Modelling and Pitch Control of a Re-configurable Unmanned Airship

by

Gilmar Tuta Navajas

Thesis submitted to the University of Ottawa
In partial fulfillment of the requirements for the
M.A.Sc. degree in
Mechanical Engineering

Department of Mechanical Engineering
Faculty of Engineering
University of Ottawa

© Gilmar Tuta Navajas, Ottawa, Canada, 2021

Abstract

Lighter than air (LTA) vehicles have many advantageous capabilities over other aircraft, including low power consumption, high payload capacity, and long endurance. However, they exhibit manoeuvrability and control reliability challenges, and these limitations are particularly significant for smaller unmanned LTA. In this thesis, a 4 m length autonomous airship with a sliding gondola is presented. A rigid keel, mounted to the helium envelope, follows the helium envelope profile from the midsection to the nose of the vehicle. Moving the gondola along the keel produces upwards of 90-degree changes in pitch angle, thereby improving manoeuvrability and allowing for rapid changes in altitude. The longitudinal multi-body equations of motion were developed for this prototype using the Boltzmann–Hamel method. An adaptive PID controller was then designed to control the pitch inclination using the gondola's position. This control system is capable of self-tuning the controller gains in real time by minimizing a pre-defined sliding condition. Experimental flight tests were carried out to evaluate the controller's performance on the prototype.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Dr. Eric Lanteigne for his guidance, support, and patience throughout the course of my studies and the development of this thesis.

Also, I would like to thank my family and friends for their support during my studies.

Table of Contents

List of Tables

List of Figures

List of Symbols

A_R	Aerodynamic vector
D	Centrifugal and Coriolis vector
E_R	Model uncertainty vector
G	Gravitational and buoyancy vector
U_R	Input force vector
x	State vector
X_b, Y_b, Z_b	Body reference frame
X_i, Y_i, Z_i	Inertial reference frame
$\vec{q}$	Generalized coordinates
x, y, z	Translational coordinates of airship CG with respect to inertial frame
$\phi, \theta\psi$	Euler angles
S_s	Gondola travelled distance
Δ_z	Altitude variations around the reference altitude
z_{ref}	Reference altitude
R_1	Rotational matrix for coordinates from the body frame to inertial frame
R_2	Rotational matrix for angular velocities from the inertial frame to inertial frame
d_{rx}	Distance between the center of the circumference and the origin of the body frame, along the x axis

l	Distance between the CV and the gondola CG, along the z axis in the body frame
R	Radius of the curve
r_w	Rail thickness and added material
$\mathrm{M_a}$	Airship and added mass matrix
$\mathrm{J_a}$	Airship and added inertia matrix
M	System mass matrix
m_x, m_y, m_z	Airship and added mass
J_x, J_y, J_z, J_{xz}	Airship and added inertia
V	Airship volume
m_1	Body 1 mass
m_2	Body 2 mass
$m_{envelope}$	Envelope mass
m_{fins}	Fins mass
m_{helium}	Helium mass
m_{rail}	Rail mass
m_{air}	Air mass
m_t	Airship total mass
k_1	Lamb's inertia ratio about x
k_2	Lamb's inertia ratio about y or z
k'	Lamb's inertia ratio about y or z
$I_{b1,x}, I_{b1,y}, I_{b1,z}$	Moment of inertia Body 1
$I_{b2,x}, I_{b2,y}, I_{b2,z}$	Moment of inertia Body 2
u_i	Quasi-velocity i_{th}
$\vec{A}$	Aerodynamic vector
A_{xb}, A_{xb}, A_{xb}	Aerodynamic forces
A_ϕ, A_θ, A_ψ	Aerodynamic moments
$\hat{P}$	Dynamic pressure
δ_{air}	Air density
δ_{ref}	Reference air density

Δ_δ	Slope of air density
α	Angle of attack
β	Side-slip angle
C_{Dho}	Hull zero-incidence drag coefficient
C_{Dfo}	Fin zero-incidence drag coefficient
C_{Dgo}	Gondola zero-incidence drag coefficient
C_{Dch}	Hull cross-flow drag coefficient
C_{Dcf}	Fin cross-flow drag coefficient
C_{Dcg}	Gondola cross-flow drag coefficient
$\left(\frac{\partial C_L}{\partial \alpha}\right)_f$	Derivative of fin lift-coefficient with respect to the angle-of-attack at zero incidence
$\left(\frac{\partial C_L}{\partial \delta}\right)_f$	Derivative of fin lift-coefficient with respect to the flap deflection angle
S_h	Hull reference area
S_f	Fin reference area
S_g	Gondola reference area
l_{fx1}	X-distance from CV to aerodynamic center of fins
l_{fx2}	X-distance from CV to aerodynamic center of fins
l_{fx3}	Z-distance from CV to aerodynamic center of fins
n_f	Fin efficiency factor accounting for the effect of the hull on the fins
n_k	Hull effciency factor accounting for the effect of the fins on the hull
I_1, I_3, J_1, J_2	Hull integrals
$\vec{T}$	Thrust vector
e	Controller error
F_t	Total thrust
e	Controller error
θ_d	Desired pitch angle

$\gamma_1, \gamma_2, \gamma_3$	Controller learning rates
K_p, K_i, K_d	Gains of PID controller
S	Sliding surface
S_{sup}	Control signal of supervisory controller
S_{pid}	Control signal of PID controller
S_d	Desired gondola position
$x_{acc}, y_{acc}, z_{acc}$	Accelerometer readings
$\dot{\theta}_{gyr}$	Gyroscope reading
θ_{acc}	Computed pitch angle based on the accelometer reading
θ_{gyr}	Computed pitch angle based on the gyroscope reading
T_s	Sample time
α_f	Parameter of the complementary filter

Chapter 1

Introduction

In recent years, unmanned aerial vehicles (UAV) have increased in popularity due to their numerous applications in military, commercial, cargo and surveillance. There are two main categories for these aircraft: rotary wings (i.e., quadcopter), and fixed wings (i.e., airplane) [1]. Unmanned lighter than air (LTA) vehicles differ from rotary and fixed wings aircraft, since LTA aircraft use a lifting gas as the main source of lift, [2] leading to lower power consumption and a considerable payload ratio. As a result of these features, research in UAV airships has recently gained traction.

1.1 Motivation

The airship industry was displaced by aeroplanes at the end of the 30s. One of the causes was the flammability of hydrogen, the main lifting gas used at that time. Due to numerous airship disasters (e.g., Hindeburg) governments around the world adopted stricter regulations against the use of hydrogen for airships. For instance, the Civilian Aviation Authority of Germany (TAR 893) [3] and the U.S. Federal Aviation Administration (FAA ADC 4.48) [4] prohibit the use of flammable chemical elements as lifting gas. In Canada, according to the Canadian Aviation Regulations, the use of hydrogen as a lifting gas is prohibited (541.7) [5].

Interest in lighter than air vehicles has recently been renewed due to technological advances. For example, the replacement of hydrogen by helium, a non-flammable and lifting gas. In addition, LTAs have shown a wide spectrum of applications like advertising, aerial imaging, environmental monitoring, and cargo. As a result, there are numerous companies around the world such as Zeppelin Luftschifftechnik GmbH & Co (Germany), Skyship Services Inc (USA), Goodyear (USA), and Hybrid Air Vehicles (UK) that have been developing different manned airship prototypes to provide diverse services. These airships are shown in Figure 1.1. Furthermore, the Quebec government invested $30 million dollars into a French-Chinese company called Flying Whales, which is developing airships that could hold up to 60 tonnes [6].

(a) LZ N07-101 - Zeppelin [7]

(b) Skyship 600 - Skyship [8]

(c) Wingfoot One - Goodyear [9]

(d) Airlander 10 - Hybrid Air Vehicles [10]

Figure 1.1: Commercial airship models

Manned airships usually use air bag systems to produce changes in the longitudinal inclination and the altitude. In contrast, unmanned airships have limited payload to carry additional equipment, thus creating feasibility challenges. For this reason,

autonomous airships have yet to be adopted commercially.

1.2 Problem Description

The lack of manoeuvrability of autonomous LTAs increases the risk of potential impact during critical manoeuvres such as descent and obstacle avoidance. Hence, this research work uses an alternative approach in order to control the pitch variations. The method used is based on the concept of a shifting centre of gravity [11], thus increasing the system's manoeuvrability in rapid descents without needing to add a considerable number of components.

This thesis aims to model, control and implement a control strategy on a reconfigurable autonomous airship prototype. The recently constructed prototype uses a sliding gondola through a keel, allowing for pitch variations up to 90 degrees.

The airship prototype is shown in Figure 1.2. The blimp length is 4 m, the radius is 0.9 m, and the volume is 6.58 m^3. The envelope is made of 0.002" polyester film. Attached to it is a square carbon fiber piece with a cross-section of 10 mm x 10 mm that acts as a rail system for the gondola. In fact, there are two rail sections; the first one is straight with a length of 1 m, while the second one is a quarter circumference with a radius of 0.9 m.

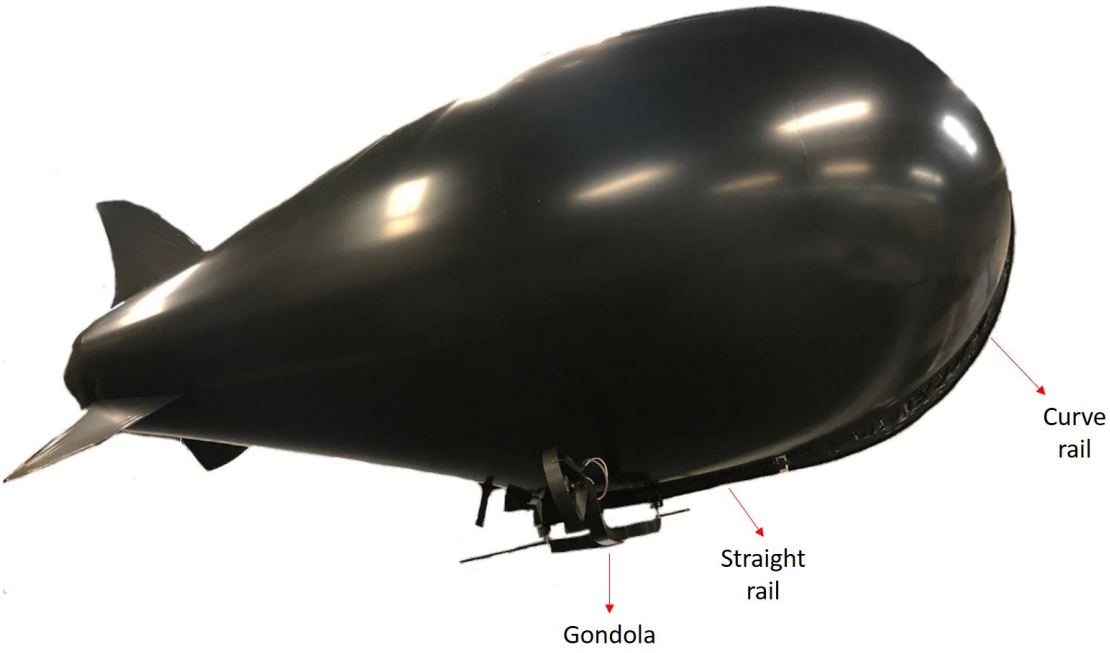


Figure 1.2: Airship prototype

1.3 Objectives

The objectives of this thesis are threefold:

- Model and simulate a dynamic model of the prototype vehicle.
- Design and simulate a non-linear controller to control the pitch angle of the prototype vehicle.
- Implement the control system in the prototype.

1.4 Thesis Layout

The remainder of the thesis is organized as follows:

- Chapter 2 introduces a review of the design, modelling, control and implementation of unmanned airships.
- Chapter 3 describes the multi-body longitudinal equations of the proposed reconfigurable autonomous airship.
- Chapter 4 presents the design of an adaptive PID control technique for controlling the pitch angle.
- Chapter 5 provides an overview of the experimental platform that was used for the experimental validation.
- Chapter 6 presents and discusses the flight test results.
- Finally, Chapter 7 provides concluding remarks and discusses possible future work.

Chapter 2

Literature Review

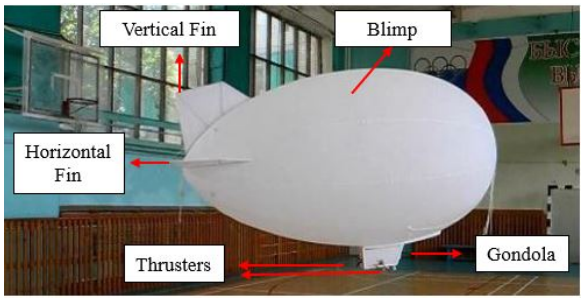
This chapter presents a brief overview of recent developments in unmanned airship technologies, modelling approaches, control techniques, as well as observations made from experimental flight tests.

2.1 Airship Design

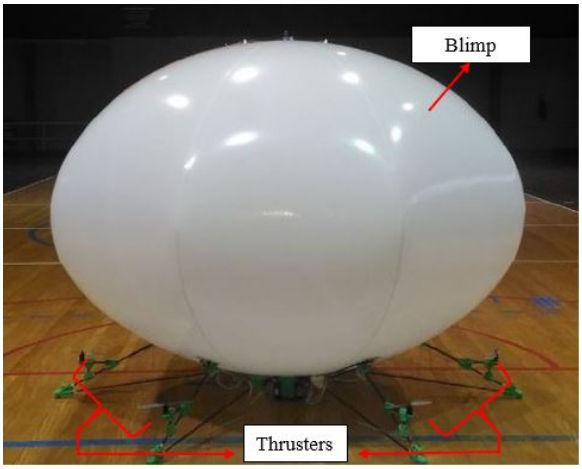
Numerous lifting gas envelope shapes have been studied and tested for applications in unmanned flights.

The most common shape that has been studied is the *prolate spheroid* [12–16]. The finesses ratio of this shape is generally optimized in order to determine the best drag and lift coefficients [17]. Moreover, this shape has a smaller drag coefficient than a sphere. An example of this shape is shown in Figure 2.1a. This type is characterized by its aerodynamic surfaces (e.g., fins, rudders, elevators). The fins are usually located in the rear, while the thrusters are often located under the envelope, as shown in Figure 2.1a. Typically, this shape integrates two types of aerodynamic control surfaces (rudders and elevators), whose function is to control the yaw and pitch angles, in addition to two vectored motors that provide a thrust to move forward. However, some configurations do not use control surfaces, but instead have unmotorized fins and servos attached to the thrusters, as in Figure 2.1a.

Another type of lighter than air unmanned vehicle shape is the *oblate spheroid* [18–20]. This shape usually has a greater lift coefficient than the prolate. However, this compromises its drag coefficient. Typically, for this configuration, the thrusters are placed around the envelope. In addition, these vehicles are finless, which means they do not have aerodynamic surfaces. Depending on the number of vectored propellers, its maneuverability will increase compared to that of the traditional airships. For instance, Figure 2.1b shows an oblate spheroid that has six thrusters. This setup provides an advantage compared to a pure hexa-copter aircraft, since the aerostatic lift will increase its endurance and payload capabilities [18]. This configuration improves the vehicle manoeuvrability. However, the extra motors increase the system's power consumption.



(a) Prolate spheroid airship [12]



(b) Oblate spheroid Airship [18]

Figure 2.1: Common shapes of airships

2.2 Airship Modelling

The dynamics model of the airship has been well studied. Models incorporate different types of physics such as: dynamics, aerodynamics, incorporation of structural flexibility, incorporation of atmospheric turbulence and effects of ballonets [2]. Airships are models that have six degrees of freedom (DOF). Most of the models found in the literature use the Newton - Euler formulation in order to obtain the equations of motion. This formulation is vector oriented. It is therefore necessary to determine the kinematics of the system [21]. On the other hand, there are some works which uses the Lagrange method [1]. This technique has an energy calculation approach. Sometimes, it is an advantage, especially when dealing with the kinematics for complex systems (e.g., multibody), but these can be tedious to manage.

2.2.1 Newton-Euler Method

One of the better known derived models for the manned airship is the YEZ-2a [22], whose equations were derived using the Newton-Euler method. Over the years, additional improvements have been added to Gomes's work. For instance, Recoskie added terms for parametric uncertainties, which are useful when evaluating controller robustness, as well as rotational aerodynamic damping [23]. Recoskie's dynamic model uses the coordinate frame depicted in Figure 2.2, and its equations of motion are expressed as [23]:

Equations of motion:

$$M\ddot{\mathrm{x}} + D(\dot{\mathrm{x}}) = A_R(\dot{\mathrm{x}}) + G + E_R + U_R \tag{2.1}$$

where M is the mass matrix, D is the centrifugal and Coriolis vector. Besides, A_R, G, E_R, and U_R are the aerodynamic vector, the gravitational and buoyancy vector, the model uncertainty vector, and the input force vector. The states are represented by the 6x1 vector x.

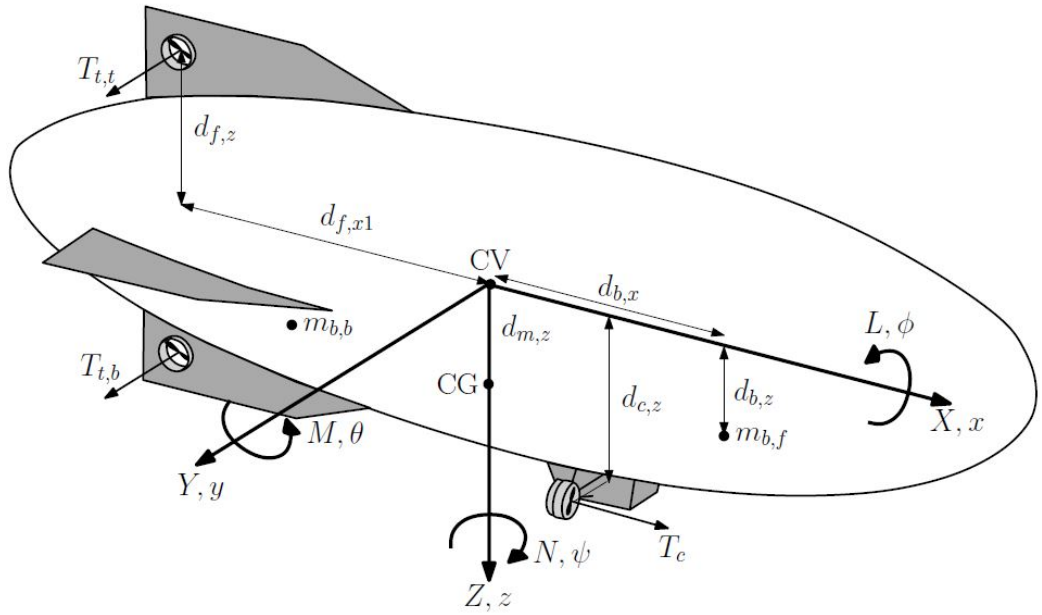


Figure 2.2: Coordinate frame [23]

$$\mathrm{x} = [x \quad y \quad z \quad \phi \quad \theta \quad \psi]^T \tag{2.2}$$

where the cartesian coordinates of the CV are x, y, and z. The Euler angles are expressed as ϕ, θ, and ψ.

The mass matrix is represented by the 6x6 matrix M [23].

$$M = \begin{bmatrix} m_x & 0 & 0 & 0 & md_{m,z} & 0 \\ 0 & m_y & 0 & -md_{m,z} & 0 & md_{m,z} \\ 0 & 0 & m_z & 0 & -md_{m,x} & 0 \\ 0 & -md_{m,z} & 0 & J_x & 0 & -J_{xz} \\ md_{m,z} & 0 & -md_{m,x} & 0 & J_y & 0 \\ 0 & md_{m,x} & 0 & -J_{xz} & 0 & J_z \end{bmatrix} \tag{2.3}$$

where the added mass for the x, y, and z axis are m_x, m_y, and m_z, respectively. The total mass of the system is m. The added inertia about the x, y, and z axes are J_x, J_y, and J_z, respectively. The added inertia about the xz plane is J_{xz}.

The forces and moments due to the centrifugal and Coriolis terms are represented by the 6x1 vector $D = [D_X \quad D_Y \quad D_Z \quad D_L \quad D_M \quad D_N]$ [23].

$$D_X = m_z \dot{z}\dot{\theta} - m_y \dot{\psi}\dot{y} + m(d_{m,x}[\dot{\theta}^2 + \dot{\psi}^2] - d_{m,z}\dot{\psi}\dot{\phi}) \tag{2.4}$$

$$D_Y = m_x \dot{x}\dot{\psi} - m_z \dot{\phi}\dot{z} - m(d_{m,x}\dot{\phi}\dot{\theta} + d_{m,z}\dot{\psi}\dot{\theta}) \tag{2.5}$$

$$D_Z = m_y \dot{y}\dot{\phi} - m_x \dot{\theta}\dot{x} + m(d_{m,z}[\dot{\theta}^2 + \dot{\phi}^2] - d_{m,x}\dot{\psi}\dot{\phi}) \tag{2.6}$$

$$D_L = (J_z - J_y)\dot{\psi}\dot{\theta} - J_{xz}\dot{\phi}\dot{\theta} + m d_{m,z}(\dot{\phi}\dot{z} - \dot{x}\dot{\psi}) \tag{2.7}$$

$$D_M = (J_x - J_z)\dot{\phi}\dot{\psi} + J_{xz}(\dot{\phi}^2 - \dot{\psi}^2) + m(d_{m,z}[\dot{z}\dot{\theta} - \dot{\psi}\dot{y}] + d_{m,x}[\dot{\theta}\dot{x} - \dot{y}\dot{\phi}]) \tag{2.8}$$

$$D_N = (J_y - J_x)\dot{\theta}\dot{\phi} + J_{xz}\dot{\theta}\dot{\psi} + m d_{m,x}(\dot{x}\dot{\psi} - \dot{\phi}\dot{z}) \tag{2.9}$$

where the D_X, D_Y, and D_Z are the components for the x, y, and z axes, respectively. Additionally, D_L, D_M, and D_N are the components for the rotations around the x, y, and z axes, respectively.

The aerodynamic terms are represented by the 6x1 vector A_R [23].

$$A_X = \hat{P}[C_{X1}\cos^2(\alpha)\cos^2(\beta) + C_{X2}\sin(2\alpha)\sin\left(\frac{\alpha}{2}\right)] \tag{2.10}$$

$$A_Y = \hat{P}[C_{Y1}\cos\left(\frac{\beta}{2}\right)\cos(2\beta) + C_{Y2}\sin(2\beta) + C_{Y3}\sin\beta\sin|\beta|] \tag{2.11}$$

$$A_Z = \hat{P}[C_{Z1}\cos\left(\frac{\alpha}{2}\right)\cos(2\alpha) + C_{Z2}\sin(2\alpha) + C_{Z3}\sin\alpha\sin|\alpha|] \tag{2.12}$$

$$A_L = \hat{P}[C_{L1}\sin(\beta)\sin|\beta|] + \frac{1}{2}\rho_a C_{L2}\dot{\phi}|\dot{\phi}| \tag{2.13}$$

$$A_M = \hat{P}[C_{M1}\cos\left(\frac{\alpha}{2}\right)\sin(2\alpha) + C_{M2}\sin(2\alpha) + C_{M3}\sin(\alpha)\sin|\alpha|] + \frac{1}{2}\rho_a C_{M4}\dot{\theta}|\dot{\theta} \tag{2.14}$$

$$A_N = \hat{P}[C_{N1}\cos\left(\frac{\beta}{2}\right)\sin(2\beta) + C_{N2}\sin(2\beta) + C_{N3}\sin(\beta)\sin|\beta|] + \frac{1}{2}\rho_a C_{N4}\dot{\psi}|\dot{\psi} \tag{2.15}$$

where the A_X, A_Y, and A_Z are the translational aerodynamics forces, and A_L, A_M, and A_N are the rotational aerodynamics moments.

The dynamic pressure is expressed as $\hat{P}$. The angle of attack and the sideslip angle relative to air are α and β, respectively. These terms can be expressed as follows:

$$\hat{P} = \frac{1}{2}\rho_a v_0^2 \tag{2.16}$$

$$\alpha = arctan\left(\frac{\dot{z}}{\dot{x}}\right) \tag{2.17}$$

$$\beta = arcsin\left(\frac{\dot{y}}{\sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}}\right) \tag{2.18}$$

The gravitational and buoyancy forces and moments due to these terms are represented by the 6x1 vector G [23].

$$G_X = (m - \rho_a V)g_1 \tag{2.19}$$

$$G_Y = (m - \rho_a V)g_2 \tag{2.20}$$

$$G_Z = (m - \rho_a V) g_3 \tag{2.21}$$

$$G_L = -m d_{mz} g_2 \tag{2.22}$$

$$G_M = m(d_{mz} g_1 - d_{mx} g_3) \tag{2.23}$$

$$G_N = m d_{mx} m g_2 \tag{2.24}$$

where the G_X, G_Y, and G_Z are the gravitational and buoyancy forces for the x, y, and z axes, respectively. Moreover, G_L, G_M, and G_N are the moments produced by these forces about the x, y, and z axes, respectively.

The uncertainties are expressed by the 6x1 vector E_R [23].

$$E_R = \sigma_2 (A_R + U_R)^T \tag{2.25}$$

where the 6x1 vector σ_2 is composed by random percentages around $\pm 10\,\%$.

The input forces are expressed by the 6x1 vector U_R [23].

$$U_R = \begin{bmatrix} T_c \\ 0 \\ 0 \\ 0 \\ d_{c,z} T_c \\ 0 \end{bmatrix} - \begin{bmatrix} g \\ d_{CB} g \end{bmatrix} \rho_a V_{b,f} - \begin{bmatrix} g \\ d_{CB} g \end{bmatrix} \rho_a V_{b,r} + \begin{bmatrix} 0 \\ T_{t,t} \\ 0 \\ d_{f,z} T_{t,t} \\ 0 \\ -d_{f,x} T_{t,t} \end{bmatrix} + \begin{bmatrix} 0 \\ T_{t,b} \\ 0 \\ -d_{f,z} T_{t,b} \\ 0 \\ -d_{f,x} T_{t,b} \end{bmatrix} \tag{2.26}$$

The generated thrust by the motors-propellers located at the gondola, top of the tail, and bottom of the tail is represented by $T_c, T_{t,t}$, and $T_{t,b}$, respectively. The

air ballasts volume located at the front and rear are expressed by $V_{b,f}$ and $V_{b,r}$, respectively. The skew symmetric matrix d_{CB} contains the distances between envelope CV and each ballonet CV.

2.2.2 Multi-body Modelling of Aircraft

Multi-body autonomous aircraft are becoming more common due to the new applications of UAV's. Therefore, multi-body models have been derived for different systems such as spacecrafts, quadcopters, hybrid (tilt and fixed wing), and airships. There are a number of different approaches used for the aforementioned models: Netwon-Euler, Kirchhoff, and Boltzmann-Hamel.

Hybrid vehicles have the advantages of both the rotor and fixed wing aircraft. They possess the vertical take-off/landing (VTOL) capabilities and the cruise performance for long endurance missions. Li et al. present a multi-body tilt UAV system [24]. composed of three bodies: the fuselage and two shafts equipped with motor propellers. The dynamic model is obtained using the Newton approach, meaning that the Newton-Euler equation is applied three times, one time per body. In some cases, this approach can be tedious, especially if the system contains complex kinematics or many bodies. Wang, Zhou, and Lin developed a UAV model with transitional capabilities [25]. This vehicle has two sets of motors: one at the front and one at the back. This configuration allows the aircraft to have VTOL and fixed-wing capabilities. Beloti, Santos, and Sarcinelli describe the case study of three quadcopters collaboratively transporting a load [26]. The UAV model was also derived using the Newton-Euler approach. The payload rotations along the x and y axes are restricted, due to the triangle formation that holds the load. Therefore, the payload position was estimated based on the assumption that the UAVs were connected between them, allowing the position to be estimated geometrically.

Kirchhoff equations are used to describe the dynamics of a rigid body interacting within an ideal fluid (incompressible, inviscid, and unbounded) [27]. The generalized Kirchhoff's equations considering external forces and moments can be expressed in

the Lagrangian form as [28]:

$$\frac{d}{dt}\frac{\partial T}{\partial v} + \hat{w}\frac{\partial T}{\partial v} = F_t \tag{2.27}$$

$$\frac{d}{dt}\frac{\partial T}{\partial w} + \hat{w}\frac{\partial T}{\partial w} + \hat{v}\frac{\partial T}{\partial v} = \tau_r \tag{2.28}$$

Reyhanoglu and Rehan described the model of a quadcopter carrying a 2 DOF manipulator [29]. The Kirchoff's equations were used to get all eight equations of motion for the system. Abdallah et al [30] derived a multi-body model of an airship carrying a cable driven robot as the payload. Due to the coupling dynamics between both systems, the whole system was modelled as an interconnection of lower order subsystems. Therefore, the equations of motion were found as two isolated systems and the motion of one system was assumed as external disturbance for the other one. The dynamics of the first subsystem (airship) were derived by the Kirchoff equations while the dynamics of the cable-driven robot (the carrying manipulator dynamic) were obtained using the Newton-Euler method.

Boltzmann-Hamel equations are useful for systems with nonholonomic constraints and/or quasi-coordinates [31]. This approach can produce the correct equations of motion, but it adds some complexity. The generalized form of the Boltzmann-Hamel equation for a system with m nonholonomic constraints can be expressed as follows [32]:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial u_r}\right) - \sum_{i=1}^{n}\frac{\partial L}{\partial q_i}\Phi_{ir} + \sum_{j=1}^{n}\sum_{l=1}^{n-m}\frac{\partial L}{\partial u_j}\gamma_{rl}^{j}u_l + \sum_{j=1}^{n}\frac{\partial L}{\partial u_j}\gamma_{r}^{j} = F_r \quad ; r = 1, ..., n-m \tag{2.29}$$

$$\gamma_{rl}^{j}(q,t) = \sum_{i=1}^{n}\sum_{k=1}^{n}\left(\frac{\partial \Psi_{ji}}{\partial q_k} - \frac{\partial \Psi_{jk}}{\partial q_i}\right)\Phi_{kl}\Phi_{ir} \tag{2.30}$$

$$\gamma_{r}^{j}(q,t) = \sum_{i=1}^{n}\sum_{k=1}^{n}\left(\frac{\partial \Psi_{ji}}{\partial q_k} - \frac{\partial \Psi_{jk}}{\partial q_i}\right)\Phi_{kt}\Phi_{ir} + \sum_{i=1}^{n}\left(\frac{\partial \Psi_{ji}}{\partial t} - \frac{\partial \Psi_{jt}}{\partial q_i}\right)\Phi_{ir} \tag{2.31}$$

$$F_r = \sum_{i=1}^{n} Q_i \Phi_{ir} \tag{2.32}$$

The additional complexity caused by using this approach can be observed in Equation 2.29. However, thanks to the different mathematical software that has become recently available, it is now feasible to implement. For instance, in Grauer et al., the equations of motion for an ornithopter were developed [33] for an aircraft composed of five bodies. The equations were derived using the Boltzmann-Hamel equations, but before applying this approach, the kinetic energy was computed. For the aerodynamic model, a simple quasi-steady thin airfoil was used, since there is not yet a developed model for this robot. Due to the length of the final equations, they are not shown in the paper. This is one of the characteristics of the Boltzmann-Hamel method. Similarly, a multi-body model for an autonomous airship has been developed by Lanteigne et al. [34]. The system is composed of two bodies: the envelope and the gondola, where the gondola has the capability of sliding along a keel, producing CG changes. The equations of motion were simplified to the longitudinal plane. This model considered the envelope centre of gravity (CG) located coincident with the CV. However, this assumption is a theoretical assumption that the components required to achieve the sliding motion (e.g., the keel) will affect the CG of the envelope.

2.3 Airship Control

This section will review important contributions to the field of unmanned airship control with a particular focus on resilience to disturbances, applicability to multi-body systems, and experimental performance.

The main categories of unmanned airship controllers include: proportional-integral-derivative (PID), sliding mode control (SMC), model predictive control (MPC), adaptive control, and intelligent controllers.

The PID method has been tested in this system [11,35,36] where Suvarna, Chung, and Pant [35] designed three PID regulators to control the altitude, pitch and velocity.

The simulation was implemented using the Robotic Operative System (ROS). The desired trajectory is defined as follows: first, the airship must reach a 10 m altitude, then it moves forward for 20 m, and finally it descends. The reference altitude is reached in approximately 120 s. This PID regulator is able to hold the height with almost no overshoot. On the other hand, the PID regulator for pitch angle has to maintain a 0° angle. However, it presents small oscillations within a range of ±1°. Lanteigne et al. [11, 36] proposed the PID technique to regulate the pitch angle of the airship. The system was linearized under the following conditions: 180 m altitude, and 0.1 N thrust. The PID method has certain advantages in regard to the implementation and the tuning process. The controller can reach a desired pitch angle in 25 s, with a maximum error of 2.1°. Although the PID method generally performs well when it is working close to the linear region, there are many factors that can affect this zone, for example unknown parameters, parameter changes, disturbances and uncertainties. For this reason, most studies focused on the implementation of more advanced techniques.

SMC is a nonlinear control technique well known for its robustness [37, 38]. Its design includes two steps [38]. The first is the design of the sliding surface, who's function is to drive the system's state trajectory and keep it onto a pre-defined surface (chosen by the designer) [37]. This surface is also known as the switching surface, because the control signal has two possible gains, and these gains are switched based on if the state trajectory is above or below the sliding surface [37]. The second step is the design of a controller that guarantees the sliding behavior [38]. De Paiva, Benjovengo, and Siqueira [39] present a SMC model for path tracking. The system was linearized around an equilibrium point and the assumed conditions were a straight flight and no wind disturbance [39]. As a result of the linearization process, the system dynamics of the system is decoupled between two motions: longitudinal and lateral. The system's stability was proven via the Lyapunov method. The chattering phenomenon was reduced by replacing the signum function with the hyperbolic tangent function [39]. The tracking controller was tested under a square trajectory. In general, the system performed well, with no overshoot, when turning the corners.

These abrupt changes in direction were caused by the low airspeed conditions, which produced high changes in the yaw angle. In addition, saturation in the rudder control signal was observed. A possible means of improving the system's performance would be to use smoother transitions in the desired trajectory to avoid abrupt changes in the airspeed.

The MPC control technique uses the dynamic model of a system to predict its future behavior as well as to choose the optimal control signal to minimize a given cost function [40–42]. It is composed of three components: the prediction model, the objective function, and the control law [40]. The prediction model refers to the mathematical model that describes the system dynamics [40]. Usually, the system's representation is divided into two parts: the plant model and the disturbance estimator. The objective function's role is to minimize the control signal that allows the output variable to follow the reference signal [40]. The control law aims to minimize the objective function and compute the predicted output [40]. Numerous authors have proposed this strategy for autonomous airships. Wen, Chen, Liang, and Duan [21] designed a model predictive control for a sensorless environment using an observer. The use of an observer is a common technique to estimate the value of the states of the system, in cases when it is not feasible to measure them through a sensor. The results show that the controller is able to deal with the saturation of the actuators. In addition, the technique can follow a desired set point with an overshoot of less than 10%. However, one of the main drawbacks of this approach is the high computational requirements. Therefore, the technique should be simplified so that it is feasible to use it in the implementation stage. Zhang, Yang, Deng, and Lin [43] proposed an MPC system for trajectory tracking that takes into consideration wind disturbances. The controller was designed by linearizing the equations of motion based on a small perturbation approach. The tracking performance of this technique was tested and compared to the SMC model, for a twisted pair trajectory. The results show a good performance in general for the MPC system. This means a rapid response with a settling time of less than 10 s for the yaw angle. However, it presents a small error and overshoot following the desired trajectory. In comparison, the SMC model showed a

lower performance, as it had a higher overshoot and a greater settling time at the beginning of the trajectory.

The design of control systems for aircraft motivated the development of the adaptive controller approach with varying parameters during the 1950s [44]. This is due to the fact that aircraft have a broad spectrum of operation points, based on their speed, altitude, and time varying dynamics [44]. The technique aims to adjust or adapt the controller parameters in order to keep an overall steady performance of the system under uncertainties, unknown plant parameters, and not modelled dynamics [45–47]. The adaptation law determines how to automatically change the controller parameters, according to the system's performance. The law depends on the stability analysis of the closed-loop system, which is usually proven with the Lyapunov methods [45]. Because of its adaptation capabilities, this technique is commonly combined with other types of regulators to improve the overall system performance. For instance, an adaptive sliding mode controller for trajectory tracking of an airship was reported by [15]. This method has a settling time of approximately 10 s for the angles and velocities. However, the required control signals are high, at approximately 5×10^9 Newton. This demonstrates that new simulations that include the actuator saturation should be carried out to compare the controller performance under this scenario. Another example of adaptive SMC was presented in [48]. Xiao, Wang, Zhou, and Duan developed an adaptive sliding mode control for altitude and position control of an airship. Additionally, in the design, the authors considered the input saturation, which is a common and unavoidable phenomenon for the thrusters. Including the input saturation in the design stage allows the simulation to closely resemble the physical prototype, due to the actuator limitations that are represented. However, the proposed technique only guarantees the closed loop system's stability; it is not able to guarantee unsaturated input to the actuators. The simulation results show that the controller reached the desired roll and pitch angle in approximately 5 s, with a small overshoot. In addition, the reference velocities were reached in less than 15 s. However, the adaptation algorithm could take more time to converge, especially if the initial conditions are far from the trim point. As a conclusion, it has been

demonstrated that the adaptive controllers are suitable for this type of aircraft, since airships have certain parameters that are difficult to measure or estimate with high accuracy (i.e., aerodynamic coefficients and inertia matrix).

One of the latest approaches to emerge is the use of intelligent controllers [47]. These controllers are able to adapt, learn, and take decisions for a specific task. Two of the most well-known techniques from this category are fuzzy logic (FL) and neural networks (NN) [47, 49]. Fuzzy logic aims to allow a computer to imitate human decision-making, based on previous experiences [49]. The process used by a fuzzy controller includes three steps: fuzzy quantification, fuzzy control rules, and fuzzy decisions. At the first step, the control variables are converted to a fuzzy quantity. The second step involves the rules being defined by the designer. Basically, these rules are a set of "If ... and Then ..." conditions. In the third step, the input is mapped according to the rules, in order to compute the control signal. Finally, the output from FL is converted into an understandable signal for the actuators. This step is called as non-fuzzy process (clarification, deblurring, and defuzzification) [49]. Yu, Xu, Zhong, and Zhu [14] combined FL with MPC controllers for path following of an autonomous airship. The authors used FL as an adaptation mechanism to estimate the B matrix of the system, taking into account the uncertainties. The B matrix is used in the control law of the MPC. This method was tested with a circle path. The results show that the adaptive fuzzy-MPC is 50 s faster than the non-adaptive approach. However, the simulation results also show that the error deflection for the control surfaces (rudder and elevators) has values of around -600° during the first 10 s. This behaviour should be improved as it means that the controller or guidance algorithm is setting reference angles beyond the limits of the surface, causing actuator saturation, which can, in turn, produce system oscillations. Similarly, Yang, Wu, and Zheng [50] described an adaptive FL backstepping controller, where fuzzy logic is responsible for estimating the uncertain terms of the model. The closed-loop stability was proven with the Lyapunov theory. The controller was tested under two scenarios: 1. known parameters 2. parameter uncertainties and wind disturbances. The settling time for both cases was around 300 s. This demonstrates the system's ability to

response under disturbances. However, the control signals show undesired oscillations that could cause problems in the actuators during the implementation stage. It has therefore been demonstrated that FL can increase the system's robustness, especially when it is combined with other techniques. In addition, it is an alternative approach that can be used to estimate the model uncertainties or unknown terms, during a flight. However, the controller could have certain aggressive behaviours that may cause negative effects in the actuators. This issue could be dealt with by softening the fuzzy rules, thus ensuring smooth behaviour.

The second intelligent controller used for autonomous airships takes advantage of neural networks (NNs). NNs are based on human reasoning and utilize machine learning principles to adapt and learn, in order to solve a specific problem [47]. Usually, the internal structure of a NN is composed by nodes, which are interconnected to create a grid [49]. This allows for a great nonlinear mapping ability between a given input and output [49], thus enabling neural networks to identify the different patterns and characteristics of a system. For these reasons, NNs have good potential when used in control systems as a controller, an adaptation mechanism or for parameter estimation. Zheng and Zhou [51] present a backstepping technique that uses a robust and adaptive radial basis function neural network to compensate for the system's uncertainties for path following and improve the tracking performance of the autonomous airship. The system was evaluated for a straight line and for a circular path, and the settling time in the simulation was around 50 s. The system exhibited good performance for wind disturbances and parameter uncertainties. Moreover, Yang and Yan [13] simulated a combination between a SMC and neural network to adjust the control law. The proposed technique was simulated and compared to the traditional SMC method, for an ascending trajectory. The results show that both controllers have similar settling times of approximately 60 s. However, the controller with NN exhibits less chattering phenomenon than the traditional SMC method. In addition, all the controllers' parameters converge on average at less than 150 s. Nevertheless, some parameter values present some oscillations during the first 50 s. This could be caused by the fact that the NN requires a certain amount of time

to start self-learning. The adapting, estimating, and robustness capabilities of NNs are demonstrated in the literature. Nonetheless, NN performance is strongly related to its internal structure (i.e., number of layers). Therefore, if the NN is more complex, it will demand more computer processing, which could be a drawback during the implementation stage.

We must take into consideration that the airship used in this project is the next generation of the aircraft presented in [11]. This vehicle is capable of changing its centre of gravity during its flight. For this reason, a review of some control strategies used for similar aerial systems with this ability were done. Palunko [52] proposed an adaptive control through linearization, for a quadcopter with dynamic changes in the CG. This capacity increases the manoeuvrability of the vehicle. In addition, the results show that the PID controller does not perform well due to these changes in the system's parameters. However, the adaptive controller does show robustness. An equivalent work was designed by Ibarra [53] for the control of roll, pitch and yaw angles using adaptive control. Correspondingly, Yildiz [54] presents a cascade model reference adaptive control (MRAC) for a reconfigurable quad tilt-wing aircraft. This controller is able to deal with unknown parameters, unexpected failures and damages. Altan [55] proposed a model reference adaptive control for the load transporting system placed on a quadrotor. The parameters of this system vary since it has to leave payloads at different target positions.

In the case of multi-body unmanned aerial vehicles, Alvarez [56] developed a nonlinear controller for a system composed of a quadcopter carrying a manipulator arm. The robotic arm movement produces changes in the centre of gravity, thus destabilizing the whole system. Nevertheless, the controller shows a robust performance under these conditions. Reyhanoglu [29] proposed a backstepping regulator for the same scenario. De Almeida [57] designed a cascade nonlinear controller for a system integrated by a quadrotor and an inverted pendulum. The technique used was feedback linearization to control the desired position. Tartaglione [58] used a MPC controller combined with PID controller for a multibody slung load system. The system is

composed of a group of multicopters, which cooperate to carry a load.

2.4 Experimental Implementation of Autonomous Airships

As previously discussed, most of the works mentioned above have been validated only through simulation tools, such as Matlab Simulink or ROS. There are few papers that present a prototype validation. This makes these projects more valuable, as the authors had to deal with the feasibility and the experimental issues during the implementation stage. Therefore, a review of relevant literature containing an experimental approach is also presented.

Pshikhopov et al. [59] developed the autonomous airship shown in Figure 2.3. This vehicle has the following features: a volume of 2000 m^3 and a length of 40 m. It has two motors as well as aerodynamic control surfaces. The airship is controlled using a robust controller that is based on the Lyapunov theory [60]. This airship communicates through telemetry, for visualization purposes. NUMECA software [61] was used to estimate the aerodynamic coefficients. Using simulation software for this process reduces the costs. The errors of the control system are: 27 m for path tracking and 1.5 m/s for the velocity. The vehicle does not perform well in terms of trajectory tracking, due to GPS limitations when estimating global coordinates in an indoor environment [12].

Fedorenko and Krukhmalev [12] fabricated the airship model shown in Figure 2.1a. The vehicle is 3.78 m long and has a volume of 5.2 m^3. The controller strategy used was robust control and it was designed via the direct Lyapunov method. Since GPS exhibits limited and weak signals in indoor environments [12], this particular model uses a 3D Compass and an optical flow camera sensor to estimate the airship's position. The details of this conversion process are described in [12]. As part of the implementation process, the author calculated a relation between the pulse width modulation (PWM) and the thrust. A polynomial function of this relation was ob-

Figure 2.3: Sterkh autonomous airship model by Pshikhopov et al [59]

tained using experimental data. MAVlink, a well-recognized protocol for UAVs, was used to enable the drone to communicate with the ground station. The whole system was implemented using ROS framework. Results show that the vehicle can track a desired trajectory. However, the system does present some oscillations that could be caused by the heavy weight of the blimp [12].

Liesk, Nahon, and Boulet [62] implemented a backstepping/Lyapunov controller in the prototype shown in Figure 2.4. This vehicle has an ellipsoidal design and is finless. Experiments were carried out in a 120 m by 80 m field. The goal was to control the altitude and speed of the blimp. The results showed that the tracking performance for the roll, pitch, and yaw was fairly good. It was able to reach set points between $\pm 180°$. However, the height controller presented an unsatisfactory behaviour, with a ± 1.5 m error. This could have been caused by inaccurate measurements given by the LiDAR sensor used to estimate the aircraft's altitude. Finally, the velocity controller had an error of approximately ± 0.5 m/s. The airship allows for manual control by a pilot, in case of a failure in the GPS system. The aircraft was able to follow the desired trajectory, but at times its performance decreased due to GPS limitations. False GPS measurements and losses in the signal affected the control system [62].

Figure 2.4: Quanser MkII unmanned airship by Liesk et al. [62]

Liu, Sang, and Jin [63] tested the experimental model shown in Figure 2.5. The vehicle is 25 m long and has a volume of 750 m^3. The prototype used an MPC controller, whose design included a dynamic model, linearized about a trim state.The Lyapunov theory was applied in order to prove the prototype's stability. The system was able to track a desired trajectory with an error of less than 5 m. The error was caused by the equipment (IMU sensor) used to estimate the position as well as by wind disturbances. The steady state error was less than 5° for the roll and ±5° for the yaw. However, the pitch angle presented some oscillations and an overshoot that could reach 20°. This could be due to the saturation of the elevators at a low-speed flight. Finally, the controller's performance under model uncertainties and failures should be included in the design and simulation, in order obtain the information needed to increase the drone's performance.

Wang, Zheng, Efimov, and Perruquetti [64, 65] developed two different prototypes. The first one [64] has a length of 132 cm and a diameter of 94 cm. It [64] has a predictor-based controller with an observer to control the altitude of the system. The communication between the drone and the ground station is done using XBee (Ra-

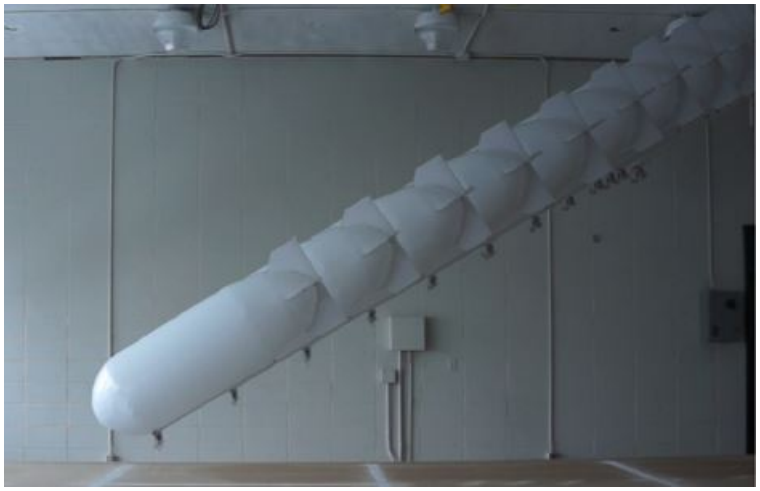
Figure 2.5: Zhiyuan-1 unmanned airship by Liu et al. [63]

diofrequency). Results show that there is approximately a thirty percent overshoot and a certain number of oscillations in the system's response. This can be caused by the difference between gravity and buoyancy [64]. This controller needs to be updated to improve its airship response. The second prototype [65] has a length of 105 cm and a diameter of 71 cm. The control technique [65] used is an output feedback controller with an observer to compensate for disturbances. The regulator was able to control the X, Y and yaw angles while the compensator strategy performed well, with a less than 10 cm error. A position capturing system (Optitrack) was used to estimate with high accuracy the position of the blimp an indoor environment.

Lanteigne et al. [11, 20] has worked on two different generations of autonomous airships. The first vehicle was based on a commercially available blimp as shown in Figure 2.6a [20]. The vehicle has a spheroid shape with a radius of 50 cm and is equipped with two DC micromotors and propellers. This work presents an optimal control for trajectory planning. The prototype was tested in an open loop. However, since in most cases the motors do not move at the same speed, a feedback controller was required to guarantee an adequate performance under different conditions (i.e., moving along a straight line). The drag coefficient was computed by experimental tests. For this, a known input was applied to the system and the response was measured using a vision tracking system [20, 66]. The second generation is shown in Figure 2.6b [11]. This prototype is an ellipsoidal blimp with a volume of 0.39 m^3

and a length of 1.83 m. A sliding gondola mechanism was proposed as an alternative for ballonets, in order to achieve changes in CG. The aircraft is equipped with a Nanowii Flight board, a bluetooth module and an inertial measurement unit. A Matlab interface is used as a ground station. The non-linear equations of motion were linearized around an equilibrium point [36]. A PID controller was designed to control the resulting linearized model. However, the operating zone of this regulator only works for small perturbations around the trim point of $\pm 5°$ angles along a straight trajectory [36]. This issue can be improved by implementing a nonlinear control technique such as one of those mentioned previously. Finally, the use of a gondola to enable faster manoeuvres was successfully proven with this prototype experiment.

(a) First generation [20]



(b) Second generation [11]

Figure 2.6: Autonomous airships by Lanteigne et al

Chapter 3

Multi-Body Modelling of a Reconfigurable Airship

The modelled prototype is shown in Figure 1.2 of Section 1.2. The airship has a keel attached to the envelope, and the gondola slides along the keel. This movement allows for pitch variations that are useful for landing manoeuvres.

In order to simplify the modelling process, the following assumptions are taken into account:

- The airship is assumed to be a rigid body with a constant volume. Therefore, the aeroelastic effects are ignored.
- The airship's virtual masses and inertia, owing to the large volume of air displaced by aircraft, are considered.
- The centre of volume (CV) coincides with the centre of buoyancy (CB).

3.1 Kinematics

The system is composed of two bodies. The first one, m_1, includes the envelope, helium, rail and fins. The second one, m_2, refers to the gondola. As it is shown in

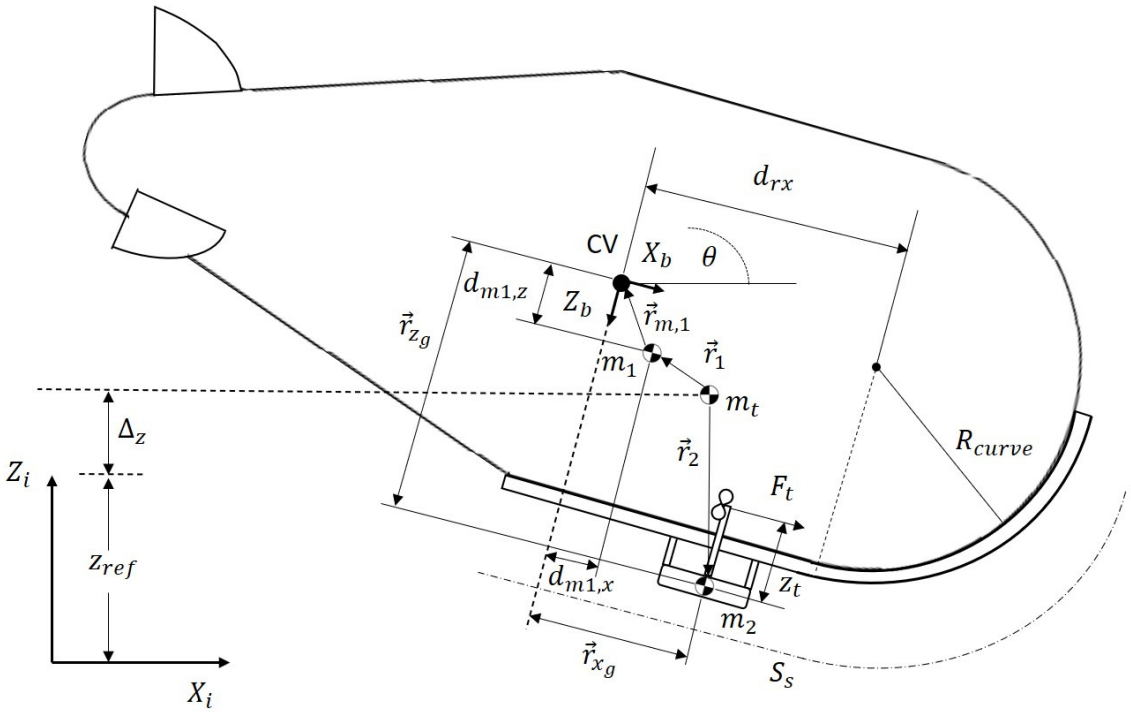


Figure 3.1: Airship coordinate system

Figure 3.1, the body reference frame $[X_b \quad Y_b \quad Z_b]$ is fixed to the centre of volume and it is not coincident with the CG of m_1. The generalized coordinates expressed in the inertial reference frame $[X_i \quad Y_i \quad Z_i]$ are described by:

$$\vec{q} = [x \quad y \quad z \quad \phi \quad \theta \quad \psi \quad S_s]^T \tag{3.1}$$

Airships operate at a specified altitude that is determined by the buoyancy force. The altitude of the total CG is given by:

$$z = \Delta_z + z_{ref} \tag{3.2}$$

Euler angles can be used to express the body rotation relative to either the body frame or the inertial reference frame. The rotations are denoted by roll (ϕ), pitch (θ), and yaw (ψ). The rotational matrix to transform the coordinates from the body reference frame to the inertial frame can be computed as:

$$R_1 = R_z(\psi)R_y(\theta)R_x(\phi) \tag{3.3}$$

where:

$$\mathrm{R_x}(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\phi) & -sin(\phi) \\ 0 & sin(\phi) & cos(\phi) \end{bmatrix} \tag{3.4}$$

$$\mathrm{R_y}(\theta) = \begin{bmatrix} cos(\theta) & 0 & sin(\theta) \\ 0 & 1 & 0 \\ -sin(\theta) & 0 & cos(\theta) \end{bmatrix} \tag{3.5}$$

$$\mathrm{R_z}(\psi) = \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.6}$$

Solving (3.3), the rotational matrix becomes:

$$\mathrm{R_1} = \begin{bmatrix} cos(\psi)cos(\theta) & -sin(\psi)cos(\phi) + cos(\psi)sin(\theta)sin(\phi) & sin(\psi)sin(\phi) + cos(\psi)sin(\theta)cos(\phi) \\ -sin(\psi)cos(\theta) & -cos(\psi)cos(\phi) - sin(\psi)sin(\theta)sin(\phi) & cos(\psi)sin(\phi) - sin(\psi)sin(\theta)cos(\phi) \\ sin(\theta) & -cos(\theta)sin(\phi) & -cos(\theta)cos(\phi) \end{bmatrix} \tag{3.7}$$

$\mathrm{R_1}$ can also be used to transform the coordinates from the inertial reference frame to the body frame. For this, it is necessary to compute its inverse ($\mathrm{R_1^{-1}}$). Since $\mathrm{R_1}$ is an orthogonal matrix its inverse is equal to its transpose ($\mathrm{R_1^{-1}} = \mathrm{R_1^T}$).

The rate of change of the Euler angles are: $[\dot{\phi} \quad \dot{\theta} \quad \dot{\psi}]^T$. On the other hand, the angular speeds described in the body frame are: $[p \quad q \quad r]^T$. Therefore, the rotational matrix that transforms the angular velocities from the inertial frame to the body fixed frame can be written as:

$$\mathrm{R_2} = \begin{bmatrix} 1 & 0 & -sin(\theta) \\ 0 & cos(\phi) & sin(\phi)cos(\theta) \\ 0 & -sin(\phi) & cos(\phi)cos(\theta) \end{bmatrix} \tag{3.8}$$

Similarly, R_2 matrix can be used to transform the angular speeds from the body frame to the rate of change of the Euler angles, computing its inverse. Consequently, the rotational matrix becomes:

$$R_2{}^{-1} = \begin{bmatrix} 1 & sin(\phi)tan(\theta) & cos(\phi)tan(\theta) \\ 0 & cos(\phi) & -sin(\phi) \\ 0 & \frac{sin(\phi)}{cos(\theta)} & \frac{cos(\phi)}{cos(\theta)} \end{bmatrix} \tag{3.9}$$

The coordinates of m_1 and m_2 CG relative to the vehicle CG expressed in the inertial frame can be determined by solving the following equations:

$$\vec{r}_1 + \vec{r}_{m,1} + \vec{r}_{z_g} + \vec{r}_{x_g} = \vec{r}_2 \tag{3.10}$$

$$\vec{r}_1 m_1 + \vec{r}_2 m_2 = 0 \tag{3.11}$$

where:

$$\begin{aligned} \vec{r}_{m,1} &= R_1[-d_{m1,x} \quad 0 \quad -d_{m1,z}]^T \\ &= \begin{bmatrix} -d_{m1z}(sin(\phi)sin(\psi) + cos(\phi)cos(\psi)sin(\theta)) - d_{m1x}cos(\psi)cos(\theta) \\ -d_{m1z}(cos(\psi)sin(\phi) - cos(\phi)sin(\psi)sin(\theta)) + d_{m1x}cos(\theta)sin(\psi) \\ -d_{m1x}sin(\theta) + d_{m1z}cos(\phi)cos(\theta) \end{bmatrix} \end{aligned} \tag{3.12}$$

$$\vec{r}_{z_g} = R_1[0 \quad 0 \quad z_g]^T = z_g \begin{bmatrix} (sin(\phi)sin(\psi) + cos(\phi)cos(\psi)sin(\theta)) \\ (cos(\psi)sin(\phi) - cos(\phi)sin(\psi)sin(\theta)) \\ -cos(\phi)cos(\theta) \end{bmatrix} \tag{3.13}$$

$$\vec{r}_{x_g} = R_1[x_g \quad 0 \quad 0]^T = x_g \begin{bmatrix} cos(\psi)cos(\theta) \\ -cos(\theta)sin(\psi) \\ sin(\theta) \end{bmatrix} \tag{3.14}$$

Substituting (3.12), (3.13) and (3.14) in (3.10) and simultaneously solving (3.10) and (3.11):

$$r_{1x} = -m_2 \frac{a_{r1x,1} + a_{r1x,2}}{(m_1 + m_2)} \tag{3.15}$$

$$r_{1y} = m_2 \frac{a_{r1y,1} + a_{r1y,2}}{(m_1 + m_2)} \tag{3.16}$$

$$r_{1z} = m_2 \frac{a_{r1z,1} + a_{r1z,2}}{(m_1 + m_2)} \tag{3.17}$$

$$r_{2x} = -m_1 \frac{a_{r2x,1} + a_{r2x,2}}{(m_1 + m_2)} \tag{3.18}$$

$$r_{2y} = -m_1 \frac{a_{r2y,1} + a_{r2y,2}}{(m_1 + m_2)} \tag{3.19}$$

$$r_{2z} = -m_1 \frac{a_{r2z,1} + a_{r2z,2}}{(m_1 + m_2)} \tag{3.20}$$

where the coefficients a are function of the gondola's position, and the position of the vehicle CG. The equations for these coefficients are presented in Appendix A.

The expression z_g can take two different values according to the gondola's position. Consequently, z_g can be determined as:

$$z_g = \begin{cases} -0.27 \geq S_s \geq 0.73 \;\; ; \;\; l \\ 0.73 < S_s \leq 2.14 \;\; ; \;\; \sqrt{(R_{curve})^2 - (sin\left(\frac{S_s - d_{rx}}{R_{curve}}\right) R_{curve})^2} \end{cases} \tag{3.21}$$

Similarly, x_g will be the same travelled distance by the gondola until the gondola reaches the curve trajectory of the rail. Consequently, x_g can be determined as:

$$x_g = \begin{cases} -0.27 \geq S_s \geq 0.73 \;\; ; \;\; S_s \\ 0.73 < S_s \leq 2.14 \;\; ; \;\; d_{rx} + sin\left(\frac{S_s - d_{rx}}{R_{curve}}\right) R_{curve} \end{cases} \tag{3.22}$$

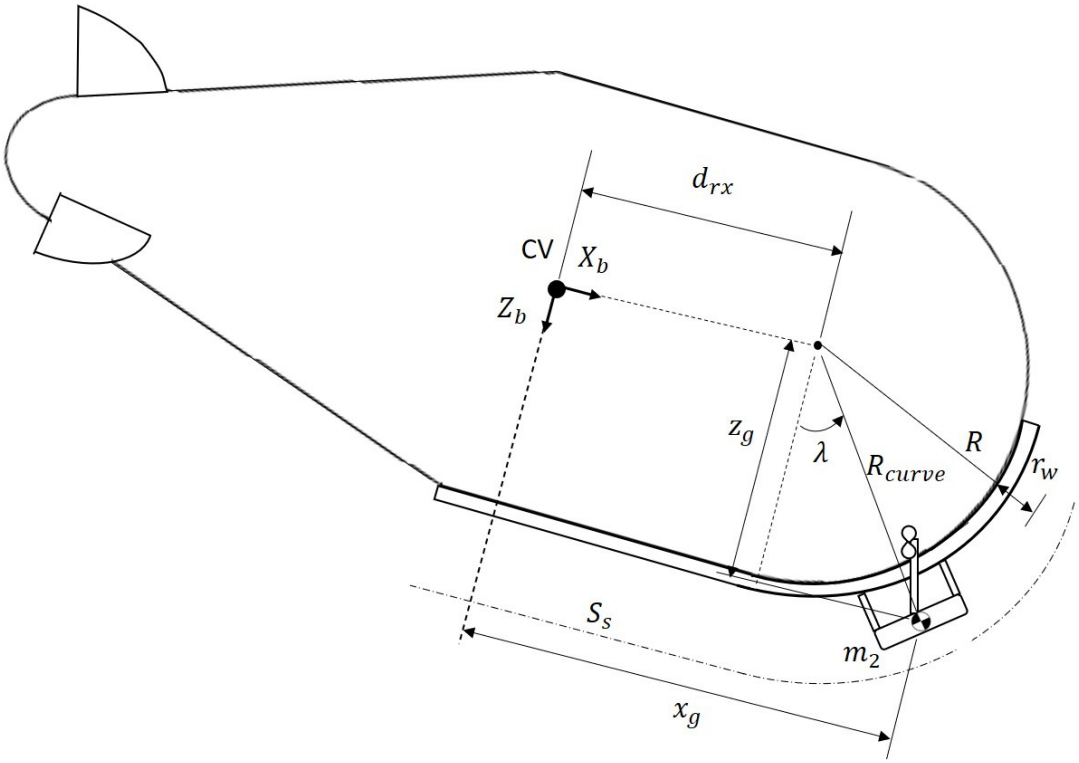


Figure 3.2: Gondola in the curve trajectory

where d_{rx} represents the distance between the body frame and the centre of circumference along the x, R_{curve} is the radius of the curve. λ represents the angle between the gondola position and the centre of the circumference measure from a parallel line to the z body frame axis. The expressions for the curve case for z_g and x_g were found based on geometry using the Figure 3.2.

Table 3.1 shows the distance parameters that describe the gondola's position over the rail, as measured from the prototype.

Table 3.1: Parameters for gondola over the straight line and curve rail condition

Symbol	Parameter	Value	Units
d_{rx}	Distance between the centre of the circumference and the origin of the body frame, along the x axis	0.73	m
l	Distance between the CV and the gondola CG, along the z axis in the body frame	0.94	m
R	Radius of the curve	0.9	m
r_w	Rail thickness + added material	0.045	m

3.2 Mass Matrix

The first body represents the envelope and the rail. Therefore, once the envelope is filled with the buoyant gas, the total mass of the first body can be computed as:

$$m_1 = m_{envelope} + m_{helium} + m_{rail} + m_{fins} \tag{3.23}$$

The mass m_2 is that of the gondola and includes all electrical components, batteries, speed controllers, and communication equipment. Then, the total mass of the system is:

$$m_t = m_1 + m_2 \tag{3.24}$$

The 6x6 mass matrix can be expressed as:

$$\mathrm{M} = \begin{bmatrix} \mathrm{M_a} & -m_t d_{CG}^T \\ -m_t d_{CG} & \mathrm{J_a} \end{bmatrix} \tag{3.25}$$

where $\mathrm{d_{CG}}$ is a skew symmetric matrix that contains the distances between the CG and the CV.

$$\mathrm{d_{CG}} = \begin{bmatrix} 0 & -m_t d_{m,z} & 0 \\ m_t d_{m,z} & 0 & -m_t d_{m,x} \\ 0 & m_t d_{m,x} & 0 \end{bmatrix} \tag{3.26}$$

The CG coordinate along the x axis can be computed as: $\frac{\sum_i m_i x_i}{m_t}$. After applying the equation:

$$d_{m,x} = \frac{x_g m_2 + d_{m1,x} m_1}{m_t} \tag{3.27}$$

where x_g represents the distance of the gondola CG along the x axis, relative to the body frame fixed to the CV. $d_{m1,x}$ refers to the distance of the Body 1 CG along the x axis, relative to the same frame.

Similarly, for the z axis: $\frac{\sum_i m_i z_i}{m_t}$. After solving the equation:

$$d_{m,z} = \frac{z_g m_2 + d_{m1,z} m_1}{m_t} \tag{3.28}$$

where z_g represents the distance of the gondola CG along z axis, relative to the body frame fixed to the CV. $d_{m1,z}$ refers to the distance of the body 2 CG along the z axis, relative to the same frame.

$\mathrm{M_a}$ contains the mass of the total physical system plus the added mass. $\mathrm{J_a}$ contains the total inertial of the physical system plus the added inertia. Therefore, $\mathrm{M_a}$ and $\mathrm{J_a}$ can be expressed as:

$$\mathrm{M_a} = \begin{bmatrix} m_x & 0 & 0 \\ 0 & m_y & 0 \\ 0 & 0 & m_z \end{bmatrix} \tag{3.29}$$

$$\mathrm{J_a} = \begin{bmatrix} J_x & 0 & -J_{xz} \\ 0 & J_y & 0 \\ -J_{xz} & 0 & J_z \end{bmatrix} \tag{3.30}$$

where m_x, m_y, m_z, J_x, J_y, and J_z can be computed based on the data provided in [67]. The provided data [67] relates the constants k_1, k_2, and k' with the thickness ratio, which depends on the geometry. A quadratic fit was used to estimate the coefficients based on the prototype's thickness ratio. The mass and inertia estimations are described in Appendix B and Appendix C, respectively.

$$m_x = (1 + k_1)(m_1 + m_2) \tag{3.31}$$

$$m_y = m_z = (1 + k_2)(m_1 + m_2) \tag{3.32}$$

$$J_x = I_{b1,x} + I_{b2,x} \tag{3.33}$$

$$J_y = (1 + k')(I_{b1,y} + I_{b2,y}) \tag{3.34}$$

$$J_z = (1 + k')(I_{b1,z} + I_{b2,z}) \tag{3.35}$$

$$J_{xz} = I_{b1,xz} + I_{b2,xz} \tag{3.36}$$

Finally, substituting (3.26), (3.30), and (3.29) in (3.25), the M matrix can be rewritten as:

$$\mathrm{M} = \begin{bmatrix} m_x & 0 & 0 & 0 & m_t d_{m,z} & 0 \\ 0 & m_y & 0 & -m_t d_{m,z} & 0 & m_t d_{m,z} \\ 0 & 0 & m_z & 0 & -m_t d_{m,x} & 0 \\ 0 & -m_t d_{m,z} & 0 & J_x & 0 & -J_{xz} \\ m_t d_{m,z} & 0 & -m_t d_{m,x} & 0 & J_y & 0 \\ 0 & m_t d_{m,x} & 0 & -J_{xz} & 0 & J_z \end{bmatrix} \tag{3.37}$$

Table 3.2 shows the mass and inertia parameters of the airship.

Table 3.2: Airship mass and inertia parameters

Symbol	Parameter	Value	Units
V	Airship volume	6.58	m^3
$m_{envelope}$	Envelope mass	1.35	kg
m_{fins}	Fins mass	0.0453	kg
m_{helium}	Helium mass	1.07	kg
m_{rail}	Rail mass	0.1944	kg
m_2	Gondola mass	3.3	kg
k_1	Lamb's inertia ratio about x	0.1664	-
k_2	Lamb's inertia ratio about y or z	0.7470	-
k'	Lamb's inertia ratio about y or z	0.3364	-
$I_{b1,x}$	Moment of inertia body 1 about x	0.846	m^2kg
$I_{b1,y}$	Moment of inertia body 1 about y	1.930	m^2kg
$I_{b1,z}$	Moment of inertia body 1 about z	1.930	m^2kg

3.3 Dynamics

3.3.1 Kinetic Energy

The total kinetic energy of the system expressed is an addition of three terms, and can be written as:

$$T = T_1 + T_2 + T_a \tag{3.38}$$

where T_1 represents the kinetic energy due to m_1. This term can be computed as:

$$T_1 = \frac{1}{2}\dot{\vec{r_1}}^T m_1 \dot{\vec{r_1}} \tag{3.39}$$

The kinetic energy due to m_2 is T_2. This term can be computed as:

$$T_2 = \frac{1}{2}\dot{\vec{r}_2}^{\,T} m_2 \dot{\vec{r}_2} \tag{3.40}$$

The kinetic energy due to the added mass is T_a. This term can be computed as:

$$T_a = \frac{1}{2}\dot{\vec{V}}^{\,T} \mathrm{M} \dot{\vec{V}} \tag{3.41}$$

where $\vec{V} = [\vec{r}\ ;\ \vec{\xi}]$ represents the system velocities. Moreover, $\vec{r} = [x\ \ y\ \ z]^T$ and $\vec{\xi} = [\phi\ \ \theta\ \ \psi]^T$ represent the translational and rotational coordinates of the system with respect to CG, respectively.

3.3.2 Potential Energy

The total potential energy of the system is an addition of two terms, and can be written as:

$$U = U_1 + U_2 \tag{3.42}$$

where U_1 represents the potential energy due to the helium envelope m_1, less the mass of the displaced air. This term can be computed as:

$$U_1 = g(m_1 - m_{air})(z + r_{1z}) \tag{3.43}$$

The blimp is intended to fly at low altitudes of around 122 m [68]. In order to compute the air pressure and mass the following assumptions were taken into account. First, the atmospheric temperature is 15°C. Second, the gradient of air density against altitude is estimated by $\Delta\rho = -1.1641 \times 10^{-4}$ kg/m^4 for low altitude flight. The estimation process of $\Delta\rho$ is explained in detail in Appendix D.

Therefore, given an altitude z, ρ_{air} can be estimated as:

$$\rho_{air} = \rho_{ref} - \Delta\rho\Delta z \tag{3.44}$$

where ρ_{ref} represents the air density at the buoyant altitude z_{ref}. Assuming a constant volume, the air mass at the reference altitude can be computed as:

$$m_{air} = \rho_{ref} V = m_t \tag{3.45}$$

Consequently, combining with (3.44), and solving for m_{air}. The air mass can be expressed as a function of m_t, as follows [34]:

$$m_{air} = \frac{\rho_{ref} - \Delta\rho\Delta z}{\rho_{ref} - \Delta\rho z_{ref}} m_t \tag{3.46}$$

The potential energy of the moving gondola m_2 is given by U_2. This term can be computed as:

$$U_2 = g m_2 (z + r_{2z}) \tag{3.47}$$

Finally, substituting (3.52), (3.53) in (3.42), the total potential energy can be rewritten as follows:

$$U = g((m_1 - m_{air})(z + r_{1z}) + m_2(z + r_{2z})) \tag{3.48}$$

3.4 Quasi-Velocities

The quasi-velocities cannot be used in the expression of kinetic energy using the Lagrange approach, since they are not integrable. Kinetic energy must be expressed in terms of the true velocities. These velocities are true in the sense that they are derivatives of the generalized coordinates [32].

For an unconstrained system (i.e., an airship), the function that relates the quasi-velocities and the time derivatives of the generalized coordinates is given by:

$$u_j = \sum_{i=1}^{n} \Psi_{ji}(q,t)\dot{q}_i + \Psi_{j,t}(q,t) \quad (j = 1, ..., n) \tag{3.49}$$

where n represents the number of degrees of freedom of the system. Comparably, the inverse relation, to transform from quasi-velocities to the first time derivative of the generalized coordinates, is expressed by:

$$\dot{q}_i = \sum_{j=1}^{m} \Phi_{ij}(q,t)u_j + \Phi_{i,j}(q,t) \quad (i = 1, ..., n) \tag{3.50}$$

In the case of a system that has m nonholonomic constraints, the last m quasi-velocities terms are set to zero. This can be expressed for the general case as follows:

$$u_j = \sum_{i=1}^{n} \Psi_{ji}(q,t)\dot{q}_i + \Psi_{q,t}(q,t) = 0 \quad (j = n - m + 1, ..., n) \tag{3.51}$$

For the case of the unmanned airship, as shown in Figure 3.1, the quasi-velocities u_{1-3} can be obtained using the transformation matrix ${R_1}^{-1}$.

$$u_1 = \dot{x}\, cos(\psi)cos(\theta) - \dot{y}\, cos(\theta)sin(\psi) + \dot{z}\, sin(\theta) \tag{3.52}$$

$$u_2 = \dot{x}\, p_1 + \dot{y}\, p_2 + \dot{z}\, p_3 \tag{3.53}$$

$$u_3 = \dot{x}\, p_4 + \dot{y}\, p_5 + \dot{z}\, p_6 \tag{3.54}$$

where:

$$p_1 = [-cos(\phi)sin(\psi) + cos(\psi)sin(\phi)sin(\theta)] \tag{3.55}$$

$$p_2 = [-cos(\phi)cos(\psi) - sin(\phi)sin(\psi)sin(\theta)] \tag{3.56}$$

$$p_3 = -cos(\theta)sin(\phi) \tag{3.57}$$

$$p_4 = [sin(\phi)sin(\psi) + cos(\phi)cos(\psi)sin(\theta)] \tag{3.58}$$

$$p_5 = [cos(\psi)sin(\phi) - cos(\phi)sin(\psi)sin(\theta)] \tag{3.59}$$

$$p_6 = -cos(\phi)sin(\theta) \tag{3.60}$$

The quasi-velocities u_{4-6} can be obtained using the transformation matrix $\mathrm{R_2}^{-1}$.

$$u_4 = \dot{\phi} - \dot{\psi} sin(\theta) \tag{3.61}$$

$$u_5 = \dot{\theta} cos(\phi) + \dot{\psi} sin(\phi) cos(\theta) \tag{3.62}$$

$$u_6 = -\dot{\theta} sin(\phi) + \dot{\psi} cos(\phi) cos(\theta) \tag{3.63}$$

There is a strong relation between the gondola position and the pitch angle [34]. Consequently, $q_5 = \theta$ was linked to $q_7 = S_s$ in the form of a nonholonomic constraint on the system. In general as it stated above, for a system with m nonholonomic constraints, the last m quasi-velocities are set to zero. Therefore, the term u_7 will be equivalent to:

$$u_7 = \dot{S}_s = 0 \tag{3.64}$$

The resulting coefficients $\Psi_{ji} \, and \, \Psi_{jt}$ are :

$$\Psi_{ji}(q,t) = \begin{bmatrix} cos(q_6)cos(q_5) & -cos(q_5)sin(q_6) & sin(q_5) & 0 & 0 & 0 & 0 \\ c_1 & c_2 & -cos(q_5)sin(q_4) & 0 & 0 & 0 & 0 \\ c_3 & c_4 & -cos(q_4)cos(q_5) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -sin(q_5) & 0 \\ 0 & 0 & 0 & 0 & cos(q_4) & sin(q_4)cos(q_5) & 0 \\ 0 & 0 & 0 & 0 & -sin(q_4) & cos(q_4)cos(q_5) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.65}$$

$$c_1 = -cos(q_4)sin(q_6) + cos(q_6)sin(q_4)sin(q_5) \tag{3.66}$$

$$c_2 = -cos(q_4)cos(q_6) - sin(q_4)sin(q_6)sin(q_5) \tag{3.67}$$

$$c_3 = sin(q_4)sin(q_6) + cos(q_4)cos(q_6)sin(q_5) \tag{3.68}$$

$$c_4 = cos(q_6)sin(q_4) - cos(q_4)sin(q_6)sin(q_5) \tag{3.69}$$

$$\Psi_{j,t}(q,t) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.70}$$

3.5 Aerodynamics

The aerodynamic forces and torques of aircraft are generally expressed in the body reference frame located at the CV. These are shown for the longitudinal case in Figure 3.3. Considering that the dynamic model is established about the CG of the whole system, the aerodynamic forces and moments are given by [23, 69]. Moreover, the $\vec{A}$ matrix includes the produced moments caused by studying the vehicle around the CG.

$$\vec{A} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & d_{m,z} & 0 & 1 & 0 & 0 \\ d_{m,z} & 0 & -d_{m,x} & 0 & 1 & 0 \\ 0 & -d_{m,x} & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} A_{xb} \\ A_{yb} \\ A_{zb} \\ A_{\phi} \\ A_{\theta} \\ A_{\psi} \end{bmatrix} \tag{3.71}$$

where the aerodynamic forces about x, y, and z axis are A_{xb}, A_{yb}, and A_{zb}, respectively. The aerodynamic moments about x, y, and z axis are A_{ϕ}, A_{θ}, and A_{ψ}, respectively [23, 69].

$$A_{xb} = \hat{P}[C_{X1}cos^2(\alpha)cos^2(\beta) + C_{X2}sin(2\alpha)sin\left(\frac{\alpha}{2}\right)] \tag{3.72}$$

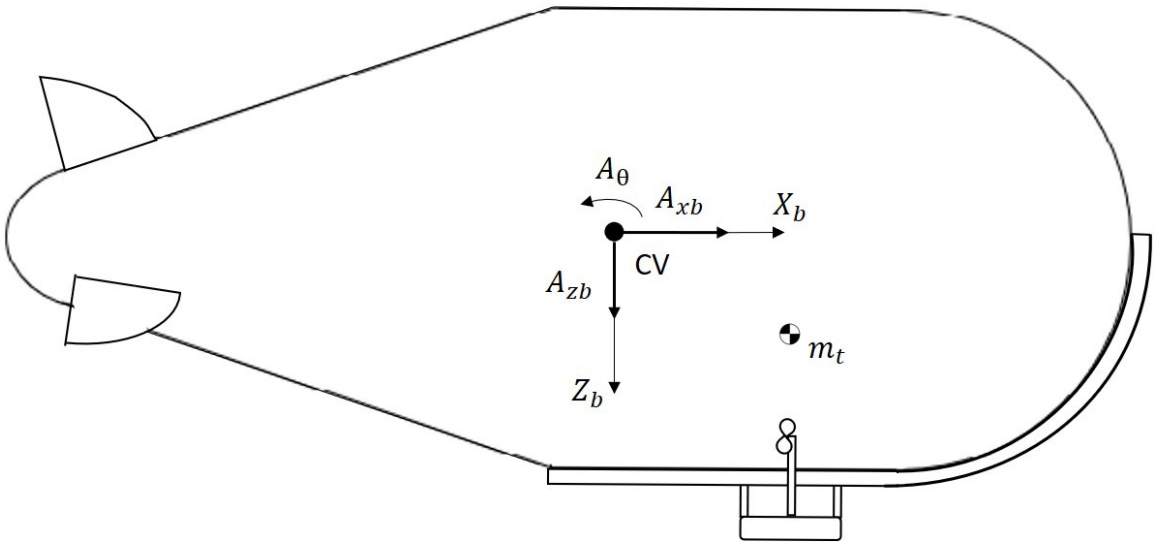


Figure 3.3: Longitudinal aerodynamic forces and moments for the airship

$$A_{yb} = \hat{P}[C_{Y1}cos\left(\frac{\beta}{2}\right)cos(2\beta) + C_{Y2}sin(2\beta) + C_{Y3}sin\beta sin|\beta|] \tag{3.73}$$

$$A_{zb} = \hat{P}[C_{Z1}cos\left(\frac{\alpha}{2}\right)cos(2\alpha) + C_{Z2}sin(2\alpha) + C_{Z3}sin\alpha sin|\alpha|] \tag{3.74}$$

$$A_{\phi} = \hat{P}[C_{L1}sin(\beta)sin|\beta|] + \frac{1}{2}\rho_a C_{L2}\dot{\phi}|\dot{\phi}| \tag{3.75}$$

$$A_{\theta} = \hat{P}[C_{M1}cos\left(\frac{\alpha}{2}\right)sin(2\alpha) + C_{M2}sin(2\alpha) + C_{M3}sin(\alpha)sin|\alpha|] + \frac{1}{2}\rho_a C_{M4}\dot{\theta}|\dot{\theta} \tag{3.76}$$

$$A_{\psi} = \hat{P}[C_{N1}cos\left(\frac{\beta}{2}\right)sin(2\beta) + C_{N2}sin(2\beta) + C_{N3}sin(\beta)sin|\beta|] + \frac{1}{2}\rho_a C_{N4}\dot{\psi}|\dot{\psi} \tag{3.77}$$

where the dynamic pressure is $\hat{P}$, the angle of attack is α and the side-slip angle is β [23, 69].

$$\hat{P} = \frac{1}{2}\rho_{air} v_0^2 \tag{3.78}$$

$$\alpha = arctan\left(\frac{\dot{z}}{\dot{x}}\right) \tag{3.79}$$

$$\beta = arcsin\left(\frac{\dot{y}}{\sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}}\right) \tag{3.80}$$

and the aerodynamic coefficients are defined as follows [23, 69]:

$$C_{X1} = -[C_{Dho} S_h \ + \ C_{Dfo} S_f \ + C_{Dgo} S_{ga}] \tag{3.81}$$

$$C_{X2} = C_{Y1} = C_{Z1} = (k_2 - k_1)\eta_k I_1 S_h \tag{3.82}$$

$$C_{Y2} = C_{Z2} = -\frac{1}{2}\left(\frac{\delta C_L}{\delta \alpha}\right)_f S_f \eta_f \tag{3.83}$$

$$C_{Y3} = -[C_{Dch} J_1 S_h \ + \ C_{Dcf} S_f \ + C_{Dcg} S_{ga}] \tag{3.84}$$

$$C_{Z3} = -[C_{Dch} J_1 S_h \ + \ C_{Dcf} S_f] \tag{3.85}$$

$$C_{L1} = C_{Dcg} S_{ga} d_{c,z} \tag{3.86}$$

$$C_{L2} = -2 C_{Dcf} S_f d_{f,z}^3 \tag{3.87}$$

$$C_{M1} = -C_{N1} = -(k_2 - k_1)\eta_k I_3 S_h l_{v,x} \tag{3.88}$$

$$C_{M2} = -C_{N2} = -\frac{1}{2}\left(\frac{\delta C_L}{\delta \alpha}\right)_F S_f \eta_f d_{f,x1} \tag{3.89}$$

$$C_{M3} = -C_{N3} = -[C_{Dch} J_2 S_h l_{v,x} + C_{Dcf} S_f d_{f,x2}] \tag{3.90}$$

$$C_{M4} = C_{N4} = -\left[C_{Dcf} S_f d_{f,x2}^3 + \frac{l_{v,z} l_{v,x}^4}{240}\right] \tag{3.91}$$

where C_{X1} and C_{X2} are the coefficients for the force of the x-axis, C_{Y1}, C_{Y2} and C_{Y3} are the coefficients for the force of the y-axis, and C_{Z1}, C_{Z2} and C_{Z3} are the coefficients for the force of the z-axis. Moreover, C_{L1} and C_{L2} are the coefficients for the moment of the x-axis, C_{M1}, C_{M2} and C_{M3} are the coefficients for the moment of the y-axis, and C_{N1}, C_{N2} and C_{N3} are the coefficients for the moment of the z-axis. Finally, the aerodynamic parameters are shown in Table 3.3.

Table 3.3: Aerodynamic parameters

Symbol	Parameter	Value	Units
C_{Dho}	Hull zero-incidence drag coefficient [67]	0.03	-
C_{Dfo}	Fin zero-incidence drag coefficient [70]	0.003	-
C_{Dgo}	Gondola zero-incidence drag coefficient [70]	1.2	-
C_{Dch}	Hull cross-flow drag coefficient [67]	1	-
C_{Dcf}	Fin cross-flow drag coefficient [71]	2	-
C_{Dcg}	Gondola cross-flow drag coefficient [72]	0.25	-
$\left(\frac{\partial C_L}{\partial \alpha}\right)_f$	Derivative of fin lift-coefficient with respect to the angle-of-attack at zero incidence [69]	5.73	-
$\left(\frac{\partial C_L}{\partial \delta}\right)_f$	Derivative of fin lift-coefficient with respect to the flap deflection angle [69]	0	-
S_h	Hull reference area, $V^{2/3}$ [SW [1]]	3.5114	m^2
S_f	Fin reference area [67]	0.5095	m^2
S_g	Gondola reference area [SW]	0.1159	m^2
l_{fx1}	x-distance from CV to aerodynamic center of fins [SW]	2.045	m
l_{fx2}	x-distance from CV to aerodynamic center of fins [SW]	2.045	m
l_{fx3}	z-distance from CV to aerodynamic center of fins [SW]	0.525	m
n_f	Fin efficiency factor accounting for the effect of the hull on the fins [67]	0.25	-
n_k	Hull effciency factor accounting for the effect of the fins on the hull [72]	1	-
I_1	Hull integrals [67]	0.1521	-
I_3		-0.2673	-
J_1		1.4164	-
J_2		0.5534	-

[1] SolidWorks

3.6 Thrust

Generally speaking, thruster forces are already expressed in terms of quasi-velocities. Thus, they do not need any transformation to be applied to the Boltzmann-Hamel equation.

This airship is equipped with two thrusters that are located on the gondola at z_t distance above the gondola CG. For the purpose of this thesis, it was assumed the same thrust, F_p, for both of them. So, the total applied thrust is $F_t = 2F_p$. Consequently, all the differential thrust moments are neglected (roll and yaw). The applied forces and torques about the global CG for the straight and curve section can be expressed as:

- Straight section

$$\vec{T} = F_t \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ z_g - d_{m,z} - z_t \\ 0 \\ 0 \end{bmatrix} \tag{3.92}$$

- Curve section

$$\vec{T} = F_t \begin{bmatrix} cos(\theta) \\ 0 \\ sin(\theta) \\ 0 \\ cos(\theta)(z_g - d_{m,z} - z_t) + sin(\theta)(x_g - d_{m,x}) \\ 0 \\ 0 \end{bmatrix} \tag{3.93}$$

3.7 Applying Boltzmann-Hamel

For this work, the interest is to control the pitch angle and the equations presented above were simplified to obtain only the longitudinal equations. It is assumed that the ϕ and ψ angles are equal to zero, since the model is constrained to the longitudinal plane. In addition, the gondola was assumed as a point mass, thus its rotational energy term is neglected. Once all of the previously presented terms are computed, the Bolztmann-Hamel equation (3.94), and its Hamel's coefficients (3.95) and (3.96) [32]. The Equations were implemented in Matlab.

$$\frac{d}{dt}\left(\frac{\partial L}{\partial u_r}\right) - \sum_{i=1}^{n} \frac{\partial L}{\partial q_i}\Phi_{ir} + \sum_{j=1}^{n}\sum_{l=1}^{n-m} \frac{\partial L}{\partial u_j}\gamma_{rl}^{j} u_l + \sum_{j=1}^{n} \frac{\partial L}{\partial u_j}\gamma_{r}^{j} = F_r \quad ; r = 1, ..., n-m \tag{3.94}$$

$$\gamma_{rl}^{j}(q,t) = \sum_{i=1}^{n}\sum_{k=1}^{n}\left(\frac{\partial \Psi_{ji}}{\partial q_k} - \frac{\partial \Psi_{jk}}{\partial q_i}\right)\Phi_{kl}\Phi_{ir} \tag{3.95}$$

$$\gamma_{r}^{j}(q,t) = \sum_{i=1}^{n}\sum_{k=1}^{n}\left(\frac{\partial \Psi_{ji}}{\partial q_k} - \frac{\partial \Psi_{jk}}{\partial q_i}\right)\Phi_{kt}\Phi_{ir} + \sum_{i=1}^{n}\left(\frac{\partial \Psi_{ji}}{\partial t} - \frac{\partial \Psi_{jt}}{\partial q_i}\right)\Phi_{ir} \tag{3.96}$$

Remark 1

The Bolztmann-Hamel equations were applied twice: when the gondola was on the straight section of the rail and when the gondola was on the curved section of the rail.

The final result equations are lengthy and are not presented. The procedure to obtain the dynamic equations are as follows:

1. Solve the kinematics to find $\vec{r_1}$ and $\vec{r_2}$.
2. Find the added masses, the inertias, as well as the d_{CG} matrix.
3. Compute the kinetic and potential energy.

4. Simplify the results to the longitudinal plane.

5. Find the Lagrangian and express it in quasi-velocities.

6. Calculate the partial derivatives $\frac{\partial L}{\partial q_i}$ and $\frac{\partial L}{\partial u_j}$, and substitute the previously defined constraint condition.

7. Apply the Boltzmann-Hamel equation, computing term by term and adding the results.

8. Add the thrust and aerodynamics forces and moments, $\vec{F} = \vec{T} + [\vec{A}^T\, 0]^T$ (Simplify to the longitudinal plane).

9. Solve for ui_1, ui_3, and ui_7.

Chapter 4

Pitch Control of a Reconfigurable Airship

4.1 Controller Requirements

Based on the current literature on airship control, the controller adopted should have the following qualities: be adequate for non-linear systems, be robust, and be simple for experimental implementation.

The adaptive self-tuning PID controller [73] was found to satisfy all of the above requirements. This nonlinear technique has adaptation capabilities that increase the method performance under variations of the plant and/or flight conditions. The consideration of a sliding condition for the adaptation law adds robustness to the system. In addition, it possesses the simplicity of the PID compensator for experimental implementation. Moreover, the inclusion of a supervisory controller guarantees closed-loop stability and helps the adaptation control during its convergence process.

The control scheme is presented in Figure 4.1, where external disturbances could refer to different factors, such as wind, external noise, and uncertainties:

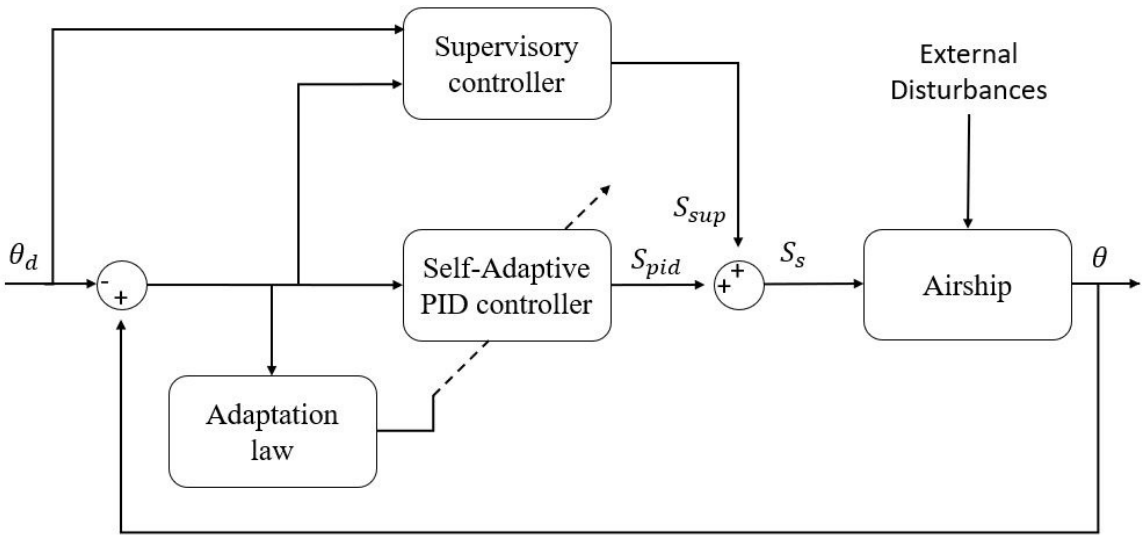


Figure 4.1: Adaptive robust self-tuning PID scheme for pitch control of an airship with sliding gondola

Section 4.2 describes in detail the design of the controller to control the pitch angle.

4.2 Design of an Adaptive Self-Tuning PID Controller for Pitch Control

The controller's design is based on the control technique shown in [73]. The technique is used for unknown and uncertain model systems. Adapting the technique to this case, one can express the pitching dynamics as a second-order system:

$$\dot{x}_1(t) = \dot{\theta}(t)$$

$$\dot{x}_2(t) = \ddot{\theta} = f(X,t) + \Delta f(X,t) + \delta(t) + u(t) \tag{4.1}$$

$$y_{out}(t) = x_1(t) = \theta$$

where $X = [x_1, x_2]^T = [\theta, \dot{\theta}]^T$ are the measurable states, f is the nonlinear nominal

function of the system, δ is the external noise, and Δf is the plant uncertainty. The system output is y_{out}, or in this case is θ. The system input is u, or in this case it is the traveling distance by the gondola, S_s.

The error e can be computed as the difference between the measure output θ (or x_1) and the desired signal θ_d. Therefore, it is defined as:

$$e = \theta - \theta_d \tag{4.2}$$

From (4.3), the error vector can be expressed as $E = [e, \dot{e}]^T$. Therefore, defining a feedback linearization controller to the system shown in Figure 4.1 results in:

$$S_s^* = -f(X,t) - \Delta f(X,t) - \delta(t) + \ddot{\theta}_d + K^T E \tag{4.3}$$

Replacing (4.3) in (4.1) results in:

$$\ddot{e} + k_1\dot{e} + k_0 e = 0 \tag{4.4}$$

From (4.4), one can see that $e \to 0$ as $t \to 0$, when there is tracking e.g. $\theta \to \theta_d$.

The control input is defined by two signals. The first one is from the adaptive PID controller and the second one is the supervisory controller.

$$S_s = S_{sup} + S_{pid} \tag{4.5}$$

- Adaptive PID based on sliding surface

  The output of the PID controller is defined as:

$$S_{pid} = K_p(e(t)) + K_i \int_0^t (e(t))dt + K_d \frac{d}{dt} e(t) \tag{4.6}$$

The PID gains are updated based on a sliding surface. Therefore, the following sliding surface S was defined:

$$S = \dot{\theta} - x_r \tag{4.7}$$

where x_r is a designed signal that satisfies the following relation:

$$\dot{x}_r = \ddot{\theta}_d + k_1\dot{e} + k_0 e \tag{4.8}$$

when the sliding mode occurs $S = 0$. Therefore:

$$\dot{\theta} = x_r \tag{4.9}$$

If (4.9) is substituted into (4.8), the result is (4.4), showing that the error will go toward zero as time tends to infinity.

For the first step, the following Lyapunov function candidate was chosen:

$$V_1 = \frac{1}{2}S^2 \tag{4.10}$$

Taking the derivative of V_1, and based on the fact that when the sliding condition is satisfied $S(t) \to 0$ as $t \to \infty$, it results in:

$$\dot{V}_1 = S\dot{S} < 0 \tag{4.11}$$

Substituting (4.1) and (4.5) into (4.11)

$$S\dot{S} = S[f(\cdot) + \Delta f(\cdot) + \delta(\cdot) + S_{pid} + S_{sup} - \dot{x}_r] \tag{4.12}$$

Minimizing (4.12) as the error function, the adaptation laws for K_p, K_i and K_d can be obtained using the gradient descent method. So, combining equations (4.6) and (4.12), the gains can be computed as:

$$\dot{K}_p = -\gamma_1 \frac{\partial S \dot{S}}{\partial K_p} = -\gamma_1 \frac{\partial S \dot{S}}{\partial S_{pid}} \frac{\partial S_{pid}}{\partial K_p} = -\gamma_1 S e(t) \tag{4.13}$$

$$\dot{K}_i = -\gamma_2 \frac{\partial S \dot{S}}{\partial K_i} = -\gamma_2 \frac{\partial S \dot{S}}{\partial S_{pid}} \frac{\partial S_{pid}}{\partial K_i} = -\gamma_2 S \int_0^t e(t) dt \tag{4.14}$$

$$\dot{K}_d = -\gamma_3 \frac{\partial S \dot{S}}{\partial K_d} = -\gamma_3 \frac{\partial S \dot{S}}{\partial S_{pid}} \frac{\partial S_{pid}}{\partial K_d} = -\gamma_3 S \frac{de(t)}{dt} \tag{4.15}$$

Now the error dynamics can be defined as:

$$\dot{E} = A_m E + B[S_s^* - S_{pid} - S_{sup}] \tag{4.16}$$

where:

$$A_m = \begin{bmatrix} 0 & 1 \\ -k_0 & -k_1 \end{bmatrix} \tag{4.17}$$

$$B_m = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{4.18}$$

Note that the eigenvalues in the A matrix must be positive to guarantee stability. In order to demonstrate closed-loop stability, the Lyapunov theory was applied. In the first step, the following Lyapunov function candidate was chosen:

$$V_2 = \frac{1}{2} E^T P E \tag{4.19}$$

where P is a positive definite symmetric matrix that satisfies the Lyapunov equation:

$$A^T P + PA = -Q \tag{4.20}$$

Taking the derivative of V_2, it results in:

$$\begin{aligned}
\dot{V_2} &= \frac{1}{2}[\dot{E}^T PE + E^T P\dot{E}] \\
&= \frac{1}{2}[(E^T A^T + B^T(S_s^* - S_{pid} - S_{sup}))PE + E^T P(AE + B(S_s^* - S_{pid} - S_{sup})] \\
&= \frac{1}{2}E^T(A^T P + PA)E + E^T PB[S_s^* - S_{pid} - S_{sup}] \\
&= -\frac{1}{2}E^T(Q)E + E^T PB[S_s^* - S_{pid} - S_{sup}] \\
&= -\frac{1}{2}E^T(Q)E + E^T PB[S_s^* - S_{pid}] - E^T PBS_{sup} \\
&\leq -\frac{1}{2}E^T(Q)E + \left|E^T PB\right|[|S_s^*| + |S_{pid}|] - E^T PBS_{sup}
\end{aligned} \tag{4.21}$$

Therefore, the supervisory controller signal can be selected as:

$$S_{sup} = sgn(E^T PB)[|g_{up}(.)| + \Delta g_{up}(\cdot) + \kappa + \left|\ddot{\theta}_d\right| + \left|K^T E\right| + |S_{pid}|] \tag{4.22}$$

where $g_{up}(.)$ is a positive upper bond on $f(X,t) \Rightarrow g_{up}(.) \geq |f(X,t)|$. Δg_{up} is a positive upper bond on $\Delta f(X,t) \Rightarrow \Delta g_{up}(\cdot) \geq |\Delta f(X,t)|$. κ is a positive upper bond on $\delta(t) \Rightarrow \kappa \geq |\delta(t)|$. Now, substituting the upper bounds and (4.22) in (4.23):

$$\dot{V}_2 \leq -\frac{1}{2}E^T(Q)E + \left|E^T PB\right|[|S_s^*| + |S_{pid}|] - E^T PBS_{sup}$$

$$\leq -\frac{1}{2}E^T(Q)E + \left|E^T PB\right|(\left|-f(X,t) - \Delta f(X,t) - \delta(t) + \ddot{\theta}_d + K^T E\right| + |S_{pid}|)$$

$$-sgn(E^T PB)E^T PB(|g_{up}(.)| + \Delta g_{up}(\cdot) + \kappa + \left|\ddot{\theta}_d\right| + \left|K^T E\right| + |S_{pid}|)$$

$$< 0 \tag{4.23}$$

Note that $\dot{V}_2 < 0$, when $E^T PB > 0 \Rightarrow sgn(E^T PB) = 1$, and for $E^T PB < 0 \Rightarrow sgn(E^T PB) = -1$. Therefore, the closed-loop stability of the system is guaranteed.

- Chattering phenomenon

  In preliminary experiments, the control law $sgn(E^T PB)$ proposed in [73] was found to induce high frequency switching of the actuators. Therefore, to improve the control law, the signum function was replaced by a saturation function. This function is defined as:

$$sat\left(\frac{E^T PB}{\varepsilon_1}\right) = \begin{cases} \frac{E^T PB}{\varepsilon_1} \geq 1 \;\; ; \;\; 1 \\ -1 < \frac{E^T PB}{\varepsilon_1} < 1 \;\; ; \;\; \frac{E^T PB}{\varepsilon_1} \\ \frac{E^T PB}{\varepsilon_1} \leq -1; \;\; -1 \end{cases} \tag{4.24}$$

  where ε_1 represents the width of the boundary layer.

- Supervisory controller disconnection

  The supervisory controller produces a boost to the control signal, pushing the states back to the region where the closed-loop stability is guaranteed. This is useful in case the adaptive control is unable to control the system. However, to achieve this, the supervisory controller must apply high control signals to the system and this can produce undesired responses. Therefore, it is not always

necessary to consider its action in the controller output. A disconnection condition was proposed to turn on/off the supervisory action from the controller output. The disconnection condition is expressed as follows:

$$f_s = \begin{cases} |e| > \varepsilon_2 \;\; ; \;\; 1 \\ \\ |e| < \varepsilon_2 \;\; ; \;\; 0 \end{cases} \tag{4.25}$$

Consequently, (4.5) can be rewritten as:

$$S_s = S_{sup} f_s + S_{pid} \tag{4.26}$$

This means that when the system is below the condition ε_2, the controller output relies solely on the adaptive-PID controller.

4.3 Simulation

In order to test the performance of the proposed control techniques, numerous simulations were carried out in Simulink. Two different reference signals were tested for the simulations: sinusoidal and pre-defined trajectory. The sinusoidal signal is ideal to adequately excite the system, helping the controller to adapt. The pre-defined trajectory was chosen to evaluate the response performance under a descending scenario. The simulations were run using the Dryden wind model as the source of external disturbances. This turbulence model is one of the most used for flight mechanics applications, since it is doable to implement in simulations [74]. For this, the model applies perturbations to the velocity and angle rate body axes of the vehicle, in the form of white noise [74].

- Sinusoidal reference

  Figures 4.2, 4.3, and 4.4 show the results for the following learning rates $\gamma_1 = 5$, $\gamma_2 = 0.1$ and $\gamma_3 = 0.1$ for K_p, K_i, and K_d, respectively. The reference signal for this case was a sinusoidal signal of 20 degrees amplitude and period of 500 s.

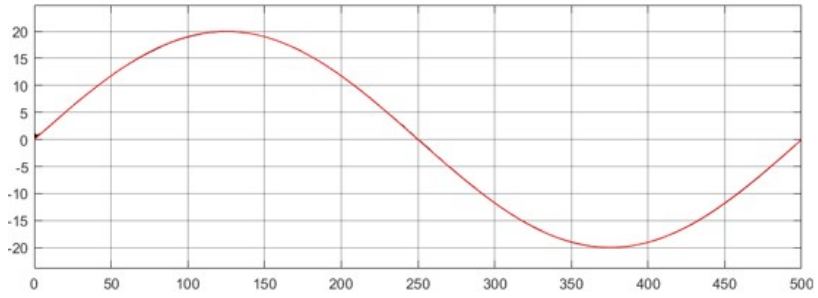


Figure 4.2: Pitch angle response for a sinusoidal reference signal with $\gamma_1 = 5$, $\gamma_2 = 0.1$ and $\gamma_3 = 0.1$

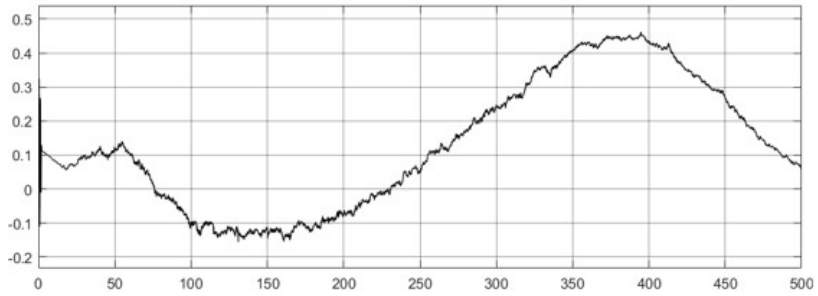


Figure 4.3: Gondola position [m] for a sinusoidal reference signal with $\gamma_1 = 5$, $\gamma_2 = 0.1$ and $\gamma_3 = 0.1$

Figure 4.2 shows that the system is able to adequately track the desired reference signal. Moreover, it can be observed a good controller performance under external disturbances. Figure 4.3 presents the gondola position required to track the reference signal.

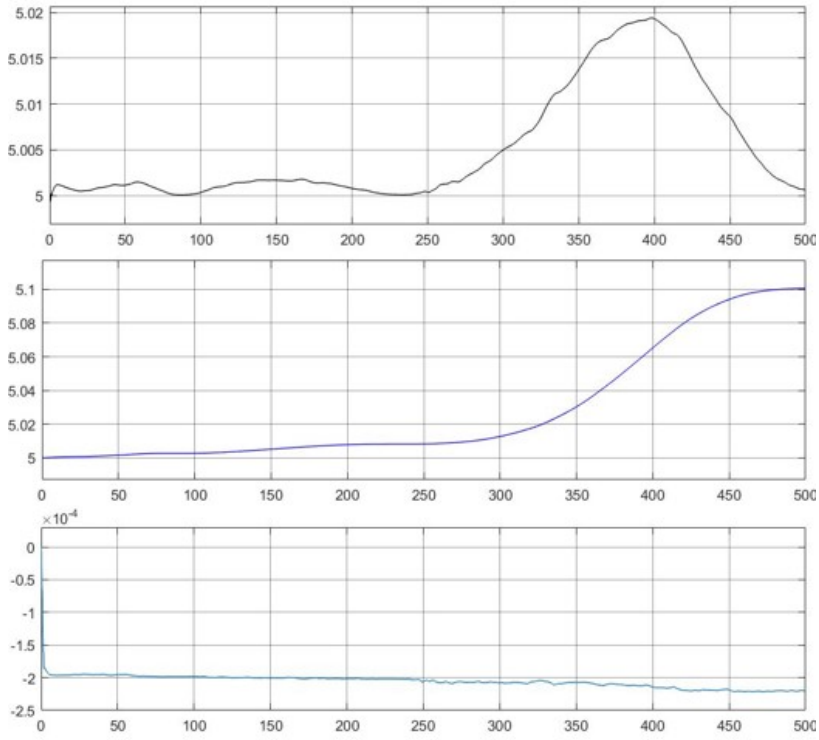


Figure 4.4: PID gains for a sinusoidal reference signal with $\gamma_1 = 5$, $\gamma_2 = 0.1$ and $\gamma_3 = 0.1$

Figure 4.4 presents how the PID gains adapt and converge along the simulation, for positive and negative pitch angles. This demonstrates that the adaptation algorithm is working properly.

- Trajectory reference

  Figures 4.5, 4.6, and 4.7 show the results for the following learning rates $\gamma_1 = 5$, $\gamma_2 = 0.1$ and $\gamma_3 = 0.1$ for K_p, K_i, and K_d, respectively. The reference signal for this case was a trajectory that started at 0 degrees and went to -20 degrees in 100 s. It held the position for 70 seconds and then went back to zero, with the same slope.

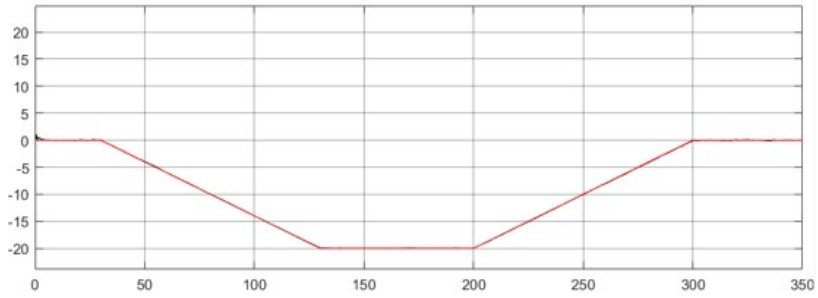


Figure 4.5: Pitch angle response for a trajectory reference signal with $\gamma_1 = 5$, $\gamma_2 = 0.1$ and $\gamma_3 = 0.1$

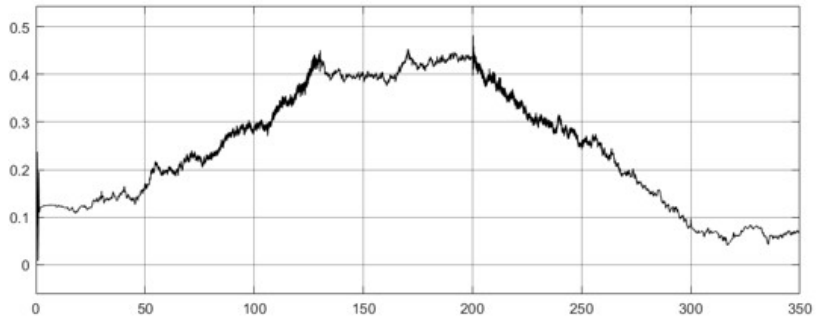


Figure 4.6: Gondola position [m] for a trajectory reference signal with $\gamma_1 = 5$, $\gamma_2 = 0.1$ and $\gamma_3 = 0.1$

Figure 4.5 shows that the system is able to adequately track the desired reference signal that recreates a landing manoeuvre. Moreover, it can be observed again the controller robustness despite the under external disturbances.

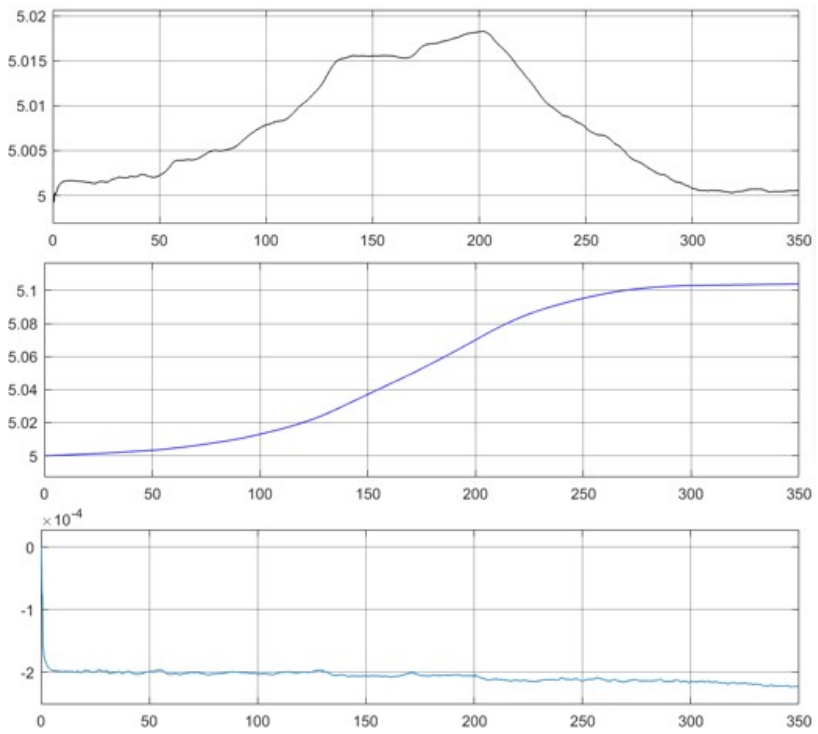


Figure 4.7: PID gains response for a trajectory reference signal with $\gamma_1 = 5$, $\gamma_2 = 0.1$ and $\gamma_3 = 0.1$

Figure 4.7 presents how the PID gains adapt and converge along the simulation, for the descending trajectory. This demonstrates that the adaptation algorithm is working properly.

Finally, it can be observed from both cases that the controller gain D is close to zero. This could be caused by the derivative term acting based on abrupt changes in the desired trajectory, or, this case, the reference signals being soft. Moreover, this value adapts based on the error derivative, hence when the desired tracking reaches $\dot{e} \to 0$. Therefore, for these scenarios and application, the derivative component of the controller is not significant.

Chapter 5

Prototype Implementation

5.1 Prototype Description

The final prototype is shown in Figure 5.1. The gondola is made of carbon fiber, and it is illustrated in Figure 5.2. The gondola has a total length of 1 m and a total width of 1.95 m. It has two arms with brushless motor propellers mounted on their tip. The thrusters can tilt thanks to two servos. In addition, attached to it are two sliding mechanisms that allow the gondola to move along the rail, as shown in Figure 5.3. The drive mechanism is powered by two DC motors that pull the gondola over the rail. There is enough friction in the gearmotors that when unpowered they can hold the position of the gondola.

Figure 5.1: Airship prototype

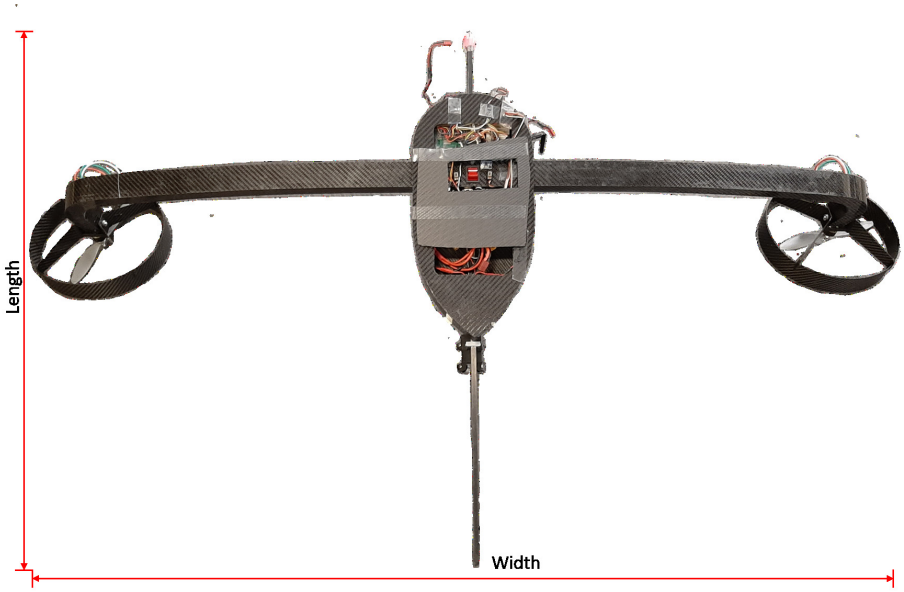


Figure 5.2: Gondola

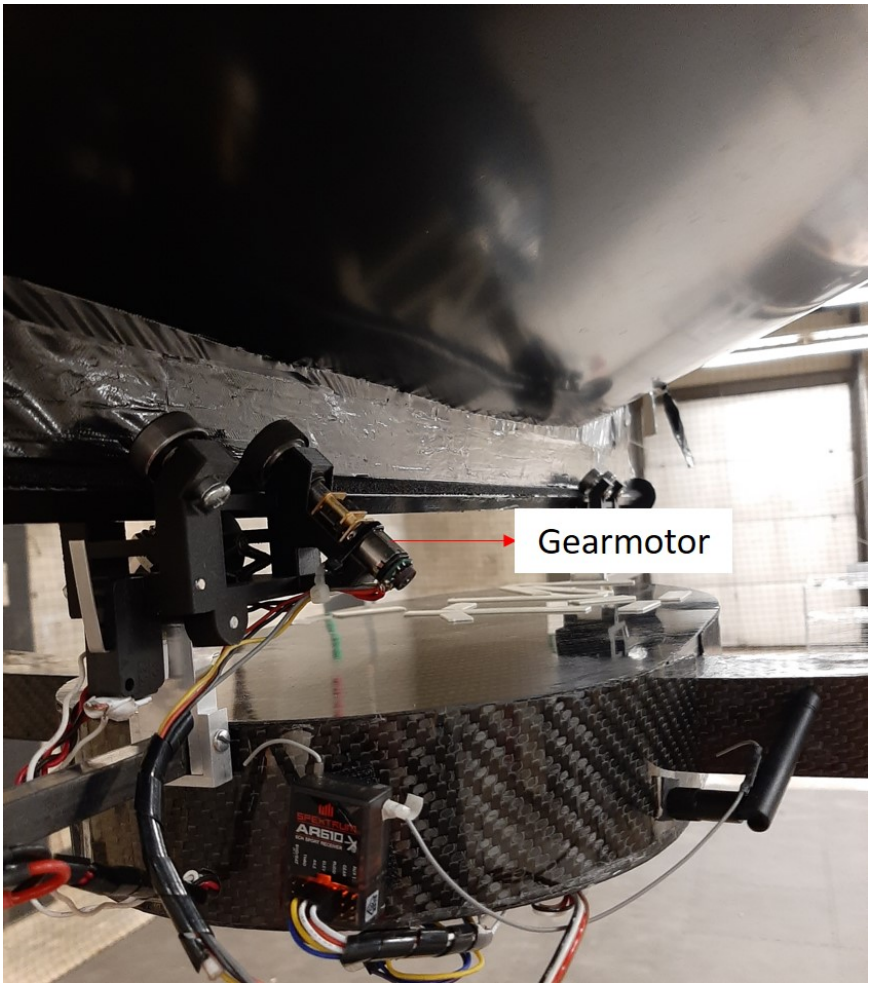


Figure 5.3: Sliding mechanism

5.2 Electrical Components

The airship is equipped with a control board that takes the readings from the sensors and sends the control signals to the actuators. For this case, the main board is a Raspberry Pi. The prototype has two sensors onboard: an IMU and an encoder. The IMU communicates with the board via I^2C, while the encoder is magnetic and coupled to one of the gondola motors. A counter chip module was added to read the encoder pulses and relay the speed to the RPI using the SPI protocol. As for the Raspberry Pi, it sends the PWM control signals to the actuators through the motor

controller, electronic speed controllers (ESCs), or directly to the actuators in the case of the servos.

The system is powered by two batteries. One supplies the control components and the other one supplies the actuators. In addition, there is a board for wireless communication with a ground station. Figure 5.4 shows the block diagram of the electrical components. Table 5.1 presents the reference for each on-board component.

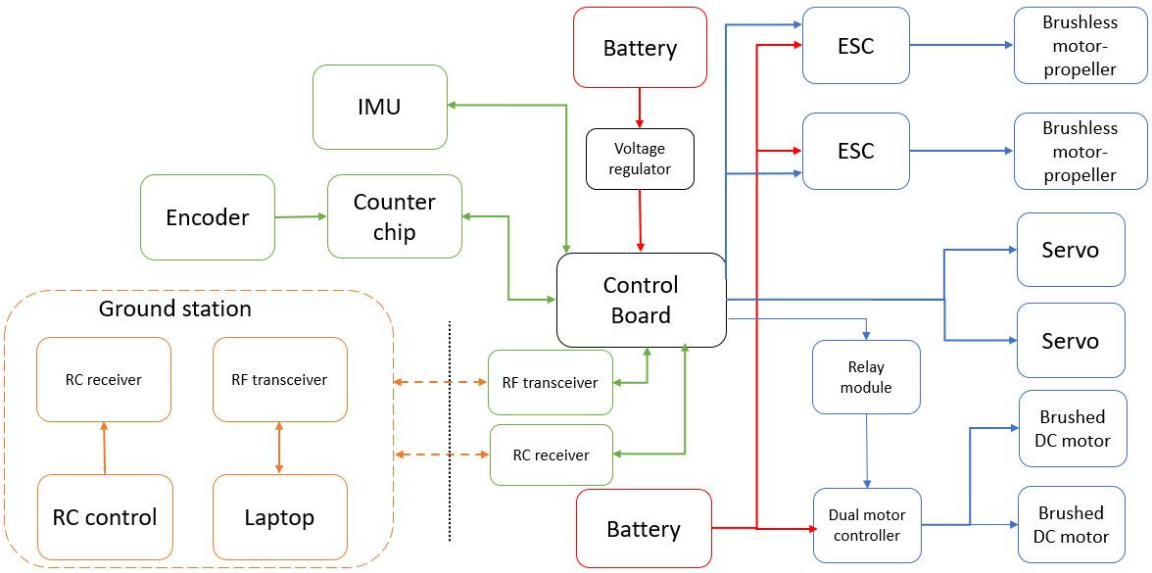


Figure 5.4: Block diagram of the electrical components

Table 5.1: Electrical components

Component	Description
Battery	Lipo 11.1 V @ 8000mAh
Brushless motor	Rimfire 400 20-30-950
Brushed DC motor	Pololu gearmotor @ 1000:1
Control board	Raspberry Pi 3b+
Counter chip	Counter click with LS7366R
Dual motor controller	Pololu TReX Jr
Encoder	Magnetic encoder
Electronic speed controller (ESC)	Brushless car ESC 45A w/ Reverse
IMU	MPU6050 - 6 DOF
RF transceiver	Module RF 7020 v4.0
Servo	HS-77BB
Radio control	Spektrum DXe transmitter
Radio receiver	AR610 receiver
Relay module	Arduino compatible 5V relay module

Figure 5.5 shows the components placed inside the gondola. The system has a main switch to turn on the board. Moreover, there are three manual switches that power every group of actuators.

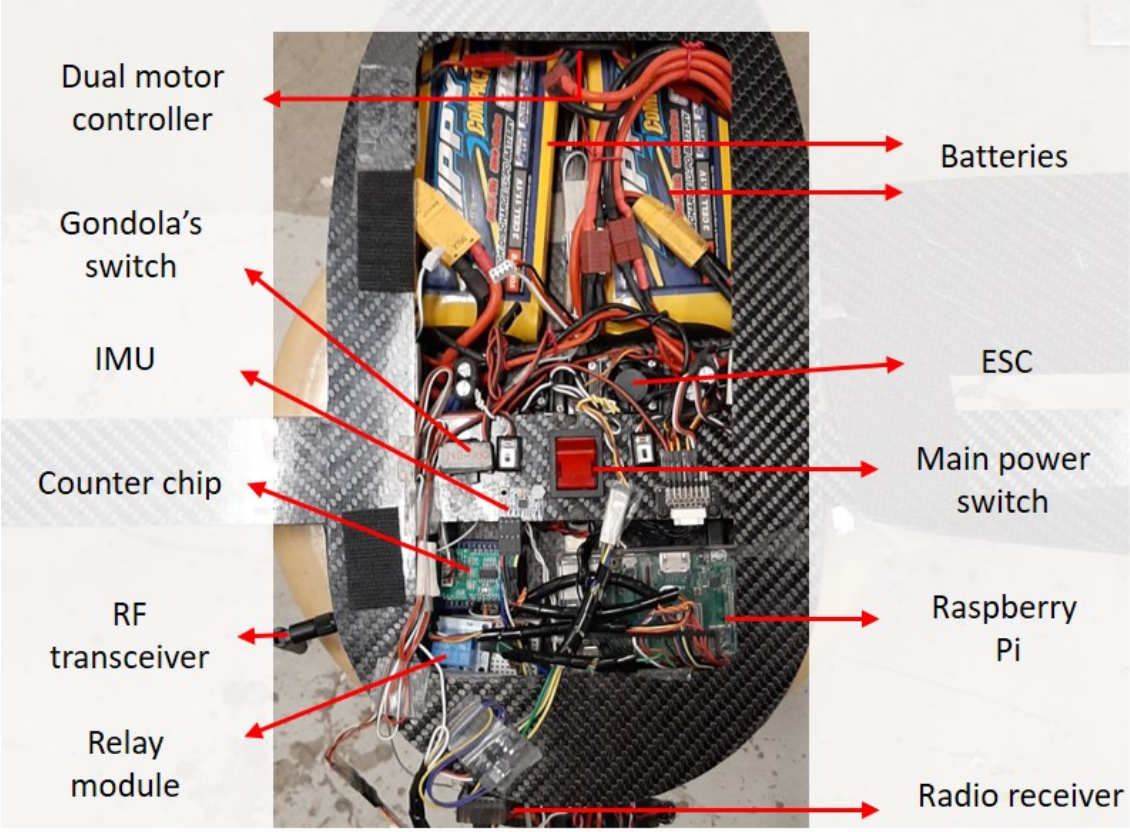


Figure 5.5: Gondola connection

5.3 Ground Control Station

The vehicle is equipped with a wireless communication board. The module allows for communication up to a distance of 3000 m. A laptop running a Simulink model acts as a ground control station. A transceiver module is connected via USB to the laptop.

The ground station user interface is divided into three sections. The first one is shown in Figure 5.6. The first module, shown in Figure 5.6, is used to arm and disarm the thrusters and the gondola motors. This button has to be activated before the system starts running a flight. In addition, this button also acts as a safety remote interrupter because it can be used at any time to immediately stop the actuators in

case of undesired or unsafe manoeuvres.

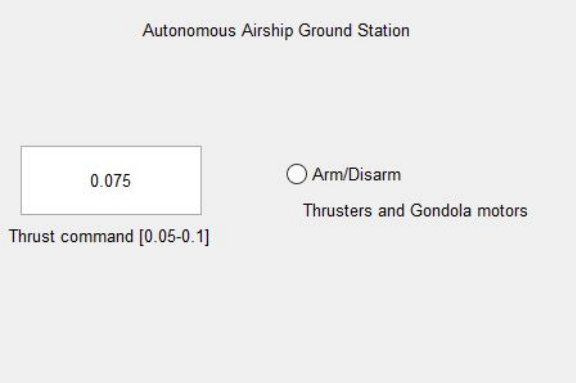


Figure 5.6: Enable interface

The second module controls the data transfer to the Pi on the gondola. The block schematic is shown in Figure 5.7. The proper and available serial has to be selected (i.e., COM4). The sent package contains four signals, one each for: the enable, the thrusters, the servos, and the reference signal for the pitch angle (θ_d).

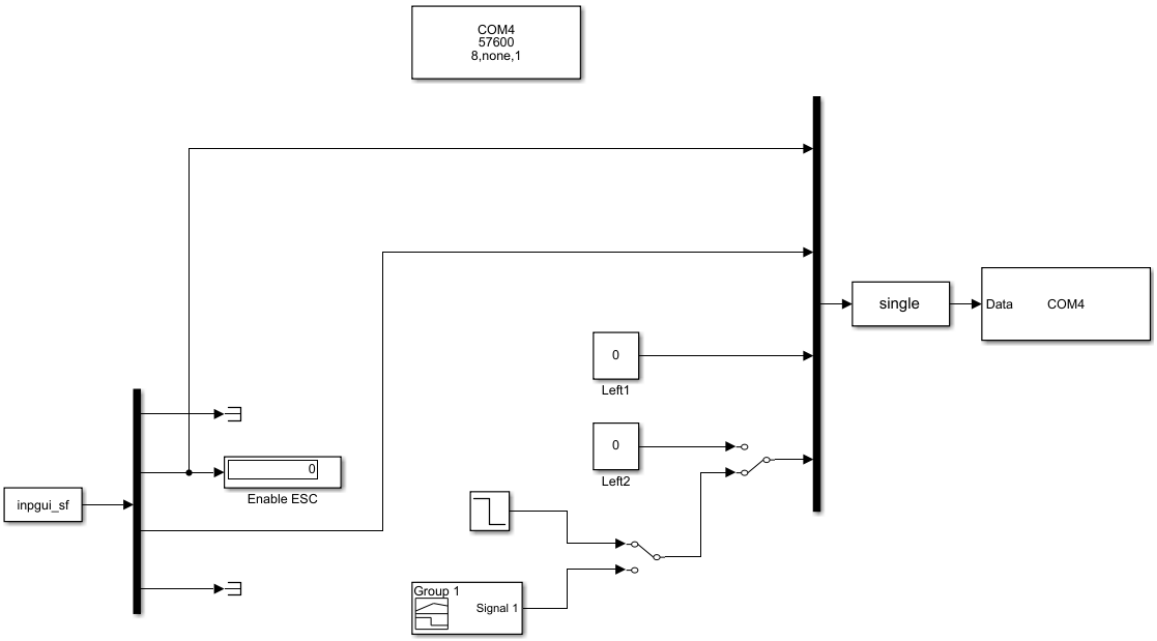


Figure 5.7: Simulink diagram of sending data from the ground station

The third module receives the data from the Raspberry Pi. The block schematic

is shown in Figure 5.8. The received package contains five signals, one each for: the pitch reference signal (θ_d), the current pitch angle (θ), the desired gondola travelled distance (S_d), the current gondola travelled distance (S_s), and the gondola motor signal.

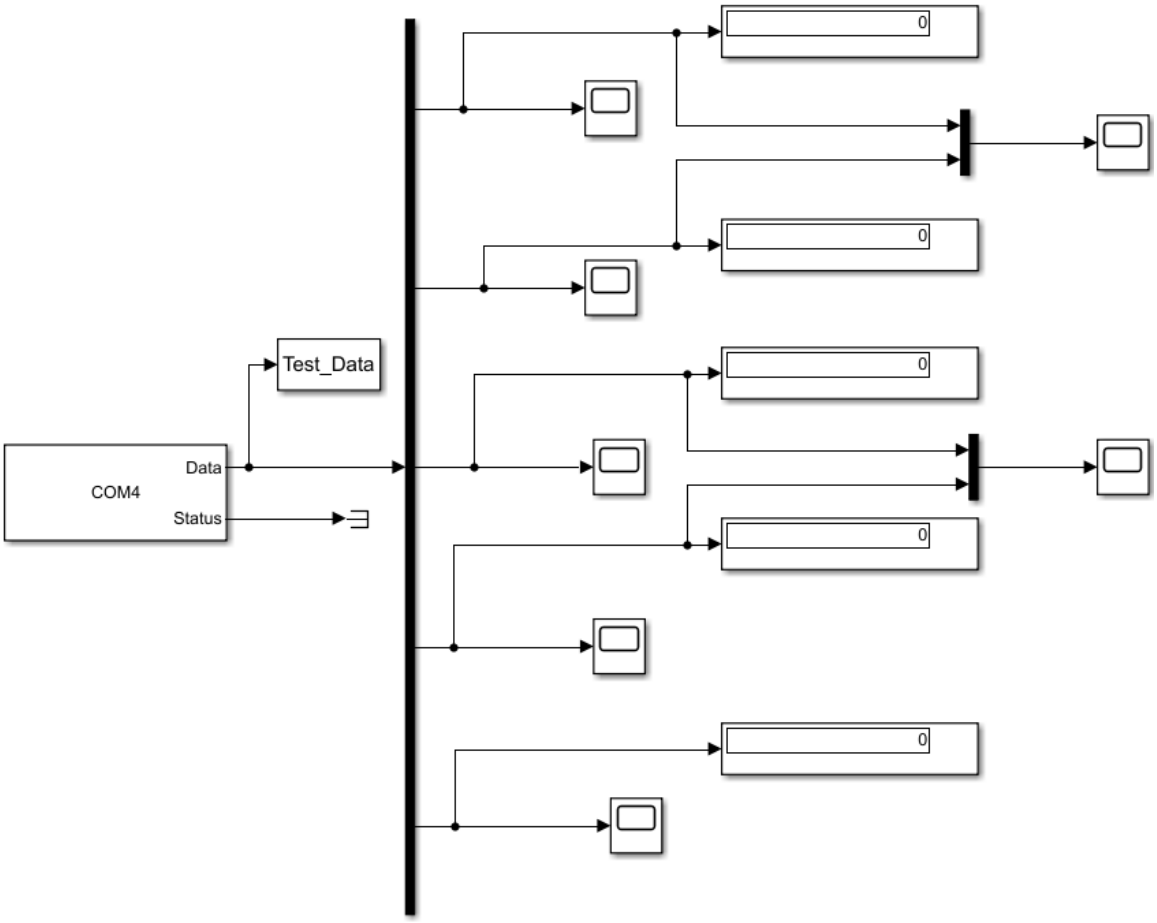


Figure 5.8: Simulink diagram of receiving data from the ground station

5.4 Orientation Estimation

The pitch angle was estimated using the readings from the IMU sensor. This inclination can be computed using only the accelerometer readings as follows, where the accelerometer readings along the x, y, and z axes are x_{acc}, y_{acc}, and z_{acc}, respectively:

$$\theta_{acc} = arctan\left(\frac{x_{acc}}{\sqrt{y_{acc}^2 + z_{acc}^2}}\right) \tag{5.1}$$

The pitch angle can also be estimated using only the gyroscope readings, as follows:

$$\theta_{gyr_{k+1}} = \theta_{gyr_k} + \dot{\theta}_{gyr_k}\, T_s \tag{5.2}$$

where the angular acceleration $\dot{\theta}_{gyr}$ is provided by the sensor and the sample time is T_s.

Although the angle θ_{acc} estimated by the accelerometer is noisy, it is reliable in the long term. In contrast, the angle θ_{gyr} is reliable in the short term, but it tends to drift as it is obtained by integration. Therefore, one of the most used approaches, thanks to its simplicity and reliability, is called the complementary filter [75]. This filter combines the advantages of both calculations, thus providing a better estimation than only using the accelerometer or the gyroscope readings.

The complementary filter applies a low-pass filter to the accelerometer reading in order to remove the undesired noise in the short term. Similarly, it applies a high-pass filter to the gyroscope reading in order to remove the drift in the long term. Hence, the filter can be implemented as follows [75]:

$$\theta = \alpha_f(\theta + \theta_{gyr}) + (1 - \alpha_f)\theta_{acc} \tag{5.3}$$

where the filter parameter is α_f and it has a value between 0 and 1. At the higher values of α_f, it relies more on the gyroscope readings. After calibrating the IMU to remove the bias and running a number of tests, α was set as 0.99.

5.5 Implemented Control Scheme

For the flight test, the control scheme shown in Figure 5.9 was implemented in Raspberry Pi.

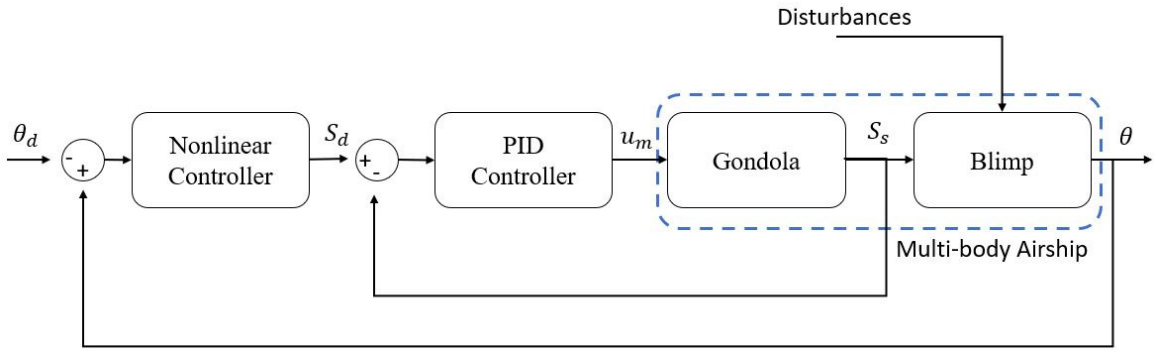


Figure 5.9: Implemented control scheme

The reference signal is the desired pitch angle (θ_d). This signal is the input of the nonlinear control technique that was described in Chapter 4. The output of this controller is the required gondola position S_d to achieve θ_d. The desired distance S_d is the input of a PID controller. The control action of this strategy is the motor signal that allows the gondola to slide over the keel. The PID controller was tuned by trial and error, with no overshoot. The inner loop was tuned as a first step, and its gains were kept during all of the tests.

Chapter 6

Experimental Results and Discussion

This chapter presents the experimental results of the adaptive self-tuning PID method for the control of the pitch angle for the airship previously described. The tests were carried out in the LTA Flight Lab at uOttawa. The experiments were performed in two areas of the lab. The first is a meshed, closed-in zone, with a width of 6 m, a height of 3.5 m, and a length of 12 m. The second area, located beside the first, was used for testing higher pitch angles.

6.1 Controller Initialization

For each test, the controller was uploaded to the Raspberry Pi, then the thruster and gearmotor switches were activated. Since the controller was continually sending and computing the commands, there was a 10-20 s initialization period in which the controller needed to readjust its output as, usually, the actuators start with the maximum signal.

6.2 Tests methodology

For the flight tests, two main correlations and effects were studied. First, the effect in the system response of the deactivation of the supervisory controller. This study is presented in Section 6.3. Secondly, the correlation between the learning rates and the system response was analyzed in Section 6.4. The goal of this set of tests was to tune the controller properly in order to eliminate the steady state error for a trajectory. It was considered suitable for the adaptation process of the controller gains the usage of ramps in the reference trajectory. Additionally, it has been reported that abrupt changes in the trajectories could produce undesired overshoot [39]. The criterion to evaluate the performance of every set of learning rates was the error. Finally, the best arrange of γ_1, γ_2, and γ_3 was chosen and it was used to assess the system response under different scenarios. The results for the final rates are shown in Section 6.5.

6.3 Supervisory Controller Effect

In this section, the supervisory controller action was studied. The supervisory controller produces high control action that creates oscillations in the system. Therefore, this signal should be disconnected when it is not needed.

Figures 6.1, 6.2, 6.3, and 6.4 show the results when the supervisory controller was not deactivated. The learning rates used were $\gamma_1 = 4$, $\gamma_2 = 0.3$ and $\gamma_3 = 0.1$ for K_p, K_i, and K_d, respectively. The step function with an initial value of -5 degrees and a final value of -15 degrees was chosen as a reference signal for this case.

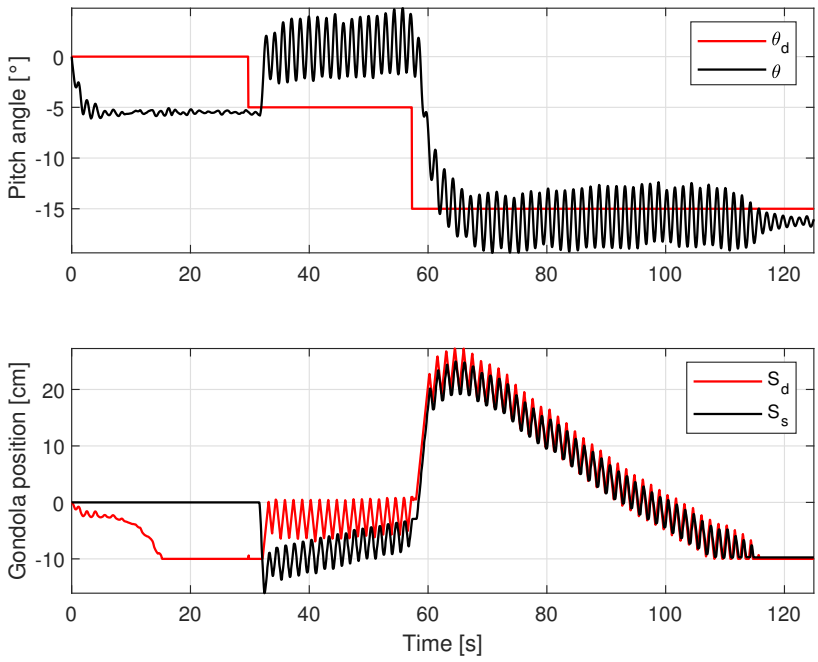


Figure 6.1: Pitch angle and gondola travelled distance control without supervisory controller disconnection for $\gamma_1 = 4$, $\gamma_2 = 0.3$ and $\gamma_3 = 0.1$

Figure 6.1 shows that the controller started computing its output, when the Simulink model was uploaded. However, the reference signal was received a couple of seconds later, when the ground station was run. This behaviour can be observed in the first 25 s of all of the figures. Figure 6.1 also shows the oscillations produced by the action of the supervisory controller during the experiment.

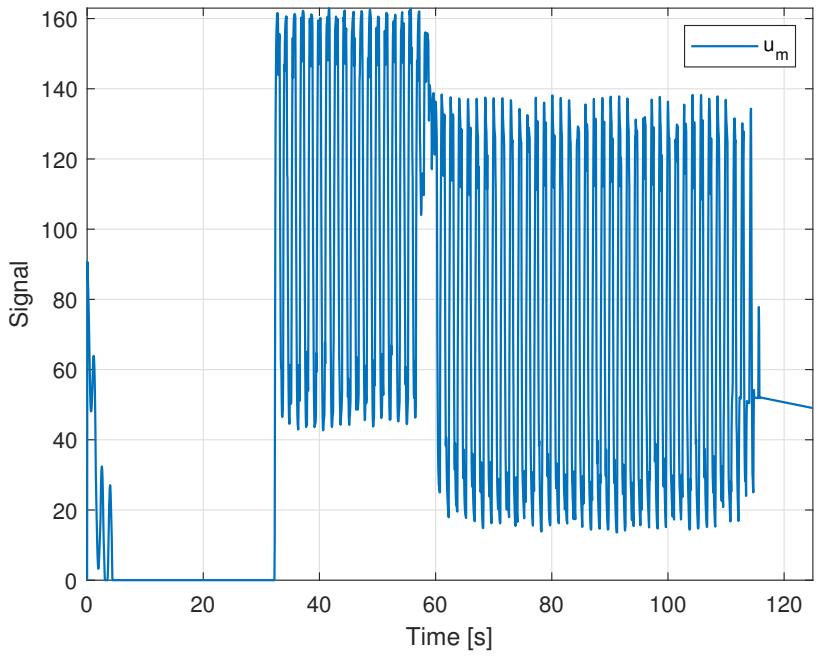


Figure 6.2: Gondola motor signal without supervisory controller disconnection for $\gamma_1 = 4$, $\gamma_2 = 0.3$ and $\gamma_3 = 0.1$

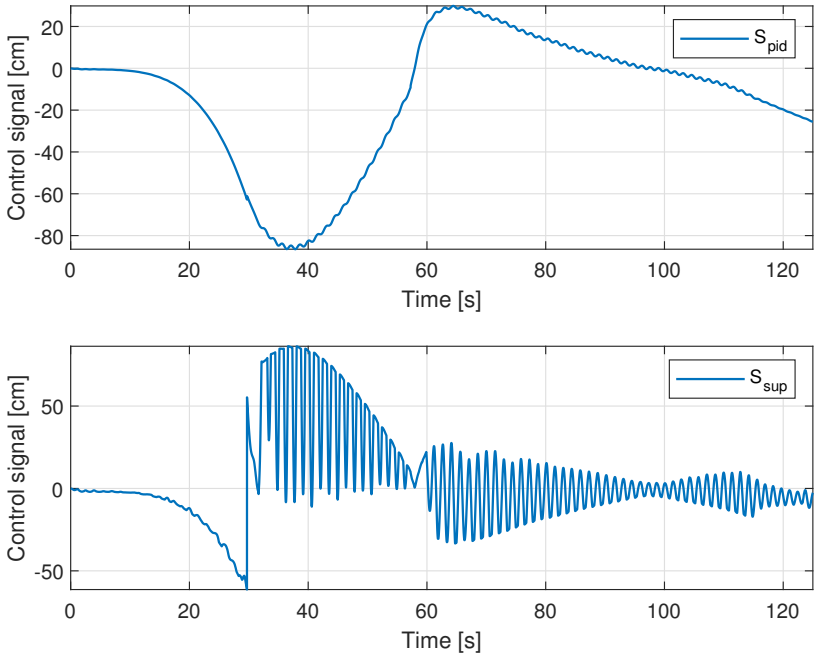


Figure 6.3: Control action of adaptive PID and supervisory controller without supervisory controller disconnection for $\gamma_1 = 4$, $\gamma_2 = 0.3$ and $\gamma_3 = 0.1$

In addition, Figure 6.2 and Figure 6.3 exhibit the chattering phenomenon in the actuator generated by the continuous action of the supervisory controller. Figure 6.3 shows how, despite the fact that the supervisory controller adds oscillations to the response, it helps the adaptive PID readjust its output signal more rapidly.

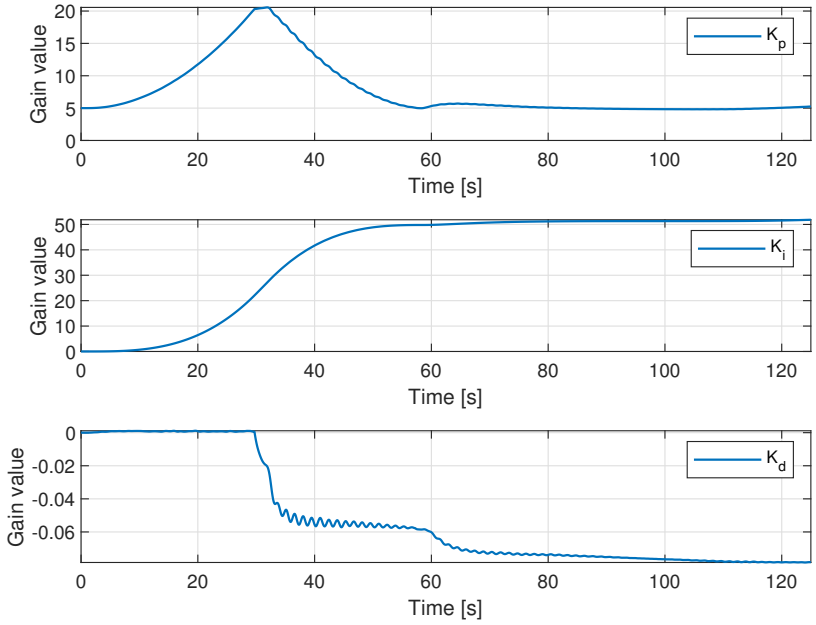


Figure 6.4: Adaptive PID gains without supervisory controller disconnection for $\gamma_1 = 4$, $\gamma_2 = 0.3$ and $\gamma_3 = 0.1$

Finally, it is possible to see, in Figure 6.4 how the PID gains converge to specific values during this test. The gains K_p and K_i converge 22 s after the - 15 degrees step reference was received.

The behaviour described above is undesired. Therefore, for the next experiments the supervisory controller was activated, based on the condition that was established in Section 4.2. Figures 6.5, 6.6, 6.7, and 6.8 show the results when the supervisory controller is disconnected, as was established before. The learning rates used were $\gamma_1 = 2.5$, $\gamma_2 = 0.01$ and $\gamma_3 = 0.1$ for K_p, K_i, and K_d, respectively. The step function with an initial value of 0 degrees and a final value of -20 was chosen as reference signal for this case.

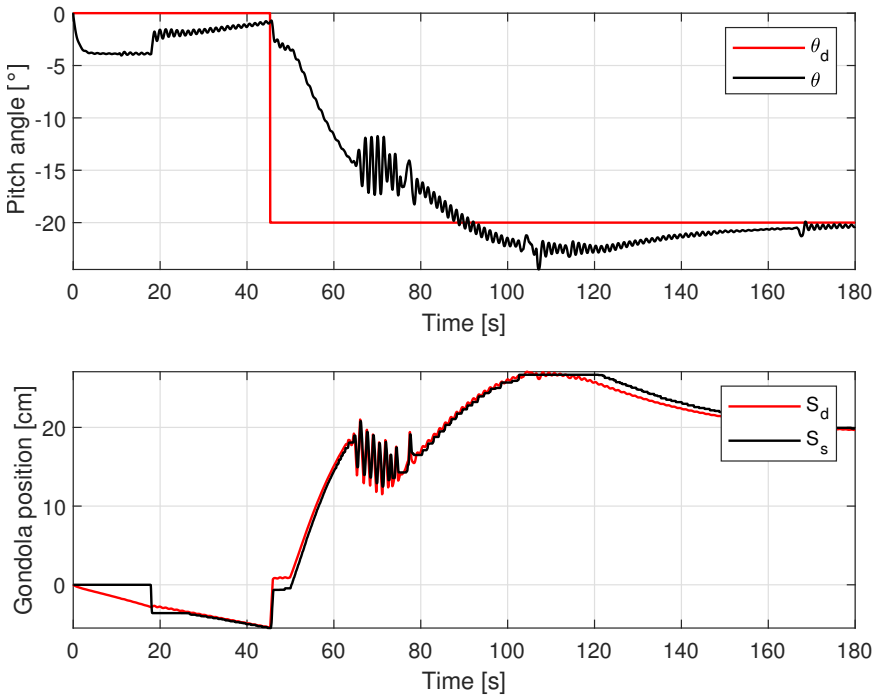


Figure 6.5: Pitch angle and gondola travelled distance control for $\gamma_1 = 2.5$, $\gamma_2 = 0.01$ and $\gamma_3 = 0.1$

Figure 6.5 shows that the controller starts computing its output, when the Simulink model is uploaded. However, the reference signal is received a couple of seconds later when the ground station is run. This characteristic is presented in the next experiments as well. Figure 6.5 also shows how the oscillations were reduced by the disconnection of the supervisory control action during the experiment.

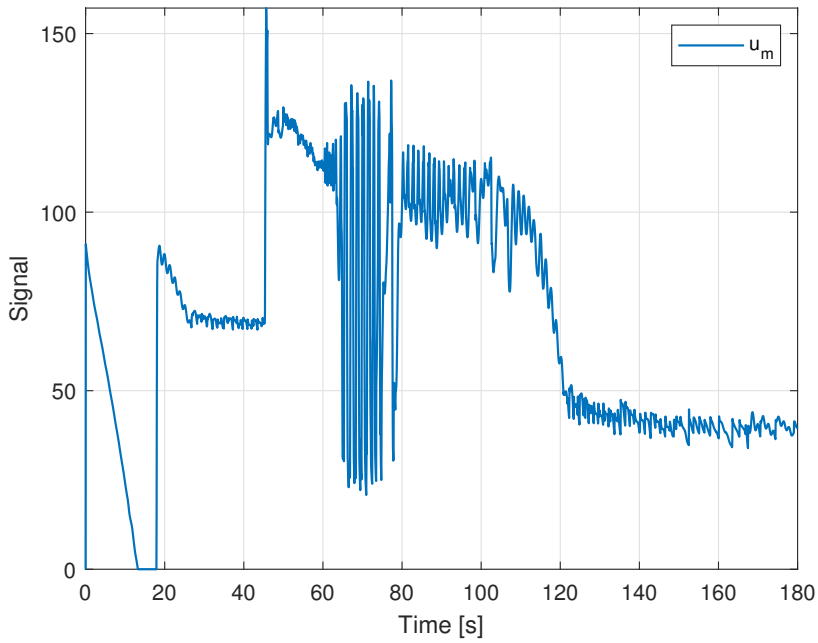


Figure 6.6: Gondola motor signal for $\gamma_1 = 2.5$, $\gamma_2 = 0.01$ and $\gamma_3 = 0.1$

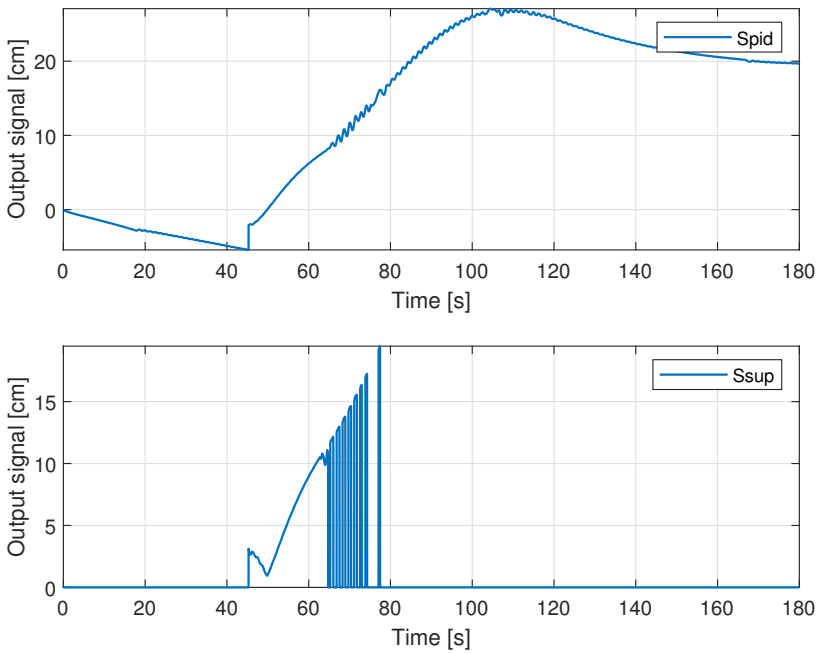


Figure 6.7: Control action of adaptive PID and supervisory controller for $\gamma_1 = 2.5$, $\gamma_2 = 0.01$ and $\gamma_3 = 0.1$

On the other hand, Figures 6.6 and 6.7 demonstrate how the chattering phe-

nomenon was reduced significantly. Moreover, Figure 6.7 shows that the supervisory controller acts as a boost for the adaptive PID. This allows the adaptive controller to adjust its gains faster.

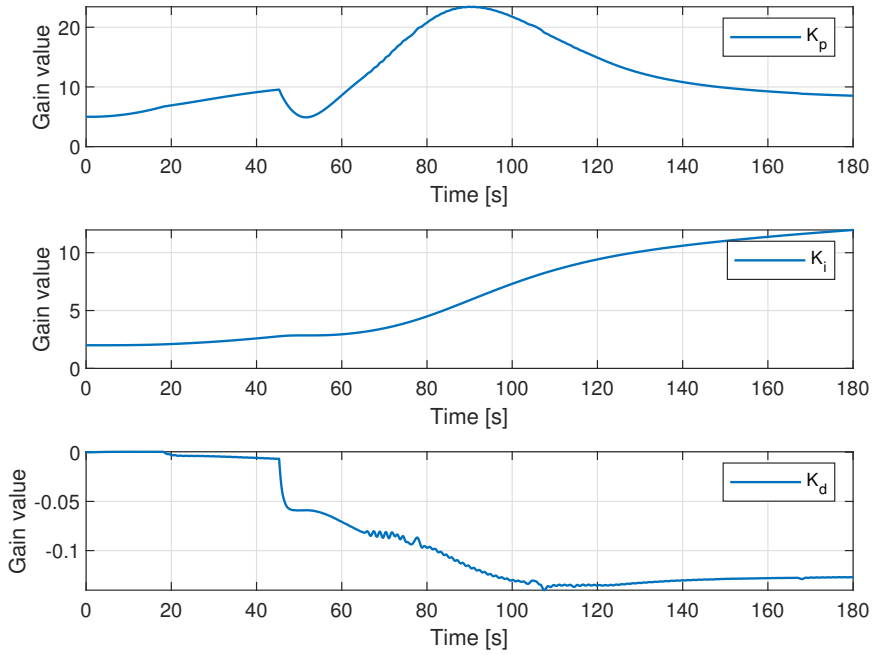


Figure 6.8: Adaptive PID gains for $\gamma_1 = 2.5$, $\gamma_2 = 0.01$ and $\gamma_3 = 0.1$

Finally, it is possible to see in Figure 6.8, how the PID gains converge to specific values during this test. However, the gains take more time to converge. This could be caused by the learning rates values. This effect is studied in Section 6.4.

6.4 Effect of Learning Rates

The learning rates of the adaptation law relate how fast the system is able to "learn" the plant dynamics and converge to specific gain values for the controller. In previous experiments, it was observed that in this case, the K_p and K_i gains have the strongest relation to the system's response. Therefore, their respective learning rates were varied and analyzed.

Figures 6.9, 6.10, 6.11, and 6.12 show the results for the following learning rates $\gamma_1 = 4$, $\gamma_2 = 0.01$ and $\gamma_3 = 0.1$ for K_p, K_i, and K_d, respectively. The reference signal for this case was a trajectory that started at 0 degrees and went to -15 degrees in 50 s. After, it held the -15 degrees inclination for 40 s and finally went back to zero degrees in 50 s.

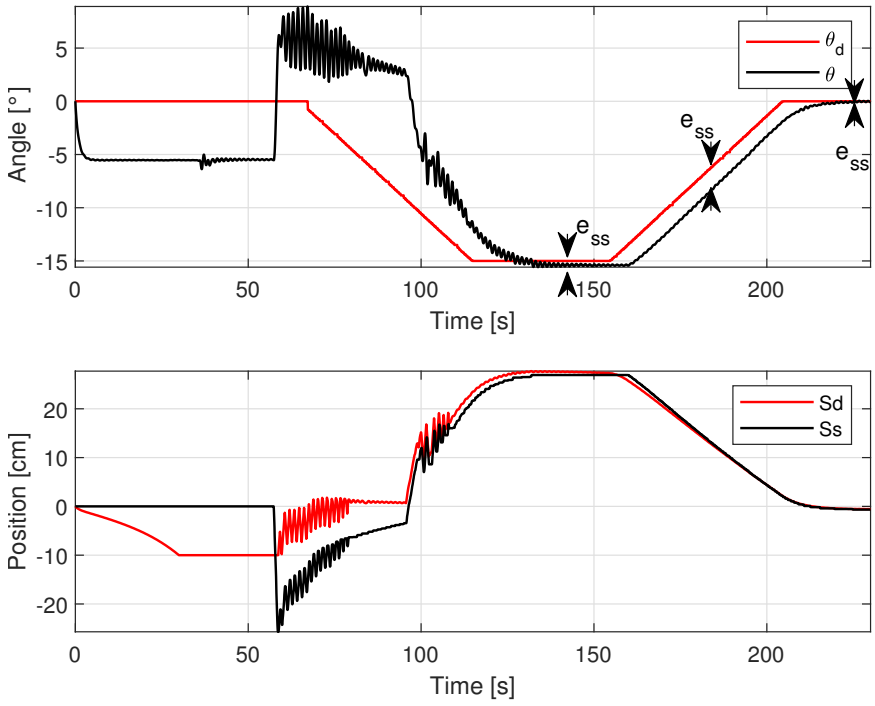


Figure 6.9: Pitch angle and gondola travelled distance control for $\gamma_1 = 4$, $\gamma_2 = 0.01$ and $\gamma_3 = 0.1$

In Figure 6.9 we see that some oscillations are present, as explained in the previous section. Figure 6.9 shows that in steady state, the system has an error of less than 2 degrees, during the ascending manoeuvre. Additionally, the controller presents an error of 0.3 degrees, when it holds the -15 degree inclination. The error present when the system goes back and holds the 0 pitch angle is less than 0.1 degrees.

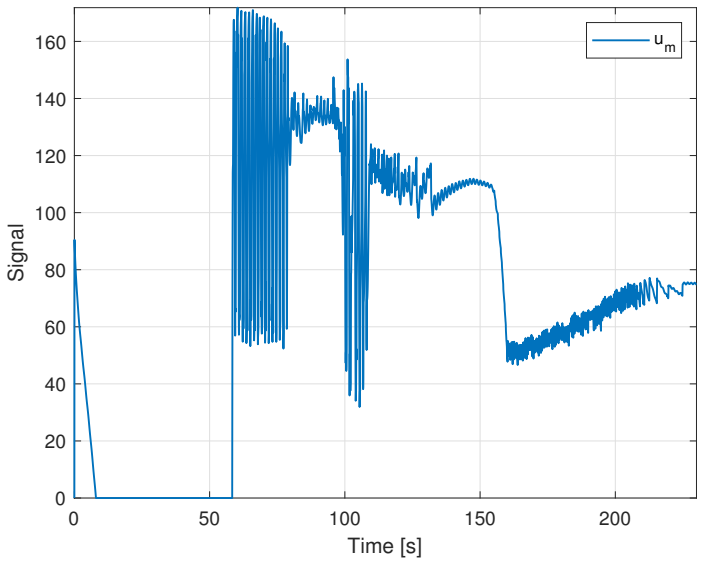


Figure 6.10: Gondola motor signal for $\gamma_1 = 4$, $\gamma_2 = 0.01$ and $\gamma_3 = 0.1$

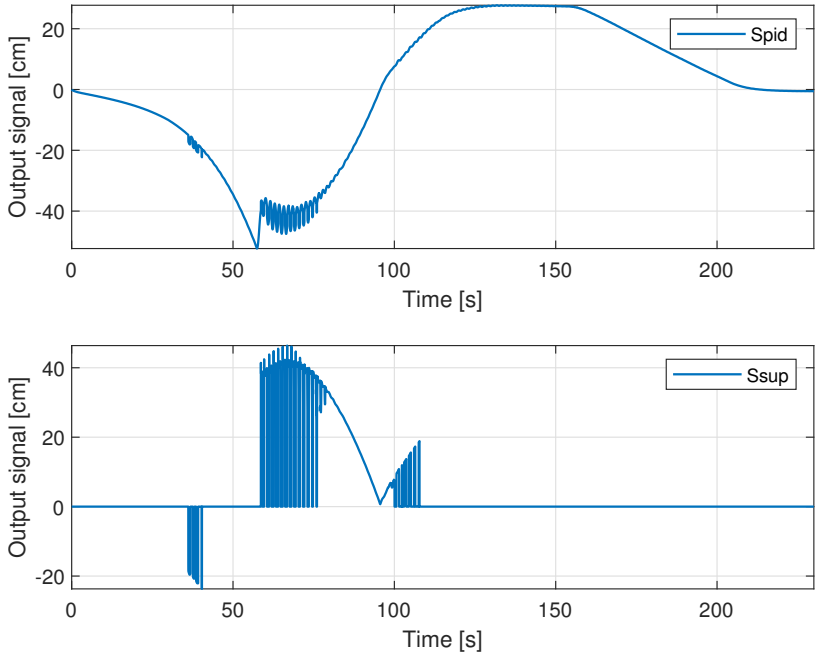


Figure 6.11: Control action of adaptive PID and supervisory controller for $\gamma_1 = 4$, $\gamma_2 = 0.01$ and $\gamma_3 = 0.1$

Figure 6.10 clearly shows how the gearmotors started with the maximum signal, once the reference signal was received. This phenomenon was described in detail in

Section 6.1. Figure 6.11 demonstrates how the supervisory controller acted from 60 s (when the reference signal was received for the first time) until 110 s. It was then disconnected until the end of the experiment. Its control produced a few oscillations at the beginning; however, it helped correct the adaptive PID controller. From 110 s until the end of the test, the control output came entirely from the adaptive controller.

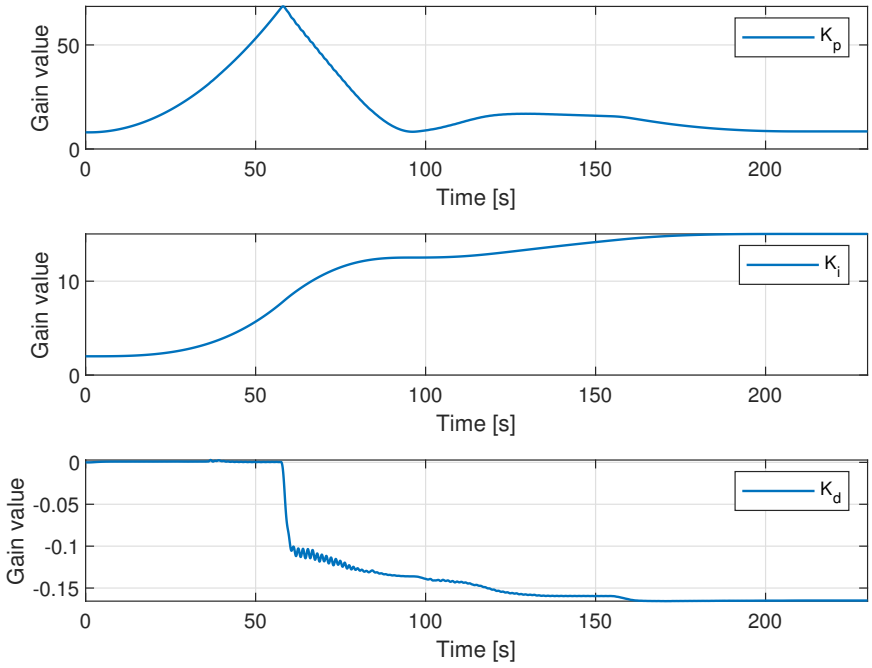


Figure 6.12: Adaptive PID gains for $\gamma_1 = 4$, $\gamma_2 = 0.01$ and $\gamma_3 = 0.1$

Figure 6.12 presents how the PID gains converge during the test, in approximately 160 seconds. This demonstrates that the adaptation algorithm is working properly. However, some improvements in the learning rates are needed.

In order to reduce and/or eliminate the steady state error obtained in the results presented above for the ramp signal, the K_i learning rate was increased. Figures 6.13, 6.14, 6.15, and 6.16 show the results for the following learning rates $\gamma_1 = 4$, $\gamma_2 = 0.3$ and $\gamma_3 = 0.1$ for K_p, K_i, and K_d, respectively. The reference signal for this case was a trajectory that started at -5 degrees and went to -20 degrees in 50 s. Finally, it held the position and then went back to -5 degrees with the same slope.

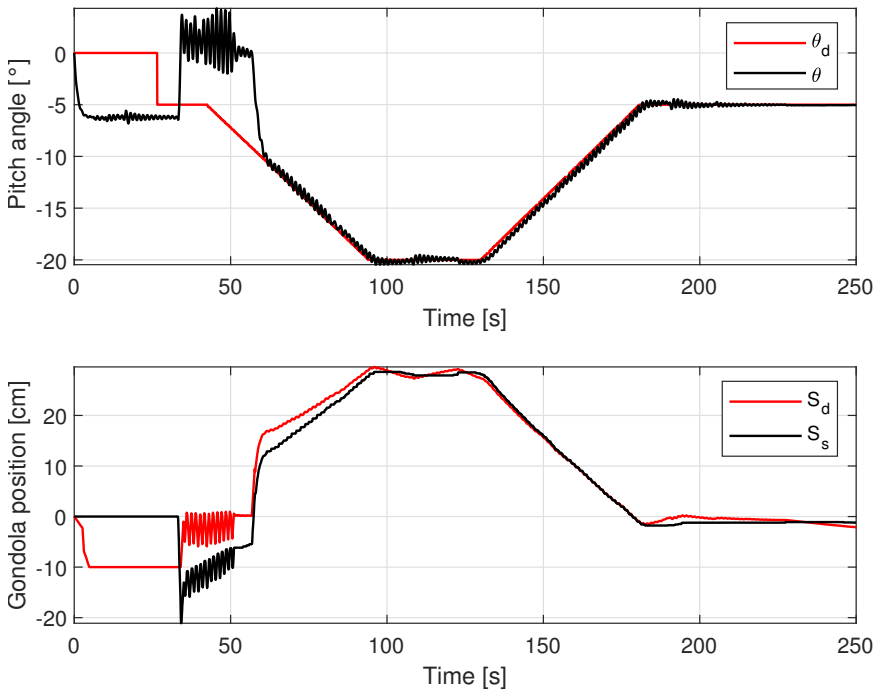


Figure 6.13: Pitch angle and gondola travelled distance control for $\gamma_1 = 4$, $\gamma_2 = 0.3$ and $\gamma_3 = 0.1$

Figure 6.13 shows that in a steady state, the system has an average error of 0.6 degrees, during the descending and ascending manoeuvres. Additionally, the controller presents a maximum error of 0.3 degrees, when it holds the -20 degrees inclination. The error present when the system went back and held -5 pitch angle is less than 0.1 degrees. The settling time was of approximately 30 s.

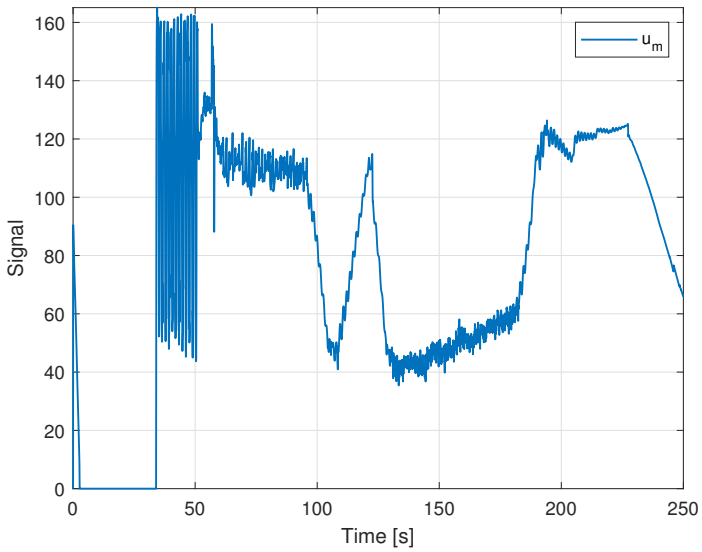


Figure 6.14: Gondola motor signal for $\gamma_1 = 4$, $\gamma_2 = 0.3$ and $\gamma_3 = 0.1$

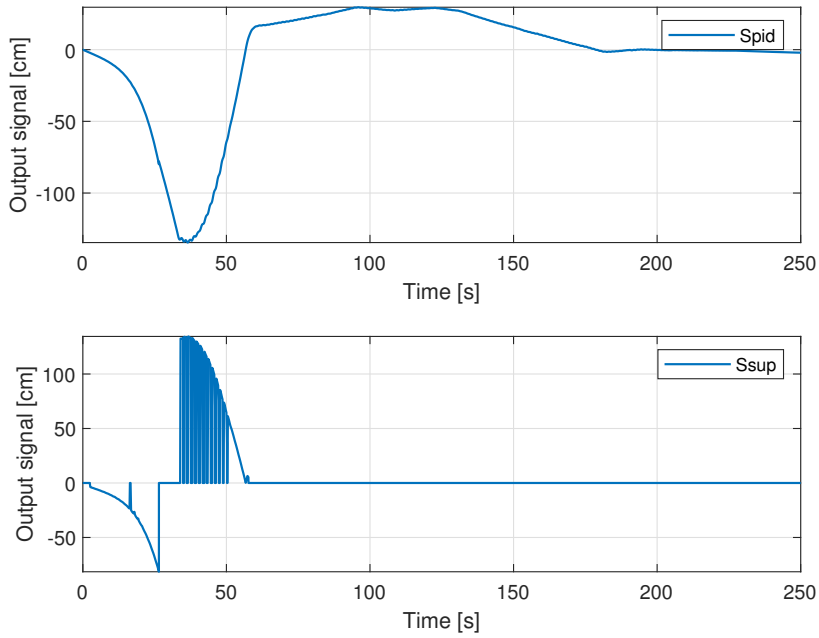


Figure 6.15: Control action of adaptive PID and supervisory controller for $\gamma_1 = 4$, $\gamma_2 = 0.3$ and $\gamma_3 = 0.1$

On the other hand, Figure 6.15 demonstrates how the supervisory controller acted from 30 s until 60 s; it then remained disconnected until the end of the experiment.

Its control action produced a few oscillations at the beginning; however, it helped to correct the adaptive PID controller. From 60 s until the end of the test, the control output came entirely from the adaptive controller.

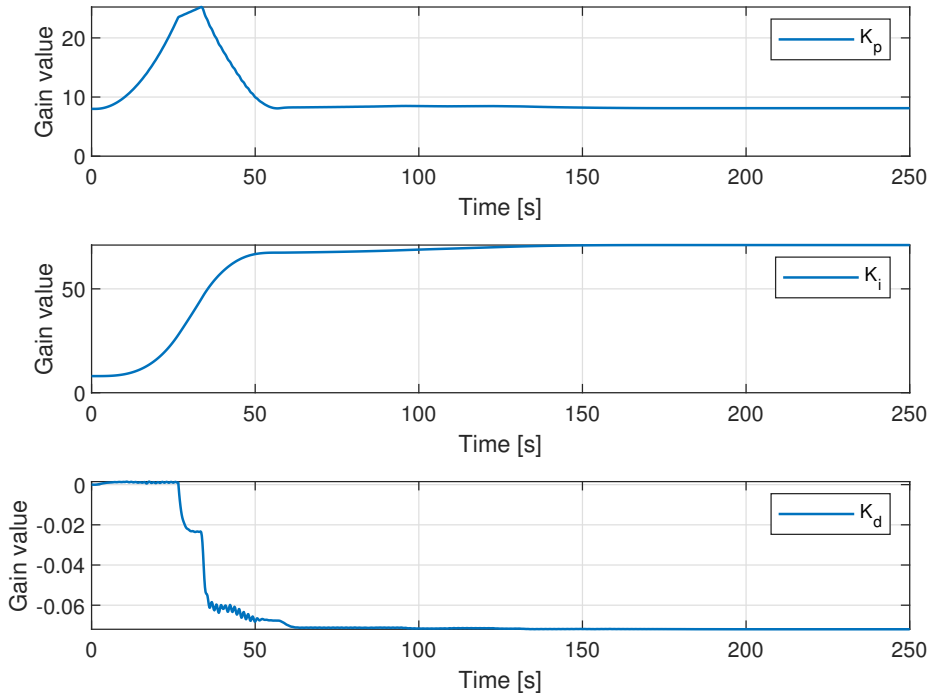


Figure 6.16: Adaptive PID gains for $\gamma_1 = 4$, $\gamma_2 = 0.3$ and $\gamma_3 = 0.1$

Figure 6.16 presents how the PID gains converged during the test, in approximately 35 s, once the first reference signal data was received. This demonstrates that the adaptation algorithm is working properly. Finally, the system performance was improved to reduce the settling time, the steady state error, and the convergence time.

More experiments that included multiples variations in these rates were carried out, but the data is not shown. A general observation for all of experiments is that the higher γ_1 values, the more undesired oscillations are present. In addition, small values produce a slow response from the system, thus reducing its tracking performance. A similar behaviour was observed for the learning rate γ_2. Contrarily, γ_3 does not show a significant effect for this case. This could be caused by how the K_d acts when the

setpoint changes greatly. For the reference signals used, the changes were made to a few periods of time. However, this probably does not give enough time and data to converge to a considerable gain. Nonetheless, it helps avoid the undesired peaks that are usually produced by the derivative terms.

6.5 Performance of the Selected Learning Rates Under Different Scenarios

In this section, the response of the selected learning rates under different types of reference signals was studied. These signals include a step function and a trajectory.

6.5.1 Step

For this case, the step function initializes at -5° and was then changed to -15 degrees. Figures 6.17, 6.18, 6.19, and 6.20 show the results for this reference signal using the following learning rates $\gamma_1 = 4$, $\gamma_2 = 0.3$ and $\gamma_3 = 0.1$ for K_p, K_i, and K_d, respectively.

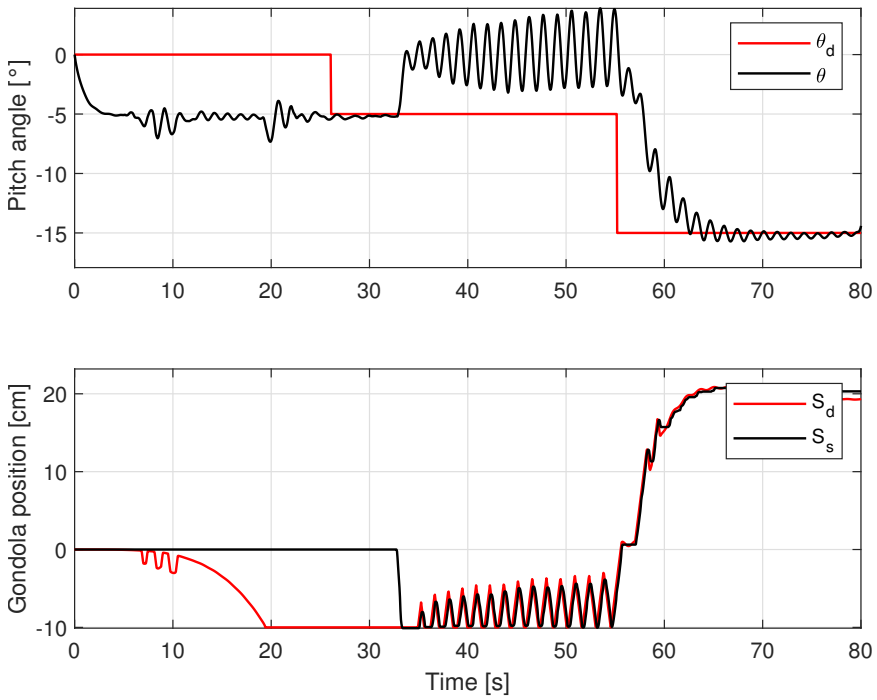


Figure 6.17: Step response of pitch angle and gondola travelled distance for $\gamma_1 = 4$, $\gamma_2 = 0.3$ and $\gamma_3 = 0.1$

Figure 6.17 shows that the system has a settling time of 8 s. Additionally, the controller has a 4 % overshoot. Despite the step reference, it is not the most suitable signal to excite the system, but the overall response is good.

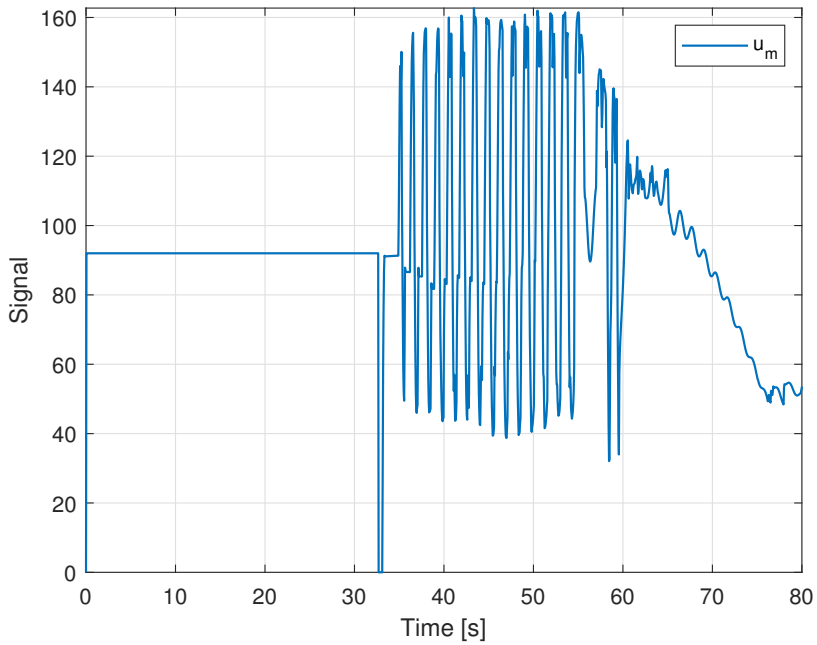


Figure 6.18: Gondola motor signal for a step reference with $\gamma_1 = 4$, $\gamma_2 = 0.3$ and $\gamma_3 = 0.1$

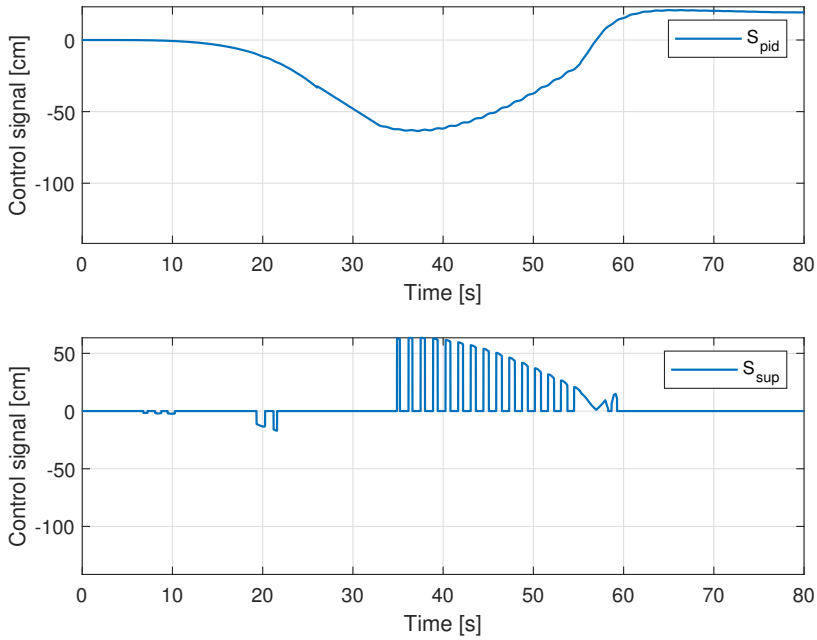


Figure 6.19: Control action of adaptive PID and supervisory controller for a step reference with $\gamma_1 = 4$, $\gamma_2 = 0.3$ and $\gamma_3 = 0.1$

Figure 6.19 demonstrates how the supervisory controller acted for only 15 s and then remained disconnected until the end of the experiment. Its control action produced a few oscillations during the transient response. However, it helped to adjust the adaptive PID controller. From 60 s until the end of the test, the control output came entirely from the adaptive controller.

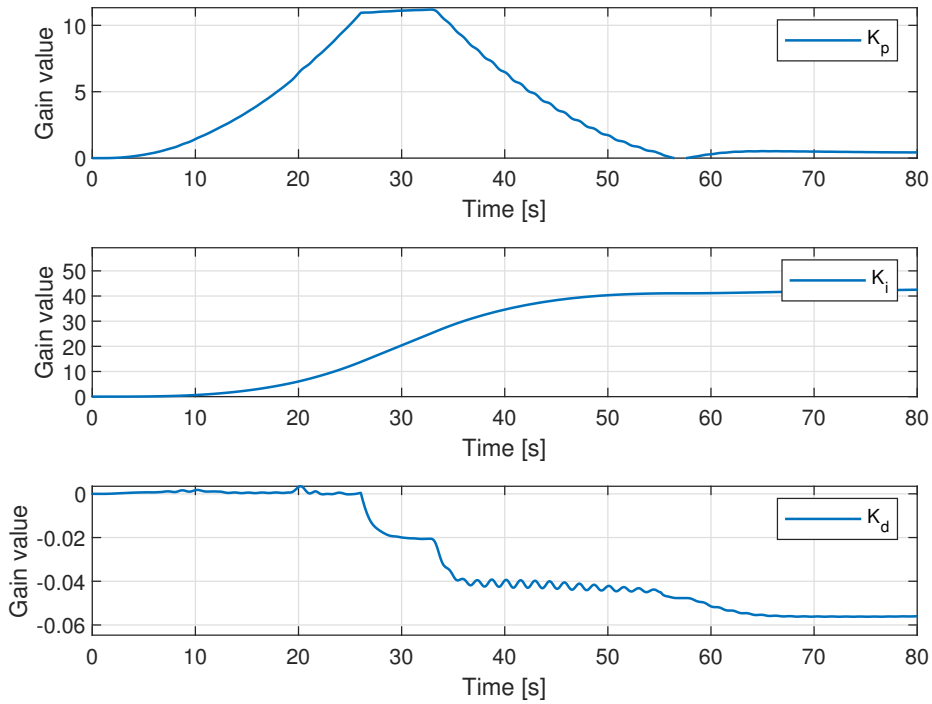


Figure 6.20: Adaptive PID and supervisory controller for step reference with $\gamma_1 = 4$, $\gamma_2 = 0.3$ and $\gamma_3 = 0.1$

Figure 6.20 shows how the PID gains converged approximately 16 s after the step function was applied. This demonstrates that the adaptation algorithm is working properly. This result agrees with those in Figure 6.19, where in the instant that the gains converged, the airship reached the desired inclination..

6.5.2 Trajectory

The desired trajectory initializes at -5 degrees and goes to -20 degrees in 30 seconds. It holds the position for 15 seconds and then goes back to -5 degrees with the same

slope. Finally, it holds the -5 degrees position for the remaining time. Figures 6.21, 6.22, 6.23, and 6.24 show the results for this reference signal using the following learning rates $\gamma_1 = 4$, $\gamma_2 = 0.3$ and $\gamma_3 = 0.1$ for K_p, K_i, and K_d, respectively.

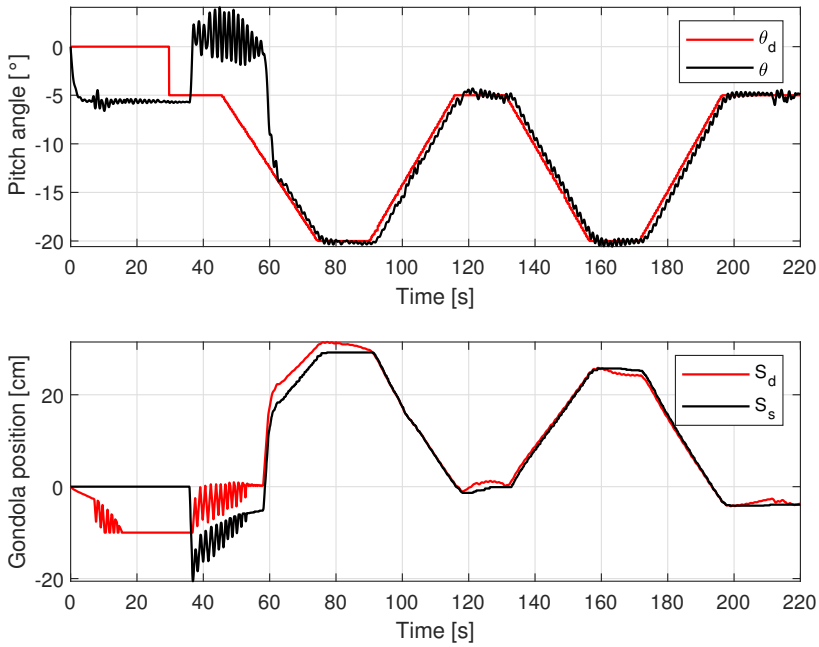


Figure 6.21: Response of pitch angle and gondola travelled distance for a trajectory reference with $\gamma_1 = 4$, $\gamma_2 = 0.3$ and $\gamma_3 = 0.1$

Figure 6.21 shows that in a steady state, the system has an error of 0.45 degrees and of 0.8 degrees, for the descending and ascending manoeuvres, respectively. Additionally, the controller presents an error of 0.3 degrees and a 1.5 % overshoot, when holding the -20 degree inclination.

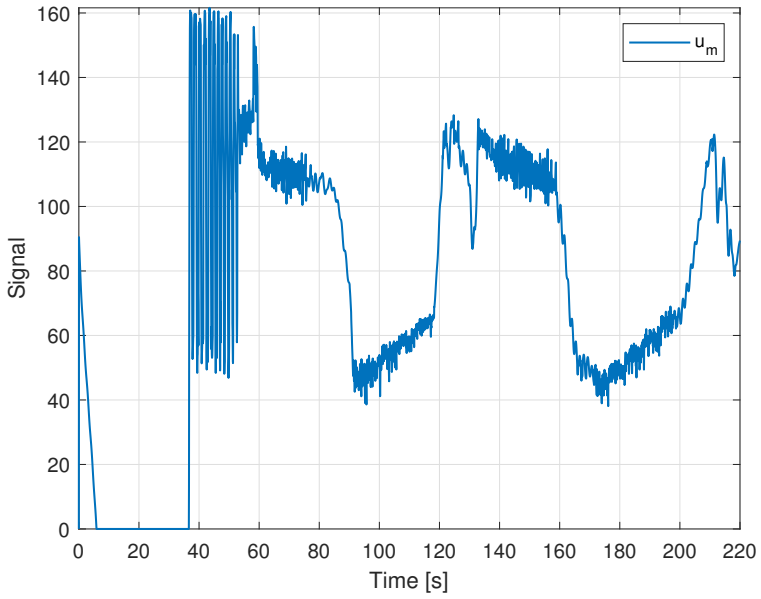


Figure 6.22: Gondola motor signal for a trajectory reference signal with $\gamma_1 = 4$, $\gamma_2 = 0.3$ and $\gamma_3 = 0.1$

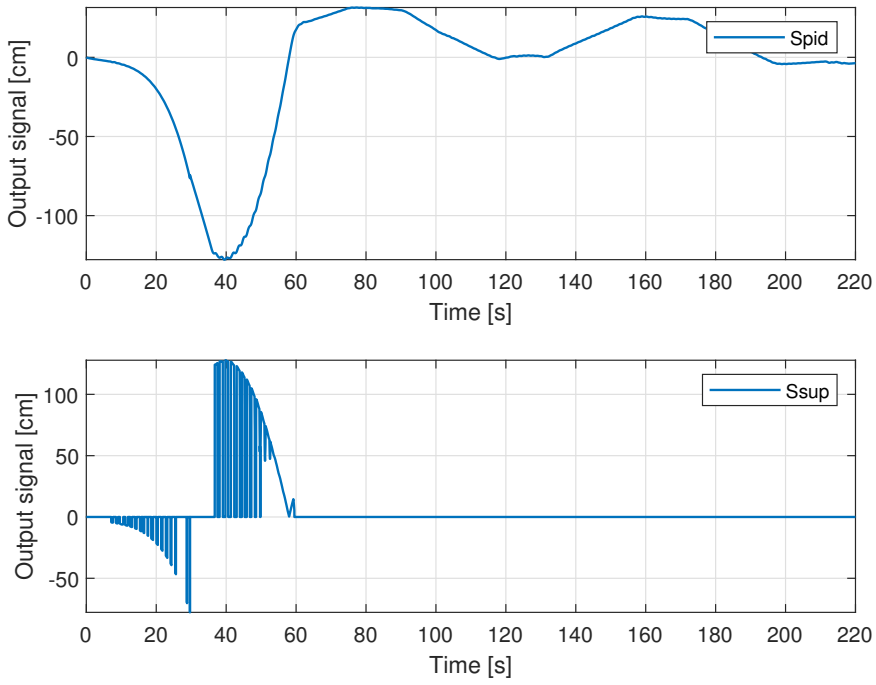


Figure 6.23: Control action of adaptive PID and supervisory controller for a trajectory reference with $\gamma_1 = 4$, $\gamma_2 = 0.3$ and $\gamma_3 = 0.1$

Figure 6.23 shows how the supervisory controller only acted for 15 s, once the reference signal was received. Once this control action supported the adaptive PID's convergence, it remained disconnected until the end of the experiment. From 110 s until the end of the test, the control output came entirely from the adaptive controller.

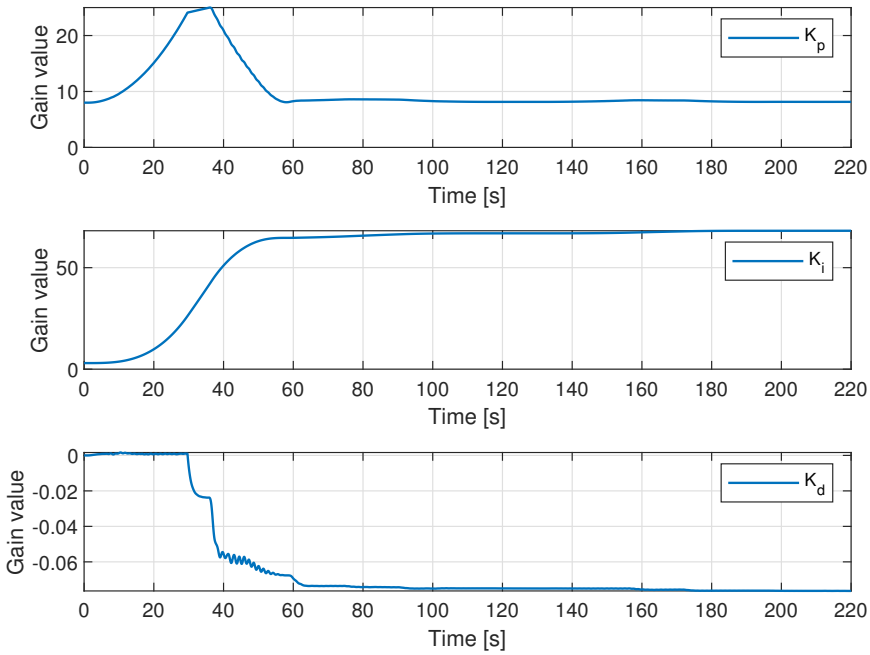


Figure 6.24: Adaptive PID gains for a trajectory reference with $\gamma_1 = 4$, $\gamma_2 = 0.3$ and $\gamma_3 = 0.1$

Figure 6.24 presents how the PID gains converge, in approximately in 25 s, after the reference signal was received.

6.5.3 Disturbance Rejection

The disturbance rejection capability of the proposed controller was tested. For this, a fan was used as a source of disturbance. The air flow was applied in the direction shown in Figure 6.25.

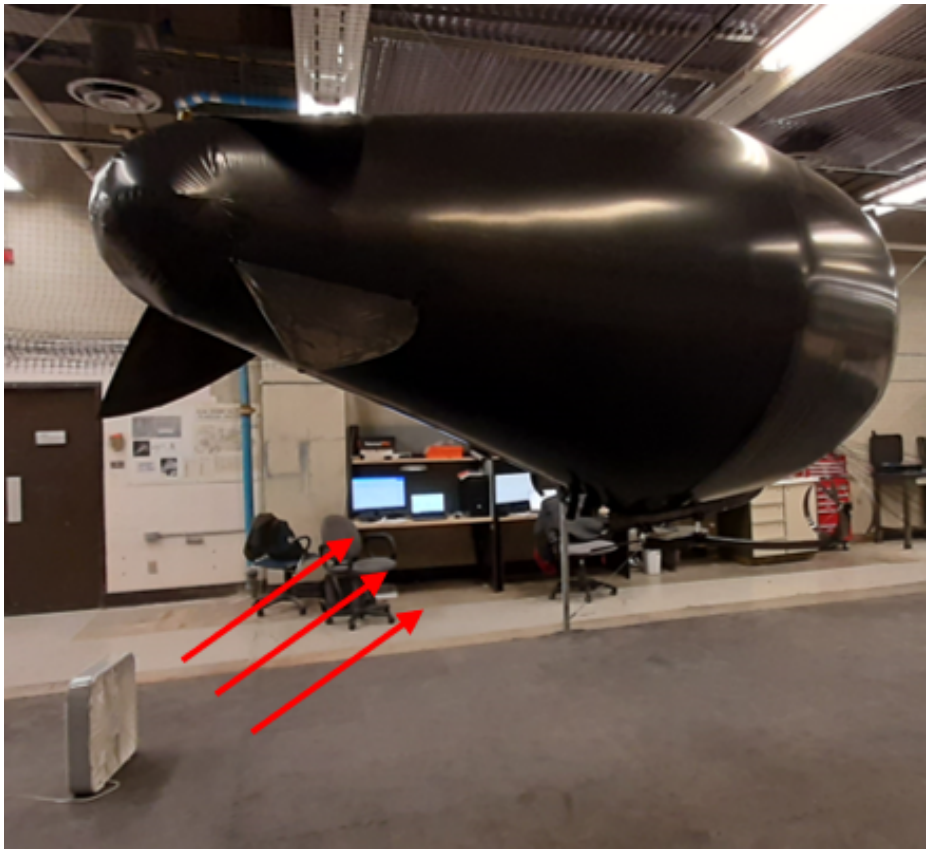


Figure 6.25: Experiment setup under wind disturbance

The reference signal was a -15° step. Figures 6.26, 6.27, 6.28, and 6.29 show the results for this reference signal using the following learning rates $\gamma_1 = 4$, $\gamma_2 = 0.3$ and $\gamma_3 = 0.1$ for K_p, K_i, and K_d, respectively.

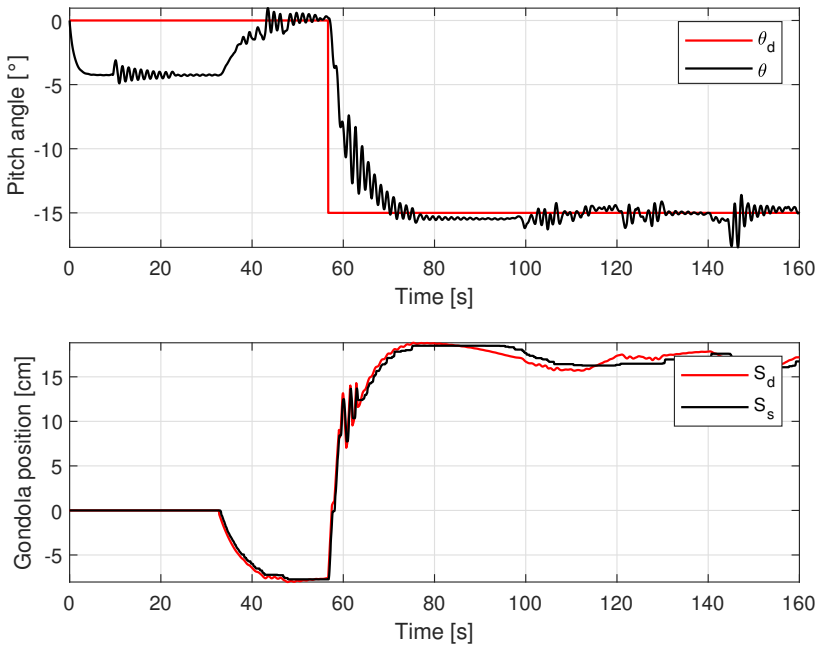


Figure 6.26: Response of pitch angle and gondola travelled distance for a step reference under disturbance with $\gamma_1 = 4$, $\gamma_2 = 0.3$ and $\gamma_3 = 0.1$.

Figure 6.26 shows that the system has an overshoot of less than 5 % and a settling time of 14.7 s. Additionally, we can observe the response of the controller to external disturbances, applied to the system when it was holding the -15° angle.

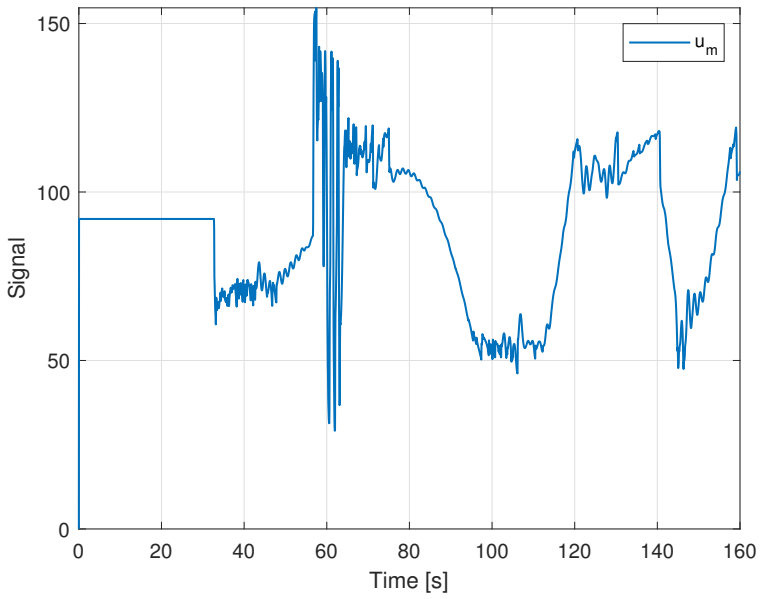


Figure 6.27: Gondola motor signal for a step reference under disturbance with $\gamma_1 = 4$, $\gamma_2 = 0.3$ and $\gamma_3 = 0.1$

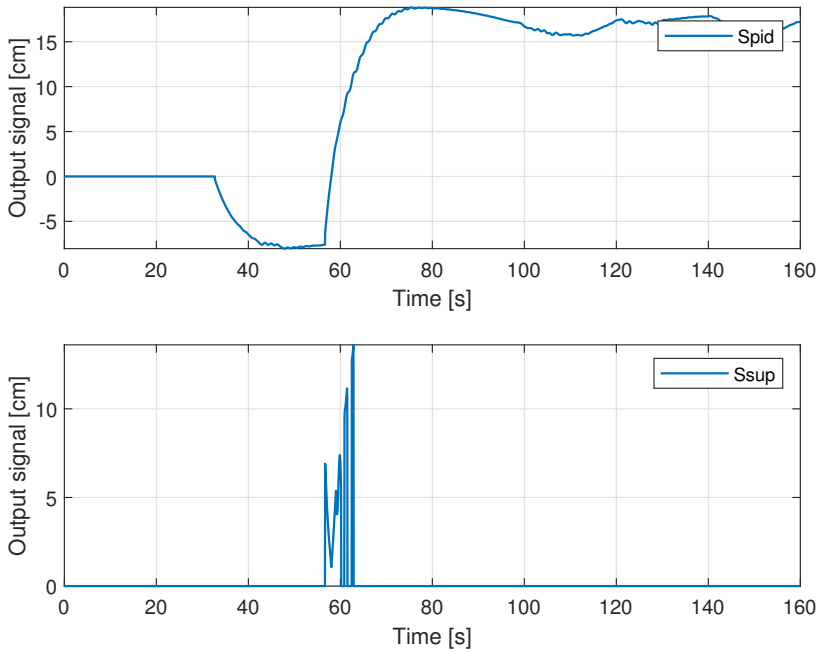


Figure 6.28: Control action of adaptive PID and supervisory controller for a step reference under disturbance with $\gamma_1 = 4$, $\gamma_2 = 0.3$ and $\gamma_3 = 0.1$

Figure 6.28 shows how the supervisory controller acted for only 8 s once the reference signal was received; it remained disconnected after 63 s. This means that the PID adaptive controller was able to manage the disturbance without the supervisory controller.

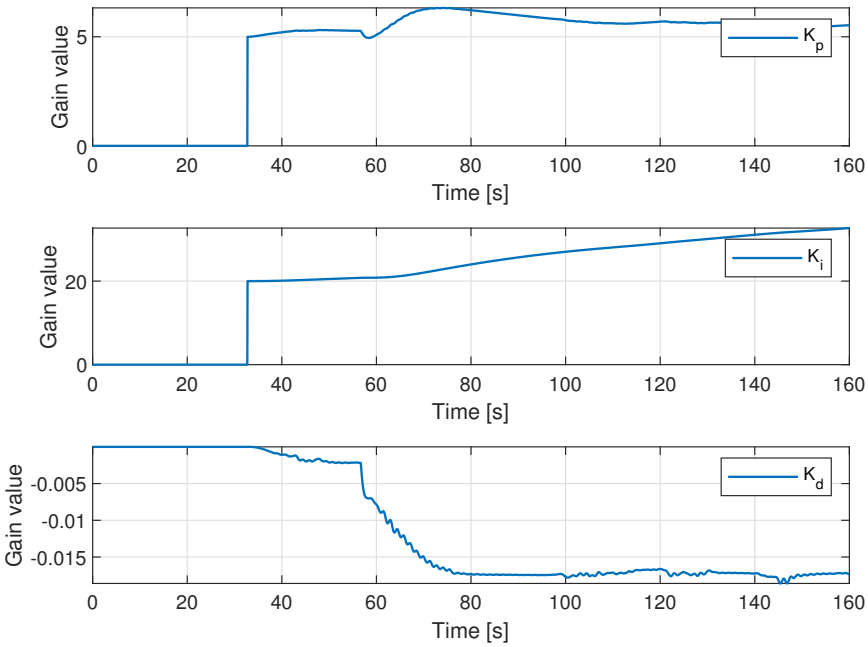


Figure 6.29: Adaptive PID gains for a step reference under disturbance with $\gamma_1 = 4$, $\gamma_2 = 0.3$ and $\gamma_3 = 0.1$.

Figure 6.29 presents how the K_p and K_d gains converged once the disturbance was applied. However, K_i required more time to adapt. This could be caused due to the γ_2 value and/or the fluctuations produced by the disturbances. Despite this behavior, the controller was able to manage the disturbance and keep the desired pitch angle.

6.5.4 Partial Actuator Failure

The adaptation capability of the proposed controller under actuator failure was evaluated. For this test, one of the gondola motors had its gears broken. The gearmotor was making a sound during its use and its torque transmission was limited. For this test, the reference signal was a -10° step. Figures 6.30, 6.31, 6.32, and 6.33 show the

results for this reference signal using the following learning rates $\gamma_1 = 3$, $\gamma_2 = 0.3$, and $\gamma_3 = 0.1$ for K_p, K_i, and K_d, respectively.

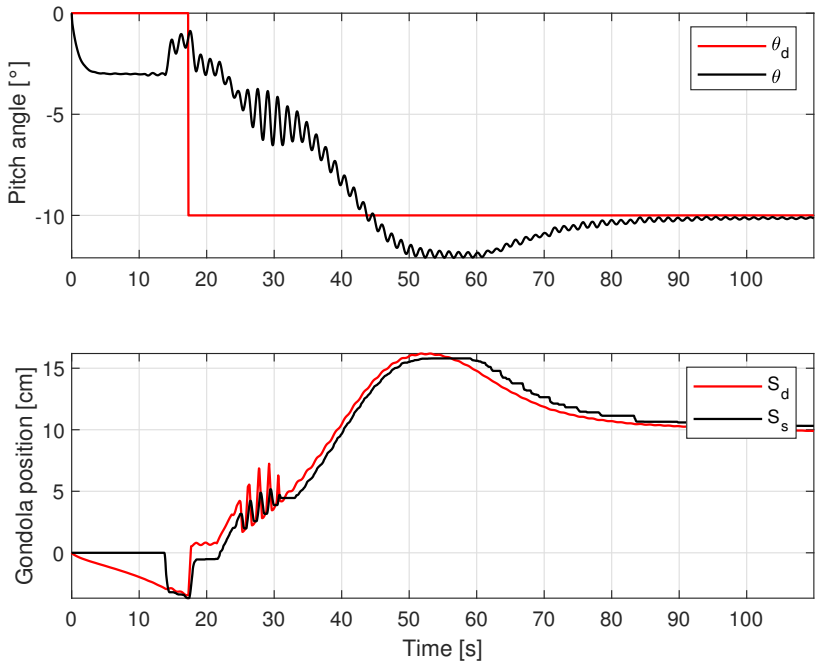


Figure 6.30: Response of pitch angle and gondola travelled distance for a trajectory reference with $\gamma_1 = 3$, $\gamma_2 = 0.3$ and $\gamma_3 = 0.1$

Figure 6.30 shows that the response has a 20 % overshoot and a settling time of 61 s, once the reference signal was received.

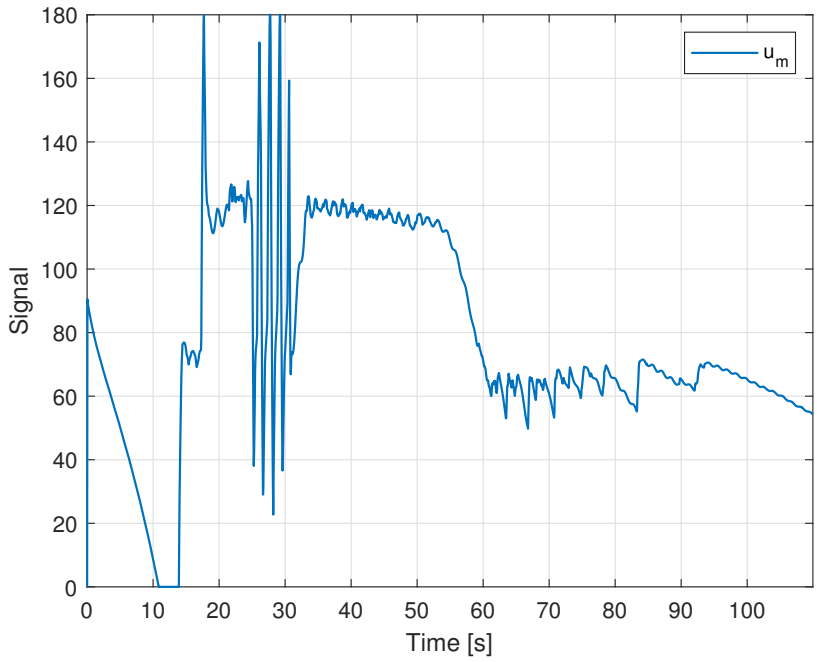


Figure 6.31: Gondola motor signal for a trajectory reference signal with $\gamma_1 = 3$, $\gamma_2 = 0.3$ and $\gamma_3 = 0.1$

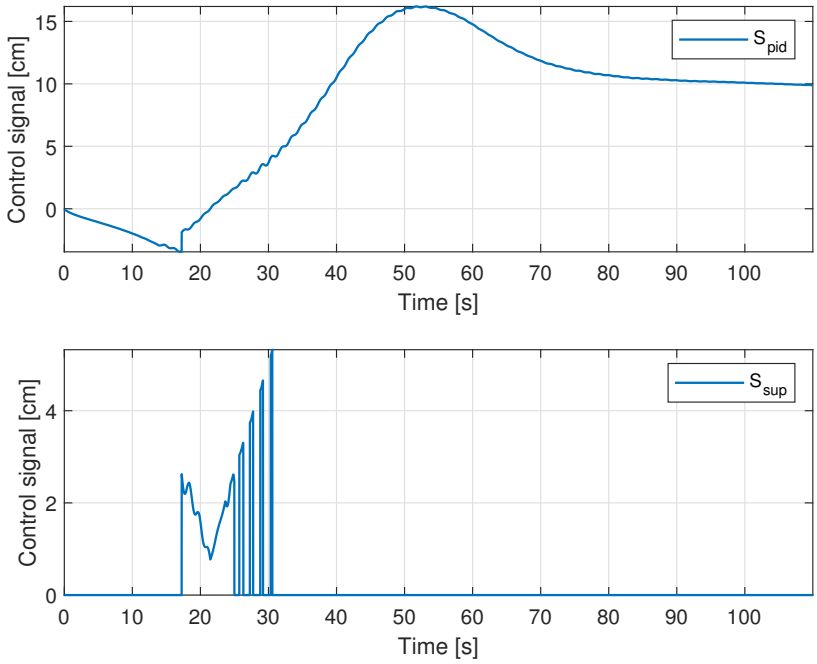


Figure 6.32: Control action of adaptive PID and supervisory controller for a trajectory reference with $\gamma_1 = 3$, $\gamma_2 = 0.3$ and $\gamma_3 = 0.1$

Figure 6.32 shows how the supervisory controller acted for only 13 s once the reference signal was received, and remained disconnected after 30 s. This means that the PID adaptive controller was able to manage the partial actuator failure by itself.

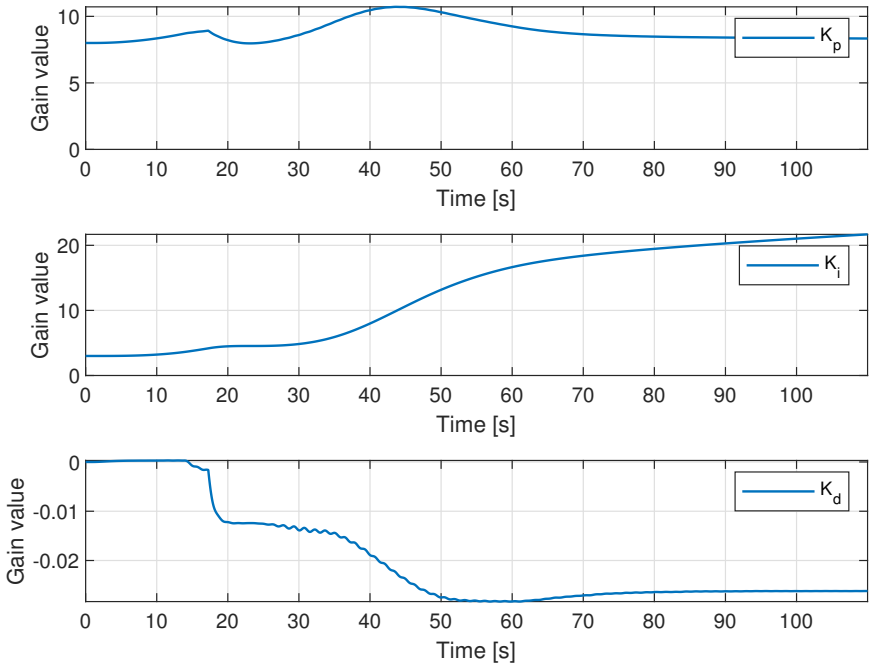


Figure 6.33: Adaptive PID gains for a trajectory reference with $\gamma_1 = 3$, $\gamma_2 = 0.3$ and $\gamma_3 = 0.1$

Figure 6.33 presents how the K_p and K_d gains converged, after the disturbance was applied. However, K_i required more time to adapt. This could be caused by the small steady state error. Despite this behavior, the controller was able to manage the actuator failure and keep the desired pitch angle.

Chapter 7

Conclusions

In this thesis, the longitudinal multi-body equations of motion of an autonomous airship with a sliding gondola were derived using the Boltzmann - Hamel approach. Then, an adaptive self-tunning PID controller with a supervisory controller was designed. The model and the controller were implemented in Matlab/Simulink. Several simulations were carried out while considering wind disturbance. The components of the experimental prototype including the hardware and graphic user interface (GUI) of the ground station were described. The designed user interface allows to switch between manual mode (using an RC control) and automatic mode (controller) remotely. Finally, numerous flight tests were performed to evaluate the performance of the pitch controller.

7.1 Contributions

- Derived the longitudinal multi-body equations of motion for an unmanned reconfigurable airship with a sliding gondola using the Boltzmann - Hamel approach.
- Developed and validated the proposed controller technique improvements via simulations and experimental tests.

- Conducted the first trajectory tracking experiments on the prototype.

7.2 Future Work

- The development of the 3D multi-body model including the longitudinal and lateral equations.
- The inclusion of experimentally-derived system parameters such as inertia and aerodynamic coefficients in the dynamic model.
- The improvement of the supervisory controller with a softer technique (e.g., fuzzy logic, Neural network, etc.).

References

[1] Yasmina Bestaoui Sebbane, S. G. Tzafestas, P. Antsaklis, P. Borne, D. G. Caldwell, C. S. Chen, T. Fukuda, S. Monaco, G. Schmidt, F. Harashima, N. K. Sinha, D. Tabak, and K. Valavanis. *Lighter than Air Robots*, volume 58 of *Intelligent Systems, Control and Automation: Science and Engineering*. Springer Netherlands, Dordrecht, 2012.

[2] Yuwen Li, Meyer Nahon, and Inna Sharf. Airship dynamics modeling: A literature review. *Progress in Aerospace Sciences*, 47(3):217–239, April 2011.

[3] The Civil Aviation Authorities Luftfahrt-Bundesamt and Rijksluchtvaartdienst. Transport Airship Requirements, 2000.

[4] Federal Aviation Administration. Airship Design Criteria - No. FAA-P-8110-2, 1995.

[5] Transport Canada. Part V - Airworthiness Manual Chapter 541 - Airships. https://tc.canada.ca/en/corporate-services/acts-regulations/list-regulations/canadian-aviation-regulations-sor-96-433/standards/part-v-airworthiness-manual-chapter-541-airships, 2020.

[6] Giuseppe Valiante. Quebec premier says returns will fly high with $30-million airship investment. https://montreal.ctvnews.ca/quebec-premier-says-returns-will-fly-high-with-30-million-airship-investment-1.4799285, February 2020.

[7] Pixabay. Zeppelin Airship Flying - Free photo on Pixabay. https://pixabay.com/photos/zeppelin-airship-flying-aviation-2604010/, 2021.

[8] Inc Skyship Services. Skyship Media. http://www.skyshipservices.com/media.html, 2020.

[9] Pixabay. Free Image on Pixabay - Blimp, Goodyear Blimp, Dirigible. https://pixabay.com/photos/blimp-goodyear-blimp-dirigible-5314195/, 2021.

[10] Pixabay. Free Image on Pixabay - Fly, Airship, Air, Sky. https://pixabay.com/photos/fly-airship-air-sky-transportation-1613149/, 2021.

[11] Eric Lanteigne, Ahmad Alsayed, Dominic Robillard, and Steven G. Recoskie. Modeling and Control of an Unmanned Airship with Sliding Ballast. *Journal of Intelligent & Robotic Systems*, 88(2-4):285–297, December 2017.

[12] Roman Fedorenko and Victor Krukhmalev. Indoor Autonomous Airship Control and Navigation System. *MATEC Web of Conferences*, 42:01006, 2016.

[13] Yueneng Yang and Ye Yan. Neural network gain-scheduling sliding mode control for three-dimensional trajectory tracking of robotic airships. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 229(6):529–540, July 2015.

[14] S. Yu, G. Xu, K. Zhong, S. Ye, and W. Zhu. Direct-adaptive fuzzy predictive control for path following of stratospheric airship. In *2017 29th Chinese Control And Decision Conference (CCDC)*, pages 5658–5664, May 2017.

[15] Zewei Zheng and Liang Sun. Adaptive sliding mode trajectory tracking control of robotic airships with parametric uncertainty and wind disturbance. *Journal of the Franklin Institute*, 355(1):106–122, January 2018.

[16] Lin Cheng, Zongyu Zuo, Jiawei Song, and Xiao Liang. Robust three-dimensional path-following control for an under-actuated stratospheric airship. *Advances in Space Research*, 63(1):526–538, January 2019.

[17] Grant E. Carichner and Leland M. Nicolai. *Fundamentals of Aircraft and Airship Design. Volume 2-Airship Design and Case Studies*, volume 2 of *AIAA Education Series*. American Institute of Aeronautics and Astronautics, Inc., 2013.

[18] Davi Antonio dos Santos and Americo Cunha. Flight control of a hexa-rotor airship: Uncertainty quantification for a range of temperature and pressure conditions. *ISA Transactions*, 93:268–279, October 2019.

[19] Farbod Khoshnoud, Ibrahim I. Esat, Clarence W. de Silva, Jason D. Rhodes, Alina A. Kiessling, and Marco B. Quadrelli. Self-Powered Solar Aerial Vehicles: Towards Infinite Endurance UAVs. *Unmanned Systems*, 08(02):95–117, April 2020.

[20] Charles Blouin, Eric Lanteigne, and Wail Gueaieb. Optimal control for the trajectory planning of micro airships. In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 885–892, Miami, FL, USA, June 2017. IEEE.

[21] Yubin Wen, Li Chen, Kuankuan Liang, and Dengping Duan. Nonlinear MPC for a Sensorless Multi-Vectored Propeller Airship Based on Sliding Mode Observer with Saturation. *Asian Journal of Control*, 21(1):248–263, January 2019.

[22] S. B. V. Gomes. *An Investigation of the Flight Dynamics of Airships with Application to the YEZ-2A*. PhD thesis, Cranfield Institute of Technology, 1990.

[23] Steven Recoskie. *Autonomous Hybrid Powered Long Ranged Airship for Surveillance and Guidance*. PhD thesis, University of Ottawa, 2014.

[24] F. Li, W. Xu, Y. Shi, M. Cai, and X. Zhang. Multi-body dynamic modeling, simulation and control strategy design of a Y6 tilt rotor UAV. In *2017 2nd International Conference on Advanced Robotics and Mechatronics (ICARM)*, pages 373–379, August 2017.

[25] Yongchao Wang, Yaoming Zhou, and Chenghao Lin. Modeling and control for the mode transition of a novel tilt-wing UAV. *Aerospace Science and Technology*, 91:593–606, August 2019.

[26] Igor Henrique Beloti Pizetta, Alexandre Santos Brandao, and Mario Sarcinelli-Filho. Cooperative Load Transportation Using Three Quadrotors. In *2019 In-*

ternational Conference on Unmanned Aircraft Systems (ICUAS), pages 644–650, Atlanta, GA, USA, June 2019. IEEE.

[27] G. Caviglia and A. Morro. Kirchhoff's equations for the rigid body motion revisited. *Meccanica*, 52(6):1485–1489, April 2017.

[28] Alexander R. Galper. Generalized Kirchhoff Equations. *IFAC Proceedings Volumes*, 33(2):111–116, March 2000.

[29] Mahmut Reyhanoglu and Muhammad Rehan. Nonlinear Dynamics and Control of Aerial Robots. In Omar Dario Lopez Mejia and Jaime Alberto Escobar Gomez, editors, *Aerial Robots - Aerodynamics, Control and Applications*. InTech, September 2017.

[30] Fida Ben Abdallah, Naoufel Azouz, Lotfi Beji, and Azgal Abichou. Modeling of a heavy-lift airship carrying a payload by a cable-driven parallel manipulator. *International Journal of Advanced Robotic Systems*, 16(4):172988141986176, July 2019.

[31] Jonathan M. Cameron and Wayne J. Book. Modeling Mechanisms with Nonholonomic Joints Using the Boltzmann-Hamel Equations. *The International Journal of Robotics Research*, 16(1):47–59, February 1997.

[32] Donald T. Greenwood. *Advanced Dynamics*. Cambridge University Press, Cambridge ; New York, digitally printed 1st pbk. version edition, 2006.

[33] Jared A Grauer and James E Hubbard. Multibody Model of an Ornithopter. *Journal of Guidance, Control, and Dynamics*, 32(5):5, 2009.

[34] Eric Lanteigne and Joshua O'Reilly. Multibody Dynamic Modeling and Control of an Unmanned Aerial Vehicle under Non-Holonomic Constraints. In *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, page 6, 2020.

[35] Sohan Suvarna, Dibyayan Sengupta, Pavankumar Koratikere, and Rajkumar S. Pant. Simulation of Autonomous Airship on ROS-Gazebo Framework. In *2019 Fifth Indian Control Conference (ICC)*, pages 237–241, January 2019.

[36] Ahmad Alsayed. Pitch and Altitude Control of an Unmanned Airship with Sliding Gondola. Master Thesis, University of Ottawa, 2017.

[37] Jinkun Liu and Xinhua Wang. *Advanced Sliding Mode Control for Mechanical Systems*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[38] Antonella Ferrara. *Sliding Mode Control of Vehicle Dynamics*. The Institution of Engineering and Technolgy, England, 2017.

[39] Ely Carneiro de Paiva, Fábio Benjovengo, and Samuel Siqueira Bueno. Sliding mode control for the path following of an unmanned airship. *IFAC Proceedings Volumes*, 40(15):221–226, 2007.

[40] E. F. Camacho and C. Bordons. *Model Predictive Control*. Advanced Textbooks in Control and Signal Processing. Springer London, London, 2007.

[41] James B. Rawlings, David Q. Mayne, and Moritz M. Diehl. *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, Madison, Wisconsin, 2nd edition edition, 2017.

[42] Saša V. Raković and William S. Levine, editors. *Handbook of Model Predictive Control*. Control Engineering. Springer International Publishing, Cham, 2019.

[43] Jiashi Zhang, Xixiang Yang, Xiaolong Deng, and Huijing Lin. Trajectory control method of stratospheric airships based on model predictive control in wind field. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 233(2):418–425, February 2019.

[44] Petros Ioannou and Jing Sun. *Robust Adaptive Control*. Prentice-Hall, Inc, 1995.

[45] Ding. *Nonlinear and Adaptive Control Systems*. Institution of Engineering and Technology, April 2013.

[46] Ranjan Vepa. *Nonlinear Control of Robots and Unmanned Aerial Vehicles: An Integrated Approach*. CRC Press, Boca Raton : Taylor & Francis, a CRC title, part of the Taylor &, zeroth edition, October 2016.

[47] Pedro A. Capó-Lugo and Peter M. Bainum. Continuous and digital control systems. In *Orbital Mechanics and Formation Flying*, pages 155–246. Elsevier, 2011.

[48] Chang Xiao, Yueying Wang, Pingfang Zhou, and Dengping Duan. Adaptive sliding mode stabilization and positioning control for a multi-vectored thrust airship with input saturation considered. *Transactions of the Institute of Measurement and Control*, 40(15):4208–4219, November 2018.

[49] Kuo-Chi Chang, Kai-Chun Chu, Yuh-Chung Lin, and Jeng-Shyang Pan. Overview of Some Intelligent Control Structures and Dedicated Algorithms. In *Automation and Control*. IntechOpen, April 2020.

[50] Yueneng Yang, Jie Wu, and Wei Zheng. Station-keeping control for a stratospheric airship platform via fuzzy adaptive backstepping approach. *Advances in Space Research*, 51(7):1157–1167, April 2013.

[51] Zewei Zheng and Yao Zou. Adaptive integral LOS path following for an unmanned airship with uncertainties based on robust RBFNN backstepping. *ISA Transactions*, 65:210–219, November 2016.

[52] Ivana Palunko and Rafael Fierro. Adaptive Control of a Quadrotor with Dynamic Changes in the Center of Gravity. *IFAC Proceedings Volumes*, 44(1):2626–2631, January 2011.

[53] Ramiro Ibarra Pérez, Gerardo Romero Galvan, Aldo Jonathan Muñoz Vázquez, Silvia Florida Melo, and David Lara Alabazares. Attitude Control of a Quadcopter Using Adaptive Control Technique. In Le Anh Tuan, editor, *Adaptive Robust Control Systems*. InTech, March 2018.

[54] Yildiray Yildiz, Mustafa Unel, and Ahmet Eren Demirel. Nonlinear hierarchical control of a quad tilt-wing UAV: An adaptive control approach. *International Journal of Adaptive Control and Signal Processing*, 31(9):1245–1264, September 2017.

[55] Aytac Altan, Ozgur Aslan, and Rifat Hacioglu. Model Reference Adaptive Control of Load Transporting System on Unmanned Aerial Vehicle. In *2018 6th International Conference on Control Engineering & Information Technology (CEIT)*, pages 1–5, Istanbul, Turkey, October 2018. IEEE.

[56] J. U. Alvarez-Munoz, N. Marchand, J. F. Guerrero-Castellanos, A. E. Lopez-Luna, J. J. Tellez-Guzman, J. Colmenares-Vazquez, S. Durand, J. Dumon, and G. Hasan. Nonlinear control of a nano-hexacopter carrying a manipulator arm. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4016–4021, Hamburg, Germany, September 2015. IEEE.

[57] Marcelino M. de Almeida and Guilherme V. Raffo. Nonlinear Balance Control of an Inverted Pendulum on a Tilt-rotor UAV. *IFAC-PapersOnLine*, 48(9):168–173, 2015.

[58] Gaetano Tartaglione, Egidio D'Amato, Marco Ariola, Pierluigi Salvo Rossi, and Tor Arne Johansen. Model predictive control for a multi-body slung-load system. *Robotics and Autonomous Systems*, 92:1–11, June 2017.

[59] Viacheslav Pshikhopov, Mikhail Medvedev, Vladimir Kostjukov, Roman Fedorenko, Boris Gurenko, and Victor Krukhmalev. Airship Autopilot Design. In *Aerospace Technology Conference and Exposition*, pages 2011–01–2736, October 2011.

[60] V. Kh. Pshikhopov and M. Yu. Medvedev. Robust control of nonlinear dynamic systems. In *2010 IEEE ANDESCON*, pages 1–7, Bogota, September 2010. IEEE.

[61] NUMECA. NUMECA International - CFD Simulation Software. https://www.numeca.com/home, 2020.

[62] Torsten Liesk, Meyer Nahon, and Benoit Boulet. Design and Experimental Validation of a Nonlinear Low-Level Controller for an Unmanned Fin-Less Airship. *IEEE Transactions on Control Systems Technology*, 21(1):149–161, January 2013.

[63] Shiqian Liu, Yuanjun Sang, and Hongbin Jin. Robust model predictive control for stratospheric airships using LPV design. *Control Engineering Practice*, 81:231–243, December 2018.

[64] Yue Wang, Gang Zheng, Denis Efimov, and Wilfrid Perruquetti. Altitude Control for an Indoor Blimp Robot. *IFAC-PapersOnLine*, 50(1):15990–15995, July 2017.

[65] Yue Wang, Gang Zheng, Denis Efimov, and Wilfrid Perruquetti. Disturbance Compensation Based Control for an Indoor Blimp Robot. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2040–2046, May 2019.

[66] Yu Liu, Yachen Zhang, and Yueming Hu. Optimal Path Planning for Autonomous Airship Based on Clonal Selection and Direct Collocation Algorithm. In *2008 IEEE International Conference on Networking, Sensing and Control*, pages 1828–1832, April 2008.

[67] G. A. Khoury, editor. *Airship Technology*. Number 10 in Cambridge Aerospace Series. Cambridge University Press, Cambridge ; New York, 2nd ed edition, 2012.

[68] Legislative Services Branch. Consolidated federal laws of canada, Canadian Aviation Regulations. https://laws-lois.justice.gc.ca/eng/regulations/SOR-96-433/page-177.html#docCont, January 2021.

[69] Joseph B Mueller, Michael A Paluszek, and Yiyuan Zhao. Development of an Aerodynamic Model and Control Law Design for a High Altitude Airship. In *AIAA 3rd "Unmanned Unlimited" Technical Conference, Workshop and Exhibit*, page 17, 2004.

[70] Ira H. Abbott and Albert E. VonDoenhoff. *Theory of Wing Sections: Including a Summery of Airfoil Data*. Dover Publ, New York, NY, 2010.

[71] Praveen Gill, Sandeep Malik, and Rajkumar S Pant. Estimation of aerodynamic characteristics of un-symmetrically finned bodies of revolutions. Technical report, Indian Institute of Technology Bombay, 2001.

[72] S P Jones and J D Delaurier. Aerodynamic estimation techniques for aerostats and airships. *J. AIRCRAFT*, page 7, 1983.

[73] Wei-Der Chang and Jun-Juh Yan. Adaptive robust PID controller design based on a sliding mode for uncertain chaotic systems. *Chaos, Solitons & Fractals*, 26(1):167–175, October 2005.

[74] Teuku Mohd Ichwanul Hakim and Ony Arifianto. Implementation of Dryden Continuous Turbulence Model into Simulink for LSA-02 Flight Test Simulation. *Journal of Physics: Conference Series*, 1005:012017, April 2018.

[75] Charlotte Treffers and Luc van Wietmarschen. Position and orientation determination of a probe with use of the IMU MPU9250 and a ATmega328 microcontroller. Technical report, TU Delft, 2016.

[76] International Civil Aviation Organization and Langley Aeronautical Laboratory. Standard Atmosphere-Tables and Data for Altitudes to 65,800 Feet. Technical Report 1235, Montreal, CA and Hampton,US.

[77] Environment and Climate Change Canada. Daily Data Report for December 2020 - Climate - Environment and Climate Change Canada. https://climate.weather.gc.ca/climate_data/daily_data_e.html?StationID=49568&timeframe=2&StartYear=1840&EndYear=2020&Day=16&Year=2020&Month=12#, October 2011.

APPENDICES

Appendix A

Kinematics Coefficients

The resulting expressions are given by:

$$a_{r1x,1} = x_g \cos(\phi)\cos(\theta) - d_{m1x}\cos(\psi)\cos(\theta) - d_{m1z}\sin(\phi)\sin(\psi) \tag{A.1}$$

$$a_{r1x,2} = z_g \sin(\phi)\sin(\psi) - d_{m1z}\cos(\phi)\cos(\psi)\sin(\theta) + z_g \cos(\phi)\cos(\psi)\sin(\theta) \tag{A.2}$$

$$a_{r1y,1} = x_g \cos(\theta)\sin(\phi) + d_{m1z}\cos(\psi)sin(\phi) - d_{m1x}\cos(\theta)\sin(\psi) \tag{A.3}$$

$$a_{r1y,2} = -z_g \cos(\psi)\sin(\phi) - d_{m1z}\cos(\phi)\sin(\psi)\sin(\theta) + z_g \cos(\phi)\sin(\psi)\sin(\theta) \tag{A.4}$$

$$a_{r1z,1} = d_{m1x}\sin(\theta) - x_g \sin(\theta) \tag{A.5}$$

$$a_{r1z,2} = -d_{m1z}\cos(\phi)\cos(\theta) + z_g \cos(\phi)\cos(\theta) \tag{A.6}$$

$$a_{r2x,1} = d_{m1x}\cos(\psi)\cos(\theta) + d_{m1z}\sin(\phi)\sin(\psi) - z_g\sin(\phi)\sin(\psi) \tag{A.7}$$

$$a_{r2x,2} = -x_g\cos(\phi)\cos(\theta) + d_{m1z}\cos(\phi)\cos(\psi)\sin(\theta) - z_g\cos(\phi)\cos(\psi)\sin(\theta) \tag{A.8}$$

$$a_{r2y,1} = d_{m1z}\cos(\psi)\sin(\phi) - d_{m1x}\cos(\theta)\sin(\psi) - z_g\cos(\psi)sin(\phi) \tag{A.9}$$

$$a_{r2y,2} = x_g\cos(\theta)\sin(\phi) - d_{m1z}\cos(\phi)\sin(\psi)\sin(\theta) + z_g\cos(\phi)\sin(\psi)\sin(\theta) \tag{A.10}$$

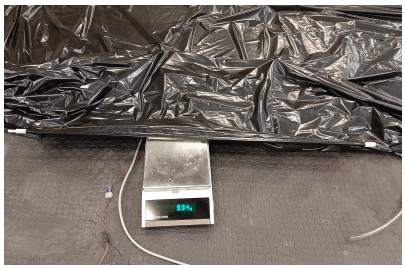
$$a_{r2z,1} = d_{m1x}\sin(\theta) - x_g\sin(\theta) \tag{A.11}$$

$$a_{r2z,2} = -d_{m1z}\cos(\phi)\cos(\theta) + z_g\cos(\phi)\cos(\theta) \tag{A.12}$$

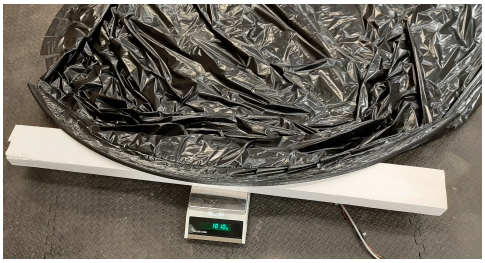
Appendix B

Body 1 Mass Estimation

Figure B.1 shows the components of Body 1, whose mass was measured using a scale.



(a) Mass measurement of the straight keel section

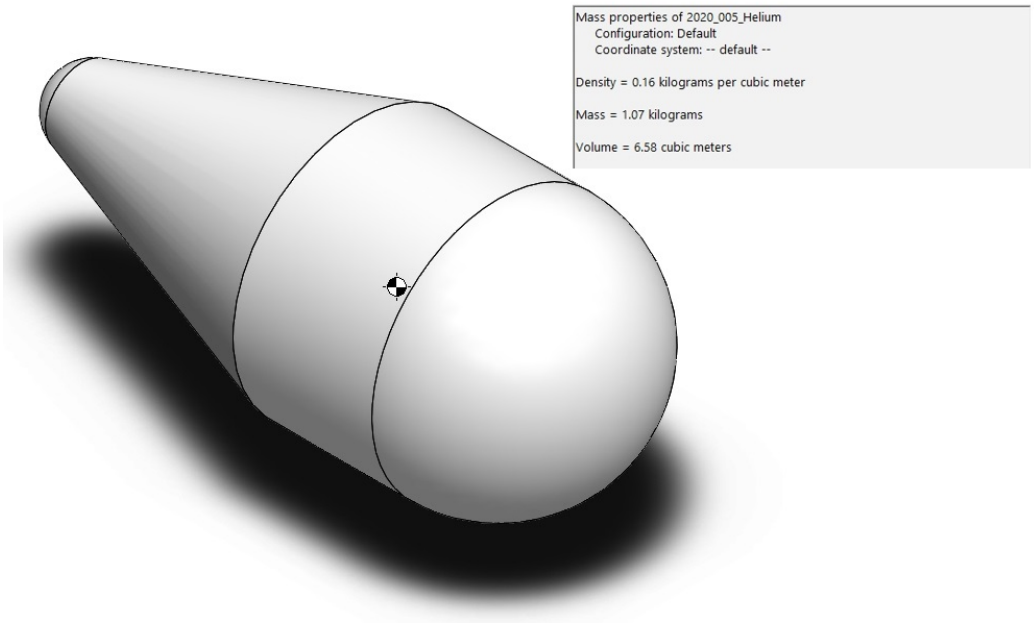


(b) Mass measurement of the curve keel section

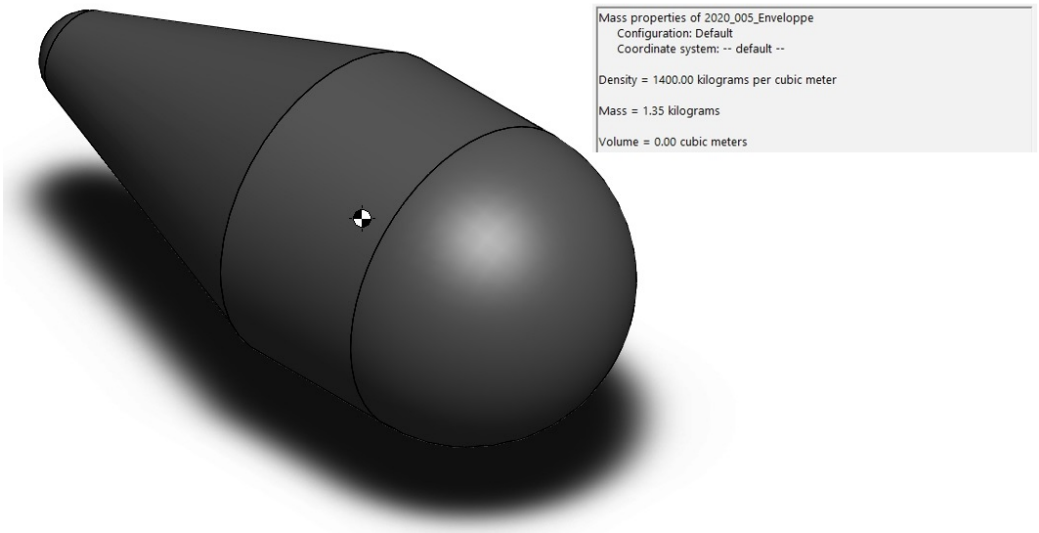
(c) Fin mass measurement

Figure B.1: Measured mass for the weight estimation of Body 1

Figure B.2 shows the components of Body 1, whose mass was estimated by CAD tools using SolidWorks.



(a) Helium SolidWorks mass



(b) Envelope SolidWorks mass

Figure B.2: Mass estimated by SolidWorks

Appendix C

Centre of Mass of Body 1 and Inertia

The center mass of Body 1 for the x axis was estimated using the following expression:

$$d_{m1,x} = \frac{(m_{envelope}x_{envelope}) + (m_{straight}x_{straight}) + \frac{(m_{curve}x_{curve}) + (m_{vfin}x_{vfin}) + (2m_{ifin}x_{ifin})}{m_{curve} + m_{vfin} + 2m_{ifin}}\Big]}{m_{envelope} + m_{straight}} \quad \text{(C.1)}$$

where the envelope mass, straight rail mass, curve rail mass, vertical fin mass, and inclined fin mass are represented by $m_{envelope}$, $m_{straight}$, m_{curve}, m_{vfin}, and m_{ifin}, respectively. Moreover, the distance along the x axis, from the CV to the CG of the envelope, straight rail, curve rail, vertical fin, and inclined fins are represented by $x_{envelope}$, $x_{straight}$, x_{curve}, x_{vfin}, and x_{ifin}, respectively. A similar procedure was carried out for $d_{m1,z}$.

Remark 2

The mass parameters are shown in Appendix B. All of the fins are assumed to be equal (same mass and dimensions).

Remark 3

Due to longitudinal symmetry, the system can be simplified to the xz plane for the CG coordinates estimation. Therefore, the straight rail was considered as a rectangle.

Similarly, the curve rail was assumed as a quarter ring. Finally, the fins were assumed as triangles.

.

Remark 4

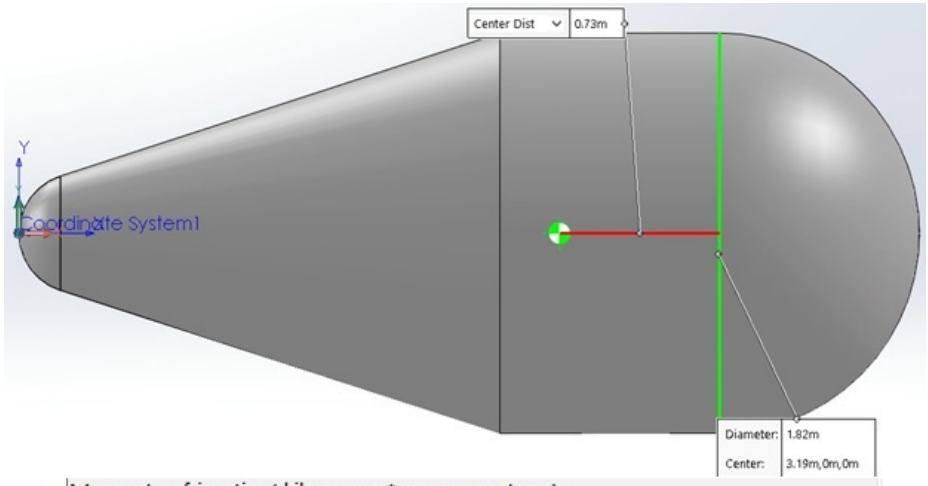
The distance $x_{envelope}$ was computed based on the information shown in Figure C.1. The other distance was estimated using the CAD information and measured values.

The rail and the fin inertias were neglected as they are too small in relation to the envelope and the helium inertia. The helium and envelope inertias were estimated using SolidWorks and they are shown in Figure C.1.

For the inertia estimation, the Steiner's Theorem is needed to compute the inertia for the CG axis of Body 1. Therefore, the helium and envelope inertia were moved to determine the inertia for Body 1. For instance, for Body 1, for the inertia of the x axis $I_{b1,x}$, the procedure is as follows:

$$I_{b1,x} = (I_{helium,x} + m_{helium} d^2_{helium,x}) + (I_{envelope,x} + m_{envelope} d^2_{envelope,x}) \tag{C.2}$$

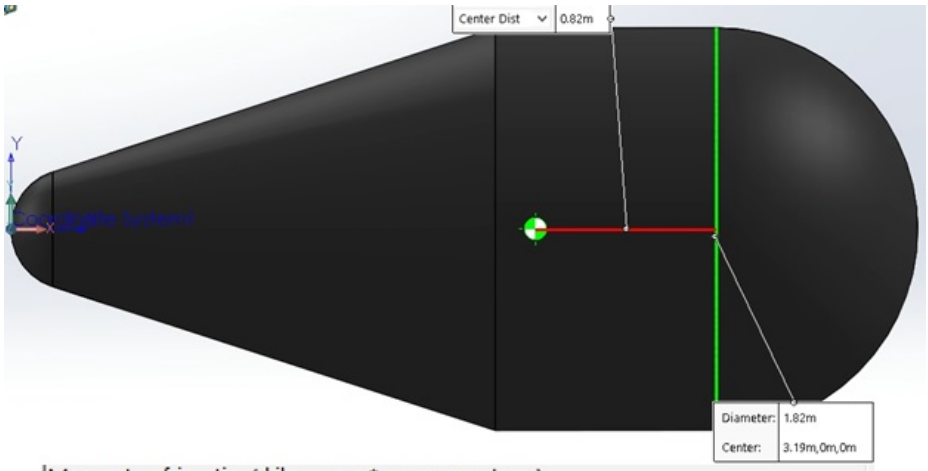
where the helium inertia, the envelope inertia, the perpendicular distance between the helium CG x axis and the Body 1 CG x axis, the perpendicular distance between the envelope CG x axis and the Body 1 CG x axis, and the helium mass are represented by $I_{helium,x}$, $I_{envelope,x}$, $d_{helium,x}$, $d_{envelope,x}$, and m_{helium}, respectively.



Moments of inertia: (kilograms * square meters)
Taken at the center of mass and aligned with the output coordinate system.

Lxx = 0.36	Lxy = 0.00	Lxz = 0.00
Lyx = 0.00	Lyy = 1.00	Lyz = 0.00
Lzx = 0.00	Lzy = 0.00	Lzz = 1.00

(a) Helium SolidWorks inertia



Moments of inertia: (kilograms * square meters)
Taken at the center of mass and aligned with the output coordinate system.

Lxx = 0.79	Lxy = 0.00	Lxz = 0.00
Lyx = 0.00	Lyy = 1.93	Lyz = 0.00
Lzx = 0.00	Lzy = 0.00	Lzz = 1.93

(b) Envelope SolidWorks inertia

Figure C.1: Inertias estimated by SolidWorks

Appendix D

Elevation vs Air Density

The value of $\Delta\delta$ was estimated based on air density against altitude data provided in [76]. Figure D.1 shows the air density for different elevations at 15 °C.

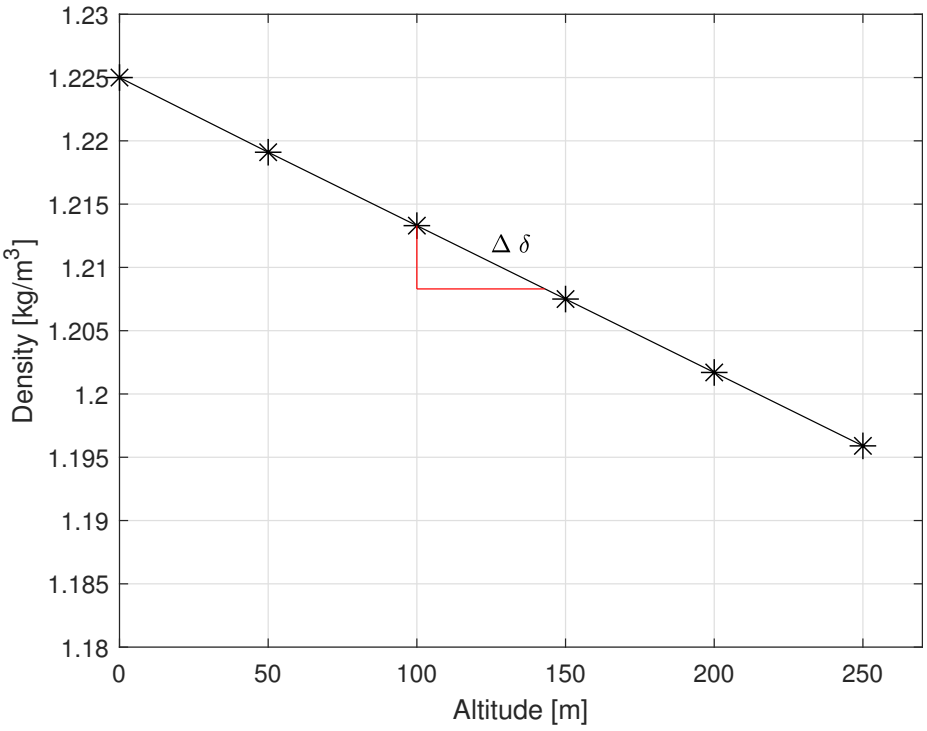


Figure D.1: Elevation vs air density

Based on the city of Ottawa's elevation (114.9 m) [77], and the maximum operating

altitude allowed for a drone in Canada (122 m) [68], the altitude of interest for our work is at approximately 236.9 m. Taking into consideration some margin for a higher take-off point, the elevation 250 m will be approximated as the maximum operating altitude. Therefore, the slope of Figure D.1 will be equal to $\Delta\delta$.

Finally, computing the slope between 0 and an elevation of 250 m, $\Delta\delta = -1.164 \times 10^{-4} kg/m^4$.

Appendix E

Operation and Safety

The prototype operation modes and safety equipment are presented in Section E.1 and Section E.2, respectively.

E.1 Operation Modes

The system has two operation modes: manual and automatic.

- Manual mode: In this mode the system is operated manually using the a radio control. It is possible to move the gondola forward and backward. In addition, it is possible to operate the propellers.
- Automatic mode: In this mode the system is operated using the user interface presented in Section 5.3. The automatic mode allows autonomous operation for the pitch control angle.

E.2 Safety

The system is equipped with two limit switches in order to prevent the gondola from going out of the keel. Figure E.1 shows the assembly of one of the limit switches.

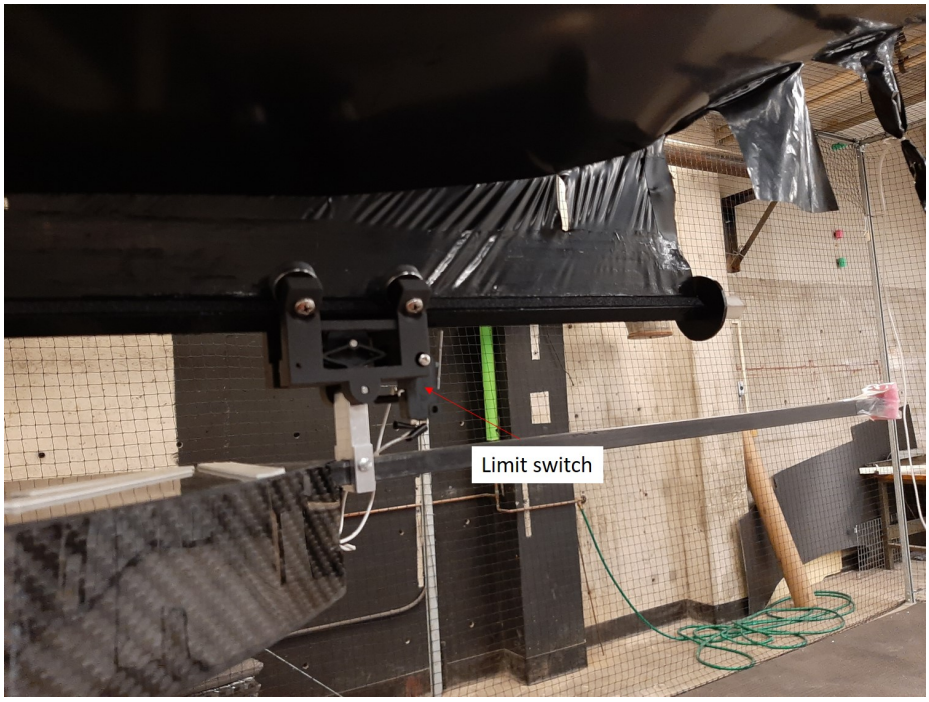


Figure E.1: Gondola limit switch

Appendix F

Flight Tests

This section presents additional flight tests that were carried out, when the controller gains were properly tuned.

F.1 Step

F.1.1 Test 1

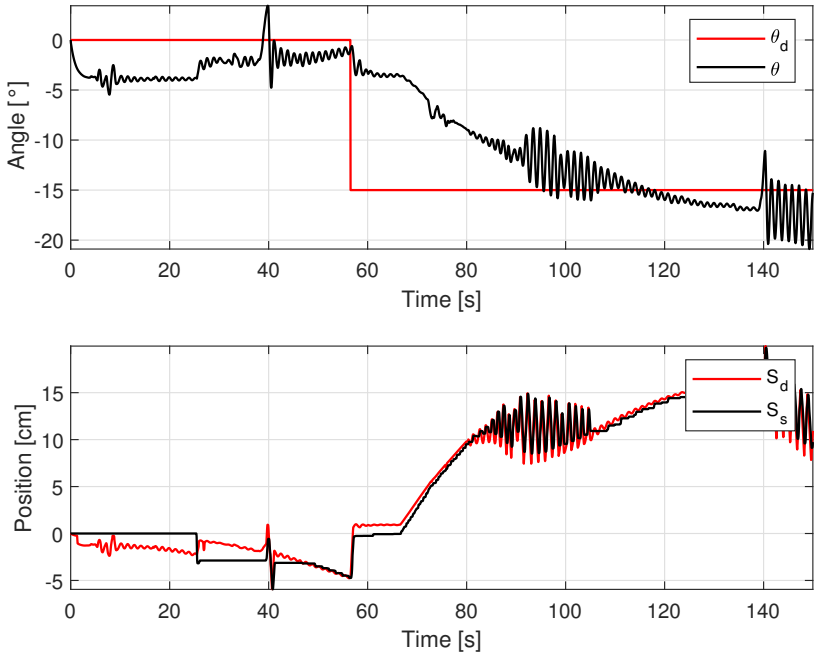


Figure F.1: Pitch angle and gondola travelled distance for $\gamma_1 = 2.5$, $\gamma_2 = 0.01$ and $\gamma_3 = 0.1$

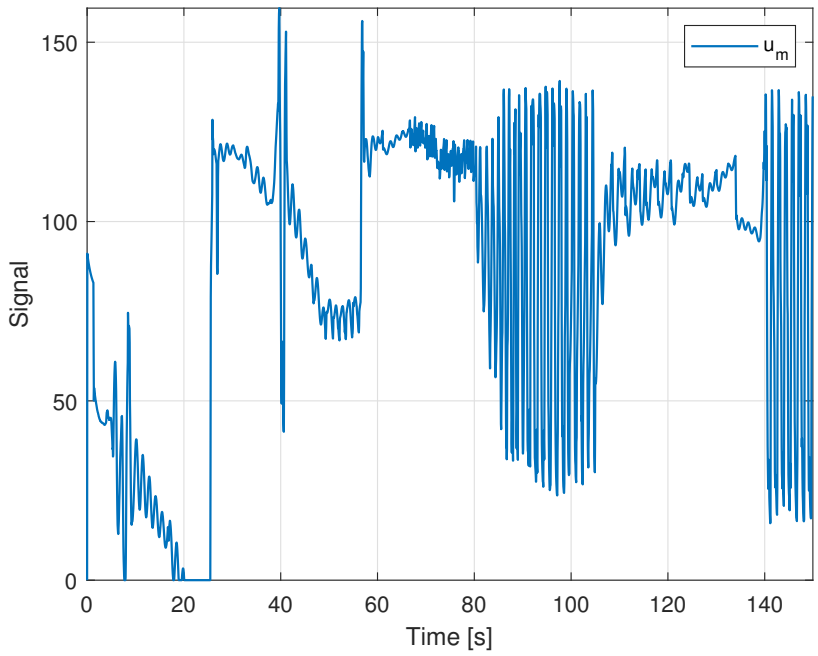


Figure F.2: Gondola motor signal for $\gamma_1 = 2.5$, $\gamma_2 = 0.01$ and $\gamma_3 = 0.1$

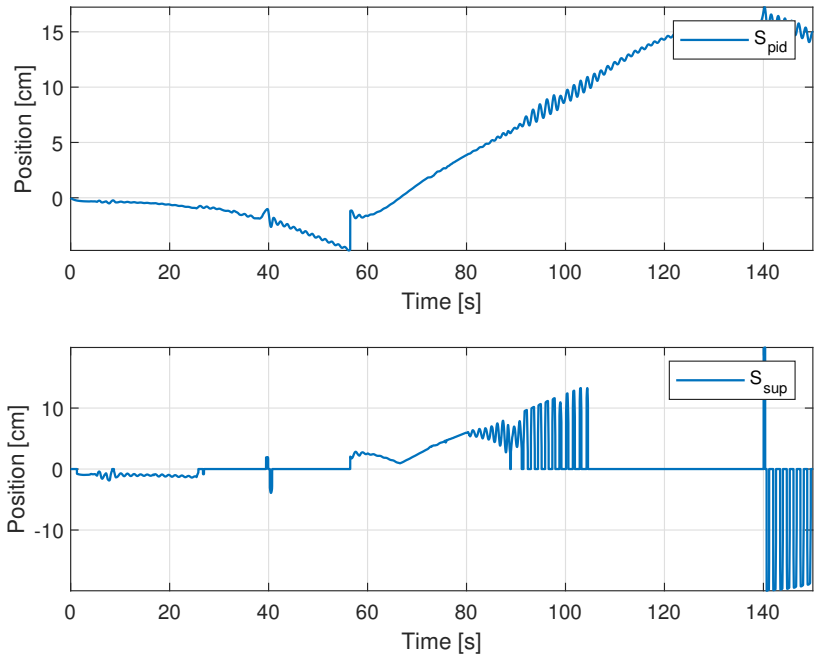


Figure F.3: Control action of adaptive PID and supervisory controller for $\gamma_1 = 2.5$, $\gamma_2 = 0.01$ and $\gamma_3 = 0.1$

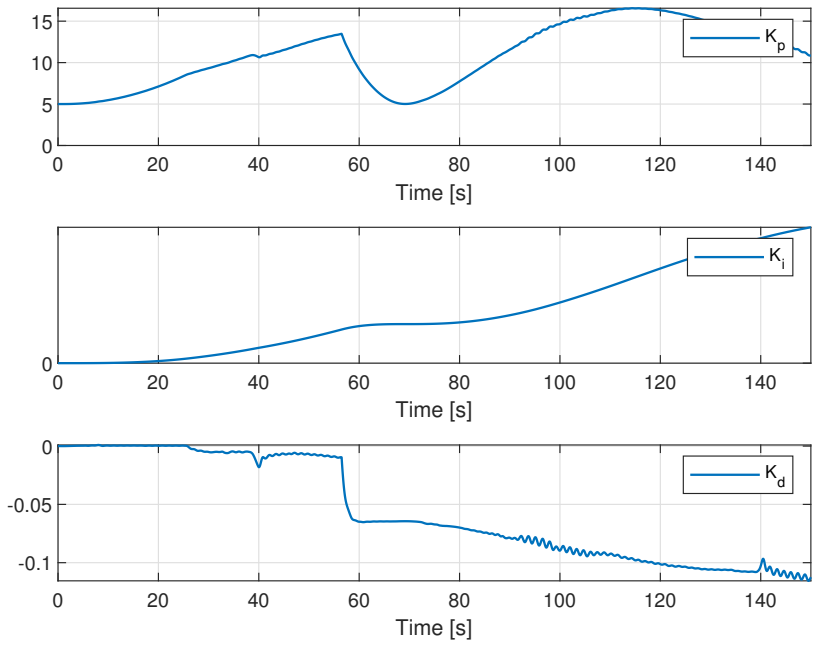


Figure F.4: Adaptive PID gains for $\gamma_1 = 2.5$, $\gamma_2 = 0.01$ and $\gamma_3 = 0.1$

F.1.2 Test 2

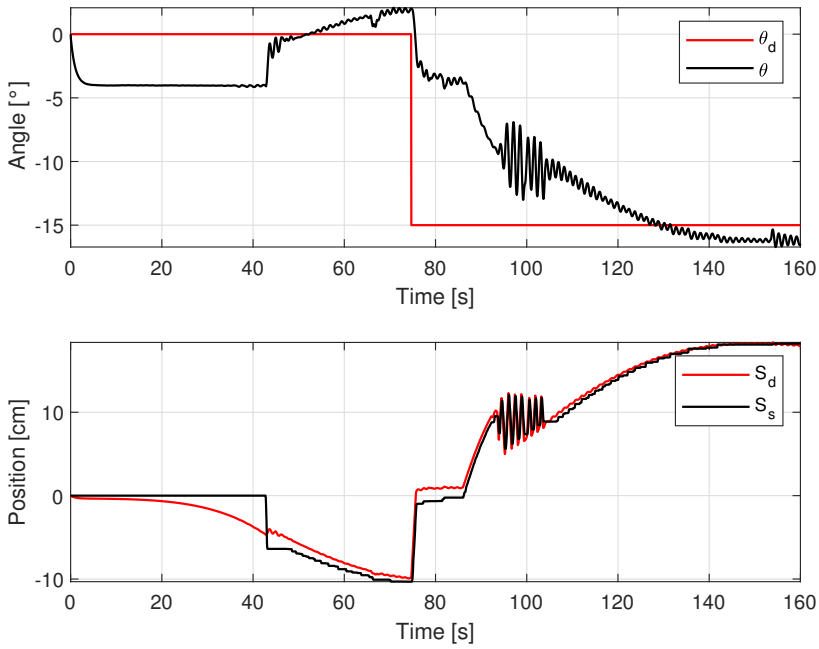


Figure F.5: Pitch angle and gondola travelled distanc $\gamma_1 = 2.5$, $\gamma_2 = 0.01$ and $\gamma_3 = 0.1$

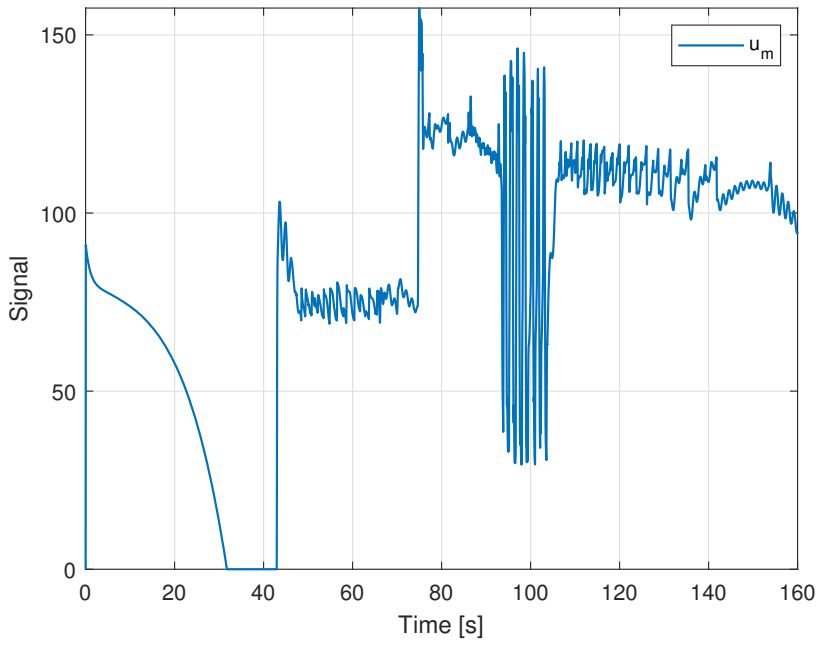


Figure F.6: Gondola motor signal for $\gamma_1 = 2.5$, $\gamma_2 = 0.01$ and $\gamma_3 = 0.1$

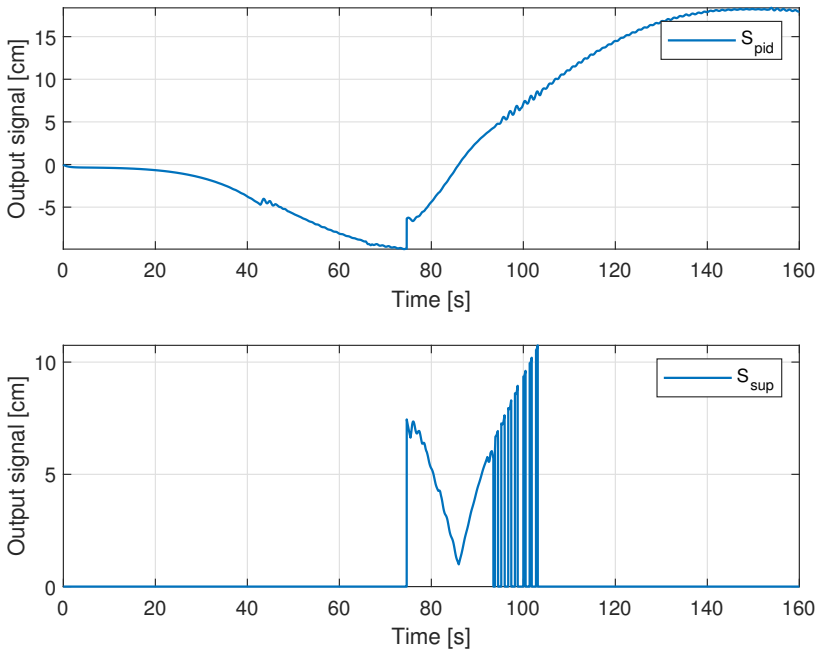


Figure F.7: Control action of adaptive PID and supervisory controller for $\gamma_1 = 2.5$, $\gamma_2 = 0.01$ and $\gamma_3 = 0.1$

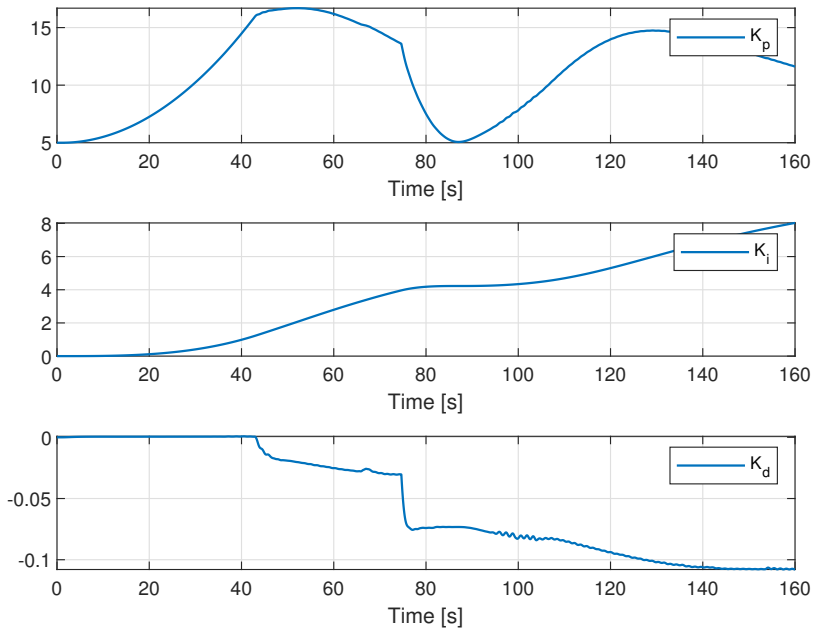


Figure F.8: Adaptive PID gains for $\gamma_1 = 2.5$, $\gamma_2 = 0.01$ and $\gamma_3 = 0.1$

F.2 Trajectory

F.2.1 Test 1

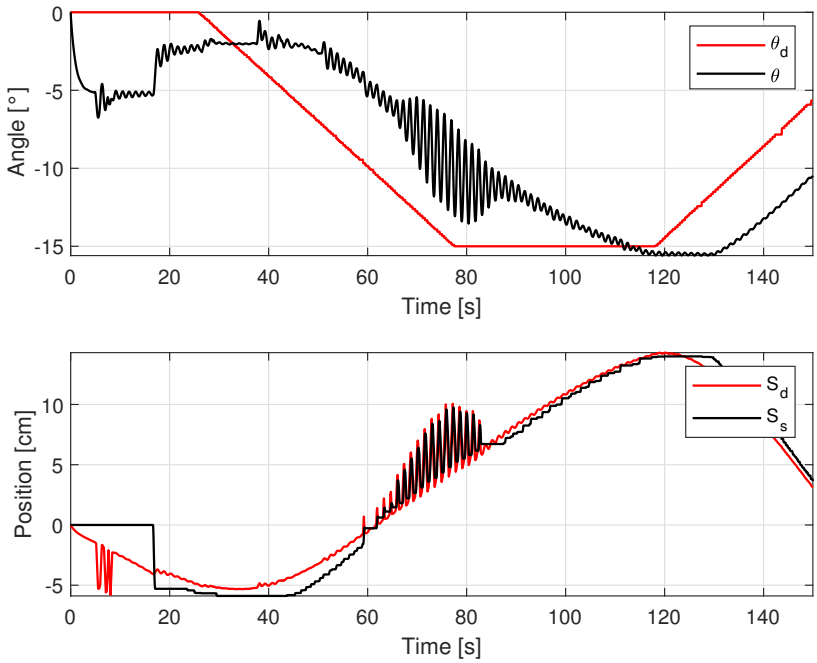


Figure F.9: Pitch angle and gondola travelled distance for $\gamma_1 = 2.5$, $\gamma_2 = 0.01$ and $\gamma_3 = 0.1$

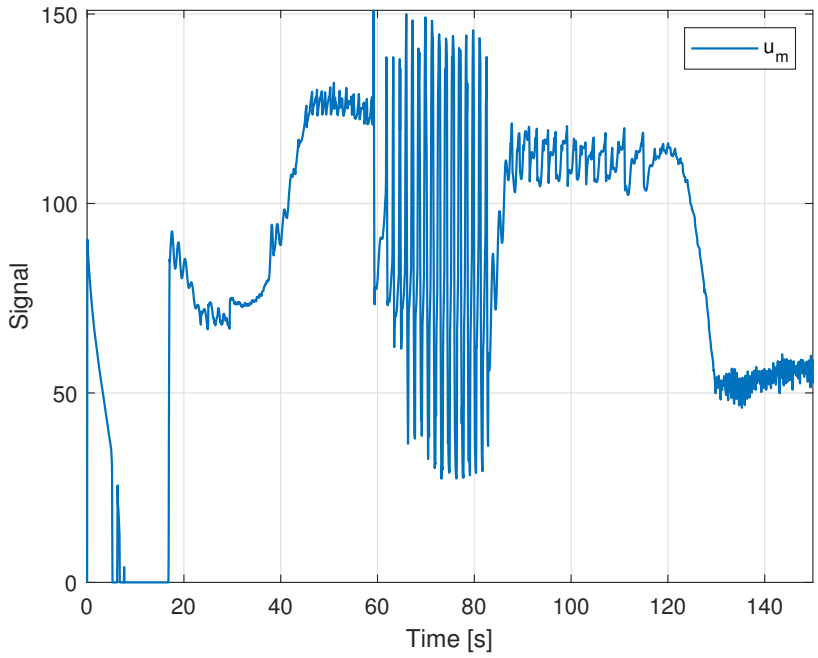


Figure F.10: Gondola motor signal for $\gamma_1 = 2.5$, $\gamma_2 = 0.01$ and $\gamma_3 = 0.1$

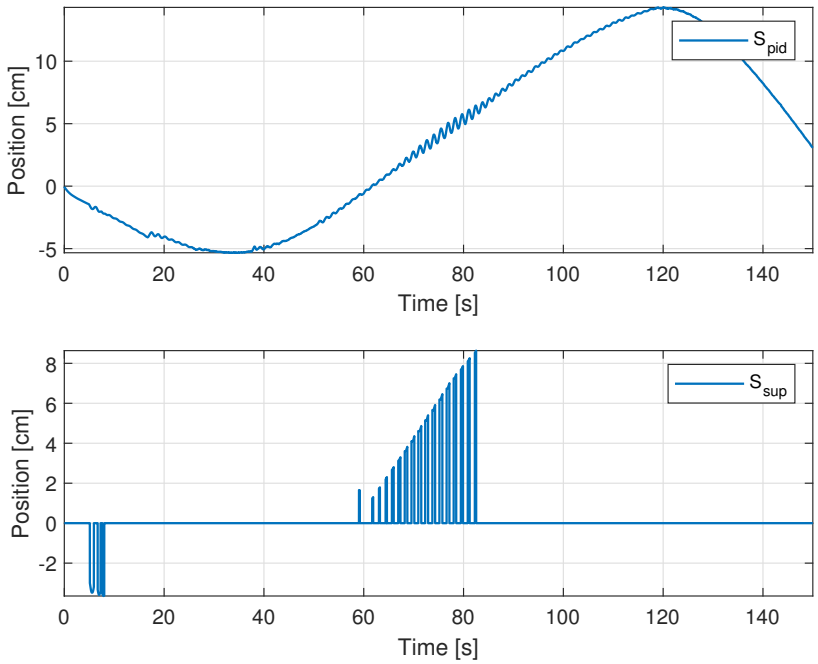


Figure F.11: Control action of adaptive PID and supervisory controller for $\gamma_1 = 2.5$, $\gamma_2 = 0.01$ and $\gamma_3 = 0.1$

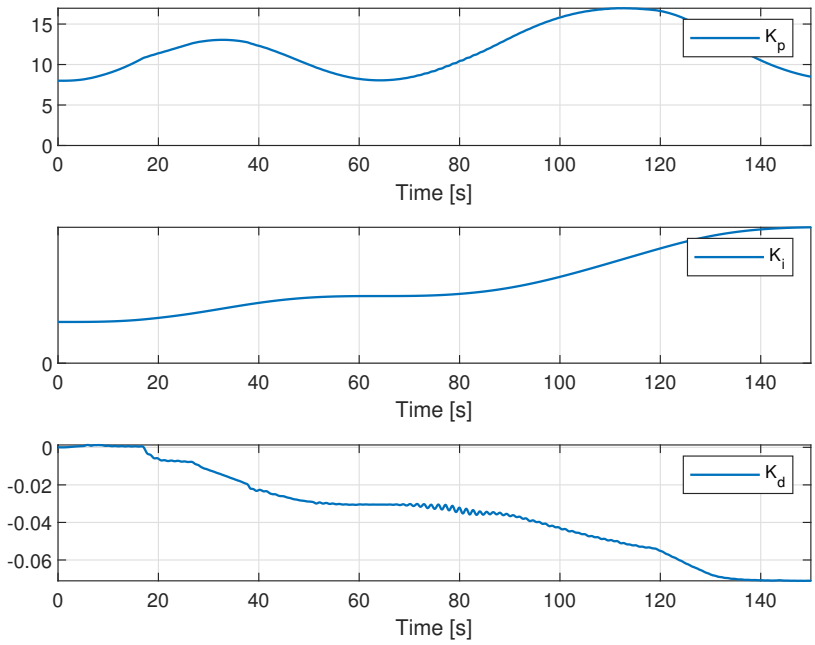


Figure F.12: Adaptive PID gains for $\gamma_1 = 2.5$, $\gamma_2 = 0.01$ and $\gamma_3 = 0.1$

F.2.2 Test 2

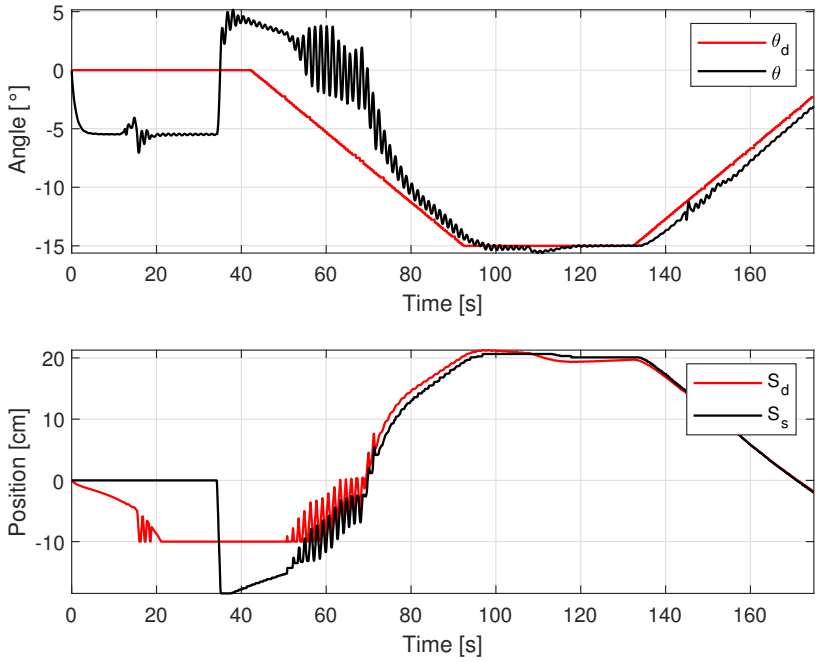


Figure F.13: Pitch angle and gondola travelled distance for $\gamma_1 = 4$, $\gamma_2 = 0.1$ and $\gamma_3 = 0.1$

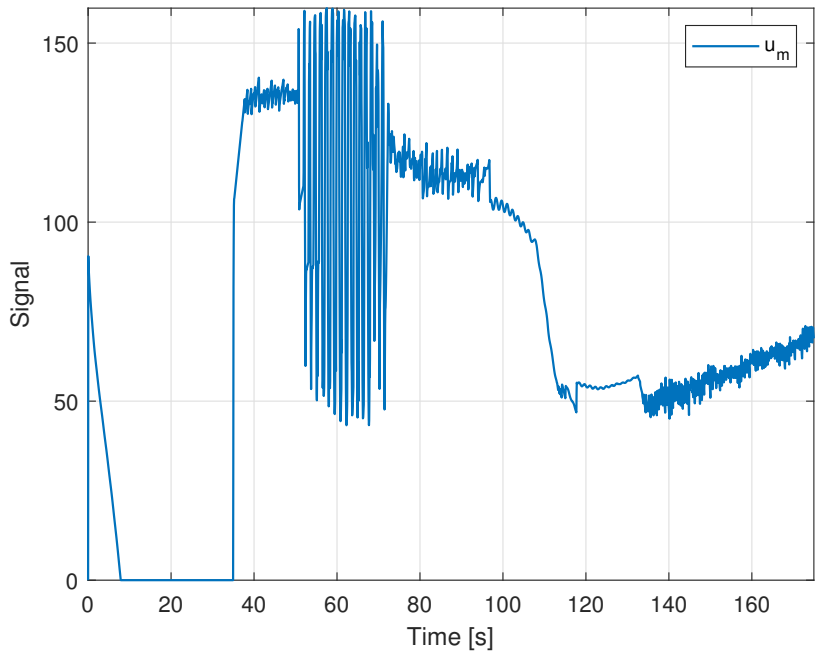


Figure F.14: Gondola motor signal for $\gamma_1 = 4$, $\gamma_2 = 0.1$ and $\gamma_3 = 0.1$

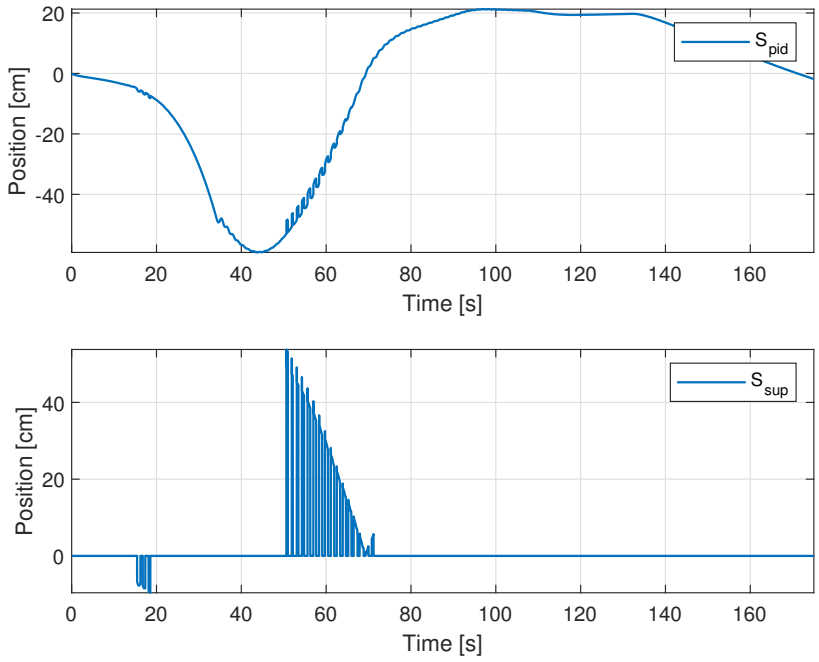


Figure F.15: Control action of adaptive PID and supervisory controller for $\gamma_1 = 4$, $\gamma_2 = 0.1$ and $\gamma_3 = 0.1$

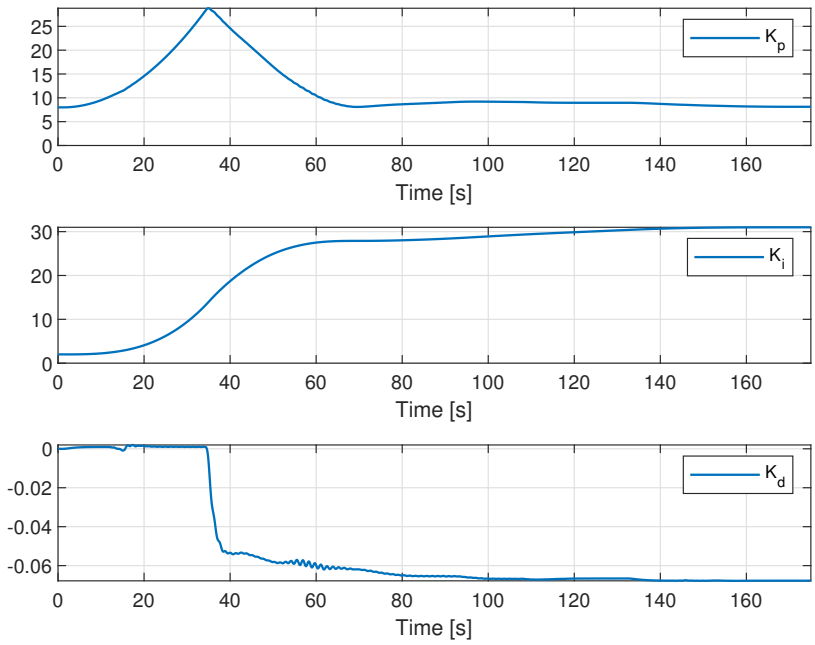


Figure F.16: Adaptive PID gains for $\gamma_1 = 4$, $\gamma_2 = 0.1$ and $\gamma_3 = 0.1$

F.2.3 Test 3

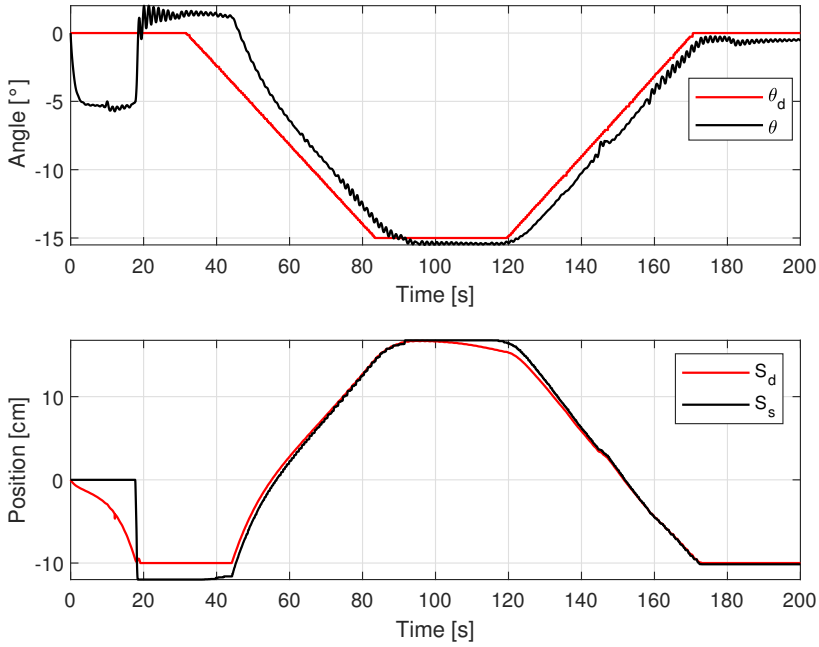


Figure F.17: Pitch angle and gondola travelled distance for $\gamma_1 = 4$, $\gamma_2 = 0.25$ and $\gamma_3 = 0.1$

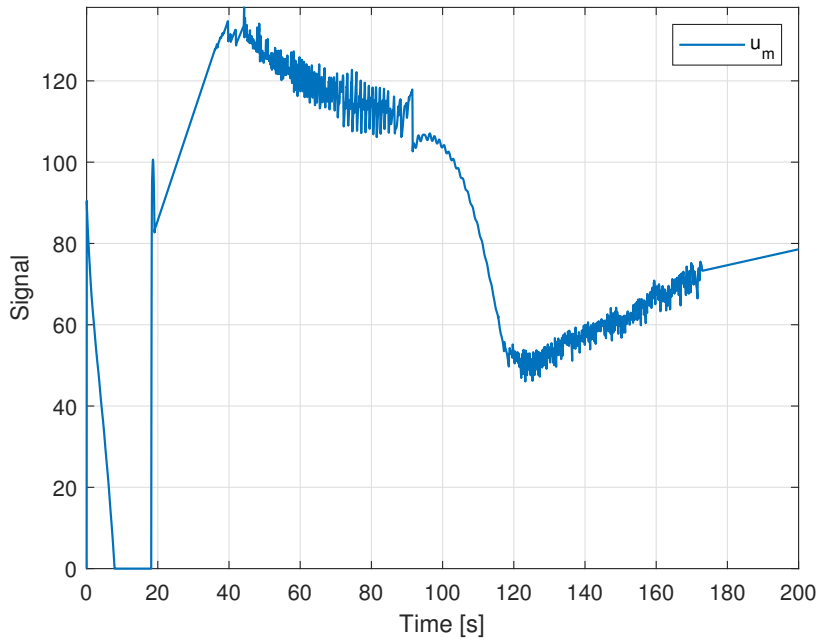


Figure F.18: Gondola motor signal for $\gamma_1 = 4$, $\gamma_2 = 0.25$ and $\gamma_3 = 0.1$

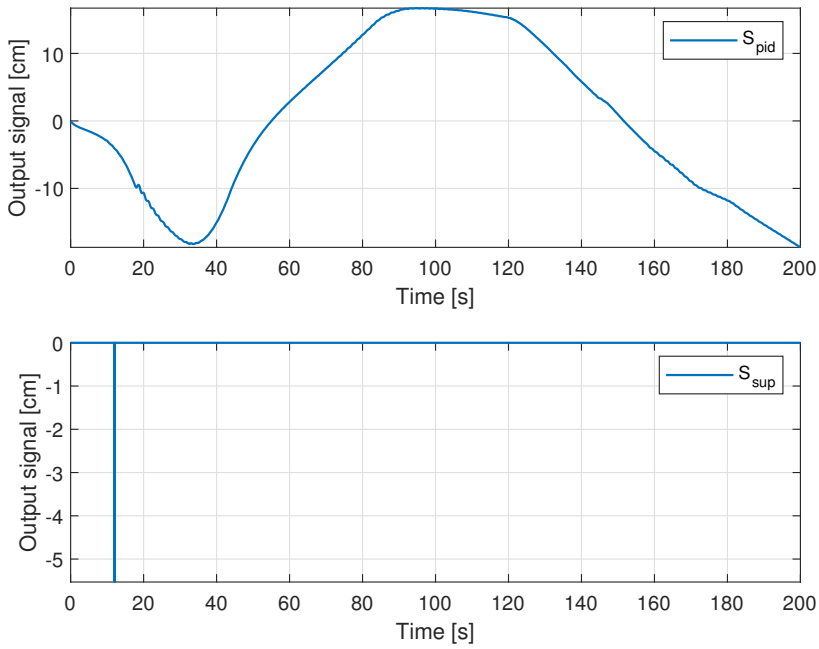


Figure F.19: Control action of adaptive PID and supervisory controller for $\gamma_1 = 4$, $\gamma_2 = 0.25$ and $\gamma_3 = 0.1$

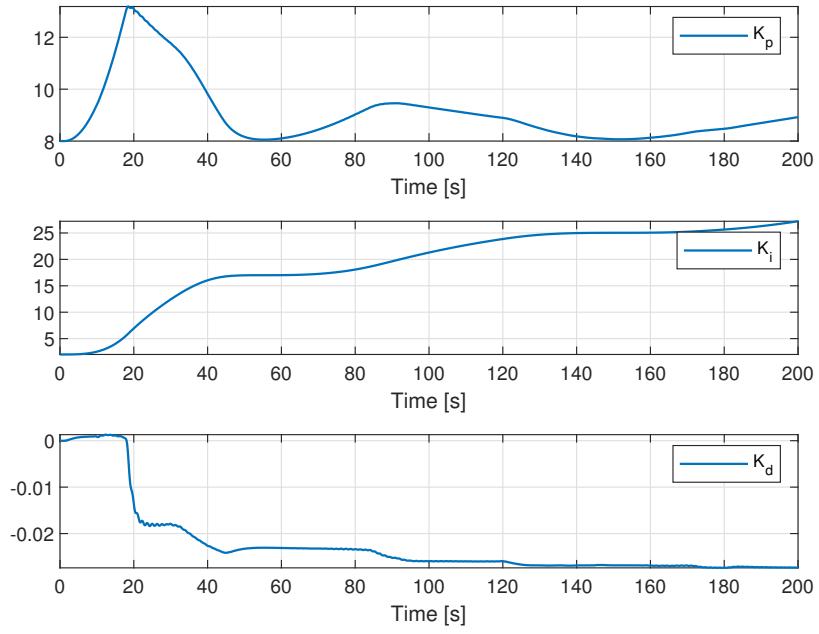


Figure F.20: Adaptive PID gains for $\gamma_1 = 4$, $\gamma_2 = 0.25$ and $\gamma_3 = 0.1$