68th International Astronautical Congress, Adelaide, Australia, 25-29 September 2017.

IAC–17–C1.7.3

A RAPID TRAJECTORY DESIGN STRATEGY FOR COMPLEX ENVIRONMENTS LEVERAGING ATTAINABLE REGIONS AND LOW-THRUST CAPABILITIES

Das-Stuart, A.
Purdue University, USA, das15@purdue.edu

Howell, K.C.
Purdue University, USA, howell@purdue.edu

Folta, D.
NASA Goddard Spaceflight Center, USA, david.c.folta@nasa.gov

Designing trajectories in dynamically complex environments is challenging and easily becomes intractable. Recasting the problem may reduce the design time and offer global solutions by leveraging phase space mapping patterns available as accessible regions, and the application of search techniques from combinatorics. A computationally-efficient search process results that produces potential trajectory concepts to meet unique design requirements over a broad range of mission types, including low-thrust scenarios. A successful framework is summarized in terms of four components: (i) Database generation - discretization of well-known dynamical structures to form a searchable 2D or 3D volume or map. (ii) Accessible regions - establishing reachable regions within the searchable database for a given thruster/engine capability. (iii) Automated pathfinding - exploiting machine learning techniques to determine the transport sequence solving for an efficient path. (iv) Convergence/optimization - once the transport sequence is determined as a globally efficient concept, it is optimized locally by more traditional numerical strategies.

1. Introduction

Recent advancements in the capabilities for space exploration offer opportunities to reach a wide array of destinations, from the Moon, to the asteroid belt, as well as to the outer planets. Such endeavors demand effective mission design strategies that trade-off diverse constraints to ensure mission success; achieving the mission goals, satisfying operations requirements, assessing safety, and managing cost are some aspects to be considered. Potentially low engine acceleration levels, uncertainty in the deployment status for secondary payload missions, and re-design for contingency scenarios are also additional challenges. Hence, the increasing complexity of mission design scenarios suggests that a rapid trajectory design framework is valuable — one that offers the exploration of broad trade-spaces and is, at least semi-automated. Such a framework is particularly beneficial in the near term; i.e., to support an efficient cis-lunar transportation architecture that also aids the emergence of new mission concepts beyond the Earth-Moon neighborhood. The knowledge gained from many prior efforts form an integral part of the current investigation.

The construction of a rapid design framework commences with a fundamental understanding of the natural dynamics in the Earth-Moon system that is leveraged to enable mission scenarios. Libration point orbits have proven to be key elements in the trajectory design for missions such as Genesis, ARTEMIS and WMAP; Distant Retrograde Orbits (DROs) may also enable future infrastructure capabilities.[1] Folta et al.[2] offer an interactive catalog of orbit families for applications in multi-body regimes; the advantages of characterizing trajectory parameters to identify potential parking and transfer options are apparent. Additional investigations demonstrate the ability to link various arcs belonging to such orbits,[3] exploit their manifolds[4] and assess their signatures on reduced-dimension Poincaré maps[5] to generate viable transfers in the Circular Restricted Three Body Problem (CR3BP); such a first step renders a suitable initial guess for higher fidelity analysis in an ephemeris model.

The utility of natural arcs can serve as a basis for transfer design, frequently lowering propellant costs even in efficient low-thrust regimes.[6] These hybrid low-thrust/low-energy trajectories are constructed by assessing the intersections of natural arcs with thrusting arcs at a selected hyperplane crossing.[7] Assumptions concerning a steering law, leveraging forward and backward propagations, and/or optimal control theory are common tools to isolate regions on the hypersurface that are attainable by a low-thrust engine. To address the challenges associated with constrained basins of convergence, and the sensitivity of indirect optimization methods to the initial guess, collocation and direct optimization strategies have emerged to solve low-thrust problems.[8]

The access to a wide range of natural arcs, to be coupled with powered arcs resulting from available thrust capabilities, results in the examination of an infinitely large trade-space to satisfy mission constraints. This situation quickly becomes intractable when addressed solely via manual search methods. Thus, some recent investigations address the problem via combinatorial optimization techniques, generally employing two different approaches. In the first approach, traditional numerical processes are employed to construct an initial guess database comprised of locally optimal solutions. Then, well known graph-search and machine learning methods are exploited to solve a multi-objective problem by examining combinations within the database to produce a global or nearly-global optimum. Conclusions from Radice and Olmo,[9] Ceriotti and Vasile,[10] Stuart and Howell,[11] as well as Furfaro and Linares [12] demonstrate the potential of heuristic methods such as Ant Colony Optimization (ACO), and Reinforcement Learning (RL) to be effective in various dynamical regimes and in uncovering local optima that may have otherwise remained unknown. Approaches employing genetic algorithm are also proven as beneficial.[13] The second type of strategy recasts the problem in terms of pathfinding — where the initial guess itself is constructed via Artificial Intelligence (AI) techniques and is then subjected to a numerical corrections process. Tsirogiannis,[14] as well as Trumbauer and Villac[15] generate impulsive transfer options by constructing a framework of pre-computed natural arcs and using graph search methods to evaluate the links. The nodes then serve as waypoints in a complex dynamical regime. Simplifying assumptions in a two-body model are exploited by Parrish[16] to employ heuristics in solving for consistent low-thrust initial guesses.

The current investigation strives to establish a framework for automated low-thrust initial guess generation within a multi-body regime, via pathfinding techniques. The foundation is developed for a process that alleviates the challenges associated with thrust law construction, the sequencing of thrust and coast arcs, and the limited solution options resulting from narrow basins of convergence when implementing traditional numerical techniques. Approximating low thrust segments via a sequence of impulsive maneuvers is effective for preliminary mission design,[17] and aids in this investigation. Both deterministic processes, as well as stochastic techniques exploiting software 'agents', enable the desired broad search capability. Future developments deliver a flexible process to accommodate varied constraints, dynamical regimes, and spacecraft parameters. Such a tool is envisioned as applicable in supporting future onboard autonomy and, therefore, beneficial in time-critical contingency planning scenarios.

The force models and numerical techniques that support the initial guess generation process are introduced in Section 2. Some artificial intelligence and machine learning strategies apprear in Section 4, and the design framework is formalized in Section 5. Sample scenarios and preliminary results are offered in Section 6.

2. Dynamical and Force Model

Although the dynamical sensitivity is evident, the CR3BP offers an opportunity to approximate the higher fidelity dynamics and exploit the natural flows that are otherwise unavailable in simpler dynamical regimes. Also, a target application for the design framework is a robust cis-lunar transportation architecture, so the methodology is applied within the context of the Earth-Moon CR3BP. Here, the Earth (*primary,* P_1) and the Moon (*secondary,* P_2) are assumed to revolve in circular orbits around their common barycenter.[18] The spacecraft mass is assumed to be negligible compared to the more massive bodies. Construction of the Equations of Motion (EOM) for the spacecraft (P_3) as viewed in a rotating frame also incorporates a thrust force to model the physical capabilities of an engine/thruster.

$$\dot{\boldsymbol{\chi}} = \begin{Bmatrix} \dot{\boldsymbol{r}} \\ \dot{\boldsymbol{v}} \\ \dot{m} \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{v} \\ \underbrace{\boldsymbol{f}(\boldsymbol{r}) + \boldsymbol{g}(\boldsymbol{v})}_{\text{natural}} + \underbrace{\frac{\mathbb{T}}{m}\hat{\boldsymbol{u}}}_{\text{low-thrust}} \\ \frac{-\mathbb{T}}{Isp\; g_o} \end{Bmatrix} \qquad [1]$$

The differential equations incorporate contributions from both the natural gravitational, and thrust acceleration sources to capture the motion of the spacecraft (s/c), and its mass-history over time. In these equations, $\boldsymbol{\chi}$ is the full state vector comprising the vehicle position and velocity vectors ($\boldsymbol{r}$ and $\boldsymbol{v}$ respectively), and vehicle mass m. The thrust magnitude is represented by $\mathbb{T}$, and the thrust direction by $\hat{\boldsymbol{u}}$ (where a caret identifies unit magnitude). *Isp* is the engine specific impulse and, g_0 is the reference gravitational acceleration. Even without the engine-specific thrusting terms, there is no closed-form solution to the natural EOMs. So, in the thrust-free problem, other quantities such as the period, stability and the Jacobi Constant (JC, an energy-like quantity) of the natural solutions help characterize the motion of the s/c within the confines of the CR3BP model. The JC is defined as:

$$JC = 2U^* - v^2 \qquad [2]$$

where U^* is a pseudo-potential term whose gradient contributes to derivation of the EOMs.

$$U^*(x, y, z) = \frac{1-\mu}{r_{13}} + \frac{\mu}{r_{23}} + \frac{1}{2}(x^2 + y^2) \qquad [3]$$

So, U^* is a function of the s/c position relative to the barycenter (x, y, z) and the two primaries r_{13} and r_{23}, and the mass ratio of P_2 to the total system mass, μ. Note from Eqn. [2] that the JC term is only constant in the autonomous system where propulsive forces are absent; its time-history resulting from low-thrust maneuvers also offers valuable insights for developing and applying the automated framework.

To deliver the thrust force in Eqn. [1], a relatively new technology, advanced ion-drive propulsion, is gaining popularity due to the ability to improve the payload delivery capabilities. This efficiency however, is delivered at lower thrust levels and therefore, potentially lengthy flight durations.[19] Currently, operational s/c are built to function at a constant specific impulse (CSI). But, it is assumed that the specific impulse is continuously tuned (Variable Specific Impulse, VSI), to optimize the locations of high and low thrust burns, thereby optimizing the propellant consumption.[20] The VSI model is promising for future endeavors and it is reasonable to assume such capability. Thus, in the current analysis, the initial guess generated is transitioned to continuous low-thrust solutions in both the CSI and VSI models. Transitions to a higher fidelity model enabled by chemical engines is also investigated.

3. Numerical Corrections and Optimization

Constructing low-thrust optimal trajectories typically involves a formulation as an optimal control problem, and solving for the time-histories of the thrust magnitude and the thrust force direction to meet the desired boundary conditions using direct or indirect approaches. In general, direct methods are more robust, but induce a large dimensionality and often require assumptions on the thrust profile (e.g., constant thrust direction over each arc). Indirect optimization methods mitigate such limitations, but often introduce numerical challenges.[21] Both approaches are employed in the current investigation.

3.1 Constant Specific Impulse Engine Parameters

In both the CSI and VSI regimes, the available thrust magnitude is a function of the engine power allocation, ($\mathbb{P}$) and efficiency (*Isp*). The relationship is modeled as:

$$\mathbb{T} = \frac{2\mathbb{P}}{Isp\, g_0} \qquad [4]$$

In the CSI regime, constant power and *Isp* parameters dictate a constant thrust magnitude. The basis of the control authority that maneuvers the s/c is an on-off engine toggle and thrust-vectoring. A continuous low-thrust trajectory is constructed from a discontinuous initial guess by employing a multiple-shooting scheme. Also, due to the inherent numerical sensitivities associated with indirect methods for the CSI trajectory design, direct methods are selected in this preliminary analysis. An iterative Newton-Raphson scheme is employed to compute the set of design vector variables ($\boldsymbol{X^*}$) that satisfy the specified constraints, $\boldsymbol{F}(\boldsymbol{X^*}) = \mathbf{0}$.

$$, i.e., \boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_i \\ \boldsymbol{u}_i \\ \tau_i \\ TD_i \end{bmatrix} \qquad \boldsymbol{F} = \begin{bmatrix} \boldsymbol{x}_{i\,desired} - \boldsymbol{x}_{i\,actual} \\ \tau_i + TD_i - \tau_{i+1} \\ \boldsymbol{\psi}_{desired} - \boldsymbol{\psi}_{actual} \\ {\boldsymbol{u}_i}^T \boldsymbol{u}_i - 1 \end{bmatrix} = \mathbf{0} \qquad [5]$$

The design vector $\boldsymbol{X}$, is comprised of the states ($\boldsymbol{x}_i$), thrust directions ($\boldsymbol{u}_i$), epoch times (τ_i), and propagation durations (TD_i) at select-nodes (i) that are available from the initial guess. In an alternative formulation, the thrust magnitude is also included as a design variable rather than pre-assigning a binary (on/off) state to the nodes and allowing the inter-node propagation times to determine the duration of the thrust and coast arcs. The constraint vector $\boldsymbol{F}$ aims to meet state-continuity, epoch-continuity, boundary condition ($\boldsymbol{\psi}$), and thrust direction unit-vector constraints at the specified nodes. The converged solution is passed through a Nonlinear Programming (NLP) software such as FMINCON or SNOPT to optimize a cost function that is formulated to offer a favorable solution to the mission objectives and constraints.

3.2 Variable Specific Impulse Engine Parameters

For power-limited VSI engines, a constant power level results in a variable thrust magnitude and *Isp* values (Eqn. [4]). These additional variables further expand the dimensionality of the problem, but the modulation freedom also aids in the VSI convergence process with poor initial guesses. So, indirect methods are well-suited for numerical computations. Previous studies[22] have demonstrated that a hybrid approach is capable of satisfying optimality conditions by offering a convergence radius comparable to direct methods, without compromising the benefits of lower dimensionality. So, a hybrid method that functions independently of the sensitive transversality conditions, but incorporates the co-states to formulate the optimal control history, is employed to converge solutions in this model. The problem is posed as a Two-Point Boundary Value Problem (2PBVP) and solved using optimal control and primer vector theory.[23, 24] The performance index J to be maximized in this investigation is the delivered mass, i.e.,

$$max\,J = m_f + \boldsymbol{\nu}_0^T\boldsymbol{\psi}_0 + \boldsymbol{\nu}_f^T\boldsymbol{\psi}_f + \int_{t_0}^{t_f}(\hat{H} - \boldsymbol{\lambda}^T\dot{\boldsymbol{\chi}})dt \qquad [6]$$

subject to the boundary conditions $\boldsymbol{\psi}$, Lagrange multipliers collected in the vector $\boldsymbol{\nu}$, and the path constraints within the integral. The position, velocity and mass co-state vector is defined as $\boldsymbol{\lambda}^T = [\boldsymbol{\lambda_r}, \boldsymbol{\lambda_v}, \lambda_m]^T$. Here, $\hat{H} = H + (\boldsymbol{u}^T\boldsymbol{u} - 1)$, is a function of the Hamiltonian, H (constant), and the thrust direction unit-magnitude constraint. The Hamiltonian in this problem is defined as:

$$H = \begin{bmatrix} \boldsymbol{\lambda_r^T v} \\ \boldsymbol{\lambda_v^T}(\boldsymbol{f}(\boldsymbol{r}) + \boldsymbol{g}(\boldsymbol{v}) + \frac{\mathbb{T}}{m}\hat{\boldsymbol{u}}) \\ -\lambda_m\frac{\mathbb{T}^2}{2\mathbb{P}} \end{bmatrix} \qquad [7]$$

Note that the mass-flow rate defined in Eqn. [1] is rewritten in Eqn. [7] in terms of the power to aid in the simplification of further derivations (i.e., $\dot{m} = \frac{\mathbb{T}^2}{2\mathbb{P}}$). Deriving the necessary conditions of the cost function J, requires solving for $dJ = 0$ and construction of the Euler-Lagrange equations:

$$\dot{\boldsymbol{\lambda}} = \left(-\frac{\partial\hat{H}}{\partial\boldsymbol{\chi}}\right) = \begin{Bmatrix} -\boldsymbol{\lambda_v^T}(\frac{\partial\boldsymbol{f}(\boldsymbol{r})}{\partial\boldsymbol{r}}) \\ -\boldsymbol{\lambda_r^T} - \boldsymbol{\lambda_v^T}(\frac{\partial\boldsymbol{g}(\boldsymbol{v})}{\partial\boldsymbol{v}}) \\ \lambda_v\frac{\mathbb{T}}{m^2} \end{Bmatrix} \qquad [8]$$

$$\left(\frac{\partial\hat{H}}{\partial\boldsymbol{u}}\right) = \boldsymbol{\lambda_v}\frac{\mathbb{T}}{m} + 2\hat{\boldsymbol{u}} = 0 \qquad [9a]$$

$$\left(\frac{\partial\hat{H}}{\partial\mathbb{T}}\right) = \boldsymbol{\lambda_v}\frac{\hat{\boldsymbol{u}}}{m} - \lambda_m\frac{\mathbb{T}}{\mathbb{P}} = 0 \qquad [9b]$$

Inspection of Eqn. [7] and Eqn. [9a] indicates that thrust direction parallel to the velocity co-state direction $\hat{\boldsymbol{u}} = \frac{\boldsymbol{\lambda_v}}{\lambda_v}$ maximizes H and, therefore, the cost function. Similarly, the power $\mathbb{P}$ also operates at a maximum, $\mathbb{P} = \mathbb{P}_{max}$. Equation. [9b] is re-arranged as $\mathbb{T} = \frac{\lambda_v\mathbb{P}_{max}}{\lambda_m m}$, and its history is extractable in a post-processing step. The VSI trajectory corrections process is solved using a free-variable and constraint formulation as described in Section 3.1. Note that a low-thrust engine is typically most efficient over longer flight durations. So, to prohibit the *Isp* values from modulating to infinity ($\dot{m} \rightarrow 0$), the VSI corrections process is a time-fixed formulation.

3.3 Chemical Engine Parameters

Construction of a converged/optimized trajectory enabled by a chemical engine allows velocity discontinuities as a design parameter. So, a process similar to the direct approach as discussed in Section 3.1 is implemented, with the exclusion of the thrust magnitude and thrust direction terms in the design and constraint vectors.

4. Pathfinding Strategies

The generation of an initial guess trajectory that transfers a s/c from departure location to a specific destination is cast as a routing problem. Exact or heuristic algorithms are often employed to construct solutions, the choice usually determined by the scope of the problem, computational capabilities with regard to time (speed-complexity) as well as storage (space-complexity), and the desired fidelity of the solution. In this analysis, both approaches are examined to better understand their applicability to trajectory design scenarios. The algorithm is formulated as an automated agent (software)-based search for a path that achieves a specified goal, when subject to a set of prescribed constraints.

4.1 Dijkstra's Algorithm

Since the inception of graph theory by Leonhard Euler in the 18th century,[25] the specific methods and variants have demonstrated great utility in varied disciplines such as routing, genomics, and artificial intelligence. In each application, the graph represents the set of available options (nodes N) under consideration.[26] The edges E or *arcs*, linking pairs of nodes are shaped by the costs between them, $Cost[< n_i, n_j >]$. Non-unitary costs establish a weighted graph that is either *directed* if the edge-costs are oriented, or *undirected* otherwise. Commencing at the *departure* node, the graph is expanded one arc at a time, $< n_i, n_j >$, until the agent arrives at the *target* node. This traversal is accomplished by: a) establishing a *frontier of neighbors* that the agent can meet in one hop when the edge costs from the current node are finite; and b) selecting a neighbor from this frontier to incrementally expand the search towards the target. Various motivating factors such as fewest nodes to the target, or a preference in time-space complexity, influence the choices in the search algorithm. Such choices consequently dictate the selection of the next node in expanding the frontier. The path, $\mathcal{P}$, to the target is, therefore, constructed via a sequence of arcs with a total path cost, $Cost(\mathcal{P}) = \sum_{i=1}^{K} Cost[< n_i, n_{i+1} >]$, where K is the cardinality of the set of arcs in the sequence.

The type of graphs encountered within the context of mission design is usually weighted and directed, due to the complex dynamics and the mass-time trade-offs that constitute the edge-costs. So, where the goal is multi-objective optimization, Dijkstra's algorithm is employed as the *exact* approach in this study; Dijkstra's formulation constructs the least-cost (shortest) pathway to the destination. This algorithm employs a form of greedy search that always selects the lowest cost neighbor along a frontier; and the cost from the departure node is updated at a later time if an alternative, but shorter route is later discovered. One approach to implement Dijkstra's algorithm with a priority queue that constantly re-stacks the nodes in ascending order of cost is summarized in Figure. 1. Such a search strategy that incorporates the least-cost expansion of each arc, guarantees the emergence of the shortest path to the target, or the global optimum.[27] Non-unique edge costs from a node can lead to non-unique global optima in certain scenarios. The algorithm also guarantees that, if a solution exists, it will be uncovered in a finite time interval. However, this finite computation time increases with an increasing search space. The time is bounded by polynomial growth of order $O(n^2)$ for naive implementations, and $O(E + nlogn)$ when employing a Fibonacci-heap to implement the priority queue.[14] Note that heap implementation becomes increasingly challenging for a complex search space. Significantly contributing to the challenge of an increased computational time is the uninformed nature of the search, i.e., the algorithm is not aware of the target/user's goals as the search evolves. Built on similar principles, the A* algorithm introduces heuristics to conduct an informed search towards the target; such an adjustment typically improves the time-complexity, depending on the implementation.[26] However, A* has not been explored in this investigation, as strict rules on the admissibility of the heuristic are challenging to satisfy in a complex dynamic environment with multi-objective optimization goals, especially with simultaneous considerations of mass-time trades.[16]

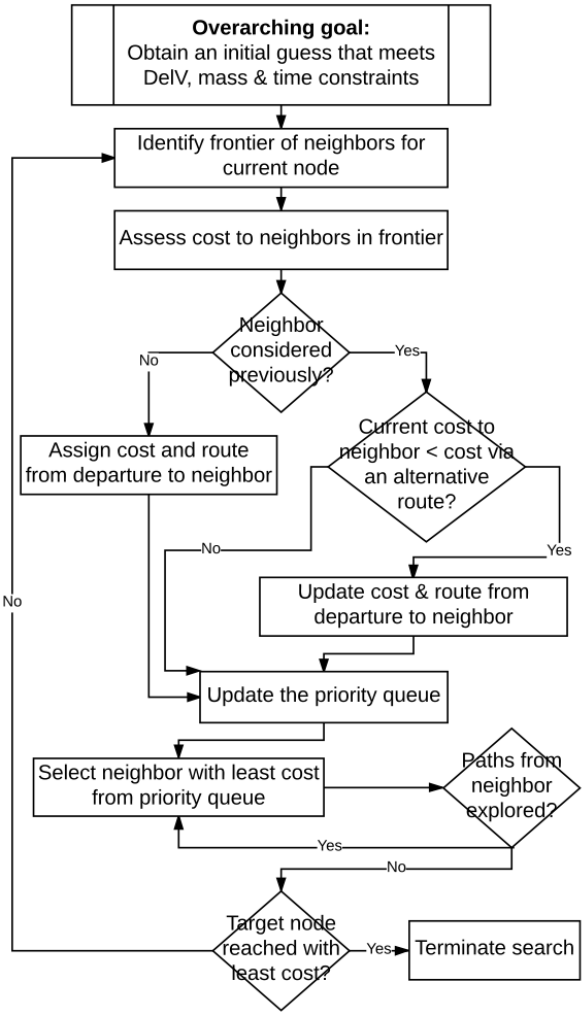

Overarching goal:
Obtain an initial guess that meets DelV, mass & time constraints
Identify frontier of neighbors for current node
Assess cost to neighbors in frontier
Neighbor considered previously?
No
Yes
Assign cost and route from departure to neighbor
Current cost to neighbor < cost via an alternative route?
No
Yes
Update cost & route from departure to neighbor
No
Update the priority queue
Select neighbor with least cost from priority queue
Paths from neighbor explored?
Yes
No
Target node reached with least cost?
Yes
Terminate search

Fig. 1: An implementation of Dijkstra's algorithm

4.2 Heuristically Accelerated Reinforcement Learning (HARL)

Any endeavor in the space environment is complex. A flexible, long-term infrastructure will benefit from a mission design approach that delivers transfer solutions while accommodating uncertainties in the environment, variable performance measures, and is responsive to shifting goals. Reinforcement Learning (RL) algorithms originating from efforts in the artificial intelligence domain are an effective tool in balancing assorted ambitions, and are implemented via an agent interacting with its environment; such an agent learns to deliver appropriate choices that lead to desirable solutions. The reinforcement learning approach is commonly formalized as a Markov Decision Process (MDP)[26] that is constructed from a tuple: $< S, A, P, R >$; S then, is a set of states available to agent; A is a set of actions available to agent; P is P(s,a,s′), the probability that action a in state s leads the agent to arrive at state s', $\mathbb{R} : S \times A \rightarrow \mathscr{R}$ is the reward received for an action a in state s. The aim is to learn an optimal policy (π^*) that involves executing an action at a given state that maximizes the cumulative rewards received by the agent over multiple episodes. A strategy to accomplish this task involves a value function ($\mathbb{V}^\pi$) that iterates over all the possible actions from a state to eventually arrive at the optimal choice ($\mathbb{V}^*$):

$$\mathbb{V}^*(s) \leftarrow \max_a \sum_{s'} P(s,a,s')[\mathbb{R}(s,a,s') + \gamma \mathbb{V}^*(s')] \qquad [10]$$

Equation. [10] is essentially the *Bellman Equation* employed widely in dynamic programming, and satisfies the necessary conditions for optimality. The prime symbol (′) indicates a state(s) accessible from the *frontier*, as introduced in Section 4.1. In complex regimes where the probabilities defining the system model and the rewards are not known a priori, it is useful to construct the problem as a model-free, state-value pair approach. A state-value update function is formulated in recursive form as:

$$\begin{aligned}\hat{\mathbb{F}}(s,a) \leftarrow \hat{\mathbb{F}}(s,a) + \alpha[&\mathbb{R}(s,a,s') \\ &+ \gamma \max_{a'} \hat{\mathbb{F}}(s',a') - \hat{\mathbb{F}}(s,a)]\end{aligned} \qquad (11)$$

and forms the repository of reinforcements from which the agent learns the desirable behavior. The discount factor $0 \leq \gamma \ < 1$ is a measure of the balance between immediate and future rewards; smaller values of γ favor immediate rewards. The learning rate of the agent is specified by α. It diminishes over episodes to avoid fixation on only the most recently gained knowledge, and enables trust on prior knowledge that is posed as reinforcements received over time. The optimal policy is then constructed as:

$$\pi^*(s) = arg \max_a \sum_{s'} \hat{\mathbb{F}}(s,a) \qquad [12]$$

The convergence to such an optimal policy using the RL approach is guaranteed by re-visiting state-action pairs infinitely many times, and the process may be prolonged in scenarios subject to a large state-space.

The time-complexity of some RL algorithms can be as challenging as exact approaches, e.g. Dijkstra's algorithm. So, it is beneficial to introduce a heuristic function that accelerates the learning process by biasing the selection of an action a, given a state s, with a reward $\mathbb{R}$. Such an algorithm is termed a Heuristically Accelerated Reinforcement Learning (HARL) algorithm, and the time-complexity depends on the accuracy of the heuristics. The ability of the heuristic function to influence the action-choice is based on the ability to exceed the variations in $\mathbb{F}(s,a)$.[28] Scenarios with large state-spaces and multiple objectives also motivate a co-ordinated effort from a multi-agent distributed HARL approach, particularly one that exploits parallel processing. In such a distributed network, the agents work cooperatively by updating a centralized reinforcement repository based on the knowledge (cumulatively discounted rewards) gained during a specific episode. Within the context of mission design, the agent continues the search until a stopping condition is satisfied, i.e. a terminal condition and/or a violation of constraints. An overview of the learning process (within the context of mission design) is illustrated in Figure 2. Clearly, the learning process within each episode is comprised of two search scenarios — exploration and exploitation. Exploration enables a training phase where the agent *learns* about likely consequences of actions in the environment; the exploitation phase enables the agent to engage in informed decisions by capitalizing on previously gained knowledge. The policy at a particular state as influenced by the state-action pair and heuristic is:[28]

$$\pi(s) = \begin{cases} \mathbb{E}(\mathbb{F}(s,a) \bowtie \xi \mathbb{H}(s,a)^\beta), & \text{if } q \geq p \\ a_{random}, & \text{otherwise} \end{cases} \qquad [13]$$

where ($\mathbb{H} : S \times A \rightarrow \mathscr{R}$) is the heuristic function, $\bowtie$ is a math operator as determined by the RL algorithm, ξ and β are weighting parameters that dictate the influence of the heuristic, and a_{random} is the action chosen randomly from all those available in state

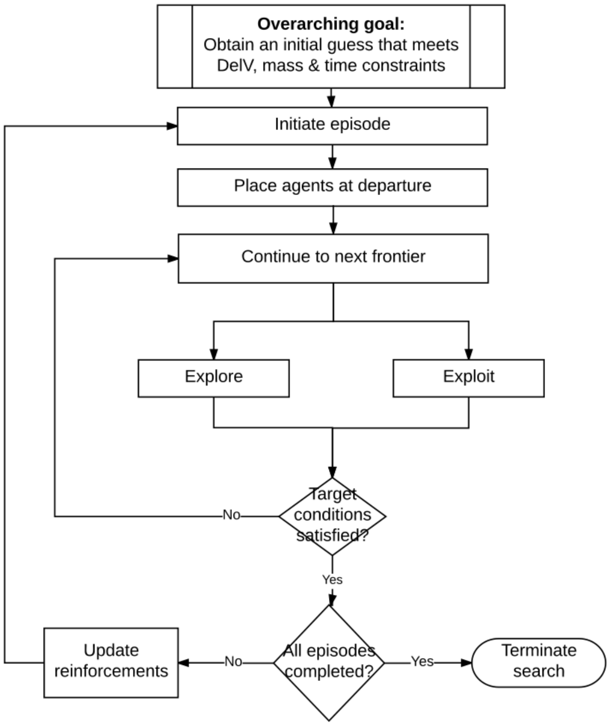

Overarching goal:
Obtain an initial guess that meets DelV, mass & time constraints
Initiate episode
Place agents at departure
Continue to next frontier
Explore
Exploit
Target conditions satisfied?
No
Yes
All episodes completed?
No
Yes
Update reinforcements
Terminate search

Fig. 2: An implementation of HARL algorithm

s. The construction of the exploitation function $\mathbb{E}$ is specific to a particular application; some variants include a greedy strategy that incorporates $max(\mathbb{E})$, a minimax approach that minimizes the losses in a maximum loss scenario, and a softmax process that stochastically selects an action.[29] In this equation, p, $(0 \leq p \leq 1)$, is the trade-off parameter between exploration and exploitation, and q is a random value from a uniform distribution in [0,1]. Greater values of p encourage exploration and lower values bias the process toward exploitation. Extensive exploration in earlier episodes is beneficial, then a gradual shift to eploitation. A convenient function to control the steady-state value of the trade-off probability, p_{ss} by the k^{th} episode (Ep) is constructed as:[11]

$$p = p_{ss} + (1 - p_{ss})e^{(-\frac{k-1}{ln(Ep)})} \quad [14]$$

The blending of heuristics with reinforcement learning is a powerful utility in solving many NP-hard and NP-complete pathfinding problems, e.g., the Traveling Salesman Problem (TSP).

5. Development of the Design Framework

A primary objective of this investigation is the development of a framework to demonstrate the automated pathfinding process within the context of mission design. The first step in the framework development is identifying the type of solution for the relevant design scenario, including the associated level of fidelity and level of optimality. The transition of initial guesses (IG) from lower-fidelity dynamical models such as the CR3BP enhances rapid design and is effective. The first phase in the framework development, therefore, involves a database of existing options, e.g., periodic and non-periodic trajectory solutions in three-body system. The next phase establishes the *reach* of the spacecraft within the given database, as influenced by its thrust-to-mass ratio, propellant efficiency and other performance characteristics. The available trajectory options and the ability of the spacecraft to transition through them dictates the subsequent phase — the *sequencing* of intermediate pathways to enable a satisfactory path between the departure and destination conditions. The final phase involves convergence and optimization of the initial guesses via numerical techniques to ensure that the mission constraints are not violated, and the requirements are fulfilled. The details associated with each phase follow.

5.1 Compilation of the Database

Exploitation of the natural flow in a dynamical system potentially reduces propellant consumption and enables otherwise unexplored mission scenarios. Such an option is especially attractive for secondary payloads that can possess acceleration levels at least an order of magnitude lower than the natural acceleration levels in the system of interest. For example, consider typical cubesat accelerations, on the order of $\approx 8.6e-5m/s^2$, to the Moon's acceleration contributions that increase to larger values when approaching the lunar vicinity (Figure 3). Note that the contour plot is not constructed all the way to P_2 only to aid with visual clarity.

The CR3BP offers a multitude of natural solutions — periodic and quasi-periodic orbits in the vicinity of the primaries and libration points, and manifold tubes that reflect the flow throughout the region. The signatures of these structures are captured either by lower-dimensional Poincaré maps at selected surfaces of section, or by discretization to facilitate *arc* extraction from the discrete initial conditions. This investigation employs the latter option. The discretization of the orbit families to develop a suitable database involves careful consideration of the inter-orbit and intra-orbit spacing; i.e. the spacing *between* the orbits and the states that are sampled *along* the or-

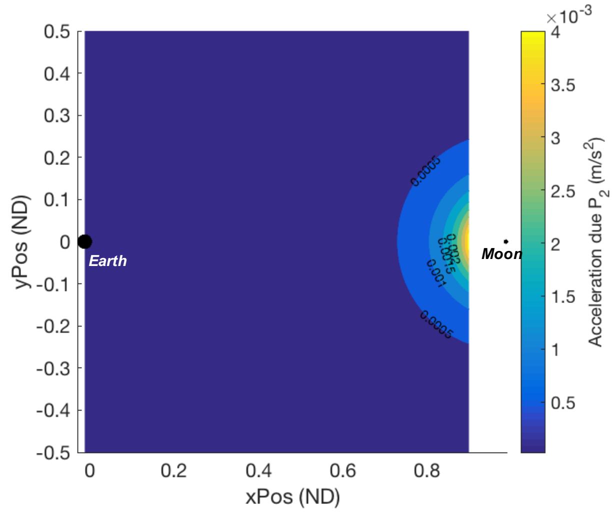


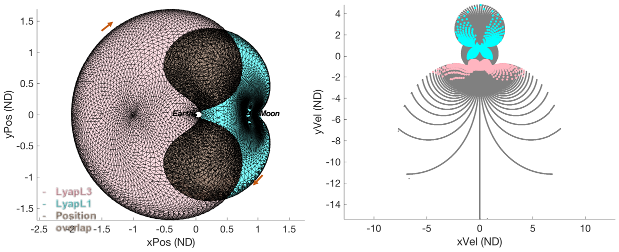
Fig. 3: Contour plot - acceleration contribution from the Moon in the Earth-Moon system

bits/trajectories respectively. Although a finer grid offers a more accessible search domain, the memory and storage limitations associated with the available computational resources also play a key role in determining the size of the grid computed to represent each orbit family. The discretization therefore directly impacts the number of intermediate conditions available to the s/c along a *frontier* and, thus, also impacts the end-to-end guess for a transfer.

5.2 Influencing the Nature of the Solution

The gravitational forces in a system clearly influence the flow throughout configuration space. Mission design is sometimes envisioned as harnessing the appropriate combination of orbit family geometries, velocities and energy levels to satisfy architecture constraints such as altitude relative to primary body, line-of-sight for communications, and eclipsing conditions, and balancing such constraints with mission objectives. Solving the problem and delivering an acceptable design can be described as an attempt at combinatorial optimization, a topic that has been studied extensively in other fields as well; the Traveling Salesman Problem (TSP) is one such example. In such an application, the cities form the *database* of possible options available to an agent, and the travel cost between each city in the database is a key contributor leading to the decisions associated with the construction of an optimal path through all the cities before returning to the departure location. Analogous to the TSP, in a mission design context, the different orbit families form the database of available options, and their location, expanse, stability and energy contribute to the costs associated with travel among them. Thus, it is advantageous to establish a metric of *connectivity* between the orbit families such that essential families are available in the searchable database for a particular mission scenario.

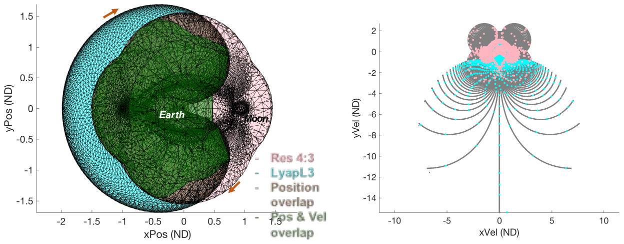
Connectivity between a pair of families can be assessed in many ways; in this analysis it is measured as the amount of overlap in the surface areas defined for their corresponding position and velocity. The surface area of a family is computed by decomposing the family into triangles using Delaunay triangulation and, then, summing the areas formed by the triangular faces. Vertices are defined to be within close proximity only if they are able to be circumscribed within a specified radius of the parameter of interest. For purposes of demonstration, a velocity bound of $60m/s$ is selected as one such radius in the figures that follow. Figure 4 illustrates that a transfer from the L_1 Lyapunov family to the L_3 Lyapunov family may be challenging for diminished thrust levels due to the sparse overlap in position and none in velocity (i.e., no points emerge that allow an arc from either family to approach the other family within the defined $60m/s$ velocity bound of each other). Thus, for low-thrust engines, such a transfer is better facilitated by incorporating an intermediate family such as a resonant family, in the database as well — one that offers the potential to use intermediate transfer arcs to bridge the gap between the L_1 and L_3 Lyapunov families in both position and velocity space (Figure 5). An assessment of the surface area of overlap in both the position and velocity domains enables the construction of a *connectivity matrix* (Table 1). The first number in each cell is the percentage of area overlap in position and the second number is indicative of the overlap in both position and velocity space. These quantities serve as a rough guide for making informed decisions about which orbit fami-



(a) Position overlap

(b) Velocity overlap

Fig. 4: Position and velocity overlap in the L_1 Lyapunov and L_3 Lyapunov families



(a) Position overlap (b) Velocity overlap

Fig. 5: Position and velocity overlap in the L_3 Lyapunov and Resonant 4:3 families

Table 1: An example family connectivity matrix. Color code: % Position overlap, % Position and velocity overlap

	Lyap L_1	**Res** $4:3$	**SP** L_4	**Lyap** L_3
Lyap L_1	-	63%, 20%	27%, 3%	29%, 0%
Res $4:3$		-	56%, 49%	58%, 30%
SP L_4			-	83%, 66%
Lyap L_3				-

lies are beneficial to incorporate into the searchable database. Note that these areas grow/shrink based upon the user-defined bounds on acceptable velocity overlap. For example, the connectivity between the L_1 Lyapunov and L_4 Short-Period families is sparse, but the Resonant 4:3 family has a decent connectivity metric to both these families. So, as a minimum, at least this resonant family would be beneficial to include during the search for a transfer between the members of the L_1 Lyapunov and L_4 Short-Period families. In brief, the connectivity matrix is a measure of 'ease of access' between different orbit families in a given dynamical regime and for a specified thrust capability.

5.3 Computation of Accessible Regions

The orbit families incorporated in a search routine constrain the type of arcs that are admissible in the construction of an initial guess. However, it is the engine/thruster capability and the propagation/horizon-time through configuration space which determine the *natural* arcs within the selected database that are *reachable* by a particular s/c. The computation of an Accessible Region (AR) originates with a perturbation of the spacecraft's current velocity within a circle/sphere (for the planar/spatial problem respectively) with a prescribed radius, followed by a propagation of the perturbed and unperturbed states for a fixed time-horizon. The resulting behavior downstream of this propagation is a stretching of the perturbed states from the unperturbed natural arc due to the influence of the existing gravitational forces in the system. A measure of the deviation of the perturbed states from the end of the natural arc is exploited to formulate a circular/spherical accessible region in a planar/spatial setting respectively. These simplified geometric shapes are implemented over the realistic irregular-shaped ARs that result from a propagation, to enable finer control over the quantity of natural arcs available to the s/c. Empirical testing has supported the decision to employ such an approximation. For both the chemical and low-thrust studies, perturbations in the s/c velocity are induced via chemical impulses, the magnitude of which is determined by the engine characteristics, the engine operation time and propagation/horizon-time.

$$\Delta V = Isp\ g_0\ ln(\frac{m_0}{m_f}) \qquad [15a]$$

$$where, m_f = m_0 - (\dot{m} \times \Delta\approx) \qquad [15b]$$

It is possible to fix the value of the impulse (ΔV) to effect the resulting engine-operation time ($\Delta\approx$), but the reverse is implemented in this investigation as a constant operation time is a more intuitive quantity to define. Inspection of Eqn. [1] and Eqn. [15] illustrates that higher specific impulse values result in smaller ΔV perturbations for a fixed operation time, and aids in delineating between the behavior of propellant efficient low-thrust and inefficient chemical regimes. Although the horizon-time does not play a role in the determination of the ΔV, it is yet another quantity that is influences the footprint of the AR. Figure 6 illustrates that bigger ΔV_s for a fixed horizon-time avail of a greater AR and therefore, more natural arcs to choose from. Longer propagation times for a fixed ΔV have a similar effect.

The generation of these ARs may be developed independently of mass updates by computing the ΔV associated with the initial mass fraction $\frac{m_0}{m_f}$ and holding this value constant for subsequent AR computations. This alternative offers the flexibility for an optimization routine to consider only the ΔV contributions from an initial guess, such as is common in the design for chemical trajectories. This choice however results in conservative estimates for the ARs, because in reality, they would grow larger with increased acceleration levels as the s/c burns more propellant and becomes lighter over time. In this preliminary study, the pursuit of a modular architecture to handle varied engine characteristics has led to the approximation of low-thrust flow with chemical impulses for the

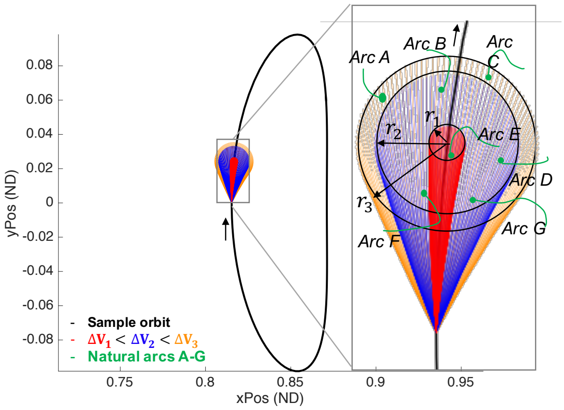


Fig. 6: Influence of ΔV_s on the footprint of accessible regions

IG generation phase. Consequently, the EOMs associated with the thrust magnitude, directions and mass are not required to be propagated for the IG generation and can support computational savings. Improving the level of fidelity of the low-thrust AR generation scheme using low-thrust perturbations directly is under consideration for future contributions.

5.4 Implementation of Automated Pathfinding

Pathfinding seeks to *sequence* appropriate conditions to maneuver the s/c from departure to destination. In this study, the available conditions are the natural arcs incorporated within the database and made available within each computed accessible region; they constitute the *frontier* of solutions introduced in Section 4. Both the exact (Dijkstra's algorithm) and heuristic (HARL) methods select a condition from within each frontier to continue the search based on varying metrics as described in this section. In each case however, the overarching goal is posed as an optimization problem that aims to prioritize the payload mass or transfer duration in the persistent *mass-time* tradeoff challenge:

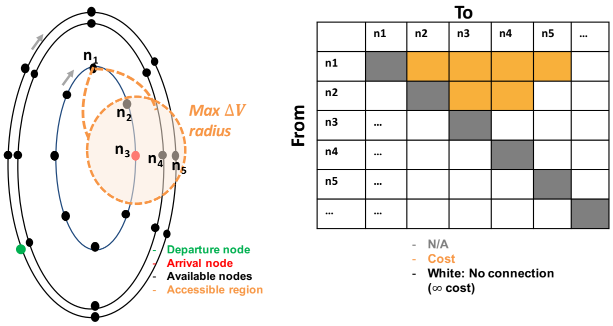
$$min\ \mathbb{J} = W_t t_f + W_p m_p \qquad [16a]$$

$$min\ \mathbb{J} = W_t t_f + W_p \Delta V \qquad [16b]$$

where, the propellant mass m_p or ΔV are always weighted against the transfer duration, t_f such that the weighting on TOF, W_t is always 1, and W_p is a design variable. The transfer duration is the aggregate sum of the horizon times associated with each natural arc assembled within the transfer sequence, and the propellant consumption/ΔV_s are computed as a result of the velocity discontinuities between each of these arcs. The schematic of an AR as presented in Figure 6(a) is used to illustrate that exercising the maximum ΔV at a given node locates a s/c on the circumference of its associated AR, whereas a coast arc places it in the center of this region. So, the ΔV required at a given node to reach another node circumscribed within an AR lies in the interval $[0, \Delta V_{max}]$. The associated mass updates are computed using Eqn. [15]. The implementations specific to the two pathfinding methods considered are outlined as follows.

Procedure for Implementing Dijkstra's Algorithm

In this investigation, Dijkstra's algorithm is implemented as a greedy search method which always selects the node from a frontier with the lowest cost from departure. The node-to-node costs are precomputed (parallelizable process) and stored in a cost adjacency matrix as depicted in Figure 7. The matrix is purposed as a directed graph that informs the agent of unique costs to traverse between the nodes, including a cost of ∞ when there exist no connections between them. These costs feed the the priority queue outlined in Section 4.1. The lowest-cost route to the destination is discovered in a serial fashion to allow the cost to a node from departure to be revised when a less expensive connection to it is established. Using Eqn. [16a] to measure the inter-node costs would require them to be computed on-line to establish the ARs based on a varying mass history. The mix of parallel and serial components to the search, in addition to the storage of an extra variable (mass-history) for each node queried can contribute to increased time-space complexity and a more challenging implementation of the algorithm. So, for both the low-thrust and chemical transfer scenarios, the inter-node costs are estimated using Eqn. [16b] where the ΔV cost is weighted against transfer time to satisfy the user's mass-time priority directive. The resulting end-to-



	n1	n2	n3	n4	n5	...
n1	▨	🟧	🟧	🟧	🟧	
n2		▨	🟧	🟧		
n3	...		▨			
n4	...			▨		
n5	...				▨	
...	...					▨

(a) Accessible nodes (b) Cost adjacency matrix

Fig. 7: A schematic of a sample AR from a node (n_1) and example cost adjacency matrix.

end transfer sequence is a global optimum for the lower-fidelity and s/c assumptions stated earlier.

Procedures for Implementing HARL Algorithm

A HARL algorithm does not guarantee optimality due to the inclusion of heuristics, but conducting multiple search episodes can enable convergence to a nearly-optimal or satisficing solution to address the mass-time priority objective. The entirely parallelizable capabilities within HARL offers a flexible architecture where the inter-node costs are computed on-line as the agent progresses through the database towards the destination. As a result, either equation in Eqn. [16] may be adopted based on the needs of the user. Referring to the process outlined in Figure 2, the agent initiates the search by randomly exploring the state-space and gradually performs more exploitation over subsequent episodes as devised by Eqn. [14]. The exploitation of a particular node from an AR is guided by a heuristic constructed as:

$$\mathbb{H} = \frac{m_f^{W_m}}{d^{W_d}} \quad [17a]$$

$$\mathbb{H} = \frac{1}{\Delta V^{W_v} \times d^{W_d}} \quad [17b]$$

$$where, d = ||\boldsymbol{x} - \boldsymbol{x}_T|| \quad [17c]$$

where, $\boldsymbol{x}$ is the full state vector associated with a particular node, and $\boldsymbol{x}_T$ is that of the target condition(s). The heuristic, $\mathbb{H}$ is crafted to account for two aspects during the decision making process for selecting a node from a frontier: (a) the discontinuities in velocity or mass loss between selected arc segments (ΔV, m_f), and (b) a measure of goodness of the selected node in terms of its proximity to the target condition(s), d. The weights are design variables, where W_v and W_m grant the user control over the desired s/c performance (ΔV and propellant consumption), and W_d influences the manner in which the transfer trajectory tends towards the destination. These qualities also render the heuristic applicable as a measure of reward for a select number of agents ranked by their overall performance (Eqn. 16) for successful completion of the transfer. Specifically, the reward is computed as the accrued sum of the heuristic values ($\mathbb{R} = \sum \mathbb{H}$) along a successful transfer path, and is assigned to each contributing node. The sheer volume of the state-space in this investigation has prompted the tracking of a node's relevance to the transfer by recording its cumulative rewards over the episodes within a global repository. This implementation is a variation on the traditional approach in reinforcement learning algorithms where a table of the link quality between the nodes is maintained. As a result, the implication of an identical link quality to a particular node from any other is allayed by enforcing ARs that constrain the accessibility between nodes. This adjustment also influences the discount factor γ to be set equal to zero such that only the immediate rewards are solely valued. This decision is also rationalized by the fact that the node-to-node cost is variable as a function of the s/c mass history. Furthermore, the quality of a node is considered only as a guide in the overall search strategy, as stochasticity is introduced in the selection of a favorable node from within an AR with probability $\mathscr{P}$:

$$\mathscr{P} = \frac{\mathbb{F}_i \ \mathbb{H}_i}{\sum \mathbb{F}_i \ \mathbb{H}_i} \quad [18]$$

This probability $\mathscr{P}$ emphasizes two important considerations which bias the selection of a node i: (a) consistency of the relevance of the node to the transfer via its accrued rewards over prior episodes ($\mathbb{F}_i$), and (b) its current attractiveness as represented by the heuristic $\mathbb{H}_i$. The aforementioned measures are enforced within a distributed, cooperative environment where the incremental knowledge imbibed by each agent combines to furnish the final transfer trajectory that satisfies the goals and constraints imposed by the user.

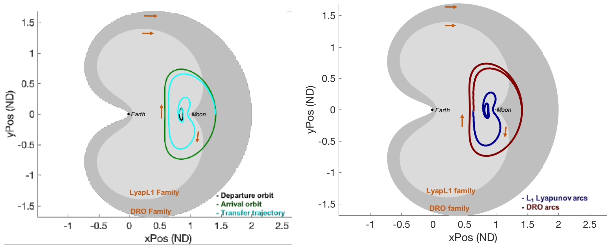
Initial Guess to Higher Fidelity Solution

In this investigation, a final solution is defined as an end-to-end transfer trajectory which adheres to the constraints imposed within a CR3BP regime augmented by the chosen engine forces. Note that such a solution is still merely an initial guess for simulation in an even higher fidelity ephemeris model. The information associated with the arcs assembled by the automated pathfinding process is sufficient to initiate the numerical corrections process described in Section 2, namely the position, velocity, mass, thrust direction, thrust magnitude and time estimates. The user possesses the freedom to choose a desired computational tool to execute the convergence/optimization process. It is important to acknowledge that varying thrust magnitudes are interspersed within the lower-fidelity solution, and so it can serve as in initial guess for a range of engine capabilities in a higher fidelity simulation. However, caution must be exercised when setting the thrust levels during the numerical corrections process because this acceptable range of thrust magnitudes is finite.

6. Results

The execution of the steps outlined in the design framework engender initial guesses for diverse transfer scenarios and are illustrated in this section. The process appropriated for transitioning these solutions to a higher fidelity engine model is also highlighted. The examples have been specifically chosen to demonstrate the flexibility and potential of the automated search strategies in satisfying varied mission objectives.

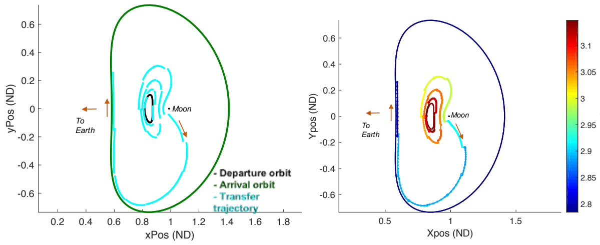
A first measure is the validation of the design process by establishing a proof-of-concept; i.e. ascertaining whether the proposed framework is indeed capable of delivering an initial guess. Figure 8(a,b) illustrate the transfer arcs resulting from soliciting transportation from an L_1 Lyapunov orbit to a DRO using Dijkstra's algorithm. According to the methodology described in Section 5, these arcs are constructed by the following steps: identification of the departure and arrival conditions, collation of a database of discretized L_1 Lyapunov and DRO conditions, determination of a user-defined engine-on and horizon time to support the computation of ARs per the s/c specifications, and initiation of a search sequence with the exact or heuristic pathfinding algorithm of interest. Finally, the initial guess is transitioned to a higher fidelity engine model to construct a converged and/or optimized end-to-end transfer path.



(a) Transfer arcs (b) Colored by family

Fig. 8: Demonstrating feasibility with Dijkstra's algorithm. Initial guess for transferring s/c between an L_1 Lyapunov orbit ($JC = 3.147$) to a DRO ($JC = 2.785$). s/c acceleration $= 9.7 \times 10^{-4} m/s^2$. Horizon time $= 0.5$ *days*.

The transfer in Figure 8(a) spirals out from the departure condition, which is a predictable geometry given that only the L_1 Lyapunov and DRO family structures are incorporated within the searchable database. Figure 8(b) identifies the types of intermediate arcs caught during the transfer - the s/c slides through various members of the L_1 Lyapunov family prior to drifting into the DRO family for the terminal orbit entry phase. As discussed in Section 4, Dijkstra's method is effective in finding an end-to-end transfer if it exists, but does so at the cost of high time and space complexity. The computational time for the solutions in Figures 8 is $\approx 6hrs$ and so, the HARL algorithm is sought to construct most of the remaining examples investigated in this study. The transfer behavior in Figure 9 is similar to that observed in Figure 8 and serves to validate the outcome of implementing the HARL algorithm. This solution was obtained in ≈ 6.5 *minutes*.

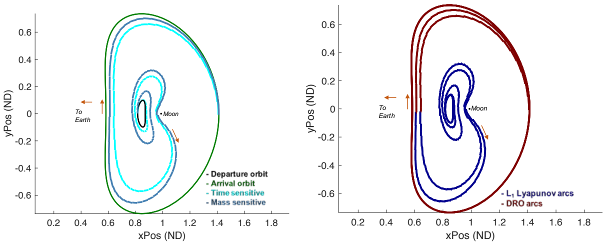


(a) Transfer arcs (b) Colored by JC

Fig. 9: Demonstrating feasibility with HARL algorithm. Initial guess for transferring s/c between an L_1 Lyapunov orbit ($JC = 3.147$) to a DRO ($JC = 2.785$). s/c acceleration $= 2.2 \times 10^{-4} m/s^2$. Horizon time $= 2.5$ *days*.

This transfer is performed using a s/c with acceleration levels comparable to that of DS1. Smaller acceleration levels induce smaller ARs for an equal time-horizon. So, this example exhibits the design option to increase the time-horizon to effect more sizable ARs and avail of a greater quantity of natural arcs to choose from within a frontier. The flexibility to control the time-horizon is also advantageous if confronted with a sparse database. However, the ability to choose conditions from a bigger AR also allows larger discontinuities between the arcs. If $\Delta V s$ are employed to approximate low thrust arcs, care must be exercised to control the propagation time so as to not exceed the horizon of validity for the low-thrust model; establishing appropriate values for this design parameter is a matter of tuning. Figure 9(b) illustrates the ability of the automated search to construct transfer geometries whilst also taking advantage of beneficial energy transition strategies such as exploiting the Oberth effect to enable an efficient energy change near the Moon. The other large energy change is observed during the transition between the two families of interest. The stopping condition for the algorithm is defined to be the first time any element of the target orbit falls within an AR and becomes a candidate for selection by an agent. So, the transfers are free to enter the target orbit at any location as long as the user-specified constraints and objectives are satisfied.

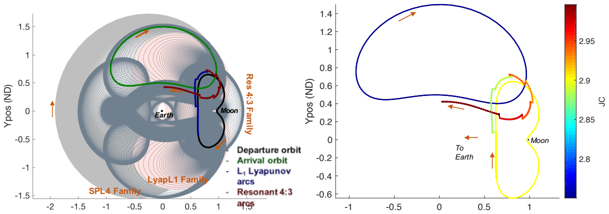
The mass-time trade is an important consideration in the design of a trajectory. So, as discussed in Section 5, the weights are adjusted in Eqn. [16] to deliver transfers with varied performance. In Figure 10, both trajectories catch different members along the families in the database to address the imposed objective function. The trajectory with greater emphasis placed on conserving propellant mass is constructed with more segments, thereby taking longer but also offers reduced propellant usage. In both cases, the algorithm deems the x-axis crossing a beneficial location to transfer to the DRO family. Again, a flyby of the Moon rapidly increases the energy to be able to slide through larger members of the Lyapunov family and reach the target orbit, rather than spiraling through additional revolutions to gradually increase the energy to that of the target orbit. The similar geometries in both transfers is an attribute of the limited variability of arcs included in the database; introducing different families could excite varied behavior when different weights are defined in the objective function. Performance measures such as TOF and Delivered Mass Fraction (DMF) are reported in the figures that follow.



(a) Transfer arcs (b) Colored by Family

Fig. 10: Demonstrating control over mass-time priority for trajectory design. Initial guess for transferring s/c between an L_1 Lyapunov orbit ($JC = 3.147$) to a DRO ($JC = 2.785$) using Dijkstra's algorithm. s/c acceleration $= 9.7 \times 10^{-4} m/s^2$. Horizon time $= 0.5$ *days*.Time-sensitive scenario: Estimated TOF ≈ 44.5 *days*, DMF $\approx 96.4\%$. Mass-sensitive scenario: Estimated TOF $\approx$ 52 *days*, DMF $\approx 98.8\%$

Recall from Section 5.2 that there exist varying degrees of similarity in characteristics between different families. The L_1 Lyapunov and L_4 short-period families have extremely low connectivity in combined position and velocity space (Table 1). So, it is beneficial to include intermediate families that serve to bridge the gap in the geometry and energy levels required to transition between these otherwise poorly connected family pairs. Not surprisingly, in Figure 11 , the automated framework successfully determines that there is value in catching the Resonant 4:3 arcs to transfer between the Lyapunov family departure and short-period family arrival. Note that unlike in the example discussed in Figure 9(b), the search strategy does not always necessarily maneuver with a monotonic increase or decrease in Jacobi constant values to transition between departure and arrival conditions at different energy levels (See Figure 11). In this example, the search terminates when the arrival conditions fall within the AR of the final arc.



(a) Transfer arcs (b) Colored by JC

Fig. 11: Demonstrating the need for intermediate-connection families. Initial guess for transferring s/c between an L_1 Lyapunov orbit ($JC = 2.90$) to a L_4 short-period orbit ($JC = 2.75$) using HARL algorithm. s/c acceleration $= 2.2 \times 10^{-4} m/s^2$. Horizon time $= 3$ *days*. Estimated TOF ≈ 24 *days*, DMF $\approx 99.3\%$

Traditional mission design strategies often involve constructing forward and reverse time propagations from a selected condition or hyperplane which aids in convergence, or helps meet specific constraints. Figure 12 illustrates that if desired, the automated search strategy can also be implemented in this manner. In this case, only the departure and destination families are included in the database. So, the transfer assumes a varied path and insertion location to that observed in the example in Figure 11. Such a strategy can also be adopted to mitigate the challenges associated with designing transfers between families with poor connectivity as defined in Section 5.2. Establishing a sound cis-lunar transport infrastructure in the near future may require conveyance between orbit ports in varied locations within the Earth-Moon system. For operational purposes, whether it be to mitigate congestion or visit/avoid certain regions in configuration space, manage the duration of travel, or simply explore the available trade-space, it is beneficial to possess the capability to maintain control over the qualitative and quantitative nature of the routes traversed through the system. Figures 13 demonstrate that by including/excluding certain families from the searchable database, the user is able to force an interior or exterior-type transfer based on the needs of the mission. Time-sensitivity is weighted

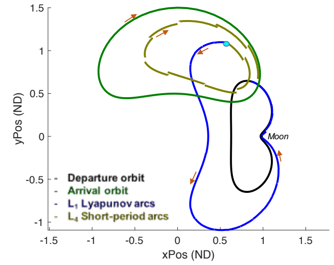
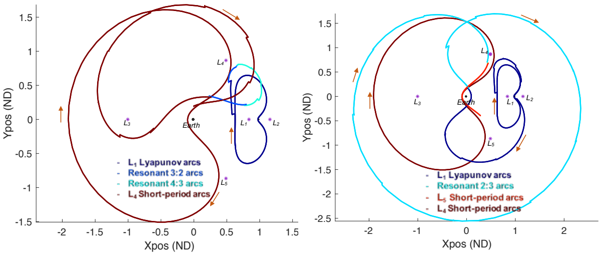


Fig. 12: Demonstrating an alternative implementation for automated pathfinding. Initial guess for transferring s/c between an L_1 Lyapunov orbit ($JC = 2.90$) to a L_4 short-period orbit ($JC = 2.75$) using HARL algorithm, and forward-reverse time propagations. Arcs are colored by family. s/c acceleration $= 2.2 \times 10^{-4} m/s^2$. Estimated TOF $\approx$ 33 *days*, DMF $\approx$ 98.6% Forward propagation horizon time = 3 *days*. Reverse propagation horizon time = 2 *days*.

more heavily in both results (Eqn. 16). Note that choosing the interior transfer clearly offers improved time-savings for a similar mass budget.

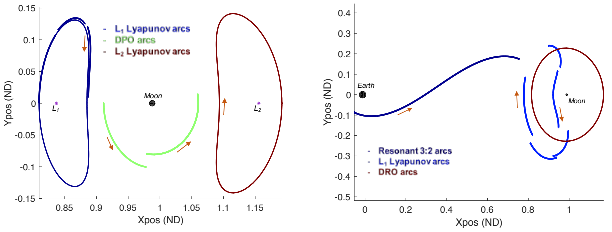


(a) Interior transfer

(b) Exterior transfer

Fig. 13: Demonstrating control over transfer geometry and performance. Initial guess for transferring s/c between an L_1 Lyapunov orbit ($JC = 2.90$) to a L_4 short-period orbit ($JC = 1.80$) using HARL algorithm. Arcs are colored by family. s/c acceleration $= 2.2 \times 10^{-4} m/s^2$. Horizon time = 3 *days*. Interior transfer: Estimated TOF $\approx$ 51 *days*, DMF $\approx$ 98.6%; Exterior transfer: Estimated TOF $\approx$ 75 *days*, DMF $\approx$ 98.1%

From the examples discussed thus far, it is evident that the automated pathfinding framework functions in diverse scenarios. So, it is also interesting to investigate whether examples similar to prior or proposed real-world mission scenarios is replicable. The transfer observed in Figure 14(a) is similar to the Libration Point Orbit (LPO) to LPO transfer scenario devised across the line of the Moon for the Artemis mission. The discontinuities are fairly large, but is expected due to the energy boosts received from the Moon in this sensitive region. This transfer also demonstrates that the framework is not restricted to finding pathways only between departure and arrival conditions at different energy levels. The example in Figure 14(b) crafts a transfer from an arbitrary condition in the Earth's vicinity to a DRO with characteristics similar to that sought for the upcoming EM-1 mission. Such a capability is very valuable, especially when faced with launch delays, uncertainties in deployment states (e.g. for secondary payloads), or for executing contingency options.



(a) LPO to LPO

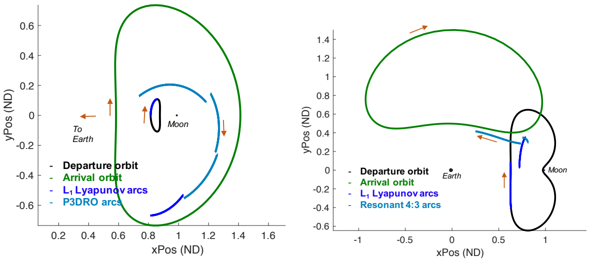
(b) Earth to lunar vicinity

Fig. 14: Demonstrating ability to execute transfers similar to real-world mission examples. Initial guess for transferring s/c between (a) an L_1 Lyapunov orbit ($JC = 3.12$) to a L_2 Lyapunov orbit ($JC = 3.12$), and (b) Earth altitude of $\approx 35,000km$ to DRO ($JC = 2.94$) using HARL algorithm. Arcs are colored by family. s/c acceleration $= 2.2 \times 10^{-4} m/s^2$. LPO-LPO transfer: Horizon time = 2 *days*., Estimated TOF$\approx$ 10 *days*, DMF$\approx$ 99.8%, Earth-DRO transfer: Horizon time = 3 *days*., Estimated TOF $\approx$ 18 *days*, DMF $\approx$ 99.6%

The capabilities discovered in the low-thrust regime are extended to the chemical realm as well. The transfer scenario presented in Figures 9 and 11 are generated for a 500kg s/c with an $Isp = 224$ *seconds* and $Thrust = 18N$. Such an effort is not intended for a direct 1:1 comparison between low-thrust and chemical engine performance. Rather, the same transfer scenarios are chosen merely to illustrate the ability to successfully extend the automated IG generation process across varied engine platforms. It is also an opportunity to reflect on the varied geometries resulting from allowing for larger discontinuities due to the greater thrust capabilities (and therefore greater ARs) accomplished with a chemical engine.

6.1 Transition to higher fidelity engine model

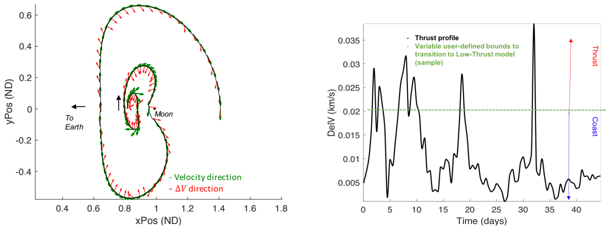
The initial guesses formulated for the various scenarios thus far are subjected to a numerical convergence and optimization process to obtain a higher fidelity solution and meet any additional constraints. The natural arcs therefore deform under the influence of the true behavior of the engine parameters introduced in an updated dynamical model. Recall that one objective of this study is to assess the generality of the framework across varied engine platforms



(a)

(b)

Fig. 15: Demonstrating extension of automated pathfinding capability to varied engine types. (a) Initial guess for transferring s/c between an L_1 Lyapunov orbit ($JC = 3.147$) to a DRO ($JC = 2.785$). (b) Initial guess for transferring s/c between an L_1 Lyapunov orbit ($JC = 2.90$) to a L_4 short-period orbit ($JC = 2.75$) using HARL algorithm. s/c acceleration $= 0.036 m/s^2$. Horizon time $= 3$ *days*.



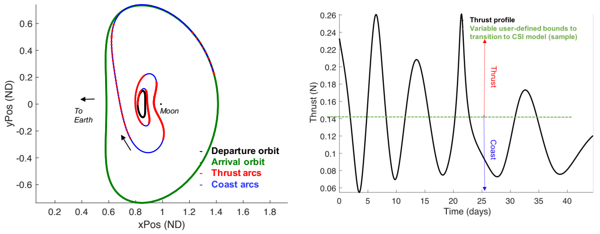
(a) $\boldsymbol{\Delta V}$ profile

(b) ΔV profile

Fig. 16: Information extracted from lower-fidelity transfer scenario in Figure 8 to support transition to low-thrust model.

(VSI, CSI, chemical) and optimization routines (direct and indirect). This avenue of investigation is commenced by extracting some useful items of information from the lower-fidelity pathways constructed thusfar. Figure 16(a) superimposes the velocity and ΔV directions undertaken by an agent during the search process to chain together a transfer sequence. Figure 16(b) presents the history associated with the magnitude of this thrust profile along the trajectory. It is computed by interpolating between the ΔV values recorded at the beginning of the discrete segment times. The highest peaks in this figure remain close to $38 m/s$, which is the ΔV_0 value calculated using Eqn. [15] for the specifications of an engine (0.22N Thrust, 4000s Isp) employed to generate the first transfer segment for a s/c weighing 1000kg. The ΔV bounds for the subsequent segments also remain close to this value due to efficient low-thrust engine propellant consumption, and therefore, relatively unchanged s/c characteristics through the transfer. So, the variability in the ΔV magnitudes in Fig. 16(b) demonstrate that the stochastic element employed in the RL framework enables initial conditions to be chosen from varying radii within a given AR bound, as required.

Together, the position, velocity, time-history and thrust magnitude and direction information constitute all the fragments of information required to initiate a numerical targeting process for the higher fidelity regimes. As discussed in Section 3.2, the VSI numerical targeting process is quite robust. So, the velocity pointing- or thrust-pointing directions serve as an adequate guess for the co-states (λ_{v_i}) in establishing control authority for maneuvering the engine. The variable Isp and thrust magnitudes for this engine model may elicit a converged solution which exhibits varied behavior from the initial guess formulated using a constant thrust and Isp value. During the convergence process, allowing the departure and arrival locations to vary on their respective orbits also allows the geometry to deviate from the original guess. Figure 17(a) illustrates the path resulting from converging the initial guess in Figure 8 using a VSI engine and enforcing position and velocity continuity throughout. The associated thrust profile is illustrated in Figure 17(b). The portions of the trajectory with exceedingly low thrust magnitudes are induced from high Isp values ([4]) and are analogous to coasting regions in a CSI model.

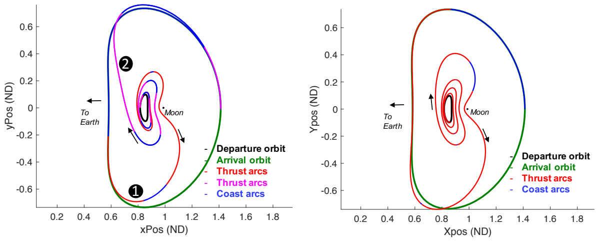


(a) VSI transfer

(b) Thrust profile

Fig. 17: Converged VSI solution of lower-fidelity transfer scenario in Figure 8. TOF $= 45$ *days*, DMF $= 99.14\%$

Transition of the initial guess to the CSI model is enabled via two approaches. Both the lower-fidelity and VSI solutions offer thrust magnitude and direction histories which can be exploited to seed the thrust and coast segments required to initiate the CSI targeting process. For example, the thrust segments can be distributed by bounding their occurrence to exist within a certain percentage of the maximum amplitude of the thrust profiles (Figure 16(b) and 17(b)). That is, the s/c coasts below the user-defined bound and thrusts above it. During the convergence process, a point-and-hold strategy for the thrust-pointing directions support the *direct* targeting process sought in this study for the CSI model. The imposed path constraints, user-defined thrust cut-off bounds (Figure 16(b) and Figure 17(b)), duration of each constant thrust-pointing segment, and the flexibility allowed for the departure and arrival locations, are all factors that can be exploited to manipulate the qualitative and quantitative nature of the converged transfer path. Figure 18 illustrates such a manipulation which results in pathways that are forced to resemble either the lower-fidelity initial guess (Figure 18(a1)) or the VSI solution (Figure 18(a2)). Note that the initial guess formulated with an acceleration magnitude of $a = 9.7 \times 10^{-4} m/s^2$ (Figure 8) is converged with a more realistic acceleration magnitude of $a = 2.2 \times 10^{-4} m/s^2$ in Figure 18. The multiple low ΔV/thrust magnitude regions in this initial guess (Figure 16(b)) result in multiple coasting arcs for the original s/c. The same arcs facilitate the net thrusting time required by a s/c with weaker acceleration levels to repair the discontinuities prevalent during a particular segment. Therefore, there exists a lower bound to the s/c acceleration levels for which a particular lower-fidelity solution can serve as a feasible initial guess.



(a) CSI transfer $a = 2.2mN$

(b) CSI transfer $a = 1.2mN$

Fig. 18: Converged CSI transfer solution for varied thrust profiles and acceleration levels. (a1) $TOF = 49 days$, $DMF = 98.5\%$. (a2) $TOF = 45 days$, $DMF = 98.9\%$. (b) $TOF = 84 days$, $DMF = 97.6\%$

The natural arc sequences (e.g. Figures 11, 13 and 14) prove to be adequate initial guesses to support the development of optimized CSI low-thrust results for distinct transfer scenarios (Figures 19). They are mass-optimal trajectories, where the transfer duration and departure/arrival conditions are offered as free-variables to the non-linear optimization process. An alternative is to constrain and control these design variables to influence the thrust durations and mass performance; i.e. the outcome of the optimization process is a function of the problem set-up consisting of the designated design variables and constraints.

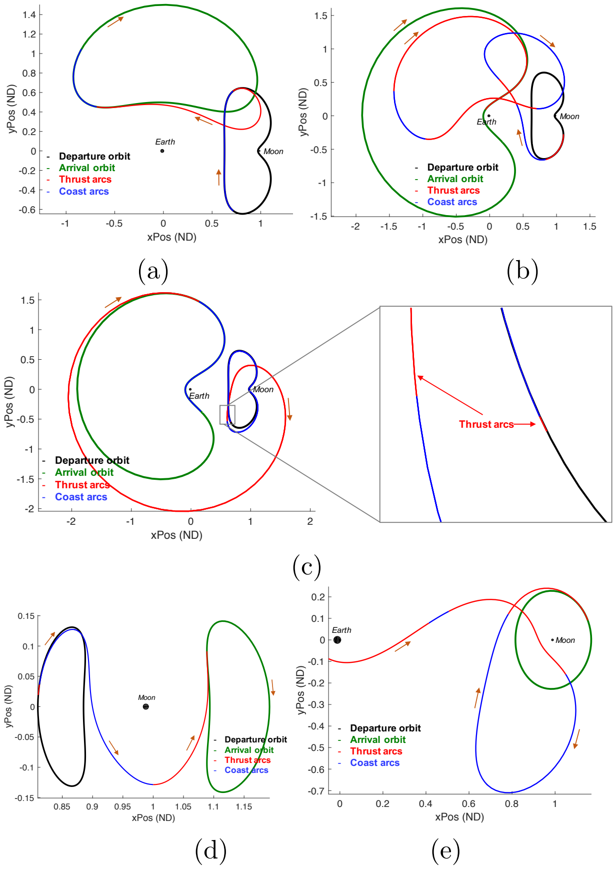
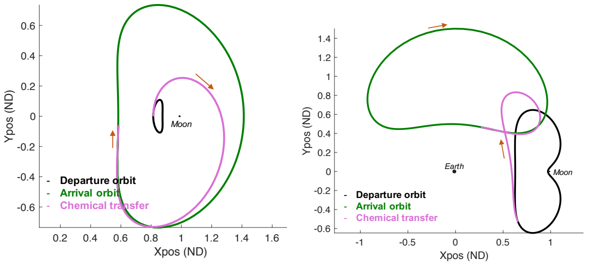


Fig. 19: Optimized CSI solutions for various transfer scenarios. (a) LyapL1 to SPL4 ($JC_{Dep} = 2.90$,$JC_{Arr} = 2.75$); (b) LyapL1 to SPL4 ($JC_{Dep} = 2.90$,$JC_{Arr} = 1.80$); (c) LyapL1 to SPL4 ($JC_{Dep} = 2.90$,$JC_{Arr} = 1.80$); (d) LyapL1 to LyapL2 ($JC_{Dep} = 3.12$,$JC_{Arr} = 3.12$); (e) Earth to DRO ($JC_{Dep} = 2.96$,$JC_{Arr} = 2.94$)

Figure 20 shines light on the characteristics of the ΔV optimized trajectories constructed from the chemical IGs presented in Figures 15. Recall that even though the discontinuities seem large in the IG, the ARs are computed using knowledge of the engine capabilities. The ability to successfully converge and produce optimized results are therefore a reflection on the competence of the adopted methodology.



(a) (b)

Fig. 20: Optimized Chemical solutions of lower-fidelity transfer scenarios in Figures 15

A comparison of the estimated and optimized performance metrics is presented in Table 2. The mass estimates are certainly more precise, although the estimates in TOF are still in the general vicinity of the optimized results. This observed discrepancy in the transfer duration results from constructing a mass-optimal final solution in contrast to the multi-objective IG generation process. The optimization process therefore incorporates variable TOF and variable locations on the departure/arrival orbits, introducing additional coast arcs as required, to maximize the delivered mass. These results also support an important motivation for this study - exploiting the natural dynamics (coast arcs) proves to benefit propellant savings and therefore, cost as well. So, if there is a need to build additional coasting capability directly into the IG generation process, the user is also able to alter the coefficients in Eqn [16] to place greater emphasis on propellant conservation (e.g. Figure 10). A noteworthy point is that a similar DMF in the comparisons in Table 2 suggests that the thrusting time alone is comparable in both the IG and optimized results. These take-aways authenticate the ability of the lower-fidelity approach to offer end-to-end transfer solutions. The user is also able to control the qualitative and quantitative nature of these solutions with confidence.

Table 2: A comparison of the performance metrics from estimated and optimized approaches

Scenario	TOF˙opt (days)	TOF_{est} (days)	DMF_{opt} (%)	DMF_{est} (%)
(a)	30	24	99.2	99.3
(b)	47	51	98.6	98.6
(c)	74	75	98.2	98.1
(d)	14	10	99.8	99.8
(e)	29	18	99.5	99.6

7. Conclusions

Trajectory design is a careful balancing act that juggles diverse constraints, priorities and requirements to enable successful missions. So, forthcoming large-scale efforts to expand the boundaries of space exploration by initiating an efficient cis-lunar transportation infrastructure necessitates a rigorous mission-design framework to offer solutions for varied spacecraft classes and objectives. Maneuvering in close proximity of the Earth and the Moon renders the CR3BP model appropriate for capturing the complex dynamical interactions in this gravitational system, and facilitating pathways for preliminary mission design. However, unlike in the two-body problem, the CR3BP entails a large state-space for which approximations are generally not available. This situation gives rise to an infinitely large combinatorial optimization problem - one that is intractable to explore thoroughly via a manual approach. So, an automated search strategy is sought where human efforts are refocussed on defining the constituents of the broader mission goals, and software agents are tasked with undertaking the laborous task of filtering various trajectory scenarios to then present only attractive options. The design of end-to-end trajectories is facilitated by constructing a framework composed of four essential steps - (1) compilation of natural flows in the CR3BP serving as a database of potential transfer pathways, (2) simulating the *reach* of the s/c to assess the accessible regions within the database, (3) implementing automated pathfinding algorithms to sequence the natural arcs and formulate a discontinuous yet complete route to the destination, and (4) transitioning the solution to a higher fidelity engine/dynamical model via a numerical corrections process. The choice of the automated pathfinding phase is influenced by the desired aims and the computational resources available to the user; both an exact and heuristic approach are examined in this investigation.

The results presented in this analysis demonstrate the feasibility and diverse capabilities of the automated framework. With relevance to a cis-lunar architecture, the user is able to design transfers between various locations within the Earth-Moon system. The geometry of the solution and performance traits related to mass-time priority are also influenceable by constraining the types of arcs included in the database, and manipulating the design parameters. The search strategy exhibits the ability to make intelligent choices based on user preferences, such as exploiting the Oberth effect when applicable to preserve propellant consumption. Differing implementations also render the methodology flexible to the incorporation of varied constraints during the design process; e.g. targeting specific arrival conditions or maintaining a minimum altitude from a primary. Furthermore, the mass-time reports from the sequencing of natural arcs have proven to be reliable estimates of the performance metrics following the optimization process.

The available state and time histories from the sequenced natural arcs position them to be suitable initial guesses for transition to a higher fidelity engine model. Even the ground work for advantageous thrust and coast locations is informed by the freedom to construct transfer segments with nil to a maximum allowed position and velocity discontinuity within the ARs. The ability to generalize this design capability across various engine platforms is realized by the ability to incorporate the *reach* of the spacecraft during the automated search process. The capacity to explore a large trade-space and test a variety of combinatorial options extracts the design process from the common restrictions of a narrow convergence radii associated with traditional trajectory design approaches.

The essence of the automated trajectory design framework is to offer a solution to the challenges associated with a large trade-space for mission design. The devised methodology is also applicable for contingency planning, and recovering from situations involving programmatic delays and unexpected events such as performance degradation during the operations phase. The cadence set in motion via the automated approach does not replace human input. Instead, it is intended to offer a deeper understanding of the design environment and therefore, improve human intuition - a critical component in the decision making process for assembling a successful mission.

8. Acknowledgements

This investigation is conducted in conjunction with the NASA Goddard Spaceflight Center, through NASA Grant NNX17AH89G. The authors also wish to acknowledge the facilities at the Purdue Eliasen Visualization Laboratory, and thank the Purdue Multi-Body Dynamics research group for valuable insights and suggestions.

References

[1] B. F. Kutter and G. F. Sowers, Cislunar-1000: Transportation supporting a self-sustaining Space Economy, *AIAA Space*, Long Beach, California, 2016.

[2] D. C. Folta, N. Bosanac, D. Guzzetti, K. C. Howell, “An Earth-Moon System Trajectory Design Reference Catalog,” *Acta Astronautica*, Vol. 100, pp. 341-353, May-June 2015.

[3] D. Guzzetti, N. Bosanac, A. Haapala, K. C. Howell and D. Folta, “Rapid Trajectory Design in the Earth-Moon Ephemeris System via an Interactive Catalog of Periodic and Quasi-Periodic Orbits.,” *Acta Astronautica*, Vol. 126, pp. 439-455, September - October 2016.

[4] M. Vaquero and K. C. Howell, “Leveraging resonant orbit manifolds to design transfers between libration point orbits in multi-body regimes,”*Advances in the Astronautical Sciences*, Vol. 148, No. 4, 2013, pp. 1999-2018.

[5] Haapala, A., and Howell, K., “Representations of Higher-Dimensional Poincar Maps with Application to Spacecraft Trajectory Design,” *Acta Astronautica*, Vol. 96, March-April 2014, pp. 23-41.

[6] F. Topputo, M. Vasile, and F. Bernelli-Zazzera, “Low Energy Interplanetary Transfers Exploiting Invariant Manifolds of the Restricted Three Body Problem,” *The Journal of Astronautical Sciences*, Vol. 53, No. 4, October-December 2005, pp. 353-372.

[7] G. Mingotti, F. Topputo and F. Bernelli-Zazzera, “Efficient Invariant-Manifold, Low-Thrust Planar Trajectories to the Moon,” *Communications in Nonlinear Science and Numerical Simulations*, Vol. 17, No. 2, February 2012, pp. 817-831.

[8] R. Prichett, K.C. Howell, and D. Grebow, “Low-Thrust Transfer Design Based on Collocation Techniques: Applications in the Restricted Three-Body Problem,” *AAS/AIAA Astrodynamics Specialist Conference*, Columbia River Gorge, Stevenson, Washington, August 21 - 24, 2017.

[9] G. Radice and G. Olmo, “Ant Colony Algorithms for Two-Impulse Interplanetary Trajectory Optimization”, *Journal of Guidance, Control and Dynamics*, Vol. 29, No. 6, 2006, pp. 1440-1444.

[10] M. Ceriotti and M. Vasile, “MGA Trajectory Planning With an ACO-Inspired Algorithm,”*Acta Astronautica*, Vol. 67, No. 9-10, 2010, pp. 1202-1217.

[11] J. Stuart, K. C. Howell, R. Wilson, “Design of End-to-End Trojan Asteroid Rendezvous Tours Incorporating Potential Scientific Value,” *Journal of Spacecraft and Rockets*, Vol. 53, No. 2, 2016, pp. 278-288.

[12] R. Furfaro and R. Linares, "Waypoint-Based Generalized ZEM/ZEV Feedback Guidance for Planetary Landing via a Reinforcement Learning Approach," *3rd IAA Conference on Dynamics and Control of Space Systems*, Moscow, Russia, May 30 - June 1, 2017.

[13] R. Beeson, V. Shah, J. Aurich, D. Ellison, "Automated Solution of Low Energy Trajectories," *AAS/AIAA Astrodynamics Specialist Conference*, Columbia River Gorge, Stevenson, Washington, August 21 - 24, 2017.

[14] G. A. Tsirogiannis, "A Graph Based Method for Mission Design,"*Celestial Mechanics and Dynamical Astronomy*, Vol. 114, No. 4, 2012, pp. 353-363.

[15] E. Trumbauer and B. Villac, "Heuristic Search-Based Framework for Onboard Trajectory Redesign," *Journal of Guidance, Control and Dynamics*, Vo. 37, No. 1, 2014, pp. 164-175.

[16] N. Parrish, "A* Pathfinding for Continuous-Thrust Trajectory Optimization," *AAS 37th Annual Guidance & Control Conference*, Breckenridge, January 2014.

[17] J. A. Sims and S. N. Flanagan, "Design of Low-Thrust Interplanetary Missions,"*AAS/AIAA Astrodynamics Specialist Conference*, Girdwood, Alaska, 1999.

[18] V. Szebehely, "Theory of Orbits," *Academic Press*, New Haven, Connecticut, 1^{st} edition, 1967.

[19] C. Saltzer, R. Craig, C. Fetheroff, "Comparison of Chemical and Electric Propulsion Systems for Interplanetary Travel," *Proceedings of the IRE*, Vol. 48, No. 4, pp.465-476, April 1960.

[20] F. R. Chang Díaz, "An overview of the VASIMR engine: High power space propulsion with RF plasma generation and heating," AIP Conference Proceedings 595, doi: 10.1063/1.1424142, 2001.

[21] F. Topputo and C. Zhang,"Survey of Direct Transcription for Low-Thrust Space Trajectory Optimization with Applications," *Abstract and Applied Analysis*, Vol. 2014, No. 2, pp. 1-15, 2014.

[22] J. Stuart, K. C. Howell, R. Wilson, "Automated Design of Propellant-Optimal, End-to-End, Low-Thrust Trajectories for Trojan Asteroid Tours," *Journal of Spacecraft and Rockets*, Vol. 51, No. 5, September 2014, pp. 1631-1647.

[23] J. -P. Marec, "Optimal Space Trajectories," *Elsevier*, Amsterdam, pp. 54, 1979

[24] R. P. Russell, "Primer Vector Theory Applied to Global Low-Thrust Trade Studies," *Journal of Guidance, Control and Dynamics*, Vol. 30, No. 2, March-April 2007, pp. 460-472.

[25] I. Gribkovskaia, Ø. Halskau, Sr. G. Laporte, "The bridges of Königsberg-A historical perspective," *Networks, An International Journal*, Vol. 49, No. 3, January 2007, pp. 199-203.

[26] D. L. Poole and A. K. Mackworth, "Artificial Intelligence, Foundations of Computational Agents," *Cambridge University Press*, New York, 2010.

[27] S. Dasgupta, C. Papadimitriou, U. Vazirani, "Algorithms," *McGraw Hill*, Boston, 2006;

[28] R. A. C Bianchi, C. h. C. Ribeiro, A. H. R. Costa, "On the Relation between Ant Colony Optimization and Heuristically Accelerated Reinforcement Learning," *1st International Workshop on Hybrid Control of Autonomous Systems - Integrating Learning, Deliberation and Reactive Control*, Pasadena, California, July 2009.

[29] B. Fernandez-Gauna, I. Etxeberria-Agiriano, M. Graña, "Learning Multirobot Hose Transoportation and Deployment by Distributed Round-Robin Q-Learning," *PLoS One*, Vol. 10, No. 7, 2015.