Politecnico di Milano
School of Industrial and Information Engineering
Department of Aerospace Science and Technology
Master of Science in Space Engineering

Deep Reinforcement Learning for Smart Small Bodies Mapping during proximity Operations

Advisor:
Prof. Michèle Lavagna
Co-Advisor:
Dr. Vincenzo Pesce

Graduation Thesis of:
Margherita Piccinin
874936

Academic Year 2017-2018

Abstract

Small bodies mapping is a crucial but challenging capability for space exploration missions. Current missions heavily rely on human intervention not only for on ground map refinement, but also for operations supervision and planning. In fact, the extreme variety of body shapes, dynamical environments and illumination conditions makes spacecraft autonomous mapping a complex task in relation to the limited computational resources available on-board. This thesis develops a method to autonomously plan the timing of observations during the mapping of an unknown small body, with particular application to imaging for stereophotoclinometry. The goal is to define a policy that improves mapping quality, while both limiting the amount of images to downlink and fastening the mapping process. The planning framework is defined as a Partially Observable Markov Decision Process (POMDP), proposing a novel problem architecture focused on data collection. Deep Reinforcement Learning (DRL) is exploited to design the planning policies, comparing two different techniques: Neural Fitted Q (NFQ) and Deep Q Network (DQN). The obtained policies are extensively tested over a wide range of different possible scenarios in order to verify their generalizing capability, which is of great importance when exploring an unknown environment. Results show that the proposed solutions are capable to deal with far-off different scenarios and outperform simple benchmarks. Then, a computational analysis is addressed to determine feasibility and limits of a possible on-board implementation of the algorithm. The proposed methodology reveals to be a promising step forward in autonomous operations, helping in decreasing the human effort during unknown small bodies mapping and increasing imaging exploitation efficiency with a simple and flexible approach.

Keywords: Autonomous Exploration, Small Bodies Mapping, Reinforcement Learning for Space Applications, Planning, Active Sensing

Sommario

LA mappatura di piccoli corpi celesti sconosciuti è una fase cruciale ed ardua per le missioni spaziali esplorative, attualmente resa possibile grazie ad un significativo intervento umano nella realizzazione della mappa e nella supervisione e pianificazione delle operazioni. Infatti, l'estrema varietà di tali oggetti celesti in quanto a forma, ambiente dinamico ed illuminazione, rende la mappatura un compito complesso in relazione alle limitate risorse computazionali disponibili a bordo. Questa tesi sviluppa un metodo per pianificare autonomamente la tempistica delle osservazioni durante la mappatura di un piccolo corpo celeste ignoto, in vista della successiva elaborazione delle immagini a terra attraverso stereo-fotoclinometria. L'obiettivo è definire una politica che velocizzi e migliori la qualità della mappatura, limitando la quantità di dati da trasmettere. La pianificazione è formulata come processo decisionale di Markov parzialmente osservabile (POMDP), proponendo un'architettura innovativa focalizzata sulla raccolta delle immagini e che adotta l'apprendimento per rinforzo come metodo di soluzione. Si confrontano le tecniche Neural Fitted Q (NFQ) e Deep Q Network (DQN), testando le politiche di pianificazione così ottenute su una vasta gamma di scenari e verificandone la generalità. I risultati mostrano che entrambe le soluzioni hanno prestazioni superiori a semplici benchmarks, si adattano a scenari alquanto differenti e migliorano l'efficienza della mappatura. Fattibilità e limiti di una possibile implementazione dell'algoritmo a bordo sono indagati con un'analisi computazionale. La metodologia proposta si rivela un promettente passo avanti verso le operazioni autonome, aiutando a diminuire lo sforzo umano durante la mappatura di piccoli corpi celesti ignoti ed aumentando l'efficienza nella raccolta delle immagini con un approccio semplice e flessibile.

Parole chiave: Esplorazione Autonoma, Mappatura di Piccoli Corpi Celesti Ignoti, Apprendimento per Rinforzo in Applicazioni Spaziali, Pianificazione, Percezione Attiva

Estratto in lingua italiana

Negli ultimi anni, frequenti missioni esplorative verso i piccoli oggetti celesti si pongono obiettivi sempre più ambiziosi, che richiedono una continua innovazione tecnologica. Queste missioni sono accomunate da operazioni volte a determinare le caratteristiche dell'oggetto celeste ignote prima della missione stessa. Ad oggi un significativo intervento umano è necessario per la buona riuscita e pianificazione di tali operazioni. Un aumento dell'autonomia del veicolo è fondamentale per una maggiore efficienza delle missioni future.

Questo lavoro di tesi mira ad aumentare l'autonomia dei veicoli spaziali durante le fasi di mappatura di piccoli oggetti celesti. In questa tesi si usano tecniche di Deep Reinforcement Learning (DRL) per ottenere una politica di pianificazione con obiettivi multipli ed in parte contrastanti: da un lato la limitazione dei dati raccolti, dall'altro una mappatura rapida, globale e con caratteristiche che agevolino la ricostruzione topografica del corpo. Tali obiettivi sono raggiunti selezionando opportunamente gli istanti di osservazione del corpo durante la sua mappatura.

Operazioni di Mappatura

Un aspetto molto importante ed oneroso delle operazioni di mappatura è la raccolta di immagini ai fini della derivazione della topografia del corpo. Una delle più comuni tecniche utilizzate per la ricostruzione della forma di un oggetto celeste è la stereofotoclinometria (SPC). Questa procedura è svolta a terra utilizzando immagini della superficie del corpo e dati di navigazione acquisiti nel corso della missione. Il complesso algoritmo di SPC richiede svariati cicli iterativi per ottenere un modello ad alta risoluzione del corpo e una mappa di albedo. Tali prodotti, oltre ad essere di interesse scientifico, sono impiegati per l'identificazione di punti di riferimento sulla superficie, detti landmarks, utilizzati ai fini della navigazione relativa tra veicolo e superficie. Per facilitare la convergenza dell'algoritmo servono da un lato accurati dati di navigazione e dall'altro immagini che presentino caratteristiche favorevoli, ovvero una grande varietà delle condizioni di illuminazione e angolazioni rispetto alla superficie tra loro simili. SPC richiede una notevole quantità di dati da trasmettere a terra: un'ottimizzazione della raccolta di immagini durante la mappatura agevolerebbe SPC e renderebbe le operazioni più efficienti.

Pianificazione autonoma

Nell'ambito della robotica il problema dell'esplorazione autonoma di un ambiente ignoto e' conosciuto come active SLAM (mappatura e localizzazione simultanee attive). In campo spaziale il problema della mappatura e navigazione attorno a piccoli corpi è stato recentemente formulato come processo decisionale di Markov parzialmente osservabile (POMDP) in analogia con la robotica terrestre. In questo contesto il veicolo spaziale è visto come un agente che si muove in un ambiente sconosciuto, mappando il corpo celeste ed allo stesso tempo navigando utilizzando i sensori di bordo ed una mappa della superficie. L'agente può prendere decisioni che influenzano le proprie percezioni dell'ambiente, ad esempio controllando la propria traiettoria.

Tuttavia il POMDP è non risolvibile nella sua forma più generale e necessita di una formulazione ridotta. In questa tesi si propone un'architettura di planning ridotta innovativa, focalizzata sulla pianificazione dei tempi di osservazione, ottimizzando la raccolta di immagini per SPC. Nel proporre tale architettura si presta particolare attenzione alle difficoltà che l'ambiente spaziale comporta: le limitate risorse di bordo in termini di potenza computazionale e memoria, la grande influenza della dinamica naturale sul moto del veicolo, con rischi legati ad un ambiente dinamico variegato e caotico. I meriti principali di questa architettura sono la possibilità di integrare la pianificazione senza rischi per la missione ed una grande flessibilità che rende la pianificazione efficacie in diversi scenari ed indipendente dalle specificità della missione.

Apprendimento per rinforzo

Il DRL è scelto come tecnica di soluzione del POMDP ridotto, poichè permette di affrontare processi decisionali complessi anche parzialmente osservabili, con spazi degli stati ampi e continui. L'utilizzo di una rete neurale per l'approssimazione del valore della ricompensa sul lungo periodo, permette di ottenere una policy in grado di generalizzare in caso di situazioni nuove, senza la necessità di un'esplorazione completa dello spazio degli stati, impossibile quando essi sono infiniti come nell'architettura di pianificazione qui proposta.

Due tecniche di DRL sono confrontate: NFQ e DQN. Le soluzioni al processo decisionale ottenute attraverso DRL sono sub-ottimali e pertanto il confronto di due tecniche differenti è indice della bontà dei risultati. Le due policy ottenute sono inoltre validate confrontando i risultati con due benchmark: una strategia uniforme, che scatta le immagini ad intervalli regolari, ed una strategia random. I risultati mostrano un sostanziale miglioramento della mappatura in relazione agli obiettivi preposti. Un'estesa analisi di sensitività conferma inoltre la capacità di pianificare la mappatura in scenari disparati, che comprendono diverse forme dei corpi, condizioni di illuminazione e dinamica relativa. Tali verifiche rivestono una particolare importanza poichè vengono utilizzate tecniche di DRL, dove il successo dell'apprendimento è particolarmente sensibile alla taratura di alcuni iperparametri dell'algoritmo. Per un apprendimento corretto è inoltre fondamentale una corretta formulazione della ricompensa: la policy potrebbe avere comportamenti indesiderati nel caso in cui la ricompensa non fosse adeguata. Per questo motivo la performance delle strategie è valutata attraverso parametri che rendano evidente quali sono le azioni eseguite dall'agente e le loro conseguenze sulla mappatura.

Infine, il tempo computazionale impiegato per prendere le decisioni è sondato con alcune analisi preliminari. L'architettura di planning è effettivamente leggera entro certi limiti, definiti tramite le analisi.

Acknowledgments

First of all, I would like to thank my parents, Mariapia and Cristiano, for their constant support and unconditional love. They have guided me through the difficulties of life and equipped me for my *space travel* in the best way possible: listening to me, advising me, valuing my opinions and giving me freedom and trust. Thanks for being always interested in my path and supporting my decisions. Also thanks to my beloved brother Pier, always generous with me, and to all my family.

Special thanks go to my advisor, Prof. Michèle Lavagna, who has given me the opportunity to afford new challenges, increasing my thirst of knowledge and enthusiasm for the future. Looking back to just one year ago, I can see how much I've learned from new and unexpected experiences. During this thesis work, she has always aimed high for me, offering support but also letting me improve my independence of thought. I would like also to thank the members of her team, who have contributed to my professional growth, sharing their knowledges and being an example to me. In particular, thanks to Vincenzo for closely advising me during this thesis work, being always helpful, positive and kind.

Then, thanks to my dear and long-time friends, Eli, Marta and Simo, for our strong bond and reciprocal understanding: your only presence is capable of making me happy. Thanks to Gaia, Giada, Giulia, Marco, Andrea and Riccardo for all the moments enjoyed together and the stimulating exchange of opinions, and to Giuli, always present since so many years. Finally, thanks to all the friends that have shared with me these years at Politecnico. In particular, I would like to thank Laura, for her special and precious friendship, and the more recent friends Francesca and Marzia, for their warmth and empathy. Thanks to Quiro, Zio, Permu, Molgo, Luca, Seba, Gare, Faggio and Azzi for the days spent side by side, that have flown so quickly, full of joy and laughters. With you all I've known a lightheartedness I could never imagine.

Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms

A	
ANH3BP	Augmented Normalized Hill Three Body Problem
ANN	Artificial Neural Network
C	
CCD	Charged Coupled Device
D	
DRL	Deep Reinforcement Learning
DQN	Deep Q Network
E	
EH	Ellipsoidal Harmonics
F	
FOV	Field Of View
G	
GTM	Global Topography Model
M	
MDP	Markov Decision Process
MLP	Multilayer Perceptron
N	
NFQ	Neural Fitted Q
P	
POMDP	Partially Observable Markov Decision Process
R	
RAAN	Right Ascension of Ascending Node

S	
SH	Spherical Harmonics
SLAM	Simultaneous Localization And Mapping
SPC	Stereophotoclinometry
SPG	Stereophotogrammetry
SRP	Solar Radiation Pressure
R	
RL	Reinforcement Learning
RPROP	Resilient Back-Propagation

1 Introduction

In the last decades space agencies have shown a deep interest in the exploration of unknown Solar System objects, like asteroids and comets. Such bodies are less evolved with respect to larger ones, as planets or moons, and for this reason they contain important clues to understand the Solar System origin and evolution. Exploration missions towards small bodies present several challenging aspects to mission design, including limited telecommunication, complex orbital dynamics and lack of knowledge of some key aspects of the exploration environment. In fact, many characteristics of the celestial body can not be fully assessed before the mission, like density, rotational state, gravitational field, thermo-physical properties and topology. Their determination becomes an essential operation, common to all small bodies missions. In particular, the body shape reconstruction requires many iterations and refinements that are performed on-ground only once the mission is in progress, thanks to the collected data. Small bodies missions often involve very complex objectives, such as landing or touching the object, for which it is necessary to have a complete and detailed knowledge of all the above mentioned characteristics.

Guidelines for future missions strongly stress the necessity to enhance spacecraft autonomy. Autonomous systems are desirable not only to reduce reliance on ground-support personnel, but also to enable complex functions that would not otherwise be possible. This is especially true for missions in which communication with ground can not always take place or constitutes a bottleneck. Exploration and mapping of small bodies currently requires a large employment of human resources for many different aspects, like trajectory design, navigation, data collection and mapping operations supervision and planning. An autonomous exploration is desirable for future missions, but considering the high correlation between all of these aspects, improving autonomy is a big challenge and the process towards full spacecraft autonomy is still at an early stage. Autonomous exploration methods are being developed in the robotics field, where the environment is in general less challenging.

The focus of this thesis is on on-board autonomous decision-making. This interesting topic presents as main challenges the limited computational resources and data storage available on-board, but offers the possibility to improve mission operations. In particular, this thesis develops an autonomous planning algorithm for the timing of observations while imaging an unknown small body. More specifically, the aim is to plan the images collection to benefit stereophotoclinometry, the most common mapping technique adopted in past missions for on-ground shape model reconstruction.

The methodology developed in this thesis can be easily adapted for other imaging techniques, like stereophotogrammetry. With the appropriate considerations, the presented methodology can also be tailored on different instruments, as thermal cameras or radars. Moreover, the applicability of the proposed approach is not limited to small bodies missions, but can be extended in different contexts where autonomous data collection is helpful.

In this chapter the thesis motivations are exposed and a literature review is presented. Then, the work intended contributions and thesis overview are outlined.

1.1 Motivation

1.1.1 Exploration of small bodies

Numerous missions devoted to study small bodies took place in the past two decades. NASA mission NEAR Shoemaker performed rendezvous with NEAR asteroids Mathilde and Eros, closing observing and characterizing in depth the latter [1], while Dawn orbited around Vesta in the main asteroid belt [2]. ESA mission Rosetta was the first mission to reach and land on a comet, 67P Churiumov-Gerasimenko [3]. More recent missions are the sample-return Hayabusa 2 to Ryugu [4] by JAXA and NASA OSIRIS-REx to Bennu [5], which is still ongoing. Some future missions have been proposed and are under plan, like NASA NEAScout, that exploits a deep-space CubeSat [6], and ESA Hera, towards the asteroid binary systems Didymos and again using CubeSats [7]. All these missions make evident the strong and long-term interest of space agencies in small bodies exploration missions. The increasing complexity of mission scenarios and objectives entails a continuous technological evolution and improvement.

1.1.2 Mapping and shape model reconstruction

Mapping is an operational phase common to all small bodies missions, despite their different objectives. Such phase is particularly demanding, as it takes a long time and requires large amounts of data to be sent on-ground, where they are analyzed and elaborated by human experts. For instance, during Rosetta operations the total number of images taken for the whole mission duration was about 100000 [8]. Moreover, the downlink of scientific data is limited and requires considerable time and economic resources: communications with far bodies are not always possible and are limited by ground stations availability.

Nevertheless, collecting data with multiple instruments like radar, optical and thermal cameras, multispectral instruments is fundamental for environment exploration and small bodies study. In particular, an important mapping operation is imaging for high resolution shape model reconstruction. The reconstruction of the body shape model has many purposes. In first place, it is of scientific interest and useful for proximity operations. In fact, it is necessary for evaluating landing sites hazards and conditions, for which the desirable map resolution has to be lower than the lander footprint. In second place, it constitutes a valid instrument for assisting in mission data elaboration. For instance, in past missions [9] it has been used to support landmark-based navigation: simulating synthetic images along with an albedo map, the landmark appearance is predicted, based on coarse camera pose and Sun direction informations. Then, the expected appearance is cross-correlated with the real image and a precise pose estimate is possible. The shape model can also improve algorithms for landmark tracking and image matching during night excursions by limb fitting. Or it can provide a cross-check with orbit determination output for the estimation of gravity model.

1.1.3 Autonomy improvement

To successfully perform the mapping process requires a great human effort, starting from planning and design of suited trajectories, then supervising navigation and mapping operations, reconstructing the shape model on ground with increasing resolution. Human support has revealed to be crucial in past missions. For instance, during Rosetta mission the initial landmark identification on images was performed manually [9]. Also the planning of mapping phases was performed by human experts. In order to build a coarse shape model with stereophotogrammetry, after requirements identification the images acquisition was planned to be on pyramidal lag orbits at 60-90 km from the surface. The number of images to be acquired was also planned in advance, selecting an image separation of roughly one hour, in order to provide a sufficiently large stereo angle between two subsequent images [10].

State of the art operations can be largely improved in the autonomy direction, challenging under many aspects, above all the harshness of the environment and complexity of requirements, leading to a greater mapping efficiency and cost effectiveness and reducing the employment of human resources.

1.1.4 Related research fields

Today, small bodies exploration continuously requires humans-in-the-loop. This is not true in other fields, were robots are capable to autonomously accomplish their tasks while exploring and mapping an unknown environment. Some examples are indoor and outdoor robots like underwater vehicles, autonomous cars or domestic robots. For what concerns space applications, a similar context can be found in planetary exploration: rovers need to perform several tasks like localization, trajectory control and path planning considering terrain traversability analysis and obstacle avoidance. Since such systems would benefit from autonomous environment exploration, current research tries to reduce human involvement in the navigational loop [11], [12].

Similar issues are encountered while mapping a small body: a spacecraft observes the environment through sensors, whose observations are used for navigation and localization purposes, while moving around the body. An autonomous spacecraft should be able to perform all necessary tasks for mapping the body, including planning the exploration. Analogies with the robotics field suggest to adopt similar approaches also for the small bodies mapping problem, but tailoring them on its peculiarities. Tools and methods to enhance spacecraft autonomy need to be carefully chosen, accounting for the very low computational power available on-board and limiting risks for mission safety. In particular, machine learning offers interesting possibilities for autonomy enhancement. A successful application of supervised and unsupervised learning is autonomous driving, where the challenges are again obstacles classification, localization, mapping and planning [13], [14]. In the space field, some research is investigating the possibilities of unsupervised learning for rovers autonomous prediction of slippage events [14] or of supervised learning for autonomous landing and hazard detection [15]. In particular, the generalizing capabilities of neural networks and their light computational architecture can be a powerful tool when dealing with a not completely known scenario. However one has to be aware that their black-box nature implies some risks.

1.2 Literature review and previous works

1.2.1 Small bodies mapping

In past years techniques and procedures for small bodies shape reconstruction have been developed and tested. Two main branches exist in shape models 3D reconstruction from images: shape from motion and shape from shading. Techniques for small bodies shape model reconstruction descend from both methods, that have a long history in computer vision [16]. Stereophotogrammetry (SPG) is derived from shape from motion and first used by Giese *et al.* in various space applications [17], [18]. SPG exploits triangulation of feature points matched in different images to reconstruct the body shape. While stereophotoclinometry (SPC) descends from shape from shading and is based on a photometric model that allows to derive slopes from pixels intensities. This method was first developed by Gaskell [19] and has been refined in successive years, being successfully employed in several missions. Variants and combinations of SPG and SPC have been also used [9]. In [20] it is developed a multiresolution photometric method that builds the shape model from progressive mesh deformation while increasing resolution.

Some research has defined criteria useful for evaluating the goodness of body mapping, based on engineering judgment. In [10] illumination and observation SPG requirements for imaging planning are provided. In [21] a method to quantify the mapping performance for SPC is proposed, comparing different orbits around small bodies and assuming a spherical shape as case study. Another research effort is towards design of trajectories that ease the body mapping, offering adequate coverage and suited illumination conditions [22].

1.2.2 Autonomous exploration

Many works are present in the robotics field concerning autonomous exploration of an unknown environment. In particular, such works deal with Simultaneous Localization and Mapping (SLAM) techniques, that have been first developed in the 80' [23]. SLAM consists in estimating a map of an unknown environment and at the same time localizing in it. The aim is to build a global and consistent map of the environment with no a priori information and taking as inputs just visual informations from a monocamera: features can be extracted and matched between different frames and the camera pose can be retrieved. A triangulation of the features is then performed through the collinearity equation, i.e. through projective geometry, to reconstruct the 3D map of the environment [24].

At the end of the 90', exploration planning has been added as third task for the robot, giving rise to active SLAM. In this framework, mapping of the environment, localization and planning of the exploration are strictly coupled. The active SLAM approach has been first proposed and tested by Feder *et al.* [25], who considered the problem of adaptive motion and sensing for feature-based mapping and localization by autonomous underwater vehicles. A major challenge in robotic mapping is to automate the data collection process, in order to build the highest quality map with the least time and cost. Successive works have proposed different formulations in terms of performance criteria and optimization procedures [26], [27]. Typical criteria try to optimize the information gain, improving map accuracy and limiting the costs.

Literature concerning small bodies autonomous mapping is really exiguous. In [28], autonomous mapping and navigation are studied, proposing to use a Structure From Motion algorithm along with a real-time optimization to retrieve the body rotation and center of mass trajectory, assuming a complete knowledge of spacecraft pose. Other works propose to use SLAM for autonomous navigation [29], [30]. These works try to overcome the reliance on ground-in-the-loop operations, but the proposed algorithms are known to have a high computational burden.

Pesce *et al.* [31] have recently driven knowledges from robotics to space field, proposing to model navigation and mapping about small bodies as a Partially Observable Markov Decision Process (POMDP). Their work is the first to present a general formulation of the active SLAM problem applied to small bodies exploration. In particular, a planning strategy for orbit selection is proposed, with a simplified solution that considers a fully observable environment.

1.2.3 Deep Reinforcement Learning

Reinforcement Learning (RL) has been widely employed for policy design in planning problems, comprising POMDP [32].

For what concerns the robotics field, Kollar *et al.* in [34] and [33] have proposed to use RL to optimize trajectories for map exploration. Also, Deep Reinforcement Learning (DRL) allows to design planning policies in partially observable scenarios. Such tool is extremely useful for tackling complex planning problems with large and continuous state spaces, that can be handled thanks to neural networks generalizing capabilities. DRL has been recently proposed for end-end driving [35], using recurrent networks that exploit time information, essential for tackling the POMDP.

It has been recently proposed to use DRL for imaging and mapping small bodies by Chan *et al.* [36]. Their work is based on the framework defined in [31] and explores the POMDP solution more deeply, designing a neural network policy to both maneuver the spacecraft and plan the images collection. However, the introduction of maneuvering somehow limits the application of their work: as the authors declare, trusting the control of a spacecraft to a neural network is inadvisable.

1.3 Intended contributions and thesis overview

This work proposes an innovative methodology to autonomously select the observation times during the imaging campaign of an unknown small body. In particular, an autonomous planning policy is designed through DRL, under the general framework of exploration planning.

The principal intended contributions of this thesis are to increase spacecraft autonomy and to enhance the mapping process, raising the efficiency of data collection without compromising the mission safety. The goal of this thesis is to produce an algorithm that can be mission-independent, robust and highly flexible, all necessary characteristics for applicability in unknown environments. Moreover, to be suitable for real applications, the proposed method aims to be computationally efficient.

The present thesis offers a different point of view, proposing a novel approach, and tries to make a step forward in this research field. The high level planning framework has been studied only in few literature works [31], [36], where the algorithms were more focused on trajectory planning.

Given the early stage of this research field and its multidisciplinary nature, the design of the proposed methodology starts from a targeted analysis of the involved topics. A scheme of all related topics is shown in Figure 1.1. These include mapping techniques for shape model reconstruction, dynamics in proximity of small bodies, the exploration problem, related to POMDP and active SLAM, solving methodologies (Reinforcement Learning) and tools (Neural Networks).

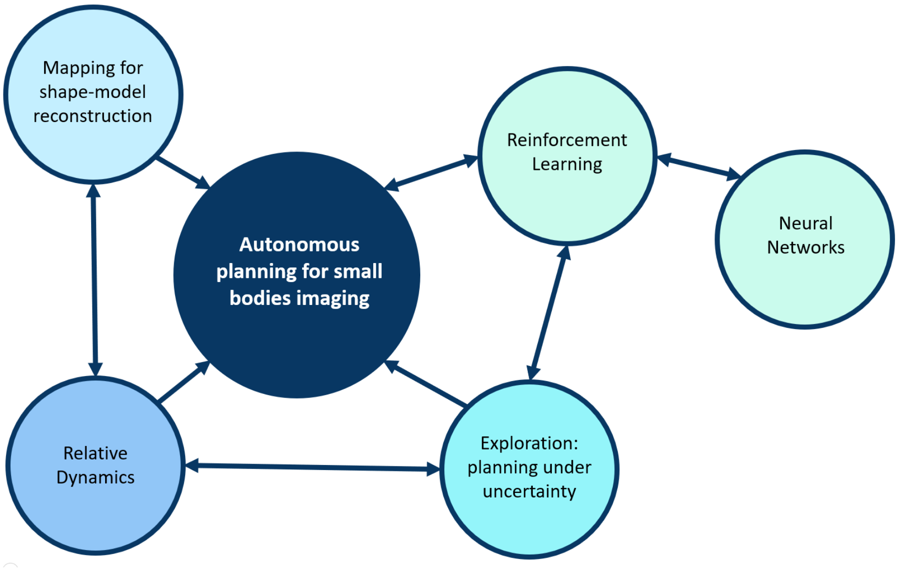

Mapping for shape-model reconstruction
Reinforcement Learning
Neural Networks
Autonomous planning for small bodies imaging
Relative Dynamics
Exploration: planning under uncertainty

Figure 1.1: Schematic overview of the involved topics

The thesis is structured as follows:

- Chapter 2 describes shape model reconstruction techniques, providing necessary information to understand which aspects of small bodies imaging can be improved and highlighting the main drivers for enhancing the mapping process.

- Chapter 3 presents background knowledges about orbital dynamics in proximity of small bodies. A general overview of typical trajectories and strategies selected for body mapping is also provided.

- In Chapter 4 POMDP mathematical formulation is introduced, then the active SLAM problem is presented, allowing to define small bodies exploration general framework.

- Chapter 5 deals with decision processes solving methods, focusing on Reinforcement Learning and presenting the chosen algorithms, Neural Fitted Q

(NFQ) and Deep Q Network (DQN). Neural Networks are briefly introduced as well, being a necessary tool for DRL algorithms.

- In Chapter 6 an innovative planning architecture for small bodies imaging is proposed. Design choices derive from the considerations reported in previous chapters. All modeling assumptions are presented and justified.
- In Chapter 7 learning scenario and hyperparameters design is described. The testing procedure is defined and the simulation environment presented. Results are shown and critically analyzed, comparing the performance of the proposed solution with simple benchmarks. A computational analysis is reported to investigate the limits of the chosen methodology.
- Chapter 8 concludes this thesis, summarizing its contributions and proposing possible future developments.
- Appendix A contains the RPROP algorithm, used for the neural network training.

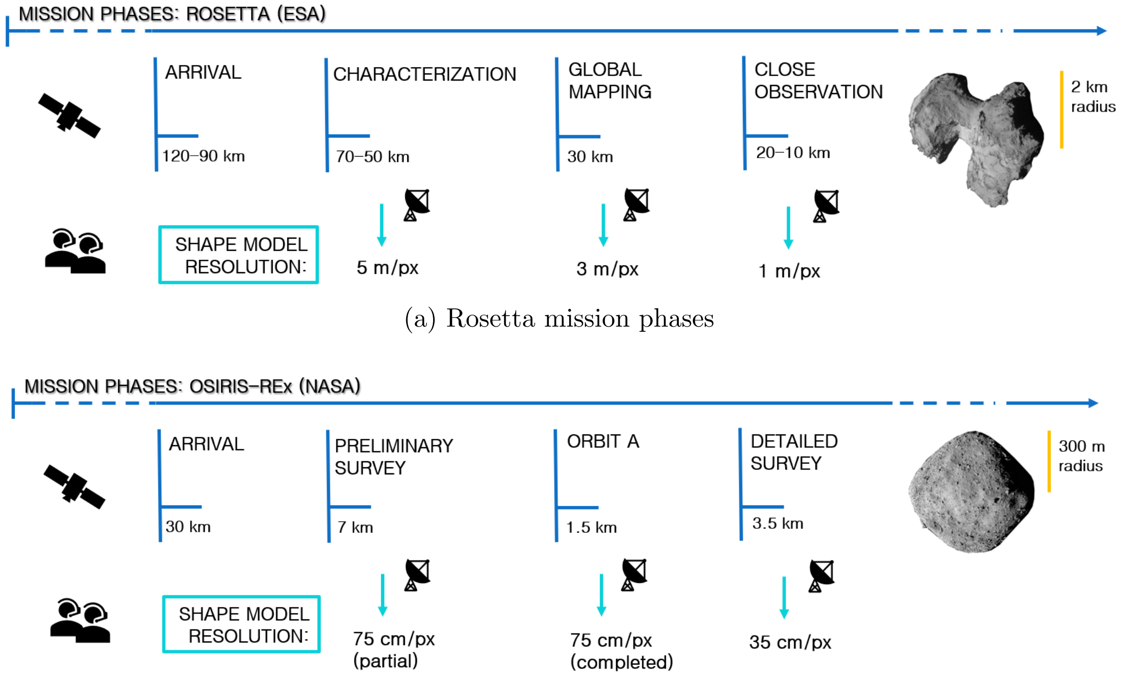
2 Small bodies mapping

THE aim of this chapter is to provide the reader with context and problematics related to small bodies mapping, giving an overview of current mapping procedures. First, the typical mission architecture and mapping phases are presented, defining a common framework for the relevant aspects to small bodies imaging. Then, it is briefly described how shape reconstruction is performed through SPC and SPG. Conclusions about key elements to enhance the mapping process are drawn.

2.1 Mapping missions common framework

2.1.1 Missions architecture

The operational phases in which the approach to the body is scheduled present many similarities in past and current missions. In Figure 2.1a and Figure 2.1b a scheme of mission phases involving image acquisition for high resolution shape model reconstruction is displayed for Rosetta and OSIRIS-REx missions [5], [9]. After the interplanetary travel the spacecraft arrives at the body and the characterization phase starts. Basic characterization of the body is done to retrieve informations like an approximated gravitational field and the body spin axis orientation and spin rate. A preliminary shape model is also built. Then, at a closer distance, the global mapping starts and the shape model is enhanced. This phase is followed by a close observation phase, needed to complete the topographical, thermal and mineralogical characterization necessary for landing site selection. During all these phases images of the body are collect and sent on ground, where they are elaborated. While the mission proceeds and the spacecraft gets closer to the body, several shape models



(a) Rosetta mission phases

(b) OSIRIS-REx mission phases

Figure 2.1: Mapping phases

are built with increasing resolution.

For the purpose of outlining a common framework, three aspects are of particular interest: the typical distances adopted while mapping the body, instrumentation characteristics and surface portion examined.

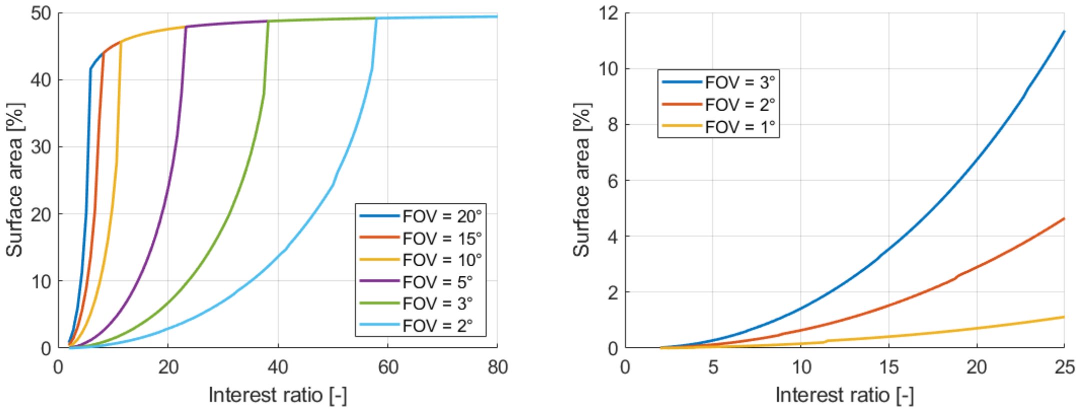
A useful quantity to describe the distance is the *interest ratio*, i.e. the ratio between distance from body center and maximum body radius. Such quantity allows to compare scenarios involving different asteroids or comets. As a general criterion, during the characterization phase the interest ratio is about 25. Then, for the global mapping phase it ranges between 15 and 10, while in the close observation phase it assumes values of 10-4. These are not strict thresholds.

For what concerns typical optical instruments, usually more than one camera is present on-board, with field of views varying from about 1° to 20°. The reason for this wide range of FOVs is that optical instruments can have different tasks, like shape reconstruction, navigation, dust or coma monitoring. Some examples of cameras devoted to imaging for topography are reported in Table 2.1 [37]–[39]. These cameras possess a Charge-Coupled Device (CCD) array, therefore the pictures quality could be affected by smearing, when the relative velocity between camera and body surface is too high or when two pictures are taken at too close time instants. It can be noticed that the FOV is really narrow, thus allowing to collect high resolution images when still far from the body surface.

Camera	Mission	FOV	Pixels
NAC	Rosetta	2.2°	2048x2048
PolyCam	OSIRIS-REx	0.792°	1024x1024
MapCam	OSIRIS-REx	3.99°	1024x1024

Table 2.1: Camera parameters

The surface percentage of a spherical body visible in the camera frame is displayed in Figure 2.2, varying distance from the body and camera field of view. As it can be seen, the visible surface portion that can vary from roughly half of the body to below the 1%. In particular, during the global mapping phase it is near to the lower bound, with typical FOVs.



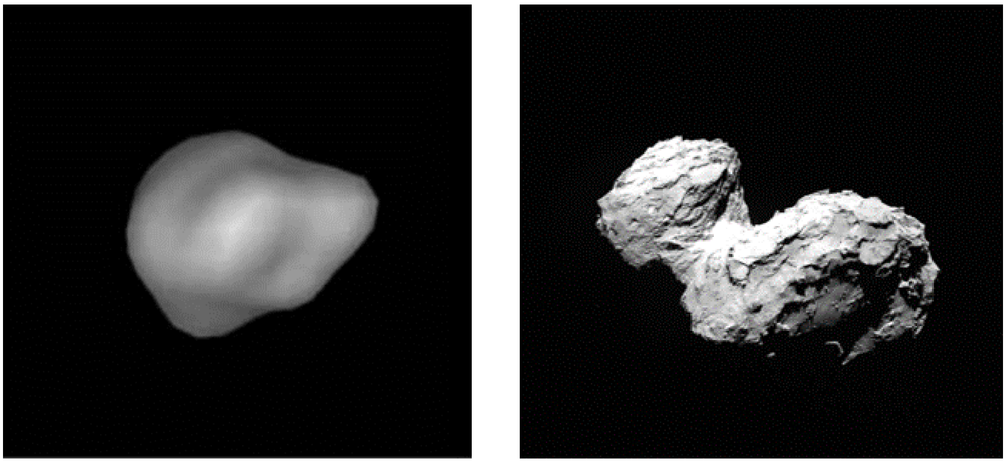
(a) Different mission phases

(b) Mapping phases

Figure 2.2: Percentage of body surface visible in the camera FOV, varying the interest ratio

2.1.2 Shape model reconstruction

Before the mission a rough shape of the body is derived from light curves inversion. In Figure 2.3 an image of comet 67P Churyumov - Gerasimenko nucleus is compared with the shape predicted from light curves. As it can be observed, in this case the expected shape is not well representative of the real one. Since they can be far different from the real object, light curves models can not be trust as accurate shape reconstructions: high fidelity topography is possible only through a close body exploration.



(a) Light curve inversion prediction [40]

(b) Real image [8]

Figure 2.3: 67P Churyumov-Gerasimenko nucleus shape

When some images of the body are available, fast and robust methods to obtain a shape model with coarse resolution are silhouette and shadow carving [41], [42]. In silhouette carving the body shape is initialized as an ellipsoid or a cube, from which volume is subsequently subtracted. This method is conservative and does not deal with concave regions, so shadow carving needs to be successively applied to carve out concavities.

The shape model is then refined during the subsequent observations of the body, until a high resolution model is obtained. Two main methods are used for this purpose: SPC and SPG. Both techniques are described in the following sections.

2.2 Stereophotoclinometry

SPC is a 3D reconstruction method that descends from shape-from-shading, combining photometry and stereoscopy. The SPC technique was first introduced by Gaskell [19], [43] and is based on the derivation of *maplets* (or *L-maps*): small scale, 3D high resolution maps, centered on points called landmarks. The creation of L-maps is achieved by estimating landmark positions and camera poses in an iterative process, exploiting a *photometric model.* The idea is that surface normals can be estimated considering how light is reflected. Once the maplets are created, they can be assembled together in the global shape model.

In this section the method overview is provided, first introducing two necessary tools: the reference frames and the photometric model.

2.2.1 Reference frames

Several reference frames are useful for our purposes. First of all, the body frame $\mathbf{b_i}$, fixed with the body rotation. Typically the body-fixed frame has the two first axes lying on the equatorial plane, with the first one defining the prime meridian. The third axis coincides with the spin axis. Given the landmark position $\mathbf{V}$ and the spacecraft position $\mathbf{W}$, both expressed in in body-fixed frame, let's define:

- the camera frame $\mathbf{c_i}$, centered on the camera and with axes $\mathbf{c_1}$ and $\mathbf{c_2}$ belonging to the image plane and $\mathbf{c_3}$ in the camera principal axis direction.
- the landmark frame $\mathbf{u_i}$ with axes aligned to East, North and landmark position direction, centered in the landmark.

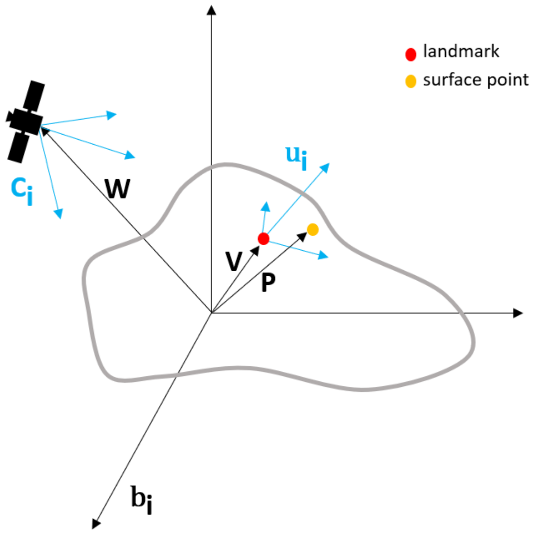


Figure 2.4: L-map and camera frames

A point on the surface with body-fixed coordinates $\mathbf{P}$ can be represented in the L-map frame with coordinates:

$$x_1 = (\mathbf{P} - \mathbf{V}) \cdot \mathbf{u_1}$$
$$x_2 = (\mathbf{P} - \mathbf{V}) \cdot \mathbf{u_2}$$
$$h(x_1, x_2) = (\mathbf{P} - \mathbf{V}) \cdot \mathbf{u_3}$$

and in the camera frame coordinates:

$$X = (\mathbf{P} - \mathbf{W}) \cdot \mathbf{c_1}$$
$$Y = (\mathbf{P} - \mathbf{W}) \cdot \mathbf{c_2}$$
$$Z = (\mathbf{P} - \mathbf{W}) \cdot \mathbf{c_3}$$

According to the classical pinhole camera model (for instance see [44]), the point can be expressed in homogeneous coordinates and then in the camera coordinate frame in the following way:

$$\left[X \; Y \; Z\right]^T \rightarrow \left[f\frac{X}{Z} \; f\frac{Y}{Z}\right]^T$$

where f is the focal length of the instrument. Therefore, the point coordinates in the image are:

$$X_1 = f\frac{(\mathbf{P} - \mathbf{W}) \cdot \mathbf{c_1}}{(\mathbf{P} - \mathbf{W}) \cdot \mathbf{c_3}}$$
$$X_2 = f\frac{(\mathbf{P} - \mathbf{W}) \cdot \mathbf{c_2}}{(\mathbf{P} - \mathbf{W}) \cdot \mathbf{c_3}}$$

The above equations can be rewritten as:

$$X_i = f\frac{(\mathbf{V} - \mathbf{W}) \cdot \mathbf{c_i} + M_{i1}x_1 + M_{i2}x_2 + M_{i3}h}{(\mathbf{V} - \mathbf{W}) \cdot \mathbf{c_3} + M_{31}x_1 + M_{32}x_2 + M_{33}h} \qquad i = 1, 2 \tag{2.1}$$

where $M_{ij} = \mathbf{c_i} \cdot \mathbf{u_j}$, with $i = 1, 2$ and $j = 1, 2, 3$. So equation 2.1 shows the geometric relation between the coordinates of a surface point in the maplet frame with its corresponding coordinates in an image taken by the camera.

2.2.2 The photometric model

The photometric model allows to link the brightness of a pixel to the surface slopes and albedo, through a reflectance function R. Given an image k, the brightness I_k of the cell $\mathbf{x} = (x_1, x_2)$ can be predicted with the photometric model:

$$I_k(\mathbf{x}) = \Lambda_k a(\mathbf{x}) R(\phi, e, i) + \Phi_k \tag{2.2}$$

where a is the relative albedo, normalized to have a unitary mean over the map, Φ_k is the background level and Λ_k is a scale factor that includes conversion from the Sun flux intensity to the pixel signal rate. Please note that the photometric model needs some parameters to be tuned in advance (Φ_k, Λ_k).

The reflectance R of a surface depends on the photometric angles shown in Figure 2.5 :

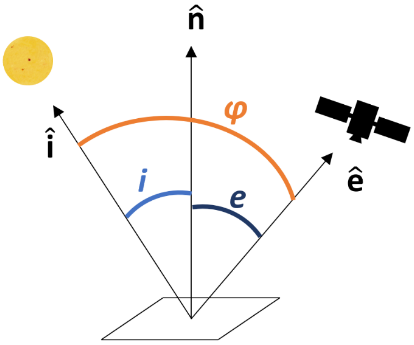


Figure 2.5: Angle definition in photometry

- The local angle of incidence i, which is the angle between Sun incidence unit vector $\mathbf{i}$ and surface normal $\mathbf{n}$.
- The local angle of emission e, which is the angle between emission, $\mathbf{e}$ (i.e the spacecraft unit vector) and surface normal.
- The phase angle ϕ, which is the angle between the Sun incidence vector and the emission vector.

Other two angles useful to define the illumination and viewing conditions are:

- The solar azimuth angle α, that is the angle from local North to the incidence vector, projected on the surface plane.
- The spacecraft azimuth angle β, between the emission vector projection on the surface plane and the local North.

Several models of the reflectance function are present in the literature [43], [45]. The reflectance function can be modeled as the weighted sum of Lambert reflectance R_L, that models a pure diffusion reflection, and Lommel-Seeliger reflectance R_{LS}, that models a specular reflection:

$$R = P(\phi)\Big[(1 - L(\phi))R_L(i) + L(\phi)R_{LS}(i,e)\Big] \tag{2.3}$$

$L(\phi)$ is a transition function, $P(\phi)$ represents an exponential decrease with the phase angle. The reflectance models are:

$$R_L(i) = \mathbf{n} \cdot \mathbf{i} \tag{2.4}$$

$$R_{LS}(e,i) = \frac{\mathbf{n} \cdot \mathbf{i}}{\mathbf{n} \cdot (\mathbf{i} + \mathbf{e})} \tag{2.5}$$

Defining t_1 and t_2 as the surface slopes respectively along $\mathbf{u_1}$ and $\mathbf{u_2}$, the local surface normal can be written as:

$$\mathbf{n} = \frac{\big[-t_1, -t_2, 1\big]^T}{\sqrt{t_1^2 + t_2^2 + 1}} \tag{2.6}$$

In conclusion, the power of the photometric model consists in the fact that the pixel intensity in an image depends only on the surface slopes and albedo, when the vectors $\mathbf{e}$ and $\mathbf{i}$ are known.

2.2.3 The problem statement

The potential unknowns of the problem are the spacecraft position vector $\mathbf{W}$, the camera oriantation $\mathbf{c_i}$, the landmark position and the heights of the map. Two different sub-problems can be identified:

1. If an ensemble of landmarks is exactly known for one image, than the camera orientation and the spacecraft location can be retrieved.
2. Conversely, if the camera pose is known for several images, the landmark location can be derived.

The global problem can be solved with an iterative method alternatively solving the two sub-problems and with optical data only except from some information: the scale, that can be provided by radio data, and the center of mass position, that can be determined with doppler data. This is linked to the center of the figure.

2.2.4 The shape model generation process

SPC procedure starts by building the single maplets, that than are assembled together. The maplet generation process consists in estimating slopes and albedo at each map cell from several images, then integrating slopes into heights, with a complex procedure that comprises different iterative loops. Insights on the overall process can be found in literature [43], [45]. A simplified scheme is shown in Figure 2.6.

The process starts with maplet Digital Elevation Model initialization: the L-map reference frame is defined and all the heights are set to zero, thus at the beginning of the maplet generation process the L-map is a flat surface.

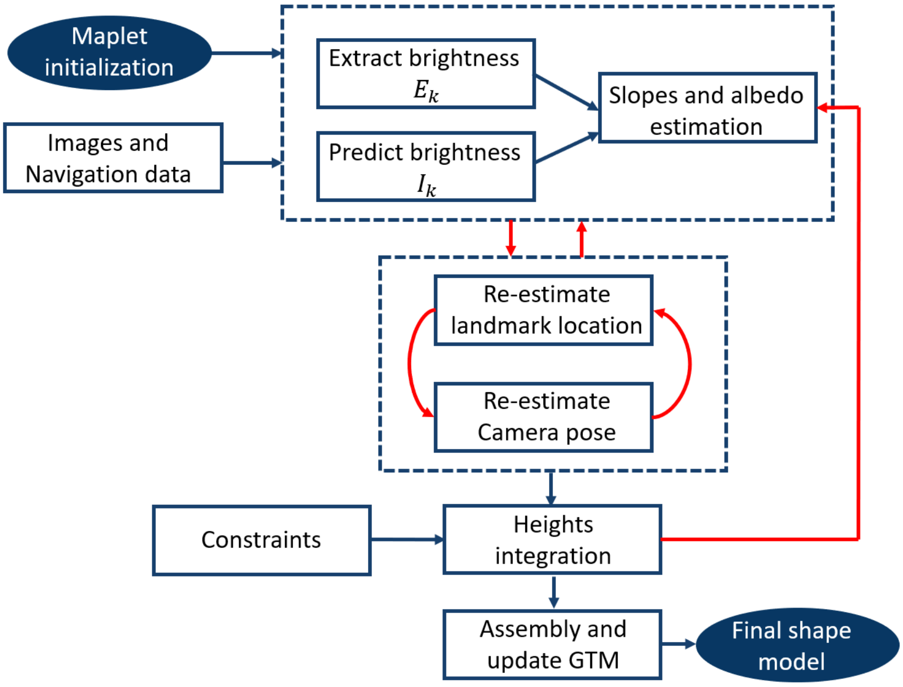


Figure 2.6: SPC process block diagram

The first step is brightness extraction. By supposing to know the current camera pose in the body fixed frame, the image brightness corresponding to the map cell $\mathbf{x}$ can be retrieved. Data are mapped from image k to each L-map cell: for each point of the map $\mathbf{x}$, the corresponding locations in image coordinates $\mathbf{X}$ are found with equation 2.1 and data at map location $\mathbf{x}$ are extracted from image k. Let's call E_k the extracted brightness. This process is also known as ortho-rectification.

The second step is brightness prediction. Brightness can be predicted from the current maplet, exploiting the photometric model in equation 2.2 and current values of slopes and albedo, that of course are functions of the maplet cell. So a predicted brightness I_k at map location $\mathbf{x}$ is computed. Again, the necessary input data are the spacecraft and Sun position in body fixed coordinates.

As third step slopes and albedo can be recomputed at each map pixel minimizing the weighted sum square residual between expected and extracted brightness in several images (3 at least).

Unfortunately, in practice $\mathbf{V}$, $\mathbf{W}$ or $\mathbf{c_i}$ are not exactly known, so the actual and predicted image will be not aligned. After several landmark positions are estimated in camera coordinates for several images, $\mathbf{V}$, $\mathbf{W}$ and $\mathbf{c_i}$ are iteratively re-estimated. This iterative process alternates the two sub-problems defined in the problem statement subsection 2.2.3.

The first sub-problem is solved by minimizing the weighted sum square residuals between the observed $\mathbf{Y}$ and predicted $\mathbf{X}$ values of landmark positions in the image, over all landmarks in the considered image.

$$\sum_{\text{landmarks}} (\mathbf{Y} - \mathbf{X})^T \mathbf{D} (\mathbf{Y} - \mathbf{X}) \tag{2.7}$$

where $\mathbf{D}$ is a weight matrix accounting for uncertainties. The observed landmark location $\mathbf{Y}$ in the image is computed by correlating predicted brightness distribution I_k with the extracted one E_k. The predicted landmark location $\mathbf{X}$ is computed through equation 2.1. The output is a new estimation of $\mathbf{W}$ and $\mathbf{c_i}$.

The second sub-problem is solved by minimizing the weighted sum square residuals between the observed $\mathbf{Y}$ and predicted $\mathbf{X}$ coordinates of one landmark over all the images containing it.

$$\sum_{\text{images}} (\mathbf{Y} - \mathbf{X})^T \mathbf{D} (\mathbf{Y} - \mathbf{X}) \tag{2.8}$$

Correlating the map images, a shift is determined and the new landmark location in the real image is computed, so the output is a new estimation of $\mathbf{V}$. With the new estimations of $\mathbf{V}$, $\mathbf{W}$ and $\mathbf{c_i}$ everything is iterated from first step on.

At this point, the estimations of slopes and albedo are still rough because heights are unknown. Please note that if the height is not correct, not only the prediction will not be realistic, but also a wrong mapping will cause the extraction of brightness from a wrong pixel. So the steps described up to now are nested inside another loop, that includes heights computation. Simplifying, the height is integrated from slopes and from nearest neighbor heights. Information coming from preliminary shape models, stereoscopy or overlapping maplets data are exploited as well. From the new computed heights, updated slopes are derived.

All the procedure described in this section is repeatedly applied to the map until convergence. After the generation of all the maplets, they are assembled together to form the Global Topography Model (GTM), a high resolution shape model. The starting point is the low resolution reference shape. For each point of the reference shape a line is run in the local normal direction until an L-map is encountered at some height. Since L-maps overlap, an average of the heights is computed and the point position of the shape model is updated. The GTM resolution can be increased ad the mission goes on.

In addition to landing site selection and scientific investigations, a possible application of maplets is refinement of navigation data. With the nominal spacecraft position and camera pose, the photometric model is used to simulate the L-map image brightness. Then, thanks to cross-correlation, the offset in the original image is found. At least 3 maplets per image are needed to estimate both spacecraft position and camera orientation. To have a very accurate and complete map is therefore essential for an accurate navigation. Since the final maplet has been generated from a large number of images, it is superpixelized and can easily be correlated with images that have 5 times the resolution of the original images. This allow to navigate closer to the body with respect to the distance at which images were taken.

2.3 Stereophotogrammetry

SPG is a dense stereo method. Stereo methods compute the position of some control points by triangulating them in several images. In other words, patches from a reference image are correlated to other images to find point correspondences. Then, operations like rotations, translations and distortions are found to go from one image to another and the 3D topography is deduced. SPG aims just to estimate the heights of the topographic model, so no albedo information is deduced. A detailed description can be found in Giese's works [17], [18].

An initial bundle block adjustment is applied starting from recorded navigation data associated to several images. Collinearity equations relate the image coordinates with camera pose and body-fixed coordinates (similarly to equation 2.1) and are solved with a least-squares adjustment, thus improving accuracy. Then, images are correlated in order to find a large number of conjugate points. An image is taken as reference and an automatic area-based matching strategy is applied: the patterns identified in the reference image patched are searched for in the stereo target images. The coordinates of conjugate points are converted to the body-fixed frame coordinates using collinearity equations. Finally the topographic model is interpolated.

SPG does not require to know any preliminary shape, but presents some limitations: the low resolution, that is comparable to the patch size, and the sensitivity to illumination changes, that affects the conjugate points determination. Nevertheless, SPG can be combined with SPC with the aim of finding a grid of anchor points distributed in the L-map surface. Simulating the appearance of an image with known slopes and albedo, anchor points can be observed. They are then used in the height integration process as known constraint heights.

2.4 Possible improvements

In conclusion, high resolution shape model reconstruction is a complex procedure that can only be performed on-ground, requiring to elaborate considerable amounts of data with several loops. Nevertheless, the process can be improved in the autonomy direction. Fundamental inputs to the process are the knowledge of Sun direction and spacecraft pose, related to navigation data, as well as images of the whole body with characteristics that could ease the selected mapping technique. In particular, two frontiers are foreseen to ease on-ground analyses and decrease human support:

- Autonomous navigation with accurate relative pose estimation.
- Image collection optimization, with appropriate illumination and stereo angles.

Both features are also linked to trajectory planning. Of course, depending on the adopted mapping technique different requirements arise: SPC benefits from large illumination variation, while SPG from constant illumination and different stereo angles. The direction in which the surface is illuminated and observed of course depends on the body shape, that is not completely known. Considering that the spacecraft is exploring an unknown environment, automation and improvement of all these aspects is not a trivial problem.

3 The relative dynamics

The necessity of taking several pictures of the same area with suited emission angles and illumination conditions makes the mapping problem tightly bonded to the relative dynamics between asteroid and spacecraft. Moreover, uniform coverage of the entire surface should be granted, producing a global map of the unknown body while orbiting around it.

Having small bodies a low gravitational attraction, perturbations play a key role in influencing the dynamics. The main forces acting on the spacecraft are the Sun attraction, the Solar Radiation Pressure (SRP) and of course the gravity field of the body. Asteroids and comets present heavily irregular shapes and therefore the gravitational field in their surroundings is irregular as well. Several models of the gravitational potential are present in literature. Different models also exist to describe the relative dynamics between spacecraft and body, depending on perturbations magnitude. Relevant contributions on this topic are related to Scheeres's works [46]–[50].

The purpose of this chapter is to give some insight into the dynamical environment possibly encountered during small bodies mapping. Typical orbit strategies are presented, a general dynamical model is described. Then magnitudes of accelerations acting on the spacecraft are briefly analyzed, comparing some possible scenarios.

3.1 Dynamical environment and typical strategies

Small bodies are characterized by different masses, dimensions, shapes and rotational states. Orbital parameters and physical data derived from ground observations and space missions can be found on JPL on-line Small Bodies Database Browser [51]. The typical body shapes can be quasi-spherical, typically for big or fast rotating asteroids, or irregular (dog-bone, two-masses, elongated), with diameters that have an order of magnitude ranging from 10^{-1} to 10^{2} km and masses that can go from 10^{18} kg to below 10^{10} kg. Also the rotational dynamics is very diversified: slow rotating asteroids can have a rotation period of the order of 10^{2} h, while for fast rotating asteroids there is a spin barrier of 2.2 h [52]. The typical case is a uniform rotation around the major axis of inertia, but also small bodies tumbling in an arbitrary rotation state exist. All the above mentioned variables concur to affect the spacecraft-body relative dynamics and the illumination conditions variation of the body surface, thus influencing the body mapping.

The dynamic environment in proximity of small bodies is challenging: the gravitational field can be highly irregular and perturbations like SRP, the gravitational perturbation due to the Sun and comet outgassing may play a dominant role. For this reason, orbits in proximity of small bodies have a non-Keplerian nature. Orbit maintenance is a problematic issue in these highly perturbed environments, where orbits are likely not stable and the spacecraft may escape or impact the body.

As a consequence of the rich and challenging dynamical environment, trajectory design and planning is strictly mission-dependent. Several possible strategies for mapping trajectories exist.

Families of stable orbits can be found in environments where SRP is the largest perturbation; these orbits do not require active control and therefore are inexpensive. In particular, three families of orbits are the most studied in literature: ecliptic, terminator and quasi-terminator orbits [22], [49], [50]. The framework in which these families are found is the Augmented Normalized Hill three Body Problem (ANH3BP). Ecliptic orbits are periodic, but may require maintenance maneuvers and are suitable to observe only the region of the body near to the ecliptic plane. Terminator orbits lie in the plane perpendicular to the Sun and are highly stable. The main drawback of this solution is that the angle between spacecraft and Sun is always 90°, limiting the imaging opportunities. Quasi-terminator orbits are particularly good for global mapping campaigns because they are stable and also offer a good variation of Sun-relative geometries. However their applicability is in practice limited, depending on mission time scales, length scales and minimum allowable orbit radius.

Other strategies are based on actively controlled trajectories. In fact, when the body mass is small such strategies can still be actuated with reasonable costs and offer the possibility to easily obtain the desired Sun-spacecraft-body relative geometry. Drawbacks are that fuel cost may become important if the strategy is extended for a long time and that maneuvers require ground supervision. Examples of controlled trajectories are direct hovering in the body-fixed or inertial frame, as well as flybys, conic-like trajectories or ping-pong orbits [21], [53], [54].

3.2 Dynamical model

In this section a general model of spacecraft dynamics around small bodies is described. Let's consider an inertial frame $\mathcal{I}$, a body-fixed frame $\mathcal{B}$ rotating with angular velocity $\boldsymbol{\omega}_{sb}$. The body frame $\mathcal{B}$ can be defined with the first two axes lying on the small body equatorial plane and the third one aligned with its spin axis. Typically the spin axis orientation is defined on catalogs through its right ascension α_0 and declination δ_0 at a given reference time.

The spacecraft is subjected to the small body gravitational field and to the other perturbing accelerations, in particular the solar radiation pressure and the Sun gravity. The spacecraft translational dynamics in the inertial frame is:

$$\ddot{\mathbf{r}}_{\mathcal{I}} = \nabla U_{\mathcal{I}} + \mathbf{a}_{p,\mathcal{I}} \tag{3.1}$$

where $\mathbf{a}_{p,\mathcal{I}}$ are the perturbing accelerations acting on the spacecraft and $\nabla U_{\mathcal{I}}$ is the body potential gradient. It has to be noticed that the potential evaluation depends on the spacecraft position relative to the body.

Defining $\mathbf{R}_{\mathcal{I}\mathcal{B}}$ as the rotation matrix from the body frame to the inertial one, $\mathbf{r}_{\mathcal{I}} = \mathbf{R}_{\mathcal{I}\mathcal{B}}\mathbf{r}_{\mathcal{B}}$, where $\mathbf{r}_{\mathcal{B}}$ is the relative position expressed in the body-fixed frame. Applying the transport theorem, the dynamics can be expressed in the body-fixed frame:

$$\ddot{\mathbf{r}}_{\mathcal{B}} + \dot{\boldsymbol{\omega}}_{sb} \wedge \mathbf{r}_{\mathcal{B}} + 2\boldsymbol{\omega}_{sb} \wedge \dot{\mathbf{r}}_{\mathcal{B}} + \boldsymbol{\omega}_{sb} \wedge \boldsymbol{\omega}_{sb} \wedge \mathbf{r}_{\mathcal{B}} = \nabla U_{\mathcal{B}} + \mathbf{a}_{p,\mathcal{B}} \tag{3.2}$$

This equation is coupled with the small body rotational dynamics. Anyway, typically small bodies have uniform rotations around their principal inertial axis, so it is possible to simplify the dynamics assuming a constant spin:

$$\ddot{\mathbf{r}}_{\mathcal{B}} + 2\boldsymbol{\omega}_{sb} \wedge \dot{\mathbf{r}}_{\mathcal{B}} + \boldsymbol{\omega}_{sb} \wedge \boldsymbol{\omega}_{sb} \wedge \mathbf{r}_{\mathcal{B}} = \nabla U_{\mathcal{B}} + \mathbf{a}_{p,\mathcal{B}} \tag{3.3}$$

Even with the assumptions made, equation 3.3 still remains very general. During an initial design phase, simplified scenarios are usually considered, to have general insights on the dynamic environment in proximity of the small body. In particular, two different regimes are described in [47]: one in which the irregularity of the gravity field is negligible and another in which all the perturbations coming from the Sun are of secondary importance.

3.3 Body gravitational potential

Small bodies have irregular shapes, therefore their gravity field is not well approximated by the one produced by a point mass. In its most general expression, the gravitational potential sensed by a point mass in proximity of a body is equal to

$$U = \int_M \frac{Gdm}{\rho} \tag{3.4}$$

where M is the total mass of the body, G is the universal gravitational constant and ρ is the relative distance between the point mass and the differential mass dm.

Several methods exist in the literature to model the gravitational potential. The most accurate one is the polyhedron model, introduced by Werner *et al.* [55]. The polyhedron model transforms the volume integral of the potential into an integral over the body surface, approximated as a polyhedron. Such model can be relied as groundtruth and has a validity domain up to the surface level, but it is computationally expensive. Since the global mapping phase is carried out far from the body surface, a less accurate model is sufficient. Harmonics expansion methods find an analytic approximation of the potential, that is valid only in a region outside a reference surface. In particular, for spherical harmonics (SH) such surface is the smallest sphere enclosing the body (also called Brillouin sphere), for the ellipsoidal ones (EH) it is the smallest ellipsoid. To make an accurate and long term dynamic model of the asteroid approach is beyond the scope of this thesis. In particular, SH are considered.

3.3.1 Spherical harmonics approximation

The potential function at a point P outside the Brillouin sphere can be expressed an infinite summation of harmonics, weighted with some mass coefficients that encode informations about the body mass distribution.

$$U = \frac{GM}{r} \sum_{n=0}^{\infty} \sum_{m=0}^{n} \left(\frac{a_e}{r}\right)^n P_{nm}(sin\phi)(C_{nm}cos(m\lambda) + S_{nm}\sin(m\lambda)) \tag{3.5}$$

where a_e is a reference radius, P_{nm} are Legendre polynomials ($m = 0$) or associated Legendre functions ($m > 0$) of degree n and order m. C_{nm} and S_{nm} are the mass coefficients and in particular, C_{n0} and S_{n0} are called *zonal harmonics*, C_{nn} and S_{nn} are referred to as *sectorial harmonics* and the remaining terms are the so called *tesseral harmonics*. The radius r, latitude ϕ and longitude λ are the spherical coordinates of the field point P in the body-fixed reference frame. In numerical applications, the expression in equation 3.5 is truncated at a maximum degree N and a normalization is performed to avoid having mass coefficients with large magnitude variations, thus avoiding numerical issues.

The acceleration sensed by the point mass is equal to the potential gradient. The gravitational acceleration is comprehensive of the central body gravity, i.e. the zero degree term in equation 3.9, and the higher degree expansion terms, which correspond to the body gravitational perturbation. Please note that gradient of potential is already expressed in the body-fixed frame and depends on the considered field point.

$$\nabla U = \mathbf{a_b} + \mathbf{a_{SH}} \tag{3.6}$$

When deriving the potential gradient in spherical coordinates $(\mathbf{u_r}, \mathbf{u_\phi}, \mathbf{u_\lambda})$ a singularity appears at the poles:

$$\nabla U = \frac{\partial U}{\partial r}\mathbf{u_r} + \frac{1}{r}\frac{\partial U}{\partial \phi}\mathbf{u_\phi} + \frac{1}{rcos\phi}\frac{\partial U}{\partial \lambda}\mathbf{u_\lambda} \tag{3.7}$$

In 1973 Pines derived a singularity-free formulation for the geopotential [56], expressing it in terms of derived Legendre polynomials, also called Helmholtz polynomials. The field point coordinates are expressed in the direction cosine coordinates

$$\begin{aligned} s &= \frac{x}{r} = cos\phi cos\lambda \\ t &= \frac{y}{r} = cos\phi sin\lambda \\ u &= \frac{z}{r} = sin\phi \end{aligned} \tag{3.8}$$

leading to the following truncated expression of the potential:

$$U = \frac{GM}{r} \sum_{n=0}^{N} \sum_{m=0}^{n} \left(\frac{a_e}{r}\right)^n H_{nm}(u) D_{nm}(s,t) \tag{3.9}$$

Where D_{nm} is the mass coefficients function and H_{nm} are the Helmholtz polynomials. The potential gradient is directly computed in cartesian coordinates. In 1988 Lundberg *et al.* compared several recursion schemes for the computation of fully normalized Helmholtz polynomials and their derivatives [57]. More recently, Fantino and Casotto developed an algorithm with a lumped coefficients approach:

$$U = \sum_{m=0}^{N} (A_m^{(1)} cos(m\lambda) + B_m^{(1)} sin(m\lambda))(cos\phi)^m \tag{3.10}$$

where $A_m^{(1)}$ and $B_m^{(1)}$ are the lumped coefficients, that are functions of fully normalized Helmholtz polynomials and mass coefficients. The derivation of the gradient of potential in cartesian coordinates and the details of the algorithm can be found in their work [58].

3.3.2 Mass coefficients computation

In real applications mass coefficients are derived from range data, but if the shape model of a body is known, as well as its density distribution, they can be computed via analytical methods [59].

By definition the mass coefficients are:

$$\begin{aligned} \begin{bmatrix} C_{nm} \\ S_{nm} \end{bmatrix} &= \frac{2-\delta_{0m}}{M} \frac{(n-m)!}{(n+m)!} \int_{body} \left(\frac{r}{a_e}\right)^n P_{nm}(sin\phi) \begin{bmatrix} cosm(\lambda) \\ sin(m\lambda) \end{bmatrix} dm \\ &= \int_{body} \begin{bmatrix} c_{nm} \\ s_{nm} \end{bmatrix} dm \end{aligned} \tag{3.11}$$

As already mentioned, a normalizing factor is commonly introduced:

$$N_{nm} = \sqrt{\frac{(2-\delta_{0m})(2n+1)(n-m)!}{(n+m)!}} \tag{3.12}$$

$$\begin{bmatrix} C_{nm} \\ S_{nm} \end{bmatrix} P_{nm} = \begin{bmatrix} C_{nm}/N_{nm} \\ S_{nm}/N_{nm} \end{bmatrix} P_{nm} N_{nm} = \begin{bmatrix} \overline{C}_{nm} \\ \overline{S}_{nm} \end{bmatrix} \overline{P}_{nm} \tag{3.13}$$

Assuming the body as a polyhedron with constant density σ, the integral over the body can be expressed in cartesian coordinates as the summation of different contributions, given by the tetrahedra composing the polyhedron. In fact, a facet with vertices $\mathbf{v_1} = [x_1 \quad y_1 \quad z_1]^T$, $\mathbf{v_2} = [x_2 \quad y_2 \quad z_2]^T$, $\mathbf{v_3} = [x_3 \quad y_3 \quad z_3]^T$ can be associated to a tetrahedron with vertices $\mathbf{v_1}$,$\mathbf{v_2}$, $\mathbf{v_3}$ and the origin. Then, to ease the integral computation, a transformation is performed from the tetrahedron to a standard simplex with vertices $[1 \quad 0 \quad 0]^T$, $[0 \quad 1 \quad 0]^T$, $[0 \quad 0 \quad 1]^T$ and the origin. This results in a coordinates change $[x \quad y \quad z] \rightarrow [X \quad Y \quad Z]$. The Jacobian of this transformation is $\mathbf{J} = [\mathbf{v_1} \quad \mathbf{v_2} \quad \mathbf{v_3}]$.

$$\begin{aligned}\begin{bmatrix}\overline{C}_{nm}\\ \overline{S}_{nm}\end{bmatrix} &= \int_{body}\begin{bmatrix}\overline{c}_{nm}\\ \overline{s}_{nm}\end{bmatrix} dm = \sigma \sum_{simplices}\int\int\int \begin{bmatrix}\overline{c}_{nm}(x,y,z)\\ \overline{s}_{nm}(x,y,z)\end{bmatrix} dxdydz \\ &= \sigma \sum_{simplices}\int\int\int \begin{bmatrix}\overline{c}_{nm}(X,Y,Z)\\ \overline{s}_{nm}(X,Y,Z)\end{bmatrix} \det(\mathbf{J}) dXdYdZ \\ &= \sigma \sum_{simplices}\int\int\int \sum_{i+j+k=n}\begin{bmatrix}\overline{\alpha}_{ijk}\\ \overline{\beta}_{ijk}\end{bmatrix}(X^iY^jZ^k)\det(\mathbf{J})dXdYdZ\end{aligned} \tag{3.14}$$

The integrands $\overline{c}_{nm}$ and $\overline{s}_{nm}$ are homogeneous polynomials of degree n and coefficients $\overline{\alpha}_{ijk}$ and $\overline{\beta}_{ijk}$. The coefficient expressions derive from the Legendre polynomials and they can be computed in a recursive manner. Solving the volume integral a final expression of the mass coefficients is obtained:

$$\begin{bmatrix}\overline{C}_{nm}\\ \overline{S}_{nm}\end{bmatrix} = \sigma \sum_{simplices}\det(\mathbf{J})\sum_{i+j+k=n}\frac{i!j!k!}{(n+3)!}\begin{bmatrix}\overline{\alpha}_{ijk}\\ \overline{\beta}_{ijk}\end{bmatrix} \tag{3.15}$$

3.4 Sun perturbations

A part from the small body attraction, the other most important accelerations acting on the spacecraft are the ones coming from the Sun: SRP and third-body perturbation.

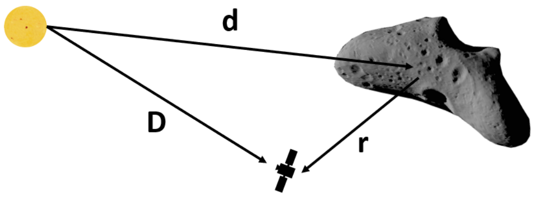


Figure 3.1: Relative Sun-spacecraft-small body position

Defining **D** as the spacecraft position relative to the Sun and **d** as the asteroid position vector relative to the Sun (see Fig 3.1), the relative position between spacecraft and asteroid in the inertial frame is:

$$\mathbf{r}_{\mathcal{I}} = \mathbf{D} - \mathbf{d} \tag{3.16}$$

Using a simple cannonball model [60], the acceleration due to Solar Radiation Pressure expressed in the inertial frame is:

$$\mathbf{a}_{\mathbf{SRP},\mathcal{I}} = \frac{P_0 A}{D^2 m} \nu C_r \hat{\mathbf{D}} \tag{3.17}$$

where A is the spacecraft cross section in light, m is the spacecraft mass, ν is the eclipse factor (equal to 0 if the spacecraft is in shadow and to 1 if in sunlight), C_r is the reflectivity coefficient, with value approximately 1, P_0 is the solar radiation pressure at 1 au (about $4.56 \times 10^{-6} \frac{\mathrm{N}}{\mathrm{m}^2}$).

While the acceleration acting on the spacecraft due to the solar gravity is the classical third body perturbation:

$$\mathbf{a}_{\mathbf{s},\mathcal{I}} = \mu_s \Big(\frac{\mathbf{d}}{d^3} - \frac{\mathbf{D}}{D^3} \Big) \tag{3.18}$$

3.5 Accelerations order of magnitude

In this section the order of magnitude of accelerations acting on the spacecraft is computed considering some test small bodies: Eros, Itokawa and Bennu. Spherical harmonics coefficients are computed up to the eighth degree exploiting shape models available on JPL Planetary Data System [61], while the bodies orbital and physical data are taken from JPL database [51].

Figures 3.3, 3.5 and 3.7 show the accelerations magnitude varying the interest ratio, including typical distances of characterization, global mapping and close observation phases. For what concerns SRP, three different mass to area ratios have been considered according to past missions data: $10 \frac{\mathrm{kg}}{\mathrm{m}^2}$, $35 \frac{\mathrm{kg}}{\mathrm{m}^2}$ and $80 \frac{\mathrm{kg}}{\mathrm{m}^2}$ [22]. Of course the irregularity of the gravity field is much more relevant when close to the body surface, as can be observed from potential gradient magnitude contours in Figures 3.2, 3.4 and 3.6.

In the case of elongated bodies like Eros the perturbations due to the irregularity of the gravity field are some orders of magnitude larger than the ones coming from the Sun, when the interest ratio is small (Figure 3.3). While for bodies with diamond or spherical shape, the spherical harmonics perturbation is much less relevant than SRP, (Figure 3.5). In other cases, like Itokawa's, SRP plays a dominant role and can be even larger than the attraction of the primary body (Figure 3.7). In other situations perturbing accelerations have a similar order of magnitude.

In conclusion, very different situations are observed and even for the same small body more scenarios are encountered varying the distance. The adoption of a unique simplified dynamic model to describe all the mapping phases or on different bodies inevitably leads to significant errors, especially when running long-time simulations. On the other hand, using a complex model that precisely describes perturbing forces does not allow to probe the dynamical environment and to find orbits suited for mapping in a repeatable way.

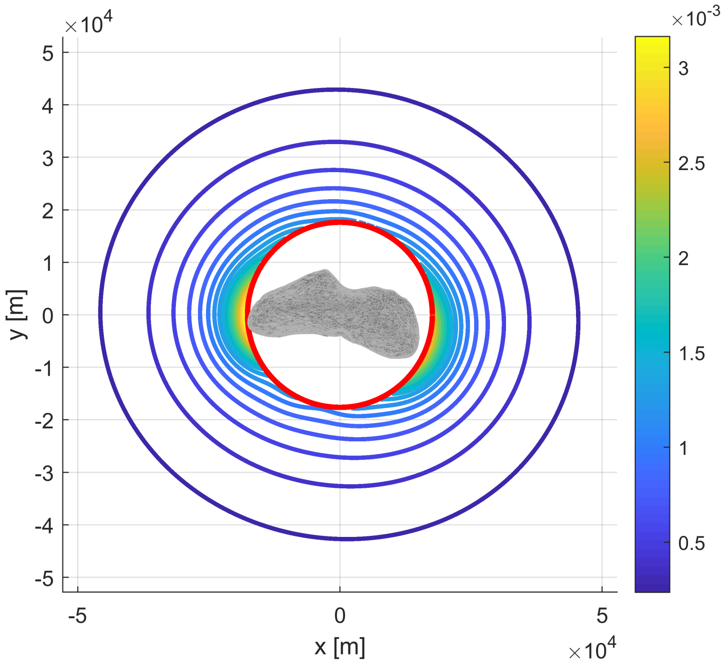


Figure 3.2: Potential gradient magnitude in proximity of Eros, outside Brillouin sphere, $[\frac{m}{s^2}]$

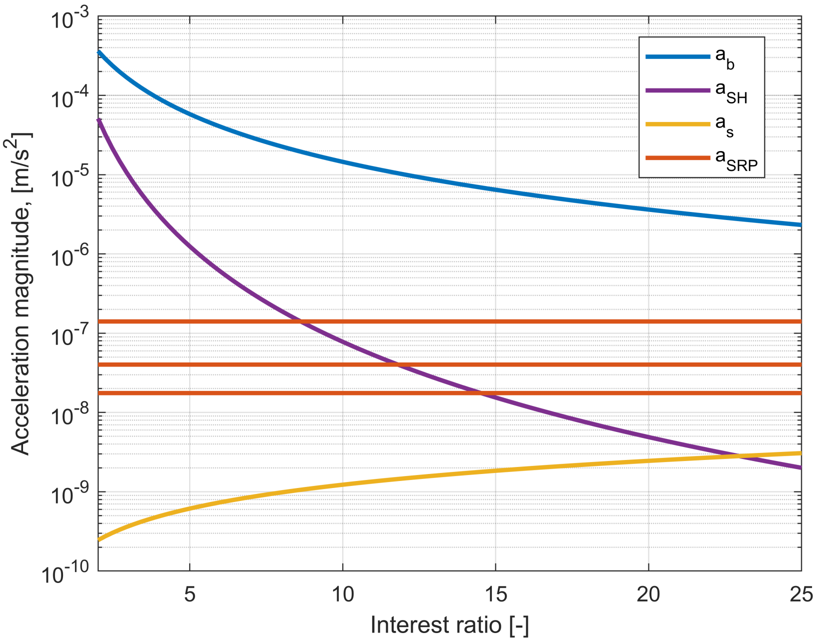


Figure 3.3: Accelerations order of magnitude acting on the spacecraft in proximity of Eros, at its aphelion

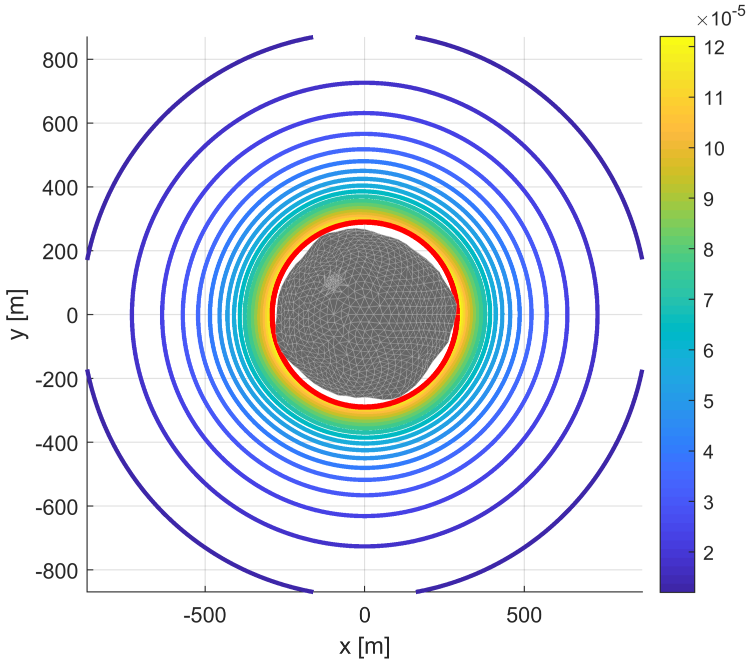


Figure 3.4: Potential gradient magnitude in proximity of Bennu, outside Brillouin sphere, $[\frac{m}{s^2}]$

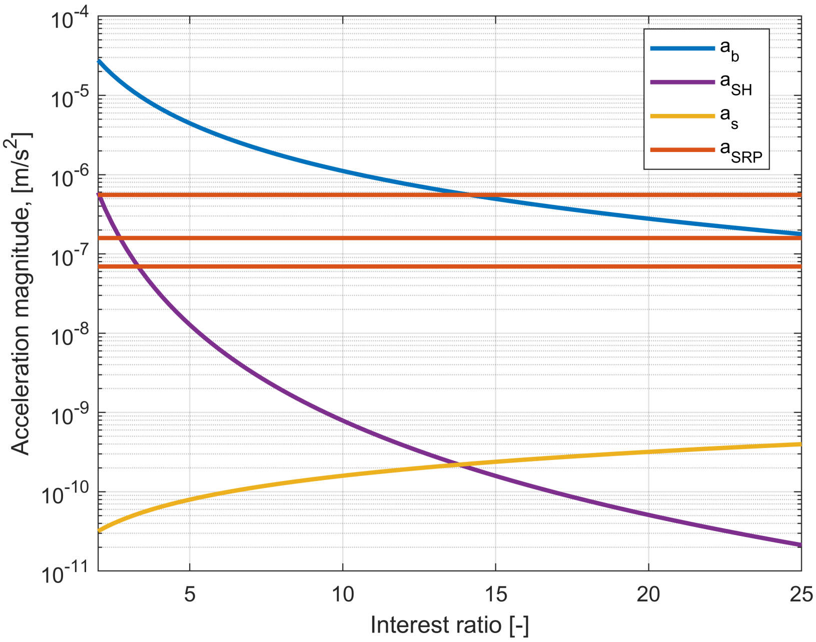


Figure 3.5: Accelerations order of magnitude acting on the spacecraft in proximity of Bennu, at its perihelion

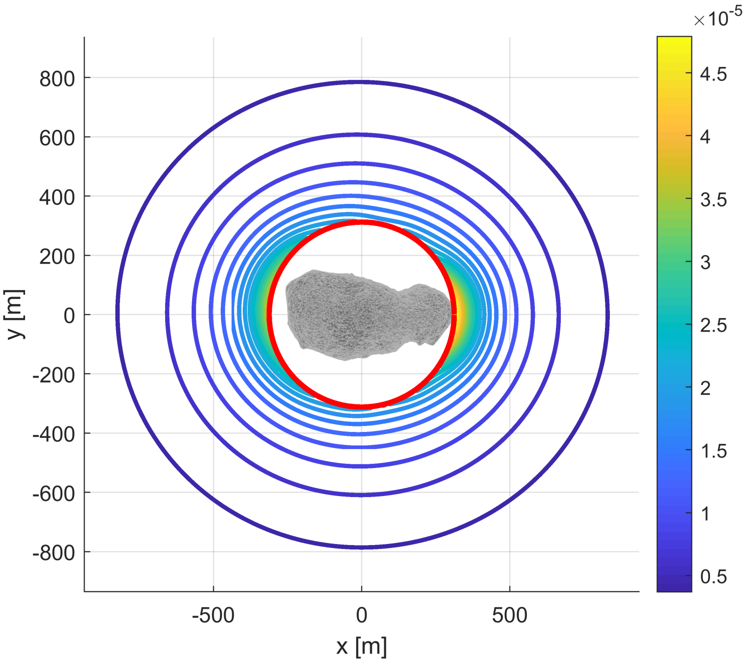


Figure 3.6: Potential gradient magnitude in proximity of Itokawa, outside Brillouin sphere, $[\frac{m}{s^2}]$

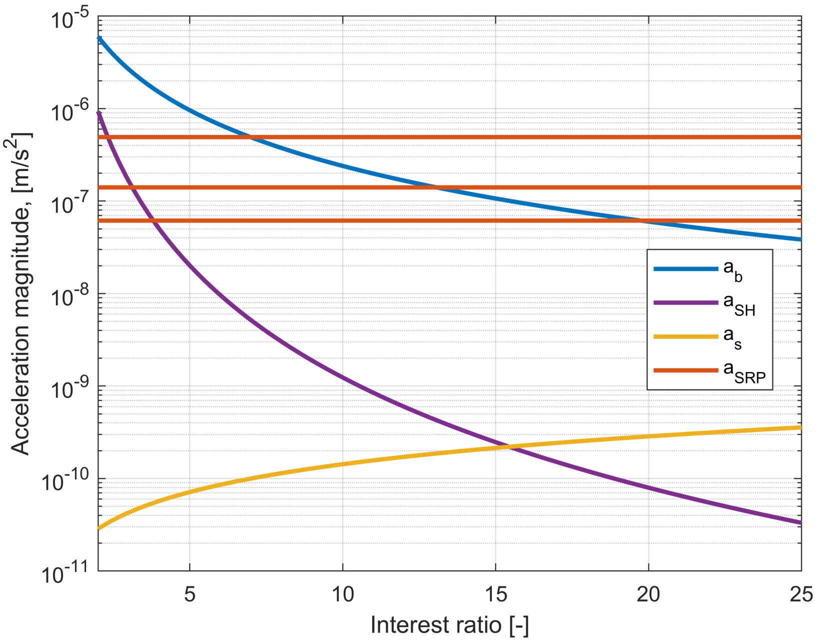


Figure 3.7: Accelerations order of magnitude acting on the spacecraft in proximity of Itokawa, at its perihelion

4 Planning under uncertainty

THE present chapter deals with the problem of planning under uncertainty, providing the general framework under which small bodies autonomous mapping falls. In the robotics field autonomous exploration of an unknown environment is typically formulated with an active SLAM approach, coupling the tasks of mapping, localization and planning. Active SLAM can be seen as an instance of POMDPs. The adaptation of this particular perspective on small bodies exploration is very recent and can be found in literature only in Pesce *et al.* [31] and Chan *et al.* [36] works.

In this chapter, the mathematical formulation of POMDPs is briefly introduced, then the active SLAM problem is presented as a general model for robotic exploration. Finally this model is specifically applied to small bodies autonomous navigation and mapping, highlighting the additional challenges that arise.

4.1 Partially Observable Markov Decision Processes

Markov Decision Processes (MDP) were introduced by Bellman in the 50s. They are based on Markov chains. A Markov chain is a stochastic process with no memory. This means that given some *states*, the process randomly evolves from one state s_k to another s_{k+1} with a transition probability that depends only on the pair (s_k, s_{k+1}) and not on past states.

In a MDP, a decision-maker called *agent* can choose between several possible actions. The transition probability to the next state depends on the chosen action and can be associated to a scalar *reward*. The agent goal is to maximize the rewards over time, with an optimal *policy*. So a MDP is characterized by:

- A state space $\mathcal{S}$.
- An action space $\mathcal{A}$.
- An immediate reward or cost function $r : \mathcal{S}$ x $\mathcal{A} \rightarrow \mathcal{R}$.
- A transition probability $\mathcal{T}(s_{k+1}|\ a_k, s_k)$, that governs the process by mapping a state-action pair to a probability distribution of states at the next time instant.

Therefore a MDP is a four-tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T} \rangle$. A solution to a MDP is a policy which maps from states to a probability distribution over actions $\pi : \mathcal{S} \rightarrow p(\mathcal{A} = a\ |\mathcal{S})$. An optimal policy $\pi^{\star}$ maximizes the reward over time.

The absence of memory in MDPs is defined by the *Markov property*: the next state depends only on the current state and action and not on past actions and states, so the future is conditionally independent of the past, given the present state. This property is essential to many solution algorithms [32]. In real applications the Markov property requirement can be difficult to meet. In order to respect it, the state information must be rich enough so that the observed state transition does not depend on additional historical information. But the agent sensors may not be able to make distinctions between world states. This phenomenon is known as *perceptual aliasing* and can be *involuntary* or *voluntary.* It is involuntary when the sensors can provide only limited or inaccurate information about the environment state. It is voluntary when the agent exploits only part of the sensor information, because of time or resources constraints.

When the state is only partially observable, the problem can be defined as a POMDP. In this case the agent can have only a partial knowledge of the environment: the state is not observable but a signal stochastically related to it is observable. So a POMDP can be described as a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T}, \Omega, \mathcal{O} \rangle$, where:

- $\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T}$ are the same spaces defined for the underlying MDP.
- Ω is the space of possible observations.
- $\mathcal{O}(o_{k+1}|\ a_k, s_{k+1})$ is the probability of making observation o_{k+1} at the next time step, given action a_k that leads to state s_{k+1}.

POMDPs are difficult to be solved, therefore it is better to reduce them in order to find a computationally tractable solution. A POMDP can be reduced to a MDP including the agent history h as internal state. The history is composed by all past actions and observations, so history at time step k will be $h_k =< a_0, o_1, a_1, ..., a_{k-1}, o_k >$. Usually the problem is tackled with a less direct approach known as *belief-space* MDP. This formulation is a tuple $\langle \mathcal{B}, \mathcal{A}, \mathcal{R}_{\mathcal{B}}, \tau \rangle$, where:

- $\mathcal{B}$ is the belief space, with belief $b_k = p(s_k|h_k)$ equal to the probability of being in state s after history h.
- $\mathcal{A}$ is the action space as in the original POMDP.
- r_B is the expected immediate reward $\mathcal{B}$ x $\mathcal{A} \rightarrow \mathcal{R}$
- $\tau(b_{k+1}|\ a_k, b_k)$ is the belief transition function, i.e. the probability of reaching the new belief b_{k+1}, starting from b_k and performing action a_k.

The optimal policy is the one that maximizes the reward in the long term, assuming to act according to that policy:

$$\pi^\star = \underset{\pi}{\operatorname{argmax}} \mathbb{E}_\pi \left[\sum_{k=0}^{T} \mathcal{R}(a_k, b_k) \right] \tag{4.1}$$

This is called also *finite horizon* problem, since reward is optimized for the next T steps of the agent. In case of *infinite horizon* optimality, the reward is maximized over the entire agent lifetime, but a discount factor $\gamma \in [0, 1]$ is considered:

$$\pi^\star = \underset{\pi}{\operatorname{argmax}} \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k \mathcal{R}(a_k, b_k) \right] \tag{4.2}$$

Several approaches exist to solve POMDPs, like methods descending from operations research, point-based algorithms and machine learning [62]–[66]. POMDPs are often computationally intractable and their complexity makes it hard to find a solution to their most general formulation. Exact solutions exist only for a very small class of problems formulated as POMDPs.

4.2 Active Simultaneous Localization And Mapping

SLAM consists in estimating a map of an unknown environment and at the same time localizing in it. Localization is the task of estimating the robot position and orientation (pose) while moving in the environment. This can be done exploiting sensors data of various types (e.g. cameras, laser, sonar), commonly are affected by noise. Moreover as the robot moves, estimation errors are accumulated and this can lead to a divergence from the real pose. The mapping goal is to build a global and consistent map of the environment. For instance, SLAM can take as inputs just visual informations from a monocamera: features can be extracted and matched between different frames and the camera pose can be retrieved. A triangulation of the features is then performed through projective geometry, to reconstruct the 3D map of the environment. So in this case landmarks are features characterized by short descriptor vectors, that can be generated with various techniques [67].

The SLAM problem can be formulated as follows. The environment map at time step k is made of a set of n landmarks $\mathbf{m_k} = \{\mathbf{m^1}, \mathbf{m^2}, \ ... \ , \mathbf{m^n}\}$, where $\mathbf{m^i}$ is the position vector of the i-th landmark. The robot pose $\mathbf{x_k}$ changes while the robot moves under the control $\mathbf{u_k}$ and can be estimated through observations of landmarks location $\mathbf{z_k}$. A scheme is shown in Figure 4.1. The agent actions have stochastic effects and the environment is perceived through noisy and partial observations, therefore SLAM can be described in probabilistic terms. If the robot state was perfectly known, the mapping problem would have been governed by probability:

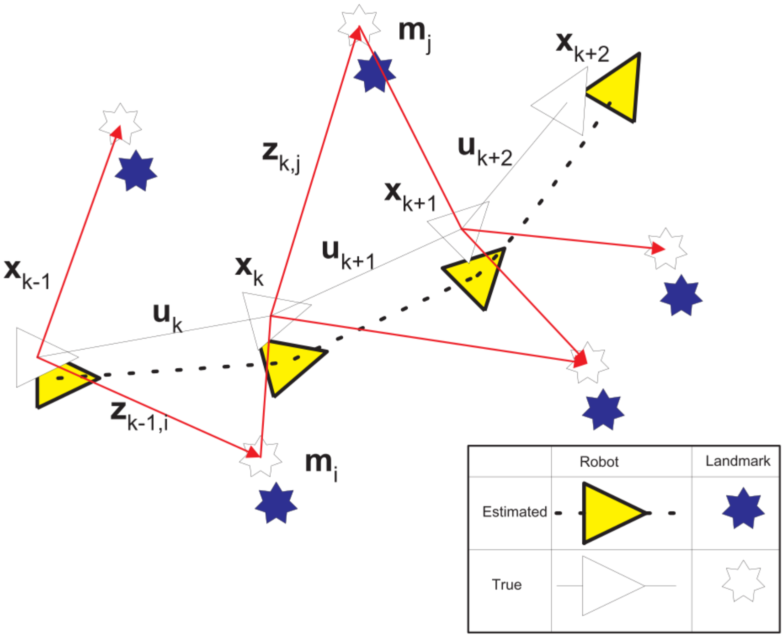


Figure 4.1: Simultaneous Localization And Mapping scheme, from [24]

$$p(\mathbf{m_k}|\mathbf{x_{0:k}}, \mathbf{z_{0:k}}, \mathbf{u_{0:k}}) \tag{4.3}$$

On the contrary, if the map state was known the localization problem would have been:

$$p(\mathbf{x_k}|\mathbf{m_{0:k}}, \mathbf{z_{0:k}}, \mathbf{u_{0:k}}) \tag{4.4}$$

In practice, the two problems are coupled:

$$p(\mathbf{m_k}, \mathbf{x_k}|\mathbf{x_0}, \mathbf{z_{0:k}}, \mathbf{u_{0:k}}) \tag{4.5}$$

There are several approaches to solve the SLAM problem, that can be based on filters [24] or on graph optimization [67]. An important aspect of SLAM structure is that landmarks estimates are highly correlated: even if their absolute position is uncertain, the relative one may be known with a higher accuracy. While the robot moves and observes an already visited landmark $\mathbf{m^i}$, this indirectly affects the estimation of other landmarks $\mathbf{m^j}$ not observed in the current step. In fact the revisited landmark can be updated also with respect to the previous poses, where it was observed along with other landmarks. For this reason SLAM largely benefits from *loop closing*, i.e. when the robot closes its path coming back to a previously visited location.

Active SLAM adds to the SLAM problem the planning task [25], [27], [66]. POMDP provides a framework to investigate the effects of actions and observations on the agent's environment perception, thus allowing to design policies that optimize the agent's interaction with the environment in some of its aspects.

Since the environment is stochastic, the problem can be described in probabilistic terms according to the belief-space MDP formulation presented in the previous section 4.1. The state vector is composed by robot pose and landmark locations $\mathbf{s_k} = (\mathbf{x_k}, \mathbf{m_k})$ and its belief is $\mathbf{b_k} = p(\mathbf{s_k}|\mathbf{z_{0:k}})$. The actions that the agent can take coincide with the control $\mathbf{a_k} = \mathbf{u_k}$. The belief is estimated from past control, belief and current observations: $\mathbf{b_{k+1}} = \tau(\mathbf{u_k}, \mathbf{b_k}, \mathbf{z_{k+1}})$. For what concerns the reward, it is usually modeled in terms of an objective function. For instance, a planner can have multiple objectives like maximizing coverage or map accuracy and minimizing navigation time, motion cost or resources utilization. Several criteria exist to formulate the objective function to be optimized by the planning policy [27], [68]. In particular, the exploration problem consists in choosing the sensing trajectory to obtain the best map.

4.3 Application to space environment

In the present section small bodies navigation and mapping is framed as a POMDP, in analogy with the active SLAM problem. Many similarities can in fact be found between classic robotic and space exploration. In both cases an autonomous robot is exploring an environment that is not completely known and moves in it collecting sensors information. Sensors data are used to localize in the environment and to build a global map of it. On-board resources are limited, both in terms of computational capabilities and energetic resources. Nevertheless, space exploration presents some differences and additional challenges with respect to the classic active SLAM, so the general framework is now tailored on small bodies navigation and mapping.

The relative spacecraft-body state can be defined as $\mathbf{x_k} = (\mathbf{r}_\mathcal{B}, \dot{\mathbf{r}}_\mathcal{B})$ where $\mathbf{r}_\mathcal{B}$ and $\dot{\mathbf{r}}_\mathcal{B}$ are the relative position and velocity defined in Chapter 3. Contrary to classic robotics, natural dynamics plays a fundamental role for the state evolution. In addition, the possibility of controlling the motion is constrained by stringent safety requirements and propellant consumption. The sensors used for navigation around small bodies can be of various types, like radar, lidar and optical. The state belief $\mathbf{b_k}$ can be retrieved with determination algorithms based on filters that may exploit also other types of data, coming from proprioceptive sensors (gyroscopes, accelerometers) or star trackers, Sun sensors. In case of optical measurements, the map of the body $\mathbf{m_k}$ is the set of landmarks on the body surface. Landmarks can be generated in different ways, as manually by human experts, with computer vision algorithms for feature extraction and matching or as centers of SPC maplets. The landmarks knowledge is refined during the mission operations and requires human supervision, as explained in Chapter 2.

Actions represent the ways in which the agent can interact with the environment. In a broad sense, the environment is everything external to the decision maker. Therefore in this case actions can be a change of trajectory or attitude, but also acquisition of sensory inputs or handling of data storage and communication. In principle, there are several objectives towards which the planning goal can be oriented, as optimization of scientific objectives with smart data collection or active SLAM-like autonomous navigation with map accuracy improvement. In practice, it is important to understand which decisions can be taken autonomously without compromising the mission safety. In addition, it has to be considered that spacecraft are complex systems, with which the planning architecture should easily interface. For instance, when planning is used for trajectory design, it should be seen as a high level decision making, letting the task of maneuvering to a suited Guidance Navigation and Control system. As explained in Chapter 3, the dynamical environment in proximity of small bodies is chaotic and errors in maneuvering can lead to impact or escape from the body. In order to solve the planning problem, it is necessary to reduce the policy space by defining proper states and actions, introducing substantial simplifications to the general POMDP formulation, otherwise computationally intractable. Up to now only few possibilities have been studied. In [31] the agent planning is reduced to a fully observable environment and it is oriented to the maximization of mapping accuracy for SPC, choosing at fixed time intervals an orbit within a subset of possibilities and thus generating an optimal orbit sequence. In [36] the agent directly controls the spacecraft for moving around the body and again benefits imaging for SPC. The agent can also decide in which time steps to take a picture and when to downlink data. In this work state and action spaces are wider and some uncertainty is considered for the spacecraft position knowledge.

5 Deep Reinforcement Learning

This chapter introduces the tools used to tackle the autonomous small bodies mapping problem. First, RL is introduced as a method to solve decision making problems, with particular attention to Q-learning. In addition, advantages in using DRL are briefly discussed. Then, basic knowledge is provided about neural networks, since they are involved in DRL algorithms. Finally, the two DRL techniques of NFQ and DQN are described in more detail.

5.1 Reinforcement Learning

Given the discrete nature of MDPs, usually a closed form solution to the problem does not exist. RL is one of the most common methods to solve MDP problems [32]. This powerful approach can be employed to solve large sequential decision making problems in both fully observable and partially observable MDPs. Policies may also be found with evolutionary methods, but this solution is feasible only when the search space is small, otherwise the optimization becomes computationally intractable.

The idea behind RL is to find a good policy by letting the agent interact with the environment and collecting experiences. A scheme is shown in Figure 5.1. The agent can understand thanks to the received reward if an action is valuable or not when being in a certain state. The agent's goal is to learn a control policy that maximizes the discounted sum of rewards over time, i.e. the *return*:

$$R_k = \sum_{k=0}^{\infty} \gamma^k r_k \tag{5.1}$$

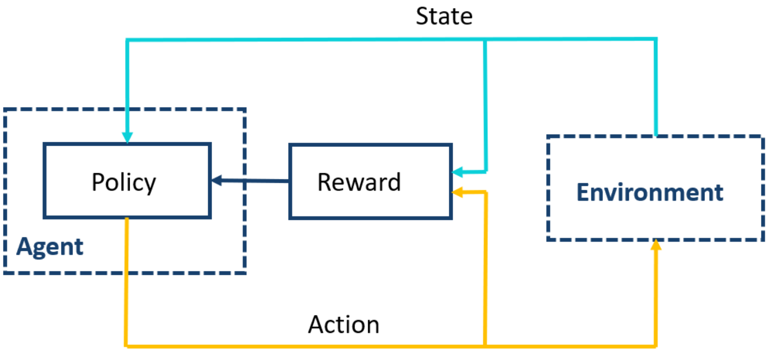


Figure 5.1: Reinforcement Learning scheme

where r_k is the immediate reward at time step k and γ is the discount factor that ranges in the interval [0,1]. If γ has a low value the immediate reward is more valued; on the contrary large values give more importance to the future rewards. Typical values used in RL algorithms are in the range $0.95 - 0.99$.

The *value function* $V_\pi(s)$ of a state s under the policy π is defined as the expected return starting from s and following π thereafter:

$$V_\pi(s) = \mathbb{E}_\pi\Big[\sum_{k=0}^{\infty} \gamma^k r_k \; | s_0 = s \;\Big] \tag{5.2}$$

Thanks to recursive relationships, it is possible to derive Bellman's equation:

$$V_\pi(s) = \sum_a \pi(a|s) \sum_{s'} p(s'|s,a) \Big[r + \gamma V_\pi(s')\Big] = \mathbb{E}_\pi \Big[r + \gamma V_\pi(s')\Big] \tag{5.3}$$

where s' is the possible next state. The optimal policy is therefore:

$$\pi^\star = \underset{\pi}{\operatorname{argmax}} \, V_\pi(s) \tag{5.4}$$

The corresponding optimal value function is:

$$V^\star(s) = \max_\pi V_\pi(s) \tag{5.5}$$

Literature about RL algorithm is really wide [32], [69]. RL approaches can be organized into two main categories: policy and value-based methods. The former searches the space of behaviors in order to find a policy that performs well in the environment; the latter estimates the utility of taking actions in given states of the world.

Popular methods are Temporal Difference methods (Q-learning), Monte Carlo methods and Dynamic Programming (policy iteration, value iteration). Every method has different advantages and disadvantages and the proper choice depends on the specific RL problem. In particular, the nature and dimension of state and action spaces are critical aspects. The optimal policy is usually not found, but a good solution is anyway obtained.

Some memory approaches exist for RL in non-Markovian domains, some of them including the history as a state or exploiting recurrent neural networks to keep memory of past states [64], [70].

When dealing with multiple objective reinforcement learning, there is more than one objective to optimize simultaneously. Two main strategies exist: singe-policy or multiple-policy. In case of single-policy, the aim is to design a single policy that simultaneously satisfies the presence of multiple objectives. So the problem is to design an objective function that represents all the preferences. The most common methods are the weighted sum approach, W-learning and analytic hierarchic process (AHP) [71]. Instead, in the Convex Hull approach optimal value-functions or policies for the single objectives are learned at once, without requiring to know the relative priorities over reward components [72].

5.1.1 Q-learning

Q-learning is a widely used value-based RL technique in which action-state couples are considered. This method can solve only MPDs with limited and discretized state and action spaces. Q-learning is based on the estimation of the *Q-value function*, also called *action-value function*.

For a given policy π, initial state s and action a, the Q-value function is the expected return, given that the decision process is governed by policy π:

$$Q_\pi(s,a) = \mathbb{E}_\pi\Big[\sum_{k=0}^{\infty}\gamma^k r_k \mid s_0 = s, a_0 = a\Big] \tag{5.6}$$

Again, exploiting recursive relationships:

$$Q_\pi(s,a) = \sum_{s'} p(s'|s,a)\Big[r + \gamma\sum_{a'}\pi(a'|s')Q_\pi(s',a')\Big] = R_t + \gamma\mathbb{E}_\pi\Big[Q_\pi(s',a')\Big] \tag{5.7}$$

The optimal policy is the one that maximizes the action value function:

$$\pi^\star = \underset{\pi}{\operatorname{argmax}}\, Q_\pi(s,a) \tag{5.8}$$

So the optimal Q-value $Q^\star(s,a)$ corresponds to the sum of expected discounted future rewards assuming that the agent acts optimally.

$$Q^\star(s,a) = \max_\pi Q_\pi(s,a) \tag{5.9}$$

In classical Q-learning the action-value function can be estimated as:

$$Q_{k+1} = (1-\alpha)Q_k(s,a) + \alpha\Big[r(s,a) + \gamma \max_b Q_k(s',b)\Big] \tag{5.10}$$

where α is a learning rate. Convergence of Q-learning has been proved by Watkins *et al.* [73].

5.2 Deep Reinforcement Learning

Problems with continuous action and state spaces are particularly difficult to be solved. If the state space is large, exploring all the states to find the optimal policy or optimal value function is computationally intractable. Artificial Neural Networks (ANN) overcome this issue by generalizing from previous encounters with different states, similar to the current one. In DRL algorithms neural networks are used as a function approximator within a RL algorithm. Extending reinforcement learning to function approximation also makes it applicable to partially observable problems, in which the full state is not available to the agent. DLR has shown excellent results, in particular the two main fields of application are video-gaming and control problems.

In analogy with RL, two main branches can be found in DRL: policy search and value-based methods [69], [74]. Policy search aims to directly find the optimal policy with gradient-based or gradient-free methods. Value based methods (Deep Q-learning) are less direct and exploit the neural network to approximate the q-value function, instead of directly the policy. When the number of states of the environment is very large or even infinite, it is not possible to experience them all and multiple times, keeping track of all the Q-value estimates. In such cases the policy can be represented by a neural network. Neural networks ability to generalize complex and non-linear behaviors turns out to be of particular use in this situation: the program can select an action, based on informations saved for similar states in past experience, without the need of experiencing all the possible states.

Anyway, continuous action spaces are not well handled by DRL. The most straightforward solution is to discretize the action space, but the number of actions increases exponentially with the number of degrees of freedom. Recent advances extend actor-critic methods that combine policy and value-based to the continuous action domain [75].

5.3 Neural Networks

ANN are numerical tools inspired to the structure of biological neural networks and capable of performing an input-output mapping after a process called *training*. They were born at the end of the 50' with Rosenblatt's perceptron and since then several types of neural networks have been developed, with a large variety of tasks (function approximation, pattern association, pattern recognition, control), architectures and training algorithms [65], [76], [77]. The training methods can be of various types and three main categories exist: supervised, unsupervised and reinforcement learning. Here the focus is on the last method. In particular, in DRL algorithms ANN are used to estimate the value-function, so with the task of function approximation.

In this section the classical Multilayer Perceptron (MLP) is described, along with back-propagation training and empirical rules on ANN use.

5.3.1 Multilayer Perceptron

MLP are one of the most common types of ANN. Their structure is made of some units called *neurons*, disposed in *layers* and connected to each other through linear or non-linear relationships.

The network input, also called *input layer*, is a vector $\mathbf{x}$ of dimensions $[R \times 1]$. A scheme of the i-th neuron of the net first layer is provided in Figure 5.2. The weighted sum of neuron inputs is:

$$v_i = \sum_{j=1}^{R} w_{ij} x_j + b_i \tag{5.11}$$

where w_{ij} is the weight from neuron j to neuron i and b_i the bias. The neuron output is:

$$y_i = f_i(v_i) \tag{5.12}$$

where f_i is the *activation function*. Typical activation functions are Heaviside function, logistic sigmoid, hyperbolic tangent or a linear function. Considering the whole

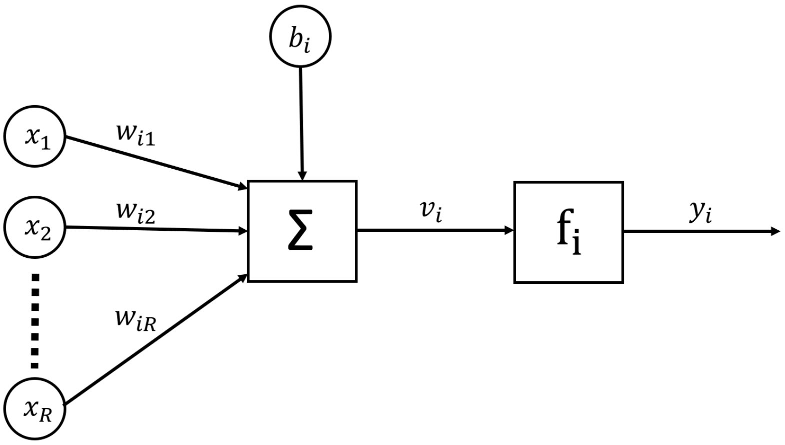


Figure 5.2: Neuron scheme

first layer, made of R^1 neurons, the input-output relationship can be written in matrix form:

$$\mathbf{y^1} = \mathbf{f^1}(\mathbf{W^1}\mathbf{x} + \mathbf{b^1}) \tag{5.13}$$

where $\mathbf{y^1} = [y_1, y_2, \ldots, y_{R^1}]^{\mathrm{T}}$ is the first layer output, $\mathbf{f^1} = [f_1, f_2, \ldots, f_{R^1}]^{\mathrm{T}}$ collects the activation functions and $\mathbf{b^1} = [b_1, b_2, \ldots, b_{R^1}]^{\mathrm{T}}$ the biases. $\mathbf{W^1}$ is the weights matrix of the first layer, of dimensions $[R^1 \times R]$. A scheme of MLP with M layers is shown in Figure 5.3, where the apex always stands for the layer number. The last layer (M) is called *output layer*, while all the layers (1,2,..., M-1) in between the input and output ones are called *hidden layers*. The final output of the MLP is vector $\mathbf{y^M}$ of dimensions $[R^M \times 1]$, also shortly named $\mathbf{y}$. Let's also define $\mathbf{y^0} = \mathbf{x}$. Input sum and output of the i-th neuron of layer m are:

$$v_i^m = \sum_{j=1}^{R^{m-1}} w_{ij}^m y_j^{m-1} + b_i^m \tag{5.14}$$

$$y_i^m = f_i^m(v_i^m) \tag{5.15}$$

Activation functions, number of layers and neurons are the ANN hyperparameters. Their choice is usually made by the designer, but optimization procedures are also possible.

5.3.2 Training: the back-propagation algorithm

The most common strategy for supervised learning is the back-propagation algorithm now explained [76], [77]. Before starting the training, the MLP weights and

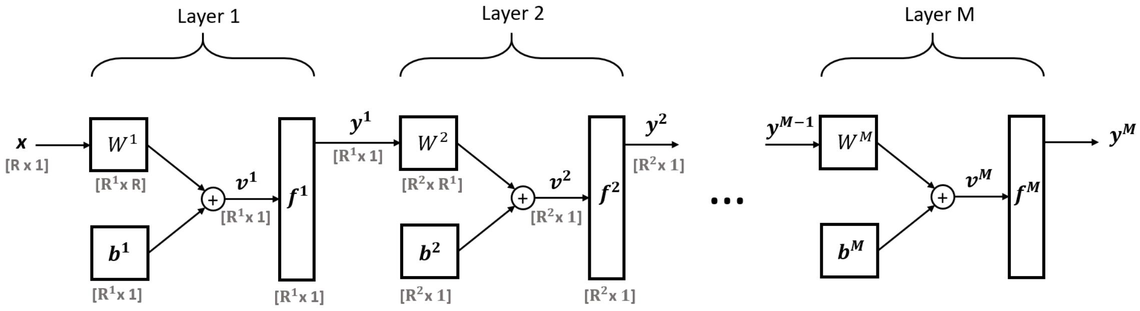


Figure 5.3: Neural network scheme, matrix notation

biases are initialized to some random values, so at the beginning the network incorrectly maps inputs to outputs. During training the parameters are adapted according to an error function until the ANN has learned to correctly perform its task. This is possible thanks to the exploitation of a *training set* $\mathcal{T}$, i.e. a set of inputs associated to the corresponding desired outputs:

$$\mathcal{T} = \{\mathbf{x}(n), \mathbf{t}(n)\}_{n=1}^{N} \tag{5.16}$$

where $\mathbf{x}(n)$ is the n-th input and $\mathbf{t}(n)$ the corresponding n-th target. The error between net output and desired target can be defined as a quadratic function:

$$\mathcal{E} = \frac{1}{2}(\mathbf{t}(n) - \mathbf{y}(n))^T(\mathbf{t}(n) - \mathbf{y}(n)) \tag{5.17}$$

At each k-th training step, weights and biases can be updated with *gradient - descent*:

$$w_{ij}^m(k+1) = w_{ij}^m(k) + \Delta w_{ij}^m(k) = w_{ij}^m(k) - \eta \frac{\partial \mathcal{E}}{\partial w_{ij}^m}(k) \tag{5.18}$$

$$b_i^m(k+1) = b_i^m(k) + \Delta b_i^m(k) = b_i^m(k) - \eta \frac{\partial \mathcal{E}}{\partial b_i^m}(k) \tag{5.19}$$

The *learning rate* η scales the derivative and therefore has an important role for the convergence of the algorithm: if its value is too small, the learning converges too slowly, if it is too high than it causes oscillations and instability problems. In adaptive methods, the learning rate is modified according to the observed behavior of the error function.

One training step is composed by *forward phase* and *backward phase*. During the forward phase, the input is propagated through all the network layers:

$$\mathbf{y^{m+1}} = \mathbf{f^{m+1}}(\mathbf{W^{m+1}}\mathbf{y^{m}} + \mathbf{b^{m+1}}) \tag{5.20}$$

for $m = 0, 1, \ldots, M-1$. In the backward phase the output error is computed and propagated backward in order to calculate the gradients. Recalling equations 5.17 and 5.14:

$$\frac{\partial \mathcal{E}}{\partial w_{ij}^m} = \frac{\partial \mathcal{E}}{\partial v_i^m}\frac{\partial v_i^m}{\partial w_{ij}^m} = \frac{\partial \mathcal{E}}{\partial v_i^m} y_j^{m-1} = s_i^m y_j^{m-1} \tag{5.21}$$

$$\frac{\partial \mathcal{E}}{\partial b_i^m} = \frac{\partial \mathcal{E}}{\partial v_i^m}\frac{\partial v_i^m}{\partial b_i^m} = \frac{\partial \mathcal{E}}{\partial v_i^m} = s_i^m \tag{5.22}$$

where s_i^m are the elements of the sensitivity vectors $\mathbf{s^m}$. For the output layer the computation is straightforward:

$$\mathbf{s^M} = -\mathbf{\dot{F}^M}(\mathbf{v^M})[\mathbf{t} - \mathbf{y}] \tag{5.23}$$

where

$$\mathbf{\dot{F}^M}(\mathbf{v^M}) = \begin{bmatrix} \dot{f}_1^M(v_1^M) & 0 & \ldots & 0 \\ 0 & \dot{f}_2^M(v_2^M) & & \\ \vdots & & \ddots & \\ 0 & & & \dot{f}s_{R^M}^M(v_{R^M}^M) \end{bmatrix} \tag{5.24}$$

Sensitivities of the other layers can be computed exploiting back-propagation from the output layer to input one. Such relation can be easily derived using equations 5.14 and 5.15:

$$\mathbf{s^m} = \frac{\partial \mathcal{E}}{\partial \mathbf{v^m}} = \left[\frac{\partial \mathbf{v^{m+1}}}{\partial \mathbf{v^m}}\right]^{\mathrm{T}} \frac{\partial \mathcal{E}}{\partial \mathbf{v^{m+1}}} = \mathbf{\dot{F}^m}(\mathbf{v^m})\left[\mathbf{W^{m+1}}\right]^{\mathrm{T}} \mathbf{s^{m+1}} \tag{5.25}$$

for $m = M-1, \ldots, 2, 1$. Finally the layers weights and biases are updated according to the gradient-descent equations 5.18 and 5.19.

It is important to interrupt the training at the proper moment. In fact, in case of early stopping the net does not completely learn the input-output mapping. On the contrary, if training is stopped too late, data are overfiedtted and the network lacks in generalizing capabilities. The training can be interrupted when the gradient vector falls below a tolerance value, because this means that the gradient descent has almost reached a minimum (although if it could be stuck in a local minimum). Another criterion is *validation.* The validation set is part of an already known data set, similarly to the training set, but not used for training. During training such set can be exploited to compute the output error; the algorithm is stopped as soon as the validation error reaches the minimum, avoiding overfitting. In practice, a technique that can be used is to stop the training if the validation error has increased for a certain amount of time since the last time it decreased.

The training process described up to now is called *incremental* training, since the network is trained sequentially feeding the inputs. On the contrary, in *batch* training the total data set is considered at once. Mini-batches are an intermediate way between the two. When using batches or mini-batches, the training algorithm differs in the definition of the error function. In particular, $\mathcal{E}$ is the mean error over the training mini-batch $\mathcal{T}_B \in \mathcal{T}$:

$$\mathcal{T}_B = \{\mathbf{x}(n), \mathbf{t}(n)\}_{n=1}^{N_B} \tag{5.26}$$

The case in which $N_B = N$ corresponds to the complete batch. The error becomes:

$$\mathcal{E} = \frac{1}{2N_B} \sum_{n=1}^{N_B} (\mathbf{t}(n) - \mathbf{y}(n))^T (\mathbf{t}(n) - \mathbf{y}(n)) \tag{5.27}$$

The gradient of the mean square error used for gradient-descent will simply be the mean of individual squared errors gradients. While batch training is more stable, it requires larger computational resources and if new data are added to the training set it is necessary to start the learning process from scratch. For what concerns incremental learning, it is more suited when reduced computational resources are available, but it strongly depends on the learning rate and is more sensible to outliers.

5.3.3 Normalizing the inputs

Normalizing the inputs can significantly improve the training performance. A procedure for normalizing the neural network inputs has been developed by LeCun *et al.* [78], in which all the inputs to the neural network are normalized to have a zero mean value over the training set. The reason behind is that if the weights have different means, their update will be more difficult during the learning process. For instance in the extreme case in which the means are all positive, the weights of neurons in the first hidden layer can only increase or decrease together. So the first step to be performed is to shift the inputs so that the mean of each of them is zero over the training set. Moreover the inputs should be uncorrelated, if possible. The second step is therefore to remove linear correlation; principal components analysis can be exploited. Finally, the inputs should have similar covariances. If a sigmoidal activation function is employed, the standard deviation value should be equal to 1. The steps are resumed in Figure 5.4.

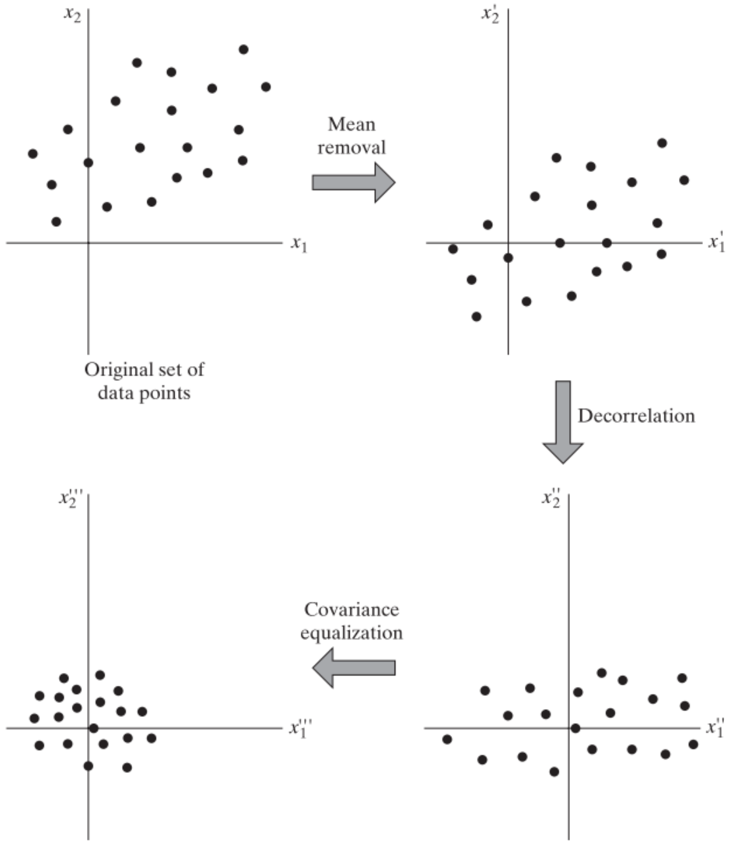


Figure 5.4: Neural network inputs normalization procedure, from [76]

5.4 Deep Q Learning algorithms

Now that both RL and ANN have been introduced, two of the most relevant Deep Q Learning algorithms are presented: NFQ and DQN. Such algorithms are here considered because of their popularity and encouraging results. It has to be noticed that a large variety of DRL algorithms (policy-based, actor-critic) could be suitable for tackling small bodies mapping, since they all present the advantages of handling large continuous state spaces, generalizing capabilities and can deal with partially observable environments. Here only NFQ and DQN are considered. NFQ was one of the first Deep Q Learning algorithms and inspired many of the successive developments. It is selected because of its relative simplicity and good documentation available. DQN is more recent and has been a significant breakthrough in the DRL research field; it presents some similarities with NFQ, but it introduces innovative mechanisms that can lead to better results.

5.4.1 Neural Fitted Q Iteration

NFQ has been developed in 2005 by Reidmiller [79].

Experiences are collected by letting the agent interact with the environment following a random policy. Triples of the form (s, a, s') are stored in a sample set $\mathcal{D}$, on which the net is trained off-line. The neural network is initialized with random parameters θ_0, then the sample set is completely swept for a certain number of iterations. At every iteration step, the net is used to estimate the Q-value. NFQ is based on Algorithm 1, with relative scheme in Figure 5.5:

Algorithm 1 Neural Fitted Q Algorithm

```
collect E experiences and store in $\mathcal{D}$
procedure NFQ
    $k = 0$
    initialize net: $\rightarrow Q_0 = Q(s, a|\theta_0)$
    while $k < N$ do
        compute target: $t_i = r(s_i, a_i, s'_i) + \gamma \max_b Q_k(s'_i, b)$
        network input: $x_i = (s_i, a_i)$
        generate pattern set: $P = [x_i, t_i], \quad i = 1 : E$
        train net: $Q_{k+1} = \text{train}(P)$
        $k = k + 1$
    end while
end procedure
```

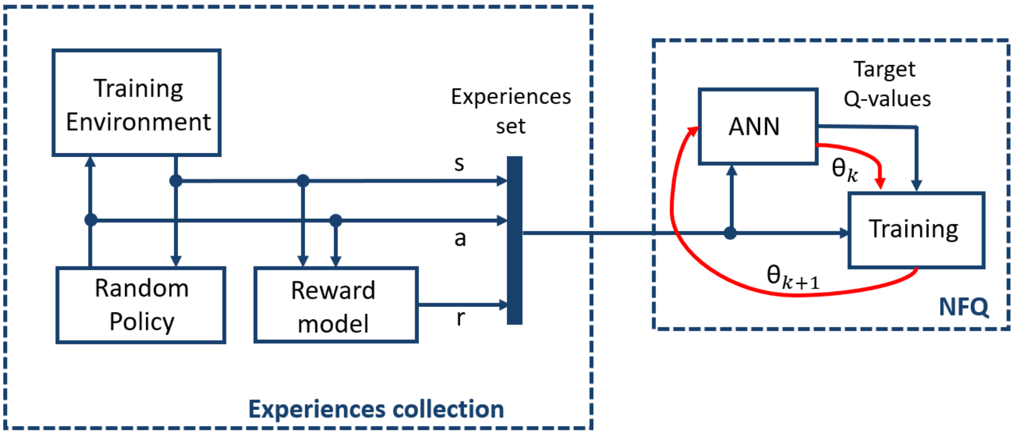


Figure 5.5: NFQ algorithm scheme

The training algorithm commonly used for NFQ is the resilient back-propagation (RPROP), by Riedmiller [80], in which the weight is adapted based on local gradient information. This direct adaptive method is robust against the choice of its initial parameter and converges with a reduced number of learning steps with respect to classical gradient-descent. Therefore it is well suited to be used in a reinforcement learning framework. A detailed description of the algorithm is reported in Appendix A.

The main difficulty in applying NFQ is the training data collection: the problem must be suited to be solved with a random policy, that can allow the agent-environment interaction and the collection of the state-action-state triples. If with a random policy the environment exploration is not sufficient, the net will be trained only on a subset of the different situations it could encounter. Another disadvantage of this method is that it involves the repeated training of the network on hundreds of iterations, so is not convenient to be used for very large networks. On the other hand, this approach is model-free, stable, data efficient and simple to implement.

Some variants of the algorithm exist. In [81] a heuristic dynamic scaling of the network output is proposed: the Q function assumes values that are unknown and the neural network output will be within the interval $I = (0, 1)$. Thanks to the dynamic scaling, the interval $(0, 1)$ is exploited at best and the need to design reward to have Q-values comprised in that interval is overcome. In [82] a *growing batch* variant is proposed: the transitions are collected with a random policy at the beginning of the algorithm and after a certain number of iterations transitions are collected again exploiting the trained policy. This approach helps to collect transitions that are more and relevant as the policy performance increases.

5.4.2 Deep Q Network

Mnih *et al.* [83] have proposed a new DRL algorithm, that successfully overcame the performance of human experts and other reinforcement learning algorithms in many Atari games.

Experiences are collected playing many episodes, during which actions are chosen according to an $\epsilon - greedy$ policy. This means that with probability ϵ the action is random and with probability $(1-\epsilon)$ the action is the one that maximizes the current Q-function. Usually the ϵ value is linearly varied between 1 and 0 during the learning, in order to exploit the Q-function only when the net starts to approximate it well, so the agent will spend more and more time in exploring only the relevant parts of the environment. The choice of the greedy parameter can be critical to correctly collect transitions, as also exploration of unknown regions of the state space is important.

Two new mechanisms are introduced with respect to the NQF method: the *target network* and *experience replay.*

- The so called *experience replay* consists in storing the agents experiences at each time-step in a data set $\mathcal{D}$, that is pooled over many episodes into a replay memory. In other words, the Q-learning updates are applied over the random experiences sampled from $\mathcal{D}$. As in the NFQ case, the samples are randomized to break correlations in the collected data. This mechanism allows not to forget past experiences during the learning.

- The *target network* is an additional network that is used for approximating the Q-value, while actions are taken according to the network that is undergoing the training. At first the two networks are equally initialized, then after every C steps of the algorithm the target net is updated and taken equal to the trained net. The main purpose of this mechanism is to stabilize the learning algorithm.

DQN is based on Algorithm 2, with scheme in Figure 5.6:

Algorithm 2 Deep Q Network Algorithm

```
Initialize replay memory $\mathcal{D}$ to capacity E
Initialize net: $\rightarrow Q_0 = Q(s, a|\theta_0)$
Initialize target net: $\rightarrow \hat{Q}_0 = Q(s, a|\theta^- = \theta_0)$
for episode = 1,M do
    Initialize sequence: $s_1$
    for k = 1,T do
        with probability $\epsilon$ select action $a_i$
        otherwise $a_i = \text{argmax}_a Q_k(s_i, a|\theta)$
        observe reward $r_i$ and new state $s'_i$
        store transition $(s_i, a_i, s'_i)$ in $\mathcal{D}$
        compute target: $t_i = r(s_i, a_i, s'_i) + \gamma \max_b \hat{Q}(s'_i, b|\theta^-)$
        network input: $x_i = (s_i, a_i)$
        sample random minibatch of transitions P from $\mathcal{D}$
        $P = [x_i, t_i], \quad i = 1 : N$
        train net: $Q_{k+1} = \text{train}(P)$
        $k = k + 1$
        every C steps reset: $\hat{Q} = Q_k$
    end for
end for
```

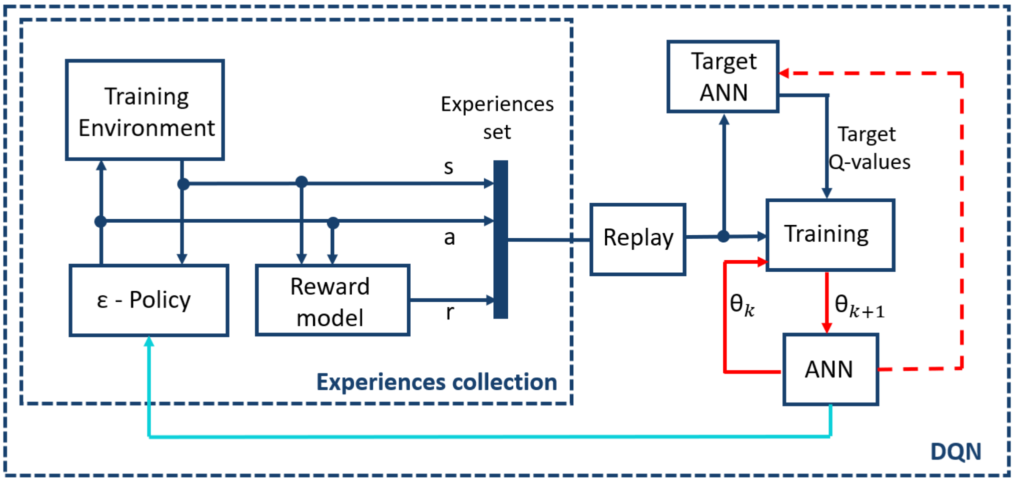


Figure 5.6: DQN algorithm scheme

One of the main reasons for the creation of DQN has been the need of fastening the training process, particularly useful when training large networks. Nevertheless, DQN can be used also when dealing with smaller networks as can lead to a better performance in addition to the quicker training. The algorithm is data efficient, as each experience is potentially used multiple times to train the network.

6 Planning architecture for small bodies imaging

THE planning framework defined in chapter 4 presents a very general point of view on small bodies exploration. In practice it needs to be reduced introducing some assumptions, to be computationally tractable and actually useful for future applications. As already pointed out, such general formulation of small bodies navigation and mapping as a POMDP is recent and not yet widely studied in literature. The few related works solve the POMDP problem with different planning objectives, approaches and assumptions [31], [36]. This thesis can be located under the same general framework, but presents a novel planning architecture in which the focus is moved on image collection timing. The proposed architecture allows an efficient small body imaging, decoupling the decision process from the spacecraft dynamics, thus eliminating safety issues and keeping the algorithm mission-independent.

In this chapter planning architecture is described, along with modeling choices and assumptions, that derive from the critical analyses of context and tools carried out in previous chapters. Then, the presented architecture is detailed through the appropriate definition of rewards, actions and states.

6.1 Planning architecture overview

As explained in Chapter 4, exploration can be seen as an instance of a continuous states and actions POMDP:

$$\pi^\star = \underset{\pi}{\operatorname{argmax}}\, \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k \mathcal{R}(\mathbf{a_k}, \mathbf{b_k}) \right] \tag{6.1}$$

where $\mathbf{b_k}$ is the belief of spacecraft and map states ($\mathbf{x_k}$, $\mathbf{m_k}$), $\mathbf{a_k}$ the actions performed by the agent following policy π and $\mathcal{R}$ the reward that models the planner objectives. Here a reduced policy space is considered in order to make the problem computationally tractable. In particular, the planning goal is to choose sensing locations to better the map knowledge, collecting images with good characteristics for enhancing the shape model reconstruction process.

Figure 6.1 shows the scheme of the proposed architecture. The planning architecture is designed to be mission-independent and computationally light to cope with limited on-board resources. In particular, the key infos needed by the algorithm are the camera characteristics, the relative pose between camera and target, the illumination conditions and a rough body geometry. Then, such data are preprocessed along with history information of already collected images. The next block is related with the autonomous decision making: if the current observation epoch is worth, then an image is taken. In particular, DRL is exploited to design the planning policy, comparing two different techniques: NFQ and DQN. To build up a successful policy, prior knowledge needs to be incorporated in the process as much as possible, having as an effect the simplification of the neural network task. It is in fact well known that neural networks perform better when their structure is reduced. Once the inputs to the neural network have been computed, they are fed into the net, that outputs the estimated Q-value. At this point, the policy simply selects the action with largest Q-value. Please note that such a policy is not random, but given one input the output will be always the same. Finally, actions are recorded along with conditions under which images were taken. Such data are important to understand how the mapping campaign is proceeding.

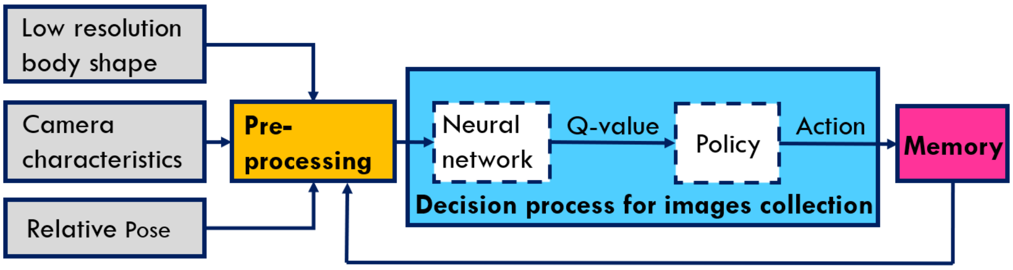


Figure 6.1: On-board algorithm scheme

Thanks to the generalizing capabilities of neural networks and to the proposed problem formulation, the presented algorithm can be trained on-ground, with no retraining on-board. The training is performed with rewards formulated to improve the on-ground SPC and to simultaneously limit the amount of collected images.

This planning framework is different form other research works under many aspects. The here proposed architecture differences from [31] because of different planning goals, less restrictive assumptions and solution method. In fact [31] mainly deals with orbit selection and reduces the environment to completely observable: the planner evaluates reward for a subset of orbits and picks the best. Observations are considered equally spaced on the orbits. It is also interesting to see differences and similarities between the present thesis and Chan's work [36], since they are two independent and contemporary works, carried out almost in parallel: a similar solution approach (DRL) is adopted, but the planning goal and model definition are different. In [36] the neural network tasks include not only observations timing, but also downlink and maneuvering: these complex objectives make it difficult and risky to integrate such an architecture on-board. In particular, learning maneuvering introduces issues related to the modeling of the dynamical environment, which are not tackled. Here a safer planning is proposed, that has a large flexibility and generality. The results of this thesis are supported by extensive tests, not carried out in the other works, and the presented architecture is applicable to a wide variety of scenarios.

A detailed definition of the here proposed reduced spaces of the decision process is presented in the next sections, in terms of environment, reward, states and actions.

6.2 Environment model

The environment here considered is *model-based*, since a model to describe it is available. This model includes all the elements relevant for the mapping campaign, highlighted in Chapter 2. In particular, to derive images acquisition conditions and body coverage, the following elements need to be considered:

- Body geometry.
- Sun direction, relative to the body.
- Camera position and orientation, relative to the body.
- Camera FOV.
- History of collected images conditions.

The body geometry is here modeled with a low resolution polyhedron, made up of triangular facets. Each facet is characterized by a normal unit vector $\hat{\mathbf{n}}$, with respect to which photometric angles defined in Chapter 2 can be computed (please refer to Figure 2.5).

The spacecraft and Sun trajectories relative to the body surface are modeled as a set of discrete position vectors. To make an accurate and long term dynamic model of the asteroid approach is beyond the scope of this thesis. The considered small bodies are assumed on a keplerian orbit around the Sun and uniformly rotating about their principal inertia axis. For what concerns the spacecraft dynamics, only the central body attraction is considered (two body problem assumption). The adopted simplified model does not want to represent all possible dynamics environments, that are countless, as explained in Chapter 3. All the considerations made in Chapter 3 make evident that more complex models would for sure be more realistic, but not more representative, since they would fit only for a very restricted class of mission scenarios. It is therefore believed more useful to adopt here a simple model, but to structure the planning architecture to be effective whatever the relative dynamics between spacecraft and small body. For all these reasons a simplified dynamic model is more convenient and sufficiently realistic to make considerations on image collection planning.

The camera is assumed always pointed towards the body center, without attitude changes. The camera FOV is simplified as conical.

6.3 Reward model

To appropriately model the reward is a delicate task, because the final policy will maximize what the designer has indicated as valuable: only if the reward is well defined the policy will be able to actually achieve the desired results. Several objectives are accounted for in the reward definition. In particular, the main goals to be achieved are:

- High quality and global map of the small body.
- Fastening of the mapping process.
- Reduction of the amount of data to be downlinked.

The first goal depends on the adopted mapping technique. In the present thesis, reward is defined to ease SPC, but if a different shape reconstruction technique is used, different scores can be defined without changing the overall procedure. Please note that some objective are in contrast and need to be balanced: a planner that maximizes the coverage would explore new areas, while a planner that maximizes the map quality would spend more time on the same areas. In addition, some constraints are present on the number of images that can be taken, because of memory and telecommunication issues. Therefore, only pictures containing relevant information should be collected and sent to ground.

In this section the facet score and facet mapping index used to model the map quality are described. Then, the overall reward is defined, including all the above described objectives.

6.3.1 Facet score and facet mapping index

Given a set of images that include a surface portion of the body, a score can be defined to assess the mapping quality for that area. The goodness of the mapping depends mainly on three factors: the illumination conditions, the camera poses and the surface topology. As detailed in Chapter 2, SPC benefits from images with large variations in illumination and small variations in view angle. So the same surface portion, i.e. the same facet of the representative shape model, should be observed several times under the proper conditions, that are now modeled. The map quality is represented by means of a low resolution shape model, in which a score S^i is associated to each facet i at the considered time instant. This score is the result of the combination of five contributions: incidence, emission, emission variation, solar azimuth angle and spacecraft azimuth angle scores. For the photometric angles definition please refer to Chapter 2. The scores for SPC here defined rely on previous works [31], with some slight changes.

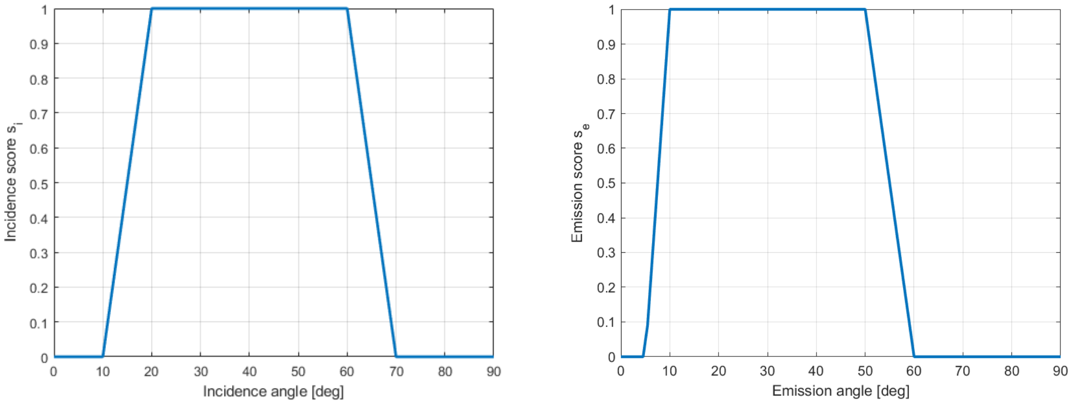
Incidence score The incidence angle i should be kept between $20° - 60°$ to avoid shadows and excessive brightness, that won't allow the extraction of useful information. Let's define the incidence score S_i^i:

$$s_i = \begin{cases} 1 & \text{if } 20° \leq i \leq 60° \\ \frac{1}{10}i - 1 & \text{if } 10° \leq i \leq 20° \\ -\frac{1}{10}i + 7 & \text{if } 60° \leq i \leq 70° \\ 0 & \text{otherwise} \end{cases} \tag{6.2}$$

$$S_i^i = \mu_j(s_i) \tag{6.3}$$

where μ_j is the mean performed over all the n taken pictures that contain the facet.

$$\mu_j(x) = \frac{1}{n}\sum_{j=1}^{n}(x_j) \tag{6.4}$$



(a) Incidence Score

(b) Emission Score

Figure 6.2: Incidence and emission scores trend

Emission score The emission angle should be kept between 10° – 50°. So in a similar manner the emission score S_e^i is defined as follows:

$$s_e = \begin{cases} 1 & \text{if } 10° \leq e \leq 50° \\ \frac{1}{5}e - 1 & \text{if } 5° \leq e \leq 10° \\ -\frac{1}{10}e + 6 & \text{if } 50° \leq e \leq 60° \\ 0 & \text{otherwise} \end{cases} \tag{6.5}$$

$$S_e^i = \mu_j(s_e) \tag{6.6}$$

The trend of emission and incidence scores for a single angle value (s_i and s_e) is shown in Figure 6.2.

Emission variation score Also, a large variation of emission angles is considered beneficial, therefore the emission variation score is:

$$S^i_{\Delta e} = \mu_j \left(\max_k \frac{2\Delta e_{jk}}{\pi} \right) \tag{6.7}$$

where $\Delta e_{jk} = |e_j - e_k|$. So for each emission angle e_j under which the i-th facet is seen, the maximum difference between the considered angle e_j and the all the other angles e_k under which the facets was observed is computed. Then all the maximum differences are normalized of $\frac{\pi}{2}$, i.e. the maximum possible emission variation, and the mean is performed.

Solar and spacecraft azimuth score Finally, the variation of solar azimuth angles α should be large and the one of spacecraft azimuth angles β small. The respective scores are computed in a similar fashion.

$$S^i_{\Delta \alpha} = \mu_j \left(\max_k \frac{\Delta \alpha_{jk}}{\pi} \right) \tag{6.8}$$

$$S^i_{\Delta \beta} = 1 - \mu_j \left(\max_k \frac{\Delta \beta_{jk}}{\pi} \right) \tag{6.9}$$

Please note that in this case the normalizing value is π. All the angle variations are defined as shown in Figure 6.3.

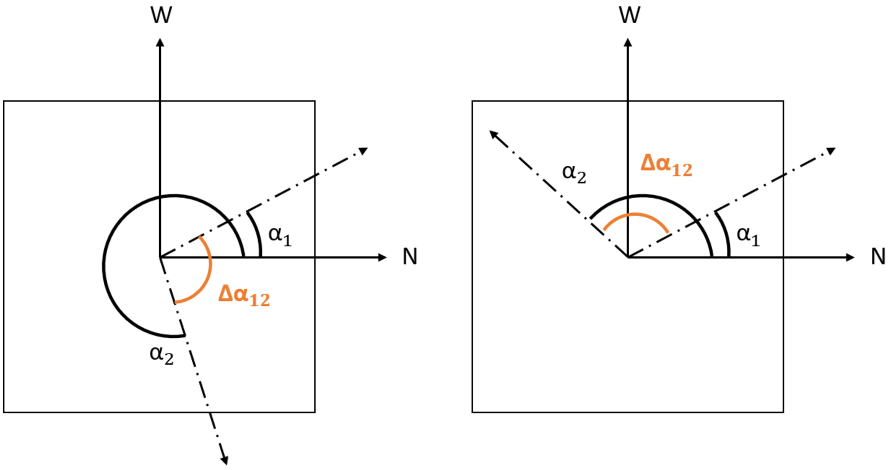


Figure 6.3: SPC facet angles variation examples, N = local North, W = local West

Facet score The overall facet score is given by the weighted sum of the different contributions. Please note that the scores defined above have values ranging between 0 and 1.

$$S^i = w_i S^i_i + w_e S^i_e + w_{\Delta e} S^i_{\Delta e} + w_{\Delta\alpha} S^i_{\Delta\alpha} + w_{\Delta\beta} S^i_{\Delta\beta} \tag{6.10}$$

The weights of the different criteria are computed according to AHP procedures. AHP is a multicriteria decision making method developed by Saaty [84]. In his work a procedure to derive weights for a set of criteria is defined, according to their relative importance and relating them in a quantitative way. A matrix A with elements a_{ij} is created to store the relative weights of the criteria, chosen according to Table 6.1.

Value of $\mathbf{a_{ij}}$	Relative importance
1	j and k equally important
3	j slightly more important than k
5	j more important than k
7	j strongly more important than k
9	j absolutely more important than k

Table 6.1: Table of relative scores

The resulting matrix here proposed for the facet scores is shown in Table 6.2. The weight vector is computed as the eigenvector associated to the maximum eigenvalue and then normalized so that the sum of all the weights is 1. The obtained (here rounded) weights are:

$$w_i = 0.56 \quad w_e = 0.17 \quad w_{\Delta e} = 0.06 \quad w_{\Delta\alpha} = 0.10 \quad w_{\Delta\beta} = 0.10 \tag{6.11}$$

	$\mathbf{S^i_i}$	$\mathbf{S^i_e}$	$\mathbf{S^i_{\Delta e}}$	$\mathbf{S^i_{\Delta\alpha}}$	$\mathbf{S^i_{\Delta\beta}}$
$\mathbf{S^i_i}$	1	5	7	5	5
$\mathbf{S^i_e}$	1/5	1	5	3	3
$\mathbf{S^i_{\Delta e}}$	1/7	1/5	1	1/3	1/3
$\mathbf{S^i_{\Delta\alpha}}$	1/5	1/3	3	1	1
$\mathbf{S^i_{\Delta\beta}}$	1/5	1/3	3	1	1

Table 6.2: Matrix A for facets scores

Facet mapping index The overall score represents the goodness of the images, but does not account for the fact that increasing the number of pictures, the knowledge of the area betters. Therefore the facet mapping index m^i is defined for the i-th facet:

$$m^i = S^i \min\left(1, \frac{n}{N}\right) \tag{6.12}$$

where N is an arbitrary number of images. For instance, if N = 3, having 3 photos with an ideal maximum score equal to 1 would be satisfying for mapping the area. When the number of images is too low, the area is not well known so an high score is not significant. On the other hand, having at least N images, additional images increase the map knowledge only if they better the score.

6.3.2 Reward definition

The immediate reward depends on both states and actions:

$$r_k = r_k(s_k, a_k) \tag{6.13}$$

If no action is taken the reward is null. When an image is collected in a forbidden state $s \in S^-$, a negative reward equal to -1 is returned to the agent and the image is not accounted for in the successive mapping. Forbidden states correspond to situations in which the image is in complete shadow or when the ideal number of images is overcame, causing problems in on-board data storage. The ultimate goal is to maximize the mapping index, therefore if the picture is taken in allowed states the reward is:

$$\tilde{r} = \mu_m \left(\frac{m_k^i - m_{k-1}^i}{m_k^i} \right) \tag{6.14}$$

where m_k^i is the mapping index of facet i at time k and μ_m stands for the mean over all the facets in the current frame. Summarizing, the overall reward is:

$$r_k = \begin{cases} -1 & \text{if } a_k = 1 \text{ and } s_k \in S^- \\ 0 & \text{if } a_k = 0 \\ \tilde{r} & \text{otherwise} \end{cases} \tag{6.15}$$

If the agent immediately takes all the photos that can be sent on ground, for all the successive time steps it will be forced to accept a zero or a negative reward. On the other hand, the long term reward will be higher if images are collected only when it is worth. This formulation of the overall reward is inspired by [82], where prohibited states are defined for modeling learning tasks in control problems.

6.4 Actions and states model

6.4.1 Actions

Here the reduced action space is defined. The agent interacts with the environment only by choosing its sensing locations, so by collecting images, without controlling its relative pose with respect to the body surface. A direct and continuous control $\mathbf{u_k}$ on the trajectory, as the one included in active SLAM problems, would be too risky for this application. As a straightforward consequence of the reduced problem definition, actions are discrete. The action at time step k is simply boolean:

$$a_k = \begin{cases} 0 & \text{if no picture is taken} \\ 1 & \text{otherwise} \end{cases} \tag{6.16}$$

The number η of pictures to be ideally taken in a certain storage time $T_{storage}$ is fixed. After this storage time images are downlinked and therefore the memory is empty again. The discrete time steps in which an action can be taken are defined in number equal to the ideal number of images times the control parameter Δ_c. Ideally with a large control parameter the final performance would be better, but also the number of decisions to be taken would be too high. Shortening the control interval the overall on-board computational time increases and also the learning time. So a trade off between performance and learning must be done when choosing Δ_c.

6.4.2 States

Also the state space is not equal to the complete space of the general POMDP: spacecraft position and orientation, Sun illumination and map representations are reduced. In particular, to directly consider the full and complete map state $\mathbf{m_k}$ would be too onerous, given the large number of landmarks on the body surface. States have been designed to synthesize only the information necessary and useful for decision making. The use of statistical quantities (mean and standard deviation) is the only solution that allows to keep the number of observed states constant despite of the change of number of facets in view. Moreover, to understand how the mapping campaign is proceeding, all the history of past actions should be part of the states as well. Of course to include the whole history in the states observation is not possible, but anyway the POMDP is reduced by making part of the history observable.

The reduced state includes memory, map and angles states, defined as follows:

$$\text{memory state}\begin{cases} s_1 & = \Big(t - \text{floor}\Big(\frac{t}{T_{\text{storage}}}\Big)T_{\text{storage}}\Big)/T_{\text{storage}} \\ s_2 & = \frac{n}{\eta} \end{cases} \tag{6.17}$$

$$\text{map state}\begin{cases} s_3 & = \frac{A_{\text{light}}}{A_{\text{view}}} \\ s_4 & = \mu_m(m^i) \\ s_5 & = \sigma_m(m^i) \\ s_6 & = \mu_M(m^i) \\ s_7 & = \sigma_M(m^i) \end{cases} \tag{6.18}$$

$$\text{angles state}\begin{cases} s_8 & = \mu_m(s_i) \\ s_9 & = \mu_m(s_e) \\ s_{10} & = \mu_m\Big(\ \max_j \frac{\Delta\alpha_{jk}}{\pi}\Big) \\ s_{11} & = \mu_m\Big(\ \max_j \frac{\Delta\beta_{jk}}{\pi}\Big) \\ s_{12} & = \mu_m\Big(\ \max_j \frac{2\Delta e_{jk}}{\pi}\Big) \end{cases} \tag{6.19}$$

Memory state The memory state provides information on the time lapse and number of collected images. The idea is that in a certain time interval T_{storage} pictures can be stored in the on-board memory before being sent on ground. The ideal number of images to communicate at every time interval is η. In particular, s_1 represents the percentage of time spent in the current storage interval, while s_2 the number of pictures taken n with respect to the ideal number η. The T_{storage} and η parameters can be tuned depending on mission constraints without affecting the algorithm. These inputs help in evaluating how the collection of a new image would impact on data storage.

Map state The map state provides general information on the mapping campaign advancement. s_3 is the fraction of area in light of the surface portion in view, thus telling the area percentage whose knowledge will actually be improved by a new picture. It can be roughly computed as the ratio between the image facets in light and the total number of facets visible in the image. s_4 and s_5 are the mean of the mapping index and its standard deviation over the surface in view, while s_6 and s_7 are the same quantities over the whole body. These data are useful to decide whether the exploration of the area under exam is worth from the coverage point of view.

Angles state The angles state gives local information about photometric angles under which the facets in view and light are seen at the present time. In particular, s_8 and s_9 are the inclination and emission scores mean. Please note that the means are performed over all the facets in view and in light for the angles of the current time instant only. While s_{10}, s_{11} and s_{12} are the facets mean of the maximum variation of current Sun azimuth, spacecraft azimuth and emission angles with respect to the angles of already take pictures. These inputs allow to evaluate the possible improvement of SPC for what concerns stereo angles and illumination conditions.

Please note that all the states can be estimated on-board through observations of the environment elements described in section 6.2. So they can be computed exploiting data that are available to the agent. In particular, during the global mapping phase it can be fairly assumed to have a knowledge of a rough shape model of the body. The camera field of view is of course known, while Sun direction and camera pose can be estimated. In addition, all the states are here assumed to be known with certainty, so the belief is about equal to the states $\mathbf{b_k} \sim \mathbf{s_k}$, therefore simplifying the POMDP. In particular, the memory data are known with certainty, while relative position, attitude and surface illumination can be estimated through sensors and determination algorithms that are typically available on-board.

It has to be highlighted that history of past observations is included as an input since historical information about conditions under which past images were taken is necessary for decision making. As explained in Chapter 4, this step is useful to eliminate the dependence of the environment state from any historical information not included in the state itself and therefore to respect the Markov property. Nevertheless, to input the full history to the neural network is not possible and data about past photometric angles and mapping quality need to be preprocessed. This implies a loss of information that makes the environment not fully observable, similarly to voluntary perceptual aliasing.

With the presented definitions of reward, states and actions, the general POMDP is reduced to:

$$\pi^\star = \underset{\pi}{\operatorname{argmax}} \, \mathbb{E}_\pi \left[\sum_{k=0}^{T} \gamma^k r_k(\mathbf{a_k}, \mathbf{s_k}) \right] = \underset{\pi}{\operatorname{argmax}} \, Q_\pi(\mathbf{a_k}, \mathbf{s_k} | \theta_\pi) \tag{6.20}$$

Please note that the choice of DRL as solving tool to design the policy perfectly fits this application. In fact, DRL can well handle large state and observation spaces, but only discrete and low dimensional action spaces, as explained in Chapter 5.

6.4.3 States computation

In order to feed the states to the neural network, a preprocessing block is included in the algorithm (see Figure 6.1). The preprocessing block has the aim of elaborating known environmental information for the generation of the neural network inputs. The main difficulty is how to exploit them without an exceeding computational cost. At each considered time instant, the facets in the camera field of view and in light are detected. This can be easily done with geometrical considerations only, thanks to the knowledge of the polyhedral shape model, the camera pose and the Sun position relative to the surface. Of course it is necessary to account for the body self-occlusion and self-shadowing. In fact, having small bodies irregular shapes, some surface areas may be not visible because hided by other portions of the body and in a similar manner, the shadow distribution on the surface is influenced by the shape irregularity. Figures 6.4 and 6.5 show the relevance of self-occlusion and self-shadowing for an irregular body.

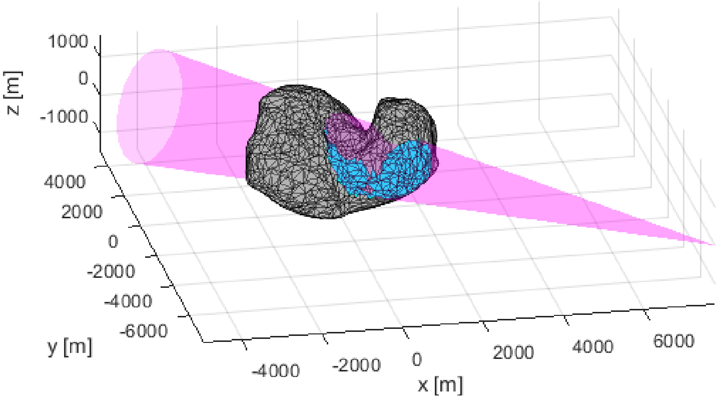


Figure 6.4: Facets in the field of view cone for comet 67P-CG. Relevance of self-occlusion.

Then, for the facets in view and in light, the photometric angles are computed. At this point, the computation of the neural network inputs is straightforward. The only historical information needed are the photometric angles under which each facet was seen in past taken images. In particular, photometric angles are stored for each facet and facets scores and mapping index are updated.

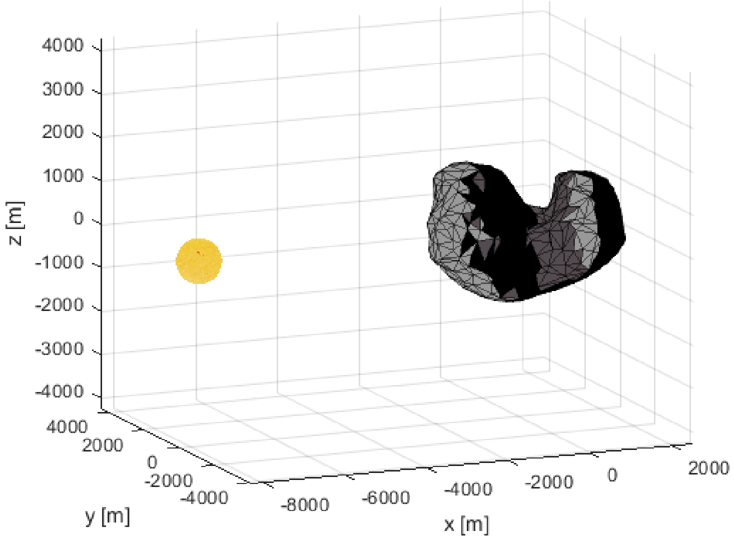


Figure 6.5: Facets in light and shadow for comet 67P-CG. Relevance of self-shadowing.

Please note that the preprocessing block is the one that has the largest computational cost, mainly because computations related to self-shadowing and self-occlusion involve a double loop on the shape model facets. Anyway this step is fundamental to increase the algorithm generality and to synthesize information provided to the network. It would also be possible not to consider the shape model directly and to implement a different algorithm architecture that does not need to preprocess geometric data, thus saving computational time. Anyway this would open many other issues, like how to handle historical information and how to represent the map state. Moreover, if geometrical information is not included in the network inputs, the task of relating camera pose, illumination and surface geometry would be transferred to the network and needed to be learned during the training. Such approach implies an increased complexity and a loss of generality: the mapping quality is strictly related to the surface geometry and there is no way to deduce it if the training has been done on a different body. So, the present approach prefers to accept the largest computational cost and critically investigate the consequences of a delay in the decisions, while keeping a large generality and independence from the specific mission.

7 Learning and test results

This chapter deals with learning and testing of the autonomous mapping planning policies. First, the learning process of the policies with DRL is commented: the chosen learning environment is described, then the ANN architecture and hyperparameters choice are motivated in detail for both NFQ and DQN. Critical aspects of the learning process and differences between the two techniques are highlighted. Then, the testing simulation environment is described. Detailed results are presented for four main test cases along with sensitivity analyses that extensively verify the policies performance and robustness, considering different possible scenarios. A comparison with simple benchmarks is made. Finally, a computational analysis is performed.

7.1 Learning

7.1.1 Learning environment

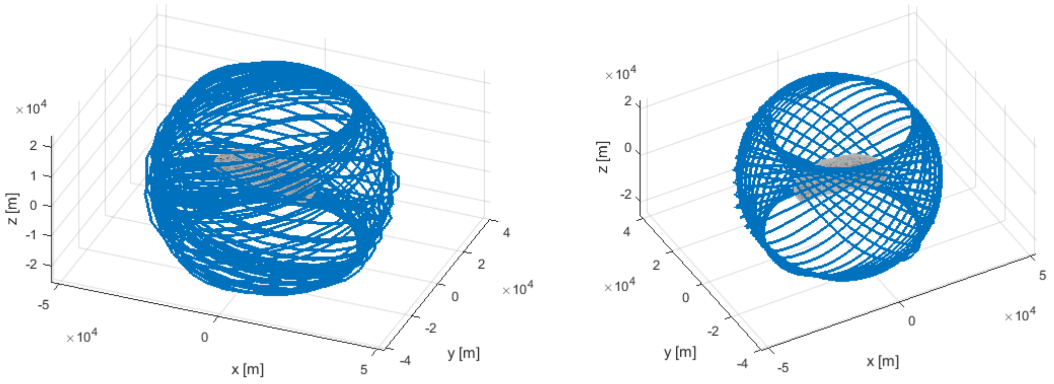
To properly define the environment with which the agent interacts during the learning is of fundamental importance for the learning success. The experiences set should be complete, i.e. it should be an exhaustive collection of all possible cases that the agent may encounter. Please note that the state is defined to be independent from the asteroid, orbit and camera characteristics. Therefore a complete training set is not a set built considering several asteroids and orbits, but a set of examples that sufficiently explores the state space and includes relevant experiences for reaching the final goal. Ideally it should contain a whole *mapping stage*, from the beginning to the end, in which all the pictures are taken with the same instrument and at about a constant distance from the asteroid. Since the resolution of the maps to be created is finer than the one already achieved at larger distances, the stage can be considered independent from past stages for what concerns the coverage of the asteroid. In a few words, the learning environment should allow the agent to collect both experiences in prohibited states S^-, to avoid them in the future, and to make very successful actions for mapping. For these reasons, in order to enhance the learning process, a somewhat unrealistic situation is selected as learning environment:

- Non-keplerian orbit around asteroid Eros in Figure 7.1.
- Camera FOV of 10°.

Before the selection of such training environment, more realistic orbits and cameras were considered. Anyway, they have been discarded because the policies so obtained revealed to be significantly inferior, because they allowed a minor variation of the spacecraft-Sun-body relative geometry, therefore limiting the mapping possibilities. This confirms the extreme importance of experiences that constitute the training set.

The chosen asteroid is Eros because it is one of the few shape models available in databases and because its elongated shape allows to image different percentages of the body surface, keeping the distance fixed. In fact, the percentage of surface in view varies between 6.4% and 0.7% with a mean of the 2.4% along the orbit. Please remember that typical values are of about $3\% - 1\%$ for the global mapping phase, as explained in chapter 2. The FOV has been chosen to image wider portions of the body with respect to typical values, allowing to reach a good mapping with relatively few images, thus reducing the number of repetitive experiences that would be necessary to map with a smaller FOV. Anyway, experiences in which the percentage of surface in view is more realistic are also collected, in the areas with maximum body radius.

Also the selected orbit has good characteristics for mapping, because the shift in RAAN due to Eros elongated shape allows to change the phase angle between spacecraft and Sun. The trajectory has been obtained considering the spherical harmonics perturbations, starting from an initial condition corresponding to osculating orbital parameters of null eccentricity, 45° inclination and radius twice the asteroid maximum one. Such an orbit has been a serendipitous finding and has been selected because of its nice characteristics. Anyway, please note again that this situation is somehow fictitious, especially because of the close proximity, and that any good mapping strategy would have been suited too, even a sequence of keplerian orbits or an imposed trajectory that does not follow natural dynamics. For what concerns the body illumination, some areas remains always in shadow, as it can be seen from Sun direction in the body-fixed frame shown in Figure 7.2. The reason is that the body spin axis inclination with respect to the ecliptic north is larger that Eros orbit inclination. This allows to frequently collect also negative experiences for the body mapping, which need to be learned and avoided. Data for asteroid orbit, rotation period and spin axis orientation are taken from JPL Small-Bodies Database [51] and International Astronomical Union reports [85].



(a) Spacecraft position in the body-fixed frame (b) Spacecraft position in the inertial frame

Figure 7.1: Spacecraft position in training environment

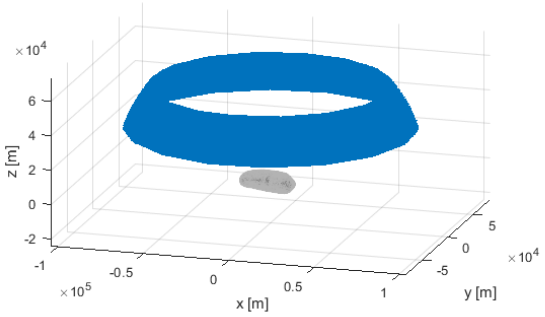


Figure 7.2: Sun direction in the body-fixed frame. Training environment.

The training simulation environment also assumes that the ideal number of images during one episode is 500, with an ideal frequency of 1 picture per hour. The downlink and control parameters defined in chapters 5 and 6 are:

- $T_{\text{storage}} = 10$ h
- $\eta = 10$
- $\Delta_c = 3$

In practice, the orbit is discretized so that the number of points in the storage time is three times the number of photos allowed. So the control interval between one action and the successive one is quite coarse. This interval can be refined in future works.

7.1.2 Neural network architecture design

The MLP used to approximate the Q-value function has the architecture shown in Figure 7.3.

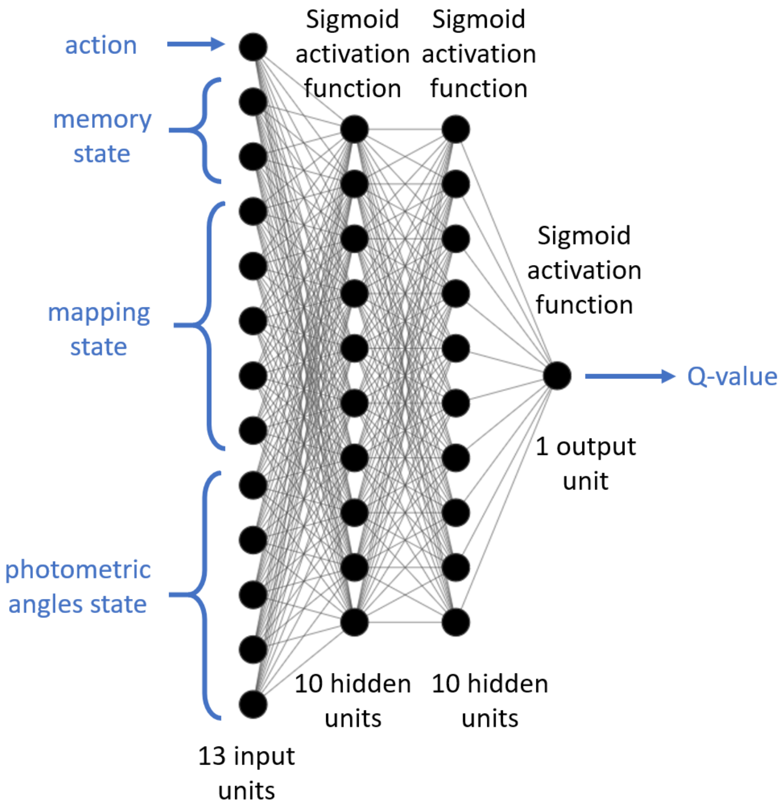


Figure 7.3: Neural Network architecture

The MLP architecture is the same for both NFQ and DQN and is kept as simple as possible, with a 13 elements input vector and a scalar output, in accordance to planning necessities (see chapter 6). For what concerns the network hyperparameters, there are two hidden layers made of 10 neurons and all activation functions are tangential sigmoids. Such hyperparameters have been empirically chosen, according to typical values for DRL algorithms with similar input and output sizes [82]. Please note that the tangential sigmoid output is limited in interval $[-1, 1]$:

$$\tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1} \tag{7.1}$$

therefore it is necessary to scale outputs, as the possible Q-value is unknown. Since reward assumes both positive and negative values, Q-value could have any sign, so activation functions with output intervals $[0, 1]$ are not appropriate. For the present application hyperparameters optimization is not fundamental, since the network architecture is really light and training is thought to be performed on-ground, with no particular constraints on computational resources. Therefore a simple check has been done at the end of the design, doubling the number of neurons in hidden layers: the final performance does not improve, confirming that the chosen architecture is robust enough.

7.1.3 NFQ learning

All the algorithms have been implemented in MATLAB environment. NFQ implementation is easier than DQN, since the experiences collection is not nested within the training algorithm, as visible from schemes in Figure 5.5 and 5.6. Therefore MATLAB Neural Network Toolbox$^{\text{TM}}$ has been used for the NFQ MLP training. After experiences collection, the set is split into validation (15%) and training (85%). Such division is used to understand when the training needs to be stopped, with usual validation checks. A test set is not considered, because all the tests are performed in a second moment interacting with environments different from the learning one. The discount factor is $\gamma = 0.95$, as commonly found in literature. RPROP is selected as training algorithm because of it robustness, with parameters as in appendix A. In particular, batch learning is preferred to incremental learning, because the training set has a low dimension (500 ideal images and $\Delta_c = 3$ lead to a total number of 1500 experiences). Input and output scaling is performed on the whole experiences set. The Mean Square Error (MSE) between network outputs and targets is reported for one training iteration in Figure 7.4, where one epoch corresponds to a weights update step on the entire set of experiences. As it can be observed, the MSE smoothly decreases until the validation check is met, i.e. the validation error stops decreasing. Of course the training error is lower than the validation one.

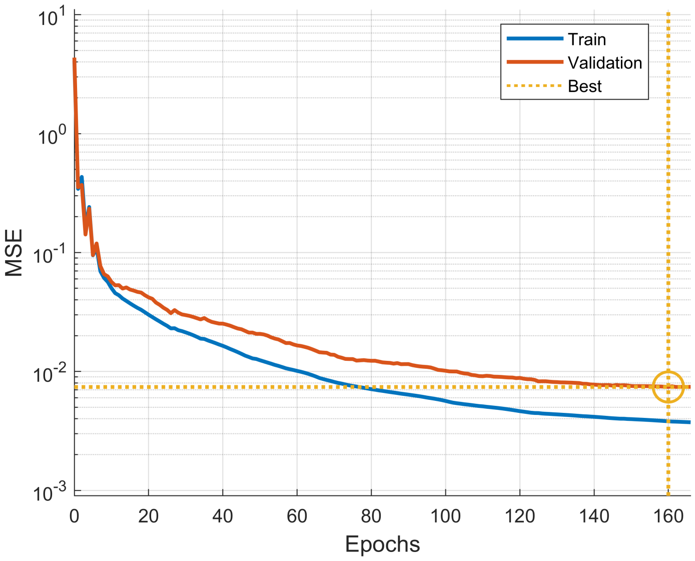


Figure 7.4: Batch learning on NFQ network

Special attention was deserved to the fine tuning of the reward model, and in particular to the prohibited memory states (please see chapter 6). At first the agent was punished each time the stored pictures exceeded the ideal number before downlink, i.e. an image was taken when $s_2 > 1$. This definition resulted too restrictive and the obtained policies collected very few images. On the contrary, relaxing the punishment lead to an excessive collection of images, whatever other conditions. A trade-off has been finally made by redefining the prohibited memory state as the one in which the memory is full ($s_2 > 1$) and the number of images overcomes the ideal value at the k-th time instant ($n_k > n_{k,ideal}$), with

$$n_{k,ideal} = \frac{k}{\Delta_c} \tag{7.2}$$

Riedmiller [82] when suggesting practical advices for NFQ implementation shows that the original NFQ algorithm can work with a fixed set of transitions that can be sampled randomly. Anyway, some artificial training transitions can be added to the training set, as the author suggests. This practice revealed to be an important mean for the policy success. Artificial transitions have been added so that each time the agent acting randomly fell into a prohibited state, the opposite action was also taken and the training episode continuing from this new state.

For what concerns NFQ stopping criterion, typical methods operate with a fixed number of iterations, without particular stopping criteria [82]. For the first iteration, the estimated Q-value is taken equal to the immediate reward, without adding the discounted Q-value. It has been found out that iterations after the first one immediately satisfy validation checks. This result was completely unexpected and means that the algorithm prematurely converges, becoming comparable to simple supervised learning. The approximation done by the net is such that the immediate reward is confused with the long-term return when trained on the whole batch. Some variants of the algorithm exist that re-initialize the network at each NFQ iteration, before the training [82]. In this case the algorithm does not prematurely converge, as shown in Figure 7.5. In particular, 50 iterations are sufficient to reach convergence.

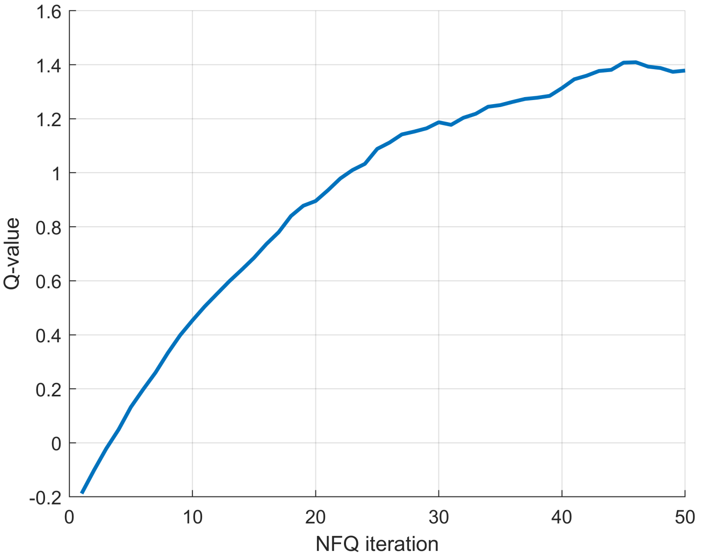


Figure 7.5: Mean of expected Q-value during NFQ learning

Both the original algorithm and the one with re-initialization result with good policies. However, the latter policy privileges again a rigid respect of memory constraints in spite of mapping improvement. On the other hand, surprisingly a supervised training seems sufficient to obtain a good performance, but only with a well studied training set. Since during tests such results proved to be satisfying and better than the ones obtained with re-initialization or different settings, the policy with one iteration only is taken as the reference for NFQ.

In conclusion, for the NFQ algorithm the neural network hyperparameters and the RPROP ones are not a big issue: the training algorithm is very stable and the network is light, so it can be not optimized. On the contrary, the most critical steps of the design are the collection of training experiences and the reward model. These aspects are tightly bound to the specific problem and to its formulation. During the design process an effort has been made in order to tune these aspects at best. Of course other possibilities may exist.

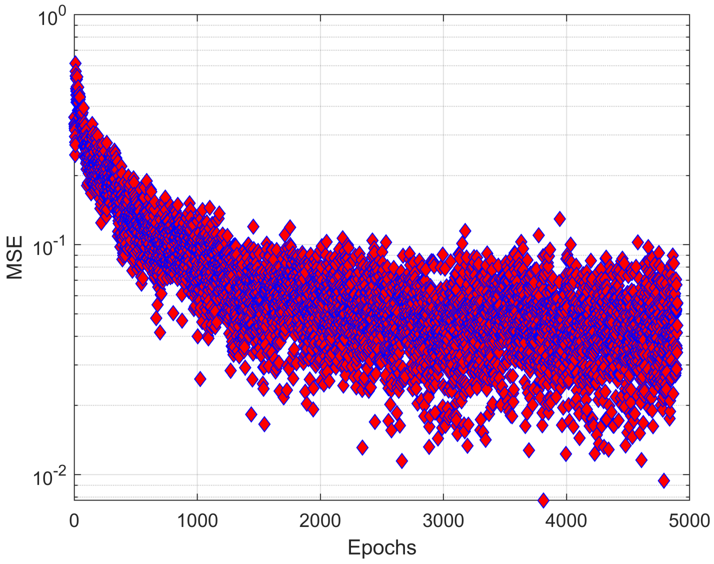
7.1.4 DQN learning

Unlike NFQ, DQN requires a continuous collection of experiences. For this reason, it is not possible to use MATLAB toolbox, therefore training algorithms have been implemented and nested in the learning loop.

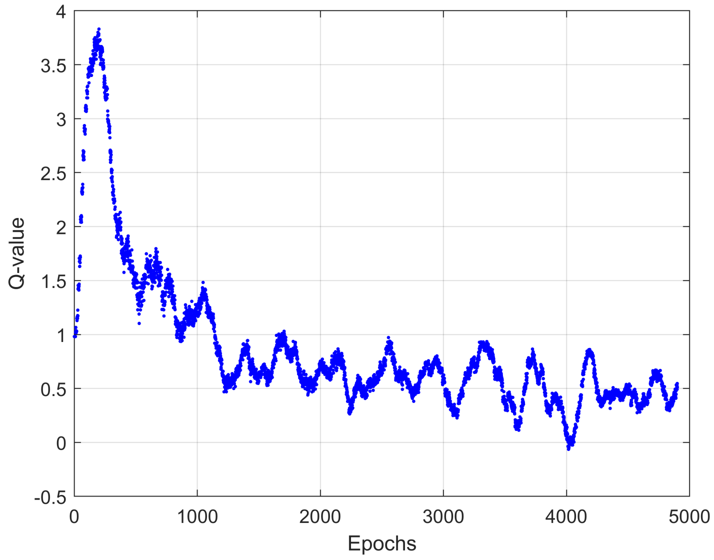
The training environment is again the one described in section 7.1.1, with same redefinition of prohibited states as for NFQ. Experiences are collected accordingly to the ϵ probability and to the current policy. Once an episode is finished, a new one is restarted from scratch. A dynamic scaling of outputs is necessary. In particular, outputs are rescaled each time the target net is updated. The discount factor is again $\gamma = 0.95$.

During training simulations it has been noticed that two important hyperparameters for the algorithm success are the target update frequency C and the mini-batch size. They have been set respectively to 10 iterations and 100 experiences. Initial replay memory size has been also set to 100. Another key parameter is the probability ϵ of following a random policy or the trained one. This value has been linearly varied, as typically done [83]. Without a proper setting of such parameters the algorithm diverges.

The MSE and Q-value evolution during the learning are reported in Figures 7.6a and 7.6b. In this case one epoch corresponds to one RPROP step on the mini-batch. As it can be noticed, MSE decreases but it is much less stable with respect to the NFQ one. This is a consequence of the different batch sizes used in the two algorithms. The Q-value has an initial peak, that corresponds to a wrong estimation, and then its value settles down oscillating. As for NFQ, this plot only shows the mean Q-value estimate on a batch of random experiences. So it does not indicate an evolution of the score achieved by the policy during learning, but



(a) Training Mean Square Error on DQN network



(b) Mean of Q-value during DQN learning

Figure 7.6: DQN learning

provides information on how the function approximation proceeds. The Q-value oscillations during learning are a typical behavior, as found in literature [83].

Exploration finishes (i.e. experiences are collected completely following the trained policy) at epoch 1000. The final number of epochs has been tuned to 5000. A possible improvement of the stopping criterion is to play one episode at each epoch and examine the final score. Such score plots require great computational resources and usually present oscillating behaviors as well. Since positive results have been obtained tuning the final number of epochs, such plots have not been evaluated, but are suggested for future developments.

7.2 Testing simulation environment

The policies obtained through DRL are extensively tested over a wide range of different possible scenarios to verify their generalizing capability, which is of great importance when exploring an unknown environment. A first numerical validation is performed comparing the DRL-based algorithm with two different simple benchmarks: a policy that takes pictures at regular intervals (UNI) and another that randomly selects the image acquisition instants (RAND). For the RAND strategy if $n_k > n_{k,ideal}$ the image is discarded and all the presented results are the mean over 100 runs. While for UNI, NFQ and DQN only 1 run is necessary, since they are deterministic policies.

Since the learning process is highly related to reward modeling and hand-tuning of the many parameters involved in the process, policy testing has a great importance. The testing scenarios here considered have been chosen also outside the boundaries of usual global mapping mission framework (defined in chapter 2), increasing mapping difficulty and challenging the policy capabilities.

The test cases have been chosen to cover all relevant aspects for the algorithm application. In particular, four different bodies are considered: Eros, on which the training has been performed, Itokawa, that presents an elongated shape, Bennu, with diamond shape, and 67P-CG, with two-masses shape. In addition, sensitivity analyses have been done varying:

- The distance from the body, that affects both relative dynamics and percentage of surface in the camera field of view.
- The body rotational period, that influences illumination conditions variation and again relative pose.
- The orbit inclination, that changes the surface portion object of the mapping.

Small bodies are assumed on keplerian orbits around the Sun [51]. Spin axis orientation and rotational period are assumed constant. This assumption is reasonable for small bodies uniformly rotating, even if in practice effects like Yorp or sublimation-induced torques change the spin state. In [85] small bodies poles right ascension and declination are provided at a fixed time with respect to the invariable plane of the Solar System, taken as reference plane. Here for simplicity those values are taken and the ecliptic is assumed coincident with the reference plane.

The body shape models have all the same number of facets (1000). For what concerns the environment parameters η, $T_{storage}$ and Δ_c are equal to the training environment. In [82] it is suggested to train the network for a specific control interval. Therefore this parameter will not be varied with respect to the training and it will be kept constant in all the simulations. The camera is assumed to have a conical FOV of 3°. The length of each episode is set to 1500 steps, with an ideal number of images of $\frac{1500}{\Delta_c} = 500$. So what determines the episode end is time. It is meaningless to terminate the episode when the mapping campaign is finished, because typically the complete body mapping can not be reached. This is due to the assumptions made: in some test cases parts of the body are always in shadow or the orbit inclination does not allow to cover the whole body. Moreover, given the mapping index definition in chapter 6, even for the single facet it is in practice not possible to reach the maximum ideal value of 1. The aim is to compare UNI, RAND, NFQ and DQN strategies during the same time lapse and see how they perform in different scenarios.

Often DRL results are compared just with the numerical final score obtained during the episode. In such a way, however it is not easy to critically analyze how policies actually behave. Being the design and learning procedure highly based on engineering judgment, the test results are presented not with final reward scores, but by means of some indexes that allow to easily understand the performance:

- Final number of collected images $I_n = n_{tot}$, (a lower value is better).
- Final mapping index $I_{map} = \mu_M(m^i_{k_{end}})$, (a higher value is better).
- Integral mapping index over the campaign $I_{sum} = \frac{1}{\Delta_c} \sum_k \mu_M(m^i_k)$, (a higher value is better).

Such parameters quickly allow to verify if the modeled reward actually leads to an improvement of the proposed tasks: data reduction and mapping enhancement and fastening.

In other literature works [31], [36] extensive tests are not presented, but only a single or a couple of scenarios are considered. Here the generality of applicability of the algorithm is proved and the performance limits assessed, considering a large variety of relevant test cases.

7.3 Case A: Eros

7.3.1 Detailed results

The basic simulation scenario for asteroid Eros has the following parameters:

- Rotational period $T_{rot} = 6$ h.
- Circular polar orbit, with interest ratio equal to 4.
- Percentage of surface in view in the range $0.3\% - 2.9\%$.

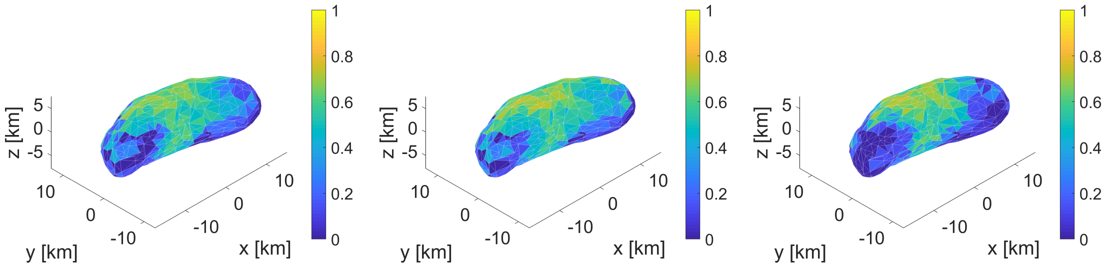
All the surface percentages indicated in this and in the following sections are computed considering the real shape model and not an equivalent sphere. The final performance indexes are displayed in Table 7.1, comparing the different strategies. As it can be observed, both NFQ and DQN perform better than UNI and RAND: the overall number of images is sensibly lower and final mapping index larger.

Strategy	**NFQ**	**DQN**	**UNI**	**RAND**
I_n	404	434	500	500
I_{map}	0.26	0.29	0.22	0.21
I_{sum}	86.50	89.67	69.43	66.61

Table 7.1: Strategies comparison for basic scenario, case A

The final mapping index is show for NFQ, DQN and UNI strategies in Figure 7.7. The mapping is limited by the fact that the area with negative z axis remains in shadow, with the considered spin axis orientation and asteroid orbit inclination.

DQN achieves a better mapping than NFQ, but more images are collected. The trends of global mapping index and taken images are shown in Figures 7.8 and 7.9 for the whole episode. In particular, Figure 7.8 helps to understand the I_{sum} values of Table 7.2: NFQ and DQN have a similar trend, superior to the one of other strategies, thus fastening the mapping process. The data storage during the mapping is shown in Figure 7.10. As explained, the constraint on memory handling has been relaxed during the learning and peaks of data collected are accepted. Such peaks are larger for DQN, but they are anyway contained considering that $\frac{n}{\eta}$ never



(a) NFQ (b) DQN (c) UNI

Figure 7.7: Eros, facets final mapping index

exceeds 2 and $\Delta_c = 3$. Finally, the light fraction of the frames (corresponding to state s_3) is shown for a representative interval in Figure 7.11, along with the boolean values of NFQ and DQN actions. When the frame is in complete shadow images are never taken: both policies have learned that imaging an area in complete shadow is not worth. On the contrary, when the frame is in complete light, sometimes it is not collected. This means that the policies do not simply collect all the images in light but have learned how to select them in base of other criteria. Actions taken by the two strategies are different in some cases.

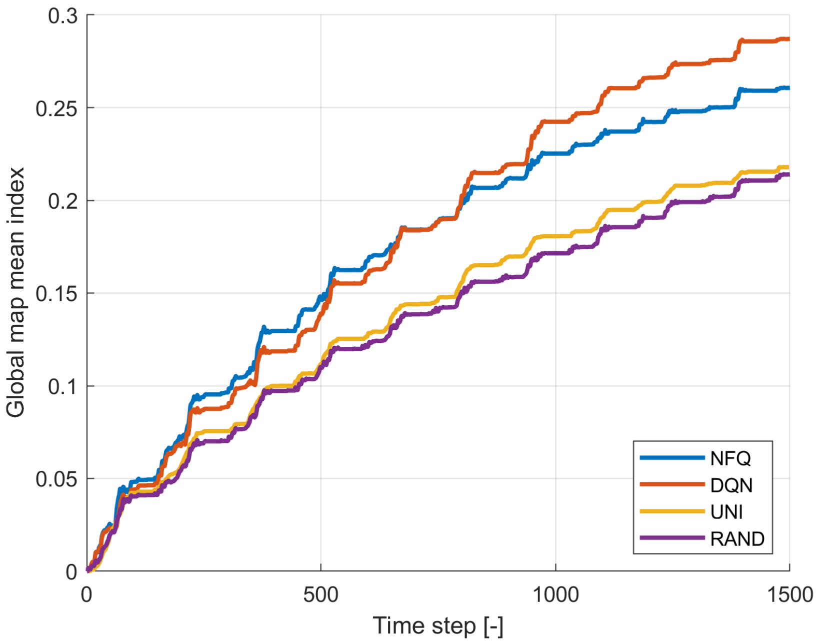


Figure 7.8: Eros, global map mean index

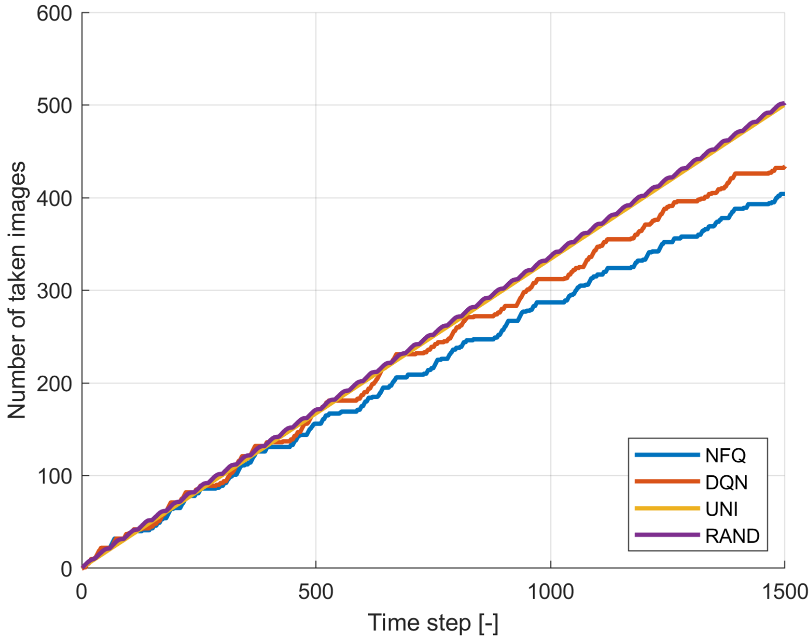


Figure 7.9: Eros, taken images

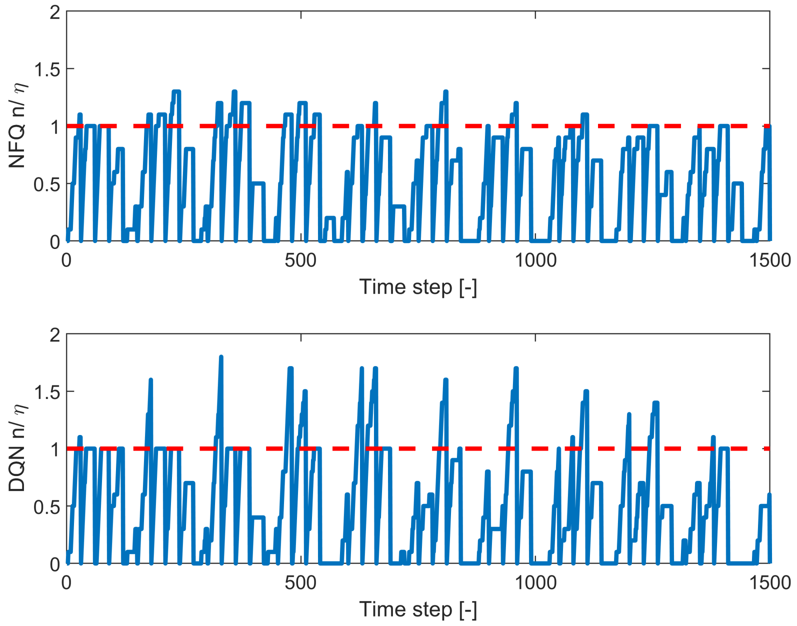


Figure 7.10: Eros, memory

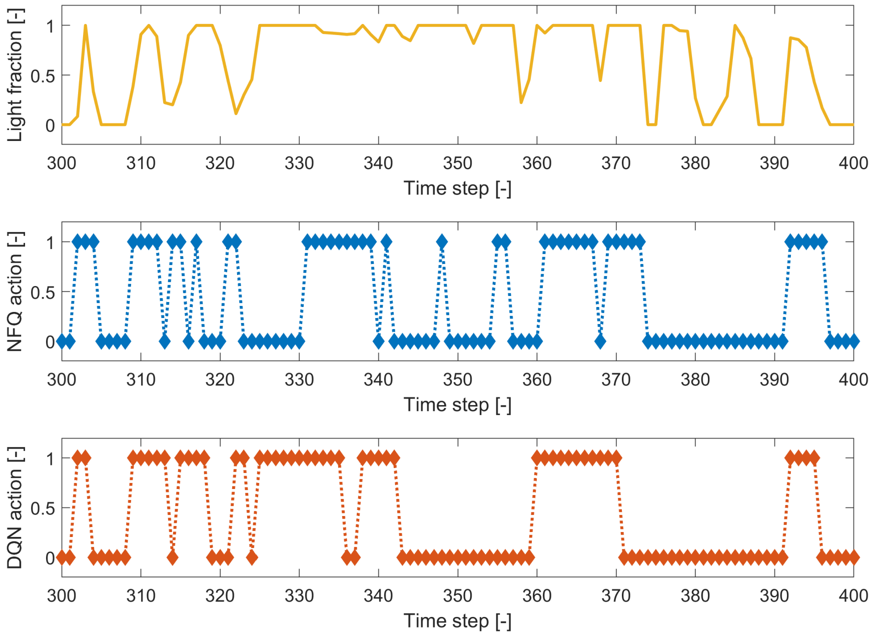


Figure 7.11: Eros, light fraction

7.3.2 Sensitivity analysis

The sensitivity analysis results are reported in Table 7.2. With interest ratio 6 the percentage of surface in view ranges between 1.1% and 6.6%, while with interest ratios 8 and 10 it is respectively in the intervals 2.7% – 12.2% and 5% – 18.4%.

NFQ and DQN show a very good generalizing capability and outperform UNI. A general trend observed is that with NFQ the number of collected pictures is lower than DQN and UNI, but with DQN a larger mapping performance is achieved, sometimes even with a lower number of pictures. In some cases, I_{map} has a similar value for the three strategies, but the goal is achieved more quickly by NFQ and DQN. The only case in which the two strategies have I_{map} slightly lower than UNI is with interest ratio 10. This may be due to two reasons: the percentage of surface in view is out of training interval and of typical mission values (please refer to chapter 2); or when a large portion of the body is imaged it is more difficult to have control on the viewing conditions of all facets in the frame. In fact 1% of the surface means to consider about 10 facets, while 10% 100: very different viewing conditions may be present in the same picture. Please note that anyway the number of pictures for DQN is only 326, so the amount of data is largely reduced in spite of a small difference in I_{map}. This is not true for NFQ.

	i = 30 deg			**i = 60 deg**			**i = 90 deg**		
	$\mathbf{I_n}$	$\mathbf{I_{map}}$	$\mathbf{I_{sum}}$	$\mathbf{I_n}$	$\mathbf{I_{map}}$	$\mathbf{I_{sum}}$	$\mathbf{I_n}$	$\mathbf{I_{map}}$	$\mathbf{I_{sum}}$
UNI	500	0.21	66.87	500	0.22	69.10	500	0.22	69.43
NFQ	430	0.25	87.64	431	0.26	87.58	404	0.26	86.50
DQN	498	0.24	83.30	493	0.26	84.91	434	0.29	89.67
	Interest Ratio = 6			**Interest Ratio = 8**			**Interest Ratio = 10**		
	$\mathbf{I_n}$	$\mathbf{I_{map}}$	$\mathbf{I_{sum}}$	$\mathbf{I_n}$	$\mathbf{I_{map}}$	$\mathbf{I_{sum}}$	$\mathbf{I_n}$	$\mathbf{I_{map}}$	$\mathbf{I_{sum}}$
UNI	500	0.33	114.38	500	0.38	142.34	500	0.41	160.01
NFQ	381	0.33	120.64	377	0.37	143.31	488	0.39	150.86
DQN	317	0.37	135.90	320	0.38	151.13	326	0.39	159.87
	T = 2 h			**T = 5 h**			**T = 12 h**		
	$\mathbf{I_n}$	$\mathbf{I_{map}}$	$\mathbf{I_{sum}}$	$\mathbf{I_n}$	$\mathbf{I_{map}}$	$\mathbf{I_{sum}}$	$\mathbf{I_n}$	$\mathbf{I_{map}}$	$\mathbf{I_{sum}}$
UNI	500	0.21	57.78	500	0.22	61.80	500	0.20	58.97
NFQ	409	0.25	79.00	402	0.25	84.18	415	0.25	74.06
DQN	459	0.28	81.41	446	0.29	86.20	469	0.25	71.76

Table 7.2: Eros, sensitivity analysis

7.4 Case B: 67P

7.4.1 Detailed results

The basic simulation scenario for comet 67P has the following parameters:

- Rotational period $T_{rot} = 12.4$ h.
- Circular polar orbit, with interest ratio equal to 6.
- Percentage of surface in view in the range $0.3\% - 2.9\%$.

The final performance indexes are displayed in Table 7.3, comparing the different strategies. Again both NFQ and DQN perform better for all three indexes. This confirms the large flexibility and generalizing capability of the proposed strategy, that meets the objectives even when the body shape is extremely irregular and different from training one.

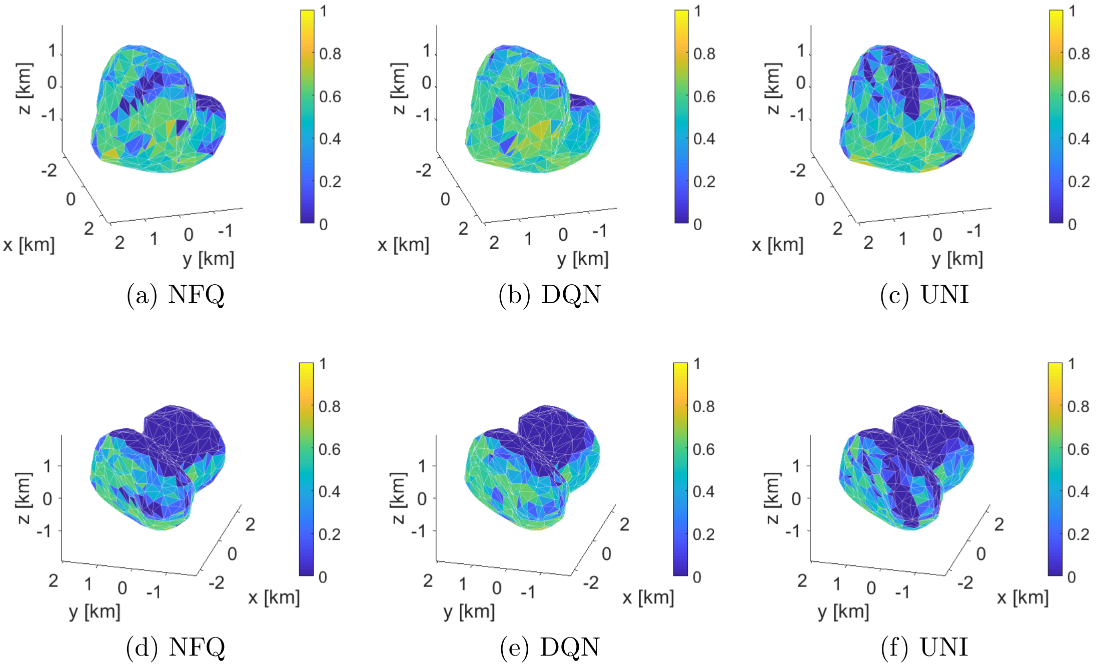
The final mapping index is show for NFQ, DQN and UNI strategies in Figure 7.12. In this case the mapping is hindered by Sun illumination but also by the significant self-shadowing and self-occlusion.

Strategy	NFQ	DQN	UNI	RAND
I_n	365	385	500	500
I_{map}	0.29	0.36	0.25	0.23
I_{sum}	91.55	111.34	67.01	65.06

Table 7.3: Strategies comparison for basic scenario, case B

Trends of global map mean index and number of taken images are shown in Figures 7.13 and 7.14. The mapping quality presents some steps: they correspond to shadow regions. By comparing the two trends it is evident that when the map quality can not increase, NFQ and DQN do not take pictures. This of course leads to a reduced number of images with respect to UNI and RAND.

The memory state is displayed in Figure 7.15. It can be observed that peaks become less and less evident as the mapping proceeds.



(a) NFQ (b) DQN (c) UNI

(d) NFQ (e) DQN (f) UNI

Figure 7.12: 67P, facets final mapping index

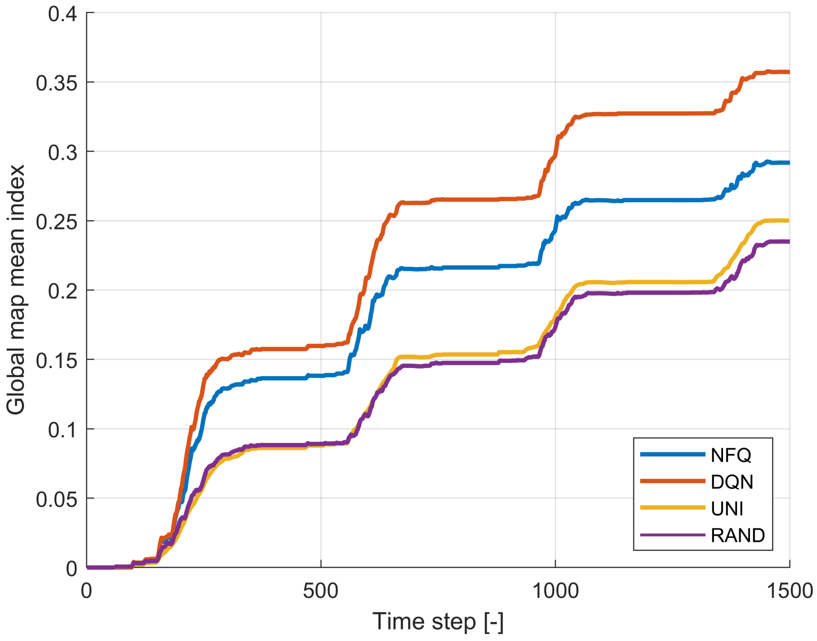


Figure 7.13: 67P, global map mean index

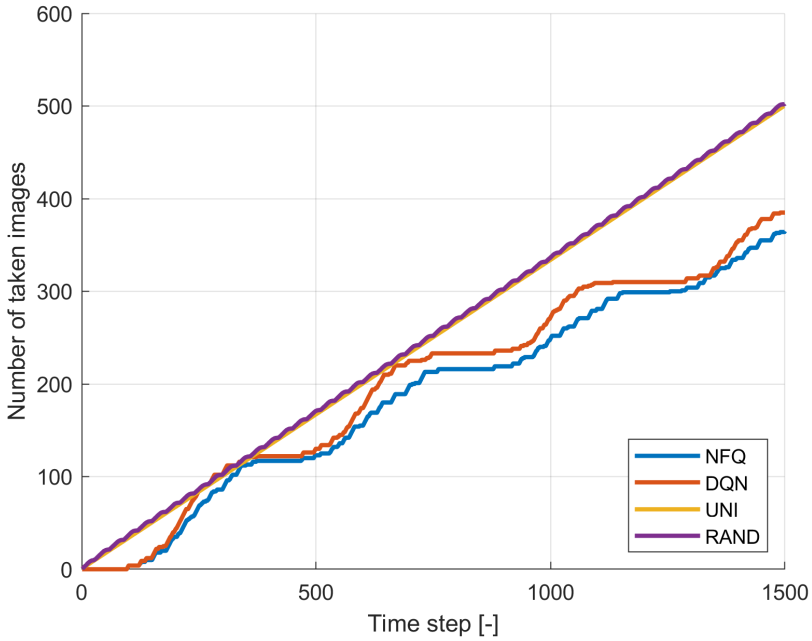


Figure 7.14: 67P, taken images

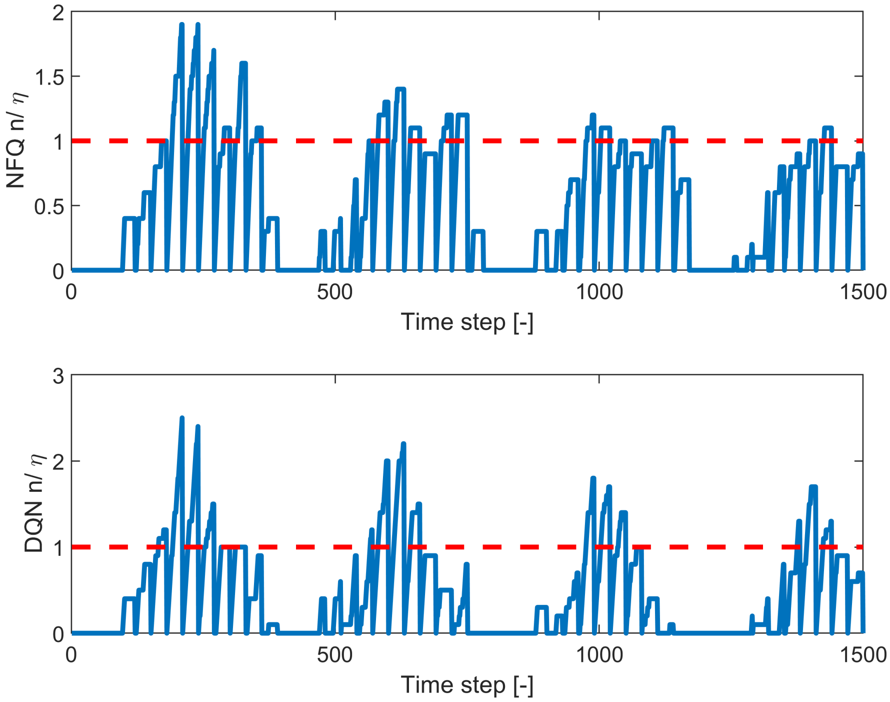


Figure 7.15: 67P, memory

7.4.2 Sensitivity analysis

The sensitivity analysis results are reported in Table 7.4. With interest ratio 8 the percentage of surface in view ranges between 0.9% and 5.2%, while with interest ratios 10 and 12 it is respectively in the intervals 1.7% – 7.4% and 3% – 10.2%.

Looking at the results it is clear that DQN performs better for both final mapping quality and fastening of the process. The typically lower number of images collected by NFQ despite of a possible gain in mapping quality, may be due to the supervised-like learning. The policy prefers not to exceed memory ideal limits, even if violating it could result in a larger return in the long run. So with a simple supervised learning good results are achieved, but the reinforcement performs globally better.

UNI confirms to be the worse strategy. Also in this case increasing the interest ratio I_{map} for the three strategies is almost identical, thus the mapping process can be made more efficient only by reducing data collection.

	i = 30 deg			**i = 60 deg**			**i = 90 deg**		
	$\mathbf{I_n}$	$\mathbf{I_{map}}$	$\mathbf{I_{sum}}$	$\mathbf{I_n}$	$\mathbf{I_{map}}$	$\mathbf{I_{sum}}$	$\mathbf{I_n}$	$\mathbf{I_{map}}$	$\mathbf{I_{sum}}$
UNI	500	0.27	79.67	500	0.27	72.15	500	0.25	67.01
NFQ	356	0.28	95.70	342	0.30	95.24	365	0.29	91.55
DQN	476	0.30	94.65	417	0.36	112.69	385	0.36	111.34
	Interest Ratio = 8			**Interest Ratio = 10**			**Interest Ratio = 12**		
	$\mathbf{I_n}$	$\mathbf{I_{map}}$	$\mathbf{I_{sum}}$	$\mathbf{I_n}$	$\mathbf{I_{map}}$	$\mathbf{I_{sum}}$	$\mathbf{I_n}$	$\mathbf{I_{map}}$	$\mathbf{I_{sum}}$
UNI	500	0.33	102.68	500	0.41	125.66	500	0.40	130.36
NFQ	310	0.32	114.30	290	0.40	126.50	279	0.40	126.52
DQN	285	0.38	134.45	285	0.41	143.01	314	0.41	133.63
	T = 2 h			**T = 5 h**			**T = 12 h**		
	$\mathbf{I_n}$	$\mathbf{I_{map}}$	$\mathbf{I_{sum}}$	$\mathbf{I_n}$	$\mathbf{I_{map}}$	$\mathbf{I_{sum}}$	$\mathbf{I_n}$	$\mathbf{I_{map}}$	$\mathbf{I_{sum}}$
UNI	500	0.21	50.94	500	0.25	62.84	500	0.24	65.59
NFQ	355	0.31	87.28	354	0.29	93.19	379	0.26	84.03
DQN	398	0.36	98.93	381	0.36	110.49	427	0.28	95.00

Table 7.4: 67P-CG, sensitivity analysis

7.5 Case C: Itokawa

7.5.1 Detailed results

The basic simulation scenario for asteroid Itokawa has the following parameters:

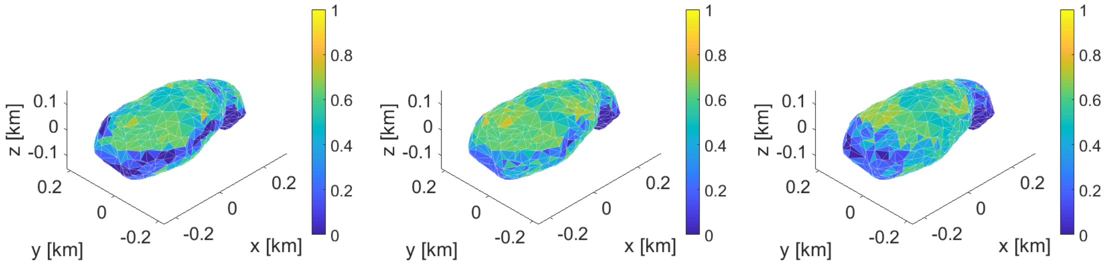
- Rotational period $T_{rot} = 12$ h.
- Circular polar orbit, with interest ratio equal to 6.
- Percentage of surface in view in the range 0.7% – 4.1%.

From results in Table 7.5 it can be seen that as always the RAND strategy is the worse. DQN outperforms NFQ for every performance index. NFQ performance is comparable to UNI, a part from images number.

Strategy	**NFQ**	**DQN**	**UNI**	**RAND**
I_n	361	346	500	500
I_{map}	0.31	0.37	0.32	0.30
I_{sum}	104.74	122.28	104.56	98.49

Table 7.5: Strategies comparison for basic scenario, case C

The shape models in Figure 7.16 again present areas always in shadow. It is important to notice that even in illuminated portions of the body the mapping index with value 1 is never reached: it is an ideal value that according to its definition (chapter 6) is impossible to reach in any realistic simulation. Anyway, it is a good meter to compare the different strategies: facets with very good scores are present in all three cases but DQN provides a better coverage.



(a) NFQ (b) DQN (c) UNI

Figure 7.16: Itokawa, facets final mapping index

Figures 7.17, 7.18 and 7.19 show mapping quality, number of images and memory state. Considerations similar to 67P can be made.

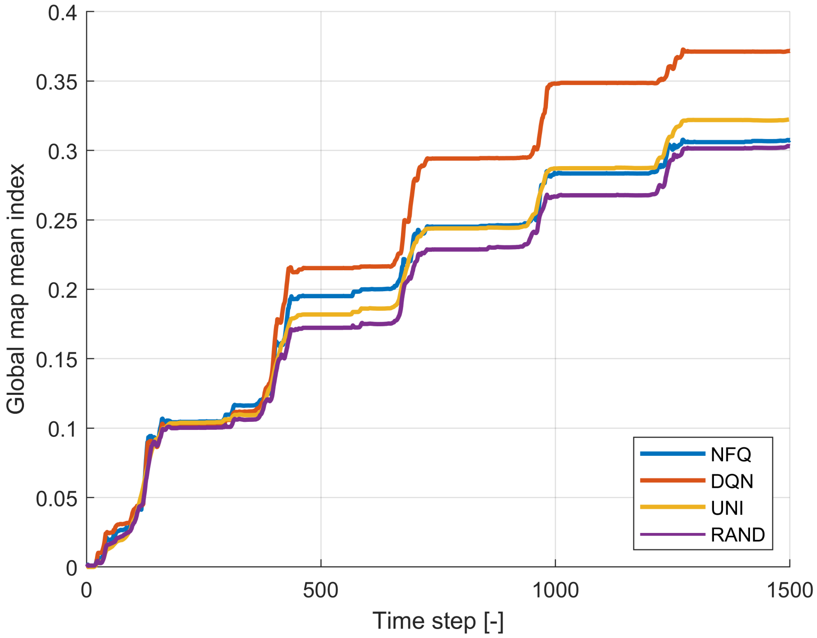


Figure 7.17: Itokawa, global map mean index

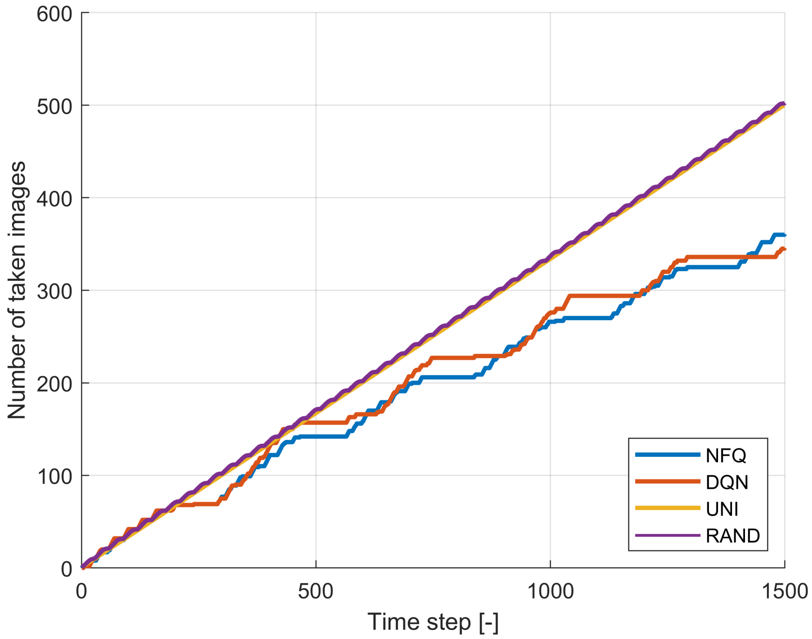


Figure 7.18: Itokawa, taken images

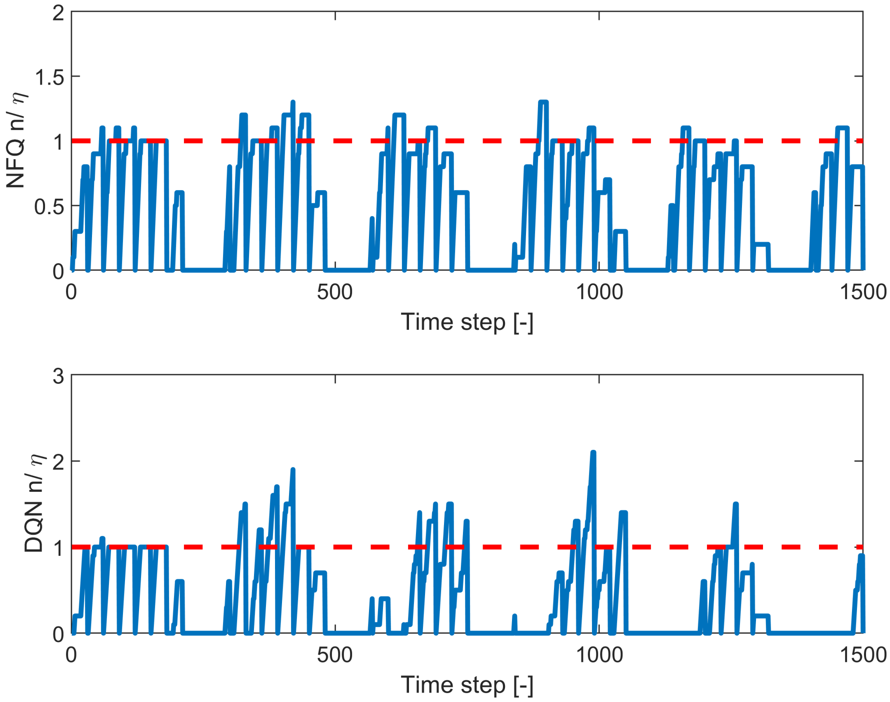


Figure 7.19: Itokawa, memory

7.5.2 Sensitivity analysis

The sensitivity analysis results are reported in Table 7.6. With interest ratio 8 the percentage of surface in view ranges between 1.7% and 6.6%, while with interest ratios 10 and 12 it is respectively in the intervals 3.4% – 10.2% and 5.7% – 15.9%. NFQ and DQN well perform, except from the case with interest ration 12, that is the most critical encountered in all the simulations.

	i = 30 deg			**i = 60 deg**			**i = 90 deg**		
	$\mathbf{I_{n}}$	$\mathbf{I_{map}}$	$\mathbf{I_{sum}}$	$\mathbf{I_{n}}$	$\mathbf{I_{map}}$	$\mathbf{I_{sum}}$	$\mathbf{I_{n}}$	$\mathbf{I_{map}}$	$\mathbf{I_{sum}}$
UNI	500	0.35	118.88	500	0.33	107.09	500	0.32	104.56
NFQ	346	0.32	119.37	364	0.31	108.98	361	0.31	104.74
DQN	404	0.36	134.23	380	0.37	117.17	346	0.37	122.28
	Interest Ratio = 8			**Interest Ratio = 10**			**Interest Ratio = 12**		
	$\mathbf{I_{n}}$	$\mathbf{I_{map}}$	$\mathbf{I_{sum}}$	$\mathbf{I_{n}}$	$\mathbf{I_{map}}$	$\mathbf{I_{sum}}$	$\mathbf{I_{n}}$	$\mathbf{I_{map}}$	$\mathbf{I_{sum}}$
UNI	500	0.40	141.87	500	0.44	164.80	500	0.45	172.72
NFQ	347	0.38	137.50	357	0.42	153.51	443	0.43	158.82
DQN	348	0.41	137.28	338	0.41	150.97	429	0.43	164.49
	T = 2 h			**T = 5 h**			**T = 12 h**		
	$\mathbf{I_{n}}$	$\mathbf{I_{map}}$	$\mathbf{I_{sum}}$	$\mathbf{I_{n}}$	$\mathbf{I_{map}}$	$\mathbf{I_{sum}}$	$\mathbf{I_{n}}$	$\mathbf{I_{map}}$	$\mathbf{I_{sum}}$
UNI	500	0.29	82.19	500	0.32	102.40	500	0.32	104.61
NFQ	345	0.33	108.89	357	0.32	111.15	353	0.31	108.94
DQN	336	0.38	118.79	315	0.38	130.02	340	0.36	116.76

Table 7.6: Itokawa, sensitivity analysis

7.6 Case D: Bennu

7.6.1 Detailed results

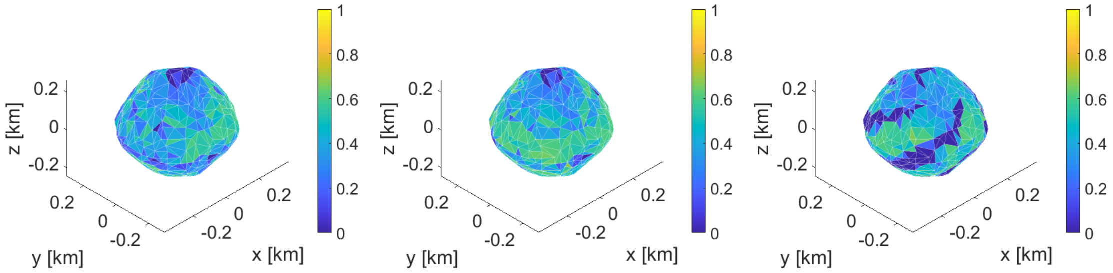
Bennu is a diamond-shaped asteroid, so the interest ratio is kept larger than in the other simulations in order to have the typical portion of body in view. The basic simulation has parameters:

- Rotational period $T_{rot} = 4.3$ h.
- Circular polar orbit, with interest ratio equal to 10.
- Percentage of surface in view in the range 1.2% – 4.4%.

Results for the basic simulation are reported in Table 7.7. DQN outperforms the other strategies. In this case RAND performs better than UNI. In fact in Figure 7.20 it can be seen that UNI is not able to grant complete coverage of the body. This is due to the imaging frequency, which is a consequence of the chosen ideal number of images η in the storage time $T_{storage}$.

Strategy	NFQ	DQN	UNI	RAND
I_n	289	268	500	500
I_{map}	0.43	0.47	0.37	0.44
I_{sum}	154.85	172.23	132.74	146.59

Table 7.7: Strategies comparison for basic scenario, case D



(a) NFQ (b) DQN (c) UNI

Figure 7.20: Bennu, facets final mapping index

Figures 7.21, 7.22 and 7.23 show mapping quality, number of images and memory state. Considerations similar to the other test cases can be made. Please note that NFQ and DQN almost always stay under the ideal memory usage (see Figure 7.23).

7.6.2 Sensitivity analysis

The sensitivity analysis results are reported in Table 7.8. 7.8. With interest ratio 8 the percentage of surface in view ranges between 1.6% and 5.2%, while with interest ratios 10 and 12 it is respectively in the intervals 2.5% − 6.6% and 3.5% − 8.1%. The generalizing capabilities of the planning policies are confirmed also for case D: when the scenario allows a better mapping the gain in performance is more evident; when the surface portion in view is large, the final mapping quality achieved is similar for all the techniques. NFQ and DQN allow for a remarkable saving in amount of images and in particular DQN outperforms the other policies in mapping fastening (larger I_{sum}).

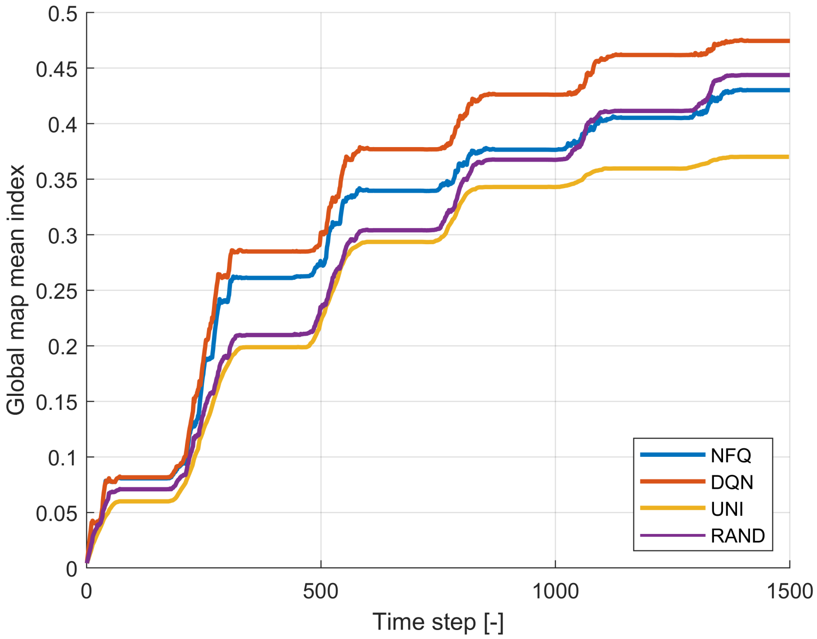


Figure 7.21: Bennu, global map mean index

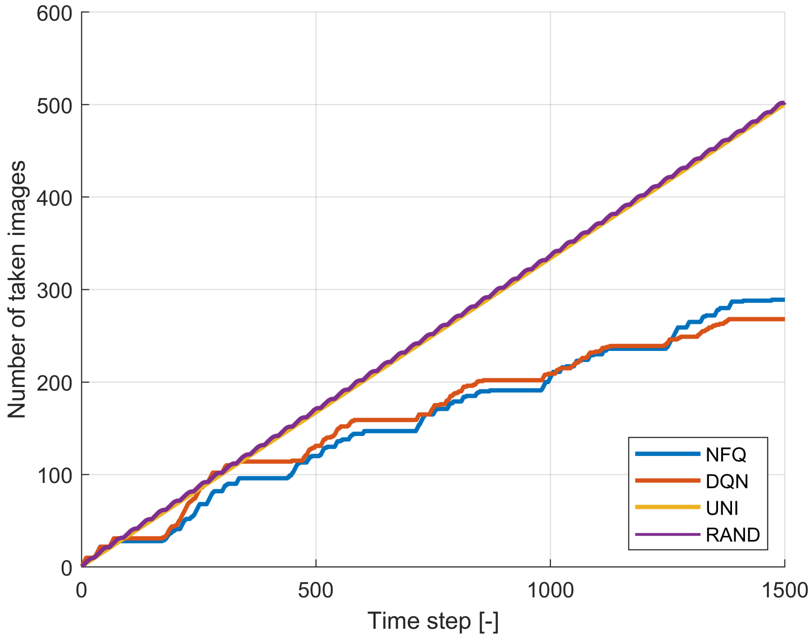


Figure 7.22: Bennu, taken images

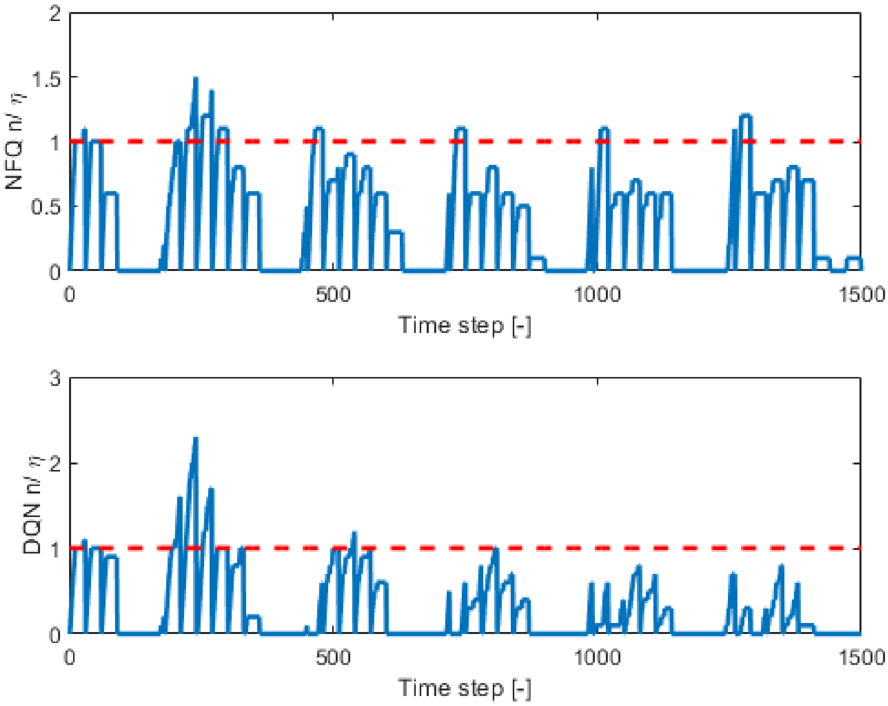


Figure 7.23: Bennu, memory

	i = 30 deg			i = 60 deg			i = 90 deg		
	I_n	I_{map}	I_{sum}	I_n	I_{map}	I_{sum}	I_n	I_{map}	I_{sum}
UNI	500	0.33	126.86	500	0.42	147.55	500	0.37	132.74
NFQ	240	0.33	129.35	225	0.41	153.51	289	0.43	154.85
DQN	316	0.34	140.14	258	0.44	167.55	268	0.47	172.23
	Interest Ratio = 11			**Interest Ratio = 13**			**Interest Ratio = 15**		
	I_n	I_{map}	I_{sum}	I_n	I_{map}	I_{sum}	I_n	I_{map}	I_{sum}
UNI	500	0.47	163.96	500	0.50	191.41	500	0.51	204.92
NFQ	309	0.47	169.14	326	0.50	187.65	369	0.50	202.05
DQN	237	0.48	182.29	230	0.49	193.11	206	0.49	201.60
	T = 2 h			**T = 5 h**			**T = 12 h**		
	I_n	I_{map}	I_{sum}	I_n	I_{map}	I_{sum}	I_n	I_{map}	I_{sum}
UNI	500	0.45	130.30	500	0.47	146.53	500	0.47	151.14
NFQ	297	0.44	150.60	296	0.42	151.48	293	0.40	145.64
DQN	292	0.48	165.38	281	0.48	171.98	288	0.47	171.08

Table 7.8: Bennu, sensitivity analysis

7.7 Final remarks

Some final remarks on the above presented results are provided in this section. The policies obtained through DRL present the desired behavior, being able to achieve all the objectives that were goal of the learning. Results show a substantial improvement of mapping operations with both techniques and for all the considered small bodies. In general DQN outperforms NFQ, allowing a faster mapping and a better final mapping index. The policies performance has been validated with extensive tests that include typical scenarios of the global mapping phase but also cases in which the mapping is limited. When the percentage of surface in view is higher than the usual of the mapping phase, the networks show good generalizing capabilities with a performance comparable to benchmarks. In typical scenarios DQN and NFQ show an excellent performance with respect to benchmarks, significantly reducing the collected data amount, improving the image quality for SPC and speeding up their collection.

Problems with memory handling are overcome accepting some peaks with respect to the initial arbitrary threshold. Images in excess between one downlink and the next one can be stored and sent on ground in successive moments when no or few images are collected. Results show that peaks in memory usage are acceptable, considering that they allow a significant performance improvement and that the overall number of images never exceeds the ideal data amount of the mapping campaign.

7.8 Computational time

An analysis of the algorithm computational performance is here presented. This is a key aspect for a possible on-board implementation: if the decision time is too long with respect to the relative dynamics, then the agent will not be able to take the picture at the expected location. The algorithm is implemented in MATLAB and run on an Intel® Core™ i7-5500U CPU, clocked at 2.4 GHz, paired to a 16 GB DDR3 memory. The performance in a real application will be different, due to the different programming language and hardware; anyway this analysis can still be useful to catch the orders of magnitude and the on-line feasibility of the proposed method. It is important to highlight that the computational time varies depending on the overall number of images taken, the portion of the surface observed and the resolution of the employed shape model. The computational time employed to take single decisions is shown in figure 7.24 for a 1000 facets sphere shape model and 1.5% of surface in the camera field fo view. The overall number of decisions is 500 and for this test each decision is enforced to be an image collection. A spherical shape model is a good test case because the percentage of surface in view does not change, keeping a constant distance from the body. Maximum, average and minimum values are reported in table 7.9: all the three values are small, so the algorithm proves to be fast. The related histogram is displayed in Figure 7.24. Changes in the computational time are mainly linked to the calculation of facets in view.

Time $[s]$	
Average	0.0335
Minimum	0.0074
Maximum	0.1325

Table 7.9: Computational time

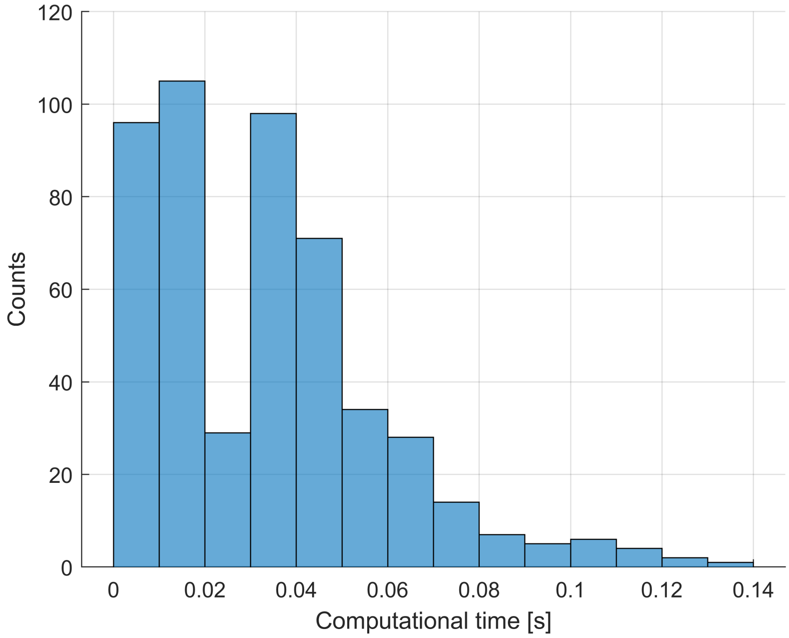


Figure 7.24: Computational time for the single decision, with 1.5% of the surface in view

Then, another analysis is done, increasing the percentage of surface in view. All the other parameters are kept as before. The mean time linearly increases with the surface portion, as displayed in Figure 7.25. The mean time can be about two orders of magnitude larger when the surface portion in view is large. Please note that percentages larger than 10% are not typical of the global mapping phase.

The shape model resolution of 1000 facets used in this thesis is believed already a good compromise, since 1% of the surface corresponds to roughly 10 facets. Anyway, in Figure 7.26 the computational time is shown for a spherical body with 1.5% of surface in view: increasing the number of facets of the polyhedron the time increases quadratically, as expected from the algorithm structure presented in chapter 6. Each point of the interpolation is again the mean time over 500 decisions.

Finally in Figure 7.27 a 50000 time steps simulation is run. As it can be observed the computational time increases with the time steps, because of the memory of past actions is kept. Anyway the increase is visible only after about 10000 steps and it is not as demanding as in the previous situations.

The computational analyses here reported highlight which are the different parameters that affect the computational time and show well visible trends. The most critical issue is believed to be the surface portion in view, since this parameter is the most likely to be varied. A delay in the decision making can cause imaging of a different area with respect to the expected one. For an hypothetical fast rotating spherical body with 2 h rotational period, the surface displacement in 1 s is of about the 0.1% of the characteristic dimension. Please observe that if the surface portion in view is large, such displacement is not significant; while for a small area in view the computational time is much smaller. Of course more accurate considerations can be made once the spacecraft dynamics is fixed.

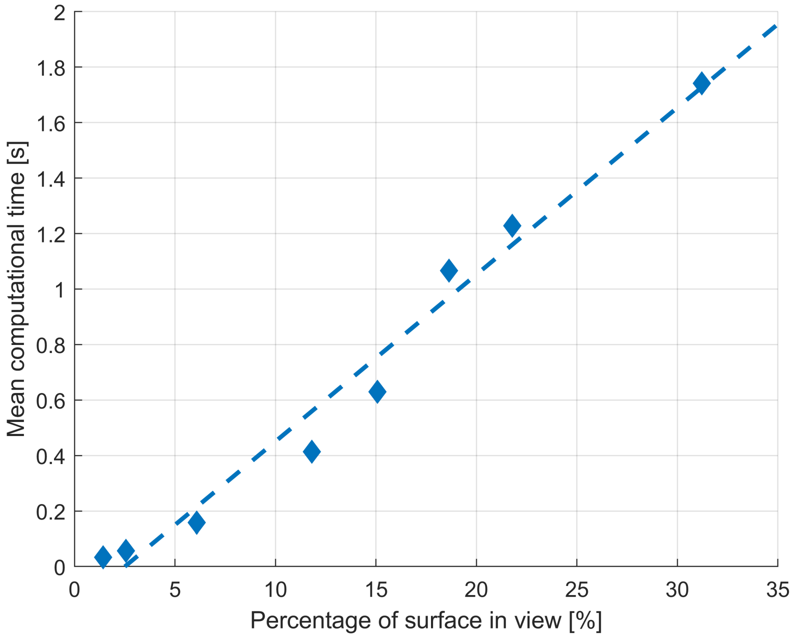


Figure 7.25: Computational time for the single decision, varying the surface portion in view. Each point is the mean computational time over 500 decisions.

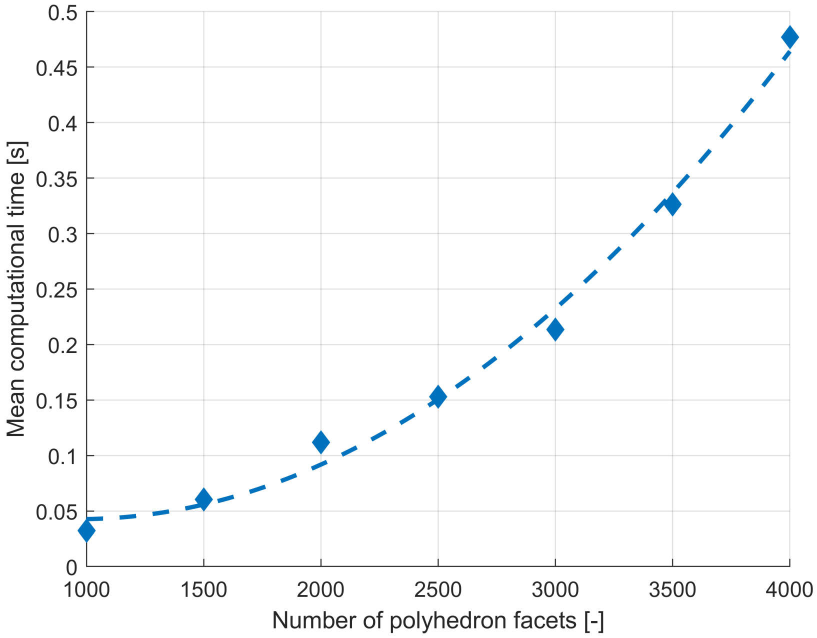


Figure 7.26: Computational time for the single decision, varying the polyhedron facets. Each point is the mean computational time over 500 decisions.

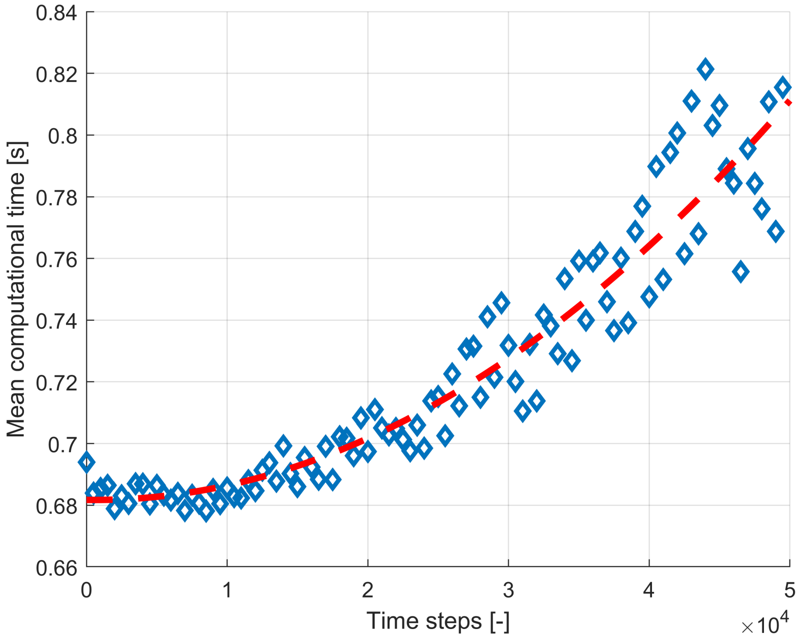


Figure 7.27: Computational time for the single decision, increasing the simulation length. Each point is the mean computational time over the past 500 decisions.

8 Conclusion

The present thesis proposes an innovative autonomous planning architecture for small bodies exploration, focused on imaging for shape model reconstruction. The proposed method exploits DRL to design a policy that improves mapping quality, fastens the mapping campaign and reduces the amount of collected data. These objectives are met by selecting the observation times while orbiting around the body, according to the relative camera-body pose, the surface illumination conditions and considering data storage and downlink limitations.

Autonomous small bodies exploration is framed as a POMDP problem, in analogy with terrestrial robotics active SLAM. A very general framework is provided, critically analyzing the additional challenges presented by space applications, linked to the harsh operative environment and complexity of space systems. The relative dynamics between spacecraft and body reveals to be an important obstacle to autonomous operations, because of its incomplete knowledge before the mission and its chaotic nature, that could mine the safety of an autonomous spacecraft. Moreover, the great variability of accelerations order of magnitude leaves no space for the definition of a simple, general and realistic dynamical model. Possible ways to ease on-ground shape model reconstruction procedures are deduced through the examination of small bodies mapping operations. In particular, the optimization of images collection, lacking in the state of the art operations, is identified as a promising step forward towards a more efficient approach to mapping. Illumination and viewing conditions of the body surface emerge as key aspects for the realization of SPC algorithm.

In light of such considerations, a novel reduced planning problem is defined, choosing DRL as solution method. The proposed planning architecture does not require re-training on-board, thanks to the introduction in the algorithm of information preprocessing. Incorporation of prior knowledges eases the ANN tasks; in parallel the algorithm architecture is kept light and suitable for reduced computational resources.

The extensive numerical tests on policies obtained through NFQ and DQN have shown substantial improvements to small imaging operations with respect to benchmarks solutions, for all the proposed objectives. DRL confirms to be a valid approach for solving the decision process problem, merging the advantages of RL and the ones of ANNs. Good planning policies are found for an otherwise computationally intractable problem, letting the agent interact with the environment and learn from experiences. In addition, the use of an ANN as function approximator results in a good generalization capability, verified through sensitivity analyses. The implementation of DQN and NFQ learning processes has emphasized the importance of reward modeling for the achievement of the desired behavior. The proper definition of prohibited states for the agent punishment is essential to trade-off between conflicting objectives. Design loops have lead to the conclusion that a flexible data management is the best way to increase mapping quality and limit the collected images at the same time.

In conclusion, the so far achieved results of the proposed approach reveal the methodology to be a promising step forward in autonomous operations, helping in decreasing the human effort during the mapping phases of unknown small bodies and increasing imaging exploitation efficiency with a simple and flexible approach.

The research contributions introduced by this thesis are here summarized:

- After a deep examination of the topics involved in autonomous small bodies mapping, a novel planning architecture with a DRL-based policy is proposed. The merits of this architecture are the decoupling of the decision-making process from spacecrafts dynamics, the autonomy improvement with very low risks for the mission and the general validity of the planning framework, which is mission-independent and does not require learning during operations.

- An extensive numerical testing is addressed. The designed policies prove to outperform benchmarks in all the scenarios typical for global mapping, that include different body shapes, relative dynamics, body coverage and illumination conditions. Limits of the algorithm applicability are tested as well.

- The planning architecture is designed to be light. A computational analysis is carried out, stressing the algorithm parameters that affect the computational time. The proposed algorithm is computationally efficient and shows a fast performance within the identified limits. The other literature works do not examine computational time of the proposed algorithm [31] or propose a solution that can not be realized with the on-board computational power [36].

Future work

Small bodies autonomous mapping is a novel research field, which leaves large spaces to further improvements. Some guidelines are here provided for future research on this challenging topic.

Learning process improvement The numerical tests presented in this thesis have shown that DRL is a promising tool for the autonomous small bodies mapping application. It is believed that several of these algorithms are suitable for the problem and a deeper investigation on such methods would allow a constructive comparison between different techniques. Anyway, beyond the algorithm choice, a more important aspect is the learning process improvement. This can be done in two directions. First, by exploiting a more realistic orbital dynamics that allows a complete body mapping during the learning and by refining the control interval. Second, by optimizing the algorithm hyperparameters (especially for the learning stop) in order to improve the policy performance and avoid at least in part the parameters hand-tuning.

Further validation It has been shown that the methodology here proposed betters the mapping process in the direction of reward, but how the reward actually meets real necessities needs further developments in order to be assessed and validated. DRL algorithms are necessarily based on hand-tuned reward functions. For this reason even if they maximize the proposed objectives, in practice they may not lead to the desired behavior. Therefore, the obtained policy needs further validation. This can be done through simulations which include the generation of synthetic images of the small body. The small body shape can be reconstructed from such images and compared with the original model, taken as a groundtruth. In this way it is possible to verify the reward functions goodness and improve their definition if necessary.

Planning architecture extension In the present work the restrictive assumption of camera pointing towards the body center has been considered. In practice, the appropriate camera control can lead to further improvements of the body imaging. In fact, changing the camera orientation allows a direct control of emission angles, while continuous body center pointing leads to an always low emission in case of a quasi-spherical body shape, thus preventing an optimal mapping. A change of the pose would overcome this issue. Moreover, this additional degree of freedom may have beneficial effects on the body coverage and mapping fastening as well, without risks and complexity related to a direct spacecraft attitude control. Therefore extending the planning architecture including also camera control is considered a good direction for future research. In addition, some work may be done to assess how uncertainties in the state belief and delays in the decision making affect the mapping performance.

Appendices

A

The RPROP algorithm

The resilient back-propagation algorithm is here presented, following the steps in [80]. The RPROP is an adaptive learning algorithm and consists of two steps: adaptation and weight update rules. The weight is updated directly according to an update-value Δ_{ij} and not proportionally to the gradient as in classical back-propagation.

The weight update rules are the following:

$$\Delta w_{ij} = \begin{cases} -\Delta_{ij}(k) & \text{if } \frac{\partial \mathcal{E}}{\partial w_{ij}}(k) > 0 \\ +\Delta_{ij}(k) & \text{if } \frac{\partial \mathcal{E}}{\partial w_{ij}}(k) < 0 \\ -\Delta w_{ij}(k-1) & \text{if } \frac{\partial \mathcal{E}}{\partial w_{ij}}(k-1)\frac{\partial \mathcal{E}}{\partial w_{ij}}(k) < 0 \\ 0 & \text{otherwise} \end{cases} \tag{A.1}$$

When the error is increasing, its derivative with respect to the weight is positive, hence the weight is decreased by its update value. When the error decreases, the derivative is negative and the weight is increased. When the derivative changes sign, the previous step was too large and the minimum was missed, so the previous update is reverted and the update-value is not adapted in the next step. In all the other cases instead, the update-value is adapted based on local gradient information with the adaptation rule:

$$\Delta_{ij} = \begin{cases} \eta^{+}\Delta_{ij}(k-1) & \text{if } \frac{\partial \mathcal{E}}{\partial w_{ij}}(k-1)\frac{\partial \mathcal{E}}{\partial w_{ij}}(k) > 0 \\ \eta^{-}\Delta_{ij}(k-1) & \text{if } \frac{\partial \mathcal{E}}{\partial w_{ij}}(k-1)\frac{\partial \mathcal{E}}{\partial w_{ij}}(k) < 0 \\ \Delta_{ij}(k-1) & \text{otherwise} \end{cases} \tag{A.2}$$

where $0 < \eta^{-} < 1 < \eta^{+}$. When the partial derivative of the error keeps the same sign than the previous step, the magnitude of the update-value is increased, in order to fasten the learning process. When the partial derivative changes sign, the algorithm has jumped over a local minimum and so the update-value is decreased. The RPROP algorithm scheme is the following:

Algorithm 3 RPROP Algorithm

```
for all weights and biases do
    if $\frac{\partial\mathcal{E}}{\partial w_{ij}}(k-1)\frac{\partial\mathcal{E}}{\partial w_{ij}}(k) > 0$ then
        $\Delta_{ij}(k) = \min\Big(\eta^{+}\Delta_{ij}(k-1), \Delta_{max}\Big)$
        $\Delta w_{ij}(k) = -\Delta_{ij}(k)\text{sign}\Big(\frac{\partial\mathcal{E}}{\partial w_{ij}}(k)\Big)$
        $w_{ij}(k+1) = w_{ij}(k) + \Delta w_{ij}(k)$
    else if $\frac{\partial\mathcal{E}}{\partial w_{ij}}(k-1)\frac{\partial\mathcal{E}}{\partial w_{ij}}(k) < 0$ then
        $\Delta_{ij}(k) = \max\Big(\eta^{-}\Delta_{ij}(k-1), \Delta_{min}\Big)$
        $w_{ij}(k+1) = w_{ij}(k) - \Delta w_{ij}(k-1)$
        $\frac{\partial\mathcal{E}}{\partial w_{ij}}(k) = 0$
    else if $\frac{\partial\mathcal{E}}{\partial w_{ij}}(k-1)\frac{\partial\mathcal{E}}{\partial w_{ij}}(k) = 0$ then
        $\Delta w_{ij}(k) = -\Delta_{ij}(k)\text{sign}\Big(\frac{\partial\mathcal{E}}{\partial w_{ij}}(k)\Big)$
        $w_{ij}(k+1) = w_{ij}(k) + \Delta w_{ij}(k)$
    end if
end for
```

Please note that the procedure here described always refers to weights and that for biases it is identical.

The algorithm is proved to be very robust to the choice of its parameters. The most common values adopted in literature are the ones proposed in [80]. The update-value is set to $\Delta_0 = 0.1$ and its limits are set to $\Delta_{max} = 50$ and $\Delta_{min} = 10^{-6}$, in order to avoid overflow and underflow problems for floating points variables. The increasing and decreasing factors are $\eta^{-} = 0.5$ (so the update-value is halved) and $\eta^{+} = 1.2$, to fasten the update-value growth while keeping a stable learning.

Bibliography

[1] D. Yeomans, P. Antreasian, J. Barriot, S. Chesley, D. Dunham, R. Farquhar, J. Giorgini, C. Helfrich, A. Konopliv, J. McAdams, *et al.*, "Radio science results during the NEAR-Shoemaker spacecraft rendezvous with Eros", *Science*, vol. 289, no. 5487, pp. 2085–2088, 2000. DOI: 10.1126/science.289.5487.2085.

[2] C. Russell, F. Capaccioni, A. Coradini, M. De Sanctis, W. Feldman, R. Jaumann, H. Keller, T. McCord, L. McFadden, S. Mottola, *et al.*, "Dawn mission to Vesta and Ceres", *Earth, Moon, and Planets*, vol. 101, no. 1-2, pp. 65–91, 2007. DOI: 10.1007/s11038-007-9151-9.

[3] K. Glassmeier, H. Boehnhardt, D. Koschny, E. Kührt, and I. Richter, "The Rosetta mission: flying towards the origin of the solar system", *Space Science Reviews*, vol. 128, no. 1-4, pp. 1–21, 2007. DOI: 10.1007/s11214-006-9140-8.

[4] Y. Tsuda, M. Yoshikawa, M. Abe, H. Minamino, and S. Nakazawa, "System design of the Hayabusa 2–Asteroid sample return mission to 1999 JU3", *Acta Astronautica*, vol. 91, pp. 356–362, 2013. DOI: 10.1016/j.actaastro.2013.06.028.

[5] D. Lauretta, S. Balram-Knutson, E. Beshore, W. V. Boynton, C. D. d'Aubigny, D. DellaGiustina, H. Enos, D. Golish, C. Hergenrother, E. Howell, *et al.*, "OSIRIS-REx: sample return from asteroid (101955) Bennu", *Space Science Reviews*, vol. 212, no. 1-2, pp. 925–984, 2017. DOI: 10.1007/s11214-017-0405-1.

[6] *NASA's future missions.* [Online]. Available: https://www.jpl.nasa.gov/missions (visited on 03/18/2019).

[7] *ESA's Hera asteroid mission.* [Online]. Available: http://www.esa.int/Our_Activities/Operations/Space_Safety_Security/Hera (visited on 03/18/2019).

[8] *Rosetta Mission Data Archive*, 2016. [Online]. Available: ftp://psa.esac.esa.int/pub/mirror/INTERNATIONAL-ROSETTA-MISSION (visited on 02/28/2019).

[9] R. P. de Santayana and M. Lauer, "Optical measurements for rosetta navigation near the comet", in *Proceedings of the 25th International Symposium on Space Flight Dynamics (ISSFD), Munich*, 2015.

[10] F. Preusker, F. Scholten, K. D. Matz, T. Roatsch, K. Willner, S. Hviid, J. Knollenberg, L. Jorda, P. J. Gutiérrez, E. Kührt, *et al.*, “Shape model, reference system definition, and cartographic mapping standards for comet 67P/Churyumov-Gerasimenko–Stereo-photogrammetric analysis of Rosetta/OSIRIS image data”, *Astronomy & Astrophysics*, vol. 583, A33, 2015. DOI: 10.1051/0004-6361/201526349.

[11] C. Wong, E. Yang, X. T. Yan, and D. Gu, “Adaptive and intelligent navigation of autonomous planetary rovers–A survey”, in *2017 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, IEEE, 2017, pp. 237–244.

[12] R. Gonzalez and K. Iagnemma, “Slippage estimation and compensation for planetary exploration rovers. State of the art and future challenges”, *Journal of Field Robotics*, vol. 35, no. 4, pp. 564–577, 2018. DOI: doi.org/10.1002/rob.21761.

[13] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, *et al.*, “Towards fully autonomous driving: Systems and algorithms”, in *2011 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2011, pp. 163–168.

[14] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, “Unsupervised learning of depth and ego-motion from video”, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1851–1858.

[15] P. Lunghi, M. Ciarambino, and M. Lavagna, “A multilayer perceptron hazard detector for vision-based autonomous planetary landing”, *Advances in Space Research*, vol. 58, no. 1, pp. 131–144, 2016. DOI: 10.1016/j.asr.2016.04.012.

[16] B. K. Horn, “Shape from shading: A method for obtaining the shape of a smooth opaque object from one view”, PhD thesis, MIT, 1970.

[17] B. Giese, J. Oberst, R. Kirk, and W. Zeitler, “The topography of asteroid Ida: A comparison between photogrammetric and two-dimensional photoclinometric image analysis”, *International Archives of Photogrammetry and Remote Sensing*, vol. 31, B3, 1996.

[18] B. Giese, J. Oberst, T. Roatsch, G. Neukum, J. Head, and R. Pappalardo, “The local topography of Uruk Sulcus and Galileo Regio obtained from stereo images”, *Icarus*, vol. 135, no. 1, pp. 303–316, 1998. DOI: 10.1006/icar.1998.5967.

[19] R. Gaskell, “Automated landmark identification for spacecraft navigation”, *Advances in the Astronautical Sciences*, vol. 109, pp. 1749–1756, Jan. 2002.

[20] C. Capanna, G. Gesquière, L. Jorda, P. Lamy, and D. Vibert, “Three-dimensional reconstruction using multiresolution photoclinometry by deformation”, *The Visual Computer*, vol. 29, no. 6-8, pp. 825–835, 2013. DOI: 10.1007/s00371-013-0821-5.

[21] T. A. Pavlak, S. B. Broschart, and G. Lantoine, “Quantifying mapping orbit performance in the vicinity of primitive bodies”, in *25th AAS/AIAA Space Flight Mechanics Meeting*, Pasadena, CA: Jet Propulsion Laboratory, National Aeronautics and Space Administration, 2015.

[22] S. B. Broschart, G. Lantoine, and D. J. Grebow, “Quasi-terminator orbits near primitive bodies”, *Celestial Mechanics and Dynamical Astronomy*, vol. 120, no. 2, pp. 195–215, 2014. DOI: `10.1007/s10569-014-9574-3`.

[23] R. Smith, M. Self, and P. Cheeseman, “Estimating Uncertain Spatial Relationships in Robotics”, *Autonomous Robot Vehicles*, vol. 1, pp. 435–461, Jan. 1986. DOI: `10.1109/ROBOT.1987.1087846`.

[24] H. Durrant-Whyte and T. Bailey, “Simultaneous localization and mapping: part I”, *IEEE robotics & automation magazine*, vol. 13, no. 2, pp. 99–110, 2006. DOI: `10.1109/MRA.2006.1638022`.

[25] H. J. S. Feder, J. J. Leonard, and C. M. Smith, “Adaptive mobile robot navigation and mapping”, *The International Journal of Robotics Research*, vol. 18, no. 7, pp. 650–668, 1999.

[26] R. Martinez-Cantin, N. de Freitas, E. Brochu, J. Castellanos, and A. Doucet, “A Bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot”, *Autonomous Robots*, vol. 27, no. 2, pp. 93–103, 2009. DOI: `10.1007/s10514-009-9130-2`.

[27] H. Carrillo, I. Reid, and J. A. Castellanos, “On the comparison of uncertainty criteria for active SLAM”, in *2012 IEEE International Conference on Robotics and Automation*, IEEE, 2012, pp. 2080–2087.

[28] F. Baldini, A. Harvard, S.-J. Chung, I. Nesnas, and S. Bhaskaran, “Autonomous Small Body Mapping and Spacecraft Navigation”, International Astronautical Federation, 2018.

[29] C. Cocaud and T. Kubota, “SURF-based SLAM scheme using octree occupancy grid for autonomous landing on asteroids”, in *Proceedings of the 10th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 2010.

[30] C. Cocaud and T. Kubota, “Autonomous navigation near asteroids based on visual SLAM”, in *Proceedings of the 23rd International Symposium on Space Flight Dynamics, Pasadena, California*, 2012.

[31] V. Pesce, A.-a. Agha-mohammadi, and M. Lavagna, “Autonomous navigation & mapping of small bodies”, in *2018 IEEE Aerospace Conference*, IEEE, 2018, pp. 1–10. DOI: `10.1109/AERO.2018.8396797`.

[32] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction.* MIT press, 2018.

[33] T. Kollar and N. Roy, “Trajectory optimization using reinforcement learning for map exploration”, *The International Journal of Robotics Research*, vol. 27, no. 2, pp. 175–196, 2008. DOI: `10.1177/0278364907087426`.

[34] T. Kollar and N. Roy, “Using reinforcement learning to improve exploration trajectories for error minimization”, in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, IEEE, 2006, pp. 3338–3343.

[35] A. E. Sallab, M. Abdou, E. Perot, and S. Yogamani, “Deep reinforcement learning framework for autonomous driving”, *Electronic Imaging*, vol. 2017, no. 19, pp. 70–76, 2017. DOI: `10.2352/ISSN.2470-1173.2017.19.AVM-023`.

[36] D. M. Chan and A.-a. Agha-mohammadi, “Autonomous Imaging and Mapping of Small Bodies Using Deep Reinforcement Learning”, in *2019 IEEE Aerospace Conference*, 2019.

[37] J. Gal-Edd and A. Cheuvront, “The OSIRIS-REx asteroid sample return mission operations design”, in *2015 IEEE Aerospace Conference*, IEEE, 2015, pp. 1–9.

[38] B. S. John Ivens, *ORX OCAMS Instrument Kernel*, 2016. [Online]. Available: `https://naif.jpl.nasa.gov/pub/naif/ORX/kernels/ik/orx_ocams_v04.ti` (visited on 02/27/2019).

[39] H. U. Keller, C. Barbieri, P. Lamy, H. Rickman, R. Rodrigo, K.-P. Wenzel, H. Sierks, M. F. A’Hearn, F. Angrilli, M. Angulo, *et al.*, “OSIRIS–The scientific camera system onboard Rosetta”, *Space Science Reviews*, vol. 128, no. 1-4, pp. 433–506, 2007. DOI: `10.1007/s11214-006-9128-4`.

[40] P. L. Lamy, I. Toth, B. J. Davidsson, O. Groussin, P. Gutiérrez, L. Jorda, M. Kaasalainen, and S. C. Lowry, “A portrait of the nucleus of comet 67P/Churyumov-Gerasimenko”, *Space science reviews*, vol. 128, no. 1-4, pp. 23–66, 2007. DOI: `10.1007/s11214-007-9146-x`.

[41] M. Lauer, S. Kielbassa, and R. Pardo, “Optical Measurements for Attitude Control and Shape Reconstruction at the Rosetta Flyby of Asteroid Lutetia”, in *23rd International Symposium on Space Flight Dynamics*, 2012, pp. 1–15.

[42] S. Savarese, M. Andreetto, H. Rushmeier, F. Bernardini, and P. Perona, “3d reconstruction by shadow carving: Theory and practical evaluation”, *International journal of computer vision*, vol. 71, no. 3, pp. 305–336, 2007. DOI: `10.1007/s11263-006-8323-9`.

[43] R. W. Gaskell, O. S. Barnouin-Jha, D. J. Scheeres, A. S. Konopliv, T. Mukai, S. Abe, J. Saito, M. Ishiguro, T. Kubota, T. Hashimoto, J. Kawaguchi, M. Yoshikawa, K. Shirakawa, T. Kominato, N. Hirata, and H. Demura, “Characterizing and navigating small bodies with imaging data”, *Meteoritics and Planetary Science*, vol. 43, no. 6, pp. 1049–1061, 2008. DOI: `10.1111/j.1945-5100.2008.tb00692.x`.

[44] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision.* Cambridge university press, 2003.

[45] R. P. de Santayana, M. Lauer, P. Muñoz, and F. Castellini, "Surface Characterization and Optical Navigation at the Rosetta Flyby of Asteroid Lutetia", in *International Symposium on Space Flight Dynamics*, 2014.

[46] D. J. Scheeres, S. J. Ostro, R. Hudson, and R. A. Werner, "Orbits close to asteroid 4769 Castalia", *Icarus*, vol. 121, no. 1, pp. 67–87, 1996. DOI: `10.1006/icar.1996.0072`.

[47] D. J. Scheeres, B. Williams, and J. Miller, "Evaluation of the dynamic environment of an asteroid: Applications to 433 Eros", *Journal of Guidance, Control, and Dynamics*, vol. 23, no. 3, pp. 466–475, 2000. DOI: `10.2514/2.4552`.

[48] D. J. Scheeres, "Orbital mechanics about small bodies", *Acta Astronautica*, vol. 72, pp. 1–14, 2012. DOI: `10.1016/j.actaastro.2011.10.021`.

[49] D. J. Scheeres, "Orbit mechanics about asteroids and comets", *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 3, pp. 987–997, 2012. DOI: `10.2514/1.57247`.

[50] D. J. Scheeres, B. Sutter, and A. Rosengren, "Design, dynamics and stability of the OSIRIS-REx sun-terminator orbits", *Advances in the Astronautical Sciences*, vol. 148, pp. 3263–3282, 2013.

[51] *JPL Small-Body Database Browser.* [Online]. Available: `https://ssd.jpl.nasa.gov/sbdb.cgi` (visited on 11/07/2018).

[52] P. Pravec and A. W. Harris, "Fast and slow rotation of asteroids", *Icarus*, vol. 148, no. 1, pp. 12–20, 2000. DOI: `10.1006/icar.2000.6482`.

[53] S. B. Broschart and D. J. Scheeres, "Control of hovering spacecraft near small bodies: application to asteroid 25143 Itokawa", *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 2, pp. 343–354, 2005. DOI: `10.2514/1.3890`.

[54] D. García Yárnoz, J.-P. Sanchez Cuartielles, and C. R. McInnes, "Alternating orbiter strategy for asteroid exploration", *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 2, pp. 280–291, 2014. DOI: `10.2514/1.G000562`.

[55] R. A. Werner and D. J. Scheeres, "Exterior gravitation of a polyhedron derived and compared with harmonic and mascon gravitation representations of asteroid 4769 Castalia", *Celestial Mechanics and Dynamical Astronomy*, vol. 65, no. 3, pp. 313–344, 1996. DOI: `10.1007/BF00053511`.

[56] S. Pines, "Uniform representation of the gravitational potential and its derivatives", *AIAa Journal*, vol. 11, no. 11, pp. 1508–1511, 1973. DOI: `10.2514/3.50619`.

[57] J. B. Lundberg and B. E. Schutz, "Recursion formulas of legendre functions for use with nonsingular geopotential models", *Journal of Guidance, Control, and Dynamics*, vol. 11, no. 1, pp. 31–38, 1988. DOI: `10.2514/3.20266`.

[58] E. Fantino and S. Casotto, "Methods of harmonic synthesis for global geopotential models and their first-, second- and third-order gradients", *Journal of Geodesy*, vol. 83, no. 7, pp. 595–619, 2009. DOI: `10.1007/s00190-008-0275-0`.

[59] R. A. Werner, “Spherical harmonic coefficients for the potential of a constant-density polyhedron”, *Computers & Geosciences*, vol. 23, no. 10, pp. 1071–1077, 1997. DOI: 10.1016/S0098-3004(97)00110-6.

[60] B. Schutz, B. Tapley, and G. H. Born, *Statistical orbit determination.* Elsevier, 2004.

[61] *PDS Asteroid/Dust Archive.* [Online]. Available: https://sbn.psi.edu/pds/shape-models/ (visited on 03/05/2019).

[62] G. Shani, J. Pineau, and R. Kaplow, “A survey of point-based POMDP solvers”, *Autonomous Agents and Multi-Agent Systems*, vol. 27, no. 1, pp. 1–51, 2013. DOI: 10.1007/s10458-012-9200-2.

[63] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, “Planning and acting in partially observable stochastic domains”, *Artificial intelligence*, vol. 101, no. 1-2, pp. 99–134, 1998. DOI: 10.1016/S0004-3702(98)00023-X.

[64] L. Lin and T. Mitchell, “Memory Approaches to Reinforcement Learning in Non-Markovian Domains”, Pittsburgh, PA, USA, Tech. Rep., 1992.

[65] A. Géron, *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems.* O’Reilly Media, Inc., 2017.

[66] A.-a. Agha-mohammadi, S. Chakravorty, and N. M. Amato, “FIRM : Sampling-based Feedback Motion Planning Under Motion Uncertainty and Imperfect Measurements”, *The International Journal of Robotics Research*, vol. 33, no. 2, pp. 268–304, 2003. DOI: 10.1177/0278364913501564.

[67] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, “ORB-SLAM: a versatile and accurate monocular SLAM system”, *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015. DOI: 10.1109/TRO.2015.2463671.

[68] L. Mihaylova, T. Lefebvre, H. Bruyninckx, K. Gadeyne, and J. De Schutter, “A comparison of decision making criteria and optimization methods for active robotic sensing”, in *International Conference on Numerical Methods and Applications*, Springer, 2002, pp. 316–324. DOI: 10.1007/3-540-36487-0_35.

[69] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, “A Brief Survey of Deep Reinforcement Learning”, *IEEE Signal Processing Magazine*, vol. 34, 2017. DOI: 10.1109/MSP.2017.2743240.

[70] S. D. Whitehead and L.-J. Lin, “Reinforcement learning of non-Markov decision processes”, *Artificial Intelligence*, vol. 73, no. 1-2, pp. 271–306, 1995. DOI: 10.1016/0004-3702(94)00012-P.

[71] C. Liu, X. Xu, and D. Hu, “Multiobjective Reinforcement Learning: A Comprehensive Overview”, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 3, pp. 385–398, 2015. DOI: 10.1109/TSMC.2014.2358639.

[72] L. Barrett and S. Narayanan, “Learning All Optimal Policies with Multiple Criteria”, in *Proceedings of the 25th International Conference on Machine Learning*, 2008, pp. 41–47. DOI: 10.1145/1390156.1390162.

[73] C. J. Watkins and P. Dayan, "Q-learning", *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992. DOI: 10.1007/BF00992698.

[74] J. Schulman, S. Levine, P. Abbeel, M. I. Jordan, and P. Moritz, "Trust Region Policy Optimization.", in *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, vol. 37, 2015, pp. 1889–1897.

[75] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning", *arXiv preprint arXiv:1509.02971*, 2015.

[76] S. S. Haykin *et al.*, *Neural networks and learning machines.* New York: Prentice Hall, 2009.

[77] M. T. Hagan, H. B. Demuth, M. H. Beale, and O. De Jesús, *Neural network design.* Pws Pub. Boston, 1996, vol. 20.

[78] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop", in *Neural networks: Tricks of the trade*, Springer, 2012, pp. 9–48. DOI: 10.1007/978-3-642-35289-8_3.

[79] M. Riedmiller, "Neural fitted Q iteration–first experiences with a data efficient neural reinforcement learning method", pp. 317–328, 2005. DOI: doi.org/10.1007/11564096_32.

[80] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The RPROP algorithm", in *Neural Networks, 1993., IEEE International Conference on*, IEEE, 1993, pp. 586–591. DOI: 10.1109/ICNN.1993.298623.

[81] T. Gabel, C. Lutz, and M. Riedmiller, "Improved neural fitted Q iteration applied to a novel computer gaming and learning benchmark", in *2011 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, IEEE, 2011, pp. 279–286. DOI: 10.1109/ADPRL.2011.5967361.

[82] M. Riedmiller, "10 steps and some tricks to set up neural reinforcement controllers", in *Neural networks: tricks of the trade*, Springer, 2012, pp. 735–757. DOI: 10.1007/978-3-642-35289-8_39.

[83] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning", *Nature*, vol. 518, no. 7540, p. 529, 2015. DOI: 10.1038/nature14236.

[84] T. L. Saaty, "A scaling method for priorities in hierarchical structures", *Journal of Mathematical Psychology*, vol. 15, no. 3, pp. 234–281, 1977. DOI: 10.1016/0022-2496(77)90033-5.

[85] B. A. Archinal, M. F. A'Hearn, E. Bowell, A. Conrad, G. J. Consolmagno, R. Courtin, T. Fukushima, D. Hestroffer, J. L. Hilton, G. A. Krasinsky, G. Neumann, J. Oberst, P. K. Seidelmann, P. Stooke, D. J. Tholen, P. C. Thomas, and I. P. Williams, "Report of the IAU Working Group on Cartographic Coordinates and Rotational Elements: 2009", *Celestial Mechanics and Dynamical Astronomy*, vol. 109, no. 2, pp. 101–135, 2011. DOI: `10.1007/s10569-010-9320-4`.