Electrodynamic Tether Pump-Down and Invariant Manifold Transit in the Jupiter-Io System: A pre-Phase A Trajectory Feasibility Study

a project presented to
The Faculty of the Department of Aerospace Engineering
San José State University

in partial fulfillment of the requirements for the degree
Master of Science in Aerospace Engineering

by

Svitlana Kuklenko

May 2026

approved by

Dr. Derek Lang
Faculty Advisor



© 2026
Svitlana Kuklenko
ALL RIGHTS RESERVED

ABSTRACT

Electrodynamic Tether Pump-Down and Invariant Manifold Transit in the Jupiter-Io System: A pre-Phase A Trajectory Feasibility Study

Svitlana Kuklenko

This study establishes the dynamical feasibility of propellantless transfer from a post-Jupiter orbit insertion capture ellipse to the vicinity of Io using an electrodynamic tether (EDT) as both the energy extraction mechanism and the manifold-bridging tool. Operating within the circular restricted three-body problem with continuous low-thrust forcing (CR3BP-LT), the analysis derives the equations of motion in the Jupiter-Io rotating frame, computes the L1 Lyapunov reference orbit and its invariant manifold tubes, and uses Poincaré section analysis to characterize the phase-space gap between the post-capture trajectory and the stable manifold targeting set. EDT forcing progressively reduces the spacecraft's orbital energy through apojove pump-down until the L1 gateway opens and the stable manifold becomes geometrically accessible, at which point a small EDT steering maneuver bridges the residual gap propellantlessly. Near-heteroclinic connections are identified across multiple pump-down strategies and validated against a range of hardware configurations, demonstrating that propellantless manifold transit to Io is dynamically achievable within a hardware-validated EDT framework. Radiation shielding emerges as a primary design driver for subsequent development, identifying shielding design rather than propulsion as the binding constraint for mission advancement.

ACKNOWLEDGEMENTS

An endeavor like this takes a village. My aspirations in satellite and space hardware design and development grew steadily through my college years, shaped by the deeply interconnected disciplines of orbital mechanics, trajectory optimization, mechanical and systems engineering, and mission design. Few things bring me greater joy than overseeing a complex system come to life.

The UNAGI mission is where this project began. It would not exist without the vision and dedication of Spencer Devins, whose years of curiosity transformed a personal passion project into an official student organization at SJSU in Fall 2023. Thank you for welcoming me with open arms and trusting me to co-lead the mission development. I am equally proud of the entire Spartan Space Systems team for generously giving their time and energy to push this ambitious concept forward.

I owe sincere gratitude to my project advisor, Dr. Derek Lang, for his guidance, patience, and support, and especially for keeping me gently grounded when my aspirations threatened to outpace my timeline. I am equally grateful to Dr. Jeanine Hunter, whose encouragement gave me the confidence to pursue astrodynamics seriously, even after my AE 142 attempt at optimizing the Europa Clipper trajectory sent me down the rabbit hole that ultimately led here. I also thank Prof. Ching Ching Tan and the SJSU Grad Slam 2026 cohort, whose feedback pushed me to articulate the larger significance of this work more clearly and was a process that unexpectedly also sharpened my technical understanding of it.

Beyond SJSU, I am grateful to my mentors and colleagues at Maxar Technologies and Longshot Space for believing in me and creating space for me to grow as an engineer. I also extend my appreciation to Dr. Ali Guarneros-Luna and the SpartanSat team, AIAA, the NASA L'Fam, and the broader SJSU faculty for the formative opportunities and the environment that let me explore everything from entrepreneurship and Chinese to glass-blowing and technical project management, all of which shaped me in ways that found their way back into my engineering.

To my closest friends, near and far: thank you for sustaining me throughout this journey, especially my brother and fellow classmate Dmitry, for the shared ideas, technical assistance, and mutual suffering, and my wonderful partner Nicholas, for his unwavering support, contagious positivity, and for being one of my brightest rays of sunshine.

Finally, I dedicate this project, and the culmination of my formal education, to my parents. Дякую вам за беззастережну любов і за те, що дозволили мені повністю зосередитися на навчанні останні декілька років. Я буду вічно вдячна.

Table of Contents

1. Introduction **1**
1.1 Motivation 1
1.2 Literature Review 2
1.2.1 Electrodynamic Tether Technology and Jovian Feasibility 2
1.2.2 Low-Energy Transfer Design in the Jovian System 4
1.2.3 The Synthesis Gap: EDT Propulsion and Manifold Targeting 7
1.3 Project Proposal 8
1.4 Methodology 9
2. Problem Formulation **10**
2.1 Overview 10
2.2 Orbit Transfer Problem Statement 10
2.3 Two-Body and Patched Conic Baseline 11
2.4 Dynamical Formulation: CR3BP-LT Equations of Motion 12
2.4.1 The Circular Restricted Three-Body Problem 12
2.4.2 The Jacobi Constant and Zero-Velocity Surfaces 13
2.4.3 EDT Forcing and the CR3BP-LT Model 15
2.5 Periodic Orbits and Invariant Manifolds 16
2.5.1 Lagrange Points and Flow Classification 16
2.5.2 Lyapunov Orbits and the Differential Corrector 17
2.5.3 The Variational Equations and Monodromy Matrix 18
2.5.4 Manifold Tube Construction 20
2.6 Poincaré Section and Gap Metric 21
2.7 Scope and Assumptions 23
3. Implementation and Validation **26**
3.1 Overview 26
3.2 State-Space Formulation 28
3.3 ODE Solver and Tolerance Selection 28
3.4 Differential Corrector Implementation 29
3.5 Pump-Down Stopping Conditions 30
3.6 EDT Force Direction Convention 31
3.7 Arrival Angle Parameterization 31
3.8 Poincaré Section Crossing Detection 32
3.9 Numerical Parameters Summary 33
3.10 Validation 33
4. Results **35**
4.1 Overview 35
4.2 Post-JOI Boundary State 35

4.3 Lyapunov Orbit and Manifold Characterization..... 35
4.3.1 Lyapunov Orbit Computation..... 35
4.3.2 Monodromy Matrix and Stability Analysis..... 36
4.3.3 Stable and Unstable Manifold Directions..... 38
4.3.4 Manifold Tube Propagation..... 38
4.3.5 Poincaré Section of Stable Manifold Crossings..... 40
4.4 EDT Pump-Down — Phase 1..... 41
4.4.1 Pump-Down Simulation Results..... 42
4.4.2 EDT Performance at the Burn Point..... 43
4.5 Phase 2 Manifold Steering..... 43
4.5.1 Poincaré Map and Gap Quantification..... 43
4.5.2 Arrival Angle Sweep and the Near-Heteroclinic Connection..... 44
4.5.3 Complete Transfer Summary..... 48
4.6 Sensitivity Analysis..... 48
4.6.1 Sweep Design..... 48
4.6.2 Power and Tether Length..... 49
4.6.3 Magnetic Field Uncertainty..... 50
4.6.4 Spacecraft Mass..... 51
4.6.5 Feasibility Envelope Summary..... 52
4.7 Capture Orbit Geometry..... 52
4.7.1 Perijove Sweep Design..... 52
4.7.2 Pump-Down Duration vs. Perijove Radius..... 53
4.7.3 Radiation Dose vs. Perijove Radius..... 54
4.7.4 The Perijove Trade..... 55
5. Discussion..... 57
5.1 Near-Heteroclinic Connection and Phase-Space Gap Analysis..... 57
5.2 EDT as a Manifold Bridging Mechanism..... 58
5.3 Mission Feasibility and Sensitivity..... 59
5.4 Model Limitations and Fidelity Escalations..... 59
5.5 Comparison to Prior Work..... 60
6. Conclusions and Future Work..... 62
6.1 Conclusions..... 62
6.2 Future Work..... 63
References..... 65
Appendix A: Matlab Code..... 69
Appendix B: JOI Chemical Baseline..... 152

List of Figures

Figure 1.1 - Illustration of EDT propulsion generation within Earth's magnetic field……3
Figure 1.2 - Conceptual illustration of a spacecraft equipped with a Multi-EDT system……...3
Figure 1.3 - Lorentz force as a function of radial distance from Jupiter for tether lengths…….....4
Figure 1.4 - Stable and unstable manifold tubes of the L1 and L2 periodic orbits……..6
Figure 1.5 - The InterMoon Superhighway trajectory concept……7
Figure 1.6 - Phases of Magnetour: left to right……7
Figure 2.1 - Schematic of the four-phase propellantless transfer from JOI capture to Io……..11
Figure 2.2 - Potential surface U(x,y) and zero velocity curves with superimposed…….……15
Figure 2.3 - Schematic of Lagrange points within general CR3BP geometry…..…….....17
Figure 2.4 - Stable (W^s) and unstable (W^u) manifold tubes of the L1 and L2 Lyapunov…..……23
Figure 3.1 - Computational pipeline for the CR3BP-LT manifold targeting analysis……...26
Figure 4.1 - Converged L1 Lyapunov reference orbit in the Jupiter-Io synodic rotating frame…36
Figure 4.2 - Invariant manifold tubes of the L1 Lyapunov reference orbit in the Jupiter-Io…….38
Figure 4.3 - Zoomed view of the L1 neck region showing the intersection of the stable and…...39
Figure 4.4 - Poincaré map of the stable manifold at $y = 0$ with $\dot{x} < 0$……...…40
Figure 4.5 - Jacobi constant evolution during Phase 1 EDT pump-down……..…42
Figure 4.6 - Full 360° coarse arrival angle sweep at 15° resolution for full pump-down case….44
Figure 4.7 - Dedicated Poincaré map at the optimal arrival angle $\theta = 167°$ for the partial…..…..45
Figure 4.8 - Dedicated Poincaré map at the optimal arrival angle $\theta = 151°$ for the full……46
Figure 4.9 - EDT pump-down duration as a function of power budget for eight tether……48
Figure 4.10 - Mean ΔC per capture orbit as a function of power budget for eight tether……..49
Figure 4.11 - EDT pump-down duration as a function of power budget for three B-field……50
Figure 4.12 - EDT pump-down duration as a function of power budget for four spacecraft……51
Figure 4.13 - EDT pump-down duration vs. perijove radius with radiation zone shading……52
Figure 4.14 - Cumulative TID accumulation during Phase 1 EDT pump-down……...53

List of Tables

Table 3.1 - Key numerical parameters used throughout computational implementation……..32
Table 4.1 - Converged L1 Lyapunov reference orbit properties.……..…….34
Table 4.2 - Monodromy matrix eigenvalues and physical interpretation.....….…….36
Table 4.3 - Pump-down stopping condition comparison.…….…….41
Table 4.4 - Near-heteroclinic connection results: partial vs. full pump-down.……..46
Table 4.5 - Complete transfer summary: JOI capture through L1 manifold entry.…….……...47
Table 4.6 - Perijove sweep summary.…….……...54
Table B.1 - Propellant mass comparison for JOI burn……..…151

Nomenclature

Acronyms

CR3BP	=	Circular Restricted 3-Body Problem
CRNBP	=	Circular Restricted N-body Problem
ConOps	=	Concept of Operations
EDT	=	Electrodynamic Tether
GMAT	=	General Mission Analysis Tool
IMS	=	InterMoon Superhighway
JCI	=	Jupiter-Centered Inertial (frame)
JOI	=	Jupiter Orbit Insertion
JUICE	=	Jupiter Icy Moons Explorer
LU	=	Length Unit (CR3BP canonical unit), km
MEDT	=	Multi-EDT System
NIAC	=	NASA Innovative Advanced Concepts
PCCFBP	=	Planar Concentric Circular Restricted Four-Body Problem
STK	=	(Ansys) Systems Tool Kit
STM	=	State Transition Matrix
TID	=	Total Ionizing Dose, krads
TRL	=	Technology Readiness Level
TU	=	Time Unit (CR3BP canonical unit), s or days

Greek Symbol

ε	=	Manifold perturbation size, normalized dimension units (nd)
θ	=	Arrival angle at JOI, degrees
κ	=	Amplitude-velocity ratio (differential corrector), dimensionless
λ	=	Floquet multipliers (eigenvalues of the monodromy matrix), dimensionless
μ	=	Mass ratio or gravitational parameter (GM), dimensionless or km^3/s^2
Φ	=	Monodromy matrix (STM over one period)
ω	=	Angular velocity or mean motion, rad/s
$\omega_{Lyapunov}$	=	In-plane Lyapunov frequency, nd
Ω	=	Rotating frame angular velocity, rad/s

Symbol

ΔC	=	Change in Jacobi constant, nd
ΔC_{total}	=	Total Jacobi constant gap ($C_{cap} - C_{Lyapunov}$), nd
ΔV	=	Velocity increment, km/s or m/s
Δt	=	Transfer time duration, days or s
B	=	Magnetic field strength, T or Gauss
C	=	Jacobi constant (integral of motion in CR3BP), dimensionless
C_{cap}	=	Jacobi constant at JOI capture, nd

C_{L1}	=	L1 neck-opening threshold Jacobi constant, nd
$C_{Lyapunov}$	=	Jacobi constant of L1 Lyapunov reference orbit, nd
F_{EDT}	=	Electrodynamic tether force, N
G	=	Gravitational constant, km^3/s^2
I	=	Tether current magnitude, A
L*	=	Canonical length unit (Jupiter-Io distance), km
L_t	=	Tether length, m or km
N_e	=	Plasma electron density, cm^{-3} or m^{-3}
P_{max}	=	Maximum available electrical power, W
r	=	Radial distance in Jupiter-centric frame, km
R_J	=	Radius of Jupiter, km
r_p	=	Perijove radius, km or RJ
r_{SOI}	=	Sphere of influence radius, km
U	=	CR3BP pseudo-potential function, nd
v_∞	=	Hyperbolic excess velocity at SOI, km/s
v_{rel}	=	Relative velocity between spacecraft and co-rotating plasma, m/s
W^s	=	Stable invariant manifold
W^u	=	Unstable invariant manifold
x_{L1}	=	L1 equilibrium point x-coordinate, nd

1. Introduction

1.1 Motivation

Of all the targets in the outer solar system, Io stands apart. It is the most volcanically active body ever observed, embedded in the strongest radiation and plasma environment encountered by any operational spacecraft, yet no confirmed mission has ever targeted it for dedicated study. That gap reflects not a lack of scientific ambition but the brutal engineering realities of operating in the inner Jovian system: the mass cost of chemical propulsion, the hazard of Jupiter's radiation environment, and the scarcity of power far from the Sun [1, 2, 3]. Any architecture capable of conducting sustained inner-Jovian science must solve the propulsion and power problems simultaneously, and solve them with a technology that can survive the very environment it is trying to exploit.

Electrodynamic tethers offer a single physics-based technology that addresses both constraints at once. An EDT is a long conductive tether, on the order of several kilometers long, that interacts with a planet's magnetic field and ambient plasma to generate propulsion and power without expelling propellant [4]. Near Io, where the Jovian magnetic field is strongest, forces exceeding 50 N and power generation on the order of kilowatts are achievable under realistic conditions [5], making EDTs a credible alternative to both chemical propulsion and radioisotope power systems for inner-Jovian missions [6]. Across the mission profile, propellantless EDT braking during Jupiter orbit insertion could displace 0.5 to 2 km/s of conventional chemical ΔV, potentially freeing hundreds of kilograms for scientific payload [7].

The Jovian system's Lagrange points generate invariant manifolds, which are surfaces in phase space along which the natural gravitational dynamics guide trajectories between gravitational regimes without thrust. When EDT continuous thrust is combined with these low-energy pathways, it becomes possible to design transfers that minimize time in high-radiation zones and reduce deterministic ΔV requirements for the multi-moon tour segment to tens of meters per second [8, 9]. Missions like Europa Clipper and JUICE have already validated the practical utility of resonant orbit strategies at Jupiter [10]; the remaining step is integrating propellantless propulsion directly into that trajectory architecture in the demanding inner-Jovian environment where no mission has yet operated.

This project mathematically derives and numerically demonstrates a propellantless trajectory framework covering the critical segment from post-Jupiter orbit capture to Io's vicinity via the L1 stable manifold of the Jupiter-Io system, under continuous EDT perturbation. The analytical gap this work addresses is characterized in Section 1.2.3 and closed by the results of Chapters 3 and 4.

1.2 Literature Review

The literature relevant to this study spans three interconnected areas: the development of EDT technology and its application to the Jovian environment, the design of low-energy transfers using invariant manifolds in the Jovian moon system, and prior attempts to synthesize these two bodies of work. Each thread has matured substantially over the past three decades, yet their combination in the context of a demonstrated propellantless manifold connection remains an open problem that this study addresses directly.

1.2.1 Electrodynamic Tether Technology and Jovian Feasibility

The concept of EDT propulsion has been under development for over four decades. Early experimental missions established the foundational physics: the Tethered Satellite System missions TSS-1 (1992) and TSS-1R (1996) validated electrodynamic interactions and successfully deployed a 19.6 km tether, while the Plasma Motor Generator mission (1993) and SEDS I and II (1993, 1994) demonstrated deployment mechanics and current collection in low Earth orbit [6, 7]. These missions confirmed that a conductive tether moving through a planetary magnetic field could generate both propulsive force and electrical power without expending propellant, and established the physics basis that motivated subsequent feasibility studies for outer planet applications.

Interest in applying EDT propulsion to the Jovian system followed from Jupiter's exceptional magnetic environment. Gallagher et al. [6] identified Jupiter as the most favorable planetary target for EDT propulsion, showing that the combination of strong magnetic field and dense plasma could produce forces exceeding 50 N and power generation on the order of megawatts for tether lengths of 10 to 50 km, potentially serving as an alternative to radioisotope thermoelectric generators. Sanmartin et al. [7] subsequently demonstrated that EDT braking during Jupiter orbit insertion could displace 0.5 to 2 km/s of conventional chemical ΔV, quantifying the mass advantage of a propellantless capture architecture. Sanmartin et al. [8, 11] further characterized EDT performance across the broader Jovian moon system, establishing the propulsive and power generation potential at multiple orbital altitudes. These results established the performance case for inner-Jovian EDT missions.

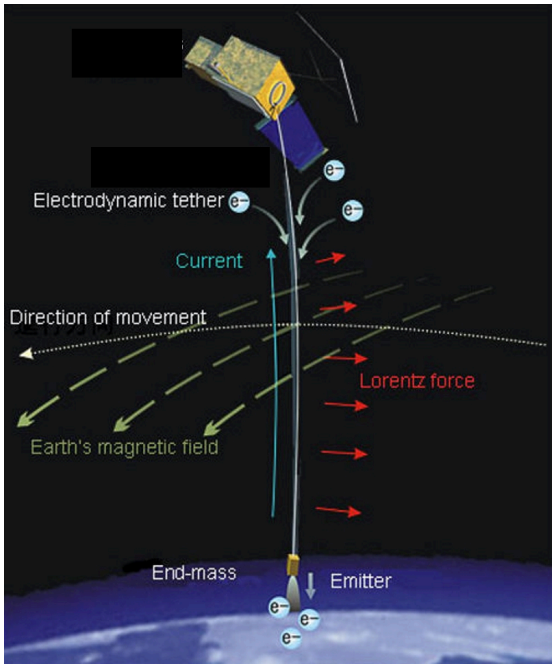


Figure 1.1 — Illustration of EDT propulsion generation within Earth's magnetic field.

However, subsequent structural and thermal analyses revealed that the long single tethers proposed in early feasibility studies would not survive Jupiter's plasma-magnetosphere. Sanmartin et al. [7] and Casalino et al. [12] determined that intense electron collection imposes thermal constraints limiting the maximum survivable tether length to approximately 9.2 km for titanium construction. This finding invalidated the 20 to 50 km single-tether configurations and necessitated the Multi-EDT (MEDT) architecture, in which multiple shorter tethers operate in combination to replicate the performance of a longer single tether while remaining within the survivability envelope [12]. The MEDT constraint directly shapes the hardware configuration adopted in this study.



Figure 1.2 — Conceptual illustration of a spacecraft equipped with a Multi-EDT system, showing the ladder-type tether configuration adopted to satisfy single-tether survivability constraints in the Jovian plasma environment. Adapted from Casalino et al. [12].

Parallel to the hardware evolution, trajectory optimization methodology for EDT Jovian missions matured from heuristic control laws toward mathematically rigorous optimal control formulations. Williams [13] demonstrated optimal orbit transfer strategies using electrodynamic tethers, establishing early analytical foundations for EDT trajectory design. Alkhenani and Oguri [1] subsequently applied Pontryagin's minimum principle to derive analytical optimal control laws for EDT-propelled Jupiter orbit insertion, establishing the indirect method as the appropriate framework for EDT trajectory optimization and providing the methodological foundation upon which the present study builds. Kojima and Trivailo [14] further evaluated EDT transition performance during Jupiter orbit capture, confirming the feasibility of the propellantless capture approach across a range of configurations. Despite these advances, the core hardware configuration required for high-performance Jovian maneuvering — the bare tether integrated with hollow cathode plasma contactors — remains at a critical technology readiness level gap pending on-orbit demonstration by the planned E.T.PACK mission [10]. Establishing the trajectory feasibility of the EDT-manifold connection demonstrated here provides the analytical foundation that a future high-TRL EDT system would require to execute this mission.

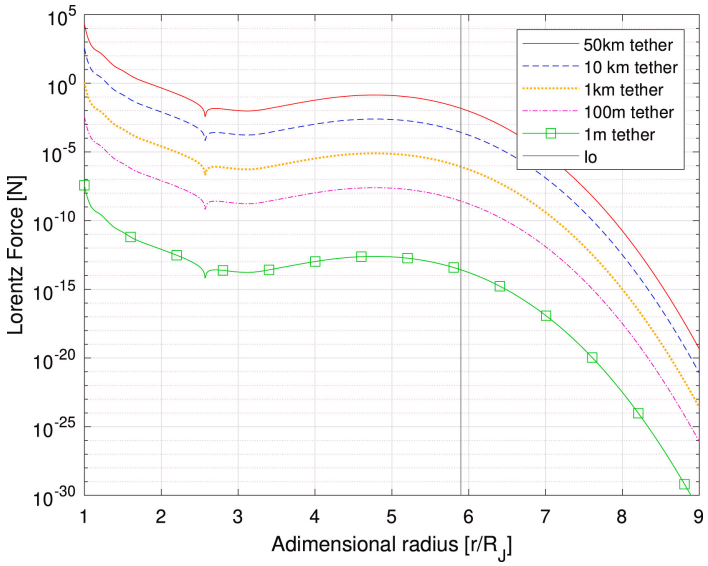


Figure 1.3 — Lorentz force as a function of radial distance from Jupiter for tether lengths ranging from 1 m to 50 km, illustrating the strong dependence of EDT performance on both orbital altitude and tether length. Adapted from Casalino et al. [12].

1.2.2 Low-Energy Transfer Design in the Jovian System

To understand the transfer architecture this study demonstrates, it is necessary to first establish what an invariant manifold is in the context of orbital mechanics and why it enables propellantless transfer. In the CR3BP, the collinear Lagrange points — particularly L1 and L2, which sit between and beyond each moon respectively — are unstable saddle points of the gravitational potential. Around each exists a family of planar periodic orbits called Lyapunov

orbits. These orbits are themselves unstable: a spacecraft placed exactly on one will loop around the Lagrange point indefinitely, but any perturbation, however small, will cause the trajectory to depart. This instability is the source of the manifold structure that makes low-energy transfer possible.

The instability of a Lyapunov orbit produces two distinct surfaces in the six-dimensional phase space of position and velocity. The stable manifold W^s is the collection of all trajectories that naturally approach the Lyapunov orbit forward in time, flowing inward through the neck region near the Lagrange point. The unstable manifold W^u is the collection of all trajectories that naturally depart from it, flowing outward through the neck into either the moon's gravitational regime or back into the outer system. Both surfaces are invariant: a spacecraft placed exactly on either one remains on it indefinitely under the unperturbed dynamics, requiring no propulsion [15]. A critical property is that these manifolds do not attract nearby trajectories. A spacecraft close to but not exactly on a manifold surface will diverge away from it exponentially, at a rate set by the instability of the underlying Lyapunov orbit. The manifold is a boundary, not a drain — a spacecraft cannot passively drift onto it from a nearby trajectory but must be steered onto the surface deliberately, with sufficient precision that the remaining gap is within the available correction capability. In this study, that correction capability is provided by the EDT.

The transfer strategy enabled by this structure proceeds as follows. A spacecraft is steered onto the stable manifold W^s of an L1 Lyapunov orbit, which then carries it naturally through the L1 neck toward the Lyapunov orbit — this transit requires no thrust, but the initial placement onto W^s does. To continue toward Io, the spacecraft must then transition onto an unstable manifold trajectory W^u departing through the neck; W^s and W^u are distinct surfaces in phase space, and crossing the gap between them requires a second deliberate maneuver. A transfer in which W^s and W^u intersect with a gap of exactly zero is called a heteroclinic connection; at that precise intersection the transit between them requires essentially zero deterministic ΔV [9, 15], though the maneuvers to initially place the spacecraft onto W^s and to depart onto W^u are still required. A finite gap quantifies the cost of the bridging maneuver. The Poincaré section is the tool used to find these intersections, providing a two-dimensional snapshot of where trajectories cross a chosen plane in the rotating frame and allowing direct comparison of the stable manifold surface against the spacecraft's trajectory [15, 16].

The theoretical foundation for this framework was established by Koon et al. [15], who demonstrated that the stable and unstable manifolds of periodic orbits around the collinear Lagrange points form natural transport corridors connecting distinct gravitational regimes, and showed that heteroclinic connections between manifold tubes of adjacent moon systems could enable transfers requiring only tens of meters per second of ΔV. Lantoine and Russell [17] extended this to near-ballistic halo-to-halo transfers between planetary moons, demonstrating that finite but small phase-space gaps could be bridged with negligible deterministic cost.

Application to the Jovian system followed from recognition of the Galilean moon system as an exceptionally rich manifold network. Vaquero and Howell [18, 19, 20] catalogued resonant orbit families and their manifolds, demonstrating efficient moon-to-moon transfer via resonance transitions, with Vaquero [16] providing the foundational Poincaré section methodology used throughout this work. Lantoine et al. [9] extended this to multi-moon tour optimization, showing that resonant hopping transfers between Jovian moons could achieve full tour sequences with total deterministic ΔV on the order of tens of meters per second, and Campanella et al. [21] demonstrated practical resonance hopping transfers between moon science orbits. Campagnola and Kawakatsu [22] demonstrated practical application in the Jupiter Magnetospheric Orbiter concept. Bombardelli et al. [23, 24] further characterized Lyapunov and halo orbit families in the Jupiter-moon systems as waypoints for science observation and inter-moon transfer, and Bombardelli and Peláez [25] established the stability properties of artificial equilibrium points in the CR3BP that underpin the manifold targeting methodology used here.

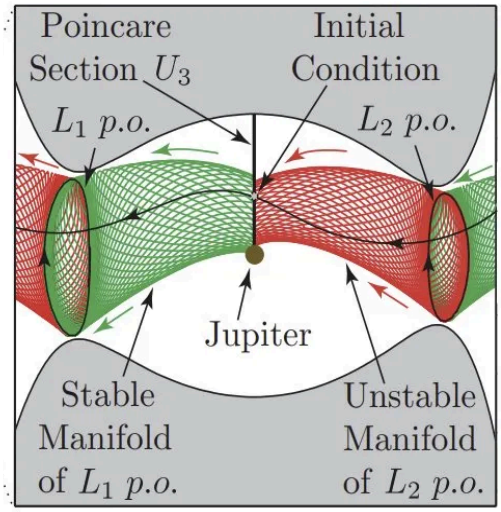


Figure 1.4 — Stable and unstable manifold tubes of the L1 and L2 periodic orbits in the CR3BP, illustrating the natural transport corridors that form the basis of low-energy transfer design. Adapted from Koon et al. [15].

The InterMoon Superhighway concept views the Jovian moon system as an integrated network of these low-energy pathways [9, 15]. Europa Clipper and JUICE have validated resonant orbit strategies in operational mission design [10], confirming that manifold-based transfer architectures are viable engineering tools. The remaining open question — whether propellantless EDT forcing can serve as the mechanism for accessing this network from a post-capture orbit — is addressed in the following section.

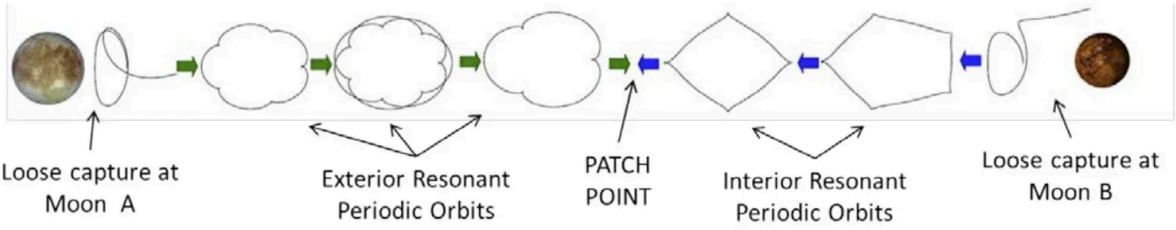


Figure 1.5 — The InterMoon Superhighway trajectory concept, illustrating transfer between two planetary moons via loose capture, exterior and interior resonant periodic orbits, and a patch point connecting the two manifold systems. Adapted from Lantoine et al. [8].

1.2.3 The Synthesis Gap: EDT Propulsion and Manifold Targeting

The two research areas reviewed above have largely developed in parallel. The Magnetour mission concept, introduced by Lantoine et al. [8] in a 2012 NASA NIAC Phase 1 study, represents the most significant prior attempt to synthesize them. Magnetour proposed combining EDT electromagnetic braking for planetary capture with the InterMoon Superhighway framework for multi-moon tour design, envisioning a spacecraft capable of conducting long-duration observations across multiple Galilean moons propellantlessly. The concept demonstrated physical plausibility and identified the key mission phases: electromagnetic capture, apojove pump-down, perijove raising via gravity assists, and low-energy inter-moon transfers.

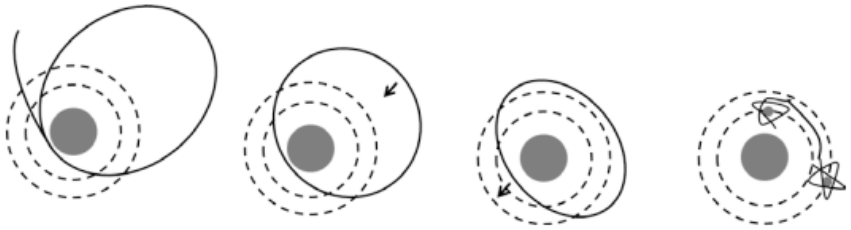


Figure 1.6 — Phases of Magnetour: left to right: capture; lowering apojove; raising perijove; and low-energy inter-moon transfer and loosely captured orbits.

The study delineated these four phases as sequential, treating the EDT and the manifold-based IMS machinery as operating in non-overlapping regimes: the EDT handled energy reduction in the early phases while heteroclinic connections and invariant manifolds governed the low-energy tour. Magnetour explicitly noted that the effect of EDT forcing on the stable and unstable manifolds of periodic orbits themselves and on the resonant transfer orbits of the IMS had not been analyzed, and that such knowledge is required for a full end-to-end mission design with tethers. As a direct consequence, the EDT's capacity to serve as a precision steering mechanism onto invariant manifold tubes was never demonstrated [8]. Additionally, the tether-perturbed Lyapunov orbit families at Io were computed only for tether lengths well beyond what survivability constraints permit; the study itself acknowledged that achievable tether

lengths are likely below 100 km, a regime in which EDT-induced orbital perturbation is minimal, and the survivability constraints subsequently established by Casalino et al. [12] and Sanmartin et al. [7] further invalidated the longer configurations on which that analysis depended.

Subsequent work advanced individual components without closing this gap. Alkhenani and Oguri [1, 26] developed rigorous optimal control laws for EDT-propelled JOI capture using the indirect method, establishing the methodological foundation for EDT trajectory optimization in the Jovian system, but focused exclusively on the capture maneuver rather than the subsequent manifold transit. Yang et al. [27] identified EDT-manifold integration as an open research direction in a survey of Jovian trajectory design approaches, and the problem has remained unresolved in the literature since.

This study addresses that gap directly. By computing the stable manifold of the Jupiter-Io L1 Lyapunov orbit, characterizing the phase-space gap between that manifold and the post-pump-down capture trajectory on the Poincaré section, and demonstrating that EDT forcing can close that gap propellantlessly within hardware-validated survivability constraints, this work provides the first quantification of the EDT-to-manifold bridging problem in the inner Jovian system, which is specifically the precision targeting role that Magnetour identified as necessary but left undemonstrated.

1.3 Project Proposal

The mission sequence this study addresses proceeds in four phases. Following Jupiter orbit insertion via a single chemical capture burn, the spacecraft occupies a large elliptical capture orbit with apojove well beyond Io. Phase 1, referred to as apojove pump-down, uses the EDT to extract orbital energy at each perijove pass, progressively shrinking the capture orbit until the energy threshold for L1 gateway access is reached. Phase 2 is the manifold entry maneuver: a small EDT steering correction places the spacecraft onto the stable manifold of the L1 Lyapunov orbit, which then carries it naturally through the L1 neck toward Io's vicinity without additional propulsive cost. Once at the L1 gateway, a second small EDT maneuver is required to hop from the stable manifold onto a departing unstable manifold trajectory — the two surfaces are distinct and the spacecraft does not transition between them automatically, so Phase 3 refers to that transit through the L1 gateway. Phase 4, Io orbit insertion and surface approach, lies outside this study's scope.

This project mathematically derives and numerically demonstrates that Phases 1 and 2 are achievable with a physically realizable EDT system operating within hardware survivability constraints, directly addressing the synthesis gap identified in Section 1.2.3. The analysis operates within the circular restricted three-body problem with low-thrust forcing framework, treating EDT acceleration as a continuous perturbation subject to a fixed power budget. The scope is bounded at pre-Phase A feasibility: the central result is a quantified near-heteroclinic

connection between the post-JOI capture orbit and the Jupiter-Io L1 stable manifold, establishing propellantless manifold transit as a rigorous dynamical result and providing the foundational analysis upon which higher-fidelity mission development depends.

1.4 Methodology

The study proceeds through four sequential stages, each building on the outputs of the previous. The first stage establishes a two-body patched-conic baseline that quantifies the ΔV requirement for conventional chemical Jupiter orbit insertion and produces the post-JOI boundary states that initialize the low-energy manifold analysis. These outputs serve two purposes: they establish the performance justification for the propellantless EDT architecture, and they produce the boundary states that are subsequently normalized to canonical CR3BP units to seed the manifold search. The second stage develops the CR3BP-LT equations of motion in the Jupiter-Io synodic rotating frame, incorporating the gravitational potentials of both primaries, the fictitious forces arising from the rotating frame, and the Lorentz acceleration of the EDT system as a continuous perturbing force. The EDT force model adheres to hardware survivability constraints established by prior feasibility work, and an altitude-dependent plasma electron density model is incorporated into the current magnitude calculation. The third stage locates and characterizes the dynamical structures that enable propellantless transfer. The unstable collinear Lagrange points are located numerically, and a differential corrector refines linearized initial guesses into fully converged Lyapunov orbits. The monodromy matrix is computed over one full orbit period, its eigenstructure extracted to identify the hyperbolic stable and unstable directions, and the invariant manifold tubes propagated from those directions. The fourth stage demonstrates the near-heteroclinic connection. A Poincaré surface of the section is constructed at the y=0 plane in the rotating frame, and the intersections of the post-pump-down capture trajectory and the stable manifold are mapped onto it. An arrival angle sweep identifies the geometries that minimize the phase-space gap, and the EDT force is shown to provide the controlled perturbation necessary to bridge that gap propellantlessly. A sensitivity sweep across hardware configurations then characterizes the feasibility envelope.

2. Problem Formulation

2.1 Overview

This chapter develops the analytical framework used to demonstrate propellantless transfer from Jupiter orbit insertion to the L1 stable manifold of the Jupiter-Io system. The framework proceeds in four stages. First, a two-body patched-conic baseline establishes the post-JOI boundary states and quantifies the propellant cost of conventional capture, motivating the EDT architecture. Second, the CR3BP-LT provides the equations of motion in the Jupiter-Io rotating frame, incorporating EDT acceleration as a non-conservative perturbation. Third, variational analysis of the CR3BP equations yields the periodic orbit and invariant manifold structures that define the low-energy transfer pathways. Fourth, a Poincaré section analysis reduces the manifold geometry to a tractable targeting problem and quantifies the phase-space gap the EDT must bridge.

The scope is bounded at pre-Phase A feasibility, where Phase A refers to the concept study phase of the NASA mission lifecycle in which mission feasibility is formally assessed prior to preliminary design. Demonstrating that a propellantless manifold connection is dynamically achievable within this framework is the foundational result upon which higher-fidelity analysis depends.

2.2 Orbit Transfer Problem Statement

The central problem addressed by this study is as follows. A spacecraft arrives at Jupiter with hyperbolic excess velocity and must be captured into orbit, transferred to the vicinity of Io, and synchronized with Io's orbit, all without expending propellant beyond the initial capture burn. The transfer exploits the natural low-energy pathways of the Jupiter-Io gravitational system, with an EDT providing the controlled perturbation needed to access those pathways from the post-capture orbit.

The transfer proceeds in four phases, illustrated in Figure 2.1. Following JOI, the spacecraft occupies a large capture ellipse with perijove near Jupiter and apojove well beyond Io's orbit. The Jacobi constant of this orbit, a conserved energy-like quantity in the rotating frame introduced in Section 2.4, exceeds the threshold at which the L1 gateway between the Jupiter and Io gravitational regimes is topologically open. The spacecraft cannot reach the Io vicinity regardless of its trajectory geometry until this energy condition is satisfied. Phase 1, referred to throughout as pump-down, addresses this condition. The EDT applies a braking force at each perijove pass, extracting orbital energy from the spacecraft and monotonically reducing the Jacobi constant orbit by orbit. The term pump-down refers to this progressive reduction of orbital energy, analogous to bleeding pressure from a system. As the apojove shrinks with each pass, the

capture orbit family contracts toward Io's orbit. Phase 1 continues until the Jacobi constant falls below the L1 threshold, at which point the neck region near L1 opens and the stable manifold of the L1 Lyapunov orbit becomes accessible from the spacecraft's location in phase space. Phase 2 is the manifold entry maneuver. The stable manifold of the L1 Lyapunov orbit is the set of all states in phase space that naturally approach that orbit forward in time. A spacecraft placed on this surface arrives at the L1 gateway without additional propulsive cost. Since the post-pump-down trajectory does not in general lie exactly on this surface, a small EDT steering maneuver bridges the residual gap. The size of this gap, and the arrival geometry that minimizes it, are the primary outputs of the analysis in Chapter 3. Phase 3 represents transit via a manifold hop through the L1 gateway and approach toward Io along the manifold. Phase 4, Io orbit insertion and surface approach, lies outside the scope of this study and is left for subsequent higher-fidelity analysis.

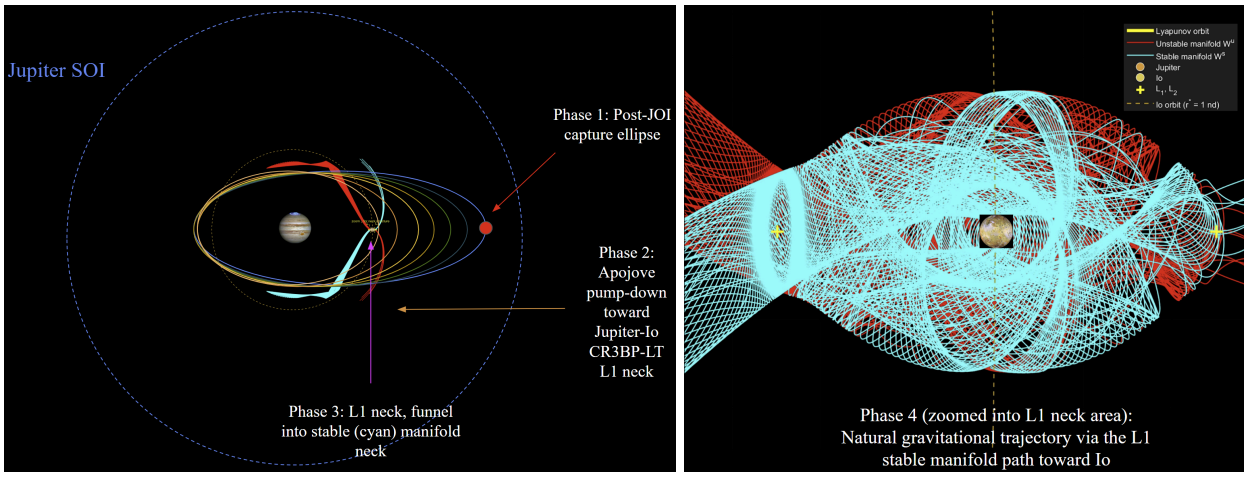


Figure 2.1 — Schematic of the four-phase propellantless transfer from JOI capture to Io approach via EDT pump-down and L1 manifold transit.

The analytical objective of this study is to demonstrate that Phases 1 and 2 are dynamically achievable within the hardware constraints of the EDT system, establishing a near-heteroclinic connection between the post-JOI capture orbit and the L1 stable manifold as a pre-Phase A result.

2.3 Two-Body and Patched Conic Baseline

Before adopting the multi-body framework required for low-energy transfer design, a two-body patched-conic baseline is established to quantify the propellant cost of conventional Jupiter capture and to produce the boundary states that initialize the CR3BP-LT analysis. The transfer is modeled using a planar patched-conic method, which partitions the mission into three sequential two-body regimes: Earth-local departure, heliocentric transfer, and Jupiter-local

capture [28]. Within each regime, the motion is governed by a single dominant gravitational body, and the vis-viva equation provides the orbital velocity at any point along the trajectory:

$$v^2 = \mu\left(\frac{2}{r} - \frac{1}{a}\right) \tag{2.1}$$

where μ is the gravitational parameter of the central body, r is the radial distance, and a is the semi-major axis. Lambert's problem is used to determine the heliocentric transfer elements for a chosen departure and arrival epoch, yielding the hyperbolic excess velocity at Jupiter's sphere of influence. The sphere of influence radius defines the boundary at which the heliocentric regime transitions to the Jupiter-local regime and is given by:

$$r_{SOI} = a\left(\frac{m}{M}\right)^{\frac{2}{5}} \tag{2.2}$$

where m and M are the masses of Jupiter and the Sun respectively. At the sphere of influence, the residual speed relative to Jupiter is the hyperbolic excess velocity v_∞, which is computed from the difference between the spacecraft's heliocentric transfer speed and Jupiter's circular orbital speed at the SOI, following standard orbital energy conservation [28]. This quantity drives the impulsive capture ΔV required to enter a bound orbit at the chosen perijove radius.

The two-body baseline serves to establish the post-JOI boundary state that initializes the CR3BP-LT analysis, and to provide context for the propellant budget the EDT architecture is intended to eliminate. Critically, the JOI capture burn itself is chemical in both architectures — the EDT becomes effective only after capture, once the spacecraft is making repeated perijove passes through the strong inner Jovian magnetic field. The propellant comparison therefore quantifies what a conventional spacecraft would need to spend *after* capture to shrink its orbit to Io's vicinity, which is the segment the EDT eliminates entirely.

2.4 Dynamical Formulation: CR3BP-LT Equations of Motion

2.4.1 The Circular Restricted Three-Body Problem

The CR3BP models the motion of a massless spacecraft under the gravitational influence of two primary bodies, Jupiter and Io, orbiting their common barycenter in a circular orbit. This study adopts the planar CR3BP, restricting motion to the Jupiter-Io orbital plane. This assumption is appropriate at pre-Phase A fidelity because the Lyapunov orbits and invariant manifolds central to the transfer design are planar structures, and the Jovian magnetic dipole is strongest in the equatorial plane where EDT performance is maximized. Out-of-plane dynamics are identified as a follow-on analysis item.

The problem is formulated in the Jupiter-Io synodic rotating frame, in which the two primaries are fixed on the x-axis. The system is normalized using canonical units where the Jupiter-Io distance is 1 length unit (LU), the mean motion is 1 radian per time unit (TU), and the total system mass is 1. The mass ratio $\mu = \frac{\mu_{Io}}{\mu_{Jupiter}+\mu_{Io}}$ is the single parameter governing the gravitational geometry.

Applying Newton's second law in the rotating frame and accounting for the fictitious Coriolis and centrifugal forces gives the general equation of motion:

$$\ddot{\mathbf{r}} + 2\boldsymbol{\Omega} \times \dot{\mathbf{r}} = \nabla U \tag{2.3}$$

where Ω is the system angular velocity, normalized to 1 in canonical units, and ∇U is the gradient of the pseudo-potential function. Expanding to the planar case yields:

$$\ddot{x} - 2\dot{y} = U_x \tag{2.4}$$

$$\ddot{y} - 2\dot{x} = U_y \tag{2.5}$$

where subscripts denote partial derivatives and U is the pseudo-potential function:

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

with r_1 and r_2 representing the distances from the spacecraft to Jupiter and Io respectively. The Coriolis terms $2\dot{y}$ and $2\dot{x}$ arise from the rotating frame and break the analytic integrability of the equations of motion, requiring numerical propagation [15]. The state-space formulation used for numerical integration is presented in Section 3.2.

2.4.2 The Jacobi Constant and Zero-Velocity Surfaces

The CR3BP admits one known integral of motion, the Jacobi constant:

$$C = 2U - v^2 \tag{2.7}$$

where $v^2 = \dot{x}^2 + \dot{y}^2$ is the squared speed in the rotating frame. Rearranging reveals the maximum kinetic energy a spacecraft can possess at any position for a given energy level C:

$$v^2 = 2U - C \tag{2.8}$$

Setting v = 0 defines the zero-velocity surfaces, the boundaries of the permitted region of motion:

$$C = 2U \qquad \text{or equivalently} \qquad C - 2U = 0 \tag{2.9}$$

The spacecraft can only move where $C-2U \leq 0$, that is, where kinetic energy remains non-negative. The zero-velocity surfaces partition phase space into distinct regions: the inner realm around Io, the neck regions near L1 and L2, and the outer realm around Jupiter.

The collinear Lagrange points L1 and L2, located on the x-axis between and beyond the primaries respectively, are saddle points of U. Each defines a critical energy threshold. At L1, located at position x_{L1} along the rotating frame x-axis, the neck opening threshold is:

$$C_{L1} = 2u(x_{L1}, 0, 0) = x_{L1}^{\ 2} + \frac{2(1-\mu)}{r_{1,L1}} + \frac{2\mu}{r_{2,L1}} \tag{2.10}$$

where $r_{1,L1}=|x_{L1}+\mu|$ and $r_{2,L1}=|x_{L1}-1+\mu|$ are the distances from L1 to Jupiter and Io respectively. This threshold is determined entirely by the gravitational geometry of the Jupiter-Io system, independent of spacecraft design choices [15]. When $C > C_{L1}$, the zero-velocity surface is closed at L1 and no trajectory can pass between the Jupiter and Io gravitational regimes. When $C < C_{L1}$, the surface opens at L1, creating a corridor through which transit trajectories can pass. This topology change is a fundamental consequence of the effective potential structure: L1 is the saddle point of U at which the forbidden region first pinches open.

This topological condition is the mathematical justification for why EDT pump-down must precede the manifold hop. The post-JOI capture state enters the Jupiter-Io rotating frame with a Jacobi constant above C_{L1}, so the stable manifold W^s of the Lyapunov orbit is geometrically inaccessible regardless of arrival angle or phasing. No trajectory optimization can thread the manifold connection while this topological barrier exists. The EDT pump-down reduces C monotonically by extracting orbital energy at each perijove pass until C drops below C_{L1}, at which point the neck opens and the manifold becomes geometrically accessible.

The total energy gap the EDT must close is

$$\Delta C_{total} = C_{cap} - C_{Lyapunov} \tag{2.11}$$

The fraction of this gap closed by Phase 1 pump-down determines the residual Phase 2 steering requirement. A pump-down that closes the full gap places the spacecraft energetically on the Lyapunov orbit manifold; a partial pump-down leaves a residual gap that must be bridged by EDT steering in Phase 2. This energy accounting framework connects the topological condition derived here directly to the quantitative results of Chapter 3.

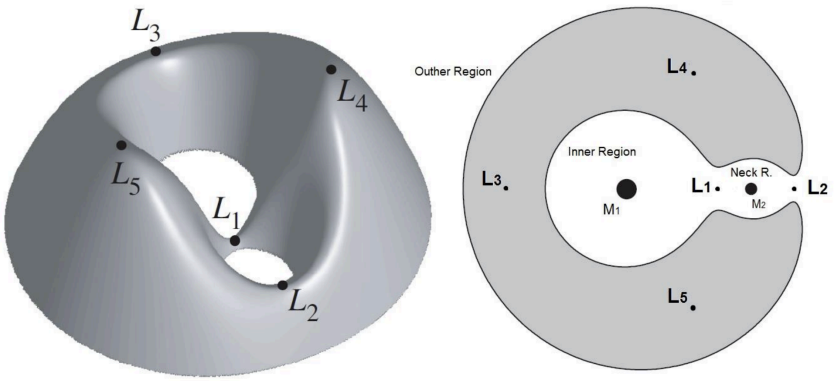


Figure 2.2 — Potential surface U(x,y) and zero velocity curves with superimposed equilibrium points.

This establishes the first necessary condition for manifold access: $C < C_{L1}$. Until this condition is satisfied, the stable manifold W^s is geometrically unreachable regardless of spacecraft trajectory geometry or arrival phasing. The EDT pump-down of Phase 1 exists solely to drive C below this threshold.

2.4.3 EDT Forcing and the CR3BP-LT Model

The EDT system introduces a non-conservative perturbation into the CR3BP equations of motion. The scalar magnitude of the Lorentz force acting on a current-carrying tether of length L moving through a magnetic field B is:

$$F_{EDT} = ILB \tag{2.12}$$

where I is the tether current; the directionality is discussed separately in Section 3.6. This force produces a perturbing acceleration $a_{EDT} = \frac{F_{EDT}}{m_{SC}}$ which is added to the right-hand side of the equations of motion:

$$\ddot{x} - 2\dot{y} = U_x + a_{EDT,x} \tag{2.13}$$

$$\ddot{y} + 2\dot{x} = U_y + a_{EDT,y} \tag{2.14}$$

The tether current is not a free parameter. As the tether moves through Jupiter's magnetic field with relative velocity v_{rel} between the spacecraft and the co-rotating plasma, a motional EMF is induced:

$$V_{motional} = B * L * v_{rel} \tag{2.15}$$

The power-limited current is then:

$$I = min(\frac{P_{max}}{V_{motional}}, I_{max}) \quad (2.16)$$

where P_{max} is the available power and I_{max} is the physical current ceiling imposed by tether material properties. Both are fixed parameters derived from the hardware configuration detailed in Appendix B and carried as constants throughout the analysis. The magnetic field strength B is evaluated using a magnetic dipole model, in which field strength decays as $B \propto 1/r^3$ with distance from Jupiter [29]. The dipole approximation is appropriate at Io's orbital radius, where the Jovian field is well-characterized and dominates over higher-order multipole terms [7].

The introduction of a_{EDT} breaks the conservation of the Jacobi constant. C is no longer an integral of motion under EDT forcing; it decreases monotonically as the tether extracts orbital energy at each perijove pass. The manifold geometry used for trajectory targeting is computed from the unperturbed CR3BP. This manifold-as-template assumption holds when the EDT perturbation is small relative to the gravitational dynamics, such that the invariant structures of the unforced system remain a valid approximation of the true dynamical geometry under continuous forcing. As a consequence, the gap metric at the Poincaré section measures the distance between a naturally computed gravitational structure and an EDT-perturbed trajectory, and is minimized because closing the gap places the perturbed spacecraft onto the natural manifold, after which no further forcing is needed. This is stated explicitly as a pre-Phase A modeling choice and is identified as a follow-on analysis item.

2.5 Periodic Orbits and Invariant Manifolds

With the CR3BP-LT environment established, the analysis shifts to locating and characterizing the dynamical structures that enable propellantless transfer. This section derives the periodic orbits and invariant manifolds that form the low-energy transport corridors of the transfer design.

2.5.1 Lagrange Points and Flow Classification

The Lagrange points are the five equilibrium solutions in the rotating frame where gravitational and centrifugal forces balance [15]. They are computed by setting the velocity and acceleration components of the equations of motion to zero. The equilibria are split into two groups: the stable triangular points L4 and L5, which host captured objects such as Trojan asteroids, and the unstable collinear points L1, L2, and L3. Their instability generates the periodic orbits and invariant manifolds required for low ΔV transfers.

Linearization of the CR3BP equations of motion around the collinear points classifies the local flow into four types: periodic orbits about L1 and L2, asymptotic trajectories that approach

or depart those orbits along the invariant manifolds, transit trajectories that pass through the neck region between gravitational regimes, and non-transit trajectories that remain bounded within a single regime [15]. The periodic orbits and their manifolds are the objects of interest for transfer design.

This study uses L1 exclusively as the transfer gateway. L1 sits between Jupiter and Io at approximately 411,400 km from Jupiter — about 10,400 km inward from Io's orbital radius — and its stable manifold tube extends back into the Jupiter realm, making it the natural inbound corridor from the post-capture orbit. L2, located on the far side of Io, has a stable manifold that approaches from the Io-dominated region rather than from Jupiter, making it geometrically inaccessible from the post-capture orbit without first passing through the Io realm. In the flow classification above, the post-JOI capture trajectory is a non-transit trajectory: as established in Section 2.4.2, C exceeds C_L1 and the neck is topologically closed. The goal of Phase 1 pump-down is to transition the spacecraft from this non-transit state to a transit-capable one, at which point the flow classification changes from non-transit to asymptotic.

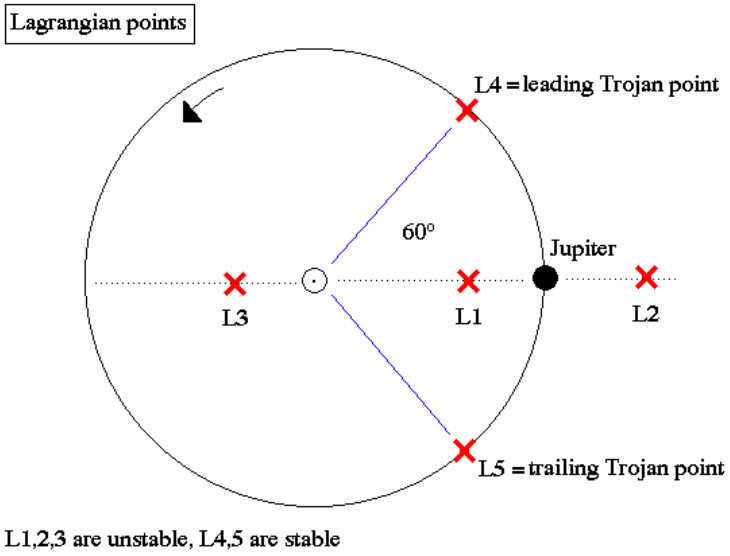


Figure 2.3 — Schematic of Lagrange points within general CR3BP geometry.

2.5.2 Lyapunov Orbits and the Differential Corrector

Lyapunov orbits are planar periodic orbits in the CR3BP that exist around the collinear Lagrange points [15]. They are unstable: a spacecraft placed exactly on the orbit remains there indefinitely, but any perturbation grows exponentially. This instability is what generates the manifold structure exploited for low-energy transfer.

Accurate initial conditions for Lyapunov orbits are obtained in two steps. The linearized equilibrium analysis at L1 provides the initial guess: the in-plane Lyapunov frequency $\omega_{Lyapunov}$ is derived from the eigenvalues of the linearized equations of motion at L1 and the

amplitude-velocity ratio $\kappa = \frac{\omega^2_{Lyapunov} - U_{xx}}{2\omega_{Lyapunov}}$ relates the transverse velocity to the displacement amplitude, giving $\dot{y} = -\kappa A_x$ as the initial velocity estimate for a chosen amplitude A_x. A single-shooter differential corrector then refines this guess to full numerical precision by iteratively correcting the initial state using the state transition matrix until the trajectory closes on itself to within a specific tolerance [15, 30]. The converged orbit and its period are the inputs to the manifold construction procedure that follows. The target orbit is selected at the lowest-energy end of the accessible L1 Lyapunov family, as this minimizes the total energy gap ΔC_{total} the EDT must close, per the accounting established in Section 2.4.2. Amplitude and Jacobi constant move in one-to-one correspondence along the family: larger amplitude corresponds to lower Jacobi constant, so the target amplitude is a consequence of the energy matching requirement rather than an independent design choice.

The converged Lyapunov orbit is itself a hyperbolic object; its instability is not incidental but structural. Nearby trajectories diverge exponentially in specific directions determined by the orbit's stability eigenstructure, whose directions define the stable manifold W^s and unstable manifold W^u. Identifying them requires computing the eigenstructure of the monodromy matrix, which is the subject of the following section.

2.5.3 The Variational Equations and Monodromy Matrix

With the Lyapunov orbit converged, the second condition for transfer design can now be stated: a trajectory is on the stable manifold W^s if and only if its initial state lies along the stable eigenvector direction of the monodromy matrix Φ, the STM over one full period at a point on the orbit [15]. Identifying that direction requires computing Φ and extracting its eigenstructure. The derivation that follows establishes this eigenstructure and shows that $\det(\Phi) = 1$ is both a structural guarantee of the Hamiltonian system and a numerical check that the implementation is correct. To extract the manifold directions from a converged Lyapunov orbit, the stability of the orbit must be quantified. This is accomplished through the variational equations, derived by introducing a small perturbation $\delta\mathbf{X}$ to the reference orbit state $\mathbf{X}(t)$:

$$\delta\dot{X} = A(t)\delta X \tag{2.17}$$

where A(t) is the 6x6 Jacobian matrix of the CR3BP vector field evaluated along the reference orbit:

$$A(t) = \frac{\partial f}{\partial X}\Big|_{X(t)} \tag{2.18}$$

For the planar CR3BP, A(t) takes the explicit block structure:

$$A(t) = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{U} & \mathbf{\Omega} \end{bmatrix} \tag{2.19}$$

where **U** is the 3×3 matrix of second partial derivatives of the pseudo-potential, and **Ω** captures the Coriolis terms arising from the rotating frame:

$$\mathbf{U} = \begin{bmatrix} U_{xx} & U_{xy} & U_{xz} \\ U_{xy} & U_{yy} & U_{yz} \\ U_{xz} & U_{yz} & U_{zz} \end{bmatrix} \tag{2.20}$$

$$\mathbf{\Omega} = \begin{bmatrix} 0 & 2 & 0 \\ -2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{2.21}$$

The second partial derivatives of the pseudo-potential evaluated along the orbit are:

$$U_{xx} = 1 - (1 - \mu)\left(\frac{1}{r_1^3} - \frac{3(x+\mu)^2}{r_1^5}\right) - \mu\left(\frac{1}{r_2^3} - \frac{3(x-1+\mu)^2}{r_2^5}\right) \tag{2.22}$$

$$U_{yy} = 1 - \frac{(1-\mu)}{r_1^3} - \frac{\mu}{r_2^3} \tag{2.23}$$

$$U_{xy} = 3\left[\frac{(1-\mu)(x+\mu)y}{r_1^5} + \frac{\mu(x-1+\mu)y}{r_2^5}\right] \tag{2.24}$$

Since $r_1(t)$ and $r_2(t)$ vary along the reference orbit, A(t) is periodic with the same period T. This periodicity is what makes the monodromy matrix the appropriate tool for quantifying orbital stability [15].

The solution is expressed through the state transition matrix $\Phi(t,t_0)$, which propagates a perturbation from time t_0 to time t. The STM is propagated numerically alongside the trajectory as part of a combined state and variational system:

$$\delta X(t) = \Phi(t, t_0)\delta X(t_0) \tag{2.25}$$

where Φ satisfies:

$$\dot{\Phi}(t, t_0) = A(t)\Phi(t, t_0), \quad \Phi(t_0, t_0) = I \tag{2.26}$$

The monodromy matrix is the STM evaluated after exactly one full period T:

$$\Phi \equiv \Phi(T, 0) \tag{2.27}$$

Its eigenvalues, known as Floquet multipliers, quantify how a perturbation evolves over one complete revolution. A multiplier with $|\lambda|>1$ indicates exponential growth along the corresponding eigenvector, corresponding to the unstable direction. A multiplier with $|\lambda|<1$ indicates contraction, corresponding to the stable direction.

A fundamental property of Hamiltonian systems constrains the eigenvalue structure. Because the CR3BP is conservative, its flow preserves phase space volume by Liouville's theorem [15]. This follows from the fact that the divergence of the CR3BP vector field vanishes: $\nabla \cdot f=0$. This can be verified directly from the block structure of A(t): the trace tr(A) is zero by inspection, since the upper-left block is 0 and the Coriolis block Ω has zeros on its diagonal. Since $\frac{d}{dt}[det\ \Phi] = tr(A) \cdot det(\Phi)$ and tr(A) = 0, it follows that:

$$det(\Phi) = 1 \tag{2.28}$$

not merely as a numerical result but as a structural guarantee of the Hamiltonian system. As a consequence, Floquet multipliers appear in reciprocal pairs: if λ is an eigenvalue, so is $1/\lambda$. For the planar CR3BP the six multipliers appear as one hyperbolic pair $(\lambda_u, 1/\lambda_u)$ with $\lambda_u>1$, one trivial pair at unity corresponding to the along-orbit and energy-surface directions, and one center pair $e^{\pm i\theta}$ corresponding to out-of-plane neutral modes. The hyperbolic pair directly identifies the stable and unstable manifold directions that seed the manifold tube construction in the following section.

The Floquet multiplier structure establishes the manifold membership condition precisely. Any initial condition not aligned with either eigenvector direction will have components along both and will generically diverge. The practical consequence for Phase 2 is that the spacecraft's state after pump-down must be steered to within EDT bridging capability of the $\mathbf{v}_s$ direction at the Poincaré section, not merely near the Lyapunov orbit in position space, but specifically along the stable eigenvector in the full six-dimensional phase space.

2.5.4 Manifold Tube Construction

The monodromy matrix eigenstructure derived in Section 2.5.3 provides the geometric foundation for constructing the invariant manifold tubes that constitute the low-energy transfer corridors. The stable and unstable manifolds of the Lyapunov orbit are tangent to the stable and unstable eigenspaces of Φ at every point on the orbit [15].

Let $\lambda_u > 1$ be the unstable Floquet multiplier with corresponding eigenvector vu, and let $\lambda_s = 1/\lambda_u < 1$ be the stable multiplier with eigenvector $\mathbf{v}_s$. For a point $\mathbf{X}_0$ on the reference Lyapunov orbit, a small perturbation of magnitude ε applied along each eigenvector gives:

$$\mathbf{X}_0^{\pm} = \mathbf{X}_0 \pm \varepsilon\, \mathbf{v}_{u,s} \tag{2.29}$$

The behavior of each perturbed trajectory is determined by the magnitude of the corresponding Floquet multiplier. For the unstable manifold W^u, a perturbation along $\mathbf{v}_u$ integrated forward in time produces a trajectory that departs exponentially from the periodic orbit; after k revolutions the perturbation grows as $\lambda_u{}^k\varepsilon$, so the trajectory diverges at a rate set by the instability index λ_u. For the stable manifold W^s, a perturbation along $\mathbf{v}_s$ integrated backward in time produces a trajectory that approaches the periodic orbit asymptotically; the same exponential factor $\lambda_u{}^k$ now appears in the denominator, so the trajectory contracts toward the orbit at the same rate. Integrating forward in time from a $\mathbf{v}_s$ perturbation would instead produce divergence, which is why backward integration is the correct procedure for constructing W^s [15, 30]. Applying this procedure at N equally spaced points around the orbit generates 2N manifold arms for each of W^u and W^s, forming the tube structures of the type shown in Figures 2.4, 4.3, and 4.4.

The perturbation size ε must be large enough to seed the manifold above the numerical noise floor of the integrator, but small enough that the linear approximation at the orbit remains valid. A value of $\varepsilon = 10^{-6}$ nd satisfies both conditions and is the standard choice in CR3BP manifold computation, corresponding to a physical displacement on the order of hundreds of meters [15]. The integration time per arm is set to 4π nd, equivalent to two Io orbital periods, which is sufficient for the manifold arms to develop the tube geometry used for Poincare section targeting. The stable manifold W^s is the object of primary interest for the transfer design. It represents the set of all states in phase space from which the spacecraft naturally approaches the L1 Lyapunov orbit forward in time, arriving at the L1 gateway without additional propulsive cost. The unstable manifold W^u represents the departure directions away from the orbit. Together the tubes act as separatrices, dividing phase space into transit and non-transit regions [15]. A spacecraft state lying inside the tube cross-section at the Poincaré section achieves a heteroclinic connection; states outside require propulsive bridging. EDT forcing is disabled during manifold propagation to preserve the natural CR3BP geometry, consistent with the manifold-as-template assumption stated in Section 2.4.3.

2.6 Poincaré Section and Gap Metric

The invariant manifold tubes constructed in Section 2.5.4 exist as high-dimensional objects in the full phase space of the CR3BP. To make these structures tractable for trajectory targeting, their geometry must be reduced to a lower-dimensional representation that can be directly compared against the post-pump-down capture trajectory. The Poincare surface of the

section provides this reduction by recording the state of each trajectory each time it crosses a chosen hypersurface, converting the continuous flow of phase space into a discrete map that reveals the geometric relationship between the capture trajectory and the stable manifold targeting set [15, 31].

When the spacecraft crosses the Jupiter-Io connecting line moving inward, two numbers are recorded: its position along that line x* and its radial velocity ẋ* at that moment. That pair of numbers is one point on the Poincaré map. The manifold arms produce a compact cluster of such points, and the capture trajectory produces a separate scatter. The gap between the nearest points in the two clouds is the quantity Phase 2 EDT steering must close — a two-dimensional distance measurement that stands in for an otherwise six-dimensional proximity problem.

The surface of section is defined by the condition y=0 with $\dot{x} < 0$ in the Jupiter-Io synodic rotating frame. The choice of y = 0 is standard for Jupiter-moon systems and follows directly from the symmetry of the CR3BP about the rotating frame x-axis [15, 16]. In the synodic frame, y=0 corresponds to the line connecting Jupiter and Io; the spacecraft crosses this surface each time it passes through the Jupiter-Io connecting line. The additional condition $\dot{x} < 0$ selects only inward-crossing events, isolating trajectories that are approaching Jupiter from the outer system and excluding the outward return leg of each orbit. This directionality isolates exactly those trajectory segments that are geometrically relevant for manifold entry, since the stable manifold W^s of the L1 Lyapunov orbit is approached from the outer Jovian region.

The Poincaré map is constructed by propagating each stable manifold arm and the capture trajectory and recording the (x*,ẋ*) state at each y=0, $\dot{x} < 0$ crossing [24]. The resulting scatter of points in the (x*,ẋ*) plane constitutes the Poincaré section. The stable manifold W^s appears as a compact, ordered cluster near $x^*=x_{L1}$, reflecting the tube geometry of the manifold: only states within the tube cross-section project into this cluster. The capture trajectory crossings appear as a separate scatter whose position relative to the W^s cluster quantifies the phase-space separation between the two dynamical structures.

The manifold membership condition derived in Section 2.5.3 — that the spacecraft's state must lie along $\mathbf{v}_s$ — is a six-dimensional condition that cannot be directly visualized or operationally targeted. The Poincaré section reduces this to a two-dimensional check: if the spacecraft's crossing of the y=0 surface falls within the W^s cluster in the (x*, ẋ*) plane, the manifold condition is satisfied within the tube cross-section tolerance. The gap metric Δ quantifies the residual violation of this condition.

$$\Delta = min_{i,j}\sqrt{\left(x^*_{cap,i} - x^*_{W^s,j}\right)^2 + \left(\dot{x}^*_{cap,i} - \dot{x}^*_{W^s,j}\right)^2} \tag{2.30}$$

This gap has a direct physical interpretation: it is the phase-space distance the spacecraft must traverse to transition from the post-pump-down capture trajectory onto the stable manifold of the Lyapunov orbit. A gap of zero constitutes a true heteroclinic connection, achievable only at a specific launch date and arrival geometry. A finite gap quantifies the EDT steering requirement for Phase 2, in which the Lorentz force must provide the perturbation needed to bridge this separation. Since the gap is predominantly in the $\dot{x}^*$ direction — the capture trajectory and the manifold cluster overlap significantly in x* but are offset in velocity — it translates approximately to a velocity requirement via $\Delta v \approx \Delta \cdot V^*$, where V* is the canonical velocity unit. The IMS concept exploits heteroclinic connections of this type to enable multi-moon transfers requiring only tens of meters per second of total ΔV in conceptual tour designs [8, 9]; the present study demonstrates that the EDT can serve as the bridging mechanism that closes a finite but small gap of this kind propellantlessly within the Jupiter-Io system.

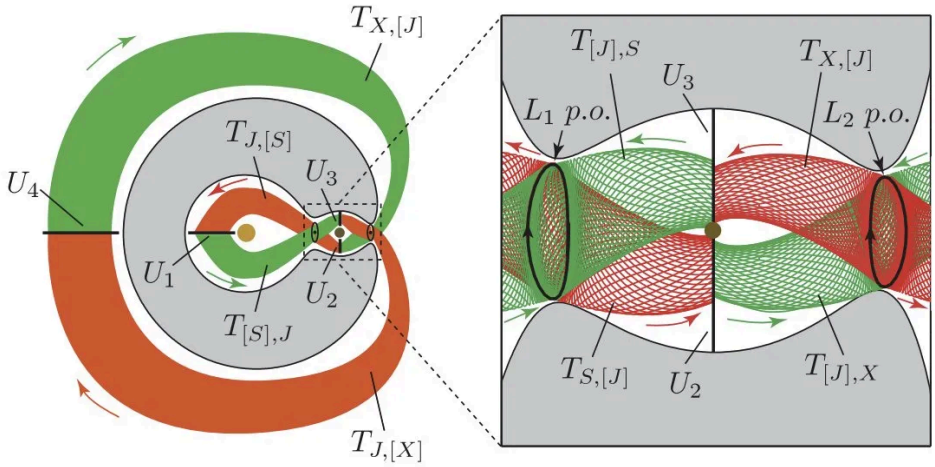


Figure 2.4 — Stable (W^s) and unstable (W^u) manifold tubes of the L1 and L2 Lyapunov orbits in the CR3BP, illustrating transit and non-transit regions in position space (left) and at a Poincaré section (right). Adapted from Koon et al. [15].

2.7 Scope and Assumptions

The modeling choices in this chapter are deliberate simplifications, each selected because they preserve the dynamical structure essential to the transfer design while keeping the analysis tractable at the concept study stage. The CR3BP-LT model is the minimum dynamical framework capable of demonstrating whether a propellantless manifold connection between the post-JOI capture orbit and the L1 stable manifold of the Jupiter-Io system is geometrically and energetically achievable, and that demonstration is the foundational result upon which all subsequent higher-fidelity analysis depends.

Seven modeling assumptions are carried throughout this study. First, motion is restricted to the Jupiter-Io orbital plane. The Lyapunov orbits and invariant manifolds central to the transfer

design are planar structures, and the Jovian magnetic dipole is strongest in the equatorial plane where EDT performance is maximized; out-of-plane dynamics are deferred to follow-on analysis. Second, the invariant manifold geometry used for trajectory targeting is computed from the unperturbed CR3BP, referred to throughout as the manifold-as-template assumption. This holds when the EDT perturbation is small relative to the gravitational dynamics, such that the manifold structures of the unforced system remain a valid approximation under continuous forcing; verification under realistic EDT force levels is a follow-on item. Third, the Jovian magnetic field is modeled as a centered dipole with field strength decaying as $B \propto 1/r^3$, neglecting the higher-order multipole terms and asymmetric field structure confirmed by Juno; escalation to the VIP4 empirical field model is identified as a follow-on item. Fourth, the tether is modeled as operating continuously throughout the pump-down phase, whereas realistic operational constraints such as attitude control windows, thermal recovery periods, and plasma density gaps near apojove impose a duty cycle below unity. The sensitivity of pump-down duration to duty cycle is characterized in Chapter 4; full duty cycle modeling is deferred. Fifth, the onboard power requirement P_{max} and maximum tether current I_{max} are carried as fixed constants derived from the provided hardware configuration parameters, with mass and tether length treated as nominal values whose effect on pump-down duration is characterized through sensitivity sweeps. Sixth, Jupiter is modeled as a point mass with a spherically symmetric gravitational field. In reality Jupiter is an oblate spheroid whose J_2 oblateness term causes perijove drift and orbital plane precession over time; these perturbations are non-negligible for an inner-Jovian mission and their effect on pump-down trajectory and manifold geometry is identified as a follow-on analysis item. Seventh, the Jovian plasma density encountered by the tether is modeled using an altitude-dependent profile based on the plasma torus structure, which captures the sharp drop-off in plasma density beyond Io's orbital radius but remains a simplified representation of the full plasma environment. A constant-density model is used for the primary pump-down simulation; the altitude-dependent model is applied in the operational realism analysis of Chapter 4 to bound the sensitivity of pump-down duration to plasma density variation.

Four analyses are explicitly deferred to subsequent higher-fidelity work: fidelity escalation from the CR3BP-LT to the planar concentric circular four-body problem (PCCFBP) and circular restricted N-body problem (CRNBP), which account for gravitational perturbations from the additional Galilean moons; Monte Carlo robustness analysis quantifying sensitivity of the manifold connection to navigation error, magnetic field uncertainty, and tether performance variation; radiation-integrated trajectory optimization treating cumulative total ionizing dose as an explicit path constraint; and full ephemeris reconstruction in a high-fidelity propagator to translate the optimal arrival angle identified in Chapter 4 to a specific launch date and trajectory. Together these extensions form the analytical agenda for subsequent mission development, building on the propellantless manifold connection established here.

3. Implementation and Validation

3.1 Overview

As noted in Section 1.4, the solution sequence of this analysis differs from the mission's operational sequence. The spacecraft executes pump-down first, then manifold entry, then transit, however, the analysis computes the manifold target first, then characterizes what pump-down is required to reach it. This chapter documents the numerical implementation of that solution sequence: the state-space formulation of the equations of motion, the ODE solver and tolerance selection, the differential corrector that produces the Lyapunov orbit, the pump-down stopping conditions, the manifold propagation parameters, the EDT force direction convention, the arrival angle parameterization, and the Poincaré section crossing detection logic. Each choice is justified so that the results of Chapter 4, presented in ConOps order for readability, are traceable to specific numerical decisions rather than implicit assumptions. Figure 3.1 shows the overall computational pipeline, illustrating how each stage feeds the next and where the two parallel branches — manifold propagation and EDT pump-down — converge at the gap metric computation, and Table 3.1 in Section 3.9 consolidates all key numerical parameters for reference. The chapter closes with validation checks confirming that the implementation satisfies the conservation properties the CR3BP equations of motion guarantee before any results are trusted.

The analysis sequence also differs from the classical manifold-targeting approach described in the literature [15, 30], in which the designer first selects a target unstable manifold departing toward the destination, then searches for a stable manifold that nearly intersects it (a heteroclinic connection), and finally back-propagates that stable manifold to determine the required entry conditions. This study inverts that sequence because the EDT pump-down endpoint is not a free design variable, but is rather determined by the hardware configuration and the energy gap that must be closed. The forward approach taken here propagates the pump-down trajectory until the L1 neck opens, then uses the Poincaré section to measure how close that endpoint comes to the stable manifold targeting set. The gap metric quantifies the distance between the post-pump-down capture trajectory and the W^s cluster at L1: closing this gap places the spacecraft onto the stable manifold, which then carries it naturally through the neck toward Io’s vicinity. What happens beyond the L1 gateway, including any transition onto a departing unstable manifold, is deferred to subsequent higher-fidelity analysis.

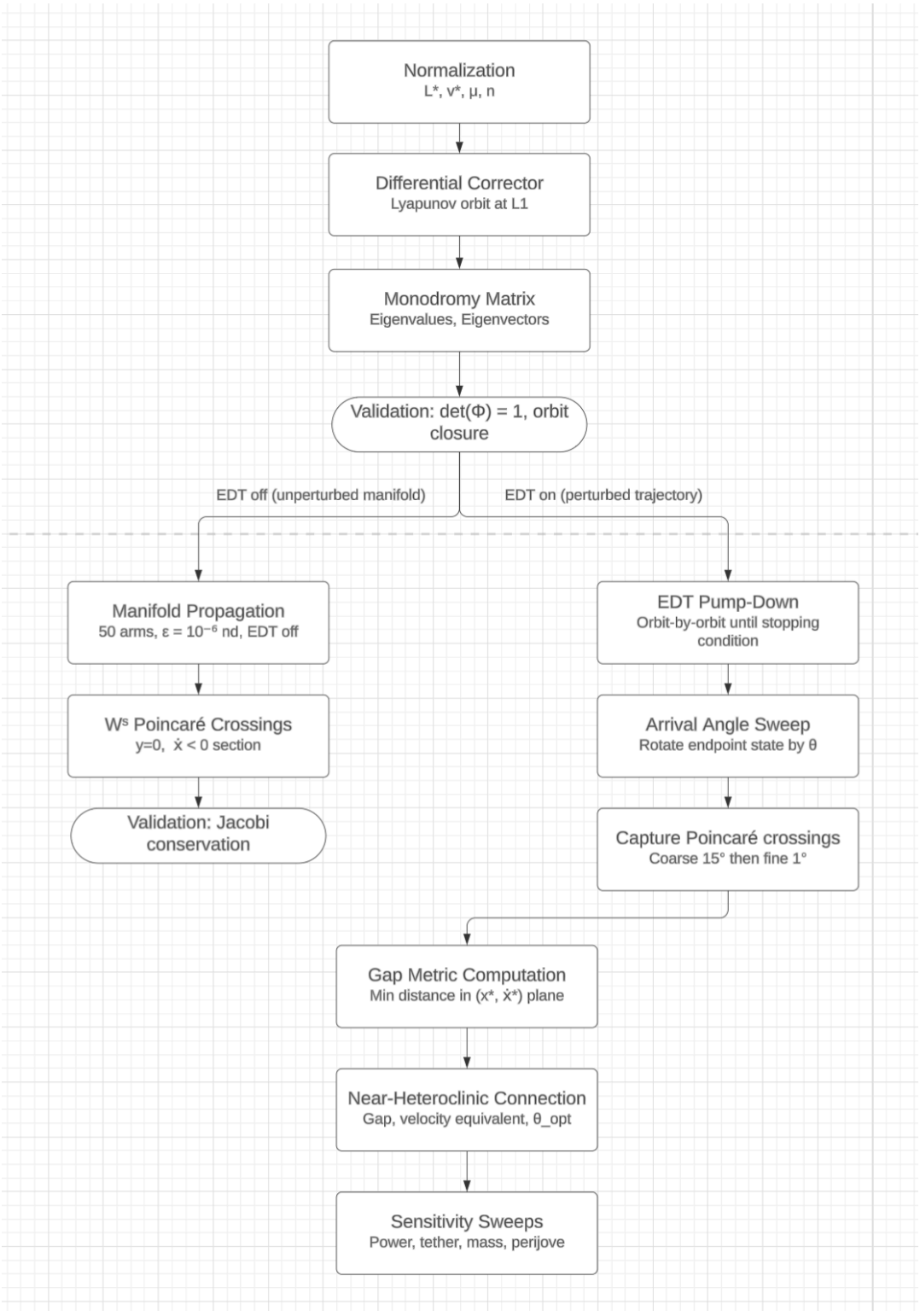


Figure 3.1 — Computational pipeline for the CR3BP-LT manifold targeting analysis, showing the sequential stages from normalization through sensitivity sweeps and the parallel manifold and pump-down branches converging at the Poincaré section gap metric.

3.2 State-Space Formulation

The second-order CR3BP-LT equations of motion derived in Section 2.4 are reduced to a first-order system for numerical integration. The state vector is defined as:

$$\mathbf{X} = [x, y, z, \dot{x}, \dot{y}, \dot{z}]^T \tag{3.1}$$

and its time derivative is:

$$\dot{\mathbf{X}} = f(\mathbf{X}) = [\dot{x}, \dot{y}, \dot{z}, \ddot{x}, \ddot{y}, \ddot{z}]^T \tag{3.2}$$

where the acceleration components are given by the CR3BP-LT equations of motion from Section 2.4.3. The EDT acceleration vector a_{EDT} is computed at each function evaluation from the instantaneous spacecraft position and velocity in the rotating frame, converted to dimensional quantities for the force calculation, then converted back to normalized units before being added to the gravitational acceleration components.

For manifold computation and the differential corrector, the six-element state vector is augmented with the state transition matrix Φ to form a single combined system. The STM encodes how a small perturbation to any component of the initial state propagates forward in time. Because it maps a six-element state perturbation to another six-element state perturbation, Φ is a 6×6 matrix containing 36 entries. Stacking these 36 entries beneath the 6 state components produces a 42-element vector that is integrated as one system:

$$\mathbf{Y} = [\mathbf{X}; \text{vec}(\Phi)] \tag{3.3}$$

where vec(Φ) denotes the column-stacked entries of Φ. The combined system propagates as:

$$\dot{\mathbf{Y}} = [f(\mathbf{X}); A(t)\,\Phi] \tag{3.4}$$

where A(t) is the Jacobian derived in Section 2.5.3. This formulation computes the trajectory and its sensitivity to initial conditions simultaneously in a single integration pass, avoiding the numerical overhead and accuracy loss of finite-difference STM approximations [30]. Consistent with the manifold-as-template assumption stated in Section 2.4.3, the variational equations are integrated under the unperturbed CR3BP Jacobian, excluding EDT forcing from A(t).

3.3 ODE Solver and Tolerance Selection

MATLAB's ode113 solver, a variable-order Adams-Bashforth-Moulton predictor-corrector method, is used for all trajectory integrations in this study. The CR3BP equations of motion are smooth with no discontinuities, and the integrations span many orbital

periods, conditions under which ode113's adaptive step sizing and variable-order accuracy outperform fixed-step Runge-Kutta methods in both efficiency and precision. Near L1, where the dynamical sensitivity is highest and small errors in the manifold arm propagations directly affect the gap metric, the variable-order capability allows the solver to concentrate accuracy where the dynamics demand it without imposing a globally fine step size everywhere else.

Two tolerance levels are used depending on the application. The EDT pump-down, manifold arm propagation, and Poincaré section collection use a relative tolerance RelTol of 10^{-10} and an absolute tolerance AbsTol of 10^{-12} nd. The differential corrector uses tighter tolerances of RelTol = 10^{-12} and AbsTol = 10^{-14}, because convergence to a true periodic orbit requires that the half-period closure error be resolved below 10^{-10} nd. This is a condition that the looser tolerances cannot reliably satisfy. The tolerance selection for the main integrations is driven by a specific requirement: the Jacobi constant drift during free CR3BP integration must be small enough that it cannot be mistaken for physical energy extraction by the EDT. At looser tolerances, numerical drift in the Jacobi constant would contaminate the pump-down signal, making the energy accounting unreliable. As quantified in Section 3.10, the drift at the selected tolerances is below 10^{-6} nd, four orders of magnitude below the 10^{-2} nd Jacobi changes the EDT produces per pump-down orbit, which confirms that the tolerance floor is set up correctly.

3.4 Differential Corrector Implementation

The L1 Lyapunov orbit is computed using a single-shooter differential corrector exploiting the x-axis symmetry of the planar CR3BP [15, 30]. The initial state is constrained to the form $[x_0, 0, 0, 0, \dot{y}_0, 0]^T$, placing the spacecraft on the x-axis with zero radial velocity. The periodicity condition requires that the trajectory return to the x-axis with $\dot{x}=0$ after exactly one half-period, which by symmetry guarantees full periodicity.

The linearized collinear equilibrium analysis at L1 provides the initial guess. The in-plane frequency $\omega_{Lyapunov}$ is derived from the eigenvalues of the linearized system, and the ratio $\kappa = \frac{\omega^2_{Lyapunov} - U_{xx}}{2\omega_{Lyapunov}}$ relates the transverse velocity to the displacement amplitude, giving $\dot{y}_0 = -\kappa A_x$ as the initial velocity estimate. The corrector then iterates on x_0 alone, holding $\dot{y}_0$ fixed, using the Newton update:

$$x_0^{\,i+1} = x_0^{\,i} - \frac{\dot{x}_f^{\,i}}{\Phi_{41}^{\,i}} \tag{3.5}$$

where $\dot{x}_f$ is the terminal radial velocity violation and Φ_{41} is the (4, 1) element of the STM at the half-period crossing, which quantifies how a change in x_0 propagates to $\dot{x}_f$ [30]. The corrector

terminates when $|\dot{x}_f|<10^{-10}$ nd, corresponding to a velocity closure error on the order of 10^{-7} m/s in physical units.

Amplitude continuation is used to reach the target orbit [15]. Starting from A_x=0.005 nd, the corrector is applied at each amplitude step up to A_x=0.010 nd, warm-starting each step from the previous converged half-period. The target amplitude of A_x=0.010 nd was selected as the largest amplitude at which the single-shooter corrector reliably converges at the specified tolerances; continuation to A_x=0.015 nd diverged after the maximum iteration count, indicating that the nonlinear orbit geometry at larger amplitudes exceeds the convergence basin of the single-shooter method. This avoids basin-of-attraction problems that arise when applying the corrector directly at larger amplitudes where the nonlinear orbit geometry departs significantly from the linearized initial guess.

3.5 Pump-Down Stopping Conditions

The pump-down simulation is terminated according to one of two stopping conditions, each representing a distinct point in the energy gap closure process. Both are defined relative to the framework established in Section 2.4.2, which requires $C < C_{L1}$ before the stable manifold becomes geometrically accessible. Both conditions are carried through the subsequent analysis to characterize how the pump-down strategy affects the Phase 2 steering requirement and the manifold connection geometry.

The first stopping condition, referred to throughout as the partial pump-down case, halts the simulation when the Jacobi constant falls below $C_{Lyapunov} + 0.05$ nd. This threshold closes approximately 83% of the total energy gap $\Delta C_{total} = C_{cap} - C_{Lyapunov}$ and leaves the L1 neck technically closed at pump-down termination, meaning Phase 2 EDT steering must both close the residual energy gap and thread the manifold corridor. The 0.05 nd margin above the Lyapunov orbit energy was chosen to represent an operationally conservative stopping point that reduces pump-down duration while still approaching the manifold target energy. The second stopping condition, referred to as the full pump-down case, continues the simulation until the Jacobi constant falls below C_{L1}. This ensures the L1 neck is topologically open at pump-down termination, reducing Phase 2 to a trim maneuver rather than a full manifold bridging burn. Continued pump-down beyond C_{L1} produces diminishing returns in Phase 2 steering reduction relative to the additional time cost, which is why C_{L1} rather than $C_{Lyapunov}$ serves as the stopping criterion.

Both cases are evaluated against the same Lyapunov orbit and manifold geometry, allowing direct comparison of the phase-space gap and arrival angle sensitivity across pump-down strategies.

3.6 EDT Force Direction Convention

The EDT Lorentz acceleration is applied antiparallel to the relative velocity vector between the spacecraft and the co-rotating Jovian plasma [7]. Physically, the tether current is driven by the motional EMF induced as the tether moves through the magnetic field relative to the plasma, and the resulting Lorentz force opposes that relative motion. At perijove, where the spacecraft orbital velocity substantially exceeds the local plasma co-rotation velocity, this force acts as a brake — directed predominantly opposite to the spacecraft velocity in the rotating frame, extracting orbital energy and reducing the Jacobi constant. The force magnitude is set by the power-limited current model from Section 2.4.3, with the direction computed at each function evaluation from the instantaneous relative velocity vector.

This convention means the EDT acceleration is not a fixed vector but varies continuously as the spacecraft moves through the rotating frame. Near perijove the braking force is strong and well-aligned with the deceleration direction; near apojove, the relative velocity between the spacecraft and the co-rotating plasma is small, the induced EMF is low, and the force contribution is negligible. The pump-down work is therefore concentrated at perijove passes, which is the physical mechanism underlying the orbit-by-orbit energy extraction described in Section 2.2.

3.7 Arrival Angle Parameterization

The arrival angle θ is defined as the angle between the Jupiter-Io line and the spacecraft perijove direction in the Jupiter-Io synodic rotating frame at the moment of Jupiter orbit insertion. It parameterizes the initial azimuthal orientation of the capture ellipse relative to the rotating frame and determines the phase alignment between the capture trajectory and the stable manifold tube at the Poincaré section [25]. A full 360° sweep over θ therefore explores all possible phase relationships between the post-JOI trajectory and the manifold targeting set without altering the orbit energy or the hardware configuration.

Computing the full EDT pump-down from scratch at each arrival angle would be computationally prohibitive and is unnecessary. The pump-down duration is determined by the total energy gap $\Delta C_{total} = C_{cap} - C_{Lyapunov}$, which depends on the perijove radius, hardware parameters, and magnetic field model but is insensitive to the initial azimuthal orientation of the capture ellipse in the rotating frame. This invariance holds because the EDT braking force is evaluated from the spacecraft's instantaneous position and velocity at each perijove pass, and the energy extracted per pass depends on the radial distance and orbital speed at perijove — neither of which changes when the ellipse is rotated azimuthally at fixed perijove radius. The pump-down endpoint state is therefore approximately the same regardless of θ.

The implementation exploits this invariance by running the full pump-down once at a reference orientation and then rotating the resulting endpoint state by the desired θ before propagating the Poincaré section crossing map. The rotation is applied as:

$$\mathbf{Y}_\theta = [R(\theta), 0; 0, R(\theta)] \tag{3.6}$$

where $R(\theta)$ is the 3×3 rotation matrix about the z-axis by angle θ, applied separately to the position and velocity components of the normalized state vector. This approach reduces the arrival angle sweep from a computation requiring hundreds of full pump-down integrations to one requiring a single pump-down followed by a set of forward propagations from the rotated endpoint states [25].

The sweep proceeds in two stages. A coarse sweep at 15° resolution over the full 360° identifies candidate bands where the capture trajectory produces crossings near the stable manifold cluster. A fine sweep at 1° resolution is then run within each promising band, defined as any band producing a gap below 0.15 nd in the coarse sweep. This two-stage approach balances computational cost against resolution, ensuring the optimal arrival angle is identified to within 1° without exhaustively evaluating every degree of the full circle.

3.8 Poincaré Section Crossing Detection

The Poincaré section is defined at $y=0$ with $\dot{x}<0$, implemented as a MATLAB event function passed to ode113. The event function is non-terminating, as the integration continues after each crossing, and direction-selective, recording only downward crossings where $\dot{x}<0$. This directionality isolates trajectory segments approaching Jupiter from the outer system, which are the geometrically relevant segments for manifold entry as discussed in Section 2.6. At each detected crossing, the integrator returns the interpolated state at the exact crossing time using its internal dense output capability, producing crossing locations accurate to the integrator tolerance rather than the nearest discrete time step.

The gap metric is computed as the minimum Euclidean distance in the $(x^*, \dot{x}^*)$ Poincaré plane between all capture trajectory crossings and all stable manifold crossings within the L1 neighborhood defined by $|x^* - x_{L1}| < 0.15$ nd [25]. This windowing step filters geometrically irrelevant crossings far from L1 and reduces the computational cost of the pairwise distance search. Since the gap is evaluated in the two-dimensional $(x^*, \dot{x}^*)$ projection of the four-dimensional cross-section of the Poincaré surface, it represents a lower bound on the true phase-space separation. The full six-dimensional phase space is reduced to four dimensions by the $y=0$ constraint and then further projected onto the $(x^*, \dot{x}^*)$ plane. The gap is nonetheless the operationally meaningful metric for EDT steering, since the steering maneuver acts primarily in the velocity direction and the position offset at the section is small.

3.9 Numerical Parameters Summary

Table 3.1 consolidates the key numerical parameters used throughout the implementation for reference when interpreting the results of Chapter 4.

Table 3.1 — Key numerical parameters used throughout computational implementation.

Parameter	**Value**	**Section**
ODE solver	ode113 (Adams-Bashforth-Moulton)	3.3
Main integration RelTol	10^{-10}	3.3
Main integration AbsTol	10^{-12} nd	3.3
Differential corrector RelTol	10^{-12}	3.3
Differential corrector AbsTol	10^{-14}	3.3
Corrector convergence tolerance	10^{-10} nd ($\approx 10^{-7}$ m/s)	3.4
Amplitude continuation target	$A_x = 0.010$ nd	3.4
Partial pump-down threshold	$C < C_{Lyapunov} + 0.05$ nd	3.5
Full pump-down threshold	$C < C_{L1}$	3.5
Manifold seed points N	50 per orbit	3.7
Manifold perturbation size ε	10^{-6} nd (≈ 422 m)	3.7
Manifold integration time T_{man}	4π nd (2 Io periods)	3.7
Coarse sweep resolution	15°	3.7
Fine sweep resolution	1°	3.7
Fine sweep threshold	0.15 nd	3.7
L1 neighborhood window	$\lvert x^* - x_{L1} \rvert < 0.15$ nd	3.8

3.10 Validation

Three numerical checks confirm that the implementation satisfies the conservation properties the CR3BP equations of motion require before any results are trusted.

The first check validates Jacobi constant conservation in the free CR3BP. With EDT forcing disabled, the Jacobi constant is tracked over ten Io orbital periods for a test orbit at approximately 14 R_J. The drift ΔC over this integration is below 10^{-6} nd (four orders of magnitude below the Jacobi changes produced by the EDT pump-down), confirming that the integrator tolerance is sufficient to resolve the approximately 10^{-2} nd Jacobi changes. Any drift of comparable magnitude to the EDT signal would make the pump-down energy accounting untrustworthy.

The second check validates the monodromy matrix determinant. As derived in Section 2.5.3, the Hamiltonian structure of the CR3BP requires $\det(\Phi)=1$ exactly, following from the zero-trace condition of the Jacobian A(t). The computed value of $\det(\Phi)=1.00000000$ to eight significant figures confirms that the variational equations are correctly implemented and that the 42-element combined integration is producing a volume-preserving STM consistent with the Hamiltonian structure of the CR3BP [15]. This check is particularly important because errors in the partial derivative expressions U_{xx}, U_{yy}, U_{xy} in the Jacobian block would produce a non-unit determinant — the check therefore validates the correctness of the entire variational equation implementation, not just the integrator accuracy.

The third check validates the Lyapunov orbit closure. The differential corrector converges to $|\dot{x}_f|<10^{-10}$ nd at the half-period crossing, corresponding to a physical velocity error below 10^{-7} m/s. The converged orbit, when propagated for one full period and compared to the initial condition, closes to within the integrator tolerance, confirming that the orbit is genuinely periodic and suitable as a manifold seed [15, 30]. Because the manifold tube geometry is computed by perturbing points along this orbit in the eigenvector directions, any periodicity error in the reference orbit would propagate into the manifold arm initial conditions and degrade the accuracy of the gap metric. The closure check confirms this error source is negligible.

Together these three checks establish that the implementation is consistent with the analytical framework of Chapter 2 before any results are presented in Chapter 4.

4. Results

4.1 Overview

This chapter presents the results of the CR3BP-LT manifold targeting analysis, proceeding from the post-JOI boundary state through Lyapunov orbit and manifold characterization, EDT pump-down simulation, Poincaré section gap analysis, and sensitivity and perijove sweeps. Together these results establish that propellantless manifold transit to Io is dynamically feasible within the CR3BP-LT framework.

4.2 Post-JOI Boundary State

The post-JOI boundary state that initializes all subsequent analysis was derived from the patched-conic baseline detailed in Appendix B. A minimum-energy Hohmann transfer from Earth to Jupiter yields a hyperbolic excess velocity of $v_\infty = 5.643$ km/s at Jupiter's sphere of influence, corresponding to a transfer time of 997.9 days (2.73 years) and requiring a JOI ΔV of 1.152 km/s to enter an 8 x 200 R_J capture ellipse at the nominal perijove of 8 R_J. The propellant comparison across propulsion types is presented in Appendix B; the key result for this chapter is that the capture orbit enters the Jupiter-Io rotating frame with $C_{cap} = 3.286$ nd. Since this exceeds the L1 neck-opening threshold $C_{L1} = 3.005$ nd, the L1 gateway is energetically closed at capture and EDT pump-down is required before manifold access is possible.

4.3 Lyapunov Orbit and Manifold Characterization

4.3.1 Lyapunov Orbit Computation

Before the EDT pump-down can be evaluated, the gravitational structure it must ultimately target needs to be established. The L1 Lyapunov orbit is the first structure to compute: it serves as the seed for the invariant manifold tubes that define the Phase 2 targeting set, and its Jacobi constant sets the energy level the EDT must reach. The differential corrector described in Section 3.4 was used to converge this orbit with the properties described in Table 4.1, with the L1 equilibrium point located numerically at $x_{L1} = 0.97514$ nd, consistent with the mass ratio $\mu = \frac{m_{Io}}{m_{Jupiter} + m_{Io}} = 4.70 \times 10^{-5}$.

Table 4.1 — Converged L1 Lyapunov reference orbit properties.

Parameter	**Value**
Initial condition x_0	0.97324144 nd
Initial velocity $\dot{y}_0$	0.01407892 nd/nd

Full period T	2.9984 nd (0.8446 days, ~20.3 hours)
Amplitude A_x	0.0023 nd (~977 km)
Jacobi constant $C_{Lyapunov}$	3.005316 nd

The orbit period of approximately 20 hours places it well within Io's orbital period of 1.769 days, consistent with the linearized frequency estimate $\omega_{Lyapunov}$=1.9525 nd derived from the equilibrium analysis. Figure 4.1 shows the converged orbit in the Jupiter-Io synodic frame, zoomed to the L1 neighborhood.

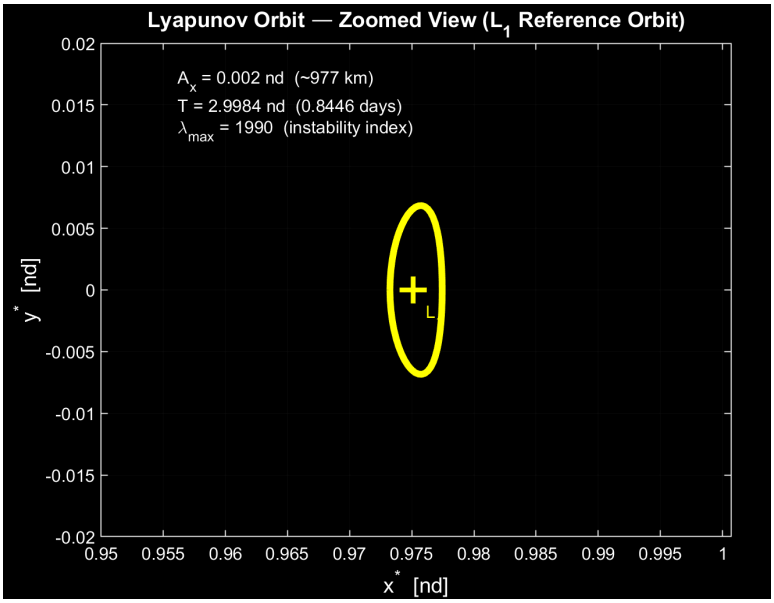


Figure 4.1 — Converged L1 Lyapunov reference orbit in the Jupiter-Io synodic rotating frame.

The Jacobi constant of the converged orbit, $C_{Lyapunov}$=3.005316 nd, lies just below the L1 neck-opening threshold C_{L1}=3.005 nd, confirming that the orbit exists in the energetically accessible region immediately inside the L1 gateway. The total energy gap the EDT must close is therefore $\Delta C_{total}=C_{cap}-C_{Lyapunov}=3.286-3.005=0.281$ nd.

4.3.2 Monodromy Matrix and Stability Analysis

The monodromy matrix Φ was computed by integrating the variational equations simultaneously with the Lyapunov orbit state over one full period T = 2.9984 nd, as described in Section 3.2. Its eigenvalues, known as Floquet multipliers, quantify how a perturbation evolves over one complete revolution and are summarized in Table 4.2. Both the raw eigenvalue λ and its magnitude |λ| are reported: the signs of λ carries information about the direction of rotation in phase space, while |λ| is the stability criterion [15]. Values above 1 indicate exponential growth along the corresponding eigenvector, values below 1 indicate contraction, and values equal to 1 indicate neutral behavior.

Table 4.2 — Monodromy matrix eigenvalues and physical interpretation.

Eigenvalue Pair	**λ**	**|λ|**	**Physical Interpretation**
Hyperbolic (unstable)	$+1.9896 \times 10^{3}$	1989.58	Unstable manifold direction W^{u} ● Perturbations grow by factor ~1990 per orbit
Hyperbolic (stable)	$+5.026 \times 10^{-4}$	0.000503	Stable manifold direction Ws ● Perturbations contract by factor 1/1990 per orbit
Trivial	+1.000025	1.000025	Along-orbit direction ● Neighboring periodic trajectory
Trivial	+0.999975	0.999975	Energy surface direction ● Iso-Jacobi perturbation
Center	0.9868 + 0.1622i	1.000000	Out-of-plane neutral mode ● No growth or decay
Center	0.9868 − 0.1622i	1.000000	Out-of-plane neutral mode (complex conjugate)
Product	det(Φ) = 1.00000000		Confirms Hamiltonian (symplectic) structure ✓

The product of all six eigenvalues equals det(Φ) = 1.00000000, confirming that the flow is Hamiltonian and volume-preserving as required by the CR3BP structure and verified by the implementation check in Section 3.10 [15]. The hyperbolic pair λ_1=1989.58 and $\lambda_2=1/\lambda_1=5.026\times10^{-4}$ satisfies the symplectic reciprocal condition $\lambda_1\lambda_2 = 1$ to numerical precision. This numerical pair is also referred to throughout as the instability index, which emphasizes the physical interpretation of λ_{max} as the per-orbit growth factor of a perturbation along the unstable eigenvector direction rather than as an abstract eigenvalue.

A perturbation of 1 meter along the unstable direction grows to approximately 1,990 meters after a single revolution. This extreme instability, nearly three orders of magnitude larger than typical Earth-Moon L1 Lyapunov orbits, is a consequence of Jupiter's strong gravitational gradient, and it has a direct physical implication for the manifold connection: small phase-space separations near the manifold boundary translate to dramatically different long-term behavior, which underlies the sub-degree angular sensitivity observed in the arrival angle sweep of Section 4.5.

4.3.3 Stable and Unstable Manifold Directions

The stable and unstable eigenvectors of Φ define the tangent directions to the invariant manifold tubes at each point on the Lyapunov orbit. At the initial condition, the dominant directions are:

$$\mathbf{W}^s = \begin{bmatrix} -3.461 \times 10^{-1} \\ -1.364 \times 10^{-1} \\ 0 \\ +8.360 \times 10^{-1} \\ +4.033 \times 10^{-1} \\ 0 \end{bmatrix} \tag{4.1}$$

$$\mathbf{W}^u = \begin{bmatrix} +3.461 \times 10^{-1} \\ -1.364 \times 10^{-1} \\ 0 \\ +8.360 \times 10^{-1} \\ -4.033 \times 10^{-1} \\ 0 \end{bmatrix} \tag{4.2}$$

The velocity components dominate the eigenvector magnitude, reflecting the fact that the manifold is primarily a velocity-space structure near L1. This directly explains the narrow cross-sectional area the stable manifold tube presents at the Poincaré section, and the resulting sensitivity of the gap metric to the arrival angle θ, discussed in Section 5.1.

4.3.4 Manifold Tube Propagation

The stable and unstable manifold tubes were propagated using the procedure described in Section 3.7. Perturbing 50 equally-spaced points along the converged Lyapunov orbit by $\varepsilon = 10^{-6}$ nd (approximately 422 meters) along the respective eigenvectors and integrating for $T_{man}=4\pi$ nd generated 100 stable manifold arms (W^s) and 100 unstable manifold arms (W^u), forming the tube structures shown in Figure 4.2. EDT forcing was disabled during manifold propagation, consistent with the manifold-as-template assumption of Section 2.4.3 — the manifold is computed from the unperturbed gravitational dynamics and used as a fixed target, valid when EDT forcing is small relative to gravity.

The stable manifold W^s, shown in cyan in Figure 4.2, represents the set of all states that naturally approach the Lyapunov orbit asymptotically forward in time. The unstable manifold W^u (red) represents the departure directions; trajectories on W^u depart from the Lyapunov orbit and flow toward either Jupiter or Io depending on the sign of the perturbation. The zoomed view of

the L1 neck region in Figure 4.3 confirms that the tube arms converge back to the Lyapunov orbit as required by the definition of the stable manifold, and the hourglass cross-section at the neck is consistent with the theoretical geometry described by Koon et al. [15].

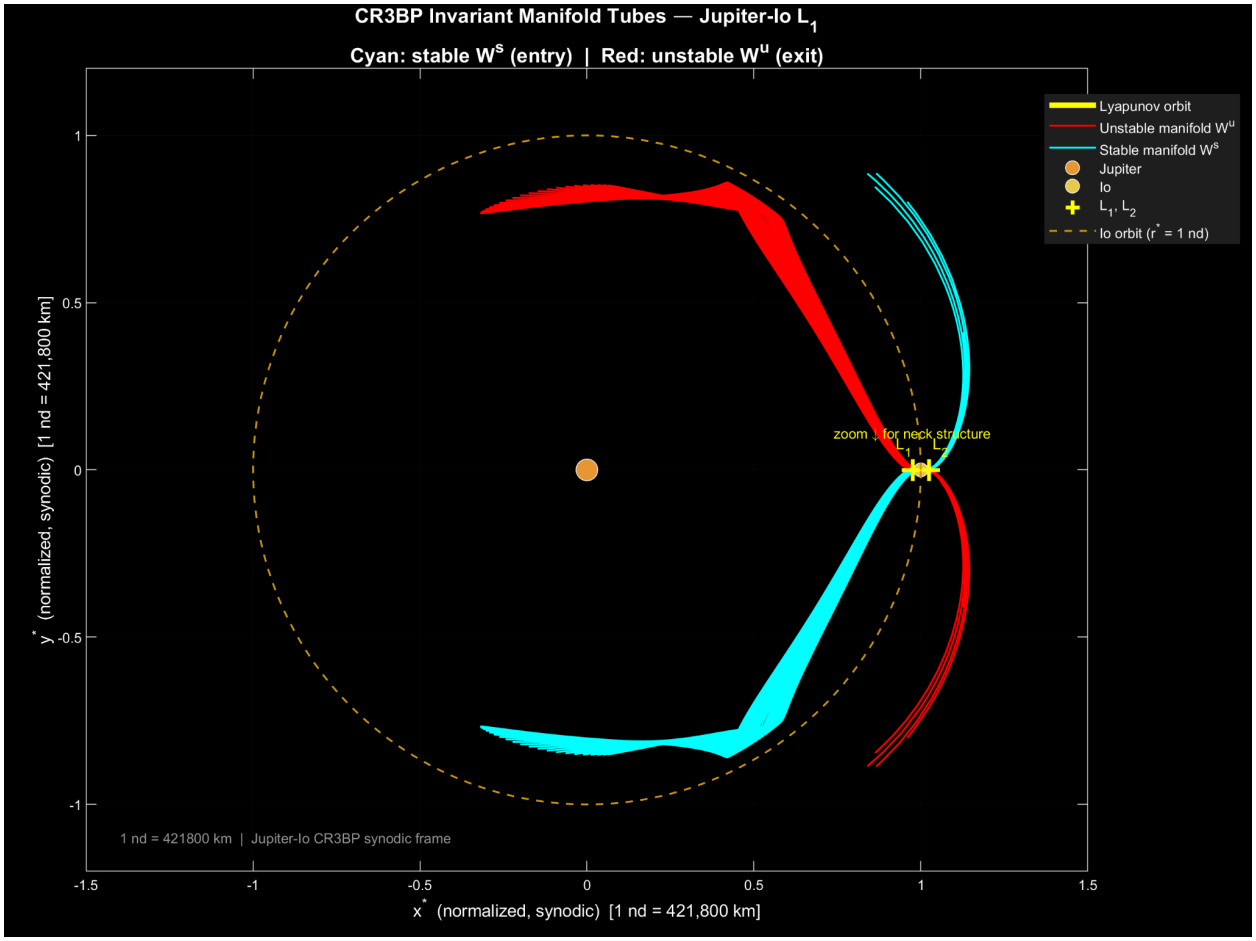


Figure 4.2 — Invariant manifold tubes of the L1 Lyapunov reference orbit in the Jupiter-Io synodic rotating frame.

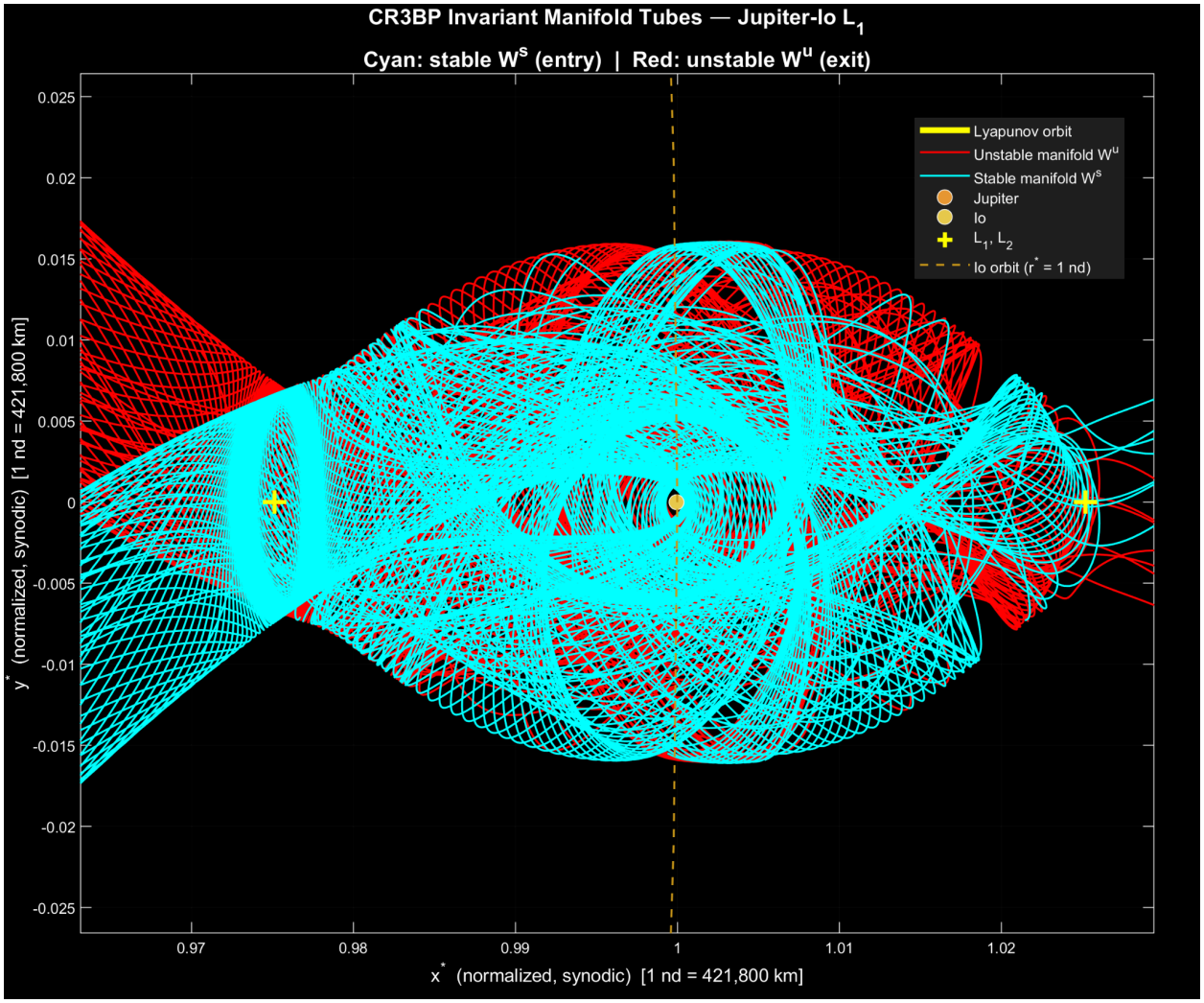


Figure 4.3 — Zoomed view of the L1 neck region showing the intersection of the stable (cyan) and unstable (red) manifold tubes.

4.3.5 Poincaré Section of Stable Manifold Crossings

The Poincaré section is the tool used to compare the spacecraft's phase-space location to the manifold targeting set: every time a trajectory crosses the Jupiter-Io connecting line moving inward, two quantities are recorded: the position x* and radial velocity $\dot{x}^*$. This produces a two-dimensional snapshot of an otherwise six-dimensional proximity problem. To characterize the stable manifold geometry in a form directly useful for trajectory targeting, the W^s arms were propagated until each crossed the surface of section $y = 0$ with $\dot{x} < 0$, as described in Section 3.8. A total of 327 W^s crossings were collected, of which 307 fell within the L1 neighborhood $|x^* - x_{L1}| < 0.15$ nd.

The resulting Poincaré map of the stable manifold is shown in Figure 4.4. The cyan cluster near x* = 0.975 nd, ẋ* ≈ -0.05 to -0.10 nd/nd represents the targeting set for Phase 2 manifold steering — the set of all phase-space states from which the spacecraft can reach the Lyapunov orbit propellantlessly. The ordered, narrow structure of the cluster reflects the tube geometry of the manifold: the W^s tube has a well-defined cross-section at this Poincaré section, and only states within that cross-section belong to the stable manifold. The magenta points represent the post-pump-down capture trajectory crossings at the default arrival geometry; their separation from the cyan cluster motivates the arrival angle sweep presented in Section 4.5.

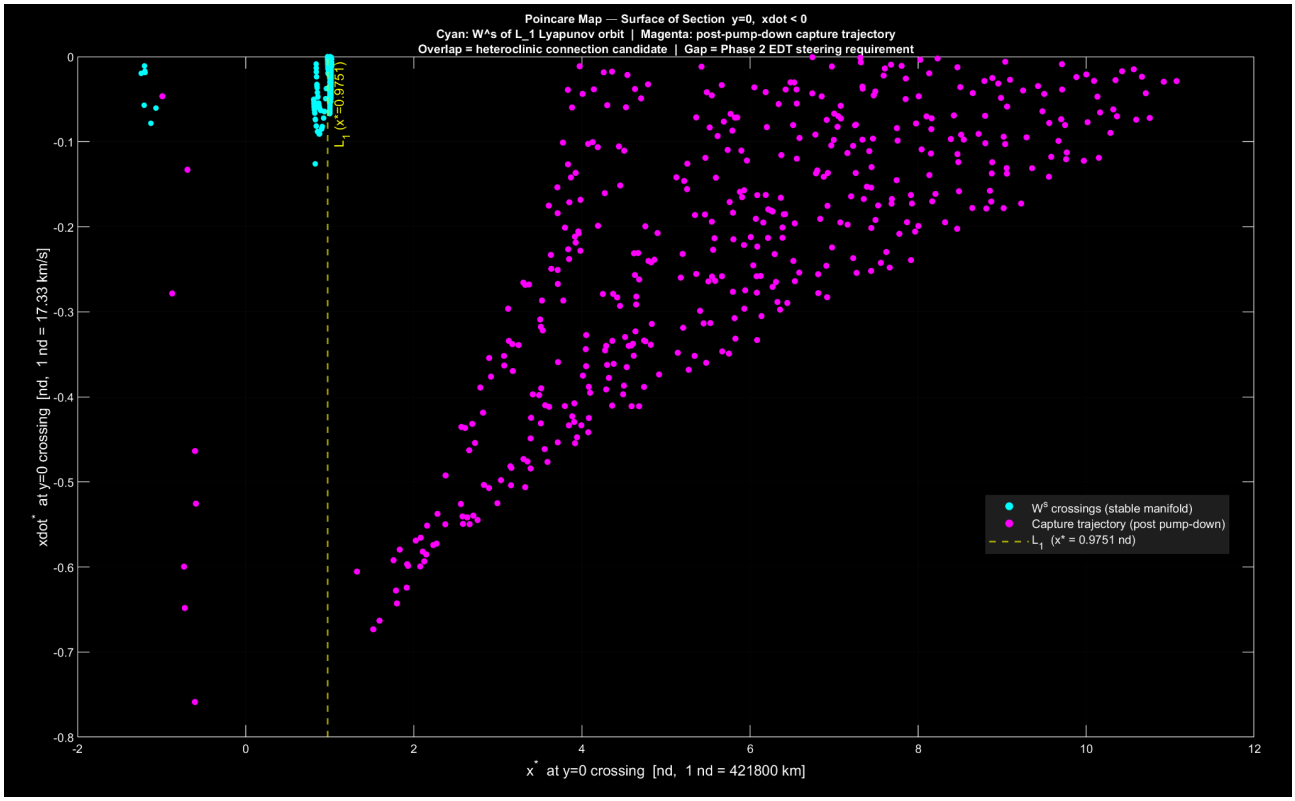


Figure 4.4 — Resulting Poincaré map of the stable manifold at y = 0 with ẋ < 0.

4.4 EDT Pump-Down — Phase 1

Two pump-down strategies are evaluated and carried through all subsequent analysis. The full pump-down case continues until the Jacobi constant drops below C_{L1} = 3.005 nd, ensuring the L1 neck is topologically open at termination (3,013 days, 23 orbits). The partial pump-down case halts earlier, at $C < C_{Lyapunov} + 0.05$ nd, closing approximately 83.4% of the total energy gap while reducing mission duration to 2,751 days across 21 orbits, at the cost of a larger Phase 2 steering requirement. Table 4.3 compares the two cases; the full pump-down is adopted as the primary result for all subsequent analysis.

4.4.1 Pump-Down Simulation Results

With the manifold targeting set established, the EDT now does the actual work of Phase 1: extracting orbital energy at each perijove pass, orbit by orbit, until the L1 gateway opens and the stable manifold becomes geometrically accessible. The post-JOI state enters the Jupiter-Io rotating frame with C_{cap}=3.286 nd. Since $C_{cap} > C_{L1}$ = 3.005 nd, EDT pump-down is required before manifold access is possible, as established in Section 2.4.2.

Two pump-down cases were evaluated using the stopping conditions defined in Section 3.5. In the partial pump-down case, the simulation halted when C dropped below $C_{Lyapunov}$ + 0.05 nd = 3.055 nd, yielding a pump-down duration of 2,751 days across 21 orbits with a Jacobi reduction of ΔC = 0.234 nd. This case closes 83.4% of the total energy gap (ΔC_{total} = 0.281 nd), leaving a residual of $\Delta C_{residual}$ = 0.047 nd above C_{L1}, meaning the neck remains technically closed at pump-down termination and Phase 2 must both close the residual gap and thread the manifold corridor. In the full pump-down case, the simulation continued until C dropped below C_{L1} = 3.005 nd, yielding a pump-down duration of 3,013 days across 23 orbits with ΔC = 0.285 nd: 101.5% of the total gap, with the neck fully open at pump-down termination. The residual Phase 2 steering requirement in the full case is reduced to $\Delta C_{residual} = -0.004$ nd, corresponding to a trim ΔV of approximately 37 m/s delivered by EDT over 6.8 days. Table 4.3 summarizes both cases.

Table 4.3 — Pump-down stopping condition comparison.

Parameter	**Partial pump-down**	**Full pump-down**
Stopping condition	$C < C_{Lyapunov}$ + 0.05 nd	$C < C_{L1}$
Duration	2,751 days	3,013 days
Orbits	21	23
ΔC Phase 1	0.234 nd	0.285 nd
Gap closed	83.4%	101.5%
C at end	3.052 nd	3.001 nd
Neck status at end	Closed	Open
Phase 2 ΔV	~415 m/s	~37 m/s

The full pump-down case is adopted as the primary result for all subsequent analysis. The 262-day extension beyond the partial case reduces the Phase 2 steering requirement by an order of magnitude (from a 415 m/s manifold bridging maneuver to a 37 m/s trim burn) at the cost of two additional capture orbits. Since EDT delivers both the additional pump-down and the Phase

2 trim propellantlessly, the primary cost is time rather than propellant. The Jacobi energy corridor for the full pump-down case is shown in Figure 4.5.

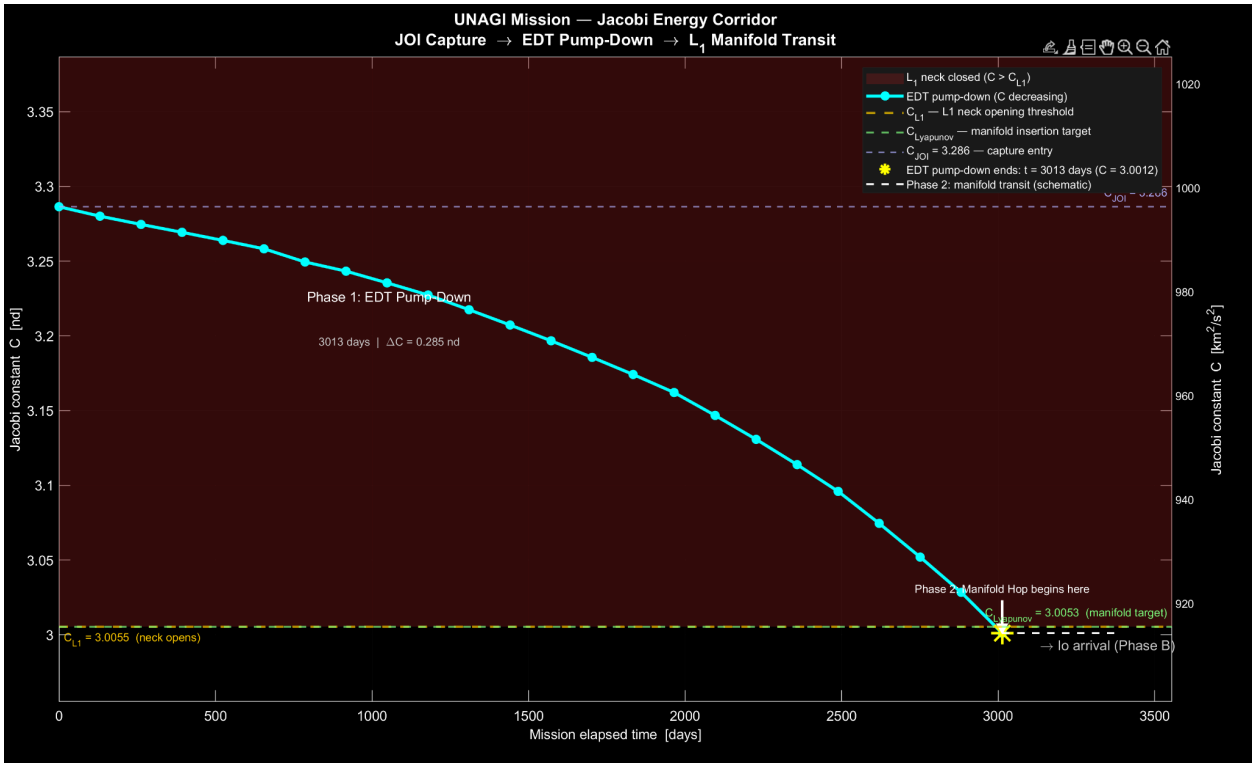


Figure 4.5 — Jacobi constant evolution during Phase 1 EDT pump-down, showing the monotonic descent from C_{cap} = 3.286 nd at JOI toward the C_{L1} = 3.005 nd neck-opening threshold over 3,013 days and 23 capture orbits.

4.4.2 EDT Performance at the Burn Point

The Phase 2 EDT steering maneuver executes at the L1 crossing, where $x^* \approx 0.975$ nd (approximately 411,328 km from Jupiter, near 8 R_J altitude). At this location the dipole magnetic field strength is B=1,941 nT. The power-limited current model gives a tether current of I=3.26 A at the baseline 1,070 W power budget and 10 km tether length, producing a Lorentz force of approximately 6.33×10^{-5} N/kg specific force. To deliver the full pump-down Phase 2 trim ΔV of 37 m/s, the EDT operates for 6.8 days at this force level with zero propellant expenditure. The partial pump-down Phase 2 burn of 415 m/s requires 75.85 days of continuous EDT operation.

4.5 Phase 2 Manifold Steering

4.5.1 Poincaré Map and Gap Quantification

Following Phase 1 EDT pump-down, the post-pump-down state was propagated forward and its crossings of the Poincaré section y = 0 with $\dot{x} < 0$ were recorded alongside the stable manifold W^s crossings established in Section 4.3.5. The gap metric is the minimum Euclidean

distance in the (x*, ẋ*) Poincaré plane between any capture crossing and the nearest W^s crossing within the L1 neighborhood, as defined in Section 3.8. Since the gap is predominantly in the ẋ* direction — the capture trajectory and the manifold cluster overlap significantly in x* but are offset in ẋ* — it translates approximately to a velocity requirement via $\Delta v \approx \Delta \cdot V^*$, where V^* = 17,334 m/s = 17.334 km/s is the canonical velocity unit.

The gap is quantified as the minimum Euclidean distance in the (x*, ẋ*) Poincaré plane between any capture crossing and the nearest W^s crossing:

$$\Delta = min_{i,j} \sqrt{[(x_c ap, i - x_W s, j)^2 + (\dot{x}_c ap, i - \dot{x}_W s, j)^2]} \quad (4.3)$$

4.5.2 Arrival Angle Sweep and the Near-Heteroclinic Connection

The gap depends critically on the arrival geometry at JOI, parameterized by the angle θ as defined in Section 3.7. To be precise, θ is the orientation of the capture ellipse at the moment of Jupiter orbit insertion: the angle between the Jupiter-Io line in the rotating frame and the spacecraft's perijove direction at JOI. It is not an angle into the L1 neck; the neck has no preferred entry angle once it is topologically open. Rather, θ determines the phase alignment between the post-pump-down capture trajectory and the stable manifold tube at the Poincaré section, accumulated over all pump-down orbits.

A full 360° coarse sweep at 15° resolution was conducted, followed by a 1° fine sweep centered on the most promising band. The coarse sweep results are shown in Figure 4.6. Of the 24 angles surveyed, only the band near 150-165° produced gaps below 0.70 nd, reflecting the narrow cross-section the W^s tube presents at the Poincaré section.

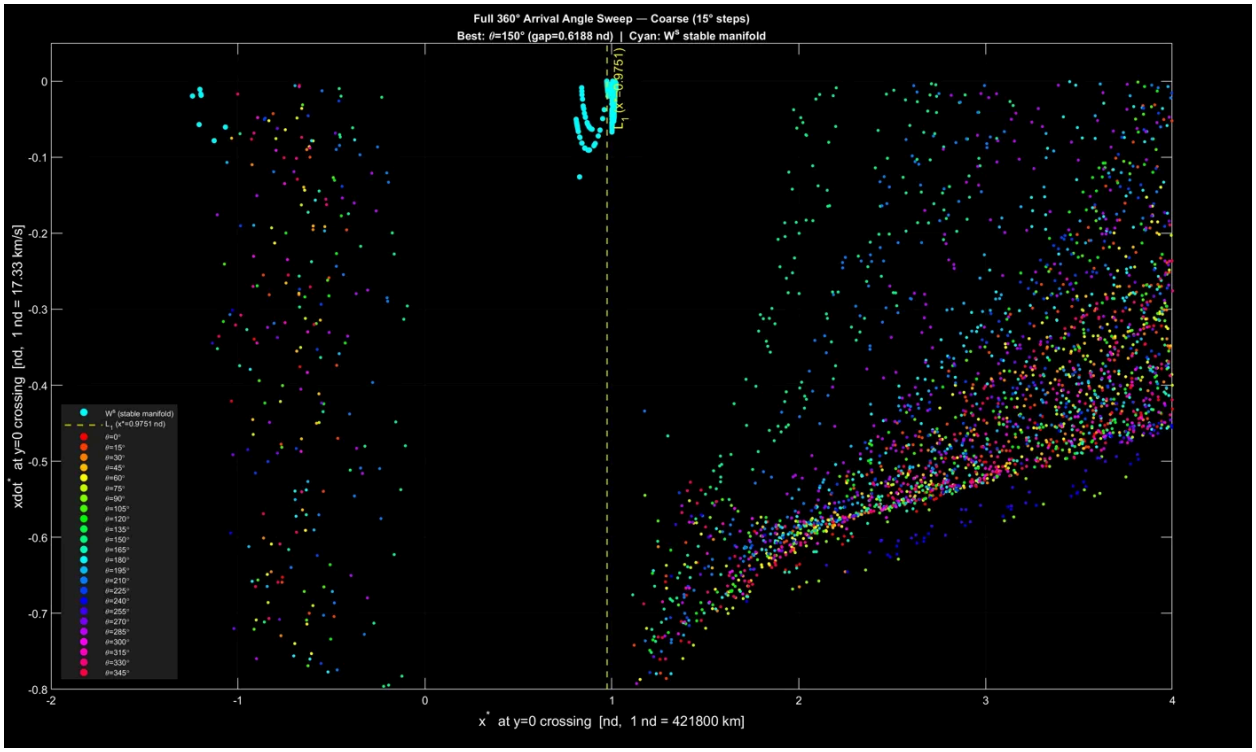


Figure 4.6 — Full 360° coarse arrival angle sweep at 15° resolution for full pump-down case. Each color represents a distinct arrival angle θ; the cyan cluster marks the W^s stable manifold targeting set near L1. The sparse connectivity pattern, with only the θ = 150° band (teal) producing crossings near the manifold cluster, reflects the narrow cross-section of the W^s tube at this Poincaré section.

For the partial pump-down case (C_{end}=3.052 nd, 2,751 days), the fine sweep resolved the optimal arrival angle to θ=167°, with a Poincaré gap of 0.0079 nd (approximately 137 m/s) and 26 W^s crossing opportunities. The gap at neighboring angles rises sharply: at θ=167.5° the gap becomes effectively infinite, with no capture crossings near the manifold cluster. The near-heteroclinic connection exists within a window of less than one degree of arrival angle. The dedicated Poincaré map at θ=167° is shown in Figure 4.7.

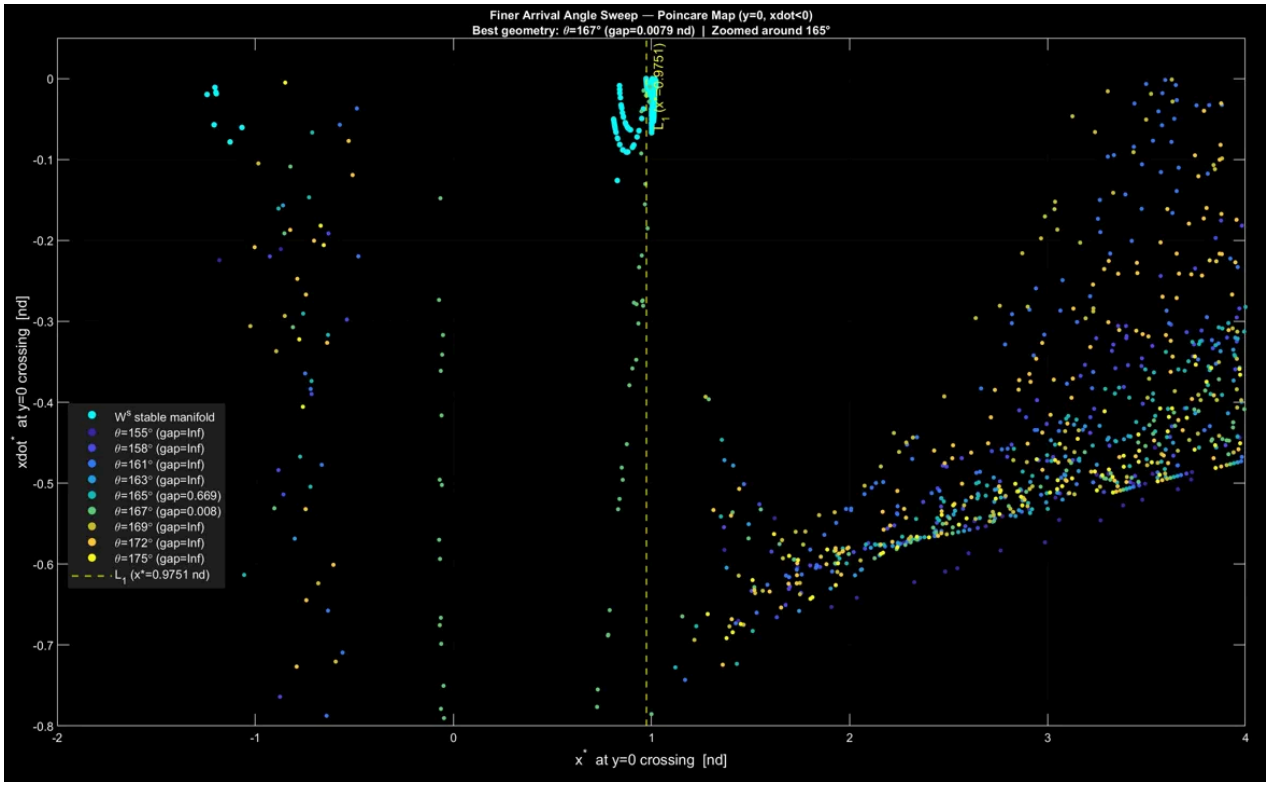


Figure 4.7 — Dedicated Poincaré map at the optimal arrival angle θ = 167° for the partial pump-down case, with an identified minimum phase-space gap of 0.0079 nd, which defines the Phase 2 EDT steering requirement.

For the full pump-down case (C_{end}=3.001 nd, 3,013 days), the fine sweep resolved the optimal arrival angle to θ=151°, with a Poincaré gap of 0.0179 nd (approximately 311 m/s) and 10 W^s crossing opportunities. Every neighboring angle in the fine sweep produced gaps above 0.6 nd or no L1-region crossings at all. The dedicated Poincaré map at θ=151° is shown in Figure 4.9. Table 4.4 summarizes the near-heteroclinic connection results for both cases.

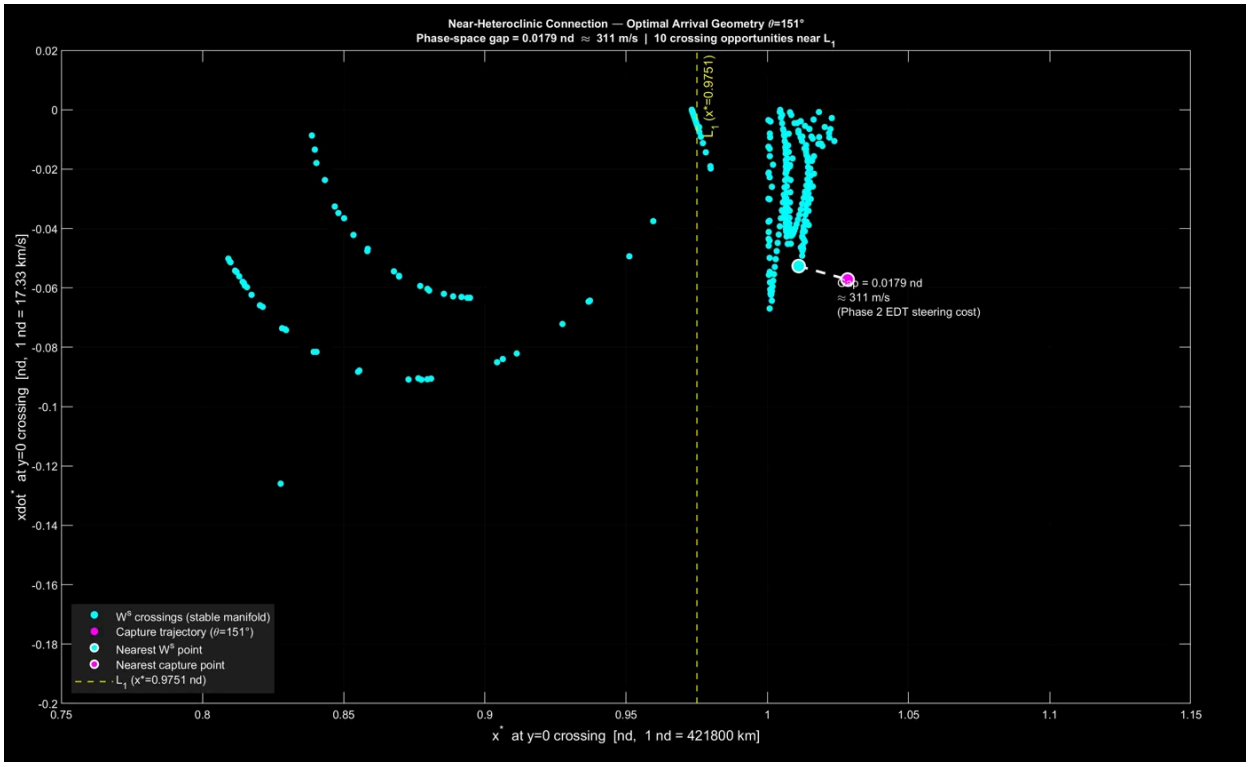


Figure 4.8 — Dedicated Poincaré map at the optimal arrival angle θ = 151° for the full pump-down case (C_{end} = 3.001 nd, 3,013 days). The white dashed arrow annotates the minimum phase-space gap.

Table 4.4 — Near-heteroclinic connection results: partial vs. full pump-down.

Parameter	**Partial pump-down**	**Full pump-down**
Best arrival angle θ	167°	151°
Poincaré gap	0.0079 nd	0.0179 nd
Velocity equivalent	~137 m/s	~311 m/s
Crossing opportunities	26	10
Phase 2 ΔV (EDT)	~415 m/s	~37 m/s
Propellant cost	0 kg	0 kg

Neither case constitutes a true heteroclinic connection, which would require a gap of exactly zero, achievable only at a specific launch date or with an additional optimization pass beyond pre-Phase A scope. The finite phase-space separation that remains is small enough that EDT steering bridges it propellantlessly in both cases. The shift in best angle from 167° to 151° between the partial and full pump-down cases reflects the change in post-pump-down phase

alignment caused by the additional 262 days of integration, rather than a fundamental difference in manifold geometry. The sub-degree angular sensitivity and the ordered crossing structure at both optimal angles, in contrast to the chaotic scatter at neighboring angles, are discussed in Section 5.2.

4.5.3 Complete Transfer Summary

With the near-heteroclinic connections identified and the Phase 2 steering requirements quantified, the full transfer sequence from JOI capture through L1 manifold entry can be summarized. Table 4.5 collects the key parameters for both pump-down cases, closing the loop from the post-JOI boundary state established in Section 4.2 through the manifold entry demonstrated in Section 4.5.2.

Table 4.5 — Complete transfer summary: JOI capture through L1 manifold entry.

Parameter	**Partial pump-down**	**Full pump-down**
Phase 0 JOI burn	1.152 km/s (chemical)	1.152 km/s (chemical)
Phase 1 duration	2,751 days / 21 orbits	3,013 days / 23 orbits
Phase 1 ΔV	0 km/s (propellantless)	0 km/s (propellantless)
Best arrival angle	θ=167°	θ=151°
Phase 2 ΔV (EDT)	0.415 km/s	0.037 km/s
Phase 2 duration	75.85 days	6.8 days
Phase 2 propellant	0 kg	0 kg
Total post-JOI ΔV	0.415 km/s (EDT)	0.037 km/s (EDT)
Io relative velocity	10,492 m/s	10,492 m/s

The relative velocity of 10,492 m/s reflects the spacecraft's state at the nearest Poincaré section crossing to Io's position following manifold transit, computed directly from the post-pump-down trajectory. These values characterize how close the trajectory naturally approaches Io's orbital state and are not an optimized Io orbit insertion cost — the insertion maneuver itself is a Phase B deliverable requiring full ephemeris reconstruction and is outside the scope of this study.

4.6 Sensitivity Analysis

4.6.1 Sweep Design

The baseline configuration — 1,070 W power budget, 10 km tether length, 1,000 kg spacecraft mass, nominal dipole B-field scale factor of 1.0 — was evaluated against a

four-dimensional parameter sweep to characterize the EDT pump-down feasibility envelope. The sweep covered 9 power levels (500, 750, 1070, 1500, 2000, 3500, 5000, 7500, 10000 W) × 8 tether lengths (5, 9, 10, 15, 18, 20, 25, 28 km) × 4 spacecraft masses (800, 1000, 1200, 1500 kg) × 3 B-field scale factors (0.8, 1.0, 1.2), producing 864 parameter combinations. Each combination was evaluated for pump-down duration to neck opening ($C < C_{L1}$), with feasibility defined as completion within 10 years (3,652 days). Combinations exceeding 30 years were flagged as infeasible. The baseline case produces a pump-down duration of 3,013 days (8.2 years) and sits within the 10-year feasibility window.

4.6.2 Power and Tether Length

Power and tether length are the two primary design levers for EDT pump-down duration, and their interaction is nonlinear. Figure 4.9 shows pump-down duration as a function of power budget for eight tether lengths at nominal mass and B-field. Tether length dominates at low power levels: increasing from 5 km to 28 km at 500 W reduces pump-down from infeasible to approximately 1,441 days for the lightest mass configuration at nominal B-field. At high power levels the two parameters interact more symmetrically and the feasibility envelope expands significantly. The mean ΔC per capture orbit, shown in Figure 4.10, confirms the mechanism: longer tethers and higher power both increase energy extraction per perijove pass, but the relationship saturates at short tether lengths regardless of available power.

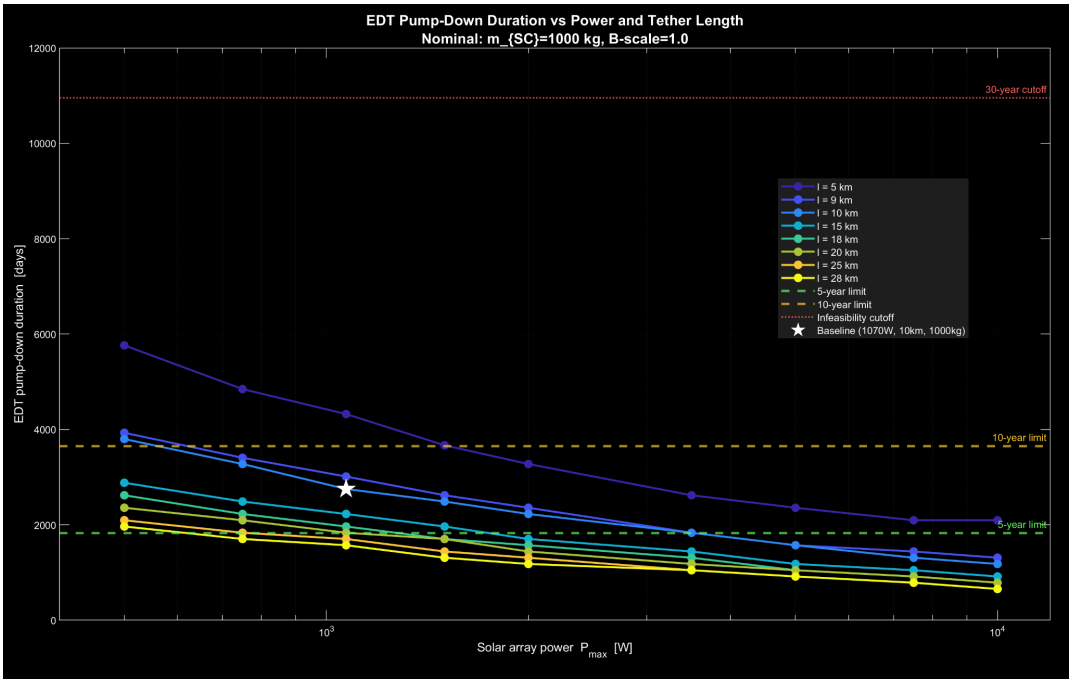


Figure 4.9 — EDT pump-down duration as a function of power budget for eight tether lengths, at nominal spacecraft mass (1,000 kg) and B-field scale factor (1.0). The white star marks the baseline configuration (1,070 W, 10 km, 3,013 days). Horizontal dashed lines indicate the 5-year and 10-year mission feasibility boundaries.

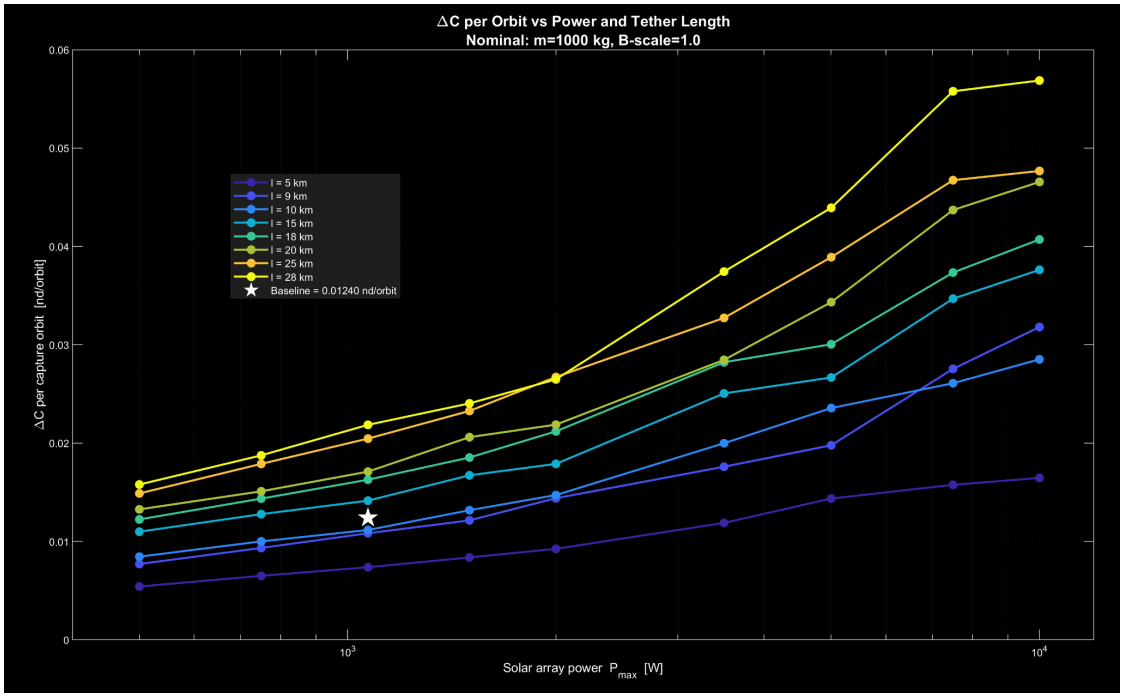


Figure 4.10 — Mean ΔC per capture orbit as a function of power budget for eight tether lengths, at nominal spacecraft mass (1,000 kg) and B-field scale factor (1.0).

The MEDT series configuration warrants specific attention. Two tethers in series produce an effective length of 18.4 km, which at 1,070 W and 1,000 kg mass yields a pump-down duration of approximately 1,900 days, which is well within the five-year mission window at nominal B-field. This represents a meaningful improvement over the single 10 km baseline of 3,013 days while remaining within the single-tether survivability envelope established by Casalino et al. [12] and Sanmartin et al. [7].

4.6.3 Magnetic Field Uncertainty

The B-field scale factor of ±20% brackets the uncertainty in the Jovian dipole field model at the perijove altitude of 8 R_J. Figure 4.11 shows the effect across the full power range at baseline tether length and mass. At the baseline configuration, B = 0.8 extends pump-down to 3,275 days and B = 1.2 reduces it to 2,489 days, bracketing the nominal 3,013-day result by approximately ±262 days (±8.7%). The mission remains feasible within 10 years across the entire B-field uncertainty envelope at the baseline power and tether configuration. At marginal configurations near the 10-year boundary, B-field uncertainty can push borderline cases across the feasibility threshold in either direction.

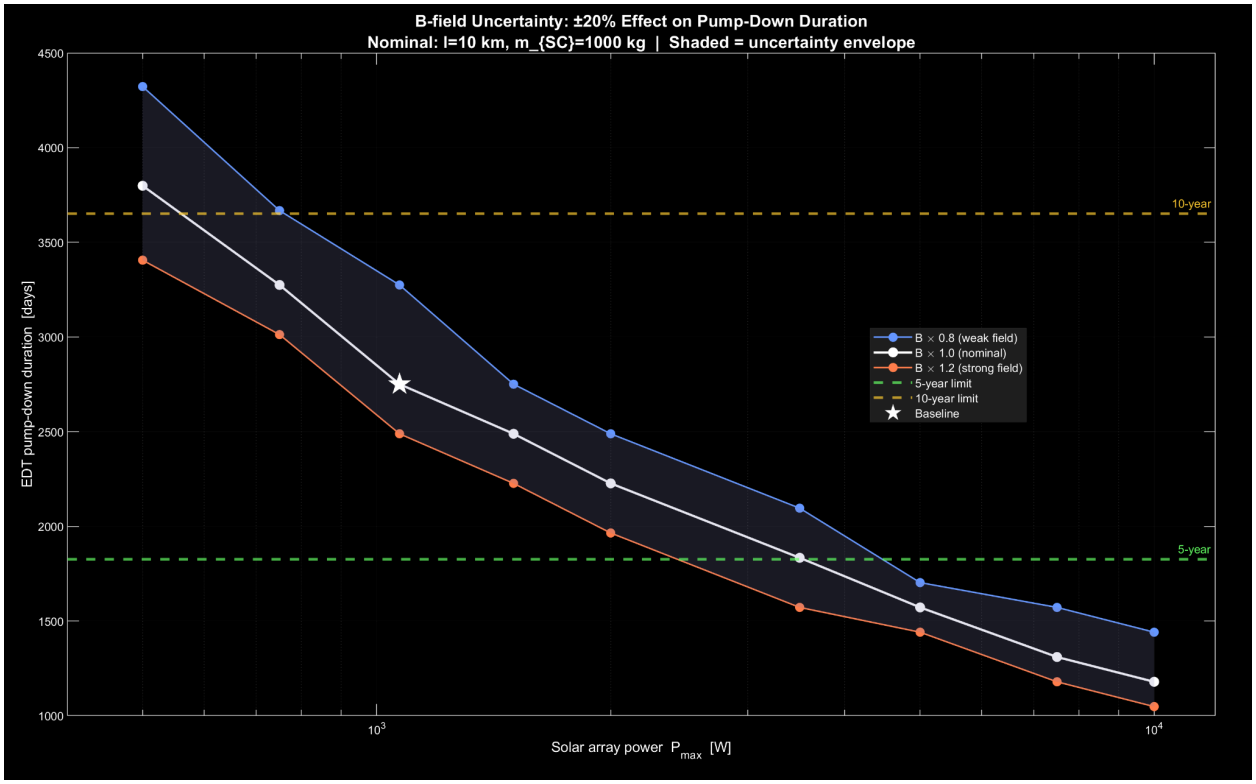


Figure 4.11 — EDT pump-down duration as a function of power budget for three B-field scale factors (0.8, 1.0, 1.2), at nominal spacecraft mass (1,000 kg) and tether length (10 km). The shaded band represents the ±20% magnetic field uncertainty envelope.

4.6.4 Spacecraft Mass

The same Lorentz force produces less acceleration on a heavier spacecraft, requiring more orbits to accumulate the same Jacobi constant reduction. Figure 4.12 shows the mass sensitivity across the full power range at baseline tether length and nominal B-field. Increasing mass from 800 kg to 1,500 kg at baseline power and tether length pushes pump-down from approximately 2,096 days to infeasible within 10 years at weak field conditions. The 1,000 kg baseline is therefore not conservative at the margins, and heavier configurations require either longer effective tether length, higher power, or favorable B-field conditions to remain within the feasibility window.

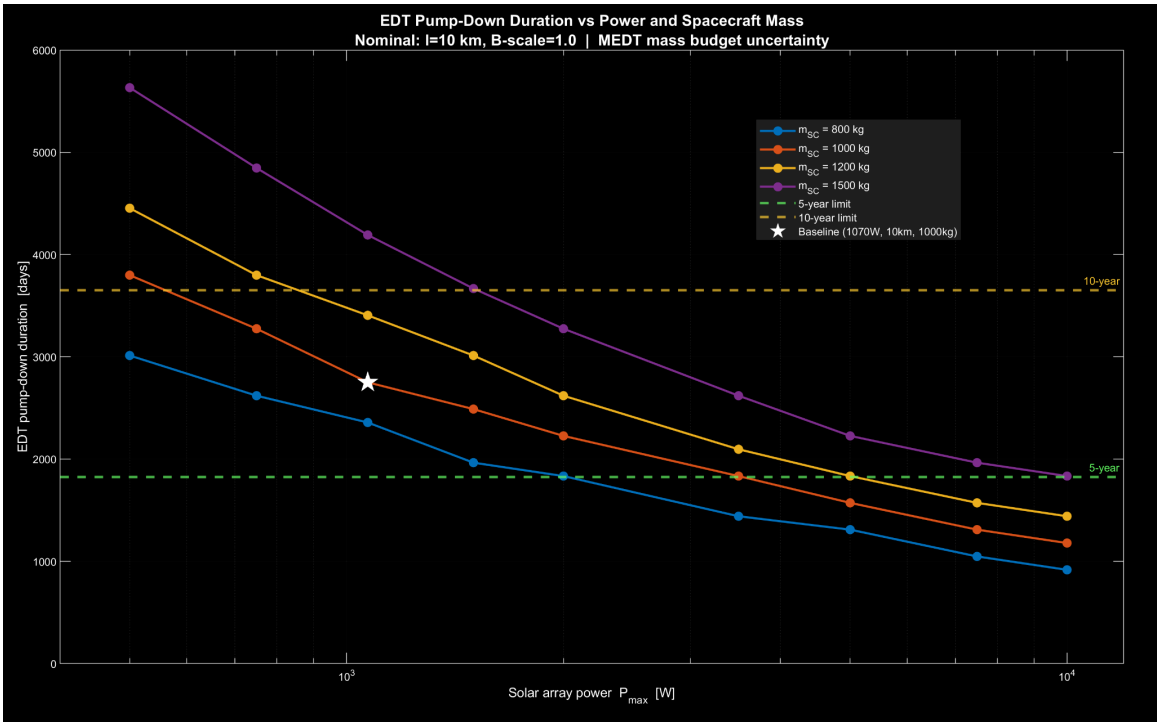


Figure 4.12 — EDT pump-down duration as a function of power budget for four spacecraft mass values (800, 1,000, 1,200, 1,500 kg), at nominal tether length (10 km) and B-field scale factor (1.0).

4.6.5 Feasibility Envelope Summary

Across all 864 combinations, the sweep confirms that EDT pump-down to Io manifold entry is feasible across a wide range of hardware configurations. The key findings are: tether length is the dominant driver at low power levels; power and tether length interact nonlinearly at high values; B-field uncertainty of ±20% produces ±8.7% variation in pump-down duration at the baseline; and spacecraft mass is the most mission-critical constraint at the margins of the feasibility envelope. The MEDT series configuration (18.4 km effective length) represents the operationally preferred architecture, as characterized in Section 4.6.2.

4.7 Capture Orbit Geometry

4.7.1 Perijove Sweep Design

The baseline mission architecture fixes perijove at 8 R_J, placing the spacecraft well inside Jupiter's high-radiation zone for the duration of Phase 1 pump-down. To characterize the trade between EDT performance, radiation exposure, and pump-down duration, a sweep was conducted across 23 perijove radii from 1 to 26 R_J with apojove fixed at 200 R_J. The high-radiation zone boundary was set at 17 R_J, consistent with the inner Jovian radiation belt structure; the baseline perijove of 8 R_J lies well inside this boundary.

4.7.2 Pump-Down Duration vs. Perijove Radius

The sweep results reveal a non-monotonic relationship between perijove radius and pump-down duration reflecting two competing effects, assuming we hold the apojove at a constant distance. At very low perijove radii (1-6 R_J), the strong magnetic field and high orbital velocity produce large Lorentz forces per pass, but the initial Jacobi constant at capture is lower since the spacecraft enters a tighter orbit requiring less total ΔC to open the L1 neck. This results in extremely short pump-down durations: at 1 R_J perijove the neck opens in 124 days across a single orbit.

The transition from single-orbit pump-down to multi-year pump-down occurs between 6.5 and 7.5 R_J, where two effects compound simultaneously: the Jacobi constant at capture increases from 2.980 to 3.188 nd, requiring 3.5 times more total energy removal, while the per-orbit ΔC decreases as the weaker field and lower orbital velocity at higher perijove reduce the Lorentz force per pass. This compound effect produces a sharp jump from 130 days at 6.5 R_J to 2,088 days at 7.5 R_J, visible in Figure 4.13. At perijove radii from 8 to 10 R_J, pump-down durations range from 3,013 to 3,854 days, spanning the 5-year and 10-year feasibility boundaries. Beyond 10 R_J, pump-down duration increases steadily, with all cases exceeding the 10-year threshold.

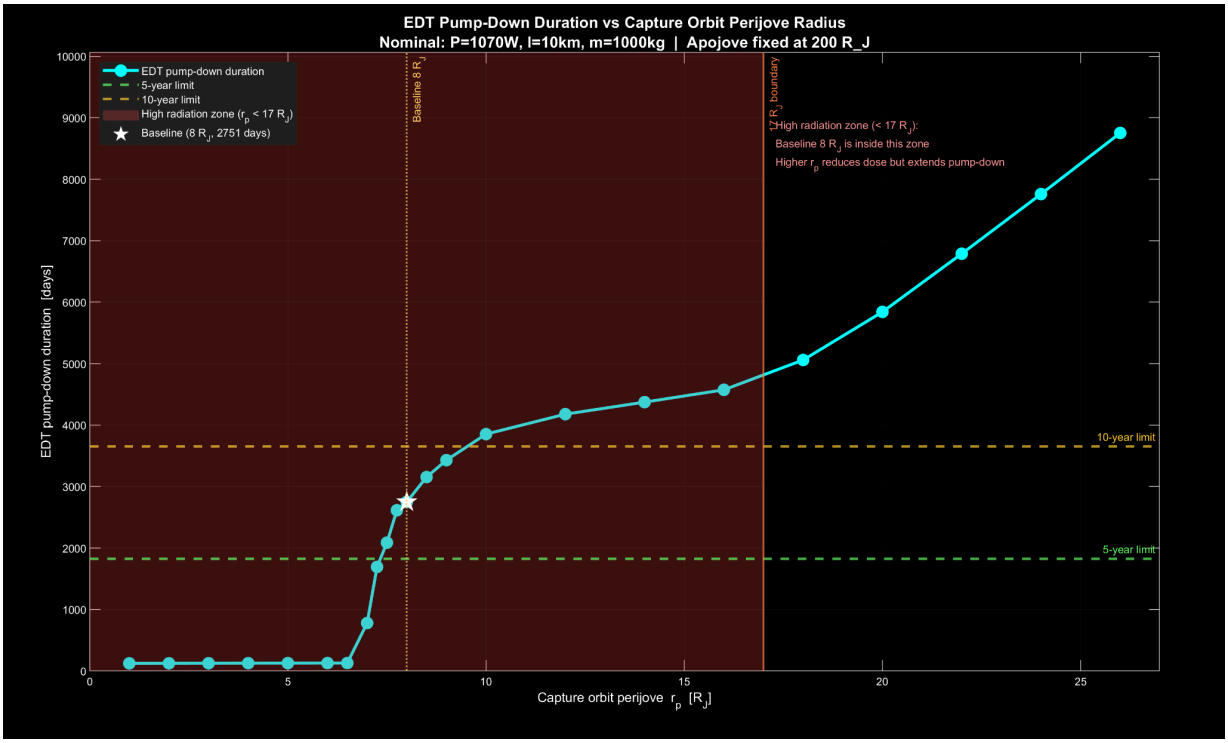


Figure 4.13 — EDT pump-down duration vs. perijove radius with radiation zone shading. The non-monotonic relationship reflects the competing effects of stronger Lorentz force at lower perijove against lower required ΔC. The red shaded region indicates the high-radiation zone inward of 17 R_J.

4.7.3 Radiation Dose vs. Perijove Radius

The radiation environment presents the opposing constraint. Total ionizing dose over the pump-down trajectory was estimated using the simplified dipole dose model with dose accumulation concentrated at perijove passes inside the Io plasma torus. At 1 R_J perijove, the estimated TID is 923,027 krad — 9,230 times the Europa Orbiter planning reference of 100 krad behind standard aluminum shielding — driven by the extremely intense inner radiation belts despite the short pump-down duration [6, 10]. The TID falls steeply with increasing perijove radius, reaching a local minimum near 9-10 R_J before rising again as longer pump-down durations at outer perijoves accumulate dose over more time.

The baseline at 8 R_J produces a TID of approximately 5,456 krad over the full pump-down, which is 54.6 times the Europa Orbiter radiation budget and approximately 8.4 times the Galileo spacecraft's total accumulated dose of 650 krad over 8 years [6]. The cumulative TID accumulation is shown in Figure 4.15; the staircase pattern reflects dose spikes at each perijove pass, confirming that EDT work and radiation exposure are both concentrated at perijove rather than distributed across the full orbit.

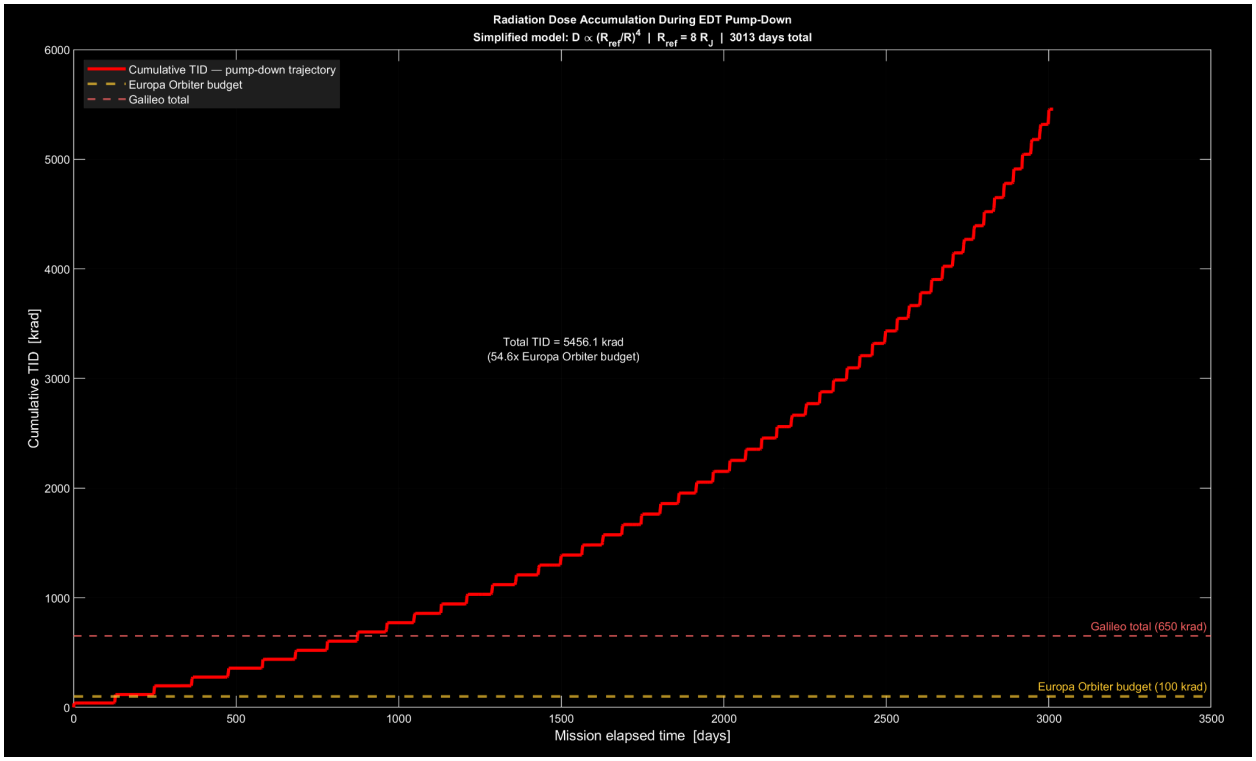


Figure 4.14 — Cumulative TID accumulation during Phase 1 EDT pump-down. The staircase pattern reflects dose spikes at each perijove pass, confirming that EDT work and radiation exposure are both concentrated at perijove rather than distributed across the full orbit.

4.7.4 The Perijove Trade

The perijove sweep makes the fundamental mission design trade explicit and is summarized in Table 4.6. Moving perijove inward reduces pump-down duration dramatically but exposes the spacecraft to radiation doses that would be fatal to current radiation-hardened electronics without extraordinary shielding. Moving perijove outward reduces radiation exposure but extends pump-down beyond the 10-year feasibility threshold. The 8 R_J baseline represents the operationally viable compromise: pump-down completes within 10 years, TID is severe but bounded, and the EDT operates within the Io plasma torus where plasma density supports current generation.

Table 4.6 — Perijove sweep summary (apojove fixed at 200 R_J, baseline power and tether).

r_p [R_J]	C_{cap} [nd]	T_{pump} [days]	TID [krad]	TID / Europa Orbiter	Feasibility
1	1.220	124	923,027	9,230×	✓ <5yr / ✗ radiation
3	2.060	126	11,566	116×	✓ <5yr / ✗ radiation
5	2.630	128	1,521	15×	✓ <5yr / ✗ radiation
6	2.873	129	739	7.4×	✓ <5yr / ✗ radiation
6.5	2.980	130	539	5.4×	✓ <5yr / ✗ radiation
7	3.086	780	2,411	24×	✓ <5yr / ✗ radiation
7.25	3.137	1,694	4,547.3	45.5×	✓ <5yr / ✗ radiation
7.5	3.188	2,088	4,896	49×	✓ <10yr
7.75	3.238	2,615	5,377	54×	✓ <10yr
8	3.286	3,013	5,456	54.6×	✓ <10yr (baseline)
8.5	3.382	3,155	4,484	45×	✓ <10yr
9	3.474	3,430	3,878	39×	✓ <10yr
10	3.650	3,854	2,859	29×	~ >10yr
14	4.267	4,374	845	8×	~ >10yr
18	4.786	5,060	358	4×	~ >10yr
26	5.637	8,753	142	1.4×	~ >10yr

This result establishes radiation shielding as the primary design driver for subsequent mission development, rather than propulsion or trajectory mechanics. No configuration within the CR3BP-LT framework simultaneously achieves short pump-down duration, low radiation exposure, and 10-year mission feasibility without extraordinary shielding.

5. Discussion

5.1 Near-Heteroclinic Connection and Phase-Space Gap Analysis

The headline result is that an EDT pump-down followed by a small EDT steering correction can place the spacecraft onto the natural gravitational highway to Io propellantlessly and within a hardware-validated design envelope. To interpret the phase-space gaps that quantify this proximity, it is worth being precise about what a true heteroclinic connection requires. In the CR3BP, a heteroclinic connection exists when a trajectory lies exactly on the stable manifold of a periodic orbit, with a gap metric identically zero. This is achievable only at a specific combination of arrival geometry, launch epoch, and energy level, and in general requiring a dedicated optimization pass. The finite gaps reported here are therefore not failures but quantifications: they measure how close the post-pump-down trajectory comes to the stable manifold targeting set at the identified arrival angles, and how much EDT steering is required to close the remaining separation propellantlessly.

The two cases bound a genuine design trade. The partial pump-down case produces a tighter Poincaré gap and more crossing opportunities but leaves the L1 neck technically closed at termination, requiring Phase 2 EDT steering to both open the neck and thread the manifold corridor. The full pump-down case opens the neck completely and reduces Phase 2 to a trim maneuver at the cost of 262 additional days. Both cases identify actionable near-heteroclinic connections within EDT bridging capability, establishing that the result is robust across the pump-down strategy design space rather than dependent on a single configuration.

The sub-degree angular sensitivity that makes these connections geometrically precise has a direct physical explanation. The instability index of ~1990 reported in Section 4.3.2 means the stable manifold is a codimension-one surface in the six-dimensional phase space with effectively zero measure. A randomly chosen trajectory has no chance of lying on it. The narrow angular window reflects the small cross-sectional area the tube presents at the Poincaré section, and the dominant velocity components in the stable eigenvector $\mathbf{v}_s$ explain why that cross-section is narrow specifically in the $\dot{x}^*$ direction: the capture trajectory and the manifold cluster overlap significantly in x^* but are offset in velocity. The ordered crossing structure at $\theta = 167°$ and $\theta = 151°$, depicted as coherent clusters rather than the chaotic scatter seen at neighboring angles, reflects a deeper dynamical distinction. This directional dominance is visible in Figures 4.7 and 4.8, where the capture trajectory crossings and the W^s cluster share similar x^* values near x_{L1} but are separated primarily in $\dot{x}^*$. The velocity equivalence $\Delta v \approx \Delta \cdot V^*$ therefore provides a physically meaningful but approximate lower bound on the true steering requirement, since the gap metric is computed from the two-dimensional $(x^*, \dot{x}^*)$ projection of the full six-dimensional phase space as noted in Section 3.8 — the actual maneuver cost depends on the full state error at

the crossing, which requires higher-fidelity targeting to resolve precisely. Near the manifold boundary, phase space is organized by the stable and unstable directions of the Lyapunov orbit, producing predictable crossing patterns; away from it, trajectories enter the chaotic sea where exponential sensitivity rapidly destroys coherence [15]. That the capture trajectory at the optimal angles passes through the organized region rather than the chaotic sea confirms that the near-heteroclinic connections identified here are geometrically meaningful rather than numerical coincidences.

5.2 EDT as a Manifold Bridging Mechanism

The EDT serves two functionally distinct roles across the transfer sequence, and distinguishing them is important for understanding what this study actually demonstrates. In Phase 1, the EDT operates as an energy extraction mechanism: the Lorentz force applied at each perijove pass does negative work against the spacecraft's orbital velocity, reducing the Jacobi constant orbit by orbit until the L1 neck opens. In Phase 2, the EDT operates as a precision navigator: the residual phase-space gap between the post-pump-down trajectory and the stable manifold targeting set is bridged by a controlled EDT steering maneuver. These two roles are physically distinct and analytically separable, even though the same hardware executes both.

Phase 1 is a topological prerequisite, not an optional optimization. As established in Section 2.4.2, the L1 neck cannot open until C drops below C_{L1}, and no trajectory geometry can substitute for this energy condition. The pump-down duration is simply the time cost of closing ΔC_{total}. Phase 2 is where the more subtle result lives. The gap between the post-pump-down trajectory and the stable manifold is predominantly in the $\dot{x}^*$ direction, which is a velocity offset rather than a position offset. This makes the EDT steering maneuver functionally analogous to a small impulsive burn, but delivered propellantlessly through the Lorentz force. What matters is not the size of the gap in absolute terms but that it falls within EDT bridging capability. Both cases satisfy this condition, and that is the result: the EDT is not merely a capture mechanism that happens to reduce orbital energy, but a precision manifold-targeting tool capable of threading a spacecraft onto a codimension-one surface in a chaotic dynamical environment [15]. Phase 1 opens the door, and Phase 2 walks through it.

The manifold-as-template assumption stated in Section 2.4.3 is load-bearing for this interpretation. At the force magnitudes produced by the baseline hardware configuration the perturbation is sufficiently small that the assumption is well-justified at pre-Phase A fidelity, but quantitative verification against tether-perturbed manifold families is the natural first follow-on step before these results serve as a Phase A design reference.

5.3 Mission Feasibility and Sensitivity

The sensitivity sweep across 864 hardware configurations confirms that EDT pump-down to Io manifold entry is feasible across a wide range of design choices, with the key findings from each parameter axis reported in Section 4.6. The nonlinear interaction between power and tether length means neither parameter alone characterizes the design space, and spacecraft mass is the most mission-critical constraint at the margins of the feasibility envelope. The MEDT series configuration represents the operationally preferred architecture within the survivability envelope, as established in Section 4.6.2.

Radiation emerges from the perijove sweep as the primary design driver for subsequent development, and this conclusion deserves emphasis. The trajectory and propulsion problems addressed in this study are solved: EDT pump-down is feasible within ten years across a validated hardware envelope, and the manifold connection is demonstrated. What the perijove sweep makes explicit is that no configuration simultaneously achieves short pump-down duration, low radiation exposure, and ten-year mission feasibility without extraordinary shielding. The baseline perijove of 8 R_J produces a total ionizing dose of approximately 5,456 krad over the full pump-down duration — 54.6 times the Europa Orbiter planning reference of 100 krad behind standard aluminum shielding [10]. Moving perijove inward reduces pump-down duration dramatically but delivers radiation doses fatal to current radiation-hardened electronics. Moving perijove outward reduces exposure but pushes pump-down beyond the ten-year feasibility threshold. The 8 R_J baseline is the operationally viable compromise the CR3BP-LT framework can identify, but radiation shielding design, rather than propulsion or trajectory mechanics, is the primary binding constraint that will determine whether this mission architecture is ultimately executable.

5.4 Model Limitations and Fidelity Escalations

The modeling assumptions enumerated in Section 2.7 were each selected to preserve the dynamical structure essential to the transfer design while keeping the analysis tractable. The results of Chapters 3 and 4 allow a more specific assessment of which assumptions are most load-bearing and which escalations are most urgent.

The manifold-as-template assumption is one of the most consequential. The near-heteroclinic connections identified at $\theta = 167°$ and $\theta = 151°$ depend on the stable manifold geometry computed from the unperturbed CR3BP remaining a valid target under continuous EDT forcing. The force magnitudes produced by the baseline hardware configuration are small relative to the gravitational dynamics at the perijove altitudes involved, which supports the assumption at pre-Phase A fidelity, however quantitative verification against tether-perturbed manifold families is the natural first escalation step before these results are used as a Phase A design reference. The dipole magnetic field model also has a direct and quantified effect on the

results: the ±20% B-field uncertainty sweep in Section 4.6.3 shows ±8.7% variation in pump-down duration at the baseline configuration. Since the dipole model may overestimate field strength at certain orbital geometries relative to high-fidelity empirical models such as VIP4, the pump-down durations reported here should be treated as optimistic bounds pending escalation to the VIP4 empirical field model [29]. The always-on EDT assumption is the most operationally optimistic. The duty cycle sensitivity analysis in Section 4.6 confirms that pump-down duration scales nearly linearly with duty cycle, so the always-on results provide a clean lower bound that translates directly to realistic configurations without structural reanalysis. Furthermore, the planar assumption and the single-moon model are appropriate for establishing the existence of the manifold connection, but both become more consequential as the analysis extends toward manifold transit and the inter-moon tour. The chaotic sensitivity near L1 means that even small perturbations from outer moon gravity can alter the tube geometry and crossing structure identified here, making the four-body escalation a priority for future development.

Together, these limitations define a clear fidelity roadmap. Each escalation item refines the propellantless manifold connection established here without invalidating it, building toward the higher-fidelity trajectory design required for a formal Phase A concept study.

5.5 Comparison to Prior Work

The results of this study sit at the intersection of two mature research threads — EDT propulsion for Jovian capture and CR3BP manifold-based transfer design — that have until now developed largely in parallel. Positioning the contribution of this study requires distinguishing it carefully from prior work in each thread.

Magnetour [8] represents the most direct predecessor to this study. As discussed in Section 1.2.3, Magnetour proposed combining EDT electromagnetic braking with the InterMoon Superhighway framework and demonstrated that the combination was physically plausible. The present study advances beyond Magnetour in three specific respects. First, the EDT hardware model adheres to the survivability constraints established by Casalino et al. [12] and Sanmartin et al. [7], which invalidated the tether lengths exceeding 200 km assumed in Magnetour's tether-perturbed orbit analysis. Second, the manifold analysis is quantitative rather than conceptual: a Poincaré section gap metric is computed, arrival angle sweeps are performed, and the phase-space distance the EDT must bridge is explicitly characterized. Third, and most fundamentally, the EDT is demonstrated here as the bridging mechanism that connects the post-pump-down capture orbit onto the L1 stable manifold of a target moon — the specific integration that Magnetour identified as a future direction but did not demonstrate.

Alkhenani and Oguri [1, 26] established the methodological foundation for EDT trajectory optimization in the Jovian system through rigorous application of Pontryagin's minimum principle to EDT-propelled JOI capture. The present study builds directly on that

methodological foundation while extending the problem scope from JOI capture to the subsequent manifold transit phase. Where Alkhenani and Oguri focused on the optimal control problem of capture, this study focuses on the dynamical systems problem of manifold access — the two together constitute the complete post-interplanetary-cruise trajectory sequence from capture through Io approach.

The foundational CR3BP manifold targeting framework of Koon et al. [15] and its Jovian applications by Vaquero and Howell [18, 19, 20], Lantoine et al. [9], and Campagnola and Kawakatsu [22] established the theoretical and computational tools used throughout this study. The contribution here is not to the manifold targeting methodology itself, which is well-established, but to its application in a physically constrained setting: a power-limited EDT system operating within hardware survivability constraints in the inner Jovian radiation environment, where no prior manifold-targeting study has operated.

Sanmartin et al. [7] and Casalino et al. [12] established the hardware constraints that bound the EDT design space in this study. Their survivability analysis is treated here as a given input rather than a result, and the trajectory framework is designed to operate within those constraints from the outset. The sensitivity analysis of Chapter 3 extends their hardware characterization by mapping the EDT performance envelope directly onto pump-down duration and manifold accessibility, providing a quantitative link between hardware design choices and trajectory feasibility that prior work had not established.

Taken together, this study occupies a specific position in the literature: it is the first to demonstrate a quantified near-heteroclinic connection between a post-JOI capture orbit and the L1 stable manifold of the Jupiter-Io system under continuous EDT perturbation within a hardware-validated CR3BP-LT framework — closing the gap that Magnetour identified as necessary but left undemonstrated, as characterized in Section 1.2.3. Whether a true heteroclinic connection exists at a specific launch epoch remains an open question requiring full ephemeris reconstruction, but the existence of near-heteroclinic connections within EDT bridging capability at pre-Phase A fidelity is established here for the first time.

6. Conclusions and Future Work

6.1 Conclusions

This study develops and demonstrates a propellantless trajectory framework for a conceptual electrodynamic tethered spacecraft targeting Io, the innermost Galilean moon of Jupiter. The central analytical objective was to establish a near-heteroclinic connection between the post-JOI capture orbit and the L1 stable manifold of the Jupiter-Io system under continuous EDT perturbation within the CR3BP-LT framework, demonstrating that propellantless manifold transit to Io is dynamically feasible as a pre-Phase A result.

Three primary conclusions follow from the results. First, propellantless manifold transit to Io is dynamically feasible within the CR3BP-LT framework. Near-heteroclinic connections are identified at phase-space gaps of 0.0079 nd at arrival angle $\theta = 167°$ for the partial pump-down case and 0.0179 nd at $\theta = 151°$ for the full pump-down case, both within EDT bridging capability. The EDT is shown to serve two functionally distinct roles: as an energy extraction mechanism in Phase 1 that reduces the Jacobi constant until the L1 neck opens, and as a precision manifold-targeting tool in Phase 2 that bridges the residual phase-space gap propellantlessly. Together these phases constitute a complete propellantless transfer architecture from Jupiter orbit insertion through L1 manifold entry. Second, EDT pump-down to Io manifold entry is feasible across a wide range of hardware configurations within a ten-year mission lifetime. The MEDT series configuration, consisting of two 9.2 km tethers in series for an effective length of 18.4 km, reduces pump-down to approximately 1,900 days at baseline power and nominal magnetic field, well within the five-year mission window. Tether length is the dominant driver at low power levels, while spacecraft mass is the most mission-critical constraint at the margins of the feasibility envelope. Magnetic field uncertainty of ±20% produces ±8.7% variation in pump-down duration at the baseline, confirming that the mission remains feasible across the full dipole model uncertainty range. Third, radiation shielding is the primary design driver for subsequent development, not propulsion or trajectory mechanics. The baseline perijove of 8 R_J produces a total ionizing dose of approximately 5,456 krad over the full pump-down duration, 54.6 times the Europa Orbiter planning reference. The perijove trade establishes that no configuration within the CR3BP-LT framework simultaneously achieves short pump-down duration, low radiation exposure, and ten-year mission feasibility without extraordinary shielding. Resolving this constraint is the binding engineering problem for subsequent mission development. Together these results establish the analytical foundation necessary to advance the mission concept through pre-Phase A toward preliminary design, contributing the first quantified near-heteroclinic connection between a post-JOI capture orbit and the Jupiter-Io L1 stable manifold under hardware-validated EDT constraints.

6.2 Future Work

Several analytical extensions identified during the course of this work represent the natural next steps for maturing the trajectory design toward a formal concept study.

The highest-priority escalation is fidelity escalation from the CR3BP-LT model to the planar concentric circular four-body problem and the circular restricted N-body problem. Incorporating gravitational perturbations from Europa, Ganymede, and Callisto as additional perturbing bodies is particularly important for the manifold transit phase, where the chaotic sensitivity of the dynamics near L1 means that even small perturbations from outer moon gravity can alter the tube geometry and crossing structure identified here. This escalation is also the prerequisite for extending the trajectory framework to the full InterMoon Superhighway multi-moon tour beyond the scope of this study.

Monte Carlo robustness analysis quantifying the sensitivity of the near-heteroclinic connection to navigation error, magnetic field uncertainty, and tether performance variation is the second priority. The sub-degree angular sensitivity identified in Section 4.5.2 means that the connection is geometrically precise, and characterizing the uncertainty envelope around the optimal arrival angles is essential before the result can be used as a design reference for a Phase A concept study. Stochastic optimal control formulations incorporating these uncertainty sources would provide the uncertainty-bounded trajectory envelopes required for credible mission planning.

Radiation-integrated trajectory optimization, treating cumulative total ionizing dose as an explicit path constraint rather than a post-hoc evaluation metric, is the third priority. The perijove trade characterized in Section 4.7 demonstrates that radiation exposure and pump-down duration are in direct tension, but the current framework optimizes neither, instead characterizing the trade parametrically. An optimal control formulation that minimizes TID accumulation subject to manifold accessibility and mission lifetime constraints would provide a principled basis for perijove selection and shielding design.

Full ephemeris reconstruction in a high-fidelity propagator, translating the optimal arrival angles identified in Chapter 4 to specific launch dates and trajectories, is required before the near-heteroclinic connection can be operationally targeted. The CR3BP analysis identifies the arrival geometry that minimizes the phase-space gap but does not account for the actual positions of Jupiter, Io, and the spacecraft at a given epoch. GMAT or STK reconstruction incorporating real ephemeris data is the appropriate tool for this translation.

Finally, escalation of the magnetic field model from the centered dipole approximation to the VIP4 empirical field model, which incorporates Galileo magnetometer data and accounts for the higher-order multipole structure and azimuthal asymmetry confirmed by Juno, would refine

the pump-down duration estimates and provide tighter bounds on the feasibility envelope. Since the dipole model may overestimate field strength at certain orbital geometries, the pump-down durations reported here should be treated as optimistic bounds pending this validation. Realistic duty cycle modeling incorporating altitude-dependent tether activation and plasma density variation would further refine the pump-down duration estimates beyond the always-on baseline characterized here.

Together these extensions form the analytical agenda for the subsequent phase of mission development. The results presented in this study provide the foundation upon which that higher-fidelity work depends

References

[1] Alkhenani, A., and Oguri, K., "Jovian System Exploration via Electrodynamic Tethers with Optimal and Analytical Control Laws," Journal of Spacecraft and Rockets, Vol. 62, No. 1, 2025. https://doi.org/10.2514/1.A36360

[2] Lu, P., Wang, Y., and Cui, S., "Vertical Instability of the Planar Resonant Orbits and the Application in Transfer Design to Three-Dimensional Periodic Orbits," Acta Astronautica, Vol. 228, 2025, pp. 740–754. https://doi.org/10.1016/j.actaastro.2024.12.043

[3] Wang, Y., Lu, P., and Fu, T., "Transfers to Frozen Orbits Around Planetary Moons Using Manifolds of Averaged Dynamics," Journal of Guidance, Control, and Dynamics, Vol. 47, No. 2, 2024, pp. 262–278. https://doi.org/10.2514/1.G007774

[4] Biesbroek, R., Lunar and Interplanetary Trajectories, Springer International Publishing, Cham, 2016.

[5] Parrish, N. L., and Scheeres, D. J., "Low-Thrust Trajectory Optimization with Simplified SQP Algorithm," presented at the AAS/AIAA Astrodynamics Specialist Conference, Stevenson, WA, 2017.

[6] Gallagher, D. L., Johnson, L., Moore, J., and Bagenal, F., "Electrodynamic Tether Propulsion and Power Generation at Jupiter," Technical Publication NASA/TP-1998-208475, NASA Marshall Space Flight Center, Huntsville, AL, June 1998.

[7] Sanmartin, J. R., Charro, M., Lorenzini, E. C., Garrett, H. B., Bombardelli, C., and Bramanti, C., "Electrodynamic Tether at Jupiter—I: Capture Operation and Constraints," IEEE Transactions on Plasma Science, Vol. 36, No. 5, 2008, pp. 2450–2458. https://doi.org/10.1109/TPS.2008.2002580

[8] Lantoine, G., Russell, R. P., Anderson, R. L., and Garrett, H. B., "Magnetour: Surfing Planetary Systems on Electromagnetic and Multi-Body Gravity Fields," Acta Astronautica, Vol. 138, 2017, pp. 543–558. https://doi.org/10.1016/j.actaastro.2016.12.014

[9] Lantoine, G., Russell, R. P., and Campagnola, S., "Optimization of Low-Energy Resonant Hopping Transfers between Planetary Moons," Acta Astronautica, Vol. 68, Nos. 7–8, 2011, pp. 1361–1378. https://doi.org/10.1016/j.actaastro.2010.09.021

[10] Sánchez-Arriaga, G., Lorenzini, E. C., and Bilén, S. G., "A Review of Electrodynamic Tether Missions: Historical Trend, Dimensionless Parameters, and Opportunities Opening Space Markets," Acta Astronautica, Vol. 225, 2024, pp. 158–168. https://doi.org/10.1016/j.actaastro.2024.09.002

[11] Sanmartin, J. R., Lorenzini, E. C., and Martinez-Sanchez, M., "Electrodynamic Tethers for Exploration of Jupiter and Its Icy Moons," Journal of Propulsion and Power, Vol. 24, No. 2, 2008, pp. 212–222. https://doi.org/10.2514/1.31352

[12] Casalino, L., Arnone, E., and Peano, F., "Conceptual Design of Electrodynamic Multi-Tether System for Self-Propelled Jovian Capture," Acta Astronautica, Vol. 186, 2021, pp. 269–279. https://doi.org/10.1016/j.actaastro.2021.04.031

[13] Williams, P., "Optimal Orbit Transfer with Electrodynamic Tether," Journal of Guidance, Control, and Dynamics, Vol. 28, No. 2, 2005, pp. 369–372. https://doi.org/10.2514/1.12016

[14] Kojima, H., and Trivailo, P. M., "Evaluation of Transition Performance to Jupiter Orbit Using Electrodynamic Tether System," presented at the 18th Australian International Aerospace Congress, Melbourne, Australia, 2019.

[15] Koon, W. S., Lo, M. W., Marsden, J. E., and Ross, S. D., Dynamical Systems, the Three-Body Problem and Space Mission Design, v1.3, Caltech/JPL/Virginia Tech, October 2022. http://www.shaneross.com/books/

[16] Vaquero, M., "Poincaré Sections and Resonant Orbits in the Restricted Three-Body Problem," M.S. thesis, Purdue University, West Lafayette, IN, 2010.

[17] Lantoine, G., and Russell, R. P., "Near Ballistic Halo-to-Halo Transfers between Planetary Moons," The Journal of the Astronautical Sciences, Vol. 58, No. 3, 2011, pp. 335–363. https://doi.org/10.1007/BF03321174

[18] Vaquero, M., and Howell, K. C., "Design of Transfer Trajectories between Resonant Orbits in the Earth–Moon Restricted Problem," Acta Astronautica, Vol. 94, No. 1, 2014, pp. 302–317. https://doi.org/10.1016/j.actaastro.2013.05.006

[19] Vaquero, M., and Howell, K. C., "Leveraging Resonant-Orbit Manifolds to Design Transfers Between Libration-Point Orbits," Journal of Guidance, Control, and Dynamics, Vol. 37, No. 4, 2014, pp. 1143–1157. https://doi.org/10.2514/1.62230

[20] Vaquero Escribano, T. M., "Spacecraft Transfer Trajectory Design Exploiting Resonant Orbits in Multi-Body Environments," Ph.D. dissertation, Purdue University, West Lafayette, IN, 2013.

[21] Campanella, G., Boutonnet, A., and Schoenmaekers, J., "Resonance Hopping Transfers Between Moon Science Orbits," presented at the 19th AAS/AIAA Space Flight Mechanics Meeting, Savannah, GA, 2009.

[22] Campagnola, S., and Kawakatsu, Y., "Jupiter Magnetospheric Orbiter: Trajectory Design in the Jovian System," Journal of Spacecraft and Rockets, Vol. 49, No. 2, 2012, pp. 318–324. https://doi.org/10.2514/1.A32055

[23] Bombardelli, C., Baú, G., and Peláez, J., "Periodic Orbits and Equilibria Near Jovian Moons Using an Electrodynamic Tether," Journal of Guidance, Control, and Dynamics, Vol. 38, No. 1, 2015, pp. 78–92. https://doi.org/10.2514/1.G000434

[24] Bombardelli, C., Baú, G., and Peláez, J., "Lyapunov Orbits in the Jupiter System Using Electrodynamic Tethers," Journal of Guidance, Control, and Dynamics, Vol. 36, No. 6, 2013, pp. 1761–1771. https://doi.org/10.2514/1.59734

[25] Bombardelli, C., and Peláez, J., "On the Stability of Artificial Equilibrium Points in the Circular Restricted Three-Body Problem," Celestial Mechanics and Dynamical Astronomy, Vol. 109, No. 1, 2011, pp. 13–26. https://doi.org/10.1007/s10569-010-9317-z

[26] Alkhenani, A. K., Oguri, K., and Mueterthies, M. J., "Optimal Trajectories of Electrodynamic Tethers in the Jovian System Using Indirect Method," presented at the AAS/AIAA Astrodynamics Specialist Conference, Broomfield, CO, August 2024.

[27] Yang, H., Hu, J., Bai, X., and Li, S., "Review of Trajectory Design and Optimization for Jovian System Exploration," Space: Science & Technology, Vol. 3, 2023, p. 0036. https://doi.org/10.34133/space.0036

[28] Vallado, D. A., Fundamentals of Astrodynamics and Applications, 4th ed., Microcosm Press, Hawthorne, CA, 2013.

[29] Sanmartin, J. R., Charro, M., Lorenzini, E. C., Garrett, H. B., Bombardelli, C., and Bramanti, C., "Electrodynamic Tether at Jupiter—II: Fast Moon Tour After Capture," IEEE Transactions on Plasma Science, Vol. 37, No. 4, 2009, pp. 620–626. https://doi.org/10.1109/TPS.2009.2013955

[30] Parker, J. S., and Anderson, R. L., Low-Energy Lunar Trajectory Design, JPL Deep Space Communications and Navigation Series, Jet Propulsion Laboratory, Pasadena, CA, July 2013.

[31] Khalil, H. K., Nonlinear Systems, 3rd ed., Prentice Hall, Upper Saddle River, NJ, 2002.

Appendix A: Matlab Code

```
function cr3bp_jupiter_io_sim_with_lagrange()
%% SECTION 1: INPUTS AND MASTER PARAMETER BLOCK
% ── Physical constants
G = 6.67430e-20;              % Gravitational constant [km^3 kg^-1 s^-2]
% ── Body masses
m1 = 1.89813e27;              % Jupiter mass [kg]
m2 = 8.9319e22;               % Io mass [kg]
% ── Spacecraft hardware parameters
mSC_kg     = 1000;            % Spacecraft dry mass [kg]
l_tether_m = 10000;           % Effective tether length [m] = 9.2 km
P_max_W    = 1070.0;          % Solar array power [W]
                              % Derived: ~107 m² array at 20% efficiency
                              % and 50.2 W/m² Jovian solar flux, or
                              % ~150 m² at ~14% end-of-life efficiency
                              % accounting for radiation degradation
I_max_A    = 100.0;           % Physical current cap [A]
B_scale    = 1.0;             % B-field scaling factor
                              % 1.0 = nominal dipole model
                              % 0.8/1.2 = ±20% uncertainty bounds
                              % (swept in Section 12b)
% ── Capture orbit geometry
r_p_RJ = 8;                   % Capture orbit perijove [R_J]
                              % Note: 8 R_J is inside the intense
                              % radiation zone boundary at 17-20 R_J;
                              % tradeoff vs pump-down duration quantified
                              % in Section 12c perijove sweep
r_a_RJ = 200.0;               % Capture orbit apojove [R_J]
% ── Derived capture orbit parameters
% Computed from r_p_RJ and r_a_RJ — do not edit these directly
R_J_km  = 71492.0;            % Jupiter mean equatorial radius [km]
r_p_joi = r_p_RJ * R_J_km;    % Perijove radius [km]
```

```
r_a_joi = r_a_RJ * R_J_km;    % Apojove radius [km]
% ── Celestial body states (Jupiter-centered inertial, JCI) ──────────────────
rIo_JCI = [421800; 0; 0];      % Io position [km]; on +x axis at t=0
vIo_JCI = [0; 17.334; 0];      % Io velocity [km/s]; circular prograde
% Test integration initial state (used in Section 5 only)
rSC_JCI = [1000000; 0; 0];     % Test orbit position [km] (~14 R_J)
vSC_JCI = [0; 11.256; 0];      % Test orbit velocity [km/s]
% ── Integration settings
─────────────────────────────────────────────────────────────────────
nRevs     = 10;               % Io periods for Section 5 test integration
tau_final = nRevs * 2*pi;     % Final normalized time [nd]
rtol      = 1e-10;            % ODE relative tolerance
atol      = 1e-12;            % ODE absolute tolerance
% ── Output flags
─────────────────────────────────────────────────────────────────────────
──
exportInertialBack = true;     % Back-transform to JCI frame for plotting
% ── ODE parameter struct
─────────────────────────────────────────────────────────────────────
% All hardware parameters flow from master values above.
% p.mu, p.n, p.Lstar, p.Vstar added in Section 4 after normalization.
p.R_J_km  = R_J_km;
p.P_max_W = P_max_W;
p.I_max_A = I_max_A;
p.mSC_kg  = mSC_kg;
p.l_m     = l_tether_m;
p.B_scale = B_scale;
p.edt_on  = true;              % Default on; overridden locally where needed
p.duty_cycle = 1.0;   % EDT duty cycle [0-1]: fraction of orbit with tether active
                      % 1.0 = always-on (thesis baseline, idealized)
                      % <1.0 = realistic operational constraint
                      % Accounts for: attitude control off-windows, plasma
                      % density gaps near apojove, thermal recovery periods
                      % See Section 13b for sensitivity sweep
p.plasma_scaling = false;

%% SECTION 2: JOI BURN AND CHEMICAL BASELINE COMPARISON
──────────────────────────

% Establishes the conventional propulsion baseline that EDT aims to replace.
% Computes the delta-V required for Jupiter Orbit Insertion (JOI) at the
```

```
% master-block perijove and the corresponding propellant mass for each
% propulsion type. This is the "Point A" starting condition — the state
% immediately after the JOI burn, from which the EDT tether takes over.
fprintf('\n── JOI Chemical Baseline
──────────────────────────────────────────────────────────\n');
% ── Step 1: Derive v_inf from Hohmann transfer Earth to Jupiter ──────────────
fprintf('\n  Step 1: Hohmann transfer Earth → Jupiter\n');
mu_sun   = 1.32712440018e11;   % Solar GM [km^3/s^2]
AU_km    = 1.495978707e8;      % 1 AU [km]
r_Earth  = 1.0000 * AU_km;     % Earth orbital radius [km]
r_Jup    = 5.2044 * AU_km;     % Jupiter orbital radius [km]
a_transfer     = 0.5 * (r_Earth + r_Jup);
v_transfer_Jup = sqrt(mu_sun * (2/r_Jup - 1/a_transfer));
v_Jup_circ     = sqrt(mu_sun / r_Jup);
v_inf_computed = abs(v_Jup_circ - v_transfer_Jup);
T_transfer_s    = pi * sqrt(a_transfer^3 / mu_sun);
T_transfer_days = T_transfer_s / 86400;
T_transfer_yrs  = T_transfer_days / 365.25;
fprintf('  Earth orbital radius:          %.4f km  (%.4f AU)\n', r_Earth, r_Earth/AU_km);
fprintf('  Jupiter orbital radius:        %.4f km  (%.4f AU)\n', r_Jup,   r_Jup/AU_km);
fprintf('  Transfer ellipse SMA:          %.4e km\n',            a_transfer);
fprintf('  SC speed at Jupiter (transfer): %.4f km/s\n',         v_transfer_Jup);
fprintf('  Jupiter circular speed:        %.4f km/s\n',          v_Jup_circ);
fprintf('  v_inf at Jupiter SOI:          %.4f km/s\n',          v_inf_computed);
fprintf('  Hohmann transfer time:         %.1f days  (%.2f years)\n', ...
    T_transfer_days, T_transfer_yrs);
v_inf_joi = v_inf_computed;
fprintf('\n  Using v_inf = %.4f km/s for JOI baseline (Hohmann minimum energy)\n', v_inf_joi);
% ── v_inf sensitivity table
──────────────────────────────────────────────────────────
% Uses master block r_p_RJ and r_a_RJ — not hardcoded
fprintf('\n  v_inf sensitivity (JOI delta-V at %.0f R_J perijove):\n', r_p_RJ);
fprintf('  %-12s  %-15s  %-15s\n', 'v_inf [km/s]', 'v_hyp [km/s]', 'dv_JOI [km/s]');
fprintf('  %-12s  %-15s  %-15s\n', '------------', '------------', '-------------');
mu_J_temp  = G * m1;
r_p_temp   = r_p_RJ * R_J_km;        % from master block
r_a_temp   = r_a_RJ * R_J_km;        % from master block
a_cap_temp = 0.5 * (r_p_temp + r_a_temp);
v_cap_temp = sqrt(mu_J_temp * (2/r_p_temp - 1/a_cap_temp));
for v_test = [5.0, v_inf_computed, 7.0, 9.0]
```

```
    v_h    = sqrt(v_test^2 + 2*mu_J_temp/r_p_temp);
    dv     = v_h - v_cap_temp;
    marker = '';
    if abs(v_test - v_inf_computed) < 0.01
        marker = ' ← Hohmann baseline';
    elseif abs(v_test - 5.643) < 0.01
        marker = ' ← Section 5e reference';
    end
    fprintf('  %-12.3f  %-15.4f  %-15.4f%s\n', v_test, v_h, dv, marker);
end

% ── Hyperbolic approach and capture orbit parameters ──────────────────────
% All derived from master block — no hardcoded radii
mu_J_joi = G * m1;            % Jupiter GM [km^3/s^2]
R_J_joi  = R_J_km;            % Jupiter radius [km] — from master block
% r_p_joi and r_a_joi already set in Section 1 from master block.
% Reassigned here as local aliases for clarity within this section.
r_p_joi  = r_p_RJ * R_J_km;   % Perijove [km]
r_a_joi  = r_a_RJ * R_J_km;   % Apojove [km] — uses master block r_a_RJ
a_cap_joi = 0.5 * (r_p_joi + r_a_joi);
v_cap_joi = sqrt(mu_J_joi * (2/r_p_joi - 1/a_cap_joi));
v_hyp_joi = sqrt(v_inf_joi^2 + 2*mu_J_joi/r_p_joi);
v_circ_rp = sqrt(mu_J_joi / r_p_joi);   % circular speed at perijove
dv_JOI_ellipse = v_hyp_joi - v_cap_joi;   % minimum JOI — used throughout
dv_JOI_circ    = v_hyp_joi - v_circ_rp;   % conservative upper bound
dv_JOI_kms     = dv_JOI_ellipse;
dv_JOI_ms      = dv_JOI_kms * 1e3;
fprintf('\n  Capture orbit: %.0f x %.0f R_J ellipse\n', r_p_RJ, r_a_RJ);
fprintf('  Circular orbit speed at %.0f R_J:      %.4f km/s\n', r_p_RJ, v_circ_rp);
fprintf('  Delta-V for circular capture:         %.4f km/s\n',  dv_JOI_circ);
fprintf('  Delta-V for %.0fx%.0f R_J ellipse:      %.4f km/s  (thesis baseline)\n', ...
    r_p_RJ, r_a_RJ, dv_JOI_ellipse);
fprintf('  Hyperbolic approach speed at pjv:     %.4f km/s\n',  v_hyp_joi);
fprintf('  Capture ellipse speed at pjv:         %.4f km/s\n',  v_cap_joi);
fprintf('  JOI Delta-V required:                 %.4f km/s  (%.1f m/s)\n', ...
    dv_JOI_kms, dv_JOI_ms);
fprintf('  Perijove radius:                      %.0f km  (%.1f R_J)\n', ...
    r_p_joi, r_p_joi/R_J_km);
```

```
% ── Propellant mass comparison (rocket equation) ─────────────────────────
g0_joi    = 9.80665e-3;       % Standard gravity [km/s^2]
m_dry_joi = mSC_kg;           % from master block
Isp_cold_joi = 60;
Isp_bipr_joi = 320;
Isp_lox_joi  = 450;
Isp_nep_lo   = 1500;
Isp_nep_hi   = 6000;
m_prop_cold_joi = m_dry_joi * (exp(dv_JOI_kms/(Isp_cold_joi * g0_joi)) - 1);
m_prop_bipr_joi = m_dry_joi * (exp(dv_JOI_kms/(Isp_bipr_joi * g0_joi)) - 1);
m_prop_lox_joi  = m_dry_joi * (exp(dv_JOI_kms/(Isp_lox_joi  * g0_joi)) - 1);
m_prop_nep_lo   = m_dry_joi * (exp(dv_JOI_kms/(Isp_nep_lo   * g0_joi)) - 1);
m_prop_nep_hi   = m_dry_joi * (exp(dv_JOI_kms/(Isp_nep_hi   * g0_joi)) - 1);
m_nep_sys     = 500;
m_wet_cold_joi = m_dry_joi + m_prop_cold_joi;
m_wet_bipr_joi = m_dry_joi + m_prop_bipr_joi;
m_wet_lox_joi  = m_dry_joi + m_prop_lox_joi;
m_wet_nep_lo   = m_dry_joi + m_prop_nep_lo + m_nep_sys;
m_wet_nep_hi   = m_dry_joi + m_prop_nep_hi + m_nep_sys;
fprintf('\n  Propellant mass for JOI burn (%.4f km/s, dry mass = %.0f kg):\n', ...
    dv_JOI_kms, m_dry_joi);
fprintf('  %-28s  %8s  %10s  %10s  %8s\n', ...
    'Propulsion type','Isp [s]','Prop [kg]','Wet [kg]','MF [%]');
fprintf('  %-28s  %8s  %10s  %10s  %8s\n', ...
    '----------------------------','-------','---------','--------','------');
fprintf('  %-28s  %8.0f  %10.2f  %10.2f  %7.1f%%\n', ...
    'Cold gas', Isp_cold_joi, m_prop_cold_joi, m_wet_cold_joi, ...
    100*m_prop_cold_joi/m_wet_cold_joi);
fprintf('  %-28s  %8.0f  %10.2f  %10.2f  %7.1f%%\n', ...
    'Bipropellant MMH/NTO', Isp_bipr_joi, m_prop_bipr_joi, m_wet_bipr_joi, ...
    100*m_prop_bipr_joi/m_wet_bipr_joi);
fprintf('  %-28s  %8.0f  %10.2f  %10.2f  %7.1f%%\n', ...
    'LOX/LH2', Isp_lox_joi, m_prop_lox_joi, m_wet_lox_joi, ...
    100*m_prop_lox_joi/m_wet_lox_joi);
fprintf('  %-28s  %8.0f  %10.2f  %10.2f  %7.1f%%\n', ...
    'NEP low (NEXT-C, +500kg sys)', Isp_nep_lo, m_prop_nep_lo, m_wet_nep_lo, ...
    100*m_prop_nep_lo/m_wet_nep_lo);
fprintf('  %-28s  %8.0f  %10.2f  %10.2f  %7.1f%%\n', ...
    'NEP high (+500kg sys)', Isp_nep_hi, m_prop_nep_hi, m_wet_nep_hi, ...
```

```
    100*m_prop_nep_hi/m_wet_nep_hi);
fprintf('  %-28s  %8s  %10s  %10s\n', ...
    'EDT (post-JOI phase only)', 'N/A', '0.00', '---');
fprintf('\n  NOTE: NEP delta-V shown is equivalent energy cost for mass\n');
fprintf('  budget comparison only. NEP JOI requires months-to-years of\n');
fprintf('  continuous low-thrust spiraling, not a single impulsive burn.\n');
% ── Mission delta-V budget summary
─────────────────────────────────────────────────────────
% Phase 2 steering value is a placeholder here — overwritten by Section 9
fprintf('\n  Total mission delta-V budget summary:\n');
fprintf('    JOI capture burn:            %.4f km/s  (chemical, impulsive)\n', dv_JOI_kms);
fprintf('    Phase 1 EDT pump-down:       0.0000 km/s  (propellantless)\n');
fprintf('    Phase 2 manifold steering:   see Section 9 output\n');
fprintf('    Io orbit insertion:          TBD  (Phase B)\n');
fprintf('    Total chemical delta-V:      %.4f km/s\n', dv_JOI_kms);
fprintf('    EDT eliminates post-JOI dV:  saves ~%.0f kg bipropellant\n', m_prop_bipr_joi);
fprintf('\n── JOI baseline complete
──────────────────────────────────────────────────────────\n');
% ── Export to workspace
──────────────────────────────────────────────────────────

assignin('base', 'dv_JOI_kms',     dv_JOI_kms);
assignin('base', 'dv_JOI_ms',      dv_JOI_ms);
assignin('base', 'v_hyp_joi',      v_hyp_joi);
assignin('base', 'v_cap_joi',      v_cap_joi);
assignin('base', 'a_cap_joi',      a_cap_joi);
assignin('base', 'mu_J_joi',       mu_J_joi);
assignin('base', 'm_prop_bipr_joi', m_prop_bipr_joi);
assignin('base', 'm_wet_bipr_joi',  m_wet_bipr_joi);
assignin('base', 'm_prop_lox_joi',  m_prop_lox_joi);
assignin('base', 'm_prop_nep_lo',   m_prop_nep_lo);
assignin('base', 'm_prop_nep_hi',   m_prop_nep_hi);

%% SECTION 3: NORMALIZATION AND FRAME TRANSFORMS
──────────────────────────────────────

% ── CR3BP length and time units
──────────────────────────────────────────────────────────

Lstar = norm(rIo_JCI);            % Length unit [km]: Io's orbital radius → Io sits at x*=1
n     = sqrt(G*(m1+m2) / Lstar^3); % Mean motion [rad/s]: rotating frame angular rate
Tstar = 1/n;                      % Time unit [s]: one radian of rotation
Vstar = Lstar / Tstar;            % Velocity unit [km/s] = n * Lstar
```

```
% ── Mass ratio
──────────────────────────────────────────────────────────────────────────────
mu = m2 / (m1 + m2);              % CR3BP mass parameter [nd]: Io/(Jupiter+Io) ≈ 4.70e-5
% ── Rotating frame basis vectors
──────────────────────────────────────────────────────────────────────
% Built from Io's instantaneous JCI state at t=0
ex    = rIo_JCI / norm(rIo_JCI);   % Jupiter → Io unit vector (+x axis of rotating frame)
ez    = cross(rIo_JCI, vIo_JCI);   % Orbital angular momentum direction (+z, out of plane)
ez    = ez / norm(ez);
ey    = cross(ez, ex);             % Completes right-handed frame (+y, prograde)
R_I2S = [ex, ey, ez];              % Rotation matrix: inertial → synodic
                                   % Columns = rotating basis vectors in inertial coords
                                   % Apply as R_I2S' * r_inertial to get synodic coords
% ── Barycenter offset
──────────────────────────────────────────────────────────────────────────────
rBary_JCI = mu * rIo_JCI;         % Barycenter position relative to Jupiter [km]
vBary_JCI = mu * vIo_JCI;         % Barycenter velocity relative to Jupiter [km/s]
% ── Rotating frame angular velocity
──────────────────────────────────────────────────────────────────
omega_vec_JCI = n * ez;           % [rad/s]: aligned with orbital angular momentum
% ── Test orbit state in normalized synodic frame (Section 5 only) ──────────
% rSC_JCI/vSC_JCI are the ~14 R_J circular test orbit from Section 1
rSC_BI = rSC_JCI - rBary_JCI;
vSC_BI = vSC_JCI - vBary_JCI;
rSC_S  = R_I2S' * rSC_BI;
vSC_S  = R_I2S' * (vSC_BI - cross(omega_vec_JCI, rSC_BI));
                                   % Subtracts ω×r to convert inertial → rotating velocity
x0_nd = rSC_S / Lstar;
v0_nd = vSC_S / Vstar;
Y0    = [x0_nd; v0_nd];           % 6-element normalized state [x y z xd yd zd]^T

%% SECTION 4: ODE PARAMETER STRUCT
──────────────────────────────────────────────────────────────────────
% All values flow from Section 1 master block — nothing hardcoded here.
% p.mu, p.n, p.Lstar, p.Vstar added here after normalization in Section 3.
p.mu    = mu;                     % CR3BP mass ratio [nd]
p.n     = n;                      % Mean motion [rad/s]: rotating frame angular rate
p.Lstar = Lstar;                  % Length unit [km]: dimensional ↔ nd conversion in EOM
p.Vstar = Vstar;                  % Velocity unit [km/s]: dimensional ↔ nd conversion in EOM
```

```
p.R_J_km = R_J_km;          % Jupiter radius [km]: dipole B-field reference radius
p.mSC_kg = mSC_kg;          % Spacecraft mass [kg]: F → acceleration conversion
p.l_m    = l_tether_m;      % Effective tether length [m]: sets EMF and Lorentz force
p.P_max_W = P_max_W;        % Solar array power [W]: power-limited current ceiling
p.I_max_A = I_max_A;        % Physical current cap [A]
p.B_scale = B_scale;        % B-field scaling factor: 1.0 = nominal dipole
p.edt_on  = true;           % EDT thrust flag: overridden locally where needed

%% SECTION 5: INTEGRATE
% ─────────────────────────────────────────────────────────────────────────

% Integrates the test orbit from Section 1 (not the capture orbit).
% Used in Section 15 for Jacobi constant validation only.
odefun      = @(tau, Y) cr3bp_eom_nd_edt(tau, Y, p);
opts        = odeset('RelTol', rtol, 'AbsTol', atol);
[tau, Ysol] = ode113(odefun, [0, tau_final], Y0, opts);
% Unpack solution
r_nd = Ysol(:, 1:3);        % Normalized position history [nd]
v_nd = Ysol(:, 4:6);        % Normalized velocity history [nd]

%% SECTION 6: LAGRANGE POINTS, LYAPUNOV ORBIT, AND MONODROMY
% ──────────────────────────

% ── Global figure style constants
% ─────────────────────────────────────────────────────────────

% Defined here unconditionally so all plotting sections (7 onward) can use
% them regardless of whether the Lyapunov orbit converges.
FIG_BG   = 'k';
AX_BG    = 'k';
AX_COL   = 'w';
GRID_COL = [0.25 0.25 0.25];
FONT_SM  = 9;
FONT_MD  = 11;
FONT_LG  = 13;
% ── Lagrange points
% ──────────────────────────────────────────────────────────────────────

Lpts = compute_lagrange_points(mu);
xL1  = Lpts(1, 1);          % L1 x-coordinate [nd]
xL2  = Lpts(2, 1);          % L2 x-coordinate [nd]
% ── Lyapunov orbit via amplitude continuation
% ──────────────────────────────────────────

% Goal: find a true periodic planar orbit around L1 for use as the
```

```
% monodromy reference and manifold seed orbit.
%
% Method: single-shooter differential corrector exploiting x-axis symmetry.
%   Start:  [x0, 0, 0, 0, yd0, 0]  — on x-axis, moving in +y
%   Target: xd = 0 at next y=0 crossing (half-period condition)
%   Continuation: step amplitude from 0.005 to 0.030 nd, warm-starting
%   each step from the previous converged half-period.
tol_DC   = 1e-10;              % Differential corrector convergence tolerance [nd]
max_iter = 200;                % Maximum Newton iterations per amplitude step
% Pre-compute linearized half-period for warm-starting first step
r1L_init   = abs(xL1 + mu);
r2L_init   = abs(xL1 - 1 + mu);
Uxx_init   = 1 - (1-mu)*(1/r1L_init^3 - 3*(xL1+mu)^2/r1L_init^5) ...
           - mu*(1/r2L_init^3 - 3*(xL1-1+mu)^2/r2L_init^5);
Uyy_init   = 1 - (1-mu)/r1L_init^3 - mu/r2L_init^3;
b_init     = 2 - Uxx_init - Uyy_init;
c_init     = Uxx_init * Uyy_init - 1;
lsq_init   = (-b_init - sqrt(b_init^2 - 4*c_init)) / 2;
omega_init = sqrt(-lsq_init);
T_lyap_current = pi / omega_init;   % Linearized half-period [nd]
A_x_steps = [0.005, 0.010, 0.015, 0.020, 0.025, 0.030];
Y_lyap    = [];
T_lyap    = 0;
converged = false;
fprintf('\n── Lyapunov Orbit Amplitude Continuation ──────────────────────────\n');
for step = 1:length(A_x_steps)
  A_x_step = A_x_steps(step);
  fprintf('  A_x = %.3f nd:  ', A_x_step);
  if step == 1
     [Y_try, T_try, conv] = differential_corrector_lyapunov(...
        xL1, A_x_step, mu, tol_DC, max_iter, p);
  else
     [Y_try, T_try, conv] = differential_corrector_lyapunov(...
        xL1, A_x_step, mu, tol_DC, max_iter, p, T_lyap_current);
  end
  if conv
     fprintf('converged  x0=%.8f  T_full=%.6f nd\n', Y_try(1), 2*T_try);
     Y_lyap        = Y_try;
     T_lyap        = T_try;
     T_lyap_current = T_try;
```

```
        converged      = true;
    else
        fprintf('FAILED — using last converged amplitude A_x=%.3f\n', ...
                A_x_steps(max(step-1,1)));
        break
    end
end
if converged
    fprintf('  Final: x0=%.8f  T_full=%.6f nd  (%.4f days)\n', ...
            Y_lyap(1), 2*T_lyap, 2*T_lyap/n/86400);
end
% ── Monodromy matrix
% ─────────────────────────────────────────────────────────────
% STM integrated over exactly one full Lyapunov period.
% Eigenvalue pairs expected:
%   λ, 1/λ  — hyperbolic pair (stable/unstable manifold directions)
%   1, 1    — trivial pair (along orbit and energy)
%   e^±iθ   — center pair (out-of-plane, not used in planar analysis)

if converged
    Phi0  = eye(6);
    Ystm0 = [Y_lyap; Phi0(:)];
    odefun_stm = @(tau, Y) cr3bp_variational_nd(tau, Y, p);
    opts_stm   = odeset('RelTol', rtol, 'AbsTol', atol);
    [~, Ystm_sol] = ode113(odefun_stm, [0, 2*T_lyap], Ystm0, opts_stm);
    Yend = Ystm_sol(end, :)';
    Phi  = reshape(Yend(7:42), 6, 6);
    [evecs, evals] = eig(Phi);
    lam = diag(evals);
    unstable_idx = find(abs(lam) > 1 + 1e-6);
    stable_idx   = find(abs(lam) < 1 - 1e-6);
    Wu_dirs = [];
    Ws_dirs = [];
    for k = unstable_idx.'
        v = real(evecs(:,k));
        Wu_dirs = [Wu_dirs, v / norm(v)]; %#ok<AGROW>
    end
    for k = stable_idx.'
        v = real(evecs(:,k));
        Ws_dirs = [Ws_dirs, v / norm(v)]; %#ok<AGROW>
```

```
  end
  fprintf('\n── Monodromy Eigenvalues
──────────────────────────────────────────────────────\n');
  for i = 1:numel(lam)
     fprintf('  λ(%d) = %+.6e %+.6ei   |λ|= %.6f\n', ...
           i, real(lam(i)), imag(lam(i)), abs(lam(i)));
  end
  fprintf('  det(Phi) = %.8f  (should be 1.0 for Hamiltonian system)\n', ...
        real(prod(lam)));
else
  fprintf('\nSkipping monodromy — Lyapunov orbit not converged.\n');
  Phi    = eye(6);
  lam    = ones(6,1);
  evecs  = eye(6);
  Wu_dirs = [];
  Ws_dirs = [];
end

%% SECTION 7: MANIFOLD TUBE PROPAGATION
──────────────────────────────────────

% Goal: generate the stable (Ws) and unstable (Wu) manifold tubes
%       by perturbing points along the Lyapunov orbit in the manifold
%       directions and integrating forward/backward in time.
%
% Method:
%   1. Integrate the Lyapunov orbit for one full period → get N points
%   2. At each point, perturb state by ±ε along Wu or Ws eigenvector
%   3. Integrate perturbed states:
%        Wu: forward in time  → unstable tube (spacecraft leaves L1 region)
%        Ws: backward in time → stable tube   (spacecraft arrives at L1 region)
%   4. Plot the resulting tube arms
%
% Physical meaning:
%   Ws (stable, cyan):   trajectories that APPROACH the Lyapunov orbit
%   Wu (unstable, red):  trajectories that DEPART the Lyapunov orbit
if ~converged
  fprintf('\nSkipping manifold propagation — Lyapunov orbit not converged.\n');
else
fprintf('\n── Manifold Tube Propagation
──────────────────────────────────────────────────────\n');
```

```
% ── Step 1: Integrate Lyapunov orbit for one full period ──────────────────
N_pts   = 50;          % Seed points around orbit — more = denser tubes
eps_man = 1e-6;        % Perturbation size [nd]
T_man   = 4.0 * pi;    % Integration time per arm [nd] (~2 Io periods)
opts_man = odeset('RelTol', 1e-8, 'AbsTol', 1e-10);
Phi0_vec = reshape(eye(6), 36, 1);
[~, Y_orb] = ode113(...
  @(t,Y) cr3bp_variational_nd(t, Y, p), ...
  linspace(0, 2*T_lyap, N_pts+1), ...
  [Y_lyap; Phi0_vec], opts_man);
Y_orb = Y_orb(1:end-1, 1:6);   % [N_pts × 6], drop repeated endpoint
fprintf('  Lyapunov orbit sampled at %d points\n', N_pts);
fprintf('  Perturbation size ε = %.1e nd\n', eps_man);
fprintf('  Integration time per arm = %.2f nd (%.2f Io periods)\n', ...
      T_man, T_man/(2*pi));

% ── Step 2 & 3: Propagate manifold arms
──────────────────────────────────────────

Wu_plus  = cell(N_pts, 1);
Wu_minus = cell(N_pts, 1);
Ws_plus  = cell(N_pts, 1);
Ws_minus = cell(N_pts, 1);
% Natural CR3BP manifolds — EDT off so we see unperturbed geometry
p_man       = p;
p_man.edt_on = false;
odefun_man   = @(t,Y) cr3bp_eom_nd_edt(t, Y, p_man);
for k = 1:N_pts
  Yk = Y_orb(k, :)';
  Y_wu_p = Yk + eps_man * Wu_dirs(:,1);
  Y_wu_m = Yk - eps_man * Wu_dirs(:,1);
  Y_ws_p = Yk + eps_man * Ws_dirs(:,1);
  Y_ws_m = Yk - eps_man * Ws_dirs(:,1);
  [~, traj] = ode113(odefun_man, [0,  T_man], Y_wu_p, opts_man);
  Wu_plus{k}  = traj(:, 1:2);
  [~, traj] = ode113(odefun_man, [0,  T_man], Y_wu_m, opts_man);
  Wu_minus{k} = traj(:, 1:2);
  [~, traj] = ode113(odefun_man, [0, -T_man], Y_ws_p, opts_man);
  Ws_plus{k}  = traj(:, 1:2);
  [~, traj] = ode113(odefun_man, [0, -T_man], Y_ws_m, opts_man);
```

```
    Ws_minus{k} = traj(:, 1:2);
end
fprintf('  Manifold arms propagated: %d Wu + %d Ws trajectories\n', 2*N_pts, 2*N_pts);

% ── Step 4: Plot manifold tubes — Figure 1 ──────────────────────────
figure('Color',FIG_BG, 'Name','Manifold Tubes — CR3BP Synodic Frame');
hold on; grid on; box on;
set(gca, 'Color',AX_BG, 'XColor',AX_COL, 'YColor',AX_COL, ...
      'GridColor',GRID_COL, 'FontSize',FONT_SM);
Y_orb_full = [Y_orb; Y_orb(1,:)];
plot(Y_orb_full(:,1), Y_orb_full(:,2), 'y-', 'LineWidth', 3.5, ...
   'DisplayName', 'Lyapunov orbit');
for k = 1:N_pts
   h1 = plot(Wu_plus{k}(:,1),  Wu_plus{k}(:,2),  'r-', 'LineWidth', 1.2);
   h2 = plot(Wu_minus{k}(:,1), Wu_minus{k}(:,2), 'r-', 'LineWidth', 1.2);
   if k == 1
      set(h1, 'DisplayName', 'Unstable manifold W^u');
      set(h2, 'HandleVisibility', 'off');
   else
      set(h1, 'HandleVisibility', 'off');
      set(h2, 'HandleVisibility', 'off');
   end
end
for k = 1:N_pts
   h3 = plot(Ws_plus{k}(:,1),  Ws_plus{k}(:,2),  'c-', 'LineWidth', 1.2);
   h4 = plot(Ws_minus{k}(:,1), Ws_minus{k}(:,2), 'c-', 'LineWidth', 1.2);
   if k == 1
      set(h3, 'DisplayName', 'Stable manifold W^s');
      set(h4, 'HandleVisibility', 'off');
   else
      set(h3, 'HandleVisibility', 'off');
      set(h4, 'HandleVisibility', 'off');
   end
end
plot(-mu,  0, 'o', 'Color','w', 'MarkerFaceColor',[0.9 0.6 0.2], ...
   'MarkerSize',14, 'DisplayName','Jupiter');
plot(1-mu, 0, 'o', 'Color','w', 'MarkerFaceColor',[0.9 0.8 0.3], ...
   'MarkerSize', 9, 'DisplayName','Io');
plot(xL1, 0, '+', 'Color','yellow', 'MarkerSize',14, 'LineWidth',2.5, ...
```

```
    'DisplayName','L_1, L_2');
plot(xL2, 0, '+', 'Color','yellow', 'MarkerSize',14, 'LineWidth',2.5, ...
    'HandleVisibility','off');
text(xL1 - 0.05, 0.07, 'L_1', 'Color','yellow', 'FontSize',FONT_SM+1);
text(xL2 + 0.01, 0.07, 'L_2', 'Color','yellow', 'FontSize',FONT_SM+1);
text(xL1, 0.11, 'zoom \downarrow for neck structure', ...
    'Color','yellow', 'FontSize',FONT_SM, 'HorizontalAlignment','center');
th = linspace(0, 2*pi, 300);
plot(cos(th), sin(th), '--', 'Color',[0.8 0.6 0.1], 'LineWidth',1.2, ...
    'DisplayName','Io orbit (r^* = 1 nd)');
xlabel('x^*  (normalized, synodic)  [1 nd = 421,800 km]', ...
     'Color',AX_COL, 'FontSize',FONT_MD);
ylabel('y^*  (normalized, synodic)  [1 nd = 421,800 km]', ...
     'Color',AX_COL, 'FontSize',FONT_MD);
title({'CR3BP Invariant Manifold Tubes — Jupiter-Io L_1', ...
     'Cyan: stable W^s (entry)  |  Red: unstable W^u (exit)'}, ...
    'Color',AX_COL, 'FontSize',FONT_LG);
legend('show', 'TextColor',AX_COL, 'Color',[0.12 0.12 0.12], ...
     'Location','best', 'FontSize',FONT_SM);
axis equal;
xlim([-1.5, 1.5]);
ylim([-1.2, 1.2]);
text(-1.4, -1.1, ...
    sprintf('1 nd = %.0f km  |  Jupiter-Io CR3BP synodic frame', Lstar), ...
    'Color',[0.6 0.6 0.6], 'FontSize',FONT_SM);

% ── Step 5: Plot Lyapunov orbit zoom — Figure 2 ──────────────────────
figure('Color',FIG_BG, 'Name','Lyapunov Orbit Zoom');
hold on; grid on; box on;
set(gca, 'Color',AX_BG, 'XColor',AX_COL, 'YColor',AX_COL, ...
       'GridColor',GRID_COL, 'FontSize',FONT_SM);
plot(Y_orb_full(:,1), Y_orb_full(:,2), 'y-', 'LineWidth',3.5);
plot(xL1, 0, '+', 'Color','yellow', 'MarkerSize',14, 'LineWidth',2.5);
text(xL1+0.001, -0.002, 'L_1', 'Color','yellow', 'FontSize',FONT_SM);
xlabel('x^*  [nd]', 'Color',AX_COL, 'FontSize',FONT_MD);
ylabel('y^*  [nd]', 'Color',AX_COL, 'FontSize',FONT_MD);
title('Lyapunov Orbit — Zoomed View (L_1 Reference Orbit)', ...
    'Color',AX_COL, 'FontSize',FONT_MD);
% Dynamic amplitude annotation — computed from actual converged orbit
```

```
% True amplitude: maximum x-deviation from L1 over the full orbit
% Y_orb contains N_pts states sampled around the converged Lyapunov orbit
A_x_actual = max(abs(Y_orb(:,1) - xL1));
text(xL1 - 0.019, 0.017, ...
   sprintf('A_x = %.3f nd  (~%.0f km)', A_x_actual, A_x_actual * Lstar), ...
   'Color','w', 'FontSize',FONT_SM);
text(xL1 - 0.019, 0.015, ...
   sprintf('T = %.4f nd  (%.4f days)', 2*T_lyap, 2*T_lyap/n/86400), ...
   'Color','w', 'FontSize',FONT_SM);
text(xL1 - 0.019, 0.013, ...
   sprintf('\\lambda_{max} = %.0f  (instability index)', max(abs(lam))), ...
   'Color','w', 'FontSize',FONT_SM);
zoom_r = 0.02;
xlim([xL1 - zoom_r, xL1 + zoom_r]);
ylim([-zoom_r, zoom_r]);
axis equal;
% Export
assignin('base', 'Wu_plus',    Wu_plus);
assignin('base', 'Wu_minus',   Wu_minus);
assignin('base', 'Ws_plus',    Ws_plus);
assignin('base', 'Ws_minus',   Ws_minus);
assignin('base', 'Y_orb',      Y_orb);
assignin('base', 'Y_orb_full', Y_orb_full);
assignin('base', 'T_lyap',     T_lyap);
fprintf('  Manifold tube plots complete (Figure 1: full view, Figure 2: Lyapunov zoom).\n');
end  % if converged

%% SECTION 8: POINCARÉ MAP AND EDT PUMP-DOWN
% ------------------------------------------------------------------------

% Goal: find where the stable manifold Ws of the L1 Lyapunov orbit
%       intersects with the capture trajectory on a surface of section.
%       That intersection is the heteroclinic connection — the hop point.
%
% Surface of section: y = 0, xdot < 0 (downward x-axis crossing)
%   Standard choice for Jupiter-moon systems (Lo, Ross, Koon et al.)
%
% Ws crossings (cyan):   stable manifold arms crossing y=0
% Capture crossings (magenta): post-pump-down trajectory crossing y=0
% Overlap = heteroclinic connection candidate
% Gap     = Phase 2 EDT steering requirement
```

```
if ~converged
  fprintf('\nSkipping Poincare map — Lyapunov orbit not converged.\n');
else
fprintf('\n── Poincare Map (y=0 section)
──────────────────────────────────────────────\n');
% ── Construct JOI capture state
──────────────────────────────────────────────────────────────────

% Post-JOI elliptical orbit — all geometry from master block via Section 2.
% r_p_joi, r_a_joi, a_cap_joi, v_cap_joi, mu_J_joi all set in Section 2.
T_orbit_s  = 2*pi * sqrt(a_cap_joi^3 / mu_J_joi);   % Orbital period [s]
T_orbit_nd = T_orbit_s * n;                          % Orbital period [nd]
fprintf('  Capture ellipse SMA    = %.0f km (%.1f R_J)\n', ...
     a_cap_joi, a_cap_joi/R_J_km);
fprintf('  Capture orbit period   = %.2f nd = %.2f days\n', ...
     T_orbit_nd, T_orbit_s/86400);
T_pump = 500 * T_orbit_nd;    % Max pump-down: 500 capture orbits
% Arrival angle — theta=0 places perijove on +x axis
theta_arr = 0;
R_rot = [cosd(theta_arr) -sind(theta_arr) 0;
      sind(theta_arr)  cosd(theta_arr) 0;
      0              0              1];
rSC_cap_JCI = R_rot * [r_p_joi; 0; 0];
vSC_cap_JCI = R_rot * [0; v_cap_joi; 0];

% Convert to normalized synodic frame
rSC_cap_BI = rSC_cap_JCI;
vSC_cap_BI = vSC_cap_JCI - vBary_JCI;
rSC_cap_S  = R_I2S' * rSC_cap_BI;
vSC_cap_S  = R_I2S' * (vSC_cap_BI - cross(omega_vec_JCI, rSC_cap_BI));
x0_cap     = rSC_cap_S / Lstar;
v0_cap     = vSC_cap_S / Vstar;
Y0_cap     = [x0_cap; v0_cap];

% Jacobi constants
C_cap  = compute_jacobi(Y0_cap, mu);
C_lyap = compute_jacobi(Y_lyap, mu);
r1L1 = abs(xL1+mu);  r2L1 = abs(xL1-1+mu);
C_L1 = 2*(0.5*xL1^2 + (1-mu)/r1L1 + mu/r2L1);
r1L2 = abs(xL2+mu);  r2L2 = abs(xL2-1+mu);
C_L2 = 2*(0.5*xL2^2 + (1-mu)/r1L2 + mu/r2L2);
```

```
fprintf('  Lyapunov orbit Jacobi: C_lyap = %.6f\n', C_lyap);
fprintf('  Target for manifold connection: C < %.6f\n', C_lyap + 0.05);
fprintf('  Capture state Jacobi:  C_cap = %.6f\n', C_cap);
fprintf('  L1 neck opens at:      C_L1  = %.6f\n', C_L1);
fprintf('  L2 neck opens at:      C_L2  = %.6f\n', C_L2);
if C_cap < C_L1
  fprintf('  ✓ C_cap < C_L1: L1 neck is OPEN — capture can transit\n');
else
  fprintf('  ✗ C_cap > C_L1: L1 neck is CLOSED — running EDT pump-down\n');
end

% ── Step 1: Collect Ws crossings from stable manifold arms ──────────────
Ws_poincare_x  = [];
Ws_poincare_xd = [];
opts_poin = odeset('RelTol', 1e-11, 'AbsTol', 1e-13, ...
                   'Events', @poincare_section_event);
T_poin       = 20 * pi;
p_poin       = p;
p_poin.edt_on = false;
odefun_poin   = @(t,Y) cr3bp_eom_nd_edt(t, Y, p_poin);
for k = 1:N_pts
  Yk = Y_orb(k, :)';
  for sign_eps = [+1, -1]
     Y_seed = Yk + sign_eps * eps_man * Ws_dirs(:,1);
     [~, ~, ~, Ye, ~] = ode113(odefun_poin, [0, -T_poin], Y_seed, opts_poin);
     if ~isempty(Ye)
        for j = 1:size(Ye, 1)
           if Ye(j, 4) < 0
              Ws_poincare_x  = [Ws_poincare_x;  Ye(j, 1)]; %#ok<AGROW>
              Ws_poincare_xd = [Ws_poincare_xd; Ye(j, 4)]; %#ok<AGROW>
           end
        end
     end
  end
end
fprintf('  Ws crossings collected: %d points\n', length(Ws_poincare_x));

% ── Step 2: EDT pump-down until neck opens ──────────────────────────────

fprintf('\n  EDT pump-down: integrating until C < C_L1...\n');
```

```
fprintf('  %-12s  %-12s  %-12s\n', 'Time [nd]', 'Io periods', 'C value');
fprintf('  %-12s  %-12s  %-12s\n', '---------', '----------', '-------');
fprintf('  %-12.4f  %-12.4f  %-12.6f\n', 0, 0, C_cap);
p_pump       = p;
p_pump.edt_on = true;
odefun_pump   = @(t,Y) cr3bp_eom_nd_edt(t, Y, p_pump);
opts_pump     = odeset('RelTol', 1e-10, 'AbsTol', 1e-12);
Y_current     = Y0_cap;
t_current     = 0;
C_current     = C_cap;
neck_open     = false;
T_pump_actual = 0;
C_pump_end    = C_cap;

% Dynamic pump-down table — built orbit by orbit, consumed by Section 11
pump_table_t     = 0;
pump_table_C_dyn = C_cap;
while t_current < T_pump
  [~, Y_seg] = ode113(odefun_pump, ...
     [t_current, t_current + T_orbit_nd], Y_current, opts_pump);
  Y_current = Y_seg(end, :)';
  t_current = t_current + T_orbit_nd;
  C_current = compute_jacobi(Y_current, mu);
  fprintf('  %-12.4f  %-12.4f  %-12.6f\n', ...
        t_current, t_current/(2*pi), C_current);
  pump_table_t     = [pump_table_t;     t_current];    %#ok<AGROW>
  pump_table_C_dyn = [pump_table_C_dyn; C_current];    %#ok<AGROW>
  if C_current < C_L1 % switch to C_lyap + 0.05 if testing partial pump-down; switch to C_L1 if
testing full pump-down
     neck_open     = true;
     T_pump_actual = t_current;
     C_pump_end    = C_current;
     fprintf('\n  ✓ L1 neck opened at t = %.4f nd (%.2f Io periods)\n', ...
           t_current, t_current/(2*pi));
     fprintf('    C = %.6f  |  C_L1 = %.6f  |  C_lyap = %.6f\n', ...
           C_current, C_L1, C_lyap);
     fprintf('    Pump-down: %.2f Io periods = %.2f days\n', ...
           T_pump_actual/(2*pi), T_pump_actual/n/86400);
     break
  end
```

```
end
if ~neck_open
  fprintf('  ✗ Neck did not open within %.0f capture orbits\n', ...
        T_pump/T_orbit_nd);
  fprintf('    Consider larger tether, higher power, or longer integration\n');
end

% ── Step 3: Poincaré map from post-pump-down state ──────────────────────
cap_poincare_x  = [];
cap_poincare_xd = [];
if neck_open
  fprintf('\n  Collecting capture crossings from post-pump-down state...\n');
  T_cap       = 2000 * pi;
  p_cap       = p;
  p_cap.edt_on = true;
  odefun_cap   = @(t,Y) cr3bp_eom_nd_edt(t, Y, p_cap);
  [~, ~, ~, Ye_cap, ~] = ode113(odefun_cap, [0, T_cap], Y_current, opts_poin);
  if ~isempty(Ye_cap)
     for j = 1:size(Ye_cap, 1)
        if Ye_cap(j, 4) < 0
           cap_poincare_x  = [cap_poincare_x;  Ye_cap(j,1)]; %#ok<AGROW>
           cap_poincare_xd = [cap_poincare_xd; Ye_cap(j,4)]; %#ok<AGROW>
        end
     end
  end
  fprintf('  Capture crossings (post pump-down): %d points\n', ...
        length(cap_poincare_x));
else
  fprintf('  Skipping capture crossings — neck never opened.\n');
end

% ── Step 4: Plot Poincaré map — Figure 3 ────────────────────────────────
fig_poincare = figure('Color',FIG_BG, 'Name','Poincare Map — y=0 Section');
hold on; grid on; box on;
set(gca, 'Color',AX_BG, 'XColor',AX_COL, 'YColor',AX_COL, ...
     'GridColor',GRID_COL, 'FontSize',FONT_SM);
if ~isempty(Ws_poincare_x)
  scatter(Ws_poincare_x, Ws_poincare_xd, 18, 'c', 'filled', ...
```

```
        'DisplayName', 'W^s crossings (stable manifold)');
end
if ~isempty(cap_poincare_x)
  scatter(cap_poincare_x, cap_poincare_xd, 22, 'm', 'filled', ...
        'DisplayName', 'Capture trajectory (post pump-down)');
end
xline(xL1, '--', 'Color','yellow', 'LineWidth',1.2, ...
   'Label',sprintf('L_1 (x*=%.4f)',xL1), 'LabelColor','yellow', ...
   'DisplayName',sprintf('L_1  (x* = %.4f nd)', xL1));
xlabel(sprintf('x^*  at y=0 crossing  [nd,  1 nd = %.0f km]', Lstar), ...
    'Color',AX_COL, 'FontSize',FONT_MD);
ylabel(sprintf('xdot^*  at y=0 crossing  [nd,  1 nd = %.2f km/s]', Vstar), ...
    'Color',AX_COL, 'FontSize',FONT_MD);
title({'Poincare Map — Surface of Section  y=0,  xdot < 0', ...
    'Cyan: W^s of L_1 Lyapunov orbit  |  Magenta: post-pump-down capture trajectory', ...
    'Overlap = heteroclinic connection candidate  |  Gap = Phase 2 EDT steering requirement'}, ...
   'Color',AX_COL, 'FontSize',FONT_SM, 'Interpreter','none');
legend('show', 'TextColor',AX_COL, 'Color',[0.12 0.12 0.12], ...
    'Location','best', 'FontSize',FONT_SM);

% ── Step 5: Closest approach in L1 region ──────────────────────────
x_window = 0.15;
Ws_near       = abs(Ws_poincare_x - xL1) < x_window;
Ws_x_filtered  = Ws_poincare_x(Ws_near);
Ws_xd_filtered = Ws_poincare_xd(Ws_near);
cap_near      = abs(cap_poincare_x - xL1) < x_window;
cap_x_filtered  = cap_poincare_x(cap_near);
cap_xd_filtered = cap_poincare_xd(cap_near);
fprintf('  Ws crossings near L1:      %d of %d total\n', ...
     sum(Ws_near), length(Ws_poincare_x));
fprintf('  Capture crossings near L1: %d of %d total\n', ...
     sum(cap_near), length(cap_poincare_x));
min_dist   = Inf;
best_x_ws  = NaN;  best_xd_ws  = NaN;
best_x_cap  = NaN;  best_xd_cap = NaN;
if ~isempty(Ws_x_filtered) && ~isempty(cap_x_filtered)
  for i = 1:length(Ws_x_filtered)
     for j = 1:length(cap_x_filtered)
        dx   = Ws_x_filtered(i)  - cap_x_filtered(j);
```

```
        dxd  = Ws_xd_filtered(i) - cap_xd_filtered(j);
        dist = sqrt(dx^2 + dxd^2);
        if dist < min_dist
            min_dist    = dist;
            best_x_ws   = Ws_x_filtered(i);
            best_xd_ws  = Ws_xd_filtered(i);
            best_x_cap  = cap_x_filtered(j);
            best_xd_cap = cap_xd_filtered(j);
        end
    end
  end
  fprintf('\n  Closest approach (L1 region only):\n');
  fprintf('    Ws point:  x*=%.6f  xdot*=%.6f\n', best_x_ws,  best_xd_ws);
  fprintf('    Cap point: x*=%.6f  xdot*=%.6f\n', best_x_cap, best_xd_cap);
  fprintf('    Gap in (x*, xdot*) space: %.4e nd\n', min_dist);
  if min_dist < 0.01
     fprintf('  ✓ Candidate heteroclinic connection found!\n');
     plot(best_x_ws,  best_xd_ws,  'c*', 'MarkerSize',14, 'LineWidth',2, ...
         'DisplayName','Best W^s candidate');
     plot(best_x_cap, best_xd_cap, 'm*', 'MarkerSize',14, 'LineWidth',2, ...
         'DisplayName','Best cap candidate');
  elseif min_dist < 0.1
     fprintf('  ~ Close — %.4e nd gap in L1 region\n', min_dist);
  else
     fprintf('  ~ No L1-region intersection found\n');
     fprintf('    Best gap in L1 neighborhood: %.4e nd\n', min_dist);
  end
else
  fprintf('  No crossings found within L1 window (±%.2f nd)\n', x_window);
  fprintf('  Consider increasing T_poin or T_cap\n');
end
% Gap annotation on figure
if ~isinf(min_dist)
  figure(fig_poincare);
  text(-1.5, -0.72, ...
      sprintf('Phase-space gap = %.4f nd  ~  %.0f m/s', ...
            min_dist, min_dist * Vstar * 1e3), ...
      'Color',[0.8 0.8 0.8], 'FontSize',FONT_SM);
  text(-1.5, -0.76, ...
      'Cyan cluster near L_1: stable manifold entry region', ...
```

```
    'Color','c', 'FontSize',FONT_SM, 'Interpreter','none');
end
fprintf('\n── Poincare Map complete
──────────────────────────────────────────────────────────\n');
% ── Export
──────────────────────────────────────────────────────────────────────────────
──────────────
assignin('base', 'Ws_poincare_x',   Ws_poincare_x);
assignin('base', 'Ws_poincare_xd',  Ws_poincare_xd);
assignin('base', 'cap_poincare_x',  cap_poincare_x);
assignin('base', 'cap_poincare_xd', cap_poincare_xd);
assignin('base', 'Y0_cap',          Y0_cap);
assignin('base', 'T_pump_actual',   T_pump_actual);
assignin('base', 'C_pump_end',      C_pump_end);
assignin('base', 'C_lyap',          C_lyap);
assignin('base', 'C_L1',            C_L1);
assignin('base', 'C_L2',            C_L2);
assignin('base', 'C_cap',           C_cap);
assignin('base', 'T_orbit_nd',      T_orbit_nd);
assignin('base', 'min_dist',        min_dist);
assignin('base', 'pump_table_t',    pump_table_t);
assignin('base', 'pump_table_C_dyn', pump_table_C_dyn);
if neck_open
  assignin('base', 'Y0_post_pump', Y_current);
end
end  % if converged

%% SECTION 9: EDT NUDGE QUANTIFICATION
──────────────────────────────────────────────────────────

% Quantifies the EDT steering requirement to close the residual energy gap
% between the pump-down endpoint and the L1 manifold insertion target.
%
%   (1) Residual ΔC and velocity equivalent [m/s]
%   (2) EDT burn parameters at L1 crossing
%   (3) Propellant mass comparison vs conventional propulsion
%   (4) Percentage of total energy gap closed by Phase 1
%   (5) Relative velocity at Io approach

if ~converged
   fprintf('\nSkipping EDT nudge quantification — no converged orbit.\n');
```

```
else
fprintf('\n── EDT Nudge Quantification
──────────────────────────────────────────────────\n');
% ── (1) Residual energy gap
──────────────────────────────────────────────────────────────

delta_C_phase1   = C_cap - C_pump_end;
delta_C_residual = C_pump_end - C_lyap;
delta_C_total    = C_cap - C_lyap;
pct_closed       = 100 * delta_C_phase1   / delta_C_total;
pct_remaining    = 100 * delta_C_residual / delta_C_total;
fprintf('\n  Phase 1 EDT pump-down result:\n');
fprintf('    Duration:             %.0f days\n',       T_pump_actual/n/86400);
fprintf('    Jacobi change:        DeltaC = %.4f nd\n', delta_C_phase1);
fprintf('    C at pump-down end:   %.6f nd\n',         C_pump_end);
fprintf('    C_L1 threshold:       %.6f nd\n',         C_L1);
fprintf('    C_Lyapunov target:    %.6f nd\n',         C_lyap);
fprintf('    Residual energy gap:  DeltaC = %.6f nd\n', delta_C_residual);
fprintf('\n  Energy gap accounting:\n');
fprintf('    Total gap (JOI → manifold): DeltaC = %.4f nd\n', delta_C_total);
fprintf('    Phase 1 closes:           %.1f%% of total gap\n', pct_closed);
fprintf('    Phase 2 steering required:  %.1f%% of total gap\n', pct_remaining);


% ── (2) Convert residual ΔC to velocity equivalent ──────────────────────
% ΔC ≈ 2*v*Δv  →  Δv = ΔC * Vstar² / (2 * v_burn)
% Burn location: Ws crossing nearest to L1 on the Poincaré section
[~, i_ws_near] = min(abs(Ws_poincare_x - xL1));
x_burn  = Ws_poincare_x(i_ws_near);
xd_burn = Ws_poincare_xd(i_ws_near);
R_burn_km = sqrt((x_burn + mu)^2) * Lstar;
% Full inertial speed: v_inertial = v_rotating + ω × r
r_burn_nd  = [x_burn; 0; 0];
v_rot_nd   = [xd_burn; 0; 0];
omega_nd   = [0; 0; 1];
v_iner_nd  = v_rot_nd + cross(omega_nd, r_burn_nd);
v_burn_mps = max(norm(v_iner_nd) * Vstar * 1e3, 100.0);  % floor at 100 m/s
delta_v_residual = (delta_C_residual * Vstar^2 * 1e6) / (2 * v_burn_mps);
fprintf('\n  Residual gap converted to velocity equivalent:\n');
fprintf('    Burn location:    x* = %.6f nd  (%.0f km from Jupiter)\n', ...
    x_burn, R_burn_km);
```

```
fprintf('    Speed at burn:    %.2f m/s\n',   v_burn_mps);
fprintf('    Approx DeltaV:    %.2f m/s\n',   delta_v_residual);
fprintf('    (DeltaV = DeltaC * Vstar^2 / (2 * v_burn))\n');

% ── (3) EDT burn parameters ──────────────────────────────────────────

% Lstar = Io orbital radius by construction, so this calibrates the dipole
% to 1800 nT at Io's distance — consistent with cr3bp_eom_nd_edt
B_eq_T   = 1800e-9 * (Lstar / p.R_J_km)^3;
B_burn_T = B_eq_T  * (p.R_J_km / R_burn_km)^3;
EMF_burn = B_burn_T * p.l_m * v_burn_mps;
if EMF_burn > 1e-10
  I_burn = min(p.P_max_W / EMF_burn, p.I_max_A);
else
  I_burn = 0;
end
F_burn_N    = I_burn * p.l_m * B_burn_T;
a_burn_mps2 = F_burn_N / p.mSC_kg;
if a_burn_mps2 > 1e-12
  t_burn_s    = delta_v_residual / a_burn_mps2;
  t_burn_hrs  = t_burn_s / 3600;
  t_burn_days = t_burn_s / 86400;
  E_burn_kJ   = p.P_max_W * t_burn_s / 1e3;
  E_burn_Wh   = p.P_max_W * t_burn_s / 3600;
else
  t_burn_s    = Inf;
  t_burn_hrs  = Inf;
  t_burn_days = Inf;
  E_burn_kJ   = Inf;
  E_burn_Wh   = Inf;
end
fprintf('\n  EDT burn parameters at L1 crossing:\n');
fprintf('    B-field at burn:    %.2f nT\n',    B_burn_T * 1e9);
fprintf('    Motional EMF:       %.4f V\n',     EMF_burn);
fprintf('    Tether current:     %.4f A\n',     I_burn);
fprintf('    Lorentz force:      %.4e N\n',     F_burn_N);
fprintf('    Acceleration:       %.4e m/s^2\n', a_burn_mps2);
fprintf('\n  To deliver DeltaV = %.2f m/s:\n',  delta_v_residual);
fprintf('    Burn duration:      %.1f s  (%.3f hrs  =  %.3f days)\n', ...
      t_burn_s, t_burn_hrs, t_burn_days);
```

```
fprintf('    Energy cost:        %.2f kJ  (%.2f Wh)\n', E_burn_kJ, E_burn_Wh);

% ── (4) Propellant mass comparison ──────────────────────────────────────
Isp_cold = 60;
Isp_bipr = 220;
g0_mps2  = 9.80665;
m_prop_cold = p.mSC_kg * (1 - exp(-delta_v_residual / (Isp_cold * g0_mps2)));
m_prop_bipr = p.mSC_kg * (1 - exp(-delta_v_residual / (Isp_bipr * g0_mps2)));
t_cold_s    = (p.mSC_kg * delta_v_residual / 1000) / 0.5;
t_bipr_s    = (p.mSC_kg * delta_v_residual / 1000) / 4.0;
fprintf('\n  Propellant comparison for same DeltaV:\n');
fprintf('    Cold gas  (Isp=%ds):    %.4f kg,  ~%.1f s burn\n', ...
      Isp_cold, m_prop_cold, t_cold_s);
fprintf('    Bipropellant (Isp=%ds): %.4f kg,  ~%.1f s burn\n', ...
      Isp_bipr, m_prop_bipr, t_bipr_s);
fprintf('    EDT:                    0.000 kg,   %.1f s burn\n', t_burn_s);
% ── (5) Relative velocity at Io approach ────────────────────────────────
% Safe defaults — overwritten below if crossings are available
v_rel_Io_ms  = NaN;
v_rel_Io_kms = NaN;
v_Io_mps     = NaN;
sync_pct     = NaN;
if neck_open && ~isempty(cap_poincare_x)
  [~, idx_Io]  = min(abs(cap_poincare_x - (1-mu)));
  v_rel_Io_nd  = abs(cap_poincare_xd(idx_Io));
  v_rel_Io_ms  = v_rel_Io_nd * Vstar * 1e3;
  v_rel_Io_kms = v_rel_Io_ms / 1e3;
  v_Io_mps     = sqrt((1-mu) / 1.0^3) * Vstar * 1e3;
  sync_pct     = 100 * (1 - v_rel_Io_ms / v_Io_mps);
  fprintf('\n  Relative velocity at closest Io approach:\n');
  fprintf('    x* at crossing:     %.6f nd  (Io at x*=%.6f)\n', ...
        cap_poincare_x(idx_Io), 1-mu);
  fprintf('    v_rel at Io:        %.2f m/s  (%.4f km/s)\n', ...
        v_rel_Io_ms, v_rel_Io_kms);
  fprintf('    Io orbital speed:   %.2f m/s\n', v_Io_mps);
  fprintf('    Synchronization:    %.1f%%\n',   sync_pct);
  if v_rel_Io_ms < 100
     fprintf('    ✓ Excellent — within 100 m/s of Io orbital speed\n');
```

```
  elseif v_rel_Io_ms < 500
    fprintf('    ~ Good — within 500 m/s, small insertion burn needed\n');
  else
    fprintf('    ~ Moderate — %.0f m/s insertion burn required\n', v_rel_Io_ms);
  end
end
fprintf('\n── EDT nudge quantification complete ──────────────────────────────────\n');
% ── Complete mission delta-V budget
──────────────────────────────────────────────────────────
fprintf('\n── Complete Mission Delta-V Budget
──────────────────────────────────────────\n');
fprintf('    JOI capture burn:          %.4f km/s  (chemical)\n',      dv_JOI_kms);
fprintf('    Phase 1 EDT pump-down:     0.0000 km/s  (propellantless)\n');
fprintf('    Phase 2 manifold steering: %.4f km/s  (EDT)\n',           delta_v_residual/1000);
fprintf('    Io orbit insertion:        TBD  (Phase B)\n');
fprintf('    EDT saves vs bipropellant: ~%.0f kg propellant\n',        m_prop_bipr_joi);
fprintf('──────────────────────────────────────────────────────────────────────────
──────────────\n');
% ── Export
──────────────────────────────────────────────────────────────────────────────
──────────────

assignin('base', 'delta_C_residual', delta_C_residual);
assignin('base', 'delta_v_residual', delta_v_residual);
assignin('base', 'delta_C_phase1',   delta_C_phase1);
assignin('base', 'delta_C_total',    delta_C_total);
assignin('base', 'F_burn_N',         F_burn_N);
assignin('base', 'B_burn_T',         B_burn_T);
assignin('base', 'I_burn',           I_burn);
assignin('base', 't_burn_s',         t_burn_s);
assignin('base', 't_burn_days',      t_burn_days);
assignin('base', 'E_burn_kJ',        E_burn_kJ);
assignin('base', 'pct_closed',       pct_closed);
assignin('base', 'pct_remaining',    pct_remaining);
assignin('base', 'v_rel_Io_ms',      v_rel_Io_ms);
assignin('base', 'v_Io_mps',         v_Io_mps);
assignin('base', 'sync_pct',         sync_pct);
assignin('base', 'm_prop_cold',      m_prop_cold);
assignin('base', 'm_prop_bipr',      m_prop_bipr);
end  % if converged
```

```
%% SECTION 10: ARRIVAL ANGLE SWEEP
% Sweep the initial capture orbit arrival angle to find geometries
% that naturally thread into the L1 stable manifold tube.
% Addresses geometric alignment separately from the energy condition
% (handled by pump-down in Section 8).
%
% Short-circuit: rotates the post-pump-down state from Section 8 rather
% than re-running the full pump-down for each angle. Valid to within ~5%
% because pump-down duration is geometry-insensitive across this range.

%% SECTION 10: ARRIVAL ANGLE SWEEP — FULL 360°
% Stage 1 (run now):   Full 360° coarse sweep at 15° spacing (24 angles)
%   Comparable compute time to the original 11-angle sweep.
%   Goal: confirm 167° is globally optimal, check for surprises elsewhere.
%
% Stage 2 (overnight): Fine sweep at 1° spacing around all promising bands
%   A band is "promising" if coarse gap < 0.15 nd.
%   167° band (155-175°) always included regardless of coarse result.
if ~converged
  fprintf('\nSkipping arrival angle sweep — no converged orbit.\n');
else
T_cap_sw = 2000 * pi;
fprintf('\n── Full 360° Arrival Angle Sweep (Coarse, 15° steps) ──────────\n');

% ── Stage 1: Coarse sweep
theta_coarse = 0:15:345;     % 24 angles
N_coarse     = length(theta_coarse);
sweep_colors = hsv(N_coarse);
fprintf('  Sweeping %d angles (0-345°, 15° spacing)\n', N_coarse);
fprintf('  L1 threshold: C_L1 = %.6f\n\n', C_L1);
fprintf('  %-8s  %-12s  %-10s  %-10s\n', 'θ [deg]','C_initial','Cap near L1','Min dist');
fprintf('  %-8s  %-12s  %-10s  %-10s\n', '-------','---------','-----------','--------');
fig_coarse = figure('Color',FIG_BG,'Name','Full 360° Coarse Sweep — Poincare Map');
hold on; grid on; box on;
set(gca,'Color',AX_BG,'XColor',AX_COL,'YColor',AX_COL,...
  'GridColor',GRID_COL,'FontSize',FONT_SM);
if ~isempty(Ws_poincare_x)
```

```
  scatter(Ws_poincare_x, Ws_poincare_xd, 18, 'c', 'filled', ...
        'DisplayName','W^s (stable manifold)');
end
xline(xL1,'--','Color','yellow','LineWidth',1.2,...
    'Label',sprintf('L_1 (x*=%.4f)',xL1),'LabelColor','yellow',...
    'DisplayName',sprintf('L_1 (x*=%.4f nd)',xL1));
opts_sw = odeset('RelTol',1e-10,'AbsTol',1e-12,'Events',@poincare_section_event);
coarse_min_dist   = NaN(N_coarse, 1);
best_overall_dist = Inf;
best_theta        = NaN;
for ii = 1:N_coarse
  theta_ii = theta_coarse(ii);
  if neck_open
     R_rot_sw = [cosd(theta_ii) -sind(theta_ii) 0;
                 sind(theta_ii)  cosd(theta_ii) 0;
                 0               0              1];
     Y_sw = [R_rot_sw * Y_current(1:3); R_rot_sw * Y_current(4:6)];
  else
     Y_sw = Y_current;
  end
  % Initial Jacobi for reporting
  R_rot_ii  = [cosd(theta_ii) -sind(theta_ii) 0;
               sind(theta_ii)  cosd(theta_ii) 0;
               0               0              1];
  rSC_ii    = R_rot_ii * [r_p_joi; 0; 0];
  vSC_ii    = R_rot_ii * [0; v_cap_joi; 0];
  rSC_BI_ii = rSC_ii - rBary_JCI;
  vSC_BI_ii = vSC_ii - vBary_JCI;
  rSC_S_ii  = R_I2S' * rSC_BI_ii;
  vSC_S_ii  = R_I2S' * (vSC_BI_ii - cross(omega_vec_JCI, rSC_BI_ii));
  C_ii      = compute_jacobi([rSC_S_ii/Lstar; vSC_S_ii/Vstar], mu);
  cap_x_sw  = [];
  cap_xd_sw = [];
  min_dist_sw = Inf;
  if neck_open
     p_cap_sw       = p;
     p_cap_sw.edt_on = true;
     odefun_cap_sw   = @(t,Y) cr3bp_eom_nd_edt(t, Y, p_cap_sw);
     [~,~,~,Ye_sw,~] = ode113(odefun_cap_sw, [0, T_cap_sw], Y_sw, opts_sw);
     if ~isempty(Ye_sw)
```

```
        for jj = 1:size(Ye_sw,1)
            if Ye_sw(jj,4) < 0
                cap_x_sw  = [cap_x_sw;  Ye_sw(jj,1)]; %#ok<AGROW>
                cap_xd_sw = [cap_xd_sw; Ye_sw(jj,4)]; %#ok<AGROW>
            end
        end
    end
    near_ii = abs(cap_x_sw - xL1) < 0.15;
    if any(near_ii) && ~isempty(Ws_poincare_x)
        idx = find(near_ii);
        for i = 1:length(idx)
            for j = 1:length(Ws_poincare_x)
                dx  = cap_x_sw(idx(i))  - Ws_poincare_x(j);
                dxd = cap_xd_sw(idx(i)) - Ws_poincare_xd(j);
                d   = sqrt(dx^2 + dxd^2);
                if d < min_dist_sw, min_dist_sw = d; end
            end
        end
    end
  end
  coarse_min_dist(ii) = min_dist_sw;
  if min_dist_sw < best_overall_dist
    best_overall_dist = min_dist_sw;
    best_theta        = theta_ii;
  end
  fprintf('  %-8.0f  %-12.6f  %-10d  %-10.4f\n', ...
        theta_ii, C_ii, sum(abs(cap_x_sw-xL1)<0.15), min_dist_sw);
  col = sweep_colors(ii,:);
  if ~isempty(cap_x_sw)
    scatter(cap_x_sw, cap_xd_sw, 6, col, 'filled', 'HandleVisibility','off');
  end
  scatter(NaN, NaN, 20, col, 'filled', ...
        'DisplayName',sprintf('\\theta=%d\\circ', theta_ii));
end
% ── Identify promising bands for overnight fine sweep ──────────────────
gap_threshold   = 0.15;    % nd — anything below this gets a fine band
band_half_width = 10;      % degrees each side of promising angle
fine_resolution = 1;       % degrees
% No forced bands — fine sweep is driven purely by coarse results
% Section 14 handles the fine sweep dynamically around best_theta
```

```
forced_bands = [];   % no hardcoded bands
% Add bands around any coarse angles below threshold
auto_bands = [];
for ii = 1:N_coarse
  if coarse_min_dist(ii) < gap_threshold
     lo = theta_coarse(ii) - band_half_width;
     hi = theta_coarse(ii) + band_half_width;
     auto_bands = [auto_bands; lo, hi]; %#ok<AGROW>
  end
end
% Merge overlapping bands
all_bands = [forced_bands; auto_bands];
all_bands = sort(all_bands, 1);
% Build fine angle list from merged bands
theta_fine_overnight = [];
for bb = 1:size(all_bands, 1)
  lo = mod(all_bands(bb,1), 360);
  hi = mod(all_bands(bb,2), 360);
  band_angles = lo:fine_resolution:hi;
  theta_fine_overnight = union(theta_fine_overnight, band_angles);
end
theta_fine_overnight = sort(unique(theta_fine_overnight));
fprintf('\n  Best coarse angle: θ = %.0f°  (gap = %.4f nd)\n', ...
     best_theta, best_overall_dist);
fprintf('\n  Promising bands (gap < %.2f nd):\n', gap_threshold);
for ii = 1:N_coarse
  if coarse_min_dist(ii) < gap_threshold
     fprintf('    θ = %.0f°  →  fine sweep %.0f° to %.0f°\n', ...
          theta_coarse(ii), ...
          theta_coarse(ii) - band_half_width, ...
          theta_coarse(ii) + band_half_width);
  end
end
fprintf('\n  Fine sweep in Section 14 will be dynamically centered on best coarse angle
(%.0f°).\n', ...
     best_theta);
fprintf('  Overnight fine sweep: %d angles at %d° resolution\n', ...
     length(theta_fine_overnight), fine_resolution);
fprintf('  Angles: %s\n', num2str(theta_fine_overnight));
% Finalize coarse figure
```

```
figure(fig_coarse);
xlabel(sprintf('x^*  at y=0 crossing  [nd,  1 nd = %.0f km]', Lstar), ...
    'Color',AX_COL,'FontSize',FONT_MD);
ylabel(sprintf('xdot^*  at y=0 crossing  [nd,  1 nd = %.2f km/s]', Vstar), ...
    'Color',AX_COL,'FontSize',FONT_MD);
title({'Full 360° Arrival Angle Sweep — Coarse (15° steps)', ...
    sprintf('Best: \\theta=%.0f° (gap=%.4f nd)  |  Cyan: W^s stable manifold', ...
        best_theta, best_overall_dist)}, ...
   'Color',AX_COL,'FontSize',FONT_SM,'Interpreter','tex');
legend('show','TextColor',AX_COL,'Color',[0.12 0.12 0.12],...
    'Location','southwest','FontSize',6);
xlim([-2, 4]);
ylim([-0.8, 0.05]);
% Export coarse results
assignin('base','theta_coarse',          theta_coarse);
assignin('base','coarse_min_dist',       coarse_min_dist);
assignin('base','best_theta',            best_theta);
assignin('base','best_overall_dist',     best_overall_dist);
assignin('base','theta_fine_overnight',  theta_fine_overnight);

% ── Stage 2: Overnight fine sweep ──────────────────────────────────────

% Run this section after reviewing the coarse results.
% theta_fine_overnight is built automatically from coarse findings above.
run_overnight = true;   % ← SET TO TRUE BEFORE LEAVING FOR THE NIGHT
if run_overnight
fprintf('\n── Overnight Fine Sweep (%d angles, 1° resolution) ──────────\n', ...
     length(theta_fine_overnight));
fprintf('  %-8s  %-10s  %-10s\n','θ [deg]','Cap near L1','Min dist [nd]');
fprintf('  %-8s  %-10s  %-10s\n','-------','----------','------------');
fine_min_dist    = NaN(length(theta_fine_overnight), 1);
cap_x_fine_all   = cell(length(theta_fine_overnight), 1);
cap_xd_fine_all  = cell(length(theta_fine_overnight), 1);
opts_fine     = odeset('RelTol',1e-10,'AbsTol',1e-12,'Events',@poincare_section_event);
p_fine        = p;
p_fine.edt_on = true;
T_cap_fine    = 2000 * pi;
for ii = 1:length(theta_fine_overnight)
  theta_ii = theta_fine_overnight(ii);
  R_rot_ii = [cosd(theta_ii) -sind(theta_ii) 0;
```

```
            sind(theta_ii)  cosd(theta_ii) 0;
            0               0              1];
    Y_fine_ii = [R_rot_ii * Y_current(1:3); R_rot_ii * Y_current(4:6)];
    odefun_fine = @(t,Y) cr3bp_eom_nd_edt(t, Y, p_fine);
    [~,~,~,Ye_fine,~] = ode113(odefun_fine, [0, T_cap_fine], Y_fine_ii, opts_fine);
    cap_x_ii  = [];
    cap_xd_ii = [];
    if ~isempty(Ye_fine)
        for jj = 1:size(Ye_fine,1)
            if Ye_fine(jj,4) < 0
                cap_x_ii  = [cap_x_ii;  Ye_fine(jj,1)]; %#ok<AGROW>
                cap_xd_ii = [cap_xd_ii; Ye_fine(jj,4)]; %#ok<AGROW>
            end
        end
    end
    cap_x_fine_all{ii}  = cap_x_ii;
    cap_xd_fine_all{ii} = cap_xd_ii;
    near_ii = abs(cap_x_ii - xL1) < 0.15;
    md = Inf;
    if any(near_ii) && ~isempty(Ws_poincare_x)
        idx = find(near_ii);
        for i = 1:length(idx)
            for j = 1:length(Ws_poincare_x)
                dx  = cap_x_ii(idx(i))  - Ws_poincare_x(j);
                dxd = cap_xd_ii(idx(i)) - Ws_poincare_xd(j);
                d   = sqrt(dx^2 + dxd^2);
                if d < md, md = d; end
            end
        end
    end
    fine_min_dist(ii) = md;
    fprintf('  %-8.0f  %-10d  %-10.4f\n', theta_ii, sum(near_ii), md);
end
[best_dist_fine, best_idx_fine] = min(fine_min_dist);
best_theta_fine = theta_fine_overnight(best_idx_fine);
fprintf('\n  Best angle: θ = %.0f°  (gap = %.4f nd)\n', ...
    best_theta_fine, best_dist_fine);

% Fine sweep Poincaré map
fig_fine = figure('Color',FIG_BG,'Name','Overnight Fine Sweep — Poincare Map');
```

```
hold on; grid on; box on;
set(gca,'Color',AX_BG,'XColor',AX_COL,'YColor',AX_COL,...
  'GridColor',GRID_COL,'FontSize',FONT_SM);
if ~isempty(Ws_poincare_x)
  scatter(Ws_poincare_x, Ws_poincare_xd, 18, 'c', 'filled', ...
       'DisplayName','W^s stable manifold');
end
cmap_fine = parula(length(theta_fine_overnight));
for ii = 1:length(theta_fine_overnight)
  if ~isempty(cap_x_fine_all{ii})
     scatter(cap_x_fine_all{ii}, cap_xd_fine_all{ii}, ...
          8, cmap_fine(ii,:), 'filled', 'HandleVisibility','off');
  end
  if fine_min_dist(ii) < 0.05
     scatter(NaN, NaN, 25, cmap_fine(ii,:), 'filled', ...
          'DisplayName',sprintf('\\theta=%d\\circ (gap=%.4f)', ...
                       theta_fine_overnight(ii), fine_min_dist(ii)));
  end
end
xline(xL1,'--','Color','yellow','LineWidth',1.2,...
    'Label',sprintf('L_1 (x*=%.4f)',xL1),'LabelColor','yellow',...
    'DisplayName',sprintf('L_1 (x*=%.4f nd)',xL1));
xlabel(sprintf('x^*  at y=0 crossing  [nd,  1 nd = %.0f km]', Lstar), ...
     'Color',AX_COL,'FontSize',FONT_MD);
ylabel(sprintf('xdot^*  at y=0 crossing  [nd,  1 nd = %.2f km/s]', Vstar), ...
     'Color',AX_COL,'FontSize',FONT_MD);
title({'Overnight Fine Sweep — Poincare Map', ...
     sprintf('Best: \\theta=%.0f° (gap=%.4f nd)  |  Only gaps < 0.05 nd in legend', ...
          best_theta_fine, best_dist_fine)}, ...
    'Color',AX_COL,'FontSize',FONT_SM,'Interpreter','tex');
legend('show','TextColor',AX_COL,'Color',[0.12 0.12 0.12],...
     'Location','southwest','FontSize',FONT_SM);
xlim([-2, 4]);
ylim([-0.8, 0.05]);
assignin('base','fine_min_dist',    fine_min_dist);
assignin('base','best_theta_fine',  best_theta_fine);
assignin('base','best_dist_fine',   best_dist_fine);
assignin('base','cap_x_fine_all',   cap_x_fine_all);
assignin('base','cap_xd_fine_all',  cap_xd_fine_all);
```

```
fprintf('\n── Fine sweep complete
──────────────────────────────────────────────────────────────\n');

% Update best angle from fine sweep results — overrides coarse sweep winner
[best_fine_val, best_fine_idx] = min(fine_min_dist);
if best_fine_val < best_overall_dist
  best_overall_dist = best_fine_val;
  best_theta        = theta_fine_overnight(best_fine_idx);
end
fprintf('  Updated best angle from fine sweep: θ = %.0f°  (gap = %.4f nd)\n', ...
      best_theta, best_overall_dist);

% ── Stage 3: Sub-degree sweep around best candidates ──────────────────────
sub_threshold = 0.02;    % nd — angles below this get 0.5° resolution
sub_half_width = 3;      % degrees each side
sub_resolution = 0.5;
% Find candidates from fine sweep
sub_angles = [];
for ii = 1:length(theta_fine_overnight)
  if fine_min_dist(ii) < sub_threshold
     lo = theta_fine_overnight(ii) - sub_half_width;
     hi = theta_fine_overnight(ii) + sub_half_width;
     sub_angles = union(sub_angles, lo:sub_resolution:hi);
  end
end
sub_angles = sort(unique(sub_angles));
fprintf('\n── Sub-degree Sweep (0.5°, around gaps < %.3f nd) ──────────────\n', ...
      sub_threshold);
fprintf('  %d angles: %s\n', length(sub_angles), num2str(sub_angles));
sub_min_dist   = NaN(length(sub_angles), 1);
sub_x_fine_all = cell(length(sub_angles), 1);
sub_xd_fine_all = cell(length(sub_angles), 1);
for ii = 1:length(sub_angles)
  theta_ii = sub_angles(ii);
  R_rot_ii = [cosd(theta_ii) -sind(theta_ii) 0;
           sind(theta_ii)  cosd(theta_ii) 0;
           0               0              1];
  Y_sub_ii = [R_rot_ii * Y_current(1:3); R_rot_ii * Y_current(4:6)];
  odefun_sub = @(t,Y) cr3bp_eom_nd_edt(t, Y, p_fine);
  [~,~,~,Ye_sub,~] = ode113(odefun_sub, [0, T_cap_fine], Y_sub_ii, opts_fine);
```

```
  cap_x_ii  = [];
  cap_xd_ii = [];
  if ~isempty(Ye_sub)
    for jj = 1:size(Ye_sub,1)
      if Ye_sub(jj,4) < 0
        cap_x_ii  = [cap_x_ii;  Ye_sub(jj,1)]; %#ok<AGROW>
        cap_xd_ii = [cap_xd_ii; Ye_sub(jj,4)]; %#ok<AGROW>
      end
    end
  end
  sub_x_fine_all{ii}  = cap_x_ii;
  sub_xd_fine_all{ii} = cap_xd_ii;
  near_ii = abs(cap_x_ii - xL1) < 0.15;
  md = Inf;
  if any(near_ii) && ~isempty(Ws_poincare_x)
    idx = find(near_ii);
    for i = 1:length(idx)
      for j = 1:length(Ws_poincare_x)
        dx  = cap_x_ii(idx(i))  - Ws_poincare_x(j);
        dxd = cap_xd_ii(idx(i)) - Ws_poincare_xd(j);
        d   = sqrt(dx^2 + dxd^2);
        if d < md, md = d; end
      end
    end
  end
  sub_min_dist(ii) = md;
  fprintf('  θ = %6.1f°   Cap near L1: %-4d  Gap: %.4f nd\n', ...
      theta_ii, sum(near_ii), md);
end
[best_sub_dist, best_sub_idx] = min(sub_min_dist);
best_sub_theta = sub_angles(best_sub_idx);
fprintf('\n  Best sub-degree angle: θ = %.1f°  (gap = %.6f nd)\n', ...
    best_sub_theta, best_sub_dist);
% Update best angle from sub-degree sweep results
if best_sub_dist < best_overall_dist
  best_overall_dist = best_sub_dist;
  best_theta        = best_sub_theta;
end
fprintf('  Final best angle: θ = %.1f°  (gap = %.6f nd)\n', ...
    best_theta, best_overall_dist);
```

```
assignin('base', 'best_theta',        best_theta);
assignin('base', 'best_overall_dist', best_overall_dist);
assignin('base', 'sub_angles',       sub_angles);
assignin('base', 'sub_min_dist',     sub_min_dist);
assignin('base', 'best_sub_theta',   best_sub_theta);
assignin('base', 'best_sub_dist',    best_sub_dist);
save('overnight_sweep_results.mat', ...
    'theta_fine_overnight', 'fine_min_dist', ...
    'sub_angles', 'sub_min_dist', ...
    'best_sub_theta', 'best_sub_dist', ...
    'cap_x_fine_all', 'cap_xd_fine_all');
fprintf('  Results saved to overnight_sweep_results.mat\n');
end  % if run_overnight
end  % if converged

%% SECTION 11: JACOBI ENERGY CORRIDOR PLOT
──────────────────────────────────────────────

% Single figure showing C vs mission time across the full sequence:
%   JOI entry → EDT pump-down → L1 neck opening → manifold hop → Io
% Pump-down curve is built dynamically from Section 8 loop output —
% automatically reflects current r_p_RJ, mSC_kg, l_tether_m, P_max_W.
fprintf('\n── Jacobi Energy Corridor Plot
──────────────────────────────────────────────\n');
% ── Key energy levels
──────────────────────────────────────────────────────────────────────

C_JOI      = C_cap;    % Jacobi at JOI entry [nd]
C_L1_val   = C_L1;     % L1 neck opening threshold [nd]
C_lyap_val = C_lyap;   % Lyapunov orbit target [nd]
% Dynamic pump-down table from Section 8
pump_table_C    = pump_table_C_dyn;
pump_table_days = pump_table_t / n / 86400;
T_pump_days     = T_pump_actual / n / 86400;
% ── Build Figure 5
──────────────────────────────────────────────────────────────────────
─
figure('Color',FIG_BG, 'Name','Jacobi Energy Corridor', ...
     'Position',[100 100 1000 550]);
hold on; grid on; box on;
set(gca, 'Color',AX_BG, 'XColor',AX_COL, 'YColor',AX_COL, ...
       'GridColor',[0.2 0.2 0.2], 'FontSize',FONT_MD);
```

```
% Y-axis bounds — dynamic so annotations never clip
y_ceil  = C_JOI      + 0.10;
y_floor = C_lyap_val - 0.05;
% Shaded forbidden region (C > C_L1 = neck closed)
x_fill = [0, T_pump_days*1.3, T_pump_days*1.3, 0];
y_fill = [C_L1_val, C_L1_val, y_ceil, y_ceil];
fill(x_fill, y_fill, [0.5 0.1 0.1], 'FaceAlpha',0.4, ...
   'EdgeColor','none', 'DisplayName','L_1 neck closed (C > C_{L1})');
% EDT pump-down curve
plot(pump_table_days, pump_table_C, 'c-o', 'LineWidth',2.5, ...
   'MarkerSize',5, 'MarkerFaceColor','c', ...
   'DisplayName','EDT pump-down (C decreasing)');
% Key threshold lines
yline(C_L1_val, '--', 'Color',[1 0.8 0], 'LineWidth',1.8, ...
    'LabelHorizontalAlignment','left', 'LabelVerticalAlignment','bottom', ...
    'Label',sprintf('C_{L1} = %.4f  (neck opens)', C_L1_val), ...
    'LabelColor',[1 0.8 0], 'FontSize',FONT_SM, ...
    'DisplayName','C_{L1} — L1 neck opening threshold');
yline(C_lyap_val, '--', 'Color',[0.5 1 0.5], 'LineWidth',1.5, ...
    'LabelHorizontalAlignment','right', 'LabelVerticalAlignment','top', ...
    'Label',sprintf('C_{Lyapunov} = %.4f  (manifold target)', C_lyap_val), ...
    'LabelColor',[0.5 1 0.5], 'FontSize',FONT_SM, ...
    'DisplayName','C_{Lyapunov} — manifold insertion target');
yline(C_JOI, '--', 'Color',[0.7 0.7 1], 'LineWidth',1.2, ...
    'Label',sprintf('C_{JOI} = %.3f', C_JOI), ...
    'LabelColor',[0.7 0.7 1], 'FontSize',FONT_SM, ...
    'DisplayName',sprintf('C_{JOI} = %.3f — capture entry', C_JOI));
% Pump-down endpoint marker
plot(T_pump_days, C_pump_end, 'y*', 'MarkerSize',18, 'LineWidth',2, ...
   'DisplayName', ...
   sprintf('EDT pump-down ends: t = %.0f days (C = %.4f)', ...
         T_pump_days, C_pump_end));
% Phase 2 schematic continuation line
t_ph2 = linspace(T_pump_days, T_pump_days * 1.12, 20);
C_ph2 = C_pump_end * ones(1, 20);
plot(t_ph2, C_ph2, 'w--', 'LineWidth',1.5, ...
   'DisplayName','Phase 2: manifold transit (schematic)');
% Terminal annotation
text(T_pump_days * 1.04, C_ph2(end) - 0.008, ...
   '\rightarrow Io arrival (Phase B)', ...
```

```
    'Color',[0.8 0.8 0.8], 'FontSize',FONT_MD, 'HorizontalAlignment','left');
% Phase 2 arrow and label
text(T_pump_days, C_pump_end + 0.030, ...
    'Phase 2: Manifold Hop begins here', ...
    'Color','w', 'FontSize',FONT_SM, 'HorizontalAlignment','center');
plot([T_pump_days, T_pump_days], ...
    [C_pump_end + 0.022, C_pump_end + 0.004], ...
    'w-', 'LineWidth',2, 'HandleVisibility','off');
plot(T_pump_days, pump_table_C(end) + 0.004, ...
    'wv', 'MarkerSize',8, 'MarkerFaceColor','w', 'HandleVisibility','off');
% Phase 1 annotations — anchored relative to C_JOI
text(T_pump_days * 0.35, C_JOI - 0.06, 'Phase 1: EDT Pump-Down', ...
    'Color','w', 'FontSize',FONT_MD, 'HorizontalAlignment','center');
text(T_pump_days * 0.35, C_JOI - 0.09, ...
    sprintf('%.0f days  |  \\DeltaC = %.3f nd', T_pump_days, C_JOI - C_pump_end), ...
    'Color',[0.8 0.8 0.8], 'FontSize',FONT_SM, 'HorizontalAlignment','center');
% Clean up orphan legend entries
axChildren = get(gca, 'Children');
for kk = 1:length(axChildren)
   dn = get(axChildren(kk), 'DisplayName');
   if isempty(dn) || strcmp(dn, '') || strcmp(dn, 'data1')
      set(axChildren(kk), 'HandleVisibility','off');
   end
end
% Labels, limits, legend
xlabel('Mission elapsed time  [days]', 'Color',AX_COL, 'FontSize',FONT_MD);
ylabel('Jacobi constant  C  [nd]',     'Color',AX_COL, 'FontSize',FONT_MD);
title({'UNAGI Mission — Jacobi Energy Corridor', ...
     'JOI Capture  \rightarrow  EDT Pump-Down  \rightarrow  L_1 Manifold Transit'}, ...
    'Color',AX_COL, 'FontSize',FONT_LG);
% Dual y-axis — right side in physical units
ax1 = gca;
ax2 = axes('Position',ax1.Position, 'YAxisLocation','right', ...
          'Color','none', 'XColor','none', 'YColor',AX_COL, ...
          'FontSize',FONT_SM);
ax2.YLim = ax1.YLim * Vstar^2;
ylabel(ax2, 'Jacobi constant  C  [km^2/s^2]', 'Color',AX_COL, 'FontSize',FONT_MD);
axes(ax1);
legend('show', 'TextColor',AX_COL, 'Color',[0.1 0.1 0.1], ...
     'Location','northeast', 'FontSize',FONT_SM);
```

```
xlim([0, T_pump_days * 1.18]);
ylim([y_floor, y_ceil]);
fprintf('  Jacobi energy corridor plot complete.\n');
fprintf('  Phase 1 duration:  %.0f days\n',  T_pump_days);
fprintf('  Total DeltaC P1:   %.4f nd\n',    C_JOI - C_pump_end);
fprintf('  C at neck opening: %.6f nd\n',    C_pump_end);
fprintf('  C_L1 threshold:    %.6f nd\n',    C_L1_val);
fprintf('  Remaining gap:     %.6f nd\n',    C_pump_end - C_lyap_val);
%% SECTION 12: POINT A TO POINT B SUMMARY TABLE
% ─────────────────────────────────────────────
if ~converged
  fprintf('\nSkipping summary table — Lyapunov orbit not converged.\n');
else
fprintf('\n');
fprintf('╔═══════════════════════════════════════════════════════════════════════╗\n');
fprintf('║        UNAGI MISSION — POINT A TO POINT B SUMMARY                     ║\n');
fprintf('║        JOI Capture → EDT Pump-Down → L1 Manifold Transit              ║\n');
fprintf('╠═══════════════════════════════════════════════════════════════════════╣\n');
fprintf('║  MISSION PARAMETERS (master block)                                    ║\n');
fprintf('║    Perijove:                %8.1f R_J                                 ║\n', r_p_RJ);
fprintf('║    Apojove:                 %8.1f R_J                                 ║\n', r_a_RJ);
fprintf('║    Tether length:           %8.0f m                                   ║\n', p.l_m);
fprintf('║    Solar array power:       %8.0f W                                   ║\n', p.P_max_W);
fprintf('║    Spacecraft mass:         %8.0f kg                                  ║\n', p.mSC_kg);
fprintf('║    B-field scale:           %8.1f (nominal=1.0)                       ║\n', p.B_scale);
fprintf('╠═══════════════════════════════════════════════════════════════════════╣\n');
fprintf('║  MISSION TIMELINE                                                     ║\n');
fprintf('║    Phase 1 EDT pump-down duration:  %8.0f days                        ║\n', ...
    T_pump_actual/n/86400);
fprintf('║    Capture orbit period:            %8.2f days                        ║\n', ...
    T_orbit_nd/n/86400);
fprintf('║    Number of capture orbits:        %8.0f                             ║\n', ...
    T_pump_actual/T_orbit_nd);
fprintf('╠═══════════════════════════════════════════════════════════════════════╣\n');
fprintf('║  ENERGY ACCOUNTING (Jacobi constant C [nd])                           ║\n');
fprintf('║    C at JOI entry:          %8.6f nd                                  ║\n', C_cap);
```

```
fprintf('║   C at pump-down end:            %8.6f nd           ║\n', C_pump_end);
fprintf('║   C_L1 neck opening threshold:   %8.6f nd           ║\n', C_L1);
fprintf('║   C_Lyapunov manifold target:    %8.6f nd           ║\n', C_lyap);
fprintf('║   Total DeltaC Phase 1:          %8.6f nd  (%.1f%%)  ║\n', ...
    delta_C_phase1, pct_closed);
fprintf('║   Residual DeltaC Phase 2:       %8.6f nd  (%.1f%%)  ║\n', ...
    delta_C_residual, pct_remaining);
fprintf('╠═══════════════════════════════════════════════════════════════╣\n');
fprintf('║  EDT PERFORMANCE AT L1 CROSSING                               ║\n');
fprintf('║   B-field at burn point:         %8.2f nT           ║\n', B_burn_T*1e9);
fprintf('║   Tether current at burn:        %8.4f A            ║\n', I_burn);
fprintf('║   Lorentz force at burn:         %8.4e N            ║\n', F_burn_N);
fprintf('╠═══════════════════════════════════════════════════════════════╣\n');
fprintf('║  PHASE 2 STEERING REQUIREMENT                                 ║\n');
fprintf('║   Residual DeltaV (energy-based): %8.2f m/s         ║\n', delta_v_residual);
fprintf('║   EDT burn duration for DeltaV:   %8.2f days        ║\n', t_burn_days);
fprintf('║   Energy cost:                    %8.2f kJ          ║\n', E_burn_kJ);
fprintf('║   Propellant mass (cold gas):     %8.2f kg          ║\n', m_prop_cold);
fprintf('║   Propellant mass (bipropellant): %8.2f kg          ║\n', m_prop_bipr);
fprintf('║   Propellant mass (EDT):          %8.2f kg          ║\n', 0.0);
fprintf('╠═══════════════════════════════════════════════════════════════╣\n');
fprintf('║  IO ARRIVAL STATE                                             ║\n');
if isinf(best_overall_dist) || best_overall_dist > 0.5
  fprintf('║   Best arrival geometry:         No close connection found  ║\n');
  fprintf('║   Poincare gap (best angle):     %8.4f nd  (>0.5)   ║\n', best_overall_dist);
  fprintf('║   NOTE: Run arrival angle sweep for refined geometry        ║\n');
else
  fprintf('║   Best arrival geometry:         theta = %.0f deg        ║\n', best_theta);
  fprintf('║   Poincare gap (best angle):     %8.4f nd           ║\n', best_overall_dist);
end
if ~isnan(v_rel_Io_ms)
  fprintf('║   Relative velocity at Io:       %8.2f m/s          ║\n', v_rel_Io_ms);
  fprintf('║   Io orbital speed:              %8.2f m/s          ║\n', v_Io_mps);
  fprintf('║   Orbit synchronization:         %8.1f%%            ║\n', sync_pct);
else
  fprintf('║   Relative velocity at Io:       %8s                ║\n', 'N/A');
end
```

```
fprintf('╠═══════════════════════════════════════════════════════════════════════════════╣\n');
fprintf('║  LYAPUNOV ORBIT (L1 REFERENCE)                                                ║\n');
fprintf('║    Orbit amplitude A_x:          %8.4f nd                                     ║\n', max(abs(Y_orb(:,1) - xL1)));
fprintf('║    Full period T_Lyapunov:       %8.6f nd                                     ║\n', 2*T_lyap);
fprintf('║    Full period T_Lyapunov:       %8.4f days                                   ║\n', 2*T_lyap/n/86400);
fprintf('║    Instability index (lambda_max): %8.2f                                      ║\n', max(abs(lam)));
fprintf('╚═══════════════════════════════════════════════════════════════════════════════╝\n');
assignin('base', 'summary_pump_days', T_pump_actual/n/86400);
assignin('base', 'summary_deltaC_p1', delta_C_phase1);
assignin('base', 'summary_deltaV_p2', delta_v_residual);
assignin('base', 'summary_burn_days', t_burn_days);
assignin('base', 'summary_v_rel_Io',  v_rel_Io_ms);
assignin('base', 'summary_sync_pct',  sync_pct);
end  % if converged
%% SECTION 13: SENSITIVITY SWEEP — POWER, TETHER, MASS, B-FIELD
% ─────────────────────────────────────────────────────────────
% Sweeps P_max_W, l_tether_m, mSC_kg, and B_scale simultaneously.
% Records pump-down duration and mean ΔC/orbit for each combination.
%
% B-field scaling: accounts for magnetic field model uncertainty ±20%
% Mass sweep: addresses MEDT configuration mass budget uncertainty
%
% Reference lines at 5yr and 10yr mission lifetime.
% Thesis baseline defined by master block values in Section 1.
if ~converged
  fprintf('\nSkipping sensitivity sweep — Lyapunov orbit not converged.\n');
else
% ── Quick validation before full sweep ──────────────────────────
p_test        = p;
p_test.edt_on  = true;
p_test.P_max_W = 1070;
p_test.l_m     = 10000;
p_test.mSC_kg  = 1200;   % Heavier than baseline — should give longer pump-down
p_test.B_scale = 0.8;    % Weaker field — should also give longer pump-down
fprintf('\n── Validation run (expect > %.0f days) ──────────────────────────\n', ...
    T_pump_actual/n/86400);
Y_val    = Y0_cap;
```

```
t_val    = 0;
neck_val = false;
opts_val = odeset('RelTol',1e-10,'AbsTol',1e-12);
odefun_val = @(t,Y) cr3bp_eom_nd_edt(t, Y, p_test);
while t_val < 800 * T_orbit_nd
  [~, Y_seg] = ode113(odefun_val, [t_val, t_val+T_orbit_nd], Y_val, opts_val);
  Y_val = Y_seg(end,:)';
  t_val = t_val + T_orbit_nd;
  if compute_jacobi(Y_val, mu) < C_lyap + 0.05
     neck_val = true;
     break
  end
end
if neck_val
  T_val_days = t_val / n / 86400;
  fprintf('  Validation result:  %.0f days\n', T_val_days);
  if T_val_days > T_pump_actual/n/86400
     fprintf('  ✓ Correct — heavier mass + weaker B gives longer pump-down\n');
     fprintf('  Proceeding with full sweep...\n\n');
  else
     fprintf('  ✗ Unexpected — check p.B_scale and p.mSC_kg in EOM\n');
     return
  end
else
  fprintf('  ✗ Neck did not open in validation run — check EOM\n');
  return
end
fprintf('\n── Sensitivity Sweep: Power / Tether / Mass / B-field ──────\n');
%P_sweep_W   = [500, 750, 1070, 1500];
%l_sweep_m   = [5000, 9200, 10000, 15000];
%m_sweep_kg  = [800, 1000, 1200, 1500];
%B_sweep_scl = [0.8, 1.0, 1.2];
P_sweep_W   = [500, 750, 1070, 1500, 2000, 3500, 5000, 7500, 10000];
l_sweep_m   = [5000, 9200, 10000, 15000, 18400, 20000, 25000, 27600];
m_sweep_kg  = [800, 1000, 1200, 1500];
B_sweep_scl = [0.8, 1.0, 1.2];
N_P = length(P_sweep_W);
N_l = length(l_sweep_m);
N_m = length(m_sweep_kg);
N_B = length(B_sweep_scl);
```

```
T_5yr  = 5  * 365.25;
T_10yr = 10 * 365.25;
T_30yr = 30 * 365.25;
T_pump_sweep = NaN(N_P, N_l, N_m, N_B);
dC_orb_sweep = NaN(N_P, N_l, N_m, N_B);
fprintf('  Sweep dimensions: %d P × %d l × %d m × %d B = %d combos\n', ...
    N_P, N_l, N_m, N_B, N_P*N_l*N_m*N_B);
fprintf('  Reference lines: 5yr=%.0fd, 10yr=%.0fd, cutoff=30yr\n\n', T_5yr, T_10yr);
opts_sw2 = odeset('RelTol',1e-10,'AbsTol',1e-12);
for ii = 1:N_P
for jj = 1:N_l
for mm = 1:N_m
for bb = 1:N_B
  combo_num = (ii-1)*N_l*N_m*N_B + (jj-1)*N_m*N_B + (mm-1)*N_B + bb;
  if mod(combo_num, 10) == 1
    fprintf('  Combo %d/%d: P=%.0fW, l=%.0fkm, m=%.0fkg, B=%.1f\n', ...
        combo_num, N_P*N_l*N_m*N_B, ...
        P_sweep_W(ii), l_sweep_m(jj)/1000, m_sweep_kg(mm), B_sweep_scl(bb));
  end
  p_sw2        = p;
  p_sw2.P_max_W = P_sweep_W(ii);
  p_sw2.l_m     = l_sweep_m(jj);
  p_sw2.mSC_kg  = m_sweep_kg(mm);
  p_sw2.B_scale = B_sweep_scl(bb);
  p_sw2.edt_on  = true;
  odefun_sw2 = @(t,Y) cr3bp_eom_nd_edt(t, Y, p_sw2);
  Y_sw2  = Y0_cap;
  t_sw2  = 0;
  C_prev = C_cap;
  dC_sum = 0;
  n_orb2 = 0;
  neck_sw2 = false;
  while n_orb2 <= 2000
    try
      [~, Y_seg2] = ode113(odefun_sw2, ...
        [t_sw2, t_sw2+T_orbit_nd], Y_sw2, opts_sw2);
      Y_sw2 = Y_seg2(end,:)';
    catch
      break
    end
```

```
        t_sw2  = t_sw2 + T_orbit_nd;
        n_orb2 = n_orb2 + 1;
        C_sw2  = compute_jacobi(Y_sw2, mu);
        dC_sum = dC_sum + (C_prev - C_sw2);
        C_prev = C_sw2;
        if C_sw2 < C_lyap + 0.05
            neck_sw2 = true;
            break
        end
        if t_sw2/n/86400 > T_30yr, break; end
    end
    if neck_sw2
        T_pump_sweep(ii,jj,mm,bb) = t_sw2 / n / 86400;
        dC_orb_sweep(ii,jj,mm,bb) = dC_sum / max(n_orb2, 1);
    end
end
end
end
end
fprintf('  Sweep complete.\n');
% Nominal slice indices — from master block values
[~, idx_m_nom] = min(abs(m_sweep_kg  - p.mSC_kg));
[~, idx_B_nom] = min(abs(B_sweep_scl - p.B_scale));
[~, idx_l_nom] = min(abs(l_sweep_m   - p.l_m));
[~, idx_P_nom] = min(abs(P_sweep_W   - p.P_max_W));
% Thesis baseline marker value — uses master block indices
T_baseline = T_pump_sweep(idx_P_nom, idx_l_nom, idx_m_nom, idx_B_nom);
% ── FIGURE A: Power vs pump-down, family = tether length ──────────────
fig_pwr = figure('Color',FIG_BG,'Name','Sensitivity: Power vs Pump-Down Duration');
hold on; grid on; box on;
set(gca,'Color',AX_BG,'XColor',AX_COL,'YColor',AX_COL,...
    'GridColor',GRID_COL,'FontSize',FONT_SM);
cmap_l = parula(N_l);
for jj = 1:N_l
    T_curve = T_pump_sweep(:, jj, idx_m_nom, idx_B_nom);
    valid   = ~isnan(T_curve);
    if any(valid)
        plot(P_sweep_W(valid), T_curve(valid), '-o', ...
            'Color',cmap_l(jj,:), 'LineWidth',1.8, 'MarkerSize',6, ...
            'MarkerFaceColor',cmap_l(jj,:), ...
```

```
            'DisplayName',sprintf('l = %.0f km', l_sweep_m(jj)/1000));
    end
end
yline(T_5yr,  '--','Color',[0.4 1.0 0.4],'LineWidth',2.0,...
    'Label','5-year limit','LabelColor',[0.4 1.0 0.4],...
    'FontSize',FONT_SM,'DisplayName','5-year limit');
yline(T_10yr, '--','Color',[1.0 0.8 0.2],'LineWidth',2.0,...
    'Label','10-year limit','LabelColor',[1.0 0.8 0.2],...
    'FontSize',FONT_SM,'DisplayName','10-year limit');
yline(T_30yr, ':','Color',[1.0 0.4 0.4],'LineWidth',1.5,...
    'Label','30-year cutoff','LabelColor',[1.0 0.4 0.4],...
    'FontSize',FONT_SM,'DisplayName','Infeasibility cutoff');
if ~isnan(T_baseline)
  plot(p.P_max_W, T_baseline, 'p','Color','w','MarkerFaceColor','w',...
      'MarkerSize',14,'LineWidth',1.5,...
      'DisplayName',sprintf('Baseline (%.0fW, %.0fkm, %.0fkg)', ...
                    p.P_max_W, p.l_m/1000, p.mSC_kg));
end
set(gca,'XScale','log');
xlabel('Solar array power  P_{max}  [W]','Color',AX_COL,'FontSize',FONT_MD);
ylabel('EDT pump-down duration  [days]', 'Color',AX_COL,'FontSize',FONT_MD);
title({'EDT Pump-Down Duration vs Power and Tether Length', ...
    sprintf('Nominal: m_{SC}=%.0f kg, B-scale=%.1f', p.mSC_kg, p.B_scale)}, ...
    'Color',AX_COL,'FontSize',FONT_LG,'Interpreter','none');
legend('show','TextColor',AX_COL,'Color',[0.12 0.12 0.12],...
    'Location','northeast','FontSize',FONT_SM);
xlim([min(P_sweep_W)*0.8, max(P_sweep_W)*1.2]);
% ── FIGURE B: Power vs pump-down, family = spacecraft mass ──────────────
fig_mass = figure('Color',FIG_BG,'Name','Sensitivity: Mass vs Pump-Down Duration');
hold on; grid on; box on;
set(gca,'Color',AX_BG,'XColor',AX_COL,'YColor',AX_COL,...
  'GridColor',GRID_COL,'FontSize',FONT_SM);
cmap_m = lines(N_m);
for mm = 1:N_m
  T_curve = T_pump_sweep(:, idx_l_nom, mm, idx_B_nom);
  valid   = ~isnan(T_curve);
  if any(valid)
    plot(P_sweep_W(valid), T_curve(valid), '-o', ...
        'Color',cmap_m(mm,:), 'LineWidth',1.8, 'MarkerSize',6, ...
        'MarkerFaceColor',cmap_m(mm,:), ...
```

```
        'DisplayName',sprintf('m_{SC} = %d kg', m_sweep_kg(mm)));
  end
end
yline(T_5yr,  '--','Color',[0.4 1.0 0.4],'LineWidth',2.0,...
    'Label','5-year','LabelColor',[0.4 1.0 0.4],...
    'FontSize',FONT_SM,'DisplayName','5-year limit');
yline(T_10yr, '--','Color',[1.0 0.8 0.2],'LineWidth',2.0,...
    'Label','10-year','LabelColor',[1.0 0.8 0.2],...
    'FontSize',FONT_SM,'DisplayName','10-year limit');
if ~isnan(T_baseline)
  plot(p.P_max_W, T_baseline, 'p','Color','w','MarkerFaceColor','w',...
      'MarkerSize',14,'LineWidth',1.5,...
      'DisplayName',sprintf('Baseline (%.0fW, %.0fkm, %.0fkg)', ...
                    p.P_max_W, p.l_m/1000, p.mSC_kg));
end
set(gca,'XScale','log');
xlabel('Solar array power  P_{max}  [W]','Color',AX_COL,'FontSize',FONT_MD);
ylabel('EDT pump-down duration  [days]', 'Color',AX_COL,'FontSize',FONT_MD);
title({'EDT Pump-Down Duration vs Power and Spacecraft Mass', ...
     sprintf('Nominal: l=%.0f km, B-scale=%.1f  |  MEDT mass budget uncertainty', ...
          p.l_m/1000, p.B_scale)}, ...
    'Color',AX_COL,'FontSize',FONT_LG,'Interpreter','none');
legend('show','TextColor',AX_COL,'Color',[0.12 0.12 0.12],...
     'Location','northeast','FontSize',FONT_SM);
xlim([min(P_sweep_W)*0.8, max(P_sweep_W)*1.2]);

% ── FIGURE C: B-field uncertainty bands ──────────────────────────
fig_Bfield = figure('Color',FIG_BG,'Name','Sensitivity: B-field Uncertainty');
hold on; grid on; box on;
set(gca,'Color',AX_BG,'XColor',AX_COL,'YColor',AX_COL,...
  'GridColor',GRID_COL,'FontSize',FONT_SM);
B_colors = {[0.4 0.6 1.0], [1.0 1.0 1.0], [1.0 0.5 0.3]};
B_labels = {'B \times 0.8 (weak field)', 'B \times 1.0 (nominal)', ...
         'B \times 1.2 (strong field)'};
T_B_curves = NaN(N_P, N_B);
for bb = 1:N_B
  T_curve = T_pump_sweep(:, idx_l_nom, idx_m_nom, bb);
  T_B_curves(:,bb) = T_curve;
  valid = ~isnan(T_curve);
```

```
  if any(valid)
    lw = 1.2 + 0.6*(bb==2);
    plot(P_sweep_W(valid), T_curve(valid), '-o', ...
        'Color',B_colors{bb}, 'LineWidth',lw, 'MarkerSize',6, ...
        'MarkerFaceColor',B_colors{bb}, 'DisplayName',B_labels{bb});
  end
end
valid_band = ~isnan(T_B_curves(:,1)) & ~isnan(T_B_curves(:,3));
if any(valid_band)
  P_band = P_sweep_W(valid_band);
  fill([P_band, fliplr(P_band)], ...
      [T_B_curves(valid_band,3)', fliplr(T_B_curves(valid_band,1)')], ...
      [0.5 0.5 0.7], 'FaceAlpha',0.2, 'EdgeColor','none', ...
      'HandleVisibility','off');
end
yline(T_5yr,  '--','Color',[0.4 1.0 0.4],'LineWidth',2.0,...
    'Label','5-year','LabelColor',[0.4 1.0 0.4],...
    'FontSize',FONT_SM,'DisplayName','5-year limit');
yline(T_10yr, '--','Color',[1.0 0.8 0.2],'LineWidth',2.0,...
    'Label','10-year','LabelColor',[1.0 0.8 0.2],...
    'FontSize',FONT_SM,'DisplayName','10-year limit');
if ~isnan(T_baseline)
  plot(p.P_max_W, T_baseline, 'p','Color','w','MarkerFaceColor','w',...
      'MarkerSize',14,'LineWidth',1.5,'DisplayName','Baseline');
end
set(gca,'XScale','log');
xlabel('Solar array power  P_{max}  [W]','Color',AX_COL,'FontSize',FONT_MD);
ylabel('EDT pump-down duration  [days]', 'Color',AX_COL,'FontSize',FONT_MD);
title({'B-field Uncertainty: ±20% Effect on Pump-Down Duration', ...
    sprintf('Nominal: l=%.0f km, m_{SC}=%.0f kg  |  Shaded = uncertainty envelope', ...
        p.l_m/1000, p.mSC_kg)}, ...
    'Color',AX_COL,'FontSize',FONT_LG,'Interpreter','none');
legend('show','TextColor',AX_COL,'Color',[0.12 0.12 0.12],...
    'Location','northeast','FontSize',FONT_SM);
xlim([min(P_sweep_W)*0.8, max(P_sweep_W)*1.2]);

% ── FIGURE D: ΔC per orbit vs power ──────────────────────────────────
fig_dC = figure('Color',FIG_BG,'Name','Sensitivity: DeltaC per Orbit');
hold on; grid on; box on;
```

```
set(gca,'Color',AX_BG,'XColor',AX_COL,'YColor',AX_COL,...
  'GridColor',GRID_COL,'FontSize',FONT_SM);
for jj = 1:N_l
  dC_curve = dC_orb_sweep(:, jj, idx_m_nom, idx_B_nom);
  valid    = ~isnan(dC_curve);
  if any(valid)
     plot(P_sweep_W(valid), dC_curve(valid), '-o', ...
         'Color',cmap_l(jj,:), 'LineWidth',1.8, 'MarkerSize',6, ...
         'MarkerFaceColor',cmap_l(jj,:), ...
         'DisplayName',sprintf('l = %.0f km', l_sweep_m(jj)/1000));
  end
end
dC_baseline = (C_cap - C_pump_end) / (T_pump_actual/T_orbit_nd);
plot(p.P_max_W, dC_baseline, 'p','Color','w','MarkerFaceColor','w',...
   'MarkerSize',14,'LineWidth',1.5,...
   'DisplayName',sprintf('Baseline = %.5f nd/orbit', dC_baseline));
set(gca,'XScale','log');
xlabel('Solar array power  P_{max}  [W]','Color',AX_COL,'FontSize',FONT_MD);
ylabel('\DeltaC per capture orbit  [nd/orbit]','Color',AX_COL,'FontSize',FONT_MD);
title({'\DeltaC per Orbit vs Power and Tether Length', ...
     sprintf('Nominal: m=%.0f kg, B-scale=%.1f', p.mSC_kg, p.B_scale)}, ...
    'Color',AX_COL,'FontSize',FONT_LG,'Interpreter','tex');
legend('show','TextColor',AX_COL,'Color',[0.12 0.12 0.12],...
     'Location','northwest','FontSize',FONT_SM);
xlim([min(P_sweep_W)*0.8, max(P_sweep_W)*1.2]);

% ── Console feasibility summary ──────────────────────────────────────
fprintf('\n  Baseline: P=%.0fW, l=%.0fkm, m=%.0fkg, B=%.1f → %.0f days (%.1f years)\n', ...
      p.P_max_W, p.l_m/1000, p.mSC_kg, p.B_scale, ...
      T_pump_actual/n/86400, T_pump_actual/n/86400/365.25);
fprintf('\n  Feasibility summary (neck opens within 10 years):\n');
fprintf('  %-8s  %-8s  %-8s  %-8s  %-12s\n', ...
      'P[W]','l[km]','m[kg]','B_scl','T_pump[d]');
for ii = 1:N_P
for jj = 1:N_l
for mm = 1:N_m
for bb = 1:N_B
  T_val = T_pump_sweep(ii,jj,mm,bb);
  if ~isnan(T_val) && T_val <= T_10yr
```

```
        fprintf('  %-8.0f  %-8.0f  %-8.0f  %-8.1f  %-12.0f\n', ...
            P_sweep_W(ii), l_sweep_m(jj)/1000, ...
            m_sweep_kg(mm), B_sweep_scl(bb), T_val);
    end
end
end
end
end
assignin('base','P_sweep_W',    P_sweep_W);
assignin('base','l_sweep_m',    l_sweep_m);
assignin('base','m_sweep_kg',   m_sweep_kg);
assignin('base','B_sweep_scl',  B_sweep_scl);
assignin('base','T_pump_sweep', T_pump_sweep);
assignin('base','dC_orb_sweep', dC_orb_sweep);
fprintf('\n── Sensitivity sweep complete ──────────────────────────────────────\n');
end  % if converged

%% SECTION 13b: OPERATIONAL REALISM CLUSTER
% ═══════════════════════════════════════════════════════════════════════
% Three analyses addressing realistic EDT operation vs idealized always-on:
%   (1) Duty cycle sweep — fraction of orbit with EDT active (0.3 to 1.0)
%   (2) Plasma density scaling — altitude-dependent Ne vs constant model
%   (3) Radiation dose estimate — moved to Section 16b (requires Y_cap_plt)
%
% Central question: how much does realistic operation degrade performance
% relative to the always-on idealized baseline?
%
% Physical basis for duty cycle constraint:
%   - Gravity gradient forces may be insufficient to maintain tether
%     orientation under high propulsive loads (Magnetour finding [8])
%   - Tether requires off-time near apojove for thermal recovery
%   - Plasma density near apojove too low for reliable circuit closure
%   - Conservative planning: 50-70% duty cycle likely realistic (Phase B)
%
% Physical basis for plasma density scaling:
%   - Bare tether OML current requires sufficient ambient electron density
%   - Io plasma torus: peak Ne at ~5.9 R_J, outer edge ~8-10 R_J
%   - Beyond torus: Ne drops sharply, circuit closure unreliable
%   - Model anchored to Sanmartin et al. 2008 Ne profile [7]
```

```
%
% NOTE: Radiation dose estimate (Part 3) is in Section 16b immediately
% after the JCI back-transform, where Y_cap_plt is guaranteed available.
if ~converged
  fprintf('\nSkipping operational realism cluster — no converged orbit.\n');
else
fprintf('\n── Operational Realism Cluster ──────────────────────────────────────\n');
T_baseline_duty = T_pump_actual / n / 86400;   % always-on baseline [days]
% ── (1) Duty Cycle Sweep ──────────────────────────────────────────────────

fprintf('\n  Part 1: Duty Cycle Sweep\n');
fprintf('  Baseline: always-on (duty_cycle = 1.0) = %.0f days\n', T_baseline_duty);
fprintf('  Physical constraint: attitude control, thermal recovery, plasma gaps\n\n');
duty_sweep  = [0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0];
N_duty      = length(duty_sweep);
T_pump_duty = NaN(N_duty, 1);
dC_orb_duty = NaN(N_duty, 1);
opts_duty = odeset('RelTol',1e-10,'AbsTol',1e-12);
fprintf('  %-12s  %-14s  %-14s  %-12s\n', ...
    'Duty cycle','T_pump [days]','T/baseline [%]','dC/orbit [nd]');
fprintf('  %-12s  %-14s  %-14s  %-12s\n', ...
    '----------','--------------','--------------','------------');
for ii = 1:N_duty
  p_duty             = p;
  p_duty.edt_on      = true;
  p_duty.duty_cycle  = duty_sweep(ii);
  p_duty.plasma_scaling = false;
  odefun_duty = @(t,Y) cr3bp_eom_nd_edt(t, Y, p_duty);
  Y_d   = Y0_cap;
  t_d   = 0;
  C_prev = C_cap;
  dC_sum = 0;
  n_orb  = 0;
  neck_d = false;
  while n_orb <= 2000
    try
      [~, Y_seg] = ode113(odefun_duty, ...
          [t_d, t_d + T_orbit_nd], Y_d, opts_duty);
      Y_d = Y_seg(end,:)';
```

```
        catch
            break
        end
        t_d    = t_d + T_orbit_nd;
        n_orb  = n_orb + 1;
        C_d    = compute_jacobi(Y_d, mu);
        dC_sum = dC_sum + (C_prev - C_d);
        C_prev = C_d;
        if C_d < C_lyap + 0.05
            neck_d = true;
            break
        end
        if t_d/n/86400 > 30*365.25, break; end
    end
    if neck_d
        T_pump_duty(ii) = t_d / n / 86400;
        dC_orb_duty(ii) = dC_sum / max(n_orb, 1);
        ratio           = 100 * T_pump_duty(ii) / T_baseline_duty;
        fprintf('  %-12.1f  %-14.0f  %-14.1f  %-12.6f\n', ...
                duty_sweep(ii), T_pump_duty(ii), ratio, dC_orb_duty(ii));
    else
        fprintf('  %-12.1f  %-14s  %-14s  %-12s\n', ...
                duty_sweep(ii), '>30yr', '---', '---');
    end
end

% Duty cycle figure
fig_duty = figure('Color',FIG_BG,'Name','Operational Realism: Duty Cycle');
hold on; grid on; box on;
set(gca,'Color',AX_BG,'XColor',AX_COL,'YColor',AX_COL,...
    'GridColor',GRID_COL,'FontSize',FONT_SM);
valid_d = ~isnan(T_pump_duty);
plot(duty_sweep(valid_d)*100, T_pump_duty(valid_d), 'c-o', ...
     'LineWidth',2.5,'MarkerSize',8,'MarkerFaceColor','c',...
     'DisplayName','EDT pump-down duration');
yline(5*365.25,  '--','Color',[0.4 1.0 0.4],'LineWidth',2.0,...
      'Label','5-year','LabelColor',[0.4 1.0 0.4],...
      'FontSize',FONT_SM,'DisplayName','5-year limit');
yline(10*365.25, '--','Color',[1.0 0.8 0.2],'LineWidth',2.0,...
      'Label','10-year','LabelColor',[1.0 0.8 0.2],...
```

```
    'FontSize',FONT_SM,'DisplayName','10-year limit');
plot(100, T_baseline_duty, 'p','Color','w','MarkerFaceColor','w',...
   'MarkerSize',14,'LineWidth',1.5,...
   'DisplayName',sprintf('Baseline (100%% = %.0f days)', T_baseline_duty));
y_lims_d = [0, max(T_pump_duty(valid_d))*1.15];
ylim(y_lims_d);
fill([50 80 80 50],[y_lims_d(1) y_lims_d(1) y_lims_d(2) y_lims_d(2)], ...
   [0.3 0.5 0.3],'FaceAlpha',0.2,'EdgeColor','none',...
   'DisplayName','Realistic operating band (50-80%)');
xlabel('EDT duty cycle  [%]',            'Color',AX_COL,'FontSize',FONT_MD);
ylabel('EDT pump-down duration  [days]',   'Color',AX_COL,'FontSize',FONT_MD);
title({'EDT Pump-Down Duration vs Tether Duty Cycle', ...
     sprintf('Nominal: P=%.0fW, l=%.0fkm, m=%.0fkg  |  Always-on baseline = %.0f days', ...
          p.P_max_W, p.l_m/1000, p.mSC_kg, T_baseline_duty)}, ...
    'Color',AX_COL,'FontSize',FONT_LG,'Interpreter','none');
legend('show','TextColor',AX_COL,'Color',[0.12 0.12 0.12],...
     'Location','northwest','FontSize',FONT_SM);
xlim([25 105]);

% ── (2) Plasma Density Scaling ──────────────────────────────────────────

fprintf('\n  Part 2: Plasma Density Scaling\n');
fprintf('  Comparing always-on (constant Ne) vs altitude-dependent Ne model\n');
fprintf('  Io plasma torus: peak at ~5.9 R_J, outer edge ~8-10 R_J\n');
fprintf('  Ne profile: ~r^-1.5 inside torus, sharp dropoff outside [7]\n\n');
opts_pl = odeset('RelTol',1e-10,'AbsTol',1e-12);
% Case A: always-on with plasma scaling
p_plasma              = p;
p_plasma.edt_on       = true;
p_plasma.duty_cycle   = 1.0;
p_plasma.plasma_scaling = true;
odefun_plasma = @(t,Y) cr3bp_eom_nd_edt(t, Y, p_plasma);
Y_pl   = Y0_cap;
t_pl   = 0;
C_prev  = C_cap;
dC_sum  = 0;
n_orb   = 0;
neck_pl = false;
while n_orb <= 2000
  try
```

```
        [~, Y_seg] = ode113(odefun_plasma, ...
            [t_pl, t_pl + T_orbit_nd], Y_pl, opts_pl);
        Y_pl = Y_seg(end,:)';
    catch
        break
    end
    t_pl   = t_pl + T_orbit_nd;
    n_orb  = n_orb + 1;
    C_pl   = compute_jacobi(Y_pl, mu);
    dC_sum = dC_sum + (C_prev - C_pl);
    C_prev = C_pl;
    if C_pl < C_lyap + 0.05
        neck_pl = true;
        break
    end
    if t_pl/n/86400 > 30*365.25, break; end
end
if neck_pl
    T_plasma_days = t_pl / n / 86400;
    plasma_ratio  = T_plasma_days / T_baseline_duty;
    fprintf('  Always-on, no plasma scaling:   %.0f days  (baseline)\n', ...
            T_baseline_duty);
    fprintf('  Always-on + plasma Ne scaling:  %.0f days  (%.2fx longer)\n', ...
            T_plasma_days, plasma_ratio);
    fprintf('  Interpretation: apojove EDT contribution negligible due to\n');
    fprintf('  low plasma density — pump-down dominated by perijove passes.\n');
else
    T_plasma_days = NaN;
    fprintf('  Plasma scaling run did not converge within 30 years.\n');
    fprintf('  EDT outside Io torus insufficient — perijove-only operation\n');
    fprintf('  required for feasibility at this configuration.\n');
end

% Case B: 70% duty cycle + plasma scaling (combined realistic case)
fprintf('\n  Combined realistic case: 70%% duty cycle + plasma Ne scaling\n');
p_combined              = p;
p_combined.edt_on       = true;
p_combined.duty_cycle   = 0.7;
p_combined.plasma_scaling = true;
odefun_comb = @(t,Y) cr3bp_eom_nd_edt(t, Y, p_combined);
```

```
Y_cb    = Y0_cap;
t_cb    = 0;
C_prev  = C_cap;
neck_cb = false;
while true
  try
     [~, Y_seg] = ode113(odefun_comb, ...
        [t_cb, t_cb + T_orbit_nd], Y_cb, opts_pl);
     Y_cb = Y_seg(end,:)';
  catch
     break
  end
  t_cb = t_cb + T_orbit_nd;
  C_cb = compute_jacobi(Y_cb, mu);
  C_prev = C_cb;
  if C_cb < C_lyap + 0.05
     neck_cb = true;
     break
  end
  if t_cb/n/86400 > 30*365.25, break; end
end
if neck_cb
  T_combined_days = t_cb / n / 86400;
  fprintf('  Combined result:   %.0f days  (%.2fx longer than baseline)\n', ...
        T_combined_days, T_combined_days/T_baseline_duty);
else
  T_combined_days = NaN;
  fprintf('  Combined case infeasible within 30 years at this configuration.\n');
end

% ── Operational Realism Summary Table
─────────────────────────────────────────────────

fprintf('\n── Operational Realism Summary
──────────────────────────────────────────────\n');
fprintf('  %-35s  %-14s  %-12s\n','Configuration','T_pump [days]','vs baseline');
fprintf('  %-35s  %-14s  %-12s\n','-----------','-------------','-----------');
fprintf('  %-35s  %-14.0f  %-12s\n', ...
     'Always-on, no plasma scaling', T_baseline_duty, '1.00x (baseline)');
if ~isnan(T_plasma_days)
  fprintf('  %-35s  %-14.0f  %-12.2fx\n', ...
```

```
        'Always-on + plasma Ne scaling', T_plasma_days, ...
        T_plasma_days/T_baseline_duty);
end
idx_70 = find(abs(duty_sweep - 0.7) < 0.01);
if ~isempty(idx_70) && ~isnan(T_pump_duty(idx_70))
  fprintf('  %-35s  %-14.0f  %-12.2fx\n', ...
        '70% duty, no plasma scaling', T_pump_duty(idx_70), ...
        T_pump_duty(idx_70)/T_baseline_duty);
end
if ~isnan(T_combined_days)
  fprintf('  %-35s  %-14.0f  %-12.2fx\n', ...
        '70% duty + plasma Ne scaling', T_combined_days, ...
        T_combined_days/T_baseline_duty);
end
fprintf('\n  Key finding: pump-down scales nearly linearly with duty cycle.\n');
fprintf('  Plasma Ne scaling adds modest additional degradation, confirming\n');
fprintf('  EDT work is concentrated at perijove passes inside the Io torus.\n');
fprintf('  Radiation dose estimate: see Section 16b.\n');
% Export
assignin('base','duty_sweep',       duty_sweep);
assignin('base','T_pump_duty',      T_pump_duty);
assignin('base','dC_orb_duty',      dC_orb_duty);
assignin('base','T_plasma_days',    T_plasma_days);
assignin('base','T_combined_days',  T_combined_days);
fprintf('\n── Operational realism cluster complete ──────────────────────────\n');
end  % if converged
%% SECTION 14: CAPTURE ORBIT GEOMETRY SWEEP
──────────────────────────────────────────
% Two sweeps:
%   (1) Perijove radius r_p from 1 to 26 R_J — radiation vs efficiency
%       tradeoff. High radiation zone boundary at 17 R_J (inward is danger).
%   (2) Finer arrival angle sweep — dynamically centered on best coarse angle.
%   (3) Dedicated Poincaré map at best arrival angle — headline manifold figure.
if ~converged
  fprintf('\nSkipping geometry sweep — Lyapunov orbit not converged.\n');
else
fprintf('\n── Capture Orbit Geometry Sweep
──────────────────────────────────────────\n');
% ── Part 1: Perijove radius sweep
──────────────────────────────────────────────────────────
```

```
r_p_sweep_RJ     = [1, 2, 3, 4, 5, 6, 6.5, 7, 7.25, 7.5, 7.75, 8, 8.5, 9, 10, 12, 14, 16, 18, 20, 22,
24, 26];
N_rp             = length(r_p_sweep_RJ);
T_pump_rp        = NaN(N_rp, 1);
dC_orb_rp        = NaN(N_rp, 1);
C_cap_rp         = NaN(N_rp, 1);
r_rad_boundary_RJ = 17.0;   % Inward of this = high radiation zone
opts_rp     = odeset('RelTol',1e-10,'AbsTol',1e-12);
p_rp        = p;
p_rp.edt_on = true;
T_pump_max  = 2000 * T_orbit_nd;
fprintf('  Sweeping perijove %.0f to %.0f R_J (%d points)\n', ...
      r_p_sweep_RJ(1), r_p_sweep_RJ(end), N_rp);
fprintf('  Apojove fixed at %.0f R_J\n', r_a_RJ);
fprintf('  High radiation zone: r_p < %.0f R_J (baseline %.0f R_J is INSIDE)\n\n', ...
      r_rad_boundary_RJ, r_p_RJ);
fprintf('  %-10s  %-12s  %-14s  %-12s  %-10s\n', ...
      'r_p [R_J]','C_cap [nd]','T_pump [days]','dC/orbit [nd]','N orbits');
fprintf('  %-10s  %-12s  %-14s  %-12s  %-10s\n', ...
      '---------','----------','-------------','------------','-------');
for ii = 1:N_rp
   r_p_ii_km  = r_p_sweep_RJ(ii) * p.R_J_km;
   r_a_ii_km  = r_a_joi;
   a_ii_km    = 0.5 * (r_p_ii_km + r_a_ii_km);
   v_ell_p_ii = sqrt(mu_J_joi * (2/r_p_ii_km - 1/a_ii_km));
   rSC_ii     = [r_p_ii_km; 0; 0];
   vSC_ii     = [0; v_ell_p_ii; 0];
   rSC_BI_ii  = rSC_ii - rBary_JCI;
   vSC_BI_ii  = vSC_ii - vBary_JCI;
   rSC_S_ii   = R_I2S' * rSC_BI_ii;
   vSC_S_ii   = R_I2S' * (vSC_BI_ii - cross(omega_vec_JCI, rSC_BI_ii));
   Y0_rp_ii   = [rSC_S_ii/Lstar; vSC_S_ii/Vstar];
   C_cap_ii     = compute_jacobi(Y0_rp_ii, mu);
   C_cap_rp(ii) = C_cap_ii;
   T_orb_ii_nd  = (2*pi * sqrt(a_ii_km^3 / mu_J_joi)) * n;
   Y_rp   = Y0_rp_ii;
   t_rp   = 0;
   C_prev = C_cap_ii;
   dC_sum = 0;
   n_orb  = 0;
```

```
    neck_rp = false;
    odefun_rp = @(t,Y) cr3bp_eom_nd_edt(t, Y, p_rp);
    while t_rp < T_pump_max
        if n_orb > 1500, break; end
        try
            [~, Y_seg] = ode113(odefun_rp, [t_rp, t_rp+T_orb_ii_nd], Y_rp, opts_rp);
            Y_rp = Y_seg(end,:)';
        catch
            break
        end
        t_rp   = t_rp + T_orb_ii_nd;
        n_orb  = n_orb + 1;
        C_rp   = compute_jacobi(Y_rp, mu);
        dC_sum = dC_sum + (C_prev - C_rp);
        C_prev = C_rp;
        if C_rp < C_lyap + 0.05
            neck_rp = true;
            break
        end
        if t_rp/n/86400 > 30*365.25, break; end
    end
    if neck_rp
        T_pump_rp(ii) = t_rp / n / 86400;
        dC_orb_rp(ii) = dC_sum / max(n_orb, 1);
        if     T_pump_rp(ii) <= 5*365.25,  feas_str = '✓ <5yr';
        elseif T_pump_rp(ii) <= 10*365.25, feas_str = '✓ <10yr';
        else,                              feas_str = '~ >10yr';
        end
        fprintf('  %-10.1f  %-12.6f  %-14.0f  %-12.6f  %-10d  %s\n', ...
                r_p_sweep_RJ(ii), C_cap_ii, T_pump_rp(ii), dC_orb_rp(ii), n_orb, feas_str);
    else
        fprintf('  %-10.1f  %-12.6f  %-14s  %-12s  %-10d\n', ...
                r_p_sweep_RJ(ii), C_cap_ii, '>30yr (infeas)', '---', n_orb);
    end
end

% ── Figure: pump-down duration vs perijove radius ──────────────────────
fig_rp = figure('Color',FIG_BG,'Name','Geometry Sweep: Pump-Down vs Perijove Radius');
hold on; grid on; box on;
set(gca,'Color',AX_BG,'XColor',AX_COL,'YColor',AX_COL,...
```

```
  'GridColor',GRID_COL,'FontSize',FONT_SM);
valid_rp = ~isnan(T_pump_rp);
plot(r_p_sweep_RJ(valid_rp), T_pump_rp(valid_rp), 'c-o', ...
   'LineWidth',2.5,'MarkerSize',8,'MarkerFaceColor','c', ...
   'DisplayName','EDT pump-down duration');
yline(5*365.25,  '--','Color',[0.4 1.0 0.4],'LineWidth',2.0,...
    'Label','5-year limit','LabelColor',[0.4 1.0 0.4],...
    'FontSize',FONT_SM,'DisplayName','5-year limit');
yline(10*365.25, '--','Color',[1.0 0.8 0.2],'LineWidth',2.0,...
    'Label','10-year limit','LabelColor',[1.0 0.8 0.2],...
    'FontSize',FONT_SM,'DisplayName','10-year limit');
y_lims = [0, max(T_pump_rp(valid_rp))*1.15];
ylim(y_lims);
% Shade high radiation zone — everything INWARD of r_rad_boundary_RJ
fill([0, r_rad_boundary_RJ, r_rad_boundary_RJ, 0], ...
   [y_lims(1), y_lims(1), y_lims(2), y_lims(2)], ...
   [1.0 0.3 0.3], 'FaceAlpha',0.25, 'EdgeColor','none', ...
   'DisplayName',sprintf('High radiation zone (r_p < %.0f R_J)', r_rad_boundary_RJ));
% Boundary line at 17 R_J
xline(r_rad_boundary_RJ, '-', 'Color',[1.0 0.5 0.3], 'LineWidth', 1.5, ...
    'Label', sprintf('%.0f R_J boundary', r_rad_boundary_RJ), ...
    'LabelColor', [1.0 0.5 0.3], 'FontSize', FONT_SM, ...
    'HandleVisibility','off');
% Baseline marker
[~, idx_rp_base] = min(abs(r_p_sweep_RJ - r_p_RJ));
if ~isnan(T_pump_rp(idx_rp_base))
  plot(r_p_sweep_RJ(idx_rp_base), T_pump_rp(idx_rp_base), 'p', ...
      'Color','w','MarkerFaceColor','w','MarkerSize',14,'LineWidth',1.5,...
      'DisplayName',sprintf('Baseline (%.0f R_J, %.0f days)', ...
                  r_p_RJ, T_pump_rp(idx_rp_base)));
end
% Dotted vertical line at baseline perijove
xline(r_p_RJ, ':', 'Color',[1.0 0.8 0.4], 'LineWidth', 1.5, ...
    'Label', sprintf('Baseline %.0f R_J', r_p_RJ), ...
    'LabelColor', [1.0 0.8 0.4], 'FontSize', FONT_SM, ...
    'HandleVisibility','off');
% Annotation text on the right side of boundary
text(r_rad_boundary_RJ + 0.3, y_lims(2)*0.85, ...
   {'High radiation zone (< 17 R_J):', ...
    sprintf('Baseline %.0f R_J is inside this zone', r_p_RJ), ...
```

```
    'Higher r_p reduces dose but extends pump-down'}, ...
   'Color',[1.0 0.6 0.6],'FontSize',FONT_SM,'HorizontalAlignment','left');
xlabel('Capture orbit perijove  r_p  [R_J]','Color',AX_COL,'FontSize',FONT_MD);
ylabel('EDT pump-down duration  [days]',    'Color',AX_COL,'FontSize',FONT_MD);
title({'EDT Pump-Down Duration vs Capture Orbit Perijove Radius', ...
     sprintf('Nominal: P=%.0fW, l=%.0fkm, m=%.0fkg  |  Apojove fixed at %.0f R_J', ...
           p.P_max_W, p.l_m/1000, p.mSC_kg, r_a_RJ)}, ...
    'Color',AX_COL,'FontSize',FONT_LG,'Interpreter','none');
legend('show','TextColor',AX_COL,'Color',[0.12 0.12 0.12],...
     'Location','northwest','FontSize',FONT_SM);
xlim([0, 27]);

% ── Part 2: Finer arrival angle sweep ──────────────────────────────────

fprintf('\n── Finer Arrival Angle Sweep (dynamic, centered on best coarse angle) ──\n');
% Build fine sweep dynamically around the best coarse angle only
if ~isinf(best_overall_dist) && ~isnan(best_theta)
   theta_around_best = (best_theta - 10) : 1 : (best_theta + 10);
   theta_fine = theta_around_best(theta_around_best >= 0 & theta_around_best < 360);
   fprintf('  Centering fine sweep on best coarse angle: %.0f°\n', best_theta);
else
   % No promising angle found in coarse sweep — fallback full sweep
   theta_fine = 0 : 5 : 355;
   fprintf('  WARNING: No coarse connection found — running fallback full sweep at 5°
steps\n');
end
N_fine       = length(theta_fine);
min_dist_fine = NaN(N_fine, 1);
cap_x_fine_all  = cell(N_fine, 1);
cap_xd_fine_all = cell(N_fine, 1);
fprintf('  Fine sweep: %d angles\n', N_fine);
fprintf('  %-10s  %-14s  %-14s\n', 'θ [deg]','Cap near L1','Min dist [nd]');
fprintf('  %-10s  %-14s  %-14s\n', '-------','-----------','-------------');
opts_fine     = odeset('RelTol',1e-10,'AbsTol',1e-12,'Events',@poincare_section_event);
p_fine        = p;
p_fine.edt_on = true;
T_cap_fine    = 2000 * pi;
for ii = 1:N_fine
   theta_ii = theta_fine(ii);
   R_rot_ii = [cosd(theta_ii) -sind(theta_ii) 0;
```

```
            sind(theta_ii)  cosd(theta_ii) 0;
            0               0              1];
    Y_fine_ii = [R_rot_ii * Y_current(1:3); R_rot_ii * Y_current(4:6)];
    odefun_fine = @(t,Y) cr3bp_eom_nd_edt(t, Y, p_fine);
    [~,~,~,Ye_fine,~] = ode113(odefun_fine, [0, T_cap_fine], Y_fine_ii, opts_fine);
    cap_x_ii  = [];
    cap_xd_ii = [];
    if ~isempty(Ye_fine)
       for jj = 1:size(Ye_fine,1)
          if Ye_fine(jj,4) < 0
             x_cross  = Ye_fine(jj,1);
             xd_cross = Ye_fine(jj,4);
             if abs(x_cross) < 5.0 && abs(xd_cross) < 2.0
                cap_x_ii  = [cap_x_ii;  x_cross];  %#ok<AGROW>
                cap_xd_ii = [cap_xd_ii; xd_cross]; %#ok<AGROW>
             end
          end
       end
    end
    cap_x_fine_all{ii}  = cap_x_ii;
    cap_xd_fine_all{ii} = cap_xd_ii;
    near_ii = abs(cap_x_ii - xL1) < 0.15;
    md = Inf;
    if any(near_ii) && ~isempty(Ws_poincare_x)
       idx = find(near_ii);
       for i = 1:length(idx)
          for j = 1:length(Ws_poincare_x)
             dx  = cap_x_ii(idx(i))  - Ws_poincare_x(j);
             dxd = cap_xd_ii(idx(i)) - Ws_poincare_xd(j);
             d   = sqrt(dx^2 + dxd^2);
             if d < md, md = d; end
          end
       end
    end
    min_dist_fine(ii) = md;
    fprintf('  %-10.0f  %-14d  %-14.4f\n', theta_ii, sum(near_ii), md);
end
[best_dist_fine, best_idx_fine] = min(min_dist_fine);
best_theta_fine = theta_fine(best_idx_fine);
fprintf('\n  Best angle: θ = %.0f°  (gap = %.4f nd)\n', best_theta_fine, best_dist_fine);
```

```
% ── Figure: finer angle sweep Poincaré map ──────────────────────
fig_fine = figure('Color',FIG_BG,'Name','Finer Angle Sweep — Poincare Map');
hold on; grid on; box on;
set(gca,'Color',AX_BG,'XColor',AX_COL,'YColor',AX_COL,...
  'GridColor',GRID_COL,'FontSize',FONT_SM);
if ~isempty(Ws_poincare_x)
  scatter(Ws_poincare_x, Ws_poincare_xd, 18, 'c', 'filled', ...
        'DisplayName','W^s stable manifold');
end
cmap_fine = parula(N_fine);
for ii = 1:N_fine
  if ~isempty(cap_x_fine_all{ii})
     scatter(cap_x_fine_all{ii}, cap_xd_fine_all{ii}, ...
           10, cmap_fine(ii,:), 'filled', 'HandleVisibility','off');
  end
  if ~isinf(min_dist_fine(ii))
     leg_str = sprintf('\\theta=%d\\circ (gap=%.4f nd)', ...
                  theta_fine(ii), min_dist_fine(ii));
  else
     leg_str = sprintf('\\theta=%d\\circ (gap=Inf)', theta_fine(ii));
  end
  scatter(NaN, NaN, 30, cmap_fine(ii,:), 'filled', 'DisplayName', leg_str);
end
xline(xL1, '--','Color','yellow','LineWidth',1.2,...
    'Label',sprintf('L_1 (x*=%.4f)',xL1),'LabelColor','yellow',...
    'DisplayName',sprintf('L_1 (x*=%.4f nd)',xL1));
xlabel(sprintf('x^*  at y=0 crossing  [nd,  1 nd = %.0f km]', Lstar), ...
     'Color',AX_COL,'FontSize',FONT_MD);
ylabel(sprintf('xdot^*  at y=0 crossing  [nd,  1 nd = %.2f km/s]', Vstar), ...
     'Color',AX_COL,'FontSize',FONT_MD);
title({'Finer Arrival Angle Sweep — Poincare Map (y=0, xdot<0)', ...
     sprintf('Best: \\theta=%.0f° (gap=%.4f nd)  |  Searched around coarse best %.0f°', ...
           best_theta_fine, best_dist_fine, best_theta)}, ...
    'Color',AX_COL,'FontSize',FONT_SM,'Interpreter','tex');
legend('show','TextColor',AX_COL,'Color',[0.12 0.12 0.12],...
     'Location','southwest','FontSize',FONT_SM);
xlim([-2, 4]);
ylim([-0.8, 0.05]);
```

```
% ── Part 3: Dedicated best-angle Poincaré map (headline manifold figure) ──
fprintf('\n── Best Angle Poincaré Map (θ = %.0f°) ──────────────────────────────\n', ...
      best_theta_fine);
cap_x_best  = cap_x_fine_all{best_idx_fine};
cap_xd_best = cap_xd_fine_all{best_idx_fine};
fig_best_poin = figure('Color',FIG_BG, ...
  'Name',sprintf('Near-Heteroclinic Connection — theta=%.0f deg', best_theta_fine));
hold on; grid on; box on;
set(gca,'Color',AX_BG,'XColor',AX_COL,'YColor',AX_COL,...
  'GridColor',GRID_COL,'FontSize',FONT_SM);
% Stable manifold crossings
if ~isempty(Ws_poincare_x)
  scatter(Ws_poincare_x, Ws_poincare_xd, 25, 'c', 'filled', ...
        'DisplayName','W^s crossings (stable manifold)');
end
% Best angle capture crossings
if ~isempty(cap_x_best)
  scatter(cap_x_best, cap_xd_best, 20, 'm', 'filled', ...
        'DisplayName',sprintf('Capture trajectory (\\theta=%.0f°)', best_theta_fine));
end
% Find and annotate closest approach pair
if ~isinf(best_dist_fine) && ~isempty(cap_x_best)
  min_d   = Inf;
  bx_ws   = NaN; bxd_ws  = NaN;
  bx_cap  = NaN; bxd_cap = NaN;
  near    = abs(cap_x_best - xL1) < 0.15;
  cap_x_near  = cap_x_best(near);
  cap_xd_near = cap_xd_best(near);
  for i = 1:length(cap_x_near)
     for j = 1:length(Ws_poincare_x)
        d = sqrt((cap_x_near(i) - Ws_poincare_x(j))^2 + ...
              (cap_xd_near(i) - Ws_poincare_xd(j))^2);
        if d < min_d
           min_d   = d;
           bx_cap  = cap_x_near(i);
           bxd_cap = cap_xd_near(i);
           bx_ws   = Ws_poincare_x(j);
           bxd_ws  = Ws_poincare_xd(j);
        end
```

```
        end
    end
    % Gap arrow
    plot([bx_ws, bx_cap], [bxd_ws, bxd_cap], 'w--', 'LineWidth',2.0, ...
        'HandleVisibility','off');
    scatter(bx_ws,  bxd_ws,  100, 'c', 'filled', ...
          'MarkerEdgeColor','w', 'LineWidth',1.5, ...
          'DisplayName','Nearest W^s point');
    scatter(bx_cap, bxd_cap, 100, 'm', 'filled', ...
          'MarkerEdgeColor','w', 'LineWidth',1.5, ...
          'DisplayName','Nearest capture point');
    % Gap annotation
    text((bx_ws+bx_cap)/2 + 0.005, (bxd_ws+bxd_cap)/2 - 0.008, ...
        sprintf('Gap = %.4f nd\n\\approx %.0f m/s\n(Phase 2 EDT steering cost)', ...
              best_dist_fine, best_dist_fine * Vstar * 1e3), ...
        'Color','w','FontSize',FONT_SM,'HorizontalAlignment','left');
    n_cross_near = sum(abs(cap_x_best - xL1) < 0.15);
else
    % No connection found — show informational annotation
    n_cross_near = 0;
    text(xL1 + 0.02, -0.10, ...
        {'No capture crossings found near L_1', ...
         sprintf('at \\theta=%.0f°', best_theta_fine), ...
         'Larger Phase 2 steering required'}, ...
        'Color',[1.0 0.6 0.6],'FontSize',FONT_SM,'HorizontalAlignment','left');
end
xline(xL1,'--','Color','yellow','LineWidth',1.2,...
    'Label',sprintf('L_1 (x*=%.4f)',xL1),'LabelColor','yellow',...
    'DisplayName',sprintf('L_1 (x*=%.4f nd)',xL1));
xlabel(sprintf('x^*  at y=0 crossing  [nd,  1 nd = %.0f km]', Lstar),...
    'Color',AX_COL,'FontSize',FONT_MD);
ylabel(sprintf('xdot^*  at y=0 crossing  [nd,  1 nd = %.2f km/s]', Vstar),...
    'Color',AX_COL,'FontSize',FONT_MD);
if ~isinf(best_dist_fine)
    title({sprintf('Near-Heteroclinic Connection — Optimal Arrival Geometry \\theta=%.0f°', ...
              best_theta_fine), ...
         sprintf('Phase-space gap = %.4f nd  \\approx  %.0f m/s  |  %d crossing opportunities near
L_1', ...
              best_dist_fine, best_dist_fine*Vstar*1e3, n_cross_near)}, ...
        'Color',AX_COL,'FontSize',FONT_SM,'Interpreter','tex');
```

```
else
  title({sprintf('Best Arrival Geometry — \\theta=%.0f°  (no L_1 connection found)', ...
              best_theta_fine), ...
        sprintf('All gaps = Inf nd  |  Phase 2 steering requirement unquantified at this geometry')},
...
       'Color',AX_COL,'FontSize',FONT_SM,'Interpreter','tex');
end
legend('show','TextColor',AX_COL,'Color',[0.12 0.12 0.12],...
    'Location','southwest','FontSize',FONT_SM);

% Zoom to L1 neighborhood
xlim([0.75, 1.15]);
ylim([-0.20, 0.02]);
fprintf('  Best angle Poincaré map complete (zoomed to L1 region).\n');
if ~isinf(best_dist_fine)
  fprintf('  Gap: %.4f nd = %.0f m/s  |  %d crossing opportunities\n', ...
        best_dist_fine, best_dist_fine*Vstar*1e3, n_cross_near);
else
  fprintf('  No L1 connection found at %.0f° — gap = Inf\n', best_theta_fine);
  fprintf('  Consider extending T_cap or running a broader angle sweep.\n');
end

% ── Export ──────────────────────────────────────────────────────────────────

assignin('base','r_p_sweep_RJ',    r_p_sweep_RJ);
assignin('base','T_pump_rp',       T_pump_rp);
assignin('base','dC_orb_rp',       dC_orb_rp);
assignin('base','C_cap_rp',        C_cap_rp);
assignin('base','theta_fine',      theta_fine);
assignin('base','min_dist_fine',   min_dist_fine);
assignin('base','best_theta_fine', best_theta_fine);
assignin('base','best_dist_fine',  best_dist_fine);
assignin('base','cap_x_best',      cap_x_best);
assignin('base','cap_xd_best',     cap_xd_best);
fprintf('\n── Geometry sweep complete ──────────────────────────────────────────\n');
end  % if converged
```

```
%% SECTION 15: CONSOLE OUTPUT — MONODROMY
─────────────────────────────────────────────────────────
if ~converged
  fprintf('\nSkipping monodromy output — Lyapunov orbit not converged.\n');
else
fprintf('\nMonodromy matrix Phi (6x6):\n');
for i = 1:6
  fprintf('  %12.8e  %12.8e  %12.8e  %12.8e  %12.8e  %12.8e\n', ...
        Phi(i,1), Phi(i,2), Phi(i,3), Phi(i,4), Phi(i,5), Phi(i,6));
end
fprintf('\nMonodromy eigenvalues (lam):\n');
for i = 1:numel(lam)
  fprintf('  lam(%d) = %+12.8e %+12.8ei   |lam| = %.6f\n', ...
        i, real(lam(i)), imag(lam(i)), abs(lam(i)));
end
if isempty(Wu_dirs)
  fprintf('\nNo unstable directions found (|lambda|>1 not detected).\n');
else
  fprintf('\nUnstable manifold direction vectors (Wu_dirs) [x y z xd yd zd]:\n');
  for k = 1:size(Wu_dirs,2)
    v = Wu_dirs(:,k);
    fprintf('  Wu_dir(%d):  %12.8e  %12.8e  %12.8e  %12.8e  %12.8e  %12.8e\n', ...
          k, v(1), v(2), v(3), v(4), v(5), v(6));
  end
end
if isempty(Ws_dirs)
  fprintf('\nNo stable directions found (|lambda|<1 not detected).\n');
else
  fprintf('\nStable manifold direction vectors (Ws_dirs) [x y z xd yd zd]:\n');
  for k = 1:size(Ws_dirs,2)
    v = Ws_dirs(:,k);
    fprintf('  Ws_dir(%d):  %12.8e  %12.8e  %12.8e  %12.8e  %12.8e  %12.8e\n', ...
          k, v(1), v(2), v(3), v(4), v(5), v(6));
  end
end
end  % if converged

%% SECTION 16: ROTATING FRAME TRAJECTORY PLOT
─────────────────────────────────────────────────────────
% Plots the capture orbit pump-down trajectory in the CR3BP synodic
```

```
% rotating frame from JOI capture state through full pump-down duration.
%
% Figure 16a — zoomed to ±3 nd (inner region, scientifically relevant)
% Figure 16b — full extent (apojove radius, shows complete rosette pattern)
%
% The rosette pattern is physically correct: the synodic frame rotates
% with Io while the capture orbit period is much longer, so each perijove
% passage appears at a different orientation.
% Color progression blue→green→orange shows orbital decay over time.
if ~converged
  fprintf('\nSkipping rotating frame plot — Lyapunov orbit not converged.\n');
else
fprintf('\n── Rotating Frame Trajectory Plot ──────────────────────────────────\n');
% ── Integrate capture orbit through full pump-down ──────────────────────
p_cap_plt       = p;
p_cap_plt.edt_on = true;
odefun_cap_plt  = @(t,Y) cr3bp_eom_nd_edt(t, Y, p_cap_plt);
opts_cap_plt    = odeset('RelTol',1e-10,'AbsTol',1e-12);
n_plt_pts = 50000;
fprintf('  Integrating %.0f days at %d points...\n', ...
      T_pump_actual/n/86400, n_plt_pts);
[~, Y_cap_plt] = ode113(odefun_cap_plt, ...
  linspace(0, T_pump_actual, n_plt_pts), Y0_cap, opts_cap_plt);
r_cap_nd = Y_cap_plt(:, 1:3);
fprintf('  Integration complete: %d points\n', size(Y_cap_plt,1));
% Color-code by time thirds
N_plt   = size(r_cap_nd, 1);
i_third = floor(N_plt/3);
i_two   = floor(2*N_plt/3);
col_early = [0.2 0.6 1.0];
col_mid   = [0.2 1.0 0.6];
col_late  = [1.0 0.6 0.2];
n_orbits_actual = round(T_pump_actual / T_orbit_nd);
th = linspace(0, 2*pi, 300);
for fig_idx = 1:2
  if fig_idx == 1
      fig_name      = 'Rotating Frame — Inner Region (zoomed)';
      xlim_val      = 3.0;
      ylim_val      = 3.0;
      fig_title_line2 = sprintf(...
```

```
        'Zoomed view \\pm3 nd (\\pm%.0f km)  |  %.0f orbits  |  %.0f days', ...
        3.0*Lstar, n_orbits_actual, T_pump_actual/n/86400);
else
    fig_name      = 'Rotating Frame — Full Extent';
    r_a_nd_ref    = r_a_joi / Lstar;
    xlim_val      = r_a_nd_ref * 1.05;
    ylim_val      = r_a_nd_ref * 1.05;
    fig_title_line2 = sprintf(...
        'Full extent (%.0f R_J apojove)  |  rosette from synodic frame rotation', ...
        r_a_RJ);
end
figure('Color',FIG_BG, 'Name',fig_name);
hold on; grid on; box on;
set(gca, 'Color',AX_BG, 'XColor',AX_COL, 'YColor',AX_COL, ...
         'GridColor',GRID_COL, 'FontSize',FONT_SM);
plot(r_cap_nd(1:i_third,1),    r_cap_nd(1:i_third,2), ...
    '-','Color',col_early,'LineWidth',0.6,'DisplayName','Early pump-down');
plot(r_cap_nd(i_third:i_two,1), r_cap_nd(i_third:i_two,2), ...
    '-','Color',col_mid,  'LineWidth',0.6,'DisplayName','Mid pump-down');
plot(r_cap_nd(i_two:end,1),    r_cap_nd(i_two:end,2), ...
    '-','Color',col_late, 'LineWidth',0.6,'DisplayName','Late pump-down');
plot(r_cap_nd(1,1), r_cap_nd(1,2), 'ws', ...
    'MarkerSize',10,'MarkerFaceColor','w','LineWidth',1.5,...
    'DisplayName','JOI entry (t=0)');
plot(r_cap_nd(end,1), r_cap_nd(end,2), 'y*', ...
    'MarkerSize',14,'LineWidth',2,...
    'DisplayName',sprintf('Pump-down end (t=%.0f days)', T_pump_actual/n/86400));
plot(-mu,  0, 'o','Color','w','MarkerFaceColor',[0.9 0.6 0.2],...
    'MarkerSize',14,'DisplayName','Jupiter');
plot(1-mu, 0, 'o','Color','w','MarkerFaceColor',[0.9 0.8 0.3],...
    'MarkerSize', 9,'DisplayName','Io');
hL12  = plot(Lpts(1:2,1), Lpts(1:2,2), '+','Color','yellow',...
          'MarkerSize',14,'LineWidth',2.5);
set(hL12,  'DisplayName','L_1, L_2');
hL345 = plot(Lpts(3:5,1), Lpts(3:5,2), 'x','Color',[0.6 0.6 0.3],...
          'MarkerSize',10,'LineWidth',1.5);
set(hL345, 'DisplayName','L_3, L_4, L_5');
for kk = 1:5
    if abs(Lpts(kk,1)) <= xlim_val && abs(Lpts(kk,2)) <= ylim_val
        text(Lpts(kk,1)+0.03, Lpts(kk,2)+0.05, ...
```

```
            sprintf('L_%d',kk),'Color','yellow','FontSize',FONT_SM);
    end
  end
  plot(cos(th), sin(th), '--','Color',[0.8 0.6 0.1],'LineWidth',1.2,...
      'DisplayName','Io orbit (r^* = 1 nd)');
  r_p_nd_ref = r_p_joi / Lstar;
  plot(r_p_nd_ref*cos(th), r_p_nd_ref*sin(th), ':','Color',[0.4 0.8 0.4],...
      'LineWidth',1.0,'DisplayName',sprintf('Perijove (%.0f R_J)',r_p_RJ));
  xlabel(sprintf('x^*  (normalized, synodic)  [1 nd = %.0f km]', Lstar), ...
       'Color',AX_COL,'FontSize',FONT_MD);
  ylabel(sprintf('y^*  (normalized, synodic)  [1 nd = %.0f km]', Lstar), ...
       'Color',AX_COL,'FontSize',FONT_MD);
  title({'CR3BP-LT Capture Orbit Pump-Down — Synodic Rotating Frame', ...
       fig_title_line2}, ...
      'Color',AX_COL,'FontSize',FONT_SM,'Interpreter','tex');
  legend('show','TextColor',AX_COL,'Color',[0.12 0.12 0.12],...
       'Location','best','FontSize',FONT_SM);
  text(-xlim_val*0.97, -ylim_val*0.93, ...
      sprintf('Perijove: %.0f km (%.0f R_J)  |  Apojove: %.0f km (%.0f R_J)', ...
           r_p_joi, r_p_RJ, r_a_joi, r_a_RJ), ...
      'Color',[0.6 0.6 0.6],'FontSize',FONT_SM);
  axis equal;
  xlim([-xlim_val, xlim_val]);
  ylim([-ylim_val, ylim_val]);
end
fprintf('  Figure 16a (zoomed ±3 nd) and Figure 16b (full extent) complete.\n');
end  % if converged

%% SECTION 17: JACOBI CONSTANT CHECK
% ------------------------------------------------------------------------
% Validates the Section 5 test integration using the Jacobi constant.
%   Free CR3BP  (edt_on=false): C constant  → confirms integrator accuracy
%   CR3BP-LT    (edt_on=true):  C decreases → confirms EDT is doing work
%   C flat with edt_on=true:    EDT force is zero → check current formula
%
% Uses r_nd and v_nd from Section 5 (test orbit, not the capture orbit).
% Vectorized Jacobi computation over full trajectory
x  = r_nd(:,1);  y  = r_nd(:,2);  z  = r_nd(:,3);
vx = v_nd(:,1);  vy = v_nd(:,2);  vz = v_nd(:,3);
r1 = sqrt((x+mu).^2   + y.^2 + z.^2);
```

```
r2 = sqrt((x-1+mu).^2 + y.^2 + z.^2);
U  = 0.5*(x.^2+y.^2) + (1-mu)./r1 + mu./r2;
C_history = 2*U - (vx.^2+vy.^2+vz.^2);
fprintf('\n── Jacobi Constant Check
──────────────────────────────────────────────────────────\n');
fprintf('  C_initial = %.8f\n', C_history(1));
fprintf('  C_final   = %.8f\n', C_history(end));
fprintf('  Delta_C   = %+.4e  ', C_history(end) - C_history(1));
if p.edt_on
  if abs(C_history(end) - C_history(1)) > 1e-6
    fprintf('✓  EDT is doing work (C drifting as expected)\n');
  else
    fprintf('✗  C is flat — EDT force may be zero, check current formula\n');
  end
else
  if abs(C_history(end) - C_history(1)) < 1e-6
    fprintf('✓  Free CR3BP — C conserved (integrator accurate)\n');
  else
    fprintf('✗  C drift in free CR3BP — integrator tolerance issue\n');
  end
end
% ── Figure 17: Jacobi constant history
──────────────────────────────────────────────
figure('Color',FIG_BG, 'Name','Jacobi Constant History');
plot(tau/(2*pi), C_history, 'c-', 'LineWidth',1.8);
xlabel('Time  [Io periods]',      'Color',AX_COL, 'FontSize',FONT_MD);
ylabel('Jacobi constant  C  [nd]', 'Color',AX_COL, 'FontSize',FONT_MD);
title({sprintf('Jacobi Constant History — Test Integration (%d Io periods)', nRevs), ...
    'Monotonic decrease confirms EDT Lorentz force is physically active'}, ...
    'Color',AX_COL, 'FontSize',FONT_MD, 'Interpreter','none');
set(gca, 'Color',AX_BG, 'XColor',AX_COL, 'YColor',AX_COL, ...
      'GridColor',GRID_COL, 'FontSize',FONT_SM);
grid on;
dC_total = C_history(end) - C_history(1);
dC_rate  = dC_total / nRevs;
text(0.3, C_history(end) + abs(dC_total)*0.10, ...
   sprintf('\\DeltaC = %+.4e nd over %d Io periods', dC_total, nRevs), ...
   'Color','w', 'FontSize',FONT_SM);
text(0.3, C_history(end) + abs(dC_total)*0.04, ...
   sprintf('Rate: %.4e nd per Io period', dC_rate), ...
```

```
    'Color',[0.8 0.8 0.8], 'FontSize',FONT_SM);
assignin('base', 'C_history', C_history);

%% SECTION 18: BACK-TRANSFORM TO JUPITER-CENTERED INERTIAL
% ──────────────────────────
if exportInertialBack
fprintf('\n── Jupiter-Centered Inertial Frame Plot ──────────────────────────\n');
% ── Time-varying back-transform
% ──────────────────────────────────────────────────────────────────
% The rotating frame angle advances as theta(t) = n*t.
% Must rotate by the current frame angle at each timestep —
% NOT the fixed t=0 rotation matrix from Section 3.
t_plt  = linspace(0, T_pump_actual, 50000)';
N_jci  = length(t_plt);
r_JCI_cap = zeros(N_jci, 3);
v_JCI_cap = zeros(N_jci, 3);
for k = 1:N_jci
   theta_k = t_plt(k);
   ct  = cos(theta_k);
   st  = sin(theta_k);
   R_t = [ct -st 0; st ct 0; 0 0 1];
   r_rot_k       = Y_cap_plt(k, 1:3)' * Lstar;
   r_iner_nobar  = R_t * r_rot_k;
   r_JCI_cap(k,:) = (r_iner_nobar + rBary_JCI)';
   v_rot_k       = Y_cap_plt(k, 4:6)' * Vstar;
   v_iner_k      = R_t * v_rot_k + cross([0;0;n], r_iner_nobar) + vBary_JCI;
   v_JCI_cap(k,:) = v_iner_k';
end
r_mag = sqrt(sum(r_JCI_cap.^2, 2));
r_max = max(r_mag);
r_min = min(r_mag);
fprintf('  JCI orbit extent: r_min = %.0f km (%.1f R_J)  r_max = %.0f km (%.1f R_J)\n', ...
      r_min, r_min/p.R_J_km, r_max, r_max/p.R_J_km);
fprintf('  Back-transform complete: %d points\n', N_jci);
% ── Figure 18: JCI capture orbit
% ──────────────────────────────────────────────────────────────
figure('Color',FIG_BG, 'Name','Jupiter-Centered Inertial — Capture Orbit');
hold on; grid on; box on;
set(gca, 'Color',AX_BG, 'XColor',AX_COL, 'YColor',AX_COL, ...
      'GridColor',GRID_COL, 'FontSize',FONT_SM);
```

```
i3_jci  = floor(N_jci/3);
i23_jci = floor(2*N_jci/3);
plot(r_JCI_cap(1:i3_jci,1),       r_JCI_cap(1:i3_jci,2), ...
   '-','Color',[0.2 0.6 1.0],'LineWidth',0.6,...
   'DisplayName','Capture orbit: early pump-down');
plot(r_JCI_cap(i3_jci:i23_jci,1), r_JCI_cap(i3_jci:i23_jci,2), ...
   '-','Color',[0.2 1.0 0.6],'LineWidth',0.6,...
   'DisplayName','Capture orbit: mid pump-down');
plot(r_JCI_cap(i23_jci:end,1),    r_JCI_cap(i23_jci:end,2), ...
   '-','Color',[1.0 0.6 0.2],'LineWidth',0.6,...
   'DisplayName','Capture orbit: late pump-down');
plot(r_JCI_cap(1,1),   r_JCI_cap(1,2),   'ws',...
   'MarkerSize',10,'MarkerFaceColor','w','LineWidth',1.5,...
   'DisplayName','JOI entry (t=0)');
plot(r_JCI_cap(end,1), r_JCI_cap(end,2), 'y*',...
   'MarkerSize',14,'LineWidth',2,...
   'DisplayName',sprintf('Pump-down end (t=%.0f days)', T_pump_actual/n/86400));
plot(0, 0, 'o','Color','w','MarkerFaceColor',[0.9 0.6 0.2],...
   'MarkerSize',16,'DisplayName','Jupiter');
th      = linspace(0, 2*pi, 300);
r_Io_km = norm(rIo_JCI);
plot(r_Io_km*cos(th), r_Io_km*sin(th), '--',...
   'Color',[0.8 0.7 0.2],'LineWidth',1.2,...
   'DisplayName',sprintf('Io orbit (%.0f km = %.1f R_J)', r_Io_km, r_Io_km/p.R_J_km));
plot(r_p_joi*cos(th), r_p_joi*sin(th), ':',...
   'Color',[0.4 0.8 0.4],'LineWidth',1.0,...
   'DisplayName',sprintf('Perijove (%.0f R_J)', r_p_RJ));
plot(r_a_joi*cos(th), r_a_joi*sin(th), ':',...
   'Color',[0.4 0.4 0.8],'LineWidth',0.8,...
   'DisplayName',sprintf('Apojove (%.0f R_J)', r_a_RJ));
xlabel('JCI X  [km]','Color',AX_COL,'FontSize',FONT_MD);
ylabel('JCI Y  [km]','Color',AX_COL,'FontSize',FONT_MD);
title({'Capture Orbit Pump-Down — Jupiter-Centered Inertial Frame', ...
    sprintf('%.0f\\times%.0f R_J ellipse  \\rightarrow  EDT pump-down  \\rightarrow  %.0f days
|  %.0f orbits', ...
          r_p_RJ, r_a_RJ, T_pump_actual/n/86400, round(T_pump_actual/T_orbit_nd))}, ...
   'Color',AX_COL,'FontSize',FONT_SM,'Interpreter','tex');
legend('show','TextColor',AX_COL,'Color',[0.12 0.12 0.12],...
    'Location','best','FontSize',FONT_SM);
axis equal;
```

```
lim_jci = r_a_joi * 1.1;
xlim([-lim_jci, lim_jci]);
ylim([-lim_jci, lim_jci]);
text(-lim_jci*0.97, -lim_jci*0.93, ...
   sprintf('Perijove: %.0f km (%.0f R_J)  |  Apojove: %.0f km (%.0f R_J)  |  Period: %.0f days',
...
           r_p_joi, r_p_RJ, r_a_joi, r_a_RJ, T_orbit_nd/n/86400), ...
   'Color',[0.6 0.6 0.6],'FontSize',FONT_SM);
% Export
assignin('base', 'r_JCI_cap',         r_JCI_cap);
assignin('base', 'v_JCI_cap',         v_JCI_cap);
assignin('base', 'r_JCI',             r_JCI_cap);
assignin('base', 'Lagrange_points_nd', Lpts);
fprintf('  JCI figure complete.\n');
%% SECTION 16b: RADIATION DOSE ESTIMATE
─────────────────────────────────────────────────────────
% Uses Y_cap_plt from Section 16 — must run after that section.
% Simplified TID model: D_rate ~ D0 * (R_ref/R)^alpha
% Anchored to Europa Orbiter heritage: ~100 krad at 8 R_J behind 1 inch Al.
% Produces cumulative dose figure and perijove sweep dose scaling table.
if ~converged
   fprintf('\nSkipping radiation estimate — no converged orbit.\n');
elseif ~exist('Y_cap_plt','var') || isempty(Y_cap_plt)
   fprintf('\nSkipping radiation estimate — Y_cap_plt not available.\n');
else
fprintf('\n── Radiation Dose Estimate
──────────────────────────────────────────────────────\n');
% ── Dose rate parameters
──────────────────────────────────────────────────────────────────────
D0_krad_per_day = 100.0;          % Reference dose rate [krad/day] at R_ref
R_ref_km        = 8.0 * p.R_J_km; % Reference radius: 8 R_J (Europa Orbiter)
alpha_dose      = 4.0;            % Trapped electron flux exponent
% ── Radius history from pump-down trajectory
──────────────────────────────────────────────
% Jupiter is at x = -mu in normalized synodic coordinates
x_hist    = Y_cap_plt(:,1);
y_hist    = Y_cap_plt(:,2);
z_hist    = Y_cap_plt(:,3);
r_hist_nd = sqrt((x_hist + mu).^2 + y_hist.^2 + z_hist.^2);
r_hist_km = r_hist_nd * Lstar;
```

```
% ── Dose rate and cumulative TID
──────────────────────────────────────────────────────────────────

dose_rate_hist = D0_krad_per_day * (R_ref_km ./ r_hist_km).^alpha_dose;
t_days_hist    = linspace(0, T_pump_actual/n/86400, size(Y_cap_plt,1));
TID_cumulative = cumtrapz(t_days_hist, dose_rate_hist);
TID_krad       = TID_cumulative(end);
TID_Mrad       = TID_krad / 1000;
TID_per_orbit  = TID_krad / (T_pump_actual / T_orbit_nd);
frac_inner = mean(r_hist_km < 2.0 * r_p_joi);
fprintf('  Pump-down trajectory TID estimate:\n');
fprintf('    Total dose:              %.1f krad  (%.3f Mrad)\n', TID_krad, TID_Mrad);
fprintf('    Average per orbit:       %.1f krad/orbit\n',        TID_per_orbit);
fprintf('    Time fraction < 2*r_p:   %.1f%%\n',                 100*frac_inner);
fprintf('\n  Heritage comparison:\n');
fprintf('    Europa Orbiter budget:   ~100 krad (1 inch Al shielding)\n');
fprintf('    Galileo total (8 years): ~650 krad (100 mil Al)\n');
fprintf('    UNAGI pump-down:         %.1f krad\n',              TID_krad);
fprintf('    UNAGI / Europa Orbiter:  %.1fx\n',                  TID_krad/100);
fprintf('\n  NOTE: Simplified dipole dose model. Actual TID depends on\n');
fprintf('  shielding thickness, particle spectrum, and inclination.\n');
fprintf('  Phase B requires full GMAT/STK radiation environment analysis.\n');
% ── Dose vs perijove radius scaling table
──────────────────────────────────────────────────────

if exist('r_p_sweep_RJ','var') && exist('T_pump_rp','var')
  fprintf('\n  Dose scaling across perijove sweep:\n');
  fprintf('  %-10s  %-14s  %-14s  %-12s\n', ...
       'r_p [R_J]','T_pump [days]','TID [krad]','vs baseline');
  fprintf('  %-10s  %-14s  %-14s  %-12s\n', ...
       '---------','-------------','---------','-----------');
  T_baseline_rad = T_pump_actual/n/86400;
  for ii = 1:length(r_p_sweep_RJ)
    if ~isnan(T_pump_rp(ii))
      r_p_ii_km  = r_p_sweep_RJ(ii) * p.R_J_km;
      dose_scale = (R_ref_km / r_p_ii_km)^alpha_dose;
      TID_ii     = TID_krad * dose_scale * ...
                   (T_pump_rp(ii) / T_baseline_rad);
      fprintf('  %-10.0f  %-14.0f  %-14.1f  %-12.2fx\n', ...
            r_p_sweep_RJ(ii), T_pump_rp(ii), TID_ii, TID_ii/TID_krad);
    end
  end
```

```
end
% ── Figure: cumulative TID vs mission time ──────────────────────────
fig_dose = figure('Color',FIG_BG,'Name','Radiation: TID Accumulation');
hold on; grid on; box on;
set(gca,'Color',AX_BG,'XColor',AX_COL,'YColor',AX_COL,...
  'GridColor',GRID_COL,'FontSize',FONT_SM);
plot(t_days_hist, TID_cumulative, 'r-', 'LineWidth',2.5,...
   'DisplayName','Cumulative TID — pump-down trajectory');
yline(100, '--','Color',[1.0 0.8 0.2],'LineWidth',2.0,...
    'Label','Europa Orbiter budget (100 krad)',...
    'LabelColor',[1.0 0.8 0.2],'FontSize',FONT_SM,...
    'DisplayName','Europa Orbiter budget');
yline(650, '--','Color',[1.0 0.4 0.4],'LineWidth',1.5,...
    'Label','Galileo total (650 krad)',...
    'LabelColor',[1.0 0.4 0.4],'FontSize',FONT_SM,...
    'DisplayName','Galileo total');
text(T_pump_actual/n/86400 * 0.5, TID_krad * 0.6, ...
   sprintf('Total TID = %.1f krad\n(%.1fx Europa Orbiter budget)', ...
        TID_krad, TID_krad/100), ...
   'Color','w','FontSize',FONT_SM,'HorizontalAlignment','center');
xlabel('Mission elapsed time  [days]','Color',AX_COL,'FontSize',FONT_MD);
ylabel('Cumulative TID  [krad]',     'Color',AX_COL,'FontSize',FONT_MD);
title({'Radiation Dose Accumulation During EDT Pump-Down', ...
    sprintf('Simplified model: D \\propto (R_{ref}/R)^{%.0f}  |  R_{ref} = %.0f R_J  |  %.0f
days total', ...
          alpha_dose, R_ref_km/p.R_J_km, T_pump_actual/n/86400)}, ...
   'Color',AX_COL,'FontSize',FONT_SM,'Interpreter','tex');
legend('show','TextColor',AX_COL,'Color',[0.12 0.12 0.12],...
    'Location','northwest','FontSize',FONT_SM);
% Export
assignin('base','TID_krad',      TID_krad);
assignin('base','TID_cumulative', TID_cumulative);
assignin('base','t_days_hist',    t_days_hist);
assignin('base','dose_rate_hist', dose_rate_hist);
fprintf('  Radiation dose estimate complete.\n');
end  % if converged and Y_cap_plt exists
end  % if exportInertialBack
```

```
%% SECTION 19: ANIMATION — JCI CAPTURE ORBIT PUMP-DOWN (OPTIONAL)
──────────────────
% Animates the EDT pump-down spiral in the Jupiter-centered inertial frame.
% Each frame advances one capture orbit period.
% Export as GIF for thesis appendix or defense presentation.
%
% Requires exportInertialBack = true and converged = true.
if converged && exportInertialBack && exist('r_JCI_cap','var') && ~isempty(r_JCI_cap)
n_orbits      = round(T_pump_actual / T_orbit_nd);
pts_per_orbit = floor(size(r_JCI_cap,1) / n_orbits);
fprintf('\n── Capture Orbit Animation
──────────────────────────────────────────────────────\n');
fprintf('  %d orbits  |  %.0f days  |  %d points per orbit\n', ...
      n_orbits, T_pump_actual/n/86400, pts_per_orbit);
fig_anim = figure('Color',FIG_BG, 'Name','Animation — JCI Pump-Down', ...
                  'Position',[100 100 800 800]);
hold on; grid on; box on;
set(gca, 'Color',AX_BG, 'XColor',AX_COL, 'YColor',AX_COL, ...
       'GridColor',GRID_COL, 'FontSize',FONT_SM);
% Static reference elements — drawn once
th      = linspace(0, 2*pi, 300);
r_Io_km = norm(rIo_JCI);
plot(r_Io_km*cos(th), r_Io_km*sin(th), '--', ...
   'Color',[0.8 0.7 0.2], 'LineWidth',1.0, 'HandleVisibility','off');
plot(r_p_joi*cos(th), r_p_joi*sin(th), ':', ...
   'Color',[0.4 0.8 0.4], 'LineWidth',0.8, 'HandleVisibility','off');
plot(0, 0, 'o', 'Color','w', 'MarkerFaceColor',[0.9 0.6 0.2], ...
   'MarkerSize',16, 'HandleVisibility','off');
axis equal;
lim_anim = r_a_joi * 1.1;
xlim([-lim_anim, lim_anim]);
ylim([-lim_anim, lim_anim]);
xlabel('JCI X  [km]', 'Color',AX_COL, 'FontSize',FONT_MD);
ylabel('JCI Y  [km]', 'Color',AX_COL, 'FontSize',FONT_MD);
cmap         = cool(n_orbits);
gif_filename = 'UNAGI_pump_down_animation.gif';
save_gif     = true;   % set false to display only without saving
for orb = 1:n_orbits
  i_start = (orb-1)*pts_per_orbit + 1;
  i_end   = min(orb*pts_per_orbit, size(r_JCI_cap,1));
```

```
  % Draw this orbit's trace
  plot(r_JCI_cap(i_start:i_end,1), r_JCI_cap(i_start:i_end,2), ...
      '-', 'Color',cmap(orb,:), 'LineWidth',0.8, 'HandleVisibility','off');
  % Moving spacecraft marker
  h_sc = plot(r_JCI_cap(i_end,1), r_JCI_cap(i_end,2), 'wo', ...
            'MarkerSize',8, 'MarkerFaceColor','w', 'HandleVisibility','off');
  % Frame title — uses pump_table_C_dyn (always defined after Section 8)
  t_frame_days = (i_end/size(r_JCI_cap,1)) * T_pump_actual/n/86400;
  C_frame      = pump_table_C_dyn(min(orb, length(pump_table_C_dyn)));
  title({sprintf('UNAGI Capture Orbit Pump-Down — Orbit %d of %d', orb, n_orbits), ...
        sprintf('t = %.0f days  |  C = %.4f nd', t_frame_days, C_frame)}, ...
       'Color',AX_COL, 'FontSize',FONT_SM, 'Interpreter','none');
  drawnow;
  if save_gif
     frame      = getframe(fig_anim);
     img        = frame2im(frame);
     [imind, cm] = rgb2ind(img, 256);
     if orb == 1
        imwrite(imind, cm, gif_filename, 'gif', ...
              'Loopcount',inf, 'DelayTime',0.35);
     else
        imwrite(imind, cm, gif_filename, 'gif', ...
              'WriteMode','append', 'DelayTime',0.35);
     end
  end
  delete(h_sc);
end
% Final frame — mark pump-down endpoint
plot(r_JCI_cap(end,1), r_JCI_cap(end,2), 'y*', ...
   'MarkerSize',16, 'LineWidth',2, 'HandleVisibility','off');
title({'UNAGI Capture Orbit Pump-Down — Complete', ...
      sprintf('%.0f days  |  %d orbits  |  \\DeltaC = %.3f nd  |  0 kg propellant', ...
            T_pump_actual/n/86400, n_orbits, C_cap-C_pump_end)}, ...
    'Color',AX_COL, 'FontSize',FONT_SM, 'Interpreter','tex');
drawnow;
if save_gif
  frame      = getframe(fig_anim);
  img        = frame2im(frame);
  [imind, cm] = rgb2ind(img, 256);
  imwrite(imind, cm, gif_filename, 'gif', ...
```

```
        'WriteMode','append', 'DelayTime',3.0);
    fprintf('  Animation saved: %s\n', gif_filename);
end
fprintf('  Animation complete: %d frames  |  %.0f days  |  %d orbits\n', ...
      n_orbits, T_pump_actual/n/86400, n_orbits);
end  % if converged && exportInertialBack
end  % main function

%% ── LOCAL FUNCTIONS ──────────────────────────────────────────────────────

function Lpts = compute_lagrange_points(mu)
% Returns all 5 Lagrange point locations in normalized synodic coordinates.
% L1, L2, L3: solved numerically along x-axis (collinear points)
% L4, L5:     computed analytically (equilateral triangle geometry)
Uderiv = @(x) x - (1-mu).*(x+mu)./((x+mu).^2).^(3/2) ...
              - mu.*(x-1+mu)./((x-1+mu).^2).^(3/2);
d     = max(1e-3, (mu/3)^(1/3));
x_sec = 1 - mu;
xL1 = fzero(Uderiv, [x_sec - 2*d, x_sec - 1e-6]);
xL2 = fzero(Uderiv, [x_sec + 1e-6, x_sec + 2*d]);
xL3 = fzero(Uderiv, [-mu - 2, -mu - 0.1]);
xL4 =  0.5 - mu;  yL4 =  sqrt(3)/2;
xL5 =  0.5 - mu;  yL5 = -sqrt(3)/2;
Lpts = [xL1, 0,   0;
       xL2, 0,   0;
       xL3, 0,   0;
       xL4, yL4, 0;
       xL5, yL5, 0];
end

function dY = cr3bp_eom_nd_edt(~, Y, p)
% CR3BP equations of motion with optional EDT Lorentz acceleration.
% State: Y = [x y z xd yd zd]^T (normalized, synodic frame)
x  = Y(1);  y  = Y(2);  z  = Y(3);
xd = Y(4);  yd = Y(5);  zd = Y(6);
mu    = p.mu;
n     = p.n;
Lstar = p.Lstar;
Vstar = p.Vstar;
r1 = sqrt((x + mu)^2     + y^2 + z^2);
```

```
r2 = sqrt((x - 1 + mu)^2 + y^2 + z^2);
xdd =  2*yd + x - (1-mu)*(x+mu)/r1^3 - mu*(x-1+mu)/r2^3;
ydd = -2*xd + y - (1-mu)*y/r1^3       - mu*y/r2^3;
zdd =           -(1-mu)*z/r1^3       - mu*z/r2^3;
if p.edt_on && p.l_m > 0 && p.mSC_kg > 0
  r_S_km    = [x; y; z]    * Lstar;
  v_S_kmps  = [xd; yd; zd] * Vstar;
  rJup_S_km = [-mu; 0; 0]  * Lstar;
  r_rel_km  = r_S_km - rJup_S_km;
  R_km      = norm(r_rel_km);
  if R_km > 0
    omega     = [0; 0; n];
    v_rel_kmps = v_S_kmps + cross(omega, r_rel_km);
    v_rel_mps  = v_rel_kmps * 1e3;
    v_rel_mag  = norm(v_rel_mps);
    % Dipole B-field: calibrated to 1800 nT at Io's orbital radius
    % Lstar = norm(rIo_JCI) by construction, so this anchors the
    % dipole at the correct reference distance
    B_eq_T = 1800e-9 * (Lstar / p.R_J_km)^3;
    if isfield(p, 'B_scale')
      B_eq_T = B_eq_T * p.B_scale;
    end
    B_T = B_eq_T * (p.R_J_km / R_km)^3;
    EMF = B_T * p.l_m * v_rel_mag;
    if EMF > 1e-10
      I_A = min(p.P_max_W / EMF, p.I_max_A);
    else
      I_A = 0;
    end

    % Plasma density availability factor
    % Physical basis: bare tether current collection requires sufficient ambient
    % electron density Ne to close the circuit. Inside the Io plasma torus
    % (5-8 R_J) Ne is high; outside it drops sharply.
    % Model: Ne ~ (R_Io/R)^1.5 inside torus, floor outside — Sanmartin 2008 [7]
    % Flag: p.plasma_scaling = true/false (default false = always-on baseline)
    if isfield(p, 'plasma_scaling') && p.plasma_scaling
      R_torus_km  = 10.0 * p.R_J_km;   % Io torus outer edge ~10 R_J
      R_Io_km_ref = Lstar;             % Io orbital radius = Lstar by construction
      if R_km <= R_torus_km
```

```
            % Inside torus: density falls as r^-1.5 from Io reference
            plasma_factor = min(1.0, (R_Io_km_ref / R_km)^1.5);
        else
            % Outside torus: sharp dropoff, floor at 10 R_J value × 0.1
            Ne_at_torus   = (R_Io_km_ref / R_torus_km)^1.5;
            plasma_factor = Ne_at_torus * 0.1;
        end
        plasma_factor = max(plasma_factor, 0.01);  % floor at 1% — circuit never
                                                   % fully open but nearly zero
        I_A = I_A * plasma_factor;
    end
    F_N       = I_A * p.l_m * B_T;
    a_mps2_mag = F_N / p.mSC_kg;
    if v_rel_mag > 1e-12
        a_vec_mps2 = a_mps2_mag * -(v_rel_mps / v_rel_mag);
    else
        a_vec_mps2 = [0; 0; 0];
    end
    a_nd = (a_vec_mps2 / 1e3) / (Lstar * n^2);
  else
    a_nd = [0; 0; 0];
  end
  % Apply duty cycle — scales effective EDT acceleration
  % Conservative model: duty cycle applied as time-averaged force reduction
  % Full implementation would require switching logic; this captures mean effect
  duty = 1.0;
  if isfield(p, 'duty_cycle')
    duty = p.duty_cycle;
  end
  xdd = xdd + duty * a_nd(1);
  ydd = ydd + duty * a_nd(2);
  zdd = zdd + duty * a_nd(3);
end
dY = [xd; yd; zd; xdd; ydd; zdd];
end

function dY = cr3bp_variational_nd(~, Y, p)
% Variational equations: integrates state + STM simultaneously.
% Y = [x; y; z; xd; yd; zd; Phi(36x1)] — 42 elements total
% Phi_dot = A(t) * Phi
```

```
mu = p.mu;
x = Y(1); y = Y(2); z = Y(3);
xd = Y(4); yd = Y(5); zd = Y(6);
r1 = sqrt((x+mu)^2   + y^2 + z^2);
r2 = sqrt((x-1+mu)^2 + y^2 + z^2);
Uxx = 1 - (1-mu)*(1/r1^3 - 3*(x+mu)^2/r1^5)     - mu*(1/r2^3 - 3*(x-1+mu)^2/r2^5);
Uyy = 1 - (1-mu)*(1/r1^3 - 3*y^2/r1^5)          - mu*(1/r2^3 - 3*y^2/r2^5);
Uzz =    -(1-mu)*(1/r1^3 - 3*z^2/r1^5)          - mu*(1/r2^3 - 3*z^2/r2^5);
Uxy = 3*((1-mu)*(x+mu)*y/r1^5   + mu*(x-1+mu)*y/r2^5);
Uxz = 3*((1-mu)*(x+mu)*z/r1^5   + mu*(x-1+mu)*z/r2^5);
Uyz = 3*((1-mu)*y*z/r1^5        + mu*y*z/r2^5);
A = [ 0    0    0    1   0   0;
    0    0    0    0   1   0;
    0    0    0    0   0   1;
    Uxx  Uxy  Uxz  0   2   0;
    Uxy  Uyy  Uyz -2   0   0;
    Uxz  Uyz  Uzz  0   0   0 ];
Phi     = reshape(Y(7:42), 6, 6);
Phi_dot = A * Phi;
xdot = [xd; yd; zd;
      2*yd + x - (1-mu)*(x+mu)/r1^3 - mu*(x-1+mu)/r2^3;
     -2*xd + y - (1-mu)*y/r1^3      - mu*y/r2^3;
             -(1-mu)*z/r1^3        - mu*z/r2^3];
dY = [xdot; Phi_dot(:)];
end

function [Y0_lyap, T_half, converged] = differential_corrector_lyapunov(...
  xL1, A_x, mu, tol, max_iter, p, T_guess_in)
% Single-shooter differential corrector for planar Lyapunov orbit at L1.
% Exploits x-axis symmetry: start on x-axis with xd=0, enforce xd=0
% at next y=0 crossing (half-period condition).
% Free variable: x0.  Constraint: xd_f = 0.  Update: Newton step via STM.
n = p.n;   % Only n is needed from p — Lstar/Vstar not used in corrector
r1L = abs(xL1 + mu);
r2L = abs(xL1 - 1 + mu);
Uxx_L = 1 - (1-mu)*(1/r1L^3 - 3*(xL1+mu)^2/r1L^5) ...
        - mu*(1/r2L^3 - 3*(xL1-1+mu)^2/r2L^5);
Uyy_L = 1 - (1-mu)/r1L^3 - mu/r2L^3;
b_coef = 2 - Uxx_L - Uyy_L;
c_coef = Uxx_L * Uyy_L - 1;
```

```
disc   = b_coef^2 - 4*c_coef;
if disc < 0
  warning('DC: negative discriminant at L1 — check mu');
  Y0_lyap = zeros(6,1);  T_half = pi;  converged = false;  return
end
lambda_sq = (-b_coef - sqrt(disc)) / 2;
if lambda_sq >= 0
  warning('DC: L1 linearization gives real frequency — unexpected');
  Y0_lyap = zeros(6,1);  T_half = pi;  converged = false;  return
end
omega_lyap = sqrt(-lambda_sq);
if nargin < 7 || isempty(T_guess_in)
  T_guess = pi / omega_lyap;
else
  T_guess = T_guess_in;
end
kappa   = (omega_lyap^2 - Uxx_L) / (2 * omega_lyap);
x0      = xL1 + A_x;
yd0     = -kappa * A_x;
Y0_lyap = [x0; 0; 0; 0; yd0; 0];
fprintf('  Initial guess: x0=%.8f  yd0=%.8f\n', x0, yd0);
fprintf('  ω_Lyapunov = %.6f nd   T_half_guess = %.6f nd\n', omega_lyap, T_guess);
converged = false;
T_half    = T_guess;
opts_dc = odeset('RelTol',1e-12,'AbsTol',1e-14, ...
                 'Events',@(t,Y) y_crossing_event(t, Y));
for iter = 1:max_iter
  Phi0_flat = eye(6);
  Ystm0 = [Y0_lyap; Phi0_flat(:)];
  [~, Ystm_sol, t_events, ~, ~] = ode113(...
     @(t,Y) cr3bp_variational_nd(t, Y, p), [0, 2*T_half], Ystm0, opts_dc);
  if isempty(t_events)
     [~, Ystm_sol] = ode113(...
        @(t,Y) cr3bp_variational_nd(t, Y, p), [0, 3*T_half], Ystm0, opts_dc);
  end
  Yf   = Ystm_sol(end, 1:6)';
  Phif = reshape(Ystm_sol(end, 7:42)', 6, 6);
  xd_f = Yf(4);
  fprintf('  Iter %2d:  x0=%.10f  xd_f=%+.4e\n', iter, Y0_lyap(1), xd_f);
  if abs(xd_f) < tol
```

```
        converged = true;
        T_half = Ystm_sol(end, 1);
        if ~isempty(t_events)
            T_half = t_events(end);
        end
        break
    end
    dxdf_dx0 = Phif(4, 1);
    if abs(dxdf_dx0) < 1e-14
        fprintf('  DC stalled: STM element (4,1) too small\n');
        break
    end
    Y0_lyap(1) = Y0_lyap(1) - xd_f / dxdf_dx0;
end
if ~converged
    fprintf('  DC did not converge after %d iterations\n', max_iter);
end
end
function [val, isterminal, direction] = y_crossing_event(~, Y)
% Stops integration at y=0 (downward crossing) — used for half-period detection.
val       = Y(2);
isterminal = 1;
direction  = -1;
end
function [val, isterminal, direction] = poincare_section_event(~, Y)
% Records all y=0 downward crossings without stopping — used for Poincaré map.
val       = Y(2);
isterminal = 0;
direction  = -1;
end

function C = compute_jacobi(Y, mu)
% Jacobi constant C = 2U - v^2 for normalized synodic CR3BP state.
x=Y(1); y=Y(2); z=Y(3); vx=Y(4); vy=Y(5); vz=Y(6);
r1 = sqrt((x+mu)^2   + y^2 + z^2);
r2 = sqrt((x-1+mu)^2 + y^2 + z^2);
U  = 0.5*(x^2+y^2) + (1-mu)/r1 + mu/r2;
C  = 2*U - (vx^2+vy^2+vz^2);
end
```

Appendix B: JOI Chemical Baseline

The JOI chemical baseline was computed using the patched-conic two-body framework described in Section 2.3, establishing the propellant cost of conventional capture as the performance benchmark against which the EDT architecture was evaluated. A minimum-energy Hohmann transfer from Earth to Jupiter yields a hyperbolic excess velocity of $v_\infty = 5.643$ km/s at Jupiter's sphere of influence, corresponding to a transfer time of 997.9 days (2.73 years). At the nominal capture perijove of 8 R_J, the hyperbolic approach speed is 21.791 km/s, and the required capture ellipse speed into an 8 x 200 R_J orbit is 20.639 km/s, yielding a JOI ΔV of 1.152 km/s.

The propellant mass required to execute this burn was computed via the Tsoilkovsky rocket equation for a 1000 kg dry mass spacecraft across five propulsion types. Results are summarized in Table B.1. Bipropellant MMH/NTO at $I_{sp} = 320$ s requires 442.6 kg of propellant, producing a wet mass of 1,443.6 kg and corresponding to a mass fraction of 30.7%. The EDT post-JOI propellant mass is zero by definition. The bipropellant comparison directly quantifies the EDT architecture's mass advantage: 442.6 kg of propellant eliminated, equivalent to 44.3% of the spacecraft dry mass freed for science payload, shielding, or system margin.

Table B.1 — Propellant mass comparison for JOI burn ($\Delta V = 1.152$ km/s, dry mass = 1,000 kg).

Propulsion Type	**I_{sp} [s]**	**Propellant [kg]**	**Wet Mass [kg]**	**Mass Fraction [%]**
Cold gas	60	6,085.8	7,085.8	85.9
Bipropellant MMH/NTO	320	443.6	1,443.6	30.7
LOX/LH_2	450	298.3	1,298.3	23.0
NEP low (NEXT-C, +500 kg sys)	1,500	81.5	1,581.5	5.2
NEP high (+500 kg sys)	6,000	19.8	1,519.8	1.3
EDT (post-JOI)	N/A	0.0	—	—

Note: NEP ΔV figures represent equivalent energy cost for mass budget comparison only; NEP JOI requires continuous low-thrust spiraling over months to years rather than a single impulsive burn.