Artificial Neural Networks for Multi-Target Low-Thrust Missions

Giulia Viavattene

Submitted in fulfilment of the requirements for the Degree of

Doctor of Philosophy
in Aerospace Engineering

James Watt School of Engineering
College of Science and Engineering
University of Glasgow

Student number:
Supervisors: Dr. Matteo Ceriotti
Prof. Colin McInnes

Date: August 22, 2022

Abstract

Multi-target missions are an attractive solution to visit multiple bodies in a single mission, increasing the scientific return and reducing the cost, compared to multiple missions to individual targets. Designing multi-target missions represents a challenging task since it requires multiple options to be estimated, given the large number of objects which can be considered as potential targets. Low-thrust propulsion systems are preferred to rendezvous multiple targets in a mission as they allow to utilise less propellant mass than high-thrust systems to perform the same trajectory. However, low-thrust trajectories are computationally expensive to compute.

This PhD thesis proposes to use artificial neural networks (ANN), as a fast and accurate estimation method for optimal low-thrust transfers. An artificial neural network and a sequence search (SS) algorithm can be designed to find solutions to three kinds of multi-target global optimisation problems: (i) multiple active debris removal missions (MADR), (ii) multiple near-Earth asteroid rendezvous (MNR) missions, with the option of returning a sample to Earth, and (iii) multi-objective optimisation of low-thrust propulsion systems for multi-target missions. MADR missions allows for the disposal of inactive satellites and larger objects, preventing the build-up of space junk and allowing to replace ageing agents in a constellation. Similarly, MNR missions allow to reduce the cost of each NEA observation and increase the possibility of visiting multiple asteroids of interest in a single mission.

The trained ANN is employed within a SS algorithm, based on a tree-search method and breadth-first criterion, to identify multiple rendezvous sequences and select those with lowest time of flight and/or required propellant mass. To compute the full trajectory and control history, the sequences are subsequently recalculated by using an optimal control solver based on a pseudospectral method. Also, to optimise the propulsion system for a given mission, a multi-objective optimisation using a genetic algorithm is performed, where ANNs are employed to quickly estimate the cost and duration of multi-target transfers.

The results show that neural networks can estimate the duration and cost of low-thrust transfers with high accuracy, for all the three applications. Employing machine learning within a sequence search algorithm to preliminary design multi-target missions allows to significantly reduce the computational time required with respect to other most commonly used methods in the literature, while maintaining a high accuracy. Given the combinatorial nature of the problem, the benefits in terms of computational time introduced by the ANN increase exponentially with a linear increase of the number of bodies in the database.

Author's Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Glasgow, Scotland, United Kingdom

August 22, 2022

Giulia Viavattene

Acknowledgements

This PhD thesis is the result of 3.5-year experience as a PhD student at the University of Glasgow. Several people have helped me and guided me during this once-in-a-lifetime (luckily?!) experience and, with these acknowledgements, I would like to thank them all.

Firstly, I would like to express my gratitude to my PhD supervisor, Dr Matteo Ceriotti. Matteo has supported me and guided me throughout the endeavour of this PhD research, fostering my growth in both the professional and personal spheres. He always took the time to celebrate with me all the successes derived from this work and, at the same time, to encourage me when needed.

I would also like to acknowledge the University of Glasgow and, specifically, the School of Engineering to allow me to pursue my PhD degree.

From the moment I started this journey in October 2018, I have always had the opportunity to learn something new every day. I have had the pleasure to share interesting tips, ideas, experiences and fun moments with many of the lectures and fellow PhD colleagues. Thanks to the SET (Space and Exploration Technology) research group for keeping this up also during these challenging years of the Covid-19 pandemic! In particular, I would like to thank Merel and Iain for the continuous support during the highs and lows of this adventure. Of course, Prof Colin McInnes, who I am grateful for welcoming me within the research group and sharing with me insightful and inspiring ideas.

I would like to thank the colleagues at Fujitsu, Astroscale and Amazon Web Services for allowing me to apply the methodology which I developed during this PhD work to a real mission scenario proposed by the industry. I am glad to know that my contribution can be practically used to efficiently design multi-target space missions.

Special thanks go to the European Space Agency (ESA) to sponsor me to attend the 70^{th} International Astronautical Congress (IAC) in Washington, D.C. (USA) and take part in the International Space Education Board (ISEB) student program. This represented a valuable experience to present my work at a technical session of one of the biggest annual congress that gathers all the space professionals. It provided me with the opportunity to create a network for future collaborations, meet and interact with experts from other universities and space agencies, who also became friends. For these reasons, I will carry this experience in my heart forever.

Although we never met in person, Jan Thimo Grundmann never stops to inspire me with his enthusiasm about space research. I am grateful that our paths crossed at

the beginning of my PhD work so that we could collaborate and explore many aspects of NEA exploration.

Last but not least, my parents and my brother who are my strongest support system. I am where I am today and who I am today thanks to you and all you did for me. I love you.

Contents

List of Figures

List of Tables

Nomenclature

The author attempted to use standard symbols and acronyms in use in Astrodynamics and Optimisation, giving alternatives wherever appropriate, and tried to be consistent throughout the document.

The following notation will be used throughout the document. All scalar quantities will be represented by non-bold italic symbols. All vectors will be denoted by bold italic symbols and treated as column vectors. All matrices will be represented by uppercase italic bold symbols.

Acronyms

ACO	Ant Colony Optimisation
ADR	Active Debris Removal
AI	Artificial Intelligence
ANN	Artificial Neural Network
CNEOS	Centre of Near-Earth Object Studies
COE	Classical Orbital Elements
CPU	Central Processing Unit
DE	Differential Evolution
ECI	Earth-Centred Inertial
EMOID	Earth Minimum Orbit Intersection Distance
ENEAS+	Exploration of NEAs with solar Sailcraft
EE	Equinoctial Elements
eH	Eccentricity and Angular Momentum Vector
ELSA	End-of-Life Service by Astroscale
EP	Electric Propulsion
ESA	European Space Agency
GA	Genetic Algorithm
GA-ANN	Genetic Algorithm and Artificial Neural Network Method
GOCE	Gravity Field and Steady-State Ocean Circulation Explorer
GTOC	Global Trajectory Optimisation Competition
IAC	International Astronautical Congress
ISEB	International Space Education Board
JAXA	Japan Aerospace Exploration Agency
JPL	NASA Jet Propulsion Laboratory

LEO	Low-Earth Orbit
LG	Legendre-Gauss points
LGL	Legendre-Gauss-Lobatto points
LGR	Legendre-Gauss-Radau points
LVLH	Local-Vertical-Local-Horizontal
MADR	Multiple Active Debris Removals
MBA	Main-Belt Asteroid
MJD	Modified Julian Day
ML	Machine Learning
MLT	Multi-objective optimisation of Low-Thrust systems
MNR	Multiple NEA Rendezvous
MSE	Mean Squared Error
NASA	National Aeronautics and Space Administration
NEA	Near-Earth Asteroid
NHATS	Near-Earth Object Human Space Flight Accessible Targets Study
NLP	Non-Linear Programming problem
OCC	Orbit Condition Code
OCP	Optimal Control Problem
PHA	Potential Hazardous Asteroid
RAAN	Right Ascension of Ascending Node
ReLU	Rectified Linear Unit
SET	Space and Exploration Technology
SIMONE	Smallsat Intercept Missions to Objects Near Earth
SNOPT	Sparse Nonlinear Optimizer
IPOPT	Interior Point Optimizer
PSM	Pseudospecral Method
PSO	Particle Swarm Optimisation
SPOUA	South Pacific Ocean Uninhabited Area
SS	Sequence Search
SS-ANN	Sequence Search and Artificial Neural Network Method
SSP	Solar Sailing Propulsion
TOF	Time Of Flight
TRL	Technology Readiness Level
TSP	Travelling Salesman Problem

Symbols

Roman Letters

a	Semimajor axis, km
a_c	Sail Characteristic Acceleration, m/s^2

a_d	Geometric Albedo
a_{max}	Maximum Acceleration, m/s^2
A	Appealing Factor
$\mathbf{A}$	Matrix of the Dynamics
A_{ij}	Elements of State Matrix $\mathbf{A}$
$\mathbf{a}$	Acceleration Vector, m/s^2
$\mathbf{a_T}$	Thrust Acceleration Vector, m/s^2
$\mathbf{a_g}$	Acceleration Vector due to Earth oblateness, m/s^2
$\mathbf{a_D}$	Acceleration Vector due to Atmospheric Drag, m/s^2
$\mathbf{b}$	Vector of the Dynamics
b_i	i-th ANN Bias
C_D	Drag Coefficient
C_{MSE}	Mean Squared Error
$C_{MSE,val}$	Mean Squared Error of Validation Set
d_k	Search Direction in Network Weight Space
D	Debris Object
$D1$	Departure Debris Object
$D2$	Arrival Debris Object
$\mathbf{D}$	Radau Pseudospectral Differentiation Matrix
e	Eccentricity
$\mathcal{E}$	Percentage Error, %
$\mathbf{e_v}$	Eccentricity Vector
$\mathbf{e}$	Network Error Vector
f	Generic Function
f	ANN Activation Function
f,g	In-plane Modified Equinoctial Elements
g_e	Gravitational Acceleration at Earth's Surface, m/s^2
$\mathbf{g}$	Gradient of Network Performance Function
$\mathbf{g}$	Path Constraints
$\mathbf{G}$	(Discretised) Path Constraints
G	Universal Gravitational Constant, m^{3}kg^{-1}s^{-2}
h	Altitude, km
h, k	Out-of-plane Modified Equinoctial Elements
H	Absolute Magnitude
$\mathbf{H}$	Angular Momentum Vector, kg m s^{-1}
$\mathbf{H}$	Hessian Matrix of Network Performance Function
i	Inclination, deg
$\mathbf{i_n}$	Local North Direction Vector
I_{sp}	Specific Impulse, s
$\mathbf{I}$	Identity Matrix
I_V	Interest Value
J	Objective Function

J_k	Zonal Harmonic Coefficient of k-th order
$\mathbf{J_e}$	Jacobian Matrix of Network Error Vector
L	True Longitude, deg
L	Integral Cost Function
$L_i(\tau)$	Lagrange Polynomial at Collocation Point τ_i
m	Mass, kg
M	Mean Anomaly, deg
m_0	Initial Mass, kg
m_{dry}	Dry Mass, kg
m_{EP}	Mass of a Thruster, kg
m_{gimbal}	Mass of EP Gimbal System, kg
m_{pl}	Payload Mass, kg
m_{prop}	Propellant Mass, kg
m_{tank}	Mass of the Tanks, kg
N_A	Number of Asteroids in a Sequence
N_{GA}	Number of Pareto Solutions
N_{max}	Optimal Number of Training Samples
N_S	Number of Solutions
$n_{thrusters}$	Number of Thrusters
$\mathbf{N}$	Acceleration Direction and Magnitude Vector
$\mathbf{N_s}$	Sail Normal Unit Vector
p	Semi-latus rectum, km
P_{max}	Maximum EP Power, W
$P_N(\tau)$	N-th Degree Legendre Polynomial
$\mathbf{Q_r}$	Transformation Matrix from LVLH to ECI frame
r	Magnitude of Position Vector, km
R	Correlation Coefficient
r_E	Average Sun-Earth Distance, km
R_e	Equatorial Radius of Earth, km
$\mathbf{r}$	Position Vector, km
S	Aerodynamic Surface Area, m^2
t	Time, s
$t_{0,f}$	Time of Flight, s
$\mathbf{t}$	Network Target Output Vector
T	Transfer Leg Duration, s
T_{max}	Maximum Thrust, N
T_p	RAAN-Phasing Time, s
T_s	Stay Time, s
T_{train}	Network Training Time, s
$\mathbf{u}$	Control Vector
$\mathbf{U}$	Discretised Control Vector
$\mathbf{u_i}$	Direction of Sail Incident Radiation Vector

$\mathbf{u_r}$	Direction of Sail Reflected Radiation Vector
v	Magnitude of Velocity Vector, km/s
$\mathbf{v}$	Velocity Vector, km/s
w	Gaussian Quadrature Weight
w_i	i-th ANN Weight
$\mathbf{w}$	Network Weight and Bias Vector
$\mathbf{x}$	State Vector
$\mathbf{x}$	Network Input Vector
$\mathscr{X}$	Input Set
$x(t)$	State Function
X	Discretised State Function
$\mathbf{y}$	Network Output Vector
$\mathscr{Y}$	Output Set

Greek Letters

α_s	Sail Cone Angle, deg
α	Weight Factor
γ	Weight Factor
δ_{dec}	Decrement of Network Weight Change
δ_{inc}	Increment of Network Weight Change
Δ	Deviation
ΔV	Velocity increment, km/s
$\delta\mathbf{g}$	Gravitational Perturbation Acceleration, km/s^2
δg_n	Local North Component of Gravitational Perturbation Acceleration, km/s^2
η	Network Learning Rate
η_{EP}	Efficiency of the EP System
θ	True anomaly, deg
θ	(half) Eclipse Angle, deg
λ_1,λ_2,λ_3	Shaping Parameters
μ	Gravitational Parameter, m^{3}s^{-2}
μ	Network Gradient Constant
μ_B	Mean of Data Batch B
μ_{dec}	Network Gradient Decrease Factor
ρ	Atmospheric Density, kg/m^3
σ	Standard Deviation
σ_B	Standard Deviation of Data Batch B
Σ	Sequence Reward or Cost
τ	Normalised Independent Variable
ϕ	Geocentric Latitude, deg
Φ	Terminal Cost Function
Φ_N	Network Function

Ψ_i	Lagrange Cost Term at Collocation Point tau_i
ω	Argument of periapsis, deg
Ω	Right Ascension of the Ascending Node, deg
∇	Network Error Gradient

Superscripts

(i)	i-th Pareto Optimal Solution
(k)	k-th Dimension of Network Input Vector
l	l-th Network Layer
$\hat{\square}$	Normalised Value

Subscripts

0	Initial Value
0	Departure Object
1	Departure Debris 1
2	Arrival Debris 2
$2,a$	Transfer from Disposal to RAAN-Phasing Orbit
$2,b$	Transfer from RAAN-Phasing Orbit to Debris 2
a	Arrival
d	Departure
$D1$	Departure Debris Object
$D2$	Arrival Debris Object
e, E	Earth
EP	Electric Propulsion
f	Final Value
f	Arrival Object
$guess$	Guessed Value
h	Out-of-plane Component
j	j-th Network Neuron
jk	Connection between the k-th and j-th Neurons
max	Maximum Value
min	Minimum Value
opt	Optimal Value
p	RAAN-Phasing
$prop$	Propellant
PT	Transfer to RAAN-Phasing Orbit and Phasing
r	Radial Component

s	Sun
SC	Spacecraft
SSP	Solar Sailing Propulsion
T	Target Object
θ	Transverse Component

1 Introduction

From ancient times, human beings have been fascinated by space. Astronomy allowed the study of the millions of stars and neighbouring planets since the early ages and, from the twentieth-century, man started sending out probes and himself in space to explore it. Recently, the interest for multi-target interplanetary missions is emerging, which are space missions with the intent to target more than one celestial body in a single mission. This represents an attractive solution for space exploration because it allows to increase the scientific return with respect to a mission to a single celestial object, while reducing the cost compared to performing a mission with multiple spacecrafts transferring to individual targets [1, 2].

Multi-target missions, which are already being conducted, target mostly small celestial bodies in our solar system [3, 4].[1] Small celestial bodies, such as near-Earth asteroids (NEAs), have been avidly studied and visited during space missions given their significant role in the geological and biological evolution of Earth, the possible exploitation of asteroids' resources, and Earth protection from future collisions [5, 6, 7]. Similarly, missions for multiple space debris removals have attracted vast interest. Considering the rapid growth of the debris population in low-Earth orbit (LEO) in the last two decades (which is expected to continue to grow)[8], removing more than one piece of debris in a mission would represent an efficient strategy of active debris removal [9, 10].

Section 1.1 introduces the concept of multi-target missions, identifying the benefits and the challenges to design such missions, with a particular focus on multiple active debris removal missions and multiple NEA rendezvous missions. While a literature review of the recent missions and studies about multi-target missions is performed in Chapter 2, this chapter aims at identifying the research need which we intend to met through this dissertation. The goals and objectives of the study are defined in Section 1.2, where the research questions are specified. The journal papers

[1] Except multi-gravity-assist missions

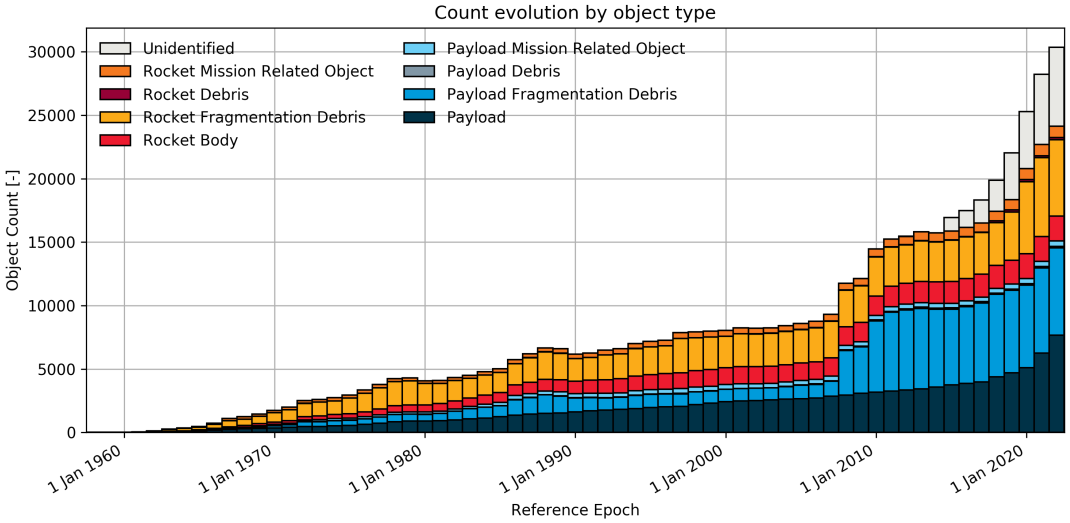


Figure 1.1: Evolution of the number of space debris in LEO. (Credits: ESA ESOC)

and conference papers which have been published and presented as a result of this research work are listed in Section 1.3. Finally, Section 1.4 presents the structure of this thesis.

1.1 Multi-Target Missions

Above 80% of the trackable objects in near-Earth space are space debris [11], which are all the man-made objects including fragments and elements thereof, in LEO or reentering the atmosphere, that are non-functional. Scientific models estimate the total number of space debris objects in LEO to be in the order of 36,000 for sizes larger than 10 cm, 1 million for sizes larger than 1 cm and more than 130 millions for sizes larger than 1 mm [12]. Many debris are too small to be tracked, but large enough to threaten human spaceflight and robotic missions. Both the debris and spacecraft travel at high speeds (approximately 7 km/s in LEO), thus even an impact of a tiny piece of debris can lead to big issues to the spacecraft and undermine the mission success.

This constitutes a potential danger to all space vehicles, including the International Space Station and other spacecraft for manned and unmanned missions, especially considering the rising population of space debris. Figure 1.1 shows the evolution of the number of space debris in LEO by object type, demonstrating how space debris have been exponentially increased in number in the past 40 years [13].

Active debris removal (ADR) is the process to dispose inactive objects from space, preventing the build-up of junk, such as non-functional spacecraft, abandoned launch vehicle stages and other large objects, and to replace faulty agents in satellite constellations [9, 10]. ADR can be particularly useful for larger debris objects that would not de-orbit naturally in a short time frame (for example due to their altitude), or those which may pose a threat to active satellites. Collisions of larger objects can also cause a sudden growth of the debris population due to fragmentation, which could eventually lead to the *Kessler syndrome*. The Kessler syndrome, first investigated by NASA scientist Donald J. Kessler in 1978, is a scenario in which the density of objects in LEO due to space pollution is so high that each collision between space debris generates more debris fragments increasing the likelihood of further collisions [14, 15].

Removing multiple debris objects in a single multiple active debris removal (MADR) mission [16, 17] can be advantageous not only from a financial prospective. Given the urgency of the clean-up of the overly-crowded low-Earth orbits, MADR missions can have a positive impact from a timing prospective, especially in cases where it is necessary to remove many objects within a limited time frame [18, 10].

Since the 1960s, NEAs have also been avidly studied, given the significant role they play in the geological and biological evolution of Earth, the possible exploitation of their resources and Earth protection from future collisions [19, 20]. In fact, the origin of life and biosphere on Earth seems to be associated to a primordial asteroid impact with Earth. Asteroid collisions continue to occur and can represent a hazard for our civilisation. Many mitigation strategies are being investigated for which asteroid detection by ground or space surveys are essential. Additionally, NEAs offer precious resources (such as water, volatiles, semiconductors and metals) which could be extracted and used to support future space activities. The presence of gold and platinum also contributes to make NEA exploration extremely valuable.

NEAs vary greatly in size, shape and composition, thus rendezvous and close-up observations are necessary to classify these objects, improve our knowledge of their diversity and support any future asteroid mission and mitigation action. According to the NASA's database [21], more than 28,000 NEAs have been discovered until now. Figure 1.2 shows the cumulative number of known NEAs over time, of which almost 10,000 NEAs are larger than 140 m in size and almost 1,000 are larger than 1 km in size. Among those objects, more than 2,200 are characterised by an Earth Minimum Orbit Intersection Distance (EMOID) lower than 0.05 AU and an estimated diameter greater than 150 m, which are classified as Potential Hazardous Asteroids (PHAs).

To reduce the cost for each observation and increase the possibility of visiting multiple asteroids of interest in a single mission, multiple NEA rendezvous (MNR) missions are becoming an attractive solution [2, 5, 3, 4]. The scientific community gives no clear priorities in the selection of NEAs to design MNR missions because something can be learned and gained from rendezvousing any asteroid. Consequently, any NEA can represent a potential candidate within a sequence of objects to be visited. It follows that, as an example, trillions of permutations between these

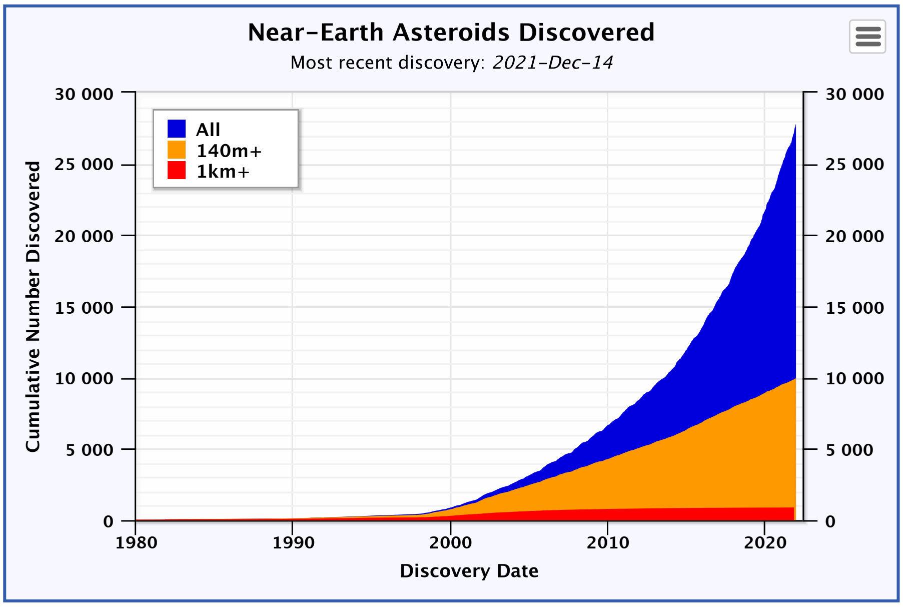


Figure 1.2: Number of discovered NEAs over time. (Credits: S. Khudikyan, A. Chamberlin, *Jet Propulsion Laboratory NASA)*

objects need to be investigated to identify sequences of three objects. It should be noticed that the number of permutations grows exponentially with a linear increase of the number of objects or of the number of objects in a sequence [22].

Additionally, the scientific community is interested in collecting samples from the surface of the asteroids, because it is expected that this will allow to learn more about the initial stages of the solar system formation. Consequently, the scope of MNR missions can be expanded to (i) include a sample return to Earth; (ii) select *interesting* asteroids (where the definition of interesting is dependent on the mission objective); and (iii) target specific asteroids within the sequence, for which a sample return to Earth is more valuable.

Multi-target missions to small objects represent one of the biggest challenges for space engineering [23]. These missions are highly demanding in terms of energy, with a velocity increment ΔV (required to transfer from the departure object to the arrival object) which can greatly outrun that of single-object missions. According to the rocket equation, a linear increase in velocity increment ΔV corresponds to an exponential increase in the propellant mass required to complete the mission [24]. For this reason, this study also aims at selecting an efficient propulsion system to keep the propellant mass ratio low. Low-thrust technologies, such as electric propulsion systems, are good candidates because of their high specific impulse, i.e. they require less propellant to deliver a given velocity increment ΔV with respect to high-thrust chemical systems [25, 26]. Similarly, solar sails can be considered as an option where the solar radiation pressure is utilised to propel the satellite without using propellant [27, 28, 29].

These low-thrust systems enable the spacecraft to have a small, but continuous and highly efficient thrust for a long time [30]. Since no closed-form, analytical solution exists for continuous low-thrust propelled spacecraft, computing a low-thrust trajectory requires the solution of an optimal control problem [31]. Thus, to design multi-target missions with a low-thrust propulsion system, a complex global optimisation problem needs to be solved, which consists of two coupled sub-problems [32]. The first is a large combinatorial sub-problem, where the sequence of objects is selected [23]. The second sub-problem calculates the solution to the optimal control problem to identify the optimal low-thrust trajectories with minimum propellant expenditure and/or time of flight (TOF) for each pair of objects. Since almost 28,000 NEAs are known to date and more than 36,000 space debris objects (for sizes larger than 10 cm) are in LEO, solving the complex global optimisation problem for multi-target missions involves an exceptional level of complexity and computational effort.

It becomes paramount to develop a methodology for a quick estimation of the transfer time and cost of low-thrust transfers. These estimations can be used to approximate the solution of the second, continuous sub-problem so that the first combinatorial sub-problem can be solved in a quicker manner with respect to previously used methodologies.

Artificial intelligence can be applied to solve complex problems. In the aerospace sector, it is used successfully in identifying low-thrust trajectories with minimum fuel consumption between main belt asteroids [33], or defining the initial guess required for the optimisation of low-thrust transfers [34], as well as improving orbit prediction accuracy [35]. A method based on an evolutionary algorithm and artificial neural networks (ANNs) was employed to determine trajectories to a single NEA using solar sailing [36]. This method was demonstrated to be more efficient in finding solutions than the traditional optimal control methods.

The application of artificial intelligence for the trajectory optimisation is investigated. Particularly, the ability of ANNs to quickly estimate the key parameters (such as cost and duration) of low-thrust transfers between pairs of objects is analysed. This would lead to develop an efficient methodology to select optimal sequences of targets. A well-trained network can potentially reduce the computational time and effort with respect to the commonly-used optimisation techniques, which is especially important given the large number of objects to permute to define sequences of encounters for both MADR and MNR missions.

There are multiple challenges in employing an ANN for the design of multi-target problems. Firstly, the selection of network inputs may affect the complex trajectory optimisation problem; thus, the use of different parametrizations of the orbit as ANN inputs needs to be investigated. Also, the phasing of the asteroids, which is essential for rendezvous and fly-by transfers, should be accounted for in these inputs. Secondly, defining the topology and the hyper-parameters of the network for the application of multi-target missions is not straightforward and requires the definition and solution of an optimisation problem. Also, it is paramount to identify the most appropriate method to generate the training database, containing a sufficiently large number of samples, and the strategy to train the network. Finally, the integration of the ANN with the algorithm which is implemented to solve the combinatorial part of the global optimisation problem to design multi-target missions needs to be considered.

As mentioned, multi-target missions are long and highly demanding in terms of energy, thus it is paramount to select the most convenient propulsion system so that the initial mass (and propellant mass) and/or the duration of the mission are minimised. The use of an ANN can eventually be investigated, not only to identify the most convenient sequence of objects for MADR and MNR missions, but also to select the most adequate propulsion system for these missions.

1.2 Objectives

Given the great interest in multi-target missions, such as space debris removals and NEA exploration, resources have been invested to identify a methodology that allows to identify the optimal sequences of objects to remove (for MADR missions) or visit (for MNR missions). Given the extensive number of objects to consider during the design of this kind of missions, it is paramount to define a technique which can quickly calculate the cost of low-thrust transfers between pairs of objects and reduce the computational time and effort required to obtain an accurate solution.

The main research question has been formulated as:

> *Can artificial neural networks be used to efficiently design multi-target space missions?*

The main research question holds several research opportunities. Those can be presented by the following research sub-questions, which represents the aims or objectives of this PhD research and can be grouped in the following categories:

Artificial neural networks

1. *Can an ANN provide an accurate, quick estimation of the cost and duration of a low-thrust transfer?*
2. *Can the ANN be integrated within a sequence search algorithm to identify the most convenient sequence of objects for multi-target missions in terms of cost and/or duration?*

Firstly, the capability of an ANN to estimate the cost and TOF of low-thrust transfers between pairs of objects is investigated. In particular, the inputs and outputs of the network shall be defined for multi-target missions and, similarly, the network architecture and hyper-parameters shall be optimised for this application to eventually improve the performance. Finally, the ANN (so trained to solve the continuous part of the global optimisation problem) is integrated with the sequence search (SS) algorithm implemented to solve the combinatorial part of the problem. The efficiency of this methodology (SS-ANN), where the SS algorithm and ANN are used together, can be evaluated.

Multiple active debris removal missions

3. *Can ANN be employed to design MADR missions and, if so, how accurately and what are the benefits in terms of computational time?*

The SS-ANN methodology is used to design MADR missions. The SS-ANN platform shall be able to compute the most convenient sequences of multiple space debris to dispose of in terms of overall duration of the mission. The accuracy of the solutions identified by the methodology can be assessed by calculating the error between the values outputted by the network and the target values of the propellant mass and time of flight. It is expected that employing machine learning techniques can greatly reduce the computational time to design MADR missions compared to previously used methods. This shall be verified by comparing the performance of the SS-ANN platform with commonly-used methods which are currently employed by the industry or in the literature.

Multiple NEA rendezvous missions

4. *Can ANN be employed to design MNR missions and, if so, how accurately and what are the benefits in terms of computational time?*

The use of the SS-ANN platform for the preliminary design of MNR missions using low-thrust propulsion systems is investigated. It shall be verified whether an ANN can enable a quick estimation of the cost and TOF of low-thrust transfers between NEAs, thus allowing for a fast analysis of thousands of transfer options. The resulting sequences of asteroids for MNR missions which allow to visit more NEAs in less time are optimised to verify the feasibility of the trajectories with the provided propulsion system. The computational time and the accuracy of the proposed SS-ANN methodology for MNR missions is compared to other methodologies, which are most-commonly used in this context and do not involve an ANN. Also, considering the interest of the scientific community in collecting samples from the surface of the asteroids and returning them to Earth for further analysis, the scope of the SS-ANN platform shall be expanded to include the possibility of sample return to Earth and targeting more interesting NEAs within a sequence.

Multi-objective optimisation of low-thrust system (MLT) analysis

5. *Can the ANN be used to optimise the propulsion system according to the goals and requirements of a given multi-target mission?*

The use of an ANN to analyse the effect of different characteristics of low-thrust systems on multi-target missions shall be studied. This is paramount especially for high-ΔV missions, such as multi-target missions. In fact, the choice of the on-board propulsion system may have a major impact on the cost (for example, in terms of propellant mass required, which would consequently affect the launch cost) and duration of the mission. The ANN shall be trained to estimate the cost and duration of transfers when using different low-thrust system. Also, the network can be integrated with an optimisation technique to select the propulsion system which can optimise the return of a mission in terms of, for example, reducing its cost or total duration.

1.3 Publications and Conferences

The content of this thesis has previously appeared, or will appear, in the following publications and conferences.

JOURNAL ARTICLES

1. Viavattene, G., Enric Grustan-Gutierrez and Ceriotti, M. [2022]. “Multi-Objective Optimization of Low-Thrust Propulsion Systems for Multi-Target Missions using ANNs”, Advances in Space Research. (in press)

2. Viavattene, G., Snelling, D., Devereux, E., Payne, N., Wokes, S., and Ceriotti, M. [2022]. “Design of Multiple Space Debris Removal Missions using Machine Learning”, Acta Astronautica, Volume 193, 277–286, https://doi.org/10.1016/j.actaastro.2021.12.051

3. Viavattene, G. and Ceriotti, M. [2021]. “Low-Thrust, Multiple NEA Mission Design with Sample Return to Earth using Machine Learning”, AIAA Journal of Spacecraft and Rockets. (under review)

4. Viavattene, G. and Ceriotti, M. [2022]. “Artificial Neural Networks for Multiple NEA Rendezvous Missions with Continuous Thrust”, AIAA Journal of Spacecraft and Rockets, Vol. 59, No. 2, Pages 574-586, http://eprints.gla.ac.uk/249958/

5. Viavattene, G. and Ceriotti, M. [2020]. “Artificial Neural Networks for Tours of Multiple Asteroids”, Springer, Volume 12344 of the Lecture Notes in Computer Science series, https://link.springer.com/chapter/10.1007%2F978-3-030-61705-9_63

6. Ceriotti, M., Viavattene, G., at al. [2020]. "Sailing at the Brink – The No-Limits of Near-/Now-Term-Technology Solar Sails ans SEP Spacecraft in (Multiple) NEO Rendezvous", ASR Advances in Space Research, Elsevier, https://doi.org/10.1016/j.asr.2020.10.017

CONFERENCE PROCEEDINGS

1. Snelling, D., Devereux, E., Payne, N., Nuckley, M., Viavattene, G., Ceriotti, M., Wokes, S., Di Mauro, G. and Brettle, H. [2021]. "Innovation in Planning Space Debris Removal Missions using Artificial Intelligence and Quantum-Inspired Computing", 8th European Conference on Space Debris, ESA/ESOC Germany (virtual due to Covid-19), http://eprints.gla.ac.uk/239211/
2. Grundmann, J.T., Viavattene, G., at al. [2021]. "More bucks for the bang: New space solutions, impact tourism and one unique science & engineering opportunity at t-6 months and counting", 7th IAA Planetary Defense Conference (PDC 2021) (virtual due to Covid-19), http://eprints.gla.ac.uk/244071/
3. Grundmann, J.T., Viavattene, G., at al. [2021]. "How we beat 2019 PDC to NYC by 2 years, within 2 years, 2 years ago", 7th IAA Planetary Defense Conference (PDC 2021) (virtual due to Covid-19). Link to the paper: ...
4. Viavattene, G. and Ceriotti, M. [2020]. "Artificial Neural Networks for Tours of Multiple Asteroids", 15th Hybrid Artificial Intelligent Systems (HAIS), Gijon Spain (virtual due to Covid-19), https://link.springer.com/chapter/10.1007%2F978-3-030-61705-9_63
5. Viavattene, G. and Ceriotti, M. [2020]. "Low-Thrust, Multiple NEA Mission Design with Sample Return to Earth using Machine Learning", AAS/AIAA Astrodynamics Specialist Conference, California, USA (virtual due to Covid-19), http://eprints.gla.ac.uk/222201/
6. Viavattene, G. and Ceriotti, M. [2019]. "Artificial Neural Network for Preliminary Multiple NEA Rendezvous Missions Using Low Thrust", 70th International Astronautical Congress (IAC), Washington DC, USA, http://eprints.gla.ac.uk/202035/
7. Grundmann, J.T., Viavattene, G., at al. [2019]. "Flights are ten a sail – Re-use and commonality in the design and system engineering of small spacecraft solar sail missions with modular hardware for responsive and adaptive exploration", 70th International Astronautical Congress (IAC), Washington DC, USA, http://eprints.gla.ac.uk/202033/
8. Grundmann, J.T., Viavattene, G., at al. [2019]. "Responsive integrated small spacecraft solar sail and payload design concepts and missions", 5th International Symposium on Solar Sailing (ISSS 2019), Aachen, Germany, http://eprints.gla.ac.uk/191892/

9. Grundmann, J.T., Viavattene, G., at al. [2019]. "Responsive exploration and asteroid characterization through integrated solar sail and lander development using small spacecraft technologies", 6th IAA Planetary Defense Conference, Washington DC, USA, http://eprints.gla.ac.uk/193783/

WORKSHOP PRESENTATIONS

1. Viavattene, G., Ceriotti, M., Devereux, E., Payne, N. and Nuckley, M. [7 July 2021]. "Industry University Collaboration Workshop Fujitsu/University of Glasgow Space Debris Project", Glasgow (virtual due to Covid-19).
2. Viavattene, G. [27 May 2020]. "Applied Machine Learning: Asteroid Rendezvous Missions", Machine Learning in Science Colloquium, Glasgow (virtual due to Covid-19).

1.4 Thesis Outline

Following the logical flow of the applied methodology, this thesis reports the work which has been performed to answer the research questions identified above. The thesis is structured in nine chapters. It commences with the present introduction.

Chapter 2 gives an overview of the previous missions and research studies which have been conducted in the sphere of space debris removal, asteroid exploration, low-thrust propulsion and artificial intelligence. These represent the necessary building blocks for the design of multi-target missions. The methods which are used throughout this thesis are presented in Chapter 3. Specifically, the orbital dynamics models, the optimal trajectory optimisation methods, the artificial neural networks and the sequence search logic are explained.

Chapter 4 analyses how ANNs can be designed and trained to approximate the complex non-linear relationship between (a) the characteristics describing the departure and arrival points of a low-thrust transfer and (b) the transfer cost and duration. Different parameterisations of the orbit are studied as network inputs and the network hyper-parameters are optimised so that an accurate performance can be obtained. Once the appropriate architecture of the network is identified, the training database needs to be generated for the multi-target missions. Part of the contents of this chapter have been published in *Springer, Volume 12344 of the Lecture Notes in Computer Science series* [37].

The performance of the network is analysed for both MADR (Chapter 5) and MNR missions (Chapter 6). The trained networks can be integrated within a sequence search algorithm based on a tree-search method and breadth-first criterion, to compute the most convenient sequence of objects. From this, the MADR and MNR sequences which most satisfy the mission objectives can be selected and optimised to fully compute the trajectory and control history. The contents of Chapter 5 have been published in *Acta Astronautica* [38]. The contents of Chapter 6 have been published in the *Journal of Spacecraft and Rockets* [39].

Additionally, Chapter 7 explores the possibility of using the SS-ANN methodology to perform sample returns to Earth, select more interesting NEAs to visit and, also, target specific NEAs within a sequence.

Chapter 8 investigates the ANN performance to analyse how different low-thrust propulsion systems can affect a multi-target mission. Ultimately, this can provide a method for a fast optimisation of the propulsion system so that the objectives of the mission can be achieved, in terms of total duration of the mission and initial mass required. Since two objective functions are considered, the ANN is integrated within a multi-objective optimisation solver to identify the optimal propulsion system for any multi-target mission. The candidate sequences which minimise the objective functions according to the mission goals can be selected and further refined through optimal control problem solvers.

Finally, Chapter 9 addresses the research questions by summarising the work performed and the research outcomes discussed in the previous chapters, and proposes ideas for future research developments.

2

Heritage

Recent missions and studies have been conducted about multi-target missions and, specifically, multiple active debris removals and multiple NEA rendezvouses using low-thrust propulsion systems. The review of the literature shows how experts have tried to tackle this complex global optimisation problem. Artificial intelligence and neural networks have been used in many fields and have been proven to solve difficult challenges by providing quick and accurate solutions, when designed and trained appropriately. From the analysis conducted in this chapter, it can be noted that the use of ANNs to design multi-target missions remains fairly unexplored, which leads us to understand that this remains an open field where, consequently, a deeper analysis could represent a great contribution to the scientific community. An overview of the past and current space debris removal missions and NEA missions is provided in Sections 2.1 and 2.2, respectively. In Section 2.3, a review of the advantages of using a low-thrust propulsion system and of the previous missions which benefit from using it is presented. Finally, an introduction on artificial intelligence and, in particular, ANNs is given in Section 2.4.

2.1 Space Debris Removal: an Overview of Past and Current Missions

The last two decades have seen the build-up of junk in LEO, such as non-functional spacecraft, abandoned launch vehicle stages and other large objects [9, 10, 40]. Missions with ADR purposes are becoming more popular to dispose of many of these debris pieces and are necessary to stabilise the growth of space debris. The European Space Agency (ESA) recommends the following principles for the selection of removal targets to make ADR more efficient in terms of the number of collisions prevented versus objects removed [8]:

- objects with a high mass are preferable as they have the largest environmental impact in case of collision;
- objects with high collision probabilities have priorities in ADR activities, e.g. they are in densely populated regions and have a large cross-sectional area;
- objects in high altitudes, where their orbital lifetime is long, should be targeted first.

The most densely populated region in LEO is around 800–1000 km of altitude. The debris population in LEO with inclination between 60 deg and 95 deg is considered as the most critical, which is a direct result of the high on-orbit population in that inclination range [41, 18]. The orbital regions which are considered as hotspots for collisions are analysed and monitored using long-term environment simulations.

The earliest suspected loss due to a debris collision was of Kosmos 1275, which was part of a 6-satellite Soviet military navigation system and disappeared in July 1981. Many other impacts have been confirmed since. For example, in 1996 the French microsatellite Cerise collided with fragments of an Ariane-1 upper-stage booster [42]; in 2009 and 2010, respectively, NASA satellite Terra suffered of a battery failure anomaly [43] and NASA satellite Aura lost power from half of one solar panel [44], which are both attributed to a debris strike. The first major collision occurred in 2009 between two Russian satellites, the operational Iridium 33 (560 kg) and the debris Kosmos 2251 (950 kg). They collided at a relative speed of about 11.7 km/s, destroying both satellites and creating thousands of pieces of new smaller debris [45].

Many of the Space Shuttle missions were affected by debris impacts. In 1983, Space Shuttle Challenger's front window was damaged by a debris strike and, in 1994, Endeavour's front window was pitted about half its depth [46]. More recently, in 2006 and 2007, respectively, a small hole through the radiator panels was bored in Space Shuttle Atlantis and Endeavour [47]. Although spacecraft are protected by Whipple shielding to protect the interior from minor debris impacts, exterior portions (such as the solar panels) cannot be protected easily [48]. This is the case for the International Space Station, which has registered over 1,400 meteoroid and debris impacts by 2019 [49].

Space debris also represent hazards on Earth. In fact, although most debris burns up in the atmosphere during their re-entry flight, larger debris can reach the ground. According to NASA's catalogues, many pieces of debris have fallen back to Earth in the past fifty years, but no significant damage has been registered to date. For example, the most recent notable one occurred in March 2021, when a Falcon 9 second stage made an uncontrolled re-entry and landed on a farm field over Washington state [50].

It follows that space debris is an urgent problem, for which ADR is a solution that is currently being investigated and demonstrator missions are being designed and flown. The RemoveDEBRIS mission, led by the University of Surrey, is a satellite research project to demonstrate various space debris removal technologies for ADR (e.g. net and harpoon capture) [51]. The ADR demonstration mission ELSA-d was launched by Astroscale in April 2021 to test the magnetic capture mechanism which they have developed, and it is currently in orbit [52]. Also, ESA with its CleanSpace initiative is looking at the required technology developments to capture debris. The ESA ClearSpace-1 mission aims at de-orbiting a Vega upper stage by 2025 [53].

Interest is also emerging for the removal of multiple debris objects in a single ADR mission [16, 17]. This can provide significant advantages not only from a financial prospective, but also from a timing prospective, especially in cases where it is necessary to remove multiple objects within a limited time frame [18, 10]. To stabilise the debris population in orbit, especially in LEO, 5 to 10 objects will need to be removed every year [54, 55]. For this to be a realistic goal, the design and implementation of multi-debris removal missions are a necessity, not an option.

Multi-debris missions can be broadly divided into two categories: (i) a servicing spacecraft travelling from one piece of debris to the next one, deploying smaller thruster modules which attach to and de-orbit each rendezvoused debris [56]; (ii) a chaser rendezvousing and de-orbiting a piece of debris, then transferring to the next debris object to de-orbit in a sequential fashion [57, 58]. It is worth noting that, when the debris object is released in a low-altitude disposal orbit, the debris will re-enter by spiralling down due to atmospheric drag [59]. Re-entries of large debris objects shall be controlled and aiming at uninhabited areas, such as the South Pacific Ocean Uninhabited Area (SPOUA).

The first category of the servicing spacecraft has been the most studied so far for MADR [16, 60], however, this approach is not well suited when capture methods requiring a tethered connection between the chaser and the debris are used (Figure 2.1). The reason behind it is that, in this case, it is inconvenient to stabilise the debris to attach the de-orbiting thrusting device. Capture methods which utilise a tethered connection, such as tethered-net and harpoon, are among the most researched and tested methods for ADR because they are able to make the debris controllable and stable [61]. For example, the RemoveDEBRIS mission [51] have successfully carried out experiments to demonstrate the capturing of an object using a net in space.

The 9^{th} Global Trajectory Optimisation Competition (GTOC) [62], named “The Kessler Run”, deals with MADR problems. It is imagined that the Kessler syndrome is triggered by an explosion in 2060, compromising the Sun-synchronous orbital environment. The competition required scientists to remove a set of 123 orbiting debris pieces that, if removed, would allow to restore the orbital environment functionalities. The winning team was able to remove of 123 pieces of debris from the Sun-synchronous LEO environment using a 10-spacecraft campaign. Each spacecraft, flying in succession over an 8-year period, rendezvouses with a series of the debris objects to deliver a de-orbit package at each debris before moving on to the next object by means of impulsive manoeuvres [63]. To easily compute the chains of objects, a simple transfer model was developed to estimate the ΔV to match the semi-major axis

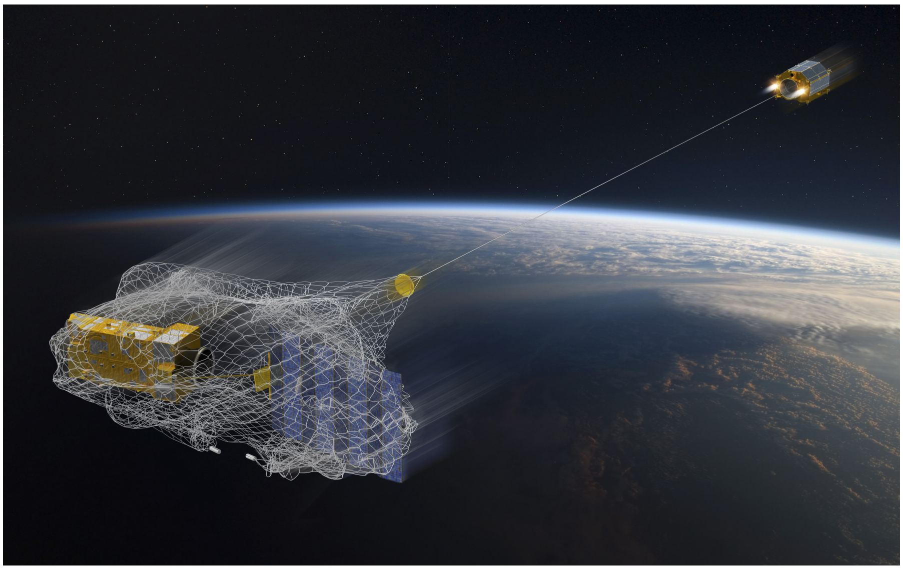


Figure 2.1: Example of a capture of a debris object conducted by using nets. (Credits: ESA)

and inclination of the target debris in a specified transfer time. Branch-and-bound searches are performed to preliminary identify the chains of debris, which are then adjusted using a genetic algorithm where the problem is formulated similarly to the time-dependent travelling salesman problem. Lastly, a final non-linear programming optimisation is performed to ensure the constraints are met and the trajectories are locally optimal.

Similar approaches have been used in the literature to tackle the MADR problem. In essence, because of the number of debris objects to consider in the analysis (i.e. 36,000 for sizes larger than 10 cm, reaching several millions of smaller sizes), a simplified model is generally used to estimate the cost and TOF of each leg to compute suitable sequences of debris objects. Then, a refined optimisation can be performed at a later stage to calculate the full trajectory and control history.

Purely algebraic calculations have been demonstrated to provide satisfactorily accurate estimations of transfer costs and times for space debris removals [64]. For instance, a drift-orbit-transfer strategy with two-impulse Hohmann transfers is used in Ref. [32], where the natural procession of the J_2 zonal term is exploited to match the targeted right ascension of ascending node (RAAN), reducing significantly the total ΔV required. Differently, a pruned database of objects can be used to reduce the amount of calculations required to identify the sequences. For example, a priority list of targets of MADR missions can be defined to enhance the mission efficiency and reduce the computational effort [16].

Total mission time and propellant mass required can be used as key parameters to evaluate the mission performance and select the most convenient sequences of objects. To this end, a travelling salesman problem can be formulated and an optimisation technique can be employed so that the removal sequences can be optimised [64]. Otherwise, a combinatorial tree search can be used to calculate the sequences, which allows for a systematic search amongst all the permutations of objects.

To select the ordering of debris to be de-orbited in a sequence, each debris can be assigned a reward value reflecting the total amount of ΔV, the duration of the debris removal transfers and/or potential collision risk reductions if the debris is de-orbited. The mission planning aims at selecting the debris sequence so that the total reward due to the debris removal is maximised. The selection of space debris to remove can also account for the size of the debris, the availability of useful orbital slots, and the type of capture mechanism [32, 57, 56].

Some studies have been conducted to prove that, when a low-thrust propulsion system is used, the required fuel mass is significantly reduced when compared to chemical propulsion systems, and that the mission duration strongly depends on the mass of the selected targets. The most efficient type of propulsion system is thus determined by the defined mission goals, however low-thrust systems generally result to be superior as it can deliver the same ΔV using less propellant mass [16, 57].

In general, as it will be detailed in Section 2.3, low-thrust transfers have no analytical closed-form solution. Thus, a numerical optimisation strategy must be used to find a solution to the trajectory design problems, which are generally computationally demanding and time consuming. This numerical optimisation is necessary to resolve the continuous part of the global optimisation problem, which is in turn necessary to solve the combinatorial part of the problem for the selection the NEA sequences.

It can be concluded that the design of MADR missions involves the following challenges. First, the selection of the debris objects to be disposed of, and their sequence, shall be identified so that the overall trajectory cost (or duration) is minimised. Second, the orbital transfers to remove a debris shall be designed so that the required propellant mass (m_{prop}) and/or TOF are minimised. It should be stressed that two problems cannot be solved independently of each other, because the first one, which is combinatorial, requires various inputs such as duration and cost of each transfer, which are obtained by solving the second problem, and vice-versa.

The complexity of the problem is also linked to the enormous number of debris currently in orbit, which continues to increase. Given that each piece of debris can represent a suitable candidate for removal within the sequence, it follows that tens of trillions of permutations should be investigated to identify a suitable MADR sequence of three objects, as an example. The number of permutations exponentially rises for a linear increase of the number of objects or when more objects in a sequence are considered. This makes the problem extremely complex and computationally expensive to solve.

A method which provides a fast optimisation of ADR missions for multiple debris removal is necessary, when numerous candidate objects are available. Since the process of trajectory optimisation, specifically for low-thrust transfers, is notoriously computationally demanding, we propose to use machine learning, which may allow for a significant reduction of the computational time. Specifically, ANN can be used to quickly estimate the cost and duration of low-thrust trajectories, given the debris orbits, so that the most effective sequences of debris to be disposed of can be identified. The ANN can be integrated within a sequence search algorithm which computes the feasible debris sequences. The candidate sequences which minimise the objective function (e.g. m_{prop} and/or TOF) can be selected and further analysed.

2.2 Asteroid Exploration: an Overview of Past and Current Missions

Asteroids are small fascinating worlds. Since the 1960s, scientists have dedicated a great effort to the study of NEAs, given the significant role they played in the geological and biological evolution of Earth. NEAs are a subset of the near-Earth object class, which includes the asteroids and comets with a perihelion distance lower than 1.3 AU.

Similarly to fossils for paleontologists, asteroids represent for us a veritable time machine and offer us the opportunity to improve our knowledge on the earliest nature of our planet and, in general, of our solar system. Also, the origin and evolution of life on Earth seem to be linked to asteroid impacts on the primordial Earth, which may have delivered the basic components for life and may have contributed to the formation of our biosphere the way it is today [65].

Asteroid collisions represent a singular hazard to our civilisation [20]. For example, the prehistoric Chicxulub impact in Mexico, 66 million years ago, caused an extinction event. Throughout recorded history, hundreds of Earth impacts have been reported, with some of them causing dramatic consequences. Two of the best-known recorded events in modern times are the 1908 Tunguska event and the 2013 Chelyabinsk meteor event, which occurred in Russia [66]. NEAs which orbit in close proximity to Earth are classified as PHAs. Many mitigation strategies are being studied based on asteroid detection by ground or space surveys, for which knowing asteroids' physical properties is paramount.

A more positive motivation for asteroid exploration is related to the precious resources which these objects offer, such as water, volatiles, semiconductors and metals. Those materials can support future space activities, e.g. hydrogen and oxygen can be used as propellant or life-support; differently, iron, nickel, titanium, manganese and cobalt can be use to build space infrastructures [67]. The presence of gold and platinum also contributes to make NEAs and NEA exploration extremely valuable.

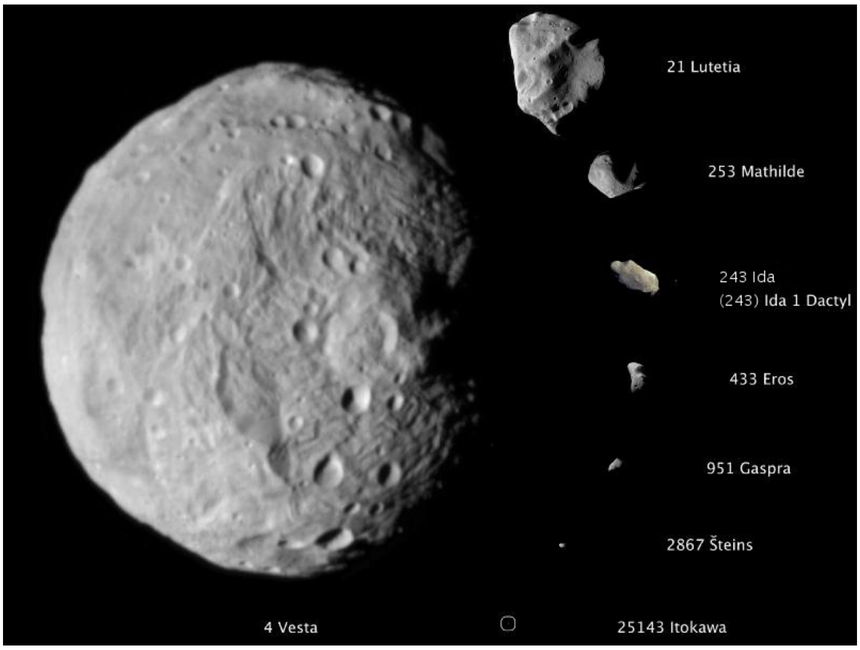


Figure 2.2: A composite image to the same scale of the asteroids, from largest to smallest: 4 Vesta, 21 Lutetia, 253 Mathilde, 243 Ida and its moon Dactyl, 433 Eros, 951 Gaspra, 2867 Šteins, 25143 Itokawa. (Credits: NASA)

Asteroids are characterised by very irregular sizes, shapes, densities, compositions and magnetic fields. As an example, Figure 2.2 shows the differences in size and shape of some well-known asteroids [68]. Given the great variety of both dynamical and physical properties, it suggests that asteroids formed in different places, times and conditions within the solar nebula. This is the reason why close-up observations of these objects constitute an exciting challenge and are necessary to improve our knowledge of their diversity and to support any future asteroid mission. To this end, performing NEA rendezvouses is one of the most important objectives in space exploration. Similarly, sample return to Earth from these bodies may ultimately be needed to provide us with more accurate answers [69].

To reduce the cost of each observation and increase the possibility of visiting multiple asteroids of interest in a single mission, multiple NEA rendezvous missions are preferred [2, 5, 3, 4]. As mentioned, MNR missions are highly demanding in terms of energy and ΔV necessary to fly the mission. For this reason, efficient propulsion systems such as low-thrust systems are the most beneficial choice so that the required ΔV can be delivered using less propellant (with respect to chemical thrusters).

Similarly to the MADR problem, since low-thrust transfers have no analytical closed-form solution, a numerical optimisation is necessary to compute transfers between NEAs. These are in turn necessary to solve the combinatorial part of the global optimisation problem for the selection the NEA sequences. Given the enormous number of permutations (i.e. of calculations) required, this makes the problem very computationally expensive and complex to solve.

Different methodologies have been presented to solve this problem. However, most of them propose to use a simplified model to compute the low-thrust transfers between pairs of NEAs so that the computational time and effort can be reduced while all the asteroid permutations are considered to identify the NEA sequence. Successively, the solutions found can be converted into feasible low-thrust trajectories by solving the full optimal control problem.

Given the large number of asteroids discovered until now and of permutations of objects required, designing such missions represents a significant challenge for the scientific community. Seven out of eleven GTOCs [62] deal with asteroid-related problems, of which six of them examine multiple NEA missions. For instance, the fourth GTOC competition [70] required to search for the largest number of NEAs which can be visited within a given mission duration using a low-thrust propulsion system. A submitted solution proposed to define the sequences by means of impulsive thrusts (as a simplified model) before performing the final low-thrust optimisation of the identified sequences.

Differently, Peloni et al. [2] used a shape-based approach to approximate the low-thrust trajectories and determine the sequences through a search-and-prune algorithm. This approach allowed them to design a mission with five NEAs as encounters within ten years using a near-term solar-sail technology [2]. A local pruning, based on heuristic astrodynamics rules, was needed to find as many sequences as possible within an acceptable amount of computational time. It was demonstrated that the shape-based trajectory model for solar sailing provides a good approximation of the trajectory. Finally, the sequences are optimised to obtain the full solar-sail trajectories and control history.

A homotopic approach is used by Tang et al. [6], to quickly approximate the low-thrust transfers. Similarly, Di Carlo et al. [7] propose to traverse the asteroid belt with a high elliptical orbit and visit as many asteroids, which are encountered along that orbit, as possible. An ANN was also employed in the determination of multiple NEA missions using solar sails with a characteristic acceleration $a_c = 0.75$ mm/s^2 [5]. However, the latter appears to be considerably higher than what is currently considered as "near-term" ($0.2 \leq a_c \leq 0.35$ mm/s^2), according to ENEAS+ (Exploration of NEAs with solar Sailcraft) mission studies [71], the Gossamer Roadmap technology study [72], and the research conducted by Peloni at al. [2].

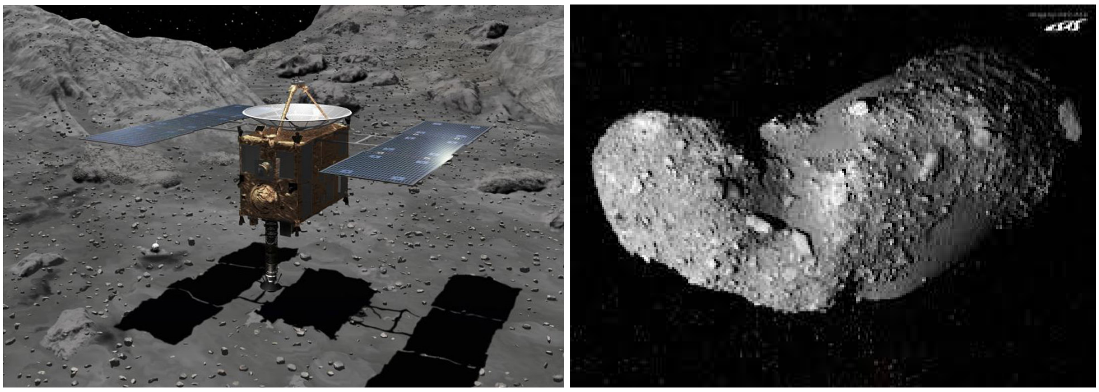
Other works on interplanetary low-thrust trajectory optimisation have been conducted. The mission design can be formulated as an hybrid optimal control problem [73], with an outer-loop solving the discrete optimisation problem and an inner loop solving the continuous optimisation problem. The outer loop utilises the "null-gene" transcription [74] and a discrete genetic algorithm. The inner loop utilises the Sims-Flanagan transcription [75] combined with the monotonic basin hopping global search algorithm [76]. The Sims-Flanagan transcription has also been used in combination with a global optimisation heuristic method to generate an initial guess for a medium-high fidelity optimisation [77]. This allows a trajectory designer to rapidly evaluate a wide range of solutions to the complex problem of identifying low-thrust, multiple-flyby trajectories. Also, the global, low-thrust, interplanetary trajectory optimisation problem has been solved using an hybridisation of a genetic algorithm and a gradient-based direct method [78]. Since it removes the difficulty and biases due to the initial guess generation, this method can provide near-globally-optimal solutions. Similarly, an automatic trajectory planning approach is developed in Ref. [79, 80], where a modified ant colony optimisation algorithm is used to find the optimal sequences of multiple gravity assists.

From these recent missions and research studies, it is possible to conclude that the general practise to solve the complex global optimisation problem to design multi-target missions is to employ a simplified method to solve the continuous part while computing the object sequences (i.e. solving the combinatorial part). Once the sequences have been identified, the final low-thrust optimisation of the most convenient sequences is performed. However, depending on the simplified method selected, the quality of the resulting trajectory can be impacted. For instance, when the sequences are identified by means of impulsive thrusts as a simplified model, this method should be able to estimate the trajectories between objects quickly but, presumably, not very accurately for low-thrust transfers between NEAs. On the other hand, more precise techniques, such as the shape-based methods, can provide a better approximation of the trajectory, but they may require an optimisation to be performed (e.g. for the shape-based methods, the optimisation aims at searching for the optimal shaping parameters, as explained in Section 3.2.2), which makes the computation of each transfer slower.

From the referred studies, it is also possible to notice that the number of objects which can be considered in the analysis can also be affected by the selected simplified method. Depending on how fast the method is to calculate each transfer between pairs of objects, the amount of calculations which can be performed to solve the multi-target problem within a reasonable amount of time can be limited. There is a clear and urgent need to define a method to quickly estimate the transfer cost and duration between asteroids, while maintaining a satisfactory accuracy of the solution. This can potentially increase the number of objects which can be handled in a reasonable time frame and allow to efficiently design multi-target missions using *near-term* low-thrust propulsion systems.

2.2.1 Sample Return to Earth

The first successful sample return missions were the Apollo Moon missions and the Russian Luna 16, 20 and 24 missions, launched between 1968 and 1976, which returned samples of the lunar surface to Earth, contributing to the understanding of



(a) Hayabusa **(b)** 25143 Itokawa

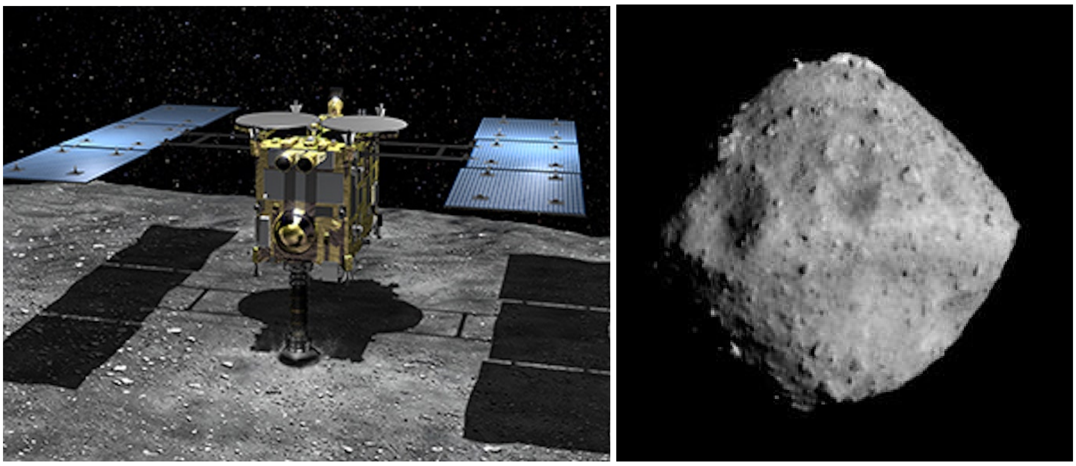
Figure 2.3: JAXA Hayabusa mission (a) to 25143 Itokawa (b) with sample return to Earth. (Credits: JAXA)

the Moon's geological history and composition [81]. Similarly to studying samples of the Moon, the scientific community is interested in collecting samples from the surface of asteroids. It is expected that this will allow us to learn more about the initial stages of the solar system formation and how life began [82, 20].

To date, two missions have returned asteroid samples and one is on its return journey. The JAXA Hayabusa probe (Figure 2.3(a)) rendezvoused with an *S*-type asteroid 25143 Itokawa (Figure 2.3(b)) and, in November 2010, it returned an asteroid sample to Earth [83]. In 2014 JAXA launched the improved Hayabusa2 probe (Figure 2.4(a)) to visit the near-Earth *C*-type asteroid 162173 Ryugu (Figure 2.4(b)). It took samples of the asteroid surface and returned to Earth in December 2020 [84]. The OSIRIS-REx mission (Figure 2.5) was launched by NASA in 2016 to return samples from near-Earth asteroid 101955 Bennu, and it has recently started its return to Earth [82].

The ΔV required to complete sample return missions to planets and small bodies can be considerably high. According to the rocket equation, with increasing ΔV the propellant mass to complete the mission increases exponentially. Also, for a given spacecraft launch mass, a higher ΔV requires the payload mass fraction to be inevitably smaller so that more propellant can be carried on board. This consequently limits the maximum mass of samples which can be returned to Earth [27].

Rendezvousing multiple NEAs with sample return to Earth can increase the scientific return of those missions [3, 4]. MNR missions can provide the possibility of visiting a larger number of asteroids, from which samples can be extracted and returned to Earth for further studies. This kind of missions can thus reduce the cost with respect to employing multiple spacecraft to individual asteroids and returning samples.



(a) Hayabusa2

(b) 162173 Ryugu

Figure 2.4: JAXA Hayabusa2 mission (a) to 162173 Ryugu (b) with sample return to Earth. (Credits: JAXA)

Figure 2.5: OSIRIS-REx mission by NASA. (Credits: NASA)

For instance, GTOC-3 proposed a multiple NEA rendezvous mission with sample return to Earth [85]. The spacecraft, equipped with an electric propulsion system and launched from Earth, must rendezvous with three asteroids from a specified group of 140 NEAs and finally rendezvous with Earth, within ten years from departure. The winning team performed a global search for the asteroid selection, where almost ballistic transfers with limited number of impulsive manoeuvres are assumed, the phasing is omitted and the ΔV is used as the optimisation criterion [86]. Once the sequences are obtained, then a low-thrust optimisation is performed where the spacecraft characteristics and problem assumptions are taken into account. Another solution proposed to use the energy differences between different orbits and the phase differences to estimate the energy required for a tour mission [87]. To this end, a hybrid algorithm of particle swarm optimisation and differential evolution is used to achieve a solution to the energy-optimal tour mission.

The determination of the NEA sequences plays a key role for MNR missions, especially when a return to Earth has to be performed. To this end, the need for a methodology which can quickly identify the most effective sequences of asteroids to visit and then return to Earth arises. This would allow the design of MNR missions in a fraction of the time needed by optimal control solvers. Additionally, it would be beneficial to define an algorithm which can also find sequences where specific asteroids of interest (for which a sample return would be more valuable) can be targeted within the sequence.

2.3 Low-Thrust Propulsion

The choice of the on-board propulsion system to perform high-ΔV missions, such as multiple-target missions and sample return missions, has a major impact on the cost (in terms of spacecraft mass and, consequently, launch cost) and duration. As mentioned, low-thrust systems can be beneficial for this type of missions because they allow to deliver the same ΔV using less propellant than high-thrust propulsion systems [25]. Low-thrust systems can enable the use of smaller, less expensive launch vehicles, making these missions more affordable.

Electric propulsion (EP) is a class of low-thrust space propulsion which utilises electrical power to accelerate a propellant by exploiting electrostatic and/or electromagnetic fields. Compared to chemical systems, the propellant is ejected faster and therefore the overall system is many times more mass-efficient. Since the first launch of an electric propulsion system in 1962 aboard the Zond-2 spacecraft, this type of propulsion have been employed into all sectors of space missions. Thanks to its performance advantages, electric propulsion is expected to further expand within the satellite propulsion market in the years to come [88].

The use of EP for deep-space science missions began with the launch of the Deep Space 1 in 1998. The primary on-board propulsion system is a single-engine ion propulsion system, provided by the NASA Solar Electric Propulsion Technology Applications Readiness project. This propulsion system is designed to deliver a total ΔV of 4.5 km/s to the spacecraft of 486 kg (with around 80 kg of propellant).

A trade study is performed in Ref. [89] by Brophy on advanced propulsion systems for deep-space missions to reach the Comet 67 P/Churyumov-Gerasimenko and return the spacecraft to Earth (Rosetta/CMSR mission by NASA and ESA). The study concluded that ion propulsion enables the use of a smaller, less expensive launch vehicle, and significantly shortens the overall trip time. Also, it resulted that increasing the specific impulse and the maximum power (i.e. greater maximum thrust available) can considerably reduce the required propellant mass. Nevertheless, high values of the specific impulse and the maximum power can result in an increase of the system mass [90]. Consequently, a trade-off is necessary to identify the most appropriate propulsion system, depending on the requirements and goals of the missions.

The use of advanced electric propulsion for navigation and to perform innovative mission control techniques was demonstrated by SMART-1 [91], ESA's first Moon mission which was launched in 2003. An electric engine was also used in the Gravity Field and Steady-State Ocean Circulation Explorer (GOCE) mission [92], which aimed at measuring Earth's gravity field and model the geoid with extremely high accuracy. GOCE was launched in 2009 and the mission ended in 2013 after a planned destructive re-entry into the atmosphere. Other missions which used electric propulsion systems are, for example, the NEOSSat, Artemis, BepiColombo [93].

The successful applications of electric propulsion systems in recent missions have made this solution even more attractive for near-future missions. As an example, the option of using an EP system for NASA's Lunar Gateway, which is an in-development mini-space station in lunar orbit to act as a short-term habitation module and science laboratory and as a solar-powered communication hub, has been investigated. At the moment, a system which can deliver 50 kW of solar electric power is considered [94].

To highlight the outstanding interest which is reserved to the low-thrust propulsion, all the multi-target problems proposed by the GTOC competitions involve the use of a low-thrust system to find feasible solutions. As mentioned, GTOC-3, for example, requires to identify a trajectory to visit three asteroids selected among 140 NEAs and return to Earth. The objective function is to maximise the sum of (i) the fraction between the final and initial mass of the spacecraft and (ii) the fraction between the stay time at each object and the total mission duration. The solution proposed by Ref. [87] requires about 440 kg of propellant mass for a spacecraft with a 2000-kg initial mass and a low-thrust engine with specific impulse $I_{sp} = 3000$ s and maximum thrust $T_{max} = 0.15$ N.

Similarly, GTOC-4 proposes a global optimisation problem to fly-by as many NEAs as possible within 10 years from departure from the Earth using a low-thrust system [70]. The objective is to maximise the number of encounters, while also maximising the final mass of the spacecraft. The winning team identified a solution which allows to fly-by 44 asteroids, when an electric propulsion system with $I_{sp} = 3000$ s and $T_{max} = 0.135$ N is used [95]. In this case, the initial mass is fixed to 1500 kg, assuming that the dry mass is 500 kg. The sequence of asteroids is preliminary calculated by solving a Lambert's problem to estimate the cost and duration of the transfers and then the trajectories associated with the lowest ΔV are optimised.

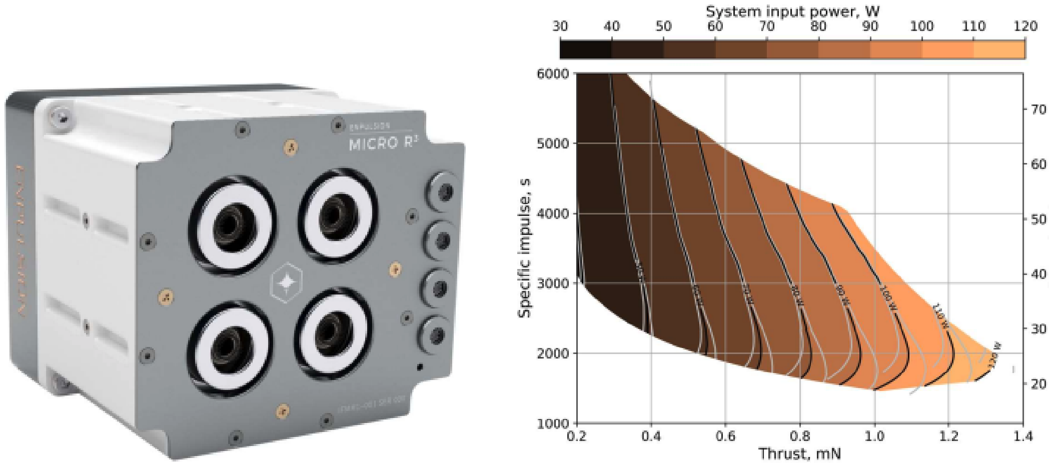
A multiple-ship mission to main-belt asteroids (MBA) is proposed in GTOC-7 [96]. A mothership is launched from Earth and releases exploration probes, which must rendezvous with one or more asteroids. The number of total asteroids visited needs to be maximised, considering that each probe has 2000 kg of mass, of which 1200 kg of propellant. The probes have autonomous electric propulsion systems with specific impulse $I_{sp} = 3000$ s and maximum thrust $T_{max} = 0.3$ N [6].

Another application of low-thrust propulsion systems for multi-target missions is proposed in Ref. [7]. The study consists in visiting the recently-discovered Atira asteroids, characterised by orbits lying completely inside the heliocentric orbit of the Earth. The goal is to visit the maximum number of Atira asteroids with the minimum propellant consumption. The mission is flown by a 700 kg spacecraft using an electric propulsion with $I_{sp} = 3000$ s and $T_{max} = 0.07$ N. A similar approach was employed in Ref. [97], where a tour of MBAs is performed. In this study, the electric propulsion system presents $I_{sp} = 3000$ s and $T_{max} = 0.15$ N. The initial mass of the spacecraft at launch is of 1000 kg, with a propellant mass of about 200 kg. Within a mission time of 7 years, 12 asteroids are visited.

On the basis of the analysis of the previous missions for multi-target rendezvous, the specific impulse used is generally comprised between 3000 and 4000 s and the maximum thrust between 0.07 and 0.3 N. However, off-the-shelf thrusters exist which can reach a higher specific impulse (e.g. 6000 s) with a lower maximum thrust. This kind of thrusters are engineered in a modular approach, with units clustering easily together to form building blocks that can be arranged for various mission profiles. Each thruster unit is very compact and weighs just a few kilograms [98]. For example, the Enpulsion Micro R3 thruster [99] is presented in Figure 2.6, where panel (a) gives a graphic representation of the thruster and panel (b) shows its operational envelope, i.e. the specific impulse and total impulse as function of the thrust and input power. Such small, lightweight, compact and efficient EP systems can be used in multi-target missions and allow for these missions to be flown also by smaller spacecraft. In fact, the strong growth observed in the small satellite sector demonstrates that micro-electric propulsion systems will enable the performance of complex missions by even the smallest satellite [88]. Because of their lower mass, employing small satellites can, eventually, reduce the launch cost (since they can be launched as piggyback or from a smaller and cheaper launch vehicle) [100, 101].

In terms of smaller satellites which can be propelled by EP systems to perform high-energy space missions, the *RemoveDebris* mission was the first mission to successfully demonstrate, in-orbit, a series of technologies that can be used for the active removal of space debris, using a microsatellite of 100 kg in mass [100]. Similarly, Astroscale Japan's ADRAS-J, an 80-kg microsatellite with EP capabilities, will launch to rendezvous with debris, demonstrate proximity operations, deliver observational data to better understand the debris environment, and de-orbit debris [102]. State-of-the-art micro-propulsion options are also examined in Ref. [103] showing that, thanks to their increased operational capabilities, they can be used for demanding missions such as ADR.

NEOSSat is a Canadian microsatellite mission, with the main purposes of determining and monitoring NEA orbits that cannot be efficiently detected from the ground. This mission demonstrates the ability of a microsatellite equipped with EP



(a) Thruster unit

(b) Operational envelope

Figure 2.6: Enpulsion Micro R3 Thruster. (Credit: Enpulsion)

micro-thrusters for multi-NEA missions [104]. The novel mission concept named SIMONE (Smallsat Intercept Missions to Objects Near Earth), whereby a fleet of microsatellites may be deployed to individually rendezvous with a number of NEAs at a very low cost, assesses the ability to accommodate the necessary electric propulsion, power, payload and other onboard systems within the severe constraints of a microsatellite [101].

Another type of low-thrust systems, which is becoming an attractive solution for high-ΔV missions, is the solar sailing propulsion (SSP). A solar sail is a large, lightweight and highly reflective membrane, which is deployed from the spacecraft and can propel it by reflecting solar photons [27]. Essentially, solar sails can provide a continuous acceleration, limited only by the lifetime of the sail film in the space environment. Since it does not require any propellant and the thrust can be theoretically provided for an extended amount of time, solar sailing can be advantageous for multi-target missions. Several studies have been carried out to demonstrate the potential of SSP [105, 5]. However, the technology readiness level (TRL) of this type of propulsion system still needs to be increased to a "flight qualified" level [106] [107].

2.3.1 Low-Thrust Trajectories Optimisation

In essence, computing the trajectory of a spacecraft corresponds at identifying the path which a spacecraft has to travel to go from A (the departure point, i.e. the initial body or orbit) to B (the arrival point, i.e. the target body or orbit). When the optimal trajectory has to be identified, the definition of *optimality* depends on the given mission requirements and/or goals, which can be the minimisation of the propellant consumption or the minimisation of the transfer time.

The literature presented in this chapter shows how, given the appealing efficiency of the low-thrust propulsion systems, its use for orbit transfers, orbit maintenance and interplanetary space missions has been studied extensively. However, the resulting trajectory design problem is particularly challenging to solve considering that the low-thrust systems provide a continuous thrust. For example, the high specific impulse and small maximum thrust generally lead to longer orbital transfers, with respect to chemical propulsion systems. Also, in case of widely spaced departure and arrival orbits, many orbital revolutions may be required to complete the transfer. Then, even if a solution is obtained, it is highly likely that the trajectory is not the global optimal solution [108, 109].

The optimisation of low-thrust trajectories has been the subject of many previous research studies. Numerical optimal control methods have been developed. The collocation method with non-linear programming can be applied to determine low-thrust interplanetary trajectories with minimum time and/or minimum propellant mass [110, 111]. Also, to determine high-accuracy optimal low-thrust transfers, the orbital transfer problem can be transcribed into a constrained nonlinear optimal control problem and can be solved using a variable-order Legendre–Gauss–Radau quadrature orthogonal collocation method [108]. It is found that the so-determined solutions are only locally optimal, as the solution depends on the initial guess.

Calculating optimal interplanetary low-thrust trajectories is usually a complex and time-consuming task, because the local optimisers (based on numerical optimisation techniques) requires an adequate initial guess which can be difficult to compute. As a consequence, novel approaches have been proposed recently. A genetic algorithm can be used to design near-optimal low-thrust trajectories as investigated in Ref. [112], where the transfer is broken into segments and, for each of these segments, a thrust direction is encoded. A global optimisation methodology for low-thrust trajectory which fuses artificial neural networks and evolutionary algorithms is also explored [109].

These novel solutions may solve the issue linked to the definition of an appropriate initial guess, but they can also be computationally expensive to run. To overcome the computational challenges, a variety of approximation methods have been proposed. A commonly-used technique is the orbital averaging, where simple approximations are calculated to describe incremental changes in the orbital elements for each orbital revolution [113].

Shape-based methods have been proposed to approximate low-thrust arcs by shaping the trajectory through a set of parameterised pseudo-equinoctial elements [114, 115, 116, 2]. As such, the shape of the trajectory which connects the departure point A to the arrival point B can be analytically defined, from which the acceleration profile to fly the identified trajectory can be retrieved. These methods do not require the computation of an initial guess, however an optimisation to select the shaping parameters may be required.

In conclusion, computing a low-thrust trajectory can be computationally very expensive, even more so for multi-target missions where millions of combinations of the objects need to be considered and investigated to identify the most convenience sequence. The definition of a methodology that can quickly estimate the cost and duration of low-thrust transfers, which can be used to resolve the combinatorial problem in a more efficient manner, is necessary. Artificial intelligence offers this opportunity, which this thesis aims at exploring.

2.4 Artificial Intelligence

Artificial intelligence (AI) is a thriving academic discipline founded in 1956 which uses *machine intelligence* to solve many challenging problems throughout industry and academia, by replicating some cognitive functions, such as learning and problem solving. AI has many practical applications and active research topics. Nowadays, AI is integrated within our daily life: e.g. advanced web search engines, self-driving cars, speech and image recognition [117]. For example, IBM was able to devise a successful chess strategy which defeated world champion Garry Kasparov in 1997 [118].

Aristotle was the first to define a set of laws governing the rational part of the mind and according to which one can generates conclusions mechanically, given initial premises. AI goal is to build physical system which can replicate these logical rules and solve complex problems *intelligently*. The earliest substantial work in the field of AI was conducted in the mid-20^{th} century, during World War II, by the English mathematician and computer pioneer Alan Turing. In 1936, Turing described an abstract computing machine, which is now commonly-know as the universal Turing machine. The Turing machine manipulates symbols on a strip of tape according to a table of rules. In 1940, Turing built the first operational computer with the purpose of deciphering German messages [119].

In 1982, the first commercial AI program, R1, was developed by Digital Equipment Corporation, which helped to configure orders of computer systems based on the system functionality. By 1986, R1 allowed the company to save about USD 40 million a year. In the following years the interest in AI increased. Nearly every major company started to have its own AI group to improve their systems. However, it is not until the first decades of the 21st century that the AI industry boomed, with AI being successfully applied to many problems in academia and industry. This was possible thanks to the collection of extensive data sets and the application of powerful computer hardware [119].

A part of AI techniques which is vastly used and studied is machine learning (ML). ML interests the algorithms which can improve their performance in a smart fashion through experience and the use of data, from which they can extract patterns and acquire their knowledge. Particularly, ML algorithms can build a complex model starting from *training data*, allowing them to make predictions and/or decisions based on their learning process (without the need to be programmed to do so) [120].

AI and machine learning are currently applied in many fields [119]. For example, in banking AI-based systems have been adopted to provide customer services, detect anomalies and credit card frauds. Finance and trading also rely on computers and data scientists to determine future patterns in the market. AI has been applied also in agriculture to help the farmers to improve the overall harvest quality and quantity while using resources in a more sustainable way. For instance, a German-based tech start-up called PEAT has developed an application which uses images of the soil to identify potential defects and nutrient deficiencies [121]. The healthcare system has also taken advantages of AI in many areas, such as medical record review, population health trending and analytics, radiology images reading, clinical diagnoses making and treatment plans.

Machine learning was successfully used to solve very complex problems also in space mission design. A method based on an evolutionary algorithm and ANNs is employed to determine trajectories to a single NEA using solar sailing [36, 105]. This method is demonstrated to be more efficient in finding solutions than the traditional optimal control methods. Hennes et al. [33] use machine learning methods (e.g. multi-objective evolutionary algorithm and machine learning regressors) to identify low-thrust trajectories with minimum fuel mass between the main belt asteroids. It results that all the proposed ML techniques ensure a distinct improvement in terms of computational time over the state-of-the-art methods (able to compute the ground truth), while improving the precision with respect more rudimentary methods, such as the Lambert's model approximation. This would suggest that machine learning could be able to provide an efficient and reliable solution to quickly estimate the cost and duration of low-thrust transfers. These can then be used in the search to solve the combinatorial part of the global optimisation problem to design multi-target missions.

Mereta et al. [34] compared the accuracy of various machine learning techniques, including ANNs, to determine the initial guess of optimal low-thrust transfers between NEAs. It is found that, without any particular tuning of the parameters of the learning algorithms, the gradient booster regressor and the neural network exhibit the best performance. Other applications of neural networks also include the solution of optimal control problems, where the network is trained to identify the optimal control history in different cases of pinpoint landing [122] and orbit prediction [35]. In the latter case, historical orbit determination and prediction data are used to train the network to improve the orbit prediction accuracy. An ANN was also employed in the determination of multiple NEA missions using solar sails with a characteristic acceleration $a_c = 0.75\ \mathrm{mm/s^2}$ [5].

ANNs based on an online framework can also be applied for low-thrust trajectory optimisation, where the online framework refers to the ability of adapting the network with new transfer data obtained from newly found asteroids [123]. This has been proved to provide better generalisation capabilities. Also, deep feed-forward ANN are applied in Ref. [23, 124] to optimise (i) the transfer time or (ii) the fuel consumption of low-thrust transfers, or (iii) the total ΔV of perturbed multi-impulse transfers. It is shown how deep networks are capable of estimating the final mass or time of optimal transfers with extremely high accuracy. In Ref. [125], deep networks are used successfully to represent, on-board, the optimal guidance profile of an interplanetary mission. The test case of Earth-Mars orbital transfer is considered.

From the studies in the literature, it can be noted that a database of samples can be defined by running the trajectory model. The generated database can then be used to train a machine learning model so that it can represent a surrogate model of the trajectory [126]. This would allow to avoid expensive evaluations of the objective function, which is particularly relevant when the trajectory computation requires to solve an optimal control problem, thus requiring a large computational time and effort (e.g. in case of optimal low-thrust transfers) [127]. For example, in case of multiple asteroid and multiple debris rendezvous missions [33, 34], machine learning techniques can be used to approximate the final mass or duration of transfers to quickly identify near-optimal launch and arrival epochs and the most convenient body sequence. Similarly, the NEA targets in a rendezvous missions can be selected by unsupervised learning techniques, if a proper metric coping with the orbital non-linearities is defined [128].

Using ANNs would allows to get rid of the initial guess (needed instead for conventional optimisation techniques) so that the cost and duration of near-global optimal trajectories can be estimating starting from only the initial and final conditions. As demonstrated by the referred studies, AI techniques generally allow to reduce the computational effort required with respect to traditional methodologies.

ANNs are inspired by the process of information in the human nervous system. They are analogue, fault tolerant, adaptive and characterised by a massively parallel distributed structure [129][130]. Neural networks are composed by processing elements, called *neurons*, which connected together are able to learn from experience, use previous examples to generalise to new ones, and identify noisy and irrelevant data from inputs so that consistent output can be produced. In particular, a neural network resembles the brain for two main aspects [130]:

- the knowledge is acquired by the network from its environment through a learning process;
- the strength of the inter-neuron connections, known as *weights*, are used to store the acquired knowledge.

To summarise, the aim of this PhD study is to explore how neural networks can be employed to estimate the cost and duration of low-thrust transfers so that the most convenient sequences of multiple targets can be computed quicker. The effect of using machine learning to solve multi-target missions on the accuracy of the identified solutions is investigated with respect to the state-of-art methods. The methodologies which are currently used to solve this kind of problems are very computationally expensive, considering the large number of objects involved, and require some simplifications to obtain a solution in a reasonable time frame. The ANN performance is investigate to demonstrate whether it can (i) accurately approximate the cost and duration of low-thrust transfers, given the departure and arrival conditions, (ii) generalise the results for all the possible combinations of objects, and ultimately (iii) reduce the computational effort required to solve the complex global optimisation problem to design multi-target missions.

3

Technical Background

This chapter provides the technical background of the models and techniques which are used within this thesis to achieve the research objectives. Section 3.1 describes the orbital dynamics models, defining the state variables, the equations of motion and the perturbing accelerations. The techniques which can be used to solve optimal control problems, including the trajectory optimisation problems to identify the optimal trajectory and related control history, are presented in Section 3.2. In Section 3.3, ANNs are presented from a technical point of view, detailing the hyper-parameters which define the network architecture and the training algorithms which can be utilised. Finally, Section 3.4 describes the tree-search method and breadth first criterion used to implement the sequence search algorithm.

3.1 Orbital Dynamics Models

In this section, the dynamics of multi-target missions is presented, where the spacecraft is modelled as a point mass with continuous low-thrust. The relative motion of a body i can be formulated, in first-order approximation, with respect to a non-rotating inertial reference frame centred at the central body k and under the influence of the gravitational attraction of body k (two-body motion) as follows [131]:

$$\frac{\mathrm{d}^2\mathbf{r}}{\mathrm{d}t^2} = -\frac{Gm_k}{r^3}\mathbf{r} = -\frac{\mu_k}{r^3}\mathbf{r} \tag{3.1}$$

where $\mathbf{r}$ is the position vector of body i with respect to body k and r indicates its magnitude, m_k is the mass of body k, and G is the universal gravitational constant with $\mu_k = Gm_k$ being the gravitational parameter of the central body k. The two-body motion leads to an orbit with the shape of a conic section (circle, ellipse, parabola, hyperbola) [131].

In the problems of MADR and MNR, which we aim at solving, the central body is Earth with gravitational parameter μ_e for MADR missions, while the Sun with gravitational parameter μ_s is the central body for the MNR missions. It should be noted that, apart from the motion of the spacecraft, the two body problem also describes the motion of the debris objects around the Earth (MADR) and the motion of the Earth and NEAs around the Sun (MNR).

To completely determine and describe the motion, it is important to define the state variables. Figure 3.1 shows how the classical orbital elements (COE), also known as Keplerian elements, identify a specific orbit [24]. Two of these elements define the shape and size of an orbit:

- Semimajor axis, a: the sum of the periapsis and apoapsis distances divided by two;
- Eccentricity, e: describing the amount by which the orbit deviates from a perfect circle (i.e. $e = 0$). Values of e between 0 and 1 form an elliptical orbit, $e = 1$ indicates a parabolic orbit and $e > 1$ defines a hyperbolic orbit.

Other two elements define the orientation of the orbital plane with respect to the reference plane (i.e. the equatorial plane for Earth-orbiting satellites (MADR) and the ecliptic plane for satellites in solar orbits (MNR)):

- Inclination, i: inclination of the orbital plane with respect to the reference plane;
- Longitude of the ascending node, Ω: the angle between the reference direction, or reference frame's vernal point and the ascending node of the orbit along the reference plane.

The remaining two elements are:

- Argument of periapsis, ω: the angle measured from the ascending node to the periapsis, to define the orientation of the orbit in the orbital plane;
- True anomaly θ at epoch t_0: it defines the position of the orbiting body along the orbit at a specific time, measured from pericentre.

The mean anomaly, M, is a fictitious angle which varies linearly with time and is mathematically convenient to describe the position of an object over time. The mean anomaly can be easily converted into the true anomaly, and viceversa [131].

Apart from the Keplerian elements, there are other sets of elements which can describe the orbit thoroughly. For example, by using the modified equinoctial elements (MEE), numerical singularities for zero eccentricity and inclination of the classical Keplerian elements can be avoided [24]. The MEE are defined from the COE as follows:

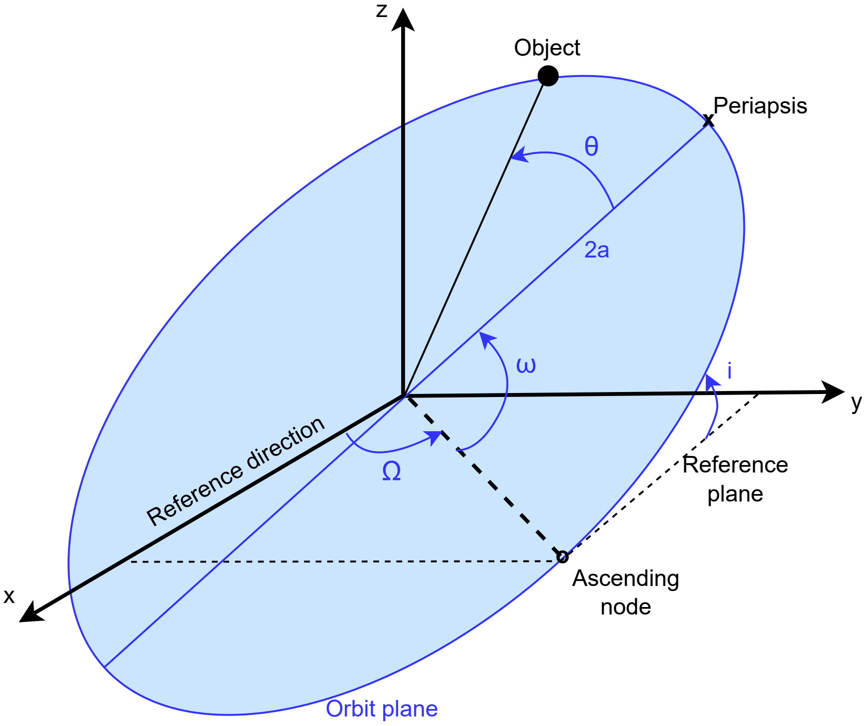


Figure 3.1: Graphic representation of the Keplerian orbital elements.

$$p = a(1-e^2) \tag{3.2}$$
$$f = e\cos(\omega+\Omega) \tag{3.3}$$
$$g = e\sin(\omega+\Omega) \tag{3.4}$$
$$h = \tan(0.5i)\sin(\Omega) \tag{3.5}$$
$$k = \tan(0.5i)\cos(\Omega) \tag{3.6}$$
$$L = \omega+\Omega+\theta \tag{3.7}$$

where p is the semi-latus rectum, f and g are the elements describing the eccentricity, h and k are elements describing the inclination, and L is the true longitude.

The dynamics of the system can be described using the following differential equations [132]:

$$\dot{\mathbf{x}}(t) = \mathbf{A}(\mathbf{x})\mathbf{a} + \mathbf{b}(\mathbf{x}) \tag{3.8}$$

where $\mathbf{x}$ is the state vector of MEE, $\mathbf{x} = [p, f, g, h, k, L]$ [24]; $\mathbf{a}$ is the perturbing acceleration in radial, transverse, and out-of-plane components; $\mathbf{A}(\mathbf{x})$ and $\mathbf{b}(\mathbf{x})$ are the matrix and the vector of the dynamics, respectively. The matrix $\mathbf{A}(\mathbf{x})$ can be fully formulated as follow:

$$\mathbf{A} = \begin{bmatrix} 0 & A_{1,2} & 0 \\ A_{2,1} & A_{2,2} & A_{2,3} \\ A_{3,1} & A_{3,2} & A_{3,3} \\ 0 & 0 & A_{4,3} \\ 0 & 0 & A_{5,3} \\ 0 & 0 & A_{6,3} \end{bmatrix} \tag{3.9}$$

where:

$$A_{1,2} = \frac{2p}{q}\sqrt{\frac{p}{\mu}} \tag{3.10a}$$

$$A_{2,1} = \sqrt{\frac{p}{\mu}}\sin(L) \tag{3.10b}$$

$$A_{2,2} = \sqrt{\frac{p}{\mu}}\frac{1}{q}\big((q+1)\cos(L)+f\big) \tag{3.10c}$$

$$A_{2,3} = -\sqrt{\frac{p}{\mu}}\frac{g}{q}\,(h\sin(L)-k\cos(L)) \tag{3.10d}$$

$$A_{3,1} = -\sqrt{\frac{p}{\mu}}\cos(L) \tag{3.10e}$$

$$A_{3,2} = \sqrt{\frac{p}{\mu}}\frac{1}{q}\big((q+1)\sin(L)+g\big) \tag{3.10f}$$

$$A_{3,3} = \sqrt{\frac{p}{\mu}}\frac{f}{q}\,(h\sin(L)-k\cos(L)) \tag{3.10g}$$

$$A_{4,3} = \sqrt{\frac{p}{\mu}}\frac{s^2\sin(L)}{2q} \tag{3.10h}$$

$$A_{5,3} = \sqrt{\frac{p}{\mu}}\frac{s^2\cos(L)}{2q} \tag{3.10i}$$

$$A_{6,3} = \sqrt{\frac{p}{\mu}}\,(h\sin(L)-k\cos(L)) \tag{3.10j}$$

while the vector $\mathbf{b}(\mathbf{x})$ can be formulated as follows:

$$\mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \sqrt{\mu p}\left(\frac{q}{p}\right)^2 \end{bmatrix} \tag{3.11}$$

where:

$$q = 1 + f\cos(L) + g\sin(L) \tag{3.12}$$

$$s^2 = 1 + \chi^2 \tag{3.13}$$

$$\chi = \sqrt{h^2 + k^2} \tag{3.14}$$

3.1.1 Perturbing Accelerations

For the MADR missions, the perturbing acceleration, $\mathbf{a}$, is given by (i) the acceleration due to thrust, $\mathbf{a_T}$ (acting only on the chaser and not on the debris objects), (ii) the acceleration due to the oblateness of Earth, $\mathbf{a_g}$, and (iii) the acceleration due to the atmospheric drag, $\mathbf{a_D}$. For MNR missions, the effects of the Earth's oblateness and atmosphere are negligible and only the thrust acceleration is considered, i.e.:

$$\text{for MADR:}\;\; \mathbf{a} = \mathbf{a_T} + \mathbf{a_g} + \mathbf{a_D} \tag{3.15}$$

$$\text{for MNR:}\;\; \mathbf{a} = \mathbf{a_T} \tag{3.16}$$

It should be noticed that, in MADR missions, the motion of the debris objects is also propagated starting from the given initial conditions and considering the same dynamics with gravitational and atmospheric perturbations (but no thrust).

Thrust

The acceleration of the spacecraft due to the thrust $\mathbf{a_T}$, acting in both MADR and MNR scenarios, is formulated as:

$$\mathbf{a_T} = \frac{T_{max}}{m}\mathbf{N} \tag{3.17}$$

where T_{max} is the maximum thrust of the propulsion system, m is the mass of the system and $\mathbf{N} = [N_r, N_\theta, N_h]^T$ indicates the acceleration direction and magnitude vector in radial, transverse, and out-of-plane components. The mass of the system m decreases with time due to the propellant consumption as described by the following mass differential equation:

$$\dot{m} = -\frac{T_{max}|\mathbf{N}|}{I_{sp}g_e} \tag{3.18}$$

where $0 \leq |\mathbf{N}| \leq 1$ is the magnitude of $\mathbf{N}$, which accounts for the thrust throttling, I_{sp} is the specific impulse of the propulsion system and g_e is the gravitational acceleration at the Earth's surface.

As mentioned, low-thrust propulsion systems are used in this study. Specifically, it is chosen to use electric propulsion for both MADR and MNR missions. However, as mentioned in Section 2.3, solar sailing propulsion is becoming an attractive solution which utilises radiation pressure exerted by the sunlight on large reflective panels (the *sails*) to generate thrust. This type of low-thrust propulsion offers the possibility of low-cost operations combined with long operating lifetimes, since no propellant is required. To analyse the performance of SSP with respect to EP and whether the ANN can be applied in both cases, the option of using solar sailing is investigated for MNR missions (Chapter 6).

When solar sailing is used, the acceleration of the spacecraft due to thrust can be formulated as:

$$\mathbf{a_T} = a_c \left(\frac{r_E}{r}\right)^2 \cos^2(\alpha_s)\mathbf{N_s} \tag{3.19}$$

where the term a_c is the so-called characteristic acceleration of the sail and it expresses the acceleration provided by a solar sail facing the Sun at the average Sun-Earth distance, i.e., r_E = 1 AU. The cone angle α_s is the angle between the Sun-spacecraft direction and the sail normal unit vector, which is $\mathbf{N_s} = [N_r, N_\theta, N_h]^T$ and $|\mathbf{N_s}| = 1$. The acceleration magnitude can thus be changed by changing the cone angle. To obtain Eq.(3.19) some approximations related to the geometrical and the optical properties of the sail shall be considered. First, the sail is considered perfectly flat, with no wrinkles and deformations which may be due to the tensing up of the membrane while propelling. Second, a perfectly-reflecting sail membrane is considered, as shown in Figure 3.2, where $\mathbf{u_i}$ and $\mathbf{u_r}$ are the incident and reflected radiation vectors, respectively.

Non-Spherical Gravitational Acceleration

The gravitational acceleration is experienced by both the chaser and debris objects in MADR missions, due to the Earth's oblateness and mass density distribution in the north-south direction (i.e. zonal harmonics only), can be defined as follows [131]:

$$\mathbf{a_g} = \mathbf{Q_r^T}\delta\mathbf{g} \tag{3.20}$$

where $\mathbf{Q_r} = [\mathbf{i_r}, \mathbf{i}_\theta, \mathbf{i_h}]$ is the transformation matrix from the rotating local-vertical-local-horizontal (LVLH) frame to the Earth-centred inertial (ECI) frame, whose com-

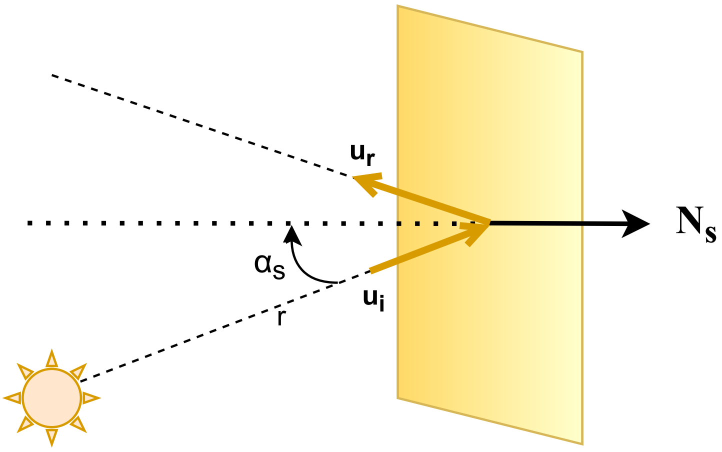


Figure 3.2: Representation of a perfectly-reflecting flat solar sail.

ponents are:

$$\mathbf{i_r} = \frac{\mathbf{r}}{|\mathbf{r}|}, \quad \mathbf{i}_\theta = \mathbf{i_h} \times \mathbf{i_r}, \quad \mathbf{i_h} = \frac{\mathbf{r} \times \mathbf{v}}{|\mathbf{r} \times \mathbf{v}|} \tag{3.21}$$

with **r** and **v** being, respectively, the position and velocity vectors of the spacecraft in the ECI frame. The perturbation acceleration $\delta\mathbf{g}$ is formulated as:

$$\delta\mathbf{g} = \delta g_n \mathbf{i_n} - \delta g_r \mathbf{i_r} \tag{3.22}$$

where $\mathbf{i_n}$ is the local north direction:

$$\mathbf{i_n} = \frac{\mathbf{e_n} - (\mathbf{e_n^T i_r})\mathbf{i_r}}{|\mathbf{e_n} - (\mathbf{e_n^T i_r})\mathbf{i_r}|} \tag{3.23}$$

and

$$\delta g_n = -\frac{\mu_e \cos(\phi)}{r^2} \sum_{k=2}^{n} \left(\frac{R_e}{r}\right)^k P_k'(\sin(\phi)) J_k \tag{3.24}$$

$$\delta g_r = -\frac{\mu_e}{r^2} \sum_{k=2}^{n} (k+1) \left(\frac{R_e}{r}\right)^k P_k(\sin(\phi)) J_k \tag{3.25}$$

with $\mathbf{e_n} = [0,0,1]$, R_e the equatorial radius of the Earth, $r = p/q$, $P_k \sin(\phi)$ representing the k-th degree Legendre polynomial whose derivative with respect to $\sin(\phi)$ is $P_k' \sin(\phi)$, with ϕ being the geocentric latitude, and J_k being the zonal harmonic coefficient for $k = \{2,3,4\}$. Beyond the fourth term, the effects are considered negligible for the purpose of this research.

Atmospheric Drag

The acceleration due to the atmospheric drag in the radial, transverse and normal components can be defined as:

$$\mathbf{a_D} = [a_{D_r} \quad a_{D_\theta} \quad 0] \tag{3.26}$$

where the out-of-plane component of $\mathbf{a_D}$ is negligible, as the net plane change is close to zero. The radial and transverse components are defined as follows:

$$a_{D_r} = -0.5\rho S C_D v v_r \tag{3.27}$$

$$a_{D_\theta} = -0.5\rho S C_D v v_\theta \tag{3.28}$$

where ρ is the atmospheric density, which can be estimated using the *Exponential Atmospheric Model*, which considers the atmosphere as composed by an ideal gas at constant temperature in a hydrostatic equilibrium [133]. This model is a compromise between accuracy and ease of implementation and considered appropriate for the level of the study. Figure 3.3 shows the atmospheric density according to the exponential model used and Table 3.1 presents the values of the density at certain altitudes. Also, S is the aerodynamic surface area, C_D is the drag coefficient and v is the velocity magnitude, i.e. $v = \sqrt{v_r^2 + v_\theta^2}$, with v_r and v_θ being its radial and tangential components:

$$v_r = \sqrt{\frac{\mu_e}{p}}\big(f\sin(L) - g\cos(L)\big) \tag{3.29}$$

$$v_\theta = \sqrt{\frac{\mu_e}{p}}\big(1 + f\cos(L) + g\sin(L)\big) \tag{3.30}$$

The numerical values of the physical parameters, which are used in this study, are detailed in Table 3.2.

3.2 Optimal Trajectory Optimisation Methods

An *optimal control problem* (OCP) must be solved to identify an optimal low-thrust trajectory. By solving an OCP, the control history can be determined so that the *performance index* or *objective* (or *cost*) *function*, J, is minimised (or maximised) while fulfilling the dynamical and physical constraints, such as the orbital dynamics described in the previous section and the initial and final conditions [134]. Without loss of generality, the problem of finding the minimum of a given objective function is considered in the remainder of this thesis.

Table 3.1: Exponential Atmospheric Model.

Altitude, km	Density, kg/m^3
350	6.66×10^{-12}
450	1.09×10^{-12}
550	2.14×10^{-13}
650	6.73×10^{-14}
750	6.95×10^{-15}
850	2.78×10^{-15}
1000	1.49×10^{-15}
1250	2.79×10^{-16}
1500	2.79×10^{-16}
2000	9.09×10^{-17}

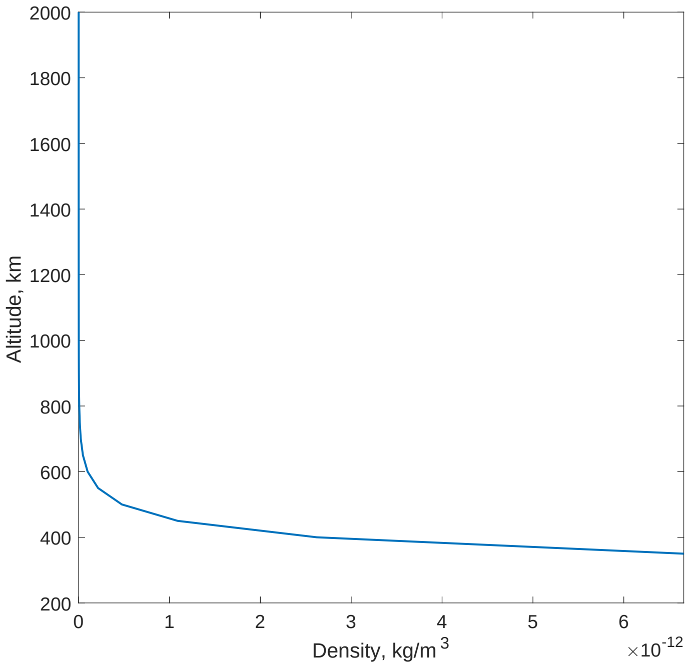


Figure 3.3: Exponential Atmospheric Model.

An OCP has the following general formulation [31, 135, 136]:

Minimise the objective function:

$$J = \Phi(\mathbf{x}(t_0), t_0, \mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} L(\mathbf{x}(t), \mathbf{u}(t), t)\,dt \tag{3.31}$$

Table 3.2: Constant parameters.

Variable	Constant	Value
Gravitational acceleration at Earth's surface	g_e	9.8066 $\mathrm{m/s^2}$
Gravitational parameter of the Sun	μ_s	1.3271 $\times 10^{20}$ $\mathrm{m^3/s^2}$
Gravitational parameter of the Earth	μ_e	3.9860 $\times 10^{14}$ $\mathrm{m^3/s^2}$
Equatorial radius of the Earth	R_e	6378.14 $\times 10^3$ m
Zonal harmonic coefficient of 2^{nd} order	J_2	1082.639 $\times 10^{-6}$
Zonal harmonic coefficient of 3^{rd} order	J_3	-2.565 $\times 10^{-6}$
Zonal harmonic coefficient of 4^{th} order	J_4	-1.608 $\times 10^{-6}$

subject to the dynamical constraints

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \tag{3.32}$$

and the boundary conditions of the state and control variables

$$\mathbf{x_{min}} \leq \mathbf{x} \leq \mathbf{x_{max}} \tag{3.33}$$

$$\mathbf{u_{min}} \leq \mathbf{u} \leq \mathbf{u_{max}} \tag{3.34}$$

and the boundary constraints at the initial and final time

$$\mathbf{g_{min}} \leq \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), t) \leq \mathbf{g_{max}} \tag{3.35}$$

and the path constraints

$$\mathbf{c_{min}} \leq \mathbf{c}(\mathbf{x}(t), \mathbf{u}(t), t) \leq \mathbf{c_{max}} \tag{3.36}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ is the state vector, $\mathbf{u}(t) \in \mathbb{R}^m$ is the control vector with respect to which the performance index is minimised and $t \in [t_0, t_f]$ is the independent variable. The objective function J is defined by Φ and L, which are the terminal cost function and integral cost function, respectively, to account for the cost over the whole time frame [137].

The space-trajectory optimisation problem aims at determining the trajectory which allows the spacecraft to reach the final target conditions starting from some initial conditions in the minimum amount of time and/or using the minimum amount of propellant [135]. The dynamics equations, defined in the previous section, describe the motion of the spacecraft in MEE. The control vector determines the thrust acceleration vector. The requirements of the mission define the constraints, e.g. initial and terminal state constraints, maximum amount of propellant available, maximum duration of the mission.

Starting from the general formulation of an OCP in Eqs.(3.31) to (3.36), the OCP for the specific case of the low-thrust transfers can be defined. The performance index to minimise can be either:

$$\text{For TOF minimisation:} \quad J = t_{0,f} \tag{3.37}$$

$$\text{For propellant minimisation:} \quad J = -m(t_f) \tag{3.38}$$

subject to the dynamical constraints described in Eq.(3.8) and Eq.(3.18) and the path constraint:

$$0 \leq |\mathbf{N}| \leq 1 \quad \text{for EP} \tag{3.39}$$

$$|\mathbf{N_s}| = 1 \quad \text{for SSP} \tag{3.40}$$

where $|\mathbf{N}|$, indicating the magnitude of $\mathbf{N}$, can vary for EP to allow for thrust throttling. The control vector is bounded so that its components N_r, N_θ, $N_h \in [-1,1]$ for EP, while N_θ, $N_h \in [-1,1]$ and $N_r \in [0,1]$ for SSP to take into account the inability of the solar sail to thrust towards the Sun. Moreover, since the intent of this work is to design rendezvous missions, the state vector needs to be bounded to ensure that the rendezvous conditions are satisfied, which are:

$$\begin{cases} \mathbf{r}(t_d) - \mathbf{r}_d(t_d) = \mathbf{0} \\ \mathbf{v}(t_d) - \mathbf{v}_d(t_d) = \mathbf{0} \end{cases} \quad \text{and} \quad \begin{cases} \mathbf{r}(t_a) - \mathbf{r}_a(t_a) = \mathbf{0} \\ \mathbf{v}(t_a) - \mathbf{v}_a(t_a) = \mathbf{0} \end{cases} \tag{3.41}$$

where $\mathbf{r}$ and $\mathbf{v}$ are the position and velocity of the spacecraft, $\mathbf{r}_d$ and $\mathbf{v}_d$ are the position and velocity of the departure body at the departure time t_d, and $\mathbf{r}_a$ and $\mathbf{v}_a$ are the position and velocity of the arrival body at the arrival time t_a, as taken from the ephemerides.

The trajectory optimisation problem is a OCP whose solution is very complex for several reasons, such as the non-linear dynamics, unknown terminal conditions which are dependent of the terminal time (which can be the optimisation variable), and the presence of time-dependent forces.

The optimisation methods, which have been developed, can be divided into *local optimisation* and *global optimisation* methods on the basis of their search capabilities [31, 138, 139, 140]. The local optimisation methods require an initial guess and, generally, the solution found is close to the initial guess chosen, thus constituting a *local optimum*. The global optimisation methods, instead, can find the *global optimum* and are inspired by nature (e.g. the biological evolution of the species) to optimise an initial population. Metaheuristic algorithms, which are generally suited for global optimisation, are detailed in Section 3.2.3.

The local optimisation methods are subdivided into *indirect methods* and *direct methods*. Indirect methods can solve the OCP by using the analytical necessary conditions for the *calculus of variations*, by introducing the Lagrange multipliers, doubling the size of the dynamics [141, 136, 135]. Differently, the OCP performance index can be minimised by means of a direct method, which consists in transcribing the continuous OCP into a discrete finite-dimensional non-linear programming problem (NLP). Direct methods can be divided into *direct shooting methods* if only the control is parametrised (while the state is propagated from its initial value) [75], and *direct collocation methods* if both control and state are parametrised [139, 142, 143].

Although the choice of the initial guess has an influence on the quality of the optimised solution in all local optimisation methods, the direct collocation method offers a convergence which is generally robust also to poor initial guesses. Also, the initial guess which needs to be provided in direct methods gives an approximated description of the trajectory. Instead, although some methods have been proposed for an efficient generation of the initial guess also for indirect methods [144], the initial guess in indirect methods has to define a guess of the Lagrange multipliers which lack of a physical meaning. This is compared to direct methods where a guess of the trajectory has to be provided. These reasons drive the choice of using direct methods for this application.

Amongst the direct methods, the direct collocation methods are preferred over the direct shooting methods because the latter may incur in computational issues due to the continuity conditions at the match points. In fact, in direct shooting methods, the trajectory is divided into control nodes, which are points the spacecraft must pass by along its trajectory. The state vector describing the trajectory is forward propagated from the preceding node and backward propagated from the successive node. The two propagated states shall meet at a match point in between the two control nodes where the two propagated states need to match [139].

Also, the metaheuristic methods are not selected to solve the trajectory OCP for mainly two reasons. First, the significant computational effort required, which is derived from the fact that (i) a numerical propagation of the dynamics, together with the random initialisation of the population, is required at every iteration and (ii) multiples runs of the method are needed to obtain a statistical confidence of the global optimality [145]. Second, metaheuristic methods cannot generally localise optima accurately, which translates to the need of a refinement of the solution through a second optimisation method (e.g. using a direct or indirect approach) to obtain the desired accuracy. For the reasons detailed, the direct collation method is selected [140, 146] and further described in Section 3.2.1. Consequently, in the remaining of this dissertation, the optimal solutions are to be considered near-optimal as a local optimisation method is used.

Direct optimisation methods are, however, expensive from the computational point of view and require the calculation of an initial guess. Thus, there may be cases where analytic methods can be beneficial. In particular, shape-based approaches are considered. These approaches aim at defining the trajectory shape by connecting the initial and final states and can identify an accurate solution (although sub-optimal) to the trajectory OCP without the need of an initial guess. They may represent an optimal choice to compute the many thousands of trajectories to generate a database which is used to train the ANNs. For this reason, shape-based methods are further detailed in Section 3.2.2.

Although they are not directly used to solve the trajectory OCP, metaheuristic methods are also employed in this PhD work. The genetic algorithm (GA) is used within the shape-based methods to obtain a sub-optimal solution. Also, the methodology which is developed to conduct the MLT analysis employs a multi-objective GA. Consequently, the GA and multi-objective GA are presented in Section 3.2.3.

3.2.1 Direct Collocation Method

In a direct method, the OCP is transcribed into a discrete NLP problem with finite dimensions. This is done by *collocating* the cost function J, the differential equations and the constraints in a finite number of discretisation points in such a way that they are reduced in non-linear algebraic expressions. For this reason this technique is called *collocation method*. In the class of direct collocation methods, both the state and control vectors are parametrised over a set of collocation points.

The most common sets of collocation points used in the literature are the Legendre-Gauss (LG) points, Legendre-Gauss-Lobatto (LGL) points and Legendre-Gauss-Radau (LGR) points. LG lies in the open interval $\tau \in (-1,1)$, LGL lies in the closed interval $\tau \in [-1,1]$ and LGR lies in the half open interval $\tau \in [-1,1)$, with τ being a normalised independent variable. Figure 3.4 shows the Legendre points as function of τ. These sets of points are obtained from the roots of a Legendre polynomial and/or linear combinations of a Legendre polynomial and its derivatives. To have a deeper understanding of the LG and LGL methods, the interested reader is referred to Refs. [139, 142, 143]. The LGR points are employed in this thesis, thus analysed in further details. The N LGR points are obtained as the roots of $P_{N-1}(\tau) + P_N(\tau)$, where $P_N(\tau)$ is the N-th degree Legendre polynomial which is defined by the Rodrigues' formula as [147]:

$$P_N(\tau) = \frac{1}{2^N N!}\frac{d^N}{d\tau^N}\left(\tau^2 - 1\right)^N \tag{3.42}$$

The advantages of using the LGR points are a straightforward implementation, the possibility of mapping between the costates of the NLP and the costates of the OCP (since the N-th degree Lagrange polynomial is equal to the function at the N collocation points as it will be described in Eq.(3.46)), and the avoidance of the Runge phenomenon, which is more likely to occur in case of equidistant discretisation points [148]. The Runge phenomenon introduces the presence of oscillations of the discretized function at the edges of an interval. Hence why selecting the appropriate type and number of points to discretise the OCP functions is paramount to obtain an

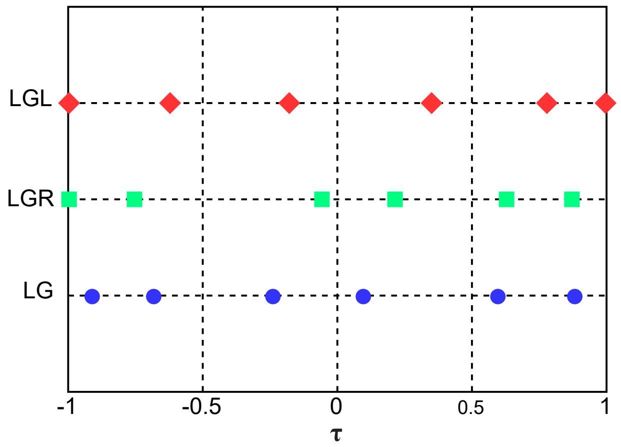


Figure 3.4: Legendre point sets: LG, LGL and LGR points.

accurate solution.

Since the Legendre polynomials represent a set of orthogonal functions only in the domain [-1,1), the LGR points requires a transformation from the time domain to the normalised independent variable τ domain, i.e.:

$$t \in [t_0, t_f) \Longrightarrow \tau \in [-1, 1) \tag{3.43}$$

It should be noticed that a linear relation exists between the time domain and the τ domain, thus it is easy to go from one domain to the other one and vice-versa [148, 149]:

$$\tau = \frac{2}{t_f - t_0} t - \frac{t_f + t_0}{t_f - t_0} \tag{3.44}$$

$$t = \frac{t_f - t_0}{2} \tau + \frac{t_f + t_0}{2} \tag{3.45}$$

Thanks to this property, a continuous-time function $x(t)$ with $t \in [t_0, t_f]$ can be approximated to a discrete polynomial expression $X(\tau)$ with $\tau \in [\tau_1, \tau_N]$ such that $\tau_1 = -1$ and $\tau_N < 1$ for N LGR collocation points (with $\tau_{N+1} = 1$ being the non-collocated point, for $N+1$ discretisation points) as [150]:

$$x(t) \approx X(\tau) = \sum_{i=1}^{N+1} x_i L_i(\tau) \tag{3.46}$$

with x_i being the value of the continuous function $x(t)$ at the collocation point τ_i and $L_i(\tau)$ being the Lagrange polynomial at the collocation point τ_i. It is important to observe that the series in Eq.(3.46) includes the Lagrange polynomial associated with the non-collocated point $\tau_{N+1} = 1$. It is possible to define $L_i(\tau)$ as [151]:

$$L_i(\tau) = \prod_{k=1,k\neq i}^{N+1} \frac{\tau - \tau_k}{\tau_i - \tau_k} \tag{3.47}$$

which satisfies the isolation property for which $L_i(\tau)$ are equal to one at the i^{th} discretisation point, i.e. $\tau = \tau_i$ and zero at all the other points. In this way, at the collocation point τ_i, $x(t_i) = X(\tau_i)$.

Thus, with the definition given above the optimal control problem can be transcribed into a NLP problem as follows:

Minimise the cost function:

$$J = \Phi(\mathbf{X}(\tau_1), \tau_1, \mathbf{X}(\tau_{N+1}), \tau_{N+1}) + \frac{t_f - t_0}{2} \sum_{i=1}^{N+1} w_i \Psi_i \tag{3.48}$$

subject to the dynamical constraints

$$f_i = \sum_{j=1}^{N+1} D_{i,j} X_j - \frac{t_f - t_0}{2} f_C(\mathbf{X}_i, \mathbf{U}_i, \tau_i) = 0 \tag{3.49}$$

the boundary conditions of state and control

$$\mathbf{x_{min}} \leq \mathbf{X}_i \leq \mathbf{x_{max}} \tag{3.50}$$

$$\mathbf{u_{min}} \leq \mathbf{U}_i \leq \mathbf{u_{max}} \tag{3.51}$$

and constraints associated with state and control

$$\mathbf{g_{min}} \leq \mathbf{G}(\mathbf{X}_i, \mathbf{U}_i, \tau_i) \leq \mathbf{g_{max}} \tag{3.52}$$

where τ_i is the considered collocation node, Ψ_i is the Lagrange cost term calculated in τ_i, w_i is the Gaussian quadrature weights associated to τ_i. By using the polynomials, the integral part of the cost function becomes algebraic, which in this way becomes faster to calculate. The functions f and f_C are respectively the residuals associated to the collocated dynamics and the continuous differential equations describing the dynamics of the system. Finally, the Radau pseudospectral differentiation matrix $\mathbf{D}$ is the time derivative of the Lagrange polynomials $L_i(\tau_i)$ with dimensions $N \times (N+1)$ [148, 150]. The NLP is easily solvable by means of well developed software and algorithms [152], such as off-the-shelf solver like SNOPT (Sparse Nonlinear Optimizer) [153] and IPOPT (Interior Point Optimizer) [154, 155, 156].

The method presented is known as *p collocation method.* Since just one function is used to approximate over the whole time range, this method is considered a global orthogonal collocation method. The advantage of using the p method is that the approximation converges *spectrally*, i.e. at an exponential rate, as a function of the number of collocation nodes. Consequently, it allows to reduce the number of collocation nodes and, thus, the degree of the polynomial to reach the desired accuracy. Because of their spectral convergence, this method is also known as *pseudospecral method* (PSM). When the exponential convergence of the Lagrange polynomials is not obtained, the number of collocation points need to be increased to achieve the same level of accuracy. In this case, a τ interval can be subdivided into *meshes*. This is known as *h collocation method*. This is particularly useful in cases where the function is not smooth enough. The non-smooth intervals of the function can be isolated by adding more meshes and defining several lower-degree polynomials to approximate the function [157].

To exploit the advantages of both p and h methods, the *ph collocation method* has been developed to make the computation of the NLP solution more efficient while reducing the size of the finite-dimensional approximation [157]. Thus, it is possible to increase both the degree of the Lagrange polynomials (taking advantage of its spectral convergence) and the number of meshes (employing different order of polynomials within each mesh). The ph method is used in GPOPS-II [158], the general-purpose MATLAB® software for solving multiple-phase OCPs.

GPOPS-II is selected because it allows for a general formulation of the OCP, including the constraints and the boundary conditions. GPOPS-II implements the variable-order LGR quadrature collocation method [142] together with an hp-adaptive mesh refinement method [159]. To use this software, no third-party products other than MATLAB® are required. The NLP problem can be solved by using the open-source NLP solver IPOPT [154, 155, 156] or SNOPT [153]. Considering that the NLP solver SNOPT generally fails to converge if a good initial guess is not available [160], IPOPT is selected and used in this PhD work. The analytical derivatives are calculated using the open-source algorithmic differentiation package ADiGator, for which the algorithm is detailed in Ref. [161]. For a comprehensive description of the mathematics and the implementation being performed in GPOPS-II, the interested reader is referred to Refs. [162, 143, 142, 163, 164] and the references therein.

To solve the low-thrust trajectory OCP, GPOPS-II requires the cost function, the differential-algebraic equations (which include both the differential equations and the path constraints), and the event constraints in the formulation of the problem, which determine the dynamics of the system as described in Section 3.1. GPOPS-II finds a solution and calculates the boundary conditions, constraints, and path errors of the OCP. These are compared against the desired error tolerances. When the tolerances are satisfied, the problem is solved and the optimal solution is found; else, a new mesh is defined and the solution is used as the initial guess of the successive iteration.

3.2.2 Shape-Based Methods

The optimisation methods which have been described so far can be computationally very expensive. In the early stages of a mission design it can be convenient to use an analytical method to quickly identify a feasible trajectory. In Ref. [114], a *shape-based approach* is proposed to represent a trajectory through a parametrised analytical curve or *shape* while computing the control history required to fly that trajectory. *Shaping functions* are defined to describe the evolution of the state in an analytical manner. In this way, the trajectory can be entirely described analytically and no propagation or discretisation of the state is needed.

It should be noticed that these methods provide a sub-optimal solution, which can be useful for preliminary mission design (such as to generate the training database for the ANNs) or as an initial guess to be fed into a finest optimiser. The sub-optimality of the solution is due to two main reasons: firstly, an optimisation is not performed; secondly, the thrust profile is retrieved only a posteriori once the shape of the trajectory is computed and no explicit constraints on the thrust acceleration vector profile can be imposed. Once the solution for a trajectory connecting the departure point and the arrival point in a central force field is obtained by means of the preferred shape-based method, the necessary acceleration that the propulsion system has to deliver can be calculated algebraically by:

$$\mathbf{a} = \ddot{\mathbf{r}} + \mu \frac{\mathbf{r}}{r^3} \tag{3.53}$$

which satisfies the two-body motion equation, as described earlier, and where $\mathbf{r}$ is the position vector in Cartesian coordinates.

Several shape-based approaches have been developed, which differ in the approximations considered, shaping functions and state variables used. Depending on the chosen shaping function, the definition and number of shaping parameters that need to be tuned to shape the trajectory may vary. However, most of the methods assume a tangential thrust for the whole duration of the trajectory.

The shape-based methods were introduced for the first time by Petropoulos and Longuski [114, 165], describing the in-plane motion by means of an *exponential sinusoid* shape. It requires the definition of six shape parameters to fully describe the shape of the trajectory and acceleration. Boundary conditions can be applied on the position, but not on the velocity. Other two of the most commonly-used shape based methods are the pseudo-equinoctial method [115] and the inverse polynomial method [166]. The former has three shape parameters to determine, while the latter does not have any because in this case the shaping parameters are calculated from the boundary conditions. Both methods can impose boundary conditions on both the position and velocity and they can describe a 3-D trajectory. However, the inverse polynomial provides an accurate result only in cases where the transfers have to realise small changes of inclination. Since the pseudo-equinoctial method offers a good trade-off between the number of the independent shaping parameters and the accuracy of the trajectory, this shape-based method is employed in this work and, thus, described in details below.

Pseudo-Equinoctial Method

De Pascale and Vasile [115] propose a shape-based method based on modified equinoctial elements, which can accurately and efficiently describe the evolution of the trajectory when a low-thrust propulsion system is used. They define two different sets of the shaping function: (i) the linear-trigonometric shape and (ii) the exponential shape.

The linear-trigonometric shape is defined as:

$$\begin{cases} p(L) = p_0 + p_f(L - L_0) + \lambda_1 \sin(L - L_0) \\ f(L) = f_0 + f_f(L - L_0) + \lambda_2 \sin(L) \\ g(L) = g_0 + g_f(L - L_0) + \lambda_2 \cos(L) \\ h(L) = h_0 + h_f(L - L_0) + \lambda_3 \sin(L) \\ k(L) = k_0 + k_f(L - L_0) + \lambda_3 \cos(L) \end{cases} \tag{3.54}$$

while the exponential shape is defined as:

$$\begin{cases} p(L) = p_0 + p_f \exp(\lambda_1(L - L_0)) \\ f(L) = f_0 + f_f \exp(\lambda_2(L - L_0)) \\ g(L) = g_0 + g_f \exp(\lambda_2(L - L_0)) \\ h(L) = h_0 + h_f \exp(\lambda_3(L - L_0)) \\ k(L) = k_0 + k_f \exp(\lambda_3(L - L_0)) \end{cases} \tag{3.55}$$

where the subscripts 0 and f indicate the initial and final conditions which are defined by the boundary conditions.

In both the linear-trigonometric and exponential shapes, λ_1, λ_2, and λ_3 are the shaping parameters associated to p, (f, g), and (h, k), respectively. The linear-trigonometric shape is advantageous to describe trajectories with tangential thrust whose magnitude can vary. The exponential shape better represents trajectories with constant tangential thrust. Thus, the former is more suited for the case of electric propulsion systems and solar sailing, and it is employed in this study.

This procedure can approximate the shape of rendezvous trajectories with minimum propellant mass or duration in the given range of launch dates, TOF and number of revolutions. Considering MNR missions, for each pair of asteroids, the pseudo-equinoctial method can identify the shape of the transfer and retrieve the control history necessary to perform the calculated transfer. To this end, the MATLAB® built-in GA is run to determine the optimal shaping parameters, λ_1, λ_2 and λ_3, to obtain

Table 3.3: Bounds of the parameters used in the shape-based method.

Parameter		Lower Bound	Upper Bound
Launch date		2020-01-01	2030-12-31
TOF, days		400	1500
Propellant ratio		–	0.6
N. revolution		0	4
Shaping parameters	λ_1	-0.5	0.5
	λ_2	-0.1	0.1
	λ_3	-0.01	0.01

the transfer shape with minimum propellant mass or minimum time of flight. The control history can be changed by changing the shape, thus the shaping parameters, so that the acceleration constraint (Eq.(3.53)) is satisfied [115]. The bounds of the parameters used by the GA in the shape-based algorithm to select the shaping parameters are specified in Table 3.3. These bounds are chosen and fixed so that low-thrust transfers with different characteristics (in terms of launch date, TOF, and number of revolutions) can be calculated by the shape-based method, depending on orbital parameters of the departure and arrival bodies.

3.2.3 Metaheuristic Methods

Metaheuristic methods are global optimisation methods. They combine a heuristic component with an "intelligent" search, allowing the method to find the global optimum. The most commonly-used metaheuristic methods are the GA [167], particle swarm optimisation (PSO) [168, 169], differential evolution (DE) [170], ant colony optimisation (ACO) [171, 79, 80], and simulated annealing [172]. These algorithms are mostly inspired by the evolution of natural systems. Examples are: the GA is inspired by the process of natural selection, according to which the population naturally evolves based on the "survival of the fittest" (which is detailed below); the ACO aims at finding the optimal solution by emulating the behaviour of ants seeking a path between their colony and a source of food; and the PSO simulates the social behaviour which drives the movement of a bird flock or fish school towards the best solution [173].

Metaheuristic methods consist mainly in defining an initial (often random) population within the search space and letting the population evolve towards the optimum by following a predetermined set of rules. For this reason, they are defined as population-based methods. The performance and optimality of these methods are analysed and verified by running the same problem with the same method several times.

Metaheuristic approaches have been used in space-trajectory optimisation when impulsive ΔV are considered, i.e. when chemical thrusters are used as the main propulsion system, since the restricted number of optimisation variables (such as TOF, launch date, available ΔV) limits the dimensionality of the search space [168, 174]. As mentioned, for low-thrust trajectory optimisation, which is a continuous OCP, metaheuristic methods need to be used in combination with other techniques. For example, if coupled with indirect methods they can be used to determine the initial values of the Lagrange multipliers [168, 175], or they can be coupled with direct methods to define the ΔV and epochs of the impulsive manoeuvres within the direct shooting method [176, 177]. Also, they can be employed to determine the optimal control law of a transfer when used in combination with an orbit averaging technique [178].

As explained in Section 3.2, metaheuristic methods are not employed in this work to solve the OCP directly. However, the GA is used to calculate the shaping parameters in the pseudo-equinoctial shape-based method and a methodology is developed which uses a multi-objective GA to conduct the MLT analysis. Thus, the logic of the GA and the multi-objective GA are presented in the following sections, respectively.

Genetic Algorithms

Genetic algorithms are the most widely used metaheuristic methods. The logic behind GAs is inspired by the natural evolution of the species, theorised in Darwin's Theory of Evolution. In essence, a population of candidate solutions, known as *individuals*, naturally evolves toward better solutions in an optimisation problem. Each individual is characterised by a set of properties, known as its *chromosomes*, which can mutate and recombine with chromosomes of the other parent to generate the offspring. In a GA, the initial population is generally randomly generated and, through an iterative process where each iteration is known as a *generation*, the *fitness* of each individual is evaluated. The value of the fitness is used to rank individuals, i.e. the solutions, depending on the objective function in the optimisation problem being solved. The fittest individuals are selected from the population to recombine and form a new generation, following Darwin's "survival of the fittest" theory [167, 179].

The GA performs an iterative process to make the population evolve. The GA evolution process is schematised in Figure 3.5, where each iteration or generation consists of the following steps:

- Evaluation: the fitness of each individual is evaluated.
- Selection: individuals are randomly selected for reproduction, with a probability which is related to the fitness of the individuals so that the fittest ones have more chances to be chosen.
- Crossover: the selected individuals (parents) recombine, or cross over, to generate new offspring (children).

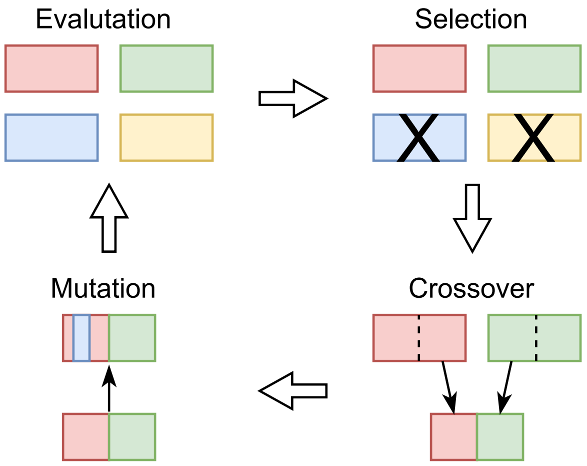


Figure 3.5: GA evolution process.

- Mutation: new offspring mutates with a mutation probability (i.e. some of the original genetic material is randomly changed).

Once the reproduction is completed with the crossover and mutation, the new population is formed and the evaluation of their fitness can be initiated again for a new iteration. Commonly, the algorithm terminates when one of the stopping criteria is met, such as when either a maximum number of generations has been produced, or the population has reached a satisfactory fitness level.

In this work, the MATLAB® built-in GA is used. The algorithm can determine the values of the design variables to achieve the minimum of the objective function, given the upper and lower bounds on the design variables (so that a solution can be found within the desired search space). For further details on how the MATLAB® built-in GA is implemented, the interested reader can refer to the MATLAB® Help Centre [180, 181]. As mentioned, the GA is used within this thesis to calculate the shaping parameters in the shape-based method (described in Section 3.2.2).

Multi-Objective Genetic Algorithm

The optimisation problems which have been discussed until now are single-objective optimisation problems, where a single objective function is involved and a single solution, which is the *optimal solution*, can be identified. However, there exist problems which require to consider more than one objective simultaneously. These are known as multi-objective optimisation problems.

The need for multiple objectives to be considered in an optimisation arises for some problems when different objectives are non-commensurable, thus difficult to combine into a single goal function a priori, that is, before the alternatives (or solutions) are available. For example, in the problem for the optimisation of low-thrust systems for multi-target missions, two objectives of the optimisation can be

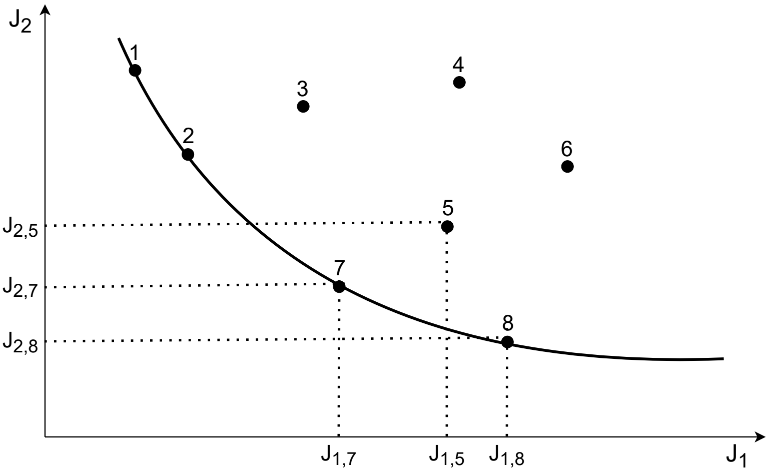


Figure 3.6: Non-dominated front or Pareto front.

defined, i.e. the initial mass of the spacecraft and the TOF. Having different unit of measure, they cannot be combined in a single fitness function without losing information about each objective. In other words, attempting in combining the objectives by specifying the constraints, before knowing the alternatives, may result in the feasible region of solutions to become empty, thus making the optimisation problem impossible to solve.

The solution to this kind of problems is generally not a single optimal solution, but a set of multiple alternatives, called *multiple Pareto-optimal solutions* or *non-dominated solutions*, which are characterised by different trade-offs between the objectives.

Considering the single-objective optimisation problem described in Eqs.(3.31) to (3.36), the multi-objective optimisation problem can be summarised as:

$$\text{Minimise } \; J_m(\mathbf{x}) \quad \text{with } \; m = 1, ..., M \tag{3.56}$$

subject to the dynamic constraints, boundary constraints and path constraints. The solution $\mathbf{x} \in \mathbf{R}^n$ is a vector of n decision variables, $\mathbf{x} = (x_1, ..., x_n)^T$. The solutions, which satisfy the constraints, constitute the *decision variable space*, $S \subset \mathbf{R}^n$, while the objective functions J_m constitute the *objective space*, $Z \subset \mathbf{R}^M$. Thus, for each solution $\mathbf{x}_i$ in the decision variable space, a point $\mathbf{J}(\mathbf{x}_i) = (\mathbf{J}_1(\mathbf{x}_i), ..., \mathbf{J}_M(\mathbf{x}_i))^T$ exists in the objective space [182, 183].

A solution $\mathbf{x}^{(1)}$ is considered to be a Pareto optimal solution if it is *non-dominated* by any other feasible solution. In particular, a solution $\mathbf{x}^{(1)}$ is said to dominate a solution $\mathbf{x}^{(2)}$ if $\mathbf{x}^{(1)}$ is not worse than $\mathbf{x}^{(2)}$ in all the objectives and if $\mathbf{x}^{(1)}$ is strictly better than $\mathbf{x}^{(2)}$ in at least one objective [184, 185]. Figure 3.6 gives a representation of a set of solutions, where the non-dominated solutions (solutions 1, 2, 7, 8) define the *non-dominated front* or *Pareto front*. As an example, point 5 is not on the Pareto front because it is dominated by both point 7 and point 8. Points 7 and 8 are not strictly dominated by any other solution, hence lying on the Pareto front.

Although multiple Pareto-optimal solutions are obtained, one of them is to be chosen. Hence, solving a multi-objective optimisation problem involves two steps: (i) an optimisation problem solver to identify the Pareto solutions and (ii) a decision-making task to select the most preferred solution [186, 183]. It should be noted that the decision-making task is independent of the OCP and requires some other factors to be taken into account. As an example, for multi-target missions, the requirements and/or goals of the given mission can be considered by the mission designer in the solution selection process.

3.3 Artificial Neural Networks

Thanks to the improved computational power of CPUs and the advancement in the research on how to train neural networks efficiently, the interest of the scientific community for AI and, specifically, for neural networks has grown in the recent years.

Artificial neural networks are designed with the intent of modelling the information processing capabilities of biological nervous systems. Therefore, to understand the general principle of ANNs, the first question to clarify is their similarities with the animal neural models.

Animal nervous systems are characterised by millions of interconnected cells. Each of these cells constitutes a very complex system dealing with incoming signals and outgoing signals in different ways. Interestingly, requiring some milliseconds to react to a stimulus, brain neurons are slower than artificial computing units, which only need a few nanoseconds. Nevertheless, the brain can solve problems which no artificial computer can yet deal with efficiently. The experts agree that the key element of natural neural systems is their capability to “control through communication” [187].

The biological neural systems are composed by neural cells or neurons, organised in architectures of layers of variable complexity. Each neuron can receive signals and produce an output response. The information is stored at the connection points between neurons, which are called *synapses*, and can be transmitted using action potentials. This is possible thanks to electrical and chemical processes at the synapses, which also determines the direction of the information flow.

Similarly, artificial neural networks are composed by neurons organised in layers. Each neuron can receive input signals and, based on activation functions, generate output signals which can then be transmitted to other neurons thanks to the interconnection pattern between the neural nodes.

Figure 3.7 schematises an artificial neuron with inputs x_i with $i = 1, ..., n$. The

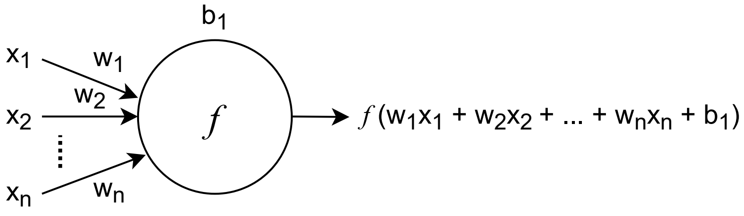


Figure 3.7: Artificial neuron.

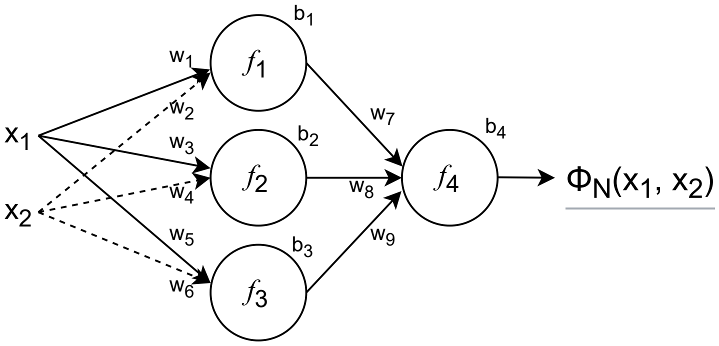


Figure 3.8: Artificial neural network.

input channels have a weight w_i associated, which the input is multiplied for, and a bias b can be associated to the neuron. The neuron integrates the input signals by adding them together, add the bias value and calculate the activation function f. This defines the neuron output [129].

An abstract representation of a neural network is presented in Figure 3.8. In essence, the neural network implements a function Φ_N, which is known as *network function*. The function Φ_N is a result of the inputs x_i ($i = 1,2$ in the example represented in the figure), the activation functions at the neurons f_j ($j = 1,...,4$ in the figure), the weights of the connections between neurons, w_k ($k = 1,...,9$ in the figure), and the biases of each neuron, b_j. It is important to note that a different selection of the weights and biases generates different network functions.

In order for the network to learn the network function Φ_N, the function Φ_N is given to the network not explicitly but implicitly through a set of input-output samples. This is a key element of the network computing capabilities, because at the moment of the training Φ_N is known only at some points (i.e. the input-output samples) but we want to *generalise* to new points as well as possible. To this end, a learning algorithm is used to optimally adjust the network weights and biases to reflect the information known and to extrapolate to new input patterns (which can be given to the network afterwards) [129].

We can conclude that three factors need to be considered when designing an ANN:

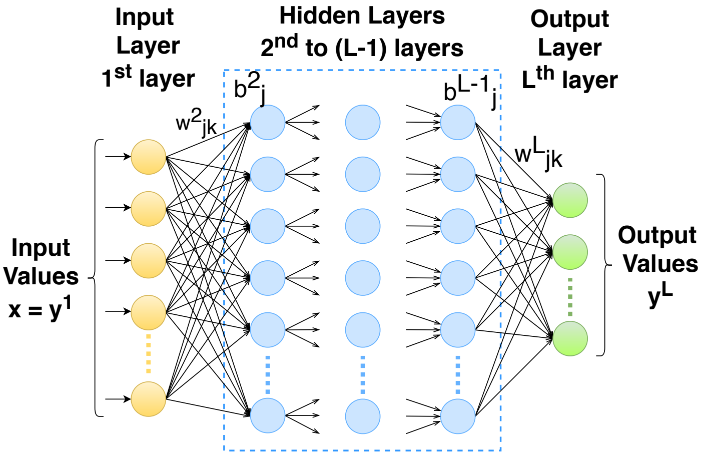


Figure 3.9: A feedforward ANN with L layers.

1. the activation function of the neurons
2. the topology of the network
3. the learning algorithm to calculate the weights and biases.

The first two concepts are analysed in detail in Section 3.3.1, while the learning algorithms and consequent network training is treated in Section 3.3.2.

3.3.1 Network Architecture

Similarly to the brain where neurons are organised hierarchically in a number of layers, the structure of a neural network is organised in layers. Figure 3.9 gives a schematic representation of a network with L layers. The first and last layers correspond to the *input layer* and *output layer*, respectively. In between, *hidden layers* are present. The first layer has inputs $\mathbf{x} = \mathbf{y^1}$, 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 output values $\mathbf{y^L}$. The number of hidden layers and the number of neurons per layer define the topology of the network.

A wide variety of ANNs exists to be functional for many types of applications and satisfy the associated requirements and objectives. The main distinction that can be made is between feedforward and recurrent networks [130]. *Feedforward ANNs* are networks where all the neurons of a layer are directly connected to every neurons of the successive layer. In this case information moves in one direction only, from input layer through the hidden layers to the output layer [117]. Differently, in a *recurrent ANN*, the neurons of a layer can be connected not only to the neurons of the successive layer, but also to the neurons of the preceding layers. This is done to obtain a time-varying behaviour, which can be used for tasks as speech recognition or hand-written recognition [188, 189, 190]. For the design of multi-target missions and, specifically, to provide an estimate of the cost and duration of low-thrust transfers, a feedforward network is sufficient and thus, these are considered within the rest of the study.

The training of the network consists in determining the weights w_{jk}^l associated to each connection between the k-th and j-th neurons and the biases b_j^l of the j-th neuron of the l-th layer. So, the output of the j-th neuron of the l-th layer equals to:

$$y_j^l = f_j^l\left(\sum_k w_{jk}^l y_k^{(l-1)} + b_j^l\right) \tag{3.57}$$

where subscripts and superscripts identify the neuron and hidden layer, respectively, f_j^l is the *activation function* of the j-th neuron of the l-th layer, and $y_k^{(l-1)}$ is the output of the k-th neuron of the previous $(l-1)$-th layer.

Depending on the objectives of the network, the activation function can be chosen so that the desired performance is obtained. For different activation functions the network can be used for *classification* or *regression* problems. Table 3.4 summarises the more commonly-used activation functions with the corresponding mathematical expressions and graphical representations. To estimate the cost and duration of optimal low-thrust trajectories, which is a function approximation problem (i.e. a *regression* problem), multi-layer networks generally use the *log-sigmoid* or *tan-sigmoid* functions in the hidden layers, and the *linear* or *rectified linear unit* (ReLU) functions in the output layer [129]. The log-sigmoid activation function (commonly referred to as *sigmoid*) generates outputs between 0 and 1 with inputs going from minus to plus infinity. In the tan-sigmoid activation function the outputs are comprised between -1 and +1 with inputs going from minus to plus infinity. The ReLU activation function returns 0 for negative input values, while it returns the values passed through it for positive input values. The linear activation function returns the values passed through it.

For example, if the sigmoid activation function is selected for the j-th neuron of the l-th layer, then the output of the j-th neuron of the l-th layer can be computed by substituting the sigmoid expression from Table 3.4 into Eq.(3.57), as follows:

$$y_j^l = \frac{1}{1 - e^{-\left(\sum_k w_{jk}^l y_k^{(l-1)} + b_j^l\right)}} \tag{3.58}$$

with $y_j^l \subseteq (0,1)$. This is different for the first layer (or input layer) where each i-th neuron provides one component of the input vector, $x_i \subseteq \mathrm{I\!R}$. When the sigmoid activation function is used, the network function is a parametrized function that maps from an input set $\mathscr{X}$ to an output set $\mathscr{Y}$ as follows:

$$\mathscr{X} \subseteq \mathrm{I\!R}^{n_i} \rightarrow \mathscr{Y} \subseteq (0,1)^{n_o} \tag{3.59}$$

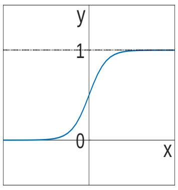
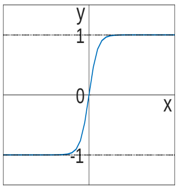
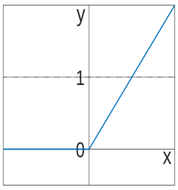
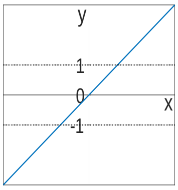
Activation Function	Equation	Plot
Log-sigmoid	$f(x) = \frac{1}{1+e^{-x}}$	
Tan-sigmoid	$f(x) = \frac{2}{1+e^{-2x}} - 1$	
ReLU	$f(x) = \begin{cases} 0, & \text{if } x < 0 \\ x, & \text{if } x \geq 0 \end{cases}$	
Linear	$f(x) = x$	

Table 3.4: Main activation functions generally used for regression problems.

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

Figures 3.10 and 3.11 show the effect of different weights and biases on a neuron output, when a sigmoid activation function is used. For an input x between -10 and 10 (selected for representation purposes), changing the weight essentially changes the steepness of the sigmoid. The highest the weight, the steepest the sigmoid, as shown in Figure 3.10. Differently, a bias value allows to shift the activation function to the left (for negative values of the bias) or right (for positive values of the bias). This is represented in Figure 3.11. Identifying the optimal values of the weights and biases during the training is critical for a successful learning.

The number of layers and neurons per layer can influence the performance of the network. There exists a combination of number of layers and neurons per layer which provides the optimal performance. However, these numbers are not known a priori and they can be obtained by trial and error. It should be noticed that a larger number of neurons and layers will increase the flexibility of the network introducing more weights and biases. As a side effect, which should be considered when designing a

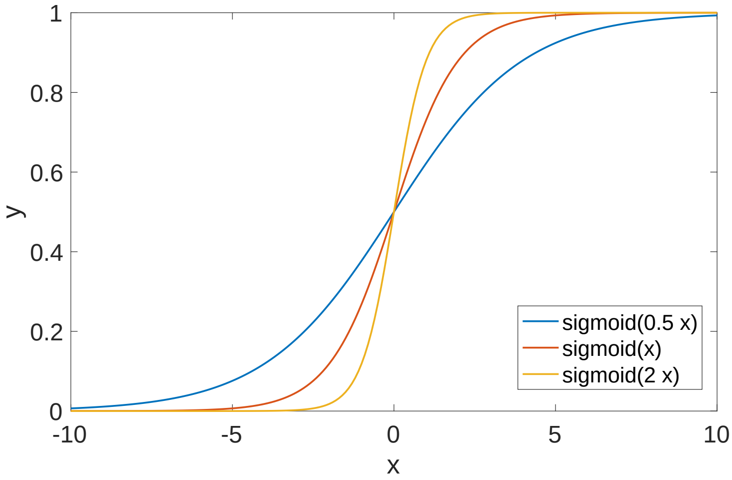


Figure 3.10: Effect of the weights on the neuron output.

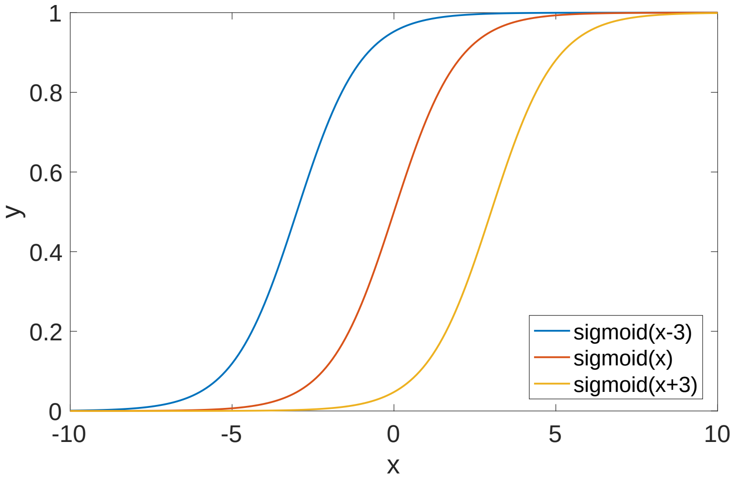


Figure 3.11: Effect of the biases on the neuron output.

neural network, more flexibility can induce to an overfitting of the data, which is an indicator of a bad generalisation of the network function, as explained in the next section [35].

3.3.2 Network Training

In theory, any function can be approximated by an ANN with a adequate number of neurons and layers. To find the network function, the network needs to be trained. To this end, a database of a number of samples containing the inputs and desired outputs shall be provided to the network. A learning algorithm is used to train the network with the given samples, i.e. to optimise the weights and biases of the network to replicate the information of the training samples and generalise to new inputs. To this end, the *network error* between the network outputs (which the ANN computes for the sample inputs) and the desired sample outputs is used to estimate the performance of the network. During the training, the learning algorithm aims at minimising the network error, which means improving the accuracy of the network [191, 129].

The weights and biases are generally initialised randomly. The mean squared error C_{MSE} is generally used as an indication of the *cost function*, which the training aims at minimising. The mean squared error (MSE) can be defined as follows:

$$C_{MSE} = \frac{1}{N}\sum_{i=1}^{N} |\mathbf{t}_i - \mathbf{y}_i^L|^2 \tag{3.60}$$

where N is the number of samples in the training data, $\mathbf{t}_i$ is the target output for the i-th sample and $\mathbf{y}_i^L$ is the output generated by the ANN. A *training or learning algorithm* is selected to adjust the weights and biases to minimise the network mean squared error and, ultimately, to improve the performance of the network [129]. This concept is also known as *supervised learning*, which is opposed to *unsupervised learning methods* that do not require training samples but instead exhibit self-organisation that captures patterns as probability densities [192].

To minimise the cost function, the weights and biases have to move in the direction of the steepest descent of the gradient of the function. The gradient can be computed in two different manners. The first one is the *incremental mode*, in which the gradient is computed and the weights and biases are updated after each input data point is given to the network. The second one is the *batch mode*, in which the gradient and the variations of the weights and biases are computed once all the inputs of a batch are given to the network. The batch mode is used in this context, since it results to be generally more efficient in the MATLAB® environment as it is emphasised in its Deep Learning Toolbox™ software [193].

Standard Gradient Descent Algorithm

The simplest training algorithm for ANNs is the standard gradient descent. This algorithm allows to update the weight and bias vector **w** so that at step k it moves in the direction of the negative gradient, i.e. in the direction of the greatest rate of decrease of the error. Thus, the weight and bias vector **w** at step $k+1$ can be formulated as [194]:

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \eta\nabla C_{MSE,k} \tag{3.61}$$

where η is the learning rate and ∇ indicates the gradient of the error at step k. The gradient is evaluated at every step and it is expected that the error decreases at each step.

The gradient descent training algorithm is already implemented in MATLAB® in the *traingd* network training function. Training occurs according to the training parameters, which can be specified as:

- *Epochs*: maximum number of epochs for the training
- *Time*: maximum time of training
- *Goal*: performance goal, i.e., desired value of the network error after training
- *Learning rate*: speed of variation of weights and biases
- *Maximum fail*: maximum number of failures to reduce the validation error, before stopping the training.

Scaled Conjugate Gradient

The scaled conjugate gradient is an advanced version of the gradient descent algorithm. While in the latter, the direction and size of each step are given by, respectively, the local negative gradient of the error function and an arbitrary learning rate; in the scaled conjugate gradient, the minimum of the error function in some search direction in the weight space is calculated and used. Thus, the weight and bias vector **w** at step $k+1$ can be formulated as [195]:

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \gamma_k d_k \tag{3.62}$$

where γ is chosen to minimise the error, i.e. $C_{MSE}(\gamma) = C_{MSE}(\mathbf{w}_k + \gamma_k d_k)$, and d_k describes the search direction in the weight space.

This algorithm is implements in the *trainscg* built-in training function in MATLAB®. The training occurs according to the training parameters, similarly to the standard gradient descent.

Resilient Back-Propagation

The one of the most commonly-used gradient descent algorithms is the *back-propagation* method, whose name reminds of the fact that the error and the corresponding weight and bias variations are computed at the output and propagated backwards through the network's layers. The weight and bias vector **w** at step $k+1$ can be formulated as [196]:

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \delta_w \mathrm{sign}(\nabla C_{MSE,k}) \tag{3.63}$$

where δ_w is initialised to $\delta_{w,0}$ and varies at each iteration so that if the sign of the gradient $\nabla C_{MSE,k}$ changes from one iteration to the next one, then δ_w is decreased by the decrement of weight change δ_{dec}. Differently, when the gradient maintains the same sign, then δ_w is increased by the increment of weight change δ_{inc}.

The *trainrp* network training function implements the gradient descent training algorithm in MATLAB®. Training occurs according to the training parameters, similarly to the standard gradient descent, with the two additional parameters of δ_{dec} and δ_{inc}.

Levenberg-Marquardt Algorithm

The type of gradient descent training algorithms which generally performs better for function approximation is the Levenberg-Marquardt algorithm. It is derived from the *Newton's method*, for which the next-step state of the weight and bias vector **w** can be computed starting from the previous-step state as follows [117]:

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \mathbf{H}_k^{-1}\mathbf{g}_k \tag{3.64}$$

with $\mathbf{H}_k$ being the Hessian matrix of the performance function and $\mathbf{g}_k$ is the gradient of the performance function at the current weight and bias values. The Newton's method generally presents a faster convergence with respect to the standard gradient descent algorithm. However, since the Hessian is complex and expensive to compute and has to be stored at each iteration, it requires a lot of storage.

The *Levenberg-Marquardt algorithm* changes the Newton's method in such a way that it becomes computationally less expensive while maintaining a fast convergence. With a performance function being the sum of the error squares, the Hessian matrix and the gradient, respectively, can be approximated as [197, 198]:

$$\mathbf{H} = \mathbf{J_e}^T\mathbf{J_e} \tag{3.65}$$

$$\mathbf{g} = \mathbf{J_e}^T\mathbf{e} \tag{3.66}$$

where $\mathbf{J_e}$ is the Jacobian matrix of the network error vector **e** with respect to the weights and biases. The Jacobian is less expensive to compute compared to the Hessian matrix. The next-step state vector can be computed introducing the Levenberg-Marquardt approximation of the Hessian matrix and gradient in the Newton's method update:

$$\mathbf{w}_{k+1} = \mathbf{w}_k - [\mathbf{J_e}^T\mathbf{J_e} + \mu\mathbf{I}]^{-1}\mathbf{J_e}^T\mathbf{e} \tag{3.67}$$

where μ is the gradient constant and **I** is the identity matrix. If μ is zero the equation formulates the Newton's method with the Hessian matrix approximation. The higher the μ, the more the method becomes a standard gradient descent with a small step size. The purpose of the algorithm is to move as quickly as possible towards the Newton's method, since it is faster and more accurate. To this end, μ is reduced after each successful step where the cost function is reduced, and vice versa.

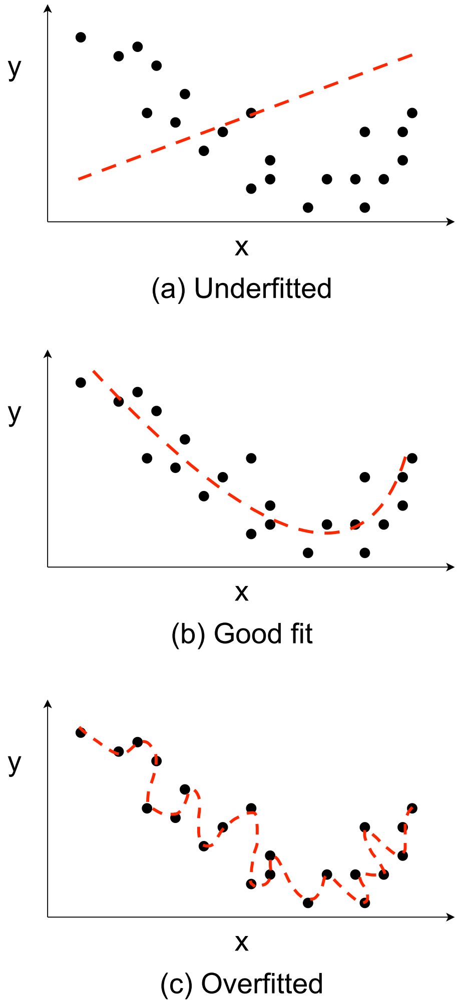
The Levenberg-Marquardt training algorithm is generally employed to train feedforward neural networks of moderate size (up to several hundred weights). It is the fastest training technique for such applications and it is also implemented in MATLAB® in the *trainlm* network training function. Training occurs according to the training parameters, similarly to the standard gradient descent but with an additional parameter of the initial value of the gradient constant, μ.

A well-trained network is characterised by the generalisation property, i.e. it is able to produce accurate outputs also for new inputs that are not encountered during the training. In other words, the network is able to generalise the network function to new situations.

One of the most common problems of training a neural network is the *overfitting*. To properly understand overfitting and how it can occur, the concepts of underfitting and good fitting should be also introduced. Figure 3.12 shows an examples of (a) underfitting, (b) good fitting and (c) overfitting which may result from training the network. In the case of underfitting, the model does not adequately capture the underlying function which relates the data. When overfitting occurs, the network learns the relation between inputs and outputs so well that the error is driven to a very small value. As a consequence, the network is not able to find a proper solution when new inputs are fed into the network. In other words, the network is not able to generalise to new situations. Finally, good fitting is obtained when the model adequately learns the function relating inputs and outputs and generalises well to new, unseen data. The training aims at reaching a good fitting.

Underfitting can occur when the model is too simple (for example, the network does not have an appropriate number of layers or neurons). Differently, overfitting can occur when the network is larger than required or the training is excessive. The generalisation capability is essential for a network to provide an accurate solution in any situation. Thus, in case of overfitting, the network performance can be improved by selecting an appropriate design for the network so that its architecture is large enough for an adequate fitting between network outputs and targets, but not too large to induce overfitting.

There are other methods that can help in this case. One that is particularly suited for large training datasets and a fast convergence is the *early stopping* method [199, 200]. It consists in dividing the available dataset in three subsets: training set, validation set and test set. The training set is used to train the network and obtain the weights and biases which minimise the mean squared error. The validation set is not used to compute the weights and biases, instead it is used after each epoch of the training and the error in the validation set is monitored so that the evidence of overfitting can be detected. This validation error generally decreases as the training error decreases in the initial phase of the training, but later on it can start increasing if the network overfits the data. The training is stopped when the validation error continues to increase after a defined number of successive training epochs. In this case, the weights and biases are set equal to those obtained at the epoch where the



(a) Underfitted

(b) Good fit

(c) Overfitted

Figure 3.12: Examples of (a) underfitting, (b) good fitting and (c) overfitting.

minimum of the validation error is achieved, i.e. the point of optimal performance, as shown in Figure 3.13. The test set is used after the training process is completed to evaluate the overall performance of the network with totally new situations.

There does not exist a unique way to divide the data points into the three subsets and the right split is generally identified by trail and error. Generally, a good indication of a proper division of the dataset in training set, validation set and test set is when the minimum of the error for the validation and test sets is reached at a roughly the same training epoch [129].

Some pre-processing steps can also be performed to make the training more efficient and enable a better generalisation [201]. One important step is the batch normalisation, i.e. to normalise the inputs and target outputs of a batch by introducing the mean and standard deviation of data. The batch normalisation can smooth the objective function, which in turn improves the performance [129]. A batch of the

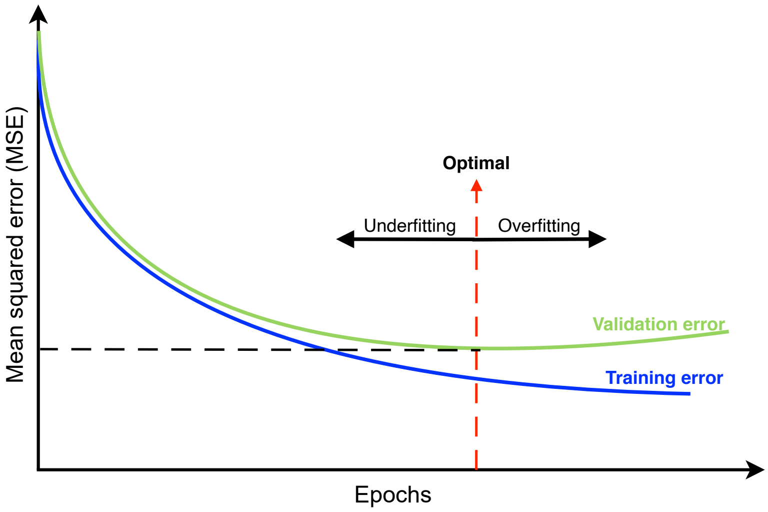


Figure 3.13: Overfitting and early stopping method.

training database can be normalised so that the mean and standard deviation are equal to zero and unity, respectively. Given a batch B of size m samples of the entire training set, the empirical mean μ_B and variance σ_B^2 of B can be calculated as:

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{3.68}$$

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m} \left(x_i - \mu_B\right)^2 \tag{3.69}$$

For a d-dimensional input vector, $\mathbf{x} = (x^{(1)}, ..., x^{(d)})$, each dimension can be normalised as:

$$\hat{x}_i^{(k)} = \frac{x_i^{(k)} - \mu_B^{(k)}}{\sigma_B^{(k)}} \tag{3.70}$$

with $k \in [1, d]$ and $i \in [1, m]$, while $\mu_B^{(k)}$ and $\sigma_B^{(k)}$ are the mean and standard deviation of the k-th dimension of the input vector.

At the end of the simulation, the network outputs can be converted back into the same units of the original target values, by performing an inverse transformation step. In essence, the batch normalisation adds a layer on top of the regular input layer to apply normalisation to every input node of the neural network. Batch normalisation has additional benefits, such as the possibility to use higher learning rates while maintaining the stability of the training algorithm.

Another useful pre-processing step is to eliminate the redundancy in the dataset, in case the input vector is very large and some components are highly correlated. The procedure consists in orthogonalising the components of the input vectors to make them uncorrelated with each other, and eliminating the components which give a minor contribution to the variation in the dataset.

3.4 Sequence Search Algorithm

Given the large number of objects and, consequently, the enormous number of possible permutations between them, identifying the best sequence of targets for both MADR and MNR missions, in terms of minimum time of flight or propellant mass required, is both a continuous and combinatorial problem. Considering that the ANN can solve the continuous part of the global optimisation problem, by estimating the trajectory cost and duration between couples of objects, a sequence search (SS) algorithm can be implemented to solve the combinatorial part. The logic of the SS algorithm to solve the combinatorial optimisation is presented in this section.

The combinatorial optimisation problem aims at minimising (or maximising) an objective function with domain which is a discrete but large configuration space. An example of a typical combinatorial optimisation problem is the travelling salesman problem (TSP). Given a list of cities, the TSP aims at identifying the shortest possible route to visit each city [202]. This constitutes one of the most complex optimisation problems to solve. It is particularly challenging when the number of potential targets to visit is very numerous.

An example of TSP is illustrated in Figure 3.14 for five cities. In the figure, three possible paths are represented: (i) the sequence in black visiting the following cities in order (1,4,2,5,3), (ii) the sequence in blue visiting the following cities in order (1,2,3,5,4), and (iii) the sequence in red visiting the following cities in order (1,4,3,2,5). However, considering five cities, 120 possible permutations are possible and can be investigated. It is worth noting that, the number of possible combinations with permutations between cities increases exponentially with the number of cities.

A cost can be associated to each segment connecting two cities, such as the amount of fuel required to travel from a city to the next one or the duration of each trip. To solve the combinatorial optimisation problem for the TSP, the optimal sequence of cities which allows to minimise the cost needs to be identified.

For multi-target missions, the combinatorial optimisation problem requires to calculate the optimal sequences of space targets which minimises the cost of the mission. For instance, the cost to minimise can be the propellant mass needed to perform the mission, or the duration of the mission. The multi-target mission problem can be considered as an advanced version of the TSP. However, some considerations are necessary.

Compared to the TSP where the city positions are fixed, in multi-target space

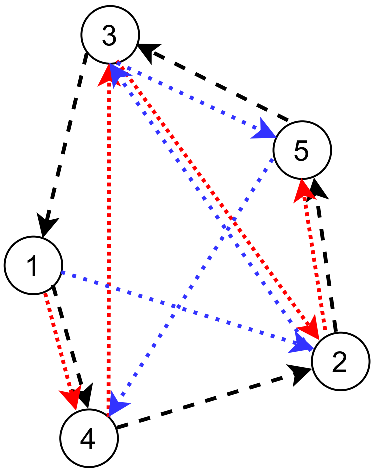


Figure 3.14: Combinatorial optimisation problem: travelling salesman problem.

missions all the objects are moving in time along their orbits, with different velocities. This means that, not only their relative position with respect to the spacecraft is continuously changing, but also the relative positions between the objects is continuously changing. Consequently, also the departure time at each object plays a crucial role in the optimisation process. These considerations contribute to make the optimisation problem even harder than the TSP.

Figure 3.15 shows an example of the combinatorial optimisation problem for multi-target missions where three objects are involved. Also, three epochs in time are represented t_1, t_2 and t_3 (respectively, plots from top to bottom). The three objects illustrated are moving along their orbits with velocity v_1 (blue), v_2 (red) and v_3 (green), respectively. As a result of their orbital dynamics, the three objects are in a different relative position with respect to each other in the three epochs.

Given three objects, six possible permutations could be considered at any given time. In the figure, at epoch t_1, the best sequence identified starts from the object moving at velocity v_1, after which the spacecraft transfers to the object moving at velocity v_2 and then to the object moving at velocity v_3. At epoch t_2, the sequence is the same of the epoch t_1, however since the relative position is changed meanwhile, a different cost would be associate to the sequence. Finally, at epoch t_3, the identified sequence (and associated cost) changes: it starts from the object moving at velocity v_1, but then the spacecraft visits the object moving at velocity v_3 and then the object moving at velocity v_2.

To solve the combinatorial optimisation problem for a multi-target space mission, a tree search method with breadth-first criterion can be adopted. The tree search allows to explore all the possible permutations between the objects, where each node of the tree represents a trajectory and how one proceeds through its branches depends on the mission objectives, e.g. the TOF or propellant consumption minimisation.

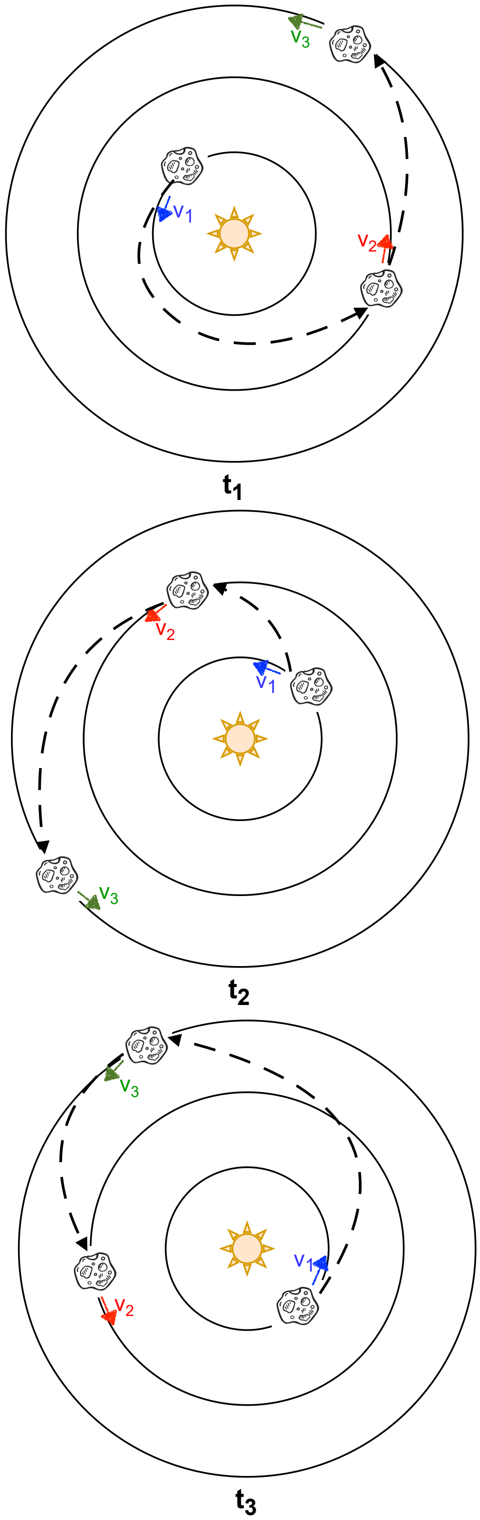
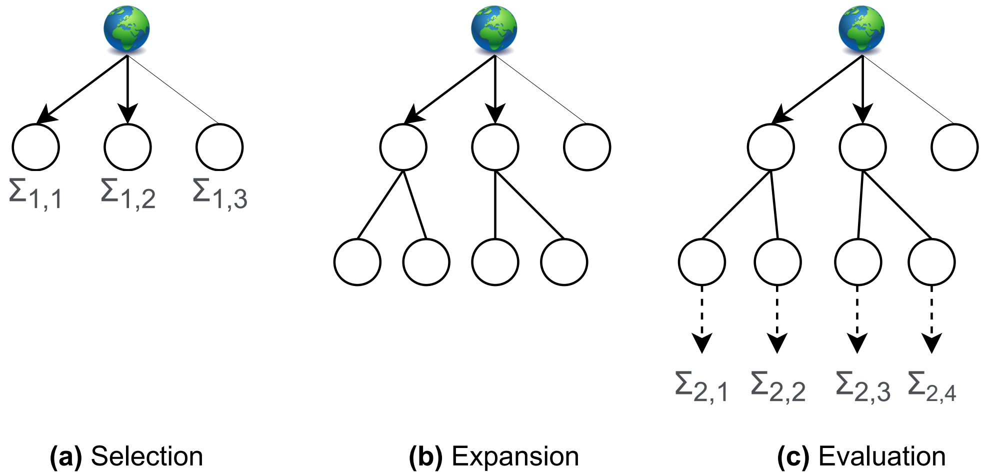


Figure 3.15: Combinatorial optimisation problem: multi-target missions.



(a) Selection **(b)** Expansion **(c)** Evaluation

Figure 3.16: Logic of the tree search and breadth first criterion.

By adopting the breadth-first search criterion, it is possible to search a tree data structure for a node that satisfies a given property. In essence, the algorithm starts at the tree root and explores all nodes at the present depth before moving on to the nodes at the next depth level. In contract, depth-first search explores a node branch as far as possible before backtracking to expand other nodes. Breadth-first criterion is more likely to find a solution node, if one exists, since it is a more systematic and comprehensive search overall [203].

Figure 3.16 shows the logic of the tree search method with breadth first criterion for an example with three objects. Considering the first object is Earth, from which the spacecraft departs, the transfer legs to each one of the objects are calculated and a reward or cost Σ can be associated. This reward or cost Σ can be the propellant mass required or the duration of the transfer, as mentioned. By comparing the rewards obtained, it is possible to select the objects with highest reward for which expanding the tree in depth is more valuable. This corresponds to the *selection* process (a). In the example represented in Figure 3.16, the rewards $\Sigma_{1,1}$ and $\Sigma_{1,2}$ are higher than the reward $\Sigma_{1,3}$, thus the related object is discarded and not considered for expansion in the next depth level. Instead, the legs from the selected objects characterised by the rewards $\Sigma_{1,1}$ and $\Sigma_{1,2}$ to every other body in the database are further investigated, in the *expansion* process (b). The next step corresponds to the *evaluation* process (c), where each node of the newly formed depth level is analysed and the reward computed. At this point, the new rewards are compared again and a new iteration commences.

The *tree policy* is chosen in order to maximise the reward (or minimise the cost) of the mission. The appropriate tree policy may vary depending on the application and can be selected by the mission designer so that the calculated sequences of objects can meet the mission requirements and goals. For the multi-target applications, the tree policy can be to minimise the cost or duration or, also, to maximise the interest value of the objects in the sequence (or a combination of them). In any case, the identification of the most suitable tree policy is essential as it defines how the reward is taken into account to proceed through the branches of the tree.

4

Network Architecture and Parameter Optimisation

The aim of this study is to investigate whether and to what extent ANNs can be used to estimate the cost of (near-)optimal low-thrust trajectories. A trajectory is the result of a steering strategy that minimises the error between the desired and actual state vectors, to depart from the departure orbit and reach the arrival orbit. As a consequence, the cost and duration of the trajectory can be computed. In this chapter, the capability of the network to calculate the cost and duration of low-thrust trajectories between pairs of objects, given the orbital elements of the objects, is investigated.

The prediction accuracy of an ANN, which is evaluated through some network performance indicators (which are defined and presented in Section 4.1), can be affected by many factors. For instance, the size of the sample dataset used for the training can determine whether underfitting, overfitting or good fitting is obtained. Similarly, the type of inputs used, the architecture and hyper-parameters of the network can affect the final ANN performance. Consequently, for the network to achieve a high accuracy, the network architecture and parameters need to be optimised.

Section 4.2 introduces the process to generate the training data and the optimal number of samples to include in the database is identified. The type of network inputs can have an impact on the learned network function which relates the inputs to the outputs and, at the same time, on the efficiency of the network to learn it. The analysis on the types of network inputs and their effect on the network performance is provided in Section 4.3. The algorithm accuracy may also fluctuate considerably because of different configurations of the hyper-parameters, which describe the structure of the network (such as the number of layers and number of neurons), and its learning parameters, such as the learning rate, gradient constant and activation function. Section 4.4 describes the steps followed to optimise the ANN architecture for the multi-target mission application and the results.

As mentioned, two main types of multi-target missions are considered in this study: MADR and MNR. These types of missions present some similarities being part of the same family of applications of multi-target missions. Although there can be some slight differences in terms of the dynamics, the purpose of the optimisation remains the same. The same can be stated for the MLT analysis where the impact of using different propulsion systems on MNR and MADR missions are studied. This is the reason for performing the ANN design optimisation for the MNR problem only, and generalising the results to the MADR and MLT cases. This approach provides accurate results for all the three applications as it will be shown later in the dedicated chapters (Chapter 5 for MADR, Chapters 6 and 7 for MNR and Chapter 8 for MLT). However, when some changes in the ANN design are required for the specific application, these changes will be detailed in the dedicated chapter.

4.1 Network Performance

The goal of training feed-forward ANNs is to find the best function fit for a set of input-output samples. Adjusting the network weights and biases allows to fine-tune the network function so that the optimal configuration can be identified. Although it is preferred that the known inputs are mapped by the network as exactly as possible to the known outputs, the network must be also capable of generalising. This means that the network is expected to compute an accurate output also for yet-unknown inputs, thanks to the learned network function. It follows that a maximum prediction accuracy and a good generalisation can become contradictory objectives as overfitting can occur. In this section, the procedure followed during this research project to evaluate the network performance is presented.

The performance of the network can be assessed by using the final mean squared error of the training, validation and test sets. A definition of the network MSE is provided by Eq.(3.60). By observing how the MSE changes over the training epochs, the effectiveness of the training process can be assessed. It is expected that the MSE decreases at every training epoch until the performance goal in terms of MSE is met. However, when the MSE of the validation set ceases to decrease and, instead, starts increasing for a number of consecutive epochs (which is a sign of overfitting beginning to occur), the training is stopped and the epoch with lowest validation MSE is selected as the one providing the optimal network configuration. A final MSE as close as possible to zero is preferred.

For a complete picture of the network prediction accuracy, the regression analysis between the network output response and the corresponding targets can be performed. This analysis provides information about how well the outputs equate the targets. Figure 4.1 provides an example of a general regression plot, with the target in

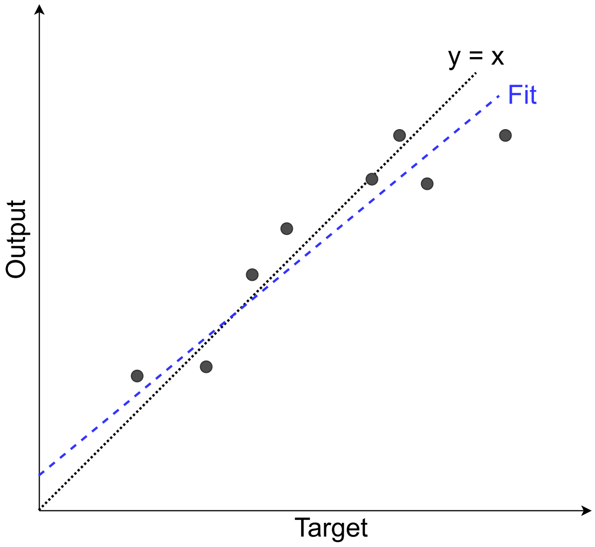


Figure 4.1: Regression plot showing the network fit of the outputs and targets.

the X-axis and ANN output in the Y-axis. The dots in the plot represent the network predictions to the training dataset and the fit is shown by the blue dashed line. A perfect fit is obtained when the slope is equal to 1 and the y-intercept is equal to 0, i.e. when the relationship between the outputs and the targets is perfectly linear ($y = x$). Also, the correlation coefficient R between the outputs and targets expresses the degree to which the network outputs and the targets are linearly related. By definition, the value of the correlation coefficient R can vary from -1 to +1 and can be determined by dividing the covariance of the two variables ($\text{cov}\left(y,x\right)$) by the product of the two variables' standard deviations (σ_y and σ_x), i.e. [22]:

$$R = \frac{\text{cov}\left(y,x\right)}{\sigma_y \sigma_x} \tag{4.1}$$

When a negative (inverse) correlation occurs, i.e. when the correlation coefficient is less than 0, it suggests that one variable increases while the other one decreases. A perfectly negative correlation is obtained when $R = -1$. When R is greater than 0, it signifies a positive correlation which means that output and target vectors move in the same direction. A positive correlation is expected for this application. If the correlation coefficient is equal to 1, there is a perfect fit between the outputs and the targets. Generally, a perfect fit is not achieved during the training of the network since some noise is necessary to ensure generalisation capabilities and avoid overfitting. Consequently, a correlation coefficient as close as possible to 1 is preferred and may indicate a good prediction accuracy.

Moreover, to spot the occurrence of overfitting during the training, the regression analysis is individually conducted for the training set, validation set, test set and, finally, the overall regression can be studied. In essence, in cases where the regression of the validation and test sets is much worse than the one of the training set, additional investigations need to be required to identify the causes and eventually resolve the overfitting.

Finally, another parameter which can be used to estimate the performance of the network is the mean percentage error which can be calculated between the network outputs y_i and targets t_i to new inputs, once the training is completed. The mean percentage error can be calculated as:

$$\mathscr{E}_y = \frac{1}{N}\sum_{i=1}^{N}\frac{|y_i - t_i|}{t_i}\cdot 100 \tag{4.2}$$

which can be calculated for both the ΔV and the time of flight of a low-thrust transfer. This error can also be used as a network performance indicator with respect to the optimisation methods, when the optimal sequences obtained by the SS-ANN methodology are recalculated and optimised. This allows to evaluate the results obtained by the ANN against the optimal values calculated by the optimiser.

4.2 Training Database Generation

Considering the orbital dynamics model describing a low-thrust transfer between two objects as presented in Section 3.1, Section 3.2 introduced some of the most commonly used optimisation techniques to find a solution to the trajectory OCP. These optimisation techniques can identify the optimal transfer and control history, once the initial conditions (the departure orbit and position of the departure body along the orbit), final conditions (the arrival orbit and position of the arrival body along the orbit) and spacecraft specifications (such as the initial mass, available propellant mass and specifics of the propulsion system) are set.

For the preliminary design of multi-target missions, obtaining the full optimal transfer and related control history is very computational expensive to be performed for each permutation of objects and unnecessary at the preliminary design stage of the sequence where the objects to visit are selected. In particular, at this stage it would suffice to obtain a quick estimation of how convenient and appealing a sequence is in terms of, for example, cost (ΔV or propellant mass) and duration $t_{0,f}$ of the mission. These values would allow the sequence search algorithm to quickly scan through the very many permutations of objects and identify the most convenient sequence to follow.

The goal of this PhD research is to investigate whether and how accurately ANNs can approximate the optimisation techniques to provide an estimation of the cost and duration of low-thrust transfers. In essence, optimisation techniques can be

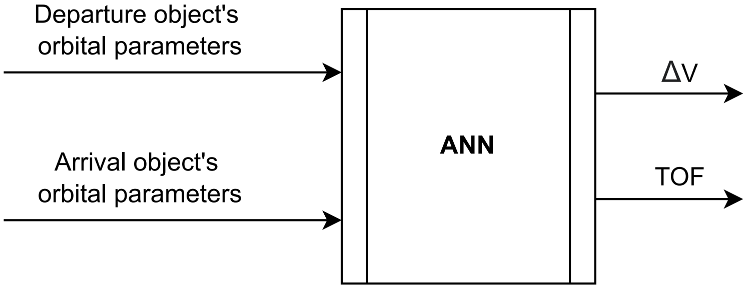

Departure object's orbital parameters
Arrival object's orbital parameters
ANN
ΔV
TOF

Figure 4.2: Schematic of the network inputs and outputs.

used to compute a number of low-thrust transfers which are included in the training database to train the network. Once the training is complete, the ANN should be able to replicate these techniques in computing the transfer cost and TOF given the orbital parameters of the departure and arrival objects as inputs (Figure 4.2). However, the ANN should provide the output in a fraction of the time which would be required for the optimisation technique to run and find a solution. It follows that the process of generating the training database, is particularly important for a successful training of the network.

Figure 4.3 shows the process of the training database generation. Once the database of objects is loaded, the departure and arrival objects are randomly selected from the object database and the selected method to compute the optimal trajectory is run. If a solution exists, then (i) the orbital parameters of the departure and arrival objects and (ii) the cost and duration of the calculated low-thrust transfer between them are stored as a sample in the training database. The procedure is repeated until N_{max} samples are calculated.

The optimal number of samples to include in the training database, N_{max}, needs to be identified. This may effect the prediction accuracy of the ANN model, but also the computational time required to generate the training database. To conduct this analysis, an ANN is trained with databases of different sizes and the results are compared. The architecture of the network is initially fixed and chosen to be similar to the one used by Mereta et al. [34], which demonstrated to perform well to determine the initial guesses of optimal low-thrust transfers between NEAs. This architecture presents two hidden layers and 80 neurons per hidden layer. The sigmoid function is used as activation function of the hidden layers and the linear activation function is used for the output layer. The standard gradient-descent algorithm is adopted for the training. The learning rate is set to 0.01, and the database is divided in 70% training set, 15% validation set, and 15% test set. This network configuration (Table 4.1) is referred to as the *default network configuration* in the proceeding of this thesis. For all the ANN training performed in this PhD thesis, the early stopping method is employed and the pre-processing steps are performed, i.e. the batch normalisation and redundancy elimination, as presented in Section 3.3.

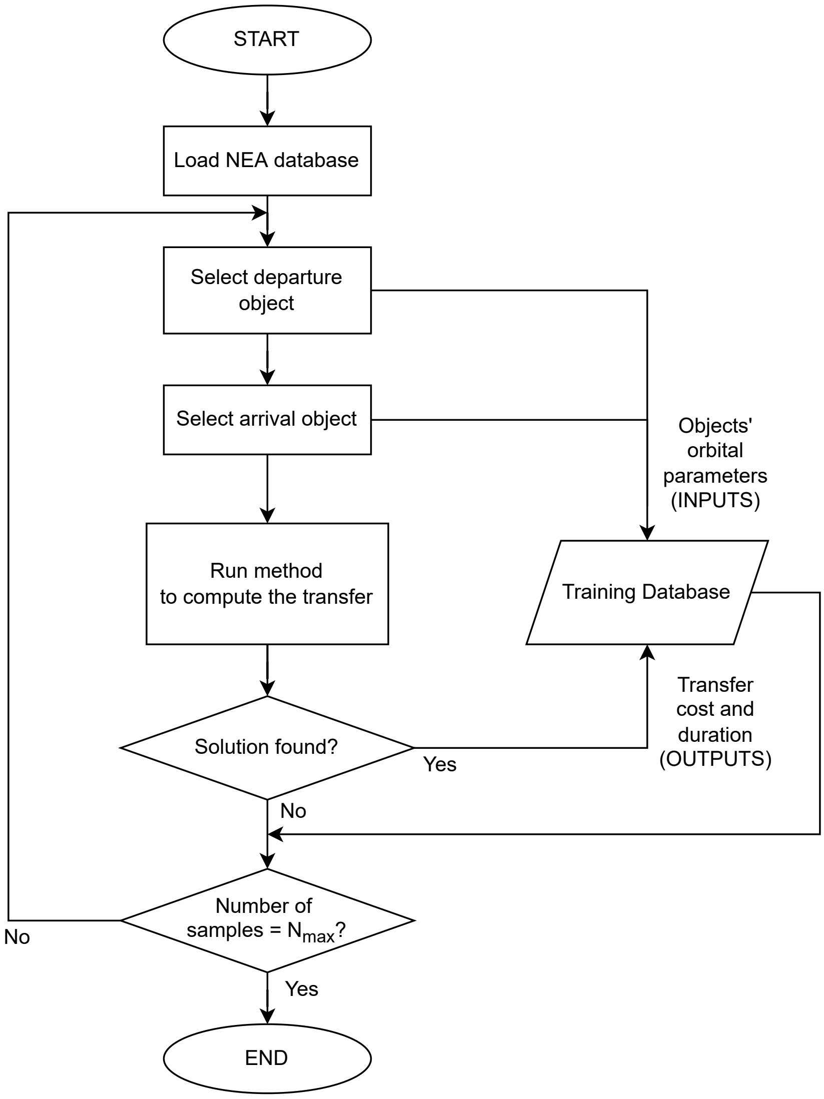

START
Load NEA database
Select departure object
Select arrival object
Run method to compute the transfer
Solution found?
Yes
No
Number of samples = N_{max}?
No
Yes
END
Training Database
Objects' orbital parameters (INPUTS)
Transfer cost and duration (OUTPUTS)

Figure 4.3: Process for the training database generation.

As the methods to compute the MADR transfers and MNR transfers may differ (as it will be further explained in Sections 5.1 and 6.1), the analysis to compute the optimal number of samples to include in the training database is conducted for both MADR and MNR cases. For the MADR case, a number of debris objects from 50 to 500 is considered to build training databases of different sizes (or number of samples) so that the optimal number of samples N_{max} can be identified. The objects are permuted so that transfers from an object to any other object are considered, which means that the number of samples included in the database is equal to the number of objects times the number of objects minus one (as transfers to the same object are obviously excluded).

Figure 4.4 shows the results of the network performance after the training is com-

Table 4.1: Default values for the network hyper-parameters.

ANN Parameter	Default
Number of hidden layers	2
Number of neurons	80
Learning algorithm	Gradient descent
Activation function	sigmoid

pleted with training databases of different sizes and of the time required to generate these databases as a function of the number of objects included in the MADR database generation process. The network performance (top plot), represented by both the overall correlation coefficient R (blue) and the validation MSE (red) in the top plot, improves as the number of objects increases. However, for a number of objects larger than 300, the ANN model starts overfitting resulting in a reduction of the final performance. As mentioned, this can happen when the database is too big and the training is too extensive. In particular, it can be noted that a database generated with 300 objects provides an overall correlation which is slightly smaller than the one obtained with a database generated with 350 objects. However, the validation MSE is inferior in the latter case suggesting that the overfitting starts occurring when 350 objects are considered. Consequently, 300 objects are selected as the optimal number of objects to include in the training database generation process for the MADR case, i.e. optimal number of samples in the training database is equal to $N_{max} = 300 \times 299 = 89,700$.

The time required to generate the MADR database, which is presented in the bottom plot of Figure 4.4, increases exponentially as the number of objects grows. The model used for MADR trajectories integrates the orbital dynamics (described in Section 3.1) of the chaser and space debris from the departure state to the arrival state. The MADR model is further detailed in Section 5.1. It is considered that one run of the MADR model to calculate a transfer between debris objects takes on average 20 seconds when a machine with a 3.4 GHz Intel Core i7 processor is used. It results that, given 300 debris objects, around 21 days are required to generate a database of 89,700 low-thrust transfers.

For the MNR case, a number of NEAs from 25 to 200 is considered to build training databases of different sizes (i.e. different number of samples) so that the optimal number of samples N_{max} can be identified. Similarly to the MADR case, all the permutations between these objects are calculated to generate a training database. The performance of the ANNs when trained with these databases with different sizes and the time needed to generate them are plotted in Figure 4.5, as a function of the number of NEAs included in the database generation process. From the top plot, it is possible to observe that the network performance improves in both overall correlation R and the validation MSE as the number of objects grows; in particular, there is a significant improvement going from 25 to 100 NEAs and for larger number

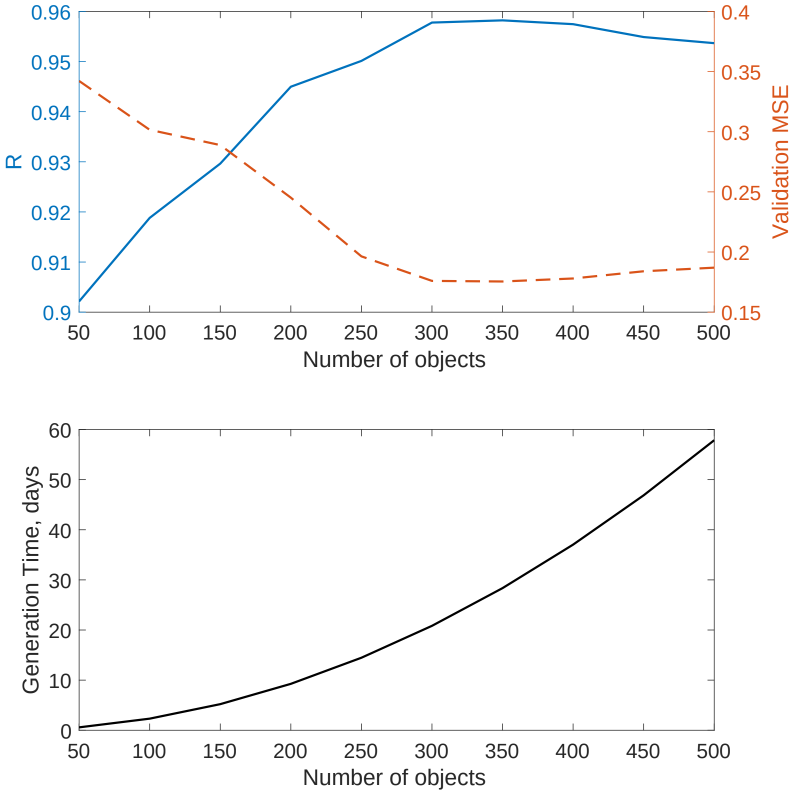


Figure 4.4: MADR missions: effect of number of objects included in the training database generation process on the ANN performance (top) and database generation time (bottom).

of NEAs the performance continues to improve but only slightly. This suggests that no overfitting occurs in this case for the number of objects considered for this analysis. However, it is expected that, for a higher number of objects, overfitting can occur as the training becomes too extensive, similarly to the case of MADR.

The generation time for the MNR database increases exponentially with the number of objects, as shown in the bottom plot of Figure 4.5. For MNR, the model employed to generate a low-thrust transfer between NEAs to include in the training database is the pseudo-equinoctial shape-based model, presented in Section 3.2.2 and further detailed in Section 6.1. One run of the model takes on average 4 minutes, considering that a successful run takes from about 1 to 4 minutes and an unsuccessful run may take up to 10 minutes, when the run is performed by a machine with a 3.4 GHz Intel Core i7 processor. This is linked to the fact that to generate the MNR transfers, the shape-based method uses a GA to calculate the (sub-)optimal trajectory. In this case, increasing the number of samples comes at a cost of an extensive database generation time, which has to be justified by a network performance gain.

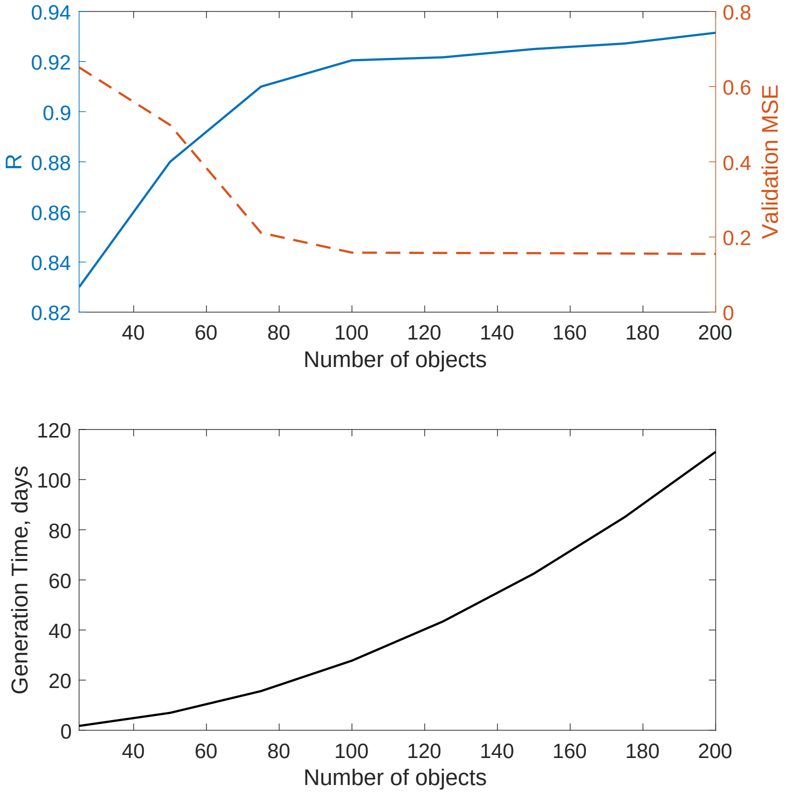


Figure 4.5: MNR missions: effect of number of objects included in the training database generation process on the ANN performance (top) and database generation time (bottom).

For these reasons, 100 objects are selected to generate the MNR training database. This means that a total of $N_{max} = 101 \times 100 = 10,100$ samples are included in the training database, considering the Earth as an additional 'object' within the database. The generation of the identified number of samples requires about 28 days.

A similar approach is applied to identify the optimal number of samples N_{max} to train the ANN for the MLT analysis. However, in this case, in addition to the parameters describing the departure and arrival objects, some additional inputs need to be considered to characterise the low-thrust propulsion system. As the same transfer can be calculated with different combinations of propulsion characteristics, instead of looking at the number of objects to permute, the number of samples is directly examined in this case.

Figure 4.6 shows the results of the network performance after the training is completed with training databases of different sizes for the MLT application as a function of the number of samples included. The MADR case is presented in the top plot, while MNR case is presented in the bottom plot. It is noted that the behaviour is very

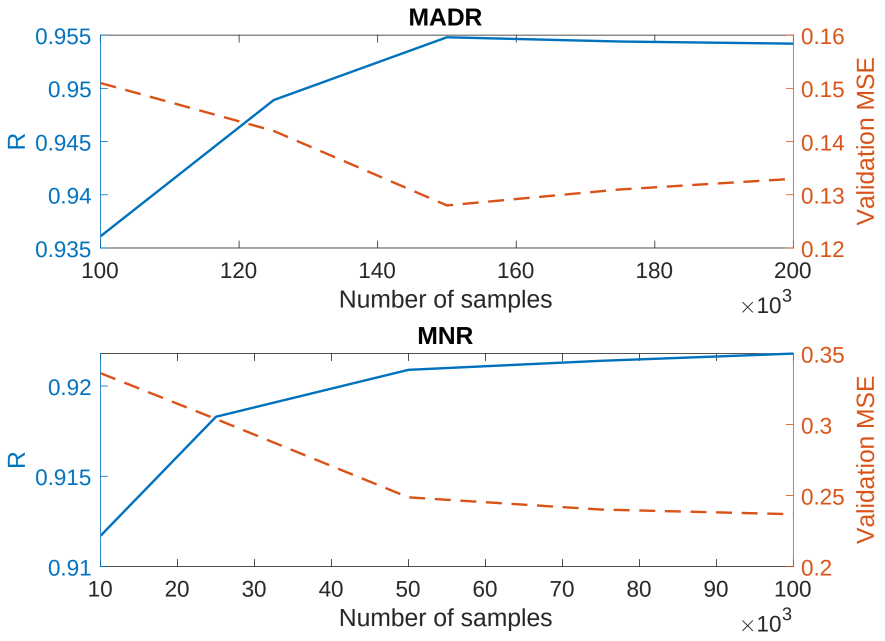


Figure 4.6: Effect of number of MLT samples on the ANN performance for MADR (top) and MNR (bottom) transfers.

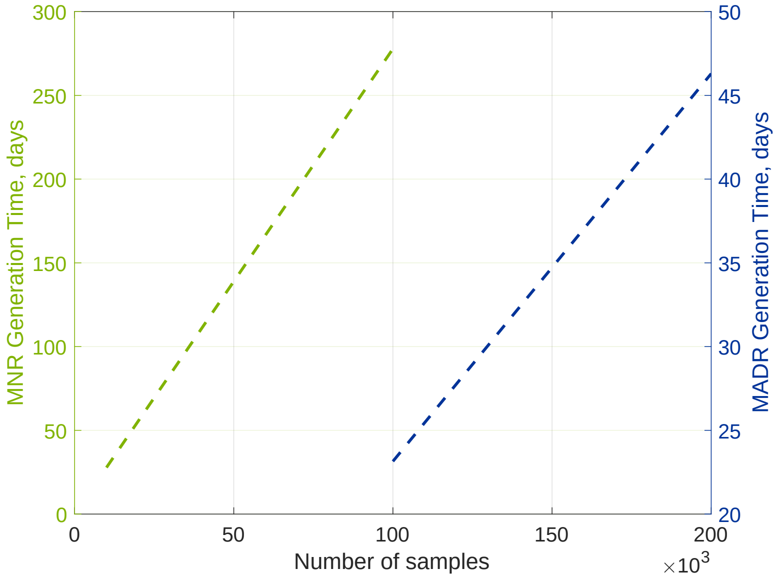


Figure 4.7: Effect of number of MLT samples on the database generation time for MADR (blue) and MNR (green).

similar to what was observed in Figure 4.4 for MADR transfer samples and in Figure 4.5 for MNR transfer samples. In the former, the ANN performance at 150,000 samples reaches a peak before starting overfitting. In the latter, the ANN performance continues to improve while increasing the number of samples, but after 50,000 samples the improvements become modest.

Figure 4.7 illustrates the time required to generate these MLT databases as a function of the number of samples, for the MADR case (blue) and MNR case (green). In both cases, the generation time increases linearly with the number of samples. It is evident from the plot that MNR transfers require considerably more time to be computed. In conclusion, for the MLT analysis with MADR transfers, the optimal number of samples N_{max} = 150,000 is selected to avoid overfitting. For the MLT analysis with MNR transfers, the optimal number of samples N_{max} = 50,000 is selected, to limit the generation time and, at the same time, realise a sufficient gain in ANN performance.

For the simulation to generate the databases, access to the University of Glasgow Computer Cluster Facility is obtained, which uses a machine with 2.3 GHz Core AMD Opteron 6376 with up to 8 GB of RAM per core and runs CentOS 6 [204]. This allowed the many samples to be split in multiple batches which were run on different nodes of the machine, so that the database could be generated quicker.

4.3 Network Inputs

To obtain the cost or ΔV and TOF of a low-thrust transfer between pairs of objects as the network output, i.e. $\mathbf{y} = [\Delta V, t_{0,f}]^T$, the inputs to the network need to describe completely the initial and final conditions. It follows that a study of the parametrisations of the departure and arrival objects' orbits is required. In cases where the spacecraft specifications (such as the spacecraft mass and the characteristics of the propulsion system) are selected and fixed, they do not need to be included within the network inputs. In fact, they will be part of the complex function which the ANN learns during the training. Differently, when different capabilities of the low-thrust propulsion systems are considered, such as in the MLT analysis, the spacecraft characteristics are also subject to the optimisation and need to be included within the network inputs. In the following of this section, the case with fixed spacecraft specifications is considered, so that the investigation can be focused on the effect of using different parameterisations of the orbits as network input on the network performance.

The input vector is required to define completely the departure and arrival orbits and the phasing of the departure and arrival body. To this end the orbital parameters can be used. However, several orbit parametrisations exist, among these [205, 24, 206, 207]:

- **COE**

$$\mathbf{x} = [a_0,\ e_0,\ i_0,\ \Omega_0,\ \omega_0,\ M_0,\ a_f,\ e_f,\ i_f,\ \Omega_f,\ \omega_f,\ M_f] \tag{4.3}$$

with the semimajor axis a, the eccentricity e, the inclination i, the RAAN Ω, the argument of perigee ω, and the mean anomaly M describing the departure orbit (subscript 0) and arrival orbit (subscript f). The same subscripts are used in the following of this section. A more detailed description of the COE is provided in Section 3.1.

- **Equinoctial elements (EE)**

$$\mathbf{x} = [L_0,\ l_0,\ h_0,\ k_0,\ \chi_0,\ \psi_0,\ L_f,\ l_f,\ h_f,\ k_f,\ \chi_f,\ \psi_f] \tag{4.4}$$

which, given the COE, can be defined as:

$$L = \sqrt{\mu a} \tag{4.5a}$$

$$l = M \tag{4.5b}$$

$$h = e\sin(\omega) \tag{4.5c}$$

$$k = e\cos(\omega) \tag{4.5d}$$

$$\chi = \tan\left(\frac{i}{2}\right)\sin(\Omega) \tag{4.5e}$$

$$\psi = \tan\left(\frac{i}{2}\right)\cos(\Omega) \tag{4.5f}$$

- **MEE**

$$\mathbf{x} = [p_0,\ f_0,\ g_0,\ h_0,\ k_0,\ L_0,\ p_f,\ f_f,\ g_f,\ h_f,\ k_f,\ L_f] \tag{4.6}$$

which are described in more detail in Section 3.1.

- **Cartesian coordinates**

$$\mathbf{x} = [\mathbf{r}_0,\ \mathbf{v}_0,\ \mathbf{r}_f,\ \mathbf{v}_f] \tag{4.7}$$

with **r** being the position vector and **v** being the velocity vector, which can be analytically calculated given the MEE by means of the following expressions:

$$\mathbf{r} = \frac{r}{s^2}\begin{bmatrix} \cos L + \gamma\cos L + 2hk\sin L \\ \sin L - \gamma\sin L + 2hk\cos L \\ 2(h\sin L - k\cos L) \end{bmatrix} \tag{4.8}$$

$$\mathbf{v} = -\frac{1}{s^2}\sqrt{\frac{\mu}{p}}\begin{bmatrix} \sin L + \gamma \sin L - 2hk\cos L + g - 2fhk + \gamma g \\ -\cos L + \gamma\cos L + 2hk\sin L - f + 2ghk + \gamma f \\ -2(h\cos L + k\sin L + fh + gk) \end{bmatrix} \tag{4.9}$$

where:

$$q = 1 + f\cos L + g\sin L \tag{4.10}$$

$$r = p/q \tag{4.11}$$

$$\gamma = h^2 - k^2 \tag{4.12}$$

$$s^2 = 1 + h^2 + k^2 \tag{4.13}$$

- **Delaunay elements**

$$\mathbf{x} = [L_0,\ G_0,\ H_0,\ l_0,\ g_0,\ h_0,\ L_f,\ G_f,\ H_f,\ l_f,\ g_f,\ h_f] \tag{4.14}$$

which, given the COE, can be computed as:

$$L = \sqrt{\mu a} \tag{4.15a}$$

$$G = L\sqrt{1-e^2} \tag{4.15b}$$

$$H = G\cos i \tag{4.15c}$$

$$l = M \tag{4.15d}$$

$$g = \omega \tag{4.15e}$$

$$h = \Omega \tag{4.15f}$$

- **Eccentricity and angular momentum vector (eH)**

$$\mathbf{x} = [\mathbf{e}_{v,0},\ \mathbf{H}_0,\ \mathbf{e}_{v,f},\ \mathbf{H}_f] \tag{4.16}$$

where, given the position and velocity vectors, **r** and **v**, the eccentricity vector $\mathbf{e}_v$ and angular momentum vector **H** can be calculated, respectively, as:

$$\mathbf{e} = \frac{1}{\mu}\left(\mathbf{v}\times\mathbf{H} - \frac{\mu\mathbf{r}}{r}\right) \tag{4.17a}$$

$$\mathbf{H} = \mathbf{r}\times\mathbf{v} \tag{4.17b}$$

with μ being the gravitational parameter of the central body.

Table 4.2: Network performance for different parametrisations of the orbit.

ANN input	Correlation	Validation MSE
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

As the objects move along their orbits with a certain orbital velocity, their relative position changes continuously. This needs to be accounted for in the network inputs because it affects the position of the departure and arrival points. Consequently, it can considerably impact the cost and duration of the resulting transfer. To consider the phasing of the objects, it is chosen to express the position of the departure and arrival objects along their orbit at a reference time, which is the departure date. By doing so, as a result of learning the network function, the network knows their relative position at the departure epoch and, given the other orbital parameters as inputs and the TOF computed as output, the network can derive their position at the arrival epoch (which is defined by the departure date and TOF).

In this section, the effect of using these different orbit parametrisations as network inputs on the network performance is investigated. It should be noted that, the benefits of using one or the other parametrisation (for instance, in terms of the singularities which may introduce or remove or in terms of the quality of the orbit description) are neglected at this stage of the analysis. Only their effect on the network prediction accuracy is considered. These parametrisations are used as input to an ANN whose architecture is initially set equal to the default network configuration, which was described in Section 4.2.

The performance of the network, in terms of correlation and validation-set MSE, is presented in Table 4.2 for each orbit parametrisation. The highest correlation is obtained when MEE are used as input ($R = 0.925$), which presents also a low validation-set MSE ($C_{MSE,val} = 0.236$). The latter is slightly lower when the eH parametrisation is used, but a poorer correlation is registered in this case. Consequently, the priority is given to the highest correlation since this represents the performance of the network in all the three training, validation and test phases. Additionally, the MEE are also used to describe the dynamics of the system, as detailed in Section 3.1, which makes the choice of this orbit parametrisation coherent and convenient. Thus, for the remaining of this thesis, MEE are used to describe the orbital parameters of the departure and arrival objects for MNR missions as the input to the network.

Table 4.3: Default values and search space for the network hyper-parameters.

ANN Parameter	Default	Search space
Number of hidden layers	2	[2,8]
Number of neurons	80	[40, 100]
Learning algorithm	Gradient Descent	Levenberg-Marquardt Resilient back-propagation Scaled conjugate gradient Gradient descent
Activation function	sigmoid	tansig, sigmoid, ReLu

4.4 Architecture Optimisation

The aim of this section is to find the best values of the network architecture and hyper-parameters. To this end, the response of the network to changing each of these parameters is analysed. The architecture of the network is defined by the number of hidden layers and the number of neurons per layer. Other hyper-parameters that can affect the performance are the learning algorithm, the activation function for each hidden layer, the learning rate or gradient constant and its increase or decrease factor.

To determine the optimal values of the network architecture and hyper-parameters, an optimisation procedure needs to be carried out. In principle, the most systematic option would be to optimise all the parameters at the same time by using, for instance, a genetic algorithm. However, the number of parameters to optimise and the need to train the network at every trial make the computational time extremely extensive. For this reason, one parameter at a time is tuned, while keeping the others fixed. First, the parameter's values are set to their default configuration values [34]. Then, one parameter is varied individually and the effect on the ANN performance is studied. The parameter is then set to the optimal value found, and the next parameter is considered for the same procedure. The default values and search space for each network parameter are detailed in Table 4.3, where the parameters are presented in the same order which is followed to optimise them during this analysis.

As mentioned earlier, the selection of the appropriate number of layers and neurons per each layer is paramount, since this can affect the accuracy of the network approximation of the continuous function after the training [208]. In fact, a larger number of neurons and layers can increase the depth and flexibility of the network introducing more weights and biases. However, at the same time, more flexibility can lead to an overfitting of the data, thus to a bad generalisation of the network function [35].

For the generation of the training database, used for this analysis, the permutations among a subset of NEAs including 100 objects are considered for a total of 10,100

Table 4.4: Effect of different values of the network hyper-parameters on its performance.

Training Algorithm	R	Validation MSE	T_{train} [min]
Levenberg-Marquardt	0.9732	0.1211	59.79
Resilient back-propagation	0.9258	0.1598	2.39
Scaled conjugate gradient	0.9386	0.1467	4.67
Gradient descent	0.9205	0.1584	76.93
Activation Function	R	Validation MSE	T_{train} [min]
tanh	0.9489	0.1609	34.31
sigmoid	0.9732	0.1211	59.79
ReLu	0.9210	0.2295	52.42

transfer samples. The samples are divided in 75% training set, 15% validation set, and 15% testing set. For the analysis, the weights and biases are initialised with the same seed at every evaluation. The effect of changing the number of layers and neurons on the correlation coefficient R (blue), MSE of the validation set (green) and training time T_{train} (red) is shown in Figure 4.8. Increasing the number of layers and neurons improves the network performance up to certain number of layers and neurons (*peak*), after which the performance starts degrading, suggesting that the ANN models start overfitting. As expected, the time required for the training process increases significantly as the depth of the network grows. The highest correlation coefficient and lowest validation-set MSE occur with a number of layers of four and a number of neurons of 80 per hidden layer.

Different training algorithms and activation functions of the hidden layers are studied. As shown in Table 4.4, each of them induces some differences in the accuracy and the training speed. The training algorithms considered for the study are the most commonly used ones for function approximation; in particular, the Levenberg-Marquardt algorithm, the resilient back-propagation algorithm, the scaled conjugate gradient and the standard gradient descent algorithm are investigated. A mathematical description of these algorithms is offered in Section 3.3.2. The Levenberg-Marquardt algorithm offers the best performance at the cost of a larger training time, compared to the other training algorithms analysed. The activation functions analysed are the tangent sigmoid (*tanh*), the logarithmic sigmoid function (*sigmoid*) and the rectified linear unit (*ReLu*) function. It results that using the sigmoid function as the activation function of the hidden layers allows for a better performance overall. It is worth noting that the activation function of the input and output layers is a linear function (Table 3.4).

As described in Section 3.3.2, for the Levenberg-Marquardt training algorithm, the gradient constant μ influences the weight and bias vector $\mathbf{w}$ as follows [197, 198]:

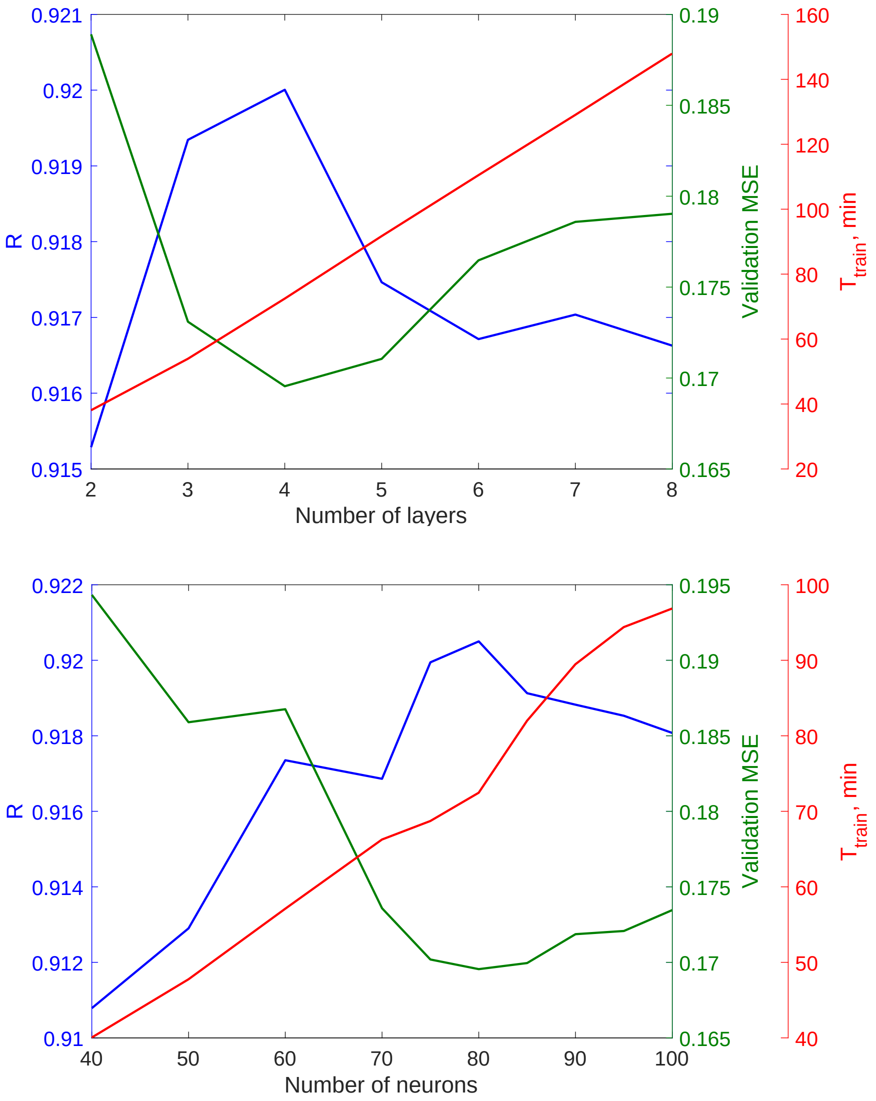


Figure 4.8: Effect of varying the number of layers (top) and neurons (bottom) on the network performance.

$$\mathbf{w}_{k+1} = \mathbf{w}_k - [\mathbf{J_e}^T\mathbf{J_e} + \mu\mathbf{I}]^{-1}\mathbf{J_e}^T\mathbf{e} \tag{4.18}$$

where $\mathbf{J}$ is the Jacobian matrix of the network error vector $\mathbf{e}$ with respect to the weights and biases of the k-th step. The initial value of the gradient constant μ is set to 0.001. When μ is large, the algorithm becomes a gradient descent with small step size. However, after each successful step (i.e. when the cost function is reduced) μ is reduced of a decrease factor μ_{dec}. The closest μ is to zero, the more the algorithm moves towards the Newton's method, which has a faster convergence and is more

Table 4.5: Optimal values for the network hyper-parameters.

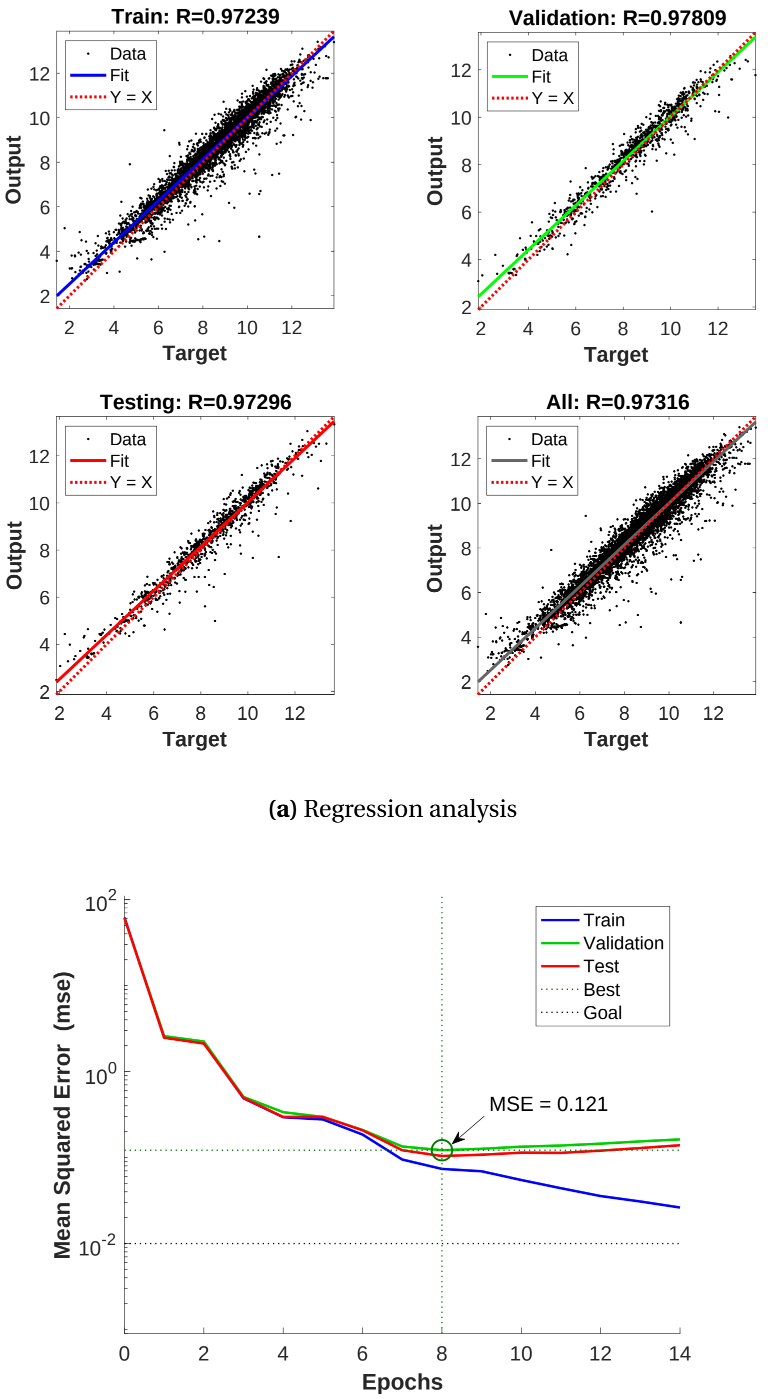
ANN Parameter	Optimal
Number of hidden layers	4
Number of neurons	80
Learning algorithm	Levenberg-Marquardt
Activation function	Sigmoid

accurate (although it requires more storage space). The decrease factor is set equal to 0.1.

From the investigation of the network hyper-parameters, the optimal structure of the network is defined and detailed in Table 4.5. To verify the network performance, the algorithm is run again but this time using the optimal values as the network configuration values. This is referred to as the *optimal network configuration* in the following of this thesis. The test confirms that the obtained values allow the network to achieve the highest performance, with a correlation coefficient of 0.9732 and a validation-set MSE of 0.1211. As mentioned, this analysis to optimise the ANN design is conducted for the MNR problem only, and the results can be generalised to the MADR and MLT cases, as it will be shown later in the dedicated chapters.

Figure 4.9 illustrates the performance of the network with the optimal network configuration. The regression plot in Fig. 4.9(a) presents how well the network outputs (Y-axis) fit the targets (X-axis) with respect to the training, validation, test sets, and all of them combined. Note that the outputs and targets are ΔV and TOF, so their normalised values are presented in the plot. Details on how the normalisation is performed are given in Section 3.3.2 (Eq.(3.70)). It can be noticed that the network fit, represented by the solid lines in the plots, is close to the perfect fit (represented by the dotted red line $y = x$), for all training, validation, test sets and overall. The final overall correlation coefficient R obtained is about 0.9732, which indicates a very accurate fitting.

Figure 4.9(b) shows how the MSE of the training, validation and test sets changes along the training epochs. It can be noticed that, in this case, the training stops before achieving the accuracy goal ($C_{MSE} = 10^{-2}$). This happens because of the early stopping method, i.e. the overfitting starts occurring. It follows that the network weights and biases are set equal to those that provided the lowest MSE for the validation set, which occurs in this case at epoch 8 and is equal to 0.1211. It should be noted that the MSE value is referred to the normalised values of ΔV and TOF. However, to obtain the MSE for ΔV and TOF individually, these can be easily computed by using their dimensional values in Eq.(3.60). This indicates that the network function accurately describes the function to compute the cost and duration of low-thrust transfers, given the orbital parameters of the departure and final objects. Since the correlation coefficient is equal to about 0.97 also for the validation and test sets, it can



(a) Regression analysis

(b) MSE variation during the training

Figure 4.9: Regression analysis (a) and performance (b) of the network with the optimal network configuration.

be deduced that this design of the network allows to generalise well to new inputs, which were not experienced during the training.

It can be concluded that (i) a neural network can, to a first approximation, replace the complex optimisation process required to compute a low-thrust transfer to identify its cost and duration; and (ii) the ANN architecture and hyper-parameters can be designed to achieve the best performance for the application on multi-target missions when a low-thrust propulsion system is used. In particular, by tuning the network inputs and hyper-parameters, the conducted analysis allowed to improve the network performance with respect to the correlation coefficient, which increases from 0.551 to 0.973, and with respect to the validation-set MSE, which decreases from 0.862 to 0.121.

5

Design of Multiple Debris Removal Missions

The proposed methodology, SS-ANN, with the sequence search algorithm and the artificial neural network working together to preliminary design multiple active debris removal missions is detailed in this chapter. The MADR mission scenario is presented in Section 5.1, where the trajectory model and the eclipse model are defined. The same section introduces the optimal RAAN-phasing orbits and the effect of choosing different phasing altitudes, in which the change in RAAN is accomplished, on the cost and duration of the mission. Section 5.2 describes the use of the ANN for MADR missions; in particular, the generation of the MADR training database and the performance of the network, once the training is completed. The sequence search logic is explained in Section 5.3, where also an analysis is provided on the computational time and accuracy of the SS-ANN methodology compared to one of the most common techniques used by the industry. Finally, in Section 5.4 the SS-ANN platform is run and three promising MADR sequences are selected. Their transfers are fully re-optimised considering a near-term low-thrust propulsion system, showing the efficiency and efficacy of SS-ANN in solving MADR missions.

5.1 Mission Scenario

As mentioned in Section 2.1, MADR missions can be broadly divided into two categories: (i) a servicing spacecraft travelling from one piece of debris to the next one and attaching a smaller thruster modules to each debris to de-orbit [56]; (ii) a chaser rendezvousing and de-orbiting pieces of debris in a sequential fashion [57, 58]. The first category has been the most studied so far for MADR [16, 60], but it is not well suited for capture methods requiring a tethered connection. Thus, to use capture methods with a tethered connection, which are among the most tested methods

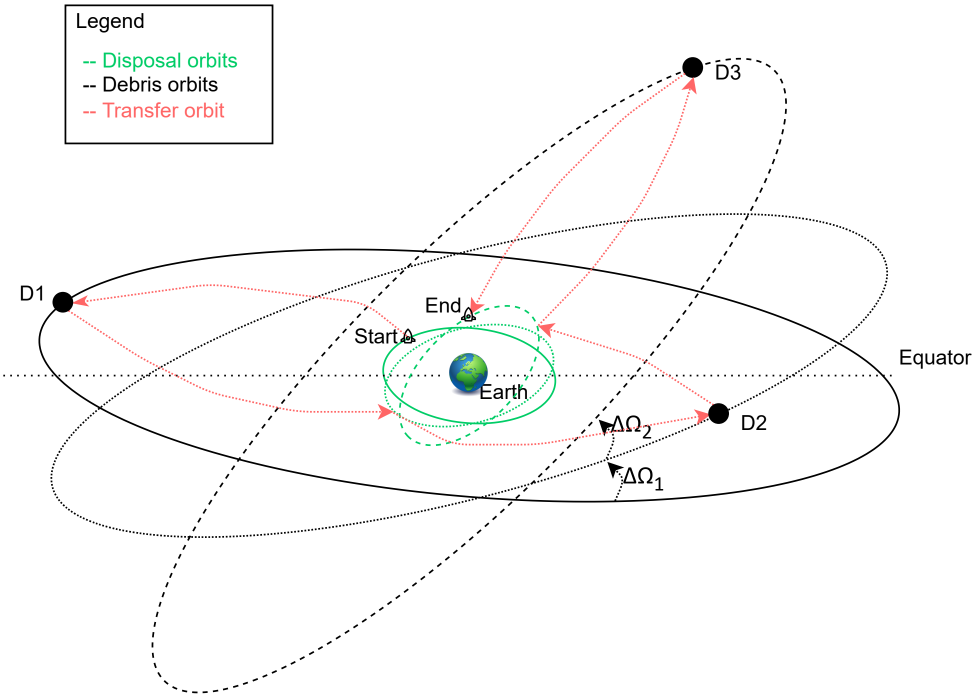


Figure 5.1: Schematic representation of the MADR mission scenario.

for ADR since they allow to make the debris object controllable and stable [61], the second category of MADR missions would be preferable. In this research work, the latter is studied.

A MADR mission requires the chaser to rendezvous and dock with the first debris object in the sequence and descent to a disposal low-Earth orbit, where the object is released for de-orbiting and re-entry before the chaser transfers to the next target object. The procedure repeats until the propellant mass is depleted. Figure 5.1 provides a representation of the mission scenario, where Di with $i = [1,2,...]$ indicates the debris objects located at different altitudes and $\Delta\Omega_i$ is the difference in RAAN between debris orbits.

It is worth noting that, when the debris object is released in a low-altitude disposal orbit, the debris will re-enter by spiralling down due to the atmospheric drag [59]. Re-entries of large debris objects shall be controlled and aiming at uninhabited areas, such as SPOUA. Semi-controlled reentries are currently under study as they could allow de-orbitation with low-thrust in the future.

In the case study of MADR missions, some assumptions are considered to keep the model simple and, consequently, minimise the computational time required to generate the database to train the ANN. Although these assumptions are chosen to be representative of a real MADR mission, they could easily be varied for future use-cases simply by re-training the ANN with the desired model.

This scenario applies to satellite constellations which are on circular orbits, at the same inclination and spaced in RAAN. However, they might be at different altitudes, for example, due to the malfunctioning of a thruster or to the depletion of their fuel and hence start de-orbiting. In this case, the rendezvous transfers between space debris objects require the chaser to match the altitude and the RAAN of the orbit of the arrival body (Debris 2, or D2) at the arrival epoch, when departing from the departure body (Debris 1, or D1). Additionally, the phasing along the orbit between the chaser and debris is neglected in the transfer model. In fact, the low-thrust transfer legs exhibit a large number of revolutions, thus the correct phasing can be attained with minimal propulsive effort, and often with little additional transfer time [209, 210] and can be calculated at a later stage.

To minimise the propellant consumption and guarantee that a larger number of debris can be disposed for the given propellant mass, it is chosen to use the thrust to achieve only the change in altitude and to exploit the Earth's oblateness gravitational perturbation (J_2) to achieve the change in Ω through *RAAN-phasing orbits*. The orbital-averaged RAAN variation rate is given by the Gauss equations as follows [206]:

$$\dot{\Omega} = -\left[\frac{3}{2}\frac{J_2\sqrt{\mu_e}r^2}{a^{7/2}(1-e^2)^2}\cos(i)\right] \tag{5.1}$$

which is experienced by both the chaser and the debris objects.

As shown in Figure 5.1, the mission scenario requires the chaser to rendezvous each debris object and transfer it to the disposal orbit (which is also circular and at the same inclination of the debris objects' orbits). This suggests an iterative procedure, starting from a state where the chaser is docked to D1, where each iteration comprises of:

1. A de-orbiting transfer from D1's orbit to the disposal orbit (with duration T_1), where the object is released
2. A transfer from the disposal orbit to the most convenient RAAN-phasing orbit (with duration $T_{2,a}$)
3. A RAAN-phasing orbit (with duration T_p)
4. A transfer from the phasing orbit to D2's orbit (with duration $T_{2,b}$)
5. A stay time at D2's orbit, for rendezvous and docking operations (with duration T_s)

where $T_{2,a}$ and $T_{2,b}$ can be combined into one transfer of duration T_2 when the convenient RAAN-phasing orbit corresponds to the disposal orbit.

It is worth noting that other options on how to subdivide the entire trajectory into iteration segments were taken into account during the design process. In particular, the two main options to consider an iteration were: (i) from the disposal orbit to the debris orbit and back to the disposal orbit or (ii) from debris Di's orbit to the disposal orbit and then to the successive debris $D(i+1)$'s orbit. The latter option was selected and further developed so that the ANN can compute the cost and duration of the low-thrust transfers while taking into account the differences in altitude and RAAN between two successive debris objects, which could not be possible in case (i) where one debris at a time is considered.

Given the initial RAAN of D1 and D2, i.e. $\Omega_{1,0}$ and $\Omega_{2,0}$, and their RAAN variation rate, $\dot{\Omega}_1$ and $\dot{\Omega}_2$, respectively, it is possible to compute the RAAN of D2, $\Omega_{2,f}$, and of the chaser, $\Omega_{SC,f}$, at time t_f when the chaser is rendezvousing D2:

$$\Omega_{2,f} = \Omega_{2,0} + \dot{\Omega}_2(T_1 + T_2 + T_p) \tag{5.2}$$

$$\Omega_{SC,f} = \Omega_{1,0} + \dot{\Omega}_{T1} T_1 + \dot{\Omega}_{T2} T_2 + \dot{\Omega}_p T_p \tag{5.3}$$

where $T_2 = T_{2,a} + T_{2,b}$, and $\dot{\Omega}_{T1}$, $\dot{\Omega}_{T2}$ and $\dot{\Omega}_p$ are the average RAAN variation rates experienced by the chaser along the transfer segments of duration T_1, T_2 and T_p, respectively. Specifically, $\dot{\Omega}_{T1}$ and $\dot{\Omega}_{T2}$ are calculated at the mean altitude considering the departure and arrival altitudes.

For the chaser to match D2's orbit at time t_f (Figure 5.2, second plot from the top), it is required that:

$$\Omega_{2,f} = \Omega_{SC,f} \tag{5.4}$$

which can be solved for the RAAN-phasing time T_p:

$$T_p = \frac{\Delta\Omega_p}{\dot{\Omega}_2 - \dot{\Omega}_p} \tag{5.5}$$

where $\Delta\Omega_p = \Omega_{1,0} + \dot{\Omega}_{T1} T_1 + \dot{\Omega}_{T2} T_2 - \Omega_{2,0} - \dot{\Omega}_2\,(T_1 + T_2)$.

As an example, Figure 5.2 shows the changes in altitude, RAAN, thrust to realise the change in altitude and mass (plots from top to bottom, respectively) as a function of time for a full transfer from D1 (h_{D_1} = 504.96 km, $\Omega_{D_1,0}$ = 158.72 deg, m_{D_1} = 270.37 kg) to D2 (h_{D_2} = 1077.40 km, $\Omega_{D_2,0}$ = 149.33 deg, m_{D_2} = 156.50 kg). The chaser starts at altitude h_{D_1} with $\Omega_{D_1,0}$, docked with D1, and carries it down to a disposal orbit and releases it. Once completed, the transfer to the next object starts, and chaser transfers to the RAAN-phasing orbit (which in this example coincides with the disposal orbit, thus $T_{2,a} = 0$ s). Once the chaser reaches the required RAAN, it transfers to D2's orbit by matching h_{D_2} and $\Omega_{D_{2,f}}$ (shown by the first subplot and second subplot). Finally, a fixed stay time, T_s = 30 days (to allow for phasing and docking) is considered where

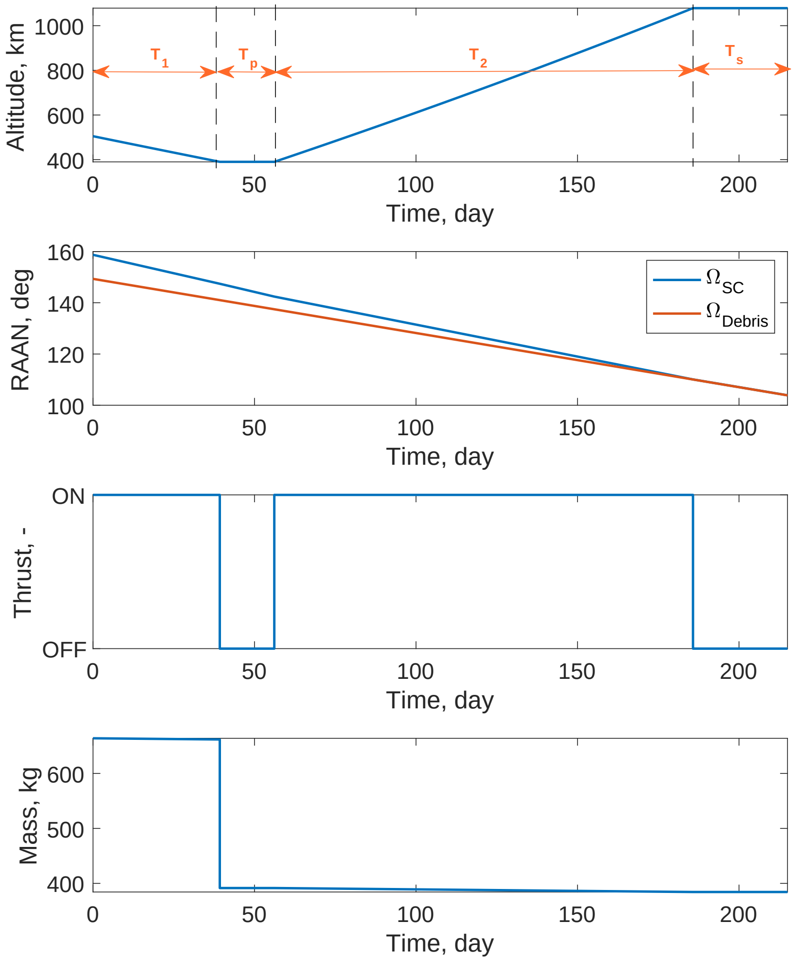


Figure 5.2: Transfer model.

the altitude is fixed and equal to h_{D_2}. Since during T_p and T_s there is no change in altitude, the thrust does not operate in these portions of the transfer, as illustrate in the third subplot. The mass of the system (forth subplot), which comprises of chaser and D1's mass during T_1, drops at the end of T_1 when the debris object is released at the disposal orbit. Over time, the mass decreases due to thrusting during the transfer legs.

As in this model the chaser is continuously thrusting with a maximum tangential thrust, the transfer legs are the fastest the system can perform. Hence, they are time-optimal. In the portions of the transfer at the phasing altitude, where the altitude is fixed and the right RAAN phasing of the next debris to dock is achieved by exploiting

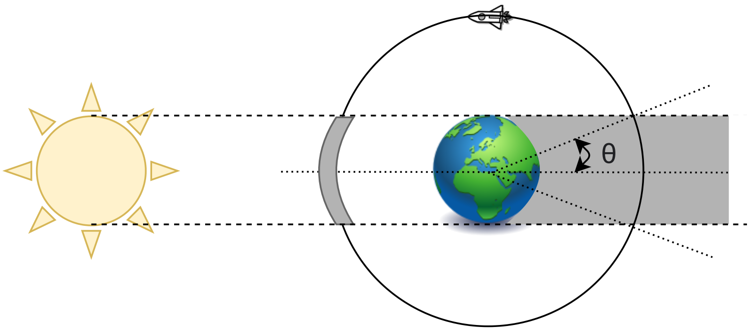


Figure 5.3: Eclipse model.

the J_2 effects, the phasing altitude and duration are optimised as explained in Section 5.1.2.

5.1.1 Eclipse Model

The chaser's EP system is powered by solar arrays which are subject to blackout periods during solar eclipse. This causes a discontinuity in power (and thrust) available to the chaser, which needs to be taken into account in the thrust model.

In the eclipse model, for which a schematic representation is presented in Figure 5.3, the Earth is assumed to be spherical and to project a cylindrical shadow region with the radius of the cylinder base equal to the Earth's equatorial radius R_e in the direction opposing the sun. The shadow region experienced by the chaser is defined by the angle θ, which is the angle between the equatorial plane and the intersection of the edge of the eclipse and the orbit of the chaser. It is assumed that the chaser cannot thrust while travelling through the shadow region. To allow the tangential acceleration to change the orbit's semimajor axis with a negligible change in eccentricity, it is required that the chaser thrusts along symmetrically opposing arcs on the orbit. An on/off duty-cycle, which is the percentage on/off time of a given thruster during a burn, is also needed to prevent the thruster from exceeding the power capability of the chaser. It is therefore assumed to use a thruster duty-cycle that is synchronous with the eclipses on each orbit. This also allows for two duty cycles of the thruster per orbit, one corresponding with the shadow region and one corresponding to the arc which is symmetrically opposed to the shadow region (indicated in grey in Figure 5.3). It follows that the chaser can thrust only outside of the two opposing arcs of angle 2θ.

In particular, to account for a suitable duty-cycle, the thruster of the chaser are considered to be off for 40% of the time, while travelling through the shadow region and the opposing arc. From this, it results that the angle θ is equal to 36 deg, which means that the eclipse angle is $2\theta = 72$ deg.

It should be noted that, for simulation purposes, instead of considering the thruster burning at full capacity for 60% of the time (being off in the shadow region and the opposing arc), it is chosen to consider the thrusters able to burn all the time but at a 60% capacity. The trajectories resulting from these two options would achieve the same transfer departing from D1 at the same departure date and arriving at D2 at the same arrival date, requiring the same amount of propellant mass. For this reason, this choice is satisfactory for the goals of this PhD research.

5.1.2 Optimal RAAN-Phasing Orbits

As mentioned, the thrust is used to realise the change in altitude between debris orbits, while the Earth's oblateness gravitational perturbation is exploited to achieve the change in RAAN. Before reaching the next debris to rendezvous, dock and de-orbit, the spacecraft transfers to a RAAN-phasing orbit at a certain altitude h_P where it takes advantage of the orbital-averaged RAAN variation rate (Eq.(5.1)) to realise the necessary RAAN change so that the chaser can then reach the next debris matching its RAAN. The chaser is required to maintain the altitude of the RAAN-phasing orbit, counteracting the atmospheric drag action, for a sufficient *phasing time* to achieve the desired RAAN.

Considering the perturbing accelerations which act on the chaser and space debris continuously, as described in Section 3.1, the altitude of the RAAN-phasing orbit is chosen as a trade off between (i) the time T_{PT} required to reach the phasing altitude and the phasing time and (ii) the ΔV_{PT} to perform the change of altitude and counteract the drag action while phasing:

$$T_{PT} = T_{2,a} + T_p + T_{2,b} \tag{5.6}$$

$$\Delta V_{PT} = \Delta V_{2,a} + \Delta V_p + \Delta V_{2,b} \tag{5.7}$$

where the subscript $(2,a)$ refers to the transfer from the disposal orbit to the phasing orbit, the subscript p refers to the RAAN-phasing orbit and the subscript $(2,b)$ refers to the transfer from the phasing orbit to D2. The minimum altitude of the phasing orbit is set equal to the disposal altitude.

The objective function used for the selection of the optimal phasing altitude can be defined as:

$$J = \alpha \Delta \hat{V}_{PT} + (1-\alpha)\hat{T}_{PT} \tag{5.8}$$

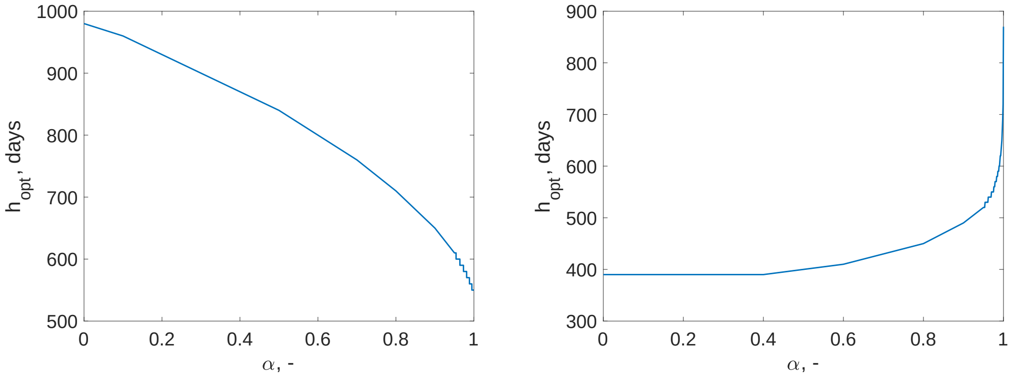
with $\alpha \in [0,1]$ being a weighting factor which can be chosen by the mission designer, to select the weight given to the normalised propellant mass $\Delta \hat{V}_{PT}$ and time of flight $\hat{T}_{PT}$ in the selection of the phasing altitude. The normalised values are calculated with respect to the maximum ΔV (obtained from the rocket equation given the amount of propellant on-board and the I_{sp} of the EP system) and maximum duration of the mission, which are defined below for the current mission.

Figures 5.4, 5.5 and 5.6 show, respectively, how the optimal altitude, the phasing time T_p and ΔV_p due to the drag, and T_{PT} and ΔV_{PT} vary as a function of α. It can be noticed that these variations also depend on the altitude of the arrival debris, in this case the results for a lower altitude of D2 of 500 km is shown in subfigures (a) and a higher altitude of 1000 km is shown in subfigures (b) (with a fixed $\Delta\Omega$ of around 360 deg to realise in both examples).

A shorter phasing time is generally required when the difference in altitude between the phasing and the D2 orbit is maximised (to achieve a greater difference of their RAAN variation rates) so that the RAAN of the arrival debris can be matched quicker. For instance, in the case of low altitudes of D2, it is convenient to wait at a higher orbit, as shown in Figure 5.4(a). Differently, for higher altitudes of D2 (subplot (b)), the optimal phasing altitude h_P results to be a lower altitude. In this case, the optimal h_P is equal to the altitude of the disposal orbit for α from 0 to 0.4, then it increases for higher values of α so that ΔV_{PT} can be reduced. However, from the figure it can be also noticed that there is a maximum altitude above which it becomes inconvenient to RAAN-phase due to the longer transfer time to reach those altitudes. In the current examples, the maximum phasing altitude is about 1000 km and 900 km for D2 with altitude h_{D2} = 500 km and h_{D2} = 1000 km, respectively.

From these optimal phasing altitude h_P, the resulting T_p and ΔV_p due to the drag are studied and presented in Figure 5.5, while T_{PT} and ΔV_{PT} are described in Figure 5.6. As expected, ΔV_p is smaller for higher phasing orbits, as the atmospheric drag action to counteract decreases. However, Figure 5.6(a) shows how reaching higher altitudes generally requires additional ΔV due to longer transfer legs, resulting in a larger ΔV_{PT}. Also, the subplots (a) in both figures show that, even when a transfer time is needed for the chaser to transfer to a higher orbit to RAAN-phase (for lower values of α), T_{PT} is minimum because in this instance RAAN-phasing at a higher altitude allows to minimise T_p. In the case of D2 with altitude h_{D2} = 1000 km (subplots (b)), for $\alpha \leq 0.4$ the optimal phasing altitude results equal to the disposal altitude, consequently the phasing time and ΔV stay constant. Since for h_{D2} = 1000 km the optimal h_P is lower than h_{D2} for any value of α (i.e. there is no transfer to phasing altitudes higher than D2's altitude), the plots describing T_p and ΔV_p (Figure 5.5(b)) and T_{PT} and ΔV_{PT} (Figure 5.6(b)) are similar. From all these figures, it can also be noticed that, generally, the greatest variations of time and ΔV due to a change of phasing altitude occur for higher α values (higher than about 0.8).

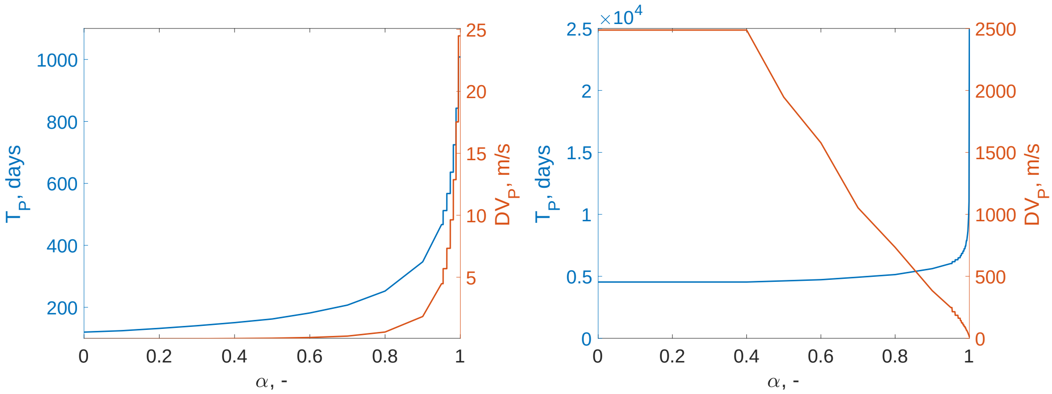
It is noted that Figures 5.4, 5.5 and 5.6 are highly dependent on the properties of the debris objects. For instance, if the objects are in an near-optimal RAAN-phasing condition after the orbital transfers, it can be more convenient to stay for a short amount of time at the disposal orbit to achieve the optimal RAAN instead of transferring to a RAAN-phasing orbit with a higher altitude (which can instead be preferable



(a) $h_2 = 500$ km

(b) $h_2 = 1000$ km

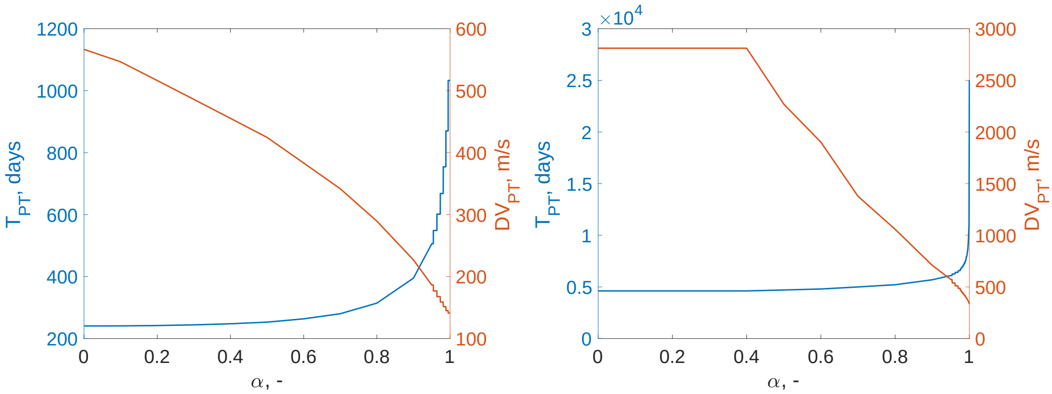
Figure 5.4: Optimal altitude h_{opt} as a function of α for arrival debris at lower (a) or higher altitude (b).



(a) $h_2 = 500$ km

(b) $h_2 = 1000$ km

Figure 5.5: T_p and ΔV_p due to drag as function of α for arrival debris at lower (a) or higher altitude (b).



(a) $h_2 = 500$ km

(b) $h_2 = 1000$ km

Figure 5.6: T_{PT} and ΔV_{PT} as function of α for arrival debris at lower (a) or higher altitude (b).

for larger $\Delta\Omega$, as suggested in Figure 5.4(a)).

Once the chaser reaches the target debris orbit of D2, a fixed stay time of 30 days is considered where the altitude is equal to D2's altitude to allow for the orbital phasing and capture operations to take place.

Figure 5.7(a) shows the variation of the phasing time T_p and the ΔV_p due to the drag on the phasing orbit, while increasing its altitude h_P (blue), as a function of the the required change in RAAN $\Delta\Omega$ raising from 0 to 360 deg (green). These results are obtained by considering $h_2 = 1000$ km and $\alpha = 0.5$. In this plot, the red markers indicate the values of T_p and ΔV_p at the optimal phasing altitude $h_{P,opt}$. As h_P increases, T_p increases exponentially. Phasing orbits at lower altitudes are preferred in this example to maximise the difference between the RAAN variation rates of the chaser at h_P and of D2 at h_2. For larger $\Delta\Omega$, the curve shifts towards higher values of T_p and ΔV_p.

Figure 5.7(b) describes T_p and ΔV_p as a function of $\Delta\Omega$ at the optimal altitude $h_{P,opt}$. As expected, while $\Delta\Omega$ increases, both T_p and the ΔV_p due to the drag increase almost linearly. However, ΔV_p presents a discontinuity at $\Delta\Omega = 250$ deg. For $\Delta\Omega \leq 250$ deg the optimal h_P is the disposal altitude, while for $\Delta\Omega > 250$ it is more convenient to phase at a slightly higher altitude ($h_P = 400$ km) to reduce the ΔV_p as a larger phasing time is required to achieve a larger $\Delta\Omega$.

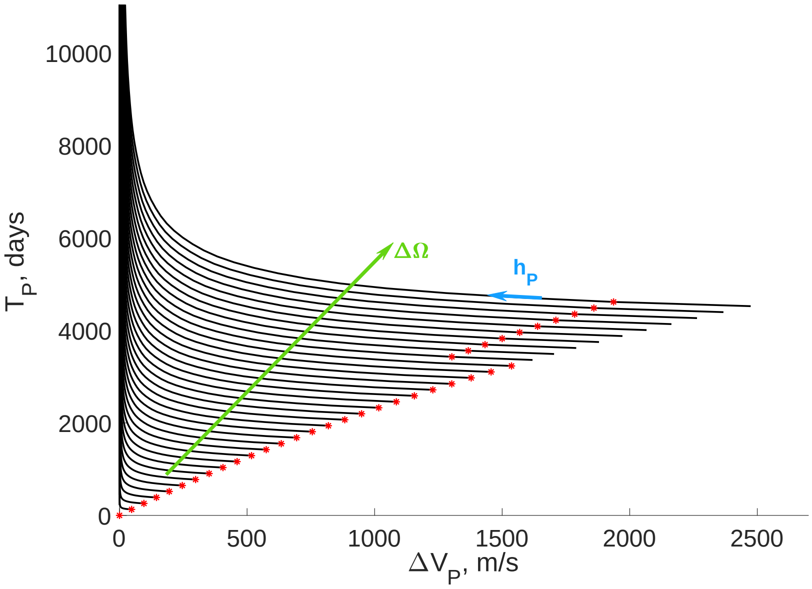
5.2 ANN for Multiple Debris Removal

Following the general description of ANNs in Section 3.3 and the detailed analysis to design the network architecture to perform optimally for the given applications in Chapter 4, this section focuses on how the ANN is trained and used to solve the MADR problem efficiently. In particular, the generation of the MADR training database and the analysis of the network performance are presented.

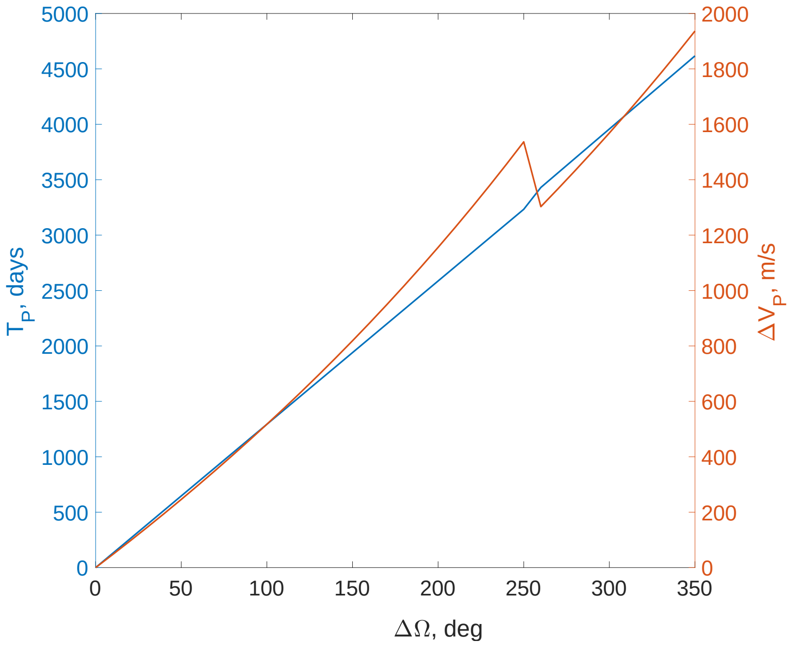
5.2.1 Training Database

As mentioned, the training database contains the inputs and the desired outputs (targets), which are used during the training of the network. According to the dynamics of the problem presented in Section 3.1 and the trajectory model introduced in Section 5.1, the low-thrust trajectories can be calculated by integrating the orbital dynamics starting from the initial conditions, so that the arrival conditions are met. This provides a reasonable accuracy (since adjustment can be performed during the transfer given the large number of revolutions [209]), and allows to avoid the use of an optimisation technique and, consequently, reduce the database generation time.

The *input vector* $\mathbf{x}$ needs to contain the orbital parameters of the departure and arrival objects, the mass of the departure debris m_{D1}, which needs to be carried to



(a) Increasing h_P



(b) At $h_{P,opt}$

Figure 5.7: T_p and ΔV_p as a function of $\Delta\Omega$ for increasing phasing altitude (a) and at the optimal altitude (b) (for $h_2 = 1000$ km, $\alpha = 0.5$).

disposal, and the initial mass of the chaser m_{SC}, which varies during the mission due to the propellant consumption. The *output vector* **y** includes the propellant mass expenditure m_{prop}, which is related to ΔV by the Tsiolkovsky rocket equation, and the TOF $t_{0,f}$ of the transfer between the departure (D1) and arrival debris (D2). As it is assumed that the debris objects' orbits are circular and at the same inclination, for this application the orbit of a debris object is completely specified by its altitude, h, and RAAN, Ω, at the departure epoch. It follows that:

$$\mathbf{x} = [h_{D1}, \Omega_{D1}, m_{D1,0}, h_{D2}, \Omega_{D2,0}, m_{SC}] \tag{5.9}$$

$$\mathbf{y} = [m_{prop}, t_{0,f}] \tag{5.10}$$

In this case, it is chosen to use the required propellant mass as an output instead of the ΔV to stay consistent, since masses are also used in the network inputs, which are needed because the debris D1's mass to dispose needs to be accounted for in the leg from the debris orbit to the disposal orbit.

To generate the database 300 debris objects are selected, with altitudes between 500 and 1500 km with an inclination fixed to 87.9 deg, RAAN between 0 and 360 deg, and mass between 100 and 300 kg. The number of 300 objects is identified as optimal in Section 4.2, because a smaller number reduces the performance of the network while a greater number introduces overfitting as the training becomes too extensive.

The range of altitude and mass of the debris orbits are selected considering the current ADR missions and, particularly, the End-of-Life Service by Astroscale (ELSA) project, which aims at targeting failed satellites and has as primary customers the emerging LEO communication providers such as Starlink and OneWeb. Starlink is SpaceX's 12,000-satellite LEO constellation to provide broadband Internet access. It orbits the planet at an altitude of about 550 km and covers the entire globe [211]. Each Starlink satellite has a mass of about 260 kg [212]. OneWeb is a LEO communication constellation of about 648 satellites (as initially planned, and with the high chance to dramatically expand the number of satellites in the near future). Each satellite is characterised by a mass of around 150 kg and operates at an 1,200 km orbit [213]. Looking broader than ELSA and the LEO communication customers, there are a few large (mass larger than 500 kg) satellites in LEO, but there is a greater number of objects which are smaller. It is likely that, in the coming years, the LEO communication constellations will dominate (in terms of sheer numbers) the LEO environment.

It was mentioned that the most densely populated, thus critical, region in LEO is around 750–1000 km altitude with an inclination between 60 deg and 95 deg [41, 18], which justifies the choice of the inclination of 87.9 deg for this MADR mission. Debris at other altitudes can be considered for removal for Space Traffic Management operation purposes. The disposal orbit is at an altitude of 390 km, which is also fixed. Thus, the training database comprises a total of 89,700 low-thrust transfers, which are divided into three sets: training, validation and test sets.

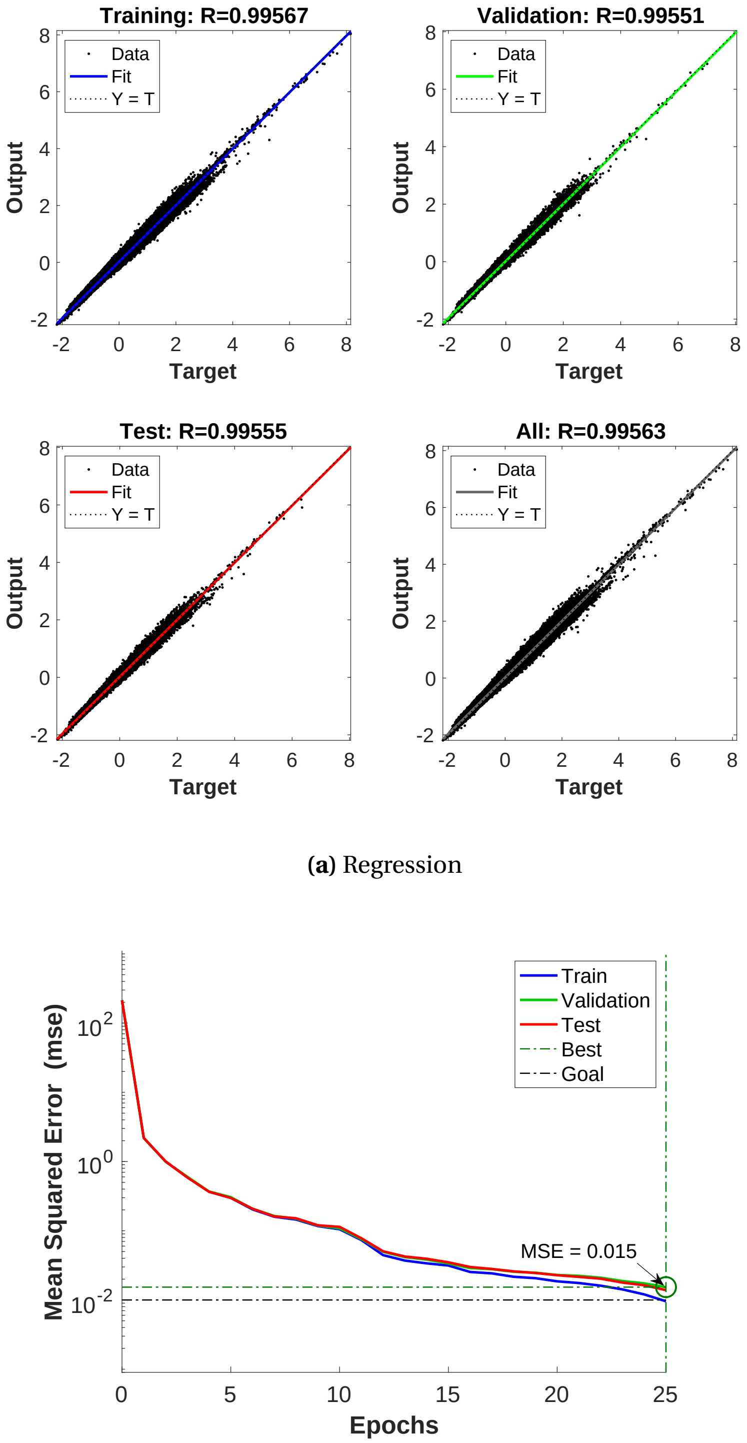
The chaser mass can vary from 300 kg to 400 kg, with the latter being the starting mass, since the on-board propellant mass is 100 kg. It is provided with an EP system which is characterised by a specific impulse I_{sp} = 2000 s and maximum thrust T_{max} = 21 mN. The value for the drag coefficient C_D is set equal to 2.2 and it is taken from the Astroscale's spacecraft with a mass of 400 kg and aerodynamic surface area of 8 m^2. Calculating the drag coefficient for satellite bodies is notoriously difficult. The value of C_D = 2.2 is used within the industry as a reasonable first guess, before a more detailed analytical number can be calculated (if needed).

5.2.2 Performance Analysis

The network with the optimal design configuration is trained using the Levenberg-Marquardt training algorithm, as described in Section 3.3.2. Figure 5.8 represents the performance of the trained network, showing the regression plots (a) and the performance plot (b). The regression plots show that the majority of the data falls in the neighbourhood of the line with a unit slope and zero y-intercept. The correlation coefficient is consequently very close to 1, being equal to 0.99 for all the training set, validation set and test set. The overall final correlation coefficient is R = 0.9956. By looking at the performance plot (b), we can notice that the final validation MSE of 0.015 is reached at epoch 25, which is when the performance goal is met.

The network performance obtained for the MADR case is compared to the one obtained for the MNR case, which was presented in Section 4.4 and which the architecture and hyper-parameters are optimised for. It follows that the identified optimal architecture and hyper-parameters provide a very high network accuracy also for MADR missions, confirming that this optimal network configuration allows for an optimal performance for multi-target missions. Also, it can be noticed that for MADR the network presents a greater performance than the MNR case (where R = 0.9732 and validation MSE of 0.121). The reason behind this is that for MNR missions the approximation of circular orbits at the same inclination is abandoned and a more complex model is used to calculate the 3-D low-thrust transfers, as it will be described in Chapter 6.

The error analysis of the network output is also performed and presented in Figure 5.9, where Eq.(4.2) is used to calculate this percentage error. The maximum error experienced, which has a small occurrence, is around ±20% for m_{prop} and TOF. However, the mean percentage error is 1.27% for the m_{prop} and 4.02% for the TOF, with the TOF having generally an higher variance compared to the propellant mass. The performance and error analyses suggest that the training of the network was successful, thus the network can predict the propellant mass expenditure and TOF to dispose debris objects with a high accuracy.



(a) Regression

(b) Performance

Figure 5.8: ANN regression (a) and performance (b) analyses.

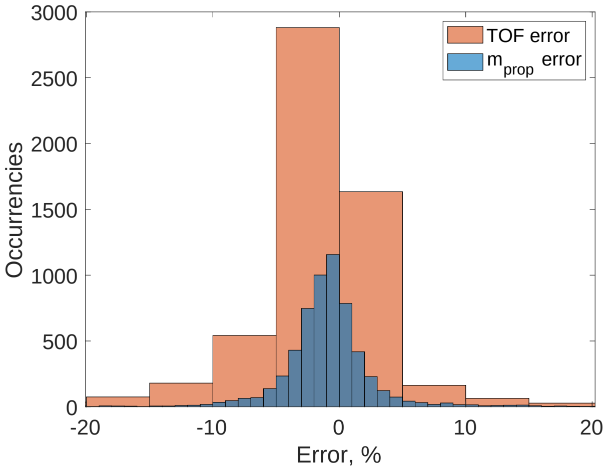


Figure 5.9: ANN error analysis.

5.3 Sequence Search

The following sequence search algorithm is implemented to identify the most promising sequences of debris to rendezvous, dock and de-orbit. The logic of the algorithm is schematically illustrated in Figure 5.10 and is based on a tree-search method and breadth-first criterion, as explained in Section 3.4. Each node of the tree represents a trajectory and how one proceeds through its branches depends on the mission objectives which, in this case, is the TOF minimisation.

A database of $N = 5{,}000$ fictitious debris objects is generated. The objects are generated with random mass $m_D \in [100, 300]$ kg, RAAN $\Omega_D \in [0, 360]$ deg and altitude $h_D \in [500, 1500]$ km. A database of fictitious objects, where some assumptions are considered (such as zero change in inclination and zero eccentricity), is used to find time-optimal transfers within a reasonable time to demonstrate the ANN capabilities. However, it can be replaced with a more complex transfer model (e.g. with fully-optimal low-thrust legs), which can be used to train the ANN. As the ANN replicates the behaviour of the model it is trained with, this would not have an impact on the benefits obtained from the use of the proposed methodology with ANN, i.e. shorter computational time while maintaining a satisfactory level of accuracy.

The SS works by selecting D_j as departure object and D_i as arrival object, with j and $i \in [1, N]$ so that all the possible permutations between the debris objects in the database can be evaluated. The RAAN of each object is updated to the value at the departure date. The trained ANN is embedded within the SS algorithm (the SS-ANN methodology) to calculate the m_{prop} and TOF of each low-thrust transfer. The $N_S = 200$ trajectories with the shorter transfer time are stored and a fixed capture time $t_s = 30$ days is added once the chaser reaches the arrival object to ensure enough time for rendezvousing and docking. At this point, the arrival object becomes the

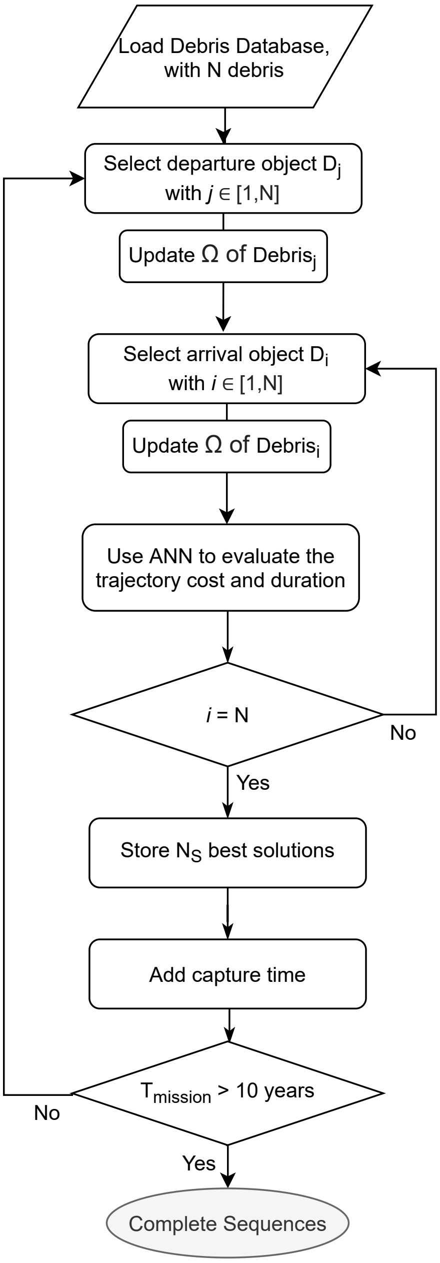

Load Debris Database, with N debris
Select departure object D_j with $j \in [1,N]$
Update Ω of $Debris_j$
Select arrival object D_i with $i \in [1,N]$
Update Ω of $Debris_i$
Use ANN to evaluate the trajectory cost and duration
$i = N$
No
Yes
Store N_S best solutions
Add capture time
$T_{mission} > 10$ years
No
Yes
Complete Sequences

Figure 5.10: Sequence search algorithm for MADR missions.

departure object of the following leg and the same procedure is iterated. The sequence is complete once the total mission duration exceeds 10 years (or until the depletion of the propellant mass). The TOF limit of 10 years applies to the entire mission, and it was introduced as a benchmark to identify, for example, how many debris could be disposed within this time frame.

For larger number of objects, the complexity and required memory of the tree

Table 5.1: Comparison of the proposed methodology SS-ANN against a current method employed by the industry [1].

Method	No. Debris	No. Captures	Computation time, min	m_{prop}, kg	TOF, days
Industry	100	4	240	30.7	2307
SS-ANN	100	4	9.37	28.6	1559

search grow exponentially. The motion of each object along its orbit needs to be calculated and updated at each iteration. Therefore, each node of the search tree is designed to record the orbital parameters of the selected target at the departure time, current flight time, current propellant mass expenditure, and the quality of the selected target (i.e. m_{prop} and $t_{0,f}$ calculated by the ANN).

It should be noted that, although in this case the TOF is chosen as the objective function to minimise when selecting the N_S best sequences at every iteration, other parameters can be eventually used in the selection process. For example, the propellant mass required to dispose the objects in the sequence, the collision risk with active satellites (if looking at the safety aspect), the need to free useful orbital slots (if looking at the economic aspect), the typology of capture mechanism (i.e. the type of capture mechanisms are indicated for certain debris objects considering for instance their tumbling or spinning motion).

5.3.1 Computational Time and Accuracy Analysis

The methodology SS-ANN is assessed, by comparing the performance with current methods employed by the industry [1], consisting of an industry expert processing the same input and providing the best solution possible within 4 hours via an iterative approach. For the comparison, the same set of input data (which is a database of 100 debris objects), same number of captures, same propellant system and same assumptions are used. Although the computational time is fixed to 4 hours for the analysis performed by the industry, the SS-ANN stops when the most convenient solution is identified. This means that it is likely that the solution obtained by the industry is sub-optimal to the one identified by the SS-ANN methodology, or that the method employed by the industry would require even more time to compute the optimal sequence.

Table 5.1 presents the results of this analysis. For the same input data the computational time required by SS-ANN is more than 26 times shorter than the time required by the current methods employed by the industry [1]. This result is obtained when 100 debris objects are considered, and it is expected that the benefits of using the SS-ANN method become even more important when a larger set of satellites is taken into account. Additionally, the m_{prop} and TOF of the sequence found by the industry greatly exceed those of the sequence found by the SS-ANN platform by 7%

and 47%, respectively. This discrepancy could have probably been avoided if more time was allowed for the analysis performed by the industry, which unfortunately was not granted in this case. However, this confirms the superiority of the SS-ANN platform for the preliminary design of MADR missions. It can be concluded that the advantages of employing the SS-ANN platform may be encountered not only in terms of the computational speed to compute and select a solution, but also in terms of the optimality of the solution which can be computed in a given amount of time. This is particularly true considering the amount of data which need to be processed to identify the most convenient sequences of objects. In the following section, the performance of the SS-ANN method is shown in terms of the accuracy to identify the most convenient transfers and, thus, the debris objects to de-orbit within a sequence.

5.4 Sequence Optimisation

To validate the outcome of the sequence search algorithm, the SS-AAN platform is run with a larger database of 5,000 objects. Three sequences are selected and their transfers are fully re-calculated using an EP system with a specific impulse I_{sp} = 2000 s and maximum thrust T_{max} = 21 mN. In particular, the method used to compute the transfers in the training database (Section 5.2.1) is used to compute the full trajectory profile. Three sequence searches are run for $\alpha = \{0, 0.95, 1\}$, i.e. in the cases where the phasing altitude is selected on the basis of the phasing time only ($\alpha = 0$), a combination of the phasing time and ΔV_{PT} ($\alpha = 0.95$) or ΔV_{PT} only ($\alpha = 1$). In particular, the value of $\alpha = 0.95$ is chosen because for this value the optimal phasing altitude and relating parameters appear to vary more rapidly, as shown by Figures 5.4 to 5.7. For each simulation the best sequence in terms of maximum number of disposed debris objects are selected.

Table 5.2 presents the characteristics (altitude h, RAAN Ω, and mass m) of the de-orbited space debris as part of the MADR sequences A, B and C and the mission characteristics (propellant mass m_{prop} and TOF) of these sequences. Sequence A, obtained using $\alpha = 0$, allows for the disposal of 13 debris in 9.73 years with a required propellant mass of 84.97 kg. Sequence B, obtained using $\alpha = 0.95$, allows for the disposal of 11 debris in 10.87 years with a required propellant mass of 60.76 kg. Sequence C, obtained using $\alpha = 1$, allows for the disposal of 5 debris in 11.10 years with a required propellant mass of 20 kg. It follows that, on average, for $\alpha = 0$ about 6.5 kg of propellant and about 270 days are needed per debris disposal. Differently, for $\alpha = 0.95$ about 5.5 kg of propellant and about 360 days are needed per debris disposal, while for $\alpha = 1$ about 4 kg of propellant and about 810 days are needed per debris disposal.

Figure 5.11 shows the change in altitude and used propellant mass along the MADR sequences A, B and C. As expected from the analysis of Figure 5.6, the choice of the

Table 5.2: Characteristics of the debris disposed in the selected sequences A ($\alpha = 0$), B ($\alpha = 0.95$) and C ($\alpha = 1$).

Seq. A	h, km	Ω, deg	m, kg	m_{prop}, kg	$\mathcal{E}_{m_{prop}}$, %	TOF, days	$\mathcal{E}_{TOF}$, %
D1	637.16	287.50	163.40	N/A	N/A	N/A	N/A
D2	512.37	283	275.27	7.02	3.99	333.51	6.44
D3	585.17	275.30	138.41	5.26	4.58	188.44	8.89
D4	533.43	280.31	155.50	8.58	4.92	201.72	15.80
D5	644.38	256.52	212.14	7.14	6.79	398.12	6.37
D6	738.66	230.86	176.71	9.04	7.55	321.09	9.02
D7	501.13	274.57	106.99	7.34	3.35	251.63	3.86
D8	557.63	253.73	179.98	5.15	4.82	330.29	4.39
D9	516.09	259.77	105.56	5.82	8.27	347.93	5.66
D10	705.37	186.35	170.46	6.57	7.53	349.41	7.58
D11	588.68	221.62	191.09	7.61	1.87	290.43	7.05
D12	600.20	208.13	171.12	6.69	10.63	337.32	15.46
D13	526.06	244.28	175.02	8.74	8.37	205.03	1.37
Seq. B	h, km	Ω, deg	m, kg	m_{prop}, kg	$\mathcal{E}_{m_{prop}}$, %	TOF, days	$\mathcal{E}_{TOF}$, %
D1	661.73	162.03	131.24	N/A	N/A	N/A	N/A
D2	504.96	158.72	270.37	6.63	9.77	311.92	10.41
D3	577.40	149.33	163.74	5.04	6.59	302.34	6.24
D4	534.05	145.48	256.27	6.35	3.73	517.09	3.88
D5	517.24	145.16	253.91	4.82	1.25	237.76	5.80
D6	514.14	138.62	139.19	6.02	7.03	460.15	3.59
D7	534.61	128.19	213.67	4.35	9.42	305.27	4.23
D8	667.95	84.76	222.82	5.21	1.40	95.18	8.00
D9	694.88	66.66	200.81	7.97	11.66	361.76	12.72
D10	517.45	123.32	171.80	9.39	9.77	768.67	6.66
D11	588.01	88.94	213.15	4.95	2.33	309.09	8.16
Seq. C	h, km	Ω, deg	m, kg	m_{prop}, kg	$\mathcal{E}_{m_{prop}}$, %	TOF, days	$\mathcal{E}_{TOF}$, %
D1	533.43	280.31	155.50	N/A	N/A	N/A	N/A
D2	512.37	283	275.27	4.71	4.53	878.50	8.30
D3	585.17	275.30	138.41	5.07	10.86	1216.67	15.49
D4	543.71	288.64	172.30	5.33	4.43	1009.69	2.05
D5	517.31	303.68	291.92	4.93	5.25	960.91	7.10

weighting parameter α, thus of the phasing altitude, can have a significant impact on the propellant mass consumption and duration of the mission. As α increases, the

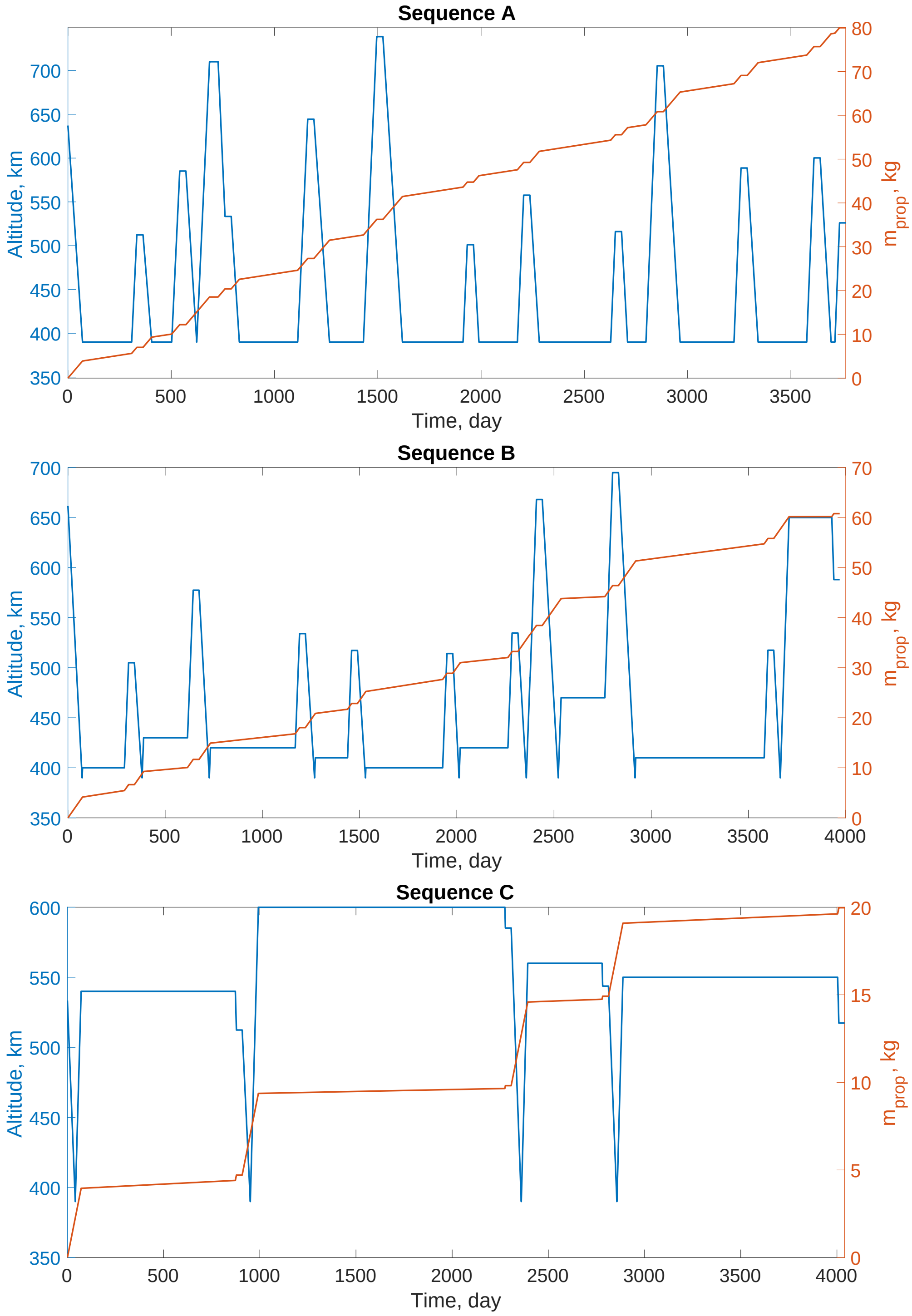


Figure 5.11: Altitude and used propellant mass for MADR sequences A, B and C.

TOF increases, while the required propellant mass decreases per debris disposal. This is particularly noticeable for the case of $\alpha = 1$, when the TOF sharply raises compared to the cases of $\alpha = 0$ and 0.95, resulting in a sensible reduction of the number of debris which can be de-orbited in about 10 years.

Table 5.2 also shows the mean percentage errors of the three MADR sequences between the values computed by the ANN and the optimal values. The mean percentage errors are calculated using Eq.(4.2). As expected, the percentage errors $\mathscr{E}$ are generally lower than 10% for the propellant mass consumption and lower than 15% for the TOF, with a final mean error of 6.18% for the m_{prop} and 7.48% for the TOF. It can be concluded that the trained network is able to estimate with a reasonable accuracy the cost and duration of low-thrust transfers to de-orbit space debris objects, while providing a dramatic reduction of the computation time.

Having tested the SS-ANN methodology with three different sequences of debris to remove, it is demonstrated that SS-ANN results to be a fast and reliable approach to conduct the preliminary design of MADR missions, offering significant advantages not only in terms of precision of the solutions but also in terms of computational time, which is dramatically reduced compared to the current most-commonly used techniques in the industry.

6

Design of Multiple NEA Rendezvous Missions

This chapter explores how machine learning can be used to find the best sequences of multiple asteroids to visit during a MNR mission. First, the mission scenario is presented in Section 6.1. Then, Section 6.2 describes how the training database of low-thrust transfers is created and used to train an ANN. A near-term low-thrust propulsion system is used. In Section 6.3, the integration of the trained network within a sequence search algorithm to search for the optimal candidate sequences of asteroids is detailed. Lastly, Section 6.4 presents the optimisation of three promising MNR sequences which are selected after running the SS-ANN method for this application. The performance of the network is assessed by evaluating the ANN prediction error and comparing the results obtained by the ANN with those obtained by the optimiser. Finally, an analysis of the computational time and accuracy of the network with respect to most commonly-used methodologies is conducted.

6.1 Mission Scenario

During an MNR mission the spacecraft is expected to rendezvous multiple NEAs. The considered scenario envisages the spacecraft to depart from Earth at a given departure date and then transfer to the first asteroid in the sequence to rendezvous. A rendezvous requires the spacecraft to match the position and velocity of an asteroid so that proximity operations can be conducted. Once this is completed, the spacecraft transfers to the next asteroid and the procedure is repeated until depletion of all propellant mass or until the maximum duration of the mission is reached. Figure 6.1 provides a schematic representation of the MNR mission scenario, where Ai with $i = 1, 2, \ldots$ indicates the NEA objects located on different orbits.

Compared to the MADR case where circular orbits at the same inclinations were

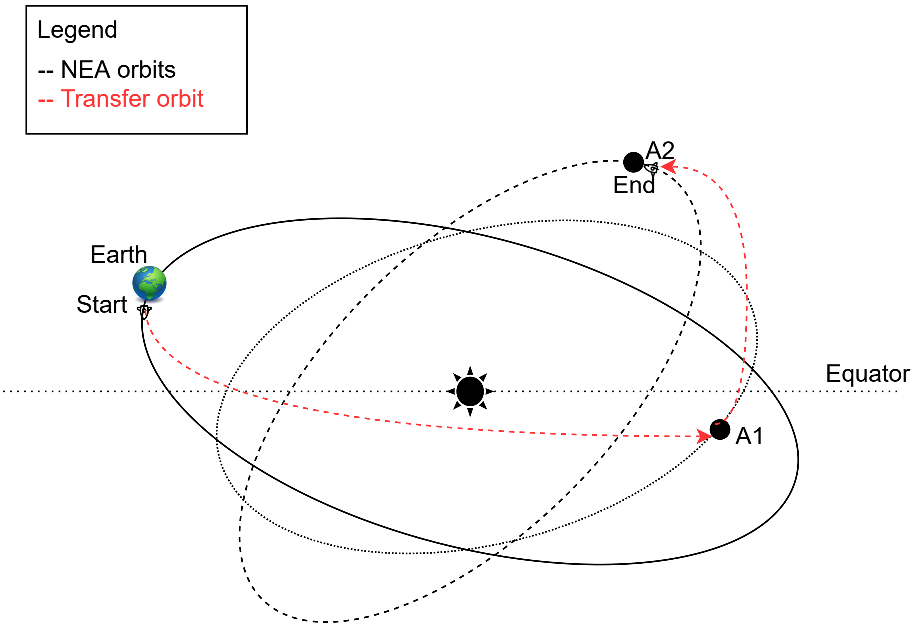


Figure 6.1: Schematic representation of the MNR mission scenario.

considered, a level of complexity is added for MNR missions where the spacecraft can transfer between objects with arbitrarily inclined and eccentric orbits. In the case study of MADR, the considered assumptions were chosen to be representative of a real MADR mission, while keeping the model simple, which allows to minimise the computational time to generate the training database for the ANN. Compared to MADR where satellite constellations are often on circular orbits, at the same inclination and spaced in RAAN, in the case study of MNR those assumptions cease to be representative of a real mission scenario. This is because NEAs are characterised by very diverse orbital characteristics which need to be taken into account during the mission design phase. The additional complexity introduced by MNR missions provides the opportunity to demonstrate the ANN capabilities when more complex transfer models are used and learnt by the network.

Determining the cost and duration of a low-thrust transfer, given the departure and arrival NEA orbital parameters, traditionally requires the solution of a low-thrust optimal control problem. As mentioned, this is very expensive computationally and requires an accurate first guess to identify an appropriate solution [140]. This represents an inefficient approach in the preliminary design phase of multi-target missions, where millions of trajectories need to be estimated to identify the best sequence of objects to follow. As discussed in the literature review in Chapter 2, MNR missions were generally designed by using simplified models to solve the continuous part of the global optimisation problem. This allows to reduce the computational time and

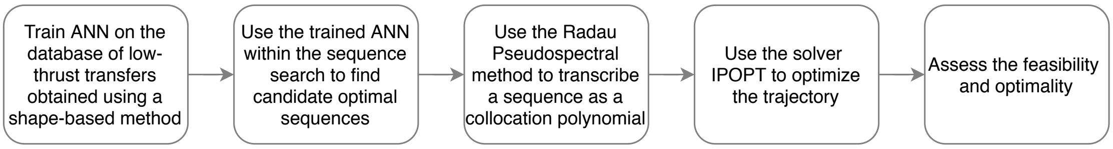

Train ANN on the database of low-thrust transfers obtained using a shape-based method
Use the trained ANN within the sequence search to find candidate optimal sequences
Use the Radau Pseudospectral method to transcribe a sequence as a collocation polynomial
Use the solver IPOPT to optimize the trajectory
Assess the feasibility and optimality

Figure 6.2: Flow of the steps of the proposed optimisation methodology.

effort to find the NEA sequences while all the asteroid permutations are considered [2, 5, 3, 4]. In this context, ANNs are employed in this work to investigate whether, when integrated with a SS algorithm, they can reduce the required computational time, while maintaining a satisfactory accuracy. The optimisation techniques can be instead used at a later stage to refine the selected sequence solution and obtain the optimal control history and trajectory for each leg of the mission.

Optimisation techniques can be used to compute the transfers to include within the training database, but they would require a high computational time to obtain a good initial guess and then find a solution for the trajectory OCP between each pair of NEAs. As mentioned in Section 3.2, the linear-trigonometric shape-based method allows to describe the trajectory analytically without the need of an initial guess and without requiring propagation or discretisation of the state vector [115]. Although the shape-based method can identify only a sub-optimal solution, this can be obtained quicker than solving the OCP which can be useful in the preliminary mission design phase (such as to generate the training database for the ANNs). In Section 4.2, the optimal number of samples to include in the training database is identified, which is of the order of tens of thousands for the MNR transfers.

In this chapter, the network performance to find the best sequences of multiple asteroids to visit during a MNR mission, using a near-term low-thrust propulsion system, is assessed. A schematic flow showing the steps of the proposed optimisation method is given in Figure 6.2. First, the ANN is trained using a database of low-thrust transfers, which are obtained using a shape-based method (described in Section 3.2.2). The trained network is then employed within a sequence search algorithm which uses a tree-search method (detailed in Section 3.4) to search for the optimal candidate sequences of asteroids. The sequences which best satisfy the mission requirements and, specifically, minimise the cost or duration of the mission, are transcribed into the collocation polynomials using the Radau pseudospectral method [157, 164] and converted into low-thrust trajectories using the NLP solver IPOPT [154]. The feasibility of the sequences is assessed by means of the local optimiser GPOPS-II [158], which was presented in Section 3.2.1. The following sections describe each of these steps in detail.

6.2 ANN for Multiple NEA Rendezvous Missions

This section presents the process followed to train the ANN for MNR missions, starting from the generation of the training database and then analysing the network performance. It is noted that a general description of ANNs is provided in Section 3.3 and a detailed analysis to design the network architecture to perform optimally for the multi-target mission applications is provided in Chapter 4.

6.2.1 Training Database

The training database includes the orbital elements of the departure (subscript 1) and arrival (subscript 2) orbits and the position of the objects along their orbits (inputs) and the cost ΔV and TOF of each transfer between pairs of these objects (outputs):

$$\mathbf{x} = [p_1,\ f_1,\ g_1,\ h_1,\ k_1,\ L_{1,0},\ p_2,\ f_2,\ g_2,\ h_2,\ k_2,\ L_{2,0}] \tag{6.1}$$

$$\mathbf{y} = [\Delta V, t_{0,f}] \tag{6.2}$$

where the MEE are used as inputs and L are calculated at the departure epoch t_0. In Section 4.3, it was shown that they allow the network to achieve a greater performance in terms of correlation coefficient between the network outputs and the targets and in terms of the validation-set MSE. This means that the best accuracy and generalisation capabilities are obtained by the network when the MEE parameterisation of the departure and arrival orbits is used as inputs to the network.

The NEA orbital elements are obtained from the NASA's near-Earth Object Program [214]. According to the NASA database, more than 28,000 objects are discovered until now (2022-04-08) and this number is continuously increasing. It follows that the selection of the asteroids to rendezvous in an MNR mission constitutes a challenging task, as it should consider the scientific value of each object, their composition, their orbital dynamics and the available launch window.

The database includes PHAs and near-Earth Object Human Space Flight Accessible Targets Study (NHATS) objects [215]. PHAs are classified as NEAs with an orbit characterised by an EMOID lower than 0.05 AU, i.e. an approach and, eventually, an impact with Earth is possible and can cause significant regional damage. In fact, PHAs are characterised by an absolute magnitude lower than 22, which means that their diameter can be larger than 100 m. Note that the absolute magnitude parameter H indicates the visual magnitude an observer would record if the asteroid was placed 1 AU away.

NHATS are objects selected by NASA which might be accessible by future human space flight missions [216]. These NEAs are selected depending on the characteristics of the asteroids themselves and of the mission required to reach each of them, while

Table 6.1: Criteria used in the selection of the NHATS database.

Parameter	Lower Bound	Upper Bound
Total ΔV, km/s	–	12
Mission duration, days	–	450
Stay time at the object, days	8	–
Launch date	2020-01-01	2045-12-31
Absolute magnitude parameter	–	26
Orbit condition code	–	7

departing from Earth. Different NHATS asteroids are chosen if different constraints are specified for the mission parameters, such as the total ΔV required for the transfer, the total mission duration, the stay time at the object before return to Earth and the launch date interval. These parameters need to be entered in the NASA database to select the NHATS since they are considered as targets for human space flights. Thus, the available NHATS depend on a given launch window and stay time before returning the humans to Earth. The criteria used for this work to select the NHATS objects to include in the training database are presented in Table 6.1 [217]. As the option of human space flight is not investigated in this work, larger ranges of these parameters are allowed to include more NHATS objects in the training database. Note that the orbit condition code (OCC) takes into account the orbit determination accuracy. In particular, the OCC is an integer between 0 and 9 and indicates how well an object's orbit is known on a logarithmic scale, where 0 indicates a well-determined orbit. Higher OCC values imply inherently greater uncertainty in the existence and ΔV associated with trajectories to those bodies. Performing more observations of these bodies would allow to reduce their orbit condition codes. For a complete explanation of the above criteria, the interested reader is referred to the JPL/NASA NHATS website [218].

From the analysis on the number of ANN training samples conducted in Section 4.2, it follows that the permutations between 100 NEAs needs to be computed for the generation of the network training database for MNR missions. It results in a number of 10,100 transfer samples. However, thanks to the generalisation capability, when running the SS-ANN method to identify the NEA sequences, a larger number of NEAs can be considered, potentially removing the need of pruning the database. Once the training of the network is successful, the ANN is able to generalise to provide the transfer costs between NEAs which are not included in the training database and with different launch dates.

Since transfers between orbits with a high inclination change Δi and high eccentricity change Δe are generally associated with a higher ΔV and transfer time, it is expected that those objects would be discarded by the sequence search algorithm.

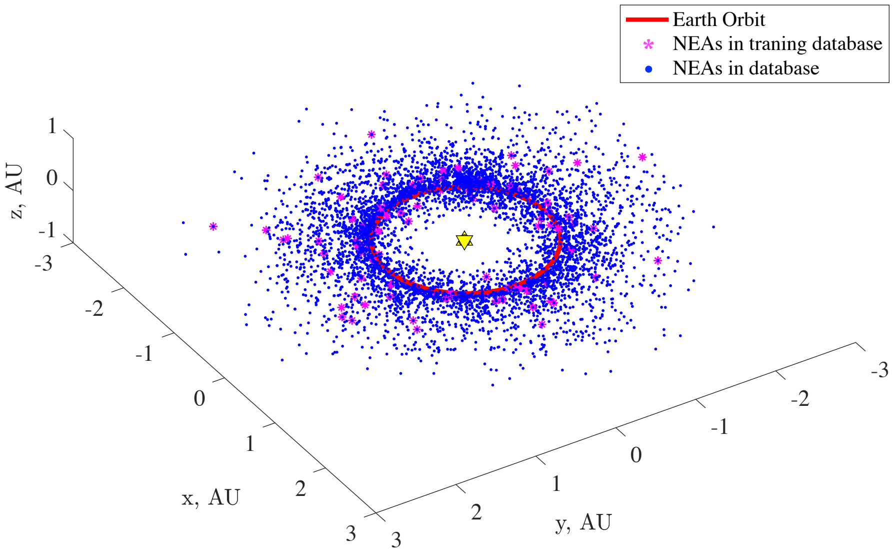


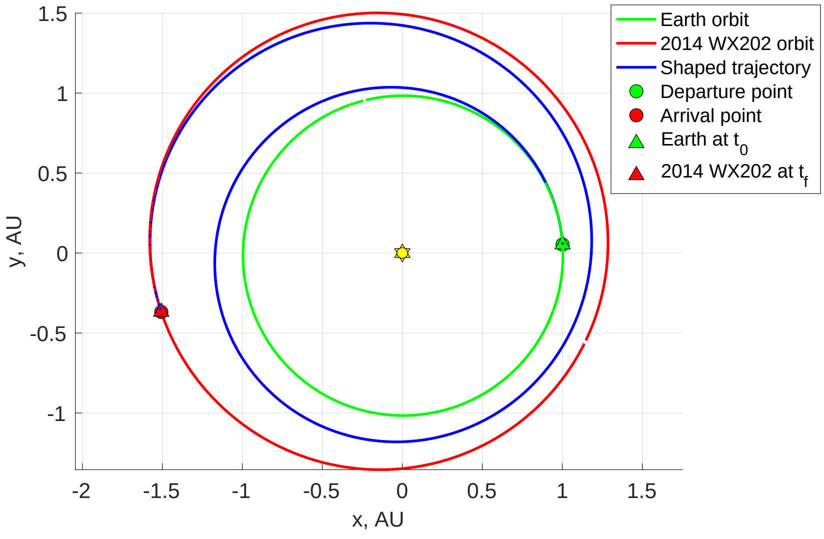
Figure 6.3: NEAs included in the database.

In fact, the SS algorithm is implemented to select NEAs in a sequence to minimise the ΔV and/or TOF. As a consequence, the highly inclined ($i \geq 20°$) and highly eccentric ($e \geq 0.4$) objects are excluded from the database, which also improves the convergence rate of low-thrust transfer solutions.

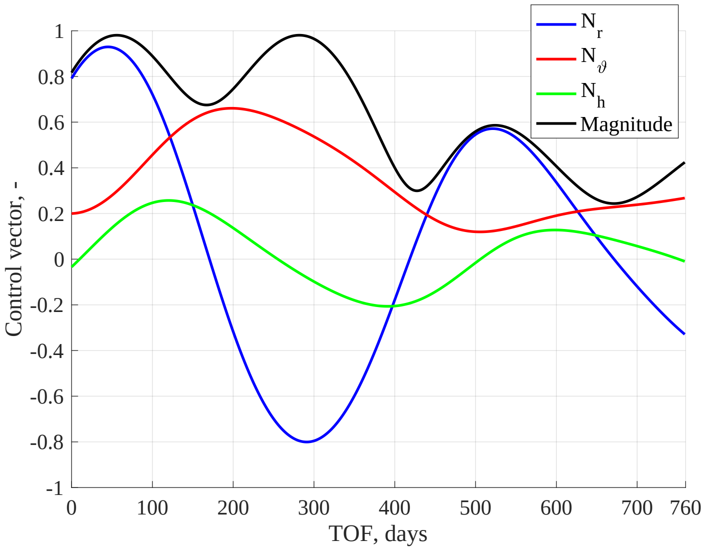
The orbital elements of the 100 NEAs selected for the training are taken from the NASA's Centre of Near-Earth Object Studies (CNEOS) [21]. The reference time of 2019-04-27 (i.e. modified Julian date or MJD of 58600) is used in the database to determine the reference position of the asteroids along their orbits. Figure 6.3 shows the 100 NEAs (represented in pink), which are used for the training, at their position with respect to the Earth's orbit (in red) on 2019-04-27. The blue dots represents the NEAs included in the set for the SS-ANN run.

The shape-based approach is used to compute the low-thrust trajectories from Earth to NEAs and between NEAs to generate the network training database. The shape of the minimum-duration rendezvous trajectory is defined analytically as defined in Section 3.2.2, for the given range of launch dates with zero departure and arrival velocity, TOF and number of revolutions. For this analysis the linear-trigonometric shape which is more suited to describe transfers with electric propulsion, is selected. For each individual transfer between two selected bodies, the shape-based method is run. The training database is built by storing, for each transfer, (i) the MEE of the departure and arrival bodies and their angular position at the departure date, which serve as network input, and (ii) the cost and TOF of the transfer with minimum TOF, which serve as network output.

The bounds of the parameters used by the GA in the shape-based algorithm to



(a) Trajectory



(b) Control history

Figure 6.4: Solution found by the shape-based method for a transfer from Earth to 2014 WX202: minimum-TOF trajectory (a) and control history (b).

select the shaping parameters are specified in Table 3.3, considering that an EP system is used with $I_{sp} = 3000$ s and $T_{max} = 0.1$ N. Figure 6.4 shows an example of a low-thrust transfer calculated by the shape-based method. The spacecraft departs from Earth on 2026-09-26 and transfer to asteroid 2014 WX202 in 759 days and requiring a total ΔV of 8.59 km/s. Plot (a) shows the trajectory with minimum required TOF and plot (b) presents the control history $\mathbf{N}(t)$ needed to fly the computed trajectory. In particular, plot (b) shows the profiles of the radial component N_r, the transverse component N_θ, out-of-plane component N_h and magnitude of the acceleration vector $\mathbf{N}$.

6.2.2 Network Performance

The selection of the network hyper-parameters to optimise the network architecture for this application was discussed in depth in Chapter 4. A network with four hidden layers and 80 neurons using the sigmoid as activation function is built. The database of 10,100 low-thrust transfers is divided into training set (70%), validation set (15%) and test set (15%), to detect and prevent overfitting. The training takes place by using the Levenberg-Marquardt algorithm, which resulted to contribute to a higher accuracy of the network function.

The final performance of the network was shown in Figure 4.9, presenting both the regression and performance plots, and discussed in Section 4.4. The final correlation coefficient R is 0.9732, which indicates a very accurate fitting. The final validation MSE is equal to 0.1211, which suggests that the network function accurately describes the relationship between the departure and arrival objects and the cost of the transfer between them, also in case of new samples, which were not included in the training.

6.3 Sequence Search

To identify the most promising sequences of asteroids to visit in a MNR mission, the sequence search algorithm, schematically illustrated in Figure 6.5, is implemented. The sequence search starts from Earth at the departure date of 2035-01-01. The departure date is chosen to be outside of the time frame used to compute the transfers included in the training of the network, so that the generalisation property of the network can be tested. Firstly, the full database of NEAs is loaded. The ephemerides of the asteroids are updated at each departure time t_i, with i indicating the i^{th} leg of the sequence. Secondly, the trained ANN is used to calculate the cost and the TOF of each transfer from Earth to all the NEAs available in the database.

Once all the transfers are evaluated, only $N_S = 200$ of the transfers with the lowest TOF are stored. This limit of N_S best transfers is set to reduce the otherwise enormous number of permutations which are required and would increase exponentially at every iteration, given the large number of NEAs. A stay time at the object of 100 days is added to allow for close-up NEA observations. The stay time of 100 days is considered a reasonable average value for this kind of missions [2, 219, 220]. At this point, the arrival object becomes the departure object of the following leg, for which the same procedure is iterated following a tree-search method. Each node of the tree represents a trajectory and the search proceeds through its branches depending on the mission objective, which, in this case, is the TOF minimisation. The sequence is complete

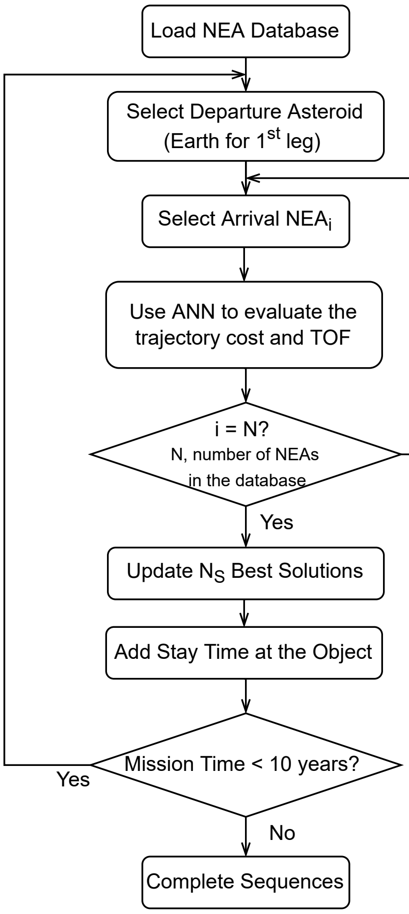

Load NEA Database
Select Departure Asteroid (Earth for 1st leg)
Select Arrival NEAi
Use ANN to evaluate the trajectory cost and TOF
i = N?
N, number of NEAs in the database
No
Yes
Update NS Best Solutions
Add Stay Time at the Object
Mission Time < 10 years?
Yes
No
Complete Sequences

Figure 6.5: MNR sequence search algorithm.

once the total mission duration exceeds 10 years.

In this work the launch date is fixed as the purpose of the algorithm is to prove the effectiveness of the use of ANN to design low-thrust, multiple asteroid missions. This can be complemented with a systematic search on several launch dates if a launch window is given. Figure 6.6 shows the sequences visiting a minimum of five asteroids in about 10 years, which were obtained by the SS-ANN methodology for the departure date 2035-01-01. Here only the first three legs are fully displayed because the number of asteroids visited grows rapidly with each leg. For illustration, only the sequences which are analysed and optimised in this section are fully shown.

Similarly to what was done in the SS for MADR missions (Section 5.3), the N_S best transfers are selected at every iterations based on the lowest TOF. It is worth noting that other factors can be taken into account for the selection of the NEAs in the sequences, such as the cost ΔV or propellant mass required for the transfer, the type or composition of the asteroids, or their dimension. In the following of this chapter

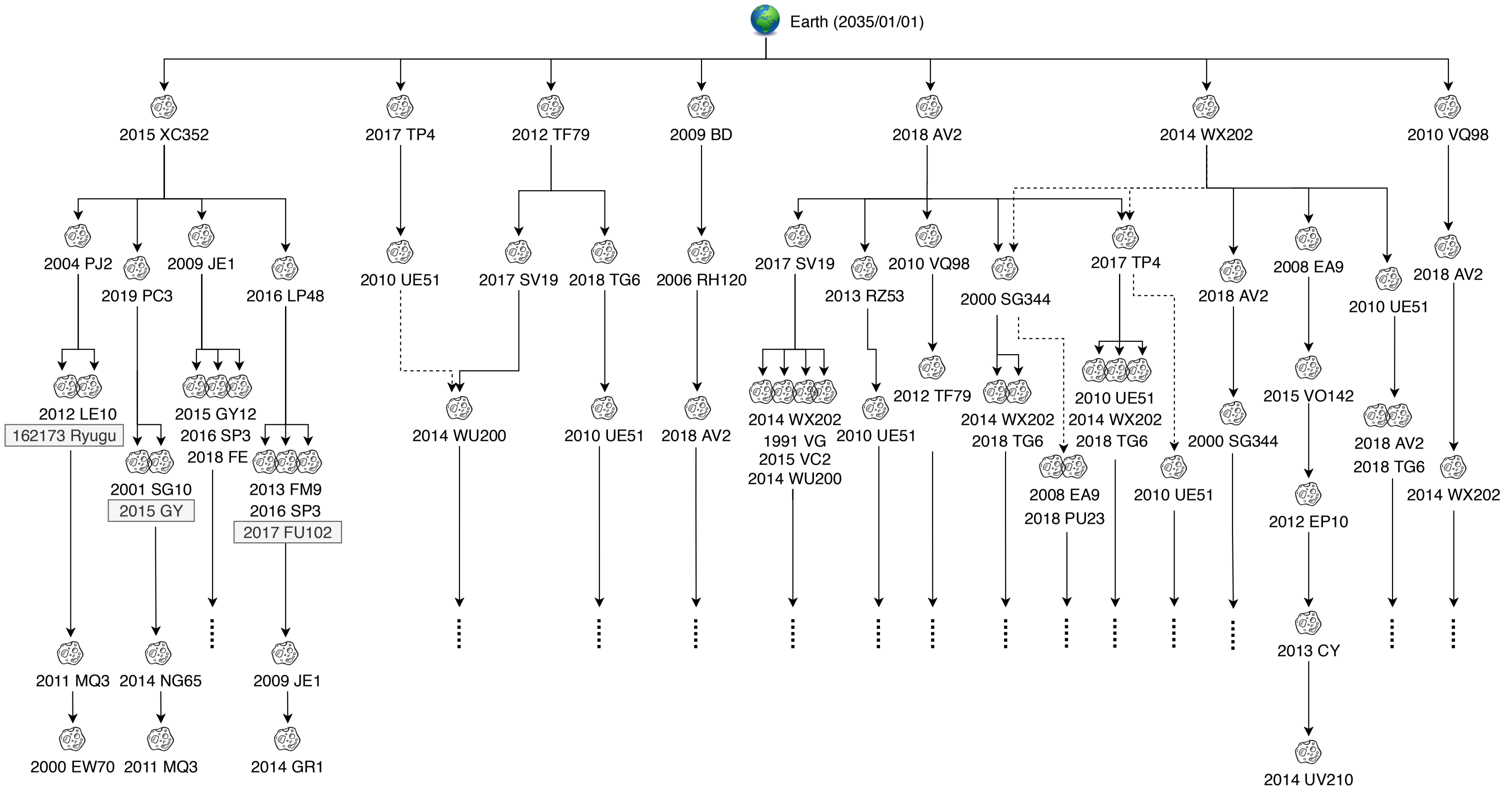


Figure 6.6: Tree graph of the MNR sequences identified by SS-ANN for the launch date 2035-01-01.

only TOF is considered, however in Chapter 7 the possibility of selecting the objects in the sequence based on other parameters (or a combination of them) is investigated and the resulting effects on the properties of the calculated sequences are assessed.

6.4 Sequence Optimisation

The SS-ANN method is run to calculate the NEA sequences and identify the most promising ones. To this end, a NEA database of 6,286 asteroids is considered, of which about 300 are PHAs and about 1,450 are NHATS. This is obtained by excluding the highly inclined ($i \geq 20°$) and highly eccentric ($e \geq 0.4$) objects from the analysis, as explained above. The reference time of 2019-04-27 (MJD 58600) is used in the database to determine the position of the asteroids along their orbits. Figure 6.3 shows the 6,286 NEAs, which are used for the simulation, at their position (blue dots) with respect to the Earth's orbit (in red) on 2019-04-27. The orbital elements of these NEAs (*input*) are obtained from the NASA's CNEOS [21].

Once the multiple NEA rendezvous sequences are found by the SS-ANN platform (Figure 6.6), the OCP is formulated as shown in Section 3.2 and solved to calculate the 3-D low-thrust trajectory by using the OCP solver GPOPS-II [158], which was described in Section 3.2.1. In this study, two types of low-thrust propulsion systems are considered: an electric propulsion (EP) system and a solar sailing propulsion (SSP) system.

The state vector of the system, **x**, is expressed in MEE (defined in Section 3.1), adjoined by the spacecraft mass when an EP system is used:

$$\mathbf{x_{EP}} = [p, f, g, h, k, L, m]^T \tag{6.3}$$

Since there is no significant mass variation in the SSP case, the spacecraft mass m is not included in the state vector, which is then defined as follows:

$$\mathbf{x_{SSP}} = [p, f, g, h, k, L]^T \tag{6.4}$$

The set of ordinary differential equations of motion, which describes the equations of the dynamics, can be formulated as:

$$\dot{\mathbf{x}}(t) = \mathbf{A}(\mathbf{x})\mathbf{a} + \mathbf{b}(\mathbf{x}) \tag{6.5}$$

which was further detailed in Eqs.(3.8) to (3.14). Similarly, the thrust acceleration which has a different definition depending on the propulsion system considered is described by Eqs.(3.17) and (3.18) when EP is used, and by Eq.(3.19) when SSP is used.

The OCP consists in finding the optimal control history $\mathbf{u}(t) \equiv \mathbf{N}(t)$, with $\mathbf{N}(t)$ being the acceleration direction and magnitude vector, so that the time of flight is minimised. Thus, the performance index to minimise is:

$$J = \int_{t_0}^{t_f} \mathrm{d}t \tag{6.6}$$

while satisfying the dynamics defined in Eqs.(3.8) to (3.14) and the following path constraints (Eq.(3.39)):

$$\begin{aligned} 0 < |\mathbf{N}| < 1 \qquad & \text{for EP} \\ |\mathbf{N_s}| = 1 \qquad & \text{for SSP} \end{aligned} \tag{6.7}$$

and the following rendezvous boundary conditions (Eq.3.41):

$$\begin{cases} \mathbf{r}(t_d) - \mathbf{r}_d(t_d) = \mathbf{0} \\ \mathbf{v}(t_d) - \mathbf{v}_d(t_d) = \mathbf{0} \end{cases} \quad \text{and} \quad \begin{cases} \mathbf{r}(t_a) - \mathbf{r}_a(t_a) = \mathbf{0} \\ \mathbf{v}(t_a) - \mathbf{v}_a(t_a) = \mathbf{0} \end{cases} \tag{6.8}$$

Details regarding the above equations are given in Section 3.2 and only the main equations are reported again here for convenience.

Given the NEA sequences found by SS-ANN, the algorithm to compute the optimal low-thrust trajectories is implemented as in the following steps:

1. The algorithm automatically computes the initial guess for each leg separately 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. When solar sailing is used, the optimal solution obtained for the EP case is adopted as the initial guess.
2. The algorithm optimises the 3-D trajectory leg by leg. During the optimisation phase, the solution of each leg is constrained to start at least 20 days after the arrival of the previous leg. This is done to avoid overlapping between two consecutive legs and allow some time for close-up observations.
3. 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 by using the TOF predicted by the neural network during the sequence search as the first guess. Also, a window of ±100 days is considered for the departure time t_0 and arrival time $t_f = t_0 + t_{0,f}$, with $t_{0,f}$ being the TOF of the transfer, as calculated by the ANN. This is done so that the best epochs for departure and arrival can be selected by the optimiser. The stay time at each asteroid is consequently adjusted considering the TOF of each leg and the departure date of the successive leg.

As detailed in Section 3.2, the quality of the optimised solution is dependent on the initial guess chosen. However, the direct collocation method, which is employed by GPOPS-II to find the solution, is selected because it offers a generally robust convergence also to poor initial guesses. To build the training database, a shape-based method is used, thus the complex function utilised by this method is learnt by the network to estimate the cost and duration of the low-thrust transfers between NEAs. Various studies confirm that a solution provided by a shape-based method can be a good initial guess for the optimisation procedure [115, 114, 220, 166]. Consequently, using the results obtained by the ANN (which simulates the shape-based method following a successful training) as a first guess within the optimisation procedure is expected to allow for a convergence to a locally optimal solution. To test this, once the solution is found, the optimiser is run again using different initial guesses, shorter or longer (with maximum variation of about ±200 days) than the values calculated by the ANN. As the optimiser converges to the same solution, it can therefore be considered as the locally-optimal solution for the given transfer leg.

Lambert's Problem

The Lambert's problem consists in finding the orbital elements of an arc of a conic section connecting two points $\mathbf{r_1}$ and $\mathbf{r_2}$ in space in a given time Δt while performing a given number of full revolutions. The outcome is referred to as Lambert arc [147]. Once the orbital parameters have been determined, the velocity vectors $\mathbf{v_1}$ and $\mathbf{v_2}$ at the extremities of the arc, i.e. at the departure and arrival points $\mathbf{r_1}$ and $\mathbf{r_2}$, are calculated. This work makes use of a multi-revolution Lambert solver which is implemented in C and further details can be found in Ref. [221].

6.4.1 NEA Sequence Analysis

The methodology and the tools described above are used to calculate the sequences of asteroids to visit with minimum TOF and, subsequently, optimise the full trajectories. To compute the $N_S = 200$ sequences, as described in Section 6.3, the SS-ANN platform required about 13.3 hours on a machine with an Intel Core i7 processor at 3.4 GHz.

Two sequences have been selected for the full optimisation. These have been selected as they present the largest variation in semi-major axis $\Delta a = 0.4$ AU (Sequence A), and largest variation in both inclination $\Delta i = 7.4$ deg and eccentricity $\Delta e = 0.25$ (Sequence B). These variations are registered in a single leg, i.e. between one asteroid and the next asteroid in a sequence. This is preferred over considering the greatest variation of these values in total because the algorithm works on asteroid-to-asteroid transfers, which is what is desired to verify and validate. The encountered bodies of the selected sequences are characterised in Table 6.2. As shown, the majority of the objects are PHA and NHATS. Note that the semi-major axis, eccentricity or inclination between two consecutive objects are not changing monotonically, as it was instead found in the MNR sequences in Ref. [2]. In fact, the variation of these variables can be positive or negative while transferring to the next object. This is particularly

Table 6.2: Orbital parameters of the NEAs in the optimised sequences A and B.

Sequence A	2015 XC352	2004 PJ2	162173 Ryugu	2011 MQ3	234145 (2000 EW70)
Classification	NHATS	PHA	NHATS	NHATS	PHA
a, AU	1.016	1.418	1.19	1.12	0.94
e, -	0.15	0.34	0.19	0.11	0.32
i, deg	4.49	2.58	5.88	5.68	5.43
Ω, deg	98.76	317.18	251.62	274.03	178.06
ω, deg	83.48	281.68	211.43	301.14	125.72
M, deg	18.90	296.80	3.98	259.48	283.31
Orbit Class	Apollo	Apollo	Apollo	Apollo	Aten
Sequence B	2015 XC352	2016 LP48	2017 FU102	2009 JE1	2014 GR1
Classification	NHATS	-	NHATS	NHATS	NHATS
a, AU	1.016	1.05	1.29	1.37	1.30
e, -	0.15	0.40	0.29	0.29	0.28
i, deg	4.49	10.40	3.01	8.03	5.81
Ω, deg	98.76	87.26	12.69	46.15	168.58
ω, deg	83.48	51.87	231.18	223.20	89.99
M, deg	18.90	74.28	120.00	335.01	322.75
Orbit Class	Apollo	Apollo	Apollo	Apollo	Apollo

Note: M is calculated on 2019-04-27

interesting because it allows the system to be more flexible and visit an unrestricted range of NEAs of interest.

These trajectories are optimised according to the methodology presented in Section 6.4 using two different propulsion systems: (i) an EP system with a maximum thrust $T_{max} = 0.1$ N and specific impulse $I_{sp} = 3000$ s and (ii) an SSP system with a characteristic acceleration $a_c = 0.2$ mm/s^2. The characteristics of the MNR mission for each of the selected sequences A and B are reported in Tables 6.3 and 6.4, respectively. The tables report the departure and arrival date, TOF and ΔV of each leg of the sequences and the stay time. Note that the values of TOF and ΔV in brackets are those calculated by the ANN in the sequence search algorithm, which were used as initial guess for the optimiser. The ANN values are shown in the tables for comparison. In the sequence search, the stay time at each object was fixed to 100 days, so it is not reported in the tables.

The high values of TOF and ΔV, which characterise Sequence A and B, depended on how these sequences were selected. In fact, the sequences of asteroids which present large variations in the semi-major axis, eccentricity and inclination between one asteroid and the next asteroid in a sequence are chosen. This was done to show that the ANN can approximate the cost and duration for low-thrust transfers between NEAs even in cases where large changes in the orbital elements are involved. In fact, this generally results in more complex trajectories to be computed compared to when

Table 6.3: Mission parameters of the optimised MNR sequence A and comparison with the values estimated by the ANN (in brackets).

Leg	Propulsion	Departure	Arrival	TOF, days	ΔV, km/s	Stay Time, days
Earth - 2015 XC352	EP	2035-01-01	2037-06-04	885 (882)	8.86 (8.96)	97
	SSP	2035-02-08	2037-04-04	746	-	20
2015 XC352 - 2004 PJ2	EP	2037-09-09	2040-01-20	863 (878)	10.67 (11.14)	54
	SSP	2037-04-24	2039-12-15	943	-	400
2004 PJ2 - 162173 Ryugu	EP	2040-03-14	2042-09-21	920 (775)	15.03 (12.93)	100
	SSP	2041-01-18	2042-10-08	628	-	338
162173 Ryugu - 2011 MQ3	EP	2042-12-29	2045-02-11	775 (637)	12.29 (9.76)	21
	SSP	2043-09-20	2045-05-13	609	-	200
2011 MQ3- 2000 EW70	EP	2045-05-09	2047-03-17	677 (702)	10.43 (8.94)	–
	SSP	2045-11-29	2047-07-28	606	-	

smaller variations of the orbital elements are required.

Sequence A is plotted in Figures 6.7 and 6.8, while Sequence B is plotted in Figures 6.9 and 6.10. In particular, as a visual representation may help, Figures 6.7 and 6.9 show the NEAs visited as function of the time of flight for Sequence A and B, respectively, so that the difference in TOF between the values calculated by the ANN (solid green line) and the values calculated by the optimiser in the cases of EP (dotted blue line) and SSP (dashed red line) can be easily appreciated. For both sequences, the solar sailing case requires slightly more time to complete the mission. The sail characteristic acceleration a_c is chosen to be conservative according to Ref. [72], so generally longer stay times with respect to the EP case are necessary to catch up with the shorter transfer times. Additionally, it is noted that the network was trained to identify the cost and duration of low-thrust transfers with the spacecraft being equipped with a EP system specifically, so a larger error for the SSP case can be expected.

The heliocentric ecliptic-plane views of the optimised trajectories are shown in Figures 6.8 and 6.10 for Sequences A and B, respectively. Earth's orbit is plotted in solid black line, while the trajectory is represented by the dotted line. The portions of the trajectory where the stay at each asteroid is performed are highlighted in solid coloured lines, as specified in the legend.

The total ΔV required to complete the MNR missions with EP system are ΔV = 57.28 km/s for sequence A (Table 6.3) and ΔV = 54.97 km/s for sequence B (Table 6.4). Considering the first sequence flown by a system with dry mass m_{dry} = 350 kg, which is similar to the one of the *Deep Space 1* by NASA [222], where an EP system is used with specific impulse I_{sp} = 3000 s, the initial mass m_0 of the spacecraft is required to be:

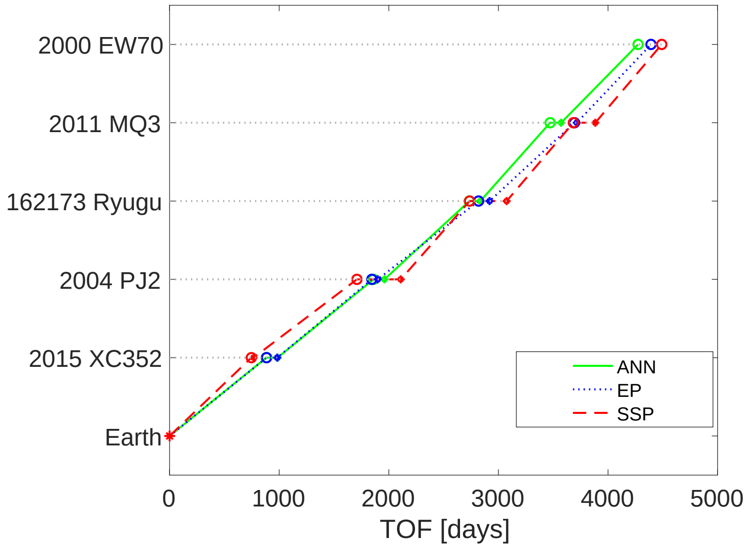


Figure 6.7: MNR sequence A: TOF per leg and stay time at each asteroid.

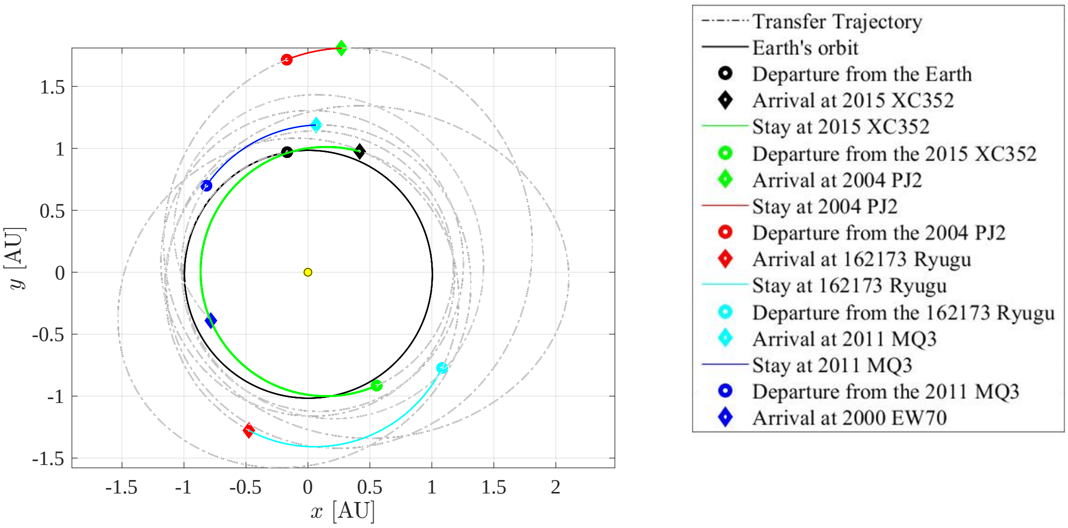


Figure 6.8: MNR sequence A: heliocentric ecliptic-plane view .

$$m_0 = m_{dry} e^{\left(\frac{\Delta V}{g_e I_{sp}}\right)} \approx 2400\,\text{kg} \tag{6.9}$$

with g_e being the standard acceleration due to gravity. This equates to a mass ratio $m_{dry}/m_0 \approx 0.15$. Considering the specifications in the 2014 GTOC problem [96], it is convenient to allow for a larger mass ratio (minimum of about $m_{dry}/m_0 = 0.4$) for a system propelled by a low-thrust engine. This does not apply to the SSP case, as ideally

Table 6.4: Mission parameters of the optimised MNR sequence B and comparison with the values estimated by the ANN (in brackets).

Leg	Propulsion	Departure	Arrival	TOF, days	ΔV, km/s	Stay Time, days
Earth - 2015 XC352	EP SSP	2035-01-01 2035-02-08	2037-06-04 2037-04-04	885 (882) 746	8.86 (8.96) -	51 277
2015 XC352 - 2016 LP48	EP SSP	2037-07-25 2037-11-28	2039-10-25 2039-08-26	821 (673) 635	8.12 (9.78) -	20 300
2016 LP48 - 2017 FU102	EP SSP	2039-11-14 2040-06-21	2042-02-10 2042-09-05	819 (859) 806	13.17 (12.25) -	24 20
2017 FU102 - 2009 JE1	EP SSP	2042-03-06 2042-09-26	2044-11-14 2045-07-27	984 (804) 1035	15.03 (13.22) -	100 31
2009 JE1 - 2014 GR1	EP SSP	2045-02-22 2045-08-27	2046-11-26 2047-09-26	642 (663) 760	9.79 (10.26) -	–

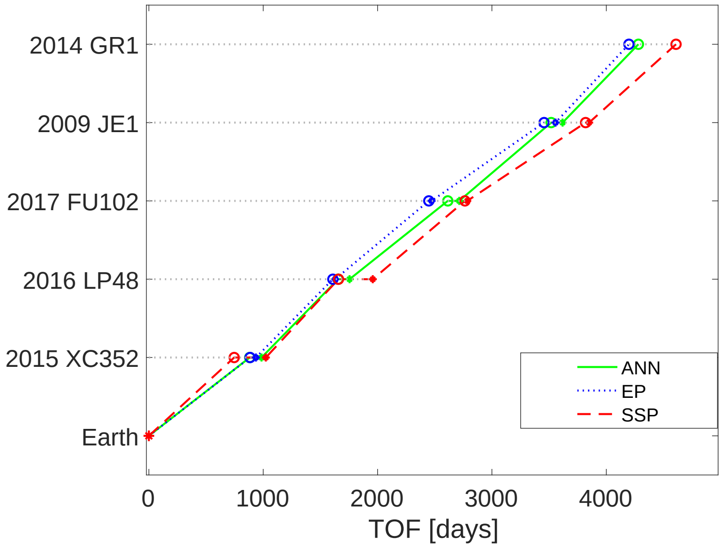


Figure 6.9: MNR sequence B: TOF per leg and stay time at each asteroid.

it does not required the use of any propellant to fly the sequences. Consequently, Sequences A and B can be flown with a near-term SSP system.

To take into account the requirement for m_{dry}/m_0 when using an EP system, the algorithm is run again including a constraint on $\Delta V \leq 5$ km/s per leg. One of the sequences obtained is selected and optimised (Sequence C). The orbital parameters of the visited bodies are detailed in Table 6.5 and the mission parameters are presented in Table 6.6. It can be noticed from Table 6.5 that the NEAs which are selected by SS-ANN in this case present more similar orbital characteristics. As a consequence

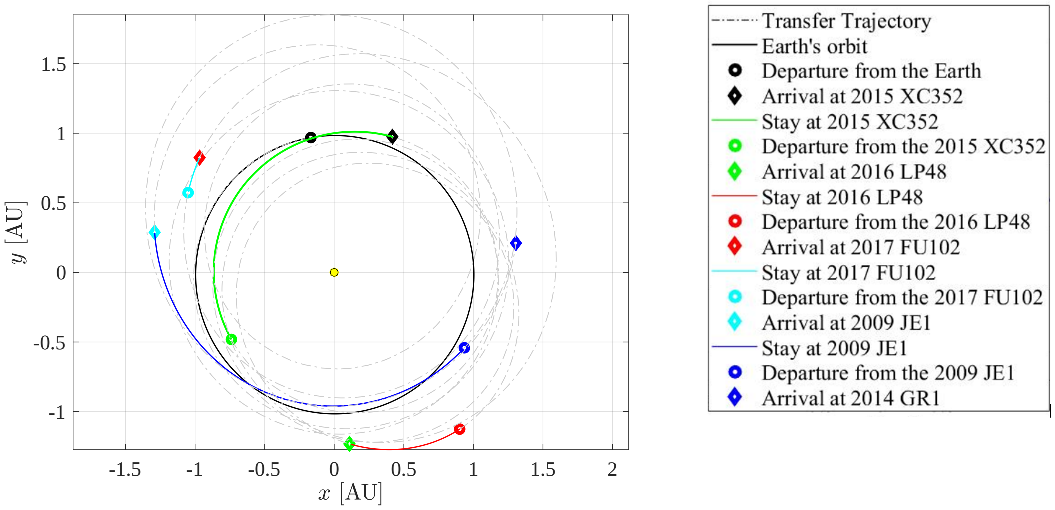


Figure 6.10: MNR sequence B: heliocentric ecliptic-plane view.

Table 6.5: Orbital parameters of the NEAs in the optimised Sequence C.

Sequence C	2014 WX202	2008 EA9	2015 VO142	2012 EP10	2013 CY	2014 UV210
Classification	NHATS	NHATS	NHATS	NHATS	NHATS	NHATS
a, AU	1.036	1.049	1.075	1.05	1.114	1.155
e, -	0.0589	0.0745	0.1259	0.1160	0.1342	0.1316
i, deg	0.41	0.44	0.28	1.03	0.78	0.60
Ω, deg	243.99	124.46	95.26	348.00	302.58	92.15
ω, deg	214.07	343.15	20.47	105.87	149.58	351.56
M, deg	51.76	134.18	336.18	293.59	114.38	134.91
Orbit Class	Apollo	Apollo	Apollo	Apollo	Apollo	Apollo

Note: M is calculated at 2019-04-27

of limiting the ΔV available per transfer leg, smaller variation in the semimajor axis a, eccentricity e and inclination i can be realised compared to Sequences A and B (which, therefore, also reduces the TOF required per transfer leg in Sequence C).

The heliocentric ecliptic-plane view of the complete trajectory is plotted in Figure 6.11. In Sequence C, six asteroids are visited in less than 10 years, requiring a total ΔV of 17.95 km/s. From Eq.(6.9), a mass fraction of 0.54 is obtained in this case.

Despite it does not allow to realise as great changes in a, e and i as in Sequences A and B, limiting the ΔV available for each leg of the sequence allows to meet the requirement on the dry mass ratio of minimum 0.4 and, consequently, on the maximum ΔV of the mission. In this way, the sequences identified by the ANN can be realistically flown with the selected EP system.

Table 6.6: Mission parameters of the optimised MNR sequence C and comparison with the values estimated by the ANN (in brackets).

Leg	Propulsion	Departure	Arrival	TOF, days	ΔV, km/s	Stay Time, days
Earth - 2014 WX202	EP	2035-01-21	2036-07-24	550 (570)	2.50 (3.13)	87
2014 WX202 - 2008 EA9	EP	2036-09-29	2038-02-09	498 (456)	3.02 (2.80)	91
2008 EA9 - 2015 VO142	EP	2038-05-11	2039-10-28	535 (555)	3.25 (3.04)	68
2015 VO142 - 2012 EP10	EP	2040-01-04	2041-05-01	483 (402)	2.92 (2.97)	100
2012 EP10 - 2013 CY	EP	2041-08-09	2042-12-15	493 (514)	2.99 (2.76)	56
2013 CY - 2014 UV210	EP	2043-02-09	2044-07-30	537 (557)	3.27 (2.91)	–

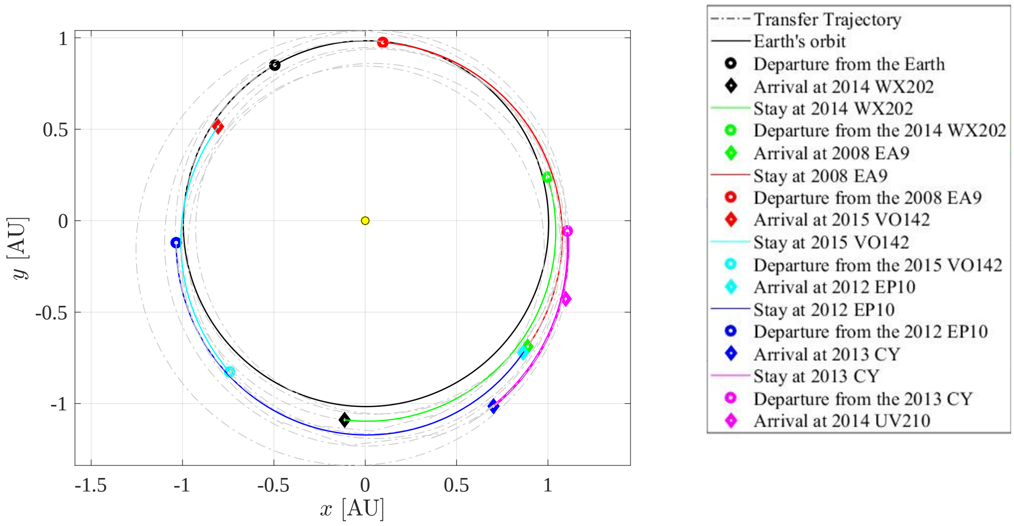


Figure 6.11: MNR sequence C: heliocentric ecliptic-plane view.

However, the successful optimisation of Sequences A, B, and C shows that the ANN is able to approximate well the cost and duration for low-thrust rendezvous transfers between NEAs, also when involving large changes of the orbital parameters.

6.4.2 ANN Prediction Error

From the sequences studied above, the differences in TOF and ΔV between the values obtained by the ANN and the results obtained from the trajectory optimisation are generally limited. To quantify this deviation, the average percentage error $\mathscr{E}$ is calculated by using Eq.(4.2) and is computed considering only the TOF and ΔV for each EP transfer leg, since these are the two parameters predicted by the network.

Table 6.7: Average percentage errors between the ANN outputs and the optimised results.

Sequence	$\mathscr{E}_{TOF}$ for EP	$\mathscr{E}_{\Delta V}$ for EP	$\mathscr{E}_{TOF}$ for SSP
A (Table 6.3)	7.8%	10.8%	13.7%
B (Table 6.4)	8.9%	9.1%	13.1%
C (Table 6.6)	6.7%	9.8%	–
Mean Error	7.8%	9.9%	13.4%

The average percentage errors are shown for each of the sequences studied in Table 6.7. When considering the error computed for sequences A, B, and C, the overall mean error is 7.8% for TOF and 9.9% for ΔV. This suggests that an ANN can obtain a satisfactory accuracy for the preliminary orbit design.

This confirms the validity of the methodology and the capability of the neural network, which is designed and trained as described in Chapter 4, to predict the cost and duration of a low-thrust transfer given the orbits of the departure and arrival asteroids.

Larger differences in TOF and ΔV with respect to the average values can sometimes occur. This is due to the fact that in the SS algorithm the stay times of 100 days is fixed and, consequently, the departure times of each leg ($t_0 = T_0 + t_s$) are imposed *a priori*, with T_0 being the mission duration until that point and t_s the fixed stay time. Differently, in the trajectory optimisation process, the stay time adapts to allow for the most convenient departure and arrival points depending on the relative position of the two bodies. This can result in a difference in departure date, arrival date, and stay time which inevitably affects the computed TOF and ΔV of the transfer.

It should be also highlighted that this error analysis is performed between the network outputs and the optimised results. It is described how the network is trained with a training database of transfers which are calculated using the linear-trigonometric shape introduced by the shape-based method [115]. As this shape-based method can compute sub-optimal solutions, it is expected that some level of error is already introduced by this, since the ANN replicates the shape-based model after the training. However, it is chosen to compute this error of the ANN outputs with respect to the optimal values (instead of the shape-based values) because ultimately these are the results which we aim at reproducing as closely as possible. Also, an analysis on the differences between the ANN output and the results from the shape-based method (i.e. the targets) is already performed in the regression and performance analysis, offered in Figure 4.9.

As an additional note, the mean percentage error for the case where SSP is used is also reported in Table 6.7, showing an overall error of 13.4% for the TOF. The error is slightly higher compared to the EP case, as expected, because the network

Table 6.8: Orbital parameters of the NEAs in the optimised Sequence X.

Sequence X	2015 XC352	2019 PC3	2015 GY	2014 NG65	2011 MQ3
Classification	NHATS	-	PHA	PHA	NHATS
a, AU	1.016	1.166	1.068	1.363	1.12
e, -	0.15	0.34	0.20	0.39	0.11
i, deg	4.49	10.13	11.81	8.35	5.68
Ω, deg	98.76	299.68	207.78	97.37	274.03
ω, deg	83.48	209.02	108.65	305.26	301.14
M, deg	18.90	94.80	64.30	293.60	259.48
Orbit Class	Apollo	Apollo	Apollo	Apollo	Apollo

Note: M is calculated on 2019/04/27

was specifically trained with EP transfers. However, this is still an acceptable error demonstrating that this methodology also allows for preliminary solar sailing mission design with near-term sail performance.

6.4.3 Example of a Non-Feasible Sequence

According to the optimisation process presented in Section 6.4, sequences for which a feasible solution per leg is not found by the optimiser are discarded. In this section, one sequence which presents an unfeasible leg is investigated with the purpose of identifying the reasons behind it.

The selected sequence, which is referred as Sequence X, has no optimal solution found by the optimiser for the third leg of the trajectory. The NEAs to be visited in Sequence X are characterised in Table 6.8. Similarly to the procedure followed for Sequences A, B and C, the legs in Sequence X are optimised and the characteristics of the mission are reported in Table 6.9. It can be noticed that no feasible solution is found for the third leg, i.e. for the transfer from asteroid 2019 PC3 to asteroid 2015 GY departing on 2040-01-27.

To identify the reason why the optimisation failed on the third leg of Sequence X, the shape-based method was run again on this particular leg. If a feasible solution is found with a similar ΔV and TOF to those calculated by the ANN, the issue has to be in the optimisation procedure. Conversely, if the shape-based method fails in finding a solution, it means that a feasible transfer for this leg cannot be represented by the linear-trigonometric shape defined in Section 3.2.2. In the latter case, the reason why the ANN cannot detect this is because the ANN is trained with low-thrust transfers for which the shape-based method is able to find a successful solution, i.e. the ANN has not learnt how to spot unfeasible transfers.

By running the shape-based method, no feasible solution is obtained for the third

Table 6.9: Mission parameters of the optimised legs in the MNR sequence X and comparison with the values estimated by the ANN (in brackets).

Leg	Propulsion	Departure	Arrival	TOF, days	ΔV, km/s	Stay Time, days
Earth - 2015 XC352	EP	2035-01-01	2037-06-04	885 (882)	8.86 (8.96)	87
	SSP	2035-02-08	2037-04-04	746	-	146
2015 XC352 - 2019 PC3	EP	2037-08-07	2039-09-28	782 (797)	12.42 (12.84)	120
	SSP	2037-07-20	2039-04-12	631	-	168
2019 PC3 - 2015 GY	EP	2040-01-27	2042-03-24	No optimal solution found		39
	SSP	2039-09-27	2041-09-22			92
2015 GY - 2014 NG65	EP	2042-05-02	2044-10-09	891 (731)	14.67 (11.32)	20
	SSP	2041-12-23	2044-11-10	1053	-	109
2014 NG65 - 2011 MQ3	EP	2044-10-29	2047-02-16	840 (802)	13.61 (14.55)	–
	SSP	2045-02-27	2047-08-08	892	-	

leg of Sequence X. Differently, a solution is found for this leg when a higher maximum thrust $T_{max} \geq 0.4$ N is allowed. This is not included in the solution space considered for this application where $T_{max} = 0.1$ N is considered. Consequently, the ANN could not identify that this particular transfer leg is unfeasible with the chosen propulsion system. In this case, the network returns an incorrect guess of the transfer cost and duration (not having been trained on such or similar transfers).

The other legs, for which an optimal solution is found, present a good accuracy of the results, similarly to what is observed in the other sequences A, B and C analysed. However, this example offers the opportunity to comprehend the limitation of the method. This is related to how the network is trained. In fact, once the ANN training is completed, the ANN replicates the behaviour of the method used to compute the samples included in the database, i.e. the shape-based method in this case. Also, despite the shape-based method can determine whether a transfer is unfeasible for the given propulsion systems, unfeasible transfers were not included in the training database and, as a consequence, the ANN was not trained to detect them. Nonetheless, the final steps of the proposed optimisation methodology, where the OCP is solved by an optimiser, are implemented to identify this and calculate optimal (feasible) solutions for MNR missions.

6.4.4 Computational Time and Accuracy Analysis

One of the advantages of using an ANN to determine the cost and TOF of the transfers between pairs of asteroids is the reduction in the computational time. This section aims at demonstrating this by comparing the computational time and performance of the trained network with other methods used in previous works. For instance, the algorithm used by Peloni et al. [2], where the tree of multiple-NEA trajectories is obtained by approximating the transfers using a shape-based method, is considered.

It should be noted that a shape-based approach for solar sailing has been developed by Peloni et al. and used for the analysis in Ref. [2]. In essence, they propose the use of a set of shaping functions for solar sailing in the *coplanar* case. This takes into account the characteristics of the solar sail dynamics. The shaping functions which describe the coplanar trajectory of a solar sail are an exponential-trigonometric shape to describe the evolution of the semilatus rectum p and linear-trigonometric shapes to describe the two in-plane elements f and g [2, 219].

Peloni et al. performed the search on a set of launch dates spanning about 10 years with a step size of three months, starting from the launch date 2019-11-28, for a total of 41 launch dates. The input database of NEAs included only PHAs and NHATS asteroids, for a total of 1,801 objects. The propulsion system used by Peloni et al. is a solar sail with $a_c = 0.2$ mm/s^2.

The objective of the analysis is to compare the time to estimate successfully the cost of a low-thrust transfer when using a shape-based method or the ANN and the average time required to complete a sequence-search run. The same database and the same launch dates, used in Ref. [2], are utilised here so that the same sequences can be obtained and a head-to-head comparison can be performed.

A database is generated, containing PHA and NHATS asteroids only and selecting the NHATS objects with the same criteria used in Ref. [2]. This results in a database containing a total of 2,768 objects. This is almost a thousand NEAs more than the one used in Ref. [2], due to the fact that the database used by Peloni at al. was generated in August 2015, and numerous NEAs have been discovered since. To partially take into account the decrease in the sail acceleration when it is not directed along the Sun-spacecraft direction, a EP system with $a_{max} = 0.1$ mm/s^2 is used in this analysis. This value of a_{max} is calculated from $T_{max} = 0.1$ N and the mass of the system of 1000 kg at departure. The fact that a different propulsion system is used for the analysis does not affect the computational time required by the SS algorithm, however it may have an effect on the accuracy of the results, so it will be taken into account during the accuracy analysis.

In Ref. [2] the runs were conducted on three computers: two with a 3.4 GHz Intel Core i7 and one with a 2.3 GHz AMD Opteron 6376. Peloni et al. give an indication of the required computational time considering only the runs carried out in the latter, which is the slowest machine. During this analysis a machine with a 3.4 GHz Intel Core i7 was used. Taking into account the performance test benchmark results of the two machines, the CPU Mark [223] suggests that the machine with a 2.3 GHz AMD Opteron 6376 is roughly 20% slower in computational speed than the 3.4 GHz Intel Core i7. Although additional factors can play a role in affecting the computing speed of the machines, this approximation is sufficient to evaluate the order of magnitude of the difference in computational power required for the two methodologies. For

Table 6.10: Speed comparison between the SS algorithm with the use ANN and without (i.e. the methodology proposed in Ref. [2]).

Simulation	SS without ANN [2]	SS-ANN
Low-thrust transfer	48 s	0.49 s
Sequence-search run	33 days	7.66 hours

this reason the time required for the simulations in Ref. [2] will be reduced by 20% for a fair comparison.

Firstly, the computational time required for the estimation of the cost and TOF of a low-thrust transfer is analysed. Each successful run of the SSP shape-based method in Peloni et al. takes on average about 60 s on 2.3 GHz AMD Opteron 6376, so we assume it would take roughly 48 s on average on the machine with a 3.4 GHz Core i7 (which is roughly 20% faster, as mentioned). Differently, it takes less than a second (0.49 s on average) for the ANN to perform the same estimation. This already suggests the huge impact of using an ANN on the computational time of a sequence-search run where thousands of transfers are investigated.

In Ref. [2], for 41 launch dates the search found 4,800 sequences with five encounters and at least one PHA. The authors specify that, for each launch date, a sequence-search run took on average 33 days (41.3 days on the 2.3 GHz AMD Opteron 6376) to complete. So, considering that 4,800 sequences are found for 41 launch dates, we can assume that, on average, almost 120 sequences per launch date are found. For the launch date 2025-04-30 the sequence-search run takes 7.66 hours to find the same number of sequences, when the ANN is used, meaning that the ANN makes the algorithm almost 100 times faster. Table 6.10 summarises the comparison in the computational time between the method in Ref. [2] and SS-ANN for the simulation of a low-thrust transfer and a sequence-search run.

Acknowledging the difference in time for the computation of a single low-thrust transfer between the shape-based method used in Ref. [2] and the ANN, it is expected that, as the number of objects in the database grows or as the number of sequences to analyse increases, the difference in computational time increases exponentially and the advantage of using an ANN is even more considerable.

Once the improvement in computational speed given by the use of an ANN is assessed, a comparison of the accuracy of the two methods is also conducted. This is done to ensure that the improvement in speed does not come at the expenses of the solution accuracy. Peloni et al. [2] present and optimise the sequence found for the launch date of 2025-04-30. The same sequence of asteroids is obtained during our sequence-search run, and it is optimised.

The results are detailed in Table 6.11, where the TOF obtained by the ANN is compared with the TOF obtained by Peloni et al. using their shape-based method.

Table 6.11: Mission parameters of a MNR mission departing from Earth on 2025-04-30. Comparison between the method used in Ref. [2] and SS-ANN.

Leg	Method	TOF, days	Optimal TOF, days	$\mathscr{E}_{TOF}$ (vs. Optimal)
Earth - 2000 SG344	Ref. [2]	680	657	3.5%
	SS-ANN	634		3.5%
2000 SG344 - 2015 JD3	Ref. [2]	500	436	14.7%
	SS-ANN	414		5.1%
2015 JD3 - 2012 KB4	Ref. [2]	644	584	10.3%
	SS-ANN	603		3.3%
2012 KB4 - 2008 EV5	Ref. [2]	647	576	12.3%
	SS-ANN	639		10.9%
2008 EV5 - 2014 MP	Ref. [2]	625	560	11.6%
	SS-ANN	606		8.2%
Average	Ref. [2]			10.5%
	SS-ANN			6.2%

To measure the accuracy of both methods, the estimated TOF values are compared to the optimal TOF values. The percentage error $\mathscr{E}_{TOF}$ (Eq.(4.2)) is reported in the last column of the table. The ANN achieves a smaller error with respect to the shape-based method in Ref. [2], which can also be due to the different propulsion systems used in the two cases. However, we can conclude that using a well-trained ANN maintains a good level of accuracy, while dramatically decreasing the computational time required.

7

Design of Multiple NEA Rendezvous Missions with Sample Return to Earth

The SS-ANN methodology is successfully applied to the case of multiple NEA rendezvous missions in the previous chapter [39]. Here, the possibility to use the SS-ANN methodology for MNR missions with sample return to Earth is investigated. In fact, sample return missions to NEAs are invaluable for the scientific community to learn more about the initial stages of the solar system formation and life evolution [82, 20].

Sample return missions to planets and small bodies represent one of the biggest challenges for space engineering. Although rendezvousing multiple NEAs with sample return to Earth increases the scientific return of those missions, they are highly demanding in terms of orbital energy, with a high required velocity change ΔV.

To account for the differences of the mission scenario, the sequence search algorithm is re-designed to allow for the sample return to Earth, as described in Section 7.1. The SS algorithm can also be implemented to compute NEA sequences so that specific asteroids, for which a sample return would be more valuable, can be targeted (Section 7.2) and to take into account the interest value of the asteroids visited for the selection of NEAs in a sequence (Section 7.3). In Section 7.4, three NEA sequences with sample return to Earth are selected and further analysed to optimise the trajectory and assess the performance of the proposed methodology.

7.1 SS-ANN for Sample Return to Earth

The mission scenario for MNR with sample return to Earth is represented in Figure 7.1, with Ai and $i = 1,2,...$ indicating the NEA objects located on different orbits. Similarly to the MNR missions described in Chapter 6, the spacecraft departs from

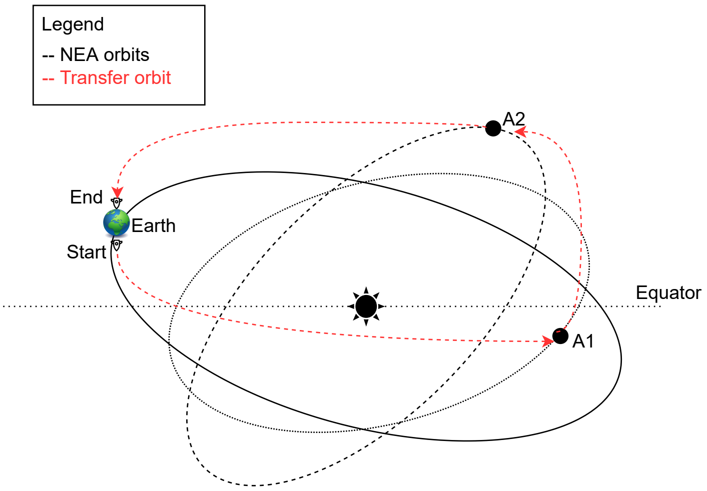


Figure 7.1: MNR with sample return to Earth: mission scenario.

Earth at a given departure date and transfers to the first asteroid in the sequence to rendezvous and, eventually, collects a sample. The spacecraft then transfers to the next asteroid and the procedure is iterated. Finally, on the last leg, the spacecraft performs the transfer to return to Earth.

For the generation of the training data, the same database of NEAs [21] which is used in the MNR problem, including PHA and NHATS, is used. The database generation process to train the network is similar to the one described in Section 6.2, where 100 NEAs are selected and permuted to calculate the low-thrust transfers between pairs of objects using the shape-based method with linear-trigonometric shape. The same bounds presented in Table 3.3 on the parameters for the GA in the shape-based method are used. However, in this case, in addition to the transfers from Earth to NEAs and transfers between NEAs, also transfers from NEAs to Earth are computed and included in the training database. This is done so that the ANN can account for the additional portion of the mission where the return to Earth is performed. The same network architecture is used and the same training algorithm is used, leading to a performance of the network comparable to the one obtained for MNR missions without sample return and outlined in Section 4.4 and Section 6.2. The final correlation coefficient of 0.9732 and validation-set MSE of 0.1211 are obtained.

The sequence search algorithm, implemented to identify the most promising sequences of asteroids to visit and return to Earth, is illustrated in Figure 7.2. To design an MNR mission with a sample return to Earth, a 10-year mission, of which a timeline is presented in Figure 7.3, is divided into three phases and the selection of the next body to visit depends on each of these phases. The SS logic is implemented to account for these phases. During the first phase, going from the departure to the

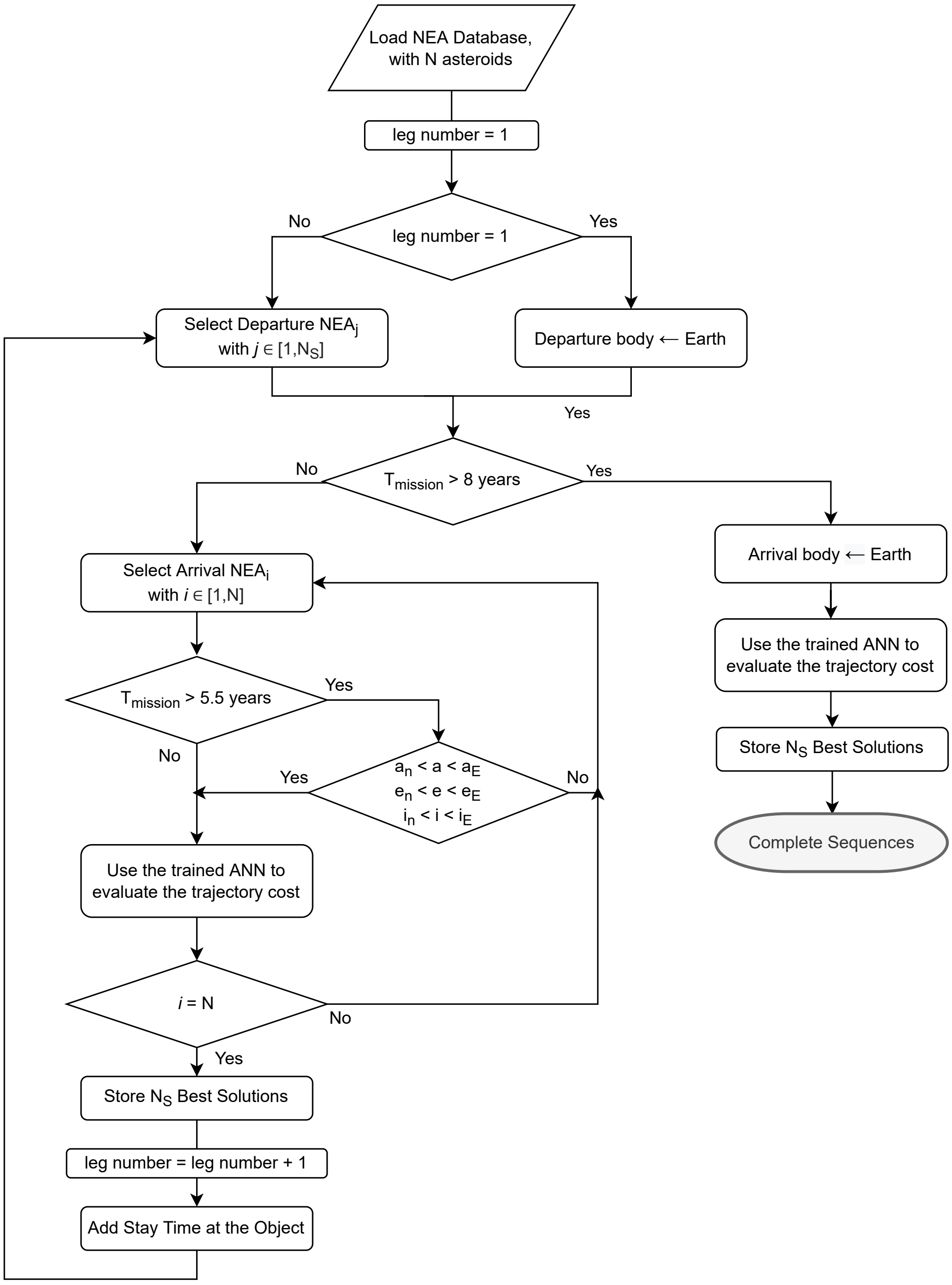

Load NEA Database, with N asteroids
leg number = 1
leg number = 1
No
Yes
Select Departure NEA$_j$ with j ∈ [1,N$_S$]
Departure body ← Earth
Yes
T$_{mission}$ > 8 years
No
Yes
Arrival body ← Earth
Use the trained ANN to evaluate the trajectory cost
Store N$_S$ Best Solutions
Complete Sequences
Select Arrival NEA$_i$ with i ∈ [1,N]
T$_{mission}$ > 5.5 years
Yes
No
a$_n$ < a < a$_E$
e$_n$ < e < e$_E$
i$_n$ < i < i$_E$
Yes
No
Use the trained ANN to evaluate the trajectory cost
i = N
No
Yes
Store N$_S$ Best Solutions
leg number = leg number + 1
Add Stay Time at the Object

Figure 7.2: Sequence search algorithm with sample return to Earth.

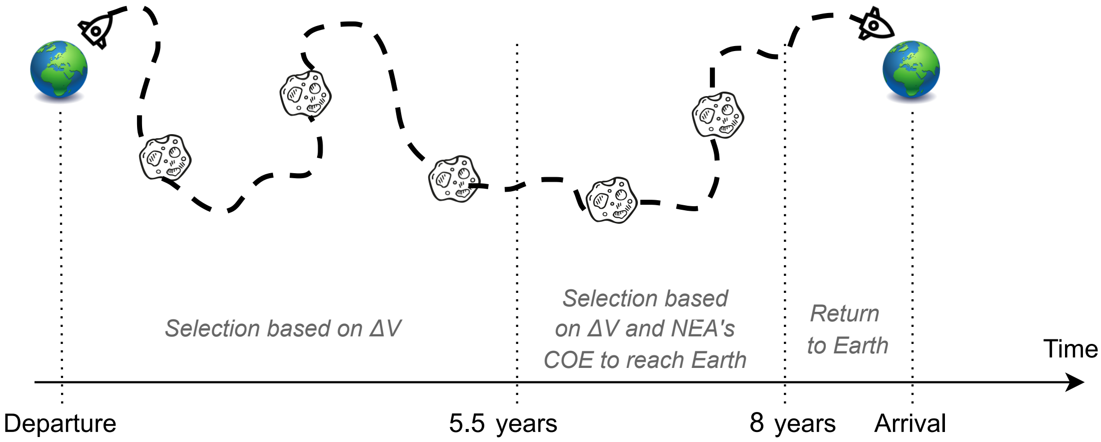


Figure 7.3: Timeline of MNR missions with return to Earth.

first 5.5 years of the mission, NEAs are selected only based on the propellant mass consumption or ΔV to transfer to them. During the second phase, the NEAs are pruned so that the semi-major axis a, eccentricity e, and inclination i of the next visited NEA are included between those of the last-visited NEA and the Earth, i.e.:

$$a_n \leq a \leq a_E \tag{7.1}$$

$$e_n \leq e \leq e_E \tag{7.2}$$

$$i_n \leq i \leq i_E \tag{7.3}$$

where the subscript n indicates the n-th asteroid visited and the subscript E refers to Earth. In essence, this constrains the search to target asteroids which are incrementally more similar to Earth in terms of their orbital elements. The expressions are defined for the case with $a_E \geq a_n$, $e_E \geq e_n$, and $i_E \geq i_n$, but they can be inverted when the opposite case occurs. This pruning of the NEAs is performed after the mission duration reaches 5.5 years (i.e. in the second half of the mission). At this point, NEAs are selected from the pruned dataset based on the ΔV required.

After the mission duration of a partial sequence reaches 8 years (value chosen to allow enough time for a feasible return leg to Earth for the 10-year mission), the third phase starts. It consists in the return to Earth (final leg), where Earth is targeted as the arrival body. Finally, the solutions with the minimum propellant mass expenditure are stored and considered as complete sequences.

Figure 7.4 shows sequences of five asteroids with return to Earth found by the SS-ANN platform for launch date 2035-08-24. As the number of permutations between asteroids grows rapidly with the number of legs, only the first three legs are plotted in full and, for illustration purposes, only one complete sequence is fully shown.

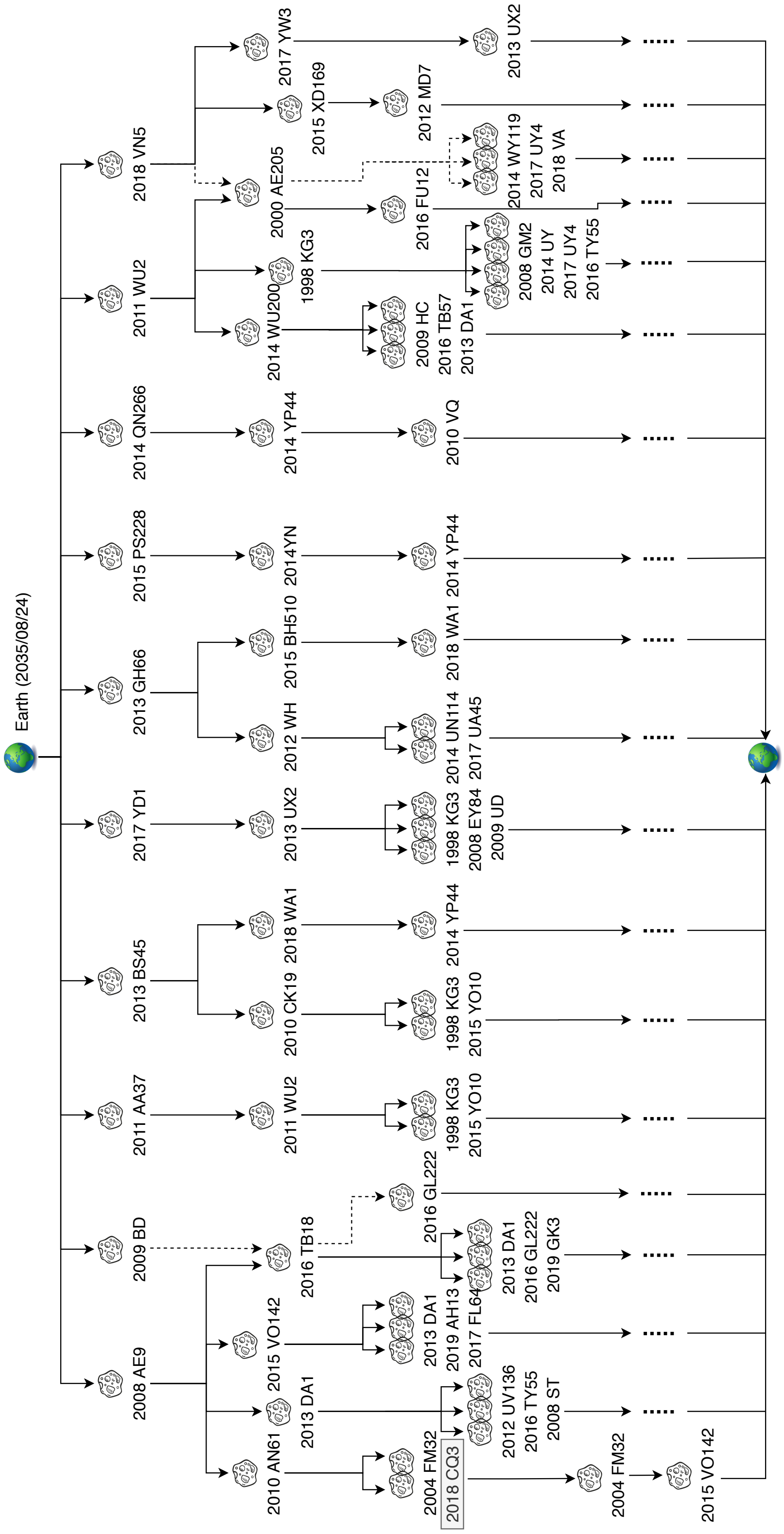


Figure 7.4: Tree graph of the MNR sequences with sample return to Earth computed by SS-ANN for the launch date 2035-08-24.

7.2 Asteroid Targeting

The missions which successfully performed a sample return to Earth were designed to target one asteroid which is reachable with the given propulsion system and is particularly interesting from the scientific point of view. For instance, 25143 Itokawa and 162173 Ryugu were selected as the target asteroids for the missions Hayabusa and Hayabusa2, respectively, because of their sizes and because they are reachable using an ion engine with a feasible ΔV [84, 83]. Similarly, NASA selected the asteroid 101955 Bennu for the mission OSIRIS-REx as it is rich in pristine carbonaceous material, which is a key element in organic molecules necessary for life [82]. In all three missions, the asteroids were primarily chosen because the sample collections and analyses were expected to improve the knowledge on the formation and evolution of the planets and, in particular, of Earth, as well as the origin of water and organic matter.

The sequence search algorithm can be enhanced in a way that enables to target a specific asteroid which is particularly interesting for the scientific community and for which a sample return would be more valuable. Figure 7.5 schematises the implemented algorithm. This algorithm searches for sequences so that the first half of the mission is focused on targeting the chosen asteroid of interest, while the second half is focused on targeting the Earth for a safe return. In the algorithm, in the first half of the mission, where the target leg has not been performed yet, the variable 'target leg' is set to false. The spacecraft visits as many asteroids as possible for observation and/or sample collection, while flying from Earth to the target body and back to Earth. To this end, a pruning of the asteroids, which are visited between the departure from Earth and the arrival at the target body, is also performed. Only the NEAs with semimajor axis, eccentricity and inclination included between those of the last-visited NEA (or Earth on the first leg) and the target asteroid are considered, i.e.:

$$a_n \leq a \leq a_T \tag{7.4}$$

$$e_n \leq e \leq e_T \tag{7.5}$$

$$i_n \leq i \leq i_T \tag{7.6}$$

where n indicates the n-th asteroid visited (or Earth for the first leg) and T the targeted asteroid. After two legs are processed, it is assumed that the spacecraft has reached an orbit from which it is possible to conveniently, in terms of cost and duration, transfer to the target body. Thus, the arrival body of the third leg is the target asteroid.

Once the target body is visited and the sample is collected, the variable 'target leg' is set to true. At this point, a second pruning of the asteroids, which are visited between the departure from the target body and the return to Earth, is performed. This is done according to Eqs.(7.1) to (7.3), to guarantee that the spacecraft can successfully return to Earth while visiting as many asteroids as possible along its way back to Earth. This

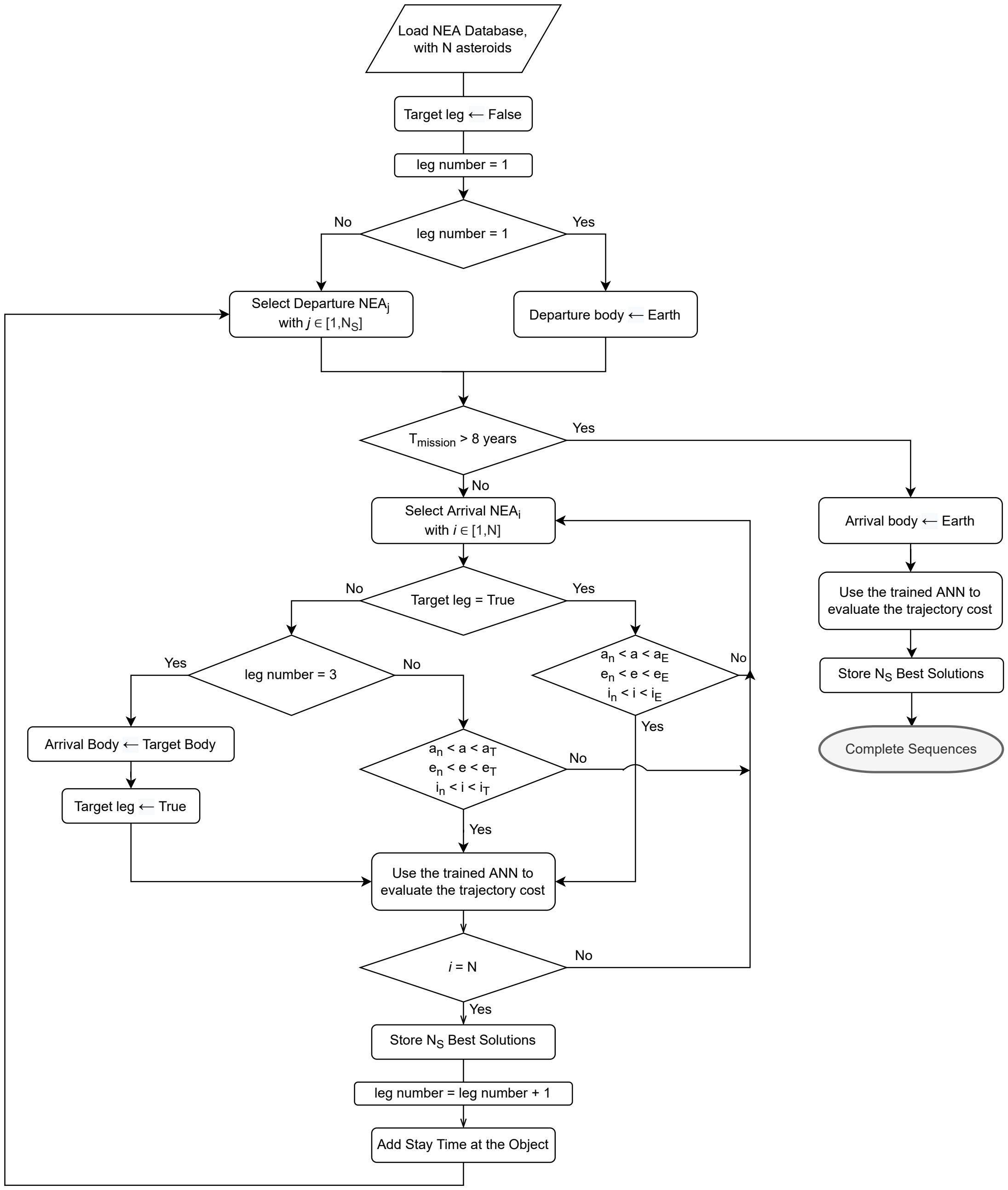

Load NEA Database, with N asteroids
Target leg ← False
leg number = 1
No
leg number = 1
Yes
Select Departure NEA$_j$ with $j \in [1,N_S]$
Departure body ← Earth
$T_{mission}$ > 8 years
Yes
No
Select Arrival NEA$_i$ with $i \in [1,N]$
Arrival body ← Earth
Use the trained ANN to evaluate the trajectory cost
No
Target leg = True
Yes
$a_n < a < a_E$
$e_n < e < e_E$
$i_n < i < i_E$
No
Store N_S Best Solutions
Yes
leg number = 3
No
Yes
Complete Sequences
Arrival Body ← Target Body
$a_n < a < a_T$
$e_n < e < e_T$
$i_n < i < i_T$
No
Target leg ← True
Yes
Use the trained ANN to evaluate the trajectory cost
i = N
No
Yes
Store N_S Best Solutions
leg number = leg number + 1
Add Stay Time at the Object

Figure 7.5: Sequence search algorithm to target one NEA of interest and return to Earth.

is schematised in Figure 7.6, where the timeline of the MNR missions with asteroid targeting and sample return is summarised.

For the scenario presented in this work, asteroid 162173 Ryugu (which was presented in Figure 2.4(b)) is targeted. 162173 Ryugu is a PHA and NHATS of the Apollo group with a diameter of approximately 1 km and has an unaltered or barely altered

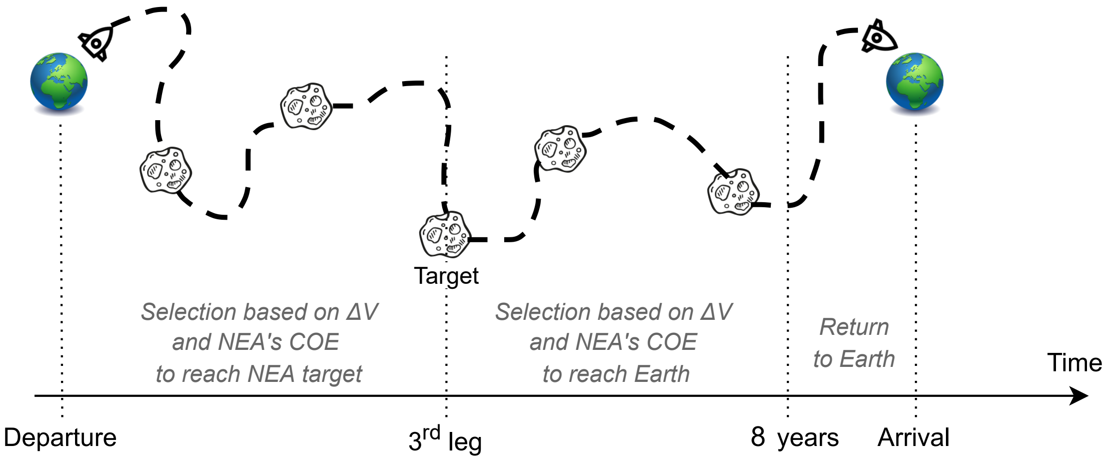


Figure 7.6: Timeline of MNR missions with asteroid targeting and return to Earth.

composition (C-type) [224]. Figure 7.7 shows the tree graph of the sequences of five asteroids, which target 162173 Ryugu in the third leg and, finally, perform a sample return to Earth. These sequences are calculated by the SS-ANN methodology for the departure date 2035-08-24.

7.3 Interest Value

In the applications where the SS-ANN methodology has been applied so far, the objective function which was minimised in the NEA selection process is the TOF (for MADR and MNR) and the ΔV (for MNR with sample return to Earth). In general, the choice of one or the other objective function depends on the constraints or requirements of a given mission; this can be easily accounted for in the SS algorithm by changing the objective function. In this section, the possibility of favouring the NEAs in a sequence on the basis of their value of interest, for which a sample return can be more valuable, is explored.

As many asteroids remain *unclassified* due to the lack of quality data or limited opportunities for ground observation [225], it is considered that the NEAs which are bigger in size are more valuable to observe and return samples of. In fact, larger asteroids are generally rarer and more suitable for landing and for the sample collection.

Apart from the asteroids which have been visited by a spacecraft in the past, the size and shape of most asteroids is yet unknown. Although most asteroids have irregular shape and only few of them are close to being spherical, the size of an asteroid can be estimated, to a first approximation, as the diameter of an equivalent sphere with a uniform surface, given its *absolute magnitude*, H, and assumed *geometric albedo*, a_d. The diameter (in km) of an asteroid can be estimated as follows [226]:

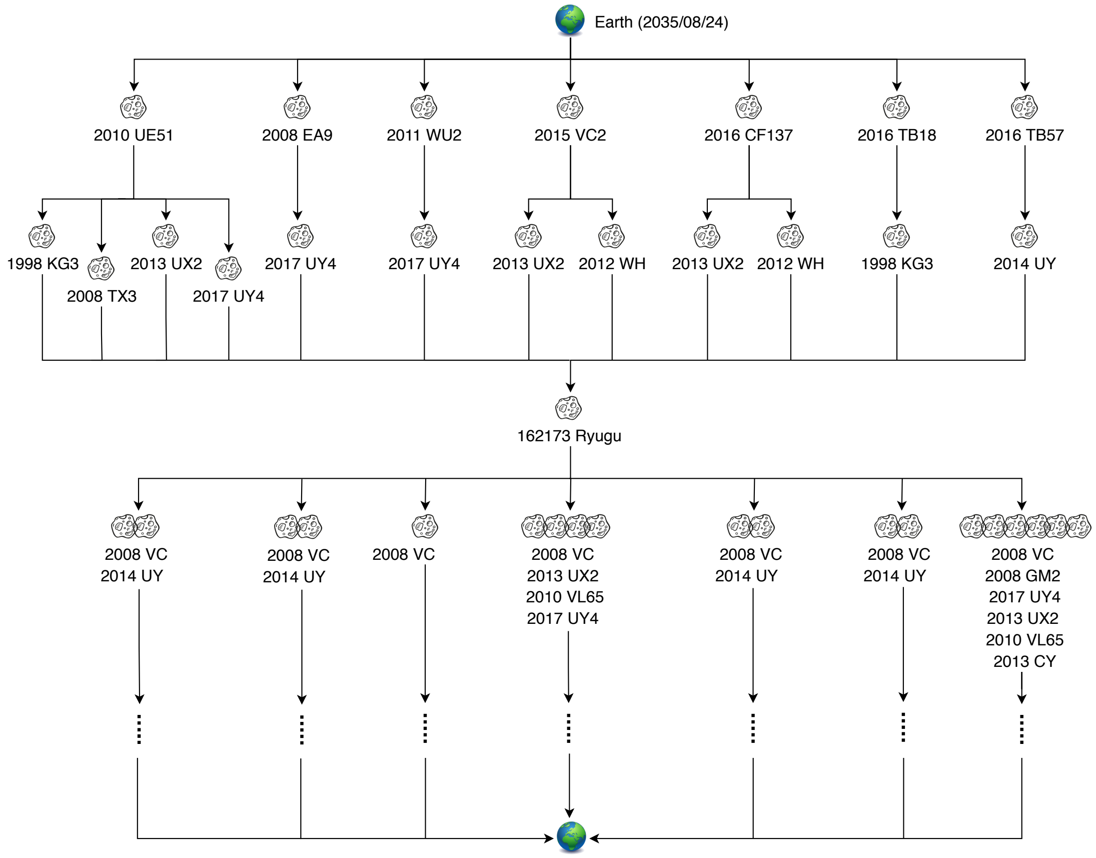


Figure 7.7: Tree graph of the MNR sequences targeting 162173 Ryugu and returning to Earth computed by SS-ANN for the launch date 2035-08-24.

$$d = 10^{(3.1236-0.5\log_{10}(a_d)-0.2H)} \quad \text{km} \tag{7.7}$$

where the albedo, a_d, is generally assumed based on the spectral class corresponding to the assumed composition of the asteroid and an average value is typically used [227]. Due to the conceivable uncertainty in both H and a_d, Eq.(7.7) provides an approximate estimation of the size of an asteroid [228]. However, considering that in Eq.(7.7) H has a larger impact than a_d in determining the asteroid's size, we will assume in the following that an asteroid with a smaller H is characterised by a larger size and represents a more interesting candidate to visit in a sequence.

To classify the obtained sequences on the basis of their scientific interest, the *interest value*, I_V, of each sequence is introduced and defined as the negative sum of the absolute magnitudes of all the visited asteroids, i.e.:

$$I_V = -\sum_{i=1}^{N_A} H_i \tag{7.8}$$

where H_i is the absolute magnitude of the i-th asteroid visited and $i \in [1, N_A]$, with N_A being the number of asteroids visited in the sequence. The greater the interest value (in magnitude and sign, i.e. the closer to zero), the larger the asteroids visited and the more interesting the sequence. It is important to underline that the interest value definition can be changed by the mission designer depending on the objectives of the mission.

To take into account the interest value for the selection of the most convenient sequences to fly, within the SS-ANN methodology, an *appealing factor A* is associated with each sequence as the weighted sum of its total ΔV and total I_V of the mission, which can be expressed as follows:

$$A = \gamma \Delta \hat{V}_{tot} - (1 - \gamma) \hat{I}_V \tag{7.9}$$

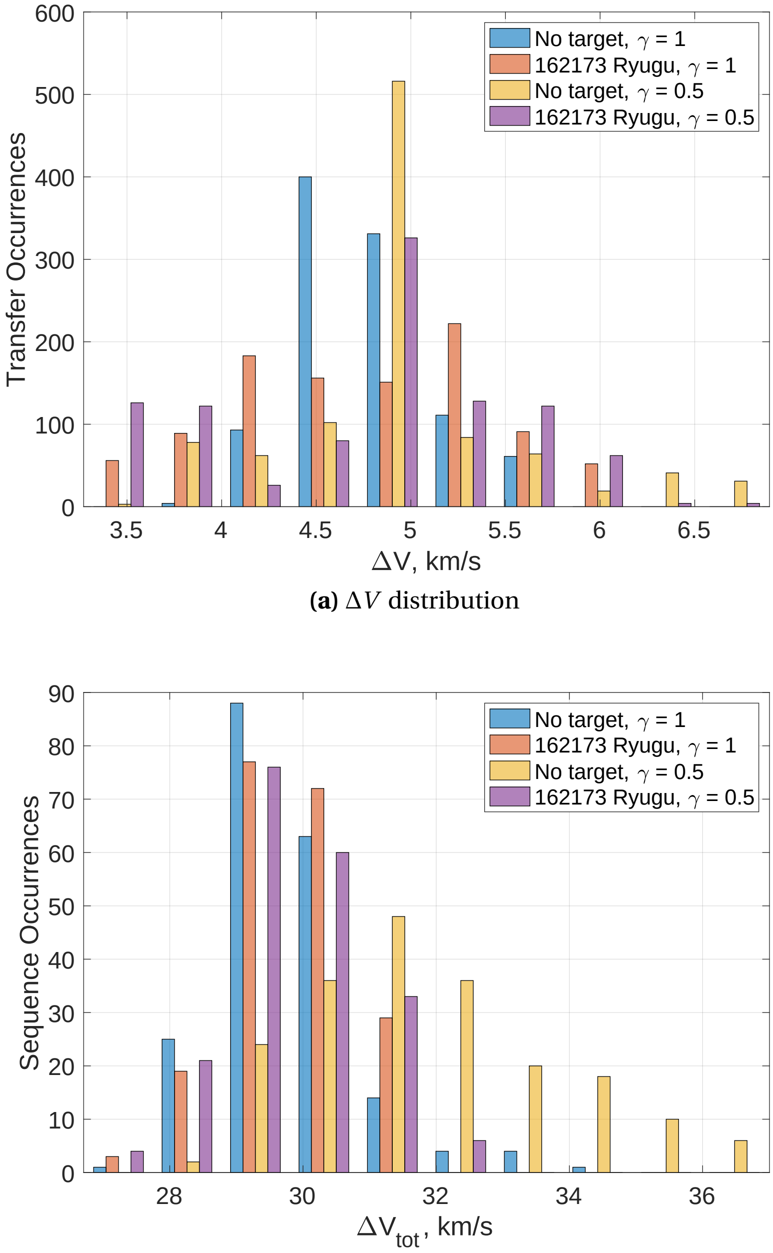
where $\gamma \in [0, 1]$ is a weight coefficient representing the relative importance given to ΔV and I_V when selecting the sequences. Note that $\Delta \hat{V}_{tot}$ and $\hat{I}_V$ refer to the normalised values (with respect to their maximum values) of the total ΔV and I_V of the sequence. The sequence search algorithms are modified so that the $N_S = 200$ best trajectories characterised by the lowest A (i.e. the most appealing sequences) are selected, i.e. a compromise between minimum ΔV and maximum interest value.

Varying the value of the weight γ and/or targeting one asteroid of interest within a sequence have an impact on the final mass expenditure and interest value of the sequence. To verify this, the SS-ANN algorithm is run for the following cases:

1. no target asteroid and $\gamma = 1$
2. 162173 Ryugu as the target asteroid and $\gamma = 1$
3. no target asteroids and $\gamma = 0.5$
4. 162173 Ryugu as the target asteroid and $\gamma = 0.5$

The sequences which are obtained for each of these cases are analysed. First, the ΔV for every transfers which are selected by SS-ANN at every iteration are investigated, of which a distribution is presented in Figure 7.8(a). Second, the ΔV_{tot} of the whole sequence for all the computed 200 sequences is studied and its distribution is shown in Figure 7.8(b). Similarly, an analysis of the interest of each NEA (i.e. their absolute magnitude, in this case) and of the interest value of the NEA sequences selected by SS-ANN in the cases 1 to 4 is performed. Figure 7.9(a) describes the distribution of the absolute magnitude, H, of the encountered NEAs in all the 200 sequences, while Figure 7.9(b) represents the distribution of the interest value, I_V, for all the sequences calculated by SS-ANN in the cases 1 to 4.

The figures indicate that when γ is lower than one, i.e. the selection of the sequences is made on the basis of both their I_V and ΔV, the selected sequences are characterised by a greater I_V and generally higher ΔV with respect to the case when only the ΔV is considered for the NEA selection (i.e. $\gamma = 1$). For $\gamma = 1$ (cases 1 and 2), it

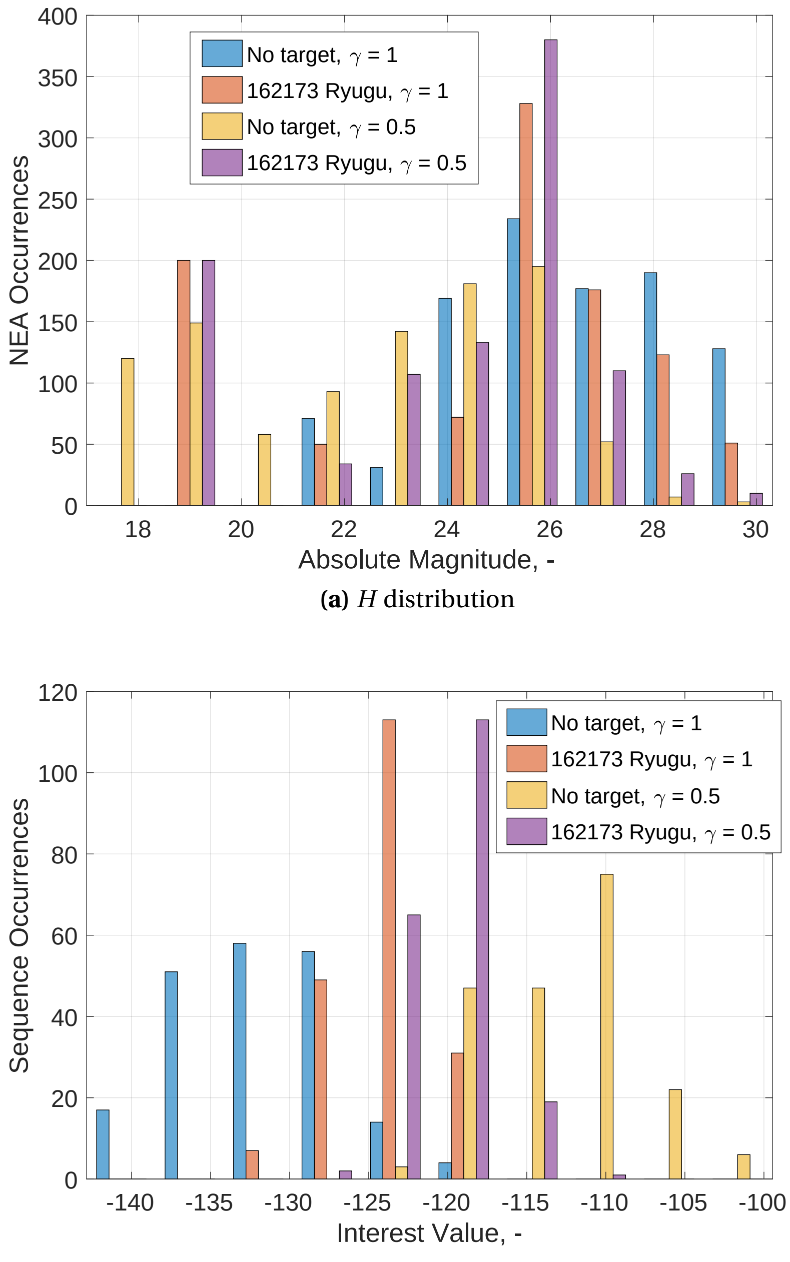


(a) ΔV distribution

(b) ΔV_{tot} distribution

Figure 7.8: Distribution of ΔV (a) and ΔV_{tot} (b) for the transfers and sequences calculated by SS-ANN for different γ values.

is possible to achieve a larger number of sequences with a greater I_V if an *interesting* asteroid is targeted within the sequences (case 2). However, it should be noted that, when $\gamma < 1$ (cases 3 and 4), the pruning of the asteroids, which is performed to ensure that the target asteroid can be conveniently reached, appears to reduce the likelihood



(a) H distribution

(b) I_V distribution

Figure 7.9: Distribution of H (a) and interest value I_V (b) for the NEAs and sequences calculated by SS-ANN for different γ values. Lower values of H indicate NEAs bigger in size, higher values of I_V (in magnitude and size) indicate higher interest.

to encounter asteroids with a larger size, compared to the case when no objects are targeted (case 3). In summary, more interesting objects can be favoured during the sequence selection process, increasing the overall appeal of the sequences generated

Table 7.1: Characteristics of the NEAs visited in Sequence A.

Sequence A	2008 EA9	2010 AN61	2018 CQ3	2004 FM32	2015 VO142
a, AU	1.05	1.16	1.17	1.10	1.08
e, -	0.07	0.13	0.15	0.16	0.13
i, deg	0.44	3.61	4.00	3.76	0.28
H, -	27.7	27.0	25.2	27.1	28.9
Estimated diameter, m	10	14	31	13	6
PHA	No	No	No	No	No
NHATS	Yes	Yes	Yes	Yes	Yes
Orbit Class	Apollo	Apollo	Apollo	Apollo	Apollo

by the SS-ANN algorithm at the cost of a larger ΔV.

7.4 Multiple NEA Sample Return Missions

To verify the outcome obtained from running the SS-ANN platform for MNR missions with sample return to Earth, three sequences are selected and fully optimised for an electric propulsion system with maximum thrust T_{max} = 0.3 N, specific impulse I_{sp} = 3000 s, and for an initial mass m_0 = 1500 kg. The sequences are selected from the following simulations:

1. Sequence A, chosen as the sequence with lowest ΔV obtained from case (1) (sequence search with no target asteroid and $\gamma = 1$)
2. Sequence B, chosen as the sequence with lowest ΔV obtained from case (2) (sequence search with 162173 Ryugu as the target asteroid and $\gamma = 1$)
3. Sequence C, chosen as the sequence with greater I_V obtained from case (3) (sequence search with no the target asteroid and $\gamma = 0.5$)

The orbital parameters of the asteroids visited during the sequences A, B and C are detailed in Tables 7.1, 7.2 and 7.3, respectively. The sequences visit five asteroids, of which some are PHA and all are NHATS except for 2000 LY27, which is a PHA and presents the lowest absolute magnitude H. The estimated size of the asteroids is calculated using Eq.(7.7) and considering an average albedo of 0.15 [228].

These sequences are further investigated by solving the OCP for the each leg of the mission. The objective of the optimisation algorithm is to find the optimal control vector that minimises the total mass expenditure while fulfilling the dynamics constraints of Eq.(3.17) at any time. The optimisation process presented in Section 6.4 is employed, where the OCP is solved leg by leg sequentially, where the values estimated by the ANN are used as the initial guess. Similarly, GPOPS-II is used as the optimiser.

Table 7.2: Characteristics of the NEAs visited in Sequence B.

Sequence B	2016 TB18	1998 KG3	162173 Ryugu	2014 UY	2008 EA9
a, AU	1.08	1.16	1.19	1.17	1.05
e, -	0.08	0.12	0.19	0.17	0.07
i, deg	1.53	5.51	5.88	3.56	0.44
H, -	24.8	22.1	19.3	25.4	27.7
Estimated diameter, m	38	131	474	29	10
PHA	No	No	Yes	No	No
NHATS	Yes	Yes	Yes	Yes	Yes
Orbit Class	Apollo	Amor	Apollo	Apollo	Apollo

Table 7.3: Characteristics of the NEAs visited in Sequence C.

Sequence C	1998 KG3	162173 Ryugu	2017 UY4	2000 LY27	2001 QC34
a, AU	1.16	1.19	1.19	1.31	1.13
e, -	0.12	0.19	0.16	0.21	0.19
i, deg	5.51	5.88	3.78	9.02	6.24
H, -	22.1	19.3	25.2	17.0	20.1
Estimated diameter, m	131	474	31	1367	328
PHA	No	Yes	No	Yes	Yes
NHATS	Yes	Yes	Yes	No	Yes
Orbit Class	Amor	Apollo	Apollo	Amor	Apollo

The initial position of the spacecraft is set equal to the position of the departure body at the given departure date. In the first leg, Earth is the departure body and the departure date is the launch date, which can be selected by the mission designer, and used in the SS-ANN algorithm. Although the launch date is fixed for this case study (2035-08-24), it is worth noting that a systematic search could be performed over a launch window. In the following legs, the last-visited asteroid becomes the departure body and the departure date can be optimised, given by the sum of the fixed launch date and the transfer time of the previous legs. To allow enough time for close-up observations and/or sample collection a minimum stay time of 20 days is enforced.

Tables 7.4, 7.5 and 7.6 describe the optimised mission characteristics of the sequences A, B and C, respectively. The departure and arrival dates, TOF, ΔV and stay time are specified for each transfer. To compare the optimised values with those estimated by the ANN, the latter are expressed within brackets in the tables. The optimisation procedure could find a solution for each of the transfers involved, showing that the trajectories calculated by the SS-ANN platform are feasible and the spacecraft can return to Earth after visiting five asteroids within ten years from departure. The heliocentric ecliptic-plane view of the optimised trajectories for the sequences A, B and C are shown in Figures 7.10, 7.11 and 7.12.

Table 7.4: Mission parameters of the optimised NEA Sequence A. Comparison of optimal results with ANN estimations (in brackets).

Leg	Departure	Arrival	TOF, days	ΔV, km/s	Stay time, days
Earth - 2008 EA9	2035-08-24	2037-04-20	605 (545)	5.96 (5.48)	83
2008 EA9 - 2010 AN61	2037-07-12	2038-11-29	505 (560)	4.80 (4.61)	20
2010 AN61 - 2018 CQ3	2038-12-19	2040-08-19	609 (550)	4.47 (4.43)	20
2018 CQ3 - 2004 FM32	2040-09-08	2042-06-08	638 (558)	4.75 (4.63)	100
2004 FM32 - 2015 VO142	2042-09-16	2043-10-05	384 (368)	3.10 (3.61)	54
2015 VO142 - Earth	2043-11-28	2045-03-31	489 (489)	4.01 (4.34)	–

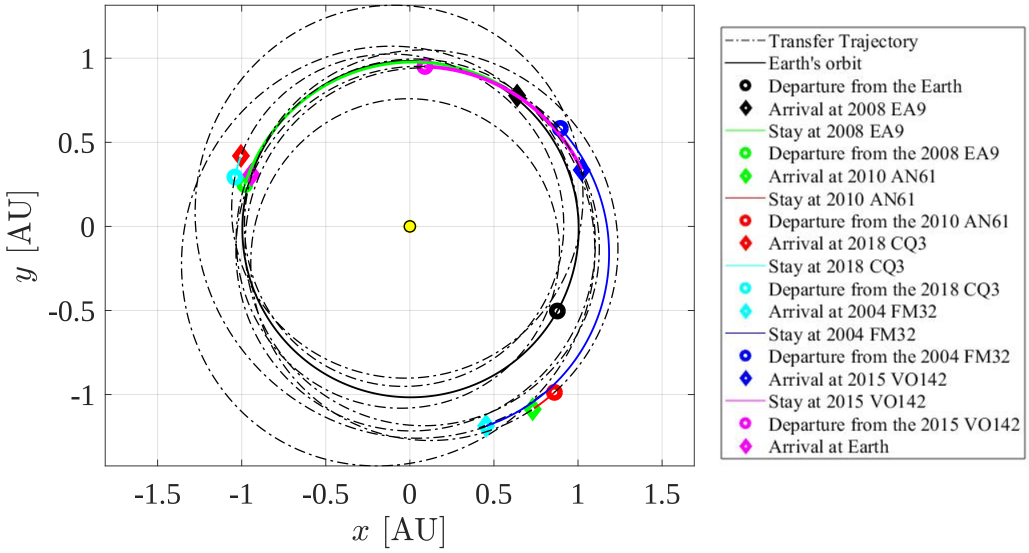


Figure 7.10: Sequence A (no asteroid targeting and $\gamma = 1$): heliocentric ecliptic-plane view.

Sequence A, obtained using $\gamma = 1$ (i.e. only the ΔV is considered for the NEA selection) and no asteroid targeted, allows to visit five NHATS objects with a required ΔV of 28.11 km/s. Sequence B, obtained using $\gamma = 1$ and targeting 162173 Ryugu, also allows for five NHATS objects, of which one PHA, to be rendezvous with a required ΔV of 27.35 km/s. Sequence C, obtained using $\gamma = 0.5$ (i.e. equal importance is given to the ΔV and I_V for the NEA selection) and no asteroid targeted, allows to rendezvous five NEAs, of which three are PHAs and 4 are NHATS, with a required ΔV of 35.75 km/s. The total interest value of the sequences A, B and C is equal to -135.9, -119.3 and -103.7, respectively.

As expected, for γ lower than 1, the ΔV to fly the NEA sequence increases, but it is

Table 7.5: Mission parameters of the optimised NEA Sequence B. Comparison of optimal results with ANN estimations (in brackets).

Leg	Departure	Arrival	TOF, days	ΔV, km/s	Stay time, days
Earth - 2016 TB18	2035-09-13	2037-02-15	521 (594)	2.51 (2.97)	47
2016 TB18 - 1998 KG3	2037-04-03	2038-11-10	586 (639)	9.89 (5.57)	43
1998 KG3 - 162173 Ryugu	2038-12-23	2039-12-13	355 (355)	4.30 (3.97)	65
162173 Ryugu - 2014 UY	2040-02-16	2040-10-09	236 (302)	2.49 (2.70)	28
2014 UY - 2008 EA9	2040-11-06	2042-04-02	512 (532)	4.94 (4.59)	117
2008 EA9 - Earth	2042-07-28	2043-08-14	382 (362)	2.61 (2.73)	–

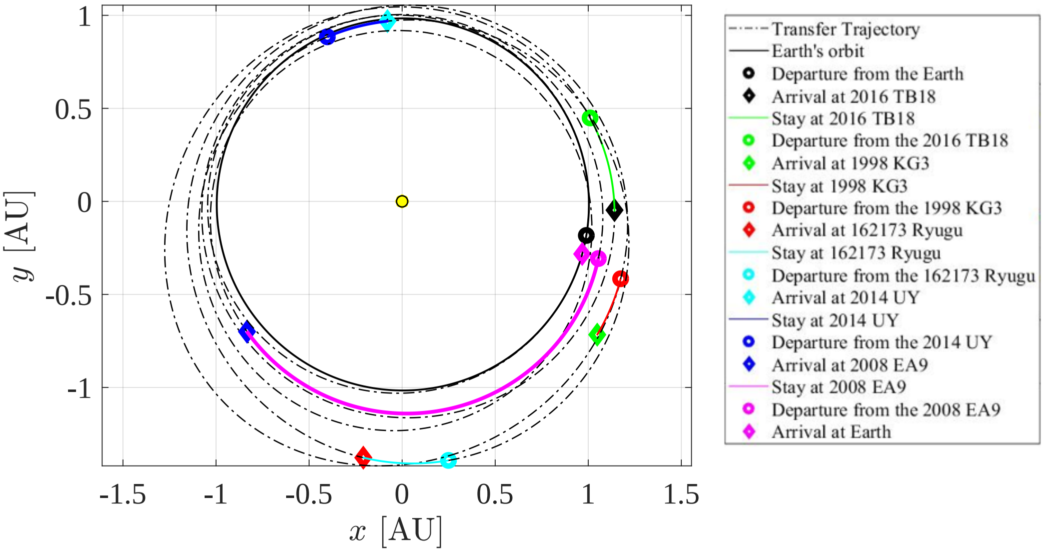


Figure 7.11: Sequence B (targeting 162173 Ryugu and $\gamma = 1$): heliocentric ecliptic-plane view.

possible to achieve a greater interest value of the mission. Also in the case of $\gamma = 1$, the interest value of the mission can be improved if an interesting asteroid is targeted specifically within the sequence. In the particular case of Sequence B, targeting a specific NEA within the sequence allows to reduce the total ΔV and TOF required compared to the case when no specific asteroid is targeted. This depends on the fact that, for asteroid targeting, the SS is designed to prune the asteroids selected also on the first portion of the mission, as shown in Figure 7.6. It should be highlighted that this is dependent also on the asteroid which is targeted.

The SS-ANN platform generates $N_S = 200$ asteroid sequences in about 7.6 hours and 3.3 hours for the cases in which no asteroid or an asteroid is targeted during the

Table 7.6: Mission parameters of the optimised NEA Sequence C. Comparison of optimal results with ANN estimations (in brackets).

Leg	Departure	Arrival	TOF, days	ΔV, km/s	Stay time, days
Earth - 1998 KG3	2035-09-11	2037-06-09	637 (655)	6.51 (6.81)	86
1998 KG3 - 162173 Ryugu	2037-09-03	2039-01-01	485 (405)	5.31 (4.94)	47
162173 Ryugu - 2017 UY4	2039-02-17	2040-07-24	523 (543)	5.79 (5.90)	98
2017 UY4 - 2000 LY27	2040-10-30	2042-07-23	631 (651)	5.68 (5.17)	120
2000 LY27 - 2001 QC34	2042-11-20	2044-09-12	662 (582)	7.29 (6.49)	109
2001 QC34 - Earth	2044-12-30	2046-01-20	386 (386)	5.87 (5.53)	–

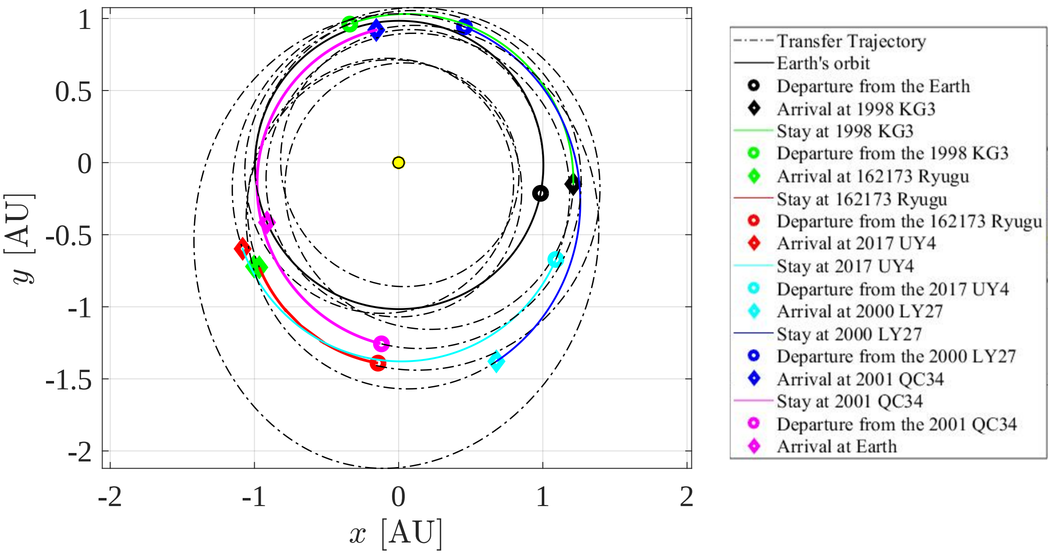


Figure 7.12: Sequence C (no asteroid targeting and $\gamma = 0.5$): heliocentric ecliptic-plane view.

search, respectively. The difference in computational time in the two cases depends on the fact that, when a specific asteroid is targeted, on that one leg, transfers to that asteroid only are considered and, on the successive leg, transfers from that asteroid only are considered. This reduces the computational time considerably, since not as many permutations need to be calculated. The simulations have been performed on a machine with an Intel Core i7 processor at 3.4 GHz. This reinforces the analysis in Section 6.4.4, where the required computational time when an ANN is used in the sequence search has been compared with the time required by other previously used methodologies [2]. In fact, the latter can require tens of days to fully compute the multiple NEA sequences when performing a similar simulation. It was demonstrated

Table 7.7: Total ΔV and I_V of the optimised sequences and the average percentage errors $\mathscr{E}$ between ANN output and the optimised results.

Sequence	ΔV_{tot}	I_V	$\mathscr{E}_{TOF}$	$\mathscr{E}_{\Delta V}$
A (Table 7.4)	28.11	-135.9	7.87%	6.69%
B (Table 7.5)	27.35	-119.3	8.28%	8.59%
C (Table 7.6)	35.75	-103.7	6.40%	6.54%
Mean Error	–	–	7.52%	7.27%

that, when an ANN is used, the algorithm results to be two orders of magnitude faster.

To evaluate how well the network performs with respect to the optimisation procedure, the deviations of the TOF and ΔV between the values estimated by the ANN and the optimal ones are calculated as the average percentage errors $\mathscr{E}$, whose formulation was provided in Eq.(4.2). Table 7.7 presents the percentage errors of the three sequences, together with a summary of the total ΔV and I_V for each of them. It can be noted again that, to fly NEA rendezvous missions with greater I_V, thus visiting asteroids of larger sizes (which are generally rarer), a larger ΔV is generally required which is however still achievable with the chosen propulsion system. The mean error of the sequences A, B and C is 7.52% for the TOF estimation and 7.27% for the ΔV estimation.

It can be concluded that the trained network is able to estimate with a reasonable accuracy the cost and duration of low-thrust transfers from the Earth, between NEAs and back to the Earth, while greatly reducing the computational time. Finally, the SS-ANN platform can compute the most convenient sequences of NEAs depending on the objective of the mission, which can be the minimisation of the TOF, the minimisation of the ΔV (i.e. propellant mass) and/or the maximisation of the interest value of the objects selected in the sequence.

8

Multi-Objective Optimisation of Low-Thrust Systems

This chapter analyses how different low-thrust propulsion capabilities can influence the outcome of multi-target missions. A methodology is developed to ultimately optimise the propulsion system so that the objectives of a mission can be achieved.

This chapter is organised as follows. The problem is formulated in Section 8.1, followed by a description of the methodology, which is proposed to address it and which uses the multi-objective GA and the ANN (GA-ANN), as described in Section 8.2. Section 8.3 presents the range of thruster capabilities which are considered for this study. The ANN implementation is given in Section 8.4, where the generation of the training database is discussed and the performance of the network is analysed. In Section 8.5, optimal MADR and MNR sequences are investigated and the performance of the proposed methodology is assessed. The candidate sequences which minimise the objective functions according to the mission goals, when flown using the selected propulsion capabilities, can be further refined through optimal control problem solvers.

8.1 Problem Formulation

The choice of the on-board propulsion system to perform high-ΔV missions, such as multiple-target missions, has a major impact on the cost (e.g. in terms of propellant mass required or total initial mass required, which would consequently affect the launch cost) and duration of the mission. As mentioned, low-thrust systems can be beneficial for this type of missions because they allow to deliver the same ΔV using less propellant than high-thrust chemical systems [25]. However, these LT systems can be characterised by various propulsion capabilities depending on their specific impulse I_{sp} and maximum thrust T_{max}. An analysis of how different values of I_{sp}

and T_{max} can affect multi-target missions, such as MADR and MNR missions, is performed. The GA-ANN methodology is proposed where a multi-objective GA and an ANN are integrated to identify the optimal propulsion capabilities to achieve the objectives of a multi-target mission.

In this study, two objectives are considered: the total duration of the mission and the initial mass of the system, given a fixed payload mass. It should be highlighted that a propulsion system which minimises the duration of the mission does not necessarily optimise the initial spacecraft mass. In fact, no unique solution exists that simultaneously minimises both objectives. For example, a mission could be flown with a lower initial mass but requiring a higher TOF, which translates into the mission being completed at a later stage. This is because a lower initial mass means that a smaller propellant mass is available and a smaller propulsion system is used, which is generally associated with lower maximum thrust available. The opposite case can also occur.

It follows that these objectives are conflicting. Thus, this problem is a multi-objective non-linear optimisation problem, for which a set of mathematically equally-good solutions can be identified. These solutions are known as Pareto-optimal solutions [183, 186], as discussed in Section 3.2.3. A multi-objective optimisation solver using a GA is employed to optimise the characteristics of the low-thrust propulsion system for a multi-target mission. In particular, the properties of an EP system, the specific impulse I_{sp} and maximum thrust T_{max}, are optimised so that the initial mass of the spacecraft and the total duration of the mission can be minimised.

An ANN is trained and integrated within the multi-objective GA to provide an estimate of the total duration and initial mass required to fly the mission. In the analysis of multi-target missions presented so far in this thesis, the ANN is trained with a fixed low-thrust propulsion system, which is selected a priori. Although this allows to obtain accurate results specific to the selected propulsion system, it requires the ANN to be re-trained every time that the use of thrusters with other characteristics needs to be explored. In this chapter, the possibility to include the propulsion system characteristics as inputs to the network to quickly estimate the cost of a trajectory, for different low-thrust systems, is investigated. The network output is used by the multi-objective GA to generate the Pareto front of optimal solutions, as detailed in the next section. The proposed GA-ANN method offers space mission designers the possibility to perform trade studies and choose the most appropriate propulsion system based on the given mission requirements and objectives.

8.2 Multi-Objective GA and ANN

Current EP thrusters offer a range of characteristics and capabilities to propel at different values of specific impulse, I_{sp}, and maximum thrust, T_{max} [98]. Multi-target

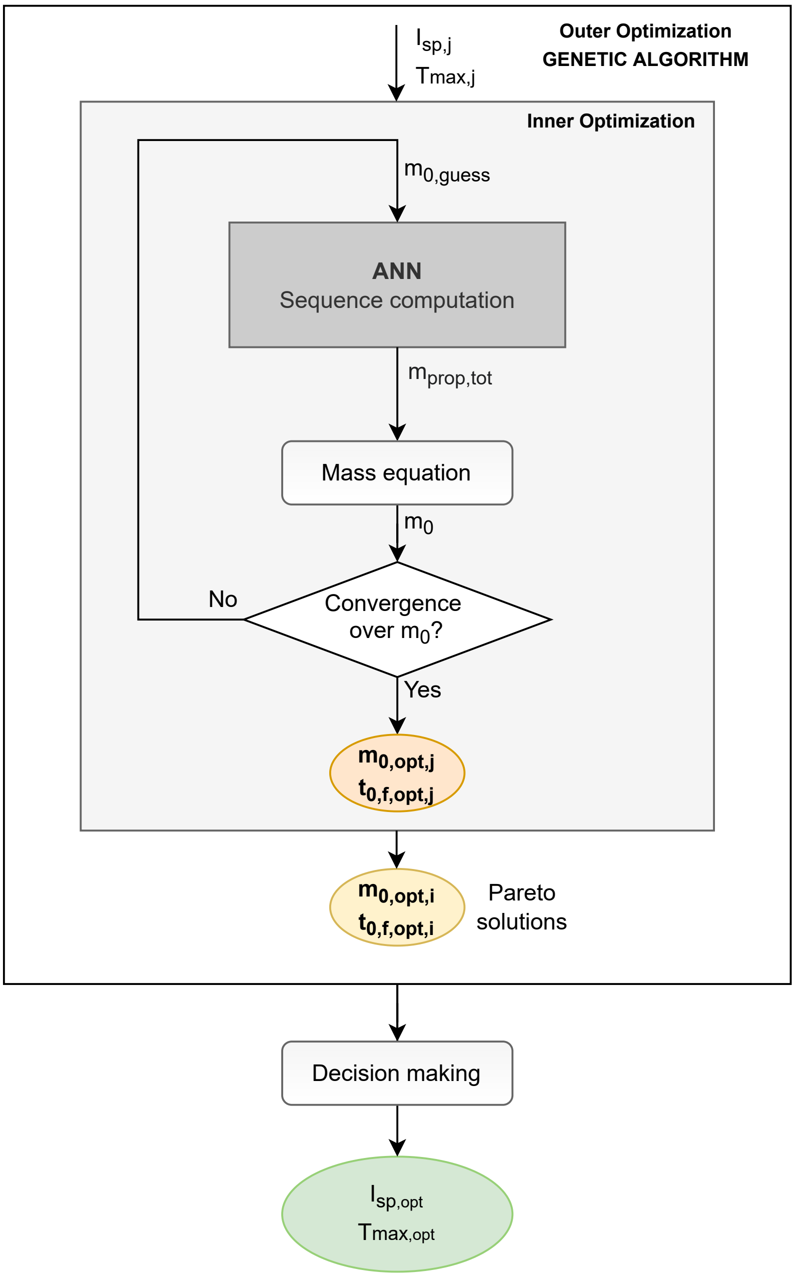


Figure 8.1: Diagram of the optimisation logic.

missions, which are highly demanding in terms of energy, require to select the most efficient propulsion system for the mission. An accurate selection of the propulsion system can lead to missions with shorter time of flight and/or less propellant mass required.

In an effort to study how the choice of the propulsion system can affect the performance of multi-target missions and ultimately to identify the optimal propulsion system for any given mission, the optimisation logic which is schematically presented in Figure 8.1 has been developed. Two nested optimisations are performed: (i) the inner optimisation to find the minimum initial mass m_0 (for a fixed payload mass) and (ii) the outer optimisation over I_{sp} and T_{max} of the EP system, to minimise the two objective functions, i.e. the initial mass, m_0, and the TOF, $t_{0,f}$.

The outer optimisation is solved by a multi-objective GA. The objective is to analyse how the selection of the propulsion capabilities affects the mission in terms of cost (i.e. required initial mass m_0 which, as it will be specified below, depends on the propellant mass and EP system mass) and duration. As both m_0 and TOF are objective functions of the GA, a multi-objective optimisation (also known as Pareto optimisation) is conducted. The multi-objective optimisation entails the process of multiple-criteria decision making, which is applied in cases where optimal decisions need to be taken as a trade-off between two or more conflicting objectives [186]. More details on the multi-objective optimisation are provided in Section 3.2.3.

The multi-objective GA can identify a number of Pareto-optimal solutions, all of which are considered equally good. These optimal solutions are obtained for different values of I_{sp} and T_{max} and the selection of one solution over the others depends on the goals and/or the requirements of the mission.

For each combination of $I_{sp,j}$ and $T_{max,j}$, with $j = 1,...,N_p$ and N_p given by the product of the number of generations times the population size in a generation, the evaluation of the objective functions requires the multi-objective GA to run the inner optimisation, and estimate the optimal $m_{0,opt,j}$ and duration $t_{0,f,opt,j}$ which can be obtained with the given EP system. To this end, the ANN is trained and used to compute the cost of low-thrust transfers, given the orbital elements of the initial and final objects, the properties of the propulsion system (i.e. I_{sp} and T_{max}) and the mass of the spacecraft at the beginning of the transfer. In fact, to optimise the low-thrust propulsion system, I_{sp}, T_{max} and m_0 need to be included as input to the network because all the three parameters are subject to the optimisation and change at every iteration influencing the network output. The ANN input and output vectors are specified in Section 8.4. In particular, the ANN estimates the total propellant mass and TOF required to fly the full sequence of objects with the given propulsion system and initial mass.

The objective of the inner optimisation is to identify the minimum initial mass which allows to fly the given multi-target mission with the given propulsion system. Once the propellant mass required to fly the mission is calculated by the ANN, the total spacecraft mass m_0 can be calculated for a preliminary mass budget as [229]:

$$m_0 = m_{prop} + m_{tank} + n_{thrusters}(m_{EP} + m_{gimbal}) + m_{pl} \tag{8.1}$$

which is referred to in Figure 8.1 as *mass equation.* Each term in the equation can be described as follows:

- m_{prop} is the propellant mass necessary to fly the desired mission and is estimated by the ANN
- m_{tank} is the mass of the tanks and can be expressed as a function of the propellant mass:

  $$m_{tank} = \rho_{tank} m_{prop} \tag{8.2}$$

  with $\rho_{tank} = 0.1$ [230].
- $n_{thrusters}$ is the number of thrusters of the EP system.
- m_{EP} is the mass of a thruster, which is a function of its power and, at a first approximation, can be expressed as:

  $$m_{EP} = k_{EP} P_{EP,max} \tag{8.3}$$

  with k_{EP} being an empirical constant [90] and $P_{EP,max}$ being the maximum power required by the thruster, which is a function of the maximum thrust available and of the specific impulse:

  $$P_{EP,max} = \frac{T_{max} I_{sp} g_e}{2\eta_{EP}} \tag{8.4}$$

  where $\eta_{EP} = 0.7$ is the efficiency of the propulsion system to convert electrical energy [231]. In this work, off-the-shelf thrusters are considered, so m_{EP} is fixed to the mass of the thruster selected, as it will be specified in Section 8.3.
- m_{gimbal} is the mass of EP gimbal system which is included when more than one thruster are used, with the purpose of compensating the misalignment of the thrust vector produced by the thrusters with respect to the centre of mass of the spacecraft. The gimbal mass needs to be proportional to the mass of the thruster:

  $$m_{gimbal} = \rho_{gimbal} m_{EP} \tag{8.5}$$

  with $\rho_{gimbal} = 0.1$ [229].
- m_{pl} is the payload mass, which in this case includes the structure and the masses of all the other required systems a part from the propulsion system and the propellant mass.

Substituting the above expressions, Eq.(8.1) can be written as follows:

$$m_0 = (1+\rho_{tank}) m_{prop} + n_{thrusters}(1+\rho_{gimbal}) m_{EP} + m_{pl} \tag{8.6}$$

where it can be concluded that minimising the initial mass m_0 is equivalent to minimising the propellant mass m_{prop} and the number of thrusters $n_{thrusters}$, given that m_{EP} and m_{pl} are fixed.

The propellant mass, which is calculated by the ANN, is a function of the initial mass m_0, the ΔV required to perform the transfer, and the specific impulse of the propulsion system I_{sp}, as defined by the rocket equation [24]:

$$m_{prop} = m_0 \left(1 - e^{-\frac{\Delta V}{I_{sp} g_e}}\right) \tag{8.7}$$

It is worth noting that it is chosen to have the initial mass m_0 as one of the objective functions of the optimisation, so that both the propellant mass and total EP system mass (given by the number of thrusters multiplied by m_{EP}) can be considered. This is done because, for example, a mission could be flown with an EP system with certain values of I_{sp} and T_{max} which allow to reduce the propellant mass, at cost of a higher number of thrusters being required (i.e. greater total mass of the EP system), which may have a negative impact on the initial mass m_0 overall.

In the inner optimisation procedure proposed, also illustrated in Figure 8.1, a guessed initial mass $m_{0,guess}$ is used as input to the ANN, which calculates the total cost and duration of the given multi-target mission. From the ANN output, which provides an estimation of m_{prop}, and from the values of $I_{sp,j}$ and $T_{max,j}$, which allow to identify the number of thrusters needed, the initial mass m_0 can be recalculated using Eq.(8.6). The resulting value of m_0 is compared to the guessed initial mass $m_{0,guess}$ and, if the absolute value of the difference $|m_0 - m_{0,guess}| \leq \epsilon$ with $\epsilon = 10^{-5}$, the convergence is reached and the minimum initial mass $m_{0,opt,j}$ to fly the desired mission is identified. Otherwise, the process is repeated using m_0 as the updated guessed initial mass, i.e. $m_{0,guess} = m_0$. This procedure is performed iteratively, until convergence. Once the inner optimisation converges, the TOF is calculated by the ANN and set as the optimal TOF, $t_{0,f,opt,j}$, which is required to fly the mission with this minimum initial mass $m_{0,opt,j}$ and the given propulsion system ($I_{sp,j}$ and $T_{max,j}$).

In conclusion, the multi-objective optimisation process, GA-ANN, where the multi-objective GA and the trained ANN are used in the outer and inner optimisations, computes the Pareto optimal solutions which are the output solution vectors of the objective functions $\mathbf{y}_{GA,opt,i}$ at the values $\mathbf{x}_{GA,opt,i}$ of the input variables of the multi-objective GA, with:

$$\mathbf{x}_{GA,opt,i} = [I_{sp,i}, T_{max,i}] \tag{8.8}$$

$$\mathbf{y}_{GA,opt,i} = [m_{0,opt,i}, t_{0,f,opt,i}] \tag{8.9}$$

where i indicates the i-th Pareto solution with $i = (1, ..., N_{GA})$ and N_{GA} being the total number of Pareto solutions identified by the GA. A final optimal solution with $\mathbf{x}_{GA-ANN} = [I_{sp,opt}, T_{max,opt}]$ and $\mathbf{y}_{GA-ANN} = [m_{0,opt}, t_{0,f,opt}]$ can be selected amongst the Pareto-optimal solutions following a decision making process, so that the goals and requirements of the mission can be met.

8.3 Low-Thrust Propulsion Systems

In this work, EP systems for small satellites are considered, but the same analysis and considerations could be expanded to larger satellites or different types of propulsion systems. Small satellites are selected to conduct this study as they represent an attractive solution for multi-target missions. They can be propelled by smaller thrusters, reducing the total ΔV of the mission. Also, smaller satellites allow to reduce the launch cost, since they can be launched as piggyback or from a smaller and cheaper launch vehicle [100, 101].

One crucial challenge to enhance the flexibility and allow multi-target missions with such small spacecraft is to develop a small, lightweight, compact and efficient EP system. The study in Ref. [89] concluded that an efficient EP propulsion system enables the use of a smaller, less expensive launch vehicle, and significantly shortens the overall trip time. Some of the missions flown using small satellites are presented in Section 2.3.

As mentioned, off-the-shelf thrusters are used in this application. Many thrusters are available on the market, with different characteristics. The models of the thrusters by Enpulsion® are selected as they offer a wide range of maximum thrust and specific impulse.

For the MADR missions, the Micro R3 thruster (which was shown in Figure 2.6a) is selected, with the operational envelope being presented in Figure 8.2. The operational envelope is obtained by approximating the ranges of maximum thrust which can be obtained for certain values of the specific impulse, according to the capability of the thruster, as detailed in the product specifications [99]. For a fixed I_{sp}, any increase in T_{max} results in a higher operational power, as expected from Eq.(8.4) (which might result in a higher mass of the external power systems, although in this analysis this is considered fixed).

It should be noted that the values of I_{sp} and T_{max} which are considered as input to the multi-objective GA are values corresponding to an operational point inside the operational envelope of the EP system. Also, although the Micro R3 thruster is offered with a fixed amount of propellant mass, this is neglected at this stage since the minimum amount of propellant mass to fly the mission is calculated by the ANN and is subject to the optimisation as part of the total initial mass m_0.

The Micro R3 thruster is a scaled technology of the Enpulsion Nano thruster [98]. Similarly, as the MNR missions generally require greater propulsion capabilities than the MADR missions, the capabilities of the Micro R3 thrusters are scaled up to obtain a thruster (referred in this paper as Mini thruster) which can reach greater T_{max} values than the Micro R3 thruster. This is achieved, to a first approximation, by simply scaling up the maximum T_{max} achievable and the mass of a Micro R3 thruster by a factor of 4, resulting in an operational envelope of the Mini thruster which is shown

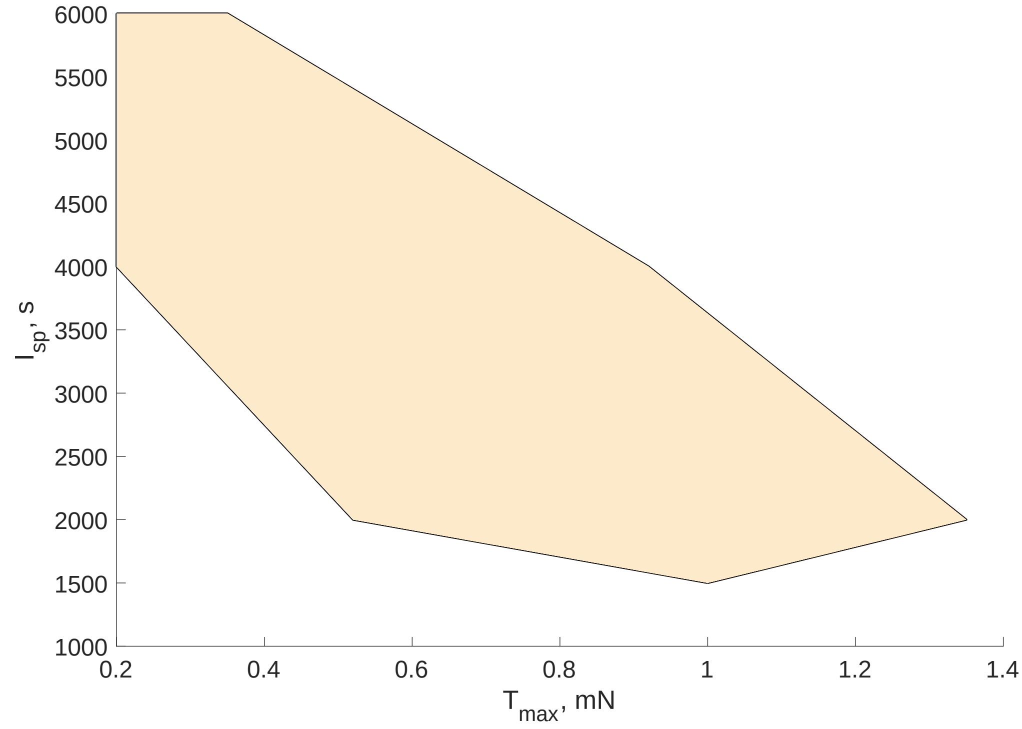


Figure 8.2: Propulsion capabilities of the Micro R3 thrusters.

in Figure 8.3.

The Enpulsion® off-the-shelf thrusters are engineered in a modular approach, with units clustering easily together to form building blocks that can be arranged for various mission profiles. The mass of one thruster m_{EP} is 2.6 kg and 9.8 kg for the Micro R3 and Mini thruster, respectively. As the mass m_{EP} of a single thruster mass is fixed in Eq.(8.1), the parameter that varies in the multi-objective optimisation is, apart from the propellant mass, the number of thrusters.

The benefit of employing more than one thruster is that the thruster operational envelope (Figure 8.2 for Micro R3 thruster and Figure 8.3 for the Mini thruster) stretches towards greater maximum values of T_{max}. For this study, the maximum number of thrusters allowed is 15.

8.4 ANN for Propulsion System Selection

An ANN is trained so that, given the orbital elements of the departure body B_1 and arrival body B_2, I_{sp} and T_{max} of the EP system and initial mass m_0 (inputs), it can provide a quick estimation of the propellant mass and TOF (outputs) required

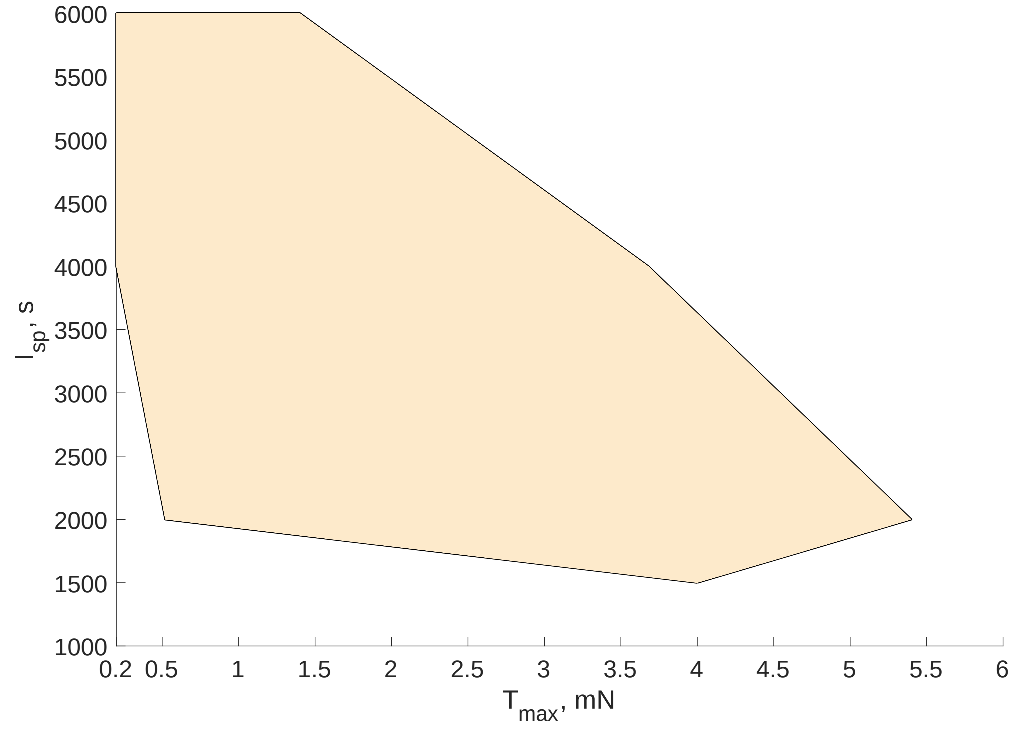


Figure 8.3: Propulsion capabilities of the Mini thrusters.

to transfer from B_1 to B_2. In particular, considering the differences in the dynamics and assumptions between the MADR and MNR missions, as detailed in the dedicated Chapters 5 and 6, the input vector $\mathbf{x}$ can be defined:

- for MADR:

$$\mathbf{x} = [h_{D1}, \Omega_{D1}, m_{D1}, h_{D2}, \Omega_{D2}, m_{SC}, I_{sp}, T_{max}] \tag{8.10}$$

  where h and Ω are the altitude and RAAN of the departure ($D1$) and arrival ($D2$) debris objects; m_{D1} is the mass of the departure debris, which needs to be carried to disposal; m_{SC} is the mass of the chaser before the transfer (with $m_{SC} = m_0$ on the first leg); and I_{sp} and T_{max} describe the properties of the propulsion system.

- for MNR:

$$\mathbf{x} = [\mathbf{x}_{MEE,1}, \mathbf{x}_{MEE,2}, m_{SC}, I_{sp}, T_{max}] \tag{8.11}$$

  where $\mathbf{x}_{MEE} = [p, f, g, h, k, L]$ are the modified equinoctial elements of the departure (subscript 1) and arrival (subscript 2) asteroid; m_{SC} is the spacecraft mass before the transfer (with $m_{SC} = m_0$ on the first leg); I_{sp} and T_{max} describe the properties of the propulsion system.

The ANN output vector can be defined as:

$$\mathbf{y} = [m_{prop}, t_{0,f}] \tag{8.12}$$

where the propellant mass m_{prop} and TOF $t_{0,f}$ required to transfer from the departure body B_1 to the arrival body B_2 are specified. It is chosen to use m_{prop} as an output of the network instead of ΔV, because the multi-objective optimisation is performed over the initial mass m_0 and the mission duration $t_{0,f}$. Since m_0 depends on m_{prop} and the number of thrusters, as explained in the previous section, it is coherent and appropriate to have the ANN to estimate the m_{prop} directly.

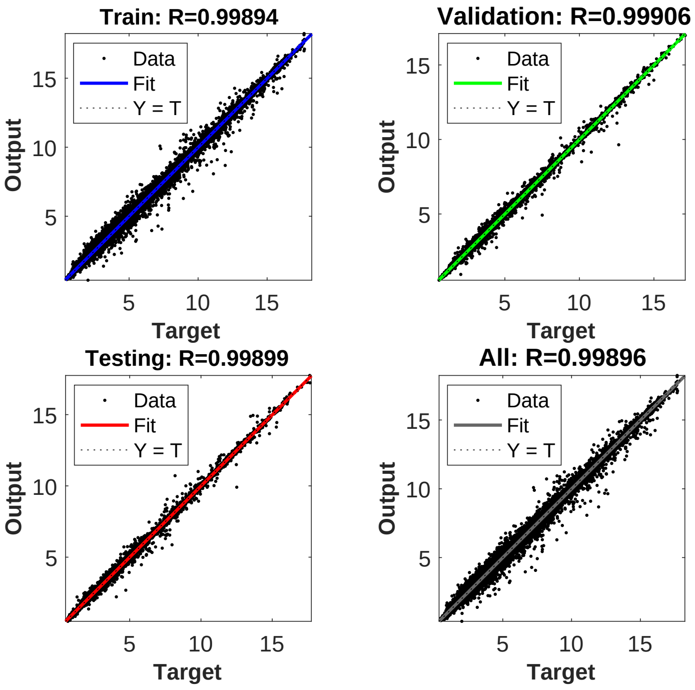
The training database contains the inputs and the desired outputs (targets), which are used during the training of the network. The network training database for the MADR case is generated by computing the low-thrust transfers for 150,000 pairs of fictitious debris objects. The data of objects used is the same employed in Chapter 5, including objects with altitudes between 500 and 1500 km and mass between 100 and 300 kg. The inclination is fixed to 87.9 deg and the disposal orbit is at an altitude of 390 km. However, since in this case small satellites are adopted, the chaser mass can vary from 40 kg to 140 kg. The number of 150,000 samples to train the ANN is the optimal number for this application, as discussed in Section 4.2.

Similarly, for the MNR case, 50,000 pairs of NEAs [214] are selected to define the training database for the ANN. As explained in Section 6.2, asteroids with eccentricity below 0.4 and inclination below 20 deg are used, as the highly inclined and highly eccentric objects would require a larger cost and duration to transfer to and from and consequently would be discarded by the sequence search. The mass of the spacecraft can vary from 40 kg to 220 kg. In the MNR case, a larger range is allowed for the initial mass because a higher thrust (thus, a heavier propulsion system) and a greater propellant mass is generally required compared to the MADR missions.

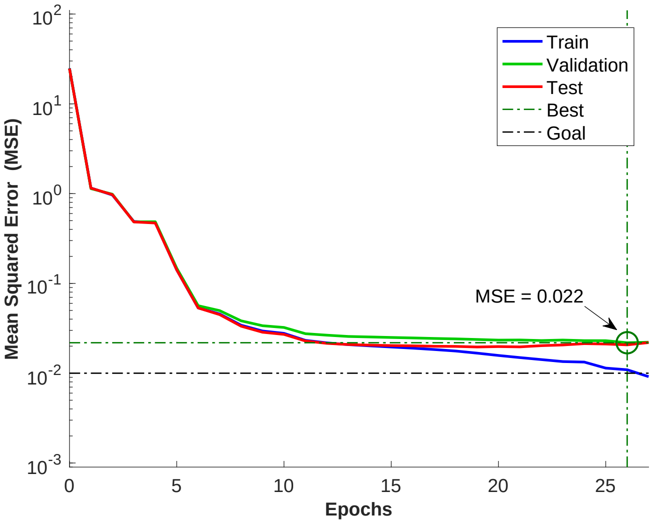
Multi-layered feedforward ANNs are selected also for this case study, with the same architecture and hyper-parameters which are presented in Table 4.5 and are demonstrated to perform optimally for the application of multi-target missions. Two networks (one for the MADR case and one for the MNR case) with four hidden layers and 80 neurons using the sigmoid as activation function are built. The training databases are divided into 70% training set, 15% validation set and 15% test set, to detect and prevent overfitting. The training algorithm used is the Levenberg-Marquardt algorithm, which allows to achieve a higher accuracy of the network output. In the next section, the performance of the networks is analysed for the case of MADR missions and MNR missions, respectively.

8.4.1 Performance Analysis

Figures 8.4 and 8.5 show the performance of the trained network for MADR and MNR, respectively. In both figures, the plots (a) present the regression analysis of



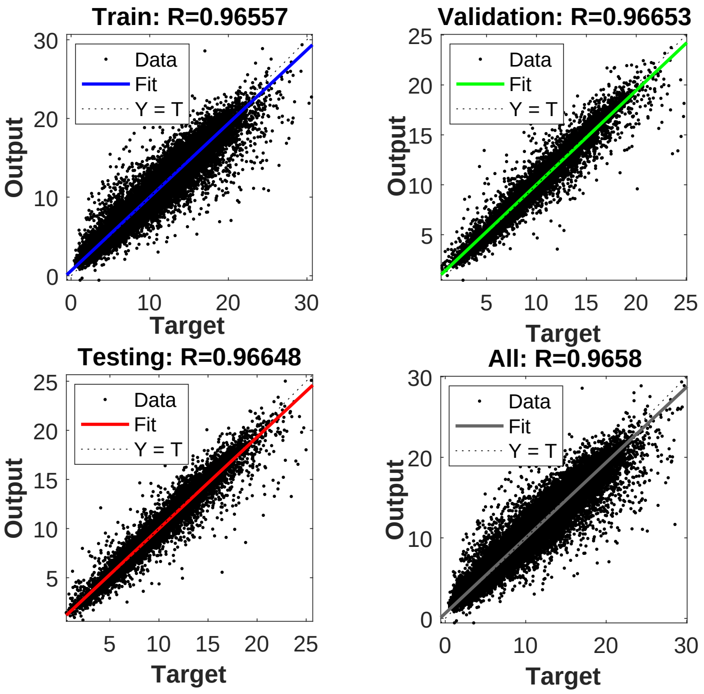
(a) Regression



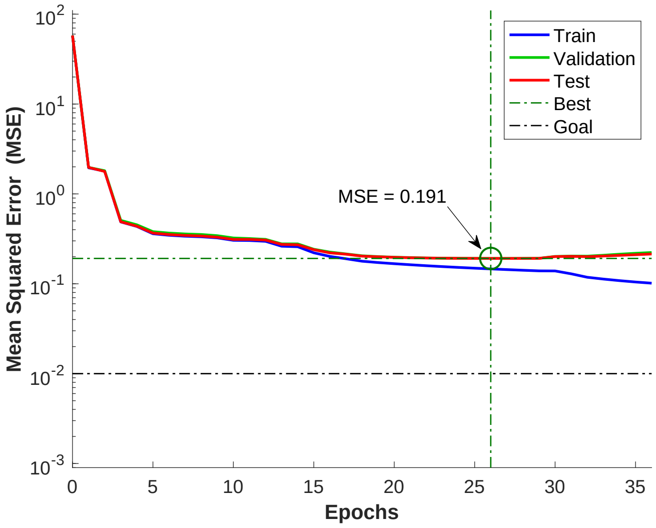
(b) Performance

Figure 8.4: Regression (a) and performance (b) analysis of the ANN trained for MADR missions.

the network outputs and the targets for the training set, validation set, test set and overall. The final overall correlation coefficient is 0.9956 for MADR and 0.9658 for MNR, which indicates that the ANN is able to approximate the fitness function which



(a) Regression



(b) Performance

Figure 8.5: Regression (a) and performance (b) analysis of the ANN trained for MNR missions.

relates inputs and outputs for both types of multi-target missions.

The performance plots (b) show how for each training epoch the MSE decreases until the performance goal is met (such as in the MADR case, Figure 8.4(b)) or before

Table 8.1: Summary of the network performance for each case study.

Case study	R	Validation MSE
MADR	0.9956	0.015
MNR	0.9732	0.121
MLT for MADR	0.9989	0.022
MLT for MNR	0.9658	0.191

the training starts overfitting (such as in the MNR case, Figure 8.5(b)). As mentioned, the latter case is verified when the validation-set MSE begins to increase while the training-set MSE continues to decrease. The final validation-set MSE of 0.022 for the MADR case and 0.191 for the MNR case suggest a very accurate performance of the trained networks.

A summary of the performance of the networks trained for all the case studies considered in this PhD work is given in Table 8.1. The ANN trained for the MADR missions achieves a better performance than the ANN for the MNR missions, as it can be noticed from both the correlation coefficient R and validation MSE. This can be linked to the increased complexity of the MNR model compared to the MADR model, which are detailed in Sections 6.1 and 5.1, respectively. In essence, in MNR missions the spacecraft can perform 3-D transfers (while in MADR missions, only transfers between circular orbits with no change in inclination are considered). Also, the GA is used in the shape-based method to find the MNR (sub-)optimal transfers to generate the training database. This difference in the network performance between MADR and MNR missions was already noticed for the case of a fixed propulsion system and fixed initial mass.

Comparing the network performance of the MLT cases to the one obtained for MADR (Chapter 5) and MNR (Chapters 6 and 7) missions with a fixed propulsion system and fixed initial mass, the network has three additional inputs, i.e. I_{sp}, T_{max} and m_0. This is to account for the various requirements on the propulsion system and initial mass, depending on the goals of the missions. Nevertheless, the network performance results to be very high, presenting only a slight reduction in accuracy compared to the cases where the EP system and initial mass of the spacecraft are fixed. This can be related to the increased complexity of the function which the network has to learn for the MLT analysis where the propulsion system capabilities and the initial mass can vary.

In Sections 5.3.1 for MADR and 6.4.4 for MNR, the impact of using ANNs on the computational time for the preliminary design of multi-target missions is demonstrated, compared to methods which are currently employed in the industry or in the literature. The benefit of ANNs in terms of computational time depends on the size of the problem, e.g. when hundreds of objects are considered (such as in the MADR analysis conducted) the algorithm speed is an order of magnitude faster, while when thousands of objects are involved in the computation (such as in the MNR analysis conducted) the algorithm is two orders of magnitude faster. This is expected as the number of calculations to be performed increases exponentially as the number of objects increases linearly.

8.5 Optimal Low-Thrust Systems for Multi-Target Missions

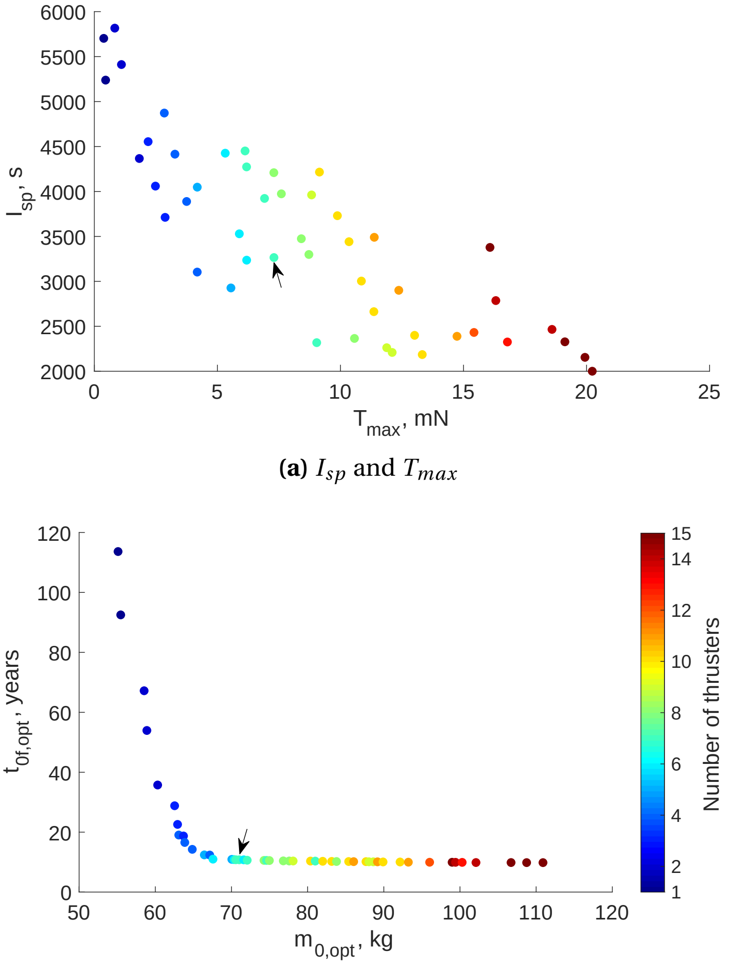
The methodology to find optimal MADR and MNR sequences is described in Chapters 5 and 6, respectively. These sequences were calculated using the SS-ANN platform, which is implemented to identify the most promising sequences of objects in terms of minimum TOF or m_{prop}. Two of the optimal sequences which were identified in those chapters and presented in Tables 8.2 and 8.3, are selected and fixed so that in this section the optimal low-thrust system to perform those missions can be found and, ultimately, the capabilities of the proposed methodology, GA-ANN, can be demonstrated.

8.5.1 Optimal Low-Thrust System for MADR Missions

The MADR sequence which is selected for this analysis is the one presented in Table 5.2, Sequence B. This mission allows for the disposal of 11 debris objects in 10.87 years with a required propellant mass of 60.76 kg, when the initial mass is 400 kg. In this section, the same MADR mission is optimised for a small satellite equipped with Micro R3 thrusters, presented in Section 8.3. In particular, the optimal values of the initial mass $m_{0,opt}$ and duration $t_{0,f,opt}$ for different properties (I_{sp} and T_{max}) of the EP system are computed and investigated, so that the optimal propulsion system for the goals and requirements of the given mission can be selected.

Figure 8.6 illustrates the EP properties and Pareto front obtained from running the GA-ANN algorithm for this MADR sequence. Plot (a) shows the values of I_{sp} and T_{max} associated to the optimal points in the Pareto plot (b), which presents the two objective functions, namely $m_{0,opt}$ on the X-axis and $t_{0,f,opt}$ on the Y-axis. The colours of the points in the plots depend on the number of the on-board thrusters required to achieve the desired T_{max}, going from 1 (dark blue) to 15 (dark red), as indicated in the colour bar in the figure.

The computational time required to run the GA-ANN method is 42.48 minutes, for a GA population size of 200. Considering that the GA is recomputing the sequence multiple times with different values and combinations of I_{sp} and T_{max}, it can be appreciated how the use of the ANN allows to considerably reduce the computational time which would be otherwise required to identify an accurate solution, as discussed



(a) I_{sp} and T_{max}

(b) Pareto front

Figure 8.6: EP characteristics, I_{sp} and T_{max} (a) and Pareto front (b) for the selected MADR mission.

in Section 8.4.1.

It can be noticed that, when the lowest values of T_{max} are used (i.e. minimum number of thrusters) only Pareto solutions with high values of I_{sp} are obtained. These solutions are characterised by a small $m_{0,opt}$, at the expenses of the TOF which becomes exponentially higher. Differently, lower values of I_{sp} and higher values of T_{max} (i.e. larger number of thrusters) allow for Pareto solutions with a shorter duration of the mission, at the expenses of a higher initial mass.

In this case, the point just after the elbow in the Pareto front (indicated by the arrow in the plots of Figure 8.6) is selected, where an initial mass of 72.04 kg (of which the propellant mass is 11.05 kg) and total TOF of 9.13 years are achieved when using seven Micro R3 thrusters which deliver a I_{sp} = 3265 s and T_{max} = 7.30 mN. This sequence is recalculated by integrating the dynamics of the system to obtain the full trajectory, as

Table 8.2: Characteristics of the selected MADR sequence flown by a microsat of 72 kg and equipped with 7 Micro R3 thrusters delivering I_{sp} = 3265 s and T_{max} = 7.30 mN.

Debris	h, km	RAAN, deg	m, kg	TOF, days	m_{prop}, kg
D1	661.73	162.03	131.24	N/A	N/A
D2	504.96	158.72	270.37	295	1.14
D3	577.40	149.33	163.74	254	0.95
D4	534.05	145.48	256.27	405	2.02
D5	517.24	145.16	253.91	195	0.99
D6	514.14	138.62	139.19	426	0.95
D7	534.61	128.19	213.67	282	0.67
D8	667.95	84.76	222.82	213	1.05
D9	694.88	66.66	200.81	313	1.74
D10	517.45	123.32	171.80	431	1.59
D11	588.01	88.94	213.15	178	0.76

explained in Section 5.4. The resulting values of mass and transfer duration can be compared with the ANN outputs to evaluate the ANN performance.

Table 8.2 presents the characteristics (altitude h, RAAN, and mass m) of the objects which are de-orbited as part of the optimised MADR sequence and the mission characteristics (TOF and m_{prop}). A fixed stay time, T_s = 30 days, at each object is considered to allow for phasing and docking. Additionally, Figure 8.7 illustrates the changes in altitude and used propellant mass of the spacecraft along the sequence. The total duration of the mission is 9.02 years and it requires a propellant mass of 11.86 kg in total. It follows that the GA-ANN methodology is able to identify the TOF and propellant mass of the system with an error of approximately 1.2% and 6.8%, respectively. Additionally, this also shows the advantages of using a micro-satellite to fly MADR missions. In fact, these results are compared with those presented in Table 5.2 for Sequence B, which were obtained for a chaser of 400 kg (of which the propellant mass is 60.76 kg) and equipped with an EP system delivering I_{sp} = 2000 s and T_{max} = 21 mN. It results that, when the optimal EP system is selected and for missions which allow for a smaller payload mass, the same sequence requires about 50 kg of propellant mass less when a micro-satellite is used, resulting in a initial mass at launch which is about 80% smaller, for a comparable duration of the mission.

8.5.2 Optimal Low-Thrust System for MNR Missions

To study the performance of GA-ANN for the MNR case, the sequence (Sequence C) presented in Tables 6.5 and 6.6 and Figure 6.11 is selected. This sequence allows to visit six asteroids in less than 10 years, requiring a total ΔV of 17.95 km/s for an initial launch mass of 1000 kg and an EP system with I_{sp} = 3000 s and T_{max} = 0.1 N.

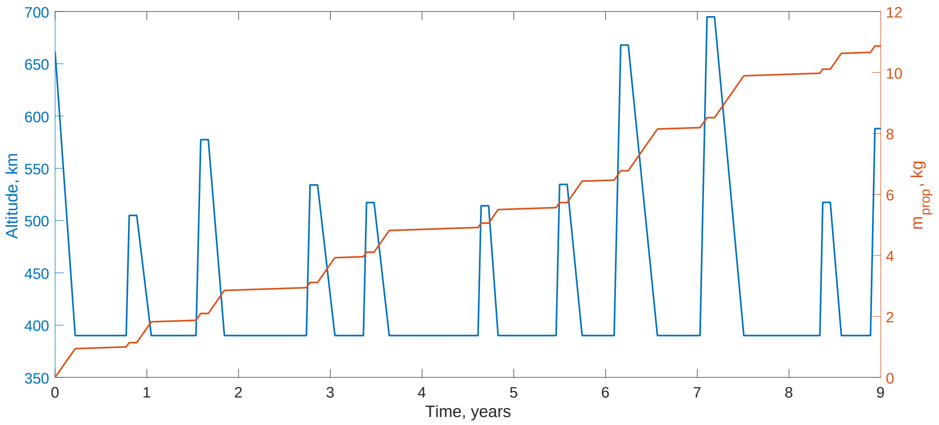


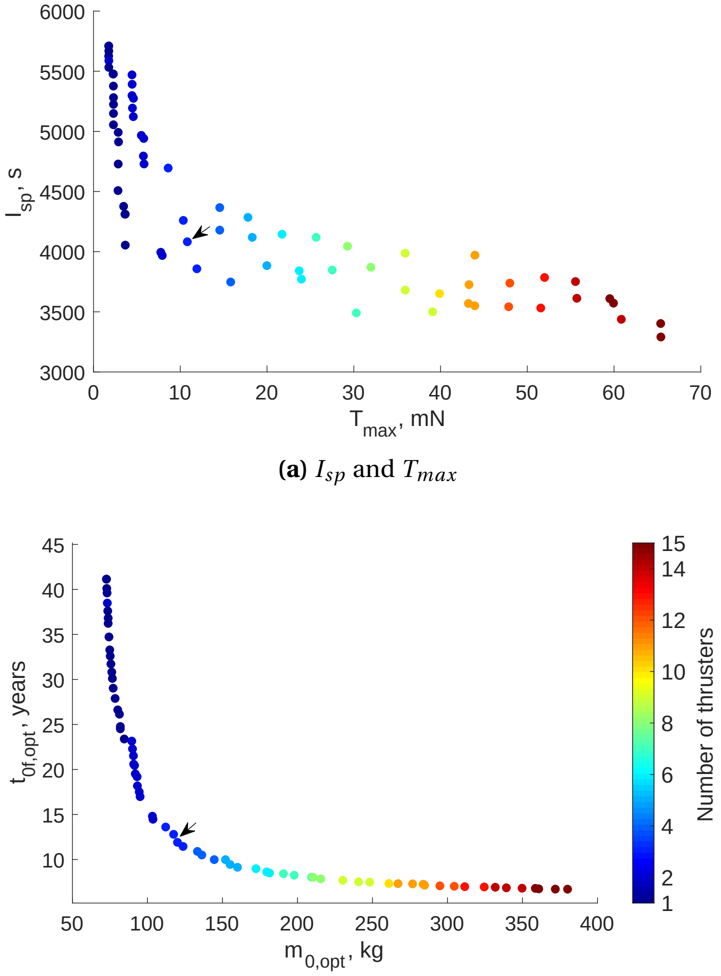
Figure 8.7: Altitude and propellant mass for the selected MADR sequence (Table 8.2).

Figure 8.8 illustrates the EP properties and Pareto front obtained from the GA-ANN methodology for the selected MNR sequence. Plot (a) shows the values of I_{sp} and T_{max} associated to the optimal points in the Pareto plot (b), which presents the two objective functions, namely $m_{0,opt}$ on the X-axis and $t_{0,f,opt}$ on the Y-axis. The colour of the points in the plots shows the number of the on-board thrusters required to achieve the desired T_{max}, going from 1 (dark blue) to 15 (dark red), as indicated in the colour bar in the figure.

Similarly to the MADR mission, high values of I_{sp} and small values of T_{max} allow to fly the sequence with a smaller initial mass but in a considerably longer amount of time. On the other hand, relatively low values of I_{sp} and high values of T_{max} allow to fly the sequence faster but a higher initial mass is required, to allow for a larger number of thrusters on-board and a consequently larger propellant mass. It should be noticed that in the MADR case all the spectrum of I_{sp} values was able to provide a Pareto-optimal solution for different values of T_{max} within the Pareto front. Differently, for this selected MNR mission only values of I_{sp} larger than 3,500 s give a solution in the Pareto front. This means that values of specific impulse below 3,500 s result in a mission associated with larger TOF and/or larger initial mass.

The point in the Pareto front (indicated by the arrow in the plots of Figure 8.8), which is selected for further optimisation, corresponds to an initial mass of 120.10 kg (of which 43.42 kg are propellant mass) which can fly the mission with total TOF of 11.91 years and a propulsion system with I_{sp} = 4082 s and T_{max} = 10.81 mN, provided by three Mini thrusters. This sequence is recalculated by solving the dynamics of the system and finding the optimal solution. The optimal control problem is solved by using GPOPS-II, as explained in Section 6.4.

Table 8.3 presents the characteristics of the transfers, i.e. departure and arrival dates, TOF, m_{prop} and stay time at the asteroid to allow for proximity operations. Additionally, Figure 8.9 illustrates the heliocentric ecliptic-plane view of the complete



(a) I_{sp} and T_{max}

(b) Pareto front

Figure 8.8: EP characteristics, I_{sp} and T_{max} (a) and Pareto front (b) for the MNR mission.

Table 8.3: Characteristics of the selected MNR sequence flown by a small satellite of 120 kg and equipped with three Mini thrusters delivering I_{sp} = 4082 s and T_{max} = 10.81 mN.

Leg	Departure	Arrival	TOF, days	m_{prop}, kg	Stay time, days
Earth - 2014 WX202	2035-03-12	2036-11-25	625	6.01	50
2014 WX202 - 2008 EA9	2037-01-17	2038-12-24	706	5.70	86
2008 EA9 - 2015 VO142	2039-03-21	2040-03-19	364	5.03	180
2015 VO142 - 2012 EP10	2040-09-15	2041-04-01	197	4.45	50
2012 EP10 - 2013 CY	2041-05-21	2043-11-21	913	9.89	50
2013 CY - 2014 UV210	2044-01-10	2046-08-24	956	11.32	–

trajectory.

It results that a small satellite of 120.10 kg is able to fly the NEA sequence equipped

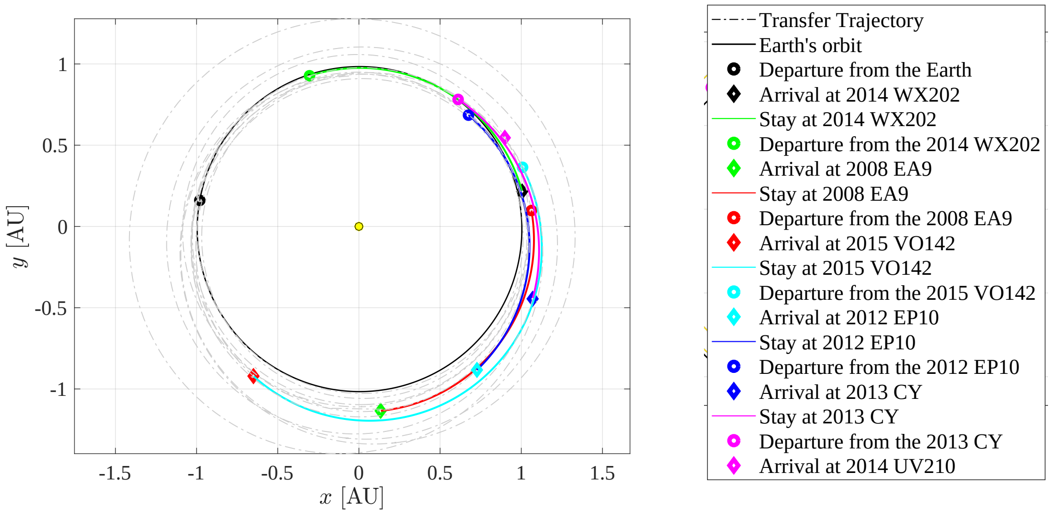


Figure 8.9: Heliocentric ecliptic-plane view of the selected MNR sequence (Table 8.3).

with three Mini low-thrust thrusters. The total duration of the mission is 11.44 years and requires 42.40 kg of propellant mass. Comparing the optimised results with those obtained by the ANN, it follows that a percentage error of 4.1% for the TOF and 2.4% for the total propellant mass is obtained.

It is worth noting that the average percentage errors achieved in the MLT analysis are generally lower than the average percentage errors obtained for the MADR and MNR missions where the EP system is fixed (detailed in Chapters 5 and 6, respectively). These errors are calculated between the network outputs and the optimal values, which are computed once the sequences have been selected and optimised. This would suggest that, when the EP system is not fixed and its characteristics are introduced as an input to the network, the ANN is outperforming the shape-based method and identifying the optimal values of propellant mass and TOF of the transfers. This may be linked to the fact that more flexibility is introduced into the network by not fixing the propulsion system, which as a consequence tends to perform better with respect to the optimised values.

It should be noticed that in Table 6.6 the same sequence was optimised for a system of initial mass 1500 kg and equipped with an EP system with I_{sp} = 3000 s and T_{max} = 0.1 N. This system requires around 685 kg of propellant mass to fly the mission [39]. Thus, this analysis also demonstrates that, when the optimal EP system is selected, a small satellite is able to fly the same sequence in a similar amount of time, but reducing significantly the necessary propellant mass and initial launch mass (provided that this is allowed by the mission requirements), for missions which allow for a smaller payload mass. For this specific MNR mission, a reduction in mass of more than 90% is registered when a small satellite is used.

9

Conclusions

The main research question which this PhD thesis aims at addressing has been formulated as:

> *Can artificial neural networks be used to efficiently design multi-target space missions?*

The main research question holds several research opportunities, which were introduced in Chapter 1 and translated into ten research sub-questions. This chapter concludes this dissertation with the purpose of summarising the work conducted and the research outcomes. The research sub-questions can now be addressed in Section 9.1, with a particular focus on the contributions of this study to the field of multi-target mission design. From answering the sub-questions, the main research question can ultimately be answered in a complete, concise and accurate manner. Finally, the possible areas of further improvement and investigation of this research topic are proposed in Section 9.2.

9.1 Summary of the Work and Findings

Multi-target missions are an attractive solution to visit multiple bodies, increasing the scientific return and reducing the cost, compared to multiple missions to individual targets. Examples of multi-target missions, which are currently being investigated, are multiple active debris removals (MADR) and multiple near-Earth asteroids rendezvous (MNR) missions. MADR missions allow for the disposal of inactive satellites, preventing the build-up of space junk and allowing to replace ageing agents in a constellation. Similarly, MNR missions allow to reduce the cost of each NEA observation and increase the possibility of visiting multiple asteroids of interest in a single mission.

Previous missions and research studies which have been conducted in the sphere of multi-target missions are analysed in Chapter 2, to understand the methods which are most commonly used and the consequent results and limitations. From this investigation, it was possible to identify the research need which we intended to meet through this dissertation.

To preliminary design multi-target missions, a complex global optimisation problem needs to be solved. This problem is composed by (i) a large discrete combinatorial part, which aims at identifying the most advantageous sequence of objects, and (ii) a continuous part, which aims at solving the optimal trajectory control problem to calculate the low-thrust trajectories between pairs of objects. Since more than 36,000 space debris objects (for sizes larger than 10 cm) are in LEO and more than 28,000 NEAs are known to date, solving the global optimisation problem to design multi-target missions is exceptionally intricate, computationally expensive and time consuming. It is evident that there is the need of developing a methodology for a quick estimation of the cost and time of flight (TOF) of low-thrust transfers. This would allow to significantly reduce the computation time required for the preliminary design of multi-target missions while maintaining a satisfactory accuracy.

This work proposes a method based on artificial neural networks (ANNs) to quickly estimate the transfer cost and time between pairs of objects using a low-thrust propulsion system. The neural network output is integrated within a sequence search (SS) algorithm based on a tree-search method and breadth-first criterion to identify the most convenient sequences of objects for MADR and MNR missions. In essence, the ANN is able to quickly solve the continuous part of the complex global optimisation problem, while the SS algorithm solves the large discrete combinatorial part. This PhD research investigates whether the proposed methodology (SS-ANN) which uses both the SS algorithm and a well-trained ANN can solve the complex global optimisation problem quicker compared to other currently used methods and accurately identify the optimal sequences of objects to satisfy the goals and requirements of multi-target missions.

The sequences which are computed by the SS-ANN platform can be further analysed and optimised by solving the dynamics of the system for each leg so that the full trajectory and control history can be computed, eventually finding a solution to the trajectory optimal control problem. This allows to verify the feasibility of the mission trajectory with the selected low-thrust propulsion system. The performance of the presented methodology is analysed by comparing the outputs computed by the network with the optimal values and by comparing the computational time required to run the SS-ANN platform with the one needed to run other techniques which are the most-commonly used in the literature. The former analysis provides an assessment of the accuracy of the proposed method, while the latter analysis investigates the benefits of using ANNs in terms of computational speed.

The orbital dynamics models for both MADR and MNR missions, the optimal optimisation methods to compute a low-thrust trajectory and control history, the artificial neural networks and the sequence search logic were presented in Chapter 3. These methods were necessary to conduct this PhD work and address the research sub-questions, as reported in the remaining of this section. For each sub-question a summary of the work performed is provided, together with a description of the main findings.

Artificial neural networks

1. *Can an ANN provide an accurate, quick estimation of the cost and duration of a low-thrust transfer?*

An ANN can be designed and trained to estimate the cost and TOF of low-thrust transfers between pairs of objects. The analysis is conducted in Chapter 4. For multi-target missions, the network inputs are the orbital elements of the departure and arrival objects and the network outputs are the resulting ΔV (or propellant mass) and TOF of the low-thrust transfers. The ANN is trained with a database of samples. The optimal number of samples to include in the training is analysed and computed as a trade-off between the time required to generate those samples and the network accuracy obtained. The architecture and hyper-parameters of the network are tuned to maximise the network performance for this application. Similarly, it is demonstrated that using different parameterisations of the orbit as network inputs affects the network performance, thus the one which allows for the highest accuracy are selected. It results that a well-trained ANN can provide an estimation of the cost and duration of a low-thrust transfer in a fraction of a second. Additionally, the performance and regression analyses of the network show that the ANN exhibits a high correlation coefficient and low mean squared error (MSE), also for the validation set (confirming that a good generalisation capability is obtained).

2. *Can the ANN be integrated within a sequence search algorithm to identify the most convenient sequence of objects for multi-target missions in terms of cost and/or duration?*

The trained ANN is integrated within a SS algorithm so that the most convenient sequences of objects can be identified. The SS algorithm is based on a tree-search method and and breadth-first criterion. The method requires the network inputs to be provided by the SS algorithm to the ANN, depending on the permutations of objects which are considered by the SS. The network outputs are then used by the SS algorithm to identify the best transfers in terms of, for example, the lowest TOF or propellant mass required. The performance of the SS-ANN methodology is investigated for both MADR and MNR missions, for which more details are provided in the responses to the next sub-questions.

Multiple active debris removal missions

3. *Can ANN be employed to design MADR missions and, if so, how accurately and what are the benefits in terms of computational time?*

An ANN is trained in Chapter 5 to estimate the cost and duration of low-thrust transfers for the disposal of multiple space debris. The trained ANN reaches a correlation of almost 0.99, which indicates a extremely high performance. The network is integrated with the SS algorithm which can compute the most convenient sequences in terms of overall duration of the mission.

The mission scenario requires the chaser to thrust to realise the change in altitude from one debris to the next one, and to exploit the Earth's oblateness gravitational perturbation to achieve the change in the right ascension of the ascending node (RAAN) through RAAN-phasing orbits. It is demonstrated that the choice of the phasing altitude has an impact on the final cost and duration of the mission and, consequently, on the number of debris which can be disposed in a given time frame.

Three sequences have been selected to verify the performance of the proposed methodology. The simulations are performed for a system with 400-kg initial mass and electric propulsion (EP) system with specific impulse I_{sp} = 2000 s and T_{max} = 21 mN. It is shown that up to 13 debris can be disposed of within 10 years, if the optimal RAAN-phasing orbits are selected. The proposed methodology ensures that a high accuracy of the computed MADR sequences is met, with a final average percentage error lower than 8% with respect to the optimal values of propellant mass and TOF.

Employing machine learning techniques within the SS algorithm greatly reduces the computational time to design MADR missions compared to previously used methods. The SS-ANN performance is compared with one commonly-used method which is currently employed by the industry. It results that SS-ANN allows to reduce the computational time to find the best MADR sequences by about 26 times, when 100 debris objects are considered. Also, being SS-ANN able to explore the full search space in less time, this methodology can select sequences which are shorter in duration and less expensive (being the propellant mass required lower).

Multiple NEA rendezvous missions

4. *Can ANN be employed to design MNR missions and, if so, how accurately and what are the benefits in terms of computational time?*

The SS-ANN methodology for the preliminary design of MNR missions using low-thrust propulsion systems is investigated in Chapter 6. Three of calculated sequences are selected and fully optimised to assess the results and demonstrate the validity of this methodology. Two different low-thrust propulsion systems are selected: an EP system (I_{sp} = 3000 s and T_{max} = 0.1 N) and a solar sailing propulsion system (with a characteristic acceleration of 0.2 mm/s^2). A final average error lower than 10% in TOF and ΔV is obtained with respect to the final optimised values.

In Chapter 7, the option of using the SS-ANN platform to address MNR missions with sample return to Earth is explored. The SS algorithm is enhanced so that the Earth can be reached at the end of the sequence and, additionally, a specific NEA of interest can be targeted within a sequence. In this case, the interest value of the objects is introduced and can be used as objective function to maximise when selecting the most convenient sequences (apart from minimising the associated ΔV). It is shown that, when the interest value is considered for the selection of the sequences, more interesting objects can be visited, increasing the overall appeal of the mission at the cost of a larger ΔV. Three sequences are selected for further analysis and are fully optimised to obtain the flight trajectory and control history. Near-term low-thrust propulsion enables to rendezvous five asteroids, and return samples to Earth in about ten years from launch.

The performance of the SS-ANN platform is compared to other methodologies, which are most-commonly used in the literature. It is demonstrated that the ANN makes the algorithm almost 100 times faster compared to when no machine learning techniques are used, while maintaining a similar level of accuracy.

Multi-objective optimisation of low-thrust system analysis

5. *Can the ANN be used to optimise the propulsion system according to the goals and requirements of a given multi-target mission?*

Chapter 8 investigates the use of an ANN to analyse the effect of different EP systems on multi-target missions. The ANN is trained with samples of low-thrust transfers calculated with different propulsion capabilities, i.e. specific impulse and maximum thrust, which are also included in the network input vector. The ANNs trained with such databases reach a high accuracy in computing the network output, as confirmed by the network performance and regression analysis.

It is demonstrated that a trained ANN can help to ultimately identify the most convenient low-thrust system to fly a given multi-target mission, when it is paired with a multi-objective genetic algorithm (GA). In this case, EP systems for small satellites are considered as they can be propelled by smaller thrusters and overall reduce the launch cost (for missions where a smaller payload mass is allowed). However, the same analysis and considerations could be expanded to larger satellites or different types of propulsion systems.

The logic of the proposed methodology (GA-ANN) includes an inner optimisation over the initial mass m_0, where the ANN is used to estimate the cost and duration of the sequence, and an outer optimisation performed by the GA over the propulsion properties. This demonstrates the effects of the different EP capabilities on a multi-target mission in terms of the required initial mass m_0 and total duration. Since two objective functions are considered, the multi-objective GA identifies a number of Pareto-optimal solutions, all of which are considered equally good, for different values of I_{sp} and T_{max}. The selection of one solution over the others depends on the goals and/or requirements of the mission and requires a decision making process.

The GA-ANN methodology is run for an MADR sequence and an MNR sequence. In both cases, a point along the Pareto front is selected for further analysis and optimisation, resulting in a limited error between the ANN results and optimal results.

In conclusion, the main research question can be answered:

> *Can artificial neural networks be used to efficiently design multi-target space missions?*

The work conducted and the research outcomes, which are presented in the chapters of this dissertation, demonstrate that ANNs are an advanced methodology which can offer a quick estimation of the cost and duration of low-thrust transfers. The ANN can be integrated with a SS algorithm which, compared to other commonly-used methodologies applied in the literature, allows to identify the most convenient sequences of objects up to 26 times faster when 100 objects are considered and up to 100 times faster when about 2,000 objects are considered. In fact, due to the nature of the tree search, a linear increase in the number of objects corresponds to an exponential increase in the number of permutations between the objects, i.e. an exponential increase in the number of calculations which need to be performed to compute the object sequences. It follows that the benefit of using an ANN within a sequence search algorithm is even more significant for larger databases of objects. To evaluate the performance of the network, the output of the network is compared to the results obtained from optimising the sequences identified by SS-ANN. It results that the final average percentage error is generally lower than 10%, confirming that a satisfactory accuracy of the ANN can be obtained when the network hyper-parameters and inputs are optimised for the application of multi-target missions.

9.2 Recommendations for Future Work

The work and outcome presented in this PhD thesis have satisfied the objectives which were identified in Chapter 1, by analysing in detail the use of ANNs for the preliminary design of multi-target missions and discussing the possible applications, such as MADR missions, MNR missions and to identify the most convenient low-thrust propulsion system for these kinds of missions (MLT analysis). However, some assumptions were used in certain instances to allow for an easier mathematical representation (and consequent treatment) of the problem. Also, some aspects of the proposed methodology were just preliminary outlined. In this section, recommendations for future work are given.

MADR dynamics. In the MADR case study (Chapter 5), some assumptions are considered to keep the model simple. In particular, a set of debris objects on circular orbits and at the same inclination is considered and the phasing along the orbit between the chaser and debris is neglected in the transfer model. It follows that the MADR transfer legs only require the chaser to realise changes in altitude and RAAN to transfer from the departure object to the arrival object. These assumptions are chosen to be representative of a real MADR mission with satellite constellations on circular orbits, at the same inclination and spaced in RAAN. However, they might be at different altitudes because they start de-orbiting (e.g. due to malfunctioning of the thrusters or to the depletion of their fuel). Also, the correct phasing can be attained with minimal propulsive effort, and often with little additional transfer time because of the large number of revolutions which characterises the low-thrust transfer legs for MADR missions. Although these assumptions allow to simplify the transfer model and, consequently, minimise the computational time required to generate the database to train the ANN, extending the analysis to include eccentric and inclined objects and the phasing between the objects and the chaser (to realise a rendezvous) would be beneficial to widen the scope of the research.

Search over launch dates. In this work the launch dates used by the SS-ANN platform to calculate the most convenient sequences of objects are fixed. This is sufficient to prove the effectiveness of using an ANN in combination with a sequence search to design multi-target missions using a low-thrust propulsion system. However, this can be complemented with a systematic search over several launch dates if a launch window is given. As a result more candidate sequences can be identified by the methodology with different objects being selected depending on the launch date. Results of such an investigation can be very interesting to a space mission designer as it provides a tool to select the best sequences of objects over an available launch window.

Non-feasible transfers. As shown in Section 6.4.3, the trained ANN is not able to detect non-feasible low-thrust transfers. This is related to how the network is trained and what samples are included in the training database. In fact, once the ANN training is completed, the ANN replicates the behaviour of the method used to compute the samples included in the database. For MNR transfers, the shape-based method is used and only transfers which were deemed feasible by the shape-based method are included in the training database. Consequently, in cases of unfeasible transfers, the network returns an incorrect guess of the transfer cost and duration (not having been trained on such or similar transfers). It is recommended to develop the methodology further so that the network can be trained also to identify transfers which are unfeasible with the given propulsion system. By doing so, sequences for which a feasible solution per leg is not found can be discarded by the SS-ANN platform and more focus can be given to sequences where all transfers are feasible.

ANN to compute the optimal sequences. The purpose of this research is to identify a methodology to efficiently solve the large combinatorial part and, simultaneously, the continuous part of the complex global optimisation problem to find the most convenient sequences of objects. In this context, this work demonstrates how using ANNs to solve the continuous sub-problem reduces the computational time, while maintaining a good accuracy. Considering the complexity of the travelling salesman problem, it would be worth exploring the possibility of training an ANN to directly tackle the optimal sequencing of the objects. To this end, the neural network could be trained with sequences of optimal multi-target missions (in terms of ΔV or TOF) so that it learns how to find novel optimal sequences depending on the goals and the requirements of a given mission.

GA-SS-ANN. Within the MLT analysis, which is presented in Chapter 8, the multi-objective GA is used with the ANN to analyse how different low-thrust propulsion capabilities affect the outcome of multi-target missions and, ultimately, to identify the optimal EP system for a specific mission. In this context, optimal MADR and MNR sequences which were selected by the SS-ANN platform, respectively, are used to demonstrate the capabilities of the GA-ANN method to find the optimal low-thrust system to perform those missions. It is recommended to examine the possibility to integrate the SS algorithm within the GA-ANN method, in order to implement an even more comprehensive platform (GA-SS-ANN) which can identify the optimal sequences of objects and, simultaneously, optimise the low-thrust system for the computed sequences.

References

[1] D. Snelling, E. Devereux, N. Payne, M. Nuckley, G. Viavattene, M. Ceriotti, S. Wokes, G. Di Mauro, H. Brettle, Innovation in Planning Space Debris Removal Missions Using Artificial Intelligence and Quantum-Inspired Computing, 8th European Conference on Space Debris, Darmstadt, Germany, 20-23 Apr (2021).

[2] A. Peloni, M. Ceriotti, B. Dachwald, Solar-Sail Trajectory Design for a Multiple Near-Earth-Asteroid Rendezvous Mission, Journal of Guidance, Control, and Dynamics 39 (12) (2016) 2712–2724. doi:10.2514/1.G000470.

[3] A. Mereta, D. Izzo, Target selection for a small low-thrust mission to near-Earth asteroids, Astrodynamics 2 (3) (2018) 249–263. arXiv:arXiv:1802.02418v1.

[4] D. Izzo, L. F. Simoes, C. H. Yam, F. Biscani, D. Di Lorenzo, A. Bernardetta, A. Cassioli, GTOC5 : Results from the European Space Agency and University of Florence, Acta Futura 8 (November) (2014) 45–56. doi:10.2420/AF08.2014.45.

[5] Y. Song, S. Gong, Solar-Sail Trajectory Design of Multiple Near Earth Asteroids Exploration Based on Deep Neural Network, Aerospace Science and Technology 91 (2019) 28–40. doi:10.1016/j.ast.2019.04.056.

[6] G. Tang, F. Jiang, J. Li, Trajectory Optimization for Low-Thrust Multiple Asteroids Rendezvous Mission, AIAA/AAS Astrodynamics Specialist Conference (2015). doi:10.1515/jnum-2014-0003.

[7] M. Di Carlo, J. M. R. Martin, N. Gomez, M. Vasile, Optimised Low-Thrust Mission to the Atira Asteroids, Advances in Space Research 59 (2017) 1724–1739. doi:10.1016/j.asr.2017.01.009.

[8] https://www.esa.int/Safety_Security/Space_Debris/Active_debris_removal, accessed: 2022-03-26.

[9] C. Bonnal, J. Ruault, M. Desjean, Active debris removal: recent progress and current trends, Acta Astronautica 85 (2013).

[10] J. Liou, A parametric study on using active debris removal for LEO environment remediation, Advances in Space Research 47 (11) (2011).

[11] www.space-track.org, accessed: 2021-09-25.

[12] www.esa.int/Safety_Security/Space_Debris/Space_debris_by_the_

numbers, accessed: 2022-03-30.

[13] https://sdup.esoc.esa.int/discosweb/statistics/, accessed: 2022-03-30.

[14] D. J. Kessler, B. G. Cour-Palais, Collision frequency of artificial satellites: The creation of a debris belt, Advances in Space Research (1978) 2637–2646.

[15] D. McKnight, R. Witner, F. Letizia, S. Lemmens, L. Anselmo, C. Pardini, A. Rossi, C. Kunstadter, S. Kawamoto, V. Aslanov, J. Dolado Perez, V. Ruch, H. Lewis, M. Nicolls, L. Jing, S. Dan, W. Dongfang, A. Baranov, D. Grishko, Identifying the 50 statistically-most-concerning derelict objects in LEO, Acta Astronautica 181 (2021). doi:https://doi.org/10.1016/j.actaastro.2021.01.021.

[16] V. Braun, A. Lupken, S. Flegel, J. Gelhaus, M. Mockel, C. Kebschull, C. Wiedemann, P. Vörsmann, Active debris removal of multiple priority targets, Advances in Space Research 51 (9) (2013).

[17] N. Berend, X. Olive, Bi-objective optimization of a multiple-target active debris removal mission, Acta Astronautica 122 (2013).

[18] N. Van der Pas, J. Lousada, C. Terhes, M. Bernabeu, W. Bauer, Target selection and comparison of mission design for space debris removal by DLR's advanced study group, Acta Astronautica 102 (2014).

[19] B. Cheng, Y. Yu, H. Baoyin, Asteroid surface impact sampling: dependence of the cavity morphology and collected mass on projectile shape, Advances in Space Research 7 (10004) (2017).

[20] J. J. Lissauer, I. de Parter, Fundamental Plantary Science, Cambridge University Press, 2013.

[21] https://cneos.jpl.nasa.gov/orbits/elements.html, accessed: 2021-10-17.

[22] B. D. Tapley, B. E. Schutz, G. H. Born, Statistical Orbit Determination, Elsevier Academic Press, Burlington, Massachusetts, 2004.

[23] H. Li, S. Chen, D. Izzo, H. Baoyin, Deep Networks as Approximators of Optimal Transfers Solutions in Multitarget Missions, Acta Astronautica (2019). doi:10.1016/j.actaastro.2019.09.023.

[24] J. T. Betts, Survey of Numerical Methods for Trajectory Optimization, Journal of Guidance, Control, and Dynamics 21 (2) (1998) 193–207. doi:10.2514/2.4231.

[25] J. R. Wertz, Orbit & Constellation Design & Management, Springer, New York,

2009.

[26] R. G. Jahn, Physics of Electric Propulsion, McGraw-Hill Book Company, New York, 2006.

[27] C. R. McInnes, Solar Sailing - Technology, Dynamics and Mission Applications, Springer, 1999. doi:10.1007/978-1-4471-3992-8.

[28] X. Zeng, S. Gong, J. Li, Fast solar sail rendezvous mission to near Earth asteroids, Acta Astronautica 105 (2014) 40–56. doi:10.1016/j.actaastro.2014.08.023.

[29] J. Heiligers, J. M. Fernandez, O. R. Stohlman, W. K. Wilkie, Trajectory design for a solar-sail mission to asteroid 2016 HO3, Astrodynamics 3 (2019) 231–246. doi:10.1007/s42064-019-0061-1.

[30] R. G. Jahn, E. Y. Choueiri, Electric Propulsion, Encyclopedia of Physical Science and Technology (Third Edition) (2003) 125–141.

[31] B. A. Conway, The Problem of Spacecraft Trajecotry Optimization, Cambridge University Press, Cambridge, UK, 2010.

[32] H. Yang, G. Tang, F. Jiang, Optimization of observing sequence based on nominal trajectories of symmetric observing configuration, Astrodynamics 2 (1) (2018) 25–37. doi:10.1007/S42064-017-0009-2.

[33] D. Hennes, D. Izzo, D. Landau, Fast Approximators for Optimal Low-Thrust Hops Between Main Belt Asteroids, IEEE Symposium Series on Computational Intelligence (SSCI) (2016). doi:10.1109/SSCI.2016.7850107.

[34] A. Mereta, D. Izzo, A. Wittig, Machine Learning of Optimal Low-Thrust Transfers Between Near-Earth Objects, Hybrid Artificial Intelligent Systems (2017) 543–553doi:10.1007/978-3-319-59650-146.

[35] H. Peng, X. Bai, Artificial Neural Network–Based Machine Learning Approach to Improve Orbit Prediction Accuracy, AIAA Journal of Spacecraft and Rockets 55 (5) (2018) 1–13. doi:10.2514/1.A34171.

[36] B. Dachwald, Optimization of interplanetary solar sailcraft trajectories, Journal of Guidance, Control, and Dynamics 27 (1) (2004). doi:10.2514/1.9286.

[37] G. Viavattene, M. Ceriotti, Artificial neural network design for tours of multiple asteroids, Springer, Lectures Notes in Computer Science (2020). doi:10.1007/978-3-030-61705-9_63.

[38] G. Viavattene, E. Devereux, D. Snelling, N. Payne, S. Wokes, M. Ceriotti, Design of multiple space debris removal missions using machine learning, Acta

Astronautica 193, (2022) 277–286. doi:10.1016/j.actaastro.2021.12.051.

[39] G. Viavattene, M. Ceriotti, Artificial Neural Networks for Multiple NEA Rendezvous Missions with Continuous Thrust, AIAA Journal of Spacecraft and Rockets (2021). doi:10.2514/1.A34799.

[40] J. Liou, N. L. Johnson, Instability of the present LEO satellite populations, Advances in Space Research 41 (7) (2008) 1046–1053.

[41] J. Liou, Collision activities in the future orbital debris environment, Advances in Space Research 38 (2006) 2102—-2106.

[42] H. Klinkrad, Space Debris: Models and Risk Analysis, Springer-Praxis (2006).

[43] A. Kelley, J. Butler, X. Xiong, X. J. S. Gu, Terra mission operations: Launch to the present (and beyond), Earth Observing Systems (2014). doi:10.1117/12.2061253.

[44] https://ntrs.nasa.gov/citations/20170005606, accessed: 2022-03-23.

[45] www.space.com/5542-satellite-destroyed-space-collision.html, accessed: 2022-03-23.

[46] E. L. Christiansen, J. L. Hyden, R. P. Bernhard, Space Shuttle debris and meteoroid impacts, Advances in Space Research 34 (5) (2004) 1097–1103. doi:10.1016/j.asr.2003.12.008.

[47] D. Lear, E. Christiansen, J. Hyde, J. Herrin, F. Lyons, J. Kerr, S. Ryan, Investigation of Shuttle Radiator Micro-Meteoroid & Orbital Debris Damage, AIAA Structures, Structural Dynamics, and Materials Conference (AIAA 2009-2361) (2009).

[48] Y. Akahoshi, T. Nakamura, Fukushige, S. Furusawa, S. Kusunoki, Y. Machida, T. Koura, K. Watanabe, S. Hosoda, T. Fujita, M. Choa, Influence of space debris impact on solar array under power generation, International Journal of Impact Engineering 35 (12) (2008) 1678–1682. doi:10.1016/j.ijimpeng.2008.07.048.

[49] J. L. Hyde, E. L. Christiansen, D. M. Lear, Observations of MMOD Impact Damage to the ISS, First International Orbital Debris Conference (2019).

[50] https://phys.org/news/2021-04-piece-spacex-rocket-debris-washington.html, accessed: 2022-03-23.

[51] J. Forshaw, G. Aglietti, S. Fellowes, T. Salmon, I. Retat, et al, The active space debris removal mission RemoveDebris. Part 1: from concept to launch, Acta Astronautica 168 (2020). doi:10.1016/j.actaastro.2019.09.002.

[52] https://astroscale.com/astroscale-celebrates-successful-

launch-of-elsa-d, accessed: 2022-03-23.

[53] https://www.esa.int/Safety_Security/Clean_Space/ESA_commissions_world_s_first_space_debris_removal, accessed: 2022-03-23.

[54] J. Liou, N. Johnson, A sensitivity study of the effectiveness of active debris removal in LEO, Acta Astronautica 64 (2009) 236–243.

[55] J. Liou, N. Johnson, N. Hill, Controlling the growth of future LEO debris populations with active debris removal, Acta Astronautica 66 (2010) 648–653.

[56] J. Yu, X.-q. Chen, L.-h. Chen, Optimal planning of LEO active debris removal based on hybrid optimal control theory, Advances in Space Research 55 (11) (2015) 2628–2640.

[57] M. Jorgensen, I. Sharf, Optimal planning for a multiple space debris removal mission using high- accuracy low-thrust transfers, Acta Astronautica 172 (2020) 56–69.

[58] C. Colombo, S. Huang, G. Borelli, F. Cavenago, M. Nugnes, J. Gonzalo, G. Gaias, M. Massari, L. Vallini, M. Petit, P. Guerrieri, M. Valli, S. Antonetti, Mission Analysis and Design for an Active Debris Removal Service for Large Constellations, 8-th European Conference on Space Debris, ESA (2021).

[59] J. Pollard, Bi-objective optimization of a multiple-target active debris removal mission, Joint Propulsion Conference and Exhibit, AIAA (1998). doi:10.2514/6.1998-3486.

[60] D. Izzo, I. Getzner, D. Hennes, L. Simoes, Evolving Solutions to TSP Variants for Active Space Debris Removal, GECCO 2015: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation (2015) 1207–1214doi:10.1145/2739480.2754727.

[61] M. Shan, J. Guo, E. Gill, Review and comparison of active space debris capturing and removal methods, Progress in Aerospace Sciences 80 (2016) 18–32.

[62] https://sophia.estec.esa.int/gtoc_portal/, accessed: 2021-12-19.

[63] A. Petropoulos, D. Grebow, D. Jones, G. Lantoine, A. Nicholas, J. Roa, J. Senent, J. Stuart, N. Arora, T. Pavlak, T. Lam, T. McElrath, R. Roncoli, D. Garza, N. Bradley, D. Landau, Z. Tarzi, F. Laipert, E. Bonfiglio, M. Wallace, J. Sims, GTOC9: Methods and Results from the Jet Propulsion Laboratory Team, NASA Jet Propulsion Laboratory (2011).

[64] H. Shen, T. Zhang, L. Casalino, D. Pastrone, Optimization of Active Debris Removal Missions with Multiple Targets, AIAA Journal of Spacecraft and Rockets

(2017). doi:10.2514/1.A33883.

[65] P. Michel, F. E. DeMeo, W. F. Bottke, Asteroids IV, The University of Arizona Press, 2015.

[66] J. S. Lewis, Rain of iron and ice, Addison-Wesley Pub. Co. (1996).

[67] M. Vergaaij, C. McInnes, M. Ceriotti, Economic assessment of high-thrust and solar-sail propulsion for near-earth asteroid mining, Advances in Space Research, Elsevier (2020). doi:10.1016/j.asr.2020.06.012.

[68] https://ssd.jpl.nasa.gov/sb/, accessed: 2022-03-24.

[69] https://solarsystem.nasa.gov/asteroids-comets-and-meteors/asteroids/in-depth/, accessed: 2022-03-24.

[70] R. Bertrand, R. Epenoy, B. Meyssignac, Final Results of the 4th Global Trajectory Optimisation Competition (2009), https://sophia.estec.esa.int/gtoc_portal/wp-content/uploads/2012/11/gtoc4_problem_description.pdf, accessed: 2021-12-19.

[71] B. Dachwald, W. Seboldt, Multiple Near-Earth Asteroid Rendezvous and Sample Return Using First Generation Solar Sailcraft, Acta Astronautica 57 (11) (2005) 864–875.

[72] B. Dachwald, H. Boehnhardt, U. Broj, U. Geppert, J. Grundmann, W. Seboldt, P. Seefeldt, P. Spietz, L. Johnson, E. Kührt, S. Mottola, M. Macdonald, R. McInnes, M. Vasile, R. Reinhard, Gossamer Roadmap Technology Reference Study for a Multiple NEO Rendezvous Mission, Advances in Solar Sailing (2014) 211–226.

[73] J. Englander, B. Conway, An Automated Solution of the LowThrust Interplanetary Trajectory Problem, AIAA Journal of Guidance, Control, and Dynamics 40 (1) (2008) 15–27. doi:10.2514/1.G002124.

[74] J. Englander, B. Conway, T. Williams, Automated Mission Planning via Evolutionary Algorithms, AIAA Journal of Guidance, Control, and Dynamics 35 (6) (2012) 1878—1887. doi:10.2514/1.54101.

[75] J. Sims, S. Flanagan, Preliminary Design of Low-Thrust Interplanetary Missions, AAS/AIAA Astrodynamics Specialist Conference, AAS paper 99-0328 (1999).

[76] M. Vasile, E. Minisci, M. Locatelli, Analysis of Some Global Optimization Algorithms for Space Trajectory Design, AIAA Journal of Spacecraft and Rockets 47 (2) (2010) 334–344. doi:10.2514/1.45742.

[77] J. Englander, D. Ellison, B. Conway, Global optimization of low-thrust, multiple-flyby trajectories at medium and medium-high fidelity, Advances in Space

Research (2014) 1539–1558.

[78] M. Vavrina, K. Howell, Global Low-Thrust Trajectory Optimization Through Hybridization of a Genetic Algorithm and a Direct Method, AIAA/AAS Astrodynamics Specialists Conference (2008). doi:10.2514/6.2008-6614.

[79] M. Ceriotti, M. Vasile, Automated Multigravity Assist Trajectory Planning with a Modified Ant Colony Algorithm, Journal of Aerospace Computing, Information and Communication 7 (9) (2010) 261–293. doi:10.2514/1.48448.

[80] M. Ceriotti, M. Vasile, MGA trajectory planning with an ACO-inspired algorithm, Acta Astronautica 67 (9-10) (2010) 1202–1217. doi:10.1016/j.actaastro.2010.07.001.

[81] A. Chaikin, A Man On the Moon: The Voyages of the Apollo Astronauts, 3rd Edition, New York: Penguin Books, 2007.

[82] D. Lauretta, OSIRIS-REx Asteroid Sample-Return Mission, Handbook of Cosmic Hazards and Planetary Defense (2015) 543–567doi:10.1007/978-3-319-03952-744.

[83] M. Yoshikawa, J. Kawaguchi, F. A., Hayabusa Sample Return Mission, Asteroids IV (2015) 397–418doi:10.2458/azu-uapress-9780816532131-ch021.

[84] Y. Tsuda, M. Yoshikawa, M. Abe, H. Minamino, S. Nakazawa, System design of the Hayabusa 2: Asteroid sample return mission to 1999 JU3, Acta Astronautica 91 (2013) 356–362. doi:10.1016/j.actaastro.2013.06.028.

[85] https://sophia.estec.esa.int/gtoc_portal/wp-content/uploads/2012/11/gtoc3_problem.pdf, accessed: 2021-12-19.

[86] https://researchgate.net/publication/305247918_3rd_Global_Trajectory_Optimisation_Competition_GTOC3_-_Results_found_by_the_CNES_team, accessed: 2021-12-19.

[87] K. Zhu, F. Jiang, J. Li, H. Baoyin, Trajectory Optimization of Multi-Asteroids Exploration with Low Thrust, Transactions of the Japan Society for Aeronautical and Space Sciences 52 (175) (2009) 47–54.

[88] D. Lev, R. Myers, K. Lemmer, J. Kolbeck, H. Koizumi, K. Polzin, The technological and commercial expansion of electric propulsion, Acta Astronautica 159 (2019) 213–227. doi:10.1016/j.actaastro.2019.03.058.

[89] J. Brophy, Advanced Ion Propulsion Systems for Affordable Deep-Space Missions, Acta Astronautica 52 (2-6) (2003) 309–316. doi:10.1016/S0094-5765(02)00170-4.

[90] R. G. Jahn, Physics of Electric Propulsion, McGraw-Hill, New York, 2–10, 1968.

[91] www.esa.int/Enabling_Support/Operations/SMART-1, accessed: 2022-03-24.

[92] www.esa.int/Enabling_Support/Operations/GOCE, accessed: 2022-03-24.

[93] www.esa.int/Enabling_Support/Space_Engineering_Technology/What_is_Electric_propulsion, accessed: 2022-03-24.

[94] https://spacenews.com/nasa-issues-study-contracts-for-deep-space-gateway-element/, accessed: 2022-03-24.

[95] https://sophia.estec.esa.int/gtoc_portal/wp-content/uploads/2014/11/gtoc4_summary_of_results.pdf, accessed: 2021-12-19.

[96] L. Casalino, G. Colasurdo, Problem Description for the 7th Global Trajectory Optimisation Competition, (2014), https://sophia.estec.esa.int/gtoc_portal/wp-content/uploads/2014/09/gtoc7_problem_description.pdf, accessed: 2021-12-19.

[97] M. Di Carlo, M. Vasile, J. Dunlop, Low-Thrust Tour of the Main Belt Asteroids, Advances in Space Research 62 (2018) 2026–2045. doi:10.1016/j.asr.2017.12.033.

[98] https://enpulsion.com/order/, accessed: 2022-12-22.

[99] https://enpulsion.com/wp-content/uploads/ENP2018-002.H-MICRO-Thruster-Product-Overview.pdf, accessed: 2021-12-22.

[100] G. Aglietti, B. Taylor, S. Fellowes, S. Ainley, D. Tye, C. Cox, A. Zarkesh, A. Mafficini, V. N., K. Bashford, T. Salmon, I. Retat, C. Burgess, A. Hall, T. Chabot, K. Kanani, A. Pisseloup, C. Bernal, F. Chaumette, A. Pollini, W. H. Steyn, RemoveDEBRIS: An in-orbit demonstration of technologies for the removal of space debris, The Aeronautical Journal 124 (1271) (2020) 1–23. doi:10.1017/aer.2019.136.

[101] N. Wells, R. Walker, S. Green, A. Ball, SIMONE: Interplanetary Microsatellites for NEO Rendezvous Missions, Acta Astronautica 59 (2006) 700–709. doi:10.1016/j.actaastro.2005.07.036.

[102] https://astroscale.com/astroscale-selects-rocket-lab-to-launch-phase-i-of-jaxas-debris-removal-demonstration-project, accessed: 2021-10-17.

[103] A. Daykin-Iliopoulos, R. Desai, Performance evaluation of micropropulsion systems with the application of Active Debris Removal, 50th AIAA/ASME/SAE/ASEE Joint Propulsion Conference 3540 (2014). doi:10.

2514/6.2014-3540.

[104] https://earth.esa.int/web/eoportal/satellite-missions/n/neossat, accessed: 2021-11-07.

[105] B. Dachwald, Evolutionary Neurocontrol: A Smart Method for Global Optimization of Low-Thrust Trajectories, AIAA/AAS Astrodynamics Specialist Conference and Exhibit (2004). doi:doi:10.2514/6.2004-5405.

[106] U. Geppert, B. Biering, F. Lura, J. Block, M. Straubel, R. Reinhard, The 3-step DLR-ESA Gossamer road to solar sailing, Advances in Space Research 48 (11) (2011) 1695–1701. doi:10.1016/j.asr.2010.09.016.

[107] http://sci.esa.int/sci-ft/50124-technology-readiness-level/, accessed: 2022-02-16.

[108] K. Graham, A. Rao, Minimum-Time Trajectory Optimization of Multiple Revolution Low-Thrust Earth-Orbit Transfer, AIAA Journal of Spacecraft and Rockets 52 (3) (2015) 711–727. doi:10.2514/1.A33187.

[109] B. Dachwald, Optimization of Very-Low-Thrust Trajectories Using Evolutionary Neurocontrol, Acta Astronautica 57 (2-8) (2005) 175–185. doi:10.1016/j.actaastro.2005.03.004.

[110] S. Tang, B. Conway, Optimization of Low-Thrust Interplanetary Trajectories Using Collocation and Nonlinear Programming, Journal of Guidance, Control, and Dynamics 18 (3) (1995) 599–604.

[111] R. Falck, J. Dankanich, Optimization of Low-Thrust Spiral Trajectories by Collocation, AIAA/AAS Astrodynamics Specialist Conference (2012).

[112] G. Rauwolf, V. Coverstone-Carroll, Near-Optimal Low-Thrust Orbit Transfers Generated by a Genetic Algorithm, Journal of Spacecraft and Rockets 33 (6) (1996) 859–862.

[113] C. Haissig, K. Mease, N. Vinh, Minimum-Fuel, Power- Limited Transfers Between Coplanar Elliptic Orbits, Acta Astronautica 29 (1) (1993) 1–15.

[114] A. E. Petropoulos, J. M. Longuski, Automated Design of Low-Thrust Gravity-Assist Trajectories, AIAA Paper 2000-4033 (2000).

[115] P. De Pascale, M. Vasile, Preliminary Design of Low-Thrust Multiple Gravity-Assist Trajectories, AIAA Journal of Spacecraft and Rockets 43 (5) (2006) 1065–1076. doi:10.2514/1.19646.

[116] D. Izzo, Lambert's Problem for Exponential Sinusoids, Journal of Guidance, Control, and Dynamics 29 (5) (2006) 1242–1245. doi:10.2514/1.21796.

[117] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, The MIT Press, Cambridge, Massachusetts, 2016.

[118] F. Hsu, Behind Deep Blue: Building the Computer That Defeated the World Chess Champion, Princeton University Press, 2002.

[119] S. Russel, P. Norvig, Artificial Intelligence: A modern approach, Prentice Hall Series in Artificial Intelligence (Third Edition), 2010.

[120] L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. A. Zadeh, J. M. Zurada, Artificial Intelligence and Soft Computing, Springer, Zakopane, Poland, 2013. doi:10.1007/978-3-642-38658-9.

[121] S. Suresh, S. R. Rammohan, K. Bharath, Impact of Smart Technologies and Artificial Intelligence (AI) Paving Path Towards Interdisciplinary Research in the Fields of Engineering, Arts, Humanities, Commence, Economics, Social Sciences, Law and Management - Challenges and Opportunities, Shanlax Publications), 2021.

[122] C. Sánchez-Sánchez, D. Izzo, Real-time optimal control via Deep Neural Networks: study on landing problems, Journal of Guidance, Control, and Dynamics 41 (3) (2018) 1122–1135. arXiv:1610.08668, doi:10.1023/B:CJOP.0000010527.13037.22.

[123] R. Xie, A. Dempster, An on-line deep learning framework for low-thrust trajectory optimisation, Aerospace Science and Technology 18 (Nov 2021).

[124] H. Li, H. Baoyin, F. Topputo, Neural Networks in Time-Optimal Low-Thrust Interplanetary Transfers, IEEE Access 7 (2019). doi:10.1109/ACCESS.2019.2946657.

[125] D. Izzo, C. Sprague, D. Tailor, Machine learning and evolutionary techniques in interplanetary trajectory design, Computing Research Repository (CoRR) abs/1802.0 (2018). arXiv:1802.00180.
URL http://arxiv.org/abs/1802.00180

[126] C. Ampatzis, D. Izzo, Machine learning techniques for approximation of objective functions in trajectory optimisation, IJCAI-09 workshop on artificial intelligence in space (2009) 1–6.

[127] A. Cassioli, D. Di Lorenzo, M. Locatelli, F. Schoen, M. Sciandrone, Machine Learning for Global Optimisation, Computational Optimization and Applications 51 (1) (2012) 279–303.

[128] D. Izzo, D. Hennes, L. F. Simões, M. Märtens, Designing complex interplanetary trajectories for the Global Trajectory Optimization competitions, Springer

Optimization and Its Applications 114 (2016) 151–176. arXiv:1511.00821, doi:10.1007/978-3-319-41508-6_6.

[129] R. Rojas, Neural Networks: A Systemic Introduction, Springer, New York, 1996. arXiv:arXiv:1411.3159v1, doi:10.1016/0893-6080(94)90051-5.

[130] S. Haykin, Neural Networks: A Comprehensive Foundation, 2nd Edition, Pearson, 1999.

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

[132] J. T. Betts, Practical Methods for Optimal Control and Estimation Using Nonlinear Programming, 2nd Edition, SIAM Press, Philadelphia, 2010.

[133] D. V. Schroeder, An Introduction to Thermal Physics, 3rd Edition, Addison-Wesley, 2000.

[134] H. H. Goldstine, A History of the Calculus of Variations from the 17th through the 19th Century, Vol. 5, Springer-Verlag, New York, USA, 1980. doi:10.1007/978-1-4613-8106-8.

[135] A. Tewari, Optimal Space Flight Navigation - An Analytical Approach, Birkhäuser, Springer, 2019. doi:978-3-030-03789-5.

[136] D. G. Hull, Optimal Control Theory for Applications, Springer-Verlag, New York, USA, 2003.

[137] I. M. Ross, A Primer on Pontryagin's Principle in Optimal Control, Publishers, Collegiate, 2015.

[138] B. A. Conway, A Survey of Methods Available for the Numerical Optimization of Continuous Dynamic Systems, Journal of Optimization Theory and Applications 152 (2) (2012) 271–306. doi:10.1007/s10957-011-9918-z.

[139] A. V. Rao, A Survey of Numerical Methods for Optimal Control, 2009 AAS/AIAA Astrodynamics Specialist Conference, AAS paper 09-334 (2009). doi:10.1515/jnum-2014-0003.

[140] K. Alemany, R. D. Braun, Survey of Global Optimization Methods for Low-Thrust, Multiple Asteroid Tour Missions, 17th AAS/AIAA Space Flight Mechanics Meeting, AAS 07-211 (2007) 1–20.

[141] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, E. F. Mishchenko, The Mathematical Theory of Optimal Processes, John Wiley & Sons., New York, NY, USA, 1963.

[142] D. Garg, M. Patterson, W. W. Hager, A. V. Rao, D. A. Benson, G. T. Huntington,

A unified framework for the numerical solution of optimal control problems using pseudospectral methods, Automatica 46 (11) (2010) 1843–1851. doi:10.1016/j.automatica.2010.06.048.

[143] D. Garg, M. A. Patterson, C. Francolin, C. L. Darby, G. T. Huntington, W. W. Hager, A. 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) (2011) 335–358. doi:10.1007/s10589-009-9291-0.

[144] D. Wu, F. Jiang, J. Li, Warm Start for Low-Thrust Trajectory Optimization via Switched System, Journal of Guidance, Control, and Dynamics 44 (9) (2021). doi:10.2514/1.G005983.

[145] M. Rosa Sentinella, L. Casalino, Hybrid Evolutionary Algorithm for the Optimization of Interplanetary Trajectories, AIAA Journal of Spacecraft and Rockets 46 (2) (2009) 365–372. doi:10.2514/1.38440.

[146] B. Dachwald, A. Ohndorf, 1st ACT global trajectory optimisation competition: Results found at DLR, Acta Astronautica 61 (9) (2007) 742–752. doi:10.1016/j.actaastro.2007.03.011.

[147] R. H. Battin, An Introduction to the Mathematics and Methods of Astrodynamics, AIAA Education Series, Reston, VA, USA, 1999. doi:10.2514/4.861543.

[148] M. Sagliano, S. Theil, Hybrid Jacobian Computation for Fast Optimal Trajectories Generation, AIAA Guidance, Navigation, and Control (GNC) Conferences (2013). doi:10.2514/6.2013-4554.

[149] G. Huntington, Advancement and Analysis of a Gauss Pseudospectral Transcription for Optimal Control Problems, PhD Thesis, Department of Aeronautics and Astronautics, Massachusetts Insitute of Technology, Cambridge, MA, USA (2007).

[150] D. Garg, Advances In Global Pseudospectral Methods For Optimal Control, Ph.D. thesis, University of Florida (2011).

[151] P. Davis, Interpolation and Approximation, Dover Publications, 1975.

[152] R. Fletcher, S. Leyffer, P. e. a. Toint, A brief history of filter methods, Preprint ANL/MCS-P1372-0906, Argonne National Laboratory, Mathematics and Computer Science Division (2006) 36.

[153] P. E. Gill, W. Murray, M. A. Saunders, SNOPT: An SQP Algorithm for Large–Scale Constrained Optimization, SIAM Review 47 (4) (2005) 99–131. arXiv:17444372724, doi:10.1137/S1052623499350013.

[154] A. Wachter, L. T. Biegler, On the Implementation of an Interior-Point Filter Line-search Algorithm for Large-scale Non-linear Programming, Mathematical Programming 106 (1) (2006) 25–57. doi:10.1007/s10107-004-0559-y.

[155] C. Büskens, D. Wassel, The ESA NLP Solver WORHP, Modeling and Optimization in Space Engineering, Springer New York 73 (2013) 85–110. doi:10.1007/978-1-4614-4469-5_4.

[156] A. Waechter, C. Laird, F. Margot, Y. Kawajir, Introduction to IPOPT: A tutorial for downloading, installing, and using IPOPT, Revision (2009).

[157] M. A. Patterson, W. W. Hager, A. V. Rao, A ph Mesh Refinement Method for Optimal Control, Optimal Control Applications and Methods 36 (4) (2014) 398–421. doi:10.1002/oca.2114.

[158] www.gpops2.com, accessed: 2022-01-12.

[159] C. L. Darby, W. W. Hager, A. V. Rao, An hp-adaptive pseudospectral method for solving optimal control problems,, Optimal Control Applications and Methods 32 (4) (2011) 476–502.

[160] F. De Groote, A. L. Kinney, A. V. Rao, B. J. Fregly, Evaluation of Direct Collocation Optimal Control Problem Formulations for Solving the Muscle Redundancy Problem, Annals of Biomedical Engineering 44 (10) (2016) 2922–2936. doi:10.1007/s10439-016-1591-9.

[161] M. J. Weinstein, A. V. Rao, Algorithm 984: ADiGator, a toolbox for the algorithmic differentiation of mathematical functions in MATLAB using source trans- formation via operator overloading, ACM Transactions on Mathematical Software (TOMS) 44 (2) (2017) 21.

[162] M. A. Patterson, A. V. Rao, GPOPS-II manual: A General-Purpose MATLAB Software for Solving Multiple-Phase Optimal Control Problems (Version 2.3), Tech. rep., University of Florida, Gainesville, FL, USA (2016).

[163] A. V. Rao, D. A. Benson, C. L. Darby, M. A. Patterson, C. Francolin, I. Sanders, G. 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). arXiv:1005.3014, doi:10.1145/1731022.1731032.

[164] M. A. Patterson, 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). doi:10.1145/2558904.

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

[166] B. J. Wall, B. A. Conway, Shape-Based Approach to Low-Thrust Rendezvous Trajectory Design, Journal of Guidance, Control, and Dynamics 32 (1) (2009). doi:10.2514/1.36848.

[167] S. N. Sivanandam, S. N. Deepa, Introduction to Genetic Algorithms, Springer-Verlag (2008) 15–37.

[168] M. Pontani, B. A. Conway, Particle Swarm Optimization Applied to Space Trajectories, Journal of Guidance, Control, and Dynamics 33 (5) (2010) 1429–1441. doi:10.2514/1.48475.

[169] M. Pontani, B. A. Conway, Optimal Low-Thrust Orbital Maneuvers via Indirect Swarming Method, Journal of Optimal Theory and Applications 162 (1) (2014) 272–292. doi:10.1007/s10957-013-0471-9.

[170] R. Storn, K. Price, Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces, Journal of Global Optimization 11 (4) (1997) 341–359. doi:10.1023/a:1008202821328.

[171] M. Dorigo, L. M. Gambardella, Ant colony system: a cooperative learning approach to the traveling salesman problem, IEEE Transactions on Evolutionary Computation 1 (1) (1997) 53–66. doi:10.1109/4235.585892.

[172] S. Kirkpatrick, C. D. J. Gelatt, M. P. Vecchi, Optimization by Simulated Annealing, Science 220 (4598) (1983) 671–680. doi:10.1126/science.220.4598.671.

[173] F. W. Glover, M. Laguna, Tabu Search, Springer, New York, NY, USA, 1997.

[174] M. Vasile, E. Minisci, M. Locatelli, On Testing Global Optimization Algorithms for Space Trajectory Design, AIAA/AAS Astrodynamics Specialist Conference and Exhibit, AIAA Paper 2008-6277 (2008). doi:10.1126/science.220.4598.671.

[175] M. Pontani, Particle swarm optimization of ascent trajectories of multistage launch vehicles, Acta Astronautica 94 (2) (2014) 852–864. doi:10.1016/j.actaastro.2013.09.013.

[176] J. A. Englander, M. A. Vavrina, D. Hinckley, Global Optimization of Low- Thrust Interplanetary Trajectories Subject to Operational Constraints, 26th AAS/AIAA Space Flight Mechanics Meeting, AAS Paper 16-239 (2016).

[177] D. Izzo, PyGMO and PyKEP: Open Source Tools for Massively Parallel Optimization in Astrodynamics (the case of interplanetary trajectory optimization), 5th

International Conference on Astrodynamics Tools and Techniques (2012).

[178] N. Sullo, P. A. Sousa-Silva, M. O. Terra, M. Ceriotti, Optimisation of Low- Thrust and Hybrid Earth-Moon Transfers, 67th International Astronautical Congress, IAC-16-C1.4.5 (2013).

[179] K. Deb, An introduction to genetic algorithms, Sadhana 24 (4) (1999) 293–315.

[180] https://uk.mathworks.com/help/gads/how-the-genetic-algorithm-works.html, accessed: 2022-01-13.

[181] https://uk.mathworks.com/help/gads/ga.html, accessed: 2022-01-13.

[182] K. Deb, Multi-objective optimization using evolutionary algorithms, Wiley, Chichester (2001).

[183] K. Miettinen, Nonlinear Multiobjective Optimization, Kluwer Academic Publishers, 1999.

[184] K. Deb, Multi-objective genetic algorithms: Problem difficulties and construction of test problems, Evolutionary Computation Journal 7 (3) (1999) 205–230.

[185] H. Kung, F. Luccio, F. Preparata, On finding the maxima of a set of vectors, Journal of the Association for Computing Machinery 22 (4) (1975) 469–476. doi:10.1126/science.220.4598.671.

[186] J. Branke, K. Deb, K. Miettinen, R. Slowinski, Multiobjective Optimization, Springer Heidelberg, 2008.

[187] A. Brown, Nerve Cells and Nervous Systems, Springer-Verlag), 1991.

[188] H. Siegelmann, E. Sontag, Turing computability with neural nets, Applied Mathematics Letters 4 (6) (1991) 77–80.

[189] A. Graves, A. Mohamed, G. Hinton, Speech recognition with deep recurrent neural networks, ICASSP'2013 (2013) 6645–6649.

[190] H. Hyotyniemi, Turing machines are recurrent neural networks, STeP'96 (1996) 13–24.

[191] M. Caudill, C. Butler, Naturally Intelligent Systems, MIT Press, Cambridge, London, 1990.

[192] J. Buhmann, H. Kuhnel, Unsupervised and supervised data clustering with competitive neural networks, IJCNN International Joint Conference on Neural Networks, IEEE 4 (1992) 796–801. doi:10.1109/ijcnn.1992.227220.

[193] https://uk.mathworks.com/help/deeplearning/ug/neural-network-

training-concepts.html, accessed: 2022-02-23.

[194] C. Bishop, Neural networks for pattern recognition, Clarendon Press, Oxford, 1995. doi:10.1016/S0893-6080(05)80056-5.

[195] M. Moller, A scaled conjugate gradient algorithm for fast supervised learning, Neural Networks, Elsevier 6 (4) (1993) 525–533. doi:10.1016/S0893-6080(05)80056-5.

[196] M. Riedmiller, H. Braun, A direct adaptive method for faster backpropagation learning: The RPROP algorithm, Proceedings of the IEEE International Conference on Neural Networks (1993) 586–591doi:10.1109/ICNN.1993.298623.

[197] K. Levenberg, A method for the solution of certain non-linear problems in least squares, Quarterly Journal of Applied Mathmatics 2 (2) (1944) 164–168.

[198] D. Marquardt, An algorithm for least-squares estimation of nonlinear parameters, Society for Industrial & Applied Mathematics 11 (2) (1963) 431–441.

[199] S. Amari, N. Murata, K. Muller, M. Finke, H. Yang, Asymptotic statistical theory of overtraining and cross-validation, IEEE Transactions on Neural Networks 8 (5) (1997) 985–996.

[200] R. Neal, Bayesian Learning for Neural Networks, Springer, New York, 1996.

[201] A. Coates, H. Lee, A. Y. Ng, An analysis of single-layer networks in unsupervised feature learning, Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2011) (2011).

[202] I. Ross, R. Proulx, M. Karpenko, An Optimal Control Theory for the Traveling Salesman Problem and Its Variants (2020). arXiv:2005.03186.

[203] T. H. Cormen, C. Leiserson, R. L. Rivest, C. Stein, Introduction to Algorithms, Third Edition, The MIT Press, 2009.

[204] http://www.gla.ac.uk/services/it/hpcc/, accessed: 2022-03-04.

[205] W. Wiesel, Spaceflight Dynamics: Third Edition, CreateSpace, Cambridge, Massachusetts, 2017.

[206] H. Curtis, Orbital Mechanics for Engineering Students, Elsevier, 2005.

[207] C. Murray, S. F. Dermott, Solar System Dynamics, Cambridge University Press, New York, 2000.

[208] V. Kůrková, Kolmogorov's theorem and multilayer neural networks, Neural Networks 5 (3) (1992) 501–506. doi:10.1016/0893-6080(92)90012-8.

[209] H. Shen, T. Zhang, L. Casalino, D. Pastrone, Optimization of Active Debris Removal Missions with Multiple Targets, Journal of Spacecraft and Rockets (2017). doi:10.2514/1.A33883.

[210] H. Shen, L. Casalino, Simple ΔV Approximation for Optimization of Debris-to-Debris Transfers, ArXiv (2020). doi:10.48550/arXiv.2004.02225.

[211] www.starlink.com/satellites, accessed: 2022-03-09.

[212] https://nssdc.gsfc.nasa.gov/nmc/spacecraft/display.action?id=2019-074A, accessed: 2022-03-09.

[213] https://directory.eoportal.org/web/eoportal/satellite-missions/o/oneweb, accessed: 2022-03-09.

[214] https://ssd.jpl.nasa.gov/sbdb_query.cgi#x, accessed: 2022-04-08.

[215] B. W. Barbee, T. Esposito, E. Pinon, S. Hur-Diaz, R. G. Mink, 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).

[216] R. C. Boden, A. M. Hein, J. Kawaguchi, Target selection and mass estimation for manned NEO exploration using a baseline mission design, Acta Astronautica 111 (2015) 198–221. doi:10.1016/j.actaastro.2015.02.018.

[217] https://cneos.jpl.nasa.gov/nhats/, accessed: 2021-01-07.

[218] https://cneos.jpl.nasa.gov/nhats/intro.html, accessed: 2022-03-14.

[219] A. Peloni, B. Dachwald, M. Ceriotti, Multiple near-earth asteroid rendezvous mission: Solar-sailing options, Advances in Space Research 62 (2017) 2084–2098. doi:10.1016/j.asr.2017.10.017.

[220] A. Peloni, Solar-sail Mission Design For Multiple Near-Earth Asteroid Rendezvous, Phd thesis, University of Glasgow, Glasgow, UK (2018).

[221] H. Shen, P. Tsiotras, Using Battin's method to obtain multiple-revolution Lambert's solutions, Proceedings of AAS/AIAA Astrodynamics Specialist Conference, Big Sky, Montana, USA (2003).

[222] D. Isbell, F. O'Donnell, J. Watson, Deep Space 1 Asteroid Flyby, NASA, Press Kit (1999).

[223] https://www.cpubenchmark.net/compare/Intel-i7-3770-vs-AMD-Opteron-6376/896vs2000, accessed: 2020-06-03.

[224] F. Vilas, Spectral characteristics of Hayabusa 2 near-Earth asteroid targets

162173 1999 JU3 AND 2001 QC34, The Astronomical Journal 135 (2008) 1101–1105. doi:10.1088/0004-6256/135/4/1101.

[225] D. J. Tholen, M. A. Barucci, Asteroid taxonomy, Asteroids II, Proceedings of the Conference, March (1989) 298–315.

[226] A. W. Harris, A. W. Harris, On the Revision of Radiometric Albedos and Diameters of Asteroids, Icarus 126 (2) (1997) 450–454. doi:10.1006/icar.1996.5664.

[227] S. R. Chesley, P. W. Chodas, A. Milani, G. B. Valsecchi, D. K. Yeomans, Quantifying the Risk Posed by Potential Earth Impacts, Icarus 159 (2) (2002) 423–432. doi:10.1006/icar.2002.6910.

[228] https://cneos.jpl.nasa.gov/tools/ast_size_est.html, accessed: 2020-07-14.

[229] M. Ceriotti, C. McInnes, Systems design of a hybrid sail pole-sitter, Advances in Space Research 48 (11) (2011) 1754–1762. doi:10.1016/j.asr.2011.02.010.

[230] R. Gershman, C. Seybold, Propulsion trades for space science missions, Acta Astronautica 45 (4-9) (1999) 541–548.

[231] S. Kitamura, Y. Ohkawa, Y. Hayakawa, H. Yoshida, K. Miyazaki, Overview and Research Status of the JAXA 150-mN Ion Engine, Acta Astronautica 61 (1-6) (2007) 360–366. doi:10.1016/j.actaastro.2007.01.010.