IAC-19-C1.2.7

Artificial Neural Network For Preliminary Multiple NEA Rendezvous Mission Using Low Thrust

Giulia Viavattene[a]
g.viavattene.1@research.gla.ac.uk

Matteo Ceriotti[b]
matteo.ceriotti@glasgow.ac.uk

[a,b] *James Watt School of Engineering, University of Glasgow, James Watt South Building, Glasgow G12 8QQ, UK.*

Abstract

Since the 1960s the study of near-Earth asteroids (NEAs) has become extremely interesting for science, Earth protection, and future exploitation of their resources. The knowledge of these objects can be considerably improved by multiple NEA rendezvous missions with close-up observations of several asteroids. Given the enormous number of NEAs, which have been discovered until now, it becomes paramount to develop a method for quick identification of the transfer time and cost. This work develops a methodology based on Artificial Neural Networks (ANN) to identify a preliminary multiple NEA rendezvous trajectory using low-thrust propulsion. It takes advantage of the ANN capability to map the transfer time and cost starting from parameters that can describe the initial and final orbits and boundary conditions of the transfer. The ANN architecture and parameters are tuned to provide an optimal performance. The outcome of the network is used as input in a combinatorial problem to search for the asteroid sequence to visit, where a tree-search method is employed. Once the multiple rendezvous sequence is identified, the feasibility of the transfer with the given propulsion system is studied. Thus, an optimal control problem is solved for each leg by means of an optimisation solver based on pseudospectral method. The performance of the presented method is assessed by conducting analyses of sequences of asteroids of interest using different low-thrust options, such as solar electric propulsion and solar sailing.

Keywords: neural network, asteroid, rendezvous, trajectory optimisation

Nomenclature

a_c	Sail Characteristic Acceleration [m/s^2]
a_{max}	Low-Thrust Maximum Acceleration [m/s^2]
b_j	Network Biases [-]
e	eccentricity [-]
f, g	In-plane Modified Equinoctial Elements [-]
$\mathcal{F}^l$	Activation function [-]
i	inclination [deg]
I_{sp}	Specific impulse [s]
j, k	Out-of-plane Modified Equinoctial Elements [-]
L	True longitude [deg]
m	Spacecraft mass [kg]
$\mathbf{N}$	Unit direction vector [-]
$nrev$	Number of revolutions [-]
p	Semilatus rectus [m]
$\mathbf{r}$	Sun-Spacecraft position vector [m]
t	Time [s]
t_i	Network Targets [varies]
T_{max}	Maximum Thrust [N]
$\mathbf{u}$	Control vector [-]
$\mathbf{x}$	State vector [varies]
w_{jk}^l	Network weights [-]
y_j^l	Network output [varies]
α	Cone angle [deg]
ΔV	Velocity increment [km/s]
$\lambda_{1,2,3}$	Shaping parameters [-]
μ	Gravitational constant [m^3/s^2]
ϕ	Phasing parameter [deg]

Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Network
COE	Classical Orbital Elements
DNN	Deep Neural Network
EE	Equinoctial Elements
GA	Genetic Algorithm
LM	Levemberg-Marquardt
MEE	Modified Equinoctial Elements
MOID	Minimum Orbit Intersection Distance
MSE	Mean Squared Error
NEA	Near-Earth Asteroid
NEO	Near-Earth Object
NHATS	NEO Human Space Flight Accessible Targets Study
NN	Neural Network
OCP	Optimal Control Problem
PHO	Potentially Hazardous Objects
SEP	Solar Electric Propulsion
SS	Solar Sailing
TOF	Time of Flight

1. Introduction

Asteroids, in particular near-Earth asteroids (NEAs), have caught the attention of scientists from all over the world. In the past they have significantly contributed to the geological and biological formation of our Planet through their collisions. Potentially Hazardous Objects (PHO), which are characterised by Earth Minimum Orbit Intersection Distance (MOID) lower than 0.05 AU and estimated diameter greater than 150 m, pose a potential threat to the Earth [1].

Recently, the exploration of NEAs has become the main goal of many space missions. Close-up observations of NEAs can improve the knowledge of these objects in terms of shape, density, gravity field, and all their characteristics that are hard to determine or predict given the high irregularity that characterises NEAs. Multiple NEA rendezvous missions are preferred to missions towards one asteroid because of the reduced cost for each transfer and the increased range of possibilities to visit multiple NEAs of interest, given the lack of information that makes the choice of a single asteroid difficult.

Low-thrust propulsion systems are particularly suited to perform this type of high-energy interplanetary missions, taking advantage of their capability to deliver the same ΔV with less propellant required than high-thrust systems. Indeed, a low-thrust propulsion system allows a small, but continuous and efficient thrust for a long period of time [2]. Options of low-thrust propulsion are, for example, electric propulsion and solar sailing. Electric propulsion systems can use the high-voltage electric fields or electromagnetic fields to ionize and accelerate the propellant [3]. They yield specific impulses that exceed those of chemical rockets by approximately one order of magnitude. One type of electric propulsion is the solar electric propulsion (SEP), where large solar cell arrays convert sunlight to electrical power [4].

Differently, a solar sail is a large, lightweight and highly-reflective membrane that, once deployed from the spacecraft, uses the reflection of photons coming from the Sun to generate thrust. This results in a small, but continuous free acceleration over time. Since it does not need any propellant and it can potentially generate thrust for an extended time, solar sailing is an attractive solution for high-ΔV missions [5].

The preliminary design of multiple NEA missions constitutes a complex global optimisation problem. This problem can be divided into two parts: a large discrete combinatorial part, which consists in the selection of the asteroid sequences, and a continuous part that implies the resolution of the spacecraft optimal control problem (OCP) from the initial to the final conditions. It should be noticed that, as the search space increases, the complexity of the global sequence search grows factorially [6]. Different methods have been developed to solve this problem. The majority of them proposes to compute the cost of a transfer by means of a simplified model and, successively, use a tree-search method to search the sequence of asteroids. For instance, Peloni et al. [7] approximated the trajectories using a shape-based approach and determined the sequence of asteroids through a search-and-prune algorithm. They were able to design a mission to visit five NEAs within 10 years using near-term solar-sail technology [8]. Other options were advanced by other authors to improve the accuracy and efficiency of the low-thrust transfers.

This project aims to investigate how artificial intelligence (AI) can be exploited to identify the optimal NEA sequence to visit. Previous works demonstrated how the use of AI can be advantageous in solving complex problems in the field of aerospace engineering and interplanetary mission design. Hennes et al. [9] used machine learning methods to identify the low-thrust trajectory with minimum fuel mass required to perform transfers between the main belt asteroids. Mereta et al. [10] compared the accuracy of various machine learning techniques, including Artificial Neural Network (ANN), to determine the initial guess of optimal low-thrust transfers between Near-Earth Objects (NEOs). In both applications the machine learning approach proved to be superior with respect to the commonly-used impulsive Lambert predictor. Song and Gong [11] designed the trajectory of the multi-target rendezvous problem, using solar sailing with a characteristic acceleration $a_c = 0.75$ mm/s^2. An ANN is used to estimate the cost of orbital transfers. The achieved fitting accuracy of about 97% of data outside the training set suggests the effectiveness of the method. A database restricted to all the NEAs with semi-major axis $0.8 < a < 1.2$ AU, eccentricity $e < 0.2$ and inclination $i < 11.5^\circ$ was used for the training.

From the the previous works it is possible to identify three key requirements of a multiple NEA rendezvous mission:

1. reducing the total mission duration or increase the number of encounters for the same mission duration
2. reducing the required acceleration
3. including asteroids of interest for scientific purposes or Earth defence in the sequence of asteroids to visit

This paper will show a methodology to train a neural network to map the cost of a trajectory and its orbital characteristics. Once trained, a neural network can potentially explore *quickly* any possible transfer between pairs of asteroids, eliminating the need of considering pruned sets of asteroids. Indeed, the training of the network can require some time, but once it is trained it is able to provide results in a immediate manner, reducing considerably the computational time and effort.

Once the preliminary cost of the possible transfers is estimated, the optimal sequence of asteroids needs to be searched. This is firstly a combinatorial problem given the large number of objects and the extensive amount of permutations between them, which can be solved using the tree-search method. The feasibility of the obtained sequences is verified thought an optimisation process based on Radau pseudospectral method.

The paper is organised as follows. In Section 2, a neural network for a quick estimation of the cost of interplanetary transfers is built and trained with a database of transfers between NEAs generated using a shape-based method. Section 3 describes the algorithm implemented to search for the sequences of NEAs, while in Section 4 the optimisation method, used to verify the feasibility of the sequences, is illustrated. The results and the performance of the method are discussed in Section 5. Finally, Section 6 completes this paper with the conclusions.

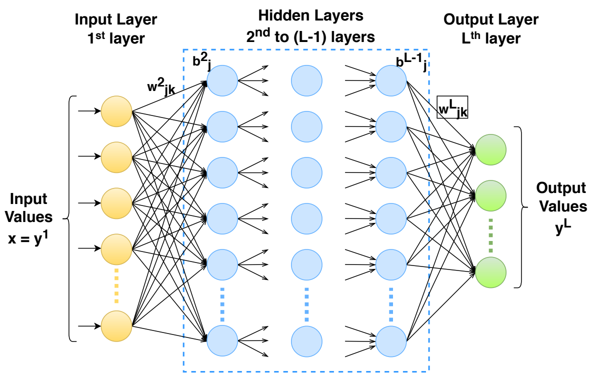


Figure 1: Illustration of an artificial Neural Network (ANN) with *L* layers.

2. Neural Network for Transfer Cost Mapping

The aim of this work is to explore how neural networks can be employed to find the best sequence of multiple asteroids to visit during the multiple rendezvous mission, and whether this improves the performance with respect to previously employed methods.

To identify the best sequence of NEAs, all the combinations of asteroids should be explored and evaluated. According to the NASA's database, almost 20,000 NEAs have been discovered until now, of which almost 2,000 are classified as PHO. It follows that multiple trillions of permutations between these objects needs to be investigated. Since low-thrust transfers have no analytical closed-form solutions, an optimisation strategy must be used to find a solution to trajectory design problems, which are generally computationally demanding.

Instead of this, we propose to use ANN, which can be trained to calculate, in a fraction of time needed with optimisation solvers, the cost of a trajectory in terms of ΔV and time of flight (TOF), given the departure and arrival orbit. This provides an estimate that can help to select candidates for further analysis through optimal control problems. Moreover, if trained accurately, ANN can generalise to future epochs and different space objects with respect to those considered during the training [12]. This is known as generalisation propriety and gives the opportunity to analyse all the possible combinations of NEAs in the sequence, without any need to reduce the database of asteroids considered in the analysis.

A wide variety of Neural Networks (NN) exists to be suitable for multiple applications. Here, however, only feedforward ANN are considered. In this kind of networks the neurons of a layer are connected directly to neurons of the successive layer, so that the information moves in one direction only, from input layer through the hidden layers to the output layer. Figure 1 presents a general illustration of a neural network with L layers. Similarly to the brain, the structure of a neural network is organised in layers. The first layer has inputs $x = y^1$, which are outputted to the next layer, the hidden layers are those comprised between the 2^{nd} and $(L-1)$-th layers, and the output layer is the L-th layer with final output values y^L.

The network needs to be trained by means of a database containing the corresponding inputs and outputs (or *targets*) with the purpose of *learning* the network function that correlates inputs and targets. The network function is intended to minimise the difference between the outputs generated by the network and the targets, i.e., to minimise the network error. The training of the network consists in determining the weights w^l_{jk} associated to each connection between the k-th and j-th neuron and the biases b^l_j of the j-th neuron of the l-th layer. So, the j-th neuron of the l-th layer equals to:

$$y^l_j = \mathcal{F}^l\left(\sum_k w^l_{jk} y^{(l-1)}_k + b^l_j\right) \tag{1}$$

where subscripts and superscripts identify the neuron and hidden layer, respectively. $\mathcal{F}^l$ is the transfer or activation function of the l-th layer, which maps from the neuron's weighted input values onto a single output value. The most commonly used activation function for feedforward networks is the *sigmoid* function, which is defined as follows:

$$s_\gamma(x) = \frac{1}{1 - e^{-x/\gamma}} \tag{2}$$

where γ defines the slope of the function. Using the sigmoid function as activation function, each j-th neuron of the hidden and outer layers is computed by means of the Eq. 1, as follows:

$$y_j = \frac{1}{1 - e^{-\left(\sum_k w^l_{jk} y^{(l-1)}_k + b^l_j\right)/\gamma_j}} \tag{3}$$

with y_j being the output of the j-th neuron and $y_j \subseteq (0, 1)$. Differently, each i-th neuron of the input layer provides one component of the input vector, $X_i \subseteq \mathbb{R}$. It follows that the *network function* is a parameterised function that maps from an input set $\mathcal{X}$ to an output set $\mathcal{Y}$:

$$N = \mathcal{X} \subseteq \mathbb{R}^{n_i} \rightarrow \mathcal{Y} \subseteq (0, 1)^{n_0} \tag{4}$$

with n_i and n_0 being the number of input and output neurons, respectively.

The architecture of a NN changes by modifying the number of layers and neurons for each layer. By changing its architecture, it is possible to influence the network's performance. Accordingly, it is essential to identify the best network's architecture to minimize the network error. For example, for certain applications, an ANN with multiple extensive layers between the input and output layers, also known as deep NN (DNN), can predict the solution to an OCP with higher accuracy [13, 14]. In this paper the best network structure for this application is determined.

The performance of the network is affected also by other parameters of the network, such as learning algorithm, activation function, learning rate, or even the type of parameters contained in the input vector. To design an optimal neural network, the training database is generated, as presented in Section 2.1, and the network's performance is investigated to define its architecture and tune its parameters (Section 2.2).

2.1. Generation of Training Database

The objective of the neural network is to determine the cost of a transfer given the orbital characteristics of the

Table 1: Bounds definition for the parameters used in the shape-based method.

Parameter	Bounds
Launch date	2020/01/01 $< t_0 <$ 2030/12/30
TOF	$400 < TOF < 1500$ days
Acceleration	$a_{max} < 0.2$ mm/s^2
Propellant ratio	$m_P/m_{TOT} < 0.6$
N. revolution	$0 < nrev < 4$
Shaping parameters	$\lambda_1 \in [-0.5, 0.5]$ $\lambda_2 \in [-0.1, 0.1]$ $\lambda_3 \in [-0.01, 0.01]$

departure and arrival orbits and the position of the objects along their orbits. To this end, a database needs to be generated that contains both the input vector, i.e., a parameterisation of the orbits and the position at a certain time of the objects, and the output vector quantifying the transfer cost in terms of ΔV and TOF.

To determine the cost of a low-thrust transfer, the transfer itself needs to be computed by solving an optimal control problem. To reduce the computational time and effort, techniques that can quickly compute a feasible, but less accurate solution can be used for a first estimation of the transfer. The shape-based method, firstly proposed by Petropoulos and Longusky [15], consists in defining a trajectory as a parametrised analytical curve connecting two points in a central force field. The necessary acceleration that the propulsive system needs to provide is obtained by computing the control thrust required to satisfy the dynamics, as follows:

$$\mathbf{a} = \ddot{\mathbf{r}} + \mu \frac{\mathbf{r}}{r^3} \tag{5}$$

where μ is the gravitational constant of the central body (the Sun) and $\mathbf{r}$ is the position vector in the Cartesian reference frame.

Amongst all the shapes for low-thrust problems proposed until now [16, 17, 18], the one proposed by De Pascale and Vasile [18] defines the solution of the full 3-D trajectory based on a set of modified equinoctial elements. They determined a linear-trigonometric shape as follows:

$$\mathbf{x} = \mathbf{x_0} + \mathbf{x_1}(L - L_0) + \lambda \sin(L - L_0 + \phi) \tag{6}$$

where $\lambda = [\lambda_1, \lambda_2, \lambda_3]$ is the vector of shaping parameters, $x = [p, f, g, h, k, L]^T$ is the vector of modified equinoctial elements, x_0 and x_1 are defined from the boundary conditions, and the phase parameter ϕ is empirically set [18].

This procedure can approximate the shape of the minimum-cost rendezvous trajectory in the given range of launch dates with zero departure and arrival velocity, TOF, and number of revolutions. In this work, the shape-based method is selected to generate the database of transfers to train the neural network.

For each pair of asteroids, the shape-based method aims at identifying the shape of the transfer, once the shaping parameters are specified, and retrieve the control history necessary to perform the obtained transfer. To this end, the MATLAB built-in genetic algorithm (GA) is run to determine the optimal shaping parameters for the transfer shape with minimum time of flight. The control history is changed by changing the shape, thus the shaping parameters, so that the acceleration constraint (Eq. 5) is satisfied [18]. Considering a SEP system with a specific impulse, I_{sp}, of 3000 s, the shape-based method is run to compute the transfers between NEAs. The bounds of the parameters required from this method are specified in Table 1.

Because of the capability of the shape-based method of finding solutions more easily when less eccentric and less inclined orbits are considered, NEAs with $e \leqslant 0.4$ and $i \leqslant 20^\circ$ constitute the training database. NEAs that are of particular interest from the scientific point of view or for their composition and orbital dynamics are included. Amongst these, Potentially Hazardous Asteroids (PHA), as defined before, and the Near-Earth Object Human Space Flight Accessible Targets Study (NHATS) [19, 20], selected by NASA as known NEAs that might be accessible by future human space flight missions. These asteroids are identified on the basis of the total ΔV required, total mission duration, stay time at the object and launch date interval. The number of objects considered is now 1313 of which about 300 are PHA and 24 are NHATS that are characterised by the following properties[1]:

- total ΔV required $\leqslant 6$ km/s
- total mission duration $\leqslant$ 450 days
- stay time at the object $\geqslant$ 8 days
- launch date: 2035-2040
- absolute magnitude parameter: H $\leqslant 26$
- orbit condition code: OCC$\leqslant 7$

where the absolute magnitude parameter H indicates the visual magnitude an observer would record if the asteroid were placed 1 AU away, and the orbit condition code (OCC) takes into account the orbit determination accuracy.

The NEA orbital characteristics are obtained from the NASA's Near-Earth Object Program[2]. The position of the asteroids along their orbits at a reference time of 2019/04/27 (2458600.5 Julian day) is considered. The combinations between only 100 NEAs are investigated for a total of 10,100 transfers stored in the *database* which is used to train the network. Once the training is completed, it is able to generalise to other NEAs and epoch, i.e., to provide an estimation of transfer cost between NEAs not included in the database and with different launch dates.

For the training process the database is divided into three sets, which are training, validation and test sets. The training set is used to train the network and obtain the weights and biases that minimise the mean square error (MSE) between the output of the network y and the targets t, which is defined as follows [21]:

$$MSE = \frac{1}{N} \sum_{i=1}^{N} ||y_i - t_i||^2 \tag{7}$$

[1] Data available through the link `https://cneos.jpl.nasa.gov/nhats/` (accessed on 2019/09/10)

[2] Data available through the link `https://cneos.jpl.nasa.gov/orbits/elements.html` (accessed on 2019/06/17)

Table 2: Network performance for different parameterisation of the orbit.

	Correlation	Validation-Set Error
COE	0.855	0.530
EE	0.856	0.487
MEE	0.925	0.236
Cartesian	0.551	0.761
Delaunay	0.694	0.862
eH	0.908	0.221

with N being the number of nodes of the output layer of the network.

Differently, the validation and test sets contain new samples for which the network is not trained, so they are used to verify the generalisation performance of the network. The validation set is used during the training to eventually detect evidence of overfitting. Indeed, this happens when the MSE of the validation set increases while the MSE of the training database continues to decrease. The training is stopped when the validation error increases after a defined number of iterations. In this case, the weights and biases are set equal to those for which the minimum of the validation error is registered. The test set is used after the training process to evaluate the performance of the network with totally new situations. The training algorithm used and the database division into training, validation and test sets are presented in the next section.

2.2. Network Architecture & Parameter Tuning

Within the MATLAB built-in neural network, many aspects need to be considered to improve the accuracy of the results, from the type of inputs to be fed into the network to the architecture of the network itself. There exists a combination of these aspects that provides the optimal performance of the network, however this is not known a priori and need to be determined by trial and error.

To analyse the performance of the network, a regression analysis between the network response and the corresponding targets shall be conducted. It gives an indication of how well the outputs represent the targets. This can be summarised by the correlation coefficient, which can vary from -1 to +1, with +1 indicating perfect matching between targets and outputs. Also, the final MSE of the validation set can give a better insight in the network performance as it is preferred to have this error as close to zero as possible.

As mentioned previously, the input to the network is chosen to be a parameterisation of the departure and arrival orbits, between which the cost of the transfer is to be computed. The use of different orbit parameterisations as input is studied with the intention to select the one for which is it easier for the network to learn a function which resembles more the complicated non-linear relationship between the orbital characteristics and the cost of the transfer. In addition to the classical orbital elements (COE), typically used for an intuitive representation of the orbit, and the modified equinoctial elements (MEE), which are used in the shape-based method, the performance of the network is analysed also with equinoctial elements (EE), Cartesian coordinates, Delaunay elements, and eccentricity and angular momentum vector (eH) [22, 23].

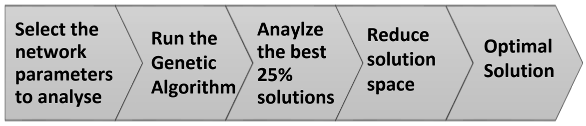


Figure 2: Methodology for Neural Network parameter tuning.

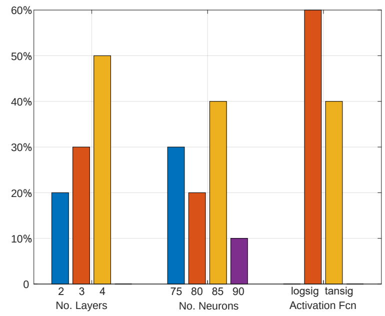


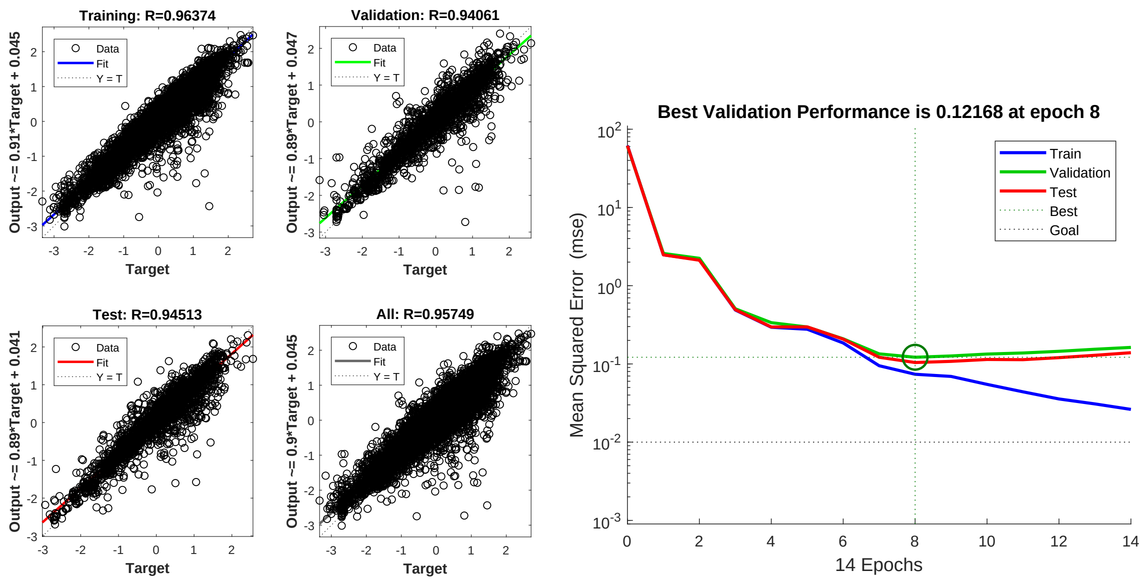
Figure 3: Percentage of the best 25% of solutions with given values of network parameters.

To this end, a network with two hidden layers and 80 neurons in each layer is generated. The activation function chosen is the *sigmoid* function and the *gradient-descent* algorithm is adopted for the training. The learning rate is set to 0.01, which is the highest value that does not cause divergence in the training process. The batch size is tuned to 200, and the database is divided so that 70% constitutes the training set, 15% the validation set, and 15% the test set. The performance of the network, in terms of correlation and network error in the validation set, is presented in Table 2 for different parameterisations of the orbit.

The best correlation is obtained when MEE are used as inputs ($C_{MEE} = 0.925$), with also a low validation-set error ($e_{MEE} = 0.236$). The latter is slightly lower ($C_{eH} = 0.221$) when the eH parameterisation is used, however a poorer correlation is registered ($C_{eH} = 0.908$). Given the small difference in validation-set error, priority is given in this case to the highest correlation, which is indeed calculated with respect to all the three training, validation and test sets. Thus, the neural network input is the departure and arrival orbits described by means of MEE.

Figure 2 illustrates the method adopted for the tuning of the network parameters. First, the parameters are changed one by one leaving the remaining parameters unvaried, so that the effect on the network performance due to the modification of each parameter can be visualised. In this way, the parameters that have more effect on the network outcome are identified. Secondly, the parameters that appear to affect more the performance are changed simultaneously through a MATLAB built-in genetic algorithm, which will search for the optimal combination of parameters [24].

The most used training algorithm for function approximation is the *Levemberg-Marquardt* (LM) gradient descent algorithm. It works as a standard back-



(a) Regression analysis

(b) MSE variation during the training

Figure 4: Regression Analysis of the network with the best combination of parameters identified by the genetic algorithm.

propagation, meaning that the error and the corresponding weight and bias variations are computed at the output and propagated backwards through the network's layers. The gradient and the variations of the weights and biases are computed in batch mode, i.e., once all the batch of inputs are applied to the network. The performance with different training algorithms, such as the BFGS quasi-Newton method and the resilient backpropagation [25], is studied. However, since a better outcome is obtained when the LM gradient algorithm is used, this is chosen for the current application.

The parameters that mostly influence the performance of the network are:

1. number of hidden layers
2. number of neurons of each hidden layers
3. activation function of each hidden layers
4. learning rate
5. batch size
6. database division rates (training, validation, test)

These parameters constitute the inputs to the genetic algorithm, which is run with fitness function being the minimization of the correlation coefficient. The learning rate, the batch size and the database division have a specific value, for which the performance is significantly better: the optimal learning rate is $l_r = 0.01$, the best batch size is 200, and the best database division is $7 : 15 : 15$ for training, validation, and test set.

Figure 3 presents the number of hidden layers, number of neurons and activation function of the hidden layers of the 25% of the results which shows better performance. A larger number of layers and neurons will increase the flexibility of the network introducing more weights. As a side effect more flexibility can induce to an overfitting of the data. Thus, it is paramount to identify the right combination. For a better performance the number of hidden layers shall be set between 2 and 4, the number of neurons in each hidden layer between 75 and 90, and the activation function being *logsig* or *tansig*. This confirms that, with a larger number for hidden layers and neurons, the outcome accuracy of the network improves.

The best performance is obtained when the network has four hidden layers with 80 neurons and the sigmoid as activation function. The performance of the selected network is presented in Figure 4. The regression plots present the network outputs with respect to the training, validation and test targets. A perfect fit is obtained when the data fall along a line with slope equal to 1 and the y-intercept is equal to 0. The final correlation obtained is about 0.96, which indicates an accurate fitting, with a MSE of the validation set of 0.12. When the test set is fed to the network or, in other words, when the network experiences new samples that are not included in the training, the total fitting accuracy is 0.94. This ensures that the network function describes well the relationship between departure and final orbits and cost of the transfer between the two. Moreover, this indicates that the network is able to generalise to new circumstances. It follows that, to a first approximation, the neural network can replace the complex optimisation process required to compute a low-thrust transfer and identify its cost and duration.

3. NEA Sequence Search

In this section the methodology to determine the most promising sequences of asteroids to visit in a multiple NEA rendezvous mission is presented. The implemented sequence search algorithm is schematically illustrated in Figure 6. The algorithm is based on a tree-search method, where each node represents a trajectory and how one proceeds through its branches depends on the mission objective.

Firstly, the full database of NEAs presented in Section 2.1 is loaded. The sequence search starts at the Earth at a specified time, which is chosen to be 2035/01/01

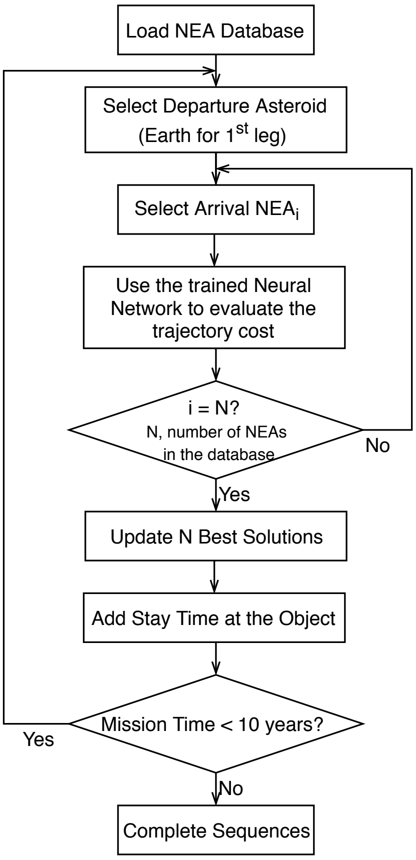


Figure 5: Flowchart of the sequence search algorithm.

($t_0 = 2464328.5$ Julian Day). Since the ephemerides of the bodies are defined in the database at a $t_{ref} = 2458600.5$ Julian day, the ephemerides are updated at each departure time t_i, with i indicating the i^{th} leg of the sequence. The trained network is applied to calculate the cost and the time of flight of each transfer from the Earth to all the NEAs available in the database. Once all the transfers are evaluated, only N_{max} = 100 of the best transfers are stored. The arrival object becomes the departure object of the following leg, for which the same procedure is iterated in the tree search. A stay time of 100 days is considered between two consecutive legs. The sequence is completed once the total mission duration reaches the maximum time allowed for the mission, which is set to 10 years.

The limit of N_{max} best transfers, which are stored at every iteration and used for the next iterations, is due to the tree nature of the search. Indeed, as the number of the feasible trajectory increases, the number of the successive transfers will increase exponentially as the iterations proceed. Examining only the N_{max} best transfers considerably reduces the computational time, while still achieving the goal of the search algorithm.

The sequence search algorithm results to be much faster to run than the ones implemented in previous works. Peloni et al. [8] carried the whole search by computing each trajectory by means of the shape-based method applied to solar sailing. This required to run a genetic algorithm every time to check the existence of each trajectory. The average computational time for each sequence search run was about 40 days, using a database containing only PHAs and NHATS asteroids. Using the same database and launch dates used in [7], the computational time reduces from 40 days to 1.7 day for each sequence run by using ANN. It should be noticed that the propulsion system used for this comparison is a solar electric propulsion with a maximum acceleration equal to the characteristic acceleration of the solar sail in the work of Peloni at al. With such a system, sequences of 7 encounters are identified during the search, instead of five as in [7].

4. Sequence Optimisation

Once the NEA multiple rendezvous sequence is defined, it needs to be optimised by solving a OCP to verify the feasibility of the transfer with the given propulsion system. To this end, two types of low-thrust propulsion systems are considered, the solar electric propulsion (SEP) and solar sailing (SS).

The state vector of the system, $\mathbf{x}$, is expressed in modified equinoctial elements and the mass of the system:

$$\mathbf{x} = (p, f, g, h, k, L, m) \tag{8}$$

and the control vector $\mathbf{u}$ coincides with the direction vector $\mathbf{N}$ in radial, transverse, out-of-plane coordinates:

$$\mathbf{u} = \mathbf{N} = (N_r, N_\theta, N_h) \tag{9}$$

The dynamics of the system under analysis can be described by the following set of ordinary differential equations of motion:

$$\dot{\mathbf{x}}(t) = \mathbf{A}(\mathbf{x})\mathbf{a} + \mathbf{b}(\mathbf{x}) \tag{10}$$

where the matrix $\mathbf{A}(\mathbf{x})$ and $\mathbf{b}(\mathbf{x})$ are the matrix and the vector of the dynamics, respectively, and $\mathbf{a}$ is the acceleration generated by the propulsion system. A full definition of $\mathbf{A}(\mathbf{x})$ and $\mathbf{b}(\mathbf{x})$ can be found in [26]. The acceleration can be described as follows:

- for solar electric propulsion

$$\mathbf{a} = \frac{T_{max}}{m}\mathbf{N} \tag{11}$$

  where T_{max} is the maximum thrust that can be generated from the considered propulsion system and m is the mass of the spacecraft, changing with time while it is thrusting.

- for solar sailing

$$\mathbf{a} = a_c \left(\frac{r_\oplus}{r}\right)^2 \cos^2\alpha \hat{\mathbf{N}} \tag{12}$$

  where a_c is the characteristic acceleration of the sail, i.e., the acceleration provided from a solar sail facing the Sun at the average Sun-Earth distance $r_\oplus$. The magnitude of the position vector with respect to the Sun is expressed by r, while the cone angle α is the angle between the Sun-spacecraft direction and and the sail normal unit vector $\hat{\mathbf{N}} = [N_r, N_\theta, N_h]^T$.

The OCP consists in determining the control $\mathbf{u} = \mathbf{N}$ so that the time of flight is minimised, while respecting the dynamic constraint defined in Eq. 10 and the path constraint, which is $0 < ||\mathbf{N}|| < 1$ for SEP to allow for the

Table 3: Mission parameters of the optimised NEA sequence with one PHA as encounter. The optimised parameters are compared with the ones from the NN in the sequence-search algorithm (in brackets).

Transfer	Stay Time [days]	Departure	Arrival	TOF [days]	ΔV [km/s]
Earth	–				
$\Downarrow$		2035/01/01	2036/03/23	447 (408)	6.29 (7.10)
2007 RN7	16				
$\Downarrow$		2036/04/08	2038/01/31	663 (500)	9.89 (8.84)
2002 RS129	263				
$\Downarrow$		2038/10/21	2039/10/25	369 (403)	5.09 (5.11)
2008 YT30	37				
$\Downarrow$		2039/12/01	2040/10/07	311 (414)	4.24 (4.88)
1999 RP28	63				
$\Downarrow$		2040/12/09	2042/05/26	533 (504)	7.6 (6.42)
2002 TS69	10				
$\Downarrow$		2042/06/05	2043/05/30	359 (401)	4.95 (3.91)
2002 GN5	94				
$\Downarrow$		2043/09/01	2044/09/29	394 (468)	5.47 (4.13)
2002 PF43*	–				

* PHA

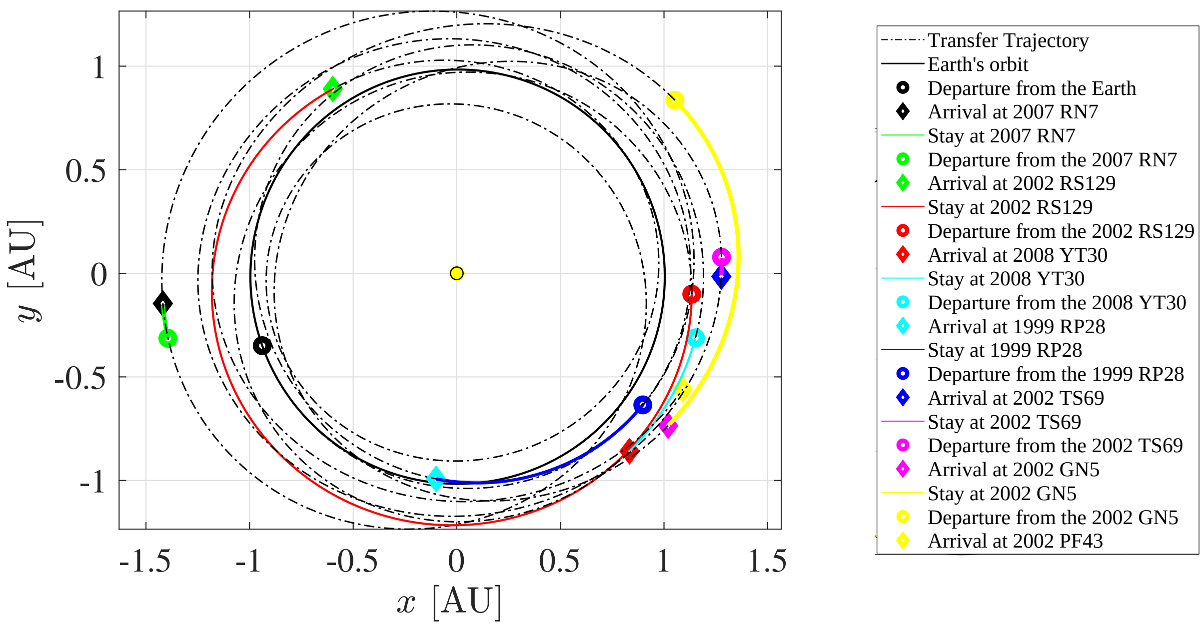


Figure 6: Heliocentric two dimensional view of the complete trajectory using SEP. First optimised sequence from Table 3.

Table 4: Mission parameters of the optimised NEA sequence with a PHA and a NHATS as encounters. The optimised parameters are compared with the ones from the NN in the sequence-search algorithm (in brackets). Also, the results when solar sailing is used are presented.

Transfer	Stay Time [days]	Departure	Arrival	TOF [days]	ΔV [km/s]
Earth	–				
⇓		2035/01/01 • 2035/01/01	2036/11/15 • 2037/03/12	684 (553) • 801	7.8 (7.04) • -
2011 AM24*	196 • 158				
⇓		2037/05/30 • 2037/08/17	2039/07/30 • 2039/11/10	791 (675) • 815	8.05 (6.9) • -
2003 MM	83 • 114				
⇓		2039/10/21 • 2040/03/03	2040/12/25 • 2041/04/27	431 (414) • 420	6.11 (6.25) • -
2006 SF6	134 • 184				
⇓		2041/05/08 • 2041/10/28	2043/04/15 • 2044/04/28	707 (524) • 913	9.04 (8.28) • -
2008 YT30	271 • 110				
⇓		2044/01/11 • 2044/08/16	2045/05/11 • 2045/11/21	486 (503) • 462	6.92 (5.24) • -
1999 FA	68 • 45				
⇓		2045/07/18 • 2046/01/05	2046/10/02 • 2047/01/24	441 (502) • 384	6.21 (4.67) • -
2019 FU2**	–				

* PHA
** NHATS
• Results using solar sailing.

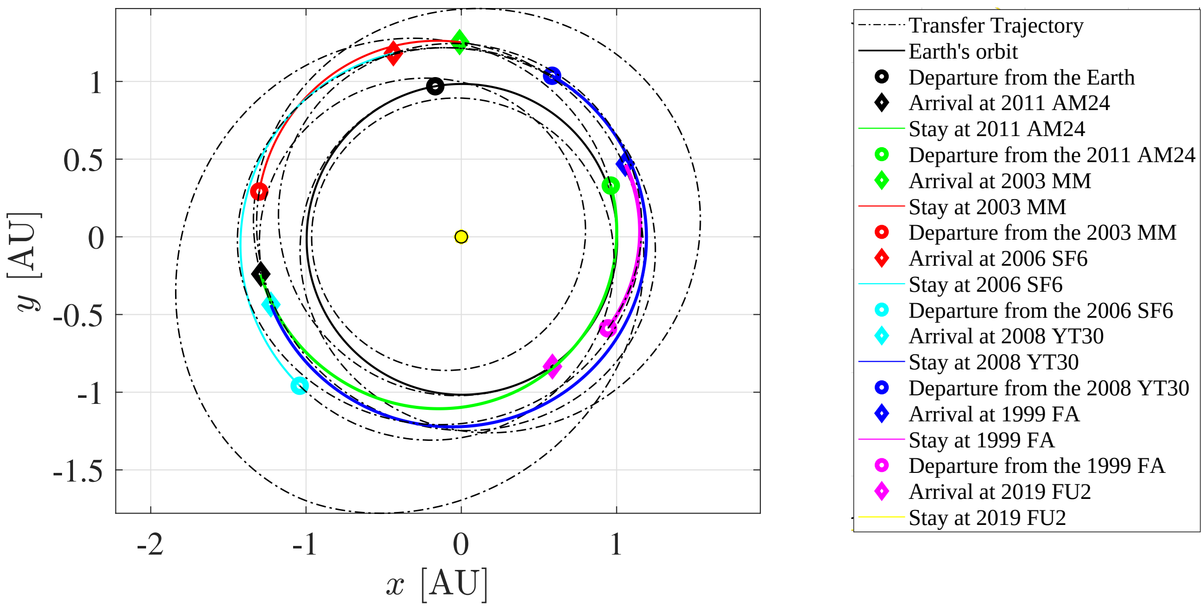


Figure 7: Heliocentric two-dimensional view of the complete trajectory using SEP. Second optimised sequence from Table 4.

throttle, and $||\mathbf{N}|| = 1$ for solar sailing. The bounds of the control vector is N_r, N_θ, $N_h \in [-1, 1]$ when a solar electric propulsion is used, while N_θ, $N_h \in [-1, 1]$ and $N_r \in [0, 1]$ in case of solar sailing to take into account the inability of the solar sail to thrust towards the sun.

The optimizer used for this purpose is GPOPS II, the general-purpose MATLAB software, which is based on pseudospectral method [27]. The OCP is transcribed into a discrete non-linear programming problem with finite dimensions by using the *ph* collocation methods [28, 29, 30]. The algorithm implemented to find the optimal low-thrust trajectory performs as follows:

1. the initial guess is computing by solving a Lambert problem, given the departure orbit, arrival orbit and desired time of flight, which is provided by the ANN during the sequence search
2. the optimised multiple NEA rendezvous trajectory is built if at least one feasible solution is found for each leg of the mission.

Each leg is optimised using as initial guess TOF the one predicted by the neural network during the sequence search. Considering the optimised arrival time of the previous leg, the departure time, t_0, of the following leg is set to allow a minimum stay time at the current object of 10 days. This ensures that no overlapping between two consecutive transfers occurs. Also, t_0 can vary up to a maximum of 100 days, so the best epoch for departure can be selected by the optimiser. Similarly, the arrival time, $t_f = t_0 + TOF$, can vary by $\pm$100 days with respect to the value provided by the network. The stay time at each asteroid is consequently adjusted considering the TOF of each leg and the departure date of the successive legs.

5. NEA Optimised Sequences

Two sequences have been selected and fully optimised to validate the proposed methodology. Considering a solar-electric propulsion system with a maximum acceleration of $a_{max} = 0.2$ mm/s^2, the two sequences have been selected amongst those with seven encounters and six encounters, respectively. The sequence with 7 encounters presents one PHA and the one with six encounters presents both a PHA and a NHATS. Tables 3 and 4 present the mission parameters of the selected sequences for multiple NEA rendezvous missions. The encountered bodies are specified, together with the departure and arrival date, TOF and ΔV. The missions are completed in 3559 days (i.e., 9.7 years) and 4292 days (i.e., 11.7 years), respectively, with a total ΔV required of $\Delta V = 43.5$ km/s and $\Delta V = 51.95$ km/s. The second trajectory is characterised by a final mission duration larger than 10 years. This is due to the fact that larger TOF computed during the optimisation requires the following departure point to shift to a later date. The projection of both trajectories in the XY-plane are illustrated in Figures 6 and 7, respectively. The stay times at each NEA are highlighted.

Considering the first sequence presented in Table 3 flown by a system with dry mass $m_{dry} = 350$ kg, which is similar to the one of the *Deep Space 1* by NASA [31], with SEP with specific impulse $I_{sp} = 3000$ s, the initial mass m_0 of the spacecraft would be:

$$m_0 = m_{dry} \exp\left(\frac{\Delta V}{g_0 I_{sp}}\right) = 1535 \text{ kg} \qquad (13)$$

with g_0 being the standard acceleration due to gravity. This equates to a mass ratio $\frac{m_{dry}}{m_0} = 0.23$. Comparing this number with the specifications offered in one of the last GTOC problems [32], it can be noticed that a larger mass ratio (minimum of about $\frac{m_{dry}}{m_0} = 0.4$) should be allowed for a system propelled by a low-thrust engine. This can be obtained by redesigning the NEA rendezvous mission; in particular, discarding the last two transfers the total ΔV reduces favoring a more convenient mass ratio.

Differently, solar sailing can be used as propulsion system, which performs best for long mission duration thanks to its propellantless nature. To verify that the NEA sequences can be flown also using solar sailing and estimate the differences, the trajectory described in Table 4 is optimized with solar sailing with $a_c = 0.3$ mm/s^2, to compensate for the fact that the acceleration of the sail diminishes when it is not directed away from the Sun. The table shows the mission parameters of the optimized SS trajectory, where the optimised trajectory obtained with SEP is used as initial guess. The total mission duration is 4406 days (12.1 years), thus requiring only 142 additional days to visit all the six asteroids compared to SEP. The full 3D representation and 2D projection of the first leg of the trajectory are shown in Figure 8, where the SEP and SS transfers are compared. Also, the control histories related to the two propulsion systems are plotted in Figure 9, showing the three components of the required acceleration with respect to the orbital reference frame and the magnitude of the acceleration over time.

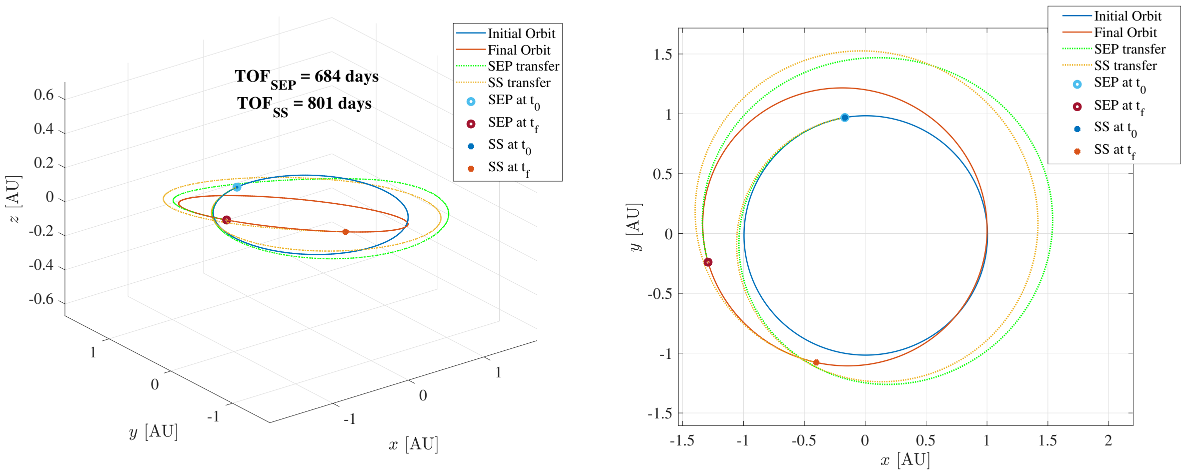
The Tables 3 and 4 also present the results in TOF and ΔV obtained from the optimization to compare them with those computed by the neural network during the sequence search (in brackets in the tables). It can be noticed that the difference in both TOF and ΔV is generally limited, but they can sometimes reach up to 80-100 days in TOF and 1.5 km/s in ΔV. This can be due to the fact that, in the sequence search algorithm, the stay time and, consequently, the departure time ($t_0 = TOF_0 + t_{stay}$) are imposed, with TOF_0 being the mission duration until that point and t_{stay} the stay time. The average percentage error is computed considering only the TOF and ΔV for each transfer, since these are the only two parameters predicted by the network. The two trajectories show that the registered average error is about 15% for TOF and 14% for ΔV, which are calculated as follows:

$$\mathcal{E}_{TOF} = \frac{1}{N}\sum_i^N \left(\frac{TOF_{opt} - TOF_{NN}}{TOF_{opt}}\right) \qquad (14)$$

$$\mathcal{E}_{\Delta V} = \frac{1}{N}\sum_i^N \left(\frac{\Delta V_{opt} - \Delta V_{NN}}{\Delta V_{opt}}\right) \qquad (15)$$

with N being the number of legs in the trajectory. This confirms the validity of the methodology and the capability of the neural network obtained in Section 2 to predict the cost of a trajectory given the departure and arrival orbits.

Comparing the obtained results with previous works [8, 11], some considerations related to the different



(a) Three dimensional representation

(b) Two dimensional representation

Figure 8: Earth to 2011 AM24 transfer using SEP and solar sailing (SS).

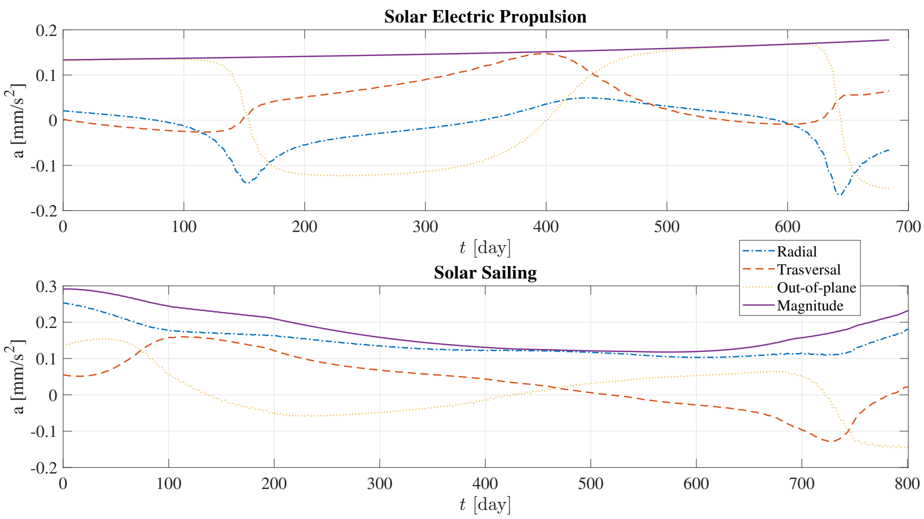


Figure 9: Earth to 2011 AM24 transfer using SEP and solar sailing (SS): acceleration components history.

propulsion system or database used in the analysis need to be taken into account. As mentioned before, the SEP system used can potentially achieve an higher thrust over time for the same maximum acceleration than the solar sailing, due to the reduced acceleration achievable by the sail when not directed away from the Sun. Thus, it is expected that, for a similar NEA database and characteristic acceleration, the SEP system can visit a larger number of encounters in the same period of time. This is highlighted from the comparison with the work of Peloni at al. [8], which found a 10-year mission with five encounters. Differently, Song and Gong [11] were able to identify a mission visiting 15 asteroids with a larger $a_c = 0.75$ mm/s^2 and a NEA database comprising NEAs whose orbital eccentricity and inclination are limited to small values. Also, they could train a DNN for the estimation of the TOF of transfer given initial and final orbits, obtaining a final error lower than 1% with respect to the values computed by the optimizer. The difference is likely due to the different training database used, which in their work was obtained computing the transfer by means of an optimisation solver providing a much finer solution with respect to the shape-based method employed in this study.

6. Conclusions

A new, faster methodology for the preliminary design of a multiple NEA rendezvous mission using low-thrust is investigated. Artificial Neural Network offers a quick estimation of the cost of an interplanetary transfer in terms of ΔV and TOF, which can be used to efficiently identify the optimal sequence of asteroids to visit. The training database is obtained by using the shape-based method to compute the approximated trajectories between NEAs. The parameters of the network are optimised and a genetic algorithm is used to identify the best combination of network parameters to ensure an high performance. A neural network with four hidden layers and 80 neurons in each hidden layer results to be excellent to map the relationship between orbit parameterisation and cost of the transfer. The final neural network, in fact, can obtain an accurate fitting between targets and outputs, confirmed by a correlation coefficient of 0.96. A sequence search algorithm is implemented to identify the most convenient sequences of asteroids. When a neural network is used in the searching algorithm, the computational time is reduced by 25 times with respect to the case when a shape-based method is used. This represents a great advantage in the preliminary design of multiple NEA missions. Finally, the selected sequences are optimised to verify the feasibility of the trajectory with the provided propulsion system.

The database used in the search contains potentially hazardous asteroids and NEO Human Space Flight Accessible Target Study so that it is possible to encounter asteroids which are relevant for scientific or defence purposes. Amongst the sequences obtained, two of them are selected and presented, which are the sequences with a PHA and with both a NHATS and a PHA as encounters. Comparing the resulting ΔV and TOF obtained from the optimisation process and those predicted by the neural network, an average error of 15% is registered. According to these results, using a neural network to predict the cost of a transfer within a sequence search algorithm for the preliminary design of multiple NEA rendezvous mission is effective to reduce the computational time.

Acknowledgements

Giulia Viavattene gratefully acknowledges the support received for this research from the School of Engineering at the University of Glasgow for funding the research under the James Watt sponsorship programme.

References

[1] J. J. Lissauer and Imke de Parter. *Fundamental Plantary Science*. Cambridge University Press, 2013.

[2] J. R. Wertz. *Orbit & Constellation Design & Management*. Springer, New York, 2009.

[3] R. G. Jahn. Physics of Electric Propulsion, 2006.

[4] D. M. Goebel and I. Katz. *Fundamentals of Electric Propulsion: Ion and Hall Thrusters* . Jet Propulsion Laboratory - California Institute of Technology, 2008.

[5] C. R. McInnes. *Solar Sailing - Technology, Dynamics and Mission Applications*. 1999.

[6] A. Mereta and D. Izzo. Target selection for a small low-thrust mission to near-Earth asteroids. *Astrodynamics*, 2(3):249–263, 2018.

[7] A. Peloni, M. Ceriotti, and B. Dachwald. Solar-Sail Trajectory Design for a Multiple Near-Earth-Asteroid Rendezvous Mission. *Journal of Guidance, Control, and Dynamics*, 39(12):2712–2724, sep 2016.

[8] A. Peloni. *Solar-sail Mission Design For Multiple Near-Earth Asteroid Rendezvous*. Phd thesis, University of Glasgow, 2018.

[9] D. Hennes, D. Izzo, and D. Landau. Fast approximators for optimal low-thrust hops between main belt asteroids. *IEEE Symposium Series on Computational Intelligence, SSCI 2016*, 2016.

[10] A. Mereta, D. Izzo, and A. Wittig. Machine learning of optimal low-thrust transfers between near-earth objects. *Lecture Notes in Computer Science*, 10334 LNCS:543–553, 2017.

[11] Yu Song and Shengping Gong. Solar-Sail Trajectory Design of Multiple Near Earth Asteroids Exploration Based on Deep Neural Network. pages 1–31, 2019.

[12] Hao Peng and Xiaoli Bai. Artificial Neural Network–Based Machine Learning Approach to Improve Orbit Prediction Accuracy. *Journal of Spacecraft and Rockets*, 55(5):1–13, 2018.

[13] C. Sanchez-Sanchez, D. Izzo, and D. Hennes. Learning the optimal state-feedback using deep networks. *IEEE Symposium Series on Computational Intelligence, SSCI 2016*, pages 1–8, 2016.

[14] C. Sánchez-Sánchez and D. Izzo. Real-time optimal control via Deep Neural Networks: study on landing problems. *Journal of Guidance, Control, and Dynamics*, 41(3):1122–1135, 2018.

[15] Anastassios E Petropoulos and James M Longuski. Automated Design of Low-Thrust Gravity-Assist Trajectories. *AIAA Paper 2000-4033*, 2000.

[16] Anastassios E Petropoulos and James M Longuski. Shape-Based Algorithm for Automated Design of Low-Thurst, Gravity Assist Trajectories. *AIAA Journal of Spacecraft and Rockets*, 41(5):787–796, 2004.

[17] Bradley J. Wall and Bruce A Conway. Shape-Based Approach to Low-Thrust Rendezvous Trajectory Design. *Journal of Guidance, Control, and Dynamics*, 32(1), 2009.

[18] P. De Pascale and M. Vasile. Preliminary Design of Low-Thrust Multiple Gravity-Assist Trajectories. *AIAA Journal of Spacecraft and Rockets*, 43(5):1065–1076, 2006.

[19] Esposito T. Pinon E. Hur-Diaz S. Mink R. G. Barbee, B.W. and D. R. Adamo. A Comprehensive Ongoing Survey of the Near- Earth Asteroid Population for Human Mission Accessibility. *AIAA Guidance, Navigation, and Control Conference, AIAA Paper 2010-8368*, 2010.

[20] B. W. Barbee, T. Esposito, E. Pinon, S. Hur-Diaz, R. G. Mink, and D. R. Adamo. A Comprehensive Ongoing Survey of the Near-Earth Asteroid Population for Human Mission Accessibility. *AIAA Guidance, Navigation and Control Conference, AIAA Paper 2010-8368*, 2010.

[21] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. The MIT Press, Cambridge, Massachusetts, 2017.

[22] W.E. Wiesel. *Spaceflight Dynamics: Third Edition*. CreateSpace, Cambridge, Massachusetts, 2017.

[23] K F Wakker. *Fundamentals Of Astodynamics*. Faculty of Aerospace Engineering, Delft University of Technology, 2015.

[24] R. Farooq. *Machine Learning for Space Trajectory Design and Optimization*. Msc thesis, University of Glasgow, 2019.

[25] H. Demuth and M. Beale. *Neural Network Toolbox For Use with MATLAB. Version 4*. The MathWorks, Inc., 2002.

[26] J. T. Betts. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. SIAM Press, Philadelphia, second edition, 2010.

[27] Michael A Patterson and Anil V Rao. GPOPS-II manual: A General-Purpose MATLAB Software for Solving Multiple-Phase Optimal Control Problems (Version 2.3). Technical report, University of Florida, Gainesville, FL, USA, 2016.

[28] Michael A. Patterson and A. V. Rao. GPOPS-II: A MATLAB Software for Solving Multiple-Phase Optimal Control Problems usign hp-Adaptive Gaussian Quadrature Collocation Methods and Sparse Nonlinear Programming. *ACM Transactions on Mathematical Software*, 41(1), 2014.

[29] A. V. Rao, David A. Benson, Christopher L. Darby, Michael A. Patterson, Camila Francolin, I. Sanders, and Geoffrey T. Huntington. Algorithm 902: GPOPS, a MATLAB Software for Solving Multiple-Phase Optimal Control Problems using the Gauss Pseudospectral Method. *ACM Transactions on Mathematical Software*, 37(2), 2010.

[30] Divya Garg, Michael A. Patterson, Camila Francolin, Christopher L. Darby, Geoffrey T. Huntington, William W. Hager, and Anil V. Rao. Direct trajectory optimization and costate estimation of finite-horizon and infinite-horizon optimal control problems using a Radau pseudospectral method. *Computational Optimization and Applications*, 49(2):335–358, 2011.

[31] NASA. Deep Space 1 Asteroid Flyby. *Press Kit*, 1999.

[32] L. Casalino and G. Colasurdo. Problem Description for the 7th Global Trajectory Optimisation Competition. *Global Trajectory Optimisation Portal*, 2014.