ARTIFICIAL INTELLIGENCE AIDED RAPID TRAJECTORY DESIGN IN COMPLEX DYNAMICAL ENVIRONMENTS

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Ashwati Das

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

May 2019

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Hsu Lo Professor Kathleen C Howell, Chair

School of Aeronautics and Astronautics

Professor James M Longuski

School of Aeronautics and Astronautics

Professor Carolin Frueh

School of Aeronautics and Astronautics

Professor David Spencer

School of Aeronautics and Astronautics

Dr. Roby S. Wilson

Jet Propulsion Laboratory, California Institute of Technology

Approved by:

Professor Wayne Chen

Head of the School Graduate Program

“Try not to become a person of success, but rather try to become a person of value.”

-Albert Einstein

This work is dedicated to my family — you are my strength.

ACKNOWLEDGMENTS

Jeff, you have been an incredibly understanding husband, and a pillar of moral and emotional support. Thank you for always believing in me, and for always being up for bouncing ideas with me — your experience and insights have helped me overcome challenging phases in my research. Our conversations have nurtured me into a critical thinker, a better researcher, and have helped me immensely to grow confidence in my abilities. Your humbleness in spite your accomplishments, and your approach to life have taught me very valuable lessons, and inspire me to be a better person every day. I am so glad to have you as my partner, to share my passion for space, and together, make a positive difference in this world.

I am deeply indebted to my family for their patience, love and support over the past several years of my graduate school career. Thank you for giving me wings, and the courage to dream big. Ma and dad, you have worked hard and sacrificed a lot to give me many opportunities — I hope that I have made it all worthwhile, and that I have made you proud! You have instilled strong and good values in me that have been my guide as I have traveled far from you. Appuppa, Ammumma, Amma, Chetta, Monu, and Wasabi, you are my bundle of joy. Your presence in my life has cushioned the journey so far, and your blessings have helped me conquer many challenges. Mom, Papa, Phill and Holly, I have gained a second family in you through Jeff. Thank you for encouraging me through the challenging times, and putting up with me working even during holidays! I'm so excited for the next phase of my life with you all. Thank you to all my family for your unshakeable confidence in me!

Professor Howell, thank you for taking me under your wing, and for offering many opportunities that have helped me set forth on my career. You've nurtured my academic understanding, and our interactions have helped me become an independent and self-driven researcher. I am also grateful for the times when you've challenged

my ideas/implementations, and asked important questions to help me pause and see the bigger picture. I cannot thank you enough for spending numerous hours on editing my papers, and helping me become a more proficient, confident and technically sound writer. I have learnt much from the ardour you have instilled in me for space exploration, and the importance of thoroughness and high standards that you work by. Our personal discussions and group meetings have broadened my understanding of research, academia, space and life in general. I am very grateful to have been accepted to join your research group. I also appreciate all the time you've taken to help write letters of recommendation in support of my career goals, and the time you've taken to help obtain financial support for tuition.

Prof. Spencer, Prof. Frueh, Prof. Longuski and Dr. Roby Wilson, thank you for being on my committee, and for your guidance and suggestions regarding many exciting avenues for further research related to AI-aided design. Prof. Spencer, thank you also for placing your trust in me to moderate the Bill Nye event for a 6000 person audience at Purdue. It was a wonderful opportunity — one that enabled me to grow confidence in my abilities even further; I am truly grateful for your trust. Roby, in addition to interacting with you as a committee member, it has been a pleasure working in your group during the internships at JPL as well. Thank you for the time you've set aside to offer your insights on working at JPL, approaching the internships, and developing the right mindset in graduate school; I have found your advice very timely during my career and consider you a valuable mentor in my life.

I have many friends to thank for their incredible patience and support. Rashmi Shah, Lisa Daniels, and Aioni Nanitsos, you three in particular have stuck by me through all the ups and downs, and kept in touch with me and checked up on me even when I go through periods of radio silence! I thank you from the bottom of my heart for always being there for me. I couldn't have made this journey without your support and love. Also, Angelica Beltrán, I thank you deeply for helping me get situated at Purdue — you took time off your work and life in Chicago to drive me down all the way to Purdue during a snow storm, and spent the whole day with me as I spoke to professors and graduate students, hoping to be accepted to Purdue. You were so sure that I would be accepted, even though I doubted myself — I will remember your kindness forever.

Prof. Dumbacher and Prof. Minton, you both have written me numerous letters of recommendations and have consistently helped me navigate my career. Lessons from taking your classes on space mission holistic design (Prof. Minton) and systems thinking (Prof. Dumbacher), have exposed me to various challenges faced in the industry as well as practical insights on tackling them effectively. Importantly, I look up to your approaches to leadership, and have learnt much from our interactions; I truly appreciate your encouragement and support that have helped me mature my career goals. Thank you for believing in me. Prof. Grant, I would like to thank you immensely for introducing me to the book "Who moved my cheese" by Spencer Johnson. The book's narrative and message have been instrumental for my success during graduate school, and I have no doubt that the book's lessons will continue to be effective for life.

Andrew Chapman, my first ever exposure to industry was in your group at BAE systems. Not only did I learn immensely from this experience, but also felt very valued even though I was an inexperienced intern at the time. I learnt what it means to be a good leader through you — you always made time to listen to everyone. I admire this quality in you very much, and have aspired to such a goal since I left Australia to come to Purdue, and hope to follow in your footsteps to be a technically sound, dependable and loved leader one day. You are a role model for me, and a dear mentor.

Prof. Peter Gibbens, you are the first person who helped me realize my passion and aptitude for dynamics and control. By incorporating many practical elements in your undergraduate curriculum at the University of Sydney (USYD), you made the classes enjoyable, practical and insightful. The hands-on research to incorporate varied flight models in the USYD flight simulator under your guidance really kicked off my understanding of the importance of pursuing research, higher education, and giving back to the community. You are a great teacher and your passion to nurture your students has always inspired me!

Chris Darby, once when I ran into a coding obstacle at CAE, you offered me some wise words of wisdom, "You can always make anything work in software, it's just a matter of finding the right way to program what you want to do". These are the words that have enabled me to overcome many coding challenges in graduate school that have initially seemed insurmountable. Thank you for your mentorship during my time at CAE. Melissa Nadeau, you once told me during our field trip to the Airforce base, "Once you realize your potential, you will be unstoppable". Although you said that to encourage me along as a new hire at CAE, I murmur these words to myself today when things get challenging to remember that you believed in me and pushed me to aim high. Thank you for being my friend and mentor.

I also have many people to thank for enriching and helping develop my technical understanding via my internships at NASA JPL. Jon Sims, Rob Lock, Austin Nicholas, Timothy McElrath, Juan Senent, Anastassios Petropoulos, Damon Landau, Charles Budney, Karl Mitchell, Mark Wallace, Gregory Lantoine, Lanre Sule, and Kaela Martin, you've not only made my time at JPL even more meaningful, but I have also brought back lessons from our interactions to incorporate in my research and graduate school life. Austin Nicholas, I have learnt so much from you during my internships at JPL. You introduced me to AI techniques and blew me away with your passion, focus and ability to create incredible outcomes to help your team. I am very honored to have worked with you.

I respect, admire and appreciate the members of the Multi-Body Dynamics research group at Purdue. You have offered useful technical insights and volunteered many hours to offer suggestions on my conference and defense presentations. You have helped me become more technically sound, and a better presenter. Importantly, I have learnt so much from all of you. Alumni from the group — Davide Guzzetti, Chris Spreen, and Ted Wahl, thank you also for offering your advice and helping me navigate challenging circumstances during my graduate school career! I truly value our friendship. I will miss you all dearly after graduation, and hope that we will stay in touch.

I would also like to thank Dr. DeLaurentis for including me in his multi-disciplinary research effort with NASA Marshall (CAN No. NNM16567212C). This opportunity has opened my eyes to the completely different field of systems engineering than my my primary discipline of trajectory design. Not only was I able to gain exposure to new people and different challenges faced by the industry, I learnt an important lesson of being able to see a topic/problem from different perspectives, and why such diversity in perspectives is important. I truly enjoyed my time working with the team and working together to produce tangible outcomes for NASA. I hope to extend this experience into the next phase of my life as well. Cesare Guariniello, thank you also for all your mentorship and guidance through this collaborative effort. I have loved working with you and learning from you. It is truly amazing that your Ph.D. work has helped grow and support a complete team of graduate students, and helped offer solutions for NASA. Bill O'Neil, Melanie Grande, Liam Durbin, Michael Dai, Chris Brand, Shubham Singh, Steve Zusack, and Shashank Tamaskar, it was such an honor to work with you all. I will remember our meetings, late night and spontaneous working sessions, and adventurous travels fondly. I am so glad to have contributed towards advancing our space industry with you all — a talented, warm, and committed team. Reggie Alexander, thank you very much for working tirelessly to organize financial support from NASA Marshall and including us as part of your team to solve interesting and tough challenges in the industry. I have loved working with you.

I would also like to thank the Purdue AAE Department and NASA (Grant No. NNX17AH89G) for their continued financial support that has enabled me to pursue my doctorate degree. Dave Folta, although we've never met in person, I am grateful for the opportunities you've helped create that have supported my higher education. I hope that the work undertaken in this investigation helps future efforts at NASA. I am also very grateful to the Zonta foundation for their support with the prestigious Amelia Earhart Fellowship, and accepting me into the Zonta family.

Last but not least, and very importantly, I would like to thank the *Purdue Mars* team. I had a wonderful time getting to know you and working with you. Kshitij Mall, Anish Vutukuri, Harith Kolganti, Nicki Liu, Denys Bulikhov, Lara Cackovich, Kyle Alvarez, Cesare Guariniello, and Max Fagin, we managed to create a meaningful impact on campus as a team! I have learnt a lot through all of you, and will cherish our memories forever.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

Table Page

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

SYMBOLS

	Circular Restricted Model
I	inertial frame
R	rotating frame
$\hat{\ }$	caret symbolizing unit vector
ND	nondimensional
$\mathtt{C}$	constant
M	dimensional mass
$\mathbb{N}$	number of bodies
P	particle
$\mathbf{R}$	dimensional position vector in inertial frame
$\mathbf{V}$	dimensional velocity vector in inertial frame
R	dimensional position magnitude in inertial frame
V	dimensional velocity magnitude in inertial frame
$\mathbf{R}^{'}$	rate of change of $\mathbf{R}$ with respect to dimensional time
$\mathbf{V}^{'}$	rate of change of $\mathbf{V}$ with respect to dimensional time
$M^{'}$	rate of change of M with respect to dimensional time
$\tilde{G}$	dimensional universal gravitational constant
G	nondimensional universal gravitational constant
$\hat{\mathrm{X}}-\hat{\mathrm{Y}}-\hat{\mathrm{Z}}$	inertial reference frame coordinates
$\hat{x}-\hat{y}-\hat{z}$	CR3BP rotating frame coordinates
t	dimensional time
B	barycenter
l^*	characteristic length
m^*	characteristic mass
t^*	characteristic time

v^*	characteristic velocity
μ	system mass parameter
$\mathcal{N}$	dimensional mean motion
n	nondimensional mean motion
$\boldsymbol{r}$	nondimensional position vector in CR3BP rotating frame
r	nondimensional position magnitude in CR3BP rotating frame
$\boldsymbol{v}$	nondimensional velocity vector in CR3BP rotating frame
v	nondimensional velocity magnitude in CR3BP rotating frame
τ	nondimensional time
$\mathbb{t}_i$	an epoch associated with a node
ω	angular velocity of frame
$[x, y, z, \dot{x}, \dot{y}, \dot{z}]$	nondimensional states in CR3BP rotating frame
U^*	pseudo-potential term in CR3BP
∇U^*	gradient of pseudo-potential term in CR3BP
$\boldsymbol{\chi}$	nondimensional vector comprising s/c states in rotating frame
$\dot{\boldsymbol{\chi}}$	rate of change in $\boldsymbol{\chi}$ with respect to τ
$\tilde{\boldsymbol{\chi}}_{Ephem}$	dimensional vector comprising of s/c states in ephemeris regime
$\tilde{\boldsymbol{\chi}}'_{Ephem}$	rate of change of $\tilde{\boldsymbol{\chi}}_{Ephem}$ with respect to dimensional time
d	nondimensional relative distance of s/c from primaries
$\boldsymbol{f}(...)$	a function of
$\boldsymbol{g}(...)$	a function of
L	Lagrange Points
θ	angle between velocity vectors
$\breve{\ }$	instantaneous quantity
$\dagger$	representing desired trajectory states
	——————— Spacecraft-related ———————
Δ	change in quantity of interest
$\widetilde{\mathbb{T}}$	dimensional thrust magnitude
$\mathbb{T}$	nondimensional thrust magnitude

$\mathbb{t}$	nondimensional burn duration
$\tilde{\mathbb{t}}$	dimensional burn duration
$\widetilde{\mathbb{P}}$	dimensional power to engine
$\mathbb{P}$	nondimensional power to engine
$\hat{u} = \hat{u}_R$	thrust pointing direction in CR3BP rotating frame
$\hat{\boldsymbol{u}}_I$	thrust pointing direction in inertial frame
$\hat{\mathbf{u}}_I$	thrust pointing direction in J2000 frame
$\widetilde{Isp}$	dimensional specific impulse
Isp	nondimensional specific impulse
$\widetilde{g}_0$	dimensional standard acceleration due to gravity
g_0	nondimensional standard acceleration due to gravity
m_0	initial s/c mass
$\dot{m}$	s/c mass-flow rate
$\mathbb{u}$	s/c controls
$\mathbb{\tau}_0$	nondimensional time along departure orbit
$\mathbb{\tau}_f$	nondimensional time along arrival orbit
J	cost function
$\boldsymbol{\mathbb{\nu}}$	lagrange multipliers
$\mathbb{\eta}$	lagrange multipliers
$\boldsymbol{\mathbb{\lambda}}$	costates
ψ	boundary conditions
$\mathcal{H}$	hamiltonian
ΔV	maneuver magnitude
	Ephemeris Model
q	central body in ephemeris regime
$\mathfrak{A}$	acceleration vectors in ephemeris regime
$\boldsymbol{\mathfrak{r}}$	nondimensional position vector in J2000 frame
$\boldsymbol{\mathfrak{v}}$	nondimensional velocity vector in J2000 frame
$[\mathtt{x}, \mathtt{y}, \mathtt{z}, \dot{\mathtt{x}}, \dot{\mathtt{y}}, \dot{\mathtt{z}}]$	nondimensional states in J2000 frame

$\breve{L}^*$	instantaneous characteristic length
$\breve{T}^*$	instantaneous characteristic time
$\breve{V}^*$	instantaneous characteristic velocity
	——— Coordinate Transformations ———
$\boldsymbol{r}_r$	s/c rotating frame position states with respect to primary
x_p	s/c x-position in rotating frame with respect to primary
$\mathbf{r}_I$	nondimensional position vector in inertial frame
$\mathbf{v}_I$	nondimensional velocity vector in inertial frame
$[\mathrm{x}, \mathrm{y}, \mathrm{z}, \dot{\mathrm{x}}, \dot{\mathrm{y}}, \dot{\mathrm{z}}]$	nondimensional states in inertial frame
C_{R2I}	direction cosine matrix rotating states from CR3BP rotating to inertial frame
C_{I2R}	direction cosine matrix rotating states from inertial to CR3BP rotating frame
$\boldsymbol{\mathfrak{R}}_I$	dimensional position vector in J2000 frame
$\boldsymbol{\mathfrak{V}}_I$	dimensional velocity vector in J2000 frame
$[\mathscr{X}, \mathscr{Y}, \mathscr{Z}]$	dimensional position states in J2000 frame
$[\dot{\mathscr{X}}, \dot{\mathscr{Y}}, \dot{\mathscr{Z}}]$	dimensional velocity states in J2000 frame
$[\mathcal{X}, \mathcal{Y}, \mathcal{Z}]$	dimensional position states in CR3BP frame
$[\dot{\mathcal{X}}, \dot{\mathcal{Y}}, \dot{\mathcal{Z}}]$	dimensional velocity states in CR3BP frame
	——— Numerical Techniques — General ———
$\Diamond$	symbolizing null space
$[\xi, \eta, \varsigma, \dot{\xi}, \dot{\eta}, \dot{\varsigma}]$	linear variational states from reference
U^*_{ij}	2^{nd} partial derivatives of U^*
$\mathbf{A}$	jacobian — partials of EOMs with respect to states
$\exists$	existential quantifier
δ	small perturbation from a given condition
ϵ	small value of a given quantity
$\forall$	for all
$\varkappa$	frequency of out-of-plane oscillations at Lagrange points

λ	eigenvalues of $\mathbf{A}$ matrix
Λ	$= \lambda^2$
β	substitute terms used to solve for λ
$\mathbf{S}$	matrix with columns = eigenvectors of $\mathbf{A}$
$\mathbf{D}$	diagonal matrix of eigenvalues of $\mathbf{A}$
$\mathcal{I}$	total number of discrete eigenvalues of $\mathbf{A}$
$\succ_i$	coefficients of general solution to EOMs
$\boldsymbol{\rho}_i$	eigenvectors of $\mathbf{A}$
$\mathfrak{S}$	matrix of $\boldsymbol{\rho}_i$
$[E^s, E^u, E^c]$	stable, unstable and center subspaces respectively
$[n_s, n_u, n_c]$	dimensions of stable, unstable and center subspaces respectively
$[\boldsymbol{\rho}_s, \boldsymbol{\rho}_u, \boldsymbol{\rho}_c]$	eigenvectors belonging to stable, unstable and center subspaces respectively
$[\mathcal{W}^s_{loc}, \mathcal{W}^u_{loc}, \mathcal{W}^c_{loc}]$	stable, unstable and center local manifolds respectively
$[\mathcal{W}^s, \mathcal{W}^u, \mathcal{W}^c]$	stable, unstable and center global manifolds respectively
$\mathbb{R}$	real domain
	——————— Numerical Techniques — STM ———————
$\boldsymbol{\Phi}$	State Transition Matrix (STM)
$\boldsymbol{X}$	free variable design vector
$\boldsymbol{F}(\boldsymbol{X})$	constraint vector
k	total number of design variables
$\mathbb{m}$	total number of constraints
$\mathfrak{J}$	jacobian — partials of constraints with respect to design variables
$\mathbb{d}$	subscript denoting desired conditions (in states)
ϖ	propagation duration
$\boldsymbol{I}$	identity matrix
$\mathbb{n}$	total number of nodes in discretized numerical corrections process
	——————— Numerical Techniques — Continuation ————
$\mathbb{s}$	step-size in pseudo-arclength continuation scheme

$\mathfrak{N}$	null vector of jacobian $\mathfrak{J}$
$\boldsymbol{G}(\boldsymbol{X})$	appended constraint vector for pseudo-arclength scheme
$\mathscr{J}$	modified jacobian matrix to account for $\boldsymbol{G}(\boldsymbol{X})$ in pseudo-arclength scheme
$\mathscr{T}$	period of orbit
q	number of returns of to a stroboscopic map
	——— Artificial Neural Networks ———
$\mathbb{o}$	number of layers in ANN
$\mathbb{z}$	logit of a neuron
$\mathbb{Z}$	logit of a layer of neurons
l	a layer of neurons in an ANN
$\mathbb{w}$	features — inputs to a neuron for ANN or conditions to be classified for SVM
$\mathbb{w}$	weights — to/from a neuron for ANNs or weights associated with a particular condition to be separated for SVMs
$\mathbb{W}$	weights — to/from a layer of neurons for ANNs or orientation of hyperplanes for SVMs
$\mathbb{b}$	bias — associated with a neuron in an ANN
$\mathbb{B}$	bias — associated with a layer of neurons for ANNs or hyperplane shift for SVMs
$\mathbb{l}$	total number of neurons in a layer l
$\mathbb{h}$	total number of neurons in a layer $l-1$
$\mathbb{K}$	total number of neurons in output layer of ANN
$\mathbb{Y}$	output of a neuron after application of activation function
$\mathbb{C}$	number of classes in classification ANN
$\mathbb{Y}_p$	ANN prediction
$\mathbb{Y}_t$	truth solution
E	error between $\mathbb{Y}_t$ and $\mathbb{Y}_p$
ι	counter for number of classes for softmax function in ANN

$\mathbb{c}$	counter for number of classes for softmax function in ANN
$\jmath$	learning rate hyper-parameter for ANNs
	—————— Support Vector Machines ——————
$\mathfrak{d}$_f	functional margin of separation for SVM
$\mathfrak{d}$_g	geometric margin of separation for SVM
ζ	slack variable for SVM
C	box constraint value for SVM
$\boldsymbol{\mathfrak{u}}$	true class vector for SVMs
κ	power indicating regularization method
$\boldsymbol{\nu}$	lagrange multipliers
N	total number of conditions to classify for SVMs
$\mathbb{y}$	classification output as ±1 for SVMs
$\mathscr{L}$	Lagrangian
$\mathfrak{K}$	kernel function
$\mathfrak{O}$	hyperparameter for kernel function
$\mathfrak{a}$	nonlinear function to facilitate kernel evalution
$\mathbb{S}$	output score for SVM
$\boldsymbol{\mathfrak{P}}$	predicted probability
$\mathscr{A}$	constant aiding in computation of $\boldsymbol{\mathfrak{P}}$
$\mathscr{B}$	constant aiding in computation of $\boldsymbol{\mathfrak{P}}$
	—————— Pathfinding ——————
N	total available nodes in pathfinding graph
n	node in a pathfinding graph
$\mathcal{P}$	path to target
K	cardinality of arcs in path sequence
$\texttt{A}^*$	a pathfinding algorithm
$< S, A, P, R >$	tuple for MDP - States, Action, Probability, Reward
s	current state during pathfinding
a	action taken during pathfinding

s'	next state during pathfinding
a'	action at next state during pathfinding
$\mathscr{R}$	reward received
π	policy
π^*	optimal policy
$\mathbb{V}^\pi$	value function under policy π
$\mathbb{V}^*$	optimal value function
$\mathbb{F}(s,a)$	state-action-value
U	state-action-value representing $\mathbb{F}(s',a')$
γ	discount factor
α	learning rate of pathfinding agent
$\mathbb{H}$	heuristic in pathfinding algorithm
$\mathbb{E}$	function incorporating value update and heuristic functions
$\mathbb{\xi}$	weighting parameters for pathfinding heuristic
$\mathbb{\beta}$	weighting parameters for pathfinding heuristic
$\mathbb{q}$	random value chosen from uniform distribution
p	tradeoff parameter between exploration and exploitation
p_{ss}	steady-state value for p
$TD(\lambda)$	Temporal difference learning parameter
	Framework
δV_n	velocity perturbations at a particular node
$\mathbb{\phi}$	planar velocity angle input for ANN prediction
$\mathbb{\psi}$	spatial velocity angle input for ANN prediction
$\mathbb{v}$	velocity magnitude input for ANN prediction
$\mathbb{M}_C$	classification model
$\mathbb{M}_R$	regression model
$\mathbb{J}$	pathfinding global cost function
W_t	weight associated with time priority in global pathfinding cost function

W_p weight associated with mass priority in global pathfinding cost function

W^m weights employed in pathfinding heuristic function

W^d weights employed in pathfinding heuristic function

W^v weights employed in pathfinding heuristic function

d_n distance in six-dimensional space between current node and target node during pathfinding

$\mathscr{P}$ probability of picking a favorable node from within an AR

ABBREVIATIONS

s/c	Spacecraft
CR3B	Circular Restricted Three Body
CR3BP	Circular Restricted Three Body Problem
EOM	Equations of Motion
CSI	Constant Specific Impulse
VSI	Variable Specific Impulse
s/c	Spacecraft
JC	Jacobi Constant
LT	Low Thrust
RT	Ephemeris Time
H.O.T	Higher Order Term
STM	State Transition Matrix
2PBVP	Two Point Boundary Value Problem
TOF	Time of Flight
NLP	Nonlinear Programming
CoV	Calculus of Variations
RL	Reinforcement Learning
MDP	Markov Decision Process
HARL	Heuristically Accelerated Reinforcement Learning
IG	Initial Guess
TSP	Traveling Salesman Problem
MSE	Mean Square Error
AR	Accessible Region
SCG	Scaled Conjugate Gradient
s.t.	such that

RKHS	Reproducing Kernel Hilbert Space
ODE	Ordinary Differential Equation
LPO	Libration Point Orbit
DFD	Days From Departure

ABSTRACT

Das, Ashwati Ph.D., Purdue University, May 2019. Artificial Intelligence Aided Rapid Trajectory Design in Complex Dynamical Environments. Major Professor: Professor Kathleen C. Howell.

Designing trajectories in dynamically complex environments is challenging and can easily become intractable via solely manual design efforts. Thus, the problem is recast to blend traditional astrodynamics approaches with machine learning to develop a rapid and flexible trajectory design framework. This framework incorporates knowledge of the spacecraft performance specifications via the computation of *Accessible Regions* (ARs) that accommodate specific spacecraft acceleration levels for varied mission scenarios in a complex multi-body dynamical regime. Specifically, pathfinding agents, via Heuristically Accelerated Reinforcement Learning (HARL) and Dijkstra's algorithms, engage in a multi-dimensional combinatorial search to sequence advantageous natural states emerging from the ARs to construct initial guesses for end-to-end transfers. These alternative techniques incorporate various design considerations, for example, prioritizing computational time versus the pursuit of globally optimal solutions to meet multi-objective mission goals. The initial guesses constructed by pathfinding agents then leverage traditional numerical corrections processes to deliver continuous transport of a spacecraft from departure to destination. Solutions computed in the medium-fidelity Circular Restricted Three Body (CR3BP) model are then transitioned to a higher-fidelity ephemeris regime where the impact of time-dependent gravitational influences from multiple bodies is also explored.

A broad trade-space arises in this investigation in large part due to the rich and diverse dynamical flows available in the CR3BP. These dynamical pathways included in the search space via: (i) a pre-discretized database of known periodic orbit families; (ii) flow-models of these families of orbits/arcs 'trained' via the supervised learning algorithms Artificial Neural Networks (ANNs) and Support Vector Machines (SVMs); and, finally (iii) a free-form search that permits selection of both chaotic and ordered motion. All three approaches deliver variety in the constructed transfer paths. The first two options offer increased control over the nature of the transfer geometry while the free-form approach eliminates the need for *a priori* knowledge about available flows in the dynamical environment. The design framework enables varied transfer scenarios including orbit-orbit transport, s/c recovery during contingency events, and rendezvous with a pre-positioned object at an arrival orbit. Realistic mission considerations such as altitude constraints with respect to a primary are also incorporated.

1. INTRODUCTION

Recent advancements in technological capabilities for space exploration offer opportunities to reach a wide array of destinations, from the Moon to the asteroid belt to the outer planets. Such endeavors demand effective mission design strategies that balance diverse objectives and constraints to ensure mission success. Challenges such as low engine acceleration levels, uncertainty in the deployment status for secondary payload missions, and re-design for contingency scenarios may also need to be addressed based upon the mission concept. Hence, a rapid and flexible trajectory design framework, one that offers the exploration of broad trade spaces, is valuable to address mission design scenarios of increasing complexity. Such a capability is particularly beneficial in the near term, i.e., to support an efficient cis-lunar transportation architecture that also supports the emergence of new mission concepts beyond the Earth-Moon neighborhood. For example, near term cis-lunar activities involving robotic exploration and a revival of the human exploration program (e.g., via the Deep Space Gateway [1] concept), demands the flexibility of diverse routes through space for differing cargo and crew transport systems.

Transfer design in a planet-moon neighborhood benefits greatly from a multi-body simulation environment incorporating the gravities of both the planet and moon simultaneously during the preliminary design phase. Thus, the analysis is conducted within the context of the Circular Restricted Three Body Problem (CR3BP). A thorough understanding of the fundamental behaviors and structures in the three body system is facilitated via Dynamical Systems Theory (DST) to enable a well-informed design strategy. The CR3BP offers a multitude of natural solutions — including periodic and quasi-periodic orbits in the vicinity of the primaries and the libration points as well as manifold behaviors that enable flow throughout the region. Furthermore, exploitation of such natural flow in a dynamical system can reduce propellant consumption and enable otherwise unavailable mission concepts. Leveraging natural motion is especially attractive for secondary payloads that often possess acceleration levels at least an order of magnitude lower than the natural acceleration levels in the system of interest. For example, consider the low-thrust acceleration capability of a typical cubesat, on the order of $\approx 8.6e-5\ m/s^2$, to the acceleration due to the Moon's gravity that increases to larger values ($> \mathcal{O}(1e-3\ m/s^2)$) as a vehicle approaches the lunar vicinity. This natural gravitational force of the Moon can be overcome with s/c acceleration levels that are several orders of magnitude greater than the cubesat capabilities. Alternatively, knowledge of the natural motion can also be combined with the lower cubesat acceleration levels to pursue the same overall objectives.

A chaotic multi-body regime, along with increased mission complexity, is a challenging combination for rapid design via solely manual strategies. Also, traditional basin-restrictive optimization techniques constrain potential alternatives. Thus, a computationally efficient and generalizable design approach is desirable, one that reveals multiple trajectory options to meet unique requirements over a broad range of mission types, including both chemical and low-thrust propulsion systems. Low-thrust transfer designs and thruster control histories in particular tend to be non-intuitive and time-intensive for design within an infinitely large design space. So, this investigation blends Artificial Intelligence (AI) techniques with traditional trajectory design approaches to automate exploration of the design space, to identify potentially productive links in a transfer sequence, and to exploit techniques from combinatorics to forge end-to-end trajectories for path-planning. In particular, the focus of the design framework development is the exploration of a multitude of transfer sequences and to deliver only the top options for further design considerations.

The challenge of searching an infinitely large trade-space is mitigated by a combination of approaches within the design framework. The components of this framework are formalized as: (I) Accessible Region (AR) computation — determine the *reach* of the spacecraft (s/c) within a finite time-horizon in a multi-dimensional and infinitely large configuration space, (II) Natural conditions — render natural conditions within these ARs through three approaches: free-form search, flow-models via supervised learning, or an *a priori* discretized database of natural states, (III) Automated pathfinding — exploit artificial intelligence and machine learning techniques to determine the transport sequence from one AR to another, solving for an efficient end-to-end path in an automated manner, and (IV) Convergence/optimization — converge and optimize the selected transport sequence by traditional numerical strategies to construct a continuous solution for a specified engine model.

The knowledge of spacecraft acceleration capabilities are exploited to evaluate an AR at any given instant in time within a multi-dimensional design space. Such a concept pares down infinite transfer possibilities to those only relevant for the spacecraft and mission parameters under consideration. In this investigation, the natural states are made available within a spacecraft AR via various approaches. The *a priori* discretized database and flow-models of known periodic orbit motion, computed by Support Vector Machines (SVMs) and Artificial Neural Networks (ANNs), offer increased control over the geometric transfer profile of the s/c. An alternative 'free-form' formulation enables insights into the transfer scenario without a requirement for prior knowledge of known natural motion within the complex dynamical regime. The natural states within these ARs are then sequenced by pathfinding software agents guided by strategically constructed global cost functions to deliver end-to-end transfer paths. Comparisons between the outcomes and performance of both an 'exact' graph-search technique, Dijkstra's algorithm, and a Heuristically Accelerated Reinforcement Learning (HARL) approach add insight to execute pathfinding and transfer a s/c from departure to destination. The automated initial guess generation process thus delivers a sequence of discontinuous arcs to satisfy one focus of the investigation — rapid generation of preliminary design solutions. However, for the solution to be implementable in a real mission scenario, traditional numerical strategies are leveraged to eliminate the discontinuities, to establish the feasibility of the continuous solution, and to further tune its optimality. The initial analysis is formulated using the assumptions of the Circular Restricted Three Body Problem (CR3BP) dynamical model. These medium-fidelity solutions are also transitioned to a point-mass ephemeris model to assess their feasibility when subject to higher-fidelity dynamical behavior.

In brief, the overarching goal of this investigation is the development of a design framework to rapidly deliver a broad range of transfer initial guesses and, thus, enable complex mission scenarios. The current design approaches are evolved by integrating dynamical systems theory, artificial intelligence and machine learning techniques, as well as numerical methods, including hybrid optimization, to expose pathways to the destination while adhering to mission requirements and constraints. The framework is flexible in that it can accommodate various design considerations, spacecraft specifications, mission constraints (e.g., altitude with respect to a primary), uncertainties (e.g., deployment states), and contingency planning efforts.

1.1 Previous Contributions

The knowledge from many prior efforts form an integral part of the current investigation. A number of significant efforts are highlighted as follows.

1.1.1 A Brief History of the CR3BP

In 1687, Sir Isaac Newton offered significant mathematical insight into the mutual dynamical interactions between multiple bodies in his book *Philosophiæ Naturalis Principia Mathematica* [2]. However, when more than two bodies are involved, no convenient closed form solution exists. In 1772, Leonhard Euler suggested simplifying assumptions when attempting to explain the motion of the Moon under the influence of the gravity fields of the Earth and the Sun. He introduced the concept of observations from the perspective of a synodic frame. Further exploration of the underlying mathematics associated with this rotating coordinate frame in the Three Body Problem (3BP) shortly led to many new discoveries. In the same year, Joseph Louis Lagrange determined five equilibrium points in the synodic frame, where these enable many unique mission applications today. Around 60 years later, Carl Gustav Jacob Jacobi published the observation of an integral of motion in the sidereal frame which was later also reformulated in the synodic frame [3, 4]. This physical constant established a link between the energy of a spacecraft along its trajectory and a set of inaccessible regions in the CR3BP configuration space, where their boundaries are denoted zero-velocity surfaces (ZVS) [4]. Such structures were employed by George William Hill (1878) in concluding that the Moon's range of motion was bounded relative to the Earth. In addition to proving that Jacobi's constant is the only available integral of the motion in the CR3BP, Poincaré (1899) suggested the existence of periodic orbits in the CR3BP system and introduced the concept that 'surfaces of sections' can offer insight concerning their evolution [5]. Based on the work of Poincaré, further developments were accomplished by Birkhoff (1915), Moulton (1920s), Wintner (1941), as well as Kolmogorov and Siegel in the 1950s [3, 6]. As the digital age trickled into the space arena in the 1960s, the pace of progress in this area of research quickened, and its breadth broadened. Today, the CR3BP framework is exploited to design trajectories for impulsive and continuous thrust systems alike, for scenarios ranging from missions to the Moon and beyond.

1.1.2 Aspects of Multi-Body Trajectory Design

Trajectory design in a complex dynamical environment benefits from the ability to quantify and predict the natural motion as well as leverage natural structures where beneficial to the mission objectives. Thus, many prior works in the areas of Dynamical Systems Theory (DST), numerical corrections, and optimization offer valuable insights. Relatively new efforts blending the field of artificial intelligence and traditional trajectory design have also contributed to an improved understanding of the challenges in traditional design capabilities and offer avenues for mitigating them as well.

Natural Structures

The construction of the design framework commences with a fundamental understanding of the natural dynamics in the Earth-Moon system and the option to leverage existing natural structures to enable mission scenarios. Dynamical Systems Theory (DST) offers insights into periodic motion that enables bounded behavior within a complex dynamical system, as well as hyperbolic manifolds that facilitate transport within this system. Such structures have proven to be key enablers for many missions such as *International Sun-Earth Explorer-3* [7], Genesis [8–10], ARTEMIS [11], and WMAP [12]. Recently, Distant Retrograde Orbits (DROs) [13], members of the Halo family [14], and resonant motion are all gaining increased attention to enable future infrastructure capabilities and support human exploration activities in the cislunar region [15]. The application of invariant tori that offer boundedness via quasi-periodic motion is also useful [16,17]. Folta et al. [18] offer an interactive catalog of orbit families for applications in multi-body regimes, where the advantages of characterizing trajectory parameters to identify potential parking and transfer options are readily apparent. Guzzetti et al. [19] demonstrate the ability to incorporate operational constraints into the design process while interacting with such a catalog.

General Trajectory Design Approaches

A summary of various lunar transfer approaches and their applications is compiled by Parker and Anderson [20]. The time critical *Apollo* missions, carrying humans to the Moon, used direct transfer options comprised solely of chemical engine ΔV maneuvers with outbound durations as short as 3 days [20]. The *SMART-1* mission took advantage of higher ISP levels offered by an electric propulsion engine to pursue a slower but propellant-efficient spiral escape from the Earth towards the Moon [21]. Mingotti and Topputo [22] generate similar low-thrust engine spiral trajectories to demonstrate a variety of transfer scenarios to the Moon. Although various low-thrust systems are under development, the two classifications considered in this investigation include ion engines that possess relatively Constant Specific Impulse (CSI) values consistent with the Dawn mission [23], as well as the VASIMIR engine where the specific impulses can vary up to an order of magnitude during operation [24]. Natural arcs can also serve as a basis for transfer design, enabling innovative concepts even in low-thrust regimes [25]. These trajectory options are often termed low-energy transfers. Time-invariant flow structures in the CR3BP and their ability to enable transport in a complex environment were investigated in depth by Conley [26], Koon et al. [27] and Gómez et al. [28]. Topputo et al. [29] generate initial conditions for transit orbits between the Earth and the Moon by exploiting the stability characteristics of the L_1 libration point and linking its end-points to departure and arrival orbits at the primaries using Lambert arcs.

Additional investigations demonstrate the ability to link arcs belonging to various natural structures (orbits and their manifolds) via chemical impulses or low-thrust schemes. One approach to constructing these hybrid powered/low-energy trajectories is eliminating discontinuities between natural arcs at a selected hyperplane crossings with impulsive maneuvers [30–33] or with thrusting arcs [34–39]. This Poincaré mapping technique captures signatures of the arcs of interest on reduced-dimension surfaces of section such as an $x - y$ plane or an apse map [40]; higher dimensional information is also overlaid as required [41]. Assumptions concerning a steering law, leveraging forward and backward propagations, and/or optimal control theory are common methods to isolate regions on the hypersurface that are attainable given a set of engine performance capabilities. An alternative to the Poincaré mapping technique is demonstrated by the 'orbit chaining' approach in Pritchett et al. [42] that 'unspools' a stack of natural and/or powered arcs [43] of interest via numerical techniques such as collocation to construct an end-to-end trajectory.

Numerical Techniques

Numerical shooting schemes aid in eliminating the discontinuities in position and velocity that are inevitably introduced during preliminary design approaches; these corrections schemes also enable path and terminal constraints to be incorporated into the transfer problem. Multiple shooting schemes, in particular, aid in managing the sensitivities associated with long trajectories, even in highly nonlinear and chaotic dynamical regimes [44]. Furthermore, such numerical corrections schemes are intrinsic to the transition of medium-fidelity CR3BP solutions to higher-fidelity analysis in an ephemeris model. These correction algorithms are typically incorporated within a larger optimization problem that seeks to meet mission objectives like the minimization of the transfer duration and/or propellant consumption. Such optimization problems can be reformulated it as a Two Point Boundary Value Problem (2PBVP) and solved via *indirect*, *direct*, or *hybrid* optimization schemes. Russell [45] details the advantage of a low-dimensional search space in indirect optimization but also notes its sensitivity and inflexibility to constraint variations. Such challenges are mitigated via direct optimization schemes, but at the cost of increased dimensionality. Stuart and Howell [46] as well as Howell and Ozimek [47] employ a hybrid optimization scheme to tap the strengths of both the indirect and direct approaches. In this approach, indirect methods employing the Calculus of Variations (CoV) and Euler-Lagrange Theory (E-LT) are exploited to derive the optimal control history without *a priori* assumptions on its form, with the terminal cost being improved via direct optimization schemes. Many readily available packages such as Sequential Quadratic Programming (SQP) or MATLAB©'s `fmincon` exist to support direct optimization. Direct transcription techniques have also proven effective in addressing challenges associated with constrained basins of convergence and the sensitivities associated with indirect optimization methods, especially for low-thrust solutions [42, 48].

Adoption of Artificial Intelligence Techniques in Trajectory Design

Access to a wide range of natural arcs, coupled with powered arcs resulting from a broad array of thrust capabilities, results in an infinitely large trade-space to satisfy mission constraints. This design challenge quickly becomes intractable when addressed solely via manual search methods. Thus, some recent investigations address the problem via combinatorial optimization techniques. In one approach, traditional numerical processes are employed to construct an initial guess database comprised of locally optimal solutions. Then, well known graph-search and machine learning methods are exploited to solve a multi-objective problem by examining combinations within the database to produce a global or nearly-global optimum. Conclusions from Radice and Olmo [49], Ceriotti and Vasile [50], Stuart and Howell [51], as well as Furfaro and Linares [52] demonstrate the potential of heuristic methods to be effective in various dynamical regimes and in uncovering local optima that may have otherwise remained unknown.

Approaches employing genetic algorithms [53, 54] and Artificial Neural Networks (ANNs) to improve initial guesses have also proven beneficial. Witsberger and Longuski [55] train Artificial Neural Networks (ANNs) that incorporate alternative transfer options from a genetic algorithm during the training phase to learn to develop interplanetary transfer solutions. De Smet et al. [56] demonstrate the ability of ANNs to learn function approximations and, thus, reduce the number of propagations required to conduct a broad transfer scenario study in the Martian system.

An alternative strategy for transfer design constructs initial guesses for these paths via Artificial Intelligence (AI) techniques, then subjects them to a numerical corrections/optimization process; this approach is the focus of the current investigation. Tsirogiannis [57], as well as Trumbauer and Villac [58], generate impulsive transfer options by constructing a framework of pre-computed natural arcs and employ graph search methods to evaluate the links. Selected nodes then serve as waypoints in a complex dynamical regime. Simplifying assumptions in a two-body model are exploited by Parrish [59] to employ heuristics in solving for consistent low-thrust solutions while, Miller and Linares [60] leverage Reinforcement Learning (RL) techniques to determine control laws to transfer a low-thrust spacecraft in a multi-body regime. The analysis in Das-Stuart et al. [61,62] combines the pathfinding and pattern recognition capabilities of machine learning strategies with knowledge of natural flows in the CR3BP to construct end-to-end transport options; this AI-aided trajectory design methodology is the focus of the analysis in this investigation.

1.1.3 Current Work

The current investigation strives to establish a framework for automated trajectory initial guess generation within a multi-body regime via pathfinding techniques, that is then optimized via traditional numerical methods. The foundation is developed for a process that alleviates the challenges associated with thrust law construction, the sequencing of thrust and coast arcs, and the limited solution options resulting from narrow basins of convergence when implementing traditional numerical techniques. Approximating low-thrust segments via a sequence of impulsive maneuvers is effective for preliminary mission design [63] and aids in this investigation. Both deterministic processes as well as stochastic techniques exploiting software ‘agents’ enable the desired broad search capability [61,62].

The organization of the current investigation is as follows:

- Chapter 2:

  The Circular Restricted Three Body Problem (CR3BP) forms the natural dynamical foundation for the analysis and results in this investigation. Thus, the assumptions and formulations, including the derivation of the equations of motion, are detailed. Furthermore, particular equilibrium solutions to these equations of motion (Lagrange points), their significance, and the implications of the only known integral of motion for the dynamical regime are presented. The equations of motion in the CR3BP are augmented to incorporate the effects of engine acceleration terms and are also modified to accommodate the effects of time-varying gravitational influences from multiple bodies on the spacecraft in an ephemeris regime. The transformations to transition between the rotating and different inertial frames are also summarized.

- Chapter 3:

  The process of determining the stability of solutions in the CR3BP and exploiting this knowledge to isolate certain types of behavior such as hyperbolic or periodic motion is introduced. The general approach to numerical corrections is also outlined, including the derivation of the State Transition Matrix (STM) to quantify the sensitivity of a transfer path to variations in the initial conditions. Numerical continuation schemes such as natural and pseudo-arclength continuation to generate *families* of periodic orbits are intrinsic to this investigation and are, thus, discussed. Various optimization schemes to address VSI, CSI, and chemical transfer scenarios, and the steps required to transition a medium-fidelity CR3BP solution to a higher-fidelity ephemeris routine, are also explored.

- Chapter 4:

  Various branches of AI are incorporated into this investigation to address different goals; this chapter details the theoretical assumptions and derivations associated with the different AI algorithms. Both Artificial Neural Networks (ANNs) and Support Vector Machines (SVMs) are introduced for *supervised* learning tasks that include pattern recognition and regression; the various architecture choices associated with their implementation are discussed. Pathfinding in this investigation is accomplished via Dijkstra's algorithm (an exact method) and Reinforcement Learning (a branch of machine learning). The fundamental differences in the implementation of these exact and heuristic paradigms as well as the associated computational costs and performance expectations are also identified.

- Chapter 5:

  The components of the design framework and the blending of machine learning techniques with traditional astrodynamics approaches to deliver initial guesses for transfers are compiled in this chapter. The notion of Accessible Regions (ARs) are explored in this investigation, along with the supporting assumptions. Three approaches to exploit transport via natural states in the ARs are outlined, namely: (i) *a priori* discretization of natural periodic orbit families, (ii) family models generated via supervised learning and, (iii) free-form. The specific implementation nuances of the pathfinding algorithms (Dijkstra's and Reinforcement Learning) for this investigation are highlighted with the details of transforming the transfer initial guess to a continuous final solution also presented.

- Chapter 6:

  A variety of transfer scenarios in the CR3BP are explored to demonstrate the ability of the design framework to successfully generate initial guesses. The examples are developed to highlight the ability to address various capabilities such as prioritizing transfer duration versus propellant consumption, customizing transfer routes based on spacecraft specifications, exploring different transfer geometries to evaluate alternatives, and re-designing transfers to address contingency scenarios. Examples are also offered to demonstrate the outcomes of transitioning the initial guess from the pathfinding process to a continuous solution in the CR3BP and then transitioning these medium-fidelity solutions to a higher-fidelity ephemeris regime.

- Chapter 7:

  A summary of the design approaches that exploit artificial intelligence capabilities to aid in trajectory design and the conclusions from these efforts are summarized. Potential avenues to advance the topic of this investigation are also offered.

2. SYSTEM MODELS

Dynamical models are key mathematical representations of the gravity forces that influence spacecraft motion. Assumptions in developing the gravitational model strongly influence the type of solutions that emerge during the trajectory design process. Although the model for gravitational forces in an ephemeris regime reasonably accurately capture the motion of a body under the influence of other celestial bodies, there exists no closed-form solutions for such a formulation. This challenge is addressed via numerical propagations, but is computationally intensive. Furthermore, depending on the scenario, analytical approximations for these solutions that are acquired by simpler models (e.g., two-body models) may not persist in the presence of additional gravitational bodies. Thus, the Circular Restricted Problem (CR3BP) is introduced — to balance solution accuracy and computational effort. Spacecraft performance characteristics are also incorporated to characterize its motion in a complex dynamical regime. The advantages and dynamical insights from exploiting a rotating frame formulation, as well as the ability to transition between the different dynamical models of interest, are also explored.

2.1 The $\mathbb{N}$-body Problem

In 1687, Newton published the law of gravitational attraction that influences the planetary motion in his book *Philosophiæ Naturalis Principia Mathematica* [2]. These formulations capture the dynamics associated with the motion of celestial bodies in addition to their kinematical representations that Johannes Kepler highlighted in the 17th century. A vector representation of Newton's 2^{nd} law of gravitation is written:

$$M_i\mathbf{R}_i^{''} = -\tilde{G}\sum_{\substack{j=1\\ j\neq i}}^{\mathbb{N}} \frac{M_i M_j}{\mathrm{R}_{ji}^3}\mathbf{R}_{ji} \tag{2.1}$$

where, for a system of $\mathbb{N}$ bodies, the differential equation aims to model the motion of a particle P_i with respect to an inertially fixed base point as the particle is influenced by the gravitational forces from $\mathbb{N}-1$ point masses, P_j. In Eqn. (2.1), M_i and M_j represent the dimensional masses of the particles P_i and P_j, respectively. The term $\tilde{G}$ reflects the universal gravitational constant, $\mathbf{R}_{ji}$ is the relative dimensional position vector of body i with respect to body j, ($\mathbf{R}_{ji} = \mathbf{R}_i - \mathbf{R}_j$), and $\mathbf{R}_i^{''}$ is the 2^{nd} derivative of the position of the object P_i with respect to dimensional time, t. Note that bold identifies vector quantities. The scalar values represent the magnitudes associated with vector quantities - e.g., R_{ji} is the magnitude of the vector $\mathbf{R}_{ji}$. The spatial $\mathbb{N}$-body problem is illustrated in Fig. 2.1 as viewed in an inertial frame; axes $\hat{\mathrm{X}}-\hat{\mathrm{Y}}-\hat{\mathrm{Z}}$ define mutually perpendicular directions. The carat $(\hat{\ })$ signifies a unit vector.

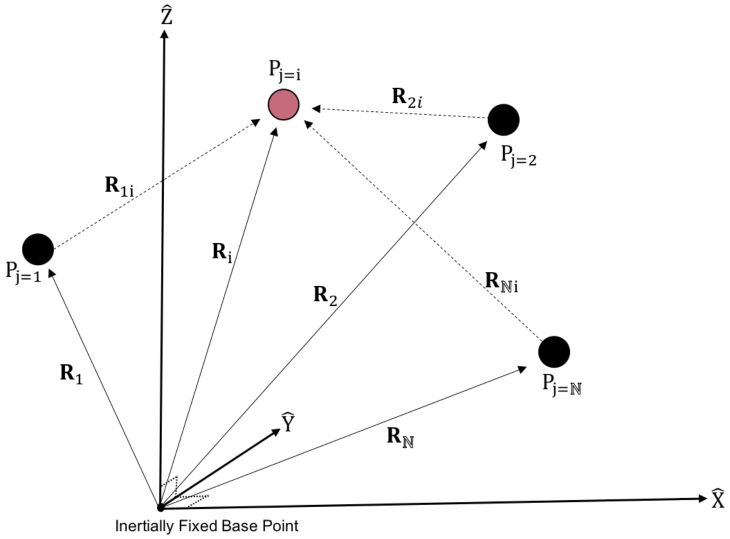


Figure 2.1: Particle P_i under the gravitational influence of $\mathbb{N}-1$ point masses, P_j

The complete mathematical model in the $\mathbb{N}$-body problem is represented via the second-order vector differential equations in Eqn. (2.1). An analytical solution for the $3\mathbb{N}$ second order scalar equations associated with Eqn. (2.1) analytically requires $6\mathbb{N}$ integrals of the motion. Only 10 integrals of motion are known: six from the equations associated with the conservation of linear momentum, three from angular momentum, and one from the conservation of energy. Even for a system of two particles ($\mathbb{N} = 2$), $6\mathbb{N} = 12$ integrals are required, greater than the total known 10 integrals. However, re-writing the equations representing the motion of the two bodies in relative terms enables computation of the well known conic sections yielding Keplerian elements; the relative formulation facilitates these solutions as the system is reduced to six 1^{st} order scalar differential equations that are analytically solvable. However, no closed form solution exists for $\mathbb{N} > 2$ scenarios, as the known 10 integrals fall short of the 18 integrals of motion (6×3) required when adding even one additional particle, i.e., $\mathbb{N} = 3$. So, in general, systems of three or more particles require numerical techniques to explore the mutual interactions between the bodies and their subsequent behavior.

2.2 Circular Restricted Three-Body Problem (CR3BP)

Although the $\mathbb{N}$-body problem is not analytically solvable, some simplifications add insight and lead to useful results. Fundamentally, the assessment of the gravitational influence of multiple bodies on a spacecraft is the focus of mission design. Incorporating only the gravitational forces of the largest contributors can offer a time-efficient baseline solution during the initial mission design phase. A number of reasonable assumptions reduce the problem to the *three-body restricted* system which models the behavior of an infinitesimal spacecraft as influenced by the gravitational forces of two massive bodies. The differential equations that govern the state of the spacecraft P_3, in this system are a specific case of Eqn. (2.1):

$$M_3\mathbf{R}_3'' = -\tilde{G}\frac{M_3M_1}{\mathrm{R}_{13}^3}\mathbf{R}_{13} - \tilde{G}\frac{M_3M_2}{\mathrm{R}_{23}^3}\mathbf{R}_{23} \tag{2.2}$$

The subscripts in Eqn. (2.2) specify the specific bodies in the three-body model, the massive primaries are assumed as P_1 and P_2, and the spacecraft, P_3. It is further assumed that the mass of the spacecraft M_3, is negligible as compared to that of the primaries (M_1 and M_2), and does not affect their motion. As a consequence, the system of two primaries now reduces to a two body problem with a known conic solution. Choosing circular paths for the two primaries relative to their common barycenter is not necessary, but simplifies the preliminary calculations and yields reasonable baseline solutions. The assumption of 'circular' conic paths for the primaries further constrains the problem to the '*Circular* Restricted Three-Body Problem' (CR3BP).

The classical CR3BP is depicted in Fig. 2.2. In this system, an inertial frame fixed at the barycenter between the two primaries, B, is defined in terms of the unit vectors $\hat{\mathrm{X}} - \hat{\mathrm{Y}}$. The unit vector $\hat{\mathrm{Z}}$ completes the right-handed triad and aligns with the direction of the orbital angular momentum vector associated with the orbits of the primaries. The $\hat{\mathrm{X}} - \hat{\mathrm{Y}}$ plane is defined to be the plane of primary motion. The

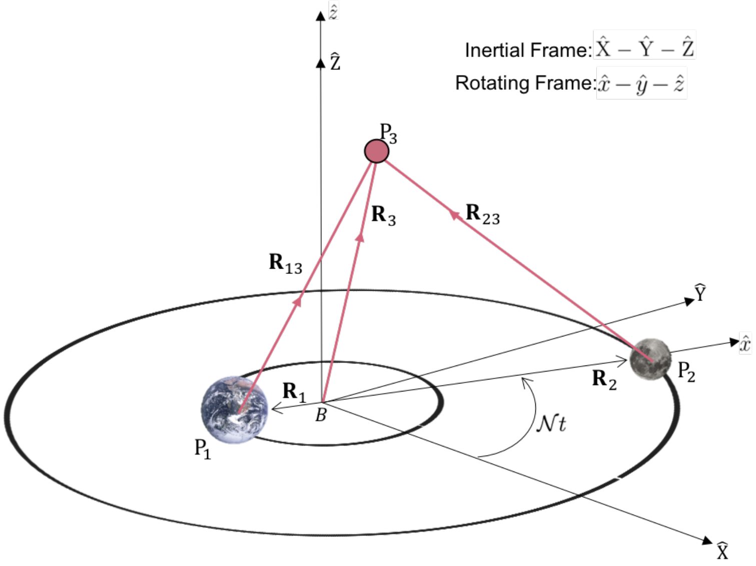


Figure 2.2: Circular Restricted Three-Body Problem (CR3BP) — illustrating time dependent motion of the primaries. As an example, the Earth and the Moon are denoted P_1 and P_2 respectively. Inertial frame defined by: $\hat{\mathrm{X}} - \hat{\mathrm{Y}} - \hat{\mathrm{Z}}$ and rotating frame defined by: $\hat{x} - \hat{y} - \hat{z}$

line joining the primaries revolves about the barycenter at a common and constant angular rate $\mathcal{N}$, the mean motion of the system:

$$\mathcal{N} = \sqrt{\frac{\tilde{G}(M_1 + M_2)}{(\mathrm{R}_1 + \mathrm{R}_2)^3}}. \tag{2.3}$$

Although the primaries revolve in planar orbits about the barycenter, the spacecraft is free to move spatially as well. In Fig. 2.2, the unit vectors $\hat{x} - \hat{y} - \hat{z}$ are defined as a rotating frame such that the unit vectors are aligned with the inertial unit vectors at time $t = 0$, and then rotate at the angular rate $\mathcal{N}$. Therefore, the orientation of this rotating frame and, consequently, that of the line joining the two primaries with respect to the inertial frame at an instant in time t, is calculated as $\mathcal{N}t$. This simplified computation results due to the assumption of circular orbits for the primaries P_1 and P_2. The distance between the spacecraft and the primaries at any given time, as viewed from the inertial frame is evaluated as [3]:

$$\mathrm{R}_{13} = [(\mathrm{X} - \mathrm{X}_1)^2 + (\mathrm{Y} - \mathrm{Y}_1)^2 + (\mathrm{Z} - \mathrm{Z}_1)^2]^{1/2} \tag{2.4a}$$

$$\mathrm{R}_{23} = [(\mathrm{X} - \mathrm{X}_2)^2 + (\mathrm{Y} - \mathrm{Y}_2)^2 + (\mathrm{Z} - \mathrm{Z}_2)^2]^{1/2} \tag{2.4b}$$

where, X and Y are the in-plane excursions and Z is the out-of-plane coordinate of P_3 relative to the barycenter as viewed in the inertial frame. The position coordinates for the primaries P_1 and P_2 in circular orbits is computed as

$$\begin{aligned} \mathrm{X}_1 &= -\mathrm{R}_1 \cos \mathcal{N}t, & \mathrm{X}_2 &= \mathrm{R}_2 \cos \mathcal{N}t \\ \mathrm{Y}_1 &= -\mathrm{R}_1 \sin \mathcal{N}t, & \mathrm{Y}_2 &= \mathrm{R}_2 \sin \mathcal{N}t \\ \mathrm{Z}_1 &= 0, & \mathrm{Z}_2 &= 0. \end{aligned} \tag{2.5}$$

Equation. (2.2) is rewritten to explicitly demonstrate the time-dependent nature of the motion of the particle P_3 under the gravitational influence of the primaries. In terms of the scalar components,

$$\mathbf{X}'' = -\tilde{G}\left[\frac{M_1(\mathrm{X} + \mathrm{R}_1\cos(\mathcal{N}t))}{\mathrm{R}_{13}^3} + \frac{M_2(\mathrm{X} - \mathrm{R}_2\cos(\mathcal{N}t))}{\mathrm{R}_{23}^3}\right] \tag{2.6a}$$

$$\mathbf{Y}'' = -\tilde{G}\left[\frac{M_1(\mathrm{Y} + \mathrm{R}_1\sin(\mathcal{N}t))}{\mathrm{R}_{13}^3} + \frac{M_2(\mathrm{Y} - \mathrm{R}_2\sin(\mathcal{N}t))}{\mathrm{R}_{23}^3}\right] \tag{2.6b}$$

$$\mathbf{Z}'' = -\tilde{G}\left[\frac{M_1\mathrm{Z}}{\mathrm{R}_{13}^3} + \frac{M_2\mathrm{Z}}{\mathrm{R}_{23}^3}\right]. \tag{2.6c}$$

The position and derivable velocity quantities in Eqns. (2.6) exist on scales of vastly different orders of magnitude. This disparity in scales could introduce numerical inaccuracies for the states during numerical integration, so it is beneficial to nondimensionalize these equations. Some quantities, such as the relative distances between primaries as well as their masses, remain constant in the CR3BP. Thus, these quantities are exploited to nondimensionalize and are termed the 'characteristic quantities'. The characteristic length l^*, is selected as the sum of the distances of the two primaries from the barycenter:

$$l^* = \mathrm{R}_1 + \mathrm{R}_2 \tag{2.7}$$

The characteristic mass m^* of the system is the sum of the masses of the two primaries:

$$m^* = M_1 + M_2 \tag{2.8}$$

Then, the characteristic time is formulated to ensure that the nondimensional gravitational parameter is unity:

$$t^* = \sqrt{\frac{(\mathrm{R}_1 + \mathrm{R}_2)^3}{\tilde{G}(M_1 + M_2)}} = \sqrt{\frac{l^{*3}}{\tilde{G}m^*}}. \tag{2.9}$$

The characteristic velocity v^* is defined as:

$$v^* = \frac{l^*}{t^*} \tag{2.10}$$

The values of these characteristic quantities for the Earth-Moon system are listed in Table 2.1; they are employed to nondimensionalise Eqns. (2.6). The nondimensional mean motion is defined as:

Table 2.1: Characteristic quantities in the Earth-Moon system

Quantity	**Value** (approximate)
Mass parameter (μ)	0.01215
Earth moon distance (l^*)	384400 km
Characteristic Time (t^*)	4.3426 days
Characteristic Velocity (v^*)	1.0245 km/s

$$\mathrm{n} = \mathcal{N}t^* = 1. \tag{2.11}$$

The nondimensional mass parameter is defined as:

$$\mu = \frac{M_2}{m^*}. \tag{2.12}$$

As apparent in Eqn. (2.12), the mass parameter, μ, is a measure of the mass of the second (smaller) primary compared to the total mass of the system; values for the mass parameter occur in the range (0,0.5]. Dividing Eqn. (2.8) by m^*, substituting μ from Eqn. (2.12), and rearranging produces the nondimensional expression for the ratio of the larger primary mass of P_1 to the system mass:

$$\frac{M_1}{m^*} = 1 - \mu. \tag{2.13}$$

Nondimensionalizing distances to the primaries is straightforward. The Center of Mass (CoM) of the system is defined at the barycenter, B, such that:

$$B\hat{x} = \frac{-M_1\ r_1\hat{x} + M_2\ r_2\hat{x}}{m^*} = \mathbf{0} \tag{2.14}$$

where, $\boldsymbol{r}_1 = \frac{\mathbf{R}_1}{l^*}$, $\boldsymbol{r}_2 = \frac{\mathbf{R}_2}{l^*}$, $r_1 = |\boldsymbol{r}_1|$ and $r_2 = |\boldsymbol{r}_2|$. Recall that these vectors are defined solely along the $\hat{x}$ direction. Substituting the nondimensional forms of the mass ratios and of Eqn. (2.7), where $r_1 + r_2 = 1$ into Eqn. (2.14) and rearranging yields:

$$\begin{aligned} (1-\mu)r_1 &= \mu(1-r_1) \\ r_1 &= \mu \qquad (2.15) \\ r_2 &= 1-\mu. \qquad (2.16) \end{aligned}$$

Again, since the primary position vectors lie in the $\hat{x}$ direction, their nondimensional positions from the barycenter are written as:

$$\boldsymbol{r}_1 = -\mu\,\hat{x} \tag{2.17}$$

$$\boldsymbol{r}_2 = 1-\mu\,\hat{x}. \tag{2.18}$$

The nondimensional position vector of P_3 as measured from the barycenter is:

$$\boldsymbol{r}_3 = \frac{\mathbf{R}_3}{l^*}. \tag{2.19}$$

The nondimensional time is expressed as:

$$\tau = \frac{t}{t^*}. \tag{2.20}$$

The relevant nondimensional quantities are now substituted into Eqn. (2.2) to render the nondimensional Equations of Motion (EOMs) for the spacecraft as viewed by an inertial observer as:

$$ {}^{I}\ddot{\boldsymbol{r}}_3 = -\frac{(1-\mu)}{r_{13}^3}\boldsymbol{r}_{13} - \frac{\mu}{r_{23}^3}\boldsymbol{r}_{23} \tag{2.21} $$

The dots above $\boldsymbol{r}$ indicate the 2^{nd} derivative of position with respect to nondimensional time. Analysis and generation of solutions for such a system was simplified in 1722 by the famous Swiss mathematician and physicist Leonhard Euler during examination of the motion of the Moon (P_3) in the Sun-Earth (P_1 and P_2, respectively) system. He proposed reformulating the equations relative to the rotating frame [64] where the primaries are fixed with respect to each other, therefore, eliminating the explicit time-dependent nature of the EOMs associated with the motion of P_3.

2.2.1 Formulation of the CR3BP EOMs in the Rotating Frame

The reformulation of the EOMs in the rotating frame enables the exploitation of certain geometries and symmetries to simplify the design of novel trajectory solutions in various mission design scenarios. An illustration of the rotating frame in the Earth-Moon system includes P_3 and the nondimensional system parameters appears in Fig. 2.3.

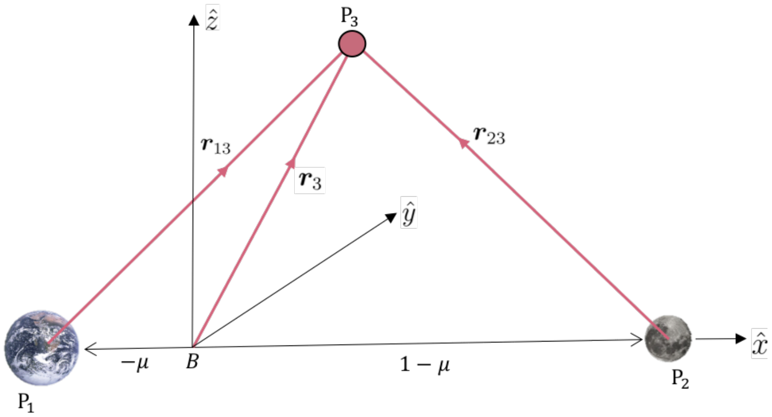


Figure 2.3: CR3BP — nondimensional system parameters in the rotating frame

The nondimensional position vector for P$_3$ with respect to the barycenter is rewritten in terms of the rotating coordinate frame as $\boldsymbol{r} = x\hat{x} + y\hat{y} + z\hat{z}$. The position of P$_3$ with respect to P$_1$ as well as P$_2$ then appear in nondimensional form as:

$$\boldsymbol{r}_{13} = (x+\mu)\hat{x} + y\hat{y} + z\hat{z} \tag{2.22}$$

$$\boldsymbol{r}_{23} = (x+\mu-1)\hat{x} + y\hat{y} + z\hat{z} \tag{2.23}$$

When reformulating the EOMs in the rotating frame, the kinematical expansions for the acceleration incorporates the Coriolis and centrifugal acceleration terms. The kinematical rate of change of velocity (acceleration) with respect to the inertial observer is derived as:

$$\frac{{}^{I}d^2\boldsymbol{r}_3}{d\tau^2} = \frac{{}^{R}d^2\boldsymbol{r}_3}{d\tau^2} + \underbrace{2{}^{I}\boldsymbol{\omega}^{R} \times \frac{{}^{R}d\boldsymbol{r}_3}{d\tau}}_{\text{Coriolis}} + \underbrace{{}^{I}\boldsymbol{\omega}^{R} \times {}^{I}\boldsymbol{\omega}^{R} \times \boldsymbol{r}_3}_{\text{Centrifugal}}. \tag{2.24}$$

In Eqn. (2.24), ${}^{I}\boldsymbol{\omega}^{R}$ is the angular velocity of the rotating reference frame with respect to the inertial frame, ${}^{I}\boldsymbol{\omega}^{R} = \mathrm{n}\hat{z} = 1\hat{z}$. The superscript on the left of a vector indicates the frame from which the quantity is being observed. Expanding Eqn. (2.24) and rewriting in Cartesian coordinate component form produces the kinematical expression:

$$\ddot{\boldsymbol{r}}_3 = (\ddot{x} - 2\mathrm{n}\dot{y} - \mathrm{n}^2 x)\hat{x} + (\ddot{y} + 2n\dot{x} - \mathrm{n}^2 y)\hat{y} + \ddot{z}\hat{z}. \tag{2.25}$$

Equating Eqn. (2.25) to Eqn. (2.21) and rearranging the terms results in the nondimensional second-order differential equations that dictate the motion of P$_3$ as expressed in terms of the CR3BP rotating frame,

$$\ddot{x} - 2n\dot{y} - \mathrm{n}^2 x = -\frac{(1-\mu)(x+\mu)}{r_{13}^3} - \frac{\mu(x-1+\mu)}{r_{23}^3} \tag{2.26a}$$

$$\ddot{y} + 2n\dot{x} - \mathrm{n}^2 y = -\frac{(1-\mu)y}{r_{13}^3} - \frac{\mu y}{r_{23}^3} \tag{2.26b}$$

$$\ddot{z} = -\frac{(1-\mu)z}{r_{13}^3} - \frac{\mu z}{r_{23}^3} \tag{2.26c}$$

where,

$$r_{13} = \sqrt{(x+\mu)^2 + y^2 + z^2}$$
$$r_{23} = \sqrt{(x+\mu-1)^2 + y^2 + z^2}$$

Note that the sum of the centrifugal (Eqn. (2.24)) and gravitational (Eqn. (2.21)) accelerations are representative of the gradient of a pseudo-potential term U^*:

$$\nabla U^* = ({}^I\boldsymbol{\omega}^R \times^I \boldsymbol{\omega}^R \times \boldsymbol{r}_3) + \left[-\frac{(1-\mu)}{r_{13}^3}\boldsymbol{r}_{13} - \frac{\mu}{r_{23}^3}\boldsymbol{r}_{23} \right] \tag{2.27}$$

where,

$$U^*(x,y,z) = \frac{1-\mu}{r_{13}} + \frac{\mu}{r_{23}} + \frac{1}{2}(x^2+y^2) \tag{2.28}$$

is a function only of the position. These terms simplify the representation of the EOMs in Eqns. (2.26), i.e.:

$$\ddot{x} = 2\dot{y} + \frac{\partial U^*}{\partial x} \tag{2.29a}$$
$$\ddot{y} = -2\dot{x} + \frac{\partial U^*}{\partial y} \tag{2.29b}$$
$$\ddot{z} = \frac{\partial U^*}{\partial z} \tag{2.29c}$$

where, $\frac{\partial U^*}{\partial x} = \nabla U^*_x$, $\frac{\partial U^*}{\partial y} = \nabla U^*_y$ and $\frac{\partial U^*}{\partial z} = \nabla U^*_z$. For convenience, the CR3BP states are compiled into the state vector:

$$\boldsymbol{\chi} = [x, y, z, \dot{x}, \dot{y}, \dot{z}]^T \tag{2.30}$$

The gradient of the pseudo-potential term is exploited to compute the equilibrium solutions to the EOMs in Eqns. (2.29).

2.2.2 Equilibrium Solutions

Some mission applications require specific orbits/configurations to be maintained for extended time intervals, for example, space weather monitoring or communication satellites. Such scenarios benefit from searches for equilibrium solutions, where P_3 is stationary with respect to an observer in the rotating frame. An equilibrium solution is realized by the particular solutions of Eqn. (2.29) — where the gravitational forces and the centrifugal forces due to the rotating frame render the accelerations and velocities of a particle relative to the rotating frame zero. The equilibrium position coordinates are determined from the gradient of the pseudo-potential term $\nabla \boldsymbol{U^*} = \mathbf{0}$, and, are termed the *Lagrange* points. In the following equations where $\nabla \boldsymbol{U^*} = \mathbf{0}$ is formulated,

$$\frac{\partial U^*}{\partial x} = -\frac{(1-\mu)(x_{eq}+\mu)}{r_{13_{eq}}^3} - \frac{\mu(x_{eq}+\mu-1)}{r_{23_{eq}}^3} + \mathrm{n}^2 x_{eq} = 0 \tag{2.31a}$$

$$\frac{\partial U^*}{\partial y} = -\frac{(1-\mu)y_{eq}}{r_{13_{eq}}^3} - \frac{\mu y_{eq}}{r_{23_{eq}}^3} + \mathrm{n}^2 y_{eq} = 0 \tag{2.31b}$$

$$\frac{\partial U^*}{\partial z} = -\frac{(1-\mu)z_{eq}}{r_{13_{eq}}^3} - \frac{\mu z_{eq}}{r_{23_{eq}}^3} = 0 \tag{2.31c}$$

the subscript '*eq*' indicates that the quantities are evaluated at the equilibrium points. It is immediately evident that Eqn. (2.31c) $= 0$ *iff* $z_{eq} = 0$. Therefore, only planar equilibrium solutions exist in the CR3BP. Also, assuming $y_{eq} = 0$ in Eqn. (2.31b) yields *zero*, suggesting that at least some equilibrium points lie along the $\hat{x}$ axis where $y_{eq} = z_{eq} = 0$. So, substituting $y_{eq} = 0$ and $z_{eq} = 0$ into Eqn. (2.31a) produces the following equation:

$$x_{eq} = \frac{(1-\mu)(x_{eq}+\mu)}{|x_{eq}+\mu|^3} + \frac{\mu(x_{eq}-1+\mu)}{|x_{eq}-1+\mu|^3} \tag{2.32}$$

which possesses 3 real roots and 2 imaginary roots when solving for x_{eq}. Only the real roots are notable when investigating the equilibrium solutions in the CR3BP and are computed via root-finding strategies (e.g., iterative Newton's method). For considerations of numerical stability, the roots are solved indirectly by evaluating the relative distances (d_i, for $i = 1, 2, 3$) of the equilibrium solutions from the nearest primary (x_{L1} and x_{L2} from P_2, and x_{L3} from P_1):

$$\mathrm{d}_1 = -(x_{L1} + \mu - 1) \tag{2.33a}$$

$$\mathrm{d}_2 = x_{L2} + \mu - 1 \tag{2.33b}$$

$$\mathrm{d}_3 = x_{L3} - \mu \tag{2.33c}$$

The d_i from Eqns. (2.33) are individually substituted back into Eqn. (2.32) to solve for the corresponding x_{Li} using Newton's iterative method:

$$\mathrm{d}_i(j+1) = \mathrm{d}_i(j) - \frac{f(\mathrm{d}_i(j))}{f'(\mathrm{d}_i(j))} \tag{2.34}$$

where j is the current iteration. A maximum number of iterations and an acceptable numerical tolerance for convergence (e.g., 1×10^{-12}) in nondimensional units are assessed to terminate the iterative process. The three solutions that lie along the $\hat{x}$ axes are labelled the collinear Lagrange points in the CR3BP and their position is retrieved by substituting the values of d_i into Eqns. (2.33). The two remaining equilibrium points result from solving Eqns. (2.31a) and (2.31b) simultaneously for $(x_{eq}, y_{eq})_{4,5}$. These solutions are available analytically, resulting in:

$$x_{4,5} = \frac{1}{2} - \mu \tag{2.35a}$$

$$y_{4,5} = \pm\frac{\sqrt{3}}{2} \tag{2.35b}$$

These final two Lagrange points are located such that they form equilateral triangles with the two primaries. The L_4 point leads P_2 by 60° and L_5 trails P_2 by 60° when their motion is observed in an inertial frame. Figure 2.4 illustrates the five Lagrange point locations in the Earth-Moon system rotating frame and Table 2.2 lists their coordinates. Although a spacecraft would be theoretically stationary if located

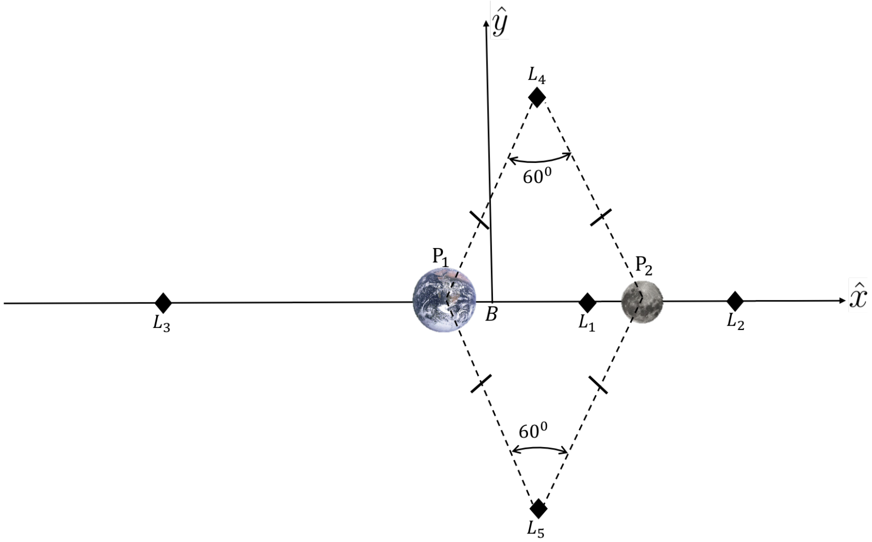


Figure 2.4: Schematic of Lagrange point locations in the CR3BP (e.g., Earth-Moon system). Not to scale.

precisely at one of these points, each solution has different stability characteristics that determine the level of difficulty associated with maintaining operations in their vicinity. The motion in the vicinity of the Lagrange points is also characterized by varied energy levels.

Table 2.2: Lagrange point locations in the Earth-Moon System

	Nondimensional		**Dimensional, *km***	
Lagrange Point	x	y	x	y
L_1	0.83691531	0	321710.2452	0
L_2	1.15568202	0	444244.1690	0
L_3	-1.00506263	0	-386346.0751	0
L_4	0.487849452	0.866025404	187529.3293	332900.1652
L_5	0.487849452	-0.866025404	187529.3293	-332900.1652

2.2.3 The Jacobi Constant

The CR3BP as formulated in the rotating frame coordinates offers access to a useful quantity, the Jacobi constant (JC), a pseudo-energy term that constrains the motion of P$_3$ within the CR3BP design space. The JC is the only known integral of the motion associated with the autonomous EOMs (Eqn. (2.29)). An expression for JC is delivered by first integrating the dot product between the gradient of the pseudo-potential

$$\nabla U^* = (\ddot{x} - 2\dot{y})\hat{x} + (\ddot{y} + 2\dot{x})\hat{y} + \ddot{z}\hat{z}, \tag{2.36}$$

and the rotating frame velocity vector components

$$\dot{\boldsymbol{r}} = \dot{x}\hat{x} + \dot{y}\hat{y} + \dot{z}\hat{z}, \tag{2.37}$$

to yield [65]:

$$\underbrace{\frac{\partial U^*}{\partial x}\frac{dx}{d\tau} + \frac{\partial U^*}{\partial y}\frac{dy}{d\tau} + \frac{\partial U^*}{\partial z}\frac{dz}{d\tau}}_{\frac{dU^*}{d\tau}} = (\ddot{x} - 2\dot{y})\dot{x} + (\ddot{y} + 2\dot{x})\dot{y} + \ddot{z}\dot{z}. \tag{2.38}$$

Simplification of Eqn. (2.38) and the integration of both terms with respect to time:

$$\int \dot{\boldsymbol{r}}_3 \cdot \ddot{\boldsymbol{r}}_3 \, d\tau = \int \underbrace{\dot{x}\ddot{x} + \dot{y}\ddot{y} + \dot{z}\ddot{z}}_{\frac{1}{2}\frac{dv^2}{d\tau}} d\tau = \int \underbrace{\frac{\partial U^*}{\partial x}\frac{dx}{d\tau} + \frac{\partial U^*}{\partial y}\frac{dy}{d\tau} + \frac{\partial U^*}{\partial z}\frac{dz}{d\tau}}_{\frac{dU^*}{d\tau}} d\tau, \tag{2.39}$$

produces

$$\int \dot{\boldsymbol{r}}_3 \cdot \ddot{\boldsymbol{r}}_3 \, d\tau = \frac{1}{2}(\dot{x}^2 + \dot{y}^2 + \dot{z}^2) = U^* - JC. \tag{2.40}$$

Equation. (2.40) simplifies to:

$$v^2 = 2U^* - JC \tag{2.41}$$

where, 'v' is the velocity magnitude as observed in the rotating frame. Consistent with Eqn. (2.41), the JC is not an explicit function of time. It is a quantity that is influenced by the velocity magnitude of P_3 and its position (via U^*) at a given instant in time in the rotating frame. In astrodynamics applications, the JC is often exploited to estimate the propellant requirements to maneuver within a 3B system.

For example, consider two intersecting arcs with velocities $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$. The change in energy between these arcs is bridged via a change in speed imparting the Δv:

$$\Delta v = |\boldsymbol{v}_1 - \boldsymbol{v}_2| = \sqrt{v_1^2 + v_2^2 - 2v_1v_2 \ cos(\theta)} \tag{2.42}$$

Assuming alignment between the velocity vectors ($\theta^0 = 0^0$) renders the minimal Δv value

$$\min \Delta v = \sqrt{v_1^2 + v_2^2 - 2v_1v_2}. \tag{2.43}$$

Also, from Eqn. (2.41):

$$v_1^2 - v_2^2 = JC_1 - JC_2 = |\Delta JC| \tag{2.44}$$

Using this knowledge and substituting for v_1 in Eqn. (2.43), the Δv calculation is rearranged as:

$$\min \Delta v = \sqrt{2v_2^2 + \Delta JC - 2v_2\sqrt{v_2^2 + \Delta JC}} \tag{2.45}$$

The Δv value in Eqn. (2.45) is minimized as $\Delta JC \to 0$. So, transfers between solutions at similar energy levels ($\Delta JC \to 0$) may require smaller maneuvers and, thus, lower amounts of propellant. A change in the direction of velocity, even between states at similar energy levels, is not accounted for by Eqn. (2.45). Thus, given the departure and arrival states in the nonlinear multiple-body regime, the energy-like JC value aids in 'estimating' the size of potential maneuvers for conducting trajectory design in the CR3BP. The JC values at the Lagrange points where $v_{L_i} = 0$ are listed in Table 2.3. As is evident from Table 2.3, the Jacobi constant values decrease from $L_1 \to L_4$ and the values are equal at L_4 and L_5. Although systems with a different mass parameter value μ do not possess the same values of the Jacobi constant at their Lagrange points, the same trend holds true between each: $JC_{L_1} > JC_{L_2} > JC_{L_3} > JC_{L_4} = JC_{L_5}$. A decrease in the JC value usually represents an increase in the energy associated with a solution. This increase in energy is dynamically significant.

Table 2.3: Jacobi constant values associated with Lagrange points

Lagrange Point	**Jacobi constant value**
L_1	3.188341
L_2	3.172160
L_3	3.012147
L_4	2.237997
L_5	2.237997

2.2.4 Zero Velocity Surfaces and Forbidden Regions

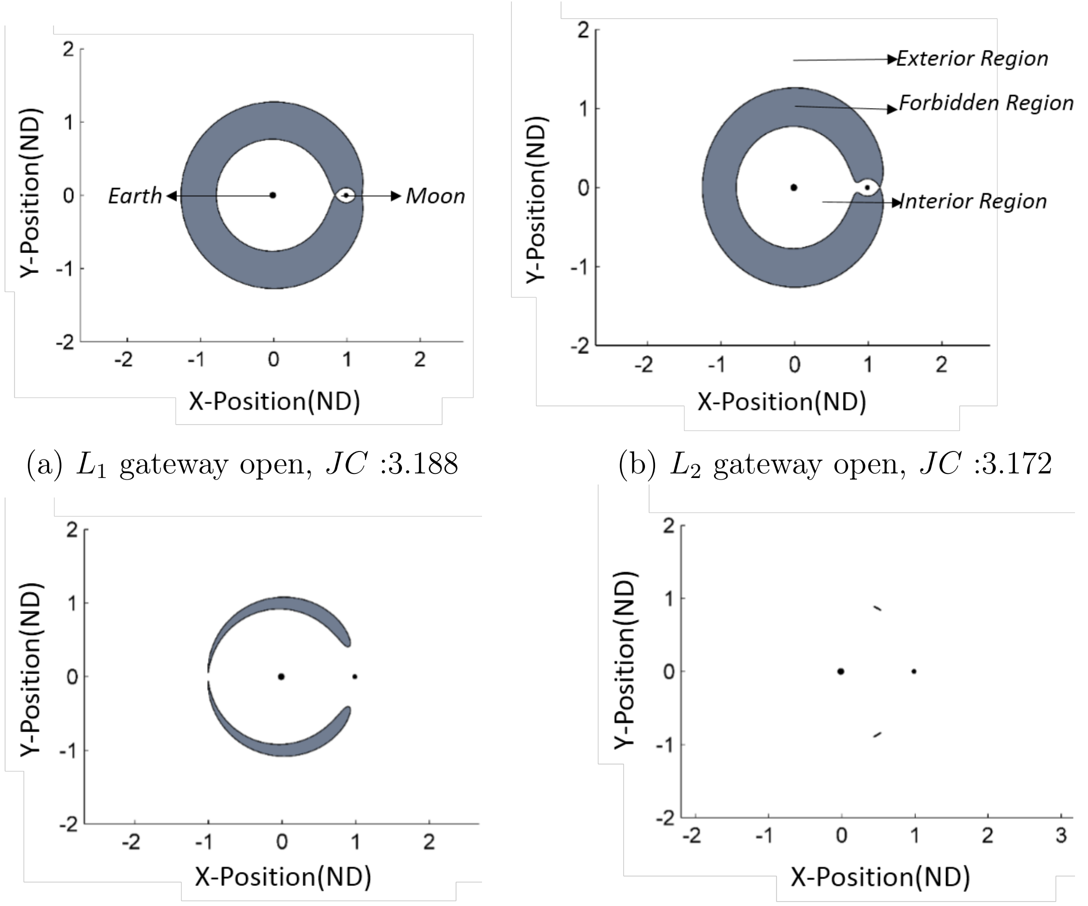
The Jacobi constant is defined in Eqn. (2.41) in terms of position in configuration space and relative speed. Equation. (2.41) indicates that some regions in the CR3BP configuration space are not traversible by P_3 when the relative speed of the spacecraft v is imaginary, i.e., when $JC > 2U^*(x, y, z)$. So, for a specified energy level JC, the spacecraft is only able to traverse spaces that satisfy the condition:

$$2U^*(x, y, z) \geq JC. \tag{2.46}$$

Solving Eqn. (2.41) for all the boundaries or areas in the CR3BP configuration space where this condition is violated (i.e., the speed v is imaginary) produces surfaces in 3D space [66,67] that are termed *forbidden* regions. The projections of these surfaces onto the $\hat{x} - \hat{y}$ plane yield Zero Velocity Curves (ZVCs), where the boundaries represent the condition $JC = 2U^*(x, y, z)$.

The grey regions bounded by the curves in Fig. 2.5 are *forbidden* to a spacecraft (s/c) because the $2U^*(x, y, z) \geq JC$ condition is not satsified. The initial state of the spacecraft determines whether it traverses the white space that is either *interior* to the *forbidden* region, or *exterior*, or both, when propagated in the CR3BP regime. As illustrated in Fig. 2.5, the Lagrange point *gateways* open at exactly their Jacobi constant values. So, traversal between the primaries is possible only when the L_1 gateway is open, and between the *interior* and *exterior* regions when, at least, the L_2 gateway is open. The increase in energy required to transition through gateways from lower energy levels is enabled by executing energy-increasing maneuvers, and capture around primaries from higher energy levels is enforced by executing energy-lowering maneuvers to close the desired dynamical gateways. In both cases, the maneuvers modify velocity in the state of the s/c and, thus, the associated JC value.

The forbidden region shrinks with higher energies (lower JC values), reducing to just two points at L_4 and L_5 at $JC = JC_{L_4} = JC_{L_5}$. If the Jacobi constant is lowered below those corresponding to the equilateral Lagrange points ($JC < JC_{L_4,L_5}$), then the spacecraft is free to travel anywhere in the entire $\hat{x} - \hat{y}$ configuration space as the zero velocity regions no longer exist in the plane. However, the existence of the



(a) L_1 gateway open, JC :3.188

(b) L_2 gateway open, JC :3.172

(c) L_1 gateway open, JC :3.012

(d) L_4, L_5 gateways open, JC :2.988

Figure 2.5: ZVCs associated with Lagrange point Jacobi constant values (rounded values). Axes denote nondimensional (ND) units.

zero velocity surface out-of-the-plane still precludes the excursion of the s/c to certain regions in the spatial configuration space.

2.3 Low-Thrust Enabled Motion in the CR3BP

The CR3BP dynamical regime is used to explore the solution space associated with emerging technologies such as advanced ion-drive propulsion. This relatively new technology is gaining popularity due to the ability to improve payload delivery capabilities. This efficiency, however, is delivered at lower thrust levels and, therefore, potentially lengthy flight durations [68]. The differential equations in Eqn. (2.29) are absorbed into the general formulation below to include contributions from both the natural gravitational and the thrust acceleration sources to capture the motion of the spacecraft (s/c), and its mass-history over time:

$$\dot{\boldsymbol{\chi}}_{LT} = \begin{Bmatrix} \dot{\boldsymbol{r}} \\ \dot{\boldsymbol{v}} \\ \dot{m} \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{v} \\ \underbrace{\boldsymbol{f}(\boldsymbol{r}) + \boldsymbol{g}(\boldsymbol{v})}_{\text{natural}} + \underbrace{\frac{\mathbb{T}}{m}\hat{u}}_{\text{low-thrust}} \\ \frac{-\mathbb{T}}{Isp \cdot g_o} \end{Bmatrix} \tag{2.47}$$

In these equations, $\boldsymbol{\chi}_{LT}$ is the full nondimensional state vector comprising the vehicle position and velocity vectors ($\boldsymbol{r}$ and $\boldsymbol{v}$ respectively) within the context of the CR3BP, and the nondimensional vehicle mass m for a general low-thrust (LT) regime. The dimensional s/c mass (M_3) is scaled by the quantity $m^*_{sc} = M_3$ to set the initial nondimensional vehicle mass quantity to unity:

$$m_0 = \frac{M_3}{m^*_{sc}} = 1. \tag{2.48}$$

The nondimensional thrust magnitude ($\mathbb{T}$), is initialized by scaling its dimensional equivalent ($\widetilde{\mathbb{T}}$):

$$\mathbb{T} = \widetilde{\mathbb{T}} \cdot \frac{t^{*2}}{{m^*}_{sc} \cdot l^* \cdot 1000} \;, \tag{2.49}$$

The thrust direction in Eqn. (2.47) is represented via $\hat{u}$ (where a caret identifies unit magnitude), Isp is the nondimensional engine specific impulse $Isp = \frac{\widetilde{Isp}}{t^*}$, and g_0 is the reference nondimensional gravitational acceleration, i.e., $g_0 = \widetilde{g}_0 \cdot \frac{t^{*2}}{l^* \cdot 1000}$. To deliver the thrust force in Eqn. (2.47), two different low-thrust engine models are examined in this investigation. Currently, operational s/c are built to function at a relatively steady specific impulse (Constant Specific Impulse, CSI). An alternative promising technology under development focuses on continuously tuning the specific impulse (Variable Specific Impulse, VSI), to optimize the locations of high and low thrust burns, thereby optimizing the propellant consumption [69]. In both the CSI and VSI models, the available thrust magnitude is a function of the engine power allocation, ($\mathbb{P}$) and efficiency (Isp). The relationship is modeled as:

$$\mathbb{T} = \frac{2\mathbb{P}}{Isp \cdot g_0} \tag{2.50}$$

where the nondimensional power quantity is defined as: $\mathbb{P} = \widetilde{\mathbb{P}} \cdot \frac{t^{*3}}{m^*_{sc} \cdot (l^* \cdot 1000)^2}$. In the CSI regime, constant power and Isp parameters dictate a constant thrust magnitude. The basis of the control authority that maneuvers the s/c for constant power $\mathbb{P}$ is an on-off engine toggle ($\mathbb{T} = 0$ or $\mathbb{T} = \mathbb{T}_{max}$) and thrust-vectoring via the pointing direction $\hat{u}$. In this regime, the constant mass-flow rate quantity is computed analytically as the constant:

$$\dot{m}_{CSI} = -\frac{\mathbb{T}}{Isp \cdot g_0} \tag{2.51}$$

This $\dot{m}$ value also facilitates the analytical computation of the spacecraft (s/c) mass history for input to the EOMs in Eqn. (2.47):

$$m(\tau) = m_0 - \frac{\mathbb{T}\tau}{Isp \cdot g_0} \tag{2.52}$$

In contrast, a power-limited VSI engine functions on variable thrust magnitude and Isp values. The mass-flow rate, is therefore, a variable quantity and is integrated at each time-step to compute the spacecraft mass-history. This quantity is rewritten as follows to simplify derivations:

$$\dot{m}_{VSI} = -\frac{\mathbb{T}^2}{2\mathbb{P}} \tag{2.53}$$

The value of specific impulse (Isp) never modulates to exactly zero for a VSI regime; thus, the engine is assumed to be always 'on' and thrusting even when the throughput is negligible. Compared to a CSI regime perspective, however, very high Isp values in a VSI regime correspond to very low thrust magnitudes (Eqn. (2.50)) for a fixed power input and, thus, coast arcs / engine-off conditions. An unbounded Isp modulation ability enables the VSI engine to transition between the capabilities of a chemical and low-thrust engine as required; this ability can be exploited to inform the applicability of a particular engine for a given mission design scenario. Computation of the thrust magnitude and pointing direction histories are a function of the low-thrust s/c type and also the various assumptions associated with the implementation of the corresponding numerical simulation.

2.4 Spacecraft Motion in the Higher-Fidelity Ephemeris Regime

The CR3BP offers an opportunity to approximate the higher fidelity dynamics and to exploit the natural flows that are otherwise unavailable in simpler dynamical regimes. In reality, however, the motion of the celestial bodies are time-dependent; consequentially, the gravitational influence from these bodies on the spacecraft motion also depends on time. Modeling the relative positions of the celestial bodies via numerical simulations can be time-consuming and computationally intensive. So, one option to balance modeling accuracy and computational effort is to exploit the time-invariant nature of the CR3BP to design trajectories for a particular mission scenario, then transition this solution to the higher-fidelity ephemeris model for validation. The mission epoch is a critical quantity during this transition as it is correlated with the nature of the dynamics in the ephemeris model; the epoch will, therefore, influence the nature of the final solution in this regime as well. A first step towards such a transition is the modification of Eqn. (2.1) to incorporate the relative motion of $\mathbb{N}$ bodies in a system with respect to each other. Representation of their gravitational influence on the spacecraft, P_3, and the inclusion of the effects from the s/c engine capabilities is, then, incorporated as well:

$$\boldsymbol{\mathfrak{R}}''_{q3} = \underbrace{-\frac{\tilde{G}(M_q + M_3)}{\boldsymbol{\mathfrak{R}}^3_{q3}}\boldsymbol{\mathfrak{R}}_{q3}}_{\text{natural dominant}} + \underbrace{\tilde{G}\sum_{\substack{j=1\\ j\neq q,3}}^{\mathbb{N}} M_j \left(\frac{\boldsymbol{\mathfrak{R}}_{3j}}{\boldsymbol{\mathfrak{R}}^3_{3j}} - \frac{\boldsymbol{\mathfrak{R}}_{qj}}{\boldsymbol{\mathfrak{R}}^3_{qj}}\right)}_{\text{natural perturbing}} + \underbrace{\frac{\tilde{\mathbb{T}}}{M}\hat{\mathfrak{u}}}_{\text{s/c engine}} \tag{2.54}$$

Here, the position vectors $\boldsymbol{\mathfrak{R}}$ are introduced in the J2000 coordinate frame. The J2000 frame is an Earth-centered inertial reference frame where the x-axis is directed along the vernal equinox, the z-axis is parallel to the Earth spin axis direction, and the y-axis completes the right-handed triad — all directions are recorded on 01 January 2000 at 12:00:00 ET (Julian Date: 2451545.0 ET) [70]. Here, ET is the acronym used for Ephemeris Time. The selection of such a specific reference epoch is a matter of convenience, and enables implementation of the numerical integrations process in the ephemeris regime by referencing the results with respect to '*days past J2000*'.

In Eqn. (2.54), the *dominant* acceleration term represents the gravitational influence of the central body q on the s/c, the body where the basepoint in the J2000 system is fixed. The *perturbing* acceleration includes the gravitational influence from all the other bodies included in the simulation; the acceleration from the s/c engine is a function of its thrust-to-mass ratio and thrust-vectoring options. The relative position quantities in Eqn. (2.54) associated with the gravitational bodies are computed from the DE421 ephemerides tracked and maintained by NASA's Jet Propulsion Laboratory; these values are accessed via SPICE toolkits [71]. The complete EOMs for a spacecraft as formulated in the ephemeris model are developed as:

$$\tilde{\chi}'_{Ephem} = \begin{Bmatrix} \boldsymbol{\mathfrak{R}}' \\ \boldsymbol{\mathfrak{V}}' \\ M' \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{\mathfrak{V}} \\ \boldsymbol{\mathfrak{A}}_{\text{dominant}} + \boldsymbol{\mathfrak{A}}_{\text{perturbing}} + \boldsymbol{\mathfrak{A}}_{\text{s/c engine}} \\ \frac{-\tilde{\mathbb{T}}}{\tilde{Isp}\cdot\tilde{g}_o} \end{Bmatrix} \tag{2.55}$$

where $\tilde{\boldsymbol{\chi}}'_{Ephem}$ represents the rates of change for the dimensional states of interest in the ephemeris model with respect to dimensional time t, the $\mathfrak{A}$ terms are the dimensional accelerations of interest as reflected in Eqn. (2.54), and all other quantities are dimensional in nature as well. Note that all celestial bodies are modeled as point-masses, and perturbation influences in addition to gravity such as J_2 effects and solar radiation pressure are not incorporated in this investigation.

2.5 Coordinate Transformations

The rotating frame coordinate system allows time-invariant dynamics, energy quantities, symmetries, and geometries to be exploited. However, to gain additional insights into the mission scenario and incorporate higher-fidelity analysis, the CR3BP solutions are transitioned to various inertial frames. The following approaches detail the steps required for such transformations.

2.5.1 Transformations between the CR3BP Rotating Frame and an Arbitrary Inertial Frame

The transformation from the CR3BP rotating frame to a celestial body-fixed arbitrary inertial frame aids in viewing or assessing characteristics such as the orbital elements associated with a trajectory. Furthermore, mission constraints may require thruster control laws to be developed with respect to such an inertial frame while preliminary design may benefit from a simultaneous analysis of the resulting trajectory outcomes in the medium-fidelity CR3BP. So, the ability to transition between the 3BP rotating frame and an arbitrary inertial frame is essential.

In Section 2.2, the CR3BP rotating frame is defined to pivot at a constant nondimensional rate ${}^{I}\boldsymbol{\omega}^{R} = \mathrm{n}\hat{z} = 1\hat{z}$, relative to an arbitrary inertial frame with its base-point fixed at the $P_1 - P_2$ barycenter. The resulting motion of the rotating frame with respect to the inertial frame is visualized in Fig. 2.6. Computation of the spacecraft

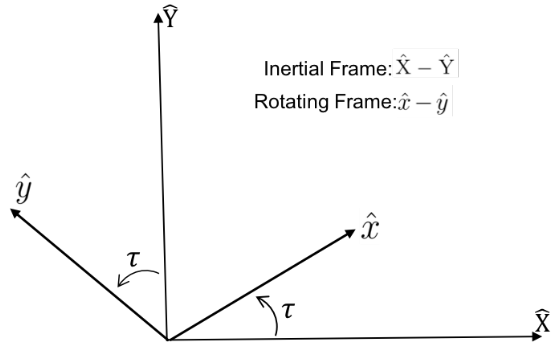

$\hat{Y}$
Inertial Frame: $\hat{X}-\hat{Y}$
Rotating Frame: $\hat{x}-\hat{y}$
$\hat{y}$
$\hat{x}$
τ
τ
$\hat{X}$

Figure 2.6: Conceptual illustration of rotating frame motion around fixed inertial frame. Both $\hat{Z}$ and $\hat{z}$ complete the right-handed triads.

states relative to a primary is initiated by translating the origin of the rotating frame from the barycenter to the central body (P_1 or P_2) of interest:

$$\begin{aligned} &\textit{With respect to } P_1: \ \boldsymbol{r}_r = \boldsymbol{r} - [-\mu,\ 0,\ 0] \quad = (x+\mu)\hat{x} + y\hat{y} + z\hat{z} \\ &\textit{With respect to } P_2: \ \boldsymbol{r}_r = \boldsymbol{r} - [1-\mu,\ 0,\ 0] \ = (x-1+\mu)\hat{x} + y\hat{y} + z\hat{z} \end{aligned} \tag{2.56}$$

Here, $\boldsymbol{r}_r$ represents the nondimensional position states in the rotating frame defined with respect to the primary. The shifted x-coordinate is represented by x_p, and the CR3BP primary-centered states are therefore defined as $\boldsymbol{\chi}_r = [\boldsymbol{r}_r, \dot{\boldsymbol{r}}_r] = [x_p, y, z, \dot{x}, \dot{y}, \dot{z}]^T$. When expressed in the rotating frame, the s/c velocity with respect to the central body is equivalent to the s/c velocity relative to the barycenter. The nondimensional inertial position states with respect to the primary $\mathbf{r}_I = [\mathrm{x}, \mathrm{y}, \mathrm{z}]^T$, are obtained by multiplying $\boldsymbol{r}_r$ by the direction cosine matrix C_{R2I} evaluated at $n\tau = \tau$:

$$\mathbf{r}_I = C_{R2I}\ \boldsymbol{r}_r \tag{2.57}$$

where C_{R2I} is developed as:

$$C_{R2I} = \begin{bmatrix} cos(\tau) & -sin(\tau) & 0 \\ sin(\tau) & cos(\tau) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.58}$$

The transformation matrix for the velocity rotation is derived from kinematic considerations relating the rotating velocities $\dot{\boldsymbol{r}}_r$ and inertial velocities $\dot{\mathbf{r}}_I$ [65]:

$$^{I}\dot{\boldsymbol{r}}_r = {}^{R}\dot{\boldsymbol{r}}_r + {}^{I}\boldsymbol{\omega}^{R} \times \boldsymbol{r}_r \tag{2.59}$$

where $^{I}\dot{\boldsymbol{r}}_r$ is the time derivative of the rotating frame position vector with respect to an inertial observer. The kinematic expansion in Eqn. (2.59) is further detailed as [65]:

$$^{I}\dot{\boldsymbol{r}}_r = (\dot{x} - \mathrm{n}y)\hat{x} + (\dot{y} + \mathrm{n}x_{\mathfrak{p}})\hat{y} + \dot{z}\hat{z} \tag{2.60}$$

Substituting the appropriate quantities for the unit vectors $[\hat{x},\ \hat{y},\ \hat{z}]$ from Eqns. (2.57) and (2.58) yields the following relationship between the inertial and rotating frame velocities:

$$\dot{\mathbf{r}}_I = \begin{bmatrix} \dot{C}_{R2I} & C_{R2I} \end{bmatrix} \begin{bmatrix} \boldsymbol{r}_r \\ \dot{\boldsymbol{r}}_r \end{bmatrix} \tag{2.61}$$

where,

$$\dot{C}_{R2I} = \begin{bmatrix} -sin(\mathrm{n}\tau) & -cos(\mathrm{n}\tau) & 0 \\ cos(\mathrm{n}\tau) & -sin(\mathrm{n}\tau) & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{2.62}$$

and, $\mathrm{n} = 1$ as defined in Section (2.11). The submatrices $\dot{C}_{R2I}$ and C_{R2I} are combined to execute the following rotation transformation for the entire state vector comprising of both the position and velocity quantities:

$$\begin{bmatrix} \mathrm{x} \\ \mathrm{y} \\ \mathrm{z} \\ \dot{\mathrm{x}} \\ \dot{\mathrm{y}} \\ \dot{\mathrm{z}} \end{bmatrix} = \underbrace{\begin{bmatrix} C_{R2I} & 0_{3\times3} \\ \dot{C}_{R2I} & C_{R2I} \end{bmatrix}}_{DCM_{R2I}} \begin{bmatrix} x_{\mathfrak{p}} \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \tag{2.63}$$

This direction cosine matrix 'DCM_{R2I}' transforms the spacecraft states from the rotating frame to a primary-centered arbitrary inertial frame. The inverse procedure acquires the rotating frame states from the inertial states and is achieved by exploiting the inverse of this matrix '$DCM_{R2I}^{-1} = DCM_{I2R}$'. Additionally, the thrusting direction unit vectors are simply rotated by C_{R2I}:

$$\hat{u}_I = C_{R2I} \; \hat{u}_R \tag{2.64}$$

The nondimensional s/c mass and thrust history values as determined in the rotating frame are transitioned unaltered to the arbitrary inertial frame as they are merely scalar magnitudes.

2.5.2 Transformations between the CR3BP Rotating Frame and J2000 Inertial Frame

The relative positions of the celestial bodies at a particular epoch influence their gravitational impact on a s/c and, thus, its motion through space. So, the CR3BP initial guesses are transitioned to the J2000 frame (introduced in Section 2.4) to incorporate the impact of mission-specific epochs into the trajectory design and to examine the resulting solutions in a higher-fidelity dynamical regime. Computation of the spacecraft states relative to the J2000 frame is pursued via two approaches, and are discussed as follows.

Implementation Approach 1

The first approach applying a coordinate transformation of the CR3BP states to the J2000 frame is initiated similar to the process in Section 2.5.1. The basepoint of the rotating frame is first translated from the barycenter to the central body P_1 or P_2 of interest (Eqn. (2.65) is repeated below for convenience):

$$\begin{aligned} &\textit{With respect to } P_1: \ \boldsymbol{r}_r = \boldsymbol{r} - [-\mu,\ 0,\ 0] &&= (x+\mu)\hat{x} + y\hat{y} + z\hat{z} \\ &\textit{With respect to } P_2: \ \boldsymbol{r}_r = \boldsymbol{r} - [1-\mu,\ 0,\ 0] &&= (x-1+\mu)\hat{x} + y\hat{y} + z\hat{z} \end{aligned}$$

Again, the nondimensional position states in the rotating frame defined with respect to the primary are re-written as $\boldsymbol{r}_r = x_{\mathfrak{p}}\hat{x} + y\hat{y} + z\hat{z}$. When expressed in the rotating frame, the s/c velocity with respect to the central body is equivalent to the s/c velocity relative to the barycenter. In the ephemeris model, the epoch determines the directions of $\hat{x}, \hat{y}, \hat{z}$ in the above equations; i.e., an instantaneous CR3BP rotating frame is defined for every epoch along the trajectory to be rotated into the higher-fidelity model. Rotation from these instantaneous CR3BP frames to the J2000 frame is initiated by defining the instantaneous vectors symbolized by '$\breve{\ }$' :

$$\hat{\breve{x}} = \frac{\breve{\boldsymbol{\mathfrak{R}}}_{12}}{\breve{\mathfrak{R}}_{12}} \tag{2.65}$$

$$\hat{\breve{z}} = \frac{\breve{\boldsymbol{\mathfrak{R}}}_{12} \times \breve{\boldsymbol{\mathfrak{V}}}_{12}}{||\breve{\boldsymbol{\mathfrak{R}}}_{12} \times \breve{\boldsymbol{\mathfrak{V}}}_{12}||} \tag{2.66}$$

$$\hat{\breve{y}} = \hat{\breve{z}} \times \hat{\breve{x}} \tag{2.67}$$

where, $\breve{\boldsymbol{\mathfrak{R}}}_{12}$ and $\breve{\boldsymbol{\mathfrak{V}}}_{12}$ represent the instantaneous relative ephemeris position and velocity vectors for P_2 relative to P_1 in J2000 coordinates, and $\breve{\mathfrak{R}}_{12}$ is the magnitude of this instantaneous relative position vector. These unit vectors facilitate the assembly of the instantaneous direction cosine matrix:

$$\breve{C}_{R2I} = [\hat{\breve{x}}, \hat{\breve{y}}, \hat{\breve{z}}] = \begin{bmatrix} \breve{C}_{11} & \breve{C}_{11} & \breve{C}_{11} \\ \breve{C}_{21} & \breve{C}_{22} & \breve{C}_{23} \\ \breve{C}_{31} & \breve{C}_{32} & \breve{C}_{33} \end{bmatrix} \tag{2.68}$$

Since the rotation from one frame to another occurs at an instantaneous configuration of the two primaries, the nondimensional CR3BP states with respect to the primary ($\boldsymbol{r}_r$ and $\boldsymbol{v}_r = \boldsymbol{v}$), are dimensionalized via the following instantaneous characteristic quantities prior to the rotation

$$\breve{L}^* = \breve{\mathfrak{R}}_{12}, \tag{2.69}$$

$$\breve{T}^* = \sqrt{\frac{\breve{L}^*}{\tilde{G}M_1 + \tilde{G}M_2}}, \text{ and} \tag{2.70}$$

$$\breve{V}^* = \frac{\breve{L}^*}{\breve{T}^*} \tag{2.71}$$

to yield the dimensional CR3BP states $\boldsymbol{\chi}_{dim,R} = [\boldsymbol{R}_R, \dot{\boldsymbol{R}}_R] = [\mathcal{X}_{\mathfrak{p}}, \mathcal{Y}, \mathcal{Z}, \dot{\mathcal{X}}, \dot{\mathcal{Y}}, \dot{\mathcal{Z}}]^T$. The dimensional J2000 position states with respect to the primary are defined such that $\boldsymbol{\mathfrak{R}}_I = [\mathscr{X}, \mathscr{Y}, \mathscr{Z}]^T$, and are constructed by multiplying $\boldsymbol{R}_R$ by the instantaneous direction cosine matrix $\breve{C}_{R2I}$ evaluated at a specified epoch:

$$\boldsymbol{\mathfrak{R}}_I = \breve{C}_{R2I}\ \boldsymbol{R}_R \tag{2.72}$$

The rotation of the velocity states is computed by exploiting the kinematical relationships as introduced in Eqn. (2.60) [65].

$${}^I\dot{\boldsymbol{R}}_R = (\dot{\mathcal{X}} - \breve{\phi}\mathcal{Y})\hat{\breve{x}} + (\dot{\mathcal{Y}} + \breve{\phi}\mathcal{X}_{\mathfrak{p}})\hat{\breve{y}} + \dot{\mathcal{Z}}\hat{\breve{z}} \tag{2.73}$$

In this problem formulation, the angular velocity ($\breve{\phi}$) of the rotating frame with respect to the J2000 frame is not constant; it is a function of the epoch and, thus, also a function of the relative states of the two primaries at a given instant in time:

$${}^I\breve{\omega}^R = \breve{\phi} = \frac{||\breve{\boldsymbol{\mathfrak{R}}}_{12} \times \breve{\boldsymbol{\mathfrak{V}}}_{12}||}{\breve{\mathfrak{R}}_{12}^2} \tag{2.74}$$

The velocity vector as observed in the inertial frame $\dot{\boldsymbol{\mathfrak{R}}}_I$ is thus computed as:

$$\dot{\boldsymbol{\mathfrak{R}}}_I = \begin{bmatrix} \dot{\breve{C}}_{R2I} & \breve{C}_{R2I} \end{bmatrix} \begin{bmatrix} \boldsymbol{R}_R \\ \dot{\boldsymbol{R}}_R \end{bmatrix} \tag{2.75}$$

where,

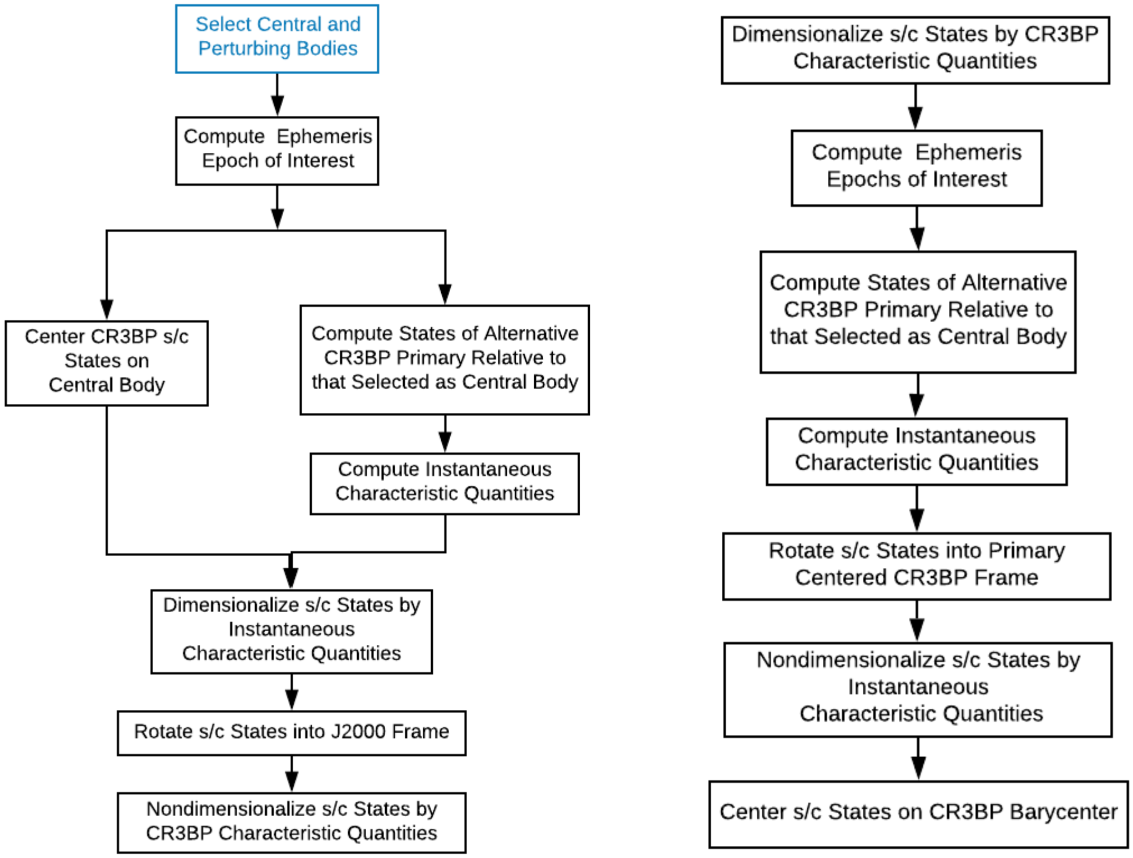
$$\dot{\breve{C}}_{R2I} = \begin{bmatrix} \breve{\phi}\breve{C}_{12} & -\breve{\phi}\breve{C}_{11} & 0 \\ \breve{\phi}\breve{C}_{22} & -\breve{\phi}\breve{C}_{21} & 0 \\ \breve{\phi}\breve{C}_{32} & -\breve{\phi}\breve{C}_{31} & 0 \end{bmatrix} \tag{2.76}$$

The submatrices $\dot{\breve{C}}_{R2I}$ and $\breve{C}_{R2I}$ are combined to execute the following rotation transformation for the entire state vector comprised of both the position and velocity states:

$$\begin{bmatrix} \boldsymbol{\mathfrak{R}}_I \\ \dot{\boldsymbol{\mathfrak{R}}}_I \end{bmatrix} = \underbrace{\begin{bmatrix} \breve{C}_{R2I} & 0_{3\times3} \\ \dot{\breve{C}}_{R2I} & \breve{C}_{R2I} \end{bmatrix}}_{D\breve{C}M_{R2I}} \begin{bmatrix} \boldsymbol{R}_R \\ \dot{\boldsymbol{R}}_R \end{bmatrix} \tag{2.77}$$

This direction cosine matrix '$D\breve{C}M_{R2I}$' transforms the spacecraft states from the rotating frame to the J2000 inertial frame. The reverse, producing the rotating frame states from the inertial states is achieved by exploiting the inverse of this matrix '$D\breve{C}M_{R2I}^{-1} = D\breve{C}M_{I2R}$'. As previously noted, it is appropriate to nondimensionalize the J2000 inertial states prior to propagation to mitigate numerical sensitivities introduced during numerical propagation; these quantities are nondimensionalized employing the CR3BP characteristic quantities introduced in Table 2.1. The aforementioned steps to rotate the states between the CR3BP and J2000 frames are summarized in Fig. 2.7. Additionally, the unit vectors reflecting the thrusting direction are simply rotated by $\breve{C}_{R2I}$:

$$\hat{\mathrm{u}}_I = \breve{C}_{R2I}\ \hat{\boldsymbol{u}}_R \tag{2.78}$$


Select Central and Perturbing Bodies
Compute Ephemeris Epoch of Interest
Center CR3BP s/c States on Central Body
Compute States of Alternative CR3BP Primary Relative to that Selected as Central Body
Compute Instantaneous Characteristic Quantities
Dimensionalize s/c States by Instantaneous Characteristic Quantities
Rotate s/c States into J2000 Frame
Nondimensionalize s/c States by CR3BP Characteristic Quantities
Dimensionalize s/c States by CR3BP Characteristic Quantities
Compute Ephemeris Epochs of Interest
Compute States of Alternative CR3BP Primary Relative to that Selected as Central Body
Compute Instantaneous Characteristic Quantities
Rotate s/c States into Primary Centered CR3BP Frame
Nondimensionalize s/c States by Instantaneous Characteristic Quantities
Center s/c States on CR3BP Barycenter

(a) CR3BP → Ephemeris

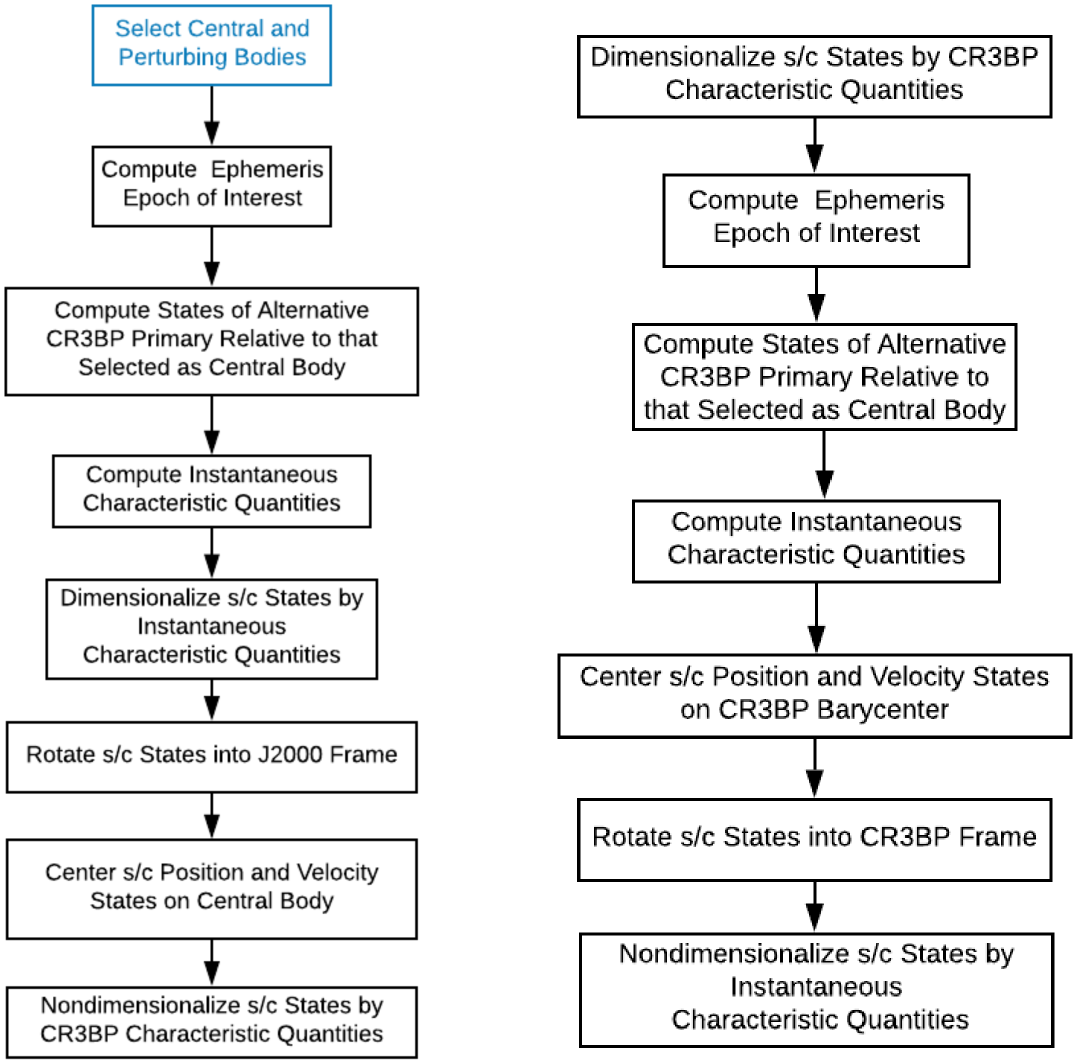
(b) Ephemeris → CR3BP

Figure 2.7: Steps to transform a set of s/c position and velocity states between the CR3BP rotating frame and J2000 inertial frame — Approach 1. The steps in the flow-diagrams are followed to rotate each state between the two frames of interest; the choice of the central and perturbing bodies remains unaltered during this process. Although all the perturbing bodies are identified, only the ephemeris states of the two primaries composing the CR3BP of interest are exploited to conduct the s/c state rotation between the frames.

Also, the nondimensional s/c mass and thrust history values in the rotating frame are transitioned without modification to the J2000 frame as they are merely scalar magnitudes.

Implementation Approach 2

The second approach applying a coordinate transformation between the CR3BP states and the J2000 frame is undertaken by modifying the order of steps that are implemented in Approach 1 [72]. This alternative approach incorporating the re-ordered steps is illustrated in Fig. 2.8; the modification leads to different J2000 inertial states, and thus, varied trajectories in the ephemeris regime from those constructed via Approach 1. Recall that, when expressed in the rotating frame, the s/c velocity with respect to the central body is equivalent to the s/c velocity relative to the barycenter. Since the s/c states are always shifted to/from the barycenter via the rotating frame coordinates in Approach 1, only the s/c position states between the barycenter and central body are altered to support the frame transformation. In the alternative Approach 2, the s/c states are shifted to/from the central body in the J2000 frame coordinates; thus, both the s/c position and velocity states are shifted between the central body and the barycenter to support the frame transformation.


Select Central and Perturbing Bodies
Compute Ephemeris Epoch of Interest
Compute States of Alternative CR3BP Primary Relative to that Selected as Central Body
Compute Instantaneous Characteristic Quantities
Dimensionalize s/c States by Instantaneous Characteristic Quantities
Rotate s/c States into J2000 Frame
Center s/c Position and Velocity States on Central Body
Nondimensionalize s/c States by CR3BP Characteristic Quantities
Dimensionalize s/c States by CR3BP Characteristic Quantities
Compute Ephemeris Epoch of Interest
Compute States of Alternative CR3BP Primary Relative to that Selected as Central Body
Compute Instantaneous Characteristic Quantities
Center s/c Position and Velocity States on CR3BP Barycenter
Rotate s/c States into CR3BP Frame
Nondimensionalize s/c States by Instantaneous Characteristic Quantities

(a) CR3BP → Ephemeris

(b) Ephemeris → CR3BP

Figure 2.8: Steps to transform a set of s/c position and velocity states between the CR3BP rotating frame and J2000 inertial frame — Approach 2. The steps in the flow-diagrams are followed to rotate each state between the two frames of interest; the choice of the central and perturbing bodies remains unaltered during this process. Although all the perturbing bodies are identified, only the ephemeris states of the two primaries composing the CR3BP of interest are exploited to conduct the s/c state rotation between the frames.

3. NUMERICAL METHODS AND PROCESSES

Mission design benefits from the exploitation of natural flows in a dynamical system to aid in the generation of reference solutions. Dynamical Systems Theory (DST) is leveraged to suppress and excite different modes of motion, while numerical corrections processes are incorporated to meet desired constraints and boundary conditions to satisfy mission requirements. Varied processes are explored that integrate numerical corrections into direct, indirect, and hybrid optimization schemes and facilitate locally optimal solutions for chemical and low-thrust engines.

3.1 Considerations in the Natural CR3BP

The motion of a spacecraft (s/c) operating in a nonlinear CR3B regime is offered via Eqn. (2.29). However, these equations possess no analytical solution and initial conditions are specified to initiate numerical propagation. So, linear variational analysis about a desired reference is exploited to offer insights into the nature and stability of the flow in its vicinity and, thus, facilitate the selection of desirable initial conditions.

3.1.1 Variational Equations of Motion

Consider a reference trajectory defined by initial conditions $\boldsymbol{\chi}^*(\tau_0) = \boldsymbol{\chi}_0^* = [x_0, y_0, z, \dot{x}_0, \dot{y}_0, \dot{z}_0]^T$. The propagation of these conditions via Eqn. (2.29) results in a time history for the reference path, i.e.,

$$\dot{\boldsymbol{\chi}}^*(\tau) = \boldsymbol{f}(\boldsymbol{\chi}^*, \tau) \tag{3.1}$$

Then, the behavior due to small perturbations $\delta\boldsymbol{\chi} = [\delta x_0, \delta y_0, \delta z_0, \delta \dot{x}_0, \delta \dot{y}_0, \delta \dot{z}_0]^T$ relative to the reference solution, $\boldsymbol{\chi}_0^*$, is modeled by a Taylor series expansion about this reference as:

$$\dot{\boldsymbol{\chi}}^*(\tau) + \delta\dot{\boldsymbol{\chi}} \approx \boldsymbol{f}(\boldsymbol{\chi}^*, \tau) + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\chi}}\bigg|_{\boldsymbol{\chi}^*} \delta\boldsymbol{\chi} + \cancel{H.O.T.}^{\nearrow neglect} \tag{3.2}$$

With only small perturbations involved, the “Higher Order Terms" (H.O.T) approach *zero* as $\boldsymbol{\chi} \to \boldsymbol{\chi}^*$ and are, thus, neglected. Furthermore, it is concluded from Eqns. (3.1) and (3.2) that a first order approximation is modeled as

$$\delta\dot{\boldsymbol{\chi}} \approx \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\chi}}\bigg|_{\boldsymbol{\chi}^*} \delta\boldsymbol{\chi} \tag{3.3}$$

where the perturbation is rewritten as $\delta\boldsymbol{\chi} = [\xi, \eta, \varsigma, \dot{\xi}, \dot{\eta}, \dot{\varsigma}]^T$ for convenience. Consequentially, the linear variational equations are derived as:

$$\ddot{\xi} - 2\dot{\eta} = U_{xx}^*\xi + U_{xy}^*\eta + U_{xz}^*\varsigma \tag{3.4a}$$

$$\ddot{\eta} + 2\dot{\xi} = U_{yx}^*\xi + U_{yy}^*\eta + U_{yz}^*\varsigma \tag{3.4b}$$

$$\ddot{\varsigma} = U_{zx}^*\xi + U_{zy}^*\eta + U_{zz}^*\varsigma. \tag{3.4c}$$

The symbol U_{ij}^* represents the 2^{nd} partial derivatives of the pseudo-potential term (U^*) with respect to the position variables along the reference solution. The partials are evaluated along the reference and, are derived as follows:

$$U_{xx}^* = 1 - \frac{(1-\mu)}{r_{13}^3} - \frac{\mu}{r_{23}^3} + \frac{3(1-\mu)(x+\mu)^2}{r_{13}^5} + \frac{3\mu(x-1+\mu)^2}{r_{23}^5} \tag{3.5a}$$

$$U_{yy}^* = 1 - \frac{(1-\mu)}{r_{13}^3} - \frac{\mu}{r_{23}^3} + \frac{3(1-\mu)y^2}{r_{13}^5} + \frac{3\mu y^2}{r_{23}^5} \tag{3.5b}$$

$$U_{zz}^* = 1 - \frac{(1-\mu)}{r_{13}^3} - \frac{\mu}{r_{23}^3} + \frac{3(1-\mu)z^2}{r_{13}^5} + \frac{3\mu z^2}{r_{23}^5} \tag{3.5c}$$

$$U_{xy}^* = \frac{3(1-\mu)(x+\mu)y}{r_{13}^5} + \frac{3\mu(x-1+\mu)y}{r_{23}^5} \tag{3.5d}$$

$$U_{xz}^* = \frac{3(1-\mu)(x+\mu)z}{r_{13}^5} + \frac{3\mu(x-1+\mu)z}{r_{23}^5} \tag{3.5e}$$

$$U_{yz}^* = \frac{3(1-\mu)yz}{r_{13}^5} + \frac{3\mu yz}{r_{23}^5} \tag{3.5f}$$

$$U_{yx}^* = U_{xy}^* \tag{3.5g}$$

$$U_{zx}^* = U_{xz}^* \tag{3.5h}$$

$$U_{zy}^* = U_{yz}^* \tag{3.5i}$$

In state space form, the system of equations in Eqns. (3.4) is represented by:

$$\delta\dot{\boldsymbol{\chi}} = \mathbf{A}(\tau)\delta\boldsymbol{\chi} \tag{3.6}$$

where $\mathbf{A}(\tau)$ is a Jacobian matrix consisting of the partials $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\chi}}$ evaluated on the reference $\partial\boldsymbol{\chi}^*$:

$$\mathbf{A}(\tau) = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\chi}}\Big|_{\boldsymbol{\chi}^*} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ U_{xx}^* & U_{xy}^* & U_{xz}^* & 0 & 2 & 0 \\ U_{xy}^* & U_{yy}^* & U_{yz}^* & -2 & 0 & 0 \\ U_{xz}^* & U_{yz}^* & U_{zz}^* & 0 & 0 & 0 \end{bmatrix} \tag{3.7}$$

Employing an equilibrium solution as the reference, the reference is then a constant. Since all these partials are time invariant for a reference at $\boldsymbol{\chi}_{L_i}$, the $\mathbf{A}(\tau)$ matrix computed with regards to the equilibrium solutions is constant for all time.

3.1.2 Motion Near the Equilibrium Solutions

The periodic solutions associated with the equilibrium Lagrange points have facilitated many missions including ARTEMIS and LISA Pathfinder, allowing these spacecraft to implement minimized propellant transits and long-term term stays in their vicinity, respectively. Such distinct capabilities may be tapped by exploiting different modes of stability at these Lagrange points.

Defining Stability

Stability evaluation is accomplished with a variety of approaches; Poisson, Hill, Laplace, Lagrange, and Lyapunov stability are all valid, for example. In this investigation, Lyapunov stability suffices to assess the behavior of an object subjected to perturbations in the vicinity of the original solution over extended periods of time. Marquez [73] states that the equilibrium solution ($\boldsymbol{\chi}_e$) is stable if it satisfies the following conditions for a small bound set surrounding the equilibrium point such that $\epsilon > 0$,

$$\begin{gathered} \exists \delta = \delta(\epsilon) > 0, \\ \|\boldsymbol{\chi}_0 - \boldsymbol{\chi}_e\| < \delta \Longrightarrow \|\boldsymbol{\chi}_\tau - \boldsymbol{\chi}_e\| < \epsilon \ \ \forall \tau \geq \tau_0 \end{gathered} \tag{3.8}$$

where δ is the bound set on the allowable small initial perturbation, $\boldsymbol{\chi}_0$ are the initial conditions in position and velocity in the vicinity of $\boldsymbol{\chi}_e$ after being perturbed from the equilibrium solution, and $\boldsymbol{\chi}_\tau$ is the state of the propagation after time τ. Furthermore, the solution is convergent if:

$$\|\boldsymbol{\chi}_0 - \boldsymbol{\chi}_e\| < \delta \Longrightarrow \lim_{\tau \to \infty} \boldsymbol{\chi}_\tau = \boldsymbol{\chi}_e \tag{3.9}$$

If an equilibrium solution is stable and convergent, it is termed *asymptotically* stable. If a perturbed solution departs the vicinity of the equilibrium point after time τ, then it is denoted an unstable equilibrium point. Such stability characteristics for the equilibrium solutions is further explored by perturbing their states and integrating the resulting conditions in the nonlinear system. A more efficient approach is a linear stability analysis by evaluating the eigenvalues of the **A** matrix derived in Eqn. (3.7). The eigenvalue characteristics reflect unique types of behavior for a perturbed reference solution in its vicinity as $\tau \to \infty$. Dawkins [74] and Rutgers [75] supply the following definitions:

- *Nonstable / Marginally Stable* The solution is marginally stable, in a linear sense, if all the eigenvalues are purely imaginary. Such eigenvalues introduce oscillations into the behavior of a solution. It is not possible to predict the stability of the nonlinear system in this case. Therefore, higher-order stability analysis are necessary.

- *Asymptotically Stable*: The solution is asymptotically stable if all the eigenvalues are real and negative, or complex and their real parts are negative. Such solutions are expected to return to the equilibrium point if perturbed.

- *Unstable*: The solution is unstable if even one eigenvalue is real and positive, or complex but with positive real parts. These solutions eventually depart the vicinity of the reference solution.

These definitions of stability aid in the assessment of the behavior of motion in the vicinity of equilibrium and periodic solutions in the CR3BP.

Stability - Equilibrium Solutions

The stability of an equilibrium solution is assessed by the behavior near the Lagrange point when a perturbation is introduced into the spacecraft state. Since all the Lagrange points are planar (Section 2.2.2), the cross-terms U^*_{xz} and U^*_{yz} evaluate to *zero* in Eqn. (3.4); thus, the in-plane and out-of-plane motions are decoupled. The out-of-plane stability near the Lagrange points is determined from the solutions to Eqn. (3.4c) and are of a simple harmonic form:

$$\varsigma = C_1 cos(\varkappa\tau) + C_2 sin(\varkappa\tau) \tag{3.10}$$

where the frequency is defined, $\varkappa = \sqrt{|U^*_{zz}|_{L_i}}$. The in-plane stability assessment is initiated by reducing the $\mathbf{A}$ matrix to include only the U^* terms comprised of in-plane components, as :

$$\mathbf{A}(\tau)_{\text{planar}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ U^*_{xx} & U^*_{xy} & 0 & 2 \\ U^*_{xy} & U^*_{yy} & -2 & 0 \end{bmatrix} \tag{3.11}$$

Next, solving for the determinant of the $\mathbf{A}_{\text{planar}}$ matrix yields the following characteristic equation:

$$\lambda^4 + (4 - U^*_{xx} - U^*_{yy})\lambda^2 + (U^*_{xx}U^*_{yy} - U^*_{yx}U^*_{xy}) = 0 \tag{3.12}$$

At the collinear libration points, the terms U^*_{xy} and U^*_{yx} are equal to *zero* since $y_{L_i} = 0$. Thus, Eqn. (3.12) reduces to:

$$\lambda^4 + (4 - U^*_{xx} - U^*_{yy})\lambda^2 + U^*_{xx}U^*_{yy} = 0 \tag{3.13}$$

Letting $\Lambda = \lambda^2$, Eqn. (3.13) is represented as a quadratic:

$$\Lambda^2 + 2\beta_1\Lambda - \beta_2^2 = 0 \tag{3.14}$$

where,

$$\beta_1 = 2 - \frac{U^*_{xx} + U^*_{yy}}{2} \tag{3.15a}$$

$$\beta_2^2 = -U^*_{xx}U^*_{yy} \tag{3.15b}$$

$$\Lambda_1 = -\beta_1 + \sqrt{\beta_1^2 + \beta_2^2} \tag{3.15c}$$

$$\Lambda_2 = -\beta_1 - \sqrt{\beta_1^2 + \beta_2^2}. \tag{3.15d}$$

At the collinear points, $U^*_{xx} > 0$ and $U^*_{yy} < 0$, resulting in $\beta_2 > 0$, and consequently, $\Lambda_1 > 0$ and $\Lambda_2 < 0$. Also, since

$$\lambda_{1,2} = \pm\sqrt{\Lambda_1}, \tag{3.16a}$$

$$\lambda_{3,4} = \pm\sqrt{\Lambda_2}, \tag{3.16b}$$

both $\lambda_{3,4}$ are purely imaginary and the two real eigenvalues possess opposing signs, $\lambda_1 > 0$ and $\lambda_2 < 0$. Thus, oscillatory, unstable and asymptotically stable behavior are observed in the plane near the collinear Lagrange points. A similar process facilitates the investigation of the eigenvalues associated with the equilateral libration points — for which all six eigenvalues are computed as purely imaginary. So, the motion in the vicinity of these points is marginally stable in a linear sense, but could still depart when nonlinearities are introduced.

Manifolds — Equilibrium Solutions

The stability information obtained via the eigenstructure of the **A** matrix offers insights into the nature of the flow near equilibrium solutions [5, 76, 77]. The general solution to Eqn. (3.6) is expressed in exponential form as:

$$\delta\boldsymbol{\chi}(\tau) = e^{\mathbf{A}(\tau-\tau_0)}\delta\boldsymbol{\chi}_0. \tag{3.17}$$

Decomposition of the matrix $\mathbf{A} = \mathbf{SDS}^{-1}$ and Eqn. (3.17) is rewritten as

$$\delta\boldsymbol{\chi}(\tau) = \mathbf{S}e^{\mathbf{D}(\tau-\tau_0)}\mathbf{S}^{-1}\delta\boldsymbol{\chi}_0, \tag{3.18}$$

where $\mathbf{S}$ is a matrix consisting of the eigenvectors $\boldsymbol{\rho}_i$ of the $\mathbf{A}$ matrix along its columns, and $\mathbf{D}$ is a diagonal matrix comprised of the corresponding eigenvalues. For $\mathcal{I}$ distinct eigenvalues, the expansion of Eqn. (3.18) evolves an initial condition $\delta\boldsymbol{\chi}_0$ at time τ as:

$$\delta\boldsymbol{\chi}(\tau) = \sum_{i=1}^{\mathcal{I}} \succ_i e^{\lambda_i(\tau-\tau_0)}\boldsymbol{\rho}_i \tag{3.19}$$

where $\succ_i$ are coefficients exploited to suppress/excite desired modes of behavior. These values are related to the initial condition $\boldsymbol{\chi}_0$ at $\tau_0 = 0$ via:

$$\delta\boldsymbol{\chi}_0 = \sum_{i=1}^{\mathcal{I}} \succ_i \boldsymbol{\rho}_i \tag{3.20}$$

So, $\boldsymbol{\succ}$ is extracted as:

$$\boldsymbol{\succ} = \mathfrak{S}^{-1}\delta\boldsymbol{\chi}_0 \tag{3.21}$$

where $\mathfrak{S} = [\boldsymbol{\rho}_1, \boldsymbol{\rho}_2, \boldsymbol{\rho}_3, \boldsymbol{\rho}_4, \boldsymbol{\rho}_5, \boldsymbol{\rho}_6]$ is a matrix comprised of the eigenvectors. An understanding of the coefficients $\succ_i$ facilitates adjustments to excite the desired mode(s), and is obtained by revisiting the stability characteristics associated with the equilibrium solutions and exploiting *manifold* theory.

All the previously derived real and imaginary eigenvalues for the collinear Lagrange points occur in reciprocal pairs and complex eigenvalues are also delivered as pairs of complex conjugates. Furthermore, the eigenvectors $\boldsymbol{\rho}_i^s$ associated with n_s eigenvalues such that $Re\{\lambda\} < 0$, $\boldsymbol{\rho}_i^u$ with n_u eigenvalues defined by $Re\{\lambda\} > 0$ and

$\boldsymbol{\rho}_i^c$ with n_c eigenvalues that include no real part, i.e., $Re\{\lambda\} = 0$, span the following subspaces [77]:

$$E^s = span\{\boldsymbol{\rho}_i^s\}_{i=1}^{n_s} \text{ stable subspace} \tag{3.22}$$

$$E^u = span\{\boldsymbol{\rho}_i^u\}_{i=1}^{n_u} \text{ unstable subspace} \tag{3.23}$$

$$E^c = span\{\boldsymbol{\rho}_i^c\}_{i=1}^{n_c} \text{ center subspace} \tag{3.24}$$

These three subspaces, comprised of linearly independent eigenvectors constitute the space $\mathbb{R}^n$ with dimension $n_s + n_u + n_c = \text{rank}(\mathbf{A})$. The subspaces E^s, E^u and E^c are invariant under $e^{\lambda_i \tau}$ and, so, any state originating within any one subspace remains in that subspace for all time τ. Guckenheimer and Holmes [77] highlight the *Stable* and *Center* manifold theorems to aid in understanding the “nonlinear analogues" for these linear system subspaces as follows:

Stable Manifold Theorem *Suppose that* $\dot{\boldsymbol{\chi}} = \boldsymbol{f}(\boldsymbol{\chi})$ *has a hyperbolic equilibrium point,* $\boldsymbol{\chi}_{L_i}$*. Then there exist local stable and unstable manifolds,* $\mathcal{W}^s_{loc}(\boldsymbol{\chi}_{L_i})$*,* $\mathcal{W}^u_{loc}(\boldsymbol{\chi}_{L_i})$*, of the same dimension,* n_s*,* n_u *, as the eigenspaces,* E^s*,* E^u*, of the linearized system (Eqn. (3.6)), and tangent to* E^s *and* E^u *at* $\boldsymbol{\chi}_{L_i}$*.* $\mathcal{W}^s_{loc}(\boldsymbol{\chi}_{L_i})$*,* $\mathcal{W}^u_{loc}(\boldsymbol{\chi}_{L_i})$ *are smooth, as is function* $\boldsymbol{f}$*.*

Given the tangency conditions, the stable manifold $\mathcal{W}^s_{loc}$ approaches the libration point asymptotically in forward time and departs it asymptotically in reverse time. The converse is true for the unstable manifold, $\mathcal{W}^u_{loc}$. These local manifolds may be extended globally by propagating $\mathcal{W}^s_{loc}$ backwards in time and $\mathcal{W}^u_{loc}$ forward in time to generate $\mathcal{W}^s$ and $\mathcal{W}^u$, respectively. Both manifold types possess the same dimension as their linear system counterparts and are time invariant, i.e, points evolving on a manifold remain in the same subspace for all time. They are also unique, which restricts the manifold of a libration point from intersecting itself, or intersecting a manifold of the same type belonging to another libration point. An unstable and stable manifold may, however, intersect. Guckenheimer and Holmes [77] describe the center manifold theorem as follows:

Center Manifold Theorem *Let $\boldsymbol{f}$ be a C^r vector field on $\mathbb{R}^n$ vanishing at the origin so that $\boldsymbol{f}(\boldsymbol{\chi}_{L_i}) = \mathbf{0}$, and let $\mathbf{A} = \mathbf{D}\boldsymbol{f}(\boldsymbol{\chi}_{L_i})$. The matrix $\mathbf{A}$ may be divided into its stable, center, and unstable parts, $n_s, n_c,$ and n_u, respectively, with*

$$Real(\lambda)\begin{Bmatrix} < 0 & \lambda \in n_s \\ = 0 & \lambda \in n_c \\ > 0 & \lambda \in n_u \end{Bmatrix} \tag{3.25}$$

Let the generalized eigenspaces be $E^s, E^c, andE^u$, respectively. Then there exist C^r stable and unstable invariant manifolds, $\mathcal{W}^s$ and $\mathcal{W}^u$ tangent to E^s, and E^u at $\boldsymbol{\chi}_{L_i}$, and a C^{r-1} center manifold, $\mathcal{W}_c$, tangent to E^c at $\boldsymbol{\chi}_{L_i}$. The manifolds $\mathcal{W}^s$, $\mathcal{W}^u$, and $\mathcal{W}^c$ are all invariant for the flow $\boldsymbol{f}$. The stable and unstable manifolds are unique, but the center manifold need not be. If $\boldsymbol{f}$ is C^∞, then there exists a C^r center manifold for any $r < \infty$.

The center manifold is associated with eigenvalues that are purely complex, providing an opportunity for periodic and quasi-periodic motion that potentially facilitate parking-orbit conditions to exist in the vicinity of the libration points. Also, the stable and unstable manifolds flow towards/from the libration points of interest and, thus, enable naturally efficient transfer opportunities. Therefore, it is important to understand departure/arrival in the vicinity of these solutions.

A particular behavior is excited by specifying a non-zero value for $\succ_i$ to step in the eigenvector direction associated with an eigenvalue, λ_i, of interest; undesired modes are suppressed by setting the relevant $\succ_i$ value to *zero*. A suitably small step in the direction of the stable $\hat{\rho}_s = \frac{\boldsymbol{\rho}_s}{|\boldsymbol{\rho}_s|}$ or unstable $\hat{\rho}_u = \frac{\boldsymbol{\rho}_u}{|\boldsymbol{\rho}_u|}$ directions supplies the initial conditions in all six-dimensions to initiate the propagation of the global manifolds $\mathcal{W}^s$ and $\mathcal{W}^u$, respectively. An example of these initial conditions propagated in reverse time to establish $\mathcal{W}^s$ for L_1 and those propagated in forward time to establish $\mathcal{W}^u$ are displayed in Figs. 3.1; the local manifolds are also highlighted.

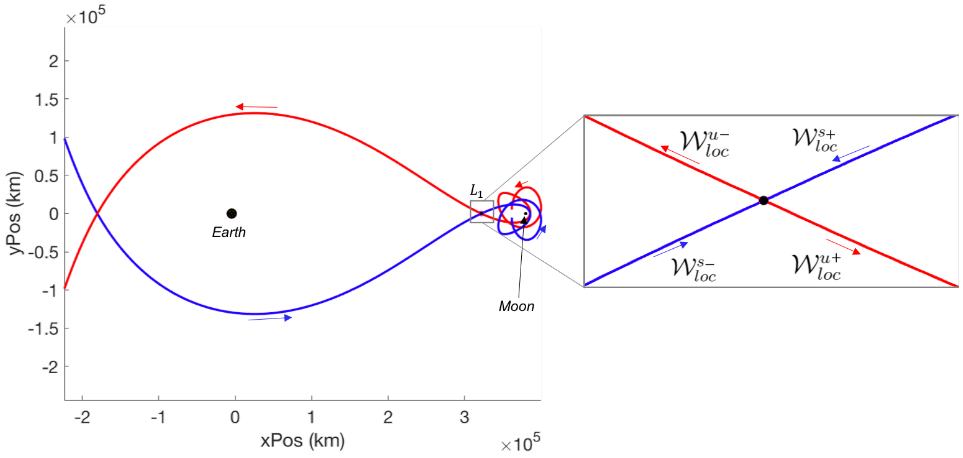


Figure 3.1: Global stable and unstable manifold arcs in the planar problem associated with L_1 in the Earth-Moon system, with a close-up view of the associated local manifold directions. The arrows display the direction of flow in forward time.

Szebehely [3] details the process of generating initial conditions in the linear system to yield periodic solutions centered around the libration points, where $\succ_s = \succ_u = 0$ and $\succ_c \neq 0$. An example of a resulting linear elliptic solution about L_1 is illustrated in Fig. 3.2 — the blue orbit is the initial guess for a nonlinear Lyapunov orbit with a period of 12 days. However, when the initial conditions developed in the linear system are propagated in the full nonlinear model, the result is not closed. Such periodic orbits are useful for varied mission applications, so constructing a periodic orbit in the nonlinear system is a key capability. Numerical techniques are developed to accomplish the essential corrections strategies.

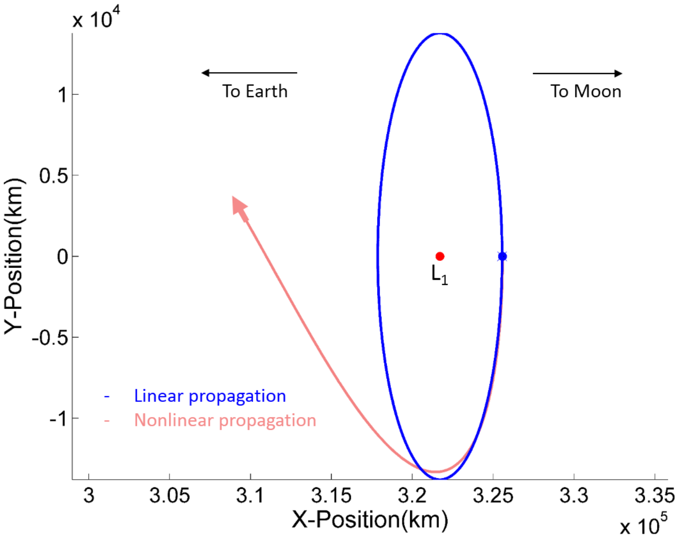


Figure 3.2: Divergence during nonlinear propagation from the intended periodic orbit constructed via linear analysis

3.1.3 The State Transition Matrix

A first step in adjusting any trajectory states is an understanding of the relationship between variations in the initial and terminal conditions along a reference path. The appropriate derivations to develop this understanding are incorporated from Stuart [78]. Consider Fig. 3.3, where the baseline spacecraft path is the trajectory evolved from the initial states, $\boldsymbol{\chi}^*_{\tau 0} = \boldsymbol{\chi}^*_0 = [x_0\ y_0\ z_0\ \dot{x}_0\ \dot{y}_0\ \dot{z}_0]^T$. The natural terminal states at time τ are $\boldsymbol{\chi}^*(\tau) = \boldsymbol{\chi}^* = [x_\tau\ y_\tau\ z_\tau\ \dot{x}_\tau\ \dot{y}_\tau\ \dot{z}_\tau]^T = [x\ y\ z\ \dot{x}\ \dot{y}\ \dot{z}]^T$. A variation in the initial conditions, $\delta\boldsymbol{\chi}_0 = [\delta x_0\ \delta y_0\ \delta z_0\ \delta\dot{x}_0\ \delta\dot{y}_0\ \delta\dot{z}_0]^T$, generates a trajectory that deviates from the nominal target conditions at the final time by $\delta\boldsymbol{\chi}_\tau = \delta\boldsymbol{\chi} = [\delta x\ \delta y\ \delta z\ \delta\dot{x}\ \delta\dot{y}\ \delta\dot{z}]^T$. The initial deviation is evaluated as:

$$\delta\boldsymbol{\chi}_0 = \boldsymbol{\chi}^p_0 - \boldsymbol{\chi}^*_0 \tag{3.26}$$

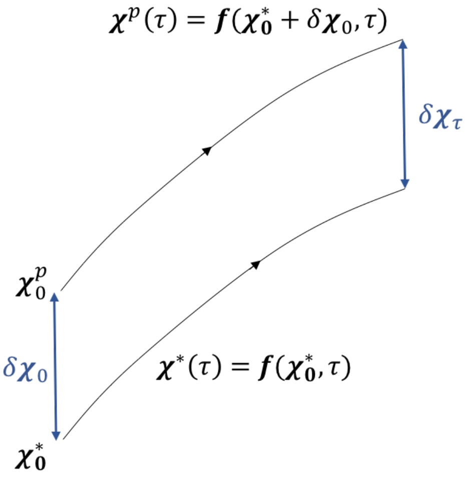

$\boldsymbol{\chi}^p(\tau) = \boldsymbol{f}(\boldsymbol{\chi}_0^* + \delta\boldsymbol{\chi}_0, \tau)$
$\delta\boldsymbol{\chi}_\tau$
$\boldsymbol{\chi}_0^p$
$\delta\boldsymbol{\chi}_0$
$\boldsymbol{\chi}^*(\tau) = \boldsymbol{f}(\boldsymbol{\chi}_0^*, \tau)$
$\boldsymbol{\chi}_0^*$

Figure 3.3: Reference $\boldsymbol{\chi}^*(\tau)$ and perturbed $\boldsymbol{\chi}^p(\tau)$ trajectories for fixed time propagations.

where the superscript p implies the perturbed path. The final deviation is then:

$$\begin{aligned} \delta\boldsymbol{\chi} &= \boldsymbol{\chi}^p(\boldsymbol{\chi}_0^p, \tau) - \boldsymbol{\chi}^*(\boldsymbol{\chi}_0^*, \tau) \\ &= \boldsymbol{\chi}^p(\boldsymbol{\chi}_0^* + \delta\boldsymbol{\chi}_0, \tau) - \boldsymbol{\chi}^*(\boldsymbol{\chi}_0^*, \tau) \end{aligned} \tag{3.27}$$

For a small initial deviation $\delta\boldsymbol{\chi}_0$, expanding $\boldsymbol{\chi}^p(\boldsymbol{\chi}_0^* + \delta\boldsymbol{\chi}_0, \tau)$ and linearizing about the reference trajectory yields:

$$\delta\boldsymbol{\chi} = \frac{\partial \boldsymbol{\chi}^p}{\partial \boldsymbol{\chi}_0}\delta\boldsymbol{\chi}_0 \tag{3.28}$$

The matrix, $\frac{\partial \boldsymbol{\chi}^p}{\partial \boldsymbol{\chi}_0}$ is rewritten as $\frac{\partial \boldsymbol{\chi}}{\partial \boldsymbol{\chi}_0}$ without the superscript p for convenience, and is denoted the State Transition Matrix (STM), $\boldsymbol{\Phi}(\tau, \tau_0)$:

$$\boldsymbol{\Phi}(\tau,\tau_0) = \begin{bmatrix} \frac{\partial x}{\partial x_0} & \frac{\partial x}{\partial y_0} & \frac{\partial x}{\partial z_0} & \frac{\partial x}{\partial \dot{x}_0} & \frac{\partial x}{\partial \dot{y}_0} & \frac{\partial x}{\partial \dot{z}_0} \\ \frac{\partial y}{\partial x_0} & \frac{\partial y}{\partial y_0} & \frac{\partial y}{\partial z_0} & \frac{\partial y}{\partial \dot{x}_0} & \frac{\partial y}{\partial \dot{y}_0} & \frac{\partial y}{\partial \dot{z}_0} \\ \frac{\partial z}{\partial x_0} & \frac{\partial z}{\partial y_0} & \frac{\partial z}{\partial z_0} & \frac{\partial z}{\partial \dot{x}_0} & \frac{\partial z}{\partial \dot{y}_0} & \frac{\partial z}{\partial \dot{z}_0} \\ \frac{\partial \dot{x}}{\partial x_0} & \frac{\partial \dot{x}}{\partial y_0} & \frac{\partial \dot{x}}{\partial z_0} & \frac{\partial \dot{x}}{\partial \dot{x}_0} & \frac{\partial \dot{x}}{\partial \dot{y}_0} & \frac{\partial \dot{x}}{\partial \dot{z}_0} \\ \frac{\partial \dot{y}}{\partial x_0} & \frac{\partial \dot{y}}{\partial y_0} & \frac{\partial \dot{y}}{\partial z_0} & \frac{\partial \dot{y}}{\partial \dot{x}_0} & \frac{\partial \dot{y}}{\partial \dot{y}_0} & \frac{\partial \dot{y}}{\partial \dot{z}_0} \\ \frac{\partial \dot{z}}{\partial x_0} & \frac{\partial \dot{z}}{\partial y_0} & \frac{\partial \dot{z}}{\partial z_0} & \frac{\partial \dot{z}}{\partial \dot{x}_0} & \frac{\partial \dot{z}}{\partial \dot{y}_0} & \frac{\partial \dot{z}}{\partial \dot{z}_0} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_{\boldsymbol{rr}} & \boldsymbol{\Phi}_{\boldsymbol{rv}} \\ \boldsymbol{\Phi}_{\boldsymbol{vr}} & \boldsymbol{\Phi}_{\boldsymbol{vv}} \end{bmatrix} \tag{3.29}$$

where the subscripts $\boldsymbol{rr}$, $\boldsymbol{rv}$, $\boldsymbol{vr}$ and $\boldsymbol{vv}$ identify the position and velocity dependencies within the STM. The following properties are also defined:

$$\boldsymbol{\Phi}(\tau_2, \tau_0) = \boldsymbol{\Phi}(\tau_2, \tau_1)\boldsymbol{\Phi}(\tau_1, \tau_0) \tag{3.30}$$

$$\boldsymbol{\Phi}(\tau_0, \tau) = \boldsymbol{\Phi}^{-1}(\tau, \tau_0) \tag{3.31}$$

$$det(\boldsymbol{\Phi}) = 1 \tag{3.32}$$

Note that at time $\tau = \tau_0$, the left and right sides of Eqn. (3.28) are equal. Therefore, $\boldsymbol{\Phi}_0 = I$, the identity matrix. The critical assumption is that the perturbed solution is very similar to the reference solution (i.e., *the perturbation is very small*) and, thus, linear theory is valid to derive the STM, essentially a linear map to advance the initial deviation. The STM is also termed a 'sensitivity matrix' because it is a measure of the sensitivity of the final variations to variations in the initial states. Note that the states vary with time along the perturbed trajectory and, so, the STM is evolved with time as well. Thus, a first order differential representation for the STM, $\dot{\boldsymbol{\Phi}}(\tau, \tau_0)$ is determined as follows:

$$\begin{aligned} \dot{\boldsymbol{\Phi}}(\tau, \tau_0) &= \frac{d}{d\tau}\left(\frac{\partial \boldsymbol{\chi}}{\partial \boldsymbol{\chi}_0}\right) = \frac{\partial}{\partial \boldsymbol{\chi}_0}\left(\frac{d\boldsymbol{\chi}}{d\tau}\right) = \frac{\partial \dot{\boldsymbol{\chi}}}{\partial \boldsymbol{\chi}}\frac{\partial \boldsymbol{\chi}}{\partial \boldsymbol{\chi}_0} \\ \dot{\boldsymbol{\Phi}}(\tau, \tau_0) &= \mathbf{A}(\tau)\boldsymbol{\Phi}(\tau, \tau 0) \end{aligned} \tag{3.33}$$

and is integrated along with the equations of motion to evaluate $\boldsymbol{\Phi}(\tau, \tau_0)$. Note that the term $\frac{\partial \dot{\boldsymbol{\chi}}}{\partial \boldsymbol{\chi}}$ is the Jacobian matrix $\mathbf{A}$, also seen in Eqn. (3.7). Recall that $\mathbf{A}$ is a constant matrix when the reference is a time-invariant equilibrium point, but for general trajectory arcs such as a reference trajectory, $\mathbf{A}(\tau)$ is not constant due to the time-varying nature of the states associated with the reference path. The components of the STM are incorporated into a numerical corrections scheme to meet the desired trajectory conditions.

3.1.4 Numerical Shooting Methods

A trajectory is constructed as a solution to the differential equations of motion that govern the flow associated with the dynamical system. Specific conditions along the trajectory are met by posing the problem as a two point boundary value problem (2PBVP) for the six equivalent 1^{st} order Ordinary Differential Equations (ODEs) associated with Eqns. (2.29). For the evolution of some unknown trajectory path, assume that a set of end conditions is specified. Then, the problem is defined as *targeting* end conditions while allowing certain initial conditions to vary. Hence, the term *shooting* is used to describe such a technique, that is, adjusting the initial conditions to successfully intercept the desired target. Such methods involve three crucial steps to formulate an iterative procedure - *targeting* the desired states, *correcting* for any deviations, and *updating* the states that are allowed to vary. The iterative process is implemented using a Newton-Raphson technique and aided by knowledge of the sensitivities via the STM.

Multi-Variable Newton-Raphson Scheme

To formulate an iterative process, a vector $\boldsymbol{X}$ is defined that consists of k design parameters labelled *free variables*; an $\mathbb{m}$-vector of desired target conditions is also formulated as a set of scalar *constraints* in the form $\boldsymbol{F}(\boldsymbol{X}) = \mathbf{0}$. In practice, solutions satisfying $\boldsymbol{F}(\boldsymbol{X}) < \epsilon$ are sought, where ϵ is a user-defined tolerance that accommodates numerical errors that accrue during the convergence process. The *free variable* design vector $\boldsymbol{X}$ incorporates items such as spacecraft states and any control variables, and/or time-of-flight (TOF) and is defined as:

$$\boldsymbol{X} = \begin{Bmatrix} X_1 \\ X_2 \\ \vdots \\ X_k \end{Bmatrix} \tag{3.34}$$

The *constraint vector* $\boldsymbol{F}(\boldsymbol{X})$ incorporates a variety of desired conditions, e.g., spacecraft states, altitudes, turn rates as well as flight path angles, and is written as:

$$\boldsymbol{F}(\boldsymbol{X}) = \begin{Bmatrix} F_1(\boldsymbol{X}) \\ F_2(\boldsymbol{X}) \\ \vdots \\ F_{\mathbbm{m}}(\boldsymbol{X}) \end{Bmatrix} = \mathbf{0}. \tag{3.35}$$

The search for an $\boldsymbol{X}^*$ that satisfies $\boldsymbol{F}(\boldsymbol{X}^*) = \mathbf{0}$ is initiated by seeding an iterative scheme with an initial guess, $\boldsymbol{X}_0$. A Taylor series expansion for $\boldsymbol{F}(\boldsymbol{X})$ about $\boldsymbol{X}_0$ results in a 1^{st} order approximation such that

$$\boldsymbol{F}(\boldsymbol{X}) \approx \boldsymbol{F}(\boldsymbol{X}_0) + \mathfrak{J}(\boldsymbol{X}_0) \cdot (\boldsymbol{X} - \boldsymbol{X}_0) = \mathbf{0}$$

or, more generally as:

$$\boldsymbol{F}(\boldsymbol{X}_j) + \mathfrak{J}(\boldsymbol{X}_j) \cdot (\boldsymbol{X}_{j+1} - \boldsymbol{X}_j) = \mathbf{0} \tag{3.36}$$

where, j represents the current iteration and $\mathfrak{J}(\boldsymbol{X}_j)$ is an $\mathbbm{m} \times k$ Jacobian matrix, the partial derivatives of the constraints with respect to the free variables evaluated at $\boldsymbol{X}_j$:

$$\mathfrak{J}(\boldsymbol{X}_j) = \frac{\partial \boldsymbol{F}(\boldsymbol{X}_j)}{\partial \boldsymbol{X}_j} = \begin{bmatrix} \frac{\partial F_1}{\partial X_1} & \frac{\partial F_1}{\partial X_2} & \cdots & \frac{\partial F_1}{\partial X_k} \\ \frac{\partial F_2}{\partial X_1} & \frac{\partial F_2}{\partial X_2} & \cdots & \frac{\partial F_2}{\partial X_k} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial F_{\mathbbm{m}}}{\partial X_1} & \frac{\partial F_{\mathbbm{m}}}{\partial X_2} & \cdots & \frac{\partial F_{\mathbbm{m}}}{\partial X_k} \end{bmatrix}. \tag{3.37}$$

Note that $\|\boldsymbol{F}(\boldsymbol{X}_{j+1})\| < \|\boldsymbol{F}(\boldsymbol{X}_j)\|$ as long as the $(j+1)^{th}$ iteration moves the generated solution $(\boldsymbol{X}_{j+1})$ closer to a true solution for the system. Also, the convergence is quadratic if the initial guesses supplied to the iteration process is sufficiently close to a true solution. The *update* for $\boldsymbol{X}_{j+1}$ is determined by rearranging Eqn. (3.36). However, the form of this rearrangement, and the approach to iteratively update

$\boldsymbol{X}_{j+1}$, may require modification depending on the total number of free variables that are available and the constraints that are specified. If the number of constraints equals the number of variables ($\mathbb{m} = k$), and a solution exists, it is a unique $\boldsymbol{X}^*$ that satisfies $\boldsymbol{F}(\boldsymbol{X}^*) = \boldsymbol{0}$. In this case, the square Jacobian matrix is mathematically inverted, and the guess for the $(j+1)^{th}$ iteration is evaluated as:

$$\boldsymbol{X}_{j+1} = \boldsymbol{X}_j - \mathfrak{J}(\boldsymbol{X}_j)^{-1}\boldsymbol{F}(\boldsymbol{X}_j) \tag{3.38}$$

In the case where the number of free variables exceeds the number of constraints ($k > \mathbb{m}$), infinite solutions exist. Here, a *minimum norm* approach is employed that minimizes the difference between $\boldsymbol{X}_{j+1}$ and $\boldsymbol{X}_j$ and, therefore, seeks the closest solution to $\boldsymbol{X}_j$. This approach offers an advantage by providing a converged solution that is similar to the initial guess. The minimum-norm update equation is formulated as:

$$\boldsymbol{X}_{j+1} = \boldsymbol{X}_j - \mathfrak{J}(\boldsymbol{X}_j)^T[\mathfrak{J}(\boldsymbol{X}_j)\mathfrak{J}(\boldsymbol{X}_j)^T]^{-1}\boldsymbol{F}(\boldsymbol{X}_j) \tag{3.39}$$

For the numerical process, the iteration proceeds only if two conditions are satisfied:

- The norm of the constraint vector does not satisfy the desired tolerance, that is, $\boldsymbol{F}(\boldsymbol{X}_j) > \epsilon$
- The iteration count does not exceed a pre-defined limit, i.e., #Iterations $\leq$ #Iterations$_{max}$.

The vector $\boldsymbol{X}_{j+1}$ produced by the update equation then becomes the initial guess for the next iteration.

Single Shooting Scheme

In one straightforward scenario, a single shooting scheme is employed to target conditions at the end of a single arc after propagating the initial conditions $\boldsymbol{\chi}_0$ for a specified time, τ. An example of varying the initial velocity components $\dot{\boldsymbol{r}}_0 = [\dot{x}_0, \dot{y}_0, \dot{z}_0]^T$ along the 'baseline' trajectory, to target specific position states $\boldsymbol{r}_{\mathbb{d}\tau} = [x_{\mathbb{d}}, y_{\mathbb{d}}, z_{\mathbb{d}}]^T$ at the final time τ on the 'desired' trajectory is illustrated in Fig. 3.4. For this scenario, the design vector $\boldsymbol{X}$ is formulated as:

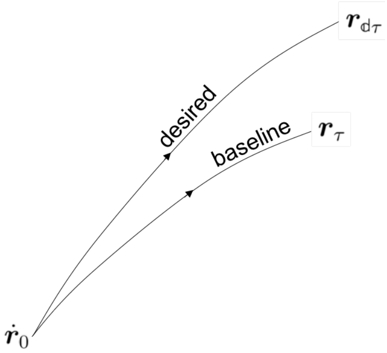


Figure 3.4: Single shooting to meet desired target position states $\boldsymbol{r}_{\mathbb{d}\tau}$.

$$\boldsymbol{X} = \begin{bmatrix} \dot{x}_0 \\ \dot{y}_0 \\ \dot{z}_0 \end{bmatrix} \tag{3.40}$$

The constraint vector, $\boldsymbol{F}$, is constructed as the difference between the conditions at the end of a propagation segment at time τ (denoted by subscript, τ) and the desired conditions (denoted by subscript, $\mathbb{d}$):

$$\boldsymbol{F}(\boldsymbol{X}) = \begin{bmatrix} x_\tau - x_{\mathbb{d}} \\ y_\tau - y_{\mathbb{d}} \\ z_\tau - z_{\mathbb{d}} \end{bmatrix} \tag{3.41}$$

The Jacobian matrix is evaluated, per Eqn. (3.37), as:

$$\mathfrak{J}(\boldsymbol{X}) = \frac{\partial \boldsymbol{F}(\boldsymbol{X})}{\partial \boldsymbol{X}} = \begin{bmatrix} \frac{\partial(x_\tau - x_{\mathbb{d}})}{\partial \dot{x}_0} & \frac{\partial(x_\tau - x_{\mathbb{d}})}{\partial \dot{y}_0} & \frac{\partial(x_\tau - x_{\mathbb{d}})}{\partial \dot{z}_0} \\ \frac{\partial(y_\tau - y_{\mathbb{d}})}{\partial \dot{x}_0} & \frac{\partial(y_\tau - y_{\mathbb{d}})}{\partial \dot{y}_0} & \frac{\partial(y_\tau - y_{\mathbb{d}})}{\partial \dot{z}_0} \\ \frac{\partial(z_\tau - z_{\mathbb{d}})}{\partial \dot{x}_0} & \frac{\partial(z_\tau - z_{\mathbb{d}})}{\partial \dot{y}_0} & \frac{\partial(z_\tau - z_{\mathbb{d}})}{\partial \dot{z}_0} \end{bmatrix} \tag{3.42}$$

Since the desired conditions at the end of the trajectory segment $\boldsymbol{\chi}_{\tau}^{\dagger}$, are user-defined and not dependent on the variable initial conditions in the design vector $\boldsymbol{X}$, the partials $\frac{\partial x_{\mathbb{d}}}{\partial \boldsymbol{X}} = \frac{\partial y_{\mathbb{d}}}{\partial \boldsymbol{X}} = \frac{\partial z_{\mathbb{d}}}{\partial \boldsymbol{X}} = \boldsymbol{0}$. So, the Jacobian matrix simplifies to:

$$\mathfrak{J}(\boldsymbol{X}) = \frac{\partial \boldsymbol{F}(\boldsymbol{X})}{\partial \boldsymbol{X}} = \begin{bmatrix} \frac{\partial x_\tau}{\partial \dot{x}_0} & \frac{\partial x_\tau}{\partial \dot{y}_0} & \frac{\partial x_\tau}{\partial \dot{z}_0} \\ \frac{\partial y_\tau}{\partial \dot{x}_0} & \frac{\partial y_\tau}{\partial \dot{y}_0} & \frac{\partial y_\tau}{\partial \dot{z}_0} \\ \frac{\partial z_\tau}{\partial \dot{x}_0} & \frac{\partial z_\tau}{\partial \dot{y}_0} & \frac{\partial z_\tau}{\partial \dot{z}_0} \end{bmatrix} = \boldsymbol{\Phi}_{\boldsymbol{rv}}(\tau, \tau 0) \tag{3.43}$$

where the elements of the $\mathfrak{J}$ matrix are recognized as a subset of the full 6 × 6 STM associated with the system of equations, $\boldsymbol{\Phi}(\tau, \tau_0)_{6\times 6}$. This scenario is an example of *fixed time* single shooting, since the time of propagation is not included as a design variable. Since the number of design variables equals the number of constraints, the update equation to compute the initial conditions is formulated as:

$$\begin{bmatrix} \dot{x}_0 \\ \dot{y}_0 \\ \dot{z}_0 \end{bmatrix}_{j+1} = \begin{bmatrix} \dot{x}_0 \\ \dot{y}_0 \\ \dot{z}_0 \end{bmatrix}_{j} - \mathfrak{J}^{-1} \begin{bmatrix} x_\tau - x_{\mathbb{d}} \\ y_\tau - y_{\mathbb{d}} \\ z_\tau - z_{\mathbb{d}} \end{bmatrix}_{j} \tag{3.44}$$

The single shooting scheme is effective in delivering converged solutions for straightforward numerical corrections scenarios; multiple shooting schemes are sought to address those that are more complex in nature.

Multiple Shooting Scheme

Multiple shooting introduces a versatile targeting strategy that accommodates numerical corrections for complex and lengthy trajectories, and even challenging constraints by distributing the associated numerical sensitivities among multiple nodes along the baseline. Figure 3.5 illustrates the arcs, each with origin at $\boldsymbol{\chi}_{i,0}$, that are propagated from the nodes i $(i = 1 \ldots \mathbb{n})$ for a duration ϖ_i. The evolution of these propagated arcs is dictated by the EOMs that reflect the dynamical and spacecraft environment incorporated during the corrections process. Similar to the single shooting

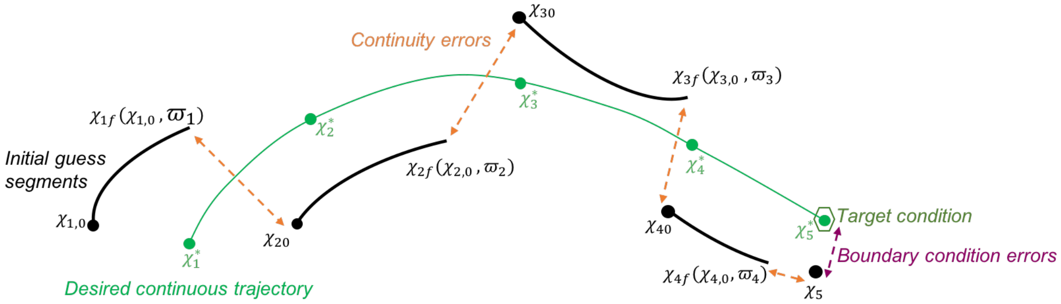

χ_{30}
Continuity errors
$\chi_{3f}(\chi_{3,0}, \varpi_3)$
$\chi_{1f}(\chi_{1,0}, \varpi_1)$
χ_2^*
χ_3^*
Initial guess segments
$\chi_{2f}(\chi_{2,0}, \varpi_2)$
χ_4^*
$\chi_{1,0}$
Target condition
χ_{40}
χ_5^*
χ_{20}
χ_1^*
Boundary condition errors
$\chi_{4f}(\chi_{4,0}, \varpi_4)$
Desired continuous trajectory
χ_5

Figure 3.5: Numerical multiple shooting schematic in the CR3BP. Modified from Schlei [79]

formulation, the aim of the multiple shooting process is the variation in conditions at the nodes via the design vector $\boldsymbol{X}$ to meet the specified boundary conditions, subject to constraints $\boldsymbol{F}(\boldsymbol{X})$:

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{\chi}_1 \\ \vdots \\ \boldsymbol{\chi}_{\mathbb{n}} \\ \varpi_1 \\ \vdots \\ \varpi_{\mathbb{n}-1} \end{bmatrix} \qquad \boldsymbol{F}(\boldsymbol{X}) = \begin{bmatrix} \boldsymbol{\chi}_{1,f} - \boldsymbol{\chi}_{2,0} \\ \vdots \\ \boldsymbol{\chi}_{\mathbb{n}-1,f} - \boldsymbol{\chi}_{\mathbb{n}} \\ \boldsymbol{\chi}_{\mathbb{n}} - \boldsymbol{\chi}_{\mathbb{d}} \end{bmatrix} = \boldsymbol{0} \tag{3.45}$$

The design vector in Eqn. (3.45) incorporates the states and propagation durations associated with each node along a discretized trajectory. The constraint vector includes state continuity constraints along the trajectory, and also specific boundary conditions to meet the state $\boldsymbol{\chi}_{\mathbb{d}}$. Note that the last node $\mathbb{n}$, is a free-floating state incorporated into the design vector to meet specific boundary conditions $\boldsymbol{\chi}_{\mathbb{d}}$; it is, therefore, not propagated. Also, since the CR3BP is a time-invariant system, there is no requirement to include epoch continuity constraints. Computation of the Jacobian matrix $\mathfrak{J}$ is intrinsic to the update of the initial conditions (Eqn. (3.39)) to satisfy the constraints imposed along the trajectory. The full Jacobian matrix $\mathfrak{J}$ is computed by combining sub-matrices that account for the state continuity and boundary condition constraints.

Jacobian Derived to Meet State Continuity Constraints, $\mathfrak{J}_S$

The magnitude of the violation of the state continuity constraints subject to variations in the initial conditions is captured via the sensitivity matrix $\mathfrak{J}_S$. This matrix is formulated by computing the partial derivatives of the state continuity constraints with respect to the constituents of the design vector:

$$\mathfrak{J}_S = \frac{\partial(\boldsymbol{\chi}_{i,f} - \boldsymbol{\chi}_{i+1,0})}{\partial \boldsymbol{X}} \tag{3.46}$$

Since $\boldsymbol{\chi}_{i,f}$ is a function of the states $\boldsymbol{\chi}_{i,0}$ and the propagation duration ϖ_i associated with the originating node i, Eqn. (3.46) is further expanded to yield the partial derivative as a function of explicit and implicit relationships [65]:

$$\mathfrak{J}_S = \frac{\partial\left(\overbrace{\boldsymbol{\chi}_{i,f}(\boldsymbol{\chi}_{i,0}, \varpi_i)}^{\text{implicit}} - \overbrace{\boldsymbol{\chi}_{i+1,0}}^{\text{explicit}}\right)}{\partial \boldsymbol{X}} \tag{3.47}$$

Note that the states at the end of a propagated arc only depend on the conditions at its originating node and no other nodes along the full trajectory. So, the explicit relationship is captured via:

$$\mathfrak{J}_{S,\boldsymbol{\chi}_{i+1,0}} = -\frac{\partial \boldsymbol{\chi}_{i+1,0}}{\partial \boldsymbol{\chi}_{i+1,0}} = -\boldsymbol{I}_{6\times 6} \tag{3.48}$$

where, $\boldsymbol{I}$ is the identity matrix. The partial derivatives associated with the implicit relationships are computed by decomposing the derivations to accommodate the contributions from each of the independent variables $\boldsymbol{\chi}_{i,0}$ as well as ϖ_i. The impact of varying specific state elements in $\boldsymbol{\chi}_{i,0}$ on those at the end of the propagated arc $\boldsymbol{\chi}_{i,f}$, is captured via the STM, $\mathbf{\Phi}$:

$$\mathfrak{J}_{S,\boldsymbol{\chi}_{i,0}} = \frac{\partial(\boldsymbol{\chi}_{i,f}(\boldsymbol{\chi}_{i,0}, \varpi_i))}{\partial \boldsymbol{\chi}_{i,0}} = \mathbf{\Phi}(\mathbb{t}_i + \varpi_i, \mathbb{t}_i) \tag{3.49}$$

where, $\mathbb{t}_i$ represents the time at the i^{th} node. The independently determined propagation duration per originating node (ϖ_i) results in different terminal conditions across an arc. The associated sensitivity is captured via

$$\mathfrak{J}_{S,\varpi_i} = \frac{\partial(\boldsymbol{\chi}_{i,f}(\boldsymbol{\chi}_{i,0}, \varpi_i))}{\partial \varpi_i} = \dot{\boldsymbol{\chi}}_{i,f}, \tag{3.50}$$

i.e., the time rate of change of the terminal states of an arc.

Jacobian Derived to Meet Specific Terminal Boundary Conditions, $\mathfrak{J}_B$

Often, a spacecraft is required to meet specific states at its destination. The sensitivities for such a constraint in response to variations in the states of the unpropagated node $\mathbb{n}$, is captured via:

$$\mathfrak{J}_{B,\boldsymbol{\chi}_{\mathbb{n}}} = \frac{\partial(\boldsymbol{\chi}_{\mathbb{n}} - \boldsymbol{\chi}_{\mathbb{d}})}{\partial \boldsymbol{\chi}_{\mathbb{n}}} = \boldsymbol{I}_{6\times6} \tag{3.51}$$

Note that $\frac{\partial \boldsymbol{\chi}_{\mathbb{d}}}{\partial \boldsymbol{\chi}_{\mathbb{n}}} = \mathbf{0}$ as $\boldsymbol{\chi}_{\mathbb{d}}$ are pre-specified constant values.

Jacobian Matrix Combined

The complete Jacobian matrix $\mathfrak{J}$ is formulated by combining the values computed for the sub-matrices $\mathfrak{J}_S$ and $\mathfrak{J}_B$. The matrix $\mathfrak{J}$ thus computes the derivatives of the constraints $\boldsymbol{F}(\boldsymbol{X})$ with respect to the design variables $\boldsymbol{X}$ in Eqn. (3.45), and is compiled as:

$$\mathfrak{J} = \begin{array}{cc} \begin{array}{ccccccccc} \chi_1 & \chi_2 & \chi_3 & \dots & \chi_{\mathbb{n}} & \varpi_1 & \varpi_2 & \dots & \varpi_{\mathbb{n}-1} \end{array} & \\ \left[\begin{array}{ccccccccc} \boldsymbol{\Phi}_1 & -\boldsymbol{I}_{6\times6} & \boldsymbol{0}_{6\times6} & \dots & \boldsymbol{0}_{6\times6} & \dot{\chi}_{1,f} & \boldsymbol{0}_{6\times1} & \dots & \boldsymbol{0}_{6\times1} \\ \boldsymbol{0}_{6\times6} & \boldsymbol{\Phi}_2 & -\boldsymbol{I}_{6\times6} & \dots & \vdots & \boldsymbol{0}_{6\times1} & \dot{\chi}_{2,f} & \dots & \boldsymbol{0}_{6\times1} \\ \vdots & \ddots & \ddots & \ddots & \boldsymbol{0}_{6\times6} & \vdots & \ddots & \ddots & \vdots \\ \boldsymbol{0}_{6\times6} & \dots & \boldsymbol{0}_{6\times6} & \boldsymbol{\Phi}_{\mathbb{n}-1} & -\boldsymbol{I}_{6\times6} & \boldsymbol{0}_{6\times1} & \dots & \dots & \dot{\chi}_{\mathbb{n}-1,f} \\ \boldsymbol{0}_{6\times6} & \dots & \dots & \dots & \boldsymbol{I}_{6\times6} & \boldsymbol{0}_{6\times1} & \dots & \dots & \boldsymbol{0}_{6\times1} \end{array}\right] & \begin{array}{c} \chi_{1,f} - \chi_{2,0} \\ \chi_{2,f} - \chi_{3,0} \\ \vdots \\ \chi_{\mathbb{n}-1,f} - \chi_{\mathbb{n}} \\ \chi_{\mathbb{n}} - \chi_{\mathbb{d}} \end{array} \end{array} \tag{3.52}$$

The rows of the matrix correspond to the constituents of the constraint vector $\boldsymbol{F}(\boldsymbol{X})$ and the columns correspond to the components of the design vector $\boldsymbol{X}$; these values are highlighted in magenta in Eqn. (3.52). This matrix is incorporated into the Newton-Raphson update as previously explored, to solve for the design variables in $\boldsymbol{X}$ to meet the constraints in $\boldsymbol{F}(\boldsymbol{X}^*)$.

3.1.5 Applications for Numerical Shooting Methods — Periodic Orbits

Recall from Section 3.1.2 that an appropriate choice of initial conditions relative to the equilibrium points excites periodic motion in the equilibrium point vicinity. The true CR3BP dynamics, however, distort a linear propagation of the initial conditions due to the nonlinearities present in the system (e.g., Fig. 3.2). So, it is useful to exploit numerical shooting schemes to target the desired conditions to enforce periodicity in the nonlinear regime.

Prior to initiating a targeting scheme, however, note two important observations. The CR3BP EOMs (i) are invariant in time reversal, and (ii) satisfy the *Mirror Theorem* as examined in Roy and Ovenden [80] under certain conditions:

Mirror Theorem

If $\mathbb{N}$ point masses are acted upon by their mutual gravitational forces only, and at a certain epoch each radius vector from the center of mass of the system is perpendicular to every velocity vector, then the orbit of each mass after that epoch is a mirror image of its orbit prior to that epoch.

Consistent with this theorem, periodic orbits the CR3BP possess two mirror configuration locations (perpendicular crossings) along the $\hat{x} - \hat{z}$ plane, rendering them symmetric across this plane. Therefore, it is advantageous to use the states at these locations as initial conditions for purposes of targeting periodic orbits. As a consequence, only the conditions at the half-period along the orbit are targeted, after which the converged states are *mirrored* across the plane of symmetry. Again, critical to this approach is the fact that just as $[x, y, z, \dot{x}, \dot{y}, \dot{z}]$ satisfy the EOMs in Eqn. (2.29) in forward time τ, by symmetry, the states $[x, -y, z, -\dot{x}, \dot{y}, -\dot{z}]$ satisfy them in reverse time $-\tau$.

Computing a Planar Lyapunov Orbit

A single shooting scheme suffices to demonstrate the targeting process to compute a simple periodic solution, namely a planar L_1 Lyapunov orbit. By definition, a planar periodic orbit that is approximately centered at L_1 is defined such that $y_0 = \dot{x}_0 = y_f = \dot{x}_f = 0$ at the two perpendicular crossings on the $\hat{x} - \hat{z}$ plane. So, the only free variables available are $[x_0, \dot{y}_0, \varpi_0]^T$ to then target the desired half-period conditions, $y_f = \dot{x}_f = 0$. If all three free variables are involved in the iterative process to generate a Lyapunov orbit, then the total number of free variables $k >$ total number of constraints $(\mathbb{m})$ and the minimum-norm solution is activated. However, to illustrate the ability to deliberately influence the nature of the final solution, only the free variables $[\dot{y}_0, \varpi]^T$ are employed in this example — allowing the departure position to remain fixed. The unique solution is extracted for the variable-time single shooting example by defining the design vector ($\boldsymbol{X}$) and constraint vectors ($\boldsymbol{F}(\boldsymbol{X})$) as follows:

$$\boldsymbol{X} = \begin{bmatrix} \dot{y}_0 \\ \varpi_h \end{bmatrix} \qquad \boldsymbol{F} = \begin{bmatrix} y_f - y_{\mathbb{d}} \\ \dot{x}_f - \dot{x}_{\mathbb{d}} \end{bmatrix} \tag{3.53}$$

where, ϖ_h is the propagation time that also reflects the half period of the orbit. Constructing the partial derivatives of the constraints with respect to the design variables yields the Jacobian $\mathfrak{J}$:

$$\mathfrak{J}(\boldsymbol{X}) = \frac{\partial \boldsymbol{F}(\boldsymbol{X})}{\partial \boldsymbol{X}} = \begin{bmatrix} \frac{\partial y_f}{\partial \dot{y}_0} & \frac{\partial y_f}{\partial \varpi_h} \\ \frac{\partial \dot{x}_f}{\partial \dot{y}_0} & \frac{\partial \dot{x}_f}{\partial \varpi_h} \end{bmatrix} = \begin{bmatrix} \frac{\partial y_f}{\partial \dot{y}_0} & \dot{y}_f \\ \frac{\partial \dot{x}_f}{\partial \dot{y}_0} & \ddot{x}_f \end{bmatrix} \tag{3.54}$$

As discussed in section 3.1.4, $\boldsymbol{X}_{j+1}$ is updated as:

$$\begin{bmatrix} \dot{y}_0 \\ \varpi_h \end{bmatrix}_{j+1} = \begin{bmatrix} \dot{y}_0 \\ \varpi_h \end{bmatrix}_{j} - \begin{bmatrix} \frac{\partial y_f}{\partial \dot{y}_0} & \dot{y}_f \\ \frac{\partial \dot{x}_f}{\partial \dot{y}_0} & \ddot{x}_f \end{bmatrix}_{j}^{-1} \begin{bmatrix} y_f - y_{\mathbb{d}} \\ \dot{x}_f - \dot{x}_{\mathbb{d}} \end{bmatrix}_{j} \tag{3.55}$$

The linear initial guess for the Lyapunov orbit is plotted in blue in Fig. 3.6. The initial conditions that emerge from the corrections process to enable a perpendicular crossing at the half-period are then propagated for the full period of the orbit to produce the Lyapunov orbit in pink in Fig. 3.6.

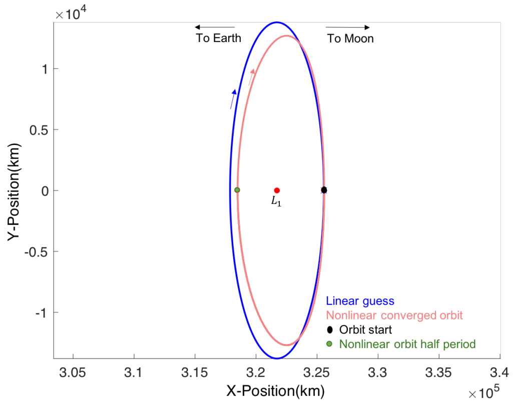


Figure 3.6: Comparison between the initial guess in the linear model, and the nonlinear converged Lyapunov orbit.

Computing a Spatial Halo Orbit

The numerical targeting approach to create periodic Lyapunov orbits is extended to generate spatial periodic solutions as well, e.g., halo orbits. By definition, these periodic orbits are constructed such that $y_0 = \dot{x}_0 = \dot{z}_0 = y_f = \dot{x}_f = \dot{z}_f = 0$ at the perpendicular crossings on the $\hat{x} - \hat{z}$ plane. The update equation to determine a unique solution is formulated by varying the x_0 value while fixing z_0

$$\begin{bmatrix} x_0 \\ \dot{y}_0 \\ \varpi_h \end{bmatrix}_{j+1} = \begin{bmatrix} x_0 \\ \dot{y}_0 \\ \varpi_h \end{bmatrix}_{j} - \begin{bmatrix} \frac{\partial y_f}{\partial x_0} & \frac{\partial y_f}{\partial \dot{y}_0} & \dot{y}_f \\ \frac{\partial \dot{x}_f}{\partial x_0} & \frac{\partial \dot{x}_f}{\partial \dot{y}_0} & \ddot{x}_f \\ \frac{\partial \dot{z}_f}{\partial x_0} & \frac{\partial \dot{z}_f}{\partial \dot{y}_0} & \ddot{z}_f \end{bmatrix}_{j}^{-1} \begin{bmatrix} y_f - y_{\mathbb{d}} \\ \dot{x}_f - \dot{x}_{\mathbb{d}} \\ \dot{z}_f - \dot{z}_{\mathbb{d}} \end{bmatrix}_{j} \tag{3.56}$$

or alternatively, i.e., varying the z_0 component while maintaining x_0:

$$\begin{bmatrix} z_0 \\ \dot{y}_0 \\ \varpi_h \end{bmatrix}_{j+1} = \begin{bmatrix} z_0 \\ \dot{y}_0 \\ \varpi_h \end{bmatrix}_{j} - \begin{bmatrix} \frac{\partial y_f}{\partial z_0} & \frac{\partial y_f}{\partial \dot{y}_0} & \dot{y}_f \\ \frac{\partial \dot{x}_f}{\partial z_0} & \frac{\partial \dot{x}_f}{\partial \dot{y}_0} & \ddot{x}_f \\ \frac{\partial \dot{z}_f}{\partial z_0} & \frac{\partial \dot{z}_f}{\partial \dot{y}_0} & \ddot{z}_f \end{bmatrix}_{j}^{-1} \begin{bmatrix} y_f - y_{\mathbb{d}} \\ \dot{x}_f - \dot{x}_{\mathbb{d}} \\ \dot{z}_f - \dot{z}_{\mathbb{d}} \end{bmatrix}_{j} \tag{3.57}$$

Simultaneous variations in both x_0 and z_0 result in a non-square Jacobian matrix and incorporation of the minimum-norm solution instead. Depending on the numerical formulation, varying both these states simultaneously could result in the process converging on a different nearby solution in the CR3BP, one that exhibits characteristics unlike the desired halo orbit. An illustration of a halo orbit appears in Fig. 3.7.

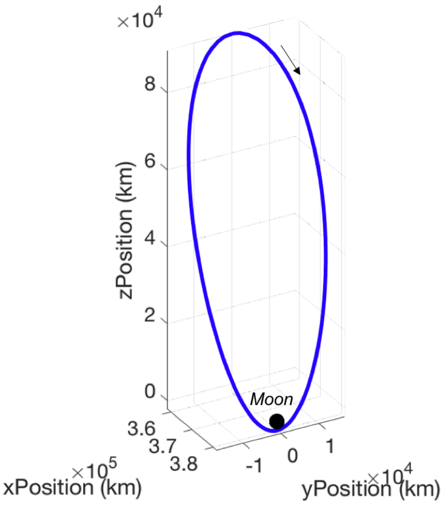


Figure 3.7: Example of a halo orbit. JC ≈ 2.9887.

3.1.6 Numerical Continuation — Generating Families of Periodic Orbits

Section 3.1.5 details the process of numerical corrections to produce planar and spatial periodic orbits in the CR3BP. However, these orbits can be classified into *families* of orbits with similar motion and characteristics in the nonlinear regime by meeting the same set of constraints. *Numerical continuation* is adapted to evolve a periodic orbit family from a particular solution. Two continuation approaches are explored in this investigation: (i) *Natural Parameter Continuation* that exploits known physical quantities such as position or velocity of the periodic orbit, and (ii) *Pseudo-Arclength Continuation* that exploits a mathematical construct representing the direction that is tangent to the family. The continuation of a family for either scheme is built on the knowledge associated with an a priori construction of at least one family member.

Natural Parameter Continuation

A natural parameter in this analysis is defined as any measurable quantity with a known physical meaning along a trajectory solution such as position, velocity, or an energy-related term such as the Jacobi Constant. As an example, a new member of the Lyapunov orbit family is generated by shifting a small step Δx along the $\pm\hat{x}$ direction from a perpendicular crossing of a known and converged Lyapunov orbit. For initial conditions on a converged Lyapunov orbit,

$$\boldsymbol{\chi}_{\text{Lyap0}} = [x_{\text{Lyap0}}, y_{\text{Lyap0}} = 0, \dot{x}_{\text{Lyap0}} = 0, \dot{y}_{\text{Lyap0}}]^T \quad \text{(planar case)}, \tag{3.58}$$

a new member in the Lyapunov family is computed by implementing the targeting scheme in Section 3.1.5, where the design variable is initialized to incorporate information from the converged Lyapunov orbit (Lyap0)

$$\boldsymbol{X} = \begin{bmatrix} \dot{y}_{0\text{Lyap1}} = \dot{y}_{0\text{Lyap0}} \\ \varpi_{h\text{Lyap1}} = \varpi_{h\text{Lyap0}} \end{bmatrix}. \tag{3.59}$$

The initial $x_{0\text{Lyap1}}$ position is fixed at the incremented value of $x_{0\text{Lyap1}} = x_{0\text{Lyap0}} \pm \Delta x\ \hat{x}$. In this manner, subsequent members are constructed to develop a subset of the L_1 Lyapunov family as displayed in Fig. 3.8. The size of the step Δx may be varied

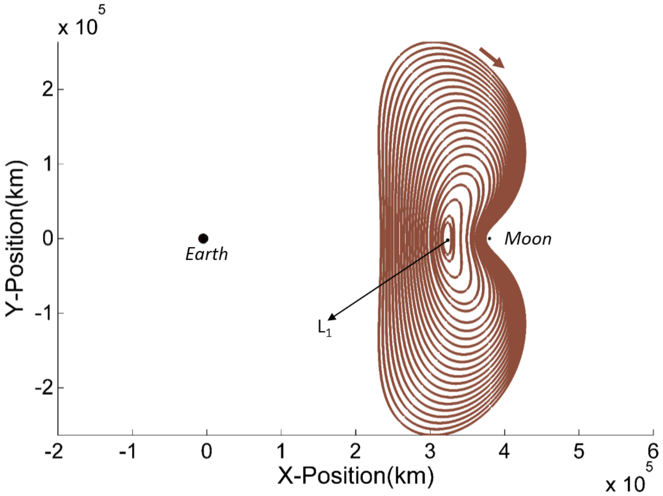


Figure 3.8: Selected trajectories composing the L_1 Lyapunov family

at any time during the continuation process to meet the requirements or address numerical sensitivities. Note that in Fig. 3.8, the trajectories become more deformed with increasing proximity to the Moon. This increased influence from a primary may also increase the sensitivities encountered during the corrections process. So, in addition to the selection of the step-size (Δx), the direction selected for numerical shooting contributes significantly to the success of this continuation strategy as well — e.g., selecting initial conditions on the left-hand x-axis crossing of an orbit to shoot towards the Moon, or selecting those on the right-hand crossing to then target conditions away from the Moon. Usually, the latter option is favored.

Figure 3.9 illustrates the northern and the southern L_1 halo families. The northern family expands in the positive $\hat{z}$ direction and southern family extends in the negative $\hat{z}$ direction — they are mirror images across the $\hat{x} - \hat{y}$ plane. These families are constructed by continuing a particular halo orbit in either the $\pm\hat{x}$ direction (when x_0 is not a design variable) or in the $\pm\hat{z}$ direction (if z_0 is not a design variable). The selection of the natural parameter to enable the continuation can be arbitrary or necessitated by the manner in which a particular family evolves. For example, a visual

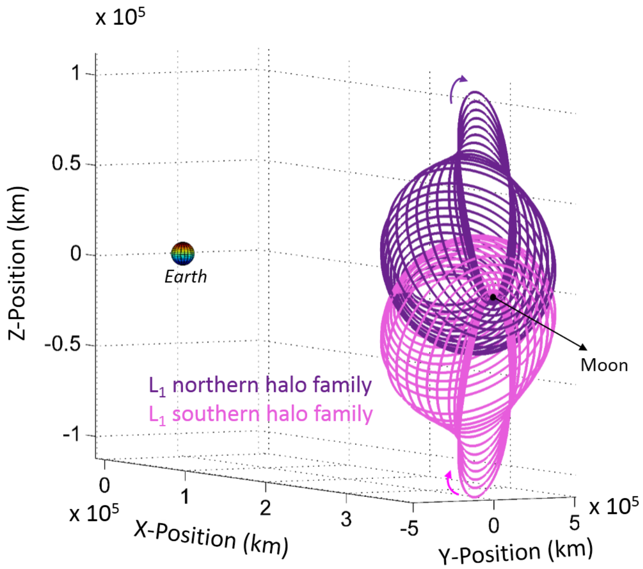


Figure 3.9: Subsets of the L_1 northern and southern halo orbit families

inspection of Fig. 3.9 reveals that the variation in the x_0 component diminishes in the geometrically near-rectilinear subset within each of the northern and southern halo families. So, x_0 may not be a beneficial parameter to conduct numerical continuation in this region. The natural parameter continuation scheme offers the flexibility to alter the parameter selected for continuation as appropriate. So, it may be more beneficial to continue the family in the z_0 parameter in the rectilinear region and continue the other members in x_0 where the variation in this parameter is more apparent from one orbit to another. Numerical instability during the continuation process is an indicator that the continuation parameter may require modification. Again, the appropriate choice of initial states dictate the conditions that contribute to the robustness of the continuation process.

Pseudo-Arclength Continuation

While natural parameter continuation benefits from human intuition and trial-and-error to enable appropriate choices for the physical quantity to continue a family, pseudo-arclength continuation takes a step of size $\mathbb{s}$ in a direction tangent to the family. This latter approach is potentially less susceptible to poor convergence outcomes resulting from varied/unknown rates of change in a particular parameter, but can also present numerical challenges in certain dynamically sensitive regions. So, it is useful to leverage the strengths of both natural parameter and pseudo-arclength continuation schemes as required, for numerical continuation purposes. The directive to step along the tangent direction from a converged orbit (i) is captured via the following update equation to initialize the design vector for computing the subsequent orbit $(i+1)$:

$$\boldsymbol{X}_{i+1} = \boldsymbol{X}_i^* + \mathbb{s}\Diamond\boldsymbol{X}_i^* \tag{3.60}$$

where, $\boldsymbol{X}_i^*$ represents the design conditions for a converged orbit, and $\Diamond\boldsymbol{X}_i^* = \mathfrak{N}(\mathfrak{J}_{\mathrm{i}})$ is the null-vector associated with the Jacobian for this converged orbit. Here, $\mathbb{s}$ scales the size of the step in the direction of $\Diamond\boldsymbol{X}_i$ and is tunable to address numerical sensitivities. A step in the tangent direction by a prescribed size $\mathbb{s}$ is enforced via the following constraint:

$$(\boldsymbol{X}_{i+1} - \boldsymbol{X}_i^*)^T \Diamond \boldsymbol{X}_i^* - \mathbb{s} = 0 \tag{3.61}$$

where $\boldsymbol{X}$ is the design vector, the subscript i represents the values associated with the current orbit in the family, and the superscript $*$ denotes quantities satisfying a converged solution. Equation (3.61) is appended to the original constraint vector to yield

$$\boldsymbol{G}(\boldsymbol{X}_{i+1}) = \begin{bmatrix} \boldsymbol{F}(\boldsymbol{X}_{i+1}) \\ (\boldsymbol{X}_{i+1} - \boldsymbol{X}_i^*)^T \Diamond \boldsymbol{X}_i^* - \mathbb{s} \end{bmatrix}. \tag{3.62}$$

A modified Jacobian matrix $\mathscr{J}$ is computed to accommodate the partials of the constraint vector $\boldsymbol{G}(\boldsymbol{X}_{i+1})$ with respect to the design variables in $\boldsymbol{X}_{i+1}$, i.e.,

$$\mathscr{J} = \frac{\partial \boldsymbol{G}(\boldsymbol{X}_{i+1})}{\partial \boldsymbol{X}_{i+1}} = \begin{bmatrix} \mathfrak{J}_{i+1} \\ \Diamond \boldsymbol{X}_i^{*T} \end{bmatrix}. \tag{3.63}$$

As a consequence of the number of design variables k equal to the number of constraints $\mathbb{m}$ in this formulation, $\mathscr{J}$ possesses full rank and, thus, yields a unique solution for $\boldsymbol{X}_{i+1}^*$ upon convergence.

3.1.7 Motion in the Vicinity of Periodic Orbits

The periodic orbit families in the CR3BP offer a multitude of opportunities for enabling mission applications. In addition to the size, duration, and orientation that factor into the selection of a periodic orbit as the focus of a particular mission, the inherent stability characteristics dictate the ease of capture into these orbits and, also, the efforts associated with maintaining the desired configuration over the mission duration.

Stability — Periodic Orbits

Stability analysis of periodic orbits is initiated by considering variations relative to a state obtained from discretizing an orbit of period $\mathscr{T}$. The periodicity condition ensures that a state $\boldsymbol{\chi}$ returns to itself (in a linear sense) after one period and is, thus, termed a *fixed-state* on a stroboscopic (time-return) map. Its discrete-time linear representation after one map return is formulated as:

$$\delta\boldsymbol{\chi}_{\mathrm{q}=1} = \boldsymbol{\Phi}(\tau + \mathscr{T}, \tau)\delta\boldsymbol{\chi}_{\mathrm{q}=0} \tag{3.64}$$

where, q represents the number of returns to the map; q = 0 is associated with the initial condition. Floquet theory [81] leverages the predictive capability of the STM, along with the relationships defined in Eqn. (3.30) to address an arbitrary number of returns q to the map:

$$\delta\boldsymbol{\chi}(\mathrm{q}\mathscr{T}) = \boldsymbol{\Phi}(\tau + \mathscr{T}, \tau)^{\mathrm{q}}\delta\boldsymbol{\chi}(\tau) \tag{3.65}$$

The general solution representing each map return is then developed as:

$$\delta\boldsymbol{\chi}_{\mathrm{q}} = \sum_{i=1}^{\mathcal{I}} c_i\lambda_i^{\mathrm{q}}\boldsymbol{\rho}_i \tag{3.66}$$

where q represents the integer returns to the stroboscopic map and c_i are the constants of the general solution. The nature of the solution is governed by λ_i, the eigenvalues of the monodromy matrix — the STM matrix after one full period along the periodic solution ($\boldsymbol{\Phi}(\tau + \mathscr{T}, \tau)$). These eigenvalues are also termed the characteristic multipliers of the system. The variable $\boldsymbol{\rho}_i$ is the corresponding eigenvector. Given a solution $\delta\boldsymbol{\chi}_{\mathrm{q}}$ to the equation Eqn. (3.65), the criteria for assessing the stability of such a Floquet system follows as [82]:

- If $|\lambda_i| < 1$ then $\lim_{\mathrm{q}\to\infty}|\delta\boldsymbol{\chi}_{\mathrm{q}}| = 0$ and the solution is asymptotically stable in a linear sense.

- If $|\lambda_i| > 1$ then $\lim_{\mathrm{q}\to\infty}|\delta\boldsymbol{\chi}_{\mathrm{q}}| \to \infty$ and the solution is unstable.

- If $|\lambda_i| = 1$ then there exists a constant such that $\lim_{\mathrm{q}\to\infty}|\delta\boldsymbol{\chi}_{\mathrm{q}}| \leq \epsilon|\delta\boldsymbol{\chi}_0|$, and the solution is oscillatory and linearly marginally stable.

Given Lyapunov's theorem, the eigenvalues of the monodromy matrix occur in reciprocal pairs [83, 84]:

Lyapunov's Theorem

If λ is an eigenvalue of the monodromy matrix $\boldsymbol{\Phi}(\tau + \mathscr{T}, \tau)$ of a time-invariant system, then λ^{-1} is also an eigenvalue. The spectrum of the monodromy matrix of a real time-invariant system is symmetric with respect to both the unit circle and the real axis.

Due to the reciprocal nature of the eigenvalues, if there exists an eigenvalue $|\lambda_i| < 1$, then there must exist an eigenvalue $|\lambda_i| > 1$. So, a solution is determined to be marginally stable *iff* the magnitudes of all the eigenvalues of the monodromy matrix are equal to unity. Since the fixed point returns to itself with each return to the stroboscopic map, at least one eigenvalue equals one, $\lambda_i = 1$. Then, via Lyapunov's theorem, there exists at least one pair of characteristic multipliers equal to unity when considering periodic orbits in the CR3BP. The instabilities associated with the fixed points or periodic orbits enable natural flows towards and away from the solution. These flows are characterized by manifold theory as explored in the following section.

Manifolds — Periodic Orbits

Stable, unstable and center subspaces as well as manifolds associated with periodic orbits aid in determining the motion in the vicinity of these orbits, and also the nature of the flow towards/away from them. These quantities are determined via theorems analogous to those explored for the libration points [5]:

$$\text{for } |\lambda_i| < 1,\ \boldsymbol{\rho}_s \in\ E^s, dim : n_s \tag{3.67}$$

$$\text{for } |\lambda_i| = 1,\ \boldsymbol{\rho}_c \in\ E^c, dim : n_c \tag{3.68}$$

$$\text{for } |\lambda_i| > 1,\ \boldsymbol{\rho}_u \in\ E^u, dim : n_u \tag{3.69}$$

Since the eigenvalues of the monodromy matrix for an orbit initiated at time τ_0, i.e., $\mathbf{\Phi}(\tau_0 + \mathscr{T}, \tau_0)$, yields stability information concerning the periodic orbit, these eigenvalues are the same for any monodromy matrix constructed with respect to time τ along the orbit. However, the eigenvectors vary along the orbit and, thus, influence the direction of flow of the manifolds to/from the orbit. The eigenvector directions at a fixed-point are computed by exploiting knowledge of the directions of the eigenvectors at the initial time $\hat{\rho}_0$, and knowledge of the STM at the desired fixed point along the orbit $\mathbf{\Phi}(\tau_i, \tau_0)$:

$$\hat{\rho}(\tau_i) = \mathbf{\Phi}(\tau_i, \tau_0)\hat{\rho}(\tau_0). \tag{3.70}$$

Here, $\hat{\rho}(\tau_i)$ supply the direction of the local stable and unstable manifolds associated with the orbit.

$$\hat{\rho}_s(\tau_i) = \frac{\boldsymbol{\rho}_s(\tau_i)}{|\boldsymbol{\rho}_s(\tau_i)|} \tag{3.71}$$

$$\hat{\rho}_u(\tau_i) = \frac{\boldsymbol{\rho}_u(\tau_i)}{|\boldsymbol{\rho}_u(\tau_i)|}. \tag{3.72}$$

A small nondimensional step ϵ in the direction of these eigenvectors delivers the initial conditions $\boldsymbol{\chi}_s(\tau_i)$ and $\boldsymbol{\chi}_u(\tau_i)$to generate the global manifolds $\mathcal{W}_i^s$ and $\mathcal{W}_i^u$ respectively.

$$\boldsymbol{\chi}_s(\tau_i) = \boldsymbol{\chi}(\tau_i) \pm \epsilon\hat{\rho}_s(\tau_i) \tag{3.73a}$$

$$\boldsymbol{\chi}_u(\tau_i) = \boldsymbol{\chi}(\tau_i) \pm \epsilon\hat{\rho}_u(\tau_i) \tag{3.73b}$$

Again, the local stable manifold is propagated in reverse time and the unstable manifold in forward time to generate the global manifolds associated with the periodic orbit. Figure. 3.10 illustrates the $\mathcal{W}_s^-$ and $\mathcal{W}_u^+$ global manfiolds for a sample L_1 Lyapunov orbit. Periodic orbit manifolds are advantageous to facilitate transport between regions near the primaries and between other locations in a complex dynamical regime. Flow within the center-manifold subspace facilitates quasi-periodic motion [85] and, thus, the ability to remain in the vicinity of the periodic orbit for some finite duration.

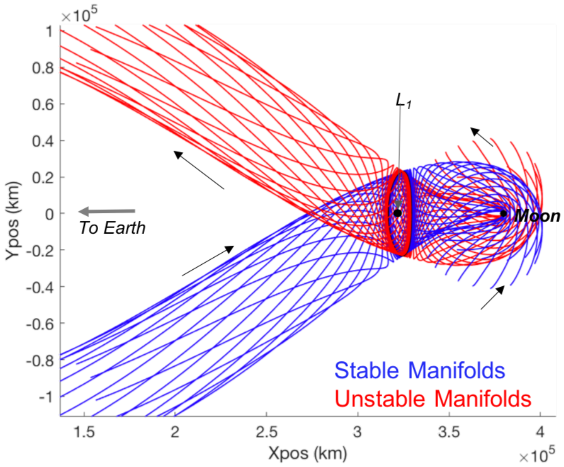


Figure 3.10: Stable and unstable manifold arcs associated with a sample L_1 Lyapunov orbit. The arrows display the direction of flow in forward time.

3.2 Considerations in the Spacecraft-Augmented CR3BP

The construction of optimal spacecraft trajectories typically involves formulating an optimal control problem and solving for the time histories of the spacecraft maneuvers, in magnitude and direction, to meet the desired boundary conditions. Nonlinear Programming (NLP) techniques are exploited to compute these control variable histories and solve the associated Two Point Boundary Value Problem (2PBVP). In general, direct methods are more robust, but can induce a large dimensionality to the problem. Indirect optimization methods mitigate such limitations, but often introduce numerical challenges [86] due to a small radius of convergence; good initial guesses are essential in such cases. Both approaches are employed in the current investigation.

For power-limited VSI engines, a constant power level results in variable thrust magnitudes and *Isp* values (Eqn. (2.50)). These additional variables further expand the dimensionality of the problem but the *Isp* modulation freedom also aids in the VSI convergence process given poor initial guesses. Thus, indirect methods are well suited for VSI-related numerical computations; in fact, a hybrid optimization scheme is implemented in Stuart et al. [46] that combines the benefits of both the direct and indirect techniques. In contrast, direct methods are employed in the CSI regime where constrained thrust control parameters cause numerical challenges. The discrete nature of the maneuvers in chemical engine-enabled trajectories renders the associated computations suitable for direct optimization schemes as well.

3.2.1 Hybrid Optimization Scheme for the VSI Regime

Trajectory optimization involves computing small variations in a functional to determine the associated extrema [87]. This approach is termed *Calculus of Variations, CoV*, the name coined by Leonhard Euler in 1756 based on the compilation of the work undertaken by himself and Joseph-Luis Lagrange [88]. The functional optimization problem in the VSI regime aims to compute the controls ($\mathbb{u} \rightarrow$ power usage as well as thrust magnitude and direction) to maximize the end-cost J_e, the spacecraft arrival mass m_f, at the destination:

$$J_e = J_e(\boldsymbol{\chi}_{LT,f}, \tau_f) = m_f \tag{3.74}$$

subject to the state equations,

$$\dot{\boldsymbol{\chi}}_{LT} = \boldsymbol{f}(\boldsymbol{\chi}_{LT}(\tau), \mathbb{u}(\tau)) \tag{3.75}$$

constraints on the thrust pointing direction,

$$\boldsymbol{u}_i{}^T \boldsymbol{u}_i - 1 \tag{3.76}$$

and boundary conditions

$$\boldsymbol{\psi}_0 = \begin{pmatrix} \boldsymbol{r}_0 - \boldsymbol{r}_{\text{dep}} \\ \boldsymbol{v}_0 - \boldsymbol{v}_{\text{dep}} \end{pmatrix} = \boldsymbol{0} \tag{3.77}$$

$$\boldsymbol{\psi}_f = \begin{pmatrix} \boldsymbol{r}_f - \boldsymbol{r}_{\text{arr}} \\ \boldsymbol{v}_f - \boldsymbol{v}_{\text{arr}} \end{pmatrix} = \boldsymbol{0} \tag{3.78}$$

where the subscripts ‘dep’ and ‘arr’ denote the departure and arrival conditions, respectively. To incorporate all the above considerations into the NLP problem, the following augmented performance index is delivered:

$$max\ J = \underbrace{m_f + \boldsymbol{\nu}_0^T \boldsymbol{\psi}_0 + \boldsymbol{\nu}_f^T \boldsymbol{\psi}_f}_{\text{end costs}} + \underbrace{\int_{\tau_0}^{\tau_f} (H - \boldsymbol{\lambda}^T \dot{\boldsymbol{\chi}}_{LT}) d\tau}_{\text{path costs}} \tag{3.79}$$

The *discrete* equality constraints at the boundaries in Eqns. (3.77) and (3.78) are adjoined to the terminal cost (Eqn. (3.74)) via Lagrange multipliers $\boldsymbol{\nu}$, and the state equations in Eqn. (3.75) are adjoined via costates $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_{\boldsymbol{r}}, \boldsymbol{\lambda}_{\boldsymbol{v}}, \mathbb{\lambda}_m]^T$, contributing to the integrand and influencing the *continuous* path costs. Here, $H = \mathcal{H} + \eta(\boldsymbol{u}^T \boldsymbol{u} - 1)$, is a function of the Hamiltonian, $\mathcal{H}$ (constant), Lagrange multipliers $\mathbb{\eta}$, and the thrust direction unit magnitude constraint. The Hamiltonian in this problem is defined as:

$$\mathcal{H} = \boldsymbol{\lambda}_{\boldsymbol{r}}^T \underbrace{\boldsymbol{v}}_{\dot{\boldsymbol{r}}} + \boldsymbol{\lambda}_{\boldsymbol{v}}^T \underbrace{(\boldsymbol{f}(\boldsymbol{r}) + \boldsymbol{g}(\boldsymbol{v}) + \frac{\mathbb{T}}{m}\hat{u})}_{\dot{\boldsymbol{v}}} - \mathbb{\lambda}_m \underbrace{\frac{\mathbb{T}^2}{2\mathbb{P}}}_{\dot{m}} . \tag{3.80}$$

This quantity proves essential in solving for the extrema of the performance index in Eqn. (3.79) as the following condition is satisfied for an optimal path via Pontryagin's maximum principle [89]:

$$H[\boldsymbol{\chi}_{LT}^*(\tau), \tau, \mathbb{u}^*(\tau), \boldsymbol{\lambda}(\tau)] \geq H[\boldsymbol{\chi}_{LT}^*(\tau), t, \mathbb{u}(\tau), \boldsymbol{\lambda}(\tau)] \tag{3.81}$$

The Euler-Lagrange equations provide the necessary conditions for computing the extrema $dJ = 0$:

$$\dot{\boldsymbol{\lambda}} = \left(-\frac{\partial H}{\partial \boldsymbol{\chi}_{LT}} \right) \tag{3.82}$$

$$\left(\frac{\partial H}{\partial \mathbb{T}} \right) = \boldsymbol{\lambda}_{\boldsymbol{v}} \frac{\hat{u}}{m} - \mathbb{\lambda}_m \frac{\mathbb{T}}{\mathbb{P}} = 0 \tag{3.83}$$

$$\left(\frac{\partial H}{\partial \hat{u}} \right) = \boldsymbol{\lambda}_{\boldsymbol{v}}^T \frac{\mathbb{T}}{m} + 2\mathbb{\eta}\hat{u} = 0 \tag{3.84}$$

Since the maximum power is assumed to be constant in the VSI regime in this investigation,

$$\mathbb{P} = \mathbb{P}_{max} \tag{3.85}$$

to maximize the Hamiltonian (Eqn. (3.80)) and, thus, the performance index J. Also, in general, the following is true when considering Eqn. (3.84) [45]: ($\boldsymbol{\lambda}_{\boldsymbol{v}}$ and $\mathbb{\eta} \neq 0$) and ($\mathbb{T}$ and $\mathbb{\eta} \neq 0$). So, $\hat{u}$ must be directed parallel to the velocity costate quantity ($\hat{u} = \frac{\boldsymbol{\lambda}_{\boldsymbol{v}}}{\mathbb{\lambda}_v}$) or anti-parallel ($\hat{u} = -\frac{\boldsymbol{\lambda}_{\boldsymbol{v}}}{\mathbb{\lambda}_v}$). The Weierstrass condition and Pontryagin's maximum principle are exploited to select the former quantity that maximizes the Hamiltonian at every instant along the path:

$$\hat{u} = \frac{\boldsymbol{\lambda}_{\boldsymbol{v}}}{\lambda_v} \tag{3.86}$$

This equation also represents Lawden's primer-vector control law [90, 91]. Equation (3.83) is re-arranged as

$$\mathbb{T} = \frac{\lambda_v \mathbb{P}_{max}}{\lambda_m m}, \tag{3.87}$$

and the thrust magnitude history is extractable in a post-processing step upon computation of the associated costates. Equation. (3.82) is expanded as follows to deliver the EOMs for the costate quantities:

$$\dot{\boldsymbol{\lambda}} = \left(- \frac{\partial H}{\partial \boldsymbol{\chi}_{LT}} \right) = \begin{cases} \dot{\boldsymbol{\lambda}}_{\boldsymbol{r}} = -\boldsymbol{\lambda}_{\boldsymbol{v}}^T \left(\frac{\partial \boldsymbol{f}(\boldsymbol{r})}{\partial \boldsymbol{r}} \right) \\ \dot{\boldsymbol{\lambda}}_{\boldsymbol{v}} = -\boldsymbol{\lambda}_{\boldsymbol{r}}^T - \boldsymbol{\lambda}_{\boldsymbol{v}}^T \left(\frac{\partial \boldsymbol{g}(\boldsymbol{v})}{\partial \boldsymbol{v}} \right) \\ \dot{\lambda}_m = \lambda_v \frac{\mathbb{T}}{m^2} \end{cases} \tag{3.88}$$

The initial value of the mass costate λ_{m_0} is set equal to unity to reduce the dimensionality of the problem. Additionally, the boundary conditions in Eqn. (3.79) are addressed via the transversality conditions computed as:

$$H d\tau - \boldsymbol{\lambda}^T d\boldsymbol{\chi}_{LT} + dJ_e = 0 \tag{3.89}$$

$$\text{subject to, } d\boldsymbol{\psi} = \mathbf{0}$$

In a pure indirect optimization scheme, constraint equations (transversality conditions) are derived from Eqn. (3.89) [47] and incorporated into a numerical corrections scheme as discussed in Section 3.1.4 to solve for the optimal control history and solution. However, these solutions are very sensitive to small variations in boundary conditions [92], leading to numerical challenges. So, a hybrid optimization option is sought that operates independently of the sensitive transversality conditions by: (i) ‘directly’ optimizing the terminal objective (maximize m_f) via NLP solvers, and (ii) incorporating the costates from the Euler-Lagrange theory to ‘indirectly’ solve for the optimal control history [46]. The problem formulation is described as follows.

Numerical Corrections Considerations for the VSI Regime

The numerical corrections process for the VSI regime is enabled by augmenting the s/c EOMs to include the costate differential equations as well. These EOMs are constructed as:

$$\dot{\boldsymbol{\chi}}_{VSI} = \begin{Bmatrix} \dot{\boldsymbol{\chi}}_{LT} \\ \dot{\boldsymbol{\lambda}} \end{Bmatrix} = \begin{Bmatrix} \dot{\boldsymbol{\chi}} \\ \dot{m} \\ \dot{\boldsymbol{\lambda}} \end{Bmatrix} = \begin{Bmatrix} \dot{\boldsymbol{r}} \\ \dot{\boldsymbol{v}} \\ \dot{m} \\ \dot{\boldsymbol{\lambda}}_{\boldsymbol{r}} \\ \dot{\boldsymbol{\lambda}}_{\boldsymbol{v}} \\ \dot{\lambda}_m \end{Bmatrix}. \tag{3.90}$$

The resulting Jacobian matrix ($\mathbf{A}_{14\times14}$) consisting of the partial derivatives of these equations with respect to the state variables is derived as:

$$\mathbf{A}_{\text{VSI},14\times14} = \begin{bmatrix} 0_{3\times3} & I_{3\times3} & 0_{3\times1} & 0_{3\times3} & 0_{3\times3} & 0_{3\times1} \\ \frac{\partial \boldsymbol{f}(\boldsymbol{r})}{\partial \boldsymbol{r}} & \frac{\partial \boldsymbol{g}(\boldsymbol{v})}{\partial \boldsymbol{v}} & \frac{-2\mathbb{P}}{m^3\lambda_m}\boldsymbol{\lambda}_{\boldsymbol{v}} & 0_{3\times3} & \frac{\mathbb{P}}{m^2\lambda_m}I_{3\times3} & -\frac{\mathbb{P}}{m^2\lambda_m^2}\boldsymbol{\lambda}_{\boldsymbol{v}} \\ 0_{1\times3} & 0_{1\times3} & \frac{\mathbb{P}\lambda_v^2}{m^3\lambda_m^2} & 0_{1\times3} & -\frac{\mathbb{P}}{m^2\lambda_m^2}\boldsymbol{\lambda}_{\boldsymbol{v}}^T & \frac{\mathbb{P}\lambda_v^2}{m^2\lambda_m^3} \\ (\frac{\partial \dot{\boldsymbol{\lambda}}_{\boldsymbol{r}}}{\partial \boldsymbol{r}})_{3\times3} & 0_{3\times3} & 0_{3\times1} & 0_{3\times3} & -\frac{\partial \boldsymbol{f}(\boldsymbol{r})}{\partial \boldsymbol{r}}^T & 0_{3\times1} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times1} & -I_{3\times3} & -\frac{\partial \boldsymbol{g}(\boldsymbol{v})}{\partial \boldsymbol{v}}^T & 0_{3\times1} \\ 0_{1\times3} & 0_{1\times3} & -\frac{3\mathbb{P}\lambda_v^2}{m^4\lambda_m} & 0_{1\times3} & \frac{2\mathbb{P}}{m^3\lambda_m}\boldsymbol{\lambda}_{\boldsymbol{v}}^T & -\frac{\mathbb{P}\lambda_v^2}{m^3\lambda_m^2} \end{bmatrix} \tag{3.91}$$

where $(\frac{\partial \dot{\boldsymbol{\lambda}}_r}{\partial \boldsymbol{r}})_{3\times3}$ is further expanded upon as:

$$\begin{aligned}
(\tfrac{\partial \dot{\boldsymbol{\lambda}}_r}{\partial \boldsymbol{r}})_{11} &= -\lambda_{v_x} U^*_{x^3} - \lambda_{v_y} U^*_{x^2y} - \lambda_{v_z} U^*_{x^2z} \\
(\tfrac{\partial \dot{\boldsymbol{\lambda}}_r}{\partial \boldsymbol{r}})_{12} &= -\lambda_{v_x} U^*_{x^2y} - \lambda_{v_y} U^*_{xy^2} - \lambda_{v_z} U^*_{xyz} \\
(\tfrac{\partial \dot{\boldsymbol{\lambda}}_r}{\partial \boldsymbol{r}})_{13} &= -\lambda_{v_x} U^*_{x^2z} - \lambda_{v_y} U^*_{xyz} - \lambda_{v_z} U^*_{xz^2} \\
(\tfrac{\partial \dot{\boldsymbol{\lambda}}_r}{\partial \boldsymbol{r}})_{21} &= -\lambda_{v_x} U^*_{x^2y} - \lambda_{v_y} U^*_{xy^2} - \lambda_{v_z} U^*_{xyz} \\
(\tfrac{\partial \dot{\boldsymbol{\lambda}}_r}{\partial \boldsymbol{r}})_{22} &= -\lambda_{v_x} U^*_{xy^2} - \lambda_{v_y} U^*_{y^3} - \lambda_{v_z} U^*_{y^2z} \\
(\tfrac{\partial \dot{\boldsymbol{\lambda}}_r}{\partial \boldsymbol{r}})_{23} &= -\lambda_{v_x} U^*_{xyz} - \lambda_{v_y} U^*_{y^2z} - \lambda_{v_z} U^*_{yz^2} \\
(\tfrac{\partial \dot{\boldsymbol{\lambda}}_r}{\partial \boldsymbol{r}})_{31} &= -\lambda_{v_x} U^*_{x^2z} - \lambda_{v_y} U^*_{xyz} - \lambda_{v_z} U^*_{xz^2} \\
(\tfrac{\partial \dot{\boldsymbol{\lambda}}_r}{\partial \boldsymbol{r}})_{32} &= -\lambda_{v_x} U^*_{xyz} - \lambda_{v_y} U^*_{y^2z} - \lambda_{v_z} U^*_{yz^2} \\
(\tfrac{\partial \dot{\boldsymbol{\lambda}}_r}{\partial \boldsymbol{r}})_{33} &= -\lambda_{v_x} U^*_{xz^2} - \lambda_{v_y} U^*_{yz^2} - \lambda_{v_z} U^*_{z^3}
\end{aligned} \tag{3.92}$$

Following the analysis in Section 3.1.3, the A_{VSI} matrix facilitates the construction of the STM for the VSI regime via:

$$\dot{\boldsymbol{\Phi}}_{VSI}(\tau, \tau_0) = \mathbf{A}_{VSI}(\tau)\boldsymbol{\Phi}_{VSI}(\tau, \tau_0) \tag{3.93}$$

The VSI trajectory corrections process employs a free-variable and constraint formulation as described in Section 3.1.4. An iterative Newton-Raphson scheme is employed to compute the set of design vector variables ($\boldsymbol{X}^*_{VSI}$) that satisfy the specified constraints, $\boldsymbol{F}(\boldsymbol{X}^*_{VSI}) = \mathbf{0}$, i.e.,

$$\boldsymbol{X}_{VSI} = \begin{bmatrix} \boldsymbol{\chi}_{VSI_i} \\ \varpi_i \end{bmatrix} \quad \boldsymbol{F}_{VSI} = \begin{bmatrix} \boldsymbol{\chi}_{i,f} - \boldsymbol{\chi}_{i+1,0} \\ \boldsymbol{\psi}_{desired} - \boldsymbol{\psi}_{actual} \\ TOF_{total} - \sum_{i=1}^{\mathfrak{n}-1} \varpi_i \end{bmatrix} = \mathbf{0} \tag{3.94}$$

The design vector $\boldsymbol{X}_{VSI}$, is comprised of the states ($\boldsymbol{\chi}_{VSI_i}$) and propagation durations (ϖ_i) at select nodes (i). The constraint vector $\boldsymbol{F}_{VSI}$ incorporates state continuity constraints, boundary conditions ($\boldsymbol{\psi}$), and maintains a specified total flight duration. If the departure and arrival conditions are constructed as a function of the propagation durations ϖ_{dep} and ϖ_{arr} relative to a reference location along their respective orbits, then the design and constraint vectors are re-formulated as:

$$\boldsymbol{X}_{VSI} = \begin{bmatrix} \boldsymbol{\lambda}_{1(\boldsymbol{r},\boldsymbol{v})} \\ \boldsymbol{\chi}_{VSI_{2}} \\ \vdots \\ \boldsymbol{\chi}_{VSI_{\mathbb{n}-1}} \\ \boldsymbol{\chi}_{LT,_{\mathbb{n}}} \\ \varpi_{dep} \\ \varpi_{arr} \\ \varpi_{1} \\ \vdots \\ \varpi_{\mathbb{n}-1} \end{bmatrix} \qquad \boldsymbol{F}_{VSI} = \begin{bmatrix} \boldsymbol{\chi}_{VSI,1_{f}} - \boldsymbol{\chi}_{VSI,2_{0}} \\ \vdots \\ \boldsymbol{\chi}_{LT,\mathbb{n}-1_{f}} - \boldsymbol{\chi}_{LT,\mathbb{n}} \\ \boldsymbol{\chi}_{\mathbb{n}} - \boldsymbol{\chi}_{arr} \\ TOF_{total} - \sum_{i=1}^{\mathbb{n}-1} \varpi_{i} \end{bmatrix} = \boldsymbol{0} \tag{3.95}$$

where $\mathbb{n}$ represents the total number of nodes along the transfer trajectory, and the subscripts 0 and f denote the beginning and end of an arc, respectively. Recall from Section 2.3 that $\boldsymbol{\chi}_{LT}$ represents the spacecraft position, velocity and mass states. Also, note that the position and velocity at the first node (departure conditions) are not incorporated into the design vector as the variable ϖ_{dep} determines these conditions on the departure orbit and, thus, at the first node as well. The mass is initiated at a nondimensional value of unity and decreases with propellant consumption during the transfer. The costates determine the direction and magnitude of thrust employed at the departure location. Also, the boundary constraint is defined such that the position and velocity of the free-floating (non-propagated) final node $\boldsymbol{\chi}_{\mathbb{n}}$ equals those on the arrival orbit $\boldsymbol{\chi}_{arr}$, as determined by the free variable ϖ_{arr}. Furthermore, a low-thrust engine is typically more efficient over longer flight durations. So, to prohibit the *Isp* values from modulating to infinity ($\dot{m} \rightarrow 0$), the VSI corrections process is a time-fixed formulation; the value TOF_{total} is the sum of all the segment durations between each node prior to the initiation of the convergence process or, alternatively, one that is defined by the user.

Derivation of the Jacobian Matrix

The complete Jacobian is constructed by deriving the partial derivatives of the constraints in $\boldsymbol{F}_{VSI}$ with respect to the design variables in $\boldsymbol{X}_{VSI}$. The same process outlined in Section 3.1.4 is followed to compute the sensitivity in state continuity constraints to variations in the states $\mathfrak{J}_{S,\boldsymbol{\chi}}$, and propagation durations $\mathfrak{J}_{S,\varpi}$. Additionally, state continuity at the end of the first transfer arc ($\boldsymbol{\chi}_{VSI,1_f}$) varies based on the value of its originating states ($\boldsymbol{\chi}_{VSI,1_0}$) which is determined as a function of ϖ_{dep}. This sensitivity is captured via the chain rule:

$$\begin{aligned}\mathfrak{J}_{S,\varpi_{dep}} &= \frac{\partial\left(\boldsymbol{\chi}_{VSI,1_f} - \boldsymbol{\chi}_{VSI,2_0}\right)}{\partial\varpi_{dep}} = \frac{\boldsymbol{\chi}_{VSI,1_f}}{\partial\varpi_{dep}} - \cancelto{0}{\frac{\boldsymbol{\chi}_{VSI,2_0}}{\partial\varpi_{dep}}} \\ &= \frac{\partial\boldsymbol{\chi}_{VSI,1_f}}{\partial\boldsymbol{\chi}_{1,0}(\varpi_{dep})}\frac{d\boldsymbol{\chi}_{1,0}(\varpi_{dep})}{d\varpi_{dep}}\end{aligned} \tag{3.96}$$

Thus, Eqn. (3.96) reduces to:

$$\mathfrak{J}_{S,\varpi_{dep}} = \boldsymbol{\Phi}_{\mathbb{1}}(\mathbb{T}_1 + \varpi_1, \mathbb{T}_1)\dot{\boldsymbol{\chi}}_{1,0} = \boldsymbol{\Phi}_{\mathbb{1}}(\mathbb{T}_1 + \varpi_1, \mathbb{T}_1)\dot{\boldsymbol{\chi}}_{dep} \tag{3.97}$$

The time at the node 'i' is represented by $\mathbb{T}_i$ and $\dot{\boldsymbol{\chi}}_{1,0}$ represents the rates of change of the position and velocity states at the end of the departure orbit propagation $\dot{\boldsymbol{\chi}}_{dep}$ as determined by the free variable ϖ_{dep}. In Eqn. (3.97), $\boldsymbol{\Phi}_{\mathbb{1}}(\mathbb{T}_1 + \varpi_1, \mathbb{T}_1)$ is defined as:

$$\boldsymbol{\Phi}_{\mathbb{1}}(\mathbb{T}_1 + \varpi_1, \mathbb{T}_1) = \boldsymbol{\Phi}_{\mathbb{1}} = \begin{bmatrix} \boldsymbol{\Phi}_{1,\boldsymbol{rr}} & \boldsymbol{\Phi}_{1,\boldsymbol{rv}} \\ \boldsymbol{\Phi}_{1,\boldsymbol{vr}} & \boldsymbol{\Phi}_{1,\boldsymbol{vv}} \\ \boldsymbol{\Phi}_{1,m\boldsymbol{r}} & \boldsymbol{\Phi}_{1,m\boldsymbol{v}} \\ \boldsymbol{\Phi}_{1,\boldsymbol{\lambda r}} & \boldsymbol{\Phi}_{1,\boldsymbol{\lambda v}} \end{bmatrix} \tag{3.98}$$

Similarly, the sensitivity associated with the state continuity constraint between the free-floating (non-propagated) node $\boldsymbol{\chi}_{\mathbbm{n}}$, and the terminal states on the arrival orbit $\boldsymbol{\chi}_{arr}$ as determined by ϖ_{arr} is computed as:

$$\begin{aligned}\mathfrak{J}_{B,\varpi_{arr}} &= \frac{\partial\Big(\boldsymbol{\chi}_{\mathbbm{n}} - \boldsymbol{\chi}_{arr}(\varpi_{arr})\Big)}{\partial\varpi_{arr}} = \cancelto{0}{\frac{\partial\boldsymbol{\chi}_{\mathbbm{n}}}{\partial\varpi_{arr}}} - \frac{\partial\boldsymbol{\chi}_{arr}(\varpi_{arr})}{\partial\varpi_{arr}} \\ &= -\frac{d\boldsymbol{\chi}_{arr}}{d\varpi_{arr}} = -\dot{\boldsymbol{\chi}}_{arr} \end{aligned} \tag{3.99}$$

The sensitivity of the total TOF constraint to the thrust durations is computed as:

$$\mathfrak{J}_{TOF,\varpi_i} = \cancelto{0}{\frac{\partial TOF_{total}}{\partial\varpi_i}} - \frac{\partial\varpi_i}{\partial\varpi_i} = -1. \tag{3.100}$$

So,

$$\mathfrak{J}_{TOF,\varpi} = -\mathbf{1}_{(1\times\mathbbm{n}-1)} \tag{3.101}$$

The complete Jacobian matrix for the VSI regime computing the derivatives of the constraints $\boldsymbol{F}_{VSI}$ with respect to the design variables $\boldsymbol{X}_{VSI}$ in Eqn. (3.95) is compiled as:

$$\mathfrak{I}_{VSI} = \begin{array}{cc} \begin{array}{ccccccccccc} \mathbb{\lambda}_{1(\boldsymbol{r},\boldsymbol{v})} & \boldsymbol{\chi}_{VSI_2} & \boldsymbol{\chi}_{VSI_3} & \dots & \boldsymbol{\chi}_{LT,\mathbb{n}} & \varpi_{dep} & \varpi_{arr} & \varpi_1 & \varpi_2 & \dots & \varpi_{\mathbb{n}-1} \end{array} & \\ \left[\begin{array}{ccccccccccc} \boldsymbol{\Phi}_1 & -\boldsymbol{I} & \mathbf{0} & \dots & \mathbf{0} & \boldsymbol{\Phi}_{\mathbb{1}}\dot{\boldsymbol{\chi}}_{dep} & \mathbf{0} & \dot{\boldsymbol{\chi}}_{1,f} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Phi}_2 & -\boldsymbol{I} & \dots & \vdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dot{\boldsymbol{\chi}}_{2,f} & \dots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{0} & \boldsymbol{\Phi}_{\mathbb{n}-1} & -\boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \dots & \dot{\boldsymbol{\chi}}_{\mathbb{n}-1,f} \\ \mathbf{0} & \dots & \dots & \dots & \boldsymbol{I} & \mathbf{0} & -\dot{\boldsymbol{\chi}}_{arr} & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \mathbf{0} & \dots & \dots & \dots & \dots & \dots & \dots & -1 & -1 & \dots & -1 \end{array}\right] & \begin{array}{c} \boldsymbol{\chi}_{VSI,1_f} - \boldsymbol{\chi}_{VSI,2_0} \\ \boldsymbol{\chi}_{VSI,2_f} - \boldsymbol{\chi}_{VSI,3_0} \\ \vdots \\ \boldsymbol{\chi}_{LT,\mathbb{n}-1_f} - \boldsymbol{\chi}_{LT,\mathbb{n}} \\ \boldsymbol{\chi}_{\mathbb{n}} - \boldsymbol{\chi}_{arr} \\ TOF_{total} - \sum_{i=1}^{\mathbb{n}-1} \varpi_i \end{array} \end{array} \tag{3.102}$$

The rows of the matrix correspond to the constituents of the constraint vector $\boldsymbol{F}_{VSI}$ and the columns correspond to the components of the design vector $\boldsymbol{X}_{VSI}$; these values are highlighted in magenta in Eqn. (3.102). Note that $\boldsymbol{I}$ in blue in Eqn. (3.102) aligns with the position and velocity columns alone for the last node. The converged solution is passed through a Nonlinear Programming (NLP) software such as MATLAB©'s `fmincon` or SNOPT [93] to optimize the end-cost function that is formulated to offer a favorable solution to the mission objectives and constraints.

3.2.2 Direct Optimization Scheme for the CSI Regime

Direct optimization methods are an alternative approach for computing locally optimal trajectories compared to the hybrid option pursued for the VSI regime. In this approach, instead of transitioning the cost function to a higher dimensional space via costates to obtain a continuous control history, the controls are discretized along the trajectory [94]. NLP techniques are then incorporated to construct the final optimal solution subject to the Karush-Kuhn-Tucker (KKT) checks to ensure that first-order optimality conditions are met. Stryk et al. [95] report that some of these choices lead to relatively less accurate outcomes for the cost function, and that the emergence of multiple 'pseudominima' can result in the final presented solution not truly satisyfing all the previously examined CoV criteria. However, the direct optimization strategy is effective in numerically challenging scenarios because of the broad radius of convergence offered [92]. Such an advantage is especially useful in mitigating the numerical sensitivities introduced by the constant and bounded thrust magnitude constraints associated with CSI-engine enabled trajectories. In the CSI regime, formulations for two design scenarios are explored: (i) transfers and, (ii) rendezvous accounting for pre-positioned space assets.

Numerical Corrections Considerations for the CSI Regime — Transfers

The transfer scenarios explored in this section aim to deliver a s/c from a departure location to its destination orbit without the requirement to rendezvous with a prepositioned object at the destination. The general numerical corrections formulation for the CSI regime is enabled by augmenting the s/c EOMs to include the thrust magnitude and direction differential equations as well. These EOMs are formulated as:

$$\dot{\boldsymbol{\chi}}_{CSI} = \begin{Bmatrix} \dot{\boldsymbol{\chi}} \\ \dot{m} \\ \dot{\mathbb{T}} \\ \dot{\boldsymbol{u}} \end{Bmatrix} = \begin{Bmatrix} \dot{\boldsymbol{r}} \\ \dot{\boldsymbol{v}} \\ \dot{m} \\ \dot{\mathbb{T}} \\ \dot{u}_x \\ \dot{u}_y \\ \dot{u}_z \end{Bmatrix} \tag{3.103}$$

The resulting Jacobian matrix ($\mathbf{A}_{11\times11}$) consisting of the partial derivatives of these equations with respect to the state variables is derived as:

$$\mathbf{A}_{\text{CSI},11\times11} = \begin{bmatrix} 0_{3\times3} & I_{3\times3} & 0_{3\times1} & 0_{3\times1} & 0_{3\times3} \\ \frac{\partial \boldsymbol{f(r)}}{\partial \boldsymbol{r}} & \frac{\partial \boldsymbol{g(v)}}{\partial \boldsymbol{v}} & -\frac{\mathbb{T}}{m^2}\hat{u} & \frac{\hat{u}}{m} & \frac{\mathbb{T}}{m}I_{3\times3} \\ 0_{1\times3} & 0_{1\times3} & 0_{1\times1} & -\frac{1}{Isp\cdot g_0} & 0_{1\times3} \\ 0_{1\times3} & 0_{1\times3} & 0_{1\times1} & 0_{1\times1} & 0_{1\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times1} & 0_{3\times1} & 0_{3\times3} \end{bmatrix} \tag{3.104}$$

The rows pertaining to the thrust magnitude and directions are zero in the $\mathbf{A}_{\text{CSI}}$ matrix because these values are held constant over a propagated arc. The quantities $\mathbb{T}$ and $\hat{u}$ are however maintained in the EOMs for generality of derivation. Later steps exploit the fact that they are constant over a thrust arc to simplify computations. The Jacobian matrix facilitates the construction of the STM via:

$$\dot{\mathbf{\Phi}}_{CSI}(\tau,\tau_0) = \mathbf{A}_{CSI}(\tau)\mathbf{\Phi}_{CSI}(\tau,\tau_0) \tag{3.105}$$

The Newton-Raphson scheme adopting the free-variable and constraint formulation described in Sections 3.1.4 and 3.2.1 is used to compute the design vector variables ($\boldsymbol{X}^*_{CSI}$) that satisfy the specified constraints $\boldsymbol{F}(\boldsymbol{X}^*_{CSI}) = \mathbf{0}$, as follows:

$$\boldsymbol{X}_{CSI} = \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{\chi}_{CSI_2} \\ \vdots \\ \boldsymbol{\chi}_{CSI_{\mathbb{n}-1}} \\ \boldsymbol{\chi}_{LT_{\mathbb{n}}} \\ \varpi_{dep} \\ \varpi_{arr} \\ \varpi_1 \\ \vdots \\ \varpi_{\mathbb{n}-1} \end{bmatrix} \quad \boldsymbol{F}_{CSI} = \begin{bmatrix} \boldsymbol{\chi}_{CSI,1_f} - \boldsymbol{\chi}_{CSI,2_0} \\ \vdots \\ \boldsymbol{\chi}_{LT,\mathbb{n}-1_f} - \boldsymbol{\chi}_{LT,\mathbb{n}} \\ \boldsymbol{\chi}_{\mathbb{n}} - \boldsymbol{\chi}_{arr} \\ \boldsymbol{u}_1{}^T\boldsymbol{u}_1 - 1 \\ \vdots \\ \boldsymbol{u}^T_{\mathbb{n}-1}\boldsymbol{u}_{\mathbb{n}-1} - 1 \end{bmatrix} = \mathbf{0} \tag{3.106}$$

The design vector $\boldsymbol{X}_{CSI}$, is comprised of the states ($\boldsymbol{\chi}_{CSI_i}$) and propagation durations (ϖ_i) at select nodes (i). In this investigation, a node is pre-assigned a binary (on/off) thrust magnitude, so this quantity is not included within the state variables $\boldsymbol{\chi}_{CSI_i}$ in the design vector $\boldsymbol{X}_{CSI}$. The variable inter-node propagation durations ϖ_i determine the duration of the thrust and coast arcs. In an alternative formulation, the thrust magnitude may be included as a design variable. The constraint vector $\boldsymbol{F}_{CSI}$ is sought to meet state continuity, boundary condition, and thrust direction unit vector constraints at the specified nodes. Since the Isp and thrust magnitudes are fixed in the adopted CSI formulation, the total flight duration is not constrained, as is the case in the VSI regime.

Derivation of the Jacobian Matrix

Many of the Jacobian sub-matrices derived for the VSI regime are applicable during the CSI corrections process as well. The sensitivity of the state continuity constraint at the end of the propagation from the first node to variations in the departure location is derived in Section 3.2.1 as:

$$\mathfrak{J}_{S,\varpi_{dep}} = \boldsymbol{\Phi}_{\mathbb{1}}(\mathbb{\tau}_1 + \varpi_1, \mathbb{\tau}_1)\dot{\boldsymbol{\chi}}_{dep} \tag{3.107}$$

The time at the node 'i' is represented by $\mathbb{\tau}_i$ and $\dot{\boldsymbol{\chi}}_{1,0}$ represents the rates of change of the position and velocity states at the end of the departure orbit propagation $\dot{\boldsymbol{\chi}}_{dep}$ as determined by the free variable ϖ_{dep}. Since only position, velocity and mass continuity are sought at each node in the CSI regime in this investigation,

$$\boldsymbol{\Phi}_{\mathbb{1}}(\mathbb{\tau}_1 + \varpi_1, \mathbb{\tau}_1) = \boldsymbol{\Phi}_{\mathbb{1}} = \begin{bmatrix} \boldsymbol{\Phi}_{1,\boldsymbol{rr}} & \boldsymbol{\Phi}_{1,\boldsymbol{rv}} \\ \boldsymbol{\Phi}_{1,\boldsymbol{vr}} & \boldsymbol{\Phi}_{1,\boldsymbol{vv}} \\ \boldsymbol{\Phi}_{1,m\boldsymbol{r}} & \boldsymbol{\Phi}_{1,m\boldsymbol{v}} \end{bmatrix} \tag{3.108}$$

The sensitivities associated with the thrust vector unit-magnitude constraint are developed as follows:

$$\mathfrak{J}_{\hat{\boldsymbol{u}},\boldsymbol{u}_i} = \frac{\partial(\boldsymbol{u}_i{}^T\boldsymbol{u}_i - 1)}{\partial \boldsymbol{u}_i} = 2\boldsymbol{u}_i \tag{3.109}$$

The complete Jacobian matrix for the CSI transfer (non-rendezvous) regime computing the derivatives of the constraints $\boldsymbol{F}_{CSI}$ with respect to the design variables $\boldsymbol{X}_{CSI}$ in Eqn. (3.106) is compiled as:

$$
\mathfrak{J}_{CSI} = \begin{array}{ccccccccccc|c}
\boldsymbol{u}_1 & \boldsymbol{\chi}_{CSI2} & \boldsymbol{\chi}_{CSI3} & \dots & \boldsymbol{\chi}_{LT,\mathbb{n}} & \varpi_{dep} & \varpi_{arr} & \varpi_1 & \varpi_2 & \dots & \varpi_{\mathbb{n}-1} & \\
\boldsymbol{\Phi}_1 & -\boldsymbol{I} & \mathbf{0} & \dots & \mathbf{0} & \boldsymbol{\Phi}_{\mathbb{1}}\dot{\boldsymbol{\chi}}_{dep} & \mathbf{0} & \dot{\boldsymbol{\chi}}_{1,f} & \mathbf{0} & \dots & \mathbf{0} & \boldsymbol{\chi}_{CSI,1_f} - \boldsymbol{\chi}_{CSI,2_0} \\
\mathbf{0} & \boldsymbol{\Phi}_2 & -\boldsymbol{I} & \dots & \vdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dot{\boldsymbol{\chi}}_{2,f} & \dots & \mathbf{0} & \boldsymbol{\chi}_{CSI,2_f} - \boldsymbol{\chi}_{CSI,3_0} \\
\vdots & \ddots & \ddots & \ddots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \vdots & \ddots & \ddots & \vdots & \vdots \\
\mathbf{0} & \dots & \mathbf{0} & \boldsymbol{\Phi}_{\mathbb{n}-1} & -\boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \dots & \dot{\boldsymbol{\chi}}_{\mathbb{n}-1,f} & \boldsymbol{\chi}_{LT,\mathbb{n}-1_f} - \boldsymbol{\chi}_{LT,\mathbb{n}} \\
\mathbf{0} & \dots & \dots & \dots & \boldsymbol{I} & \mathbf{0} & -\dot{\boldsymbol{\chi}}_{arr} & \mathbf{0} & \dots & \dots & \mathbf{0} & \boldsymbol{\chi}_{\mathbb{n}} - \boldsymbol{\chi}_{arr} \\
2\boldsymbol{u}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \dots & \dots & \dots & \dots & \dots & \mathbf{0} & \boldsymbol{u}_1{}^T\boldsymbol{u}_1 - 1 \\
\mathbf{0} & 2\boldsymbol{u}_2 & \mathbf{0} & \mathbf{0} & \dots & \dots & \dots & \dots & \dots & \dots & \mathbf{0} & \boldsymbol{u}_2{}^T\boldsymbol{u}_2 - 1 \\
\vdots & \mathbf{0} & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & 2\boldsymbol{u}_{\mathbb{n}-1} & \mathbf{0} & \dots & \dots & \dots & \dots & \mathbf{0} & \mathbf{0} & \boldsymbol{u}^T_{\mathbb{n}-1}\boldsymbol{u}_{\mathbb{n}-1} - 1
\end{array}
\tag{3.110}
$$

The rows of the matrix correspond to the constituents of the constraint vector $\boldsymbol{F}_{CSI}$ and the columns correspond to the components of the design vector $\boldsymbol{X}_{CSI}$; these values are highlighted in magenta in Eqn. (3.110). Note that the $2\boldsymbol{u}_i$ values in blue in the matrix contained in Eqn. (3.110) align with the $\boldsymbol{u}$ columns alone for a given node. Similarly, $\boldsymbol{I}$ in blue in Eqn. (3.110) aligns with the position and velocity columns alone for the last node. Again, a Nonlinear Programming (NLP) software such as MATLAB©'s `fmincon` or SNOPT [93] are adopted to optimize the converged solution obtained from the numerical corrections process.

Numerical Corrections Considerations for the CSI Regime — Rendezvous

Rendezvous scenarios aim to deliver a chaser s/c to a target (pre-positioned object) at the destination orbit. Although the EOMs, and the $\mathbf{A}_{\text{CSI}}$ matrix remain unmodified from Eqns. (3.103) and Eqns. (3.104) respectively, additional constraint measures are taken as follows to address sensitivities associated with the rendezvous TOF and arrival states.

$$\boldsymbol{X}_{CSI_{Rndv}} = \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{\chi}_{CSI\,2} \\ \vdots \\ \boldsymbol{\chi}_{CSI\,\mathbb{n}-1} \\ \boldsymbol{\chi}_{LT\,\mathbb{n}} \\ \varpi_{dep} \\ \varpi_{arr} \\ \varpi_1 \\ \vdots \\ \varpi_{\mathbb{n}-1} \end{bmatrix} \qquad \boldsymbol{F}_{CSI_{Rndzv}} = \begin{bmatrix} \boldsymbol{\chi}_{CSI,1_f} - \boldsymbol{\chi}_{CSI,2_0} \\ \vdots \\ \boldsymbol{\chi}_{LT,\mathbb{n}-1_f} - \boldsymbol{\chi}_{LT,\mathbb{n}} \\ \boldsymbol{\chi}_{\mathbb{n}} - \boldsymbol{\chi}_{arr} \\ {\boldsymbol{u}_1}^T \boldsymbol{u}_1 - 1 \\ \vdots \\ \boldsymbol{u}_{\mathbb{n}-1}^T \boldsymbol{u}_{\mathbb{n}-1} - 1 \\ \varpi_{dep} + \sum_{i=1}^{\mathbb{n}-1} \varpi_i - \varpi_{arr} \end{bmatrix} = \mathbf{0} \qquad (3.111)$$

The design and constraint vectors in Eqn. (3.111) for the rendezvous formulation assume that the target and chaser s/c begin at a reference state along their respective nominal paths at epoch $\mathbb{\tau}_0 = 0$. Note that $\boldsymbol{\chi}_{arr}$ represents the target object's state, and is a function of the duration of propagation along the arrival orbit from its reference state. So, as explored in the non-rendezvous CSI transfer scenario, the chaser aims to meet the target's 'state' via the state continuity constraint between the last node along the chaser's path ($\boldsymbol{\chi}_{\mathbb{n}}$) and the target's state $\boldsymbol{\chi}_{arr}$. However, given a rendezvous scenario, the time associated with the chaser's arrival at the target state $\boldsymbol{\chi}_{arr}$, must equal the target's epoch at this state as well. This goal is captured via the last time constraint in $\boldsymbol{F}_{CSI_{Rndzv}}$ in Eqn. (3.111). The Jacobian matrix components associated with the time constraint formulation presented in Eqn. (3.111) are computed as follows.

$$\mathfrak{J}_{Rndzv,\varpi_{dep}} = 1 \tag{3.112}$$

$$\mathfrak{J}_{Rndzv,\varpi_{i}} = 1 \tag{3.113}$$

$$\mathfrak{J}_{Rndzv,\varpi_{arr}} = -1 \tag{3.114}$$

And so, the Jacobian matrix constructed in Eqn. (3.110) is appended with one extra row to account for the sensitivities to the time constraint from variations in the initial conditions. The complete Jacobian matrix for the CSI rendezvous regime computing the derivatives of the constraints $\boldsymbol{F}_{CSI_{Rndzv}}$ with respect to the design variables $\boldsymbol{X}_{CSI_{Rndzv}}$ in Eqn. (3.111) is compiled as:

$$\Im_{CSI_{Rndzv}} = \begin{array}{c} \begin{array}{ccccccccccc} \boldsymbol{u}_1 & \boldsymbol{\chi}_{CSI_2} & \boldsymbol{\chi}_{CSI_3} & \dots & \boldsymbol{\chi}_{LT,\mathbb{n}} & \varpi_{dep} & \varpi_{arr} & \varpi_1 & \varpi_2 & \dots & \varpi_{\mathbb{n}-1} \end{array} \\ \left[\begin{array}{ccccccccccc} \boldsymbol{\Phi}_1 & -\boldsymbol{I} & \boldsymbol{0} & \dots & \boldsymbol{0} & \boldsymbol{\Phi}_{\mathbb{1}}\dot{\boldsymbol{\chi}}_{dep} & \boldsymbol{0} & \dot{\boldsymbol{\chi}}_{1,f} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Phi}_2 & -\boldsymbol{I} & \dots & \vdots & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \dot{\boldsymbol{\chi}}_{2,f} & \dots & \boldsymbol{0} \\ \vdots & \ddots & \ddots & \ddots & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \vdots & \ddots & \ddots & \vdots \\ \boldsymbol{0} & \dots & \boldsymbol{0} & \boldsymbol{\Phi}_{\mathbb{n}-1} & -\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \dots & \dots & \dot{\boldsymbol{\chi}}_{\mathbb{n}-1,f} \\ \boldsymbol{0} & \dots & \dots & \dots & \boldsymbol{I} & \boldsymbol{0} & -\dot{\boldsymbol{\chi}}_{arr} & \boldsymbol{0} & \dots & \dots & \boldsymbol{0} \\ 2\boldsymbol{u}_1 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \dots & \dots & \dots & \dots & \dots & \dots & \boldsymbol{0} \\ \boldsymbol{0} & 2\boldsymbol{u}_2 & \boldsymbol{0} & \boldsymbol{0} & \dots & \dots & \dots & \dots & \dots & \dots & \boldsymbol{0} \\ \vdots & \boldsymbol{0} & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & 2\boldsymbol{u}_{\mathbb{n}-1} & \boldsymbol{0} & \dots & \dots & \dots & \dots & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \dots & \boldsymbol{0} & 1 & -1 & 1 & 1 & \dots & 1 \end{array}\right] \end{array} \begin{array}{c} \\ \boldsymbol{\chi}_{CSI,1_f} - \boldsymbol{\chi}_{CSI,2_0} \\ \boldsymbol{\chi}_{CSI,2_f} - \boldsymbol{\chi}_{CSI,3_0} \\ \vdots \\ \boldsymbol{\chi}_{LT,\mathbb{n}-1_f} - \boldsymbol{\chi}_{LT,\mathbb{n}} \\ \boldsymbol{\chi}_{\mathbb{n}} - \boldsymbol{\chi}_{arr} \\ \boldsymbol{u}_1{}^T\boldsymbol{u}_1 - 1 \\ \boldsymbol{u}_2{}^T\boldsymbol{u}_2 - 1 \\ \vdots \\ \boldsymbol{u}_{\mathbb{n}-1}^T\boldsymbol{u}_{\mathbb{n}-1} - 1 \\ \varpi_{dep} + \sum_{i=1}^{\mathbb{n}-1} \varpi_i - \varpi_{arr} \end{array} \tag{3.115}$$

The rows of the matrix correspond to the constituents of the constraint vector $\boldsymbol{F}_{CSI_{Rndvz}}$ and the columns correspond to the components of the design vector $\boldsymbol{X}_{CSI_{Rndvz}}$; these values are highlighted in magenta in Eqn. (3.115). Note that the $2\boldsymbol{u}_i$ values in blue in the matrix contained in Eqn. (3.115) align with the $\boldsymbol{u}$ columns alone for a given node. Similarly, $\boldsymbol{I}$ in blue in Eqn. (3.115) aligns with the position and velocity columns alone for the last node. Note that an alternative formulation for the rendezvous constraint in $\boldsymbol{F}_{CSI_{Rndzv}}$ in Eqn. (3.111) is also possible. The duration along the arrival orbit ϖ_{arr}, may be excluded as a design variable from the vector $\boldsymbol{X}_{CSI_{Rndzv}}$ and, instead, the rendezvous state $\boldsymbol{\chi}_{arr}$, may be computed by propagating the target from its reference state for the TOF associated with the chaser's trajectory while accounting for its starting epoch as well, i.e. $\boldsymbol{\chi}_{arr}(\varpi_{dep} + \sum_{i=1}^{\mathsf{n}-1} \varpi_i)$ [78]. For such a formulation, the last rendezvous time constraint in $\boldsymbol{F}_{CSI_{Rndzv}}$ in Eqn. (3.111) is eliminated as it is unnecessary. A Nonlinear Programming (NLP) software such as MATLAB©'s `fmincon` or SNOPT [93] are adopted to optimize the converged solution obtained from the numerical corrections process.

3.2.3 Direct Optimization Scheme for the Chemical Engine Regime

No new forcing terms are augmented to the natural CR3BP dynamics to execute solutions for a chemical engine. Instead, construction of a converged/optimized trajectory enabled by a chemical engine simply allows velocity discontinuities at specified nodes; the following example allows velocity discontinuities at all nodes. So, the CR3BP EOMs and $\mathbf{A}$ matrix (Eqn. (3.7)) are applicable in chemical engine transfer scenarios as well. The numerical targeting approach to compute the design vector variables ($\boldsymbol{X}^*_{Chem}$) that satisfy the specified constraints, $\boldsymbol{F}(\boldsymbol{X}^*_{Chem}) = \boldsymbol{0}$ is similar to that incorporated for the low-thrust regimes and is developed as follows:

$$\boldsymbol{X}_{Chem} = \begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{\chi}_2 \\ \vdots \\ \boldsymbol{\chi}_{\mathbb{n}} \\ \varpi_{dep} \\ \varpi_{arr} \\ \varpi_1 \\ \vdots \\ \varpi_{\mathbb{n}-1} \end{bmatrix} \qquad \boldsymbol{F}_{Chem} = \begin{bmatrix} \boldsymbol{r}_{1f} - \boldsymbol{r}_{20} \\ \vdots \\ \boldsymbol{r}_{\mathbb{n}-1f} - \boldsymbol{r}_{\mathbb{n}0} \\ \boldsymbol{\chi}_{\mathbb{n}} - \boldsymbol{\chi}_{arr} \end{bmatrix} = \boldsymbol{0} \tag{3.116}$$

The design vector $\boldsymbol{X}_{Chem}$, is comprised of the states ($\boldsymbol{\chi}_i$) and propagation durations (ϖ_i) at select nodes (i). The constraint vector $\boldsymbol{F}_{Chem}$ is sought to meet position continuity along the trajectory and boundary conditions on the arrival orbit. If velocity continuity is also desired at certain nodes along the path, then, the state continuity constraints are modified accordingly to include the velocity states as well in Eqn. (3.116).

Deriving the Jacobian

The relevant Jacobian quantities derived in Sections 3.1.4, 3.2.1 and 3.2.2 are adopted to execute the targeting scheme with a chemical engine where velocity discontinuities are allowed within the trajectory. When velocity continuity is not sought at a particular node,

$$\mathfrak{J}_{S,\boldsymbol{\chi}_{i,0}} = \boldsymbol{\Phi}_i = \begin{bmatrix} \boldsymbol{\Phi}_{\boldsymbol{rr}} & \boldsymbol{\Phi}_{\boldsymbol{rv}} \end{bmatrix} \tag{3.117}$$

and,

$$\mathfrak{J}_{S,\varpi_i} = \dot{\boldsymbol{r}}. \tag{3.118}$$

Also, since the velocity components at the first node are incorporated as design variables in $\boldsymbol{X}_{Chem}$, the sub-matrix $\mathfrak{J}_{S(1f,\varpi_{dep})}$ is modified from Eqn. (3.107) to:

$$\mathfrak{J}_{S,\varpi_{dep}} = \boldsymbol{\Phi}_{\mathbb{1}} \boldsymbol{v}_{dep} = \boldsymbol{\Phi}_{\boldsymbol{rr}} \boldsymbol{v}_{dep} \tag{3.119}$$

where, $\boldsymbol{v}_{dep}$ is the velocity on the departure orbit as influenced by ϖ_{dep}. The complete Jacobian matrix computing the derivatives of the constraints $\boldsymbol{F}_{Chem}$ with respect to the design variables $\boldsymbol{X}_{Chem}$ in Eqn. (3.116) is compiled as:

$$\mathfrak{I}_{Chem} = \begin{array}{cc} \begin{array}{ccccccccccc} \boldsymbol{v}_1 & \boldsymbol{\chi}_2 & \boldsymbol{\chi}_3 & \dots & \boldsymbol{\chi}_{\mathbb{n}} & \varpi_{dep} & \varpi_{arr} & \varpi_1 & \varpi_2 & \dots & \varpi_{\mathbb{n}-1} \end{array} & \\ \left[\begin{array}{ccccccccccc} \boldsymbol{\Phi}_1 & -\boldsymbol{I} & \mathbf{0} & \dots & \mathbf{0} & \boldsymbol{\Phi}_{\mathbb{1}}\boldsymbol{v}_{dep} & \mathbf{0} & \dot{\boldsymbol{r}}_{1,f} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Phi}_2 & -\boldsymbol{I} & \dots & \vdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dot{\boldsymbol{r}}_{2,f} & \dots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{0} & \boldsymbol{\Phi}_{\mathbb{n}-1} & -\boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \dots & \dot{\boldsymbol{r}}_{\mathbb{n}-1,f} \\ \mathbf{0} & \dots & \dots & \dots & \boldsymbol{I} & \mathbf{0} & -\dot{\boldsymbol{\chi}}_{arr} & \mathbf{0} & \dots & \dots & \mathbf{0} \end{array}\right] & \begin{array}{c} \boldsymbol{r}_{1f} - \boldsymbol{r}_{20} \\ \boldsymbol{r}_{2f} - \boldsymbol{r}_{30} \\ \vdots \\ \boldsymbol{r}_{\mathbb{n}-1f} - \boldsymbol{r}_{\mathbb{n}0} \\ \boldsymbol{\chi}_{\mathbb{n}} - \boldsymbol{\chi}_{arr} \end{array} \end{array} \tag{3.120}$$

The rows of the matrix correspond to the constituents of the constraint vector $\boldsymbol{F}_{Chem}$ and the columns correspond to the components of the design vector $\boldsymbol{X}_{Chem}$; these values are highlighted in magenta in Eqn. (3.120). Again, a Nonlinear Programming (NLP) software such as MATLAB©'s `fmincon` or SNOPT [93] are adopted to optimize the converged solution obtained from the numerical corrections process. Since velocity discontinuities are allowed along the trajectory, the optimizer is set up to minimize the total maneuver requirements for a given transfer scenario.

3.3 Considerations in the Higher-Fidelity Ephmeris Regime

The ephemeris regime is considered in this investigation to validate the solutions generated in the lower-fidelity CR3BP. As a proof-of-concept, only the CSI-engine solutions generated in the CR3BP are transitioned to the higher-fidelity simulation and subjected to numerical corrections in this regime. Thus, the system of equations for the low-thrust augmented ephemeris model in the J2000 frame (Eqn. (2.55)) are nondimensionalized and further expanded to include the thrust magnitude and direction quantities for the CSI regime introduced in Section 3.2.2:

$$\dot{\boldsymbol{\chi}}_{EphemCSI} = \begin{Bmatrix} \dot{\boldsymbol{\mathfrak{r}}} \\ \dot{\boldsymbol{\mathfrak{v}}} \\ \dot{m} \\ \dot{\mathbb{T}} \\ \dot{\mathfrak{u}}_x \\ \dot{\mathfrak{u}}_y \\ \dot{\mathfrak{u}}_z \end{Bmatrix} \tag{3.121}$$

As discussed in Section 3.2.2, the thrust magnitude and directions are maintained in the EOMs for generality of derivation even though they are constant over a thrust arc. The resulting Jacobian matrix ($A_{11\times11}$) consisting of the partial derivatives of these equations with respect to the state variables is derived as:

$$A_{\text{EphemCSI},11\times11} = \begin{bmatrix} 0_{3\times3} & I_{3\times3} & 0_{3\times1} & 0_{3\times1} & 0_{3\times3} \\ \left(\frac{\partial\dot{\boldsymbol{\mathfrak{v}}}}{\partial\boldsymbol{\mathfrak{r}}}\right)_{3\times3} & 0_{3\times3} & -\frac{\mathbb{T}}{m^2}\hat{\boldsymbol{\mathfrak{u}}} & \frac{\hat{\mathfrak{u}}}{m} & \frac{\mathbb{T}}{m}I_{3\times3} \\ 0_{1\times3} & 0_{1\times3} & 0_{1\times1} & -\frac{1}{Isp\cdot g_0} & 0_{1\times3} \\ 0_{1\times3} & 0_{1\times3} & 0_{1\times1} & 0_{1\times1} & 0_{1\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times1} & 0_{3\times1} & 0_{3\times3} \end{bmatrix} \tag{3.122}$$

where,

$$\begin{aligned}
\left(\frac{\partial\dot{\boldsymbol{\mathfrak{v}}}}{\partial\boldsymbol{\mathfrak{r}}}\right)_{11} &= G(m_q+m_3)\left(\frac{3\mathtt{x}_{q3}^2}{\mathfrak{r}_{q3}^5}-\frac{1}{\mathfrak{r}_{q3}^3}\right)+G\sum_{\substack{j=1\\j\neq q,3}}^{\mathbb{N}} m_j\left(\frac{3\mathtt{x}_{3j}^2}{\mathfrak{r}_{3j}^5}-\frac{1}{\mathfrak{r}_{3j}^3}\right)\\
\left(\frac{\partial\dot{\boldsymbol{\mathfrak{v}}}}{\partial\boldsymbol{\mathfrak{r}}}\right)_{12} &= G(m_q+m_3)\left(\frac{3\mathtt{x}_{q3}\mathtt{y}_{q3}}{\mathfrak{r}_{q3}^5}\right)+G\sum_{\substack{j=1\\j\neq q,3}}^{\mathbb{N}} m_j+\left(\frac{3\mathtt{x}_{3j}\mathtt{y}_{3j}}{\mathfrak{r}_{3j}^5}\right)\\
\left(\frac{\partial\dot{\boldsymbol{\mathfrak{v}}}}{\partial\boldsymbol{\mathfrak{r}}}\right)_{13} &= G(m_q+m_3)\left(\frac{3\mathtt{x}_{q3}\mathtt{z}_{q3}}{\mathfrak{r}_{q3}^5}\right)+G\sum_{\substack{j=1\\j\neq q,3}}^{\mathbb{N}} m_j+\left(\frac{3\mathtt{x}_{3j}\mathtt{z}_{3j}}{\mathfrak{r}_{3j}^5}\right)\\
\left(\frac{\partial\dot{\boldsymbol{\mathfrak{v}}}}{\partial\boldsymbol{\mathfrak{r}}}\right)_{22} &= G(m_q+m_3)\left(\frac{3\mathtt{y}_{q3}^2}{\mathfrak{r}_{q3}^5}-\frac{1}{\mathfrak{r}_{q3}^3}\right)+G\sum_{\substack{j=1\\j\neq q,3}}^{\mathbb{N}} m_j\left(\frac{3\mathtt{y}_{3j}^2}{\mathfrak{r}_{3j}^5}-\frac{1}{\mathfrak{r}_{3j}^3}\right)\\
\left(\frac{\partial\dot{\boldsymbol{\mathfrak{v}}}}{\partial\boldsymbol{\mathfrak{r}}}\right)_{23} &= G(m_q+m_3)\left(\frac{3\mathtt{y}_{q3}\mathtt{z}_{q3}}{\mathfrak{r}_{q3}^5}\right)+G\sum_{\substack{j=1\\j\neq q,3}}^{\mathbb{N}} m_j+\left(\frac{3\mathtt{y}_{3j}\mathtt{z}_{3j}}{\mathfrak{r}_{3j}^5}\right)\\
\left(\frac{\partial\dot{\boldsymbol{\mathfrak{v}}}}{\partial\boldsymbol{\mathfrak{r}}}\right)_{33} &= G(m_q+m_3)\left(\frac{3\mathtt{z}_{q3}^2}{\mathfrak{r}_{q3}^5}-\frac{1}{\mathfrak{r}_{q3}^3}\right)+G\sum_{\substack{j=1\\j\neq q,3}}^{\mathbb{N}} m_j\left(\frac{3\mathtt{z}_{3j}^2}{\mathfrak{r}_{3j}^5}-\frac{1}{\mathfrak{r}_{3j}^3}\right)\\
\left(\frac{\partial\dot{\boldsymbol{\mathfrak{v}}}}{\partial\boldsymbol{\mathfrak{r}}}\right)_{21} &= \left(\frac{\partial\dot{\boldsymbol{\mathfrak{v}}}}{\partial\boldsymbol{\mathfrak{r}}}\right)_{12}\\
\left(\frac{\partial\dot{\boldsymbol{\mathfrak{v}}}}{\partial\boldsymbol{\mathfrak{r}}}\right)_{31} &= \left(\frac{\partial\dot{\boldsymbol{\mathfrak{v}}}}{\partial\boldsymbol{\mathfrak{r}}}\right)_{13}\\
\left(\frac{\partial\dot{\boldsymbol{\mathfrak{v}}}}{\partial\boldsymbol{\mathfrak{r}}}\right)_{32} &= \left(\frac{\partial\dot{\boldsymbol{\mathfrak{v}}}}{\partial\boldsymbol{\mathfrak{r}}}\right)_{23}
\end{aligned} \tag{3.123}$$

From Section 2.4, 'q' is the central body and the subscripts 'j' and '3' refer to the perturbing body and the spacecraft respectively. The Jacobian matrix facilitates the construction of the STM via:

$$\dot{\mathbf{\Phi}}_{EphemCSI}(\tau, \tau_0) = \mathbf{A}_{EphemCSI}(\tau)\mathbf{\Phi}_{EphemCSI}(\tau, \tau_0) \tag{3.124}$$

Unlike the implementations in the CR3BP, the transfers in the Ephemeris simulation do not depart or arrive at a particular closed orbit, as such orbits do not exist in this higher-fidelity regime. So, a *stacking* process is adopted to depart from/arrive in the vicinity of the original CR3BP departure and destination orbits. This process involves stacking revolutions of the departure and arrival orbits as illustrated in the schematic in Fig. 3.11. The states within the stacked orbits are discretized and appended on either end of the CR3BP transfer trajectory to produce a longer end-to-end path. The times are also adjusted accordingly to increase monotonically from the beginning of the departure stack and along the transfer that terminates at the end of the destination stack. Such a stacking process aims at maintaining the geometry of the original CR3BP transfer and the associated transfer performance metrics as the epoch-dependent quantities in the design vector $\boldsymbol{X}_{EphemCSI}$ vary during the numerical corrections process. The stack of discretized CR3BP states and times are then transformed into the J2000 frame as explored in Section 2.5.2 prior to conducting the numerical corrections process.

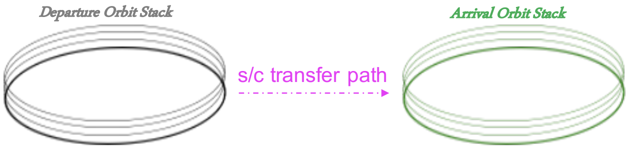


Figure 3.11: Schematic — Preparing the CR3BP solution for transition to the ephemeris model

The Newton-Raphson scheme adopting the free-variable and constraint formulation described in Section 3.1.4 is adopted to compute the design vector variables

($\boldsymbol{X}^*_{EphemCSI}$) that satisfy the specified constraints, $\boldsymbol{F}(\boldsymbol{X}^*_{EphemCSI}) = \boldsymbol{0}$. The design considerations for the targeting formulation in the Ephemeris regime are developed as follows:

$$\boldsymbol{X}_{EphemCSI} = \begin{bmatrix} \boldsymbol{\chi}_1 \\ \boldsymbol{\mathfrak{u}}_1 \\ \boldsymbol{\chi}_{EphemCSI_2} \\ \vdots \\ \boldsymbol{\chi}_{EphemLT_{\mathbb{n}}} \\ \mathbb{\tau}_2 \\ \vdots \\ \mathbb{\tau}_{\mathbb{n}} \\ \varpi_1 \\ \vdots \\ \varpi_{\mathbb{n}-1} \end{bmatrix} \quad \boldsymbol{F}_{EphemCSI} = \begin{bmatrix} \boldsymbol{\chi}_{EphemCSI,1_f} - \boldsymbol{\chi}_{EphemCSI,2_0} \\ \vdots \\ \boldsymbol{\chi}_{EphemLT,\mathbb{n}-1_f} - \boldsymbol{\chi}_{EphemLT,\mathbb{n}} \\ \boldsymbol{\chi}_{Ephem,\mathbb{n}} - \boldsymbol{\chi}_{\mathbb{d}} \\ \mathbb{\tau}_1 + \varpi_1 - \mathbb{\tau}_2 \\ \vdots \\ \mathbb{\tau}_{\mathbb{n}-1} + \varpi_{\mathbb{n}-1} - \mathbb{\tau}_n \\ {\boldsymbol{\mathfrak{u}}_1}^T \boldsymbol{\mathfrak{u}}_1 - 1 \\ \vdots \\ \boldsymbol{\mathfrak{u}}^T_{\mathbb{n}-1} \boldsymbol{\mathfrak{u}}_{\mathbb{n}-1} - 1 \end{bmatrix} = \boldsymbol{0} \tag{3.125}$$

The design vector $\boldsymbol{X}_{EphemCSI}$, is comprised of the states ($\boldsymbol{\chi}_{EphemCSI_i}$), epoch times ($\mathbb{\tau}_i$), and propagation durations (ϖ_i) at select nodes (i). Note that in this investigation, the epoch at the first node is held constant at a user-defined value, and the mass is held constant at a nondimensional value of unity. Thus, $\mathbb{\tau}_1$ and m_1 are not included in the design vector. As discussed in Section 3.2.2, the thrust magnitude is also not included in the design vector $\boldsymbol{\chi}_{EphemCSI}$. State and epoch continuity as well as thrust pointing constraints are enforced throughout the trajectory as per the constraint vector $\boldsymbol{F}_{EphemCSI}$. Note that the constraint $\boldsymbol{\chi}_{EphemCSI,\mathbb{n}} - \boldsymbol{\chi}_{\mathbb{d}}$ may be omitted from $\boldsymbol{F}_{EphemCSI}$ if only state continuity is sought and there is no requirement to meet specific terminal states $\boldsymbol{\chi}_{\mathbb{d}}$. The epoch at the final node $\mathbb{\tau}_{\mathbb{n}}$ and the epoch continuity constraint at the last node may also be omitted from the design and constraint vectors respectively, if there is no requirement to arrive at the destination condition at a pre-specified epoch. Recall that this last node is free-floating and not propagated in the problem set-up for this investigation.

Deriving the Jacobian

Many of the Jacobian sub-matrices computed in Sections 3.1.4 and 3.2.2 are applicable in the ephemeris regime as well. Unlike the CR3BP, the ephemeris model is not autonomous and so a variation in the epoch $\mathbb{T}_i$ leads to a variation in the gravitational forces imparted to the spacecraft and, thus, the states at the end of a propagated arc. The sensitivity of the state continuity constraints at the end of a propagation to variations in the epoch of the originating node is captured via the Jacobian matrix $\mathfrak{J}_{S,\mathbb{T}_i}$ computed as:

$$\mathfrak{J}_{S,\mathbb{T}_i} = \frac{d}{d\tau}\left(\frac{\partial \boldsymbol{\chi}_{EphemCSI,i,f}}{\partial \mathbb{T}_i}\right) = \frac{\partial}{\partial \mathbb{T}_i}\left(\frac{d\boldsymbol{\chi}_{EphemCSI,i,f}}{d\tau}\right) \tag{3.126}$$

where $\mathbb{T}_i$ is the epoch at a particular node, and the time along the trajectory is represented by $\tau \in [\mathbb{T}_i, \mathbb{T}_i + \varpi_i]$. The denominators are interchangeable in Eqn. (3.126) because τ and $\mathbb{T}_i$ are independent variables and enable the following calculations:

$$\mathfrak{J}_{S,\mathbb{T}_i} = \frac{\partial}{\partial \mathbb{T}_i}\left(\frac{d\boldsymbol{\chi}_{EphemCSI}}{d\tau}\right) = \frac{\partial}{\partial \mathbb{T}_i}\dot{\boldsymbol{\chi}}_{EphemCSI} \tag{3.127}$$

Recall that

$$\dot{\boldsymbol{\chi}}_{EphemCSI} = \boldsymbol{f}\left(\boldsymbol{\chi}_{EphemCSI}, \tau, \sum_{\substack{j=1\\ j\neq q,3}}^{\mathbb{N}} \boldsymbol{\mathfrak{r}}_{qj}(\mathbb{T}_i, \tau)\right) \tag{3.128}$$

and so $\dot{\boldsymbol{\chi}}_{EphemCSI}$ is a function of the states of the s/c, the integration time, and also the relative positions of the perturbing bodies j with respect to the primary body $\boldsymbol{\mathfrak{r}}_{qj}(\mathbb{T}_i, \tau)$. This knowledge is substituted into Eqn. (3.127) to yield

$$\mathfrak{J}_{S,\mathbb{T}_i} = \frac{\partial}{\partial \mathbb{T}_i}\boldsymbol{f}\left(\boldsymbol{\chi}_{EphemCSI}, \tau, \sum_{\substack{j=1\\ j\neq q,3}}^{\mathbb{N}} \boldsymbol{\mathfrak{r}}_{qj}(\mathbb{T}_i, \tau)\right) \tag{3.129}$$

$$= \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\chi}_{EphemCSI}}\frac{\partial \boldsymbol{\chi}_{EphemCSI}}{\partial \mathbb{T}_i} + \frac{\partial \boldsymbol{f}}{\partial \tau}\cancelto{0}{\frac{d\tau}{d\mathbb{T}_i}} + \sum_{\substack{j=1\\ j\neq q,3}}^{\mathbb{N}} \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\mathfrak{r}}_{qj}}\frac{d\boldsymbol{\mathfrak{r}}_{qj}}{d\mathbb{T}_i} \tag{3.130}$$

The term $\frac{\partial \tau}{\partial \mathbb{t}_i}$ is *zero* as the two variables are independent as discussed earlier. Also,

$$\frac{d\boldsymbol{\mathfrak{r}}_{qj}}{d\mathbb{t}_i} = \boldsymbol{\mathfrak{v}}_{qj}, \tag{3.131}$$

the velocity of body j relative to the primary body q, and is accessed via ephemeris data. The terms in $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\mathfrak{r}}_{qj}}$ represent the sensitivities in the s/c states to changes in the relative positions of the perturbing bodies from the primary body, and are computed via a straight-forward finite-differencing approach. And, since $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\chi}_{EphemCSI}}$ is recognized as the Jacobian matrix $\mathbf{A}_{EphemCSI}$,

$$\mathfrak{J}_{S,\mathbb{t}_i} = \frac{d}{d\tau}\left(\frac{\partial \boldsymbol{\chi}_{EphemCSI}}{\partial \mathbb{t}_i}\right) = \mathbf{A}_{EphemCSI}(\tau)\frac{\partial \boldsymbol{\chi}_{EphemCSI}}{\partial \mathbb{t}_i} + \sum_{\substack{j=1 \\ j\neq q,3}}^{\mathbb{N}} \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\mathfrak{r}}_{qj}} \frac{d\boldsymbol{\mathfrak{r}}_{qj}}{d\mathbb{t}_i} \tag{3.132}$$

Note that the value of $\frac{\partial \boldsymbol{\chi}_{EphemCSI}}{\partial \mathbb{t}_i}$ is available at any point along a trajectory by integrating Eqn. (3.132) along with the EOMs. The integration is initialized by setting:

$$\frac{\partial \boldsymbol{\chi}_{EphemCSI}}{\partial \mathbb{t}_i}(0) = \mathbf{0} \tag{3.133}$$

as the epoch $\mathbb{t}_i$ and state variables $\boldsymbol{\chi}_{EphemCSI}$ are independent variables.

Epoch continuity is also sought along the trajectory. The epoch continuity constraint at the end of a propagation from a particular node i (where $i \neq \mathbb{n}$) is extracted from the $\boldsymbol{F}(\boldsymbol{X})$ vector in Eqn. (3.45) as

$$\mathbb{t}_i + \varpi_i - \mathbb{t}_{i+1} \; = 0 \tag{3.134}$$

The partial derivatives of this constraint computed with respect to the variations in the epoch quantities yield:

$$\mathfrak{J}_{\mathbb{t},\mathbb{t}_i} = \frac{\partial(\mathbb{t}_i + \varpi_i - \mathbb{t}_{i+1})}{\partial \mathbb{t}_i} = 1 \tag{3.135a}$$

$$\mathfrak{J}_{\mathbb{t},\mathbb{t}_{i+1}} = \frac{\partial(\mathbb{t}_i + \varpi_i - \mathbb{t}_{i+1})}{\partial \mathbb{t}_{i+1}} = -1 \tag{3.135b}$$

The impact of variations in the propagation duration from an originating node ϖ_i on the epoch continuity constraint is also a straightforward computation:

$$\mathfrak{J}_{\mathbb{t},\varpi_i} = \frac{\partial(\mathbb{t}_i + \varpi_i - \mathbb{t}_{i+1})}{\partial \varpi_i} = 1 \tag{3.136}$$

The complete Jacobian matrix for the EphemCSI regime computing the derivatives of the constraints $\boldsymbol{F}_{EphemCSI}$ with respect to the design variables $\boldsymbol{X}_{EphemCSI}$ in Eqn. (3.125) is compiled as:

$$\mathfrak{I}_{EphemCSI} =$$

$$\begin{array}{cccccccccccccc|c}
[\boldsymbol{\chi}_1, \boldsymbol{\mathfrak{u}}_1] & \boldsymbol{\chi}_{EC2} & \boldsymbol{\chi}_{EC3} & \ldots & \boldsymbol{\chi}_{EL,\mathbb{n}} & \mathbb{\tau}_2 & \mathbb{\tau}_3 & \ldots & \ldots & \mathbb{\tau}_{\mathbb{n}} & \varpi_1 & \varpi_2 & \ldots & \varpi_{\mathbb{n}-1} & \\
\hline
\boldsymbol{\Phi}_1 & -\boldsymbol{I} & \ldots & \ldots & \mathbf{0} & \mathbf{0} & \ldots & \ldots & \ldots & \mathbf{0} & \dot{\boldsymbol{\chi}}_{1,f} & \ldots & \ldots & \mathbf{0} & \boldsymbol{\chi}_{EC,1_f} - \boldsymbol{\chi}_{EC,2_0} \\
\mathbf{0} & \boldsymbol{\Phi}_2 & -\boldsymbol{I} & \ldots & \mathbf{0} & \mathfrak{I}_{S,\mathbb{\tau}_2} & \ldots & \ldots & \ldots & \mathbf{0} & \mathbf{0} & \dot{\boldsymbol{\chi}}_{2,f} & \ldots & \mathbf{0} & \boldsymbol{\chi}_{EC,2_f} - \boldsymbol{\chi}_{EC,3_0} \\
\mathbf{0} & \ddots & \ddots & \ddots & \mathbf{0} & \mathbf{0} & \ddots & \ddots & \ddots & \mathbf{0} & \mathbf{0} & \ddots & \ddots & \vdots & \vdots \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\Phi}_{\mathbb{n}-1} & -\boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathfrak{I}_{S,\mathbb{\tau}_{\mathbb{n}-1}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dot{\boldsymbol{\chi}}_{\mathbb{n}-1,f} & \boldsymbol{\chi}_{EL,\mathbb{n}-1_f} - \boldsymbol{\chi}_{EL,\mathbb{n}} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\chi}_{E,\mathbb{n}} - \boldsymbol{\chi}_{\mathbb{d}} \\
\mathbf{0} & \ldots & \ldots & \ldots & \mathbf{0} & -1 & \ldots & \ldots & \ldots & \mathbf{0} & 1 & \ldots & \vdots & \mathbf{0} & \mathbb{\tau}_1 + \varpi_1 - \mathbb{\tau}_2 \\
\mathbf{0} & \ldots & \ldots & \ldots & \mathbf{0} & 1 & -1 & \ldots & \ldots & \mathbf{0} & \mathbf{0} & 1 & \ldots & \mathbf{0} & \mathbb{\tau}_2 + \varpi_2 - \mathbb{\tau}_3 \\
\mathbf{0} & \ddots & \ddots & \ddots & \mathbf{0} & \mathbf{0} & \ddots & \ddots & \ddots & \mathbf{0} & \mathbf{0} & \ddots & \ddots & \vdots & \vdots \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & 1 & -1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 1 & \mathbb{\tau}_{\mathbb{n}-1} + \varpi_{\mathbb{n}-1} - \mathbb{\tau}_n \\
2\boldsymbol{\mathfrak{u}}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\mathfrak{u}}_1{}^T \boldsymbol{\mathfrak{u}}_1 - 1 \\
\mathbf{0} & 2\boldsymbol{\mathfrak{u}}_2 & \ldots & \ldots & \mathbf{0} & \mathbf{0} & \ldots & \ldots & \ldots & \mathbf{0} & \mathbf{0} & \ldots & \ldots & \mathbf{0} & \boldsymbol{\mathfrak{u}}_2{}^T \boldsymbol{\mathfrak{u}}_2 - 1 \\
\mathbf{0} & \ddots & \ddots & \ddots & \mathbf{0} & \mathbf{0} & \ldots & \ldots & \ldots & \mathbf{0} & \mathbf{0} & \ddots & \ddots & \vdots & \vdots \\
\mathbf{0} & \ldots & \ldots & 2\boldsymbol{\mathfrak{u}}_{\mathbb{n}-1} & \mathbf{0} & \mathbf{0} & \ldots & \ldots & \ldots & \mathbf{0} & \mathbf{0} & \ldots & \ldots & \mathbf{0} & \boldsymbol{\mathfrak{u}}_{\mathbb{n}-1}^T \boldsymbol{\mathfrak{u}}_{\mathbb{n}-1} - 1
\end{array} \tag{3.137}$$

The rows of the matrix correspond to the constituents of the constraint vector $\boldsymbol{F}_{EphemCSI}$ and the columns correspond to the components of the design vector $\boldsymbol{X}_{EphemCSI}$; these values are highlighted in magenta in Eqn. (3.137). Also, in Eqn. (3.137), the subscripts $Ephem$, $EphemLT$ and $EphemCSI$ have been abbreviated to E, EL and EC, respectively, for document formatting purposes. Note that only the s/c position and velocity states are varied on the natural orbit 'stack' portions of an ephemeris transfer; however, the subscript EC for $EphemCSI$ has been maintained even for these initial nodes and the corresponding state continuity constraints in Eqn. (3.137) for generality. The $2\mathfrak{u}_i$ values in blue in the matrix contained in Eqn. (3.137) align with the $\mathfrak{u}$ columns alone for a given node. Similarly, $\boldsymbol{I}$ in blue in Eqn. (3.137) aligns with the position and velocity columns alone for the last node. A Nonlinear Programming (NLP) software such as MATLAB©'s `fmincon` or SNOPT [93] can be adopted to optimize the converged solutions if desired.

4. MACHINE LEARNING STRATEGIES

The creation of an initial guess for a reference solution prior to numerical corrections can be a labor intensive process dependent upon human intuition. So, automated processes to develop and curate effective initial guesses are sought to aid in the exploration of large and complex trade-spaces. In this investigation, the generation of an initial guess for a trajectory that transfers a s/c from a departure location to a specific destination is cast as a routing problem. Such a route is constructed from the appropriate sequencing of available waypoints in the design space to meet global objectives. Thus, the design process is predicated on the ability to recognize desirable waypoints and then construct pathways between them. Techniques from the discipline of Machine Learning (ML), a data-driven approach within the field of Artificial Intelligence (AI) to *learn* complex and nonlinear relationships, are adopted to address these separate design considerations. Specifically, traversal towards a destination is achieved by exploiting Reinforcement Learning (RL) pathfinding methods to strategically assemble advantageous intermediate arcs. These arcs/waypoints are made available for selection by the pathfinding algorithms via random choice or Supervised Learning (SL) strategies. The applicability of graph-search methods, such as Dijkstra's algorithm from the field of AI, to s/c pathfinding is also investigated.

4.1 Supervised Learning Algorithms

Supervised learning algorithms are often employed for pattern recognition to aid in data classification and as function approximations to implement regression tasks. A regression model seeks to approximate a function that maps inputs to continuous output variables; classification models map the inputs to discrete output *classes*. Since the 1950s, numerous algorithms have emerged to achieve these goals, where these algorithms are generally categorized into *generative* and *discriminative* strategies. Generative methods assume a probabilistic model of the joint probability distribution for the inputs ($\mathbb{x}$) conditioned on the output classes ($\mathbb{y}$), i.e., $P(\mathbb{x}|\mathbb{y})P(\mathbb{y})$, and exploit Bayes Rule to inform the most likely class to which data point $\mathbb{x}$ belongs via the posterior probability $P(\mathbb{y}|\mathbb{x})$ [96]:

$$P(\mathbb{y}|\mathbb{x}) = \frac{P(\mathbb{x}|\mathbb{y})P(\mathbb{y})}{P(\mathbb{x})} \tag{4.1}$$

Generative models are, however, susceptible to the prior belief $P(\mathbb{y})$ imposed regarding the distribution of the output classes $\mathbb{y}$, and, furthermore, incorrect modeling assumptions on the likelihood distribution $P(\mathbb{x}|\mathbb{y})$ can lead to inaccurate class predictions $P(\mathbb{y}|\mathbb{x})$ as well. In contrast, discriminative methods predict the posterior probability $P(\mathbb{y}|\mathbb{x})$ or a score indicating the output class from the data samples by 'direct' modulation of a decision boundary via function approximations. Since this approach does not require any probabilistic insights (that could be flawed) into the nature of data, in general, discriminative methods are generally more robust than generative ones. Thus, discriminative methods are employed in this investigation. The particular algorithm selected from within the discriminative group depends greatly on the unique strengths to address the nature, size, and complexity of the problem and the ability to balance the desired levels of accuracy and computational efficiency. In this investigation, both Artificial Neural Networks (ANNs) and Support Vector Machines (SVMs) are used to identify complex relationships in a highly nonlinear and high-dimensional multi-body regime. Both techniques *learn* from samples of input data paired with the corresponding categorical assignments (labels) or continuous outputs during a training phase. Successful training enables inferences concerning the underlying functions relating the variables in the design space and, subsequently, leads to the satisfactory prediction of outputs when applied to previously unseen datasets.

4.1.1 Artificial Neural Networks

Artificial Neural Networks are a collection of algorithms inspired by the functioning of biological neural networks to solve complex problems. Their inception lies in the *perceptron*, an artificial adaptation of a single biological neuron, first examined by Frank Rosenblatt in the late 1950s [97]. In the 1960s, the perceptron's inability to process data that is not linearly separable temporarily halted progress [98]. A decade later, the solution to overcome this challenge by introducing at least one extra layer of artificial neurons evolved into multi-layer feedforward networks and, thus, ‘deep learning’. The most basic feed-forward network consists of an input, hidden and output layer; the schematic of a slightly more sophisticated 3-layer network (2 hidden layers) is portrayed in Fig. 4.1. Similar to human brains, the iterative and dynamic strengthening of signals between neurons enable the identification of important relationships to process the data and report results in the output layer. In an ‘$\mathbb{o}$’-layer network, the output or logit $\mathbb{z}_j$, of the j^{th} neuron in layer l is formulated as the inner product between the signal strengths from neurons i, in previous layer $l-1$ (captured via the weights $\mathbb{w}_{i,j}$), and the inputs to it $\mathbb{x}_{i,j}$ [100]:

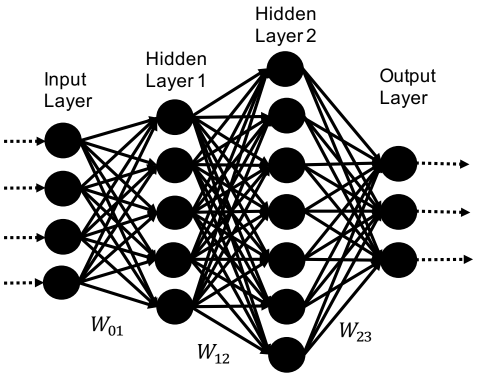


Figure 4.1: Sample Artificial Neural Network Architecture *(Modified from Bapu [99])*

$$\mathbb{z}_j^{(l)} = \sum_{i=1}^{\mathbb{h}} \mathbb{w}_{i,j} \cdot \mathbb{x}_{i,j} + \mathbb{b}_j \tag{4.2}$$

Here, $\mathbb{b}$ is a constant bias term that identifies the range of inputs that exercise the greatest influence over the outputs of a neuron j which belongs to the set $j = 1...\mathbb{l}$. The cardinality of this set $\mathbb{l}$, represents the total number of neurons in the layer l, and $\mathbb{h}$ is the total number of neurons in the layer $l - 1$. The output from a hidden layer is modeled as the matrix operation:

$$\mathbb{Z}^{(l)} = \mathbb{W}^T \mathbb{X} + \mathbb{B}^{(l)} \tag{4.3}$$

where, $\mathbb{W}$ is an $[\mathbb{h} \times \mathbb{l}]$ matrix, $\mathbb{X}$ is an $[\mathbb{h} \times 1]$ vector, and $\mathbb{Z}$ and $\mathbb{B}$ are $[\mathbb{l} \times 1]$ vectors. Accurate predictions are achieved via iterative tuning of the weighting matrices and bias terms. Various techniques to accomplish this goal render different varieties of ANNs — a detailed explanation of different ANN approaches and their differing purposes is presented in Haykin [101]. A feed-forward network suffices to address the goals of this preliminary investigation. In these networks, the data is always propagated forward from the input layer through the hidden layers towards the output layer; i.e., the data is not cycled back through any layers that have already been passed. Also, no links exist between neurons within a given layer, and there is no requirement to enforce an equal number of neurons within each hidden layer as well. The choices involved in building an ANN architecture play a key role in its performance and applicability for a given scenario. These choices comprise the identification of useful 'features' within a dataset to aid in pattern recognition, the assessment of the quality and quantity of the data incorporated during ANN training, and measures to develop a generalizable trained ANN model that accurately captures the underlying complex relationships within the dataset. These considerations are summarized as follows [102].

A **Feature and Label Initialization**

In a supervised learning application, a neural network is most commonly trained to *learn* a function (regression network) or classify the data into categories (classification network) recognized by their *labels*. Thus, careful selection of the user-specified inputs (features) at the input layer as well as the corresponding user-specified labels or continuous values from the output layer ‘$\mathbb{o}$’, greatly influence the *learning* exhibited by the neurons; the efficiency of the networks is impacted as well. The multiplicative and additive operations of the weighted values through each hidden layer also yield fresh features within the hidden layers, introducing additional insights for the network.

B **Data Allocation**

Within a sample-based platform, the quantity of data to which a network is exposed and the proportion of available points from each data-type to be classified/regressed also impact the training outcomes. The *goodness* of a trained network is assessed via the validation and testing phases on a subset of the supplied training data. So, appropriate allocation within a larger dataset for the training, validation, and testing phases is an essential consideration. A common approach is to initiate this partition by setting aside 70% of the data for *training*, 20% of the data for *validation*, and 10% of the data for *testing* purposes. However, these proportions are best determined based on empirical testing for a particular scenario. The quality of the data in the training set, including any missing data points, are also critical considerations during training a network.

C **Training Components**

Training an ANN involves continual updates for the weights (signal strengths) to identify critical relationships within the dataset. This goal is achieved as a function of the techniques flowing the information forward through the hidden layers (activation functions), and those tracing the errors backwards (backpropagation) to adjust the associated weight parameters accordingly. Both aspects are explored as follows.

C.a. Activation Function

In a biological network, a neuron fires only after the received signal strength exceeds a threshold value. An artificial neuron’s logit is, thus, passed through an activation function to help emulate this firing behavior and introduce nonlinearity into the outputs to enable the learning of complex relationships. Equation (4.3) is then modified as follows to compute a layer's output:

$$\mathbb{Y}^{(l)} = \boldsymbol{f}(\mathbb{W}^T\mathbb{X} + \mathbb{B}^{(l)}) = \boldsymbol{f}(\mathbb{Z}^{(l)}) \tag{4.4}$$

The frequency with which the spikes (outputs) are released from a neuron is determined by the activation function, $\boldsymbol{f}(\mathbb{Z})$. The nuances associated with a particular learning problem guide the selection of this function. Although a variety of options are available, the linear, hyperbolic tangent, and softmax functions have been incorporated for their relevance to this investigation. These activation functions are readily available within MATLAB©'s Math, Statistics and Optimization toolbox.

Linear Function (Regression model output layer)

A linear activation function is often incorporated as the output layer in regression networks, where the expected output from a given neuron can be unbounded. This function is represented as:

$$\boldsymbol{f}(\mathbb{z}) = \mathtt{C}\mathbb{z} \tag{4.5}$$

where $\mathtt{C}$ is a constant, and the outputs of the activation function are merely a linear transformation of the inputs to the neuron. Since this function is not capable of representing complex nonlinear relationships, it is typically not employed within the hidden layers of an ANN.

Hyperbolic Tangent Function (Hidden layers)

The *tanh* function introduces nonlinearities to the logit and, thus, the ability of the ANN to learn complex relationships as the data passes through the hidden layers. This function scales the standard logistic sigmoid function to be centered around a zero-mean and bounds a neuron's output $\mathbb{y}$ to remain within the range $[-1, 1]$ that aids with the numerical process during training. This function is mathematically formulated as:

$$\boldsymbol{f}(\mathbb{z}) = \frac{e^{2\mathbb{z}} - 1}{e^{2\mathbb{z}} + 1}. \tag{4.6}$$

When employing the *tanh* function, the magnitude of a neuron's output is therefore small for lower logit values and close to unity for larger logit values. This symmetric function possesses steep gradients around zero mean, aiding convergence [103].

Softmax Function (Classfication model output layer)

A softmax function, usually employed in the output layer, parses the knowledge gained through the network into a probability distribution over $\mathbb{C}$ outcomes (classes). This function is modeled as:

$$\boldsymbol{f}(\mathbb{z}_{\iota}) = \frac{e^{\mathbb{z}_{\iota}}}{\sum_{\mathbb{c}=1}^{\mathbb{C}} e^{\mathbb{z}_{\mathbb{c}}}} \tag{4.7}$$

where, $\iota = 1, 2, 3, ...\mathbb{C}$. The sum of the outputs of the neurons in the layer add to 1; such a rule is useful for classification problems, as higher probabilities are associated with strong predictions about an input belonging to a certain class [104]. However, this approach must be exercised with caution as the presence of a datapoint that does not belong to any of the expected classes can lead to false conclusions.

C.b. Error Minimization

During the training phase, appropriate activation functions facilitate the prediction of real values or classes at the output layer for a regression and classification network, respectively. The aim of the training phase, then, is to modulate the weights $\mathbb{W}$ over multiple iterations to minimize the error between the predicted and true values at the output layer. *Backpropagation* algorithms offer feedback to the activated neurons and, thus, create an opportunity for the weights (signal strengths) to be updated prior to the information being passed forward through the layers in the next iteration. The updates to the weights are informed by the rate of change of the error E between the true solution ($\mathbb{Y}_t$) and the ANN prediction ($\mathbb{Y}_p$) at the output layer ($\mathbb{o}$), with respect to the weights in each layer, $\mathbb{w}_{i,j}^{(l)}$:

$$\Delta \mathbb{w}_{i,j}^{(l)} = -\jmath \frac{\partial E}{\partial \mathbb{w}_{i,j}^{(l)}} \tag{4.8}$$

Here, $\jmath$ is a hyper-parameter denoted the 'learning rate' that is tunable to aid with the training phase [100]. The goal is a maneuver strategy towards the global minimum of a multi-dimensional error surface defined by the weight coordinates; a sample error surface appears in Fig. 4.2. A variety of optimization algorithms may be considered to compute the global minimum on an error surface, where Haykin [101] offers an in-depth discussion on this topic. A gradient descent approach, such as the steepest-descent algorithm, suffers from fluctuations in the path traversed towards the global minimum. This challenge is mitigated by a second order method such as the traditional Newton's method that exploits the curvature of the error surface by computing the Hessian. However, saddle-point-type conditions and potential ill-conditioning of the Hessian matrix in a high-dimensional design space results in poor convergence properties for this root-finding method [100–102]; it is also computationally intensive to compute the Hessian matrix. Such drawbacks are mitigated by leveraging the Levenberg-Marquardt algorithm that incorporates approximate Hessian information [101], as well as the less memory-intensive Scaled Conjugate Gradient (SCG) algorithm that incorporates Hessian-derivable information without actual computation of the Hessian explicitly [105]. These algorithms are readily available

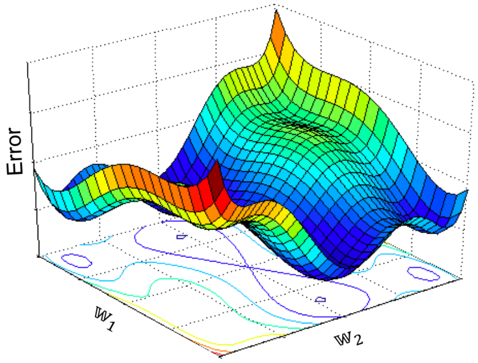


Figure 4.2: Example of an error surface as a function of arbitrary weights w_1 and w_2 with multiple local minima. The cooler colors indicate smaller errors. *(Modified from https://i.stack.imgur.com/TY1L1.png)*

in MATLAB©'s optimization toolbox. The appropriate choice of optimization technique to support the backpropagation effort to minimize prediction errors in an ANN is problem-specific and influences the values of the weight updates in Eqn. (4.8) as well. Also, the error function E in Eqn. (4.8) depends on the activation function employed in the output layer and, thus, the problem type. The error/loss functions employed in this investigation are outlined in the following discussions.

Softmax Classification Output Layer and Cross-Entropy Loss

Classification problems often assess the probability distribution over the output classes computed by neurons to categorize datapoints. The predicted probability distribution for the output of a neuron ι at the softmax output layer from Eqn. (4.7) is set equal to the predicted output class values, $\mathbb{y}_{p_\iota}$:

$$\mathbb{y}_{p_\iota}^{(\mathbb{o})} = \boldsymbol{f}(\mathbb{z}_\iota^{(\mathbb{o})}) \tag{4.9}$$

Cross-entropy loss (error) is often implemented for classification problems where the information from a softmax layer is representative of a probability distribution [106] and is defined at the output layer $(\mathbb{o})$ as:

$$E = -\sum_{\iota=1}^{\mathbb{C}} \mathbb{y}_{t_\iota}^{(\mathbb{o})} log(\mathbb{y}_{p_\iota}^{(\mathbb{o})}), \tag{4.10}$$

where $\mathbb{y}_{t_\iota}$ are the true probabilities of belonging to a particular class. Note that binary classification is a particular example of the multi-class classification problem. The process of weight updates in each layer begins with establishing a relationship between the error and the logit at the output layer [107]:

$$\frac{\partial E}{\partial \mathbb{z}_\iota^{(\mathbb{o})}} = -\sum_{j=1}^{\mathbb{C}} \frac{\partial E}{\partial \mathbb{y}_{p_j}^{(\mathbb{o})}} \frac{\partial \mathbb{y}_{p_j}^{(\mathbb{o})}}{\partial \mathbb{z}_\iota^{(\mathbb{o})}} \tag{4.11}$$

where,

$$\frac{\partial E}{\partial \mathbb{y}_{p_\iota}^{(\mathbb{o})}} = -\frac{\mathbb{y}_{t_\iota}^{(\mathbb{o})}}{\mathbb{y}_{p_\iota}^{(\mathbb{o})}} \tag{4.12}$$

Also, note that the derivative of the softmax function in Eqn. (4.7) with respect to Eqn. (4.9) yields:

$$\begin{aligned} \frac{\partial \mathbb{y}_{p_\iota}^{(\mathbb{o})}}{\partial \mathbb{z}_j^{(\mathbb{o})}} &= \begin{cases} \frac{e^{\mathbb{z}_\iota}}{\sum_{\mathbb{c}=1}^{\mathbb{C}} e^{\mathbb{z}_\mathbb{c}}} - \left(\frac{e^{\mathbb{z}_\iota}}{\sum_{\mathbb{c}=1}^{\mathbb{C}} e^{\mathbb{z}_\mathbb{c}}}\right)^2 & \iota = j \\ -\frac{e^{\mathbb{z}_\iota} e^{\mathbb{z}_j}}{\left(\sum_{\mathbb{c}=1}^{\mathbb{C}} e^{\mathbb{z}_\mathbb{c}}\right)^2} & \iota \neq j \end{cases} \\ &= \begin{cases} \mathbb{y}_{p_\iota}^{(\mathbb{o})}(1 - \mathbb{y}_{p_j}^{(\mathbb{o})}) & \iota = j \\ -\mathbb{y}_{p_\iota}^{(\mathbb{o})} \mathbb{y}_{p_j}^{(\mathbb{o})} & \iota \neq j \end{cases} \end{aligned} \tag{4.13}$$

Thus, Eqn. (4.11) is re-written as:

$$\begin{aligned} \frac{\partial E}{\partial \mathbb{z}_\iota^{(\mathbb{o})}} &= \frac{\partial E}{\partial \mathbb{y}_{p_\iota}^{(\mathbb{o})}} \frac{\partial \mathbb{y}_{p_\iota}^{(\mathbb{o})}}{\partial \mathbb{z}_\iota^{(\mathbb{o})}} + \sum_{j \neq \iota} \frac{\partial E}{\partial \mathbb{y}_{p_j}} \frac{\partial \mathbb{y}_{p_j}}{\partial \mathbb{z}_\iota^{(\mathbb{o})}} \\ &= -\mathbb{y}_{t_\iota}^{(\mathbb{o})}(1 - \mathbb{y}_{p_\iota}^{(\mathbb{o})}) + \sum_{j \neq \iota} \mathbb{y}_{t_j}^{(\mathbb{o})} \mathbb{y}_{p_\iota}^{(\mathbb{o})} \\ &= -\mathbb{y}_{t_\iota}^{(\mathbb{o})} + \mathbb{y}_{t_\iota}^{(\mathbb{o})} \mathbb{y}_{p_\iota}^{(\mathbb{o})} + \sum_{j \neq \iota} \mathbb{y}_{t_j}^{(\mathbb{o})} \mathbb{y}_{p_\iota}^{(\mathbb{o})} \\ &= -\mathbb{y}_{t_\iota}^{(\mathbb{o})} + \sum_{j=1}^{\mathbb{C}} \mathbb{y}_{t_j}^{(\mathbb{o})} \mathbb{y}_{p_\iota}^{(\mathbb{o})} \\ &= -\mathbb{y}_{t_\iota}^{(\mathbb{o})} + \mathbb{y}_{p_\iota}^{(\mathbb{o})} \sum_{j=1}^{\mathbb{C}} \mathbb{y}_{t_j}^{(\mathbb{o})} \end{aligned} \tag{4.14}$$

As the term $\sum_{j=1}^{\mathbb{C}} \mathbb{y}_{t_j}^{(\mathbb{o})}$ in Eqn. 4.14 equals 1,

$$\frac{\partial E}{\partial \mathbb{z}_\iota^{(\mathbb{o})}} = \mathbb{y}_{p_\iota}^{(\mathbb{o})} - \mathbb{y}_{t_\iota}^{(\mathbb{o})} \tag{4.15}$$

A simple error computation between the true and predicted probability distributions across the output classes at the output layer thus informs the classification ANN training process.

Linear Regression Output Layer and Mean Square Error

A regression ANN predicts values of continuous variables by learning complex function approximations via an iterative training process. These values that are output from a neuron φ at the final linear ANN layer (output layer) are characterized as:

$$\mathbb{y}_{p_\varphi}^{(\mathbb{O})} = f(\mathbb{z}_{\varphi}^{(\mathbb{O})}) = \mathbb{z}_{\varphi}^{(\mathbb{O})} \tag{4.16}$$

The Mean Square Error (MSE) is appropriate for the continuous and unbounded variable outputs associated with a linear output layer for regression.

$$E = \frac{\sum_{\varphi=1}^{\Upsilon}(\mathbb{y}_{t_\varphi}^{(\mathbb{O})} - \mathbb{y}_{p_\varphi}^{(\mathbb{O})})^2}{\Upsilon} \tag{4.17}$$

The rate of change of the error with respect to the within-layer logit is then:

$$\begin{aligned}
\frac{\partial E}{\partial \mathbb{z}_{\varphi}^{(\mathbb{O})}} &= \frac{\partial E}{\partial \mathbb{y}_{p_\varphi}^{(\mathbb{O})}} \frac{\partial \mathbb{y}_{p_\varphi}^{(\mathbb{O})}}{\partial \mathbb{z}_{\varphi}^{(\mathbb{O})}} \\
\frac{\partial E}{\partial \mathbb{y}_{p_\varphi}^{(\mathbb{O})}} &= -\frac{2}{\Upsilon}(\mathbb{y}_{t_\varphi}^{(\mathbb{O})} - \mathbb{y}_{p_\varphi}^{(\mathbb{O})}) \\
\frac{\partial \mathbb{y}_{p_\varphi}^{(\mathbb{O})}}{\partial \mathbb{z}_{\varphi}^{(\mathbb{O})}} &= 1
\end{aligned} \tag{4.18}$$

Equation (4.18) can be re-written as:

$$\frac{\partial E}{\partial \mathbb{z}_{\varphi}^{(\mathbb{O})}} = \phi(\mathbb{y}_{p_\varphi}^{(\mathbb{O})} - \mathbb{y}_{t_\varphi}^{(\mathbb{O})}) \tag{4.19}$$

where, $\phi = \frac{2}{\Upsilon}$. Thus, the regression ANN training is informed by a function comprising the error between the true and predicted values of the continuous variables at the output layer.

Backpropagation of the Error to Hidden Layers

Equations (4.15) and (4.19) describe the error with respect to the logit at the last or ouput layer. This error is propagated back through to the input layer to trace the influence of the weights in the hidden layers that lead to the final error, E_{total} [101].

For example, the relationship between the error E_{total} at the output layer $\mathbb{o}$ with $\mathrm{k} = 1...\mathbb{K}$ neurons, and the weight $\mathbb{w}_{i,j}$ to a neuron j in the final hidden layer l from a neuron i in a preceding layer $(l-1)$ is captured via partial derivatives as follows:

$$\frac{\partial E_{\text{total}}}{\partial \mathbb{w}_{i,j}^{(l)}} = \left(\sum_{\mathrm{k}=1}^{\mathbb{K}} \frac{\partial E_{\text{total}}}{\partial \mathbb{y}_{p,\mathrm{k}}^{(\mathbb{o})}} \frac{\partial \mathbb{y}_{p,\mathrm{k}}^{(\mathbb{o})}}{\partial \mathbb{z}_{\mathrm{k}}^{(\mathbb{o})}} \frac{\partial \mathbb{z}_{\mathrm{k}}^{(\mathbb{o})}}{\partial \mathbb{y}_{p,j}^{(l)}}\right) \frac{\partial \mathbb{y}_{p,j}^{(l)}}{\partial \mathbb{z}_{j}^{(l)}} \frac{\partial \mathbb{z}_{j}^{(l)}}{\partial \mathbb{w}_{i,j}^{(l)}} \tag{4.20}$$

Note that the predicted output from a layer $\mathbb{y}_{p,j}^{(l)}$, is equivalent to the input entering the next layer $\mathbb{x}_{j,\mathrm{k}}^{(\mathbb{o})}$. The derivatives are computed via the relationships previously defined. The relationship computed in Eqn. (4.20) is substituted back into the weight-change computation in Eqn. (4.8) at every epoch (iteration) to update the value of each $\mathbb{w}_{i,j}$ so as to minimize E_{total}.

D **Generalizability**

The *Universal Approximation Theory*, first proved in the 1980s by Cybenko [108], demonstrated that a single hidden layer could approximate any continuous function. This realization was extended by Funahoshi [109] in 1989 to clarify that a combination of hidden layers and neurons can be exploited by an ANN architecture to approximate any continuous function. However, more neurons/layers do not necessarily result in better approximations. In fact, the challenge in *bias-variance* trade-off necessitates a careful selection of the *size* of the ANN [110]. For example, a small network may result in insufficient parameters to satisfactorily capture the underlying complexities of the problem leading to a high bias and under-fitting of the relationships in the training set. In contrast, a large network can tend to over-fit the function, thus, leading to modeling even the noise in the dataset and exhibiting high variance when subject to new and unseen data. The goal is, therefore, generalizability — the ability to strike a balance between adequate learning of the underlying relationships on known data and adequate predictions of the relationships on unmodeled data. One approach to the iterative training process to establish a generalizable ANN model is summarized in Fig. 4.3.

An appropriate network size that balances the speed of training, accuracy and generalizability of the model is most successful. Various forms of *regularization* [100] techniques, such as ‘early stopping’, are useful measures to mitigate the over-fitting challenge [101]. This technique validates the model periodically with data that is ‘unseen’ during training, and terminates the training when the validation accuracy begins to stagnate or increase even as the training accuracy continues to decrease. Other valid regularization approaches are also applicable, as detailed by Bishop [111] and Goodfellow [100]. Upon completion/termination of the training phase, if the trained ANN is unable to make acceptable predictions on either the base training data or a further set of unseen test data, then the network is re-visited to adjust factors such as the number of hidden layers, number of neurons in these layers, and the choice of activation functions, regularization techniques, and backpropagation algorithms. The quantity and quality of the training data introduced to the ANN may also require reconsideration.

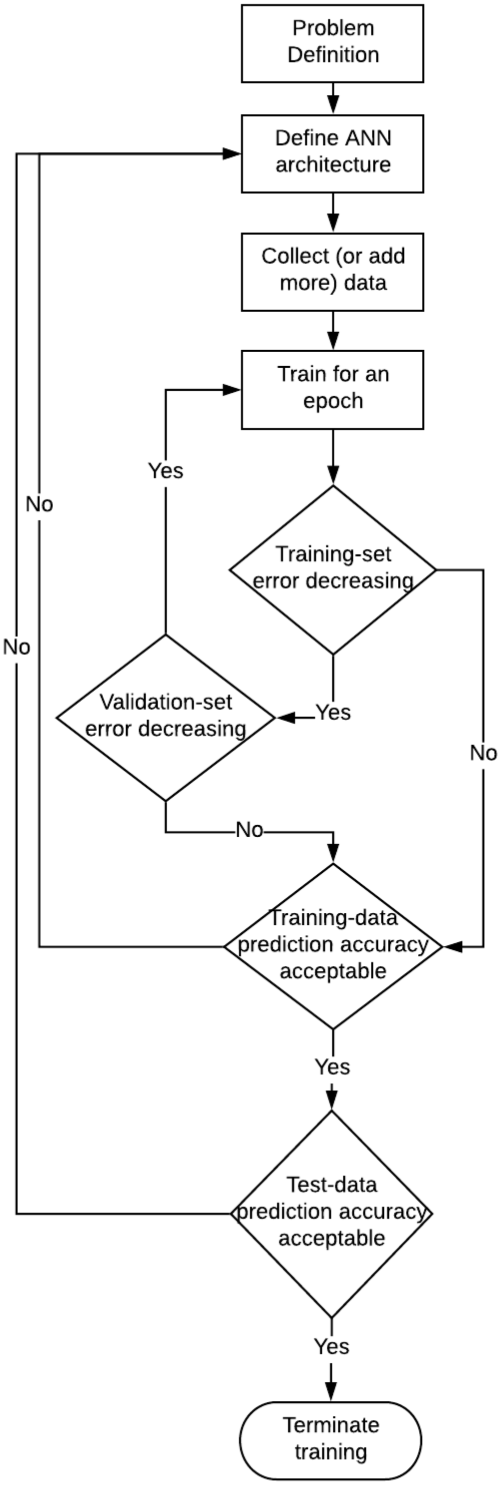

Problem Definition
Define ANN architecture
Collect (or add more) data
Train for an epoch
Yes
No
Training-set error decreasing
No
Validation-set error decreasing
Yes
No
No
Training-data prediction accuracy acceptable
Yes
Test-data prediction accuracy acceptable
Yes
Terminate training

Figure 4.3: An implementation of Artificial Neural Network Work Flow. *(Modified from Buduma [102])*

4.1.2 Support Vector Machines

Similar to ANNs, Support Vector Machines (SVMs) also originate with the perceptron; these algorithms are also able to discriminate between data within a non-separable set. However, rather than the multi-layered filtering techniques adopted by ANNs, SVMs rely on the construction of maximum margin hyperplanes to deliver conclusions on the relationships within the dataset. Vapnik offered the underlying mathematics supporting this decision-making approach in 1963; further developments over 2-3 decades have led to the current form of SVMs [112].

Discussion of the SVM approach is initiated with the classification of a linearly separable set of data into the binary classes, $Class1$ (positive) and $Class2$ (negative). As illustrated in Fig. 4.4(a), there exist multiple hyperplane options to separate such data. Given any selected separating hyperplane, the classification of a data point i identified by $\mathbb{x}_i$ is determined by the value of its projection along the direction $\mathbb{W}$ orthogonal to the hyperplane, i.e.,

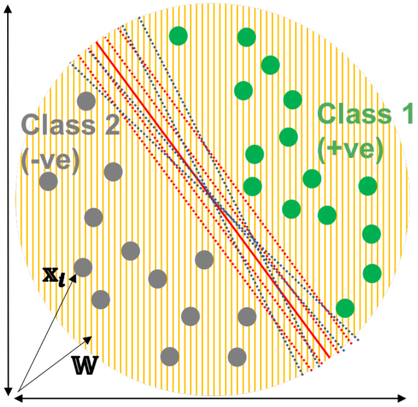


Figure 4.4: Schematic illustrating multiple separating hyperplane options.

$$\mathbb{W}^T \mathbb{x}_i > \mathtt{C} \quad \text{positive class} \tag{4.21a}$$

$$\mathbb{W}^T \mathbb{x}_i < \mathtt{C} \quad \text{negative class} \tag{4.21b}$$

or, equivalently,

$$\mathbb{W}^T \mathbb{x}_i + \mathbb{B} > 0 \quad \text{positive class} \tag{4.21c}$$

$$\mathbb{W}^T \mathbb{x}_i + \mathbb{B} < 0 \quad \text{negative class} \tag{4.21d}$$

where, $\mathtt{C} = -\mathbb{B}$. Thus, $\mathbb{W}^T \mathbb{x}_i + \mathbb{B} = 0$ on the decision boundary (at the selected hyperplane). The parameter weights $\frac{\mathbb{W}}{||\mathbb{W}||}$ are orthogonal to the hyperplane and influence its orientation, with the bias parameter $\mathbb{B}$ shifting the hyperplanes within the coordinate space. The determination of the parameters $\mathbb{W}, \mathbb{B}$ to produce the most effective separating hyperplane is pursued by construction of the notion of a *functional margin* between a data point and the selected hyperplane $\hat{\mathfrak{d}}_f^i$, and is defined as [113]:

$$\hat{\mathfrak{d}}_f^i = \mathbb{y}_i(\mathbb{W}^T \mathbb{x}_i + \mathbb{B}) \tag{4.22}$$

where, $\mathbb{y}_i$ is set equal to $+1$ and -1 for the positive and negative samples respectively. Thus, when a data sample is classified correctly, $\mathbb{y}_i(\mathbb{W}^T \mathbb{x}_i + \mathbb{B}) > 0$. The functional margin for the entire problem is defined as $\hat{\mathfrak{d}}_f = min(\hat{\mathfrak{d}}_f^i)$ and is representative of the confidence and accuracy of the classification prediction [114]. An increased confidence in the prediction thus means the maximization of $\hat{\mathfrak{d}}_f^i$. However, an arbitrary scaling of a constraint on the parameters $\mathbb{W}$ and $\mathbb{B}$ bears no impact on the modulation of the decision boundary to determine the most efficient discriminator [114], because, for example:

$$\mathbb{W}^T \mathbb{x}_i + \mathbb{B} = 0 = 2\mathbb{W}^T \mathbb{x}_i + 2\mathbb{B}. \tag{4.23}$$

Thus, the concept of a geometric margin is also introduced, one that normalizes the output of the functional margin by $||\mathbb{W}||$:

$$\hat{\mathfrak{d}}_g^i = \mathbb{y}_i\left(\frac{\mathbb{W}}{||\mathbb{W}||}^T \mathbb{x}_i + \frac{\mathbb{B}}{||\mathbb{W}||}\right) \tag{4.24}$$

The geometric margin for the problem is computed as $\hat{\mathfrak{d}}_g = min(\hat{\mathfrak{d}}_g^i)$, and this value is equal to the functional margin when $||\mathbb{W}||= 1$. A conceptual representation of such a physical margin is depicted in Fig. 4.5. Thus, appropriate values for $\mathbb{W}$ and $\mathbb{B}$ to construct the most effective decision boundary are sought by maximizing the geometric margin between the data points and the separating hyperplane. Posed as an optimization problem, this objective is denoted

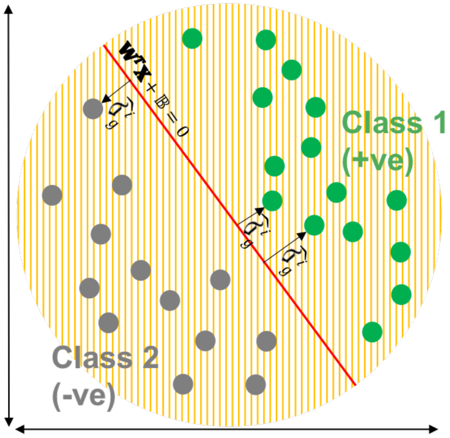


Figure 4.5: Schematic illustrating geometric margins to individual data points.

$$\underset{\boldsymbol{W},B}{maximize} \quad \frac{\hat{\mathfrak{d}}_f}{||\mathbb{W}||} \tag{4.25a}$$

$$\text{s.t.,} \quad \mathbb{y}_i(\mathbb{W}^T\mathbb{x}_i + \mathbb{B}) \geq \hat{\mathfrak{d}}_f \tag{4.25b}$$

The functional margin is set equal to unity ($\hat{\mathfrak{d}}_f = 1$) in Eqn. (4.25) for mathematical convenience, resulting in the objective function:

$$\underset{\boldsymbol{W},B}{maximize} \quad \frac{1}{||\mathbb{W}||} \tag{4.26a}$$

$$\text{s.t.,} \quad \mathbb{y}_i(\mathbb{W}^T\mathbb{x}_i + \mathbb{B}) \geq 1, \quad i = 1...\mathrm{N} \tag{4.26b}$$

where, N is the total number of data points to be classified. Note that such a modification is equivalent to scaling $\hat{\mathfrak{d}}_f$ by a constant value; thus, appropriate scaling of the parameters $\mathbb{W}$ and $\mathbb{B}$ honors the relationship defined in Eqn. (4.22). Equation (4.26a) is then again modified to present a convex quadratic formulation for the objective function, increasing computational efficiency:

$$\underset{\boldsymbol{W},B}{minimize} \quad \frac{1}{2}||\mathbb{W}||^2 \tag{4.27a}$$

$$\text{s.t.,} \;\; \mathbb{y}_i(\mathbb{W}^T\mathbb{x}_i + \mathbb{B}) \geq 1, \quad i = 1 ... \text{N} \tag{4.27b}$$

Such a formulation that seeks to maximize the geometric margin renders the SVM a *maximal margin classifier* [113, 114]. Note that the minimum value of the linear constraint specifying the magnitude of the functional margin in Eqn. (4.27b) is 1. So, any data points lying on the hyperplane $\mathbb{W}^T\mathbb{x}_i + \mathbb{B} \pm 1 = 0$ support the computation of the margin maximizing boundaries and are thus termed *support vectors*. The above discussed mathematical formulations are visualized conceptually in Fig. 4.6. In reality, data are frequently not easily separable; the formulation in Eqn. (4.27) is not robust to outliers as well. So, slack variables, ζ, are also introduced [115] to facilitate

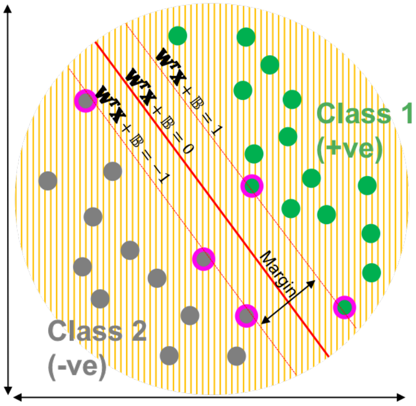

$\mathbb{W}^T\mathbb{X} + \mathbb{B} = 1$
$\mathbb{W}^T\mathbb{X} + \mathbb{B} = 0$
$\mathbb{W}^T\mathbb{X} + \mathbb{B} = -1$
Class 1
(+ve)
Margin
Class 2
(-ve)

Figure 4.6: Schematic illustrating formulation for SVM. The support vector data points are highlighted with magenta boundaries.

a user-defined tolerance for misclassification error. Equation (4.28) then assumes a soft-margin form where the objective function is appended with penalty measures:

$$\text{minimize}\frac{1}{2}||\mathbb{W}||^2+C\sum_{i=1}^{\mathrm{N}}\zeta_i{}^{\kappa} \tag{4.28a}$$

$$\text{subject to: } \mathbb{y}_i(\mathbb{W}^T\mathbb{x}_i+\mathbb{B})\geq 1-\zeta_i{}^{\kappa} \quad i=1...\mathrm{N} \tag{4.28b}$$

$$\zeta_i^{\kappa}\geq 0 \quad i=1...\mathrm{N} \tag{4.28c}$$

where, $\mathbb{y}_i = \pm 1$ indicates the true classes, and the variable C is exploited to trade-off the desired size of the discriminating margin and the flexibility allowed with misclassifications. The selection of κ determines the regularization technique to manage overfitting and establish a generalizable trained model. This investigation employs the L_1-norm regularization ($\kappa = 1$) to exploit the features with the most significant impact on the classification process [116]. The effect of choosing the L_2-norm regularization has not been considered in this investigation and remains an open avenue of research.

Solving for the Weights and Biases of the Separating Hyperplane

Solving for $\mathbb{W}$ and $\mathbb{B}$ is initiated by constructing the primal form of the Lagrangian, where the constraints in Eqn. (4.28b) and (4.28c) are appended with non-negative Lagrange multipliers $\nu_i \geq 0$ and $\mathbb{v}_i \geq 0$ and adjoined with the objective function [117]. Thus, Eqn. (4.28a) is modified as:

$$\underset{\boldsymbol{W},B,\zeta}{min}\ \mathscr{L}_p=\frac{1}{2}\mathbb{W}^T\mathbb{W}+C\sum_{i}^{\mathrm{N}}\zeta_i-\sum_{i}^{\mathrm{N}}\nu_i[\mathbb{y}_i(\mathbb{W}^T\mathbb{x}_i+\mathbb{B})-1+\zeta_i]-\sum_{i}^{\mathrm{N}}\mathbb{v}_i\zeta_i \tag{4.29}$$

Determining the stationary points of $\mathscr{L}_p$ with regards to the primal variables $\mathbb{W}$, $\mathbb{B}$ and ζ leads to the following conditions:

$$\mathbb{W} = \sum_i \nu_i \mathbb{y}_i \mathbb{x}_i \tag{4.30a}$$

$$0 = \sum_i \nu_i \mathbb{y}_i \tag{4.30b}$$

$$C - \nu_i - \mathbb{v}_i = 0 \tag{4.30c}$$

These relationships are substituted into Eqn. (4.29) to construct the dual formulation (Eqn. (4.31)), useful in scenarios requiring analysis in a higher-dimensional space

$$\underset{\nu}{max}\ \mathscr{L}_d = \sum_i^{\mathrm{N}} \nu_i - \frac{1}{2}\sum_i^{\mathrm{N}}\sum_j^{\mathrm{N}} \nu_i \nu_j \mathbb{y}_i \mathbb{y}_j \langle \mathbb{x}_i, \mathbb{x}_j \rangle \tag{4.31a}$$

$$s.t. \sum_i \nu_i \mathbb{y}_i = 0 \tag{4.31b}$$

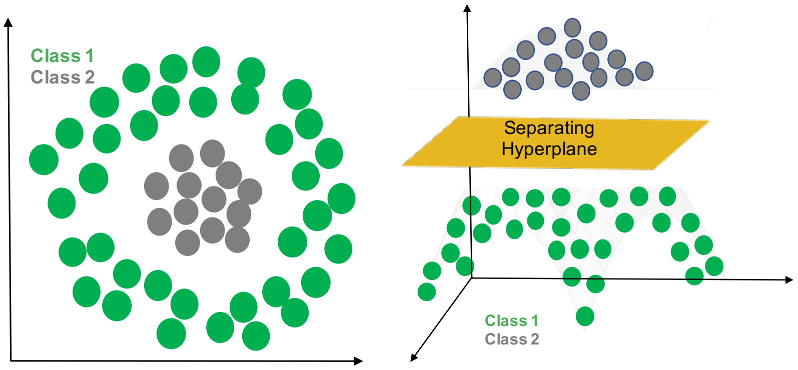
$$0 \leq \nu_i \leq C \tag{4.31c}$$

where, $\langle \mathbb{x}_i, \mathbb{x}_j \rangle$ represents the inner product between the data points. The term C is labeled a *box constraint* since it bounds the values of the Lagrange multipliers ν_i. The Karush-Kuhn-Tucker (KKT) conditions supply the necessary conditions for optimality and dictate the values for ν_i [115]:

$$\begin{cases} \mathbb{y}_i(\mathbb{W}^T\mathbb{x}_i + \mathbb{B}) \geq 1 & \nu_i = 0 \\ \mathbb{y}_i(\mathbb{W}^T\mathbb{x}_i + \mathbb{B}) \leq 1 & \nu_i = C \\ \mathbb{y}_i(\mathbb{W}^T\mathbb{x}_i + \mathbb{B}) = 1 & 0 < \nu_i < C \end{cases} \tag{4.32}$$

Since only the non-zero values of ν_i contribute towards the maximization process of the Lagrangian, the corresponding $\mathbb{x}_i$ assume the role of *support vectors* for the problem. The value of $\mathbb{B}$ is extracted from the third KKT condition in Eqn. (4.32), i.e., $\mathbb{y}_i(\mathbb{W}^T\mathbb{x}_i + \mathbb{B}) = 1$ as $\zeta_i = 0$ when $0 < \nu_i < C$.

In many real-world applications, and especially in astrodynamics, the data is often not linearly separable; i.e., the data is non-separable. This challenge is mitigated by mapping the available information to a higher-dimensional space where a separating hyperplane for the same data exists (e.g., Fig. 4.7). Note that the Lagrangian in Eqn. (4.31) is a function of the inner products of the data points. In higher-dimensional space, this inner product is achieved by applying an expensive nonlinear transformation of the data to such a higher dimension via $\mathbb{\alpha}(\mathbb{x}_i)$ followed by the inner product $\langle \mathbb{\alpha}(\mathbb{x}_i), \mathbb{\alpha}(\mathbb{x}_j) \rangle$. An alternative is a *kernel trick* that exploits a function that directly represents such an inner product in higher-dimensional feature space. A function is identified as a 'kernel' via Reproducing Kernel Hilbert Space theory (RKHS) [118] that evaluates the proximity of functions in a space that is represented by the Euclidean space generalized to infinite dimensions (Hilbert space) [119]. Such a Kernel function, $\mathfrak{K}(\mathbb{x}_i, \mathbb{x}_j)$, must also satisfy the *Mercer kernel* property [101], that is, the kernel must be symmetric positive semi-definite [114]. Thus, the dual form



(a) (b)

Figure 4.7: Schematic demonstrating transformation of (a) non-separable data into (b) separable data in a higher dimension.

of the modified Lagrangian $\mathscr{L}_d$, is also convex, eliminating potential encounters with problematic local-minima encountered with ANNs during optimization:

$$\underset{\nu}{max}\ \mathscr{L}_d = \sum_{i}^{\mathrm{N}} \nu_i - \frac{1}{2} \sum_{i}^{\mathrm{N}} \sum_{j}^{\mathrm{N}} \nu_i \nu_j \mathbb{y}_i \mathbb{y}_j \mathfrak{K}(\mathbbm{x}_i, \mathbbm{x}_j) \tag{4.33}$$

The Radial Basis Function (RBF) / Gaussian Mercer kernel that represents the inner product in an infinite dimensional feature space [120] is adopted in this investigation:

$$\mathfrak{K}(\mathbbm{x}_i, \mathbbm{x}_j) = e^{-\gamma ||\mathbbm{x}_i - \mathbbm{x}_j||^2}. \tag{4.34}$$

Since the kernel functions represent the inner product between two data points, such a function is also representative of the alignment of the associated vectors; i.e., these functions support classification by assessing the alignment/similarity between the data points [114]. The hyper-parameter γ is representative of the inverse of the variance of the Gaussian kernel function. The smaller the value of γ, the further the influence of a support vector such that, a point could be assigned to the same class as a distant support vector. Such an approach may lead to a trained model that insufficiently captures the underlying complexities of the problem (under-fitting). In contrast, a high value of γ could lead to over-fitting. Thus, this hyper-parameter is tuned to achieve the required balance between bias (under-fitting) and variance (over-fitting) for a given problem. Since the dual formulation is a direct function of the inner product of the original input training data points, the computational burden is not drastically increased even when the kernel function transforms the problem into a higher-dimensional feature space.

The computationally efficient Sequential Minimal Optimization (SMO) algorithm is adopted to solve the convex optimization problem — optimizing one pair of Lagrange multipliers at a time, selected via heuristics [121]. The appropriately tuned weights $\mathbbm{W}$ and bias $\mathbb{B}$ parameters via the optimized Lagrange multipliers deliver either a positive or negative output score ($\mathbbm{W}^T \mathbbm{x}_i + \mathbb{B} = \mathbb{S}_i$), one that aids in binary classification of the data. Confidence in the classification is obtained by transforming this score to an output's posterior probability, based on the prior belief as determined by the selected kernel function [104]. A parametric form of the sigmoid function is employed to extract the appropriate mapping of the scores to approximate the posterior probability [104]:

$$\mathfrak{P}(\mathbb{S}_i) = \frac{1}{1 + \exp(\mathscr{A}\mathbb{S}_i + \mathscr{B})} \tag{4.35}$$

The term $\mathfrak{P}(\mathbb{S}_i)$ in Eqn. (4.35) represents the predicted probabilities of the data-points belonging to the class of interest. Here, $\mathscr{A}$ and $\mathscr{B}$ are scaling and intercept values, respectively, computed by an SVM score transform function (available as a MATLAB© function) as the classification error on the predicted and true classes within the training data are iteratively minimized.

Generalizability

As discussed with ANNs, the generalizability of a model enables it to render accurate predictions on unseen data. The balance between training and testing accuracy is achieved by regularization measures, with the appropriate tuning of values for C and other hyper-parameters introduced during the mapping to a higher-dimensional feature-space. The geometric-complexity of the hyperplane separation margin delivered by the hyper-parameters (box constraint C and kernel parameter $\mathfrak{v}$), determine the total number of support vectors and, thus, also the memory requirements associated with the trained model. Higher values of C and $\mathfrak{v}$ encourage lower misclassifications during training by constructing stricter separation boundaries, but such over-fitting may lead to a poorly generalizable model for class prediction on unseen data. These hyper-parameters are therefore tuned to achieve a bias-variance balance either via a grid-search or optimization techniques that minimize the classification errors on training and testing data sets. In this investigation, the readily available *BayesOpt* library [122] in MATAB© is incorporated for their computation. The accuracy of a trained model may be tested using either hold-out or cross-validation sets [104]. The hold-out technique sets aside a small fraction of the data (commonly $25\% - 30\%$) for testing purposes which is not used during training. In the cross-validation approach, the data is equally partitioned into $\Bbbk$ *folds* such that each *fold* is independently exploited as a validation set for $\Bbbk$ rounds of training. The implementation of the SVM approach to aid in CR3BP orbit family classification goals is explored in Section 5.2.4.

4.2 Pathfinding Algorithms

In this investigation, automated pathfinding efforts are implemented to sequence a desirable initial guess for an end-to-end route from departure to destination in the CR3BP regime. The automated search is implemented via both (i) exact and (ii) Reinforcement Learning (RL) algorithms. In the former approach, a pre-specified rule specifies a path that always follows the minimum-cost route to construct a globally optimal trajectory. In contrast, in the second method (RL), software *agents* 'learn' beneficial routes by random exploration of the design space followed by exploitation of the knowledge gained over multiple iterations. In this analysis, an 'agent' is an abstraction for the software representation of a decision-making process in lieu of a human operator. The choice between the two approaches is usually determined by the scope of the problem, computational capabilities with regard to time (time-complexity) as well as storage (space-complexity), and the acceptable sub-optimality of the solution. In this analysis, both approaches are examined to better understand their applicability to trajectory design scenarios. An introduction of Dijkstra's algorithm (exact approach) is followed by a discussion of reinforcement learning via a heuristic approach.

4.2.1 Dijkstra's Algorithm

Since the inception of graph theory by Leonhard Euler in the 18^{th} century [123], the specific methods and variants of this branch of mathematics have demonstrated great utility in varied disciplines such as routing, genomics, and artificial intelligence. In each application, the graph represents the set of total available nodes/options (N) under consideration [124]. The edges E or *arcs*, linking pairs of individual nodes (n) are shaped by the costs between them, $Cost[< n_i, n_j >]$. Non-unitary costs establish a weighted graph that is either *directed* if the edge-costs are oriented or *undirected* otherwise. Commencing at the *departure* node, the graph is expanded one arc at a time, $< n_i, n_j >$, until the agent arrives at the *target* node. This traversal is accomplished by: a) establishing a *frontier of neighbors* that the agent can meet in one hop when the edge costs from the current node are finite; and b) selecting a neighbor from this frontier to incrementally expand the search towards the target. Various motivating factors such as fewest nodes to the target or a preference in terms of time-space complexity influence the choices in the search algorithm, dictating the selection of the next node in expanding the frontier. The path, $\mathcal{P}$, to the target is constructed via a sequence of arcs with a total path cost, $Cost(\mathcal{P}) = \sum_{i=1}^{K} Cost[< n_i, n_{i+1} >]$, where K is the cardinality of the set of arcs in the sequence.

The type of graphs encountered within the context of mission design are usually weighted and directed due to the complex dynamics and the mass-time trade-offs that constitute the edge-costs. So, Dijkstra's algorithm is employed as the *exact* approach in this investigation when the goal is multi-objective optimization; Dijkstra's formulation constructs the least-cost (shortest) pathway to the destination. This algorithm employs a form of greedy search that always selects the lowest cost neighbor along a frontier; the cost from the departure node is updated at a later time if an alternative, lower-cost route is later discovered. One approach in implementing Dijkstra's algorithm is a priority queue that constantly re-stacks the nodes in ascending order of cost as summarized in Figure 4.8. Such a search strategy that incorporates the least-cost expansion of each arc, guarantees the emergence of the shortest path to the target, i.e., the global optimum [125]. Non-unique edge costs from a node can lead to non-unique global optima in certain scenarios. The algorithm also guarantees that, if a solution exists, it will be uncovered in a finite time interval. However, this finite

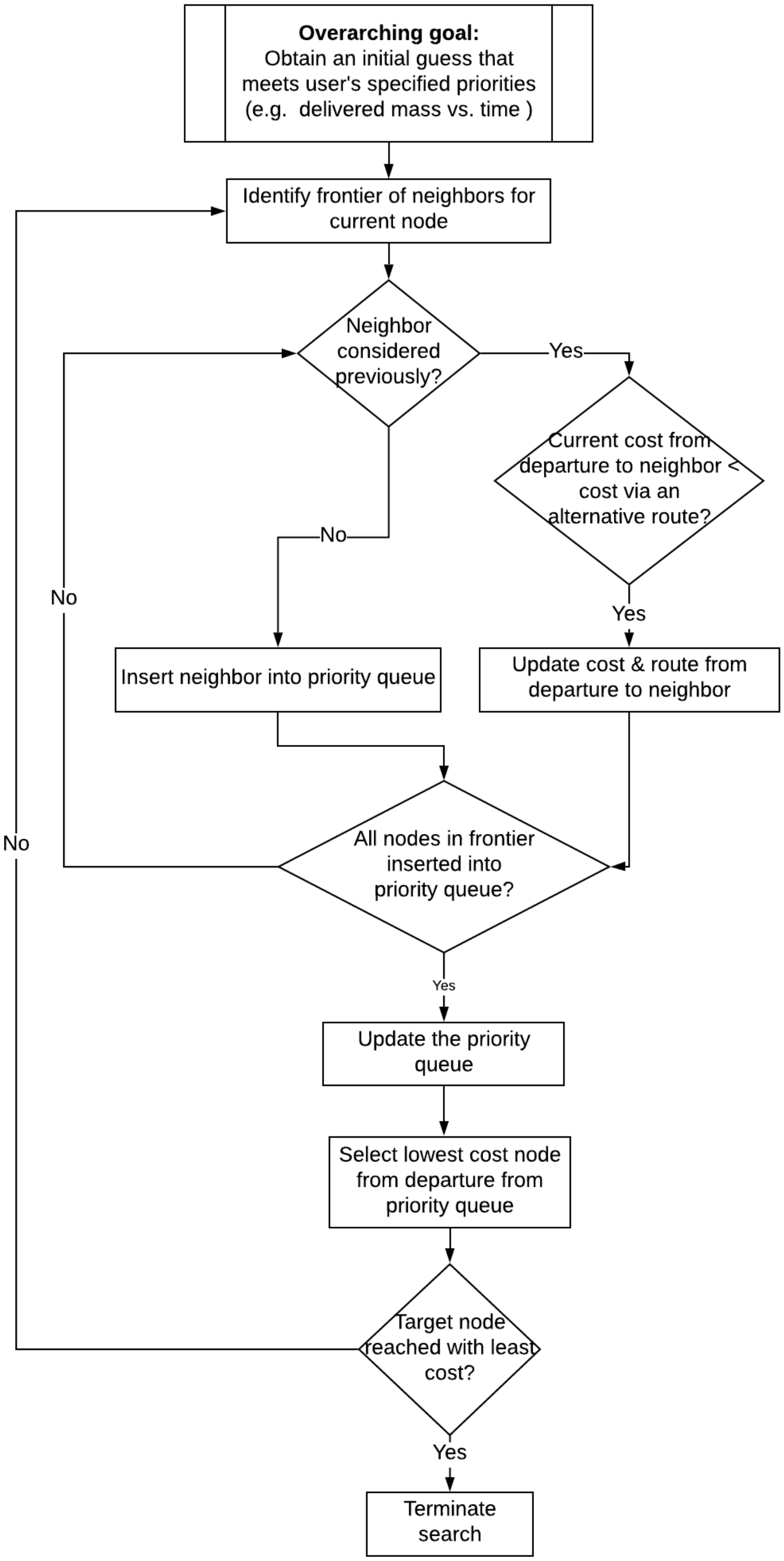

Overarching goal:
Obtain an initial guess that meets user's specified priorities (e.g. delivered mass vs. time)
Identify frontier of neighbors for current node
Neighbor considered previously?
Yes
Current cost from departure to neighbor < cost via an alternative route?
No
Insert neighbor into priority queue
Yes
Update cost & route from departure to neighbor
No
All nodes in frontier inserted into priority queue?
No
Yes
Update the priority queue
Select lowest cost node from departure from priority queue
Target node reached with least cost?
Yes
Terminate search

Figure 4.8: An implementation of Dijkstra's algorithm

computation time increases with an increasing search space. The time is bounded by polynomial growth of order $O(n^2)$ for naive implementations, but can be improved with data structure implementations such as priority queues and *heaps* [57]. Note that heap implementation can become increasingly challenging for a complex search space. A significant factor leading to increased computational time is the uninformed nature of the search, i.e., Dijkstra's algorithm is not aware of the target/user's goals as the search evolves. Built on similar principles, the $\texttt{A}^*$ algorithm introduces heuristics to conduct an informed search towards the target; such an adjustment typically improves the time-complexity, depending on the implementation [124]. However, $\texttt{A}^*$ has not been explored in this investigation, as strict rules on the admissibility of the heuristics are challenging to satisfy in a complex dynamic environment with multi-objective optimization goals, especially with simultaneous considerations of mass-time trades [59]. A more flexible heuristic approach is executed via reinforcement learning.

4.2.2 Heuristically Accelerated Reinforcement Learning (HARL)

Any endeavor in the space environment is complex. A flexible and sustainable infrastructure will benefit from a mission design approach that delivers transfer solutions while accommodating uncertainties in the environment, variable performance measures, and shifting mission goals. Reinforcement Learning (RL) algorithms originating in the artificial intelligence domain are an effective tool in balancing sometimes conflicting goals. Modeled as an agent interacting with its environment, an agent successively learns to deliver appropriate choices that lead to desirable outcomes. The reinforcement learning approach is commonly formalized as a Markov Decision Process (MDP) [124] that is constructed from a tuple: $< S, A, P, R >$; S is a set of states available to an agent; A is a set of actions available to an agent; P is $P(s,a,s')$, the probability that action a in state s leads the agent to arrive at state s'; and, $R : S \times A \rightarrow \mathscr{R}$ is the reward received for an action a in state s. The aim is an optimal policy (π^*) that executes an action at a given state to maximize the cumulative rewards received by the agent over multiple episodes. A thorough treatment of various Reinforcement Learning (RL) algorithms is presented in Sutton and Barto [126] — the common aim of all the algorithms is the formulation of an agent's policy/strategy at a given state to meet the objectives in a given problem. A gradual transition towards the optimal policy may be achieved via *policy iteration* or *value iteration* methods. In the former approach, the policy (function that maps a state to an action) is explicitly and directly modified over multiple episodes. In contrast, during 'value-iteration' the update of the current policy towards the target policy is implicitly handled via iterative and direct updates of the value of a state or state-action pair. An actor-critic method presents a hybrid approach that exploits the strengths of both the value-based and policy-based techniques [126]. A brief introduction to the value iteration method adopted in this investigation is detailed in the following discussion; the specific implementation details for application in the CR3BP pathfinding problem are examined in Section 5.3.2.

A strategy to determine the optimal policy (π^*) is initiated by assessing the state-value $\mathbb{V}^\pi(s)$, that is, a measure of the expected returns for taking actions under the policy π from a state s at time-step t. The value function is mathematically formulated as:

$$\mathbb{V}^\pi(s) = \mathrm{E}_\pi \left\{ \sum_{\mathrm{k}=0}^{N} \gamma^{\mathrm{k}} R_{t+k} | s_t = s \right\} \tag{4.36}$$

The discount factor $0 \leq \gamma \ \leq 1$ is a measure of the balance between rewards gained from the immediate and future time-steps; smaller values of γ favor immediate rewards. A discounted-reward approach is especially important in infinite horizon RL scenarios. That is, for scenarios with no specified end-time ($N = \infty$), setting $\gamma < 1$ aids in maintaining a finite value for the accumulated rewards. For a finite horizon problem, the state-value is computed by delaying the assignment of cumulative discounted rewards along the entire path until the end of an episode (Monte-Carlo learning) or by looking ahead a specified number of steps (temporal difference learning, $TD(\lambda)$) [126]. Here, $\lambda \in (0, 1)$, such that, $TD(0)$ looks ahead 1-step after invoking an action a from a state s, and $TD(1)$ is comparable to Monte-Carlo learning. An iteration over all the possible actions from an originating state evolves the system to eventually arrive at the optimal choice ($\mathbb{V}^*$). As an example, the update for $\mathbb{V}^*$ in terms of a simple 1-step look-ahead is detailed in Eqn. (4.37):

$$\mathbb{V}^*(s) \leftarrow \underset{a}{max} \sum_{s'} P(s,a,s')[R(s,a,s') + \gamma \mathbb{V}^*(s')] \tag{4.37}$$

Equation (4.37) is a form of the *Bellman Equation* that is employed widely in dynamic programming and satisfies the necessary conditions for optimality. The prime symbol ($'$) indicates a state(s) accessible from the *frontier*, as introduced in Section 4.2.1. In complex regimes where the transition probabilities defining the system model and the rewards are not known a priori, it is useful to construct the problem as a model-free, state-action-value approach. A state-action-value update function is formulated in recursive form as:

$$\mathbb{F}(s,a) \leftarrow \mathbb{F}(s,a) + \alpha[R(s,a,s') + \gamma \mathrm{U} - \mathbb{F}(s,a)] \tag{4.38}$$

and forms the repository of reinforcements from which the agent learns the desirable behavior. The agent takes an action a in state s and is then assumed to follow a certain policy from state s' onwards; an opportunity for the agent to experience the consequences of varied actions a from state s over time is, thus, conceived. Equation. (4.38) is formulated as an *on-policy* update if an agent takes an action a' from state s' based on experiences sampled from the current control policy π (i.e., $\mathrm{U} = \mathbb{F}(s',a')$); an *off-policy* update is when the agent follows a different policy in state s' (e.g., $\mathrm{U} = \underset{a'}{max}\mathbb{F}(s',a')$) [126]. The learning rate of the agent is specified by α, which balances the information acquired from recently rewarded pathways versus a reliance on knowledge about pathways from prior episodes.

The convergence in response to an optimal policy using the RL approach is guaranteed by re-visiting state-action pairs infinitely many times, and the process may be prolonged in scenarios subject to a large state-space. Thus, the time-complexity of some RL algorithms can be as challenging as exact approaches, e.g., Dijkstra's algorithm. Therefore, it is beneficial to introduce a heuristic function that accelerates the learning process by biasing the selection of an action a, given a state s, possessing a reward R. Such an algorithm is termed a Heuristically Accelerated Reinforcement Learning (HARL) algorithm [127], and the time-complexity depends on the accuracy of the heuristics. The ability of the heuristic function to influence the action choice is based on the ability for variations in the heuristic to exceed the variations in the reinforcement $\mathbb{F}(s, a)$ [128]. Scenarios with large state-spaces and multiple objectives benefit from a coordinated effort in a multi-agent distributed HARL approach, particularly one that exploits parallel processing. In such a distributed network, the agents work cooperatively by updating a centralized reinforcement repository based on the distributed knowledge (cumulatively discounted rewards) gained during a specific episode. Thus, parallel computing capability with RL offers a time-advantage over Dijkstra's algorithm for problems with a broad trade-space, as Dijkstra's algorithm is not parallelizable. Within the context of mission design, a RL pathfinding agent continues the search until a stopping condition is satisfied, i.e., a terminal condition and/or a violation of constraints. An overview of the learning process (within the context of mission design) is illustrated in Figure 4.9.

The learning process within each episode is comprised of two search scenarios — exploration and exploitation. Exploration enables a training phase where the agent *learns* about likely consequences of actions in the environment; the exploitation phase enables the agent to engage in informed decisions by capitalizing on previously gained knowledge. The policy at a particular state as influenced by the state-action pair and heuristic is [128]:

$$\pi(s) = \begin{cases} \mathbb{E}[\mathbb{F}(s, a) \bowtie \mathbb{\xi}\mathbb{H}(s, a)^{\mathbb{\beta}}], & \text{if } \mathbb{q} > p \\ a_{random}, & \text{otherwise} \end{cases} \tag{4.39}$$

where ($\mathbb{H} : S \times A \to \mathscr{R}$) is the heuristic function, $\bowtie$ is a math operator as determined by the RL algorithm and its implementation, $\mathbb{\xi}$ and $\mathbb{\beta}$ are weighting parameters that

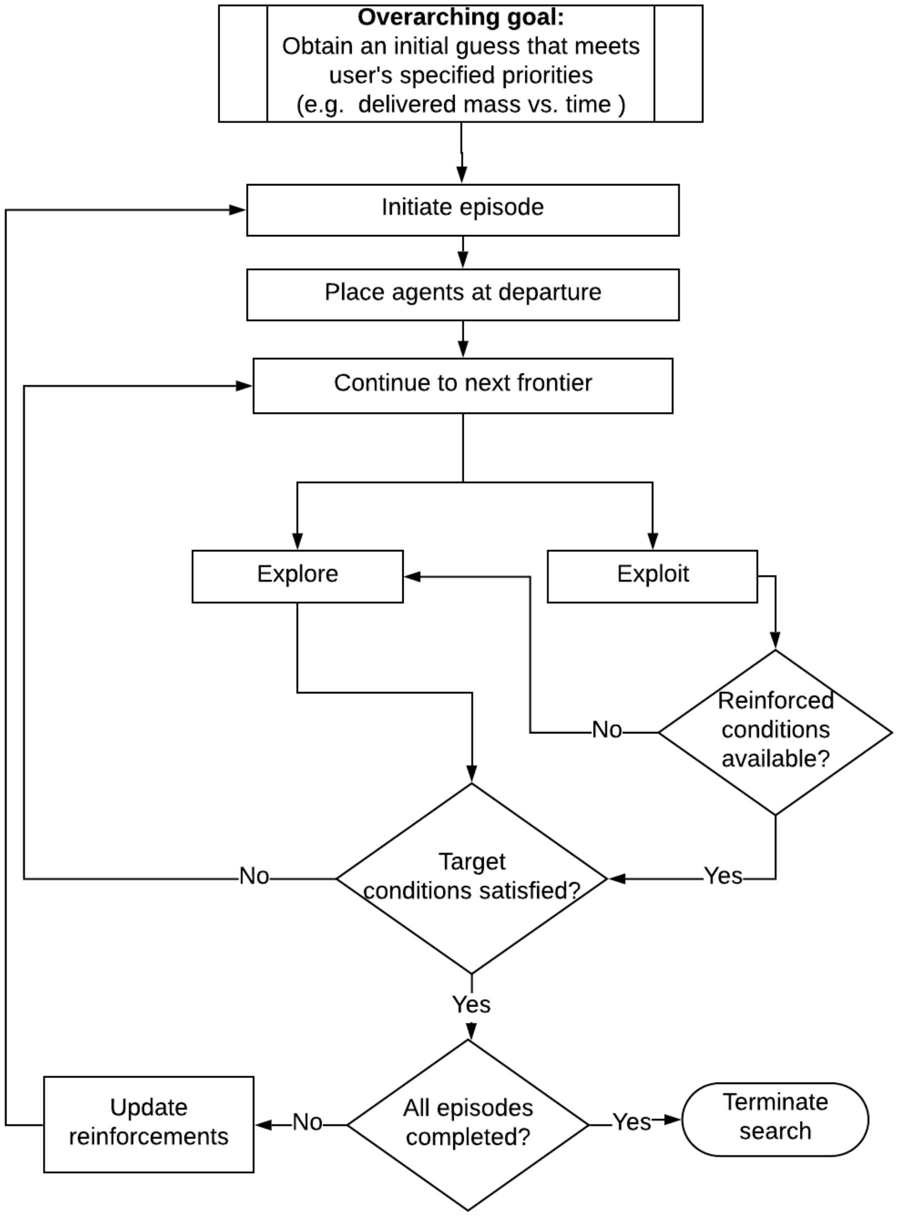

Overarching goal:
Obtain an initial guess that meets user's specified priorities (e.g. delivered mass vs. time)
Initiate episode
Place agents at departure
Continue to next frontier
Explore
Exploit
Reinforced conditions available?
No
Yes
Target conditions satisfied?
No
Yes
All episodes completed?
No
Yes
Update reinforcements
Terminate search

Figure 4.9: An implementation of HARL algorithm

dictate the influence of the heuristic, and a_{random} is the action selected randomly from all those available in state s. The construction of the exploitation function $\mathbb{E}$ is specific to a particular application; some variants include a greedy strategy that incorporates $max(\mathbb{E})$, a minimax approach that minimizes the losses in a maximum loss scenario, and a 'softmax' or other stochastic function to select an action [129]. In this equation, p, $(0 \leq p \leq 1)$, is the trade-off parameter between exploration and exploitation, and $\mathbb{q}$ is a random value from a uniform distribution in [0,1]. Greater values of p encourage exploration and lower values bias the process toward exploitation. Extensive exploration in earlier episodes is beneficial, followed by a gradual shift to exploitation to capitalize on knowledge gained about the environment. A convenient function to control the steady-state value of the trade-off probability, p_{ss}, by the k^{th} episode from a total number of episodes Ep, is constructed as [51]:

$$p = p_{ss} + (1 - p_{ss})e^{(-\frac{k-1}{ln(Ep)})} \tag{4.40}$$

So, although a value of $\mathbb{q}$ chosen from the uniform distribution implies equal probability for an agent to explore or exploit conditions within an AR, a logarithmically decaying value of p biases the proportion of exploration vs. exploitation from one episode to another. Additionally, as illustrated in Fig. 4.9, an agent is directed to revert to exploration if no reinforced conditions are available when the value of $\mathbb{q}$ exceeds that of p — such a measure mitigates the challenges associated with trajectory design within a large state-space. Also, an agent's search is terminated if any user-defined constraints such as minimum altitude from a primary are violated by a particular state's selection and no alternative states are available to continue the AR construction process. The blending of heuristics with reinforcement learning is a powerful utility in solving many NP-hard and NP-complete pathfinding problems, e.g., the Traveling Salesman Problem (TSP).

5. DEVELOPMENT OF THE DESIGN FRAMEWORK

The current investigation strives to establish a framework for automated trajectory initial guess generation within a multi-body regime via artificial intelligence techniques. The four essential components illustrated in Fig. 5.1 (top row) serve as the foundation for the design framework. These components are summarized as: (I) Accessible regions — determines the *reach* of the s/c within a finite time-horizon in a multi-dimensional and infinitely large configuration space; (II) Natural Conditions — renders natural conditions for the s/c to evaluate within these ARs; (III) Automated pathfinding — exploits artificial intelligence and machine learning techniques to determine the transport sequence for an efficient end-to-end path in an automated process; and, (IV) Convergence/optimization — where the selected transport sequence is converged and optimized to construct a continuous solution, given a specified engine model, by more traditional numerical strategies. Here, the first three components of the AI-aided framework work towards uncovering a variety of initial guesses for transfer pathways for a particular scenario. Although high accuracy is not sought for the initial guess generation phase, the fourth framework component eliminates any discontinuities to present an accurate end-to-end continuous solution for the given dynamical model assumptions.

The ARs serve to customize the route-finding effort for a spacecraft given the engine specifications. The second component incorporates available states for implementing an informed search, influencing the transfer profile, and minimizing propellant consumption where possible. These natural conditions are made available as illustrated in Fig. 5.1 (bottom row) either via (a)randomly generated states within a given AR, (b) *a priori* discretized natural periodic orbit families, or (c) randomly generated states that are selectable only if recognized by a supervised learning algorithm such as Artificial Neural Networks (ANNs) or Support Vector Machines (SVMs) as

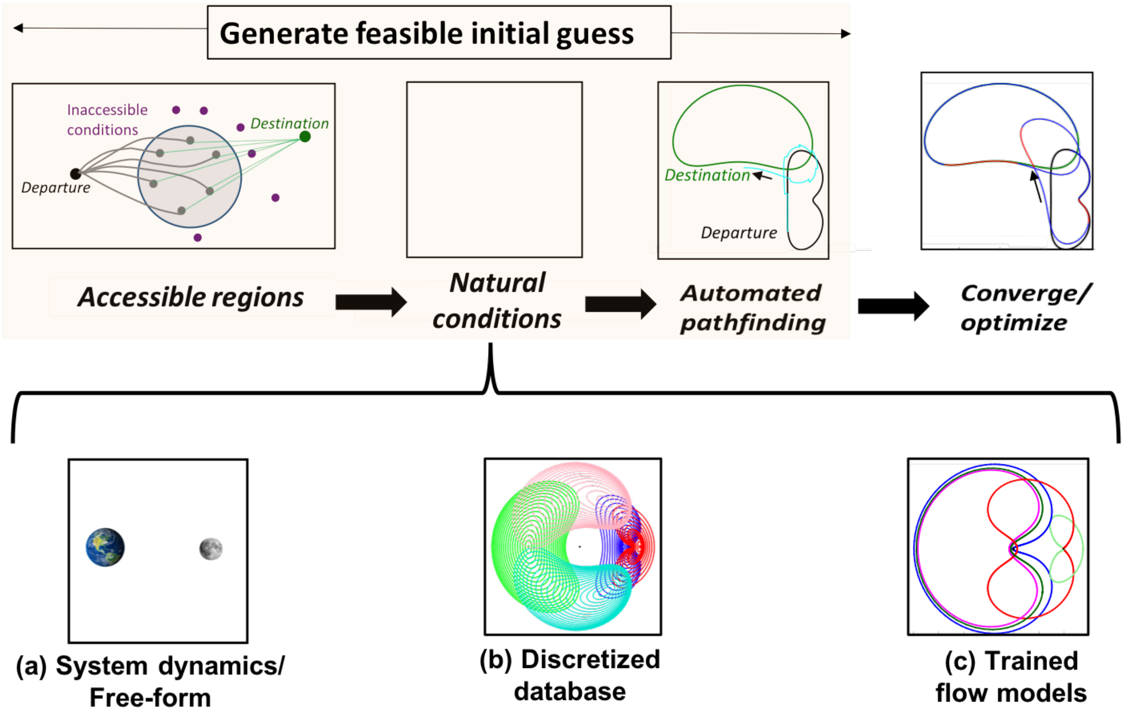


Figure 5.1: (top row) Overview of design framework; (bottom row) The three alternatives investigated for the natural condition generation component in the framework (i.e., for the 2^{nd} component in the top row represented by the empty box placeholder).

members of a particular periodic orbit family. The ANNs and SVMs *learn* to capture the *general characteristics* of the flow associated with notable periodic orbits families, rather than satisfy any set of exact conditions. The pathfinding efforts sequence advantageous conditions from within each AR to compute a path that satisfies a set of global objectives. The type of pathfinding algorithm implemented is determined by the available computational resources as well as the type of solution that is sought for the relevant design scenario, including the associated level of fidelity and delivered level of optimality. Finally, the transition of initial guesses (IG) from lower-fidelity dynamical models such as the CR3BP to higher-fidelity regimes via well-studied numerical techniques enhances rapid design and is effective. Further details associated with each of the four components and the specific implementations of the machine learning theory explored in Chapter 4 are examined.

5.1 Computation of Accessible Regions

An accessible region establishes the *reach* of a s/c in the design space, where its characteristics are influenced by the spacecraft thrust-to-mass ratio, propellant efficiency, and other performance characteristics. The computation of an Accessible Region (AR) from a particular node 'n', originates with a perturbation of the spacecraft current velocity within a circle/sphere (for the planar/spatial problem, respectively) with a prescribed radius. This step is followed by a propagation of the perturbed and unperturbed states for a pre-determined duration. The resulting downstream behavior is a stretching of the perturbed states from this unperturbed natural arc due to the influence of the existing gravitational forces in the system. To simplify implementation, the deviations of the perturbed states from the end of the natural arc are mapped to a circular/spherical Accessible Region (AR) in a planar/spatial setting, respectively, rather than using the true deformations. Figure 5.2 illustrates the resulting close approximation of the true ARs via such a design choice. The potential for greater distortions of the true ARs during operations in the vicinity of highly nonlinear regions is managed by constraining their allowable footprint through smaller velocity perturbations or shorter propagation durations. The impact of these dual control parameters are explored in more detail in the ensuing discussions. For both the chemical and low-thrust engine analyses, the perturbations in the s/c velocity at node 'n', (δV_n), are induced via chemical impulses; the magnitude is determined via the relationship

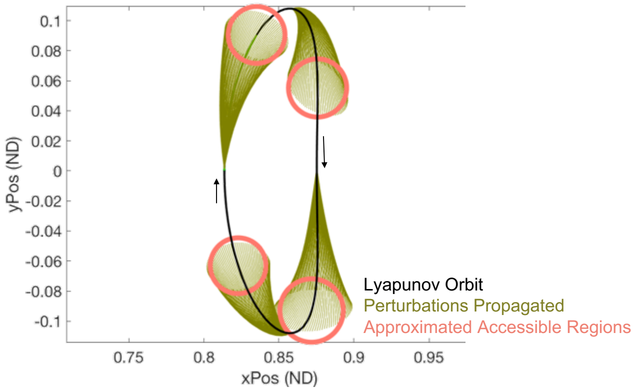
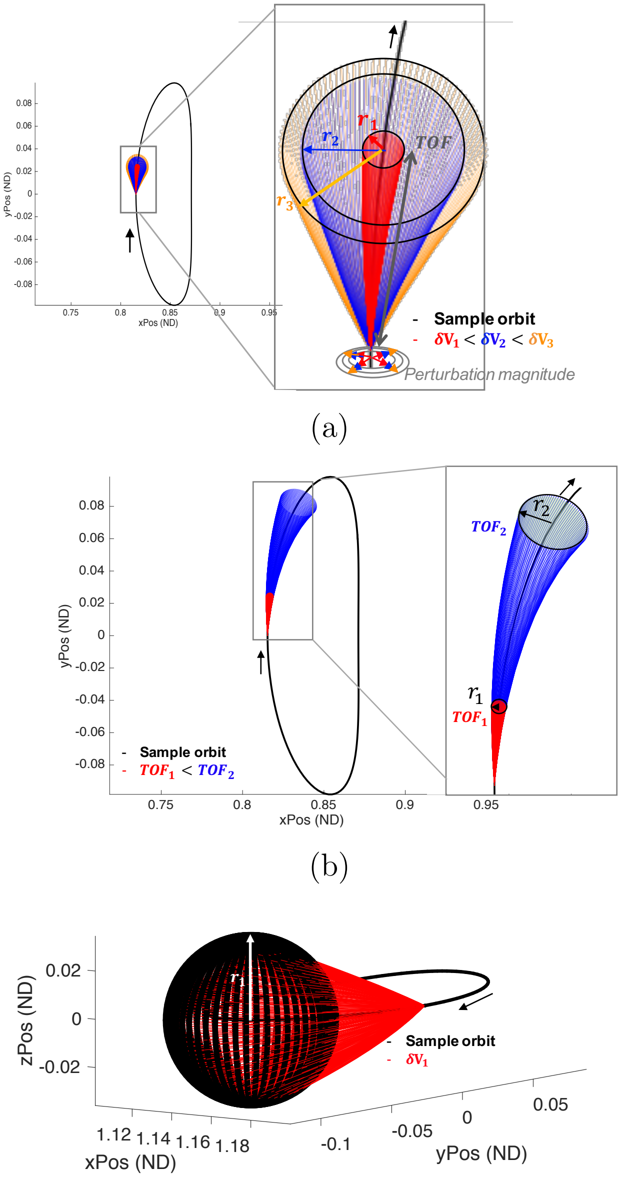


Figure 5.2: An example illustrating the actual ARs (regions enclosed by the end of the propagated arcs) and approximated ARs (pink circles) associated with different conditions along an L_1 Lyapunov orbit.

$$\delta V_n = Isp\ g_0\ \ln(\frac{m_0}{m_f}) \tag{5.1a}$$

$$\text{where,}\ \ m_f = m_0 - (\dot{m} \times \delta\mathbb{t}_n), \tag{5.1b}$$

once the engine characteristics (e.g., specific impulse, mass flow rate) and the user-defined engine operation time ($\delta\mathbb{t}_n$) are determined. Note that a chemical engine's finite burn duration is typically much shorter than the long-duration continuous burns executed by low-thrust systems. Figure 5.3(a) illustrates the difference in the footprint sizes for varying perturbation magnitudes for a fixed propagation time; a larger AR enables access to more natural arcs. Inspection of Eqn. (2.47) and Eqn. (5.1) indicates that higher specific impulse values result in smaller δV_n perturbations at a given node 'n' for a fixed operation / burn time ($\delta\mathbb{t}_n$), aiding in delineating the behavior of propellant efficient low-thrust and less efficient chemical regimes. Thus, for a fixed operation time, the footprint of the AR for a chemical engine is larger than that of a low-thrust engine. Longer propagation durations increase the footprint of the AR



(a)

(b)

(c)

Figure 5.3: Influence of δV on the accessible region footprints in the (a) planar case (c) spatial case. (b) Influence of TOF on the footprint of accessible regions in the planar case.

(Fig. 5.3(b)), and an AR in the spatial regime is illustrated in Fig. 5.3(c). Note that the ARs in the search algorithm accommodate both the position and velocity states.

5.2 Population of Conditions within Accessible Regions

Mission design is the process of harnessing the appropriate combination of orbit family geometries, velocities, and energy levels to satisfy architecture constraints such as altitude relative to a primary body or line-of-sight for communications, as well as balancing such constraints with mission objectives. Solving the trajectory design problem and delivering an acceptable path through space is inherently a combinatorial optimization problem that assesses the costs and benefits of incorporating blends of natural flows and s/c operation routines to deliver different possible outcomes. *A priori* knowledge of the natural flows in a system is, thus, greatly beneficial in informing trajectory design. This knowledge is, however, incomplete for an infinitely large state-space and so may prove limiting, especially during unexpected and time-constrained scenarios. Thus, to incorporate both known and unknown natural motion in a broad exploration of the trade-space is attractive. This goal is satisfied by allowing a s/c to query its current AR for known natural motion made available by a *database* (discretized conditions/family models) or allowing a *free-form* search that indiscriminately selects both ordered and chaotic motion during the search.

5.2.1 Free-Form — Instantaneous Generation of Initial Conditions

Accessible regions aid in constraining the searchable design space at any given instant in time. However, when *a priori* knowledge of the dynamics in the system is lacking, natural conditions can be instantaneously generated within an AR, in real-time [62]. Operating under the assumptions of a complex and highly nonlinear dynamical regime, however, these conditions are a blend of both ordered and chaotic motion. Ordered motion is distinguished from chaotic by predicting the behavior over all time within the CR3BP. However, it is beneficial to exploit any available and mission-enabling conditions that do not violate mission constraints. Thus, chaotic conditions may offer suitable candidates if the resulting natural motion is predictable over some specified and acceptable time-frame. Once states are populated within the bounds specified by an AR, a receding horizon technique tests for predictability as illustrated in Fig. 5.4. Long-term predictability is observed by propagating a condition from within an AR for some lengthy duration; this look-ahead time-frame is a design parameter and varies based on the dynamical system and/or mission considerations. If the propagated state does not escape the system as determined by a user-specified threshold for boundedness δ, then it is set aside for selection by the s/c even it is chaotic in nature. Note from Fig. 5.4 that only a small fraction along this propagation time is captured for constructing an AR. This unrestricted approach to natural state selection offers the potential to uncover non-intuitive transfer geometries otherwise not available from known and ordered natural families. Thus, the ability to select chaotic states broadens the design options and eliminates the manual time investment required to identify beneficial families to include in the searchable volume. The disadvantages are the diminished ability to specify the nature of various pathways traversed by the spacecraft, as well as the requirement to perform on-line propagations of a candidate state to confirm that it does not escape the system, possibly slowing the pathfinding process. Thus, discretized, *a priori* known natural conditions and flow-models are also useful to mitigate such challenges.

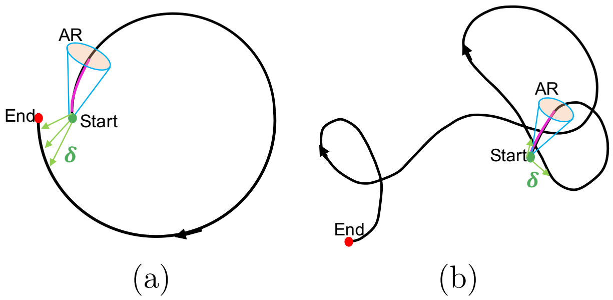


Figure 5.4: Two sample scenarios (a) and (b) illustrating look-ahead trajectory segments from *Start* to *End*, the boundedness value, δ, and the relatively small fraction along the look-ahead segment to establish an Accessible Region (AR).

5.2.2 Database — A General Discussion

Incorporation of *a priori* knowledge into the search process enables greater influence over the transfer profile. However, this approach benefits from human intuition and the introduction of appropriate combinations of ordered motion/families that do not lead to time and/or propellant-prohibitive transfer solutions. Therefore, it is important to incorporate various considerations that influence the connections between orbits belonging to different families. For example, the ease of connection is a function of the location, expanse, stability and energy levels associated with each of the families, and the extent of the s/c *reach* at a given instant in time. So, the greater the similarity in position and velocity between two orbit families, the more convenient is the transport between their members. For example, there exists a visible overlap in position between the L_1 and L_3 Lyapunov families in Fig. 5.5. However, the flow of motion between members of each family is in opposing directions within this overlapping region, and the velocity magnitudes could also be vastly different. So, an unacceptable ΔV may be required for a s/c to establish a sufficiently large *reach* to alter its course from one of these families to arrive at conditions along the other. Such an option may render the transfer expensive, impractical, or infeasible. In contrast, as illustrated in Fig. 5.6, the velocity directions are more aligned within

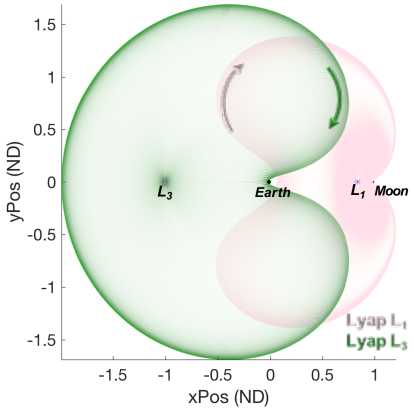


Figure 5.5: Position overlap between the L_1 Lyapunov and L_3 Lyapunov families. Arrows indicate direction of flow for orbits within each family.

the large position overlap regions between the L_3 Lyapunov and $4:3$ Resonant families. Hence, relatively smaller maneuvers may allow transport between these family members at comparable energy levels in contrast to that required for the former case discussed in Fig. 5.5.

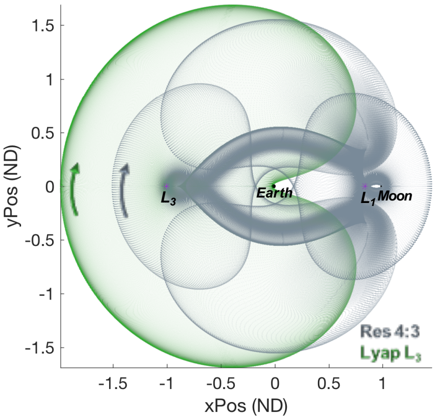


Figure 5.6: Position overlap between the L_3 Lyapunov and Resonant 4:3 families. Arrows indicate direction of flow within each family.

A critical observation for transport is the geometric semblance between families that possess similarities in both position and velocity. Hence, for this preliminary investigation, family geometry is exploited as a visual cue to assess the potential connectivity between orbit families. This qualitative assessment is then exploited to selectively include families within a searchable database to aid in the transfer construction, influence the geometry of the transfer itself, and reduce computational effort. For example, Fig. 5.7 illustrates that geometrically, there are certain regions in the Resonant $4:3$ family that resemble motion in the L_1 Lyapunov family, and other areas that resemble motion in the L_3 Lyapunov family. Hence, it is advantageous to include the Resonant $4:3$ family in the searchable database when designing transfers between the L_1 and L_3 Lyapunov families — the resonant family offers intermediate arcs with suitable flows to transport the s/c between the departure and destination orbit families. Including intermediate families is analogous to building a multi-dimensional bridge that links two regions that are otherwise inaccessible [42,61].

An alternative option is the inclusion of as many orbit families as practical for the computational efficiency of the platform; such an option precludes any requirement for human intuition during the design process. A more rigorous quantitative analysis to identify advantageous intermediate families to include within a searchable database remains desirable, especially when the user is unfamiliar with a particular Three Body (3B) system. Prior knowledge of the dynamical flows in a system are explored via two avenues in this investigation: (i) A priori discretization of families, and (ii) Incorporating *models* of these families via supervised learning techniques. The processes associated with these approaches are explored in the following section.

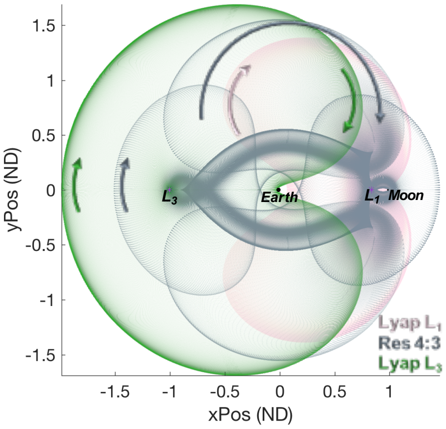


Figure 5.7: Illustrating similarities and differences in geometry between the L_3 Lyapunov, Resonant 4:3, and L_1 Lyapunov families. Arrows indicate direction of flow within each family.

5.2.3 Database — A Priori Discretization of Families

The CR3BP offers a multitude of natural solutions — periodic and quasi-periodic orbits in the vicinity of the primaries and the libration points, as well as manifold behaviors that reflect the flow throughout the region. The signatures of these natural structures are captured either by lower-dimensional Poincaré maps at selected surfaces of section or by discretization of the natural flows to facilitate *arc* extraction from the discrete initial conditions. This investigation employs the latter option and identifies the available discretized states within each AR. The discretization of the orbit families to develop a suitable database involves careful consideration of the inter-orbit and intra-orbit spacing; i.e., the spacing *between* the orbits and the states that are sampled *along* the orbits/trajectories, respectively. In this investigation, the orbits within a family are computed every 100 km apart, and the intra-orbit discretization involves computing the states along every 10% of a particular orbit's arc length. These choices have served adequately for demonstrating the functionality and feasibility of the framework in this preliminary investigation; as a design choice, the discretization scheme is free to be modified as required to support the application of interest. Depending on the span of the family, this discretization choice can require the computation of ≈ 7 million conditions on average per family. The discretization, therefore, directly impacts the number of intermediate conditions available to the s/c along a *frontier* and, thus, also impacts the end-to-end guess for a transfer. A finer grid offers a more accessible search domain, but the memory and storage limitations associated with the available computational resources also limit the size of the grid to represent each orbit family. The optimal orbit family discretization scheme could be problem-dependent and would require further investigation to draw broad conclusions. Such a topic is beyond the scope of this investigation; the current aim is a framework to construct a feasible initial guess that is straightforwardly transitioned to a continuous solution. The process by which a discretized database is exploited within an AR is illustrated in Fig. 5.8. Each consecutive AR is constructed by selecting a state from within the discretized database and propagating it forward in time after applying a velocity perturbation. An alternative option to using discrete states and computing the optimal discretization scheme is the development of *models* that capture the general characteristics of a given family.

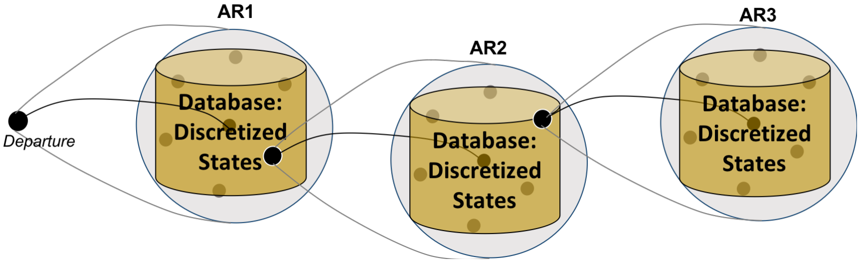

AR1
AR2
AR3
Departure
Database: Discretized States
Database: Discretized States
Database: Discretized States

Figure 5.8: Illustrating the process by which the discretized database of *a priori* known natural motion is leveraged within an AR

5.2.4 Database — Flow Models via Supervised Learning Strategies

The free-form search is powerful for identifying potential options for a new scenario where the design possibilities are unknown. However, the unrestricted approach diminishes the capacity to bound the motion to a particular geometric profile. Introducing an *a priori* discretized database comprised of select families addresses this challenge, but at the cost of increased computational memory and available solutions based on the grid/mesh for the implemented discretization. So, flow-models for these natural families are developed via Artificial Neural Networks (ANNs) and Support Vector Machines (SVMs) to address these drawbacks. The design strategy involves a regression network to fully determine the states for the conditions populated within an AR and then confirming that they belong to a family of interest via classification schemes. The computations involved with state determination and identification as well as the decision-making process for state selection are captured via the flow-diagram in Fig. 5.9. Only the states that are identified as 'belonging' to families of interest from a given AR are permitted for selection by an agent to construct the subsequent AR in support of the pathfinding effort. The details associated with the steps in the flow-diagram are explored as follows.

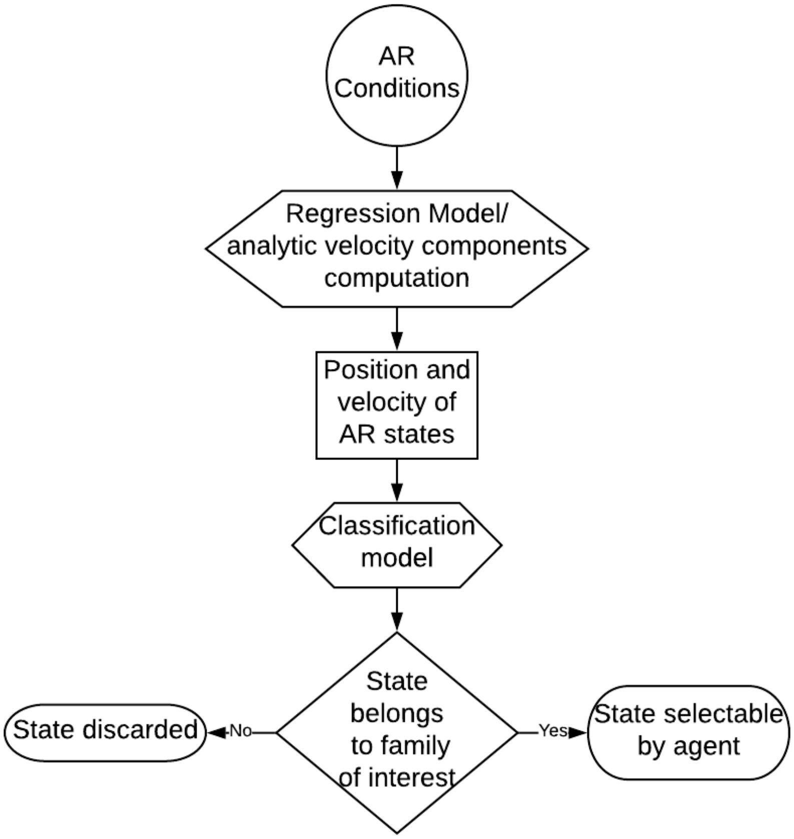

AR Conditions
Regression Model/ analytic velocity components computation
Position and velocity of AR states
Classification model
State belongs to family of interest
No
State discarded
Yes
State selectable by agent

Figure 5.9: Flow diagram summarizing computation and decision-making process for state selection from an AR via flow-models

Implementation of Regression Flow Models

The position and velocity attributes establish the unique identity of a state within a 3B system — assigning its ‘belonging’ to a particular periodic orbit family and its location within the family as well. Thus, a first step in incorporating specific family conditions within the searchable volume involves resolving the identity of populated data points within each AR. This task is accomplished by training neural networks to compute the velocity components associated with a family corresponding to positions populated in an AR. Therefore, the training input attributes must also be instantaneously extractable from within an AR. Such quantities include the position components of the points within an AR and the general direction of flow in the region as suggested by the in-plane and spatial velocity angles (ϕ and ψ) corresponding to the center AR state. These attributes are exploited to train the regression models as illustrated in Fig. 5.10. The states employed to train the ANNs are extracted from

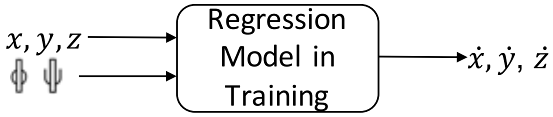


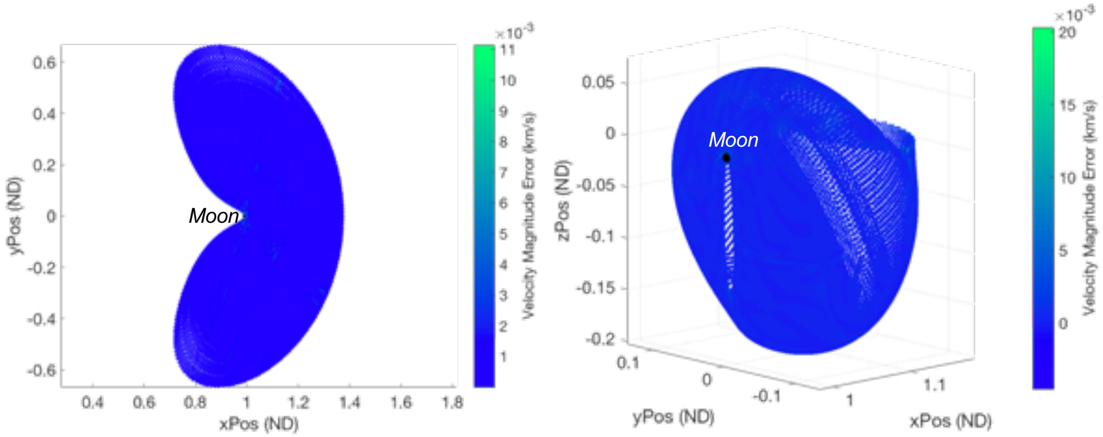
Figure 5.10: Input and output components for a regression model in training

the same conditions obtained via the discretization scheme explored in Section 5.2.3. A subset of the discretized dataset belonging to a particular periodic orbit family ($\approx 1\%$ of the total data, amounting to $\approx 60,000$ points) is partitioned into training, validation and test data for the ANN to develop the particular family model. The Levenberg-Marquardt technique along with evaluation of the Mean Square Errors (MSE) and early stopping conditions, as introduced in Section 4.1.1, are employed for the training process. Three-layer networks (2 hidden layer, 1 output layer) with ≈ 25 neurons per hidden layer on average have proven satisfactory in modeling the underlying relationships in the data for the various orbit families in this investigation. These neural network architecture components are design choices, and so, a modification to these values may be warranted for other scenarios.

The results associated with training both a planar and spatial orbit family are reported in Table 5.1. Recall that the computation of the Mean Squared Error (MSE) between the true and predicted values at the output layer of a regression ANN quantifies the quality of the model's training (Section 4.1.1). Each trained model is queried using the entire dataset for the corresponding periodic orbit family to assess their prediction capabilities for the remaining 99% of the unseen dataset. Figure 5.11 illustrates the outcomes for the planar L_2 Lyapunov family and the spatial L_2 southern halo orbit family flow models in terms of velocity prediction error. The largest discrepancies are observed to be near the primary. Insufficient training samples from

Family	Training MSE	Validation MSE	Testing MSE
(a) L_2 Lyapunov	$5.88e-08$	$6.30e-08$	$6.47e-08$
(b) L_2 Southern Halo	$1.84e-08$	$2.07e-08$	$2.62e-08$

Table 5.1: Quantification of training, validation and testing errors for sample periodic orbit family regression flow models. (a) L_2 Lyapunov, (b) L_2 southern halo.



(a) L_2 Lyapunov

(b) L_2 Southern halo

Figure 5.11: Examples demonstrating ability of regression ANN flow models to predict the velocities for (a) planar L_2 Lyapunov, (b) spatial L_2 southern halo orbit families.

a given region in the family also lead to poor prediction outcomes for that region. Further improvement on the training accuracy via further tuning of the ANN architectures is not pursued here, as these models satisfactorily aid in the generation of initial guesses for this preliminary investigation. During real-time pathfinding, the angular components (ϕ and ψ) for the AR center state's velocity only serve to aid in approximating the true velocity components for the positions of the other states in the AR. This implementation operates under the assumption that the flow-directions emanating from within a given AR are similar. Confidence in these approximations improves when verified by classification schemes.

An inspection of the figures in Fig. 5.12 demonstrates that a position can be non-unique for different velocity conditions, within the same family. A comparison of

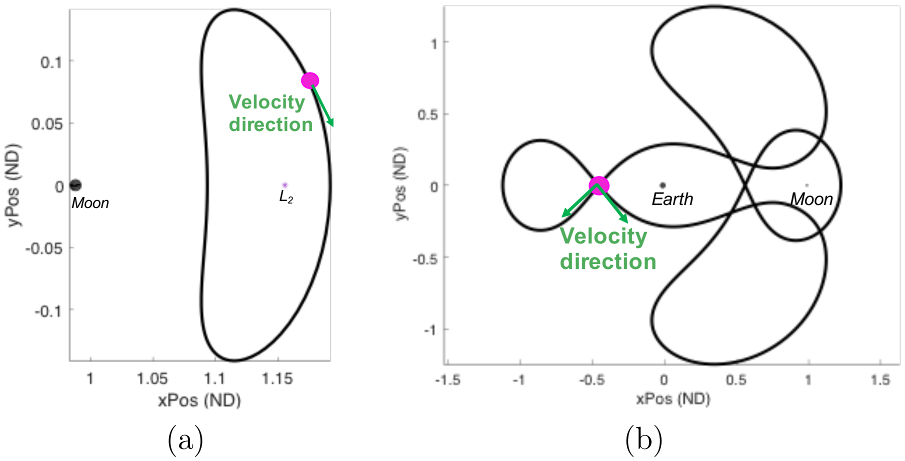


Figure 5.12: (a) Each position along an L_2 Lyapunov orbit possesses a unique velocity direction. (b) Each position along a Resonant 4:3 orbit does not possess a unique velocity direction.

Figs. 5.12(b) and 5.13 illustrate that large prediction errors do not necessarily occur at overlapping positions for a single orbit, but rather, where different family members overlap in position space as well. Thus, the results in Fig. 5.13 suggest that additional training inputs may be required by the regression model (Fig. 5.10) under more complex circumstances, as demonstrated for the Resonant 4:3 family. At these junctures,

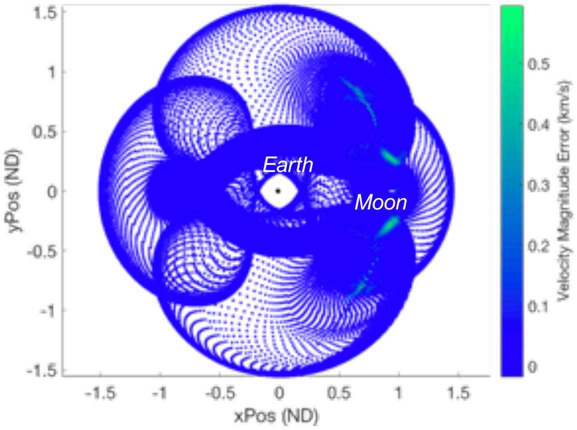


Figure 5.13: .

Challenge for regression models to resolve velocities for overlapping position conditions within a family, when no insight into the associated energy level for a state is provided as an input.

a lack of information about the differing energy levels (and, thus, velocity magnitudes) attributable to different family members inhibits accurate predictions. The velocity magnitudes are not available as inputs to the training process; because the exact velocity for a particular position within an AR is not known, and, in fact, is the output. Thus, an alternative strategy in contrast to Fig. 5.10 is employed as per Fig. 5.14. In the absence of a regression model, a set of velocity magnitudes are estimated (v_{est}) for each position state in the AR. These magnitudes range from the spacecraft instantaneous velocity at the center of the specified AR to a maximum value that establishes the radius of the velocity AR. Incorporation of the velocity magnitude estimate v_{est}, along with the angular facets (ϕ and ψ), resolves the cartesian velocity components analytically. Thus, a sufficiently large pool of solutions is populated to introduce to the classification models for filtration and identification. Both the regression models and the analytical computations successfully support pathfinding by delivering *family-like* velocity conditions when the inputs from an AR are in the vicinity of actual states from a family. The latter approach demands an increased online computational effort (i.e., during the pathfinding phase), whereas the former requires an offline time investment to train and pre-generate the regression models. The analytical approach is especially useful when the regression models are challenged to deliver predictions for the more complex periodic orbit families with multiple member overlaps. Nevertheless, the approximations and assumptions for computing the velocities aid the pathfinding agents in sequencing natural arcs that capture the *general* characteristics of the true conditions within each AR to generate a feasible initial guess — one that facilitates higher-fidelity analysis.

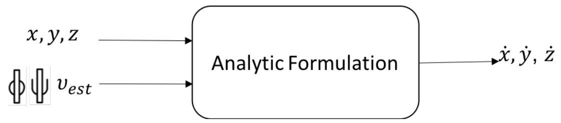


Figure 5.14: Analytic approach to computing velocity component approximations

In general, a regression network always produces an output based on the best-fit model determined during the training phase, even when the bounds on the input variables are violated. The analytical approach also always delivers a solution for the given inputs. So, recognizing invalid queries and disregarding the corresponding outputs is also important — one approach might include the specification of bounds for the inputs for each periodic orbit family regression/analytical model. However, this technique is not a failsafe process since families might assume complex structures in configuration space, possess self-intersections, or lack monotonic growth in space. Thus, classification models are also trained and adopted to judge the validity of the outputs of the regression models/analytic computations associated with a particular family. For the sake of clarity, a trained regression model is labelled $\mathbb{M}_R$ and a trained classification model as $\mathbb{M}_C$.

Implementation of Classification Flow Models

Classification of states into orbit families, or desirable and undesirable categories, are typically executed via various techniques. One approach is training models that distinguish conditions between a select variety of families as presented in Fig. 5.15(a). However, if an input state is introduced that does not belong to the available families in the trained classification model $\mathbb{M}_C$, it is re-assigned to a family with the closest relationship. An alternative, more sophisticated approach trains separate models for each family (Fig. 5.15(b)) — and a simple assessment determines if a given condition belongs. Thus, a binary classifier is trained. The two classes are — *Class1*: conditions from the desired family, and *Class2*: any other conditions that do not belong. This second class is termed *chaos* in association with a particular $\mathbb{M}_C$. It is generated by introducing random perturbations (within a prescribed radius) in position and velocity from conditions corresponding to each family (Fig. 5.16(a)). This perturbation radius is a design parameter. Figure 5.16(b) illustrates the binary classes that are employed to train a model for an L_2 Lyapunov family. The ANN and SVM

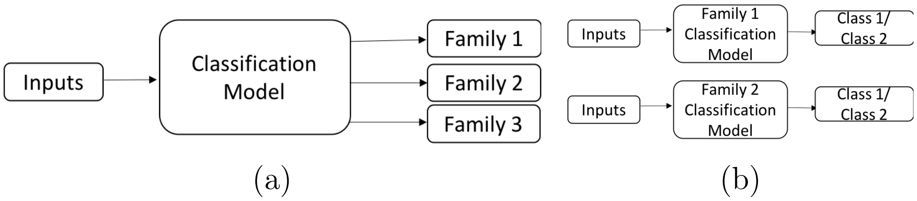


Figure 5.15: Example classification model options: (a) Train classifier to identify a state as belonging to one of the families available within the outputs. (b) Train a classifier to identify a state as belonging to a particular family or not.

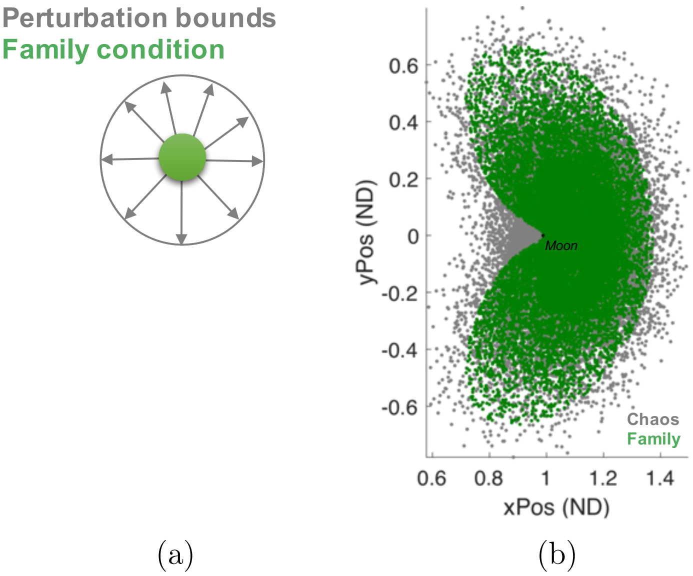


Figure 5.16: (a) Generating chaos for family conditions. (b) Family (green) and chaos (grey) conditions for the L_2 Lyapunov family.

binary classifiers are trained by supplying labeled examples from the two-classes, and minimizing the cross-entropy loss at the output layer in a 2-hidden layer network, by employing the Scaled Conjugate Gradient (SCG) descent routine (Section 4.1.1). These models are trained to accept the position and velocity information for a certain condition, and then report the associated classification output/posterior probabilities of the state belonging to the family/chaos. The state then belongs to the class with the greater value for these probabilities; Fig. 5.17 captures this process. These input conditions to an $\mathbb{M}_C$ can either be outputs from an $\mathbb{M}_R$ or state conditions selected directly from an AR (e.g., during free-form search). Once each of the conditions from an AR passes the regression and classification computations and requirements, they become available for selection by the pathfinding agents as per Fig. 5.9.

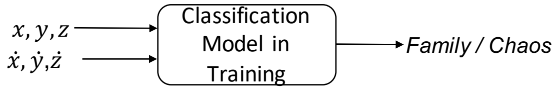
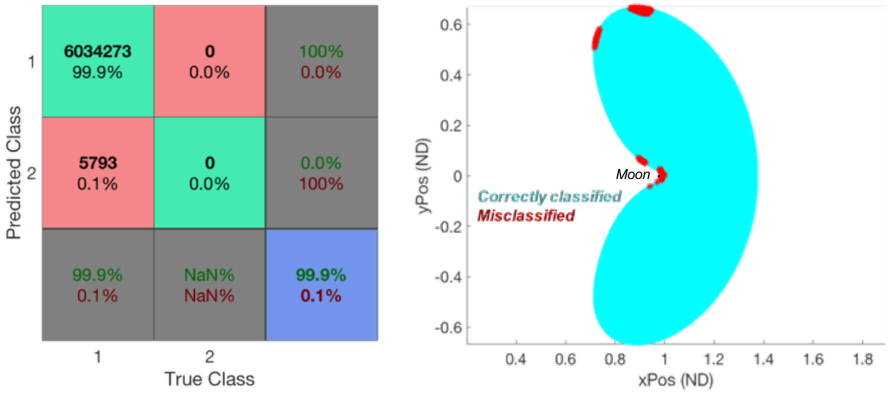


Figure 5.17: Input and output components for a classification model in training

Similar to the assessment of the accuracy of the regression networks, the trained classification models are also tested for reliability. Figure 5.18 illustrates a *confusion* matrix that aids in quantifying the accuracy of predictions from a trained model. In this example, the trained model is one that distinguishes between the true L_2 Lyapunov conditions (class 1) and the chaotic conditions (class 2) as introduced in Fig. 5.16(b). In this example, the performance of the trained model is tested by assessing its ability to accurately classify the entire population of a discretized periodic orbit family for an output layer probability cut-off of 50%; i.e., a state is identified as belonging to the L_2 Lyapunov family if the output probability for class 1 is $> 50\%$. The confusion matrix in Fig. 5.18(a) illustrates that there are no chaotic conditions in the input dataset and that $\approx 0.1\%$ of the total data have been misclassified as chaos. These misclassifications are identified in red in Fig. 5.18(b), and their locations are traced back to areas of relatively sparse L_2 Lyapunov training states from Fig. 5.16(b), and also the highly nonlinear region near the primary (Moon). Some approaches to increasing this accuracy include incorporating additional data in the areas corresponding to misclassifications as well as assessing the impact of varying the choices for the ANN architecture. Hence, training the models is an iterative process.

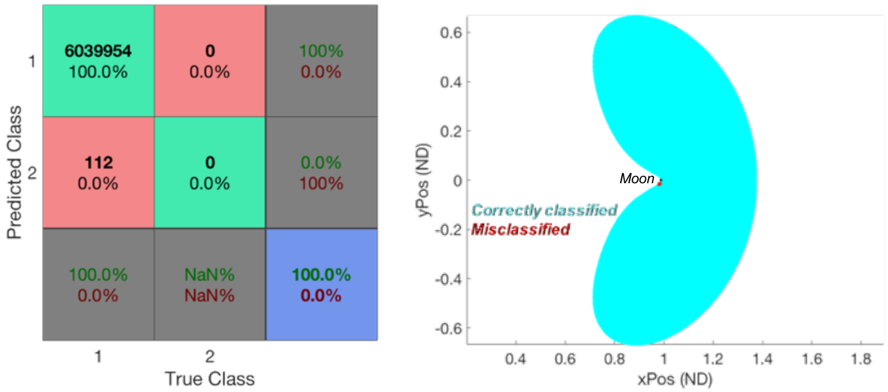


(a) (b)

Figure 5.18: Quantifying performance of trained L_2 Lyapunov ANN classification model on the full population of the periodic orbit family. Output layer probability cut-off = 0.5.

The SVM classification models are trained via Gaussian kernels that transform the data to a higher-dimensional space where they are separable. The values for the corresponding kernel hyper-parameter γ (in Eqn. (4.34)) and the box constraint C that aid in training SVM models are computed via cross-validation and Bayesian optimization [122] toolboxes available in MATLAB (as discussed in Section 4.1.2) that minimize the classification errors. The parameters γ and C are varied in unison so that the values that correspond to the minimum classification error for a subset of the training data are extracted to train the entire training data-set to develop the final L_2 Lyapunov family SVM model. A grid-search across ranges of γ and C is also a valid option to determine suitable values for these hyper-parameters.

The tuned hyper-parameters aid in constructing an optimized classifier, that subsequently leads to fewer classification errors on unseen data when compared to the ANN results in Fig. 5.18. Figure 5.19(a) reports the performance of the trained SVM model when it is directed to classify all the discretized conditions for the L_2 Lyapunov family. A smaller number of data points have been misclassified as chaos in Fig. 5.19(a) compared to Fig. 5.18(a), and the misclassifications that do occur exist closer to the primary (Fig. 5.19(b)), where the dynamics are highly nonlinear.

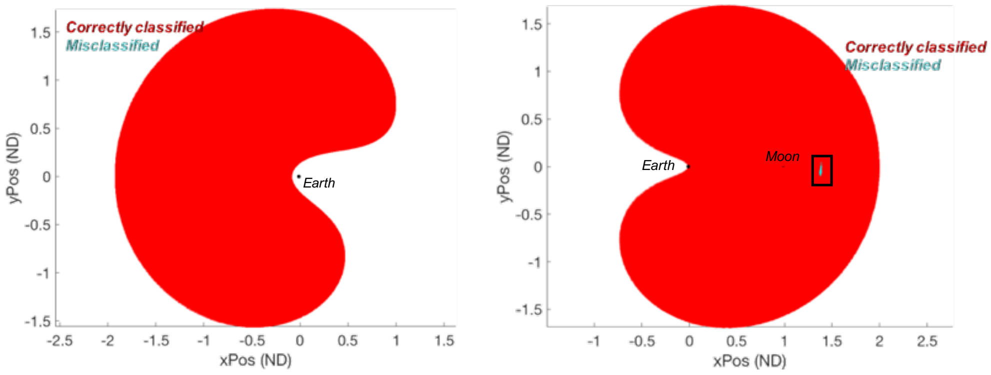


(a) (b)

Figure 5.19: Quantifying performance of trained L_2 Lyapunov SVM classification model on the full population of the periodic orbit family. Posterior probability cut-off = 0.5.

These binary classifiers are developed to identify any ‘non-belonging’ condition as *chaos*. Thus, an additional test checks whether these models are able to classify conditions belonging to other periodic orbit families as chaos. Referring to the previous discussion on the degree of overlap in position/velocity-space and also to the direction of flow along various periodic orbit families (e.g., Fig. 5.7), the L_2 Lyapunov SVM model accurately classifies all the L_4 short-period conditions as chaos (Fig. 5.20(a)), due to the low geometric similarities between these families. In contrast, certain regions of the DRO family that overlap in position and possess a semblance to the L_2 Lyapunov family flow characteristics are more challenging for the L_2 Lyapunov SVM model to label as *chaos*. These hard-to-distinguish areas are colored in cyan within the highlighted box in Fig. 5.20(b).

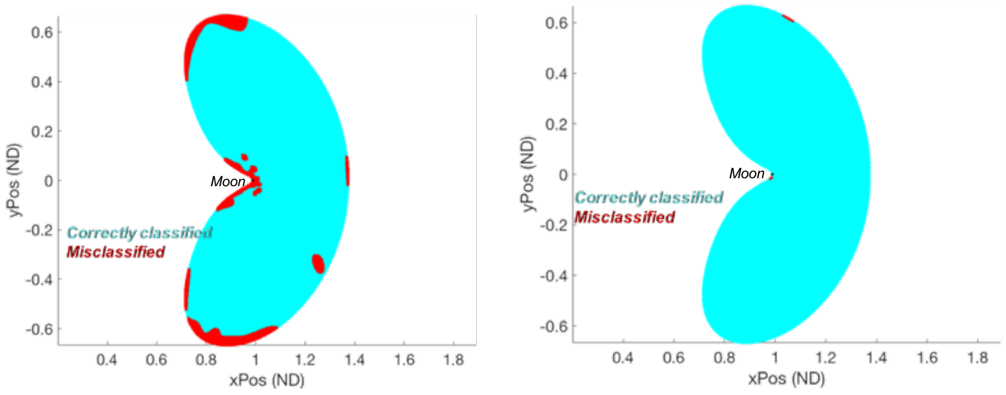
The output probabilities of the output layer/posterior probabilities always sum to 1. So, the higher the probability output associated with the *family* class, the higher the confidence that the condition is not chaotic. Note that a lower cut-off does assign conditions as *belonging to a family* that actually do not truly belong to the test family, but qualifies as family-like (e.g., L_2 Lyapunov-like) motion for selection during the pathfinding phase. Such a classification cut-off lever is, thus, exploited to capture



(a) (b)

Figure 5.20: Quantifying performance of trained L_2 Lyapunov SVM classification model by assessing its capability to identify other periodic orbit families as *chaos*. (a) L_4 short period family, (b) DRO family. The boxed region in (b) highlights the conditions that the L_2 Lyapunov model finds challenging to distinguish as a DRO/L_2 Lyapunov member due to the geometric similarities between these families in this particular region.

the desired general characteristics of the flow associated with a particular family, for supporting the path-finding phase. It is observed from Fig. 5.21, that the ANN $\mathbb{M}_C$



(a) (b)

Figure 5.21: Comparing performance of trained L_2 Lyapunov ANN and SVM classification models for the given training assumptions and choices, and highlighting the impact of selecting varied probability cut-offs (0.9 in this example). (a) ANN model (b) SVM model

is not as successful at correctly classifying many Lyapunov L_2 conditionsfor the same cut-off value, compared to the SVM model. As mentioned earlier, the accuracy of the ANN models could be improved with additional training data and architecture modifications. However, such an effort has not been pursued here and the ANN classifiers are not incorporated during the pathfinding phase. The SVM models, even with very similar training conditions, offer superior and sufficiently accurate classification models to demonstrate one of the primary aims of this investigation, the ability to constrain the geometric profile of the initial guess to involve only specified families. Passing a classification test marks a regressed input-output pair (position and velocity states) as acceptable for selection from an AR by the pathfinding phase. The implementation of both the exact and heuristic pathfinding approaches are explored in this analysis.

5.3 Implementation of Automated Pathfinding

Pathfinding is implemented to *sequence* the appropriate conditions for maneuvering a s/c from origin to destination. The available conditions are the natural states populated within an AR either randomly or via the flow-models or discretized database; they constitute the *frontier* of solutions at any given time as introduced in Section 4.2. Both the exact (Dijkstra's algorithm) and heuristic (HARL) algorithms select a state from within each frontier to continue the search based on varying metrics. A sample schematic illustrating the search process is displayed in Fig. 5.22. Note that the black arcs result from propagating an unperturbed natural condition from a spacecraft's current state; the dimension of each computed AR is a function of the respective velocity perturbation magnitude δV_n and the propagation duration; and the time required to reach any state within a particular AR is equivalent to this propagation duration required to reach its center state (the state at the end of an unperturbed natural arc propagation). According to the discussion in Section 5.1 and Fig. 5.3, a $\Delta V = 0\ m/s$ is required to traverse the natural arcs to the center of an AR, and the ΔV_n is approximated to be equal to $\delta V_n\ m/s$ to reach any state along an AR's circumference. So, the maneuver size required to reach any state along the radius of an AR is approximated to lie between these values, i.e., $0\ m/s \leq \Delta V_n \leq \delta V_n\ m/s$. The pathfinding process terminates when the destination condition falls within an AR with HARL or is reached via the minimum cost path using Dijkstra's algorithm. This destination condition could be any state belonging to a discretized arrival orbit or could be a specific user-defined destination. When implementing either of the pathfinding schemes, the overarching goal is posed as an optimization problem. For example, the following relationships present the prioritization of minimizing propellant consumption (m_p) / total ΔV versus minimization of the transfer duration (t_f) to address the persistent *mass-time* tradeoff challenge:

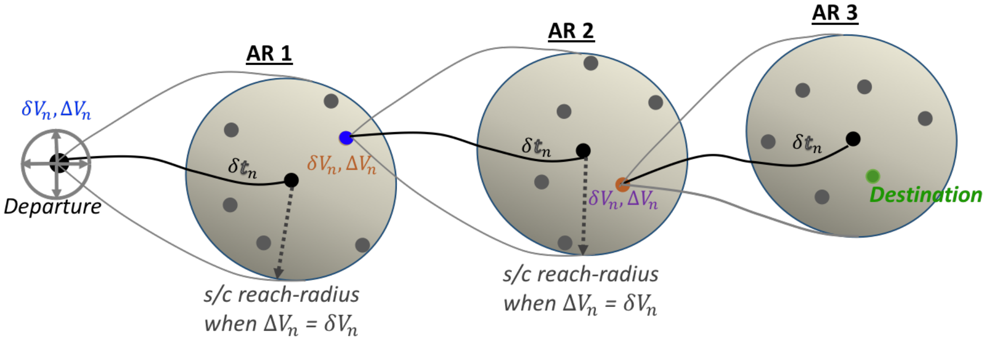


Figure 5.22: Schematic illustrating the roles of δV_n, ΔV_n and δt_n in single node expansions from a given AR. The colored conditions are states selected from an AR to enable a sample sequenced pathway. The black arcs represent the natural propagations, i.e. propagations when $\Delta V_n = 0\ m/s$. Note that $0\ m/s \leq \Delta V_n \leq \delta V_n\ m/s$ to reach the states populated within an AR, and $\Delta V_n = \delta V_n\ m/s$ to reach the states along its circumference.

$$min\ \mathbb{J} = W_t t_f + W_p m_p \tag{5.2a}$$

$$min\ \mathbb{J} = W_t t_f + W_p \Delta V \tag{5.2b}$$

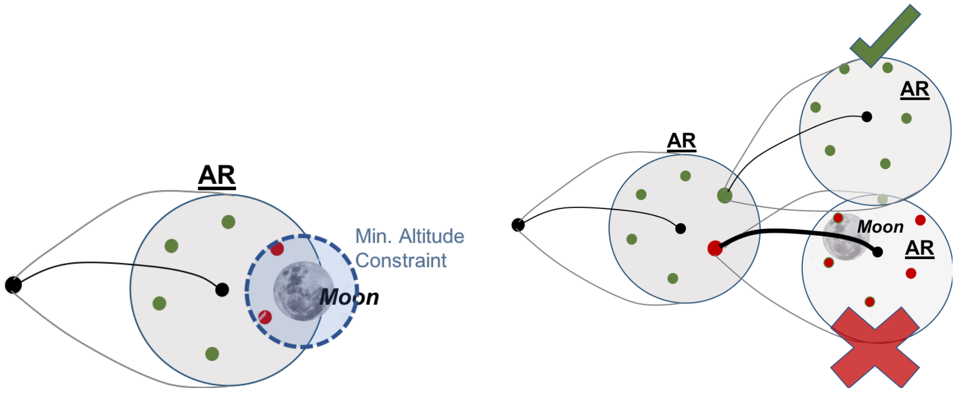
where, the propellant mass m_p or ΔV are always weighted against the transfer duration, t_f, such that the weighting on TOF, W_t is always 1, and W_p is a design variable. Note that the ΔV represents the sum of the maneuver costs accumulated to transfer from one frontier to another within the sequenced path ($\sum \Delta V_n$), whereas δV_n in Eqn. (5.1) is the velocity perturbation computed to generate each of the ARs in this sequence. The transfer duration t_f in Eqn. (5.2) is the aggregate sum of the propagation durations associated with each natural arc assembled within the transfer sequence. The propellant consumption/ΔV_n are computed as a result of the velocity discontinuities between each of these arcs. In this preliminary investigation, the pursuit of a modular architecture to accommodate varied engine characteristics leads to the approximation of low-thrust flow with chemical impulses for the initial guess (IG) generation phase. Consequently, the EOM associated with the mass flow is not propagated for the IG generation. Rather, the propellant consumption history is post-processed analytically via the rocket equation upon computation of each ΔV_n in the pathfinding sequence, increasing computational efficiency.

The flexibility of the framework admits alternative cost function formulations as well. For example, much like the mass-time trade, spacecraft rendezvous is another challenging goal in mission design. In a rendezvous problem, a chaser s/c seeks to meet the exact time-dependent states of a target object. One approach to realizing this goal given the existing framework is to (i) exploit the successive AR computation scheme for a chaser s/c to arrive at states along a target object's path, and (ii) simultaneously minimize the difference in epoch between the chaser s/c terminal state that also corresponds to the target state at some instant in time. The global cost function is mathematically formulated in the form:

$$min \ \mathbb{J} = |epoch_{fC} - epoch_{fT}| \tag{5.3}$$

where the subscripts fC and fT indicate the final conditions for the chaser s/c and target, respectively. Note that Eqn. (5.3) does not incorporate information about the chaser or target states. Given that satisfaction of the state and epoch component requirements for a rendezvous are independent in the implemented approach, the pathfinding algorithm does not uncover a path that truly completes the rendezvous with the target; rather, a path with minimal error in terms of the rendezvous epoch is sought. Such an implementation serves to construct a feasible initial guess for a numerical corrections process that enforces a strict rendezvous constraint. Although unexplored in this investigation, alternative implementations where the rendezvous constraint is enforced during the initial guess generation process also produce viable options.

The role of the global cost function is to determine beneficial routes by assessing the alignment of a transfer path metrics to the user requirements. The construction of these pathways results from the active pathfinding agents; the arcs are selected from available states within a given AR that do not violate user-defined constraints. Some examples involving altitude violations lead to a state being discarded from an AR as illustrated in Fig. 5.23. The states highlighted in red violate a minimum altitude constraint relative to the primary and are, thus, not selectable to progress the AR sequence. In another scenario (Fig. 5.23(b)), a state that leads to a propagation through the minimum altitude constraint is also not considered for selection. In this investigation, the minimum altitude constraint enforced during initial guess generation is 250 km from a primary surface. The pathfinding implementations specific to



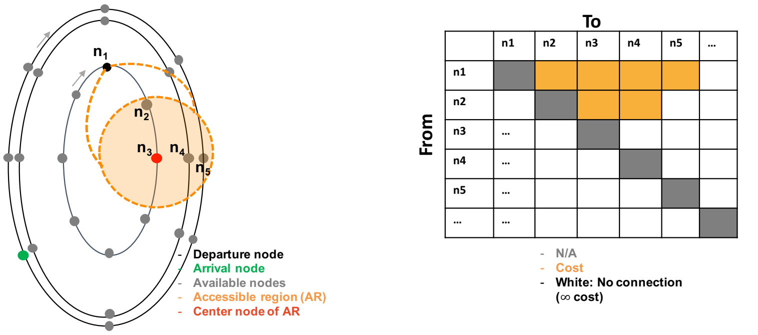
(a) (b)

Figure 5.23: Schematics — Examples of altitude violation that terminate an agent's search toward the destination

the exact method (Dijkstra's algorithm) as well as the HARL approach distinguish the selection of an AR state.

5.3.1 Implementation of Dijkstra's Algorithm

In this investigation, Dijkstra's algorithm is implemented as a greedy search method that always selects the node from a frontier with the lowest cost since departure. The node-to-node costs are pre-computed (parallelizable process) and stored in a cost adjacency matrix as depicted in Figure 5.24(b). The matrix is purposed as a directed graph that informs the agent of unique costs to traverse between the nodes, including an infinite cost when no connections exist between them. These costs feed the priority queue discussed in Section 4.2.1. The lowest-cost route to the destination emerges in a serial fashion to allow an update to the cost from departure to a specified node when a less expensive connection is established. In pursuing the mass-time priority objective, using Eqn. (5.2a) to measure the inter-node costs requires on-demand computation to establish the ARs based on varying mass history. The mix of parallel and serial components during the search, in addition to the storage of an extra variable (mass-history) for each node contributes to an increased time-space complexity and a more challenging algorithm implementation. Thus, for both



(a) Accessible nodes

To

From	n1	n2	n3	n4	n5	...
n1	■	🟧	🟧	🟧	🟧	
n2		■	🟧	🟧		
n3	...		■			
n4	...			■		
n5	...				■	
...	...					■

- N/A
- Cost
- White: No connection (∞ cost)

(b) Cost adjacency matrix

Figure 5.24: Dijkstra's algorithm implementation — (a) A schematic of a sample AR from a node (n_1). (b) A sample cost adjacency matrix.

the low-thrust and the chemical transfer scenarios, the inter-node cost (*arc* cost) is estimated using Eqn. (5.2b) where the ΔV cost is weighted against transfer time to satisfy the mass-time priority directive. The s/c final delivered mass is extracted by incorporating knowledge of the total expended ΔV value into the rocket equation. The resulting end-to-end transfer sequence is a global optimum under the earlier s/c assumptions including a lower-fidelity model. The rendezvous problem is not pursued via Dijkstra's algorithm in this investigation.

5.3.2 Implementation of HARL Algorithm

A HARL algorithm does not guarantee optimality due to the inclusion of heuristics, but multiple search episodes enable convergence to a nearly-optimal solution to address the global objective. The parallelizable capabilities within HARL offers a flexible architecture where the inter-node costs (*arc* costs) are computed on-demand as an agent progresses in its search towards the destination. As a result, either equation in Eqn. (5.2) may be adopted. Given the process in Fig. 4.9, $\approx$ 50 agents per episode, and $\approx$ 30 episodes per transfer scenario are representative quantities for the HARL implementation that have proven sufficient to uncover initial guesses for end-to-end pathways that link the departure and destination conditions in this investigation. These quantities are design choices and may be varied depending on the application. Furthermore, the search for a transfer path is posed as an episodic problem, i.e., the search by a particular agent terminates if its transfer-path duration exceeds an imposed upper-bound. The other conditions for search termination are arrival at the destination condition or violation of a user-defined constraint (e.g., altitude from a primary). Note that the pathway search over all the agents within a given episode is typically run in parallel. An agent within an episode (i) initiates the search for a suitable initial guess by randomly exploring the state-space (selecting states at random from within the ARs) and (ii) gradually employs more exploitation (capitalization on previously gained knowledge about the state-space) over subsequent episodes as specified by Eqn. (4.40). During 'exploitation', an agent only selects from nodes that have previously supported successful transport to the destination and are, thus, already reinforced. The selection of a particular reinforced node from an AR during the 'exploitation' phase is guided by a heuristic $\mathbb{H}_n$, one that is crafted to accommodate two aspects during the decision making process: (a) the propellant consumption required to transfer from the current state to the reinforced state within the AR, and (b) a measure of 'goodness' d_n, for the selected node in terms of its proximity to the target condition(s) in both the position and velocity dimensions. These heuristic considerations are visualized in Fig. 5.25. The propellant consumption is evaluated in proxy

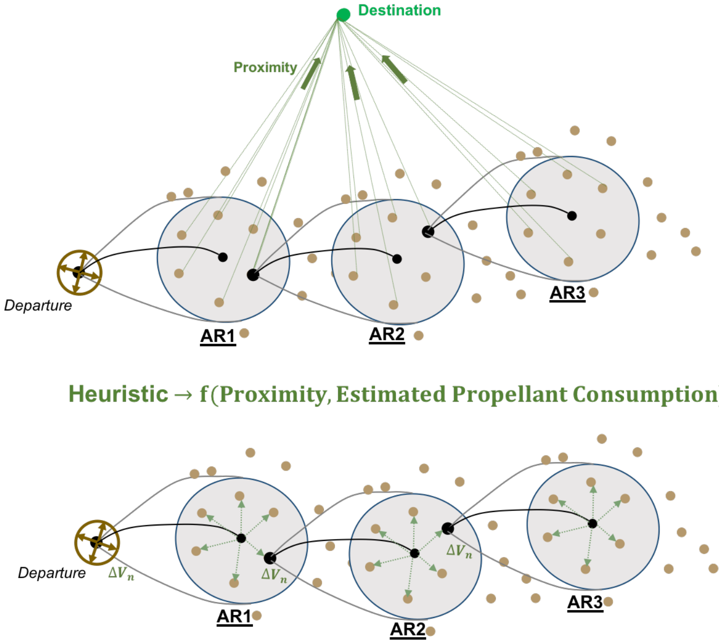


Figure 5.25: Schematic of Employed Heuristic Function Components

by the amount of ΔV required for the state transition, or the s/c mass at the end of a propagated arc after such a maneuver/burn ${m_f}_n$. These factors are incorporated mathematically into the heuristic function to yield:

$$\mathbb{H}_n = \frac{{m_f}_n^{W_m}}{{d_n}^{W_d}} \tag{5.4a}$$

$$\mathbb{H}_n = \frac{1}{{\Delta V_n}^{W_v} \times {d_n}^{W_d}} \tag{5.4b}$$

$$\text{where}, d_n = ||\boldsymbol{\chi}_n - \boldsymbol{\chi}_T|| \tag{5.4c}$$

where $\boldsymbol{\chi}_n$ is the state vector (position and velocity) associated with a particular node in the AR, and $\boldsymbol{\chi}_T$ is the state vector corresponding to the target condition(s). The weights are also design variables, where W_v and W_m grant control over the desired s/c performance (ΔV_n / delivered mass) and W_d influences the manner in which the transfer trajectory tends towards the destination. These qualities also render the heuristic applicable as a measure of reward. Specifically, the reward for a state-action pair contributing to a successful transfer path is computed as the accrued sum of the heuristic values along the path ($R_n = \sum \mathbb{H}_n$). The sheer volume of the state-space and state-action combinations in this investigation has prompted the tracking of only a node's relevance to the transfer by recording its cumulative returns over the evaluated episodes within a global repository. This implementation is a variation on the traditional approach in state-action-value reinforcement learning algorithms where a table of the state-action pairs and corresponding values are maintained. The actions and their impact are, however, incorporated by the generation of the ARs that constrain the accessibility between nodes. The action 'a' undertaken from the current state 's' is the executed $\Delta\mathbf{V}_n$ and is processed as a function of the state s' selected from within the associated AR (as described in Section 5.1). Recall that the state s' is selected randomly during 'exploration' or as influenced via heuristics during 'exploitation', as examined in Section 4.2.2. The discount factor γ is set equal to zero as a design choice in this preliminary investigation, where future studies of the impact of variation in its values may offer additional insights. In contrast to the Monte Carlo [126] based approach that computes returns for entire episodes, a temporal difference [126] approach where the state-action returns are computed prior to the completion of an episode are also beneficial for investigation. Note that the node-to-node cost is variable and a function of the s/c mass history. So, the automated pathfinding strategy in this investigation aims to establish general desirable pathways to the destination rather than identify distinct and discrete beneficial nodes in configuration space. This goal is pursued by firstly identifying top ranking transfers according to the global cost function (Eqn. (5.2)), followed by reinforcing the states along such paths according to their respective rewards received and also reinforcing the neighboring states via the same rewards, as illustrated by the schematic in Fig. 5.26. As a consequence, additional local variability is introduced even during the exploitation phase to encourage local improvements during the search for a pathway that addresses the global objectives. Furthermore, the quality of a node is only a guide in the overall search strategy, as stochasticity is introduced in the selection of a favorable node from within an AR with the probability $\mathscr{P}_n$ during the exploitation phase:

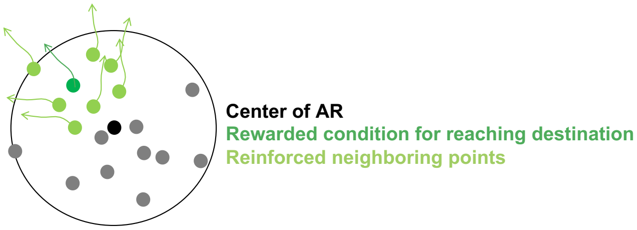

Center of AR
Rewarded condition for reaching destination
Reinforced neighboring points

Figure 5.26: Reinforcing neighbors of the rewarded condition within an AR to aid the pathfinding process and also introduce further variability to the stochastic search.

$$\mathscr{P}_n = \frac{\mathbb{F}_n \ \mathbb{H}_n}{\sum \mathbb{F}_n \ \mathbb{H}_n} \tag{5.5}$$

This probability $\mathscr{P}_n$ emphasizes two important considerations that bias the selection of a node n: (a) perceived relevance of the node to the transfer via its accrued returns ($\mathbb{F}_n$) over episodes, and (b) its current attractiveness as represented by the heuristic $\mathbb{H}_n$. The HARL design choices are enforced within a distributed, cooperative environment where the incremental knowledge acquired by each agent combines to furnish attractive transfer trajectories that are then ranked by their overall performance (e.g., Eqn. 5.2) over multiple episodes. The aim of this process is the extraction of the transfer solution that optimizes the global cost function while satisfying the goals and constraints.

5.4 Initial Guess to Higher-Fidelity Solution

In this investigation, a final solution is defined as an end-to-end transfer trajectory that adheres to the constraints imposed within a CR3BP regime, augmented by the selected engine forces. Note that such a solution is merely an initial guess for a simulation in the CR3BP and the higher-fidelity ephemeris model. The information associated with the arcs that are assembled by the automated pathfinding process is sufficient to initiate the numerical corrections process described in Chapter 3, namely via knowledge of the position, velocity, mass, thrust direction, thrust magnitude and time estimates. The user possesses the freedom to select a desired computational tool for executing the convergence/optimization process. Note that varying thrust magnitudes are interspersed within the lower-fidelity solution, thus, it can serve as an initial guess for a range of engine capabilities in a higher-fidelity simulation. However, the range of thrust magnitudes is bounded in the corrections updates.

6. APPLICATIONS AND RESULTS

The execution of the components in the design framework produce initial guesses for diverse transfer scenarios and then transition these solutions to a continuous solution via a higher-fidelity engine model. The converged solutions in the CR3BP are then exploited as an initial guess to construct solutions that account for varied gravitational effects from the epoch-dependent locations of the primaries, including the Sun. The potential of the automated pathfinding strategy and design framework is demonstrated via the following broad applications in both the planar and spatial realms:

1. *Validate Design Concept — Initial Guess Generation*

   The proposed framework is assessed for its capability to deliver an initial guess via Dijkstra's and the HARL pathfinding algorithms. The challenges and advantages associated with incorporating each pathfinding approach for initial guess generation are also explored.

2. *Explore Mass-Time Trade*

   The ability to prioritize global objectives such as the balance of transfer duration and payload delivery is investigated.

3. *Demonstrate Transition of Initial Guess to Continuous Solution*

   Initial guesses are constructed via discrete natural arcs. Accordingly, the details associated with extraction of the appropriate information to initiate the numerical corrections process is examined.

4. *Exploit Intermediate Bridge Families to Accomplish Mission Goals*

   A priori knowledge of periodic motion in the CR3BP is incorporated to enable specific mission scenarios.

5. *Transfer Geometry Considerations*

   Deliberate choices regarding the implementation of the search strategy and natural motion are exploited to enforce the rate of transition towards the destination and the region of space traversed by the spacecraft. The flexibility of the HARL algorithm to accommodate discretized, flow-models, and free-form natural condition transport methods is highlighted. The differences in outcomes to be expected with each of these three natural condition transport approaches for initial guess generation are also discussed.

6. *Incorporate s/c Performance Specifications*

   From cubesat-class to higher low-thrust and chemical impulse capabilities, the consequences of varying the acceleration levels for a particular mission scenario are considered.

7. *Demonstrate s/c Recovery and Rendezvous Capability*

   Failures such as thrust degradation are imposed along a nominal flight path and the ability to recover the transfer to deliver a s/c to its destination is explored. The ability to execute rendezvous with a pre-positioned target at the destination is also investigated.

8. *Transition CR3BP Solutions to Higher-Fidelity Ephemeris Regime*

   The existence of higher-fidelity ephemeris solutions corresponding to the CR3BP transfers are assessed, and the impact of the epoch-dependent gravitational forces on the ephemeris trajectories is also examined.

6.1 Initial Guess Generation Capability

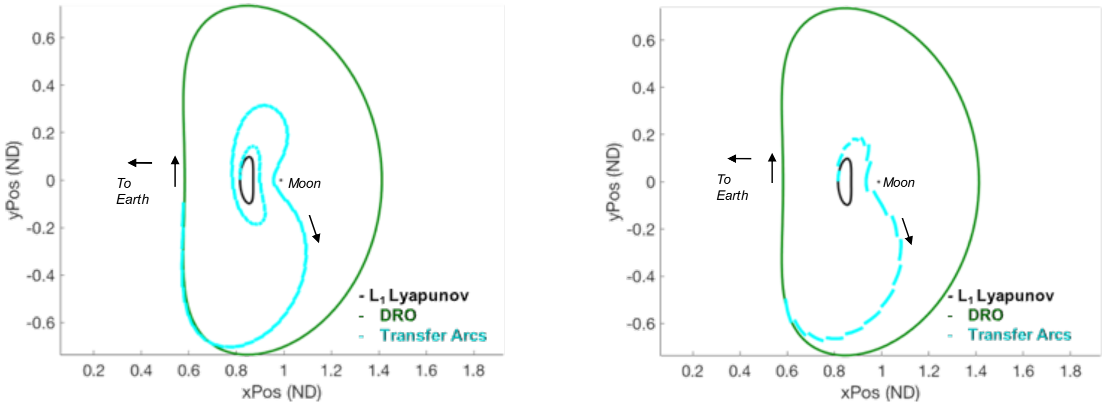
The validation of the design process is established by a proof-of-concept; i.e., assessing the proposed framework for its capability to deliver an initial guess. The initial guess for a particular transfer scenario is constructed by exploiting both Dijkstra's and the HARL algorithms to sequence advantageous natural arcs from the

Accessible Regions (ARs), and thus, compute an end-to-end path from departure to destination. The associated challenges and advantages with the implementation of these algorithms and the related outcomes are explored. The type of natural condition transport approach (*a priori* discretized database, flow-models, or free-form) are also identified within the *Performance metrics* tables for each initial guess that is constructed in this chapter.

6.1.1 Exploiting Dijkstra's Algorithm

The transfers explored via Dijkstra's algorithm are constructed for a low-thrust s/c with $\widetilde{\mathbb{T}} = 0.22\ N$, $\widetilde{Isp} = 4000\ s$. The s/c acceleration, a function of the thrust magnitude and s/c mass, $\frac{\widetilde{\mathbb{T}}}{M_0}$, is set equal to $a \approx 9.7 \times 10^{-4}\ m/s^2$, an unrealistically high value in this example, given today's low-thrust capabilities at $\approx 2 \times 10^{-4} m/s^2$ (Deep Space 1, DS1). This high value is adopted to balance high computational times associated with Dijkstra's algorithm and the ability to *reach* discretized conditions within the searchable database via sufficiently large δV (Eqn. (5.1)) for relatively low burn durations ($\delta \mathbb{t}_n$). The combination of these low $\delta \mathbb{t}_n$ values and resulting ΔV magnitudes aid in reducing the computation time associated with Dijkstra's algorithm. Figure 6.1 illustrates two different global optima for two transfer cases from an L_1 Lyapunov orbit to a DRO via different imposed engine burn durations. The inter-orbit and intra-orbit discretization schemes are consistent between both scenarios and plays a key role in the number of nodes that are *reachable* during each segment.

The 0.5 *day* propagations ($\delta V \approx 40\ m/s$ per segment) results in a cost adjacency matrix with 24,550,337 available connections; larger accessible regions resulting from 1 *day* burn durations ($\delta V \approx 84\ m/s$ per segment) leads to ≈ 20 times more connections (489,428,125). A more efficient and faster transfer is uncovered in Fig. 6.1(b) with access to a greater variety of conditions within the database. However, the price for soliciting a broader search capability with Dijkstra's algorithm is computation



(a) $\delta\mathbb{t}_n = 0.5\ days$ (b) $\delta\mathbb{t}_n = 1\ day$

Figure 6.1: Demonstrating ability to uncover initial guesses using Dijkstra's algorithm for varied burn durations $\delta\mathbb{t}_n$. Departure: L_1 Lyapunov orbit ($JC = 3.147$), Destination: DRO ($JC = 2.785$)

Table 6.1: Performance metrics for Fig. 6.1. Demonstrating ability to uncover initial guesses using Dijkstra's algorithm for varied burn durations $\delta\mathbb{t}_n$

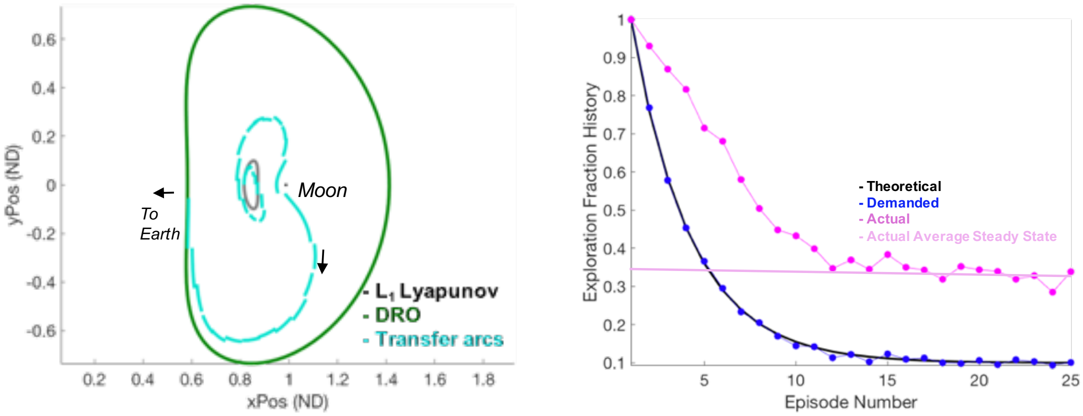
Scenario	TOF_{est} (days)	$\frac{m_f}{m_0}_{est}$ (%)
Fig. 6.1(a): $\delta\mathbb{t}_n = 0.5\ days$	36.5	97.86
Fig. 6.1(b): $\delta\mathbb{t}_n = 1\ day$	18.0	98.70

<table>
<tr><td colspan="3">Natural Condition Transport Approach — <i>A Priori</i> Discretized Database</td></tr>
<tr><td colspan="3">s/c Specifications</td></tr>
<tr><td>$a_0 = 9.7 \times 10^{-4}\ m/s^2$</td><td>$\widetilde{\mathbb{T}} = 0.22\ N$</td><td>$\widetilde{Isp} = 4000\ s$</td></tr>
</table>

time. The transfer in Fig. 6.1(a) is constructed in 6.22 *hours*, whereas computing the shortest path amongst many more possibilities in Fig. 6.1(b) compounds the computation time to 21.26 *hours*. Note that the adjacency matrix computation is parallelizable, and contributes to ≈ 1.5 *hours* of the total computation time for each sample case, but Dijkstra's search for the optimal route through the adjacency matrix connections cannot be parallelized. The pathfinding routine is implemented via a naive priority queue approach that sorts the node connections from least costly to most expensive. Dijkstra's finite search duration efficiency can be improved with data structure implementations such as *heaps.* Such avenues have not been pursued in this investigation because, regardless of the implementation choice, an increased computation duration is unavoidable with Dijkstra's algorithm when additional nodes/connections are introduced to the search. Such a disadvantage limits the number of orbit families that are incorporated into the searchable database as well, thus diminishing the variety of initial guess solutions that are uncovered for the transfers. The transfer scenarios in Fig. 6.1 are implemented using Matlab 2017a on a 64-bit, Intel Xeon 2.60GHz server with 24 cores and a Red Hat 6 operating system. Note that the *space* complexity also increases with additional nodes, as a record of the resulting additional connections are also required to be maintained in computer memory.

6.1.2 Exploiting HARL Algorithm

In contrast to the challenges observed with Dijkstra's algorithm, *space* complexity with the HARL algorithm is managed by only tracking the rewards associated with a particular node rather than its connections as well (Section 5.3.2). Additionally, the computation time associated with the HARL algorithm does not necessarily increase with the introduction of additional nodes. The search efficiency is, however, impacted by the quality of the heuristics and the *reach* of the s/c via its accessible regions. For the same transfer scenario introduced in Fig. 6.1, the combination of the accessible regions afforded by the 0.5 *day* propagations and the stochastic nature of HARL lead to the inability to compute a solution in the 6.22 *hours* required by Dijkstra's algorithm. However, increasing the propagation duration to 1 *day* per segment grows the ARs and, thus, improves the ability to deliver a transfer in a shorter search duration. Figure 6.2(a) illustrates a solution that presents similar transfer characteristics to those in Fig. 6.1(a), but only requires ≈ 1.5 *hours*, and thus $\approx 25\%$ of the computation time. The reduction in computation time is even more noticeable in a 1:1 comparison between the solutions acquired using longer propagation durations at 1 *day* per segment and different methods; the HARL solution is computed in 7% of the time required by Dijkstra's implementation in Fig. 6.1(b). As noted in Section 4.2.2, a comparison of Fig. 6.2(a) with the Dijkstra solution in Fig. 6.1(b) reveals that the HARL algorithm does not necessarily produce the global optimum for a prescribed set of design conditions due to its stochastic nature. However, recognizing the advantages in computational time and storage savings, the HARL algorithm is preferentially adopted for most transfer scenarios in this investigation.



(a) Transfer arcs IG

(b) Exploration history

Figure 6.2: (a) Transfer profile and (b) exploration history associated with $\delta\mathbb{t}_n = 1$ *day* search via HARL. Departure: L_1 Lyapunov orbit ($JC = 3.147$), Destination: DRO ($JC = 2.785$)

Table 6.2: Performance metrics for Fig. 6.2. Performance metrics for transfer via HARL for $\delta\mathbb{t}_n = 1$ *day*.

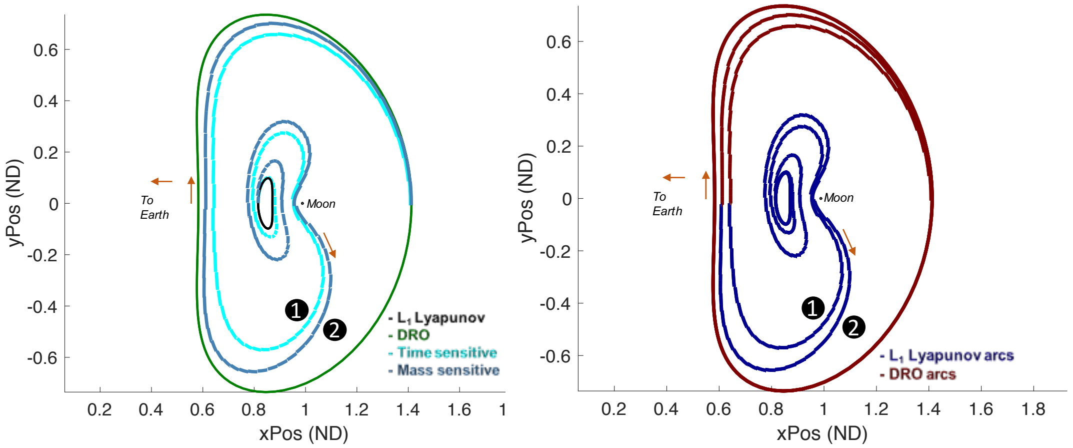
Scenario	TOF_{est} (days)	$\frac{m_f}{m_0}_{est}$ (%)
Fig. 6.2(a)	36	97.05
Natural Condition Transport Approach — *A Priori* Discretized Database		
s/c Specifications		
$a_0 = 9.7 \times 10^{-4}\ m/s^2$	$\widetilde{\mathbb{T}} = 0.22\ N$	$\widetilde{Isp} = 4000\ s$

The amount of 'exploration' by the agents over 25 episodes for the transfer scenario represented in Fig. 6.2(a) is captured in Fig. 6.2(b). Note that, given the design framework options, the natural conditions within the discretized database cannot be reinforced for 'exploitation' until at least one agent uncovers a successful end-to-end route. The theoretical exploration values (black curve), computed via Eqn. (4.40), and the level of exploration *demanded* during the practical search (blue curve) based on the random values of $\mathbb{q}$ drawn from a uniform distribution (Eqn. (4.39)), align well. Even though the general trend of exploration reduction over subsequent episodes is maintained, the actual exploration executed by the agents over each episode is offset from the theoretical and demanded values. Such an outcome is a consequence of the agents being required to revert to 'exploration' if they encounter ARs where the value of $\mathbb{q}$ demands 'exploitation', but there exists no knowledge of beneficial conditions available to 'exploit' that successfully reach the destination. Such a reversion directive is enforced to equip the agents to address the challenge of navigating an expansive state-space and be successful at uncovering pathways to the destination. The exploration profile could potentially be modulated towards the theoretical curve by experimenting with an increased number of states that are reinforced in the neighborhood of rewarded conditions (Fig. 5.26) per successful episode. Such trade-offs are not pursued in this preliminary investigation.

6.2 Explore Mass-Time Trade

The balance of propellant consumption versus TOF is a persistent challenge in space travel. Figure 6.3 demonstrates the ability of the design framework to prioritize such outcomes during the design phase. To enable suitable comparisons, the departure and arrival nodes are constrained to be the same between both the scenarios constructed in Fig. 6.3 using Dijkstra's algorithm. A burn duration of $\delta\mathbb{t}_n = 0.5$ *days* per segment facilitates the various intermediate arcs leveraged during the transfers in Fig. 6.3; the s/c slides through members of the L_1 Lyapunov family prior to drifting into the DRO family for the terminal orbit entry phase. These transfers are computed over a similar computation interval to those uncovered using Dijkstra's algorithm in

Fig. 6.1(a). The development of the discontinuous initial guess is followed by its transition to continuous low-thrust solutions.



(a) Transfer arcs IG

(b) IGs colored by family

Figure 6.3: Demonstrating influence over mass-time priority for trajectory design (Dijkstra's algorithm employed). (a1 & b1) Time-priority transfer, (a2 & b2) Mass-priority transfer. Departure: L_1 Lyapunov orbit ($JC = 3.147$), Destination: DRO ($JC = 2.785$)

Table 6.3: Performance metrics for Fig. 6.3. Mass-time priority comparisons for IGs constructed via Dijkstra's algorithm.

Scenario	W_p	TOF_{est} (days)	$\frac{m_f}{m_0}_{est}$ (%)
Fig. 6.3 (a1): Time-priority	1e-5	44.5	96.4
Fig. 6.3 (a2): Mass-priority	1e5	52.0	98.8

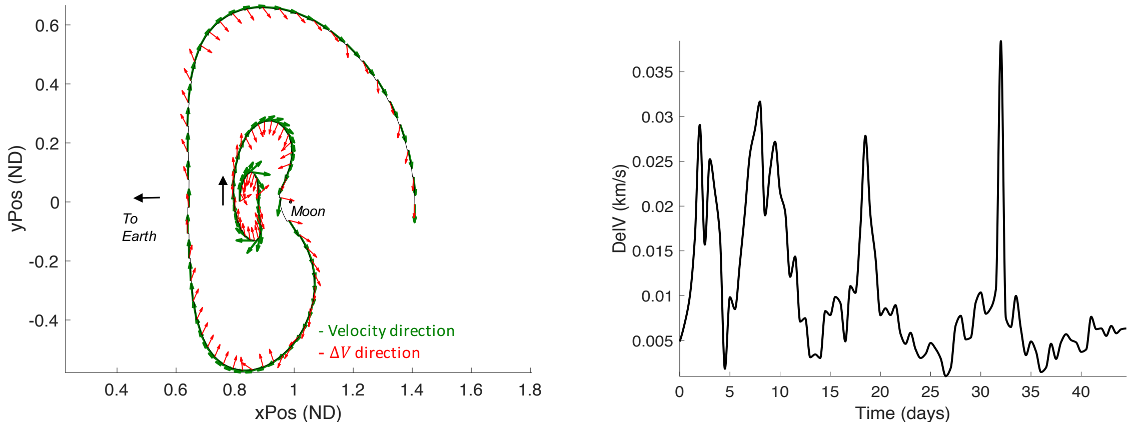
Natural Condition Transport Approach — *A Priori* Discretized Database		
s/c Specifications		
$a_0 = 9.7 \times 10^{-4}\ m/s^2$	$\widetilde{\mathbb{T}} = 0.22\ N$	$\widetilde{Isp} = 4000\ s$

The delivery of transfers with varied performance is achieved by adjusting the weights in Eqn. (5.2) and thus, meeting different goals via the objective function. In Fig. 6.3(a1) the TOF is prioritized with $W_p = 1e - 5$ in Eqn. (5.2), and Fig. 6.3(a2) is constructed with the priority on payload delivered by shifting the weight such that

$W_p = 1e5$. The scale of these weights are design variables and are modified based on the user's problem formulation. With greater emphasis placed on conserving propellant mass, the trajectory from Table 6.3 as illustrated in Fig. 6.3(a2 & b2) is constructed with 104 intermediate segments and has a longer TOF, but also offers reduced propellant usage when compared to the time-priority scenario in Fig. 6.3(a1 & b1) with 15 less intermediate segments. Although the transfer profiles are different in both cases to address the global objectives (e.g., prioritizing propellant conservation vs. lowering transfer duration), the algorithm deems the x-axis crossing a beneficial location for transfers to the DRO family (Fig. 6.3(b)). A flyby of the Moon rapidly increases the energy to slide through larger members of the Lyapunov family, rather than spiraling through additional revolutions to gradually increase the energy to that of the target orbit. Note that conditions within ARs that violate or lead to a violation during propagation (e.g., minimum altitude constraint), do not qualify for selection by an agent to construct the transfer pathway. Both solutions spiral out from the departure condition, which is a predictable geometry given that only the L_1 Lyapunov and DRO family structures are incorporated within the searchable database.

6.3 Demonstrate Transition of Initial Guess to Continuous Trajectory

Any discrete natural arcs deform under representative engine parameters introduced in an updated dynamical model. Recall that one objective of the current analysis is an assessment of the framework across varied engine platforms (VSI, CSI, chemical). So, the transfer path constructed in Fig. 6.3(a1 & b1) is adopted as an example to illustrate the process of transitioning a discontinuous initial guess to a continuous solution via numerical corrections. In Fig. 6.4(a) the velocity and ΔV_n directions undertaken by an agent during the search process to chain together a transfer sequence for the time-priority scenario in Fig. 6.3(a1 & b1) are superimposed. The maneuver directions are approximated by computing the difference in velocity directions between the state at the end of a natural arc segment and the one initiating the next segment in the transfer sequence. This approach suffices for the transition of discontinuous initial guesses to continuous solutions for the sample scenarios. The imparted ΔV history along the trajectory appears in Fig. 6.4(b). The smooth profile is achieved by interpolating between the ΔV_n values associated with the discrete segments. The variability in the magnitudes in Fig. 6.4(b) demonstrates that the framework allows initial conditions to be selected from varying radii within a given AR bound as required, to satisfy the global objectives.



(a) Maneuver direction history

(b) Maneuver magnitude history

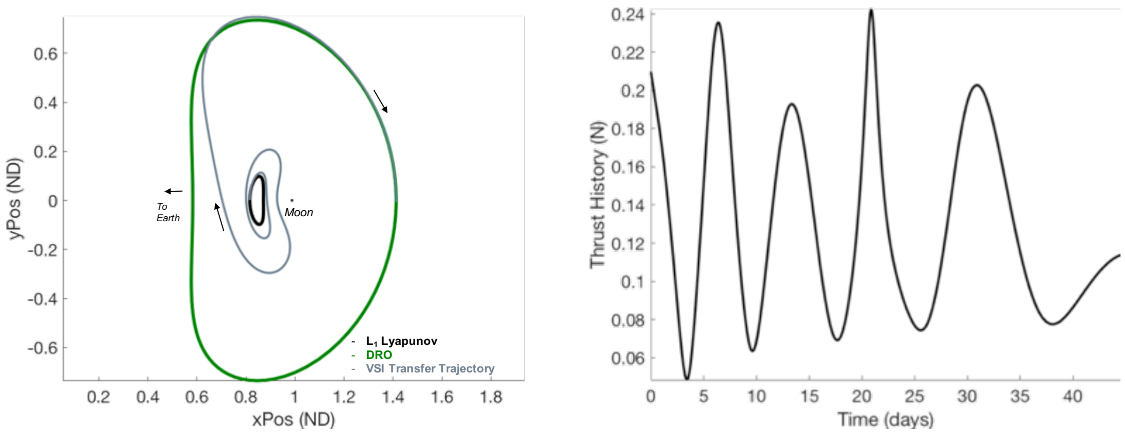
Figure 6.4: Information extracted from lower-fidelity transfer scenario in Fig. 6.3 (a1 & b1) to support transition to low-thrust model. (a) Maneuver direction history, (b) Maneuver magnitude history.

6.3.1 VSI Model Transfer

The position, velocity, time history and thrust magnitude and direction constitute sufficient information to initiate a numerical targeting process in a higher fidelity regime. Recall that a high low-thrust s/c acceleration value ($a \approx 9.7 \times 10^{-4}\ m/s^2$) in Fig. 6.3 (a1 & b1) supports the design choices to effect a lower computation time when employing Dijkstra's algorithm. However, while transitioning the IG to a continuous solution via the VSI regime, the s/c acceleration is maintained at lower values that are comparable to flight-proven low-thrust missions such as Deep Space 1 (DS1).

Figure 6.5(a) illustrates the path resulting from converging the initial guess in Fig. 6.3(a1 & b1) using a VSI engine. Position and velocity continuity are enforced throughout. For an arbitrarily selected 1000 kg spacecraft, the modulating thrust history illustrated in Fig. 6.5(b) conveys that the s/c acceleration magnitude varies along the VSI transfer path, but does not exceed $\approx 2.5 \times 10^{-4}\ m/s^2$. Thus, this example also serves to demonstrate that an initial guess sequence formulated using a particular s/c acceleration level can in some circumstances, be converged in the low-thrust regime with a weaker acceleration magnitude as well.

A lowered s/c acceleration capability and the incorporation of different s/c engine models and assumptions during numerical corrections contribute to a variation in the VSI transfer profile from the initial guess. During the convergence process, allowing the departure and arrival locations to vary along their respective orbits also allows the geometry to deviate from the original guess. The regions along the thrust profile (Fig. 6.5(b)) with very low thrust magnitudes are induced from high *Isp* values (Eqn. (2.50)) and are analogous to coasting regions in a CSI model. No specific distinction between thrust and coast locations are indicated in Fig. 6.5(a) because the modulation of *Isp* values in a VSI model leads to the engine always being on and thrusting, even when the throughput is negligible.



(a) VSI transfer

(b) VSI Thrust profile

Figure 6.5: Converged VSI solution from lower-fidelity transfer scenario in Fig. 6.3 (a1 & b1).

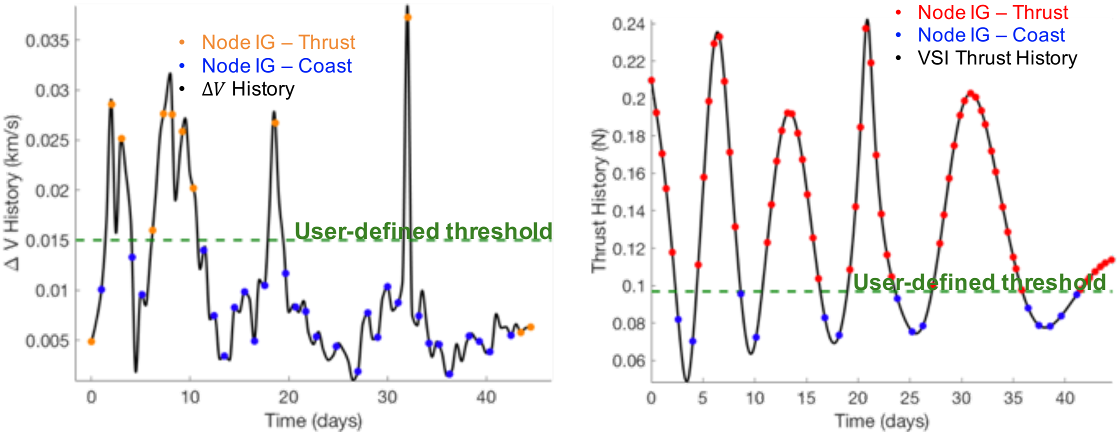
Table 6.4: Performance metrics for VSI transfer in Fig. 6.5

Scenario	TOF (days)	$\frac{m_f}{m_0}$ (%)
Fig. 6.5	44.50	99.14
s/c Specifications		
$\widetilde{\mathbb{P}} \approx 5.0\ kW$	$\widetilde{\mathbb{T}} = Variable$	$\widetilde{Isp} = Variable$

6.3.2 CSI Model Transfer

Transition of the initial guess to a CSI model is enabled via either the lower-fidelity raw IG or the converged VSI solutions. The time and state (position, velocity, mass, thrust/coast status and thrust direction) information for seeding the node conditions in the CSI numerical corrections process are extracted from knowledge of these conditions along the transfer paths in Fig. 6.4 (a) or Fig. 6.5 (a). The multiple relatively low ΔV/thrust magnitude regions in the initial guess (Fig. 6.4(b), Fig. 6.5 (b)) result in multiple coasting arcs for the original s/c possessing the higher acceleration magnitude from Fig. 6.3(a1 & b1). These same arcs facilitate the additional thrusting time required by a s/c with weaker acceleration levels ($a_0 = 2.2 \times 10^{-4}\ m/s^2$) to execute the transfer. Therefore, there exists a lower bound to the s/c acceleration levels for which a particular lower-fidelity solution can serve as a feasible initial guess. The thrust magnitude and specific impulse are fixed at the following values for the CSI engine in this example: $\widetilde{\mathbb{T}} = 0.22\ N$, $\widetilde{Isp} = 4000\ s$.

The convergence process is initiated by allocating the nodes with a thrust or coast status based on the magnitude of the ΔV (if exploiting the raw initial guess as a seed) or thrust magnitude (if exploiting the VSI solution as a seed); the status for a particular node depends on the metric value above or below a user-defined threshold. Such a user-defined threshold is indicated by the green line in Fig. 6.6 (a and b). Note that the first and last propagated nodes are always initialized with active thrust to bridge the energy gaps required to depart from and arrive at the desired conditions, even if the associated ΔV or thrust magnitudes fall below the user-defined threshold (as in Fig. 6.6). The placement of the seed conditions are merely a guide to initiate the convergence phase — the propagation duration (ϖ_i) between each node is a design variable (Eqn. (3.106)) , enabling the nodes to shift during the numerical corrections process to obtain a continuous CSI solution.

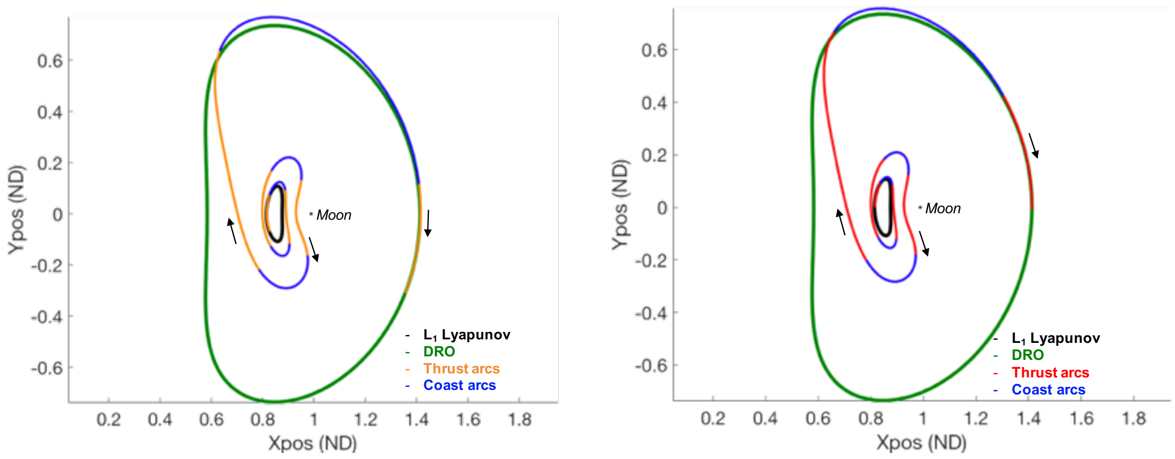


(a) Seeds along raw IG

(b) Seeds along VSI transfer

Figure 6.6: Indicating sample bounds for the user-defined thresholds on (a) the ΔV profile for the raw initial guess, and (b) thrust profile for the VSI solution. Also illustrated are the seeds for the thrust and coast nodes along the transfer profiles to transition the initial guesses into the CSI final continuous solution.

An approximately 1 *day* separation between the nodes on the raw transfer profile and approximately 0.5 *day* separation between those on the VSI transfer profile, along with the illustrated user-defined thresholds in Fig. 6.6 are exploited as seed conditions to settle on final solutions that are similar in geometry and performance metrics for the CSI engine (Fig. 6.7). Different node placement choices are required in both scenarios to arrive at a similar final solution in order to address the inherent differences in the underlying transfer characteristics of the raw and VSI solutions. However, both these avenues adequately serve in the transition of an initial guess to the CSI continuous solution. The Thrust Duration is abbreviated to TD in the *Performance metrics* tables for all the examples that follow. Furthermore, given the choice of either the raw or VSI transfer profiles to obtain the CSI solution, the



(a) From Raw IG

(b) From VSI Transfer

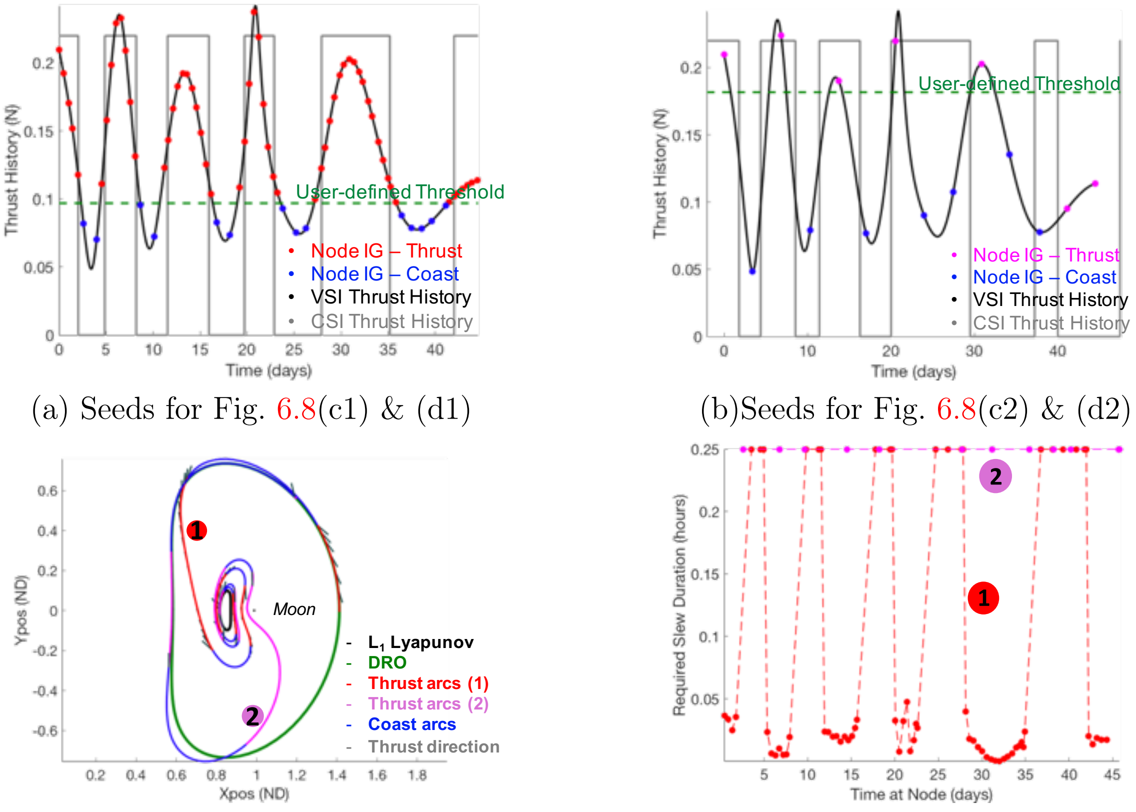
Figure 6.7: Optimized CSI transfer solutions for user-defined thresholds and node seed conditions obtained from (a): Raw IG and (b): VSI solution, in Fig. 6.6.

Table 6.5: Performance metrics for CSI transfers in Fig. 6.7 constructed from raw IG and VSI transfer solutions.

Scenario	TOF (days)	TD (days)	$\frac{m_f}{m_0}$ (%)	ΔV_{equiv} (m/s)
Fig. 6.7(a): From raw IG	48.20	22.35	98.92	427.24
Fig. 6.7(b): From VSI transfer	44.57	22.78	98.90	435.37

s/c Specifications		
$a_0 = 2.2 \times 10^{-4}\ m/s^2$	$\widetilde{\mathbb{T}} = 0.22\ N$	$\widetilde{Isp} = 4000\ s$

height of the user-defined thresholds to seed the thrust/coast arcs and the spacing between the nodes can be further modulated as a design lever, engendering different final CSI transfer profiles. The VSI solution computed in Fig. 6.5 is adopted as an example to construct the corresponding CSI solutions as shown in Fig. 6.8(c). The bang-bang CSI thrust profiles associated with these solutions are superimposed over the VSI thrust histories in Fig. 6.8(a, b). In general, the on/off switches for the CSI engine align with the peaks and troughs of the VSI solutions respectively, more closely in Fig. 6.8(a) than in Fig. 6.8(b) as the conditions for the thrust/coast behavior of the VSI transfer is more closely replicated via the seed conditions in Fig. 6.8(a). As a result, the final solution in Fig. 6.8(c1) resulting from the seed conditions in Fig. 6.8(a) follows the VSI baseline solution more closely. In contrast, the ≈ 3 *day* placement of the nodes in Fig. 6.8(b) allows for greater variation in the final solution in Fig. 6.8(c2) relative to the baseline. Thus, the combination of the user-defined threshold and node placement strategies on a specified initial guess are exploited to deliver a variety of CSI geometries and performance metrics.



(a) Seeds for Fig. 6.8(c1) & (d1)

(b)Seeds for Fig. 6.8(c2) & (d2)

(c) CSI transfers

(d) Turn rate histories

Figure 6.8: (a & b): User-defined thrust threshold selection for seeding initial guesses for CSI corrections process; (c): Converged CSI transfer solutions based on selected thresholds in (a & b); (d): Required s/c slew durations between nodes for solutions in (c).

During convergence, a point-and-hold strategy for the thrust-pointing directions (i.e. constant thrust direction between nodes) support the *direct* targeting process for

Table 6.6: Performance metrics for CSI transfers Fig. 6.8 constructed from different thrust node seeds on the same VSI transfer.

<table>
<tr><th>Scenario</th><th>TOF (days)</th><th>TD (days)</th><th>$\frac{m_f}{m_0}$ (%)</th><th>ΔV_{equiv} (m/s)</th></tr>
<tr><td>Fig. 6.8(c1)</td><td>44.57</td><td>22.78</td><td>98.90</td><td>435.37</td></tr>
<tr><td>Fig. 6.8(c2)</td><td>49.63</td><td>25.23</td><td>98.78</td><td>482.56</td></tr>
<tr><td colspan="5">User-Threshold for Thrust Arc Seeds for Fig. 6.8(c1): $\approx 0.1\ N$</td></tr>
<tr><td colspan="5">User-Threshold for Thrust Arc Seeds for Fig. 6.8(c2): $\approx 0.18\ N$</td></tr>
<tr><td colspan="5">s/c Specifications</td></tr>
<tr><td colspan="5">$a_0 = 2.2 \times 10^{-4}\ m/s^2$ $\widetilde{\mathbb{T}} = 0.22\ N$ $\widetilde{Isp} = 4000\ s$</td></tr>
</table>

the CSI model. Also, as mentioned earlier, the duration between these nodes is free to change once the convergence process is underway. Assuming a capability of 0.1 degree/sec s/c turn rate [130], Fig. 6.8(d) illustrates that both solutions only require a maximum of 0.25 hours to complete the thrust pointing requirements associated with the final CSI solutions. The minimum duration between any two nodes along the final solution in Fig. 6.8(c1) is 0.24 days, and 0.12 days for those in Fig. 6.8(c2). So, neither solution violates the turn-rate requirement in this example. The thrust pointing directions are also superimposed in grey along the trajectories in Fig. 6.8(c) to aid with qualitative visualization of the thrust history. The aim of this investigation is a technique to uncover initial guesses to seed the convergence process. Once a continuous solution is computed, then further constraints related to geometry and performance, such as turn-rate limitations, are incorporated. So, although such topics are explored in the transfer scenario in Fig. 6.8, they are not investigated in detail in subsequent examples. Generation of the initial guesses and converged continuous pathways via the design framework are aimed at providing a sufficient starting point for potential transfer solutions between orbits of interest, which are investigated and modified further if required — mission specific finer-tuning efforts are beyond the scope of this analysis.

As discussed in Section 4.2, Dijkstra's method is effective in constructing an end-to-end transfer if one exists, but at the cost of increased time and space complexity.

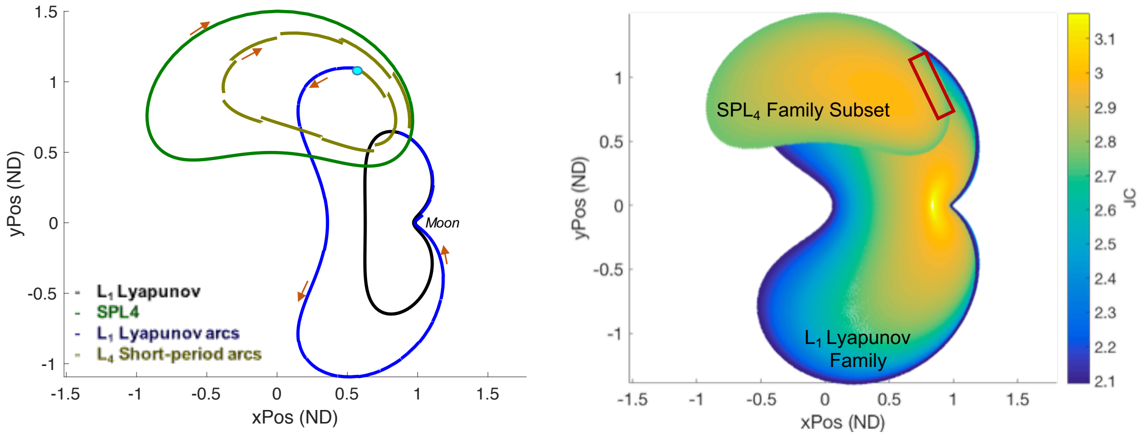
The HARL implementation is a stochastic process and may not always result in delivering the globally optimal solution, but varied performance outcomes are achieved by varying engine burn durations, the weights on the associated heuristic and cost functions and extracting different continuous final solutions. Thus, given its flexibility, the HARL algorithm is employed to construct the remaining examples in this investigation. This preliminary analysis demonstrates the potential of machine learning methods to mitigate the challenges associated with manual searches through a large or infinite trade space in a complex dynamical regime.

6.4 Exploit Intermediate Bridge Families to Accomplish Mission Goals

Establishing a sustainable cis-lunar transport infrastructure may require links between orbital ports in varied locations within the Earth-Moon system. Section 5.2.2 identifies examples of poor connectivity between various regions in space and the associated transfer design challenges as well. The following examples explore avenues to mitigate such challenges.

6.4.1 Examples — Planar Realm

One traditional strategy often adopted to construct initial guesses that aid in the convergence process is incorporating forward and reverse-time propagations from a selected condition or hyperplane towards the arrival and departure conditions respectively. The transfer scenario in Fig. 6.9(a) benefits from such a forward-reverse time automated search. Similar to the discussions associated with Figs. 5.5, 5.6 and 5.7, the L_1 Lyapunov (departure) and L_4 short-period (arrival) families possess little geometric overlap, where such overlaps typically assist in an end-to-end solely forward-time search routine. The red box in Fig. 6.9(b) illustrates the small position overlap region of interest between the L_1 Lyapunov family and a subset of the SPL4 family where the flow direction is also similar. Such a narrow overlap region renders a challenge for solely forward-time propagation to identify connections between the departure and destination orbits of interest in Fig. 6.9(a).



(a) Fwd-Rev propagations

(b) Overlap in families

Figure 6.9: (a) Demonstrating forward and reverse time searches to connect orbits belonging to families with little geometric overlap. (b) Families of interest colored by JC values. Red box: a visual estimate of the area of relatively close geometric overlap between the two families of interest. Departure: L_1 Lyapunov orbit ($JC = 2.90$), Destination: L_4 short-period orbit ($JC = 2.75$).

In contrast, there is value in incorporating the Resonant 4:3 family in the search process, as it possesses geometric overlap between both the L_1 Lyapunov and L_4 short-period families (Fig. 6.10). Such an overlap broadens the potential transfer options by offering intermediate arcs from the resonant family to connect the flows between the departure and destination families in a forward-time search. The transfer from Fig. 6.10 is colored in terms of JC values in Fig. 6.11(a). It is evident from this figure that the search strategy does not always traverse a path with a monotonic increase or decrease in Jacobi constant values to transition between different energy levels. In fact, the intermediate arcs possess higher JC values than both the departure and destination orbits. The initial guess is converged to formulate a continuous and optimized solution in Fig. 6.11(b) that offers a similar geometry and performance metrics compared with the initial guess.

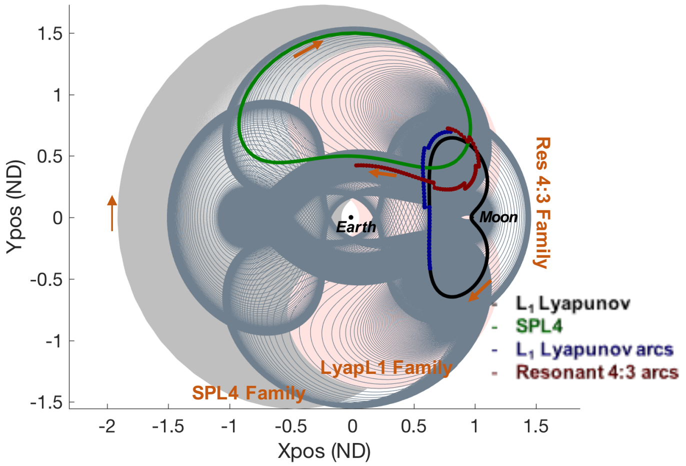
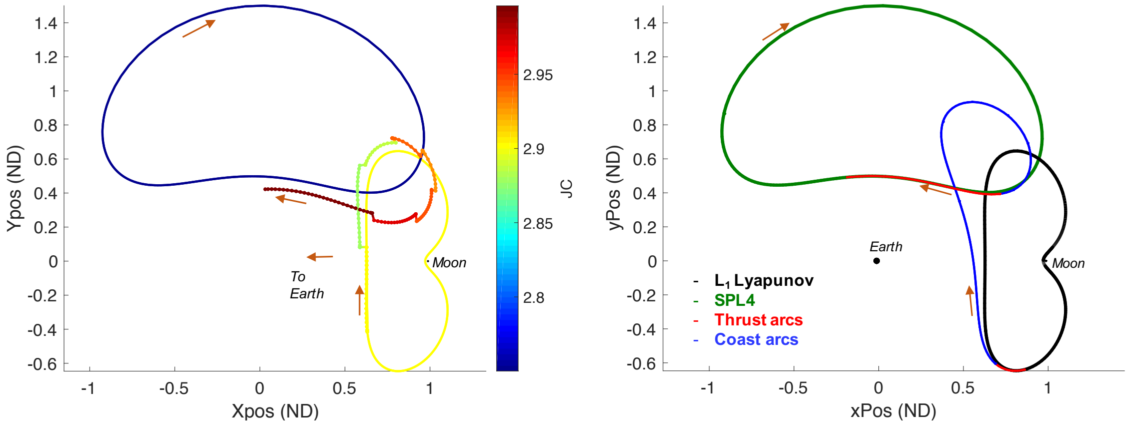


Figure 6.10: Demonstrating the ability to construct a purely forward-time end-to-end transfer initial guess by incorporating an intermediate-bridging family. Departure: L_1 Lyapunov orbit ($JC = 2.90$), Destination: L_4 short-period orbit ($JC = 2.75$), Bridging family: Resonant 4:3 family



(a) IG colored by JC

(b) Mass optimal

Figure 6.11: (a) Energy transitions and (b) mass optimal low-thrust solution. Departure: L_1 Lyapunov orbit ($JC = 2.90$), Destination: L_4 short-period orbit ($JC = 2.75$)

Expanding the range of design scenarios is insightful. The examples in Fig. 6.12 demonstrate the benefits of incorporating bridging families during the transfer design. The example in Fig. 6.12(a) illustrates that the framework is not restricted to pathways between departure and arrival conditions that are at different energy levels. This transfer scenario also demonstrates that bridging families are not restricted to

Table 6.7: Performance metrics for Fig. 6.11 illustrating the outcomes from incorporating intermediate bridge families.

Scenario	TOF (days)	TD (days)	$\frac{m_f}{m_0}$ (%)	ΔV_{equiv} (m/s)
Fig. 6.11(a): Initial guess:	24.00	-	99.3	-
Fig. 6.11(b): Mass optimal:	28.82	6.80	99.67	129.39

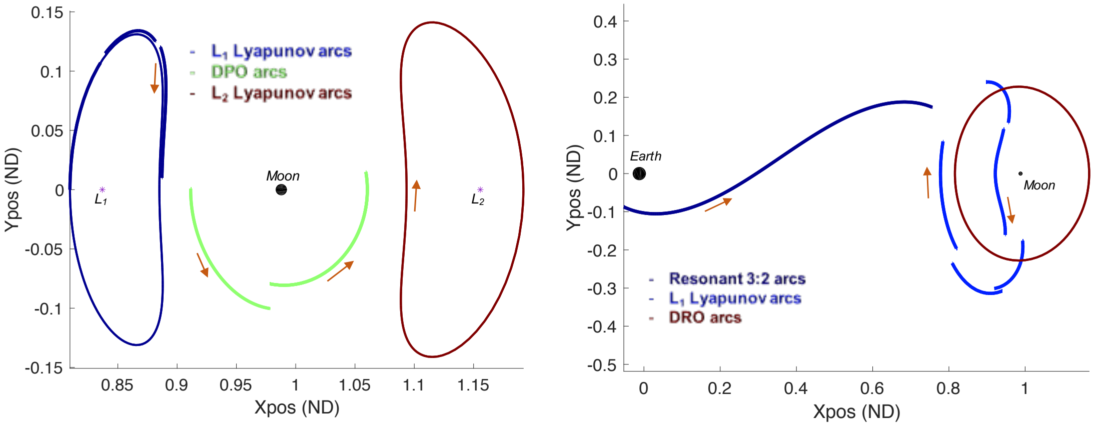
Natural Condition Transport Approach — *A Priori* Discretized Database		
s/c Specifications		
$a_0 = 2.2 \times 10^{-4}\ m/s^2$	$\widetilde{\mathbb{T}} = 0.22\ N$	$\widetilde{Isp} = 4000\ s$

resonant families, and that other periodic orbit families can also serve in this capacity — the bridging family in Fig. 6.12(a) is the Distant Prograde Family (DPO). Furthermore, Fig. 6.12(b) delivers a transfer from an arbitrary condition in the Earth's vicinity to a DRO. The ability to initiate a search sequence from such an arbitrary condition, in addition to known periodic orbit motion (prior examples) demonstrates the ability of the framework to support the exploration for contingency options related to uncertainties in the deployment states (e.g., for secondary payloads) or recovering from launch delays which would alter the Earth departure conditions.

Table 6.8: Performance metrics for Fig. 6.12 illustrating the outcomes from incorporating intermediate bridge families.

Scenario	TOF (days)	$\frac{m_f}{m_0}$ (%)
Fig. 6.12(a): LPO-LPO transfer	10	99.8
Fig. 6.12(b): Earth-lunar vicinity transfer	18	99.6

Natural Condition Transport Approach — *A Priori* Discretized Database		
s/c Specifications		
$a_0 = 2.2 \times 10^{-4}\ m/s^2$	$\widetilde{\mathbb{T}} = 0.22\ N$	$\widetilde{Isp} = 4000\ s$

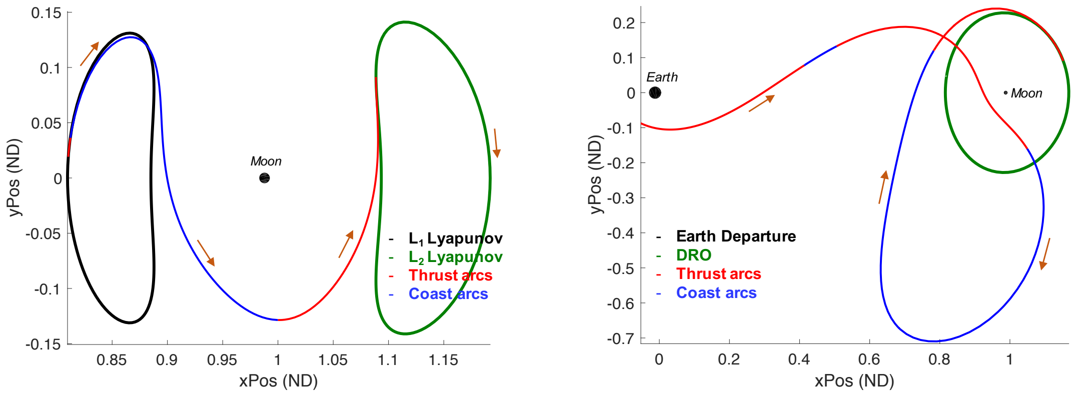


(a) LPO to LPO

(b) Earth to lunar vicinity

Figure 6.12: Varied scenarios to demonstrate importance of intermediate bridging arcs.
(a) Libration Point Orbit - Libration Point Orbit (LPO) transfer. Departure: L_1 Lyapunov orbit ($JC = 3.12$), Destination: L_2 Lyapunov orbit ($JC = 3.12$).
(b) Departure: Earth altitude of $\approx 35,000 km$, Destination: DRO ($JC = 2.94$)

The optimized and continuous transfers are constructed and appear in Fig. 6.13. Discrepancies in the performance measures between the initial guess and optimized solutions can arise from constructing a mass-optimal final solution in contrast to the multi-objective IG generation process and are also due to the design choices associated with the thrust/coast seeds and node placements. Exploiting the natural dynamics (coast arcs) can prove beneficial for propellant savings. So, if there is a need to build additional coasting capability directly into the IG generation process, the user is also able to investigate alteration of the coefficients in Eqns. (5.2) and (5.4) to place greater emphasis on propellant conservation (e.g., Fig. 6.3). The results, however, illustrate the ability of the HARL algorithm to construct initial guesses that are suitable for transition to the continuous model.



(a) LPO to LPO

(b) Earth to lunar vicinity

Figure 6.13: Optimized CSI solutions for initial guesses in Fig. 6.12.

Table 6.9: Performance metrics for CSI transfers in Fig. 6.13.

Scenario	TOF (days)	TD (days)	$\frac{m_f}{m_0}$ (%)	ΔV_{equiv} (m/s)
Fig. 6.13(a): LPO-LPO transfer	14	5.09	99.80	96.83
Fig. 6.13(b): Earth-lunar vicinity transfer	29	10.97	99.50	209.07

s/c Specifications		
$a_0 = 2.2 \times 10^{-4}\ m/s^2$	$\widetilde{\mathbb{T}} = 0.22\ N$	$\widetilde{Isp} = 4000\ s$

6.4.2 Examples — Spatial Realm

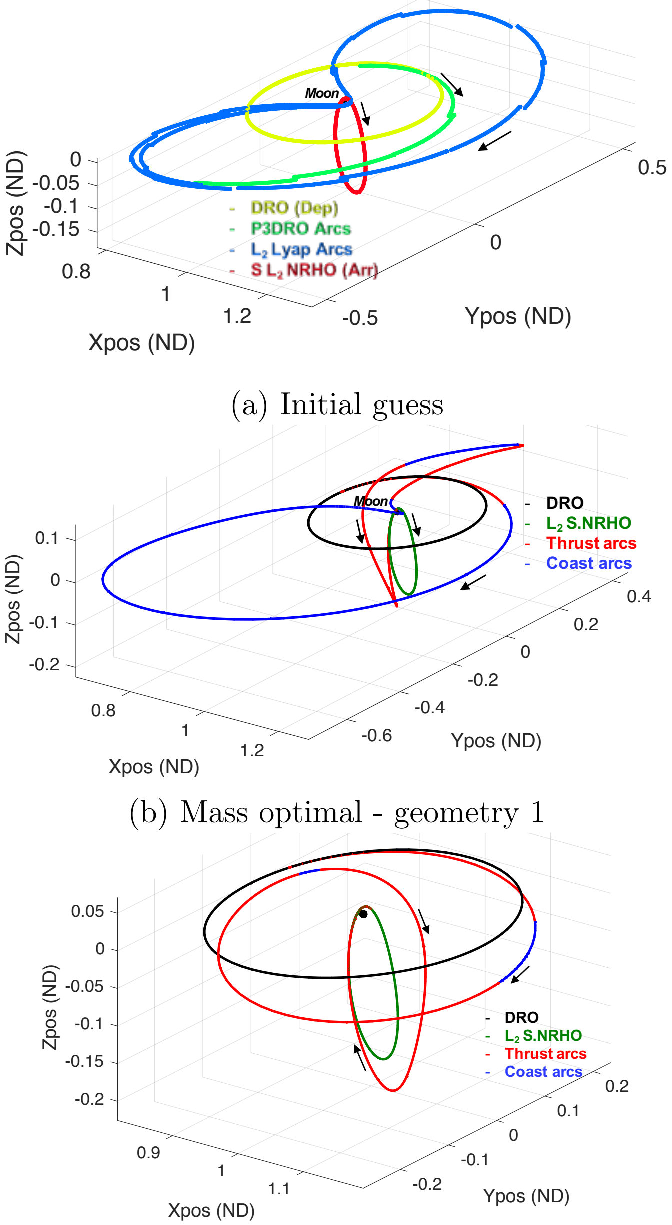
Transfers between a planar DRO and spatial NRHO orbit appears in Fig. 6.14 to illustrate other types of mission scenarios. In particular, this example demonstrates the potential for planar arcs to serve as beneficial intermediate bridging conditions even for spatial transfers. Once the initial guess is uncovered by the HARL algorithm, the segments in Fig. 6.14(a) are not constrained to remain within the plane during the numerical corrections process. So, the arcs shift freely in space to satisfy the state and time continuity constraints. The broad characteristics from the initial guess, e.g., the transition into an L_2 Lyapunov-like motion and the lunar flybys with a subsequent entry into the NRHO, are preserved in Fig. 6.14(b). As observed in the planar case

(Fig. 6.8), the framework exploits the same initial guess to introduce varied thrust profiles based on user-defined thrust threshold values to extract multiple local optima as apparent in Fig. 6.14(b & c). It is quite useful to extract solution variations to facilitate trade-off analyses and enable diverse mission scenarios.

Table 6.10: Performance metrics for IG and CSI transfers in Fig. 6.14.

Scenario	TOF (days)	TD (days)	$\frac{m_f}{m_0}$ (%)	ΔV_{equiv} (m/s)
Fig. 6.14(a): Initial guess	50.00	-	98.77	-
Fig. 6.14(b): LT Geometry 1	49.72	23.50	98.86	449.19
Fig. 6.14(c): LT Geometry 2	24.19	22.29	98.92	426.95

Natural Condition Transport Approach — *A Priori* Discretized Database		
s/c Specifications		
$a_0 = 2.2 \times 10^{-4}\ m/s^2$	$\widetilde{\mathbb{T}} = 0.22\ N$	$\widetilde{Isp} = 4000\ s$



(a) Initial guess

(b) Mass optimal - geometry 1

(c) Mass optimal - geometry 2

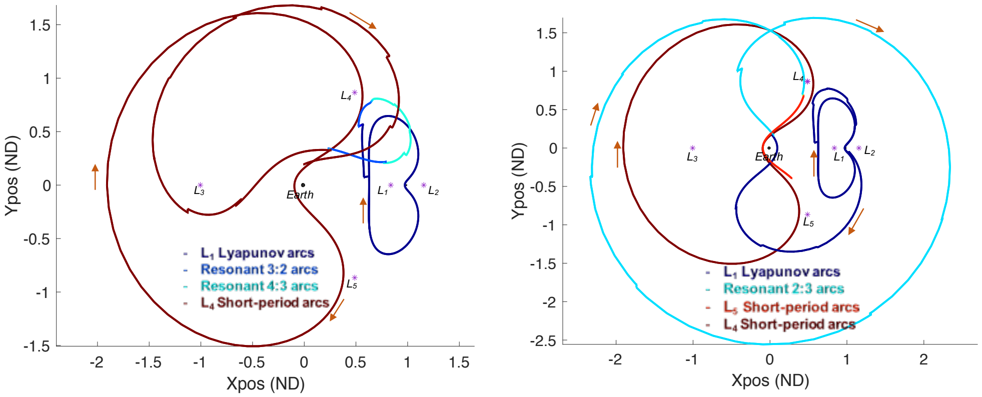
Figure 6.14: Demonstrating ability to transfer between planar and spatial families. Departure: ≈ 13 *day* DRO ($JC = 2.94$), Destination: ≈ 6.65 *day* Southern L_2 Halo orbit (NRHO, $JC = 3.05$).

6.5 Transfer Geometry Considerations

The examples thus far illustrate the advantage of incorporating *a priori* discretized states from known periodic orbits into the design framework to facilitate end-to-end initial guesses. A conscious choice to selectively include/exclude certain periodic orbit families from the search offer the opportunity to influence the transfer profile or restrict the s/c to certain regions in configuration space. However, it is also beneficial to explore the design space without *a priori* knowledge about the available natural conditions in the dynamical system via the free-form search. Such a capability renders the design framework flexible to be exploited during investigations within any dynamical regime. Alternatively, for operational purposes, it may be beneficial to maintain control over the qualitative and quantitative nature of the routes without restricting the pathfinding agents to waypoints within an *a priori* discretized catalog. So, the trained flow models are also exploited to permit admissible intermediate arcs during the search. A further option to influence the transfer profile via the management of the thrust profile is also examined.

6.5.1 Selective Inclusion of Bridge Families via Discretized Database

Variety in transfer geometry is managed by consciously including/excluding certain bridge families from the searchable conditions to influence the path traversed by the s/c. For example, the transfer scenarios in both Figs. 6.15(a) and 6.15(b) link the same departure (L_1 Lyapunov) and arrival (L_4 short-period) orbits. However, different bridging families results in varied final transfer profiles — Fig. 6.15(a) incorporates 3:2 and 4:3 Resonant arcs, whereas, Fig. 6.15(b) incorporates 2:3 Resonant and L_5 short-period arcs. Thus, by restricting the types of natural conditions that are available to be sampled within the discretized database, the designer gains control over the spacecraft traversable regions in configuration space to suit specific mission applications; e.g., visiting distant space assets en-route via an *exterior* transfer (Fig. 6.15(b)), or remaining in the vicinity of the primaries for communication-related constraints during an *interior* transfer (Fig. 6.15(a)). In this example, the interior transfer clearly offers improved time savings for a similar mass budget as well. The initial guess performance metrics are good estimates for the continuous solutions, and are readily transformed to the low-thrust solutions as displayed in Fig. 6.16. The corrections process shifts the geometry of the mass optimal transfer in Fig. 6.16(b) from its corresponding IG in Fig. 6.15(b), but the geometry is maintained more closely between the interior transfer IG and corresponding mass optimal solution.



(a) Interior transfer

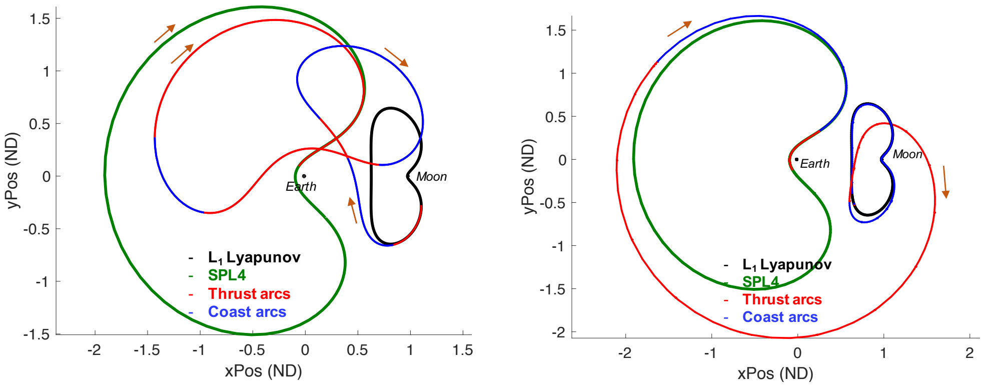
(b) Exterior transfer

Figure 6.15: Demonstrating influence over transfer geometry. Departure: L_1 Lyapunov orbit ($JC = 2.90$), Destination: L_4 short-period orbit ($JC = 1.80$).

Table 6.11: Performance metrics for Fig. 6.15 illustrating ability to influence transfer geometry via intermediate transfer arcs.

Scenario	TOF (days)	$\frac{m_f}{m_0}$ (%)
Fig. 6.15(a): Interior transfer	51.00	98.6
Fig. 6.15(b): Exterior transfer	75.00	98.1

<table>
<tr><td colspan="3">Natural Condition Transport Approach — <i>A Priori</i> Discretized Database</td></tr>
<tr><td colspan="3">s/c Specifications</td></tr>
<tr><td>$a_0 = 2.2 \times 10^{-4}\ m/s^2$</td><td>$\widetilde{\mathbb{T}} = 0.22\ N$</td><td>$\widetilde{Isp} = 4000\ s$</td></tr>
</table>



(a) Interior transfer

(b) Exterior transfer

Figure 6.16: Mass optimal CSI solutions for initial guesses in Fig. 6.15.

Table 6.12: Performance metrics for CSI transfers in Fig. 6.16.

Scenario	TOF (days)	TD (days)	$\frac{m_f}{m_0}$ (%)	ΔV_{equiv} (m/s)
Fig. 6.16(a): Interior transfer	56.68	26.87	98.58	514.79
Fig. 6.16(b): Interior transfer	74.15	32.58	98.42	624.21

s/c Specifications		
$a_0 = 2.2 \times 10^{-4}\ m/s^2$	$\widetilde{\mathbb{T}} = 0.22\ N$	$\widetilde{Isp} = 4000\ s$

6.5.2 Exploiting Free-Form Search — Planar Realm

A free-form search where pathfinding agents are free to select from either chaotic or ordered motion within an AR (Section 5.2.1) offers a variety of transfer geometries, permitting greater intuition into the flows within the system. The search for connections in a chaotic regime results in very diverse transfer options for any s/c type, especially due to the existence of manifold structures that are a natural transport mechanism in a multi-body regime. Figure 6.17 highlights an L_1 Lyapunov departure orbit and an arrival DRO and illustrates the direction and span of natural flows that can be expected to depart the originating Lyapunov orbit. Accordingly, the free-form search for connections from the Lyapunov orbit to the DRO illustrated in Fig. 6.18

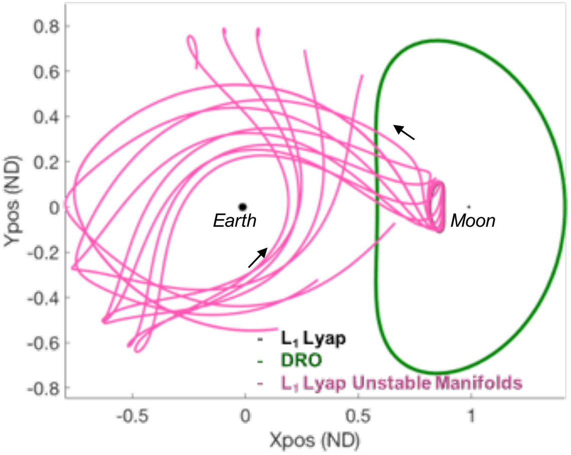
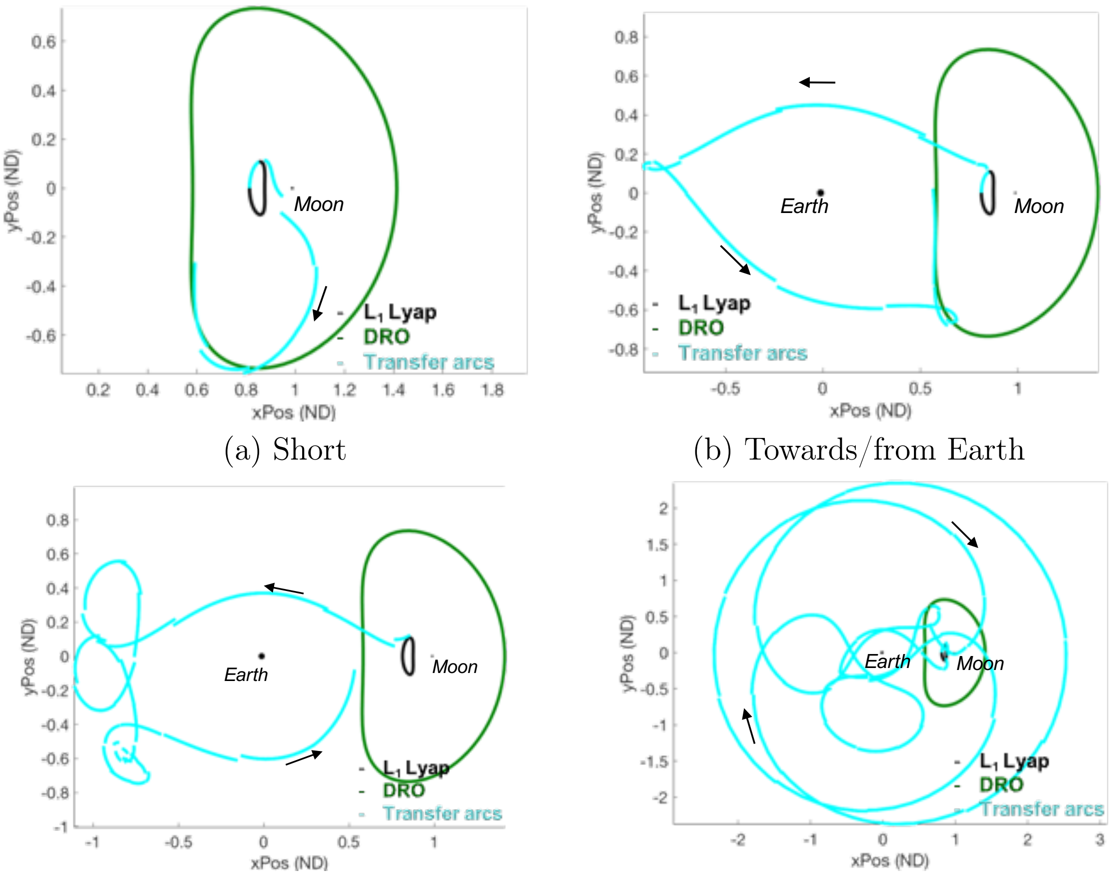


Figure 6.17: Unstable manifolds of L_1 Lyapunov departure orbit propagated for multiple days for illustration purposes only.

are not restricted to the vicinity of these orbits. Many geometries that flow towards the Earth and return to vicinity of the destination orbit are influenced by the natural flows that emanate from the L_1 Lyapunov orbit. Geometries that extend through the exterior of the system prior to reaching the destination are also uncovered via the free-form search.

Table 6.13: Performance metrics for Fig. 6.18 illustrating ability to compute varied transfer profiles via free-form search in the planar realm.

Scenario	TOF_{est} (days)	$\frac{m_f}{m_0}_{est}$ (%)
Fig. 6.18(a): IG short	24	99.52
Fig. 6.18(b): IG towards/from Earth	45	98.89
Fig. 6.18(c): IG towards/from Earth multi-loop	108	97.74
Fig. 6.18(d): IG chaotic	147	96.43
Natural Condition Transport Approach — Free-Form Search.		
s/c Specifications		
$a_0 = 2.2 \times 10^{-4}\ m/s^2$	$\widetilde{\mathbb{T}} = 0.22\ N$	$\widetilde{Isp} = 4000\ s$



(a) Short

(b) Towards/from Earth

(c) Towards/from Earth multi-loop

(d) Chaotic

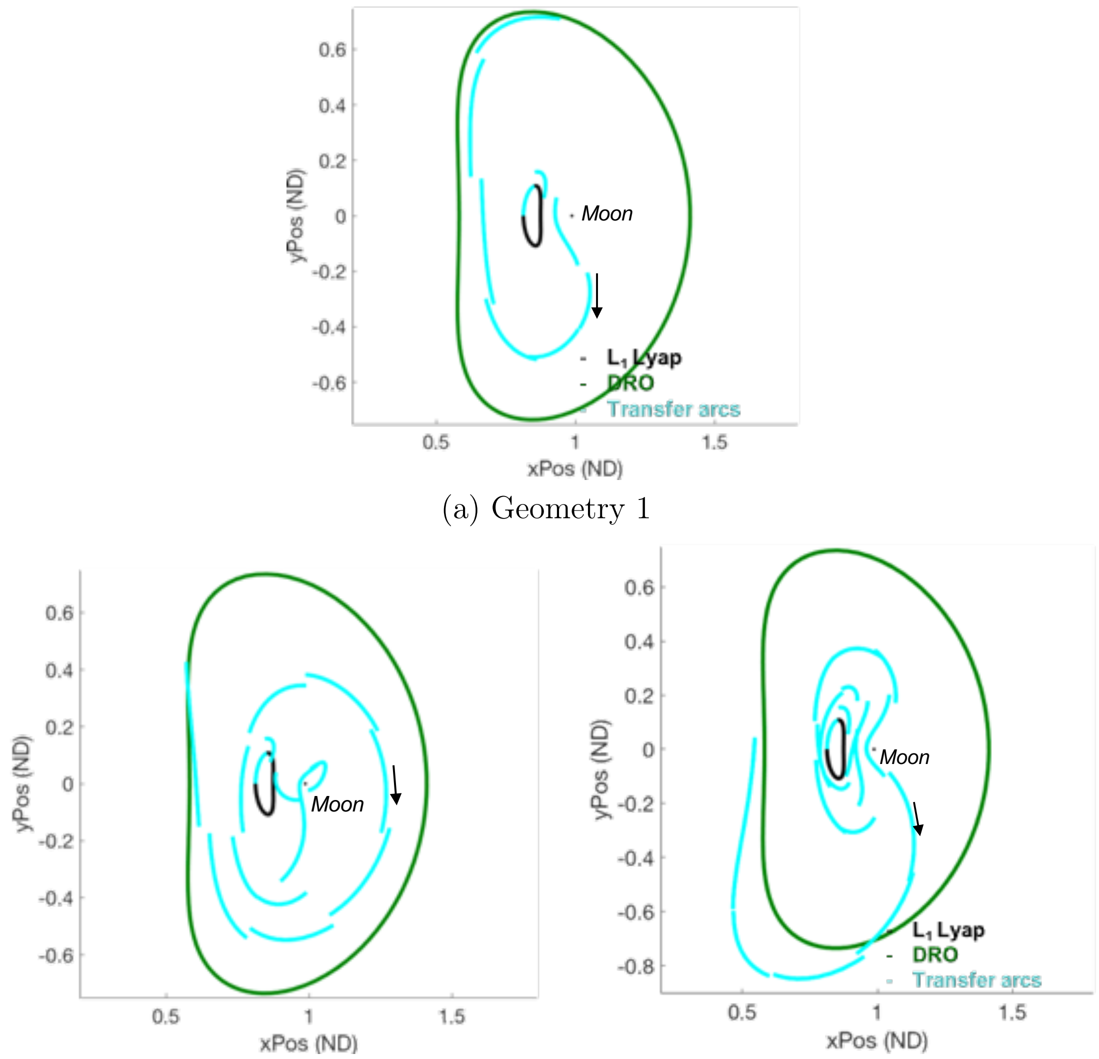
Figure 6.18: Free-form search to uncover varied transfer geometries between L_1 Lyapunov (JC: 3.14) and DRO (JC: 2.785) for initial approximations. Each IG is composed of 3 *day* natural arcs.

Each of the solutions in Fig. 6.18 offers contrasting potential to satisfy mission constraints and performance outcomes. For example, in addition to meeting mission geometry considerations, trades in the TOF and delivered mass fraction between the different solutions are also available to determine a viable option for the mission (Table 6.13). The performance metrics presented in Table 6.13 correspond to the top ranking solutions presented in Fig. 6.18 computed as a result of prioritizing TOF as a global objective. Through such varied transfer choices, the designer gains the ability to improve or expand intuition about the solution space for the given transfer scenario as well. For example, some structures in Fig. 6.18 are comparable to resonant orbit motion, that can be incorporated into the pathfinding search to conduct more detailed analysis if desired. Although the advantage of the free-form search is its potential to establish variety across solutions, it can be challenging to restrict the search to specific areas of the configuration space. Exploiting the regressed and classified flow-models are one option to address such a limitation without restricting the pathfinding to pre-discretized natural states.

6.5.3 Exploiting Natural Family Models — Planar Realm

Incorporation of natural family/flow models permits only those states within an AR to be selected by pathfinding agents if the states are recognized as belonging to a periodic orbit family of interest (Section 5.2.4). If, for example, the desirable connection profile between the L_1 and DRO orbits in Fig. 6.18 is consistent with motion closely following the flows in the families of the departure and destination orbits alone (much like the geometry identified in Fig. 6.18(a)), then only the L_1 Lyapunov and DRO family flow-models need to be incorporated to deliver sufficient options. So, by incorporating these families, different high-ranking examples are illustrated in Fig. 6.19; these solutions offer further insight to the transfer trade-space identified in Fig. 6.18. The delivered mass-fractions are comparable for similar TOFs in the free-form (Table. 6.13, Fig. 6.18 (a & b)) and flow-model (Table. 6.14, Fig. 6.19 (a & b)) results. Engine burns in the vicinity of a primary or close primary swing-bys give rise to large accessible regions and so allow large discontinuities between an arc emanating from such a burn near the Moon and a subsequent arc (e.g., Fig. 6.19(c)). Larger discontinuities lead to increased propellent consumption and, thus, a smaller delivered mass-fraction estimate for the scenario in Fig. 6.19(c) (Table. 6.14) compared with that for the longer transfer scenario in Fig. 6.18(c) (Table. 6.13) that does not possess burns near the primaries or close swing-bys of these bodies.

The different approaches to provide natural conditions within an AR are capable of rendering vastly different transfer geometries. Unlike in Fig. 6.18, the transfer



(a) Geometry 1

(b) Geometry 2

(c) Geometry 3

Figure 6.19: Restricting search space and nature of transfer arcs using trained flow models to transfer between L_1 Lyapunov (JC: 3.14) and DRO (JC: 2.785)

profiles in Fig. 6.19 are, in general, restricted to motion resembling either the L_1 Lyapunov or DRO families. However, there exists one short arc that encircles the Moon in an anti-clockwise direction in Fig. 6.19(b) that does not resemble motion in either the L_1 Lyapunov or DRO families. Recall that the regression and classification errors associated with flow models are the greatest in highly nonlinear regions near the primaries (Section 5.2.4). Thus, it is possible for states to be incorrectly recognized as

Table 6.14: Performance metrics for Fig. 6.19 illustrating ability to compute varied transfer profiles resembling motion associated with the incorporated flow-models in the planar realm.

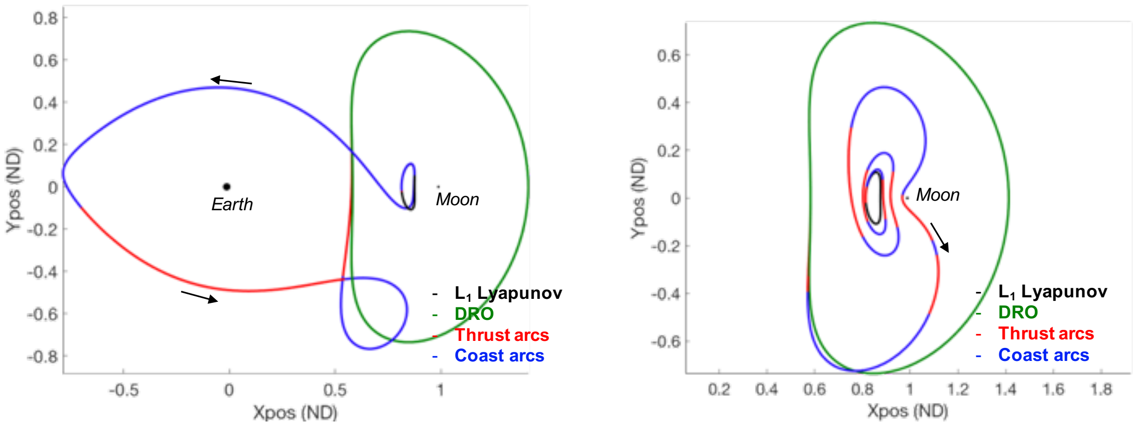
Scenario	TOF_{est} (days)	$\frac{m_f}{m_0}_{est}$ (%)
Fig. 6.19(a): Geometry 1	21	99.34
Fig. 6.19(b): Geometry 2	45	98.31
Fig. 6.19(c): Geometry 3	57	97.64

Natural Condition Transport Approach — Trained Flow Models.		
s/c Specifications		
$a_0 = 2.2 \times 10^{-4}\ m/s^2$	$\widetilde{\mathbb{T}} = 0.22\ N$	$\widetilde{Isp} = 4000\ s$

belonging to the L_1 Lyapunov or DRO families in this region. However, in general, the flow models deliver transfer arcs that exhibit L_1 Lyapunov-like or DRO-like behavior to connect the departure and arrival orbits in Fig. 6.19. An attempt to improve the prediction capabilities of the flow-models in the vicinity of the primaries is an area of open investigation beyond the scope of this preliminary study.

Transition of Initial Guesses to Low-Thrust Solutions

The free-form and flow-model based approaches (Figs. 6.18 and 6.19) offer varied end-to-end initial guesses for the user to make a selection from. The selection that best addresses mission requirements is then transitioned to a continuous solution via a numerical corrections process. For example, the mass-optimal low-thrust solutions for two selected initial guesses, the transfer scenarios in Fig. 6.18(b) and Fig. 6.19(c) are captured in Fig. 6.20(a & b), respectively. The mass-optimal solutions improve the delivered mass-fractions (Table 6.15) from the estimated values in Tables 6.13 and 6.14, and the TOF values in the initial guess tables are reasonable estimates for the final continuous solutions.



(a) (b)

Figure 6.20: Optimized low-thrust solutions constructed from initial guesses in Figs. 6.18 and 6.19.

Table 6.15: Performance metrics for CSI transfers in Fig. 6.20.

Scenario	TOF (days)	TD (days)	$\frac{m_f}{m_0}$ (%)	ΔV_{equiv} (m/s)
Fig. 6.20(a): From free-form search (Fig. 6.18(b))	52.32	11.67	99.43	222.45
Fig. 6.20(b): From flow-models (Fig. 6.19(c))	58.29	20.48	99.01	391.16

s/c Specifications		
$a_0 = 2.2 \times 10^{-4}\ m/s^2$	$\widetilde{\mathbb{T}} = 0.22\ N$	$\widetilde{Isp} = 4000\ s$

6.5.4 Exploiting Free-Form Search — Spatial Realm

The free-form and model-based pathfinding is extendable to the spatial realm as well. The search dimensions are, however, expanded when including the out-of-plane position and velocity components during the search process, thus, leading to potentially increased computational resource requirements to uncover transfer options. As such, rather than a blind search, one option is to propagate the manifolds of the departure and/or destination orbits to aid with seeding potential waypoints. As an example, the departure orbit in Fig. 6.21 (southern L_2 NRHO with a periapsis altitude of ≈ 1763 km) is stable and possesses no unstable manifolds. So, the stable manifolds of the destination halo orbit (periapsis altitude of $\approx 49,215$ km) are propagated in reverse-time to gain intuition into the transfer problem. It is evident from

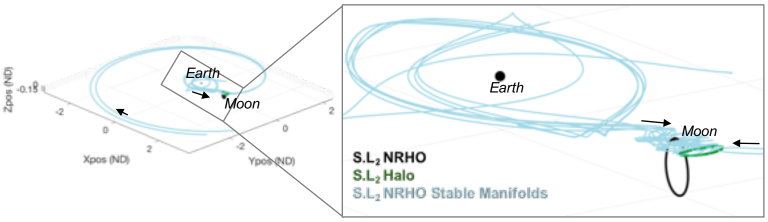


Figure 6.21: Stable manifolds approaching southern L_2 halo orbit

Fig. 6.21 that both interior and exterior flows toward the planar halo orbit exist. However, the manifolds for this orbit all remain relatively planar as well; no obvious connecting pathways from the NRHO to the destination orbit, especially in terms of the exterior flows, are exhibited. So, an alternative strategy employs the accessible region generation process in reverse-time. The states within ARs generated in reverse time from the arrival orbit should lead to this destination orbit in forward time. The states along the destination orbit are perturbed by the maximum δV possible for the selected engine burn-time and for the given s/c capabilities (as discussed in Section 5.1), and are propagated in reverse time for a duration to create an accessible region. The states within this first accessible region are then each similarly perturbed and propagated in reverse time for the specified time horizon; this process is repeated multiple times as deemed appropriate for the specific problem under consideration. Figure 6.22 illustrates the resulting *seed* states in configuration space. Although these waypoints follow a similar geometrical path to the stable manifolds as illustrated in Fig. 6.21, this latter approach produces a greater variety of spatial waypoints between the departure and destination orbits as well. The seed states, pre-configured with reinforcements (Fig. 6.22) are introduced to the reinforcement learning paradigm to construct transfers from the NRHO to the almost planar halo. Such reinforced states are selectable in the spatial transfer scenarios if they exist within a user-defined bound

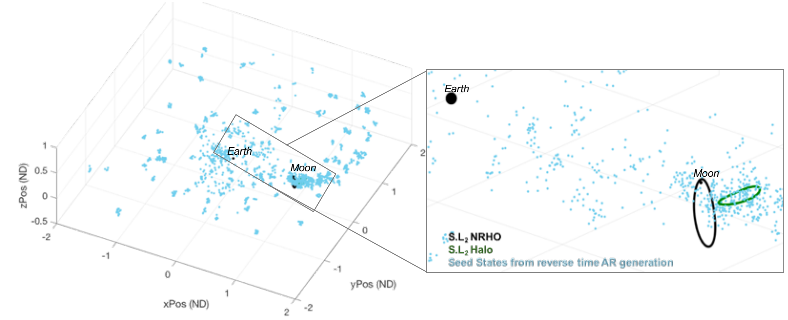
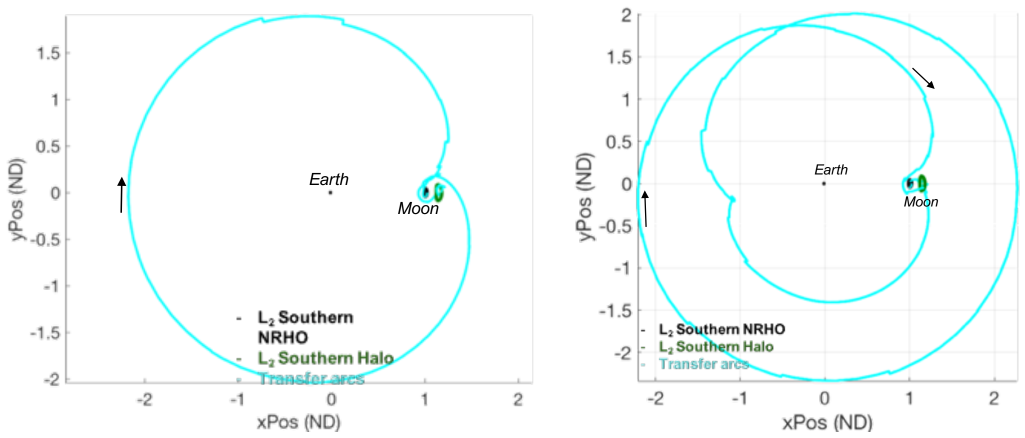


Figure 6.22: Seed states resulting from accessible region generation performed in reverse time from destination orbit (L_2 southern halo).

of a s/c instantaneous AR to aid in the pathfinding process within an infinitely large state-space. Examples of two transfer geometries from such a search are plotted in Fig. 6.23.



(a) Geometry 1

(b) Geometry 2

Figure 6.23: Free-form spatial transfer example geometries from southern L_2 NRHO (Periapsis Altitude $=$ 1763.31 km) to southern L_2 halo (Periapsis Altitude $=$ $49,215.45$ km).

Table 6.16: Performance metrics for Fig. 6.23 illustrating ability to compute varied transfer profiles via free-form search in the spatial realm.

Scenario	TOF_{est} (days)	$\frac{m_f}{m_0}_{est}$ (%)
Fig. 6.23(a): Geometry 1	75	95.39
Fig. 6.23(b): Geometry 2	129	92.92

Natural Condition Transport Approach — Free-Form Search.		
s/c Specifications		
$a_0 = 2.2 \times 10^{-4}\ m/s^2$	$\widetilde{\mathbb{T}} = 0.22\ N$	$\widetilde{Isp} = 4000\ s$

The results in this figure illustrate that the stochastic nature of the reinforcement learning paradigm ensures variety in the transfer geometries, even with the inclusion of *suggested* waypoints. Similar to the planar case discussion, the free-form is capable of highlighting natural flows in the system that are advantageous to construct the transfer scenario. Some structures in Fig. 6.23 are comparable to resonant orbit motion, that can be incorporated into the pathfinding search to conduct more detailed analysis if desired.

6.5.5 Exploiting Natural Family Models — Spatial Realm

The free-form search conveniently reveals many transfer geometry alternatives. However, it is challenging to control the region traversed by the s/c using this approach. So, one option to investigate interior transfers between the departure and destination orbits in Fig. 6.23 is to restrict the search to within the trained flow-model for the southern L_2 halo family. Such a transfer profile is constructed in Fig. 6.24. The interior delivery predicts a reduction in the total TOF by at least one month compared to the exterior geometries and also conserves propellant costs. Notwithstanding this individual result, the exploration of varied geometries for different mission applications via both free-form and flow-models is key during the initial design phase.

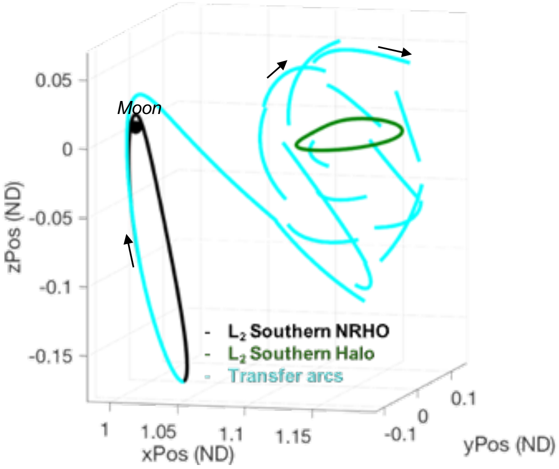


Figure 6.24: Trained flow-model enabled spatial transfer example from southern L_2 NRHO (Periapsis Altitude = 1763.31 km) to southern L_2 halo (Periapsis Altitude = $49,215.45$ km).

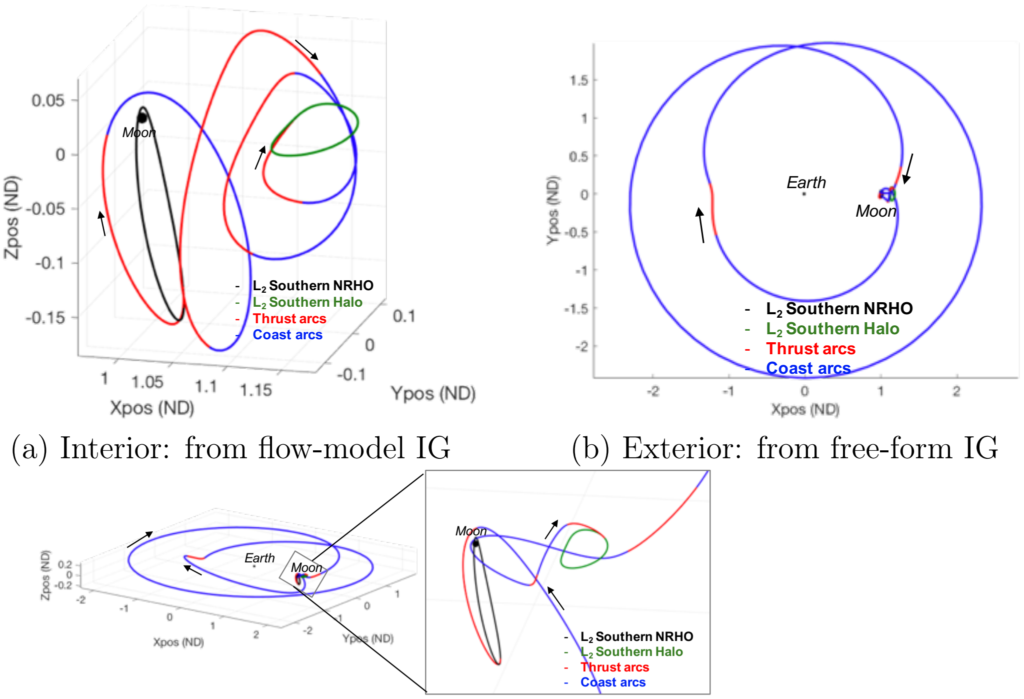
Table 6.17: Performance metrics for Fig. 6.24 illustrating the resemblance of the transfer arcs to motion associated with the incorporated halo family flow-model in the spatial realm.

Scenario	TOF_{est} (days)	$\frac{m_f}{m_0}_{est}$ (%)
Fig. 6.24	42	96.42

Natural Condition Transport Approach — Trained Flow Models.		
s/c Specifications		
$a_0 = 2.2 \times 10^{-4}\ m/s^2$	$\widetilde{\mathbb{T}} = 0.22\ N$	$\widetilde{Isp} = 4000\ s$

Transition of Initial Guesses to Low-Thrust Solutions

The state, time, thrust magnitude and direction histories from the free-form and model-based initial guesses from Fig. 6.23(b) and Fig. 6.24 are input to the optimization scheme to produce the low-thrust continuous transfers that appear in Fig. 6.25.



(a) Interior: from flow-model IG

(b) Exterior: from free-form IG

(c) Zoom-in of (b)

Figure 6.25: Mass optimal low-thrust transfers from L_2 NRHO to southern L_2 halo — (a) constructed from initial guess in Fig. 6.24 and (b) constructed from initial guess in 6.23(b), (c) Side-view of (b)

Table 6.18: Performance metrics for CSI transfers in Fig. 6.25.

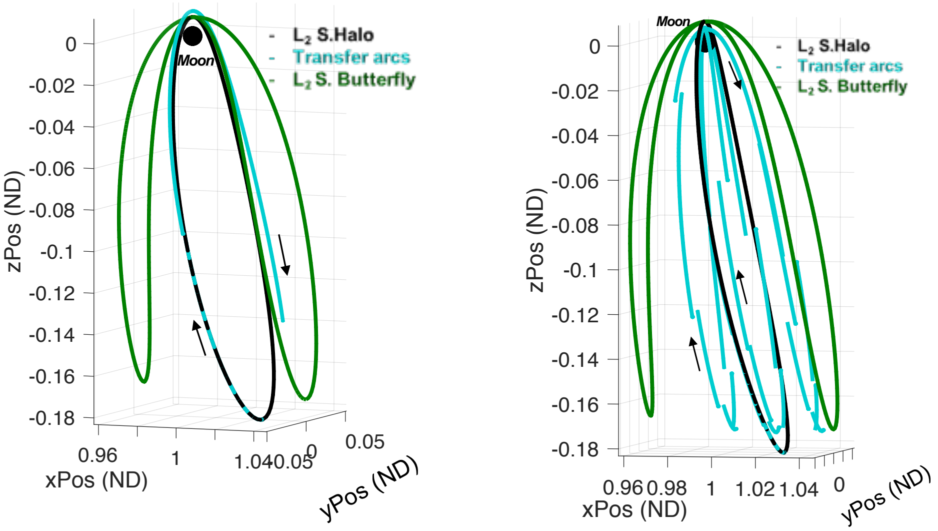
Scenario	TOF (days)	TD (days)	$\frac{m_f}{m_0}$ (%)	ΔV_{equiv} (m/s)
Fig. 6.25(a): From flow-model IG (Fig. 6.24)	48.48	26.86	98.70	513.85
Fig. 6.25(b): From free-form IG (Fig. 6.23(b))	131.32	23.88	98.84	456.56

s/c Specifications		
$a_0 = 2.2 \times 10^{-4}\ m/s^2$	$\widetilde{\mathbb{T}} = 0.22\ N$	$\widetilde{Isp} = 4000\ s$

These continuous solutions maintain a similar geometry and comparable TOF values to those for their corresponding initial guesses. The interior transfer in the scenarios computed in Fig. 6.25(a) enables faster transport between the departure and destination, with one fewer lunar flyby. Consequently, the increased thrust duration elicits higher propellant consumption than the exterior transfer. The outcome for the delivered mass-fraction comparisons between the interior and exterior transfers is contrary to the comparison of the predictions of these values from the initial guesses. A contributing factor for this discrepancy is that, as mentioned earlier, burns near the Moon or close approaches of it can lead to large discontinuities between the natural arcs computed via the ARs; the exterior flyby that possesses multiple close approaches is thus impacted. The ability to select pre-reinforced states in the vicinity of an AR in the spatial free-form scenarios to aid in pathfinding as discussed earlier is also a contributing factor. However, both initial guesses prove sufficient to construct their respective mass-optimal continuous transfers for comparisons in higher-fidelity models.

6.5.6 Controlling Arc Horizon Time

An alternative approach for generating different transfer profiles for a given spacecraft and scenario is exercising control over the propagation time along each segment during the pathfinding process. Figure 6.26(a) illustrates a fairly direct transfer geometry between an NRHO and butterfly orbit. However, even with such a transfer profile dynamically possible, some mission scenarios and/or constraints may require a gradual shift into the destination orbit regardless of spacecraft capability. As such, Fig. 6.26(b) demonstrates the flexibility within the design framework to allow incorporation of additional revolutions into the transfer geometry during the pathfinding process. Such an outcome is achieved by shortening the propagation duration for each arc, which then reduces the size of the associated accessible region and, thus, decreases the range and type of natural conditions available to the spacecraft within the searchable database. The algorithm then transitions at a slower rate across a given family, increasing the number of intermediate revolutions for a given transfer.



(a) $\delta \mathbb{t}_n = 2.5\ days$ (b) $\delta \mathbb{t}_n = 1\ day$

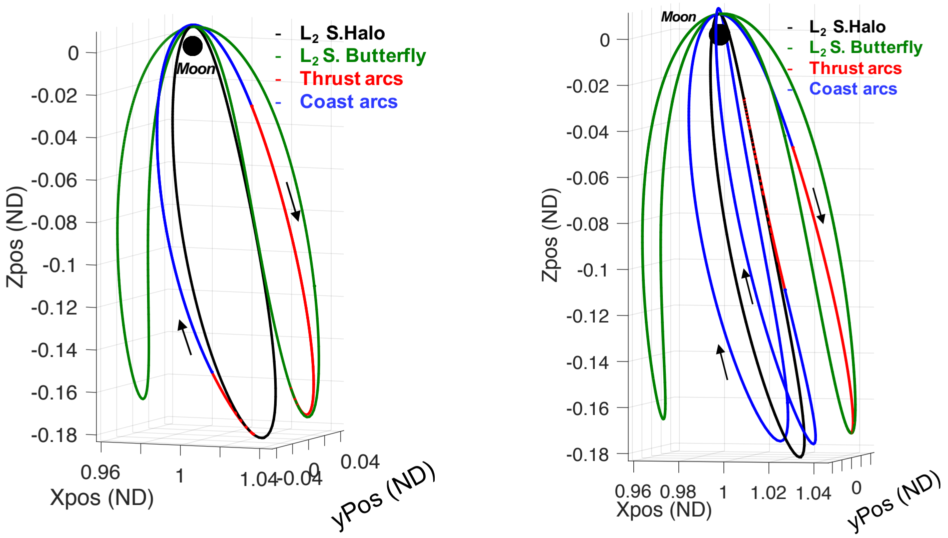
Figure 6.26: Ability to influence rate of transition profile via propagation horizon time — (a) rapid transition, (b) gradual transition. Departure: Southern L_2 NRHO ($r_p = 3500\ km$), Destination: Southern L_2 Butterfly ($r_p = 3500\ km$)

Table 6.19: Performance metrics for Fig. 6.26 illustrating varied transfer profiles resulting from varied burn durations $\delta\mathbb{t}_n$ per transfer segment

Scenario	TOF (days)	TD (days)	$\frac{m_f}{m_0}$ (%)	ΔV_{equiv} (m/s)
Fig. 6.26(a): $\delta\mathbb{t}_n = 2.5\ days$	5.0	-	99.90	-
Fig. 6.26(b): $\delta\mathbb{t}_n = 1\ day$	21	-	99.61	-

Natural Condition Transport Approach — *A Priori* Discretized Database		
s/c Specifications		
$a_0 = 2.2 \times 10^{-4}\ m/s^2$	$\widetilde{\mathbb{T}} = 0.22\ N$	$\widetilde{Isp} = 4000\ s$

The associated mass-optimal solutions are constructed in Fig. 6.27. The initial guesses are readily transitioned to continuous trajectories and optimized, where the resulting solutions retain the general performance traits associated with their corresponding initial guesses.



(a) From IG with $\delta\mathbb{t}_n = 2.5\ days$ (b) From IG with $\delta\mathbb{t}_n = 1\ day$

Figure 6.27: Mass optimal CSI solutions for initial guesses in Fig. 6.26.

Table 6.20: Performance metrics for CSI transfers in Fig. 6.27.

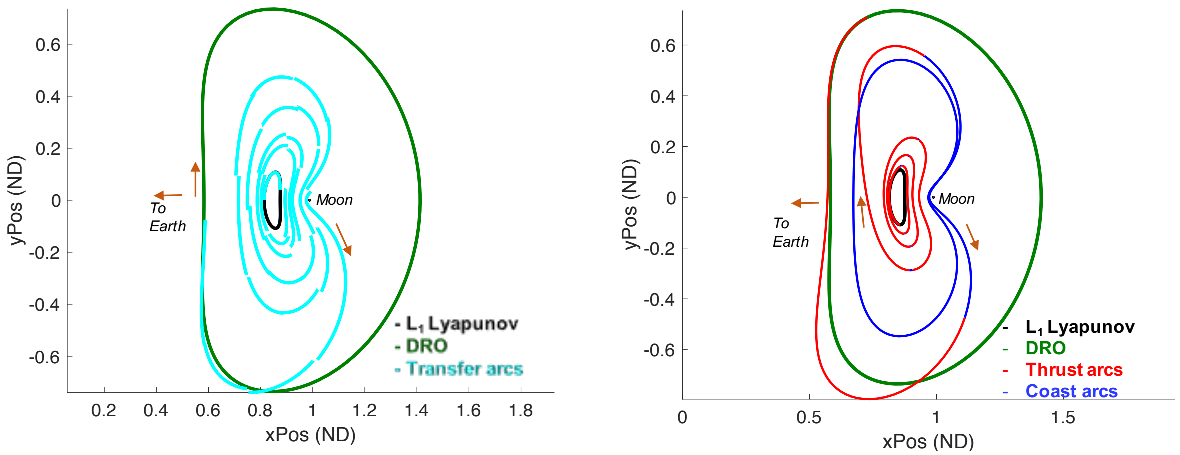
Scenario	TOF (days)	TD (days)	$\frac{m_f}{m_0}$ (%)	ΔV_{equiv} (m/s)
Fig. 6.27(a): From IG with $\delta \mathbb{t}_n = 2.5\ days$	7.06	5.03	99.76	95.79
Fig. 6.27(b): From IG with $\delta \mathbb{t}_n = 1\ day$	16.24	4.04	99.80	76.84
s/c Specifications				
$a_0 = 2.2 \times 10^{-4}\ m/s^2$	$\widetilde{\mathbb{T}} = 0.22\ N$		$\widetilde{Isp} = 4000\ s$	

6.6 Incorporate s/c Performance Specifications

The ability to incorporate s/c performance specifications while conducting mission architecture trades is a valuable utility. Using the proposed search scheme, an opportunity exists to evaluate the appropriateness of specific s/c specifications for a given mission application. Additionally, relating the trajectory outcomes to propulsion system capabilities offers insights into technology development strategies to best enable and enhance future missions.

6.6.1 Varied Acceleration Levels — Planar Realm

A 14 kg, 6U-class cubesat follows a transfer path in Fig. 6.28 from a Lyapunov departure to a DRO destination orbit. These s/c specifications are representative of current cubesat capabilities, e.g., NASA Goddard's 6U cubesat for the Lunar IceCube mission [38]. The flexibility to incorporate the s/c specifications into the design process informs the use of multiple intermediate revolutions required to build up the energy change required to transfer a s/c with low acceleration capabilities ($a_0 = 8.57 \times 10^{-5} m/s^2$) to the DRO; in this case, the consequence is a long TOF. Once the target orbit falls within an AR, it becomes a candidate for selection by an agent and the search terminates for this agent. The ability of the automated search process to exploit beneficial energy transition strategies such as performing lunar passes is



(a) Initial guess

(b) Mass optimal

Figure 6.28: Transferring a cubesat-class spacecraft between — Departure: L_1 Lyapunov orbit ($JC = 3.147$) and Destination: DRO ($JC = 2.785$)

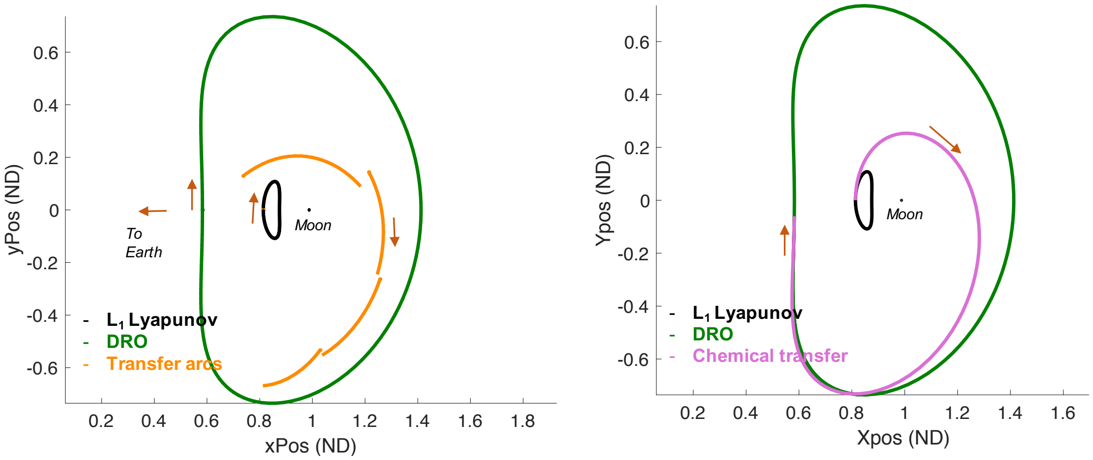
Table 6.21: Performance metrics for the planar cube-sat transfer scenario in Fig. 6.28

Scenario	TOF (days)	TD (days)	$\frac{m_f}{m_0}$ (%)	ΔV_{equiv} (m/s)
Fig. 6.28(a): Initial guess	99.00	-	98.23	-
Fig. 6.28(b): Mass optimal	112.33	73.31	97.78	549.03

Natural Condition Transport Approach — *A Priori* Discretized Database		
s/c Specifications		
$a_0 = 8.6 \times 10^{-5}\ m/s^2$	$\widetilde{\mathbb{T}} = 1.2\ mN$	$\widetilde{Isp} = 2500\ s$

illustrated in Fig. 6.28(b). Results in the low-thrust regime are shifted to the chemical realm for comparisons. The transfer scenario in Fig. 6.29 is generated for a chemical engine with burn duration $\delta\mathbb{t}_n \approx 1\ hour$ and arc propagation durations on the order of days. Access to a greater acceleration magnitude ($a_0 = 0.036\ m/s^2$) enables the transfer to occur more quickly than the cubesat. This example is also an opportunity to reflect on the varied geometries available by allowing larger discontinuities due to the greater thrust capabilities (and, therefore, greater ARs) that are accomplished with the chemical engine. The decreased number of arcs in the transfer sequence accessible to the more powerful chemical engine enables a quicker transfer compared to the longer path that is required by the lower-thrust s/c (Fig. 6.28) to bridge the energy gap between the departure and destination orbits. While the ΔV value is lower for the path constructed for the chemical engine in this scenario, it is also important to consider the efficiency of the two engines via their respective specific impulse values.

The ability to successfully converge and produce optimized results across multiple engine types reflects on the capabilities of the implemented methodology.



(a) Initial guess

(b) ΔV optimal

Figure 6.29: Demonstrating extension of the automated pathfinding capability to the chemical engine regime for transfer between same conditions as in Fig. 6.28.

Table 6.22: Performance metrics for the planar chemical transfer scenario in Fig. 6.29

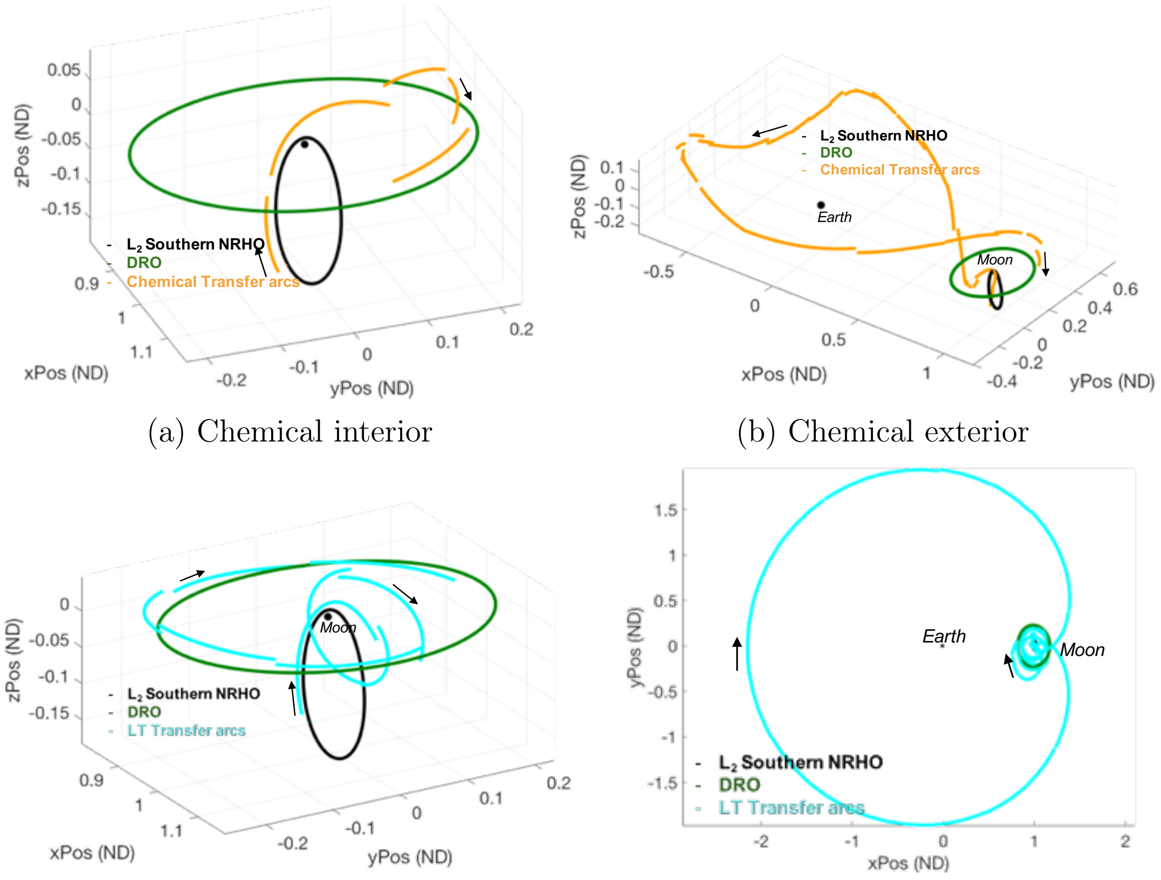
Scenario	*TOF* (days)	ΔV (m/s)
Fig. 6.29(a): Initial guess	15.0	450.24
Fig. 6.29(b): ΔV optimal	20.01	399.86
Natural Condition Transport Approach — *A Priori* Discretized Database		
s/c Specifications		
$a_0 = 3.6 \times 10^{-2}\ m/s^2$	$\widetilde{\mathbb{T}} = 18\ N$	$\widetilde{Isp} = 224\ s$

6.6.2 Varied Acceleration Levels — Spatial Realm

Comparisons of transfer solutions associated with different s/c capabilities in the spatial realm are undertaken via both free-form and *a priori* discretized database natural condition transport approaches. The former approach investigates transfers between a southern L_2 NRHO and DRO for a chemical and low-thrust s/c, and the latter approach assesses the performance outcomes for transferring varied capabilities of low-thrust s/c between a southern L_2 halo and southern L_2 NRHO.

Comparison of Chemical and Low-Thrust Solutions

The reverse-time seeding of waypoints to the destination (Fig. 6.22) is especially useful in illuminating pathways under circumstances where no natural flows exist into/out of stable orbits in the spatial realm, as with the southern L_2 NRHO and DRO (Fig. 6.30). Also, instead of adopting flow-models to maintain an interior geometry in this example, the alternative strategy of producing smaller accessible regions via shorter propagation durations (e.g., Fig. 6.26(b)) to induce a more gradual exit from the NRHO is adopted. Activating the low-thrust engine over a time-horizon of 3 days in Fig. 6.30(d) results in an equivalent $\Delta V \approx 57\ m/s$ per transfer arc from the NRHO. This ΔV kick, along with the additional boost received from the Moon during the NRHO departure produces large ARs and exposes the s/c to arcs that lead to external transfers. Reducing the time horizon to 2 days reduces the 'kick' to an equivalent $\Delta V \approx 38\ m/s$ and shrinks the size of the ARs sufficiently to enable interior transfers as well (Fig. 6.30(c)). Similarly, a 1 hour engine-burn time and 1 day propagation duration for the chemical s/c enables interior transfers as well. The appropriate burn duration (and so, the corresponding ΔV magnitude) for a particular scenario is currently determined empirically. So, the example in Fig. 6.30 reveals that trained flow models are not 'necessary' to construct interior transfers, even though exploiting the models may be convenient and beneficial for extracting additional geometries as already demonstrated.



(a) Chemical interior

(b) Chemical exterior

(c) Low-thrust interior

(d) Low-thrust exterior

Figure 6.30: Manipulating engine-burn time and, thus, maneuver and AR size to construct initial guesses linking the spatial and planar realm between two stable orbits (free-form searches only). Departure: NRHO ($rPAlt = 1763.31\ km$) to DRO (JC: 2.935, Period: $\approx 13\ days$). Note, $rP \rightarrow$ Periapsis.

The TOF values for the examples in Figs. 6.30(a & c) illustrate that low-thrust capability improvement from very low cubesat acceleration levels (e.g., Fig. 6.28) can allow ion-engines to deliver solutions with comparable transfer durations to chemical engines. Note that the estimated ΔV for the exterior transfer initial guess in 6.30(b) is high at 2.64 km/s. This initial guess is constructed by implementing maneuvers at 1 day time-intervals; in reality, a chemical transfer would consist of fewer burns and longer ballistic arcs. However, once the initial guess is acquired, numerical corrections and optimization eliminate undesirable maneuvers and render a more realistic

Table 6.23: Performance metrics for Fig. 6.30 illustrating varied spatial transfer profiles for chemical and low-thrust engines for the same transfer scenario.

Scenario	Estimated TOF (days)	Estimated $\frac{m_f}{m_0}$(%)	Estimated ΔV (m/s)
Fig. 6.30(a): Chemical interior	9	-	711.61
Fig. 6.30(b): Chemical exterior	48	-	2640
Fig. 6.30(c): Low-thrust interior	20	97.93	-
Fig. 6.30(d): Low-thrust exterior	126	90.16	-

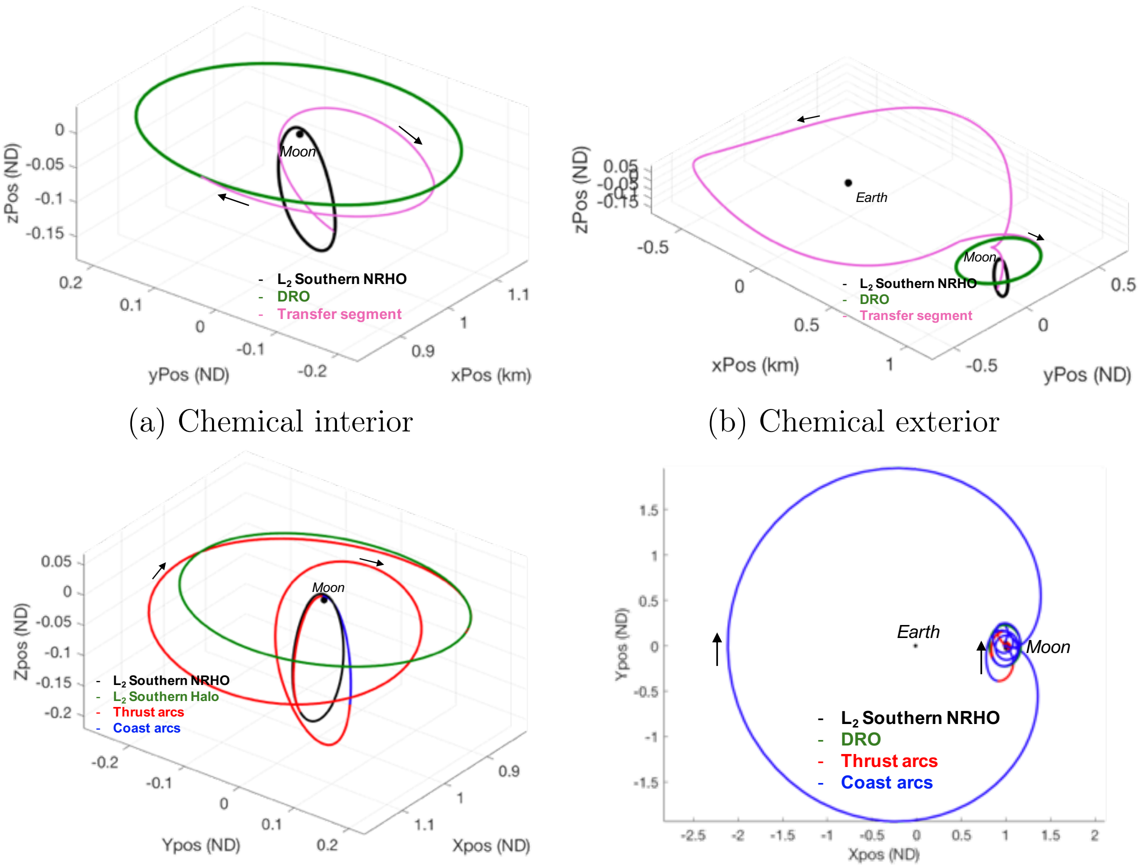
Natural Condition Transport Approach — Free-Form Search.		
Low-Thrust s/c Specifications		
$a_0 = 2.2 \times 10^{-4}\ m/s^2$	$\widetilde{\mathbb{T}} = 0.22\ N$	$\widetilde{Isp} = 4000\ s$
Chemical s/c Specifications		
$a_0 = 3.6 \times 10^{-2}\ m/s^2$	$\widetilde{\mathbb{T}} = 18\ N$	$\widetilde{Isp} = 224\ s$

chemical transfer solution (Fig. 6.31). The initial guesses prove sufficient in enabling

Table 6.24: Performance metrics for spatial chemical and low-thrust transfer scenarios in Fig. 6.31.

Scenario	TOF (days)	TD (days)	$\frac{m_f}{m_0}$(%)	ΔV_{equiv} (m/s)
Fig. 6.31(a): Chemical interior:	10.78	-	-	481.04
Fig. 6.31(b): Chemical exterior:	50	-	-	626.48
Fig. 6.31(c): Low-thrust interior	28.86	26.64	98.71	509.67
Fig. 6.31(d): Low-thrust exterior	124.16	21.17	98.97	404.54

Low-Thrust s/c Specifications		
$a_0 = 2.2 \times 10^{-4}\ m/s^2$	$\widetilde{\mathbb{T}} = 0.22\ N$	$\widetilde{Isp} = 4000\ s$
Chemical s/c Specifications		
$a_0 = 3.6 \times 10^{-2}\ m/s^2$	$\widetilde{\mathbb{T}} = 18\ N$	$\widetilde{Isp} = 224\ s$



(a) Chemical interior

(b) Chemical exterior

(c) Low-thrust interior

(d) Low-thrust exterior

Figure 6.31: ΔV Optimal chemical and mass-optimal low-thrust solutions from free-form search initial guesses in Fig. 6.30.

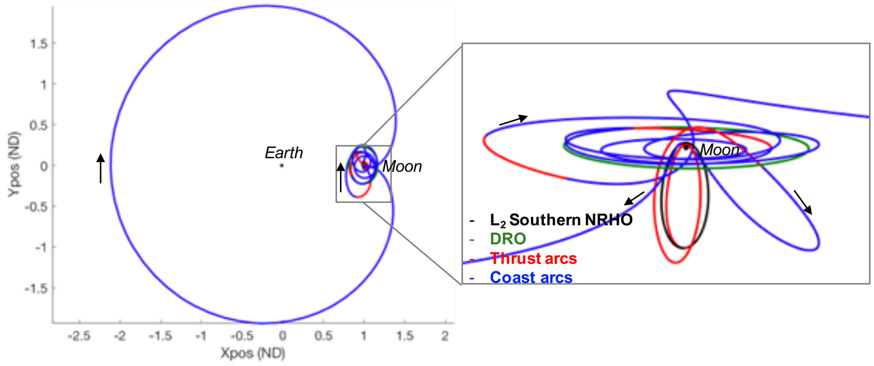


Figure 6.32: Zoomed-in view of solution in Fig. 6.31(d).

the chemical and low-thrust spacecraft to transition between varied energy levels, inclinations, and locations in configuration space. As discussed for the example in Fig. 6.25(c), the multiple passes by the Moon in Fig. 6.31(d) (Fig. 6.32) to effect the required transfer trajectory characteristics result in a higher delivered mass-fraction compared to the shorter mass-optimal low-thrust solution in Fig. 6.31(c). As discussed earlier, the ability to select pre-reinforced states in the vicinity of an AR in the spatial free-form scenarios to aid in pathfinding, and the greater allowable discontinuities between natural arcs near the Moon lead to a poorer initial estimate of the delivered mass fraction in Fig. 6.30(d) that employs multiple revolutions to capture into the DRO compared to the scenario in Fig. 6.30(c).

Comparison of Low-Thrust Capabilities

The results displayed in Fig. 6.33 illustrate that the design framework offers intuition on the technology levels required to execute transfer scenarios within a desired time frame and mass budget in the spatial realm as well. Three acceleration levels demonstrate performance trades-offs — $3 \times 10^{-5} m/s^2$ corresponds to the performance characteristics on par with the Powered Propulsion Element (PPE) for the Deep Space Gateway; $2.2 \times 10^{-4} m/s^2$ is closer to the level observed with the DS1 mission, and $4.4 \times 10^{-4} m/s^2$ is selected arbitrarily and is just over twice the acceleration magnitude associated with DS1. All three spacecraft depart a near-planar L_2 halo orbit and travel to an L_2 Near Rectilinear Halo Orbit (NRHO) with a periapsis radius of $\approx$ 3500 km, expected to support potential near term cis-lunar operations. Only the southern L_2 halo family conditions are incorporated within the searchable database in this example, so all the transfer arcs belong to this family. As in the planar case, the lower acceleration levels lead to an increased number of revolutions to equip the low-thrust engine with adequate time to bridge the energy and inclination differences between the departure and destination conditions. Increased acceleration levels are able to improve the travel duration to weeks rather than months. The ability to conduct such performance comparisons aids in assessing the benefits of improved acceleration levels to overall mission capabilities and thus, enables effective mission planning.

Table 6.25: Performance metrics for Fig. 6.33 illustrating varied transfer profiles in the spatial realm via different low-thrust s/c acceleration capabilities for the same transfer scenario.

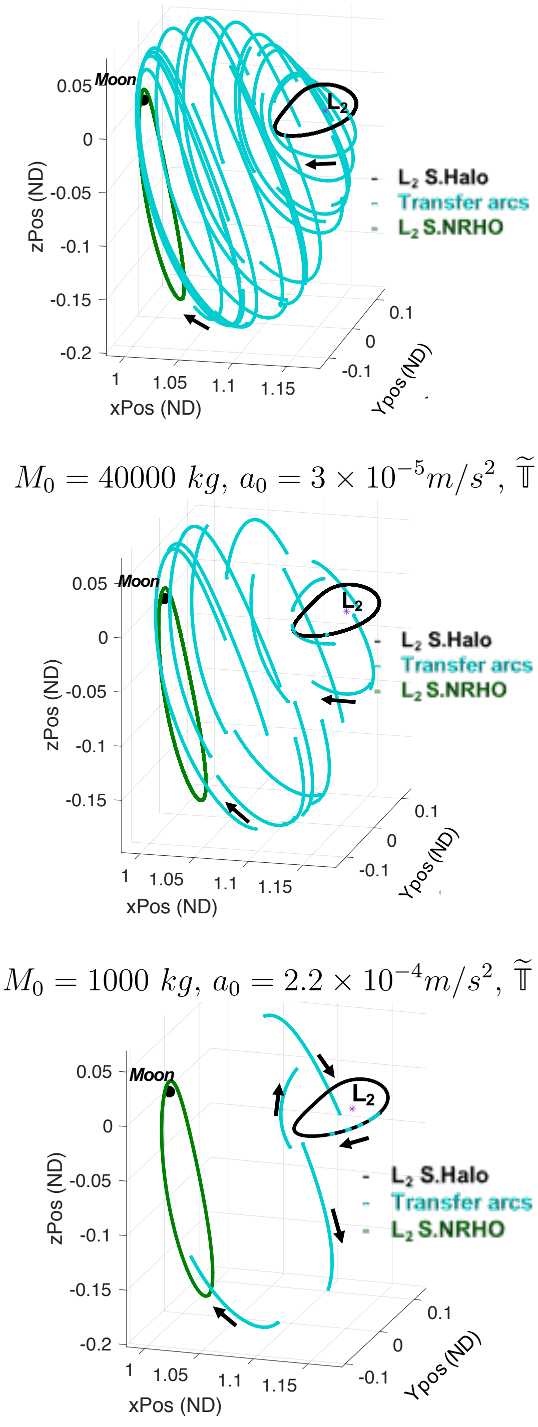
Scenario	TOF (days)	$\frac{m_f}{m_0}$
Fig. 6.33(a): PPE Acceleration	168	99.40
Fig. 6.33(b): ≈ DS1 Acceleration	60	98.34
Fig. 6.33(c): ≈ ×2 DS1 Acceleration:	15	99.07
Natural Condition Transport Approach — *A Priori* Discretized Database		
s/c Specifications		
$\widetilde{Isp} = 4000\ s$		

The initial guesses are readily transitioned to continuous low-thrust solutions and the optimized CSI transfers are displayed in Fig. 6.34. A similar equivalent ΔV value representing the required energy change is observed for the traversals in all three transfer scenarios. Additionally, the maintenance of the same engine efficiency

Table 6.26: Performance metrics for CSI transfers in Fig. 6.34.

Scenario	TOF (days)	TD (days)	$\frac{m_f}{m_0}$ (%)	ΔV_{equiv} (m/s)
Fig. 6.34(a): PPE Acceleration	171.30	154.80	98.97	403
Fig. 6.34(b): ≈ DS1 Acceleration	59.70	20.7	99.00	396
Fig. 6.34(c): ≈ ×2 DS1 Acceleration	17.23	11.96	98.84	457
s/c Specifications				
$\widetilde{Isp} = 4000\ s$				

(i.e., $\widetilde{Isp} = 4000s$) results in similar mass fractions ($\frac{m_f}{m_0}$) for these examples. However, the absolute value of the propellant consumed varies due to a different initial mass in each case as listed in the figures.

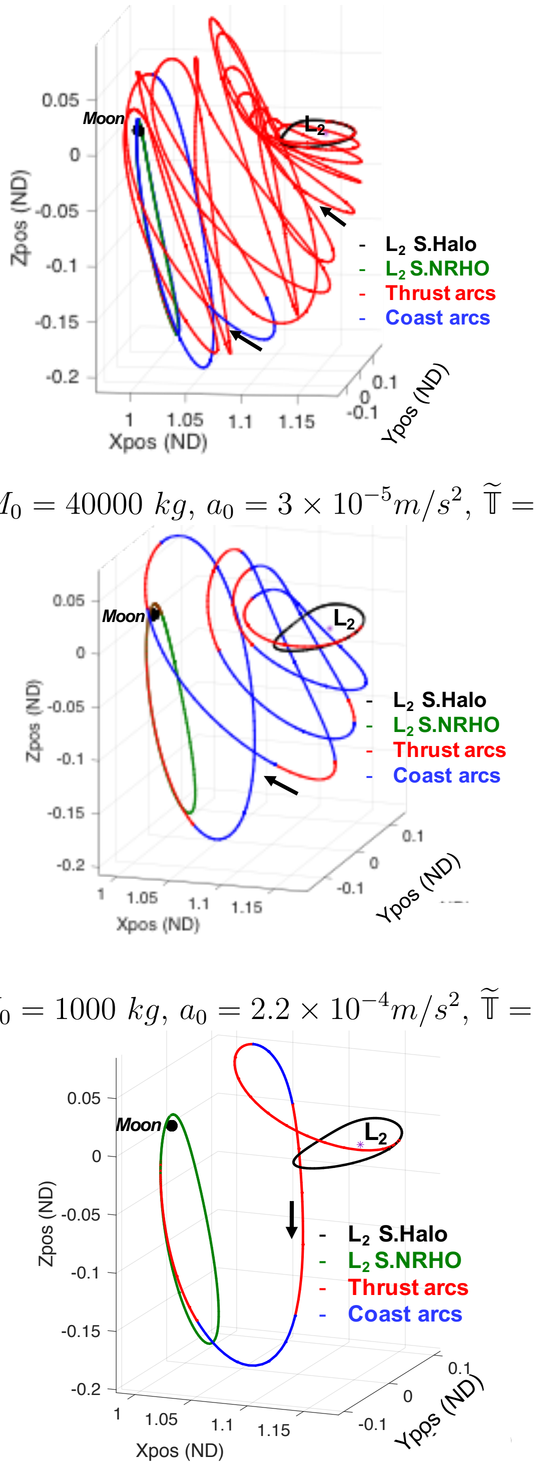


(a) $M_0 = 40000\ kg$, $a_0 = 3 \times 10^{-5} m/s^2$, $\widetilde{\mathbb{T}} = 1.2\ N$

(b) $M_0 = 1000\ kg$, $a_0 = 2.2 \times 10^{-4} m/s^2$, $\widetilde{\mathbb{T}} = 0.22\ N$

(c) $M_0 = 500\ kg$, $a_0 = 4.4 \times 10^{-4} m/s^2$, $\widetilde{\mathbb{T}} = 0.22\ N$

Figure 6.33: Initial guess generation in spatial realm — impact of spacecraft acceleration magnitudes on mission profile



(a) $M_0 = 40000\ kg$, $a_0 = 3 \times 10^{-5} m/s^2$, $\widetilde{\mathbb{T}} = 1.2\ N$

(b) $M_0 = 1000\ kg$, $a_0 = 2.2 \times 10^{-4} m/s^2$, $\widetilde{\mathbb{T}} = 0.22\ N$

(c) $M_0 = 500\ kg$, $a_0 = 4.4 \times 10^{-4} m/s^2$, $\widetilde{\mathbb{T}} = 0.22\ N$

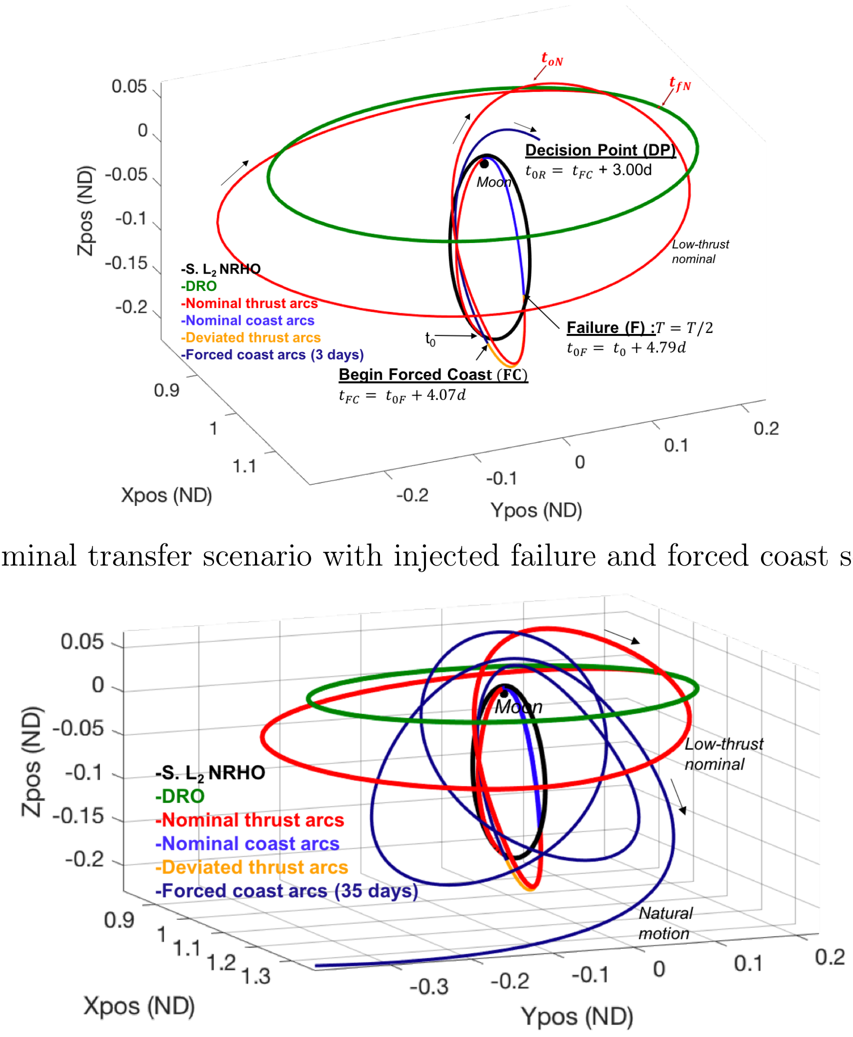
Figure 6.34: Low-thrust optimized solutions for initial guesses in Fig. 6.33

6.7 Explore Contingency Scenarios

System failure and/or degradation during flight can cause the s/c to significantly deviate from its nominal transfer trajectory, leading to undesirable impacts on the mission. For example, a safing event for the Dawn spacecraft resulted in ≈ 4 *days* of missed thrust that then led to a re-design of the s/c's baseline approach to Ceres with an ultimate delay of ≈ 26 *days* to capture into a particular science orbit [131]. Monte carlo simulations incorporating information from historical data in Imken et al. [132] also suggest that the typical duration of inoperability due to safing events is ≈ 3.5 *days*. In such situations, there is a critical need for rapid trajectory design solutions to minimize adverse mission outcomes. This investigation assesses the ability of the machine learning algorithms to recover during contingency scenarios via a motivating example where the engine throughput is compromised. In particular, two types of recovery scenarios are considered: (i) recovery to the arrival orbit, and (ii) recovery to rendezvous with a pre-positioned target on the arrival orbit. In this preliminary investigation, orbit determination errors and higher-fidelity ephemeris models are not considered.

6.7.1 Recovery to Arrival Orbit Conditions

Transfer scenarios motivated by long-term scientific observations or end-of-life disposal may seek to transition a s/c to desired orbital conditions that satisfy orbit stability, communications, and other considerations. The transfer from an NRHO to a DRO in Fig. 6.31(c) is one such scenario and is thus chosen as the motivating example to demonstrate s/c recovery options via machine learning strategies. The nominal transfer path along with the simulated failure and post-failure routes are illustrated in Fig. 6.35(a). All the contingency examples are executed for a s/c initial mass of $M_0 = 1000\ kg$, with the s/c performance metrics specific to Fig. 6.35 outlined in Table 6.27. The acronym DFD for all the contingency examples refer to Days From Departure on the NRHO (i.e., from the location on the NRHO where the s/c departs



(a) Nominal transfer scenario with injected failure and forced coast segments

(b) Outcome from no corrective thrusting action after DP

Figure 6.35: Contingency scenario example — Departure: NRHO (Periapsis Altitude = 1763.31 km), Arrival: DRO (JC: 2.935, Period: $\approx$ 13 $days$). (a) Engine failure simulated $\approx$ 5 $days$ after departure from NRHO, and the trajectory deviates due to diminished engine throughput, (b) Illustration of natural motion resulting from no corrective action for 35 $days$ post DP.

on the nominal trajectory). In this example, the engine throughput is diminished from $\widetilde{\mathbb{T}} = 0.22\ N$ by 50% at $t_{0F} \approx 5\ days$ into the nominal transfer (the equivalent to losing the thrust capability of one engine if there were 2 engines to begin with). Such a failure leads the s/c to drift from the nominal trajectory over $\approx$ 4 $days$, which is followed by a commanded forced coast at $t_{FC} \approx 8.86\ days$ for a duration

Table 6.27: Performance metrics for Fig. 6.35

Scenario	t_{0R} (DFD)	t_{0N} (DFD)	t_{fN} (DFD)	$t_{fN} - t_{0N}$ $(days)$	$\left(\frac{m_f}{m_0}\right)_N$ (%)
Fig. 6.35(a)	11.86	11.86	28.86	17	98.71
s/c Specifications for Nominal Trajectory					
$a_0 = 2.2 \times 10^{-4}\ m/s^2$			$\widetilde{\mathbb{T}} = 0.22\ N$	$\widetilde{Isp} = 4000\ s$	

of 3 days to simulate a safe mode recovery operation. The Decision Point (DP) indicates the epoch t_{0R} at which the recovery trajectory is initiated. The terminology associated with the contingency scenarios examined in this section is summarized in Table 6.28. The duration of the post-failure propagation and forced coast segment in Fig. 6.35(a) do not account for orbit determination but are intended to introduce significant deviations from the nominal trajectory to investigate the capabilities of the design framework and machine learning techniques to construct recovery options. The ability to recover to the nominal transfer path after the failure is a function of the failure circumstances (e.g., where along the trajectory the failure occurs); effective

Table 6.28: Definition of symbols employed in contingency-preparation-related results.

Symbols	Definition
$\left(\frac{m_f}{m_0}\right)_N$	Delivered mass fraction on nominal trajectory.
t_{0F}	Absolute epoch of failure initiation on nominal path.
t_{FC}	Absolute epoch of forced coast initiation.
t_{0N}	Absolute epoch on nominal path corresponding to absolute epoch of recovery initiation on recovery path.
t_{fN}	Absolute epoch on nominal path corresponding to its terminal state.
$t_{0R} =$ DP	Absolute epoch on recovery path corresponding to the termination of the forced coast segment / initiation of the recovery path from the Decision Point (DP).
t_{fR}	Absolute epoch on recovery path corresponding to its terminal state at the target.

response may also require feedback control. The results presented in this section do not recover the nominal path because of the extensive deviations introduced from the nominal route. The forced-coast segment in Fig. 6.35(a) is extended for an additional 32 *days* in Fig. 6.35(b) to illustrate the spacecraft's departure from the vicinity of the destination in the absence of no corrective measures over an extended duration.

The recovery of a s/c from the failure scenario in Fig. 6.35(a) is implemented via the design framework components discussed in Chapter 5, specifically via free-form initial guess generation followed by numerical corrections to deliver a continuous end-to-end path. Since the recovery is not constrained to strict time-frames in this non-rendezvous transfer scenario, the global mass-time cost function (Eqn. (5.2a)) is incorporated with priority placed on TOF. The recovery path resulting from restoring the s/c with full thrust at t_{0R} is obtained by executing the pathfinding framework to target discretized states over 1 period (≈ 13 *days*) of the DRO. The initial guess for such a transfer computed by the pathfinding agents is presented in Fig. 6.36. Although the duration of the recovery arcs reflect a similar TOF to that observed in the nominal path after t_{0N} (comparing Tables 6.27 and 6.29), the path assumed during recovery is substantially different from the nominal route. Such an outcome is expected,

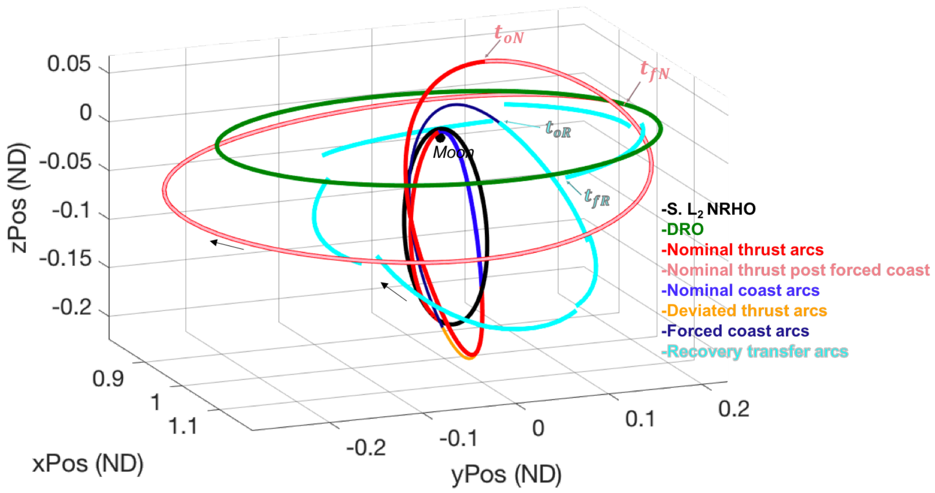


Figure 6.36: Initial guess to recover with $\widetilde{\mathbb{T}} = 0.22$ N from failure introduced in Fig. 6.35(a)

Table 6.29: Performance metrics for s/c recovery scenario in Fig. 6.36

Scenario	t_{0R} (DFD)	t_{fR} (DFD)	$t_{fR} - t_{0R}$ $(days)$	$\frac{m_f}{m_{0\ est}}$ $(\%)$
Fig. 6.36(a)	11.86	27.86	16	97.92

Natural Condition Transport Approach — Free-Form Search.
s/c Specifications Pre-Failure
$a_0 = 2.2 \times 10^{-4}\ m/s^2$ $\widetilde{\mathbb{T}} = 0.22\ N$ $\widetilde{Isp} = 4000\ s$
s/c Specifications Post-Forced Coast
$a_0 = 2.205 \times 10^{-4}\ m/s^2$ $\widetilde{\mathbb{T}} = 0.22\ N$ $\widetilde{Isp} = 4000\ s$

as the combination of the imposed failure and forced-coast segment render a loss in altitude of $\approx 25,000\ km$ from the nominal solution at $t_{0N} = t_{0R} = 11.86\ days$ and, thus, also results in varied trajectory options for the s/c states on the two trajectories at this epoch to continue towards the destination. The initial guess is successfully transformed into a continuous mass-optimal low-thrust solution in Fig. 6.37. The mass-optimal objective inserts coasting time along the recovery solution, leading the s/c to reach the destination orbit a week after the nominal epoch at t_{fN}. However, it is evident that the thrust durations are similar between the nominal and recovery solutions. The thrusting period of $\approx 19\ days$ in Fig. 6.37 enables the s/c to execute the energy and plane change required to maneuver into the DRO. This thrust

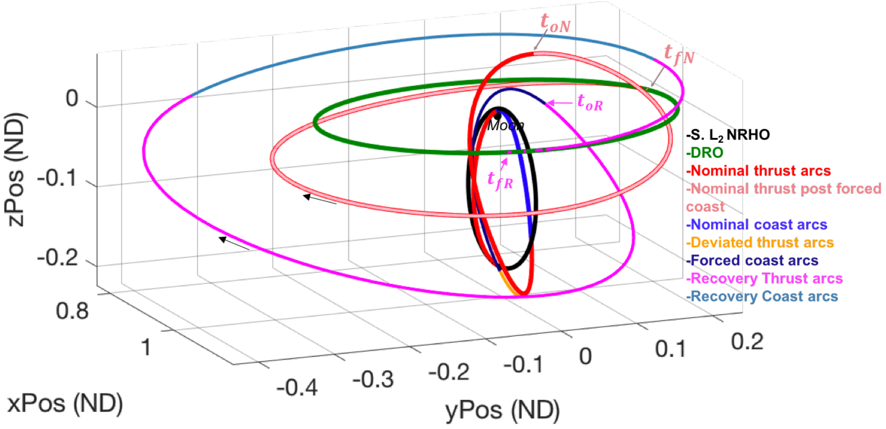


Figure 6.37: Low-thrust mass-optimal solution for initial guess presented in Fig. 6.36.

Table 6.30: Performance metrics for mass optimal CSI s/c recovery in Fig. 6.37

Scenario	t_{0R} (DFD)	t_{fR} (DFD)	$t_{fR} - t_{0R}$ $(days)$	TD $(days)$	$\frac{m_f}{m_0}$ $(\%)$
Fig. 6.37	11.86	36.43	24.57	19	98.86
s/c Specifications Pre-Failure					
$a_0 = 2.2 \times 10^{-4}\ m/s^2$		$\widetilde{\mathbb{T}} = 0.22\ N$		$\widetilde{Isp} = 4000\ s$	
s/c Specifications Post-Forced Coast					
$a_0 = 2.205 \times 10^{-4}\ m/s^2$		$\widetilde{\mathbb{T}} = 0.22\ N$		$\widetilde{Isp} = 4000\ s$	

duration results in a mass fraction of 98.86% and is close to the value of 97.92% estimated by the initial guess (IG). As discussed in earlier examples, the lower mass estimate for the IG results due to the large discontinuities between the transfer arcs in Fig. 6.36. The discrepancy between the IG and continuous solution results may be reduced by taking measures to better approximate the ARs in the highly nonlinear region in the vicinity of the primary and/or enforcing stricter rules on state selection from an AR. Also, the initial guess is constructed to prioritize time, whereas the continuous transfer optimizes final delivered mass; thus, better alignment of the objectives of the two stages of the design process in future investigations may result in different outcomes. Given the many potential design/optimization choice combinations, the example in Fig. 6.37 is useful as it demonstrates the ability to successfully optimize propellant consumption for a short duration recovery solution uncovered by the automated pathfinding algorithm.

In practical applications, it may not be always possible to recover the s/c with full thrust capabilities after an engine malfunction. The transfer scenario in Fig. 6.38 explores the outcome of maintaining the thrust levels at a diminished 50% capability after t_{0N} to deliver the s/c to the DRO. The diminished thrust and, therefore, acceleration levels available to the s/c in the example in Fig. 6.38 lead to a prolonged recovery duration compared to the solution in Fig. 6.36. The resulting geometry and arrival locations on the DRO are also different. The associated mass-optimal solution is presented in Fig. 6.39. As discussed for the example in Fig. 6.37, the

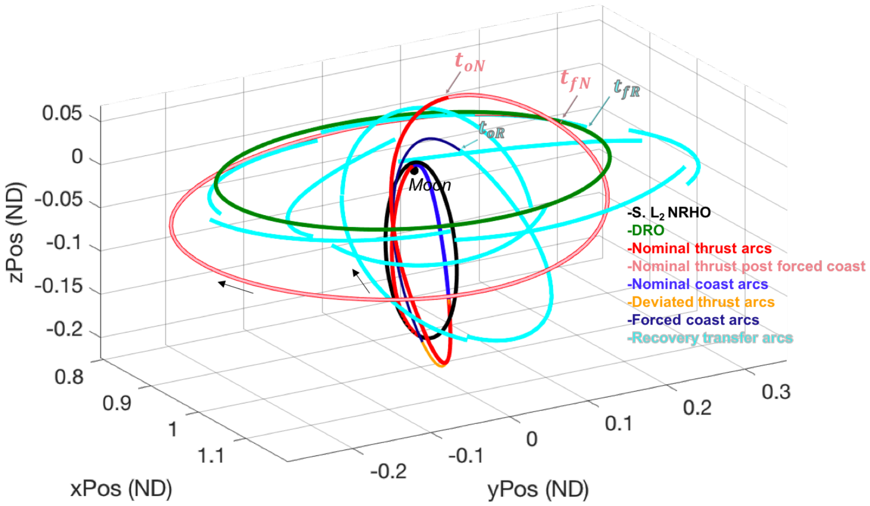


Figure 6.38: Initial guess to recover with $\widetilde{\mathbb{T}} = 0.11\ N$ from failure introduced in Fig. 6.35(a)

Table 6.31: Performance metrics for s/c recovery scenario in Fig. 6.38

Scenario	$t_{0R}\ (DFD)$	$t_{fR}\ (DFD)$	$t_{fR} - t_{0R}\ (days)$	$\frac{m_f}{m_{0\ est}}$ (%)
Fig. 6.38(a)	11.86	47.86	36	97.94
Natural Condition Transport Approach — Free-Form Search.				
s/c Specifications Pre-Failure				
$a_0 = 2.2 \times 10^{-4}\ m/s^2$ $\quad \widetilde{\mathbb{T}} = 0.22\ N$ $\quad \widetilde{Isp} = 4000\ s$				
s/c Specifications Post-Forced Coast				
$a_0 = 1.1025 \times 10^{-4}\ m/s^2$ $\quad \widetilde{\mathbb{T}} = 0.11\ N$ $\quad \widetilde{Isp} = 4000\ s$				

Table 6.32: Performance metrics for mass optimal CSI s/c recovery in Fig. 6.39

Scenario	$t_{0R}\ (DFD)$	$t_{fR}\ (DFD)$	$t_{fR} - t_{0R}\ (days)$	$TD\ (days)$	$\frac{m_f}{m_0}$ (%)
Fig. 6.39	11.86	57.11	45.25	43.53	98.72
s/c Specifications Pre-Failure					
$a_0 = 2.2 \times 10^{-4}\ m/s^2$ $\quad \widetilde{\mathbb{T}} = 0.22\ N$ $\quad \widetilde{Isp} = 4000\ s$					
s/c Specifications Post-Forced Coast					
$a_0 = 1.1025 \times 10^{-4}\ m/s^2$ $\quad \widetilde{\mathbb{T}} = 0.11\ N$ $\quad \widetilde{Isp} = 4000\ s$					

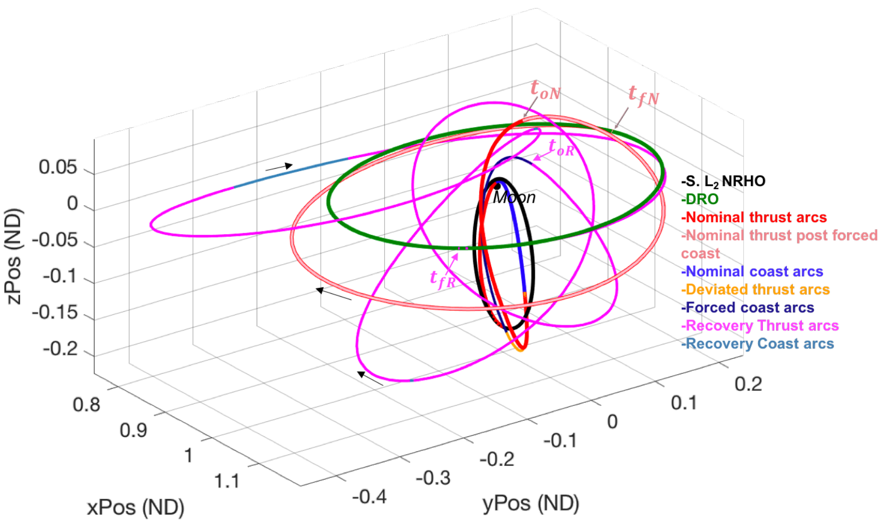
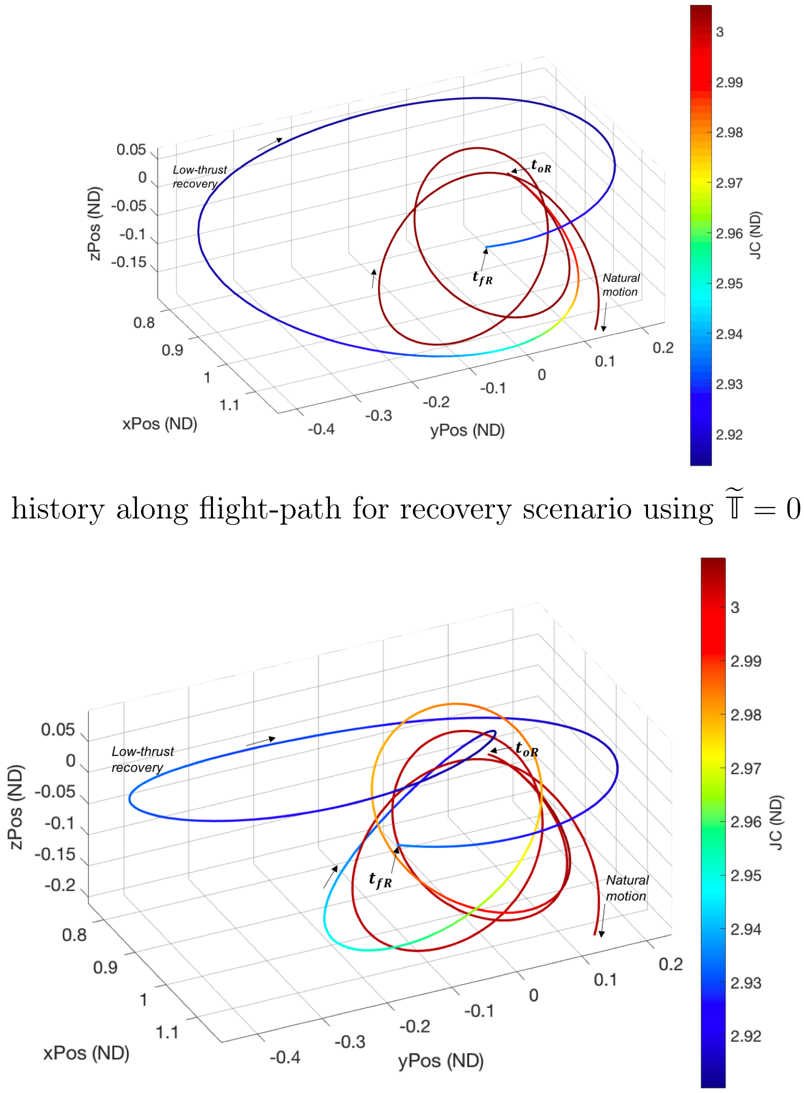


Figure 6.39: Low-thrust mass optimal solution for initial guess presented in Fig. 6.38.

mass-optimal solution eliminates the discontinuities in the IG and incorporates coast arcs where possible to minimize propellant consumption. The result is a recovery path that is almost twice as long compared to the case in Fig. 6.36, but note that the delivered mass fraction is similar. Recall that mass-flow rate is proportional to the s/c thrust levels (Eqn. (2.51)). So, a degradation in thrust magnitude by 50% leads to an equally diminished rate of propellant consumption; a trajectory that is twice as long thus leads to similar overall propellant consumption.

The change in inclination and energy levels required to complete the recovery lead to varying thrust durations based on the s/c thrust capabilities. The history of such energy changes for both the $\widetilde{\mathbb{T}} = 0.22\ N$ and $\widetilde{\mathbb{T}} = 0.11\ N$ recovery scenarios are illustrated in Fig. 6.40 (a) and (b), respectively. The path resulting from the absence of corrective measures at t_{0R} (i.e., natural motion) is represented by the constant maroon curve indicative of a JC value > 3. Both the $0.22\ N$ and $0.11\ N$ spacecraft thrust to deviate from the natural course of motion and maneuver towards the DRO; the more powerful s/c is able to transition through the energy levels more quickly than the $0.11\ N$ s/c. Interestingly, both solutions maneuver through energy levels greater than that of the arrival orbit prior to orbit insertion ($JC = 2.935$).

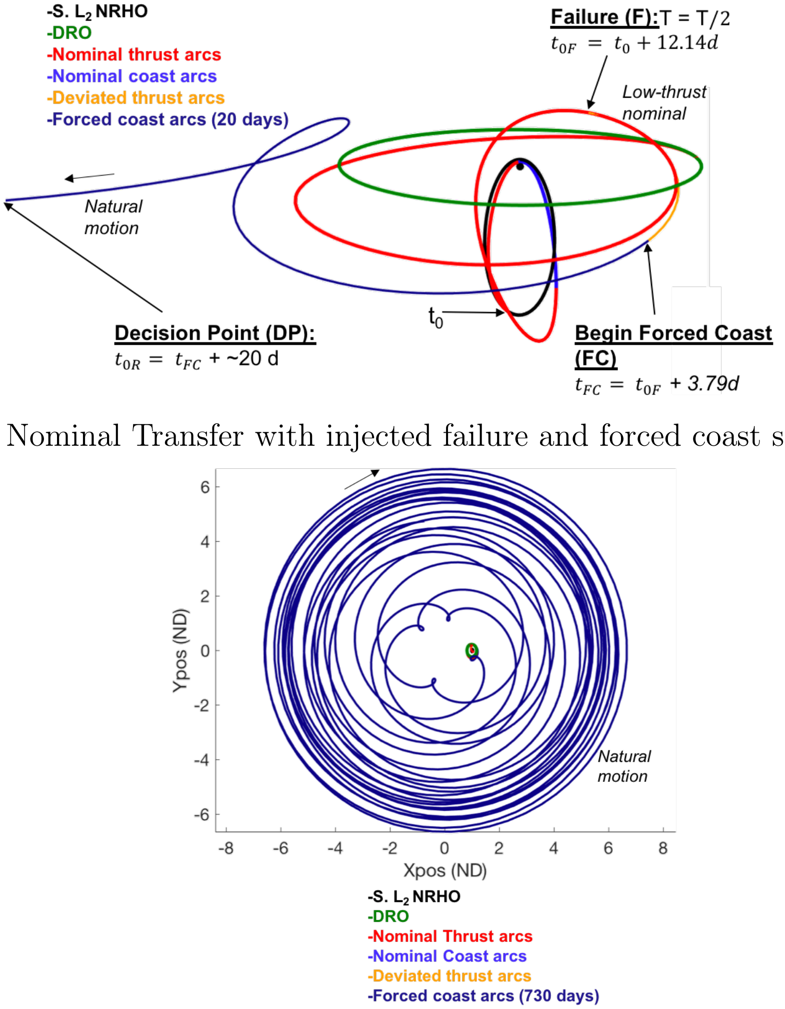


(a) JC history along flight-path for recovery scenario using $\widetilde{\mathbb{T}} = 0.22\ N$

(b) JC history along flight-path for recovery scenario using $\widetilde{\mathbb{T}} = 0.11\ N$

Figure 6.40: The Jacobi Constant (JC) history over the low-thrust flight paths in (a) Fig. 6.37 for s/c with 0.22 N thrust during recovery and, (b) Fig. 6.39 for s/c with 0.11 N thrust during recovery

The contingency scenarios discussed thus far exploit the free-form pathfinding approach to construct states within the ARs on the fly. It is also possible to leverage *a priori* known knowledge about the dynamical system via the database or flow-models to construct a recovery solution. This approach is pursued via the example in Fig. 6.41(a) that exploits the *a priori* discretized database approach. In this scenario, the s/c thrust magnitude is diminished from $\widetilde{\mathbb{T}} = 0.22\ N$ by 50% at $t_{0F} \approx 12\ days$ into the nominal transfer. Such a failure leads the s/c to drift from the nominal trajectory over $\approx 4\ days$, and is followed by a commanded forced coast at $t_{FC} \approx 15.93\ days$ for a duration of 20 days to simulate a safe mode operation. The Decision Point (DP) indicates the epoch t_{0R}, at which the recovery trajectory is initiated; the s/c is clearly traveling away from the destination at the DP. Note that the fields for t_{0N}



(a) Nominal Transfer with injected failure and forced coast segments

(b) Outcome from no corrective thrusting action after DP

Figure 6.41: Contingency scenario example — Departure: NRHO (Periapsis Altitude = $1763.31km$), Arrival: DRO (JC: 2.935, Period: $\approx 13\ days$. (a) Engine failure simulated $\approx 12\ days$ after departure from NRHO, and the trajectory deviates due to diminished engine throughput, (b) Illustration of natural motion resulting from no corrective action for 730 $days$ post DP

Table 6.33: Performance metrics for Fig. 6.41

Scenario	t_{0R} (DFD)	t_{0N} (DFD)	t_{fN} (DFD)	$t_{fN} - t_{0N}$ $(days)$	$\left(\frac{m_f}{m_0}\right)_N$ $(\%)$
Fig. 6.41(a)	35.93	N/A	28.86	N/A	98.71
s/c Specifications for Nominal Trajectory					
$a_0 = 2.2 \times 10^{-4}$ m/s^2		$\widetilde{\mathbb{T}} = 0.22$ N		$\widetilde{Isp} = 4000$ s	

and $t_{fN} - t_{0N}$ in Table 6.33 are not-applicable because the recovery is initiated at $t_{0R} = 35.93$ $days$, which even exceeds the epoch (t_{fN}) at the terminal state on the nominal trajectory. An inspection of Fig. 6.41(b) illustrates that the s/c drifts away from the Earth-Moon system and does not return to the vicinity of the destination orbit (DRO) even after 2 $years$ if no corrective action is taken at the end of the forced coast segment. Fortunately, it is possible to leverage many natural solutions such as the natural families portrayed in Fig. 6.42 to prevent system escape and usher the s/c back to the DRO. The spatial natural families offer the arcs required

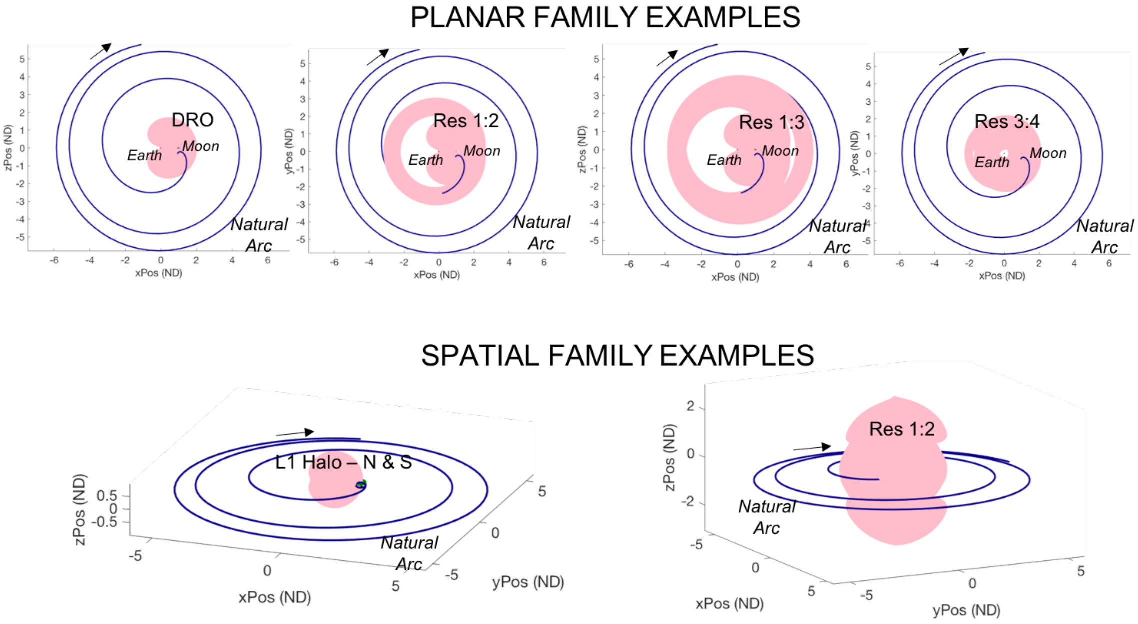


Figure 6.42: Examples of potential natural family arcs to support s/c recovery from the contingency scenario in Fig. 6.41.

to reduce the $\approx 25,000$ km out-of-plane s/c component to the plane and effect the required inclination change, while many planar resonant families span the extent of the configuration space traversed by a portion of the natural arc post the decision point (Fig. 6.42). Furthermore, since the JC is a function of both position and velocity magnitude, the similarity in JC color between the natural arc and certain members of the resonant families that it crosses (Fig: 6.43), indicates that the velocity magnitudes are also similar in these position overlap regions. Thus, the geometry between these structures is often comparable, as discussed in Section 5.2.2. A similarity in geometry presents an opportunity for a s/c on the natural arc post forced coast to 'catch' arcs belonging to the resonant families as they are more likely to appear in a s/c's instantaneously generated AR. During a manual design process, the choice of the natural orbit(s) or sequences of orbits to incorporate from the candidates in Fig. 6.42 are not immediately clear. It may also be time-consuming to manually explore various alternatives. However, with access to parallel computing resources, it is possible to incorporate many orbit families within the searchable database and direct the pathfinding agents to present attractive options from the episodic searches. Two outcomes resulting from such a search are presented in Fig. 6.44. The solution in Fig. 6.44(a) presents evidence of the pathfinding agents' search through conditions belonging to various families; in particular, arcs from six different families (planar and spatial) are sequenced to form the final transfer solution. It is also evident that

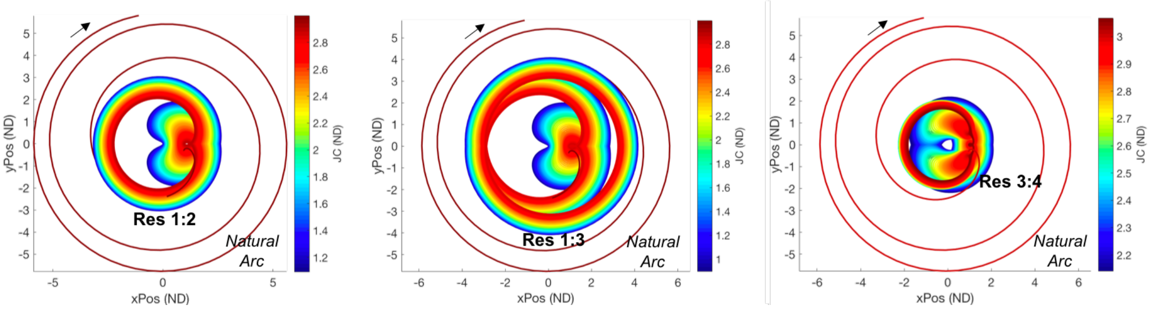
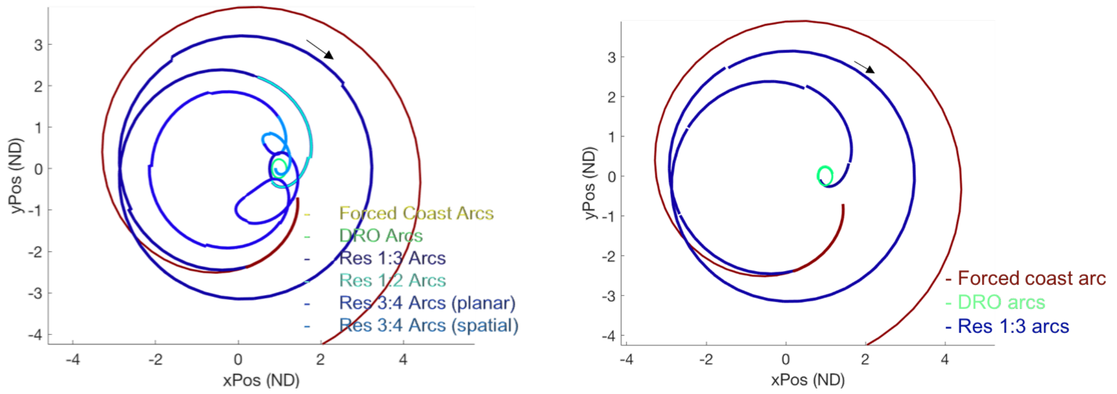


Figure 6.43: Examples of potential natural family arcs to support s/c recovery from the contingency scenario in Fig. 6.41, colored by JC values.



(a) Transfer IG Alternative #1 (b) Transfer IG Alternative #2

Figure 6.44: Sample recovery initial guesses for scenario in Fig. 6.41 for a s/c with $\widetilde{\mathbb{T}} = 0.11\ N$ from t_{0R}.

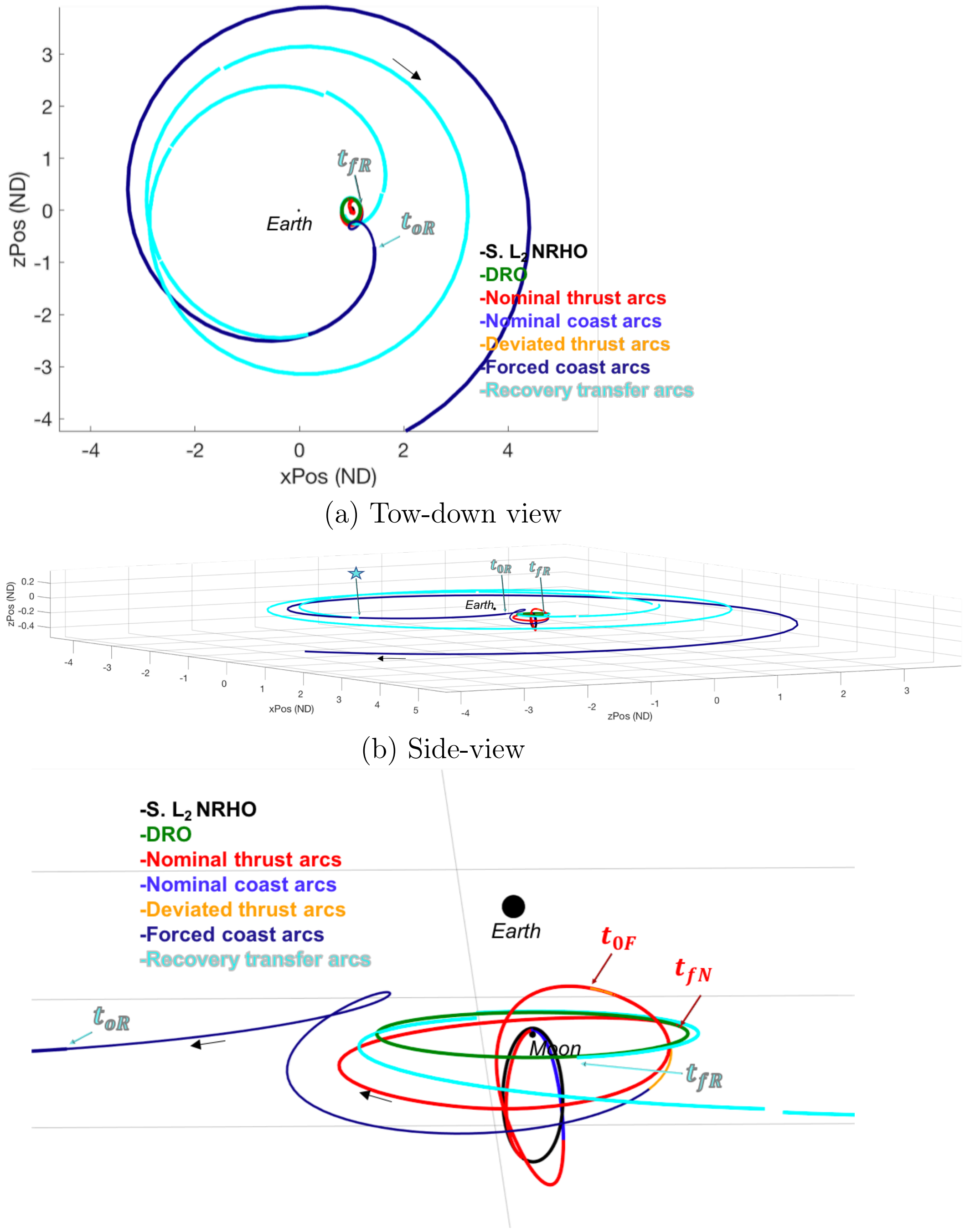
Table 6.34: Performance metrics for s/c recovery scenario in Fig. 6.44

Scenario	$t_{0R}\ (DFD)$	$t_{fR}\ (DFD)$	$t_{fR} - t_{0R}\ (days)$	$\frac{m_f}{m_0}_{est}\ (\%)$
Fig. 6.44(a)	35.93	210.93	175	96.71
Fig. 6.44(b)	35.93	112.93	77	98.33

<table>
<tr><td colspan="3">Natural Condition Transport Approach — <i>A Priori</i> Discretized Database</td></tr>
<tr><td colspan="3">s/c Specifications Pre-Failure</td></tr>
<tr><td>$a_0 = 2.2 \times 10^{-4}\ m/s^2$</td><td>$\widetilde{\mathbb{T}} = 0.22\ N$</td><td>$\widetilde{Isp} = 4000\ s$</td></tr>
<tr><td colspan="3">s/c Specifications Post-Forced Coast</td></tr>
<tr><td>$a_0 = 1.1065 \times 10^{-4}\ m/s^2$</td><td>$\widetilde{\mathbb{T}} = 0.11\ N$</td><td>$\widetilde{Isp} = 4000\ s$</td></tr>
</table>

the solution in Fig. 6.44(b) bears a resemblance to certain parts of the transfer in Fig. 6.44(a). That is, the pathfinding agents are able to filter arcs from complete end-to-end solutions to present improved alternatives for the user to choose from. The shorter duration solution in Fig. 6.44(b) 'catches' efficient sets of arcs to present a more propellant-efficient transfer; thus, this route is extracted for transition to a continuous low-thrust solution (Fig. 6.45).

The initial guess in Fig. 6.45(a) illustrates that the s/c veers away from the escaping natural (forced coast) propagation to remain bounded within the Earth-Moon system and gradually maneuvers towards the destination orbit. A comparison of fam-



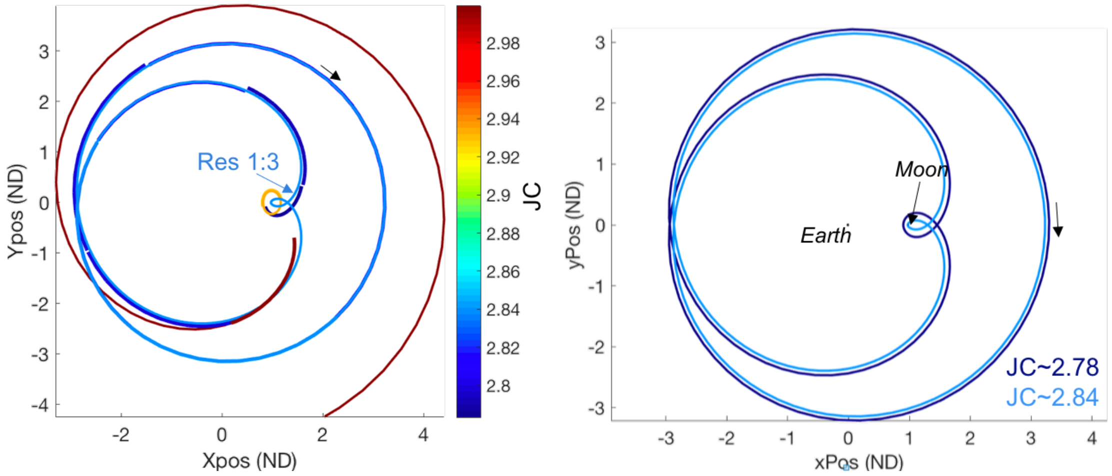
(a) Tow-down view

(b) Side-view

(c) Zoom-in view

Figure 6.45: Initial guess selected from options presented in Fig. 6.44

ily arcs in Fig. 6.44(b) and the location of the star in the spatial representation in Fig. 6.45(b) signifies the moment the s/c transitions from the forced coast arc and onto members belonging to the available family conditions in the searchable database. The agents thus exploit the forced coast segment that naturally transports the s/c to the plane. Figure. 6.45(c) displays a close-up of the s/c arrival at the DRO. The results are indicative of the sequencing of arcs from the Resonant 1:3 family in Fig. 6.44(b), explored in more detail in Fig. 6.46. Note that the Resonant 1:3 orbit at a $JC \approx 2.84$ is superimposed on the recovery path in Fig. 6.46(a) to highlight that most of the initial guess path follows the motion associated with this member of the Resonant 1:3 family. Some higher energy arcs from this family are leveraged to aid with the inclination and energy changes required at the initial stages of the recovery path and towards the arrival segment where a portion of the family member with $JC \approx 2.78$ better aligns with the destination orbit (DRO) geometry as seen from Fig. 6.46(b). The resulting mass-optimal continuous low-thrust solution is presented in Fig. 6.47.



(a) Resonant 1:3 orbit overlaid on IG arcs (b) Sample resonant orbits for IG arcs

Figure 6.46: (a) The Jacobi Constant (JC) history over the initial guess flight path in Fig. 6.45, and (b) Sample resonant orbits exploited for IG generation

Approximately 48 *days* of thrusting leads to a final mass fraction of 98.27%, close to the final mass prediction from the IG of 98.33% presented in Table 6.34. The TOF of 119 days is also close to the initial guess of ≈ 113 *days*. As explained with Fig. 6.46, the s/c thrusts initially to address inclination and energy changes as well as

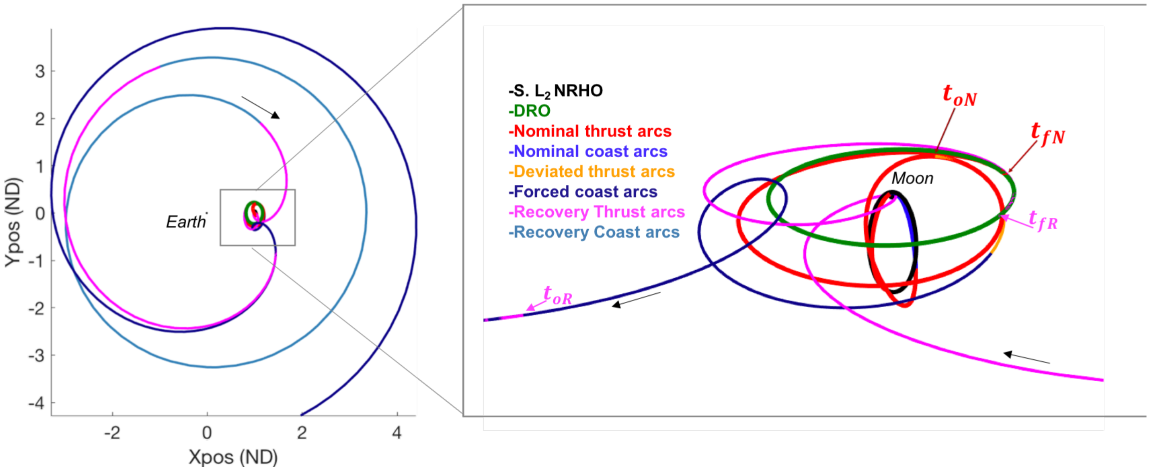


Figure 6.47: Low-thrust mass-optimal solution for initial guess presented in Fig. 6.45.

Table 6.35: Performance metrics for mass optimal CSI recovery in Fig. 6.47

Scenario	t_{0R} (DFD)	t_{fR} (DFD)	$t_{fR} - t_{0R}$ $(days)$	TD $(days)$	$\frac{m_f}{m_0}$ $(\%)$
Fig. 6.47	35.93	119.15	83.22	47.88	98.27
s/c Specifications Pre-Failure					
	$a_0 = 2.2 \times 10^{-4}\ m/s^2$		$\widetilde{\mathbb{T}} = 0.22\ N$	$\widetilde{Isp} = 4000\ s$	
s/c Specifications Post-Forced Coast					
	$a_0 = 1.1065 \times 10^{-4}\ m/s^2$		$\widetilde{\mathbb{T}} = 0.11\ N$	$\widetilde{Isp} = 4000\ s$	

towards the end to deliver the geometry required for orbit insertion. The continuous energy variations associated with the thrusting locations are captured in Fig. 6.48. The coasting regions represented by the constant navy blue color aid in maintaining the approach geometry required to transport the s/c to the lunar vicinity. The JC reduces and increases (energy increases and reduces) along the trajsfer path to account for the inclination and energy modulations required en-route to the DRO.

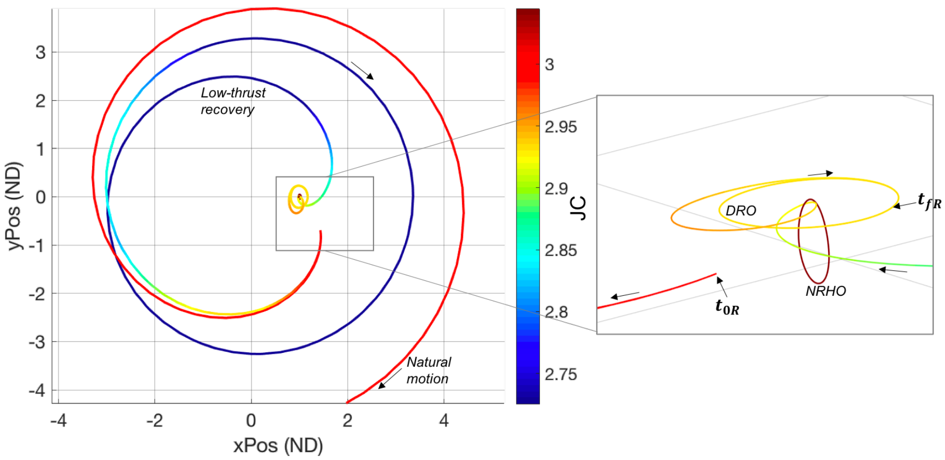


Figure 6.48: The Jacobi Constant (JC) history over the low-thrust flight path in Fig. 6.47.

6.7.2 Recovery to Perform Rendezvous on Arrival Orbit

The examples in Section 6.7.1 investigate the potential of the automated pathfinding algorithm to transport a s/c to the desired arrival orbit conditions. However, these examples do not incorporate rendezvous considerations that require the transport to be conducted within a strict TOF to meet a target s/c's states. Thus, to rendezvous, a global cost function that seeks to uncover the transfer with the least discrepancy in epoch between the chaser and target s/c at a given state (Eqn. (5.3)) is implemented. The failure scenario is the same as that which has been considered in Fig. 6.35(a) and is presented again in Fig. 6.49 for convenience. The un-failed low-thrust trajectory represents the nominal path traveled by the chaser s/c to meet a target s/c on the DRO. To induce a failure along this nominal path, the engine throughput is diminished from $\widetilde{\mathbb{T}} = 0.22\ N$ by 50% at $t_{0F} \approx 5\ days$ into the nominal transfer and is equivalent to losing the thrust capability of one engine (if there were two to begin with). Such a failure leads the s/c to drift from the nominal trajectory over $\approx 4\ days$, which is followed by a commanded forced coast at $t_{FC} \approx 8.86\ days$ for a duration of 3 days to simulate a safe mode recovery operation. The Decision Point (DP) indicates the epoch t_{0R}, at which the rendezvous recovery trajectory is initiated. Again, the terminology associated with the rendezvous scenarios examined in this section is summarized in Table 6.28. Given the assumption that a target s/c is already on the DRO and continues its path on this orbit, the goal of the chaser s/c is to begin its recovery at the DP (t_{0R}) and meet the target anywhere along the DRO with minimal error in the rendezvous epoch. That is, the ARs formed instantaneously during the pathfinding search inform the chaser s/c of of its ability to *reach* the target s/c's states, but does so with no consideration of the associated epoch of the target s/c's states. Then, the pathways that lead to minimal error in the rendezvous epoch are

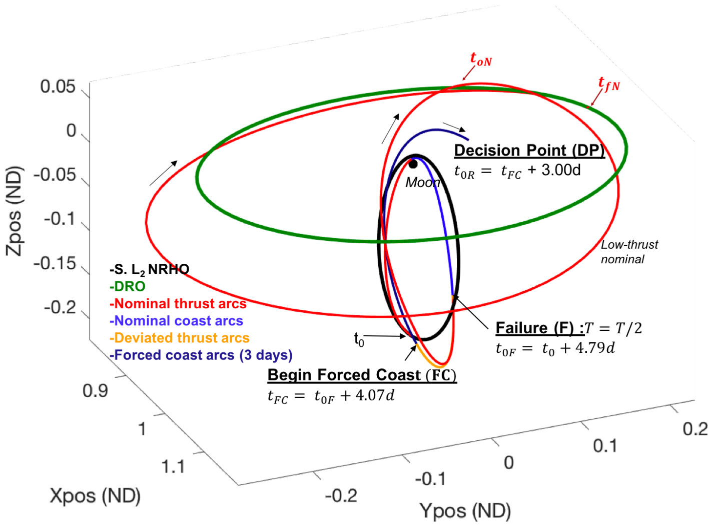


Figure 6.49: Contingency scenario example — Departure: NRHO (Periapsis Altitude = 1763.31 km) to DRO (JC: 2.935, Period: ≈ 13 $days$). Engine failure simulated ≈ 5 $days$ after departure from NRHO, and the trajectory deviates due to diminished engine throughput.

Table 6.36: Performance metrics for Fig. 6.49

Scenario	t_{0R} (DFD)	t_{0N} (DFD)	t_{fN} (DFD)	$t_{fN} - t_{0N}$ $(days)$	$\left(\frac{m_f}{m_0}\right)_N$ $(\%)$
Fig. 6.49(a)	11.86	11.86	28.86	17	98.71
s/c Specifications for Nominal Trajectory					
$a_0 = 2.2 \times 10^{-4}\ m/s^2$		$\widetilde{\mathbb{T}} = 0.22\ N$		$\widetilde{Isp} = 4000\ s$	

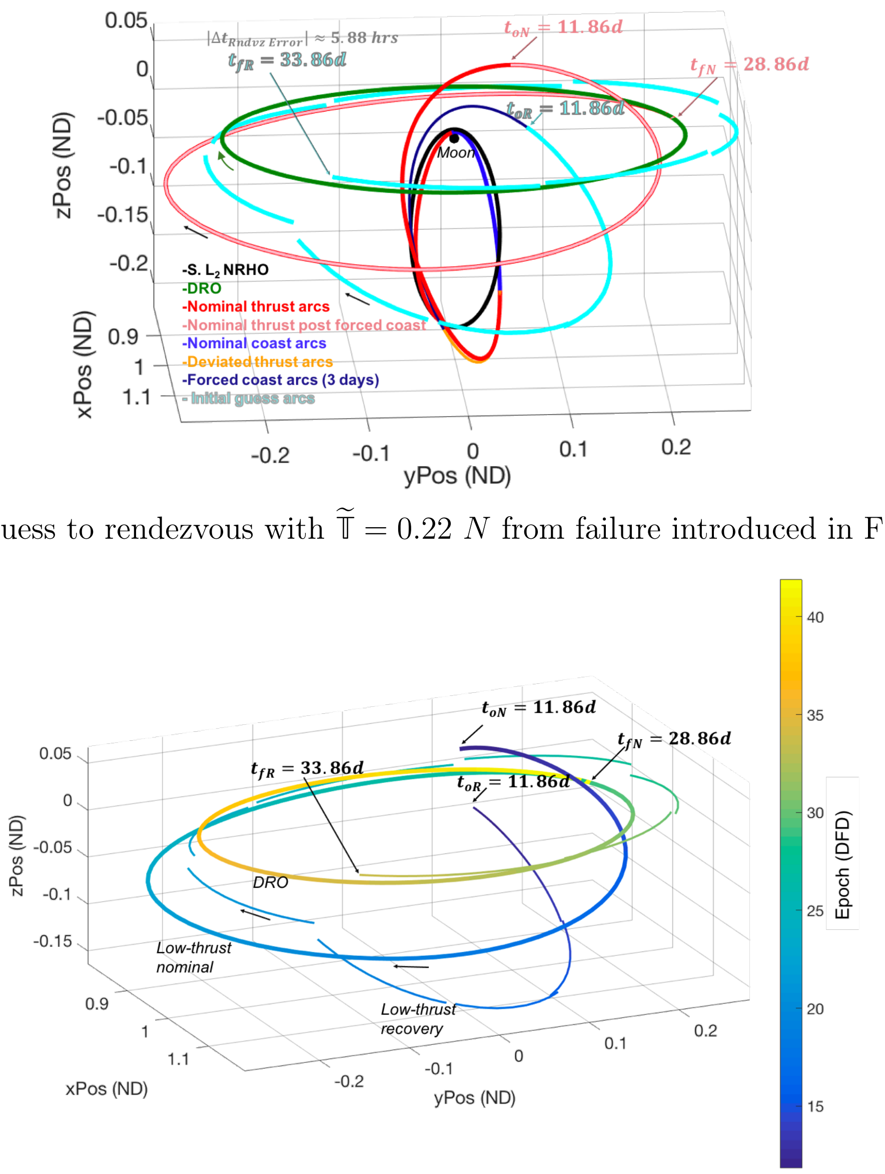
ranked according to the global cost function, and the most attractive options are reinforced. Such a design choice, where the tracking of the target states and epoch are considered independently of one another, is chosen such that the design framework only seeks to uncover an initial guess via the pathfinding agents that is then corrected by numerical techniques. Other design approaches where the states and epoch are considered jointly for rendezvous via machine learning is beyond the scope of this preliminary investigation.

The initial guess constructed by the agents when the chaser s/c is re-equipped with full thrust at t_{0R} is displayed in Fig. 6.50(a) The sequence of cyan transfer arcs in Fig. 6.50(a) presents an initial guess for a trajectory that delivers the chaser s/c to the target state to within 5.88 *hours* of the associated true epoch. For example, a particular target s/c state falls within the chaser s/c's AR at $t_{fR} = 33.86$ *days*. However, in reality, the target s/c is at this state at either $t_{fR} - 5.88$ *hours* or $t_{fR} + 5.88$ *hours*, and thus, the chaser s/c has not performed a 'true' rendezvous in the initial guess. Also, as seen in Fig. 6.50(b), the states on the chaser's nominal trajectory are not *reachable* by the chaser's ARs on the recovery path for a given epoch. That is, in this example, the large initial deviation between the nominal and recovery trajectories prohibit the chaser s/c from re-assuming the nominal path where it is able to rendezvous with the target on the DRO at the nominal terminal epoch of $t_{fN} = 28.86$ *days*.

Table 6.37: Performance metrics for s/c rendezvous scenario in Fig. 6.50

Scenario	t_{0R} (DFD)	t_{fR} (DFD)	$t_{fR} - t_{0R}$ $(days)$	$\frac{m_f}{m_{0\;est}}$ (%)
Fig. 6.44(a)	11.86	33.86	22	98.42

<table>
<tr><td colspan="3">Natural Condition Transport Approach — Free-Form Search.</td></tr>
<tr><td colspan="3">s/c Specifications Pre-Failure</td></tr>
<tr><td>$a_0 = 2.2 \times 10^{-4}\ m/s^2$</td><td>$\widetilde{\mathbb{T}} = 0.22\ N$</td><td>$\widetilde{Isp} = 4000\ s$</td></tr>
<tr><td colspan="3">s/c Specifications Post-Forced Coast</td></tr>
<tr><td>$a_0 = 2.205 \times 10^{-4}\ m/s^2$</td><td>$\widetilde{\mathbb{T}} = 0.22\ N$</td><td>$\widetilde{Isp} = 4000\ s$</td></tr>
</table>

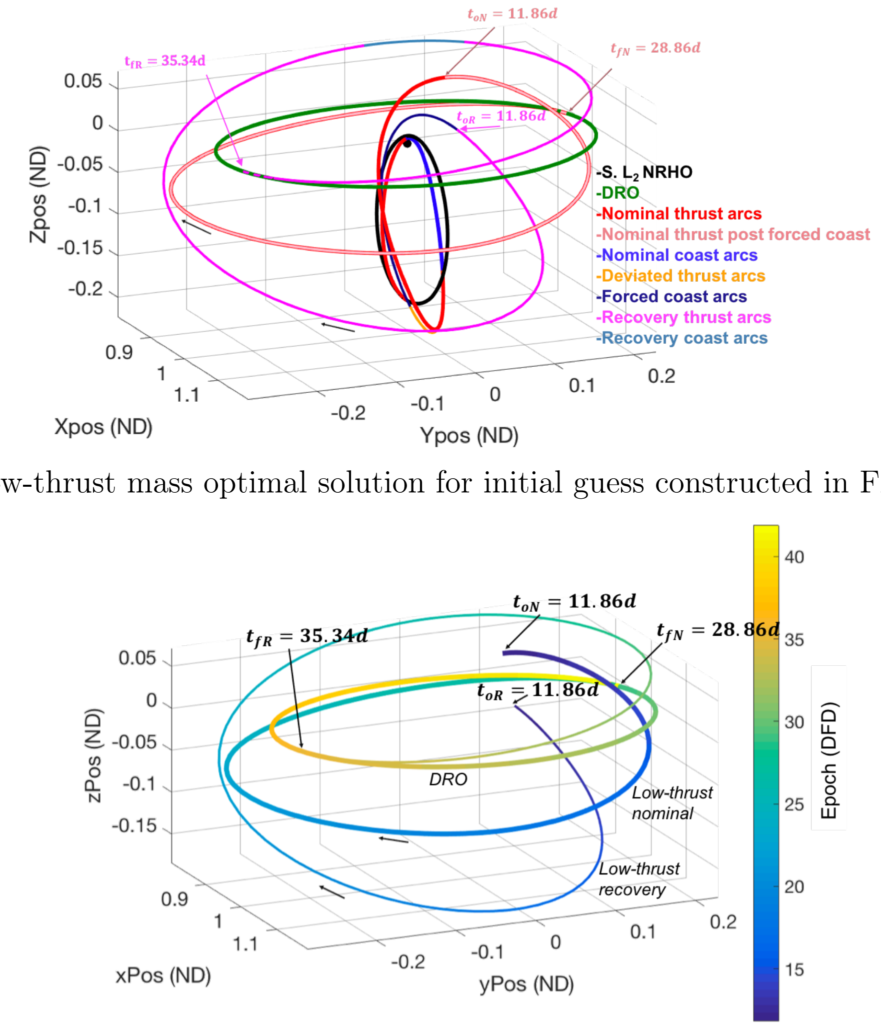


(a) Initial guess to rendezvous with $\widetilde{\mathbb{T}} = 0.22\ N$ from failure introduced in Fig. 6.49

(b) Time history along initial guess arcs in (a)

Figure 6.50: Sample rendezvous recovery initial guess for scenario in Fig. 6.49 for a chaser s/c with $\widetilde{\mathbb{T}} = 0.22\ N$ from t_{0R}. (a) Transfer arcs, (b) Transfer arcs colored by epoch (DFD)

The solution in Fig. 6.50 offers the state and epoch conditions required to initiate a numerical corrections process to eliminate the discontinuities. The targeting approach discussed in Section 3.2.2 is incorporated where the chaser meets the variable state and epoch of the target along the DRO. Note that, unlike the general rendezvous formulation introduced in Eqn. (3.111), the reference epochs at the departure condition and arrival orbit are not $t = 0$ in this example; these values are $t_{0R} = 11.86\ DFD$ and $t_{fN} = 28.86\ DFD$, respectively. Also, the departure epoch is held fixed for this recovery scenario such that $\varpi_{dep} = 0$. Thus, the last rendezvous time constraint in $\boldsymbol{F}_{CSI_{Rndzv}}$ in Eqn. (3.111) is modified to $\tau_{0R} + \sum_{i=1}^{\mathbb{n}-1} \varpi_i - \tau_{fN} - \varpi_{arr}$. Note that τ_{0R} and τ_{fN} are the nondimensional epochs associated with the quantities t_{0R} and t_{fN}, and are constant quantities. The thrust duration of 22.41 *days* is close to the



(a) Low-thrust mass optimal solution for initial guess constructed in Fig. 6.50

(b) Time history along mass-optimal rendezvous solution in (a)

Figure 6.51: (a) Low-thrust mass-optimal solution for initial guess presented in Fig. 6.50, and (b) Time history along mass-optimal low-thrust solution in (a).

Table 6.38: Performance metrics for mass optimal CSI rendezvous in Fig. 6.51

Scenario	t_{0R} (DFD)	t_{fR} (DFD)	$t_{fR} - t_{0R}$ $(days)$	TD $(days)$	$\frac{m_f}{m_0}$ (%)
Fig. 6.51	11.86	35.34	23.48	22.41	98.69
s/c Specifications Pre-Failure					
$a_0 = 2.2 \times 10^{-4}\ m/s^2$ $\quad \widetilde{\mathbb{T}} = 0.22\ N$ $\quad \widetilde{Isp} = 4000\ s$					
s/c Specifications Post-Forced Coast					
$a_0 = 2.205 \times 10^{-4}\ m/s^2$ $\quad \widetilde{\mathbb{T}} = 0.22\ N$ $\quad \widetilde{Isp} = 4000\ s$					

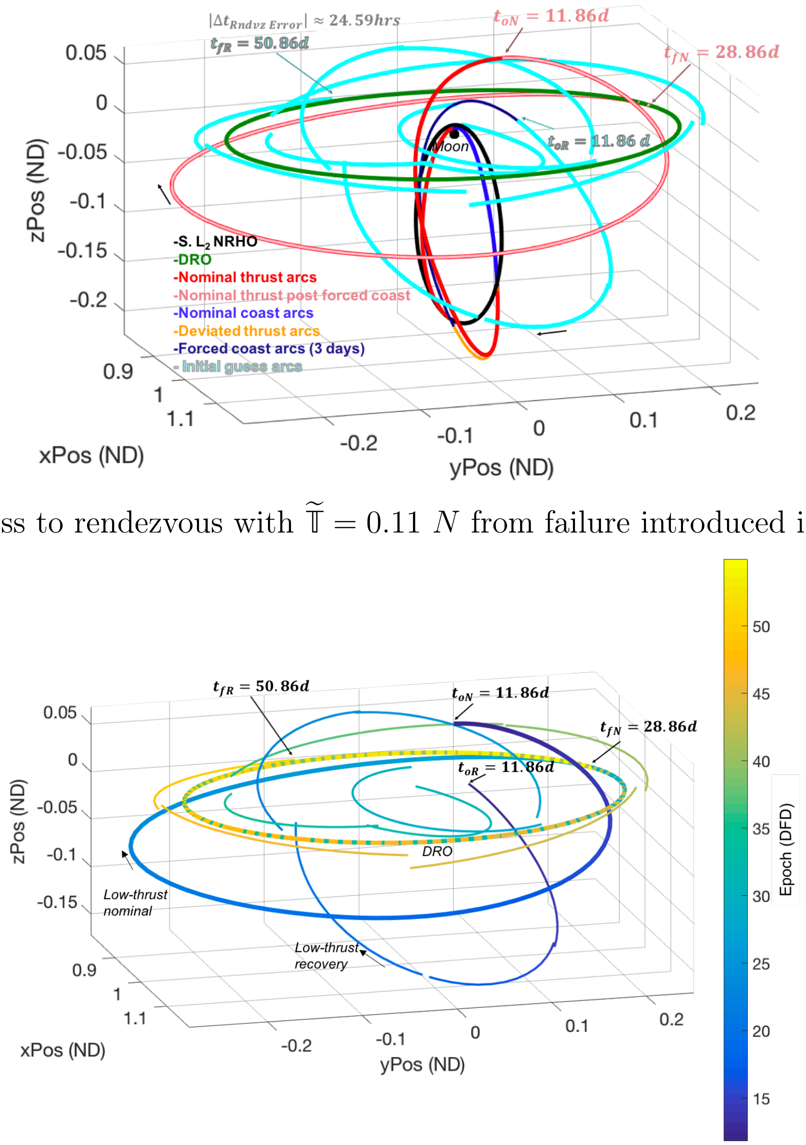
transfer time of 22 *days* estimated by the initial guess, and the mass-optimal transfer transports the chaser s/c to the target at 35.34 DFD with an additional ≈ 1 *day* of coasting time incorporated. The initial guess for the delivered mass-fraction is also a reliable estimate in this example. The evolution of time/epoch along the continuous low-thrust trajectory is illustrated in Fig. 6.51(b); this knowledge informs the user of an ≈ 1 *week* delay from the nominal epoch in reaching the target on the DRO due to the injected failure and the availability of a full recovery thrust magnitude of $\widetilde{\mathbb{T}} = 0.22\ N$ from t_{0R} onwards.

The rendezvous example explored in Fig. 6.50 is extended to a scenario where a diminished thrust magnitude ($\widetilde{\mathbb{T}} = 0.11\ N$) is maintained after t_{0R}. The recovery arcs transporting the chaser to the target are presented in Fig. 6.52(a). Much like the ≈ 36 *day* transfer in Fig. 6.38, a recovery with a lower thrust magnitude results in a longer TOF compared to that observed in the example in Fig. 6.50. A terminal

Table 6.39: Performance metrics for s/c rendezvous scenario in Fig. 6.52

Scenario	t_{0R} (DFD)	t_{fR} (DFD)	$t_{fR} - t_{0R}$ $(days)$	$\frac{m_f}{m_{0\ est}}$ (%)
Fig. 6.52	11.86	50.86	39	79.89

Natural Condition Transport Approach — Free-Form Search.
s/c Specifications Pre-Failure
$a_0 = 2.2 \times 10^{-4}\ m/s^2$ $\quad \widetilde{\mathbb{T}} = 0.22\ N$ $\quad \widetilde{Isp} = 4000\ s$
s/c Specifications Post-Forced Coast
$a_0 = 1.1025 \times 10^{-4}\ m/s^2$ $\quad \widetilde{\mathbb{T}} = 0.11\ N$ $\quad \widetilde{Isp} = 4000\ s$



(a) Initial guess to rendezvous with $\widetilde{\mathbb{T}} = 0.11\ N$ from failure introduced in Fig. 6.49

(b) Time history along initial guess arcs in (a)

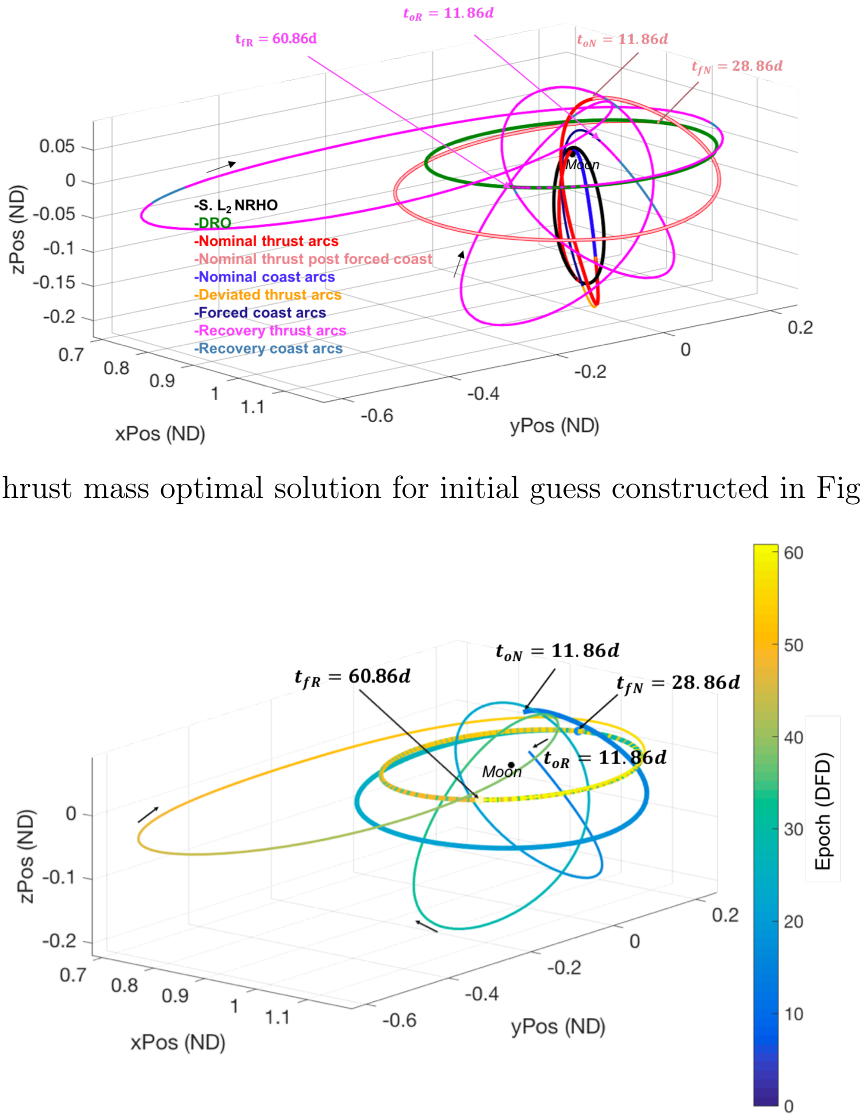
Figure 6.52: Sample rendezvous recovery initial guess for scenario in Fig. 6.49 for a chaser s/c with $\widetilde{\mathbb{T}} = 0.11\ N$ from t_{0R}. (a) Transfer arcs, (b) Transfer arcs colored by epoch (DFD)

epoch of $t_{fR} = 50.86\ DFD$ on the recovery trajectory occurs at $\approx 1.7\ revolutions$ from the nominal rendezvous epoch of $t_{fR} = 28.86\ DFD$ on the $\approx 13\ day$ period DRO. Thus, it is essential to prolong the look-ahead time from 1 period (as with the example in Fig. 6.50) to 2 periods to construct a solution for this rendezvous scenario.

The complicated nature of the transfer leads to a high rendezvous error of ≈ 1 *day* as illustrated in Fig. 6.52. Furthermore, maneuvers during close swing-bys of the Moon in this initial guess compute large ARs where the ability to select states that are far apart leads to large discontinuities between the arcs and contributes to the low estimate for the final delivered mass fraction in Table 6.39. Also, although permitting selection of pre-reinforced states in the vicinity of an AR as discussed earlier may lead to greater state discontinuities, this approach aids in minimizing the rendezvous epoch error with the diminished thrust capability. Nevertheless, the initial guess offers all the state and epoch information to compute a successful continuous low-thrust rendezvous solution as illustrated in Fig. 6.53. The mass-optimal trajectory incorporates an additional 10 *days* to the initial guess estimate of the TOF to account for the large state and epoch discontinuities and deliver the chaser s/c to the target s/c at $t_{fR} = 60.86$ *days* from departure. The extended and elongated deviation of the chaser s/c from the DRO as seen in Fig. 6.53(b) aids in the phasing required to perform the rendezvous as well. The time-evolution knowledge along this recovery path informs the user of an ≈ 1 *month* delay from the nominal epoch in reaching the target on the DRO due to the injected failure and the constricted thrust magnitude of $\widetilde{\mathbb{T}} = 0.11\ N$ from t_{0R} onwards.

Table 6.40: Performance metrics for mass optimal CSI rendezvous in Fig. 6.53

<table>
<tr><th>Scenario</th><th>$t_{0R}\ (DFD)$</th><th>$t_{fR}\ (DFD)$</th><th>$t_{fR} - t_{0R}\ (days)$</th><th>$TD\ (days)$</th><th>$\frac{m_f}{m_0}$ (%)</th></tr>
<tr><td>Fig. 6.53</td><td>11.86</td><td>60.86</td><td>49</td><td>45.96</td><td>98.66</td></tr>
<tr><td colspan="6">s/c Specifications Pre-Failure</td></tr>
<tr><td colspan="6">$a_0 = 2.2 \times 10^{-4}\ m/s^2$ $\widetilde{\mathbb{T}} = 0.22\ N$ $\widetilde{Isp} = 4000\ s$</td></tr>
<tr><td colspan="6">s/c Specifications Post-Forced Coast</td></tr>
<tr><td colspan="6">$a_0 = 1.1025 \times 10^{-4}\ m/s^2$ $\widetilde{\mathbb{T}} = 0.11\ N$ $\widetilde{Isp} = 4000\ s$</td></tr>
</table>



(a) Low-thrust mass optimal solution for initial guess constructed in Fig. 6.52

(b) Time history along mass-optimal rendezvous solution in (a)

Figure 6.53: (a) Low-thrust mass-optimal solution for initial guess presented in Fig. 6.52, and (b) Time history along mass-optimal low-thrust solution in (a).

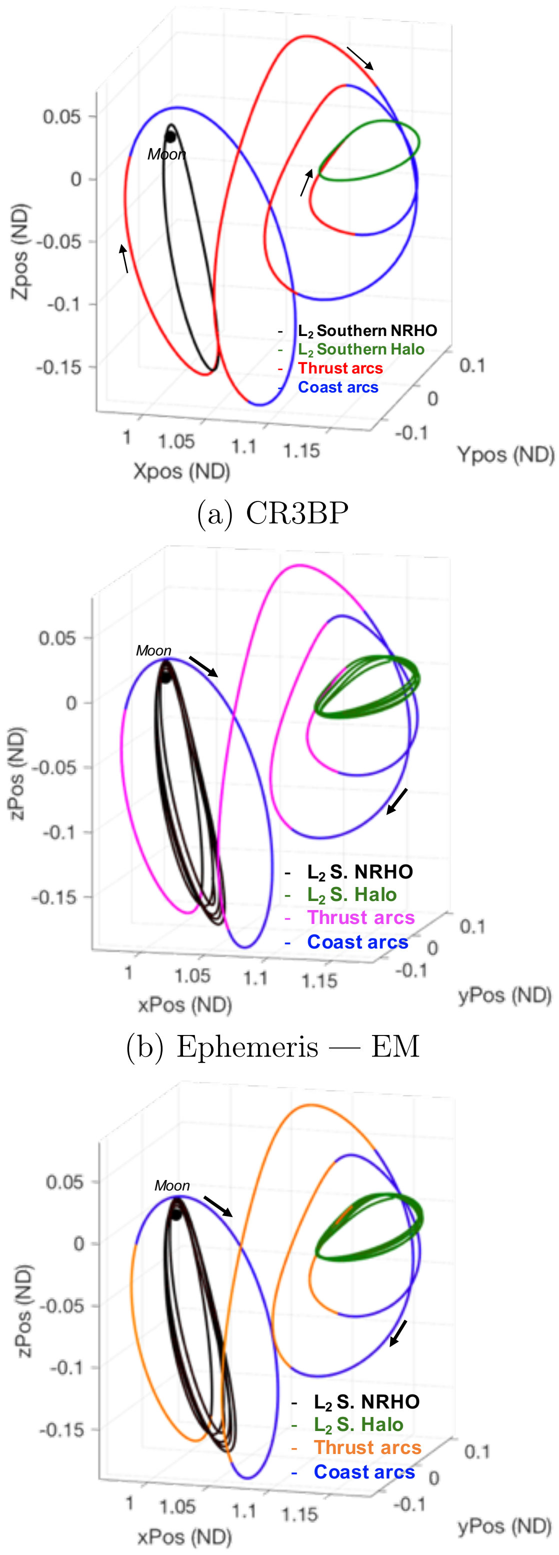
6.8 Transition of CR3BP Solutions to the Ephemeris Model

The general dynamics of a 3B system are captured during transfer design when a solution is constructed in the CR3B model. However, this solution is viable only if it enables an end-to-end trajectory that satisfies mission requirements in the epoch-dependent ephemeris model as well. Since the solution characteristics are expected to vary based on epoch considerations, this investigation does not search for optimal solutions in the ephemeris regime. The focus is maintained on whether the CR3BP optimal transfer can be converged to a solution in the ephemeris regime, observing the similarities in the resulting epoch-dependent transfer characteristics. A select number of examples are chosen as a proof-of-concept. The transition of each CR3BP transfer to the ephemeris regime is initiated by stacking several revolutions of the departure and arrival orbits and connecting these stacks via the CR3BP mass-optimal low-thrust transfer trajectories as explored in Section 3.3. Then, the combined trajectory sequence is passed through a numerical corrections process that targets state and epoch continuity within the ephemeris model. The acronym EM represents an ephemeris regime where the gravities of the Earth and the Moon are incorporated, while EMS represents the regime where those of the Earth, Moon, and Sun are incorporated. Although the numerical corrections process is undertaken with respect to an inertial frame and the ephemeris regime, the solutions are plotted in the CR3BP rotating frame for illustrative purposes.

The southern L_2 NRHO to southern L_2 planar halo transfer scenario discussed in Fig. 6.25 is replotted in Fig. 6.54(a) for convenience and is transitioned successfully to two ephemeris models as shown in Fig. 6.54(b) and (c). A similar TOF and

Table 6.41: Performance metrics for CSI converged solutions in the ephemeris model in Fig. 6.54

Scenario	TOF (days)	$\frac{m_f}{m_0}$(%)
Fig. 6.54(a): CR3BP	48.48	98.70
Fig. 6.54(b): EM — 01 Jan 2020	45.29	98.86
Fig. 6.54(c): EMS — 01 Jan 2020	45.30	98.86
s/c Specifications		
$a_0 = 2.2 \times 10^{-4}\ m/s^2$ $\quad \widetilde{\mathbb{T}} = 0.22\ N$ $\quad \widetilde{Isp} = 4000\ s$		



(a) CR3BP

(b) Ephemeris — EM

(c) Ephemeris — EMS

Figure 6.54: Southern L_2 NRHO to Southern L_2 halo — transition of CR3BP interior transfer to the ephemeris model and the assessment of the influence of various gravitational bodies on this transfer.

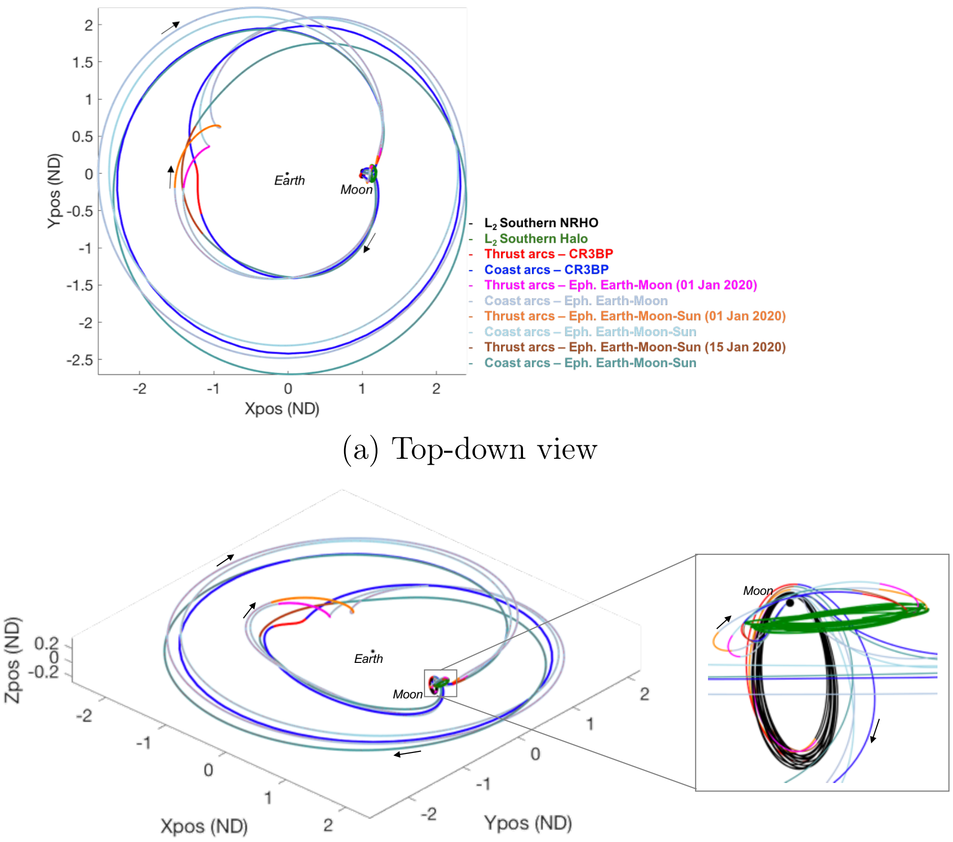
delivered mass fraction is evident in Table 6.41 for all the three cases investigated. Lower values of the mass fraction represent greater thrust durations and, so, greater propellant consumption along the trajectory. It is important to note that there is no strict requirement for the departure and arrival conditions in these ephemeris solutions to match the CR3BP departure and arrival orbit conditions; i.e., only continuity in the states is enforced, even at the boundaries. Since the s/c moves in the vicinity of the Moon for this particular transfer scenario, the transfer profile that is shaped by the primary gravitational influences of the Earth and the Moon in the CR3BP is maintained in the ephemeris solutions for the selected epoch as well. Accordingly, it is also interesting to investigate a transfer scenario that travels exterior to the Earth-Moon system and is potentially more sensitive to the gravitational influence of the Sun (Fig. 6.55).

The transfer studied in Fig. 6.25(c) serves as an example to study the influence of the epoch-dependent gravitational forces on a s/c that travels exterior to the Earth-Moon system prior to orbit insertion. Similar mass fractions represent similar thrust durations along the transfers for all four scenarios. Although the general transfer geometry is maintained, the trajectories in this figure vary based on the gravitational bodies incorporated in the ephemeris model and the choice of the departure epoch.

Table 6.42: Performance metrics for CSI converged solutions in the ephemeris model in Fig. 6.55

Scenario	TOF (days)	$\frac{m_f}{m_0}$ (%)
Fig. 6.55(a): CR3BP	131.32	98.84
Fig. 6.55(b): EM — 01 Jan 2020	130.20	98.84
Fig. 6.55(c): EMS — 01 Jan 2020	130.24	98.84
Fig. 6.55(d): EMS — 15 Jan 2020	130.41	98.82

s/c Specifications		
$a_0 = 2.2 \times 10^{-4}\ m/s^2$	$\widetilde{\mathbb{T}} = 0.22\ N$	$\widetilde{Isp} = 4000\ s$



(a) Top-down view

(b) Side-view

Figure 6.55: Southern L_2 NRHO to Southern L_2 halo — transition of exterior CR3BP transfer to ephemeris model, and observation of the influence of various gravitational bodies and the impact of variation in departure epoch on the transfer.

Also, as with Fig. 6.54, the departure and arrival states are not directed to start from and terminate at a particular state and the TOF is also free to vary; thus, these factors also impact the final converged geometry and performance. The CR3BP and the EMS - 15 Jan 2020 transfer scenario exhibit similar geometries given a similar arrival condition and similar TOF. The other two scenarios exhibit similar geometries to each other for the same reason.

The outcomes associated with constraining the boundary conditions is examined in the transfer scenario in Fig. 6.56. Note that the departure and arrival orbits in this figure are solutions from the CR3BP and not their counterparts propagated in the ephemeris model and rotated to the CR3BP as computed in the prior figures. Here, these orbits are plotted only to offer reference for the transfer segment of the solution. Both the CR3BP and ephemeris solutions depart from the same inertial state

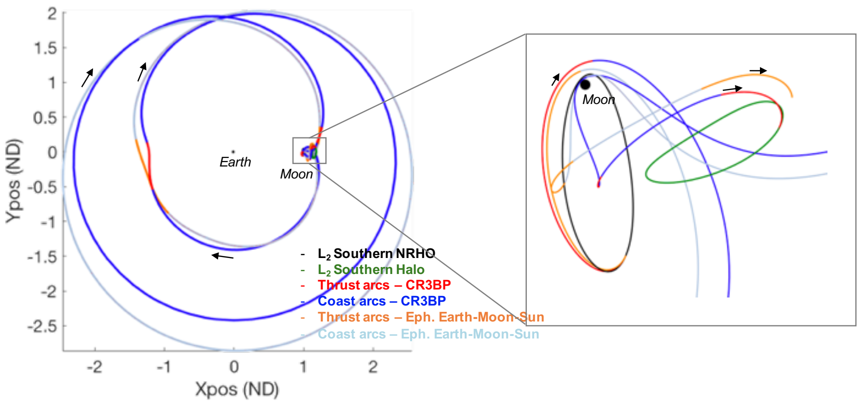


Figure 6.56: Southern L_2 NRHO to Southern L_2 halo — transition of exterior CR3BP transfer to ephemeris model, and observation of the gravitational influence of the Earth, Moon, and Sun as well as the effect of imposing fixed inertial boundary conditions subject to free TOF.

Table 6.43: Performance metrics for CSI converged solutions in the ephemeris model in Fig. 6.56

Scenario	TOF (days)	$\frac{m_f}{m_0}$ (%)
Fig. 6.56(a): CR3BP	131.32	98.84
Fig. 6.56(b): EMS — 01 Jan 2020	130.06	98.65
s/c Specifications		
$a_0 = 2.2 \times 10^{-4}\ m/s^2$	$\widetilde{\mathbb{T}} = 0.22\ N$	$\widetilde{Isp} = 4000\ s$

and epoch. At arrival, the same inertial state is targeted, but the TOF is allowed to vary (Table 6.43). So, the s/c terminates at a different state in the E-M rotating frame as seen in Fig. 6.56. Such a design decision carries implications for the mission outcomes and so careful consideration of the design variables is important. As seen in Fig. 6.57, when the TOF is also fixed, the s/c is able to recover the terminal state from the CR3BP solution. Similar to Fig. 6.56, the arrival and departure orbits in this figure are propagated under the assumption of the CR3BP and are plotted only to offer reference for the transfer segment. Even though the boundary conditions

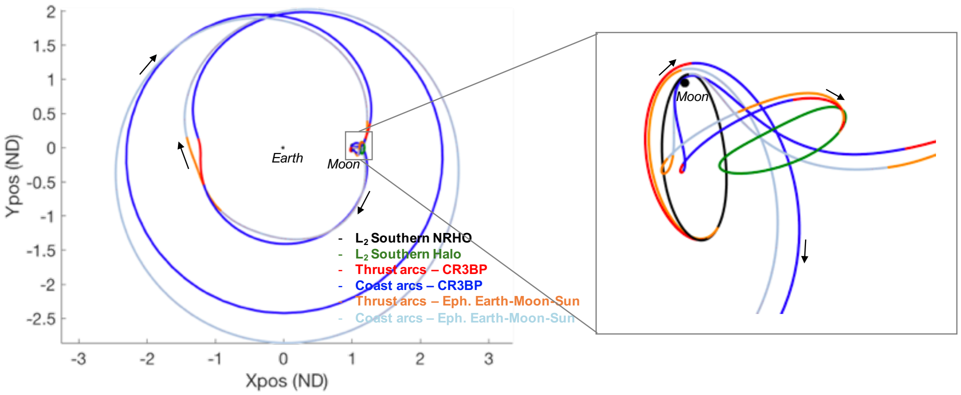


Figure 6.57: Southern L_2 NRHO to Southern L_2 halo — transition of exterior CR3BP transfer to ephemeris model, observing the gravitational influence of the Earth, Moon, and Sun as well as the effect of imposing fixed boundary conditions and fixed TOF.

Table 6.44: Performance metrics for CSI converged solutions in the ephemeris model in Fig. 6.57

Scenario	TOF (days)	$\frac{m_f}{m_0}$ (%)
Fig. 6.57(a): CR3BP	131.32	98.84
Fig. 6.57(b): EMS — 01 Jan 2020	131.32	98.53

s/c Specifications		
$a_0 = 2.2 \times 10^{-4}\ m/s^2$	$\widetilde{\mathbb{T}} = 0.22\ N$	$\widetilde{Isp} = 4000\ s$

and TOF are fixed, the time between the nodes are allowed to vary to incorporate flexibility into the numerical corrections process. The epoch-dependent gravitational influence of the Earth, Moon, and Sun result in slightly varied geometries and delivered mass fractions between the CR3BP and the ephemeris solutions. Also, note that the ephemeris transfer profile is expected to be different between this example where the boundary conditions are fixed and the solution for 01 January 2020 in Fig. 6.55, where only state continuity is sought at the boundaries. Thus, the examples in this section illustrate the ability to successfully transition the CR3BP solutions that are themselves computed via initial guesses constructed by the pathfinding agents to an ephemeris model with varied selections for a number of gravitating bodies and epoch.

7. CONCLUDING REMARKS

Trajectory design is a careful balance that juggles diverse constraints, priorities, and requirements to enable successful missions. Forthcoming efforts to expand the boundaries of space exploration by initiating an efficient cis-lunar transportation infrastructure necessitate a rigorous mission design framework to offer solutions for varied spacecraft classes and objectives. The CR3BP model serves as an appropriate model for capturing the complex dynamical interactions in the Earth-Moon system, facilitating natural pathways for preliminary design. However, in contrast to the two body problem, the CR3BP represents a large state space for which closed-form solutions and simplifying approximations are generally not available. Thus, an infinitely large combinatorial optimization problem emerges — one that is intractable to explore thoroughly via manual approaches alone. Thus, an automated search strategy is pursued where human efforts are refocused on defining the broader mission goals and software agents are tasked with undertaking the laborious task of filtering various trajectory scenarios to deliver the most attractive transfer options.

7.1 Investigation Summary

The design of end-to-end trajectories is facilitated by the construction of a framework that is comprised of four essential components: (I) simulation of the *reach* of the s/c to assess instantaneous accessible regions (ARs) within the dynamical regime; (II) exploitation of natural flows in the dynamical regime (CR3BP) to serve as potential waypoints for transfer paths; (III) implementation of automated pathfinding algorithms to sequence natural arcs from the ARs and formulate a discontinuous, yet complete, route to the destination; and, (IV) transition of the initial guess to a continuous solution via a numerical corrections process. The Artificial Intelligence (AI) strategies incorporated in the framework facilitate rapid multi-dimensional pattern-recognition and rapid decision making to uncover a variety of routes as initial guesses for an end-to-end transfer. Spacecraft state, time and engine thrusting information are also readily extractable from such initial guesses to initialize traditional numerical corrections processes to deliver the final solution. The CR3BP continuous and optimized solutions are then successfully transitioned to a higher-fidelity ephemeris regime to assess their paths under the time-varying gravitational influences of the Sun, Earth, and Moon. The processes adopted to develop the four framework components and the insights gained into the resulting capabilities are explored.

7.1.1 Framework Component — Accessible Regions

Knowledge of spacecraft (s/c) engine specifications (thrust magnitude and specific impulse) are exploited to approximate the unique region of space that is *accessible* at the end of a burn arc over a fixed time-horizon. This bounded volume of space, termed the s/c *Accessible Region* (AR), is computed by the explicit propagation of velocity-perturbed states for a pre-specified duration under the influence of the nonlinear dynamics of the CR3BP. Thus, the size of the manageable velocity perturbation and length of the time-horizon contribute to the nature of the computed AR. Consequently, spacecraft that are capable of delivering larger maneuver magnitudes (e.g., chemical versus low-thrust) engender larger ARs for a fixed time-horizon that then facilitate access to a broader set of natural states for selection by the pathfinding agents. Since the span of the ARs reflect the performance capabilities of a particular s/c, its incorporation during solution construction informs propulsion system trades for the scenario of interest as well. Exploitation of the ARs aid in shrinking an otherwise infinitely large search space that is often required to undertake a broad survey of scenarios for a particular transfer.

7.1.2 Framework Component — Natural Conditions

In this investigation, alternative approaches are explored to incorporate natural arc segments to complete an end-to-end transfer. A free-form search based solely on system dynamics instantaneously generates natural states within an AR for s/c selection. Searches adopting this scheme in a higher-dimensional spatial realm benefit from seeding waypoints generated via reverse-time AR computations from the destination. The free-form approach facilitates an unconstrained search that broadens the design options via traversal of both chaotic and ordered motion, offering insights into the dynamical regime including natural structures for more detailed analysis in support of the particular transfer scenario. In contrast, the incorporation of *a priori* known structures aids in restricting the search-space to desired regions of the configuration space, transfer geometries, and energy levels. Known natural structures are implemented via: (a) introducing pre-discretized orbit families or (b) employing supervised learning techniques such as Artificial Neural Networks/Support Vector Machines to exploit the pattern recognition capabilities and develop 'flow-models' to recognize the approximate dynamical behavior associated with these families. Statistical techniques involving posterior probabilities inform the level of confidence associated with the predictions of these machine learning tools. The flow-models liberate the pathfinding agents from restriction to waypoints defined within the pre-discretized catalog and, thus, mitigate challenges associated with the sensitivity of a solution to the gradation of the discretization scheme. This supervised learning approach does, however, require additional upfront time investment to train the desired 'flow-models'. In this investigation, for similar implementation and computational efforts, the SVM approach has proven superior to the ANN implementation for pattern recognition. Additional efforts to modify the ANN architecture to improve the associated results have not been attempted in this preliminary investigation.

7.1.3 Framework Component — Pathfinding Schemes

The pathfinding phase serves to forge a sequence of AR frontiers by the selection of advantageous states from within each AR, to connect the departure and destination conditions. The selection of the specific automated pathfinding technique is influenced by the desired aims and the computational resources available; both an exact and heuristic approach are examined. The *exact* approach via Dijkstra's algorithm offers globally optimal solutions for the user-defined cost function by tracking the least-cost pathway from departure at any given instant in time. Such an effort requires the maintenance of a record of the details concerning multiple pathways and is an uninformed search in that it that lacks *a priori* knowledge of the destination until it is reached. These factors result in a relatively inefficient search strategy via Dijkstra's algorithm for the current analysis. Furthermore, the finer the discretization of the searchable database, the greater the time and space/storage complexity that is introduced — limiting the number of periodic orbit families incorporated in the search and, thus, limiting the broad trade-space analysis. The non-parallelizable search components associated with the implementation of Dijkstra's algorithm also contribute to increased computation times. In contrast, although the Heuristically Accelerated Reinforcement Learning (HARL) algorithm may not deliver global optimality due to its stochastic nature and use of heuristics, the parallelizable computational aspect permits a large number of searches that then deliver initial guesses suitable for further manipulation using numerical techniques to arrive at varied local optima. The HARL algorithm is implemented as a distributed and cooperative network of agents that strive to forge end-to-end routes via 'exploration' and 'exploitation' phases over multiple episodes. During 'exploration', the agents perform random actions in the design space to *learn* the consequences of actions from a given state. Eventually, the agents embark on an 'exploitation' phase where the knowledge accumulated during 'exploration' is leveraged to further refine the trade-space. During 'exploitation', a heuristic function influences the selection of valuable states from within a given AR.

The heuristic function in this investigation employs knowledge of the distance of states within an AR to the final destination in both the position and velocity dimensions while also accounting for the amount of propellant required to reach the AR states from the s/c's current state. This measure of *goodness* of these states is combined with knowledge of their accumulated rewards over time to present a probabilistic distribution of their selection likelihood and, thus, guides a pathfinding agent's decision process for state selection from an AR during 'exploitation'.

7.1.4 Framework Component — Traditional Optimization Techniques

The AR, natural transport, and pathfinding components of the design framework construct a sequence of discontinuous natural arcs that satisfy the global objective function. The discontinuities in position and velocity (when $\Delta V's$ are not allowed) are eliminated by passing the pathfinding agents' natural arc sequence as an initial guess to traditional nonlinear programming solvers. Both direct and hybrid optimization schemes are leveraged as applicable, to support the convergence efforts for transfers using chemical, Variable Specific Impulse (VSI), and Constant Specific Impulse (CSI) low-thrust spacecraft. All the essential inputs for initiating an iterative numerical corrections process such as the position, velocity, and mass states as well as thrust magnitude, direction, and time along the transfer path are all readily available as a result of the arc sequence uncovered by the agents. Path and boundary constraints are enforced during the corrections process to transform the discontinuous sequence to a continuous end-to-end transfer in the CR3BP. The CR3BP solutions then prove to be suitable initial guesses to seed the ephemeris corrections process that accounts for the higher-fidelity time-varying dynamics. Advantageous thrust and coast locations emerge due to the freedom afforded in the search scheme to construct transfer segments with discontinuities in position and velocity states between them during the pathfinding phase.

7.1.5 Demonstrated Framework Capabilities

The results from this analysis demonstrate the feasibility and diverse capabilities of the automated initial guess generation design framework in both the planar and spatial realms. With relevance to a cis-lunar architecture, examples illustrate the potential of the framework to support the design of complex transfer architectures for space transportation in the near future. The pathfinding agents are able to construct transfers between orbits of varying natures of stability in addition to arbitrary initial conditions near a primary that could represent the variable deployment state of a spacecraft. The geometry of the solutions are influenced by constraining the types of states permitted within an AR and manipulating the design parameters. The incorporation of intermediate bridging families that bear geometrical similarities to both the departure and arrival orbit families is also demonstrated to be beneficial in solution construction. The search strategy exhibits the ability to make intelligent choices based on user preferences, such as exploiting lunar flybys when applicable to preserve propellant consumption. Differing implementations also render the methodology flexible to the incorporation of varied constraints during the design process, e.g., targeting specific arrival conditions and maintaining a minimum altitude relative to a primary.

The goals of the pathfinding process are pursued by the definition of an overarching global objective. For example, the TOF-payload delivery trade-space is explored by posing a weighted mass-time priority global cost function that, for example, favors longer but more propellant-efficient transfers when payload mass delivery is prioritized. Alternatively, a global cost function to minimize the time-to-rendezvous error uncovers initial guesses for routes that are numerically corrected to enable a chaser s/c to rendezvous with the states and absolute time of a target object in its destination orbit. Applicable to both the planar and spatial realms, the mass-time performance reports from the sequencing of natural arcs have, in general, proven to be reliable estimates of the performance metrics following the numerical corrections/optimization process.

The ability to generalize the design capability across various engine models (low-thrust and chemical) is realized by incorporating spacecraft particular specifications when establishing its reach via ARs. The design framework is, thus, beneficial in envisioning the impact of technology developments for enhanced mission throughput as well. The flexibility to introduce varied thrust profiles along a particular initial guess also supports extraction of multiple transfer geometries and local minima. The capacity to explore a large trade space and test a variety of combinatorial options offsets common restrictions, such as a narrow convergence radii associated with some traditional trajectory design approaches.

7.2 Conclusions — Implications of AI-Aided Trajectory Design

The AI techniques surveyed in this investigation offer various strengths that mitigate the challenge of conducting broad manual searches through infinite transfer options in a complex dynamical regime given limited analysis time as well as human resources. For example, (i) the mission-objective dictated Reinforcement Learning (RL) searches facilitate a methodical and quantitative approach to forging and ranking multiple transfer paths; (ii) parallel computing capabilities via the RL paradigm also enable rapid and efficient broad searches; and, (iii) the pattern-recognizing flow-models developed via supervised learning facilitate the blending of human intuition and experience with automated searches during preliminary design. The proposed architecture offers a trade-off between long-term efficiency and the up-front time investment required for the assembly of design-aiding AI elements such as the training of flow-models. The one-time and up-front efforts to architect the AR computations and flow-model training then afford the flexibility to accommodate varied s/c specifications and natural flows to enable rapid trade-space evaluations for varied mission applications. Likewise, the flow-model training is built upon pre-existing databases of known orbit families in the CR3BP; so the models can be stored alongside these already existing databases for easy future access for a multitude of applications beyond those investigated in this analysis. Furthermore, the means to automate the route-finding task via AI alleviates human effort related to laborious combinatorial searches, permits rapid implementation of shifting mission requirements, and facilitates an accelerated understanding of the implications of various design choices — all leading to well-informed trajectory construction and decision-making for complex mission scenarios. Such strengths render AI-aided trajectory design as a promising new design approach in the field of astrodynamics.

7.3 Recommendations for Future Work

This preliminary investigation opens many additional avenues of research related to the blending of artificial intelligence techniques with traditional astrodynamics design efforts. Improvements to various components of the design framework and further research endeavors are highlighted as follows.

1. *Accessible Region*

   Currently, both chemical and low-thrust spacecraft states are perturbed by small chemical impulses and are propagated forward in time to construct the ARs. This choice facilitates a modular architecture to demonstrate the functionality of the design framework for various s/c types in this preliminary investigation. For future efforts focused on low-thrust applications alone, incorporating the true operation of the ion engine during the initial guess generation phase is a key next step by allowing thrust magnitude variations that are bounded, followed by propagation of the full low-thrust equations of motion to compute the ARs. Furthermore, the true ARs are currently approximated by simple geometric shapes as highlighted in Section 5.1. However, such approximations may inadequately capture the true regions that are accessible by a s/c when it moves in the highly nonlinear dynamics in the vicinity of a primary. Thus, it is beneficial to pursue measures to better capture s/c ARs in such nonlinear areas of the configuration space.

2. *Machine Learning*

   The design choices associated with the supervised learning and Reinforcement Learning (RL) techniques employed in this investigation serve to provide proof-of-concept for the ability to construct initial guesses via the design framework. However, there are numerous avenues to improve the fidelity of predictions associated with the supervised learning techniques and performance associated with HARL. For example, experimentation with different ANN architecture choices, e.g., the number of hidden layers and neurons, as well as the type of activation functions incorporated to capture the underlying relationships in the data and the optimization algorithms employed to minimize prediction error via backpropagation are all valid pursuits; varied choices for the kernel and regularization functions employed for SVM analysis may also impact the results. Furthermore, the current framework choices and the value-based HARL approach supports the construction of beneficial initial guess routes in a discrete state and control environment. It is valuable to explore the effect of varied heuristic functions, as well as the outcomes from employing alternative reinforcement learning techniques (e.g., actor-critic) that exploit the benefits of both value and policy-based methods and enables continuous state space transitions as well. Alternative RL implementations for the rendezvous problem in Section 6.7 may mitigate the challenges highlighted in the associated discussions as well. The impact of introducing partially discounted rewards, alternative reward shaping functions, and temporal difference learning are also useful pursuits. With regards to Dijkstra's algorithm, the incorporation of alternative data structures may improve computational efficiency.

3. *Spacecraft Modeling and Numerical Methods*

   The numerical corrections and optimization process for the CSI regime in this analysis employs a bang-bang control that results in coast segments or thrust segments at the s/c maximum thrust capability. In reality, an ion engine operates within the range of thrust magnitudes bounded by its maximum thrust capability. Incorporation of such a variable but bounded thrust magnitude model is useful for future efforts, as the initial guess is also generated under such an assumption that permits maneuver magnitudes ranging from 0 m/s to a maximum velocity perturbation size during state selection from an AR (Section 5.1). Furthermore, with regards to the optimization process to deliver a continuous solution, it would be useful to compare the generated mass-optimal outcomes with results obtained from a weighted objective function of propellant mass and time during optimization, that prioritizes the same weighted objective quantity pursued during the initial guess generation phase.

4. *Mission Operations*

   In addition to the altitude constraints imposed during the pathfinding and initial guess generation phase, the introduction of various other mission considerations such as viewing geometry, eclipsing constraints, and orbit determination errors may highlight the relevance of RL for other interesting mission applications as well. Furthermore, the current investigation and associated results focus solely on the Earth-Moon CR3BP. However, the general methodology is extendable to other dynamical systems and higher-fidelity dynamical models as well.

This investigation blends Artificial Intelligence (AI) strategies with traditional astrodynamics approaches to mitigate many challenges identified for designing trajectories in complex dynamical regimes subject to large state-spaces and trade-spaces. The intersection of the AI and astrodynamics fields facilitates enhanced and quick insights into the iterative mission design process, and also enables rapid re-design. The insights gained from this analysis highlight many avenues for continued investigation.

REFERENCES

[1] K. Hambleton (Most recent edits). Deep Space Gateway to Open Opportunities for Distant Destinations, 2018. Published by: National Aeronautics and Space Administration (NASA), https://www.nasa.gov/feature/deep-space-gateway-to-open-opportunities-for-distant-destinations, Last Accessed Date: [11 April 2019].

[2] I. Newton. *The Principia: Mathematical Principles of Natural Philosophy*. University of California Press, California, USA, 1999. Translation by I. B. Cohen and A. Whitman.

[3] V. Szebehely. *Theory of Orbits*. Academic Press, Connecticut, USA, first edition, 1967.

[4] J. Barrow-Green. *Poincaré and the Three Body Problem*. Volume 11, History of Mathematics, American Mathematical Society, Providence, Rhode Island, 1997.

[5] L. Perko. *Differential Equations and Dynamical Systems*. Springer-Verlag, New York, USA, 1991.

[6] F. R. Moulton. *Periodic Orbits*. Carnegie Institution of Washington, Washington, first edition, 1920.

[7] National Aeronautics And Space Administration (NASA). ISEE 3, 2019. NASA Space Science Data Coordinated Archive, https://nssdc.gsfc.nasa.gov/nmc/spacecraft/display.action?id=1978-079A, Last Accessed Date: [07 June 2014].

[8] D. Burnett, B. Barraclough, R. Bennet, M. Neugebauer, L. Oldham, C. Sasaki, D. Sevilla, N. Smith, E. Stansbery, D. Sweetnam, and R. Wiens. The Genesis Discovery Mission: Return of Solar Matter to Earth. *Space Science Reviews*, 105(3-4), 2003.

[9] K. Howell, B. Barden, R. Wilson, and M. Lo. Trajectory Design Using a Dynamical Systems Approach with Application to GENESIS. *Advances in the Astronautical Sciences*, 97(Part II):1665–1684, 1997.

[10] M. Lo, B. Williams, W. Bollman, D. Han, Y. Hahn, J. Bell, E. Hirst, R. Corwin, P. Hong, K. Howell, B. Barden, and R. Wilson. Genesis Mission Design. In *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Massachusetts, USA, 1998.

[11] T. Phillips. Dead Spacecraft Walking, 2013. http://www.nasa.gov/mission_pages/artemis/news/dead-walking.html#.U7NYXPldUaA, Last Accessed Date: [01 July 2014].

[12] F. Markley, S. Andrews, J. O'Donnell, and D. Ward. The Microwave Anisotropy Probe (MAP) Mission. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, California, USA, 2002.

[13] B. F. Kutter and G. F. Sowers. Cislunar-1000: Transportation supporting a self-sustaining Space Economy. In *AIAA Space*, California, USA, 2016.

[14] E. M. Zimovan, K. C. Howell, and D. C. Davis. Near Rectilinear Halo Orbits And Their Application in Cis-Lunar Space. In *3rd International Academy of Astronautics Conference on Dynamics and Control of Space Systems*, Moscow, Russia, 2017.

[15] S. L. McCarty, W. K. Sjauw, L. M. Burke, and M. L. McGuire. Analysis of Near Rectilinear Halo Orbit Insertion with a 40-kW Solar Electric Propulsion System. In *AAS/AIAA Astrodynamics Specialist Conference*, Utah, USA.

[16] Z. Olikara and D. Scheeres. Numerical Method for Computing Quasi-Periodic Orbits and Their Stability in the Restricted Three-Body Problem. In *1st IAA/AAS Conference on the Dynamics and Control of Space Systems*, Porto, Portugal, 2012.

[17] B. P. McCarthy and K. C. Howell. Trajectory Design Using Quasi-Periodic Orbits in the Multi-Body Problem. In 29^{th} *AAS/AIAA Space Flight Mechanics Meeting*, Hawaii, USA, 2019.

[18] D. C. Folta, N. Bosanac, D. Guzzetti, and K. C. Howell. An Earth-Moon System Trajectory Design Reference Catalog. *Acta Astronautica*, 100:341–353, 2015.

[19] D. Guzzetti, N. Bosanac, A. Haapala, K. C. Howell, and D. Folta. Rapid Trajectory Design in the Earth-Moon Ephemeris System via an Interactive Catalog of Periodic and Quasi-Periodic Orbits. *Acta Astronautica*, 126:439–455, 2016.

[20] J. S. Parker and R. L. Anderson. *Low Energy Lunar Trajectory Design.* John Wiley and Sons, New Jersey, USA, 2014.

[21] J. Schoenmaekers. Post-Launch Optimisation of the SMART-1 Low-Thrust Trajectory to the Moon. email: Johan.schoenmaekers@esa.int.

[22] G. Mingotti and F. Topputo. Ways to the Moon: A Survey. *Spaceflight Mechanics: Proceedings of the AAS/AIAA Spaceflight Mechanics Meeting*, 140(3):2531–2547, 2011.

[23] National Aeronautics and Space Administration (NASA). Dawn, 2019. NASA Space Science Data Coordinated Archive, https://nssdc.gsfc.nasa.gov/nmc/spacecraft/display.action?id=2007-043A, Last Accessed Date: [14 March 2019].

[24] A. V. Ilin, L. D. Cassady, T. W. Glover, M. D. Carter, and F. R. Chang Díaz. A Survey of Missions using VASIMR for Flexible Space Exploration. Prepared by the Ad Astra Rocket Company in fulfillment of Task Number 004 of NASA PO Number NNJ10HB38P, April 2010.

[25] F. Topputo, M. Vasile, and F. Bernelli-Zazzera. Low Energy Interplanetary Transfers Exploiting Invariant Manifolds of the Restricted Three Body Problem. *The Journal of Astronautical Sciences*, 53(4):353–372, 2005.

[26] C. C. Conley. Low Energy Transit Orbits in the Restricted Three-Body Problem. *Society for Industrial and Applied Mathematics Journal on Applied Mathematics*, 16:732–746, 1968.

[27] W. S. Koon, M. W. Lo, J. E. Marsden, and S. D. Ross. Heteroclinic connections between Periodic Orbits and Resonance Transitions in Celestial Mechanics. *Chaos*, 10:427–469, 2000.

[28] 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.

[29] F. Topputo, M. Vasile, and F. B. Zazzera. Earth-to-Moon Low Energy Transfers Targeting L_1 Hyperbolic Transit Orbits. *Annals of New York Academy of Sciences*, 1065:55–76, 2005.

[30] M. Vaquero and K. C. Howell. Leveraging Resonant Orbit Manifolds to Design Transfers between Libration Point Orbits in Multi-Body Regimes. *Journal of Guidance, Control and Dynamics*, 37(3):1143–1157, 2014.

[31] A. Haapala, and K. C. Howell. A Framework for Construction of Transfers Linking Periodic Libration Point Orbits in the Earth-Moon Spatial Circular Restricted Three-Body Problem. *Journal of Bifurcations and Chaos*, 26(5), 2016.

[32] L. Capdevila, and K. C. Howell. A Transfer Network Linking Earth, Moon, and the Triangular Libration Point Regions in the Earth-Moon System. *Advances in Space Research*, 62(7):1826–1852, 2018.

[33] M. Kakoi, K. C. Howell, and D. Folta. Access to Mars from Earth-Moon Libration Point Orbits: Manifold and Direct Options. *Acta Astronautica*, 102:269–286, 2014.

[34] G. Mingotti, F. Topputo, and F. Bernelli-Zazzera. Transfers to Distant Periodic Orbits Around the Moon via their Invariant Manifolds. *Acta Astronautica*, 79:20–32, 2012.

[35] G. Mingotti, F. Topputo and F. Bernelli-Zazzera. Efficient Invariant-Manifold, Low-Thrust Planar Trajectories to the Moon. *Communications in Nonlinear Science and Numerical Simulations*, 17(2):817–831, 2012.

[36] G. Mingotti, F. Topputo, and F. Bernelli-Zazzera. Low-Energy, Low-Thrust Transfers to the Moon. *Celestial Mechanics and Dynamical Astronomy*, 105(61), 2009.

[37] J. S. Parker, R. L. Anderson, and A. Peterson. Surveying Ballistic Transfers to Low Lunar Orbit. *Journal of Guidance, Control and Dynamics*, 36(5), 2013.

[38] N.Bosanac, A. D. Cox, K. C. Howell, and D. C. Folta. Trajectory Design for a Cislunar CubeSat Leveraging Dynamical Systems Techniques: The Lunar IceCube Mission. *Acta Astronautica*, 144:283–296, 2018.

[39] M. Ozimek and K. C. Howell. Low-Thrust Transfers in the Earth-Moon System Including Applications to Libration Point Orbits. *Journal of Guidance, Control, and Dynamics*, 33(2), 2010.

[40] K. C. Howell, D. C. Davis, and A. Haapala. Application of Periapse Maps for the Design of Trajectories Near the Smaller Primary in Multi-Body Regimes. *Journal of Mathematical Problems in Engineering, Special Issue: Mathematical Methods Applied to the Celestial Mechanics of Artificial Satellites*, 2012, 2011.

[41] A. Haapala and K.C. Howell. Representations of Higher-Dimensional Poincaré Maps with Application to Spacecraft Trajectory Design. *Acta Astronautica*, 96:23–41, 2014.

[42] R. Pritchett, K.C. Howell, and D. Grebow. Low-Thrust Transfer Design Based on Collocation Techniques: Applications in the Restricted Three-Body Problem. In *AAS/AIAA Astrodynamics Specialist Conference*, Washington, USA, 2017.

[43] R. Prichett, A. D. Cox, K. C. Howell, D. C. Folta, and D. Grebow. Low-Thrust Trajectory Design Via Direct Transcription Leveraging Structures from the Low-Thrust Restricted Problem. In *69th International Astronautical Congress*, Bremen, Germany, 2018.

[44] H. B. Keller. *Numerical Solution of Two Point Boundary Value Problems.* Society for Industrial and Applied Mathematics, Philadelphia, 1976.

[45] R. P. Russell. Primer Vector Theory Applied to Global Low-Thrust Trade Studies. *Journal of Guidance, Control and Dynamics*, 30(2):460–472, 2007.

[46] J. Stuart, K. C. Howell and R. Wilson. Automated Design of Propellant-Optimal, End-to-End, Low-Thrust Trajectories for Trojan Asteroid Tours. *Journal of Spacecraft and Rockets*, 51(5):1631–1647, 2014.

[47] M. Ozimek. *Low-Thrust Trajectory Design and Optimization of Lunar South Pole Coverage Missions.* PhD dissertation, School of Aeronautics and Astronautics, Purdue University, 2010.

[48] D. Grebow, M. Ozimek, and K. C. Howell. Design of Optimal Low-Thrust Lunar Pole-Sitter Missions. *Journal of the Astronautical Sciences*, 58(1):55–79, 2011.

[49] G. Radice and G. Olmo. Ant Colony Algorithms for Two-Impulse Interplanetary Trajectory Optimization. *Journal of Guidance, Control and Dynamics*, 29(6):1440–1444, 2006.

[50] M. Ceriotti and M. Vasile. MGA Trajectory Planning With an ACO-Inspired Algorithm. *Acta Astronautica*, 67(9-10):1202–1217, 2010.

[51] J. Stuart, K. C. Howell and R. Wilson. Design of End-to-End Trojan Asteroid Rendezvous Tours Incorporating Potential Scientific Value. *Journal of Spacecraft and Rockets*, 53(2):278–288, 2016.

[52] R. Furfaro and R. Linares. Waypoint-Based Generalized ZEM/ZEV Feedback Guidance for Planetary Landing via a Reinforcement Learning Approach. In *3rd IAA Conference on Dynamics and Control of Space Systems*, Moscow, Russia, 2017.

[53] R. Beeson, V. Shah, J. Aurich, and D. Ellison. Automated Solution of Low Energy Trajectories. In *AAS/AIAA Astrodynamics Specialist Conference*, Stevenson, Washington, 2017.

[54] S. L. McCarty, L.M. Burke, and M. L. McGuire. Parallel Monotonic Basin Hopping for Low Thrust Trajectory Optimization. In *AIAA SciTech Forum*, Florida, USA, 2018.

[55] P. Witsberger and J. Longuski. Interplanetary Trajectory Design Using a Recurrent Neural Network and Genetic Algorithm. In *AAS/AIAA Astrodynamics Specialist Conference*, Utah, USA, 2018.

[56] S. De Smet, D. J. Scheeres, and J. S. Parker. Systematic Exploration of Solar Gravity Driven Orbital Transfers in the Martian System using Artificial Neural Networks. In *AAS/AIAA Astrodynamics Specialist Conference*, Utah, USA, 2018.

[57] G. A. Tsirogiannis. A Graph Based Method for Mission Design. *Celestial Mechanics and Dynamical Astronomy*, 114(4):353–363, 2012.

[58] E. Trumbauer and B. Villac. Heuristic Search-Based Framework for Onboard Trajectory Redesign. *Journal of Guidance, Control and Dynamics*, 37(1):164–175, 2014.

[59] N. Parrish. A* Pathfinding for Continuous-Thrust Trajectory Optimization. In *AAS 37th Annual Guidance & Control Conference*, Colorado, USA, 2014.

[60] D. Miller and R. Linares. Low-Thrust Optimal Control via Reinforcement Learning. In *29th AAS/AIAA Space Flight Mechanics Meeting*, Hawaii, USA, 2019.

[61] A. Das-Stuart, K. C. Howell, and D. Folta. Rapid Trajectory Design in Complex Environments Enabled by Reinforcement Learning and Exact Methods. *Acta Astronautica (Accepted: 18 April 2019).*

[62] A. Das-Stuart, K. C. Howell, and D. Folta. Rapid Trajectory Design in Complex Environments Enabled Via Supervised and Reinforcement Learning Strategies. In *69th International Astronautical Congress*, Bremen, Germany, 2018.

[63] J. A. Sims and S. N. Flanagan. Design of Low-Thrust Interplanetary Missions. In *AAS/AIAA Astrodynamics Specialist Conference*, Alaska, USA, 1999.

[64] A. Verdun. Leonhard Euler's early lunar theories 1725-1752. *Archive for History of Exact Sciences*, 67(5):477–551, 2013.

[65] T. Pavlak. *Trajectory Design and Orbit Maintenance Strategies in Multi-Body Dynamical Regimes.* PhD dissertation, School of Aeronautics and Astronautics, Purdue University, 2013.

[66] A. F. Haapala. Trajectory Design Using Periapse Maps and Invariant Manifolds. Master's thesis, School of Aeronautics and Astronautics, Purdue University, 2010.

[67] T. M. Vaquero Escribano. *Poincaré Sections and Resonant Orbits in the Restricted Three-Body Problem.* PhD dissertation, School of Aeronautics and Astronautics, Purdue University, 2010.

[68] C. Saltzer, R. Craig, and C. Fetheroff. Comparison of Chemical and Electric Propulsion Systems for Interplanetary Travel. *Proceedings of the IRE*, 48(4):465–476, 1960.

[69] F. R. Chang Díaz. An overview of the VASIMR engine: High power space propulsion with RF plasma generation and heating. *AIP Conference Proceedings*, 595(1):3, 2001.

[70] NASA JPL-Navigation and Ancillary Information Facility. An Overview of Reference Frames and Coordinate System sin the SPICE Context. https://naif.jpl.nasa.gov/pub/naif/toolkit_docs/Tutorials/pdf/individual_docs/17_frames_and_coordinate_systems.pdf, Last Accessed Date: [4 December 2018].

[71] NASA JPL-Navigation and Ancillary Information Facility. The SPICE Toolkit. https://naif.jpl.nasa.gov/naif/toolkit.html, Last Accessed Date: [4 December 2018].

[72] K. Boudad, A. Cox, and E. Zimovan. Personal Conversations, 2019.

[73] Marquez. Chapter 3: Lyapunov Stability I: Autonomous Systems. Clemson University, College of Engineering and Science,Intelligent Systems. Slides for Electrical Engineering Course 874.

[74] P. Dawkins. Paul's Online Math Notes. http://tutorial.math.lamar.edu/Classes/DE/PhasePlane.aspx, Last Accessed Date: [8 July 2014].

[75] Rutgers. Stability and Asymptotic Stability. http://www.math.rutgers.edu/~zeilberg/critical.html, Last Accessed Date: [20 June 2014].

[76] T. S. Parker and L. O. Chua. *Numerical Algorithms for Chaotic Systems*. Springer-Verlag, New York, USA, 1989.

[77] J. Guckenheimer and P. Holmes. *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*. Springer-Verlag, New York, USA, 1983.

[78] J. R. Stuart. *A Hybrid Systems Strategy For Automated Spacecraft Tour Design and Optimization*. PhD dissertation, School of Aeronautics and Astronautics, Purdue University, 2014.

[79] W.Schlei. *Interactive Spacecraft Trajectory Design Strategies Featuring Poincaré Map Topology*. PhD dissertation, School of Aeronautics and Astronautics, Purdue University, 2017.

[80] A. E. Roy and M. W. Ovenden. On the Occurrence of Commensurable Mean Motions in the Solar System: II. The Mirror Theorem. *Monthly Notices of the Royal Astronomical Society*, 115(3):296–309, 1955.

[81] I. Panardo. Stability of Periodic Systems and Floquet Theory. Master's thesis, Universit à Degli Studi di Padova, 2015.

[82] W. G. Kelly and A. C. Peterson. *Difference Equations: An Introduction with Applications*. Academic Press, California, USA, 2001.

[83] L. R. Irrgang. Investigation of Transfer Trajectories To and From the Equilateral Libration Points L_4 and L_5 in the Earth-Moon System. Master's thesis, School of Aeronautics and Astronautics, Purdue University, 2008.

[84] V. A. Yakubovich and V. M. Starzhinskii. *Differential Equations with Periodic Coefficients*. John Wiley & Sons, New York, USA, 1975.

[85] A. F. Haapala. *Trajectory Design in the Spatial Circular Restricted Three-Body Problem Exploiting Higher-Dimensional Poincaré Maps*. PhD dissertation, School of Aeronautics and Astronautics, Purdue University, 2014.

[86] F. Topputo and C. Zhang. Survey of Direct Transcription for Low-Thrust Space Trajectory Optimization with Applications. *Abstract and Applied Analysis*, 2014(2):1–15, 2014.

[87] O. Bolza. *Lectures on the Calculus of Variations (Reprint)*. Dover Publications, New York, USA, 2018.

[88] R. Thiele, and Editors: R. E. Bradley and C. E. Sandifer. *Studies in the history and Philosophy of Mathematics, Chatper: Euler and the Calculus of Variations*. Elsevier Science, Amsterdam, Netherlands, 2007.

[89] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mischchenko. *The Mathematical Theory of Optimal Processes*. Interscience, A Division of John Wiley & Sons, New Jersey, USA, 1962.

[90] D. F. Lawden. *Optimal Trajectories for Space Navigation*. Butterworths, London, United Kingdom, 1963.

[91] J. -P. Marec. *Optimal Space Trajectories*. Elsevier, Amsterdam, Netherlands, 1979.

[92] J. T. Betts. Survey of Numerical Methods for Trajectory Optimization. *Journal of Guidance, Control and Dynamics*, 21(2):193–207, 1998.

[93] P. E. Gill, W. Murray and M. A. Saunders. SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization. *Society for Industrial and Applied Mathematics Journal of Optimization*, 12(4):979–1006, 2002.

[94] P. Williams. A Gauss-Lobato Quadrature Method for Solving Optimal Control Problems. *ANZIAM Journal*, 47:101–115, 2006.

[95] O. von Stryk and R. Bulirsch. Direct and Indirect Methods for Trajectory Optimization. *Annals of Operations Research*, 37(1):357–373, 1992.

[96] A. Y. Ng and M. I. Jordan. On Discriminative vs. Generative Classifiers: A Comparison of Logistic Regression and Naive Bayes. *Advances in Neural Information Processing Systems*, 2, 2002.

[97] F. Rosenblatt. The Perceptron: A Probabilistic Model For Information Storage And Organization In The Brain. *Psychological Review*, 65(6):386–408, 1958.

[98] M. Minsky and S. Papert. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, Massachusetts, USA, 1969.

[99] J. Bapu. The Artificial Neural Networks Handbook: Part 1, 2018. `https://www.datasciencecentral.com/profiles/blogs/the-artificial-neural-networks-handbook-part-1`, Last Accessed Date: [28 August 2018].

[100] I. Goodfellow, Y. Bengio and A. Courville. *Deep Learning*. MIT Press, Massachusetts, USA, 2016.

[101] S. Haykin. *Neural Networks and Learning Machines*. Prentice Hall, New Jersey, USA, 2008.

[102] N. Buduma (with contributions from N. Locascio). *Fundamentals of Deep Learning*. O'Reilly Media, California, USA, 2017.

[103] Y. A. LeCun, L. Bottou, G. B. Orr, and K-R. Müller. Efficient BackProp. *Neural Networks: Tricks of the Trade, Lecture Notes in Computer Science*, 7700:9–48, 2012.

[104] J. Platt (Editors A. J. Smola and P. L. Bartlett and B. Schölkopf and and D. Schuurmans). *Probabilities for SV Machines. In: Advances in Large Margin Classifiers*. MIT Press, Massachusetts, USA, 2000.

[105] M. F. Møller. A Scaled Conjugate Gradient Algorithm For Fast Supervised Learning. *Neural Networks*, 6:525–533, 1993.

[106] M. Akay. *Handbook of Neural Engineering*. IEEE Press, New Jersey, USA, 2007.

[107] P. Dahal. Classification and Loss Evaluation - Softmax and Cross Entropy Loss, 2018. https://deepnotes.io/softmax-crossentropy, Last Accessed Date: [23 August 2018].

[108] G. Cybenko. Approximation by Superpositions of a Sigmoidal Function. *Mathematics of Control, Signals, and Systems*, 2:303–314, 1989.

[109] K. Funahashi. On the Approximate Realization of Continuous Mappings by Neural Networks. *Neural Networks*, 2:183–192, 1989.

[110] G. Stuart, E. Bienenstock and R. Doursat. Neural Networks and the Bias/Variance Dilemma. *Neural Computation*, 4(1):1–58, 1992.

[111] C. M. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford, United Kingdom, 1995.

[112] C. Cortes and V. Vapnik. Support-Vector Networks. *Machine Learning*, 20(3):273–297, 1995.

[113] M. E. Mavroforakis and S. Theodoridis. A Geometric Approach to Support Vector Machine (SVM) Classification. *IEEE Transactions on Neural Networks*, 17(3):671–682, 2006.

[114] A. Ng. CS229 Lecture notes, 2018. Stanford University, http://cs229.stanford.edu/notes/cs229-notes3.pdf.

[115] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge, United Kingdom, 2004.

[116] J. Zhu, S. Rosset, R. Tibshirani, T. J. Hastie, and Editors: S. Thrun and L. K. Saul and B. Schölkopf. 1-Norm Support Vector Machines. In *Advances in Neural Information Processing Systems 16, Proceedings from Neural Information Processing Systems Conference*, British Columbia, Canada, 2003.

[117] N. Cristianini and J. Shawe-Taylor. *Support Vector Machines and Other Kernel-Based Learning Methods.* Cambridge University Press, Cambridge, United Kingdom, 2000.

[118] V. I. Paulsen and M. Raghupathi. *An Introduction to the Theory of Reproducing Kernel Hilbert Spaces.* Cambridge University Press, Cambridge, United Kingdom, 2016.

[119] M. Gagne. Hilbert Space Theory and Applications in Quantum Mechanics, 2013.

[120] M. Claesen, F. De Smet, J. A. K. Suykens, and B. De Moor. Fast Prediction With SVM Models Containing RBF Kernels. *arXiv: Machine Learning : arXiv: 1403. 0736v3*, 2014.

[121] J. Platt and Editors B. Schokopf and C. Burges and A. Smola. *Fast Training of Support Vector Machines Using Sequential Minimal Optimization, In: Advances in Kernel Methods - Support Vector Learning.* MIT Press, Massachusetts, USA, 1998.

[122] J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems*, 25:2960–2968, 2012.

[123] I. Gribkovskaia, Ø. Halskau and Sr. G. Laporte. The bridges of Königsberg-A historical perspective. *Networks, An International Journal*, 49(3):199–203, 2007.

[124] D. L. Poole and A. K. Mackworth. *Artificial Intelligence, Foundations of Computational Agents.* Cambridge University Press, New York, USA, 2010.

[125] S. Dasgupta, C. Papadimitriou, and U. Vazirani. *Algorithms.* McGraw Hill, Massachusetts, USA, 2006.

[126] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction.* MIT Press, Massachusetts, USA, second edition, 2018.

[127] R. A. C Bianchi, C. H. C Ribeiro, and A. H. R. Costa. Heuristically Accelerated Reinforcement Learning: Theoretical and Experimental Results. *Frontiers in Artificial Intelligence and Applications*, 242:169–174, 2012.

[128] R. A. C Bianchi, C. H. C. Ribeiro, and A. H. R. Costa. On the Relation between Ant Colony Optimization and Heuristically Accelerated Reinforcement Learning. In *1st International Workshop on Hybrid Control of Autonomous Systems - Integrating Learning, Deliberation and Reactive Control*, California, USA, 2009.

[129] B. Fernandez-Gauna, I. Etxeberria-Agiriano, and M. Graña. Learning Multi-robot Hose Transportation and Deployment by Distributed Round-Robin Q-Learning. *PLOS One*, 10(7), 2015.

[130] M. Rayman (Director and Chief Engineer, Dawn Mission). Dawn Journal: Attitude Adjustment, 2010. Planetary Society, http://www.planetary.org/blogs/guest-blogs/marc-rayman/2807.html, Last Accessed Date: [31 Oct 2018].

[131] D. J. Grebow, G. J. Whiffen, D. Han, and B. M. Kennedy. Dawn Safing Approach to Ceres Re-Design. In *AIAA/AAS Astrodynamics Specialist Conference, AIAA SPACE Forum*, California, USA, 2016.

[132] T. Imken, T. Randolph, M. DiNicola, and A. Nicholas. Modeling Spacecraft Safe Mode Events. In *IEEE Aerospace Conference*, Montana, USA, 2018.

8. VITA

Ashwati Das graduated with her Bachelor's Degree in Aerospace Engineering from the University of Sydney, Australia. She went on to work as an aircraft software and integration specialist on the Hercules, Black Hawk and A380 simulators with CAE Australia. Her yearning to be involved in and contribute towards the space industry prompted her to resume her studies at Purdue University. Her Master's research focused on evaluating the feasibility of exploiting solar sails as a primary mode of transport, to enable propellant-free transfers in complex dynamical regimes. Her Ph.D. research has focused on exploiting artificial intelligence techniques, specifically supervised and reinforcement learning, to discover transport options for chemical and low-thrust propulsive capabilities in complex dynamical regimes. She has completed three summer internships with the NASA Jet Propulsion Laboratory, California Institute of Technology, contributing to mission design trade studies for the Mars Sample Return and Europa Lander mission concepts. She has also worked on trajectory design applications with NASA Goddard Spaceflight Center and Orbital ATK, focused on transport applications to support near-term exploration endeavors in the Earth-Moon system. As part of a multi-disciplinary Purdue research team, she has also contributed towards evaluations of Moon/Mars space systems architecture alternatives via a System-of-Systems (SoS) operability and developmental risk analysis for NASA Marshall. While a graduate student, she has been supported by AAE departmental funding, NASA Goddard and NASA Marshall research grants, and the Zonta Amelia Earhart Fellowship. Being a strong space advocate, she has also led events in the community to help create awareness of space endeavors, and loves inspiring the younger generation to join the space journey!