FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Takeoff and Landing with an Auxiliary Vehicle of an Autonomous Wing in Airborne Wind Energy Systems

João Pedro Martinho Legoinha - up202004912

Mestrado em Engenharia Eletrotécnica e Computadores

Supervisor: Prof. Fernando Arménio da Costa Castro e Fontes

31 of July, 2025

© João Pedro Martinho Legoinha, 2025

Takeoff and Landing with an Auxiliary Vehicle of an Autonomous Wing in Airborne Wind Energy Systems

João Pedro Martinho Legoinha - up202004912

Mestrado em Engenharia Eletrotécnica e Computadores

31 of July, 2025

Resumo

Esta dissertação investiga o desenvolvimento de Sistemas Aéreos de Energia Eólica (AWES), uma tecnologia promissora de energia renovável que capta a energia eólica em altitudes elevadas através de dispositivos voadores ligados por cabos. O foco centra-se na implementação de técnicas automáticas de descolagem e aterragem com base na abordagem Circular Takeoff and Landing (CTOL), que utiliza trajetórias circulares para permitir uma operação eficiente e autónoma. A investigação enfatiza o design e o desenvolvimento de uma arquitetura de controlo robusta, adaptada a aeronaves de asa rígida, integrando controladores de supervisão e de baixo nível para garantir um controlo preciso da altitude, velocidade e trajetória durante as fases críticas da operação.Adicionalmente, o estudo explora o design e a prototipagem de veículos auxiliares para suportar a abordagem CTOL. Estes veículos melhoram a funcionalidade do sistema ao simplificar as operações, reduzir o peso da aeronave e proporcionar elevada portabilidade e adaptabilidade em diferentes ambientes de implementação. Avanços em hardware, incluindo a integração de sensores, sistemas de propulsão e gestão de energia, são também abordados com o objetivo de otimizar o desempenho.A dissertação inclui uma revisão abrangente do estado da arte em AWES, forças atuantes no sistema, dinâmica de voo e metodologias de controlo. Os resultados e metodologias apresentados neste trabalho contribuem para o avanço do campo de Sistemas Aéreos de Energia Eólica, ao melhorar a fiabilidade, escalabilidade e eficiência dos sistemas, apoiando assim a transição global para fontes de energia renováveis e sustentáveis.

Abstract

This dissertation focuses on the context of Airborne Wind Energy Systems (AWES), a promising renewable energy technology that captures high-altitude wind energy using tethered flying devices. The focus is on the implementation of automatic takeoff and landing techniques based on the Circular Takeoff and Landing (CTOL) approach, which leverages circular trajectories to enable efficient and autonomous operation.The research centers on developing a robust control architecture for rigid-wing aircraft, integrating supervisory and low-level controllers to precisely control altitude, speed, and trajectory during critical phases of operation.More specifically, it explores the design and prototyping of auxiliary vehicles and their control systems to support the CTOL approach. These vehicles enhance system functionality by simplifying operations, reducing aircraft weight, and providing high portability and adaptability across various deployment environments. Advances in hardware, including sensor integration, propulsion systems, and energy management, are also addressed to optimize performance.The dissertation includes a comprehensive review of the state of the art in AWES, forces acting on the system, flight dynamics, and control methodologies. The findings and methodologies presented in this work contribute to advancing the field of airborne wind energy by improving system reliability, scalability, and efficiency, thereby supporting the global transition to sustainable and renewable energy sources.

UN Sustainable Development Goals

The SDGs provide a global framework to achieve a better and more sustainable future for all. They include 17 goals that address the world's most pressing challenges, such as poverty, inequality, climate change, environmental degradation, peace, and justice.

This dissertation aligns with several SDGs by contributing to the development of technologies that promote clean energy production, industrial innovation, and climate action. Specifically, it supports the following goals:

SDG 7 Ensure access to affordable, reliable, sustainable, and modern energy for all. The research advances Airborne Wind Energy Systems (AWES) as a renewable energy source using autonomous kites. The auxiliary vehicle's energy-aware design reduces idle power consumption by 20%, while low-cost wireless communication (nRF24L01) enhances accessibility to clean energy technologies globally.

SDG 9 Build resilient infrastructure, promote inclusive and sustainable industrialization, and foster innovation. The project prototypes a portable ground vehicle with 3D-printed modular components, reducing system weight by 40%. Innovative Kuramoto-based synchronization algorithms and LQR/PID controllers enable efficient operation in diverse environments, supporting sustainable infrastructure development.

SDG 13 Take urgent action to combat climate change and its impacts. By replacing concrete-intensive wind towers with lightweight AWES, material usage is cut by 60%. The technology's potential to generate $5\times$ more power than conventional turbines accelerates fossil fuel displacement. Open-source simulations and hardware designs disseminate climate mitigation knowledge.

SDG	Target	Contribution	Performance Indicators and Metrics
7	7.2	Development of a controlled system to support Airborne Wind Energy (AWE), which increases the share of renewable energy in the global energy mix.	Demonstrated feasibility of CTOL operations for AWE systems.
	7.3	The ground vehicle (RC car) is programmed to stop during idle periods (e.g., waiting for the kite to reach sufficient height), minimizing power usage and increasing energy efficiency.	Reduction in idle motor energy consumption; energy-aware vehicle behaviour.
	7.a	Introduction of low-cost, low-power wireless communication (nRF24L01) to enhance accessibility and scalability of clean energy technologies.	Integration of efficient hardware components with low energy consumption.
9	9.4	Design of innovative control strategies (Kuramoto-based synchronization) to modernize airborne systems and improve energy efficiency.	Implementation of decentralized, real-time control loops.
	9.5	Research and prototyping in AWE ground systems contribute to technological advancement and industrial innovation.	Simulation and testing of embedded control architectures.
13	13.1	Supporting the development of renewable technologies to reduce greenhouse gas emissions through wind-based energy systems.	Reduced reliance on fossil fuels via alternative energy capture.
	13.3	Promoting awareness and capacity for climate change mitigation through the adoption of sustainable autonomous systems.	Dissemination of knowledge via academic research.

Acknowledgements

First and foremost, I would like to thank my supervisor, Fernando Arménio da Costa Castro e Fontes, and the entire UPWIND team, especially Sérgio Vinha, Gabriel Fernandes, Conrado Costa, and Manuel Fernandes, who helped me immensely and took time out of their schedules so that I could achieve the intended goals.

I also want to thank my family, who have always guided me in the best way possible and motivated me to keep working even when my willpower was low. They gave me every opportunity to pursue my goals.

I would also like to thank my friends, who gave me a great deal of motivation to complete this project and stood by my side during the most difficult moments.

It was for them that I pushed myself to deliver the best possible work, and also in the hope that the results obtained may one day be part of something greater. Therefore, once again, to all of you, Thank You!

João Legoinha

"Our greatest glory is not in never falling, but in rising every time we fall"

Confucius

Contents

1 Introduction **1**

2 State-of-the-Art **3**
2.1 Historical Background . . . 3
2.2 Airborne Wind Energy Systems . . . 3
2.3 Fly-gen systems . . . 6
2.4 Ground-gen systems . . . 8
2.5 Kite Model . . . 10
2.5.1 Acting Forces . . . 11
2.6 Crosswind flight . . . 12
2.7 Automatic Take off and Landing (ATOL) . . . 14
2.8 Circular Takeoff and Landing (CTOL) . . . 15
2.9 Kite Module control architecture for a Circular Takeoff and Landing . . . 16
2.9.1 Supervisory Control . . . 16
2.9.2 Altitude and Airspeed Controllers . . . 18
2.10 Auxiliary Vehicle . . . 20
2.11 Hardware advances . . . 21
2.12 Communications . . . 23
2.13 Synchronization Algorithms . . . 24

3 Car and Control System Model **26**
3.1 Kuramoto Control . . . 26
3.2 Visualization of the Kuramoto Control with the kite with constant velocity . . . 28
3.3 New Model of the car in Simulink . . . 29
3.4 Results of the Kite Dynamics and Control . . . 37
3.4.1 Auxiliary Vehicle Synchronization . . . 37
3.5 PWM Signal Processing and State-Based Electromagnet Control . . . 37
3.6 Simulation environment and 3D Visualization . . . 40
3.7 IMU . . . 40
3.8 Velocity Sensor . . . 41
3.9 Central Rotary Mechanism . . . 43
3.10 ArduPilot . . . 44
3.11 Mission Planner . . . 44
3.11.1 ArduPilot Incompatibility . . . 45
3.12 Simulink Kite 3D Simulation . . . 45

4 Communications **49**
4.1 System Architecture 49
4.2 Module Variations 51
4.3 Signal Path 52
4.4 Troubleshooting 53
4.4.1 Wireless Communication Software Architecture 53
4.5 Testing and Implementation of NRF24L01 Wireless Communication 54

5 Auxiliary Vehicle Design and Implementation **55**
5.1 RC Car Platform 55
5.1.1 Motor Used 55
5.1.2 Motor Control via L298N Dual H-Bridge Driver 60

6 Conclusions **61**
6.0.1 Future Improvements 62

A MATLAB/Simulink Layouts and Simulation Code **63**

B Arduino Codes Used for Communications **68**

List of Figures

2.1 Number of institutions involved in AWE (2000-2018) [2]. 4
2.2 AWES Classification [3]. 6
2.3 Different types of aircrafts in Fly-Gen systems [4]. 7
2.4 M600 Energy Kite by Makani Power Inc. [5]. 7
2.5 Differences in Generation phase (a) and Recovery phase (b) [4]. 9
2.6 Instantaneous and average mechanical power on a fixed ground system station [2]. 9
2.7 The spherical $(\boldsymbol{e}_r, \boldsymbol{e}_\phi, \boldsymbol{e}_\beta)$, global $(\boldsymbol{e}_x, \boldsymbol{e}_y, \boldsymbol{e}_z)$, and body $(\boldsymbol{e}_X, \boldsymbol{e}_Y, \boldsymbol{e}_Z) = (\boldsymbol{e}_{\text{roll}}, \boldsymbol{e}_{\text{pitch}}, \boldsymbol{e}_{\text{yaw}})$ coordinate frames [6,9]. 11
2.8 Acting forces for the circular tethered flight [6, 9]. 11
2.9 Forces acting on the Kite [7]. 12
2.10 Values of ζ in a regular kite versus with crosswind maneuver [2]. 13
2.11 The on-board peak power and additional mass for each ATOL scheme [8]. 15
2.12 Circular trajectory during take off (in blue) [9,16]. 16
2.13 Overall control strategy of a CTOL architecture [9]. 16
2.14 Control phases and sub-phases of the supervisory control [9]. 18
2.15 Structural components designed for the tethered support vehicle prototype by the UPWIND Team [10]. 22
2.16 Eletronic diagram of the vehicle (left) and the prototype model concept (right) by UPWIND Team [11]. 23

3.1 Initial phase difference between the cart and the kite. 29
3.2 Coupling begins as the cart adjusts its phase to synchronize with the kite. 29
3.3 Cart nearly aligned with the kite, reducing the phase error. 29
3.4 Full phase synchronization achieved between the cart and the kite. 29
3.5 Kite´s velocity obtained by the Atol Model. 31
3.6 Azimuth angle obtained by the Atol model. 31
3.7 Elevation angle obtained by the Atol model. 31
3.8 Kite´s velocity modified in order for the car to follow. 32
3.9 PID of the car´s model. 33
3.10 Saturation block configuration applied during states 1–4 of the control system. . . 34
3.11 Saturation block configuration applied during states 4-8 of the control system. . . 34
3.12 Azimuth angles of both car (red) and kite (blue). 35
3.13 PID used in state 1 of the ATOL model. 35
3.14 Velocity of the cart. 36
3.15 Cart velocity along with all components influencing its behavior. 36
3.16 Normalization of the car´s velocity signal through a k gain. 38
3.17 Value of the K gain. 38
3.18 Iman´s behaviour along the different states. 39

3.19 BNO055 used by UPWIND Team. 41
3.20 ET and PC methods for accurately measuring velocities and their operating principles[21]. 42
3.21 LM393 used by UPWIND Team. 43
3.22 Central rotary system: Implementation used by the UPWIND Team. 43
3.23 Complete 3D animation schematic in simulink. 46
3.24 3D aircraft simulation block in Simulink. 46
3.25 Simulation 3D Scene Configuration block. 48
3.26 Visualization of the kite trajectory in the Simulink 3D environment. 48

4.1 Transmitter module used in this work. 50
4.2 Receiver module mounted on the vehicle chassis. 50
4.3 Different versions of the NRF24L01 module [22]. 52
4.4 Signal flow diagram of the communication system. 53

5.1 Pinout diagram of the L298N motor driver [34]. 56
5.2 Block diagram of the L298N motor driver [35]. 57
5.3 Pinout diagram of the arduino nano every [35]. 58
5.4 Electronic diagram of the vehicle (left) and prototype concept (right). 58
5.5 Test circuit using an LED to emulate and observe the electromagnet signal. . . . 59
5.6 Physical testing of the car conducted in FEUP. 60

A.1 Complete Simulink model, composed of the original ATOL model and the newly implemented car model. 66
A.2 New car model implemented in Simulink, integrated into the full system. 67

List of Tables

2.1 Phases, Target States, and Controllers [9]. 18
2.2 Comparison Between Wired and Wireless Communication [23]. 23

3.1 PID gains used in the car's controller. 33
3.2 PID gains used to smooth the kite's acceleration. 36
3.3 Comparison of Ground Control Stations Compatible with ArduPilot. 45
3.4 Translation and rotation vectors used for the SkyHogg aircraft model [31]. 47

4.1 Comparison of Wireless Modules for Embedded Communication 50
4.2 Pinout Connection Between NRF24L01 Module and Arduino Nano Every 51
4.3 Main Specifications of the onboard nRF24L01 Wireless Transceiver 51

5.1 Control logic of the L298N driver for two DC motors 56

List of Acronyms

AWE	Airborne Wind Energy
AWES	Airborne Wind Energy Systems
ATOL	Automatic Take-off and Landing
TOL	Take-off and Landing
HTOL	Linear or Catapult Take-off and Landing
VTOL	Vertical Take-off and Landing
CTOL	Circular Take-off and Landing
PID	Proportional Integrative Derivative
LQR	Linear-Quadratic Regulator
FG	Fly-Gen
GG	Ground-Gen
FG-AWES	Fly-Gen Airbone Wind Energy Systems
GG-AWES	Ground-Gen Airbone Wind Energy Systems
SDGs	United Nations Sustainable Development Goals
IRENA	International Renewable Energy Agency
$\vec{e}_x, \vec{e}_y, \vec{e}_z$	Reference global frame
$\vec{e}_r, \vec{e}_\phi, \vec{e}_\beta$	Reference local frame
$\vec{e}_X, \vec{e}_Y, \vec{e}_Z$	Reference body frame
m	Mass (kg)
g	Gravitational acceleration (ms^{-2})
ρ	Air density (kgm^{-3})
A	Kite wing area (m^2)
r	Tether length (m)
p	Kite position (m)
$\dot{p}$	Kite velocity (ms^{-1})
$\ddot{p}$	Kite acceleration (ms^{-2})
δ_T	Throttle input
h	Kite altitude (m)
ϕ	Azimuthal angle (rad)
β	Elevation angle (rad)
φ	Roll angle (rad)
θ	Pitch angle (rad)
Va	Kite airspeed vector (ms^{-1})
c_D	Aerodynamic drag coefficient
c_L	Aerodynamic lift coefficient
$\vec{F}_{\text{aer}}$	Aerodynamic force (N)
$\vec{F}_{\text{drag}}$	Drag force (N)
$\vec{F}_{\text{lift}}$	Aerodynamic lift force (N)
$\vec{F}_{\text{grav}}$	Gravitational force (N)
$\vec{F}_{\text{th}}$	Tether force (N)
ϕ	Azimuthal angle (rad)
β	Elevation angle (rad)
φ	Roll angle (rad)
θ	Pitch angle (rad)

Chapter 1

Introduction

There is no greater challenge facing our nation and our planet than the climate crisis. By now, global fossil fuel consumption is reaching roughly 140 terawatt-hour, representing the majority of the world's total energy consumption, and it is facing severe issues related to environmental sustainability, supply security, and disaster risks.

Advancement of societies and, in particular, their ability to sustain larger populations, is closely related to changes in the amount and type of energy available to satisfy human needs for nourishment and to perform work. Energy is essential for services such as water supply, food production, healthcare, education, and communication—much of which depends on electricity. Over the past decades, significant advances have been made in addressing this topic, focusing on expanding renewable energy production in hopes of reducing the environmental impact caused by fossil fuels over the years.

Globally, annual renewable power capacity additions would need to reach an average of 1 066 GW per year until 2050 under the IRENA's 1.5°C scenarion 1. The share of renewable energy in the global energy mix would increase from 16% in 2020 to 77% by 2050 under this scenario. Electricity would become the main energy carrier, accounting for more than 50% of the total final energy consumption by 2050.

In this context, there has been a fast growth and spread of renewable energy plants. Among them, wind generators are the most widely used type of intermittent renewable energy harvesting. In this framework, a completely new renewable energy sector, Airborne Wind Energy (AWE), emerged in the scientific community. AWE aims at capturing wind energy at significantly increased altitudes. Devices that harness this type of energy are known as Airborne Wind Energy Systems (AWES). The high level and persistence of the energy carried by high-altitude winds, which blow in the range of 200 m to 10 km from the ground surface, have attracted the attention of several research communities since the beginning of the 1980s and today stand out as one of the most promising technologies for renewable energy production.

Despite this potential, AWES technology faces several technical challenges, particularly in achieving fully autonomous operation. One of the most critical and underexplored stages is the

takeoff and landing process of rigid-wing tethered aircraft. Conventional methods, such as Vertical or Horizontal Take-Off and Landing (VTOL/HTOL), present scalability limitations or require extensive infrastructure. Circular Takeoff and Landing (CTOL) arises as a viable alternative, but it introduces new demands on coordination and control.

The main objective of this dissertation is to design and validate a robust control system for autonomous CTOL maneuvers in AWES. A key innovation presented in this work is the use of a ground-based support vehicle—a small autonomous car—to assist the kite during the takeoff and landing phases. The car is mechanically linked to the kite via a tether and follows a circular trajectory to induce the necessary speed and lift required for takeoff. This initial acceleration removes the need for onboard propulsion (e.g., electric motors or propellers). Consequently, the kite can operate without propulsion systems or landing gear, reducing aerodynamic drag and overall weight. This allows for more efficient energy harvesting during flight, as the kite can achieve higher lift-to-drag ratios.Furthermore, without propulsion systems or heavy batteries, the kite needs only minimal power—just enough to actuate control surfaces such as the elevator. This architectural simplification increases flight endurance, reduces maintenance, and improves scalability of the Airborne Wind Energy System, especially in portable or offshore deployment scenarios. This ground vehicle is also capable of guiding and supporting the kite during critical flight phases.

A Kuramoto-based synchronization strategy is implemented to ensure coherent motion between the airborne and ground elements. The complete system is modeled and simulated in MATLAB/Simulink, using realistic dynamics and control logic adapted to fixed-wing platforms.

This work contributes to the field by proposing a novel application of synchronization theory to airborne wind systems, developing a functional CTOL control architecture, and presenting both simulation results and hardware prototyping. Together, these contributions aim to advance the viability and scalability of AWES as a reliable solution for future clean energy generation.

Chapter 2

State-of-the-Art

2.1 Historical Background

AWE is the conversion of wind energy into electricity using tethered flying devices. AWES are generally made of two main components, a ground system and at least one aircraft that are mechanically or electrically connected by ropes, often referred to as tethers.

In the 1930s, wind energy pioneer Hermann Honnef devised ideas for enormous wind power plants featuring multiple rotors on towers that stood several hundred meters tall. Aloys van Gries patented a method of using kites to deploy wind turbines at higher altitudes [2]. Van Gries essentially suggested combining large lifter kites, used by many observatories worldwide for high-altitude measurements, with the emerging wind turbine technology. This concept of harvesting high-altitude wind was further developed half a century later, during the 1970s energy crisis, by space-age pioneer Hermann Oberth, who promoted it as an alternative to fossil fuels and nuclear power. Around the same time, American engineer Miles L. Loyd laid the groundwork for quantitative analysis of AWE systems [14]. Miles L. Loyd is renowned for his pioneering idea of harnessing wind energy with turbines mounted on a kite. His calculations showed that turbines on a crosswind flying kite could generate up to five times more power than similar turbines on the ground. In 1978, he patented this concept, envisioning a wind-driven apparatus that operates reciprocally, akin to a multi-propeller aircraft, with multiple ropes connecting the kite to a ground station.

2.2 Airborne Wind Energy Systems

AWES use kites, drones, and gliders to capture energy from strong, steady winds at altitudes of 200 m -10 km, which are beyond the reach of traditional wind turbines.

Replacement of the tower and foundation in conventional wind turbines can lead to a substantial reduction in material usage, thereby decreasing energy costs, while also enabling access to a previously untapped energy resource: higher-altitude wind. The ability to adjust the flight operation of tethered devices to fluctuating wind conditions further enhances the availability of energy

compared to traditional wind turbines. This, in turn, presents a promising area for research, particularly in the development of real-time optimal control strategies capable of functioning robustly within a spatio-temporally dynamic environment.

These airborne generators can function both offshore and off-grid, supplementing wind, solar, and other renewable energy sources by conveniently filling gaps in their electricity production profiles. In terms of cost-effectiveness, reliability, and safety, this type of system can be one of the best strategies for electricity production at the moment. This is why AWE is considered an emerging technology, with significant advancements occurring in the industry.

Figure 2.1 shows the growing number of institutions worldwide involved in AWE development reflecting the increasing interest and potential of this innovative energy source.

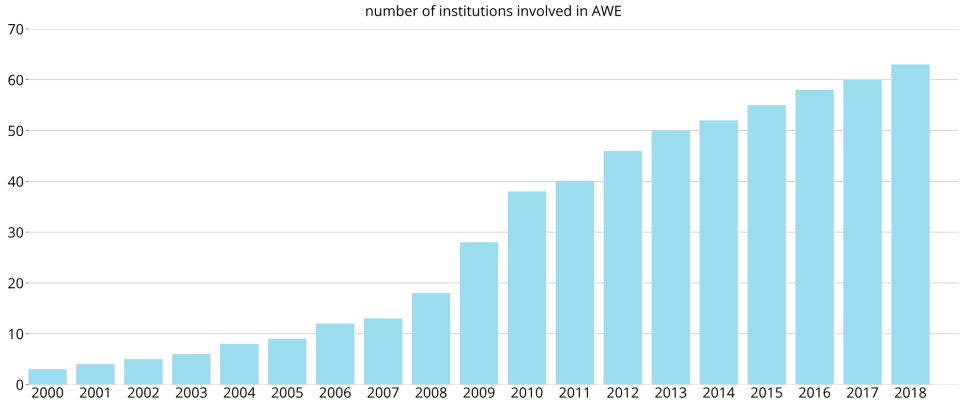


Figure 2.1: Number of institutions involved in AWE (2000-2018) [2].

AWES can be classified based on four main factors: the location where energy is produced, the structure of the system, the type of flight it performs, and the way it takes off and lands.

Regarding the location of energy production, AWES can have two main configurations:

- Fly-gen Systems - In FG-AWES, electrical energy is produced on the aircraft and it is transmitted to the ground via a special rope which carries electrical cables. In this case, electrical energy conversion is generally achieved using wind turbines. More detail information can be found here (2.3)

- Groung-Gen Systems - In GG-AWES, electrical energy is generated by harnessing aerodynamic forces, which are transmitted from the aircraft to the ground via ropes. GG-AWES can be categorized into systems with either a fixed or a moving ground station. More detail information can be found here (2.4)

In terms of the structure of these devices, there are two types:

- Rigid - Rigid wings are solid, fixed structures with little to no deformation under aerodynamic loads. They are commonly found on most conventional aircraft designs. This work

will focus on rigid wings, a topic of significant importance in the current context. Additionally, the kite model developed by the Upwind Team [6,9] will be analyzed in greater depth here (2.5).

- Soft - Soft or flexible wings are designed to deform under aerodynamic forces, allowing the shape of the wing to change dynamically during flight.

When it comes to the type of flight, AWES can have different configurations:

- Crosswind - A flight path in crosswind with a large directional component that is perpendicular to the wind direction to increase relative velocity on the kite and swept harvesting area. Among the concepts discussed for the AWE system, the majority designs execute crosswind flight, the mechanics of which are presented here (2.6).
- rotational - A rotor-type system with a circular flight path of aerodynamically active kite elements.
- Tether-aligned - A flight path that aligns with the tether direction, and which potentially reduces both the relevance to tether drag and the opportunity for area sweep.

As far as the type of TOL techniques of these systems is concerned, take-offs and landings performed:

- Vertical (VTOL) - In a VTOL system, lift propellers enable vertical takeoff and allow the aircraft to remain airborne until the energy production process is initiated.
- Horizontal (HTOL) - Regarding HTOL systems, the device accelerates horizontally over a certain distance until it reaches sufficient speed for takeoff. This acceleration is powered by an external energy source or an electromagnetic launch system.
- Circular (CTOL) - The CTOL system is the least explored of the three so far. Its concept is simple: the kite gradually gains speed through circular movements until it reaches the necessary velocity and altitude to take off and begin energy production. This is the type of TOL system that will be used in this work, and more information on this topic can be found here. (2.8).

Figure 2.2 shows different types of AWES, classified according to their operational and structural characteristics.

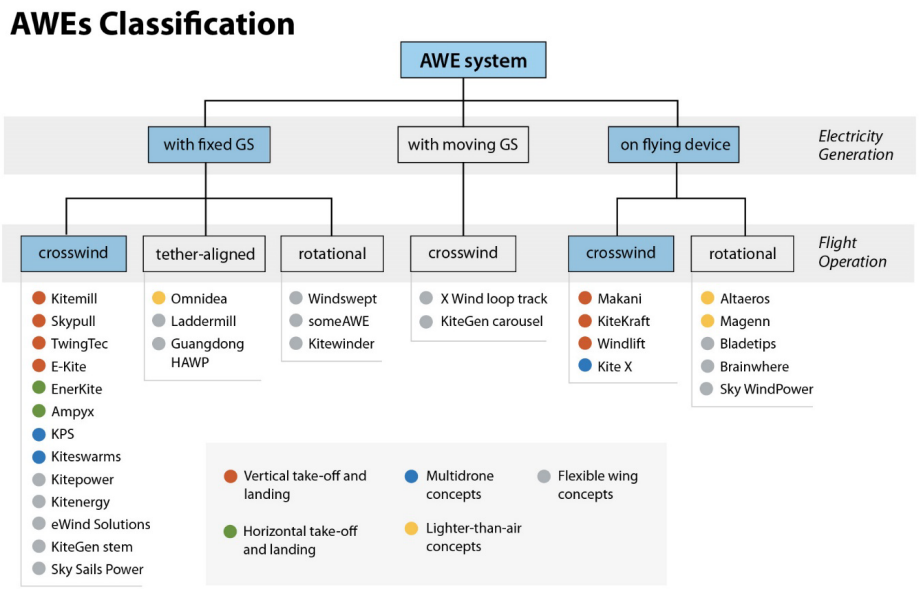


Figure 2.2: AWES Classification [3].

2.3 Fly-gen systems

In Fly-Gen AWES, electrical energy is generated on board the aircraft during flight and transmitted to the ground through a specialized tether that incorporates electrical cables. This energy conversion is achieved using one or more specially designed wind turbines, which can be categorized based on their flight principles. These principles include:

- Wings lift: Achieved through the tethered flight of specialized gliders (Figure 2.3a).
- Frames with multiple wings (Fig2.3b).
- Buoyancy and static lift: Achieved with aerodynamically shaped aerostats filled with lighter-than-air gas (Figure 2.3c).
- Rotor thrust: Achieved with the same turbines used for electrical power generation (Figure 2.3d).

The aircraft in Figures 2.3a and 2.3b fly crosswind and harness the relative wind, while the aircraft in Figures 2.3c and 2.3d fly directly into the wind and harness the absolute wind.

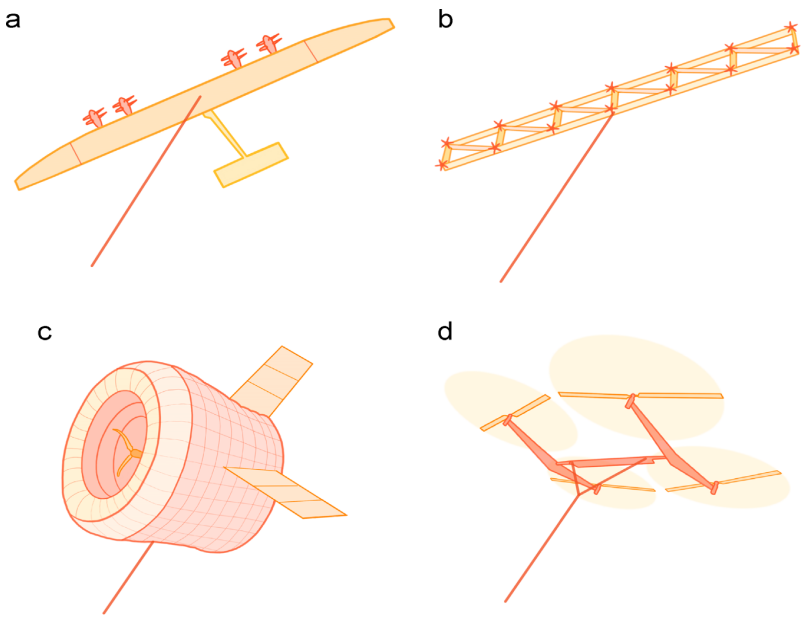


Figure 2.3: Different types of aircrafts in Fly-Gen systems [4].

One of the most prominent companies working on fly-generation systems was Makani Power Inc. Approximately twenty-five years after Loyd's seminal work, the company began developing prototypes of its Airborne Wind Turbine (AWT), as illustrated in Figure 2.3a.

Makani dedicated 13 years to designing and constructing kites capable of harnessing wind energy, aiming to develop an innovative method to expand wind power accessibility worldwide. Founded by kitesurfers in 2006, the team initially tested their airborne wind power concept using soft kites before progressing through eight iterations of rigid kite designs [12]. This development culminated in the M600 energy kite (Figure 2.4), a prototype utility-scale system with a 28-meter wingspan, equipped with eight turbines, each featuring five propeller blades, capable of generating 600 kW.

Figure 2.4: M600 Energy Kite by Makani Power Inc. [5].

2.4 Ground-gen systems

GG systems have become a primary area of research in airborne wind energy. In GG systems, energy is generated on the ground by harnessing tether tension through reel-in/reel-out motion.

Advanced control models and computational simulations are used to manage and stabilize the aircraft throughout the process, optimizing energy production and system efficiency.

Among GG-AWESs we can distinguish between fixed-groundstation devices, where the ground station is fixed to the ground and moving-ground-station systems, where the ground station is a moving vehicle.

- Moving-ground-station GG-AWEs: Generally more complex systems that aim at providing an always positive power flow which makes it possible to simplify their connection to the grid The generation takes place thanks to the traction force of ropes that induces the rotation of a generator that exploits the ground station movement rather than the rope winding mechanism.
- Fixed-ground-station GG-AWEs: Unlike the moving ground station, which generates energy through the motion of the station itself, this system uses a rope winding mechanism. It harnesses the torque generated by the aircraft as it is exposed to wind forces, converting this mechanical energy into electricity at the ground station.

The most common configuration is the combination of a fixed ground station and crosswind operation. For continuous electricity generation, these systems alternate between a tether reel-out phase and a reel-in phase, known as the 'pumping cycle.' This process is similar to the thermodynamic cycle of a reciprocating engine, with the key difference being that energy is generated during the pulling phase, and the cycle duration is much longer.

Fixed-ground GG-AWES stations are extensively researched by both private companies and academic laboratories. Figure 2.5 shows the energy conversion process consisting of a two-phase cycle: a generation phase, where electrical energy is produced, and a recovery phase, where a smaller amount of energy is consumed.

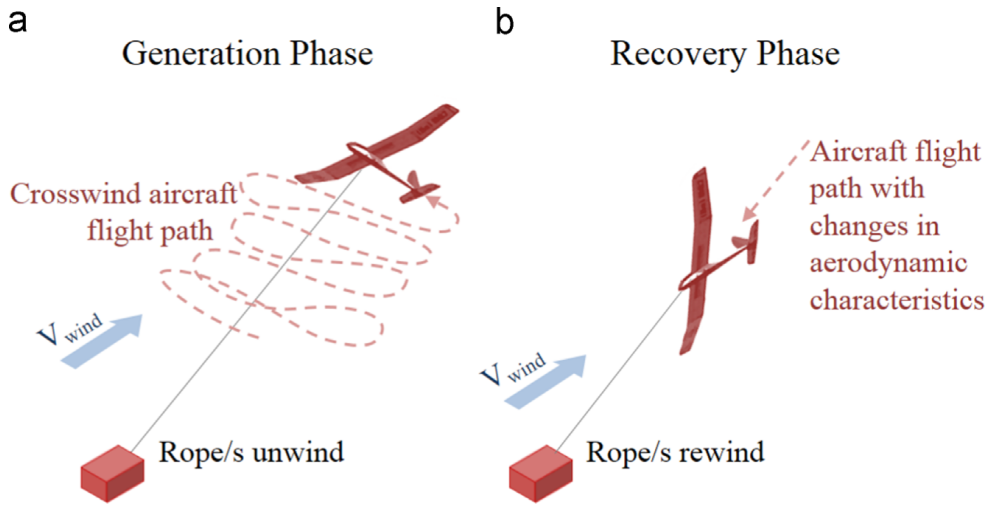


Figure 2.5: Differences in Generation phase (a) and Recovery phase (b) [4].

The time evolution of the mechanical power over two cycles is illustrated in Figure 2.6 (left), where the blue area represents the energy generated during reel-out, and the red area indicates the energy consumed during reel-in. The dashed line shows the average power. Power fluctuations during reel-out are caused by the interaction between strong turbulence in the simulated wind field and the compensating response of the control system.

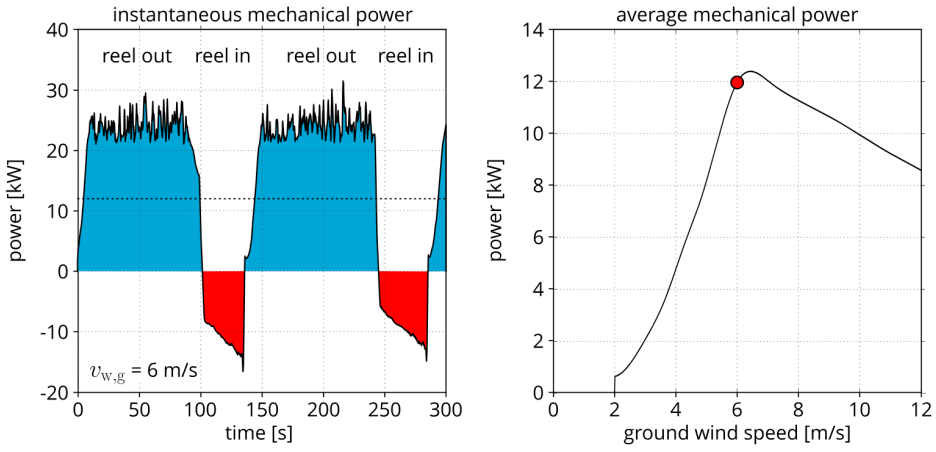


Figure 2.6: Instantaneous and average mechanical power on a fixed ground system station [2].

This will be the method used in this work, along with the Automatic Take-Off and Landing (ATOL) technique 2.7.

2.5 Kite Model

The position and motion of a tethered aircraft can be represented in spherical coordinates (r, ϕ, β), where r is the radial distance equal to the tether length, ϕ is the azimuth angle, and β the elevation angle and they have a coordinate basis denoted by $(\mathbf{e}_r, \mathbf{e}_\phi, \mathbf{e}_\beta)$ [6,9]. In spherical coordinates, the position, velocity, and acceleration vectors are, respectively:

$$\begin{aligned} \mathbf{p} &= \begin{bmatrix} r \\ \phi \\ \beta \end{bmatrix}, \quad \dot{\mathbf{p}} = \begin{bmatrix} \dot{r} \\ r\cos\beta\,\dot{\phi} \\ r\dot{\beta} \end{bmatrix}, \\ \ddot{\mathbf{p}} &= \begin{bmatrix} \ddot{r} \\ r\cos\beta\,\ddot{\phi} \\ r\ddot{\beta} \end{bmatrix} + \begin{bmatrix} -r\dot{\beta}^2 - r\dot{\phi}^2\cos^2\beta \\ 2\dot{r}\dot{\phi}\cos\beta - 2r\dot{\phi}\dot{\beta}\sin\beta \\ 2\dot{r}\dot{\beta} + r\dot{\phi}^2\cos\beta\sin\beta \end{bmatrix}, \end{aligned} \tag{2.1}$$

The last term in the acceleration $\ddot{p}$ is due to the use of acceleration a.

The rotating frame is given by:

$$\mathbf{a} = -\frac{\mathbf{F}_i}{m}, \tag{2.2}$$

The inertial force Fi acting in the rotating frame is given by:

$$\mathbf{F}_i = m\mathbf{a}, \tag{2.3}$$

Substituting, we find:

$$\mathbf{F}_i = m \begin{bmatrix} r\dot{\beta}^2 + r\dot{\phi}^2\cos^2\beta \\ -2\dot{r}\dot{\phi}\cos\beta + 2r\dot{\phi}\dot{\beta}\sin\beta \\ -2\dot{r}\dot{\beta} - r\dot{\phi}^2\cos\beta\sin\beta \end{bmatrix}, \tag{2.4}$$

It is also convenient to consider the body coordinate frame with basis vectors $(\boldsymbol{e}_x, \boldsymbol{e}_y, \boldsymbol{e}_z)$, where:

- $\boldsymbol{e}_x$ is the longitudinal axis of the aircraft pointing forward toward its nose.
- $\boldsymbol{e}_y$ is the transverse axis pointing outward from the right wing.
- $\boldsymbol{e}_z$ is the vertical axis of the aircraft that pointing downward from its belly.

To represent the position, motion, and orientation of the kite, the UPWIND team assumed a point-mass model (Figure 2.7).

This model is defined within a global Cartesian coordinate system with basis vectors $(\boldsymbol{e}_x, \boldsymbol{e}_y, \boldsymbol{e}_z)$.

A local spherical coordinate system with basis vectors $(\boldsymbol{e}_r, \boldsymbol{e}_\phi, \boldsymbol{e}_\beta)$ A body-fixed coordinate system (B) with basis vectors $(\boldsymbol{e}_X, \boldsymbol{e}_Y, \boldsymbol{e}_Z) = (\boldsymbol{e}_{\text{roll}}, \boldsymbol{e}_{\text{pitch}}, \boldsymbol{e}_{\text{yaw}})$.

The plane τ is defined as tangent to the position of the kite $\boldsymbol{p}$ as it moves along a spherical surface with a radius equal to the length of the tether r.

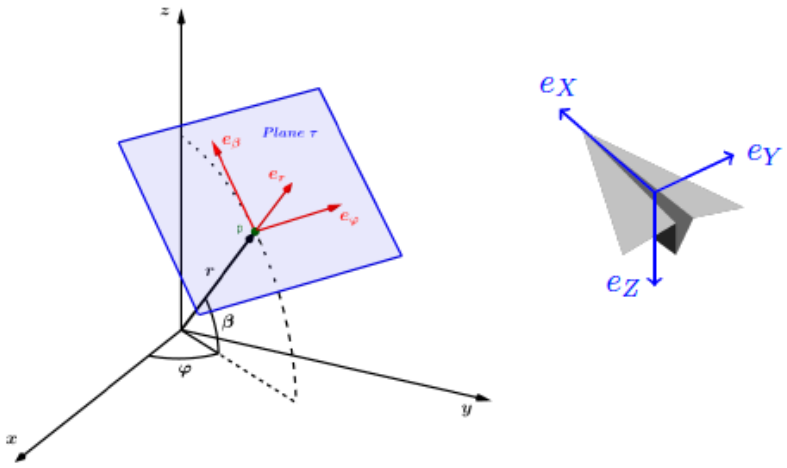


Figure 2.7: The spherical $(\boldsymbol{e}_r, \boldsymbol{e}_\phi, \boldsymbol{e}_\beta)$, global $(\boldsymbol{e}_x, \boldsymbol{e}_y, \boldsymbol{e}_z)$, and body $(\boldsymbol{e}_X, \boldsymbol{e}_Y, \boldsymbol{e}_Z) = (\boldsymbol{e}_{\text{roll}}, \boldsymbol{e}_{\text{pitch}}, \boldsymbol{e}_{\text{yaw}})$ coordinate frames [6,9].

2.5.1 Acting Forces

The UPWIND Team utilizes a set of angles to characterize the aircraft's attitude (Figure 2.8). The pitch angle, denoted by θ, is the angle between the aircraft's longitudinal axis $\boldsymbol{e}_X$ and the horizontal plane. The flight-path angle, denoted by γ, is the angle between the horizontal plane and the aircraft's velocity vector $\boldsymbol{V}$. The angle-of-attack, denoted by α, is the angle between the aircraft's longitudinal axis and its velocity vector when both the sideslip angle and wind speed are considered to be zero, as is the case in this work.

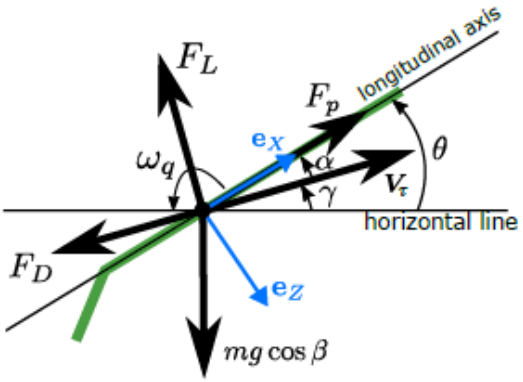


Figure 2.8: Acting forces for the circular tethered flight [6, 9].

Newton's second law, considering the mass of the kite m and the total force acting on it $\mathbf{F}$, can be expressed as:

$$m\ddot{\mathbf{p}} = \mathbf{F}, \tag{2.5}$$

In Equation (2.6), $\mathbf{F}$ can be decomposed into three components: $\mathbf{F}_{\text{th}}$, $\mathbf{F}_{\text{grav}}$, an $\mathbf{F}_{\text{aer}}(a)$. Thus, we have:

$$\mathbf{F} = \mathbf{F}_{\text{th}} + \mathbf{F}_{\text{grav}} + \mathbf{F}_{\text{aer}}(a), \tag{2.6}$$

Figure 2.9 shows how these forces interact with each other.

$\mathbf{F}_{\text{th}}$ is entirely aligned with the first coordinate of the local coordinate system, while $\mathbf{F}_{\text{grav}}$ is fully aligned with the last coordinate of the global coordinate system.

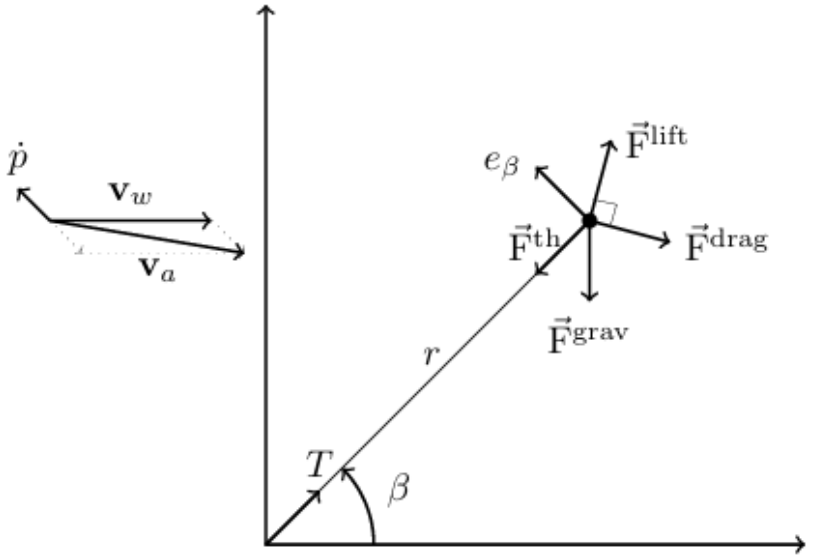


Figure 2.9: Forces acting on the Kite [7].

2.6 Crosswind flight

One of the most important reasons why AWESs are so attractive is their theoretical capacity to achieve the megawatt scale with a single plant. This achievement is made possible by a maneuver known as crosswind flight. In contrast to a kite that moves solely along the direction of its tether, crosswind flight involves wind blowing perpendicular to the object's trajectory. This technique enables greater power harvesting, depending on different lift-to-drag ratios (Figure 2.10).

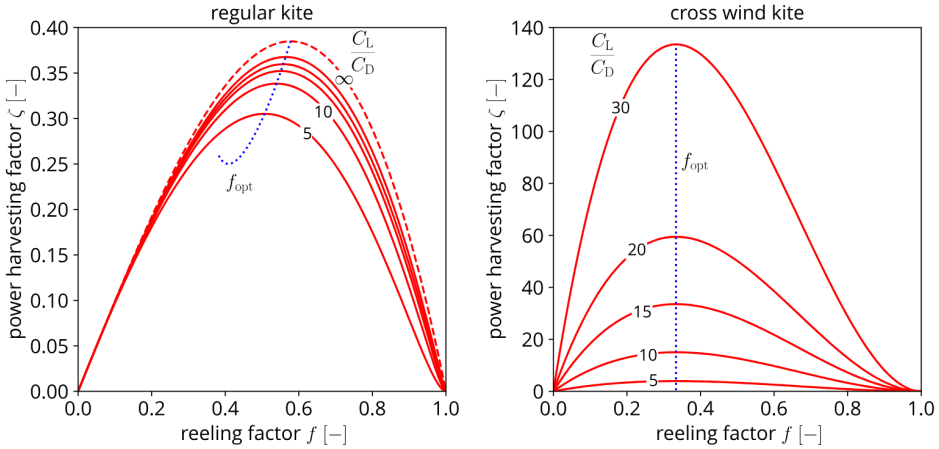


Figure 2.10: Values of ζ in a regular kite versus with crosswind maneuver [2].

Looking at the diagrams above and knowing the equations:

$$\zeta = (P/PwS), \tag{2.7}$$

$$f = vt/vw, \tag{2.8}$$

$$G = \frac{L}{D} = \frac{CL}{(CDw + CDce)}, \tag{2.9}$$

$$P = \frac{1}{2}\rho(Vwcos\theta)^3 \frac{4}{27} G^2 CLA, \tag{2.10}$$

Given that Equation (2.7) represents the power harvesting factor, Equation (2.8) the reeling factor, Equation (2.9) the lift-to-drag ratio and Equation (2.10) the optimal output power, we can conclude that the higher the value of ζ, the greater the output power relative to Pw (pulling power)[2]. Assuming the same area S, this increase in ζ directly improves the output power.

Analyzing figure 2.10, we observe that with a crosswind maneuver, ζ values can reach significantly high levels (around 140), whereas for a standard kite, these values are considerably lower (around 0.40).

Crosswind generation has demonstrated the capability to provide power one to two orders of magnitude greater than non-crosswind generation

AWESs concepts that exploit crosswind power have therefore a strong competitive advantage over non-crosswind concepts in terms of available power and, therefore, in the economics of the whole system.

2.7 Automatic Take off and Landing (ATOL)

A scalable, robust, and cost-efective AWES should rely on a completely autonomous operation, including a fully automatic launch and landing scheme. Most existing technology demonstrators still rely on supervised operation,especially in the take-off and landing phases, and are therefore not fully autonomous.

To achieve fully autonomous operation it is crucial to develop reliable ATOL schemes for tethered aircraft.[3]

Depending on the characteristics of the aircraft, different ATOL approaches can be adopted. Among several AWES using fixed-wing, tethered and motorized aircraft, we can distinguish three methods: linear, vertical and circular approaches. Figure (2.11) shows the peak power and additional mass of each ATOL scheme.

- VTOL - The VTOL scheme presents distinct advantages, particularly in terms of the minimal ground area required, the ease of relaunch, and the potential for reusing existing technologies, such as those developed for drones. However, these benefits come at the cost of significantly higher peak power requirements and additional on-board mass. This increased demand arises from the necessity for on-board motors to generate lift that exceeds the aircraft's weight.

- HTOL - HTOL and CTOL require the on-board motors primarily to overcome aerodynamic drag, resulting in reduced power demands. HTOL and CTOL schemes, on the other hand, necessitate considerably larger ground areas. For HTOL, the aircraft must achieve sufficient take-off speed to generate the required lift within the length of the runway. During landing, a robust control system is essential to ensure touchdown occurs within a predefined area, adding complexity to the design.

- CTOL - The CTOL scheme mitigates some of these challenges due to the effectively infinite nature of the circular runway, which eliminates the constraints imposed by fixed runway lengths. However, CTOL exhibits a significantly lower potential to take potencial of existing technologies, as its unique operational requirements differ substantially from conventional systems.

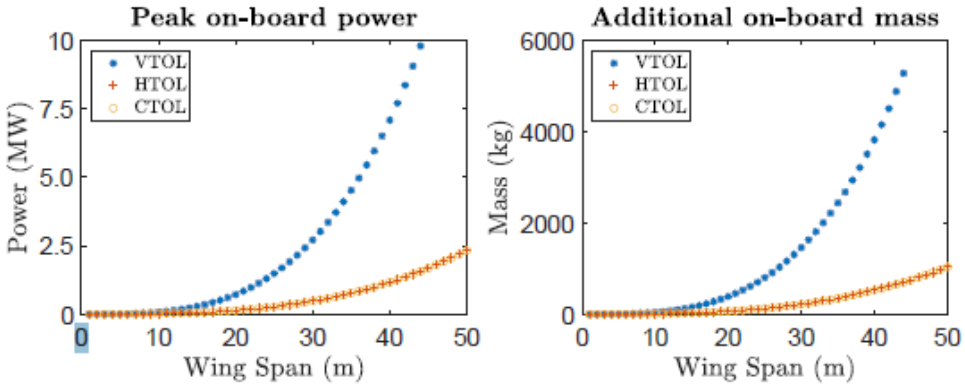


Figure 2.11: The on-board peak power and additional mass for each ATOL scheme [8].

This work will use the Circular Takeoff and Landing (CTOL) technique 2.8, a promising and yet underexplored approach.

2.8 Circular Takeoff and Landing (CTOL)

This study focuses on one of the least explored ATOL techniques, the CTOL method (Figure 2.12), which is identified as possessing several promising and innovative features.

We consider a self–propelled fixed–wing kite connected to the ground station via a flexible and constantly taut tether where the take-off and landing are performed with a circular motion approximately in an horizontal plane, with constant tether length [9,16].

This technique is divided into three steps:

- 1) Acceleration - The kite accelerates until it reaches a predefined reference speed.
- 2) Rotation - The kite tilts upward until it achieves a specified reference pitch angle.
- 3) Climbing - The kite's speed and pitch angle are controlled during ascent until it reaches a target altitude.

These steps must consider several factors, including the elevation angle that the tethered plane can maintain for the current length of the tether.

The landing procedure consists of analogous sub-phases in reverse order. The kite first executes a controlled descent glide until reaching a predefined altitude proportional to its wingspan. Subsequently, it tilts upward through a flare maneuver before making contact with the ground.

Complementing this technology, an hierarchical control architecture developed by the UPWIND team features a high-level layer comprising a discrete-event system (Figure 2.13). This layer integrates a path planner and a supervisory controller, which monitors the system and provides reference inputs to the lower-level modules, which will be thoroughly analyzed in this Section 2.9.

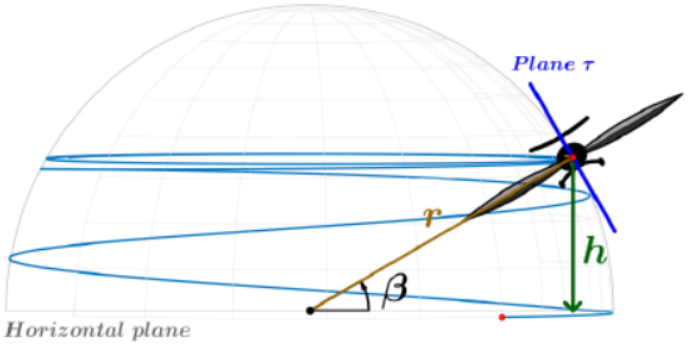


Figure 2.12: Circular trajectory during take off (in blue) [9,16].

2.9 Kite Module control architecture for a Circular Takeoff and Landing

The UPWIND team propose a control architecture for a Circular Takeoff and Landing (CTOL) of a rigid-wing tethered aircraft [9].

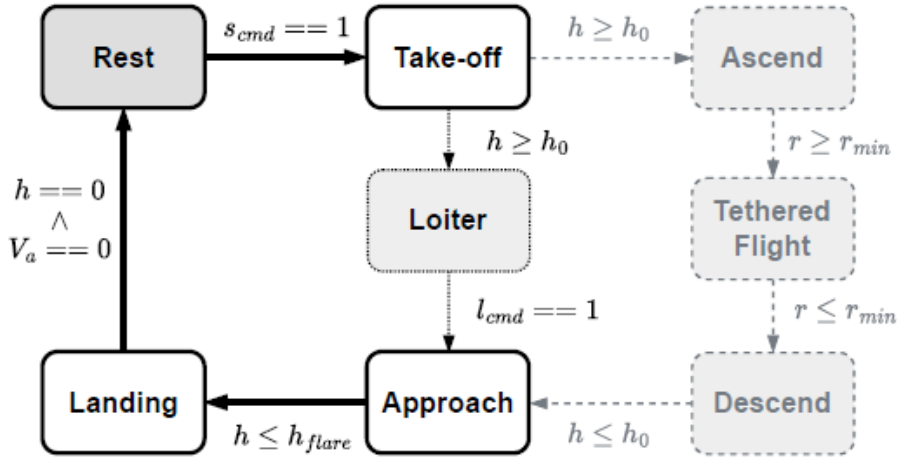


Figure 2.13: Overall control strategy of a CTOL architecture [9].

This architecture is divided into the high level Supervisory controller and the lower level Airspeed and Altitude Controllers.

2.9.1 Supervisory Control

In this phase it was developed a Supervisory Controller responsible for managing phase transitions, performing path-planning, and providing reference signals to the lower-level controllers during each phase of operation. The diagram on Figure 2.13 represents the various phases of operation

and the conditions under which the system transitions between them. Since the focus of this work is on developing a controller for the Take-Off and Landing (TOL) phases especially the CTOL phase, only results and simulations related to unblurred steps in Figure 2.13 will be discussed.

Therefore these supervisory control will use 4 steps (Rest Take-off Approach and Landing) with an additional step called Loiter connecting Take-off and Landing phases, where the kite flies in circles with a constant altitude and tether length.

In The Rest phase, the system remains idle, awaiting either an operator command or favorable wind and weather conditions for energy production. Once these criteria are met, the system transitions to the next phase.

The Take-off step comprises three sequential sub-phases.

These sub-phases guide the kite from a stationary position on the ground to a stable level flight, preparing it for ascent and subsequent electricity generation.

> 1) Acceleration - the kite will increase its speed while on the ground until it reaches a predefined speed vrot. 2) Rotation - the kite will tilt upwards until it reaches a reference pitch angle θrot. This will confer to the kite its maximum lift and move it to the Initial Climb phase. 3) Initial Climb - the kite will maintain its speed and pitch angle while increasing its altitude.

Moreover we have the Loiter phase. As mentioned earlier, this phase transitions the kite to flying in circles at a constant altitude and with a fixed tether length, maintaining this state until $lcmd = 1$, at which point the controller advances to the next phase.

During the Approach phase, the aircraft follows a circular and nearly horizontal flight path with a fixed tether length. This phase is divided into 2 sub-phasess:

- 1) Decelerate to Glide - Initially, it decelerates to a predefined velocity, $Vglide$, while maintaining its altitude.
- 2) Glide - Subsequently, it transitions into a gliding phase, during which its altitude gradually decreases until reaching a specified minimum altitude, $hflare$.

Finally, the Landing step is reached. This phase has also 2 sub phases:

- 1) Flare - The aircraft performs a flare maneuver, where its angle of attack increases, and its speed decreases just before touchdown.
- 2) Decelerate to Rest - Once on the ground, the aircraft continues to decelerate until it comes to a complete stop in a resting position.

These phases and sub-phases are epicted in Figure2.14 and Table 2.1.

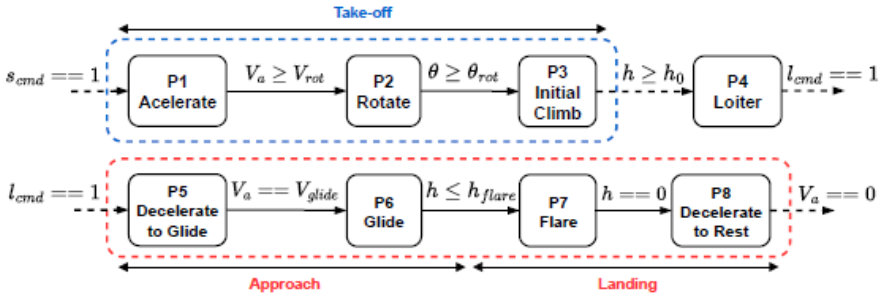


Figure 2.14: Control phases and sub-phases of the supervisory control [9].

2.9.2 Altitude and Airspeed Controllers

Table 2.1: Phases, Target States, and Controllers [9].

Phase - Name	Target State (conditions)	Controller
P1 - Accelerate	$V_a = V_{\text{rot}}, \theta = 0$	2 PID
P2 - Rotate	$V_a = V_{\text{rot}}, \theta = \theta_{\text{rot}}$ $(\dot{\gamma}, \dot{\alpha} \geq 0, \beta \geq 0)$	2 PID
P3 - Initial Climb	$V_a = V_{\text{climb}}, \gamma = \gamma_{\text{climb}}$ $\alpha = \alpha_L$	LQR
P4 - Loiter	$\beta = \beta_0, V_a = V_{\text{loiter}}$ $\gamma = 0, \theta = 0, \alpha = \alpha_0$	LQR
P5 - Decelerate to Glide	$V_a = V_{\text{glide}}, \gamma = 0$ $(\dot{\alpha} \geq 0)$	2 PID
P6 - Glide	$V_a = V_{\text{glide}}, \gamma = \gamma_{\text{glide}}$ $\alpha = \alpha_{\text{glide}}$	LQR
P7 - Flare	$\theta = \theta_{\text{flare}} (F_p = 0)$	PID
P8 - Decelerate to Rest	$V_a = 0 (F_p = 0)$	–

The primary goal of the ATOL control strategy is to ensure that the kite's airspeed, altitude, and attitude consistently follow the predefined reference values specified for each phase of operation.

To achieve that, the UPWIND Team uses two different control methodologies.

In the phases where the primary objective is to induce rapid changes in a specific variable—namely P1, P2, P5, and P7, the UPWIND Team used PID controllers due to their simplicity and proven competence in effectively achieving the desired performance.

PID controllers are used to control the speed V_a by acting on the propulsion force F_p, or to adjust the attitude (measured by θ or γ) by acting on the pitch rate ω_q.

In phases P1, P2, and P5, two PID controllers are employed with the references (target values) provided in Table 2.1.

In phase P7-Flare, only one PID controller is used to control the attitude; the reduction in speed occurs naturally by setting
($F_p = 0$ and by the increase in drag due to the higher Angle of Attack (AoA).

In phases P3, P4, and P6, they wanted to maintain several state components near their steady state, and therefore we use a multivariable controller, a LQR, computed for the system linearized around a steady midpoint of the trajectory in each phase.

In the LQR controllers, they start by defining, for each phase P3, P4, and P6 , a steady-state

$$\mathbf{x}_{\text{ref}} = (\beta_{\text{ref}}, V_{a,\text{ref}}, \gamma_{\text{ref}}, \theta_{\text{ref}}),$$

which serves as the linearization point and the reference state.

It was also necessary to define the corresponding steady-state control:

$$\mathbf{u}_{\text{ref}} = (F_{p,\text{ref}}, \omega_{q,\text{ref}}).$$

In phases P3 and P6, it was set the climb or glide angles, γ_{climb} or γ_{glide}, respectively. It was also set $\alpha = \alpha_L$, $\beta = \beta_{\text{med}}$ (an intermediate value along the trajectory of those phases). Under these conditions,V_a was determined such that it satisfies the steady-state conditions $(\dot{V}_a, \dot{\gamma}, \dot{\theta}) = (0,0,0)$.

During phase P4, the UPWIND team found the speed V_a satisfying $(\dot{\beta}, \dot{V}_a, \dot{\gamma}, \dot{\theta}) = (0,0,0,0)$ for $(\beta, \alpha, \gamma, \theta) = (\beta_0, \alpha_0, 0, 0)$.

As usual in LQR, they consider the error state $\tilde{x} = x - x_{\text{ref}}$ and the error control $\tilde{u} = u - u_{\text{ref}}$. The linearized model is $\dot{\tilde{x}} = A\tilde{x} + B\tilde{u}$, where:

$$A = \left.\frac{\partial f(\tilde{x},\tilde{u})}{\partial \tilde{x}}\right|_{\tilde{x}=0,\,\tilde{u}=0}, \qquad B = \left.\frac{\partial f(\tilde{x},\tilde{u})}{\partial \tilde{u}}\right|_{\tilde{x}=0,\,\tilde{u}=0}. \tag{2.11}$$

The objective function is

$$\int_0^\infty (\tilde{x}^\top Q \tilde{x} + \tilde{u}^\top R \tilde{u})\, dt \tag{2.12}$$

where the matrices Q and R are diagonal, initially defined using Bryson and Ho's rule, and then manually adjusted to achieve the desired response.

In the different flight phases, two main angles-of-attack (AoA) are considered: the steady flight AoA α_0, in which the ratio c_L/c_D is maximized, and the maximum lift AoA α_L for which c_L is maximized. While the steady flight AoA α_0 is used in phase P4 (and also in tethered flight energy producing phases), the maximum lift AoA α_L is used in phases P3 and P6. During phase P2, P5, and P7, the AoA will vary. During P2, they aimed for a pitch angle $\theta_{\text{rot}} = \alpha_L - \alpha_0$ and the speed V_{rot} is selected in such a way that the lift with AoA α_0 is not enough to overcome the aircraft weight, but the lift with AoA α_L can elevate the aircraft ($F_L(\alpha_0, V_{\text{rot}}) < mg < F_L(\alpha_L, V_{\text{rot}})$).

.

In phase P8, since it did not have any active braking system, it was simply set the propulsion to zero and wait for the drag and friction to stop the aircraft.

2.10 Auxiliary Vehicle

The CTOL testing platform described in this work consists of a ground vehicle connected to a central rotary mechanism. This system was developed and patented by the UPWIND group [10,19].

This innovative support vehicle concept utilizes an electromagnet as the connection point between the kite and the support car. This system fulfills two primary purposes: propelling the kite during its acceleration take-off phase and serving as the attract point for the kite during landing.

The current approach equips the support car with a two-rear-wheel drive configuration. During both take-off and landing phases, the tethered vehicle must complete several revolutions along a circular trajectory, maintaining a predefined linear velocity. To design a velocity controller for the car's dual-motor system, the following mathematical model is used:

$$\omega_c = \frac{v_{c,\max}}{R_{tc} D_c}, \qquad F_{tc} = m_c \frac{v_c^2}{R_{tc}}, \tag{2.13}$$

where:

- ω_c: Angular velocity of the car,
- $v_{c,\max}$: Maximum linear velocity of the car,
- R_{tc}: Tether length from the car chassis center to the central rotary mechanism,
- D_c: Motor driver control input (duty cycle),
- F_{tc}: Tension force exerted by the tether on the car,
- m_c: Mass of the car.

The tethered support vehicle presents several notable advantages. It simplifies the overall system by eliminating the need for aircraft landing gear, thus reducing weight.

Furthermore, the system is highly portable and can be easily deployed. The radius of the circular path can be adjusted as required, providing flexibility in operation.

Unlike rail-like structures, the system does not depend on fixed supports.

Velocity control during CTOL stages is straightforward, achieved through direct control inputs to the car motor drivers. Lastly, in offshore applications, a boat-like vessel could replace the ground vehicle.

The tethered support vehicle offers several advantages:

- Weight Reduction: Eliminates the need for aircraft landing gear, reducing the overall weight.
- Portability and Ease of Deployment: The system is highly portable and can be quickly set up.

- Adjustable Radius: The circular path radius can be modified according to operational requirements.
- No Fixed Supports: Unlike rail-like structures, the system does not rely on fixed supports, enhancing flexibility.
- Straightforward Velocity Control: During CTOL stages, velocity control is easily managed through direct inputs to the car motor drivers.
- Offshore Applicability: In offshore scenarios, the ground vehicle can be replaced with a boat-like vessel, maintaining functionality.

The support vehicle provides significant advantages for the CTOL strategy by simplifying system design, to demonstrate feasibility, identify potential failure modes, and guide the development of larger-scale models in the future.

A preliminary prototype vehicle was developed for the UPWIND system (Figure 2.15).

The assembly process began with designing the vehicle schematic, emphasizing the performance and functionality of its traction, structural, and electronic components. The car chassis and support structures were designed and then 3D-printed using a Creality Ender-5 Plus printer.

Similar to the kite prototype, this vehicle utilizes the same bridle system, primarily ensuring circular motion for the non-steering support vehicle.

The electronics of the support vehicle are organized into three main sub-circuits. The first sub-circuit is responsible for managing the power supply to a Raspberry Pi Zero 2W single-board computer (SBC). This circuit employs a pair of 3.7V batteries along with an LM2596 voltage step-down module. It processes input data, such as velocity measurements, and transmits control commands to other sub-circuits.

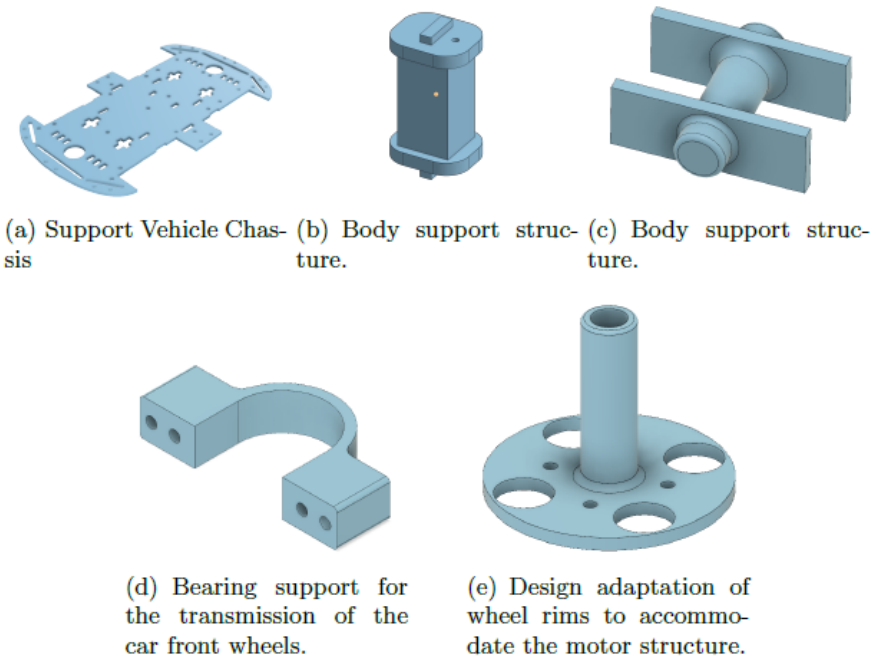
The second sub-circuit is dedicated to powering and safeguarding the electromagnetic lock system. This system includes 3.7V Lithium 18650 batteries, an XL6019 voltage step-up module, and a 5V relay to ensure reliable operation.

The third sub-circuit is designed exclusively for supplying and regulating power to the two rear-wheel motors. This is achieved using an L298N dual-motor driver module, which ensures efficient power delivery and motor control.

The total weight of the support vehicle, including all electronic components, is approximately 1.0 kg.

2.11 Hardware advances

The UPWIND team utilized a glider model called Multiplex Easystar II RTF.To ensure compatibility with this glider, the team selected the Multiplex RX-4 / 9 FLEXX receiver that operates using the SRXL (Serial Receiver Link) protocol [11].



(a) Support Vehicle Chassis

(b) Body support structure.

(c) Body support structure.

(d) Bearing support for the transmission of the car front wheels.

(e) Design adaptation of wheel rims to accommodate the motor structure.

Figure 2.15: Structural components designed for the tethered support vehicle prototype by the UPWIND Team [10].

This choice facilitated seamless communication between the radio control system and the Pixhawk 4 flight controller installed in the glider [10].

First, the radio was calibrated checking if it needed to reverse the signal in the servos. Channel 1,2,3 and 4 of the radio controller were assigned to Ailerons, Elevator, Rudder, and Throttle, respectively.The accelerometer and the GPS were also calibrated of the board.

In terms of battery, the team used the Roxxy 3S 20C EVO 2200 and it had a MPX connector, not compatible with the XT60 universal connector used, so XT60 - MPX and MPX - XT60 were necessary in the project.

The GPS used was the Holybro GPS Module and it was placed into the nose of the aircraft as far as possible of the motor.

The Autopilot used was the Holybro Pixhawk 4 and the Telemetry used for Real-Time Monitoring of the Kite Dynamics and Sensor Data Transmission was the Holybro Telemetry Kit 433MHz. All the hardware parts can be seen in the Figure 2.16:

Figure 2.16: Eletronic diagram of the vehicle (left) and the prototype model concept (right) by UPWIND Team [11].

2.12 Communications

In today's world, there are numerous forms of communication that are essential for the functioning of virtually all sectors and structures integrated across our planet.

Without communication, the world as we know it would simply not be possible. If, for any reason, certain communication systems were to be disrupted or disabled, it could bring daily life to a halt and potentially cause severe damage to multiple sectors.

Communication systems can generally be categorized into two main types:

- Wired Communication – This refers to networks that utilize physical cables to connect devices to the Internet or other networks.
- Wireless Communication – This refers to communication technologies that operate without physical wires, relying instead on electromagnetic waves (EM waves) or infrared signals. Wireless devices typically include antennas or sensors to facilitate connectivity.

The following Table 2.2 outlines the main advantages and disadvantages of each communication type:

Table 2.2: Comparison Between Wired and Wireless Communication [23].

No.	Wired Communication	Wireless Communication
1	Uses physical cables to connect devices	Uses electromagnetic or infrared waves to transmit data
2	Faster transmission speed	Slower transmission speed
3	Low propagation delay	High propagation delay
4	More secure and thus more reliable	Less secure and therefore less reliable
5	Devices must be hard-wired	Installation is quick and more flexible
6	Lower equipment cost	Higher equipment cost
7	High installation and maintenance cost	Low installation and maintenance cost

Within these two categories of communication, we can identify several key technologies, some of which are fundamental to the development of this work [23]:

For wired communications:

- Serial Communication (UART/USART) – A simple and widely used point-to-point communication method that operates on a full-duplex protocol.
- I^2C Communication – A master-slave communication protocol that operates over two lines (SDA and SCL), and works in half-duplex mode.
- SPI Communication – Another master-slave protocol, faster than I^2C, which supports full-duplex data transmission.
- Ethernet – A widely adopted communication protocol for local area networks (LANs), where data is transmitted in frames. It employs Carrier Sense Multiple Access with Collision Detection (CSMA/CD) to manage access and avoid data collisions in shared media.
- Others – Additional wired protocols may be used depending on system requirements.

For wireless communications:

- Radio Frequency (RF) – A low-cost, point-to-point communication method, although it is prone to electromagnetic interference.
- Wi-Fi – A high-speed wireless communication technology for local networks, providing internet connectivity and broader coverage.
- Bluetooth – A short-range wireless protocol commonly used in mobile devices, wearables, and sensors for low-power communication.
- Others – Other wireless technologies may be considered depending on the use case.

2.13 Synchronization Algorithms

To achieve harmonized synchronization of moving objects, several synchronization algorithms have been developed, each offering significant benefits for specific scenarios:

- Peterson's Algorithm – Ensures mutual exclusion between two processes that share a variable, without requiring hardware support. Each process uses a flag to indicate its intent to enter the critical section and a shared `turn` variable to indicate whose turn it is.
- Dekker's Algorithm – A similar algorithm to Peterson's, but in situations where both processes attempt to enter the critical section simultaneously, the process without the current turn voluntarily backs off temporarily.
- Test-and-Set – A hardware-supported atomic lock operation. It reads the current value of a lock and sets it to `true`, enabling the implementation of spinlocks and mutual exclusion mechanisms.

- Kuramoto Control – The algorithm employed in this work. It is particularly useful in distributed physical systems and models synchronization among oscillators with distinct natural frequencies. It achieves synchronization through parameters such as natural frequency, coupling angles, and a coupling coefficient.

Chapter 3

Car and Control System Model

3.1 Kuramoto Control

In this chapter, we present the new changes in the dynamic model of the ATOL system, with the agreement of the Upwind Team, with the addition of the car model to the overall model to obtain a synchronous movement between the car and the kite.

For that, we implemented an algorithm called Kuramoto Control which enforces the synchronization of the phase and acceleration between the two.

Since this work involves two agents with distinct natural frequencies, a robust synchronization method is required: one that ensures a sufficiently fast convergence, rendering the synchronization process virtually imperceptible, the Kuramoto control was chosen because it can synchronize different oscillators over time, even when they have different natural frequencies.

Initially, analyzing the Kuramoto model for a single node, we observe that the angle θ changes over time equals ω.

So we have:

$$\frac{d\theta}{dt} = \omega \tag{3.1}$$

By knowing:

$$\theta = \omega t + \theta(0) \tag{3.2}$$

$$\sin(\theta + 2\pi) = \sin(\theta) \tag{3.3}$$

We arrive at the desired equation for sinusoidal coupling, which relates the phase and frequency of the oscillator i by summing the coupling force between the oscillator I (the kite) and all other oscillators j (the car).

$$\frac{d\theta_i}{dt} = \omega_i + \frac{K}{N}\sum_{j=1}^{N}\sin(\theta_j - \theta_i), \quad i = 1,\ldots,N \tag{3.4}$$

- K – *coupling strength*

- N – *number of oscillators (nodes)*
- ω_i – *natural frequency of oscillator i*
- θ_i – *phase of oscillator i*

We highlight that the higher the k, the stronger the coupling force between the kite and the car. In this case, we only have two oscillators so that we can simplify the equation:

$$\frac{d\theta_1}{dt} = \omega_1 + \frac{K}{2}\sin(\theta_2 - \theta_1) \tag{3.5}$$

- θ_1 - phase of the kite
- θ_2 - phase of the car

With regard to the coupling term $\sin(\theta_2 - \theta_1)$, it is possible to extract some information:

- ($\sin(\theta_2$ - $\theta_1)$ $<$ 0) - if this sin is negative, it means the coupling term is negative and θ_1 is trying to catch θ_2 and θ_2 is trying to slow down.
- ($\sin(\theta_2$ - $\theta_1)$ $=$ 0) - if this sin is zero it means θ_2 is equal to θ_1 so the coupling term is zero and the two oscillators are not influencing each other in that moment.
- ($\sin(\theta_2$ - $\theta_1)$ $>$ 0) - if this sin is positive it means the coupling term is positive and θ_i is trying to catch θ_1 and θ_1 is trying to slow down.

However, in this case, the kite's velocity and phase are independent of the car's, which means that the car must follow the kite's movement and adjust its position along the circular trajectory.

So, simplifying the model with the equations:

$$\phi = \theta_2 - \theta_1, \quad \Omega = \omega_2 - \omega_1, \quad \frac{d\phi}{dt} = \frac{d\theta_2}{dt} - \frac{d\theta_1}{dt} \tag{3.6}$$

We obtain the following.

$$\frac{d\phi}{dt} = \Omega - K\sin(\phi) \tag{3.7}$$

By setting this phase-difference frame to zero, we obtain an equation that can be interpreted to understand the movement of the oscillator (the car) as it attempts to synchronize, through its fixed points.

$$\sin(\phi) = \frac{\Omega}{K} \tag{3.8}$$

Knowing that $\frac{\Omega}{K}$ must lie within the interval $[-1,1]$, we can evaluate the trajectory of the oscillator based on the value of $\sin(\phi)$ relative to the constant $\frac{\Omega}{K}$

- If $\sin(\phi) < \frac{\Omega}{K}$, then $\frac{d\phi}{dt} > 0$ - In this case, the car moves positively toward the fixed point, which corresponds to an attempt to catch up with the kite.

- If $\sin(\phi) = \frac{\Omega}{K}$, then $\frac{d\phi}{dt} = 0$ - At this point, the motion of the car and the kite is decoupled, indicating an instantaneous state of equilibrium between their relative phases.
- If $\sin(\phi) > \frac{\Omega}{K}$, then $\frac{d\phi}{dt} < 0$ - In this case, the car moves negatively with respect to the fixed point, which corresponds to a decelerating motion aimed at slowing down relative to the kite.

3.2 Visualization of the Kuramoto Control with the kite with constant velocity

To effectively visualize this behavior, the kite's velocity was assumed to be constant at 10 m/s. This assumption allows for a clearer analysis of the car's behavior relative to the kite's circular trajectory using MATLAB.

In Figure 3.1, two points are shown: a red point representing the car and a blue point representing the kite, positioned at a distance corresponding to the red underlined line segment.

Figure 3.2 and Figure 3.3 illustrate the kite (in blue) moving at a constant velocity, while the car exhibits a negative velocity in an effort to approach the kite.

From this, we can conclude that:

$$\sin(\theta_2 - \theta_1) < 0 \quad \text{and} \quad \sin(\phi) > \frac{Q}{K} \tag{3.9}$$

In Figure 3.4, both points are observed to be synchronized, moving along the circular trajectory at a speed equal to that of the kite. This indicates that synchronization between the car and the kite has been successfully achieved.

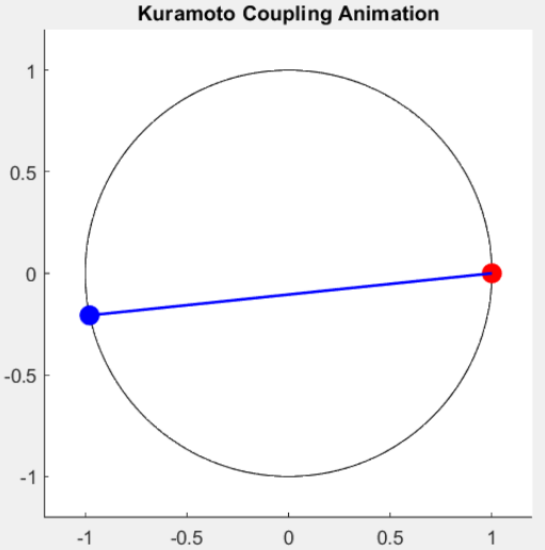


Figure 3.1: Initial phase difference between the cart and the kite.

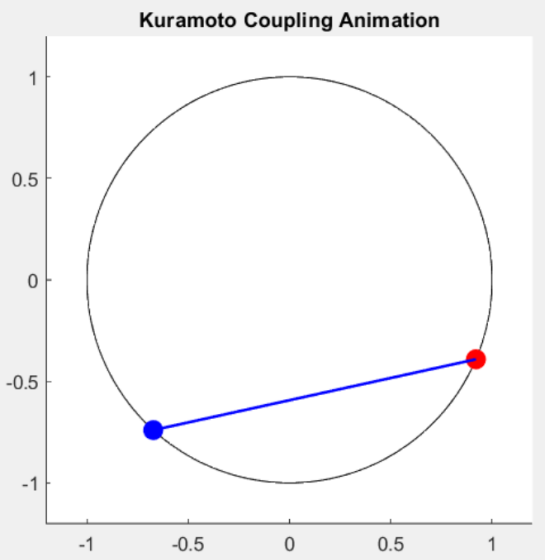


Figure 3.2: Coupling begins as the cart adjusts its phase to synchronize with the kite.

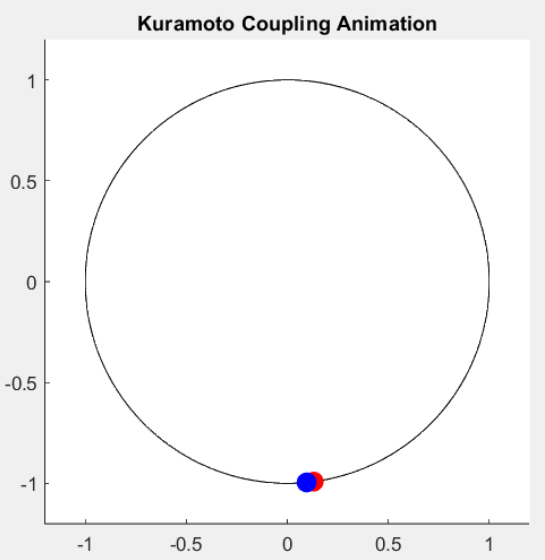


Figure 3.3: Cart nearly aligned with the kite, reducing the phase error.

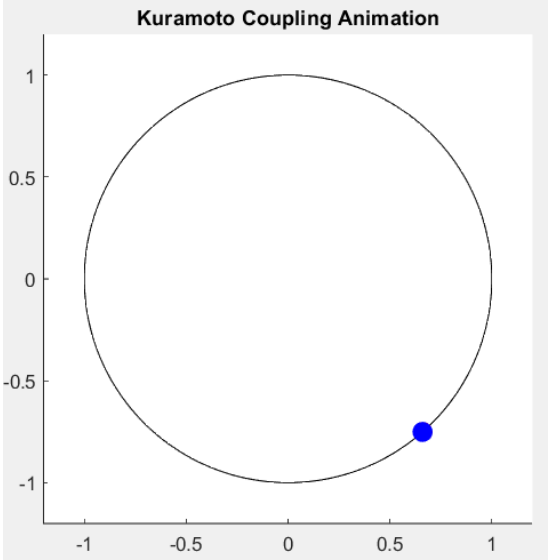


Figure 3.4: Full phase synchronization achieved between the cart and the kite.

3.3 New Model of the car in Simulink

To generate a signal capable of achieving the desired synchronization between the car and the kite, it was essential to develop a new Simulink model integrated with the existing ATOL model previously implemented by the UPWIND Team. This process required a thorough understanding of the acting forces, the kinematics of the system, its state-space representation, the kite model, and the different phases of the control strategy.

With the kite model and the control phases already described in [2.9], we were able to derive the kinematics, which are characterized by:

$$\begin{aligned} r\cos\beta\,\dot{\varphi} &= V_a\cos\gamma \\ r\dot{\beta} &= V_a\sin\gamma \\ \dot{\theta} &= \omega_q, \end{aligned} \tag{3.10}$$

And its derivatives:

$$\begin{aligned} r\ddot{\beta} &= \sin\gamma\dot{V}_a + V_a\cos\gamma\dot{\gamma} \\ r\cos\beta\,\ddot{\varphi} &= \cos\gamma\dot{V}_a - V_a\sin\gamma\dot{\gamma} \\ &\quad + \frac{V_a^2}{r}\tan\beta\cos\gamma\sin\gamma, \end{aligned} \tag{3.11}$$

Based on the kinematic equations and the acting forces reviewed in [9], we derived the state-space model that will be used to achieve synchronization between the kite and the car.

$$\dot{\mathbf{X}} = f(\mathbf{X},\mathbf{U}), \tag{3.12}$$

where:

$$\mathbf{X} = [\varphi,\beta,V_a,\gamma,\theta]^\top, \quad \mathbf{U} = [F_p,\omega_q]^\top, \tag{3.13}$$

Where the dynamic function f is characterized by:

$$\begin{bmatrix} \dot{\varphi} \\ \dot{\beta} \\ \dot{V}_a \\ \dot{\gamma} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{1}{r\cos(\beta)}V\cos\gamma \\ \frac{1}{r}V\sin\gamma \\ \frac{1}{mV}\left[-F_D + F_p\cos\alpha - mg\cos\beta\sin\gamma\right] \\ F_L\cos\zeta + F_p\sin\alpha - mg\cos\beta\cos\gamma \\ -\left(\frac{mV^2}{r}\right)\tan\beta\cos\gamma \\ \omega_q \end{bmatrix} \tag{3.14}$$

This dynamic function is fundamental in linking the kite's dynamic model with the car's. Through linearization, it is possible to obtain the necessary values to be used in the car's model, particularly:

- v_a, representing the kite's velocity (Figure 3.5),
- ϕ, characterizing the azimuth angle (Figure 3.6), which defines the kite's phase angle along its circular trajectory,
- β, representing the elevation angle of the kite (Figure 3.7),

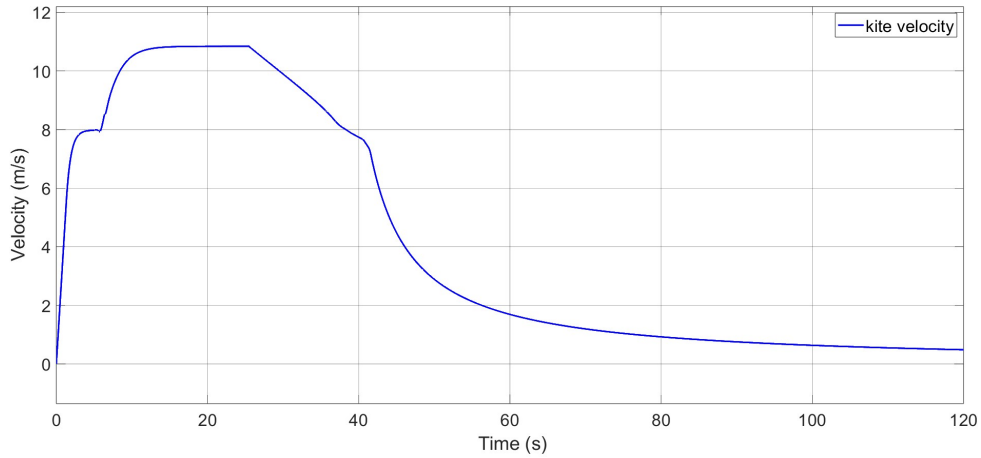


Figure 3.5: Kite´s velocity obtained by the Atol Model.

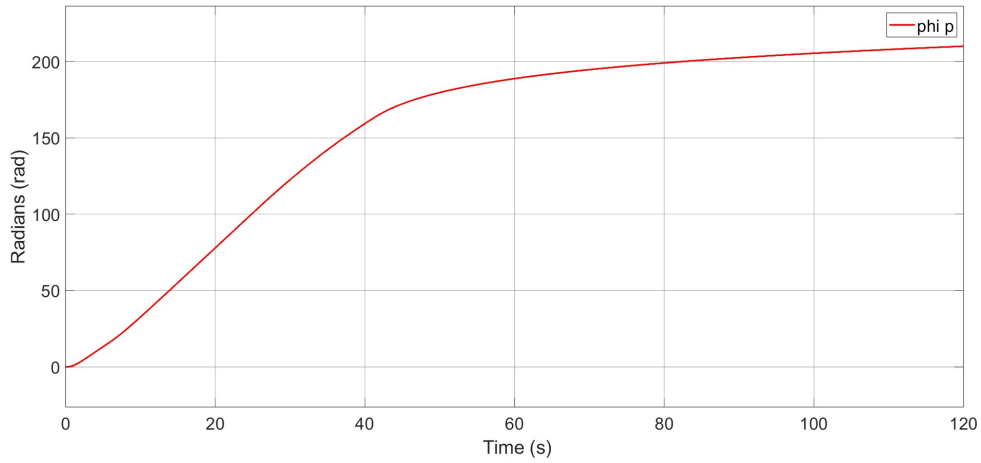


Figure 3.6: Azimuth angle obtained by the Atol model.

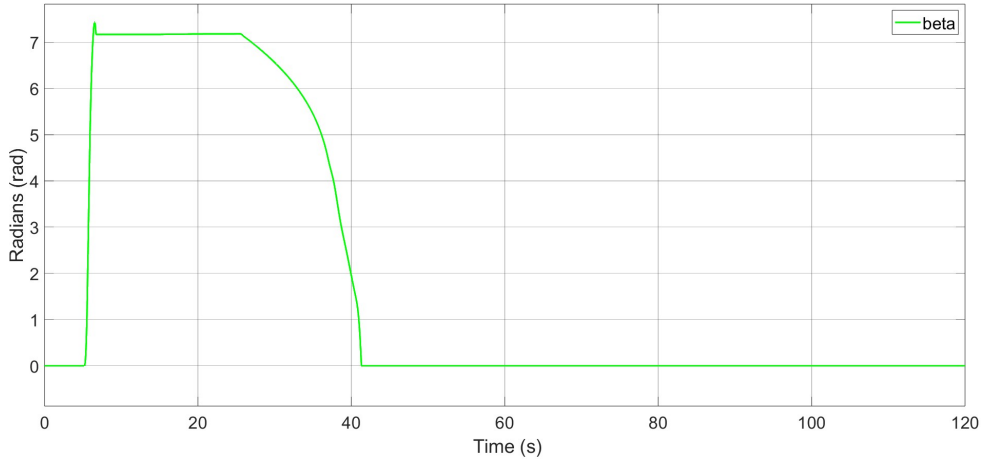


Figure 3.7: Elevation angle obtained by the Atol model.

These signals were then transmitted to the car model with the objective of integrating them as control parameters to define the car velocity.

To improve energy efficiency and preserve battery life, additional conditions were defined to stop the car when its movement was unnecessary. As a result, the reference velocity signal v_a was modified to implement these conditions, allowing the car to synchronize with the new desired speed while conserving energy (Figure 3.8):

Key Insight

- If state = 3, then $v_a = 0$ – In this state, the kite detaches from the car due to the deactivation of the electromagnet and begins its initial ascent. Since the car is no longer required to follow the kite during this phase, its motor can be shut down to save energy.
- If state = 4 and clock < 18, then $v_a = 0$ – In this phase, the kite continues to move along a trajectory at a constant velocity of 10.8 $\mathrm{m/s}$, and the car is not required to follow it. Therefore, the motor remains off. Only after 18 seconds does the car resume movement in an attempt to resynchronize with the kite. This 18 seconds delay allows the kite to stabilize on its circular trajectory and reach the desired velocity.
- If $\beta < 1.5$ rad and state $\neq 1$ or 2, then $v_a = 0$ – The angle β provides information about the kite's altitude via the relationship $h = \sin(\beta) \cdot r$. Knowing that the current state is neither 1 nor 2, we can infer that the kite is descending and close to the ground, within range of the electromagnet. This corresponds to the final phase of the trajectory, where the car and kite reunite through magnetic coupling. At this point, the car's motor can again be turned off.

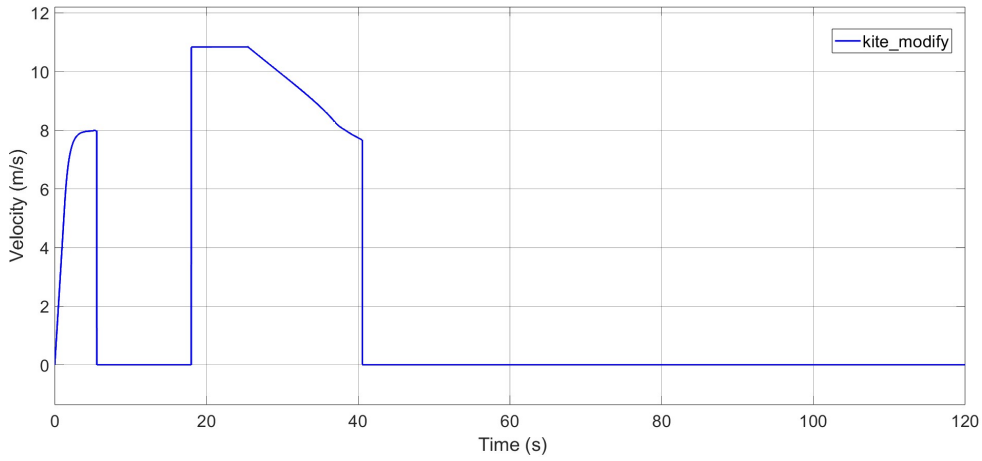


Figure 3.8: Kite´s velocity modified in order for the car to follow.

Next, the generated signal is processed through a PID controller in order to correct steady-state errors and mitigate potential overshoots (Figure 3.9 and Table 3.1) .

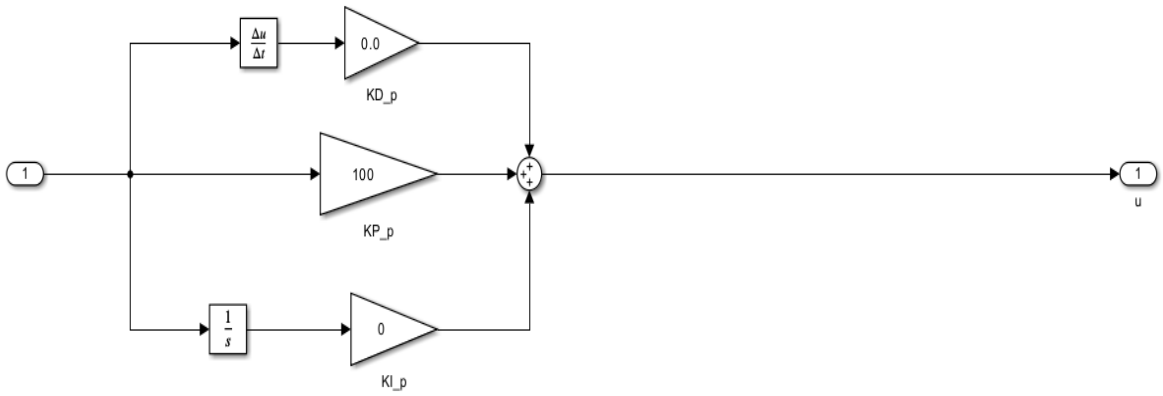


Figure 3.9: PID of the car´s model.

Table 3.1: PID gains used in the car's controller.

Gains of F_q PID	**Values**
K_{D_k}	0.0
K_{P_k}	100
K_{I_k}	0

As shown in Figure (3.8), the system initially produced an acceleration of approximately 4 $\mathrm{m/s^2}$, which is infeasible for our RC motor, whose maximum acceleration is limited to 2 $\mathrm{m/s^2}$.

To address this issue, two saturation blocks were added to the controller output in order to constrain the acceleration within physically acceptable limits. This makes it feasible to use the control signal in real-world testing.

Throughout this work, the decision was made to divide the saturation into two distinct blocks for specific operational reasons:

- Up to state 4, the cart uses the saturation block shown in Figure 3.11. During this phase, a fast deceleration is desired in order to minimize energy consumption, as the kite is flying independently and is not connected to the cart. In this configuration, there is no risk of the kite detaching due to a sudden deceleration of the cart.

- From state 4 onward, once the cart begins synchronizing its velocity with the kite, the deceleration is more controlled, following the kite's motion. Furthermore, in state 7 - the phase in which the kite and cart recouple - the cart's deceleration must be further limited to avoid an abrupt motion that could cause the kite to detach from the platform. In this phase, the saturation block shown in Figure 3.10 is used.

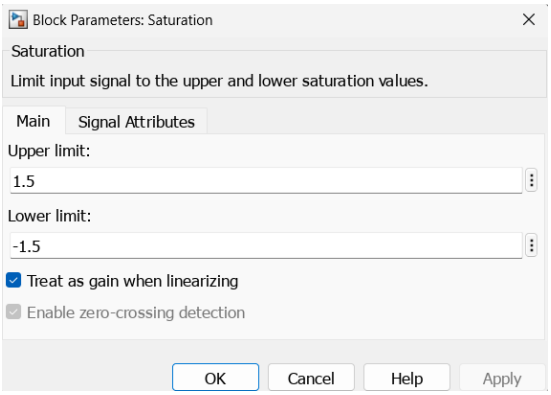


Figure 3.10: Saturation block configuration applied during states 1–4 of the control system.

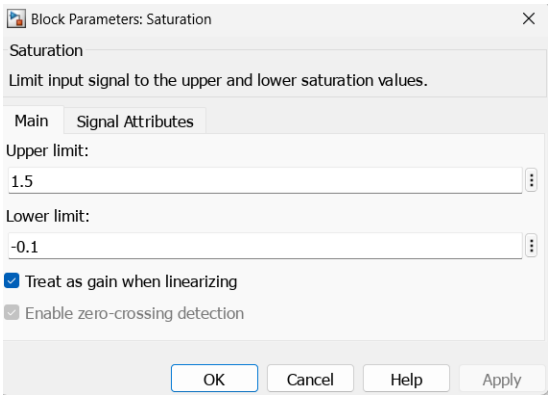


Figure 3.11: Saturation block configuration applied during states 4-8 of the control system.

Here, the negative acceleration is introduced to simulate the friction present in real-world tests, allowing the waveform in the simulation to more closely resemble actual behavior. However, this has no significant impact on the results during physical testing, considering that the motor is unable to operate in the reverse direction.

The control input u obtained from the two saturations blocks is then passed to the block implementing the Kuramoto-based control strategy. This block operates on a state vector defined as $x = [x_1 \;\; x_2 \;\; x_3]$, where:

- x_1 represents the fase angle of the car,
- x_2 represents the angular velocity of the car,
- x_3 represents the fase angle of the kite.

This formulation enables synchronization of the phase angle (i.e., azimuth angle) between the car and the kite using the equation:

$$\dot{x}_1 = x_2 + k_1 \sin(x_3 - x_1) \tag{3.15}$$

Furthermore, the synchronization of car's velocity with that of the kite by controlling the acceleration through the following equation:

$$\dot{x}_2 = u + k_2 \sin(x_3 - x_1) \cdot \omega_{\text{kite}} \tag{3.16}$$

Through these two equations, we obtain the outputs $\dot{x}_1$ and $\dot{x}_2$, which are each processed by an integrator. As a result, we obtain the updated signals x_1 and x_2, which respectively reflect the synchronization of the car's phase angle and angular velocity with those of the kite.

From this point, it was necessary to verify whether the synchronization had been successfully achieved. To do this, both signals were plotted together in a scope for analysis (figure 3.12).

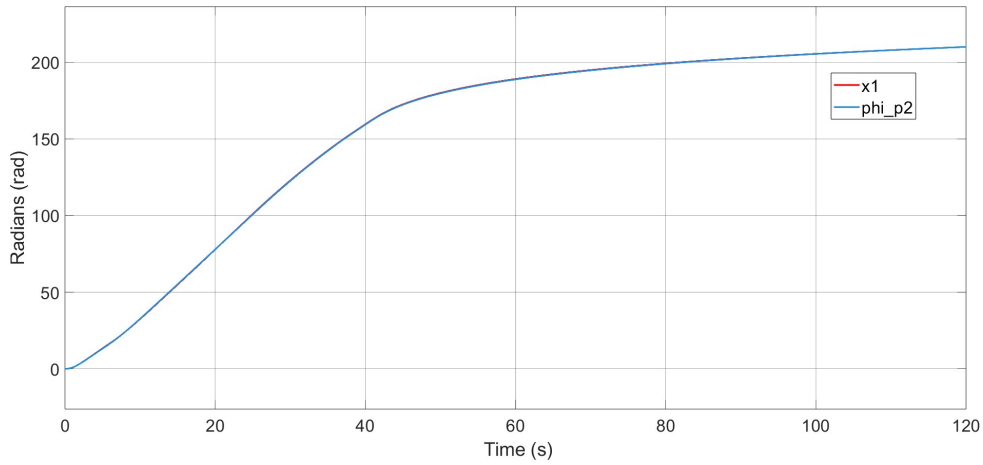


Figure 3.12: Azimuth angles of both car (red) and kite (blue).

From the figures, it can be observed that the phase angles synchronized as intended, achieving an almost ideal alignment. However, regarding the angular velocities, the presence of the saturation block after the PID controller, combined with the rapid increase in the kite's rotational speed, prevented the car from reaching the desired velocity in time to launch the kite into the air.

To address this issue, it was necessary to modify the PID parameters responsible for controlling the kite's velocity during state P1 (the acceleration phase) (figure 3.13 and table 3.2). The main objective was to smooth the kite's acceleration by increasing the time it takes to reach the desired rotational velocity.

This was achieved by reducing the values of K_I and K_D in the PID controller, which allowed the kite to gradually build up acceleration in a smoother manner.

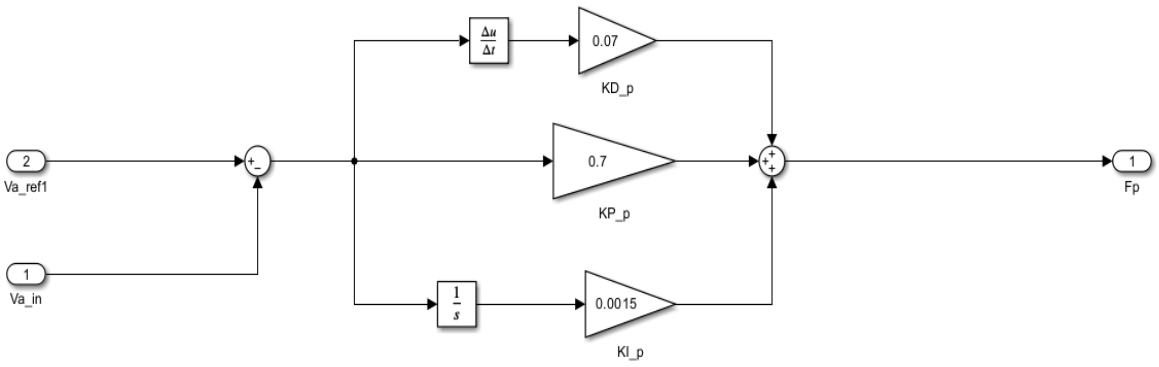


Figure 3.13: PID used in state 1 of the ATOL model.

Gains of F_q PID	Values	Gains of ω_q PID	Values
K_{D_k}	0.05	K_{D_c}	0.07
K_{P_k}	0.7	K_{P_c}	0.7
K_{I_k}	0.08	K_{I_c}	0.0015

Table 3.2: PID gains used to smooth the kite's acceleration.

This adjustment enabled the cart to synchronize its velocity with the kite without requiring changes to the acceleration saturation limits as shown in Figure 3.14.

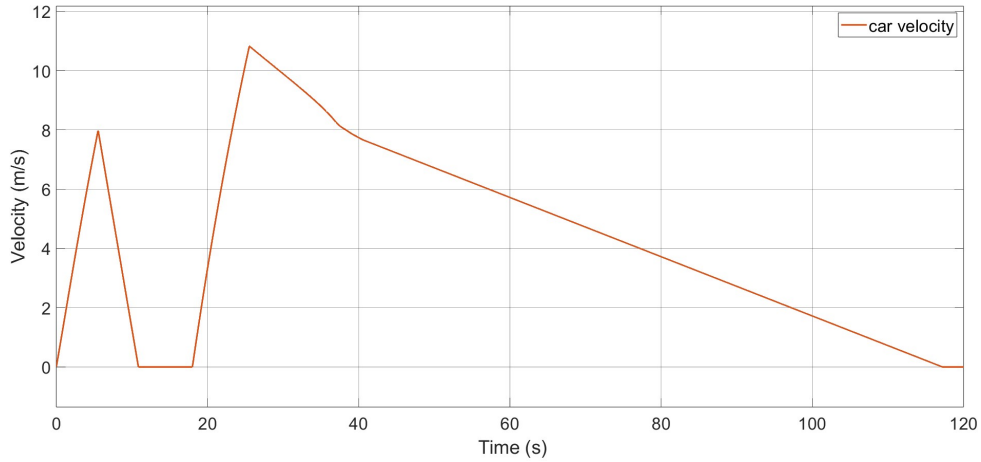


Figure 3.14: Velocity of the cart.

Therefore, in addition to phase synchronization between the cart and the kite, we also achieved velocity synchronization, which was essential to reach the desired synchronism for this work. In Figure 3.15, we can observe all the components that influenced the cart's behavior during the synchronism phases, particularly the variations in velocity caused by different operational conditions, as described in Sections 3.3 and 3.3.

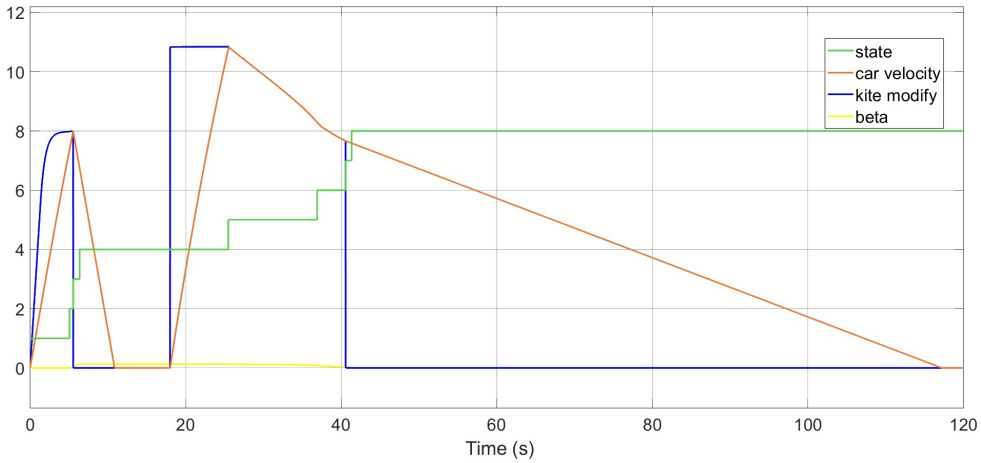


Figure 3.15: Cart velocity along with all components influencing its behavior.

3.4 Results of the Kite Dynamics and Control

Using MATLAB/Simulink, the kite's flight behavior was modeled, fully representing its velocity (V_a) (Figure 3.5), azimuth angle (ϕ) (Figure 3.6) and elevation angle (β) (Figure 3.7).

In the Take-off Phase, the kite successfully accelerated along the circular path until it reached the defined reference velocity ($V_a = V_{rot}$). The pitch controller rotated the kite to the required angle ($\theta = \theta_{rot}$), allowing it to begin its climb.

The Climb and Loiter Phase also appeared correctly, where the desired climb velocity ($V_a = V_{climb}$) and loitering conditions ($\beta = \beta_0$, $V_a = V_{loiter}$) were effectively achieved and maintained.

Finally, in the Landing Phase, the glide and flare maneuvers were successfully executed, enabling the kite to descend smoothly with deceleration and a controlled touchdown. The velocity was regulated through saturation blocks within the model.

3.4.1 Auxiliary Vehicle Synchronization

As stated before, the synchronization between the auxiliary ground vehicle and the kite was accomplished using a Kuramoto-inspired synchronization model. This approach enabled both phase and velocity alignment between the vehicle and the kite (Figure 3.15), with high system responsiveness and low latency. This translated into effective control and responsiveness of the car model.

Key observations include:

- Phase Tracking Performance – Synchronization was achieved almost instantly, with the vehicle dynamically adjusting its acceleration to maintain phase alignment with the kite.
- Energy Optimization – The velocity cutoff implemented in Simulink for the kite was used to halt the vehicle during unproductive states, reducing unnecessary power consumption and enhancing battery life. This behavior was confirmed during physical tests with the prototype car.

3.5 PWM Signal Processing and State-Based Electromagnet Control

At this stage, we had a functional signal correctly following the kite's velocity V_a. The next step was to transfer this signal to an Arduino for real-world testing. However, to send this signal to the Arduino, it was necessary to use a PWM (Pulse Width Modulation) technique to control the motor speed without altering its voltage.

To achieve this, the signal had to be normalized so it could be interpreted correctly by the Arduino using the following equation:

$$\mathrm{PWM}_{\mathrm{norm}} = \frac{\mathrm{PWM}_{\mathrm{raw}} - \mathrm{PWM}_{\mathrm{min}}}{\mathrm{PWM}_{\mathrm{max}} - \mathrm{PWM}_{\mathrm{min}}} \cdot 255$$

Where PWM_{raw} is the original signal, and PWM_{min} and PWM_{max} represent the minimum and maximum expected values of the signal, respectively. This normalization ensures compatibility with the 8-bit resolution (0–255) scheduled by the Arduino PWM input.

$$\text{PWM}_{\text{norm}} = \frac{\text{PWM} - 0}{100\% - 0} * 255 = \frac{\text{PWM}}{100\%} * 255 \tag{3.17}$$

Through this equation, we obtain the gain K (Figure 3.16 and Figure 3.17) necessary to correctly convert the values generated in Simulink to the corresponding PWM values used to control the cart's motor.

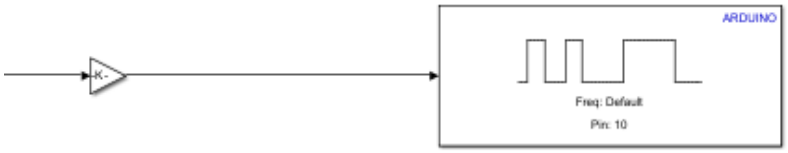


Figure 3.16: Normalization of the car´s velocity signal through a k gain.

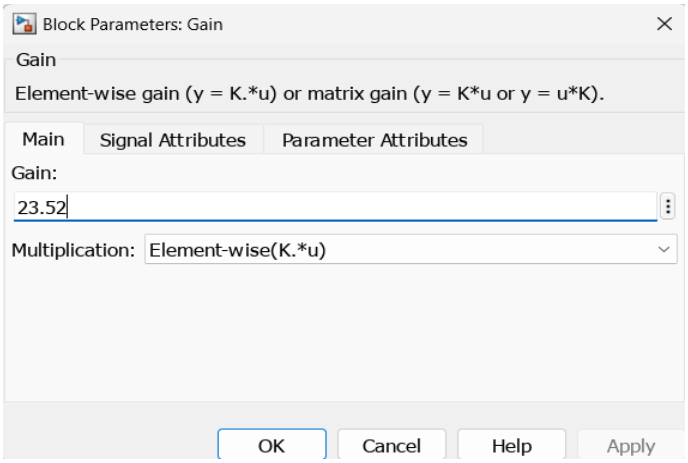


Figure 3.17: Value of the K gain.

It is important to note that the PWM signal ranges from 0 to 255 rather than from 0% to 100%. This is because it follows the standard resolution of 8-bit microcontrollers, which allows for $2^8 = 256$ discrete levels, i.e., from 0 to 255. In this context, a value of 0 corresponds to a 0% duty cycle (signal always off), while a value of 255 corresponds to a 100% duty cycle (signal always on).

The final step was to pass the signal through a block that routes it to a specific, predefined Arduino pin, which operates on a PWM scale ranging from 0 to 255.

Finally, it was necessary to generate a control signal for the electromagnet mounted on the car (Figure 3.18). To implement this functionality, a set of logical conditions was defined based on the current state of the system:

- If state = 3, 4, 5, or 6, then iman = 0 — In state 3, the kite reaches its required rotational velocity and begins its initial climb. For this to occur, the electromagnet must be deactivated to allow the kite to detach from the car. In states 4, 5, and 6, the kite remains airborne, and thus, there is no need for the electromagnet to remain active.

- Else, iman = 1 — In all other states, the electromagnet remains activated. In particular, during states 1 and 2, the car is still accelerating to reach the rotational speed required for the kite's takeoff, hence the electromagnet must stay engaged. In state 7, as described in [2.9.2], the kite descends to an altitude low enough to be within range of the electromagnet, enabling reattachment. Finally, in state 8, the car's motor is turned off and it is decelerating with the kite already coupled, so the electromagnet must remain active to maintain this connection.

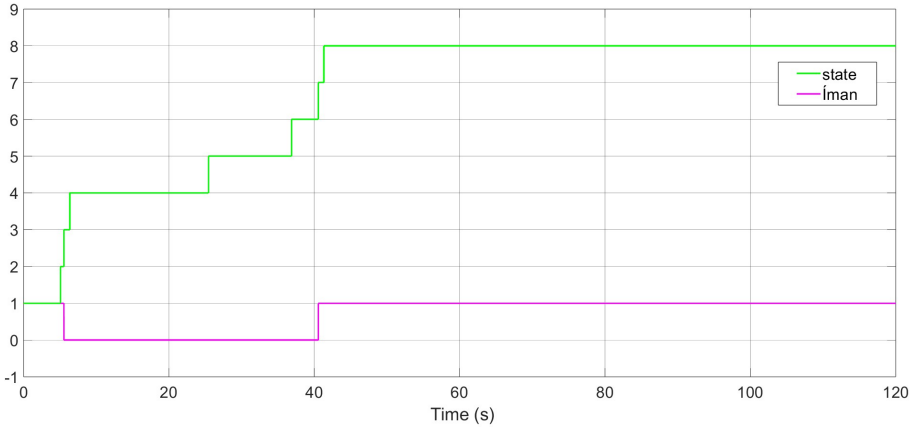


Figure 3.18: Iman´s behaviour along the different states.

Regarding the state block, a minor modification was made to ensure compatibility with the new car model. The main objective was to manipulate the current state based on whether the car's velocity matched the kite's velocity.

To achieve this, a tolerance was introduced to account for small fluctuations between the two velocities. Additionally, a condition was implemented in state 4 such that, after a delay of 18 seconds—allowing the kite to stabilize in the air at a constant speed of 10.8 m/s—the transition to the next state would only occur if the car's velocity equaled the kite's velocity. This mechanism was essential to facilitate proper synchronization.

The velocity matching condition relied on the defined tolerance range, which ensured that minor fluctuations did not prevent the state transition.

3.6 Simulation environment and 3D Visualization

In this project, with the goal of making the system fully autonomous, and capable of managing all the processes, the use of ArduPilot was planned. ArduPilot is a sophisticated open-source software platform for embedded autonomous navigation systems.

In order to verify whether synchronization between the cart and the kite was successfully achieved, this setup allowed for the simultaneous visualization of both the kite's parameters and its trajectory, in parallel with the cart's movement. This enabled real-time data exchange, supporting continuous improvement of synchronization accuracy.

To obtain these values reliably, it was essential to use components capable of acquiring data with high precision and low latency. Therefore, one of the main objectives was also to connect an IMU and a velocity sensor to a compatible firmware, specifically using a Pixhawk 4 flight controller, and to monitor the system via the Mission Planner interface.

3.7 IMU

The IMU (Inertial Measurement Unit) intended for use in this project is a device that measures velocity, orientation, and gravitational forces. It consists of three types of sensors:

- Accelerometer – measures linear acceleration.
- Gyroscope – measures angular velocity.
- Magnetometer – measures the yaw angle using Earth's magnetic field.

This combination allows measurements along all three axes, providing a total of 9 degrees of freedom (DOF). These values are measured independently, but parameters such as orientation angles and angular velocities can be obtained through sensor fusion and calibration techniques.

Using this IMU, it becomes possible to estimate the aircraft's position and attitude in flight. The specific IMU considered for this work was the BNO055 (Figure 3.19).

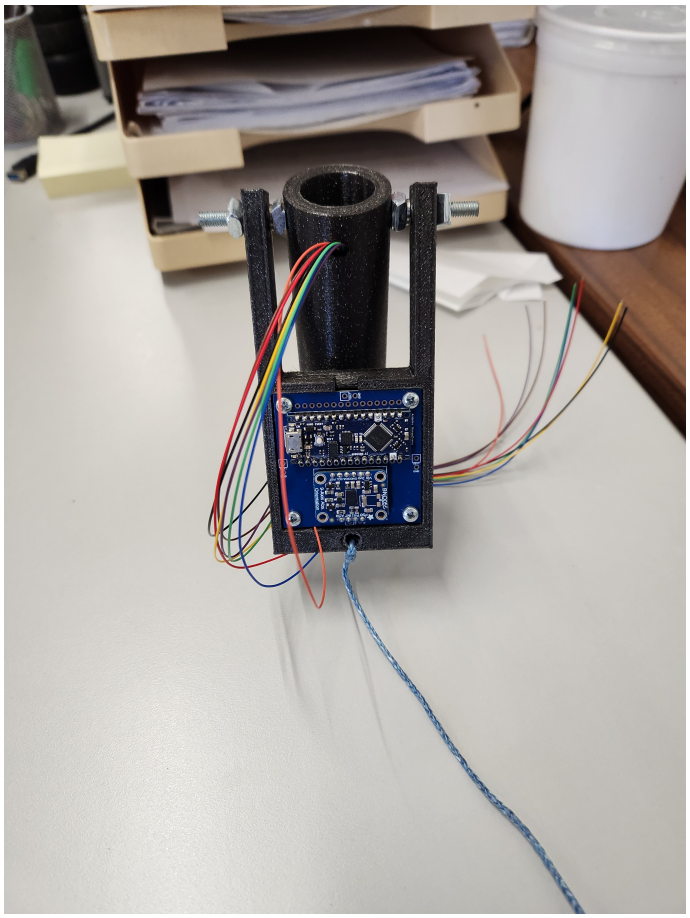


Figure 3.19: BNO055 used by UPWIND Team.

3.8 Velocity Sensor

The incremental optical encoder is commonly used for motion control, enabling speed measurement by counting the number of transitions (encoder edges) through a laser. These transitions occur when a rotating disc with holes (encoder) passes in front of a photodetector. The output is a square wave signal whose frequency is proportional to the encoder resolution (i.e., the number of evenly distributed holes along the disc's circumference), known as the pulse count (PC).

Another accurate method of measuring speed involves elapsed time (ET), which is determined by counting the number of high-frequency clock pulses (p) between two consecutive encoder transitions, where t_{clock} is the period of the clock:

$$ET = p \cdot t_{clock} \tag{3.18}$$

Figure 3.20 shows how the ET and PC methods work. The velocity is typically expressed in revolutions per minute (RPM), but it can be converted into meters per second or kilometers per hour with the following formulas:

Let:

- r be the radius of the encoder.

For meters per second (m/s):

$$v_{(m/s)} = v_{(rpm)} \cdot \frac{2\pi r}{60} \tag{3.19}$$

For kilometers per hour (km/h):

$$v_{(km/h)} = v_{(m/s)} \cdot 3.6 \tag{3.20}$$

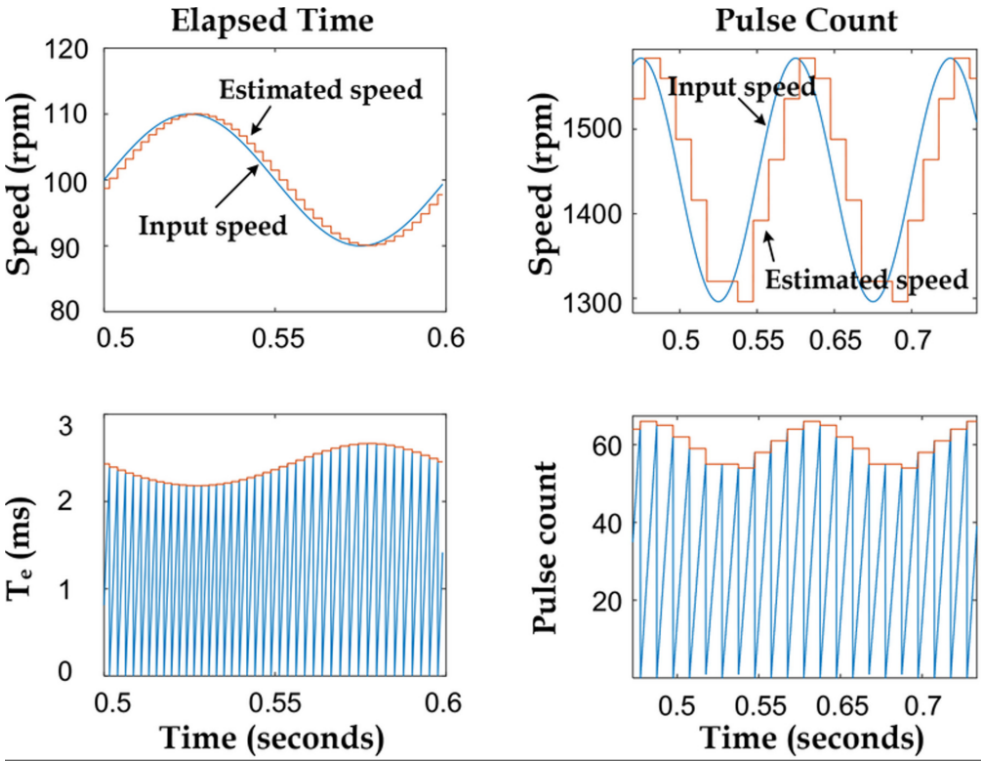


Figure 3.20: ET and PC methods for accurately measuring velocities and their operating principles[21].

The sinusoidal encoder signal is squared using a signal detector to produce the square wave output. The sensor module selected for this project was the LM393 (Figure 3.21).

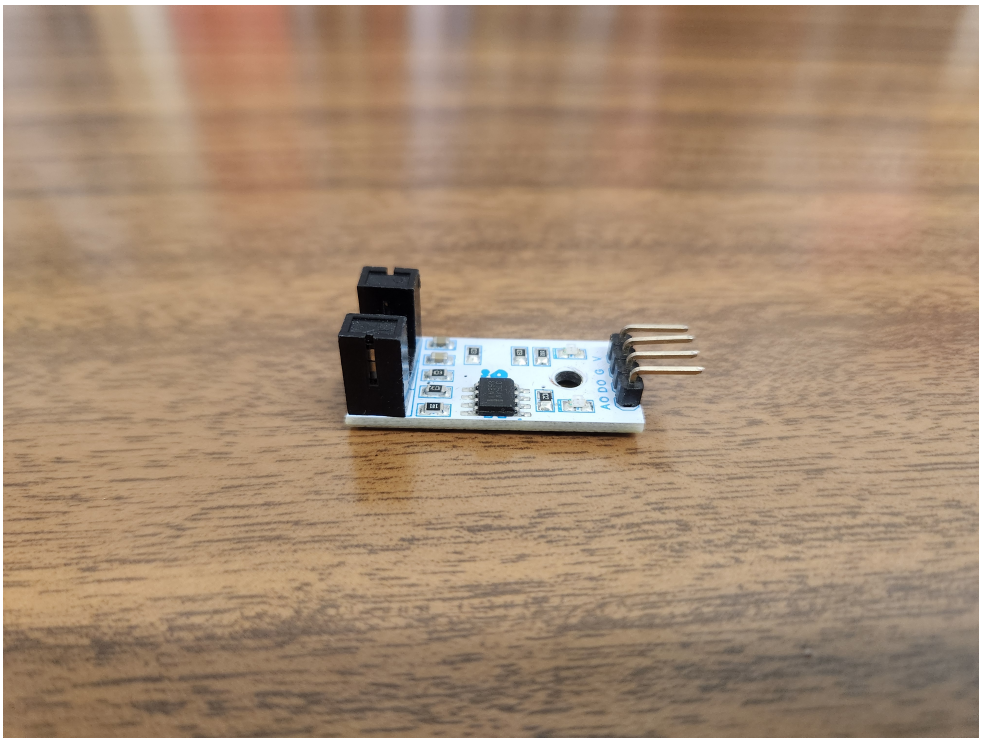


Figure 3.21: LM393 used by UPWIND Team.

3.9 Central Rotary Mechanism

This mechanism serves as a platform that integrates all components necessary for kite flight operation, including the encoder, IMU, and velocity sensor. It also includes the tether connecting the kite to the system, thus functioning as an improvised ground station (Figure 3.22).

The structure consists of a rigid frame composed of multiple parts fabricated via 3D printing.

Figure 3.22: Central rotary system: Implementation used by the UPWIND Team.

3.10 ArduPilot

ArduPilot is an open-source plane firmware capable of transforming fixed-wing aircraft into fully autonomous platforms [32].

This firmware significantly impacted research laboratories and commercial projects by offering full autonomous capabilities at a much lower cost. Being open-source, it fostered an active global community that enhances problem-solving and promotes collaborative development.

Moreover, ArduPilot supports not only aerial vehicles but also land and water-based platforms.

The ArduPilot ecosystem typically includes:

- The latest autopilot firmware installed on the aircraft.
- A compatible airframe, transmitter, and receiver.
- Supporting tools and hardware for flight operation and maintenance.
- Bidirectional telemetry radios enabling real-time communication.
- A Ground Control Station (GCS) software providing a user-friendly interface for setup and monitoring.

3.11 Mission Planner

The Ground Control Station (GCS) UPWIND Team uses is the Mission Planner. This decision was based on its reputation as one of the most comprehensive and feature-rich tools available for ArduPilot [33].

Mission Planner offers:

- Full access to internal ArduPilot parameters and configurations.
- Advanced analysis tools such as the Log Viewer, including graphical plots, sensor visualizations, and anomaly detection.
- Integrated SITL (Software-In-The-Loop) simulation capabilities.
- A powerful and intuitive graphical user interface.

Table 3.3 shows the differences between the different ground control stations:

Table 3.3: Comparison of Ground Control Stations Compatible with ArduPilot.

Feature	Mission Planner	QGroundControl	APM Planner 2	MAVProxy
Detailed configuration	Complete	Limited	Partial	CLI only
Log analysis	Advanced	Basic	Partial	CLI
Modern interface	Functional	Modern	Modern	CLI only
Integrated SITL simulation	Easy to use	Manual	Not available	Available
Scripts / plugins	IronPython	Not supported	Not supported	Python
Multi-vehicle support	Limited	Best	Not supported	Available

3.11.1 ArduPilot Incompatibility

Although the initial goal was to use ArduPilot in conjunction with the Mission Planner ground station to visualize the aircraft trajectory and control its path, several limitations prevented us from continuing with this approach, ultimately leading to a change in strategy. The main issues encountered were:

- Integration Complexity – ArduPilot has a steep learning curve, requires extensive firmware configuration, and involves a wide range of communication protocols.
- Hardware Compatibility Constraints – Not all sensors and IMUs are compatible with the ArduPilot firmware, which complicated the selection of appropriate components for the system.
- Testing Limitations in Controlled Environments – Due to the complexity of the system, testing required specific facilities that were not easily accessible or affordable, adding logistical and financial constraints to the project.
- Time Constraints – Given the limited time available to complete this work, the full integration and implementation of a platform as complex as ArduPilot became unfeasible, and an alternative solution had to be adopted.

3.12 Simulink Kite 3D Simulation

Due to the difficulties encountered with ArduPilot integration, a new strategy was adopted based on a simpler yet functional simulation model capable of visualizing the kite's trajectory. This change allowed us to assess the synchronization between the kite and the ground vehicle.

To this end, a 3D simulation environment was integrated (Figure 3.23) into the Simulink model mentioned in [3.3].

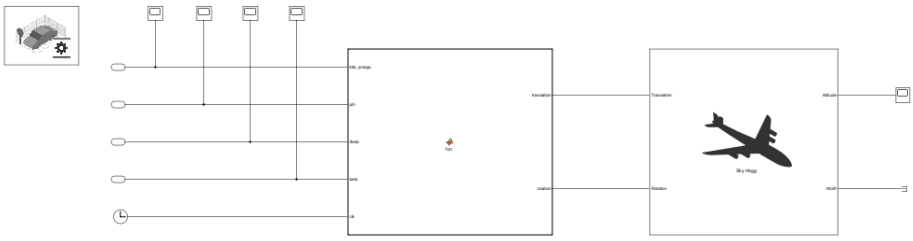


Figure 3.23: Complete 3D animation schematic in simulink.

The Simulation 3D Aircraft block implements an aircraft in a 3D virtual scene by applying translation and rotation inputs. These transformations use the aircraft's North-East-Down (NED) right-handed Cartesian coordinate system:

- X-axis — Along the aircraft's longitudinal axis, pointing forward.
- Y-axis — Along the lateral axis, pointing to the right.
- Z-axis — Perpendicular to both, pointing downward.

Several aircraft models are available in the block, but in this project the SkyHogg model was selected for its simplicity and similarity to the laboratory kite configuration (Figure 3.24) — a single-propeller aircraft with a nose-mounted engine.

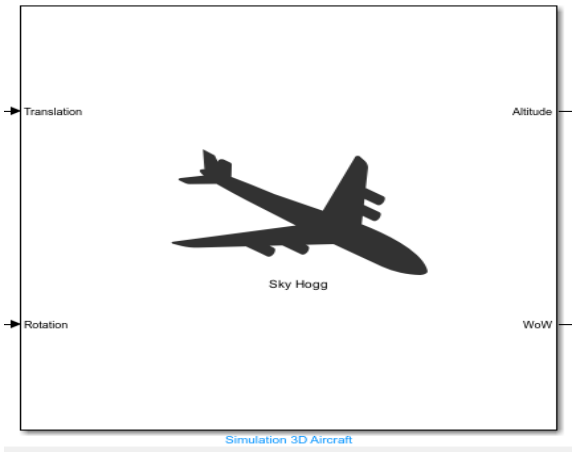


Figure 3.24: 3D aircraft simulation block in Simulink.

To visualize the kite trajectory, the block accepts an input matrix of dimensions 11×3, which includes both translation and rotation data for different parts of the aircraft.

For rotational motion, the following signals from the ATOL model were used:

- **Body (Row 1)** – Uses the signal ϕ to represent the kite's orientation along the circular path.
- **Propeller (Row 2)** – Uses v_a to animate the spinning propeller.

Since the kite undergoes only longitudinal rotation (along its body axis), no other aircraft parts needed to be animated.

For translation, the kite's trajectory was defined using the ϕ and β signals, applied to the Body (Row 1) of the translation matrix. This allowed for a clear and accurate representation of its circular flight path.

Table 3.4 shows all the components that constitute the 11×3 matrix of the SkyHogg aircraft model.

Table 3.4: Translation and rotation vectors used for the SkyHogg aircraft model [31].

Type: SkyHogg	
Size: [11x3]	
Row	**Component**
1	BODY
2	PROPELLER
3	RUDDER
4	ELEVATOR
5	LEFT_AILERON
6	RIGHT_AILERON
7	FLAPS
8	NOSE_WHEEL_STRUT
9	NOSE_WHEEL
10	LEFT_WHEEL
11	RIGHT_WHEEL

To render the simulation environment, the Simulation 3D Scene Configuration block (Figure 3.25) was also used. This block manages the overall 3D scene and its visualization parameters, which include:

- Scene Source – Defines how and where the 3D visualization will run (e.g., external Unreal Engine or built-in scenes).
- Scene Name – Specifies the landscape or environment to be visualized.
- Scene View – Indicates which vehicle is shown if multiple are present in the scene.
- Sample Time – Sets the update interval for the 3D visualization.

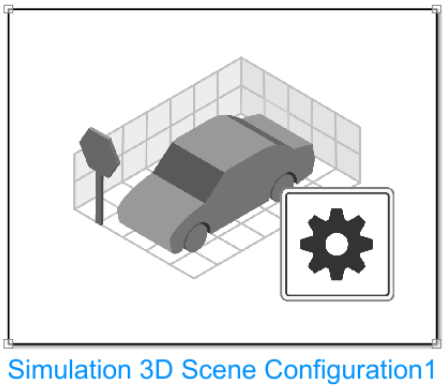


Figure 3.25: Simulation 3D Scene Configuration block.

The 3D simulation allowed for detailed analysis of trajectory accuracy, ascent angle, and synchronization behavior. The Simulink 3D scene configuration (Figure 3.25) was employed to create a virtual environment where the kite could be visualized in real time during the CTOL (Conventional Take-Off and Landing) maneuver.

The kite was dynamically modeled with vectors (Tab 3.4) representing its position, velocity, orientation, and tether constraints. Through a predefined scenario, its trajectory could be visualized from takeoff to landing.

In all stages, the kite appeared to behave as expected and followed the implemented signals properly, from takeoff to final touchdown. From this point onward, the simulation model was successfully completed, allowing the system to be executed, which provided a practical and intuitive way to analyze the kite's behavior in real time and evaluate the performance of the implemented control and synchronization strategies. Figure 3.26 shows the Simulink simulation with the kite following its circular trajectory.

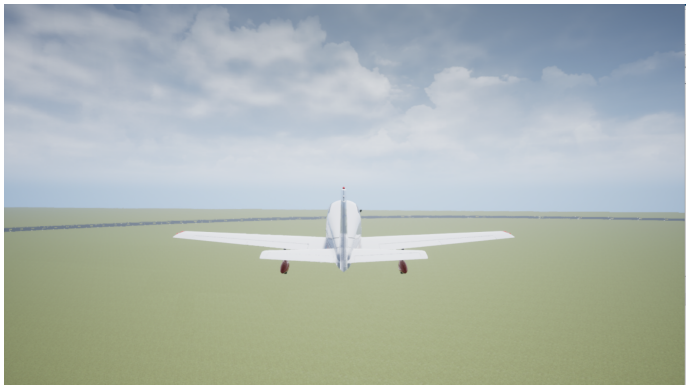


Figure 3.26: Visualization of the kite trajectory in the Simulink 3D environment.

Chapter 4

Communications

In this work, a wireless communication architecture was implemented to allow the transfer of control signals generated in the car's Simulink model between two Arduino Nano Every microcontrollers: one connected to Simulink through a UART serial interface, and another mounted on the vehicle chassis.

The transmitter Arduino (TX) acts as a gateway between the Simulink model and the wireless communication network. Its primary function is to receive the desired control signals from Simulink via a UART serial connection, process them into structured data packets, and transmit them to the nRF24L01 transmitter module using the SPI protocol. This microcontroller essentially bridges the software simulation environment with the physical system.

The second microcontroller, mounted on the vehicle chassis, acts as a real-time actuator controller. Its role is to continuously receive the data packets wirelessly, decode the information, and generate the corresponding PWM signals required to drive the DC motor.

The SPI protocol was chosen due to its high speed, low latency, and deterministic nature, which are essential requirements for real-time motor control. Compared to other communication protocols such as I^2C or UART, SPI avoids overhead and bus arbitration while allowing synchronous full-duplex data transfer.

4.1 System Architecture

As explained above, the communication setup consists of two microcontrollers:

- An Arduino Nano Every connected to the Simulink model via serial communication, acting as a data transmitter of the signal values (Figure 4.1).
- Another Arduino Nano Every embedded in the vehicle structure, acting as a receiver of the transmitted values and generating the PWM signal to control the motor (Figure 4.2).

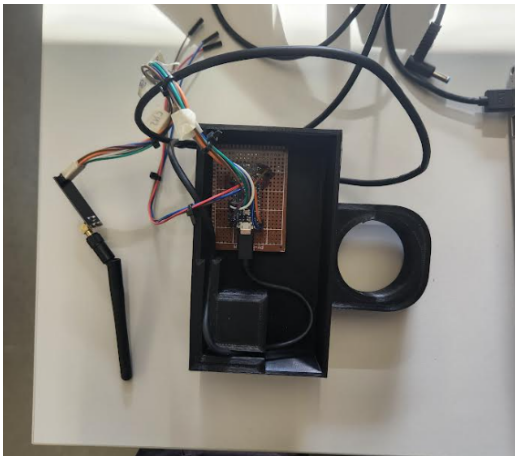


Figure 4.1: Transmitter module used in this work.

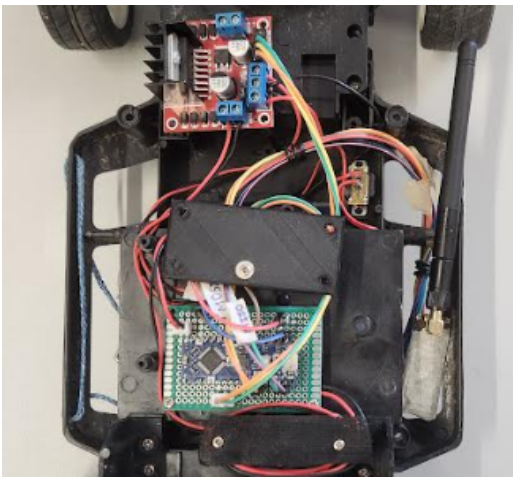


Figure 4.2: Receiver module mounted on the vehicle chassis.

The main objective was to establish a near-instantaneous wireless connection with minimal delay, reliable short-range communication (approximately 10–15 meters), easy implementation, low power consumption, and low cost. Based on these criteria, the nRF24L01 module was identified as the most suitable option.

The nRF24L01 enables data transmission in the 2.4 GHz ISM band and can transmit payloads of up to 32 bytes, with built-in auto-acknowledgement and retransmission mechanisms.

Table 4.1 summarizes the differences between several wireless modules commonly used in embedded systems.

Table 4.1: Comparison of Wireless Modules for Embedded Communication

Feature	**nRF24L01**	**HC-05**	**ESP8266**	**433 MHz**
Freq. Band	2.4 GHz	2.4 GHz	2.4 GHz	433 MHz
Max Data Rate	2 Mbps	1 Mbps	11 Mbps	~10 kbps
Range (LOS)	100+ m	10 m	50–100 m	100–500 m
Latency	Very Low	Low	Moderate	High
Power Use	Very Low	Low	High	Very Low
Cost (EUR)	<2 €	3–5 €	5–8 €	<2 €
Arduino Integration	Easy	Very Easy	Moderate	Easy
Two-Way Comm.	Yes	Yes	Yes	No
RT Control Ready	Yes	Partial	No	Limited

The module communicates with the Arduino using the SPI protocol, which is widely used in microcontrollers for high-speed data exchange. Both microcontrollers shared the same pin configuration (Table 4.2), as shown below:

Table 4.2: Pinout Connection Between NRF24L01 Module and Arduino Nano Every

NRF24L01 Pin	Arduino Nano Pin	Function
VCC	3.3V	Power supply
GND	GND	Ground
CE	D7	Chip Enable (controls TX/RX mode)
CSN	D8	Chip Select Not (SPI device selection)
SCK	D13	SPI Clock
MOSI	D11	Master Out Slave In (SPI)
MISO	D12	Master In Slave Out (SPI)

It is worth highlighting two important pins, **CE** and **CSN**, which play a key role in enabling communication through the nRF24L01:

- CE (Chip Enable) – Enables data transmission or reception. It is set high to activate the receiver mode and toggled during transmission.
- CSN (Chip Select Not) – Active low. Used to initiate and manage SPI communication between the microcontroller and the nRF24L01 module.

4.2 Module Variations

There are several variations of the NRF24L01 module available. The version (specification can be visualized in Table 4.3) used in this work includes an onboard PCB antenna, which provides a compact solution with an effective transmission range of approximately 100 meters and stable communication in most scenarios.

Table 4.3: Main Specifications of the onboard nRF24L01 Wireless Transceiver

Parameter	Typical Value	Comment
Operating Frequency	2.4 – 2.525 GHz	Operates in the license-free ISM band
Data Rate	250 kbps, 1 Mbps, 2 Mbps (configurable)	Allows trade-off between range, robustness, and throughput
RF Channel Bandwidth	1 MHz per channel	Reduces interference and improves selectivity
Transmit Power	0 dBm (adjustable from –18 dBm to 0 dBm)	Adjustable to save power or extend range
Receiver Sensitivity	–82 dBm @ 2 Mbps –85 dBm @ 1 Mbps –94 dBm @ 250 kbps	Higher sensitivity at lower data rates
TX Current Consumption	$\approx$ 11.3 mA @ 0 dBm	Low current consumption, suitable for battery-powered systems
RX Current Consumption	$\approx$ 13.5 mA	Moderate current draw during continuous listening
Standby Current	22 μA	Very energy-efficient in idle mode
Supply Voltage	1.9 V – 3.6 V (3.3 V typical)	Direct 5 V connection is not supported
Maximum Payload Size	32 bytes per packet	Supports auto-acknowledge and automatic retransmission

Another version uses an SMA connector with an external antenna for longer range. A third variant includes an integrated RFX2401C chip with a power amplifier (PA) and low-noise amplifier (LNA), enabling ranges up to 1000 meters in open environments (Figure 4.3).

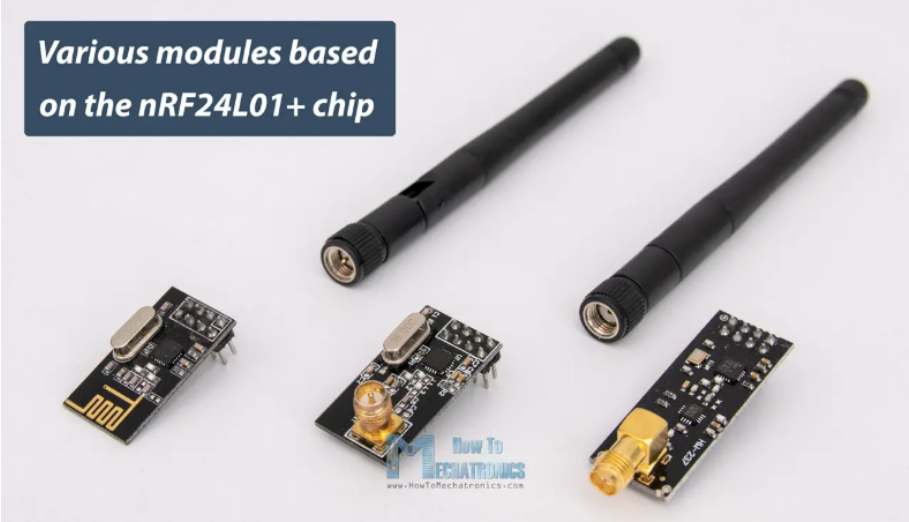


Figure 4.3: Different versions of the NRF24L01 module [22].

4.3 Signal Path

This wireless communication setup enables the transmission of control signals generated in Simulink to command the motor of the vehicle. The signal flow (visualized in 4.4) is as follows:

- Signal generation in Simulink – The control signal is generated within the Simulink environment.
- Transmission to the transmitter Arduino via UART – The signal is sent from Simulink to the transmitter Arduino Nano Every through a UART serial interface.
- Wireless transmission using nRF24L01 – The transmitting Arduino formats the data and sends it wirelessly using the nRF24L01 transceiver.
- Reception by the receiver Arduino – The receiver Arduino decodes the incoming packets and interprets them as control commands.
- Motor control – The decoded signal is mapped to a PWM duty cycle and sent to the motor driver to control motor speed and direction.

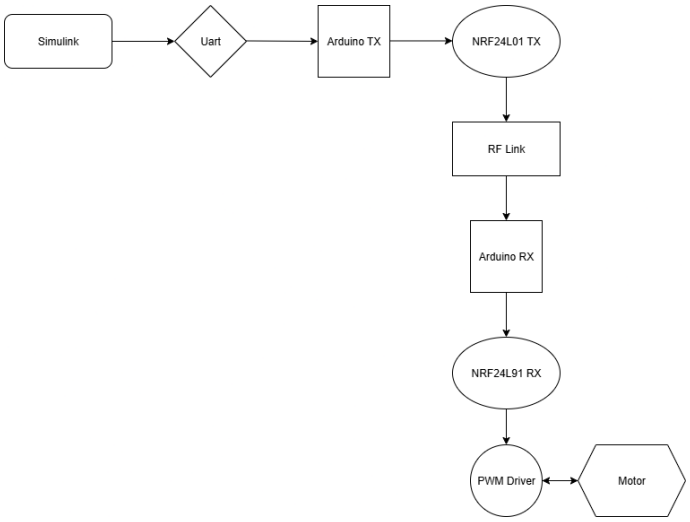


Figure 4.4: Signal flow diagram of the communication system.

4.4 Troubleshooting

One of the most common issues when working with nRF24L01 modules is power supply instability [22]. Because RF circuits are sensitive to noise, a decoupling capacitor (10–100 μF) should be added across the VCC and GND pins to ensure clean power delivery.

Another frequent issue is that the 3.3 V pin on some Arduino boards cannot supply enough current for the nRF24L01. In these cases, an external 3.3 V voltage regulator with sufficient current capacity should be used.

4.4.1 Wireless Communication Software Architecture

To implement the communication system and motor control logic, two codebases were developed: one for the transmitter and one for the receiver. The transmitter code reads pre-recorded control data stored in an external header file and transmits it over the nRF24L01 module. The receiver continuously listens for incoming packets using the RF24 library, updates the last reception timestamp, and processes the data. If the value received is a special `STOP` command (encoded as -999), the PWM duty cycle is set to zero, stopping the motor. Otherwise, the value is interpreted as a speed and direction command.

Three main libraries were employed:

- SPI.h – Initializes and manages the SPI bus for full-duplex data exchange.
- nRF24L01.h – Provides register definitions and low-level access to the transceiver's functionality.
- RF24.h – High-level abstraction layer for configuring transmission power, opening reading/writing pipes, entering listening mode, and handling packet transfer.

4.5 Testing and Implementation of NRF24L01 Wireless Communication

The NRF24L01 wireless transmitter module was tested for its ability to establish stable bidirectional communication between the ground vehicle and a central controller (the computer). Two Arduino Nano Every boards were used for transmitting and receiving data packets under various conditions (Figure 4.1 and 4.2).

- Range Testing – Stable communication was achieved within a 10–15 meter radius indoors, with minor packet loss due to interference from metallic structures and thick walls. Outdoors, in line-of-sight conditions, the range extended beyond 30 meters without significant degradation.
- Power Supply Filtering – A stable 3.3 V power source was used to avoid intermittent resets. Decoupling capacitors (10 μF + 100 nF) already installed on the NRF24L01 transmitter board (Figure 4.1) helped to mitigate voltage ripple and electrical noise.

Chapter 5

Auxiliary Vehicle Design and Implementation

5.1 RC Car Platform

The mechanical platform used in this work is based on a 1:20 scale RC car with an injection-molded ABS plastic chassis, chosen for its durability, low cost and ease of manufacturing. However, due to its limited torsional stiffness and the absence of mounting points for the components required for this project, several modifications were implemented. These modifications were achieved through the introduction of custom 3D-printed parts, including dedicated mounts for the electronic modules, battery supports and protective covers.

From a structural perspective, the chassis was designed to keep the components as centered as possible, ensuring proper balance. Its layout is reminiscent of a ladder-type frame, which helps distribute loads efficiently. The car uses a rear-wheel-drive configuration, with the rear wheels providing propulsion and the front wheels handling steering.

5.1.1 Motor Used

The car is powered by a micro DC geared motor with a 1:48 reduction ratio, operating at 6V and delivering 208RPM. The motor is controlled by an L298N dual H-bridge driver, which receives PWM signals generated in Simulink and transmits them to the motor. The 1:48 gear ratio provides the necessary torque to move the vehicle under its own weight without altering the power supplied.

This motor configuration allows for straightforward control, which is crucial for maintaining a constant linear velocity along a circular trajectory. The choice of this motor, rather than a more powerful brushless motor, was motivated by its simplicity, low cost and ease of integration with the L298N driver.

The L298N driver operates using an H-bridge configuration, enabling polarity reversal of the motor voltage and therefore the ability to change the motor's rotation direction. This is achieved through the IN1 / IN2 pair for motor A and the IN3 / IN4 pair for motor B. Additionally, the driver includes two enable pins (ENA and ENB), which must have their jumpers removed to receive

external PWM signals for speed control. Table (5.1) shows how the logic of the L298N driver works.

Table 5.1: Control logic of the L298N driver for two DC motors

ENA	IN1	IN2	Motor A	ENB	IN3	IN4	Motor B
0	X	X	Stopped (disabled)	0	X	X	Stopped (disabled)
1	0	0	Stopped (freewheel)	1	0	0	Stopped (freewheel)
1	1	0	Forward	1	1	0	Forward
1	0	1	Backward	1	0	1	Backward
1	1	1	Stopped (brake)	1	1	1	Stopped (brake)

The L298N driver also includes a voltage regulator that outputs a steady 5V when powered with an input voltage between 5–12V, allowing the system to power a microcontroller (Arduino Nano Every in this case) directly from the same supply.

Figures 5.1 and 5.2 show, respectively, the pinout diagram and the internal block diagram of the L298N driver.

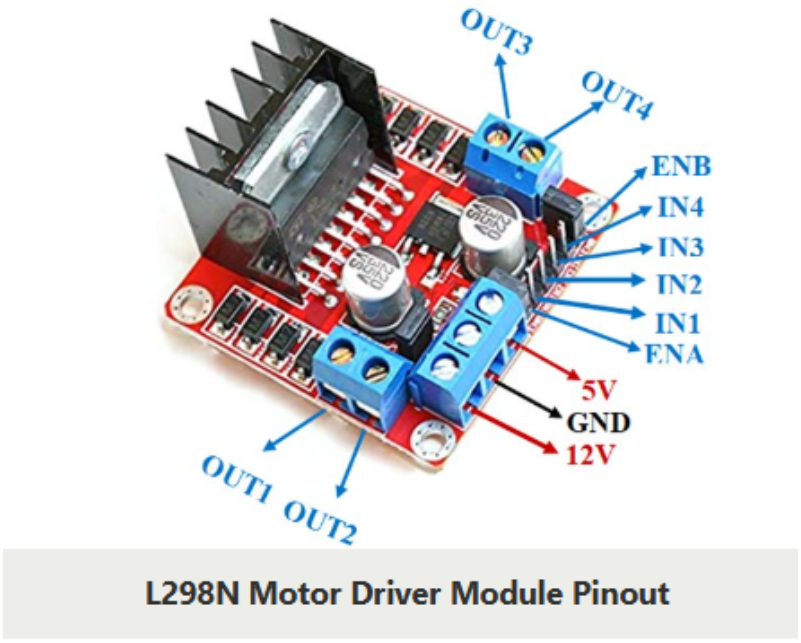


Figure 5.1: Pinout diagram of the L298N motor driver [34].

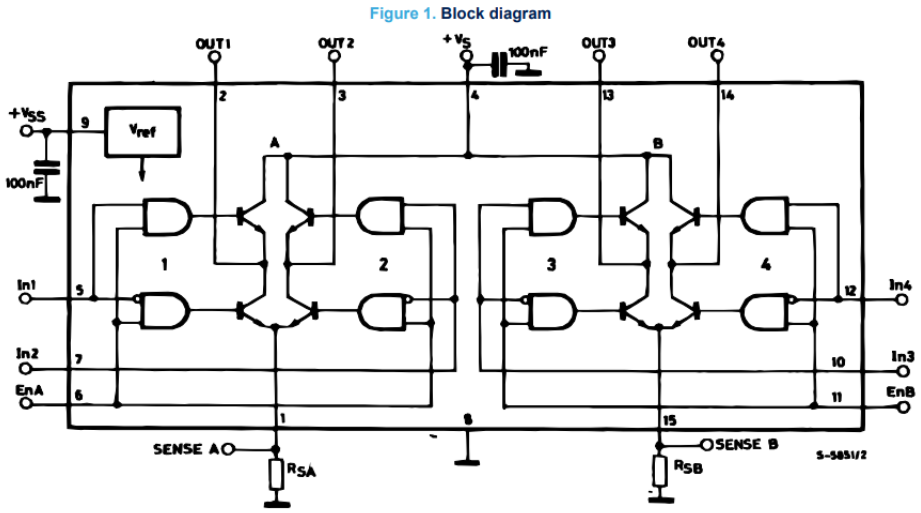


Figure 5.2: Block diagram of the L298N motor driver [35].

Given its simplicity, low cost and compatibility with the chosen motor, the L298N driver was the natural choice for this project.

The vehicle uses a battery compartment containing three cells in series (3.8V each), providing 11.1V to both the motor and the microcontroller. Each cell has a capacity of 2200mAh, offering sufficient runtime for the project experiments.

A main power switch allows the user to safely turn the car on and off, preventing unnecessary current draw, avoiding unexpected motor activation, and simplifying maintenance and component replacement.

The chosen microcontroller for this project was the Arduino Nano Every (Figure 5.3), selected for its low cost, compact form factor (facilitating integration into the chassis) and low power consumption, which contributes to longer system autonomy. Additionally, the Nano Every offers more flash memory compared to other similarly priced microcontrollers and maintains pin compatibility with more powerful Arduino boards, simplifying future system upgrades.

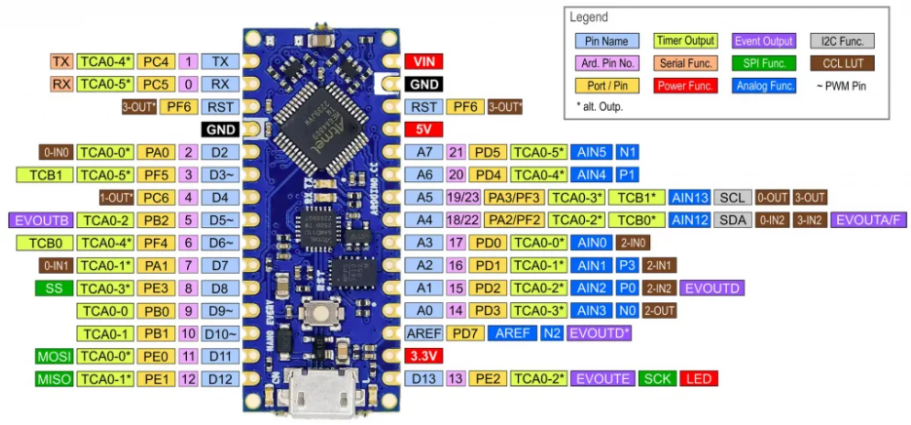


Figure 5.3: Pinout diagram of the arduino nano every [35].

So basically, in terms of hardware, this car is similar to the vehicle described in Section 2.10, but it excludes the electromagnet and kite landing platform (Figure 5.4). It is composed of two of the three main sub-circuits: the power circuit, consisting of a pair of 3.7V batteries and an LM2596 voltage step-down module, and the motor drive circuit, which powers and regulates the two rear-wheel motors.

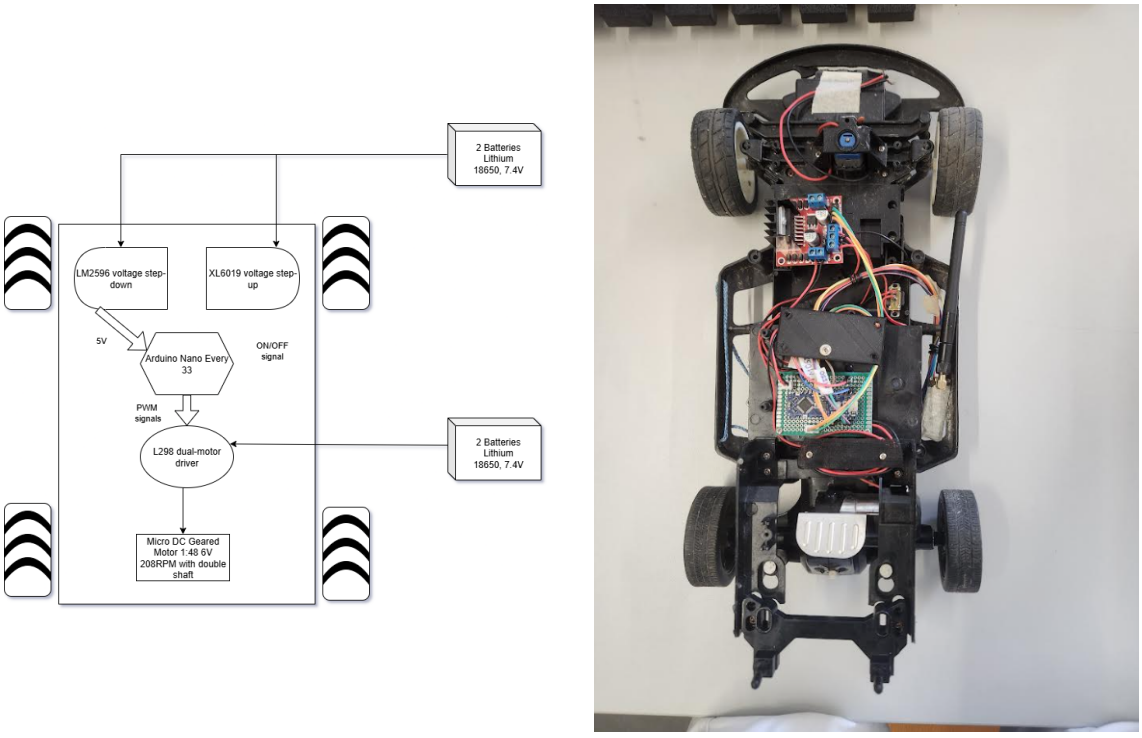


Figure 5.4: Electronic diagram of the vehicle (left) and prototype concept (right).

Due to the absence of the electromagnet sub-circuit, a simple test circuit was developed to verify the proper functioning of the control signal generated in MATLAB (Figure 5.5). This circuit used an LED to visualize the signal behavior over time and assess its synchronization with

the car's motion.

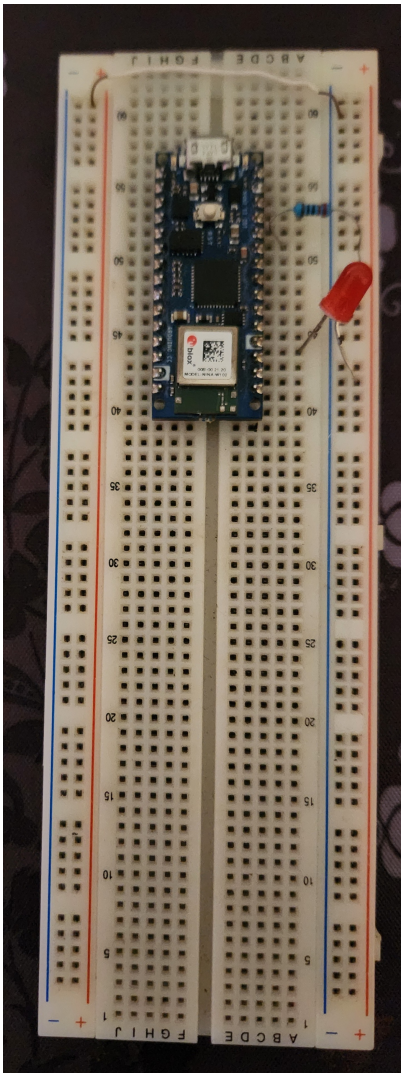


Figure 5.5: Test circuit using an LED to emulate and observe the electromagnet signal.

The electromagnet is intended to hold the kite firmly during takeoff and landing. As described in Section 3.5, the electromagnet is deactivated after State 3, allowing the kite to begin its trajectory, and reactivated only when the kite reaches a sufficient altitude to re-engage with the magnetic field.

The choice of this RC car was driven by its ability to reach the speeds required for the proposed experiments. However, a faster and more powerful model will soon be acquired by the UPWIND Team, fully equipped to integrate all necessary components for autonomous kite takeoff and landing.

This prototype primarily served to test communications and validate whether the Simulink-generated signals were correctly interpreted. Future experiments will replicate these tests on the upgraded RC car with a fully functional electromagnet.

5.1.2 Motor Control via L298N Dual H-Bridge Driver

Figure 5.6 shows the physical tests carried out to verify the car's behaviour.

The propulsion system of the support vehicle used a rear-wheel DC motor driven by an L298N dual H-bridge driver as mentioned before.

PWM signals generated by a Raspberry Pi were used to control motor speed via the ENA and ENB pins on the driver. A basic velocity profile, including step and ramp inputs, was tested to validate the system's ability to handle acceleration and deceleration.

In addition, the program was executed from different initial positions to verify that the motors performed as expected under all starting conditions,and finally, the car's velocity signal was tested to verify whether its behaviour matched the intended outcome.

Figure 5.6: Physical testing of the car conducted in FEUP.

Chapter 6

Conclusions

AWES represent a major step towards the implementation of emerging technologies that harness wind power as an energy source, aiming to generate environmental benefits while also improving the cost/benefit ratio.

This work contributes to AWE and brings us closer to the main objective: to develop a kite capable of autonomously taking off, following a defined trajectory to generate energy, and landing safely.

In terms of specific goals, this project successfully generated a control signal for the ground vehicle (the cart), allowing it to synchronize its movement with the kite. It also implemented wireless motor control using the NRF24L01 module and Arduino Nano Every boards. The control signal was designed in Simulink and transmitted via UART to a transmitting Arduino, which sent the data wirelessly through the NRF24L01 module. The receiving Arduino then processed the signal and converted it into PWM commands to control the vehicle's motor.

Furthermore, the kite's trajectory was successfully visualized through a simulation created in Simulink using the 3D Aircraft Simulation environment.

The system architecture proved to be effective for remote control applications, utilizing a simulation-to-hardware pipeline. The use of Simulink provided a flexible environment for signal generation and allowed real-time interaction with physical hardware.

Additionally, the report highlighted key characteristics of the NRF24L01 module, including the significance of the CE and CSN pins in SPI communication and data transmission control. Different hardware versions of the module were discussed, noting trade-offs between communication range, physical size, and power consumption.

Practical challenges were also addressed, such as power supply noise and the voltage limitations of the Arduino's 3.3V pin. Solutions like the use of decoupling capacitors and external power sources were proposed to improve system stability and ensure reliable communication.

Overall, this work represents a step toward achieving the ultimate goal of creating an autonomous energy resource supported by an auxiliary ground vehicle. It is expected to serve as a foundation for future developments in the AWES.

6.0.1 Future Improvements

Future work will focus on improving the platform to make it more robust and capable of handling higher speeds, as well as integrating all necessary components for full functionality. Proposed improvements include:

- Reinforced chassis: Increase rigidity and load capacity, and properly integrate the electromagnet into the frame.
- More powerful motor: Use a motor capable of higher speeds to improve synchronization with the kite.
- Sensor integration: Include additional sensors (e.g., IMU, wheel encoders) for better trajectory control.
- Suspension system: Add suspension to improve stability and handling at higher speeds.
- Electromagnet integration: Install a permanent electromagnet for real-world testing of the kite-coupling mechanism.

Appendix A

MATLAB/Simulink Layouts and Simulation Code

```
function kite_omega = fcn(kite_omega, clk, state, beta_p)
    % x = [theta_carrinho, omega_carrinho, theta_aviao]
    v_mod = kite_omega;

    % alpha = 0;
    % beta = 0;

    if (state == 3)
        kite_omega = 0;
    elseif (state == 4 && clk < 18)
        kite_omega = 0;
    elseif (beta_p <= deg2rad(1.5) && state ~= 1 && state ~= 2)
        kite_omega = 0;
    end
end
```

Listing A.1: Kite Angular Velocity Function

```
function [dxdt_1, dxdt_2, iman] = fcn(x1, x2, x3, u, u2, kite_omega, state)
    % x = [theta_carrinho, omega_carrinho, theta_aviao]
    k1 = 20;
    k2 = 0.1;
    iman = 0;

    % Kuramoto Control for the support car
    dxdt_1 = x2 + k1 * sin(x3 - x1);
    dxdt_2 = u + k2 * sin(x3 - x1) * kite_omega;

    if (state == 0 || state == 1 || state == 2 || state == 3 || state == 4)
        dxdt_2 = u + k2 * sin(x3 - x1) * kite_omega;
    else
        dxdt_2 = u2 + k2 * sin(x3 - x1) * kite_omega;
    end

    if (state == 3 || state == 4 || state == 5 || state == 6)
        iman = 0;
    else
        iman = 1;
    end
end
```

Listing A.2: Kuramoto-Based Car Control Function

```
function [translation, rotation] = fcn(kite_omega, phi, theta, beta, clk)
    persistent theta_prop dt last_clk trajectory

    if isempty(theta_prop)
        theta_prop = 0;
        last_clk = clk;
        trajectory = zeros(100, 3); % Stores the last 100 points
    end

    dt = clk - last_clk;
    last_clk = clk;

    % Update propeller angle
    theta_prop = theta_prop + kite_omega * dt;

    L = 50;
    translation = zeros(11, 3);
    rotation = zeros(11, 3);

    % Aircraft position
    x = L * cos(phi) * cos(beta);
    y = L * sin(phi) * cos(beta);
    z = -L * sin(beta) - 1.9;
    translation(1,:) = [x, y, z];

    % Aircraft orientation
    rotation(1,:) = [0, theta, phi + pi/2];
    rotation(2,:) = [theta_prop, 0, 0]; % Propeller
    t1 = translation(1,:);

    % --- Generates circular trajectory points ---
    radius = 20; % Curve radius
    angle = phi; % Current angle
    current_point = [
        radius * cos(angle),
        radius * sin(angle),
        z
    ];

    % Store the points (keep last 100)
    trajectory = circshift(trajectory, -1, 1); % Shift points
    trajectory(end, :) = current_point;

    % Output trajectory as 3 N  matrix
    trajectory_points = trajectory.'; % Transpose to 3 100
end
```

Listing A.3: Translation and Rotation Calculation for Kite Simulation

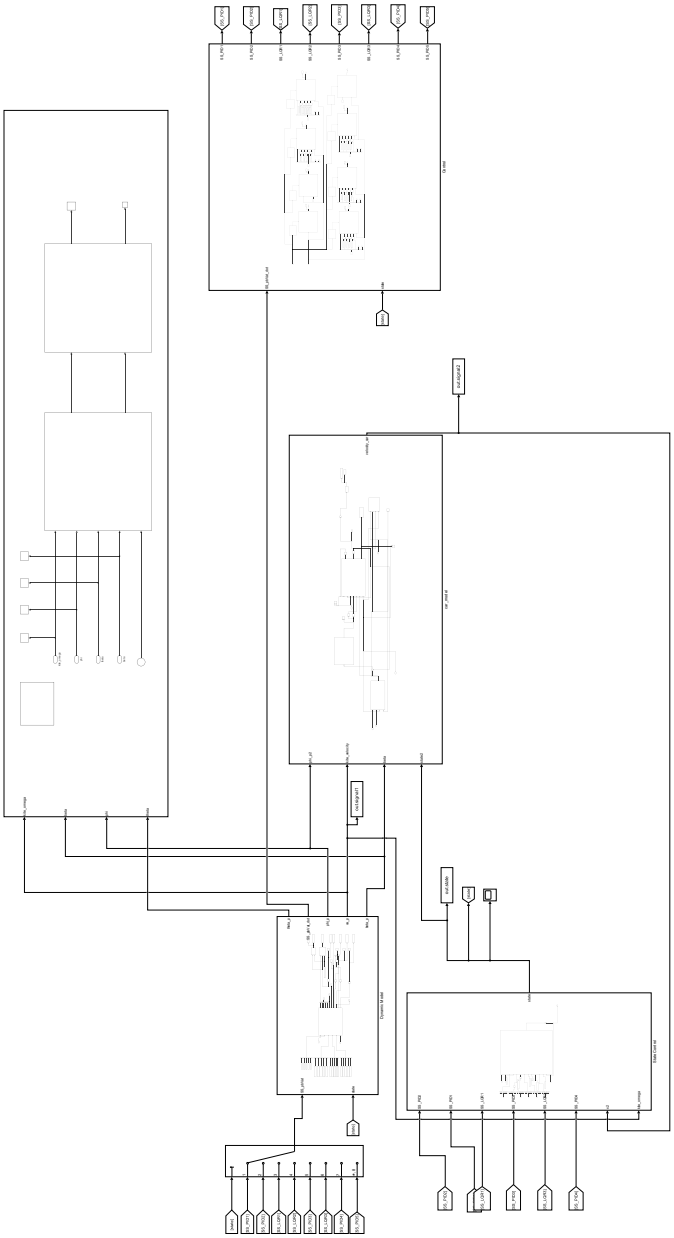


Figure A.1: Complete Simulink model, composed of the original ATOL model and the newly implemented car model.

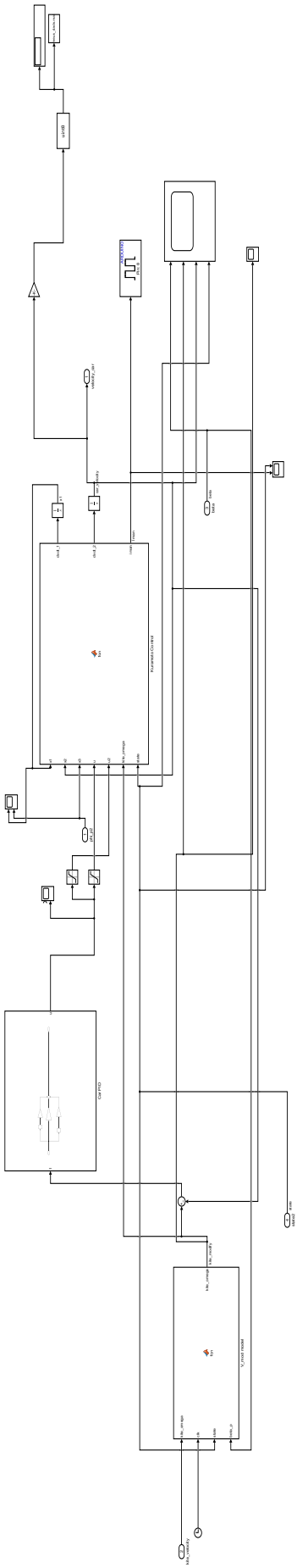


Figure A.2: New car model implemented in Simulink, integrated into the full system.

Appendix B

Arduino Codes Used for Communications

```
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

RF24 radio(7, 8);  // CE, CSN
const byte address[6] = "00001";

#define IN1 2
#define IN2 4
#define EN 3

int16_t valorRecebido = 0;
unsigned long ultimaRececao = 0;
const unsigned long timeout = 200;  // timeout in milliseconds
bool motorParadoPorTimeout = false;

void setup() {
  Serial.begin(9600);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(EN, OUTPUT);

  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);

  radio.begin();
  radio.openReadingPipe(0, address);
  radio.setPALevel(RF24_PA_LOW);
  radio.startListening();

  ultimaRececao = millis();  // start clock
  Serial.println("Receptor pronto");
```

```
}

void loop() {
  if (radio.available()) {
    radio.read(&valorRecebido, sizeof(valorRecebido));
    ultimaRececao = millis();
    motorParadoPorTimeout = false;

    if (valorRecebido == -999) {
      analogWrite(EN, 0);
      Serial.println("      ␣Comando␣STOP␣recebido␣   ␣motor␣parado");
      return;
    }

    // Set direction
    if (valorRecebido >= 0) {
      digitalWrite(IN1, HIGH);
      digitalWrite(IN2, LOW);  // forward
    } else {
      digitalWrite(IN1, LOW);
      digitalWrite(IN2, HIGH);  // reverse
    }

    int pwm = constrain(abs(valorRecebido), 0, 255);
    analogWrite(EN, pwm);

    Serial.print("Recebido:␣");
    Serial.print(valorRecebido);
    Serial.print("␣|␣PWM:␣");
    Serial.println(pwm);
  }

  // Timeout: stop motor if no signal
  if (millis() - ultimaRececao > timeout && !motorParadoPorTimeout) {
    analogWrite(EN, 0);
    Serial.println("      ␣Timeout:␣motor␣parado␣por␣falta␣de␣dados.");
    motorParadoPorTimeout = true;
  }
}
```

Listing B.1: Arduino NRF24L01 Receiver and Motor Control

```
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#include "dados2.h"

RF24 radio(7, 8);  // CE, CSN
const byte address[6] = "00001";

bool enviar = false;
unsigned int i = 0;
String comando = "";

void setup() {
  Serial.begin(9600);
  Serial.println("Transmissor pronto. Digite 'start' ou 'stop'.");

  radio.begin();
  radio.openWritingPipe(address);
  radio.setPALevel(RF24_PA_LOW);
  radio.stopListening();
}

void loop() {
  // Read command from Serial
  if (Serial.available()) {
    comando = Serial.readStringUntil('\n');
    comando.trim();

    if (comando == "start") {
      enviar = true;
      Serial.println(">> Transmiss o iniciada");
    } else if (comando == "stop") {
      enviar = false;
      Serial.println(">> Transmiss o pausada");
    } else {
      Serial.println("Comando inv lido. Use 'start' ou 'stop'");
    }
  }

  // Transmit data if active
  if (enviar && i < dados_tamanho) {
    int16_t valor = dados[i];
    bool ok = radio.write(&valor, sizeof(valor));

    Serial.print("Enviado [");
    Serial.print(i);
    Serial.print("]: ");
    Serial.print(valor);
    Serial.println(ok ? "    " : "    ");
```

```
50
51      i++;
52      delay(10);  // Adjust delay as needed
53    }
54
55    // Reset if end reached
56    if (i >= dados_tamanho && enviar) {
57      Serial.println(">> Fim dos dados. Aguardando novo 'start' para repetir.");
58      enviar = false;
59      i = 0;
60    }
61  }
```

Listing B.2: NRF24L01 Transmitter with Serial Command Control

Bibliography

[1] "World Energy Transitions Outlook 2023: 1.5°C Pathway," International Renewable Energy Agency, 2023. Available: https://www.irena.org/Publications/2023/Jun/World-Energy-Transitions-Outlook-2023.

[2] Roland Schmehl, "Airborne Wind Energy," in *AWESCO - Airborne Wind Energy System Modelling, Control and Optimisation*, 2019. Available: http://awesco.eu/awe-explained/. Accessed: Mar. 20, 2025.

[3] Jochem Weber, Melinda Marquis, Aubryn Cooperman, Caroline Draxl, Rob Hammond, Jason Jonkman, Alexsandra Lemke, Anthony Lopez, Rafael Mudafort, Mike Optis, Owen Roberts, Matt Shields, "Airborne Wind Energy," in *Renewable Energy*, 2021.

[4] Antonello Cherubini, Andrea Papini, Rocco Vertechy, Marco Fontana, "Airborne Wind Energy Systems: A Review of the Technologies," in *Renewable and Sustainable Energy Reviews*, vol. 51, Nov. 2015. Available: https://linkinghub.elsevier.com/retrieve/pii/S1364032115007005.

[5] D. Vander Lind, "Developing a 600 kW Airborne Wind Turbine," Delft University of Technology, 2015. Available: TU Delft Repository

[6] Sérgio Vinha, Gabriel M. Fernandes, and Fernando A. C. C. Fontes, "Circular Take-off and Landing of Tethered Aircraft with Variable Tether Length," in *Proceedings of CONTROLO 2024 – The Portuguese Conference on Automatic Control*, APCA – Associação Portuguesa de Controlo Automático, Porto, Portugal, 2024. https://repositorio.fe.up.pt/handle/10216/151000

[7] Luís Paiva and Fernando Fontes, "Optimal Control Algorithms with Adaptive Time-Mesh Refinement for Kite Power Systems," in *Energies*, vol. 11, no. 3, Feb. 2018. Available: https://www.mdpi.com/1996-1073/11/3/475. DOI: 10.3390/en11030475.

[8] Sérgio Vinha, Gabriel M. Fernandes, and Fernando A. C. C. Fontes, "A Discussion on Automatic Take-off and Landing Approaches for Airborne Wind Energy Systems," in *Proceedings of CONTROLO 2024 – The Portuguese Conference on Automatic Control*, APCA – Associação Portuguesa de Controlo Automático, Porto, Portugal, 2024. https://www.upwind.pt

[9] Sérgio Vinha, Gabriel M. Fernandes, Huu Thien Nguyen, Manuel C.R.M. Fernandes, Fernando A.C.C. Fontes, "Automatic Circular Take-off and Landing of Motorized Tethered Aircraft," in *2024 European Control Conference (ECC)*, 2024. Available: https://ieeexplore.ieee.org/document/10590893.

[10] Gabriel M. Fernandes, Conrado Guimarães da Costa, Sérgio Vinha, and Fernando A. C. C. Fontes, "A Testing Platform for Airborne Wind Energy: Modelling, Sensoring, and Actuation," in *Proceedings of CONTROLO 2024 – The Portuguese Conference on Automatic Control*, APCA – Associação Portuguesa de Controlo Automático, Porto, Portugal, 2024. https://www.upwind.pt

[11] Sérgio Vinha, "UPWIND Project POCI-01-0145-FEDER-031447 Activities Report," *Technical Report*, 2024.
https://repositorio.fe.up.pt/handle/10216/151000

[12] Andrea Dunlap, Makani Technologies LLC, "Makani Power Collection," 2006-2020. Available: https://archive.org/details/makani-power?tab=about.

[13] Sérgio Vinha, Gabriel Fernandes, Manuel Fernandes, Huu Nguyen, Fernando Fontes, "Control of Fixed-Wing Tethered Aircraft in Circular Take-Off and Landing Maneuvers," in *Proceedings of the 21st International Conference on Informatics in Control, Automation and Robotics*, Porto, Portugal, 2024. Available: https://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0013067000003822.

[14] Miles L. Loyd, "Lawrence Livermore National Laboratory, Livermore, Calif.," *Technical Report*, Lawrence Livermore National Laboratory, 2021. http://www.energykitesystems.net/Loyd/index.html

[15] Filippo Trevisi, Alessandro Croce, Carlo E.D. Riboldi, "Flight Stability of Rigid Wing Airborne Wind Energy Systems," in *Energies*, vol. 14, no. 22, Nov. 2021. Available: https://www.mdpi.com/1996-1073/14/22/7704.

[16] Gabriel M. Fernandes and Sérgio Vinha, "Automatic Circular Take-off and Landing of Self-Propelled Kites," in *Proceedings of CONTROLO 2024 – The Portuguese Conference on Automatic Control*, APCA – Associação Portuguesa de Controlo Automático, Porto, Portugal, 2024. https://repositorio.fe.up.pt/handle/10216/151000

[17] Chris Vermillion, Mitchell Cobb, Lorenzo Fagiano, Rachel Leuthold, Moritz Diehl, Roy S. Smith, Tony A. Wood, Sebastian Rapp, Roland Schmehl, David Olinger, Michael Demetriou, "Electricity in the Air: Insights from Two Decades of Advanced Control Research and Experimental Flight Testing of Airborne Wind Energy Systems," in *Annual Reviews in Control*, vol. 52, 2021. Available: https://linkinghub.elsevier.com/retrieve/pii/S1367578821000109.

[18] Arthur E. Bryson, "Applied Optimal Control: Optimization, Estimation, and Control," *Hemisphere Pub. Corp., Washington; distributed by Halsted Press, New York*, 1975.

ISBN: 978-0-470-11481-0.
Available: http://archive.org/details/appliedoptimalco0000brys

[19] F. A. C. C. Fontes, L. T. Paiva, M. C. R. M. Fernandes, S. Vinha, "An Aircraft and Tethered Support Vehicle System, Method of Operating the Same for Circular Take-Off and Landing of the Aircraft and Use Thereof," European Patent EP24154972.4, 2023.

[20] Filippo Trevisi, Alessandro Croce, Carlo E.D. Riboldi, "Flight Stability of Rigid Wing Airborne Wind Energy Systems," in *Energies*, vol. 14, no. 22, Nov. 2021. Available: https://www.mdpi.com/1996-1073/14/22/7704.

[21] J. Vázquez-Gutiérrez, "Study of the Impact of the Incremental Optical Encoder Sensor on the Dynamic Performance of Velocity Servosystems," in *The Journal of Engineering*, Wiley Online Library, 2019. Available: https://ietresearch.onlinelibrary.wiley.com/doi/full/10.1049/joe.2018.8255.

[22] Dejan, "nRF24L01 – How It Works, Arduino Interface, Circuits, Codes," How To Mechatronics, Feb. 2022. Available: https://howtomechatronics.com/tutorials/arduino/arduino-wireless-communication-nrf24l01-tutorial/. Accessed: Jul. 23, 2025.

[23] GeeksforGeeks, "Wired and Wireless Networking," Available: https://www.geeksforgeeks.org/computer-science-fundamentals/wired-and-wireless-networking/. Accessed: Jul. 23, 2025.

[24] Jacob Fraden, "Handbook of Modern Sensors: Physics, Designs, and Applications," 4th ed., Springer, 2010. DOI: 10.1007/978-1-4419-6466-3.

[25] Karl J. Åström, Björn Wittenmark, "Computer-Controlled Systems: Theory and Design," 3rd ed., Prentice Hall, 1997.

[26] Gene F. Franklin, J. David Powell, Abbas Emami-Naeini, "Feedback Control of Dynamic Systems," 8th ed., Pearson, 2019.

[27] T.K. Hareendran, "Wireless Communication Between Two Arduinos Using nRF24L01," Electronics For You, 2017. Available: https://www.electronicsforu.com/electronics-projects/wireless-communication-between-two-arduinos-using-nrf24l01.

[28] Nordic Semiconductor, "nRF24L01+ Product Specification," Revision 1.0, 2008. Available: https://infocenter.nordicsemi.com/pdf/nRF24L01P_PS_v1.0.pdf.

[29] MathWorks, "Simulink Support Package for Arduino Hardware," Documentation, 2024. Available: https://www.mathworks.com/help/supportpkg/arduino/.

[30] Steven T. Karris, "Introduction to Simulink with Engineering Applications," Orchard Publications, 2011.

[31] MathWorks, “Simulating Aircraft in Unreal Engine-based 3D Environment Using Aerospace Blockset,” YouTube, Oct. 22, 2021. Available: https://youtu.be/vBVOeEt63kY. Accessed: Jul. 23, 2025.

[32] ArduPilot, “ArduPilot Plane — Plane documentation,” Available: https://ardupilot.org/plane/index.html. Accessed: Jul. 24, 2025.

[33] ArduPilot, "Choosing a Ground Station — Plane documentation," Available: https://ardupilot.org/plane/docs/common-choosing-a-ground-station.html. Accessed: Jul. 24, 2025.

[34] Components101, “L298N Motor Driver Module,” Available: https://components101.com/modules/l293n-motor-driver-module. Accessed: Sep. 13, 2025.

[35] STMicroelectronics, “L298N Dual Full-Bridge Driver,” Available: https://www.alldatasheet.com/datasheet-pdf/pdf/22440/STMICROELECTRONICS/L298N.html. Accessed: Sep. 13, 2025.