}_u$. The perturbed state is then propagated forward in time to generate a trajectory arc that approaches the periodic orbit for $t \to -\infty$ [3,70]. A subset of the trajectories lying on the hyperbolic stable (unstable) manifold is obtained by perturbing multiple locations of the same periodic orbit along the direction of the local stable (unstable) eigenvalue, and propagating backward (forward) in time. The generated trajectories lying on numerically-generated approximation of the hyperbolic stable or unstable manifold likely have a slightly different Jacobi constant with respect to the originating orbit. The different energy level is due to the perturbation imparted to the state along the orbit in the direction of the stable or unstable eigenvalue.

Quasi-periodic trajectories foliating the surface of invariant 2-tori in the CR3BP can also present nearby hyperbolic stable and unstable manifolds. Indeed an invariant 2-torus in the CR3BP is unstable if the differential of the invariance condition $\boldsymbol{DS}$ presents a circle of eigenvalues with $R_i > 1$ in the Gauss plane. Similarly to the periodic orbit case, a subset of trajectories lying on the stable manifold can be computed by leveraging the spectral decomposition of the local $\boldsymbol{DS}$. A set of states that approximates an invariant curve along the investigated torus is defined as:

$$\boldsymbol{X}_{QPT} = \text{vec}(\boldsymbol{U}(\theta_1, \theta_2)) \in \mathbb{R}^{6N_Q} \tag{3.27}$$

At the same location, the differential of the invariance condition $\boldsymbol{DS}$ is computed. To compute arcs lying on the stable manifold on the set of points $\boldsymbol{X}_{QPT}$, the eigenvector $\boldsymbol{N}_s$ associated with the real eigenvalue lying on the circle with $R_s < 1$ is selected. Then, a small perturbation ϵ is applied on the states included into $\boldsymbol{X}_{QPT}$ along the direction of $\boldsymbol{N}_s$, obtaining [75]:

$$\boldsymbol{X} = \boldsymbol{X}_{QPT} \pm \epsilon \boldsymbol{N}_s \in \mathbb{R}^{6N_Q} \tag{3.28}$$

Ultimately, arcs lying on the stable manifold of the 2-tori can be retrieved by propagating $\boldsymbol{X}$ backward in time. Similarly, the eigenvector $\boldsymbol{N}_u$ associated with the real eigenvalue of the local $\boldsymbol{DS}$ which lies on the circle with $R_u = 1/R_s > 1$ is used to obtain initial states for a set of trajectories lying on the unstable manifolds and originating near $\boldsymbol{X}_{QPT}$. These states are then propagated forward in time to generate arcs lying on the local unstable manifolds. The procedure is repeated at different locations along the investigated 2-torus to obtain a set of trajectories lying on the stable or unstable hyperbolic manifold. Figure 3.5 provides a visual example of stable and unstable manifolds for a quasi-halo in the Earth-Moon system.

Stable and unstable manifolds of periodic orbits and invariant tori are often used in trajectory design to generate arcs that naturally approach or depart from a solution of interest. Moreover, when arcs from the stable and unstable manifolds of different periodic or quasi-periodic solutions intersect in the solution space, a heteroclinic transfer exists [83]. Heteroclinic connections naturally transfer a spacecraft between the investigated solutions in infinite time. Perfect heteroclinic connections between two solutions cannot be generated in the CR3BP due to the required infinite time

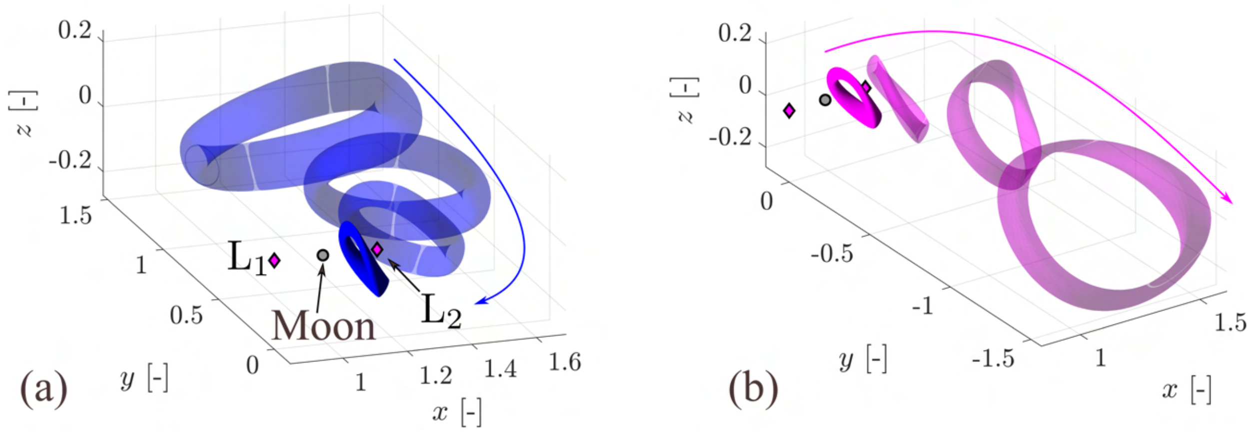

Figure 3.5: Examples of trajectories from the (a) stable and (b) unstable manifolds of a quasi-halo at $C_J = 3.1$ in the Earth-Moon system.

to construct a stable and unstable manifold arc. However, approximations of heteroclinic transfers that naturally connect periodic orbits have been studied in a variety of systems [3, 24]. The study of natural connections between two invariant tori represents a challenging task, due to the higher dimensionality of the solution. Nevertheless, initial efforts from Olikara and Scheeres have already shed lights into feasible natural connections between invariant tori, in specific scenarios [30].

3.7 Poincaré Map

Poincaré maps are a tool from dynamical system theory, often leveraged at different phases of the trajectory design process to investigate the available solution space [3, 38]. A Poincaré map is constructed using two elements: a dynamical flow and a surface of section. A general nonlinear dynamical system models the dynamics governing the spacecraft motion as:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) \tag{3.29}$$

with the spacecraft state $\boldsymbol{x} \in \mathbb{R}^6$. From a general initial condition $\boldsymbol{x}_0 = [x_0, y_0, z_0, \dot{x}_0, \dot{y}_0, \dot{z}_0]$ at time t_0, the spacecraft state can be integrated to a time t using the system of differential equations in Eq. (3.29), generating a new solution $\boldsymbol{x}(t; t-t_0)$, or equivalently a flow $\phi_t(\boldsymbol{x}_0)$ [83]. The propagation is forward in time if $t > t_0$, or backward in time when $t < t_0$. Then, a surface of section Σ is defined to be everywhere transverse to the flow [83]. Assuming $\boldsymbol{x}_0 \in \Sigma$, the first intersection of the flow $\phi_t(\boldsymbol{x}_0)$ with the same surface Σ represents the Poincaré mapping $P(\boldsymbol{x}_0)$. Subsequent intersections of the mapping with the same surface Σ reflect recursive applications of the mapping operator. For example, the third intersection of the flow with the surface of section Σ is $P(P(P(\boldsymbol{x}_0)))$, shortly addressed in this investigation as $P^3(\boldsymbol{x}_0)$. Moreover, a two-sided Poincaré map records the intersections of the dynamical flow in both the positive and negative direction, while a one-sided map only in one predefined direction. A conceptual representation of the Poincaré mapping is represented in Fig. 3.6. Often, the term Poincaré map is used to refer to the set of intersections of the investigated dynamical flow with the considered surface of section. Practically, the surface of section Σ is chosen to possess a lower dimension with respect to the available solution space: therefore, the discrete mapping generates a lower dimensional representation of the investigated flow, by sampling the flow at specific locations. The reduction of dimensionality of the investigated flow performed by the Poincaré mapping preserves the dynamical features of the original flow with an appropriate definition of the associated hyperplane [83]. For this reason, Poincaré maps can be used to reduce the dimensionality of a dynamical flow, simplifying the visualization process, and enabling a simpler analysis of the dynamical characteristics of the analyzed flow.

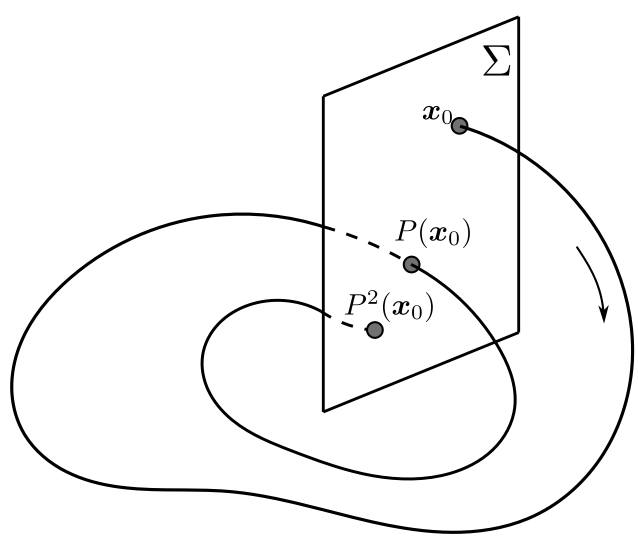

Figure 3.6: Conceptual visualization of a Poincaré mapping.

A surface of section Σ is usually defined through a mathematical formulation, used by the trajectory designer for a desired application. For example, a hyperplane can be initialized as a surface of section by fixing a coordinate in the available space. A hyperplane defined by $y = 0$ is often leveraged in the planar CR3BP to investigate the solution space near periodic orbits intersecting the x-axis [73]. Other representations of the hyperplane comprise: stroboscopic mapping, where the flow is sampled at locations that are equally-separated in time; apses; occurrence of a certain event, represented by a mathematical formulation $g(\boldsymbol{x})$ [3, 6, 84]. In particular, this work often leverages surface of sections defined by the apse condition. In the CR3BP, an apse is defined by the orthogonality condition between the position and velocity vector with respect to one of the primaries. In the rotating frame, the apse condition is mathematically formulated as:

$$(x - x_{\mathrm{P}_i})\dot{x} + y\dot{y} + z\dot{z} = 0 \tag{3.30}$$

where $x_{\mathrm{P}_1} = \mu$ and $x_{\mathrm{P}_2} = 1 - \mu$, representing the location of the primaries along the x-axis in the rotating frame. The sign of the time derivative of the apse condition, presented in Eq. (3.30), is used to classify the apse as a periapsis or an apopsis. Mathematically, the definition of an apse as a periapsis or an apoapsis is formulated as:

$$\begin{cases} \dot{x}^2 + \dot{y}^2 + \dot{z}^2 + (x - x_{P_i})\ddot{x} + y\ddot{y} + z\ddot{z} \geq 0 & \text{if periapsis} \\ \dot{x}^2 + \dot{y}^2 + \dot{z}^2 + (x - x_{P_i})\ddot{x} + y\ddot{y} + z\ddot{z} \leq 0 & \text{if apoapsis} \end{cases} \tag{3.31}$$

Periapsis and apoapsis Poincaré maps are often leveraged for informed trajectory design in a variety of applications [3, 84]. However, investigating a periapsis map can represent a challenging task, especially when the available space becomes higher-dimensional, as for three-dimensional motion.

To present an overview of a Poincaré map generation process, and the challenge represented by visualizing a higher-dimensional Poincaré map, two periapsis maps are constructed in the Sun-Earth CR3BP. A first periapsis map is generated by seeding trajectories in a neighborhood of the Earth at $C_J = 3.00088$. The set of trajectories is constructed from a grid of initial states initialized on a periapsis map in the planar Sun-Earth CR3BP [13, 85]. Therefore, an initial state takes the general form $\boldsymbol{x} = [x_0, y_0, 0, \dot{x}_0, \dot{y}_0, 0]$. The position coordinates (x, y) are selected to lie within the zero velocity curve near the Earth, and in the region delimited by the L_1 and L_2 gateways. In this region, the initial conditions are generated from a grid of 200 equally-spaced x-coordinates between L_1 and L_2, and 200 equally-spaced y-coordinates in the range $[-0.01, 0.01]$. For each (x, y) combination, the apse condition in Eq. (3.30) is used to retrieve the magnitude of the velocity as:

$$v = \sqrt{2U - C_J} \tag{3.32}$$

For each (x, y) that generates a real value number for v, a velocity unit vector is initialized to satisfy the periapsis condition in Eqs. (3.30) and (3.31). The velocity unit vector is constructed to produce an initial prograde periapsis, generating an angular momentum vector with respect to the Earth with a positive component along the z-axis. Each initial perigee is then propagated for a maximum of 20 successive intersections with the periapsis map. Moreover, a trajectory is prematurely terminated if it impacts with the Earth or it escapes from the L_1 or L_2 gateways. The set of the intersections of the analyzed trajectories with the common periapsis hyperplane populates a Poincaré map, projected in the configuration space in Fig. 3.7(a). In this map, each crossing is reported with a black dot, the Earth as a gray central circle, and the ZVC as a blue area. In this representation, each perigee is associated with a unique planar trajectory since the energy level is constrained. Moreover, patterns on the map representation allow to identify dynamical features of the investigated set of trajectories.

To demonstrate the inherent challenge of analyzing a higher-dimensional map, a periapsis map is constructed in the spatial Sun-Earth CR3BP at $C_J = 3.00088$ [85]. This map is constructed with a similar procedure as for the planar map represented in Fig. 3.7(a). However, the grid of initial location is expanded along the z-axis to generate initial perigees in the six-dimensional space. Therefore, an initial state takes the general form $\boldsymbol{x} = [x_0, y_0, z_0, \dot{x}_0, \dot{y}_0, \dot{z}_0]$ in this example. In particular the position coordinates are selected from a grid of: 50 equally-spaced x-coordinates between L_1 and L_2; 50 equally-spaced y-coordinates in the range $[-0.01, 0.01]$; 50 equally-spaced z-coordinates in the range $[-0.01, 0.01]$. For each (x, y, z) combination, the magnitude of the velocity vector is generated leveraging Eq. (3.32). Solving for the velocity unit-vector, satisfying the periapsis condition in Eqs. (3.30) and (3.31), requires enforcing a fourth constraint. For this example, 50 equally-spaced values of $\dot{z}$ are considered in the range $[-v, v]$. Then, each initial state generating a prograde periapsis is propagated forward in time for a maximum of 5 intersections with the periapsis surface of section. Impact with the Earth and natural escape through the L_1 and L_2 gateways are also considered as termination criteria. The recorded intersections with the hyperplane are projected in the configuration space and visualized with black markers in Fig. 3.7(b). In this figure, the boundary of the ZVS is represented with a semi-transparent blue surface. Differently from the planar map in Fig. 3.7(a), the three-dimensional visualization does not supply a bijective representation of the analyzed set of trajectories, due to the higher-dimensionality of the considered scenario. Moreover, the three-dimensional representation does not allow to identify patterns in the map representation due to data obscuration.

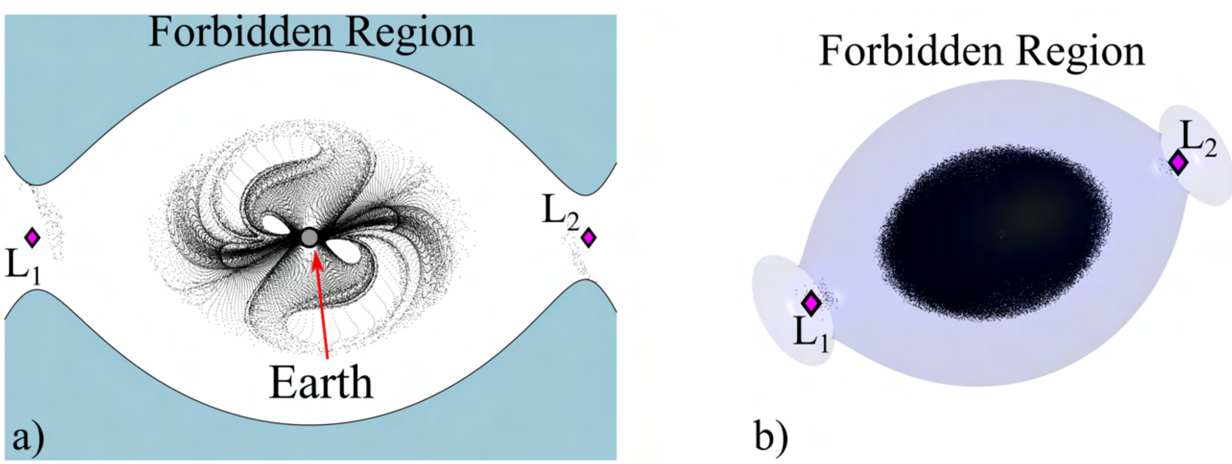

Figure 3.7: Example Poincaré maps reporting prograde perigees in the Sun-Earth system at $C_J = 3.00088$ for (a) planar and (b) spatial trajectories.

3.8 Recovering Trajectories in Higher Fidelity Models

Paths generated in low-fidelity dynamical models, such as the CR3BP, often represent valid approximations of trajectories retrieved in higher-fidelity models, offering a rapid and reliable solution for investigating the solution space during the early stages of the trajectory design process. After identification of arcs satisfying mission-specific requirements, trajectories generated in the low-fidelity CR3BP are often transformed in end-to-end trajectories propagated in higher-fidelity models. This transformation enables to incorporate secondary perturbations within the dynamical model, as well as epoch-depending events in the trajectory path, enhancing the fidelity of the final generated solution [86].

In this investigation, the transformation process is designed to convert a trajectory generated in the CR3BP, and represented in the nondimensional rotating frame centered at the P_1-P_2 barycenter, into a path propagated in a perturbed point mass ephemeris model expressed in an inertial frame and centered at P_1. A similar algorithm can be applied to generate a final trajectory centered at P_2. The transformation process is divided into two major parts, forming a nested structure: an internal multiple-shooting and an external optimization scheme. The optimization is leveraged in this investigation to enforce desired features on a previously converged trajectory. For the multiple-shooting, the trajectory in the CR3BP is converted into a first guess solution for the leveraged point mass ephemeris model. The procedure can be structured according to the following phases, following the approach presented in Sec. 2.4.2 [86]:

1) Trajectory discretization: the path is discretized into multiple arcs, each corresponding to a specified initial epoch. In this implementation, only the initial epoch of the original path is selected by the user. The epochs at the beginning of the subsequent arcs are computed leveraging the temporal separation between the different states: for example, the epoch at the beginning of the second arc is equivalent to the initial epoch of the trajectory, augmented by the propagation time of the first arc.

2) Translation to P_1: the states at the beginning of each arc are translated into the P_1-centered rotating frame by augmenting the x-coordinate by the system-depending parameter μ.

3) State dimensionalization: the states at the beginning of each arc are dimensionalized by multiplying the instantaneous characteristic length and time. Since the instantaneous char-

acteristic quantities are used at this step, each initial state is likely scaled with slightly different quantities, depending on the corresponding epoch.

4) State rotation: the dimensional states are rotated from the rotating frame to the inertial frame, centered at P_1.

5) State nondimensionalization: the states at the beginning of each arc are nondimensionalized, using the average characteristic quantities. This step helps construct states that have components in relatively similar ranges, reducing numerical sensitivity and aiding the subsequent correction schemes. Moreover, leveraging average characteristic quantities prevent the introduction of epoch-dependent terms in the transformation.

6) Initial guess construction: the initial guess vector $\boldsymbol{V}$ is constructed by appending the nondimensional states at each node, together with the associated epochs.

The generated first guess can be straightforwardly corrected leveraging a differential corrector scheme, as presented in Sec. 3.2. In this case, the scheme introduces a constraint vector $\boldsymbol{F}(\boldsymbol{V})$, enforcing state and time continuity at each node. Additional constraints can also be appended to the constraint vector, for example enforcing the initial state or the initial epoch to specified values [52, 86]. Direct application of rapid differential corrector schemes to the problem of trajectory refinement generally produces valid end-to-end trajectory in higher-fidelity models. However, the problem formulation does not enforce any geometrical similarity between the original trajectory in the CR3BP and the corrected path in the high-fidelity model. Thus, the corrected trajectory may present slight geometrical discrepancies with respect to the originating solution, especially when considering longer arcs near gravitationally sensitive areas. Therefore, it is sometimes preferable to generate an end-to-end solution in the higher-fidelity model that is geometrically similar to the starting trajectory retrieved in the CR3BP.

Among the variety of feasible approaches, this investigation leverages an optimization scheme to generate trajectories in a higher-fidelity models that minimize the geometrical distance between the low- and the higher-fidelity representations of the trajectory. In particular, the optimization scheme is designed in the traditional form:

$$\boldsymbol{V} = \arg\min_{\boldsymbol{V}} f(\boldsymbol{V}) \qquad \text{subject to } \boldsymbol{F}(\boldsymbol{V}) = \boldsymbol{0} \tag{3.33}$$

In this formulation, $\boldsymbol{V}$ incorporates the spacecraft states at each node and the associated epochs, and it is iteratively corrected to minimize the cost function $f(\boldsymbol{V})$, simultaneously satisfying the equality constraint $\boldsymbol{F}(\boldsymbol{V})$. For this formulation, the equality constraint vector is composed the state and time continuity constraints at each node, following a similar approach to what detailed for the orbit correction scheme in Sec. 3.2. To generate a solution in the higher-fidelity dynamical model that minimizes the distance from the original trajectory in the CR3BP, the cost function $f(\boldsymbol{V})$ is mathematically formulated as:

$$f(\boldsymbol{V}) = \sum_{i=1}^{N} \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i(t_i)\|^2 \tag{3.34}$$

where $\boldsymbol{x}_i \in \mathbb{R}^6$ represents the nondimensional spacecraft state at the i-th node with $i = \{1, \ldots, N\}$, t_i is the associated epoch, and $\|\cdot\|$ is the ℓ^2-norm. In Eq. (3.34), $\bar{\boldsymbol{x}}_i(t_i)$ represents the spacecraft state along the original solution in the CR3BP that possesses the closest distance from $\boldsymbol{x}_i$. To compute $\bar{\boldsymbol{x}}_i(t_i)$, the original trajectory in the CR3BP is first dimensionalized, leveraging the instantaneous characteristic quantities at t_i. Then, the entire trajectory is rotated in the P_1-centered inertial frame, and ultimately nondimensionalized using the average system-dependent characteristic quantities, as the ones presented in Table 2.1. Then, $\bar{\boldsymbol{x}}_i(t_i)$ is computed as the state from the scaled trajectory in the inertial frame with minimum distance from $\boldsymbol{x}_i$.

Leveraging the optimization scheme in the trajectory transfer process generally produces arcs in higher-fidelity models that closely match the geometry of the originating solution in the CR3BP, although requiring a larger computational cost. The nested approach with multiple shooting and optimization is leveraged throughout this work to generate trajectories in a higher-fidelity point mass ephemeris model.

Chapter 4

Machine Learning

Machine learning (ML) refers to a set of data-driven techniques that leverage algorithms to perform tasks for decision-making or prediction. One of the most popular definition of ML was given by Tom Mitchell as [87–89]:

> ”A computer program is said to learn from experience E with respect to some class of tasks T, and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.”

Techniques from machine learning are usually divided into three categories, depending on the nature of the task T, the selected performance metric P and the leveraged experience E [88]. These three areas are unsupervised learning, supervised learning and reinforcement learning:

- Unsupervised learning corresponds to a set of techniques used to extract latent information from an unlabeled dataset. Therefore, the dataset $\mathscr{D} = \{\boldsymbol{x}_i\}_{i=1}^{N}$ is constituted only by datapoints. From a probabilistic perspective, methods from unsupervised learning try to fit an unconditioned distribution on datapoints $p(\boldsymbol{x})$. Examples of unsupervised learning methods comprise clustering, manifold learning, and semi-supervised learning [88].

- Supervised learning represents the extended set of techniques that learns a mapping between input and output $f : \mathbb{X} \rightarrow \mathbb{Y}$ from a dataset $\mathscr{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{N}$. From a probabilistic perspective, supervised learning can be considered as the set of techniques that try to fit the conditional distribution $p(\boldsymbol{y}|\boldsymbol{x})$. Examples of supervised learning methods include classification and regression techniques [88].

- Reinforcement learning represents a class of algorithms where an agent is trained to learn an optimal policy by continuous interaction with an environment. A policy $\pi(\cdot)$ can be expressed as a deterministic or stochastic process, used to generate an action $\boldsymbol{a} = \pi(\boldsymbol{x})$. After every interaction with the environment, the agent can receive a reward. The role of the agent is then to maximize the cumulative reward received during a sequence of interactions with the environment. Different from supervised learning, the process of agent training can be considered as learning with a critic rather than learning with a teacher [88]. Moreover, the input-output pairs that populate the dataset $\mathscr{D}$ are generated through continuous interactions between the agent and the environment during the training, and are therefore not readily available at the beginning of the training.

This dissertation approaches different challenges of trajectory design with the aid of techniques from each of these three categories. In particular, techniques from unsupervised and reinforcement learning finds significantly more applications throughout this investigation, and are therefore detailed in Secs. 4.1 and 4.2.

4.1 Unsupervised Learning

Unsupervised learning techniques learn latent information contained within an unlabeled dataset without any external guidance. These techniques prevent the expensive process of labeling, often time-consuming for large datasets, and avoid the need of dataset pre-partition, that often introduces bias if the task is not well-defined. However, since the process is unsupervised, these techniques might generate unpredictable behavior [88]. Example of techniques from unsupervised learning include clustering and manifold learning, presented in Secs. 4.1.1 and 4.1.2.

4.1.1 Clustering

Clustering algorithms partition an unlabeled dataset into a finite number of groups, such that data in the same cluster are considered similar, while data in different clusters are deemed dissimilar [90]. Each of the N datapoints in the dataset is associated with a set of features, expressed by an M-dimensional feature vector, generating a dataset $\mathscr{D} \in \mathbb{R}^{N \times M}$. The generated clustering result, corresponding to the constructed partitioning of the dataset $\mathscr{D}$, is governed by the feature set used to represent each datapoint. The selected features represent an application-specific task, that needs to be engineered to grant useful information from the clustering result. Moreover, the selected clustering algorithm influences the quality of the generated results. The wide array of existing clustering methods is commonly classified in partitioning, density-based and hierarchical techniques [10, 12]:

- Partitioning methods: for these techniques, a prescribed number of clusters k, with $k \leq N$, is selected prior to the clustering processing. Initially, a number of cluster centers is assigned leveraging algorithm-specific techniques, and then iteratively corrected based on a distance-based metric. This class of clustering algorithms is generally effective for small- to medium-size datasets, often generating spherical-shaped clusters in the M-dimensional space. Examples of partitioning clustering methods include k-means, k-medoids (also known as Partitioning Around Medoids, or PAM) and Clustering LARge Applications (CLARA) [90–92].

- Density-based methods: these algorithms associate each datapoint in the dataset $\mathscr{D}$ to a local density information, based on proximity with other datapoints. Clusters are then constructed in each region of large density, and are separated by other clusters by the low-density areas. Using density instead of distance allows the generation of clusters of any shape, and the identification of outliers as noise. Examples of density-based clustering are Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and the Ordering Points to Identify the Clustering Structure Although (OPTICS) [90, 93, 94].

- Hierarchical methods: this set of techniques structure the dataset $\mathscr{D}$ into a hierarchy or tree, using a bottom-up (agglomerative) or top-down (divisive) approach. Hierarchical methods can leverage distance or density to generate similarities between datapoints. Example of hierarchical clustering algorithms include the Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) and Chameleon [90, 95, 96].

The Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) is employed in this investigation, and falls under both the density-based and the hierarchical categories; this algorithm is summarized in Sec. 4.1.1.1. The generated clustering result from HDBSCAN heavily depends on the input parameters and the characteristics of the dataset. Cluster validity indices can be used to validate the clustering result, aiding the user in the parameters selection process, and are therefore presented in Sec. 4.1.1.2. After generating a clustering result, each group can be summarized with a unique solution, called the representative, and presented in Sec. 4.1.1.3. Additionally, the representative solutions can aid the user to improve the visualization of higher-dimensional map. Eventually, an overview of distributed clustering, used to improve the computational efficiency of processing very large datasets, is highlighted in Sec. 4.1.1.4.

4.1.1.1 HDBSCAN

The Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) is a clustering algorithm that uses a density-based and hierarchical structure to group datapoints into clusters. The algorithm is introduced by Campello et al., and converts DBSCAN into a hierarchical method, also preventing the user selection of unintuitive parameters [14]. Once a dataset is input to HDBSCAN, a set of labels is generated that assigns each datapoint to a distinct cluster, or label it as noise.

In the formulation leveraged in this work, clustering with HDBSCAN is governed by two parameters and a distance metric. Indeed, although the algorithm is a density-based approach, it necessitates the introduction of a distance metric $d : \mathbb{R}^M \times \mathbb{R}^M \rightarrow \mathbb{R}$ to generate a definition of density. Examples of suitable distance metrics comprise the ℓ^2-norm, the Hausdorff distance and the Manhattan distance. The first input parameter is denoted as $m_{\text{pts}} \in \mathbb{N}^+$, and corresponds to the number of datapoints used to define the core distance $d_{\text{core}} : \mathbb{R}^M \rightarrow \mathbb{R}$. This is defined as the distance of the analyzed datapoint from the (m_{pts} - 1)-th nearest neighbor in the M-dimensional feature vector space, computed using the selected metric $d(\cdot, \cdot)$. A large core distance is associated with a datapoint residing in a low-density region, while a small core distance is typical of datapoints located in a highly dense regions. The core distance of each datapoint in the analyzed dataset allows the computation of a mutual reachability distance, obtained for each pair of datapoints as:

$$d_{\text{reach}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \max\left\{d_{\text{core}}(\boldsymbol{x}_i), d_{\text{core}}(\boldsymbol{x}_j), d(\boldsymbol{x}_i, \boldsymbol{x}_j)\right\} \tag{4.1}$$

where $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathscr{D}$. A conceptual visualization of the core distance between two datapoints in a planar Euclidean space is presented in Fig. 4.1, using the ℓ^2-norm as distance metric and $m_{\text{pts}} = 6$.

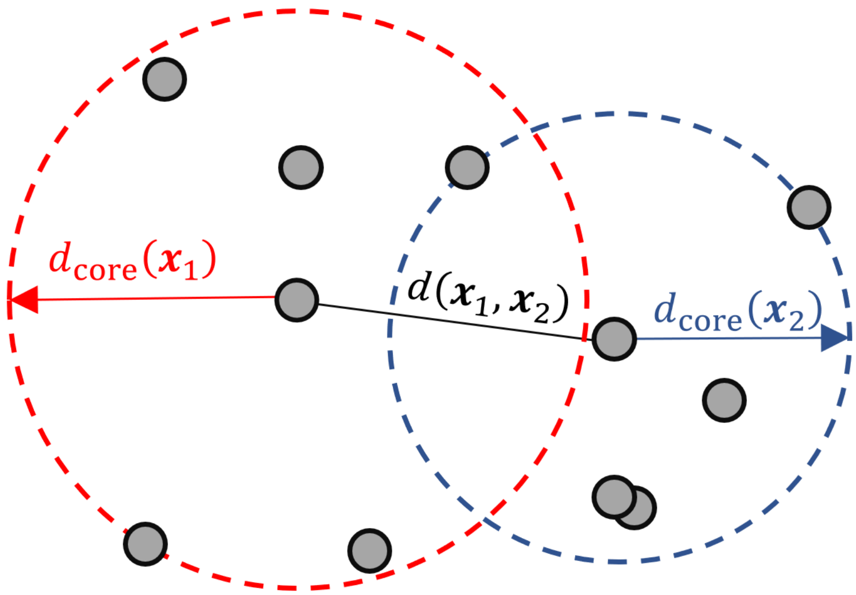

Figure 4.1: Example of core distance used by HDBSCAN (reproduced from McInnes et al. [97]).

The mutual reachability distance between each pair of datapoints is leveraged to infer information on the areas where the dataset possesses a large density of samples. The computed mutual reachability distances are used to construct a tree, where each node is represented by a datapoint and the weight connecting a pair of nodes is the associated mutual reachability distance. To provide a computationally efficient mechanism to construct the tree, a minimum spanning tree is selected. This tree represents a subset of the fully-connected tree, where: all nodes are connected; there is no cycle in the structure; only the edges between nodes generating the minimum total sum of weights are retained [90]. Then, the constructed minimum spanning tree is leveraged to generate a hierarchical structure of the dataset with a bottom-up approach: starting from the edge with minimum weight, corresponding to the minimum mutual reachability distance, a dendrogram is built by iteratively increasing the weights, until all the datapoints are collected.

HDBSCAN selects clusters based on their stability along the dendrogram. To compute the stability associated with each cluster, the dendrogram is processed in a top-down approach. At the top of the dendrogram, a single structure exists grouping the entire analyzed dataset. However, by progressively decreasing the mutual reachability distance, corresponding to stepping towards the bottom of the dendrogram, two different outcomes can be generated: a collection of points falls off from the parent structure; the parent structure splits into more children structures. The distinction between these two outcomes is decided by the user, through the specification of the second input parameter: the minimum cluster size $m_{\text{clSize}} \in \mathbb{N}^+$. If the number of points falling from the structure is lower than the input m_{clSize}, these points are regarded as leaving the structure, and the parent structure is preserved in the tree. However, if at least m_{clSize} points fall from the parent structure or the parent structure is split into two children structures. Iterating this approach through the dendrogram generates a condensed tree, used to obtain stability information of each cluster [97]. Indeed, within the condensed tree, each structure is associated with a value of mutual reachability distance of birth and death: these are related to the threshold values along the tree where the analyzed structure started existing and where it either reached the minimum mutual reachability of the hierarchy, or it split into two or more children structures. A stability value for each structure is then computed from the associated mutual reachability values of birth and death. With the condensed tree and the stability for each structure, the tree is ultimately crossed in a bottom-up approach. Initially, all the leaf nodes are selected clusters. If the sum of the stabilities of the children clusters is lower than the stability of the parent, the parent becomes a selected cluster, and the previous children are unselected. The approach is continued until reaching the root of the condensed tree, eventually returning the generated cluster differentiation [97]. According to Campello et al., the HDBSCAN algorithm is $\sim \mathcal{O}(MN^2)$ in time and $\sim \mathcal{O}(MN)$ in memory storage, when the clustering is performed on an $(N \times M)$-dimensional dataset [14]. In this work, the HDBSCAN algorithm is accessed through the *hdbscan* library in Python, with a computational complexity that approaches $\sim \mathcal{O}(N\log(N))$ [15].

4.1.1.2 Cluster Validity Index

Tuning the input parameters used by HDBSCAN poses a challenging task on the users, especially when possessing little to no a-priori knowledge of the underlying structure of the processed dataset. To verify the quality of the clustering result, the user can employ different validation criteria, often based on inspecting a single validation index. The existing validation criteria can be generally divided into two broad categories: external and internal. An external validation criteria is leveraged when a ground-truth solution of a dataset exists. Direct comparison with ground-truth clustering results is however of very limited use in real-world applications, since clustering is by definition an unsupervised learning task [16]. Conversely, internal validation criteria only leverage the available dataset to assess the quality of the clustering result. Among the internal methods, relative criteria seek to validate the clustering result by comparing two clustering structures and assess which one performs better. A variety of relative criteria are available, including the Davies-Bouldin index, the Dunn's index and the silhoutte width criterion (SWC) [98]. Although extremely efficient, these validation criteria are not suitable for validating density-based clustering results, because they are specifically tailored towards globular clusters. Other researchers incorporate graph theory or density-based concepts into cluster validation: however, these approaches struggle with arbitrarily shaped clusters, or present unintuitive selection criteria to evaluate the quality of the clustering result [98–101].

In an effort to define a relative validation criteria to evaluate the performance of density-based clustering, Moulavi et al. introduce the density based clustering validation ($DBCV$) index [16]. The $DBCV$ is governed by the maximum internal sparseness of each cluster and the highest density regions between pairs of clusters. To compute the $DBCV$, assume the dataset $\mathcal{D}$ is divided into l clusters $\mathcal{C}_i$, with $i \in \{1, 2, \ldots, l\}$. Then, Moulavi et al. introduce a parameterless metric, called the all-points core distance $a_{\text{pts,core}} : \mathbb{R}^M \to \mathbb{R}$ to represent the density information associated with each datapoint. The selected metric reflects the density in a neighborhood of each datapoing since: it considers all the points within each cluster; it is comparable to the metric distance $d(\cdot,\cdot)$ used in the definition of the core distance in Eq. (4.1); it approximates the distance from a k-th nearest neighbor, with k not too large [16]. Then, to compute the distance between two points in a cluster, the same definition of mutual reachability distance detailed in Eq. (4.1) is used, although replacing the core distance with the all-points core distance. With the generated all-points mutual reachability distances between each pair of datapoints in each cluster, a minimum spanning tree is generated. Note the all-points mutual reachability distances are computed only among members of the same clusters, and each cluster is associated with a unique minimum spanning tree [15, 16].

The minimum spanning trees of each cluster enclose the required information to compute the sparseness of the clusters and the separation between clusters. Indeed, the maximum edge of the minimum spanning tree defines the density sparseness of a cluster $DSC(\mathscr{C}_i)$, since it is associated with the lowest density region. The density separation between clusters $DSPC(\mathscr{C}_i, \mathscr{C}_j)$ is computed using the minimum all-points mutual reachability distance between the minimum spanning trees associated with two distinct clusters. Using the DSC and the $DSPC$, the validity index of each cluster can be computed as [16]:

$$V_C(\mathscr{C}_j) = \frac{\min_{k\in\{1,\dots,l\}, j\neq k}(DSPC(\mathscr{C}_j, \mathscr{C}_k)) - DSC(\mathscr{C}_j)}{\max\left[\min_{k\in\{1,\dots,l\}, j\neq k}(DSPC(\mathscr{C}_j, \mathscr{C}_k)), DSC(\mathscr{C}_j)\right]} \tag{4.2}$$

The validity index of the generated clusters are leveraged to compute the $DBCV$ as:

$$DBCV = \sum_{j=1}^{l} \frac{|\mathscr{C}_j|}{|\mathscr{D}|} V_C(\mathscr{C}_j) \tag{4.3}$$

where the cardinality of a set $|\cdot|$ represents the total amount of points in the considered set. From the mathematical definition in Eq. (4.3), it can be verified that $-1 \leq DBCV \leq 1$. A computationally efficient approximations of the validity indices and the $DBCV$, available in the *hdbscan* library in Python, are leveraged in this investigation [15].

4.1.1.3 Cluster Representative

Simultaneous visualization of the clustering result often represents a challenging task, particularly for higher-dimensional datasets. However, each cluster can be summarized by a unique solution, representing the features of the associated cluster. Visualizing a set of representative solutions can significantly improve the visualization of the generated differentiation of a dataset, aiding in the analysis of the investigated dataset. Different criteria can be leveraged to identify a representative solution for each cluster. In this investigation, the definition of medoid of a cluster is used to compute a representative solution for each cluster. The medoid is a member of a cluster that is most similar to all the other members in the same cluster. This definition is particularly advantageous to generate representative solutions of density-based clustering results [12, 90]. In particular, the medoid for the j-th cluster $\mathscr{C}_j$ is computed as:

$$\boldsymbol{x}_{\mathrm{med}}^{(j)} = \underset{\boldsymbol{x}_k^{(j)} \in \mathscr{C}_j}{\arg\min} \sum_{i=1,\, i \neq k}^{|\mathscr{C}_j|} d\left(\boldsymbol{x}_i^{(j)},\, \boldsymbol{x}_k^{(j)}\right) \tag{4.4}$$

where $d(\cdot, \cdot)$ corresponds to the same metric used in the HDBSCAN algorithm [12, 90].

4.1.1.4 Distributed Clustering

Clustering an entire dataset in a single step can sometimes represent a challenging or impossible task. For example, an entire dataset might be unavailable on a unique machine or at a single time, due to technical, security, or economic issues. Moreover, the dataset might be too large to be processed in a single batch. In these situations, techniques from distributed clustering can be used to efficiently and accurately process singular subsets of the original dataset in a distributed approach across different computational machines and times, eventually generating a similar or improved clustering result with respect to the centralized approach [102].

In this investigation, distributed clustering is used to aid the computational burden of clustering a very large dataset. Different machines can cooperate to cluster partitions of the dataset, at different times, later sharing their output to obtain a unique partitioning. A distributed data mining approach usually consists of four fundamental phases [102]. In the first step, the dataset is split into different groups which are locally clustered by different machines, generating the local models. Then, information about the local models is shared among the different machines. This step is designed to enhance rapid data sharing, low storage requirements, and sufficient preservation of the structure in each local model. For the third step, the local models are aggregated to generate a global clustering result of the entire dataset, also called the global model. In the fourth and final phase, the global model is returned to each machine, that assigns each datapoints to one of the generated global cluster. Bendechache et al. demonstrate that distributed clustering scales well with large dataset, minimizes communications between machines, and can outperform centralized clustering in both result quality and computational time [17].

4.1.2 Dimensionality Reduction

Dimensionality reduction represents a form of unsupervised learning that tries to learn a mapping from a high-dimensional space $\mathbb{R}^M$ onto a low-dimensional representation $f : \mathbb{R}^M \rightarrow \mathbb{R}^L$, with $L < M$. If the learned mapping is parametric, the embedded representation $\boldsymbol{\varkappa}$ can be generated from the high-dimensional datapoint $\boldsymbol{x}$ leveraging a set of parameters as $\boldsymbol{\varkappa} = \boldsymbol{f}(\boldsymbol{x}; \boldsymbol{\theta})$. Parametric embeddings are often leveraged at the early phases of deep learning pipelines for data pre-processing [103–105]. When the mapping is not parameteric, the dimensionality reduction algorithm tries to directly learn the embedding $\boldsymbol{\varkappa}$ for each of the high-dimensional input $\boldsymbol{x}$: this version is often leveraged to visualize high-dimensional datasets, representing a rapid dimensionality reduction approach [88].

Different algorithms exist to project a high-dimensional dataset onto a lower-dimensional representation. These algorithms are often grouped in two main categories: linear and nonlinear dimensionality reduction techniques. Principal component analysis (PCA) represents one of the most widely used linear dimensionality reduction algorithms [88]. In PCA, an orthogonal projection of the high dimensional dataset is sought, using the largest eigenvectors of the empirical covariance matrix, computed with the available dataset [88]. PCA and other linear dimensionality reduction techniques are computationally lightweight, but struggle to generate embeddings for highly nonlinear high-dimensional datasets.

Algorithms for nonlinear dimensionality reduction are often separated in two families: autoencoders and manifold learning. Autoencoders comprise a wide array of solutions, including basic, de-noising, variational and contractive autoencoders [88,106–108] . They represent parametric dimensionality reduction techniques that learn mappings between the high- and low-dimensional representation: they can also be employed to generate embeddings of samples that are not included in the training set. Neural networks can be used to incorporate nonlinearities in the learned mapping. Autoencoders represent powerful dimensionality reduction techniques, that often require large dataset and intensive computational resources to learn the mapping [88].

Manifold learning techniques represent nonlinear dimensionality reduction algorithm that assume the high-dimensional dataset lies on a curved and low-dimensional manifold. A manifold is a topological space $\mathcal{X}$ where each point $\boldsymbol{x} \in \mathcal{X}$ has a neighborhood that is equivalent to an M-dimensional Euclidean space. When a differentiable manifold allows an inner product operator the space is called a Reimannian manifold. Manifold learning techniques are nonparameteric dimensionality reduction algorithms, therefore they cannot generally construct embeddings for out-of-the-dataset samples [109]. However, they are often easier to train, and more flexible with respect to autoencoders [88]. A variety of algorithms for manifold learning exists, distinguished by the structure of the assumed manifold, and the associated computational strategies [88]. The most widely used techniques include: Isomap, local linear embedding, Laplacian eigenmaps or spectral embedding, and t-SNE [110–113]. Among the state-of-the-art solutions for nonlinear dimensionality reduction, the Uniform Manifold Approximation and Projection (UMAP) represents a computationally lightweight solution to find a nonparameteric mapping from the high-dimensional to the low-dimensional representation [32]. Recently, a parameteric version of UMAP has also been developed [114]. UMAP is engineered to minimize the difference in global structure between the low-dimensional embedding and the higher-dimensional dataset. An overview of the UMAP algorithm is presented in Sec. 4.1.2.1 [32].

4.1.2.1 UMAP

UMAP is a dimensionality reduction algorithm based on manifold learning. It uses topology theory to construct a fuzzy representation of an input high-dimensional dataset. Then, UMAP optimizes a low-dimensional representation to minimize the topological distance from the high-dimensional representation. With this approach, UMAP generates a bijective mapping between the high- and low-dimensional representation. Leveraging topology in the algorithm provides a mathematical framework that motivates the underlying approach, also aiding to solve potential issues. In the presented investigation, three parameters govern the UMAP algorithm: n_n, n_c, and m_{dist} [32, 115].

The first step of the UMAP algorithm is the construction of a topological representation of the high-dimensional dataset. The algorithm initially assumes the high-dimensional dataset is uniformly distributed: the latter represents a strong assumption that can be violated on a variety of datasets. However, UMAP also assumes the distance metric varies on the higher-dimensional representation: regions where the high-dimensional datapoints are sparse are associated to stretched areas, while highly dense regions are related to compressed areas. Enforcing a varying distance metric over the dataset validates the application of the uniform distribution assumption. With these assumptions, a local distance can be computed using Reimannian geometry, allowing each point to be associated with its own local metric. Practically, a unit-radius hyper-sphere centered at an isolated datapoint would look like a large sphere in an Euclidean space, while the same unit-radius hyper-sphere would shrink for datapoints in highly dense regions. This approach generates the same result of using an n_n-neighbor graph, where the choice of n_n determines the degree of estimation of the local Reimannian metric: a small n_n is associated with a very local approximation that more accurately captures the local structure, while a large n_n is associated with a broader approximation, capturing the global structure of the dataset. The number of datapoints used by UMAP to locally approximate the manifold, $n_n \in \mathbb{N}^+$, represents a user-selected parameter. Then, UMAP converts the graph into a fuzzy representation, decreasing the likelihood of connections between datapoints as the radius grows. The fuzzy graph representation might generate a large number of isolated samples for applications in high-dimensional dataset. Therefore, UMAP assumes the graph is also locally connected. Then, fuzzy union set theory is used to generate the final weighted graph, used as a fuzzy topological representation of the high-dimensional dataset [32, 115].

The constructed fuzzy topological representation of the dataset is leveraged in the second phase of the UMAP algorithm to generate a similar fuzzy topological structure of a low-dimensional representation. To uncover this representation, UMAP needs to identify the underlying low-dimensional manifold. However, the algorithm assumes the low-dimensional manifold is the n_c-dimensional Euclidean space, where the distance metric is already defined as the Euclidean norm. The dimensionality of the projected manifold, $n_c \in \mathbb{N}^+$, represents a user-selected parameter. Moreover, the minimum distance between two datapoints in the low-dimensional representation, $m_{\text{dist}} \in \mathbb{N}^+$, is selected by the user: lower values generate more compact representations, used for example in a clustering pipeline. The low-dimensional representation is then initialized using spectral embedding, and iteratively corrected using stochastic gradient descent by minimizing the cross-entropy between the weights of the high- and low-dimensional graphs. The overall computational complexity of UMAP is driven by the n_n-neighbor search, empirically approximated as $\sim \mathcal{O}(N^{1.14})$, and the stochastic gradient descent step, $\sim \mathcal{O}(n_n N)$ [32]. In this investigation, UMAP is accessed via the *umap-learn* library in Python [32].

4.2 Reinforcement Learning

Reinforcement learning comprises a set of techniques where an agent interacts with an environment to generate a sequence of actions that optimize a final task. The actor iteratively interacts with the environment to improve its performance in the designed task. After the training, the agent is deployed in a scenario to actively leverage the learned decision mechanisms. The learning and inference phases are often referred to as training and testing, respectively. During these two phases, the agent and the environment exchange information. In particular, Fig. 4.2 depicts a conceptual visualization of the agent-environment interaction: the agent receives a representation of the environment state $\boldsymbol{s} \in \mathcal{S}$ from the environment, and consequently generates an action $\boldsymbol{a} \in \mathcal{A}(\boldsymbol{s})$. Then, the obtained action is leveraged by the environment, together with the current state $\boldsymbol{s}$, to generate a new representation of the state $\boldsymbol{s}^*$. The information associated with the tuple $(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{s}^*)$ is used by the environment to generate a reward $r \in \mathcal{R}$. The reward is used by the agent at the next iterate to measure the quality of the state-action pair $(\boldsymbol{s}, \boldsymbol{a})$ in contribution to the ultimate task of the agent. The entire process of generating an action and a subsequent reward from a starting state is often referred to as step, and corresponds to a single complete interaction between the agent and the environment. The criteria used by the agent to select the current action, given the current state representation of the environment, is called the policy $\pi(\boldsymbol{a}|\boldsymbol{s})$. The policy can be represented by a deterministic function mapping the state to an action, or as a distribution over the space of the action $\mathscr{A}(\boldsymbol{s})$. A stochastic policy can be used during training to allow exploration of the action space. The same stochastic policy can then be converted into a deterministic function at test time, where exploitation of the trained policy is preferred to generate an optimal action. The action selection process of a policy generally continues until certain termination conditions are met. When a termination criteria is met, the environment resets the current state representation to a state sampled from an initial set $\mathscr{S}_0$. The sequence of agent-environment interactions, from the starting step till termination criteria are met, is often referred to as an episode.

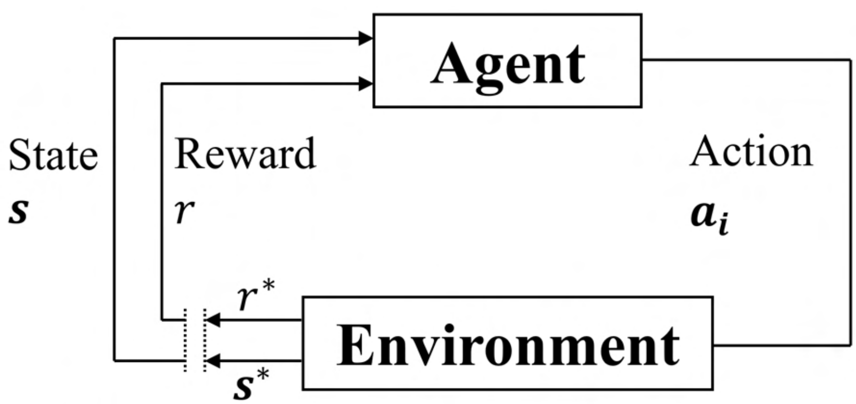

Figure 4.2: Conceptual visualization of the agent-environment interaction in reinforcement learning (reproduced from Sutton and Barto, Fig. 3.1 [116]).

To allow a rigorous mathematical description, a reinforcement learning framework is often formulated as a Markov Decision Process (MDP) [116]. An MDP benefits from the Markov property, stating that the future state of the system uniquely depends on the current state and selected action. In an MDP, the state might be considered as fully observable. Conversely, the agent cannot observe the full state representation in partially observable MDPs. In this investigation, each state is assumed as fully observable, such that observation and state are used interchangeably. The future state $\boldsymbol{s}^* \in \mathcal{S}$ and reward $r \in \mathcal{R}$ possess a well-defined continuous distribution, depending on the current state and action, described as $p(\boldsymbol{s}^*, r|\boldsymbol{s}, \boldsymbol{a})$. This distribution can be marginalized to generate a state-transition distribution for the state $\boldsymbol{s}^*$ as:

$$p(\boldsymbol{s}^*|\boldsymbol{s}, \boldsymbol{a}) = \int_r p(\boldsymbol{s}^*, r|\boldsymbol{s}, \boldsymbol{a}) \tag{4.5}$$

or an expected reward, as:

$$r(\boldsymbol{s}, \boldsymbol{a}) = \mathbb{E}\left[r_t|\boldsymbol{s}_t = \boldsymbol{s}, \boldsymbol{a}_t = \boldsymbol{a}\right] = \int_r r \int_{\boldsymbol{s}^*} p(\boldsymbol{s}^*, r|\boldsymbol{s}, \boldsymbol{a}) \tag{4.6}$$

where the subscript $(\cdot)_t$ indicates a random variable associated with the current sample t. The goal of an agent is to select an action that leads to the maximization of the cumulative return over the episode. At the t-th step along an episode with total number of steps N, the return can be represented through a weighted sum of rewards as:

$$G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots + \gamma^{N-t-1} r_N = \sum_{i=0}^{\infty} \gamma^i r_{t+i+1} \tag{4.7}$$

The discount factor $\gamma \in [0, 1)$ reflects the importance of future rewards with respect to the current step. Indeed, a discount factor $\gamma \to 0$ generates a return that assigns more importance to the immediate reward, while if $\gamma \to 1$ future rewards gain more importance in the return formulation [116]. The return in Eq. (4.7) can be represented in a recursive formulation as

$$G_t = r_{t+1} + \gamma G_{t+1} \tag{4.8}$$

To have a mathematical representation of the expected return, Eq. (4.7) is used to generate the definition of the value. Two value functions are usually defined in the literature to generate information on the expected return. The first value function, often referred to as state-value function, corresponds to an expectation of the return conditioned on the current state. Its mathematical formulation for the state at time t is:

$$\begin{aligned}
v_\pi(\boldsymbol{s}) =& \mathbb{E}[G_t|\boldsymbol{s}_t = \boldsymbol{s}] \\
=& \mathbb{E}[r_{t+1} + \gamma G_{t+1}|\boldsymbol{s}_t = \boldsymbol{s}] \\
=& \int_{\boldsymbol{a}} \pi(\boldsymbol{a}|\boldsymbol{s}) \int_{\boldsymbol{s}^*,r} p(\boldsymbol{s}^*, r|\boldsymbol{s}, \boldsymbol{a}) \left[r + \gamma v_\pi(\boldsymbol{s}^*)\right]
\end{aligned} \tag{4.9}$$

where the subscript $(\cdot)_\pi$ indicates the sequence of actions for each state of the episode is generated from the same policy $\pi(\boldsymbol{a}|\boldsymbol{s})$, and the expectation is over the action, reward and future state. Equation 4.9 expresses the recursive nature of the state-value function, and encodes the Bellman equation for v_π. Solving the Bellman equation is at the foundation of a variety of reinforcement learning algorithms. A second function, called the action-value function or q-value function, can be expressed as:

$$q_\pi(\boldsymbol{s}, \boldsymbol{a}) = \mathbb{E}[G_t|\boldsymbol{s}_t = \boldsymbol{s}, \boldsymbol{a}_t = \boldsymbol{a}] \tag{4.10}$$

where the expectation is over the reward and future state. Differently from the state-value function, the q-value function is conditioned on both current state and action. The practical difference is that, for the state-value function $v_\pi(\cdot)$, the policy $\pi(\boldsymbol{a}|\boldsymbol{s})$ is leveraged to retrieve the actions for the entire sequence of states in the episode, starting from $\boldsymbol{s}_t$. Conversely, the q-value function might potentially follow a different policy for the immediate action selection $\boldsymbol{a}_t$; however, starting from the subsequent state $\boldsymbol{s}_{t+1}$, the q-value function follows the actions sampled from $\pi(\boldsymbol{a}|\boldsymbol{s})$. For this reason, the relation between the state- and action-value function is:

$$\begin{aligned}
v_\pi(\boldsymbol{s}) &= \int_{\boldsymbol{a}} \pi(\boldsymbol{a}|\boldsymbol{s}) q_\pi(\boldsymbol{s}, \boldsymbol{a}) \\
q_\pi(\boldsymbol{s}, \boldsymbol{a}) &= \int_{\boldsymbol{s}^*,r} p(\boldsymbol{s}^*, r|\boldsymbol{s}, \boldsymbol{a}) \left[r + \gamma v_\pi(\boldsymbol{s}^*)\right]
\end{aligned} \tag{4.11}$$

Correct estimation of the return by the value function forms the foundation of a variety of algorithms in reinforcement learning. For problems with continuous state and action space, the value function can be approximated with neural networks, detailed in Sec. 4.2.1. The value network can therefore be trained through repeated interactions between the agent and the environment. However, the agent might not possess perfect knowledge of the model governing the state transition $p(\boldsymbol{s}^*, r|\boldsymbol{s}, \boldsymbol{a})$:

in this situation the agent cannot actively leverage this information for its prediction, and the algorithm used for training are referred to as model-free learning. Value-based, policy-gradient and actor-critic approaches constitute three of the main classes of model-free learning algorithms, and are outlined in Sec. 4.2.1. Among the state-of-the-art algorithms for training policies in continuous state and action spaces, trust region policy optimization and the family of proximal policy optimization, introduced in Sec. 4.2.3, represent alternatives to balance exploration and exploitation during training. Eventually, a brief introduction of transfer learning in machine learning, used in this investigation to accelerate the training in challenging environments, is introduced in Sec. 4.2.4.

4.2.1 Feedforward Neural Networks

Neural networks serve as function approximators $\hat{f} : \mathbb{R}^n \to \mathbb{R}^m$, mapping a generic input $\boldsymbol{x} \in \mathbb{R}^n$ into an output $\hat{\boldsymbol{y}} \in \mathbb{R}^m$, and modeled as $\hat{\boldsymbol{y}} = \hat{f}(\boldsymbol{x})$. Neural networks are constructed to closely reproduce the authentic and unmodeled relation between input and ground truth $\boldsymbol{y} = f(\boldsymbol{x})$. Among the different architectures of neural networks, this investigation leverages deep feedforward networks, also called multi-layer perceptrons (MLP). Differently from recurrent architectures, feedforward networks do not present feedback connections, where the output is fed back as part of the input; instead, the information flows in a single direction in feedforward networks, from the input to the output [89]. A feedforward network, shortly referred to as neural network or network in this investigation, is typically represented by a composition (or network) of layers, each acting as a function. Given a generic input $\boldsymbol{x}$, each layer can be associated to a mapping $f_i(\cdot)$, with $i \in \{1, 2, \dots, N_l\}$. For example, in a network with three layers, the entire mapping can be expressed as $\hat{f}(\boldsymbol{x}) = f_1(f_2(f_3(\boldsymbol{x})))$. The amount of layers N_l used to approximate the mapping determines the depth of the network. Among the different layers $f_i(\cdot)$ of a deep network, the first and last layers are called input and output layer. The input of the input layer represents the input of the network $\boldsymbol{x}$, while the output of the output layer represents the output of the network $\hat{\boldsymbol{y}}$. Hidden layers can be introduced between the input and output layer. A hidden layer receives as input the output of the previous layer, and its output corresponds to the input of the next layer [89].

The layers of a deep network can also be considered as a set of units, called neurons or perceptrons, that act in parallel on the same input. Different layers can have distinct number of neurons, each describing the width of the associated layer. The units are called neurons since they act like neuron cells: they receive inputs from a set of neurons but they only generate a single output. In a fully-connected feedforward network, each neuron receives as input: the output of the neurons of the previous layer of the same network if the neuron is not part of the input layer; the input vector $\boldsymbol{x}$ if the neuron resides in the input layer. A conceptual visualization of a multi-layer feedforward neural network, depicted as a collection of neurons, is presented in Fig. 4.3.

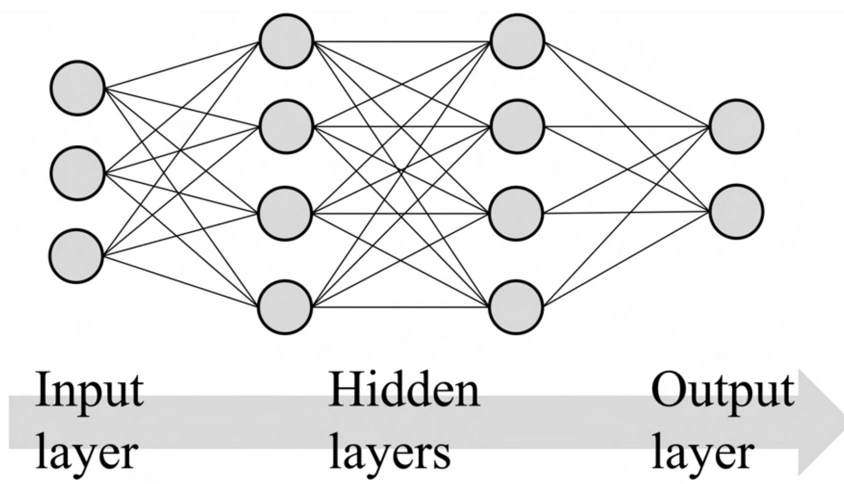

Figure 4.3: Conceptual visualization of a feedforward fully-connected neural network.

Each perceptron in every layer of the neural network receives an input and generates an output. For a unit of the initial layer, the perceptron takes the network input $\boldsymbol{x}$ and generates the output $\boldsymbol{h}_1 \in \mathbb{R}^l$. The perceptron is a deterministic version of the logistic regression, with the following mathematical formulation:

$$\boldsymbol{h}_1 = H(\tilde{\boldsymbol{h}}_1) = H(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}) \tag{4.12}$$

where $\boldsymbol{b} \in \mathbb{R}^l$ is a bias vector, the weight matrix is $\boldsymbol{W} \in \mathbb{R}^{l \times n}$, and $H : \mathbb{R} \to \{0, 1\}$ represents the Heaviside step function [88]. With the formulation presented in Eq. (4.12), the input is first linearly mapped into a temporary vector $\tilde{\boldsymbol{h}}_1 = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}$, leveraging the weight and bias vectors

$\boldsymbol{W}$ and $\boldsymbol{b}$, and then transformed in a value within the range $\{0, 1\}$ by the Heaviside function [88]. The Heaviside function can be replaced with an arbitrary differentiable activation function $\phi : \mathbb{R}^l \to \mathbb{R}$. The selection of an optimal activation function is often problem-dependent, although common representations include: sigmoid, arc tangent, rectified linear unit, softmax. With the activation function, the mathematical manipulation performed by a perceptron becomes:

$$\boldsymbol{h}_1 = \phi(\boldsymbol{w}\boldsymbol{x} + \boldsymbol{b}) \tag{4.13}$$

Different neurons can have distinct activation functions, although neurons joining the same layer are often associated to the same $\phi(\cdot)$. To aid readability and compactness of the representation, the parameters of a neural network, composed by the weights and biases of each neuron, are often included in a single representation $\boldsymbol{\theta}$, and the associated network can be expressed as $\hat{f}(\cdot\,; \boldsymbol{\theta})$.

Two common characteristics for activation functions used in many networks are nonlinearity and differentiability. Nonlinearity allows to generate a mapping $\hat{f}(\cdot\,; \boldsymbol{\theta})$ that can approximate a nonlinear relation between input and output in a network. Differentiability allows to iteratively train the network parameters through the definition of a loss function, expressed as:

$$l(\boldsymbol{y}, \boldsymbol{x}; \boldsymbol{\theta}) = l(\boldsymbol{y}, \hat{\boldsymbol{y}}) = l(\boldsymbol{y}, f(\boldsymbol{x}; \boldsymbol{\theta})) \tag{4.14}$$

The loss function $l : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ is a problem-dependent mapping that converts the output of the network $\hat{\boldsymbol{y}}$, and the ground truth $\boldsymbol{y}$ associated with the same input $\boldsymbol{x}$, to a real number. Loss functions can be characterized by assuming a defined distribution for the likelihood $p(\boldsymbol{y}|\boldsymbol{x})$, often used in probabilistic approaches for machine learning to express a model. For neural networks used in linear regression tasks, the likelihood can be assumed as a Gaussian distribution, generating a loss equivalent to the mean squared error when using maximum likelihood estimation to retrieve the parameters $\boldsymbol{\theta}$ [88]. When a certain loss function is initialized, the parameters $\boldsymbol{\theta}$ can be updated in order to minimize the losses $l_i \in \{l(\boldsymbol{y}_1, \hat{\boldsymbol{y}}_1), l(\boldsymbol{y}_2, \hat{\boldsymbol{y}}_2), \ldots, l(\boldsymbol{y}_N, \hat{\boldsymbol{y}}_N)\}$ over a batch of input and output $\{(\boldsymbol{x}_1, \boldsymbol{y}_1), (\boldsymbol{x}_2, \boldsymbol{y}_2), \ldots, (\boldsymbol{x}_N, \boldsymbol{y}_N)\}$. Stochastic gradient descent can be used to update $\boldsymbol{\theta}$ in the opposite direction of maximum growth of the loss with respect to the parameters as:

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \alpha \frac{\partial l_i}{\partial \boldsymbol{\theta}} \tag{4.15}$$

where α is the learning rate, used to stabilize the update process and avoid large steps along the direction of the gradient. The gradient is subtracted from the parameters to minimize the loss. The derivative of the loss function with respect to the network parameters can be computed because of the linear mapping of the input inside the formulation of the perceptrons, and the differentiability of the leveraged activation functions. The process of retrieving the gradient of the loss with respect to the network parameters, involving chains of partial derivatives, is often called backpropagation.

4.2.2 Value-Based, Policy-Gradient and Actor-Critic Methods

Reinforcement learning methods are generally categorized as: value-based, policy-gradient, and actor-critic. This section outlines the fundamental structure and mathematical background of these methodologies.

Value-based methods:

Value-based algorithms try to estimate the value function, usually corresponding to $q_\pi(\boldsymbol{s}, \boldsymbol{a})$. The value function is often initialized at the beginning of the training process, and then iteratively adjusted. The optimal policy at a given state $\boldsymbol{s}_i$ can be obtained by selecting the action that maximizes the value. In reinforcement learning scenarios with continuous observation space, the value function can be approximated via a neural network $q_\pi(\boldsymbol{s}, \boldsymbol{a}; \boldsymbol{\chi})$, with parameters $\boldsymbol{\chi}$. The parameters are iteratively adjusted according to a user-designed and problem-dependent loss. Usually, the mean squared error over the considered batch of experience between the approximated value from $q_\pi(\boldsymbol{s}, \boldsymbol{a}; \boldsymbol{\chi})$ and the true value is set to:

$$l = \frac{1}{N} \sum_{\boldsymbol{a}_i, \boldsymbol{s}_i} \left(q_\pi(\boldsymbol{s}_i, \boldsymbol{a}_i; \boldsymbol{\chi}) - q_{\text{true}}(\boldsymbol{s}_i, \boldsymbol{a}_i; \boldsymbol{\chi}) \right) \tag{4.16}$$

where N represents the dimension of the dataset $\mathscr{D} = \{(\boldsymbol{a}_i, \boldsymbol{s}_i)\}_{i=1}^N$. Various techniques exist to compute the true value $q_{\text{true}}(\boldsymbol{s}_i, \boldsymbol{a}_i; \boldsymbol{\chi})$. If the episode associated with the pair $(\boldsymbol{s}_i, \boldsymbol{a}_i)$ is completely recorded in the batch $\mathscr{D}$, the true value $q_{\text{true}}(\boldsymbol{s}_i, \boldsymbol{a}_i; \boldsymbol{\chi})$ can be computed according to Eq. (4.7) by the cumulative reward starting at $(\boldsymbol{s}_i, \boldsymbol{a}_i)$ and following the recorded episode. If the episode associated with the pair $(\boldsymbol{s}_i, \boldsymbol{a}_i)$ is not concluded in the batch $\mathscr{D}$, or if termination criteria of episodes are difficult to meet, the cumulative reward can be approximated as the sum of the immediate reward at the end of each step, augmented by the value obtained from a value function $q_{(\cdot)}$ at the subsequent step. Alternative formulations of value-based approaches can simultaneously train multiple and distinct value functions to reduce the maximization biases [56, 116]. However, these approaches can generally be leveraged when the action space is discrete and the observation space is extremely vast. Prior training with supervised learning or mechanisms like replay-buffer can be used to improve the quality of the training process [57, 58]

Policy-gradient methods:

Unlike value-based mechanisms, policy-gradient algorithms directly learn a policy $\pi(\boldsymbol{a}|\boldsymbol{s})$. The policy is often represented as a stochastic distribution, that encourages exploration early on in the training process and it never becomes deterministic. For policy-gradient methods with continuous action and state space, the policy can be approximated with a neural network $\pi(\boldsymbol{a}|\boldsymbol{s}; \boldsymbol{\theta})$, with parameters $\boldsymbol{\theta}$. Approximating with a neural network structure allows the policy to be differentiable. The parameters are usually initialized at the beginning of the learning process and iteratively updated when a sufficient batch of experiences is available. A loss function used to update a policy-gradient method is based on the value at each state as:

$$l_P(\boldsymbol{s}_i, \boldsymbol{\theta}) = v_\pi(\boldsymbol{s}_i) \tag{4.17}$$

Differently from the value-based approach where the loss function is minimized, in the case of Eq. (4.17) the policy parameters are iteratively adjusted to maximize the cost. Gradient ascent can be used to adjust $\boldsymbol{\theta}$ when a sufficient batch of experiences is available. It can be demonstrated through the policy gradient theorem that the gradient of the loss in Eq. (4.17) is [116]:

$$\frac{\partial l_P}{\partial \boldsymbol{\theta}} = \frac{\partial v_\pi(\boldsymbol{s}_i)}{\partial \boldsymbol{\theta}} = \mathbb{E}\left[G_t \frac{1}{\pi(\boldsymbol{a}_t|\boldsymbol{s}_t; \boldsymbol{\theta})} \frac{\partial \pi(\boldsymbol{a}_t|\boldsymbol{s}_t; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right] \tag{4.18}$$

This expression enables the practical generation of an approximation of the gradient by sampling the argument of the expectation. Indeed, the single parameter update can be executed as:

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i + \alpha G_t \frac{1}{\pi(\boldsymbol{a}_t|\boldsymbol{s}_t;\boldsymbol{\theta})} \frac{\partial \pi(\boldsymbol{a}_t|\boldsymbol{s}_t;\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \tag{4.19}$$

where α is the learning rate. The update algorithm in Eq. (4.19) is often called REINFORCE, and represents a basic policy-based algorithm [116]. Policy-gradient algorithms are generally more sample-efficient than value-based methods, since they are often implemented in an *on-policy* configuration, where the experiences are retrieved by leveraging the current policy to determine the action $\boldsymbol{a}_t$ given the state $\boldsymbol{s}_t$. Conversely, value-based approaches are more stable since they encourage larger exploration of the state and action space and are generally more data efficient [117].

Actor-critic methods:

Different variations of the basic policy-gradient formulation for parametrized policies in Eq. (4.19) exist. For example, it can be demonstrated that any function that is not dependent on the action can be leveraged as a baseline for the return G_t in Eq. (4.19). If an approximation of the value $v_\pi(\boldsymbol{s};\boldsymbol{\chi})$ is leveraged as a baseline, the update can be formulated as:

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i + \alpha(G_t - v_\pi(\boldsymbol{s_t};\boldsymbol{\chi})) \frac{1}{\pi(\boldsymbol{a}_t|\boldsymbol{s}_t;\boldsymbol{\theta})} \frac{\partial \pi(\boldsymbol{a}_t|\boldsymbol{s}_t;\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \tag{4.20}$$

which corresponds to an algorithm called REINFORCE with baseline [116]. Equation 4.20 typically improves the performance of the trained policy with respect to the basic formulation in Eq. (4.19), by lowering the variance of the gradient [57]. Moreover, the difference between return and value used to scale the gradient of the policy in Eq. (4.20) can also be interpreted as the advantage of taking the action $\boldsymbol{a}_t$ at the state $\boldsymbol{s}_t$. Indeed, G_t represents the estimate of $q_\pi(\boldsymbol{a}_t, \boldsymbol{s}_t)$, defining the advantage as:

$$A(\boldsymbol{a}_t, \boldsymbol{s}_t) = q_\pi(\boldsymbol{a}_t, \boldsymbol{s}_t) - v_\pi(\boldsymbol{s}_t) \tag{4.21}$$

The update rule in Eq. (4.20) requires the definition and training of an approximation of the value function $v_\pi(\boldsymbol{s};\boldsymbol{\chi})$. The latter can be simultaneously updated with the policy, using for example the loss function in Eq. (4.16), reformulated for the state-value function. The update rule of the

REINFORCE with baseline requires the definition and storage of a stochastic policy $\pi(\boldsymbol{a}_t|\boldsymbol{s}_t;\boldsymbol{\theta})$, together with a value function $v_\pi(\boldsymbol{s};\boldsymbol{\chi})$. In literature, this approach is still considered as a policy-gradient approach: a stochastic policy is directly stored and iteratively corrected, eliminating the need to retrieve the policy from the maximization of the value function [116]. A third category of reinforcement learning algorithms is generated when the formulation in Eq. (4.20) is slightly modified. Indeed, the return G_t can also be approximated, using a first-order truncation as:

$$G_t = r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{N-t-1} r_N \approx r_{t+1} + \gamma v_\pi(\boldsymbol{s}_{t+1};\boldsymbol{\chi}) \tag{4.22}$$

The approximation in Eq. (4.22) generates a new family of algorithms known as actor-critic methods. The difference between an actor-critic formulation and a policy-gradient approach lies in the formulation of Eqs. (4.20) and (4.22). Indeed, an actor-critic method learns an approximation of the value function, similar to what a policy-gradient method can do. However, the approximated value function is also used in actor-critic methods for bootstrapping: the return is formulated in part from an existing approximation of the value. Actor-critic formulations are often preferred in state-of-the-art applications of reinforcement learning to a variety of problems, thanks to their wide applicability and convergence properties [118, 119]. State-of-the art actor-critic algorithms also incorporate additional features for improving policy convergence properties and performance. For example, multiple agents can be simultaneously leveraged to generate experiences in an asynchronous approach following either identical, or distinct policies. With an asynchronous implementation, the policy can then be corrected by sampling from the generated experiences, typically generating improved performances and convergence properties. When combined with a formulation using advantage function in the update rule, as in Eq. (4.20), these algorithms are called asynchronous advantage actor-critic (A3C) [57, 120].

4.2.3 Trust Region and Proximal Policy Optimization

Basic policy-gradient or actor-critic methods based on policy update rules generally allow to train policies with good performances. However, the update rule might lead to unstable behavior for on-policy methods. Indeed, the formulation of the update in Eq. (4.20) cannot prevent from large variations of the policy parameters $\boldsymbol{\theta}$, especially when the same experiences are reused in multiple batches. Two state-of-the-art solutions have been recently presented that counteract the destabilizing behavior of regular policy gradient approaches.

Trust Region Policy Optimization

Trust region policy optimization (TRPO) is an approach presented by Schulman et al. that reformulates the optimization problem for the policy parameters $\boldsymbol{\theta}$ as a constrained optimization. TRPO is an on-policy method, that can be used in either discrete or continuous action spaces. In particular, the problem is outlined as:

$$\boldsymbol{\theta} = \arg\max_{\boldsymbol{\theta}} \mathscr{L}^{\text{TRPO}}(\boldsymbol{\theta}, \boldsymbol{\theta}_{\text{old}}) \quad \text{subject to} \quad \tilde{D}_{\text{KL}}(\boldsymbol{\theta}, \boldsymbol{\theta}_{\text{old}}) \leq \delta \tag{4.23}$$

where $\boldsymbol{\theta}$ and $\boldsymbol{\theta}_{\text{old}}$ represent the current and old values of the parameters of the policy, $\delta \in \mathbb{R}$ is a small boundary number defining the trust region, $\mathscr{L}(\boldsymbol{\theta}, \boldsymbol{\theta}_{\text{old}})$ is a surrogate advantage cost function, and $\tilde{D}_{\text{KL}}(\boldsymbol{\theta}, \boldsymbol{\theta}_{\text{old}})$ represents an average KL-divergence. In particular, the surrogate cost function in Eq. (4.23) is expressed as [121, 122]:

$$\mathscr{L}^{\text{TRPO}}(\boldsymbol{\theta}, \boldsymbol{\theta}_{\text{old}}) = \hat{\mathbb{E}}\left[\frac{\pi(\boldsymbol{a}_t|\boldsymbol{s}_t; \boldsymbol{\theta})}{\pi(\boldsymbol{a}_t|\boldsymbol{s}_t; \boldsymbol{\theta}_{\text{old}})}\hat{A}_t\right] \tag{4.24}$$

representing the estimated expectation over a ratio of two policies, scaled by an estimated advantage function. The estimated expectation $\hat{\mathbb{E}}[\cdot]$ is leveraged in a stochastic gradient descent approach to approximate the real expectation $\mathbb{E}[\cdot]$ with the sampled experiences. The two policies appearing in the objective function formulation in Eq. (4.24) are associated with distinct iterations of the same parametrization, $\boldsymbol{\theta}$ and $\boldsymbol{\theta}_{\text{old}}$. Ultimately, the advantage function is approximated with the term $\hat{A}_t \approx A(\boldsymbol{a}_t, \boldsymbol{s}_t)$, since the estimated value-function is leveraged. The surrogate cost function represents a measure of how the current policy performs with respect to the old policy.

The constraint in the optimization problem formulation in Eq. (4.23) is expressed as:

$$\tilde{D}_{\text{KL}}(\boldsymbol{\theta}, \boldsymbol{\theta}_{\text{old}}) = \hat{\mathbb{E}}\left[D_{\text{KL}}(\pi(\cdot|\boldsymbol{s}_t; \boldsymbol{\theta}_{\text{old}}), \pi(\cdot|\boldsymbol{s}_t; \boldsymbol{\theta}))\right] \tag{4.25}$$

where D_{KL} represents the Kullback-Leibler divergence, a measure of the difference between distributions based on the entropy. The nonlinear inequality constraint provides a boundary on the possible update of the parameters $\boldsymbol{\theta}$, by constraining the distance between the policy before and after the update. The constrained optimization in Eq. (4.23) is solved by Schulman et al. leveraging a first order expansion of the surrogate cost function and a quadratic approximation of the constraint in a neighborhood of $\boldsymbol{\theta}_{\mathrm{old}}$. It can be demonstrated that the gradient of the surrogate cost function is identical to the policy gradient in Eq. (4.18). Using the Lagrange form of the optimization problem in Eq. (4.23), and the conjugate gradient algorithm, TRPO provides an update rule for the parameters $\boldsymbol{\theta}$ that prevents from destabilization of the policy [122].

Proximal Policy Optimization

Algorithms from the proximal policy optimization family are introduced by Schulman et al. to overcome the hurdle of defining the trust region value δ [117]. Among the proximal policy optimization algorithms, this investigation focuses on the clipped version, which is referred to as PPO throughout the manuscript. Specifically, the selected objective function overcomes the necessity of predefining a trust region by clipping the surrogate cost function as [123]:

$$\mathscr{L}^{\mathrm{CLIP}}(\boldsymbol{\theta}, \boldsymbol{\theta}_{\mathrm{old}}) = \hat{\mathbb{E}}\left[\min\left(\frac{\pi(\boldsymbol{a}_t|\boldsymbol{s}_t;\boldsymbol{\theta})}{\pi(\boldsymbol{a}_t|\boldsymbol{s}_t;\boldsymbol{\theta}_{\mathrm{old}})}\hat{A}_t, g(\epsilon, \hat{A}_t)\right)\right] \tag{4.26}$$

where ϵ represents the clipping factor and the function $g(\epsilon, \hat{A}_t)$ is [123]:

$$g(\epsilon, \hat{A}_t) = \begin{cases} (1+\epsilon)\hat{A}_t & \text{if } \hat{A}_t \geq 0 \\ (1-\epsilon)\hat{A}_t & \text{if } \hat{A}_t < 0 \end{cases} \tag{4.27}$$

With the formulation in Eqs. (4.26) and (4.27), the potential step along the gradient is clipped by $g(\epsilon, \hat{A}_t)$: if the advantage is positive ($\hat{A}_t \geq 0$), the policy ratio is clipped by the upper boundary $(1+\epsilon)$; if the advantage is negative ($\hat{A}_t < 0$), the policy ratio is clipped by the lower boundary $(1-\epsilon)$. In both cases the clipping formulation prevents the new policy to step too far from the old policy, eliminating dramatic changes in the parameters [123]. To aid convergence of the policy, PPO is implemented in this investigation in an A3C configuration, using an unconstrained maximization scheme. In particular, a composite loss function is constructed by adding three loss terms as:

$$\mathscr{L}^{\mathrm{PPO}}(\boldsymbol{\theta}, \boldsymbol{\chi}, \boldsymbol{\theta}_{\mathrm{old}}) = \mathscr{L}^{\mathrm{CLIP}}(\boldsymbol{\theta}, \boldsymbol{\chi}, \boldsymbol{\theta}_{\mathrm{old}}) - c_1 \mathscr{L}^{\mathrm{VF}}(\boldsymbol{\chi}) + c_2 \mathscr{L}^{\mathrm{S}}(\boldsymbol{\theta}) \tag{4.28}$$

where c_1 and c_2 represent two relative scaling factors, weighting the impact of each loss term. In particular, $\mathscr{L}^{\mathrm{CLIP}}(\boldsymbol{\theta}, \boldsymbol{\chi}, \boldsymbol{\theta}_{\mathrm{old}})$ represents the PPO loss function in Eq. (4.26), with the explicit dependence on the value network's parameters $\boldsymbol{\chi}$ from the estimated advantage, since an actor-critic configuration is used. The second loss term serves to improve the approximation of the value function throughout the learning process as:

$$\mathscr{L}^{\mathrm{VF}}(\boldsymbol{\chi}) = \hat{\mathbb{E}}\left[(v_\pi(\boldsymbol{s}_t; \boldsymbol{\chi}) - r_t)^2\right] \tag{4.29}$$

The last term in the composite loss used in this investigation for PPO is:

$$\mathscr{L}^{\mathrm{S}}(\boldsymbol{\theta}) = \hat{\mathbb{E}}\left[S[\pi(\cdot|\cdot\,; \boldsymbol{\theta})](\boldsymbol{s}_t)\right] \tag{4.30}$$

corresponding to the average sampled entropy computed on the experienced steps. This loss term encourages exploration of the policy. Ultimately, the generalized advantage estimation (GAE) formulation is leveraged in this investigation to compute the estimated advantage in Eq. (4.26) as:

$$A(\boldsymbol{s}_t, \boldsymbol{a}_t) \approx \hat{A}_t^\pi(\boldsymbol{s}_t, \boldsymbol{a}_t) = \sum_{\ell=0}^{N} (\gamma\lambda)^\ell \delta_{t+\ell} \tag{4.31}$$

where δ_t is the temporal difference, defined as $\delta_t = r_t - v_\pi(\boldsymbol{s}_t) + \gamma v_\pi(\boldsymbol{s}_{t+1})$, and λ is the GAE factor. The generalized advantage estimation allows to considerably decrease variance, maintaining a sufficient level of bias [124].

4.2.4 Transfer Learning

Transfer learning is leveraged to learn an optimal policy in a target environment, using the combined information associated with the same target environment, and a source environment [125]. Regular reinforcement learning can be conceived as a form of transfer learning without a source environment. Transfer learning is generally used for both supervised and reinforcement learning tasks [58, 125]. During the transfer learning process, the difference between the target and the source environments can be associated to any of the defining features of an MDP, comprising: state; action; reward; transition dynamics of the environment; and initial state set. In this investigation, two examples of transfer learning are outlined, where the target and source environments differ for: the dynamical model governing the state transition $p(\boldsymbol{s}^*, r|\boldsymbol{s}, \boldsymbol{a})$; the reward formulation $r(\boldsymbol{s}, \boldsymbol{a})$. Moreover, different knowledge can be transferred from the source to the target environments, including: the dynamics, the policy, the value. In this investigation, the approximations generated by the neural networks associated with the policy and the value are transferred between environments. This approach is often referred to as policy transfer, where the target agent initially follows a teacher policy that is pretrained with the source environment [125].

Chapter 5

Unsupervised Learning for Higher-Dimensional Poincaré Maps

Poincaré maps represent a fundamental tool leveraged during different phases of the trajectory design process [2,3,126]. A well-constructed Poincaré map can reduce the dimensionality of the problem, aiding the visualization of large trajectory design spaces. However, Poincaré maps can be challenging to investigate when the generating dynamical flow is high-dimensional. In these cases, a two-dimensional representation of the map might not capture the entire design space, therefore generating potential loss of information. Additionally, a spatial representation of the crossings of a higher-dimensional Poincaré map can suffer from phenomenon as data obscuration, that complicate the analysis process. Previous contributions to the problem of analysis and visualization of higher-dimensional Poincaré maps have focused on incorporating additional dimensions in the map visualization process, including analytical and user-defined separation criteria, and introducing constraints to reduce the dimensionality of the map [3, 7]. These approaches address the problem of trajectory uniqueness, although some datasets may still be challenging to analyze.

An alternative solution for visualization of a high-dimensional Poincaré map has been recently introduced by Bosanac, using a technique from unsupervised learning [13]. This method uses clustering to group trajectories generating crossings on a planar periapsis Poincaré map in a completely unsupervised approach. This technique is used to generate autonomous partitioning of trajectories based on geometrical similarity, without introducing additional constraints or augmenting the dimensionality of the visualization. Moreover, clustering allows the generation of representative solutions that can additionally reduce the visualization burden of high-dimensional spaces. This chapter builds upon the approach demonstrated by Bosanac, with a focus on using the unsupervised learning approach to autonomously partition high-dimensional Poincaré maps. Specifically, three distinct examples are presented, introducing different techniques for clustering very large datasets in a distributed data mining approach. First, an example of distributed clustering, leveraging cluster proximity in the phase space, is presented and analyzed in Sec. 5.1. The approach is refined in Sec. 5.2, using a technique from manifold learning to group similar trajectories and clusters across different dynamical models. Eventually, Sec. 5.3 focuses on using a clustering summary to demonstrate the capability of data-driven approaches to visually investigate natural transport mechanisms.

5.1 Distributed Clustering for Spatial Poincaré Maps

In this section, a methodology is presented to cluster a set of spatial trajectories based on geometrical similarity. A distributed clustering technique, inspired by tomography, is presented in Sec. 5.1.1 to reduce the required computational resources associated with clustering a large dataset. This procedure is demonstrated in Sec. 5.1.2 in the context of a spatial Poincaré map generated in the spatial autonomous Sun-Earth CR3BP. Eventually, the generated results are compared with a partitioned large dataset, clustered in a single batch.

5.1.1 Method Overview

To efficiently cluster a set of spatial trajectories based on geometrical similarity, a distributed clustering approach that is inspired by tomography is presented. First, distinct sets of initial conditions are defined on multiple hyperplanes, and the crossings of these trajectories with a Poincaré map are grouped into datasets. Each dataset of crossings is separately processed in a distributed approach to generate local models. Then, clusters across distinct partitions are compared, and aggregated based on their regions of existence in the phase space. The technical approach for implementing this clustering procedure is summarized as follows:

Partitioning the available design space: different sets of initial conditions are defined within the available design space, as a means to reduce the computational efforts of clustering a large dataset. Among the variety of existing approaches, initial conditions can be seeded at the intersection between a surface of section, defining a Poincaré map, and a set of additional hyperplanes. For trajectories exhibiting spatial motion in the CR3BP, possible definitions of these additional hyperplanes comprise sets of mutually orthogonal hyperplanes passing through the location of the smaller primary, such as $z = 0$, $y = 0$ and $x = 1 - \mu$. Alternative examples include a set of parallel but distinct hyperplanes.

Seeding initial conditions in each partition: sets of initial conditions are seeded on the intersections between a user-defined surface of section, leveraged to define the Poincaré map, and each of the additional hyperplanes corresponding to the different partitions. These initial conditions represent spacecraft states, leveraged in the next phases of this method to generate a set of trajectories. Additional constraints can be incorporated to facilitate the partitioning, reducing the dimensionality of the design space. For example, candidate surface of sections can correspond to perigee maps, defined as in Eqs. (3.30) and (3.31). Also, valid constraints can be introduced to limit the available design space: for trajectories in the Earth vicinity in the spatial CR3BP, initial conditions can be seeded between the L_1 and L_2 gateways using a similar procedure to the approach outlined in Sec. 3.7 for a planar map, with initial conditions at a fixed value of Jacobi constant, and with initial $\dot{z} = 0$. In the example presented in this investigation, a grid in the position space is defined to initialize a set of spacecraft states populating a periapsis map with initial $\dot{z} = 0$: when the initial conditions are seeded on a $x = 1 - \mu$ hyperplane, the grid is populated using N_y equally-spaced values in $[y_{\min}, y_{\max}]$ and N_z equally-spaced values in $[z_{\min}, z_{\max}]$; analogously, when a hyperplane $y = 0$ is used, N_x equally-spaced values in $[x_{\min}, x_{\max}]$ and N_z equally-spaced values in $[z_{\min}, z_{\max}]$ are used to seed the initial conditions. For each initial condition defined in configuration space, the magnitude of the velocity is generated as $v = \sqrt{2U - C_J}$. If v possesses a real value, the considered spacecraft location is associated to a point in configuration space residing within the limits of the zero velocity surface, at the specified energy value.

Eventually, the direction of the velocity vector is computed by leveraging the periapsis condition.

Generating the datasets: for the set of initial conditions populating the i-th partition, the associated trajectories are propagated in the Sun-Earth CR3BP until satisfying one of the following termination criteria: completing a total of N_{ret} apses with respect to the Earth, corresponding to $N_{\mathrm{ret}}/2$ subsequent perigees; passing within a distance of 10^{-5} from the Earth in dimensionless units; or passing through either the L_1 or L_2 gateways. Each trajectory generated from the i-th set and performing at least two apses is then converted to an M-dimensional feature vectors. To facilitate a geometrical comparison between arcs, the feature vectors are designed to approximate each trajectory by sampling the arc at specific locations. The designed sampling has to sufficiently retain the geometrical characteristics of the summarized arc, simultaneously allowing a computationally efficient processing of a dataset of feature vectors. Different approaches exist to extract information from trajectories and generate a dissimilarity measure between two sample arcs [127]: to enhance computational efficiency, this investigation samples trajectories at the apse locations, and a straightforward ℓ^2-norm is used to obtain a measure of geometrical difference between feature vectors. This approach is preferred for its computational efficiency, and the already confirmed applicability to a similar scenario in the planar Sun-Earth CR3BP by Bosanac [13]. Alternative and traditional approaches for spatiotemporal trajectory comparison might also encode the geometric difference between the summarized arcs, although requiring larger computational effort [127]. Alternative representations are the scope of future research. For the presented implementation, the trajectories are summarized via a feature vector that reflects a sequence of perigees and apogees, as outlined in Sec. 3.7. Inspired by the approach presented by Bosanac, each trajectory arc is encoded into a feature vector $\boldsymbol{t}_j$, including successive returns of the trajectory to an apsis Poincaré map about the secondary in the Sun-Earth CR3BP. Each feature vector is constructed as a sequence of scaled sub-feature vectors, each incorporating the information of a crossed apsis. In particular, a sample feature vector, encoding the information of a specific arc, is mathematically described as:

$$\boldsymbol{t}_j = \left[\boldsymbol{s}_{j,0},\, \boldsymbol{s}_{j,1},\, \ldots,\, \boldsymbol{s}_{j,N_{\text{ret}}+1}\right]^T \in \mathbb{R}^{14N_{\text{ret}}} \tag{5.1}$$

where the vector $\boldsymbol{s}_{j,k}$ encodes the information about the k-th apsis of the j-th trajectory relative to the Earth [13]. Each apsis vector $\boldsymbol{s}_{j,k}$ incorporates spatiotemporal information of the recorded apsis, and it is mathematically formulated as:

$$\boldsymbol{s}_{j,k} = \left[\tilde{t}_{j,k}, \tilde{x}_{j,k},\, \tilde{y}_{j,k},\, \tilde{z}_{j,k},\, \dot{\tilde{x}}_{j,k},\, \dot{\tilde{y}}_{j,k},\, \dot{\tilde{z}}_{j,k}\right]^T \in \mathbb{R}^7 \tag{5.2}$$

where the tilde operator $\tilde{(\cdot)}$ indicates the use of normalization, used to reduce ill-conditioning between the components of the feature vector, that are scaled within the range $[-1, 1]$. In particular, $\tilde{t}_{j,k}$ refers to the time of the recorded apsis from the start of the trajectory, and normalized by the total propagation time along the trajectory. Then, $\tilde{x}_{j,k}$, $\tilde{y}_{j,k}$, $\tilde{z}_{j,k}$, $\dot{\tilde{x}}_{j,k}$, $\dot{\tilde{y}}_{j,k}$, and $\dot{\tilde{z}}_{j,k}$ are the position and velocity components of the recorded apsis in the rotating frame and with respect to the Earth, normalized using the minimum and maximum value for each column. If an arc is prematurely terminated at the k-th apsis, a placeholder $\boldsymbol{s}_{i,o} = \pm\,[0,\,10,\,0,\,0,\,0,\,0,\,0]^T$ is used for each $k \leq o \leq N_{\text{ret}} + 1$, with positive sign if $o \bmod 2 = 1$. This placeholder is inspired by the work from Bosanac on the planar map in the Sun-Earth system, and it is used to generate sufficient differentiation from trajectories that do not present premature termination, and to maintain a constant feature vector length across the dataset [13]. Then, the set of trajectories converted in feature vectors is grouped to form a collection of datasets $\boldsymbol{T}_i$, $i \in \{1, 2, \ldots, p\}$, each associated with one partition and corresponding to a hyperplane.

Clustering each partition: each of the p generated partitions is individually clustered using HDBSCAN. The selected clustering algorithm provides an ideal approach for differentiating trajectories based on similarity due to the small to no a-priori knowledge of the trajectory space. However, the HDBSCAN algorithm is governed by two input parameters: the number of datapoints used to define the core distance, m_{pts}, and the minimum cluster size m_{clSize}. Moreover, the ℓ^2-norm is selected as a similarity measure between feature vectors. For this investigation, an optimal choice of parameters is assumed to generate a clustering result with: large $DBCV$, the validity index detailed in Sec. 4.1.1.2, and introduced by Moulavi et al. [16]; a low percentage of trajectories identified as noise; a moderate amount of clusters, avoiding either an excessively large or small amount of groups. The selection criteria are inspired by the work from Bosanac [13]. To reduce the user workload, the designed method seeks an optimal combination of input parameters for only one partition. Then, to ensure consistency across the dataset, the same combination is leveraged to cluster the remaining $p-1$ partitions.

Aggregate clusters across partitioned datasets: the clusters associated with each partition are used to identify a minimal set of unique clusters across the partitions. Aggregation of clusters across distinct partitions is performed by locating intersections between clusters in the phase space. If a cluster from one partition does not intersect a cluster of another partition, it is considered a standalone cluster. Noise points across partitions are merged into a single set.

The presented procedure is demonstrated in the context of a prograde periapsis Poincaré map describing spatial motion in the Sun-Earth CR3BP at a value of Jacobi constant $C_J = 3.00088$. The initial conditions are seeded in the Earth proximity, with initial $\dot{z} = 0$ to aid visualization and allow uniqueness of the representation for a three-dimensional projection. Eventually, the clustering result is validated through a visual comparison with a more computationally-intensive clustering result of the full dataset associated with the Poincaré map.

5.1.2 Clustering Spatial Maps in the Sun-Earth CR3BP at $C_J = 3.00088$

To demonstrate the presented approach, an example is investigated for clustering a set of trajectories initialized on a prograde periapsis map, and propagated in the Sun-Earth CR3BP at a value of Jacobi constant $C_J = 3.00088$. Three partitions of the dataset are generated by initializing spacecraft states at the intersections of the surface of section, used to define the map, with three additional hyperplanes. The first dataset corresponds to a set of perigees that are initialized on the plane of the primaries, mathematically formulated as $z = 0$: the trajectories propagated from perigees on this hyperplane remain in the plane of the primaries due to leveraged dynamical model. The second dataset is populated with initial perigees lying on the $y = 0$ hyperplane. The last dataset is populated with initial perigees located on the $x = 1 - \mu$ hyperplane. Moreover, the spacecraft states constructed on each hyperplane are constrained with an initial $\dot{z} = 0$. The initial perigees are seeded using: a number $N_x = 301$ of equally-spaced x-coordinates between the locations of L_1 and L_2; a number $N_y = 301$ of equally-spaced y-coordinates in the range $[-0.01, 0.01]$; and a number $N_z = 301$ of equally-spaced y-coordinates in the range $[-0.01, 0.01]$.

Dataset	Constraint	$[m_{\text{pts}},\, m_{\text{clSize}}]$	$\lvert \boldsymbol{T}_i \rvert$	N_{clusters}	$DBCV$	Noise level %
1	$z = 0$	[50, 100]	31544	14	0.20177	6.21
2	$y = 0$	[50, 100]	26108	25	0.10868	3.99
3	$x = 1 - \mu$	[50, 100]	18639	8	0.18575	0.22

Table 5.1: HDBSCAN parameters and results for each partition of the dataset used to generate a prograde periapsis map.

Trajectories in each partition are generated by propagating the associated initial perigees forward in time for up to six subsequent apses with respect to the Earth, in the Sun-Earth CR3BP. The intersections of the propagated trajectories with a periapsis map centered at the Earth are recorded, and used to populate the feature vectors. The feature vectors, summarizing the geometry of the different arcs, are grouped for each partition and form a collection of datasets $\boldsymbol{T}_i$, with $i \in \{1, 2, 3\}$. Next, HDBSCAN is used along with the input parameters listed in Table 5.1 to perform clustering on each of the individual partitions of dataset. The input parameters selected in Table 5.1 reflect relatively large values of $DBCV$, a moderate number of clusters, and a reduced percentage of datapoints evaluated as noise for each partition: these parameters are selected according to a clustering result of a similar planar periapsis map processed by Bosanac, and constructed in the Sun-Earth CR3BP at $C_J = 3.00088$ [13]. An extended parameter exploration may be performed in future research [13]. The cluster aggregation process is then applied to identify clusters across multiple partitions that mutually intersect in the phase space. After the cluster aggregation process, the initial perigees of the three distinct datasets are projected onto the configuration space and reported in Fig. 5.1. In this figure, perigees are colored in shades of blue and red depending on the assigned clusters: if two perigees are colored with the same hue, the associated trajectories are deemed geometrically similar by the presented approach leveraging HDBSCAN; analogously, if two perigees are represented with distinct colors, the associated trajectories are considered as geometrically dissimilar by the clustering aggregation approach. In the figure, the Earth is colored as a central gray circle, while the L_1 and L_2 equilibrium points are reported as magenta diamonds. Figure 5.1 supplies the trajectory analyst with valuable information on the regions of existence associated with solutions with similar geometries. Moreover, the figure highlights the variety of distinct geometries existing in a region near the Earth, in the Sun-Earth system at $C_J = 3.00088$. Note that the visualized representation is governed by the limited set of identified hyperplanes, the selected HDBSCAN input parameters, and the utilized grids for the initial conditions seeding. Different sets of selected parameters can result in distinct partitioning of the datasets and a dissimilar final clustering result. However, the clustering result would still summarize the solution space, and aid the human analyst.

The fidelity of the prograde periapsis map constructed in the Sun-Earth CR3BP at a value of Jacobi constant $C_J = 3.00088$, presented in Fig. 5.1, is increased by introducing several additional hyperplanes. Specifically, sets of initial conditions are initialized to lie on a set of planes that are identified on the configuration space by a fixed value of the y-coordinate. These hyperplanes are defined in the range $y \in \{-4, -3.5, -3, \ldots, 3, 4\} \times 10^{-3}$ in dimensionless units. The cluster aggregation process detailed in Sec. 5.1.1 is then iterated for this new sets of datapoints: trajectories are propagated from the generated initial conditions, converted into feature vectors, and used to construct datasets that are clustered with HDBSCAN, using the same values of input parameters presented in Table 5.1. After clustering the different datasets, the cluster aggregation step is applied to merge clusters mutually intersecting in the phase space. The initial perigees populating the resulting higher-fidelity Poincaré map are projected onto the configuration space, and displayed in Fig. 5.2. To mitigate the effect of data obscuration, only a fraction of the ag-

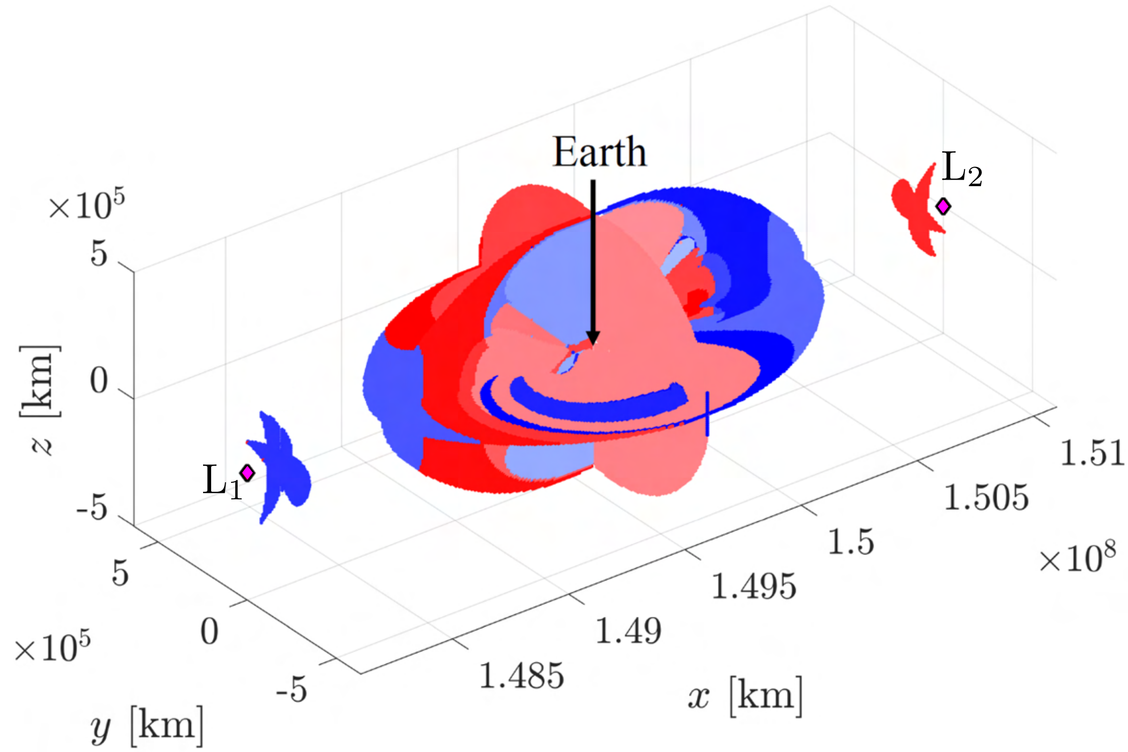

Figure 5.1: Poincaré map reflecting prograde periapses in the Sun-Earth CR3BP at a Jacobi constant $C_J = 3.00088$ and $\dot{z} = 0$ following the identified dataset partitioning and clustering aggregation process.

gregated clusters are visualized, reporting the initial perigees with markers colored in shades of blue and red, corresponding to the assigned cluster and consistent with Fig. 5.1. In the figure, the Earth is reported as a gray central circle, while the L_1 and L_2 equilibrium points are depicted with magenta diamonds. The presented cluster aggregation approach demonstrates the capability of aggregating clusters that intersect in the phase space, generating three-dimensional skeletons of groups of initial perigees, as displayed in Fig. 5.2. In the figure, four aggregated clusters are identified with numbers in the set $\{0, 1, 2, 3\}$, and the associated representative solutions, corresponding to the medoid computed according to Eq. (4.4), are displayed in Fig. 5.3 in the Sun-Earth rotating frame. In each frame populating the figure, green circles locate the initial conditions of each medoid, the Earth is identified by a gray central circle and red diamonds correspond to the L_1 and L_2 equilibrium points. The semi-transparent blue surface corresponds to the zero velocity surface. The representative solutions displayed in Fig. 5.3 exhibit different geometries, illustrating the capability of the clustering approach to group crossings of a map based on the geometry of the associated trajectories.

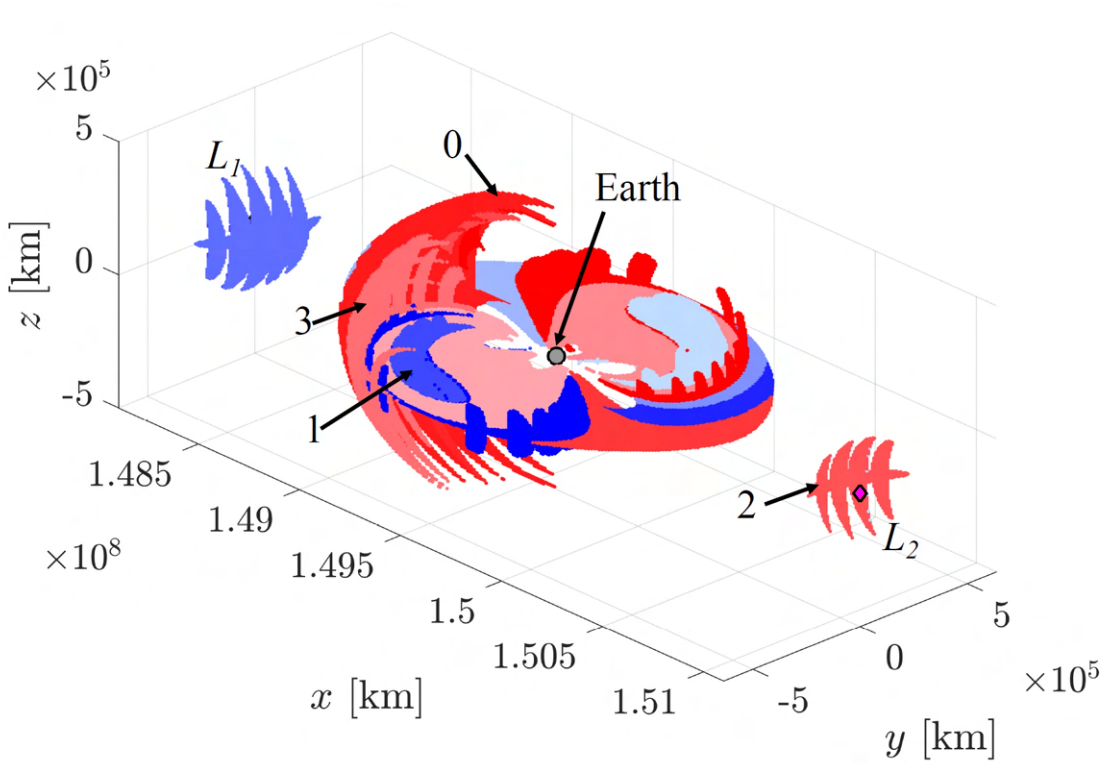

Figure 5.2: Poincaré map reflecting prograde periapses in the Sun-Earth CR3BP at a Jacobi constant $C_J = 3.00088$ and $\dot{z} = 0$ following the partitioning and cluster aggregation procedure, constructed using a large number of partitions. Four selected clusters are labeled.

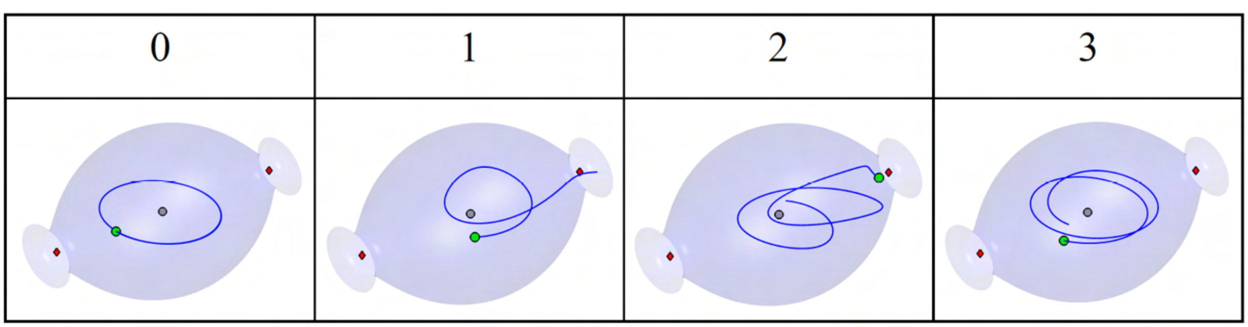

Figure 5.3: Representatives of the clusters labeled on the map in Fig. 5.2, plotted in the Sun-Earth rotating frame.

Ultimately, the clustering result generated with the distributed approach, presented in Fig. 5.2, is visually compared for validation with a clustering result processed over a single large dataset of feature vectors, composed by the entire large data associated with all initial conditions. For this large dataset, initial conditions, corresponding to prograde perigees, are defined by seeding 301 equally-spaced x-coordinates between L_1 and L_2, 301 equally-spaced y-coordinates in the range $[-0.01, 0.01]$, and 301 equally-spaced $z-$coordinates in the range $[-0.01, 0.01]$, with $\dot{z} = 0$. Using this grid, a complete dataset of $|\boldsymbol{T}| = 542,446$ spatial trajectories is generated, and clustered via HDBSCAN in a single batch. Using as input parameters for HDBSCAN $[m_{\text{pts}},\, m_{\text{clSize}}] = [200,\, 500]$, the clustering result presents 13 distinct groups of a trajectories, and a noise level of 0.2481%. Larger values of the input parameters for HDBSCAN are selected to balance the larger cardinality of the processed dataset. The clustering result is displayed in Fig. 5.4, with only a subset of the initial conditions plotted to prevent data obscuration, and colored by their cluster assignment. The same subsets of clusters identified in Fig. 5.2 are also labeled in Fig. 5.4. The overall structure of the clusters displayed in Fig. 5.4 is consistent with the results presented for the clustering aggregation process in Figs. 5.1 and 5.2, aside from the color differences due to the use of different coloring schemes. However, the amount of initialized spacecraft states of each partition for the distributed clustering approach is generally one order of magnitude lower than the single large batch of trajectories displayed in Fig. 5.4, with significant ramifications on the required computational resources. Furthermore, comparison of Figs. 5.2 and 5.4 reveals that both approaches recover clusters with perigees that cover similar regions of existence in the configuration space. However, the two approaches may not result in exactly the same amount of total clusters, since the clustering result is influenced by the input parameters selection and the properties of the dataset.

5.2 Clustering Maps across Different Dynamical Models

This general approach is extended to cluster a set of trajectories based on geometrical similarity across distinct dynamical models or at different values of the independent values, to assess cluster persistence. Associating clusters across distinct models can help a trajectory designer in

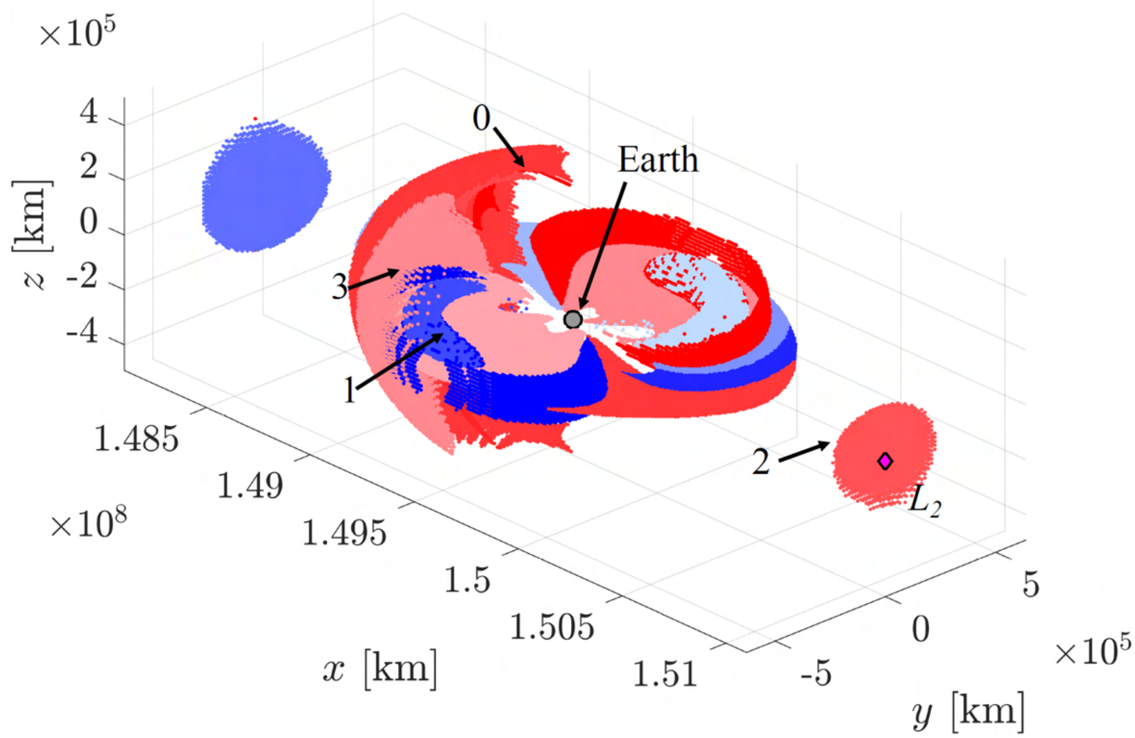

Figure 5.4: Poincaré map reflecting prograde periapses in the Sun-Earth system at a Jacobi constant $C_J = 3.00088$ and $\dot{z} = 0$ constructed by clustering the full dataset in a single step. The same four clusters identified in Fig. 5.2 are labeled.

understanding the evolution of the design space throughout the different model refinements of an end-to-end trajectory. Moreover, correlating clusters across distinct values of the independent variable can generate fundamental insights on the time evolution of the design space. First, the method is introduced in Sec. 5.2.1. The procedure is then demonstrated on a variety of examples in Sec. 5.2.2.

5.2.1 Method Overview

To aggregate clusters across different partitions and track the persistence of trajectories across distinct dynamical models and values of the independent time-like variable, distributed data mining is again useful. First, a collection of perigee maps is constructed, and clustered individually using HDBSCAN to reduce the computational complexity of processing a large dataset. Then, UMAP is used to re-assign datapoints classified as noise to a nearby cluster in the projected space. A number of datapoints is sampled from the clusters of each dataset, and used in a cluster aggregation process to generate a global summary result. Aggregated clusters across distinct maps correspond to trajectories that persist in their general geometry across various dynamical models, or across different independent variables. The procedure is structured in three steps, examined in the remaining of this section.

5.2.1.1 Constructing and Clustering Individual Perigee Maps

The investigation of cluster persistence across distinct dynamical models begins with the construction of a set of perigee maps across each model and for various parameters. To prevent the construction of very large datasets and provide an initial demonstration of cluster persistence across distinct models, a unique set of initial conditions is constructed: this set is defined using planar prograde perigees, presenting a value of the Jacobi constant $C_J = 3.00088$ in the Sun-Earth CR3BP [13, 128]. Once this set of initial conditions is constructed in the Sun-Earth CR3BP, the same spacecraft states are leveraged to generate trajectory arcs higher-fidelity dynamical models. Specifically, the set of initial conditions is populated with planar position vectors, seeded within the vicinity of the Earth in the Sun-Earth CR3BP: this investigation uses up to 401 equally-spaced x-coordinates between the locations of the Sun-Earth L_1 and L_2 equilibrium points, a total of 401 equally-spaced locations of the $y-$coordinate in the range $[-0.01, 0.01]$, and a constant value of the z-coordinate equal to 0. Once a position vector is identified, the magnitude of the velocity vector is computed similarly to the method presented in Sec. 5.1.1, and the periapsis map construction example described in Sec. 3.7.

With the constructed set of initial conditions, multiple datasets are generated for distinct dynamical model and different values of independent variables, where appropriate. For the dataset populated by trajectories generated in the CR3BP, the set of initial conditions, corresponding to prograde perigee, is propagated using the Sun-Earth CR3BP model. Each trajectory terminates when one of the termination criteria discussed in Sec. 5.1 is met. The propagated trajectories are then summarized in feature vectors, constructed to retain the geometrical features of the associated arc. Specifically, each arc is modeled as a feature vector composed as:

$$\boldsymbol{t}_i = \left[\boldsymbol{s}_{i,0},\, \boldsymbol{s}_{i,1},\, \ldots,\, \boldsymbol{s}_{i,k},\, \ldots,\, \boldsymbol{s}_{i,N_a}\right]^T \in \mathbb{R}^{5(N_{\text{ret}}+1)} \tag{5.3}$$

where each apse vector is formulated as:

$$\boldsymbol{s}_{i,k} = \left[\tilde{t}_{i,k}, \tilde{x}_{i,k},\, \tilde{y}_{i,k},\, \dot{\tilde{x}}_{i,k},\, \dot{\tilde{y}}_{i,k}\right]^T \in \mathbb{R}^5 \tag{5.4}$$

where the tilde operator $\tilde{(\cdot)}$ indicates the use of normalization, used to reduce ill-conditioning between the components of the feature vector, that are scaled within the range $[-1, 1]$. In particular, $\tilde{t}_{j,k}$ refers to the time of the recorded apsis from the start of the trajectory, and normalized by the total propagation time along the trajectory. Then, $\tilde{x}_{j,k}$, $\tilde{y}_{j,k}$, $\dot{\tilde{x}}_{j,k}$, and $\dot{\tilde{y}}_{j,k}$ are a subset of the position and velocity components of the recorded apsis in the rotating frame and with respect to the Earth, scaled by the distance between the Earth and L_2 and the maximum velocity within the dataset. Note that trajectories starting on the $z = 0$ plane do not exhibit out-of-plane motion in the low-fidelity model. However, for higher-fidelity models such as the point mass ephemeris, trajectories initialized on the plane of the primaries can exhibit a non negligible z-component along the propagated arc. For the majority of trajectories used in this investigation, the out-of-plane component is negligible, motivating the reduction of the apsis vector in Eq. (5.4). If an arc is prematurely terminated at the k-th apsis, a placeholder $\boldsymbol{s}_{i,o} = \pm\,[0,\, 10,\, 0,\, 0,\, 0]^T$ is used for each $k \leq o \leq N_{\text{ret}} + 1$, with positive sign if $o \bmod 2 = 1$ [13].

This procedure is leveraged to populate various datasets of trajectories constructed in higher-fidelity dynamical models. The complete dataset generated in the Sun-Earth CR3BP is composed of $|\boldsymbol{T}| = 31544$ map crossings, each associated with a 35-dimensional feature vector. A similar dataset of trajectories is populated in the Sun-Earth ER3BP, leveraging the same set of initial conditions, although propagated using the ER3BP equations of motions. This procedure is repeated multiple times for various initial values of the independent variable f_0, representing the initial true anomaly of the system. For datasets generated in the point mass ephemeris model, each initial condition that is originally expressed in the nondimensional rotating frame, centered at the Earth, is transformed into the GCRF, given a specified initial modified Julian date (MJD). After propagating the apses with the dynamics defined by the point mass ephemeris model, the apsis sequence calculated with respect to the Earth are recorded, and the spacecraft state at each apsis location is transformed into the rotating frame at the associated epoch, and scaled consistent with the formulations in Eqs. (5.3) and (5.4). This procedure is repeated multiple times for various values of the initial modified Julian date, governing the relative configuration in the Sun-Earth system.

After constructing each dataset, HDBSCAN is used to generate distinct clustering results. Each set $\boldsymbol{T}_i$ is clustered using a set of input parameters $[m_{\text{pts}}, m_{\text{clSize}}] = [200, 100]$: these parameters are maintained constant across distinct datasets to generate consistent partitioning. Larger values of input parameters for HDBSCAN are selected with respect to the previous approach detailed in Sec. 5.1.1 due to the more refined grid leveraged for the example discussed in this section. To demonstrate the results obtained with HDBSCAN, the dataset associated with trajectories generated in the Sun-Earth CR3BP at a Jacobi constant of $C_J = 3.00088$ is clustered and presented in Fig. 5.5(a). The clustering results in 13 distinct groups, projected onto the configuration space in dimensional coordinates and in the rotating Sun-Earth frame. In the figure, each cluster is identified by perigees that are colored in shades of blue and red, depending on the assigned cluster: if two perigees are colored with the same hue, the associated trajectories are deemed geometrically similar by HDBSCAN. The equilibrium points are displayed as red diamonds in this figure, while the zero velocity curves outline the gray shaded forbidden regions. For the dataset generated in the Sun-Earth CR3BP, HDBSCAN identifies 6.23% of the dataset as noise, indicated via black points in Fig. 5.5(a). Note that alternate feature vector formulations, or input parameters used by HDBSCAN, might generate distinct clustering results. The same set of initial perigees are utilized to construct a second dataset of trajectories, propagate in the Sun-Earth ER3BP and with an initial system anomaly of $f_0 = \pi/2$. After populating the dataset with the associated feature vectors, HDBSCAN is leveraged to generate a clustering result, depicted in Fig. 5.5(b). Note that, in this figure and similar maps constructed in higher-fidelity models, the equilibrium points and ZVCs calculated in the CR3BP are overlaid only to supply perspective. Analysis of Fig. 5.5(b)

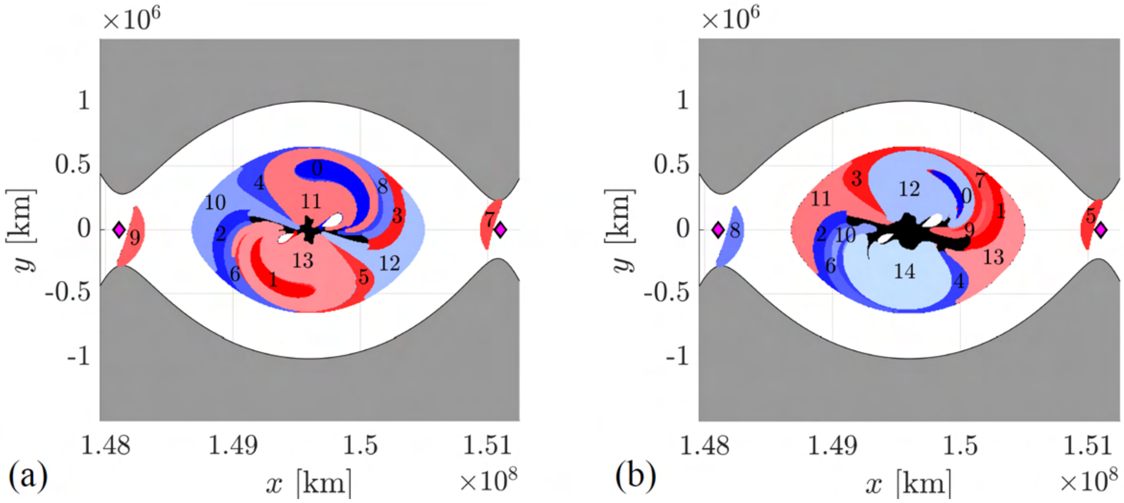

Figure 5.5: Clustered perigee maps near the Earth, constructed for the same set of initial conditions in: (a) the Sun-Earth CR3BP at $C_J = 3.00088$ and (b) the Sun-Earth ER3BP with $f_0 = \pi/2$.

reveals that this map, constructed with initial true anomaly $f_0 = \pi/2$ in the Sun-Earth ER3BP, admits clusters with similar regions of existence in the configuration space to those obtained for the CR3BP in Fig. 5.5(a), with some slight shifts and distortion. Note that perigees in Fig. 5.5(a-b) that are reported with identical hue are not associated with the same cluster. However, it is possible that some of the clusters in Fig. 5.5(b) correspond to trajectories with a similar geometry to those captured on the clustered map in Fig. 5.5(a).

5.2.1.2 Noise Reassignment with UMAP

Although HDBSCAN represents a valid algorithm for discovering clusters of distinct densities within a dataset, it may assign datapoints near the boundaries of clusters or in the sensitive regions close to the Earth as noise. Manifold learning algorithms as UMAP can be used to assign these noise points, after the clustering process, to nearby groups of geometrically similar solutions. For this step, the entire dataset of future vectors is projected by UMAP onto a lower-dimensional representation: noise points that join large areas on the projected space with datapoints assigned to clusters are relabeled. To provide an example of the noise reassignment procedure, consider the same perigee map constructed using the Sun-Earth ER3BP model with initial true anomaly

$f_0 = \pi/2$, as displayed in Fig. 5.5(b). The associated high-dimensional dataset $\boldsymbol{T}$, populated by the feature vectors extracted from the trajectories associated with each map crossing in this dynamical model, are projected onto a three-dimensional representation using UMAP. The input parameters are selected as $n_n = 200$ and $m_{\text{dist}} = 0.0$, producing a lower-dimensional representation that focuses on the global structure of the dataset. The projected dataset is then displayed in Fig. 5.6(a) in its three-dimensional representation, with each datapoint colored according to the associated cluster, consistent with Fig. 5.5(b). Moreover, group of projected points are identified with the same labels used in Fig. 5.5(b). Inspecting the three-dimensional projection, UMAP demonstrates a successful separation of the higher-dimensional dataset in a similar manner to the groupings discovered by HDBSCAN, despite not being provided with the labels generated by the clustering algorithm. However, the colored projection of UMAP presents a few significant differences. First, some clusters, as the group assigned to number 8, are split by UMAP and appear into distinct sub-groups in the projected embedding. Second, some larger clusters, such as clusters 12 and 14, appear bounded by a large set of points assigned as noise, and colored in the representation in black. Since the lower dimensional embedding constructed by UMAP preserves the global structure of the data, but not necessarily the density, the generated projection is leveraged to refine the clustering result, by reassigning those noise points appearing at the boundaries of a cluster to coincide with the nearby group. The reassignment procedure used in this investigation computes the distance of each point identified by HDBSCAN as noise to a maximum of 2000 randomly-selected points from each cluster in the lower-dimensional representation. Then, a noise point is reassigned to a specific cluster if the distance to any member in that cluster is lower than 0.5. Of course, this distance represents a user-selected parameter, depending on the used dataset and the final embedding by UMAP; however, using the distance computed in the embedded space generates a more robust approach than performing it in the original M-dimensional feature vector space. This approach is applied to the map generated with the Sun-Earth ER3BP model, and displayed in Fig. 5.6(b) with a consistent coloring scheme to what adopted in Fig. 5.5(b). Comparison between the maps in Fig. 5.5(b) and Fig. 5.6(b) reveal the fraction of noise within the dataset decreases from

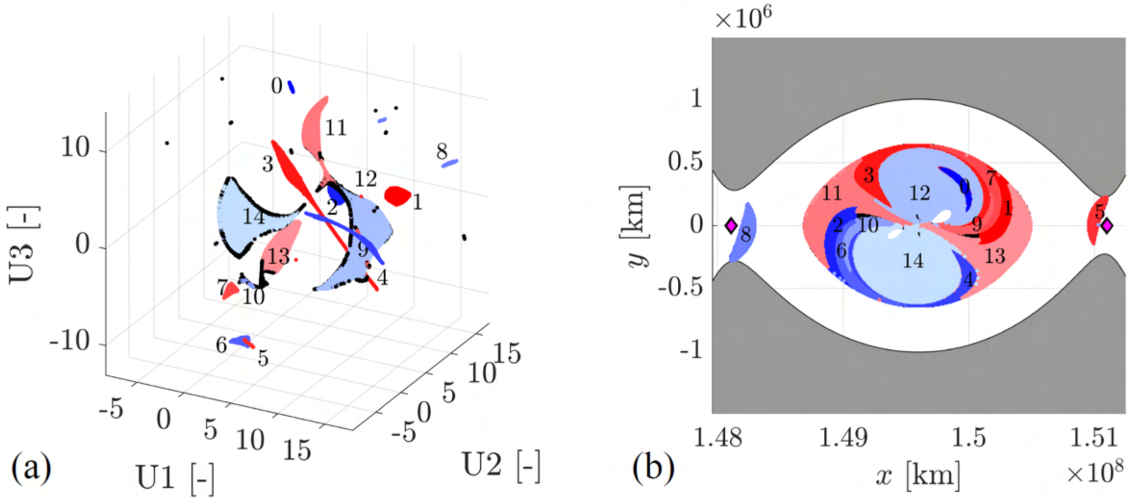

Figure 5.6: Noise reassignment applied to the map in Fig. 5.5(b), composed of trajectories generated in the ER3BP with $f_0 = \pi/2$: (a) UMAP projection of the pre-processed map and (b) post-processed map.

11.46% following application of HDBSCAN to 2.18% after noise reassignment. Visual inspection of Fig. 5.6(b) suggests that this result is reasonable: the noise points are reassigned to nearby clusters in the original M-dimensional dataset. Therefore, this approach is applied to each clustered map. The resulting maps are then used to perform cluster correlation and associate clusters of similar solutions across distinct dynamical models and values of the independent variable.

5.2.1.3 Cluster Aggregation Process

After partitioning each dataset of trajectories and reassigning boundary datapoints labeled as noise, a cluster aggregation procedure using UMAP is designed to correlate clusters of geometrically similar solutions across distinct maps. These maps can be generated leveraging different dynamical models, or using the same model but at different initial values of the independent variable. UMAP is used in the method to perform cluster correlation in the low-dimensional representation, preventing a less robust and challenging investigation in the full M-dimensional space due to the curse of dimensionality and the sensitivity of states within distinct regions of the phase space. Moreover, the projected representation constructed by UMAP preserves the relative distances between datapoints, placing members with similar feature vectors in nearby regions while separating dissimilar members.

Since the minimum distance between two datapoints m_{pts} on the UMAP projection represents a user-defined parameter, a distance-based cluster correlation procedure that is implemented in the lower-dimensional space is relatively straightforward to construct. This procedure is implemented as follows:

Generate, cluster, and reassign noise for each map: a collection of D distinct datasets, $\{\boldsymbol{T}_1, \boldsymbol{T}_2, \ldots, \boldsymbol{T}_D\}$ is constructed for each perigee map, as detailed in Sec. 5.2.1.1. These datasets can be generated using distinct dynamical models, or different sets of initial independent variables. Then, each dataset is clustered with HDBSCAN, using input parameters $m_{\text{pts}} = 200$ and $m_{\text{clSize}} = 100$, and resulting with a number of clusters G_i for each map, with $i \in \{1, 2, \ldots, D\}$. Eventually, the datasets are projected onto a three-dimensional manifold via UMAP, and noise points are reassigned, as discussed in Sec. 5.2.1.2.

Produce a global cluster summary: each cluster $\boldsymbol{S}_i^j$, with $j \in \mathbb{N}^+$, generated from each dataset $\boldsymbol{T}_i$ of feature vectors is sampled to produce a representation of the cluster with up to $n_{\text{sub}} = 300$ datapoints. The sampled feature vectors from the clusters of the considered maps are used to populate a single reduced global dataset $\boldsymbol{P}$. The sampled datapoints are identified from each cluster by leveraging the soft-clustering modification of HDBSCAN, available in the Python *hdbscan* clustering library. With soft-clustering, each datapoint is associated with a probability to join a certain cluster. In this investigation, soft-clustering is used to select up to n_{sub} datapoints that represent the associated cluster. A threshold probability $p > 0.8$ is used to sample datapoints of $\boldsymbol{S}_i^j$ and populate $\boldsymbol{P}$. For a generic cluster $\boldsymbol{S}_i^j$, if the amount of members satisfying this condition is larger than n_{sub}, these points are uniformly sampled, and a total of n_{sub} is used to populate $\boldsymbol{P}$. After sampling each cluster $\boldsymbol{S}_i^j$, the reduced dataset $\boldsymbol{P}$ is populated with a total of $\sum_{i=1}^{D} G_i$ clusters $\boldsymbol{Q}^j$, each represented by up to n_{sub} members.

Construct a lower-dimensional embedding of the global dataset: UMAP is leveraged to process the generated global dataset $\boldsymbol{P}$, obtaining a three-dimensional projection of the higher-dimensional dataset. The input parameters for the embedding are selected as $n_{\text{n}} = 100$ and

$m_{\text{dist}} = 0.0$ to construct a low-dimensional representation that preserves the global structure of the original dataset.

Aggregate clusters across distinct maps: the embedding generated from the global dataset $\boldsymbol{P}$ is automatically processed to correlate clusters of trajectories exhibiting geometrical similarity across distinct maps. The distance on the projected space of each members of $\boldsymbol{Q}^j$ from the members of all clusters in $\boldsymbol{P}$ is calculated, excluding points sampled from the same original map $\boldsymbol{T}_i$. For each point, the minimum constructed distance is recorded, and used to compute an average minimum distance between two distinct clusters $\boldsymbol{Q}^j$ and $\boldsymbol{Q}^k$ of the embedding of $\boldsymbol{P}$. If two clusters $\boldsymbol{Q}^j$ and $\boldsymbol{Q}^k$ possess a minimum average distance lower than a user-defined threshold value t_{avg}, the two clusters are correlated and are assigned the same cluster ID.

If two clusters are correlated, the associated trajectories are assumed to be geometrically similar, although generated in different dynamical models or at different values of the associated independent variable. Indeed, UMAP processes the input global cluster summary $\boldsymbol{P}$ with no knowledge of the different labels, but only using a summary of the geometries associated with the investigated trajectories. For this reason, fundamental knowledge on the evolution and persistence of trajectories of a specific geometry across distinct models can be inferred in a robust, data-driven approach. The devised method is applied to three different examples in the next sections, associated with cluster persistence across multiple dynamical models and values of the independent variable.

5.2.2 Cluster Persistence Across Dynamical Models

The designed data-driven procedure, leveraging distributed clustering for correlating trajectories exhibiting similar geometries across maps generated in distinct dynamical models, is demonstrated in this section in the Sun-Earth system. Using the same set of initial conditions generated in the CR3BP, several perigee maps are constructed using as dynamical models the Sun-Earth CR3BP, the Sun-Earth ER3BP and point mass ephemeris models. Each map is first associated to a dataset of feature vectors $\boldsymbol{T}_i$, clustered with HDBSCAN and processed to reduce the percentage of points assigned as noise via UMAP. Then, the cluster aggregation process detailed in Sec. 5.2.1 is applied to correlate clusters from distinct maps. Maps with correlated clusters are eventually visualized to investigate cluster persistence across distinct models and values of the independent variables. The approach is demonstrated with three examples studying the evolution of arcs, described by their geometry, across: 1) various values of the initial true anomaly f_0 in the ER3BP; 2) distinct initial epochs in the point mass ephemeris model; and 3) across dynamical models.

5.2.2.1 Cluster Persistence Across Initial True Anomalies in the Sun-Earth ER3BP

The set of initial perigees is leveraged to generate distinct sets of trajectories in the Sun-Earth ER3BP, each initialize at distinct values of the initial true anomaly f_0, and then combined in a data-driven approach to assess cluster persistence. Specifically, 10 different sets of trajectories are constructed using the following distinct values of the initial true anomaly of the primaries: $\boldsymbol{f}_0 = [0, \pi/6, 5\pi/24, \pi/4, 7\pi/24,\ \pi/3, \pi/2, 13\pi/24, 7\pi/12, 2\pi/3]$. After propagating the trajectories and populating the distinct datasets, the maps are clustered individually using HDBSCAN with the selected input parameters, as described in Sec. 5.2.1.1. Then, the noise reassignment technique presented in Sec. 5.2.1.2 is used to reduce the amount of noise for each partition. The clusters formed from these processed datasets are then sampled to form a reduced global dataset, projected onto a three-dimensional embedding via UMAP to form a global cluster summary, as described in Sec. 5.2.1.3. To automatically correlate clusters across distinct maps, a minimum average distance of $t_{\text{avg}} = 1.5$ is considered for clusters in the three-dimensional embedding: this value represents a user-defined and iteratively selected quantity, that can influence the cluster aggregation process. The adopted minimum distance threshold results in a global cluster summary composed of 26 unique clusters across the perigee maps constructed at the specified values of the initial true anomaly in the ER3BP. The aggregated clusters are presented in Fig. 5.7 for a subset of the leveraged maps in this example, each labeled in the top-left corner by the associated initial true anomaly for the primaries. In each frame of the presented figure, the maps are projected onto configuration space, and perigees are colored in shades of blue and red, depending on the assigned cluster: when two perigees are marked with the same hue, the associated trajectories are deemed as geometrically similar by the proposed clustering aggregation approach. Figure 5.7 also reports a gray arrow, indicating the increasing value of f_0 used to generate the trajectories associated with each map crossing. Moreover, each map reports the L_1 and L_2 equilibrium points and the ZVCs at a value of the Jacobi constant $C_J = 3.00088$: these features are displayed only for reference. Each frame in Fig. 5.7 reports labels on each of the generated clusters. To supply insights onto the geometrical distinction between arcs joining distinct clusters, the representative solutions of clusters 4 to 7 and 17 to 20 are plotted in Fig. 5.8, colored in shades of blue based on the originating initial system anomaly: darker shades are associated with lower values of f_0. In each frame of Fig. 5.8 the displayed trajectories start from the associated green markers, and are propagated for up to six subsequent apsides, the Earth is represented as a central gray circle, while L_1 and L_2 are indicated with magenta diamonds.

The maps in Fig. 5.7 are analyzed to assess the cluster persistence and evolution across distinct values of initial true anomaly in the Sun-Earth ER3BP. Specifically, the first two maps, corresponding to initial anomalies of $f_0 = 0$ and $f_0 = \pi/4$, admit a total of 12 clusters labeled from 0 to 11. The clusters populating these two maps present regions of existence that do not shift or distort significantly as the true anomaly is set equal to these two values. Among the identified clusters, the groups labeled from 0 to 9 exist within narrow regions of the configuration space, while clusters 10 and 11 encompass large regions. Recall that different feature vectors and input parameters may result in a distinct clustering result. In these analyzed frames, two small white lobes appear close to the Earth: these are associated with initial perigees that present trajectories that naturally escape the Earth vicinity before completing at least one apse and for this reason are excluded from the analyzed datasets.

The investigation progresses with the perigee map labeled with an initial true anomalies $f_0 = \pi/2$. In this map new clusters emerge in the regions of the configuration space previously occupied by clusters 10 and 11 at $f_0 = 0$ and $f_0 = \pi/2$. Some of the clusters, including groups 17, 19 and 20 are populated by trajectories that naturally escape the Earth vicinity through the

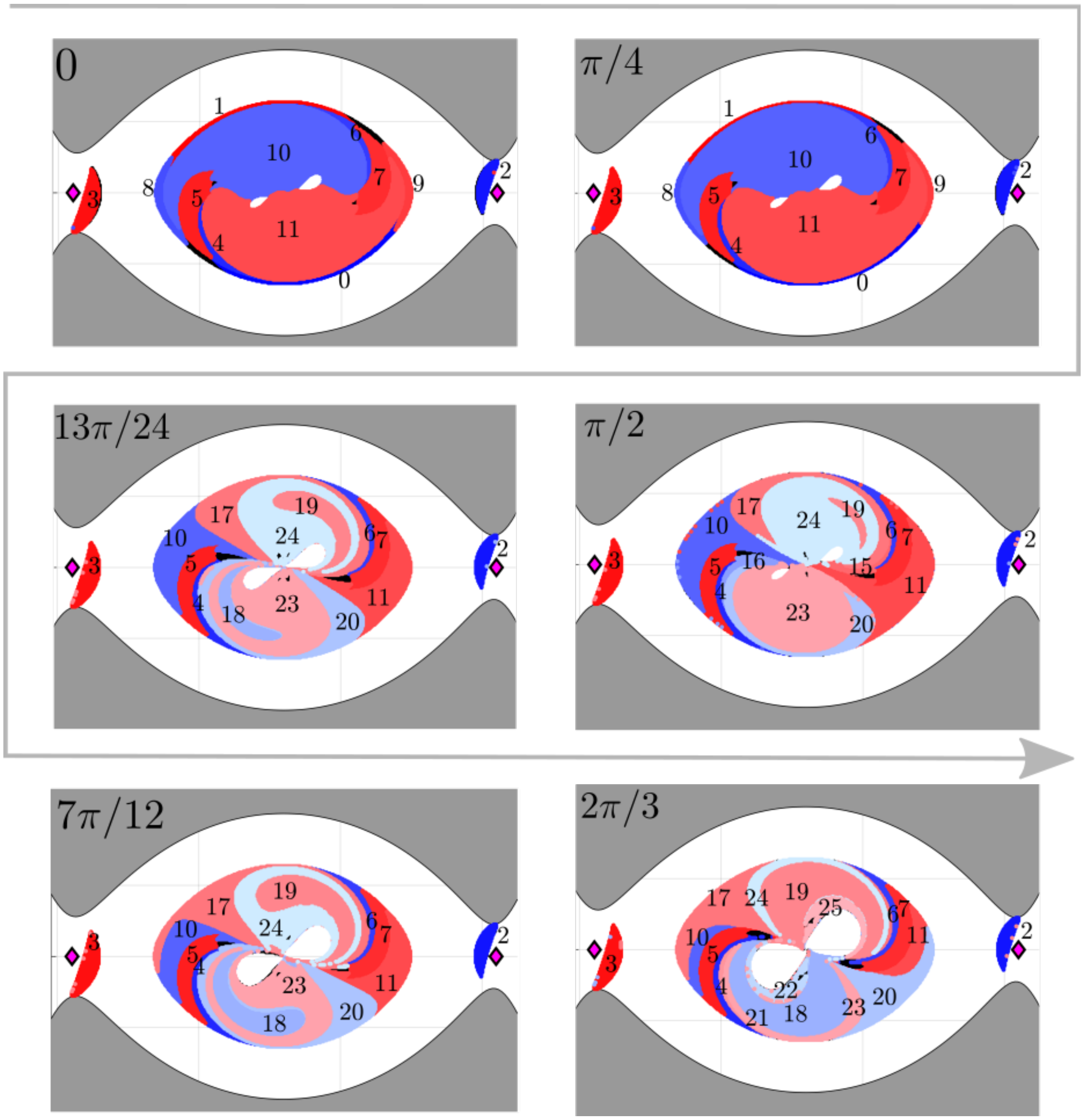

Figure 5.7: Clustered perigee maps for various values of the initial true anomaly in the Sun-Earth ER3BP. Clusters of similar solutions are correlated across maps.

x-coordinates corresponding to the L_1 and L_2 gateways in the Sun-Earth CR3BP, before completing a sequence of seven apses with respect to the Earth. As discussed in Sec. 5.1.2, the trajectories in these clusters, presented for some initial perigee in Fig. 5.8, are governed by the invariant manifolds from tori near L_1 and L_2 in the Sun-Earth CR3BP, as demonstrated in Sec. 5.4. Other subsets of clusters, such as groups 4 to 7, maintain their overall shape and size between these first three perigee maps. Similarly, the size of the small white lobes near the Earth presents no evident distortion

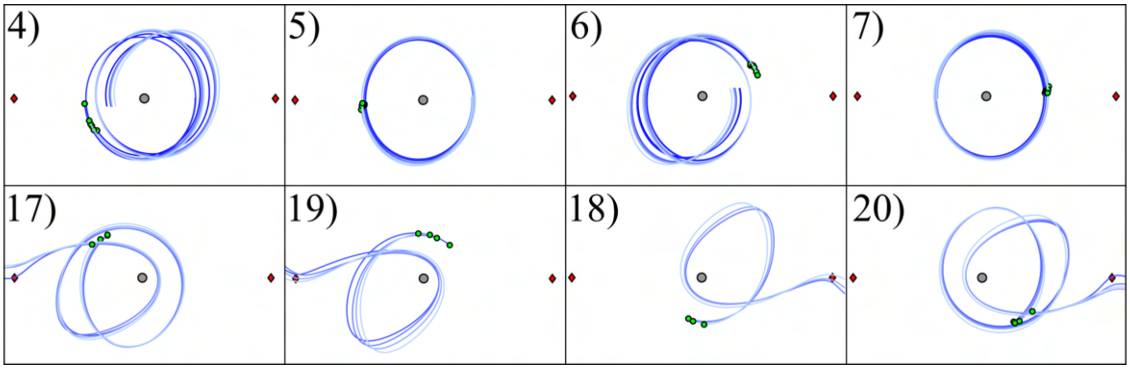

Figure 5.8: Selected cluster representatives from the maps in Fig. 5.7; darker trajectories are associated with maps constructed at lower values of the initial true anomaly.

with respect to the first two maps.

Rapid change in groups of solutions appear in the presented maps when the initial true anomaly increases beyond $f_0 = \pi/2$. Specifically, the map associated with initial true anomaly $f_0 = 13\pi/24$ presents larger region of clusters that are formed by trajectories naturally departing the Earth vicinity after 5 apses, such as clusters 17 and 20. Moreover, a new group of trajectories naturally transiting through the L_1 and L_2 bottlenecks appear in the map, labeled as cluster 18 and reported for some map in Fig. 5.8. This cluster presents an anti-symmetric configuration to cluster 19. Clusters 18 and 19, as depicted in Fig. 5.8, represent trajectories that resemble the stable manifold associated with the L_1 and L_2 Lyapunov orbits in the Sun-Earth CR3BP, as demonstrated by Bosanac [13]. However, cluster 18 emerges at a value of the true anomaly larger than $\pi/2$. For the remaining clusters, groups 10 and 11 continue to shrink across progressively increasing value of the true anomaly, while clusters 2 to 7 remain relatively unchanged. The white small lobes near the Earth encompass larger regions of the design space. This result reflects a larger amount of trajectories naturally departing from the Earth region as the true anomaly progresses.

The last two maps, presented at the bottom of Fig. 5.7, possess a significantly different cluster partitioning with respect to the maps constructed at lower values of the initial true anomaly. First, clusters 17 to 20, associated with trajectories naturally escaping the Earth region before recording seven apsides, encompass a larger region of the configuration space in the $f_0 = 7\pi/12$ map. When the initial true anomaly is increased further to $f_0 = 2\pi/3$, clusters 18 and 19 each split clusters 23 and 24, respectively, into two regions on the map. After the split, the cluster aggregation process identifies the emergence of new clusters, assigned to labels 22 and 25. Conversely, clusters 2 to 7 remain relatively unchanged throughout the different values of initial true anomaly. In addition, the central white lobes, corresponding to trajectories that naturally depart the Earth vicinity before completing one additional apsis, encompass progressively larger portions of the design space for these last two maps in Fig. 5.7.

From a global perspective, the presented data-driven approach enables the assessment of the persistence of groups of geometrically similar trajectories across various values of the initial true anomaly, supplying valuable insights into the evolution of the solution space. Throughout the maps presented in Fig. 5.8, distinct behaviors are evident. First, the initial true anomaly severely impact the size and shape of the clusters associated with escaping trajectories, for the set of initial conditions examined in this section. Similarly, regions of the design space associated with rapidly escaping trajectories increases. However, other groups of trajectories, as the clusters labeled with identifiers from 2 to 7, remain relatively unaffected by the changes in the independent variable presented in this example. Extending the presented example to encompass a wider range of initial true anomalies and a three-dimensional design space would likely result in a deeper understanding of the persistence of trajectories with a specific geometry in the ER3BP as this quantity is changed, and it is therefore left for future research.

5.2.2.2 Cluster Persistence Across Epochs in the Point Mass Ephemeris Model

A collection of perigee maps, constructed in a point mass ephemeris model of the Sun-Earth system, is generated for various values of initial epoch, clustered, and processed to assess cluster persistence. For this example, only the gravitational influence of the Sun and the Earth are considered within the point mass ephemeris equations of motion. The considered perigee maps are initialized at 8 distinct initial epochs, corresponding to $\boldsymbol{t}_0 = [29020, 29023, 29025, 29028, 29030, 29033, 29035, 2940]$ MJD. The initial dates span a total timeframe of 20 days, from Jun 19, 2020 at 12:00 pm to Jul

9, 2020 at 12:00 pm UTC. The used discretization in initial epochs is selected consistent with the sensitivity of the solution space in the point mass ephemeris model to changes in the initial epoch. After selecting the initial epochs, the datasets are populated: the same initial condition set employed in the CR3BP and ER3BP, corresponding to prograde perigees, is propagated forward in time in the point mass ephemeris model of the Sun-Earth system. Recall that this dynamical model includes the ephemerides of the Sun and Earth in their true orbits: thus, an initial condition with initial $z = \dot{z} = 0$ results in a propagated arc that exhibits small components of motion out of the plane of the primaries. Thus, the $[z, \dot{z}]$ components are neglected according to the feature vector formulation in Eq. (5.4), and are not considered in the presented implementation.

After the dataset construction, HDBSCAN is leveraged to cluster the different maps, while UMAP is used to reassign noise points populating the boundaries of each cluster in the three-dimensional projected space, as described in Secs. 5.2.1.1 and 5.2.1.2. Then, the cluster aggregation process described in Sec. 5.2.1.3 is implemented with a threshold distance of $t_{\text{avg}} = 1$ on a reduced global dataset in the point mass ephemeris model, selected via visual inspection of the generated representation. A subset of the generated maps is presented in Fig. 5.9, in a framework consistent with Fig. 5.7: six of the processed 8 maps are presented in distinct frames, labeled with their associated initial epoch, in the point mass ephemeris model of the Sun-Earth system. The gray arrow indicates the increasing initial epoch throughout the image. In each frame, perigees assigned to an identical cluster are reported with the same shade of blue or red, reflecting geometrical similarity of the associated trajectories. Moreover, the equilibrium points are displayed as magenta diamonds, while the ZVCs bounds the gray forbidden regions. Recall that the equilibrium points and the ZVC exist only in the CR3BP and reported in the figure to provide reference. Different perigees across distinct maps in Fig. 5.9 are reported in the same color, identifying solutions with a similar geometries at different initial epochs. To supply the cluster persistence, Fig. 5.10 displays the representative solutions for several clusters at each initial epoch over which the cluster exists, colored in shades of blue based on the initial date: darker shades are associated with low values of the initial epoch. In each frame of Fig. 5.10, the presented trajectories begin at the green markers,

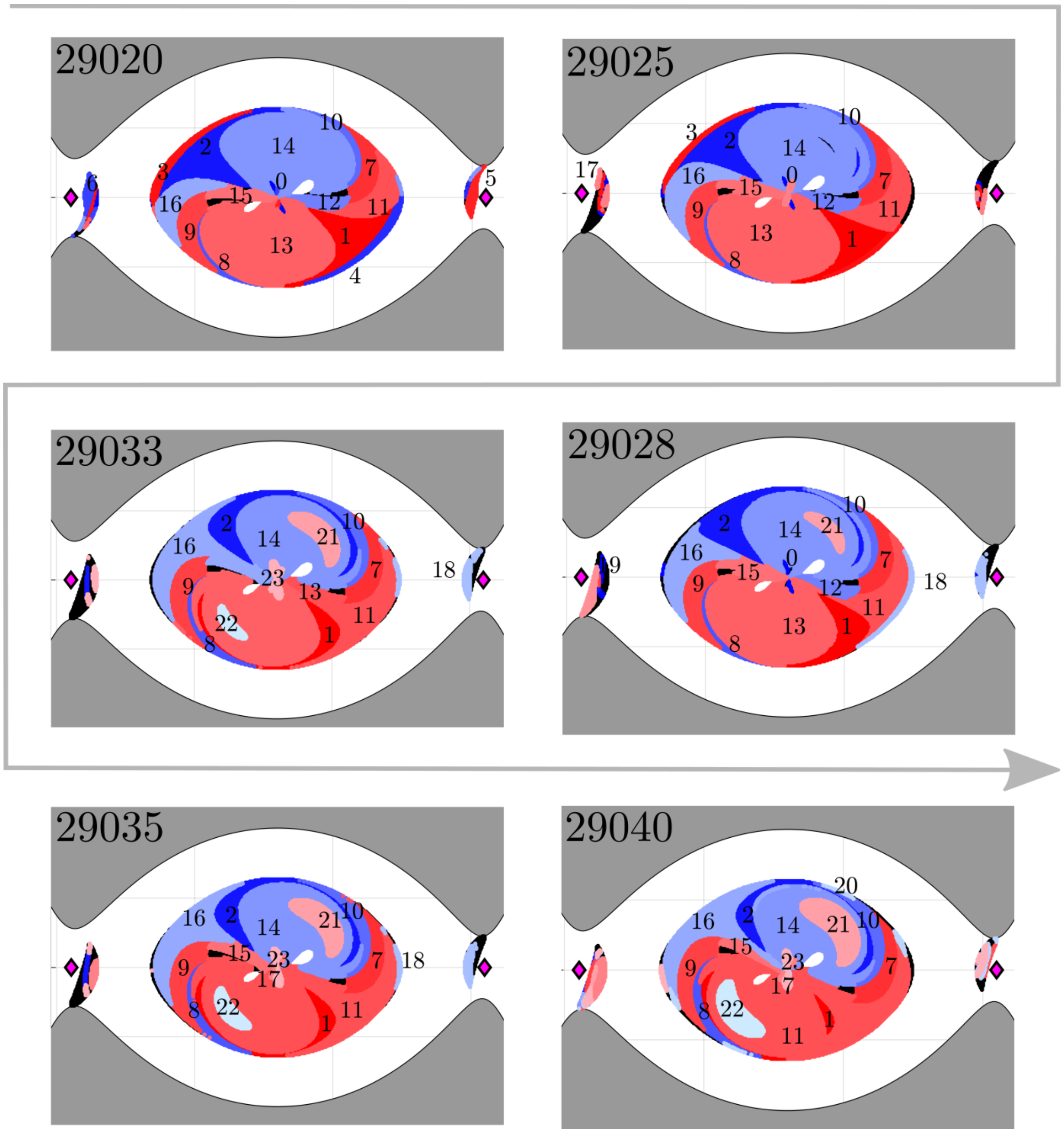

Figure 5.9: Clustered perigee maps for various values of initial epoch in the Sun-Earth point mass ephemeris model. Clusters of similar solutions are correlated across maps.

the Earth is represented as a gray circle, while L_1 and L_2 are indicated by magenta diamonds.

The collection of maps in Fig. 5.9 reports various clusters that exist across the entire range of initial epochs. These clusters appear to distort their shape when considering the epoch evolution determined by the gray arrow. For example, clusters 13 and 14 encompass large regions of the design space near the center of each perigee map. Clusters 7 to 10, located at the boundaries of

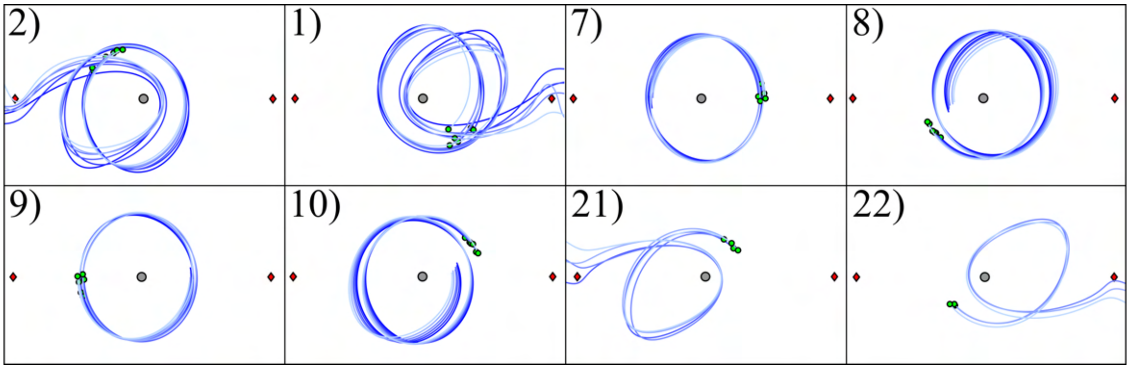

Figure 5.10: Selected cluster representatives from the maps in Fig. 5.9, propagated in a point mass ephemeris model; darker trajectories are associated with maps constructed at lower values of the initial epoch.

these clusters, tend to persist throughout the investigated range of epochs. The groups assigned with identifier 1 and 2 occupy a significant region of the perigee map at lower initial epochs, but they gradually shrink as the epoch increases: these clusters are composed by trajectories exhibiting natural departure from the Earth region after completing 5 apses, as reported in Fig. 5.10, resembling trajectories influenced by the invariant tori near L_1 and L_2 in the Sun-Earth CR3BP.

Other clusters only appear at specific ranges of the initial epochs, assuming identical sets of initial conditions for each depicted map. For example, cluster 21 emerges for an initial epoch of $t_0 = 29028$ MJD, persisting until $t_0 = 29040$ MJD. However, trajectories presenting similar geometry are already identifiable in the map with $t_0 = 29025$ MJD, although labeled as noise since their number is lower than the selected threshold of $m_{\text{clSize}} = 200$. Cluster 21 is composed of those trajectories naturally departing the Earth region before performing three apsides through the gateway at L_1. A similar example is presented by cluster 22, appearing at $t_0 = 29033$ MJD until the end of the investigated time frame. This cluster is composed by trajectories that naturally depart through the L_2 gateway. The regions of existence associated with clusters 21 and 22 expand with progressing the initial epoch t_0, resulting in a wider array of trajectories departing through the L_1 and L_2 gateways.

The set of perigee maps presented in Fig. 5.9 demonstrates the existence of regions of existence associated with distinct geometries in a point mass ephemeris model, and the persistence of such regions across different epochs. The generated clusters present different shapes and sizes across the investigated time frame: clusters 7 to 10, corresponding to bounded prograde motion, present relatively unchanged shapes; clusters associated with trajectories naturally departing the Earth region, as for clusters 1, 2, 21, and 22, present significant distortions, and their existence depends strongly on the initial epoch. Some trajectories pose significant challenge for HDBSCAN and the clustering aggregation process, under the selected initial condition discretization, input parameters, and feature vector definition. This is especially relevant for arcs associated with the map crossings near the Earth, at the boundaries of the central oval-shaped map, and near the Sun-Earth L_1 and L_2 points. Indeed, clusters 0, 5, 6, 9, 18, 17 and 23 are formed by a low number of trajectories, and rapidly change their shape throughout the different map representations. Note that the existence and time evolution of these clusters is governed by the employed discretization for the grid used to initialize the spacecraft states, and the subsequent UMAP correlation. A finer grid could improve the generated partitioning and better correlate this structures, at the expense of a significant adjunct computational effort.

5.2.2.3 Cluster Persistence Across Distinct Dynamical Models

The cluster aggregation process can also be employed to investigate cluster persistence across perigee maps generated in distinct dynamical models from the same initial condition set. This section demonstrates this approach by analysis the aggregation process across three datasets: a perigee map generated and clustered in the Sun-Earth CR3BP at $C_J = 3.00088$; a map constructed in the Sun-Earth ER3BP; and a map generated in the Sun-Earth point mass ephemeris model. These maps use the same set of initial conditions, constructed in the Sun-Earth CR3BP according to Sec. 5.2.1. Then, the cluster aggregation process is implemented with $t_{\text{avg}} = 1$, and the resulting partitioning are displayed in Fig. 5.11, each labeled by the dynamical model and the value of the independent variable. Specifically, the map presented in the bottom left is generated in the Sun-Earth ER3BP, using $f_0 = 7\pi/12$, while the map presented in the bottom right is constructed using the point mass ephemeris model, at an initial epoch $t_0 = 29035$ MJD. Figure 5.11 reveals useful insights concerning the geometry of trajectories propagated in distinct dynamical models. For example, clusters populated with trajectories naturally escaping from the Earth vicinity in forward time are successfully correlated across distinct maps. Specifically, clusters 0 and 1 group trajectories naturally departing the Earth vicinity before 3 apses, while clusters 4 and 5 group trajectories naturally escaping the L_1 and L_2 gateways after 5 apses. Other groups of trajectories, existing across each of the three dynamical models, are successfully correlated and numbered, for example clusters 2 and 3, 6 and 8, and 10 and 11.

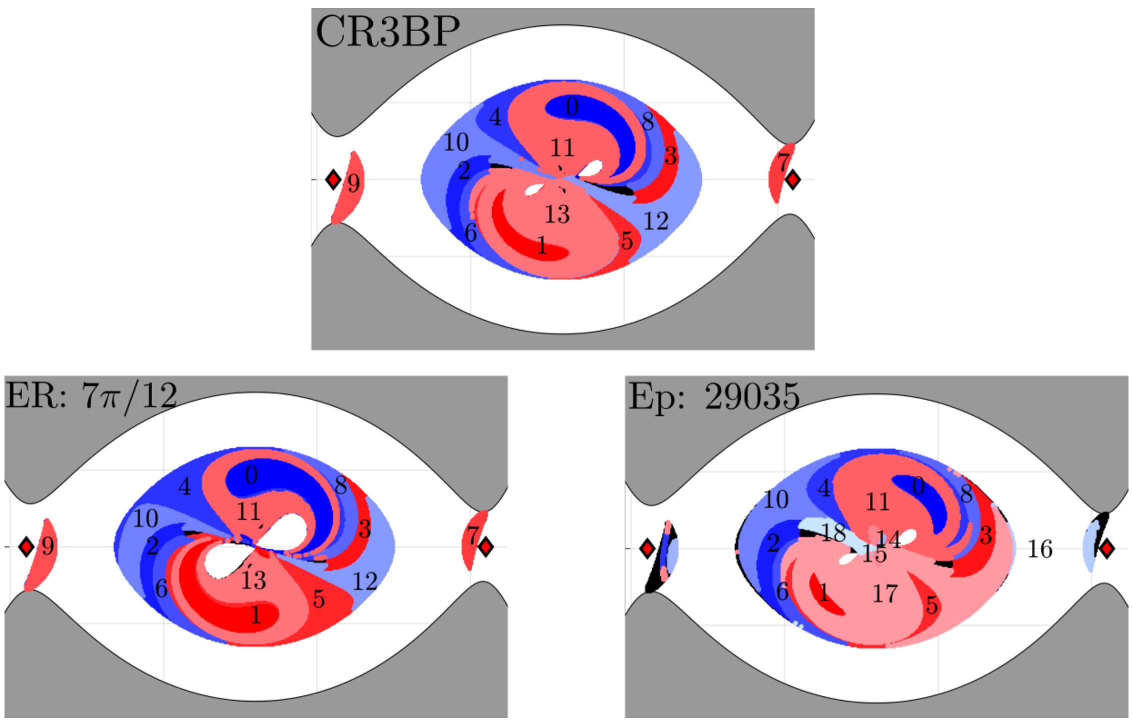

Figure 5.11: Clustered perigee maps for the same initial conditions in the: (top) CR3BP, (bottom-left) ER3BP at $f_0 = 7\pi/12$, (bottom-right) Sun-Earth point mass ephemeris model at $t_0 = 29035$ MJD. Clusters of similar solutions are correlated across maps.

Figure 5.11 provides relevant insights for trajectories that are correlated only for a subset of the generated maps. In particular, for negative values of y, some solution geometries persist between the CR3BP and ER3BP, but not in the ephemeris model. Consider clusters 12 and 13, existing in the CR3BP and ER3BP. In the ephemeris map, the same region onto configuration space that accommodates cluster 12 and 13 in the CR3BP and ER3BP maps, is instead occupied by cluster 17.

To analyze this aspect, Fig. 5.12 reports the UMAP projections of the reduced dataset $\boldsymbol{P}$, obtained from the three non-aggregated maps. In the projection, the sampled datapoints are colored and labeled according to the final clustering, consistent with Fig. 5.11. While many sampled datapoints appear as condensed and well-grouped cluster in the projected space, cluster 17 appear split in two halves. Two zoomed-in regions in the neighborhood of clusters 12 and 13 are displayed in the right part of the figure, highlighting the splitting of cluster 17 onto the projected space. The adopted color scheme reports the datapoints associated with cluster 12 in light-blue, the darker red shade to cluster 13, and the lighter red shade to cluster 17. From the zoom-in views, cluster 17 appears in regions of the projected space nearby datapoints within clusters 12 and 13. Therefore, clusters 12, 13 and 17 might reasonably be considered as composed of either 1 or 2 distinct clusters. However, the designed cluster correlation approach does not make this decision, likely due to the selected value of t_{avg}. A larger threshold distance value could potentially produce a different result, and address this issue.

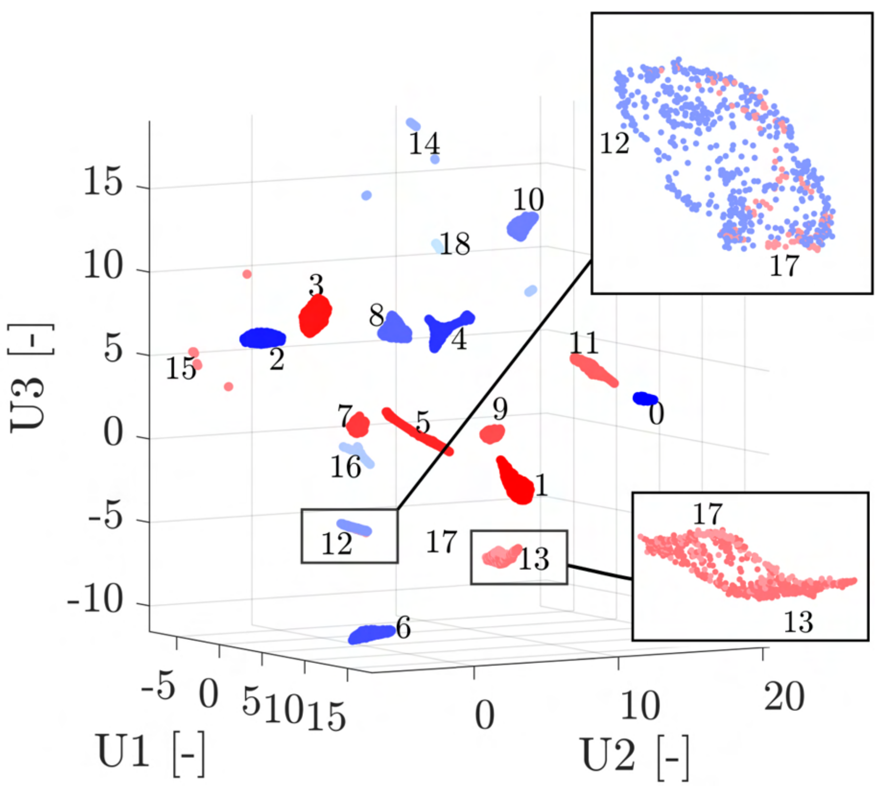

Figure 5.12: UMAP projection of the global reduced dataset presented in Fig. 5.11.

5.3 Examining Fundamental Transport Mechanisms

In this section, the data-driven clustering result is used to investigate the natural transport mechanisms governing spacecraft motion near the Earth. Indeed, the design space near the Earth in the Sun-Earth CR3BP at a value of Jacobi constant $C_J = 3.00088$ is governed by the stable and unstable manifolds of periodic and quasi-periodic solutions near the L_1 and L_2 gateways [1, 129]. Trajectories residing inside the boundaries of the stable and unstable manifolds of these solutions are associated with arcs naturally escaping from the Earth region. Likewise, arcs near the stable and unstable manifolds, that are not propagated for the needed time to experience natural departure, represent a valuable asset to generate insights into the dynamical characteristics of the flow near the Earth in the Sun-Earth system. These solutions are also leveraged in a variety of transfer scenarios in multi-body systems [5]. However, the representation of the four-dimensional crossings of stable and unstable manifolds with a well-defined Poincaré map can represent a challenging task. This issue might be mitigated by analyzing groups of trajectories along these hyperbolic manifolds.

One approach to rapidly evaluate the governing nature of the hyperbolic invariant manifolds emanated from the invariant tori near L_1 and L_2 is to project the arcs onto a data-driven summary of the solution space. This approach is demonstrated by Bonasera and Bosanac [130]. First, a global clustering result is formed that summarizes the available geometries in the design space near the Earth in the Sun-Earth CR3BP at $C_J = 3.00088$. This global clustering result is constructed using an alternative method that represents a more robust generalization of the approach presented in Sec. 5.1.1, and it is primarily developed by Bosanac within a collaboration between Bonasera and Bosanac [130]. This global clustering result is leveraged in this investigation to generate insights on the hyperbolic invariant manifolds emanated by the tori near the L_1 and L_2 gateways. Specifically, trajectories along the stable manifolds of tori near the gateways in the Sun-Earth CR3BP are constructed and projected onto the global clustering result, as presented in Sec. 5.3.1. Ultimately, the projected manifold arcs are compared with clusters contained in the global clustering result in Sec. 5.3.2. This example serves as an additional demonstration of the benefit of data-driven approaches to autonomously group trajectories from unseen datasets based on geometrical similarity, aiding a trajectory designer to understand natural transport mechanisms and to support initial guess construction. The results presented in this section are first published in Celestial Mechanics and Dynamical Astronomy, 133, 51 (2021) by Springer Nature [130].

5.3.1 Projecting Hyperbolic Manifold Arcs onto the Global Clustering Result

To exploit the clustering result to summarize trajectories in an unseen dataset, a k-weighted nearest neighbor classifier is trained to learn a mapping from a dataset of trajectories to a cluster identifier. In particular, the M-dimensional feature vectors of a global clustering result generated in the paper by Bonasera and Bosanac are used as input to the classifier, while the associated cluster, identified with an integer number, corresponds to the label or ground truth [130]. It is assumed that the final dataset, generated by an iterative sampling process, is sufficiently representative of the entire dataset of trajectories over the initial partitions. The classifier is configured to compare each datapoint to its 10 neighbor solutions, using the Euclidean norm to generate a dissimilarity measure between inputs, and weighted by the square of the distance. A five-fold cross-validation is applied to improve the performance of the classifier, given the potentially low amount of samples in the final dataset. Ultimately, the classifier is used to assign each of the feature vectors in the original partitions to one of the groups in the global clustering result.

A dataset of trajectories along the hyperbolic stable manifold arcs is generated and assigned to distinct clusters using a global clustering result obtained with spatial trajectories sampled in a neighborhood of the Earth, in the Sun-Earth CR3BP at a value of Jacobi constant $C_J = 3.00088$. The generation of the dataset begins with the computation of the periodic orbits near the L_1 and the L_2 gateway. Specifically, at a Jacobi constant value $C_J = 3.00088$, the Lyapunov and vertical periodic orbits near the gateways are computed using the approach detailed in Sec. 3.2: at the selected level of energy, members from the halo and axial orbit families do not exist. The selected orbits are depicted in Fig. 5.13. In the central subfigure, the Earth is depicted with a gray circle, while the L_1 and L_2 points are presented with magenta diamonds. The constructed Lyapunov and vertical orbits are visualized in the Sun-Earth rotating frame with black and copper lines, respectively, and zoomed-in in two frames located at the margins of the figure. Segments of the zero velocity surface in a neighborhood of the L_1 and L_2 equilibrium points are depicted using semi-transparent blue surfaces, surrounding the depicted periodic orbits at the bottlenecks. The orbits present non-empty hyperbolic and central manifolds, indicating the existence of near quasi-periodic trajectories foliating unstable 2-tori. Dynamical characteristics of the periodic orbits are listed in Table 5.2, reporting: periods and initial conditions in nondimensional units, and the eigenvalues associated with the stable and unstable manifolds of the monodromy matrices.

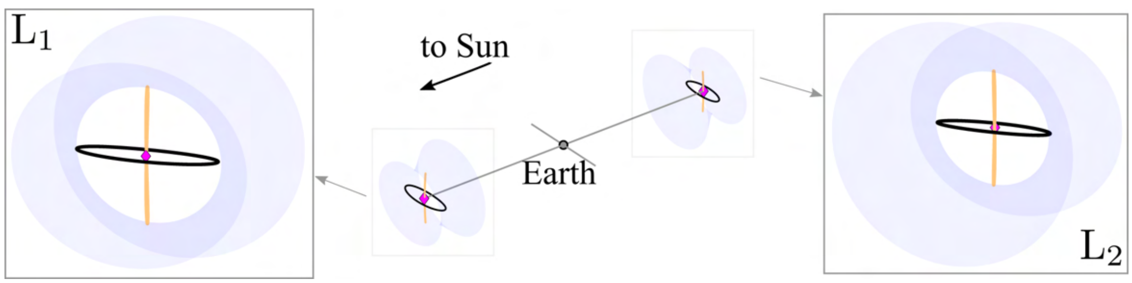

Figure 5.13: Periodic orbits near L_1 and L_2 in the Sun-Earth CR3BP at $C_J = 3.00088$.

	Orbit	Period [-]	$\boldsymbol{x}_0$ [-]	Eigenvalue
L_1	Lyap	3.0189495	$[0.9895177, 0, 0, 0, 0.0036028, 0]$	$\lambda_S = 5.00 \times 10^{-4}$, $\lambda_U = 2004$
	Vert	3.1247260	$[0.990063, 0, 0, 0, 0.000073, 0.003271]$	$\lambda_S = 3.81 \times 10^{-4}$, $\lambda_U = 2626$
L_2	Lyap	3.0588881	$[1.0095682, 0, 0, 0, 0.0029462, 0]$	$\lambda_S = 5.14 \times 10^{-4}$, $\lambda_U = 1946$
	Vert	3.1691039	$[1.010011, 0, 0, 0, -0.000047, 0.002587]$	$\lambda_S = 3.90 \times 10^{-4}$, $\lambda_U = 2563$

Table 5.2: Period, initial state and stable and unstable eigenvalues, λ_S and λ_U, respectively, of the monodromy matrix for the Lyapunov and vertical orbits at L_1 and L_2 in the Sun-Earth CR3BP at $C_J = 3.00088$.

One-parameter families of invariant 2-tori near the Lyapunov and vertical orbits are generated following the approach detailed in Sec. 3.5. Specifically, the Lyapunov orbit is initially leveraged to generate a near invariant 2-torus, enforcing the value of the Jacobi constant to $C_J = 3.00088$ across each point of the approximation of the first invariant curve. Then, the generated torus is used in a continuation approach with pseudo-arclength to retrieve more members of the families of invariant 2-tori near L_1 and L_2. The computed families of tori are bounded by the Lyapunov and the vertical orbits [25]. Both families are approximated using 50 members, leveraged in the later steps of the approach to generate arcs along the stable hyperbolic manifolds. A subset of the selected 50 members of invariant tori is depicted in Fig. 5.14 in the Sun-Earth rotating frame and with colors consistent with Fig. 5.13 and associated with the maximum z-excursion of the solution: invariant tori associated with small out-of-plane displacement resembles Lyapunov orbit, and are therefore colored in shades of black, while tori presenting large out-of-plane amplitude lie close to the vertical orbits and are therefore colored in shades of copper. In the same figure, segments of the ZVS are displayed using semi-transparent blue curves to provide a dimensional reference.

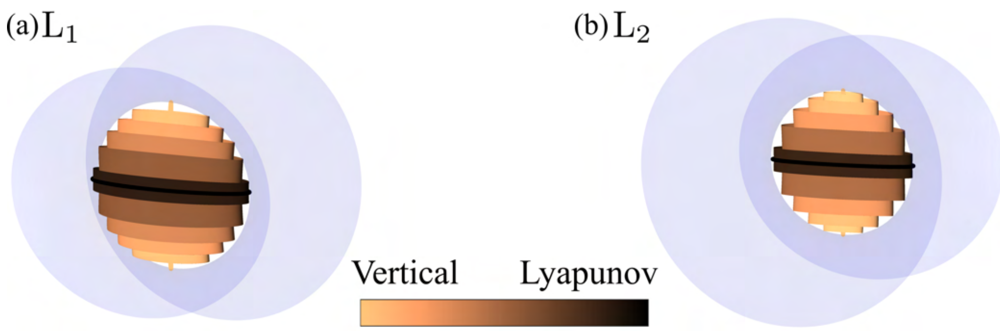

Figure 5.14: Representative members of families of invariant 2-tori near (a) L_1 and (b) L_2 at $C_J = 3.00088$ in the Sun-Earth CR3BP.

Application of the stability analysis detailed in Sec. 3.5 confirms the generated families of tori inherit the unstable nature of the bounding Lyapunov and vertical orbits. Therefore, arcs can be generated from these structures to approximate the stable hyperbolic manifolds. To generate an approximation of the global stable manifolds, 12,525 states along each torus are perturbed along the direction of the local stable eigenvalue of the differential of the torus invariance condition. These perturbed states populate a set of initial conditions that are propagated backward in time to generate arcs lying along the hyperbolic stable manifolds of the constructed tori. Up to four consecutive perigees per each manifold arc are recorded, all presenting values of θ in the approximate range $[-22.2°, 22.2°]$. These perigees are used as initial conditions for trajectories lying on the stable manifold arcs: indeed, propagating forward in time the recovered perigees, stable manifold arcs are generated that asymptotically converge towards the originating invariant 2-torus. Propagating forward in time these perigees along the stable manifold allows to generate a dataset of trajectories that is consistent with the analyzed dataset detailed in the collaborative work of Bonasera and Bosanac [130]: indeed, these trajectories are propagated for up to three subsequent perigees, using the same set of termination criteria. The constructed set of trajectories propagated forward in time is used to populate a dataset of 35-dimensional feature vectors, reflecting the geometry of the associated arcs via the sequence of apse [130].

Eventually, the generated dataset of feature vectors, representing arcs along the hyperbolic stable manifolds of the tori near L_1 and L_2, are assigned to distinct clusters. The classifier is then used to process the constructed feature vectors of the manifold arcs, and generate a set of labels for each arc [130]. Autonomous cluster identification using the classifier prevents the trajectory designer from a challenging and time-consuming manual classification of a large set of manifold trajectories based on geometrical similarities. The classified set of trajectories is used in the following subsections to investigate the geometry of the manifold arcs, and generate insights on the governing nature of the invariant manifolds near L_1 and L_2.

5.3.2 Analyzing Clusters Governed by Stable Manifolds near L_1 and L_2

The characteristics of arcs along the stable manifolds of invariant 2-tori at L_1 and L_2 in the Sun-Earth system at $C_J = 3.00088$ is investigated in a data-driven approach to generate insights on the governing nature of these structures onto the design space near the Earth. Specifically, the generated stable manifold arcs are divided into two subsets, corresponding to trajectories experiencing natural escape through the L_1 and L_2 regions within three revolutions around the Earth, and arcs presenting bounded motion in the Earth region.

Figure 5.15 reflects the first subset of stable manifold arcs. Specifically, the center of the figure presents the projections onto the three-dimensional configuration space of a subset of four-dimensional groups of perigees from the global clustering result: these are composed of the clusters that the considered perigees of the stable manifold are assigned to by the classifier, colored in shades of blue and red. In the central figure, the map populated with perigees presenting $z = \theta = 0$ is overlaid with semi-transparent markers to provide visual reference. Insets are located at the four corners of Fig. 5.15, reporting the collection of perigees along the stable manifold arcs associated with trajectories that are assigned to the indicated clusters in the central figure. Each inset is labeled using the nomenclature $\mathbb{S}_{p,q}^{\mathrm{L}_i}$: $\mathbb{S}$ indicates that the perigees in the inset are associated to stable manifold arcs originating at the invariant tori near L_1 or L_2; the superscript L_i, with $i \in \{1, 2\}$ specifies the location of the tori generating the stable manifold arcs; the first subscript p indicates the number of returns to the perigee map experienced by the arcs originating at the perigees depicted in the inset, before approaching the torus in forward time; the second subscript q is used to differentiate the subsets of perigees that are associated to the same number of returns to the map, but are assigned by the classifier to distinct clusters. Moreover, each of the depicted insets reports: the collection of initial perigees of the stable manifold arcs that are assigned to the indicated cluster in the central figure, in shades from black to copper consistent with the originating tori in Fig. 5.14; a subset of the originating invariant 2-tori at the L_1 or L_2 bottleneck; a single arc along the stable manifolds, depicted in a shade from black to copper, consistent with the out-of-plane deviation of the originating invariant torus; the medoid of the indicated cluster, depicted in a shade of blue or red, consistent with the central figure; the Earth as a gray central circle and a portion of the ZVS centered at the tori structure near the L_1 or L_2 bottleneck, for reference.

The results depicted in Fig. 5.15 confirm the expected association between arcs experiencing early departure from the Earth region and stable manifold arcs originating from the governing invariant tori near the L_1 and L_2 equilibrium points. The initial perigees depicted in the insets in Fig. 5.15 are associated to the first and second returns to the perigee map of the stable manifold arcs emanated from the investigated tori, when propagated backward in time. The stable manifold arcs are located in almost identical regions of these structures, and are classified in the same clusters of trajectories experiencing early departure, highlighting the governing nature of the investigated invariant tori. Moreover, these observations are consistent with previous contributions from other

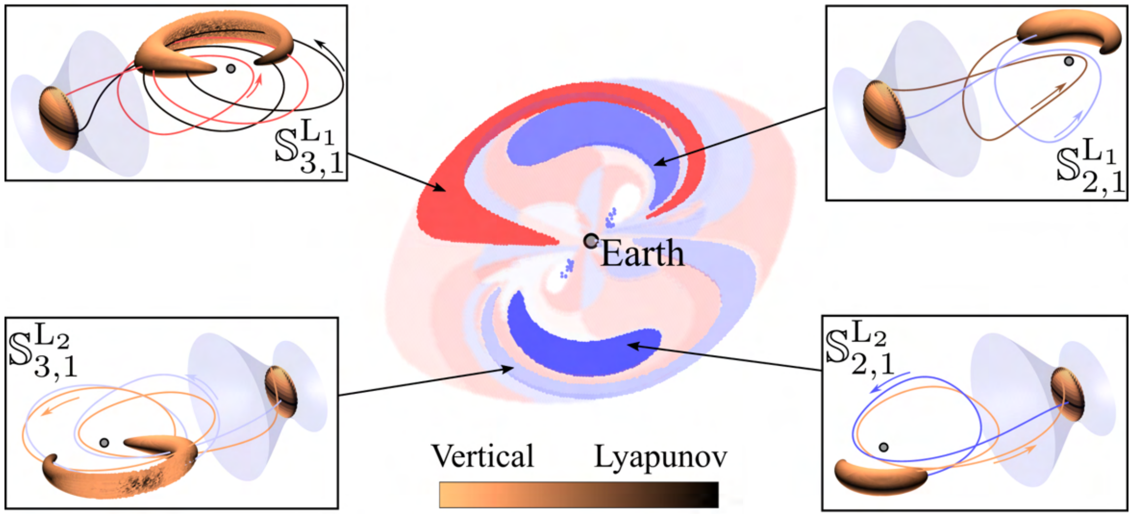

Figure 5.15: Collection of perigees along the stable manifolds emanated from invariant tori near L_1 and L_2 in the Sun-Earth CR3BP at $C_J = 3.00088$ that experience approach to the gateway regions before completing three revolutions. Central figure visualizes the projections of the perigees of the global clustering result associated with trajectories naturally escaping the Earth region, colored in shades of blue and red. The insets report the perigees along the stable manifold arcs and associated with trajectories presenting geometrical similarity, depicted consistently with the approached torus. Reproduced with permission from Springer Nature.

researchers on the governing nature of periodic and linearized quasi-periodic solutions in the CR3BP [1, 131]. However, the result in this section visually confirms the association between departing trajectories and manifold arcs emanated from invariant 2-tori near L_1 and L_2, in a data-driven approach that allows reduced human intervention and enhanced visualization.

Stable manifolds arcs emanating from the invariant tori near L_1 and L_2 that present association with trajectories that remain in a bounded region near the Earth for three revolutions are also examined. Figure 5.16 visualizes the initial perigees of arcs along the stable manifold that are assigned by the trained classifier to specific groups from the global clustering result, in a configuration consistent with Fig. 5.15. However, the depicted insets also report: the continuation of the manifold arcs for a subsequent crossing with the periapsis map with a gray line, to highlight natural approach of the arcs towards the associated torus; the collection of perigees of the global clustering result assigned to the same cluster, overlaid in shades of blue and red semi-transparent markers consistent with the indicated cluster in the central figure, to highlight that the perigees of the stable manifold overlap only a portion of the associated clusters.

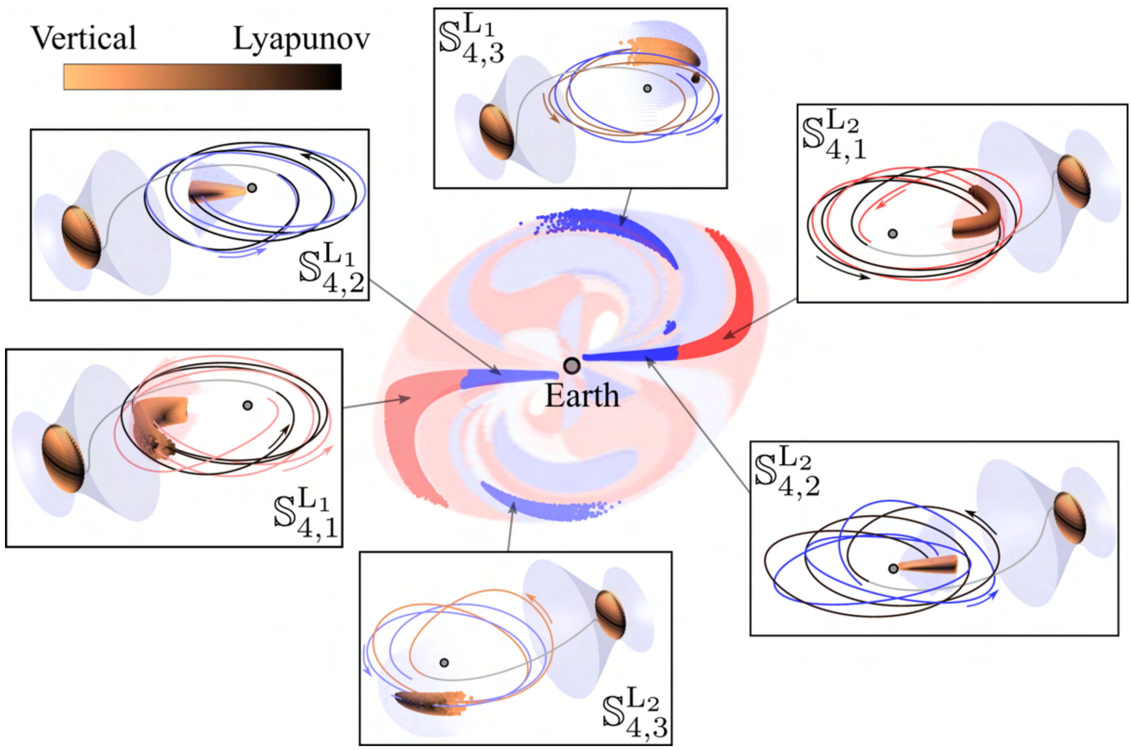

Figure 5.16: Collection of perigees along the stable manifolds emanated from invariant tori near L_1 and L_2 in the Sun-Earth CR3BP at $C_J = 3.00088$ that experience temporary capture in the Earth region for three revolutions. Central figure visualizes the projections of these perigees on the configuration space, colored in shades of blue and red. Insets reporting the perigees of the stable manifold arcs presenting geometrical similarity, colored consistently with the approached torus. Reproduced with permission from Springer Nature.

Stable manifold arcs emanated from tori near the L_1 and L_2 gateway regions also demonstrate direct influence on subsets of trajectories that experience temporary capture in the Earth region. Indeed, sets of perigees contained in specific clusters obtained from the global clustering result appear in the same region of perigees along the stable manifold arcs, and generate trajectories that are geometrically similar to a subset of stable manifold arcs. Therefore, these trajectories likely experience natural departure from the Earth region if propagated for more than three returns to the periapsis map. However, during the short propagation time used to construct the global clustering result discussed by Bonasera and Bosanac, the trajectories associated with the perigees visualized in the center of Fig. 5.16 inherit the geometrical features of the spatial stable manifold arcs from tori near the L_1 and L_2 gateway regions [130]. This result demonstrates the additional benefit of leveraging the presented data-driven approach to infer significant insight on the governing nature of the unstable invariant tori. The latter allows to investigate and visualize natural transport mechanisms in higher-dimensional design spaces that can be used for actual trajectory design, without requiring the challenging generation of analytic separation criteria and removing substantial burden from the human analyst.

5.4 Summary of Contributions

This chapter presents the use of data-driven approaches for an unsupervised differentiation of trajectory arcs on higher-dimensional Poincaré maps based on geometrical similarity [13]. The devised method, leveraging HDBSCAN in a distributed clustering approach, is used to correlate groups of similar geometries across partitions of the same spatial map, or maps generated from different dynamical models and values of the independent variable. First, a method to correlate clusters based on mutual intersections in the phase space is presented, and demonstrated to reduce the computational and visualization efforts of clustering a three-dimensional periapsis map in the Sun-Earth CR3BP at $C_J = 3.00088$. This approach is expanded by using UMAP to automatically correlate clusters of distinct maps, and demonstrating cluster persistence across model of increasing fidelity, as the CR3BP, the ER3BP and a point mass ephemeris model. This approach assists the trajectory designer by: enhancing visualization of arcs generated in higher-dimensional design spaces; assessing the evolution of groups of geometrically similar trajectories across models of increasing fidelity; understanding the geometries of arcs that are often leveraged during trajectory design, such as stable manifold trajectories. This section concludes the chapter by discussing the scientific contributions of the presented approach, the benefits of leveraging unsupervised learning for analyzing a higher-dimensional Poincaré map, and potential avenues for future research.

5.4.1 Scientific Contributions of the Approach

A data-driven approach is used to autonomously differentiate datasets of trajectories generated in models of increasing fidelity, including the CR3BP, the ER3BP, and a point mass ephemeris model. The main scientific contributions of this approach include:

1) Unsupervised learning for high-dimensional Poincaré maps: tools from unsupervised learning as HDBSCAN, UMAP, and distributed clustering are used to autonomously partition a dataset of trajectories captured by higher-dimensional Poincaré maps. The result is a summary of the solution space.

2) Cluster aggregation: distributed clustering is used to enable clustering of a very large dataset of trajectories. Mutual intersection in phase space is used to aggregate clusters on a map in the Sun-Earth CR3BP. The lower-dimensional projections of a subset of crossings from different Poincaré maps is also used to identify cluster persistence across distinct dynamical models, or different values of the independent variable.

3) Governing structures in the spatial CR3BP: the global clustering result is applied to classify a set of arcs along the stable manifold of tori near the L_1 and L_2 gateways in the Sun-Earth CR3BP at $C_J = 3.00088$. The classified stable manifold arcs possess similar geometrical features to a set of groups retrieved from the clustering results, demonstrating the governing nature of the investigated stable manifold arcs on the solution space near the Earth in a data-driven approach.

The presented data-driven approach can facilitate analysis of high-dimensional Poincaré maps, enabling enhanced visualization and reducing the burden of a human analyst on a variety of tasks, comprising the construction of trajectory segments, and analysis of structures governing the available design space.

5.4.2 The Value of Unsupervised Learning for Higher-Dimensional Poincaré Maps

To enable the autonomous differentiation of trajectories based on geometrical similarity, HDBSCAN, distributed clustering, and UMAP are used to construct a global clustering result, or assess cluster persistence across models of increasing fidelity, given as input a dataset of feature vectors. These techniques generate multiple benefits for investigating a high-dimensional Poincaré map. The main benefits of unsupervised learning a higher-dimensional Poincaré map include:

1) Autonomous differentiation: in the provided examples, trajectories are grouped based on their geometric characteristics. The partitioning is performed in an unsupervised approach, that mitigates the need for a priori identification of analytical and problem-dependent separation criteria, that are often cumbersome to formulate for a human analyst.

2) Enhanced visualization: the partitioning divides the design space in regions of trajectories with similar geometrical features, and allows the analyst to summarize the solution space with a reduced set of representative solutions. The trajectory designer can then focus on arcs with specific geometries and potentially remove trajectories that are not of interest.

3) Knowledge discovery: the global clustering result can potentially highlight the existence of regions of the solutions space that are influenced by known dynamical structures in the model. For example, regions of the design space that are governed by the stable manifolds of tori near L_1 and L_2 are visualized and analyzed in Sec. 5.3.2. Likewise, cluster aggregation enables the assessment of cluster persistence across distinct models, and cluster evolution across values of the independent variable in higher-fidelity regimes. Such insight informs model selection during the trajectory design process.

5.4.3 Avenues for Future Work

The application of unsupervised learning techniques on Poincaré maps may benefit from ongoing work:

1) An expert trajectory designer might find unintuitive the selection process associated with: the input parameters for HDBSCAN; the amount of subdivisions used to partition the dataset; the input parameters used to process the dataset with UMAP; or the number of nearest neighbors used to train the k-nearest neighbor classifier. For the selection of a subset of these parameters, the presented investigation provides a robust approach, leveraging different clustering performance criteria. However, other parameters are often heuristically selected, and heavily depending on the nature of the problem. Different parameters and a distinct selection of the trained classifier might result in different global clustering results. Future work may include devising meaningful heuristics or automated approaches to selecting these quantities.

2) The computational complexity of the devised approach warrants further improvement. Indeed, rapid result generation represents an important asset during the fast-paced phases of the trajectory design process. For the scenario presented in this chapter, HDBSCAN represents a computationally lightweight solution, clustering a single partition in a few seconds on the same Intel Core i7-2600K @ 3.40GHz. UMAP is applied in the devised method at different instances, and it generally requires a few minutes for each application on an identical computational machine. The classifier is trained on the same machine in a few minutes. Future versions of the devised method might replace UMAP and the selected classifier with a different algorithm to more rapidly generate a global clustering result.

Chapter 6

Constructing Natural Transfers between Quasi-Periodic Trajectories

Hyperbolic invariant manifolds are dynamical structures that asymptotically depart and approach unstable periodic and quasi-periodic solutions. These structures are of great interest to understand natural transport mechanisms in the solar system and to aid in the construction of trajectories with reduced propellant consumption. Among the variety of existing periodic and quasi-periodic trajectories, solutions near resonances are often investigated in celestial mechanics to understand natural transport mechanisms of small bodies, like Kuiper Belt Objects (KBOs), and to investigate the feasibility of interesting trajectory arcs for space missions, like the Transiting Exoplanet Survey Satellite (TESS) and the Interstellar Boundary Explorer (IBEX) [5,20,21,23]. The study of natural connection mechanisms between two different periodic orbits near resonances in the CR3BP has been the subject of different studies. These investigations highlight a rich design space for natural connections between pairs of orbits near resonance in the CR3BP [3,5,24]. However, the solution space can be exponentially expanded by incorporating bounded quasi-periodic solutions nearby orbits near resonance in the available solution space. These quasi-periodic trajectories foliate the surface of invariant tori, which can present hyperbolic invariant manifolds, therefore allowing the existence of departing and approaching natural flows. Olikara examines two approaches to leverage the hyperbolic dynamical flow near invariant tori and design natural connections between two different tori. In a first approach, a trajectory is constrained to depart from an invariant torus with fixed geometrical characteristics, while the characteristics of the arrival torus are retrieved after generating the natural connection. This approach is demonstrated to generate a transfer in a neighborhood of the secondary in the CR3BP, and its feasibility relies on the definition of a very limited solution space, where trajectories are excluded if trespassing one of the two bottlenecks. In the second approach, Olikara leverages a planar solution connecting two fully defined periodic orbits, to retrieve a natural connection between two invariant tori using numerical continuation [30]. This section extends the finding of Olikara, by presenting a flexible technique that enables the construction of natural connections between two unstable invariant tori in the CR3BP. The presented methodology, outlined in Sec. 6.1, does not require any constraint applied on the solution space, and does not necessitate the existence of a planar solution between two nearby periodic orbits. The devised methodology is demonstrated in a scenario connecting two invariant tori near resonances in the CR3BP, presented in Sec. 6.2, and subsequently extended to generate families of geometrically similar transfers in Sec. 6.3. However, the presented technique provides a general and flexible framework, that can be applied to other pairs of invariant tori presenting the same level of energy and unstable hyperbolic invariant manifolds. Ultimately, the main findings of the chapter are summarized in Sec. 6.4. The presented approach and demonstrated examples appear in a conference paper by Bonasera and Bosanac [132]

6.1 Method Overview

This section presents a methodology leveraging a technique from manifold learning to construct initial guesses for natural transitions between two invariant 2-tori. The analyzed tori are unstable and are generated by constraining the average energy level of the first invariant curve to an identical value, following the techniques presented in Sec. 3.5. This methodology is comprised of two fundamental phases:

1) Initial guess construction: a discontinuous initial guess for a natural transfer between two spatial invariant tori is constructed. Initially, two families of invariant tori, with non-empty hyperbolic manifolds, are generated by enforcing an identical value of average Jacobi constant for the initial approximating invariant curve as a constraint in the formulation of

the differential corrector, as outlined in Sec. 3.5. Constructing families of quasi-periodic trajectories enables the generation of families of geometrically similar transfers connecting families of tori, performed at the end of the second phase. However, during the initial guess construction only one member per family is necessary. To aid the correction process during the second phase, one invariant torus per family is selected with comparable out-of-plane displacements. The selected invariant tori are leveraged to construct the transfer: the crossings of the hyperbolic manifolds emanating from the selected members of these families with a common surface of section are used to generate a higher-dimensional Poincaré map. Then, UMAP is used to project the higher-dimensional crossings onto a lower dimensional representation. The alternative representation of a Poincaré map constructed by UMAP supports the identification of an initial guess for a natural connection between the analyzed invariant tori. Regions of the projected space with nearby crossings of the two manifolds are visually identified to select a pair of crossings constituted by a stable and unstable arc intersection. The pair of crossings is propagated to generate an initial guess for a natural transfer between the investigated tori near distinct resonances.

2) Trajectory correction and continuation: the initial guess is corrected with an optimizer. The constrained optimization problem is formulated to enforce a continuous trajectory, simultaneously minimizing the distance of the initial and final states of the transfer from the considered two tori. Then, the corrected natural connection is used in a continuation scheme to construct a family of transfers with similar geometry, connecting other members of the families of invariant tori.

These phases are further elaborated in the remaining part of this section.

6.1.1 Phase 1: Initial Guess Construction

The first phase in the process of constructing a natural connection between two invariant 2-tori corresponds to the identification of an initial guess for a finite time approximation of the transfer. The process used in this investigation is summarized as follows:

1) *Construct two families of invariant 2-tori*: two planar periodic orbits at the same value of Jacobi constant, are identified. Both orbits present: a nontrivial pair of stable/unstable hyperbolic manifolds, to exhibit nearby dynamical flows naturally departing and approaching the orbit; a nonempty center manifold, indicating that a nearby family of quasi-periodic trajectory exists. The step enforces the existence of a nontrivial nearby hyperbolic manifold because the nearby invariant tori, originated from the considered periodic orbit at the same energy level, tend to inherit the stability characteristics of the originating orbit. The numerical approach described in Sec. 3.5 is used to calculate these two families of invariant tori. Moreover, the constraint vector of the problem formulation is augmented with the average value of the Jacobi constant over the set of states approximating the first invariant curve. This step produces two families of unstable quasi-periodic trajectories that possess a similar Jacobi constant.

2) *Identify the surface of section*: a surface of section is defined to generate insights on the flow associated with the hyperbolic invariant manifolds of the selected families of invariant tori. For the examples discussed in this investigation, a $y = 0$ hyperplane is employed with no additional constraints on the sign of the velocity component at each crossing of the surface of section, producing a two-sided map. Constraining the sign of a velocity component is a traditional approach when leveraging a Poincaré map to visually investigate the existence of connecting trajectories between two periodic orbits [3, 5, 24]. However, constraining a component of the spacecraft state might remove feasible candidates from the available solution space. Using UMAP to project the crossings, as described at the later steps of the presented methodology, eliminates the necessity of constraining the trajectory space, potentially allowing the investigation of a wider solution space.

3) *Record the crossings on the map*: one invariant torus is selected from each family and the crossings of the emanated hyperbolic manifolds with the selected hyperplane are recorded.

Both selected tori possess a similar maximum out-of-plane displacement: this step is empirically demonstrated to aid the subsequent correction mechanism, performed during the second phase, for invariant tori near resonances; however, this step might not be required for different pairs of quasi-periodic trajectories. Subsets of the stable and unstable manifolds associated with each of the selected invariant tori are generated using a small displacement ϵ along the eigenvectors associated with the real stable and unstable eigenvalue of $\boldsymbol{DS}$, respectively, as outlined in Sec. 3.6. Up to N_C crossings of each manifold arc are recorded with the defined surface of section. Often, the hyperbolic invariant manifolds associated with the selected invariant tori tend to remain in the vicinity of the originating solution for a few revolutions. These crossings do not present great diversity and are generally not useful for generating natural connections between the investigated tori. Therefore, the first N_I intersections of each manifold arc with the common surface of section are removed from the analyzed dataset. Removing a subset of the solution space improves the computational efficiency and visual investigation of the later steps of the presented methodology.

4) *Project the crossings onto a lower-dimensional space*: a single dataset is populated with the recorded map crossings associated with the hyperbolic manifolds from the investigated tori. Each five-dimensional datapoint in this dataset corresponds to a crossing with the defined surface of section. UMAP is leveraged to project the dataset onto a two-dimensional Euclidean space: the dimensionality of the embedding is selected to prevent data obscuration from representations with more than two dimensions, and reflects the dimensionality of states along the surface of a torus that are identified by two angular quantities $[\theta_1(t), \theta_2(t)]$. The two input parameters used for the projection performed by UMAP are selected as $n_n = 100$ and $m_{\text{dist}} = 0.0$. This parameter selection supplies a representation that emphasizes compactness and retains the global structure of the Poincaré map crossings. Moreover, a low value selected for n_n enables a relatively rapid construction of the embedding from UMAP.

5) *Construct an initial guess from the UMAP embedding*: the projected lower-dimensional representation preserves the structure of the original dataset of five-dimensional crossings. For this reason, areas of the projected space where the crossings of the hyperbolic manifolds have a low relative distance are investigated as candidate regions for locating natural transfers between the investigated tori. In these areas, two crossings (one from each of the stable and unstable manifolds) that lie nearby in the lower-dimensional representation are selected to construct an initial guess. Recall that UMAP generates a bijective mapping between the low- and high-dimensional dataset: each solution in the planar embedding corresponds to a unique spacecraft state in the five-dimensional phase space. Thus, two points in the low-dimensional embedding correspond to two unique crossings in the higher-dimensional phase space. Once a pair of crossings is identified in the low-dimensional representation, the associated map crossings in the five-dimensional phase space are propagated backward and forward in time to generate the associated unstable and stable manifold arcs, respectively. Then, a number of revolutions N_r of the associated quasi-periodic trajectories is concatenated to the beginning and to the end of the transfer to form a suitable initial guess. Appending multiple arcs along the invariant tori facilitates the convergence to a natural transfer, despite the discontinuity generated by the steps along the stable and unstable eigenvalues during the initialization of the manifold arcs.

The presented procedure for generating an initial guess is demonstrated in this investigation for a variety of transfers between different invariant tori near resonances in the Earth-Moon CR3BP. After generation of an initial guess, the constructed arc is corrected with a constrained optimization scheme, and continued across the generated members of the families of quasi-periodic trajectories to generate a family of geometrically similar transfers.

6.1.2 Phase 2: Transfer Correction and Numerical Continuation

In the second phase of the proposed method, the initial guess is employed to first recover a single continuous solution connecting the analyzed invariant tori near resonances. Consequently, the retrieved end-to-end solution is continued across the family of invariant tori to construct a family of transfers with similar geometry connecting two families of invariant tori in the CR3BP. In this investigation, the correction is implemented with a multiple shooting algorithm and a constrained optimization scheme, designed to: enforce continuity between the hyperbolic invariant manifold arcs; ensure the transfer is bounded by the selected tori; enforce the energy level. The correction is formulated as an optimization problem, solved with Matlab's *fmincon* function [133].

In the problem formulation, the objective function is designed to minimize the full-state displacement between the transfer's initial state and the departing torus, and the transfer's final state and the arrival torus. The constraint vector is represented by the set of continuity constraints that are enforced at each node along the transfer, excluding the initial and final states of the trajectory. Moreover, the energy of the initial state along the initial guess is constrained to have the same Jacobi constant of the state along the departing torus that originated the unstable manifold arc. The formulation is presented as an optimization problem because difficulties in the convergence process are observed when leveraging Newton's method to recover the continuous end-to-end trajectory connecting the analyzed tori. Optimization addresses this challenge by providing a more robust approach for the trajectory correction. Nevertheless, the presented approach supplies solutions that correspond to natural connections, within a numerical tolerance. To characterize the optimization problem for the generation of a continuous transfer between two invariant 2-tori in the CR3BP, mathematical definitions of the free variable vector, constraint vector, and the objective function are presented.

Free variable vector: First, the initial guess, constructed through the first phase of this approach, is discretized into N arcs. The states at the beginning of each arc form the free variable vector as:

$$\boldsymbol{V} = [\boldsymbol{x}_1,\, \boldsymbol{x}_2,\, \ldots,\, \boldsymbol{x}_N,\, t_{1,2},\, t_{2,3},\, \ldots,\, t_{N-1,N}]^T \in \mathbb{R}^{7N-1} \tag{6.1}$$

with $\boldsymbol{x}_i$ for $i \in \{1, 2, \ldots, N\}$ representing the six-dimensional states at the beginning of each arc and $t_{j,j+1}$ for $j \in \{1, 2, \ldots, N-1\}$ denoting the propagation time from node j to node $j+1$. The free variable vector is iteratively corrected to optimize the generic minimization problem

$$\boldsymbol{V} = \arg\min_{\boldsymbol{V}} f(\boldsymbol{V}) \qquad \text{subject to } \boldsymbol{F}(\boldsymbol{V}) = \boldsymbol{0} \tag{6.2}$$

Constraint vector: In the formulation proposed in Eq. (6.2), the equality constraint vector $\boldsymbol{F}(\boldsymbol{V})$ contains the energy and continuity constraints as:

$$\boldsymbol{F}(\boldsymbol{V}) = \begin{bmatrix} C_J(\boldsymbol{x}_1) - \bar{C}_J \\ \boldsymbol{x}_1(t_1; t_{1,2}) - \boldsymbol{x}_2 \\ \boldsymbol{x}_2(t_2; t_{2,3}) - \boldsymbol{x}_3 \\ \vdots \\ \boldsymbol{x}_{N-1}(t_{N-1}; t_{N-1,N}) - \boldsymbol{x}_N \end{bmatrix} \in \mathbb{R}^{6N+5} \tag{6.3}$$

where $\bar{C}_J$ is the Jacobi constant associated with the state of the departing torus where the unstable manifold arc starts from, while $\boldsymbol{x}_j(t_j; t_{j,j+1})$ represents the spacecraft state propagated from $\boldsymbol{x}_j(t_j)$ for a time $t_{j,j+1}$, for $j \in \{1, 2, \ldots, N-1\}$. Recall that both families of invariant tori are constructed with a constrained average Jacobi constant enforced over the states along the associated first invariant curves. This constraint often generates families of invariant tori with similar energy levels, and with extended out-of-plane displacement. However, the sequence of states along a unique invariant curve of a generated torus can exhibit different values of Jacobi constant, since only the average value of the Jacobi constant is enforced. Although two families of tori are generated with the same value of constrained average Jacobi constant, the manifold arcs originated from two members of these families can be associated with states at slightly different energy levels. The largest variations of Jacobi constant across states of the same invariant curve is generally observed for invariant tori with relatively wide out-of-plane displacements within each family, often representing the last members of the generated families. In these scenarios, the stable and unstable manifold arcs that form the first and last portion of the initial guess solution, respectively, can present a slightly different energy level. Therefore, the transfer can converge to a trajectory that approximately represents a natural connection between the analyzed tori.

Objective function: The objective function $f(\boldsymbol{V})$ highlighted in Eq. (6.2) is designed to reflect the discontinuity between the terminal and starting states along the transfer and the associated tori; conceptually, the objective function reflects the requirement that the beginning of the transfer naturally flows away from the departing torus and the end of the transfer naturally flows into the arrival torus. This definition is mathematical formulated as:

$$f(\boldsymbol{V}) = \|\boldsymbol{x}_1 - \boldsymbol{x}_{T1}\|^2 + \|\boldsymbol{x}_N - \boldsymbol{x}_{T2}\|^2 \tag{6.4}$$

where $\boldsymbol{x}_{T1}$ and $\boldsymbol{x}_{T2}$ represent the closest states along the associated tori to the initial and final states $\boldsymbol{x}_1$ and $\boldsymbol{x}_N$ along the transfer, respectively. The first part of the objective function, $\|\boldsymbol{x}_1 - \boldsymbol{x}_{T1}\|^2$, is here leveraged to detail an overview of the process for the computation of $\boldsymbol{x}_{T1}$. The same procedure can be repeated to compute $\boldsymbol{x}_{T2}$. Recall that a torus is approximated by a finite set of invariant curves, each approximated by a finite set of points. Therefore, the state on the surface of the torus with the smallest distance from $\boldsymbol{x}_1$, and the point of the set of states used to approximate the torus with the smallest distance from $\boldsymbol{x}_1$ likely correspond to two distinct solutions. The latter is just an approximation of $\boldsymbol{x}_{T1}$, and cannot be used to compute the first term in $f(\boldsymbol{V})$. However, the available set of points used to approximate the torus can be used to find $\boldsymbol{x}_{T1}$. In particular, the problem of finding a state lying on the torus surface and with the lowest distance from $\boldsymbol{x}_1$ is recast into the problem of finding two relative angular quantities, (τ_1, τ_2). These relative angular quantities are each associated with the two fundamental angular coordinates that describe each state on a 2-torus, (θ_1, θ_2), respectively. To compute the pair (τ_1, τ_2), a reference invariant curve $\boldsymbol{U}(\theta_1, \theta_2)$ along the analyzed torus is identified. To help the procedure, this reference invariant curve is selected as the curve containing the closest state from $\boldsymbol{x}_1$, given the finite set of invariant curves used to approximate the torus. Recall again that this state is not $\boldsymbol{x}_{T1}$, due to the inherent approximation of the invariant curve. The rotation of the set of points along the selected reference curve $\boldsymbol{U}(\theta_1, \theta_2)$ by two angles (τ_1, τ_2) generates a new invariant curve $\boldsymbol{U}(\theta_1+\tau_1, \theta_2+\tau_2)$. Practically, a rotation along the longitudinal coordinate θ_1 is performed by propagating each state along the invariant curve $\boldsymbol{U}(\theta_1, \theta_2)$ for a propagation time that is associated to the angular quantity τ_1, ultimately generating $\boldsymbol{U}(\theta_1 + \tau_1, \theta_2)$. Moreover, the invariance condition of a torus applied to an approximated invariant curve, introduced in Eq. (3.23), presents the definition of a rotational matrix $\boldsymbol{R}(\cdot)$. This operator can be used to perform the second rotation τ_2 along the transversal angular coordinate θ_2. The second rotation can be written as:

$$\boldsymbol{U}(\theta_1 + \tau_1, \theta_2 + \tau_2) = \boldsymbol{R}(\tau_2)\boldsymbol{U}(\theta_1 + \tau_1, \theta_2) \tag{6.5}$$

If $\boldsymbol{U}(\theta_1 + \tau_1, \theta_2 + \tau_2)$ contains $\boldsymbol{x}_{T1}$ and $\boldsymbol{x}_1$ lies on the torus surface, then:

$$\boldsymbol{x}_1 - \boldsymbol{U}(\theta_1 + \tau_1, \theta_2 + \tau_2)|_1 = \boldsymbol{0} \tag{6.6}$$

where the subscript $\cdot|_1$ indicates that only the first element of the set of points approximating $\boldsymbol{U}(\theta_1 + \tau_1, \theta_2 + \tau_2)$ is used. However, $\boldsymbol{U}(\theta_1 + \tau_1, \theta_2 + \tau_2)$ likely does not contain $\boldsymbol{x}_{T1}$, therefore (τ_1, τ_2) are iteratively corrected leveraging a single-shooting approach. Empirically, the method tends to quickly converge from a general initial guess $(\tau_1, \tau_2) = (0.01, 0.01)$, if the torus contains enough invariant curves. A minimum of $M = 25$ invariant curves per torus is selected in this investigation. After obtaining the set of angles (τ_1, τ_2) that enables to retrieve $\boldsymbol{x}_{T1}$, the same procedure is repeated to evaluate the second term in the objective function using the arrival torus.

The optimization problem in Eq. (6.2) is solved using interior point in the MATLAB routine *fmincon*. Since the average Jacobi constant over the first invariant curve is used as a constraint to construct both families of invariant tori, a transfer is considered a natural connection if it corresponds to an objective $f(\boldsymbol{V}) \leq 10^{-12}$ and a constraint $\|\boldsymbol{F}(\boldsymbol{V})\|_2 \leq 5 \times 10^{-14}$. The threshold in the objective function corresponds to a cumulative approximate displacement of 400 m and 1 mm/s in the Earth-Moon system from the departing and arrival torus, and it is deemed reasonable given the impact of the difference in the Jacobi constants between states of the invariant curves associated with the connected tori. Recall that the original departing and arrival states that correspond to the stable and unstable manifold arcs may potentially have a slightly different Jacobi constant; thus, an exact natural connection would be impossible to generate.

After the optimization strategy recovers a single point transfer between the selected pair of invariant tori, a continuation scheme is used to generate a family of transfers with similar geometries between different pairs of members of the two families of invariant tori. The continuation approach is designed to follow a nested iteration structure with an external and an internal loop. In the internal loop, when a first transfer is found, the continuation scheme maintains fixed the departing torus, but steps along the contiguous members of the arrival family of invariant tori. To perform the correction of the natural transfer for the new pair of invariant tori with the optimization scheme in Eq. (6.2), the previously obtained solution is employed as an initial guess. Then, if the optimization converges to a transfer connecting the two members of the families of tori with the defined thresholds in the objective and constraint vector, the converged transfer is used as an initial guess for the next step of the internal iteration when a new step along the family of arrival tori is performed. The internal iteration terminates when there are either no more members along the arrival torus family or a feasible transfer cannot be computed. Then, the process returns to the external loop: the first solution obtained in the inner loop is used as an initial guess to obtain a new transfer. To generate this new transfer, the continuation scheme steps along the departing family of invariant tori. After convergence, a new inner loop is started. This procedure enables the computation of transfers with similar geometries to the initial guess, connecting spatial quasi-periodic trajectories along the two selected families. Note that this investigation only searches for the existence of one transfer between each combination of the available invariant tori and within the neighborhood of the initial guess. Similar solutions may also appear by varying the departure and arrival locations along each torus. Nevertheless, the implemented approach enables a preliminary analysis of the natural transitions between invariant tori.

6.2 Recovering Natural Transfers between Prescribed Invariant 2-Tori near Resonances

The methodology described in Sec. 6.1 is leveraged to construct and analyze different natural transfers between two families of quasi-periodic trajectories near resonances in the Earth-Moon CR3BP. Specifically, the methodology is demonstrated to generate transfers departing an invariant torus near the 3:2 resonance and approaching an invariant torus near the 1:2 resonance. A point solution is constructed using a Poincaré map and dimensionality reduction, in combination. After recovering a natural transfer connecting two tori near the 3:2 and 1:2 resonances, the design space is analyzed and additional transfers with different geometries are generated. This section concludes by demonstrating the approach to connect invariant tori near different combinations of resonances in the Earth-Moon CR3BP.

Families of invariant tori near the 3:2 and 1:2 resonances are generated in the Earth-Moon CR3BP. The planar resonant orbits depicted in the center of Fig. 6.1 are employed to compute members of the nearby families of invariant tori. These planar orbits near the 3:2 and 1:2 resonances exist at a energy level of $C_J = 2.73$ with a period of $T_{3:2} \approx 55.92$ days and $T_{1:2} \approx 50.54$ days, respectively. Both generated orbits have nonempty spatial central and hyperbolic manifolds at this energy level. Families of invariant tori are generated, enforcing the average Jacobi constant over the first invariant curve to a value $C_J = 2.73$. To construct an initial guess for generating a torus, a perturbation of $\epsilon = 5 \times 10^{-5}$ in Eq. (3.24) is used to step along the eigenvector associated with the oscillatory mode of the periodic orbits. Each torus is computed using $N_Q = 25$ states along an invariant curve and $M_Q = 25$ nodes along the orbit. The initial guess of the invariant torus is corrected according to the approach presented in Sec. 3.5. After generating an initial torus near the 3:2 and 1:2 resonances, pseudo-arclength continuation is applied to generate up to a total of 20 members per family. Each member in the family also satisfies the average value of the Jacobi constant over the first invariant curve constraint, set equal to $C_J = 2.73$. The continuation scheme generates two families of invariant 2-tori near resonances. The members with the largest out-of-plane components in configuration space are visualized on the sides of Fig. 6.1.

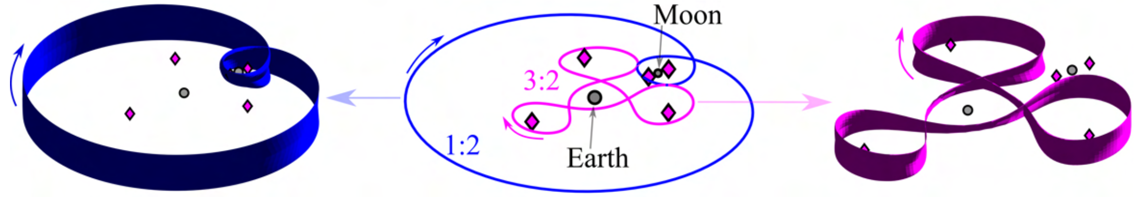

Figure 6.1: Example orbits and invariant tori near the 3:2 and 1:2 resonances in the Earth-Moon CR3BP at $C_J = 2.73$.

The methodology presented in Sec. 6.1 is demonstrated by constructing a natural connection departing the 3:2 torus and approaching the 1:2 torus visualized in Fig. 6.1. The unstable manifold of the torus near the 3:2 resonance and the stable manifolds of the torus near the 1:2 resonance are sampled using a grid of 101 locations over the longitudinal angular coordinate θ_1 and 25 points over the transversal angular coordinate θ_2. Therefore, the manifold of each torus is approximated with 2525 manifold arcs. The hyperbolic manifold arcs associated with each torus are calculated using an initial displacement of 100 km along the eigenvectors associated with the real unstable and stable eigenvalues, respectively, of the $\boldsymbol{DS}$ matrix, used during the construction of the tori. Each manifold arc is propagated for up to $N_C = 12$ returns to the $y = 0$ surface of section, in any direction. However, the first $N_I = 7$ map crossings for each manifold trajectory are excluded from this analysis, since they tend to remain close to the generating invariant tori. The selected intersections of the manifold arcs with the common surface of section produce a total of 15,114 and 14,925 crossings for the 3:2 and the 1:2 resonances, respectively. Often, to identify an intersection between an unstable and a stable manifold arc, the associated crossings are visualized on a projected space, where each coordinate has a physical meaning [1]. In particular, the crossings of the unstable and stable manifolds of the analyzed tori with the $y = 0$ surface of section are projected on the $(x,\, \dot{x})$ plane and the $(x,\, \dot{x},\, z)$ space, and are visualized in Fig. 6.2(a-b), respectively. Both representations highlight the complexity and richness of the solution space, with a large set of crossings been placed in the same regions. However, two crossings that appear on the same location in Fig. 6.2(a-b) are not often associated with a feasible natural connection between the investigated invariant tori. Indeed, both the two- and three-dimensional representations do not completely represent the higher-dimensional intersections of the invariant manifolds with the surface of section. Thus, two map crossings that are located nearby in either of these two- or three-dimensional projections may not be close in the full six-dimensional phase space. Including a fourth dimension in the map visualization, for example augmenting with color information or glyphs at every crossing, as well as introducing further constraints in the representation could mitigate this problem [3, 24, 28]. However, including more dimensions in the representation could complicate the visualization and analysis of the Poincaré map, while the design space could significantly shrink with additional constraints, potentially eliminating feasible solutions from the map.

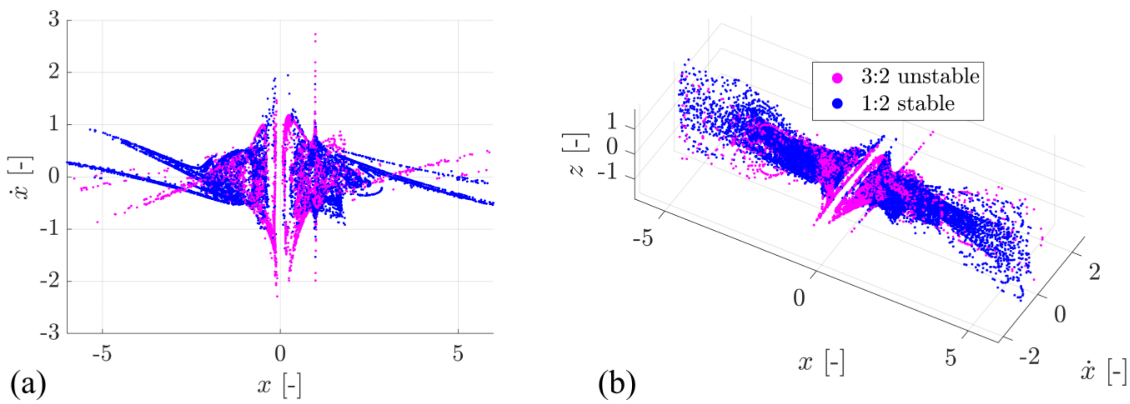

Figure 6.2: Poincaré map of the intersections of the hyperbolic invariant manifolds with the $y = 0$ plane emanated from the invariant tori near the 1:2 (blue) and 3:2 (magenta) resonances at $C_J = 2.73$ in the Earth-Moon system: (a) projection onto the $(x, \dot{x})$ plane and (b) projection onto the $(x, \dot{x}, z)$ space.

To alleviate the visualization burden of higher-dimensional Poincaré maps representation, UMAP is employed to project the six-dimensional crossings onto a two-dimensional Euclidean space. The reduced dimensionality of the projection allows to straightforwardly identify potential connections between the manifold arcs, avoiding cumbersome representations of the Poincaré map or the application of constraints to the solution space. In particular, the map crossings associated with both the stable and unstable manifolds are combined to form the complete dataset that is input to UMAP. The input parameters for UMAP are selected as $n_n = 100$, $m_{\text{dist}} = 0.0$, and $n_c = 2$, enabling a compact final representation onto a two-dimensional projected space. After processing the dataset with UMAP, the projected crossings are visualized in Fig. 6.3. The projection of the entire dataset is depicted in the center of this figure. The projected crossings associated with the stable manifold arcs, naturally approaching the 1:2 torus, are depicted with blue markers, while magenta markers are used to visualize the projected crossings associated with the unstable manifold arcs that naturally departs the 3:2 torus. Note that the coordinates of the projected space, labeled U_1 and U_2 in Fig. 6.3, do not retain any physical meaning. However, UMAP projects the dataset in order to maintain a topological similarity between the original dataset, constituted by the five-dimensional crossings with the $y = 0$ plane, and the projected crossings in the two-dimensional representation. This procedure preserves the global structure of the dataset. As a result, two map crossings that are close in the full phase space are expected to be located nearby in the two-dimensional projection. Analysis of the projected representation, visualized at the center of Fig. 6.3, reveals that there are a variety of regions where both blue and magenta points are nearby. In these regions the associated crossings of the stable and unstable manifolds may potentially cross the $y = 0$ surface of section with similar six-dimensional state vectors. Four of these regions are zoomed-in and framed on the sides of the central map representation. Each frame is identified by a label T_i, with $i = \{1, 2, 3, 4\}$ and it is used to generate a single transfer with a distinct geometry. However, the same region can host transfers with different geometries: indeed, the dataset of five-dimensional datapoints incorporates information on the crossings, without additional information on the geometrical characteristics of the generating manifold arc; therefore, geometrically different arcs emanated from the same torus can intersect the surface of section in similar points, and are projected in a nearby region by UMAP. Moreover, additional regions of overlapping magenta and blue point clouds appear in the center of Fig. 6.3, but are not analyzed in the remaining part of this section. These items represent additional explorations left for future investigations.

Each framed region on the two-dimensional projection calculated by UMAP is used to generate a point solution for a natural transition between quasi-periodic trajectories associated with tori near the 3:2 and 1:2 resonances in the Earth-Moon CR3BP. A pair of nearby crossings on the projected space is selected in each of the framed regions displayed at the boundaries of Fig. 6.3. Each pair is populated by one crossing from the unstable manifold associated with the invariant torus near the 3:2 resonance, and one crossing from the stable manifold associated with the invariant torus near the 1:2 resonance. The selected pair for each frame corresponds to the pair presenting the closest distance in the two-dimensional space. Besides the visualization advantage, using UMAP to select subregions of the embedding for searching feasible solutions generates two computational benefits: 1) a brute-force search over every available pair, each composed by a cross-

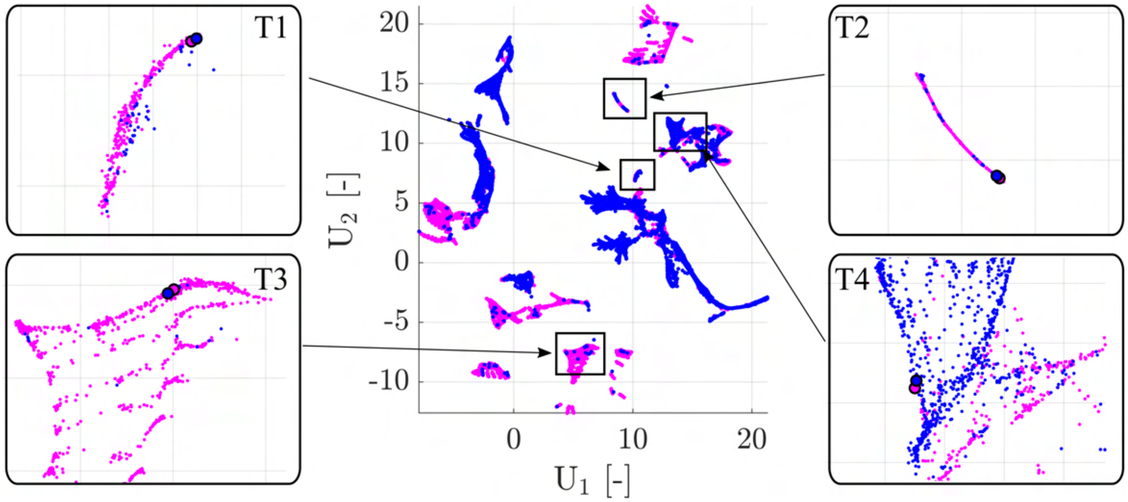

Figure 6.3: Center: projection of the crossing dataset in Fig. 6.2 onto a two-dimensional Euclidean space calculated by UMAP. Boundaries: zoomed-in perspectives of four regions of intersections between the stable and unstable manifolds of each invariant torus.

ing from the stable manifold and a crossing of the unstable manifold, is avoided; 2) the distances between crossings are processed on the two-dimensional solution space, rather than the original six-dimensional space. After selecting a pair of crossings for each frame in Fig. 6.3, an initial guess for a natural transfer is constructed by propagating towards the originating invariant tori. Four revolutions along each torus are appended to the beginning and the end of the initial guess to aid the subsequent correction process. The numerical corrections procedure described in Sec. 6.1 is implemented to recover a nearby natural transfer for each frame, displayed in Fig. 6.4.

In this figure, each transfer is labeled to correspond to the associated zoomed-in view in Fig. 6.3, marked by the labels T1 to T4. Each depicted transfer converges to a solution with an objective $f(\boldsymbol{V}) \leq 10^{-12}$, and satisfying the constraint with $\|\boldsymbol{F}(\boldsymbol{V})\|_2 \leq 5\times10^{-14}$ in a computational time of approximately 15 seconds per trajectory on an i7-2600K @ 3.40GHz. Furthermore, each transfer lies close to the initial guess. The retrieved natural transfers begin at the magenta circle marker on the invariant torus associated with the 3:2 resonance, and terminate at the blue circle marker on the invariant torus near the 1:2 resonance. Each transfer is colored in magenta for the portion associated with the unstable manifold emanated from the departing torus; the remaining

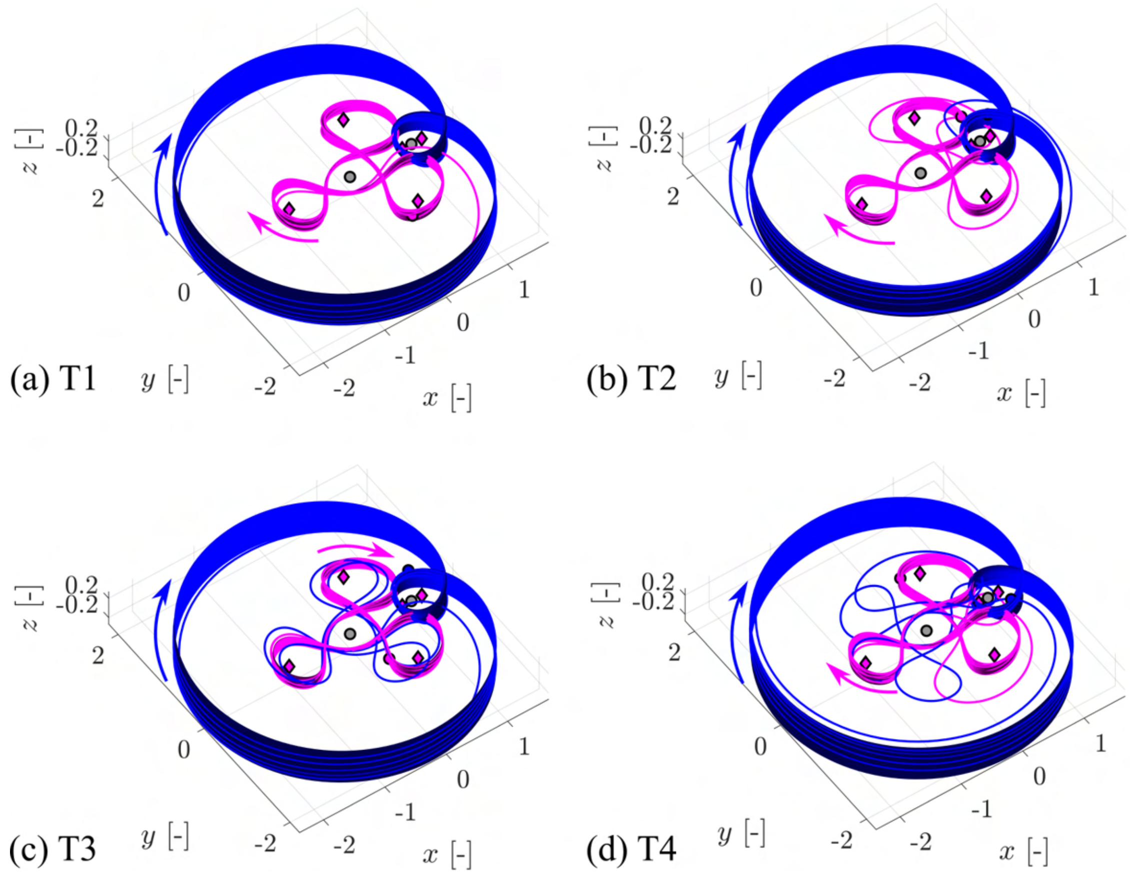

Figure 6.4: Selected natural transfers between invariant tori near the 1:2 and 3:2 resonances. The transfers are labeled by the region in Fig. 6.3 used to identify suitable map crossings.

part of each trajectory, associated with the approaching to the arrival torus, is colored in blue. These transfers each present a distinct geometry due to the different manifold arcs used to generate the initial guess. The intersections of these trajectories with the surface of section at $y = 0$ are displayed in the planar Poincaré map representation on the $(x, \dot{x})$ plane and the $(x, \dot{x}, z)$ space in Fig. 6.5(a-b). In this figure, the crossings of the 3:2 unstable manifold and the 1:2 stable manifold are displayed with magenta and blue transparent markers, respectively, while the intersections of the transfers depicted in Fig. 6.4 with the $y = 0$ plane appear as gray circles. These figures confirm that the crossings associated with the constructed transfers lie near the intersections of two manifold crossings on the surface of section in the traditional planar representation of a Poincaré map.

The same approach demonstrated in this section can be applied to construct transfers between

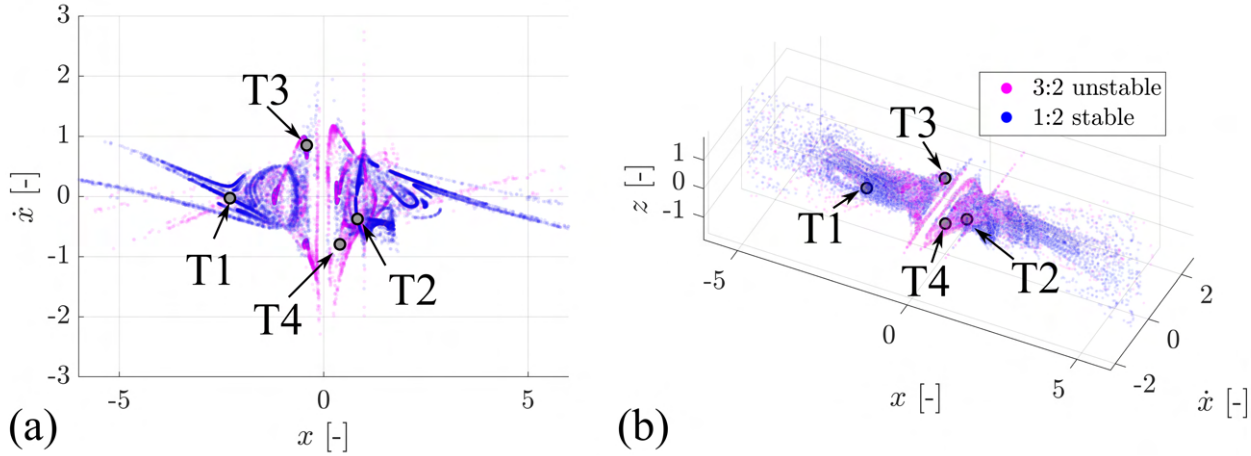

Figure 6.5: Crossings of the transfers (gray) in Fig. 6.4 overlaid on the Poincaré map of the hyperbolic invariant manifolds from the selected invariant tori near the 1:2 (blue) and 3:2 (magenta) resonances at $C_J = 2.73$, using a surface of section at $y = 0$: (a) projection onto the $(x, \dot{x})$ plane and (b) projection onto the $(x, \dot{x}, z)$ space.

tori near different resonances in the Earth-Moon system. Figure 6.6(a) visualizes the projected crossings of arcs from the unstable manifold emanated from a torus near the 3:1 resonance and the projected crossings of stable manifold arcs approaching a torus near the 1:3 resonance. The manifold crossings are constructed with invariant tori with a fixed average Jacobi constant over the first invariant curve of $C_J = 3$, and are propagated for up to $N_C = 18$ crossings, although removing the first $N_I = 10$ intersections. To construct an initial guess for a transfer departing the torus near the 3:1 resonance and naturally approaching the torus near the 1:3 resonance, a framed region from the UMAP projection in Fig. 6.6(a) is investigated. From this region, an initial guess is selected and corrected with the methodology highlighted in Sec. 6.1. The generated natural transfer is depicted in Fig. 6.6(b), using the a coloring scheme consistent with Fig. 6.4. The same procedure is repeated to construct a transfer departing a torus near the 2:3 resonance and naturally approaching a torus near the 1:5 resonance in the Earth-Moon system at $C_J = 2.6$. The UMAP projections associated with the manifolds emanated from the analyzed tori, and the constructed transfer, are visualized in Fig. 6.7(a-b), respectively. The transfers visualized in Figs. 6.6 and 6.7 represent only one solution among the variety of feasible natural transfers visualized in the associated UMAP representations.

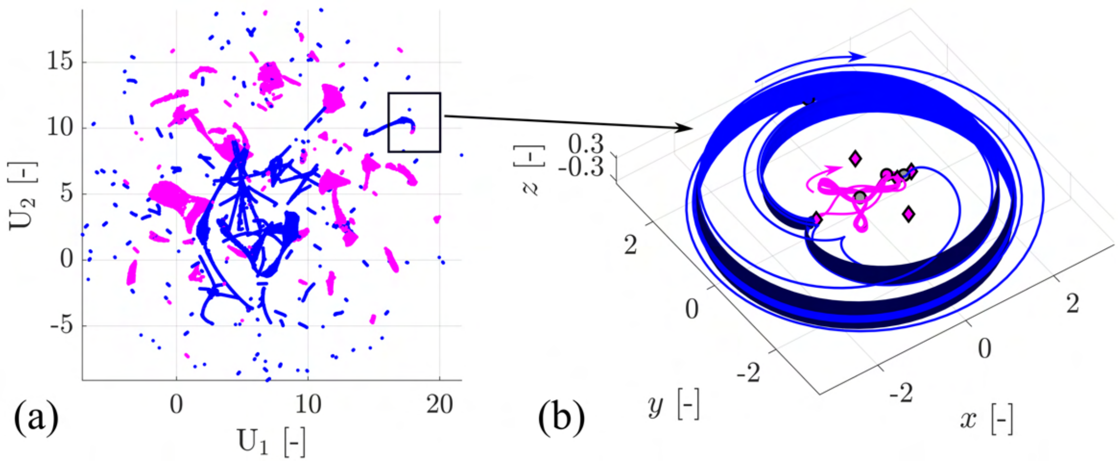

Figure 6.6: (a) UMAP projections of the crossings from the hyperbolic manifolds of two invariant tori near the 1:3 (blue) and the 3:1 (magenta) resonances with the $y = 0$ plane; (b) natural transfer departing the torus near the 3:1 resonance and naturally approaching the 1:3 torus near resonance, initialized by leveraging the UMAP projection, in the Earth-Moon system at $C_J = 3$.

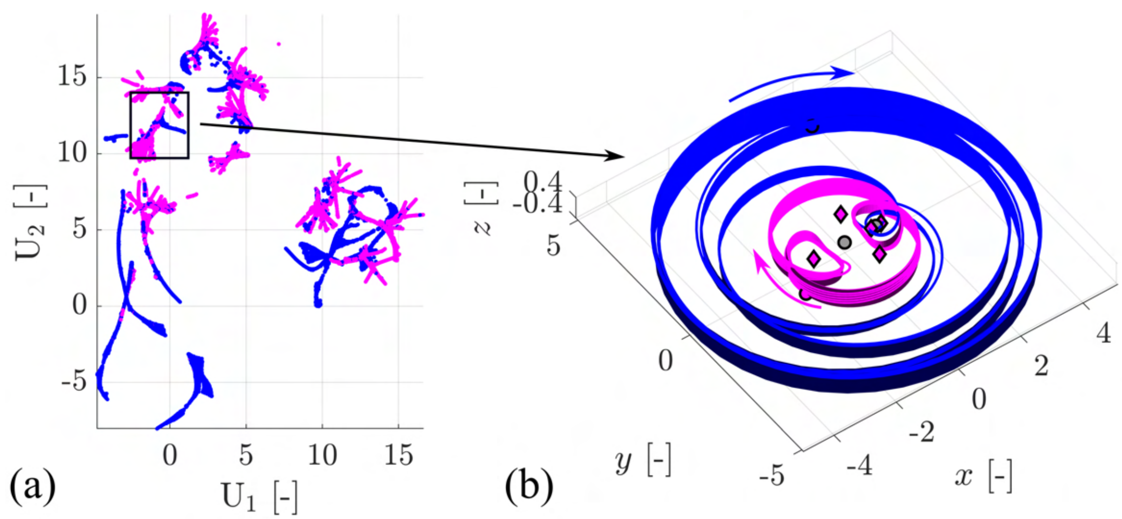

Figure 6.7: (a) UMAP projections of the crossings from the hyperbolic manifolds of two invariant tori near the 1:5 (blue) and the 2:3 (magenta) resonances with the $y = 0$ plane; (b) natural transfer departing the torus near the 2:3 resonance and naturally approaching the 1:5 torus near resonance, initialized by leveraging the UMAP projection, in the Earth-Moon system at $C_J = 2.6$

6.3 Generating Transfers Between Families of Invariant Tori near Resonances

In this section, continuation is used to generate families of transfers with similar geometry across multiple invariant 2-tori near resonance in the Earth-Moon system. To demonstrate the continuation approach detailed in Sec. 6.1, the families of tori near the 3:2 and 1:2 resonances at $C_J = 2.73$ are used. In particular, the natural transfer T1 in Fig. 6.4(a) is considered in a first investigation. Continuation is applied to compute transfers with a similar geometry between unique pairs of invariant tori. However, the continuation scheme is not used to find similar transfers connecting the tori at various longitudinal and transverse angles. Following the application of the continuation procedure to transfer T1 in Fig. 6.4(a), Fig. 6.8 presents a summary of the converged family of natural transfers for this particular transfer geometry, naturally departing members from the family of invariant tori near the 3:2 resonance and naturally approaching members from the family of invariant tori near the 1:2 resonance.

In the top-right inset of Fig. 6.8, a grid-like representation summarizes the obtained transfers. The horizontal and vertical axes depict the maximum out-of-plane component of the position vector at apogee along the departure and arrival tori, respectively. Each black circle in this plot indicates the existence of a feasible natural transfer, computed with the optimization problem summarized in Eq. (6.2) and satisfying the requirements $f(\boldsymbol{V}) \leq 10^{-12}$ and $\|\boldsymbol{F}(\boldsymbol{V})\|_2 \leq 5 \times 10^{-14}$. From the grid-like representation, four feasible transfers are highlighted with red circles, labeled as A1 to A4, and depicted in four insets in Fig. 6.8. In each inset depicting a transfer in black, the Earth and the Moon appear as gray circles, together with the Earth-Moon Lagrange points as magenta diamonds. The xy-projections in each inset demonstrate that the overall transfer geometry is consistent throughout the family. However, as observed in the gray lateral xz- and yz-projections, each transfer connects invariant tori with different out-of-plane displacements. For example, transfer A1 connects the last members of the 3:2 and 1:2 families of invariant tori, which present the largest out-of-plane displacement at the apoapses. Therefore, the A1 transfer exhibits the largest out-of-plane motion. Conversely, transfer A3 presents almost a planar transfer, since it connects the members of the two families with relatively small out-of-plane component at apoapsis. The remaining two transfers, labeled as A2 and A4, connect invariant tori with large relative difference in the maximum out-of-plane displacement at apoapsis. In particular, transfer A2 naturally connects a member of the family of tori near the 3:2 resonance with a relatively low out-of-plane displacement at the

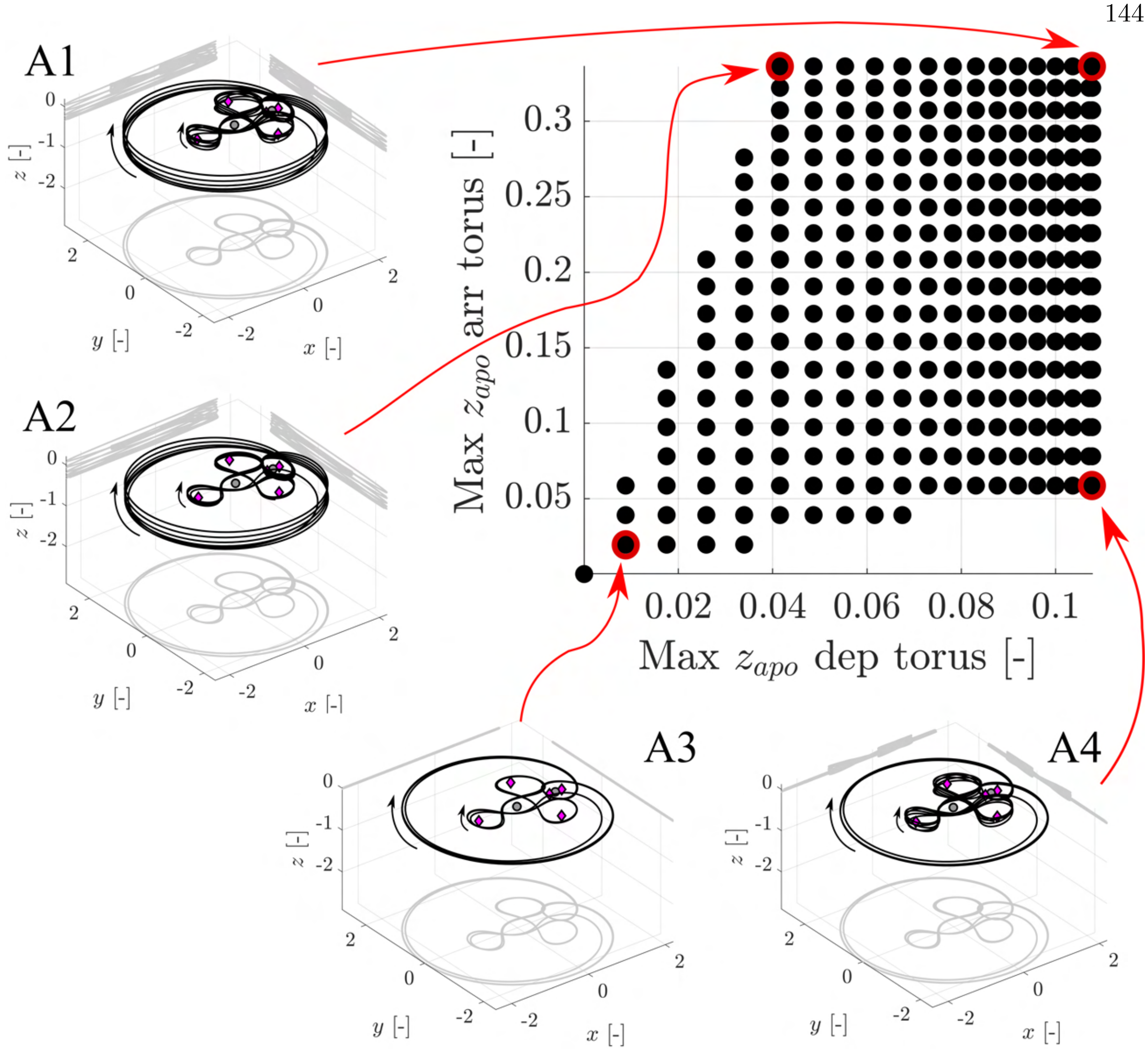

Figure 6.8: Summary of the continuation scheme for natural transfers continued from Fig. 6.4(a). Black markers reflect a natural transfer from a departing 3:2 invariant torus to an arrival 1:2 invariant torus, identified by the maximum out-of-plane displacement at apoapsis. Four sample transfers are visualized in the boundaries and indicated with red circles in the top-right grid.

apoapsis, with a member of the family of tori near the 1:2 resonance with a relatively large out-of-plane displacement at the apoapsis. Conversely, transfer A4 exhibits an initial large out-of-plane displacement, culminating with almost planar motion. Note that transfers A1 and A4 are localized in the same column in Fig. 6.8, therefore they are associated to the same departing torus:

although presenting solutions departing the same structure, the initial spacecraft states of these transfers along the departing torus differs to accommodate trajectories with distinct out-of-plane displacements. Likewise, transfers A1 and A2 share the same row in the grid-like representation, thus presenting transfers approaching the same arrival torus, although culminating at different locations. Analysis of Fig. 6.8 reveals useful insights into the existence of natural transitions between invariant tori within each family. Specifically, given a fixed initial invariant torus near the 3:2 resonance, a transition only exists to selected arrival tori near the 1:2 resonance and vice versa. For this particular transfer geometry, the existence of these transitions appears to be linked to the relative difference in the maximum out-of-plane component along each invariant torus. Moreover, for initial invariant tori near the 3:2 resonance with a small out-of-plane deviation at apoapsis, only tori near the 1:2 resonance with a small out-of-plane component are naturally accessible. As the tori evolve along each family, expanding their maximum out-of-plane displacement at apoapsis, natural transitions occur at a larger range of differences in the maximum out-of-plane components.

Analysis of the existence of natural transfers with a similar geometry to the T4 transfer in Fig. 6.4(d) is performed for the families of invariant tori near the 3:2 and 1:2 resonances at $C_J = 2.73$. The T4 transfer presented in Fig. 6.4(d), naturally connects the last computed members of the investigated families of tori. However, there is an evident geometric difference between transfer T1 and T4, in Fig. 6.4(a) and (d) respectively. Indeed, the transfer labeled as T4 presents a transient with multiple loops in the configuration space around L_4 prior to a final flyby with the Moon that directs the spacecraft towards a natural approach to the torus near the 1:2 resonance. To investigate the associated family of transfers, transfer T4 is leveraged as an initial guess to start the continuation scheme outlined in Sec. 6.1. The results of the continuation scheme are summarized in Fig. 6.9 using a configuration consistent with Fig. 6.8. A top-right grid-like visualization represents each constructed natural transfer presenting a similar geometry to Fig. 6.4(d). Moreover, the phase space representation of four natural transfers is highlighted in four insets at the boundaries of the figure, connected by red arrows to the associated red-circled dots in the grid. The transfer appearing on the top-left, and labeled with the identifier B1, naturally connects the invariant tori with the largest out-of-plane displacement at apoapsis, while transfer B3 presents almost a planar transfer, since it connects the members of the two families with a relatively small out-of-plane component at apoapsis. Transfers B2 and B4 complete the figure, and present trajectories naturally connecting tori with relatively large differences for the out-of-plane displacement. Analysis of Fig. 6.9 reveals that this family of transfers accounts for a larger number of feasible solutions satisfying the optimization requirements. In particular, more solutions appear on the second and third row and columns of Fig. 6.9 with respect to the analogous of Fig. 6.8. These solutions connect tori with relative large difference in out-of-plane displacement. Such a result is likely motivated by the multiple close flybys performed by transfer T4 with respect to transfer T1 in Fig. 6.4. For this reason, the existence of natural transitions between spatial invariant tori near two distinct resonances appears to be influenced by the transfer geometry.

Ultimately, each family of continuous and natural transfers presented in Figs. 6.8 and 6.9 can be straightforwardly expanded by leveraging known symmetries of the CR3BP and the symmetry of the investigated families of invariant tori. Indeed, as outlined in Sec. 2.1, if a trajectory $\boldsymbol{x}(t) = [x(t), y(t), z(t), \dot{x}(t), \dot{y}(t), \dot{z}(t)]$ exists, then also $\bar{\boldsymbol{x}}(t) = [x(t), y(t), -z(t), \dot{x}(t), \dot{y}(t), -\dot{z}(t)]$ satisfies the dynamics in Eq. (2.5) [71]. Note how the existence of one family and the associated mirrored counterpart can be interpreted as two different branches bifurcating from the planar solution connecting the associated periodic orbits.

6.4 Summary of Contributions

This chapter presents an approach for generating natural connections between unstable invariant 2-tori in the CR3BP. The presented methodology first generates two families of unstable invariant tori, and records the crossings of the hyperbolic invariant manifolds emanated from two members of the families with a common surface of section. The crossings are then projected on a lower dimensional representation using UMAP, and visually investigated to retrieve a discontinuous initial guess solution. The initial guess is then corrected into a continuous trajectory that minimizes the distance of the first and last states of the transfer from the investigated tori. This solution is

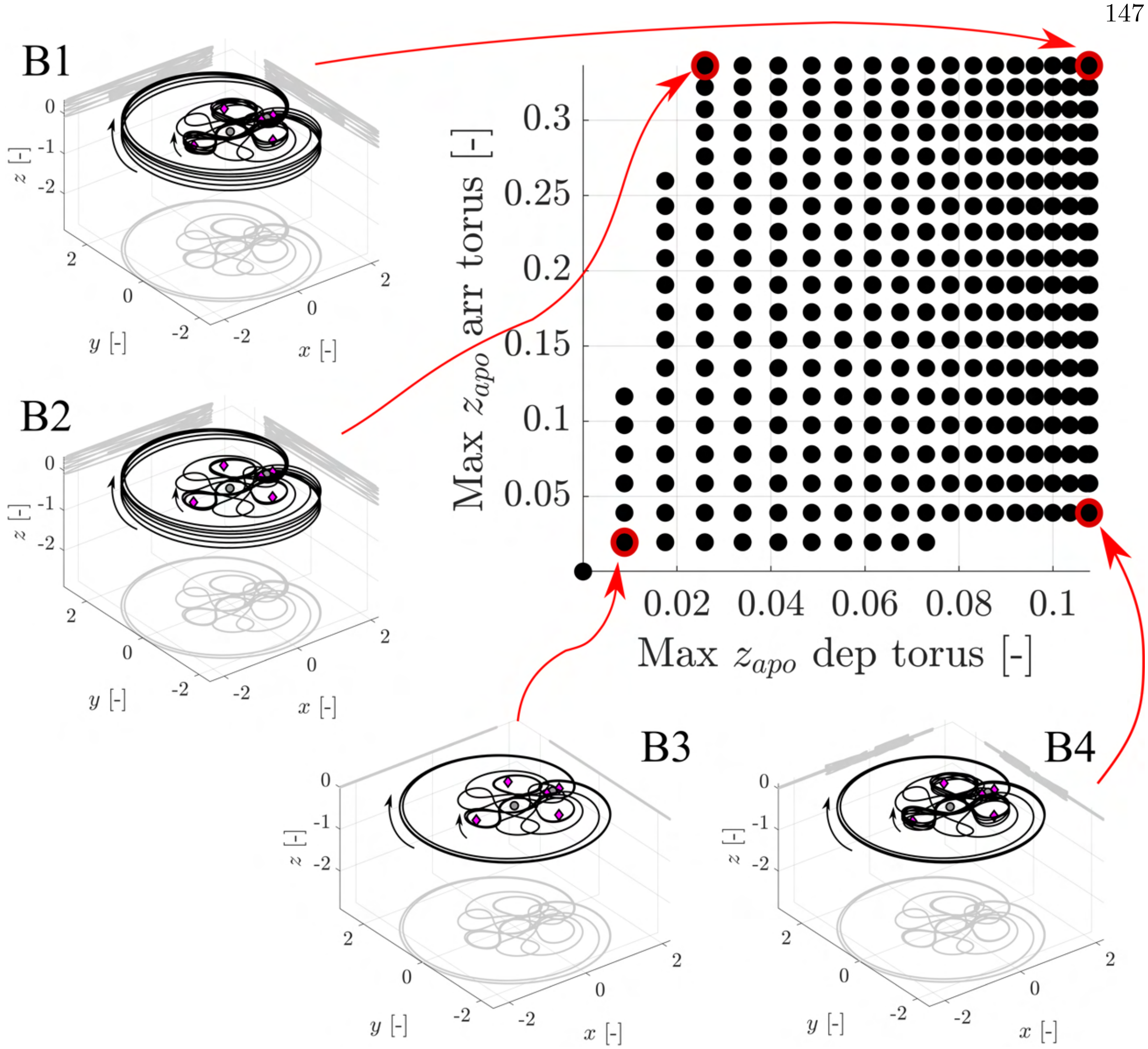

Figure 6.9: Summary of the continuation scheme for natural transfers continued from Fig. 6.4(d). Black markers reflect a natural transfer from a departing 3:2 invariant torus to an arrival 1:2 invariant torus, identified by the maximum out-of-plane displacement at apoapsis. Four sample transfers are visualized in the boundaries and indicated with red circles in the top-right grid.

continued across members of the family of invariant tori, obtaining a family of natural transfers with similar geometry. The method is demonstrated to construct transfers between tori near orbital resonances in the Earth-Moon CR3BP. This section concludes the chapter by discussing the scientific contributions of the presented results, the benefit of leveraging UMAP for projecting a higher-dimensional Poincaré map, and potential avenues for future research.

6.4.1 Scientific Contributions of the Presented Results

The presented methodology is demonstrated on a specific scenario where connections between two families of tori near orbital resonances in the Earth-Moon system are generated. The proposed technique provides four main scientific contributions:

1) Flexibility of the devised approach: the method is specifically designed to be torus-, system- and model-agnostic. Although it is demonstrated for specific scenarios in the Earth-Moon CR3BP, the technique can be adapted to connect: invariant tori near orbits that do not lie near resonances, for example quasi-halo trajectories; invariant tori in different dynamical systems and models, for example in the Sun-Earth CR3BP, as well as in more complex models; an invariant torus with a trajectory arc departing or approaching another periodic orbit. Moreover, the proposed methodology eliminates the necessity of constraining the location or the shape of the tori.

2) A-priori definition of the connected tori: both departing and arrival structures must be determined before initializing the method. This enables a trajectory designer to specify both the departing and arrival structure, aiding for the construction of end-to-end trajectories.

3) Use of UMAP for higher-dimensional Poincaré maps: UMAP is leveraged to reduce the dimensionality of the dataset of manifold crossings, enhancing visualization. The projected dataset can aid the trajectory designer in the identification of multiple natural connections between distinct invariant tori.

4) Generation of a family of natural connections between invariant tori: this work presents a technique for generating families of natural connections between families of invariant tori near orbital resonances.

6.4.2 Benefits of Manifold Learning for Higher-Dimensional Poincaré Maps

Manifold learning is employed in this chapter to reduce the dimensionality of a dataset, populated by five-dimensional crossings of manifold arcs with a surface of section. The generated projection aids an astrodynamicist to investigate areas of a Poincaré map where a natural connection between tori might exist. The application of manifold learning to the problem of visualization of a Poincaré map presents a variety of benefits, including:

1) Enhanced visualization: the information contained in a dataset populated by five-dimensional datapoints is projected and visualized through a two-dimensional representation, preserving the original global structure of the higher dimensional dataset. This process allows an astrodynamicist to obtain an understanding of the global structure of the available solution space with a simple planar representation.

2) Improved computational efficiency: the computational complexity of UMAP is $\mathcal{O}(n^{1.14})$ [32]. The enhanced visualization allows to investigate only specific regions of the dataset, reducing the overall computational effort of the searching algorithm. If a brute-force approach is selected, the searching on the selected regions has a computational complexity of $\mathcal{O}(n_{\text{sub}}^2)$, while the same algorithm on the original dataset is $\mathcal{O}(n^2)$, with $n_{\text{sub}} \ll n$. In general, including UMAP may benefit the computational cost if the complexity of the searching algorithm is larger than $\mathcal{O}(n^{1.14})$. Additionally, the lower-dimensional representation reduces the computational complexity associated with the distance metric used to compare two points in the dataset: instead of computing distances that employ the original higher-dimensional solution space, the method can use the coordinates of the lower dimensional representation.

6.4.3 Avenues for Future Work

The approach presented in this chapter may benefit from ongoing work, including:

1) According to the definition of UMAP in Sec. 4.1.2, the higher- and lower-dimensional

datasets have a similar, yet not identical, topological configuration. Therefore, the distance between two datapoints in the lower-dimensional representation might be distorted when compared to the distance between the same crossings in the higher-dimensional dataset. In the extreme case, if the original dataset contains two identical datapoints, their lower-dimensional representations might be placed in two distinct, yet nearby, configurations in the lower-dimensional representation. This challenge can potentially be solved by leveraging the parametric version of UMAP, that trains a neural network for learning the embedding [32]. Investigation of alternative approaches or further analysis of this phenomenon represents the scope of future research.

2) Using UMAP to reduce the dimensionality of a dataset requires a new set of expertise from what usually possessed by a trajectory designer. Specifically, the selection criteria for the leveraged input parameters are dataset- and problem-dependent. A correct final selection requires the trajectory designer to understand the intrinsic mechanism behind UMAP, and to iteratively explore different input parameter sets. Thus, future work might include a thorough parameter exploration to aid a trajectory designer in the projection of high-dimensional crossings of interesting trajectories.

Chapter 7

Autonomous Maneuver Design with Reinforcement Learning

Maneuvers are often implemented to alter a trajectory path and enable a spacecraft to fulfill the mission requirements. Among the variety of scenarios where maneuvers represent a fundamental asset for a mission, this chapter focuses on station-keeping and orbital transfer maneuver design. Specifically, sequences of station-keeping maneuvers are designed to enforce bounded motion with respect to a reference trajectory. The sequence of maneuvers is necessary to avoid the spacecraft natural departure due to the underlying governing dynamics. Conversely, maneuvers for an orbital transfer scenario are leveraged to transition a spacecraft from a starting orbit to an arrival path. Different approaches have been designed to identify sequence of maneuvers for station-keeping and orbital transfer scenario, especially leveraging techniques from traditional optimization and dynamical systems theory [38, 53]. However, these approaches often require significant human involvement throughout each phase of the trajectory generation process, and large computational resources to compute the required sequence of maneuvers. In this chapter, reinforcement learning is leveraged to train policies for autonomous generation of station-keeping and orbital transfer maneuvers. The trained policies allow to reduce human workload and computational requirements during the fast-paced phases of trajectory design process. In particular, policies for autonomous station-keeping maneuver design in a higher-fidelity point mass ephemeris model are investigated in Sec. 7.1, followed by the main results on an initial exploration of reinforcement learning to different orbit transfer scenarios in the Earth-Moon CR3BP in Secs. 7.2 and 7.3. Ultimately, a summary of this chapter, and a thorough description of the potential impact of the demonstrated approach for autonomous maneuvering in multi-body systems is presented in Sec. 7.4.

7.1 Station-Keeping Maneuver Design in the Sun-Earth System

In this section, a reinforcement learning algorithm is leveraged to train and evaluate a policy for station-keeping maneuver design in a point mass ephemeris model. First, a baseline station-keeping scenario, modeled after the Nancy Grace Roman Space Telescope, is introduced in Sec. 7.1.1. The station-keeping problem is then translated into a reinforcement learning environment in Sec. 7.1.2, identifying a step, an episode, an action and a state vector, and a reward formulation. To favor a comparison with traditional optimization, a constrained optimization scenario is formulated and presented in Sec. 7.1.3. Ultimately, the defined reinforcement learning scenario is used to train and evaluate a policy in a point mass ephemeris model in Sec. 7.1.4, compared with the performances of the constrained optimization in Sec. 7.1.5.

7.1.1 Scenario Overview

A scenario modeled after the Nancy Grace Roman Space Telescope is used as a framework for a spacecraft operating near a Sun-Earth L_2 southern quasi-halo trajectory. In this framework, the spacecraft is assumed possessing a cross-sectional area of 49.6 m^2, a dry mass of 6877 kg and full reflectivity $k = 2$ [38]. The modeled scenario requires the spacecraft to exhibit bounded motion with respect to a southern halo orbit nearby L_2 in the Sun-Earth system, with a period of $T_r \approx 180$ days, corresponding to a periodic orbit with a value of Jacobi constant $C_J = 3.00078$, displayed in Fig. 7.1(a) in the Sun-Earth rotating frame using dimensionless coordinates relative to the Earth. When a point mass ephemeris model is used to generate a reference path for the spacecraft, the correction scheme summarized in Sec. 3.8 is leveraged. The orbit displayed in Fig. 7.1(a) is employed as a first guess to construct a nearby continuous trajectory in a point mass ephemeris model, incorporating the gravitational influence of the Sun, the Earth, and the Moon, augmented by SRP. The reference trajectory is defined for a time span $[e_0, e_f] = [29390.308, 37066.086]$ MJD, corresponding to the period between 24 Jun 2021 19:22:22 and 30 Jun 2042 14:02:41 UTC, and it is visualized in Fig. 7.1(b) in the dimensionless Sun-Earth rotating frame, with the Earth at the origin. Thanks to the leveraged correction scheme, the corrected trajectory represents a continuous arc in a higher-fidelity dynamical model, with minimized state deviation from the generating halo orbit near L_2 in the Sun-Earth CR3BP of Fig. 7.1(a).

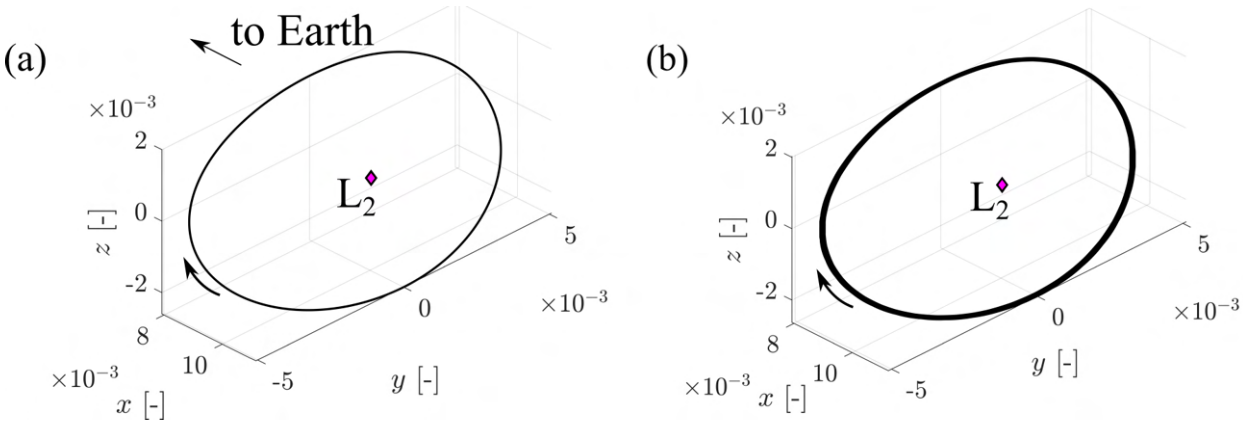

Figure 7.1: Reference trajectories near the Sun-Earth L_2: (a) southern halo orbit in the CR3BP and (b) nearby quasi-halo bounded trajectory in a Sun-Earth-Moon ephemeris model, with minimized state-wise distance from (a) and spanning a period $[e_0, e_f] = [29390.308, 37066.086]$ MJD.

Typically, impulsive station-keeping maneuvers are used to mitigate the impact of uncertainties, momentum unloads, and off-nominal maneuvers on the spacecraft trajectory. These motivate the design of robust maneuver planners. This investigation only considers regular momentum unloads, modeled as instantaneous changes in velocity, denoted $\Delta \boldsymbol{v}_{\mathrm{MU}}$. Moreover, the modeled scenario assumes full state knowledge and nominal maneuver performance. According to the Nancy Grace Roman Space Telescope scenario, each momentum unload maneuver is defined as a three-dimensional instantaneous change in velocity, with magnitude of 8.7 mm/s and a random direction. A sequence starting with one station-keeping maneuver and continuing with three consecutive momentum unload maneuvers and coast arcs corresponds to a momentum unload cycle. In this cycle, a coasting arc of $t_{\mathrm{MU}} = 130$ hours is assumed between two consecutive maneuvers, therefore obtaining a single momentum unload cycle in 520 hours [38]. Differently from the momentum unload maneuvers, no constraint is applied to the magnitude of the station-keeping maneuvers.

7.1.2 Reinforcement Learning Problem

Translating a problem into an RL scenario requires the definition of various components that are leveraged to train a policy. Specifically, an RL scenario is characterized by the definition of: observation and action vectors, a reward, a step and an episode, and an environment. In this investigation, the RL environment for the station-keeping scenario is formulated to be agnostic of the leveraged dynamical model. Therefore, the environment can be used to train policies in both low- and high-fidelity dynamical models. In the designed environment, a single episode constitutes up to τ steps, as indicated by the blue arcs in Fig. 7.2. At the beginning of each step, the spacecraft state, marked with a gray circle in Fig. 7.2, is corrected by an impulsive maneuver $\Delta\boldsymbol{v}$. Each maneuver is generated by the actor policy, and it is marked by a violet arrow in Fig. 7.2. During training, the corrected spacecraft state is then propagated for a time $\Delta t = 520$ hours, corresponding to one momentum unload cycle. During the testing phase of the learned policy in the long-term station-keeping scenario, the spacecraft path in each step is perturbed by three consecutive momentum unload maneuvers. The information concerning the state of the spacecraft and the current epoch is accessed at the beginning of each step through the continuous environment state vector $\boldsymbol{s}$. This state vector for the station-keeping environment is defined as:

$$\boldsymbol{s} = [\tilde{\boldsymbol{x}}_{\text{ref}}, \delta\tilde{\boldsymbol{x}}, \tilde{t}_E, \tilde{s}] \in \mathbb{R}^{14} \tag{7.1}$$

and it is composed of the spacecraft state along the reference trajectory $\tilde{\boldsymbol{x}}_{\text{ref}}$ that is associated to the current epoch, depicted as a black dot along the reference path in Fig. 7.2, the isochronous deviation of the spacecraft state from the reference state, $\delta\tilde{\boldsymbol{x}}$, marked by black arrows in Fig. 7.2, the current epoch $\tilde{t}_E$, and a cardinal identifier of the maneuver number along the current episode $\tilde{s}$. The tilde operator leveraged in Eq. (7.1) indicates a scaled quantity, used to generate components of the environment state vector that generally present values in the range $[-1, 1]$ during training. Each non-terminal step begins with an environment state vector corresponding to the current state and epoch of the spacecraft. The environment and spacecraft configuration at the beginning of each step is computed at termination of the preceding step. The initial step for each episode initializes the environment state vector with a model-dependent approach. If the dynamical model used for propagating the spacecraft state is the low-fidelity CR3BP, an initial state along the reference orbit $\tilde{\boldsymbol{x}}_{\text{ref}}$ is randomly selected and the initial epoch randomly assigned from the range $[e_0, e_f]$ MJD. If the point mass ephemeris model is employed for the propagation of the spacecraft state in the environment, each episode is initialized by randomly selecting the initial epoch, t_E to produce the corresponding spacecraft state vector, $\tilde{\boldsymbol{x}}_{\text{ref}}$ along the reference trajectory. A relative state vector, $\delta\tilde{\boldsymbol{x}}$, is then randomly generated in both approaches to initialize the spacecraft state within an hypersphere of the reference state $\tilde{\boldsymbol{x}}_{\text{ref}}$. In particular, each component of the relative state vector is randomly sampled from a continuous uniform distribution in the range $[-1, 1]$. The order of magnitude of the perturbation is selected from a triangular distribution between -5 and 0.2 and centered at 0, multiplied by a scaling factor of 300 km and 5.96 mm/s in each position and velocity component to obtain the final perturbation. This triangular distribution is designed to encourage episode initialization near the reference path, representing a fundamental asset for the long-term station-keeping scenario with dynamical perturbations. Ultimately, the step number identifier is computed at each step as $\tilde{s} = 2i/\tau - 1$, where $i \in \{\tau, \tau - 1, \ldots, 0\}$ is the current step number across the generated episode. The step number identifier is initialized with a value $\tilde{s} = 1$ at the beginning of the initial step of an episode, and culminates at a value $\tilde{s} = -1$ at the end of the τ-th step. The quantity $\tilde{s}$ largely affects the training process, because it helps the critic to distinguish between similar spacecraft states that occur at different time steps along different episodes.

The environment state vector is leveraged at the beginning of each step to generate an action $\boldsymbol{a} \in \mathbb{R}^3$. The state is input to the policy, generating as a output a three-dimensional mean and a three-dimensional standard deviation. During training, the action is sampled from a multi-variate Gaussian constructed with the obtained three-dimensional mean and standard deviation. During testing, the action is defined only using the mean of the distribution. This approach enables exploration during training, and exploitation at testing time. The generated continuous action, $\boldsymbol{a}$, is formulated as a three-dimensional vector that reflects an impulsive maneuver, $\Delta\boldsymbol{v} = \nu\boldsymbol{a}$ in the Sun-Earth rotating frame. The scaling parameter $\nu \approx 0.14$ m/s is selected according to the

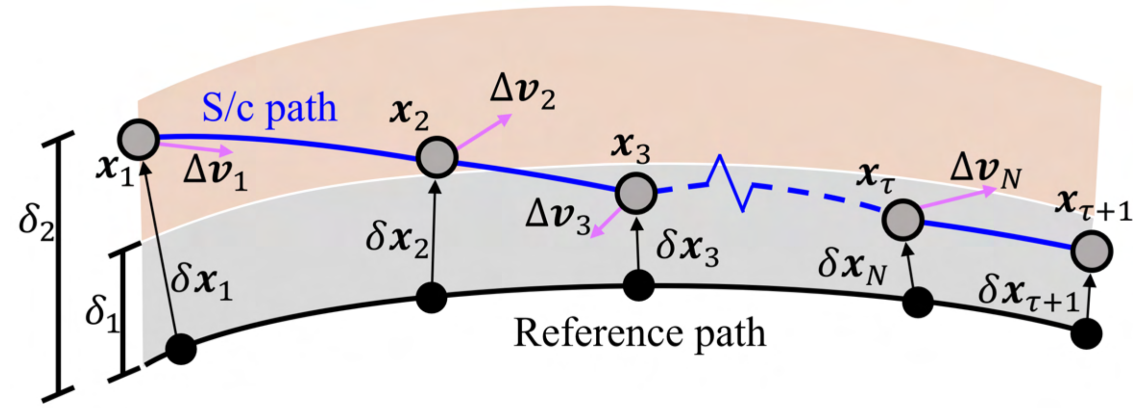

Figure 7.2: Conceptual representation of the regions separating different reward formulations.

investigate system, and favors the generation of actions within $[-1, 1]$. Then, the spacecraft state is augmented with the constructed $\Delta \boldsymbol{v}$. At the end of the step, the current state and epoch of the spacecraft is used to update the environment state $\boldsymbol{s}$, used at the beginning of the subsequent state of the current episode.

Lastly, the reward is formulated to simultaneously consider the information associated with the state of the spacecraft at the beginning and the end of the step. In this investigation, reward shaping is applied to aid the policy learn a behavior that favors: bounded motion with respect to the reference trajectory; and reduced maneuver magnitude. The reward is then composed of three formulations, each associated with a specific spatial region nearby the reference state.

1) Exterior region: when at the end of the considered step the spacecraft position exceeds a relative threshold value $\|\delta \boldsymbol{r}(t_0 + \Delta t)\| > \delta_2$, the spacecraft is considered in the exterior relative region. In this work, the threshold is selected as $\delta_2 \approx 1.67 \times 10^{-5}$, corresponding to a position error magnitude of up to 2499 km. The external region is represented by the area above the orange section in Fig. 7.2. When the spacecraft is in the exterior region, a large negative reward is assigned to the analyzed state-action pair. The value of the assigned reward reflects a heavy penalization due to the significant departure from the reference trajectory. After assigning the reward to the state-action pair, the associated episode is also terminated.

2) Central region: when the spacecraft position at the end of the considered step is within the thresholds $\delta_1 \leq \|\delta \boldsymbol{r}(t_0 + \Delta t)\| \leq \delta_2$, the spacecraft is considered in a central relative region. A value $\delta_1 \approx 6.68 \times 10^{-7}$, corresponding to a relative distance of 100 km, is used in this work. In this configuration, the spacecraft is considered in an acceptable bounded region near the reference path, but not within a desired region. The reward in the central region is formulated as a weighted sum of two terms: a quantity associated with the relative distance of the spacecraft from the reference trajectory, measured isochronously; a value associated with the amount of propellant consumed for the maneuver at the beginning of the step. The central region is represented as an orange area in Fig. 7.2.

3) Internal region: when the spacecraft position is within a threshold $\|\delta \boldsymbol{r}(t_0 + \Delta t)\| \leq \delta_1$, the spacecraft is considered in the internal relative region. The reward is formulated to reduce the maneuver magnitude and maintain the spacecraft within the internal region. The mathematical definition of the reward follows the description used for the middle region, since it is composed by the weighted sum of two terms. However, two modifications are included for the term associated with the displacement from the reference trajectory. Independently from the actual full-state relative distance, the positional distance from the reference is fixed to δ_1. Moreover, the term associated with the spacecraft displacement from the reference trajectory is scaled by a value $B = \ln(\delta_1/2)/\ln(\delta_1) \approx 1.05$. Empirically, it is observed that larger values for the scaling parameter, B, result in policies that rapidly drive the spacecraft within the internal region. The internal region is represented by the gray area in Fig. 7.2.

Based on the distinction of three fundamental regions of motion, the mathematical formulation of the piece-wise reward is:

$$r = \begin{cases} -B\ln(\delta_1) + K(1 - \|\boldsymbol{a}\|) & \text{if } \|\delta\boldsymbol{r}(\Delta t)\| \leq \delta_1 \\ -\ln(\|\delta\boldsymbol{r}(\Delta t)\|) + K(1 - \|\boldsymbol{a}\|) & \text{if } \|\delta\boldsymbol{r}(\Delta t)\| \leq \delta_2 \wedge \|\delta\boldsymbol{r}(\Delta t)\| > \delta_1 \\ -100 & \text{otherwise} \end{cases} \tag{7.2}$$

In this formulation, a relative weighting factor $K = 15$ is selected to balance the effects of the isochronous state deviation of the perturbed path at the end of the time step relative to the reference, $\delta\boldsymbol{r}(t_0 + \Delta t)$, and the maneuver magnitude. In the piece-wise reward formulation of the internal and central region, the isochronous spacecraft displacement is processed through a natural logarithm. A formulation with a negative natural logarithm is selected due to the continuous and unbounded characteristic of the function for an argument converging towards zero. Processing the full-state displacement through the natural logarithm generates progressively larger rewards when the state deviation converges to zero. Ultimately, a linear formulation is selected for the action contribution within the reward to provide an upper positive limit. Unbounded formulations might generate policies that prefer to not act in the considered scenario. A conceptual overview of the training process for the environment leveraging the point mass ephemeris model, composed by episode initialization, interaction between the environment and the policy, and step execution, is summarized in Fig. 7.3.

7.1.2.1 Hyperparameters and Neural Network Structure Selection

The performance of the trained policy is governed by a set of training parameters and the designed structure of the neural networks, forming together a set of hyperparameters that must be selected. The training parameters govern the training process, and are often selected to balance the trade-off between exploration and exploitation, allowing the generation of a final and converged policy that maximizes the value. The structure of the neural networks heavily impacts the range and the quality of the network outputs: the width and depth of the networks are crafted to balance the bias-variance trade-off, preventing potential overfitting of wider and deeper network

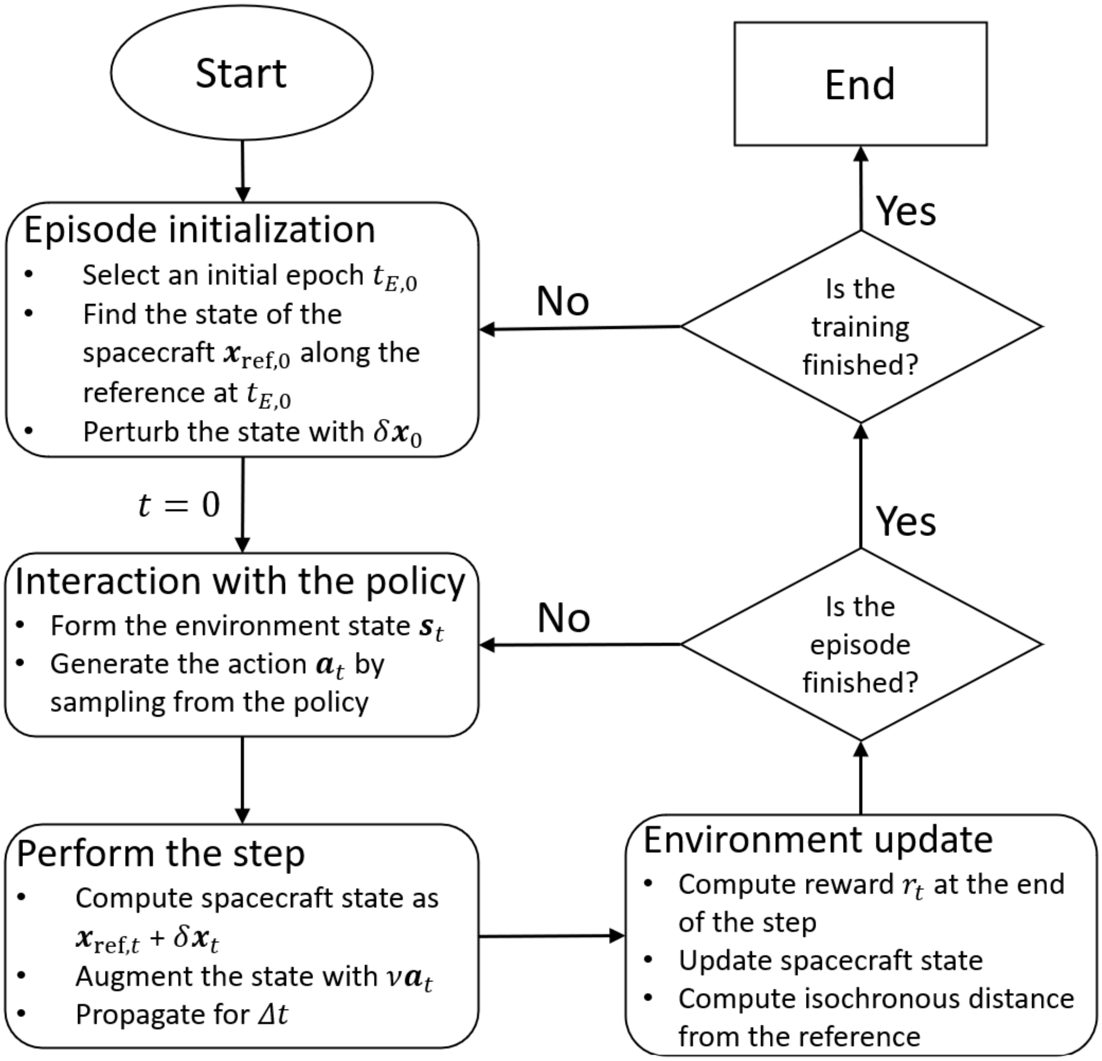

Figure 7.3: Flowchart of the station-keeping point mass ephemeris RL environment.

and the underfitting of more shallow and smaller configurations. Different strategies exist for tuning these hyperparameters, including: random search, grid search, Bayesian optimization, and using an outer RL-based implementation [134, 135]. This investigation optimizes these hyperparameters through Bayesian optimization: this optimization scheme is selected due to its demonstrated sample efficiency for problems requiring large computational resources for the cost function evaluation, such as those that learn an efficient RL policy operating in chaotic environments. Moreover, in the computation of the associated objective function, Bayesian optimization does not compute the derivatives of the objective with respect to the input hyperparameters, and represents an advantageous optimization scheme for problems with unknown closed-form expression of the cost function and with less than 20 dimensions [134, 135].

Bayesian Optimization: an Overview

In Bayesian optimization, an unknown cost function $f : \mathbb{T} \rightarrow \mathbb{R}$ over a compact hyperparameter set $\mathbb{T}$, is considered as a random function with a set prior $p(f)$. The prior represents the belief on the space of possible cost functions. When a new sample of the input $\boldsymbol{\vartheta}_i \in \mathbb{T}$ is drawn, the function evaluation is obtained as $f(\boldsymbol{\vartheta}_i)$. Accumulated knowledge on observations, corresponding to sampled inputs and associated function evaluations $\boldsymbol{D}_i = \{(\boldsymbol{\vartheta}_0, f(\boldsymbol{\vartheta}_0)), \ldots, (\boldsymbol{\vartheta}_i, f(\boldsymbol{\vartheta}_i))\}$, allows the definition of a likelihood of the observations given the cost function $p(\boldsymbol{D}_i|f)$. The prior on the function space and the likelihood can be combined leveraging Bayes' theorem to retrieve the posterior as:

$$p(f|D_i) \propto p(\boldsymbol{D}_i|f)p(f) \tag{7.3}$$

The posterior represents an updated belief about the unknown cost function, and it is used to construct an acquisition function. The latter is leveraged to obtain the next sampled input $\boldsymbol{\vartheta}_{i+1}$, balancing exploration of areas with large uncertainty with the exploitation of regions where the expected cost function is large. Among the variety of feasible models used for defining the prior $p(f)$, Gaussian processes (GP) priors represent an often selected alternative due to their favorable characteristics [134, 136]. A GP assumes the distribution on the cost function is a jointly multivariate Gaussian, allowing to express the posterior mean and covariance function as a function of the observed $\boldsymbol{D}_i$. Different kernels can be used to express the covariance function, with popular examples including the squared exponential functions and the Matérn kernel [134, 136]. When a new input set $\boldsymbol{\vartheta}_i$ is sampled and explored to generate the associated objective value, the algorithm increases its confidence in a neighborhood of the visited set. In the proposed investigation, new samples are drawn from the acquisition function until a specified termination condition is satisfied.

Bayesian Optimization Strategy

In this work, Bayesian optimization is used to guide the selection of a suitable set of hyperparameters $\boldsymbol{\vartheta}_i$ governing: the objective function of PPO; the training process; and the width and depth of the neural networks used in the RL-based maneuver planner to model the actor and the critic. In this formulation, each cost function evaluation $f(\boldsymbol{\vartheta}_i)$ is associated with the performances of the associated converged actor neural network, and it is therefore evaluated at the end of a full training process. The environment used to train an RL-policy during the Bayesian optimization leverages the Sun-Earth CR3BP dynamical model to propagate the spacecraft state, reducing the overall computational effort. During optimization, a first set of N_1 hyperparameters are selected randomly from $\mathbb{T}$. This allows an initial exploration of the parameter space. Then, a second set of N_2 hyperparameters are drawn according to the acquisition function, balancing exploration and exploitation of the already partially explored parameter set $\mathbb{T}$. After a total of $N_1 + N_2$ cost function evaluations, a set of hyperparameters and neural network structures is selected from the explored collection, and employed to train policies in an environment leveraging the point mass ephemeris model to propagate the spacecraft state. This single set of hyperparameters, retrieved in an environment leveraging a low-fidelity dynamical model for the spacecraft state propagation, does not necessarily represent an optimal set for an environment employing a higher-fidelity model. However, the CR3BP represents a good approximation of a point mass ephemeris model. Therefore, this investigation assumes that the hyperparameter set selected from the optimization will result in a training process that learns policies with sufficient performances in a higher-fidelity scenario. In this work, the python toolbox *BayesianOptimization*, using the Matérn kernel and the upper confidence bound acquisition function is used to perform Bayesian optimization and guide the selection of an optimal hyperparameters set [137].

Hyperparameters and cost function design

In this investigation, the set of hyperparameters $\boldsymbol{\vartheta}$, representing the input of the cost function $f(\cdot)$, includes the parameters governing PPO, the training process and the network structures. Specifically, the parameters governing PPO and the training process are: discount factor, clipping rate, number of epochs and batches, value and entropy coefficients, and GAE factor. Moreover, to aid the training process honing in a locally optimal policy, the learning rate is modeled as a sequence of progressively decreasing steps as:

$$\alpha(u) = \alpha_0 + \left\lfloor \frac{u\, n_{\text{steps}}}{n_{\text{upd}}} \right\rfloor \frac{\alpha_f - \alpha_0}{n_{\text{steps}}} \tag{7.4}$$

where n_{upd} is the total number of updates performed on the networks' parameters $\boldsymbol{\theta}$ over one complete training, $u = \{1, 2, \ldots, n_{\text{upd}}\}$ is the current update number, α_0 and α_f are the values of learning rate at the beginning and end of the training process, respectively, and n_{steps} is the total number of steps for the learning rate over one complete training [138]. The parameters governing the selected learning rate scheduling identified in Eq. (7.4), α_0, α_f, and n_{steps}, are also included in the hyperparameter space. Eventually, the width and the depth of the neural networks used to approximate the actor and the critic are appended to the hyperparameters list. Based on the work by Andrychowicz et al. and Sullivan and Bosanac, the remaining features of the neural networks are held fixed. In particular, the nonlinearities in the neural networks are modeled with $\tanh(\cdot)$ activation functions, and orthogonal initialization is used to setup the networks' parameters at the beginning of each training [61, 139].

After sampling a set of hyperparameters $\boldsymbol{\vartheta}_i$, the objective function is computed at the end of each training process. The cost function is defined as the sum of two terms: the average discounted cumulative reward of the last training epoch and the mean derivative of the value with respect to the epoch over the last 10 epochs. This formulation favors the selection of set of hyperparameters that train effective policies with large average value at convergence, and exhibit potential improvements at the end of the training process.

Selected Hyperparameters

For this work, a total of $N_1 + N_2 = 130$ samples are leveraged for the optimization: the first $N_1 = 50$ simulations correspond to evaluations of random sets of input parameters, while the subsequent $N_2 = 80$ runs reflect iterations where the hyperparameters sets $\boldsymbol{\vartheta}_i$ are selected according to the acquisition function. During optimization, each training process accumulates a maximum of $2{\times}10^6$ time steps. The number of steps used per training process during the Bayesian optimization is set to a lower value with respect to the training process of the final policies evaluated in the next sections. This allows to reduce the computational complexity of the optimization process. However, the lower number of steps used during a single training process of the optimization process can be slightly different from the optimal hyperparameter set of an optimization performed with a larger number of steps per training. For this reason, three criteria are considered for the ultimate choice of an optimal set of hyperparameters:

1. Large objective function evaluation in the Bayesian optimization: a large objective function is associated with sets of hyperparameters that produce converged policies presenting a combination of final large average reward and potential growth for extended training.

2. Persistence of selected regions in the hyperparameters space: the acquisition function might heavily prefer exploiting regions of the hyperparameter space with reduced uncertainty, instead of exploring different regions associated with larger uncertainty. This behavior can be interpreted as a sign of near-optimum found by the optimizer in the regions of lower uncertainty.

3. General recommendations: a variety of authors apply PPO to different and complex dynamical environments. Their works also include general recommendations on optimal hyperparameters for training effective policies [61, 117, 124, 139, 140].

As an example illustrating the hyperparameter selection process, a subset of two searched hyperparameters associated with the structures of the neural networks, is presented in Fig. 7.4. In this representation, each subfigure reports two plots: the picture on the top depicts with red circles the value of the objective function on the x-axis and the associated hyperparameter on the y-axis; the representation on the bottom reports with cyan circles the iteration of the Bayesian optimization on the x-axis and the associated value of the hyperparameter on the y-axis, together with a vertical dashed red line marking the first $N_1 = 50$ simulations. These representations respectively visualize for the presented hyperparameters the first two listed criteria, associated with large objective function evaluation and persistence of the selected hyperparameter. The actor width and the critic width are reported in Fig. 7.4(a-b), respectively. In particular, large cost functions are associated with small number of neurons per layer for the actor neural network and large number of neurons for the critic network, as highlighted by the top plots in Fig. 7.4(a-b). Moreover, the associated representations on the bottom of these two subfigures highlight how the optimizer focuses on a small actor width and a large critic width for a large portion of the optimized N_2 runs. According to the listed selection criteria, the selected actor and critic width, together with the entire set of hyperparameters and neural network structures are summarized in Table 7.1, along with the ranges explored via Bayesian optimization.

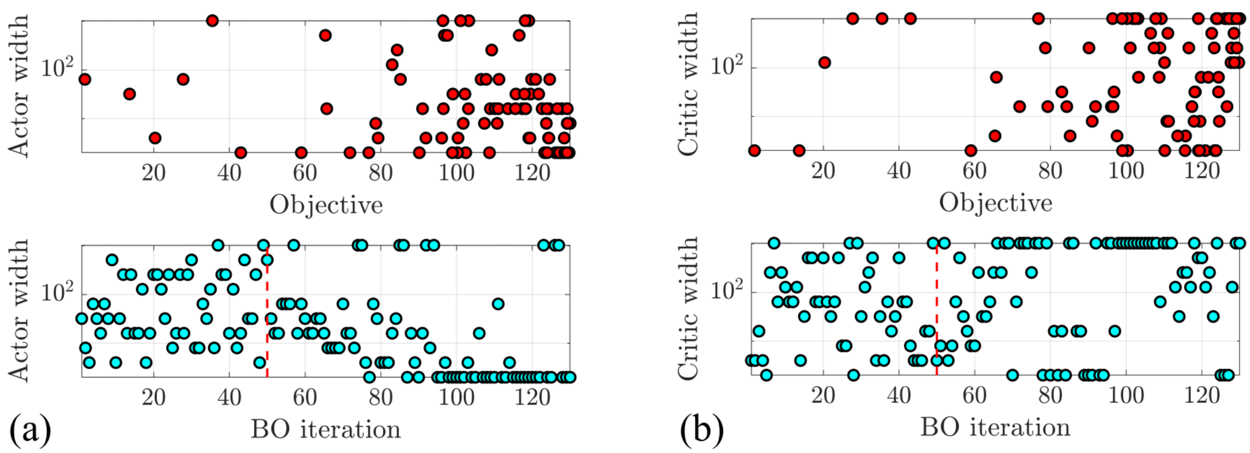

Figure 7.4: Cost function and number of the Bayesian optimization run associated with the (a) actor and (b) critic width hyperparameter exploration.

Table 7.1: Selected hyperparameters for the training and the neural network (NN) structure.

Parameter	Value	B.O. Ranges	Parameter	Value	B.O. Ranges
Initial learning rate, l_{r0}	5×10^{-3}	$[10^{-5}, 10^{-1.5}]$	Clipping parameter, ϵ	0.02	$[10^{-5}, 10^{-2}]$
Final learning rate, l_{rf}	5×10^{-4}	-	Discount factor, γ	0.99	$[0.9, 0.999]$
Learning rate schedule	8 steps	$[2, 8]$ steps	GAE factor, λ	0.99	$[0.8, 0.999]$
Number of epochs, N_e	5	[2, 10]	Actor NN depth, D_{act}	3	$[1, 6]$
Number of batches, N_b	6	[2, 10]	Actor NN width, W_{act}	16	$[2, 2^{10}]$
Value coefficient, c_1	1×10^{-3}	$[10^{-4}, 10^{-0.3}]$	Critic NN depth, D_{cri}	1	$[1, 6]$
Entropy coefficient, c_2	7×10^{-3}	$[10^{-6}, 10^{-1}]$	Critic NN width, W_{cri}	1024	$[2, 2^{10}]$

7.1.3 Translating Station-Keeping Maneuver Design into an Optimization Problem

A constrained optimization scenario is introduced to demonstrate the capability of the policy trained with PPO to generate locally optimal sequences of station-keeping maneuvers. Constrained optimization is usually leveraged to generate sequences of station-keeping maneuvers. The optimization problem is designed with a multiple shooting based approach, where a sequence of maneuvers is optimized to minimize the cumulative maneuver magnitude, given an initial fixed spacecraft state and epoch. The point mass ephemeris model is used to propagate the spacecraft state in the multiple shooting scheme. A general constrained optimization problem is formulated as:

$$\boldsymbol{V} = \arg\min_{\boldsymbol{V}} f(\boldsymbol{V}) \quad \text{subject to} \quad \boldsymbol{F}(\boldsymbol{V}) = \boldsymbol{0},\ \boldsymbol{G}(\boldsymbol{V}) \leq \boldsymbol{0} \tag{7.5}$$

where the free variable vector is $\boldsymbol{V}$, the cost function is $f(\boldsymbol{V})$, the equality nonlinear constraint vector is $\boldsymbol{F}(\boldsymbol{V})$ and the inequality nonlinear constraint vector is $\boldsymbol{G}(\boldsymbol{V})$. A conceptual visualization of this optimization problem, supporting the discussion in this section, is reported in Fig. 7.5.

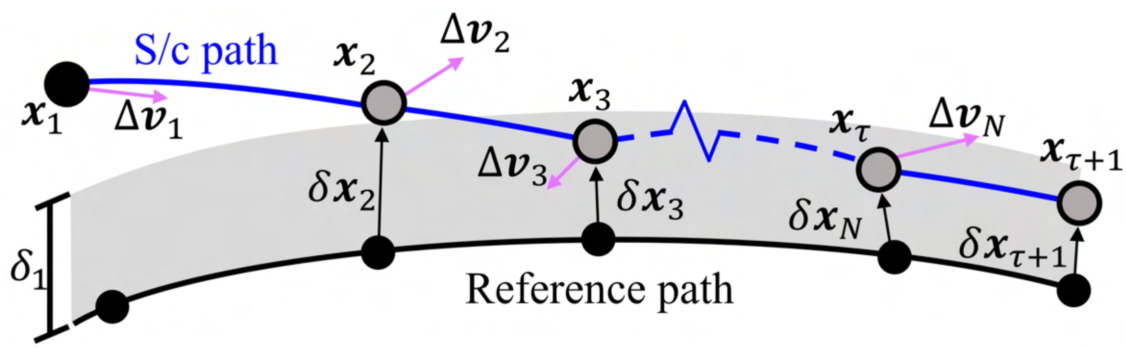

Figure 7.5: Conceptual representation of the optimization scenario.

The equality and inequality nonlinear constraint vectors $\boldsymbol{F}(\boldsymbol{V})$ and $\boldsymbol{G}(\boldsymbol{V})$ are defined to generate a converged solution presenting continuity of the trajectory at the nodes and a sequence of spacecraft states within the internal region, marked by the threshold relative distance δ_1 from the reference trajectory. The mathematical formulation of the equality constraint vector is:

$$\boldsymbol{F}(\boldsymbol{V}) = \begin{bmatrix} \boldsymbol{x}(t_1; \Delta t) - \boldsymbol{x}_2 \\ \boldsymbol{x}(t_2; \Delta t) - \boldsymbol{x}_3 \\ \vdots \\ \boldsymbol{x}(t_{\tau-1}; \Delta t) - \boldsymbol{x}_\tau \end{bmatrix} \in \mathbb{R}^{6(\tau-1)} \tag{7.6}$$

where $\boldsymbol{x}(t_i; \Delta t)$ represents the spacecraft state at the end of each arc τ. This spacecraft state is computed by augmenting the spacecraft state at the beginning of each arc $\boldsymbol{x}_i$ with the maneuver $\Delta \boldsymbol{v}_i = \nu \boldsymbol{a}_i$, and propagating for a time Δt with the point mass dynamical model. The nonlinear inequality constraint is formulated to enforce the spacecraft position at the end of each arc to lie within the internal region, associated with a relative distance δ_1 from the reference trajectory. The mathematical formulation of the inequality constraint is:

$$\boldsymbol{G}(\boldsymbol{V}) = \begin{bmatrix} \|\boldsymbol{r}_2 - \boldsymbol{r}_{\text{ref},2}\| - \delta_1 \\ \|\boldsymbol{r}_3 - \boldsymbol{r}_{\text{ref},3}\| - \delta_1 \\ \vdots \\ \|\boldsymbol{r}(t_\tau; \Delta t) - \boldsymbol{r}_{\text{ref},\tau+1}\| - \delta_1 \end{bmatrix} \in \mathbb{R}^{\tau} \tag{7.7}$$

where $\| \cdot \|$ represents the ℓ^2-norm, $\boldsymbol{r}_i$ is the nondimensional position vector of the state $\boldsymbol{x}_i$ in the non-pulsating Sun-Earth frame, and $\boldsymbol{r}(t_\tau; \Delta t)$ is the position vector of the spacecraft at the end of the τ-th arc. Formulating an inequality constraint to enforce bounded motion with respect to the reference trajectory, as opposed to incorporating boundedness in the reward formulation as outlined in Eq. (7.2), produces a different solution between the RL-based and the constrained optimization approach. However, in a traditional station-keeping maneuver design scheme, boundedness is often formulated as a constraint [38,52]. A tolerance of 5×10^{-14} is defined for satisfaction of the equality constraints.

Eventually, the objective function is formulated to minimize the cumulative maneuver magnitude over the maneuver sequence. The mathematical formulation of the cost function is :

$$f(\boldsymbol{V}) = \sum_i \boldsymbol{a}_i^T \boldsymbol{a}_i \tag{7.8}$$

Although different from the reward formulation in the RL problem, the objective is represented by the sum of the squared ℓ^2-norms to provide a globally first-order continuous cost function. This feature generally aids convergence to a feasible final solution. In a constrained optimization formulation, different initial conditions can lead to disparate final solutions. For this reason, the sequence of spacecraft states and maneuvers generated by the converged RL-based policy are leveraged as an initial guess for the optimization problem. Transferring the RL-based solution into a first guess for the optimization problem also favors a fair comparison between the RL-based and the locally optimal solutions to the constrained optimization problem. The constrained optimization is performed in this work using sequential quadratic programming in Matlab's *fmincon* procedure, employing a step tolerance of 1×10^{-14} and 100 iterations as stopping conditions [133].

7.1.4 Station-Keeping Maneuver Design with Reinforcement Learning in an Ephemeris Model

In this section, the RL maneuver planner is first trained and then applied to a spacecraft operating near an L_2 quasi-halo trajectory in the higher-fidelity point mass ephemeris model with SRP perturbation. The policy used to design impulsive station-keeping maneuvers is trained to follow the quasi-halo trajectory displayed in Fig. 7.1(b). Moreover, the policy is trained without dynamical perturbations exerted by momentum unload maneuvers, and accounting for a total of $\tau = 8$ maneuvers per episode. Each maneuver is separated by a coast arc of $\Delta t = 520$ hours. After training, the policy is first validated using a set of trajectories generated in the environment with unperturbed dynamical model. Ultimately, the policy is tested on a long-term station-keeping scenario with dynamical perturbations exerted by momentum unload maneuvers.

7.1.4.1 Policy Training

Two separate approaches for training the policy are presented. In particular, a training process in the higher-fidelity model without a priori domain knowledge is compared with a training process leveraging transfer learning. Transfer learning is used to accelerate the computationally expensive training, typical of a scenario leveraging high-fidelity dynamical models. In this section, the process of transfer learning applied to the station-keeping scenario is divided into two parts:

1) First, an actor and a critic are trained using an environment that leverages the autonomous Sun-Earth CR3BP for a maximum of 5×10^6 step, using the parameters and neural networks structure presented in Table 7.1.

2) Second, the previously converged actor and critic are used to initialize the neural networks for the training in the point mass ephemeris model. To prevent abrupt modifications of the pre-trained networks, a constant learning rate is set at $\alpha = 5 \times 10^{-4}$ throughout the 5×10^6 steps of the second training.

The RL scenario presented in Sec. 7.1.1 is used for both phases without perturbations from the momentum unloads. However, the halo orbit visualized in Fig. 7.1(a) is used as a reference trajectory for the first phase, while the quasi-halo trajectory depicted in Fig. 7.1(b) is selected as reference path in the second phase.

The performance of the policy trained by using transfer learning are compared with a new policy, trained in the point mass ephemeris model with the same parameters presented in Table 7.1, for a maximum total of 1×10^7 steps, and without leveraging transfer learning. The trend of the discounted cumulative reward over the three different training procedures is analyzed in Fig. 7.6. In particular, Fig. 7.6(a) visualizes the average value, per update, for the policy trained uniquely with the point mass ephemeris model and without leveraging transfer learning. In this case, the total training requires 72.76 hours on an Intel Core i7-2600K @ 3.40GHz using 6 logical cores. The value of the policy reaches a plateau at the end of the training process, visually confirming convergence in the value. The average value over the last 20 updates is used as a metric to provide a quantitative representation of the quality of the converged policy: averaging over the last 20 updates removes the oscillations in the value experienced at the end of the training process. The policy in Fig. 7.6(a) reaches an average value over the last 20 updates equal to 129.63. Figure 7.6(b) presents the trend of the value for the policy trained with transfer learning. In particular, the progression of the average value for the first phase, associated with training a policy in an environment leveraging the Sun-Earth CR3BP, and the trend of the average value for the training of the transferred policy in the point mass ephemeris model are colored in red and blue, respectively. The transfer, corresponding to the change in dynamical model, is marked by the intersection between the red and blue lines. The training times are approximately 37 minutes and 36.25 hours for the policies trained in the CR3BP and the point mass ephemeris model, respectively, on an Intel Core i7-2600K @ 3.40GHz using 6 logical cores. At the end of the training process, the final average value over the last 20 updates for the transferred policy is equal to 129.88, slightly larger than the equivalent obtained for the policy trained without transfer learning, depicted in Fig. 7.6(a). Moreover, the change in average value experienced by the policy in the transition between the two environment is equal to -0.0133, confirming the characteristic of the CR3BP in sufficiently-approximating the point mass ephemeris model, and the geometrical similarity between the reference trajectories depicted in Fig. 7.1(a-b). Figure 7.6 demonstrates the applicability of transferring a policy from low- to a higher-fidelity dynamical model to generate a policy with comparable performance with respect to the policy obtained without transfer learning. Moreover, the transfer learning approach demonstrates improved computational performance throughout the training process.

7.1.4.2 Policy Validation

A validating experiment is designed to compare the performance of the converged policy generated with transfer learning, and the converged policy obtained without transfer learning. The experiment is designed to evaluate both policies on a variety of trajectories, in the same scenario of the training scenario. A batch of 100 episodes per policy is generated, where each episodes terminates after a maximum of 8 steps. Each episode starts with an initial environment state vector: this state vector is employed by both policies to generate the first actions. Due to the different network parameters, the two policies produce two distinct actions associated with the first environment state. Therefore, the two converged policies generate two distinct trajectories, constituted by different sequences of state and action vectors, although starting from identical

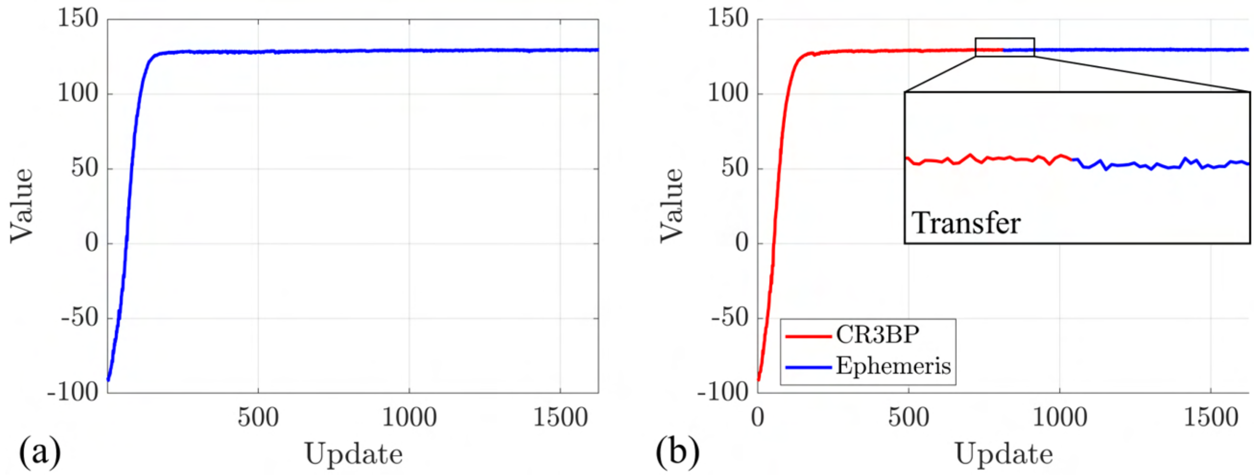

Figure 7.6: Comparison of training value between (a) a policy trained in the point mass ephemeris model and (b) a policy initially trained in the Sun-Earth CR3BP and later transferred to the point mass ephemeris model.

initial environment state vectors at the beginning of each episode.

In the experiment, five quantities are recorded for each episode: the sum of the magnitudes of the station-keeping maneuvers; the total reward, defined as the sum of the rewards after each step in an episode; the positional displacement from the reference, obtained by averaging over the states at the beginning of each step; the velocity displacement from the reference, obtained by averaging over the states at the beginning of each step; and the success of the episode. The latter takes value 0 if the episode is prematurely terminated with a reward $r = -100$, following the reward formulation outlined in Eq. (7.2), and it takes a value of 1 if the policy successfully performs the entire set of steps within the episode. These five quantities, retrieved over a set of 100 trajectories per policy, are averaged for each policy. The success of the episodes is converted in a percentage value for each policy, and presented in Table 7.2 with the remaining evaluation quantities. The reported values highlight that the policy trained with transfer learning can generate sequences of station-keeping maneuvers leading to a slightly larger average reward and a lower average maneuver magnitude when compared to the policy generated without transfer learning. Moreover, the average position and velocity deviations of the policy trained with transfer learning present slightly lower values with respect to the policy trained without transfer learning. Ultimately, both policies successfully design station-keeping maneuvers that enable bounded motion near the reference trajectory in the experiment.

Table 7.2: Characteristics of sequences of 8 maneuvers and generated trajectories constructed using policies trained with and without transfer learning (TL) and evaluated using 100 common initial conditions.

Policy trained:	Without TL	With TL
Average total Δv [m/s]	0.050	0.044
Average $\|\delta \boldsymbol{r}\|$ [km]	29.652	26.713
Average $\|\delta \boldsymbol{v}\|$ [m/s]	0.01245	0.01189
Average total reward	233.64	234.23
Success rate [%]	100	100

7.1.4.3 Policy Testing

The policy trained with transfer learning is evaluated to design a sequence of impulsive station-keeping maneuvers for a spacecraft operating near the selected L_2 quasi-halo trajectory in a point mass ephemeris model incorporating the gravitational influence of the Sun, the Earth, and the Moon, augmented with SRP perturbation and regular momentum unloads. The policy is tested with a different dynamical model with respect to the model leveraged during the training process. This evaluation allows to test the robustness of the retrieved policy with respect to periodic dynamical disturbances. In this example, one initial state for the spacecraft is defined by applying a perturbation from the reference path at an initial epoch $t_0 = 29400$ MJD, corresponding to 24 Jun 2021, 09:42:02 UTC. The first station-keeping maneuver is selected by evaluating the converged policy with the initial environment state. A momentum unload cycle, corresponding to the obtained station-keeping maneuver and three consecutive and equally-spaced momentum unload maneuvers, is applied to the spacecraft state. The environment state vector at the end of this momentum unload cycle supplies the environment state vector at the beginning of the next time step that is used to evaluate the policy again. This process repeats for approximately 10.33 years, approximately corresponding to 175 momentum unload cycles. The cardinal identifier within the environment state vector in Eq. (7.1) is assigned a constant value $\tilde{s} = 1$: this corresponds to designing each station-keeping maneuver with a far-sighted approach, balancing displacement from the reference trajectory and total maneuver magnitude over a subsequent $\tau = 8$ steps.

The resulting controlled trajectory is displayed in blue in Fig. 7.7(a) in the Sun-Earth rotating frame with dimensionless coordinates relative to the Earth. This figure reports the station-keeping maneuvers generated by the policy as scaled purple arrows. Then, Fig. 7.7(b) displays the associated maneuver magnitudes: the 175 station-keeping maneuvers that occur over these 10.33 years require a total maneuver magnitude of $\Delta v_{\text{total}} = 2.285$ m/s; the obtained total maneuver magnitude is comparable to the 2.2 m/s over approximately 10 years obtained by Bosanac et al. for a similar mission scenario, although using constrained optimization [38]. To supplement this information, Fig. 7.8 displays the time history of the magnitudes of the displacements in position and velocity between the reference and perturbed trajectory. At the beginning of the maneuver sequence, the spacecraft is relatively far from the reference trajectory and occasionally exceeds the 100 km boundaries. However, after the early transient, the policy is capable of maintaining the trajectory within 150 km and 0.06 m/s of the reference, even in the presence of perturbations exerted by the momentum unloads. Moreover, the trajectory rarely exceeds the 100 km distance threshold from the reference quasi-halo after the initial phase. Together, these results indicate that the RL implementation successfully produces a policy that generates station-keeping maneuvers for a spacecraft to remain near the L_2 quasi-halo trajectory in the point mass ephemeris model, with a low required control effort, even as momentum unloads perturb the path.

7.1.5 Comparing Optimization and Reinforcement Learning for Station-Keeping Maneuver Design

The constrained optimization scenario introduced in Sec. 7.1.3 is used to analyze the performance of the policy trained by leveraging transfer learning in the point mass ephemeris model.

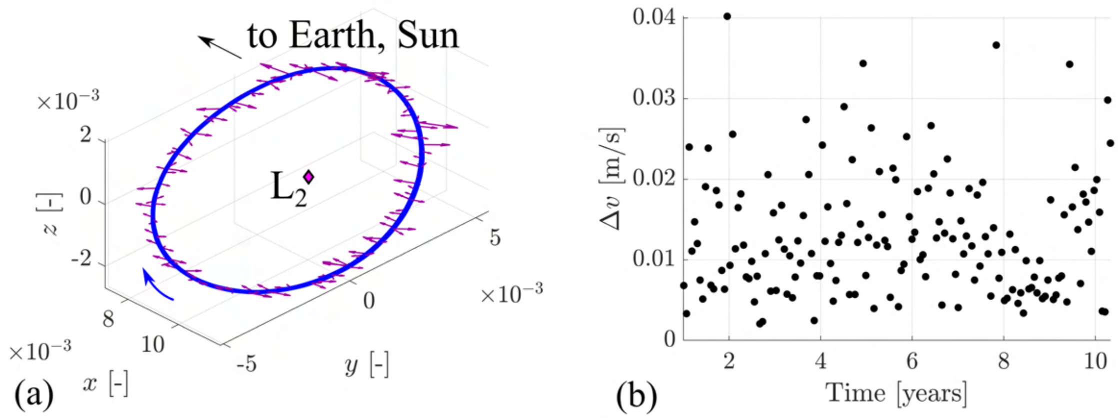

Figure 7.7: Station-keeping maneuvers in the point mass ephemeris model with regular momentum unloads: (a) single controlled trajectory propagated for approximately 10.33 years with maneuvers and (b) associated maneuver magnitudes.

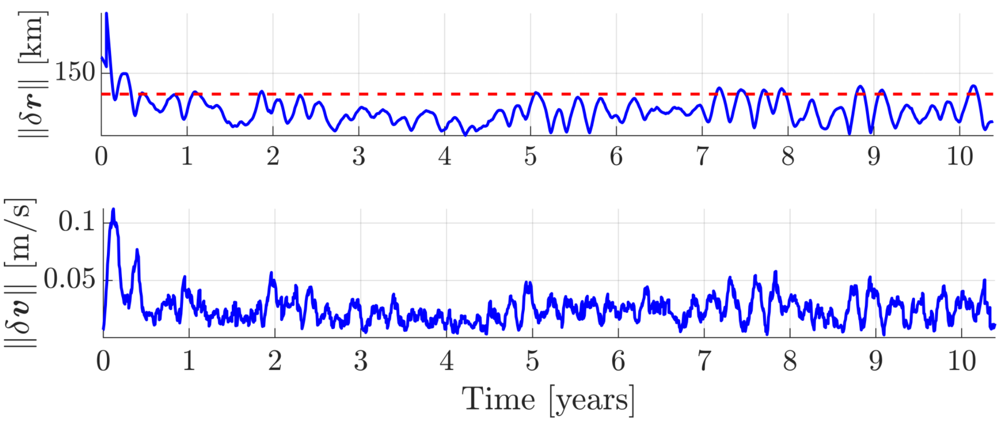

Figure 7.8: Time history of the magnitude of the position (top) and velocity (bottom) of the controlled trajectory in Fig. 7.7, relative to the reference quasi-halo trajectory.

First, the constrained optimization and the trained policy are compared on a baseline corresponding to a single trajectory. Then, statistical analysis on a batch of trajectories is pursued to infer generalized insights into the performances of the RL policy compared to the constrained optimization setup. Momentum unloads are not applied to the trajectories analyzed in this section to favor a fair comparison between the results of the RL policy and the constrained optimization scheme.

7.1.5.1 Single Trajectory Comparison

A single trajectory is generated in the point mass ephemeris model using the RL policy trained with transfer learning. The trajectory is then converted into a free variable vector and used as an initial guess for the constrained optimization scenario, as outlined in Sec. 7.1.3. Following, the performances of the optimization scenario and the trained policy are compared on this single trajectory. Fig. 7.9 provides a visual overview of the comparison. In particular, Fig. 7.9(a) depicts in blue and red the isochronous positional displacements from the reference orbit of the trajectories obtained from the RL policy and the constrained optimization scheme, respectively. The values of the displacements at the beginning of each step or at the nodes are marked with colored circles. The $\delta_1 = 100$ km threshold positional distance, associated with the internal region in the reward formulation of Eq. (7.2), is reported as a black dashed line to provide a visual reference. The figure demonstrates the maneuver sequence generated with the constrained optimization prioritizes an accelerated entrance of the spacecraft within the 100 km boundary, while the RL policy tends to delay entering the 100 km threshold. The magnitude of the maneuvers leveraged by the trajectories obtained through the RL and the optimization approach are reported in Fig. 7.9(b). Moreover, the rewards computed at the end of each step with Eq. (7.2) are visualized in Fig. 7.9(c). Both figures use the same coloring scheme introduced in Fig. 7.9(a) to distinguish between the performances of the RL policy and the constrained optimization scenario. Fig. 7.9(b-c) confirm the same behavior observed in Fig. 7.9(a), also highlighting the differences between the optimization and the RL scenario: indeed, the trajectory obtained by leveraging the RL policy presents a lower magnitude of the initial station-keeping maneuver, corresponding to a slightly larger initial step reward with respect to the solution obtained via constrained optimization. However, the solution obtained with the constrained optimization presents lower maneuver magnitudes, as well as slightly higher rewards, starting from the fourth step of the episode. The trajectory employing the RL policy slightly exceeds the 100 km boundary approximately after 75 days. This feature is also noted in the long-term station-keeping scenario depicted in Fig. 7.8 and it is associated with the absence of any hard constraints in the RL formulation for the inner region. Indeed, the trained policy might prefer driving the spacecraft beyond the 100 km threshold for a limited amount of steps, favoring lower maneuver magnitude, as a result of a relative lower scaling parameter B used in Eq. (7.2). The total station-keeping maneuver magnitudes for the trajectories obtained with the RL policy and the constrained optimization scheme are $\Delta v_{\mathrm{RL}} = 0.118$ m/s and $\Delta v_{\mathrm{fmin}} = 0.142$ m/s, reflecting the lower initial station-keeping maneuver magnitudes reported in Fig. 7.9(b) significantly weighs the comparison in favor of the RL policy for the investigated example. The sums of the rewards at the end of each step, shortly addressed as total reward, are $r_{\mathrm{tot,RL}} = 228.43$ and $r_{\mathrm{tot,fmin}} = 229.76$, highlighting slightly superior performances of the optimization scheme over the RL policy for the investigated trajectory.

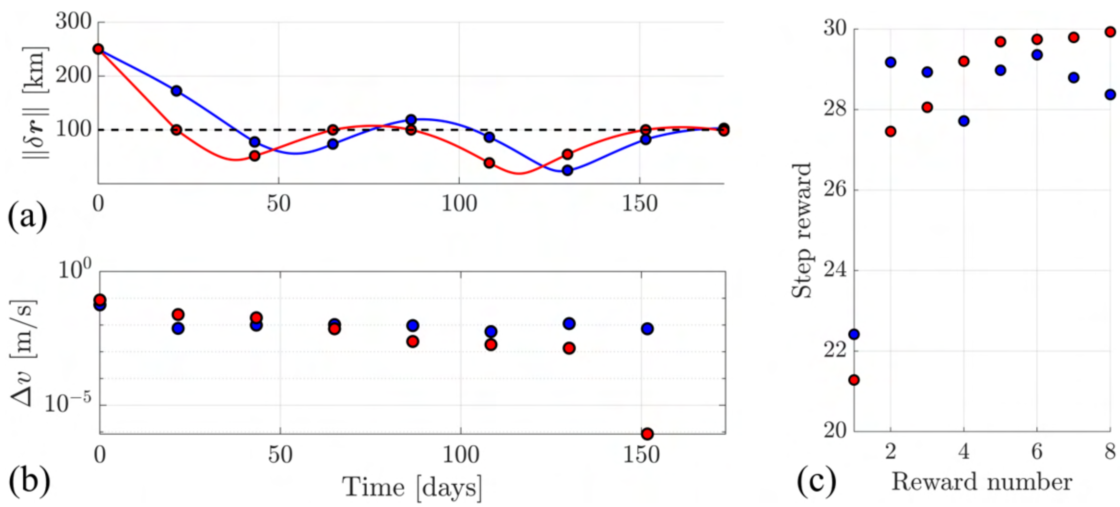

Figure 7.9: Comparison between the RL policy (blue) and the optimization (red) on a single trajectory: (a) isochronous positional distance; (b) station-keeping maneuver magnitudes; (c) reward at the end of each step.

7.1.5.2 Comparison on Multiple Trajectories

The comparison between the performance of the trained RL policy and the constrained optimization scheme is extended to a larger dataset of trajectories. A batch of 500 distinct trajectories is generated in the point mass ephemeris model, with sequences of station-keeping maneuvers obtained from the RL policy trained with transfer learning. Each trajectory is then converted into a free variable vector and used as an initial guess for the optimization scenario outlined in Sec. 7.1.3. This approach generates 500 sequences of environment state vectors and associated actions: each sequence represents a unique trajectory, optimized with Matlab's *fmincon* procedure [133]. Note that the sequence of maneuvers for the entire batch of 500 trajectories is generated in approximately 3 minutes with an Intel Core i7-2600K @ 3.40GHz using the converged actor neural network in python; however, optimizing the same set of trajectories requires approximately 1.5 days on the same machine with Matlab.

The trajectories obtained using the RL policy and the optimization scheme are compared by leveraging three metrics: the sum of the station-keeping maneuver magnitudes over each episode; the sum of the step rewards over each episode, also referred to as total reward; the average positional isochronous displacement from the reference trajectory. Three histograms are populated with these quantities, and are reported in Fig. 7.10. The histograms associated with the metrics generated from the trajecories obtained with the RL policy and the constrained optimization scheme are depicted in semi-transparent blue and red vertical bars, respectively. The average quantities are reported with dashed vertical lines, consistently colored. The top histograms in Fig. 7.10 demonstrate the trajectories obtained with the RL policy and the constrained optimization present, on average, similar total maneuver magnitude over the analyzed batch of trajectory samples. However, the RL policy generates a histogram with a longer tail on the right side of the figure, associated with larger maneuver sequences, while the trajectories obtained with the constrained optimization produce a histogram presenting a more pronounced tail on the left part of the histogram, associated with lower total maneuver magnitude. This difference reflects the slightly lower average total maneuver magnitude for the trajectories retrieved with the constrained optimization, as highlighted by the dashed vertical red line. The histograms reporting the total rewards are displayed in the center of Fig. 7.10, extending the result outlined with the single trajectory example in Fig. 7.9 to the whole analyzed batch. Indeed, the histogram associated with the total reward of the trajectories obtained with the constrained optimization scheme are slightly shifted to the right with respect to the histogram generated by the trajectories obtained with the RL policy. The latter observation suggests larger rewards, on average, provided by the constrained optimization scheme with respect to the sequences generated with the RL policy. This observation is confirmed by the third set of histograms, reporting the average displacements from the reference trajectory in the bottom of Fig. 7.10. In particular, the majority of the mass of the histogram associated with the trajectories obtained from the constrained optimization is concentrated around the mean of 69 km, while the histogram associated with the trajectories obtained with the RL policy generates a much more spread envelope. This difference is due to absence of any hard constraints in the RL formulation for the inner region, resulting in a delayed entrance of the spacecraft within the internal region of Eq. (7.2). For this reason, the trajectories obtained with the constrained optimization produce, on average, larger total reward. Trajectories placing their contributions in the histograms at the extrema of the plots are associated with paths having initial conditions relatively far from the 100 km radius of the internal region, and paths having initial conditions well within the same area.

The study presented in Fig. 7.10 is extended to infer statistical insights on the comparison between the RL policy and the constrained optimization over single trajectories. For each pair of trajectories, corresponding to one solution generated with the RL policy and the associated one solution generated with the constrained optimization scheme, three quantities are generated and presented in three histograms in Fig. 7.10: the difference in total station-keeping maneuver magnitude; the difference in the total reward; and the difference in isochronous positional displacement averaged over the entire episode. The histograms are obtained by subtracting the analyzed quantities obtained from the RL policy from the associated quantities obtained by the trajectories generated with the constrained optimization: a histogram placing the majority of its mass, or its mean reported as a dashed red line, into the positive region outlines superior performances of the constrained optimization scenario, while a histogram placing the majority of its mass to the left of the origin supports superior performances of the RL policy. The top of Fig. 7.10 displays the histogram reporting the difference of maneuver magnitude per trajectory. This histogram noticeably places the majority of its mass, and the average value, to the left of the origin. This feature indicates average inferior performances of the RL policy with respect to the *fmincon* optimization scheme,

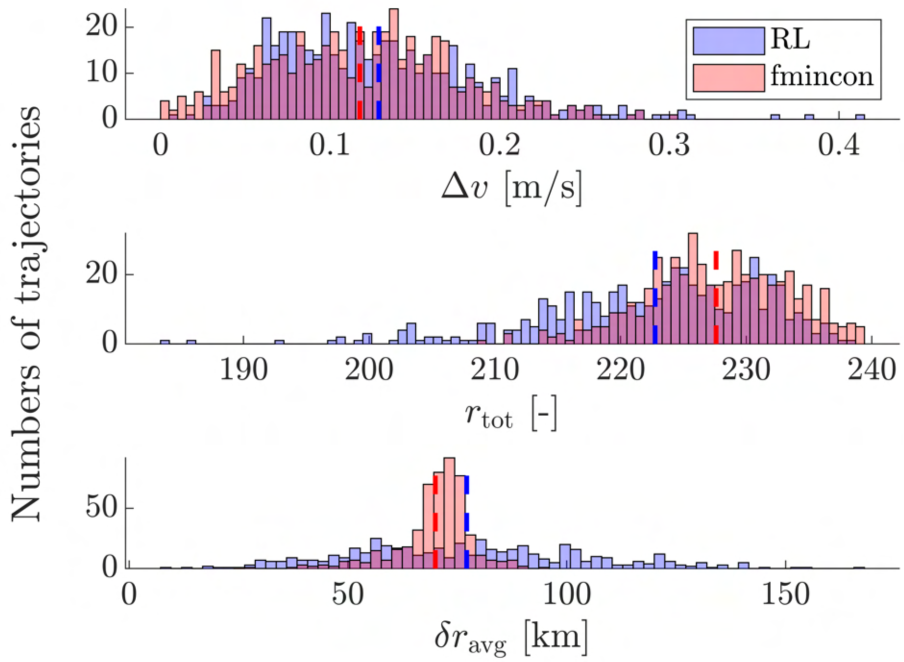

Figure 7.10: Histograms of a batch of 500 trajectories generated with the RL policy (blue) and the constrained optimization (red). Total maneuver magnitude in the top, total reward in the center, and average positional isochronous displacement in the bottom. Mean values reported via dashed vertical lines.

for the batch of analyzed 500 trajectories. The center of Fig. 7.10 displays the histogram reporting the difference of total reward per trajectory. This histogram presents the majority of its mass to the right of the origin, highlighting average superior performances of the optimization scheme with respect to the RL policy, for the batch of analyzed 500 trajectories. The last histogram, appearing in the bottom of Fig. 7.10, reports the difference in average isochronous positional displacement between the trajectories obtained with the constrained optimization approach and the RL policy. Almost every sample populating the positive region of the histogram is associated with trajectories with initial condition inside the internal region. Conversely, almost every sample populating the negative region is associated with trajectories with initial condition outside the 100 km radius. The depicted differences in performances between the RL policy with respect to the optimization scheme are again due to absence of any hard constraints in the RL formulation for the inner region.

The trained RL policy and the constrained optimization scheme generate distinct sequences of maneuver magnitudes over each episode. Indeed, the RL policy tends to present a lower initial station-keeping maneuver magnitude and produce station-keeping maneuver sequences that delay the entrance of the spacecraft within the internal region. This observation is confirmed in Fig. 7.12: the difference between the first impulses along each arc and the difference between the last station-keeping impulses along each arc are reported in the top and the bottom of the figure, respectively, using a consistent representation to Fig. 7.11. A positive mean value for the histogram reported in the top demonstrates a larger initial average value of station-keeping maneuver magnitude for the trajectories retrieved with the constrained optimization scheme. Conversely, a negative mean in the bottom figure reflects a lower average final station-keeping maneuver magnitude for the trajectories generated via constrained optimization. Since the total maneuver magnitude is primarily guided by the maneuvers early on in each episode, as depicted in Fig. 7.9(b), and the optimization scheme generates sequences of maneuvers that enforce the spacecraft within the 100 km boundary from the second step in each episode, the RL policy produces an average lower total maneuver magnitude and an average lower total reward, per trajectory, with respect to the optimization scenario.

7.2 Transfer between Prescribed Orbits in the Earth-Moon System

Proximal policy optimization is leveraged in this section to train a policy for autonomous generation of sequences of impulsive maneuvers enabling transfers between prescribed orbits in the Earth-Moon system. The results introduced in this section represent an exploratory study, assessing the applicability of reinforcement learning to generate policies for challenging transfer scenarios between prescribed periodic orbits. Additional insights can be generated with extensive exploration of different parameters governing the training process and the identified RL scenario, as well as with a thorough comparison with alternative state-of-the-art solutions for generating transfers between periodic orbits. These aspects are not addressed in the presented investigation, but represent interesting avenues for future research. For this work, the investigated scenario is initially presented in Sec. 7.2.1, and converted into a reinforcement learning framework in Sec. 7.2.2.

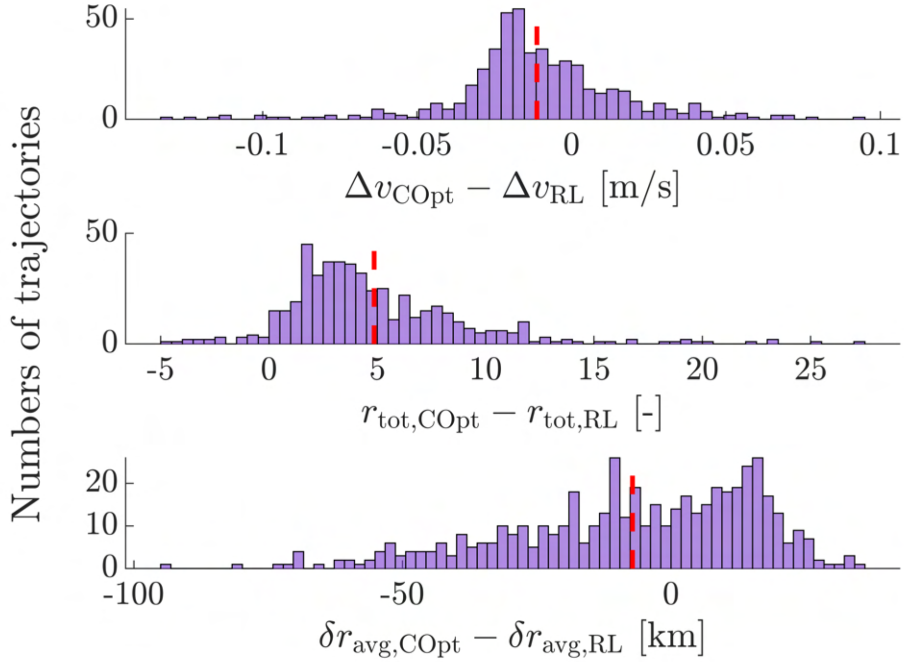

Figure 7.11: Histograms of a batch of 500 trajectories generated with the RL policy and the constrained optimization. Difference of total maneuver magnitude in the top, difference of the total reward in the center, and difference in the average positional isochronous displacement in the bottom. Mean values reported via dashed vertical red lines.

Eventually, a policy is trained and validated in a variety of orbit transfer scenarios in Sec. 7.2.3.

7.2.1 Scenario Overview

Motivated by recent interest in cislunar space, an impulsive orbit transfer scenario is presented in the Earth-Moon system [141,142]. In particular, two distinct members in the family of southern halo orbits near L_2 are utilized to define the boundary conditions of the connecting transfers. The selected members of the family are visualized in the spatial and the xz-perspective in configuration space in Fig. 7.13(a-b). The spacecraft is initially assumed to be located along the starting orbit, and the maneuver sequence is designed to autonomously drive the spacecraft towards the arrival orbit. In Fig. 7.13, the starting and arrival orbits are visualized in blue and red, respectively, the Moon is highlighted with a gray circle, while the L_2 equilibrium point is reported with a magenta

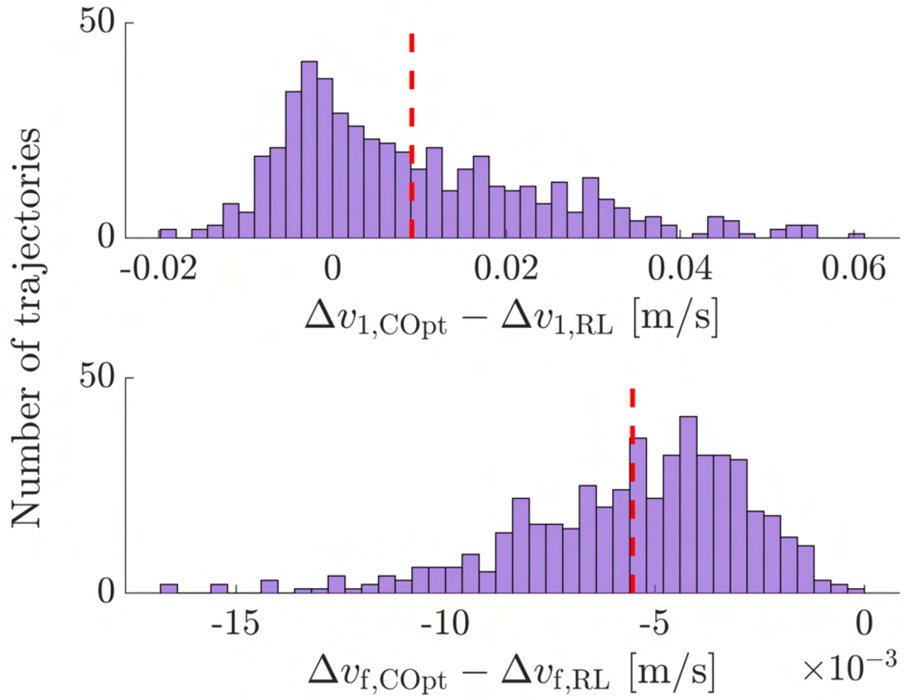

Figure 7.12: Histograms of a batch of 500 trajectories generated with the RL policy and the constrained optimization scheme. Difference of initial (top) and final (bottom) station-keeping maneuver magnitudes. Mean values reported via dashed vertical red lines.

marker. Moreover, members of the family of southern halo orbits near L_2 in the Earth-Moon system are reported in the configuration space with transparent gray markers. The orbits are reported in the dimensionless Earth-Moon rotating frame, centered at the Moon. The departing blue orbit exists at an energy level $C_J = 3.146$ and has a period $T \approx 14.79$ days, while the arrival red orbit is associated with an energy level of $C_J = 3.084$ and a period $T \approx 14.07$ days. The departing and arrival orbits depicted in Fig. 7.13 are used as a baseline for the orbit transfer scenario: the baseline serves as a demonstrating framework for the devised techniques. However, the flexibility of the designed approach is extended to generate transfers connecting different members of the family of orbits depicted in semi-transparent gray markers in Fig. 7.13 at the later stages of this section.

In the investigated scenario, a sequence of impulsive maneuvers is designed to transfer the spacecraft from the departing to the arrival orbit. Each maneuver is modeled as an impulsive three-

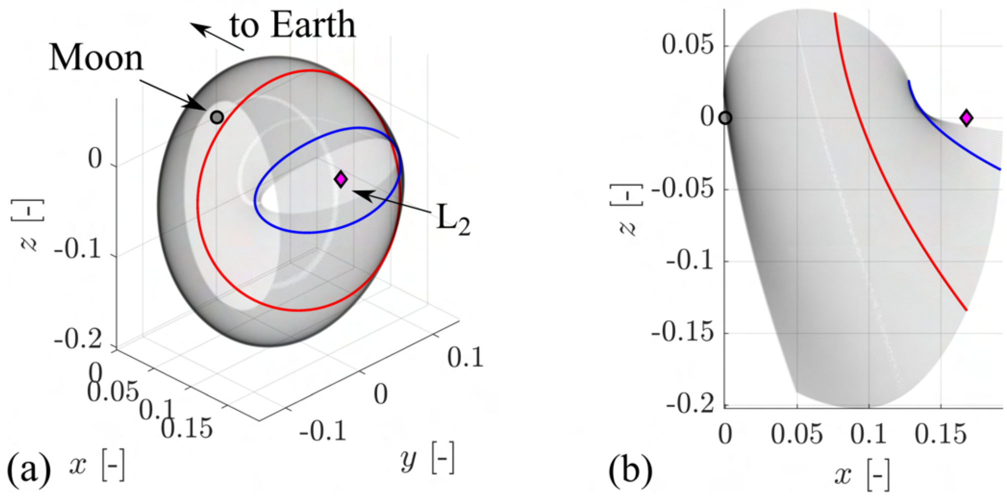

Figure 7.13: Departing (blue) and arrival (red) orbit in the Earth-Moon system for the orbit transfer scenario in the (a) spatial- and (b) xz-perspective in configuration space.

dimensional vector $\Delta\boldsymbol{v}$, instantaneously modifying the spacecraft velocity components. A total number of maneuvers τ is applied for each transfer scenario. The total number of maneuvers is fixed throughout the maneuver sequence generation, and it represents an upper boundary constraint: indeed, a relative small displacement from the arrival orbit can potentially be obtained by using a number of maneuvers $\bar{\tau} < \tau$. The spacecraft is propagated in the Earth-Moon CR3BP model between two consecutive impulsive maneuvers, for a time $\Delta t \in [\Delta t_{\min}, \Delta t_{\max}]$. Therefore, at each maneuver location, both the maneuver's $\Delta\boldsymbol{v}$ and the propagation time Δt need to be determined.

7.2.2 Translating Orbit Transfer into a Reinforcement Learning Problem

The orbit transfer problem is translated into a reinforcement learning scenario by the definition of: environment state and action vectors, a reward, a step and an episode, and an environment. The environment is designed to be agnostic of the dynamical system and the leveraged arrival and departing orbits, allowing the application to different scenarios connecting distinct pairs of orbits. The environment is constructed by the definition of a step and an episode, and it is conceptually represented in Fig. 7.14. Each episode is composed of a maximum of τ steps, with the first step corresponding to the beginning of the episode. At the beginning of each episode, the state of the environment is initialized to represent the spacecraft state. In particular, a spacecraft state $\boldsymbol{x}_0 \in \mathbb{R}^6$ is selected randomly along the departing orbit. Then, a displacement $\delta\boldsymbol{x}_0 \in \mathbb{R}^6$ from $\boldsymbol{x}_0$ is computed. Together, the state and displacement information are leveraged by the environment to populate the environment state, used to generate an action $\boldsymbol{a}_0 \in \mathbb{R}^4$. The action generated by the policy contains information regarding both the impulsive maneuver executed at the current spacecraft state $\boldsymbol{x}_0$, and the propagation time separating the spacecraft state $\boldsymbol{x}_0$ from the beginning of the next step. After propagating the spacecraft state with the unperturbed dynamical model of the Earth-Moon CR3BP, the spacecraft state at the end of the propagation arc is leveraged to compute the reward for the step. Then, a new step for the episode can start, leveraging the information of the spacecraft state at the end of the previous step to initialize the environment state. The process repeats until termination criteria are met at the end of a step.

Mathematical formulations of the environment state and the action are designed to be representative of the spacecraft state and the final goal of the policy. The mathematical formulation of the state in the proposed environment is expressed as:

$$\boldsymbol{s}_i = [\tilde{\boldsymbol{x}}_i, \delta\tilde{\boldsymbol{x}}_i, \tilde{s}] \in \mathbb{R}^{13} \tag{7.9}$$

where the tilde operator $\tilde{(\cdot)}$ represents scaled quantities, used to maintain the elements of the environment state within the range $[-1, 1]$ throughout the training process, therefore favoring convergence of the trained networks. The state in Eq. (7.9) comprises the scaled spacecraft state $\tilde{\boldsymbol{x}}_i$ in the rotating frame, the scaled displacement $\delta\tilde{\boldsymbol{x}}_i$ computed as the six-dimensional state difference from the closest state of the arrival orbit, and the maneuver number along the current episode $\tilde{s} = 2i/\tau - 1$, with $i \in \{\tau, \tau - 1, \ldots, 0\}$. The maneuver number is initialized at $\tilde{s} = 1$ at the initial step and it is leveraged to improve convergence properties of the trained networks. The environment state is used at the beginning of each step as an input to the actor neural network to generate an action, mathematically formulated as:

$$\boldsymbol{a}_i = [\Delta\tilde{\boldsymbol{v}}_i, \Delta\tilde{t}_i] \in \mathbb{R}^4 \tag{7.10}$$

where the tilde operator is leveraged to represent scaled quantities. The action information is used to obtain the impulsive maneuver and the propagation time. In particular, the first three components of the action vector are used to generate the impulsive maneuver, computed in the nondimensional rotating frame as $\Delta \boldsymbol{v}_i = \nu \Delta \tilde{\boldsymbol{v}}_i$. The scaling factor ν represents a system-dependent quantity, enabling the first three components of the action vector to generally remain bounded in $[-1, 1]$ throughout each episode. For the Earth-Moon system investigation presented in this work the scaling factor is selected as $\nu = 3 \times 10^{-3}$. The last component of the action vector in Eq. (7.10) is associated with the propagation time between the start of two consecutive steps, used to generate a propagation time in the interval $[\Delta t_{\min}, \Delta t_{\max}]$ for the analyzed step as:

$$\Delta t_i = \left(\tanh\left(\Delta \tilde{t}_i\right) + 1\right) \frac{\Delta t_{\max} - \Delta t_{\min}}{2} + \Delta t_{\min} \tag{7.11}$$

where the hyperbolic tangent $\tanh(\cdot)$ is used to enforce the time remains in the interval $[-1, 1]$. With the action formulation presented in Eq. (7.10), the policy governs both the magnitude of the selected impulse and the propagation time.

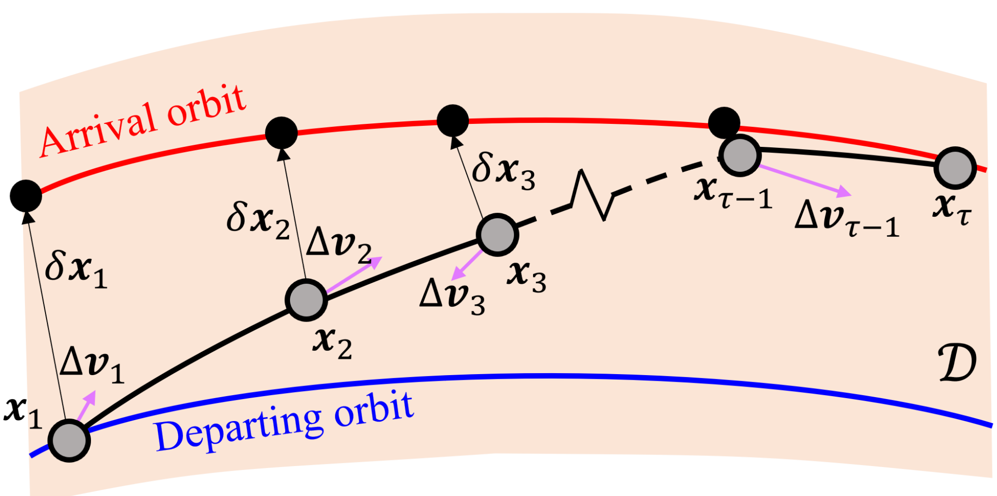

Figure 7.14: Conceptual representation of the orbit transfer scenario between prescribed orbits.

At the end of each step, a reward is computed to reflect the benefit of the state-action pair towards the ultimate goal of the orbit transfer scenario. In particular, the reward is formulated as a piecewise function, separating the design space in two regions:

1) Internal region: the spacecraft is assumed in the internal region if the position vector is

contained in a boundary set $\mathscr{D} \subset \mathbb{R}^3$. The boundary set $\mathscr{D}$ is constructed as a box in the configuration space that contains both the departing and the arrival orbit. For this reason, the limiting values of $\mathscr{D}$ along each component in the configuration space are retrieved by using the departing and arrival orbits. In particular, the upper and lower bounds along each dimension are set proportional to the maximum and minimum values of the associated dimensions of the departing and arrival orbit. For example, the lower boundary of $\mathscr{D}$ along the x-direction, x_l, is computed as:

$$x_l = \delta_{1,x} \min[\min(\boldsymbol{o}_{\text{dep},x}), \min(\boldsymbol{o}_{\text{arr},x})] \tag{7.12}$$

where $\delta_{1,x} > 0$ is a scaling factor, while $\boldsymbol{o}_{\text{dep},x}$ and $\boldsymbol{o}_{\text{arr},x}$ represent the x-components of the states along the departing and arrival orbits, respectively. Similarly, the upper boundary of $\mathscr{D}$ along the x-dimension, x_u, is set equal to:

$$x_u = \delta_{2,x} \max[\max(\boldsymbol{o}_{\text{dep},x}), \max(\boldsymbol{o}_{\text{arr},x})] \tag{7.13}$$

where $\delta_{2,x} > 0$. The upper boundary x_u corresponds to the scaled maximum between the maxima of the arrival and departing orbits along the x-dimension. The two scaling factors $\delta_{1,x}$ and $\delta_{2,x}$ are leveraged to extend the internal region, empirically leading to improved convergence properties. The upper and lower bounds of $\mathscr{D}$ along the remaining two positional components are similarly retrieved using the minima and maxima along the y- and z-coordinates of the arrival and departing orbits, respectively. The reward in the internal region is formulated as a weighted sum of two terms reflecting: displacement from the arrival orbit; maneuver magnitude in the associated step. The internal region is presented in Fig. 7.14 in orange.

2) External region: the spacecraft is considered in the external region if the position vector of the spacecraft at the end of the step is not contained in $\mathscr{D}$. The external region is represented by the area beyond the orange region in Fig. 7.14. When the spacecraft is in the external region, a large negative reward is assigned to the analyzed state-action

pair. The value of the assigned rewards reflects a heavy penalization, due to the significant departure from the internal region. After assigning the penalizing reward, the associated episode is terminated.

Based on the distinction between two fundamental regions of motion, the mathematical formulation of the piecewise reward is:

$$r_i = \begin{cases} \left(\log\left(\|\delta\boldsymbol{x}_i\|\right)\right)^2 + K\|\Delta\tilde{\boldsymbol{v}}_i\| & \text{if } \boldsymbol{r}_i \in \mathscr{D} \\ -100 & \text{otherwise} \end{cases} \tag{7.14}$$

In this formulation, $\delta\boldsymbol{x}_i$ is computed as the displacement of the current spacecraft state from the closest state of the arrival orbit. A formulation with the negative logarithm is selected to favor a continuous and positively unbounded representation for displacements $\delta\boldsymbol{x}_i \rightarrow 0$. The square of the natural logarithm encourages the generation of policies with improved performances for the approaching phase to the arrival orbit. A linear formulation is selected for the action contribution within the reward to provide an upper null limit for the action contribution within the reward. Note that the leveraged scaling factors governing the dimension of the internal region $\mathscr{D}$ might limit the explored transfer geometries. A conceptual overview of the training process for the orbit transfer scenario is presented in Fig. 7.15.

For this investigation, the majority of the hyperparameters governing the PPO cost function, the training process, and the neural network structures are inherited from the Bayesian optimization approach detailed in Table 7.1 for the station-keeping scenario. In particular, only a subset of parameters are modified, corresponding to: final learning rate $\alpha_f = 5 \times 10^{-5}$; learning rate schedule with 6 steps; critic neural network width of 128 nodes. These modifications are empirically demonstrated to generate policies with optimal performances. Although the orbit transfer environment differs from the station-keeping scenario, the majority of the hyperparameters is retained due to the similarity of the leveraged chaotic dynamical model, and the general recommendations of a variety of authors applying PPO to complex and chaotic dynamical systems [61, 117, 124, 139, 140].

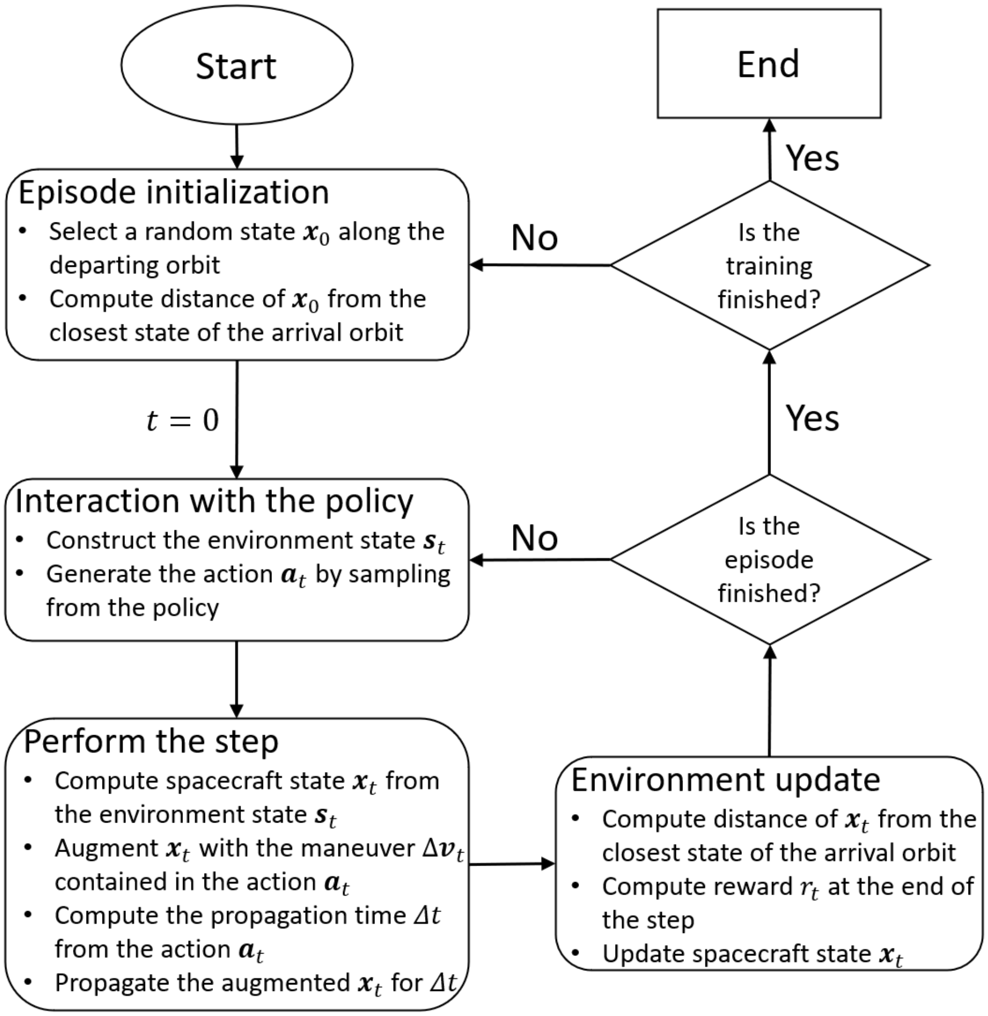

Figure 7.15: Flowchart of the training process for the orbit transfer RL scenario.

7.2.3 Generating Transfers between Prescribed Orbits

In this section, an RL maneuver planner is trained and applied to an orbit transfer scenario between periodic orbits in the Earth-Moon CR3BP. The policy is trained for the baseline scenario, constructing a transfer between the orbits depicted in Fig. 7.13. Then, the retrieved policy is validated on a batch of trajectories transferring the spacecraft from the departing to the arrival orbits. Ultimately, different policies are trained and demonstrated for orbit transfer scenarios connecting periodic orbits that differ from the baseline framework visualized in Fig. 7.13.

7.2.3.1 Training in the Baseline Orbit Transfer Scenario

A policy is trained to autonomously generate sequences of maneuvers enabling transfers that connect a departing and arrival halo orbit near L_2. The connected orbits used for this demonstrative analysis are depicted in blue and red, respectively, in Fig. 7.13. The orbit transfer scenario is designed to account for a maximum total number of maneuvers $\tau = 50$ per episode, a propagation time $\Delta t \in [\Delta t_{\min}, \Delta t_{\max}]$ with $\Delta t_{\min} = 5$ hours and $\Delta t_{\max} = 60$ hours, scaling factors for the internal region $\mathscr{D}$ equal to $\boldsymbol{\delta}_1 = [\delta_{1,x}, \delta_{1,y}, \delta_{1,z}] = [0.99, 1.2, 1.2]$ and $\boldsymbol{\delta}_2 = [\delta_{2,x}, \delta_{2,y}, \delta_{2,z}] = [1.01, 1.2, 1.2]$, and a relative weight for the maneuver contribution within the reward $K = 15$. A conservative and large number of maneuvers $\tau = 50$ is selected to accommodate transfers requiring a large number of impulsive maneuvers. The total number of maneuvers is also associated with the selected boundaries of the propagation time separating two consecutive maneuvers $[\Delta t_{\min}, \Delta t_{\max}]$. Indeed, the upper limit of the propagation time for an entire episode, equal to 50 steps of 60 hours each, is $\Delta t_{\text{episode, max}} = 125$ days, corresponding to more than 8 periods of the departing and arrival orbits. A lower boundary $\Delta t_{\min}$ is selected to eliminate consecutive steps with short coasting time. The scaling factors $\boldsymbol{\delta}_1$ and $\boldsymbol{\delta}_2$ are scenario-dependent quantities impacting the convergence properties of the training process, and the performance of the obtained policy: a larger internal region $\mathscr{D}$ reflects larger policy exploration, although empirically associated with poor performances of the generated policy for the investigated scenario due to the close proximity of the investigated orbits to the Moon. Moreover, the selected components of $\boldsymbol{\delta}_1$ and $\boldsymbol{\delta}_2$ depend on the investigated orbits: both the arrival and departing orbits have positive minima and maxima along the x-axis, that are larger in magnitude with respect to the minima and maxima along the y- and z-dimension, respectively represented by a negative and positive value. Ultimately, recall the presented results serve as a demonstration of the application of reinforcement learning in the challenging and chaotic orbit transfer scenario: an optimal selection of parameters governing the investigated scenario might likely result in a policy with improved performances.

The identified set of scenario-dependent parameters defines the environment for the orbit transfer scenario, enabling the training of policies for the autonomous generation of maneuver sequences. In particular, two separate approaches for training the policy are here presented. In a first approach, a policy is trained leveraging the described hyperparameters of the selected scenario, for a maximum of 1.5×10^7 steps. The value during training is visualized in Fig. 7.16 (a) in blue. This figure visualizes a relatively contained improvement of the policy over the learning process, reaching convergence towards the end. In a second approach, a different policy is trained by leveraging transfer learning to encourage improved performance. Initially, the policy is trained for a maximum of 1×10^7 steps using an identical set of parameters with respect to the policy retrieved from the first approach. However, for this second training, the relative weight for the maneuver contribution within the reward is set as $K = 0$. When the training is terminated, the converged neural networks are used to initialize the actor and the critic for a subsequent learning. This second training differs from the precedent on two aspects: 1) the learning parameter is initially set to $\alpha_0 = 5 \times 10^{-4}$, allowing a reduced early exploration and preventing from network disruption; 2) a weighting factor $K = 15$ is used for the maneuver contribution within the reward. The value of the first and the second training process for the second approach are depicted in Fig. 7.16(b) in blue and red, respectively. The transfer in the reward formulation is zoomed-in in the figure, and highlights a sudden loss in performance of the policy. The degraded average value is associated to the negative impact of the scaled maneuver magnitude in the reward formulation. The approaches presented in Fig. 7.16 generate distinct policies, with different final values: for the first approach, the policy obtain a value $r_{\mathrm{avg},20} = 78.862$, averaged over the last 20 updates; the second approach constructs a policy with a value $r_{\mathrm{avg},20} = 1270.457$ over the last 20 updates. The difference in value is also evident from the comparison in Fig. 7.16 (a-b), where the approach employing transfer learning reaches larger average values at the end of the training process, although using the same cumulative amount of training experiences. The different values at convergence depicted in Fig. 7.16 (a-b) are extensively analyzed in the next section, focusing on the performances of the trained policies.

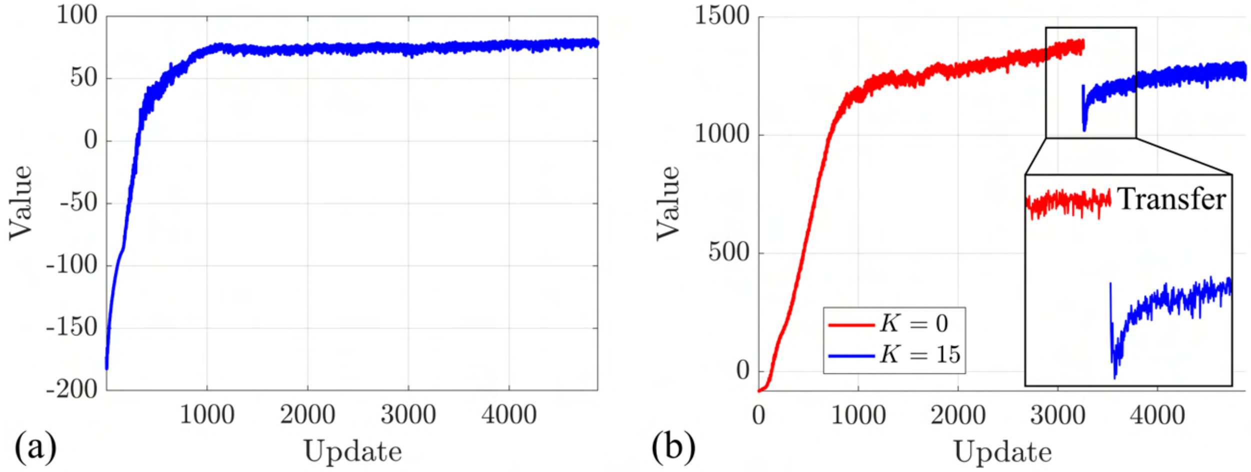

Figure 7.16: Value of training in the baseline orbit transfer scenario in the Earth-Moon system depicted in Fig. 7.13 (a) without and (b) with transfer learning in the reward formulation.

7.2.3.2 Validating the Baseline Scenario

After the training, the converged policies are validated on a batch of 1000 episodes. For each episode, one environment state vector is constructed, corresponding to a specific, yet random, initial location along the departing orbit. The constructed set of 1000 environment states is used as a set of initializing environment states for the two converged policies. Due to the different approach and different training processes, the converged policies likely generate a pair of distinct trajectories for each identical initial environment state. Five metrics are generated from each trajectory: final position displacement from the arrival orbit $\|\delta \boldsymbol{r}_{\text{end}}\|$; final velocity displacement from the arrival orbit $\|\delta \boldsymbol{v}_{\text{end}}\|$; total maneuver magnitude over an episode Δv_{tot}; cumulative reward over an episodes r_{tot}; and number of performed steps per episode $\bar{\tau}$. Then, the policies are validated and compared by averaging the five metrics on the validation set of 1000 trajectories. The results are reported in Table 7.3.

The approach using transfer learning generates a policy with superior performances over the policy generated without transfer learning, for the examined set of validating trajectories. Specifically, the policy generated with transfer learning can construct transfers with relatively

Policy trained	$\|\delta \boldsymbol{r}_{\text{end}}\|$ [km]	$\|\delta \boldsymbol{v}_{\text{end}}\|$ [m/s]	Δv_{tot} [m/s]	r_{tot} [-]	$\bar{\tau}$
Without TL	25026.35	151.94	0.46	200.73	50
With TL	16.37	0.08	237.80	1731.74	50

Table 7.3: Average final position displacement, final velocity displacement, total maneuver magnitude and total reward over an episode and number of maneuvers per episode for policies trained with and without transfer learning (TL) and evaluated using a set of 1000 initial conditions.

small final position and velocity displacements, relatively larger total maneuver magnitude and total reward. The final position and velocity deviations of the sampled trajectories generated with the policy not using transfer learning are relatively large, suggesting with the relatively low total maneuver magnitude and total reward, that the policy does not actually succeed to transfer the spacecraft towards the arrival orbit. Extended training with a distinct reward formulation and a different scheduling for the learning rate might generate a policy with improved performance that does not require transfer learning. Despite the distinct performance, both investigated policies generate sequences of maneuvers that maintain bounded motion within the internal region $\mathscr{D}$ for each of the 1000 trajectories per policy.

The distinct performance between the trained policies are analyzed with a trajectory example. A single trajectory generated by the two policies is reported in Fig. 7.17(a) in the configuration space. In both representations, the arrival and departing orbits are reported in red and blue, respectively, and indicated with labels. The transfer generated by the policy trained with transfer learning is depicted with a straight black line, while the associated transfer of the policy trained without transfer learning is reported with a black dashed line. Both trajectories are initialized from the same spacecraft state along the departing orbit, and depicted with a blue circle in the figures. However, the distinct trajectories reach two distinct final spacecraft states, marked with two red dots at the end of both arcs. The sampled trajectories visually confirm the superior performance of the policy trained with transfer learning, and the inability of the policy trained without transfer learning to transfer the spacecraft to the arrival orbit for this example. The trajectory obtained with the policy trained with transfer learning is supplemented in Fig. 7.17(a) by a series of magenta arrows along the path, representing the designed impulsive maneuvers. The magnitudes of the impulsive maneuvers, and the associated rewards computed at the end of each step, for the trajectory generated with the policy trained with transfer learning are reported in Fig. 7.17(b-c). These figures highlight an initial transient towards the arrival orbit, reflected by negative rewards and large magnitudes of the designed orbit transfer maneuvers. Specifically, the minimum in the sequence of rewards is recorded approximately at the 10th day of the trajectory, corresponding to the largest maneuver magnitude along the sequence. After the 13th day, the spacecraft has approached the arrival orbit, therefore the policy generates a sequence of impulsive maneuvers with a relatively small magnitude, reflected in large rewards. Overall, the length of the coast arcs separating the impulsive maneuvers varies along the transfer and the total maneuver magnitude for the investigated transfer equals $\Delta v_{\text{tot}} = 239.96$ m/s. Alternative approaches applied to transfer design between different southern halo orbits near L_2 in the Earth-Moon system demonstrate similar total maneuver magnitudes [86, 141].

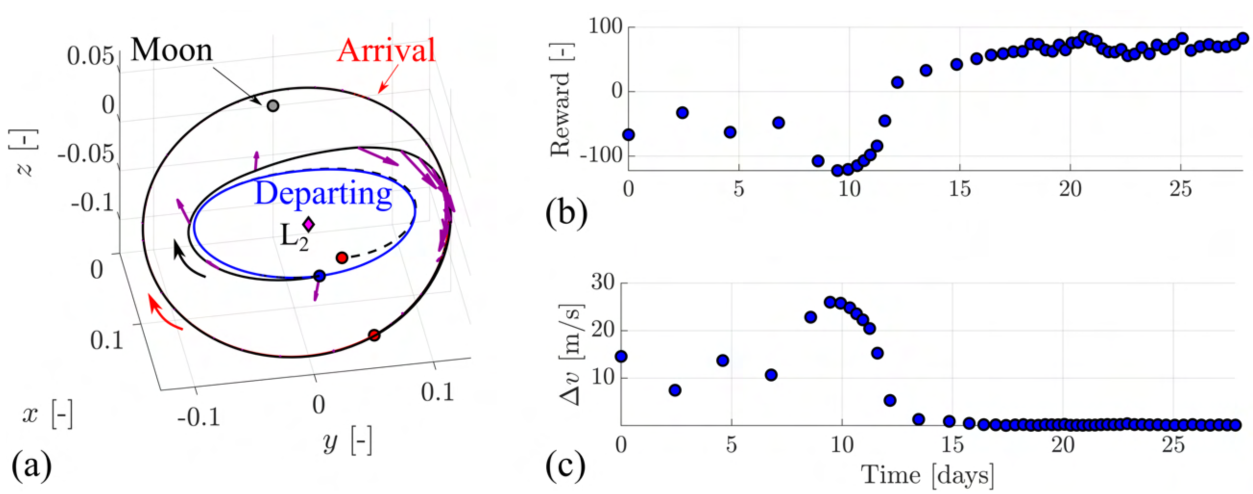

Figure 7.17: Comparison between sampled trajectories in the Earth-Moon system, generated via policies trained (solid black) with and (dashed black) without transfer learning and visualized in configuration space. The initial and final states are reported with blue and red circles, respectively.

The batch of 1000 episodes used to validate the trained policies is employed to uncover potential suboptimal performances of the trained policy. Due to the evident superior performance in average value at convergence, only the policy trained with transfer learning is investigated for the remainder of this section. In particular, this policy is analyzed using four of the metrics analyzed in Table 7.3 and converted in histograms. The histograms allow to detect edge-cases where the trained networks can present suboptimal performance. First, the final position displacement from the arrival orbit $\|\delta \boldsymbol{r}_{\text{end}}\|$ and the final velocity displacement from the arrival orbit $\|\delta \boldsymbol{v}_{\text{end}}\|$ are used to populate two histograms, presented in the top and bottom, respectively, of Fig. 7.18. The dimensional displacements are converted into natural logarithmic scales to visualize compact representations. The trajectories present a uniform distribution about the mean values, reported in Table 7.3 for the policy trained with transfer learning. Also, a few solutions present relatively small and large final displacements from the arrival orbit, with extrema represented by $\max(\|\delta \boldsymbol{r}_{\text{end}}\|) = 313.96$ km, $\min(\|\delta \boldsymbol{r}_{\text{end}}\|) = 0.48$ km, $\max(\|\delta \boldsymbol{v}_{\text{end}}\|) = 0.91$ m/s, $\min(\|\delta \boldsymbol{v}_{\text{end}}\|) = 2.74 \times 10^{-3}$ m/s. Moreover, only three trajectories exceed a final displacement larger or equal than 130 km from the arrival orbit, confirming the efficiency and the robustness of the devised policy in generating maneuvers to approach the arrival orbit.

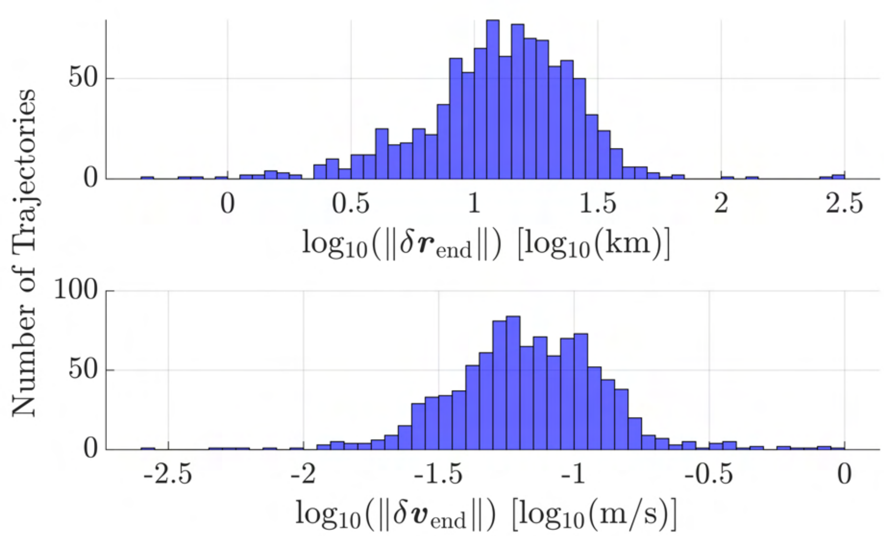

Figure 7.18: Histograms of (a) final position displacement and (b) final velocity displacement, over a batch of 1000 trajectories obtained with a policy trained with transfer learning.

Ultimately, two more distributions of the entire batch of 1000 trajectories are constructed to generate further insights on eventual suboptimal performances of the retrieved policy. In particular, two histograms are populated with the total maneuver magnitude Δv_{tot} and the cumulative reward over an episode r_{tot}, and presented in Fig. 7.19. Both histograms present large mass in a neighborhood of their mean values, reported in Table 7.3 for the policy trained with transfer learning. A small amount of samples present relatively small and large total maneuver magnitude and rewards, with extrema represented by $\max(\Delta v_{\mathrm{tot}}) = 342.57$ m/s, $\min(\Delta v_{\mathrm{tot}}) = 216.50$ m/s, $\max(r_{\mathrm{tot}}) = 2258.79$, $\min(r_{\mathrm{tot}}) = -4284.86$. Only five trajectories present a total maneuver magnitude of more than 280 m/s, or a reward lower than 50, and the batch contains three trajectories with a reward lower than -400. These represent the episodes where the policy present the lowest performance, and are associated with the samples appearing in the right part of the Δv_{tot} histogram and the left of the r_{tot} histogram, respectively in the top and bottom part of Fig. 7.19.

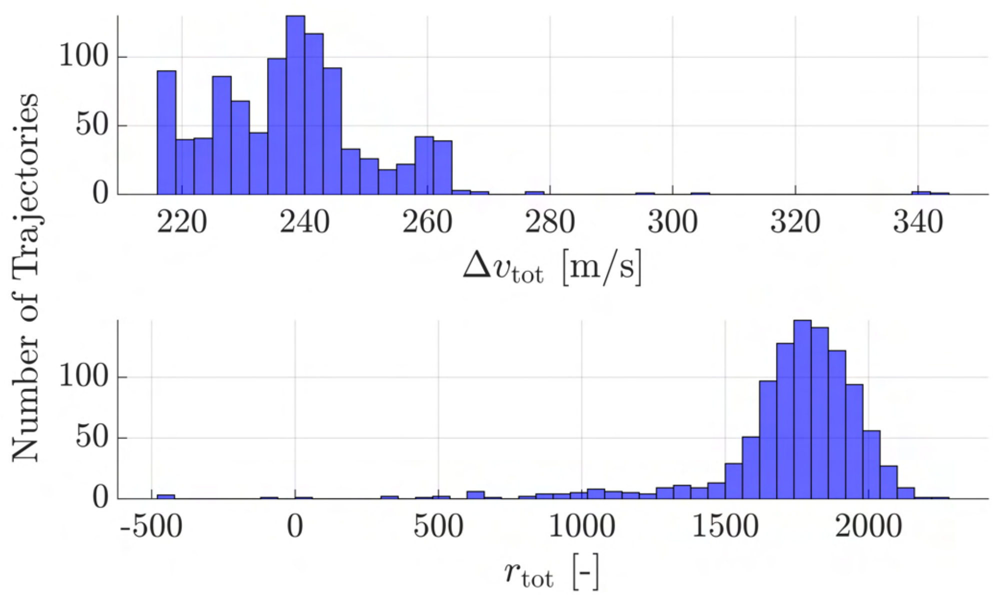

Figure 7.19: Histograms of (a) total maneuver magnitude and (b) total reward over an episode, over a batch of 1000 trajectories obtained with a policy trained with transfer learning.

One sample from the highlighted subset of low-performing arcs is further analyzed, by projecting the associated trajectory onto the configuration space in Fig. 7.20(a). The arrival and departing orbits are depicted in blue and red, respectively, while the spacecraft trajectory is presented in black. The starting and arrival spacecraft positions are marked using blue and red circles, respectively, while the sequence of maneuvers is reported with magenta arrows. The visual representation of the arc highlights how the trained policy successfully generates a maneuver sequence to transfer the spacecraft towards the designed arrival orbit. Fig. 7.20(b-c) report the sequences of rewards and maneuver magnitudes along the analyzed trajectory as a function of the number of days since the beginning of the transfer. The analyzed transfer generates a sequence of 25 maneuvers for the first 23 days of the transfer, corresponding to a sequence of negative rewards. For the last 8 days of the transfer, the policy generates a sequence of maneuvers with small magnitudes, corresponding to positive rewards. Moreover, the maneuvers appear more separated in time during the first 23 days, besides an initial phase of three maneuvers separated by short propagation arcs. Conversely, the maneuvers present relatively short propagation arcs during the last 8 days of the transfer, associated with the portion of the trajectory where the spacecraft displacement from the arrival orbit is relatively small. The sequence of large maneuvers and rewards during the first 23 days of the transfer negatively impacts the total reward of $r_{\text{tot}} = -428.49$ and the total maneuver magnitude of $\Delta v_{\text{tot}} = 342.57$ m/s presented by the sampled transfer, explaining the suboptimal performance of the policy for the analyzed trajectory. However, alternative approaches applied to transfer design between different southern halo orbits in the Earth-Moon system demonstrate similar total maneuver magnitudes [86, 141]. Additional analysis on the decision mechanism behind the maneuver placement and direction along the trajectory, and a comparison of the generated performance of the policy with distinct approaches for orbit transfer maneuver generation is the subject of future research.

7.2.3.3 Application to Different Orbit Transfer Scenarios

The orbit transfer scenario is designed to be agnostic of the leveraged arrival or departing orbits. For this reason, the methodology detailed in this section can be applied with minor or null modifications to autonomously generate transfers connecting different combinations of orbits. Three different and progressively more challenging transfer scenarios are here introduced. Recall

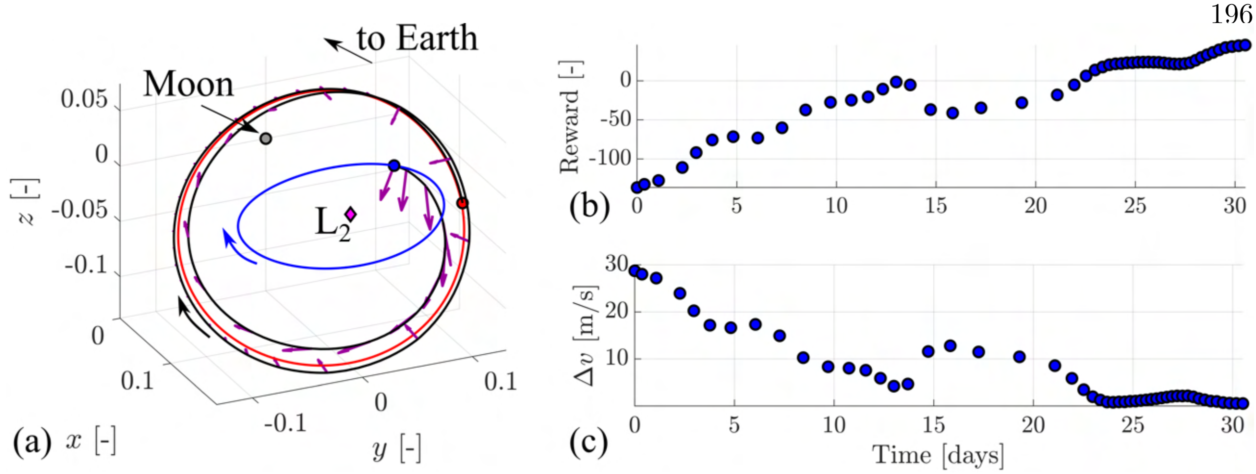

Figure 7.20: Example of suboptimal trajectory generated in the Earth-Moon system by the policy trained with transfer learning: (a) trajectory arc and associated (b) rewards and (c) maneuver magnitudes at the end of each step. The initial and final states are reported with blue and red circles, respectively.

the focus of this section is to demonstrate the capability of the reinforcement learning technique, combined with the designed scenario, to train policies for the autonomous design of sequences of impulsive orbit transfer maneuvers: additional efforts to improve the performance of the generated policies are the subject of future research.

Transfer between halo orbits near L_2

A first policy is trained to connect a pair of southern halo orbits near L_2 in the Earth-Moon system, that differs from the baseline scenario presented in Fig. 7.13. In particular, a departing orbit with $C_J = 3.129$ and period $T = 14.623$ days, and an arrival orbit with $C_J = 3.016$ and period $T = 10.793$ days are selected for this first example. The training process and a sample trajectory are depicted in Fig. 7.21 (a-b). In particular, the training process leverages an identical set of hyperparameters used for the baseline scenario, employing the same number of updates for the neural network. A sample trajectory, generated with the trained policy at the end of the training process, is projected onto the configuration space in Fig. 7.21(b). The arrival and departing orbits are depicted in red and blue, respectively. The spacecraft path is presented in black, and the starting and arrival spacecraft states are reported with blue and red circles, respectively. Moreover, the direction of the maneuvers is represented by magenta arrows. In particular, the maneuvers present larger values and general alignment during the approaching phase to the arrival halo orbit. During the last part of the transfer, the obtained maneuvers present small magnitudes, reflecting the bounded motion with respect to the arrival orbit. The total maneuver magnitude and total reward are $\Delta v_{\text{tot}} = 408.17$ m/s and $r_{\text{tot}} = 455.52$, respectively: a policy with improved performance can be generated with scenario-specific refinement, as well as with prolonged training. Alternative approaches leveraging constrained optimization for transfer designs between southern halo orbits in the Earth-Moon system demonstrate similar total maneuver magnitudes [86]. However, Fig. 7.21(a-b) demonstrates the applicability of the trained policy to connect a different pair of orbits in the same family of the baseline scenario investigated in Sec. 7.2.3.2.

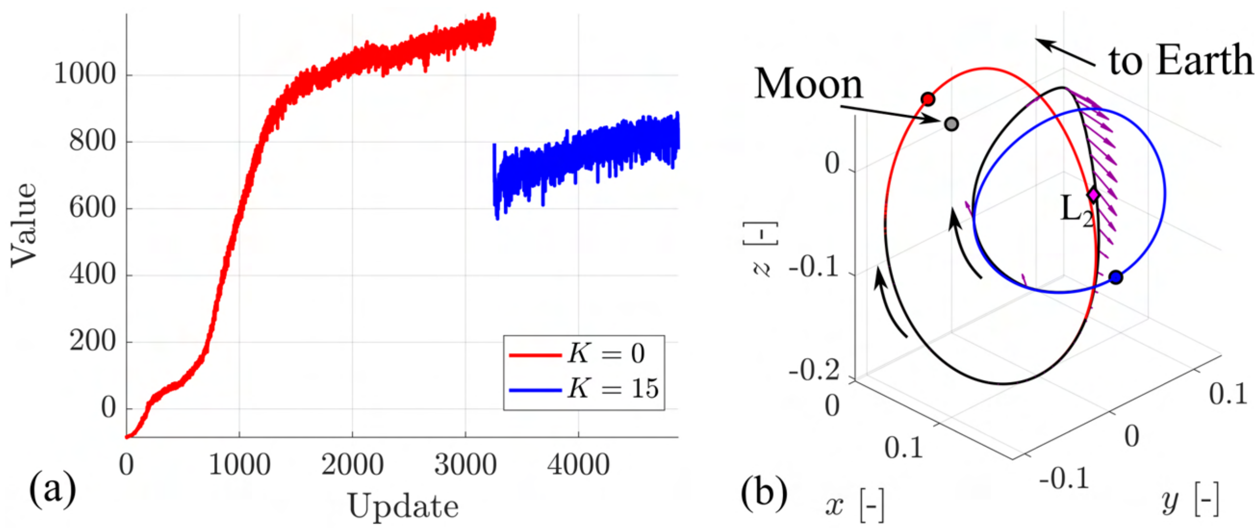

Figure 7.21: Example scenario for a transfer between two southern halo orbits near L_2 in the Earth-Moon CR3BP: (a) value per update and (b) sample trajectory generated by the policy.

Transfer between halo orbits near L_2 and L_1

A second policy is trained to generate maneuver sequences enabling a spacecraft to transfer from a southern halo orbit near L_2 to a southern halo orbit near L_1 in the Earth-Moon system. In particular, a departing orbit with $C_J = 3.102$ and period $T = 14.320$ days, and an arrival orbit with $C_J = 3.001$ and period $T = 10.539$ days are selected for this second example. The average value throughout the training process is depicted in Fig. 7.22. The training environment presents two major modifications with respect to the baseline scenario, comprising a constant learning rate after the transfer of the policy, marked by a change in color in Fig. 7.22, and adjunct termination criteria, corresponding to a reward $r = -100$ for those trajectories presenting a lunar periapsis with height lower than 400 km. The adjunct termination criteria empirically causes lower average value, and a relative larger fluctuation after the transfer of the policy, as demonstrated by the trend of the value colored in blue in Fig. 7.22. Additional training and the incorporation of constraints within the objective function proposed for PPO might enable training policies that do not exhibit a slight deterioration of the performance after the transfer. After training, the policy is demonstrated on a transfer randomly initialized along the investigated departing orbit. The generated trajectory is visualized in the spatial and xy-perspective in Fig. 7.23(a-b) in the rotating Earth-Moon system centered at the Moon. The arrival and departing orbits are depicted in red and blue, respectively. The spacecraft path is presented in black, and the starting and arrival spacecraft states are reported with blue and red circles, respectively. Moreover, the direction of the maneuvers is represented by magenta arrows. In particular, the maneuvers in Fig. 7.23(a-b) exhibit large magnitudes at the beginning of the transfer, allowing the spacecraft to depart from the initial orbit, and during the approach to the arrival orbit after the lunar flyby. Small impulsive maneuvers are observed prior to approaching the lunar flyby, and during the last phase of the transfer, exhibiting bounded motion with respect to the arrival orbit. The total maneuver magnitude and total reward are $\Delta v_{\text{tot}} = 443.36$ m/s and $r_{\text{tot}} = 119.02$, respectively. Fig. 7.23(a-b) supports the applicability of the trained policy to connect two orbits from different families of orbits near the Moon.

Transfer between orbits near orbital resonances

A third and last policy is trained to generate sequences of maneuvers enabling transferring a spacecraft between two orbits near resonances in Earth-Moon CR3BP. In particular, a spatial departing orbit near the 2:1 resonance with $C_J = 2.160$ and period $T = 27.319$ days, and a spatial arrival orbit near the 3:2 resonance with $C_J = 2.921$ and period $T = 53.365$ days are selected. The average value throughout the training process and a sample trajectory are depicted in Fig. 7.24(a-

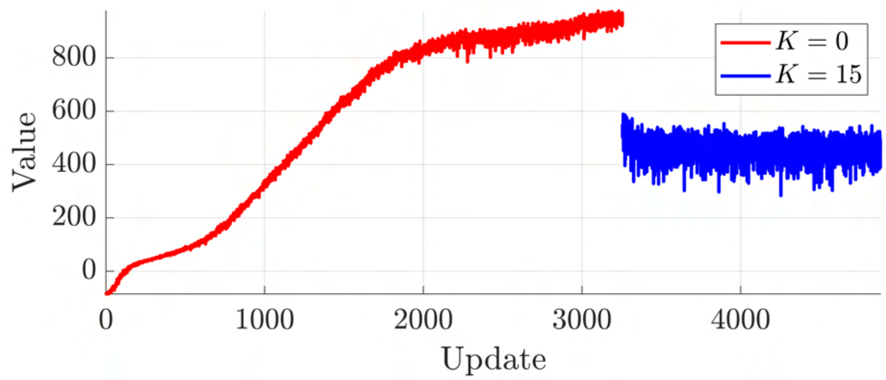

Figure 7.22: Average value per update for a transfer from a southern halo orbits near L_2 to a southern halo orbit near L_1 in the Earth-Moon CR3BP.

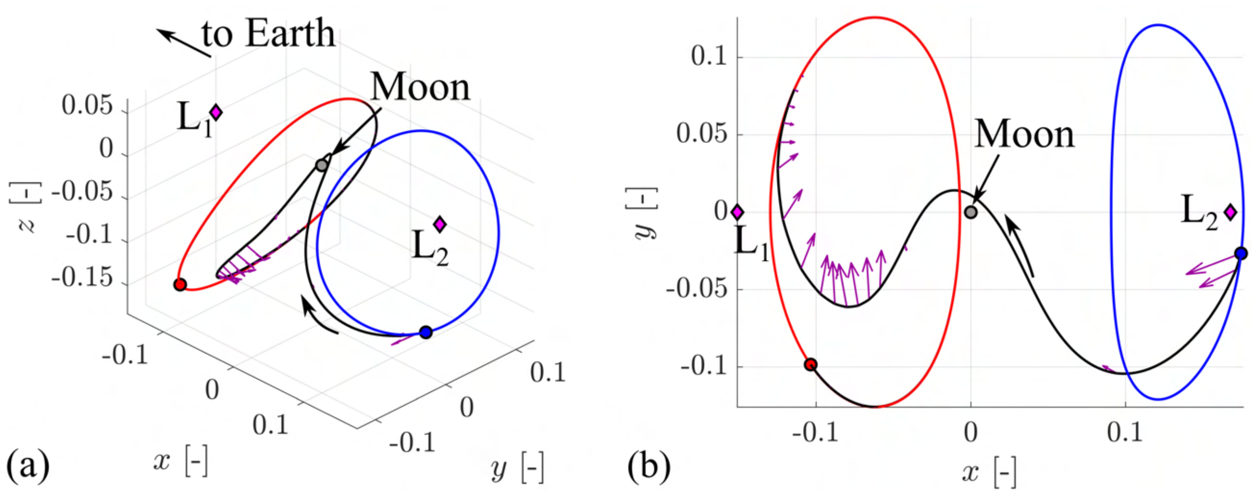

Figure 7.23: Sample transfer from a southern halo orbits near L_2 to a southern halo orbit near L_1 in the Earth-Moon CR3BP, generated by the converged policy in Fig. 7.22: (a) spatial and (b) xy-perspective.

c). The training environment for this final example differs from the baseline scenario only for a selected weighting factor $K = 3$ in the reward formulation for the second training after the policy transfer. The adopted relative weight in the reward formulation allows to train a policy with a positive average value after the transfer, due to the relatively large separation of energy level between the selected orbits. Additional training might generate a policy with superior performance, as demonstrated by the absence of a final plateau after the transfer in Fig. 7.24(b). A sample trajectory, generated with the trained policy at the end of the training process is presented in Fig. 7.24(a-c). The arrival and departing orbits are depicted in red and blue, respectively. The spacecraft path is presented in black, and the starting and arrival spacecraft states are reported with blue and red circles, respectively. Moreover, the direction of the maneuvers is represented by magenta arrows. In particular, the maneuvers in Fig. 7.24(a,c) exhibit large magnitudes and small time separation during the approach to the arrival orbit. Relatively small impulsive maneuvers and large coasting arcs are observed during the remaining part of the transfer. The total maneuver magnitude and total reward are $\Delta v_{\text{tot}} = 735.74$ m/s and $r_{\text{tot}} = 406.37$, respectively. The total maneuver magnitude is larger with respect to the previous examples due to the largeer difference in energy between the investigated periodic orbits. Future research focusing on generating policies with superior performance might enable transfers with reduced total maneuver magnitude.

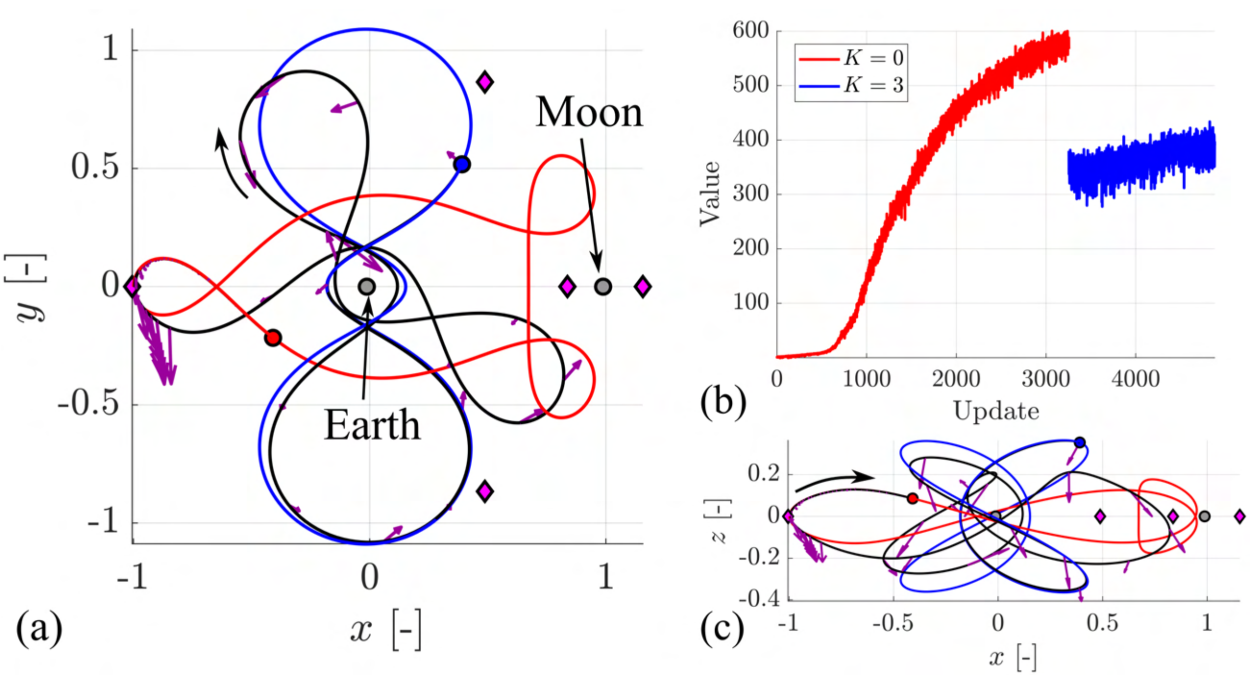

Figure 7.24: Training process and sampled trajectory for a scenario of a transfer between two spatial orbits near resonances in the Earth-Moon CR3BP: (a) xy-perspective of the sampled trajectory, (b) average value over the training process, and (c) xz-perspective of the sampled trajectory.

Overall, the designed environment is applicable, with either null or slight modifications, to a variety of transfer orbit scenarios, here demonstrated only in the Earth-Moon system. The methodology can also be applied to train policies for autonomous generation of maneuver sequences in distinct systems.

7.3 Transfer Design in a Family of Orbits in the Earth-Moon System

In this section, an orbit transfer scenario between members of a family of orbits is designed and investigated. The results introduced in this section represent an exploratory study, assessing the applicability of reinforcement learning to generate policies for challenging transfer scenarios in a family of orbits. Additional insights can be generated with extensive exploration of different parameters governing the training process and the identified RL scenario, as well as with a thorough comparison with alternative state-of-the-art solutions for generating transfers between periodic orbits. These aspects are not addressed in the presented investigation, but represent interesting avenues for future research. For this investigation, the proposed scenario represents a generalization of the orbit transfer framework between prescribed members, presented in Sec. 7.2. Specifically, a policy is trained in this section to generate sequences of impulsive maneuvers enabling a spacecraft to transfer between members of a family of orbits in the Earth-Moon system. The combination of departing and arrival orbits from the family is selected at the beginning of each episode in the reinforcement learning formulation: therefore the departing and arrival orbits likely change between distinct episodes. The orbit transfer scenario is first presented in Sec. 7.3.1, and converted into a reinforcement learning framework in Sec. 7.3.2. Then, a policy is trained and validated in a variety of orbit transfer scenarios in Sec. 7.3.3.

7.3.1 Scenario Overview

The orbit transfer between members of a family of orbits is modeled after the scenario presented in Eq. (7.10) for the transfer design between prescribed orbits in the Earth-Moon CR3BP, and it is motivated by recent interests in cislunar activities [141, 142]. In particular, the family of southern halo orbits near L_2 in the Earth-Moon system is used as a baseline to generate transfers between different orbits. Different combinations of departing and arrival orbits can be selected from this family. Specifically, Fig. 7.25 visualizes four feasible combinations of orbits in the analyzed family. In each subfigure, the entire framework of orbits, used in this initial investigation to provide candidate orbits, is visualized with semi-transparent gray markers. For each frame, the selected combinations of orbits are represented by a departing and arrival orbit, respectively depicted in blue and red. In the investigated scenario, the family of southern halo orbits near L_2 in the Earth-Moon CR3BP is approximated by 200 distinct members. The energy levels and the periods associated with the first member of the family are $C_{J,1} = 3.1517$ and $T_1 = 14.828$ days, while the energy level and period of the last member of the family is $C_{J,f} = 3.0330$ and $T_f = 7.474$ days.

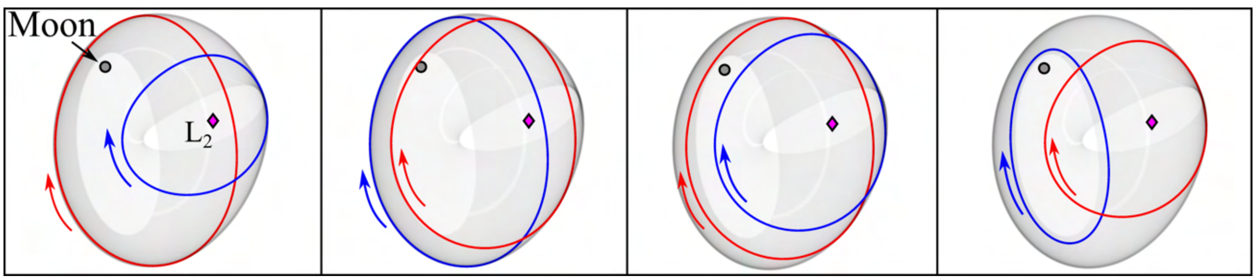

Figure 7.25: Example of combinations of departing (blue) and arrival (red) orbits from the family of southern halo orbits near L_2 in the Earth-Moon CR3BP.

A sequence of impulsive maneuvers is designed to generate a transfer that departs the initial orbit and arrives to the final orbit orbit. Both orbits are selected before initializing the transfer. Each maneuver is modeled as an impulsive three-dimensional vector $\Delta\boldsymbol{v}$, and two consecutive impulsive maneuvers in the sequence are separated by a propagation time $\Delta t \in [\Delta t_{\min}, \Delta t_{\max}]$. Therefore, at each maneuver location the devised methodology determines the impulsive three-dimensional vector $\Delta\boldsymbol{v}$, and the propagation time Δt separating the current maneuver from the subsequent maneuver. Eventually, a maximum number of maneuvers τ is applied for each transfer: this represents an upper boundary constraint since a relative small displacement from the arrival orbit can potentially be obtained by using a number of maneuvers $\bar{\tau} < \tau$.

7.3.2 Translating the Scenario into a Reinforcement Learning Problem

The orbit transfer problem between members of a family of orbits is translated into a reinforcement learning scenario by the definition of: an action and an environment state vector, a reward, a step, an episode, and an environment. The environment is designed to be agnostic with respect to the system and the considered family of orbits, following the orbit transfer scenario between prescribed orbits in Sec. 7.2.2. The agent interacts with the environment, modeled with the Earth-Moon dynamical model, to learn a policy for the autonomous maneuver generation. A single interaction between the agent and the environment is executed within a step. A maximum number τ of consecutive steps generates an episode. At the beginning of each episode, corresponding to an initial step, the departing and arrival orbits are randomly selected from the framework of 200 distinct members used to approximate the family of southern halo orbits around L_2 in the Earth-Moon system. To prevent the generation of transfers between two nearby members of the family, the pair of departing and arrival orbits is enforced to be separated by at least one member of the leveraged family. Thanks to the randomness of the orbit selection process, two episodes are likely initialized with distinct combinations of arrival and departing orbits. After the orbit selection, a nondimensional initial spacecraft state $\boldsymbol{x}_0 \in \mathbb{R}^6$ is randomly initialized along the departing orbit, in the rotational Earth-Moon frame. Then, the deviation of $\boldsymbol{x}_0$ from the closest state of the arrival orbit, $\delta\boldsymbol{x}_0 \in \mathbb{R}^6$, is computed. The information comprising the spacecraft state, the deviation from the arrival orbit, and the arrival and departing orbits is leveraged by the actor to generate an action $\boldsymbol{a}_0 \in \mathbb{R}^4$. The action contains information on the impulsive maneuver $\Delta\boldsymbol{v}$ and the time Δt separating the beginning of the current step from the subsequent one. After augmenting $\boldsymbol{x}_0$ with the generated maneuver $\Delta\boldsymbol{v}$, the state is propagated with the dynamical model of the Earth-Moon CR3BP for a time Δt. The spacecraft state at the end of the current step is then leveraged to initialize an environment state for the beginning of the subsequent state. The process repeats until termination criteria are met at the end of a step. A new step is initialized if the training termination criteria are not met.

A mathematical formulation of the environment state and the action is proposed to incorporate information of the spacecraft state and the considered orbit transfer scenario. The environment state is defined as:

$$\boldsymbol{s}_i = [\tilde{\boldsymbol{x}}_i, \delta\tilde{\boldsymbol{x}}_i, \tilde{s}_i, \tilde{o}_d, \tilde{o}_a] \in \mathbb{R}^{15} \tag{7.15}$$

where the tilde operator $\tilde{(\cdot)}$ is used to indicate quantities scaled within $[-1, 1]$. The environment state in Eq. (7.15) comprises: the scaled spacecraft state at the beginning of the step $\tilde{\boldsymbol{x}}_i$; the scaled displacement of $\boldsymbol{x}_i$ from the closest state of the arrival orbit, $\delta\tilde{\boldsymbol{x}}_i$; the maneuver number along the current episode $\tilde{s} = 2i/\tau - 1$, with $i \in \{\tau, \tau - 1, \ldots, 0\}$; two scaled indices $\tilde{o}_d$, $\tilde{o}_a \in [-1, 1]$, reflecting the order of the departing and arrival orbit in the framework of 200 members of the family, respectively. The environment state is leveraged by the actor at the beginning of each step to generate an action vector, formulated as:

$$\boldsymbol{a}_i = [\Delta\tilde{\boldsymbol{v}}_i, \Delta\tilde{t}_i] \in \mathbb{R}^4 \tag{7.16}$$

The definition of the action is identical to the orbit transfer scenario with prescribed orbits detailed in Sec. 7.2.2. In particular, the impulsive maneuver is retrieved as $\Delta\boldsymbol{v}_i = \nu\Delta\tilde{\boldsymbol{v}}_i$, with $\nu = 3 \times 10^{-3}$, and the propagation time is converted within the time interval $\Delta t \in [\Delta t_{\min}, \Delta t_{\max}]$ using Eq. (7.11).

A reward function is formulated to reflect the benefit of a state-action pair towards the fulfillment of an efficient orbit transfer. The reward, representing the output of the reward function, is computed at the end of each step. For this framework, the reward function formulation is inherited with minor modifications from the orbit transfer scenario between prescribed orbits. Indeed, a piecewise reward function is designed depending on the spacecraft location. Two regions are identified as:

1) Internal region: the spacecraft is assumed in the internal region if the position vector is contained in a boundary set $\mathscr{D} \subset \mathbb{R}^3$. The boundary set $\mathscr{D}$ is constructed as a box in the configuration space that contains the investigated family of orbits. In particular, the upper and lower bounds along each dimension are set proportional to the maximum and minimum

values of the associated dimensions of the departing and arrival orbit. For example, the lower boundary of $\mathscr{D}$ along the x-direction, $\boldsymbol{x}_l$, is computed as:

$$\boldsymbol{x}_l = \delta_{1,x} \min[\min(\boldsymbol{\varsigma}_{1,x}), \min(\boldsymbol{\varsigma}_{2,x}), \ldots, \min(\boldsymbol{\varsigma}_{200,x})] \tag{7.17}$$

where $\delta_{1,x} > 0$ is a scaling factor, while $\boldsymbol{\varsigma}_{i,x}$ represent the x-components of the states along the i-th orbit in the family. Similarly, the upper boundary of $\mathscr{D}$ along the x-dimension, $\boldsymbol{x}_u$, is set equal to:

$$\boldsymbol{x}_u = \delta_{2,x} \max[\max(\boldsymbol{\varsigma}_{1,x}), \max(\boldsymbol{\varsigma}_{2,x}), \ldots, \max(\boldsymbol{\varsigma}_{200,x})] \tag{7.18}$$

where $\delta_{2,x} > 0$. The upper boundary $\boldsymbol{x}_u$ corresponds to the scaled maximum among the maxima of the family of orbits along the x-dimension. The lower and upper boundaries of $\mathscr{D}$ along the y- and z-components are generated using the minima and maxima along the second and third position component of the batch of orbits, respectively. An example of the internal region $\mathscr{D}$ for the family of southern halo orbits near L_2 in the Earth-Moon system is represented in the spatial, xy- and xz-projections, in Fig. 7.26(a-c), using a semi-transparent orange surface. In the internal region, the reward is formulated as a weighted sum of two terms, reflecting: displacement from the arrival orbit; amount of consumed propellant at the beginning of the associated step.

2) External region: the spacecraft is considered to reside in the external region when the position vector is not contained in $\mathscr{D}$. When the spacecraft is in the external region, a penalizing negative reward is assigned to the associated state-action pair, due to significant departure from the internal region. After assigning the penalizing reward, the associated episode is terminated.

The mathematical formulation of the reward for this scenario is identical to the orbit transfer scheme between prescribed orbit, outlined in Eq. (7.14). Note that the leveraged scaling factors governing the dimension of the internal region $\mathscr{D}$ might limit the explored transfer geometries. A conceptual overview of the training process for the orbit transfer scenario between members of a

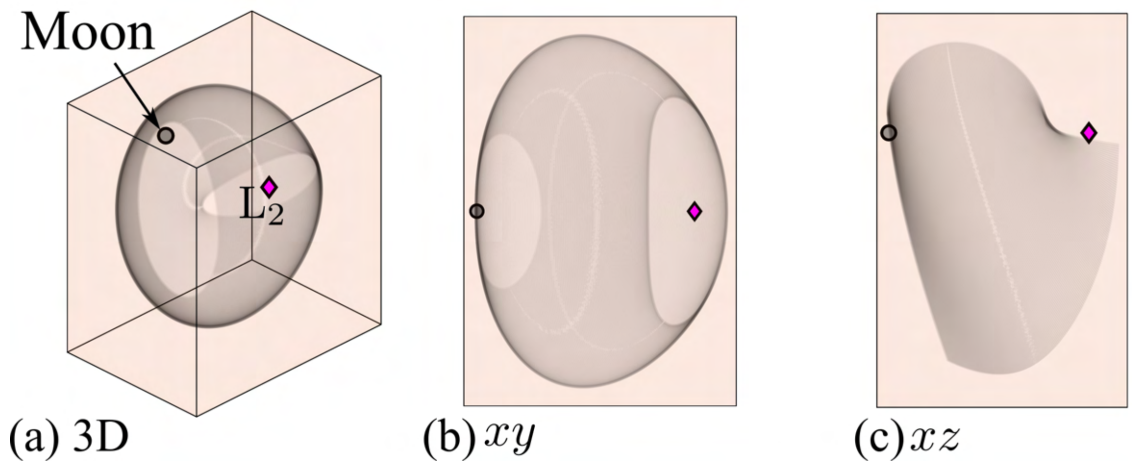

Figure 7.26: Example of internal region $\mathscr{D}$ for the family of L_2 southern halo orbits in the Earth-Moon system, depicted as a semi-transparent orange surface.

family of orbits is presented in Fig. 7.27. Ultimately, the set of hyperparameters governing the PPO cost function, the training process, and the neural network structures is selected to be identical to the set used for the orbit transfer scenario between prescribed orbits, and detailed in Sec. 7.2.2

7.3.3 Generating Transfers between Families of Orbits

The RL maneuver planner for the orbit transfer scenario between members of a family of orbits is trained and validated in this section. The scenario is initially tested to autonomously generate sequences of maneuvers to connect orbits of the southern halo family near L_2 in the Earth-Moon CR3BP. The designed environment is then tested to connect other families of orbits in the Earth-Moon system. Recall this investigation is primarily focused on exploring the applicability of reinforcement learning to a challenging orbit transfer scenario: policies with superior performance might be generated with extensive exploration of the involved parameter space governing the training process and the underlying RL scenario.

7.3.3.1 Training the Baseline Scenario

A policy is trained to generate a sequence of maneuvers connecting two members in the family of southern halo orbits near L_2 in the Earth-Moon system. The orbits are selected randomly at

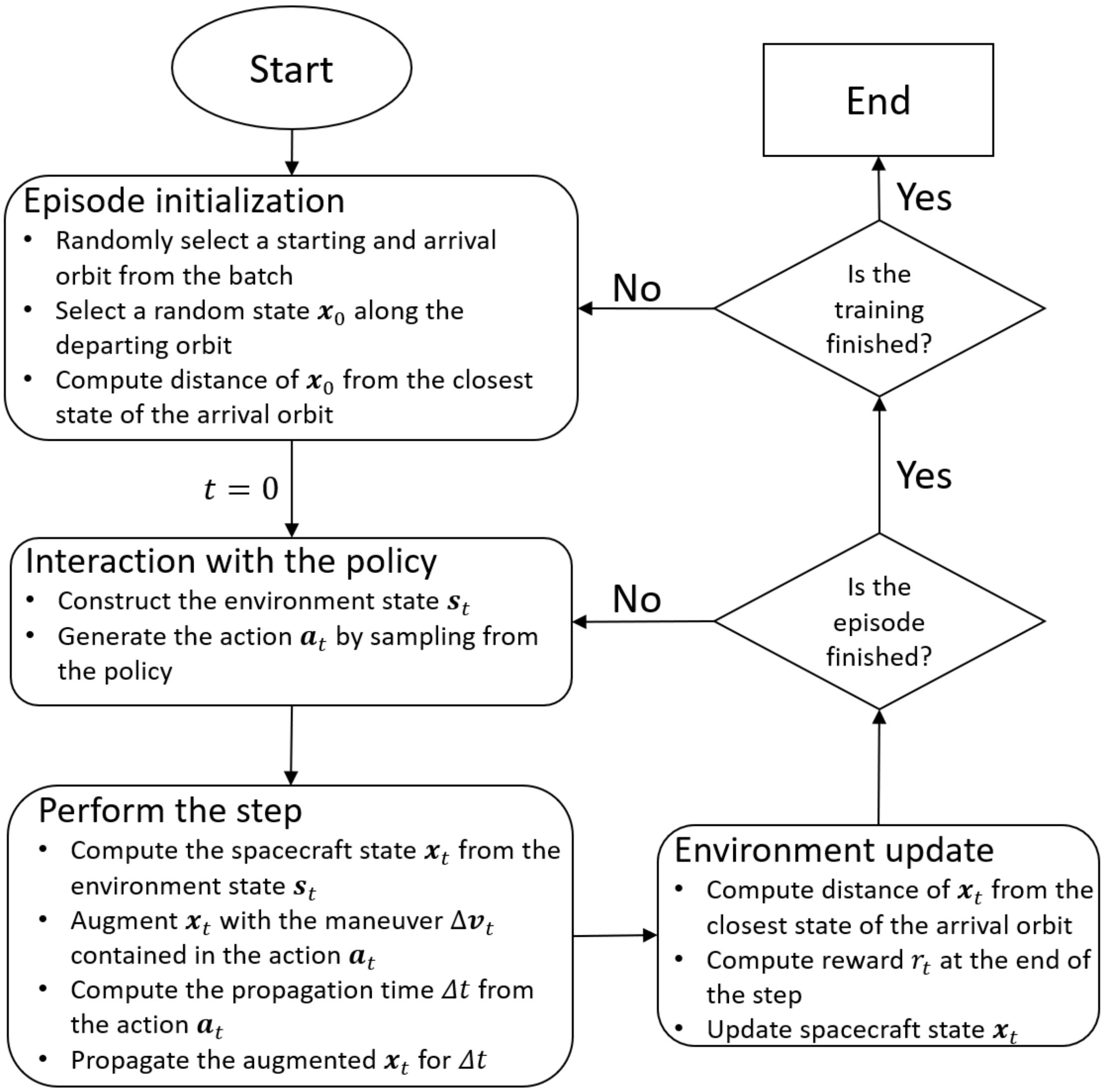

Figure 7.27: Flowchart for the orbit transfer RL environment between members of a family of orbits.

the beginning of each episode, and a maximum number of maneuvers $\tau = 50$ is enforced for each transfer scenario. Similarly to the orbit transfer between prescribed orbits, the propagation time at each step is constrained within $[\Delta t_{\min}, \Delta t_{\max}]$, with $\Delta t_{\min} = 5$ hours and $\Delta t_{\max} = 60$ hours. The selected scaling factors for the internal region $\mathscr{D}$ are $\boldsymbol{\delta}_1 = [\delta_{1,x}, \delta_{1,y}, \delta_{1,z}] = [1.02, 1.3, 1.3]$ and $\boldsymbol{\delta}_2 = [\delta_{2,x}, \delta_{2,y}, \delta_{2,z}] = [0.98, 0.7, 0.7]$: the leveraged scaling factors allow larger exploration from the policy with respect to the prescribed orbits scenario, enabling a wider array of transfer geometries. Ultimately, the relative weighting factor for the action contribution in the reward formulation is set to $K = 1$: larger weights are empirically linked to policies with worse performances.

The identified set of hyperparameters defines the investigated environment, allowing training of a policy for the autonomous generation of maneuver sequences for transfers between members of a family of orbits. Motivated by the results of the simpler environment detailed in Sec. 7.2.3, transfer learning is again leveraged to generate a policy with the described set of hyperparameters. The training process is performed with a maximum of 1.5×10^7 steps, divided in two consecutive phases. Initially, a maximum set of 1×10^7 steps and a weighting factor $K = 0$ in the reward formulation are used to train a policy that emphasizes transfers with relatively low final displacements from the arrival orbit. After completion of the first training, the converged actor and critic networks are used to initialize the neural networks for a second training, using a maximum of 5×10^6 steps and a relative weighting factor $K = 1$ in the reward formulation. The second part of the training process allows the policy to generate transfers with reduced total maneuver magnitude. To prevent policy disruption, and construct a policy that generates transfers towards the final orbit with a relatively low maneuver magnitude, the learning rate for the second part of the training process is initially set at $\alpha_0 = 5 \times 10^{-4}$, and progressively reduced with a sequence of 8 equally distributed step functions to an ultimate learning rate $\alpha_f = 5 \times 10^{-7}$. The trend of the value, averaged over the steps available at each update, is visualized in Fig. 7.28: the initial training, associated with a weighting factor $K = 0$ is reported in red, while the second part of the training, accounting for the maneuver magnitude within the reward formulation with $K = 1$ is depicted in blue. Fig. 7.28 demonstrates the value of pretraining a policy in a simplified scenario to accelerate the training process in a more challenging scenario, associated with a more convoluted reward formulation. Moreover, the initial training with $K = 0$ is empirically necessary to construct a policy that generates transfers connecting pairs of orbits within the family that are relatively distant in the design space.

7.3.3.2 Validating the Baseline Scenario

After training, the converged policy is validated using a batch of 5000 episodes, corresponding to distinct transfers between potentially different combinations of departing and arrival orbits. Five

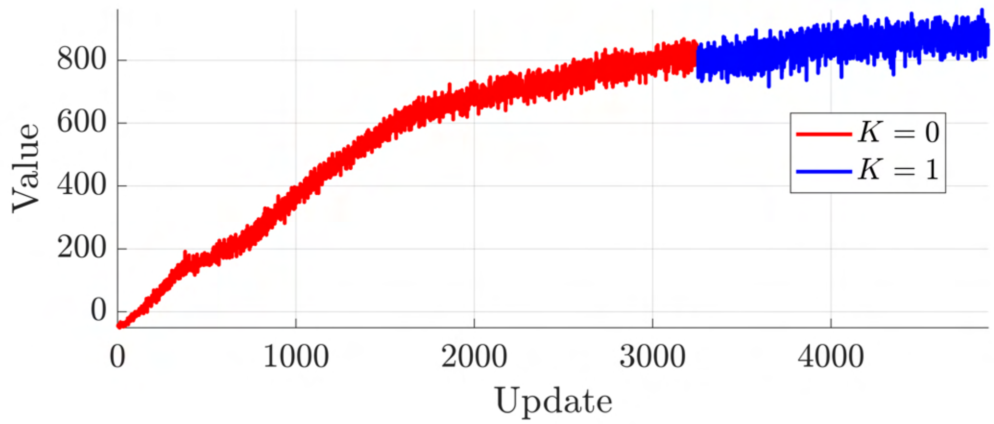

Figure 7.28: Average value per update for a transfer between members of a family of southern halo orbits near L_2 in the Earth-Moon CR3BP.

metrics are used to evaluate the performance of the generated policy: final position displacement from the arrival orbit $\|\delta \boldsymbol{r}_{\text{end}}\|$; final velocity displacement from the arrival orbit $\|\delta \boldsymbol{v}_{\text{end}}\|$; cumulative reward over an episode r_{tot}; total maneuver magnitude over an episode Δv_{tot}; and number of performed steps per episode $\bar{\tau}$. The analyzed policy is then evaluated by averaging the five metrics on the validation set of 5000 trajectories. The averaged values are reported in Table 7.4.

$\|\delta \boldsymbol{r}_{\text{end}}\|$ [km]	$\|\delta \boldsymbol{v}_{\text{end}}\|$ [m/s]	Δv_{tot} [m/s]	r_{tot} [-]	$\bar{\tau}$
461.5592	7.6143	289.162	1874.858	49.581

Table 7.4: Average final position displacement, final velocity displacement, total maneuver magnitude and total reward over an episode and number of maneuvers per episode for a policy trained with transfer learning and evaluated using a set of 5000 initial conditions.

The quantities presented in the table reflect the complexity of the investigated environment. Indeed, the position and velocity displacements over the batch of 5000 trajectories are one and two order of magnitudes larger, respectively, than the same metrics evaluated on the batch of 1000 trajectories in the prescribed orbit transfer scenario, reported in Table 7.3. Also, the average total maneuver magnitude is slightly larger with respect to the transfer scenario between prescribed orbits, reflecting the higher complexity of the environment and the presence of transfers connecting relatively distant orbits within the family. However, alternative approaches applied to transfer design between different southern halo orbits near L_2 in the Earth-Moon system demonstrate similar total maneuver magnitudes [86, 141]. Ultimately, 71 transfers do not complete the maximum of 50 maneuvers per episode, resulting in the 1.42% of trajectories exceeding the internal region $\mathscr{D}$.

Additional insights into the performance of the generated policy can be obtained by visualizing the distributions of the batch of trajectories over the evaluation metrics. In particular, the values of the final position and velocity displacements from the arrival orbit, $\|\delta\boldsymbol{r}_{\text{end}}\|$ and $\|\delta\boldsymbol{v}_{\text{end}}\|$, are converted into histograms and reported in the top and bottom, respectively, of Fig. 7.29. For each distribution, the trajectories completing 50 maneuvers are represented with blue semi-transparent columns, while the trajectories exhibiting premature termination are populate semi-transparent red columns. The average values, indicated in Table 7.4, are highlighted using vertical dashed red lines. The distributions highlight how the majority of the trajectories in the batch is uniformly located near the respective means. However, a few samples tend to generate large position and velocity deviations from the arrival orbit. In particular, trajectories terminating without performing 50 impulsive maneuvers are associated with large final position and velocity deviations because they exceed the internal region $\mathscr{D}$.

To investigate the characteristics of the transfers exhibiting large final position and velocity displacements, the validation metrics are evaluated with respect to the combination of arrival and departing orbits associated with each transfer. Recall that the original framework, leveraged to randomly assign a departing and arrival orbit at the beginning of each episode, is composed of 200 members of the family of southern halo orbits near L_2 in the Earth-Moon system. To distinguish among the members of the family and generate a monotonic series associated with the family members, the distance from the Moon at the apolune $r_{2,\text{Apo}}$ is selected to represent each member of the family. Then, each transfer in the batch of 5000 trajectories, associated with a specific combination of departing and arrival halo orbits, is colored in shades from blue to red, and reported with circles in the center of Fig. 7.30. The x- and y-axis of the central figure reflect the distance from the Moon at the apolune of the departing and the arrival orbits, respectively. To generate additional insights on the variety of possible transfer geometries, four transfers are highlighted in

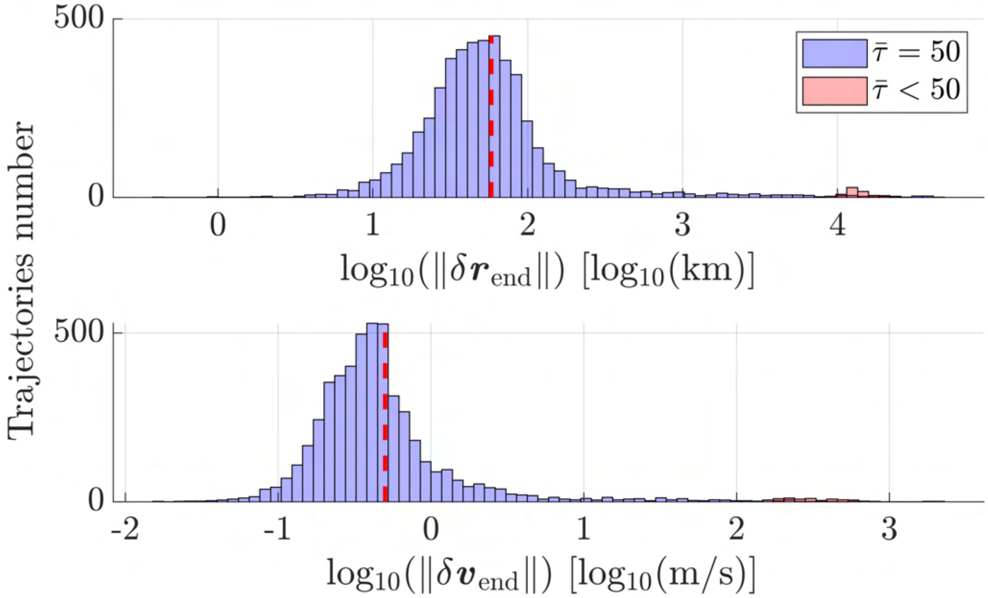

Figure 7.29: Histograms of (top) final position displacement and (bottom) final velocity displacement from the arrival orbits. Constructed from a batch 5000 trajectories generated in the Earth-Moon system, using a trained RL policy for spacecraft transfer between members of the L_2 halo orbits.

the central figure with red circles, and the connected orbits are expanded on the sides of Fig. 7.30. In each lateral frame, the departing and arrival orbits are colored in blue and red, respectively, while the framework of 200 members within the family is visualized with transparent gray markers. The coloring in the central figure suggests that the required total maneuver magnitude Δv_{tot} is a function of the relative distance, and therefore the relative energy, between the connected orbits. In particular, the region near the diagonal, spanning the central frame from the bottom-left region to the top-right corner, is populated by transfers requiring the lowest amount of total propellant: in this region, the connected orbits in the transfer exhibit moderate relative distance. The total maneuver magnitude approximately monotonically increases when moving towards the top-left and bottom-right corners from the central diagonal. In these regions, the departing and arrival orbits present large relative distance and energies, therefore increasing the required total maneuver magnitude for generating the transfer. The 71 transfers corresponding to trajectories prematurely terminating their episode with $\bar{\tau} < 50$ are circled in black and are entirely located in the bottom-right corner of the central frame in Fig. 7.30. The transfers populating this region present a challenge for the trained policy due to the close passages with the Moon during the approach phase to the arrival orbit. Conversely, the trained policy can complete every transfer which is located in the top-left corner, associated with departing orbits that have a low-altitude apolune. Both the color and the difficulty presented by the trained policy reflect the general challenging scenarios of transfers approaching orbits with low perilune. A similar representation is leveraged to visualize the final position and velocity displacements from the arrival orbit with respect to the combination of departing and arrival orbits. In particular, Fig. 7.31(a) report the final displacement $\|\delta \boldsymbol{r}_{\text{end}}\|$, while the final velocity deviation $\|\delta \boldsymbol{v}_{\text{end}}\|$ is reported in Fig. 7.31(b). These representations confirm the validity of the insights generated from Fig. 7.30. Moreover, the figure suggests that large final deviations from the arrival orbit are associated with transfers with a low apolune arrival orbit, reflecting the sensitivity of the region.

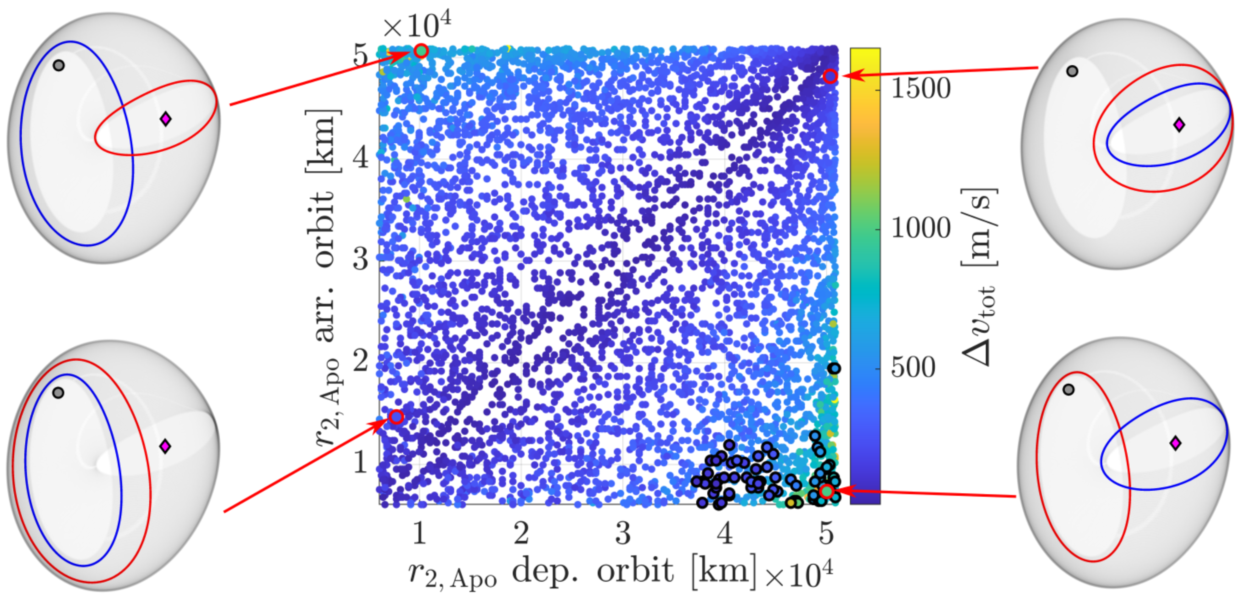

Figure 7.30: Representation of the total maneuver magnitude for the batch of 5000 transfers generated in the Earth-Moon CR3BP.

Ultimately, a sampled subset of the transfers from the validating batch of 5000 trajectories is visualized to demonstrate the performance of the trained policy to autonomously generate sequences

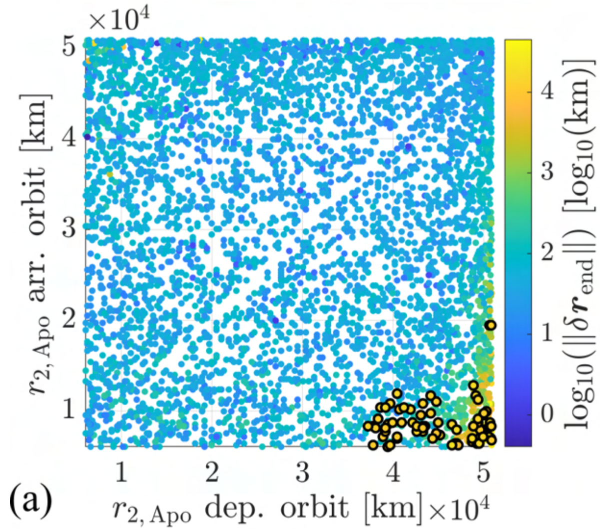

Figure 7.31: Representation of (a) final position deviation and (b) final velocity deviation from the arrival orbits for the batch of 5000 transfers generated in the Earth-Moon CR3BP.

of impulsive maneuvers and enable transfers between distinct L_2 halo orbits in the Earth-Moon system. Specifically, four transfers are visualized in the configuration space in Fig. 7.32: in the central row, each transfer is presented in the three-dimensional view, while in the bottom row the transfers are visualized from the planar xy-perspective. In each frame, the departing and arrival orbits are reported in blue and red, respectively, while the leveraged framework of southern halo orbits near L_2 is depicted with semi-transparent gray markers. The transfers are visualized with black arcs, initialized on the blue circle along the departing orbit and terminating on the red circle along the arrival orbit. Along each transfer, the scaled maneuver directions and magnitudes are reported using magenta arrows. The maneuvers generally present large magnitudes for the first part of the transfers, associated with the departing arc from the initial orbit and the approaching phase to the arrival orbit. The converged policy succeeds to autonomously generate maneuver sequences that transfer the spacecraft from the departing to the arrival orbit, for the transfers depicted in Fig. 7.32. The geometry of the transfer is influenced by the selected combination of orbits, and the total maneuver magnitude, presented in the first row of Fig. 7.32, reflects the challenge of the transfer scenario, the distance of the connected orbits, and their distinct energy levels. The visualized total maneuver magnitudes demonstrate similar values to other studies leveraging alternative approaches for maneuver design in similar orbit transfer regimes [86, 141].

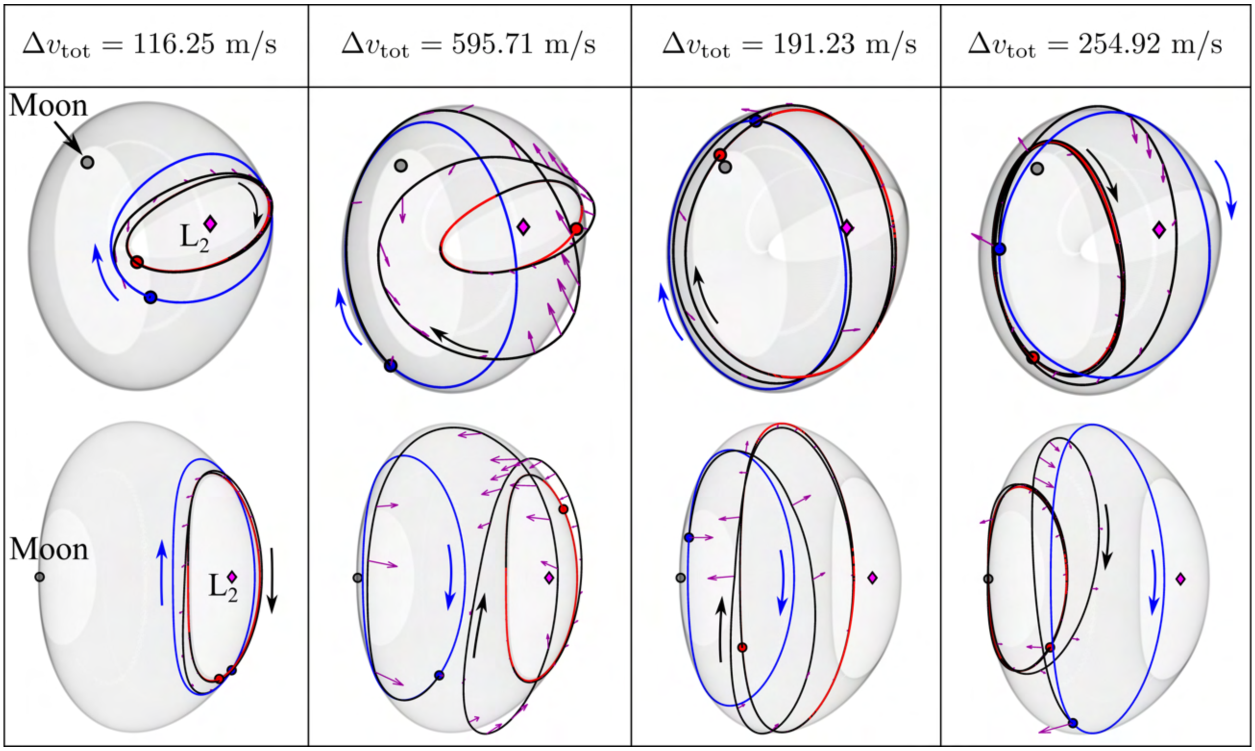

Figure 7.32: Examples of transfers between distinct combination of southern halo orbits near L_2 in the Earth-Moon system.

7.3.3.3 Application to Different Families of Orbits

The orbit transfer design scenario between members of a family of orbits is designed to be agnostic of the leveraged family. Therefore, the same approach can be applied to generate sequences of maneuvers between members of multiple families of orbits in the Earth-Moon CR3BP. Two different and challenging transfer scenarios are here introduced and analyzed. Recall the scope of this investigation is to assess the feasibility of the presented approach for autonomous generation of sequences of maneuvers for orbit transfer scenario: additional exploration for policy improvement, and comparison with distinct state-of-the-art techniques for transfer design represent avenues for future research.

Transfers between halo orbits near $\mathbf{L}_1$

A first policy is trained to connect members of the family of southern halo orbits near L_1 in the Earth-Moon system. In particular, the family is approximated with 200 members, uniformly distributed along the family. The energy level and the periods associated with the first and last members of the considered set of orbits are $C_{J,1} = 3.174$, $T_1 = 11.911$ days and $C_{J,\text{end}} = 3.003$, and $T_{\text{end}} = 8.294$ days. The policy trained for this scenario leverages the same set of training and scenario-dependent parameters employed for the orbit transfer between members of the family of orbits near L_2, detailed in Sec. 7.3.2. In particular, the training process is separated in two phases: an initial learning for a maximum number of steps of 1×10^7 with $K = 0$, used to uncover transfer solutions, is followed by a second learning using a maximum of 5×10^6 steps with $K = 1$ used to reduce the total maneuver magnitude over the uncovered solutions. The policy converged at the termination of the initial learning is employed to initialize the actor and critic networks for the second phase. The time series of the average value is visualized in Fig. 7.33: the value associated with the first and second phase is reported in blue and red, respectively, visually demonstrating the benefit of transfer learning to generate a performing initializing solution for the second part of the training. Similarly to the scenario detailed in Sec. 7.3.2, an initial training process is necessary to uncover transfer solutions that connect relatively distant orbits.

The converged policy is used to generate sequences of maneuvers for four distinct transfers, reported in Fig. 7.34. In the figure, each column is associated to a distinct transfer, transferring a randomly sampled initial condition to a desired arrival orbit. Each transfer is visualized in the spatial configuration space and in the xy-projection in the central and bottom row, respectively. In each frame, the departing and arrival orbits are reported in blue and red, respectively, the transfer is visualized in black, while the starting location along the departing orbit and the arrival state along the arrival orbit are reported with blue and red circles. The direction and magnitude of the scaled maneuvers are represented along each transfer with magenta arrows. The maneuvers generally present large magnitudes for the first part of the transfers, associated with the departing arc from the initial orbit and the approaching phase to the arrival orbit. The performance of the visualized

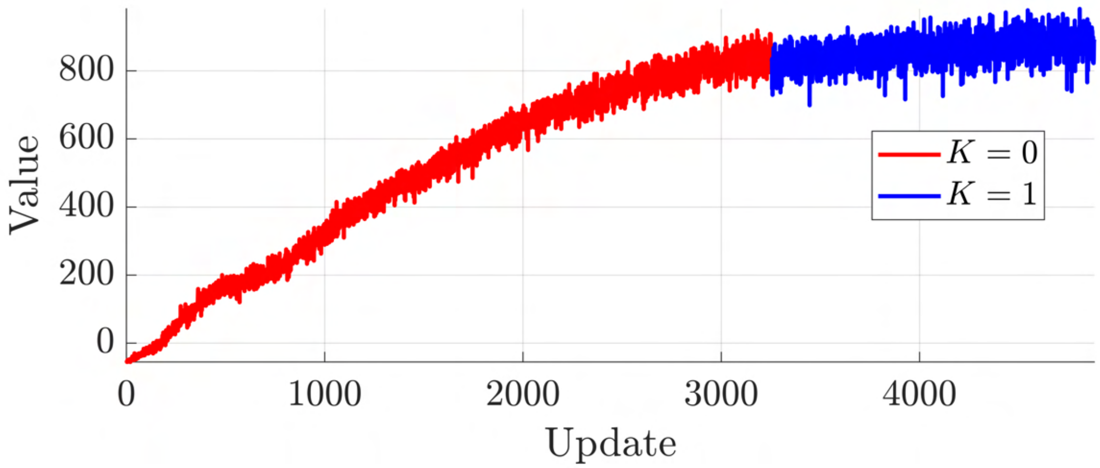

Figure 7.33: Average value per update for the orbit transfer scenario between members of a family of southern halo orbits near L_1 in the Earth-Moon system.

transfers are highlighted along the first row in Fig. 7.34, reporting the associated total maneuver magnitude. In particular, transfers associated with combinations of orbits with relative low distance and energy are accomplished with low to moderate amount of propellant, while transfers between relatively distant orbits require larger amount of Δv_{tot}. The visualized total maneuver magnitudes demonstrate similar values to other studies leveraging alternative approaches for maneuver design in similar orbit transfer regimes [86].

Transfers between halo orbits near L_2 and L_1

In the previous examples, the policies are trained to construct sequences of maneuvers for transfers between members of the same orbit family. However, the flexibility of the investigated scenario also allows to train policies generating maneuver sequences for transfers between members of distinct orbit families. Indeed, the environment does not require the arrival and departing family of orbits to be identical. This feature is demonstrated in a final example, where a policy is trained to generate sequences of impulsive maneuvers enabling a spacecraft to depart from members of the southern halo orbits near L_2, and approach members of the southern halo orbits near L_1 in the Earth-Moon system. Similarly to the previous examples, the combination of arrival and departing

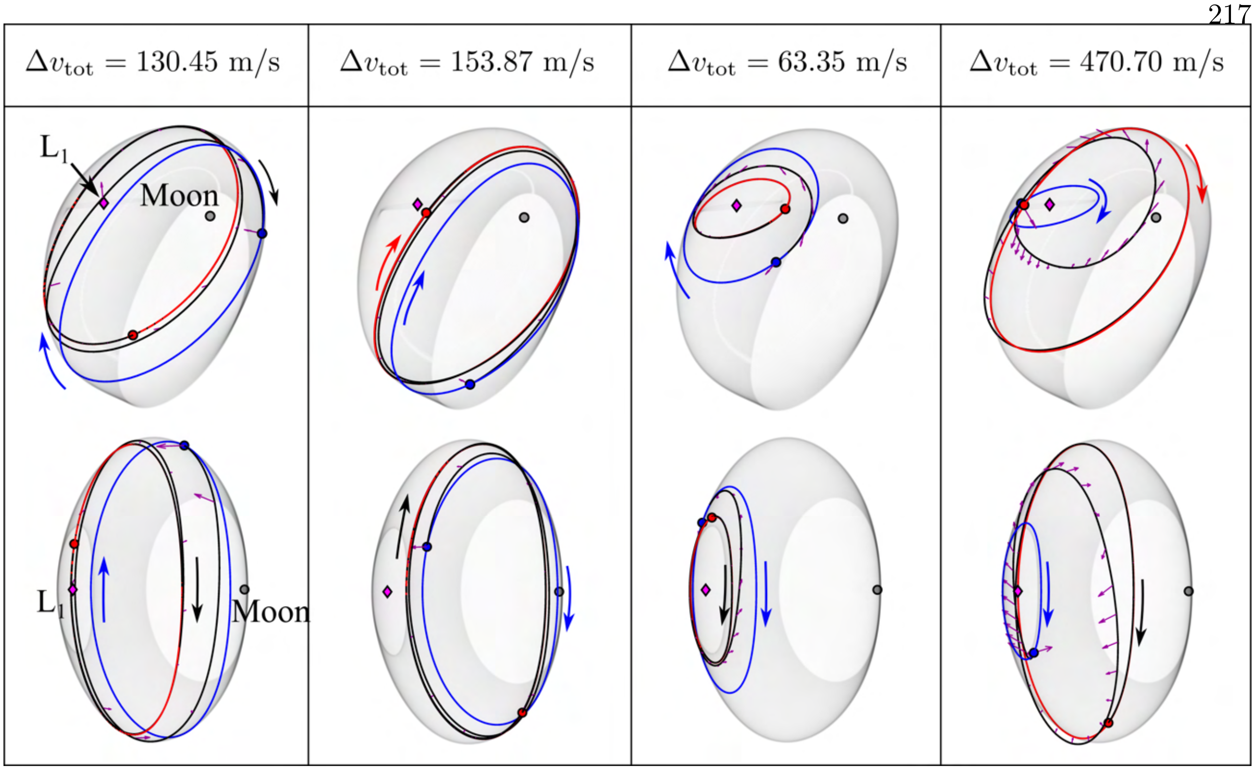

Figure 7.34: Examples of transfers between distinct combination of southern halo orbits near L_1 in the Earth-Moon system.

orbits are randomly selected at the beginning of each episode. Moreover, the same set of hyperparameters leveraged to generate transfers for the L_2 and L_1 southern halo orbit families is here used. For this example, the set of 200 southern L_2 halo orbits used to generate the transfers in Fig. 7.32 forms the departing orbit family; similarly, the set of 200 southern L_1 halo orbits used to generate the transfers in Fig. 7.34 constitutes the arrival orbit family. Transfer learning is used to train a policy that prioritizes small final deviations from the arrival orbit, and reduced total maneuver magnitude over each episode. A first phase of the training process uses a maximum of 1×10^7 steps to uncover a wide array of transfers with a weighting factor in the reward formulation of $K = 0$. The policy is then transferred to a new training process, accounting for a maximum of 5×10^6 steps, and characterized by a weighting factor $K = 1$. The training procedure is empirically demonstrated to generate a policy that can uncover transfers between relatively distant orbits. However, alternative approaches can potentially train policies with similar or improved performances. The time series of the averaged value averaged is reported in Fig. 7.35. The transfer of the policy to an environment with $K = 1$ demonstrates an immediate degradation of the performance of the policy, eventually compensated by the second part of the training.

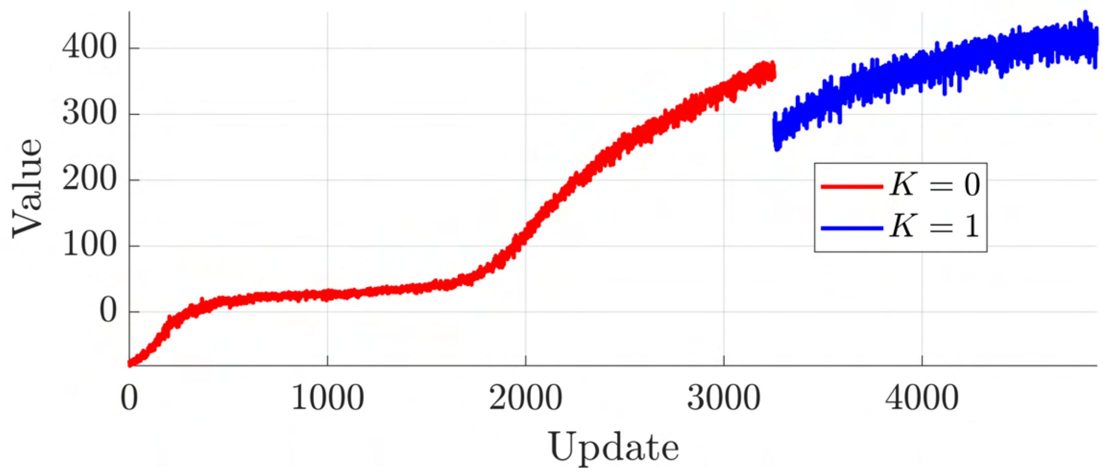

Figure 7.35: Average value per update for the orbit transfer scenario connecting members of the family of southern halo orbits near L_2 to members of the family of southern halo orbits near L_1 in the Earth-Moon system.

The converged policy is leveraged to generate sequences of maneuvers to construct four transfers that depart from members of the L_2 southern halo orbit family and approach members of the L_1 southern halo orbit family in the Earth-Moon CR3BP. The spatial representation in the position space and the xy-projection of the analyzed transfers are reported in the central and bottom row, respectively, of Fig. 7.36. In each frame, the departing and arrival trajectories are colored in blue and red, respectively. Moreover, the families of departing and arrival orbits are visualized using semi-transparent blue and red markers, respectively. The transfers start from the blue circles along the departing orbit and terminate at the red circles along the arrival orbit, following the trajectories depicted in black. Magenta arrows report in the figure the magnitude and direction of the scaled maneuvers. The maneuvers generally present large magnitudes for the first part of the transfers, associated with the departing arc from the initial orbit and the approaching phase to the arrival orbit. The figure demonstrates the capability of the trained policy to generate maneuver sequences to design transfers between members of the considered families of orbits. However, the visualized transfers are associated with relatively large total maneuver magnitude per episode, reported for each solution in the top row of Fig. 7.36. Larger values of weighting factor K in the reward formulation, together with episodes accounting for a larger maximum number of maneuvers, represent two approaches for potentially improving the performance of the policy.

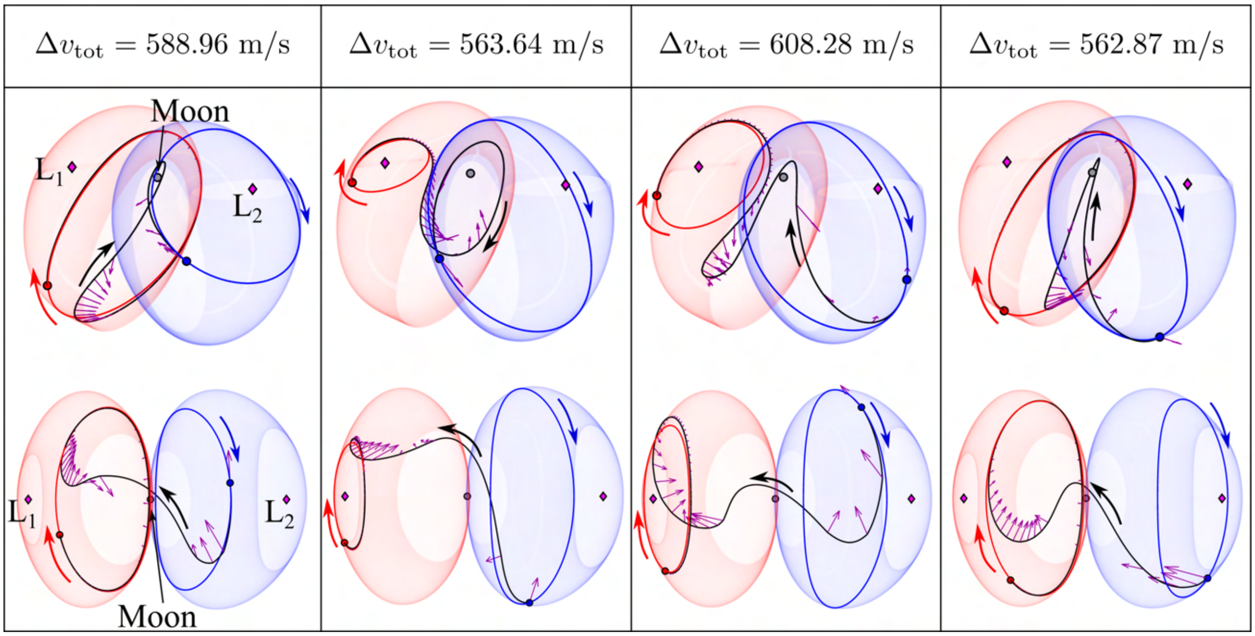

Figure 7.36: Examples of transfers connecting distinct combinations of southern halo orbits near L_2 and southern halo orbits near L_1 in the Earth-Moon system.

7.4 Summary of Contributions

This chapter applies reinforcement learning to the problem of autonomous maneuver generation in multi-body systems. For each type of maneuver, a scenario is initially identified, and converted into a reinforcement learning framework, leveraged to train a policy using a method from the proximal policy optimization family. Eventually, the trained policies are validated to assess the associated performance.

In a first example, a policy is trained for the generation of impulsive station-keeping maneuvers, enabling bounded motion near a quasi-halo trajectory constructed in a point mass ephemeris model, but geometrically resembling a halo orbit in the Sun-Earth CR3BP. A scenario is initially identified after the Nancy Grace Roman Space Telescope, and converted into a reinforcement learning framework. A set of hyperparameters governing the training process is selected leveraging Bayesian optimization on a simplified station-keeping RL scenario. Then, a policy is trained using transfer learning from an environment modeled after the Sun-Earth CR3BP, and validated on a batch of trajectories to evaluate the associated performance. The converged policy is initially tested on a long-term station-keeping scenario modeled with a dynamically perturbed higher-fidelity dynamical model, and ultimately evaluated against a batch of trajectories corrected with traditional constrained optimization. The constructed policy successfully generates sequences of impulsive maneuvers that enable bounded motion with respect to the identified reference quasi-halo trajectory. Moreover, the converged policy generates sequences of impulsive maneuvers with similar performance with respect to the constrained optimization scheme, although requiring significantly reduced computational resources for the generation process.

Policies are also trained for autonomous generation of transfers with impulsive maneuvers between orbits in the Earth-Moon CR3BP. In the first application, a policy is trained to generate sequences of impulsive maneuvers enabling a spacecraft to transfer between two prescribed orbits. A baseline scenario is proposed, leveraging two halo orbits near L_2 in the Earth-Moon CR3BP. The scenario is converted into a reinforcement learning framework, and used to train a policy. Transfer learning is again leveraged to learn a policy that enables the construction of transfers with small final deviation from the arrival orbit, and reduced maneuver magnitude. The policy is then validated on a batch of trajectories, demonstrating the capability to transfer a spacecraft between the prescribed orbits. The reinforcement learning scenario is then leveraged to construct policies that enable transfers between distinct combinations of orbits, differing from the baseline scenario. This first orbit transfer scenario is expanded in a generalized orbit transfer framework, where policies are trained for the generation of sequences of impulsive maneuvers that enable transfer between members of a family of orbits. For this application, the departing and arrival orbits are selected at the beginning of each step, and likely differ between consecutive episodes. The more challenging scenario is initially demonstrated to construct policies that transfer a spacecraft between members of the southern L_2 halo orbits in the Earth-Moon system. The scenario is transformed into a reinforcement learning framework, leveraged to train a policy using transfer learning. The converged policy is validated using a large batch of trajectories, connecting distinct combinations of departing and arrival orbits within the family of southern L_2 halo orbits. The policy does, however, struggle with transfers to selected arrival orbits with a relative low perilune. After evaluation, the system-agnostic reinforcement learning framework is demonstrated to train policies connecting members of the southern halo orbit family near L_1 in the Earth-Moon system, and members of the families of L_2 and L_1 southern halo orbits in the same system. Overall, policies trained in different orbit transfer scenarios demonstrate the capability to construct sequences of impulsive maneuvers enabling transfers of distinct geometries in the Earth-Moon system: these sequences can serve as initial guesses for trajectories in higher-fidelity dynamical models or with continuous control, as well as for rapid investigation of available transfer solutions between distinct orbits in low-fidelity dynamical models.

7.4.1 Scientific Contributions of the Presented Applications

This chapter demonstrates the application of reinforcement learning to constructing policies that can autonomously generate sequences of impulsive maneuvers for trajectory design. Distinct examples in different dynamical systems are introduced, corresponding to a variety of scientific contributions, comprising:

1) Policy training in a point mass ephemeris model: the results presented in Sec. 7.1.4 demonstrate a successful application of reinforcement learning techniques for designing station-keeping maneuvers in a higher-fidelity point mass ephemeris model. The obtained policy successfully generates maneuver sequences that generate bounded motion with respect to an underlying reference trajectory, in the presence of dynamical perturbations.

2) Proximal policy optimization for impulsive maneuver design: a member of the family of proximal policy optimization algorithms is used to construct a policy for rapid station-keeping and orbit transfer impulsive maneuver design. In the presented cases, the policy successfully recovers sequence of maneuvers enabling bounded motion with respect to a reference trajectory, or transfer between distinct combinations of periodic orbits.

3) Transfer learning to reduce complexity of policy training: in the presented examples, transfer learning is leveraged to train policies in complex dynamical models. Specifically, transfer learning is used in the station-keeping maneuver design scenario to accelerate training in higher-fidelity models, while it is used for the orbit transfer scenario to enable the construction of policies that generate a variety of transfers with challenging reward formulations.

4) Comparison with constrained optimization: the policy generated for the station-keeping maneuver design scenario is compared in Sec. 7.1.5 to solutions generated by a constrained optimization schemes. The comparison reveals the two approaches generate similar solutions, approximately verifying similar performances between the investigated approaches.

5) Transfers between members of families of orbits: different policies are constructed in Sec. 7.3 to generate sequences of impulsive maneuvers that enable transfers between members of a family of orbits in the Earth-Moon CR3BP. The approach is subsequently extended to transfer spacecraft between members of distinct families of orbits in the same system.

7.4.2 Value of Reinforcement Learning for Autonomous Maneuvering in Multi-Body Systems

An algorithm from the proximal policy optimization family, shortly referred to as PPO, is used in this chapter to train policies for the autonomous generation of sequences of impulsive maneuvers in different trajectory design scenarios. The leveraged method presents different benefits with respect to a more traditional approaches, including:

1) Reduced computational resources for model evaluation: the trained actor neural network can be used to generate sequences of maneuvers for rapid trajectory design. The actor's model consists of a small number of layers and nodes, that are relatively more efficient to be evaluated with respect to a traditional constrained optimization scheme. Yet, the two approaches generate solutions with similar performance. The reduced computational resources during evaluation might enable future onboard autonomous maneuver design, as well as rapid Monte Carlo investigations.

2) Evaluation of derivatives: the reinforcement learning implementation does not require the definition of first or second order derivatives for the reward formulation. Therefore, RL can leverage discontinuous, or non-differentiable reward formulations.

3) Reduced analyst expertise on the design space: the operator leveraging reinforcement learning techniques for trajectory design does not need complete expertise of the available solution space. Indeed, the RL agent may discover feasible paths that maximize the long-term reward without any instruction from the operator. Moreover, traditional optimization schemes often heavily rely on the construction of a good first guess solution, subsequently refined during the constrained optimization scheme: with a reinforcement learning approach, the operator does not necessitate to provide a first guess solution.

4) Exploration of the design space: early exploration, granted by the stochastic policy used during training, enables to experience various state-action pairs, preventing the policy from honing in on a local optimum. This behavior may not be observed in constrained optimization schemes that only produce local optima.

5) Importance of the long-term result: an RL agent is trained to maximize the return, generally corresponding to a weighted sum of the future rewards.

6) Model independence: the trained policy does not heavily depend on the leveraged model, given the stochastic nature of the leveraged actor and the exploration-exploitation trade-off

during training. Conversely, a variety of traditional constrained optimization approaches rely on a perfect knowledge of the model. In trajectory design, the dynamical environment is never perfectly known, requiring robustness for the trajectory design mechanism. The robustness of a policy for autonomous generation of station-keeping maneuver sequences is demonstrated in Sec. 7.1.4.

These summarized benefits can support future autonomous maneuvering in multi-body systems, that can potentially enable onboard trajectory design with reduced computational resources, and a significantly reduced human effort in the trajectory design process.

7.4.3 Ongoing Challenges in Applying Reinforcement Learning to Autonomous Maneuver Design

Challenges in leveraging techniques from reinforcement learning for training policies for autonomous maneuver sequence generation that require further exploration include:

1) Large computational resources for model training: training a reinforcement learning policy requires a large amount of data, that can be computationally expensive to generate. For example, in the station-keeping scenario modeled in a point mass ephemeris model and described in Sec. 7.1.4, the policies are trained in 72 and 36 hours, respectively, with and without transfer learning on an Intel Core i7-2600K @ 3.40GHz using 6 logical cores. However, training is performed only once, and the same policy can be used to generate different trajectory arcs. Policy training might be significantly accelerated with different methods comprising higher-performance computational machines and transfer learning from environments modeled after lower-fidelity dynamical models.

2) Constraint implementation: the preliminary reinforcement learning algorithm prevents the inclusion of hard and soft constraints in the loss function formulation. Thus, constrains are incorporated in the environment via early stopping criteria and associated penalizing rewards. However, implementing constrains with early stopping criteria can negatively

impact the performance of the trained policy. Constrained MDPs and safe reinforcement learning might offer an alternative approach to incorporating these constraints.

3) Hyperparameters tuning: the performance of the trained policy is significantly impacted by a set of tunable hyperparameters, governing the training process, the loss function, and the designed environment. These hyperparameters are often tuned with a combination of operator expertise and traditional parameters selection available from similar studies. However, each reinforcement learning scenario is likely associated with an optimal set of parameters, potentially generating a policy with superior performances. Therefore, parameters optimization approaches, leveraging for example Bayesian optimization, can be adopted to find an optimal set of hyperparameters. These optimization methods rely on multiple training of reinforcement learning policies, significantly impacting the required computational resources. Heuristics and systematic approaches to rapidly and efficiently select these quantities may be useful.

4) Interpretability: whereas results obtained from constrained optimization can be interpreted by investigating the associated derivatives, an action generated from a reinforcement learning agent, potentially modeled by a nonlinear neural network, might not be easily interpretable by the operator. Moreover, it is often challenging to interpret the impact of multiple sets of hyperparameters on the performances of the trained policies. The lack of interpretability of the obtained results, and the set of hyperparameters governing the policy training, often impact the trust of practitioners on the generated results from the agents. Techniques such as explainable machine learning and manifold learning might be leveraged to enhance the explainability of the generated results.

Chapter 8

Concluding Remarks

Throughout this investigation, techniques from machine learning are leveraged to address specific challenges in trajectory design in multi-body systems. This chapter summarizes the results detailed in this manuscript, and presents potential avenues for future research.

8.1 Summary of Results

Constructing an end-to-end trajectory arc that fulfills mission requirements is often a challenging task that demands large human involvement and computational resources. Traditional approaches to arc construction in multi-body systems leverage techniques from dynamical system theory and constrained optimization. These tools can: aid an analyst to investigate the geometry of existing solutions and patterns in the available design space; help an astrodynamicist generating transfers that require small maneuver magnitude and target motion near existing solutions in low-fidelity models; assist a trajectory designer in constructing maneuver sequences that enable a variety of space missions. These techniques often require large human expertise to investigate the high-dimensional solution spaces typical of spatial trajectories constructed in low- and high-fidelity dynamical models, and generally necessitate large computational resources to generate optimal maneuver sequences in very large design space. Methods from machine learning can be leveraged to aid the astrodynamicist throughout the different phases of the trajectory design process. Techniques from unsupervised learning can serve to generate an autonomous partition of a high-dimensional dataset of trajectories, algorithms from manifold learning can improve the identification of spatial arcs, while methods from reinforcement learning can train policies for rapid generation of sequences of maneuvers in both low- and high-fidelity dynamical models.

In this investigation, techniques from unsupervised learning as clustering, manifold learning, and distributed data mining, are leveraged to design a method for the autonomous partitioning of large datasets of trajectories based on geometrical similarities. This method can aid a human analyst to investigate chaotic and vast solution spaces. In an initial approach, datasets of prograde periapsis are initialized in the Earth region on mutually orthogonal planes and propagated with the Sun-Earth circular restricted three-body problem (CR3BP) to generate a set of arcs. These are then transformed into finite-dimensional feature vectors, reflecting the sequences of encountered apse, and used to populate different datasets to help reduce the required computational resources of the subsequent steps. The clustering algorithm Hierarchical Density-Based Spatial Clustering of Application with Noise (HDBSCAN) is used to partition each dataset in distinct clusters. Then, clusters of distinct partitions that intersect in the phase space are merged into unique solutions, generating three-dimensional groups of trajectories that are geometrically similar. The approach is demonstrated to reduce the visualization burden typical of higher-dimensional Poincaré maps. This method is then expanded to enable cluster correlation across datasets populated with trajectories generated in dynamical models of increasing fidelity, including the CR3BP, the ER3BP, and a point mass ephemeris model. Each dataset leverages an identical set of initial conditions, corresponding to prograde perigees generated at a specific value of energy in the Sun-Earth CR3BP. The datasets are then clustered using HDBSCAN, and correlated across distinct maps using UMAP to assess cluster persistence across distinct dynamical models, and cluster evolution across datasets constructed at different values of the independent variables. These clustering results are also used to demonstrate the governing nature of arcs along the stable hyperbolic manifolds of invariant tori near L_1 and L_2 on the design space near the Earth in the Sun-Earth CR3BP. The presented data-driven methodology produce a variety of benefits, aiding a human analyst throughout the early phases of the trajectory design process where large datasets of trajectories are investigated. Specifically, the presented method allows to: summarize a set of trajectories into a relatively small collection of representative solutions, describing the available arc geometries; focus on specific regions of the dataset, reducing the burden of visualization; visualize regions of existence of each geometry, conferring information on the sensibility of each arc; avoid the definition of problem-dependent analytic separation criteria; visualize cluster persistence across distinct dynamical models for trajectory refinement; assess the evolution of groups of geometrically similar trajectories generated in nonautonomous dynamical models; highlight groups of trajectories that are governed by natural transport mechanisms existing in low-fidelity dynamical models.

Manifold learning is also leveraged in this investigation to aid the identification of natural connections between spatial invariant tori. This method can serve an astrodynamicist to identify natural paths existing between bounded trajectories near periodic orbits, that are often leveraged to motivate natural migration of small bodies in the solar system. Likewise, natural transport mechanisms of invariant tori can be employed for trajectory design to expand the solution space. A method is presented to construct and correct an initial guess of a transfer naturally departing and approaching invariant tori. The approach first identifies two invariant tori that possess nearby stable and unstable manifolds. An approximation of the global hyperbolic manifolds is constructed by sampling initial locations along the tori. These initial conditions are then perturbed along the direction of the locally stable or unstable hyperbolic manifold, and propagated in the identified dynamical model. Multiple intersections of these arcs with a common hyperplane are recorded, and a dataset of crossings is populated. The constructed high-dimensional dataset is then processed with UMAP, to generate a low-dimensional representation of the generated crossings. Stable and unstable manifolds arc are then selected based on proximity of the projected crossing in the embedded space. Multiple revolutions along the distinct tori are appended to the identified stable and unstable manifold arc to construct an initial guess solution for a natural transfer between tori. The solution is then corrected using multiple shooting and constrained optimization, enforcing trajectory continuity at the nodes, and natural departure and arrival from the connected tori. The retrieved single-point solution is then leveraged in a continuation scheme to construct families of geometrically similar transfers that connect families of invariant tori. The devised methodology is demonstrated to construct a variety of natural transfers between invariant tori near distinct mean-motion resonances in the Earth-Moon CR3BP. One of the generated natural transfer is utilized to construct a family of transfer with similar geometries that connect two families of invariant tori near resonance. The devised methodology can serve a trajectory designer to investigate natural transfers solutions between invariant tori, expanding the available design space for mission design. Moreover, the presented method can be leveraged to further analyze transport mechanisms of small bodies naturally transitioning between different resonances in the solar system.

Reinforcement learning is also used to train different policies for autonomous generation of sequences of impulsive maneuvers. Two different scenarios are analyzed to demonstrate the capability of policies trained via reinforcement learning to autonomously generate sequences of impulsive maneuvers. In a first example, a policy is trained to generate sequences of impulsive station-keeping maneuvers near a quasi-halo trajectory in a point mass ephemeris model, minimizing displacement from the reference trajectory and low control effort. The scenario is initially converted into a reinforcement learning environment, and a policy is trained leveraging transfer learning to accelerate the training process in such a computationally expensive dynamical model. The trained policy is validated using a batch of generate trajectories, and tested on a long-term station-keeping example, incorporating dynamical perturbations within the dynamical model. Eventually, the performance of the trained policy is compared with the results of a constrained optimization scheme. This first example demonstrates the capability of reinforcement learning to train policies that generate sequences of impulsive maneuvers for station-keeping: the generated policy exhibits robustness towards dynamical perturbations, and possess a similar performance with respect to constrained optimization scheme. In a second exploratory example, a policy is trained to assess the capability of a reinforcement learning technique to train a policy for autonomous generation of sequences of maneuvers enabling transfer between orbits in the Earth-Moon CR3BP. Initially, a reinforcement learning scenario is designed, and utilized to train a policy. Transfer learning is leveraged to enable training policies that balance small displacement from the arrival orbit and low propellant consumption. The trained policy is validated using a set of trajectories, and for multiple combinations of orbits. Eventually, the scenario is expanded to train a policy that can generate transfers between members of a family of orbits. The trained policy is demonstrated in a variety of examples connecting members of different families of orbits in the Earth-Moon CR3BP. Through different examples, policies trained via reinforcement learning demonstrate the capability of generating sequences of impulsive maneuvers that can enable rapid trajectory design in both low- and high-fidelity dynamical models. Trained policies can significantly aid a trajectory designer for rapid generation of end-to-end arcs that fulfill a variety of mission requirements, without leveraging computationally expensive approaches as constrained optimization. Moreover, for computationally lightweight models, reinforcement learning may enable future autonomous on-board design of maneuvers for a variety of tasks comprising station-keeping and orbit transfers.

8.2 Recommendations for Future Work

A variety of aspects discussed throughout this manuscript can be further expanded for additional development in future research. A list of recommended items is presented, structured into three main parts and corresponding to the main results of this work, including:

Unsupervised learning for higher-dimensional Poincaré maps

1.1) One of the presented examples is used to assess cluster persistence across distinct dynamical models. However, the constructed datasets use a unique set of initial conditions, retrieved in the low-fidelity CR3BP. Additional insights into cluster geometries can be obtained by leveraging datasets of initial conditions generated directly in the used dynamical models, enabling the generation of a larger trajectory dataset that better represents the available solution space. Moreover, the analysis can be expanded to trajectories associated with a wider variety of initial conditions and model parameters.

1.2) The stable hyperbolic manifolds emanated from invariant tori near L_1 and L_2 demonstrate to govern regions of the design space near the Earth in the low-fidelity Sun-Earth CR3BP, for the presented example. Additionally, clusters associated with trajectories naturally escaping the Earth region are identified in higher fidelity models comprising the Sun-Earth ER3BP and a point mass ephemeris model of the Earth and the Sun. Further analysis can focus on investigating dynamical equivalents in higher fidelity models of these governing structures existing in the low-fidelity CR3BP.

Manifold learning for constructing natural transfers between invariant tori

2.1) Recent advancement in astrodynamics indicate the existence of invariant tori in higher-fidelity models, as the bi-circular restricted four-body problem [80, 81]. The method used in this investigation to generate natural transfers between tori in the CR3BP can also be applied, with minor modifications, to verify the existence of natural transport mechanisms between tori existing in higher-fidelity models. This result would enable the construction of transfers that more closely resemble end-to-end trajectories in high-fidelity models.

2.2) The devised methodology can be applied to generate families of transfers between invariant tori that are not analyzed in this investigation. For example, investigating natural arcs approaching solutions at the Earth-Moon L_1 and L_2 might inform trajectory design for future cislunar missions.

2.3) The current implementation of the presented methodology accounts for a constrained optimization scheme to generate single-point solutions. This step is used to numerically correct long arcs that often present close passage in sensible regions of the design space. However, the constrained optimization significantly impacts the required computational resources allocated for correcting the initial guess. Rapid correction mechanisms might be considered to replace the constrained optimization step.

2.4) UMAP is leveraged in this investigation to project the crossings of a set of trajectories arc with a common hyperplane. Different approaches, like variational autoencoders and the parametric UMAP, can also be leveraged to generate parametric mapping. These architectures can be used as generative models, further extending the solution space [88].

Reinforcement learning for autonomous design of impulsive maneuver sequences

3.1) The station-keeping scenario is applied to a specific example, emulating the Nancy Grace Roman Space Telescope framework. The RL scenario may be applied to a variety of reference trajectories in different dynamical models, including trajectories in cislunar space and invariant tori near periodic orbits. Policies converged in environments leveraging nearby reference trajectories can be used to initialize the neural networks, accelerating the training in high-fidelity dynamical models.

3.2) Transfer learning is leveraged in this investigation to accelerate the training of policies generating impulsive station-keeping maneuvers in a point-mass ephemeris model. Transfer learning might also be used to reduce the required training time of policies seeking to generate bounded motion with respect to different reference trajectories in the same, or different, dynamical system.

3.3) For the second example, a reinforcement learning scenario is formulated to train a policy for the generation of sequences of impulsive orbit transfer maneuvers. However, the performance of the trained policy are not compared with other state-of-the-art solutions. A comparison might enable further verification of the leveraged scenario, leading to the design of a policy with improved performances.

3.4) A recent investigation of applied reinforcement learning to station-keeping maneuver design near halo orbits in the Earth-Moon system reveals that off-policy actor-critic formulations can recover performing sequences of impulsive maneuvers [68]. A wider variety of reinforcement learning approaches can similarly be applied to the scenario described in this investigation, and their performance can be compared to investigate their applicability for autonomous maneuver design in chaotic environments.

3.5) In both maneuver design scenarios, boundedness and localized spacecraft motion are enforced by early termination criteria, that negatively impact the learning process of the trained policies. The application of algorithms from safe reinforcement learning allows to incorporate constraints in the objective function formulation, therefore preventing premature episode termination.

Bibliography

[1] 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, West Sussex, United Kingdom, 2011.

[2] I. Elliott, C.J. Sullivan, N. Bosanac, J Stuart, and F. Alibay. Designing low-thrust trajectories for a smallsat mission to sun-earth l5. Journal of Guidance, Control and Dynamics, 43(10):1854–1864, October 2020.

[3] A. Haapala. Trajectory Design in the Spatial Circular Restricted Three-Body Problem Exploiting Higher-Dimensional Poincaré Maps. PhD thesis, Purdue University, West Lafayette, IN, 2014.

[4] K. Oshima, S. Campagnola, C. H. Yam, Y. Kayama, Y. Kawakatsu, N. Ozaki, Q. Verspieren, K. Kakihara, K. Oguri, and R. Funase. EQUULEUS Mission Analysis: Design of the Transfer Phase. In 31st International Symposium on Space Technology and Science, Matsuyama, Japan, June 2017.

[5] W. S. Koon, M. W. Lo, J. E. Marsden, and S. D Ross. Resonance and Capture of Jupiter Comets. Celestial Mechanics and Dynamical Astronomy, 81(27):27 – 38, September 2001.

[6] B.F. Villac and D.J. Scheeres. On the Concept of Periapsis in Hill's Problem. Celestial Mechanics and Dynamical Astronomy, 90:165–178, 2004.

[7] G. Gómez, W.S. Koon, M.W. Lo, J.E. Marsden, J. Masdemont and S.D. Ross. Connecting Orbits and Invariant Manifolds in the Spatial Restricted Three-Body Problem. Nonlinearity, 17:1571–1606, 2004.

[8] N. Nakhjiri and B. F. Villac. Automated Stable Region Generation, Detection, and Representation for Applications to Mission Design. Celestial Mechanics and Dynamical Astronomy, 123(1):63–83, 2015.

[9] B.F. Villac, R.L. Anderson, and A.J. Pini. Computer Aided Ballistic Orbit Classification Around Small Bodies. The Journal of the Astronautical Sciences, 63(3):175–205, 2016.

[10] T.R. Smith and N. Bosanac. Constructing a Set of Motion Primitives in the Circular Restricted Three-Body Problem via Clustering. In AAS/AIAA Astrodynamics Specialist Conference, Portland, ME, August 2019.

[11] T.R. Smith and N. Bosanac. Motion Primitives Summarizing Periodic Orbits and Natural Transport Mechanisms in the Earth-Moon System. In AAS/AIAA Astrodynamics Specialist Conference, Lake Tahoe, CA (Virtual), August 2020.

[12] T. R. Smith and N. Bosanac. Constructing motion primitive sets to summarize periodic orbit families and hyperbolic invariant manifolds in a multi-body systems. Celestial Mechanics and Dynamical Astronomy, 134(7), 2022.

[13] N. Bosanac. Data Mining Approach to Poincaré Maps in Multi-Body Trajectory Design. Journal of Guidance, Control and Dynamics, 43(6):1190 – 1200, June 2020.

[14] R.J.G.B. Campello, D Moulavi, and J. Sander. Density-Based Clustering Based on Hierarchical Density Estimates. Springer, Berlin, Heidelberg, 2013.

[15] L. McInnes, J. Healy, and S. Astels. hdbscan: Hierarchical Density Based Clustering. The Journal of Open Source Software, 2(11), February 2017.

[16] D. Moulavi, P.A. Jaskowiak, R.J.G.B. Campello, A. Zimek, and J Sander. Density-Based Cluster Validation. In Proceedings of the SIAM International Conference on Data Mining, Philadelphia, PA, April 2014.

[17] M. Bendechache, N. Le-Khac, and M. Kechadi. Efficient Large Scale Clustering based on Data Partitioning. In 2016 IEEE International Conference on Data Science and Advanced Analytics, pages 612–621, Montreal, QC, Canada, 2016.

[18] E. Januzaj, H.-P. Kriegel and M. Pfeifle. DBDC: Density Based Distributed Clustering. In: Bertino E. et al. (eds) Advances in Database Technology - EDBT 2004. EDBT 2004. Lecture Notes in Computer Science, 2992:1 – 18, 2004.

[19] M. Woodard, D. Folta, and D. Woodfork. ARTEMIS The First Mission to the Lunar Libration Orbits. International Symposium on Spacecraft Flight Dynamics, Toulouse, France, January 2009.

[20] J Carrico, D. Dichmann, L. Policastri, J. Carrico, T. Craychee, J. Ferreira, M. Intelisano, R. Lebois, M. Loucks, T. Schrift, and R. Sherman. Lunar-Resonant Trajectory Design for the Interstellar Boundary Explorer (IBEX) Extended Mission. Advances in the Astronautical Sciences, 142:771 – 789, December 2011.

[21] D. Dichmann, J. Parker, C. Nickel, and S. Lutz. Trajectory Design for the Transiting Exoplanet Survey Satellite. International Symposium on Spacecraft Flight Dynamics, Laurel, MD, May 2014.

[22] A.A. Wolf and J.C. Smith. Design of the cassini tour trajectory in the saturnian system. Control Engineering Practice, 3(11):1611–1619, 1995.

[23] P.S. Lykawka and T. Mukaia. Resonance sticking in the scattered disk. Icarus, 192:238 – 247, 2017.

[24] M. Vaquero. Spacecraft Transfer Trajectory Design Exploiting Resonant Orbits in Multi-Body Environments. PhD thesis, Purdue University, West Lafayette, IN, 2013.

[25] A.F. Haapala and K.C. Howell. Trajectory Design Strategies Applied to Temporary Comet Capture Including Poincaré Maps and Invariant Manifolds. Celestial Mechanics and Dynamical Astronomy, 116:299 – 323, 2013.

[26] N. Baresi, Z.P. Olikara and D.J. Scheeres. Fully numerical methods for continuing families of quasi- periodic invariant tori in astrodynamics. Journal of Astronautical Sciences, 65(27):157 – 182, 2018.

[27] Á. Jorba. Numerical Computation of the Normal Behavior of Invariant Curves of n-Dimensional Maps. Nonlinearity, 14(5):943 – 976, August 2001.

[28] G. Gómez and J.M. Mondelo. The Dynamics Around the Collinear Equilibrium Points of the RTBP. Physica D, 157(4):283 – 321, October 2001.

[29] Z.P. Olikara and D.J. Scheeres. Numerical method for computing quasi-periodic orbits and their stability in the restricted three-body problem. IAA Conference on Dynamics and Control of Space Systems, Porto, Portugal, March 2012.

[30] Z.P. Olikara. Computation of Quasi-Periodic Tori and Heteroclinic Connections in Astrodynamics Using Collocation Techniques. PhD thesis, University of Colorado Boulder, Boulder, Colorado, 2016.

[31] D. B. Henry and D. J. Scheeres. Quasi-periodic Orbit Transfers via Intersecting Torus Whiskers. In AAS/AIAA Astrodynamics Specialist Conference, Charlotte, NC (Virtual), January 2021.

[32] L. McInnes, J. Healy, and J. Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. ArXiv e-prints, February 2018. arXiv: 1802.03426.

[33] J. Cao, M. Spielmann, X. Qiu, X. Huang, D.M. Ibrahim, A.J. Hill, F. Zhang, S. Mundlos, L. Christiansen, F.J. Steemers, C. Trapnell, and J. Shendure. The Single-Cell Transcriptional Landscape of Mammalian Organogenesis. Nature, 566:496 – 502, 2019.

[34] A. Diaz-Papkovicha, L. Anderson-Trocmé, C. Ben-Eghan and S. Gravel. UMAP Reveals Cryptic Population Structure and Phenotype Heterogeneity in Large Genomic Cohorts. PLos Genetics, 15(11), 2019.

[35] T. Bloch, C. Watt, M. Owens, L. McInnes and A.R. Macneil. Data-Driven Classification of Coronal Hole and Streamer Belt Solar Wind. Solar Physics, 295(41), 2020.

[36] T.A. Pavlak, K.C. Howell. Strategy for Optimal, Long-Term Stationkeeping of Libration Point Orbits in the Earth-Moon System. AIAA/AAS Astrodynamics Specialist Conference, Minneapolis, MN, August 13-16, 2012.

[37] A. Farrés, G. Gómez, J.J. Masdemont, C. Webster and D.C. Folta. The Geometry of Station-keeping Strategies around Libration Point Orbits. 70th International Astronautical Congress, Washington DC, 2019.

[38] N. Bosanacm, C.M. Webster, K.C. Howell and D.C. Folta. Trajectory Design for the Wide Field Infrared Survey Telescope Mission. Journal of Guidance, Control and Dynamics, 42(9):1899 – 1911, September 2019.

[39] L. Breger and J. P. How. Gauss's Variational Equation-Based Dynamics and Control for Formation Flying Spacecraft. Journal of Guidance, Control, and Dynamics, 30(2):437–448, 2007.

[40] G. Gaias and S. D'Amico. Impulsive Maneuvers for Formation Reconfiguration Using Relative Orbital Elements. Journal of Guidance, Control, and Dynamics, 38(6):1036–1049, 2015.

[41] T. Bennett and H. Schaub. Continuous-Time Modeling and Control Using Nonsingular Linearized Relative-Orbit Elements. Journal of Guidance, Control, and Dynamics, 39(12):2605–2614, 2016.

[42] B. G. Marchand and K. C. Howell. Control Strategies for Formation Flight in the Vicinity of the Libration Points. Journal of Guidance, Control and Dynamics, 28(6):1989 – 1911, November - December 2005.

[43] L. Millard and K. Howell. Optimal reconfiguration maneuvers for spacecraft imaging arrays in multi-body regimes. Acta Astronautica, 63(12):1283–1298, 2008.

[44] M. Bando and A. Ichikawa. Formation Flying Along Halo Orbit of Circular-Restricted Three-Body Problem. Journal of Guidance, Control, and Dynamics, 38(1):123–129, 2015.

[45] I. Elliott and N. Bosanac. Geometric Relative Orbital Element Set for Motion Near a Periodic Orbit with Oscillatory Modes. In AAS/AIAA Astrodynamics Specialist Virtual Conference, Lake Tahoe, CA (Virtual), August 2020.

[46] I. Elliott and N. Bosanac. Spacecraft Formation Control Near a Periodic Orbit using Geometric Relative Coordinates. In AAS/AIAA Astrodynamics Specialist Conference, Charlotte, NC (Virtual), January 2021.

[47] J.J.C.M. Bik, P.N.A.M. Visser, and O. Jennrich. Stabilisation of the triangular LISA satellite formation. In 57th International Astronautical Congress, Valencia, Spain, October 2006.

[48] C.P. Escoubet and R. Schmidt. Cluster II: Plasma Measurements in Three Dimensions. Advances in Space Research, 25(7-8):1305–1314, 2000.

[49] J.J. Guzman and A. Edery. Mission design for the MMS tetrahedron formation. In IEEE Aerospace Conference Proceedings, Big Sky, MT, 2004.

[50] L. Plice, A.D. Dono Perez, and S. West. Helioswarm: Swarm Mission Design in High Altitude Orbit for Heliophysics. In AAS/AIAA Astrodynamics Specialist Conference, Portland, ME, August 2019.

[51] J.S. Parker and R.L. Anderson. Low-Energy Lunar Trajectory Design. John Wiley & Sons, Inc., Hoboken, New Jersey, 2014.

[52] T.A. Pavlak. Trajectory Design and Orbit Maintenance Strategies in Multi-body Dynamical Regimes. PhD thesis, School of Aeronautics & Astronautics, Purdue University, West Lafayette, IN, May 2013.

[53] C. J. Sullivan, J. Stuart, R. L. Anderson, and N. Bosanac. Designing Low-Thrust Transfers to High-Inclination Science Orbits via Hybrid Optimization. Journal of Spacecraft and Rockets, 58(5):1 – 13, May 2021.

[54] R.L. Anderson and M.W. Lo. Role of Invariant Manifolds in Low-Thrust Trajectory Design. Journal of Guidance, Control, and Dynamics, 32(6):1921–1930, May 2012.

[55] R.E. Pritchett, E.M. Zimovan and K.C. Howell. Impulsive and Low-Thrust Transfer Design Between Stable and Nearly-Stable Periodic Orbits in the Restricted Problem. In 28th AIAA/AAS Space Flight Mechanics Meeting, 2018.

[56] V. Mnih, K. Kavukcuoglu, D. Silver, A. Rusu, J. Veness, M. Bellemare, A. Graves, M. Riedmiller, A. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. Nature, 518:529–33, 02 2015.

[57] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Harley, T. P. Lillicrap, D. Silver, K. Kavukcuoglu. Asynchronous Methods for Deep Reinforcement Learning. arXiv preprint arXiv:1602.01783v2, 2016.

[58] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, D. Hassabis. Mastering the Game of Go with Deep Neural Networks and Tree Search. Nature, 529:484–489, 2016.

[59] A. Das-Stuart, K.C. Howell nad D.C. Folta. Rapid Trajectory Design in Complex Environments Enabled by Reinforcement Learning and Graph Search Strategies. Acta Astronautica, 171:172–195, 2020.

[60] D. Miller and R. Linares. Low-thrust optimal control via reinforcement learning. 29th AAS/AIAA Space Flight Mechanics Meeting, Ka'anapali, HI, 2019.

[61] C.J. Sullivan and N. Bosanac. Using Multi-Objective Deep Reinforcement Learning to Uncover a Pareto Front in Multi-Body Trajectory Design. AAS/AIAA Astrodynamics Specialist Conference, (Virtual) South Lake Tahoe, CA, August 9-13, 2020.

[62] C.J. Sullivan and N. Bosanac. Using Reinforcement Learning to Design a Low-Thrust Approach into a Periodic Orbit in a Multi-Body System. 30th AIAA/AAS Space Flight Mechanics Meeting, Orlando, FL, 2020.

[63] C.J. Sullivan, N. Bosanac, R.L. Anderson, A. Mashiku, and J.R. Stuart. Exploring Transfers between Earth-Moon Halo Orbits via Multi-Objective Reinforcement Learning. In IEEE Aerospace Conference, (Virtual) March 2021.

[64] N. LaFarge, D. Miller, K.C. Howell and R. Linares. Guidance for Closed-Loop Transfers using Reinforcement Learning with Application to Libration Point Orbits. 30th AIAA/AAS Space Flight Mechanics Meeting, Orlando, FL, 2020.

[65] A. Scorsoglio, R. Furfaro, R. Linares and M. Massari. Actor-Critic Reinforcement Learning Approach to Relative Motion Guidance in Near-Rectilinear Orbit. 29th AAS/AIAA Space Flight Mechanics Meeting, Ka'anapali, HI, 2019.

[66] D. Guzzetti. Reinforcement Learning and Topology of Orbit Manifolds for Stationkeeping of Unstable Symmetric Periodic Orbits. In AAS/AIAA Astrodynamics Specialist Conference, Portland, ME, July 2019.

[67] A.B. Molnar. Hybrid Station-Keeping Controller Design Leveraging Floquet Mode and Reinforcement Learning Approaches. Master's thesis, School of Aeronautics & Astronautics, Purdue University, West Lafayette, IN, December 2020.

[68] N. LaFarge, K.C. Howell, and D. C. Folta. An Autonomous Stationkeeping Strategy for Multi-Body Orbits Leveraging Reinforcement Learning. In AIAA SciTech Forum, San Diego, CA, January 2022.

[69] S Bonasera, I Elliott, C.J. Sullivan, Bosanac N., N. Ahmed, and J.W. McMahon. Designing impulsive station-keeping maneuvers near a sun-earth l2 halo orbit via reinforcement learning. In AAS/AIAA Astrodynamics Specialist Conference, Charlotte, NC (Virtual), January 2021.

[70] V. Szebehely. Theory of Orbits: The Restricted Problem of Three Bodies. Academic Press, London, UK, 1967.

[71] A. Miele. Revisit of the Theorem of Image Trajectories in the Earth-Moon Space. Optimization Theory and Applications, 147(3):483–490, 2010.

[72] S. Campagnola, M.W. Lo and P. Newton. Subregions of Motion and Elliptic Halo Orbits in the Elliptic Restricted Three-Body Problem. Advances in the Astronautical Sciences, 130:1541–1556, 2007.

[73] N. Bosanac. Leveraging Natural Dynamical Structures to Explore Multi-Body Systems. PhD thesis, Purdue University, West Lafayette, IN, 2016.

[74] SPICE Toolkit. `https://naif.jpl.nasa.gov/naif/aboutspice.html`, 2016. Accessed March 2021.

[75] N. Baresi. Spacecraft Formation Flight on Quasi-periodic Invariant Tori. PhD thesis, University of Colorado Boulder, Boulder, Colorado, 2015.

[76] S. Campagnola, B.B. Buffington, Lam T, A.E. Petropoulos, and E. Pellegrini. Tour design techniques for the europa clipper mission. Journal of Guidance, Control, and Dynamics, 42(12):2615–2626, 2019.

[77] C.D. Murray and S.F. Dermott. Solar System Dynamics. Cambridge University Press, Cambridge, 1999.

[78] R.L. Anderson and M.W. Lo. Dynamical Systems Analysis of Planetary Flybys and Approach: Planar Europa Orbiter. Journal of Guidance, Control, and Dynamics, 33(6):1899 – 1912, 2010.

[79] E. Barrabés and G. Gómez. Spatial p-q Resonant Orbits of the RTBP. Celestial Mechanics and Dynamical Astronomy, 84:387 – 407, 2002.

[80] B. P. McCarthy and K. C. Howell. Characterization of Families of Low-Energy Transfers to Cislunar Four-Body Quasi-Periodic Orbits. In AIAA SciTech Forum, San Diego, California, January 3 - 7, 2022.

[81] J.J. Rosales, A. Jorba and M. Jorba. Families of Halo-like invariant tori around L2 in the Earth-Moon Bicircular Problem. Celestial Mechanics and Dynamical Astronom, 133(16), 2021.

[82] B. P. McCarthy and K. C. Howell. Leveraging Quasi-Periodic Orbits for Trajectory Design in Cislunar Space. Astrodynamics, 2021.

[83] F. Verhulst. Nonlinear Differential Equations and Dynamical Systems. Springer-Verlag, Berlin, Heidelberg, 1996.

[84] M. Paskowitz Possner. Orbit Design and Control of Planetary Satellite Or- biters in the Hill 3-Body Problem. PhD thesis, University of Michigan, Ann Arbor, MI, 2007.

[85] S. Bonasera and N. Bosanac. Applications of Clustering to Higher-Dimensional Poincaré Maps in Multi-Body Systems. In 30th AIAA/AAS Space Flight Mechanics Meeting, Orlando, FL, January 2020.

[86] A. Das. Artifical Intelligence Aided Rapid Trajectory Design in Complex Dynamical Environments. PhD thesis, Purdue University, West Lafayette, IN, 2019.

[87] T. Mitchell. Machine Learning. McGraw Hill, 1997.

[88] K.P. Murphy. Probabilistic Machine Learning: An Introduction. MIT Press, 2022.

[89] I. Goodfellow, Y. Bengio and A. Courville. Deep Learning. MIT Press, 2016. `http://www.deeplearningbook.org`.

[90] J. Han and M. Kamber. Data Mining: Concepts and Techniques, Second Edition. Ch.7, Proquest EBook Central: Elsevier Science and Technology, Waltham, MA, 2006.

[91] S. P. Lloyd. Least squares quantization in pcm. IEEE Trans. Information Theory, 28:128 – 137, 1982.

[92] L. Kaufman and P. J. Rousseeuw. Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons, Inc., 1990.

[93] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases. In Proc. 1996 Int. Conf. Knowledge Discovery andData Mining(KDD'96), pages 226–231, Portland, OR, 1996.

[94] M. Ankerst, M. Breunig, H.-P. Kriegel, and J. Sander. OPTICS: Ordering points to identify the clustering structure. In Proc. 1999 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'99), pages 49–60, Philadelphia, PA, 1999.

[95] T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH: An efficient data clustering method for very large databases. In Proc. 1996 ACM-SIGMOD Int. Conf. Management ofData (SIGMOD'96), pages 103–114, Monreal, Canada, 1996.

[96] G. Karypis, E.-H. Han, and V. Kumar. Chameleon: A hierarchical clustering algorithm using dynamic modeling. COMPUTER, 32:68 – 75, 1999.

[97] L. McInnes, J. Healy, and S. Astels. How HDBSCAN works, 2016. Accessed February 2022.

[98] L. Vendramin, R. J. G. B. Campello and E. R. Hruschka. Relative clustering validity criteria: A comparative overview. Statistical Analysis and Data Mining, 3(4):209–235, 2010.

[99] N. Pal and J. Biswas. Cluster validation using graph theoretic concepts. Pattern Recognition, 30(6):847–857, 1997.

[100] M. Halkidi and M. Vazirgiannis. Clustering validity assessment: finding the optimal partitioning of a data set. In Proceedings 2001 IEEE International Conference on Data Mining.

[101] M. Halkidi and M. Vazirgiannis. A Density-based Cluster Validity Approach using Multi-representatives. Pattern Recognition Letters, 29:773–786, 2008.

[102] C.C. Aggarwal and C.K. Reddy. Data Clustering: Algorithms and Applications. Ch. 2.1.2, 3, 9.2, Chapman and Hall CRC, 2018.

[103] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient Estimation of Word Representations in Vector Space. In ICLR, 2013.

[104] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In NIPS, pages 3111–3119, 2013.

[105] J. Pennington, R. Socher and C. Manning. GloVe: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014.

[106] S. Rifai, P. Vincent, X. Muller, X. Glorot and Y. Bengio. Contractive Auto-Encoders: Explicit Invariance During Feature Extraction. In Proceedings of the 28th International Conference on Machine Learning, Bellevue, WA, 2011.

[107] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio and P.A. Manzagol. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. Journal of Machine Learning Research, 11:3371–3408, 2010.

[108] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. ArXiv e-prints, 2013. arXiv:1312.6114.

[109] Y. Bengio, O. Delalleau, N.L. Roux, J.F. Paiement, P. Vincent, and M. Ouimet. Learning eigenfunctions links spectral embedding and kernel PCA. Neural computation, pages 2197–2219], 2004.

[110] J. Tenenbaum, V. de Silva and J. Langford. A global geometric framework for non- linear dimensionality reduction. Science, 290:2319–2323, 2000.

[111] S.T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. Science, 290:2323–2326, 2000.

[112] M. Belkin and P. Niyogi. Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering. Advances in neural information processing systems, (14), 2001.

[113] L. v. d. Maaten and G. Hinton. Visualizing Data using t-SNE. Journal of Machine Learning Research, 9(86):2579–2605, 2008.

[114] T. Sainburg, L. McInnes and T.Q. Gentner. Parametric UMAP: learning embeddings with deep neural networks for representation and semi-supervised learning. ArXiv e-prints, 2020.

[115] L. McInnes, J. Healy, and S. Astels. How UMAP works, 2016. Accessed February 2022.

[116] R.S. Sutton and A.G. Barto. Reinforcement learning: An introduction. MIT press, 2018.

[117] J. Schulman, F. Wolski, P. Dhariwal, A. Radford and O. Klimov. Proximal Policy Optimization Algorithms. arXiv:1707.06347, 2017.

[118] S.P. Singh, T.S. Jaakkola, and M.I. Jordan. Learning without state-estimation in partially observable markovian decision processes. In ICML, 1994.

[119] V.R. Konda and J.N. Tsitsiklis. Actor-critic algorithms. Advances in Neural Information Processing Systems 12: Proceedings of the 1999 Conference, Denver, CO, pages 1008–1014, 2000.

[120] A. Nair, P. Srinivasan, S. Blackwell, C. Alcicek, R. Fearon, A. De Maria, V. Panneershelvam, M. Suleyman, C. Beattie, S. Petersen, S. Legg, V. Mnih, K. Kavukcuoglu, and D. Silver. Massively parallel methods for deep reinforcement learning. In ICML Deep Learning Workshop, 2015.

[121] S. Kakade and J. Langford. Approximately optimal approximate reinforcement learning. ICML, 2:267–274, 2002.

[122] J. Schulman, S. Levine, P. Abbeel, M. Jordan and P. Moritz. Trust Region Policy Optimization. In International Conference on Machine Learning, pages 1889–1897, 2015.

[123] J. Achiam. Spinning Up in Deep Reinforcement Learning. 2018.

[124] J. Schulman, P. Moritz, S. Levine, M. Jordan and P. Abbeel. High-Dimensional Continuous Control Using Generalized Advantage Estimation. arXiv preprint arXiv:1506.02438, 2015.

[125] Z. Zhu, K. Lin and J. Zhou. Transfer Learning in Deep Reinforcement Learning: A Survey. arXiv preprint arXiv:2009.07888v4, 2020.

[126] K. Oguri, K. Oshima, S. Campagnola, K. Kakihara, N. Ozaki, N. Baresi, Y. Kawakatsu, and R. Funase. EQUULEUS Trajectory Design. The Journal of the Astronautical Sciences, 67:950–976, 2020.

[127] Y. Zheng and X. Zhou. Computing with Spatial Trajectories. Ch. 2, Springer New York., 2011.

[128] S. Bonasera and N. Bosanac. Unsupervised Learning to Aid Visualization of Higher-Dimensional Poincaré Maps in Multi-Body Trajectory Design. In 2020 AAS/AIAA Astrodynamics Specialist Conference, Lake Tahoe, CA (Virtual), August 2020.

[129] A. Delshams, M. Gidea and P. Roldan. Arnold's mechanism of diffusion in the spatial circular restricted three-body problem: a semi-analytical argument. Physica D Nonlinear Phenomena, 334(2):29 – 48, September 2016.

[130] S. Bonasera and N. Bosanac. Applying Distributed Data Mining to Higher-Dimensional Poincaré Maps in the Circular Restricted Three-Body Problem. Celestial Mechanics and Dynamical Astronomy (in preparation).

[131] D.C. Davis, S.M. Phillips, and B.P. McCarthy. Trajectory design for Saturnian Ocean Worlds orbiters using multidimensional Poincaré maps. Acta Astronautica, 143:16–28, February 2018.

[132] S. Bonasera and N. Bosanac. Transitions between quasi-periodic orbits near resonances in the circular restricted three-body problem. In AAS/AIAA Astrodynamics Specialist Conference, Lake Tahoe, CA (Virtual), August 2020.

[133] MathWorks. MATLAB, 2021. Natick, MA, USA.

[134] M.T. Young, J.D. Hinkle, R. Kannan and A. Ramanathan. Distributed bayesian optimization of deep reinforcement learning algorithms. Journal of Parallel and Distributed Computing, 139:43–52, May 2020.

[135] H. S. Jomaa, J. Grabocka and L. Schmidt-Thieme. Hyp-RL: Hyperparameter Optimization by Reinforcement Learning. arXiv:1906.11527v1, 2019.

[136] E. Brochu, V.M. Cora and N. de Freitas. A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning. University of British Columbia, Department of Computer Sciences, December 2010.

[137] F. Nogueira. Bayesian Optimization: Open source constrained global optimization tool for Python, 2014. Accessed November 2020.

[138] N. Bosanac, S. Bonasera, C. J. Sullivan, J. McMahon, and N. Ahmed. Reinforcement Learning for Reconfiguration Maneuver Design in Multi-Body Systems. AAS/AIAA Astrodynamics Specialist Virtual Conference, August 2021.

[139] M. Andrychowicz, A. Raichuk, P. Stańczyk, M. Orsini, S. Girgin, R. Marinier, L. Hussenot, M. Geist, O. Pietquin, M. Michalski, S. Gelly and O. Bachem. What Matters in On-Policy Reinforcement Learning? A Large-Scale Empirical Study. Google Research, Brain Team, 2020.

[140] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup and D. Meger. Deep Reinforcement Learning that Matters. In Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, 2018.

[141] E. Zimovan Spreen. Trajectory Design and Targeting for Applications to the Exploration Program in Cislunar Space. PhD thesis, Purdue University, West Lafayette, IN, 2021.

[142] E.M. Zimovan Spreen, K.C. Howell and D.C. Davis. Near Rectilinear Halo Orbits and Nearby Higher-Period Dynamical Structures: Orbital Stability and Resonance Properties. Celestial Mechanics and Dynamical Astronomy, 132(28), June 2020.