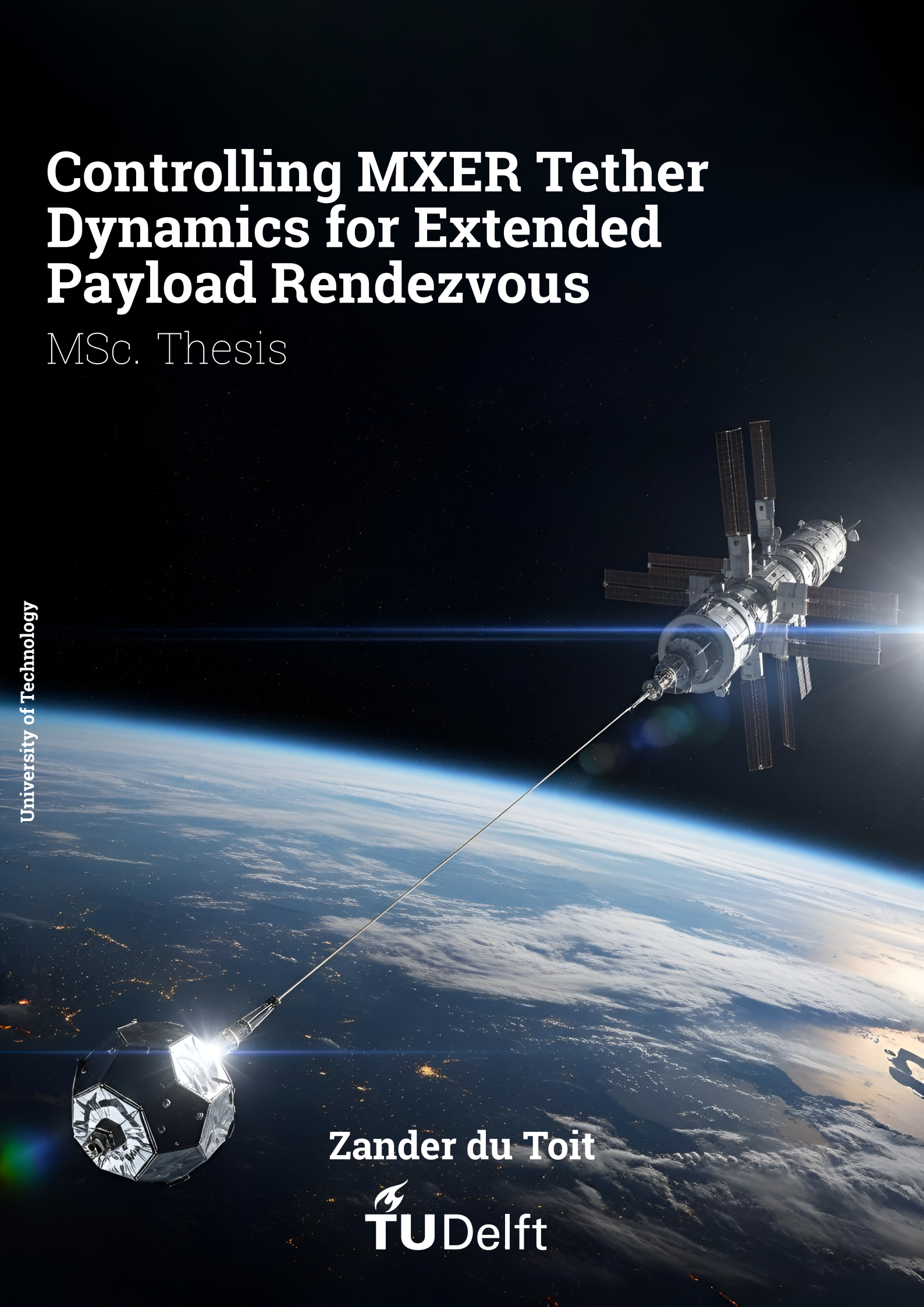
Controlling MXER Tether Dynamics for Extended Payload Rendezvous

MSc. Thesis



Zander du Toit

University of Technology

TU Delft

This page is intentionally left blank.

Controlling MXER Tether Dynamics for Extended Payload Rendezvous

MSc. Thesis

by

Zander du Toit

To obtain the degree of Master of Science

at the Delft University of Technology

to be defended publicly on Friday, 29 August, 2025.

Project Duration:	November, 2024 - August, 2025
Faculty:	Faculty of Aerospace Engineering, Delft
Thesis Committee:	Ir. M.C. Naeije (Supervisor)
	Dr. Ir. D. Dirkx (Chair)
	Dr. Ir. E. van Kampen (External examiner)

Cover:	AI-generated using Gemini (Modified)
Style:	TU Delft Report Style, with modifications by Daan Zwaneveld

A digital version of this thesis is available at http://repository.tudelft.nl/.

TU Delft

Preface

I would like to express my heartfelt gratitude to Mr. Marc Naeije for his invaluable guidance throughout this project, and for his continued support and tireless patience over the past months. His generosity extended well beyond the technical scope of this thesis, and I remain deeply appreciative of the opportunities and accommodations he made possible along the way.

I am grateful to Dr. Erik-Jan van Kampen for generously sharing his expertise on control theory and offering thoughtful advice that helped shape the direction of this work. I also wish to thank Dr. Dominic Dirkx for his insightful questions early in the project, which prompted critical reflection on and improvements to the project content.

To my friends and flatmates, thank you for enduring my endless ramblings about the thesis and for reminding me, time and again, that everything would turn out fine.

Finally, I want to thank my family for their constant love, support, and prayers from afar. I carry you with me always, and I hope to see you soon.

Zander du Toit
Delft, August 2025

Summary

This thesis addresses the critical challenge of extending the brief payload rendezvous window for Momentum Exchange with Electrodynamic Reboost (MXER) tether systems, a transformative technology for propellantless space transportation. The research systematically conducted a comparative analysis of three distinct actuator configurations using a two-dimensional rigid-body dynamical model, and evaluated a conventional optimal control method against a modern, model-free Reinforcement Learning (RL) algorithm.

The investigation definitively identifies the reeler actuator configuration as the most effective for extending the rendezvous window in an unconstrained dynamic environment. This configuration, which incorporates an intermediate reeling mass, achieved a threefold improvement, extending the uncontrolled rendezvous window of 0.6 seconds to 1.8 seconds. This duration, achieved within specified trajectory tracking tolerances of 10 m for position and 10 m/s for velocity relative to the payload, significantly outperformed both the baseline tip-reeling (0.8 s) and climber (1.0 s) configurations. This superior performance is primarily attributed to the reeler's enhanced control authority over the tether tip's velocity profile, enabling more effective counteraction of the characteristic V-shaped relative velocity curve inherent to rendezvous.

In the unconstrained scenario, both the conventional iterative Linear Quadratic Regulator (iLQR) and the model-free Soft Actor-Critic (SAC) RL agent successfully developed control policies, matching the 1.8-second rendezvous window extension. However, the SAC agent's policy exhibited less smooth, sporadic actuator usage, a trait undesirable in practical applications due to potential structural loads, component wear, and the excitation of unmodelled high-frequency wave dynamics.

The study of constrained control revealed the inherent difficulty of the problem. When realistic operational limits on tether tension, g-loads, and actuator usage were imposed, neither the Augmented-Lagrangian iLQR (AL-iLQR) nor the SAC-based controller could achieve a sustained rendezvous window. The AL-iLQR proved overly conservative, satisfying constraints but failing to exploit the system's full dynamic potential. Conversely, the SAC agent, guided by a simple penalty-based reward function, did not robustly enforce critical constraints, notably violating tension requirements, which would lead to system failure.

Verification and validation studies confirmed the fidelity of the rigid-body model. A variance-based sensitivity analysis highlighted tether length uncertainty as the dominant factor affecting rendezvous accuracy. Additionally, a comprehensive hyperparameter optimisation study for the SAC RL agent identified the learning rate and batch size as highly influential parameters for performance. A brief generalisation test also showed that the RL agent, trained on the reeler configuration, did not successfully generalise to the climber configuration, though its velocity control performance indicated potential for improvement.

Ultimately, this thesis successfully addressed its primary research questions, demonstrating how actuator configuration influences rendezvous window controllability and affirming RL's potential, albeit with current limitations concerning constraint satisfaction and control smoothness. All project goals, from model derivation and iLQR implementation to the deployment and evaluation of the SAC RL algorithm, were addressed, laying foundational groundwork for future advancements in MXER tether control.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

Abbreviation	Meaning
2D	Two-Dimensional
AL-iLQR	Augmented-Lagrangian Iterative Linear Quadratic Regulator
ANN	Artificial Neural Network
COM	Centre of Mass
DQN	Deep Q-Network
EDT	Electrodynamic Tether
GEO	Geostationary Earth Orbit
iLQR	Iterative Linear Quadratic Regulator
AL-iLQR	Augmented Lagrangian iLQR
LEO	Low Earth Orbit
LTO	Lunar Transfer Orbit
MET	Momentum Exchange Tether
MDP	Markov Decision Process
MXER	Momentum-Exchange Electrodynamic Reboost
RL	Reinforcement Learning
SAC	Soft Actor-Critic

Mathematical Notation

Notation	Meaning
$\mathbf{a}$	Vector or matrix quantity (bold symbol)
a	Scalar quantity (regular symbol)
$\dot{q}$	Time derivative of q
$\ddot{q}$	Second derivative of q with respect to time
$\|\mathbf{a}\|$	Norm of vector $\mathbf{a}$
$\langle \mathbf{a}, \mathbf{b} \rangle$	Vector dot product
$\mathbb{E}[\cdot]$	Expectation operator
$\nabla_\theta J(\theta)$	Policy gradient with respect to parameters θ

Latin Symbols

Symbol	Meaning	Units
a	Semi-major axis	m
d_i	Local distance of mass i	m
d_{CM}	Centre of mass offset	m
I_{CM}	Moment of inertia about COM	kgm^2
L	Tether length	m
L_i	Spooled tether length on mass i	m
m	Mass (generic)	kg
m_i	i-th point mass	kg
m_{tot}	Total system mass	kg
M	Total system mass (alt. notation)	kg
R_{CM}	Orbital radius of COM	m
R_p	Payload orbital radius	m
t	Time	s
T	Tether tension	N
W	Rendezvous window duration	s

Greek Symbols

Symbol	Meaning	Units
α	Orientation angle (tether dynamics)	rad
γ	RL discount factor	–
λ	Lagrange multiplier	–
μ	Augmented Lagrangian penalty weight	–
μ_E	Earth gravitational parameter	$\mathrm{km}^3/\mathrm{s}^2$
θ	Policy parameters (RL) or angular displacement	–
τ_g	Gravity-gradient torque	$\mathrm{N\,m}$
ω	Angular velocity (rotation rate)	$\mathrm{rad\,s}^{-1}$
ϕ	Parameters of value function in RL	–

Subscripts

Subscript	Meaning
0	Initial value
f	Final value
p	Payload
i	Instance or element index
tip	Tether tip
CM	Centre of mass

Reinforcement Learning Jargon Terms

Term	Symbol	Definition
Action	a_t	The decision made by the agent at time step t.
Actor	-	The component of the actor-critic system that represents the policy π_θ, responsible for selecting actions.
Advantage Function	A_t	A measure of how much better or worse an action a_t is compared to the critic's estimated value of state s_t, defined as $A_t = G_t - V_\phi(s_t)$.
Critic	-	The component of the actor-critic system that estimates the value function $V_\phi(s)$, providing feedback to the actor.
Discount Factor	γ	A scaling factor that determines the importance of future rewards compared to immediate ones.
Expected Return	$\mathbb{E}[G_t \mid s_t]$	The expected cumulative reward obtained by following a policy from a given state s_t.
Policy	$\pi_\theta(a \mid s)$	A mapping from states s to a probability distribution over actions a, parameterised by θ.
Policy Parameters	θ	The set of parameters that define the policy π_θ.
Return	G_t	The cumulative, discounted future reward from time t onwards, defined as $G_T = \sum_{k=0}^{T} \gamma^k r_k$.
Reward	r_t	The immediate scalar feedback received after taking action a_t in state s_t.
Soft Actor-Critic (SAC)	-	An off-policy, actor-critic reinforcement learning algorithm that maximises a trade-off between return and entropy to encourage exploration and stability during training.
State	s_t	The representation of the environment at time step t.
Value Function	$V_\phi(s)$	A function estimating the expected return from state s, parameterised by ϕ.

1

Introduction

1.1. Introduction to Momentum Exchange Tethers

1.1.1. Background and Motivation

Space tethers present a compelling alternative to traditional rocket propulsion, enabling novel approaches to space transportation and in-space operations. Their primary advantage lies in facilitating propellantless or near-propellantless manoeuvres, significantly reducing reliance on expendable propellants and thereby minimising launch mass [1]. This inherent advantage translates to potentially lower mission costs and increased payload capacity [2]. Two fundamental principles underpin the propellantless nature of tether propulsion: momentum exchange and electrodynamic interactions [3].

Momentum Exchange Tethers

Momentum Exchange Tethers (METs) are a subclass of tethered systems specifically designed to exploit the exchange of orbital momentum between a tether facility and a payload. By rotating the tether, typically in an elliptical orbit, a payload initially captured in low Earth orbit (LEO) can be released into a higher-energy orbit [2]. This momentum transfer effectively boosts or lowers the payload's orbit without expending onboard propellant, with the tether facility sacrificing some of its orbital energy in the process [4].

Electrodynamic Tethers

Electrodynamic tethers (EDTs) offer another propellantless propulsion method by exploiting the interaction between a conductive tether and a planet's magnetosphere [3]. As the tether moves through the magnetic field, an electromotive, Lorentz force is induced, which can be used to generate thrust or drag by controlling the current flow within the tether [2]. This interaction allows EDTs to raise or lower their orbits, change inclinations, and perform other orbital manoeuvres without consuming propellant.

Momentum Exchange and Electrodynamic Reboost Tethers

Combining the strengths of both METs and EDTs, Momentum Exchange and Electrodynamic Reboost (MXER) tethers offer a highly efficient and versatile approach to space

transportation [5]. These systems utilise rotational momentum exchange, akin to METs, for payload transfer, while incorporating electrodynamic thrusting, similar to EDTs, for orbit maintenance and reboost [6]. This synergistic combination minimises propellant consumption for payload transfer while enabling the tether facility to maintain its operational orbit, paving the way for reusable space infrastructure [7].

Operational Principle of MXER Tethers

The operation of a typical MXER tether boosting a payload's orbit is depicted in Figure 1.1. A payload is launched into a (typically circular) low Earth holding orbit. The MXER tether orbits at a higher altitude while rotating in a prograde direction. The tether rotates at a rate such such that the relative distance and velocity between the tether tip and the payload decreases to (near) zero at an instant in time. When the tether tip and payload have match in position and and velocity, the payload is captured by the tether tip becoming part of the tether. After capture, the payload rotates with the tether tip for half a rotation after which it is released.

The effect of releasing the payload from the tether tip at this point on the payload's orbital energy is twofold. First, the payload is released at an increased altitude, and second, the release takes place at a velocity higher than the payload's initial LEO velocity due to the prograde tether rotation. Thus both the potential and kinetic components of the payload's orbital energy is increased. The energy gained by the payload is of course equal to the energy lost from the tether through this momentum transfer. As a result the tether system falls to a slightly lower orbit, and needs to be reboosted to re-attain its original energy state. The reboosting is primarily achieved through electrodynamic thrusting against the geomagnetic field. Once reboosted, the tether is ready to repeat the process again for a new payload. Thus, the tether can effectively be thought of as a reusable, propellantless "third stage" that provides an additional velocity increase to the payload after its insertion into LEO by traditional rocket stages.

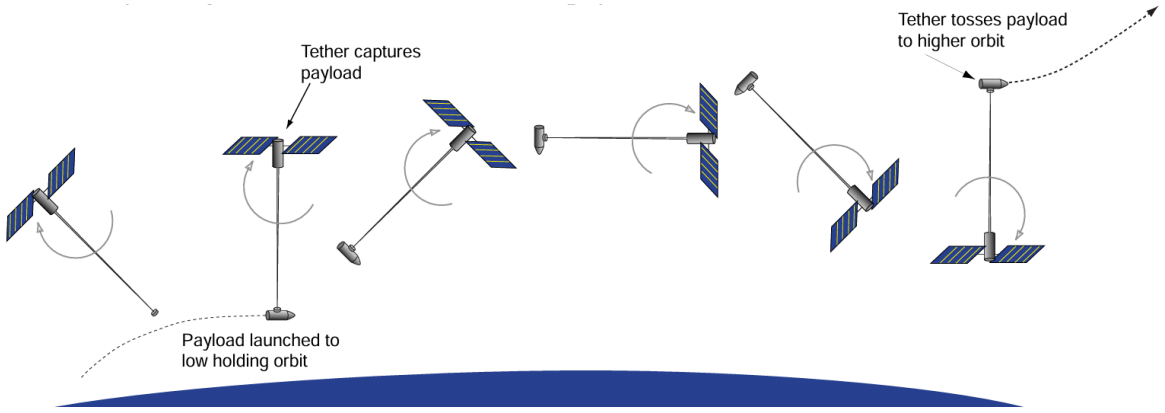


Figure 1.1: Operating principle of a momentum exchange tether [6]. Here the payload and the tether moves along their respective orbits from left to right. Not to scale.

Advantages and Applications of MXERs

MXER systems offer compelling advantages over traditional rockets, including reduced propellant requirements, system reusability, rapid transfer times, and mission versatility

[1]. The reduced propellant needs from the translate to substantial mass savings at launch and lower mission costs, making space missions more accessible and affordable [2]. The reusability of the tether facility significantly reduces recurring costs for payload transfers into higher orbits, offering a long-term cost-effective solution for such transfers with projected launch cost savings savings of 50-80% depending on the tether system [8]. Furthermore, the impulsive nature of momentum exchange enables rapid transfer times, comparable to or even surpassing those achievable with conventional chemical rockets [9]. MXER systems can be tailored for a wide range of mission scenarios, including efficient transfers between LEO and Lunar Transfer Orbit (LTO) [6], access to Geosynchronous Earth-Orbit (GEO) [10], and potentially even interplanetary missions [9].

1.1.2. MXER System Components and Architectures

Key Components and Operational Principal

MXER systems, building on the foundations of METs and EDTs, typically comprise a long, high-strength tether; a grapple mechanism at the tether tip for payload capture and release; and a counterweight and control station at the opposite end, responsible for maintaining tension and stability, and managing the system's dynamics and operations respectively [7]. An example of a MXER tether proposed for boosting payloads to Lunar transfer orbit (LTO) is shown in Figure 1.2. The tether serves as the primary link between the control station and the grapple mechanism, facilitating momentum transfer and electrodynamic interactions. The grapple mechanism, located at the tether tip, is crucial for successful payload capture and release, requiring robust design for handling velocity errors and ensuring secure docking. The control station houses the necessary hardware and software for monitoring the system's state, executing control laws, and managing power distribution. The counterweight, often a massive object or another spacecraft, is usually located near or coincident with the control station to concentrate the effect of their masses on the tether.

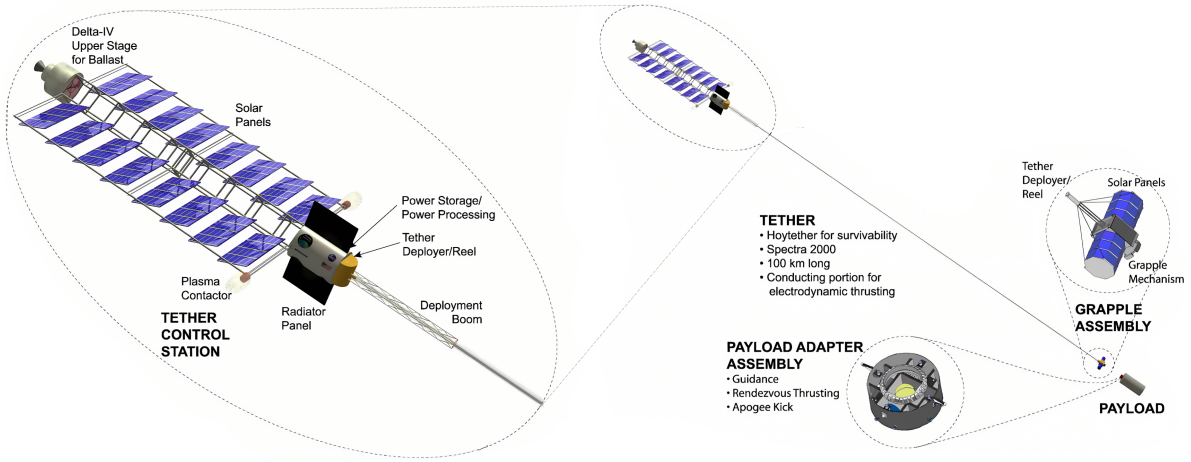


Figure 1.2: Example design of a tether boost facility. Adapted from [7]. Not to scale.

Multi-tether Systems

While a single tether as described in Section 1.1.1 can be useful as an effective "third stage" for sending payloads outward and away from Earth, the idea of momentum exchange tethers truly becomes compelling when the use of more than one tether, or multi-tether systems, are considered. One such example is the prosed Cislunar Tether Transport System, which envisions a two-tether configuration: one in elliptical Earth orbit and another in Lunar orbit [7]. This system is designed for monthly, round-trip payload transport between LEO and the lunar surface, leveraging momentum exchange for efficient and near propellantless transfer. The Earth-based tether picks up the payload from its low Earth holding orbit as described in Section 1.1.1 and tosses it into an LTO. Once the payload is sufficiently close, the Lunar tether captures it through a process that is essentially the reverse of the tossing process, and lowers the payload to either a low Lunar orbit (LLO), or into a decaying orbit that leads to payload touchdown on the Lunar surface. This process is depicted in Figure 1.3. As the Lunar tether captures the incoming payload, some of the payload's momentum is transferred to the tether, increasing its orbital energy. This increase in energy can then be used to toss other payloads from the Moon to the Earth, reversing the roles of the two tethers. Upon a payloads return to the Earth from the Moon, it is captured by the Earth-based tether, now transferring some of its momentum to the tether before being lowered to an LEO or decaying orbit.

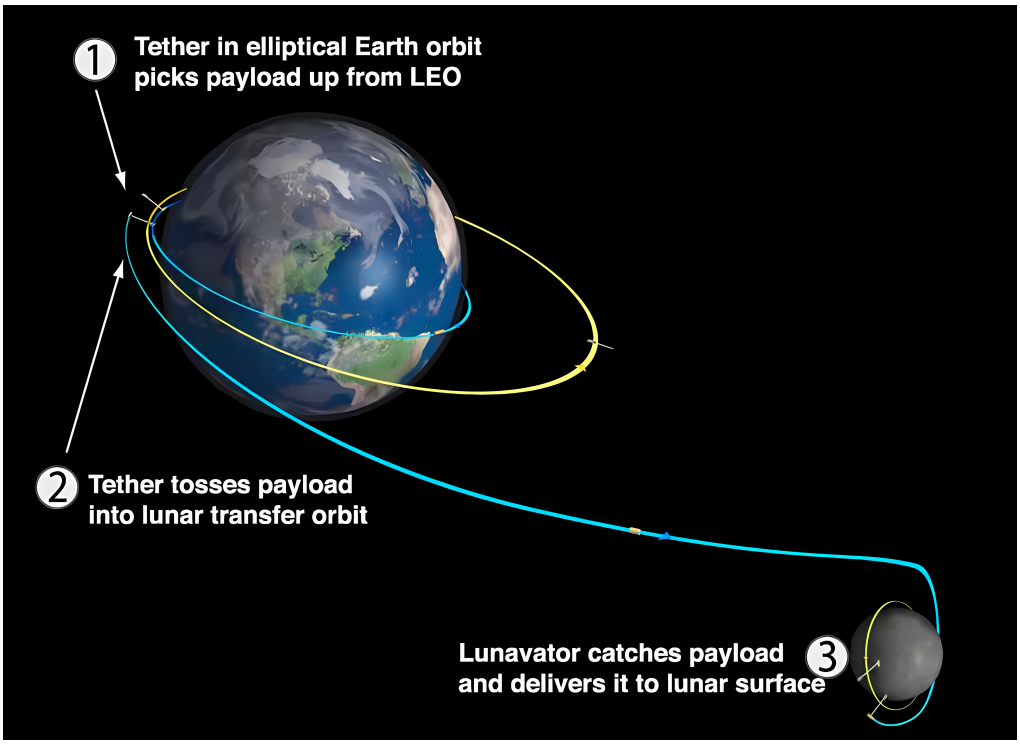


Figure 1.3: The Cislunar Tether Transport System concept. Adapted from [6]. Not to scale.

The Cislunar Tether Transport System could, in an ideal world, operate through momentum conservation alone, with the two tethers exchanging momentum through continued payload transfers. In reality, payloads may require velocity corrections during their

travels between these bodies, though it has been shown that such corrections are miniscule, on the order of 25 m/s [7]. Making use of electrodynamic thrust to reboost the Earth-based tether allows the cadence of tether-assisted transfers to be increased, as the Earth-based tether does not have to wait for an Earth-bound payload to regain previously transferred momentum and energy.

Besides the Cislunar Tether Transport System, other larger and more complex systems have been proposed for facilitating rapid Earth-Mars transport, for both crewed and non-crewed missions [9] [11]. These systems highlight the versatility of the MXER concept and its potential for addressing diverse mission needs in the Earth-Moon system and beyond.

Tether Material and Survivability

Upon first introduction to these tether systems, concern of feasibility might be expressed regarding suitable materials and the tether's survivability in the space environment. The choice of tether material is of course a critical design decision for MXER systems, influencing performance, durability, and feasibility. Key material properties include tensile strength for withstanding dynamic loads, density for minimising tether mass, elasticity for controlling wave propagation, and conductivity for electrodynamic thrust. Existing, high-performance engineering materials such as Spectra 2000, Zylon PBO and Dyneema 66 are promising candidates due to their favourable balance of these properties [7]. While these materials are not naturally electrically conductive and are primarily considered for their structural properties, a conductive tether can be achieved through interwoven conductive martial. Advanced materials like carbon nanotubes offer higher strength and inherent conductivity, but face challenges in scalability and cost [12].

Besides material selection, ensuring long-term survivability in the harsh space environment is equally crucial. A key design consideration for long-term viability is mitigating the effects of micrometeoroids and space debris impacts. The Hoytether™ design addresses this by using an open net structure with redundant load-bearing lines [13]. This innovative architecture allows the tether to reliably survive impacts from objects up to 30 cm in size [11], as the interconnected secondary lines redistribute loads around damaged primary segments [13]. This distributed design, with slack secondary lines initially unstressed, provides a high margin of safety, allowing the tether to operate under high stress while maintaining structural integrity for tens of years [14]. This survivability is crucial for long-term economic viability, enabling the tether to handle frequent traffic and maximise its operational lifetime.

1.2. Thesis Focus and Structure

The present work aims to answer two fundamental research questions. The first concerns the influence of different mechanical actuator configurations on the controllability of a spinning MXER tether's rendezvous window. The second investigates the effectiveness of modern Deep Reinforcement Learning as a control strategy for the MXER rendezvous problem, comparing it directly against a conventional (near) optimal control baseline. To address these questions, this research is centred on a comparative study that is presented as a standalone research paper, supported by extensive background literature

and detailed validation and analysis. As a result, this thesis consists of this central research paper and its accompanying material.

The accompanying material aims to provide a discussion of topics that are supportive to the research, and to offer greater depth in the discussion of selected topics where such detail would not be warranted within the page limits of a conference paper.

The background for the research is presented across two chapters. Chapter 2 provides a comprehensive literature review on Momentum Exchange Tethers. It begins with a discussion of tether dynamics and common mathematical modelling approaches in Section 2.1, followed by a review of the control challenges associated with payload rendezvous and capture and an overview of existing control methods in Section 2.2. Chapter 3 presents a literature review of Reinforcement Learning, establishing the theoretical foundation for the DRL-based controller. It covers the core concepts of RL, the definition of state, action, and reward spaces, different agent architectures, and the specific model-free algorithm relevant to this work.

Chapter 4 synthesises the preceding literature reviews to formally establish the Research Gap and Opportunity. This chapter identifies the specific gaps in the current body of knowledge that this thesis aims to address and logically derives the research questions and project goals that guide the subsequent work.

Chapter 5 contains the core research contribution of this thesis, presented in the form of the paper titled *Conventional and Reinforcement Learning Control of MXER Tether Dynamics for Extended Payload Rendezvous*. This paper details the comparative analysis of the three actuator configurations and the two control methodologies, presenting the primary results and findings of the research. The paper has been submitted to the proceedings of the 2025 International Astronautical Congress (IAC), and will be presented at this congress in September 2025.

Chapter 6 details the Verification, Validation, and Robustness analysis that underpins the results presented in Chapter 5. This chapter provides a unified discussion of the measures taken to ensure the correctness of the methods, including the numerical integrator selection, validation of the rigid-body model, and verification of the selected control methods in Section 6.1. It also presents the sensitivity and tuning analysis, covering the impact of physical model parameter uncertainty and the process for determining the RL agent's hyperparameters in Section 6.2.

Finally, Chapter 7 concludes the thesis by outlining the main results and offering recommendations for future work. Section 7.1 summarises the key conclusions. Section 7.2 provides actionable recommendations for both future mission design and further research. The chapter, and the thesis, are concluded with a final assessment of the research questions and an analysis of the project's compliance with its stated goals.

2

Momentum Exchange Tether Literature Review

2.1. Dynamics and Modelling of MXER Tethers

2.1.1. Tether Dynamics Considerations

Orbital Motion and Tether Rotation Coupling

The centre-of-mass (COM) of a tethered system follows a standard Keplerian orbit, largely independent of tether rotation. In rigid-body tether models this is explicitly verified: for example, NASA simulations confirm that the centre of mass (COM) translation obeys classical Kepler motion [15]. The rotating tether end-masses thus move about the COM with little effect on its overall orbital trajectory. In practice the in-plane tether rotation (rotation in the orbital plane) typically dominates the motion by design, with any out-of-plane tilts or nutations being much smaller under nominal conditions. Gravitational-gradient torque tends to align long non-rotating tethers roughly along the local vertical direction, coupling librational motion of the tether to the orbital motion. Some studies propose the active utilisation of gravity-gradient forces and torques for controlling both the rotational and center-of-mass orbital parameters of spinning tethers, typically by varying the tether length at appropriate times [6], [8].

External Perturbations

Long tethers are subject to various environmental perturbations. Key forces include:

Higher-order gravity effects (oblateness and harmonics)
In addition to the orbital precession caused by Earth's oblateness (and higher-order gravity modelling terms), the non-uniformity of Earth's gravity can induce slow tether librations in both the in- and out-of plane directions (orbital plane), though to a lesser extent for the latter [16]. For long tethers gravity-gradient effects also manifest as rotation-rate enhancing or retarding torques for the Earth-approach and departure phases of a tether's rotation respectively.

Aerodynamic drag
The orbit-lowering effect of drag on satellites in low Earth orbit is well known. In LEO

environments, the extended tether may experience significant drag depending on its length and altitude. For example, the drag on the lower 10 km of a tether contributes significantly to the overall system's drag, and this tether portion has about 20 times the drag of its attached satellite [17]. Besides orbital decay, aerodynamic drag can induce out-of-plane librations for tethers in inclined orbits due to Earth's equatorial oblateness and the atmosphere's rotation following that of Earth [17]. For tethers with notably low orbital perigees, drag may lead to heating and erosion of the tether surface, which impacts the service-life of the tether. For these combined reasons, the perigees of MXER tether orbits are usually chosen to be as high as possible, while still satisfying the payload's rendezvous orbit requirement. For typical MXER tether parameters, simulations have shown that a 1% variation in aerodynamic forces (due to aerodynamic modelling parameter uncertainties) results in a shift in the tether tip's predicted position of about 2 metres after completing one full orbit [18].

Geomagnetic forces
A conducting or charged tether moving through Earth's magnetic field experiences Lorentz forces. These forces can be used for electrodynamic boosting, but may also perturb the tether motion. During the perigee approach for payload capture the tether system would not normally apply any electrodynamic thrusting forces [18], rather relying on pure dynamical motion and possibly tether reeling for tip control. Despite this rule of thumb, a minor electrical leakage may occur causing a small current to flow in the conductive tether portions thereby causing an unwanted disturbance force. Such a hypothetical situation was simulated and results showed that such a leak causes a shift in tether tip position of a about 3 metres after one full orbit [18].

Solar radiation pressure (SRP)
Long tethers have a non-negligible surface area, and thus SRP can produce small transverse torques. Simulations of typical MXER tether systems show tether tip shifts of 1 to 4 metres after one orbit when taking variations in SRP forces into account. These simulations considered force variations up to 5% due to geometric and reflective parameter uncertainties [18].

Mass loss
Long-term exposure to atomic oxygen in LEO can erode the tether causing mass loss and degradation of tether material properties. Some experiments have investigated the effectiveness of tether material coatings for the protection against reactive atomic oxygen environments. Two coatings, a copolymer and metallised nickel, proved promising [19]. In addition to the reactions with atomic oxygen, other processes like outgassing, sublimation and micro-meteorite damage can result in tether mass loss. These combined effects have been estimated to result in 5 to 150 grams of mass loss per day, which causes the tether's centre of mass to shift slowly [18]. Fortunately, from a dynamics perspective these effects have a minimal impact on actual tether tip displacement for rotating tethers, with estimates suggesting displacements less than a millimetre [18].

Thermal effects
Cyclic heating causes thermal expansion/contraction of the tether material. Over a long tether this can produce length changes (or tension changes) and can excite low-frequency oscillations. Tether length fluctuations lead to a fluctuating moment of inertia, and thus

tether rotation rate. Estimates for a Zylon tether, a commonly proposed tether material, predict a tether tip drift of 163 metres per degree Kelvin from its expected position [18].

Material creep
Polymeric or metallic tethers under sustained tension exhibit viscoelastic creep, gradually lengthening or experiencing stress relaxation over time. This material creep can be significant for typical MXER systems, with estimates placing the tether elongation rate at around 2 metres per day for Zylon tethers. The resulting tether tip drift after one orbit is non-negligible, at about 40 metres from its anticipated position [18].

Clearly, thermal effects and material creep have a much greater impact on the predicted tether tip position than the other perturbations. While the smaller environmental effects can be partially accounted for in models and uncertainty analyses, the large influence of thermal effects and material creep will need to be actively monitored in a real tether system with accurate state estimation.

Tether Elasticity and Wave Propagation

Long tethers effectively behave like elastic strings under tension, supporting longitudinal (axial) waves and transverse (flexural) waves that propagate at different speeds. The different propagation speeds of these waves lead to "stiff" dynamics, with the longitudinal wave speed far exceeding that of the transverse waves [18]. These travelling waves lead to spatio-temporal length and tension variations along the tether. When the tether length or tension changes (e.g. during deployment or after capture), pulse-like waves travel along the tether causing oscillatory motion of the tip masses or payload, impacting capture or re-boost dynamics.

Managing and eliminating these wave oscillations is non-trivial as tethers typically have variable degrees of natural damping in the out-of-plane, in-plane transverse and longitudinal directions. (As a brief aside, here "plane"refers to the orbital plane, though for many MXER tether proposals, the tether's rotation plane is chosen to be the same as the orbital plane. Thus going forward, "in-plane" will refer to both the orbital and rotation planes.) To shortly mention these variable degrees of natural damping:

- Out-of-plane oscillations are predominantly decoupled with in-plane motions and not subject to natural damping [3][20].
- In-plane transverse oscillations are weakly coupled with longitudinal oscillations, and natural damping of these oscillations only occur over exceptionally long timescales of months to years [3].
- Longitudinal oscillations are damped out much faster than their transverse counterparts, with roughly threefold amplitude decays in as little as 10 minutes for typical tether systems [3]. This effective natural damping is attributed in part to the inter-fibre friction present in braided-fibre tethers [3].

Clearly, additional damping strategies are necessary to effectively manage these tether oscillations. Some suggested damping options include longitudinal dampers, carefully timed and tuned electrodynamic thrusting [21], actively moving masses [3] and reeling the tether in an out as necessary [8].

For analysis of their string-like behaviour, tethers can be modelled in a variety of ways,

depending on the desired level of accuracy and insight gained from the model. Analysis approaches vary from simpler vibrating-string (chord) equations [15], or to more complex modal decomposition approaches [18]. The next section looks at different approaches for modelling the dynamics of space tethers.

2.1.2. Mathematical Modelling Approaches

There exists a variety of tether models in literature, each with its benefits and drawbacks. The main categories to which these models may fall into, some being mutually inclusive, are rigid-body models, flexible and extensible models, lumped-mass models, and continuous models. These models are discussed in the next sections, and summarised in Table 2.1.

Rigid-Body Models

Rigid-body models, as the name implies, treat the tether system (in-whole or in part) as a rigid, non-deformable structure. These models capture the fundamental dynamics of rotating and librating tethers, providing useful and insightful results without resorting to extreme computational requirements or increased mathematical complexity. Here, wave effects are of course not captured. Models in this category are often used for the development and testing of system controllers, control laws and tether configurations [22].

Flexible and extensible models

On the opposite side of the spectrum, are flexible and extensible models. These models can capture the transverse deflections of flexible tethers, or the longitudinal extensions and contractions, or a combination of both of these effects. As a step up from rigid-body models, considerations could for example be limited to the axial direction to study the effects of longitudinal elasticity, without the complex transverse deformation behaviour [20]. Alternatively, the extensible models can be expanded to transverse deflections in two or three dimensions to capture in-plane and out of plane effects [23]. Models in this category are mathematically more complex and typically dependent in-part on hyperbolic partial differential equations (PDEs) for the description of wave dynamics [18]. These models can range from computationally manageable (at or near real-time simulation) for purely extensible models to computationally complex and demanding for full three-dimensional, some of which are only suitable offline (non-real-time) simulation. This computational requirement is largely due to the next two categories.

Lumped-Mass Models

For lumped-mass models, the tether system's mass is represented by discrete "effective" masses. This may be as simple as dumbbell models, which group the tether system's mass into two masses at each end of a massless the tether [17], or more advanced discretisation methods, where the tether is modelled as a series of massive nodes connected by springs and possibly dampers [23]. Approaches of the latter kind can become computationally demanding for systems that strive for high accuracy, since a high number of discretisation nodes are necessary to capture highly accurate dynamics [18].

Continuous Models

Continuous models maintain the continuum nature of the tether, opting to rather solve the wave-like PDEs with finite element methods, partial dimension discretisation (like the method of lines), or modal decomposition approaches. Significant work has been done on modelling flexible and extensible tethers using eigenfunction expansion with extreme accuracy. With a sufficient number of eigenfunctions capturing tether modes, the tether's position can be predicted with 9-digit accuracy, which equates to sub-metre accuracy on tether lengths often in excess of 90 km [18]. This approach not only benefits from high accuracy, but also computational efficiency, being thousands of times faster than conventional discretisation methods such as the spring-damper lumped-mass method [18]. This computational efficiency makes it suitable for both online and offline use. The significant accuracy and speed come at the expense of mathematical complexity and the requirement of highly precise model inputs.

Modelling Methods Comparative Summary

Table 2.1: Comparison of Tether Modelling Approaches

Model Category	Description	Key Features	Benefits	Drawbacks	Computational Cost	Typical Applications
Rigid-Body Models	Treats tether (whole or in part) as a non-deformable structure.	Inextensible, massless or fixed-length tether; Captures spin and libration dynamics	Mathematically simple; Low computational burden; Enables rapid controller design and testing	Cannot represent elastic oscillations or slack effects	Very low	Control law development; Preliminary dynamic studies
Flexible & Extensible Models	Models tether elasticity: longitudinal extension/contraction and/or transverse deflection.	Axial-only wave-like PDE for pure extensibility; Full 2D/3D transverse-axial coupling via hyperbolic PDEs	Captures axial and/or bending waves; Adjustable fidelity (axial only, 2D, or full 3D)	Mathematically complex; 3D PDEs may be too heavy for real-time without simplification (depends on lumped mass vs continuous approach)	Medium to high (depending on fidelity)	Targeted simulation of wave dynamics; Detailed capture-phase studies
Lumped-Mass Models	Discretises tether into nodes (point-masses) connected by springs/-dampers, from dumbbell to fine chains.	Adjustable number of masses; Spring/-damper elements represent elasticity and damping	Balances fidelity and simplicity; Handles nonlinear slack and damping; Relatively straightforward implementation	Accuracy requires many nodes (numerical stiffness); Computation scales with DOF	Moderate (increases with more with nodes)	Mid- to high fidelity simulations; Controller-in-the-loop testing; Model validation
Continuous Models	Maintains continuum by solving wave PDEs via FEM, method of lines or modal (eigenfunction) decomposition.	Finite-element or modal decomposition; Eigenfunction expansions for high-order modes	Highest positional fidelity (sub-metre over 90 km tethers); Very efficient modal solutions	High mathematical complexity; Demands precise model inputs (mode shapes, material properties, tether and environment states)	Low to moderate (for modal) or high (for FEM)	Offline high-precision design; Model validation; Some online use via modal reduction

2.2. Control Challenges and Existing Control Methods for Space Tethers

2.2.1. Rendezvous and Payload Capture Challenges

The success of momentum-exchange tethers hinges on overcoming complex rendezvous and payload capture challenges. This section will delve into how rendezvous is defined, the conditions required for a successful capture, the inherent difficulties posed by the short rendezvous window, and other significant challenges associated with tether-payload rendezvous.

Defining the Rendezvous Problem

In the context of momentum-exchange tethers, rendezvous is fundamentally different from conventional orbital docking and rendezvous. Instead of matching position, velocity, and acceleration over an extended period, tether-payload rendezvous requires matching the position and velocity of the tether tip and the payload at one specific point in time and space [8]. The acceleration of the tether tip and the payload are *not* matched [5]. Rather, after capture the tether accelerates the payload to match its rotational motion, thereby facilitating a transfer of momentum from the tether system to the payload.

Conditions required for the tether tip to rendezvous with the payload include:

- **Orbital Configuration** - The tether facility is typically placed in an elliptical orbit, and its rotation is timed so that the tether is oriented vertically below the central body and swinging backward when the facility reaches perigee [8]. The payload is usually in a circular low Earth orbit. See Figure 2.1 for a depiction of this configuration.
- **Velocity Matching** - The angular rotation rate of the tether is set so that the tether tip velocity is the difference between the orbital velocity of the tether's centre of mass (COM) and the payload's orbital velocity [24]. This allows the tether tip and payload to instantaneously match position and velocity at (or near) perigee as depicted in Figure 2.1b.
- **Orbital Resonance** - To ensure multiple rendezvous opportunities and minimise manoeuvring, the semi-major axis of the tether facility's orbit can be chosen so that its orbital period is a multiple of the payload's orbital period [6]. For example, a 5:2 ratio yields a capture opportunity every two orbits for the tether, or every five orbits for the payload. For a payload in LEO with an orbital period of about 90 minutes, a 5:2 ratio would allow for a capture opportunity roughly every 3 hours and 45 minutes. During this time the tether needs to be controlled to synchronise its rotation state at the predicted time of capture with the required rotation state. In addition to the capture considerations, the target trajectory's requirements also need to be considered when planning a capture and launch. For example, for a trans-lunar injection, the payload may need to be captured and launched such that it arrives at the Moon during the lunar crossing of the ascending or descending node.

To better understand the tether-payload capture behaviour, the approaching tether can

be imagined as a wheel rolling around the payload's orbit with the tether tip making momentary contact with the payload [5]. This is the ideal moment to perform the capture procedure. From the payload's perspective, the tether tip descends rapidly, comes to a stop for an instant, and then rapidly ascends again.

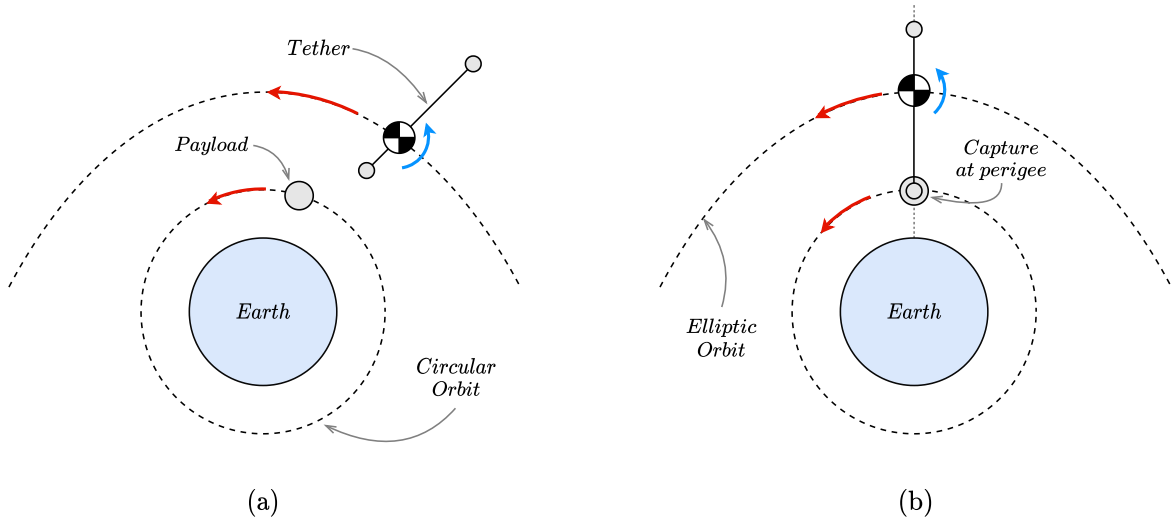


Figure 2.1: Orbital configuration of payload and tether system: (a) approach, and (b) capture at perigee. Not to scale.

The Short Rendezvous Window

Orbital rendezvous and docking for conventional target and chaser spacecraft occurs slowly, over minutes or even hours, with extremely high precision (millimeter positional error, millimeters per second velocity error [25]). For tether systems, rendezvous with the payload must happen nearly instantaneously, typically in a window of only a couple of seconds [26]. This is about two orders of magnitude less time than traditional space docking scenarios [8].

The same acceleration mismatch that allows for momentum transfer from the tether to the payload is the key reason for this short rendezvous window. The tether's tip experiences significant centrifugal accelerations (e.g. 1 - 3 g [7]) due to the tether's rotation relative to its orbital frame, while the payload is in a free-fall condition. This acceleration disparity means that under ideal conditions, the relative position and velocity between the tether tip and payload only reaches zero at a single point in time. The tether tip's non-Keplerian trajectory further complicates the position and velocity synchronisation, making the time window for rendezvous very narrow.

Rendezvous Challenges

A number of challenges need to be overcome to make rendezvous in such a short window possible.

- **High accuracy requirements** - To know the tether tip's position and velocity to a sufficient degree for payload capture requires extremely accurate state estimation and propagation of the tether trajectory (and underlying dynamics) [8], as well as

accurate manoeuvring of the payload to be in the exact right place at the right time with the correct velocity.

- **Limited time for manoeuvres** - The very short window means there is minimal time for a terminal guidance system to make necessary manoeuvrers and achieve capture. Thus responsive and quick-acting actuators are be needed for controlling and guiding the tether tip to the desired position ahead of time. This again ties into the high accuracy requirements, as the uncertainty in both the actuation levels as well as the tether system's response should be minimised.
- **Error tolerance** - Realistically, matching position and velocity of both the tether tip and payload at the exact time is near-impossible. Thus the capture process should have a degree of error tolerance. Both the tether system and the payload may cooperate to improve this error tolerance, but for the sake of minimising propellant use, the tether may be expected to be the primary source of error tolerance. Specifically, the brunt of this responsibility falls on the capture mechanism located on the tether tip, which must be tolerant of several meters of positional error and a few meters per second of velocity error [27].
- **Payload scheduling flexibility** - Perhaps more of a limitation than a challenge to overcome, tether transport systems have a relatively inflexible schedule for payload transfers due to the precise orbital mechanics involved and short launch windows [8]. This requires accurate timing of payload orbits if propellant use is to be minimised.

2.2.2. Review of Existing Control Methods

A wide array of control strategies and actuation methods has been investigated to manage the complex dynamics of space tethers. These methods range from simple, predefined deployment profiles to sophisticated optimal control and non-linear feedback laws. This section provides a review of the prominent control methodologies and actuation techniques found in the literature, establishing the context for the control approaches selected and evaluated in this work.

Conventional Control Methods

The control of tether systems, particularly for rendezvous, has been approached from several theoretical and practical angles. These can be broadly categorised into predefined strategies, linear and non-linear feedback control, and optimal control frameworks.

Predefined and Simple Feedback Strategies For less dynamically complex manoeuvres, such as initial deployment, simple open-loop or feedback strategies have proven effective. For instance, the SEDS missions successfully utilised braking-only deployers, where the tether is passively pulled out by the gravity-gradient force while controlled friction is applied to manage the deployment rate [22]. Another study considered a deployment method that involves the tether tip mass releasing either itself or a separate a grapple fixture at a controlled unspooling rate for extending the rendezvous window. The tip or grapple mechanism is deployed at minimum deployment tension with a predefined braking duration, placing the grapple in a nearly free-fall trajectory that closely

matches the payload's trajectory thereby extending the time time in close proximity to the payload [28].

Linear and Non-linear Control Given the inherent non-linearity of tether dynamics, both linear and non-linear control methods have been explored.

- **Linear Control** approaches are often considered suitable for the terminal phase of rendezvous, where system states are close to a desired reference trajectory and dynamics can be reasonably linearised [26]. Linear quadratic feedback controllers (LQR) have been applied to stabilise tether attitude and minimise deviations from a reference path, particularly during deorbit manoeuvres [29]. However, the effectiveness of linear controllers diminishes as orbital eccentricity increases or when large initial errors are present [27].
- **Non-linear Control** methods are generally better suited to the highly non-linear dynamics of spinning tethers in elliptical orbits. Techniques such as non-linear model predictive control (MPC) and non-linear receding horizon controllers [30] have been employed to manoeuvre a tether tip to a desired libration cycle for rendezvous. These methods can handle the system's complex behaviour more robustly than their linear counterparts. Other approaches, such as fuzzy logic-based feedback, have also been used to manage large, non-linear motions during deployment [31].

Optimal Control Optimal control theory is a powerful framework for tether control, as it can determine control laws that manoeuvre the system between initial and terminal states while minimising a specified cost function. It has been used to generate open-loop control laws for both pre-capture libration pumping and post-capture damping of oscillations [32]. A variety of cost functions have been employed, with objectives such as minimising control effort, power consumption, tether tension, or libration angle. For elastic tethers, minimising strain or its derivatives is often prioritised to ensure smooth dynamics and avoid structural damage. To solve these complex optimisation problems, numerical techniques like direct transcription methods (e.g., the Legendre pseudospectral method) are commonly used [32], [33].

Other Control Strategies Several other strategies have been developed to address specific challenges. To manage the high accuracy requirements of rendezvous, cooperative manoeuvres involving both the tether system and the payload are often necessary [26]. Furthermore, to account for inevitable tracking errors, error-tolerant capture mechanisms, such as using a large open tether loop with a harpoon, have been proposed to accommodate position and velocity misalignments [34].

Actuation Methods

The implementation of any control law relies on effective actuation. Several methods have been proposed and developed to influence tether dynamics, with the most prominent being electrodynamic thrusting and mechanical methods that alter the system's moment of inertia.

Electrodynamic Thrusting Electrodynamic tethers utilise a conductive wire that, when carrying a current through a planetary magnetic field, generates a Lorentz force. This force is perpendicular to both the tether and the magnetic field lines, providing a means of propellantless thrust [22]. This method is primarily used for orbit boosting or de-orbiting and is the core technology for the energy reboost phase of MXER systems. In the context of rendezvous, distributed Lorentz forces can also act as a control actuator, providing a torque to correct in-plane positional errors or to damp in-plane and out-of-plane librations [26].

Changing the Moment of Inertia A powerful way to control a tether's rotational dynamics is by actively changing its mass distribution, and therefore its moment of inertia. This is the principle behind the three configurations investigated in this work.

- **Tether Reeling** is the most fundamental mechanical actuation method. By reeling the tether in or out from the control station or the tip, the tether's length is changed, directly affecting the system's moment of inertia and rotational speed. This technique is used to "pump" the tether's dynamics to the correct phase (i.e. orientation) and amplitude (i.e. length) for rendezvous and to subsequently damp oscillations after capture. While highly effective for controlling in-plane motion, reeling alone is less effective for managing out-of-plane librations [22].
- A **Climbing Actuator Mass**, or crawler, is an auxiliary mass that moves along the tether's length between the endpoints [35]. This motion alters the system's mass distribution without changing the tether length, providing another mechanism to control the moment of inertia and rotational dynamics. Studies have shown that a crawler can pride prolonged proximity with the payload in the case of librating tethers [36].
- A **Reeling Actuator Mass** combines the concepts of the climber and tip reeling. This is an intermediate mass that contains its own tether segment and reeling mechanism. This allows for independent control over different segments of the tether, offering more sophisticated manipulation of the system's overall mass distribution, length, and moment of inertia.

Other Actuation Methods To supplement these primary methods, other actuators have been considered. Thrusters, located on the main satellite or the tether tip, can be used to augment control, particularly for out-of-plane manoeuvres where other methods are less effective. While propellant-based thrusters can potentially be effective at controlling out-of-plane motions, they come at the expense of propellant use. This is of course counter to the main goal of the tether system, which is to provide a near propellantless space transport alternative. Besides thrusters, actively moving the tether's attachment point at the tether tip or the control station can act as a wave-absorbing controller, suppressing unstable transverse vibrations that may be excited by other forces. The same effect could potentially be achieved with the intermediate climbing and reeling actuator masses previously mentioned.

3

Reinforcement Learning Literature Review

3.1. Introduction to Reinforcement Learning

Definition of Important Terms
The literature on reinforcement learning contains numerous jargon terms. For definitions of all the jargon terms used in the following sections, please refer to Table 6.

Machine learning encompasses a variety of approaches to enable computers to learn from data and improve their performance on specific tasks. These approaches can be broadly divided into three main categories: supervised, unsupervised, and reinforcement learning [37].

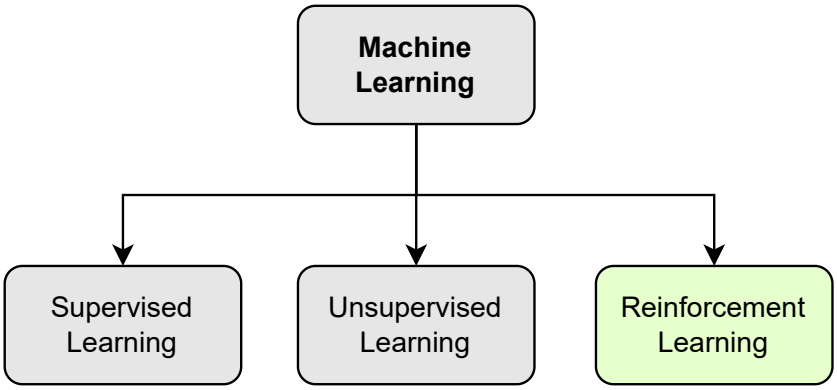


Figure 3.1: The three broad categories of machine learning

Supervised learning focuses on learning a mapping from inputs to outputs given a labelled dataset, essentially learning by example. [37]. Unsupervised learning, in contrast, aims to find hidden structures and patterns in unlabelled data, such as through clustering or dimensionality reduction. [37]. Reinforcement Learning (RL) occupies a unique position within this landscape, as it focuses on learning optimal behaviours through trial and error by interacting with an environment [37]. Where supervised and unsupervised

learning typically operates on supplied datasets, RL mostly generates its own learning data through interaction with an environment, and receiving feedback on the quality of those interactions through rewards. This paradigm is particularly suited for tasks where outcomes depend on a series of actions rather than isolated predictions as in the cases of supervised and unsupervised learning.

3.1.1. Core Concepts of Reinforcement Learning

At its core, RL can be divided into two entities, the agent and the environment, as well as three information streams or signals, namely observations, actions and rewards [38]. The general relationships between the agent, the environment, observations, actions and rewards are depicted in Figure 3.2.

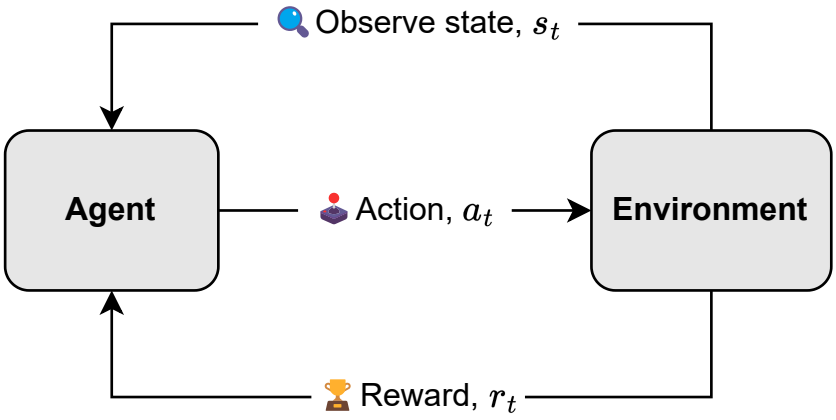


Figure 3.2: Schematic representation of reinforcement learning, where the agent observes state(s) from the environment, acts upon those observations and receives a reward for a proposed action at each time step t.

The first entity, the agent, acts as the central decision-maker in the RL model, which learns to make optimal or near-optimal decisions to achieve one or more specific goals through its trail-and-error interaction with the environment. The second entity, namely the environment, makes up the world that the agent interacts with, and thus encompasses everything outside of the agent [39]. It can be either a physical, real-world space, or a virtual space such as a simulated dynamical model for a MXER tether system.

The agent's interaction with the environment typically takes place over a sequence of discrete time steps, with one cycle as depicted in Figure 3.2 taking place per time step. This cycle starts with the agent observing the current state of the environment, to build a representation of the current environment situation or configuration. While full environment states are available to the agent in some cases, it is often the case that the environment state is only partially observed [38]. Based on this observation, the agent takes an action according to its policy, which is essentially an internal strategy for selecting action(s) for a given a state. The environment, in turn, transitions to a new state and provides the agent with a scalar reward, which serves as feedback on the desirability of the action taken and the new state [37]. As might be expected, actions which promote transitions to more desirable states are reinforced with increasing

rewards and transitions to less desirable states lead to reduced or even negative rewards. The agent's ultimate goal is to learn an optimal policy that maximises the expected cumulative reward over time by learning from the consequences of its actions [37].

This ability of RL agents to learn from the causal relationship between actions and observations is crucial for successful sequential decision-making, since the agent's current action influences not only immediate rewards but may also impact future states and rewards. This aspect further distinguishes RL from other machine learning paradigms and makes it suitable for complex tasks like playing games [40], robotics [41], and autonomous control systems [42], which are all areas where a degree of foresight may be valuable.

3.2. Defining State, Action, and Reward Spaces

In reinforcement learning, defining the state, action, and reward spaces is crucial for formulating the problem and designing an effective learning agent. These three components dictate how the agent perceives the environment, what actions it can take, and what feedback it receives. For instance, in autonomous vehicle control, the observation might include the vehicle's current speed and distance to nearby obstacles, the action could be a steering angle or throttle command, and the reward might reflect how well the vehicle stays in its lane while avoiding collisions. The state, action and reward spaces are further discussed in subsequent sections.

3.2.1. State Representations

As previously alluded to, the **state** represents the current situation of the environment and and and observation of the state provides the agent with the information it needs to make decisions. States are sometimes confused with **observations**, although these can differ. While a state forms a complete description of the environment at a given time, an observation is a description of the state that the agent can perceive, often limited by available sensor measurements. As such, an observation can be thought of as a subset of the state [43]. In the context of a MXER tether, the environment state may include a full dynamical description of the tether and payload as well as detailed information about perturbing effects, and surrounding bodies. An example observation on the other hand, may only include important tether information like rotational rate, maximum tension, as well as the position and velocity of key points on the tether such as the tether tip, control station and actuators.

Feature Engineering

Though it might initially seem beneficial to specify observations that include as much of the environment state as possible for informed decision-making, it is good practice to carefully select or engineer observation features that are most relevant to the task. This process is often called **feature engineering**. Such features can potentially be derived or engineered from the raw observations to create or extract more informative representations of the environment state that can help the agent make better decisions [44].

Effective feature selection in reinforcement learning has two noteworthy benefits, namely

a reduction in dimensionality and an improvement in prediction accuracy [45]. The benefit of a reduction in dimensionality becomes clear when RL models with underlying artificial neural networks (ANN) are considered. If a large observation is passed as input to an ANN, the network requires a sufficient amount of neurons to capture and process the information contained therein, ultimately requiring larger networks. In this context, neurons are the basic processing units in an artificial neural network that transform input data through weighted connections; more neurons (and thus larger networks) enable richer representations but increase computational cost. As such smaller networks are preferred where possible. The second benefit of effective feature selection is its potential to improve prediction accuracy, through identifying and retaining relevant features while eliminating redundant and irrelevant ones [46]. It is again helpful to think of this selection of key features in the context of ANNs, where the underlying structure of relationships between input parameters and their effects can be learned due to the ability of ANNs to act as universal function approximators [47]. By focusing on relevant features as inputs into ANNs, these networks can learn more accurate and reduced-noise approximations of the underlying nature of these input-output relationships [48].

Though the process of feature engineering can have significant benefits for the computational requirements and performance accuracy of an RL model, the selection or engineering of such features is not always trivial, and often requires an understanding of the underlying environment dynamics and control problem characteristics. In the case of the MXER tether system a simple feature engineering example may be to use the tether tip position and velocity relative to that the payload as an input to the RL agent, rather than the absolute positions and velocities of both the tether tip and the payload. Such a change reduces the dimension of the input into the RL agent while potentially simplifying the internal agent encoding of the distance and velocity difference between these two points. Another example may be to augment the observed state with the control input from the previous time step to improve the agent's "understanding" of the rate of actuator use and its practical limits.

3.2.2. Action Spaces

The **action space** of an RL agent defines the set of possible actions that the agent can take at each time step. Actions can be **discrete**, such as choosing from a finite set of options (e.g. left or right), or **continuous**, such as applying a force or torque within a certain range. In the case of MXER tether systems, the actions might involve adjusting the tether's length or applying forces to an actuator mass [8].

Action limits and smoothness

When designing the action space, it is important to consider actuator-specific limitations, such as actuator saturation or the maximum rate of change of control inputs. For example, in order to protect actuator hardware it might be necessary to limit the actions to a certain range. In some applications enforcing reasonably smooth and continuous actions are crucial to either protect actuator hardware or to adhere to actuation actions that are physically possible. This is particularly relevant for RL agents that are trained using virtual or simulated environments, and are intended to be deployed on real-world hardware. The consequences of rapid actuator use and unrealistic actuation rates might

not have detrimental effects in the simulated environments, but may be catastrophic if not prevented in real-world environments.

A variety of methods exist for enforcing or encouraging smooth actuator behaviour, with the simplest method requiring only modification of the reward function to reward actions that change minimally over time and penalise rapid changes. A slightly more advanced method is to change the RL agent's action output from a direct control action a_t, to an incremental action change Δa_t with limits on the allowable increment values. This approach effectively limits the rate of change of the action to the maximum allowable action increment divided by the discrete time step of the environment evolution. While the RL-environment-interaction often takes place over a fixed time step, this incremental action approach might be unsuitable for environments with dynamics that evolve over variable time steps, as the effective allowable action rate can then vary based on the time step size if not appropriately handled. Another potential drawback of this method is the need to augment the observed state passed to the RL agent with the previous control action value a_{t-1}, such that it can be used and learned as a basis for applying the next control action $a_t = a_{t-1} + \Delta a_t$. More advanced methods of obtaining smoother actions also exist. An example of such a method is Conditioning for Action Policy Smoothness (CAPS) [49], which modifies the loss function for the learned policy of an RL agent by introducing two new loss terms, one for spatial and one for temporal action smoothness.

3.2.3. Reward Function Design

The **reward function** provides feedback to the agent about the desirability of its actions. Designing an effective reward function is crucial for the success of RL, as it guides the agent towards the desired behaviour [37]. In many applications, the agent receives a final or **terminal reward** upon completing the task or when an episode ends. However, relying solely on such final rewards can make the learning process slow and inefficient, especially in tasks with sparse or delayed rewards [43]. To address this issue, **reward shaping** can be employed, where intermediate or frequent rewards are provided to guide the agent towards the goal [50][51]. For example, in the context of tether control, while the ultimate goal is to have the tether tip rendezvous with the payload by matching the positions and velocities of both, other aspects like tether tension, rotational rate and actuation rate may also be of importance. In such cases shaping the reward to encourage actions that reduce tether tension, limit rotational rate variation and discourage excessive actuator use can be beneficial.

Although reward shaping is a reasonably accessible way to introduce soft constraints into the RL agent, it is important to keep the reward function as abstract as possible to avoid artificially limiting the exploration of the agent. Overly strict reward functions may discourage the agent from exploring potentially beneficial states or actions, leading to suboptimal policies [52].

3.3. Model-Free vs. Model-Based Reinforcement Learning

Reinforcement learning methods can be broadly categorised into two main types; either model-free or model-based, depending on whether they make use of an internal model of the environment during their learning and decision making.

3.3.1. Definitions

Model-free approach

Model-free methods, learn a policy directly from interactions with the environment, without explicitly modelling the environment's dynamics within the agent. These methods do not require any prior knowledge of the system dynamics and purely rely on trial-and-error experience with the environment [37] as depicted in Figure 3.3.

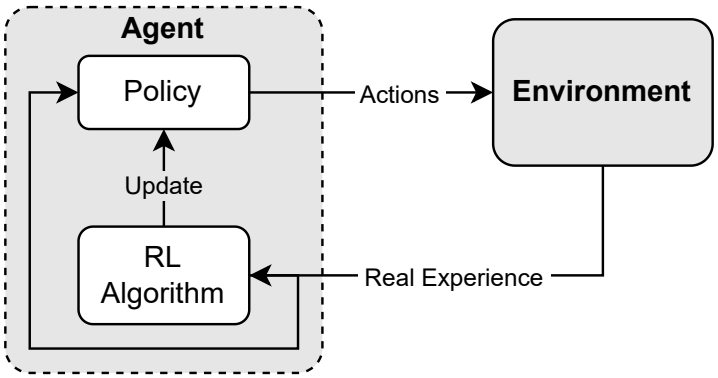


Figure 3.3: Model-free reinforcement learning with learning experience coming only from environment interactions.

Model-based approach

Model-based methods leverage a model of the environment within the agent to plan and make decisions [38]. Such an approach is depicted in Figure 3.4. This internal environment model can be either:

- **Explicit existing dynamic models**: These are pre-existing mathematical models that describe the system's dynamics, such as those based on physical equations or system identification techniques.
- **Learned internally using neural networks**: These models are learned from data collected through interactions with the environment. Neural networks are commonly used to approximate the dynamics due to their ability to represent complex, non-linear functions [53].

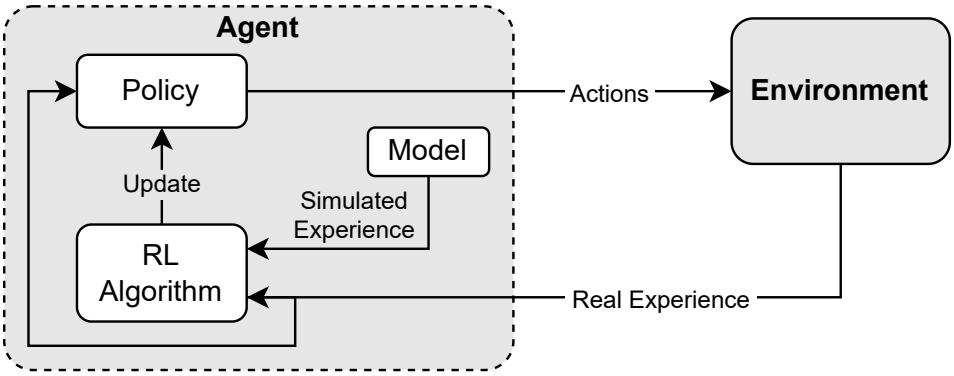


Figure 3.4: Model-based reinforcement learning with learning experience coming from both real and simulated environment interactions.

3.3.2. Comparative Discussion

Model-free and model-based RL methods each have their own strengths and weaknesses. Model-free methods are generally easier to implement and can be applied to a wider range of problems, as they do not require a model of the environment. However, they often suffer from high sample complexity, meaning that they require a large amount of data generated from environment interactions to learn an effective policy [37]. Model-based methods, on the other hand, can be more sample-efficient, as they can use the model to plan and simulate the consequences of actions without actually interacting with the environment [54]. These simulated experiences can potentially run faster than the real-time speed of the actual environment, allowing the agent to effectively plan its real actions ahead of time based on their simulated outcomes. However, the practical usefulness of these simulated experiences rely on the accuracy of the learned or provided environment model, which means that any errors or biases in the model can lead to suboptimal or even unsafe policies [55]. Model-based methods also tend to be more computationally expensive than model-free methods, as they involve planning and optimisation over the model.

Table 3.1: A comparative overview of model-free and model-based reinforcement learning.

Feature	Model-Free RL	Model-Based RL
Sample Efficiency	Generally lower	Potentially higher
Computational Cost at Runtime	Lower	Higher
Ease of Implementation	Easier	More complex
Reliance on Model Accuracy	Not applicable	High
Control Accuracy	Can be high with sufficient data	Depends on model accuracy

3.3.3. Suitability for Tether Dynamics

Reinforcement learning environments can generally take the form of a real-world, hardware-based environment, such as autonomous driving or robotics, or virtual environments, such as video games or high-fidelity simulations of real-world processes.

For the case where an environment is itself a simulation model, the model-free approach is attractive for its reduced computational cost. Simulated environments can typically be executed faster than their real-time, hardware-based counterparts, and are not subject to physical degradation such as actuator wear or system breakdown. These advantages diminish the importance of sample efficiency, making the model-free approach even more suitable. In the case of space tether systems, and MXER tethers in particular which remain in the conceptual and prototyping stages, research is inherently constrained to simulation-based studies. These simulations often rely on simplifying assumptions that sacrifice fidelity for speed, further reducing the cost of agent-environment interactions. Consequently, sample efficiency is not a critical concern. Furthermore, widely adopted RL libraries such as `stable-baselines3` facilitate the implementation of model-free algorithms with minimal overhead. Additionally, since the application of reinforcement learning to MXER space tether control remains largely unexplored, starting with model-free methods allows for the assessment of RL's viability in this context without committing excessive resources to more intricate model-based alternatives that may ultimately prove unsuitable. Taken together, these factors strongly support the use of model-free reinforcement learning over model-based alternatives in the context of this work.

3.4. Common Reinforcement Learning Agent Architectures

Besides their reliance (or lack thereof) on an internal environment model, reinforcement learning algorithms can be categorised based on their underlying architecture. These algorithms often fall into on of three main categories including value-based, policy-based, and actor-critic architectures. The main idea(s), benefits and developments for each category is briefly presented in following sections. For more in-depth information, the reader is referred to [37].

3.4.1. Value-Based Architectures

Core Principles and Mathematical Relationships

Value-based architectures focus on learning a state-value function, $V(s)$, or an action-value function, $Q(s, a)$, which estimates the expected cumulative reward for each state or state-action pair, respectively. The policy is then derived implicitly from the value function, typically by selecting actions that maximise the output of one of these value functions [37].

Benefits, Drawbacks, and Suitability

Value-based approaches lend themselves well to problems with discrete state and action spaces, where a finite number of states or state-action pairs can be tested for value-function maximisation. However, without modification these approaches are less suited

to continuous action domains where finding the action that leads to a maximised value function becomes intractable as the possible number of actions approach infinity. These methods perform well in tasks with terminal rewards and low-dimensional state spaces but often require function approximation such as Deep Q-Networks (DQN) in high-dimensional problems [56].

Historical Perspective and Recent Developments

Classical methods such as SARSA and Q-learning emerged during the 1980s [57] and 1990s [58]. The integration of deep neural networks led to Deep Q-Networks (DQN) [56], which marked a breakthrough in handling high-dimensional inputs. Variants like Double DQN [59] and Dueling DQN [60] have since addressed issues such as overestimation bias and remain state of the art for many discrete control tasks.

3.4.2. Policy-Based Architectures

Core Principles and Mathematical Relationships

Policy-based methods directly optimise a policy $\pi_\theta(a|s)$ parametrised by θ without explicitly learning a value function. The goal is to maximise the expected return J, which serves as a measure of the policy's performance [61].

$$J(\theta) = \mathbb{E}_{\pi_\theta}[G_t] = \mathbb{E}_{\pi_\theta}\left[\sum_{k=0}^{\infty} \gamma^k r_k\right].$$

where γ is the discount factor, and r is the reward at time step k. Maximising the expected return is typically formulated as an optimisation problem over the parameters θ for which policy gradient methods can be used to approximate the gradient for steepest ascent optimisation approaches [38]. Applying the parameters that lead to a maximal J to the parametrised policy, yields the desired (near) optimal policy $\pi^*(a|s)$.

Benefits, Drawbacks, and Suitability

Policy-based approaches naturally handle continuous action spaces and can learn stochastic policies [62], making them ideal for environments with uncertainty [63]. They avoid the value-function maximisation step required by value-based methods but suffer from high variance in gradient estimates and often require many samples[64][65]. Variance reduction techniques—such as the use of baselines are common [66], though they blur the line with actor-critic methods.

Historical Perspective and Recent Developments

Early work on policy gradients such as REINFORCE appeared in the 1990s [67]. Despite challenges with high variance, subsequent developments like the use of baselines, natural policy gradients [68], and trust region methods [69] have improved stability. Modern algorithms like Proximal Policy Optimisation (PPO) incorporate surrogate objectives and clipping mechanisms to limit policy updates, thereby improving robustness in continuous control tasks [70].

3.4.3. Actor-Critic Architectures

Core Principles and Mathematical Relationships

Actor-critic methods combine the strengths of both value-based and policy-based approaches by maintaining two networks: an actor, which represents the policy $\pi_\theta(a|s)$, and a critic, which estimates the state-value function $V(s)$ or action-value function $Q(s,a)$. The critic, as the name implies, critiques the actions of the actor, thereby providing it useful feedback for improvement. As an example [71], the critic's estimate of the state value function $V(s)$ can be used to compute an advantage function $A(s,a)$ which serves as a measure of the benefit of a particular action over the average value of a given state.

$$A(s,a) = G_t - V(s),$$

This advantage is then used in the policy gradient update of the actor [72]

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \ln \pi_\theta(a|s)\, A(s,a)\right].$$

The use of an actor-critic architecture with an advantage function can help to significantly reduce the variance of the policy gradient update [72] which helps to improve model stability and convergence.

Benefits, Drawbacks, and Suitability

Actor-critic methods are highly versatile, operating efficiently in both discrete and continuous domains [73][74]. The critic's value estimates lower the variance of the gradient estimates and accelerate convergence [71]. However, the need to simultaneously learn two functions adds complexity, and instability can occur if the critic's estimates are inaccurate [75]. They are well suited for tasks where managing long-term returns and action variability is critical, such as robotics and autonomous control [76].

Historical Perspective and Recent Developments

Originating from ideas in the 1980s and formalised in the late 1990s, actor-critic methods have evolved from linear approximators to deep architectures. Modern variants, such as Asynchronous Advantage Actor-Critic (A3C) [77], Soft Actor-Critic (SAC) [78], and Distributional Soft Actor-Critic [79], incorporate techniques like entropy regularisation and distributional value estimation to improve exploration and stability in high-dimensional environments.

3.4.4. Additional considerations

Besides the overall value-based, policy-based or actor-critic architectures that algorithms can fall under, there are numerous smaller underlying architecture categories and modifications that can change how an algorithm works. Some of these can be added or removed from algorithms like Lego bricks, while others are more stringent, requiring significant changes to the underlying algorithm. Two such additional architecture considerations that has significantly improved algorithms are on- and off-policy learning, and hindsight experience replay.

On- vs off-policy learning

As yet another categorisation, RL algorithms can fall into on-policy or off-policy categories. On-policy algorithms, such as Proximal Policy Optimisation (PPO) [70], learn strictly from data collected using the current policy. In contrast, off-policy algorithms, such as Soft Actor-Critic (SAC) [78], can learn from data collected using a different policy, which allows for greater sample efficiency by reusing past experiences. Naturally, such data from other policies requires storage for later reuse, which may lead off-policy methods to greater memory and thus hardware requirements [80].

Hindsight Experience Replay

Hindsight experience replay (HER) is a technique that can improve the sample efficiency of off-policy RL algorithms [81]. In HER, the agent not only learns from its actual experiences but also from "hindsight" experiences, where it imagines that it had a different goal than the one it actually pursued. This allows the agent to learn from sparse or delayed rewards more effectively. HER is particularly useful for environments with sparse rewards. This makes it attractive as a possible method for relaxing the need for complicated or restrictive reward shaping [81].

3.4.5. Selected algorithm - Soft Actor-Critic (SAC)

Architecture type

Off-policy, actor-critic architecture

Key features

SAC is an off-policy actor-critic algorithm that incorporates the concept of maximum entropy reinforcement learning [78]. This means that the agent not only tries to maximise the expected return but also the entropy of its policy by acting as randomly as possible [82]. This encourages exploration and makes the learning process more robust to noise and other disturbances. (Packer et al., 2018). To mitigate overestimation bias from the critic, SAC incorporates two action-value functions (two critics) and uses the minimum prediction of the two for each policy update. For the first version of SAC, the exploration-exploitation trade-off was controlled by a hyperparameter [82], but a subsequent version learns a separate temperature parameter to automatically balance this trade-off [78].

Success in tasks with high-dimensional state-action spaces

SAC has been shown to be effective in tasks with high-dimensional state and action spaces, such as robotic claw manipulation [78] and robotic locomotion [83].

3.5. Reinforcement Learning in Control Systems

Reinforcement learning has emerged as a powerful tool in control systems, offering solutions to complex, non-linear, and high-dimensional problems that traditional control methods may struggle to address [84]. This section explores the relationship between RL and conventional control strategies, highlighting the advantages of RL in various applications.

3.5.1. Conventional Optimal Control vs Reinforcement Learning

Traditional optimal control methods, such as Model Predictive Control (MPC) and dynamic programming, typically rely on explicit models of the system dynamics and solve optimisation problems to find the optimal control policy [85]. These methods can provide strong theoretical guarantees on performance and stability, but they often require accurate models and can be computationally expensive, especially for high-dimensional systems.

Reinforcement learning, on the other hand, offers a more data-driven approach to optimal control. Since RL algorithms learn from data generated through interactions with the environment, the need for explicit models or pre-defined control laws is eliminated [37]. This makes them particularly well-suited for complex, non-linear systems where accurate models are difficult to obtain [84].

RL algorithms have a few additional advantages over traditional optimal control methods, including being able to adaptability and scalability to control high-dimensional systems. Through continual training, RL methods can adapt to changes in the environment or system dynamics by continuously learning from new experiences [86]. RL methods have also been proven to excel with high-dimensional sensory inputs [56] while conventional control methods can fall prey to the curse of dimensionality of high-dimensional systems. A noteworthy drawback though, is that while some RL implementations can offer stability guarantees [87], such guarantees are not always a focus of RL methods, and should be explicitly included in the method design where required for control tasks [84].

3.5.2. Applications in similar domains

Considering the benefits of RL over traditional control, it comes with little surprise that RL methods have been successfully applied to a variety of non-linear control problems as well as space-based applications. Some examples include:

- **Non-linear dynamic systems**: RL has been used to control various non-linear dynamic systems, including robotic manipulators [88], autonomous vehicles [89], and process control systems [90].
- **Space applications**: RL has found applications in several space-related control problems, including spacecraft rendezvous and docking [91], satellite attitude control [92], and trajectory design [93].

These examples demonstrate the potential of RL for solving complex control problems in domains with similar characteristics to tether control, such as high dimensionality, non-linearity, and the domain of space and orbital dynamics.

3.6. Challenges and Limitations of RL in Control Systems

While RL methods can and have been used successfully for the control of dynamical systems, there are a few important challenges and limitations of RL to take into consideration before applying these methods to a control problem.

3.6.1. Sample Efficiency

Reinforcement learning algorithms, particularly model-free methods, are often data-hungry and require a large number of interactions with the environment to learn an effective policy [37]. This can be a significant limitation in real-world applications or computationally expensive environments, where data collection is time-consuming or costly. For example, in high-fidelity tether simulations, each simulation run may take a significant amount of time, limiting the number of interactions that can be performed within a reasonable time-frame.

3.6.2. Transferability

Transferring learned policies from simulation to real-world applications or from simplified models to more detailed simulators can be challenging. [94]. The discrepancies between the training environment and the target environment can lead to a significant drop in performance. This is often referred to as the "reality gap" or "sim2real gap", with a number of techniques having been proposed to address this issue [95].

3.6.3. Computational costs

Training deep RL agents can be computationally expensive, especially for complex tasks and environments. Large-scale RL applications often require significant computational resources, such as powerful CPUs, GPUs, and large amounts of memory. (OpenAI et al., 2018). This can limit the applicability of RL to problems where such resources are available.

Some approaches to reduce the computational costs and speed up the training of RL include using more efficient network architectures [96] and making use of distributed training [97]. After training, model compression techniques can make learned policies more suitable for deployment on resource constrained devices [98].

3.6.4. Hyperparameter Tuning

Similar to gain tuning for transitional control methods, RL algorithms and their underlying neural networks may require tuning of hyperparameters for optimal performance. The number of hyperparameters for different RL methods may become uncomfortably large, which can make finding suited hyperparameter combinations a non-trivial and time-consuming task. Several techniques are available for hyperparameter optimisation, with common options including grid search, random search [99] and Bayesian optimisation [100]. Frameworks like Optuna [101] offer a way of automating the hyperparameter tuning process.

3.6.5. Reward-goal Mismatch

As mentioned in Section 3.2.3, sparse rewards may lead to slow, inefficient learning while overly strict rewards can artificially restrict an agent's exploration. Another reward-related phenomena to be aware of is so called "reward hacking", where the agent finds a way to exploit the reward function without actually achieving the intended goal [102].

Reward hacking behaviour is often a symptom of a mismatch between the intended goal, and the (mis)specified reward function's conveyance of that goal to the RL agent [103]. Careful crafting of the reward function may remove or reduce this behaviour, but may also lead to overly strict rewards. Reward function design may thus have to go through several iterations before a reward function that sufficiently captures the true goal of the agent is found.

3.6.6. Explainability and Interpretability

Deep RL policies, especially those represented by large neural networks, can be difficult to interpret and understand. This "black box" nature of deep RL can be a serious limitation in safety-critical applications, where it is important to be able to verify and validate the behaviour of the agent [104]. While methods that provide some insights into the behaviour of RL agents exist [105], developing more interpretable and explainable RL algorithms remains an active area of research.

4

Research Gap and Opportunity

4.1. Research Opportunity

The literature review in the preceding chapters establishes that while Momentum Exchange with Electrodynamic Reboost (MXER) tethers present a transformative technology for space transportation, their practical implementation is hindered by significant control challenges. The primary obstacle is the extremely short rendezvous window for payload capture, which is a direct consequence of the system's non-linear, and high-speed rotational dynamics. Although the literature details various control methods and actuation configurations, a critical analysis reveals two principal gaps where further research is needed.

First, while several mechanical actuation methods for controlling a tether's moment of inertia have been proposed, namely tip reeling, a climbing actuator mass, and a reeling actuator mass, the comprehensive analysis of the latter two's control authority has largely been confined to librating (slowly oscillating) tethers. The performance of the more complex climber and reeler configurations, when applied to the distinct, rotational dynamics of a spinning MXER tether, remains largely unquantified. A direct, comparative study is required to determine which of these configurations offers the greatest potential for extending the rendezvous window in a MXER operational scenario.

Second, existing control strategies face a fundamental trade-off. Conventional optimal control methods, such as iLQR, can provide robust and stable control but often rely on simplified dynamical models, like rigid-body formulations, to remain computationally tractable. Their performance and computational cost can degrade significantly when applied to higher-fidelity models that include the elastic and non-linear effects inherent in long tethers. Conversely, the literature on Reinforcement Learning (RL) presents a compelling, data-driven alternative. Modern model-free RL algorithms, particularly actor-critic architectures like Soft Actor-Critic (SAC), are inherently suited to solving complex, non-linear control problems without requiring an explicit, and potentially simplified, system model. By learning directly from interaction with a simulated environment, RL offers the potential to discover sophisticated control policies that are difficult to derive using conventional techniques.

Therefore, a clear research opportunity emerges from the intersection of these two gaps: to perform a systematic investigation that, firstly, evaluates the comparative effectiveness of different actuator configurations on a spinning MXER system and, secondly, provides a direct comparison between a well-established conventional (near) optimal controller and a modern DRL agent for this challenging control problem. Such a study would not only address a specific gap in the tether control literature but also provide valuable insights into the practical proof-of-concept application of modern machine learning techniques to tether systems. This opportunity directly motivates the research questions and goals outlined in the subsequent sections.

4.2. Configuration Definitions

This section provides a brief descriptive definition of the three configurations considered in this work. The three configurations are depicted diagrammatically in Figure 4.1, with annotations to identify their main components and the component motion relative to the tether.

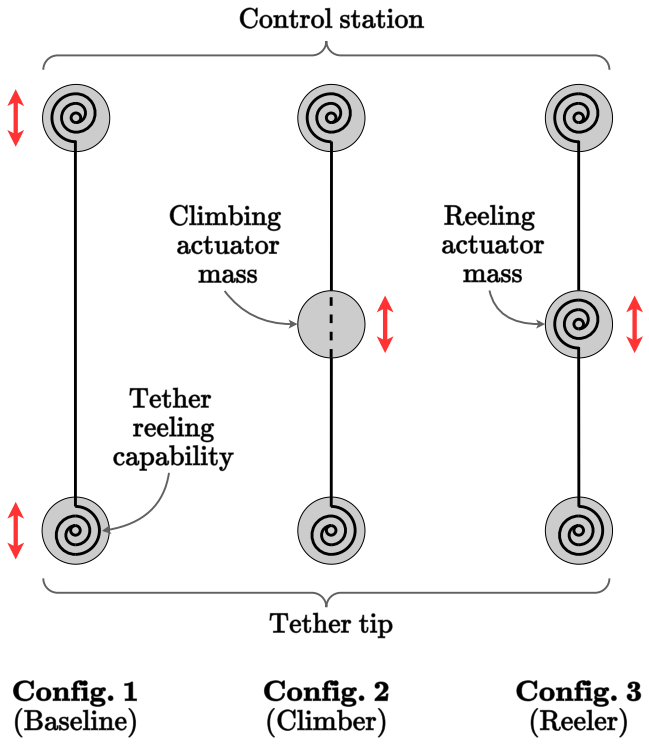


Figure 4.1: Three main actuator configurations in literature. From left to right, tether tip reeling (the typical MXER tether design), climbing actuator mass, and reeling actuator mass. Red arrows indicate motion of masses relative to tether. All configurations support tether reeling at both the control station and the tip, top and bottom red arrows are only omitted from config. 2 and 3 to emphasise intermediate mass motion.

The first actuation configuration, depicted on the left in Figure 4.1, serves as the *baseline* design. It reflects the conventional approach proposed for large-scale MXER tether

systems [6], [28]. In this setup, the effective tether length is controlled by adjusting the amount of tether unspooled from the end masses [106]. This relative motion is illustrated by the red arrows in Figure 4.1. While traditional designs focus on reeling at the tether tip, this work also incorporates reeling capability at the control station. By coordinating reeling actions at both ends, the system can modulate the overall tether length and shift the tether system's centre of mass (COM). This enables partial, yet coupled, control over two key aspects:

- **Rotational arm of the tether tip**: The distance from the COM affects the tip's position relative to the payload during capture.
- **Rotational moment of inertia**: Changes in mass distribution (i.e. moment of inertia) influence angular velocity via conservation of angular momentum, which in turn affects the tether tip's tangential velocity through the relation $V_T = r\omega$.

Thus, the baseline configuration offers control over both the position and velocity of the tether tip, though not independently.

This dual-end reeling capability is retained in the two enhanced configurations shown in Figure 4.1, where intermediate actuator masses are added to augment rather than replace the system's degrees of freedom.

The second configuration of Figure 4.1, termed the *climber*, builds upon the baseline by introducing a climbing actuator mass between the tether tip and control station [35]. This actuator behaves like an elevator, a constant mass moving along the tether without altering its length. The climber's motion shifts the system's COM, thereby influencing the tether tip's position and velocity through a lengthening or shortening of the rotational arm. Additionally, the climber alters the mass distribution, affecting the moment of inertia and angular velocity. A key advantage of this setup is its ability to counterbalance COM shifts caused by tether reeling, or to accelerate changes in moment of inertia more effectively than the baseline. These capabilities offer greater flexibility in achieving desired control states.

The third configuration, known as the *reeler*, also features an intermediate actuator mass, but divides the tether into two segments: one between the tip and actuator, and another between the actuator and control station (configuration 3 in Figure 4.1). Unlike the climber, the reeler mass can reel tether in and out. Reeling the first segment shortens the distance to the tether tip; reeling the second segment brings the actuator (and tip) closer to the control station. Simultaneous reeling of both segments allows dynamic positioning of the actuator relative to both ends [20]. In this study, the actuator is configured to affect only the segment connected to the tether tip. The reeler configuration retains the advantages of the climber, with the added benefit of enhanced control over tether length and its rate of change thanks to the third reeling mass. This results in a broader range of achievable tether tip states during controlled payload capture.

4.3. Reference Tether System

Around the turn of the millennium, work from Tethers Unlimited, Inc. proposed the Cislunar Tether Transport System, which would launch 1000kg payloads from low earth

orbit (LEO) to a Lunar Transfer Orbit (LTO) once every 30 days [7]. Although other, larger proposed MXER tethers also exist, like those intended for frequent Earth-Mars travel, the Cislunar Tether Transport system will be used as a baseline tether design due its increased short-term feasibility over larger systems. By building on this design the three actuator configurations in Figure 4.1 can be tested to gauge their effect on a realistic MXER tether system. An example of what such a system may look like is shown in Figure 1.2. System masses, characteristics and orbital information are listed in Table 4.1, 4.2, and 4.3.

Table 4.1: Mass breakdown of the Cislunar Tether Transport system components [7].

System component	Mass	
Payload	1 000	kg
Tether	8 274	kg
Control station	15 004	kg
Tether tip	650	kg
Total tether system (excl. payload)	**23 928**	kg

Table 4.2: General tether characteristics of the Cislunar Tether Transport system [7].

Tether characteristics	Value	Unit
Length	100	km
Tip velocity at catch	1 555	m/s
Tip velocity at toss	1 493	m/s
Rotational period	330	sec

Table 4.3: Parameters related to the payload and tether system orbits [7].

General (pre-catch) orbital information	Payload	Tether	Unit
Perigee altitude	300	382	km
Apogee altitude	300	11 935	km
Perigee velocity	7 726	9 281	m/s
Apogee velocity	7 726	3 426	m/s
Semi-major axis	6 678	12 537	km
Eccentricity	0	0.461	-
Inclination	0	0	deg
Orbital period	5 431	13 970	sec
Orbital period	90.5	232.8	min

4.4. Research Questions

This section presents two main research questions and their supporting sub-questions:

RQ.1 *How does the actuator configuration influence the controllability of the rendezvous window of a rotating momentum exchange tether?*

(a) What are the dominant dynamical factors affecting the tether tip motion and rendezvous window in a simplified MXER system model?

(b) What are the key performance indicators (KPIs) for evaluating the controllability of the rendezvous window for the considered actuator configurations?

RQ.2 *Can reinforcement Learning (RL) be used as an advanced control strategy to effectively extend the rendezvous window for payload capture in a rotating momentum exchange tether system?*

(a) What are the suitable state and action spaces for representing the MXER system and control inputs within a reinforcement learning framework?

(b) How do different reward function formulations impact the performance of the RL algorithms?

(c) How does the chosen SAC RL algorithm [78] compare against the ILQR controller in terms of rendezvous window extension and suitability to the tether control task?

4.5. Project Goals

The three main goals of the proposed research project are:

G.1 To derive and implement a 2D tether dynamics model which can be used and adapted to simulate and ultimately compare the dynamical behaviour of three tether systems with different actuator configurations.

G.2 To implement an iterative linear quadratic regulator (ILQR) controller to establish a baseline for the trajectory-tracking control of the 2D tether dynamics model.

G.3 To implement and test the model-free SAC RL algorithm [78]) as a control method for the rendezvous dynamics of tether-payload capture.

4.5.1. Project Subgoals

1. To derive and implement a 2D tether dynamics model.
   (a) To derive a generalised model suitable for all three considered tether configurations.
   (b) To identify suitable control input(s) to the dynamics model for effective tether tip control.
2. To implement an ILQR controller for trajectory tracking baseline performance establishment.

(a) To implement an ILQR controller capable of working with general, black-box dynamics functions

(b) To expand the ILQR controller to incorporate general (soft) constraints on the dynamics and controls.

3. To implement and test the model-free SAC RL algorithm for trajectory tracking control of the tether dynamics.

(a) To implement a RL-friendly training environment of the tether dynamics.

(b) To compare the performance of the SAC RL algorithm against the ILQR controller based on their suitability to the tether control task.

(c) To identify the most impactful hyperparameters for the best-ranked RL model and their relative effects on tether control performance.

(d) To improve the tether control performance of the RL algorithm through tuning of commonly identified impactful hyperparameters.

4.6. Method Overview

The methodology used to address the two main research questions posed in Section 4.4 was broadly divided into four sections: (1) creating dynamical tether models, (2) applying a conventional control algorithm to these models, (3) setting up the best performing model in an RL environment, and (4) creating, training, and evaluating RL agents. The process outlined in Figure 4.2 followed sections (1) and (2) from left to right. A short description of each of the main steps in Figure 4.2 is provided below:

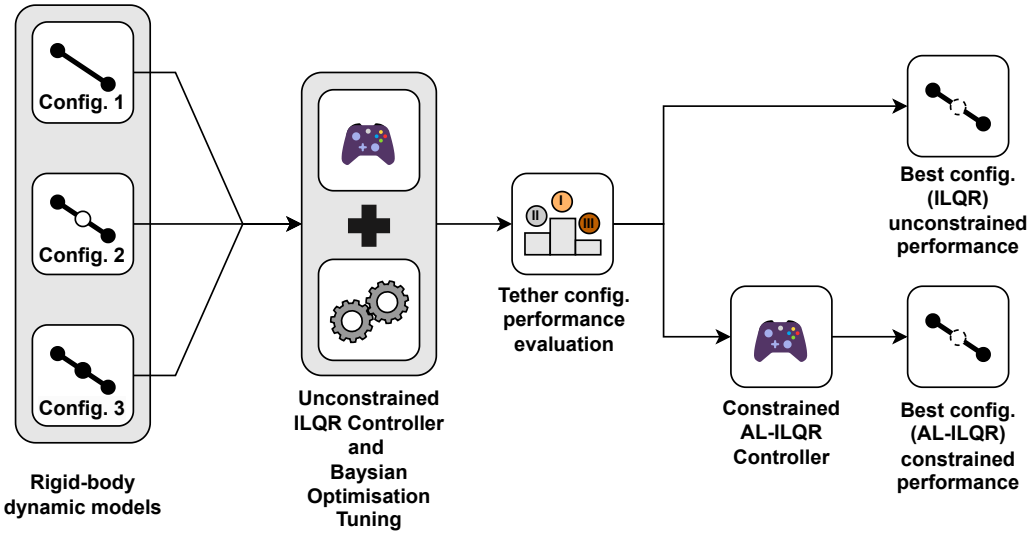


Figure 4.2: Diagrammatic depiction of the methodology used to determine the best performing tether configuration with ILQR as a conventional control method.

- **Rigid-body dynamic models:** A simplified, two-dimensional rigid-body tether model was derived to be used in conjunction with a control algorithm for controlling the tether tip motion. The model was derived in such a way to accommodate all three tether system configurations, with slight modifications where necessary.

- **Unconstrained controller and Bayesian optimisation tuning:** A conventional ILQR control algorithm was implemented to generate actuator commands such that the rigid-body tether models could track a reference trajectory over time. The control was subject to *unconstrained* tether dynamics to determine ideal tracking performance. In conjunction with this controller, three Bayesian optimisation studies (one for each configuration) were conducted to tune the controller parameters to maximise the duration the tether tip spent within a specified position and velocity tolerance relative to the payload. The controller tuning study was conducted using the Optuna framework [101].
- **Evaluate configuration performance:** The trajectory tracking performance of the three configurations under tuned control parameters was compared to the uncontrolled trajectory as well as to each other to determine the best performing configuration.
- **Isolate best unconstrained performance:** Based on the performance evaluation, the best performing tether configuration under unconstrained conditions was isolated for further study. Its unconstrained control performance was used as a baseline for comparison in the next section on unconstrained RL control.
- **Constrained controller:** An ILQR control algorithm was implemented within an augmented Lagrangian framework (AL-ILQR) to generate actuator commands such that the identified, best-performing tether configuration could track a reference trajectory over time. The control was subject to *constrained* tether dynamics to assess performance under more realistic operational conditions. The main constraint categories included tether tension limits, point mass g-load limits, and limits on actuator use.
- **Determine constrained performance:** The trajectory tracking performance of the best tether configuration under constrained control was determined. This constrained control performance was used as a baseline for comparison in the next section on constrained RL control.

Similarly, Figure 4.3 outlines the process for sections (3) and (4). A short description of each of the main steps in Figure 4.3 is provided below:

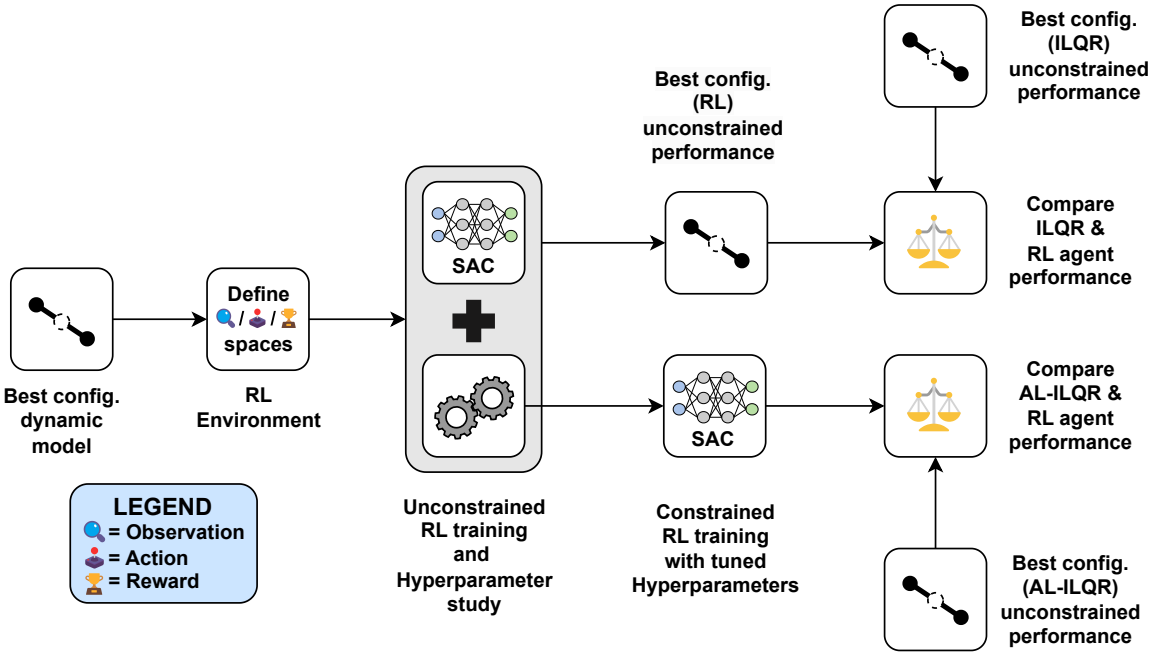


Figure 4.3: Diagrammatic depiction of the methodology used to determine the best performing RL agent, and its comparison to the conventional control baseline. Note that steps may have been revisited if iteration was deemed necessary during the project.

- **RL Environment:** The dynamical model of the best performing tether configuration from the final step in Figure 4.2 was integrated into a custom Farama Gymnasium[1] (previously OpenAI Gym) environment. The observation, action, and reward spaces necessary for RL agent-environment interaction were defined.
- **Unconstrained RL agent training and hyperparameter tuning:** The chosen RL agent using the SAC algorithm [78] available in the Stable-Baselines 3 library was trained on the unconstrained environment defined in the previous step. In conjunction with this training, a combined RL hyperparameter and reward parameter study was conducted to tune the RL controller's underlying algorithm and neural network hyperparameters, along with the reward parameters, to maximise the duration the tether tip spent within a specified position and velocity tolerance relative to the payload. The hyperparameter study was conducted using the Optuna framework [101].
- **Agent unconstrained performance evaluation:** After training the tuned RL agent on the unconstrained environment, its trajectory tracking performance was evaluated, focusing on the duration the tether tip spent within a specified position and velocity tolerance relative to the payload.
- **Compare unconstrained ILQR and RL control performance:** The controlled trajectories resulting from the unconstrained ILQR and RL controllers were compared to determine their relative trajectory tracking and rendezvous window extension performance.
- **Constrained RL training with tuned hyperparameters:** After hyperparam-

[1]Farama's Gymnasium is a standard and widely used interface for RL environments [107]

eter tuning for the unconstrained RL agent, its underlying neural network and reward parameters were used to train a new RL agent on the constrained environment.

- **Compare constrained ILQR and RL control performance:** The control performance of the tuned RL agent was evaluated against the conventionally controlled baseline under constrained tether dynamics to determine their relative trajectory tracking and rendezvous window extension performance.

4.7. Planning

4.7.1. Work Breakdown Structure

The project work breakdown structure, shown in Section B.1 is divided into four phases: developing dynamical models, controlling these models with conventional algorithm(s), setting up RL environments, and creating, training, and evaluating RL agents for the unconstrained and constrained cases. Under each phase various tasks and subtasks detail the work completed. Some sub-tasks were repeated for each tether configuration.

4.7.2. Rough Timeline

A rough timeline of the overall project is given in Figure B.2 of Section B.2, with more detailed timelines for each project phase in Figure B.3 to B.6. The durations for each phase are listed in Table 4.4. Each of the phase durations were divided into nominal and buffer times. The buffer times served as contingency plans in the event that one or more subtasks of a project phase required more time than anticipated, and ended up being necessary.

Table 4.4: Project phase approximate time allocations. Note that phases overlap in time.

Phase	Nominal Time [Weeks]	Buffer Time [Weeks]	Total Time [Weeks]
1. Dynamical models	6	3	9
2. Conventional Control	6	2	8
3. RL Environments	4	3	7
4. RL Agents	6	3	9

5

IAC 2025 Paper: Conventional and Reinforcement Learning Control of MXER Tether Dynamics for Extended Payload Rendezvous

Supporting material for the IAC paper is included in Appendix A.

76th International Astronautical Congress (IAC), Sydney, Australia, 29 September - 03 October 2025.

IAC–25–95733

Conventional and Reinforcement Learning Control of MXER Tether Dynamics for Extended Payload Rendezvous

Zander du Toit[a*], Marc Naeije[b]

[a] *Department of Aerospace Engineering, Delft University of Technology, The Netherland*
[b] *Department of Aerospace Engineering, Delft University of Technology, The Netherlands*
[*] *Corresponding author*

Abstract

Momentum Exchange with Electrodynamic Reboost (MXER) tethers transfer captured payloads to higher orbits using a long, rotating tether. This transfer occurs through a momentum exchange from the tether to the payload, after which the tether's orbital energy is restored via electrodynamic thrusting. MXER tethers offer a sustainable, reusable, and near-propellantless alternative to rockets for orbital and interplanetary transfer of payloads. However, the short rendezvous window for tether payload capture, typically lasting mere seconds, presents a significant challenge to the use of these tether systems. This research investigates the control of MXER tether dynamics, aiming to improve payload capture success by extending the rendezvous window. This work compares three actuator configurations (a baseline tip-reeling system, a climbing actuator mass, and a reeling actuator mass) previously studied for librating tethers, adapting them for a rotating MXER system based on the Cislunar Tether Transport System design. A 2D rigid-body model is used to simulate the system dynamics. Initially, a conventional iterative Linear Quadratic Regulator (iLQR) establishes a baseline for control performance. Subsequently, the model-free Soft Actor-Critic (SAC) Deep Reinforcement Learning (RL) algorithm is implemented and trained. Both control methods were tested with and without dynamic system constraints. The performance of each configuration is evaluated based on rendezvous window extension and constraint satisfaction. In the unconstrained case, the reeler configuration is shown to be the most effective, extending the rendezvous window to 1.8 seconds from the 0.6 seconds for the uncontrolled case. The SAC RL algorithm matches the performance of the tuned iLQR controller, but produces a less smooth control policy with sporadic actuator use. The constrained control proved more challenging, with neither the augmented-Lagrangian iLQR nor the SAC-based controller managing to extend the rendezvous window; the former was overly conservative, while the latter failed to satisfy operational constraints.

Nomenclature

Symbol	Meaning	Units
R_{CM}	COM orbital radius	m
R_p	Payload orbital radius	m
L	Tether length	m
L_i	Spooled tether length on mass i	m
α	Orientation angle	rad
ω	Rotation rate	rad/s
m_i	i-th point mass	kg
m_{tot}	Total mass	kg
d_i	Local distance of mass i	m
d_{CM}	COM offset	m
μ_E	Earth grav. parameter	km^3/s^2
I_{CM}	Moment of inertia	kgm^2
τ_g	Gravity-gradient torque	N m
T	Tether tension	N
W	Rendezvous window	s
γ	RL discount factor	–
$\Phi(t)$	Potential reward function	–

Acronyms/Abbreviations

Term	Acronym
Momentum eXchange Electrodynamic Reboost	MXER
Iterative Linear Quadratic Regulator	iLQR
Augmented-Lagrangian iLQR	AL-iLQR
Reinforcement Learning	RL
Soft Actor-Critic	SAC
Centre of Mass	COM
Degrees of Freedom	DOF

1. Introduction

Tethered space systems present a revolutionary approach to space exploration and transportation, offering significant benefits over traditional rocket-based propulsion. The core advantages of these systems stem from their capacity for momentum exchange and electrodynamic propulsion, enabling manoeuvres that are either propellantless or near-propellantless. These

capabilities have given rise to a wide array of proposed applications, from payload boosting and de-orbiting to satellite deployment and space debris removal.

A particularly promising application is the Momentum eXchange with Electrodynamic Reboost (MXER) tether system. An MXER system typically consists of a long, high-strength rotating tether with a counterweight at one end and a capture mechanism at the other. By rotating in an elliptical orbit, the system can capture a payload from a lower orbit and subsequently release it into a higher-energy trajectory. The energy transferred to the payload is drawn from the tether's orbital momentum, which is later restored using electrodynamic thrusting against the geomagnetic field, making the system reusable and sustainable. Despite these advantages, the operational viability of rotating MXER tethers faces a significant challenge: the rendezvous and capture of the payload. The rendezvous window at the tether's tip is extremely short, typically on the order of seconds, demanding highly accurate prediction and control of the tether's dynamics and precise guidance of the payload. Extending this brief window could drastically improve the feasibility and reduce the complexity of the payload capture effort.

This research aims to build upon these findings by investigating different actuator configurations and control methodologies to extend the rendezvous window. This paper addresses this by, firstly, adapting two actuator configurations previously studied for librating tethers, namely a climbing actuator mass and a reeling actuator mass, to a rotating MXER system to evaluate their control potential. Secondly, it moves beyond conventional control techniques by implementing a modern Deep Reinforcement Learning (RL) algorithm to manage the system's complex, non-linear dynamics. RL is particularly attractive for this task due to its ability to handle high-dimensional state spaces.

This paper is structured as follows: Section 2 provides a background on the control methods and the reference tether system. Section 3 details the 2D rigid-body dynamic model of the tether system. Section 4 outlines the implementation of the conventional and RL-based controllers. Finally, Section 5 presents and discusses the results of the comparative analysis, followed by conclusions in Section 6.

2. Background

2.1 Reference MXER tether

The analysis in this paper is grounded in a realistic baseline design to ensure the relevance of its findings. The chosen reference system is the Cislunar Tether Transport System, originally proposed by Tethers Unlimited, Inc [1]. This system was selected for its detailed design and its relative near-term feasibility compared to larger, more ambitious MXER tether concepts. It provides a credible framework for testing the three actuator configurations shown in Fig. 1 and evaluating their impact on the performance of a representative MXER tether system. The key mass properties, tether characteristics, and orbital parameters of the baseline design are summarised in [2] and in the supporting material to this paper [3].

2.2 Tether configurations

The challenge of controlling tether dynamics to extend the payload rendezvous window has been approached using various actuator configurations. Previous work in this domain has predominantly focused on the control of librating tethers, which undergo a slow, pendular motion. This research adapts two such established configurations, depicted in Fig. 1, to the more dynamic context of a spinning MXER tether system to evaluate their comparative performance against the baseline Cislunar Tether Transport System configuration.

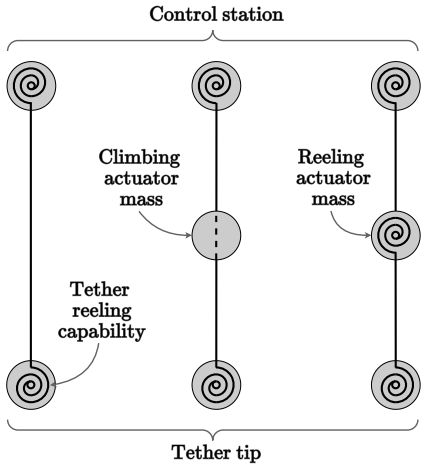


Fig. 1. Three different tether system configurations. From left to right, the baseline, climber and reeler configurations.

The first configuration, the baseline, involves reeling the tether in or out at the tether tip and control station. This directly controls the effective length of the tether, and consequently the position of the tip, by increasing or decreasing the length of the unspooled segment. All subsequent configurations share the ability to reel in or out portion of the tether at the control station (i.e. the counterweight) and tether tip.

The second configuration features a climbing actuator mass. This "climber" is an auxiliary mass that traverses the length of the tether, moving between the control station and the tether tip. During its motion, this intermediate mass remains constant, and does not affect the tether length. Its movement does however indirectly influence the tether tip's position by altering the overall mass distribution and dynamics of the system.

The third configuration, termed the "reeler", employs an intermediate actuator mass that divides the tether into two distinct segments. This intermediate mass, like the control station and tether tip, has reeling capabilities, though limited to the tether segment between it and the tether tip. By reeling the first segment in, the distance between the actuator and the tether tip is reduced. Similarly, reeling the second segment at the control station alters the distance between the actuator and the control station. Coordinated reeling of both segments allows for control over the position of the actuator mass relative to the tether's endpoints as well as the overall tether length, thereby controlling the system's dynamics and the ultimate position of the tether tip.

While the latter two configurations have been successfully applied to control librating tethers, their effectiveness in the high-speed, rotating environment of an MXER system has not been thoroughly investigated. This paper therefore undertakes a direct comparison of their potential to extend the payload capture window under these more demanding conditions.

2.3 Augmented-Lagrangian iLQR

Trajectory optimisation is a powerful framework for controlling complex dynamical systems, summarising an underlying optimisation problem that seeks to minimise a cost function subject to dynamics and path constraints. These problems are typically discretised in time, leading to discrete optimisation problems solved by either direct or indirect methods. Direct methods treat states and controls as decision variables and utilise general-purpose non-linear programming solvers, often transcribing the problem via direct collocation. While robust, direct methods can be slow and require large optimisation packages. In contrast, indirect methods such as the iterative Linear Quadratic Regulator (iLQR) exploit the Markovian structure of the problem, implicitly enforcing dynamics by simulating them forward [4]. These methods break the problem into a sequence of smaller sub-problems and improve on simpler shooting methods by incorporating a feedback policy during forward simulation. iLQR's fast computation low memory requirements can make it suitable for embedded implementation in some cases, however, the standard method is considered less numerically robust and not suited for handling non-linear state and input constraints without modification [5].

The augmented Lagrangian iLQR (AL-iLQR) approach adds constraint handling capabilities to the standard iLQR method [4]. The core of AL-iLQR lies in the conventional iterative LQR approach, where at each step, non-linear costs and dynamics are approximated using first order Taylor series expansions, transforming the problem into a sequence of discrete LQR sub-problems that optimise deviations about a nominal trajectory. After an initial rollout, the algorithm proceeds through a backward pass and a forward pass. In the backward pass, the optimal feedback policy (comprising feed forward and linear feedback gains) is computed by starting from the end of the trajectory and working backward, leveraging dynamic programming principles. Following this, the forward pass updates the nominal state and control trajectories by simulating the dynamics forward, applying these computed optimal feedback gains. A crucial component of the forward pass is a line search, which scales the feed forward term to ensure that each iteration results in an adequate reduction of the overall cost, driving the solution toward convergence. To handle additional path constraints, iLQR is embedded within an augmented Lagrangian framework. This involves augmenting the cost function with Lagrange multipliers λ and penalty terms μ for the constraints $g(x, u)$ as shown in Eq. 1. During each "inner" iLQR solve, these λ and μ terms are treated as constants. After the inner iLQR converges to a solution, the Lagrange multipliers are updated, and the penalty terms are monotonically increased. This process of inner iLQR solves followed by updates to the dual variables and penalties forms the outer loop of the augmented Lagrangian algorithm, iteratively improving constraint satisfaction and optimality [4].

$$J = x_f^T Q_f x_f + \int_{t_0}^{t_f} x^{\mathrm{T}} Q x + u^{\mathrm{T}} R u \, dt + \int_{t_0}^{t_f} \lambda^{\mathrm{T}} g(x,u) + \frac{1}{2} g(x,u)^{\mathrm{T}} I_\mu g(x,u) \, dt \quad (1)$$

A simple modification to the cost function in Eq. 1 enables state tracking, rather than just simple regulation toward zero. This is achieved by replacing the continuous and final states x, and x_f with the actual and target state differences of the form $\Delta x = x - x_{ref}$. The controller then determines the control values u such that $x \to x_{ref}$ subject to the constraints and the chosen cost coefficient matrices Q, Q_f and R.

The AL-iLQR process for finding a suitable trajectory is captured in the following steps:

1. Initialise a nominal control sequence and algorithmic parameters (e.g., tolerances, penalty coefficients, scaling factors).

2. Outer Loop (Augmented Lagrangian):

   (a) Inner Loop (iLQR):

   - Backward Pass: Linearise cost and dynamics around the nominal trajectory to compute feedback and feed forward terms.
   - Forward Pass: Roll out dynamics using the updated control policy and apply a line search to ensure cost reduction.
   - Repeat until inner convergence criteria are met (e.g., low cost reduction, small feed forward terms, or maximum iterations reached).

   (b) Update constraint terms: After inner iLQR convergence, adjust Lagrange multipliers (λ) and penalty scalars (μ) based on constraint violation.

   (c) Convergence check: Repeat outer loop if constraint satisfaction or cost reduction are insufficient.

2.4 Bayesian optimisation

Bayesian optimisation is a powerful methodology for optimising "black-box" objective functions. It is particularly well-suited for scenarios where objective functions are costly to compute, cannot be evaluated exactly (due to noise or indirect mechanisms), or offer no efficient mechanism for estimating gradient information [6]. This approach has demonstrated success across various fields, including sciences, engineering, and notably, hyperparameter tuning of complex machine learning models under limited observation budgets [7]. At its core, Bayesian optimisation relies on Bayesian inference, which systematically uses probability to reason about uncertain quantities, treating the objective function as a random variable. A statistical surrogate model, commonly a Gaussian process, is maintained to represent the current belief about the objective function, capturing assumptions like smoothness and correlations. This probabilistic model quantifies uncertainty across the domain. The optimisation policy then leverages this uncertainty by evaluating the merit of potential observation locations using an acquisition function, which balances exploitation (sampling promising regions) with exploration (sampling uncertain regions to gain information) [6].

A typical Bayesian optimisation process follows the steps outlined below [6]:

1. Initialise the dataset: Begin with either an empty dataset or a small set of initial observations. These initial points are typically collected using model-independent methods such as (quasi)random sampling.

2. Construct initial surrogate model: Use the initial data to construct a statistical surrogate model that approximates the objective function.

3. Iterative optimisation: Repeat the following steps until a termination condition is satisfied:

   (a) Inspect the current dataset D and surrogate model.

   (b) Select the next query location x by maximising the acquisition function over the input domain. This optimisation is significantly cheaper than evaluating the actual objective function.

   (c) Evaluate the true objective function at x to obtain the corresponding observation y.

   (d) Update the dataset with the new sample (x, y) and refine the surrogate model using Bayesian inference.

4. Check termination condition: This could be based on a fixed evaluation budget, convergence threshold, or other criteria.

5. Return the optimal solution: Identify and return the input x that is predicted to yield the maximum value of the objective function.

3. Rigid-body model

3.1 Orbital and geometric definitions

As part of the rigid-body tether model, the following assumptions are made: 1. the payload and the tether's COM follow unperturbed, counter-clockwise, circular and elliptic Keplerian orbits respectively, 2. the tether rotates about its COM in a prograde direction in the orbital plane, 3. no external forces act on the tether other than Earth's Newtonian gravity, 4. the tether is modelled as a continuous, rigid rod with a time-variable length, 5. the tether has a constant linear density along its length, 6. other tether components such as the counterweight (also known as the control station), tether tip, and any intermediate masses are modelled as point masses with time variable mass, 7. the tether length and point mass positions along the tether are controlled with reeling and unspooling of the tether inside each point mass. The orbital configuration resulting from these assumptions is shown diagrammatically in Fig. 2.

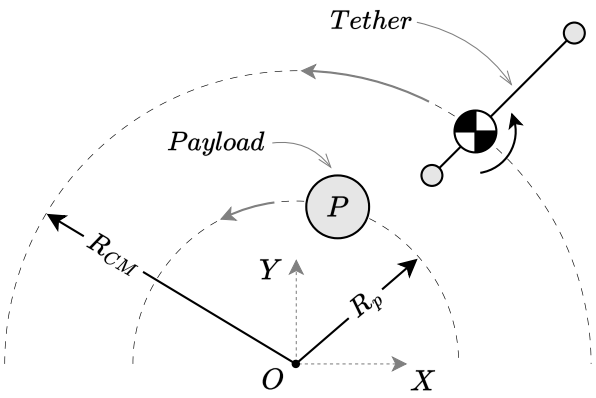


Fig. 2. Orbital configuration of the payload and tether in inner circular and outer elliptical orbits respectively, with the Earth-centred inertial frame OXY. Not to scale.

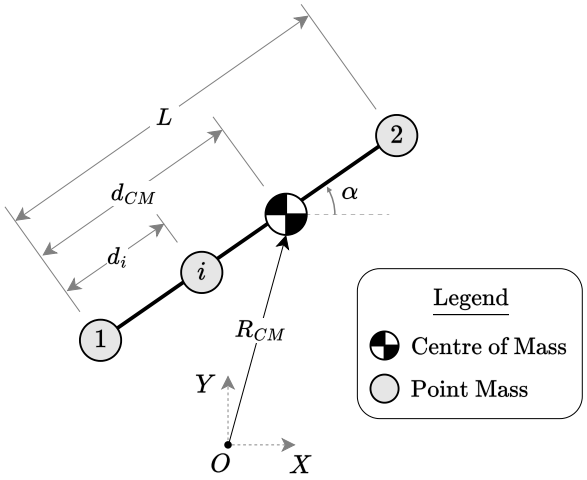


Fig. 3. Distance measurements of point masses along the tether relative to the tether tip, as well as the definition of the orientation angle relative to the OXY frame. Not to scale.

The payload and tether orbits are set up in such a way that the tether will align perfectly with the radial vector from Earth at the point of perigee. At this point in time, referred to as the nominal capture point, the tether tip and payload have the same position and velocity states.

The tether model accommodates two or more point masses, with the minimum requirement being the two end masses. The i^{th} point mass m_i is located at a distance d_i from the tether tip, which is designated m_1. Each d_i is measured in a body-fixed, tip-centred reference frame aligned with the tether's orientation. Similarly, the COM offset distance d_{CM} is measured from m_1. By definition $d_1 = 0$ and $d_2 = L$. The tether's orientation angle α is measured relative to the global horizontal. These definitions are depicted in Fig. 3. For the upcoming rotational kinematics, it is useful to express a point mass' distance from the tether system's COM as a scalar radius value given in Eq. 2. It is further helpful to express the tether system's orientation by a simple unit vector in the OXY frame, tangent to the tether length and defined to be positive from m_1 to m_2 in the body fixed frame

$$r_i = d_i - d_{CM} \tag{2}$$

$$\hat{\boldsymbol{e}}_t = \begin{bmatrix} \cos\alpha \\ \sin\alpha \end{bmatrix} \tag{3}$$

3.2 Kinematics

The ultimate goal is to control the tether system in such a way that the tether tip's state (position and velocity) is as close to the payload's state for as long as possible. It is therefore necessary to determine the state of the tether system's point mass(es) at any time during the simulation. This is done via kinematic relations. The position and velocity vectors of a generalised point mass i in the global OXY frame are given by

$$\boldsymbol{R}_i = \boldsymbol{R}_{cm} + r_i \hat{\boldsymbol{e}}_t \tag{4}$$

$$\dot{\boldsymbol{R}}_i = \dot{\boldsymbol{R}}_{cm} + \dot{r}_i \hat{\boldsymbol{e}}_t + \boldsymbol{\omega} \times r_i \hat{\boldsymbol{e}}_t \tag{5}$$

Eq. 4 and 5 are simply the kinematic relations for a moving point in a moving and rotating reference frame. The point mass vector results from these relations are dependent on the tether's rotation rate ω, the mass' location on the tether r_i, and the rate at which its location on the tether changes or $\dot{r}_i$.

3.3 Dynamics

During its operation, the tether system is subject to time-varying gravitational and control forces. These forces influence the $\dot{r}_i$ and ω values in Eq. 5, which in turn has an affect on Eq. 4. To study these time-varying affects, we turn our attention to the tether system dynamics.

Based on the measurement definitions in Section 3.1, the tether's scalar rotational equation of motion can be defined as

$$\tau_g = I_{CM}\,\dot{\omega} + \dot{I}_{CM}\,\omega \tag{6}$$

along with the vector-form equation of motion for the generalised point mass m_i in the OXY frame

$$\dot{m}_i \dot{\boldsymbol{R}}_i + m_i\,\ddot{\boldsymbol{R}}_i = \Delta \boldsymbol{T}_i + m_i\,\boldsymbol{g}_i \tag{7}$$

The continuous tether's equation of motion is given by

$$\rho\, \ddot{\boldsymbol{R}} = \boldsymbol{T}' + \rho\, \boldsymbol{g} \tag{8}$$

with ρ the tether density per unit length, and T' the spatial tension derivative along the tether.

Now that the equations of motions have been broadly defined, some of their terms will be defined in more detail. First, it should be noted that the gravitational torque on the left hand side of Eq. 6 includes the distributed gravitational forces on the massive tether and the forces on the discrete point masses. Its vector form is calculated at any point in time via

$$\begin{aligned}\boldsymbol{\tau}_g(t) &= \sum_{i=1}^{N} [\boldsymbol{d}_i \times \boldsymbol{F}_{g,i}] + \int_0^L \boldsymbol{d}_s(s) \times d\boldsymbol{F}_g \\ &= \sum_{i=1}^{N} \left[(d_i - d_{cm})\hat{\boldsymbol{e}}_t \times \left(-\frac{\mu_E m_i}{R_i^3} \right) \boldsymbol{R}_i \right] \\ &\quad + \int_0^L (s - d_{cm})\hat{\boldsymbol{e}}_t \times \left(-\frac{\mu_E \rho}{R_s(s)^3} \right) \boldsymbol{R}_s(s)\, ds\end{aligned} \tag{9}$$

The point mass and tether segment global position vectors in Eq. 9 can be further expanded to

$$\boldsymbol{R}_i = \boldsymbol{R}_{cm} + (d_i - d_{cm})\hat{\boldsymbol{e}}_t \tag{10}$$

$$\boldsymbol{R}_s(s) = \boldsymbol{R}_{cm} + (s - d_{cm})\hat{\boldsymbol{e}}_t \tag{11}$$

where $\boldsymbol{R}_s(s)$ is a parametrisation along the tether length with $s \in [0, L]$, and $s = 0$ defined at m_1. The tether system's moment of inertia is calculated from the contributions of the discrete masses and the continuos tether via Eq. 12. The moment of inertia time derivative, also used in Eq. 6, is given in Eq. 13.

$$I_{CM} = \sum_{i=1}^{N} \left[m_i r_i^2 \right] + \frac{\rho}{3} \left[(L - d_{CM})^3 + d_{CM}^3 \right] \tag{12}$$

$$\begin{aligned}\dot{I}_{CM} &= \sum_{i=1}^{N} \left[\dot{m}_i r_i^2 + 2 m_i r_i \dot{r}_i \right] \\ &\quad + \frac{\rho}{3} \left[(L - d_{CM})^2 \left(\dot{L} - \dot{d}_{CM} \right) + d_{CM}^2 \dot{d}_{CM} \right]\end{aligned} \tag{13}$$

The values for I_{CM} and $\dot{I}_{CM}$ depend on the COM offset as measured from the tether tip, and its rate of change. These are given by Eq. 14 and Eq. 15 respectively.

$$d_{CM} = \frac{1}{m_{tot}} \left(\frac{1}{2} \rho L^2 + \sum_{i=1}^{N} d_i m_i \right) \tag{14}$$

$$\dot{d}_{cm} = \frac{1}{m_{tot}} \left(\rho L \dot{L} + \sum_{i=1}^{N} \left[\dot{d}_i m_i + d_i \dot{m}_i \right] \right) \tag{15}$$

Each discrete point mass is modelled as having two mass components, namely a fixed and variable component as shown in Eq. 16. The fixed mass term encapsulates all non-reeling related mass of the point, while the variable mass term accounts for variation in the length of the spooled tether segment contained within the mass at any point in time. The mass rate of change of each point mass is thus directly proportional to the length rate with which the tether is reeled in or out at that mass, as shown in Eq. 15. Reeling in the tether at any point mass is defined as a positive mass (gain) rate.

$$m_i = m_{i0} + \rho L_i \tag{16}$$

$$\dot{m}_i = \rho \dot{L}_i \tag{17}$$

Returning to Eq. 7 for a moment, the acceleration term on the left hand side of Eq. 7, can be expanded using the rotating reference frame relation acceleration as in Eq. 18

$$\begin{aligned}\ddot{\boldsymbol{R}}_i &= \ddot{\boldsymbol{R}}_{CM} + \ddot{r}_i \hat{\boldsymbol{e}}_t + (2\boldsymbol{\omega} \times \dot{r}_i \hat{\boldsymbol{e}}_t) \\ &\quad + (\dot{\boldsymbol{\omega}} \times r_i \hat{\boldsymbol{e}}_t) + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times r_i \hat{\boldsymbol{e}}_t)\end{aligned} \tag{18}$$

Since the tether is modelled as a rigid-body, Eq. 18 not only applies to the point masses, but also to any point s along the tether if a parametrised value $r(s)$ is used instead of r_i. The s parametrisation is the same as specified in Eq. 11. This just leaves the ΔT_i term in Eq. 7 to be defined. This change in tension over a point mass relates the tension in the tether on either side of a point mass, as shown in Eq. 19. The two end points are special cases of this relation.

$$\begin{aligned}\Delta T_i &= T_{i+} - T_{i-} \\ \Delta T_1 &= T_1 - 0 = T_1 \\ \Delta T_2 &= 0 - T_2 = -T_2\end{aligned} \tag{19}$$

The generalised, scalar value for ΔT_i is found by substituting Eq. 11 and Eq. 18 into Eq. 7, and taking the dot product with $\hat{\boldsymbol{e}}_t$, which yields

$$\begin{aligned}\Delta T_i &= m_i \left[\langle \ddot{\boldsymbol{R}}_{cm}, \hat{\boldsymbol{e}}_t \rangle + \ddot{r}_i - \omega^2 r_i - \langle \boldsymbol{g}_i, \hat{\boldsymbol{e}}_t \rangle \right] \\ &\quad + \dot{m}_i \left[\langle \dot{\boldsymbol{R}}_{cm}, \hat{\boldsymbol{e}}_t \rangle + \dot{r}_i \right]\end{aligned} \tag{20}$$

Thus the two equations 6 and 7 are now fully defined in terms of the rotation state variables (α & ω), along with the d_i, $\dot{d}_i$, $\ddot{d}_i$ and L_i, $\dot{L}_i$, $\ddot{L}_i$ variables.

The d_i, $\dot{d}_i$, $\ddot{d}_i$ variables can be eliminated from the set of dependent variables by considering the model assumption of an inelastic tether, along with overall length conservation of the tether.

The overall tether consists of one or more segments, with each segment falling between between point masses.

The length of each segment is comprised of two components, namely the unspooled section and the spooled tether sections on each of the contributing point masses. For example, the first tether (i.e. the "baseline") configuration described in Section 2.2 has one tether segment, defined between each of the end masses m_1 and m_2. Both of these point masses have a portion of the tether segment spooled internally. Similarly, the third tether configuration (i.e. the "reeler" configuration) has two segments; one between m_1 and m_3, and the other between m_3 and m_2. For the reeler configuration, point masses m_1 and m_3 have a portion of the first tether segment spooled internally, while only m_2 has a portion of the second tether segment spooled internally. Under the model's assumptions of an inelastic tether and length conservation, the length of each segment is constant, and the length of all segments can be determined with the vector equation:

$$\boldsymbol{L}_S^{\mathrm{T}} = \boldsymbol{D}_i^{\mathrm{T}} + \boldsymbol{H}\boldsymbol{L}_i^{\mathrm{T}} \tag{21}$$

where the segment lengths, point mass distances from m_1 and spooled tether lengths inside each point mass are given by Eq. 22, 23, and 24 respectively.

$$\boldsymbol{L}_S^{\mathrm{T}} = \begin{bmatrix} l_{S1} & l_{S2} & \dots & l_{S(N-1)} \end{bmatrix} \tag{22}$$

$$\boldsymbol{D}_i = \begin{bmatrix} d_2 & d_3 & \dots & d_N \end{bmatrix} \tag{23}$$

$$\boldsymbol{L}_i = \begin{bmatrix} L_1 & L_2 & \dots & L_N \end{bmatrix} \tag{24}$$

The $\boldsymbol{H}$ in Eq. 21 is an $[N-1 \times N]$ matrix comprised of element-wise Heaviside step functions that effectively determine which point masses have a spooled portion of each tether segment. The full $\boldsymbol{H}$ matrix is given in Eq. 25.

$$\boldsymbol{H} = \begin{bmatrix} H(d_2-d_1) & H(d_2-d_2) & \dots & H(d_2-d_N) \\ H(d_3-d_1) & H(d_3-d_2) & \dots & H(d_3-d_N) \\ \vdots & \vdots & \ddots & \vdots \\ H(d_N-d_1) & H(d_N-d_2) & \dots & H(d_N-d_N) \end{bmatrix} \tag{25}$$

The Heaviside step function $H(x)$ is defined as 0 for $x < 0$ and 1 for $x \geq 0$.

For the case of three point masses, the $\boldsymbol{H}$ matrix simplifies to a constant $[2\times3]$ matrix since the third point mass always falls between m_1 and m_2, such that $d_1 < d_3 < d_2$, with $d_1 = 0$ and $d_2 = L$ by definition. This makes the $\boldsymbol{H}$ matrix trivial to invert, and the point mass local distance values can be easily determined by solving for $\boldsymbol{D}_i$ from Eq. 22 given that elements of $\boldsymbol{L}_i$ are known.

The values for $\dot{d}_i$ and $\ddot{d}_i$ can be determined in terms of $\dot{L}_i$ and $\ddot{L}_i$, by taking the first and second time derivative of Eq. 21, respectively. The constant $\boldsymbol{L}_S$ vector falls away under differentiation, resulting in Eq. 26 and Eq. 27

$$\dot{\boldsymbol{D}}_i = -\boldsymbol{H}^{\mathrm{T}}\dot{\boldsymbol{L}}_i \tag{26}$$

$$\ddot{\boldsymbol{D}}_i = -\boldsymbol{H}^{\mathrm{T}}\ddot{\boldsymbol{L}}_i \tag{27}$$

After this variable elimination process, the dynamic system is fully defined by the time t, the rotation state α and ω, and the L_i, $\dot{L}_i$, $\ddot{L}_i$ values for each point mass.

It should be noted that the $\boldsymbol{H}$ matrix in Eq. 25 undergoes slight modification for the case of the second "climber" tether configuration. Since the intermediate climber mass does not accumulate nor disperse part of the tether, and the tether simply passes through it, all its contributions to the $\boldsymbol{H}$ matrix are set to zero, except for the case where $H(d_3 - d_3) = 1$. This is done because the L_3, $\dot{L}_3$, and $\ddot{L}_3$ take on a different meaning in this case, as the position, velocity and acceleration relative to this point mass' starting point on the tether, rather than the length, length rate, and length acceleration of the spooled tether. This was done such that the same model can be used for all three configurations with only minor changes.

3.4 Tether tension

Next, the attention is turned toward determining the tension along the tether. By solving for $\ddot{\boldsymbol{R}}$ from Eq. 8, and equating this result to that of Eq. 18, an expression for the spatial tension derivative along the tether is found. Since the tension is only defined along the tether direction, taking the dot product of this tension derivative expression with the orientation unit vector yields a scalar differential equation for $T'(s)$.

$$T'(s) = \rho\Big[\langle \ddot{\boldsymbol{R}}_{cm}, \hat{\boldsymbol{e}}_t\rangle + \ddot{r}(s) - \omega^2 r(s) - \langle \boldsymbol{g}(s), \hat{\boldsymbol{e}}_t\rangle\Big] \tag{28}$$

By noting that the tension jumps described by Eq. 20 are effectively boundary conditions at each of the point masses, the tension distribution along the tether can be determined at each point in time by solving Eq. 28 as a boundary value problem. Under the modelling assumptions of a rigid tether, the $\ddot{r}(s)$ term is everywhere along the tether, except at the point masses where it reduces to $\ddot{r}_i$. For moderate $\ddot{r}_i$ values, Eq. 28 is dominated by the centrifugal $\omega^2 r(s)$ terms, and thus the tension profile effectively follows a quadratic profile between point masses, with discrete jumps in tension before and after a point mass.

3.5 Tether control

Though there are various ways of controlling the tether dynamics described in previous sections, this work opted to use the $\ddot{L}_i$ reeling acceleration values for each point

mass as control inputs. This choice has the advantage of being a straight forward and independent control input into the system's dynamics, while also ensuring state continuity despite non-smooth and possibly interrupted control commands. By specifying the $\ddot{L}_i$ value for each point mass at each point in time, the remaining $\dot{L}_i$, L_i, α and ω state components can be determined by integration from an initial state.

As previously noted, each point mass has a fixed length of internally pre-spooled tether, except for the intermediate actuator mass of the climber configuration. Thus the control implementation should guard against the depletion of this pre-spooled length below zero and prevent it from increasing beyond a specified limit value L_i^* during the simulation of the controlled dynamics. This can be achieved with a simple variable transformation of the spooled length value L_i to variable z_i with the use of a scaled sigmoid function given in Eq. 29. The transformation keeps $L_i \in [0, \bar{L}_i]$, while letting z_i change as needed.

$$L_i = L_i^* \sigma(z_i) \quad \text{with} \quad \sigma(z_i) = \frac{1}{1 + \exp(-z_i)} \tag{29}$$

Since $\ddot{L}_i$ serves as the control input, an expression for its transformed counterpart $\ddot{z}_i$ is needed. Taking the first and second time derivatives of Eq. 29 and solving for the $\dot{z}_i$ and $\ddot{z}_i$ values yield

$$\dot{z}_i = \frac{\dot{L}_i}{L_i^* \sigma[1-\sigma]}, \; \ddot{z}_i = \frac{\ddot{L}_i}{L_i^* \sigma[1-\sigma]} - \dot{z}_i^2[1-2\sigma] \tag{30}$$

With this transformation, Eq. 29 can be used to simulate the spooled length dynamics of the tether system in the z_i domain, while still accepting the $\ddot{L}_i$ control inputs and allowing the L_i and $\dot{L}_i$ values to be calculated as needed. Crucially, this transformation has no effect on the rotation state variables α and ω, which are used as described in Section 3.3.

3.6 Payload capture

It should be noted that this work does not consider the physical capture of the payload or the dynamics resulting from payload capture, but rather focuses on extending the rendezvous window before capture. However, the model derived in this section can be adapted to include such a capture. This is done by updating the tip mass to account for the payload mass and adjusting the tether orbit's semi-major axis to reflect the momentum transfer at the point in time where capture occurs. The remainder of the model remains valid and can be used without any further modifications.

4. Method

4.1 Equivalent configurations

To enable a fair and meaningful comparison between the proposed tether configurations, several key requirements were imposed. Firstly, all configurations were constrained to operate within identical orbital parameters for both the payload and the tether system. Secondly, each tether was required to maintain the same centre-of-mass (COM) offset from the tether tip, ensuring that the tether tip could reach the payload precisely at the perigee point during capture. Additionally, the total system mass was held constant across all configurations, and the rotational kinetic energy was preserved to ensure dynamic equivalence.

Beyond these constraints, the tether length and its linear mass density were also kept constant. This uniformity in physical parameters ensured that any observed differences in system behaviour could be attributed solely to the configuration of the tether masses, rather than to variations in fundamental properties. To further ensure equivalence, the moment of inertia of each tether system was matched as closely as possible to that of the baseline configuration. This was critical in maintaining a consistent angular velocity, ω, for the uncontrolled tether, thereby ensuring that the tip speed during an uncontrolled approach to the payload remained nearly identical across all configurations. Achieving this required solving a constrained optimisation problem. Specifically, the objective was to minimise the squared difference between the target and actual moments of inertia, subject to the constraints of equal total mass and identical COM offset. Once the optimal mass distribution is identified, the angular velocity ω can be computed using the relationship derived from rotational kinetic energy, given by $E_{rot} = \frac{1}{2} I_{CM} \omega^2$.

A basin hopping algorithm [8] from the SciPy library was employed to solve this optimisation problem, with multiple runs conducted to ensure convergence to a global minimum. The optimisation yielded the mass values for the three point masses comprising the tether system, as well as the local position of the actuator mass relative to the tether tip. The resulting parameters are summarised in Table 1. The final moment of inertia and angular velocity differed from the baseline configuration by only 0.006% and 0.003%, respectively. Consequently, the variation in the tether tip's uncontrolled approach velocity was negligible, less than one centimetre per second, rendering the three configurations effectively indistinguishable in this regard.

Table 1. Tether system mass distribution parameters for the baseline, climber and reeler tether configurations.

Parameter	Baseline	Climber & Reeler	Unit
m_1	650	615	kg
m_2	15004	14640	kg
m_3	-	615	kg
d_3	-	59.18	km

4.2 Uncontrolled baseline performance

To get a baseline for the rendezvous window duration, the uncontrolled reference tether (baseline) configuration was simulated over a 12 second window using the RK45 integrator with state evaluations every $\Delta t = 0.1$ seconds. This simulation was centred at perigee with the nominal capture point at $t = 0$ with $t \in [-6, 6]$ seconds. Realistically, performing payload capture at an instant is unfeasible. Thus the capture mechanism at the end of the tether tip needs to accommodate positional capture errors of several metres and velocity capture errors of a couple of metres per second [9] [10]. A Monte-Carlo analysis of the model presented in Section 3 was performed in supporting work [3] to test the model's response to uncertainties in its defining parameters. This analysis showed that a positional tolerance of 10 metres was sufficient to accommodate all positional capture errors for the modelling assumptions specified in Section 3. A liberal value of 10 m/s was applied to the velocity tolerance, although this value is at the upper end of what can be considered reasonable for a capture mechanism (similar to that presented in [9]). The capture requirements for the tether system can of course be relaxed slightly if the payload contributes to error correction through corrective manoeuvres at the expense of propellent expenditure.

To formalise the concept of the rendezvous window, we first define the set of time instances during which both the position and velocity of the tether tip remain within specified tolerances relative to the payload. Mathematically, this is expressed in Eq. 31 as

$$S_W = \left\{ t \in [t_0, t_f] \,\middle|\, \Delta R(t) < \epsilon_r \ \wedge \ \Delta \dot{R}(t) < \epsilon_v \right\} \tag{31}$$

$$\Delta R(t) = \| \boldsymbol{R}_{\text{tip}}(t) - \boldsymbol{R}_{\text{payload}}(t) \| \tag{32}$$

$$\Delta \dot{R}(t) = \| \dot{\boldsymbol{R}}_{\text{tip}}(t) - \dot{\boldsymbol{R}}_{\text{payload}}(t) \| \tag{33}$$

where $\epsilon_r = 10$ m and $\epsilon_v = 10$ m/s are the position and velocity tolerances, respectively. The rendezvous set S_W thus represents the set of time intervals during which the tether tip is sufficiently close to the payload in both position and velocity for the capture mechanism to initiate capture. The rendezvous window W is then defined as the length (in seconds) of the longest continuous subinterval $\tau \subseteq S_W$ of this set.

$$W = \max_{\tau \subseteq S_W} (\text{length}(\tau)) \tag{34}$$

This duration serves as the performance metric for the optimisation process described in the following section.

4.3 Unconstrained iLQR Control

To compare the respective rendezvous window extension performance of the respective configurations, the AL-iLQR controller was used to control the tether dynamics without any constraints. By dropping the constraint terms from Eq. 1 the AL-iLQR is reduced to a conventional iLQR controller. The trajectory-tracking form of the iLQR controller was used to drive the tether tip's X and Y position and velocity components to toward matching those of the moving payload as measured in the global OXY frame. Although no general state or control constraints were imposed in this section, saturation limits were placed on the control values such that $\ddot{L}_i \in [-100, 100]$ m/s^2. These limit values served as cut-off points to prevent excessive control commands.

Despite similar overall tether motion between configurations, their tip behaviour can differ due to the different actuation methods and degrees of freedom present in these configurations. Thus, the iLQR cost coefficient matrices $\boldsymbol{Q}$, $\boldsymbol{Q}_f$ and $\boldsymbol{R}$ were tuned independently for all three configurations. For finer tuning of the controller, the $\boldsymbol{Q}$ and $\boldsymbol{Q}_f$ matrices were further subdivided into positional and velocity components according to Eq. 35 and Eq. 36. The $\boldsymbol{R}$ matrix was defined as a diagonal $[N \times N]$ matrix for the tether's N point masses, with a single R_u value along the diagonal.

$$\boldsymbol{Q} = \text{diag}(Q_r, Q_r, Q_v, Q_v) \tag{35}$$

$$\boldsymbol{Q}_f = \text{diag}(Q_{fr}, Q_{fr}, Q_{fv}, Q_{fv}) \tag{36}$$

During experimentation with the different tether configurations, it was noted that configurations performed better or worse at different simulation durations. For this reason the simulation duration was added to the $\boldsymbol{Q}$, $\boldsymbol{Q}_f$ and $\boldsymbol{R}$ matrices (and sub-elements where applicable) as a tuning parameter.

The tuning the controller for each configuration was formulated as an optimisation problem where the rendezvous window, as defined in Section 4.2, should be max-

imised. For this optimisation the diagonal elements of $\boldsymbol{Q}$, $\boldsymbol{Q}_f$ and $\boldsymbol{R}$ were bounded to the range $[10, 10^4]$ as continuous variables, and the simulation duration was limited to the categorical set of $[8, 9, ..., 15]$ seconds.

The optimisation was performed with the Optuna library, which is an open source hyperparameter optimisation framework that makes use of Bayesian optimisation [7]. This optimisation framework is ideal for the controller tuning, since the calculation of the rendezvous window is effectively a black-box objective with no efficient options for estimating gradient information. As a general rule of thumb for the use of Bayesian optimisation, it is typically recommended to use a number of trials equal to ten times the number of decision variables (or more, depending on resource availability and time constraints). This recommendation is made such that the underlying Bayesian surrogate model can create a useful approximation of the objective function. Thus 60 trials were used for the 6 decision variables of each configuration's tuning process.

The three configurations were then compared based on the rendezvous window achieved with their respective tuned controller values, with the best performing configuration being selected for further analysis under constrained AL-iLQR control. These results are discussed in Section 5.

4.4 Constrained AL-iLQR Control

The best-performing configuration identified in the unconstrained iLQR control section was subsequently subjected to trajectory tracking under realistic operational constraints using an Augmented Lagrangian iLQR (AL-iLQR) controller. The previously tuned cost matrices $\boldsymbol{Q}$, $\boldsymbol{Q}_f$ and $\boldsymbol{R}$ were retained for consistency.

The following constraints were imposed to reflect practical limitations in tether system design and operation:

- **Tension limits**: Tether tension was constrained between $T_{\text{min}} = 0.1$ kN to maintain positive tension and $T_{\text{min}} = 472$ kN, based on Spectra 2000 material properties and safety factors from the Cislunar Tether Transport System baseline [1].
- **G-load limits**: A 5g acceleration limit relative to the COM was applied to all point masses, reflecting tolerable transient loads observed in Quad-Trap mechanism testing [11]. Of the tether's point masses, the tether tip is expected to experience the highest accelerations due to centripetal effects.
- **Reeling power limits**: Reeling actuators were limited to $P_{\text{max}} = 2$ MW, based on first order estimates from high-performance motor power densities [12]. Though this preliminary value serves as a starting point, a dedicated study is recommended to establish more accurate power constraints. The limit applies only during tether retraction, when actuators must overcome tension forces.
- **Reel-out rate limits**: The reel-out velocity was capped at $\dot{L}_{\text{max}} = 100$ m/s, based on deployment rates observed in prior tether deployment and rendezvous studies [13].
- **Reel-out acceleration limits**: To prevent unrealistic outward acceleration commands, the reel-out acceleration was bounded by the natural centrifugal and gravitational acceleration experienced by each point mass. This was enforced by requiring a positive tension jump $\Delta T_i > 0$ for point masses between the COM and the tether tip (inclusive), and a negative tension jump $\Delta T_i < 0$ for those between the COM and the counterweight, as defined in Eq. 20.

These constraints ensure that the controller operates within feasible physical limits, while still enabling effective trajectory tracking. To further evaluate the system's adaptability and performance, the same configuration was subsequently tested under RL-based control using a custom environment. This transition is discussed in the following section.

4.5 Tether RL Environment

The tether dynamics derived in Section 3 were used to construct a custom reinforcement learning (RL) environment, implemented according to the Farama Gymnasium API standard [14]. This standard provides a unified interface for a wide range of RL libraries, including the Stable-Baselines 3 library employed in this work. Within the environment, the observation and action spaces were defined, along with a reward function designed to guide the training of RL agents.

4.5.1 Observations

As the primary objective of the RL controller is to minimise the difference between the tether tip and payload states, the observation vector was defined as the element-wise difference between their respective state vectors. This difference serves as an error signal to be reduced over time. To provide temporal context, the current simulation time t was appended to the observation vector. The complete observation vector is given in Eq. 37:

$$O_{RL}(t) = [\Delta x,\ \Delta y,\ \Delta v_x,\ \Delta v_y,\ t] \tag{37}$$

Although not explicitly encoded within the environment, these observations were normalised using the `VecNormalize` functionality available in Stable-Baselines 3. This normalisation ensures that all observation elements are scaled comparably before being passed to the underlying neural networks, which is essential for stabilising learning. This is particularly important in the present context, as the observation elements span different orders of magnitude.

4.5.2 Actions

The RL agent was tasked with controlling the tether system using the same reeling acceleration parameters $\ddot{L}_i$ as those used in the AL-iLQR controller. For each point mass, the agent's action space was normalised to the range [-1, 1], which was then scaled to the physical saturation limits of [−0.1, 0.1] km/s^2 for the $\ddot{L}_i$ values. This scaling mirrors the control constraints imposed in both the unconstrained and constrained AL-iLQR control scenarios, ensuring consistency across control methodologies.

4.5.3 Reward Function

The reward function used for RL control closely mirrors the cost function of the AL-iLQR controller, with a sign inversion to reflect the maximisation objective typical in RL. The reward was composed of six distinct components:

1. A quadratic penalty for non-zero state tracking errors.
2. A quadratic penalty for control effort.
3. A fixed penalty applied at each time step.
4. A positive reward that increases linearly for each time step during which the tether tip remains within the desired tracking tolerance.
5. A potential-based shaping reward that incentivises consecutive reductions in the tracking error signal.
6. A quadratic penalty for any violated constraints.

Each component is briefly described, along with its mathematical formulation.

The quadratic penalties for the state tracking errors and control effort behave in the same way as the constraint-free cost function of the AL-iLQR controller. The tether tip and payload state difference vector $\Delta \boldsymbol{X}$ and control vector $\boldsymbol{u}$ are scaled with square, diagonal matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ respectively as shown in Eq. 38.

$$R_{\text{tracking error}} = -\Delta \boldsymbol{X}^{\mathrm{T}} \boldsymbol{A} \Delta \boldsymbol{X}, \; R_{\text{effort}} = -\boldsymbol{u}^{\mathrm{T}} \boldsymbol{B} \boldsymbol{u} \quad (38)$$

with the coefficient matrices defined as square diagonal matrices with elements A_{track} and B_{effort} respectively.

$$\boldsymbol{A} = \boldsymbol{I} \cdot A_{\text{track}} \quad \text{and} \quad \boldsymbol{B} = \boldsymbol{I} \cdot B_{\text{effort}} \quad (39)$$

Adding a fixed penalty at each time step of $R_{\text{step}} = -C_{\text{step}}$ makes all states slightly undesirable. This component motivates the agent to take actions to find a less bad state. To specifically encourage the agent to maintain stable tracking, a bonus is awarded that increases linearly with the number of consecutive time steps the tether tip remains within the predefined tolerance window, as shown in Eq. 40. This component directly incentivises persistence, making a policy that stabilises within the target zone more valuable than one that repeatedly enters and exits.

$$R_{\text{stay}} = C_{\text{stay}} \cdot N_t \quad (40)$$

where N_t is the count of consecutive time steps in tolerance and C_{stay} is a scaling hyperparameter. Three alternative reward schemes for R_{stay} were also evaluated: a constant bonus C_{stay} awarded whenever tracking error was within tolerance; a Gaussian reward increasing as error decreased below the threshold; and a curriculum-based scheme applying the fixed bonus for sub-tolerance tracking, with tolerances linearly reduced from 100 m & 100 m/s to 10 m & 10 m/s over the first 500k of 2M steps. None of these alternative approaches yielded sustained rendezvous windows.

A potential-based shaping reward is included to provide a dense, policy-invariant guidance signal towards the reference trajectory [15]. This component rewards the agent for any action that reduces the tracking error from one step to the next, accelerating learning without altering the optimal policy of the true objective.

$$R_{\text{shaping}}(t) = \gamma \Phi(t+1) - \Phi(t) \quad (41)$$

where γ is the reward discount factor and $\Phi(t) = -k_{\text{shape}} \cdot \|\Delta \boldsymbol{X}(t)\|$ is the potential function based on the tracking error norm, with k_{shape} a scaling hyperparameter. Finally, to ensure the agent operates within physical limits, a penalty is applied for any violation of the constraints defined in Section 4.4. This penalty scales quadratically with the magnitude of the violation, creating a strong deterrent that strongly encourages the agent to learn solutions that adhere to the system's operational envelope.

$$R_{\text{constraints}}(t) = -\sum_i w_i \cdot \max(0, g_i)^2 \quad (42)$$

where $g_i > 0$ represents the magnitude of the i^{th} constraint violation and w_i is its corresponding weight.

These components collectively form a dense reward

structure, meaning that a reward signal is issued at every time step of the training episode. Dense rewards are particularly beneficial in continuous control tasks, as they provide frequent feedback to the agent, accelerating convergence and improving sample efficiency. In contrast to sparse rewards, which may only be issued at the end of an episode or upon achieving a specific goal, dense rewards help guide the agent through intermediate states and encourage incremental improvements in performance.

4.6 Unconstrained RL control

Similar to the unconstrained iLQR control, the best performing tether configuration was first tested in an unconstrained RL environment before moving to the more challenging constrained case. This was done by excluding the control effort and constraint components listed in the previous section from the environment reward.

4.6.1 Hyperparameter and reward parameter tuning

Within the context of the unconstrained environment, the performance of the SAC RL agent was optimised by tuning a total of ten parameters, the four unconstrained reward function coefficients (A_{track}, C_{step}, C_{stay}, k_{shape}), and six underlying RL model hyperparameters.

The six key hyperparameters of the chosen Soft Actor-Critic (SAC) algorithm were identified for tuning by consulting common practices documented in the RL Baselines3 Zoo [16], a repository of tuned agents. These parameters include:

- **Learning Rate**: Controls the step size for gradient updates.
- **Batch Size**: Number of samples drawn from the replay buffer per update.
- **Replay Buffer Size**: The maximum number of transitions stored for experience replay.
- **Tau (τ)**: The smoothing coefficient for updating the target networks.
- **Discount Factor (γ)**: Determines the importance of future rewards.
- **Network Architecture**: The number and size of hidden layers for the policy and Q-value networks.

The specific values considered for these hyperparameters are detailed in the supporting material for this paper [3].

A preliminary analysis indicated that a fully robust optimisation, involving multi-seed evaluations for every trial, would be computationally prohibitive. Therefore, a pragmatic, two-phase optimisation strategy was designed and implemented to find a high-performing parameter set within a reasonable time-frame.

The core of the approach was an Optuna study that combined the reward parameter and RL hyperparameter tuning. This combined approach was chosen over sequential tuning to capture the potentially complex interplay between the reward signal and its learning hyperparameters. By tuning all ten parameters simultaneously, the optimisation framework could explore these interdependencies and discover more effective solutions. To balance thoroughness with computational feasibility, the following steps were taken:

1. **Phase 1: High-Throughput Discovery**
   The primary optimisation was conducted over 100 trials, where each trial involved training an agent with a single, fixed seed. This approach intentionally traded per-trial robustness for a significantly broader and faster exploration of the 10-dimensional hyperparameter space. The primary trade-off here was accepting that the performance value for any single trial could be influenced by the stochastic initialisation of beneficial RL neural network parameters. An additional assumption was made to complete the training process within a reasonable timeframe, namely that agents that learn to perform well early on in their training will maintain improved performance in extended training runs. This assumption allowed the optimisation trials to be performed at a reduced number of overall training time steps, further improving computational feasibility.

2. **Custom Optimisation Metric**
   The objective function for the optimisation was not the environment's raw reward signal. Instead, a custom callback was implemented to calculate a more task-relevant metric, namely the the rendezvous window defined in Section 4.2. This ensured the optimisation was directly aligned with the project's primary performance goal.

3. **Phase 2: Robust Validation**
   Upon completion of the discovery phase, the top three candidate parameter sets were identified. Each of these candidates was then subjected to a more robust validation process. This involved retraining an agent with each candidate set three separate times, each time using a different random seed. The retraining took place over double the number of training time steps used in phase 1 to allow further agent learning. The final performance of each candidate was determined by the average of its custom metric across these multiple runs. This step served to filter out any

"lucky" trials from the discovery phase and identify the parameter set that was most consistently effective. The results for this phase are presented and discussed in Section 5.2.

This two-phase strategy made a deliberate trade-off of accepting noise and uncertainty during the initial wide-ranging search, and then systematically eliminating that uncertainty for a small number of top candidates. This hybrid approach enabled a comprehensive yet computationally tractable optimisation study, leading to the selection of a robust and high-performing final RL agent.

4.7 Constrained RL control

Following the unconstrained optimisation, the best-performing hyperparameters and reward coefficients were applied to the constrained environment. This phase of the investigation was designed to evaluate the agent's performance under more realistic operational conditions. To this end, the reward function was augmented with two additional components, R_{effort} and $R_{\text{constraints}}$, mirroring the approach taken for the constrained AL-iLQR controller in Section 4.4.

5. Results and Discussion

5.1 Unconstrained iLQR control

The results of the independent cost function coefficient tuning optimisations are presented in Table 2. The main points of interest are the achieved rendezvous window durations when each configuration is subject to iLQR control under their respective tuned cost coefficient values. These durations are 0.8 s, 1.0 s and 1.8 s for the baseline, climber and reeler configurations respectively. All of the controlled configurations improve on the uncontrolled tether's rendezvous window of 0.6 s. From these results it is immediately clear that the reeler configuration outperforms the other two configurations by a significant margin. The climber configuration only slightly improves on the baseline. The significance of the difference between these two values is further reduced when considering the simulation time step of $\Delta t = 0.1$ s, which necessarily means these rendezvous window values are accurate to $\pm$ 0.05 s. Taking this uncertainty into account means that in the worst case, the climber configuration only improves upon the baseline by 1 second. The same applies when comparing the controlled baseline performance to the uncontrolled case. Though a reduced Δt value will shrink the uncertainty associated with these rendezvous window values, the practical worth of reduced uncertainty is limited for this work, as the modelling assumptions limit the true accuracy of the derived rigid-body dynamic model. Of higher importance is that the controlled tether configurations improve upon the uncontrolled case and upon each other with margins that exceed the uncertainties.

It is also interesting to note from Table 2 that the optimal simulation time found for the climber configuration is less than that of the other two configurations. As a sanity check, multiple different simulation durations were tested for all configurations, and specifically a longer simulation duration of 11 s was also tested for the climber configuration with the tuned control coefficients, but these tests yielded worse rendezvous window values. The uncontrolled case saw the same 0.6 s window regardless of simulation time.

Table 2. Tuned iLQR cost function coefficients, simulation duration and resulting rendezvous window for the unconstrained control of the baseline, climber and reeler tether configurations.

Parameter	**Uncontrolled**	**Baseline**	**Climber**	**Reeler**
Q_r	-	6183	1096	4578
Q_v	-	8727	2306	4206
Q_{fr}	-	289	330	1282
Q_{fv}	-	510	1316	46
R	-	1232	11	101
Simulation duration [s]	11	11	8	11
Rendezvous window [s]	0.6	0.8	1.0	1.8

The tether tip trajectories for the uncontrolled, and controlled baseline, climber and reeler configurations are compared against the payload trajectory in Fig. 4. From the relative position and velocity magnitudes in Fig. 4a and 4b it is seen that while the tether tip position is closely matched to that of the payload with relative ease, the relative velocity difference is much harder to keep within the desired tolerance. This is due to the fundamental rotating tether behaviour which sees the tether tip approach the payload with a high downward vertical velocity, momentarily coming to a stop from the payload perspective, and then rapidly moving away with a near vertical upward velocity. The behaviour described here is evident in the V-shaped curves in Fig. 4b. The reeler configuration ultimately outperforms the other configurations due to its ability to "flatten" its relative velocity curve in the vicinity of the payload more than the other configurations, thereby prolonging the

time spent within the velocity tolerance. Additional state-wise plots are provided in the supporting material to this work [3].

The performance differences between the baseline, climber and reeler configurations can be attributed to their controllable degrees of freedom (DOF) and how these affect the kinematics of the tether tip. The climber configuration has one additional DOF compared to the baseline, which mainly affects the I_{CM} and thus the tether system's rotation rate and thereby the tip velocity through the cross-product term in Eq. 5. While the $\dot{r}_1$ term in Eq. 5 (as applied to the tip mass m_1) does also change when the climber mass is actuated, this change is limited to the small corresponding change of the COM offset as per Eq. 2. The reeler configuration, in addition to the moment of inertia (and thus rotation rate), also affects the length of the tether tip's rotation arm and its rate of change. Thus the reeler configuration has more substantial control over the last two terms of Eq. 5.

5.2 Unconstrained RL control

The training curves for the three top performing unconstrained RL optimisation trials are depicted in Fig. 5. Both the average return in Fig. 5a and the rendezvous window duration in Fig. 5b show noisy trends that generally tend upward. These noisy upward trends are indicative of RL agent learning instability. In both plots trial 91 comes out as the top performer after the retraining process described in Section 4.6.1, though there is significant variation in its performance over the multiple retraining runs. Similar variation is seen for the other two top performing trials. Such noisy curves and high variation after hyperparameter and reward tuning imply that, although the agent can learn, its results are not robust and may be difficult to reproduce, neither of which are desirable characteristics for a system controller. The best performing trial was further trained to 6 million steps, but showed no further improvement in rendezvous extension.

The relative position and velocity magnitudes of the RL-controlled, unconstrained trajectory are plotted against the uncontrolled case in Fig. 6. The RL agent's performance was found to be dependent on the simulation start time, producing better results for a slightly asymmetric simulation starting at $t_0 = -7$ s and continuing until $t_f = 6$ s. During this simulation window, the RL controlled reeler configuration achieves a rendezvous window of 1.8 s, matching that of the iLQR controlled reeler configuration. As with the iLQR controlled case, the RL agent manages to track the payload's position reasonably well, while the velocity tracking again proves to be the more challenging task. Compared to the iLQR case, the RL controller shows an increased ability to flatten the V-shaped relative velocity curve in the vicinity of the payload. This is evident in Fig. 6b around the $t = 0$ s mark. The resulting relative velocity curve in Fig. 6b is noticeably less smooth than its iLQR counterpart indicating sporadic actuator use. This behaviour is undesirable in practical tether applications, as it may induce elevated tension loads and can excite unmodelled high-frequency wave-like dynamics, thereby undermining the tether control of the tether system.

5.3 Constrained AL-iLQR and RL control

The training curves for the constrained RL environment are shown in Fig. 7. Unlike the constrained case, Fig. 7a shows a definite and early sign of plateaued learning. The initial extreme negative returns are due to constraint violations, and the return values quickly rise as the agent learns that constraint violations are undesirable. At the scale of the initial returns, the curve in Fig. 7a may look smooth, but upon closer inspection via the inset plot and its scale, it is evident that the results remain somewhat noisy and variant. Fig. 7b indicates that no reasonable rendezvous window duration was achieved. As with the unconstrained case, the agents were further trained to 6 million steps, with no noteworthy improvement in the rendezvous window result.

The resulting trajectories for both the constrained AL-iLQR and constrained RL controllers are presented in Fig. 8. In both cases, no rendezvous window was identified in which the position and velocity tolerance criteria were simultaneously satisfied. As shown in Fig. 8a, neither controller successfully tracked the payload's positional trajectory: the RL controller failed to reach the required tolerance altogether, while the AL-iLQR controller overshot the target. For the relative velocity magnitude components illustrated in Fig. 8b, both controllers briefly entered the tolerance range, as indicated by the V-shaped curves touching the dashed payload reference line. However, these brief intersections offer no meaningful improvement over the uncontrolled baseline, rather worsening the rendezvous performance for the considered tolerances.

Importantly, the AL-iLQR controller produced a trajectory that strictly satisfied all imposed constraints, though none were active throughout the motion. The constraint closest to activation was the g-load on the tether tip, which reached approximately 3g which is well within the allowable 5g limit and comparable to the uncontrolled case. This suggests that while the controller respects the constraints, it does not fully exploit them to optimise performance.

In contrast, the RL controller briefly activated the maximum tension and tether tip g-load constraints early in the

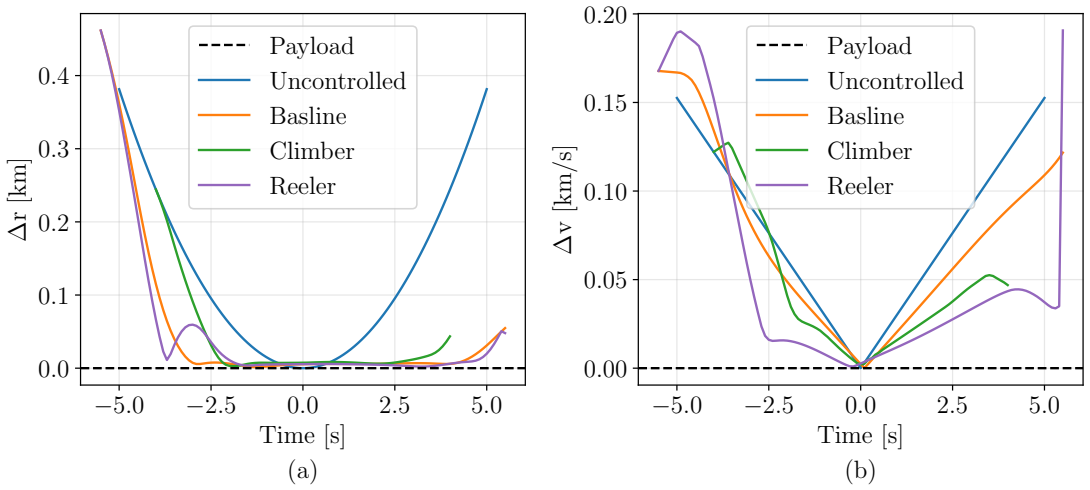


Fig. 4. Comparison of the tip trajectories of the uncontrolled tether against the iLQR controlled baseline, climber and reeler configurations. The subplots (a) and (b) show the magnitudes of the relative position and velocity difference vectors between the tether tip and payload respectively.

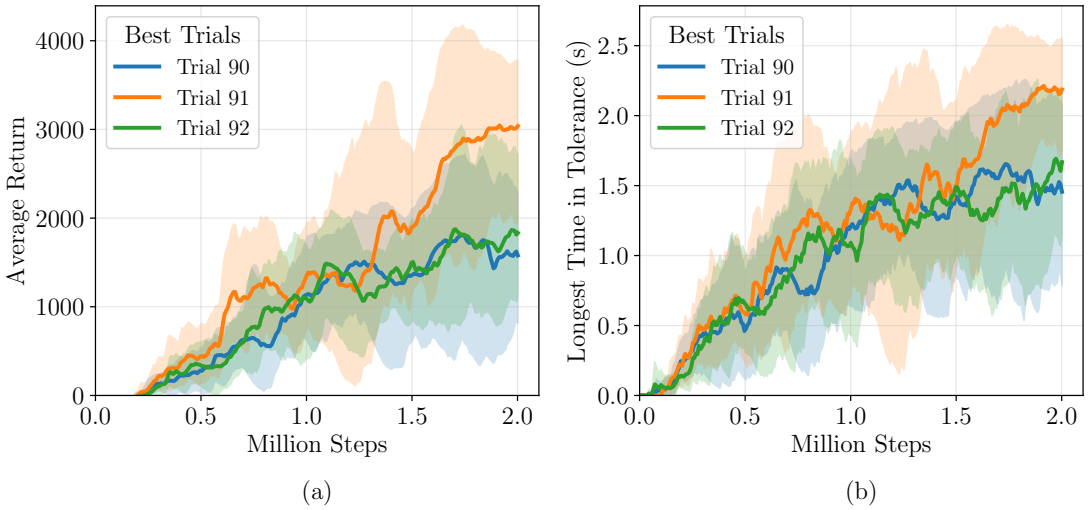


Fig. 5. The training curves for the three top performing optimisation trials averaged over three different random seed values: (a) the average return, and (b) the rendezvous window duration. The solid lines indicate average performance over three retraining runs, each with a different random seed value, and the shaded regions indicate the performance variation across these retrained runs. Results are plotted after smoothing via exponential weighted moving average to clarify trends.

trajectory. Thereafter, all constraints remained satisfied except for the minimum tension limit, which was violated for a substantial portion of the trajectory. Specifically, the tension dropped to zero during the latter half of the motion, resulting in slack tether conditions. This behaviour is undesirable, as it violates the assumptions of the rigid-body model and may lead to uncontrolled tether dynamics in a real system.

Two key conclusions can be drawn from these results. First, the current AL-iLQR controller is insufficient for identifying truly optimal trajectories under active constraint conditions. A more robust optimisation method

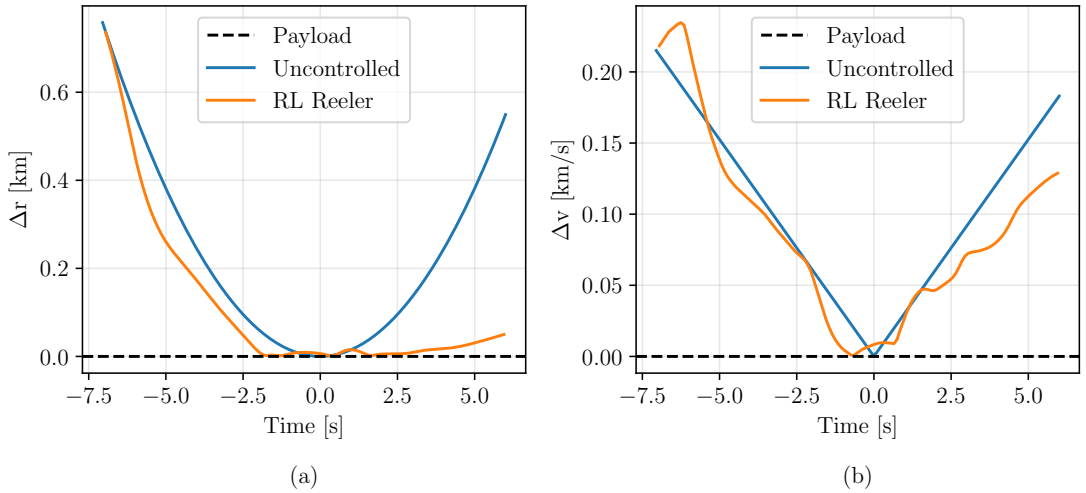


Fig. 6. Comparison of the tip trajectories of the uncontrolled tether against the RL controlled reeler configuration. Plots (a) and (b) show the magnitudes of the relative position and velocity difference vectors between the tether tip and payload respectively.

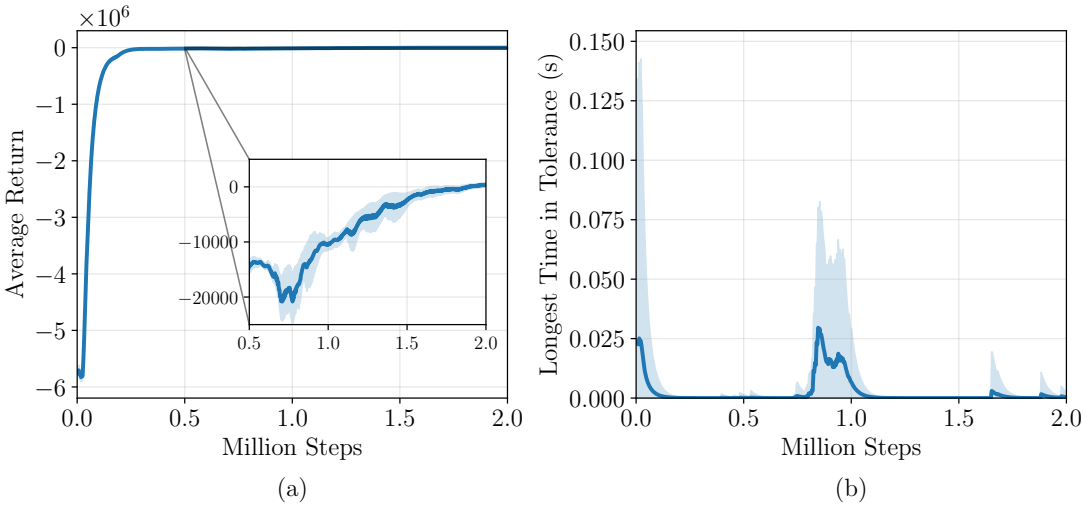


Fig. 7. The training curves for constrained RL environment averaged over three different random seed values: (a) the average return, and (b) the rendezvous window duration. The solid lines indicate average performance over three retraining runs, each with a different random seed value, and the shaded regions indicate the performance variation across these retrained runs. Results are plotted after smoothing via exponential weighted moving average to clarify trends. The inset plot in (a) shows the average return from 0.5 to 2 million time steps.

is required to achieve constraint-limited performance with meaningful constraint activation. Second, the model-free SAC RL algorithm, combined with simple penalty-based constraint incorporation in the reward function, does not offer a reliable solution to the constrained trajectory tracking problem. Constraint-aware RL approaches such as dedicated algorithms [17] or augmented objective methods [18] may be better suited to this task.

6. Conclusions

This research investigated the control of a rotating MXER tether system to extend the payload rendezvous

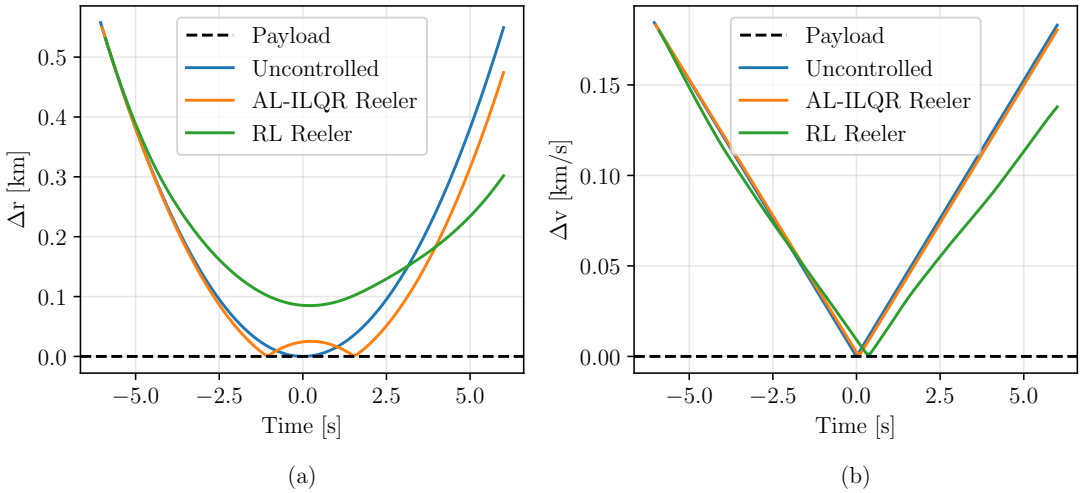


Fig. 8. Comparison of the tip trajectories of the uncontrolled tether against the constrained AL-iLQR and constrained RL control of the reeler configuration. Plots (a) and (b) show the magnitudes of the relative position and velocity difference vectors between the tether tip and payload respectively.

window, a critical challenge for the operational viability of such systems. A comparative analysis was conducted on three distinct actuator configurations, namely a baseline tip-reeling system, a climbing actuator mass, and a reeling actuator mass, using a 2D rigid-body model based on the Cislunar Tether Transport System. The performance of these configurations was evaluated under both a conventional optimal control method, iLQR, and a modern RL algorithm, SAC.

The primary contribution of this work is the clear identification of the reeler configuration as the most effective for extending the rendezvous window in an unconstrained environment. It extended the capture window to 1.8 seconds, a significant improvement over the 1.0-second window of the climber, the 0.8-second window of the baseline, and the 0.6-second window of the uncontrolled system. This superior performance is attributed to its enhanced authority over the tether tip's velocity profile. In this unconstrained case, both the fine-tuned iLQR controller and the SAC RL agent achieved this 1.8-second extension. However, the SAC agent produced noisy training results and a less smooth control policy, characterised by sporadic actuator commands. Such behaviour is undesirable in practice, as it can induce high structural loads in the tether, component wear in the actuators, and potentially excite unmodelled wave-like dynamics, thereby compromising the controllability of the tether system. One technique that can improve control smoothness is the penalisation of large differences of successive actions with an additional reward term. This can be done either by comparing actions to immediate or a rolling average of preceding actions. Both options add complexity to the reward which may require additional tuning to achieve desired performance.

The investigation into constrained control revealed significant challenges. When realistic operational limits on tether tension, g-loads, and actuator use were imposed, neither the AL-iLQR nor the SAC controller succeeded in extending the rendezvous window. The AL-iLQR method, while successfully satisfying all constraints, proved to be overly conservative and did not exploit the full dynamic capabilities of the system. Conversely, the SAC agent, guided by a simple penalty-based reward function, failed to robustly enforce the constraints, notably violating the minimum tension requirement, which would lead to a slack tether and a potential loss of control.

Based on these findings, several avenues for future work are recommended. The AL-iLQR framework could be enhanced with a more robust optimisation method, such as direct transcription, to better navigate the constrained trajectory space and operate closer to the system's physical limits. For RL, future efforts should move beyond simple penalty functions and explore dedicated constraint-aware algorithms or state-augmented objective methods to ensure safe and reliable operation within constraints. Finally, the findings of this study could be validated with a higher-fidelity, three-dimensional or elastic (or both) tether model to account for out-of-plane dynamics and longitudinal and transverse wave propagation, which is cru-

cial for the development of a flight-ready control system.

References

[1] R. P. Hoyt, "Cislunar Tether Transport System," NIAC-07600-011, May 30, 1999. [Online]. Available: https://www.niac.usra.edu/files/studies/final_report/7Hoyt.pdf.

[2] R. P. Hoyt, "The Cislunar Tether Transport System Architecture," Jul. 20, 2000.

[3] Z. du Toit, "Controlling mxer tether dynamics for extended payload rendezvous," M.S. thesis, Delft University of Technology, 2025. [Online]. Available: https://repository.tudelft.nl.

[4] B. E. Jackson, *AL-iLQR Tutorial*, Jun. 12, 2019. [Online]. Available: https://bjack205.github.io/projects/2019/06/12/ilqr_tutorial.html.

[5] T. A. Howell, B. E. Jackson, and Z. Manchester, "ALTRO: A Fast Solver for Constrained Trajectory Optimization," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov. 2019, pp. 7674–7679. DOI: 10.1109/IROS40897.2019.8967788.

[6] R. Garnett, *Bayesian Optimization*. Cambridge, United Kingdom: Cambridge University Press, 2023, ISBN: 978-1-108-42578-0.

[7] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[8] D. J. Wales and J. P. Doye, "Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms," *The Journal of Physical Chemistry A*, vol. 101, no. 28, pp. 5111–5116, 1997.

[9] K. Sorensen, "Conceptual design and analysis of an MXER tether boost station," in *37th Joint Propulsion Conference and Exhibit*, Jul. 8, 2001. DOI: 10.2514/6.2001-3915.

[10] P. Williams, "A Review of Space Tether Technology," *Recent Patents on Space Technology*, vol. 2, no. 1, pp. 22–36, 2012. DOI: 10.2174/1877611611202010022.

[11] S. L. Canfield, "Developing Capture Mechanisms and High-Fidelity Dynamic Models for the MXER Tether System," NASA/CR-2007- 215076, Sep. 1, 2007. [Online]. Available: https://ntrs.nasa.gov/citations/20080002099.

[12] R. Jansen *et al.*, "High efficiency megawatt motor conceptual design," in *2018 Joint Propulsion Conference*, 2018, p. 4699.

[13] R. Hoyt, "Tether rendezvous methods," *Interim Report on NIAC Phase II Contract*, pp. 07 600–034, 2000.

[14] M. Towers *et al.*, "Gymnasium: A standard interface for reinforcement learning environments," *arXiv preprint arXiv:2407.17032*, 2024.

[15] A. Y. Ng, D. Harada, and S. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *Icml*, Citeseer, vol. 99, 1999, pp. 278–287.

[16] A. Raffin, *Rl baselines3 zoo*, https://github.com/DLR-RM/rl-baselines3-zoo, 2020.

[17] W. Zhao, R. Chen, Y. Sun, T. Wei, and C. Liu, *State-wise constrained policy optimization*, 2024. arXiv: 2306.12594 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2306.12594.

[18] M. Calvo-Fullana, S. Paternain, L. F. Chamon, and A. Ribeiro, "State augmented constrained reinforcement learning: Overcoming the limitations of learning with rewards," *IEEE Transactions on Automatic Control*, vol. 69, no. 7, pp. 4275–4290, 2023.

6

Verification, Validation and Robustness

6.1. Verification and Validation

This chapter details the verification, validation, and robustness analysis performed to ensure the fidelity of the derived rigid-body model and the integrity of its outputs. The process begins with the verification of the numerical methods, followed by the validation of the physical model against established benchmarks, and concludes with an analysis of the system's sensitivity to parameter uncertainty and the tuning of the control algorithms.

First, a comparative study of several numerical integration methods is conducted to select an optimal solver that balances computational efficiency with numerical accuracy, a crucial consideration for the extensive simulations required for controller training and tuning. Second, the rigid-body tether model itself is validated by comparing its dynamic behaviour and tension profiles against a well-established dumbbell model and reference data from existing literature. Finally, a variance-based sensitivity analysis is performed using a Monte-Carlo simulation to quantify the impact of uncertainties in the tether's physical parameters on the rendezvous dynamics. This analysis informs the required capture tolerances and highlights the most critical parameters for accurate modelling. The chapter also details the hyperparameter tuning process for the reinforcement learning controller, which is vital for optimising its performance.

6.1.1. Integrator Comparison and Benchmarking

An integrator comparison study was performed to balance the accurate propagation of the rigid-body model's dynamics with the computational effort required for this propagation. The model dynamics presented in Chapter 5 (the IAC paper) are governed by well behaved, ordinary differential equations. The decision of opting for a rigid-body approach rather than a longitudinally extensible or transversely flexible tether avoids the challenge of stiff differential equations, and allows the use of explicit time integration methods. For this project, four such methods were identified as candidates, two being

fixed time step, and two being variable time step methods

- Simple forward Euler method (fixed step)
- Classical fourth-order Runge–Kutta method (RK4) (fixed step)
- Second-order Runge–Kutta method with third-order interpolation (RK23) (variable step)
- Fourth-order Runge–Kutta method with fifth-order interpolation (RK45) (variable step)

Baseline solution

To evaluate these integration methods, a baseline is needed for comparison. The rigid-body model's non-linear dynamics does not have a closed form solution that can be used as ground truth, and thus the use a of higher-order, increased accuracy integration method can be used to create reference solution to compare against. The eighth-order Dormand–Prince integrator (DOP853) with tight error tolerances (rtol $= 10^{-14}$ and atol $= 10^{-15}$) was chosen to generate the reference trajectory on a uniform grid of 100 points over the interval [0, 60] seconds. All other methods are then compared against this reference.

Control scenario and parameter sweeps

To test the reeling dynamics of the system, a small, fixed reeling acceleration of 0.1 m/s^2 was applied to each point mass. This positive acceleration is defined such that the length of tether spooled in each point mass increases over time, which forces non-trivial motion over the simulation time window.

Each fixed step integration method was run with a number of different time step sizes Δt, while the variable step methods were tested with a range of absolute and relative tolerances such that rtol = atol = tol.

$$\Delta t \in \{5.0,\ 2.0,\ 1.0,\ 0.5,\ 0.2,\ 0.1,\ 0.05,\ 0.02,\ 0.01,\ 0.005,\ 0.002,\ 0.001\}\ \text{s}$$

$$\text{tol} \in \{10^{-3},\ 10^{-4},\ \ldots,\ 10^{N-1},\ 10^{N}\}, \quad \text{with} \quad N = -14$$

As with the reference DOP853 solution, each integrator was run over the time window [0, 60] s, and the solutions were interpolated onto the common DOP853 grid. The number of function evaluations for each candidate integration method was logged over all test cases. To determine the comparative accuracies of the methods, the vector difference was taken between the reference solution and each integrator test case. The maximum of the norm of these position and velocity difference vectors was then use as the absolute error. The relative error was also calculated for completeness. These errors, number of function evolutions and swept parameters are combined in the plots of Figure 6.1. The number of function evaluations (shown in Figure 6.1b) serve as an indicator for the computational effort required by each method for a target accuracy. The error behaviour of all integration methods was tested for both position and velocity states. For brevity, only the positional error plots are presented in Figure 6.1, as the overall trends for both cases were nearly identical, with the positional case exhibiting slightly larger errors, making it the more limiting scenario.

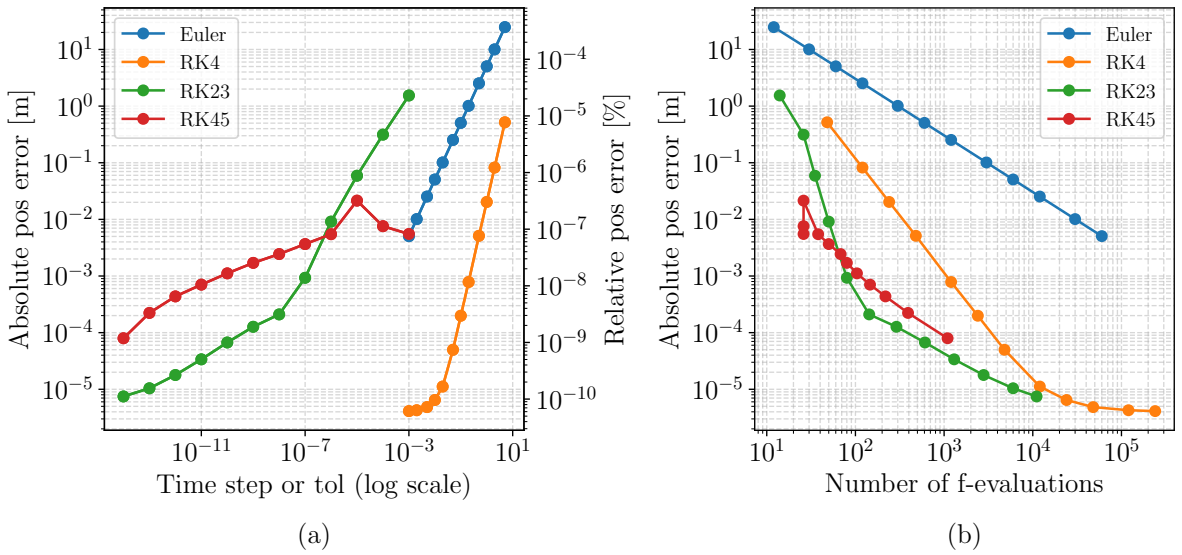


Figure 6.1: Integration method performance comparison for positional error compared to the DOP853 reference solution. The first figure (a) shows both absolute as well as relative errors against the time step size for fixed step method or tolerance for variable step methods, while (b) shows the absolute positional error against the number of evaluations of the tether's dynamics function.

Integration method selection

The selection integrator criteria are based on two simple requirements. The first is an absolute accuracy in the 10^{-3} to 10^{-2} m range. For this project, extreme accuracy is not required as the simplifying assumptions made for the rigid-body model defined in Chapter 5 neglects perturbation effects on the tether system and its orbit. The required accuracy of the model is at most on the metre to sub metre-level, and thus an integrator accuracy of two to three orders of magnitude below this level is acceptable. The second, and arguably more important requirement, is a low number of function evaluations for the tether system's dynamics. The number of function evaluations directly translate to computational effort, which needs to be kept to a minimum. This is especially true for the case of training an RL agent for tether system control, as the training process requires more than hundreds of thousands of dynamic function evaluations. Selecting an integrator that performs within the required accuracy at the lowest number of function evaluations is therefore paramount to prevent the training of RL agent(s) from taking up significant amounts of time.

From the visualisations in Figure 6.1, there are some integrators that clearly perform better than others. Figure 6.1b clearly shows the forward Euler method as the least accurate of the lot, with higher errors than all other methods for the whole range of function evaluations. The RK4 method performs slightly better, but still suffers from reduced accuracy at lower numbers of function evaluations. Thus both fixed-step methods can be ruled out. The variable step size methods perform much better, showing the desired levels of accuracy at around 10 to 100 function evaluations. For higher numbers of function evaluations, the RK23 method performs better than its RK45 counterpart. However, in the desired position accuracy range of 10^{-3} to 10^{-2} m, the RK45 method takes the cake for lowest number of function evaluations. The corresponding absolute and relative tolerance levels fall in the range of 10^{-6} to 10^{-3}. The default tolerance

values for the Scipy library's ODE solver, `solve_ivp`, falls nicely in this range with its rtol $= 10^{-3}$ and atol $= 10^{-6}$. Based on the selection criteria outlined above the RK45 method was selected, and used with the default `solve_ivp` tolerances.

6.1.2. Rigid-body Tether Model Validation

To validate the derived rigid-body model, its response was compared against a dumbbell tether model [17]. Although the dumbbell model is conceptually simpler than the rigid-body model, it (and variations of it) has been shown to be useful as a baseline and comparison tool for more complex models [108] [109].

As mentioned in Section 2.1.2, the dumbbell model makes the simplifying assumption of grouping the tether's mass into two masses, one at each of the endpoints. Based on the mass concentration at the endpoints, this model necessarily assumes a massless tether. To partially account for influence of the tether's mass on the system's dynamics, the tether mass can be divided up and added to the endpoint masses. For a constant cross section tether, this grouping half of of the tether mass with each end point keeps the system's centre of mass at its true position. This is crucial for comparison with the rigid-body model, as both models assume the system's COM follows an unperturbed, elliptic, Keplerian orbit. Further, the dumbbell model considered here assumes a fixed tether length with no motion of the point masses relative to the tether. This assumption is common, but not always necessary, as dumbbell models with spring-like tethers have been used in literature [110] to model longitudinal oscillations. The fixed tether length assumption made here means the dumbbell model has a constant moment of inertia.

Dumbbell model kinematics and dynamics

Similar to the rigid-body model defined in Chapter 5, the dumbbell's kinematics define the position and velocity vectors of a point mass on the tether in the global OXY frame at any point in time according to Eq. 6.1 and 6.2.

$$\boldsymbol{R}_i = \boldsymbol{R}_{cm} + \boldsymbol{r}_i \tag{6.1}$$

$$\ddot{\boldsymbol{R}}_i = \dot{\boldsymbol{R}}_{cm} + \boldsymbol{\omega} \times \boldsymbol{r}_i \tag{6.2}$$

As with the rigid-body model, the scalar r_i is the discarnate from the COM to the point mass, while $\boldsymbol{r}_i = r_i \hat{\boldsymbol{e}}_t$ is the point mass' position vector in the global frame relative to the COM. This vector always points along the tether's length to the orientation unit vector $\hat{\boldsymbol{e}}_t$ Note that since there is no relative motion between the point masses and the tether, there is no $\dot{\boldsymbol{r}}_i$ term in Eq. 6.2, unlike the kinematic velocity relation for the rigid-body model. The time-varying ω value is determined by integration of the rotational equation of motion

$$\tau_g(t) = I_{CM}\, \dot{\omega} \tag{6.3}$$

where τ_g is the gravitational torque due to the varying gravitational forces on the discrete point masses, and I_{CM} is the constant moment of inertia.

Through integration of Eq. 6.3 and by propagating the orbit of the COM, the dumbbell tether's motion is fully defined, and can be used as a validation tool for the rigid-body model derived in Chapter 5.

To compare the two models, both were set up with the same Keplerian orbit, the same overall mass, and the same COM offset as measured from the tether tip. To match he dumbbell model's mass and COM offset to that of the rigid-body model, each half of the tether's mass was was added to the discrete point masses at either ends of the tether. The dynamics of the respective models were then integrated over a total period of 1 minute (from -30, to 30 seconds) under no control input. As with the rigid-body model in Chapter 5, the simulation window was centred at perigee such that the separation between the tether tip (of both models) and the idea payload reaches zero momentarily at $t = 0$ s. To enforce this symmetry, the effective simulation duration is only 30 seconds, as the tethers were simulation outward from $t = 0$ s in both the positive and negative time directions. The 30 second integration period (in each direction) is longer than that considered during the controlled approach for the actual rigid-body model in Chapter 5 to examine the agreement between the models both inside and and outside the regular controlled window of 10 to 12 seconds. It should be noted that the actual simulation was not performed in two outward segments as illustrated above. The image of symmetric integration was merely given to describe the effective behaviour of the tether models centred around the perigee crossing point. The tether models were integrated backward in time from $t = 0$ to find their respective initial conditions, and then integrated forward for the whole 60 second duration.

First, the attention is turned to the rotation states over time of the two tether models. The tethers' orientation angle α, defined relative to the x axis of the global OXY frame described in Chapter 5, is given in Figure 6.2a, and the tether rotation rates in 6.2b. From these results it is seen that the tether orientation for both models remains effectively indistinguishable over the considered time window. The difference in rotation rate is more pronounced, mainly due to the small range of values considered on the vertical axis of Figure 6.2b. The dumbbell model is seen to rotate slower further away from the perigee crossing point (where both tether models are perfectly aligned with their orbital radial vector by design), but increases its rotation rate faster than the rigid-body model. This comes down to the mass distribution differences between the models. As previously mentioned, the tether mass was added to and split between the two end masses of the dumbbell model to maintain the same overall tether system mass and COM offset as the rigid-body model. Hence, the dumbbell end masses are heavier than those of the rigid-body, resulting in a higher moment of inertia, and generally slower rotation. The larger masses also have the effect of increased gravitational attraction, and thus increased gravitational torque, which results in the dumbbell model undergoing more pronounced rotational acceleration than the rigid-body model. Both of these effects, slower general rotation and increased angular acceleration rates, are visible in Figure 6.2b.

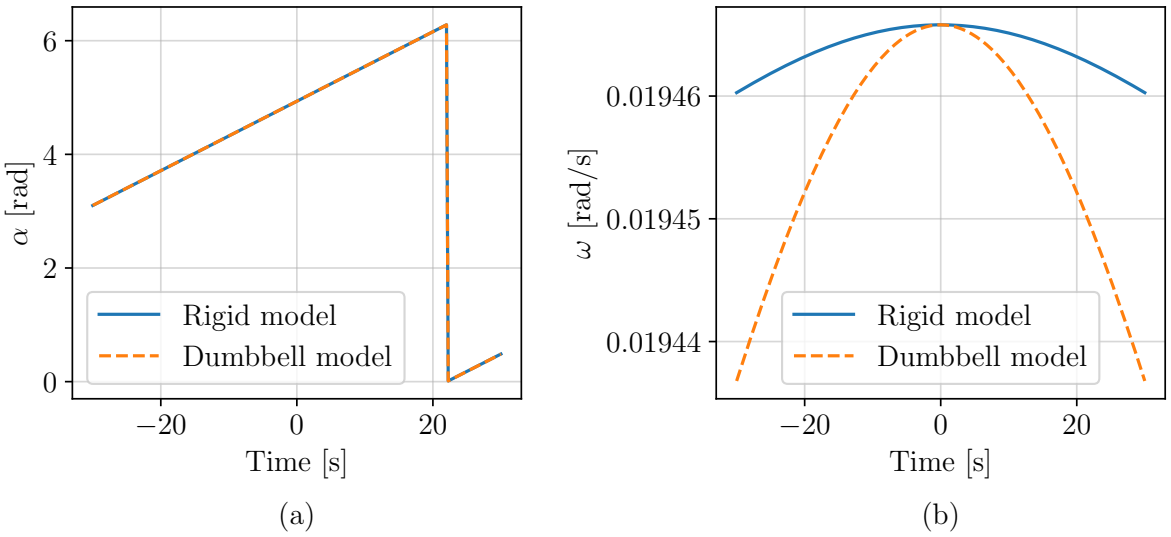


Figure 6.2: Comparison of the tether's (a) orientation angle and (b) rotation rate for the rigid-body and dumbbell models. Simulated over a 60 second window centred at perigee.

Though the differences in rotation state between the two models are clearly present in small scale, their effects are minor on the actual position and velocity states of the tether tip. The position and state component differences are shown in Figure 6.3. It is important to note that the differences presented here are determined using the rigid-body model as the reference, as it is expected to be the more accurate of the two models. As a clarifying example, the difference in x position seen in Figure 6.3a is calculated as $\Delta X = X_{dumbbell} - X_{rigid-body}$. The same applies for all other state components in Figure 6.3a through 6.3d. For Figure 6.3e and 6.3f the vector difference between the position and velocity states were calculated first before taking the norm, to obtain the magnitudes of the actual position and velocity separation between the models.

For good agreement between the dumbbell and rigid-body models, these differences should be minor. It is perhaps easiest to start with the the magnitude difference curves of Figure 6.3e and 6.3f to assess the agreement. The position magnitude difference curve in Figure 6.3e shows a clear diverging trend in both the positive and negative time directions. This is to be expected, as the rotation rate of the dumbbell model slows down almost quadratically compared to the rigid-body model as previously seen in Figure 6.2b. This slower rotation rate means the endpoints of the two tethers will diverge over time as the rigid-body model rotates its tip away faster than the dumbbell model can catch up. However, this effect is minor over these time scale considered, resulting in a maximum positional separation of about 20 metres over 30 seconds. As previously mentioned, this $[-30, 30]$ second window is longer than the actual controlled simulation window of 10 to 12 seconds. When considering this actual window, the difference in position falls to below 0.3 metres. An even smaller separation in model prediction is noted from the velocity magnitude graph in Figure 6.3f where the difference drops to single digit centimetre values. These positional and velocity difference values are sufficiently small that the overall tether motion inside these time windows can be considered effectively identical for both models at the levels of relative accuracy required in the control problem.

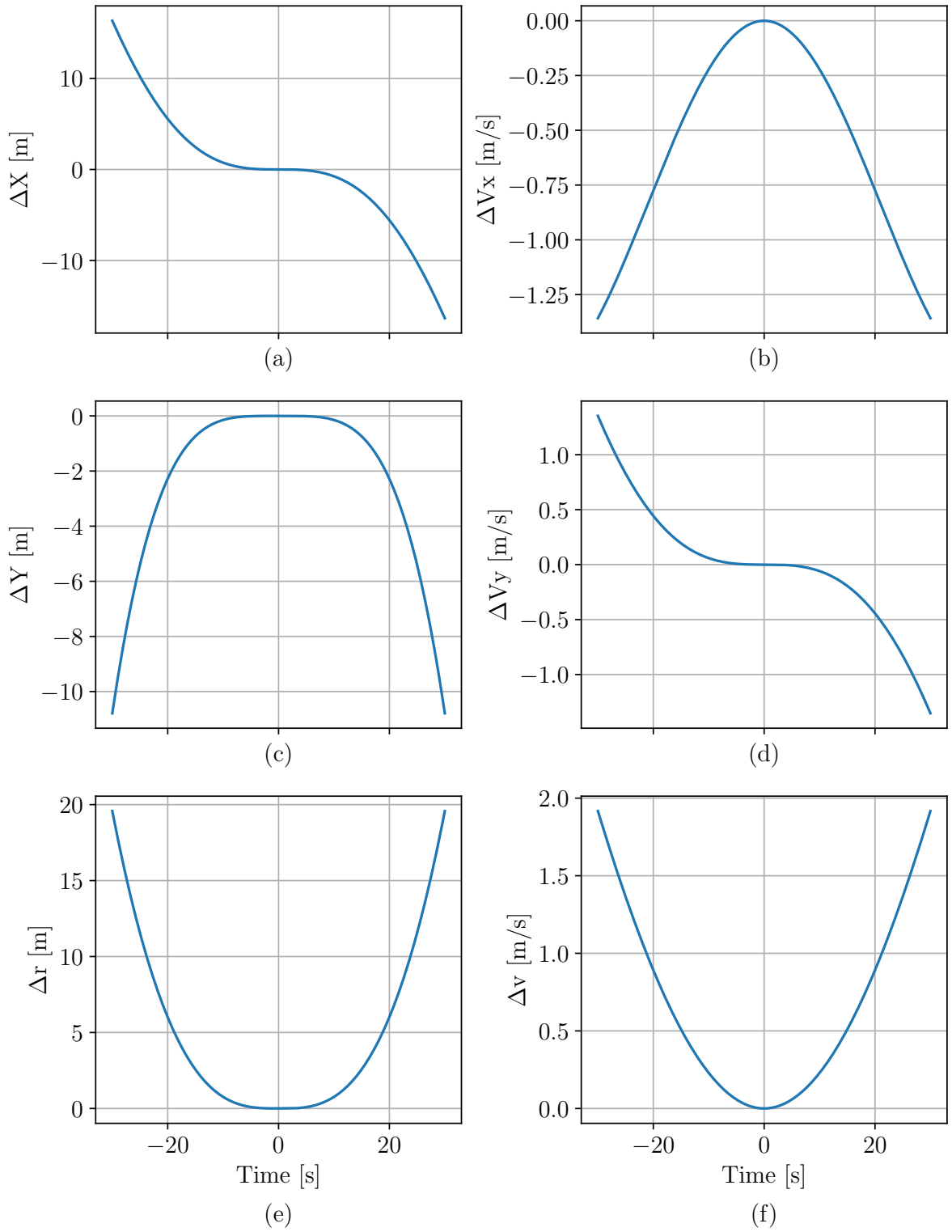


Figure 6.3: The difference between the dumbbell and rigid-body tether model states over two 30 second windows centred at perigee. The graphs in the left column are the differences in (a) x position, (c) y position and (e) position magnitude. The right column contains the differences in (b) x velocity, (d) y velocity, and (f) velocity magnitude. The difference in each graph is calculated as dumbbell state minus rigid-body state.

It is interesting to note the different curve shapes for the state component differences in Figure 6.3a through d. These curves tell the same story as the differing rotation rate values seen in Figure 6.2, and are in fact, a direct result of this rotation rate difference. The s-shaped curves for the x position and y velocity differences both start out positive, crosses the perigee point at $t = 0$ seconds with zero slope, and ends up negative. These curves are descriptive of lead-lag behaviour. Since both these components start out positive, it indicates that the dumbbell tether model starts out at a higher horizontal position, and vertical velocity separation from the payload compared to the rigid-body model. Thus the tip motion of the rigid-body tether leads for the first half of the simulation time (-30 to 0 seconds), with the dumbbell model's tip lagging behind. For the second half of the simulation, the roles reverse and the rigid-body tether tip now lags that of the dumbbell model. This means the rigid-body model remains slightly closer to the payload over the whole simulation period, speeding up at first, and slowing down later on. These effects tend toward being negligible for smaller simulation windows, but is again a confirmation of the expected rotation behavioural differences due to the underlying assumption of a massive and massless tethers which lead to different rotational moments of inertia.

From the results presented in Figure 6.2 and 6.3, along with the discussions in this section, it can be concluded that the rigid-body model derived in Chapter 5 corresponds excellently to the dumbbell model over smaller time windows. Since the dumbbell model has been used extensively in literature for first approaches to modelling and initial control design, this newly derived model can then also be used for such tasks as is done in Chapter 5.

Tether tension

For a dumbbell model with a massless tether, the only tension contributions come from the point masses m_i at the tether's endpoints and the forces acting on these point masses. Thus the tension is constant along the tether, but not in time. This can also be shown from the equation of motion for the tether if the tether's mass (or linear density) tends to zero. The tether's equation of motion is given in Chapter 5, and is repeated in Eq. 6.4 for convenient reference.

$$\rho \ddot{\boldsymbol{R}} = \boldsymbol{T}' + \rho \boldsymbol{g} \tag{6.4}$$

It follows from Eq. 6.4 that for $\rho \to 0$, the spatial tension derivative also tends to zero $\boldsymbol{T}' \to 0$, meaning a constant tension along the tether.

The same logic leads to the opposite conclusion for the rigid-body model with a massive tether. Here spatial tension derivative is non-zero and potentially non-linear. Thus the dumbbell model cannot be used as an accurate validation method for the rigid-body model's tension distribution, and an alternative is needed. Fortunately, examples of tether tension profiles can be found in literature, and can thus be used as a means of comparison.

An example tension distribution is given in [18] for a typical MXER tether with 90 km length, tip mass of 250 kg, counterweight mass of 11000 kg, and 8 embedded masses of 200 kg each, placed at 10 km intervals. The tether rotates at 0.8 degrees per second in

free space (no gravitational or external forces). The tether mass is unfortunately not given, but it is mentioned that the tether material is Zylon, and that each of the uniform 10 km tether segments are sized such that they all undergo a strain of $\delta = 0.01$. To get a linear density estimate for the example tether, the cross sectional area for each 10 km tether section can be calculated from the simple stress-strain relation in Eq. 6.5

$$\frac{T_i}{EA_i} = \delta \tag{6.5}$$

where T_i is the tension in the i^{th} tether segment, E is the elastic modulus of the material, and A_i is the segment's uniform cross-sectional area. From here the tether's average density per unit length can determined with

$$\rho_{\text{avg}} = \frac{1}{L}\sum_{i=1}^{10} \rho_{\text{vol}} A_i l_i \tag{6.6}$$

where ρ_{vol} is the volumetric density of the tether material (in kg/m^3) and $l_i = 10km$ is the segment length. This density per length is necessary for the rigid-body model, which assumes a constant tether cross-sectional profile over the whole tether length. Thus two material properties of Zylon, also known as poly(p-phenylene-2, 6-benzobisoxazole), are needed, E and ρ^*. As the example tether source [18] did not specify which Zylon variant was used for its tension calculations, two variants were identified from literature [111] with the material properties of interest listed in Table 6.1.

Table 6.1: Material properties for two variants of Zylon [111]

Zylon Variant	**Elastic modulus [GPa]**	**Density [kg/m^3]**
AS (Regular)	180	1540
HM (High Modulus)	270	1560

The average tether linear density was calculated using both sets of material properties, and were found to be $\rho_{avg}^{AS} = 28.6$ kg/km, and $\rho_{avg}^{HM} = 42.3$ kg/km for the regular and high modulus variants respectively. These values were used as upper and lower bounds for the tension distribution calculation via the rigid-body model. Figure 6.4 shows the reference tension data points [18] on the vertical axis, against the point mass positions along the tether. The shaded region is the tension distribution calculated using the rigid-body tether model with the linear density values for AS and HM Zylon making up the upper and lower bounds respectively. The solid line indicates the rigid-body model tension distribution as calculated with a linear density averaged between that of the AS and HM Zylon materials, namely $\rho_{avg} \approx 35.5$ kg/km.

Figure 6.4 shows good agreement between the rigid-body model's tension profile calculated using the averaged linear density (solid line), and the reference data points (red markers with dashed lines). Both tension profiles show the expected tension jumps across each discrete point mass, due to the sudden increase of mass along the tether. This mass increase requires higher tension to maintain the rotational motion of the tether. Both tension profiles also show good agreement and a clear trend of increasing tension from

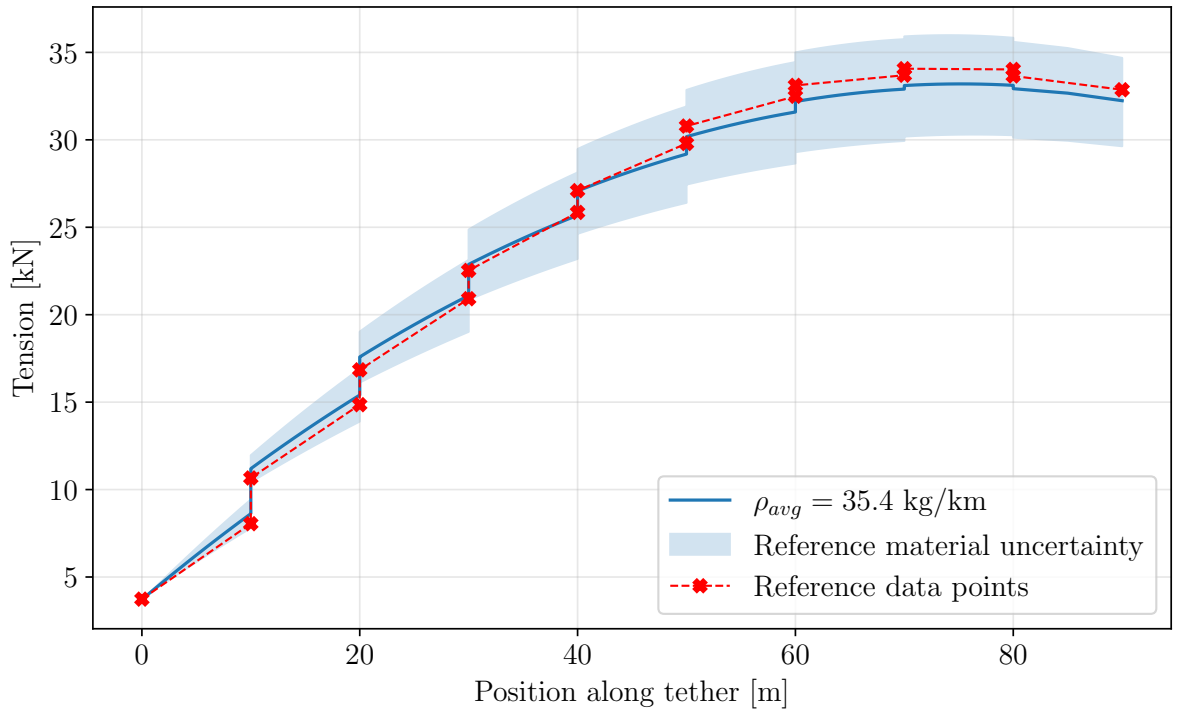


Figure 6.4: Comparison of the rigid-body model's tension distribution against reference data points found in literature. Reference data digitised from [18]. Rigid-body model tension distribution calculated with averaged tether linear density based on data in Table 6.1

the tether tip up to the COM, followed by a slight decrease moving away from the COM toward the counterweight. This maximum tension at the COM is expected, as this is where the tension acting from either end of the tether is balanced.

The disagreement between the two tension profiles, though slight, is noticeable, and can be attributed to three main reasons. The first, and likely most impactful, being the difference in model assumptions regarding the tether segment cross-sectional areas. The rigid-body model assumes a uniform cross-sectional area along the whole length of the tether, whereas the reference data is calculated for reducing cross sectional areas for tether segments further away from the COM. This becomes clear when comparing the tension values to the left of the Figure 6.4, where the reference data has a reduced cross-sectional area (and thus reduced linear density) compared to the averaged value used for the rigid-body tension calculation, and to the right of the figure where the opposite occurs. Where the rigid-body model assumes a higher linear density (and thus higher segement mass), the tension it predicts is higher than that of the reference data. The opposite is also true. Where it assumes a lower linear density than the reference, the tension it predicts is lower due to a lower segment mass. The second reason for the discrepancy, though likely a reduced contribution, is that the exact material properties for the reference tether remain unknown, and the the calculated tension profile is based on a best estimate from other literature sources. Thirdly, and least impactful, is that the reference data points were digitised and extracted from a scanned plot presented in [18]. This digitisation process was performed three times to get data with improved accuracy and quantifiable uncertainty. The uncertainty of the digitised results were

minimal, contributing less than the span of a single red marker in Figure 6.4, and was thus excluded to maintain clarity.

Based on these results, it is concluded that the rigid-body model produces sufficiently accurate tension results for the purposes used in Chapter 5.

6.1.3. AL-ILQR Controller Verification

To verify the correct implementation of the AL-iLQR controller described in Chapter 5, a test case was used. This test case consisted of controlling a simple pendulum to track a time-varying target angle subject to a maximum absolute angle constraint. Both the target angle trajectory and the constraint limit value were selected such that they fall outside the typical small-angle assumption of 10–15° to test the controller under non-linear dynamics.

The test case involved commanding the pendulum to follow a sinusoidal reference trajectory with an amplitude of 45° over a 10-second period, while being subject to an angle constraint of ±35°. The pendulum's initial angle was offset slightly from the target trajectory. The results of this constrained trajectory tracking problem are shown in Figure 6.5.

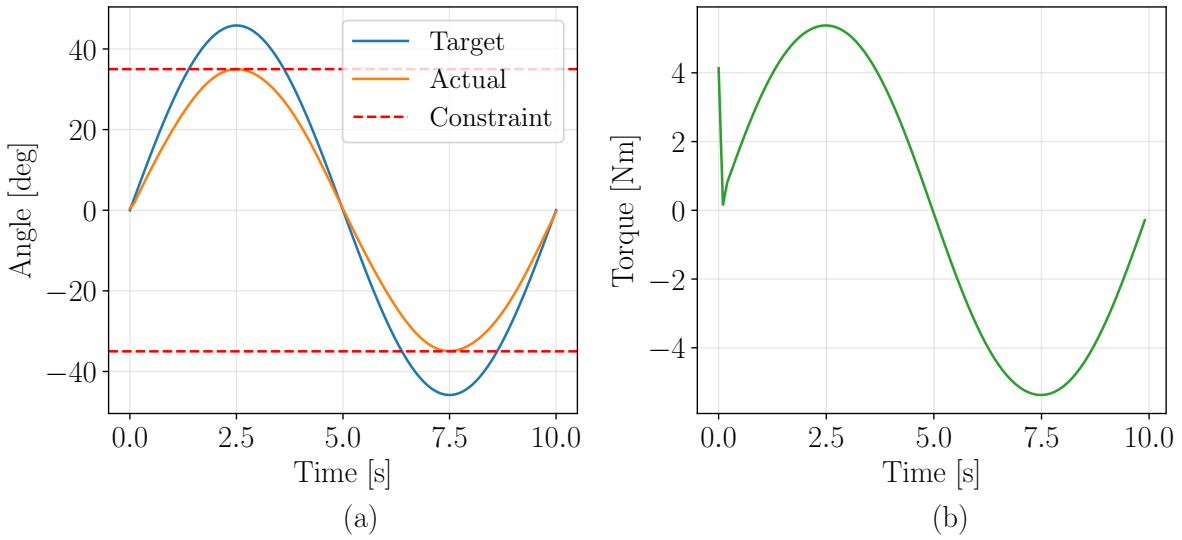


Figure 6.5: Trajectory tracking control of a simple pendulum using the AL-iLQR controller, with (a) the target and actual pendulum angles, and (b) the applied control torque.

Figure 6.5(a) shows the controller successfully tracks the target trajectory when the reference angle is within the feasible region. When the target trajectory exceeds the ±35° limit, the controller successfully respects the imposed constraint, causing the pendulum's actual trajectory to touch the limit value and continuing the tracking once the reference returns to the feasible region. It is interesting to note that the actual trajectory does not saturate the angle value against the constraint limit when the target trajectory exceeds this limit. This behaviour can likely be attributed to the underlying incremental linearisation of the dynamics within the AL-iLQR controller which seems to discourage sudden

and discontinuous trajectory changes. The smooth trajectory behaviour with momentary rather than continued constraint activation was observed repeatedly in different tests despite significant increases in the state error coefficient matrix $\boldsymbol{Q}$ and decreases in the control coefficient matrix $\boldsymbol{R}$ (both defined in Chapter 5). This serves as an indication that the AL-iLQR controller, as implemented in this work, prioritises general constraint satisfaction rather than a truly optimal trajectory with maximally activated constraints.

The corresponding control torque, shown in Figure 6.5(b), reflects this behaviour. The torque profile is smooth for the majority of the trajectory, with a brief valley at the start attributable to the slight offset in starting angle between the pendulum and the target trajectory.

The successful enforcement of these state constraints, and the logical control response required to do so, serve as a verification that the implemented AL-ILQR algorithm is capable of finding a trajectory that satisfies the imposed constraints, though not necessarily a truly optimal one.

6.1.4. Reinforcement Learning Controller Verification

Similar to the AL-iLQR controller, the chosen Stable Baselines 3 implementation of the model-free SAC RL algorithm was tested on a simple and widely used benchmark environment to verify correct integration and usage. For this purpose, the Gymnasium Pendulum-V1 environment was selected. This environment represents the classic control problem of swinging up and stabilising a pendulum in the upright position at zero degrees.

The RL agent was trained using the default settings provided by the Stable Baselines 3 SAC implementation. No additional constraints were imposed, as the test case relied solely on the default Gymnasium environment setup.

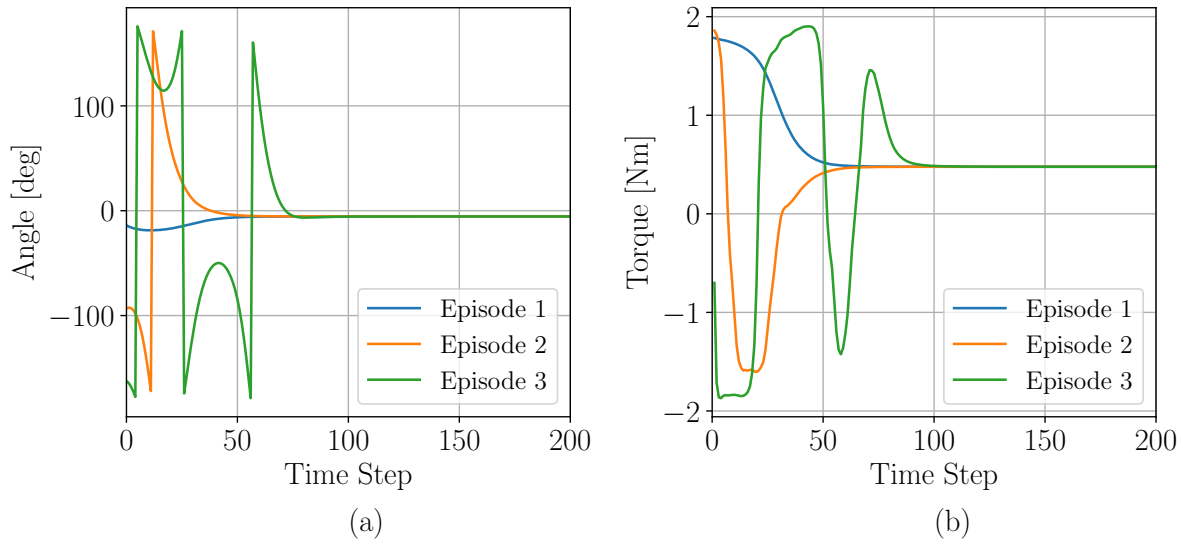


Figure 6.6: Regulating control of a simple pendulum using the RL SAC algorithm as a controller for three different initial conditions. Sub-figure (a) shows the pendulum angles, and (b) the applied control torque plotted against the episode time steps.

Figure 6.6(a) shows that the RL agent successfully regulated the pendulum to a near-zero angle across three episodes, each initialised with a different starting condition. The slight deviation from the 0° target can be attributed, at least in part, to the Pendulum-V1 environment's reward function, which penalised non-zero state and control values quadratically [107]. This formulation encourages minimal deviation but does not strictly enforce convergence to exactly zero.

Figure 6.6(b) presents the corresponding control torque profiles. In all three episodes, the torque values eventually settled around 0.5 Nm. This offset from zero again reflects the influence of the reward function, which can be adjusted if more precise regulation is required.

Overall, the results of this test case demonstrated that the SAC algorithm was capable of solving a basic regulation task. The successful control behaviour confirmed that the user-side implementation of the RL agent was correctly configured and functioning as intended.

6.2. Sensitivity Analysis and Tuning

This section presents the sensitivity analysis and parameter tuning processes for the rigid-body tether model and the Reinforcement Learning (RL) agent.

First, a variance-based sensitivity analysis is conducted to formally assess how uncertainties in the tether system's physical parameters impact the predictive accuracy of the rigid-body model. This analysis employs a Monte-Carlo simulation framework and the Sobol method to quantify the influence of each parameter on the final tether tip state.

Following this, the section details the hyperparameter optimisation for the Soft Actor-Critic (SAC) RL agent. An automated tuning study is performed to optimise both

the reward function coefficients and the underlying algorithmic hyperparameters. The relative influence of each parameter on the agent's performance is then evaluated using functional ANOVA (fANOVA) to identify the most critical factors for effective control.

6.2.1. Rigid-body Tether Model Parameters

The rigid-body model defined in Chapter 5 requires a number of parameters to define the tether system. These inputs can be broadly classified as either orbital parameters for the definition of the payload and tether system orbits, or physical parameters that describe the tether system's configuration. Chief among these physical configuration parameters are the point masses and their positions along the tether, followed by the linear density of the tether. These parameters directly influence the dynamic behaviour of the tether system as it approaches the payload rendezvous point at the tether system's orbital perigee.

Physical parameter sensitivity analysis

For large tether systems such as the baseline tether mentioned in Section 4.3, and especially for even larger and more massive tether systems, using accurate values for the physical configuration parameters becomes important for accurate predictions of the tether tip states over time. To asses the effect that uncertainty in these parameters has on the tether tip state prediction of the rigid-body model, a variance-based sensitivity analysis was conducted. This analysis was based on a Monte-Carlo simulation of the baseline tether system configuration over a simulation period of 40 seconds such that $t = 0$ coincides exactly for the nominal tether configuration. Thus the initial rotational state (at $t = -20$ s) was kept constant as the rotational state of the uncontrolled baseline tether system at this time across all simulations.

The parameters varied for each simulation are identified in Table 6.2, along with their nominal baseline values, and a variation of these values. Each of the identified parameters are sampled from uncorrelated normal distributions defined by the the baseline value as the mean, and a standard deviation defined such that 3 standard deviations make up the parameter variation value. This approach effectively captures the full extent of the variation within the distribution bounds ($3\sigma = 99.7\%$ of possible parameter variation). A total of 2560 samples were generated using the Sobol sampling method. The remaining parameters are taken exactly as listed in Section 4.3.

Table 6.2: Physical parameters varied for the Monte-Carlo simulation, with their baseline (mean) values and the maximum variation for each parameter. Variation values taken from [110].

Physical parameter	Baseline value	Maximum variation [%]
Tether linear density ρ [kg/km]	82.74	0.01
Tip mass m_1 [kg]	650	0.01
Counterweight mass m_2 [kg]	15004	0.01
Total tether length L [km]	100	0.01

For each of the sample simulations, the smallest separation between the tether tip and

payload states were logged at the nominal rendezvous point of $t = 0$ s. One pair of positional and velocity difference vectors, $[\Delta X, \Delta Y]$ and $[\Delta V_X, \Delta V_Y]$ respectively, were thus collected for each simulation, with their components calculated as:

$$\Delta X = X_{tip} - X_{payload}^{t=0} \quad \text{and} \quad \Delta Y = Y_{tip} - Y_{payload}^{t=0}$$

$$\Delta V_X = V_{X,tip} - V_{X,payload}^{t=0} \quad \text{and} \quad \Delta V_Y = V_{Y,tip} - V_{Y,payload}^{t=0}$$

Here the tether tip and payload states are measured in the global OXY frame as defined in Chapter 5. The magnitudes of these difference vectors were calculated and processed with the SAlib python library's Sobol analysis method [112][113] to produce first and second order sensitivity indices. From here onward, these indices are shortened to S_1 and S_2 respectively. The S_1 and S_2 indices for the position and velocity difference magnitudes are listed in Table 6.3 and Table 6.4 respectively. Each table includes the corresponding confidence values denoted as $S_{1,conf}$ and $S_{2,conf}$ as well. These parameters represent the half-widths of the 95% confidence interval for the specified index [113].

Table 6.3: First order Sobol Sensitivity indices (S_1) and confidence interval half-widths ($S_{1,conf}$) for the position and velocity difference magnitudes between the tether tip and the payload at $t = 0$.

Physical Parameter	**Pos.** S_1	**Pos.** $S_{1,conf}$	**Vel.** S_1	**Vel.** $S_{1,conf}$
ρ	0.0313	0.0114	0.0149	0.0106
m_1	0.0046	0.0006	−0.0002	0.0006
m_2	0.0244	0.0165	0.0225	0.0163
L	0.8924	0.1410	0.9390	0.1299

From the S_1 values in Table 6.3, it is clear that the total tether length L has the largest individual effect on the variance of both the position and velocity tip-payload separations, based on its comparatively high S_1 values (about 89% and 94% variance contribution just from L). The effects of the other parameters are nearly negligible compared to that of L. This insight makes sense, the overall tether length contributes significantly to the positioning of the tether tip relative to the COM of the tether system. A higher value for L results in an increased COM offset distance (see equation for d_{CM} in Chapter 5). This effect not only changes the distance between the tether tip and the payload at the nominal rendezvous point, but also contributes to an increased rotational tangential velocity of the tip relative to the payload due to the tethers rotation rate. Thus an accurate input L value is crucial for accurate tip state representation.

After the tether length, it is the counterweight mass and linear density that contribute to tip state variation. These effects are quite small, but can be attributed to the change in moment of inertia of the tether system resulting from a different mass distribution. This also explains why the tip mass effects are even smaller; the masses of the tether and counterweight are at least an order of magnitude higher than that of the tip. Hence the scale of the tip mass changes barely affect the tether system's moment of inertia, which

impacts the rotation rate through conservation of angular momentum. An increased moment of inertia will lead to slower rotation of the uncontrolled tether, which then leads to a mismatch in the position and velocity of the tip and payload.

Table 6.4: Second order Sobol Sensitivity indices (S_2) and confidence interval half-widths ($S_{2,conf}$) for the position and velocity difference magnitudes between the tether tip and the payload at $t = 0$.

Parameter combinations	**Pos.** S_2	**Pos.** $S_{2,conf}$	**Vel.** S_2	**Vel.** $S_{2,conf}$
$(\rho,\ m_1)$	−0.0262	0.0570	−0.0234	0.0340
$(\rho,\ m_2)$	−0.0386	0.0556	−0.0320	0.0389
$(\rho,\ L)$	−0.0034	0.0623	−0.0249	0.0685
$(m_1,\ m_2)$	−0.0059	0.0119	−0.0027	0.0129
$(m_1,\ L)$	−0.0015	0.0108	0.0040	0.0144
$(m_2,\ L)$	0.0541	0.0585	0.0226	0.0602

Considering the S_2 values in Table 6.4 also yields interesting insights. The S_2 indices capture the variation in model outcome due to interaction between parameters. From Table 6.4 it is clear that the $(m_2,\ L)$ combination has the largest impact on the relative position and velocity separations between the tip and payload, though much smaller than the S_1 effects. The same argument made for the increased COM offset can be made here again, as increases in both L and m_2 drive a COM offset increase. The S_2 values of other combinations are mostly negative with comparatively large $S_{2,conf}$ values. However, because the positional $S_{2,conf}$ value is close to its S_2 counterpart, the 95% confidence interval (calculated as $[S_2 - S_{2,conf}, S_2 + S_{2,conf}]$), has a lower bound very close to zero. This means that the effect may be effectively zero. This is even clearer for the velocity case where $S_2 < S_{2,conf}$, which means the confidence interval contains the zero value and that the contribution can be neglected. The negative values for the other parameters are indications of noisy results, and may be concluded as statistically insignificant, as their 95% confidence intervals are also tightly packed around zero.

Monte-Carlo results and error ellipses

The Monte-Carlo simulation results are not only useful for determining model sensitivities, but also to gain insight on the position and velocity tolerances that a capture mechanism may need to account for. This section was inspired by [110], and closely follows the method laid out in their work. The authors of [110] proposed that the effective capture error "area" can be modelled as an ellipse (called the error ellipse) for the case of planar models, and an ellipsoid for a full three-dimensional case. Using the tether tip positional and velocity data gathered from the Monte-Carlo simulations, an error ellipse can be fitted to the position and velocity data. The fitting process is as follows:

1. Calculate the centroid of the data (x and y component means)
2. Centre the data by subtracting the centroid

3. Calculate the moments of inertia $I_{xx} = \sum x^2$, $I_{yy} = \sum y^2$ and product of inertia $I_{xy} = \sum xy$ by treating each data point as a particle
4. Assemble the moments and product of inertia into an inertia tensor
5. Determine the principal moments of inertia and principal directions through eigen value and eigen vector decomposition of the inertia tensor
6. Calculate the ellipse semi-major and semi-minor axes from the maximum and minimum principal inertias respectively
7. Calculate the error ellipse orientation angle from the major and minor princpal vectors

With these steps applied to the gathered Monte-Carlo position and velocity data, Figure 6.7 shows the error ellipses found.

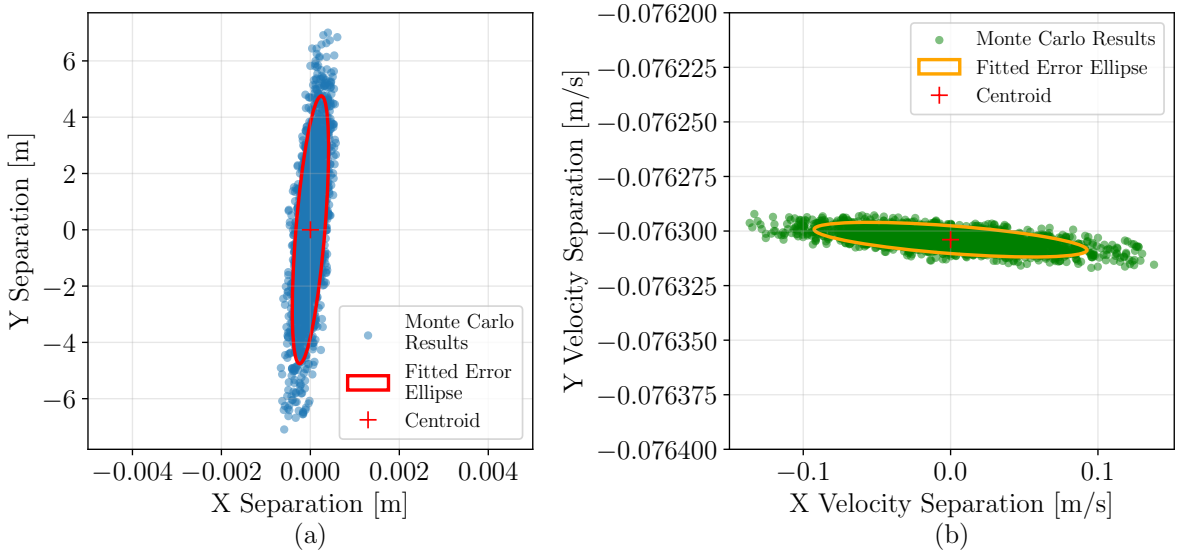


Figure 6.7: The error ellipses fit to the (a) positional and (b) velocity difference data. The zero point in both graphs represent the payload's state at time $t = 0$. Note that the vertical and horizontal axes have different scales.

While it may appear that the fitted ellipses poorly capture the outer sections along their major axes, it should be noted that the concentration of error points within the ellipse is so high that the inner points far outweigh the outer points in the moment of inertia calculations discussed earlier, resulting in a tighter error ellipse.

The keen reader might observe that both the position and velocity error ellipses have extremely high aspect ratios. The horizontal and vertical axes in these visualisations have been scaled to improve the ellipse visibility. However, the true aspect ratio (major to minor axis) is about 14560 : 1 for both ellipses. This is a much higher aspect ratio than the 10 : 1 ratio found in [110]. This difference likely comes down to the different models used. The work in [110] makes use of a flexible, elastic tether model. The resulting scale of the error ellipse much larger in absolute distance terms, as a result of the extensibility of the tether. The minor axis of the error ellipse is also larger than

that found in this work, likely due to the transverse flexibility considerations included in their model [110].

Since the position error point distribution lies predominantly along the y axis of Figure 6.7a, the positional error can undergo dimensional reduction by directly projecting these points onto the vertical axis. Usually, a complete principal component analysis would be preformed on data, but due to the sheer scale of the ellipse aspect ratio, the direct projection onto the vertical axis is sufficient. A normal distribution was then fit to the projected data to estimate its standard deviation from the mean. Figure 6.8 shows the outcome of this fitting, with the data points grouped into 100 bins. The resulting normal distribution curve has a mean of $\mu = 0$ m and a standard deviation $\sigma = 2.38$ m. This standard deviation value was used to determine an estimate for the positional capture tolerance a capture mechanism might have such that it would catch the payload with the tether tip being located at all of these error points. Taking a value of $4\sigma \approx 10$ m as the capture tolerance includes more than 99.99% of the error points. Even though this model has shown very little variation along the x-direction, this 10 m tolerance was also extended in the x-direction such that payload capture should occur whenever the tether tip is within a radial distance of 10 m from the payload. For more complex and higher fidelity models, such as the extensible, flexible model used in [23] and [110], this capture tolerance can and should be extended to account for perturbative as well as transverse and longitudinal wave effects within the tether. For the rigid-body model presented in Chapter 5 the 10 m value is deemed sufficient. The velocity error presented in Figure 6.7b is much smaller, and the velocity tolerance should not be taken as the maximum span of the error points along the horizontal axis, as this would be much too restrictive. Instead, a 10 m/s tolerance is applied here too, to make the tracking of the payload's trajectory a little more forgiving. In reality, the velocity error tolerance is more restrictive, with capture mechanisms needing to be tolerant to differences of only a few metres per second [27]. This can be counteracted in part by having the payload provide some of the change in velocity by thrusting on a planned approach. This is of course unwanted and should be minimised, as the main benefit of the tether system is its self-sufficiency and momentum transfer without the need for propellant expenditure.

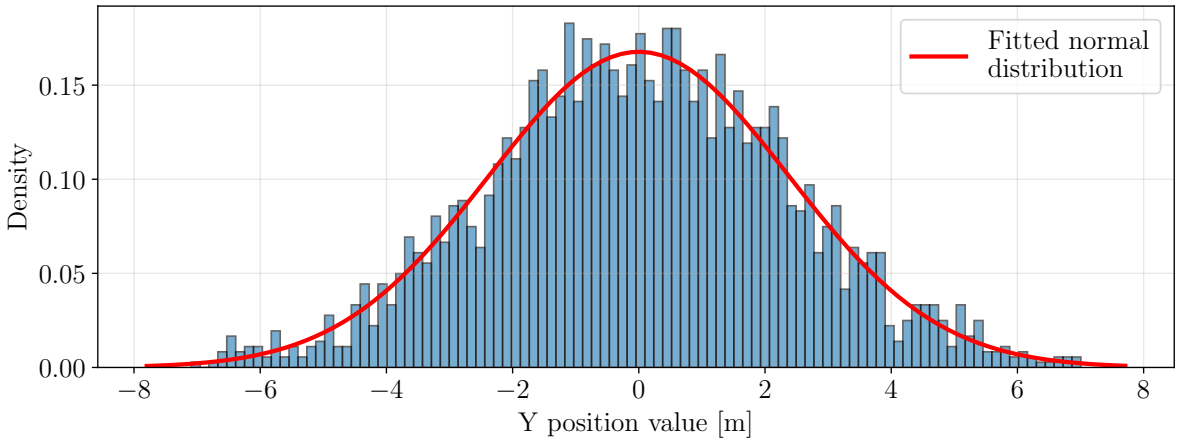


Figure 6.8: Normal distribution fit to the y position data of Figure 6.7, with mean of 0 m, and standard deviation of 2.38 m.

6.2.2. Reinforcement Learning Hyperparameter Impact Determination

Within the context of the unconstrained environment, the performance of the SAC RL agent was optimised by tuning a total of ten parameters, the four unconstrained reward function coefficients (A_{track}, C_{step}, C_{stay}, k_{shape}), and six underlying RL model hyperparameters.

The six key hyperparameters of the chosen Soft Actor-Critic (SAC) algorithm were identified for tuning by consulting common practices documented in the RL Baselines3 Zoo [114], a repository of tuned agents. These parameters include:

- **Learning Rate**: Controls the step size for gradient updates.
- **Batch Size**: Number of samples drawn from the replay buffer per update.
- **Replay Buffer Size**: The maximum number of transitions stored for experience replay.
- **Tau (τ)**: The smoothing coefficient for updating the target networks.
- **Discount Factor (γ)**: Determines the importance of future rewards.
- **Network Architecture**: The number and size of hidden layers for the policy and Q-value networks.

To assess the relative influence of the hyperparameters, the Optuna framework's fANOVA sensitivity analysis was employed, which estimates the contribution of each hyperparameter to the variance in the optimisation objective (the rendezvous window duration). The resulting importance scores are shown in Table 6.5, followed by the tuned hyperparameter values for the best-performing trial in Table 6.6.

Table 6.5: Relative reward and RL hyperparameter importance for the improvement of the rendezvous window duration during the combined Optuna study.

Hyperparameter	fANOVA Importance
Tracking Reward Scale (A_{track})	0.011
Shaping Scale (k_{shaping})	0.021
Step Penalty (C_{step})	0.023
In-Tolerance Bonus (C_{stay})	0.0091
Network Architecture	0.018
Learning Rate	**0.407**
Discount Factor (γ)	0.022
Batch Size	**0.085**
Target Update Rate (τ)	0.026
Buffer Size	0.003

Table 6.6: Tuned reward and RL hyperparameters for the best performing trial from the optimisation study.

Hyperparameter	**Tuned Values**
Tracking Reward Scale (A_{track})	7.1375
Shaping Scale (k_{shaping})	7.4719
Step Penalty (C_{step})	0.36677
In-Tolerance Bonus (C_{stay})	11.870
Network Architecture	[400, 300]
Learning Rate	9.8145e-4
Discount Factor (γ)	0.92858
Batch Size	512
Target Update Rate (τ)	0.015775
Buffer Size	10^6

Of the considered parameters, the learning rate exhibited the highest importance at 0.407, followed by the batch size with an importance of 0.085. The remaining hyperparameters had much lower importance, with values at or below 0.02. Notably, the tuned values for both the learning rate (0.001) and batch size (512) reached the upper bounds of their respective search ranges. This observation suggests that the performance of the RL algorithm may benefit from increased values for both hyperparameters, as they were identified as the dominant factors influencing the optimisation objective. However, these bounds were selected based on general stability guidelines for the application of the Soft Actor-Critic (SAC) algorithm to a variety of control tasks, as documented in the RL Baselines3 Zoo repository [114]. Any further increases would require careful monitoring to ensure training stability. This highlights a key takeaway that while most hyperparameters have a limited impact, the learning rate and batch size are critical and likely represent a primary avenue for future performance gains. Additionally, the relatively low importance scores of the reward parameters indicate that these parameters may benefit from a stand-alone tuning process, independent of the RL hyperparameters to better understand their relative importance amongst each other.

The RL hyperparameters and reward parameters were jointly tuned using a unified Optuna optimisation study comprising 100 trials, as described in Chapter 5. The considered value ranges for each parameter are listed in Table 6.7 and Table 6.8, respectively. Each trial involved training an RL agent across 60 parallel environments, distributed over the logical cores of a multi-core CPU. This parallelisation significantly accelerated individual trial runtimes by enabling simultaneous environment rollouts. A subset of RL hyperparameters was held constant throughout the study; these are summarised in Table 6.9.

Table 6.7: Reward function parameters considered in the optimisation study.

Hyperparameter	Function	Rationale	Tuning values
Tracking Reward Scale (A_{track})	Scales the magnitude of the core reward for minimising tracking error.	Determines the primary incentive for achieving the main objective of accurate tracking.	Uniform: [1.0, 20.0]
Shaping Scale (k_{shaping})	Scales the potential-based shaping reward for reducing tracking error.	Influences learning speed by providing a dense guidance signal without altering the optimal policy.	Uniform: [0.1, 10.0]
Step Penalty (C_{step})	Applies a constant negative reward at every timestep.	Encourages the agent to solve the task efficiently and discourages inaction or looping behaviour.	Uniform: [0.01, 1.0]
In-Tolerance Bonus (C_{stay})	Scales the bonus for each consecutive step inside the tolerance window.	Explicitly incentivises stable, persistent tracking over brief, repeated entries into the target zone.	Uniform: [1.0, 20.0]

Table 6.8: RL algorithm hyperparameters considered in the optimisation study.

Hyperparameter	Function	Rationale	Tuning values
Network Architecture	Size of the hidden layers in the actor and critic networks.	Affects the model's capacity to represent the policy and value functions, and its computational complexity.	Categorical: {[128, 128], [256, 256], [400, 300]}
Learning Rate	Step size for updating the neural network weights during training.	Influences the speed and stability of convergence; too high can cause instability, too low can cause slow learning.	Log-uniform: [1e-5, 1e-3]
Discount Factor (γ)	Weighting factor for future rewards in the Bellman equation.	Determines the agent's planning horizon; values closer to 1.0 encourage more farsighted behaviour.	Uniform: [0.9, 0.999]
Batch Size	Number of samples drawn from the replay buffer for each gradient update.	Affects the stability of the gradient estimate and the computational efficiency of the training step.	Categorical: {64, 128, 256, 512}
Target Update Rate (τ)	The interpolation factor for the soft update of the target networks.	Ensures stable learning by having the target networks track the main networks slowly and smoothly.	Uniform: [0.005, 0.02]
Buffer Size	The maximum number of past experiences to store for training.	Affects data diversity and the decorrelation of samples, which is crucial for stable off-policy learning.	Categorical: [100k, 500k, 1M, 2M]

Table 6.9: RL algorithm hyperparameters held constant during the optimisation study.

Hyperparameter	Function	Rationale	Fixed Value
Policy	Specifies the type of neural network policy to be used.	"MlpPolicy" is the standard policy for continuous control tasks with vector-based observations.	"MlpPolicy"
Learning Starts	Number of random steps to take before starting training updates.	Ensures the replay buffer is populated with diverse experiences before the agent begins to learn.	25 000
Gradient Steps	Number of gradient updates to perform per data collection cycle.	Balances the ratio of data collection to policy optimisation effort.	60
Train Frequency	Defines how often the learning update is triggered.	Determines how frequently the agent learns from newly collected experience.	(60, "step")

6.2.3. Brief Reinforcement Learning Generalisation Test

To evaluate the generalisation capability of the unconstrained reinforcement learning (RL) agent trained on the reeler tether configuration, the agent was subsequently applied to the climber configuration. Ideally, the baseline configuration would also have been included in this comparison. However, due to its reduced number of degrees of freedom (two, compared with three for both the climber and reeler), the baseline could not be incorporated without substantial modifications to the RL environment. The resulting tether tip trajectories are presented in Figure 6.9.

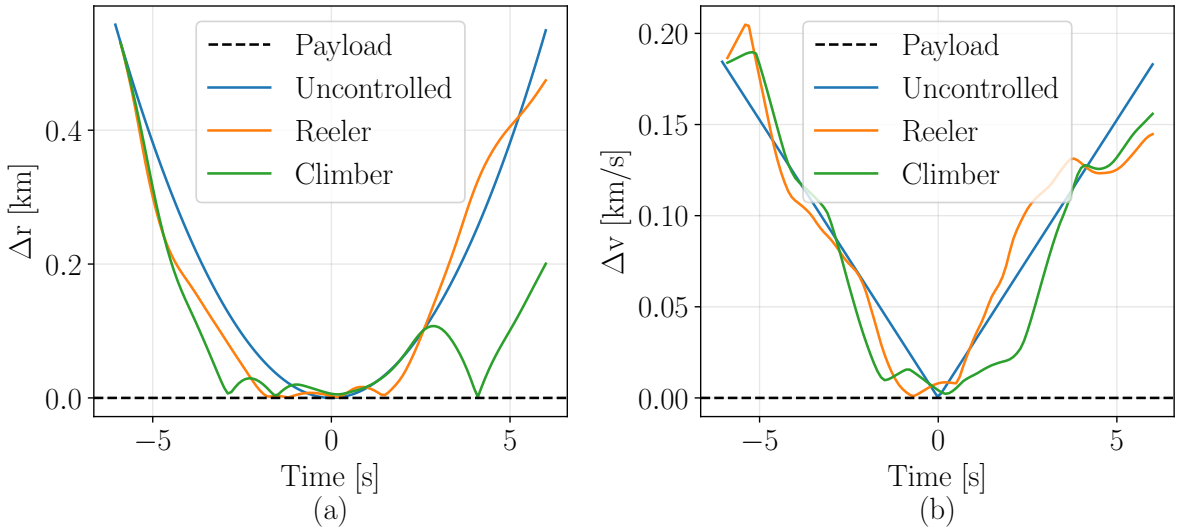


Figure 6.9: Comparison of tether tip trajectories for the uncontrolled case, and the unconstrained RL-controlled climber and reeler configurations. Subplots (a) and (b) show the magnitudes of the relative position and velocity difference vectors between the tether tip and payload, respectively.

As with the reeler configuration, Figure 6.9a shows that the relative position of the climber tether oscillates about the payload. However, these oscillations are more pronounced than in the reeler case, causing the intervals during which the climber tip remains within the positional tolerance to be repeatedly interrupted. Examination of the V-shaped velocity profile in Figure 6.9b indicates that the RL agent is able to reduce the sharpest point of the velocity curve in the vicinity of the payload, although not to the same extent as in the reeler configuration.

Unlike in previous cases with both AL-iLQR and RL control, the RL agent applied to the climber configuration results in the positional component becoming the limiting factor. The maximum duration within tolerance is 0.8 s for position, compared with 1.1 s for velocity. Consequently, the overall rendezvous window for this case is 0.8 s at the specified tolerance levels of 10 m and 10 m/s. This is shorter than the 1.0 s window obtained with the unconstrained and tuned iLQR solution reported in Chapter 5. On this basis, the current RL agent trained on the reeler configuration cannot be said to generalise successfully to the climber configuration. Nevertheless, it is noteworthy that the velocity component achieves a rendezvous time of 1.1 s, which exceeds the 1.0 s velocity result of the tuned AL-iLQR climber case. This suggests that with further hyperparameter tuning, the RL agent's generalisation performance could be improved.

7

Conclusion and Recommendations

The aim of this research was to address the critical control challenge of extending the short payload rendezvous window for Momentum Exchange with Electrodynamic Reboost (MXER) tether systems. This was accomplished by developing a verifiable 2D rigid-body dynamical model, performing a comparative analysis of three distinct actuator configurations, and evaluating the performance of a conventional optimal control method against a modern, model-free Deep Reinforcement Learning (DRL) algorithm. The main conclusions drawn from this work are presented in Section 7.1. Following this, a set of recommendations is offered for the design of future tether missions and for the direction of subsequent research in Section 7.2. The work is concluded with a final assessment of the motivating research questions in Section 7.3 and a review of the project's compliance with its stated goals in Section 7.4.

7.1. Conclusions

This research has led to several key conclusions regarding the control of spinning MXER tethers for extended rendezvous.

The primary contribution of this work is the definitive identification of the **reeler actuator configuration as the most effective** for extending the rendezvous window in an unconstrained dynamic environment. The reeler configuration, which utilises an intermediate reeling mass, extended the capture window to 1.8 seconds, a threefold improvement over the 0.6-second uncontrolled baseline and significantly outperforming both the baseline tip-reeling (0.8 s) and climber (1.0 s) configurations. This superior performance stems from its enhanced control authority over the tether tip's velocity profile, allowing it to more effectively counteract the V-shaped relative velocity curve that characterises the rendezvous.

In the unconstrained case, both the conventional **iterative Linear Quadratic Regulator (iLQR)** and the model-free **Soft Actor-Critic (SAC) DRL** agent successfully learned control policies to achieve the 1.8-second rendezvous window extension. However, the SAC agent produced a less smooth control policy, characterised by sporadic actuator use. This behaviour is undesirable in practical applications, as it can induce

high structural loads, cause component wear, and potentially excite unmodelled high-frequency wave dynamics, thereby compromising system reliability.

The investigation into constrained control revealed the significant difficulty of the problem. When realistic operational limits on tether tension, g-loads, and actuator use were imposed, **neither the Augmented-Lagrangian iLQR (AL-iLQR) nor the SAC-based controller succeeded in extending the rendezvous window**. The AL-iLQR method, while successfully satisfying all constraints, proved to be overly conservative, failing to exploit the system's full dynamic capabilities. Conversely, the SAC agent, guided by a simple penalty-based reward function, failed to robustly enforce critical constraints, notably violating the minimum tension requirement, which would result in a slack tether and a potential loss of control in a real system.

Finally, the verification and validation studies confirmed the fidelity of the developed rigid-body model for the scope of this research. The model's dynamics aligned well with established dumbbell models, and a variance-based sensitivity analysis revealed that uncertainty in the **total tether length is the dominant factor affecting rendezvous accuracy**, far outweighing the influence of mass properties. This highlights the critical need for high-precision manufacturing and state estimation in operational tether systems.

7.2. Recommendations

Based on the conclusions of this work, the following recommendations are made for the design of future tether missions and for subsequent research and development.

7.2.1. Recommendations for Mission Design

- **Actuator Configuration Selection:** For future spinning MXER tether systems where extending the rendezvous window is a primary objective, the reeler configuration is strongly recommended over simpler tip-reeling or climbing-mass designs due to its superior control authority.
- **Control System Smoothness:** Mission specifications for tether control systems should include stringent requirements for control policy smoothness. Controllers that exhibit sporadic, high-frequency actuation, as observed with the DRL agent, should be avoided or modified to ensure the structural integrity and long-term reliability of the tether and its mechanical components.
- **Prioritise Tether Length Metrology:** The sensitivity analysis demonstrated that tether length is the most critical parameter for rendezvous accuracy. Mission designers should prioritise the development and integration of robust on-orbit state estimation systems to accurately measure and monitor the tether's length throughout its operational life.

7.2.2. Methods and Future Research

- **Advanced Constrained Optimal Control:** The conservative nature of the AL-iLQR controller suggests that more robust optimisation methods are needed.

Future research should investigate advanced non-linear trajectory optimisation and control techniques that can more effectively navigate the constrained trajectory space and identify solutions that operate closer to the system's physical limits without violating them. Examples may include pseudospectral or direct collocation methods.

- **Constraint-Aware Reinforcement Learning:** The failure of the simple penalty-based approach for the SAC agent indicates a need for more sophisticated methods. Future work in applying DRL to this problem should move beyond simple penalties and explore dedicated constraint-aware RL algorithms. Techniques such as Lagrangian-based methods, state-augmented objectives, or other safe-RL approaches should be investigated to ensure reliable operation within critical constraints.
- **High-Fidelity Model Validation:** The conclusions of this study are based on a 2D rigid-body model. The next critical step is to validate these findings using a higher-fidelity, possibly three dimensional elastic tether model. Such a model would account for out-of-plane dynamics, longitudinal and transverse wave propagation, and bending stiffness, all of which are crucial for the development of a flight-ready control system.
- **Automated Hyperparameter Optimisation:** The hyperparameter sensitivity analysis for the SAC agent identified the learning rate and batch size as highly influential. It is recommended to investigate the effect of pushing these hyperparameters beyond the heuristic values explored in this work for potential performance improvements. Future DRL implementations should continue or expand upon the use of advanced automated hyperparameter optimisation frameworks, such as Bayesian optimisation or evolutionary algorithms, to systematically tune the agent for improved performance, stability, and sample efficiency.

7.3. Final Assessment of Research Questions

This section provides a direct assessment of the research questions posed in Chapter 4.

Table 7.1: Final assessment of research questions.

Code	Question	Answer
RQ.1	**How does the actuator configuration influence the controllability of the rendezvous window of a rotating momentum exchange tether?**	The actuator configuration significantly influences controllability. The reeler configuration demonstrated the highest level of control, extending the rendezvous window to 1.8 s. The climber (1.0 s) and baseline tip-reeling (0.8 s) configurations showed progressively less effectiveness. The reeler's superior performance is attributed to its greater authority in shaping the tether tip's velocity profile.
SQ.1a	What are the dominant dynamical factors affecting the tether tip motion and rendezvous window in a simplified MXER system model?	The dominant dynamical factors are the centrifugal accelerations from the tether's rotation combined with its orbital motion. This creates a distinct V-shaped relative velocity profile between the tether tip and the payload, which fundamentally limits the rendezvous window to less than a few seconds. The ability of a control system to "flatten" this velocity curve is the primary determinant of its success.
SQ.1b	What are the key performance indicators (KPIs) for evaluating the controllability of the rendezvous window for the considered actuator configurations?	The primary KPI is the **rendezvous window duration**, defined as the longest continuous period during which the relative position and velocity between the tether tip and payload remain within specified tolerances (10 m and 10 m/s). Secondary KPIs considered were the smoothness of the control policy and the satisfaction of operational constraints.

Continued on next page

Table 7.1: Final assessment of research questions (continued).

Code	Question	Answer
RQ.2	**Can Reinforcement Learning (RL) be used as an advanced control strategy to effectively extend the rendezvous window for payload capture in a rotating momentum exchange tether system?**	Yes, in an unconstrained setting, the SAC DRL algorithm proved capable of learning a control policy that matched the performance of a tuned iLQR controller, successfully extending the rendezvous window to 1.8 s. However, when operational constraints were introduced, the implemented DRL approach failed to find a viable solution that both extended the window and respected all constraints, indicating that more advanced, constraint-aware RL methods are required for practical application.
SQ.2a	What are the suitable state and action spaces for representing the MXER system and control inputs within a reinforcement learning framework?	A suitable **state space** consists of the element-wise difference between the tether tip and payload state vectors (relative position and velocity), augmented with the current simulation time. This provides the agent with a complete representation of the tracking error. A suitable **action space** consists of the normalised reeling acceleration for each active reeling mechanism, which directly maps to the control inputs of the dynamical model.

Continued on next page

Table 7.1: Final assessment of research questions (continued).

Code	Question	Answer
SQ.2b	How do different reward function formulations impact the performance of the RL algorithms?	A dense reward function, composed of quadratic penalties for state error and control effort, potential-based reward shaping, and a bonus for maintaining tolerance, proved effective for learning in the unconstrained environment. Three alternative stay-in-tolerance schemes, namely a fixed bonus, a Gaussian error-dependent bonus, and a curriculum-based tolerance tightening, were also evaluated, but none yielded sustained rendezvous windows. For the constrained case, augmenting this function with simple quadratic penalties for constraint violation was insufficient to produce a safe and effective policy, leading to constraint violations.
SQ.2c	How does the chosen SAC RL algorithm compare against the ILQR controller in terms of rendezvous window extension and suitability to the tether control task?	The SAC agent matched the rendezvous window extension of the tuned iLQR baseline. It required extensive hyperparameter optimisation, but once tuned, the training process was computationally efficient. The resulting control effort was less smooth than that of the iLQR controller making the current implementation less suited to the control task than the iLQR controller.

7.4. Final Assessment of Compliance with Project Objectives

This section provides a final review of the project's compliance with the main goals and subgoals outlined in Chapter 4

Table 7.2: Final assessment of compliance with project objectives.

Code	Goal	Achieved	Executive Summary
G.1	**To derive and implement a 2D tether dynamics model which can be used and adapted to simulate and ultimately compare the dynamical behaviour of three tether systems with different actuator configurations.**	✓	A generalised 2D rigid-body tether model was successfully derived, implemented, and presented in the conference paper in Chapter 5. The model was validated against established benchmarks in Chapter 6, confirming its fidelity for this study.
G.2	**To implement an iterative linear quadratic regulator (ILQR) controller to establish a baseline for the trajectory-tracking control of the 2D tether dynamics model.**	✓	An Augmented-Lagrangian iLQR controller was developed and implemented as described in Chapter 5. This controller was used to establish the performance baseline for both the unconstrained and constrained control scenarios, against which the DRL agent was compared.

Continued on next page

Table 7.2: Final assessment of compliance with project objectives (continued).

Code	Goal	Achieved	Executive Summary
G.3	**To implement and test the model-free SAC RL algorithm as a control method for the rendezvous dynamics of tether-payload capture.**	✓	The model-free SAC algorithm was successfully implemented. A custom training environment was built, and the agent's performance was optimised through extensive hyperparameter tuning. The SAC agent's final performance was comprehensively tested and compared against the iLQR baseline in both unconstrained and constrained scenarios, as detailed in Chapter 5.
SG.1a	To derive a generalised model suitable for all three considered tether configurations.	✓	The dynamical model presented in the paper in Chapter 5 is generalised, accommodating all three actuator configurations (baseline, climber, and reeler) through minor modifications to the system's mass distribution matrices.
SG.1b	To identify suitable control input(s) to the dynamics model for effective tether tip control.	✓	Tether reeling acceleration for each active point mass was identified and implemented as the direct control input. This choice provided direct, independent control and ensured state continuity, proving effective for both the iLQR and RL controllers.
SG.2a	To implement an ILQR controller capable of working with general, black-box dynamics functions	✓	The core iLQR implementation was designed to interface with a generic dynamics function, making it adaptable to different models. This modularity was essential for its application to the tether system.

Continued on next page

Table 7.2: Final assessment of compliance with project objectives (continued).

Code	Goal	Achieved	Executive Summary
SG.2b	To expand the ILQR controller to incorporate general (soft) constraints on the dynamics and controls.	✓	The iLQR was embedded within an Augmented Lagrangian framework, enabling the incorporation of soft constraints on tether tension, g-loads, and actuator limits, as detailed in the constrained control analysis in Chapter 5.
SG.3a	To implement a RL-friendly training environment of the tether dynamics.	✓	A custom environment conforming to the Gymnasium API standard was developed. This environment integrated the tether dynamics, defined the state and action spaces, and implemented the dense reward function necessary for effective agent training.
SG.3b	To compare the performance of the SAC RL algorithm against the ILQR controller based on their suitability to the tether control task.	✓	A detailed comparative analysis was performed in Chapter 5. The comparison covered rendezvous window extension, control policy smoothness, and constraint satisfaction, providing a clear assessment of the relative strengths and weaknesses of each approach for this specific task.
SG.3c	To identify the most impactful hyperparameters for the best-ranked RL model and their relative effects on tether control performance.	✓	A variance-based hyperparameter importance analysis (fANOVA) was conducted in Chapter 6. This analysis identified the learning rate and batch size as the most critical hyperparameters influencing the performance and stability of the SAC agent.

Continued on next page

Table 7.2: Final assessment of compliance with project objectives (continued).

Code	Goal	Achieved	Executive Summary
SG.3d	To improve the tether control performance of the RL algorithm through tuning of commonly identified impactful hyperparameters.	✓	An extensive, two-phase hyperparameter optimisation study using the Optuna framework was performed, as described in Chapter 5 and detailed in Chapter 6. This process systematically tuned all ten reward and agent hyperparameters, leading to the high-performing final agent used for the comparative analysis.

Notes

References

[1] M. Cartmell and D. McKenzie, "A review of space tether research," *Progress in Aerospace Sciences*, vol. 44, no. 1, pp. 1–21, Jan. 2008, ISSN: 03760421. DOI: 10.1016/j.paerosci.2007.08.002.

[2] J. A. Carroll, "Tether applications in space transportation," *Acta Astronautica*, vol. 13, no. 4, pp. 165–174, Apr. 1986, ISSN: 00945765. DOI: 10.1016/0094-5765(86)90061-5.

[3] V. V. Beletsky and E. M. Levin, *Dynamics of Space Tether Systems* (Advances in the Astronautical Sciences 83). San Diego, Calif: Published for the American Astronautical Society by Univelt, Inc, 1993, 499 pp., ISBN: 978-0-87703-370-7 978-0-87703-371-4.

[4] J. A. Carroll, "Guidebook for analysis of tether applications," NASA-CR-178904, Mar. 1, 1985. [Online]. Available: https://ntrs.nasa.gov/citations/19870000736.

[5] K. Sorensen, "Conceptual design and analysis of an MXER tether boost station," in *37th Joint Propulsion Conference and Exhibit*, Jul. 8, 2001. DOI: 10.2514/6.2001-3915.

[6] R. P. Hoyt, "Cislunar Tether Transport System," NIAC-07600-011, May 30, 1999. [Online]. Available: https://www.niac.usra.edu/files/studies/final_report/7Hoyt.pdf.

[7] R. P. Hoyt, "The Cislunar Tether Transport System Architecture," Jul. 20, 2000.

[8] Tethers Unlimited, Inc., "Moon & Mars Orbiting Spinning Tether Transport Architecture Study," NASA Institute for Advanced Concepts, Final Report on NIAC Phase II Contract 07600-034, Aug. 31, 2001, p. 336. [Online]. Available: https://www.niac.usra.edu/files/studies/final_report/373Hoyt.pdf.

[9] G. Nordley, "Tether-tossed Mars mission examples," in *37th Joint Propulsion Conference and Exhibit*, Salt Lake City,UT,U.S.A.: American Institute of Aeronautics and Astronautics, Jul. 8, 2001. DOI: 10.2514/6.2001-3375.

[10] R. P. Hoyt, "Momentum-exchange/electrodynamic-reboost tether facility for deployment of microsatellites to GEO and the Moon," *AIP Conference Proceedings*, vol. 552, no. 1, pp. 508–513, 2001, ISSN: 0094-243X. DOI: 10.1063/1.1357969.

[11] G. D. Nordley and R. L. Forward, "Mars-earth rapid interplanetary tether transport system: I. initial feasibility analysis," *Journal of Propulsion and Power*, vol. 17, no. 3, pp. 499–507, 2001.

[12] L. W. Taylor, O. S. Dewey, R. J. Headrick, *et al.*, "Improved properties, increased production, and the path to broad adoption of carbon nanotube fibers," *Carbon*, vol. 171, pp. 689–694, 2021.

[13] R. FORWARD, "Failsafe multistrand tether structures for space propulsion," in *28th Joint Propulsion Conference and Exhibit*, 1992, p. 3214.

[14] R. Forward and R. Hoyt, "Failsafe multiline hoytether lifetimes," in *31st Joint Propulsion Conference and Exhibit*, 1995, p. 2890.

[15] NASA, *Tether Dynamics Simulation*. Feb. 1, 1987. [Online]. Available: https://ntrs.nasa.gov/citations/19870009388.

[16] B. Vossoughi, M. Kanzow, and S. Klinkner, "Modeling and control of perturbation torques and mass distribution impact on a tethered system for a 12u cubesat in sun-synchronous orbit," *Acta Astronautica*, vol. 225, pp. 565–575, 2024.

[17] J. A. Carroll, "Tether fundamentals," Jun. 1, 1986. [Online]. Available: https://ntrs.nasa.gov/citations/19860018938 (visited on 06/23/2025).

[18] E. M. Levin, *Dynamic Analysis of Space Tether Missions* (Advances in the Astronautical Sciences v. 126). San Diego, Calif: Published for the American Astronautical Society by Univelt, Inc, 2007, 453 pp., ISBN: 978-0-87703-537-4.

[19] M. M. Finckernor, K. A. Gitlemeier, C. W. Hawk, and E. Watts. "Low Earth Orbit Environmental Effects on Space Tether Materials." (Jan. 1, 2005), [Online]. Available: https://ntrs.nasa.gov/citations/20050215563.

[20] P. Williams, C. Blanksby, P. Trivailo, and H. Fujii, "In-plane payload capture using tethers," *Acta Astronautica*, vol. 57, no. 10, pp. 772–787, Nov. 1, 2005, ISSN: 0094-5765. DOI: 10.1016/j.actaastro.2005.03.069.

[21] M. L. Cosmo and E. C. Lorenzini, "Tethers in Space Handbook," NASA/CR-97-206807, Dec. 1, 1997. [Online]. Available: https://ntrs.nasa.gov/citations/19980018321.

[22] P. Williams, "A Review of Space Tether Technology," *Recent Patents on Space Technology*, vol. 2, no. 1, pp. 22–36, 2012. DOI: 10.2174/1877611611202010022.

[23] S. L. Canfield, "Developing Capture Mechanisms and High-Fidelity Dynamic Models for the MXER Tether System," NASA/CR-2007- 215076, Sep. 1, 2007. [Online]. Available: https://ntrs.nasa.gov/citations/20080002099.

[24] K. F. Sorensen, S. L. Canfield, and M. A. Norris. "Design Rules and Analysis of a Capture Mechanism for Rendezvous between a Space Tether and Payload." (Jan. 1, 2006), [Online]. Available: https://ntrs.nasa.gov/citations/20070001536 (visited on 12/05/2024).

[25] W. D. Dorland, "Dynamic testing of docking system hardware," no. 19730010158, Nov. 1, 1972. [Online]. Available: https://ntrs.nasa.gov/citations/19730010158.

[26] J. Westerhoff, "Active Control for MXER Tether Rendezvous Maneuvers," in *39th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, Huntsville, Alabama: American Institute of Aeronautics and Astronautics, Jul. 20, 2003, ISBN: 978-1-62410-098-7. DOI: 10.2514/6.2003-5218.

[27] P. Williams, "Dynamics and Control of Spinning Tethers for Rendezvous in Elliptic Orbits," *Journal of Vibration and Control*, vol. 12, no. 7, pp. 737–771, Jul. 1, 2006, ISSN: 1077-5463. DOI: 10.1177/1077546306065710.

[28] R. Hoyt, "Tether rendezvous methods," *Interim Report on NIAC Phase II Contract*, pp. 07 600–034, 2000.

[29] Y. N. Andres, F. Zimmermann, and U. M. Schoettle, "Optimization and control of the early deployment phase during a tether assisted deorbit maneuver," in *International Symposium on Space Technology and Science, 22 nd, Morioka, Japan*, 2000, pp. 1795–1800.

[30] P. Williams, C. Blanksby, P. Trivailo, and H. A. Fujii, "Receding horizon control of tether system using quasilinearisation and chebyshev pseudospectral approximations," *Advances in the Astronautical Sciences*, vol. 116, pp. 539–558, 2003.

[31] R. Licata, "Tethered system deployment controls by feedback fuzzy logic," *Acta astronautica*, vol. 40, no. 9, pp. 619–634, 1997.

[32] P. Williams, "In-Plane Payload Capture with an Elastic Tether," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 4, pp. 810–821, Jul. 2006, ISSN: 0731-5090, 1533-3884. DOI: 10.2514/1.17474.

[33] P. Williams, "Optimal Deployment/Retrieval of Tethered Satellites," *Journal of Spacecraft and Rockets*, vol. 45, no. 2, pp. 324–343, Mar. 2008, ISSN: 0022-4650, 1533-6794. DOI: 10.2514/1.31804.

[34] E. C. Lorenzini, "Error-Tolerant Technique For Catching a Spacecraft With a Spinning Tether," *Journal of Vibration and Control*, vol. 10, no. 10, pp. 1473–1491, Oct. 2004, ISSN: 1077-5463, 1741-2986. DOI: 10.1177/1077546304042062.

[35] C. Blanksby and P. Trivailo, "Assessment of actuation methods for manipulating tip position of long tethers," in *IAF, International Astronautical Congress, 50 th, Amsterdam, Netherlands*, 1999.

[36] P. Williams and C. Blanksby, "Prolonged Payload Rendezvous Using a Tether Actuator Mass," *ResearchGate*, 2004. DOI: 10.2514/1.6074.

[37] R. S. Sutton and A. Barto, *Reinforcement Learning: An Introduction* (Adaptive Computation and Machine Learning), Second edition. Cambridge, Massachusetts London, England: The MIT Press, 2020, 526 pp., ISBN: 978-0-262-03924-6.

[38] S. L. Brunton and J. N. Kutz. "Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control," Higher Education from Cambridge University Press. (May 5, 2022).

[39] C. Banerjee, K. Nguyen, C. Fookes, and M. Raissi. "A Survey on Physics Informed Reinforcement Learning: Review and Open Problems." arXiv: 2309.01909 [cs]. (Sep. 5, 2023), pre-published.

[40] D. Silver, J. Schrittwieser, K. Simonyan, *et al.*, "Mastering the game of go without human knowledge," *nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[41] P. Kormushev, S. Calinon, and D. G. Caldwell, "Reinforcement learning in robotics: Applications and real-world challenges," *Robotics*, vol. 2, no. 3, pp. 122–148, 2013.

[42] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, "Optimal and Autonomous Control Using Reinforcement Learning: A Survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 6, pp. 2042–2062, Jun. 2018, ISSN: 2162-237X, 2162-2388. DOI: 10.1109/TNNLS.2017.2773458.

[43] C. Packer, K. Gao, J. Kos, P. Krähenbühl, V. Koltun, and D. Song, "Assessing Generalization in Deep Reinforcement Learning," *ArXiv*, Sep. 27, 2018. [Online]. Available: https://www.semanticscholar.org/paper/caea502325b6a82b1b437c62585992609b5aa542.

[44] MathWorks. "Define reward and observation signals in custom environments." Accessed on 2 February 2025, MathWorks. (2025), [Online]. Available: https://www.mathworks.com/help/reinforcement-learning/ug/define-reward-and-observation-signals.html (visited on 02/02/2025).

[45] K. Liu, Y. Fu, L. Wu, X. Li, C. Aggarwal, and H. Xiong, "Automated feature selection: A reinforcement learning perspective," *IEEE Transactions on Knowledge*

and Data Engineering, vol. 35, no. 3, pp. 2272–2284, 2023. DOI: 10.1109/TKDE.2021.3115477.

[46] Y. G. Jahed and S. A. S. Tavana, "Enhancing classification performance via reinforcement learning for feature selection," *ArXiv*, vol. abs/2403.05979, 2024. DOI: 10.48550/arXiv.2403.05979.

[47] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, Jan. 1989, ISSN: 08936080. DOI: 10.1016/0893-6080(89)90020-8.

[48] Y. Huang, W.-d. Jin, Z. Yu, and B. Li, "Supervised feature selection through deep neural networks with pairwise connected structure," *Knowl. Based Syst.*, vol. 204, p. 106 202, 2020. DOI: 10.1016/j.knosys.2020.106202.

[49] S. Mysore, B. Mabsout, R. Mancuso, and K. Saenko, "Regularizing action policies for smooth control with reinforcement learning," 2021.

[50] S. Ibrahim, M. Mostafa, A. Jnadi, H. Salloum, and P. Osinenko, "Comprehensive Overview of Reward Engineering and Shaping in Advancing Reinforcement Learning Applications," *IEEE Access*, vol. 12, pp. 175 473–175 500, 2024, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2024.3504735.

[51] A. Gupta, A. Pacchiano, Y. Zhai, S. Kakade, and S. Levine, "Unpacking Reward Shaping: Understanding the Benefits of Reward Engineering on Sample Complexity," in *Advances in Neural Information Processing Systems*, vol. 35, Dec. 6, 2022, pp. 15 281–15 295. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/hash/6255f22349da5f2126dfc0b007075450-Abstract-Conference.html.

[52] A. Abouelazm, J. Michel, and J. M. Zoellner, "A review of reward functions for reinforcement learning in the context of autonomous driving," *arXiv preprint arXiv:2405.01440*, 2024.

[53] A. Nagabandi, G. Kahn, R. S. Fearing, and S. Levine, "Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 7559–7566. DOI: 10.1109/ICRA.2018.8463189.

[54] W. Sun, N. Jiang, A. Krishnamurthy, A. Agarwal, and J. Langford, "Model-based rl in contextual decision processes: Pac bounds and exponential improvements over model-free approaches," in *Annual Conference Computational Learning Theory*, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:59600077.

[55] N. Lambert, A. Wilcox, H. Zhang, K. S. J. Pister, and R. Calandra, "Learning accurate long-term dynamics for model-based reinforcement learning," *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 2880–2887, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:229211063.

[56] V. Mnih, K. Kavukcuoglu, D. Silver, *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[57] G. A. Rummery and M. Niranjan, *On-line Q-learning using connectionist systems*. University of Cambridge, Department of Engineering Cambridge, UK, 1994, vol. 37.

[58] C. J. C. H. Watkins, "Learning from delayed rewards," 1989.

[59] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, 2016.

[60] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning," in *International conference on machine learning*, PMLR, 2016, pp. 1995–2003.

[61] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in Neural Information Processing Systems*, S. Solla, T. Leen, and K. Müller, Eds., vol. 12, MIT Press, 1999. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/1999/file/464d828b85b0bed98e80ade0a5c43b0f-Paper.pdf.

[62] H. Kimura, "Reinforcement learning for continuous action using stochastic gradient ascent," *Intelligent Autonomomus Systems*, 1998.

[63] J. Schmidhuber and J. Zhao, "Direct policy search and uncertain policy evaluation," in *Aaai spring symposium on search under uncertain and incomplete information, stanford univ*, 1998, pp. 119–124.

[64] C. Wu, A. Rajeswaran, Y. Duan, *et al.*, "Variance reduction for policy gradient with action-dependent factorized baselines," *ArXiv*, vol. abs/1803.07246, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:4043645.

[65] M. Gargiani, A. Zanelli, A. Martinelli, T. Summers, and J. Lygeros, "Page-pg: A simple and loopless variance-reduced policy gradient method with probabilistic gradient estimation," in *International Conference on Machine Learning*, PMLR, 2022, pp. 7223–7240.

[66] G. Tucker, S. Bhupatiraju, S. Gu, R. Turner, Z. Ghahramani, and S. Levine, "The mirage of action-dependent baselines in reinforcement learning," in *International conference on machine learning*, PMLR, 2018, pp. 5015–5024.

[67] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, pp. 229–256, 1992.

[68] S. M. Kakade, "A natural policy gradient," *Advances in neural information processing systems*, vol. 14, 2001.

[69] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 37, 2015, pp. 1889–1897. [Online]. Available: https://proceedings.mlr.press/v37/schulman15.html.

[70] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. "Proximal Policy Optimization Algorithms." arXiv: 1707.06347 [cs]. (Aug. 28, 2017), prepublished.

[71] V. Konda and J. Tsitsiklis, "Actor-critic algorithms," in *Advances in Neural Information Processing Systems*, S. Solla, T. Leen, and K. Müller, Eds., vol. 12, MIT Press, 1999. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/1999/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

[72] D. Silver, *Lectures on reinforcement learning*, URL: https://www.davidsilver.uk/teaching/, 2015.

[73] Z. Wang, V. Bapst, N. Heess, *et al.*, "Sample efficient actor-critic with experience replay," *arXiv preprint arXiv:1611.01224*, 2016.

[74] Y. Wu, E. Mansimov, R. B. Grosse, S. Liao, and J. Ba, "Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation," *Advances in neural information processing systems*, vol. 30, 2017.

[75] S. Parisi, V. Tangkaratt, J. Peters, and M. E. Khan, "TD-regularized actor-critic methods," *Machine Learning*, vol. 108, no. 8, pp. 1467–1501, Sep. 1, 2019, ISSN: 1573-0565. DOI: 10.1007/s10994-019-05788-0.

[76] I. Grondman, L. Busoniu, G. A. Lopes, and R. Babuska, "A survey of actor-critic reinforcement learning: Standard and natural policy gradients," *IEEE Transactions on Systems, Man, and Cybernetics, part C (applications and reviews)*, vol. 42, no. 6, pp. 1291–1307, 2012.

[77] V. Mnih, "Asynchronous methods for deep reinforcement learning," *arXiv preprint arXiv:1602.01783*, 2016.

[78] T. Haarnoja, A. Zhou, K. Hartikainen, *et al.* "Soft Actor-Critic Algorithms and Applications." arXiv: 1812.05905 [cs]. (Jan. 29, 2019), pre-published.

[79] J. Duan, Y. Guan, S. E. Li, Y. Ren, Q. Sun, and B. Cheng, "Distributional soft actor-critic: Off-policy reinforcement learning for addressing value estimation errors," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 11, pp. 6584–6598, 2022. DOI: 10.1109/TNNLS.2021.3082568.

[80] L. Graesser and W. L. Keng, *Foundations of deep reinforcement learning: theory and practice in Python*. Addison-Wesley Professional, 2019.

[81] M. Andrychowicz, F. Wolski, A. Ray, *et al.*, "Hindsight experience replay," *Advances in neural information processing systems*, vol. 30, 2017.

[82] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*, PMLR, 2018, pp. 1861–1870.

[83] T. Haarnoja, S. Ha, A. Zhou, J. Tan, G. Tucker, and S. Levine, "Learning to walk via deep reinforcement learning," *arXiv preprint arXiv:1812.11103*, 2018.

[84] L. Buşoniu, T. de Bruin, D. Tolić, J. Kober, and I. Palunko, "Reinforcement learning for control: Performance, stability, and deep approximators," *Annual Reviews in Control*, vol. 46, pp. 8–28, 2018, ISSN: 1367-5788. DOI: https://doi.org/10.1016/j.arcontrol.2018.09.005.

[85] E. F. Camacho and C. Bordons, *Model Predictive Control* (Advanced Textbooks in Control and Signal Processing). London: Springer London, 2007. DOI: 10.1007/978-0-85729-398-5.

[86] S. Padakandla, "A survey of reinforcement learning algorithms for dynamically varying environments," *ACM Comput. Surv.*, vol. 54, no. 6, 2021. DOI: 10.1145/3459991.

[87] M. Han, L. Zhang, J. Wang, and W. Pan, "Actor-critic reinforcement learning for control with stability guarantee," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6217–6224, 2020. DOI: 10.1109/LRA.2020.3011351.

[88] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[89] B. R. Kiran, I. Sobh, V. Talpaert, *et al.*, "Deep reinforcement learning for autonomous driving: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 4909–4926, 2021.

[90] T. A. Badgwell, J. H. Lee, and K.-H. Liu, "Reinforcement learning–overview of recent progress and implications for process control," *Computer Aided Chemical Engineering*, vol. 44, pp. 71–85, 2018.

[91] L. Federici, B. Benedikter, and A. Zavoli, "Deep learning techniques for autonomous spacecraft guidance during proximity operations," *Journal of Spacecraft and Rockets*, vol. 58, no. 6, pp. 1774–1785, 2021.

[92] H. Dong, X. Zhao, and H. Yang, "Reinforcement learning-based approximate optimal control for attitude reorientation under state constraints," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 4, pp. 1664–1673, 2020.

[93] H. Holt, R. Armellin, A. Scorsoglio, and R. Furfaro, "Low-thrust trajectory design using closed-loop feedback-driven control laws and state-dependent parameters," in *AIAA Scitech 2020 Forum*, 2020, p. 1694.

[94] A. Rajeswaran, S. Ghotra, B. Ravindran, and S. Levine, "Epopt: Learning robust neural network policies using model ensembles," *arXiv preprint arXiv:1610.01283*, 2016.

[95] Z. Zhu, K. Lin, A. K. Jain, and J. Zhou, "Transfer learning in deep reinforcement learning: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[96] H. Cai, T. Chen, W. Zhang, Y. Yu, and J. Wang, "Efficient architecture search by network transformation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, 2018.

[97] M. Spryn, A. Sharma, D. Parkar, and M. Shrimal, "Distributed deep reinforcement learning on the cloud for autonomous driving," in *Proceedings of the 1st International Workshop on Software Engineering for AI in Autonomous Systems*, 2018, pp. 16–22.

[98] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *arXiv preprint arXiv:1510.00149*, 2015.

[99] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization.," *Journal of machine learning research*, vol. 13, no. 2, 2012.

[100] F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Sequential model-based optimization for general algorithm configuration," in *Learning and Intelligent Optimization: 5th International Conference, LION 5, Rome, Italy, January 17-21, 2011. Selected Papers 5*, Springer, 2011, pp. 507–523.

[101] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[102] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete problems in ai safety," *arXiv preprint arXiv:1606.06565*, 2016.

[103] A. Pan, K. Bhatia, and J. Steinhardt, "The effects of reward misspecification: Mapping and mitigating misaligned models," *arXiv preprint arXiv:2201.03544*, 2022.

[104] A. Heuillet, F. Couthouis, and N. Díaz-Rodríguez, "Explainability in deep reinforcement learning," *Knowledge-Based Systems*, vol. 214, p. 106 685, 2021.

[105] C. Glanois, P. Weng, M. Zimmer, *et al.*, "A survey on interpretable reinforcement learning," *Machine Learning*, pp. 1–44, 2024.

[106] D. G. Stuart, "Guidance and control for cooperative tether-mediated orbital rendezvous," *Journal of Guidance, Control, and Dynamics*, vol. 13, no. 6, pp. 1102–1108, Nov. 1990. DOI: 10.2514/3.20585.

[107] M. Towers, A. Kwiatkowski, J. Terry, *et al.*, *Gymnasium: A standard interface for reinforcement learning environments*, 2024. arXiv: 2407.17032 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2407.17032.

[108] Y. Ahn, W. Jang, J. Lee, and J. Chung, "Dynamic Analysis of Tethered Satellites with a Payload Moving Along a Flexible Tether," *Applied Sciences*, vol. 14, no. 20, p. 9498, Oct. 17, 2024, ISSN: 2076-3417. DOI: 10.3390/app14209498.

[109] M. Kruijff, "Tethers in Space," Ph.D. dissertation, TU Delft, 2011. [Online]. Available: https://resolver.tudelft.nl/uuid:9d437e58-82c0-4af1-935f-69ba5573c7a2 (visited on 11/22/2024).

[110] S. L. Canfield, D. L. Chlarson, and K. Sorensen, "A Comparison of Three Passive Capture Mechanisms for Tether Momentum Exchange," American Society of Mechanical Engineers Digital Collection, May 20, 2009, pp. 239–249. DOI: 10.1115/DETC2007-35625. [Online]. Available: https://dx.doi.org/10.1115/DETC2007-35625.

[111] Teijin Frontier (U.S.A.), Inc. "Zylon." (), [Online]. Available: https://www.teijin-frontier-usa.com/zylon/ (visited on 07/21/2025).

[112] T. Iwanaga, W. Usher, and J. Herman, "Toward SALib 2.0: Advancing the accessibility and interpretability of global sensitivity analyses," *Socio-Environmental Systems Modelling*, vol. 4, p. 18 155, May 2022. DOI: 10.18174/sesmo.18155. [Online]. Available: https://sesmo.org/article/view/18155.

[113] J. Herman and W. Usher, "SALib: An open-source python library for sensitivity analysis," *The Journal of Open Source Software*, vol. 2, no. 9, Jan. 2017. DOI: 10.21105/joss.00097. [Online]. Available: https://doi.org/10.21105/joss.00097.

[114] A. Raffin, *Rl baselines3 zoo*, https://github.com/DLR-RM/rl-baselines3-zoo, 2020.

A

IAC Paper: Supporting material

A.1. Unconstrained Conventional Control

A.1.1. State Trajectories

The unconstrained, iLQR controlled configuration comparison results presented in Chapter 5 are extended to include the state components in Figure A.1.

From Figure A.2a and A.1c it is clear that the unconstrained iLQR controller manages to track the positional components of the payload's trajectory quite well. The horizontal velocity component V_x in Figure A.1b also manages to remain within the 10 m/s tolerance from the payload for an extended time, while the vertical velocity component proves to be the true driver behind the short rendezvous duration. All configurations in Figure A.1d show this velocity component cutting across the payload's velocity trajectory, with the slope differences between these curves reducing from the baseline, to the climber and ultimately the reeler configuration which manages to remain near zero vertical velocity longer than the other configurations. This "flatter" velocity curve ultimately helps to flatten out the relative velocity curve of Figure A.1f in the vicinity of the payload, thereby extending the rendezvous window.

A.1.2. Control Output

The control effort for the three iLQR controlled configurations are given in Figure A.2. As mentioned in the methodology overview in Chapter 5, the controlled dynamics was constraint free, but control saturation of 100 m/s^2 was applied to the actuator in each point mass.

Figure A.2a shows mostly smooth control curves for the baseline configuration, with brief saturation being reached by both the tether tip and control station's reeling actuators before returning to more modest reeling acceleration values. The climber configuration's control effort in Figure A.2b is noticeably less smooth with extended periods of control saturation for all point masses. The control becomes jagged around the $t = 0$ s mark, which is where the vertical velocity crossing occurs as shown in Figure A.1. These jagged controls could be due to a local, poor linearisation of the underlying dynamics.

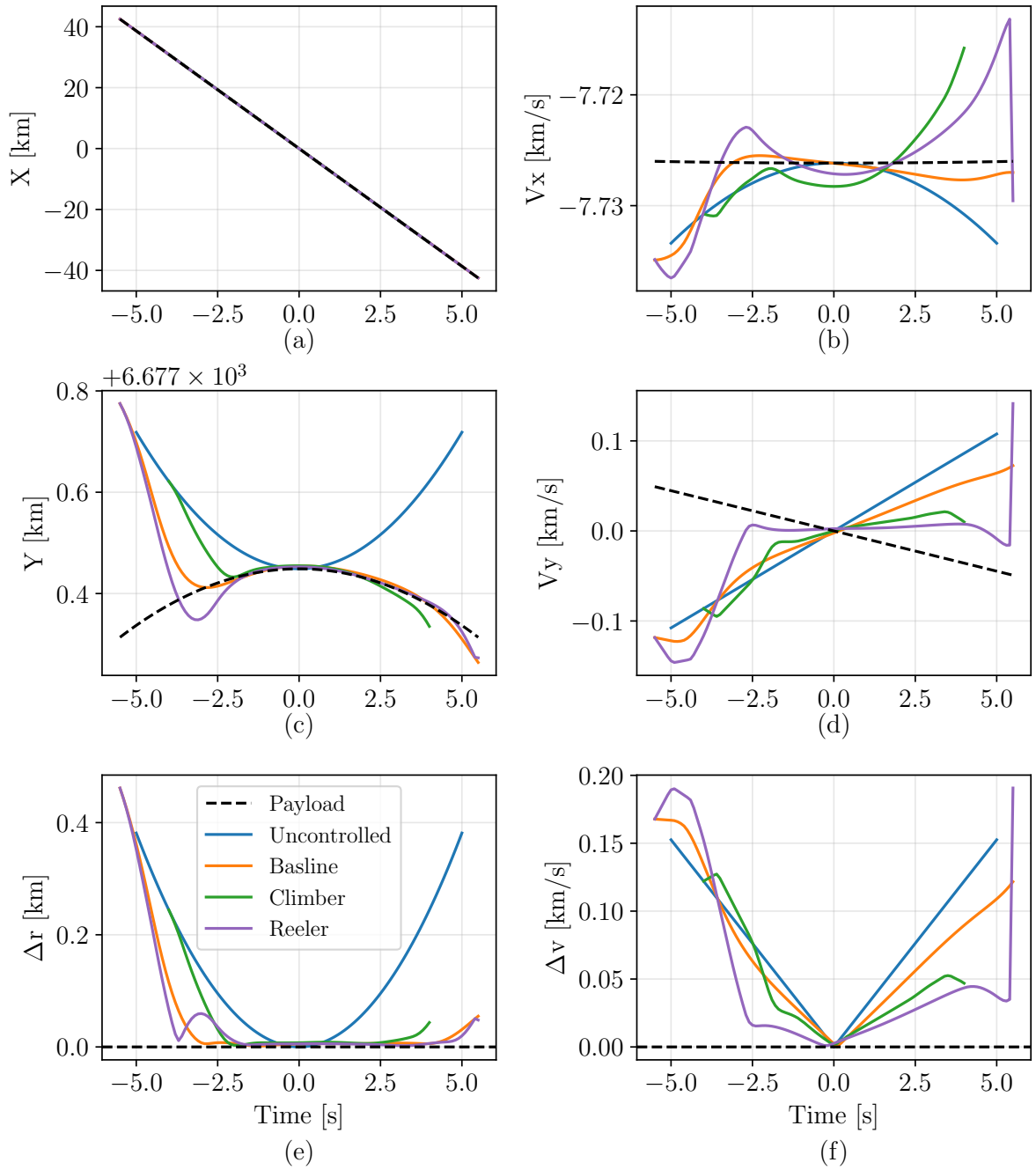


Figure A.1: Comparison of the tip trajectories of the uncontrolled tether against the iLQR controlled baseline, climber and reeler configurations. The subplots (a) to (d) show the x and y position and velocity states for the different tether configurations against that of the payload over time. Plots (e) and (f) show the magnitudes of the relative position and velocity difference vectors between the tether tip and payload respectively.

Though no clear negative effects of this jagged control is evident in the state trajectories of Figure A.1, such behaviour is nevertheless unwanted, and can be addressed with penalties on extreme control rates if needed. Finally, the reeler's control results in Figure A.2c again shows extended regions of control saturation for all point masses, though the control values are much smoother than the climber configuration in the

non-saturated regions.

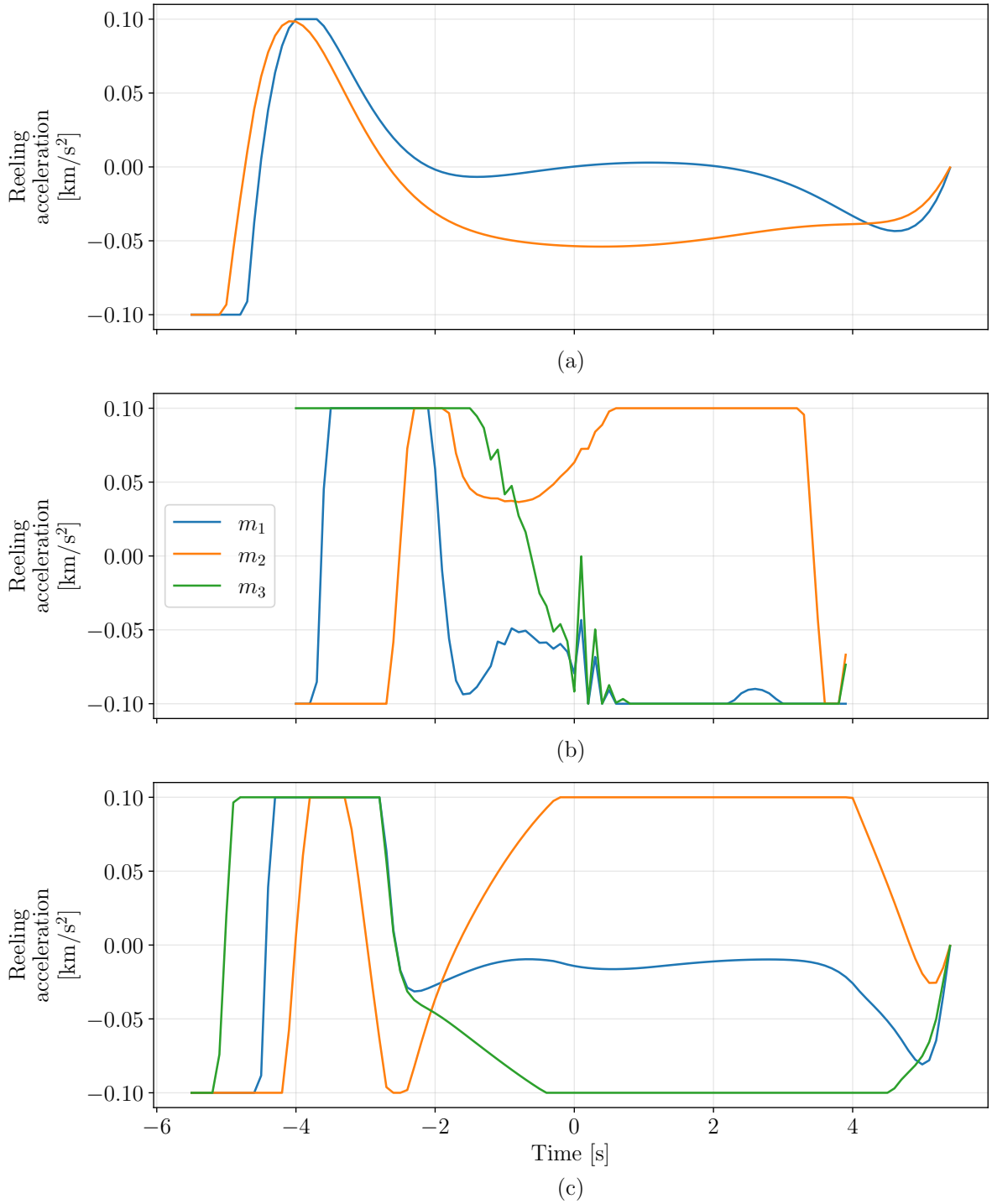


Figure A.2: Reeling acceleration controls for the (a) baseline, (b) climber, and (c) reeler tether system configurations under unconstrained iLQR control, subject to actuator saturation. The legend is consistent across all subplots.

A.2. Unconstrained RL Control

A.2.1. State Trajectories

The reeler configuration's unconstrained, RL controlled results presented in Chapter 5 are extended to include the state components in Figure A.3.

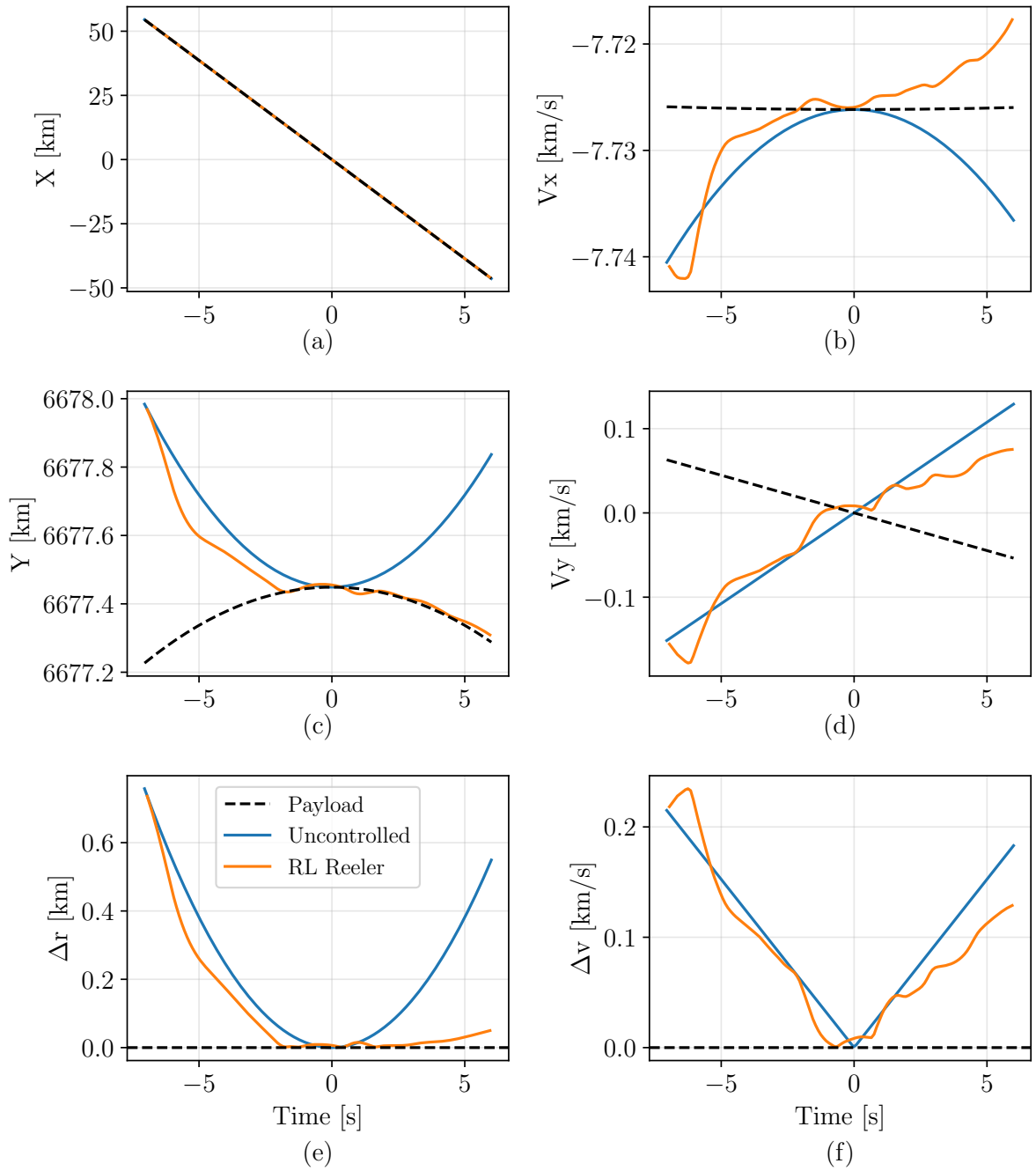


Figure A.3: Comparison of the tip trajectories of the uncontrolled tether against the RL controlled reeler configuration. The subplots (a) to (d) show the x and y position and velocity states for the different tether configurations against that of the payload over time. Plots (e) and (f) show the magnitudes of the relative position and velocity difference vectors between the tether tip and payload respectively.

Figure A.3a and A.3c again show good tracking of the payload's positional trajectory, though the vertical tracking is less smooth that the iLQR counterpart. The velocity curves of Figure A.3b and A.3d also show less smooth state behaviour when compared to the iLQR results in Figure A.1. However, as mentioned in Chapter 5, the RL controller shows an improved ability to locally approximate the payload's velocity, which becomes clear when considering the locally near-zero slope of the vertical velocity component in Figure A.3 around $t = 0$ s. This locally flat curve allows the RL controller to match the rendezvous window of the iLQR case.

A.2.2. Control Output

Chapter 5 mentioned the sporadic nature of the unconstrained RL control based on the non-smoothness of the state trajectory curves in Figure A.3. This becomes even clearer when considering tbe control effort directly as depicted in Figure A.4. Here it is seen that the actuators in all three point masses cycle between highly positive and highly negative controls, exerting much more control effort than the iLQR case shown in Figure A.2c for the same rendezvous window end result. This is undesired as the repeated and excessive control effort increases the power requirements from the limited on-board power supply. Furthermore, such actuator cycling can contribute to increased system wear and reduce the overall lifespan of the tether system.

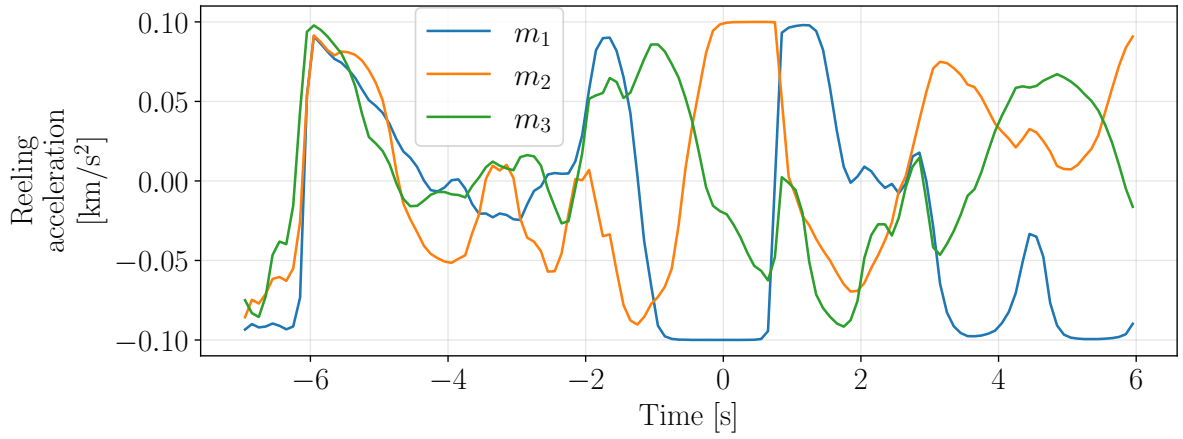


Figure A.4: Reeling acceleration controls for the reeler configuration under unconstrained RL control.

A.3. Constrained AL-iLQR and RL Control

A.3.1. State Trajectories

The reeler configuration's constrained, AL-iLQR and RL controlled results presented in Chapter 5 are extended to include the state components in Figure A.5.

As discussed in Chapter 5, neither the AL-iLQR nor the RL controller were able to steer the reeler tether system toward a sustained rendezvous window. Of the two controllers, the AL-iLQR controller finds a trajectory that satisfies the constraints by effectively staying close to the unconstrained trajectory. It manages to remain close to the pay-

load's velocity trajectory around $t = 0$ as shown by Figure A.5b, d and f. The AL-iLQR controller overshoots the payload's position by a significant margin in the vertical direction before completing the upward swing-through due to the tether rotation, as shown in Figure A.5c, and e. The RL controller's resulting velocity trajectory is similar to that of the uncontrolled and AL-iLQR cases in Figure A.5d, and f, but completely misses the payload's position in Figure A.5c, and e.

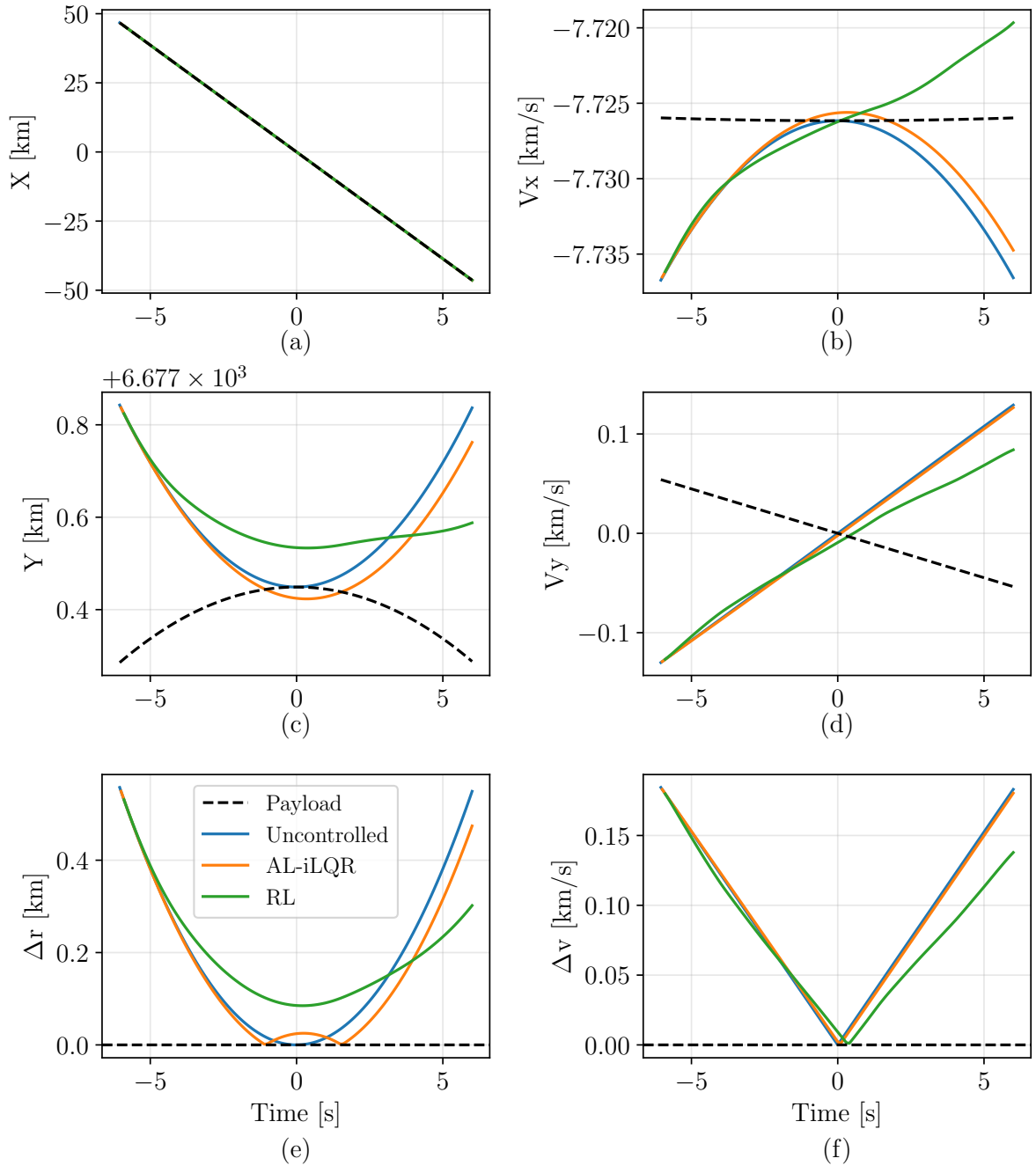


Figure A.5: Comparison of the tip trajectories of the uncontrolled tether against the constrained AL-iLQR and RL control results of the reeler configuration. The subplots (a) to (d) show the x and y position and velocity states for the different tether configurations against that of the payload over time. Plots (e) and (f) show the magnitudes of the relative position and velocity difference vectors between the tether tip and payload respectively.

A.3.2. Control Output

The control output for the constrained AL-iLQR and RL controllers are depicted in Figure A.6a and b, respectively. Both cases show low control values for the whole simulation duration, indicating the general preference of inaction over constraint violation for both controllers. The RL control result is again noticeably less smooth when compared to the AL-iLQR case.

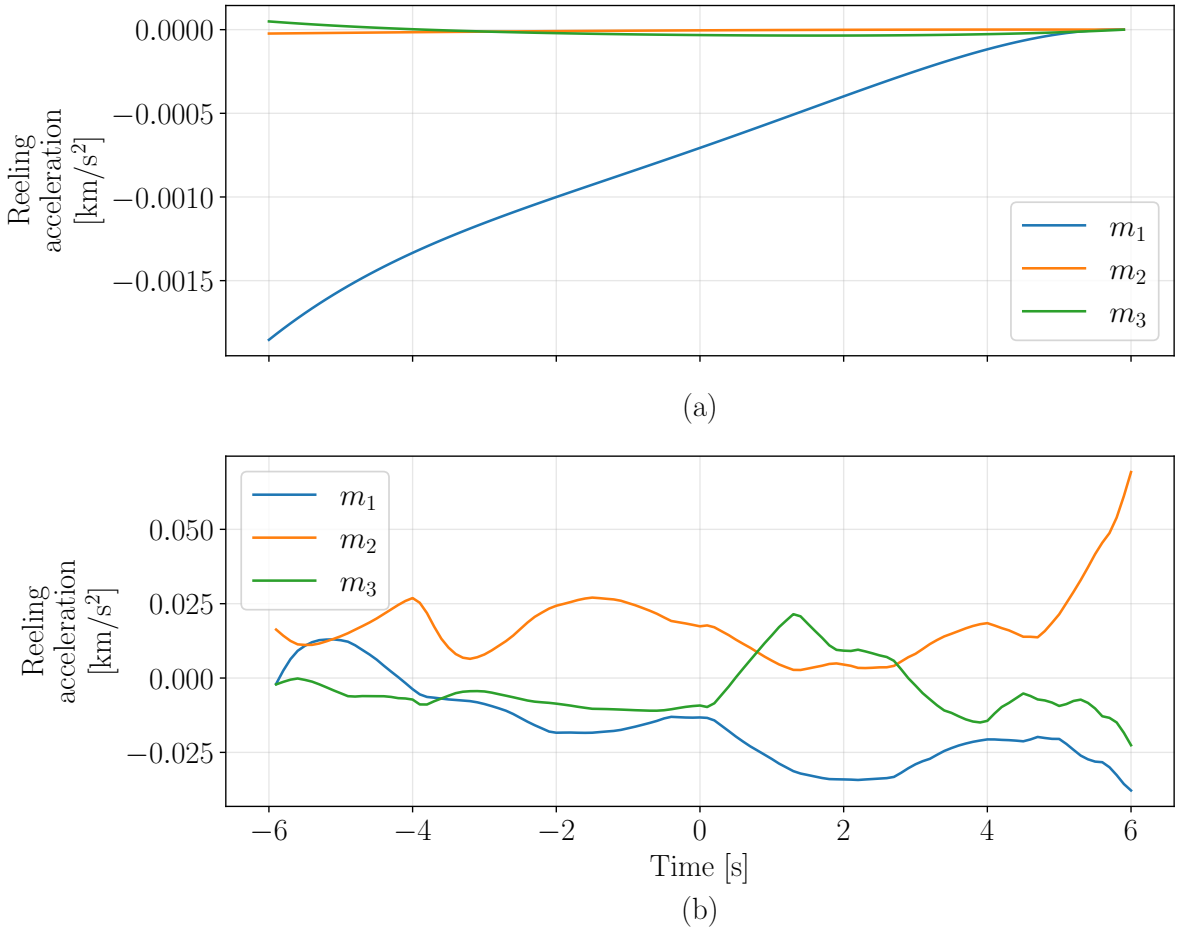


Figure A.6: Reeling acceleration controls for the reeler configuration under constrained (a) AL-iLQR, and (b) RL control.

A.3.3. Constraint Behaviour

The normalised constraint behaviour for the AL-iLQR and RL controllers are presented in Figure A.7 and A.8 respectively, with the maximum values for each constraint listed in Table A.1 and Table A.2. The specific constraint formulations are described in Chapter 5. There it is also mentioned that these constraints are used in the normalised negative-null form, which means values less than or equal to zero indicate constraint satisfaction, while values above zero are indicative of constraint violations. Constraint values which are exactly or extremely close to zero, are considered active, and directly impacts and limits the (near) optimal trajectory.

Constraints for AL-iLQR control

Consulting Figure A.7 and Table A.1 shows that the AL-iLQR controller found a feasible trajectory that respects all constraints. This trajectory is however, as discussed in earlier in this section and in Chapter 5, overly conservative, and does not reach a sustained rendezvous window. It is clear from Table A.1 that no constraints are active, with most values being comfortably below zero, and some being significantly less than zero which means some trajectory improvement is left unrealised.

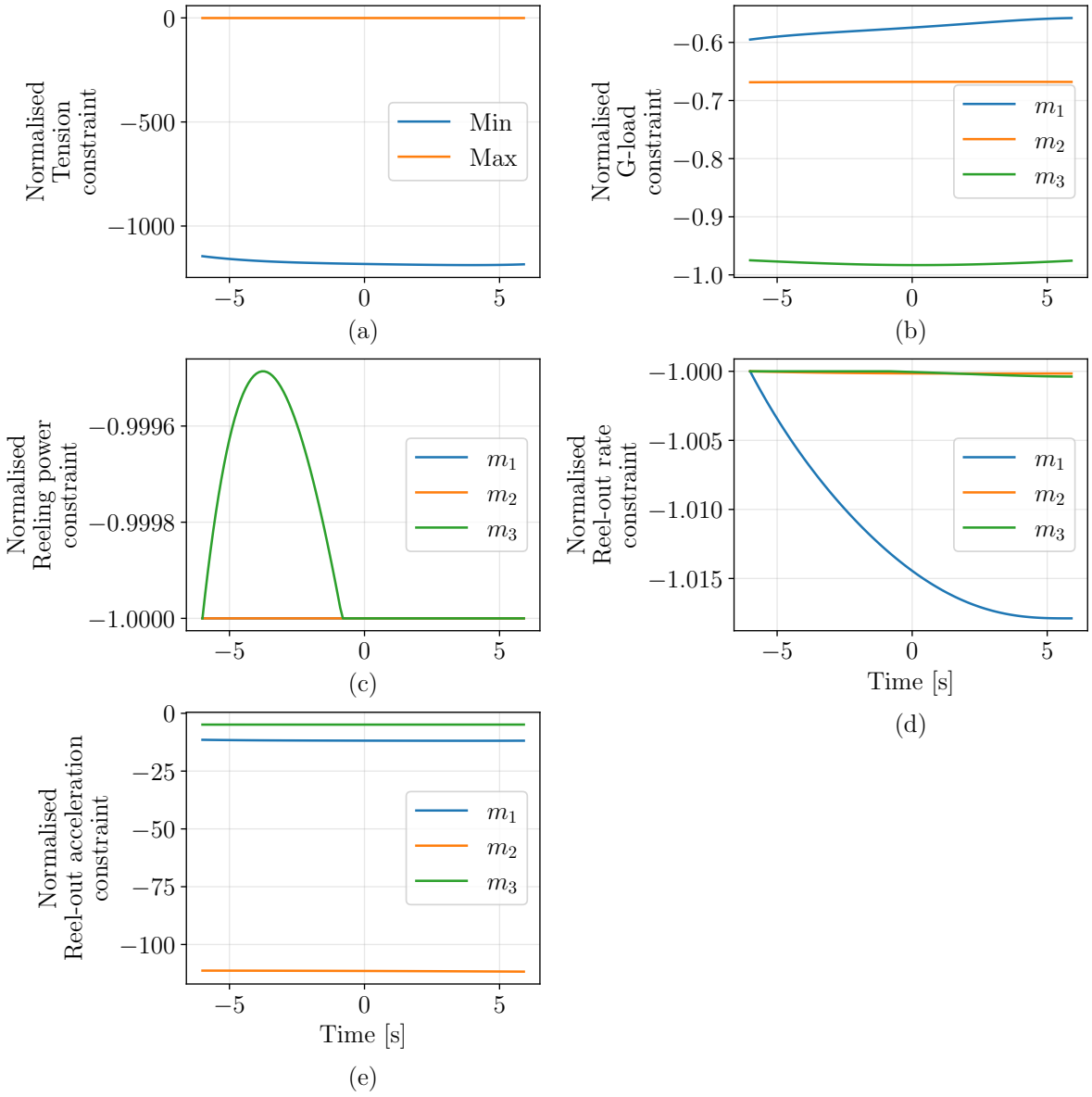


Figure A.7: Normalised constraints for AL-iLQR control in negative null form. (a) Minimum and maximum tether tension, (b) point mass g-loads relative to COM, (c) point mass reeling power, (d) point mass reel-out rates, and (e) the point mass reel-out accelerations.

Table A.1: Maximum constraint values and their associated point masses for the AL-iLQR control

Normalised Constraint	Maximum Value	Point Mass
Maximum tension	-0.7502	–
Minimum tension	-1145	–
G-load	-0.5581	m_1
Reeling power	-0.9994	m_3
Reel-out rate	-1.0	m_1, m_2, m_3
Reel-out acceleration	-4.879	m_3

Constraints for RL control

Unlike the AL-iLQR case, Figure A.8 and Table A.2 show that all constraints except the tension constraints are satisfied. Both the minimum and maximum tension constraints are violated under the RL control, with the maximum tension reaching nearly 10% more than the allowable limit value, and the minimum tension saturating at 1, which indicates a slack tether in one or more of the tether segments. These conditions can independently lead to catastrophic failure of the tether system, either through breaking the tether in the former case, or resulting in uncontrollable motion or sudden tension spikes in the for the latter.

Table A.2: Maximum constraint values and their associated point masses for the RL control

Normalised Constraint	Maximum Value	Point Mass
Maximum tension	0.0888	–
Minimum tension	1.0	–
G-load	-0.0379	m_3
Reeling power	-0.2168	m_2
Reel-out rate	-1.0	m_1, m_2, m_3
Reel-out acceleration	-0.8791	m_1

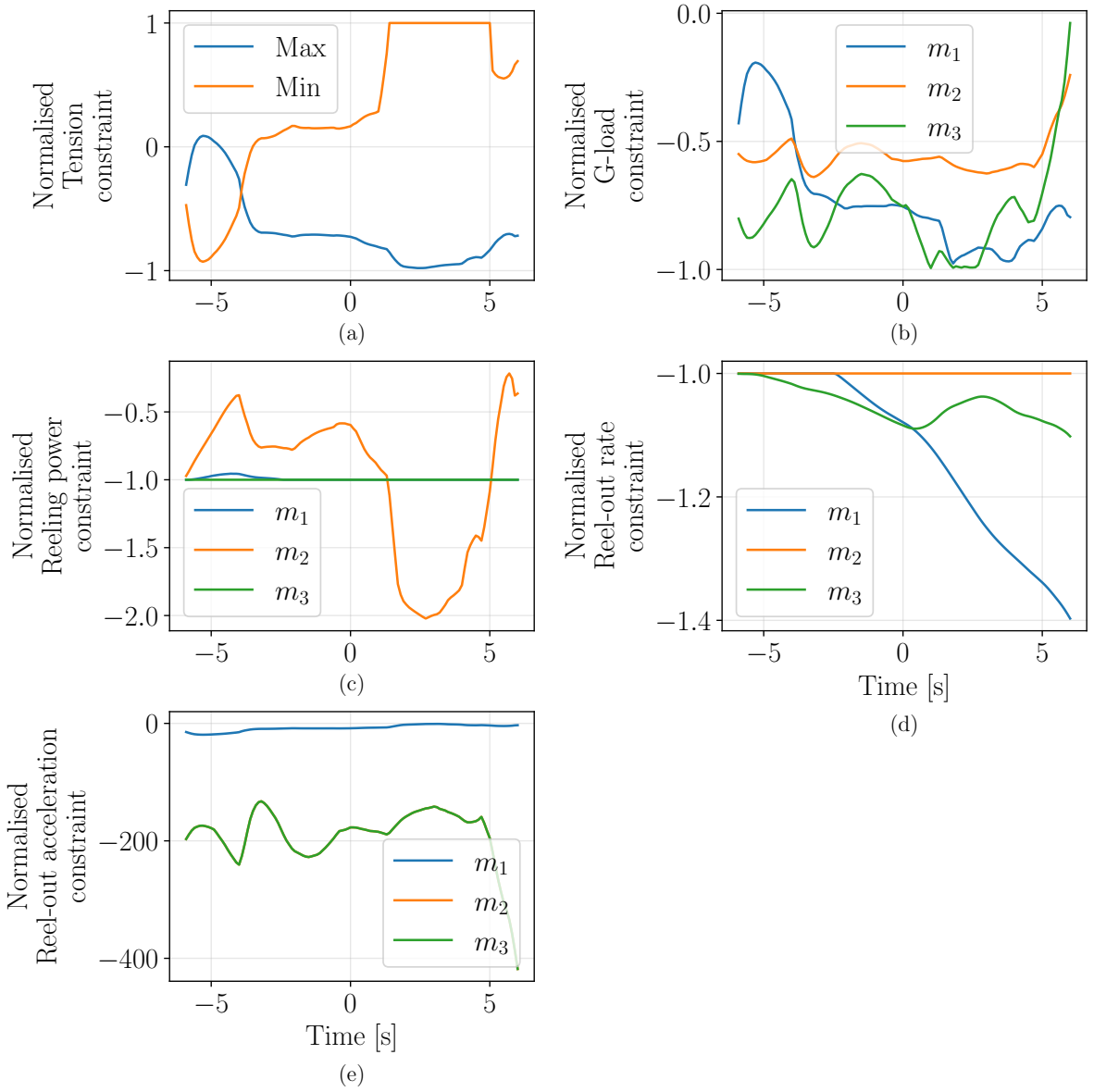


Figure A.8: Normalised constraints for RL control in negative null form. (a) Minimum and maximum tether tension, (b) point mass g-loads relative to COM, (c) point mass reeling power, (d) point mass reel-out rates, and (e) the point mass reel-out accelerations.

B

Planning and WBS

B.1. Work Breakdown Structure

The project work breakdown structure, shown in Figure B.1 was divided into four phases: developing dynamical models, controlling these models with conventional algorithm(s), setting up RL environments, and creating, training, and evaluating RL agents for the unconstrained and constrained cases. Under each phase various tasks and subtasks detail the work completed.

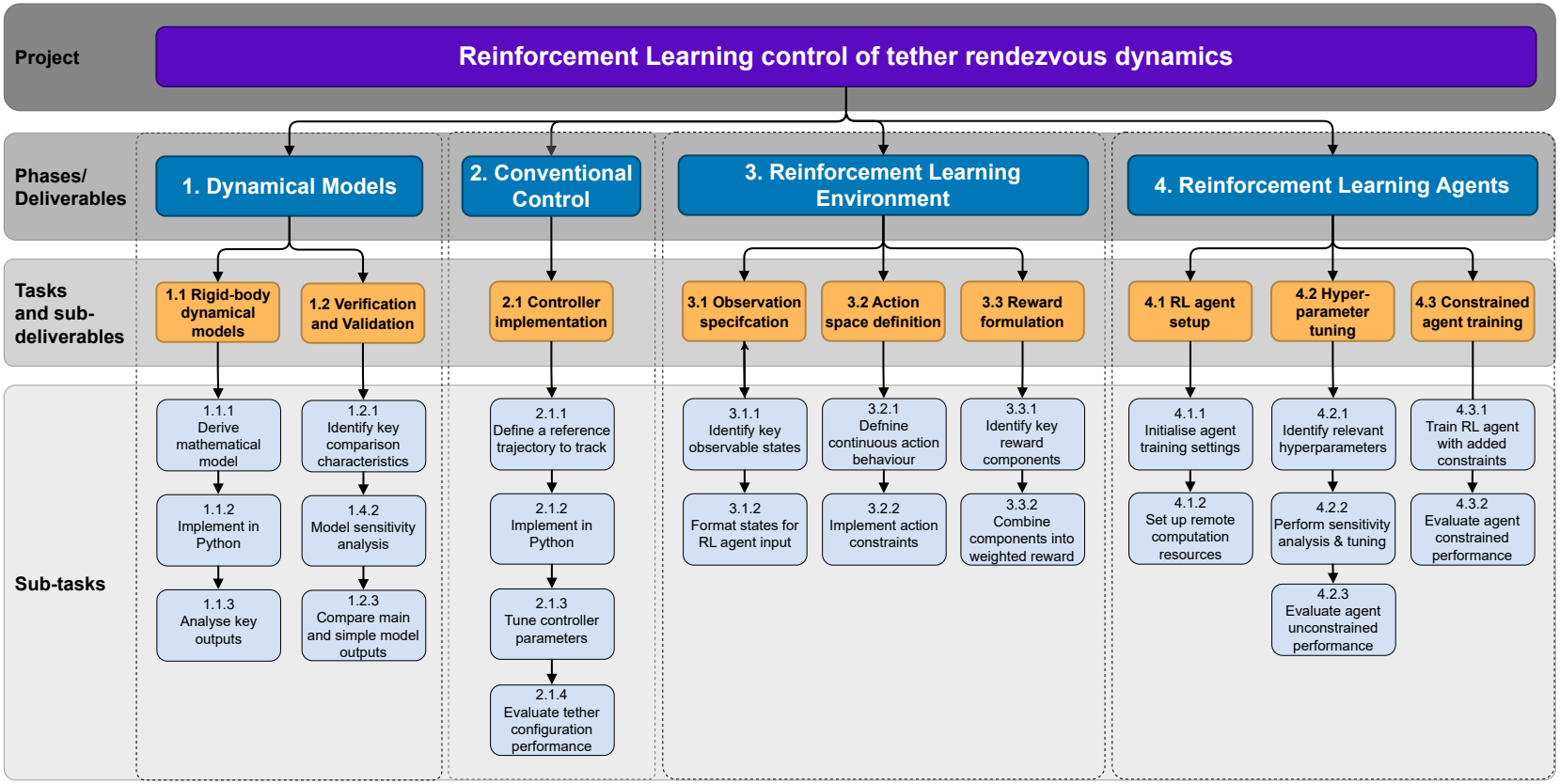


Figure B.1: Proposed work breakdown structure

B.2. Project and Phase Gantt Charts

A rough timeline of the overall project is given in Figure B.2 of Section B.2, with more detailed timelines for each project phase in Figure B.3 to B.6. Each of the phase durations were divided into nominal and buffer times, where the allocated buffer times are proportional to the overall phase duration. The buffer times served as contingency plans in the event that one or more subtasks of a project phase required more time than anticipated.

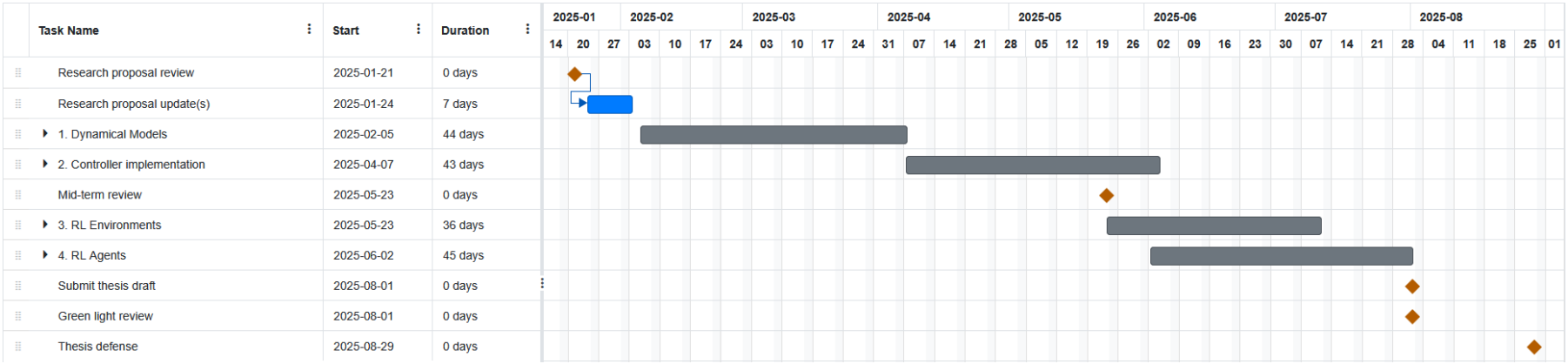


Figure B.2: Timeline overview including major milestones and main project phases.

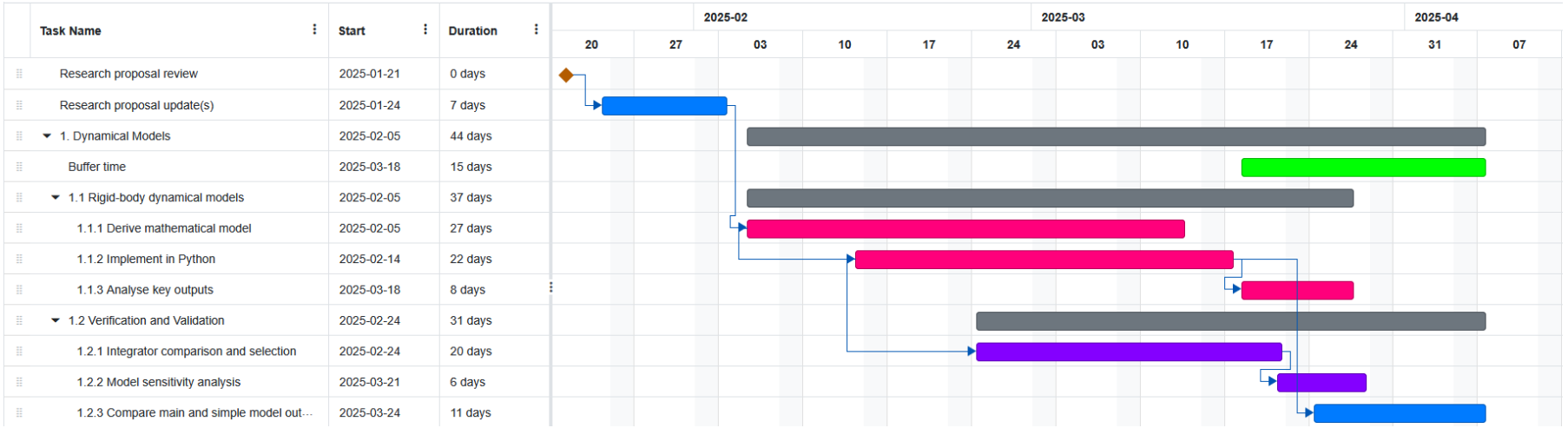


Figure B.3: Timeline overview of Phase 1 - Dynamic models.

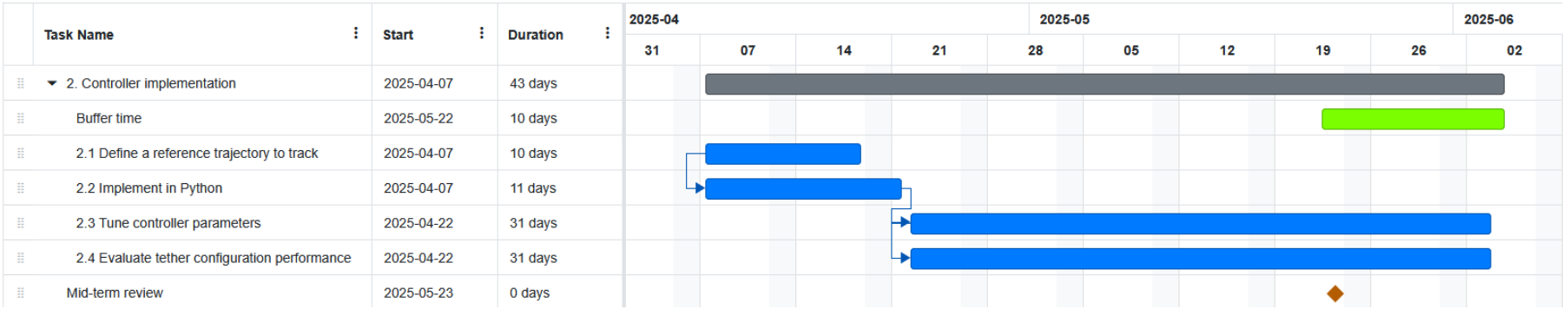


Figure B.4: Timeline overview of Phase 2 - Conventional Control.

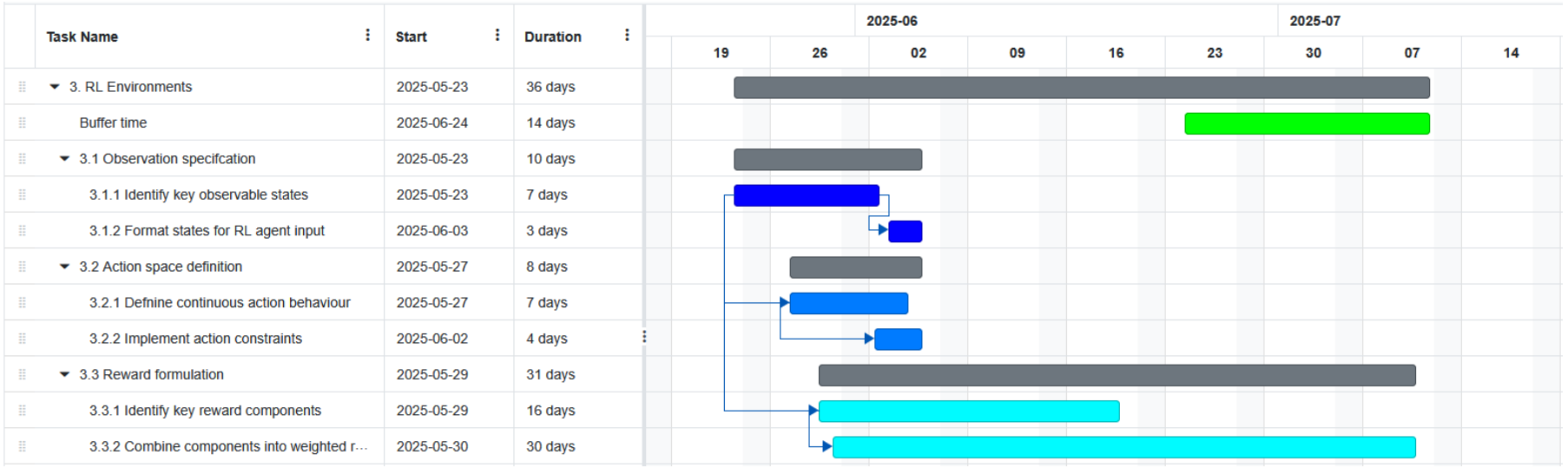


Figure B.5: Timeline overview of Phase 3 - RL Environments.

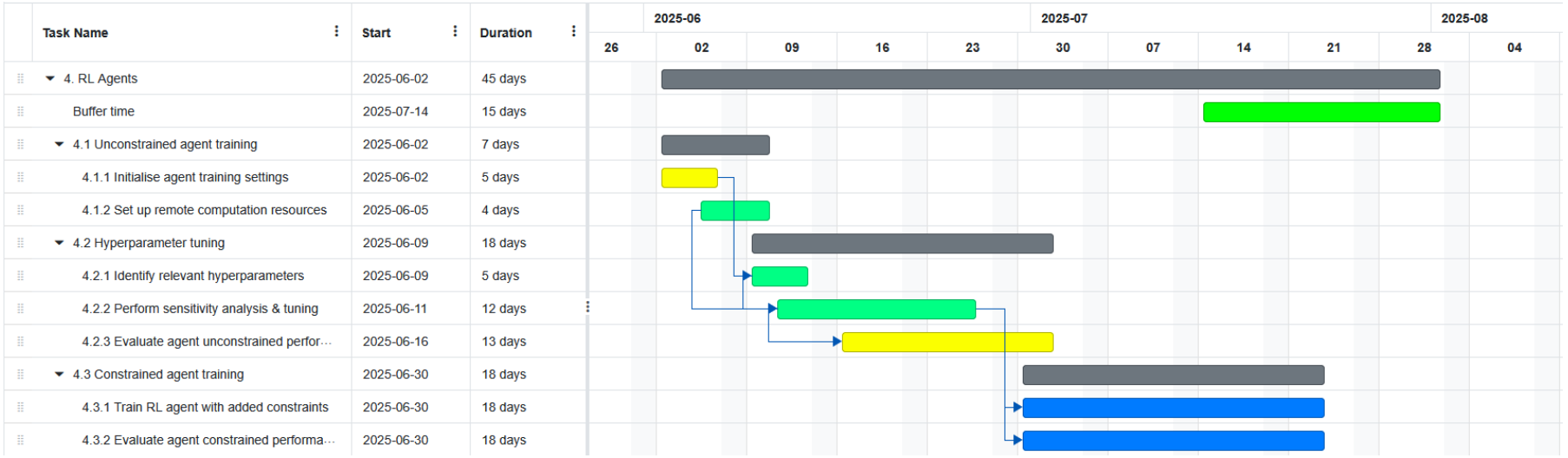


Figure B.6: Timeline overview of Phase 4 - RL Agents.