75th International Astronautical Congress (IAC), Milan, Italy, 14-18 October 2024.

IAC–24–C1.7.4.83115

Cislunar Trajectory Design and Maneuver Autonomy for NASA's Moon to Mars Architecture

Aaron Houin[a*], **Rohan Sood**[b]

[a] *Mission Designer, NASA Marshall Space Flight Center, Martin Rd SW, Huntsville, AL 35808, and Ph.D. Student, Astrodynamics and Space Research Laboratory, Aerospace Engineering and Mechanics, University of Alabama, Tuscaloosa, AL 35487, aaron.j.houin@nasa.gov*

[b] *Associate Professor, Astrodynamics and Space Research Laboratory, Aerospace Engineering and Mechanics, University of Alabama, Tuscaloosa, AL 35487, rsood@eng.ua.edu*

[*] *Corresponding author*

NASA's Moon to Mars architecture is an ambitious roadmap of manned cislunar and deep space exploration. The extensive amount of orbital assets required will place a significant burden on ground-based resources, such as communication networks and operations facilities. Spacecraft autonomy is essential for maintaining a vast number of complex missions beyond Earth orbit. To achieve full autonomy, spacecraft must be able to employ methods of robust maneuver design without an explicit dependence on commands sent from the ground. This level of autonomy is needed not only for stationkeeping, but also for outbound and inbound transfers. To address the need of spacecraft maneuver design autonomy, this work investigates the use of neural networks (NNs) in a supervised learning environment. A supervised learning approach for NNs allows for a curated training data set, consisting exclusively of perturbations applied to a desired mission concept of operations (ConOps). The proposed approach allows humans on the ground to design a specific mission ConOps before flight, then employ NNs to fly the mission robustly and autonomously. This investigation numerically tests maneuver autonomy in four highly sensitive regions of flight: orbit raising, translunar injection burns, powered lunar flybys, and invariant manifold insertion burns. These straining cases are contextualized by testing them in a demonstration mission, targeting an Earth-Moon L_3 orbit. The study first establishes feasibility by automating impulsive burn maneuvers. However, some guidance algorithms will need more intensive commands, such as inertial pointing and angular rates. To validate this method, NN maneuver autonomy is applied to a finite burn model of the demonstration mission. The use of sequential, mission specific maneuvers provide an appropriate testbed to demonstrate the robustness of a NN trained on feasible perturbed states. Moreover, these scenarios provide preliminary proof-of-concept for fully autonomous missions that execute maneuvers without dependence upon explicit command uplinks. As a result, the technological advancement proposed in this work may significantly ease the strain on ground-based mission operations. This would enable complex and autonomous mission execution in cislunar and deep space regimes, filling a technology gap required to support future manned missions.

Nomenclature

ΔV - Change in velocity
Δt - Change in time
h_i - Neural network hidden layer
J - Neural network *Loss* function
I_{sp} - Specific impulse
$\vec{r}$ - Position states
$\vec{v}$ - Velocity states
$\vec{u}$ - Finite burn control vector

Acronyms/Abbreviations

M2M - Moon to Mars
ConOps - Concept of Operations
NN - Neural network
CR3BP - Circular Restricted Three-Body Problem
EPPR - Ephemeris Pulsing Rotating
DRM - Design Reference Mission
ARB - Apogee Raise Burn
TLI - Trans Lunar Injection
OPF - Outbound Powered Flyby
MI - Manifold Insertion

1. Introduction

Following the successful flight of the Artemis 1 mission [1, 2], NASA has ambitious plans for the future of both scientific and manned space flight. Mission design is well underway for a wide range of upcoming operations, including Artemis flights [3], Human Lander System development [4], and Deep Space Gateway utilization [5]. To coordinate and track objectives and technologies required for lunar and martian mission, NASA formed the Moon to Mars Architecture (M2M) [6]. The M2M ar-

chitecture is a roadmap for long-term space exploration. As such, it contains a wide array of goals and objectives that will require dedicated, space-based orbital assets to complete. The sheer number of spacecraft required to support future exploration goals will drastically increase to provide critical infrastructure, such as lunar navigation [7, 8]. Ground-based communications and navigation facilities, like NASA's Deep Space Network (DSN) are already facing strenuous scheduling constraints to maintain current missions [9]. The multitude of additional cislunar and deep space vehicles needed for the M2M architecture will put significant strain on facilities required for ground-based operations. Spacecraft autonomy could significantly reduce the demand placed on communications and navigation systems.

To achieve fully autonomous spacecraft operations, three major functions would need to be automated: navigation (i.e., estimate spacecraft states), maneuver design (i.e., generate a command for the spacecraft engine), and guidance/control (i.e., executing the command). These components are summarized in the schematic shown in Figure 1. While extensive research has been conducted on autonomous navigation [10–13] and guidance [14, 15], autonomous maneuver design still has much to be explored. This is particularly true for high-thrust transfer trajectory maneuvers. This study focuses on maneuver design autonomy for transfers that fly through highly sensitive dynamical regions. Regions of high sensitivity are of particular interest due to the complex gravitational regime of the cislunar multi-body system, where many M2M spacecraft will be [16, 17].

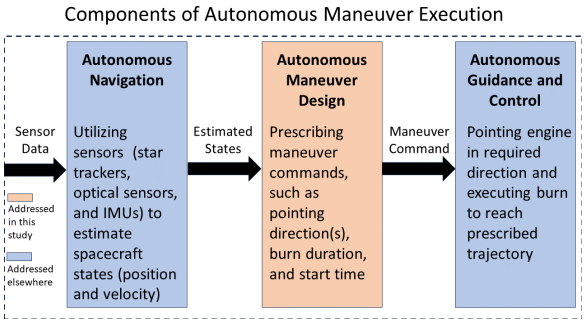

Fig. 1. Components of Autonomous Maneuvers

The ability for spacecraft to make decisions and execute them, without requiring human input, is the overarching goal of autonomy in spaceflight. The most prevalent approach to implement this level of independent decision making is artificial intelligence (AI) and machine learning (ML) [18]. While there are multiple methods for implementing AI, a very common approach is to use artificial neural networks (NNs) [19]. These NNs can be computationally lightweight, yet have proved to be effective in a wide range of astrodynamics applications. Neural networks have been successfully used to map the dynamics of complex orbital regimes [20, 21] and to systematically design transfer orbits [22]. Machine learning has also been shown to have extensive uses for trajectory initial guess generation [23] and path planning [24]. Methods are actively being developed to evaluate and quantify spacecraft autonomy requirements for on-board applications [25, 26]. Finally, NNs have shown promising results when applied to in-situ transfer design; autonomously selecting optimal trajectory arcs and guidance waypoints [27–29] and autonomously performing correction maneuvers [30, 31] are two prominent applications.

Similarly, this study focuses on addressing challenges associated with deploying AI onboard a spacecraft for autonomous trajectory design. However, the focus is on utilizing NNs to autonomously prescribe and execute maneuvers as part of a predefined mission Concept of Operations (ConOps). The goal is not to have AI redesign the transfer trajectory during flight. Rather, NNs are trained to autonomously apply optimal maneuvers subject to perturbations arising from highly sensitive orbital regimes or maneuver execution errors. The end result is an autonomous NN that dynamically updates optimal guidance targets during flight, without depending on maneuver commands from the ground.

2. Materials and methods

This study leveraged multiple tools to assess neural network enabled spacecraft autonomy in highly sensitive regions. A summary of the programs used, along with mission ConOps and assumptions, is detailed in this section.

2.1 Autonomous Maneuver Design Workflow

When a spaceflight mission enters the operations phase, the engineering workload is not finished. Even after initial deployment and checkout procedures, the vehicle will frequently require communications with ground-based assets [9]. From a flight mechanics standpoint, the typical maneuver design and execution workflow for transfer trajectories is as follows [32]: First, ground-based antennas must acquire raw navigation measurements, which are filtered to determine spacecraft estimated states. This process of acquiring tracking data is repeated multiple times according to a navigation cadence, where the prediction/correction process updates with each measurement. Measurements are typically stopped a few hours before the next prescribed maneuver in the mission ConOps. With the state estimated, and a design reference mission (DRM) already defined, engineers on the ground generate the opti-

mal maneuver the spacecraft must execute. Once a maneuver has been designed, ground-based antennas are once again used to communicate with the spacecraft and upload the maneuver to the flight computer. After the vehicle receives the maneuver file, the guidance and control systems execute the burn.

It is important to recall that this process of multiple ground contacts is required for each maneuver, and for all vehicles that are performing transfers to a final target orbit. As such, it becomes obvious that the expected increase in cislunar spacecraft needed to support the M2M architecture will put unprecedented strain on ground-based facilities. This study attempts to explore the feasibility of removing the required communication contact for maneuver uplink by deploying onboard NNs that are already trained on optimized trajectories. This approach allows mission planners to select the ConOps, including the deterministic burns, before flight. The aim is to then have NNs for each maneuver specifically trained on maneuver parameters such as engine pointing direction, execution time, and burn duration. To test the feasibility of such concept, a DRM with multiple maneuvers that occur in highly sensitive regions was developed. Training data was generated by applying statistical perturbations to the DRM initial conditions, then conventional numerical optimization techniques were applied to determine optimal maneuver parameters. The dataset of optimized maneuver parameters was used to train the NNs. This eliminates the need to reoptimize the trajectory on the ground and upload new maneuver files before each burn gets executed. It is important to note that this study assumes perfect navigation state knowledge. Operational spaceflight applications will certainly have navigation error. However, this study seeks to first confirm that NNs can indeed be used to continuously interpolate optimal maneuver parameters in highly sensitive regimes. The impact of navigation error getting passed into the NN input layer, along with how to mitigate resulting downstream errors, is left as future work.

2.2 Trajectory Modeling and Optimization

All trajectory design and numerical optimization was done with the Astrodynamics Software and Science Enabling Toolkit (ASSET). ASSET was developed by the University of Alabama under a NASA ROSES grant [33]. It specializes in solving spacecraft optimal control problems (OCPs) with a direct transcription collocation routine, leveraging a custom dual-primal interior point optimizer [34]. Additionally, ASSET is capable of converging highly sensitive trajectories that arise from the multi-body regime [35], with quick computation speed and parallelizability [36, 37]. Since ASSET is a thread safe optimization software, the parallel computing tool Dask was used to quickly generate large sets of training data [38].

A significant focus of the M2M campaign will be on cislunar exploration and infrastructure. Unlike the relatively stable dynamics of the two-body regime, the Earth-Moon system (and the resulting orbits within the system) can be quite chaotic. A common representation of the cislunar gravitational environment is the Circular Restricted Three-Body Problem (CR3BP). The CR3BP model uses a rotating reference frame, with the origin at the gravitational barycenter of two celestial bodies. This dynamics model assumes that the primary body (P_1) and the secondary body (P_2) orbit around their shared barycenter in circular orbits, with a constant angular rate, ω. The basis vectors of the rotating frame align the x-axis along the line connecting P_1 and P_2, and the z-axis aligned with the system angular momentum. The y-axis completes the right hand set. A third body, often representing a spacecraft or asteroid, is introduced into the model with an assumed mass significantly lower than that of the celestial bodies. The third body's position in the CR3BP, $\vec{r}$, is referenced from the barycenter of P_1 and P_2. Finally, the positions of the third body with respect to the primary and secondary bodies ($\vec{r}_{P1P3}$ and $\vec{r}_{P2P3}$, respectively) are required to derive the equations of motion. Note that the frame is nondimensionalized by the mass ratio of the system, μ. The CR3BP frame is shown in Figure 2, and the equations of motion (EOMs) are shown in Eqn. [1], [2], and [3].

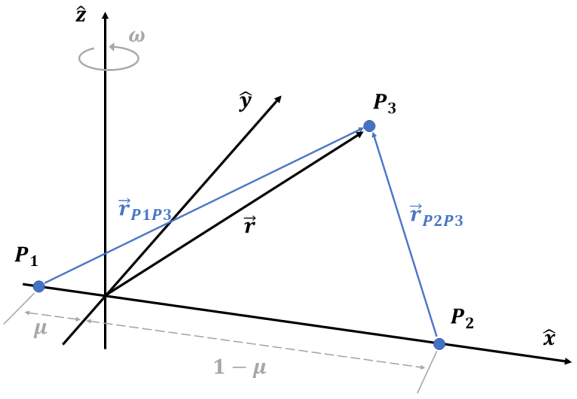

Fig. 2. CR3BP Frame

$$\ddot{x} - 2\dot{y} - x = -\frac{(1-\mu)(x+\mu)}{r_{P1P3}^3} - \frac{\mu(x-1+\mu)}{r_{P2P3}^3} \quad [1]$$

$$\ddot{y} + 2\dot{x} - y = -\frac{(1-\mu)y}{r_{P1P3}^3} - \frac{\mu y}{r_{P2P3}^3} \quad [2]$$

$$\ddot{z} = -\frac{(1-\mu)z}{r_{P1P3}^3} - \frac{\mu z}{r_{P2P3}^3} \quad [3]$$

While the CR3BP is a great approximation of the cislunar environment for initial guess generation, the simplifying assumptions of the frame don't capture the full complexity of the Earth-Moon system. Two significant sources of error between the CR3BP and ephemeris specific dynamics are perturbations from other celestial bodies (especially the Sun) and the eccentricity of the Moon's orbit about Earth. To properly model these nonlinear inputs to the model, the Ephemeris Pulsing Rotating frame (EPPR) was used for all trajectory design and optimization in this study. The EPPR utilizes the same basis vectors ($\hat{x}$, $\hat{y}$, $\hat{z}$) and mass ratio (μ) as the CR3BP, but includes perturbations from other celestial bodies. This relies on JPL SPICE kernels for planetary ephemerides. Additionally, the EPPR accounts for the pulsation of the rotating frame by dynamically scaling the characteristic length of the system, enforcing a nondimensional distance between P_1 and P_2 to be unity at all times.

2.3 Demonstration Transfer Trajectory

To test the viability of applying NNs to the maneuver design process, a cislunar design reference mission (DRM) that passes through multiple highly sensitive dynamical regions is desired. The DRM used to achieve this is a transfer trajectory from an elliptical Earth orbit to an Earth-Moon L_3 Lyapunov orbit, with a lunar flyby. The Lyapunov orbit family consists of planar periodic solutions to the CR3BP and are very unstable [39]. As such, a transfer targeting the stable invariant manifold structure of an L_3 orbit is a great candidate DRM for introducing a series of sensitive maneuvers. This highly sensitive DRM acts as a stress test for how well NNs are able to map the gradients of optimal burn parameters when subjected to statistical perturbations.

The DRM used for this study begins in a highly elliptical Earth orbit, then utilizes a sequence of four maneuvers to transfer out to an L_3 Lyapunov orbit. First, an apogee raise maneuver (ARB) is performed at the periapsis of the initial orbit. After one revolution in the higher energy staging orbit, a translunar injection (TLI) burn occurs, setting the spacecraft on a trajectory towards the lunar flyby. Next, an outbound powered flyby (OPF) of the Moon utilizes the engines to intersect the stable manifold position states. Finally, a manifold insertion (MI) burn occurs, placing the spacecraft on a ballistic path towards the final orbit. Figure 3 shows the DRM trajectory projected onto the x-y plane of the EPPR frame. Additionally, the initial conditions for the Earth orbit are listed in Table 1. Initial conditions are given in the EPPR frame (which shares the same basis vectors and nondimensionality as the CR3BP), hence the Earth-Moon barycenter serves as the origin.

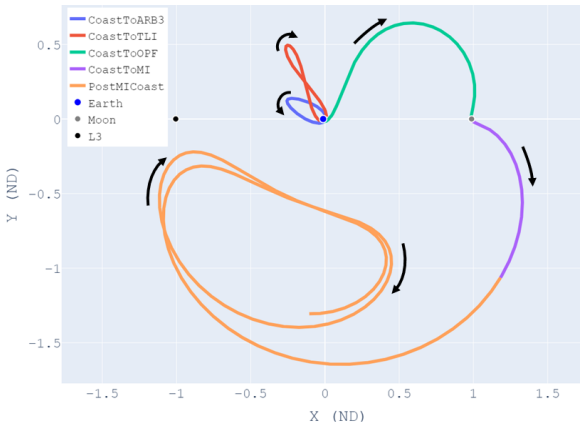

Fig. 3. Design Reference Mission

Table 1. Design Reference Mission Initial Conditions

	Earth-Moon Nondimensional
$\vec{r}_{x0}$	3.94128033e-03
$\vec{r}_{y0}$	2.31562755e-02
$\vec{r}_{z0}$	-5.68687575e-06
$\vec{v}_{x0}$	-2.83677916
$\vec{v}_{y0}$	7.19450850
$\vec{v}_{z0}$	-1.50837840e-3

*Given in the EPPR/CR3BP Frame

2.4 Neural Network Supervised Learning

One of the most common approaches to AI automation is to implement artificial neural networks (NNs). A NN is a numerical structure of nodes, sometimes called neurons, that get aligned into multiple layers. This study uses fully connected neural networks, where any given node is connected to all surrounding nodes by scalar weight values. The fully connected structure first has an input layer (x), where data/measurements are passed into the model. The input layer nodes (x_i) are then fully connected to an arbitrary number of intermediate layers, called hidden layers (h). It has been proven that a single hidden layer of arbitrarily large size can be a universal approximator [40]. However, it is often more computationally efficient to instead have multiple, smaller hidden layers. Finally, after passing through all the hidden layers a specific NN has, the nodes are passed to the output layer (y). A schematic of a fully connected NN with two hidden layers is shown in Figure 4. Some NNs use the output layer as a classifier, where the final numerical values predict what category (of predefined options) the input data belongs to. However, this study uses the output layer of the NNs as regressors, where the output nodes of the network are numerical maneuver parameters. Node connections utilize an "activa-

tion function", also referred to as a transfer function, to produce an output value. This study used the highly popular rectified linear unit (RELU) activation function for all neurons. While a tanh function would introduce nonlinearity to the NN, RELU doesn't struggle with the vanishing gradient problem [19].

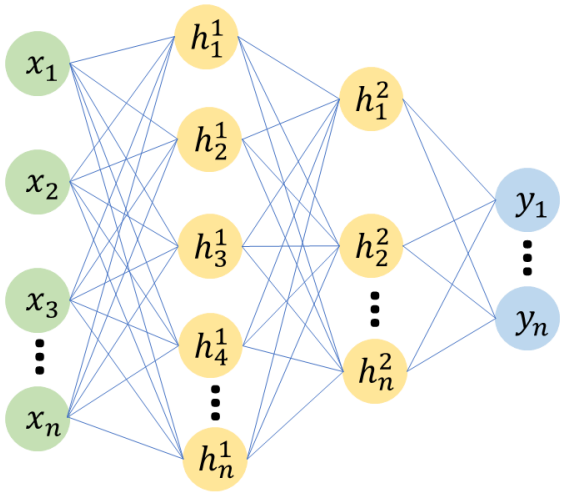

Fig. 4. Fully Connected Neural Network

In an untrained state, neural networks are nothing more than a series of random operators connected to one another. To make a NN useful, the weights and biases of the model need to be trained. There are many different methodologies for training a neural network. This study uses a supervised learning environment, where a large dataset of desired input/output pairs are used to model the desired performance. In a supervised learning environment, training data is run through the NN input layer, then the error between the NN output layer and the desired training data output gets computed. This error value is then "back propagated" through the model, modifying the model parameters to better approximate the gradient of the data. The commonly used Adam optimizer was used to facilitate network training. All NN implementations used in this study leveraged the PyTorch machine learning library.

Once a neural network is instantiated, there are a variety of parameters used to classify and fine-tune the training process of the model. One such training parameter is the *Loss* function (J). The *Loss* function is used to assess the error between the output layer of a NN (y_{pred}) and the training data ($y_{training}$). The resulting error value from the *Loss* function is back propagated through the model during the supervised learning process. This function is show in its generalized form in Eqn. [4].

$$Loss = J(y_{pred}, y_{training}) \quad [4]$$

The best *Loss* function to use is highly specific to the type of data a NN is trying to learn. For regression problems, the most common *Loss* function is the Mean Squared Error (MSE), also called the L2, function. This function evaluates the error to be the squared difference between prediction and training values, as shown in Eqn. [5]. The MSE is the default function in the PyTorch machine learning library. However, by definition, the L2 function punishes the model for making large mistakes in prediction, but is less severe for small errors in prediction. This acts contrary to the goal of deploying NNs for maneuver autonomy in highly sensitive regions. In such a case, small errors will quickly grow into larger ones downstream. To counteract this issue, this study uses the Mean Absolute Error (MAE), also referred to as the L1 *Loss* function. As shown in Eqn. [6], the MAE function penalizes errors proportionally. This serves to make a regression model more robust to outliers. The L1 *Loss* function was used for all NNs trained in this study.

$$J_{MSE}(y_{pred}, y_{training}) = (y_{pred} - y_{training})^2 \quad [5]$$

$$J_{MAE}(y_{pred}, y_{training}) = |y_{pred} - y_{training}| \quad [6]$$

Finally, an important factor when attempting to use NNs as regression models is scaling of the inputs and outputs. Like most computational processes, NNs work best when data is scaled to be within the range of zero to unity. As such, the dimensional data fed to the input layer, along with the regression results coming from the output layer, need to be scaled. Two common methods of scaling input and output data to NNs are "Standard Scalars" and "MinMax Scalars." As the names imply, the Standard Scalar function records the statistical properties of the training data. Next, it removes the mean from all data and scales the features by unit variance. Alternatively, the MinMax Scalar function simply performs scaling by the range of the data extrema. MinMax Scalars were used for this study to make the NN as robust as possible to outlier points. The Scikit-Learn Python package implementation of this scalar function was used for all models in this study.

3. Theory and calculation

With a topic as broad as autonomy, it is important to discuss assumptions and methodology utilized for any analysis conducted. This section addresses the process used to determine neural network parameters in this study, along with the mission ConOps used.

3.1 Neural Network Parameters

When constructing and training a neural network, there are multiple important parameters to consider. One of the most significant parameters is the size of the NN. Because this study didn't have prior knowledge of the most suitable number of nodes and layers, a network sizing study was conducted. Maneuver parameters of the impulsive apogee raise burn (ARB) were used for the training data set. The ARB impulsive maneuver is discussed in more detail in section 3.2.

To assess the most suitable network structure, a trade study was conducted by training multiple models of differing sizes with the same dataset. NN structures were created for multiple permutations of the number of hidden layers and the number of nodes per layer. Parameters used for these permutations are shown in Table 2. Note that a hidden layer with size zero nodes indicates that the layer is not used. The code used to create network permutations is compatible with up to four hidden layers, but a maximum depth of three hidden layers was used for this study. All networks were trained with a learning rate (lr) of 0.001, and training was conducted in batches for 2,000 epochs.

Table 2. Permutable Layer Widths for Network Sizing

Network Layer	Number of Permutable Nodes
Input	6
h_1	100, 50
h_2	50, 30
h_3	30, 0
h_4	0
Output	3

The results of the impulsive maneuver network sizing study can be seen below in Fig. 5. The plot shows the *Loss* function evaluation of a validation subset of the training data at the end of each training epoch. Notice that most models converge to a final loss value close to 0.005. It is important to note that due to the use of input and output data scaling functions, the *Loss* function value doesn't have any dimensional interpretation. While all network sizes tended towards the same final loss, the 6_100_50_30_0_3 sized network has the lowest stochastic noise. Hence, it was selected as the network size for all impulsive maneuvers in this study. Note that all models demonstrate oscillation around the converged value. To address this noise, an early stopping method was applied to training. The early stopping algorithm records the lowest *Loss* function evaluation after each training epoch, saving the best weights and biases. This process helps prevent overfitting the network to the training data, which would sacrifice generality.

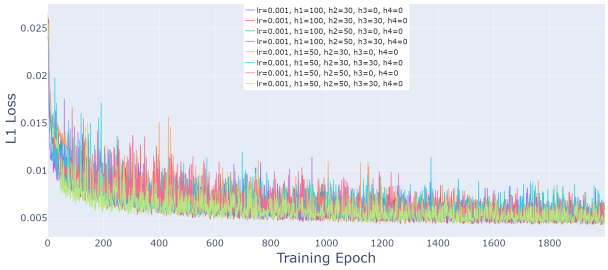

Fig. 5. Impulsive Maneuver Network Sizing Study

To ensure that the NNs used in the finite burn transfer case still have the capacity to address increased complexity, a second network sizing study was conducted. Similar to the first study, the ARB maneuver parameters were used as training data. However, this time the *finite burn* ARB maneuver parameters were utilized. Finite burn methodology is addressed in more detail in section 3.3. The same layer sizes outlined in Table 2 were used, and the resulting *Loss* function evaluations are shown in Fig. 6. Again, the 6_100_50_30_0_3 NN structure proved to be the most appropriate size. Hence, the same network size was used for the finite burn transfer scenario. Notice that the finite burn networks converge to a larger L1 loss compared to the impulsive networks. While this may at first seem alarming, recall that inputs and outputs are scaled relative to the training data properties. Additionally, the maneuver parameters are different between the two studies. As such, a final converged *Loss* function is not expected to be identical between the two cases.

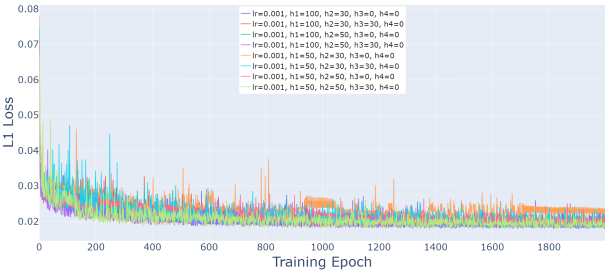

Fig. 6. Finite Burn Maneuver Network Sizing Study

3.2 Impulsive Transfer

The first and (relatively) simpler scenario used to study spacecraft maneuver autonomy is the impulsive transfer case. This transfer utilizes the same design reference mission (DRM) and sequence of maneuvers discussed in section 2.3 to target the stable manifolds of an Earth-Moon

L_3 Lyapunov orbit. All burns are treated as impulsive, instantaneous ΔVs in this case. To generate training data, statistical perturbations were applied to the initial conditions (listed in Table 1) and the transfer was optimized, subject to the perturbed boundary value. Perturbations were added randomly to each training data case, with a one standard deviation (std) position displacement of 50 m and a velocity perturbation with one std of 5 m/s. Optimization was performed within the ASSET toolkit, with the combined ΔV of the four burns serving as the objective function. Rendezvous with the L_3 Lyapunov manifold structure was enforced via a Signed Distance Field (SDF) method [41]. An SDF encapsulates the entire stable manifold structure, including perturbations due to ephemeris dynamics, as an equality constraint on the final coast phase. The SDF target is able to represent a manifold insertion even when the optimal rendezvous point changes. This helps demonstrate the ability of NN prescribed maneuvers to adjust to perturbations and maneuver execution errors. The L_3 Lyapunov stable invariant manifold trajectory used to generate the target SDF is shown in Figure 7.

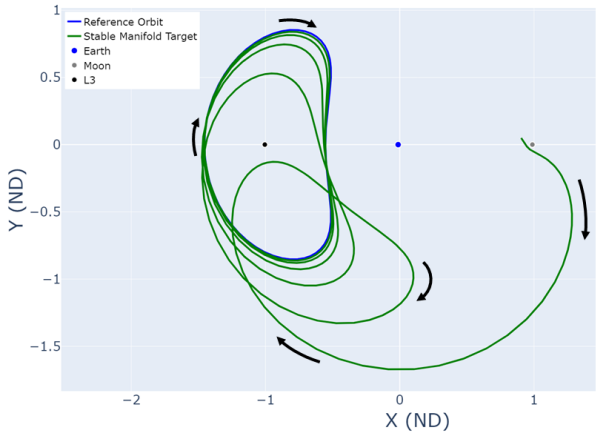

Fig. 7. L_3 Lyapunov Stable Invariant Manifold

Applying neural networks to autonomous burn prescription relies on maneuver specific NNs. Each maneuver in the ConOps has its own set of networks used to predict the properties of the upcoming burn by sampling the coast trajectory states leading up to execution. For the impulsive transfer scenario, each maneuver has two distinct NNs: one that predicts the three-dimensional impulsive ΔV direction and magnitude, and another that prescribes how much longer the vehicle needs to coast before executing the maneuver. It may seem computationally cumbersome for a flight computer to store eight NNs for a ConOps with only four burns. However, it is important to remember that individual networks are relatively lightweight, both to store and to sample.

This study performs the process of generating data and training NNs sequentially. For example, once the ARB networks are fully trained, they are used in training data generation for the TLI burn. This process of training downstream NNs with outputs of previous maneuver NNs helps prevent cascading error. In essence, each maneuver network is trained on the expected range of states resulting from previous autonomous maneuvers. A summary of the NN training data generation workflow is shown in the flowchart in Figure 8. Once this process is completed for one maneuver, the data is used to train the ΔV and timing NNs for that maneuver. Then the data generation procedure is repeated for the next maneuver, until NNs are created for all burns.

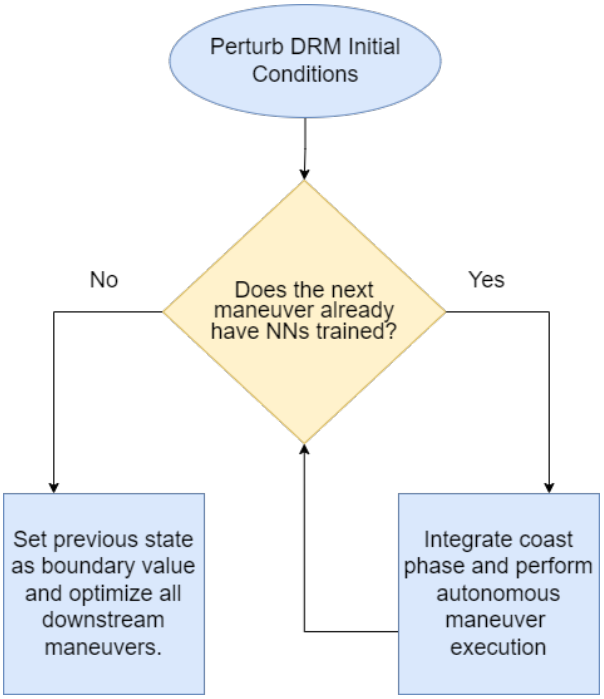

Fig. 8. Training Data Generation Flowchart

While each maneuver in the impulsive transfer scenario has two NNs with different outputs (ΔV and time until burn), the input layer is the same for both networks. Each network expects six states as inputs, corresponding to the spacecraft position and velocity during the coast arc leading up to a maneuver. This enables the NNs to be sampled multiple times while the spacecraft is coasting. As such, the process of applying the NN for autonomous maneuver execution consists of integrating the spacecraft for some fixed duration (30 minutes in this study), sampling the networks to get maneuver parameters and time

until execution, then repeating the process. This iterative loop of integrating ballistically and sampling the NNs continues until some cutoff time. Once the maneuver cutoff time is reached (5 hours before predicted burn time in this study), the simulation coasts to the estimated burn time and executes the autonomously prescribed impulsive maneuver. This study treats the cutoff time of five hours before burn execution as a placeholder, where a technology demonstration mission would be able to downlink the autonomously generated maneuver parameters. The burn could then be inspected and validated before the vehicle actually executes the burn.

The highly sensitive DRM used for this study intentionally introduced an additional layer of complexity to this process of autonomous maneuver design. Because the mission ConOps was selected as a stressing case, small changes in maneuver parameters (both the ΔV and the timing) results in drastically different converged trajectories. To mitigate the impact of such sensitivities, a rolling average filter was applied to NN outputs. This filter, which computes the rolling average output for each maneuver parameter from the previous 25 samples, proved to be highly effective in producing consistent maneuvers. The impact of navigation error applied to the rolling average filter is left as future work.

3.3 Finite Burn Transfer

With the promising results of impulsive ΔV maneuver design autonomy, the natural extension of such a study is to apply similar methods to a finite burn transfer. This higher fidelity trajectory uses a fixed thrust, fixed I_{sp} engine model to accelerate the vehicle during dedicated thrust arcs. Vehicle properties used in this study are designed to emulate a conventional vehicle designed for cislunar exploration. Spacecraft of this class will be required to support the M2M architecture. Engine properties used in this study are shown in Table 3. The DRM for the finite burn trajectory is similar to the impulsive scenario, with the same initial conditions and statistical perturbation values. Again, the process of creating the maneuver autonomy NNs follows the training data generation process outlined in Figure 8. The maneuvers are still trained sequentially in order to avoid cascading error that could result from training later maneuvers in the DRM without incorporating autonomous maneuver design for the previous burns.

Table 3. Finite Burn Engine Properties

Parameter	Value
Thrust	22 N
I_{sp}	225 s
Initial Mass	225.5 kg

When generating training data, the finite burn model is optimized with respect to the total mass utilization by the engine. Again, an L_3 Lyapunov orbit SDF is applied as a boundary condition for orbit insertion. To produce optimized trajectories, ASSET is configured to minimize the objective, with engine pointing direction, coast timing, and burn duration as control variables. To ensure the DRM was a stressing case, engine pointing rates were unconstrained.

A challenge with the finite burn scenario that isn't present with the impulsive case is the necessity of predicting an entire engine pointing time history. The highly sensitive ConOps chosen as a stressing case for this study requires the spacecraft to slew the engine pointing during maneuvers. While inertially fixed pointing is more common for flight operations, slewing a vehicle engine is sometimes required to reduce burn losses. To address this challenge, a second-order polynomial control law was utilized as the maneuver parameters.

Using the coefficients of a polynomial control law for the NN output addresses the challenge of engine slewing during maneuver execution. However, a single polynomial can only represent one-dimensional output. As such, a NN that samples the spacecraft states as an input and returns polynomial coefficients describes the engine pointing for a single Cartesian direction. Hence, to fully prescribe the time history of a slewing finite burn maneuver, three NNs are utilized to command engine pointing. One NN is required for each direction. This study used the Earth-centered VNC frame basis vectors as the NN predicted engine pointing directions. The VNC frame is commonly used by guidance algorithms, and assumes the instantaneous velocity ($\vec{v}$) as a basis vector. Next, the two-body angular momentum is treated as a second basis vector, and a cross product is used to complete the right hand set. Equations 7, 8, and 9 show the derivation of the VNC frame, given the inertial state information.

$$\vec{v} = \vec{v}_{inertial} \quad [7]$$

$$\vec{n} = \vec{r}_{inertial} \times \vec{v}_{inertial} \quad [8]$$

$$\vec{c} = \vec{v} \times \vec{n} \quad [9]$$

One final complication introduced by the finite burn model is the need to know how long a burn should last. To address this, another single output NN is used, similar to the ones used to predict time until execution. The end result is that each maneuver in the finite burn trajectory requires five NNs: three VNC engine pointing commands (given as second-order polynomials), one to predict time until burn start, and one to predict the required burn duration. While five NNs per maneuver may at first seem excessive, recall that each network only requires around 60 mb of RAM. Reducing the number of required NNs, along with determining the feasibility of utilizing the same network for multiple maneuvers, is left as future work.

4. Results

This section will address the results of both the impulsive and the finite burn transfer scenarios utilizing autonomous maneuver design. Recall that both transfers were designed with a fixed DRM, then statistical perturbations were applied to the initial conditions. The standard deviation of position error was 50 m, and for velocity it was 5 m/s. These statistical variations, alongside the highly sensitive gravitational regions of the DRM, serve as a stress test of the autonomous maneuvers' ability to adjust to perturbations.

4.1 Impulsive Transfer Results

The baseline impulsive transfer follows the same ConOps shown in Figure 3. The nominal impulsive ΔV magnitudes are shown below in Table 4. Transfers were deployed in perturbed initial conditions, then NNs were sampled every 30 minutes and filtered maneuver parameters were updated. This process repeated until the burn was predicted to occur within five hours. At that cuttoff time, the spacecraft coasted until the NN estimated burn time. The autonomously prescribed maneuver was then executed. After one autonomous maneuver was executed, the neural networks for the next maneuver were loaded and the process of sampling coast states to prescribe the next burn repeated. This procedure continued until all maneuvers were completed. Figure 9 shows five randomly generated transfers utilizing the autonomous impulsive maneuver design methodology. All coast arcs are shown in green, then the post manifold insertion burn trajectories are colored uniquely.

Table 4. Baseline Impulsive Transfer ΔVs

Maneuver	ΔV (m/s)
ARB	120
TLI	129.54
OPF	128.6
MI	1e-6
Total	378.14

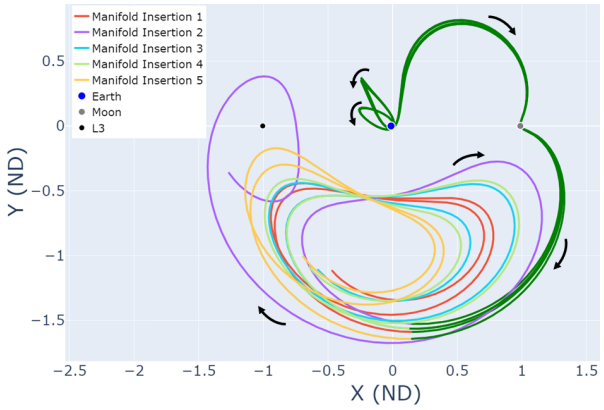

Fig. 9. Autonomous Impulsive Maneuver Trajectories

An SDF of the L_3 Lyapunov stable manifold structure was used as the orbit insertion criteria. As such, the final trajectories in Figure 9 follow the manifold motion, leading toward the reference orbit after the MI maneuver is executed. To assess how well the NNs performed autonomous maneuver design, 1,000 perturbed trajectories were generated. If the post-MI trajectory was inserted within the L_3 Lyapunov orbit stable manifold SDF, the autonomous maneuvers were considered successful. Of the 1,000 randomized trajectories run, 997 of them (99.7%) resulted in final states within the target SDF.

4.2 Finite Burn Transfer Results

As discussed in Subsection 3.3, the methodology of the finite burn autonomous maneuver design is very similar to the impulsive transfer techniques. The baseline mission used the engine parameters listed previously in Table 3. Table 5 below records the nominal finite burn duration and approximated ΔVs for the DRM. The same standard deviations were used for statistical perturbations applied to the initial conditions (50 m position, 5 m/s velocity). Again, coast trajectories were integrated in sections of 30 minutes, then the maneuver autonomy NNs were sampled. A rolling average filter was applied to all maneuver parameters predicted by the networks. Once the estimated time until burn was below five hours, the autonomously

prescribed burn was executed. Figure 10 shows a random selection of five autonomous finite burn transfers. Again, the transfer trajectories are shown in green, and the post manifold insertion coast arcs are colored uniquely. Notice that the autonomous maneuvers responded flexibly to deliver the spacecraft into the stable manifold SDF at different locations.

Table 5. Baseline Finite Burn Transfer ΔVs

Maneuver	Duration (min)	ΔV (m/s)
ARB	31.31	191.3
TLI	12.78	83.09
OPF	21.38	146.44
MI	0.25	1.75
Total	65.72	422.58

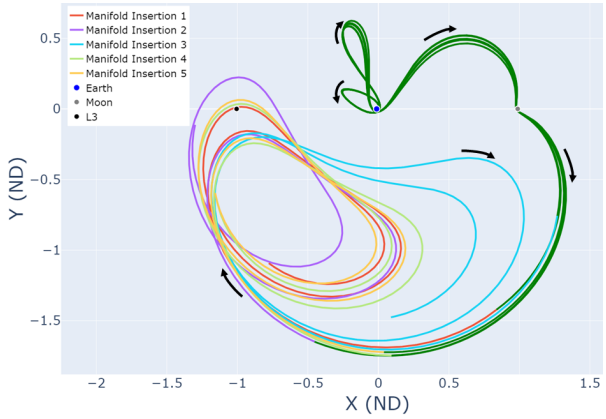

Fig. 10. Autonomous Finite Burn Trajectories

Similar to the impulsive scenario, 1,000 randomized finite burn trajectories were generated with the L_3 stable manifold SDF as the convergence criteria. Of all the finite burn trajectories with autonomous maneuver design, 975 resulted in final coast phases within the SDF target (97.5%).

5. Discussion

The results of this study demonstrate the strength of NNs trained in a supervised learning environment for autonomous, on-board maneuver design. A ConOps with highly sensitive maneuvers, along with statistically perturbed initial conditions, proved to be a good testbed for examining how well NNs respond to deviations from the original DRM. As shown by the high percentages of converged transfers (both in the impulsive and finite burn cases), the proposed method of maneuver autonomy is promising. The impulsive ΔV analysis was useful for developing a procedure that directly benefited the more complex finite burn methodology.

Constructing training data that incorporates the autonomous maneuver output from all previous maneuvers in the ConOps was highly advantageous. This method (following the logic shown in Figure 8) mitigates the issue of cascading error. Additionally, using a supervised learning approach is an effective way to determine a DRM before launch. Then NNs can be used to prescribe near-optimal maneuver parameters during flight, while maintaining the original mission ConOps. Such an approach leverages the strength of traditional mission design and optimization methods. Simultaneously, it enables robust and autonomous execution of deterministic burns during operations. Spacecraft equipped with this capability could, in theory, conduct nominal transfer operations without relying on ground-based maneuver commands.

Utilizing this methodology is especially useful for scenarios where optimal maneuver targets can change due to perturbations or execution error. The invariant manifold insertion target used in this study is one such case, where slightly different lunar flybys can result in drastically different optimal insertion states. These dynamically sensitive, multi-body orbits don't always have fixed inertial targets, as would be needed for on-board iterative methods. In contrast, on-board NNs that are already trained on optimal maneuver execution are not dependent on fixed target states. As such, NNs can autonomously adapt to perturbations and robustly insert into three-body orbit structures, despite these structures not having fixed target parameters (such as Keplerian elements). This flexibility is demonstrated in Figures 9 and 10. These trajectory plots exhibit how the autonomous NN maneuvers adapt to perturbations and insert into the stable manifold structure at different points.

Obviously, there is still a significant amount of work to be done before fully autonomous spacecraft can become a reality. However, this study positions itself as an incremental step towards developing the methodology required for such a goal. The procedure of sampling spacecraft states every 30 minutes until a cutoff time of five hours before burn serves as a possible strategy for a future flight test demonstration. While the overarching goal is to have fully autonomous spacecraft, it would be infeasible and irresponsible to have a spacecraft execute an unverified maneuver. That is, at least until the technology has been fully vetted and verified. However, for a flight demonstration mission, sampling NNs until five hours before maneuver execution would give sufficient time to downlink and verify the parameters. Should the maneuver be infeasible, a ground operations team would have sufficient time to up-

load a proper command using conventional design methods.

Requiring ground-based communications with a spacecraft before each maneuver is antithetical to the long term goal of fully autonomous spacecraft. However, the proposed flight test demonstration still has benefits to spacecraft operations in the short term. For example, a relatively small team could utilize the proposed methodology to operate a large constellation of spacecraft that require maneuvers to transfer to a target orbit. Instead of needing to reoptimize and upload every maneuver for each spacecraft, the autonomously prescribed maneuver could be downlinked at the sampling cutoff time. Then, after verifying the maneuver generated by the NNs, a simple go-ahead command could be uplinked. This would allow the vehicle to perform the autonomously prescribed burn. However, if burn parameters do not pass verification, there would still be sufficient time to generate and uplink a proper maneuver using conventional methodology.

This study demonstrates promising results for autonomous maneuver design. Nevertheless, there is still much work to be done to achieve fully autonomous vehicles. One important issue not addressed in this study is the impact of navigation error. The NNs used here have perfect spacecraft state knowledge at all times. While such a configuration serves as a sufficient proof-of-concept, navigational uncertainty is guaranteed to occur during flight operations. Quantifying the impact of navigation error on the proposed method, along with ways to mitigate the consequences, have been left as future work.

6. Conclusions

NASA's Moon to Mars campaign is an ambitious roadmap for groundbreaking science and exploration. While the overall goals of the campaign are exciting, the the technical challenges should not be understated. The cislunar architecture needed to support such operations will require a significant amount of new spacecraft to be deployed. Ground-based operations facilities are already straining to meet the demand of vehicles currently on orbit. To mitigate the impact of launching more vehicles, spacecraft autonomy will be required. This study serves as a proof-of-concept for using neural networks trained in a supervised learning environment to perform autonomous, on-board maneuver design. The supervised learning approach allows mission planners to enforce a DRM derived in advance, benefiting from conventional trajectory design methods. Simultaneously, deploying NNs for maneuver design has proved to result in robust burn parameters, even when subjected to perturbations and highly sensitive orbital regimes. This method is especially beneficial for scenarios where optimal maneuver states can change drastically, such as invariant manifold targets. While there are still challenges to overcome, this is a promising step towards fully autonomous maneuver execution.

7. Acknowledgements

The authors would like to thank the NASA Human Lander System and the Solar Cruiser program offices for making this research possible. Additionally, appreciation is extended to the Guidance, Navigation, and Mission Design branch at NASA Marshall Space Flight Center for supporting this study.

References

[1] T. Atkins *et al.*, "Artemis 1 flight instrumentation data quality assessment and processing," in *AIAA SCITECH Forum*, Orlando, FL, Jan. 2024.

[2] M. Sarafin, L. Logan, A. Kshatriya, and D. Elburn, "Artemis 1: Test flight buys down risk for humanity's return to the moon," in *International Astronautical Congress*, Baku, Azerbaijan, Aug. 2023.

[3] A. Houin and R. Sood, "Optimal launch windows for artemis iii and beyond leveraging contourmaps," in *IAF 73rd International Astronautical Congress (IAC)*, Paris, France, Sep. 2022.

[4] G. Chavers, L. Watson-Morgan, M. Smith, N. Suzuki, and T. Polsgrove, "Nasa's human landing system: The strategy for the 2024 mission and future sustainability," in *IEEE Aerospace Conference*, Big Sky, MT, Mar. 2020.

[5] M. Duggan and T. Moseman, "Deep space gateway architecture to support multiple exploration demonstration goals," in *IEEE Aerospace Conference*, Big Sky, MT, Mar. 2018, pp. 1–8.

[6] A. Burg, K. G. Boggs, K. Goodliff, E. McVay, G. Benjamin, and D. Elburn, "Architecture robustness in nasa's moon to mars capability development," in *2021 IEEE Aerospace Conference*, 2021, pp. 1–12.

[7] E. Anzalone, J. Getchius, jared Leggett, B. Ashman, J. Parker, and L. Winternitz, "Lunar navigation beacon network using global navigation satellite system receivers," in *International Astronautical Congress*, Washington, DC, Oct. 2019.

[8] E. J. Anzalone *et al.*, "Ground-testing of one-way ranging from a lunar beacon demonstrator payload (lunar node -1)," in *International Technical Meeting of The Institute of Navigation*, Long Beach, California, Jan. 2022.

[9] D. Abraham *et al.*, "Nasa deep space communications: Future mission trends and their implications," in *International Conference on Space Operations*, Dubai, United Arab Emirates, Mar. 2023.

[10] S. Bhaskaran, "Autonomous navigation for deep space missions," Stockholm, Sweden, Jun. 2012. DOI: 10.2514/6.2012-1267135.

[11] D. Wang, l. Maodeng, H. Xiangyu, and X. Zhang, *Spacecraft Autonomous Navigation Technologies Based on Multi-source Information Fusion*. Jan. 2021, ISBN: 978-981-15-4878-9.

[12] A. Heintz and M. A. Peck, "Autonomous optical navigation for resident space object exploration," in *AIAA Scitech 2020 Forum*, Orlando, FL, Jan. 2020.

[13] C. D. Norman *et al.*, "Autonomous navigation performance using natural feature tracking during the osiris-rex touch-and-go sample collection event," *The Planetary Science Journal*, vol. 3, no. 5, p. 101, May 2022.

[14] L. Capra, A. Brandonisio, and M. Lavagna, "Network architecture and action space analysis for deep reinforcement learning towards spacecraft autonomous guidance," *Advances in Space Research*, vol. 71, no. 9, pp. 3787–3802, 2023, Application of Artificial Intelligence in Tracking Control and Synchronization of Spacecraft, ISSN: 0273-1177. DOI: https://doi.org/10.1016/j.asr.2022.11.048.

[15] D. Izzo, E. Blazquez, R. Ferede, S. Origer, C. D. Wagter, and G. C. H. E. de Croon, "Optimality principles in spacecraft neural guidance and control," *Science Robotics*, vol. 9, no. 91, eadi6421, 2024. DOI: 10.1126/scirobotics.adi6421.

[16] B. Park and K. Howell, "Characterizing transition-challenging regions leveraging the elliptic restricted three-body problem: L2 halo orbits," in *AIAA SCITECH 2024 Forum*, Orlando, FL, Jan. 2024.

[17] M. S. Suraj, R. Aggarwal, M. C. Asique, A. Mittal, M. Jain, and V. K. Paliwal, "Effect of three-body interaction on the topology of basins of convergence linked to the libration points in the r3bp," *Planetary and Space Science*, vol. 205, p. 105 281, 2021, ISSN: 0032-0633. DOI: https://doi.org/10.1016/j.pss.2021.105281.

[18] R. Michalski, J. Carbonell, and T. Mitchell, *Machine learning. an Artificial Intelligence approach. Volume 2*. Springer, Jan. 1983, vol. Vol. II. DOI: 10.1007/978-3-662-12405-5.

[19] J. Zupan, "Introduction to artificial neural network (ann) methods: What they are and how to use them," *Acta Chimica Slovenica*, vol. 41, no. 3, p. 327, 1994.

[20] S. De Smet, D. Scheeres, and J. Parker, "Representing dynamics in the eccentric hill system using a neural network architecture," *Astrodynamics*, vol. 3, Aug. 2019. DOI: 10.1007/s42064-019-0050-4.

[21] J. Yan, H. Yang, and S. Li, "Ann-based method for fast optimization of jovian-moon gravity-assisted trajectories in cr3bp," *Advances in Space Research*, vol. 69, no. 7, pp. 2865–2882, 2022, ISSN: 0273-1177. DOI: https://doi.org/10.1016/j.asr.2022.01.019.

[22] S. De Smet, D. Scheeres, and J. Parker, "Leveraging artificial neural networks to systematically explore solar gravity driven transfers in the martian system," *The Journal of the Astronautical Sciences*, vol. 66, pp. 1–40, Mar. 2019. DOI: 10.1007/s40295-018-00149-w.

[23] G. A. Tsirogiannis, "A graph based methodology for mission design," *Celestial Mechanics and Dynamical Astronomy*, vol. 114, pp. 353–363, 2012.

[24] N. L. Parrish, "A* pathfinding for continuous-thrust trajectory optimization," in *AAS 37th Annual Guidance Control Conference*, Breckenridge, CO, Jan. 2014.

[25] E. Vassev and M. Hinchey, *Autonomy requirements engineering for space missions*. Springer, 2014.

[26] E. Vassev and M. Hinchey, "Autonomy requirements engineering: A case study on the bepicolombo mission," Jul. 2013, pp. 31–41. DOI: 10.1145/2494444.2494472.

[27] E. Trumbauer and B. Villac, "Heuristic search-based framework for onboard trajectory redesign," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 1, pp. 164–175, 2014.

[28] A. Das-Stuart, K. Howell, and D. Folta, "Rapid trajectory design in complex environments enabled by reinforcement learning and graph search strategies," *Acta Astronautica*, vol. 171, pp. 172–195, 2020, ISSN: 0094-5765. DOI: https://doi.org/10.1016/j.actaastro.2019.04.037.

[29] N. B. LaFarge, D. Miller, K. C. Howell, and R. Linares, "Guidance for closed-loop transfers using reinforcement learning with application to libration point orbits," in *AIAA Scitech 2020 Forum*, Orlando, FL, Jan. 2020. DOI: 10.2514/6.2020-0458.

[30] A. Das-Stuart and K. Howell, "Contingency planning in complex dynamical environments via heuristically accelerated reinforcement learning," in *AAS/AIAA Astrodynamics Specialist Conference*, Portland, ME, Aug. 2019, pp. 1–21.

[31] A. Rubinsztejn, R. Sood, and F. E. Laipert, "Neural network optimal control in astrodynamics: Application to the missed thrust problem," *Acta Astronautica*, vol. 176, pp. 192–203, 2020, ISSN: 0094-5765. DOI: https://doi.org/10.1016/j.actaastro.2020.05.027. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0094576520303106.

[32] B. Nejad, "The flight dynamics facility," in *Introduction to Satellite Ground Segment Systems Engineering: Principles and Operational Aspects*. Cham: Springer International Publishing, 2023, pp. 93–117. DOI: 10.1007/978-3-031-15900-8_6.

[33] J. Pezent, J. Sikes, W. Ledbetter, R. Sood, K. Howell, and J. Stuart, "Asset: Astrodynamics software and science enabling toolkit," in *AIAA SciTech Forum*, San Diego, CA, Jan. 2022. DOI: 10.2514/6.2022-1131.

[34] J. Pezent, J. Sikes, R. Sood, K. Howell, and J. Stuart, "Enhancements to the astrodynamics software and science enabling toolkit (asset)," in *AAS/AIAA Astrodynamics Specialist Conference*, Big Sky, MT, Aug. 2023.

[35] A. Houin, R. Sood, and D. Tyler, "Roll torque mitigating trajectories for multi-body solar sail orbits," in *AAS Guidance, Navigation and Control Conference*, Breckenridge, CO, Feb. 2024.

[36] C. Sandel and R. Sood, "Natural and forced spacecraft loitering in a near rectilinear halo orbit," *The Journal of the Astronautical Sciences*, vol. 71, May 2024. DOI: 10.1007/s40295-024-00446-7.

[37] B. Blumenthal and R. Sood, "Local lyapunov exponent augmented differential corrections process for cislunar trajectory targeting," *The Journal of the Astronautical Sciences*, vol. 70, p. 30, Aug. 2023. DOI: 10.1007/s40295-023-00396-6.

[38] M. Rocklin, "Dask: Parallel computation with blocked algorithms and task scheduling.," in *Python In Science Conference*, Austin, TX, Jul. 2015, pp. 126–132.

[39] D. J. Grebow, M. T. Ozimek, K. C. Howell, and D. C. Folta, "Multibody orbit architectures for lunar south pole coverage," *Journal of Spacecraft and Rockets*, vol. 45, no. 2, pp. 344–358, 2008.

[40] K. Hornik, M. B. Stinchcombe, and H. L. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359–366, 1989.

[41] J. Sikes, J. Pezent, and R. Sood, "Applications of signed distance fields to spacecraft trajectory design," in *AAS/AIAA Astrodynamics Specialist Conference*, Big Sky, MT, Aug. 2023.