UNIVERSITAT POLITÈCNICA DE VALÈNCIA

School of Informatics

Development and testing of an embedded control system for the levitation of a Hyperloop vehicle using Reinforcement Learning.

End of Degree Project

Bachelor's Degree in Data Science

AUTHOR: Albert Bonet, Hugo

Tutor: Onaindia de la Rivaherrera, Eva

Cotutor: Aso Mollar, Ángel

Experimental director: GARCIA BOHIGUES, MIGUEL

ACADEMIC YEAR: 2023/2024

Acknowledgements

First, I would like to thank Dr. Eva Onaindía, Ángel Aso, and Miguel García for their help during the process and for introducing me to the world of investigation. Without their knowledge, kindness, and interest in innovation, this project would not have been possible.

Thanks also to the whole team of Hyperloop UPV for their efforts to bring Vèsper to reality, for supporting me as a second family, and for an unforgettable year. Especially, thanks to Álvaro Pérez and Stefan Costea for being the best company during this trip.

Apart from the technical aspects of the project, I would like to thank Aitana Palacios for her unconditional support not only during this project but the whole degree. I cannot and do not want to imagine where I would be if it were not for you.

Lastly, thanks to my family for believing in me, for understanding the relevance of the project, and for taking care of me after long working days. I love all of you.

Resum

Hyperloop és el ja anomenat 'transport del futur', un nou mitjà de transport que utilitza la combinació de levitació i buit per evitar la fricció durant tot el trajecte. Aquest aspecte el fa més ràpid, sostenible i eficient. Hyperloop UPV proposa un sistema de levitació basat en la unió d'imants permanents i electroimants que minimitza el consum energètic si es compta amb un control precís. El present Treball de Fi de Grau desenvolupa un sistema de control de la levitació del vehicle hyperloop que l'equip de Hyperloop UPV presentarà a la competició de l'European Hyperloop Week, mitjançant aprenentatge per reforç. Per a aconseguir-ho, es programa un simulador de la dinàmica del moviment del vehicle i l'entorn d'entrenament de l'agent. Posteriorment, s'entrena l'agent per a aquesta tasca i s'analitza la qualitat de les seues accions i el seu rendiment en un sistema empotrat que utilitza un microcontrolador STM32H723ZGT6.

Paraules clau: Intel·ligència Artificial, Aprenentatge per Reforç, Sistema de Control, Levitació, Hyperloop, Xarxes Neuronals, Sistemes Empotrats

Resumen

Hyperloop es el denominado "transporte del futuro", un nuevo medio de transporte que emplea la combinación de levitación y vacío para evitar el rozamiento en su trayecto, lo que lo convierte en un medio más rápido, sostenible y eficiente. Hyperloop UPV propone un sistema de levitación basado en la unión de imanes permanentes y electroimanes que minimiza el consumo de corriente si se cuenta con un sistema de control preciso. El presente Trabajo de Fin de Grado desarrolla un sistema de control de la levitación del prototipo de hyperloop que el equipo de Hyperloop UPV presentará a la competición de la European Hyperloop Week, mediante aprendizaje por refuerzo. Para ello, se programa un simulador de la dinámica de movimiento del vehículo y el entorno de entrenamiento del agente. Posteriormente, se entrena al agente para esta tarea y se analizan tanto la calidad de sus acciones como su rendimiento en un sistema empotrado que utiliza un microcontrolador STM32H723ZGT6.

Palabras clave: Inteligencia Artificial, Aprendizaje por Refuerzo, Sistema de Control, Levitación, Hyperloop, Redes Neuronales, Sistemas empotrados

Abstract

Hyperloop is called the "transport of the future", a new means of transportation that uses a combination of levitation and vacuum to avoid friction along its path, making it a faster, more sustainable, and efficient mode of transport. Hyperloop UPV proposes a levitation system based on the combination of permanent magnets and electromagnets that minimizes current consumption if it is managed with a precise control system. This paper develops a levitation control system for the Hyperloop prototype that the Hyperloop UPV team will present at the European Hyperloop Week competition, using reinforcement learning. For this purpose, a simulator of the vehicle's motion dynamics and the agent's training environment are programmed. Subsequently, the agent is trained for this task, and both the quality of its actions and its performance in an embedded system using an STM32H723ZGT6 microcontroller are analyzed.

Key words: Artificial Intelligence, Reinforcement Learning, Control System, Levitation, Hyperloop, Neural Networks, Embedded Systems

Contents

Appendix

List of Figures

List of Tables

List of algorithms

CHAPTER 1
Introduction

This chapter serves as an introduction to the work described in the subsequent chapters. Firstly, the motivation of the project is described in Section 1.1. Section 1.2 lists the objectives of the project. Then, Section 1.3 briefly introduces the structure of the documents and the topics related to each of its parts. Finally, Section 1.4 links the content of the project with the generic student outcomes highly valued in the university.

1.1 Motivation

This project is rooted in the work developed as part of the Hyperloop UPV team and more specifically, as its Team Captain. Hyperloop UPV is a team of 49 students from different degrees of Universitat Politècnica de València (UPV) with the aim of developing the technology of the transport of the future, hyperloop.

Hyperloop consists of a mode of transportation that combines magnetic levitation with vacuum infrastructures to eliminate friction, thus reaching incredibly high speeds with a small amount of energy spent.

The work developed at Hyperloop UPV brings two main aspects to the table. On the one hand, creating the transport of the future provides each member with the possibility to transform the world. Hyperloop UPV brings the opportunity to work on a faster and more sustainable means of transport and to approach it to all kinds of audiences. On the other hand, Hyperloop UPV as part of the initiative of *Generación Espontánea* provides its members with an environment where they can develop soft and hard skills while working on a real-world problem, living the experience of a real engineering and multidisciplinary project. The aspect that differentiates Hyperloop UPV from other groups under the same initiative, is the development of something that is not invented yet. This aspect obliges and offers the opportunity to think outside of the box and create new solutions to unsolved problems with state-of-the-art technologies, such as linear synchronous motors, supercapacitors, vacuum infrastructures, and, in the context of this Final Degree Project, levitation.

Besides, a substantial part of this work has been carried out under a collaboration fellowship funded by the Valencian Research Artificial Intelligence Institute (VRAIN). During the 8-month grant, I had the opportunity to work jointly with a group of researchers who actively investigate Reinforcement Learning (RL), a learning technology that was deemed suitable to address the levitation of a hyperloop vehicle.

Reinforcement Learning is an Artificial Intelligence (AI) technique not taught in the bachelor degrees of the UPV and just shallowly explained in some master degrees. Therefore, the possibility of working with a group of experimented investigators on this topic highly enriched my knowledge and this project.

Considering my three years of experience inside the team of Hyperloop UPV, having started as a member the first year levitation was implemented, I have been able to perceive the importance of levitation control in terms of efficiency and performance and to detect issues in the current implementation. Once delved into RL techniques, it was clear that they could be a suitable solution for said issues. Therefore, the project aims to develop a neural network model trained with RL to control the levitation of the new vehicle of Hyperloop UPV and to evaluate this approach as a solution in terms of performance and computational complexity, as the model is run inside an embedded system.

1.1.1. Generación Espontánea

Generación Espontánea (GE) is the paradigm in which Hyperloop UPV exists. GE is an initiative from the UPV where students from different degrees can form groups with social, artistic, or engineering objectives. In this paradigm of projects, the UPV provides economic and material resources, technical advice, and a space to work. However, the biggest teams like Hyperloop UPV have become almost completely independent, minimizing the required technical advice and counting on more than 90 sponsors.

1.2 Objectives

The main objectives intended to be fulfilled in the project are:

- Investigate the dynamic and physical functioning of the vehicle and create a simulator of its behavior.
- Train a model able to control the vertical levitation of one coil.
- Train a model able to control the levitation of two coils.
- Train a model able to control the levitation of the complete system, with 10 coils.
- Optimize the model in terms of computational complexity.
- Evaluate the performance of the model in its dedicated embedded system.

1.3 Organization of the document

The following section provides a brief explanation of the content of the different parts of the document.

1. Introduction

This initial chapter includes the motivation of the project, the different objectives to be accomplished, and the impact of the environment of this thesis on the generic student outcomes or soft skills.

2. Related Work

- **Hyperloop**: Provides sufficient context about the concept of hyperloop to understand the document. The concept is explained in detail, also informing about the origin of the idea, which emerged earlier than is commonly thought, and the current state of the technology.
- **Hyperloop UPV**: Defines what Hyperloop UPV is, details the trajectory of the team, and illustrates how it is organized internally. An explanation of how the international competition is developed and its relevance to hyperloop technology is also included.
- **Vèsper**: Details the functioning of the new vehicle of Hyperloop UPV, focusing on the systems of major relevance for this project.

3. Control System with Reinforcement Learning

This section provides a technical context of the project, explaining what Reinforcement Learning (RL) is, why it should be applied to the Control System of Vèsper, and which requirements are imposed on the Control System by external factors. It also explains the methodology to be followed and the different concepts regarding the design and training of the RL system.

4. One degree of freedom

The first step for the design of the Control System is to control a system with just one Degree Of Freedom (DOF), consisting of a single levitation unit able to get closer or farther from the steel plate. This 1 DOF control can be tested on a custom test bench to ensure the functioning of all systems involved in the levitation of Vèsper.

5. Two degrees of freedom

The 2 DOF control is developed as an intermediate point between the 1 DOF control and the 5 DOF control. This algorithm is not tested in real-life applications but serves as a testing point to ensure that the dynamic model of the vehicle is being correctly scaled.

6. Five degrees of freedom

The 5 DOF control is the model that will be running during the competition to allow Vèsper to levitate. It is the final stage of the development of the system and the most complex one.

7. Conclusions

This chapter states the conclusions of the project and its functioning, the degree of achievement of the objectives, and the linkage with the contents of the Data Science Degree.

8. Future Work

The last chapter of the document proposes further improvements to achieve better results, and the next steps to continue with the project.

1.4 Generic student outcomes

Apart from technical knowledge, the UPV highlights the relevance of generic student outcomes or soft skills in the professional development of every student. In the context of Hyperloop UPV, both technical skills and generic student outcomes are key aspects of daily activities. This section explains the linkage of each of the differentiated generic student outcomes with the current project.

- **Social and environmental commitment**: The concept of hyperloop itself has important social and environmental benefits. Hyperloop is created to interconnect cities and cultures, but mainly to enhance sustainability by eliminating friction. In this aspect, levitation control has a crucial role in bringing efficiency to the system.

- **Innovation and creativity**: Levitation is an innovative solution for sustainable and fast mobility. Moreover, applying Reinforcement Learning to levitation control has never been done before.

- **Teamwork and leadership**: This project has been developed in the context of two groups working together, where I have been the link between them. On the one hand, the group of investigators of VRAIN provided the necessary technical advice about RL. On the other hand, the team of Hyperloop UPV where I am the leader and where teamwork is a must. This project accepts the requirements of other departments of the team to create a common system that is more powerful than the sum of its parts.

- **Effective communication**: The proper fact of writing a memoir of the project and presenting it on a tribunal enhances this generic student outcome. Moreover, the exchange of information between VRAIN and Hyperloop UPV has been a key aspect of the success of the project.

- **Responsability and decision-making**: The methodology followed, the changes to make at each stage of the process, and the experiments developed along the project reflect this aspect. Moreover, the use of a new technique that is not taught in the bachelor's degree in Data Science enhances this outcome.

CHAPTER 2

Related Work

The following chapter provides context about the environment of the project, offering insights into hyperloop and related technologies. Additionally, it shares information about the Hyperloop UPV team, their duties, and their new vehicle to ensure a comprehensive understanding of the context of the project.

2.1 Hyperloop

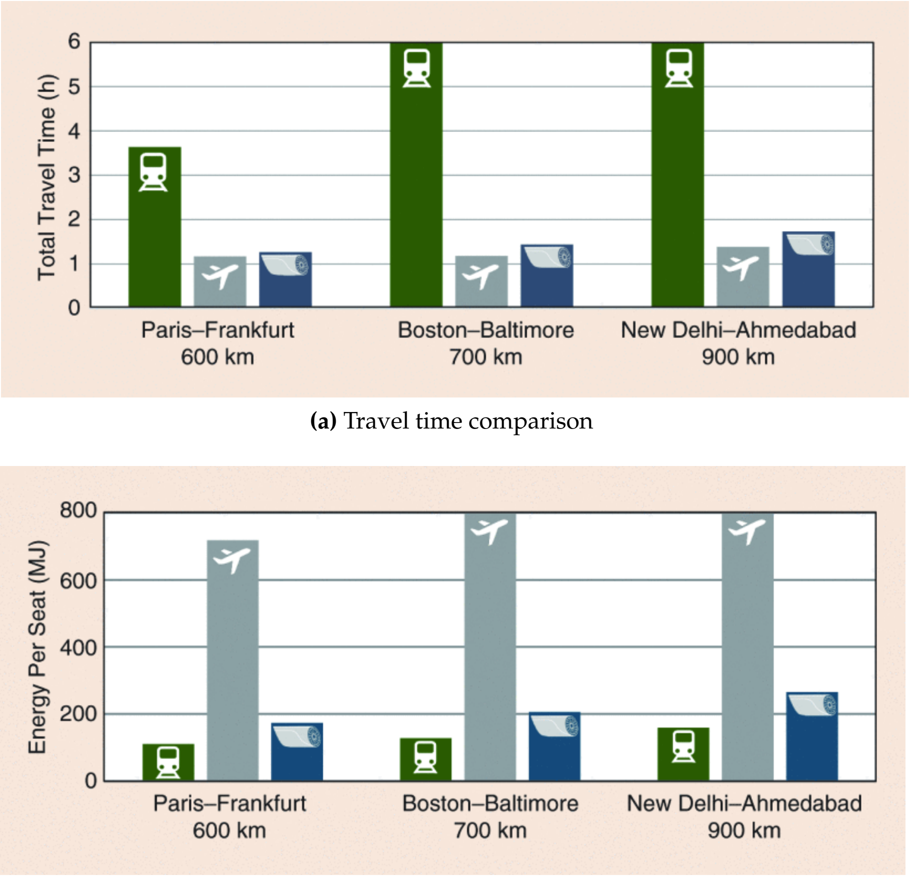
In this section, hyperloop technology will be explained to provide sufficient understanding to comprehend the thesis. We will delve into the concept of hyperloop and the aspects that make it different from other means of transport, the origins of this technology, and the current state of hyperloop both nationally and globally.

2.1.1. Concept

Hyperloop, also referred to as the transport of the future or the fifth means of transport, consists of a series of capsules, similar to the train, which achieve great speeds by spending small amounts of energy. Based on levitation and vacuum as its core technology, hyperloop technology aims to achieve both passenger and cargo interurban transportation in a faster and more sustainable way.

Regarding levitation, hyperloop relies on electromagnetic forces to avoid friction with the surface, while near-vacuum environments significantly reduce losses related to air friction. The combination of those technologies not only allows hyperloop to achieve speeds of around 1000 km/h but also make it with high efficiency. The efficiency provided by the lack of friction together with the use of electric sources of energy make hyperloop a significantly sustainable mode of transportation.

Currently, the main competitors of hyperloop are high-speed trains and planes. On the one hand, high-speed trains are significantly slower, as shown in Figure 2.1a. The Spanish AVE travels from Valencia to Madrid at a maximum speed of 231 km/h [1] and the German ICE 3 travels at a maximum speed of 300 km/h [2]. On the other hand, although planes achieve similar speeds to hyperloop, they produce high amounts of greenhouse gasses and their energy consumption per seat is not comparable, as depicted in Figure 2.1b. Moreover, airlines count on external factors that produce time losses apart from the journey itself.



(a) Travel time comparison

(b) Energy consumption comparison

Figure 2.1: Comparison in terms of speed and energy consumption among the train, the plane, and hyperloop

Source: Evacuated-Tube, High-Speed, Autonomous Maglev (Hyperloop) Transport System for Long-Distance Travel: An overview [4]

Transrapid Maglev is a closer alternative. It consists of a high-speed train that incorporates levitation. No hyperloop standards exist in terms of levitation technology, which makes it difficult to compare the efficiency between Maglev and hyperloop regarding said aspect. However, Maglev technology does not eliminate friction with the air, which limits its velocity to a maximum of 431 km/h [3].

2.1.2. Origin

Despite the futuristic look of this technology, the concept of hyperloop was born more than 200 years ago. The English mechanical engineer George Medhurst described in 1799 a mode of transportation involving tunnels at different pressure levels, which were constructed during the 19th century and known as “atmospheric railways” [4].

Later, in 1904, Robert Goddard came up with the idea of magnetic levitation in his publication “The Limit of Rapid Transit”. Both magnetic levitation and vacuum tunnels were developed in parallel, the first one resulting in maglev trains once combined with the linear induction motor and the second one in inventions like the Large Hadron Collider. It was not until the 1970s that levitation and vacuum were combined in a single concept with the proposal of Swissmetro [6]. Swissmetro was conceived as a subway that was able to levitate inside a 5-meter wide tunnel with an environmental pressure of

100 mbar. Although it would have achieved a theoretical speed of 500 km/h, it was never implemented.

2.1.3. Current state

In the current century, the development of more advanced railways and the increasing impact of mobility on the environment emphasized the need for innovation in the field of transportation. For that purpose, Elon Musk published in 2013 a whitepaper called "Hyperloop Alpha", with a proposal of a hyperloop concept including air bearings for levitation and linear motors for traction. Nevertheless, instead of patenting this concept, he left it open source, which resulted in the creation of companies such as Hyperloop Transportation Technologies or Virgin Hyperloop –later called Hyperloop One–.

Although Elon Musk did not further develop this concept, his company SpaceX created the Hyperloop Pod Competition, a student competition for universities around the world to motivate the development of innovative ideas. The Hyperloop Pod Competition led to the creation of successful groups of students such as the one from the Technical University of Munchen, which years later accomplished a passenger test as a research group, or Hyperloop UPV, from Universitat Politècnica de València, which created a spin-off called Zeleros.

In 2021, after a reflection period caused by the pandemic of 2020, Hyperloop UPV, together with Swissloop from ETH Zurich, Delft Hyperloop from TU Delft, and Hyped from the University of Edinburgh, created the European Hyperloop Week (EHW). The EHW revolutionized the idea of competition, introducing the participation of companies from the field of hyperloop and other technological fields, and the divulgation to the general public. At the moment, the EHW has become the most relevant hyperloop competition around the globe.

2.2 Hyperloop UPV

Hyperloop UPV is a multidisciplinary team linked to the program *Generación Espontánea* from Universitat Politècnica de València (UPV) whose purpose is to develop innovative hyperloop technology. More than 40 students form the team and each year they develop completely functional vehicles to validate new technologies, demonstrate their applicability to the concept of hyperloop, and divulge this technology to the general public.

Its team members embody student representation from all faculties of the UPV, demonstrating the wide range of fields committed to innovation and depicting the inherent multidisciplinarity of hyperloop as a revolution in the world of mobility. Apart from students from different engineering –Mechanical Engineering, Electrical Engineering, Computer Science. . . –, business, and artistic degrees, the team counts on two Faculty Advisors; Tomás Baviera, PhD in Journalism, and Vicente Dolz, PhD in Mechanical Engineering. This combination allows a close experience to entrepreneurship and real-world problem solving, giving the opportunity to not only improve on hard skills but also generic student outcomes.

2.2.1. History

As mentioned in the previous section, Hyperloop UPV was born thanks to the competition organized by SpaceX. In 2015, five students from the UPV teamed up to participate in the SpaceX Design Weekend –a subcompetition inside the Hyperloop Pod Competition–

that took place in Texas in January 2016. This generation of students is internally referred to as H1, the first generation of the team, and they received the awards for the Best Concept Design and the Best Propulsion System.

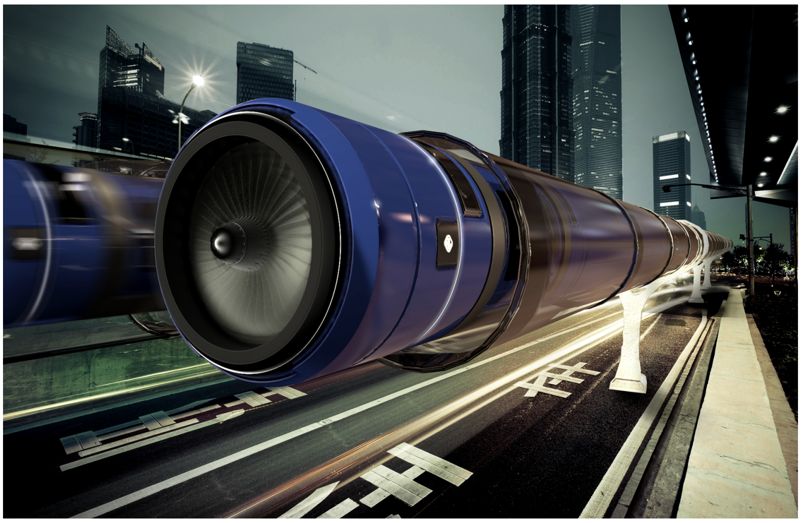
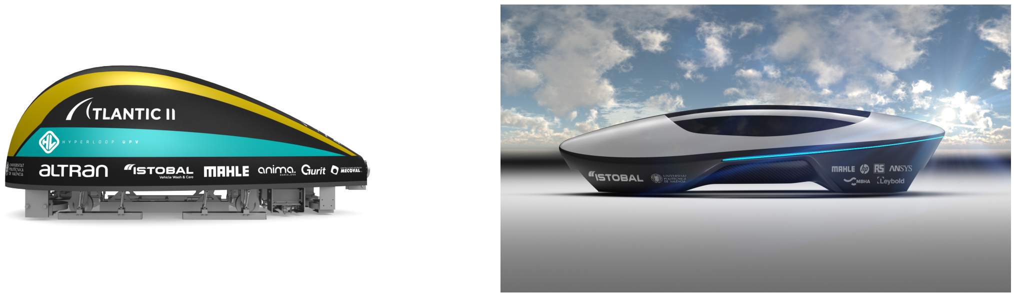


Figure 2.2: The design of Hyperloop UPV for the SpaceX Design Weekend

Source: Hyperloop UPV (2016)

After such successful participation, Hyperloop UPV was officially created. The team continued to grow and participate in the Hyperloop Pod Competition in the following years, settling as one of the top 8 best teams in the world. In the next two years they created Atlantic II and Valentia, shown in Figure 2.3.



(a) Atlantic II

(b) Valentia

Figure 2.3: Atlantic II and Valentia

Source: Hyperloop UPV (2017 & 2018)

In the last year of the Hyperloop Pod Competition, Hyperloop UPV developed a significantly improved vehicle, Turian. Turian was the most compact one, containing more than 400 sensors and achieving a maximum speed of 470 km/h. The prototype also incorporated a highly aerodydnamic carbon fiber fairing. All the aforementioned aspects made Turian win the Innovation Award in the competition of 2019, becoming the 8th-best prototype out of more than 700 competitors.

Figure 2.4: Turian

Source: Hyperloop UPV (2019)

During the season of 2019/2020, the pandemic impeded the development of a new prototype. Instead, and without certainty about the future of the competition, Hyperloop UPV decided to improve Turian as much as the safety measures allowed.

This year served as a reflection period. The team deeply thought about their goals as hyperloop developers and about the final objective of the Hyperloop Pod Competition.

The sixth generation of Hyperloop UPV aimed to reorient the student paradigm around hyperloop. This generation brought to life the European Hyperloop Week with Delft Hyperloop, Hyped, and Swissloop. The EHW awarded scalability and real-world implementation above speed or compactness without an application to real hyperloops.

In the next two years, Hyperloop UPV developed highly scalable prototypes, culminating with Kénos and Atlas. Kénos, the vehicle, was able to levitate without friction. Atlas was the name of the tube, which for the first time in the competition provided a vacuum environment. Including electronics and sensors, the infrastructure achieved sufficient relevance to be named, as the vehicle.

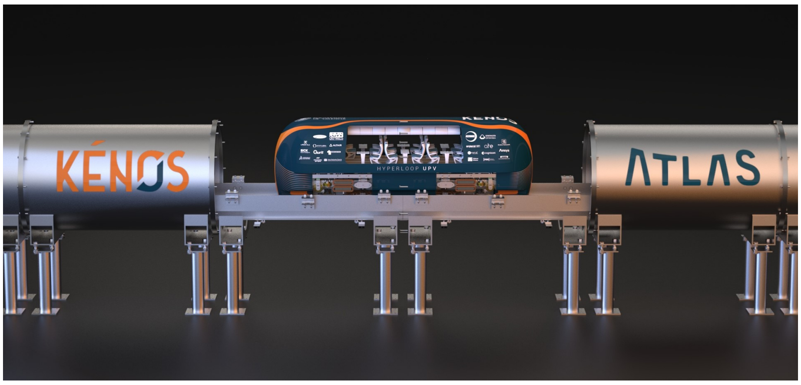


Figure 2.5: Kénos & Atlas

Source: Hyperloop UPV (2023)

2.2.2. Organization of the team

The team of Hyperloop UPV develops all the different fields involving hyperloop technology. To achieve that, the members are divided into working groups called subsystems. Each subsystem is led by a Project Manager, who is focused on the mid-term objectives of the project. All Project Managers are supervised and guided by three Technical Directors in charge of the long-term picture. Of the three Technical Directors, one is selected as the Team Captain, who is the main reference and the representative of the team.

The subsystems of the team are explained hereunder:

- **Electromagnetics** involves the design of all electromagnetic systems, including the levitation units, the transformers of the chargers, and the motors. Electromagnetics also develops the control system of each of them.

- **Structures & Mechanisms** develops all the mechanical systems of the vehicle and infrastructures, including the chassis, the brakes, and the test benches.

- **Hardware** designs and validates the electronic boards of the vehicle and infrastructure.

- **Firmware** programs all the electronic boards designed by Hardware to make them able to function properly and communicate with the control station.

- **Software** designs and programs the control station of the vehicle, allowing the team to send orders and receive data.

- **Partners** is in charge of searching for sponsors for the team, bringing the necessary resources for the project to be successful.

- **Outreach** works on the external image of the team, which includes graphic design, social media, and the corporate image of Hyperloop UPV.

- **Economics** manages the economic resources provided by the UPV and brought by Partners to achieve an optimized cost distribution.

2.2.3. European Hyperloop Week

As was mentioned before, Hyperloop UPV was consolidated with its participation in the Hyperloop Pod Competition hosted by SpaceX in Los Angeles. For 4 years, the team took part in this competition alongside other university teams from Europe and the United States.

However, at the end of 2019, Hyperloop UPV chose to embark on a huge challenge with 3 of the best European universities (University of Edinburgh, Delft University of Technology, and Federal Institute of Technology Zurich): creating a new hyperloop event to share the most innovative technology and to create a community driven by the vision of turning hyperloop into a reality.

That is how the European Hyperloop Week was born. It can be defined as an event that goes beyond mere competition since it is sustained by the insatiable curiosity of very talented students and their eagerness to learn. During a week, leading companies in the tech industry and university teams from all over the world share the work and development that they carried out regarding hyperloop. The main goal is to foster innovation through collaboration, as it is stated in the mission of the EHW.

Figure 2.6: Hyperloop UPV during the first edition of the EHW, developed in Valencia

Source: Las Provincias (2021)

2.3 Vèsper

The edition of the EHW 2024 will take place in Zurich, Switzerland. This year, Hyperloop UPV introduces Vèsper, which coupled with Atlas and a revolutionary booster motor aims to create not just a prototype, not only a vehicle but a complete and scalable hyperloop transportation system.

Vèsper weighs 246.95 kg. Its external dimensions –including the aeroshell– are 2.4 m in length, 0.69 m in width and 0.63 m in height. All the systems that integrate the vehicle can be seen in the exploded view of Figure 2.7.

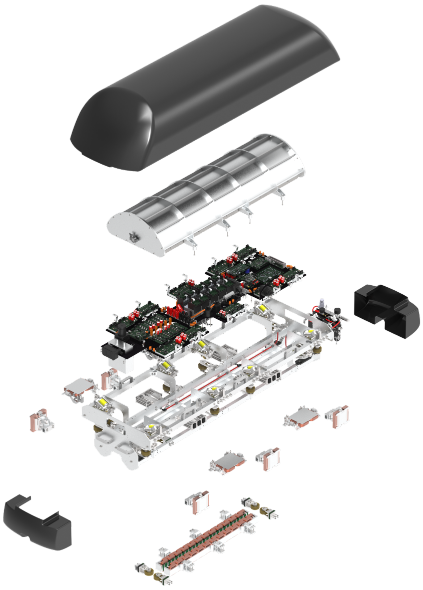


Figure 2.7: Exploded view of Vèsper

Source: Hyperloop UPV (2024)

As the team states, "This year, Vèsper is set to not only awaken the world to the immense potential of hyperloop technology but also to illuminate the correct path forward, guiding us even through the darkest moments of the innovation process".

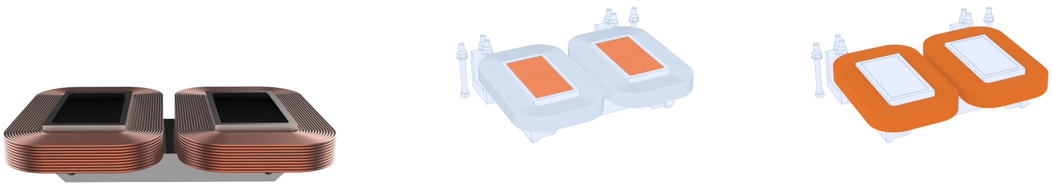
2.3.1. Levitation of Vèsper

The Levitation & Guiding System is one of the most relevant systems of Vèsper, as it allows the vehicle to avoid friction with the surface, thus needing significantly smaller amounts of force to accelerate.

The Levitation & Guiding System is composed of four Hybrid ElectroMagnetic Suspension units (HEMS) –a combination of an electromagnet with a permanent magnet– and four ElectroMagnetic Suspension units (EMS) –just the electromagnet–. The former ones are designed to handle vertical levitation while the latter ones are optimized to control horizontal levitation.

HEMS

The HEMS units control the vertical levitation of Vèsper, which is the main task while levitating. In fact, vertical and horizontal levitation are often referred to as "levitation" and "guiding" respectively. Figure 2.8 shows an HEMS unit designed by Hyperloop UPV.



(a) HEMS unit

(b) Location of the permanent magnets

(c) Location of the coils

Figure 2.8: HEMS unit designed by Hyperloop UPV

Source: Hyperloop UPV (2024)

Both coils and magnets are joined to a laminated electrical steel core, whose purpose is to drive magnetic flux in the correct direction.

The principle of the HEMS units is based on using the magnetic forces produced by permanent magnets to counter the constant force of gravity. As the position where those forces are equal is unstable, electromagnets control perturbances to maintain the vehicle in the correct position. Figure 2.9 depicts the functioning of the HEMS units.

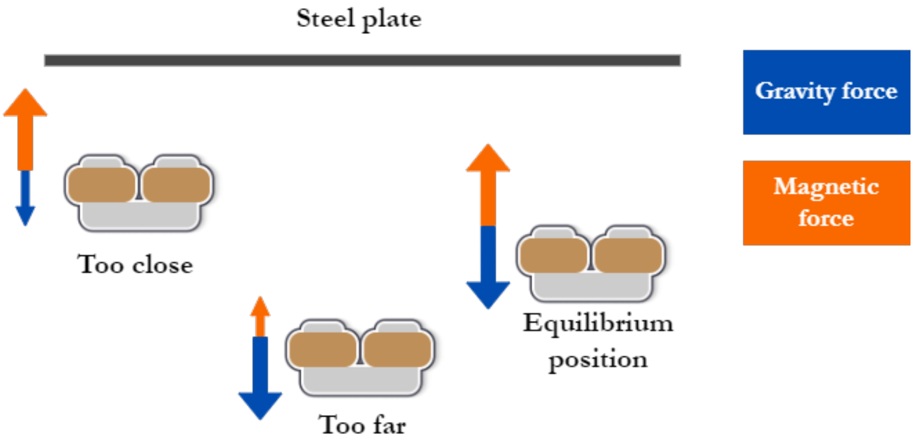


Figure 2.9: Functioning of the HEMS units

Using HEMS units instead of other technologies –e.g., superconductors– brings various advantages such as minimizing energy consumption when a proper control system is incorporated, ease of manufacturing, and cost-effectiveness.

EMS

Horizontal levitation units, or EMS units, follow the same basic principle. Nevertheless, no permanent magnets are needed as no constant force is being applied to the vehicle horizontally. Thus, EMS units consist uniquely of electromagnets to correct horizontal deviations to keep the vehicle in the middle of the infrastructure. Figure 2.10 shows an EMS unit designed by Hyperloop UPV.

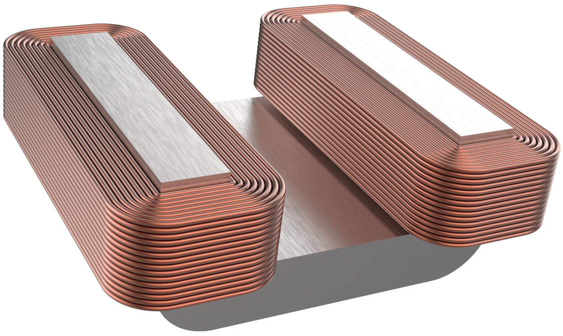


Figure 2.10: EMS unit designed by Hyperloop UPV

Source: Hyperloop UPV (2024)

2.3.2. Electronic System of Vèsper

Apart from the electromagnetic design of the levitation units, the Levitation & Guiding System relies on a complicated electronic circuit. This circuit includes the Levitation Control Unit (LCU) and 10 Levitation Power Units (LPU).

The LPU is the board in charge of providing the units with the requested amount of amperage. However, the board is not capable of providing a concrete value of current, but a value of voltage which increases or decreases the current flowing through the coil.

The LCU is in charge of executing the control algorithm for the levitation system. For that purpose, the Firmware and Hardware subsystems decided to include two microcontrollers STM32H723ZGT6 which allow the board to make the appropriate calculations, and to communicate with other boards.

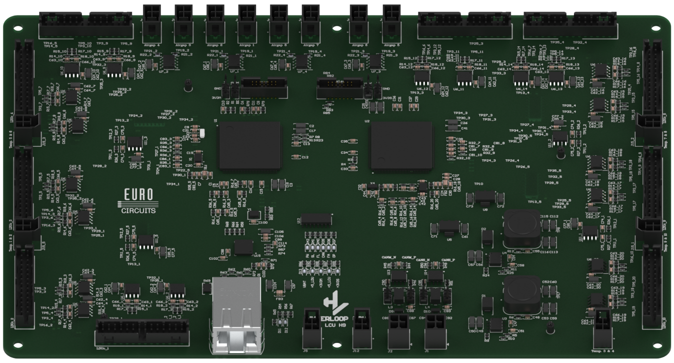


Figure 2.11: Levitation Control Unit

Source: Hyperloop UPV (2024)

Microcontroller (STM32H723ZGT6)

The selected Micro Controller Unit (MCU) is the STM32H723ZGT6, able to operate at up to 480MHz and with a dual-bank Flash memory of up to 2 Mbytes, and 1 Mbyte of RAM.

As an STMicroelectronics MCU, the STM32H723ZGT6 needs to be programmed through the official STM32 Cube IDE and configured through the interface of STM32 Cube MX. This aspect makes it necessary to use C or C++ as the programming language for the MCU. However, the lack of long-term memory inside the board makes it necessary to translate the control algorithm directly to code for the board.

X-CUBE-AI

Luckily, the tool STM32 X-CUBE-AI allows translating from Tensorflow or ONNX Artificial Intelligence (AI) models to C code to be executed directly inside the MCU. In fact, this tool provides the necessary methods to profile and optimize the algorithm.

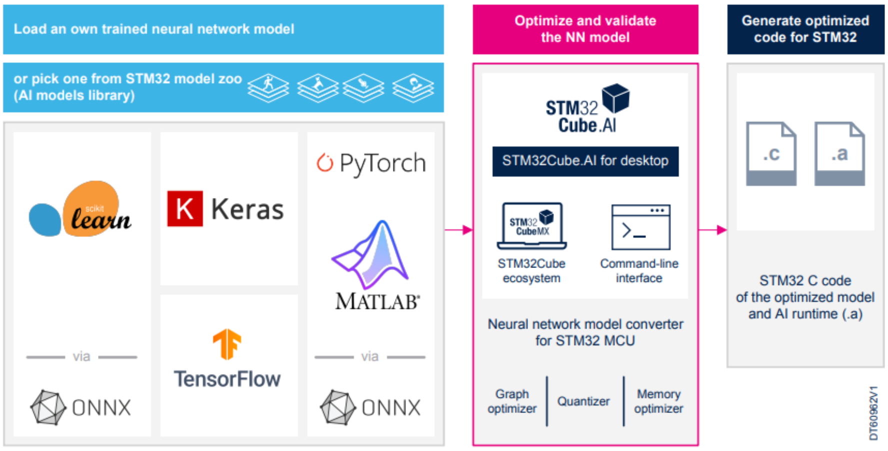

Load an own trained neural network model
or pick one from STM32 model zoo (AI models library)
scikit learn
via
ONNX
Keras
TensorFlow
PyTorch
MATLAB
via
ONNX
Optimize and validate the NN model
STM32 Cube.AI
STM32Cube.AI for desktop
STM32 CubeMX
STM32Cube ecosystem
Command-line interface
Neural network model converter for STM32 MCU
Graph optimizer
Quantizer
Memory optimizer
Generate optimized code for STM32
.c
.a
STM32 C code of the optimized model and AI runtime (.a)
DT60962V1

Figure 2.12: Artificial intelligence (AI) software expansion for STM32Cube

Source: Data brief - X-CUBE-AI - Artificial intelligence (AI) software expansion for STM32Cube [8]

CHAPTER 3
Control System with Reinforcement Learning

This chapter revolves around the design of the Control System for the levitation of Vèsper and the use of Deep Reinforcement Learning techniques to control the vehicle. The chapter is organized as follows. Firstly, we present a brief overview of Control System and Reinforcement Learning notions. Section 3.2 outlines the external requirements imposed on the Control System and Section 3.3 justifies the selected proposal, describing the benefits compared to the current one. Finally, the last section explains the methodology, the different pieces of the creation of the system, and the steps followed to develop it.

3.1 Introduction

A control system is, according to the National Institute of Standards and Technology (NIST), "a system in which deliberate guidance or manipulation is used to achieve a prescribed value for a variable" [9]. Translated to the specific Control System for the levitation of Vèsper, it consists of an algorithm to stabilize the vehicle at the equilibrium height despite the natural perturbations of the dynamics of Vèsper.

An advanced control system is a key aspect of the levitation of the vehicle, as the efficiency regarding energy consumption directly depends on the optimal use of the electromagnets. The optimal control system should spend the minimum amount of current to control the perturbations of the vehicle, keeping the current levels around zero.

A PID control –and all the controls from this family such as the PI or the PD– is a classical controller widely used in the industry. It includes three actions: proportional (P), integral (I), and derivative (D).

- **P**: Outputs a value proportional to the current error; $u_P(t) = K_P \cdot e(t)$
- **I**: Returns a value proportional to the cumulative error, which makes the control slower; $u_I(t) = K_I \int_0^t e(\tau)\, d\tau$
- **D**: Predicts future error accelerating the control; $u_D(t) = K_D T_D \frac{de(t)}{dt}$, where T_D is a value called *derivative time*.

The final equation for a PID is:

$$u(t) = K_P \cdot e(t) + K_I \int_0^t e(\tau)\, d\tau \; + K_D T_D \frac{de(t)}{dt} \tag{3.1}$$

The control system used for Kénos, the previous vehicle, consisted of a cascade control for each coil divided into a PID controller to set the objective current given the position of the vehicle at each time step and a PI controller to set the target voltage given the objective current set by the previous controller.

In this document, a Deep Reinforcement Learning agent is proposed as a substitute for the previous control system. The necessary tools developed for the training process are explained and the performance of the agent is discussed and evaluated in the document.

3.1.1. Reinforcement Learning

Understanding the definition and functioning of Reinforcement Learning (RL) is a key aspect to achieving a complete comprehension of this thesis.

RL is about figuring out how to act in different situations to maximize a reward, without being directly told which actions to take. Learners must experiment to find the most rewarding actions. Actions can influence not only immediate rewards but also future situations and rewards. This trial-and-error learning and the impact of delayed rewards are key features distinguishing reinforcement learning [10].

Reinforcement Learning is different from *supervised learning* as the latter is learning from a training set and trying to generalize for examples not present in it, while the former one is learning from experience and understanding the interaction with the environment. RL is also different from *unsupervised learning* as, although it is not learning from a labeled dataset, it is not trying to find any hidden structure inside the data but maximizing a reward function. Therefore, Reinforcement Learning is considered the third machine learning paradigm.

RL has been applied to numerous games in the past, for example, Tic-Tac-Toe. Nevertheless, with the sharp increase in autonomous systems being developed in recent years, this type of learning has been introduced in much more complex environments such as autonomous driving [11], robotics [12], or even Large Language Models (LLM) [13].

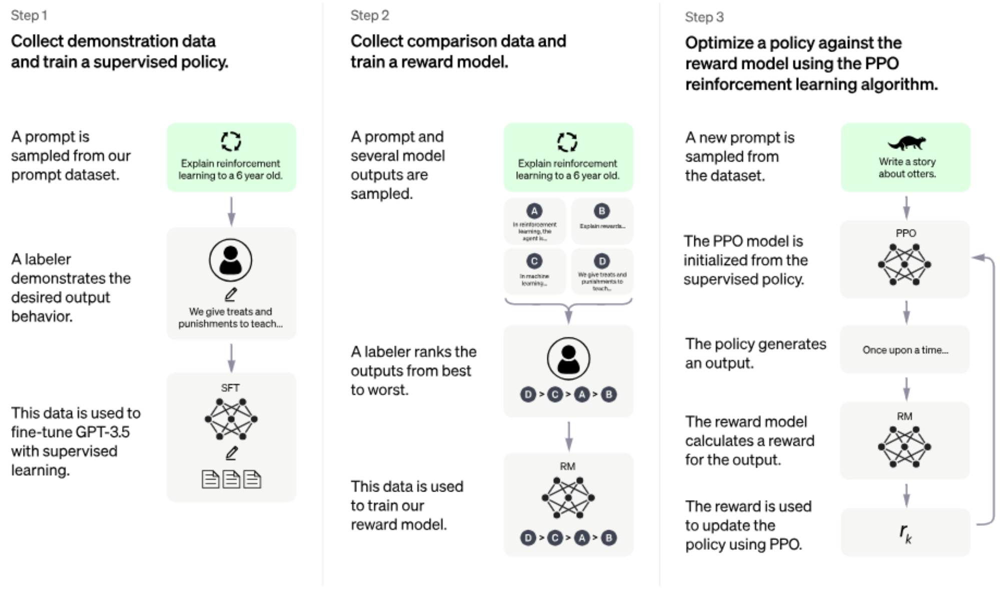

Step 1
Collect demonstration data and train a supervised policy.
A prompt is sampled from our prompt dataset.
Explain reinforcement learning to a 6 year old.
A labeler demonstrates the desired output behavior.
We give treats and punishments to teach...
This data is used to fine-tune GPT-3.5 with supervised learning.
SFT
Step 2
Collect comparison data and train a reward model.
A prompt and several model outputs are sampled.
Explain reinforcement learning to a 6 year old.
A
In reinforcement learning, the agent is...
B
Explain rewards...
C
In machine learning...
D
We give treats and punishments to teach...
A labeler ranks the outputs from best to worst.
D > C > A > B
This data is used to train our reward model.
RM
D > C > A > B
Step 3
Optimize a policy against the reward model using the PPO reinforcement learning algorithm.
A new prompt is sampled from the dataset.
Write a story about otters.
The PPO model is initialized from the supervised policy.
PPO
The policy generates an output.
Once upon a time...
The reward model calculates a reward for the output.
RM
The reward is used to update the policy using PPO.
r_k

Figure 3.1: ChatGPT training process

Source: OpenAI official website [13]

Apart from the concept of Reinforcement Learning, it is important to know its elements; the policy, the reward signal, and the value function.

The **policy** refers to the function used to decide which action to take given a certain state. The policy can be a lookup table, a simple mathematic equation, or even a neural network –Deep Reinforcement Learning–. The policy is the only element used when using the RL model for inference.

The **reward signal** can be thought of as the *treat* or punishment received by the environment after each action. After every decision of the agent, a numerical value called the *reward* is received. The reward is the main cause of change in the policy, as the agent is trying to find the policy that maximizes the received reward.

The **value function** aims to classify the benefit of reaching a certain state. While the reward signal is the immediate value obtained by an action, the value function maps the future value that the agent can obtain if it has already reached a certain state.

Let us set an example to enhance comprehension. Imagine a man is trying to return home to see his family. This person is using *Google Maps* for better orientation. *Google Maps* is the policy because it tells that person which road to take at each intersection. The main reward is the positive feeling of arriving home and seeing his family, but there can be negative rewards such as getting his shoes dirty if he steps in the mud. The value function is the experience of this person once he reaches a place that he knows is closer or further to his house.

Markov Decision Process (MDP)

A problem, in order to be solved using RL techniques, needs to be mathematically formulated as a Markov Decision Process (MDP), a classical formalization of a process of decision-making. In an MDP, actions affect immediate rewards and future ones by influencing the future reachable states. This means that the value of an action is state-dependent. MDPs are idealized mathematical representations of reinforcement learning problems to provide tractability and comprehension of the optimal policy.

In an MDP, an *agent* or decision-maker transitions from a *state* S_t, at timestep t, to another state S_{t+1} by applying an *action* A_t, and receives a *reward* R_t from the *environment*. An MDP does not need the history of all the states and actions previously taken, but only the current state to decide which action to take. The objective of the agent is to maximize the cumulative reward or *expected return*.

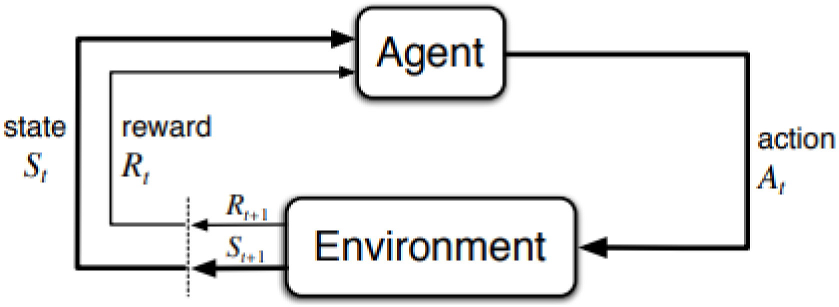


Figure 3.2: The agent–environment interaction in a Markov decision process

Source: Reinforcement learning : an introduction [10]

Exploration vs Exploitation

The trade-off between *exploration* and *exploitation* is a crucial aspect of the RL process. This trade-off is based on the problem that exclusively selecting the best actions discovered at a certain moment can prevent the agent from finding better rewards.

For example, if we always travel to the same place during the summer holidays, we may enjoy our vacations. However, if we sometimes vary our destination, we can have a horrible vacation but we can also find an even more enjoyable place.

In that example, *exploitation* refers to visiting the same place where we know we are going to enjoy. On the other hand, *exploration* refers to leaving our comfort zone, without certainty about if we are going to find a better or worst destination.

In formal terms, *exploitation* refers to looking for the maximum possible expected reward while *exploration* refers to avoiding getting stuck in a local maximum, exploring new areas of the multidimensional space of the reward function.

3.2 Requirements

This section enumerates the external requirements imposed on the Control System:

- The Control System must allow its inference at a frequency of 1 kHz –1000 inferences per second– in the microcontroller, meaning that the asymptotic maximum time consumption of an inference is 1 ms.
- The range of amperage while levitating must be inside the range [-45, 45] A.
- The range of voltage while levitating must be inside the range [-100, 100] V.
- The maximum and minimum distances between the vertical levitation unit and the steel plate are 22.5 mm and 9 mm respectively, as those are the distances where the mechanical guidance touches the infrastructure.
- The maximum and minimum distances between the horizontal levitation unit and the steel plate are 16 mm and 4 mm respectively, as those are the distances where the mechanical guidance touches the infrastructure.

3.3 Justification

The proposal for the control system of Kénos brings several drawbacks:

- Using two controllers per coil duplicates the parameters to tune per coil and highly increases the total amount of parameters. Moreover, it does not allow taking into account the behavior of the rest of the levitation and guiding units.
- The control system needs to be manually adjusted through a trial and error process every time a change is introduced. For example, this is the case when the experiments performed at Valencia need to be repeated at Zurich.
- The process of translating the measurements of the sensors into the exact position of the vehicle, required for the control, is highly computationally expensive as it includes a significant amount of sines and cosines.

Taking the aforementioned drawbacks into account, an RL Control System has the following benefits:

- Deep Neural Networks can obtain better results than PID controllers, as they are more complex, so both PID and PI controllers can be substituted for a single one.
- Deep Neural Networks with fully connected layers provide connections among the outputs of the different coils.
- The RL Control System can be automatically fine-tuned in case the characteristics of the control change, such as the forces produced by the coils, the desired height, or for better adaptation to the real scenario once implemented on the vehicle.
- The RL Control System does not use the position of the vehicle, but the measures of the sensors to decide which actions to take.

3.4 Methodology

This section explains the methodology followed during this project, including the step-by-step process and the different components of it.

The methodology of the current Final Degree Project is based on Incremental Learning. Incremental Learning consists of gradually introducing new information in the model so that it can reach greater goals without forgetting the basic aspects. This approach is based on a teaching strategy, which is also called Incremental Learning, which was then transferred to machine learning. In the case of this project, the complexity of the environment and the actor progressively increases with two main objectives:

- The number of coils in the simulator progressively increases so that perfect functioning of the modelization can be ensured, detecting errors in the earliest stage possible and enhancing comprehension of the dynamics of the model.
- The complexity of the environment gradually increases so that the model first learns the basic control of the vehicle and then it will adapt to more difficult tasks.

Section 3.4.1 describes the progression followed to increase complexity gradually.

3.4.1. Guidelines

The development process of this project is divided into three steps.

First, the simulator of one coil is modelized to accurately mimic the functioning of a levitation unit. A model is trained to control this unit, first learning to act over the desired current and then over the voltage of the coil. This step is named the one degree of freedom (1 DOF) control and consists of learning how to control the vertical movement of the levitation. This degree of freedom, or vertical displacement, is the main purpose of the levitation control, thus being the first to be tested and proven possible to be controlled.

Once the model works properly, the simulator is adapted to take into account two coils, thus adding the difficulty of the angle between them and multiple outputs. This step is the 2 DOF control and introduces a new obstacle by adding the necessity to control the angle. The 2 DOF control is relevant to test the improved simulator, avoiding the accumulation of errors by allowing to detect them in an earlier stage, and to test the capabilities of this techniques in a more complex scenario.

Lastly, all 10 coils are included in the simulator. As the main purpose of slowly adding more coils is to ensure the correct functioning of the simulator, there is no need to develop middle steps between two and 10 coils, as there is no significant increase in complexity between those middle steps. Nevertheless, this last complicated simulator requires gradual progress. Instead of adding more coils, every degree of freedom is blocked at the beginning and the model is trained to deal with a small amount of them. Then, they are unlocked progressively until the model can control all of them. This is the 5 DOF control, the main objective of the project and the control that is included to the levitation of Vèsper in the end. The control of five out of the total existing six degrees of freedom by the same system provides robustness and enhances complete integration and efficiency.

3.4.2. Simulator

In this document, the word "simulator" refers to the modelization of the dynamics of the vehicle. The customization of the Levitation & Guiding System impedes the use of a commercial, widely used simulator as it is done with car or plane simulators. By contrast, this aspect obliges the development of a custom modelization of the dynamics of the vehicle. The simulator acts as the real coils of the vehicle, calculating the next measurement of the sensors given the action stated by the agent.

The presence of a simulator brings numerous advantages. First, it reduces the time of experimentation. An experiment that would take a few minutes of development and an hour of preparation can be massively reproduced in a few nanoseconds. Moreover, the simulator eliminated the need for physical elements to proceed with the experiments, saving in costs of material and human resources, saving in space required for it, and eliminating risks.

The simulator includes a model per type of coil –HEMS and EMS–, and a model of the complete vehicle –with two coils for 2 DOF and ten coils for 5 DOF–.

The 1 DOF model includes a Resistance-Inductance circuit –commonly called RL circuit but named RI circuit in this document to avoid confusion with Reinforcement Learning– to translate from the voltage received to the current flowing through the coil, two modules to calculate the vertical and horizontal forces produced by the coil, and a module to calculate the next air gap with the steel plate if it is working alone. This model is deeply explained in Sections 4.3.1 and 4.4.1.

The model of 2 DOF replaces the module of the coil for the air gap calculations with trigonometrical calculations to take into account both coils. More details about this simulator are given in Section 5.1.

The model of 5 DOF replaces the module of air gap calculations of the 2 DOF model with more complex calculations based on the Newton-Euler Dynamic Equations of Motion [14] to take into account multiple angles and momentum. Apart from that, the 5 DOF model calculates both the air gap between the unit and the steel plate and the measures of the sensors. Section 6.2 delves into the functioning of this dynamic model.

3.4.3. Environment

By contrast with the term *simulator*, *environment* refers to the elements that communicate the agent with the simulator, evaluating the action of the agent at each time step, starting and finishing the *episodes* –which are a set of steps with a dependency among them–, and providing graphical visualization for better understanding of the behavior of the Control System.

The environment is built over the library Gymnasium [15], a library based on Gym from OpenAI which serves as an API for multiple RL environments. Nevertheless, the environment is also customized for the project, including the following modules:

- The initializer selects the initial state to begin and the variables to be used in the training process.
- The *step module* executes the action decided by the agent, computes the reward corresponding to said action, and stops the episode once the *truncated* –sudden stop of the episode– or *terminated* –the episode came to a natural end because it levitated for the desired period of time– flag is triggered.
- The *render module* graphically depicts the evolution of an episode.
- The *reset module* restarts a new episode, deciding the initial state.

3.4.4. Training process

This section describes the common aspects of the training processes carried out and the main variations tried among them. For the training process, the library *Stable-Baselines3* [16], built over PyTorch [17] which allows for lower-level personalization if necessary.

During the training process, a linear schedule for the learning rate has been implemented. This way, as the training progresses and gets closer to a solution, the weight changes are less significant.

During the process, a maximum time of five seconds –5000 steps as the frequency of the Control System is 1 kHz– has been established for each episode, varying between including or not the possibility of truncating the episode when the system touches the floor or ceiling. This time must be necessary for the control to stabilize.

Different models have been compared at several stages of the project, all of them proved to be useful with continuous action and observation spaces, and they are mentioned in the section where they were used. Nevertheless, the main aspects that have varied among the experiments are related to the definition of the MDP –reward, action space, and observation space–, as these aspects have been demonstrated to be the most critical ones.

The evolution of the training process has been monitored with the platform Weights & Biases [18], guaranteeing a clear view of the different parameters involved. Besides, an *EvalCallback* from Stable-Baselines3 has been included to validate the functioning of the agent at a certain frequency of steps, stopping the process if an appropriate expected return is achieved.

CHAPTER 4
One degree of freedom

This chapter delves into the development of the Control System for One Degree of Freedom (1 DOF), the first approximation to the solution to the problem. The 1 DOF control is the maximum simplification of the system which has complete sense, as well as the minimal system that can be tested in a real scenario –combining the designed boards, the levitation units, and the control system–.

First, in Section 4.1 an overview of the functioning of the 1 DOF control is presented, including a deeper explanation of its importance. Then, Section 4.2 describes the test bench used to validate this control. Finally, the two developed approaches are explained.

Section 4.3 describes the aspects involved in the first approach for this control. Based on the philosophy of last year, this first control tries to act over the current flowing in the coil. This approach simplifies the problem and relies on the existence of a latter control translating from current to voltage.

Section 4.4 explains the final proposal for one levitation unit, the voltage control. This approach provides a clear advantage over the proposal of last year, as just one model acts over the voltage instead of having two models to accomplish it.

4.1 Overview

The one degree of freedom control consists of a single levitation unit that attracts or repels from a steel plate. The only permitted movement, the only degree of freedom, is the vertical displacement –Z-axis–. This movement is one-dimensional. The rest of the displacements and rotations are disabled.

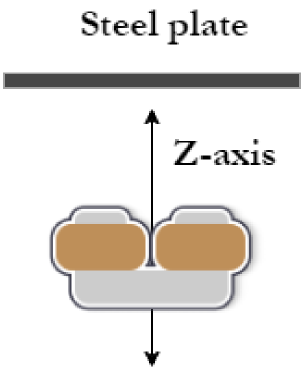


Figure 4.1: Allowed displacement in 1 DOF control

The 1 DOF control is an important step of the process, as it is the minimum unit of functioning of the control. This first step serves as the base for the rest of the project. The 1 DOF control aims to check if creating an RL agent capable of maintaining a single levitation unit at the desired height despite perturbations is possible.

With the 1 DOF control, multiple aspects of the project can be tested. Firstly, the proper emulation of the functioning of the coil, which is the basic feature of the simulator, can be ensured. Apart from that, the capabilities of the new technique, the RL agent, can be checked. We can evaluate the benefits of the new control in a simple first stage. Moreover, the electromagnetic design of the units can be characterized and compared to the results of the simulations. Lastly, we can observe if the boards have accomplished the desired requirements, and if the control meets the margins of computational cost.

The most relevant aspect of this first stage is the possibility of testing a simpler version of the system in a real scenario. To achieve that, a 1 DOF test bench was designed by the team, explained in Section 4.2.

4.2 1 DOF test bench

The *1 DOF test bench* is the structure built to test the 1 DOF control. This structure is made of aluminum profiles with linear guides attached to a steel plate. The steel plate incorporates an appendix where weights can be adhered, making it possible to match this weight with the kilograms the unit needs to hold during a run. The structure also includes an extension where the air gap sensor is located. Figure 4.2 shows the appearance of the test bench.

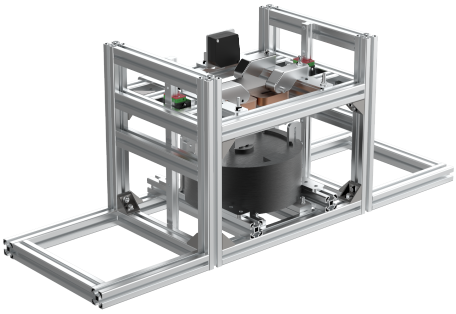


Figure 4.2: 1 DOF test bench

Source: Hyperloop UPV (2023)

This structure restricts any possible rotation or movement except the vertical displacement, allowing, for example, to validate that the simulator of the dynamics of the coil is correct.

4.3 Current control

The first approximation to the 1 DOF control consists of substituting the PID of the approach of the previous year with a Reinforcement Learning agent. In this approach, the model must decide which current to apply to the coil at each time step, assuming that the coil can reach that level of amperage instantly. The objective of this simplification of the first stage is to construct a reliable base for the project.

4.3.1. Dynamic model

The dynamic model –also called simulator or plant– emulates the behavior of the coil. Table 4.1 shows the parameters conditioning the behavior of the plant, based on the levitation units of the eighth generation of the team –as the new parameters were not known yet.

Parameter	Value
Maximum air gap (mm)	23
Minimum air gap (mm)	10
Objective height (mm)	19.5
Frequency (kHz)	1
Current range (A)	±45
Vehicle weight (kg)	200

Table 4.1: Plant characteristics

This plant includes the following modules:

- **Force Module**: It consists of a regression model trained on a dataset of experimental data. The model is a multilayer perceptron with five hidden layers containing the following number of neurons: 16, 32, 64, 32, and 16, each followed by a ReLU activation function. The model receives the air gap and current as the input and outputs the resultant force of the coil, F_C.
- **Position Computation**: Knowing the vertical force exerted by the coil and the gravity force corresponding to a quarter part of the vehicle, g, the total force is computed, F_T. Once the total force is divided by a quarter of the mass of the vehicle, m, the acceleration, a, is obtained. Once the acceleration is integrated, the velocity, v, is obtained, which is integrated again to calculate the position, p. This last value is multiplied by 1000 to be converted from meters to millimeters.

The complete algorithm of the model is shown in Algorithm 4.1:

Algorithm 4.1 Plant - Simple coil

```
Require: $airgap \in [10, 23]$
Require: $I \in [-45, 45]$
  $F_B = verticalforce(airgap_t, I_t)$
  $P = (m/4)g$
  $F_T = P - F_C$
  $a = \frac{F_T}{m}$
  $v = \int_{T=0}^{T=t} a_T dT$
  $p = \int_{T=0}^{T=t} v_T dT$
  $airgap_{t+1} = 1000p$
  return $airgap_{t+1}$
```

Note that the position is measured as the distance between the levitation unit and the steel plate. If the air gap is close to 10, the unit is at the top, while if it is close to 23, it is close to the floor.

4.3.2. MDP

The next step is to define the Markov Decision Process of the problem, consisting of the observation space –or set of states–, the action space, and the reward function.

Observation space

The observation space is composed of the air gap, the error with the reference height, the velocity, and the current at a certain time step, $S_t \in \mathbb{R}^4$.

The presence of the velocity is essential as it is the only reference for the model to detect if it is going up or down. If the state only contained measures about the position, it would perceive as beneficial a position a bit under the reference height even if it was approaching extremely fast to the floor, not producing enough force to stop the unit from crashing.

Action space

The action space contains only a value representing the amperage the coil needs to apply in the next step, $A_t \in \mathbb{R}$, $A_t \in [-45, 45]$. It can be observed that the action space is continuous, as the observation space, an aspect that conditions the RL algorithms can be applied.

Reward function

Lastly, the reward function divides the space into different zones with different values. The reward function is defined as follows.

$$R_t = \begin{cases} -500 & if \quad the\ vehicletouches\ the\ floor\ or\ ceiling \\ -100 * |error| & if \quad error \geq 0.5 \\ -50 * |error| & if \quad error \geq 0.1 \\ 100 & otherwise \end{cases} \tag{4.1}$$

4.3.3. Results

To check the viability of approaching this problem using Reinforcement Learning, a Soft Actor-Critic (SAC) [19] algorithm was used. The SAC algorithm is divided into two trainable models: the Actor and the Critic. The Actor is in charge of deciding which action to take based on the current state, whereas the Critic evaluates the decision taken by the Actor. At inference time, only the Actor is used.

The models used in this approach are two neural networks, both with 2 hidden layers of 256 neurons each, with ReLU as the activation function. For the output, the actor counts on two neurons to return the mean and standard deviation of a distribution from which the action is extracted. At inference time, the standard deviation is eliminated to make the model deterministic. During the training process, the learning rate has been set to 0.001, with a linear schedule to reduce its value with each episode. Adam [28] was the optimizer method used. The training process has been carried on in a machine with the following specifications:

Processor	11th Gen Intel(R) Core(TM) i5-11400F @ 2.60GHz 2.59 GHz
Kernels	6
Threads	12
RAM	16 GB
GPU	NVIDIA GeForce GTX 1650

Table 4.2: Machine specifications

The results of the training process are shown below. Figure 4.3 depicts a take off from the floor –remember that high values of air gap mean that the coil is close to the floor–, while Figure 4.4 illustrates a take off from the ceiling, starting with the coil magnetically attached to the steel plate.

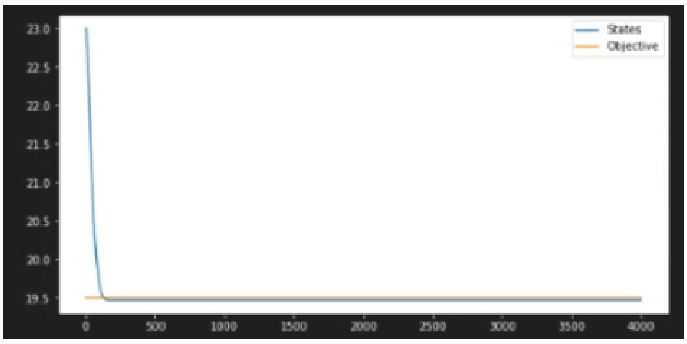


Figure 4.3: Air gap at each time step during 5 seconds, take off from the floor

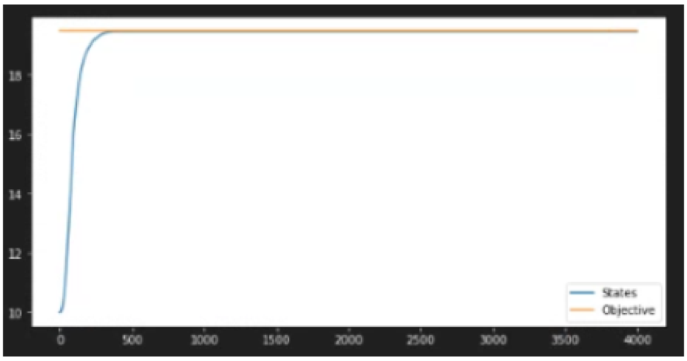


Figure 4.4: Air gap at each time step during 5 seconds, take off from the ceiling

The coil reaches the desired height after approximately a quarter of a second in both cases, evidencing the viability of approaching the problem using Reinforcement Learning.

4.4 Voltage control

The next step towards a more realistic control is to assume that it is not possible to select the exact value of current flowing through the copper coil at each time step. On the contrary, the applied voltage is going to alter the value of current, similar to speed varying the position of a corps.

4.4.1. Dynamic model

Now, the plant grows including a RI circuit, calculating the current flowing through the coil from the voltage applied. Moreover, the parameters of the dynamic model have been updated to match the design of the ninth generation.

Parameter	Value
Maximum air gap (mm)	22.5
Minimum air gap (mm)	9
Objective height (mm)	16.3
Frequency (kHz)	1
Current range (A)	±45
Voltage range (V)	±100
Vehicle weight (kg)	250

Table 4.3: Plant characteristics

Including the computation of the current, the new plant is described in Algorithm 4.2:

Algorithm 4.2 Plant - Complex coil

```
Require: $airgap \in [9, 22.5]$
Require: $I \in [-45, 45]$
Require: $V \in [-100, 100]$
  $I = \int_{T=0}^{T=t} \frac{\frac{V_t}{R} - I_t}{\frac{L_t}{R}} dT$
  $F_C = verticalforce(airgap_t, I_t)$
  $P = (m/4)g$
  $F_R = P - F_B$
  $a = \frac{F_R}{m}$
  $v = \int_{T=0}^{T=t} a_T dT$
  $p = \int_{T=0}^{T=t} v_T dT$
  $airgap_{t+1} = 1000p$
  return $airgap_{t+1}$
```

The maximum and minimum levels of current and voltage are established by the Hardware subsystem based on the requirements of the boards.

4.4.2. MDP

As the dynamic model changes, the MDP of the problem must be reformulated.

Observation space

The observation space has been simplified by eliminating the air gap, maintaining the error, velocity, and current, $S_t \in \mathbb{R}^3$.

Action space

The action space now represents the voltage applied to the coil, $A_t \in \mathbb{R}$, $A_t \in [-100, 100]$. The action space is still continuous.

Reward function

Two different reward functions have been used in this approach. The first is the same as in Equation 4.1. The second one is defined as follows.

$$R_t = -|error| \tag{4.2}$$

Simpler reward functions tend to perform better, so function 4.2 follows this idea.

4.4.3. Results

This time, with the aim of optimizing the model for the microcontroller by minimizing the amount of FLOating Point Operations (FLOPS) inside the board, the two hidden layers of the Actor network varied from 256x256 to 4x4 neurons. The process followed for training, as the model faces a more complex situation, is divided into three steps:

- Learn how to maintain the position at the objective height.
- Increase the range of heights where the episode can start from 14.3 mm to 18.3 mm.
- Increase the range of possible beginnings to [9, 22.5].

The best results were obtained with a model of 8x8 neurons and with reward function 4.1. The training process is the same as in Section 4.3.3 –learning rate, linear decay...–, except during the fine-tuning stage, where the learning rate starts at 10^{-5}. These specifications obtain the best results at every stage –1 DOF, 2 DOF, and 5 DOF–, except the number of neurons that is increased for the 5 DOF control. The results are shown in Figures 4.6 and 4.5.

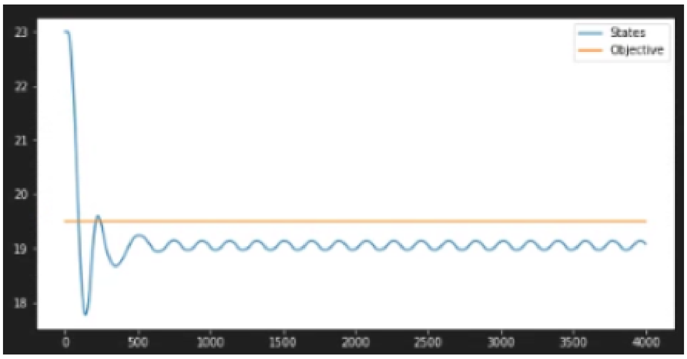


Figure 4.5: Air gap at each time step during 5 seconds, take off from the floor

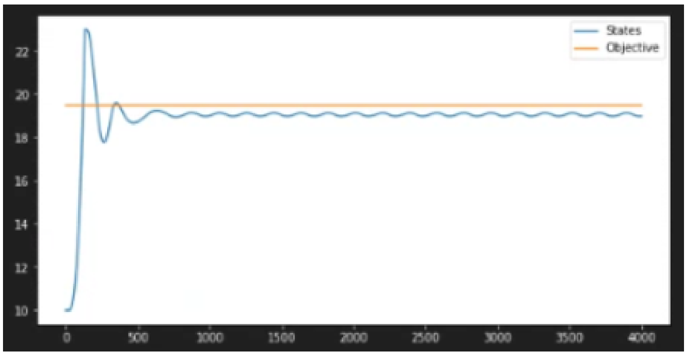


Figure 4.6: Air gap at each time step during 5 seconds, take off from the ceiling

It can be observed that the agent oscillates around a suboptimal zone –note that SAC is a method that obtains a suboptimal result to avoid overfitting–. Once the reward function is simplified as 4.2, the Critic better evaluates the actions of the Actor, being able to reduce the Actor network up to 8x8 neurons with the results shown in Figures 4.8 and 4.7

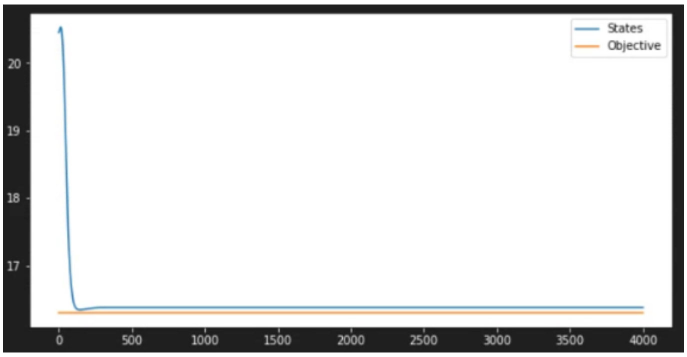


Figure 4.7: Air gap at each time step during 5 seconds, take off from the floor

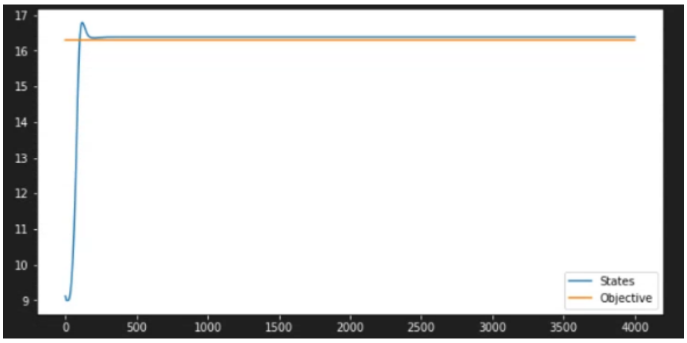


Figure 4.8: Air gap at each time step during 5 seconds, take off from the ceiling

The simplification of the reward function reduces the oscillation and the gap with the objective height once the controller is stabilized. With the aim of comparing different models, two other algorithms have been compared: Proximal Policy Optimization (PPO) [20] and Twin Delayed DDPG (TD3) [21]. TD3 is able to obtain similar results as in Figure 4.6 and 4.5, being outperformed by SAC. PPO, however, is not able to stabilize the system.

4.4.4. Performance analysis in the microcontroller

A key aspect of the 1 DOF control, as stated at the beginning of the chapter, is the possibility of testing the performance inside the boards as a starting point. To validate the computational resources needed to execute the inference of the model, the Actor is exported as ONNX file. Then, the STM X-CUBE-AI package of STM32CubeMX is used to translate the model into C++ code for the STM32H723ZGT6 microcontroller. This program offers different levels and options of compression in case it is necessary.

Table 4.4 depicts the time spent on inference with the translated model. As the inference is applied at a frequency of 1 kHz, the asymptotic maximum time spent on inference can be 1 ms –assuming no other operations are performed inside the microcontroller, which is too optimistic but serves as a baseline–.

Comp. Level	Comp. Type	RMSE	Inference time (ms)
None	None	0	0.042
Low	Time	0.000001	0.005
Low	RAM	0.000001	0.005
High	Time	0.003664497	0.006
High	RAM	0.0036645	0.006

Table 4.4: Performance of the model inside the microcontroller

It can be noted that the model spends only 4.2% of the maximum available time for inference. This percentage is reduced to 0.5% with the lowest level of compression and insignificant loss in performance.

CHAPTER 5
Two degrees of freedom

This chapter revolves around the next step of the project, the 2 DOF control. The importance of this step is not linked to real-world testing but as a middle point in the increase of complexity of both the dynamic and the inference model.

In this stage, the dynamic model significantly increases its difficulty, and the fact that two coils have to coordinate themselves to achieve satisfactory results complicates the task. Achieving proper results in this stage reduces the gap in complexity between the 1 DOF and 5 DOF controls, ensuring having achieved a reliable step in the middle with a simplification of the new concepts that need to function in the 5 DOF.

Section 5.1 describes the new dynamic model, composed of two coils which introduce a new dimension. Section 5.2 illustrates the new MDP for the problem, introducing the concept of frame-stacking. Finally, Section 5.3 shows the obtained results.

5.1 Dynamic model

The new dynamic model perceives the vehicle as a two-dimensional corps, as a segment with a coil at each side. Therefore, now the possible movements are the displacement on the Z-axis, and the rotation on the Y-axis, as shown in Figure 5.1.

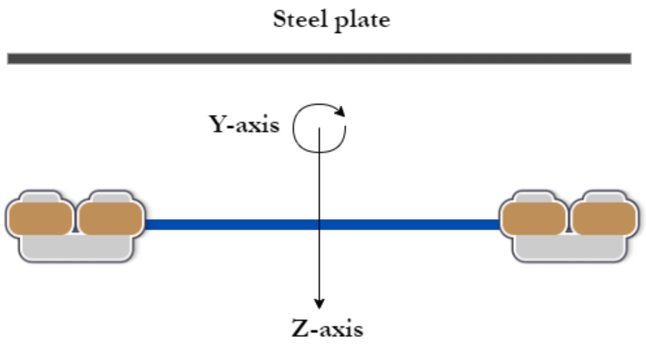


Figure 5.1: Allowed movements in 2 DOF control

This new setup requires an adaptation of the simulator, considering the effect of the action of a coil on the other. To make this possible, the momentum M of the force is computed to calculate the angle generated by the application of both forces. Then, using trigonometry, the air gap of each coil is deduced. Algorithm 5.1 shows the mathematical operations applied in this process.

Algorithm 5.1 Plant - 2 DOF

```
Require: $airgap_1, airgap_2 \in [9, 22.5]$
Require: $I_1, I_2 \in [-45, 45]$
Require: $V_1, V_2 \in [-100, 100]$
  $inertia = 70; D_{cm} = 0.7$
  $I_1 = \int_{T=0}^{T=t} \frac{\frac{V_{1t}}{R} - I_{1t}}{\frac{L_t}{R}} dT$
  $I_2 = \int_{T=0}^{T=t} \frac{\frac{V_{2t}}{R} - I_{2t}}{\frac{L_t}{R}} dT$
  $F_{C1} = verticalforce(airgap_{1t}, I_{1t})$
  $F_{C2} = verticalforce(airgap_{2t}, I_{2t})$
  $M = F_{C1} D_{cm} - F_{C2} D_{cm}$
  $\alpha = \frac{M}{inertia}$
  $\omega = \int_{T=0}^{T=t} \alpha_T dT$
  $\theta = \int_{T=0}^{T=t} \omega_T dT$
  $P = (m/2)g$
  $F_T = P - F_{C1} - F_{C2}$
  $a = \frac{F_T}{m/2}$
  $v = \int_{T=0}^{T=t} a_T dT$
  $p = \int_{T=0}^{T=t} v_T dT$
  $zpos_{t+1} = 1000p$
  $airgap_{1t+1} = zpos + 1000 D_{cm} \sin\theta$
  $airgap_{2t+1} = zpos - 1000 D_{cm} \sin\theta$
  return $airgap_{1t+1}, airgap_{2t+1}$
```

Some important details are that D_{cm} refers to the distance of the coils to the center of mass of the vehicle and that the reference system selected takes the vertical force towards the steel plate –up– as negative. In contrast, the force of the gravity is positive. The latter aspect can be seen in Figure 5.1.

5.2 MDP

The modelization of the MDP for the 2 DOF problem is based on the successful results on the 1 DOF control.

Observation space

As it was noted with the 1 DOF control, the state needs to contain a feature referencing the direction in which the system is moving. However, the Firmware Lead of the previous year stated that the calculations to provide the velocities and positions on each axis once the 5 DOF was applied brought a significant number of operations that the microcontroller had trouble dealing with.

This experience led to the search for other representations for the state, finding the concept of *frame-stacking* [22].

5.2.1. Frame stacking

The concept of frame-stacking is often used on sequential problems such as time series or videos. It consists of stacking the last N values of the state to provide sequential knowledge to the representation. For example, if applied to the context of video data, N frames –images– would be concatenated as an array as the input for the model.

In the context of this project, the observation space concatenates the air gap measures of the last two time steps, and the current measures of the last step –as for the current there is no need to control the variation but just the previous value–. Considering this explanation, $S_t \in \mathbb{R}^6$.

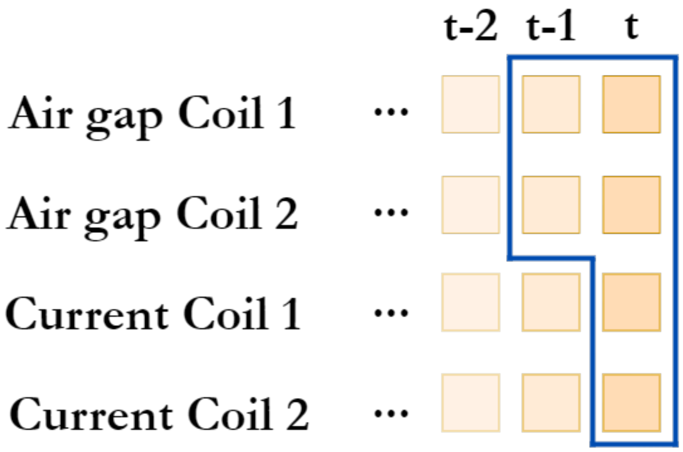


Figure 5.2: Frames forming the state of the MDP

Action space

The action space is duplicated as it depicts the voltage applied to each coil, $A_t \in \mathbb{R}^2$, $\forall i \in \{1,2\}\ A_t[i] \in [-100, 100]$.

Reward function

Four different reward functions have been used during the experiments due to the rise in complexity:

$$R_1 = -|error_1| - |error_2| \tag{5.1}$$

$$R_2 = alive - |error_1| - |error_2| \tag{5.2}$$

$$R_3 = -|error_1| - |error_2| - 10000crash \tag{5.3}$$

$$R_4 = -((\frac{error_1}{\max(error_1)})^2 + (\frac{error_2}{\max(error_2)})^2) - 0.4crash + R_{bonus} \tag{5.4}$$

$$R_{bonus} = \begin{cases} -0.2, & \text{if } error_{t-1} < error_t \\ 0.2, & \text{if } error_{t-1} > error_t \\ 0, & \text{otherwise} \end{cases} \tag{5.5}$$

The reward function 5.1 follows the principle of the successful 1 DOF representation. Functions 5.2 and 5.3 incorporate a new element to prevent the system from crashing –touching the floor or ceiling–. The former uses a positive parameter *alive* with the value of 20 –as it will always be higher in absolute value than the maximum sum of errors in a time step– to give the agent a positive reward if it is able to maintain the vehicle levitating for the maximum amount of time. The latter uses the opposite strategy, penalizing the agent with a big negative reward if it crashes. This way, the accumulated reward is always worse if it crashes than if it is able to levitate during the whole episode.

The reward function 5.4 follows the idea of [23]. In this last reward function, the errors are normalized to fit the range $[0, 1]$ and it includes a penalization in case of crashing. Moreover, a penalization or a positive reward is added if the error increases or decreases respectively, accelerating convergence.

5.3 Results

Given the demonstrated superiority of SAC in the previous step and the advantages provided by the Actor-Critic structure, as the combination of a simple Actor with a complex Critic provides high performance and fast inference at the same time, no other algorithms have been tested in this step.

The Actor-network includes two hidden layers of eight neurons each, with ReLU as the activation function. For the output, the actor counts on two neurons to return the mean and standard deviation of a distribution from which the action is extracted. At inference time, the standard deviation is eliminated to make the model deterministic. The Critic-network counts on two hidden layers of 256 neurons each, with ReLU as the activation function.

During the training process, the learning rate has been set to 0.001, with a linear schedule to reduce its value with each episode. Nevertheless, for the fine-tuning stages, the starting learning rate is diminished to 10^{-5}. Adam [28] was the optimizer method used. The training process has been carried on in a machine with the following specifications:

Processor	11th Gen Intel(R) Core(TM) i5-11400F @ 2.60GHz 2.59 GHz
Kernels	6
Threads	12
RAM	16 GB
GPU	NVIDIA GeForce GTX 1650

Table 5.1: Machine specifications

The reward function 5.1 did not achieve a desirable result, as shown in Figure 5.3. The result of this first reward function led to the definition of function 5.2, which finds a way to keep levitating during the whole episode but without a satisfactory performance.

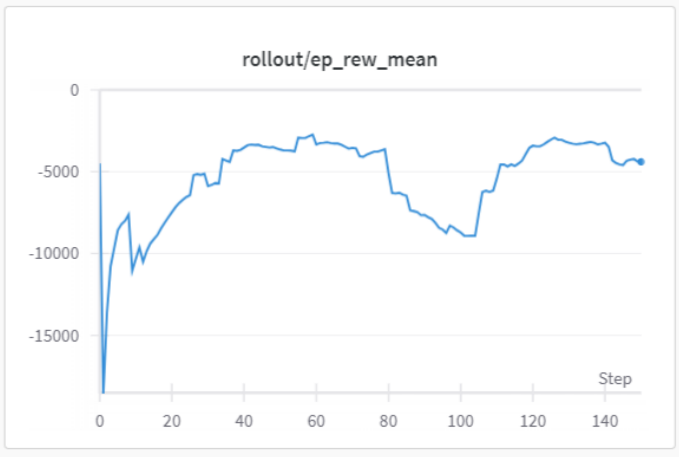


Figure 5.3: Evolution of the reward with reward function 5.1

Reward 5.3 achieves acceptable performance at the objective height but is unable to adapt to other initialization with variations of the initial height and angle. However, after introducing function 5.4, the results obtained are finally successful. Figure 5.4 shows the evolution of the reward function along the episodes of the training process..

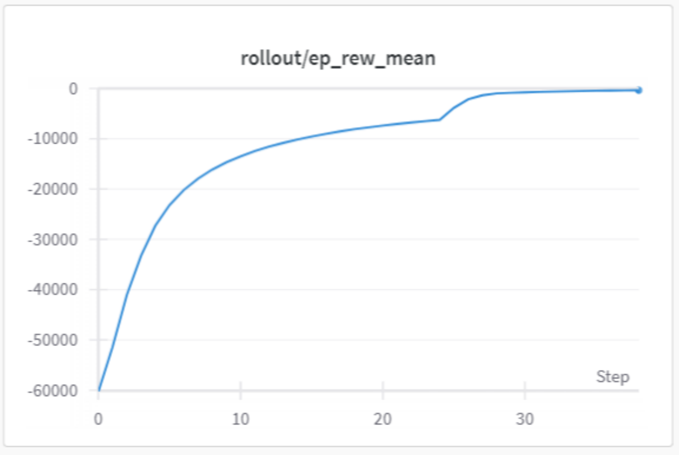


Figure 5.4: Evolution of the reward with reward function 5.4

The performance of the best model, obtained with reward function 5.4 are shown in Figures 5.5, 5.6, and 5.7.

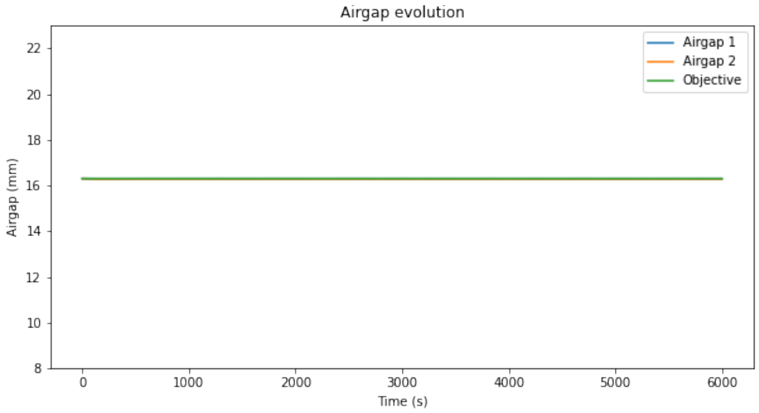


Figure 5.5: Performance initializing from equilibrium position

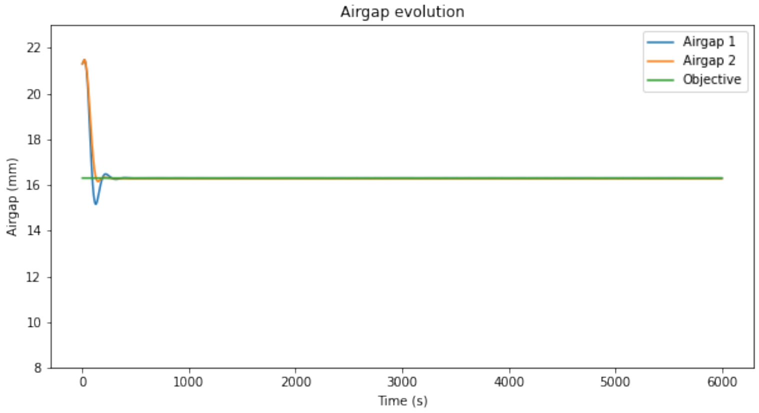


Figure 5.6: Performance initializing from below the equilibrium position

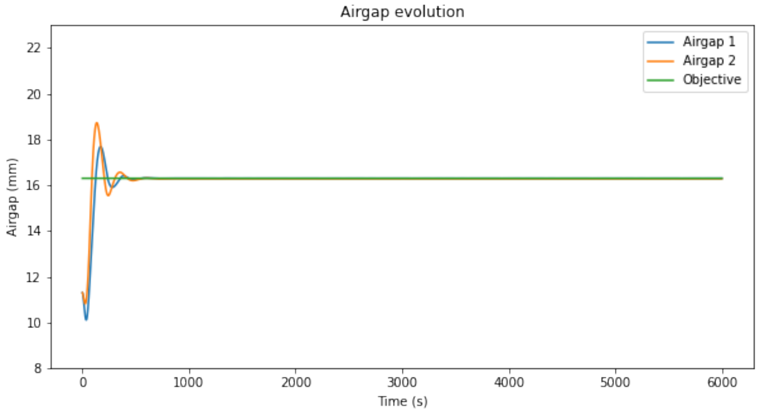


Figure 5.7: Performance initializing from above the equilibrium position

The agent is not only able to maintain the equilibrium position but also to correct its position from different positions and angles.

CHAPTER 6

Five degrees of freedom

The 5 DOF control is the last stage of the process, the final goal of the project. The aim is to create a control system able to manage 10 different coils to stabilize the angles in the X, Y, and Z-axis, and the position in the Y and Z-axis. In other words, the control system needs to stabilize the vehicle from all possible perturbations and keep it at the desired height and centered inside the infrastructure. The level of complexity significantly increases from the previous experiments, although important learned concepts can be applied to it.

In this chapter, the development of the 5 DOF control is described. Section 6.1 introduces the main aspects to understand what the 5 DOF control is. Then, Section 6.2 provides the details of the dynamic model, clearing the reference system established, describing the calculations of the angles and positions, and explaining how the real air gap measures are computed from the previous results. By including these last calculations, the vehicle does not need to calculate the position and angle from the air gap measures, eliminating computationally expensive operations during inference.

Section 6.3 defines the new mathematical formulation of the problem, the Markov Decision Process involving all five degrees of freedom. After that, Section 6.5 describes the experiments performed. As this stage is significantly more complex than the previous ones, it needs to be divided into substages consisting of blocking some degrees of freedom but allowing the model to control all the coils already. Therefore, the model can progressively adapt the control to the new obstacles included in the training process. After every substage, Section 6.6 illustrates the results with the complete control system functioning.

This control system is the final one, the control system that is being incorporated into Vèsper for the competition. Therefore, it also needs to be tested inside the microcontroller, with the same requirements as in Section 3.4.4. Considering this situation, Section 6.7 evaluates the performance of the vehicle inside the microcontroller and the possibility of running at the desired requirements.

6.1 Overview

The 5 DOF control is in charge of maintaining the vehicle at the desired height and laterally centered inside the infrastructure, mitigating the perturbations generated by the system. The system controls five out of the six total degrees of freedom, leaving just the longitudinal movement to the motor.

The 5 DOF control is the final version, the system that is incorporated in Vèsper to achieve the best possible levitation. An accurate control system minimizes current consumption and eliminates friction with the ground, being the key aspect for the sustainability and efficiency of the vehicle. Moreover, an optimized control system in terms of computational complexity allows for incorporating redundant firmware protections and prevents the electronics from being saturated because of high amounts of operations.

Specifically, the 5 DOF control acts over the voltage of 10 coils, divided into four HEMS units for vertical levitation and six EMS units for horizontal levitation, arranged as depicted in Figure 6.1.

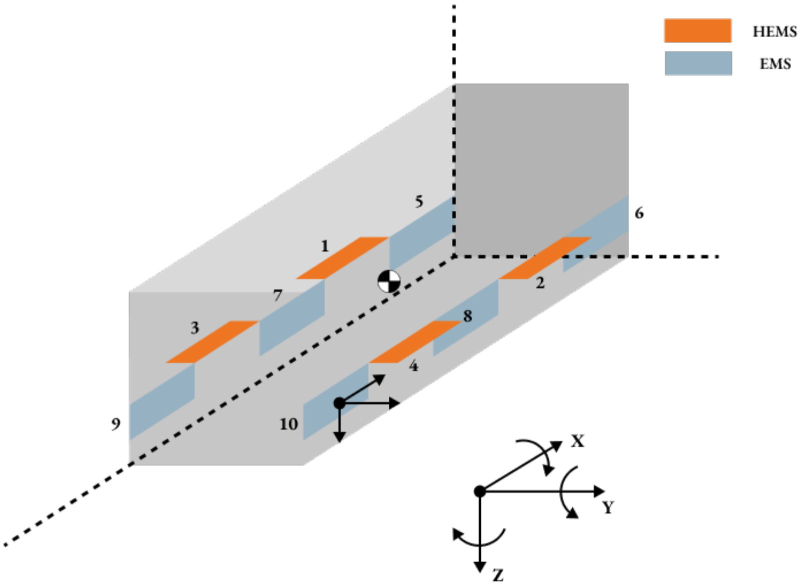


Figure 6.1: Location and numeration of each coil in Vèsper

Source: Hyperloop UPV (2024)

By coordinating all 10 coils, the 5 DOF control must be able to maintain the minimum possible rotation in all three axes, as well as keeping the vehicle at the desired vertical and horizontal positions. Moreover, the control system must achieve this objective with minimum consumption. Lastly, the control system needs to be run inside the microcontroller at a frequency of 1 kHz.

6.2 Dynamic model

The new dynamic model involves the addition of three main changes. The first change is that the dynamic model introduces a new type of coil, the EMS. The EMS model is built over the structure of the HEMS. Nevertheless, instead of vertically, the EMS produces forces horizontally. Moreover, the forces produced have a different characterization for the same current value. The last change is that the current flowing through the EMS units is always positive.

The second change is motivated by the duration of the training process. Given the frequency of execution of the inference and the sequential nature of the episodes, minimal time spent on each time step computation significantly increases the temporal length of the training process. Therefore, the prediction models have been replaced with a lookup table as the one extracted from the simulations, but with a 10x increase in precision. This method avoids running computationally and temporally expensive simulations and achieves a 4x increase in training speed.

The last change is the process followed to compute the air gaps of the coils and the air gap measures of the sensors. With the rise of complexity of the dynamic model, some trigonometric calculations are not enough for this task. Thus, the new computations are based on the Newton-Euler Dynamic Equations of Motion [14]. The process is explained in Sections 6.2.1, 6.2.2, and 6.2.3.

6.2.1. Definition of the degrees of freedom and the reference system

The first concept that needs to be clarified is the definition of the degrees of freedom and the reference system for the measures. The reference system, the definition of the axes, is shown in Figure 6.2.

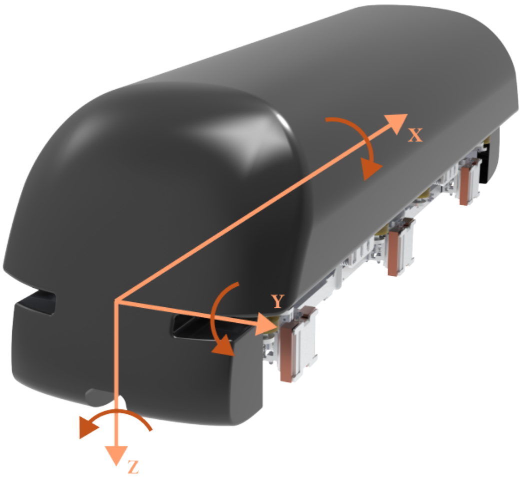


Figure 6.2: Definition of the axes in Vèsper

Source: Hyperloop UPV (2024)

The defined axes lead to six different degrees of freedom:

- **X-axis movement**: Corresponding to the longitudinal movement, the direction in which the track is built. This degree of freedom is controlled by the motor, which makes the vehicle move forward and backward.
- **X-axis rotation**: Corresponding to the roll. This degree of freedom is controlled by the vertical units, depending on the coordination of units 1 and 3 with 2 and 4.
- **Y-axis movement**: Corresponding to the lateral movement. This degree of freedom depends on the EMS units, whose aim is to maintain the vehicle centered.
- **Y-axis rotation**: Corresponding to the pitch. This degree of freedom is controlled by the vertical units, depending on the coordination of units 1 and 2 with 3 and 4.
- **Z-axis movement**: Corresponding to the vertical movement, the direction in which the vehicle levitates. This degree of freedom is controlled by the HEMS units, whose aim is to maintain the vehicle at the objective height.
- **Z-axis rotation**: Corresponding to the jaw. This degree of freedom is controlled by the EMS units, depending on the coordination of units 5 and 10 with 6 and 9.

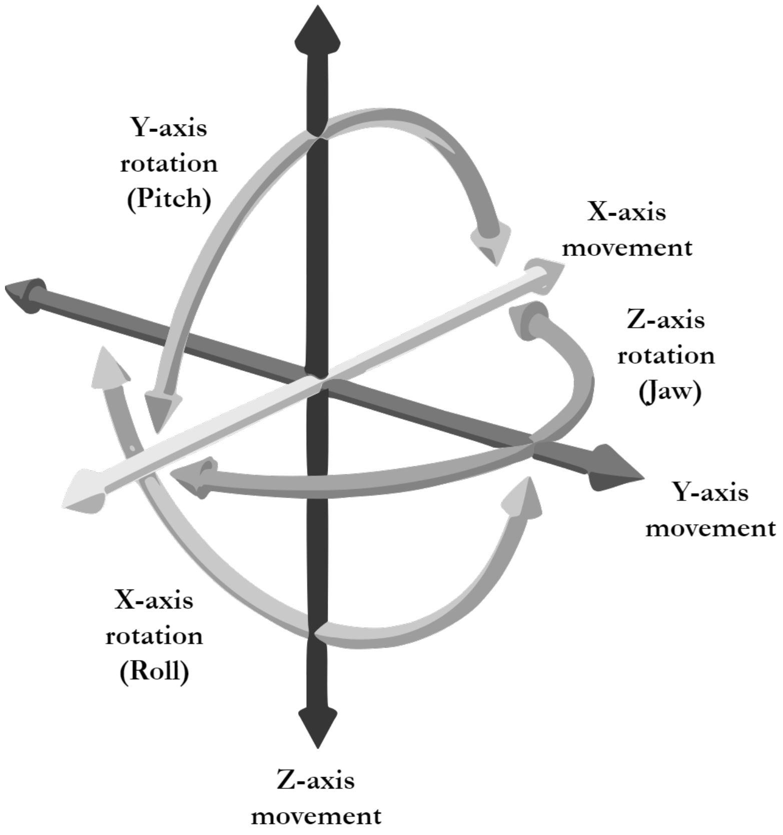


Figure 6.3: Definition of the degrees of freedom

6.2.2. Position and angle calculations

Once the reference system is clear, the next step is describing the first stage of the dynamic model: the calculation of the positions –X and Y-axis– and angles –X, Y, and Z-angles–.

This process is similar to the dynamic model of the 2 DOF control but increases the angles, positions, and coils calculated. Algorithm 6.1 formalizes this first stage.

Algorithm 6.1 Positions and angles calculations - 5 DOF

```
Require: $\forall i \in [1, 10],\ airgap[i] \in [9, 22.5]$
Require: $\forall i \in [1, 10],\ I[i] \in [-45, 45]$
Require: $\forall i \in [1, 10],\ V[i] \in [-100, 100]$
  for $i \leftarrow 1$ to 10 do
    $I[i] = \int_{T=0}^{T=t} \frac{\frac{V[i]_t}{R} - I[i]_t}{\frac{L_t}{R}} dT$
    $F[i]_V = verticalforce(airgap[i]_t, I[i]_t)$
    $F[i]_H = horizontalforce(airgap[i]_t, I[i]_t)$
  end for
  for $ax \in [''x'', ''y'', ''z'']$ do
    $M[ax] = momento(ax, F_V, F_H)$
  end for
  $zpos_{t+1} = get\_zpos(F_V)$
  $ypos_{t+1} = get\_ypos(F_H)$
```

Note that the abstraction of the functions *momento*, *get_zpos*, and *get_ypos* is since the internal calculations follow the same steps as in the 2 DOF control.

6.2.3. Transformation to air gap measures

The last stage of the dynamic model consists of transforming the position and angle of the vehicle into the air gap measures of the sensors. As stated before, this process allows for the agent to directly receive as input the measures of the sensors, relevantly reducing the number of calculations at inference time. The measures of the sensors depend not only on the distance to the steel plates but also the angle with it. Algorithm 6.2 formalizes this first stage.

Algorithm 6.2 Air gap calculations - 5 DOF

```
Require: $\forall i \in [1, 10],\ airgap[i] \in [9, 22.5]$
  for $i \leftarrow 1$ to 4 do
    $airgap\_sensor[i] = -zpos - 1000(pos\_sensor_y[i] \cdot sin(\theta_x) + pos\_sensor_z[i] \cdot cos(\theta_x) + pos\_sensor_x[i] \cdot sin(\theta_y)) - width\_L$
  end for
  for $i \leftarrow 5$ to 8 do
    if $i\%2 == 0$ then
      $airgap\_sensor[i] = -ypos - 1000(pos\_sensor_x[i] \cdot sin(\theta_z) + pos\_sensor_y[i] \cdot cos(\theta_x) + pos\_sensor_z[i] \cdot sin(\theta_x))$
    else
      $airgap\_sensor[i] = -total\_distance_y - (ypos - 1000(pos\_sensor_x[i] \cdot sin(\theta_z) + pos\_sensor_y[i] \cdot cos(\theta_x) + pos\_sensor_z[i] \cdot sin(\theta_x))$
    end if
  end for
```

Note that, even though there are six EMS units, only four lateral sensors are used in the control system.

6.3 MDP

The mathematical formulation of the problem is similar to the 2 DOF control, especially regarding the observation and action space.

Observation space

The observation space makes use of the frame-stacking technique, concatenating the air gap measures of the last two time steps and the current measures of the last step. Considering this approach and the fact that the complete system includes eight sensors and 10 coils, $S_t \in \mathbb{R}^{26}$.

Action space

The action space continues to be an array of the voltage applied to each coil at the given time step, $A_t \in \mathbb{R}^{10}$.

Reward function

The reward function is a compound function of the different degrees of freedom to be controlled, varying the weights depending on the last included step of the progress. The main function is shown in Equation 6.1.

$$R_t = 1 - \alpha|\hat{\theta}_x| - \beta|\hat{\theta}_y| - \gamma|\hat{\theta}_z| - \delta|\hat{pos}_z| - \epsilon|\hat{pos}_y| - \zeta\left|\frac{\sum_{i=1}^{4}\hat{I}_i}{4}\right| + R_{bonus} \tag{6.1}$$

In Equation 6.1, $\alpha, \beta, \gamma, \delta, \epsilon$ and ζ are the weights of each variable and their sum is equal to one. Symbol θ refers to each angle and pos to the position. The last value of the function aims to minimize the power consumption of the system. The symbol $\hat{X}$ indicates that variable X is normalized as follows:

$$\hat{X} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{6.2}$$

After the experiments, the values at each step of the process are applied as stated in Table 6.1.

Parameter	θ_X **&** θ_Y	pos_Z	pos_Y **&** θ_Z
α	0.4	0.4	0.15
β	0.6	0.3	0.25
γ	0	0	0.15
δ	0	0.3	0.2
ϵ	0	0	0.15
ζ	0	0	0.1

Table 6.1: Parameter values at each stage

Lastly, R_{bonus} serves to accelerate convergence. It acquires the value of 0.1 in case the error –regarding the degree of freedom which is the focus of the training stage– decreases, and -0.1 if it increases in respect to the previous time step.

6.4 Neural Network

The neural networks involved in the training process have been an object of experimentation to optimize the inference process. In the SAC algorithm, there are mainly two neural networks involved: the Actor and the Critic. As explained in Chapter 3, only the Actor affects the process of inference, being the only one critical to be optimized.

In that sense, the Actor-network has been simplified as much as possible without compromising its effectiveness while controlling levitation. Throughout the experiments, there have been mainly three aspects varying to find the best combination.

- The number of layers and neurons per layer have been reduced to the minimum possible, achieving the final results with a neural network of 2 hidden layers of 64x64 neurons.
- The representation of the states has been established as explained in the previous section, Section 6.3, as a combination of two frames of the sensor measures and just the last frame of the currents of the coils, to minimize the input shape. Moreover, the sensor measures have been represented as the error from the objective height instead of the measure itself, which limits the range to smaller numbers around 0.
- The final architecture is a simple Feed-Forward neural network. Although LSTM and Transformer architectures usually provide better results on sequential data, the difference in computational complexity is not worth the improvement in the case of Transformers and the experiments evidenced that LSTM architectures add computational complexity without a performance improvement.

To enhance the trade-off between exploration and exploitation, the final layer outputs the mean and standard deviation of a normal distribution from which the action value is extracted. Nevertheless, during the validation and test process, the model becomes deterministic, meaning that for a state the action is always the same.

The final architecture of the model is illustrated in Figure 6.4.

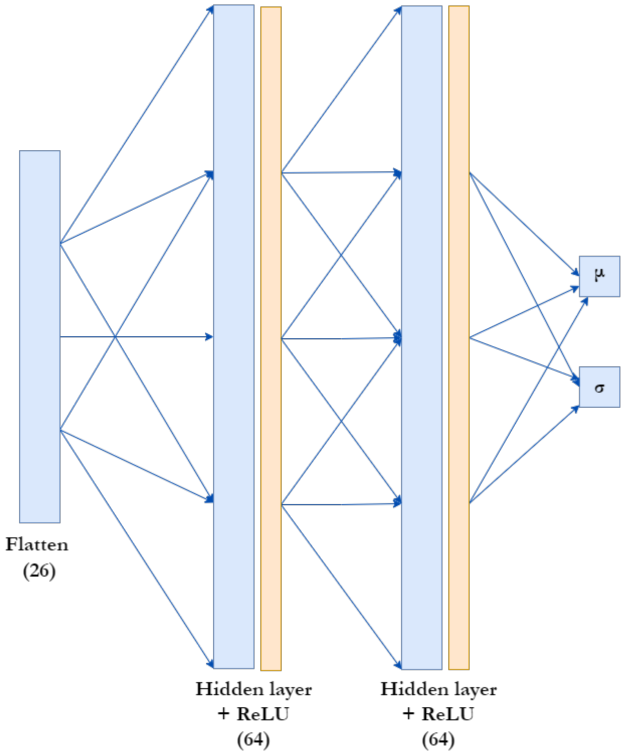


Figure 6.4: Final architecture of the Actor-network

6.5 Experiments & results

This section describes the different experiments performed in this project and illustrates the obtained results. Each stage is linked to a column of Table 6.1. Each of them consists of either fine-tuning the previous model or training an agent from scratch –in the case of the first stage– to control the newly incorporated degree of freedom together with the ones introduced during the previous steps.

The Actor-network includes two hidden layers of 64 neurons each, with ReLU as the activation function. For the output, the actor counts on two neurons to return the mean and standard deviation of a distribution from which the action is extracted. At inference time, the standard deviation is eliminated to make the model deterministic. The Critic-network counts on two hidden layers of 256 neurons each, with ReLU as the activation function.

During the training process, the learning rate has been set to 0.001, with a linear schedule to reduce its value with each episode. Nevertheless, for the fine-tuning stages, the starting learning rate is diminished to 10^{-5}. Adam [28] was the optimizer method used. The training process has been carried on in a machine with the following specifications:

Processor	11th Gen Intel(R) Core(TM) i5-11400F @ 2.60GHz 2.59 GHz
Kernels	6
Threads	12
RAM	16 GB
GPU	NVIDIA GeForce GTX 1650

Table 6.2: Machine specifications

6.5.1. X & Y angles

The first stage consists of training an agent from scratch to control the angles θ_X and θ_Y. They are introduced together in the first stage because they can be understood as the same task, teaching the vertical levitation units to stabilize the vehicle without focusing on elevating it to the desired height. Moreover, this task has the best simplicity-importance ratio, which leads to its selection as the first stage. To achieve it, the simulator restricts every degree of freedom but those angles.

In this stage, the vehicle is kept at the objective distance and centered in the infrastructure. The simulator of Vèsper eliminates the vertical and horizontal displacement and the jaw rotation from the calculations, simplifying the scenario of states. As the Z-axis rotation and the horizontal displacement are eliminated, the EMS units are useless at this moment, and the training process is focused on the HEMS units.

As can be seen in Table 6.1, $\gamma = \delta = \epsilon = 0$, avoiding giving a reward for behavior not caused by the agent but by the restrictions introduced. Therefore, the agent first develops the ability to control the selected degrees of freedom. The training process results, finished after the agent accomplished a mean episode reward higher than 5000 in the validation callback, are shown hereunder in Figure 6.5.

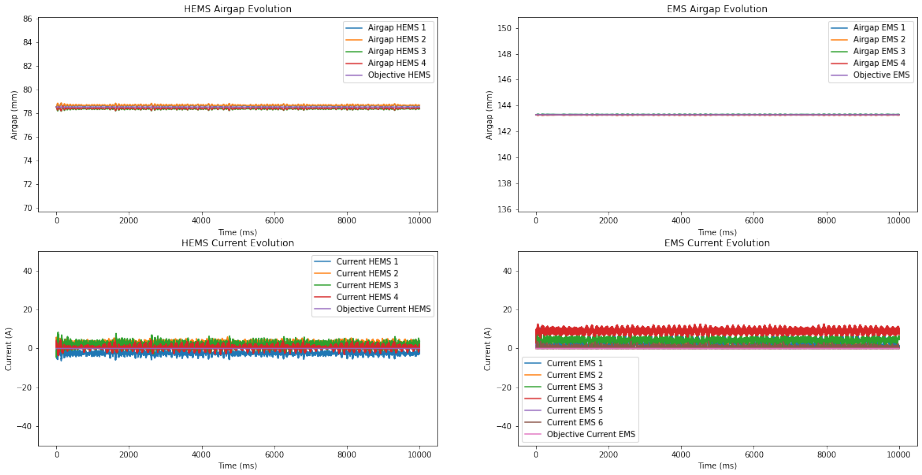


Figure 6.5: Results controlling the angles θ_X and θ_Y

The agent is capable of controlling the two degrees of freedom with an impressive performance, with less than 1 mm of difference between the measures of the different sensors during the whole episode. It is important to highlight that, even though $\zeta = 0$, the agent found that the best possible behavior entails spending the minimum possible current during levitation.

Note that the duration of the testing episode is 10 seconds, which is twice the duration of the episodes of training, and the agent maintains the same level of performance.

6.5.2. Z position

Regarding vertical levitation, the other independent task is to maintain the vehicle at the desired height, controlling pos_Z. This aspect incorporated is the same as the 1 DOF control. Nevertheless, the increased number of read values and decided actions, together with the need to maintain the behavior learned in the previous step, raise the complexity of the process.

For this stage, the vertical displacement is unlocked in the simulator, allowing the vehicle to fall down or up. For that reason, $\delta > 0$, which means that the agent obtains results from maintaining a correct height while still stabilizing the angles θ_X and θ_Y.

For that purpose, the model from the previous step has been fine-tuned with the new reward function to acquire this new capability. The results are shown in Figure 6.6.

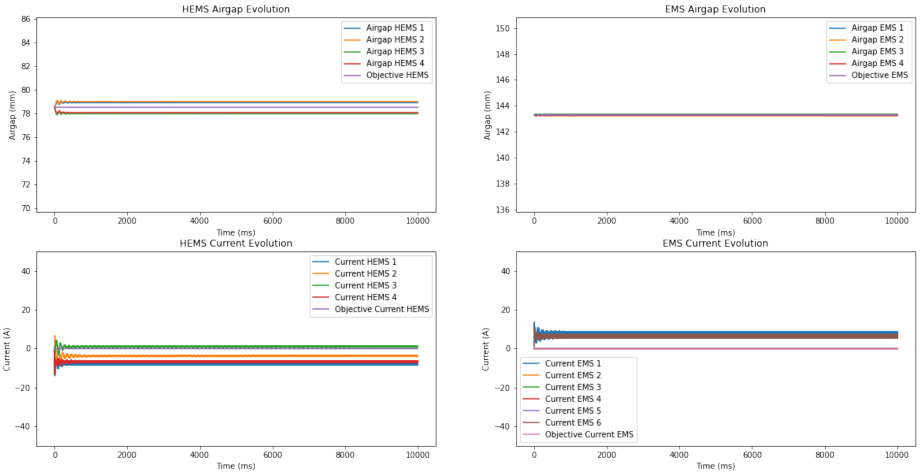


Figure 6.6: Results controlling the angles θ_X and θ_Y, and the position in the Z-axis

The agent is capable of maintaining not only a stable angle as before but also the correct height. Nevertheless, it can be observed that angle θ_Y has worsened concerning the previous step, as it is the most difficult angle to control and now the agent has to focus on more aspects of the levitation. However, the oscillations have been mitigated.

By achieving the control of the vehicle at this stage, the principal function of the levitation control, the control of the HEMS, is dominated.

6.5.3. Y position & Z angle

The last stage is the control of the horizontal levitation, the EMS units, which corresponds to the angle θ_Z and position pos_Y. The EMS control is significantly easier and less important than the vertical control.

It is easier as there are no constant horizontal forces induced on the vehicle, in contrast to the gravity and magnetic forces in the vertical levitation control. Therefore, moving away from the objective position does not aggravate the situation, whereas in vertical control, the further Vèsper is from the desired height, the more complicated it is to return.

The horizontal control is less important because the friction created by horizontal fluctuations has a smaller effect on the performance of the vehicle than the friction created by the weight of the vehicle with the ground.

To achieve control over all five degrees of freedom, the new two degrees of freedom are unlocked on the simulator, and parameters γ and ϵ are changed to be greater than 0. Thus, the agent obtains a reward by controlling them. Once again, the model is fine-tuned after these changes to acquire the new capabilities needed to control every degree of freedom. The results are shown in Figure 6.7.

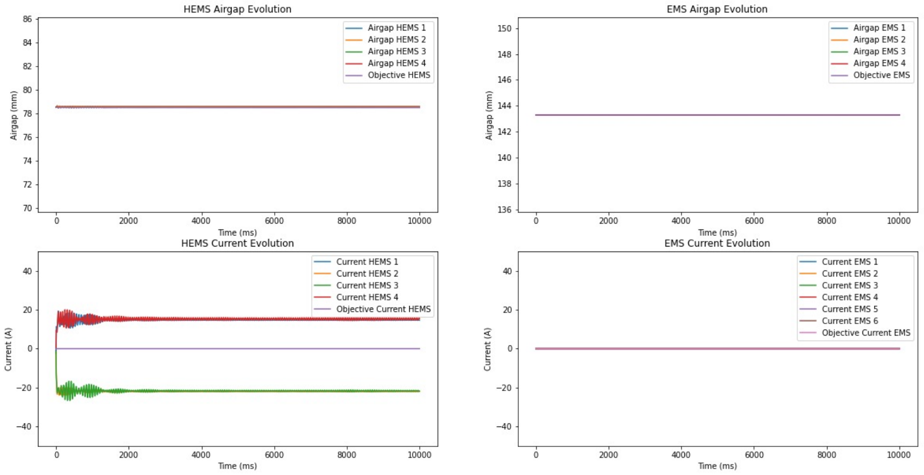


Figure 6.7: Results controlling the angles θ_X, θ_Y, and θ_Z, and positions pos_Y and pos_Z

The final model is capable of stabilizing Vèsper in all five degrees of freedom, with minimal perturbances and controlling all 10 coils at the same time.

6.6 Discussion of results

The training process has achieved the objective with impressive performance, with oscillations smaller than 0.5 mm in the measures –note that the sensors include a random noise in a range of ±0.1 mm–. However, Figure 6.7 evidences the lack of efficiency of the system as each coil consumes a mean current of around 20 A.

To solve the problem of efficiency, the parameter ζ is introduced, $\zeta > 0$. This component of the reward function leads the agent to minimal consumption, without losing control over the five degrees of freedom. The final results are shown in Figure 6.8.

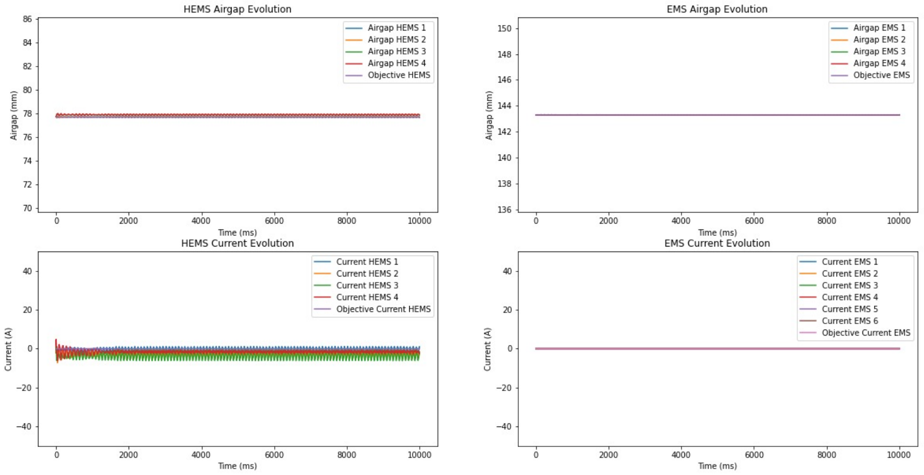


Figure 6.8: Results controlling the angles θ_X, θ_Y, and θ_Z, and positions pos_Y and pos_Z with minimal consumption

The new component of the reward function solves the issue detected. Therefore, the control system maintains the vehicle at the desired position with stability and minimal energy consumption.

6.7 Performance in the microcontroller

Once the model works as required, it needs to be validated inside the control boards to ensure that it is possible to be executed at a frequency of 1 kHz. It is important to remember that implementing the inference at that frequency would be impossible if each computation lasted for 1 ms or more, being this the maximum available time for inference.

The procedure to profile the neural network is the same as in Section 4.4.4. First, the model is exported as an ONNX file. Then, the STMCubeMX is configured according to the electronics system of Vèsper –selecting the model of MCU, the pinout distribution, the frequency of the clock...–. After that, the X-CUBE-AI package is used to translate the ONNX model into C++ code for the board, and different compression levels are compared in terms of performance inside the microcontroller. The results are illustrated in Table 6.3

Comp. Level	**Comp. Type**	**RMSE**	**Inference time (ms)**
None	None	0	0.069
Lossless	RAM	0.0000001	0.069
Lossless	Time	0.0000001	0.069
Low	RAM	0.0042485	0.1
Low	Time	0.0042485	0.1

Table 6.3: Performance of the model inside the microcontroller

Again, the compression of the program does not reduce the inference time, even increasing sometimes, but deteriorates the performance in terms of error. Thus, the selected model is the original one.

The model can execute the inference of an observation in 0.069 ms, which corresponds to 6.9% of the maximum available time for inference. The model not only can control the levitation of the vehicle but is also highly capable of being executed at a frequency of 1 kHz inside the STMH723ZGT6.

CHAPTER 7

Conclusions

This chapter shares the conclusions drawn after the development of the project. First, the fulfillment of the objectives of the project is analyzed in Section 7.1, and then Section 7.2 describes the relation of the project with different courses from the Bachelor′s Degree in Data Science.

7.1 Fulfillment of objectives

After finishing the project, it can be stated that all the objectives were fulfilled. The part of the document where each is accomplished is shown hereunder.

- The simulator of the vehicle has been successfully created, starting in Chapter 4 and ending in Chapter 6 with the complete dynamic model.
- The model for the control of the vertical levitation of one coil is explained in Chapter 4.
- The model for the control of the levitation of two coils is explained in Chapter 5.
- The model for the control of the levitation of the complete system, with 10 coils, is explained in Chapter 6.
- The model was optimized at each stage of the process by reducing the size of the neural network and utilizing the simplest architecture possible, especially in Chapters 4 and 6.
- The performance of the model in its dedicated embedded system is evaluated at two stages, in Section 4.4.4 and 6.7.

All statements considered, it can be concluded that Reinforcement Learning is a suitable substitute for classical control methods, achieving energy savings and great performance.

7.2 Applicable theory

This section relates the contents of the project with the knowledge acquired from the different courses of the bachelor′s degree in Data Science.

- **Continuous modeling and simulation (14023)**: The course consists of the development of mathematical models to simulate real-life processes, especially via differential equations. This course has a direct link to the dynamic model of the vehicle developed in this project.
- **Evaluation, deployment, and monitorization of models (14028)**: This course includes information about many different AI techniques, such as the Random Forests

used to predict the inductance of the coil in the first dynamic models. Moreover, the course provides knowledge about CD/CI techniques used to work as a team.

- **Infrastructure for data processing (14016)**: This course provided the necessary hardware base to understand the functioning of the embedded system in which the model develops the process of inference.
- **Predictive and descriptive models II (14011) & Scalable Machine Learning techniques (14009)**: Both courses provided information about the functioning of neural networks, different architectures such as Feed-Forward or Long Short-Term Memory (LSTM), and different ways of optimizing the process of inference and training.
- **Programming Fundamentals (14002), Programming (14003) & Algorithmics (14007)**: The courses which provided the necessary knowledge for successful programming, especially in Python –the main language used in this project– but transferrable to other languages –such as C++–.
- **Project management (14018)**: Both during the planning of the Final Degree Project and as the Team Captain of Hyperloop UPV, the Project management techniques learned have been a crucial aspect of the process.

CHAPTER 8

Future work

This chapter describes the work proposed as a continuation of this project, with ideas for improvement both already implemented but unfinished or not yet applied.

The first proposal is to create a more robust control system. Although the control system developed can maintain the vehicle at the desired height, it is unable to deal with highly detrimental positions or angles, thus crashing before reaching the objective position.

The process of implementing this first proposal has already started. To achieve it, the 5 DOF model is being trained from the beginning with random initialization of angles and positions. As the current model cannot deal with these situations some adjustments have been made.

- The stack of frames of the sensor measures has been changed from two time steps to four. This way, the model has a longer temporal perspective and can perceive phenomena as acceleration.
- The action space has been normalized to the range [-2.5, 2.5] as it is the range where the gradient of the hyperbolic tangent –the last activation function of the model– is significantly greater than 0, as shown in Figure 8.1.
- The negative bonuses received once the measure is worsening have been eliminated. This has been removed because, with highly detrimental initialization, reducing the speed of the vehicle once it is approaching the floor or ceiling to change the direction of movement produced worse cumulative rewards than letting the vehicle crash.

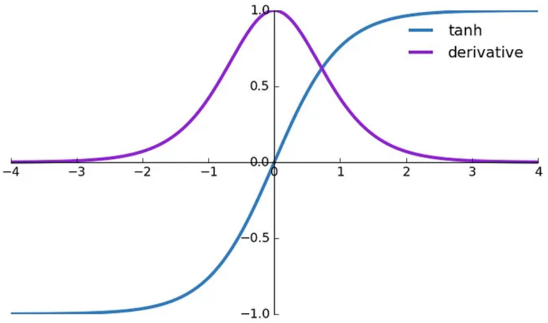


Figure 8.1: Hyperbolic tangent gradient

Source: Extracted from [24]

At the moment, the new model has been able to learn how to control the angles θ_X and θ_Y, and the position pos_Z from the most detrimental situations, with both changes implemented. The results are shown in Figure 8.2.

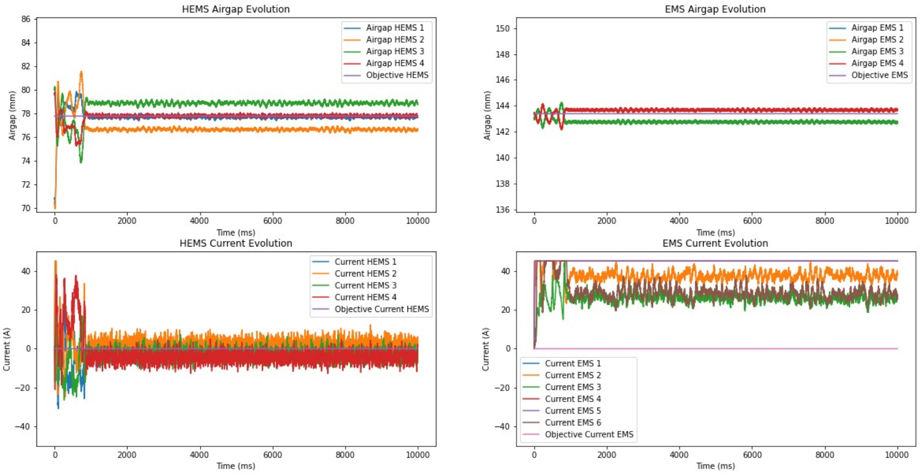


Figure 8.2: Example of the results controlling angles θ_X and θ_Y, and position pos_Z

The model successfully stabilizes the vehicle. There is a difference of two millimeters between the front and back measures, which is a small difference taking into account that the length of Vèsper is two meters. Moreover, although there is no restriction in terms of current spending, it can be observed that the model gathers the values of current around 0 once the vehicle is stabilized. Note that the EMS units are not taking part in the control at this stage, thus spending high amounts of current without any effect on the control.

The next step consists of including the horizontal control and the component of the reward that focuses on minimizing the energy spent.

The second proposal for further work is to incorporate the code generated by X-CUBE-AI into the boards of Vèsper, thus validating the functioning of the control system in real-life scenarios. This proposal is directly dependent on the timeline of the team, having to wait until the control and power boards are ready to perform levitation, and the vehicle and infrastructure are completely assembled. Once the control system is validated in real life, it is time to compete against internationally known universities such as MIT, TU Delft, or ETH Zurich in the EHW.

Bibliography

[1] Press release of Renfe, December 19th 2020. Consulted at https://www.renfe.com/es/es/grupo-renfe/comunicacion/renfe-al-dia/sala-de-prensa/valencia/Aniversario-Ave-Valencia-22millones#:~:text=En%20cuanto%20a%20la%20velocidad,hora%2C%20ha%20afirmado%20el%20ministro.

[2] Deutsche Bahn. Consulted at https://int.bahn.de/es/trenes/trenes-de-larga-distancia/ice_3.

[3] The International Maglev Board. Consulted at https://www.maglevboard.net/en/facts/26-transrapid-maglev-shanghai.

[4] Lluesma, Federico and Arguedas, Antonio and Hoyas, Sergio and Sánchez, Alberto and Vicén, Juan. Evacuated-Tube, High-Speed, Autonomous Maglev (Hyperloop) Transport System for Long-Distance Travel: An overview. *IEEE Electrification Magazine*, 9:4:67–73, 2021. https://ieeexplore.ieee.org/document/9632845.

[5] Janić, M. Estimation of direct energy consumption and CO2 emission by high speed rail, transrapid maglev and hyperloop passenger transport systems. *International Journal of Sustainable Transportation*, 15:9:696–717, 2021. https://doi.org/10.1080/15568318.2020.1789780.

[6] Michele Mossi and Pierre Rossel. Swissmetro: a revolution in the high-speed passenger transport systems. *Swiss Transport Research Conference*, March, 2001. https://www.strc.ch/2001/mossi.pdf.

[7] Musk, E. Hyperloop Alpha. *SpaceX/Tesla Motors*, August, 2013. https://www.tesla.com/sites/default/files/blog_images/hyperloop-alpha.pdf.

[8] STMicroelectronics *Data brief - X-CUBE-AI - Artificial intelligence (AI) software expansion for STM32Cube*. https://www.st.com/resource/en/data_brief/x-cube-ai.pdf.

[9] National Institute of Standards and Technology (NIST). Consulted at https://csrc.nist.gov/glossary/term/control_system.

[10] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning : an introduction*. Second edition. Cambridge, MA : The MIT Press, [2018]. Series: Adaptive computation and machine learning series.

[11] Kiran, B Ravi and Sobh, Ibrahim and Talpaert, Victor and Mannion, Patrick and Sallab, Ahmad A. Al and Yogamani, Senthil and Pérez, Patrick. Deep Reinforcement Learning for Autonomous Driving: A Survey. *IEEE Transactions on Intelligent Transportation Systems*, 23:6:4909–4926, 2022. https://ieeexplore.ieee.org/document/9351818.

[12] Kober J, Bagnell JA, Peters J. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32:11:1238–1274, 2013. https://journals.sagepub.com/doi/pdf/10.1177/0278364913495721?casa_token=1LPSKuIzXqMAAAAA:srxbftSU6P7zh1_Ys5cn94oV8xGnDgXFEl5ahDHyV_uyuyp2k_JFVT5HoijSkqiYXpjnrz-sWiDssw.

[13] Introducing ChatGPT by OpenAI. Consulted at https://openai.com/blog/chatgpt.

[14] Eric Stoneking. Newton-Euler Dynamic Equations of Motion for a Multi-body Spacecraft. *NASA Goddard Space Flight Center*, Greenbelt, MD 20771, USA. https://ntrs.nasa.gov/api/citations/20080044854/downloads/20080044854.pdf.

[15] Gymnasium Documentation. Consulted at https://gymnasium.farama.org/.

[16] Stable-Baselines3 Documentation. Consulted at https://stable-baselines3.readthedocs.io/en/master/.

[17] PyTorch Documentation. Consulted at https://pytorch.org/.

[18] Weights & Biases Webpage. Consulted at https://wandb.ai/site.

[19] Tuomas Haarnoja and Aurick Zhou and Pieter Abbeel and Sergey Levine. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. *arXiv*, 1801.01290, 2018. https://arxiv.org/pdf/1801.01290.

[20] John Schulman and Filip Wolski and Prafulla Dhariwal and Alec Radford and Oleg Klimov. Proximal Policy Optimization Algorithms. *arXiv*, 1707.06347, 2017. https://arxiv.org/pdf/1707.06347.

[21] Scott Fujimoto and Herke van Hoof and David Meger. Addressing Function Approximation Error in Actor-Critic Methods. *arXiv*, 1802.09477, 2018. https://arxiv.org/pdf/1802.09477.

[22] Xu Tian and Jun Zhang and Zejun Ma and Yi He and Juan Wei. Frame Stacking and Retaining for Recurrent Neural Network Acoustic Model. *arXiv*, 1705.05992, 2017. https://arxiv.org/pdf/1705.05992.

[23] Passalis, N., Tefas, A. Continuous drone control using deep reinforcement learning for frontal view person shooting. *Neural Comput & Applic*, 32, 4227–4238, 2020. https://doi.org/10.1007/s00521-019-04330-6.

[24] Activation Functions with Derivative and Python code: Sigmoid vs Tanh Vs Relu. Consulted at https://medium.com/@omkar.nallagoni/activation-functions-with-derivative-and-python-code-sigmoid-vs-tanh-vs-relu-44d23915c

[25] The 17 Goals. *United Nations*. Consulted at https://sdgs.un.org/goals.

[26] Linear Motor Aircraft Launch System Takes the Steam Out of Catapults. Consulted at https://insights.globalspec.com/article/1365/linear-motor-aircraft-launch-system-takes-the-steam-out-of-catapults.

[27] Pulling together: Superconducting electromagnets. Consulted at https://www.home.cern/science/engineering/pulling-together-superconducting-electromagnets.

[28] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv*, 1412.6980, 2017. https://arxiv.org/abs/1412.6980.

APPENDIX A

Sustainable Development Goals

Degree in which the work is related to the Sustainable Development Goals (SDGs):

Sustainable Development Goals	**High**	**Medium**	**Low**	**Not applicable**
Goal 1. **No Poverty.**				X
Goal 2. **Zero Hunger.**				X
Goal 3. **Good Health and Well-being.**				X
Goal 4. **Quality education.**			X	
Goal 5. **Gender Equality.**				X
Goal 6. **Clean Water and Sanitation.**				X
Goal 7. **Affordable and Clean Energy.**	X			
Goal 8. **Decent Work and Economic Growth.**		X		
Goal 9. **Industry, Innovation and Infrastructure.**	X			
Goal 10. **Reduced Inequalities.**				X
Goal 11. **Sustainable Cities and Communities.**	X			
Goal 12. **Responsible Consumption and Production.**			X	
Goal 13. **Climate Action.**	X			
Goal 14. **Life Below Water.**				X
Goal 15. **Life on Land.**				X
Goal 16. **Peace, Justice and Strong Institutions.**				X
Goal 17. **Partnerships for the goals.**	X			

Reflection on the relationship between the TFG and the SDGs:

The SDGs are a group of 17 goals developed by the General Assembly of the United Nations in 2015 to tackle the main problems of the world before 2030. Among those problems, there can be found topics such as sustainability, biodiversity, equality, globalization...

The description of each goal is shared at [25], showing the description and objectives of each of them.

The current project has accomplished the development of a control system for the levitation of a hyperloop vehicle using Reinforcement Learning, minimizing energy consumption. Therefore, the most related SDGs are the following:

- **Goal 7. Affordable and Clean Energy**: The use of levitation with the developed levitation control minimizes the energy spent. The system spends less than 2 A per levitation unit to maintain the position of stability.
- **Goal 9. Industry, Innovation and Infrastructure**: The development of innovations like hyperloop brings several new technologies to the table that can be used in many different fields. Linear motors are already being used to launch aircraft [26]. Electromagnets are being used at CERN to guide the particles through a precise path [27]. Reinforcement Learning can also be applied to the control of different systems in other industries to achieve automation.
- **Goal 11. Sustainable Cities and Communities**: The main goal of hyperloop as a mode of transportation is to provide more sustainable mobility. By reducing the friction with the air and the ground it outperforms the current alternatives for its optimal range of distances, connecting cities with a faster and more sustainable means of transport.
- **Goal 13. Climate Action**: Hyperloop is a fully electric means of transport, with zero emissions. Therefore, as shown in Chapter 2, this mode of transportation is significantly less pollutant than the train or plane.

Moreover, this project is also related, in a less significant way, with **Goal 8. Decent Work and Economic Growth**, as hyperloop intends to allow global connection by reducing commuting time and allowing to expand the limits of mobility.

The project, although at a lower level, is also related to the following SDGs:

- **Goal 4. Quality education**: The initiative of *Generación Espontánea* aims to achieve a greater level of education, approaching real-world engineering scenarios to the students.
- **Goal 12. Responsible Consumption and Production**: The developed control system minimizes electricity consumption. Moreover, hyperloop is thought to transport multiple people in a fast and sustainable way, with minimal energy spending.