(Preprint) AAS 23-185

COSTATES FEEDBACK CONTROL FOR MASS-OPTIMAL LOW-THRUST TRANSFERS

Yuri Shimane[*], Dario Izzo[†], and Koki Ho[‡]

Designing efficient low-thrust trajectories involves solving optimal control problems, which can be computationally intensive. One promising approach to tackle this challenge is to use a neural network (NN) as a scheme for obtaining the control input that guides the spacecraft to its targeted orbit. This work explores the use of a NN for learning the costates from the current and targeted states, which can be used together with Pontryagin's maximum principle and optimal control theory to derive the control to provide a feedback loop for controlling the spacecraft. In effect, this is a policy approximation scheme, even though it does not explicitly have the controls as its output. The proposed method is applied to a set of orbits departing from near-ecliptic near-Earth object targeting the Earth's orbit for a mass optimal orbit transfer.

INTRODUCTION

Rapid construction of optimal low-thrust transfers is a central problem that arises in many mission design applications. For example, feasibility studies for near-Earth asteroids (NEA) typically involve constructing trajectories to a large set of objects in the Solar system. Traditional approaches for solving the optimization problem involved in low-thrust trajectory design are typically regarded in terms of indirect or direct methods. Indirect methods involve leveraging optimal control theory to construct a two-point boundary value problem (TPBVP) whose solution is the (local) optimal trajectory. In contrast, direct methods convert the design task into a nonlinear programming problem (NLP) with the states and the controls as variables at discrete times along the transfer. Both methods have their unique advantages and disadvantages, but one common issue with these two approaches is the fact that they involve numerically solving either a BVP or an NLP, with convergence being dependent on having a good initial guess.

This can become problematic for example when conducting space architecture trade studies, where large numbers of optimal or near-optimal trajectories must be evaluated. One approach to tackle this challenge is to use feedback controllers and simply integrate the trajectory forward in time with the controller in the loop. While simpler controllers working on linearized dynamics struggle to perform well on the nonlinear two-body problem, Lyapunov controllers, such as the Q-law studied by Petropoulos[1–3] and later used by multiple works,[4–7] provide an efficient framework for designing sub-optimal transfers. The sub-optimality inherently comes from the feedback architecture and the use of pre-determined heuristic rules for thrusting or coasting at a given instance

[*]PhD Candidate, School of Aerospace Engineering, Georgia Institute of Technology, GA 30332, USA
[†]Advanced Concepts Team (ACT), European Space Research & Technology Centre (ESTEC), Keplerlaan 1, 51014 AG Noordwijk, Netherland
[‡]Associate Professor, School of Aerospace Engineering, Georgia Institute of Technology, GA 30332, USA

along the transfer. As a remedy to this drawback, Holt et al[8] considered the use of reinforcement learning (RL) to tune these heuristic rules according to the current state of the spacecraft.

The use of neural networks (NN) in the context of solving orbit transfer optimal control problems has seen increasing attention over the past few years.[9–15] The NNs studied in these works, commonly termed as Guidance and Control Network (G&CNET) can be categorized for learning the value or the objective value,[12] the initial costates,[11,14] or the control, also known as policy approximation.[10,12–15] The data sets in these works vary, but typically involve interplanetary orbits in the vicinity of Earth's orbit. For example, Li et al[11] considered a data set of Earth-Mars transfers and Earth-asteroid transfers, while Izzo and Öztürk[12] used a data set of Earth-Venus orbit transfers. While traditional trajectory design typically has to trade between open-loop optimization or sub-optimal solution obtained from feedback form, an NN-based scheme is able to yield near-optimal trajectories in a feedback form, hence providing a completely different paradigm for trajectory design.

Successful training of a NN requires a sufficiently large data set, which would be expensive to generate if the optimal control problem was to be solved multiple times as TPBVPs. Meanwhile, if the data set can be designed to contain trajectories within a boxed region in state space, the so-called *backward generation of optimal examples*[12] may be leveraged. This technique involves backward propagation of states and costates that meet the transversality condition, thus generating an arbitrary optimal trajectory via a simple numerical integration.

In this work, we propose a NN for learning the costates of mass-optimal low-thrust transfers and use this as a feedback scheme for guiding the spacecraft. This is in effect an indirect policy approximation, where the NN is combined with optimal control theory to obtain the policy. While leveraging NN to learn the initial costates for solving time-optimal OCP with an indirect method has previously been studied by Li et al,[11] the primary use for the costates NN in this work is its use as a feedback controller. This way, we can robustly obtain near-optimal trajectories, even with the mass-optimal problem, which is known to have a much smaller convergence radius than the time-optimal problem solved. This is enabled by the nature of our data set involving training samples collected along many locations along multiple transfers, compared to Li et al's dataset involving only the initial costates of optimal transfers between single initial and final orbits, respectively. Furthermore, if the theoretical, local optimal trajectory is sought, the costates NN can still be used effectively to generate an initial guess. We also explore this avenue of application by formulating a multiple-shooting approach to solve the indirect problem.

This paper is organized as follows: first, the orbit transfer problem, along with the relevant optimal control theory, is introduced. Then, the design of the neural network is discussed. This is followed by a demonstration of the proposed approach using a data set based on NEOs. Finally, the last Section provides conclusive remarks.

ORBIT TRANSFER PROBLEM AND PONTRYAGIN'S MAXIMUM PRINCIPLE

The orbit transfer problem consists of guiding the spacecraft from its initial state to a targeted orbit; it differs from rendez-vous problem as we do not require the spacecraft to match the phase with a particular target. In this work, the transfer problem is studied in the modified equinoctial elements (MEE). As such, the target orbit is given by 5 of the slow MEE elements, $\boldsymbol{x}_{oe}^* = [p^*, f^*, g^*, h^*, k^*]$. In this section, we start by introducing the two-body dynamics of the spacecraft in MEE. Then, Pontryagin's Maximum Principle (PMP) is discussed.

Spacecraft Dynamics

It is convenient, in the process of deriving the optimal control problem, to express the dynamics in terms of the perturbed and secular components; let $\boldsymbol{x}_{oe} = [p, f, g, h, k, L]$ and define $\boldsymbol{B}(\boldsymbol{x}_{oe})$ and $\boldsymbol{D}(\boldsymbol{x}_{oe})$ such that

$$\sqrt{\frac{\mu}{p}}\boldsymbol{B}(\boldsymbol{x}_{oe}) = \begin{bmatrix} 0 & \frac{2p}{w} & 0 \\ \sin L & [(1+w)\cos L + f]\frac{1}{w} & -\frac{g}{w}(h\sin L - k\cos L) \\ -\cos L & [(1+w)\sin L + g]\frac{1}{w} & \frac{f}{w}(h\sin L - k\cos L) \\ 0 & 0 & \frac{1}{w}\frac{s^2}{2}\cos L \\ 0 & 0 & \frac{1}{w}\frac{s^2}{2}\sin L \\ 0 & 0 & \frac{1}{w}(h\sin L - k\cos L) \end{bmatrix} \tag{1}$$

$$\boldsymbol{D}(\boldsymbol{x}_{oe}) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \sqrt{\frac{\mu}{p^3}}w^2 \end{bmatrix}^T \tag{2}$$

Then, the dynamics take the compact form

$$\boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}) = \begin{bmatrix} \dot{\boldsymbol{x}}_{oe} \\ \dot{m} \end{bmatrix} = \begin{bmatrix} \frac{c_1}{m}\boldsymbol{B}(\boldsymbol{x}_{oe})\boldsymbol{u}(t) + \boldsymbol{D}(\boldsymbol{x}_{oe}) \\ -c_2 u(t) \end{bmatrix} \tag{3}$$

where c_1 is the maximum thrust, and c_2 are the maximum thrust and the mass-flow rate given by

$$c_2 = \frac{c_1}{I_{\text{sp}} g_0} \tag{4}$$

Pontryagin's Maximum Principle

Consider the minimum-control problem with the cost index $\mathcal{J}$ given by

$$\mathcal{J} = \int_0^{t_f} \mathcal{L}(\boldsymbol{x}(t), \boldsymbol{u}(t), t)\mathrm{d}t = \int_0^{t_f} [u - \varepsilon \log[u(1-u)]]\,\mathrm{d}t \tag{5}$$

Here, ε is a homotopy parameter that can be gradually decreased from 1 to approach 0, and transition from an energy-optimal problem to a mass-optimal problem. Figure 1 shows the optimal control magnitude profiles for a sample orbit transfer problem with decreasing ε.

From optimal control theory, by adjoining the dynamics to the cost, the Hamiltonian $\mathcal{H}$ is given by

$$\begin{aligned} \mathcal{H}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{u}) &= \mathcal{L} + \boldsymbol{\lambda}^T \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}) \\ &= [u - \varepsilon\log[u(1-u)]] + \frac{c_1}{m}\boldsymbol{\lambda}^T\boldsymbol{B}(\boldsymbol{x}_{oe})\boldsymbol{u} + \lambda_L\sqrt{\frac{\mu}{p^3}}w^2 - \lambda_m c_2 u \end{aligned} \tag{6}$$

where $\mathcal{L}$ is the Mayer cost, $\boldsymbol{\lambda} \in \mathbb{R}^n$ is the costates vector of the orbital elements, and λ_m is the costate of the mass.

The costates dynamics are given by

$$\dot{\boldsymbol{\lambda}} = -\frac{\partial \mathcal{H}}{\partial \boldsymbol{x}} \tag{7}$$

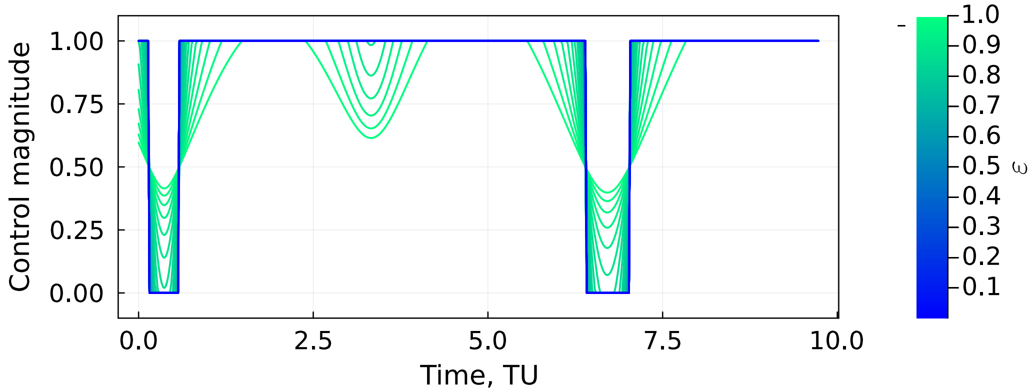

Figure 1. Example of optimal control magnitude profile smoothing via homotopy parameter ε

The full expression for this derivative is provided in the Appendix of Izzo and Öztürk.[12] From Pontryagin's Maximum Principle (PMP), the optimal control $\boldsymbol{u}^*$ is given by

$$\boldsymbol{u}^* = \underset{\boldsymbol{u}}{\operatorname{argmin}}\, \mathcal{H} \tag{8}$$

which is given by

$$\boldsymbol{u}^* = -u^*(t) \frac{\boldsymbol{B}(\boldsymbol{x}_{oe})^T \boldsymbol{\lambda}}{\|\boldsymbol{B}(\boldsymbol{x}_{oe})^T \boldsymbol{\lambda}\|_2} \tag{9}$$

where the thrust magnitude u^* is given by

$$u^*(t) = \frac{2\varepsilon}{2\varepsilon + SF(t) + \sqrt{4\varepsilon^2 + SF(t)^2}} \tag{10}$$

with the switching function $SF(t)$ is given by

$$SF(t) = 1 - \frac{c_1}{m} |\boldsymbol{B}(\boldsymbol{x}_{oe})^T \boldsymbol{\lambda}| - c_2 \lambda_m \tag{11}$$

Transversality Conditions

For an orbit transfer problem, the five slow elements are targeted, while the final true longitude, final mass, and final time t_f are free. As such, the transversality conditions are

$$\lambda_L(t_f) = 0, \;\; \lambda_m(t_f) = 0, \;\; \mathcal{H}(t_f) = 0 \tag{12}$$

The optimal control problem can thus be posed as a boundary value problem (BVP) through the indirect method, with the shooting function $\boldsymbol{\Phi} : \mathbb{R}^8 \to \mathbb{R}^8$ given by

$$\boldsymbol{\Phi}(\boldsymbol{\lambda}(0), t_f) = \begin{bmatrix} p(t_f) - p^* \\ f(t_f) - f^* \\ g(t_f) - g^* \\ h(t_f) - h^* \\ k(t_f) - k^* \\ \lambda_L(t_f) \\ \lambda_m(t_f) \\ \mathcal{H}(t_f) \end{bmatrix} \tag{13}$$

Effectively, the variables $\boldsymbol{\lambda}(0)$ and t_f must be chosen to satisfy $\boldsymbol{\Phi} = \boldsymbol{0}$.

DATA SET PREPARATION

Training neural networks typically require large data sets, which may in some cases be prohibitively expensive to generate. However, in the case of learning optimal trajectories, large data sets may be generated efficiently through the so-called *backward generation of optimal examples*.[12] At its core, this method is a data augmentation scheme, where a large number of optimal trajectory data can be generated from a few pre-solved trajectories. The computational benefit is substantial, as solving the optimal control problem in an indirect manner can be a cumbersome task, especially for the mass optimal problem with $\varepsilon \to 0$. In this section, the specific implementation of the backward generation process that is used in this work.

Backward Generation of Optimal Examples

The data set is generated via two steps; initially, around 100 “seed” optimal trajectories are computed between randomly sampled initial states $\boldsymbol{x}(0)$ and the target slow states $\boldsymbol{x}^*_{oe}$ from a pre-defined set $\boldsymbol{X}$. Then, the final states and costates of each seed optimal transfer are perturbed and corrected to meet the transversality conditions (13). Finally, the new, perturbed states and costates are propagated backward in time, where a sample is recorded at multiple points in time along each propagation.

Seed Generation To design the seed optimal transfer, the indirect optimal control problem must be solved. This is done via single shooting with equation (13) as the shooting function. A value of $\varepsilon = 1$ is used to increase the radius of convergence. Then, ε is gradually reduced and the shooting problem is resolved, using the previous solution as initial guess, until we arrive at a solution with $\varepsilon = 10^{-5}$.

Seed Perturbation Once a seed optimal transfer is obtained, we seek to modify its final states and costates in such a way that the new pair of states and costates still obey the transversality conditions (13). This resulting pair corresponds to the final targeted states and corresponding costates of an entirely different optimal transfer than its seed.

Obtaining a new valid states-costates pair first involves perturbing the seed final states and costates; this will, in all practical cases, violate the transversality conditions and therefore warrants a corrective step. Note that in Izzo and Öztürk,[12] the data set consisted of orbit transfers heading to Venus orbit; hence, the target slow states were fixed, and the costates perturbation was chosen in such a way that the transversality is satisfied. In contrast, in this work, since the final states are also to be perturbed, a perturbation-correction scheme is necessary.

Let $\boldsymbol{y}^*_f \in \mathbb{R}^{14}$ be the final states and costates of a seed optimal transfer. We initially perturb this by

$$\begin{aligned} \tilde{\boldsymbol{y}}^{(0)}_f &= \boldsymbol{y}^*_f + \delta \boldsymbol{y} \\ &= \boldsymbol{y}^*_f + \begin{bmatrix} \delta p & \delta f & \delta g & \delta h & \delta k & \delta L & \delta\lambda_p & \delta\lambda_f & \delta\lambda_g & \delta\lambda_h & \delta\lambda_k & 0 & 0 \end{bmatrix}^T \end{aligned} \tag{14}$$

Note that $\delta\lambda_L = \delta\lambda_m = 0$ since the transversality conditions requires $\lambda_L = \lambda_m = 0$. Assuming that the new optimal transfer sought has a target state that is moved by $\begin{bmatrix} \delta p & \delta f & \delta g & \delta h & \delta k & \delta L \end{bmatrix}$, the only transversality condition that is yet to be satisfied is the Hamiltonian $\mathcal{H}(t_f) = 0$. Let $R_{\mathcal{H}}$ be the residual of the Hamiltonian, given by the first two terms of $\mathcal{H}$,

$$R_{\mathcal{H}}\big|_{\boldsymbol{y}=\tilde{\boldsymbol{y}}_f} = [u - \varepsilon \log [u(1-u)]] + \frac{c_1}{m} \boldsymbol{\lambda}^T \boldsymbol{B}(\boldsymbol{x}_{oe}) \boldsymbol{u} \tag{15}$$

Then, the aim is to correct the perturbations vector $\delta \boldsymbol{y}$ such that $R_{\mathcal{H}} = 0$. This is a multi-variable root-solving problem with 12 variables (where λ_L and λ_m are dropped since they are fixed to 0) and a scalar root. Hence, a minimum-norm update law given by

$$\delta \tilde{\boldsymbol{y}}_f^{(i+1)} = \delta \tilde{\boldsymbol{y}}_f^{(i)} - \left(DR^{(i)}\right)^T \left[(DR^{(i)})(DR^{(i)})^T\right]^{-1} R_{\mathcal{H}} \left(\delta \tilde{\boldsymbol{y}}_f^{(i)}\right) \tag{16}$$

where $DR^{(i)}$ is the Jacobian given by

$$DR^{(i)} = \left.\frac{\partial R_{\mathcal{H}}}{\partial \tilde{\delta \boldsymbol{y}}_f}\right|_{\boldsymbol{y}=\boldsymbol{y}_f^* + \delta \tilde{\boldsymbol{y}}_f^{(i)}} \tag{17}$$

is used until $R_{\mathcal{H}} \approx 0$ within a tolerance, set to 10^{-10}. Note that the minimum-norm update is beneficial in our case as it allows for the convergence to the closest local root from the initially perturbed states and costates $\tilde{\boldsymbol{y}}_f^{(0)}$.

NEURAL NETWORK DESIGN

The aim of this work is to design a NN that can be used as part of a feedback controller of the spacecraft. Previous studies have looked into training a NN to learn the control inputs, namely the thrust magnitude and the thrust direction, directly.[9–13] As an alternative approach, this work proposes learning the costates and leveraging optimal control theory to obtain the control inputs.

The proposed method has a few notable advantages; on top of being able to guide the spacecraft's path via feedback, the obtained trajectory's optimality (13) can also be checked as the costates history is obtained as well. Furthermore, the NN can be leveraged to generate an initial guess to obtain the optimal trajectory via multiple shooting, by first obtaining a transfer via feedback, then sampling a few points along the transfer as initial guesses for the control nodes.

Network Design

The network's inputs and outputs, the loss function, and the overall structure of the network are now described.

Inputs To predict the optimal control, the NN requires the current state of the spacecraft $\boldsymbol{x} \in \mathbb{R}^7$. In addition, if the NN is to be used for orbit transfers to various final orbits, the targeted slow states $\boldsymbol{x}_{oe}^* \in \mathbb{R}^5$ must also be included as inputs. Note that the true longitude L is a periodic variable, which can become a nuisance when trying to learn the behavior of the dynamics. Intuitively, considering the case where L is defined in the range $[0, 2\pi]$, for some small ϵ, the behavior of the dynamics for $L \approx \epsilon$ and $L \approx 2\pi - \epsilon$ is expected to be similar, even though the difference of L is close to 2π. To avoid this issue, the true longitude may be expressed in terms of two auxiliary variables L_x and L_y, both defined in $[-1, 1]$, such that

$$L_x = \cos(L), \;\; L_y = \sin(L) \tag{18}$$

and

$$L = \operatorname{atan2}(L_y, L_x) \tag{19}$$

By replacing L with L_x and L_y, the network input dimension increases by 1.

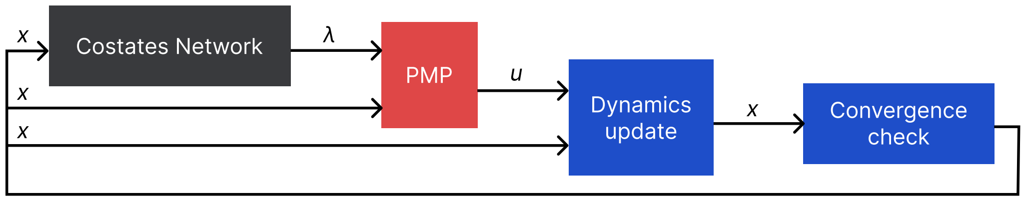

Figure 2. Schematic for feedback controlled dynamics with costates-feedback controller

Outputs The outputs must give sufficient information with reasonable accuracy to recreate the optimal control $\boldsymbol{u}$. In this work, we design the NN to return the costates $\boldsymbol{\lambda} \in \mathbb{R}^7$ that corresponds to the inputs, from which the optimal control can be reconstructed via Pontryagin's minimum principle.

Loss Function The choice of the loss function has a critical impact on the performance of the neural network. Fundamentally, the regression process involves approximating the true value via the prediction of the model; the mean-squared error (MSE) is

$$\ell_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^{N} (\boldsymbol{z}_i - \hat{\boldsymbol{z}}_i)^2 \tag{20}$$

where $\boldsymbol{z}$ are the true and $\hat{\boldsymbol{z}}$ are the predicted values of the outputs. While not included in this work, the use of additional terms such as the residual on the transversality condition may also be considered, as was explored in Izzo and Öztürk.[12]

Network Structure A few network structure, along with activation functions, has been experimented. As suggested by previous works, this type of application tends to require fewer hidden layers than other, more complicated learning tasks.[12] Here, we employ 3 fully-connected hidden layers, each with 256 neurons, using a ReLU activation function.

Feedback Controller Scheme

The costates-feedback controller consists of using the NN indirectly to generate the control input $\boldsymbol{u}$. Specifically, the NN takes in the states as input and outputs the costates $\boldsymbol{\lambda}$; through PMP, $\boldsymbol{u}$ is determined by applying equations (9) and (10). Figure 2 shows the feedback architecture with this controller. The black box at the top left is the NN, while the red PMP box corresponds to the evaluation of the two aforementioned equations. Taking the state $\boldsymbol{x}$ and the control $\boldsymbol{u}$, the state is updated through the dynamics, and after each state update, convergence to the target is checked. Note that although the costates are approximated at each time-step through the NN, it is never propagated.

RESULTS

The proposed NN architecture is applied for orbit transfer problems to NEOs. Firstly, the training data set is summarized. Then, the performance of the neural network for varying hyper-parameters and the choice of architectures are summarized.

NEO data set

The data set used for this experiment consists of 28,054,626 training data and 7,013,657 testing data. Figure 3 shows the distribution of the initial Keplerian elements from which optimal transfers

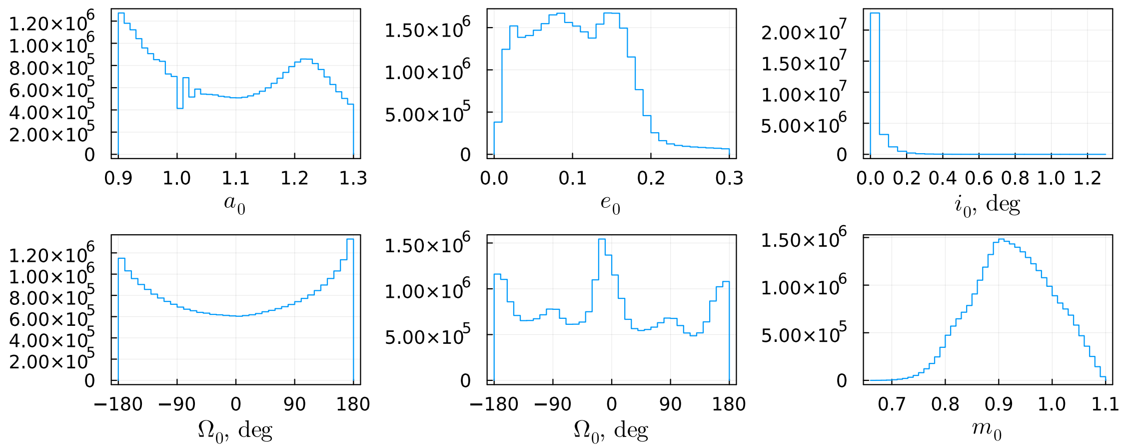

Figure 3. Distribution of asteroid Keplerian elements in training data

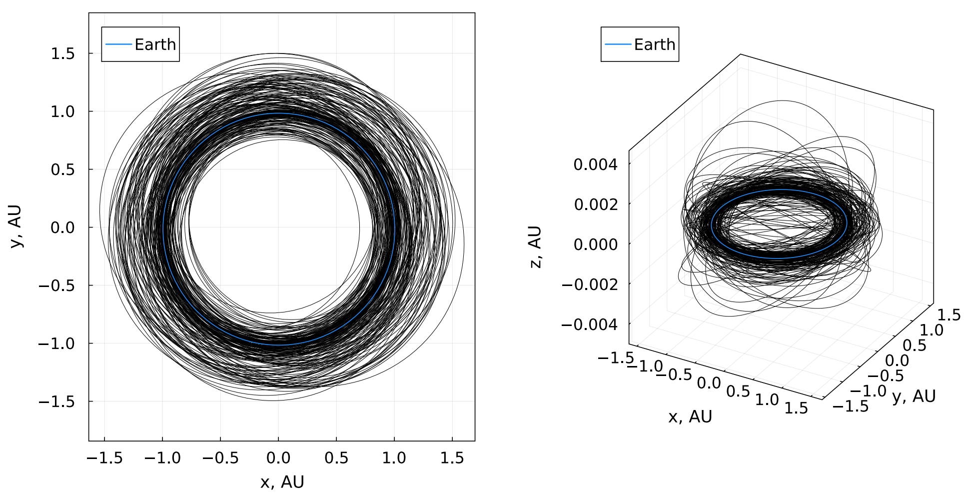

Figure 4. Sample of 200 training orbits from which the spacecraft departs, targeting the Earth's orbit

to Earth, whose slow elements are given by

$$\begin{bmatrix} p^* \\ f^* \\ g^* \\ h^* \\ k^* \end{bmatrix} = \begin{bmatrix} 0.99969 \\ -0.00376 \\ 0.01628 \\ -7.702 \times 10^{-6} \\ 6.188 \times 10^{-7} \end{bmatrix} \tag{21}$$

Figure 4 shows the orbit of 200 randomly selected samples from the training data-set. As seen in these two Figures, the considered data-set in this work consists of a relatively larger range in semi-major axis and eccentricity and a relatively smaller range in inclination. We note that this small inclination renders the data-set easier to learn, as the initial and targeted orbital planes are not too far from being aligned.

Table 1. Keplerian elements of test asteroid problem

SMA, DU	ECC	INC, deg	RAAN, deg	AOP, deg	TA, deg
1.287430	0.213730	0.124686	-119.976687	-53.522961	229.592676

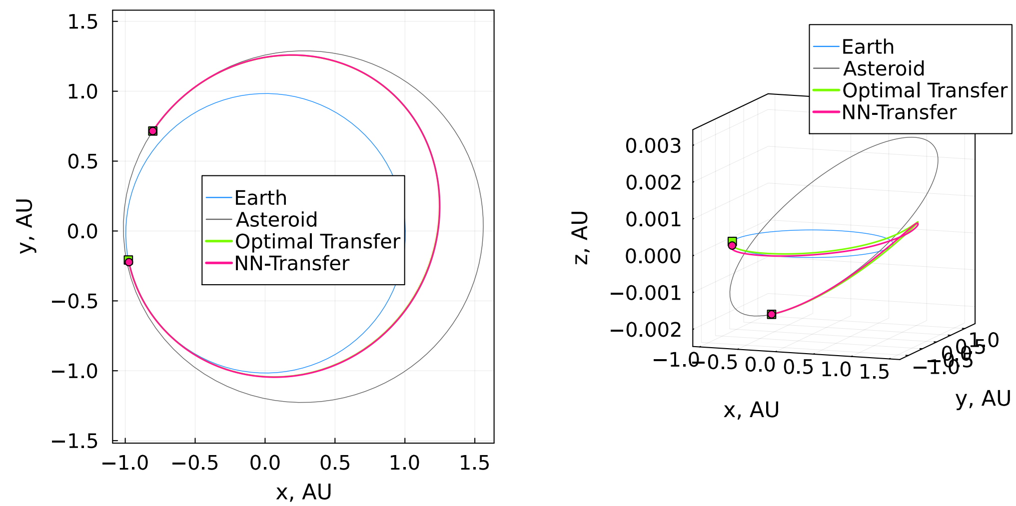

Figure 5. Example optimal transfer (guide) and costates-feedback-controlled transfer (pink) to asteroid

Transfer Design Application

The trained network's performance is tested for transfer problems to asteroids. As an example, consider the transfer problem to a fictitious object with Keplerian elements given in Table 1. The actual optimal transfer and the NN feedback controlled transfer are shown in Figure 5 and the time-history of the costates, control magnitude, and 2-norm distance from the target elements (i.e. the Earth's elements) are shown in Figure 6.

The trained network is used to design transfers from a sample of 70,136 cases in the test data. Among them, 96.6% of the cases arrived at the targeted semi-parameter p^* within an error of 0.001, or an offset of 1.495×10^5 km. From these cases, Figure 7 shows the distribution of the final mass and time of flight offset, as well as the 2-norm offset of the final spacecraft elements from the target state $\boldsymbol{x}_{oe}^*$.

CONCLUSION

In this work, the use of a neural network for learning mass-optimal orbit transfers has been explored. The proposed network is able to bring the spacecraft from initial orbits that are not restricted to a single orbit (such as that of a planet) to Earth with a success rate of over 95%. Overall, the proposed framework provides an inexpensive approach that enables fast evaluation of these optimal transfers; the data generation can be completed by simply solving initial value problems by propagating arbitrary state and costate pairs that satisfy the transversality condition. The neural network itself does not consist of many hidden layers compared and may be trained with a moderate amount of time and hardware resources. Finally, the trained network may be used as a

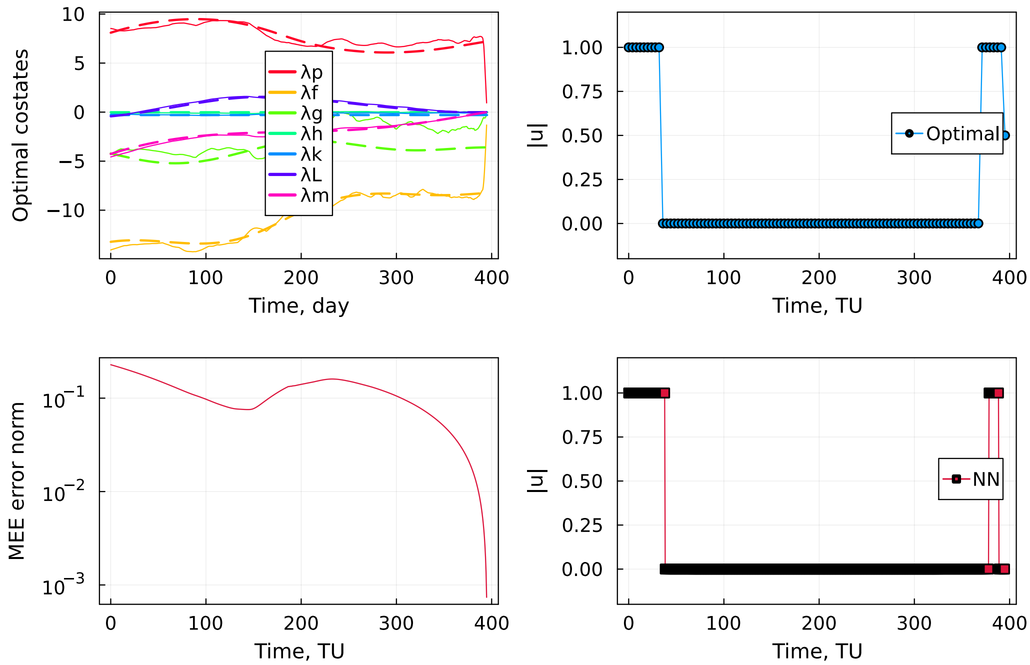

Figure 6. Time-history of costates, 2-norm offset from target elements, and control magnitude

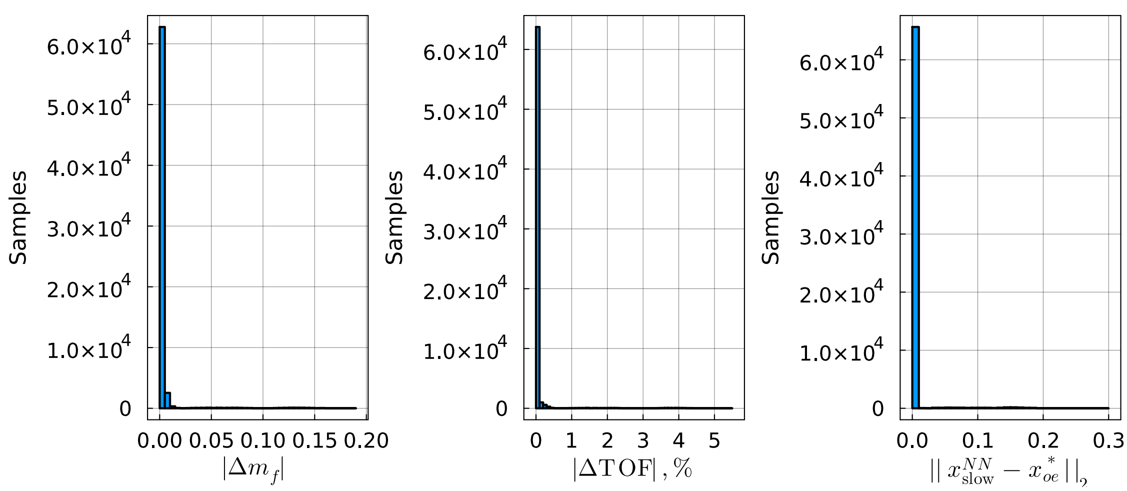

Figure 7. Histogram of final mass, time of flight, and states offset among converged cases, corresponding to 96.6% of the total test cases

feedback controller, thus providing near-optimal transfers in a non-iterative manner.

The use of a neural network is particularly useful in contexts where a fast evaluation of approximate low-thrust transfers forms the building block of a larger problem. Space logistics problems such as crewed campaigns of deep space, on-orbit servicing, as well as complex trajectory design challenges such as the Global Trajectory Optimisation Competition (GTOCs) are examples where this type of tool may lend itself well.

It is noted that if exact optimal transfers that meet transversality conditions at a higher precision are required, the neural network-based feedback control alone is inadequate. Nevertheless, since the network is constructed to return the costates as opposed to the controls, it can be used to generate initial guesses for indirect method-based trajectory optimization problems with an arbitrarily fine temporal mesh.

REFERENCES

[1] A. E. Petropoulos, "Simple Control Laws for Low-Thrust Orbit Transfers," *AAS/AIAA Astrodynamics Specialists Conference*, 2003.

[2] A. E. Petropoulos, "Low-thrust orbit transfers using candidate Lyapunov functions with a mechanism for coasting," *AIAA/AAS Astrodynamics Specialist Conference*, No. August, 2004, 10.2514/6.2004-5089.

[3] A. E. Petropoulos, "Refinements to the Q-law for low-thrust orbit transfers," *AAS/AIAA Space Flight Mechanics Meeting*, 2005.

[4] B. B. Jagannatha, J. B. H. Bouvier, and K. Ho, "Preliminary design of low-energy, low-thrust transfers to halo orbits using feedback control," *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 2, 2019, pp. 260–271, 10.2514/1.G003759.

[5] R. Epenoy and D. Pérez-Palau, "Lyapunov-based low-energy low-thrust transfers to the Moon," *Acta Astronautica*, Vol. 162, No. December 2018, 2019, pp. 87–97, 10.1016/j.actaastro.2019.05.058.

[6] J. L. Shannon, M. T. Ozimek, J. A. Atchison, and C. M. Hartzell, "Q-law aided direct trajectory optimization of many-revolution low-thrust transfers," *Journal of Spacecraft and Rockets*, Vol. 57, No. 4, 2020, pp. 672–682, 10.2514/1.A34586.

[7] M. Wijayatunga, R. Armellin, H. Holt, L. Pirovano, and A. A. Lidtke, "Design and Guidance of a Multi-Active Debris Removal Mission," 2022, 10.48550/ARXIV.2210.11701.

[8] H. Holt, R. Armellin, N. Baresi, Y. Hashida, A. Turconi, A. Scorsoglio, and R. Furfaro, "Optimal Q-laws via reinforcement learning with guaranteed stability," *Acta Astronautica*, Vol. 187, No. March, 2021, pp. 511–528, 10.1016/j.actaastro.2021.07.010.

[9] C. Sánchez-Sánchez, D. Izzo, and D. Hennes, "Learning the optimal state-feedback using deep networks," 2016, pp. 1–8.

[10] C. Sánchez-Sánchez and D. Izzo, "Real-time optimal control via deep neural networks: Study on landing problems," *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 5, 2018, pp. 1122–1135, 10.2514/1.G002357.

[11] H. Li, H. Baoyin, and F. Topputo, "Neural networks in time-optimal low-thrust interplanetary transfers," *IEEE Access*, Vol. 7, 2019, pp. 156413–156419, 10.1109/ACCESS.2019.2946657.

[12] D. Izzo and E. Öztürk, "Real-time guidance for low-thrust transfers using deep neural networks," *Journal of Guidance, Control, and Dynamics*, Vol. 44, No. 2, 2021, pp. 315–327, 10.2514/1.G005254.

[13] D. Izzo and S. Origer, "Neural representation of a time optimal, constant acceleration rendezvous," *Acta Astronautica*, No. August, 2022, 10.1016/j.actaastro.2022.08.045.

[14] J. Shi, J. Wang, L. Su, Z. Ma, and H. Chen, "A Neural Network Warm-Started Indirect Trajectory Optimization Method," *Aerospace*, Vol. 9, No. 8, 2022, 10.3390/aerospace9080435.

[15] E. Schiassi, A. D'ambrosio, K. Drozd, F. Curti, and R. Furfaro, "Physics-Informed Neural Networks for Optimal Planar Orbit Transfers," *Journal of Spacecraft and Rockets*, Vol. 59, No. 3, 2022, pp. 834–849, 10.2514/1.A35138.