Politecnico di Torino

EMBRY-RIDDLE Aeronautical University™
PRESCOTT, ARIZONA

Politecnico di Torino

Department of Mechanical and
Aerospace Engineering (DIMEAS)

Master's Degree in Aerospace Engineering

A Deep Reinforcement Learning Framework for Autonomous and Time-Critical Collision Avoidance in Low Earth Orbit

Supervisors

Prof. Davide Conte
Prof. Paolo Maggiore

Candidate

Alberto Preti

Academic Year 2024/2025

Abstract

The increase in the population of objects in orbit, together with the exponential growth of mega constellations, is significantly increasing the risk of collisions in Low Earth Orbit (LEO). This trend makes collision avoidance crucial to prevent the generation of new debris which, triggering a chain reaction, could lead to what the Kessler syndrome predicted, compromising future access to Earth orbit. Traditional approaches to collision avoidance maneuver planning based on ground operations are not scalable to the expected future traffic volumes and are ineffective in time-critical scenarios, where the time window available to plan and implement a maneuver can be reduced to just a few hours before the time of closest approach (TCA).

Due to these challenges, this thesis proposes a framework based on Deep Reinforcement Learning for the autonomous planning of time-critical collision avoidance maneuvers in LEO, characterized by decision windows as short as $1 - 2$ hours before TCA. The problem is formalized as a fully observable Markov decision process, in which an agent interacts with a simulated environment through the application of instantaneous Δv impulses. The action chosen by the agent is evaluated through a reward function designed to minimize the probability of collision, keep the satellite within its operational orbit, and optimize the consumption of the available propulsion budget. The simulated environment with which the agent interacts is a customized orbital propagator that allows, in addition to an unperturbed two-body model, the selection of perturbations relevant to the scenario under consideration, such as atmospheric drag, Earth's gravitational harmonics, solar radiation pressure, and third-body gravitational perturbations from the Sun and Moon.

To learn an optimal maneuvering strategy, the agent training is conducted using the Proximal Policy Optimization algorithm, based on an actor-critic architecture consisting of two multilayer perceptron neural networks. Moreover, to ensure generalization capability, a database of collision scenarios was generated to expose the agent to different collision geometries with space debris during the training phase. The evaluation of the learned optimal policy demonstrated the effectiveness of the proposed framework. The agent successfully planned avoidance maneuvers in scenarios never encountered during training with a total Δv on the order of 2 m/s ensuring that the satellite remained within its nominal operational orbit. In addition, a post-maneuvers analysis through the orbital propagation of the satellite, including most of the perturbations, verifies the absence of recurrent close approaches with the same debris.

The proposed framework is therefore fully compatible with time-critical scenarios, since the time required for maneuver planning corresponds to the inference time of the policy neural network, which is on the order of a few seconds.

Acknowledgements

A heartfelt thank you goes to all the people who, through their presence or support, have contributed to this journey.

To Professor Davide Conte, my mentor at Embry-Riddle Aeronautical University. The best professor, guide, and friend I could have ever wished for to accompany me through this thesis journey, which closes such an important chapter of my life. For welcoming me in Arizona and making me feel at home, and for his extraordinary availability and kindness.

To Professor Paolo Maggiore, already my supervisor during my Bachelor's thesis, for always showing, throughout these years at Politecnico, genuine attention and great openness in supporting me, making high-valuable educational experiences possible.

To my parents, Lucia and Mario, for your unconditional support throughout these years and for your opposite ways of educating me. To you, Mom, for always allowing me the freedom to make my own choices, right or wrong, without ever judging me, allowing me to live experiences that will stay in my heart for the rest of my life. To you, Dad, who with your severe words helped me not to lose direction, encouraging me to keep improving and reminding me that to reach 10, you must give 11.

To Martina, to my Tata. For giving me the chance to give my all in a journey that initially seemed impossible, completely revolutionizing my life. For all the times you chose to put me first, even when you needed me. For accepting my absences, my silence, and for believing in a version of myself that I had yet to see. For supporting, without questions, every choice I made, without ever judging me. For your unconditional support, for the certainty with which you believed in my potential, and for the enthusiasm with which you celebrated every small milestone I reached. For your love, which has given me, gives me, and I am sure will continue to give me the propellant to reach the stars.

To my siblings, Irene, Lidia, and Federico, in order of seniority, otherwise you'd get offended. For directly or indirectly motivating me throughout my entire university journey: making me understand that studying engineering was possible and that I was not alone, reminding me that with hard work everything is within reach, and above all for making me laugh almost every day by reminding me how many times I failed Calculus I.

To aunt Cinzia and uncle Giorgio, for showing me that there is always time to reach your goals, and for always supporting me.

To Marta, a lifelong friend, who in a dark moment reopened the door to old friendships that have now become the most important in my life. Lorenzo and Gianluca, the paratrooper and the pilot, lifelong friends united by an deep passion for aerospace, and the only ones who tolerate me when I talk about these topics for hours. Alice, who brings to our group of Piedmontese the warmth and joy that define her traditions. Manuela, for her generosity.

To Santo, for sharing unforgettable life experiences with me, and for understanding and accepting me during a moment of distance. For never doubting my abilities, to the point of being willing to bet even his last coin on me. I promise I will keep making it grow.

To Jack, companion in courses, exams, team, and even in this thesis. To our memorable adventures in Arizona, our endless car conversations, our evenings at Walmart, and the Sonic burger.

To the PoliTOrbital team, which has truly changed the course of my life, opening the door to academic and professional experiences I could have never imagined. In particular to Cristina, for seeing something in me from that very first interview, something worth supporting, and for still believing in it today.

Thank you.

Un sentito ringraziamento va a tutte le persone che hanno contribuito, con la loro presenza o il loro supporto, a questo percorso.

Al Professor Davide Conte, mio mentore all'Embry-Riddle Aeronautical University. Il miglior professore, guida e amico che avrei mai potuto desiderare per accompagnarmi in questo percorso di tesi, che chiude un capitolo così importante della mia vita. Per avermi accolto in Arizona facendomi sentire a casa, e per la sua costante disponibilità e straordinaria gentilezza.

Al Professor Paolo Maggiore, già mio relatore durante la tesi triennale, per aver sempre mostrato, in questi anni al Politecnico, un'attenzione autentica e una grande apertura nel sostenermi, rendendo possibili percorsi formativi di grande valore.

Ai miei genitori, Lucia e Mario, al vostro incondizionato appoggio e sostegno in questi anni e alla vostra opposta dualità educativa. A te mamma, per avermi sempre lasciato libero di prendere le mie scelte, giuste o sbagliate che fossero, senza mai giudicarmi, permettendomi di vivere esperienze che rimarranno nel mio cuore per il resto della mia vita. A te papà che con le tue parole severe mi hai aiutato a non perdere la direzione, spronandomi a migliorare sempre e ricordandomi che per avere 10 bisogna dare 11.

A Martina, alla mia Tata. Per avermi dato la possibilità di dare tutto me stesso in un percorso che inizialmente sembrava impossibile rivoluzionandomi completamente la vita. Per le volte in cui hai scelto di mettermi davanti a tutto, anche quando avresti avuto bisogno di me. Per aver accettato le mie assenze, i miei silenzi e per aver creduto in una versione di me che io ancora non vedevo. Per aver assecondato, senza fare domande, ogni mia singola scelta, senza mai giudicarmi. Per il tuo sostegno incondizionato, per la certezza con

cui hai creduto nelle mie potenzialità e per l'entusiasmo con cui hai sempre celebrato ogni mio piccolo traguardo. Per il tuo amore che mi ha dato, mi da e sono sicuro continuerò a darmi il propellente per raggiungere le stelle.

Ai miei fratelli, in ordine di anzianità altrimenti vi offendete, Irene, Lidia e Federico, che direttamente o indirettamente mi avete motivato durante tutto il mio percorso universitario. Facendomi capire che affrontare ingegneria fosse possibile e che non ero solo, ricordandomi che con il duro lavoro tutto è raggiungibile, e soprattutto facendomi ridere quasi ogni giorno ricordandomi il numero di volte che ho fallito l'esame di Analisi I.

A zia Cinzia e zio Giorgio, per avermi fatto capire che c'è sempre tempo per arrivare ai propri obiettivi sostenendomi sempre.

A Marta, amica da una vita, che in un momento buio mi ha riaperto la porta a vecchie amicizie, divenute oggi le più importanti della mia vita. Lorenzo e Gianluca, il paracadutista e il pilota, amici da sempre accomunati da una profonda passione per l'aerospazio, e gli unici a sopportarmi quando parlo per ore di questi argomenti. Alice, che porta nel nostro gruppo di piemontesi il calore e la felicità che contraddistinguono le sue tradizioni. Manuela, per la sua generosità.

A Santo, per aver condiviso insieme esperienze di vita indelebili e per avermi compreso ed accettato in un momento di lontananza. Per non aver mai dubitato delle mie capacità, al punto da essere disposto a scommettere anche la sua ultima monetina, continuerò a farla fruttare.

A Jacopo, compagno di corsi, esami, team e perfino di tesi. Alle nostre avventure memorabili in Arizona, agli infiniti discorsi in macchina, alle serate da Walmart e al panino di Sonic.

Al team PoliTOrbital, che ha letteralmente cambiato il corso della mia vita, aprendomi a esperienze accademiche e professionali che non avrei mai potuto immaginare. In particolare a Cristina, per aver visto in me, fin da quel primo colloquio, qualcosa che valeva la pena sostenere, e per continuare ancora oggi a crederci.

Grazie.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Space Debris

On October 4, 1957, with the launch of Sputnik 1 by the Soviet Union, the space age officially began. That milestone, which inaugurated the space race between the United States and the Soviet Union, also represented the starting point of a process that expressed the curiosity, ambition, and capacity for innovation of the human species. At the same time, that historic event gave rise to a chain of consequences that today threatens to permanently compromise the possibility of accessing and exploiting space in a sustainable manner. Since the early years of space exploration, human activities in orbit have produced a growing number of non-functional objects, exceeding the number of operational satellites. Fragments generated by collisions or explosions, components released during operations, satellites and rocket stages that are now no longer in use, all of these fall under the definition of space debris. More precisely, the international standard ISO 24113:2023 defines space debris as:

> *"Objects of human origin in Earth orbit or re-entering the atmosphere, including fragments and elements thereof, that no longer serve a useful purpose" [2].*

The problem is far from marginal. Since the 1970s, it has been recognized that the presence of these objects could evolve into a systemic threat to the sustainability of space activities [3]. Today, the risk associated with space debris extends far beyond traditional scientific and institutional missions, and increasingly affect the new space economy. This sector, characterized by the rapid deployment of large commercial constellations and satellite services, is critically dependent on the availability of a safe and predictable orbital environment, a necessary condition for ensuring its long-term sustainable development. The inherently global nature of the problem lies in the fact that every fragment in orbit, regardless of the country or operator that generated it, can pose a risk to any other spacecraft. This international dimension has motivated the creation of shared guidelines and standards. A decisive milestone was taken in 2002 with the publication of the Space Debris Mitigation Guidelines by the Inter-Agency Space Debris Coordination Committee (IADC), which provided the first common framework of reference for space agencies and

operators. These guidelines have served as the foundation for technical standards and national legislation, including the international standard ISO 24113 and the European standards ECSS-U-AS-10C, as well as the specific requirements of the European Space Agency (ESA). The common goal is to reduce the production of new debris and mitigate the risks associated with existing ones.

To ensure the safe and sustainable use of orbits, so-called protected regions have been introduced, within which specific debris mitigation requirements must be applied [4]. In particular, two main regions have been identified:

- *Low Earth Orbit (LEO) Protected Region*: a sphere extending from the Earth's surface up to an altitude of 2000 km.
- *Geosynchronous Earth Orbit (GEO) Protected Region*: a toroidal section centered on the geostationary altitude of 35786 km, bounded vertically by ±200km and latitudinally by ±15°.

Despite regulatory efforts and the progressive adoption of international guidelines, the orbital environment continues to evolve in a critical state. The number of artificial objects in orbit, together with their cumulative mass and exposed cross-section area, has grown steadily since the beginning of the space age, as highlighted in ESA's Annual Space Environment Report 2025 [5] and shown in Figure 1.1.

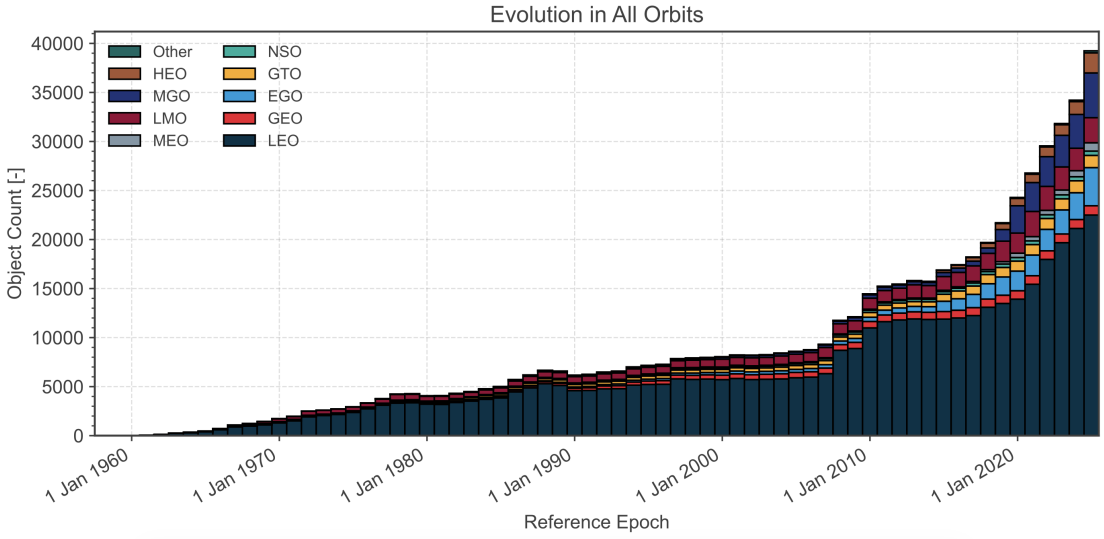


Figure 1.1: Evolution of number of object per orbit type

As illustrated in Figure 1.1 (sourced from [5]), LEO is the most congested orbital region, both in terms of satellite and debris population and in terms of its exposure to increasing space traffic associated with large telecommunications constellations. According to ESA's Meteoroid And Space debris Terrestrial Environment Reference (MASTER) model, as of August 2024 the estimated population includes approximately 54,000 objects larger than 10 cm (of which approximately 9,300 are active satellites), 1.2 million fragments between 1 and 10 cm, and over 130 million objects in the 1 mm-1 cm range [5].

The spatial density of objects in LEO, as illustrated in Figure 1.2 (sourced from [5]), indicates that the highest concentration, particularly for objects larger than 10 cm, occurs between 500 and 600 km altitude. In this region, the high orbital density increases the probability of fragmentation events, which are one of the main mechanisms for generating new debris. These events, referred to in literature and regulations as break-ups, are defined as the partial or total destruction of an object in orbit, resulting in the production of space debris [2]. The main causes include collisions with other objects, explosions due to residual energy that has not been adequately passivated (e.g., propellants or batteries), and structural deterioration. Statistics from the last two decades show that there have been an average of about 10–11 unintentional break-ups per year, with varying effects depending on the orbital lifetime of the fragments generated [6]. These phenomena highlight a self-sustaining process: as orbital density grows, the likelihood of collisions increases, and each collision can generate hundreds or thousands of new fragments. This is precisely the mechanism described by NASA astrophysicist Donald J. Kessler, who in 1978 first introduced the concept of what is now commonly referred to as the Kessler Syndrome [3]. His study demonstrated that, once a certain critical density in LEO was exceeded, fragments generated by random collisions between cataloged objects would become a primary source of new debris, with an exponential increase in the orbital population even in the absence of further launches.

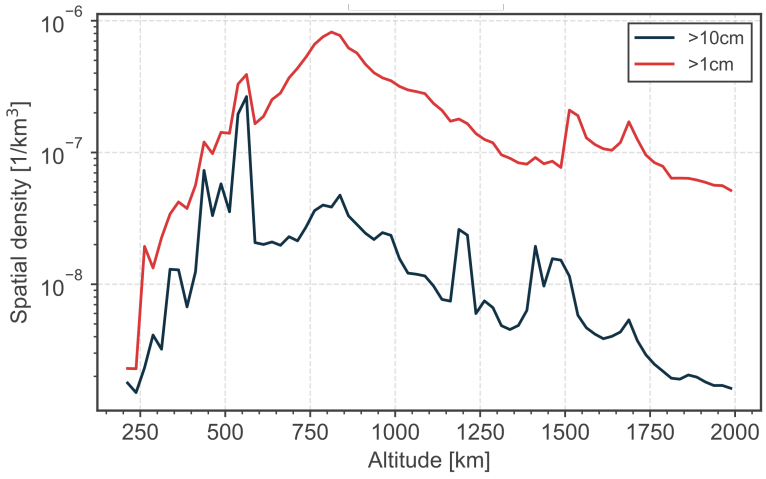


Figure 1.2: Density profiles in LEO for different space object size ranges from the 01/08/2024 MASTER reference population

More recent simulations conducted by ESA and IADC confirm the validity of this scenario: even in the extreme case of immediate termination of all future launches, collisions between objects already in orbit would lead to a further increase in the population in LEO [4]. This trend is clearly illustrated in Figure 1.3 (sourced from [6]), where the blue curve shows that, even under the extreme assumption of no further launches after 2024, the number of catalogued objects larger than 10 cm in LEO is expected to continue increasing due to collisions among existing debris.

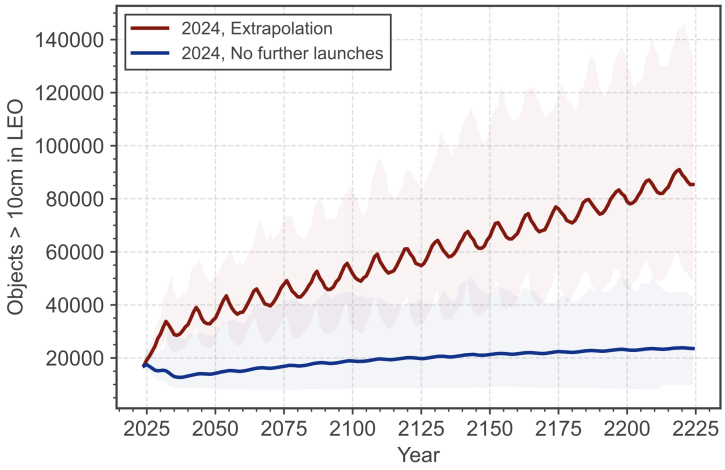


Figure 1.3: Number of objects larger than 10cm in LEO in the simulated scenarios of long-term evolution of the environment

1.1.1 Space Debris Mitigation Strategies

The scenario outlined above highlights how the current orbital environment is not sustainable in the long term. For this reason, the international community has gradually defined and adopted a set of mitigation strategies, which now constitute the benchmark for the design and operation of satellites and launch systems. These strategies can be grouped into four fundamental objectives: avoiding the release of space debris into Earth's orbit during normal operations, reducing the probability of break-up events, ensuring the removal of spacecraft at the end of their operational life (Post-Mission Disposal), and preventing on-orbit collisions.

Prevention of Space Debris Generation The first objective of mitigation strategies is to limit the intentional or accidental release of rocket and spacecraft components into space, for example during separation phases or routine maneuvers. Good design practices require a debris-aware approach, i.e., a design that considers the potential contribution of the system to the orbital environment from the earliest stages. This is also the context for more recent initiatives, such as ESA's Zero Debris Approach strategy, which aims to progressively reduce the impact of missions through stricter standards and innovative concepts such as design for demise (designing components to be completely destroyed upon re-entry into the atmosphere) and design for removal (design to facilitate post-mission removal using standardized docking systems or interfaces). In addition to preventing the generation of new debris, another highly relevant approach is active debris removal (ADR) techniques, which aim to eliminate objects already in orbit. These include conceptual solutions such as nets, harpoons, magnets, and robotic arms, often combined with advanced trajectory optimization algorithms that would allow multiple targets to be captured in the same mission. If implemented on a large scale, these techniques could make a significant contribution to reducing the population of debris in orbit. However, both design-for-removal solutions and active debris removal missions are still at an insufficient stage of technological maturity.

Although several demonstration projects and experimental missions are under development, no large-scale operational removal campaign has yet been completed, highlighting that these technologies, while promising, are not yet ready for systematic deployment. Moreover, the lack of direct economic return discourages investment in such missions.

Reducing the Probability of Break-up A second pillar of debris mitigation strategies is the prevention of unintentional fragmentation events, which constitute one of the main sources of new debris generation in orbit. In this context, a central role is played by passivation, defined in ISO 24113 [2] as: "*permanently depleting, irreversibly deactivating, or making safe all on-board sources of stored energy capable of causing an accidental break-up.*" The primary sources of stored energy that must be safely passivated include propellant tanks, batteries, high-pressure vessels, pyrotechnic devices, flywheels, and momentum wheels. These components should preferably be passivated as soon as they are no longer required for mission operations or disposal maneuvers. Passivation is considered an effective measure to substantially reduce the likelihood of accidental explosions that could generate new debris.

Post Mission Disposal At the end of their operational life, satellites and orbital stages must be removed from protected regions (LEO and GEO) by means of de-orbiting maneuvers or by exploiting natural decay due to atmospheric drag, or, in case of GEO, by transferring them to graveyard orbits. International regulations stipulate that satellites in LEO must not remain in orbit for more than 5 years, after the end of their mission. ESA was one of the first to implement this regulation and also requires that the cumulative probability of collision between end of life and re-entry remain below 10^{-3} [5].

On-Orbit Collision Prevention On-Orbit Collision prevention is the most immediate and scalable measure to contain the increase in space debris. The Collision Avoidance (CA) process is based on conjunction assessment, i.e., the systematic analysis of trajectory predictions based on orbital surveillance data and related uncertainties about the state of the satellite and the space debris. Through the process of conjuntion risk assessment, key encounter parameters, time of closest approach, minimum distance, and probability of collision are estimated. If the estimated probability exceeds the risk thresholds defined by international standards, the spacecraft operator is required to plan and execute a collision avoidance maneuver (CAM). Unlike other mitigation techniques, which require long development cycles or structural modifications, collision avoidance can be implemented quickly even on satellites already in operation, in many cases through simple software updates. Furthermore, CA remains indispensable even in the hypothetical scenario of "zero future launches": as ESA and IADC simulations show, collisions between objects already in orbit would still generate new debris (Figure 1.3).

1.2 Collision Avoidance

In light of the above considerations, collision avoidance is not only an effective measure for mitigating the growth in orbital debris, but also a fundamental component of space traffic management and the near-term sustainability of the orbital environment. It should not, however, be regarded as a single operational intervention, but rather as a structured process composed of multiple interconnected phases. Several definitions and categorizations have been proposed in literature, this thesis adopts a classification that integrates the terminology presented in the *NASA Conjunction Assessment and Collision Avoidance Handbook* [7] and by *Patnala et al.* [8], in order to systematically describe the set of activities necessary to identify, assess, and mitigate a conjunction event.

As shown in Figure 1.4, spacecraft collision avoidance can be logically decomposed into two major processes: the **Collision Detection Process**, which encompasses all activities related to space surveillance, orbital predictive modeling, and conjunction assessment; and the **Collision Avoidance Process**, which covers conjunction risk assessment and the subsequent mitigation phase. Together, these processes constitute the overall framework of a Collision Avoidance Operation.

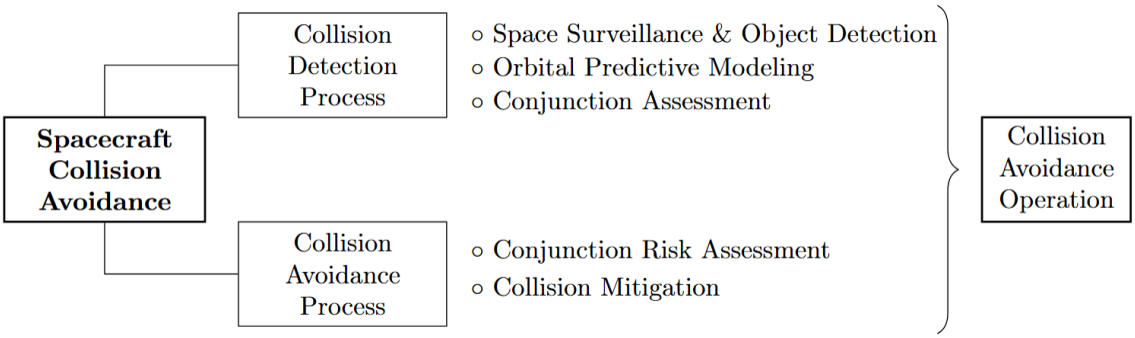


Figure 1.4: Collision Avoidance Framework

1.2.1 Collision Detection Process

Within the collision detection process three main components can be identified:

- *Space Surveillance and Object Tracking*
- *Orbital Predictive Modeling*
- *Conjunction Assessment*

Space surveillance and object tracking is a phase that represents the prerequisite for any collision avoidance operation. An accurate catalog of objects in orbit, together with appropriate data management and processing, is a necessary condition for satellite operators to be rapidly informed on a collision risk and to take the necessary corrective actions [8]. In this context, Space Situational Awareness (SSA) is defined as the comprehensive knowledge of the position of objects in near-Earth space, including their past, present, and predicted future status. The observations supporting SSA are mainly provided by Space Surveillance Networks (SSN), a global networks of radars and optical telescopes that continuously track

and catalog orbiting objects. Long-range radars, such as the U.S. Space Force's Space Fence or the Space Surveillance Telescope (SST), provide information on the trajectory, speed, and size of objects, and are particularly effective in characterizing the population of debris in LEO. At these altitudes, radar systems are able to accurately identify objects larger than or approximately equal to 10 cm [9]. While these fragments pose the most destructive threats in the event of impact, they have the advantage of being continuously tracked by existing space surveillance networks. The 10 cm threshold is explicitly specified in *ESA Space Debris Mitigation Requirement 5.3.3.5* [10], which demands that every object placed into Earth orbit to be trackable by a space surveillance segment to support collision avoidance processes. In particular, for the protected LEO region, traceability is guaranteed if at least one dimension of the object exceeds 10 cm. Object larger than this threshold can be reliably tracked, making collision avoidance maneuvers become both feasible and necessary. However, as demonstrated by the ESA MASTER model, the majority of objects in orbit are smaller than 1 cm. While these fragments represent the largest share of the population in terms of number, they typically cause only localized or limited damage. Consequently, they are primarily addressed through passive protection measures, such as shielding systems and resilient spacecraft design solutions [11].

The detection process also relies on **orbital predictive modeling**, which provides the foundation of any reliable conjunction assessment. Once raw tracking data has been collected, it must be processed and converted into meaningful orbital information, typically in the form of a state vector that defines the initial conditions for orbit propagation. In this context, the accuracy of the propagation models becomes a determining factor. Simplified analytical approaches, such as general perturbation models (e.g., SGP4) used in the widely adopted Two-Line Elements (TLE) format, are commonly employed to maintain large-scale catalogs of resident space objects. However, due to their limited accuracy, these methods are not suitable for detailed conjunction assessment and for supporting operational decision-making. For this reason, higher fidelity orbit propagation models are required to achieve the precision necessary to account for all relevant perturbative forces acting on the object of interest and to determine whether potential intersections with a protected satellite may occur.

Finally, the **conjunction assessment** (also referred to as screening [7]) is the process of systematically comparing the trajectory of the primary object with those of the cataloged population to identify potential close approaches. This is typically carried out by surveying a predefined screening volume around the object of interest: any secondary object whose projected trajectory enters this volume is classified as a potential conjunction threat. The accuracy of the assessment critically depends on the use of high-precision propagators, which are required to provide reliable predictions of relative motion. Within this framework, a conjunction is formally defined as the predicted passage of a secondary object within a protection volume surrounding the primary spacecraft at a given epoch [10]. The outcome of the process is the identification of situations where the protection volume of a satellite is at risk of being violated, thereby triggering the subsequent steps of the collision avoidance process.

1.2.2 Collision Avoidance Process

Once a potential conjunction event has been identified through conjunction assessment, a series of procedures are initiated to analyze the situation in detail and assess whether it is necessary to plan and execute a collision avoidance maneuver. The Collision Avoidance Process can be divided into two main phases:

- *Conjunction Risk Assessment*
- *Collision Mitigation*

The **conjunction risk assessment** is the core of collision avoidance process, as it provides operators with the means to quantify and interpret the associated risk and, consequently, to determine whether a corrective action is required. To this end, the so-called close approach analysis is performed to derive, from the available orbital data, the key geometric parameters of the encounter: the minimum separation distance, the time of closest approach, the relative velocity, and the conjunction angle. These quantities form the basis for estimating the collision probability (P_c), which serves as the primary metric for risk evaluation. P_c is widely regarded as one of the fundamental parameters for classifying the severity of a conjunction and for supporting decisions on the execution of an avoidance maneuver [12] and has become the internationally accepted standard for collision risk evaluation. In literature, several formulations for computing the probablity of collision have been proposed by *Foster and Estes* [13], *Chan* [14], *Alfano* [15], and *Patera* [16], reflecting different levels of approximation and underlying assumptions. The most relevant for the context of this thesis will be examined in detail in Section 4.1.2, together with its numerical implementations. From an operational standpoint, the decision to carry out a maneuver is based on comparing the estimated Pc against predefined thresholds. For spacecraft operating in LEO, the space industry and major agencies have largely converged on a threshold value of 10^{-4} per conjunction event, above which mitigation measures are recommended [7, 10, 12]. This threshold reflects a pragmatic compromise between ensuring spacecraft safety and minimizing the operational impact and resource expenditure associated with frequent maneuvers.

Once the risk has been quantified, the subsequent step is **collision mitigation**, referred to the planning and execution of strategies aimed at reducing the risks associated with on-orbit conjunctions between operational spacecraft and other resident space objects [10]. Several approaches are available to achieve this objective, including propulsive maneuvers (using chemical or electric propulsion), attitude changes to exploit aerodynamic drag variations or to present a minimal frontal area to the relative velocity vector to minimize the likelihood of collision [7]. In the present work, however, only propulsive collision avoidance maneuvers (CAMs) are considered, specifically impulsive maneuvers performed with chemical propulsion systems, where the risk reduction is obtained through the application of finite velocity increments (Δv). The feasibility of such maneuvers is intrinsically linked to the concept of spacecraft maneuverability, i.e., the capability of a satellite to alter its trajectory through controlled thrusting, which depends on its propulsion system, attitude control capabilities, and available propellant reserves. CAMs are generally

planned and executed in a time window preceding the time of closest approach (TCA), and their effectiveness is strongly influenced by the decision timing. Early maneuvers, conducted several orbital revolutions before the TCA, are typically less costly in terms of Δv but carry the risk of being unnecessary if subsequent tracking data reveals a reduced collision probability. Conversely, maneuvers executed closer to TCA benefit from improved state knowledge and therefore higher certainty in risk mitigation but demand larger Δv to both avoid the threatening object and restore the spacecraft to its nominal orbit [17]. This trade-off between timeliness and propellant expenditure is central to operational decision making in CA. Both NASA and ESA have defined quantitative requirements to assess the effectiveness of CAMs. In particular, NASA, based on the work performed by *Hall* [18] recommends that a maneuver reduce the P_c by at least 1.5 orders of magnitude, reflecting the point of diminishing returns in lifetime risk reduction, while ESA requires a more conservative reduction of two orders of magnitude, which also accounts for orbit determination uncertainties [7, 10].

1.2.3 Collision Avoidance Operation

In current operational practice, collision avoidance remains a highly structured process and still largely dependent on human intervention. The standard workflow begins with continuous monitoring of Conjunction Data Messages (CDMs), standardized notifications issued by the Combined Space Operations Center (CSpOC) or other space surveillance data providers. CDMs contain essential information about the predicted conjunction, including TCA, minimum distance, P_c, and covariance matrices of the objects involved, and are the primary operational reference for risk assessment [19]. These messages are typically issued approximately 72 hours before TCA and subsequently updated at regular intervals of 6–8 hours, allowing for progressively refined risk estimates as new observations are incorporated. When a conjunction is classified as critical, flight dynamics teams conduct maneuver analyses to evaluate possible options in terms of timing, thrust direction, and velocity magnitude, often relying on simplified strategies such as in-track thrusting to combine effectiveness and operational practicality [20]. These options are then coordinated with mission planners and operators to ensure compatibility with platform constraints, communication windows, and mission objectives. The planning phase includes preparing maneuver commands, reserving multiple ground-station passes for uplink, and validating that the proposed maneuver will not generate secondary high-risk encounters. To support this process, a number of specific tools and services have been developed over the years. Among the most relevant are the Conjunction Assessment Risk Analysis (CARA) [21], operated by NASA Goddard Space Flight Center to protect non-human missions, and the Collision Risk Assessment System (CRASS), developed by ESA in collaboration with GMV, which provides collision risk estimates and generates operational alerts [22]. Also within ESA, a central role is played by the Collision Risk Assessment and Avoidance Manoeuvres Computation (CORAM) tool, used by the ESA Space Debris Office for the analysis and optimization of CAMs, capable of managing scenarios with multiple encounters and operational constraints [22].

1.3 Motivation and Problem Statement

As discussed in the previous section, the planning and validation of CAMs still remains strongly dependent on mission operators and requires several hours, or even days, between the receipt of a CDM and the actual execution of the maneuver. This process, although effective, is not easily scalable to the growing number of satellites in orbit [8]. With the continuous expansion of space traffic, operators today must manage an increasing number of potential conjunctions, involving both other active satellites and space debris, often under extremely limited time constraints.

This situation has become critical with the rise of mega-constellations. At the start of 2024, the Starlink constellation, which had about 6000 active satellites at the time, was already doing almost 50000 collision avoidance maneuvers every six months, about 275 maneuvers per day [23]. According to the latest semi-annual report submitted by SpaceX to the Federal Communications Commission (FCC), this figure rose to approximately 144000 maneuvers between December 2024 and May 2025, an increase of approximately 200%, due to both the adoption of more conservative risk thresholds and the increase in the number of operational satellites, now exceeding 7000 [24]. Simulations conducted by *Lewis et al.* [25] confirm this trend: for a constellation of 36000 satellites, each spacecraft would need to perform an average of approximately 17 CAMs per year, equating to a total of approximately 360000 maneuvers per year, a figure that doubles if more restrictive risk thresholds are applied. Volumes of this order make continuous manual supervision impractical, highlighting the limitations of CAM planning processes based exclusively on human intervention.

1.3.1 The Short-Notice Gap in Collision Avoidance

Hobbs and Feron [26] proposed a temporal taxonomy that classifies collision avoidance systems into four categories, strategic, tactical, detect-and-avoid, and last-instant, based on the time remaining until the TCA. The first two categories correspond to current operational practice, respectively for intervals greater than 72 hours and between 24 and 72 hours from TCA. On the other hand, there are no operational systems or systems with a high technology readiness level for the detect-and-avoid (< 24 h) and last-instant (< 1 h) categories. This time segment, which is still not covered, represents a critical technological gap, where decisions must be made in extremely short times, but the planning systems currently based on the ground segment are too slow to provide an effective response.

It is therefore essential to develop autonomous onboard systems capable of planning and executing collision avoidance maneuvers in a very short time. Automation allows each satellite to independently calculate its own maneuver using a minimum set of data received from the ground, such as P_c, TCA, and the trajectory of the secondary object involved, without the need for immediate human validation. By transferring decision-making capacity on board, it is possible to drastically reduce reaction times in the event of critical encounters and, at the same time, lighten the operational load of mission control centers, achieving a scalable approach compatible with the future growth of large satellite

constellations [8,26]. Furthermore, this approach would also offer a significant advantage to small space companies, which often operate with limited resources and without a dedicated mission control center, allowing them to efficiently and safely manage collision avoidance operations and more easily access space through automation.

1.3.2 Limitation of Classical Optimization Approaches

Classical optimization methods have been widely used to address the problem of autonomous CAM planning. Convex optimization approaches, such as the one proposed by *Pavanello et al.* [27], implement a sequential convex programming methodology to generate low-propellant-consumption maneuvers in scenarios characterized by multiple close encounters. Similarly, *Dutta et al.* [28], further explored the convex optimization formulation by introducing the propagation of orbital uncertainty of debris and analyzing the evolution of associated uncertainties, with the goal of enabling the satellite to return to its nominal orbit. Analytical and semi-analytical methods have also been proposed, *Palermo et al.* [29], developed an analytical model for the design of low-thrust maneuvers, characterized by high reliability and the absence of iterative procedures. Analogously, *Gonzalo et al.* [30] presented a semi-analytical approach for low thrust for CAM based on proximal motion equations, expressed in terms averaged Keplerian elements, capable of providing compact and computationally efficient expressions for large-scale analysis or potential on-board implementations.

Alongside deterministic methods, global optimization and metaheuristic approaches have been developed. *Kim et al.* [31] proposed a genetic algorithm for generating optimal maneuvers in the presence of energy and probabilistic constraints, while *Zhang et al.* [32] introduced a global optimization algorithm called Timeline Club Optimization, developed within the traditional Ant Colony Optimization framework and applied to multiple debris removal missions. Finally, *Seong et al.* [33] compare different heuristic strategies in scenarios characterized by multiple threatening objects.

Overall, all these studies have contributed significantly to the advancement of avoidance maneuver planning methodologies, introducing increasingly sophisticated analytical and numerical tools. However, they share some inherent limitations: heavy dependence on the accuracy of the reference dynamic model, high computational costs associated with solving optimization problems, and poor adaptability to changing operating conditions, which requires the problem to be solved from the ground up for each new conjunction, without the possibility of reusing previous knowledge. In addition, the long computational time needed to compute an optimal solution prevents their use in time-critical conjunctions, where the decision windows before the TCA may be only a few hours. These limitations make traditional methods less suitable for operational contexts characterized by high orbital traffic density and extremely narrow decision windows, highlighting the need to develop more autonomous, adaptive, and computationally efficient approaches.

1.3.3 Reinforcement Learning for CAM Planning

Reinforcement Learning (RL) and its extension with deep learning techniques, Deep Reinforcement Learning (DRL), represent an alternative paradigm that directly addresses the limitations of traditional optimization methods in the context of autonomous CAM planning. Unlike the approaches mentioned earlier, RL allows an optimal control policy to be learned through repeated interaction with a simulated environment, without the need to explicitly know the system equations. This model-free approach is based on a trial-and-error process, in which the agent explores different sequences of actions and evaluates their consequences using a reward function, which in the field of CAM can simultaneously encompass different performance criteria, allowing the balancing of often conflicting objectives such as minimizing collision risk, propellant consumption, and orbital deviation. This mechanism allows effective decision-making strategies to be constructed even in the presence of nonlinear dynamics, time-varying disturbances, real-time decision-making, and multi-objective constraints, conditions that make explicit modeling in classical methods complex or prohibitive [34, 35].

The representation capabilities of neural networks in DRL formulations also allow the agent's behavior to be generalized to scenarios never encountered during training, exploiting the variety of experiences accumulated in interaction with the environment. Another crucial advantage of this approach, compared to traditional optimization techniques, lies in the separation between the training phase and the deployment phase. Once the optimal policy has been learned, its implementation is reduced to the execution of a function that associates the observed state with the corresponding action, with extremely low computational cost. This approach enables a workflow in which the agent is trained in simulation and, once the optimal policy is learned, it can be implemented on board with minimal computational effort and memory usage [35]. The clear separation between the computationally expensive offline training and the extremely fast on-board inference makes DRL methods particularly attractive for short-notice conjunction events, where time constrains preclude solving a new optimization problem from scratch.

Building on these considerations, several studies have already demonstrated the effectiveness of DRL in various space applications. A comprehensive analysis of the state of the art is presented by *Tipaldi et al.* [36], which offers a complete overview of the main studies that have successfully applied reinforcement learning techniques to the space sector. The applications cover a wide range of domains, from interplanetary trajectory design [37, 38], in which RL is used to plan Earth–Mars transfers in the presence of dynamic uncertainties, low-thrust constraints, and minimum propellant consumption criteria, to planning maneuvers in multi-body systems [39, 40], where reinforcement learning was used to design optimal trajectories in the context of the Circular Restricted Three-Body Problem (CR3BP) up to the field of guidance, rendezvous and docking maneuvers [35] and the orbital control of satellite constellations [41].

1.3.4 Problem statement

In light of the considerations presented, this thesis aims to develop and evaluate a DRL framework for the autonomous planning of CAMs in scenarios characterized by limited decision time windows, involving an active and maneuverable satellite and a single non-cooperative space debris. The objective is not to develop a new DRL algorithm or demonstrate its superiority over existing methods, but rather to evaluate whether such a framework can effectively address the operational limitations associated with time-critical CAMs planning, where traditional ground-based processes are not sufficiently rapid. The work focuses on the time window between one and two hours before the expected TCA. Within this context, the technical challenge lies in determining, from a minimal set of close-approach information, a sequence of impulsive maneuvers capable of reducing the collision probability below operational thresholds, respecting the available propulsive budget, and maintaining the orbital configuration within limits compatible with a rapid return to nominal operations. In this perspective, this thesis considers the operational scenario in which the DRL framework learns a policy during an offline training phase. Once the training is completed, the resulting policy is used to determine, in real time, the most appropriate sequence of maneuvers based solely on the observed state, thanks to very short inference times (on the order of seconds) and without requiring further training cycles. The work therefore aims to evaluate the extent to which the learned policy is able to generalize across different conjunction scenarios and to provide response times consistent with an autonomous, scalable and timely planning of CAMs in a constrained decision time windows.

Chapter 2

Orbital Dynamics and Perturbation Modeling

This chapter presents the fundamental concepts and mathematical formulation of the two-body problem, providing a solid theoretical basis for the orbital dynamics framework adopted in this thesis. The content aims not only to describe the ideal motion of a satellite subject to Earth's gravitational attraction alone, but also to clearly define the main conventions, equations and representations of the orbital state that will be used throughout the work.

The first part introduces the unperturbed two-body problem, starting with Newton's laws and ending with the analytical equation of the orbit. The chapter then focuses on the constants of motion and the two main ways of representing the orbital state: Cartesian and classical. A detailed description of the conversion between these two representations is also given. In the last section, the ideal model is extended through an overview of the main perturbative forces acting on real satellite orbits, such as atmospheric drag, solar radiation pressure, third-body effects and Earth's gravitational irregularities, and concludes with the implementation of the orbital propagator used throughout this work.

2.1 Unperturbed Two-Body Problem

The unperturbed two-body problem represents a fundamental idealization within orbital dynamics, widely adopted to describe the relative motion between celestial bodies, such as planets, natural satellites, and spacecraft. It describes the motion of two point masses, one primary (M) and one secondary (m), which interact exclusively through *Newton's law of gravity* [42]:

$$\mathbf{F}_g = -G\frac{Mm}{r^3}\mathbf{r} \tag{2.1}$$

where G =6.674 30 $\times 10^{-11}\,\mathrm{N\,m^2\,kg^{-2}}$ is the gravitational constant.
The geometric configuration of the system is defined by the position vector $\mathbf{r}$, representing the instantaneous distance between the two bodies. This vector originates at the center of mass of the primary body and points toward the position of the secondary body. In the absence of external perturbations, the motion occurs on a fixed plane in three-dimensional

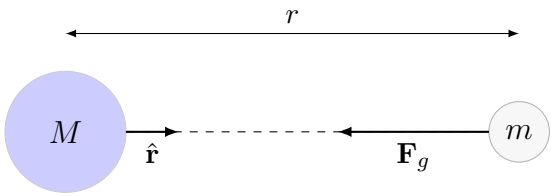


Figure 2.1: Newton's Law of Gravity

space, known as the *orbital plane*, whose existence will be justified in later sections. This configuration provides the geometric framework on which the subsequent dynamical formulation is based.

In order to analyze a more specific situation that aligns with the problem delineated in the thesis, M is considered to be the Earth's mass ($m_\oplus$), and m is designated as the mass of a satellite (m_s) that orbits the Earth.

2.1.1 Assumption and Model Description

To derive the equation of motion of the two-body problem, it is necessary to introduce some assumptions:

- The mass of the secondary body (m_s) is negligible compared to the mass of the primary (attractive) body ($m_\oplus$)
- The reference system used is inertial, or more precisely pseudo-inertial, since in reality each celestial body is subject to relative motion with respect to an absolute system.
- The bodies involved (primary and secondary) are assumed to be spherically symmetric, with uniform density. This assumption allows both to be modeled as point masses.
- No additional external forces are present: the only interaction is the gravitational one, acting along the conjunction between the centers of mass of the two bodies.

Two reference systems are considered:

- An ideal inertial reference frame (RF) XYZ, fixed in space or with constant orientation and origin in uniform rectilinear motion.
- An inertial RF IJK, centered in the center of the primary body, translated with respect to the previous one but not rotating with respect to it. In the context of this thesis, Earth-Centered-Inertial (ECI) RF is considered (a detailed description of which is provided in Appendix A).

2.1.2 Derivation of the Equation of Motion

In the ECI RF, Newton's law of gravitation for the Earth's gravitational force acting on the satellite can be expressed as follows:

$$\mathbf{F}_g = -G\frac{m_\oplus m_s}{r^2}\left(\frac{\mathbf{r}}{r}\right) \tag{2.2}$$

where $\mathbf{r}$ is the position vector of the satellite relative to the center of the Earth.

As shown in Figure 2.2, the position vector of the Earth and the satellite with respect to the inertial system XYZ are respectively $\mathbf{r}_\oplus$ and $\mathbf{r}_s$, thus $\mathbf{r}$ is:

$$\mathbf{r} = \mathbf{r}_s - \mathbf{r}_\oplus \tag{2.3}$$

Since the vectors are expressed with respect to an inertial reference system, it is possible to determine the relative acceleration of the satellite with respect to the Earth by deriving twice with respect to time:

$$\ddot{\mathbf{r}} = \ddot{\mathbf{r}}_s - \ddot{\mathbf{r}}_\oplus \tag{2.4}$$

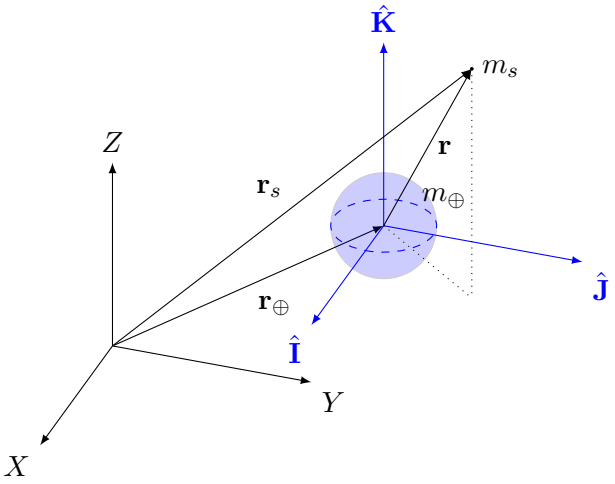


Figure 2.2: Two-body configuration in the IJK inertial frame.

Applying Newton's second law and the law of gravitation, the forces acting on the two bodies are:

$$\mathbf{F}_{g_s} = m_s \ddot{\mathbf{r}}_s = -G\frac{m_\oplus m_s}{r^2}\left(\frac{\mathbf{r}}{r}\right) \tag{2.5}$$

$$\mathbf{F}_{g_\oplus} = m_\oplus \ddot{\mathbf{r}}_\oplus = G\frac{m_\oplus m_s}{r^2}\left(\frac{\mathbf{r}}{r}\right) \tag{2.6}$$

After simplifying m_s in Eqn. (2.5) and $m_\oplus$ in Eqn. (2.6), subtracting Eqn. (2.6) from Eqn. (2.5) gives:

$$\ddot{\mathbf{r}} = -G\frac{m_\oplus + m_s}{r^2}\left(\frac{\mathbf{r}}{r}\right) \tag{2.7}$$

Given the initial assumptions of the two-body problem, the mass of the second body can be neglected with respect to the mass of the first body. It is therefore possible to assume $m_\oplus + m_s \approx m_\oplus$ and so $G(m_\oplus + m_s) \approx Gm_\oplus$. It is now convenient to introduce the Earth gravitational parameter $\mu \triangleq Gm_\oplus$, which, substituted into Eqn. (2.7) leads to the equation of motion of the unperturbed two-body problem:

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^2}\left(\frac{\mathbf{r}}{r}\right) \tag{2.8}$$

This second-degree, nonlinear, vector differential equation is often called *relative* two-body EOM because it represents the equation of motion of the secondary body with respect to the primary body.

2.1.3 Constants of motion

It is useful to introduce two fundamental constants of orbital motion into the two-body problem, as they provide significant information about the dynamical behavior of the secondary body and make it possible to predict its position and velocity over time. The conservation of *specific angular momentum* allows for the determination of the orbital plane and the geometry of the trajectory. Additionally, the *specific mechanical energy* is utilized to classify the type of orbit and to calculate certain characteristic quantities, such as the semi-major axis and the orbital period.

Specific Angular Momentum

In the context of the two-body problem, the angular momentum is commonly expressed in its *specific form* (per unit mass), as this eliminates the dependence on the satellite's mass, assumed negligible, and highlights the purely geometric and kinematic nature of the motion. The specific angular momentum can then be easily obtained by cross-multiplying Eqn. (2.8) with the position vector $\mathbf{r}$:

$$\mathbf{r} \times \ddot{\mathbf{r}} + \mathbf{r} \times \frac{\mu}{r^2}\left(\frac{\mathbf{r}}{r}\right) = 0 \tag{2.9}$$

Since $\mathbf{r} \times \mathbf{r} = 0$, the second term of Eqn. (2.9) disappears while the first term can be written through the following differential:

$$\frac{d}{dt}(\mathbf{r} \times \dot{\mathbf{r}}) = \cancel{\dot{\mathbf{r}} \times \dot{\mathbf{r}}} + \mathbf{r} \times \ddot{\mathbf{r}} = \mathbf{r} \times \ddot{\mathbf{r}} \tag{2.10}$$

Substituting this differential into Eqn. (2.9), since the derivative results in zero, it follows that the integrated quantity must remain constant over time. By rearranging the expression and replacing the derivative of the position vector ($\mathbf{r}$) with velocity ($\mathbf{v}$), it is possible to obtain:

$$\mathbf{h} = \mathbf{r} \times \mathbf{v} = \text{constant} \tag{2.11}$$

This definition shows that the specific angular momentum ($\mathbf{h}$) is a vector perpendicular to both the position vector and the velocity vector, and therefore orthogonal to the plane determined by them, called the *orbital plane*. This implies that, in the unperturbed two-body problem, the motion of the satellite is confined to a fixed plane whose orientation in space remains constant in time.

It is also useful to consider the magnitude of specific angular momentum, as it introduces an angle of particular relevance to orbital dynamics: *the flight path angle* (γ), defined as the angle between the satellite's velocity vector and the local horizontal (i.e., the direction perpendicular to the radial vector).

The magnitude of the specific angular momentum can be expressed as:

$$h = rv\cos\gamma \tag{2.12}$$

At periapsis and apoapsis, the flight path angle is zero because the velocity vector is tangent to the orbit and parallel to the local horizontal. At these points, Eqn. (2.12) simplifies to:

$$h = rv = r_a v_a = r_p v_p \tag{2.13}$$

Specific Mechanical Energy

The second constant of motion, the specific mechanical energy, can be obtained by taking the dot product of both sides of Eqn. (2.8) with the velocity vector $\dot{\mathbf{r}}$

$$\dot{\mathbf{r}} \cdot \ddot{\mathbf{r}} = -\dot{\mathbf{r}} \cdot \frac{\mu}{r^2}\left(\frac{\mathbf{r}}{r}\right) \tag{2.14}$$

In the orbital motion, the acceleration of the satellite is entirely radial, that is, directed along the position vector $\mathbf{r}$. This implies that $\mathbf{r}$ and $\ddot{\mathbf{r}}$ are parallel and that the angle between $\mathbf{r}$ and $\dot{\mathbf{r}}$ remains constant over time. Under these conditions, it is possible to rewrite Eqn. (2.14) in a scalar form:

$$v\dot{v} + \frac{\mu}{r^2}\dot{r} = 0 \tag{2.15}$$

Two derivatives with respect to time can be recognized in this expression:

$$\frac{d}{dt}\left(\frac{v^2}{2}\right) = v\dot{v} \qquad \frac{d}{dt}\left(-\frac{\mu}{r}\right) = \frac{\mu}{r^2}\dot{r} \tag{2.16}$$

The former represents the derivative of the *specific kinetic energy*, while the latter represents the derivative of the *specific potential energy*. Substituting in Eqn. (2.15), it yields:

$$\frac{d}{dt}\left(\frac{v^2}{2} - \frac{\mu}{r}\right) = 0$$

The sum of kinetic energy and specific potential constitutes the *specific mechanical energy*, and since its time derivative is zero, it is concluded that this quantity remains constant along the orbit:

$$\mathscr{E} = \frac{v^2}{2} - \frac{\mu}{r} + c \tag{2.17}$$

The integration constant, often denoted c, depends on the choice of reference level for the potential energy and can take on an arbitrary value. In astrodynamics, it is conventional to set $c = 0$, which is equivalent to considering the gravitational potential energy to be zero at infinity. With this convention, the specific potential energy is always negative along the orbit. In the case of a closed orbit (e.g., elliptical), the satellite's total energy is conserved, but it is distributed differently along the trajectory: at perigee (the point closest to the attractive center), the satellite has high kinetic energy and lower potential,

at apogee (the farthest point), the opposite happens, with lower kinetic energy and higher potential. In some cases, it can be useful to define $\mathscr{E}$ through the semi-major axis (a) of the orbit, particularly when the position and velocity vector are not known. Assuming the position of the satellite is at the perigee of the orbit, the substitution of the specific angular momentum magnitude, Eqn. (2.13), into Eqn. (2.15) results in:

$$\mathscr{E} = \frac{v^2}{2} - \frac{\mu}{r} = \frac{h^2}{2r_p^2} - \frac{\mu}{r_p} \tag{2.18}$$

Using the equalities $r_p = a(1-e)$, and $h = \sqrt{\mu a(1-e^2)}$:

$$\mathscr{E} = \frac{\mu a(1-e^2)}{2a^2(1-e)^2} - \frac{\mu}{a(1-e)} \tag{2.19}$$

After some simplification, this expression (valid for all non parabolic orbits) leads to the following result:

$$\mathscr{E} = -\frac{\mu}{2a} \tag{2.20}$$

Eqn. (2.17) and (2.20) are sometimes combined giving the traditional form of the *vis-viva equation*:

$$\mathscr{E} = \frac{v^2}{2} - \frac{\mu}{r} = -\frac{\mu}{2a} \tag{2.21}$$

2.1.4 Analytical Orbit Equation

Starting from the equation of motion of the unperturbed two-body problem, Eqn. (2.8), it is possible to derive an analytical solution in closed form that describes the trajectory of the secondary body. In particular, by integrating the equation in the orbital plane and taking advantage of conservation of specific angular momentum, the orbit equation in polar form is obtained:

$$r = \frac{h^2/\mu}{1 + (B/\mu)\cos\theta} = \frac{p}{1 + e\cos\theta} \tag{2.22}$$

where the quantities in the equation, depicted in Figure 2.8, are defined as follows:

- B: *constant of integration*, related to the eccentricity of the orbit;
- θ: *true anomaly*, the angle between the perigee vector and the instantaneous position of the satellite along the orbit;
- h^2/μ: corresponds to the *semilatus rectum* (p), the distance from the focus (where the attractive body is located) to the orbit, measured perpendicular to the major axis;
- B/μ: represents the *eccentricity* (e), parameter that determines the geometric shape of the orbit ("roundness").

The existence of this analytical solution confirms and generalizes Kepler's first law, showing that every orbit subject to a central gravitational force is a conic section.

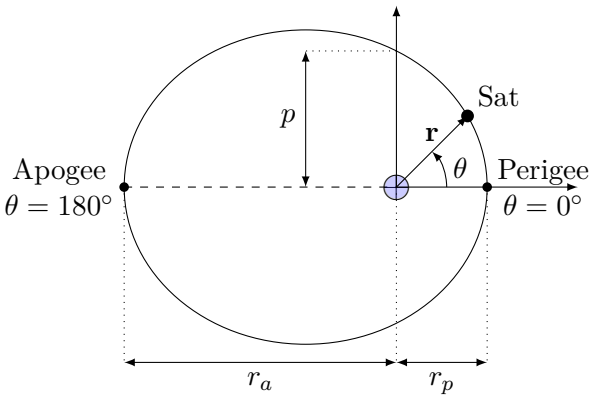


Figure 2.3: Geometric Parameters Polar Form Orbit

The value of eccentricity has a central role in the classification of orbits. Together with the specific mechanical energy, it makes it possible to identify the type of conic section described by the orbital trajectory and to determine whether the orbit is closed or open. The combination of these two parameters makes it possible to distinguish circular, elliptical, parabolic and hyperbolic orbits, as reported in Table 2.1.

Orbit Type	Eccentricity e	Specific Energy ε	Boundedness
Circular	$e = 0$	$\varepsilon < 0$	Closed
Elliptical	$0 < e < 1$	$\varepsilon < 0$	Closed
Parabolic	$e = 1$	$\varepsilon = 0$	Open (limiting case)
Hyperbolic	$e > 1$	$\varepsilon > 0$	Open

Table 2.1: Classification of conic orbits

2.2 State Representation

In order to completely describe the motion of a satellite in space, six independent quantities are required, which define the so-called *state* of the satellite. These quantities can be expressed in equivalent forms, the two main ones being:

- **Cartesian state vector**, consisting of three components of the position vector ($\mathbf{r}$) and three components of the velocity vector ($\mathbf{v}$), expressed in an inertial RF. This representation is suitable for numerical propagation, since it directly provides the derivatives needed for integration of the equations of motion, but it has limited physical interpretability in terms of the size, shape and orientation of the orbit.
- **Classical Orbital Elements**, six scalar parameters derived from the geometry of conic sections that clearly describe the shape, size, orientation, and temporal component of the orbit. They are commonly used for qualitative analysis and understanding of orbital motion.

Both representations provide a complete set of initial conditions for solving the two-body problem and must be referenced to a well-defined coordinate system, typically the ECI RF.

2.2.1 Cartesian Orbital Elements

Although the ECI coordinates of position and velocity do not provide an immediate understanding of the shape, size and orientation of the orbit in space, they provide a very effective basis for numerical integration of the equations of motion, as mentioned earlier. The position of the satellite in the ECI RF can be expressed as:

$$\mathbf{r} = x\hat{\mathbf{I}} + y\hat{\mathbf{J}} + z\hat{\mathbf{K}} \tag{2.23}$$

where x, y, and z represent the coordinates of the position vector. The magnitude of the position vector thus results in:

$$r = \sqrt{\mathbf{r} \cdot \mathbf{r}} = \sqrt{x^2 + y^2 + z^2} \tag{2.24}$$

Differentiating $\mathbf{r}$ with respect to time, the velocity vector is obtained:

$$\mathbf{v} = \frac{dx}{dt}\hat{\mathbf{I}} + \frac{dy}{dt}\hat{\mathbf{J}} + \frac{dz}{dt}\hat{\mathbf{K}} \tag{2.25}$$

The magnitude of the velocity vector is then:

$$v = \sqrt{\mathbf{v} \cdot \mathbf{v}} = \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2 + \left(\frac{dz}{dt}\right)^2} \tag{2.26}$$

The three components of Eqn. (2.23) and Eqn. (2.25) constitute the six initial conditions necessary for numerical integration of the system of coupled second-order differential equations. Taking the derivative of Eqn. (2.23) twice, and equating the resulting acceleration to Newton's Law of Universal Gravitation gives the two-body equation of motion for this system:

$$\frac{d^2\mathbf{r}}{dt^2} = -\frac{\mu}{r^3}\mathbf{r} \tag{2.27}$$

The numerical integration of this equations (which is discussed in more detail in Section 2.4) makes it possible to determine the trajectory for any type of orbit. However, to obtain a more meaningful representation, these Cartesian elements should be converted into the corresponding classical orbital elements, as discussed in the next section.

2.2.2 Classical Orbital Elements

While Figure 2.4 provides a geometric visualization of the majority of the classical orbital elements, the formal definitions and physical meaning of each of them are as follow:

- **Semi-major axis** (a) is a geometric quantity corresponding to half the major axis of the conic section describing the orbital trajectory. In practical terms, it represents the average distance between the orbiting body and the focus of the orbit, coincident with the center of mass of the central body. For elliptical orbits, a provides a direct measure of the size of the orbit, while in the special case of circular orbits it coincides with the radius.

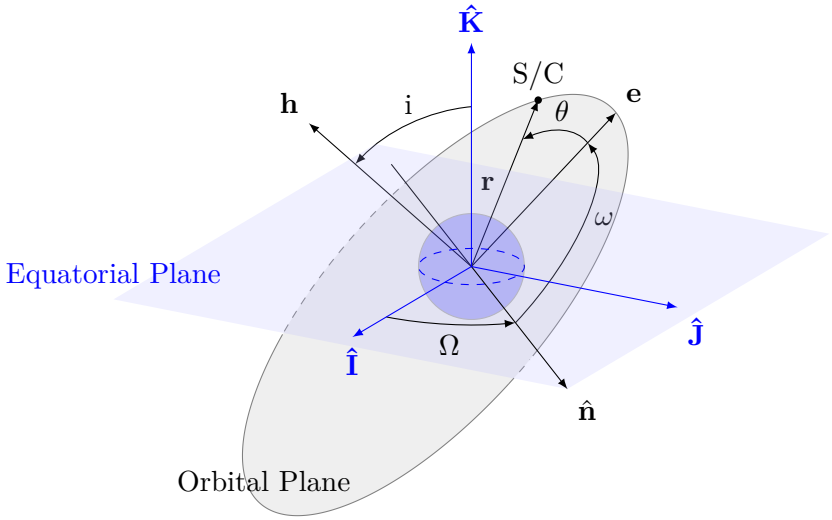


Figure 2.4: Classical Orbital Elements

- **Eccentricity** (e) is a dimensionless quantity that characterizes the shape of the orbital trajectory. e is the magnitude of the eccentricity vector $\mathbf{e}$, a vector quantity that lies in the orbital plane and points toward periapsis. Physically, eccentricity quantifies the degree to which the orbit deviates from circularity (e=0). As described in Table 2.1, eccentricity is therefore an essential parameter to classify the conic types.

- **Inclination** (i) is the angle between $\hat{\mathbf{K}}$ and the specific angular momentum vector ($\mathbf{h}$), that describes the orientation of the orbital plane with respect to the Earth's equatorial plane. i takes values between 0° and 180°. In particular, an inclination of 0° or 180° identifies an *equatorial* orbit, between 0° and 90° degrees a *prograde* orbit, and between 90° and 180° a *retrograde* orbit.

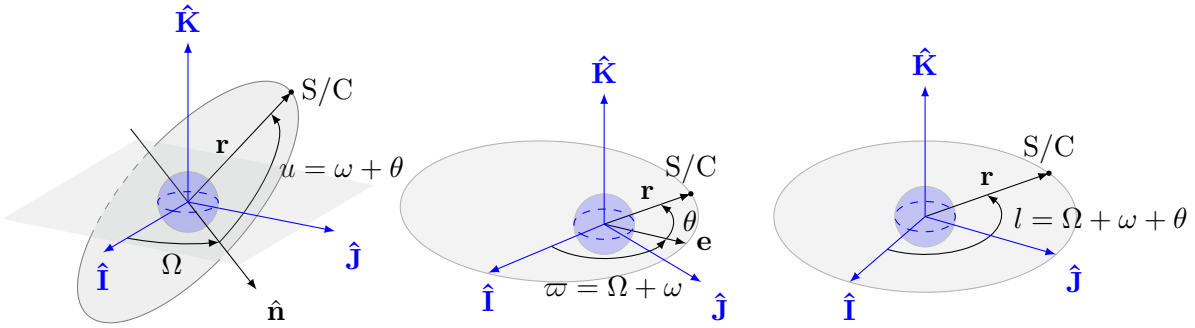
- **Right Ascension of Ascending Node** (RAAN, Ω) is the angle, measured in the equatorial plane counterclockwise, between $\hat{\mathbf{I}}$ and the direction of the *ascending node* ($\hat{\mathbf{n}}$). The latter is the point at which the satellite crosses the Earth's equator going from the Southern to the Northern hemisphere. The *descending node*, on the other hand, is the diametrically opposite point where the satellite crosses the equator from North to South. The union of the two nodes defines the node line, which lies on the equatorial plane and locates the line of intersection between the orbital and equatorial planes. The vector associated with this direction is usually denoted by $\mathbf{n}$ and is called the *node vector*. The RAAN is well defined only for inclined orbits ($i \neq 0°, 180°$) and can take values between 0° and 360°.

- **Argument of Perigee** (ω) is the angle measured in the orbital plane, counterclockwise to the direction of the satellite's motion, between the direction of the ascending node and the direction of perigee. ω specifies the position of the closest point to the central body along the orbit, relative to the line of nodes. The value of ω is between 0° and 360°.

- **True Anomaly** (θ) is the angle that locates the instantaneous position of the satellite along the orbit. θ is measured in the orbital plane from the direction of perigee to the current position of the satellite, following the direction of orbital motion. The value of θ ranges from 0° and 360° and, unlike the other orbital elements, is not constant in time, but changes continuously during motion. True anomaly thus provides the dynamic information that completes the description of the orbital state at a given instant.

Singularities

While the use of classical orbital elements provides a meaningful physical understanding of orbit geometry in space, there are two cases in which they exhibit singularities.



(a) Argument of Latitude (u) (b) Longitude of periapsis (ϖ) (c) True Longitude (l)

Figure 2.5: Singularities in Classical Orbital Elements

The first singularity occurs for an exactly circular orbit ($e = 0$). In a circular orbit, since there is no periapsis, the angle ω (argument of the perigee) is not defined. Furthermore, in the absence of a periapsis, the true anomaly θ, which represents the time evolution of the satellite position measured from the periapsis, is also not uniquely defined. In this case, an alternative element called *argument of latitude* (u) is introduced (see Figure 2.5a), defined as the angle between the direction of the ascending node $\hat{\mathbf{n}}$ and the instantaneous position of the satellite along the orbit.

The second singularity occurs in the case of an exactly equatorial orbit ($i = 0°, 180°$). Since in such a case the orbital plane coincides with the equatorial plane, a crossing of the orbital plane in the ascending node never occurs, and therefore Ω is not defined. To remedy this deficiency, a new parameter is defined, called *longitude of periapsis* (ϖ) (see Figure 2.5b), which represents the angle between $\hat{\mathbf{I}}$ and the eccentricity vector ($\mathbf{e}$).

The situation is further complicated when the orbit is simultaneously circular and equatorial ($e = 0$ and $i = 0°, 180°$), as in the case of ideal geosynchronous orbits. In such a scenario, neither ω nor Ω are defined, and another parameter is defined, the *true longitude* (l) (see Figure 2.5c), defined as the angle between $\hat{\mathbf{I}}$ and the instantaneous position of the satellite along the orbit. Although the latter case mainly represents a mathematical occurrence (getting a perfectly circular and equatorial orbit is actually operationally unfeasible) near-zero values of eccentricity and inclination can still cause numerical problems in integrating the equations of motion. For this reason, in numerical

integration it is often preferred to use the Cartesian representation of orbital states, which has no such structural singularities.

2.2.3 Conversion between Representation

Many initial conditions provided by operational databases, mission analysis tools or analytical formulations are expressed through classical orbital elements, while numerical integration of equations of motion typically requires the use of Cartesian coordinates. Similarly, at the end of the propagation, it may be useful to reconvert the dynamical state to elemental form to analyze the evolution of physically significant parameters.

In this context, the ability to accurately perform the conversion between representations is a key step to ensure both proper initialization of simulations and clear interpretation of the results.

From Classical Orbital Elements to Cartesian Coordinates

Given the set of classical orbital elements $\{a, e, i, \Omega, \omega, \theta\}$ it is possible to compute the position and velocity vectors $\mathbf{r}$ and $\mathbf{v}$ in the ECI RF starting from the definition of semilatus rectum (p) and the orbit equation in polar form, respectively:

$$p = a(1 - e^2) \tag{2.28}$$

$$r = \frac{p}{1 + e\cos\theta} \tag{2.29}$$

Once the perifocal reference system (PQW) (a detailed description of which is given in Appendix A), has been defined, it is possible to express the position and velocity vectors of the satellite, $\mathbf{r}^P$ and $\mathbf{v}^P$, respectively, within that system according to the following formulations:

$$\mathbf{r}^P = \begin{bmatrix} r\cos\theta \\ r\sin\theta \\ 0 \end{bmatrix} \tag{2.30}$$

$$\mathbf{v}^P = \begin{bmatrix} -\sqrt{\frac{\mu}{p}}\sin\theta \\ \sqrt{\frac{\mu}{p}}(e + \cos\theta) \\ 0 \end{bmatrix} \tag{2.31}$$

Referring now to Figure 2.6, a transformation is made between the ECI RF and the perifocal RF through the following direction cosine matrix (DCM):

$$\mathbf{C}^{EP} = \mathbf{R}_z(\Omega)\mathbf{R}_x(i)\mathbf{R}_z(\omega) \tag{2.32}$$

where the rotation matrices $\mathbf{R}_z(\Omega)$, $\mathbf{R}_x(i)$, and $\mathbf{R}_z(\omega)$, are respectively:

$$\mathbf{R}_z(\Omega) = \begin{bmatrix} \cos\Omega & \sin\Omega & 0 \\ -\sin\Omega & \cos\Omega & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.33}$$

$$\mathbf{R}_x(i) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos i & \sin i \\ 0 & -\sin i & \cos i \end{bmatrix} \tag{2.34}$$

$$\mathbf{R}_z(\omega) = \begin{bmatrix} \cos\omega & \sin\omega & 0 \\ -\sin\omega & \cos\omega & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.35}$$

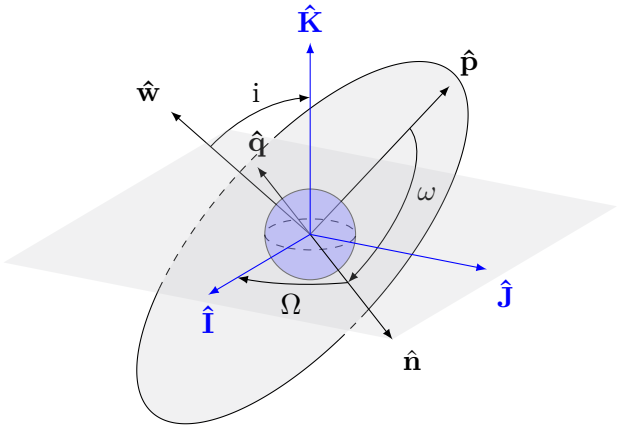


Figure 2.6: Perifocal to ECI RF Transformation

To perform the transformation from the PQW frame to the ECI frame a rotation of Ω is performed around $\hat{\mathbf{K}}$, followed by a rotation of i about the new $\hat{\mathbf{I}}'$, and finally a rotation of ω about $\hat{\mathbf{K}}$. The position and velocity vectors in the ECI RF ($\mathbf{r}$ and $\mathbf{v}$) can therefore be written as:

$$\begin{aligned} \mathbf{r} &= \mathbf{C}^{EP}\mathbf{r}^P \\ \mathbf{v} &= \mathbf{C}^{EP}\mathbf{v}^P \end{aligned} \tag{2.36}$$

From Cartesian Coordinates to Classical Orbital Elements

Given the state vectors in Cartesian coordinates ($\mathbf{r}$ and $\mathbf{v}$), along with their magnitudes, it is possible to determine the set of classical orbital elements $\{a, e, i, \Omega, \omega, \theta\}$ by leveraging the constants of motion and applying purely geometric considerations related to the orientation of the orbit in space.

The first orbital element, the *semimajor axis (a)*, is obtained from the vis-viva equation, Eqn. (2.20):

$$a = \frac{1}{\left(\frac{2}{r} - \frac{v^2}{\mu}\right)} \tag{2.37}$$

To determine the *eccentricity* (e), the eccentricity vector $\mathbf{e}$ is first computed from its definition:

$$\mathbf{e} = \frac{\mathbf{r} \times \mathbf{h}}{\mu} - \frac{\mathbf{r}}{r} \tag{2.38}$$

The eccentricity is then obtained as the magnitude of the eccentricity vector:

$$e = |\mathbf{e}| \tag{2.39}$$

Since the *inclination* (i) is defined as the angle between the unit vector $\hat{\mathbf{K}}$ and the specific angular momentum vector $\mathbf{h}$ (with magnitude $h = |\mathbf{h}|$), it can be computed using the definition of the dot product:

$$\cos i = \frac{\hat{\mathbf{K}} \cdot \mathbf{h}}{h} \tag{2.40}$$

As the inclination is conventionally defined within the range $[0° - 180°]$, no quadrant check is required when evaluating the inverse cosine.

The *RAAN* (Ω) is determined using the *node vector* $\mathbf{n}$ which is defined as the cross product between $\hat{\mathbf{K}}$ and the specific angular momentum vector $\mathbf{h}$:

$$\hat{\mathbf{n}} = \hat{\mathbf{K}} \times \mathbf{h} \tag{2.41}$$

Since Ω is defined as the angle between $\hat{\mathbf{I}}$ and the nodal vector $\mathbf{n}$, it can be computed using the definition of the dot product:

$$\cos \Omega = \frac{\hat{\mathbf{I}} \cdot \hat{\mathbf{n}}}{n} \tag{2.42}$$

However, unlike the inclination, the RAAN can vary from 0° to 360°, so it is necessary to determine the appropriate quadrant for the resulting angle. This can be done by inspecting the sign of the y-component of the node vector $\mathbf{n}$:

$$\Omega = \begin{cases} \cos^{-1}\left(\frac{\hat{\mathbf{I}} \cdot \hat{\mathbf{n}}}{n}\right) & n_y \geq 0 \\ 2\pi - \cos^{-1}\left(\frac{\hat{\mathbf{I}} \cdot \hat{\mathbf{n}}}{n}\right) & n_y < 0 \end{cases} \tag{2.43}$$

The *argument of periapsis* (ω) is defined as the angle between the node vector $\mathbf{n}$ and the eccentricity vector $\mathbf{e}$, both of which lie in the orbital plane. It can be computed using the scalar (dot) product:

$$\cos \omega = \frac{\mathbf{e} \cdot \hat{\mathbf{n}}}{en} \tag{2.44}$$

Like the RAAN, the argument of periapsis can vary from 0° to 360°. To determine the appropriate quadrant, the sign of the z-component of the eccentricity vector must be inspected.

$$\omega = \begin{cases} \cos^{-1}\left(\frac{\mathbf{e} \cdot \hat{\mathbf{n}}}{en}\right)) & e_z \geq 0 \\ 2\pi - \cos^{-1}\left(\frac{\mathbf{e} \cdot \hat{\mathbf{n}}}{en}\right)) & e_z < 0 \end{cases} \tag{2.45}$$

Finally, *true anomaly* (θ), defined as the angle between the eccentricity vector (position of the perigee) and the instantaneous position of the satellite is computed like the last two elements using the definition of the dot product:

$$\cos \theta = \frac{\mathbf{e} \cdot \mathbf{r}}{er} \tag{2.46}$$

Since the true anomaly is defined within the range 0° to 360°, the appropriate quadrant can be determined by inspecting the sign of the radial component of the velocity ($v_r = \mathbf{v} \cdot \hat{\mathbf{r}}$):

$$\theta = \begin{cases} \cos^{-1}\left(\frac{\mathbf{e}\cdot\mathbf{r}}{er}\right) & v_r \geq 0 \\ 2\pi - \cos^{-1}\left(\frac{\mathbf{e}\cdot\mathbf{r}}{er}\right) & v_r < 0 \end{cases} \tag{2.47}$$

2.3 Perturbations Modeling

Perturbations represent, in general, all those deviations from the ideal, simplified, undisturbed motion typical of the two-body problem. Physical reality does not conform exactly to the simplifying assumptions underlying Keplerian motion, so the ideal solutions constitute only an approximation of the observed real motion. Under operational conditions, in fact, the motion of a satellite is affected by various additional forces and external influences that alter its trajectory, generating even significant deviations from the theoretical two-body orbit. Although in many cases perturbative forces are relatively small in magnitude, their cumulative effect over time can significantly alter the orbital motion of a satellite. In some scenarios, their long-term influence can become comparable to or even exceed that of the primary gravitational force. A rigorous dynamical model would require the inclusion of a wide range of perturbations, including those of modest instantaneous strength, because of their potentially large integrated impact. Depending on the specific scenario analyzed in this thesis, different perturbative effects are considered. For short-term scenario, in which the orbital propagation spans over a few hours, the dominant perturbations in LEO are the *atmospheric drag* and the *irregularities of the Earth's gravitational field*, due to the Earth's non perfect sphericity. For longer propagation periods, spanning several days, additional perturbations such as the *third-body gravitational effects* (particularly the Moon and Sun), and the *solar radiation pressure* are included, although their magnitude in LEO remains considerably smaller compared to the previously mentioned ones.

Before describing each perturbation in detail, it is useful to introduce some classifications generally adopted in literature. A first distinction can be made on the basis of the nature of the perturbative force, dividing perturbations into two basic categories:

- **Conservative forces**: these are forces of a gravitational nature that, while changing the orientation or shape of the orbit, do not alter the total mechanical energy of the system. This category includes effects due to interaction with third bodies and those resulting from an accurate representation of the Earth's gravitational field.
- **Non-conservative forces**: these are forces that vary the total mechanical energy of the satellite and act in a dissipative or propulsive manner. These include atmospheric drag and solar radiation pressure.

A further classification can be made on the basis of the effects produced on the time evolution of Keplerian orbital elements. Perturbations, in fact, cause these elements to vary over time in different ways (as depicted in Figure 2.7), resulting in:

- **Secular variations**: monotonous and cumulative changes in the orbital elements, typically linear in nature over time.

- **Long-period variations**: oscillations of the elements that develop on longer timescales than the orbital period.
- **Short-period variations**: rapidly oscillating effects that recur several times within a single orbital period.

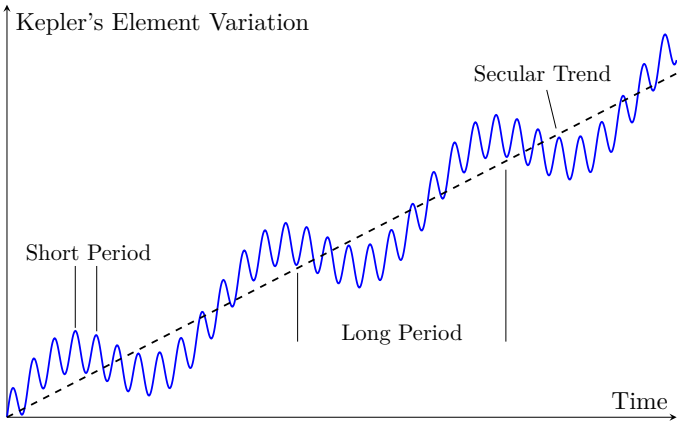


Figure 2.7: Periodic Variations due to Perturbations

2.3.1 Atmospheric Drag

Among all the perturbative forces acting on a satellite in LEO, the most significant is atmospheric drag, generated by the interaction between the spacecraft and particles in the Earth's atmosphere. Although at such altitudes the atmospheric density is extremely low, the satellite's high orbital velocity, typically on the order of several kilometers per second, makes impacts with atmospheric molecules non-negligible. These repeated impacts result in a net force acting in the opposite direction of motion, progressively retarding the satellite's motion. Being a non-conservative force, drag causes a loss of mechanical energy in the system. This results, over time, in a reduction in the semi-major axis and an increase in orbital eccentricity. As a result, the perigee of the orbit gradually lowers until leading, in the absence of corrective action, to atmospheric reentry of the satellite. Although this effect may initially appear to be an operational limitation, if properly considered at the mission design stage, it can prove beneficial. In particular, for sufficiently low operational orbits, drag action can ensure natural satellite reentry within the 5-year period established by the *ESA Space Debris Mitigation Requirements* [10].

The basic expression for the magnitude of the perturbative acceleration due to aerodynamic drag originates from the standard drag force formulation:

$$D = \frac{1}{2} C_D A \rho v^2 \tag{2.48}$$

Since drag acts in the direction opposite to the motion, the corresponding perturbative force is expressed as:

$$\mathbf{F}_{drag} = -\frac{1}{2} C_D A \rho v_{rel}^2 \frac{\mathbf{v}_{rel}}{v_{rel}} \tag{2.49}$$

where:

- C_D is the dimensionless *drag coefficient*, which quantifies the susceptibility of the satellite to aerodynamic forces. C_D is often approximated as 2.2 when not precisely known.
- ρ is the *atmospheric density* at the satellite's altitude, which is the most challenging parameter to estimate accurately.
- A is the projected *cross-section area* of the satellite perpendicular to the direction of motion, which depends on the satellite's orientation and is thus nontrivial to evaluate.
- $\mathbf{v}_{rel}$ is the *satellite's velocity relative to the rotating atmosphere*, which is not the velocity vector typically found in the state vector:

$$\mathbf{v}_{rel} = \frac{d\mathbf{r}}{dt} - \boldsymbol{\omega}_{\oplus} \times \mathbf{r} \tag{2.50}$$

  with $\boldsymbol{\omega}_{\oplus} = 7.292\,11 \times 10^{-5}\,\mathrm{rad\,s^{-1}}$ being the angular velocity vector of Earth's rotation.

The perturbative acceleration due to drag is then straightforwardly obtained by dividing the force by the satellite's mass:

$$\mathbf{a}_{drag} = \frac{\mathbf{F}_{drag}}{m_s} = -\frac{1}{2}\frac{C_D A}{m_s}\rho v_{rel}^2 \frac{\mathbf{v}_{rel}}{v_{rel}} \tag{2.51}$$

Atmospheric Density

The determination of atmospheric density deserve particular attention, as it represent one of the main sources of uncertainties in modelling atmospheric drag. As discussed by *Vallando* [1], the density of the upper atmosphere varies continuously as a result of the combined effects of molecular composition, incident solar flux, and geomagnetic activity. Given the dynamic and complex nature of these interactions, several empirical density models have been developed (e.g., NRLMSISE-00, Jacchia, and DTM) which require solar and geomagnetic indices as input. However, due to the computational efficiency required by the RL training performed in this thesis (as will be discussed in later chapters), a simpler static approach has been adopted. In particular, the atmospheric density is approximed with an exponential model that assumes a spherically symmetric distribution of particles and a density decreasing exponentially with altitude according to:

$$\rho(h) = \rho_0 \exp\left[-\frac{h - h_0}{H}\right] \tag{2.52}$$

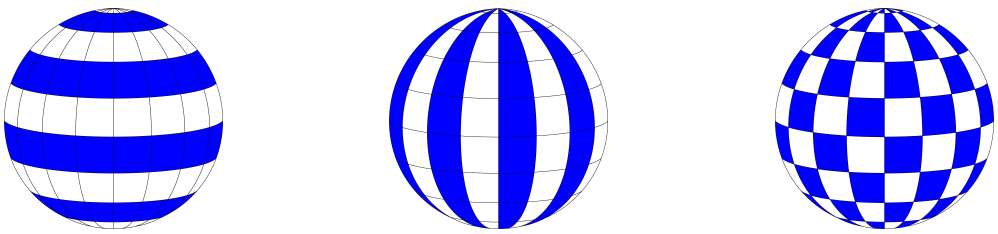
where $h = r - R_{\oplus}$, with $R_{\oplus} = 6378.1366$ km Earth's mean equatorial radius, is the altitude above Earth's surface, ρ_0 is the reference density at altitude h_0, and H is the scale height. The values of ρ_0, h_0, and H for different altitude ranges are reported in Table B.1 (Appendix B).

2.3.2 Earth's Gravitational Potential

In the framework of the unperturbed two-body problem, Earth is considered a perfectly spherical body with uniform mass distribution. This assumption leads to a simple expression for the gravitational potential, $U = -\mu/r$, and generates central, inverse-square accelerations directed toward the Earth's center of mass. However, Earth is not a perfect sphere. It is better approximated as an *oblate spheroid*, and its internal mass distribution is far from homogeneous. These deviations from spherical symmetry lead to a gravitational field that is not uniform and varies with position. As a consequence, satellite motion is affected by additional accelerations not accounted for in the two-body model. The actual gravitational potential of the Earth generates a set of equipotential surfaces that are not spherical. The surface that best represents the Earth's gravitational "shape" is the *geoid*, which is defined as the surface perpendicular at every point to the direction of the gravitational acceleration. The geoid is affected by local mass concentrations such as mountain ranges, ocean trenches, and the planet's overall flattening at the poles. To model these effects a more accurate representation of the Earth's gravity field requires an aspherical potential, where the deviations from the spherical term are expressed through a hierarchy of correction terms that depend on the internal mass distribution. These correction terms are classified into three categories of spherical harmonic, each associated with different symmetries of the mass distribution:

- **Zonal harmonics**: are symmetric about the Earth's rotation axis and do not vary with longitude. The most significant of these is the J_2 term, which reflects the planet's equatorial bulge and is the dominant source of gravitational perturbation in LEO.
- **Sectoral harmonics**: vary with longitude but not with latitude. Sectorial terms become relevant in regions with repeated patterns in longitude, such as tectonic belts or oceanic ridges.
- **Tesseral harmonics**: are the most general class, varying with both latitude and longitude. They represent localized mass anomalies, such as continental-scale features, and form a checkerboard-like pattern of gravitational highs and lows over the Earth's surface.

A visual representation of these harmonics can be obtained by mapping their associated potential terms onto a globe: zonal terms appear as horizontal bands (Figure 2.8a), sectorial as vertical slices (Figure 2.8b), and tesseral as patch-like patterns (Figure 2.8c).



(a) Zonal Harmonics (b) Sectoral Harmonics (c) Tesseral Harmonics

Figure 2.8: Spherical Harmonics Term

The perturbations arising from the Earth's gravitational potential produce measurable effects on satellite motion, especially in LEO. Among these, the most relevant are the secular perturbation caused by the zonal harmonics (particularly J_2) that cause a gradual shift of the RAAN and argument of perigee. In addition, all harmonic components contribute to periodic oscillations in the orbital elements. The magnitude of these perturbations generally increases with orbit eccentricity and decreases with increasing semimajor axis. Low-altitude and elliptical orbits are particularly sensitive to the Earth's gravitational irregularities.

Earth Gravitational Potential Modeling with EGM2008

In order to capture the effects of Earth's non-spherical mass distribution on satellite dynamics, the gravitational potential must be expressed in a form that accounts for such irregularities. This is typically done using global geopotential models derived from spherical harmonic expansions.

The Earth Gravitational Model 2008 (EGM2008) is one of the most widely used geopotential reference models to represent the Earth's gravitational field. It is based on a spherical harmonic expansion with coefficients computed up to degree and order $n_{\max} = 2190$ [43]. According to this model, the Earth's gravitational potential, expressed in a geocentric spherical RF is expressed as a sum of fully-normalized spherical harmonic terms:

$$U(r,\psi,\lambda) = \frac{\mu}{r}\left[1 + \sum_{n=2}^{n_{\max}} \left(\frac{R_\oplus}{r}\right)^n \sum_{m=0}^{n} \left(\bar{C}_{nm}\cos m\lambda + \bar{S}_{nm}\sin m\lambda\right)\bar{P}_{nm}(\cos\psi)\right] \tag{2.53}$$

where ψ is the *geocentric colatitude*, measured from z-axis of the Earth-Centered Earth-Fixed (ECEF) RF (a detailed description of which is given in Appendix A), λ is the *geocentric longitude*, measured in the equatorial plane from the x-axis of ECEF RF, $R_\oplus$ is *Earth's mean equatorial radius*, $\bar{C}_{nm}$ and $\bar{S}_{nm}$ are the *fully-normalized spherical harmonic coefficients* of degree n and order m in ECEF RF, and finally $\bar{P}_{nm}(\cos\theta)$ are the *fully-normalized associated Legendre functions*.

The μ/r in Eqn. (2.53), represents the spherically symmetric gravitational potential (equivalent to the classical two-body central potential), while the two summations, $\sum_{n=2}^{n_{\max}}\sum_{m=0}^{n}(\dots)$, model the deviations from spherical symmetry due to Earth oblateness and local mass anomalies.

To quantify the perturbating acceleration resulting from this aspherical potential, which directly effects the orbital motion of the satellite, the gradient of $U(r,\psi,\lambda)$ is computed as:

$$\mathbf{a}_{sph} = \nabla U(r,\psi,\lambda) = \frac{\partial U}{\partial r}\hat{\mathbf{r}} + \frac{1}{r}\frac{\partial U}{\partial \psi}\hat{\boldsymbol{\psi}} + \frac{1}{r\sin\psi}\frac{\partial U}{\partial \lambda}\hat{\boldsymbol{\lambda}} \tag{2.54}$$

The acceleration obtained must then be transformed into the ECI RF through the intermediate ECEF RF, in order to be used within the numerical integration process.

Although truncating the spherical harmonic expansion allows a trade-off between modeling accuracy and computational efficiency, the explicit evaluation of the potential

gradient and the subsequent RF transformations makes this approach computationally expensive for this work. Therefore, the analytical expression for the perturbing accelerations, in ECI RF, associated with the zonal harmonics up to J_6 are adopted in this thesis. Among these, the contribution of J_2 and J_3 are reported below, as they represent the dominant components of the gravitational perturbation in LEO.

$$\mathbf{a}_{J_2} = -\frac{3J_2\mu R_\oplus^2}{2r^5}\begin{bmatrix}\left(1-5\frac{r_K^2}{r^2}\right)r_I\\ \left(1-5\frac{r_K^2}{r^2}\right)r_J\\ \left(3-5\frac{r_K^2}{r^2}\right)r_K\end{bmatrix} \tag{2.55}$$

with $J_2 = 1.08262668 \cdot 10^{-3}$.

$$\mathbf{a}_{J_3} = -\frac{5J_3\mu R_\oplus^3}{2r^7}\begin{bmatrix}\left(3r_K-7\frac{r_K^3}{r^2}\right)r_I\\ \left(3r_K-7\frac{r_K^3}{r^2}\right)r_J\\ \left(6r_K^2-7\frac{r_K^4}{r^2}-\frac{3}{5}r^2\right)\end{bmatrix} \tag{2.56}$$

with $J_3 = -2.5324105 \cdot 10^{-6}$. The remaining terms up to J_6, are provided in Appendix C.

2.3.3 Solar Radiation Pressure

Like atmospheric drag, solar radiation pressure also constitutes a non-conservative perturbation, as it alters the mechanical energy of the system. This force is generated by the interaction between the photons emitted by the Sun and the satellite surface: the energy carried by the photons, upon impact, can be absorbed or reflected depending on the optical properties of the surface material, generating a net change in momentum. The magnitude of the effect depends on the apparent surface exposed to the Sun, i.e., the portion of the satellite orthogonal to the direction of propagation of the radiation. A particularly complex aspect of modeling this perturbation lies in the difficulty of accurately characterizing solar cycles, variations in solar activity and the illuminated geometry of the satellite. During phases of high activity, such as solar storms, the pressure exerted may exceed, in terms of instantaneous acceleration, the one due to other perturbative forces, especially for higher orbits. Conversely, during quiet periods of solar activity, the effect may be almost negligible.

To derive the formulation for estimating the acceleration caused by solar radiation pressure, it is necessary to first estimate the energy intensity of the incident radiation from the Sun. For simplicity, in the context of solar radiation pressure analysis, reference is often made to the solar radiation constant or *solar flux* (SF), the average of which is equal to:

$$SF = 1367\,\mathrm{W\,m^{-2}} \tag{2.57}$$

This constant represents the intensity of the power radiated by the Sun reaching, per unit area, an area placed perpendicular to the Sun's rays at the average Earth-Sun distance. It constitutes an estimate of the energy density incident on the Earth system under average conditions. However, the value of solar flux is not perfectly constant, but undergoes variations related to solar activity and, to a more regular extent, to the variation of the Earth-Sun distance during the year. To account for these effects, a time-varying formulation is proposed in [44], which allows the intensity of solar radiation to be approximated more accurately as a function of the day of the year:

$$SF = \frac{1358}{1.004 + 0.0034 \cos D_{aphelion}} \tag{2.58}$$

where D_{aphelion} represents the number of days elapsed since aphelion, expressed as a fraction of the Earth's period of revolution (365 days).

Now dividing the intensity of incident radiation by the speed of light in vacuum ($c = 299\,792.458\,\text{km}\,\text{s}^{-1}$), the value of the pressure exerted by solar radiation is obtained as the change in momentum per unit time and area:

$$p_{srp} = \frac{SF}{c} \tag{2.59}$$

Subsequently, the force exerted by solar radiation pressure is expressed as:

$$\mathbf{F}_{srp} = -p_{srp} c_R A_\odot \frac{\mathbf{r}_{sat\odot}}{r_{sat\odot}} \tag{2.60}$$

where:

- $A_\odot$ represents the *area of the satellite directly exposed to solar radiation.* A common approximation is to assume that this surface constantly maintains a perpendicular attitude with respect to the direction of the solar rays, i.e., $\phi_{\text{inc}} = 0°$ (see Figure 2.9). Although this assumption is unrealistic in an operational context, it allows for a preliminary estimate of the effects of solar pressure.
- c_R is the *reflectivity coefficient*, a dimensionless value between 0.0 and 2.0, which depends on the optical properties of the surface materials and the geometry of the satellite. Accurate determination of c_R is particularly complex: it can vary over time and is difficult to predict, especially in the case of satellites characterized by composite surfaces and different materials.
- $\mathbf{r}_{sat\odot} = \mathbf{r}_{\oplus\odot} - \mathbf{r}$ is the relative position vector between the satellite and the Sun.

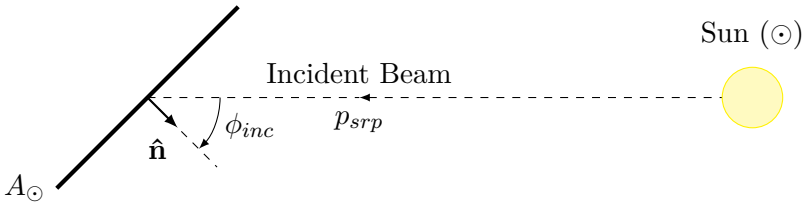


Figure 2.9: Incident Solar Radiation

To determine the perturbative acceleration associated with solar radiation pressure, it is sufficient to divide the calculated force by the mass of the satellite:

$$\mathbf{a}_{srp} = \frac{\mathbf{F}_{srp}}{m_s} = -\frac{p_{srp} c_R A_\odot}{m_s} \frac{\mathbf{r}_{sat\odot}}{r_{sat\odot}} \tag{2.61}$$

Overall, solar radiation pressure induces periodic variations in all orbital elements, which can surpass the influence of atmospheric drag at altitudes above approximately 800 km.

2.3.4 Third-Body Effects

In the case of LEO satellites, the gravitational influence exerted by external celestial bodies, such as the Moon and the Sun, is of significantly less magnitude than other perturbative sources already discussed, such as drag and the irregularity of the Earth's gravitational field. However, the effects of these perturbations may become non-negligible when considering long-term propagation scenarios, as they can induce gradual variations in the orbital elements of the satellite. The third-body perturbation results from the difference between the gravitational attraction exerted by the perturbing body on the satellite and that exerted on the central body (Earth). This effect is purely gravitational, and therefore conservative in nature, falling into the category of conservative perturbation.

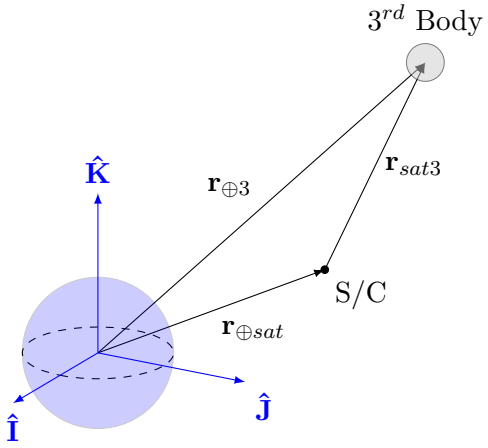


Figure 2.10: Three Body Geometry

As shown in Figure 2.10, the geometry of the third-body perturbation is defined by two position vectors: $\mathbf{r}_{\oplus 3}$ representing the position of the perturbing body with respect to Earth, and $\mathbf{r}_{sat3} = \mathbf{r}_{\oplus 3} - \mathbf{r}_{\oplus sat}$ representing the position of the same body with respect to the satellite. The perturbation is obtained as the difference between the gravitational acceleration exerted by the perturbing body on the satellite and that exerted on the Earth, and is expressed as:

$$\mathbf{a}_3 = \mu_3 \left(\frac{\mathbf{r}_{sat3}}{r_{sat3}^3} - \frac{\mathbf{r}_{\oplus \mathbf{3}}}{r_{\oplus 3}^3} \right) \tag{2.62}$$

where $\mu_3 = G\, m_3$ is the gravitational parameter of the third body. For the Sun, $\mu_\odot = 1.327\,124\,400\,18 \times 10^{11}\,\mathrm{km}^3/\mathrm{s}^2$, while for the Moon, $\mu_{\leftmoon} = 4.904\,869\,5 \times 10^3\,\mathrm{km}^3/\mathrm{s}^2$.

Two contributions can be distinguished in Eqn. (2.62) as proposed by [1]. The *direct term*, represented by the term $\frac{\mathbf{r}_{sat3}}{r_{sat3}^3}$, describes the acceleration exerted directly by the perturbing body on the satellite, and the *indirect term* , expressed by $\frac{\mathbf{r}_{\oplus 3}}{r_{\oplus 3}^3}$, represents the acceleration exerted by the same perturbing body on Earth.

Moon and Sun Ephemeredis

The third-body perturbation strongly depends on the quality of the ephemeris data used to determine the position of the perturbing body with respect to the Earth, $\mathbf{r}_{\oplus 3}$. The same applies to the solar radiation pressure, where knowing the Sun's position relative to Earth ($\mathbf{r}_{\oplus\odot}$)is essential to correctly evaluate the direction of the incident radiation and the resulting force on the spacecraft. In the context of this thesis, the **Moon ephemerides** are determined using the formulation proposed by *Simpsons* [45], which is based on JPL's Development Ephemeris DE200 and represent the Moon's position through sine and cosine terms derived from a Fourier transform:

$$\mathbf{r}_{\oplus\mathbb{C}} = \sum_{m=1}^{N_n} \boldsymbol{a}_{nm} \sin\left(\boldsymbol{\omega}_{nm} T_{JD} + \boldsymbol{\delta}_{nm}\right) \tag{2.63}$$

where $\mathbf{r}_{\oplus\mathbb{C}}$ is Moon's center of mass position in ECI RF, N_n is the order of the series, $\boldsymbol{a}_{nm}$, $\boldsymbol{\omega}_{nm}$, and $\boldsymbol{\delta}_{nm}$ represent the amplitudes, frequencies, and phase constant of the series, respectively, and T_{JD} is time measured in Julian centuries since Epoch J2000. In this work, a seventh-order series is adopted, as reported in [46], where the corresponding numerical values of the matrices $\boldsymbol{a}_{nm}$, $\boldsymbol{\omega}_{nm}$, and $\boldsymbol{\delta}_{nm}$ are provided.

To determine the **Sun ephemerides** the same analytical approximation reported in [46] is adopted. The computation starts from the definition of the Sun's mean longitude in the ecliptic coordinate system referred to the J2000 epoch:

$$\lambda_{M_\odot} = 280.460^\circ + 36{,}000.771\, T_{JD} \tag{2.64}$$

from which the Sun's mean anomaly is obtained as:

$$M_\odot = 357.5291092^\circ + 35{,}999.050\, T_{JD} \tag{2.65}$$

The true longitude of the Sun, representing the apparent position of the Sun along the ecliptic, corrected for the effect of Earth's orbital eccentricity can be approximated as a function of the Sun's mean anomaly, yielding:

$$\lambda_\odot = \lambda_{M_\odot} + 1.914666471 \sin(M_\odot) + 0.019994643 \sin(2M_\odot) \tag{2.66}$$

The Sun-Earth distance, in astronomical units (AU) is then approximated as:

$$r_{\oplus\odot} = 1.000140612 - 0.016708617 \cos(M_\odot) - 0.000139589 \cos(2M_\odot) \tag{2.67}$$

Finally, the Earth-Sun position vector in the ECI RF, is obtained as:

$$\mathbf{r}_{\oplus\odot} = r_{AU} \begin{bmatrix} \cos(\lambda_\odot) \\ \cos(\varepsilon)\sin(\lambda_\odot) \\ \sin(\varepsilon)\sin(\lambda_\odot) \end{bmatrix} \tag{2.68}$$

with ε, mean inclination of the ecliptic, is used to rotate the coordinates from the ecliptic to the equatorial plane.

2.4 Custom Orbital Propagator

After introducing the main perturbative effects that influence the motion of a satellite in LEO, it is necessary to define how the orbit evolves over time starting from a given epoch and initial condition. In the context of this thesis, different scenarios characterized by different operational needs require the precise determination of the state evolution of a satellite and/or space debris over time. This information is essential to perform conjunction risk assessment analyses and evaluate the orbital evolution following the execution of a set of CAMs . Since these scenarios have different requirements in terms of dynamic accuracy and computational cost, for example, during the training phase of the RL agent, in the evaluation of the learned policy, or in the propagation of the post-CAMs, a dedicated orbital propagator has been developed, capable of integrating the equations of motion under a customizable set of dynamic models.

2.4.1 Cowell's Formulation

Given the flexibility needs previously discussed, the orbital propagator developed in this work is based on Cowell's formulation, which provides a versatile framework for orbit integration. In this approach, the motion of the satellite is described by a set of second-order differential equations expressed in Cartesian coordinates. The total acceleration is defined as the sum of the central gravitational attraction, i.e. the two-body problem equation of motion, and all additional perturbative contributions:

$$\frac{d^2\mathbf{r}}{dt^2} = -\frac{\mu}{r^3}\mathbf{r} + \mathbf{a}_p \tag{2.69}$$

where $\mathbf{a}_p = \sum_i \mathbf{a}_i$ represents the total perturbing acceleration acting on the satellite, obtained as the sum of all the individual contributions included in the selected dynamical models, such as atmospheric drag ($\mathbf{a}_{drag}$), Earth's gravitational harmonics ($\mathbf{a}_{J_2}$), solar radiation pressure ($\mathbf{a}_{\text{srp}}$), and third-body effects ($\mathbf{a}_3$).

One of the main advantages of this formulation, lies in its flexibility, any arbitrary perturbing acceleration can be directly incorporated into the equations of motion as an additive term. This property makes Cowell's formulation particularly well suited for numerical propagation adopted in this thesis, where each perturbation can be treated independently and added linearly to the total acceleration.

2.4.2 Numerical Integration of the Equation of Motion

The numerical integration of the equation of motion is carried out using the explicit Runge-Kutta method of order 8(5,3), known as DOP853, implemented in the `solve_ivp` routine of the Python `SciPy` library. For numerical integration purpose, the set of second order differential equation of motion of the Cowell's formulation, Eqn. (2.69), is rewritten as an equivalent system of six first-order differential equation:

$$\dot{\mathbf{X}}(t) = \begin{bmatrix} \dot{\mathbf{r}}(t) \\ \dot{\mathbf{v}}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{v}(t) \\ -\frac{\mu}{r^3}\mathbf{r}(t) + \mathbf{a}_p(t) \end{bmatrix} \tag{2.70}$$

where $\dot{\mathbf{X}}(t)$ is simply the derivative of the state vector $\mathbf{X}(t)$:

$$\mathbf{X}(t) = \begin{bmatrix} \mathbf{r}(t) \\ \mathbf{v}(t) \end{bmatrix} \tag{2.71}$$

The initial condition, necessary for the numerical integration, is defined by the state vector at the reference epoch t_0:

$$\mathbf{X}(t_0) = \begin{bmatrix} \mathbf{r}(t_0) \\ \mathbf{v}(t_0) \end{bmatrix} \tag{2.72}$$

It can be derived through different approaches, for instance, from Two-Line Elements (TLEs) or from defined Keplerian orbital elements converted into Cartesian Coordinates using the procedure described in Section 2.2.3. The integration is then performed with an adaptive internal time step automatically controlled by DOP853 integrator based on the specified error tolerances (`rtol` and `atoll`). In addition to the internal step control, the temporal resolution of the stored results is defined by the selected output time grid. A fine temporal discretizations is adopted, for instance, in conjunction risk assessment analysis, where a higher time resolution is required to accurately capture the evolution of the relative distance between the satellite and the debris.

In conclusion, this chapter has presented the theoretical background and numerical formulation required to model the orbital motion of satellite (or any other space objects) under both ideal and perturbed conditions. The following chapter introduces the fundamentals of RL, its extension with deep learning techniques, DRL, and the algorithm adopted in this thesis to train the agent. These elements are later integrated with the orbital dynamics framework developed in this chapter to enable autonomous CAM planning in LEO.

Chapter 3

Reinforcement Learning

Reinforcement Learning (RL) is a field of Machine Learning (ML) that focuses on the problem of how an agent learns to make decisions in a dynamic environment in order to achieve one or more objectives. The agent interacts with the environment through actions and receives feedback in response, which can take the form of rewards or penalties, depending on the effects produced by its choices. This process develops according to trial and error logic, in which the agent explores different behavioral strategies and evaluates their consequences, progressively adapting its decisions. The goal is to maximize the cumulative reward over time, i.e., to identify those sequences of actions that, in the long run, lead to the most favorable outcomes.

Before addressing the mathematical framework that describes RL, it is necessary to frame its role within the broader discipline of ML. A brief introduction to the main learning paradigms, namely supervised and unsupervised learning, allow a better understanding of why RL is considered a distinct approach with its unique characteristics.

3.1 Introduction to Machine Learning

ML can be defined as the set of methodologies and algorithms that enable a computer system to progressively improve its performance in executing a task thanks to accumulated experience. One of the best known and most commonly accepted definitions, proposed by *Tom M. Mitchell* [47], states that:

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

In more practical terms, this definition implies that a system can be considered "capable of learning", by repeatedly performing a certain task, when it is able to use data from previous experiences to refine its future decisions or predictions. The key element is therefore the presence of past data that serves as experience and drives the process of continuous improvement. ML is therefore a discipline that allows the development of algorithms capable of automatically adapting to data, without having to specify all the

rules of the problem in advance. This feature makes it a particularly powerful tool for dealing with complex situations, where it is difficult or impractical to formalize every possible scenario using predefined deterministic rules.

3.1.1 Supervised Learning

Supervised learning is the most commonly used and established ML paradigm. This approach is based on the idea that an algorithm can learn from a set of labeled data, consisting of input-output pairs, known as labeled examples. The goal is to build a predictive model capable of generalizing the knowledge acquired, i.e., providing the correct answer even when faced with new data that has never been seen before. The supervised learning process typically consists of two distinct phases. During the *training phase*, the algorithm processes the labeled data to identify the underlying relationships between inputs and outputs, until it builds a model capable of representing them. Subsequently, in the *prediction or inference phase*, this model is used to deduce the output corresponding to a new input, thus replicating the behavior "learned" from the training set. There are two fundamental tasks that fall within this paradigm:

- *Classification*: consists in assigning each example to a discrete category, predicting which class a given sample belongs to. Typical examples are image recognition, medical diagnosis, or document categorization.
- *Regression*: consists of predicting a continuous variable, such as a number or quantity. In this case, the model provides a numerical estimate that is as close as possible to the actual value.

The main driver behind supervised learning is the use of past experience to improve future performance. In this context, experience coincides with training data, which forms the basis of the learning process. The quality and representativeness of this data play a crucial role, an incomplete or noisy dataset inevitably compromises the accuracy of the model, reducing its ability to generalize.

3.1.2 Unsupervised Learning

Unsupervised learning is another fundamental paradigm of ML. Unlike supervised learning, in this case there is no labeled data or input-output pairs from which to learn. The goal is therefore not to predict a target variable, but rather to directly analyze the raw data in order to identify underlying structures, patterns, or natural groupings within it. For this reason, unsupervised learning is often described as a pattern discovery or knowledge discovery approach. The main tasks of unsupervised learning include:

- *Clustering*: dividing data into homogeneous groups (clusters), in which elements of the same group are similar, while those of different groups show significant differences.
- *Dimensionality reduction*: reducing the number of variables while maintaining the fundamental properties of the data, such as the distance relationships between samples.

- *Anomaly detection*: identifying examples that deviate from the general behavior of the data, considered as outliers or rare events of particular interest.

3.1.3 Reinforcement Learning as a ML paradigm

Within the broad field of ML, RL occupies a special position. Unlike supervised methods, in which models are trained on labeled data sets provided by an external supervisor, RL is based on a trial and error process. The agent does not know in advance what the correct actions are, but discovers them progressively thanks to feedback obtained from interactions with the environment. At first glance, it might seem to be part of unsupervised learning, because it is not based on labeled examples provided by an external supervisor. However, as pointed out by *Suttun & Barto* [48], it also differs from the latter because it does not aim to identify hidden structures or correlations in the data. RL has such unique characteristics (interaction, feedback, trial and error) that it is considered a third paradigm in the context of ML, distinct from both supervised and unsupervised learning. Furthermore, one of the challenges that emerges in RL, and not in other types of learning, is the trade-off between exploration and exploitation [48]. To achieve a high level of reward, an RL agent must prioritize actions that have proven effective in generating positive results in the past. At the same time, however, it is also necessary for the agent to try new actions that have never been selected before, in order to discover potentially better alternatives. The agent must therefore know how to exploit the experience already acquired to continue to obtain rewards, but at the same time explore new possibilities to improve the quality of its future decisions.

3.2 Reinforcement Learning Formalism

3.2.1 Markov Decision Process

Reinforcement Learning is formalized through Markov Decision Processes (MDP), which define the environment in terms of states, actions, and rewards, describing how the agent interacts with it in a sequential decision-making process, as shown in Figure 3.1. At a generic discrete time instant t, the agent observes the state $s_t \in \mathcal{S}$ and selects an action $a_t \in \mathcal{A}$. The environment receives this action, updates its dynamics, and moves on to the next time step. At the same time, it returns the new state s_{t+1} and a reward $r_{t+1} \in \mathcal{R}$ to the agent. The (s_t, a_t, r_{t+1}) tuple is commonly called an experience. This interaction can repeat indefinitely or end when a terminal state or a maximum number of time steps $t = \mathcal{T}$ is reached. The interval from the initial moment $t = 0$ to the termination of the environment is defined as an episode. A sequence of experiences observed in an episode is usually indicated as a trajectory, $\tau = (s_0, a_0), (s_1, a_1, r_2), (s_2, a_2, r_3) \dots$. To learn the optimal strategy for solving the proposed problem, an agent typically needs a large number of episodes, which can range from a few hundred to thousands, depending on the complexity of the task at hand.

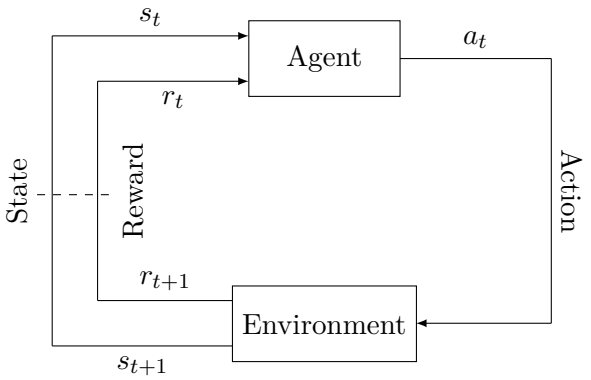


Figure 3.1: Agent–Environment interaction in a Markov Decision Process

In addition to this operational description, an MDP can be formalized mathematically as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$ [49], where:

- $\mathcal{S}$: set of states representing the environment.
- $\mathcal{A}$: set of actions available to the agent in each state.
- $\mathcal{P}(s_{t+1}|s_t, a_t)$: transition probability function, defines the likelihood of reaching the state s_{t+1} from s_t after taking action a_t.
- $\mathcal{R}(s_t|a_t)$: reward function assigned to the agent after taking action a_t in state s_t
- $\gamma \in [0, 1]$: discount factor, weighs the importance of future rewards against immediate ones.

The fundamental characteristic of an MDP is the Markov property, according to which the future evolution of the system depends exclusively on the current state and action, resulting in independence from the sequence of past states and actions. In other words, the future is conditionally independent from the past given the present. Formally:

$$\mathcal{P}(s_{t+1}|s_t, a_t, s_{t-1}, a_{t-1}, \dots) = \mathcal{P}(s_{t+1}|s_t, a_t) \tag{3.1}$$

To finalize the formulation of the problem, it is necessary to introduce the concept of return (G_t) which represents the objective that the agent attempts to maximize during interaction with the environment. Considering a trajectory τ of an episode, the return starting from a generic instant t is defined as the sum of future rewards discounted appropriately:

$$G_t \doteq r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \tag{3.2}$$

The value of γ determines the relative importance between immediate rewards and future rewards, such that:

- If $\gamma = 0$, the agent is short-sighted, as it exclusively pursues the maximization of the immediate reward r_{t+1}.

- If $\gamma \approx 1$, the agent takes a long-term view, attributing significant weight to future rewards as well.
- For intermediate values of γ, behavior is a compromise between short-term and long-term optimization.

3.2.2 Policy and Value Functions

After introducing the concept of return as a measure of agent performance, it is natural to ask how the agent selects its actions to maximize this objective. To this end, the concept of policy (π) is introduced, which represents the strategy adopted to choose an action based on the current state. A policy can be:

- *Stochastic*: when it defines a probability distribution over the possible actions in state s. In this case, the policy is formally expressed as $\pi(a|s)$, i.e., the probability of performing action $a \in \mathcal{A}$ given the current state $s \in \mathcal{S}$.
- *Deterministic*: when it associates each state $s \in \mathcal{S}$ with a specific action $a = \pi(s)$.

Once a policy has been defined, it is essential to assess how effective it is in guiding the agent towards the goal. This is why value functions are introduced, which estimate the expected return that the agent can accumulate over time starting from a state and behaving according to the policy in question. The *state-value function* $v_\pi(s)$ is the expected return starting from state s and following policy π:

$$v_\pi(s) = \mathbb{E}_\pi[G_t|s_t = s] \tag{3.3}$$

where $\mathbb{E}_\pi[\cdot]$ denotes the expected value of a variable given that the agent follows policy π, and t is any time step. The *action-value function* $q_\pi(s, a)$, also called the *quality function* represents instead the expected return starting from state s, taking action a, and then following policy π:

$$q_\pi(s, a) = \mathbb{E}_\pi[G_t|s_t = s, a_t = a] \tag{3.4}$$

One of the fundamental properties of value functions is that they satisfy a recursive relation known as the *Bellman expectation equation*. This states that the value of a state (or state–action pair) can be expressed as the sum of the expected immediate reward and the discounted future value, assuming that the agent continues to follow policy π. For the state value function, it is given by:

$$v_\pi(s) = \mathbb{E}_\pi\big[r_{t+1} + \gamma v_\pi(s_{t+1}) \mid s_t = s\big], \tag{3.5}$$

while for the action value function:

$$q_\pi(s, a) = \mathbb{E}_\pi\big[r_{t+1} + \gamma q_\pi(s_{t+1}, a_{t+1}) \mid s_t = s,\, a_t = a\big]. \tag{3.6}$$

3.2.3 Optimality

A fundamental objective of reinforcement learning is to determine an optimal policy, denoted by π_*, i.e., a strategy that maximizes the expected return from each state. More formally, a policy π is considered optimal if its value function $v_\pi(s)$ is greater than or equal to that of any other policy π' for all states $s \in S$:

$$v_{\pi_*}(s) \geq v_{\pi'}(s), \quad \forall \pi', \ s \in S. \tag{3.7}$$

All optimal policies share the same optimal state value function, $v_*(s)$, and the same optimal action value function, $q_*(s, a)$, defined respectively as:

$$v_*(s) = \max_\pi v_\pi(s), \tag{3.8}$$

$$q_*(s, a) = \max_\pi q_\pi(s, a). \tag{3.9}$$

Analogously to the Bellman expectation equation, the optimal value functions satisfy a recursive relationship known as the *Bellman optimality equation*. For the state-value function, it states that the value of a state under an optimal policy must equal the expected return obtained by taking the best possible action in that state. Formally, this can be written as:

$$v_*(s) = \max_{a \in A(s)} q_*(s, a). \tag{3.10}$$

$$= \dots \tag{3.11}$$

$$= \max_a \mathbb{E}_{\pi_*}[r_{t+1} + \gamma v_*(s_{t+1}) | s_t = s, a_t = a] \tag{3.12}$$

Analogously, the Bellman optimality equation can also be formulated for the action-value function $q^*(s, a)$, relating each state–action pair to the immediate reward and the optimal value of the successor states.

The explicit resolution of Bellman's optimality equations represents a possible approach to determining an optimal policy, and thus to solving the reinforcement learning problem. In practice, this can be pursued through three main families of methods. *Dynamic Programming* (DP) methods require complete knowledge of the environment model, transition probabilities and reward functions, and allow value functions to be calculated iteratively and policies to be improved. *Monte Carlo* (MC) methods, on the other hand, do not require a model and estimate value functions as averages of returns obtained from complete episodes of interaction with the environment. Finally, *Temporal Difference* (TD) methods combine elements of both approaches: like Monte Carlo, they learn directly from experience without an explicit model, but, like dynamic programming, they update bootstrap estimates based on the values of subsequent states. These three families form the theoretical foundation on which the various reinforcement learning algorithms are developed.

3.2.4 Overview of RL Algorithms

Over the theoretical foundations of RL, numerous algorithms have been developed over time that address the decision-making problem from different perspectives, each with specific strengths and limitations. To better understand how these approaches fit within the discipline, it is useful to introduce a classification scheme that highlights the main conceptual subdivisions based on [50, 51] (see Figure 3.2).

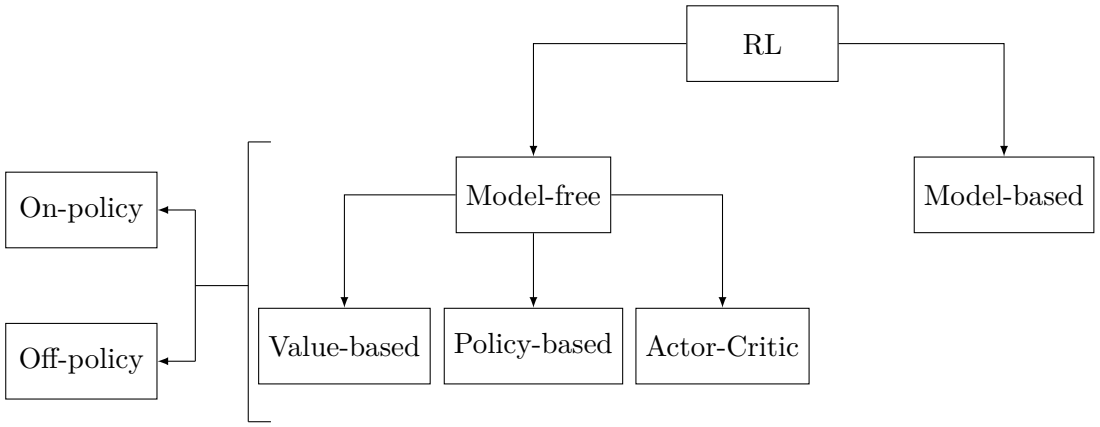


Figure 3.2: Overview of Reinforcement Learning Algorithms

Model-based vs Model-free Methods The main difference concerns the availability of a model of the environment. In *model-based* methods, the agent has (or learns) an explicit representation of the dynamics of the system, including transition probabilities and the reward function, which can be used to simulate trajectories and plan future actions. In *model-free* methods, on the other hand, no model is used: the agent learns directly from experience gathered through interaction with the environment. This simplicity reduces the computational costs of modeling, but generally requires a greater number of interactions to converge towards an effective strategy.

Policy-based vs Value-based Methods In *policy-based* methods, the agent directly learns the policy, modeled as a probability distribution on actions conditional on the state. This type of methods is guaranteed to converge to a locally optimal policy [48] but suffers of very high variance and sample-inefficiency. Algorithms such as REINFORCE are examples of this. *Value-based* methods are based on learning a value function that estimates the expected return from a state, or a state-action pair. The policy is then derived by choosing the actions that maximize this estimate. These approaches are typically more sample-efficient than policy based because they have lower variance and better use of gathered data but they are not guaranteed to converge to an optimum [52]. Classic algorithms such as SARSA and Q-learning belong to this category.

Actor-critic Methods *Actor-critic* methods are an intermediate category that combines the strengths of the two previous approaches. The actor updates the policy, while the critic evaluates actions using a value function, providing a low-noise correction signal.

This architecture reduces the variance of policy-based methods while maintaining their flexibility, and forms the basis of many modern extensions.

On-policy vs Off-policy Methods A distinction that cuts across all of the above categories concerns the relationship between the target policy, i.e., the one to be learned, and the behavior policy, i.e., the one that generates the training data. In *On-policy* methods, the two policies coincide: the agent learns directly from the same strategy it adopts to explore the environment. A typical example is SARSA, in which the update policy is the same as the one that produces the observed trajectories. In *Off-policy* methods, on the other hand, the target policy may differ from the behavior policy. In this scenario, the agent can learn the optimal strategy even from data generated by another policy, which may be more exploratory or already available. Q-learning is the canonical example of this approach.

3.3 Function Approximation & Deep RL

Classical RL algorithms rely on tabular representations of value and policy functions. In simple or low-dimensional environments, these tabular approaches are effective because the agent can explicitly store the value associated with each state-action pair. However, as the dimensionality of the problem increase, the number of possible states and actions grows exponentially with the number of variables, leading to what is known as the *"curse of dimensionality"* [48] and the computational cost increase exponentially.

To overcome this problem, modern RL algorithms employ function approximators to represent the policy and value functions. Rather than storing explicit values in a table, the agent learns a parameterized function that generalizes across the state-action space. This allows the learning process to scale efficiently to problems with high-dimensional or continuous domains.

Among various types of function approximators, *artificial neural network (ANN)*, or simply *neural network (NN)*, have become the most widely used due to their demonstrated ability to model highly nonlinear relationships and to approximate arbitrary functions. This concept is examined in greater detail by introducing the *Multilayer Perceptron (MLP)*, a class of NN widely used as general purpose function approximators for modeling complex nonlinear relationships.

3.3.1 Multilayer Perceptron

An MLP is a feedforward NN in which neurons are fully connected to each other and use nonlinear activation functions. Its architectural flexibility and ability to approximate any function, under certain conditions, make this model a fundamental element in the Deep Learning framework. The MLP consists of an *input layer*, one or more intermediate layers (*hidden layers*), and an *output layer*. Each layer consists of elementary units, neurons, which operate as computational nodes fully connected to the next layer.

Mathematically, the total input to neuron j in the first hidden layer is expressed as [53]:

$$a_j = \sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)} \quad , \quad z_j = h(a_j) \tag{3.13}$$

where:

- x_i denotes the *input variables* of the network, with $i = 1, \ldots, D$, D is the input space dimension.
- $w_{ji}^{(1)}$ are the *weights* connecting input x_i to neuron j in the first layer determining the relative importance.
- $w_{j0}^{(1)}$ represent the *bias*.
- The superscript $^{(1)}$ indicates that the corresponding parameters belong to the first layer of the network

It can therefore be stated that Eqn. (3.13) implies that, for each neuron, an *activation* (a_j) is generated computed as the weighted sum of the neuron's inputs, to which a bias term is added. Each activation is then transformed by a nonlinear and differentiable *activation function*, $h(\cdot)$, to give:

$$z_j = h(a_j) \tag{3.14}$$

The activation function introduces the nonlinearity necessary for the network to approximate complex relationships between input and output. Among the most common are the sigmoid function, $h(\cdot) = \frac{1}{1+e^{-(\cdot)}}$, the hyperbolic tangent, $h(\cdot) = \tanh(\cdot)$, and the rectified linear unit, ReLU, $h(\cdot) = \max(0, (\cdot))$.

The operation describing the calculation of the activation for each neuron (Eqn. (3.13)) can, for the l-th layer, be expressed in matrix form as:

$$\mathbf{a}^{(l)} = \mathbf{W}^{(l)} \mathbf{z}^{(l-1)} + \mathbf{b}^{(l)} \tag{3.15}$$

$$\mathbf{z}^{(l)} = h^{(l)} \left(\mathbf{a}^{(l)} \right) \tag{3.16}$$

Each layer l receives as input the activation vector from the previous layer $\mathbf{z}^{(l-1)}$, combines it linearly using the weight matrix $\mathbf{W}^{(l)}$ and the bias vector $\mathbf{b}^{(l)}$, and applies an element-by-element nonlinear transformation $h^{(l)}$. The input layer is simply $\mathbf{z}^{(0)} = \mathbf{x}$ and the output layer is $\mathbf{y} = \mathbf{z}^{(L)}$, where L is the total number of layers in the network. Figure 3.3 shows an MLP with one input layer, one hidden layer, and one output layer, which, following the nomenclature proposed in [53], is a two-layer MLP.

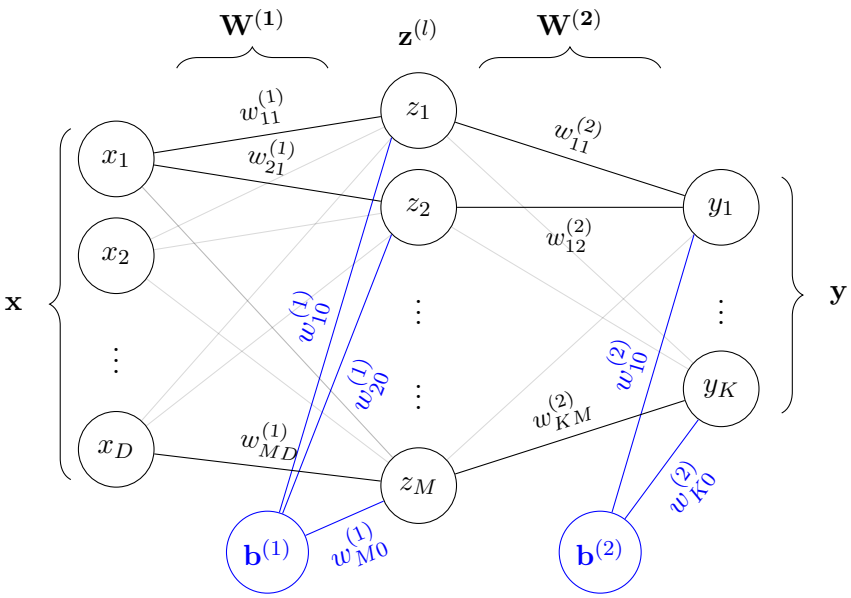


Figure 3.3: Architecture of a Multilayer Perceptron

The overall forward pass process through all layers performs a composition of linear and nonlinear transformations, which can be expressed as:

$$\mathbf{y} = F(\mathbf{x}, \boldsymbol{\theta}) \tag{3.17}$$

where $F(\cdot)$ represents the function calculated by the neural network, parameterized by the parameter vector $\boldsymbol{\theta} = \left\{\mathbf{W}^{(l)}, \mathbf{b}^{(l)}\right\}_{l=1}^{L}$.

MLP Training

Training an MLP means finding the values of $\boldsymbol{\theta}$ that minimize a loss function L, which measures the error between the target output, and the network-predicted output [52]. Minimizing this function is central to optimizing the performance of the MLP in approximating nonlinear functions. According to the type of problem being solved, there is a natural choice in the loss function [53]. For a fairly simple regression task, the commonly used loss function is the sum of squares error. Given a training set consisting of input vectors $\mathbf{x}_n$, with $n = 1, \ldots, N$, and the corresponding target vectors $\mathbf{t}_n$, the objective is to minimize:

$$L(\boldsymbol{\theta}) = \frac{1}{2} \sum_{n=1}^{N} \left\| \mathbf{y}\left(\mathbf{x_n}; \boldsymbol{\theta}\right) - \mathbf{t}_n \right\|^2 \tag{3.18}$$

Minimization of the loss function is achieved through optimization algorithms. The most basic and intuitive one is *sequential gradient descent*, which iteratively adjust the network parameters in the direction of the steepest decrease of the loss:

$$\boldsymbol{\theta}_{new} = \boldsymbol{\theta}_{old} - \eta \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) \tag{3.19}$$

with $\eta > 0$ is the learning rate, which controls the magnitude of the update steps.

Computing the gradient of the loss function can be complex because the output of each neuron indirectly depends on the parameters of the previous layers. To this end, the *backpropagation algorithm* [54] is used, which allows all the required derivatives to be determined efficiently by recursively applying the chain rule backwards. The iterative process underlying the optimization of neural network parameters can therefore be summarized in the following steps:

1. *Forward pass*: inputs are passed through the various layers of the MLP to calculate the activations of the neurons and get the network output.
2. *Loss function calculation*: network output is compared with the corresponding target value to evaluate the current performance of the model. The overall error is quantified using a loss function (e.g., SSE).
3. *Backward pass*: the backpropagation algorithm is applied to efficiently compute the derivatives of the loss function with respect to the network weights and biases.
4. *Network parameter update*: weights and biases are updated to reduce the loss function. In addition to gradient descent, which is the simplest method but can be a *"poor algorithm"* due to its slow convergence [53], more advanced optimizers are commonly used, such as *Adam* [55], which improve numerical stability and training speed.

This cycle of operations forms the core of the supervised learning algorithm for feed-forward networks, and is repeated iteratively on all examples in the dataset until convergence or a predefined stopping criterion is reached.

3.3.2 Overview of DRL Algorithms

The introduction of DL techniques, such as MLPs, as function approximators within the traditional RL framework allows one to overcome its inherent limitations and efficiently handling high-dimensional and continuous state-action spaces. When such DL architectures are employed with RL, the resulting approach is referred to as *Deep Reinforcement Learning (DRL)* [56]. Taking policy-based algorithms as an example, their extension to the DRL framework means that the objective is to learn a policy, represented by a feedforward neural network whose parameters θ[1] are adjusted through interaction with the environment. The policy, commonly referred to as policy network π_θ, is parameterized by θ, therefore, the process of learning an optimal policy corresponds to finding the optimal set of network parameters θ^* that maximize the expected cumulative reward [52]. For this reason, in the context of DRL, such algorithms are referred to as policy gradient methods, as they update the parameters of the policy network by performing gradient ascent on the expected return.

In general, DRL algorithms can be grouped into the same main families of methods described in Section 3.2.4, depending on how the policy and value functions are represented

[1]To simplify the notation, from this point onward, the NNs parameters will be written in regular font, while still representing vectors or sets of parameters as before, unless otherwise specified.

and updated. Among them, the actor-critic architecture is the foundation upon which many modern algorithms have been built, including Trust Region Policy Optimization (TRPO), Proximal Policy Optimization (PPO), Deep Deterministic Policy Gradients (DDPG), and Soft Actor-Critic (SAC).

3.4 Proximal Policy Optimization

The Proximal Policy Optimization (PPO) algorithm, introduced by *Schulman et al.* [57] is an on-policy actor-critic method belonging to the family of policy gradient algorithms, due to the way it updates the policy represented by the actor network.

Being of the actor-critic type, PPO relies on two independent NNs with complementary roles. The actor network, parameterized by θ, represents the stochastic policy $\pi_\theta(a|s)$, which defines the probability of selecting an action a_t given the current state s_t. In continuous action spaces, such as the one considered in this study, $\pi_\theta(a|s)$ outputs the mean and standard deviation of a learned Gaussian distribution from which the actions are sampled. On the other hand, the critic network, with parameters w, provides an estimate of the state-value function $v_w(s)$, quantifying the expected cumulative reward that can be obtained from that state under the current policy.

PPO improves upon the basic policy gradient methods by modifying how the policy is updated. Traditional policy gradient algorithms [58] directly adjust the policy parameters in the direction of the estimated gradient of expected cumulative reward. The absence of a constraint on the magnitude of this update often results in abrupt policy changes, degraded performance, or training instability. To mitigate this issue, Trust Region Policy Optimization (TRPO) [59] introduced a constraint to limit how much the new policy can diverge from the previous one, enforcing that updates occur within a so-called trust region. While TRPO provides more stable learning, it requires computationally expensive second-order optimization procedures.

3.4.1 Theoretical Formulation

PPO was designed to overcome these limitations by replacing the explicit constraint of TRPO with a clipping mechanism embedded within the objective of the policy function, which directly governs the update of the actor's parameters θ. This *clipped surrogate objective*, at timestep t, is formally expressed as:

$$J_t^{CLIP}(\theta) = \hat{\mathbb{E}}_t \left[\min \left(r_t(\theta)\hat{A}_t \ , \ \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right] \tag{3.20}$$

where the expectation $\hat{\mathbb{E}}_t[\dots]$ indicates the empirical average over a finite batch of samples collected during the rollout phase. The term $\hat{A}_t = q(s_t, a_t) - v(s_t)$ is an estimator of the *advantage function* at timestep t, which quantifies the relative benefit of executing a particular action a_t in a state s_t compared to the expected value of the state when following

the current policy. The term $r_t(\theta)$ represent the *probability ratio*, defined as:

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \tag{3.21}$$

which measure how the updated policy modifies the likelihood of selecting the same action a_t in state s_t compared to the previous policy. The clipping operation, the second term in Eqn. (3.20), constrain this ratio within the interval $[1-\epsilon,\ 1+\epsilon]$, with ϵ clip ratio, restricting the magnitude of policy update.

The parameters of the critic network w are updated through a loss function, referred to as *critic loss* or *value function loss* (L_t^{VF}), which corresponds to a squared error loss between the predicted and target state values.

$$L_t^{VF} = \left(v_w(s_t) - v_t^{\text{targ}}\right)^2 \tag{3.22}$$

The overall PPO objective function, denoted as $J_t^{PPO}(\theta, w)$, combine the clipped surrogate objective of the actor (policy network), the value function loss of the critic (value network), and an additional entropy term $S[\pi_\theta](s_t)$ that encourage policy exploration and prevents premature convergence toward deterministic behaviors. It is expressed as:

$$J_t^{PPO}(\theta, w) = \mathbb{E}_t\left[J_t^{CLIP}(\theta) - c_1 L_t^{VF}(w) + c_2 S[\pi_\theta](s_t)\right] \tag{3.23}$$

where c_1 and c_2 are scalar coefficient that weight the relative importance of the value loss and entropy terms, respectively. Using, for example, gradient ascent as the optimizer, the parameters of the two networks are updated in order to maximize J_t^{PPO}:

$$\theta^{\text{new}} = \theta^{\text{old}} + \eta_\theta \nabla_\theta J_t^{PPO}(\theta, w) \tag{3.24}$$

$$w^{\text{new}} = w^{\text{old}} + \eta_w \nabla_w J_t^{PPO}(\theta, w) \tag{3.25}$$

In the case of the actor, the update of the parameters θ is aimed at maximizing the objective function, improving the quality of the learned policy and, thanks to the entropy term, favoring the exploration of alternative actions. As for the critic, the objective includes the loss function of the value $L_t^{VF}(w)$ with a negative sign. Therefore, while formally maintaining an update for gradient ascent, the resulting effect on the parameters w is equivalent to a minimization of the value loss. In other words, the update of the critic can be interpreted as an implicit gradient descent on the error function of the estimated value.

3.4.2 PPO Training Loop

The training process of PPO alternates between two main phases, a data collection phase, during which the current policy interacts with the environment, and an optimization phase, during which the gathered data are used to update the parameters of the actor and critic networks.

1. **Data Collection (Rollout Phase)** At the beginning of each iteration, the current policy $\pi_{\theta_{\text{old}}}$ interacts with the environment for a fixed number of timesteps T storing each transition (s_t, a_t, r_t, s_{t+1}) in a dedicated *rollout buffer*. It contains, for every timesteps, the observed state s_t, the chosen action a_t, the reward r_t, the termination flag, the value estimate $v_w(s_t)$ and the log-probability $\log \pi_{\theta_{\text{old}}}$. If N_{env} environment instances, running under the same policy, are employed, the experience gathered corresponds to $N \cdot T$ transition in total. During this phase, the networks parameters remain fixed, the agent simply samples trajectory on-policy to create a consistent dataset for optimization. Once the rollout is complete, the information in the buffer is post-processed to compute the advantage estimates $\hat{A}_t$ and the corresponding value targets V_t^{targ}. In most PPO implementations, the advantage is computed through the Generalized Advantage Estimation (GAE) [60].

2. **Optimization Phase** The entire buffer, containing $N_{\text{env}} \cdot T$ transition, forms a *training batch*, which is subsequently and randomly divided into smaller *mini batches* m of size $M < N_{\text{env}} \cdot T$. During each iteration, the optimization proceeds for K epochs, multiple passes over the same batch to improve data efficiency. During each epoch for each mini-batch, the PPO objective $J_t^{PPO}(\theta, w)$ is computed and the parameters of both the actor and critic networks are updated via gradient ascent (typically using the Adam optimizer). After all epochs are complete, the parameters of the new policy π_θ are copied into $\pi_{\theta_{\text{old}}}$, the rollout buffer is cleared, and the next training iteration begins.

This entire cycle is repeated until the total number of iteration N_{iter} is reached. The complete training procedure of PPO is summarized below in Algorithm 1.

Algorithm 1 PPO Training Loop

Initialize: actor parameters θ, critic parameters w, and old policy $\pi_{\theta_{\text{old}}}$
for iteration = 1, 2, ..., N_{iter} **do**
  for actor $n = 1, 2, \ldots, N_{\text{env}}$ **do**
    Run policy $\pi_{\theta_{\text{old}}}$ in the environment for T timesteps
    Store each transition (s_t, a_t, r_t, s_{t+1}) in the rollout buffer
    Save $v_w(s_t)$ and $\log \pi_{\theta_{\text{old}}}(a_t|s_t)$ for each step
  end for
  Compute advantage estimates $\hat{A}_t$ and value targets v_t^{targ}
  for epoch = 1 to K **do**
    for each mini-batch m **do**
      Compute the PPO objective $J_t^{PPO}(\theta, w)$
      Update actor parameters: $\theta \leftarrow \theta + \eta_\theta \nabla_\theta J_t^{PPO}$
      Update critic parameters: $w \leftarrow w + \eta_w \nabla_w J_t^{PPO}$
    end for
  end for
  Copy new policy parameters: $\pi_{\theta_{\text{old}}} \leftarrow \pi_\theta$
  Clear rollout buffer
end for

Chapter 4

Deep Reinforcement Learning Framework Design

This chapter presents the DRL framework developed in this thesis to autonomously plan time-critical CAMs in Low Earth Orbit. Its objective is to translate the theoretical principle of DRL into an operational architecture capable of evaluating conjunction risk, decide and execute impulsive maneuvers under time constrains, with computational efficiency compatible with large scale training. The chapter begins with the formulation of the conjunction risk assessment process, introducing the computation of the collision probability and the key geometric quantities that characterize close approaches. Then, the CAM planning problem is formalized as a MDP, detailing the definition of the state and action spaces, the transition dynamics governed by the orbital propagator, and the reward function design balancing safety, orbital compliance, and propulsive efficiency. The following section describes the generation of a large and diverse database of artificial conjunction scenarios used for training and validation. Finally, the implementation of the complete DRL framework is presented, illustrating its modular architecture within the training loop, laying the foundation for the quantitative results discussed in the next chapter.

4.1 Conjunction Risk Assessment

Before addressing the computation of collision probability, it is necessary to define two key quantities rigorously: the *distance of closest approach* (d_{min}) and the *time of closest approach* (t_{TCA}). Let $\mathbf{r}(t)$ be the position of the active satellite and $\mathbf{r}_d(t)$ that of the debris, propagated over a time interval $t \in [t_0, t_f]$ with constant step Δt. The distance between the satellite and the debris is defined as:

$$d(t) = ||\mathbf{r}(t) - \mathbf{r}_d(t)|| \tag{4.1}$$

The minimum value of $d(t)$ identifies the distance of closest approach:

$$d_{min} = \min_{t \in [t_0, t_f]} d(t) \tag{4.2}$$

The time associated with the minimum distance is therefore the so-called time of closest approach and is defined as:

$$t_{tca} = \arg\min_{t} d(t) \tag{4.3}$$

These two quantities, $d_{\min}$ and t_{tca}, provide the geometric foundation for conjunction risk assessment. However, before addressing the actual computation of the collision probability, is essential to account for the uncertainties affecting the state of the two objects involved cause they play a critical role in estimating P_c. These uncertainties are typically modeled as zero-mean three-dimensional Gaussian distributions and are represented through covariance matrices. When focusing solely on the positional component, the covariance matrix defines a three-dimensional uncertainty ellipsoid centered at the estimated position of the object. This ellipsoid describes a region of equal probability density, characterizing the spatial dispersion of the object's likely location at a given confidence level.

4.1.1 Collision Probability

In the most general case (illustrated in Figure 4.1), it is assumed that at initial time t_0 the estimated satellite state $\mathbf{X}(t_0)$, consisting of position and velocity, is known, with associated initial covariance matrix $\mathbf{\Sigma}(t_0)$. Similarly, for the debris, known state $\mathbf{X}_d(t_0)$ and associated covariance matrix $\mathbf{\Sigma}_d(t_0)$ is assumed. The orbits and covariances are propagated to the TCA and the collision probability can be evaluated using the propagated states and covariances at TCA, i.e., $\mathbf{X}(t_{\text{tca}})$, $\mathbf{X}_d(t_{\text{tca}})$, $\mathbf{\Sigma}(t_{\text{tca}})$,$\mathbf{\Sigma}_d(t_{\text{tca}})$.

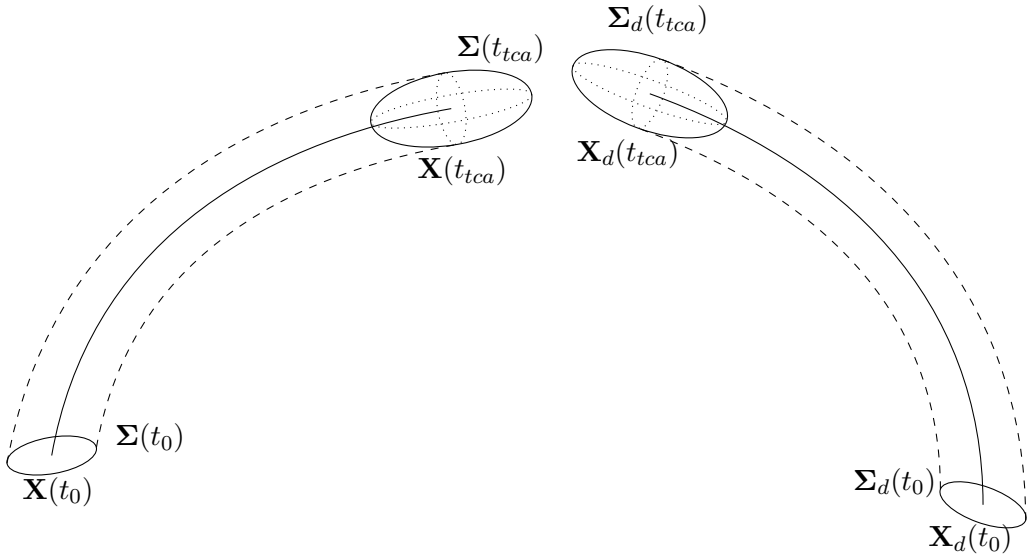


Figure 4.1: P_c Calculation Problem Description

It is evident that determining P_c between two orbiting objects is by no means trivial. It is necessary not only to know precisely the relative states and propagate them consistently to the TCA, but also to deal correctly with the associated uncertainties, which may evolve over time. In particular, estimation of the satellite covariance matrix can be obtained from data provided by the vehicle operator or from independent orbital surveillance sources, but its determination is not always straightforward. Even more complicated is the estimation

of the debris covariance matrix and state, which is often not accurately tracked and may be characterized by tumbling motion, and dynamic uncertainty. This general approach, while accurate, carries a high computational cost. In particular, in the context of this thesis, based on RL where the evaluation of P_c is performed thousands of times during agent training, it is necessary to adopt appropriate simplifications that allow the probability of collision to be estimated quickly, while maintaining a sufficient level of realism for the validity of the decision model.

The following simplifications are adopted. The *bodies* are *modeled as rigid spheres* of known radius [15], since it is difficult to know their attitude. The hard-body radius (HBR) R_s and R_d are thus defined for satellite and debris, respectively (estimation of which is described in Section 4.2).

It is then assumed that, although the trajectories are actually curved and the covariances are not constant, during the closest approach phase, the *relative motion is rectilinear* [61]. This assumption is justified for LEO, where relative velocities between objects in conjunction often exceed 5 km/s [62]. In this context, trajectories can be approximated as linear during the encounter, covariances can be considered constant, and velocity uncertainties are neglected. This implies that the 6×6 covariance matrices, which took into account uncertainties in both position and velocity, are reduced to 3×3 matrices representing only position uncertainties.

Finally assuming that the positions of the two objects are unrelated and that the probability density functions are Gaussian and zero mean, then [13] shows how *the covariances of the two objects can be combined into a single covariance*, centered at the center of the primary body, the satellite, constant over the encounter period. The resulting covariance matrix is then:

$$\mathbf{\Sigma}_{3D} = \mathbf{\Sigma} + \mathbf{\Sigma}_d \tag{4.4}$$

The probability density function (PDF) of the debris in a relative position $\mathbf{r}_d$ with respect to the satellite is defined as:

$$p(\mathbf{r}_d) = \frac{1}{\sqrt{(2\pi)\det(\mathbf{\Sigma}_{3D})}} \exp\left(-\frac{1}{2}\mathbf{r}_d^T \mathbf{\Sigma}_{3D}^{-1} \mathbf{r}_d\right) \tag{4.5}$$

The ellipsoid associated with this distribution is centered on the estimated position of the satellite and represents the region of three-dimensional uncertainty in which the debris may be found. During the encounter, the debris, with velocity equal to the relative velocity ($\mathbf{v}_{rel} = \mathbf{v}_s - \mathbf{v}_d$), crosses this ellipsoid along a straight trajectory. The region traversed by the secondary body is thus a cylindrical tube centered on the secondary (Figure 4.2). The radius of this tube is given by the sum of the rigid radii of the two bodies, i.e:

$$R = R_s + R_d \tag{4.6}$$

This value represents the *combined HBR*: it incorporates the physical dimensions of the two bodies into the probability calculation, and is used to determine whether a geometric overlap occurs.

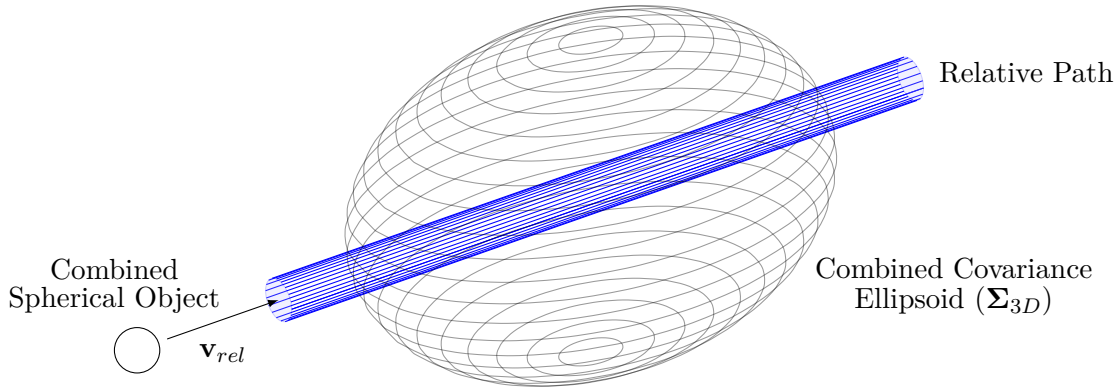


Figure 4.2: Description of 3D Encounter Geometry

The collision probability then corresponds to the integral of the PDF over the cylindrical volume:

$$P_c = \iiint_V p(\mathbf{r}_d)\, dx\, dy\, dz \tag{4.7}$$

Despite the simplifications introduced, the triple integral of Eqn. (4.7) is quite hard to compute. However, due to the assumption of rectilinear motion and high relative velocity, a further simplification can be introduced to reduce the problem from three to two dimensions. The *encounter plane* is defined as the plane orthogonal to the relative velocity at TCA. Both the combined covariance ellipsoid and the transverse section of the tube traversed by the secondary body are projected onto this plane. In this system, the dimension along the direction of relative motion can be considered separable and the integral along the direction of motion approximated to one [15]. As a result, as shown in Figure 4.3, the three-dimensional tube is reduced to a circle on the encounter plane, of radius R centered on the debris, and the projected covariance ellipsoid becomes an ellipse centered on the satellite.

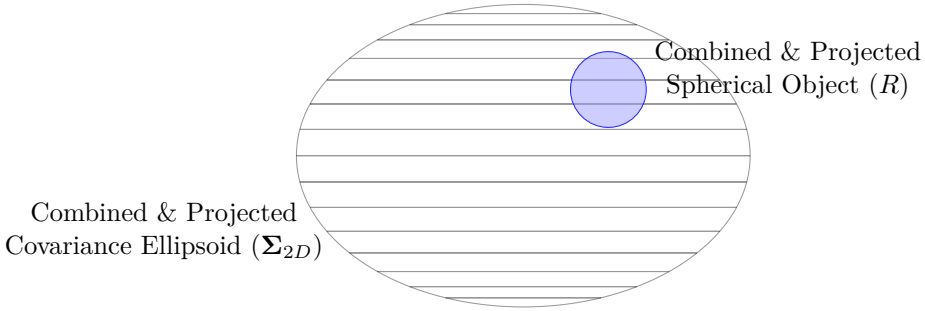


Figure 4.3: Description of 2D Encounter Geometry

The collision probability is thus reduced to a two-dimensional integral on the encounter plane:

$$P_c = \frac{1}{2\pi\sqrt{\det(\mathbf{\Sigma}_{2D})}} \int_{-R}^{R} \int_{-\sqrt{R^2-x^2}}^{\sqrt{R^2-x^2}} \exp\left(-\frac{1}{2}\boldsymbol{\rho}^\top \mathbf{\Sigma}_{2D}^{-1} \boldsymbol{\rho}\right) dy\, dx \tag{4.8}$$

where $\mathbf{\Sigma}_{2D}$ is the projected combined covariance matrix on the encounter plane, $\boldsymbol{\rho} = \mathbf{r}_{2D} - \mathbf{r}_d$, with $\mathbf{r}_{2D} = [x\ y]^\top$ and $\mathbf{r}_b$ is the relative position at TCA projected to the two-dimensional coordinates.

To be consistent with the analytical formulation discussed in the next section, the diagonalized form of the covariance matrix is adopted [15]. In this formulation, the combined covariance matrix is rotated such that its principal axes align with the local coordinate system of the encounter plane:

$$\mathbf{\Sigma}_{2D} = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix} \tag{4.9}$$

where σ_x^2 and σ_y^2 are the variances along the minor and major axes of the covariance ellipse, respectively. The coordinate system (x, y) is centered at the primary object (the satellite), with the x-axis aligned with the direction of minimum uncertainty, the ellipse minor axis, and the y-axis aligned with the ellipse major axis. The projected miss distance components (x_m, y_m) are defined in this rotated frame. The probability collision Eqn. (4.8) then becomes:

$$P_c = \frac{1}{2\pi\,\sigma_x\sigma_y} \int_{-R}^{R} \int_{-\sqrt{R^2-x^2}}^{\sqrt{R^2-x^2}} \exp\left\{ -\frac{1}{2} \left[\left(\frac{x + x_m}{\sigma_x} \right)^2 + \left(\frac{y + y_m}{\sigma_y} \right)^2 \right] \right\} dy\,dx \tag{4.10}$$

An alternative formulation can be defined by introducing the aspect ratio (AR), defined as the ratio between the standard deviations σ_y and σ_x:

$$AR = \frac{\sigma_y}{\sigma_x} \tag{4.11}$$

Eqn. (4.10) can then be rewritten as:

$$P_c = \frac{1}{2\pi\,\sigma_x^2\,AR} \int_{-R}^{R} \int_{-\sqrt{R^2-x^2}}^{\sqrt{R^2-x^2}} \exp\left\{ -\frac{1}{2} \left[\left(\frac{x + x_m}{\sigma_x} \right)^2 + \left(\frac{y + y_m}{\sigma_x\,AR} \right)^2 \right] \right\} dy\,dx \tag{4.12}$$

Despite the various simplifications introduced to reduce the dimensionality and complexity of the problem, the calculation of the collision probability still requires evaluating a two-dimensional integral over the encounter plane, Eqn. (4.12).

4.1.2 Alfano's Maximum Analytical Approximation

In computationally demanding contexts, such as those considered in this work, it becomes advantageous to replace the explicit evaluation of the integral with an accurate closed-form approximation.

To this end, the formulation proposed by *Alfano* [15] is adopted, which allows estimating a maximum probability of collision (P_{max}) through a series of closed-form approximations based on a small number of parameters: the miss distance (d_{min}), the combined HBR (R) and the aspect ratio (AR) of the projected covariance ellipse. As depicted in Figure 4.4, in the encounter plane, the miss distance vector $\mathbf{d}_{min}$ forms an angle θ with respect to the major axis of the covariance ellipse. Alfano proposes to determine the orientation of $\mathbf{d}_{min}$ that maximizes the probability of collision in a two-dimensional formulation.

Thus taking into account θ, Eqn. (4.12) becomes:

$$P_c = \frac{1}{2\pi\,\sigma_x^2\,AR}\int_{-R}^{R}\int_{-\sqrt{R^2-x^2}}^{\sqrt{R^2-x^2}} \exp\left\{-\frac{1}{2}\left[\left(\frac{x+d_{min}\sin\theta}{\sigma_x}\right)^2+\left(\frac{y+d_{min}\cos\theta}{\sigma_x\,AR}\right)^2\right]\right\} dy\,dx \tag{4.13}$$

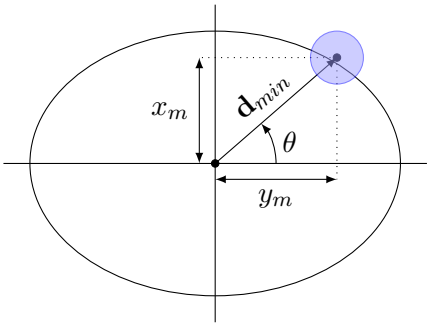


Figure 4.4: Projected Position Relative to θ Angle.

Taking the derivative of Eqn. (4.13) with respect to the angle θ and setting it equal to zero, the probability is maximized when $\theta = \pi/2$, that is, when the vector $\mathbf{d}_{min}$ is aligned with the major axis of the ellipse, thus $x_m = 0$ and $y_m = d_{min}$. Accordingly, Eqn. (4.13) is rewritten as:

$$P = \frac{\exp\left\{\left[-\frac{1}{2}\left(\frac{d_{min}^2}{\sigma_x^2 AR^2}\right)\right]\right\}}{2\pi\sigma_x^2 AR}\int_{-R}^{R}\int_{-\sqrt{R^2-x^2}}^{\sqrt{R^2-x^2}} \exp\left\{-\frac{1}{2}\left[\left(\frac{x}{\sigma_x}\right)^2+\left(\frac{y^2+2y\,d_{min}}{\sigma_x^2\,AR^2}\right)^2\right]\right\} dy\,dx \tag{4.14}$$

Since the integral in Eqn. (4.14) still does not admit a closed-form analytical solution, the exponential term is expanded in a power series, and the derivative of the resulting expression is taken with respect to σ_x, in order to determine the value of σ_x that maximizes the probability. Since no exact solution exists, suitable analytical approximations are introduced for both the optimal σ_x and the corresponding maximum probability P_{max}. Among the various formulations proposed in [15], the approximation $P_{\max,2}(\sigma_{x1})$ was chosen in the present work because it represents an effective compromise between accuracy and computational cost.

$$P_{max2} = \frac{R^2}{384[AR^5(\sigma_{x1})^6]}\exp\left\{\left[-\frac{1}{2}\left(\frac{d_{min}}{AR(\sigma_{x1})}\right)^2\right]\right\}(aa2+bb2+cc2) \tag{4.15}$$

with:

$$\sigma_{x1} = \sqrt{\frac{(AR^2+1)R^2+2\,d_{min}^2+\sqrt{[(AR^2+1)R^2]^2+4\,d_{min}^4}}{8AR^2}} \tag{4.16}$$

$$aa2 = 192\,AR^4(\sigma_{x1})^4 \tag{4.17}$$

$$bb2 = -(24\,AR^4R^2 - 24\,R^2\,AR^2)(\sigma_{x1})^2 \tag{4.18}$$

$$cc2 = (3\,AR^4+2\,AR^2+3)R^4 + 24\,d_{min}^2R^2 \tag{4.19}$$

The approximation adopted gives very accurate results, with a maximum relative error of approximately 0.31% for thresholds less than $P_c < 0{,}01$ and 0.0025% for $P_c < 0{,}001$, values fully compatible with the accuracy requirements in operational applications.
When the combined covariance matrix is not known, as in this study, it is necessary to adopt a representative value of AR based on statistical analysis. In this regard, [15] suggests the use of reference values obtained from a study of more than 26,000 simulated conjunctions. The results show that 99% of the conjunctions have an aspect ratio $AR \leq 40$. For this reason, to remain conservative and to represent the majority of conjunction scenarios analyzed in this thesis, an aspect ratio of $AR = 40$ is adopted.

4.2 Geometric and Physical Modeling of Satellite and Debris

This section describes the physical and geometric characterization of both the satellite and the debris. These parameters are required not only for the computation of the collision probability, where quantities such as the combined HBR are involved, but also for the orbital propagation phase, where non-gravitational perturbations such as atmospheric drag and solar radiation pressure must be modeled accurately. In particular, these effects depend on the object's mass and on its effective projected cross-sectional area A for atmospheric drag, as well as the area directly exposed to solar radiation $A_\odot$ for SRP modeling.

4.2.1 Hard Body Radius

In the calculation of collision probability, the estimate of the combined HBR has a crucial role because, as can be seen in Eqn. (4.15), the collision probability P_c varies approximately with $\sqrt{R}$. An overestimation of R can therefore artificially increase the value of Pc by up to an order of magnitude. It is therefore essential to avoid excessive conservatism, favoring physically and justified values [7].

Satellite Hard Body Radius - R_s

The estimation of the HBR of the primary object, R_s, can be approached in several ways, as discussed by *Mishiku et al.* [62] and in the *NASA Conjunction Assessment and Collision Avoidance Handbook* [7]. The simplest approach assumes a *fixed combined HBR*, accounting for both bodies rather than the primary alone, historically set to about 20m, however often resulting in a strong overestimation of P_c. A more refined method define R_s as the radius of the minimum *circumscribing sphere* enclosing the entire spacecraft, measured between the center of mass and the structurally most distant point. Although conservative, this approach is easily applicable and does not require the information on the satellite's attitude or projected geometry. Alternatively, a more realistic approach, that still does not require knowledge of the satellite attitude, projects the three-dimensional geometry of the spacecraft into a plane, considering the *maximum projected area*, $A_{\max}$, it can present in any orientation.

The equivalent HBR is the defined as:

$$R_s = \sqrt{\frac{A_{max}}{\pi}} \tag{4.20}$$

If the information of the satellite attitude at TCA is available, it is possible to calculate the *area* actually *projected* into the *conjunction plane*, A_{tca} and determine the HBR:

$$R_s = \sqrt{\frac{A_{\text{tca}}}{\pi}} \tag{4.21}$$

Since the determination of the satellite attitude and detailed three-dimensional geometry goes beyond the scope of this thesis, the estimation of R_s was conducted following a more data-driven approach. Since the collision probability (P_c) tends to be overestimated when, in the absence of precise information on the uncertainties in the positions of the debris and the satellite, it is calculated according to the maximum probability formulation (as in Section 4.1.2) with an AR set to represent most possible encounters without considering specific individual cases, the further overestimation of the combined HBR would lead to an unrealistic increase in (P_c), potentially generating false alarms and unnecessary avoidance maneuvers. To mitigate this effect, R_s was determined by combining the maximum projected area methodology with actual data from ESA DISCOS (Database and Information System Characterizing Objects in Space)[1] [63]. Through the *DISCOSweb API*, all active payloads (defined by `objectClass = Payload` and `active = True`) were extracted together with their maximum cross-sectional area (`xSectMax`). At the same time, the orbital parameter associated with the same objects were retrieved and filtered to ensure consistency with the operational scenario in consideration. In particular, only orbits whose semi-major axis and inclination fall within a defined range around the nominal values of the reference satellite were selected, specifically within ±100 km, and within ±10° respectively. The two data sets acquired, active payloads and orbits, were cross-referenced using the unique identifier `discosId`, selecting only active satellites belonging to the orbital band of interest. For each satellite identified, the respective HBR is calculated as in Eqn. (4.20):

$$R_{s,i} = \sqrt{\frac{\texttt{xSectMax}_i}{\pi}} \tag{4.22}$$

where $\texttt{xSectMax}_i$ represents the maximum cross-sectional area of the i-th satellite, provided by the DISCOS database. The effective single value of the satellite HBR is then defined as the sum of the mean value, $\bar{R}_s$, and one standard deviation, σ_{R_s}, of the distribution:

$$R_s = \bar{R}_s + \sigma_{R_s} \tag{4.23}$$

[1]The ESA DISCOS database serves as a single-source reference for launch information, object registration details, launch vehicle description, and spacecraft information for all trackable, unclassified object. It comprises more than 40.000 entries, including data on launch events, debris, rocket bodies, and operational spacecraft. The maintenance and operation of the database through the dedicated APIs is formally acknowledge.

Debris hard body radius - R_d

The characterization of debris in inherently more complex than that of active satellites. Comprehensive catalogs such as ESA DISCOS mainly cover trackable object, and even those debris that can be tracked often lack reliable geometric or physical information. Therefore, the debris HBR, R_d, is determined through statistical analysis such as the one proposed by *Baars and Hall* [64]. Based on the processing of radar data from the Space Fence system, the size distribution of unknown secondary objects is quantified by analyzing approximately 230000 "known-on-unknown" conjunctions. The results show that approximately 97.5% of the secondary objects have a radius of less than 0.36 m, which is therefore adopted in this thesis as the representative value for R_d.

4.2.2 Cross-Sectional Area

The cross-sectional area is a key parameter in the orbital propagator, as it directly influences the modeling of the non-gravitational perturbations. Determining the exact projected area in the velocity direction for atmospheric drag and the area directly exposed to solar radiation for srp would require continuous knowledge of the satellite and debris attitude, which, as previously mentioned, lies beyond the scope of this thesis. For this reason, a simplified assumption is adopted: the same effective cross-sectional area is used for both perturbations, defining A_s and A_d for the satellite and debris, respectively.

The *satellite cross-sectional area* is obtained via the DISCOSWeb API, similarly to the estimation of R_s, by selecting active payloads, applying the same orbital filtering, and collecting the average cross-sectional area (`xSectAvg`) for each selected satellite. The resulting sample is then analyzed statistically and a single representative value of A_s is defined as:

$$A_s = \bar{A}_s + \sigma_{A_s} \tag{4.24}$$

For the estimation of the *debris cross-sectional area*, the object is approximated as a sphere, a common assumption given its typical irregular shape and chaotic tumbling motion. The area A_d is therefore computed from the previously defined fixed HBR as the projection into a plane of the sphere as:

$$A_d = \pi R_d^2 \tag{4.25}$$

4.2.3 Mass

Finally, the last parameter required to propagate both satellite and debris while accounting for non-gravitational perturbation is the mass. For the *satellite mass*, m_s, the value is obtained through the same DISCOSWeb API procedure retrieving the `mass` parameter of active payload using the same orbital filtering. The estimation of *debris mass*, m_d, which is considerably more uncertain to determine, is based on the statistical analysis presented by *Smith et al* [65]. Based on their findings, a constant representative area-to-mass ratio of $(A/M)_d = 0.1\ m^2/\text{kg}$ is adopted, determining consequently m_d from its cross-sectional

area A_d as:

$$m_d = \frac{A_d}{(A/M)_d} \tag{4.26}$$

Having established the physical and geometrical characterization of both the spacecraft and the debris, and defined all parameters required for conjunction risk assessment and orbital propagation, the set of quantities governing collision risk scenarios has been fully determined. The next step is to move from the quantitative definition of these parameters to the formulation of an autonomous decision process capable of using them to evaluate conjunction risk and plan effective collision avoidance maneuvers within the available time-critical windows.

4.3 Markov Decision Process Formulation

The problem of planning time-critical CAMs can be naturally formulated as a MDP, in which an agent interacts with a simulated environment, represented in this work by the high precision orbital propagator introduced in Section 2.4, to autonomously learn effective avoidance strategies against potential collision with a single debris in LEO. The MDP framework provide a rigorous representation of the decision making process in terms of *states, actions, transition dynamics, and rewards*, and therefore represents an ideal formalism for describing the CAM planning problem addressed in this thesis. Within this formulation the agent attempts to to determine a policy that maps the orbital configuration of the spacecraft to an appropriate control action, aiming to minimize the collision risk while keeping the spacecraft within its operational orbital limits and optimizing the available propulsive budget. Although in many spacecraft operations sensor noise or orbital uncertainties may render the system partially observable, in this work is assumed that the system is fully observable, the observation available to the agent coincides with the real environment state. The problem is therefore formalized as a *fully observable MDP*.

4.3.1 State Space

The definition of the state space directly determines how the agent perceives and interacts with the environment. To satisfy the Markov property, the state s_t must contain all the information necessary to predict the future evolution of the system. It should provide the agent with sufficient and relevant information to interact effectively with the environment dynamics while avoiding unnecessary complexity. This reflects a balance between *completeness*, the ability to encode all relevant environmental information, and *complexity*, which affects both computational cost and training stability [52].

Based on these considerations, and noting that in the context of CAM planning the state should capture variables that reflect the perceived hazard level of the conjunction [34], the state at timestep t is defined as[2]:

$$s_t = (\mathbf{r}_t \ , \ \mathbf{v}_t \ , \ \mathbf{r}_{\text{rel},t} \ , \ \mathbf{v}_{\text{rel},t} \ , \ \Delta v_{\text{frac},t} \ , \ \tau_{tca,t} \ , \ d_{\min,t} \ , \ P_{c,t}) \tag{4.27}$$

[2]Although expressed as a tuple for clarity, this representation is ultimately handled as a numerical vector in the training frameworks, $s_t \in \mathbb{R}^{16}$

where $\mathbf{r}_t \in \mathbb{R}^3$ and $\mathbf{v}_t \in \mathbb{R}^3$ denote the spacecraft position and velocity in the ECI RF. The vectors $\mathbf{r}_{rel,t} = \mathbf{r}_t - \mathbf{r}_{d,t} \in \mathbb{R}^3$ and $\mathbf{v}_{rel,t} = \mathbf{v}_t - \mathbf{v}_{d,t} \in \mathbb{R}^3$ represent, respectively, the relative position and velocity of the spacecraft with respect to the debris, expressed in the same ECI RF. The scaler $\Delta v_{\text{frac},t} = 1 - \Delta v_{used}/\Delta v_{max}$ represent the remaining fraction of the total Δv budget, Δv_{max}. The variable $\tau_{tca,t} = t_{tca,t} - t_{curr}$ denotes the remaining time to the predicted TCA at current step t, while $d_{min.t}$ e $P_{c,t}$ corresponding to the estimated minimum distance and collision probability always at the current step t. These variables, which play a fundamental role in conjunction risk assessment, are introduced and derived in detail in a dedicated section later in this chapter.

4.3.2 Action Space

The definition of the action space must satisfy two properties: completeness and validity [66]. Completeness ensures that the set of possible actions allows the agent to reach the desired objective, missing any essential control input could prevent convergence toward an optimal policy. Validity, on the other hand, requires that all actions within the action space are physically feasible and consistent with the operational constraint of the system.

Although electric propulsion is increasingly adopted in LEO missions, their low continuous thrust levels and long actuation times makes them unsuitable for time-critical CAM planning. Given the short-notice nature of conjunction scenarios, only impulsive maneuvers are considered. Consequently, the actions a_t available to the agent correspond to instantaneous changes in the spacecraft velocity components along the three axis of ECI RF, ensuring that the completeness property is satisfied [3]:

$$a_t = (\Delta v_{x,t}, \Delta v_{y,t}, \Delta v_{z,t}) \tag{4.28}$$

Since the actions are selected by the policy network, their raw value are constrained by the activation function applied to the actor network's output layer. In this work, a hyperbolic tangent (tanh) activation is used, which bounds the action components within the normalized interval $[-1, 1]$. The actions are subsequently mapped into meaningful velocity increments in the range $[-\Delta v_{\text{max,t}}, +\Delta v_{\text{max,t}}]$ representing the maximum admissible Δv along each inertial axis at each step t . Moreover, a minimum Δv threshold, Δv_{thr}, is imposed to emulate the finite resolution of impulsive thrusters and prevents the generation of unrealistically small maneuvers that would be infeasible in real spacecraft operations. If the computed Δv falls below Δv_{thr}, the action is set to zero, satisfying the validity property.

4.3.3 Transition Dynamics

Once the agent selects an action a_t according to the policy distribution generated by the actor network, the environment applies this control input to update its internal physical state, triggering the transition from the current state s_t to the next state s_{t+1}. This

[3] As for the state representation, the tuple is ultimately handled as a numerical vector within the training framework, $a_t \in \mathbb{R}^3$

process establishes the connection between the decision-making logic of the agent and the dynamical evolution of the simulated orbital environment.

At each decision step t, the environment receives the action vector a_t selected by the agent, representing the instantaneous velocity change along the three inertial axes of the ECI RF. The propulsive budget is subsequently updated:

$$\Delta v_{\text{used}} \leftarrow \Delta v_{\text{used}} + \Delta v_t \tag{4.29}$$

with $\Delta v_t = \sqrt{\Delta v_{x,t}^2 + \Delta v_{y,t}^2 + \Delta v_{z,t}^2}$ being the magnitude of the applied impulse. The impulsive maneuver is then applied as an instantaneous change in the spacecraft velocity modifying its state vector as:

$$\mathbf{v}_{t^+} = \mathbf{v}_t + \Delta \mathbf{v}_t \qquad , \qquad \mathbf{X}_{t^+} = \begin{bmatrix} \mathbf{r}_t \\ \mathbf{v}_{t^+} \end{bmatrix} \tag{4.30}$$

where $\mathbf{X}_{t^+}$ defines the post-maneuver initial condition for the subsequent orbital propagation. The custom orbital propagator introduced in Section 2.4 is then employed to integrate the spacecraft equation of motion from $t \cdot \Delta t$ to $(t+1) \cdot \Delta t$, over the decision timestep Δt. Depending on the required simulation fidelity, the propagation is performed under a simple two-body problem or by progressively including selected perturbations.

Once the propagation is complete, a conjunction risk assessment is performed over a fine temporal grid of 1s, spanning from the current time, $(t+1) \cdot \Delta t$, to the predicted $t_{tca,t}$, in order to evaluate the post-maneuver parameters such as the predicted time of closest approach ($t_{tca,t+1}$), the minimum relative distance between the spacecraft and the debris ($d_{min,t+1}$), and the corresponding collision probability ($P_{c,t+1}$). These quantities quantify the effect of the agent's action on the conjunction geometry and directly determine the scaler reward r_{t+1} assigned to the executed action, as discussed in Section 4.3.4.

Finally the environment updates the new state:

$$s_{t+1} = (\mathbf{r}_{t+1} \ , \ \mathbf{v}_{t+1} \ , \ \mathbf{r}_{\text{rel},t+1} \ , \ \mathbf{v}_{\text{rel},t+1} \ , \ \Delta v_{\text{frac},t+1} \ , \ \tau_{tca,t+1} \ , \ d_{\text{min},t+1} \ , \ P_{c,t+1}) \tag{4.31}$$

and returns it to the agent, closing the interaction loop. This iterative process continues until a termination condition is met as seen in Section 4.3.5.

4.3.4 Reward Function Design

The definition of the reward function represents one of the most critical components of a RL framework. Its design requires a deep understanding of the environment dynamics and optimization objectives to determine which actions should be encouraged or penalized. The reward is a scalar signal ($r_t \in \mathbb{R}$) that can be dense or sparse [52]. Sparse reward delivers nonzero values only at the end of an episode, when a termination condition is met, while dense reward provides continuous feedback, positive, negative or zero, at every step. Although sparse formulation is simpler to define, they often result in poor sample efficiency since the agent receives limited information about which intermediate actions contribute to success or failure. Dense reward, on the other hand, enables faster learning

but requires careful tuning to avoid biasing the agent toward short-term goals. In both cases, the magnitude and scaling of the reward must be carefully adjusted; if the numerical scales of the individual components differ significantly, the training process can become unstable, causing the agent to prioritize some objective over others.

In this work, a hybrid reward strategy was adopted, the agent receives dense reward throughout an episode, providing intermediate feedback on its performance, and an additional terminal reward is assigned whenever a termination event occurs, either corresponding to a success or a failure. Following a common approach in the literature [67, 68], all dense rewards are defined to be non-positive at every step, assuming negative values to penalize undesired behavior and becoming zero only when the agent acts according to the defined objective. A positive terminal reward is provided only when the episode terminates under a *success* condition. Since the goal of this thesis is to train a DRL agent capable of performing CAMs by balancing multiple and potentially conflicting objectives, the dense reward follows the typical weighted-sum formulation, in which each term corresponds to a specific performance goal: minimizing collision probability, maintaining the operational orbit within predefined limits, and optimize the consumption of the available Δv budget. The overall reward can thus be expressed as:

$$r_{\text{tot}} = c_{P_c} \cdot r_{P_c} + c_{\Delta\text{COE}} \cdot r_{\Delta\text{COE}} + c_{\Delta v} \cdot r_{\Delta v} + r_{\text{terminal}} \tag{4.32}$$

where c_{P_c}, $c_{\Delta\text{COE}}$, and $c_{\Delta v}$ are reward weighting coefficients that determine the relative importance of each component. The definition and purpose of each reward term in Eqn. (4.32) are described in the following subsections.

Collision Probability Term - r_{P_c}

As discussed in Section 1.2.2, according to [7], CAMs are considered successful when the post-maneuver collision probability is reduced by at least 1.5 order of magnitude below the nominal threshold of 10^{-4}. In line with this standard, a target collision probability value, here referred to as $P_{c,goal} = 3 \cdot 10^{-6}$ is defined as the reference level the agent must achieve to ensure a safe post-maneuver configuration. The reward associated with the collision probability is then designed to encourage the agent to minimize P_c toward this goal value. Drawing inspiration from *Mu et al.* [34], the corresponding reward term r_{P_c} is defined as:

$$r_{P_c} = \begin{cases} 0, & P_c \leq P_{c,\text{goal}}, \\ -\dfrac{1}{2}\left[1 - \dfrac{\log_{10}(P_{c,t}/P_{c,\text{goal}})}{\log_{10}(P_{c,\text{goal}})}\right], & P_{c,\text{goal}} < P_c \leq 1 \end{cases} \tag{4.33}$$

This formulation produces a reward that penalize configurations with high collision probability while saturating at zero once the target value is achieved. The logarithmic scaling ensures a smoother reward gradient in the range of interest, allowing the agent to receive meaningful feedback even when P_c varies by several orders of magnitude.

Figure 4.5 illustrates the resulting reward function. The curve remains flat (zero reward) for $P_c \leq P_{c,goal}$, representing a safe post-maneuver condition, and decreases monotonically as the probability of collision increases.

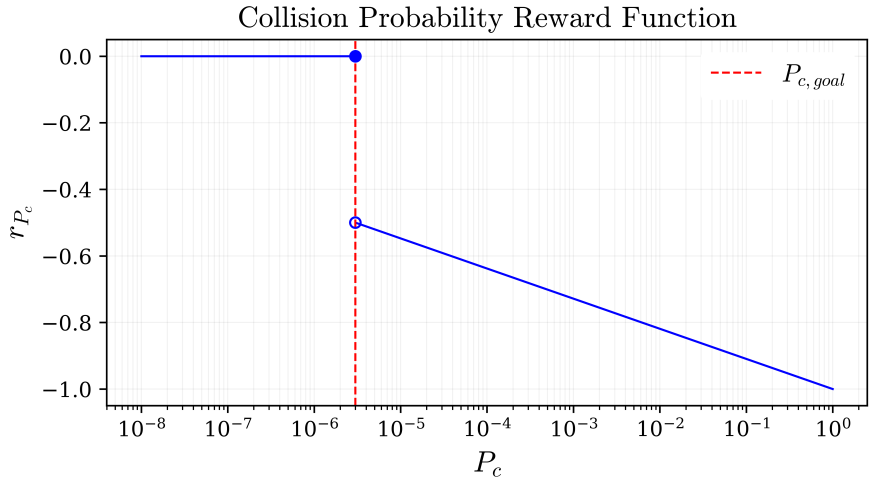


Figure 4.5: Collision Probability Reward Function

Orbit Deviation Term - $r_{\Delta\mathbf{COE}}$

In addition to minimizing the collision probability, the CAM planning problem addressed in this work aims to ensure that the satellite post-maneuver orbit remains within its operational limits. Consequently, the reward function is designed to discourage excessive orbital deviations caused by avoidance impulses. Unlike the formulation proposed by *Kazemi et al.* [69], where distinct reward functions are defined for each orbital element to compensate for their not uniform order of magnitude, in this work a normalized formulation is adopted. The goal is to ensure that each orbital element contributes equally to the overall reward, they are therefore scaled by their own acceptable deviation threshold τ_{COE}. The resulting normalized deviation ρ is then:

$$\rho = \frac{|\Delta\text{COE}|}{\tau_{\text{COE}}}, \tag{4.34}$$

where $\Delta\text{COE} = \text{COE}_t - \text{COE}_{ref,t}$, represents the deviation of each orbital element from the reference orbit propagated over the same time window without any applied maneuver. To ensure a smooth penalization of orbital deviation, a Huber-like function $\phi(\rho)$ was defined as follows:

$$\phi(\rho) = \begin{cases} 0, & \text{if } \rho \leq 1, \\ \dfrac{1}{2}(\rho - 1)^2, & \text{if } 1 < \rho \leq 1 + \delta, \\ \delta\left[(\rho - 1) - \dfrac{1}{2}\delta\right], & \text{if } \rho > 1 + \delta, \end{cases} \tag{4.35}$$

with $\delta = 0.5$ defines the extent of the quadratic transition region. The use of this Huber-like formulation alternates between a quadratic region, which gently penalizes small deviation

beyond the threshold, with a linear region that prevents excessive divergence for large deviations, improving learning stability. The reward contribution associated with each orbital element is expressed as:

$$r_{\Delta\text{COE}_i} = -\phi(\rho_i) \tag{4.36}$$

where ρ_i represents the normalized deviation of the i-th orbital element from its nominal value. The overall reward accounting for the deviation of all monitored orbital elements (namely $a,\ e,\ i,\ \Omega,\ \omega$) is then computed as the sum of the individual components:

$$r_{\Delta\text{COE}} = \sum_i r_{\Delta\text{COE}_i} \tag{4.37}$$

An example of the proposed formulation is illustrated in Figure 4.6, which shows the reward function associated with the eccentricity deviation with a threshold $\tau_e = 0.001$. For deviation smaller than τ_e, no reward is applied, as $|\Delta e|$ exceeds the threshold, the reward gradually decreases according to the quadratic segment until $\rho = 1 + \delta$, beyond which the function follows a linear trend.

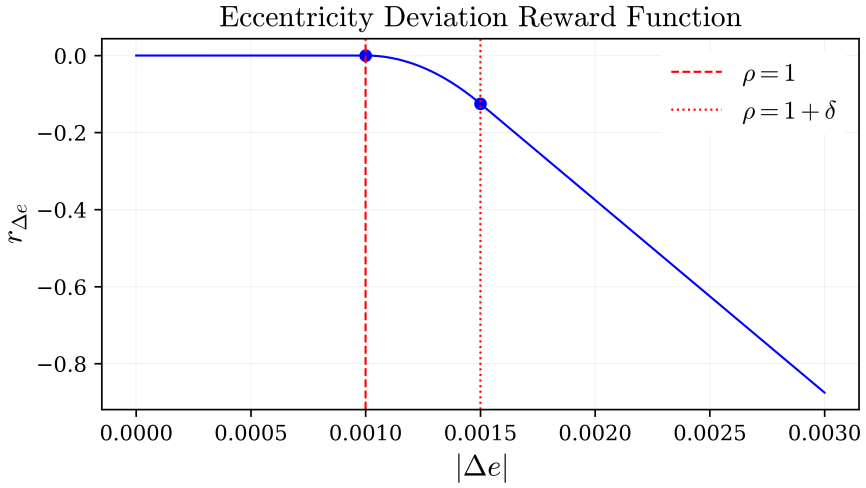


Figure 4.6: Eccentricity Deviation Reward Function

Propulsion Budget Term - $r_{\Delta v}$

The propulsion budget term represents the component of the reward function directly linked to the action selected by the agent at each step. It is defined as:

$$r_{\Delta v} = -\frac{\Delta v_t}{\sqrt{3} \cdot \Delta v_{\max,t}} \tag{4.38}$$

with $\Delta v_t = \sqrt{\Delta v_x^2 + \Delta v_y^2 + \Delta v_z^2}$ magnitude of the action selected at step t. The normalization factor $\sqrt{3}$ ensures consistency across the three action directions, such that the maximum combined impulse within the admissible actin space corresponds to a penalty of unit magnitude and keeps $r_{\Delta v}$ bounded within $[-1, 0]$.

Differently from the other reward terms, where the weighting coefficient remain constant throughout an episode, the coefficient associated with the propellant consumption term $c_{\Delta v}$ is dynamically increased. When the primary objective is already satisfied, $P_c \leq P_{c,goal}$ and COEs within threshold, the increased weight of $r_{\Delta v}$ effectively discourage the agent from performing unnecessary maneuvers. Furthermore, to reinforce this behavior and promote a "no-action" policy, when r_{P_c}, $r_{\Delta COE}$, $r_{\Delta v} = 0$, indicating that all primary goals have been achieved and the agent has correctly chosen to apply no control action, a step reward bonus, $r_{bonus,t} = +2$ is applied.

4.3.5 Termination Conditions and Terminal Reward

The termination conditions for each training episode are designed to represent realistic operational outcomes in the proposed CAM scenario, accounting for both successful and failure cases. Each termination event is associated with a terminal reward r_{terminal}, which provides the agent with with a final evaluation of its overall episodic performance. The episode can terminate under two main categories of events:

1. **Propulsion budget violation**
   The episode terminates when the cumulative Δv used, Δv_{used} exceed the maximum available budget, Δv_{MAX}, i.e., when $\Delta v_{\text{frac},t} = 0$. This event, corresponding to the exhaustion of the available propulsive capability, is penalized with a large negative terminal reward to discourage fuel inefficient behaviors:

$$r_{\text{terminal}} = -20 \quad \text{if} \quad \Delta v_{\text{frac},t} = 0 \tag{4.39}$$

2. **Time of Closest Approach reached**
   When the predicted TCA $t_{tca.t}$ falls between two consecutive time steps, the orbital propagation within the current step is interrupted and the integration is stopped at TCA, terminating the episode. In this case, the final step may be shorter than the nominal time interval. At this point, the termination outcome depends on the relative miss distance $d_{\text{min},tca}$ and on the post-maneuvers collision probability $P_{c,\text{tca}}$:

   - ***Collision***: if $d_{\text{min}} \leq$ HBR the event is classified as a physical collision and a negative reward is assigned:

$$r_{\text{terminal}} = -20 \tag{4.40}$$

   - ***Successful Mitigation***: if $P_{c,\text{tca}}$ satisfy the safety condition $P_{c,\text{tca}} \leq P_{c,goal}$ and the spacecraft orbit at TCA, is within is thresholds, i.e. $\rho_{tca} \leq 1$ the episode is considered successful. The agent is then rewarded with a large positive reward:

$$r_{\text{terminal}} = +25 \tag{4.41}$$

- ***Failed Mitigation***: if no collision occurs but $P_{c,\text{tca}} > P_{c,goal}$ and/or $\rho_{tca} > 1$ the episode is classified as an unsuccessful mitigation attempt and terminates with a negative terminal reward to indicate a suboptimal outcome:

$$r_{\text{terminal}} = -20 \tag{4.42}$$

These termination outcomes can be summarized as:

$$r_{\text{terminal}} = \begin{cases} +25, & \text{if } P_{c,\text{tca}} \leq P_{c,goal} \text{ and } COE_{tca} \text{ within threshold} \\ -20, & \text{if mitigation failed or collision occurred} \\ -20, & \text{if } \Delta v_{\text{frac},t} = 0 \end{cases} \tag{4.43}$$

The magnitude of the terminal reward is intentionally set higher than that of the dense reward, ensuring that the final outcome of the episode dominates the cumulative reward signal. In this formulation, the terminal reward acts as a global performance indicator, reinforcing successful CAM strategies and guiding the learning process toward safe, efficient and mission compliant maneuver planning.

4.4 Conjunction Scenario

In order to train and validate the DRL agent to plan collision avoidance maneuvers effectively and generalize them to any operating condition, a large number of conjunction scenarios are required. However, the scarcity of real conjunction data, combined with their limited heterogeneity and accessibility, makes the exclusive use of observational datasets insufficient for statistically significant training. In literature, this problem has been addressed with different approaches: through generative probabilistic models, which simulate the conjunction assessment process and produce synthetic CDM series consistent with real orbital populations and with the uncertainties of radar and optical observation [70], simulated datasets for all-vs-all conjunction screening in which the orbits of a large catalog of objects are propagated and combined to generate a massive set of synthetic close approaches, which are then used to train and test machine learning algorithms capable of automatically recognizing pairs of objects at risk of collision [71].

In this study a different approach is adopted. Real data are not used to generate close approach scenarios, but rather the logic of randomized scenario generators introduced by *Gremyachikh et al.* [72] is followed. This methodology is integrated with the back-propagation procedure from TCA and with the stochastic perturbation of position and velocity conditions proposed by *Bourriez et al.* [73]. From a RL perspective, this process can be interpreted as a form of *domain randomization*, designed to expose the agent to a wide variety of encounter geometries, thereby improving its ability to generalize beyond the specific training scenarios. The database generation phases are described below.

4.4.1 Satellite Initialization

The spacecraft is initialized at the reference time t_0 using a set of the six classical orbital elements $\{a,\ e \in [0,1),\ i,\ \Omega,\ \omega$ and $\theta\}$. The values of these parameters are chosen in order to place the spacecraft within the orbital range of interest for the study conducted in this thesis, i.e. LEO. Subsequently, the orbital elements are converted into the corresponding cartesian state vector in ECI RF, expressed as:

$$\mathbf{X}(t_0) = \begin{bmatrix} \mathbf{r}(t_0) \\ \mathbf{v}(t_0) \end{bmatrix} \tag{4.44}$$

which constitutes the initial condition for the numerical integration of the equations of motion. After defining the semi-major axis and the inclination, the HBR (R_s), the cross-sectional area (A_s) and the mass (m_s) of the satellite are determined through the procedure described in Section 4.2 by interacting with the *DISCOSweb API*.

4.4.2 Close Approach Database Generator

Once the satellite's orbital status has been defined, a set of N_{CA} close approach scenarios between the satellite and the debris is generated. This process allows the generation of a database of potential collision events to be used for training and validating the DRL agent. The methodological procedure adopted for their generation follows 6 steps detailed below.

1. **TCA sampling** For each close approach scenario, a nominal TCA is defined, sampled from a uniform distribution within a predefined time window:

$$t_{tca} \sim \mathcal{U}(t_0 + t_{min}, t_0 + t_{max}) \text{ s} \tag{4.45}$$

   The use of a uniform distribution ensures equal probability of extraction for all instances included in the interval considered, avoiding the introduction of temporal biases in the generation of conjunctions.

2. **Spacecraft propagation** The state of the satellite, defined by $\mathbf{X}(t_0)$, is propagated from the reference epoch t_0 to the selected conjunction time t_{tca}, by numerically integrating its equations of motion. The satellite state vector at TCA, $\mathbf{X}(t_{tca})$, is then stored and used as a reference condition for the subsequent generation of debris.

3. **Perturbation of debris position at TCA** To create realistic close approach scenarios, the position of the debris at TCA is obtained by introducing a Gaussian perturbation to the position of the satellite at the same moment in time:

$$\mathbf{r}_d(t_{tca}) = \mathbf{r}(t_{tca}) + \mathcal{N}(0, \sigma_{\text{pos}}\mathbf{I}_3) \tag{4.46}$$

   where the standard deviation σ_{pos} is itself randomly drawn from a uniform interval. This double randomization, uniform on the standard deviation and normal on the actual position, allows to captures both the variability between different scenarios and the stochastic uncertainty in the relative geometry of the encounter.

4. **Perturbation of debris velocity at TCA** To complete the debris state definition at TCA, a rotation is applied to the spacecraft velocity vector to generate different encounter geometries and relative approach directions. The rotation is performed about the direction orthogonal to the local orbital plane, defined by the cross product between the debris position and the spacecraft velocity at the time of closest approach:

$$\hat{\mathbf{w}} = \frac{\mathbf{r}_d(t_{TCA}) \times \mathbf{v}_s(t_{TCA})}{|\mathbf{r}_d(t_{TCA}) \times \mathbf{v}_s(t_{TCA})|} \tag{4.47}$$

The debris velocity is then obtained by rotating the spacecraft velocity vector by a random angle α, uniformly sampled within $[-\alpha_{max}, -\alpha_{min}] \cup [\alpha_{min}, \alpha_{max}]$ intervals, around the unit vector $\hat{\mathbf{w}}$. The transformation is expressed through Rodrigues rotation formula, which provides a general representation of a finite rotation of a vector around an arbitrary axis:

$$\mathbf{R}_{\hat{\mathbf{w}}}(\alpha) = \mathbf{I}\cos\alpha + (1 - \cos\alpha)\hat{\mathbf{w}}\hat{\mathbf{w}}^T + [\hat{\mathbf{w}}]_\times \sin\alpha \tag{4.48}$$

where $[\hat{\mathbf{w}}]_\times$ is the skew-symmetric matrix associated with the cross product by $\hat{\mathbf{w}}$. Accordingly, the rotated debris velocity is given by:

$$\mathbf{v}_d(t_{TCA}) = \mathbf{R}_{\hat{\mathbf{w}}}(\alpha)\mathbf{v}_s(t_{TCA}) \tag{4.49}$$

Renaming temporarily $v_d(t_{TCA})$ as $\mathbf{v}'$ for simplicity and $\mathbf{v}_s(t_{TCA})$ as $\mathbf{v}$, expanding Eqn. (4.49) yields:

$$\mathbf{v}' = \mathbf{v}\cos\alpha + (\hat{\mathbf{w}} \times \mathbf{v})\sin\alpha + \hat{\mathbf{w}}(\hat{\mathbf{w}} \cdot \mathbf{v})(1 - \cos\alpha). \tag{4.50}$$

In this specific case, the rotation axis $\hat{\mathbf{w}}$ is constructed to be orthogonal to the spacecraft velocity vector, the scalar product $(\hat{\mathbf{w}} \cdot \mathbf{v})$ becomes zero, and the last term of Eqn. (4.50) vanishes. The resulting simplified form is therefore:

$$\mathbf{v}_d(t_{TCA}) = \mathbf{v}_s(t_{TCA})\cos\alpha + (\hat{\mathbf{w}} \times \mathbf{v}_s(t_{TCA}))\sin\alpha \tag{4.51}$$

This expression corresponds to a planar rotation of the spacecraft velocity, it preserves the magnitude of the velocity while altering its direction, thereby generating distinct encounter geometries and varying relative approach angles between the spacecraft and the debris. Finally, to emulate modeling uncertainties and observation noise, the magnitude of the debris velocity is stochastically perturbed by multiplying it by a random scaling factor s drawn from a normal distribution centered at one:

$$\mathbf{v}_{\text{DEB}}^{\text{TCA}} \leftarrow s\mathbf{v}_{\text{DEB}}^{\text{TCA}}, \qquad s \sim \mathcal{N}(1, \sigma_{\text{vel}}) \tag{4.52}$$

where σ_{vel} is the standard deviation.

5. **Debris backpropagation**. At this point the state vector of the debris at the time of close approach is known:

$$\mathbf{X}_d(t_{tca}) = \begin{bmatrix} \mathbf{r}_d(t_{tca}) \\ \mathbf{v}_d(t_{tca}) \end{bmatrix} \tag{4.53}$$

and, after defining its geometric and physical characteristics (R_d, A_d, and m_d) as described in Section 4.2, it is propagated backward from t_{TCA} to the initial time t_0 obtain the corresponding initial state:

$$\mathbf{X}_d(t_0) = \begin{bmatrix} \mathbf{r}_d(t_0) \\ \mathbf{v}_d(t_0) \end{bmatrix} \tag{4.54}$$

6. **Conjunction risk assessment & Database generation** Having determined the initial states of the satellite and debris, together with their respective geometric and physical characteristics, it is now possible to perform a conjunction risk assessment. Both bodies are propagated forward in time by numerically integrating the equations of motion, using a fine, common time grid to ensure uniformity in the comparison of trajectories. From this propagation, t_{tca}, d_{min}, and P_c are obtained. For each successfully generated scenario, the initial state vector of the debris, the conjunction time, the minimum distance, and the corresponding collision probability are then stored in a dedicated database file. This archive constitutes the Collision Scenarios Database used in the subsequent training and validation phases of the DRL agent. An example of the resulting database structure, is reported below in Table 4.1.

DEB	$\mathbf{r}_d(t_0)$ [km]	$\mathbf{v}_d(t_0)$ [km/s]	t_{tca} [s]	$d_{\min}$ [km]	P_c [-]
001	$[x_1, y_1, z_1]$	$[v_{x_1}, v_{y_1}, v_{z_1}]$	t_{tca_1}	$d_{\min_1}$	P_{c_1}
002	$[x_2, y_2, z_2]$	$[v_{x_2}, v_{y_2}, v_{z_2}]$	t_{tca_2}	$d_{\min_2}$	P_{c_2}
⋮	⋮	⋮	⋮	⋮	⋮
N_{CA}	$[\cdots]$	$[\cdots]$	$\cdots$	$\cdots$	$\cdots$

Table 4.1: Close Approach Database Structure

All the propagation phases described above, both forward and backward in time, are performed using the custom orbital propagator developed in Section 2.4. The modular implementation of the propagator allows the generation of different databases depending on the perturbation included in the dynamic model, enable the creation of distinct datasets for the development, testing, and final deployment phases of the DRL framework.

4.5 DRL Framework Architecture

After analyzing in detail all the fundamental components that contribute to the definition of a framework for the autonomous planning of CAM in time-critical scenarios, from the characterization of the metrics used for the conjunction risk assessment to the formulation of the problem as an MDP, it is now possible to integrate these elements into a consistent architecture. This section presents the overall structure of the training framework developed in this work, describing its logical organization and the interaction flows between the main modules, as schematically illustrated below in Figure 4.7.

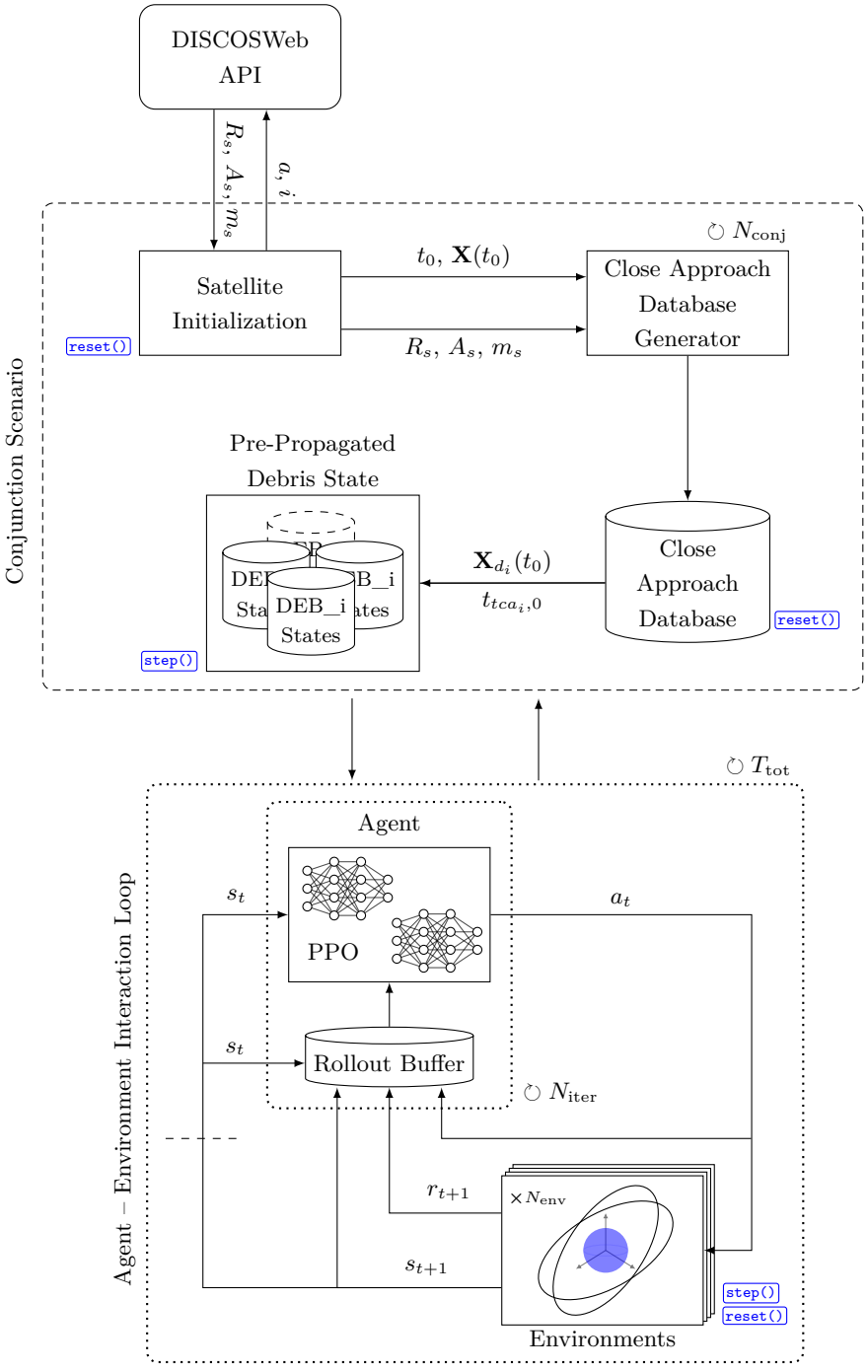


Figure 4.7: DRL Framework Architecture

4.5.1 Conjunction Scenario Module

This module is responsible for preparing the initial orbital conditions of the satellite and generating realistic conjunction scenarios used throughout the training and evaluation process. Its structure reflects the methodology described in Section 4.4 and is implemented to seamlessly interface with the other components of the framework. Once the close approach database has been generated, all the debris contained are propagated from their initial state to the corresponding TCA defined in the database ($t_{tca_i,0}$) on a fine time grid of 1s. The resulting trajectories $\mathbf{X}_{d_i}(t)$ are then stored to be easily accessible during training. This process significantly reduces the computational cost, avoiding the need to propagate debris trajectories at each step of the environment, and ensures that the state of the satellite and that of the debris are always compared on the same time grid during the conjunction risk assessment phase.

4.5.2 Agent - Environment Interaction Module

The interaction between the agent and the environment in this framework follows a hybrid temporal organization, combining an episodic structure, in which each episode corresponds to an individual conjunctions scenario, with a step-based interaction scheme adopted during PPO training.

Episodic Level

As illustrated in Figure 4.7, the tags `reset()` and `step()` identify the framework components that are involved at the episodic level during training. Each episode represents an individual conjunction scenario between the spacecraft and a selected debris object, within which the agent interacts with the environment through successive simulation steps. At the beginning of each episode, the environment is reset to its initial conditions. The satellite is restored to the nominal orbital state defined in the *Satellite Initialization* module, while a debris is randomly selected from the *Close Approach Database*. Based on the information contained in the database and on the satellite's state, the initial state observed by the agent s_{t_0} is constructed:

$$s_{t_0} = (\mathbf{r}_{t_0}\ ,\ \mathbf{v}_{t_0}\ ,\ \mathbf{r}_{\mathrm{rel},t_0}\ ,\ \mathbf{v}_{\mathrm{rel},t_0}\ ,\ \Delta v_{\mathrm{frac},t_0}\ ,\ \tau_{tca,t_0}\ ,\ d_{\mathrm{min},t_0}\ ,\ P_{c,t_0}) \tag{4.55}$$

At each step, the agent selects an action a_t, corresponding to an impulsive Δv. The updated spacecraft state is then propagated over the decision timestep Δt. To evaluate the effect of the maneuver executed at step t, a conjunction risk assessment is performed, as described in Section 4.3.3, by cross-matching the propagated satellite trajectory with the pre-computed debris states associated with the current episode. This allows the framework to compute the post-maneuver conjunction parameters $(t_{\mathrm{tca},t+1}, d_{\mathrm{min},t+1}, P_{c,t+1})$. These quantities define the new state s_{t+1}, which is returned to the agent, closing the interaction loop. The process continues until a termination condition is met, marking the end of the episode and triggering a new reset for the subsequent scenario.

Step - Based Level

At the step-based level, the interaction between the agent and the environment develops within the data collection phase of the PPO training loop. During this phase, the current policy $\pi_{\theta_{\text{old}}}$ interacts with the environment over a fixed number of timesteps T, generating sequences of transitions $(s_t,\ a_t,\ r_t,\ s_{t+1})$. Each transition corresponds to a discrete simulation interval, during which the environment propagates the spacecraft dynamics according to the impulsive maneuver selected by the agent and computes the resulting reward reflecting the achieved performance. The resulting reward r_t, together with the transition $(s_t,\ a_t,\ r_t,\ s_{t+1})$, is stored in the rollout buffer, as shown in Figure 4.7. To accelerate the data collection process, N_{env} parallel environment instances are employed, each running an independent conjunction scenario under the same policy. In this work $N_{\text{env}} = 4$ parallel environment instances are employed allowing multiple trajectories to be sampled simultaneously, significantly decreasing the overall training time. Once T timesteps have been collected for each of the N_{env} parallel environment instances, the rollout buffer contains a consistent dataset of $N_{\text{env}} \cdot T$ transitions. These data are subsequently processed to compute the advantage estimates $\hat{A}_t$ and value targets V_t^{targ} , which are then used during the optimization phase to update the parameters of the actor and critic networks. After each optimization cycle, the new policy π_θ replaces the old one, and a new rollout iteration begins, continuing until the total number of iteration N_{iter} is reached.

Stable-Baseline3 Implementation

To ensure consistency, modularity, and compliance with the state of the art DRL standards, the PPO training algorithm has been implemented using the open source library Stable-Baseline3 (SB3) [74]. SB3, developed in PyTorch, provides a robust and standardized framework for defining, training, and evaluating DRL agents, allowing a seamless integration between the custom orbital environment and the PPO training pipeline. Since SB3 operates on a step-based logic, the training process was organized around the number of agent-environment interaction rather than on full episodes. Accordingly, the total number of training timestep is define:

$$T_{\text{tot}} = N_{\text{iter}} \cdot N_{env} \cdot T \tag{4.56}$$

Chapter 5

Training and Evaluation

The objective of this chapter is to assess the capability of the proposed DRL framework to plan effective avoidance maneuvers in time-critical conjunction scenario, between a maneuverable satellite and a single non-cooperative debris, characterized by a short time notice between conjunction detection and the expected TCA. A perturbed orbital dynamics model is considered, accounting for the effects of atmospheric drag and the first two zonal harmonics of Earth's gravitational potential, J_2 and J_3. The choice to include only these perturbations, while neglecting other such as SRP and third-body effect, is justified by the fact that they are the most relevant for satellite in LEO and for the short time scale considered in this work. Moreover, including the neglected perturbation, would considerably increase the computational cost of integrating the equations of motion at each decision step, without providing any meaningful improvement in the simulation fidelity.

5.1 Experimental Setup

The experimental setup introduced in this section, with parameters defining the simulation environment, establishes the overall configuration used to train and evaluate the DRL agent.

5.1.1 Simulation Parameters

The first parameters to be defined concerns the temporal resolution of the decision process and the operational limits of the impulsive maneuvers performed by the agent. These parameters determine how frequently the satellite can perform an avoidance maneuver and the maximum velocity increments allowed per step per axis ($\Delta v_{max,t}$) and per episode (Δv_{MAX}). Table 5.1 summarize the adopted values.

Parameter	Explanation	Adopted Value
Δt	Decision timestep	120 s
$\Delta v_{max,t}$	Maximum axis maneuver increment	0.04 m/s
Δv_{thr}	Maneuver threshold	0.012 m/s
Δv_{MAX}	Propulsive budget	3 m/s

Table 5.1: Operational Parameters

The decision timestep was set to 120 s, representing the interval at which the agent interacts with the environment by selecting a new action. This value results from a trade-off between control capability and computational efficiency. A shorter interval would increase the number of actions per episode, enabling more precise maneuvering but substantially raising the total number of training steps required for policy convergence, and consequently, the computational cost. On the other hand, larger timestep would reduce the agent's ability to finely adjust the trajectory, potentially reducing the effectiveness of the avoidance strategy in time-critical scenarios.

Satellite Initialization

The satellite state is initialized at the reference epoch $t_0 = 15$ May 2025, 13 : 51 : 00 (UTC) and defined by the classical orbital elements listed in Table 5.2, together with the corresponding state vector in ECI RF at the same epoch.

COE	Value
a [km]	7000.00
e [-]	0.05000
i [deg]	35.0000
Ω [deg]	0.00000
ω [deg]	10.0000
θ [deg]	30.0000

$\mathbf{X}(t_0)$	Value
$\mathbf{r}(t_0)$ [km]	[5126.90, 3523.98, 2467.52]
$\mathbf{v}(t_0)$ [km/s]	[−4.92218, 5.04588, 3.53316]

Table 5.2: Satellite's initial state at epoch t_0

As described in Section 4.2, the physical and geometric characteristics of the satellite were defined through the *ESA DISCOSWeb API*, by filtering all active payloads within the orbital region surrounding the reference orbit. A statistical analysis on the resulting datased, comprising 1351 active satellites, returned an average satellite HBR of $\bar{R}_s = 3.916$ m with a standard deviation $\sigma_{R_s} = 1.504$ m. To define a single, conservative but physically consistent value, the representative radius used is defined as $R_s = \bar{R}_s + \sigma_{R_s}$. Similarly, the statistical analysis of the average cross-sectional area yielded $\bar{A}_s = 21.307\ m^2$ with $\sigma_{A_s} = 11.901\ m^2$ leading to a representative value of $A_s = \bar{A}_s + \sigma_{A_s}$. From the same dataset, the average mass of the filtered satellites is $m_s = 505$ kg. The complete set of adopted geometric and physical parameters for both the satellite and the debris, derived for the last as discussed in Section 4.2, is summarized in Table 5.3 below.

Satellite		Debris	
Parameter	*Value*	*Parameter*	*Value*
R_s [m]	5.42	R_d [m]	0.36
A_s [m^2]	30.208	A_d [m^2]	0.4071
m_s [kg]	505	m_d [kg]	4.071

Table 5.3: Satellite and Debris Characteristics

5.1.2 Close Approach Database Generation

The close approach database constitutes the core of the training and evaluation process as it provides the agent with a wide variety of conjunction geometries and collision risk conditions. The nominal conjunction time t_{tca} is sampled between a predefined time window starting from the initial epoch t_0, whose limits are set to $t_{min} = 3600$ s and $t_{max} = 7200$ s. This range was selected to reproduce time-critical collision avoidance scenario, consistent with the motivation of this thesis. The disturbances applied to the position and velocity of the spacecraft to get the debris state at the TCA are generated according to the distributions defined in Section 4.4.2, with uniformly sampled standard deviations:

$$\sigma_{\text{pos}} \sim \mathcal{U}(0.1, 1.2) \text{ km} \tag{5.1}$$

$$\sigma_{\text{vel}} \sim \mathcal{U}(0, 0.1) \text{ km/s} \tag{5.2}$$

The parameters defining the uniform distributions σ_{pos} and σ_{vel} were tuned to generate a database where 100% of the generated encounters exhibit a probability of collision $P_c \geq 10^{-4}$, a value high enough to require the execution of an avoidance maneuver. The rotation angle to get the debris velocity is bounded within $\alpha_{\text{min}} = 10°$ and $\alpha_{\text{max}} = 45°$, ensuring sufficiently distinct encounter geometries while avoiding unrealistic head-on configurations. The variety of encounter geometries of the generated database enables the learned policy to generalize effectively, allowing the agent to perform avoidance maneuvers satisfying the imposed objective, even in conjunction never encountered during training. This concept, often referred to as domain randomization in DRL, represents one of the main strength of the adopted approach, as it eliminates the need to retrain the model for each new conjunction, significantly reducing computational cost and training time.

The complete database consists of $N_{CA} = 50$ independent conjunction scenarios. Of these, 80% are used for training, while the remaining 20% are only used for the evaluation of the learned policy. For completeness, both the training and evaluation complete database are reported in Appendix D. To provide a visual overview of the generated database, Figure 5.1 shows the orbital configuration of the training set.

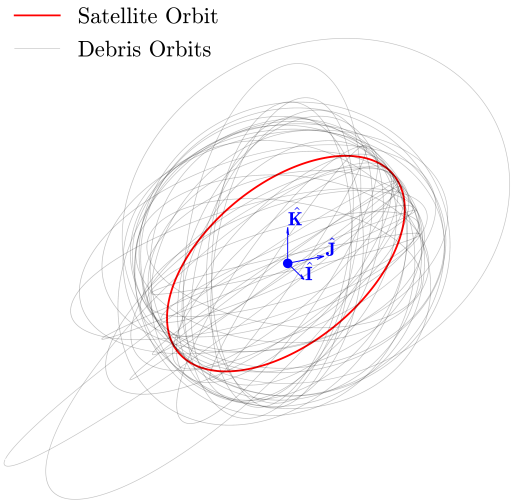


Figure 5.1: Training Set Orbital Configuration (ECI RF)

5.1.3 Deep Reinforcement Learning Training Configuration

The DRL training configuration was defined to ensure both a stable and computationally efficient learning process. The main parameters influencing the training performance, summarized in Table 5.4, were selected through a trade-off between stability and computational cost.

Variable	Explanation	Adopted Value
N_{env}	Environment instances	4
T	Rollout steps	512
N_{iter}	Policy update iterations	500
T_{tot}	Total steps	1024000
K	Number of epochs	10
M	Mini Batch Size	256

Table 5.4: PPO Algorithm Training Parameters

To accelerate the data collection phase of the PPO training loop, each policy update was performed on 4 parallel instances of the environment. At each iteration, the current policy $\pi_{\theta_{old}}$ was rolled out for 512 steps over each environment instance, resulting in a total rollout length of 2048 state transition per iteration. A larger rollout length increases the set of sample used to estimate the advantage function, reducing the variance of the policy gradient, but also increases the policy lag, since samples are collected with slightly outdated parameters. Conversely, shorter rollouts increase the update frequency but lead to higher variance, more noise, in the advantage estimates. The chosen value of 512 steps (2048 over the 4 instances) is adopted to balance stable gradient estimates while keeping the policy update frequency adequate.

Before performing the parameters update, the 2048 sample training batch was randomly divided into mini-batches of 256 transitions. Each mini-batch was then used to optimize the policy and value networks according to the PPO objective function, $J_t^{PPO}(\theta, w)$, presented in Section 3.4.1. The optimization over each mini batch was repeated for $K = 10$ epochs, allowing multiple passes through the data to refine the policy parameters before collecting new trajectories. A mini-batch size of 256 was found to effectively smooth the stochasticity of the gradient without excessively increasing computational cost. Larger mini-batches typically lead to more stable training but require more memory and reduce the beneficial randomness in gradient updates. The number of epochs per update was set to 10, which enables multiple optimization passes through the same batch to exploit the collected data more efficiently, while preventing overfitting to on-policy samples, a risk that increases for higher epoch counts.

The number of policy-update iterations was set to 500, corresponding to a total of $1.024 \cdot 10^6$ environment steps. This configuration was determined empirically after multiple training campaigns, observing that after approximately one million interactions the average episode reward reached a plateau and the policy became stable and nearly deterministic, indicating convergence of the learning process.

Proximal Policy Approximation Hyperparameter

The Hyperparameters specifically defining the PPO update process and its surrogate objective function, $J_t^{PPO}(\theta, w)$, are summarized in Table 5.5.

Variable	Explanation	Adopted Value
$\eta_\theta,\ \eta_w$	Networks learning rate	$3 \cdot 10^{-4}$
–	Actor structure	$[16, 64, 64, 3]$
–	Critic structure	$[16, 64, 64, 1]$
γ	Discount factor	0.99
ϵ	Clip ratio	0.2
c_1	Value loss weight	0.5
c_2	Entropy weight	0.005

Table 5.5: PPO Algorithm Hyperparameters

Most of the PPO hyperparameter adopted in this work correspond to the default values of the SB3 PPO implementation, since empirical tuning confirmed that these values provide the best trade-off between training stability and overall performance within the developed framework. The default architecture of the actor and critic networks, both implemented with two hidden layers of 64 neuron each, ensures an efficient structure without unnecessary computational cost. The hyperbolic tangent (tanh) activation function was adopted for all layers in both networks, producing outputs bounded in the range [-1, 1]. This property is particularly beneficial for the actor network, where the three output neurons correspond to the components of the continuous action vector and must subsequently be mapped to physical realistic impulsive maneuvers along each direction of the ECI RF.

The discount factor γ determines the relative importance assigned to the immediate and future reward. In the context of this thesis, where the objective of each episode is to reach a terminal success condition representing the completion a series of CAMs, the discount factor was set to 0.99. This choice encourages the agent to adopt a long-term strategy that consider the cumulative effect of multiple actions rather than focusing solely on immediate improvements.

The entropy coefficient c_2 was changed with respect to the default configuration. It introduces a term in the PPO objective that penalizes overly deterministic policies, encouraging exploration. In early stage of the training, it is important for the agent to explore a wide range of possible policies, to effectively discover the optimal control strategy that yields the most discounted return. This coefficient must be carefully tuned because excessively high values lead to persistent random behavior and prevent convergence, while low value accelerate convergence but increase the risk of premature exploitation and convergence to suboptimal policies.

Reward Coefficient Tuning

The tuning of the reward coefficient, was carried out through several training runs, during which different combination were tested to achieve a stable and balanced policy behavior.

The final configuration, reported in Table 5.6, was selected as the best compromise between the main learning objective, i.e. minimizing collision probability, minimizing orbital deviation and ensuring propulsive efficiency. As discussed in Section 4.3.4, each coefficient determines the relative importance assigned to a specific term in the total reward function, influencing the agent's decision making strategy during training.

Reward Coefficient	Adopted Value
C_{P_c}	2.5
$C_{\Delta COE}$	4.5
$C_{\Delta v}$	0.1, 0.5

Table 5.6: Reward coefficients

The coefficient associated with the propulsive term, $C_{\Delta v}$, was implemented as a dynamic weight that increases from 0.1 to 0.5 once the two primary objectives, collision risk mitigation and orbital maintenance are achieved. This was designed to discourage the agent from applying unnecessary maneuvers once a safe and compliant configuration has been reached. The initial low value ensures that fuel consumption is penalized only marginally during the early phase of an episode, allowing the agent to focus on learning effective avoidance strategies before optimizing propulsive efficiency.

Finally, Table 5.7 reports the thresholds adopted for normalizing the deviations of the COEs with respect to the reference orbit.

τ_{COE}	Adopted Value
τ_a [km]	1
τ_e [-]	0.005
τ_i [deg]	0.05
τ_Ω [deg]	0.05
τ_ω [deg]	0.05

Table 5.7: COEs Deviation Threshold - τ_{COE}

5.2 Training Performances

Once the experimental setup had been fully defined, training was performed over 500 N_{iter}, corresponding to a total of 1024000 steps, and required approximately 12 hours to complete. This section analyzes the main metrics tracked during the learning process, with the aim of evaluating both the stability of PPO algorithm and the evolution of policy behavior during training. The analysis is organized by considering both indicators directly related to the dynamics of the update algorithm and metrics that describe how the policy progressively develops behaviors consistent with the objectives of the collision avoidance problem.

5.2.1 Proximal Policy Optimization Training Performance

Among the indicators commonly used to assess the stability of PPO training process, two quantities are particularly relevant: the evolution of the total loss (Figure 5.2), and the explained variance (Figure 5.3).

Total Loss The total loss is defined as the opposite of the PPO objective function, $J_t^{PPO}(\theta, w)$. Since the PPO objective must be maximized for the policy to improve, the loss is minimized during training. The behavior of the total loss over the entire training process is reported in Figure 5.2.

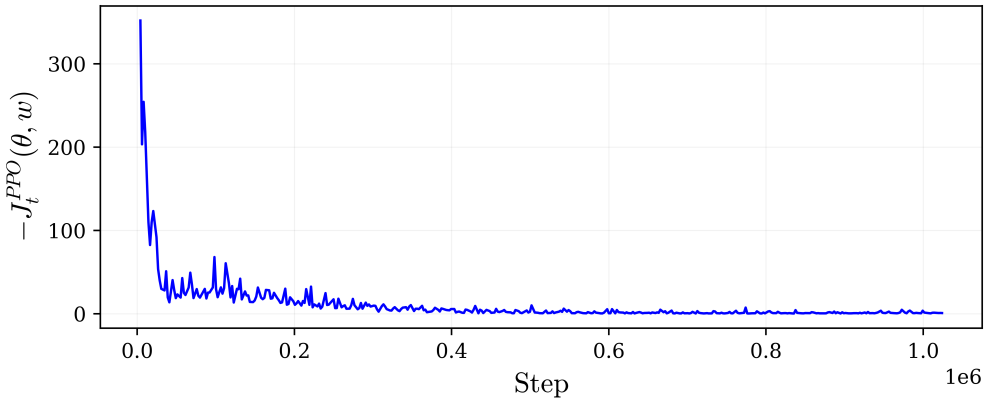


Figure 5.2: Evolution of PPO total loss over 1024000 training step

The loss shows a steep decrease during the initial training phase, indicating the rapid adjustment of both policy and value networks parameters. After this initial stage, the loss progressively stabilizes around low values, indicating consistent policy refinement and reduced parameters updates. The absence of large fluctuation in the later stages confirms that the agent reaches a stable training regime, approaching a nearly deterministic control action.

Explained Variance The explained variance (EV) quantifies how accurately the value network $v_w(s_t)$ predicts the target state value v_t^{targ} and it is formally defined as:

$$EV = 1 - \frac{\sigma^2(v_t^{\text{targ}} - v_w(s_t))}{\sigma^2(v_t^{\text{targ}})} \tag{5.3}$$

The goal during training is to achieve EV values close to $EV \approx 1$, indicating that the critic provides highly accurate predictions of the target value. The evolution of the EV over the entire training process is reported in Figure 5.3.

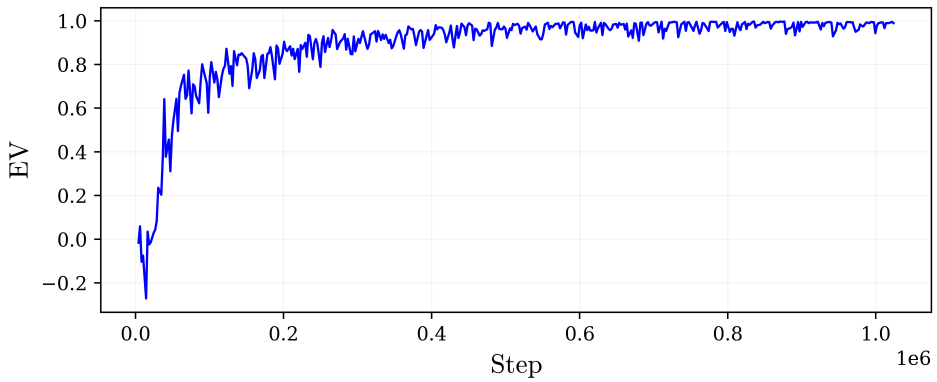


Figure 5.3: Evolution of EV over 1024000 training step

At the beginning of the training, the EV is negative, indicating that the critic is unable to provide meaningful estimates of the target value, an expected behavior since the networks start from untrained, random parameters. Within the first 200000 training step, the EV raises to approximately 0.8, highlighting that the critic progressively learn an accurate mapping between states and target values. In the later stages of the training, the metric reaches values close to 1 with very small oscillation, confirming that value prediction have become highly reliable.

5.2.2 Policy Training Performance

The policy performance during training is assessed by observing the evolution of the average episode reward, the average success rate, the average episode burn count, and the average Δv_{used} per episode, reported respectively in Figure 5.4, Figure 5.5, and Figure 5.6.

The average reward is computed as the sum, over the episodes contained in each PPO iteration cycle N_{iter}, of the dense rewards and the terminal reward accumulated within each episode. During the initial phase of training, within the first 200000 steps, the reward increases rapidly, followed by a more gradual, steady growth until the end of training, where the maximum value is reached.

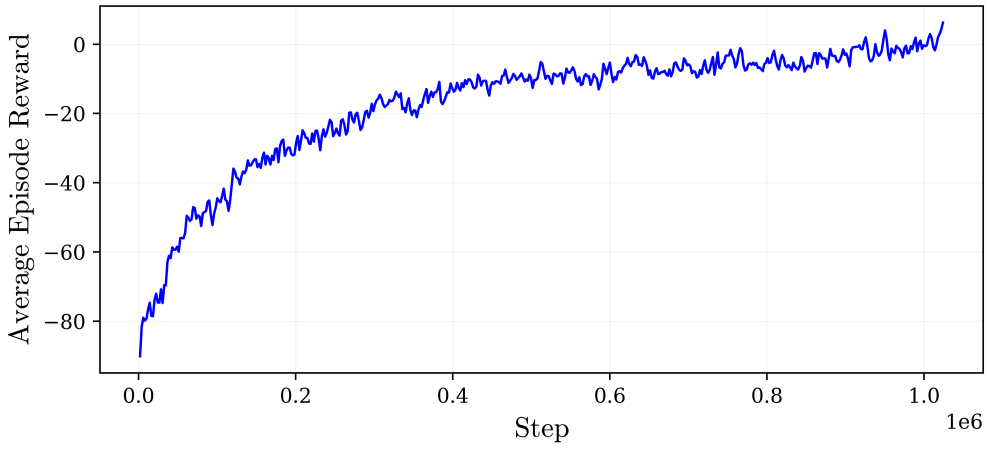


Figure 5.4: Evolution of Average Episode Reward during training

This behavior, observed in Figures 5.4, is consistent with a policy that, as training progresses, achieves success conditions more frequently, reducing the penalties associated with the dense terms of the reward function. The success rate, measuring the fraction of episodes ending with a successful mitigation condition, i.e. $P_{c,\mathrm{tca}} \leq P_{c.goal}$ and $\rho_{\mathrm{tca}} \leq 1$, provides a more directly interpretable performance metric.

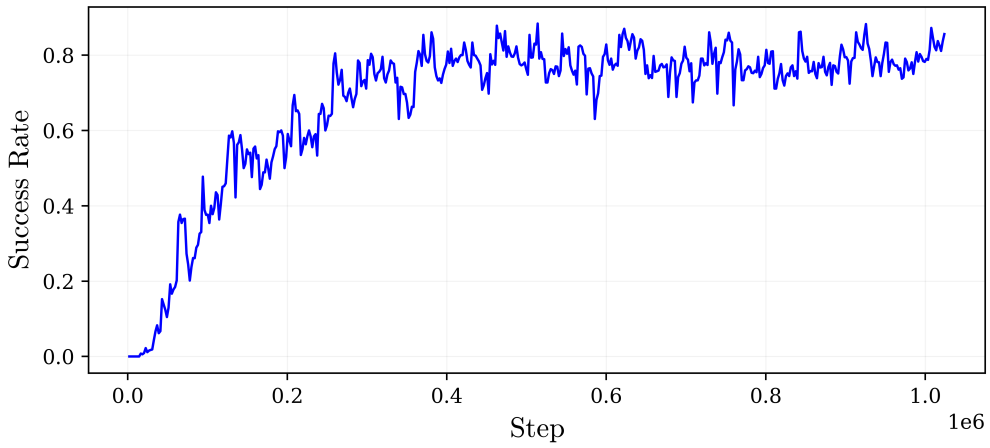


Figure 5.5: Evolution of success rate during training

As shown in Figure 5.5, after an initial exploration phase, in which almost all episodes result in failure, the success rate increases rapidly and reaches values around 0.5–0.6 within the first 200000 steps. This behavior explains the rise observed in the average episode reward during the same interval. After approximately 400000 steps, the success rate stabilizes around 0.8–0.85, indicating that the policy has learned a robust avoidance strategy with respect to the variability of conjunction scenarios. The fact that the average reward continues to gradually increase suggests that the policy is not only improving the probability of success, but is also refining how it achieves success, progressively reducing both orbital deviations and propulsive resources.

A more detailed explanation of this refinement phase can be seen in the average episode burns count and the average episode Δv_{used}, shown in Figures 5.6. The average number of burns per episode quantifies the number of non-zero impulse maneuvers performed before the end of an episode. The trend shows that, in the initial and intermediate phases of training, this value remains almost constant, at around 46 impulses per episode. In this stage, the policy is mainly oriented towards achieving successful mitigation, distributing the pulses throughout the entire episode at the expense of propulsive efficiency. Once the success rate stabilizes around 0.8, the average number of burns begins to decrease significantly, converging towards values close to 40 per episode. This reduction indicates that, after learning to reliably satisfy safety and orbit constraints, the policy tends to eliminate redundant maneuvers, exploiting the structure of the reward function and, in particular, the presence of the step bonus in the absence of action when all constraints are satisfied.

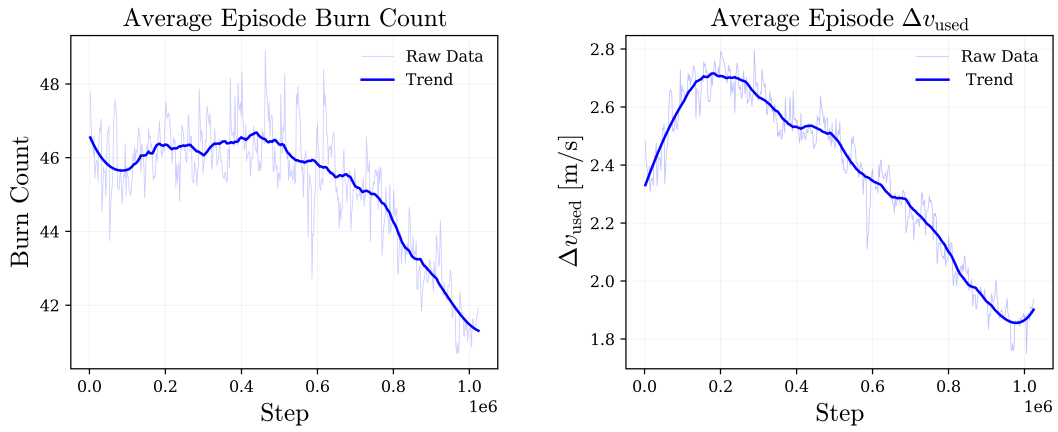


Figure 5.6: Evolution of burn count (left) and Δv_{used} (right) trend during training

A similar trend is observed when analyzing the average value of Δv_{used} per episode, shown in the right panel of Figure 5.6. In the early stages of training, the curve shows an increase from approximately 2.3 to approximately 2.7 m/s. In this exploratory phase, the policy tends to adopt more aggressive strategies, using a greater amount of the propulsive budget to achieve successful conditions more frequently. Around the same training horizon in which the success rate stabilizes (approximately 400000 steps), the average Δv_{used} begins to decrease progressively, converging towards values of the order of 1.9 m/s at the end of training. This transition indicates that, once a reliable avoidance strategy has been learned, the policy begins to fully exploit the structure of the reward function, which makes the use of Δv progressively more penalizing when the terms r_{P_c} and $r_{\Delta\text{COE}}$ are already close to zero.

5.3 Learned Policy Evaluation

After analyzing the proposed framework's training behavior, the collision avoidance capabilities of the learned policy are assessed through an evaluation phase. During training, the agent samples actions from a Gaussian distribution parameterized by the actor network, enabling the stochastic exploration necessary for policy optimization. During the evaluation phase, however, the actor network parameters are "frozen" at the final training iteration (at step 1024000) and executed in deterministic mode: no sampling is performed, and the selected action corresponds to the mean of the learned distribution. This ensures fully reproducible rollouts and allows for an objective evaluation of the final policy's performance.

In this work, the evaluation is carried out on the 20% portion of the generated close-approach database that was never used during training, corresponding to a set of 10 close approach scenarios.

5.3.1 Policy Generalization Capabilities

The evaluation phase allows to verify the generalization capability of the learned policy, i.e., its ability to transfer the avoidance strategies acquired during training to conjunction

scenarios never encountered before. The evaluation close approach database present a wide variability both in terms of orbital configuration and in the associated collision risk metrics, minimum miss distance, TCA, and initial collision probability. This diversity is fundamental for assessing the robustness of the learned policy with respect to diverse encounter geometries. Figure 5.7 and Table 5.8 show, respectively, the orbital configuration of the 10 scenarios and their collision risk metrics.

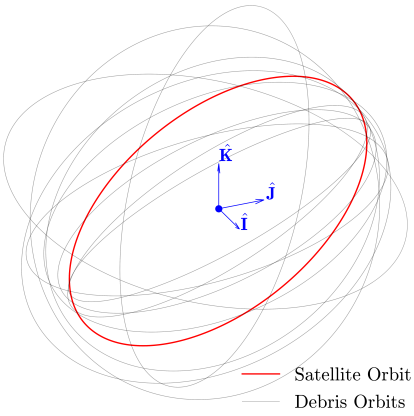


Figure 5.7: Evaluation Set Orbital Configuration (ECI RF)

DEB	t_{tca} [s]	$d_{\min}$ [km]	P_c
001	6192	0.458	2.21×10^{-3}
002	5528	1.426	2.40×10^{-4}
003	4952	0.173	1.19×10^{-2}
004	6107	0.899	5.99×10^{-4}
005	6179	0.685	1.02×10^{-3}
006	4995	1.567	1.99×10^{-4}
007	7025	0.559	1.51×10^{-3}
008	5106	0.791	7.69×10^{-4}
009	5748	0.592	1.35×10^{-3}
010	6580	0.442	2.36×10^{-3}

Table 5.8: Evaluation Close Approach Database

The deterministic rollout of the learned policy on the ten evaluation scenarios showed remarkable performance. The policy was able to achieve a successful mitigation in all 10 cases, meeting the two main objectives: reducing the P_c below $P_{c,goal}$ and maintaining the satellite orbit within operational limits. In particular, success was achieved with an average value of Δv_{used} of 2 m/s, with an average burn count of 42, and an average episode reward of approximately 14, results are fully consistent with the trends observed during the final stages of the training phase.

The observed behavior highlights the main strength of the adopted DRL framework: once trained, the policy acts as a mapping from the observed state to learned optimal action, obtained through real-time inference of the policy network. It is therefore not necessary to retrain the model for each new conjunction scenario. This highlights the potential of this approach in the context of time-critical collision avoidance maneuvers, where speed and reliability of the decision-making process play a crucial role.

5.3.2 Policy Rollout Analysis on a Representative Conjunction Scenario

To evaluate the operational behavior of the learned policy, its response is examined in detail on a specific conjunction scenario from the evaluation database. The objective is not to verify whether or not the policy meets the success criteria, already demonstrated in the previous section, but to show *how* the sequence of actions generated by the policy allows the objectives of the collision avoidance problem to be achieved.

In particular, the scenario associated with debris DEB007, reported in Table 5.8, is considered. This case represents a typical time-critical conjunction, perfectly representative of the operational context addressed in this thesis. The TCA is less than two hours after the initial epoch t_0, specifically after 7025 seconds (15 May 2025, 15:48:05 UTC). The nominal configuration has a minimum distance of 0.559 km and a collision probability of $1.511 \cdot 10^{-3}$, values that make it necessary to perform a mitigation action. Figure 5.8 shows the close approach geometry of the scenario, with a zoom around the TCA to highlight the critical nature of the encounter.

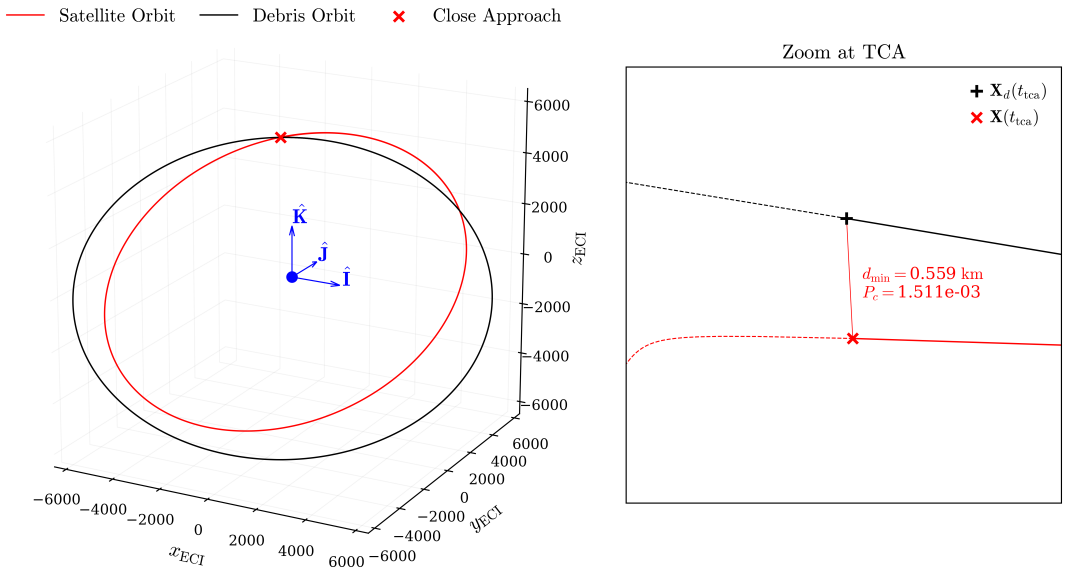


Figure 5.8: Close Approach Geometry

The deterministic rollout of the learned policy is then carried out over this specific scenario. At each decision step, the agent observes the state of the environment and selects the action as the mean of the learned policy distribution, resulting in a deterministic control input. The action are applied as impulsive Δv along the three axis of the ECI RF. The complete sequence of impulses generated during the episode is shown in Figure 5.9.

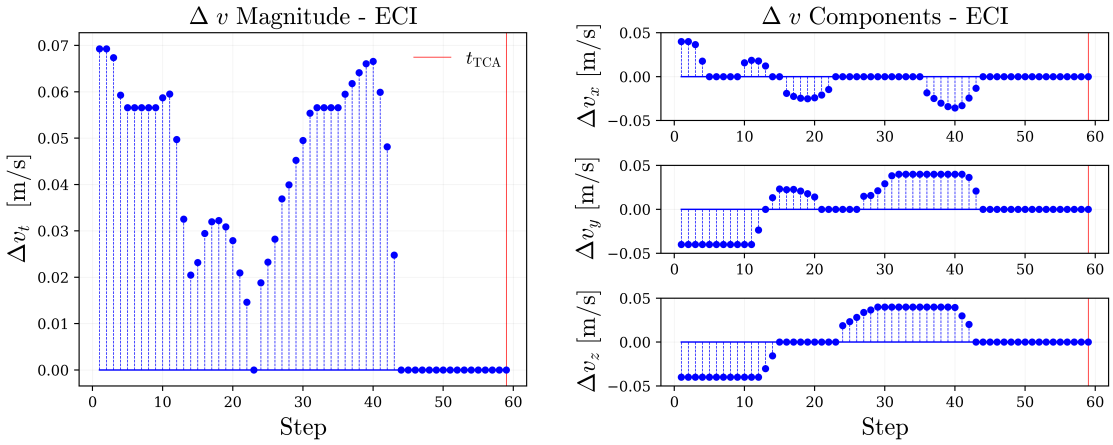


Figure 5.9: Collision Avoidance Maneuvers Sequence - ECI RF

The maneuvers sequence reflects the structure of the designed reward function. Once the primary objectives have been achieved, the penalty associated with propulsion budget increases, while the agent is rewarded for avoiding unnecessary maneuvers. This explains the progressive reduction in the magnitude of the impulses after step 40 and the complete absence of maneuvers between step 44 and the TCA. The effectiveness of the maneuver sequence is quantified in Figure 5.10, which reports the evolution of the minimum miss distance and collision probability at each decision step during the deterministic policy rollout.

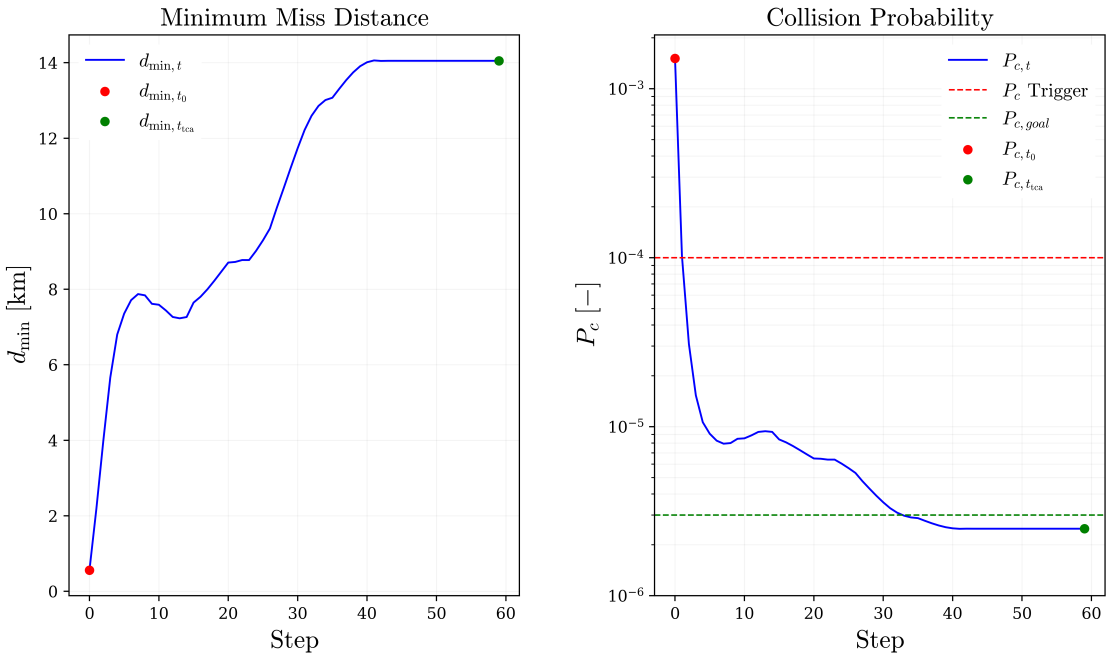


Figure 5.10: Minimum Miss Distance and Collision Probability Evolution

In the initial phase, the first velocity increment produce a rapid increase in the minimum distance (from less than 1 km to about 8 km in the first eight steps), leading to a reduction in the probability of collision by more than two orders of magnitude. After this initial rise, $d_{\min}$ exhibits a short plateau, consistent with the evolution of the orbital elements relative to the reference orbit, shown in Figure 5.11. In this interval the argument of perigee temporarily exceeds its operational limits, inducing the policy, to reduce the intensity and change the direction of velocity increment in order to restore orbital compliance. Once adequate margins are restored, the policy resumes applying higher magnitude impulses to further increase $d_{\min}$ and bring P_c below $P_{c,\text{goal}}$. It is interesting to note that, in order to achieve a more consistent reduction in the probability of collision, the policy temporarily accepts a deviation of the semi-major axis beyond the operational limit. Once P_c falls below the target value, the remaining impulses are aimed not towards further modifying P_c, but toward bringing a back within limits, ensuring full compliance with the success conditions at the TCA. The episode therefore ends with an effective avoidance that is also consistent with the mission objectives.

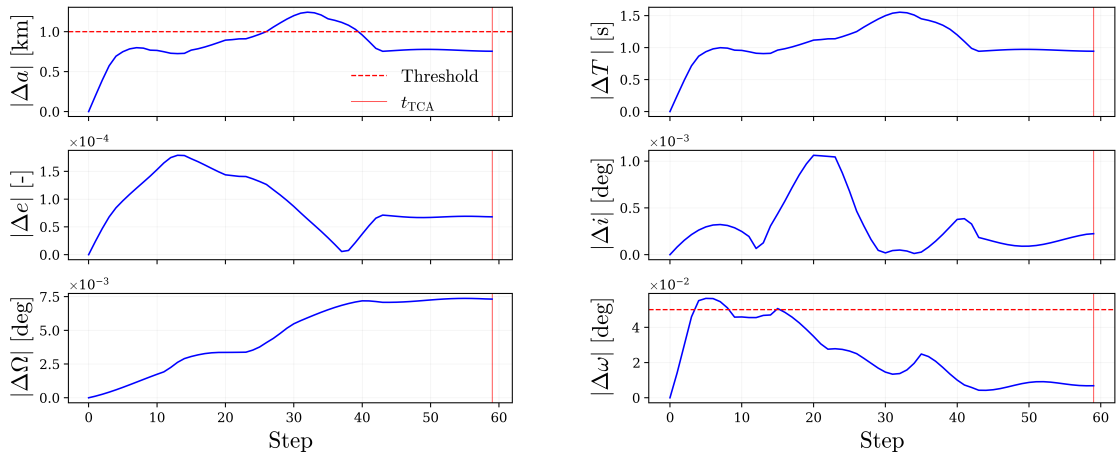


Figure 5.11: COEs Deviation During Maneuvers Sequence

Analyzing the deviation of the other orbital elements during the maneuvers sequence confirms the desired behavior. The magnitude of the Δv components selected by the policy ensure that eccentricity, inclination, and RAAN always remain within their respective operating thresholds. Table 5.9 summarizes the deviations of the five orbital elements at TCA from the nominal values of the reference orbit. As shown, the policy keeps the satellite's orbit within the required limits, with very only minimal percentage variations. This allows the satellite to resume its nominal activities immediately after TCA, without requiring further corrective orbit maintenance maneuvers.

COEs at t_{tca}	**Reference Orbit**	**Post CAMs Orbit**	**$\lvert\Delta$COE$\rvert$**	**Relative Variation**
a [km]	6996.74	6995.99	$7.55 \cdot 10^{-1}$	-0.0107194 %
e [-]	0.0493261	0.0493942	$6.81 \cdot 10^{-5}$	$+0.138061$ %
i [deg]	34.9857	34.9859	$2.24 \cdot 10^{-4}$	$+0.000571662$ %
Ω [deg]	359.455	359.447	$7.31 \cdot 10^{-3}$	-0.00222562 %
ω [deg]	10.6684	10.6616	$6.83 \cdot 10^{-3}$	-0.0637396 %

Table 5.9: COEs Relative Variation at TCA

Although the velocity increments are applied in the ECI RF, chosen to ensure consistency with the orbital propagator, a more intuitive and informative understanding of the policy behavior is obtained by analyzing the maneuvers in the Radial-Transverse-Normal (RTN) RF (a detailed description of which is given in Appendix A). The maneuvers sequence is shown in Figure 5.12, which reports the components of the impulsive Δv along the R, T, and N axes, together with the azimuth (Az) and elevation (El) angles that describe the direction of the overall Δv vector in the local system.

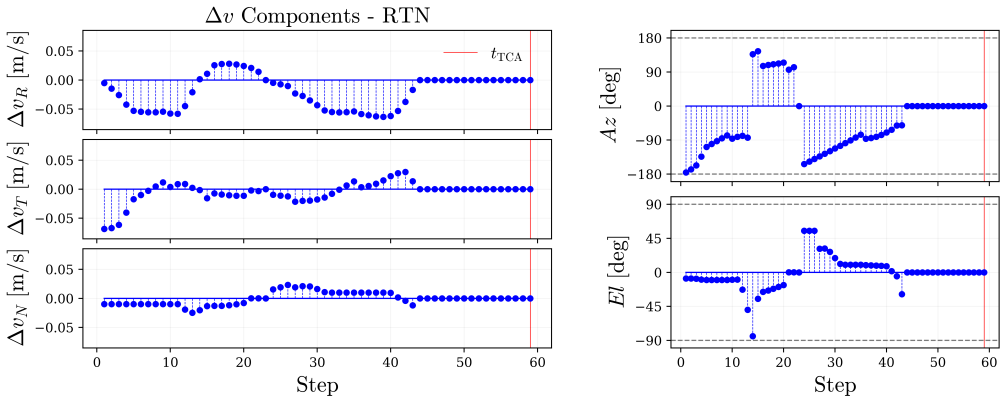


Figure 5.12: Collision Avoidance Maneuvers Sequence - RTN RF

Analyzing the Δv components in the RTN RF is possible to directly interpret the contribution of each maneuver to the change in the geometry of the encounter. In the first steps, the transverse component Δv_T is dominant and drives the rapid increase in the $d_{\min}$. Subsequently, the applied maneuvers are mostly oriented along the radial axis. Radial corrections, for a comparable Δv, produce more contained variations in semi-major axis and eccentricity than purely tangential impulses. For this reason, the policy tends to favor radial impulses to keep the orbital elements as close as possible to their respective operating limits, while continuing to reduce P_c. The most relevant policy behavior emerges from the analysis of the normal component Δv_N. Out-of-plane impulses directly modify i and Ω and are therefore significantly more expensive in terms of Δv than in-plane corrections. For this reason, the magnitude of Δv_N remains systematically lower than the radial and transverse components. This trend is reflected in the evolution of the elevation angle: in the first half of the episode, the elevation is mainly negative, while in the second half it becomes positive. The sequence of normal impulses with opposite signs compensates for the variation in i and Ω.

Lastly, the final configuration after maneuvers is shown in Figure 5.13.

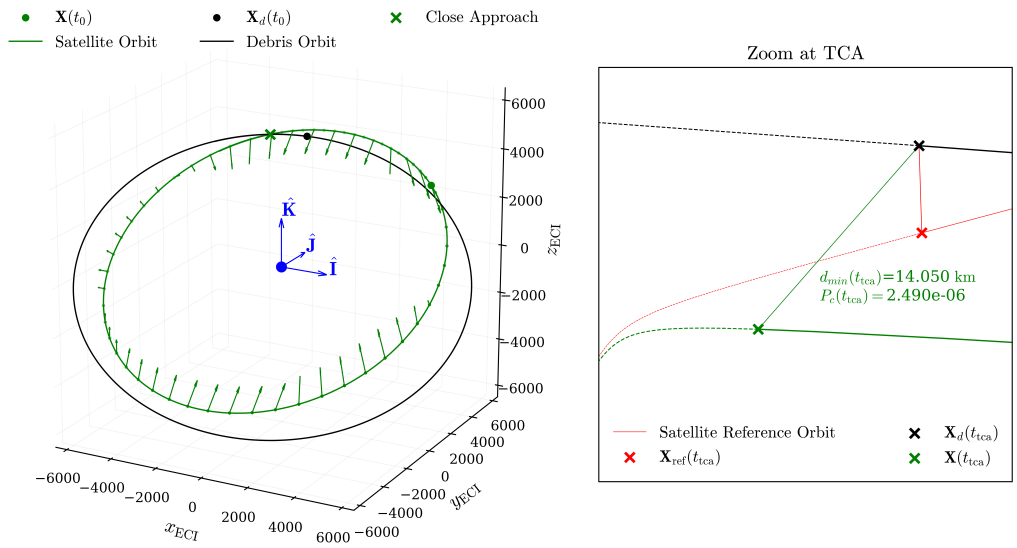


Figure 5.13: Post Maneuvers Close Approach Geometry

The left panel reports the orbital geometry resulting from the entire avoidance maneuvers sequence. The green arrows represent the velocity increments applied during the episode, where their direction and length indicates the orientation and magnitude of the resulting Δv vector, while their length reflects its magnitude. Consistently with the analysis in the RTN reference system, the impulses are mainly oriented along the radial direction, while the out-of-plane component remain very limited. The right panel provide a zoomed view of the relative geometry at TCA, comparing satellite position after the maneuvers with the debris orbit and the reference orbit. The displacement produced by the maneuvers sequence results a final minimum distance of $d_{\min}(t_{\text{tca}}) = 14.050$ km, a significant increase compared to the initial 0.556 km. As a result, the P_c is reduced from $1.511 \cdot 10^{-3}$ to $2.490 \cdot 10^{-6}$, well below the target threshold of $3 \cdot 10^{-6}$ and nearly three orders of magnitude lower than initial value. Overall the maneuver sequence generated by the learned policy successfully resolves the conjunction scenario, increasing the geometric separation and reducing the collision probability to a safe level while maintaining the orbital configuration within the prescribed operational limits.

5.3.3 Post Maneuvers Conjunction Risk Assessment

After assessing the effectiveness of the proposed approach in planning collision avoidance maneuvers, it is also necessary to verify that the maneuvers performed do not result in a new increase in the risk of collision with the same object involved in the close approach. The probability of collision must be below the threshold of 10^{-4} in the period following the maneuvers. In particular, requirement *5.3.3.3(f)* of the ESA Space Debris Mitigation Requirements [10] states that, for missions in LEO the collision probability with any cataloged object must not exceed 10^{-4} in the four days following the planned maneuver.

To verify compliance with this requirement, a conjunction risk assessment was performed between the spacecraft and the debris considered in the close approach scenario for four days following the TCA. Both states of satellite and debris were propagated using the custom orbital propagator described in 2.4, including all relevant perturbations modeled in this thesis: atmospheric drag, zonal harmonics from J_2 to J_6, SRP, and gravitational perturbations from the Sun and Moon. The temporal evolution of the relative distance and the corresponding collision probability are shown in Figure 5.14.

The four days propagation shows that following the maneuvers performed, the two objects do not return to a high-risk configuration. As shown by the red vertical line in Figure 5.14, the minimum relative separation occurs 32 hours, 15 minutes, and 5 seconds after the TCA, corresponding to 17 May 2025, 00:03:05 (UTC). At that time, the relative distance between the satellite and debris is 422.929 km, while the probability of collision is $P_c = 2.748 \cdot 10^{-9}$, well below the operational threshold of 10^{-4}.

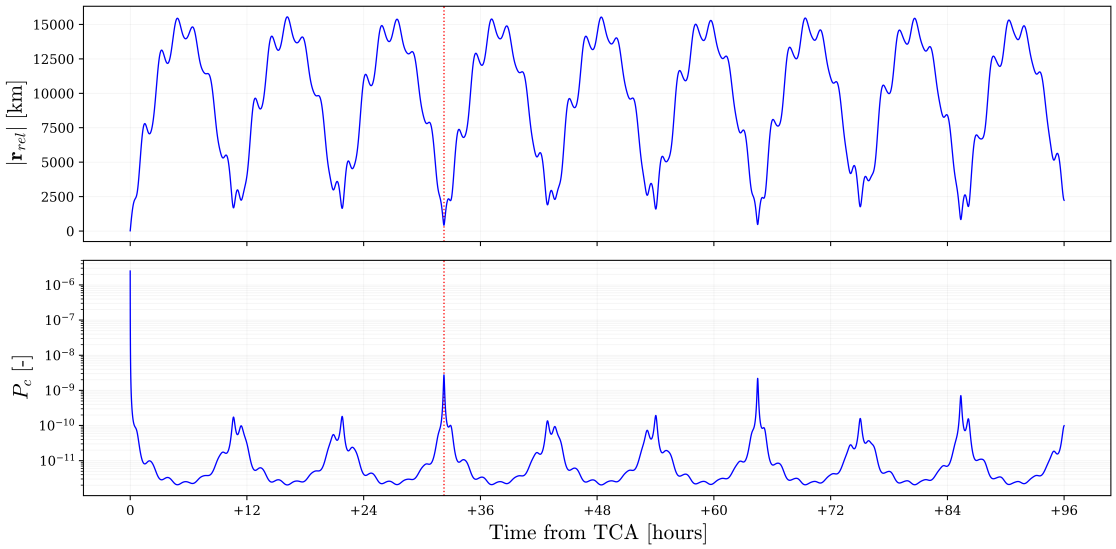


Figure 5.14: Relative distance and collision probability during the four day post-maneuvers propagation

This result confirms that the sequence of maneuvers planned by the learned policy not only effectively resolves the conjunction event at TCA, but also guarantees the absence of further critical events in the following four-day interval, fully satisfying the post-CAMs safety requirement.

Chapter 6

Conclusions & Future Work

After outlining the context, the problem statement, the theoretical foundations, and the formulation, training, and evaluation of the proposed DRL framework, this chapter summarizes the main conclusions of the work and presents possible future developments aimed at further improving its performance and operational applicability.

6.1 Conclusions

This thesis addressed the challenge of autonomously planning CAMs in LEO under time-critical constraints, motivated by the growing density of space traffic, the operational limitations of ground-based procedures, and the technological gap identified for short-notice conjunction events. As highlighted in Section 1.2.3, current CAM workflows may require hours to days between the reception of a CDM and the execution of a maneuver, a timeframe incompatible with short-notice conjunction events, particularly in the context of dense orbital environments and large constellations. These considerations motivated the central objective of this thesis: to develop and evaluate a DRL framework for the autonomous planning of CAMs in scenarios characterized by limited decision time windows, involving an active and maneuverable satellite and a single non-cooperative debris object. The results obtained confirmed that the proposed framework fully satisfied this objective. Training performance analyses showed a stable learning process, with convergence of the success rate to approximately 86% and progressive refinement of the policy's propulsive efficiency. The evaluation of the learned policy on a database of ten unseen conjunction scenarios showed successful mitigation in all cases, thereby confirming the extent to which the learned policy is able to generalize across different conjunction geometries and to provide response times consistent with an autonomous planning of CAMs within a constrained decision window. The detailed rollout examination on a representative conjunction scenario further illustrated the operational behavior of the learned strategy: the collision probability was reduced by almost three orders of magnitude while the orbital deviations at TCA remained within the prescribed limits, confirming that the sequence of avoidance maneuver do not compromise the spacecraft's ability to resume its nominal operations. Finally, the post-maneuver analysis confirmed that the application of the maneuvers sequence does not introduce new short-term risks, as demonstrated by the four-day propagation following

TCA. The collision probability remained consistently below 10^{-8} indicating full compliance with operational safety requirements and the absence of condition that would necessitate an additional avoidance maneuver. Equally important, the time required for the maneuver planning corresponds solely to the inference time of the policy network, on the order of seconds, demonstrating that the framework can operate onboard without the need for retraining, even in the presence of previously unseen conjunctions scenario.

Overall, this thesis demonstrated that a DRL-based approach can serve not merely as a theoretical alternative to classical CAM planning, but as a valuable solution to autonomously plan effective collision avoidance maneuvers in time-critical scenarios. The results confirm the feasibility of adopting RL for next-generation autonomous CAM systems and lay the foundation for future developments aimed at integrating such frameworks into operational space traffic management architectures.

6.2 Future Work

Reduced Decision Timestep A first extension of the proposed framework concerns the reduction of the decision timestep with which the agent selects the action to be applied. The current architecture uses a 120 second interval, which has proven effective in ensuring stability during training, allowing an overall success rate of approximately 86% to be achieved, producing maneuver sequences capable of resolving the majority of conjunction scenarios. However, it has been observed that the remaining 14% of failures are mainly due to the agent's limited ability to react appropriately when the TCA approaches the lower limit of the available time window (t_{min}), which is approximately one hour. A reduction in the decision step, supported by greater computational resources needed to handle a higher number of iterations during training, would allow the framework to be extended to even more stringent scenarios. This improvement would potentially make it possible to plan CAM maneuvers in time windows of less than one hour compared to the TCA, expanding the applicability of the method to the category of "Last-Instant" CAM proposed in [26].

POMDP Formulation A second possible improvements is to reformulate the CAM planning problem as a Partially Observable Markov Decision Process (POMDP). Unlike the fully observable MDP adopted in this thesis, in a POMDP the agent cannot directly access the true state of the environment, it receives an observation that provide only partial, noisy, or uncertain information about the underlying dynamical state. This formulation is particularly relevant for realistic conjunction scenarios, where orbit determination errors, sensor noise, and asynchronous updates limit the accuracy of the state information available. The mathematical formulation of an POMDP is an extension of the MDP tuple through the introduction of two additional components $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{O}, \mathcal{Z}, \gamma)$, where:

- $\mathcal{O}$: observation space, set of all possible observation ($o_t \in \mathcal{O}$) the agent receive from the environment.
- $\mathcal{Z}$: observation model, defined as $\mathcal{Z}(s_{t+1}, a_t, o_t) = \mathcal{P}(o_t|s_{t+1}, a_t)$. It describe the probability of observing o_t after taking action a_t and transitioning to the hidden state s_{t+1}.

Although the environment evolves according to a hidden state, the agent must still make decision based solely on the limited information provided by the observations. For this reason, POMDP formulation introduce the concept of a belief state $b_t(s_t)$, a probability distribution that represent the agent best estimate of the state given all past actions and observations.

Multi-Debris and Multi-Agent Scenarios Addressing multi-debris scenarios would require extending the representation of the system state to include multiple encounter geometries simultaneously, along with a reformulation of the reward function capable of assessing the overall risk generated by all objects involved. If both objects are maneuverable satellite, the problem takes on the nature of a multi-agent system, in which each satellite must plan its own maneuver while taking into account the actions of the other. Configurations of this type can be addressed using Multi-Agent Reinforcement Learning techniques, which allow cooperative interactions to be modeled, thus opening up the possibility of addressing more complex conjunctions and more realistic operational scenarios.

Appendix A

Reference Frames

A.1 Earth-Centered Inertial (ECI)

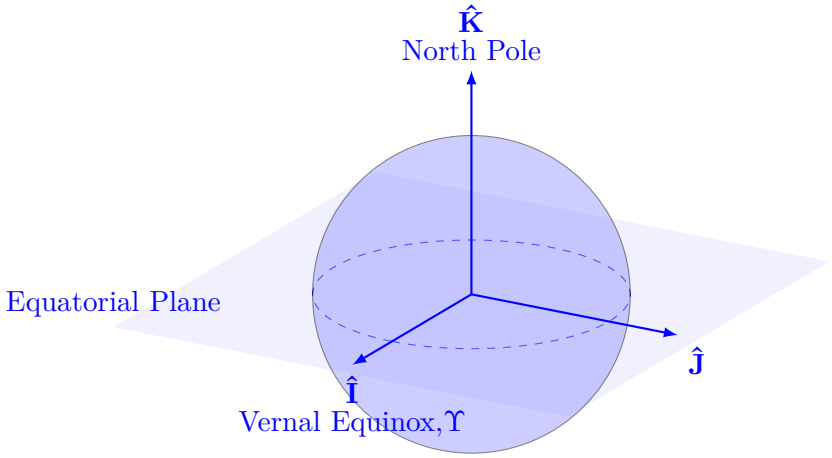
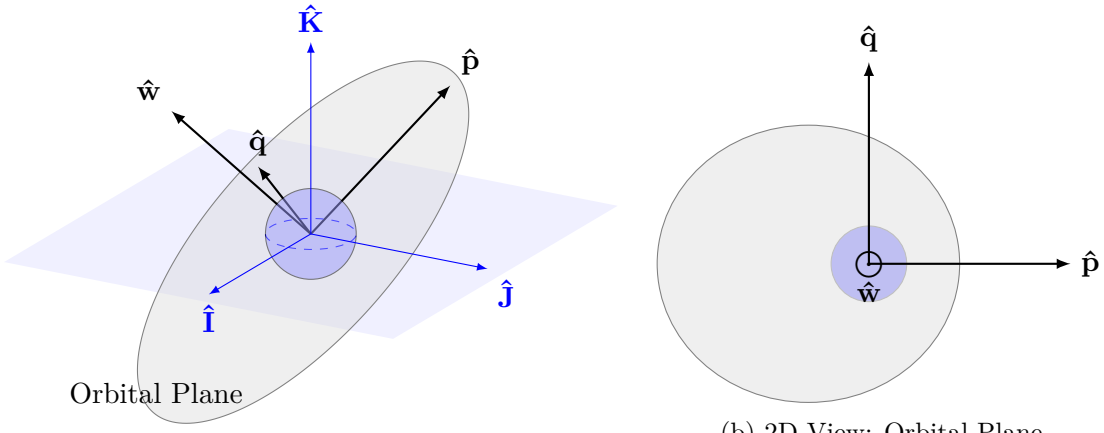


Figure A.1: Earth-Centered Inertial (ECI) Reference Frame

The study of the orbital motion of a satellite requires, as a fundamental first step, the definition of an inertial reference system in which to express kinematic and dynamic quantities. The reference system most commonly adopted in orbital mechanics for Earth-orbiting satellites is the equatorial geocentric inertial system, known as Earth-Centered Inertial (ECI) (Figure A.1). The ECI RF is centered in Earth's center of mass with the equatorial plane as the reference plane. The three orthogonal axes defining the frame are:

- $\hat{\mathbf{I}}$: directed toward the vernal equinox, or the point of intersection of the equatorial plane and the ecliptic plane, at the constellation Aries (Υ) at J2000.
- $\hat{\mathbf{K}}$: orthogonal to the equatorial plane, directed toward Earth's Geographic North pole.
- $\hat{\mathbf{J}}$: obtained through right hand rule, such that the right-handed triad is completed.

A.2 Perifocal (PQW)



(a) 3D View

(b) 2D View: Orbital Plane

Figure A.2: Perifocal (PQW) Reference Frame

The perifocal reference frame (PQW) (Figure A.2) is an inertial coordinate system whose origin is located at the focus of the orbit, which coincides with the Earth's center of mass. The fundamental plane of the frame is defined by the orbital plane of the satellite and the three orthogonal axes defining the frame are:

- $\hat{\mathbf{p}}$: is aligned with the eccentricity vector $\mathbf{e}$ and thus points toward the perigee of the orbit.
- $\hat{\mathbf{w}}$: is normal to the orbital plane and is therefore parallel to the specific angular momentum vector $\mathbf{h}$.
- $\hat{\mathbf{q}}$: obtained through right hand rule, such that the right-handed triad is completed.

Because this system maintains a fixed orientation with respect to perigee, it is particularly suitable for describing orbits with non-zero eccentricity. In the presence of a perfectly circular orbit, where the direction of perigee is not defined, the system loses physical meaning and alternative systems are typically used.

A.3 Earth-Centered Earth-Fixed (ECEF)

The ECEF RF (Figure A.3) is a non-inertial, Earth-fixed reference frame with its origin located at the Earth's center of mass. Unlike the ECI frame which is inertial, the ECEF x and y axes rotate with the Earth at its sidereal rotation rate $\boldsymbol{\omega}$. The fundamental plane of the frame is defined by the equatorial plane and the three orthogonal axes defining the frame are:

- $\hat{\mathbf{I}}'$: directed from the Earth's center toward the intersection of the equator and the Greenwich meridian (GM).
- $\hat{\mathbf{K}}'$: aligned with the Earth's rotation axis and pointing toward the Earth's Geographic North Pole.

- $\hat{\mathbf{J}}'$: completes the right-handed triad according to the right-hand rule, and lies in the equatorial plane.

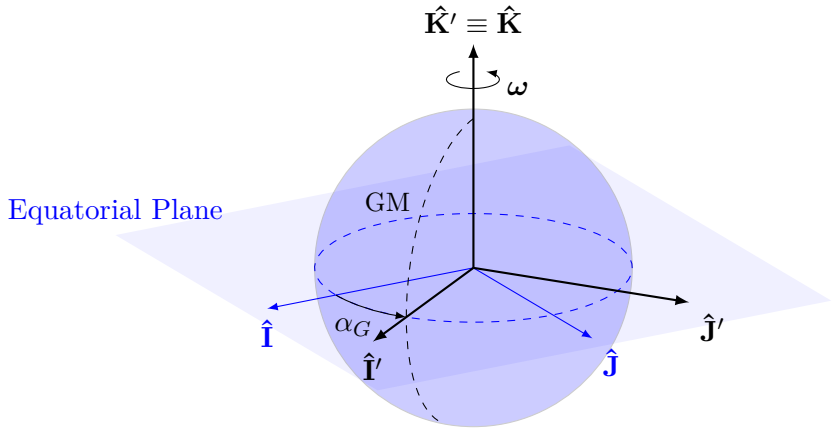
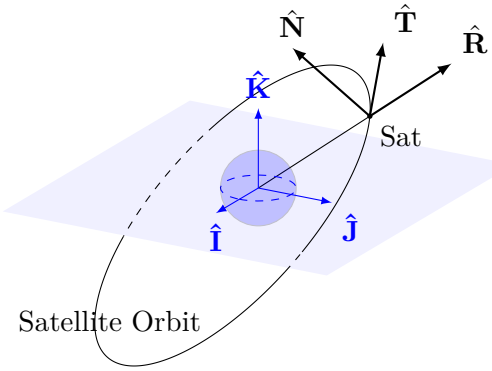


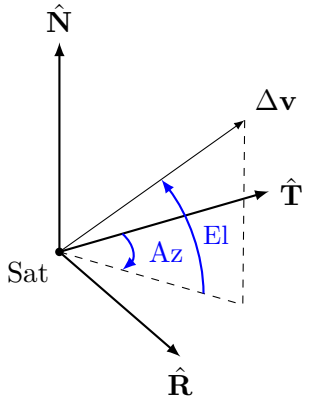
Figure A.3: Earth-Centered Earth-Fixed (ECEF) Reference Frame

Because the ECEF frame rotates with the Earth, the orientation of $\hat{\mathbf{I}}'$ and $\hat{\mathbf{J}}'$ changes over time with respect to inertial space. The angular displacement between the inertial $\hat{\mathbf{I}}$ axis (ECI frame) and the rotating $\hat{\mathbf{I}}'$ axis (ECEF frame) is quantified by the Greenwich sidereal angle α_G, which increases steadily at the Earth's rotation rate.

A.4 Radial-Transverse-Normal (RTN)



(a) 3D view: RTN frame along the orbit



(b) 2D view: RTN frame in the orbital plane

Figure A.4: Radial-Tangential-Normal (RTN) Reference Frame

The Radial-Transverse-Normal (RTN) RF (Figure A.4) is a non-inertial, satellite centered reference frame moving with it along its orbit. Its origin coincides with the instantaneous position of the satellite, and the three orthogonal axes are defined as follows:

- $\hat{\mathbf{R}}$: directed outward from the Earth toward the satellite, aligned with the position vector $\mathbf{r}$.

- $\hat{\mathbf{N}}$: is perpendicular to the instantaneous orbital plane, aligned with the satellite's specific angular momentum vector.
- $\hat{\mathbf{T}}$: lies in the orbital plane completing the right-handed triad. It points in the direction of the velocity vector only in case of circular orbit or at the at the apsides in an elliptic orbit.

Because the RTN RF is tied to the satellite rather than to inertial space, it provides a natural representation of the direction of impulsive Δv maneuvers. For this reason, the maneuver direction is often described using the azimuth and elevation angles defined in Figure A.4b. In this formulation, the azimuth (Az) is measured in the instantaneous RT plane, from the transverse axis $\hat{\mathbf{T}}$ to the projection of the $\Delta\mathbf{v}$ vector onto the RT plane. The elevation (El) quantifies instead the out-of-plane component of the burn and is defined as the angle between the RT plane and the $\Delta\mathbf{v}$ vector.

Appendix B

Atmospheric Density Model

Altitude Range h [km]	Base Altitude h_0 [km]	Nominal Density ρ_0 [kg/m^3]	Scale Height H
0–25	0	1.225×10^{0}	7.249
25–30	25	3.899×10^{-2}	6.349
30–40	30	1.774×10^{-2}	6.682
40–50	40	3.972×10^{-3}	7.554
50–60	50	1.057×10^{-3}	8.382
60–70	60	3.206×10^{-4}	7.714
70–80	70	8.770×10^{-5}	6.549
80–90	80	1.905×10^{-5}	5.799
90–100	90	3.396×10^{-6}	5.382
100–110	100	5.297×10^{-7}	5.877
110–120	110	9.661×10^{-8}	7.263
120–130	120	2.438×10^{-8}	9.473
130–140	130	8.484×10^{-9}	12.636
140–150	140	3.845×10^{-9}	16.149
150–180	150	2.070×10^{-9}	22.523
180–200	180	5.464×10^{-10}	29.740
200–250	200	2.789×10^{-10}	37.105
250–300	250	7.248×10^{-11}	45.546
300–350	300	2.418×10^{-11}	53.628
350–400	350	9.518×10^{-12}	53.298
400–450	400	3.725×10^{-12}	58.515
450–500	450	1.585×10^{-12}	60.828
500–600	500	6.967×10^{-13}	63.822
600–700	600	1.454×10^{-13}	71.835
700–800	700	3.614×10^{-14}	88.667
800–900	800	1.170×10^{-14}	124.64
900–1000	900	5.245×10^{-15}	181.05
1000–	1000	3.019×10^{-15}	268.00

Table B.1: Nominal atmospheric density ρ_0 and scale height H values for different altitude bands, used in the exponential density model [1]

Appendix C

Zonal Harmonic Accelerations

This appendix contains the analytical expression of the perturbing accelerations in ECI RF, due to the zonal harmonics J_4, J_5, and J_6 of the Earth's gravitational potential.

$$\mathbf{a}_{J_4} = \frac{15 J_4 \mu R_\oplus^4}{8 r^7} \begin{bmatrix} \left(1 - 14\frac{r_K^2}{r^2} + 21\frac{r_K^4}{r^4}\right) r_I \\ \left(1 - 14\frac{r_K^2}{r^2} + 21\frac{r_K^4}{r^4}\right) r_J \\ \left(5 - \frac{70 r_K^2}{3 r^2} + 21\frac{r_K^4}{r^4}\right) r_K \end{bmatrix} \tag{C.1}$$

with $J_4 = -1.6198976 \cdot 10^{-6}$.

$$\mathbf{a}_{J_5} = \frac{3 J_5 \mu R_\oplus^5}{8 r^9} \begin{bmatrix} \left(35 - 210\frac{r_K^2}{r^2} + 231\frac{r_K^4}{r^4}\right) r_I r_K \\ \left(35 - 210\frac{r_K^2}{r^2} + 231\frac{r_K^4}{r^4}\right) r_J r_K \\ \left(105 - 315\frac{r_K^2}{r^2} + 231\frac{r_K^4}{r^4}\right) r_K^2 \end{bmatrix} - \frac{15 J_5 \mu R_\oplus^5}{8 r^7} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{C.2}$$

with $J_5 = -2.2771610 \cdot 10^{-7}$.

$$\mathbf{a}_{J_6} = -\frac{J_6 \mu R_\oplus^6}{16 r^9} \begin{bmatrix} \left(35 - 945\frac{r_K^2}{r^2} + 3465\frac{r_K^4}{r^4} - 3003\frac{r_K^6}{r^6}\right) r_I \\ \left(35 - 945\frac{r_K^2}{r^2} + 3465\frac{r_K^4}{r^4} - 3003\frac{r_K^6}{r^6}\right) r_J \\ \left(245 - 2205\frac{r_K^2}{r^2} + 4851\frac{r_K^4}{r^4} - 3003\frac{r_K^6}{r^6}\right) r_K \end{bmatrix} \tag{C.3}$$

with $J_6 = 5.4068120 \cdot 10^{-7}$.

Appendix D

Close Approach Database

DEB	$\mathbf{r}_d(t_0)$ [km]	$\mathbf{v}_d(t_0)$ [km/s]	t_{tca} [s]	$d_{\min}$ [km]	P_c
001	[467.967, 4347.31, 5398.57]	[-6.41711, -2.86274, 3.58412]	6192	0.458118	2.2065×10^{-3}
002	[-755.284, 4144.85, 6202.49]	[-7.36232, -0.182678, 0.410769]	5356	1.26344	3.0543×10^{-4}
003	[-1904.75, 1994.89, 7381.44]	[-7.04445, -0.047336, -0.681761]	5211	0.426261	2.5268×10^{-3}
004	[5359.29, 2311.87, 3248.36]	[-3.23043, 7.13186, 0.551784]	6042	0.705294	9.6278×10^{-4}
005	[4274.93, 5317.92, 815.698]	[-3.69814, 2.22787, 6.40325]	3617	0.308667	4.5637×10^{-3}
006	[-10622.8, 4454.43, 2154.22]	[-2.50943, -4.24058, -0.889435]	5014	0.795060	7.6179×10^{-4}
007	[-4148.06, 6012.35, 2489.56]	[-6.62773, -2.14777, -2.24300]	5960	0.550005	1.5579×10^{-3}
008	[2953.63, 4287.55, 4543.31]	[-6.87245, 3.40725, 1.32521]	7102	0.747937	8.5855×10^{-4}
009	[669.964, 5959.04, 3397.74]	[-6.21070, 3.24817, -3.70314]	6386	0.553929	1.5368×10^{-3}
010	[-7031.88, 1358.74, 4553.26]	[-4.28207, -4.56624, -2.71694]	6120	1.40333	2.4796×10^{-4}
011	[5453.37, 2718.06, 2735.31]	[-2.92778, 7.21270, -1.05454]	5841	1.03339	4.5470×10^{-4}
012	[-11984.6, -309.393, -2855.91]	[-0.332975, -4.69999, -1.09920]	5673	1.18696	3.4568×10^{-4}
013	[2178.89, 5102.63, 3964.85]	[-7.01736, 0.682540, 3.62021]	6169	1.35246	2.6682×10^{-4}
014	[5411.05, 3832.52, -809.691]	[-4.47319, 6.45775, -0.443981]	5143	0.741512	8.7315×10^{-4}
015	[197.308, 7777.36, 3575.84]	[-5.14280, -1.06479, 3.84913]	4045	0.953253	5.3323×10^{-4}
016	[2090.75, 6237.39, 2179.02]	[-7.14589, 3.16362, -0.281288]	5621	1.80896	1.4962×10^{-4}
017	[-1437.03, -12673.9, -10168.5]	[3.64834, -1.05307, 0.471961]	6717	0.936942	5.5168×10^{-4}
018	[4026.22, 1367.86, 5320.43]	[-4.92978, 5.38774, 2.66847]	4319	1.75571	1.5880×10^{-4}
019	[-1999.30, 7965.72, 2979.38]	[-5.93725, -1.40163, 2.04166]	4494	0.639595	1.1644×10^{-3}
020	[1160.66, 7516.18, 2861.94]	[-5.23655, -0.260367, 4.31993]	4018	1.15698	3.6365×10^{-4}
021	[-3692.47, 2872.73, 7056.58]	[-6.36410, -0.126048, -1.98559]	5110	0.998829	4.8630×10^{-4}
022	[-9459.97, 1148.17, -1780.32]	[-1.40233, -5.00533, -3.24118]	6240	1.24136	3.1630×10^{-4}
023	[-7194.07, 4429.76, -3205.53]	[-3.28173, -4.70390, -3.00802]	6065	0.766715	8.1791×10^{-4}
024	[4564.47, 4448.35, 2129.98]	[-5.22228, 5.92276, -0.409791]	5729	1.47018	2.2605×10^{-4}
025	[-5970.87, 3966.29, 1411.66]	[-4.19417, -3.00250, -5.87193]	7169	1.11873	3.8868×10^{-4}
026	[3262.32, 4113.76, 4336.30]	[-6.41506, 4.49272, 0.777966]	6653	1.63528	1.8293×10^{-4}
027	[-5994.14, 5136.63, -1456.13]	[-3.78226, -3.76117, -4.83211]	6436	1.04779	4.4244×10^{-4}
028	[-12929.96, 598.238, -467.707]	[-0.528020, -4.35173, -1.36771]	5355	0.267145	5.8837×10^{-3}
029	[-10191.1, 5793.76, 7266.62]	[-1.92200, -3.92595, -0.846698]	4245	1.34212	2.7092×10^{-4}
030	[3396.73, 2063.33, 5587.52]	[-6.51703, 0.890902, 4.23147]	5331	1.44806	2.3297×10^{-4}
031	[-2478.74, 6282.04, 2163.45]	[-5.91708, -3.57872, 3.67412]	7185	0.716475	9.3370×10^{-4}
032	[3445.74, 5851.50, 1287.14]	[-6.42447, 3.99050, 1.74067]	4995	1.43082	2.3858×10^{-4}
033	[-2338.62, 7976.63, 2584.67]	[-5.99610, -1.47158, 1.90440]	4590	1.77137	1.5601×10^{-4}
034	[1933.17, 6891.35, -166.630]	[-6.97990, 2.70739, 1.17714]	4992	1.24107	3.1644×10^{-4}
035	[4358.52, 5174.26, 954.401]	[-5.42962, 4.35518, 3.47416]	4435	1.99063	1.2365×10^{-4}
036	[-1369.30, 3652.06, 6680.10]	[-7.17999, -0.075094, -0.244657]	5222	1.06419	4.2906×10^{-4}
037	[2110.58, 6280.26, 1762.71]	[-5.45176, 0.341094, 5.64246]	6642	1.10580	3.9772×10^{-4}
038	[-11891.8, 1625.03, 2552.59]	[-1.58451, -1.54806, -4.26859]	5160	0.850716	6.6709×10^{-4}
039	[-3974.20, -13981.6, -10144.1]	[3.08160, -1.35032, 1.13709]	6404	0.808155	7.3779×10^{-4}
040	[6151.74, 2212.51, 1242.77]	[-2.71644, 4.74435, 5.58222]	5685	1.12968	3.8125×10^{-4}

Table D.1: Training Close Approach Database

DEB	$\mathbf{r}_d(t_0)$ [km]	$\mathbf{v}_d(t_0)$ [km/s]	t_{tca} [s]	$d_{\min}$ [km]	P_c
001	[467.967, 4347.307, 5398.571]	[-6.41711, -2.86274, 3.58412]	6192	0.458118	2.2065×10^{-3}
002	[561.898, 6963.380, 1628.544]	[-7.43314, 1.62464, -0.731262]	5528	1.426482	2.4003×10^{-4}
003	[2694.093, 6060.478, 2416.162]	[-6.79123, 3.02703, 1.81285]	4952	0.172708	1.1934×10^{-2}
004	[5581.519, 2074.697, 2992.339]	[-3.29375, 6.91115, 1.73640]	6107	0.898907	5.9859×10^{-4}
005	[5048.038, 3076.906, 3144.339]	[-4.61490, 6.13075, 1.78429]	6179	0.685082	1.0189×10^{-3}
006	[-6850.207, 6952.588, 456.375]	[-4.63678, -3.38594, 1.31016]	4995	1.566881	1.9916×10^{-4}
007	[-1888.737, 5769.374, 3541.734]	[-7.42632, -2.57757, 0.681196]	7025	0.558891	1.5107×10^{-3}
008	[-3483.600, 3189.945, 6959.575]	[-6.44309, -0.179087, -1.84252]	5106	0.791221	7.6905×10^{-4}
009	[5198.445, 3513.116, 2324.196]	[-4.68029, 6.08768, 1.78158]	5748	0.592227	1.3512×10^{-3}
010	[3271.124, 5384.705, 2553.700]	[-6.44764, 2.24095, 3.95083]	6580	0.441737	2.3632×10^{-3}

Table D.2: Evaluation Close Approach Database

Bibliography

[1] D. A. Vallado, *Fundamentals of Astrodynamics and Applications*. Hawthorne, California: Microcosm Press, 2013.

[2] "Space debris mitigation requirements," Tech. Rep. ISO 24113:2024, International Organization for Standardization, Geneva, Switzerland, 2024.

[3] D. J. Kessler and B. G. Cour-Palais, "Collision frequency of artificial satellites: The creation of a debris belt," *Journal of Geophysical Research: Space Physics*, vol. 83, no. A6, pp. 2637–2646, 1978.

[4] "Iadc space debris mitigation guidelines," Tech. Rep. A/AC.105/C.1/2025/CRP.9, Inter-Agency Space Debris Coordination Committee (IADC), Vienna, Austria, 2025.

[5] "Esa's annual space environment report," LOG GEN-DB-LOG-00288-OPS-SD, ESA Space Debris Office, Darmstadt, Germany, Mar. 2025.

[6] "Iadc report on the status of the space debris environment," Tech. Rep. A/AC.105/C.1/2025/CRP.10, Inter-Agency Space Debris Coordination Committee (IADC), Vienna, Austria, 2025.

[7] "Nasa spacecraft conjunction assessment and collision avoidance best practices handbook," Tech. Rep. NASA/SP-20230002470 Rev 1, National Aeronautics and Space Administration (NASA), Washington, D.C., USA, 2023.

[8] S. Patnala and A. Abdin, "Spacecraft collision avoidance: data management, risk assessment, decision planning models and algorithms," in *Space Data Management*, pp. 15–45, Springer, 2024.

[9] A. Manis, J. A. Arnold, J. Murray, B. Buckalew, C. Cruz, and M. Matney, "An overview of ground-based radar and optical measurements utilized by the nasa orbital debris program office," in *2nd International Orbital Debris Conference (IOC II)*, 2023.

[10] ESA Space Debris Mitigation WG, "ESA Space Debris Mitigation Requirements," Tech. Rep. ESSB-ST-U-007 Issue 1, European Space Agency (ESA), 2023.

[11] "Managing mega-constellation risks in leo," tech. rep., Viasat Inc., Carlsbad, CA, USA, Nov. 2022. White Paper, updated November 2022.

[12] S. Aida and M. Kirschner, "Critical conjunction detection and mitigation," 2015.

[13] J. L. Foster and H. S. Estes, "A parametric analysis of orbital debris collision probability and maneuver rate for space vehicles. nasa," *National Aeronautics and Space Administration, Lyndon B. Johnson Space*, 1992.

[14] K. Chan, "Short-term vs. long-term spacecraft encounters," in *AIAA/AAS astrodynamics specialist conference and exhibit*, p. 5460, 2004.

[15] S. Alfano, "Relating position uncertainty to maximum conjunction probability," *The Journal of the Astronautical Sciences*, vol. 53, no. 2, pp. 193–205, 2005.

[16] R. P. Patera, "General method for calculating satellite collision probability," *Journal of Guidance, Control, and Dynamics*, vol. 24, no. 4, pp. 716–722, 2001.

[17] I. F. Stroe, A. D. Stanculescu, P. B. B. Ilioaica, M. Nita, A. Butu, D. Escobar, J. Tirado, B. Bija, and D. Saez, "Autonomous collision avoidance system," in *8th European Conference on Space Debris*, p. 109, 2021.

[18] D. T. Hall, "Determining appropriate risk remediation thresholds from empirical conjunction data using survival probability methods," in *2019 AAS/AIAA Astrodynamics Specialist Conference*, no. AAS 19-631, 2019.

[19] S. Dural, U. Tugcular, and B. Daser, "General collision avoidance maneuver decision algorithm," in *8th European Conference on Space Debris*, pp. 20–23, 2021.

[20] S. Aida, "Conjunction risk assessment and avoidance maneuver planning tools," 2016.

[21] L. K. Newman, A. K. Mashiku, M. D. Hejduk, M. R. Johnson, and J. D. Rosa, "Nasa conjunction assessment risk analysis updated requirements architecture," (Portland, Maine, USA), 2019.

[22] K. Merz, J. Siminski, B. B. Virgili, V. Braun, S. Flegel, T. Flohrer, Q. Funke, A. Horstmann, S. Lemmens, F. Letizia, F. Mclean, S. Sanvido, and V. Schaus, "Esa's collision avoidance service: Current status and special cases," (Darmstadt, Germany), ESA Space Debris Office, 2021.

[23] T. Pultarova, "Spacex starlink satellites made 50,000 collision-avoidance maneuvers in the past 6 months. what does that mean for space safety?." `https://www.space.com/spacex-starlink-50000-collision-avoidance-maneuvers-space-safety`, July 2024.

[24] SpaceX, "Spacex gen1 and gen2 semi-annual reports." Electronic Filing to the Federal Communications Commission, July 2025.

[25] H. G. Lewis and G. Skelton, "Safety considerations for large constellations of satellites," vol. 11, pp. 439–445, Elsevier, 2024.

[26] K. L. Hobbs and E. M. Feron, "A taxonomy for aerospace collision avoidance with implications for automation in space traffic management," in *AIAA Scitech 2020 Forum*, p. 0877, 2020.

[27] Z. Pavanello, L. Pirovano, R. Armellin, A. De Vittori, and P. Di Lizia, "A convex optimization method for multiple encounters collision avoidance maneuvers," in *AIAA Scitech 2024 Forum*, p. 0845, 2024.

[28] S. Dutta and A. K. Misra, "Convex optimization of collision avoidance maneuvers in the presence of uncertainty," *Acta Astronautica*, vol. 197, pp. 257–268, 2022.

[29] M. F. Palermo, P. Di Lizia, R. Armellin, *et al.*, "Numerically efficient methods for low-thrust collision avoidance maneuver design," in *8th European Conference on Space Debris, ESA/ESOC*, pp. 1–15, ESA, 2021.

[30] J. Gonzalo Gomez, C. Colombo, P. Di Lizia, *et al.*, "A semi-analytical approach to low-thrust collision avoidance manoeuvre design," in *INTERNATIONAL ASTRONAUTICAL CONGRESS: IAC PROCEEDINGS*, pp. 1–9, 2019.

[31] E.-H. Kim, H.-D. Kim, and H.-J. Kim, "A study on the collision avoidance maneuver optimization with multiple space debris," *Journal of Astronomy and Space Sciences*, vol. 29, no. 1, pp. 11–21, 2012.

[32] N. Zhang, Z. Zhang, and H. Baoyin, "Timeline club: An optimization algorithm for solving multiple debris removal missions of the time-dependent traveling salesman problem model," *Astrodynamics*, vol. 6, no. 2, pp. 219–234, 2022.

[33] J. D. Seong and H. D. Kim, "Collision avoidance maneuvers for multiple threatening objects using heuristic algorithms," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 229, no. 2, pp. 256–268, 2015.

[34] C. Mu, S. Liu, M. Lu, Z. Liu, L. Cui, and K. Wang, "Autonomous spacecraft collision avoidance with a variable number of space debris based on safe reinforcement learning," *Aerospace Science and Technology*, vol. 149, p. 109131, 2024.

[35] C. E. Oestreich, R. Linares, and R. Gondhalekar, "Autonomous six-degree-of-freedom spacecraft docking with rotating targets via reinforcement learning," *Journal of Aerospace Information Systems*, vol. 18, no. 7, pp. 417–428, 2021.

[36] M. Tipaldi, R. Iervolino, and P. R. Massenio, "Reinforcement learning in spacecraft control applications: Advances, prospects, and challenges," *Annual Reviews in Control*, vol. 54, pp. 1–23, 2022.

[37] D. Miller, J. A. Englander, and R. Linares, "Interplanetary low-thrust design using proximal policy optimization," in *2019 AAS/AIAA Astrodynamics Specialist Conference*, no. GSFC-E-DAA-TN71225, 2019.

[38] A. Zavoli and L. Federici, "Reinforcement learning for robust trajectory design of interplanetary missions," *Journal of Guidance, Control, and Dynamics*, vol. 44, no. 8, pp. 1440–1453, 2021.

[39] N. B. LaFarge, D. Miller, K. C. Howell, and R. Linares, "Autonomous closed-loop guidance using reinforcement learning in a low-thrust, multi-body dynamical environment," *Acta Astronautica*, vol. 186, pp. 1–23, 2021.

[40] C. J. Sullivan, N. Bosanac, R. L. Anderson, A. K. Mashiku, and J. R. Stuart, "Exploring transfers between earth-moon halo orbits via multi-objective reinforcement learning," in *2021 IEEE aerospace conference (50100)*, pp. 1–13, IEEE, 2021.

[41] B. Smith, R. Abay, J. Abbey, S. Balage, M. Brown, and R. Boyce, "Propulsionless planar phasing of multiple satellites using deep reinforcement learning," *Advances in Space Research*, vol. 67, no. 11, pp. 3667–3682, 2021.

[42] I. Newton, D. Bernoulli, C. MacLaurin, and L. Euler, *Philosophiae Naturalis Principia Mathematica*. Londini: Excudit G. Brookman; impensis T. T. et J. Tegg, 1833.

[43] N. Pavlis, S. Holmes, S. Kenyon, and J. Factor, "An earth gravitational model to degree 2160: EGM2008," in *Proceedings of the General Assembly of the European Geosciences Union*, (Vienna, Austria), Apr. 2008.

[44] J. R. Wertz, *Spacecraft Attitude Determination and Control*. Dordrecht, Holland: D. Reidel Publishing Company, 1978.

[45] D. G. Simpson, "An alternative lunar ephemeris model for on-board flight software use," 1998.

[46] D. B. Spencer and D. Conte, *Interplanetary Astrodynamics*. CRC Press, 1st ed., 2023.

[47] T. M. Mitchell, *Machine Learning*. New York: McGraw-Hill, 1997.

[48] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, Massachusetts: MIT Press, 2nd ed., 2018.

[49] D. Silver, "Reinforcement Learning: Lecture 2 – Markov Decision Processes." `https://www.davidsilver.uk/teaching/`, 2011. University College London, Reinforcement Learning Course.

[50] I. Charles, "A complete taxonomy of reinforcement learning algorithms: From basics to cutting edge." `https://medium.com/@itzcharles03/a-complete-taxonomy-of-reinforcement-learning-algorithms-from-basics-to-cutting-edge-dc51878caf77`, 2024.

[51] M. Yu, "A taxonomy of reinforcement learning algorithms." `https://medium.com/@ym1942/a-taxonomy-of-rl-algorithms-341fd1b4c659`, 2023.

[52] L. Graesser and W. L. Keng, *Foundations of Deep Reinforcement Learning: Theory and Practice in Python*. Boston, MA, USA: Addison-Wesley Professional, 2019.

[53] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[54] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[55] K. D. B. J. Adam *et al.*, "A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, vol. 1412, no. 6, 2014.

[56] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[57] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[58] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," *Advances in neural information processing systems*, vol. 12, 1999.

[59] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*, pp. 1889–1897, PMLR, 2015.

[60] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.

[61] L. Chen, X.-Z. Bai, Y.-G. Liang, and K.-B. Li, *Orbital Data Applications for Space Objects: Conjunction Assessment and Situation Analysis*. Singapore: Springer, 2021.

[62] A. K. Mashiku and M. D. Hejduk, "Recommended methods for setting mission conjunction analysis hard body radii," in *2019 AAS/AIAA Astrodynamics Specialist Conference*, no. GSFC-E-DAA-TN71115-1, 2019.

[63] European Space Agency, "ESA DISCOS: Database and Information System Characterising Objects in Space." `https://discosweb.esoc.esa.int/`, 2025.

[64] L. Baars and D. Hall, "Processing space fence radar cross-section data to produce size and mass estimates," in *2022 AAS/AIAA Astrodynamics Specialist Conference*, no. AAS 22-586, 2022.

[65] T. Smith, Z. Folcik, and R. Linares, "Challenges in orbital debris modeling: A comparative analysis of nasa sbm and space fence data," in *Advanced Maui Optical and Space Surveillance (AMOS) Technologies Conference*, p. 137, 2024.

[66] J. Zhu, F. Wu, and J. Zhao, "An overview of the action space for deep reinforcement learning," in *Proceedings of the 2021 4th international conference on algorithms, computing and artificial intelligence*, pp. 1–10, 2021.

[67] G. Falcone and Z. R. Putnam, "Deep reinforcement learning for autonomous aerobraking maneuver planning," in *AIAA Scitech 2022 Forum*, p. 2497, 2022.

[68] J. Blaise and M. C. Bazzocchi, "Space manipulator collision avoidance using a deep reinforcement learning control," *Aerospace*, vol. 10, no. 9, p. 778, 2023.

[69] S. Kazemi, N. L. Azad, K. A. Scott, H. B. Oqab, and G. B. Dietrich, "Satellite collision avoidance maneuver planning in low earth orbit using proximal policy optimization," in *2024 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–9, IEEE, 2024.

[70] G. Acciarini, F. Pinto, S. Metz, S. Boufelja, S. Kaczmarek, K. Merz, J. A. Martinez-Heras, F. Letizia, C. Bridges, and A. G. Baydin, "Spacecraft collision risk assessment with probabilistic programming," *arXiv preprint arXiv:2012.10260*, 2020.

[71] E. Stevenson, V. Rodríguez Fernández, H. Urrutxua, V. Morand, and D. Camacho Fernandez, "Artificial intelligence for all vs. all conjunction screening," 2021.

[72] L. Gremyachikh, D. Dubov, N. Kazeev, A. Kulibaba, A. Skuratov, A. Tereshkin, A. Ustyuzhanin, L. Shiryaeva, and S. Shishkin, "Space navigator: A tool for the optimization of collision avoidance maneuvers," *arXiv preprint arXiv:1902.02095*, 2019.

[73] N. Bourriez, A. Loizeau, and A. F. Abdin, "Spacecraft autonomous decision-planning for collision avoidance: A reinforcement learning approach," *arXiv preprint arXiv:2310.18966*, 2023.

[74] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of machine learning research*, vol. 22, no. 268, pp. 1–8, 2021.