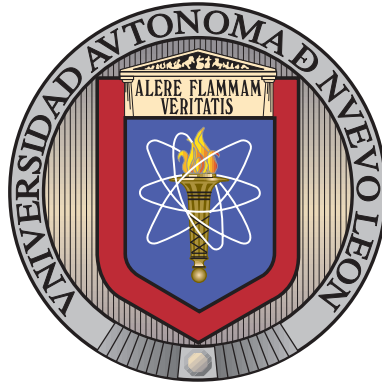


UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

SUBDIRECCIÓN DE ESTUDIOS DE POSGRADO



APPLICATION OF AN INTELLIGENT ALGORITHM  
IN THE SYSTEM IDENTIFICATION OF A  
QUADROTOR

BY

ALEJANDRO JIMÉNEZ FLORES

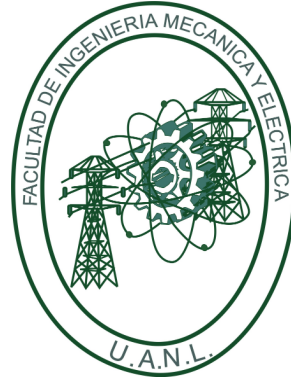
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF  
MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING  
WITH ORIENTATION IN FLIGHT DYNAMICS

NOVEMBER 2024

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

SUBDIRECCIÓN DE ESTUDIOS DE POSGRADO



APPLICATION OF AN INTELLIGENT ALGORITHM  
IN THE SYSTEM IDENTIFICATION OF A  
QUADROTOR

BY

ALEJANDRO JIMÉNEZ FLORES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF  
MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING  
WITH ORIENTATION IN FLIGHT DYNAMICS

NOVEMBER 2024

**UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN**  
**Facultad de Ingeniería Mecánica y Eléctrica**  
**Posgrado**

Los miembros del Comité de Evaluación de Tesis recomendamos que la Tesis “Application of an Intelligent Algorithm in the System Identification of a Quadrotor”, realizada por el estudiante Alejandro Jiménez Flores, con número de matrícula 2082177, sea aceptada para su defensa como requisito parcial para obtener el grado de Maestría en Ingeniería Aeronáutica con Orientación en Dinámica de Vuelo.

**El Comité de Evaluación de Tesis**

Dr. Octavio García Salazar  
Director

Dr. Edmundo Javier Ollervides Vázquez  
Co-director

Dr. Luis Arturo Reyes Osorio  
Revisor

Dr. Luis Antonio Amézquita Brooks  
Revisor

Dr. Ricardo Chapa García  
Revisor

Vo.Bo.

  
Dr. Simón Martínez Martínez  
Subdirector de Estudios de Posgrado

Institución 190001

Programa 563510

Acta Núm. 4427

Ciudad Universitaria, a 15 de octubre del 2024

*A mis padres Rosa y Roberto  
por ser mi apoyo.*

*A mi hermano Roberto  
por ser mi ejemplo.*

*A mi novia Carolina  
por ser mi motivación.*

# CONTENTS

---

<b>List of Figures</b>	<b>ix</b>
<b>List of Tables</b>	<b>xv</b>
<b>Acknowledgements</b>	<b>xvii</b>
<b>Summary</b>	<b>xviii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 State of the art . . . . .	2
1.2.1 Forces and moments measurement . . . . .	2
1.2.2 System identification . . . . .	4
1.2.3 Artificial neural networks . . . . .	5
1.2.4 Previous work . . . . .	6
1.3 Justification . . . . .	7
1.4 Hypothesis . . . . .	7

---

1.5	General objective . . . . .	8
1.5.1	Specific objectives . . . . .	8
1.6	Results . . . . .	9
<b>2</b>	<b>Low-cost UAV testing bench</b>	<b>10</b>
2.1	Electronic devices . . . . .	11
2.2	Load cell working principle . . . . .	11
2.3	Design . . . . .	13
2.4	Mathematical model . . . . .	15
2.5	FEM simulations . . . . .	21
2.5.1	Parameters setting . . . . .	22
2.5.2	Simulations results . . . . .	24
2.6	Fabrication . . . . .	27
2.6.1	Electronics enclosure . . . . .	27
2.6.2	Al-6061 testing bench . . . . .	29
2.6.3	PLA testing bench . . . . .	30
2.6.4	PLA specimens . . . . .	31
2.7	Calibration and programming . . . . .	32
2.7.1	Calibration weights . . . . .	33
2.7.2	Load cells calibration . . . . .	34
2.7.3	Testing bench software . . . . .	38

---

2.8	Static and dynamic tests . . . . .	40
2.8.1	Static tests results . . . . .	41
2.8.2	Dynamic tests results . . . . .	43
<b>3</b>	<b>Quadrotor design and construction</b>	<b>51</b>
3.1	Avionics . . . . .	51
3.2	Design . . . . .	53
3.2.1	Conceptual design . . . . .	53
3.2.2	Preliminary design . . . . .	54
3.2.3	Detailed design . . . . .	55
3.3	Construction . . . . .	56
3.4	Operation and connectivity . . . . .	57
3.5	Experimental tests . . . . .	62
3.5.1	Force-torque sensor . . . . .	62
3.5.2	Experimental bench . . . . .	63
3.5.3	Quadrotor tests . . . . .	64
3.5.4	Tests results . . . . .	66
<b>4</b>	<b>Quadrotor system identification</b>	<b>71</b>
4.1	Mathematical model (white box approach) . . . . .	71
4.1.1	Equations of motion . . . . .	72
4.1.2	Parameter determination . . . . .	78

---

4.1.3	Model simulation . . . . .	81
4.2	Artificial neural networks . . . . .	85
4.2.1	Nodes and layers . . . . .	85
4.2.2	Learning methodology . . . . .	87
4.2.3	Generalized delta rule . . . . .	88
4.2.4	Training data processing . . . . .	91
4.2.5	Back-propagation algorithm . . . . .	92
4.3	Parameter estimation (grey box approach) . . . . .	94
4.3.1	Neural network for estimation of aerodynamic parameters . . . . .	94
4.3.2	Neural network for estimation of inertial parameters . . . . .	100
4.3.3	Estimated parameters and evaluation . . . . .	102
4.4	Data-driven identification (black box approach) . . . . .	103
4.4.1	Neural network for data prediction . . . . .	105
4.4.2	Data prediction evaluation . . . . .	106
<b>5</b>	<b>Conclusions</b>	<b>108</b>
<b>A</b>	<b>Appendix</b>	<b>110</b>
A.1	Neural networks data . . . . .	110
A.1.1	Thrust model . . . . .	110
A.1.2	Torque model . . . . .	111
A.1.3	Inertial model . . . . .	111



---

A.2 Quadrotor experiments data . . . . .	111
<b>References</b>	<b>113</b>

# LIST OF FIGURES

---

1.1	Different types and sizes of quadrotors [1], [2], [3] . . . . .	1
1.2	Thrust stand for medium and large UAVs [23] . . . . .	3
1.3	Difference between simulation and system identification [36] . . . . .	4
1.4	Comparison between a biological (left) and artificial (right) neuron [49]	5
2.1	Different types of load cells [57] . . . . .	11
2.2	Strain gauges in a single point load cell [58] . . . . .	12
2.3	Tension and compression zones in a load cell under an applied force [59] . . . . .	12
2.4	Low-cost UAV testing bench overall design . . . . .	13
2.5	Design process of the low-cost UAV testing bench . . . . .	14
2.6	Testing bench elements . . . . .	15
2.7	Designed pieces utilized in the testing bench . . . . .	15
2.8	Testing bench reference frames and notation of forces and distances .	17
2.9	Testing bench as cantilever beam approach with simplified loads for explanation . . . . .	18

---

2.10	Load cell force projection under different force inputs . . . . .	19
2.11	Testing bench fixing pieces with different materials . . . . .	21
2.12	ANSYS® Workbench static structural analysis . . . . .	22
2.13	Testing bench materials assignment for FEM simulations . . . . .	23
2.14	Testing bench mesh for FEM simulations . . . . .	23
2.15	Testing bench boundary conditions for FEM simulations . . . . .	24
2.16	Testing bench FEM total deformation . . . . .	25
2.17	Testing bench FEM equivalent stress . . . . .	25
2.18	Testing bench FEM simulations results . . . . .	26
2.19	Testing bench electronics connection diagram . . . . .	27
2.20	Testing bench PCB soldered . . . . .	28
2.21	Testing bench electronics enclosure . . . . .	28
2.22	All testing bench electronics connected . . . . .	29
2.23	Testing bench Al-6061 fixing pieces . . . . .	29
2.24	Al-6061 low-cost UAV testing bench . . . . .	30
2.25	Testing bench PLA fixing pieces . . . . .	30
2.26	PLA low-cost UAV testing bench . . . . .	31
2.27	Type IV tensile test specimens . . . . .	31
2.28	Type IV tensile test specimens under tension . . . . .	32
2.29	Stress-strain curves of PLA . . . . .	33

---

2.30	Testing bench calibration weights . . . . .	34
2.31	Balances used to measure the calibration weights . . . . .	35
2.32	Load cells calibration scenario . . . . .	35
2.33	Load cells calibration process using calibration weights . . . . .	36
2.34	Testing bench software: load cells calibration . . . . .	39
2.35	Testing bench operation flowchart . . . . .	40
2.36	Testing bench software: reading load cells DV data and waiting for stabilization . . . . .	41
2.37	Testing bench software: load cells DV data stabilized and taring pro- cess completed . . . . .	42
2.38	Testing bench software: reading forces $F_x$ , $F_y$ , and $F_z$ in UAV body frame $\{B\}$ . . . . .	43
2.39	Testing bench software: all read forces $F_x$ , $F_y$ , and $F_z$ in UAV body frame $\{B\}$ . . . . .	44
2.40	All exported forces $F_x$ , $F_y$ , and $F_z$ read in UAV body frame $\{B\}$ . . .	45
2.41	Static force tests with different platforms using calibration weights over $z$ axis . . . . .	45
2.42	Static force tests results over $z$ axis . . . . .	46
2.43	Testing bench static force tests using calibration weights over $x$ axis .	46
2.44	Testing bench static force tests using calibration weights over $y$ axis .	47
2.45	Static force tests results over all axis and correlation coefficients . . .	48
2.46	Dynamic force tests with different platforms using a DJI <sup>®</sup> Tello UAV	49

---

2.47	Dynamic force tests results over $z$ axis . . . . .	50
3.1	Quadrotor components . . . . .	52
3.2	Quadrotor conceptual design . . . . .	54
3.3	Quadrotor CAD components . . . . .	54
3.4	Quadrotor preliminary design phase . . . . .	55
3.5	Quadrotor detailed design . . . . .	56
3.6	Printing quadrotor frame . . . . .	56
3.7	Quadrotor frame . . . . .	57
3.8	The quadrotor . . . . .	58
3.9	Quadrotor dynamics . . . . .	59
3.10	Quadrotor dynamics according to their motors spin intensity . . . . .	59
3.11	Communication process between computer and quadrotor . . . . .	61
3.12	Miniature force-torque sensor . . . . .	63
3.13	Miniature force-torque sensor evaluation kit . . . . .	63
3.14	Miniature force-torque sensor software interface . . . . .	65
3.15	Experimental bench design . . . . .	66
3.16	Experimental bench . . . . .	67
3.17	Miniature force-torque sensor assembled in experimental bench . . . . .	67
3.18	Quadrotor dynamics simulation tests . . . . .	68
3.19	Rotation of quadrotor motors according to LED light . . . . .	69

---

3.20	Tests for the characterization of the quadrotor propellers . . . . .	69
3.21	Obtaining quadrotor tests results from force-torque sensor . . . . .	70
3.22	Quadrotor propellers data obtained from tests . . . . .	70
4.1	Inertial and body reference frames . . . . .	72
4.2	Rotation by $\psi$ about $z_I$ . . . . .	73
4.3	Rotation by $\theta$ about $y_2$ . . . . .	74
4.4	Rotation by $\phi$ about $x_3$ . . . . .	74
4.5	Quadrotor forces and moments convention . . . . .	76
4.6	Polynomial regression models of quadrotor propellers . . . . .	80
4.7	Quadrotor moments of inertia matrix obtained in CAD software . . . . .	81
4.8	Quadrotor attitude control system . . . . .	82
4.9	Quadrotor attitude control simulation results . . . . .	84
4.10	A node that receives three inputs . . . . .	85
4.11	Commonly used activation functions . . . . .	86
4.12	Neural network structure composed of layers of nodes . . . . .	87
4.13	Supervised learning scheme of a neural network . . . . .	88
4.14	Neural network that consists of two input and two output nodes . . . . .	90
4.15	How the different methods select training data and update the weights . . . . .	91
4.16	Neural network that consists of two input and two output nodes and a hidden layer with three nodes . . . . .	92

---

4.17 Neural network that consists of one input and one output nodes and a hidden layer of $n$ nodes . . . . .	95
4.18 Neural network thrust rotor model . . . . .	98
4.19 Neural network torque rotor model . . . . .	99
4.20 Effect of neurons in the hidden layer over MSE . . . . .	99
4.21 Neural network that consists of six input and three output nodes . . .	100
4.22 Mean Squared Error through epochs on inertial parameters estimation	102
4.23 Comparison of quadrotor angular accelerations between white box and grey box approaches . . . . .	104
4.24 Neural network that consists of six input and output nodes and two hidden layers of $n$ and $m$ nodes . . . . .	105
4.25 Mean Squared Error through epochs on data-driven identification . .	106
4.26 Comparison between real and estimated forces and moments from data-driven identification . . . . .	107

# LIST OF TABLES

---

2.1	Selected electronic devices for the low-cost UAV testing bench . . . . .	11
2.2	Components of load cells forces where $f_{nx}$ , $f_{ny}$ , and $f_{nz}$ are considered as input forces . . . . .	19
2.3	UAV testing bench distances . . . . .	20
2.4	Mechanical properties of Al-6061 and PLA . . . . .	22
2.5	PLA mechanical properties obtained by tensile tests . . . . .	32
2.6	Calibration weights measurements data . . . . .	34
2.7	Calibration weights combinations and load cells DV . . . . .	37
2.8	Coefficients $a_0$ and $a_1$ for load cells linear regression models . . . . .	38
3.1	Selected electronic components for the quadrotor . . . . .	53
3.2	Miniature force-torque sensor outline specifications . . . . .	64
4.1	Quadrotor parameters (white box approach) . . . . .	81
4.2	PID attitude controller gain factors . . . . .	83
4.3	Quadrotor parameters (grey box approach) . . . . .	103



---

A.1 Quadrotor motors signals for dynamics simulation tests . . . . .	112
--	-----

# ACKNOWLEDGEMENTS

---

I want to thank my thesis committee for taking the time and effort to revise my work.

In the same way, I want to thank my thesis advisors Edmundo Javier Ollervides Vazquez and Octavio Garcia Salazar for all of their support and guidance along my trajectory, both academic and personal.

I want to say thanks to my academic colleagues Carlos Gellida, Pablo Tellez, Victor Dominguez, and Eduardo Barrientos for their support and orientation during my studies.

I also want to thank the Center for Research and Innovation in Aeronautical Engineering and the Mechanical and Electrical Engineering faculty from the Autonomous University of Nuevo Leon for letting me be part of the institution and for providing the space and resources to help me achieve my results.

Lastly, I want to thank CONACyT for allowing me to study for a postgraduate degree through a scholarship.

Without all of you, this would not have been possible. Thank you for believing in me and for all of your support.

# SUMMARY

---

Alejandro Jiménez Flores.

Candidate to obtain the degree of Master of Science in Aeronautical Engineering  
with Orientation in Flight Dynamics.

Universidad Autónoma de Nuevo León.

Facultad de Ingeniería Mecánica y Eléctrica.

Study title: APPLICATION OF AN INTELLIGENT ALGORITHM IN THE SYSTEM  
IDENTIFICATION OF A QUADROTOR.

Number of pages: 121.

**OBJECTIVES AND STUDY METHOD:** The general objective of this research work is to identify the system of a quadrotor employing neural networks, a type of intelligent algorithm. The system identification is addressed according to three approaches; white box, grey box, and black box approaches, where the last two intelligent algorithms are employed. The specific objectives are listed below:

- To develop a low-cost testing bench to measure forces and moments exerted by a UAV.
- To construct a quadrotor to control the rotational speed of each motor independently.

- To carry out experimental tests with the low-cost testing bench to measure forces and moments exerted by the quadrotor.
- To identify the quadrotor system according to the white, grey, and black box approaches.
  - White box: To develop and simulate the quadrotor mathematical model according to physical principles.
  - Grey box: To develop and train a neural network to estimate the quadrotor mathematical model parameters.
  - Black box: To develop and train a neural network to predict the quadrotor forces and moments, using experimental data only.

On the other hand, the study method of this work is divided into three major parts. The first part consisted of designing and developing a testing bench that can measure forces and moments exerted on a UAV. The second part emphasizes designing and constructing a quadrotor from scratch so that its four motors can be controlled independently. The third part takes an issue on machine learning, where artificial neural networks are employed in order to carry out the quadrotor system identification according to the white, grey, and black box approaches.

**CONTRIBUTIONS AND CONCLUSIONS:** This research work addressed multiple subjects in diverse areas of engineering, from which the conclusions are presented as follows:

A low-cost UAV testing bench was designed and developed to measure forces and moments exerted by a UAV. Additionally, a mini force-torque sensor was acquired and an experimental base was created in order to measure all six quadrotor degrees of freedom.

The design and construction of a quadrotor from scratch was carried out and its four motors could successfully be characterized and controlled independently.

Consequently, experimental tests with the quadrotor and experimental base were performed effectively. Thus, forces and moments data were acquired and stored for the system identification process.

The quadrotor system identification was carried out according to each proposed method which are; white, grey, and black box approaches. The white box approach consisted of deriving the quadrotor equations of motion and simulating its dynamics. Conversely, in the grey box approach the parameters from the mathematical model were obtained by training a neural network with experimental and simulation data. Lastly, the black box approach focused only on the experimental data to train a neural network and thus identify the quadrotor system. In all three methods, the obtained results showed good precision and accuracy.

On the other hand, this research work contributed to two congress presentations, which are listed below:

- Design and fabrication of a low-cost drone testing bench. International Materials Research Congress 2022.
- A design modification of a quadrotor frame based on fused deposition modeling. International Conference on Unmanned Aircraft Systems 2024.

Advisor signature: \_\_\_\_\_

Dr. Octavio García Salazar

## CHAPTER 1

# INTRODUCTION

---

## 1.1 BACKGROUND

During the last few years, Unmanned Aerial Vehicles (UAVs) have been the focus of research and development of many scientists and engineers around the world due to their extended applications in a great variety of fields. A quadrotor is a type of UAV that possesses ideally four rotors with their propellers which are located in the vertices of a square and when spinning generate a thrust force in perpendicular direction to the plane of its rotors. There are different types and sizes of quadrotors depending on their application (see Figure 1.1).



Figure 1.1: Different types and sizes of quadrotors [1], [2], [3]

Quadrotors are UAVs with high maneuverability and the ability of vertical take-off landing, for this reason, they are adequate for aerial surveillance, photography, and search and rescue missions, among other applications [4]. Authors in

reference [5] designed a multi-vehicle testbed of quadrotor to facilitate testing of multi-vehicle control algorithms. In [6] it was conducted a simulation and experimental tests of quadrotors for the generation of formation trajectories. In reference [7] a quadrotor is presented with a collision avoidance application thanks to a real-time kinodynamic planning framework. In [8] multi quadrotor UAVs are contemplated to carry out precision agriculture by implementing a distributed control strategy. On the other hand, some authors take an issue with octorotors, for instance in [9] a PIV fuzzy gain scheduling flight controller is developed for a Mini-UAV octorotor. In [10] it is presented the modeling and simulation of an octorotor with a manipulator arm using the Denavit-Hartenberg convention and Newton-Euler method. Lastly, in [11] authors present the development of a feedback nonlinear controller for an octorotor designed with a nonlinear proportional velocity term, plus a linear integral term.

## 1.2 STATE OF THE ART

### 1.2.1 FORCES AND MOMENTS MEASUREMENT

The possibility of measuring all the forces and moments that are exerted on a vehicle was recognized five decades ago [12] and currently it is of practical importance in many engineering fields such as aerospace, control, robotics, biomechanical, automotive, among others [13], [14], [15], [16], [17], [18] and not limited to even clinical environments [19]. Figure 1.2 shows a commercial thrust stand mainly used to measure forces and moments exerted by the propellers of medium and large UAVs.

Authors from reference [20] showed the application of forces and moments in the development of a controller for a quadrotor, in [21] they focused on forces with a Tri-Tilt rotor while in [22] they involved forces and moments with a generic car model.



Figure 1.2: Thrust stand for medium and large UAVs [23]

The device capable of measuring the magnitude and direction of forces and moments exerted on a vehicle is usually referred to as a six-axis force-moment sensor, meaning that it can measure the quantities  $F_x$ ,  $F_y$ ,  $F_z$ ,  $M_x$ ,  $M_y$ ,  $M_z$  [15], [18]. Nevertheless, there is a great variety of these sensors according to their application. For instance, [24] propose the development and calibration of a three-axis force sensor because, according to their research, moments data may not be required. On the other hand, [25] carried out the application of a force/moment sensor for a humanoid robot foot where knowing the 6 Degrees of Freedom (DoF) quantities is essential for the performance of such vehicle, that is to walk safely. In the same way, [26] present research on a 6-foot underwater robotic vehicle where this kind of sensors play an important role in measuring hydrodynamic forces and moments acting on the robot under its specific and complex operating conditions, which in their case it is underwater.

Although there exists new innovative research on sensors such as air-lubricated force/moment [27] and Fiber Bragg grating sensors [28], one of the most common force/moment sensors used nowadays are those based on piezoresistive strain gauges [14], [15], [18], [29], [30], [31], [32].

Even though force/moment sensors have a wide range of applications in robotics,



there are too few in the aeronautical field, and therefore, too expensive that cannot be afforded by many research institutions. For instance, authors from reference [33] have developed a six DoF force/moment bench for quadrotors, nevertheless, the process and complexity of such work make it inappropriate when simplicity and low-cost manufacturing is desired.

### 1.2.2 SYSTEM IDENTIFICATION

System identification enables researchers to develop mathematical models of dynamic systems using measured data [34], where input-output data of a physical system is fitted to a mathematical model of the system [35]. Figure 1.3 shows the general difference between simulation and system identification in terms of aircraft.

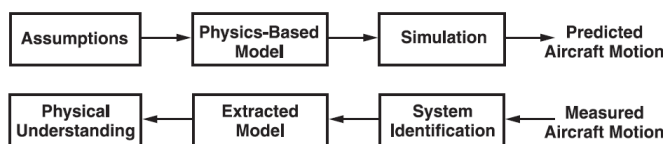


Figure 1.3: Difference between simulation and system identification [36]

The understanding and prediction of the behavior of complex systems relies on mathematical models [37]. There are several ways to identify the mathematical model of a system, for instance, the white box approach is the case when a physical system is perfectly known and can be modeled based on physical laws. On the other hand, the grey box approach refers to when the model structure of a system is known but several parameters are unknown and remain to be determined. Lastly, the black box approach is the case where there is no physical insight when modeling, therefore omitting the necessity of any previous model [38], [39], [40].

Dynamic mathematical models can describe the behavior of a quadrotor [41]. For example, authors in [42] proposed a standard generic model in order to standardize the quadrotor mathematical representation for control design, simulation, and estimation. In [43] it was applied the Extended Kalman Filter (EKF) in the

modeling and system identification for the auto-stabilization of a quadrotor. In [44] it was carried out a methodology which provides high precision and predictable maneuvering control of quadrotor micro aerial vehicles, focusing on the extension of the flight envelope of the mathematical model and enhancement of agile maneuvering capability. In [45] the authors considered the aerodynamic effects on a quadrotor in a high-speed regime and gray-box models were identified from flight data employing a stepwise system identification approach. On the other hand, in [46] an improved propulsion model for multirotor small UAVs was developed considering effects such as airflow speed, direction, rotor interaction, and other phenomena related to specific configurations.

### 1.2.3 ARTIFICIAL NEURAL NETWORKS

The human brain is the most complex structure known and its comprehension has been one of the most difficult challenges faced by science [47]. An Artificial Neural Network (ANN) can be considered a highly simplified model of the biological neural network (see Figure 1.4) and consists of interconnected processing units [48].

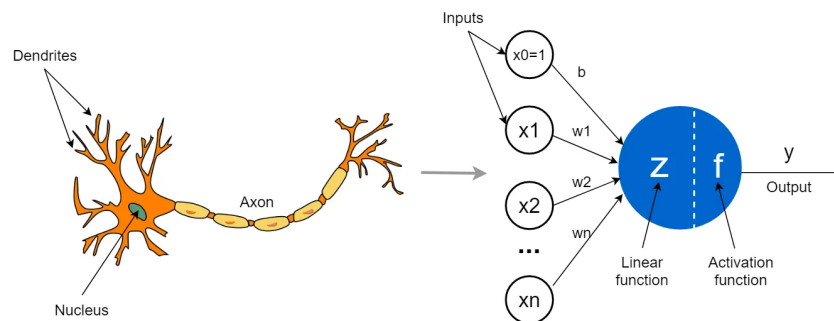


Figure 1.4: Comparison between a biological (left) and artificial (right) neuron [49]

ANNs have a wide range of applications including image recognition, classification, control, and system identification [50]. Authors in [51] used adaptive fuzzy spiking neurons for the identification of the propulsion subsystem of a multi-rotor

UAV and the parameters of the electric subsystem of an induction motor. In [52] it was developed a Deep Neural Network (DNN) based algorithm that enhances the tracking performance of a classic feedback trajectory controller for quadcopters. In [53] diverse deep learning architectures, including ANNs, were employed in order to identify a quadcopter UAV system. On the other hand, in [54] it is presented a real-time identification of a quadcopter carried out with Radial Basis Function neural network (RBF) trained with Minimal Resource Allocating Network (MRAN) algorithm, where the model is capable of identifying and modeling the quadcopter flight dynamics.

#### 1.2.4 PREVIOUS WORK

Considering the measurement of forces and moments on UAVs, there is previous work that is related to one of the objectives of this research work. In [55] a test bench was created to measure forces and moments produced by the propulsion system of small and micro UAVs in a wind tunnel, where part of the test bench is composed of translational and radial load cells. Although the test bench is designed and constructed to measure forces and moments exerted by the propulsion system of UAVs, it is designed to be implemented in a closed environment, in this case, a wind tunnel. Therefore, it differs from the purpose of the testing bench considered in this work, which is planned to be used in an open environment, where the airflow generated by the propellers is planned not to interfere with the floor and surrounding walls.

### 1.3 JUSTIFICATION

There are many types of force-torque sensors on the market, however, the great majority are high-cost due to their accuracy and number of degrees of freedom. For this reason, there is a need to design and build a low-cost testing bench that replicates the operation of a force-torque sensor for UAV applications.

Most of commercial quadrotors are not open source and therefore controlling their motors independently is complicated. Constructing a quadrotor from scratch is necessary to have direct access to all of its components and thus to be able to control its motors independently.

The mathematical model of a quadrotor can be derived by several methods such as the Newton-Euler approach, on which inertial and aerodynamic parameters are considered and complex to calculate. Intelligent algorithms such as neural networks, with the help of experimental databases, are proposed to estimate the unknown and complex quadrotor parameters in a precise manner.

### 1.4 HYPOTHESIS

- The development of a low-cost UAV testing bench can satisfactorily replace force-torque sensors and be used for UAV applications.
- Constructing a quadrotor from scratch permits the independent control of its four motors so that they can be characterized.
- Artificial neural networks can be applied in the quadrotor system identification to calculate parameters and predict forces and moments with the help of experimental databases.

## 1.5 GENERAL OBJECTIVE

The general objective of this research work is to identify the system of a quadrotor employing neural networks, a type of intelligent algorithm. The system identification is addressed according to three different approaches; white box, grey box, and black box approaches, where intelligent algorithms are employed in the last two.

### 1.5.1 SPECIFIC OBJECTIVES

- To develop a low-cost testing bench to measure forces and moments exerted by a UAV.
- To construct a quadrotor to control the rotational speed of each motor independently.
- To carry out experimental tests with the low-cost testing bench to measure forces and moments exerted by the quadrotor.
- To identify the quadrotor system according to the white, grey, and black box approaches.
  - White box: To develop and simulate the quadrotor mathematical model according to physical principles.
  - Grey box: To develop and train a neural network to estimate the quadrotor mathematical model parameters.
  - Black box: To develop and train a neural network to predict the quadrotor forces and moments, using experimental data only.

It is important to mention that adjustments were made during the research work. The low-cost testing bench provided partial data, for this reason, a miniature

force-torque sensor was considered and an experimental bench was constructed. The sensor was assembled into the bench and thus the experimental tests were successfully carried out.

## 1.6 RESULTS

This research work contributed to two congress presentations, which are listed below:

- Design and fabrication of a low-cost drone testing bench. International Materials Research Congress 2022.
- A design modification of a quadrotor frame based on fused deposition modeling. International Conference on Unmanned Aircraft Systems 2024.

## CHAPTER 2

# LOW-COST UAV TESTING BENCH

---

The main objective of the low-cost UAV testing bench is to measure the forces and moments that are exerted by a UAV. The testing bench is designed and fabricated considering the cost and reliability to provide accurate data. The design incorporates load cells that are arranged to be deflected in specific directions to acquire all the 6 degrees of freedom (DoF) data that are applied by the vehicle. A mathematical model based on forces and moments with respect to the UAV center of gravity was developed. Computer-aided design was performed to draw the components needed to support the load cells. Polylactic acid (PLA) and Aluminum 6061 alloy (Al-6061) were the selected materials, based on their mechanical properties and ease of manufacture. Finite Element Method (FEM) simulations were carried out to determine the deformation and stress distribution of the bench considering each material. Fabricated components and load cells were assembled along with the electronics needed for operation. Tests were conducted using a UAV, where forces exerted on both PLA and Al 6061 testing benches were compared with existing sensors, yielding positive outcomes. The proper selection of materials plays an important role in the design and fabrication, cost-stiffness ratio, and accuracy of recorded data. The low-cost testing bench is aiming to aid the research on UAVs where high-cost force/torque sensors cannot be accessible.

## 2.1 ELECTRONIC DEVICES

Research was carried out to find the elemental electronic devices needed in the testing bench in order to measure and process all the forces and moments exerted by a UAV. These devices were properly selected according to their scope and considerations of the proposed testing bench. Table 2.1 shows the selected electronic devices to be implemented, as well as their functioning.

Table 2.1: Selected electronic devices for the low-cost UAV testing bench

Device	Functioning
1 kg load cell	Transducer that converts a force into a measurable electrical output
ADS 1256 analog-to-digital converter	Sensor that converts analog to digital signal
Arduino <sup>®</sup> UNO	Integrated circuit used to record instructions written in a programming language

## 2.2 LOAD CELL WORKING PRINCIPLE

A load cell is an electro-mechanical device that is used to measure force and torque. When they are properly designed and used, they can provide very accurate and reliable data. Load cells can be used in many scenarios where weighting measurements are needed, such as in robotic arms, robot walking, industrial machinery, level of a tank, lift units, and others [56].



Figure 2.1: Different types of load cells [57]



There are different types of load cells depending on the type and value of forces and moments that are going to be measured. Figure 2.1 shows some examples of load cells used in different applications and scenarios. In this work, the single-point load cell is selected due to its direction of force application, cost, and compact size.

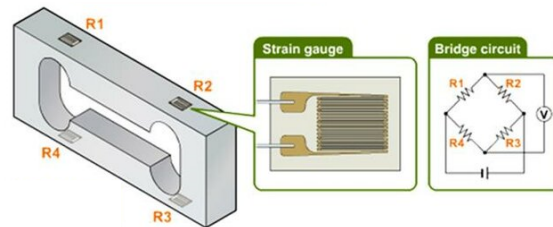


Figure 2.2: Strain gauges in a single point load cell [58]

A typical single-point load cell consists of a spring element, which is generally a bar of aluminum (with a narrow section) that includes strain gauges attached to it. This type of load cell contains a total of four strain gauges, two located at the top and two at the bottom surface of the spring element, and they are connected forming a Wheatstone bridge circuit (see Figure 2.2).

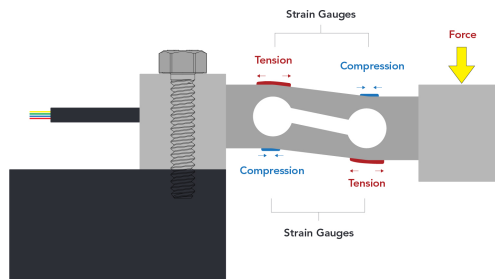


Figure 2.3: Tension and compression zones in a load cell under an applied force [59]

When the load cell is attached from one end and a force is applied on the other one, there will be tension and compression zones in the spring element, as seen in Figure 2.3. These mechanical changes in the spring element surface will directly impact the strain gauges so that the electrical resistance of each strain gauge will also vary. The intensity of the applied force will be proportional to the intensity of the electrical resistance variation in the strain gauges [56]. In this way, the Wheatstone

bridge configuration created with the strain gauges is useful to measure the voltage differential in each load cell, which the latter will be necessary for the calibration to estimate applied forces.

## 2.3 DESIGN

The overall design of the testing bench is proposed as shown in Figure 2.4. A rigid connector tube is fixed with a UAV at one end and to the testing bench at the other one, in such a way that all the elements are totally fixed. When in operation, the forces and moments exerted by the UAV will be directly transmitted to the testing bench through the connector tube. In this way, the forces and moments experienced by the testing bench frame  $\{T\}$  will be unlike to the ones experienced in the UAV body frame  $\{B\}$ , therefore the transformation  $\{T\} \rightarrow \{B\}$  has to be contemplated in the mathematical model in order to obtain the desired data from the UAV.

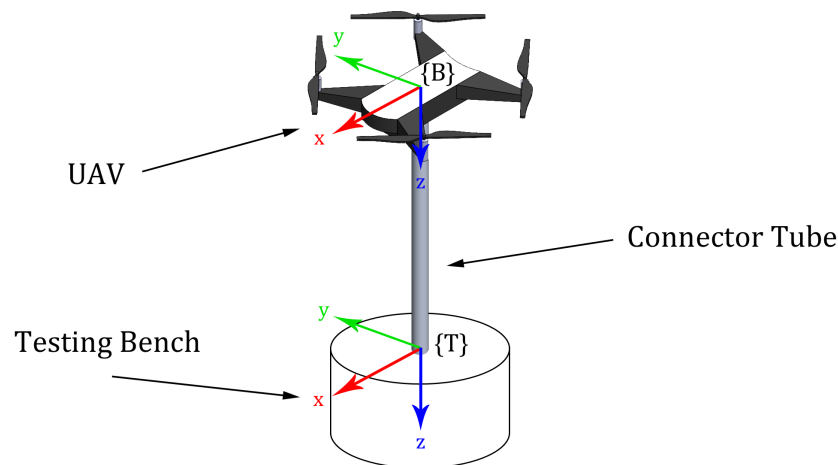


Figure 2.4: Low-cost UAV testing bench overall design

Since the single point load cell has been previously chosen as the sensor that will measure the forces and moments exerted by the UAV, a proper design for the testing bench considering these types of sensors needs to be proposed. In this manner, the main objective to be achieved in the design of the testing bench is to successfully

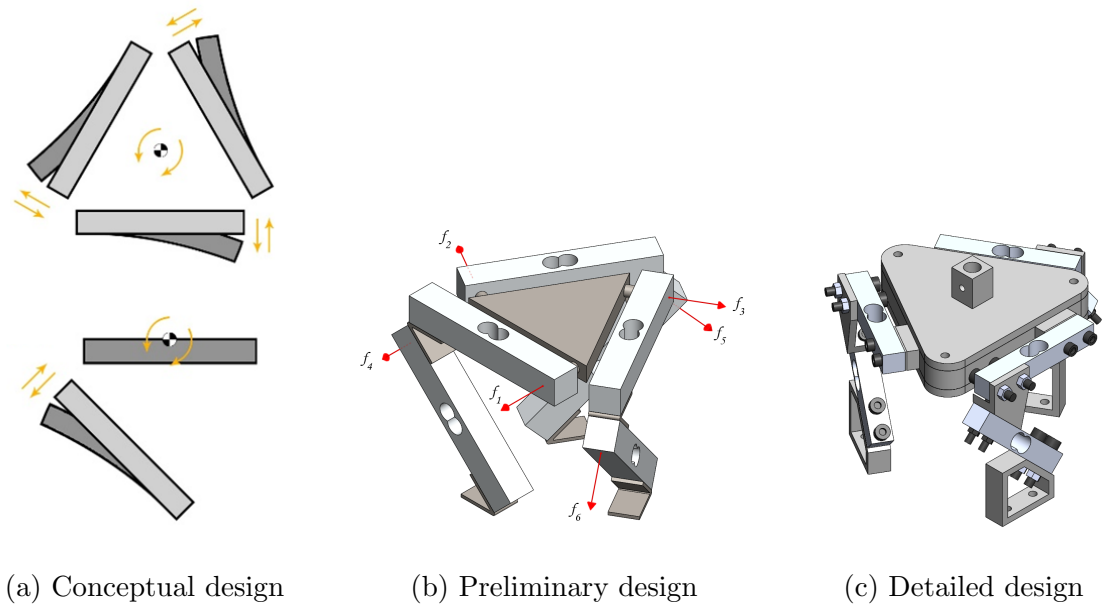


Figure 2.5: Design process of the low-cost UAV testing bench

acquire all the UAV 6 degrees of freedom data (that is  $F_x$ ,  $F_y$ ,  $F_z$ ,  $M_x$ ,  $M_y$ , and  $M_z$ ) by utilizing an arrangement of single point load cells, where at least one load cell has to be able to deform when applying a force/moment in one axis from the testing bench frame  $\{T\}$ . Thus, when applying a force/moment in any of the three axes of the testing bench frame, at least one load cell has to be deformed. In this way, the design of the testing bench considers a total of six load cells that are positioned and oriented in specific ways to allow them all to correctly deform and therefore to provide reliable data.

Once the idea of the testing bench was stated, the design process started iterating. Figure 2.5 shows the design process of the testing bench, starting with a conceptual design, followed by the preliminary design, and finishing with the detailed design. Iterations were necessary in order to comply with the specifications of the testing bench, considering electronic devices, distances and angles between elements, geometries of fixing elements, and materials.

The testing bench final design considers four different rigid pieces, which fix the load cells in their correct position and connect to the tube that will be holding

the UAV. Figure 2.6 illustrates more in detail the fixing pieces, as well as the load cells in their correct arrangement. The model of the testing bench was created in the Computer Aided Design (CAD) software Solidworks®.

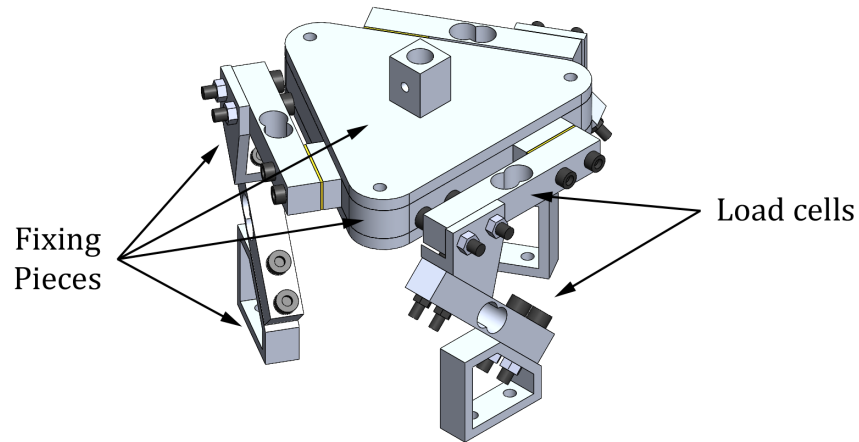


Figure 2.6: Testing bench elements

Figure 2.7 shows the four different pieces designed to fix the load cells and the connector tube to the testing bench. The design of such pieces was thought to be simple for manufacturing purposes while complying with their objective.

## 2.4 MATHEMATICAL MODEL

The equations that allow the calculation of all the UAV forces and moments were developed based on the load cells arrangement from the testing bench design. Prior

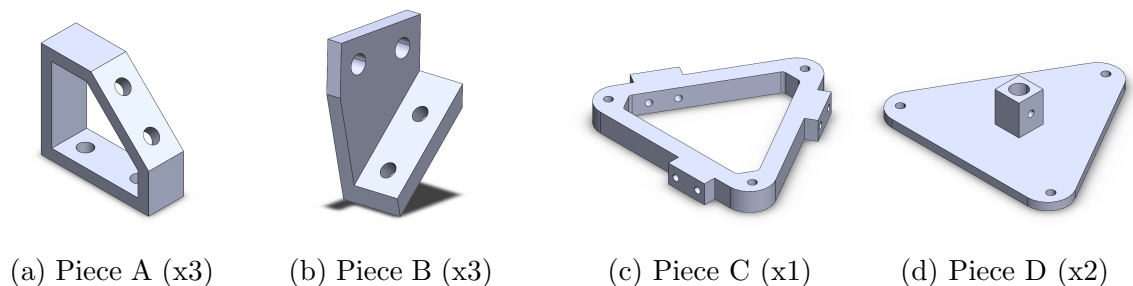


Figure 2.7: Designed pieces utilized in the testing bench

to the above, the reference frames for both the UAV and testing bench were chosen to be the North-East-Down (NED) reference frame. Moreover, the distances from the force application section of each load cell with respect to the testing bench frame  $\{T\}$  were also determined, as well as the connector tube distance from the testing bench frame  $\{T\}$  to the UAV frame  $\{B\}$ . Figure 2.8 illustrates the previous information from different views.

The first attempt in the mathematical model to acquire the six degrees of freedom experienced on  $\{B\}$  was to propose the equations for the sum of both forces and moments exerted on  $\{T\}$  and then transforming them into  $\{B\}$  as a function of distance  $L$ . Nevertheless, the aforementioned approach omitted physical considerations which were detected in the experimental phase; consequently the results on  $\{B\}$  possessed inconsistencies. For this reason, a new approach was proposed in order to get the testing bench to provide correct values while taking into account its structural environment.

The new approach considers the testing bench as a cantilever beam (see Figure 2.9) where in a simplified way the unknown force  $F$  applied on  $\{B\}$  is determined thanks to a sum of moments with respect  $\{T\}$  taking into account the reaction forces  $R_n$  generated and provided by each load cell. Since the testing bench is totally rigid, when loads are applied the sum of moments is equal to zero at any point. In this way, and considering the distances  $D$  and  $L$ ; the forces applied on  $\{B\}$  can be correctly obtained, as expressed in Equation 2.1 where there is one equation and one unknown variable ( $F$ ) which leads to a unique solution.

$$F \cdot L = (R_1 - R_2 + R_3) \cdot D \quad (2.1)$$

The above approach is able to compute all three forces ( $F_x$ ,  $F_y$ , and  $F_z$ ) that are applied on  $\{B\}$ , nonetheless, when trying to acquire moments data, the system of equations becomes indeterminate as shown in Equation 2.2 where for each axis there is one equation but two variables ( $F$  and  $M$ ) resulting in an infinite number

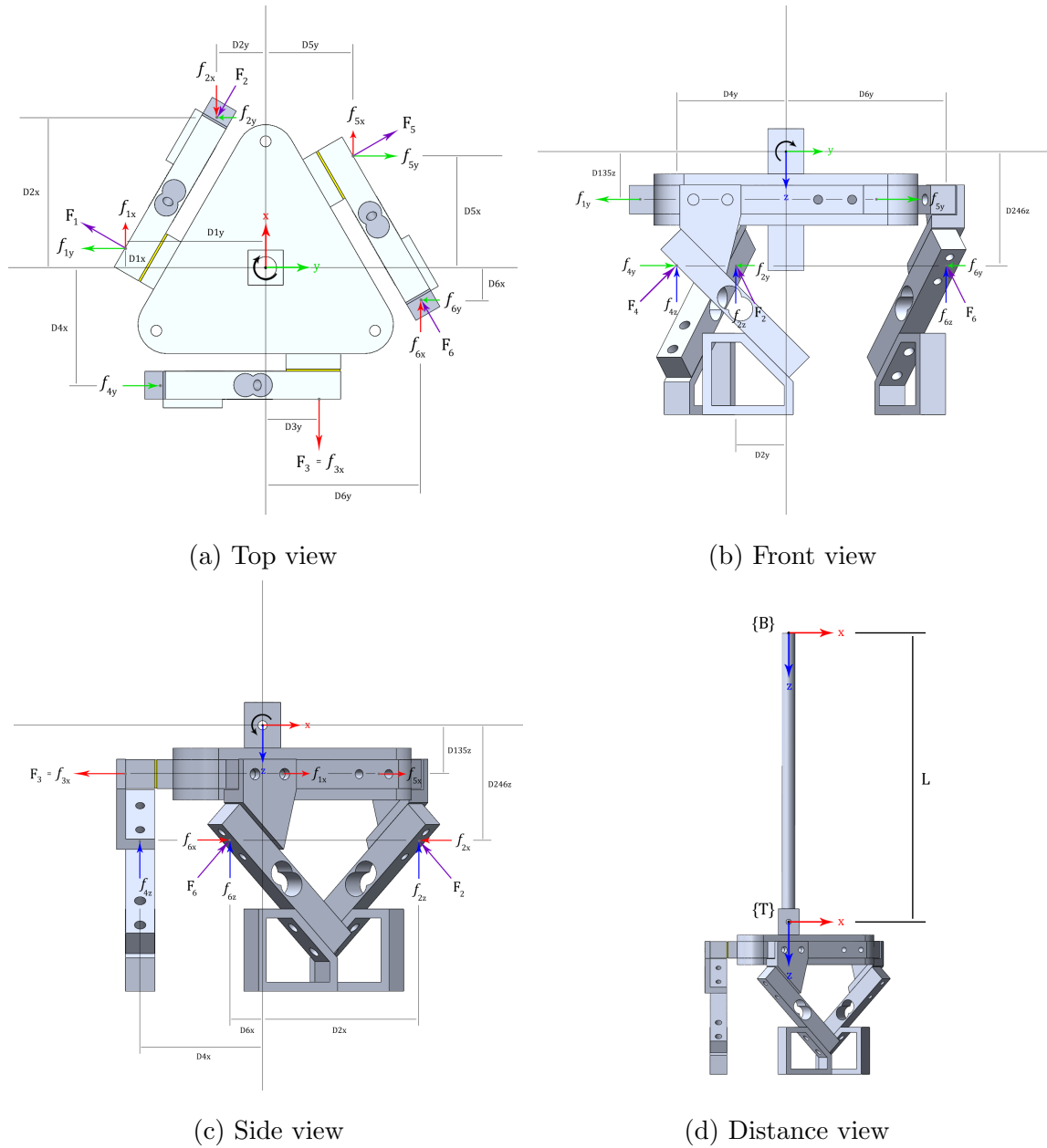


Figure 2.8: Testing bench reference frames and notation of forces and distances

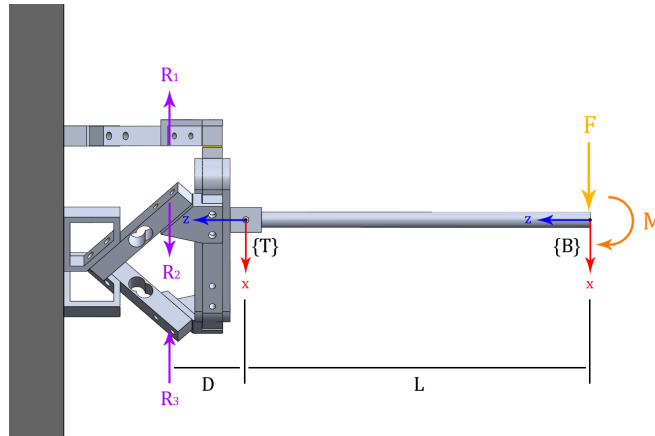


Figure 2.9: Testing bench as cantilever beam approach with simplified loads for explanation

of solutions. For this reason, the scope of the testing bench is reduced to calculating only the force variables in  $\{B\}$ .

$$F \cdot L + M = (R_1 - R_2 + R_3) \cdot D \quad (2.2)$$

Since the load cell is used to sense a force applied perpendicularly to its longitudinal distance, the data it provides is the perpendicular force the load cell experiments, which does not always coincide with the force that is actually being exerted since it depends on the direction of the force application. By not knowing the direction of this input force application, the load cell would not be able to provide the actual applied force on each axis. For this reason, the directions of the input forces were limited to the axes  $x$ ,  $y$ , and  $z$ , and thus the testing bench scope is once again reduced to only measuring forces directly applied on the aforementioned axes.

Figure 2.10 shows graphically how the projection forces change depending on the input force direction and Table 2.2 shows the equations developed to find the input forces  $f_{nx}$ ,  $f_{ny}$ , and  $f_{nz}$  as a function of the load cell perpendicular force  $F_n$ , where expressions  $C_{30}$ ,  $C_{45}$ , and  $C_{60}$  correspond to the cosine operations  $\cos 30$ ,  $\cos 45$ , and  $\cos 60$ , respectively.

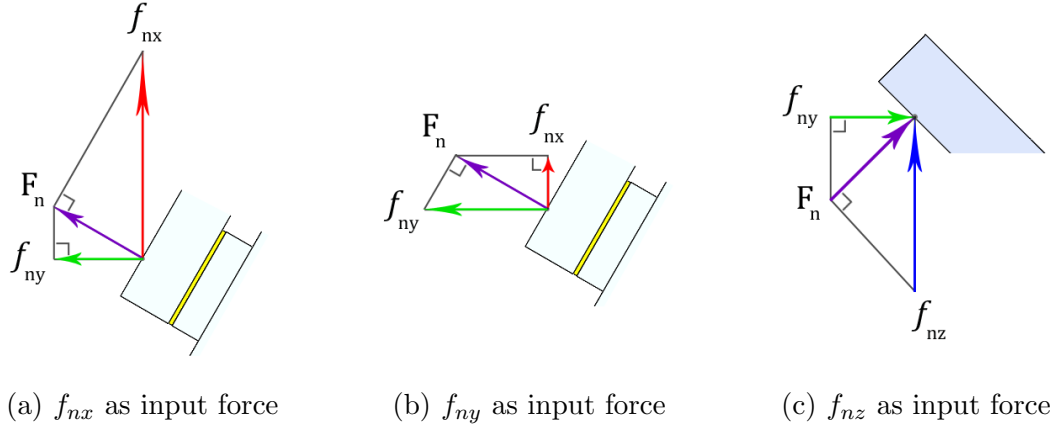


Figure 2.10: Load cell force projection under different force inputs

Table 2.2: Components of load cells forces where  $f_{nx}$ ,  $f_{ny}$ , and  $f_{nz}$  are considered as input forces

Axis	Cell 1	Cell 2	Cell 3	Cell 4	Cell 5	Cell 6
x	$f_{1x} = \frac{F_1}{C_{60}}$	$f_{2x} = \frac{F_2}{C_{30}C_{45}}$	$f_{3x} = F_3$	$f_{4x} = N/A$	$f_{5x} = \frac{F_5}{C_{60}}$	$f_{6x} = \frac{F_6}{C_{30}C_{45}}$
y	$f_{1y} = \frac{F_1}{C_{30}}$	$f_{2y} = \frac{F_2}{C_{60}C_{45}}$	$f_{3y} = N/A$	$f_{4y} = \frac{F_4}{C_{45}}$	$f_{5y} = \frac{F_5}{C_{30}}$	$f_{6y} = \frac{F_6}{C_{60}C_{45}}$
z	$f_{1z} = N/A$	$f_{2z} = \frac{F_2}{C_{45}}$	$f_{3z} = N/A$	$f_{4z} = \frac{F_4}{C_{45}}$	$f_{5z} = N/A$	$f_{6z} = \frac{F_6}{C_{45}}$

The distances required to carry out the sum of moments from any load exerted on the testing bench with reference to  $\{T\}$  are presented in Table 2.3.

Equations 2.3, 2.4 and 2.5 are the sum of moments around  $x$ ,  $y$ , and  $z$  axis respectively. On the other hand, Equations 2.6, 2.7 and 2.8 allow the calculation of the forces on  $\{B\}$ . These equations are derived according to the forces and distances presented in Figure 2.8.

Note that Equation 2.8 does not incorporate the sum of moments since the load cells deform equally when a vertical input force is applied. Therefore, it was chosen the approach of contemplating the deflections of all load cells.



Table 2.3: UAV testing bench distances

X axis	Y axis	Z axis
$D_{1x} = 9.0262850 \text{ mm}$	$D_{1y} = 64.086980 \text{ mm}$	$D_{135z} = 21.20 \text{ mm}$
$D_{2x} = 68.585472 \text{ mm}$	$D_{2y} = 22.541324 \text{ mm}$	$D_{246z} = 50.457916 \text{ mm}$
$D_{4x} = 53.814095 \text{ mm}$	$D_{3y} = 24.226497 \text{ mm}$	$L = 225 \text{ mm}$
$D_{5x} = 50.987810 \text{ mm}$	$D_{4y} = 48.126099 \text{ mm}$	
$D_{6x} = 14.771377 \text{ mm}$	$D_{5y} = 39.860483 \text{ mm}$	
	$D_{6y} = 70.667423 \text{ mm}$	

$$\begin{aligned} \sum M_x &= f_{1y}D_{135z} + f_{2y}D_{246z} - f_{4y}D_{246z} - f_{5y}D_{135z} + f_{6y}D_{246z} \\ &\quad + f_{2z}D_{2y} + f_{4z}D_{4y} - f_{6z}D_{6y} \end{aligned} \quad (2.3)$$

$$\begin{aligned} \sum M_y &= f_{1x}D_{135z} - f_{2x}D_{246z} - f_{3x}D_{135z} + f_{5x}D_{135z} + f_{6x}D_{246z} \\ &\quad + f_{2z}D_{2x} - f_{4z}D_{4x} - f_{6z}D_{6x} \end{aligned} \quad (2.4)$$

$$\begin{aligned} \sum M_z &= f_{1x}D_{1y} - f_{2x}D_{2y} + f_{3x}D_{3y} - f_{5x}D_{5y} - f_{6x}D_{6y} - f_{1y}D_{1x} \\ &\quad - f_{2y}D_{2x} - f_{4y}D_{4x} + f_{5y}D_{5x} + f_{6y}D_{6x} \end{aligned} \quad (2.5)$$

$$\sum F_x = \frac{-\sum M_y}{L} \quad (2.6)$$

$$\sum F_y = \frac{\sum M_x}{L} \quad (2.7)$$

$$\sum F_z = -(|F_1| + f_{2z} + |F_3| + f_{4z} + |F_5| + f_{6z}) \quad (2.8)$$

Using the previous equations and representing them as vectors and matrices, the mathematical model of the testing bench is presented in Equation 2.9.

$$\mathbf{F} = \mathbf{M}\mathbf{f} - \begin{bmatrix} 0_{2 \times 3} \\ 1_{1 \times 3} \end{bmatrix} |\mathbf{f}_{135}| \quad (2.9)$$

where  $\mathbf{F} = [F_x \ F_y \ F_z]^T$  is the vector of forces exerted on  $\{B\}$ ,  
 $\mathbf{f} = [F_1 \ F_2 \ F_3 \ F_4 \ F_5 \ F_6]^T$  is the vector of perpendicular forces experienced by

each load cell,  $\mathbf{f}_{135} = [F_1 \ F_3 \ F_5]^T$  is the vector of perpendicular forces experienced only by load cells 1, 3, and 5, and

$$\mathbf{M} = \begin{bmatrix} -\frac{D_{135z}}{C_{60}L} & \frac{1}{C_{45}L} \left( \frac{D_{246z}}{C_{30}} - D_{2x} \right) & \frac{D_{135z}}{L} & \frac{D_{4x}}{C_{45}L} & -\frac{D_{135z}}{C_{60}L} & -\frac{1}{C_{45}L} \left( \frac{D_{246z}}{C_{30}} - D_{6x} \right) \\ \frac{D_{135z}}{C_{30}L} & \frac{1}{C_{45}L} \left( \frac{D_{246z}}{C_{60}} + D_{2y} \right) & 0 & \frac{1}{C_{45}L} (D_{4y} - D_{246z}) & -\frac{D_{135z}}{C_{30}L} & \frac{1}{C_{45}L} \left( \frac{D_{246z}}{C_{60}} - D_{6y} \right) \\ 0 & -\frac{1}{C_{45}} & 0 & -\frac{1}{C_{45}} & 0 & -\frac{1}{C_{45}} \end{bmatrix}$$

is the matrix that converts the load cells input forces from  $\{T\}$  into the desired forces on  $\{B\}$ .

## 2.5 FEM SIMULATIONS

Since the testing bench is proposed to be fabricated in both Al-6061 and PLA materials (see Figure 2.11), Finite Element Method (FEM) simulations considering these materials were carried out. The FEM analysis has the main purpose of obtaining the testing bench deformations and stresses under different loads, so that it can be anticipated whether the testing bench is going to provide accurate data or not, depending on the implemented material. These simulations were carried out in ANSYS<sup>®</sup> Workbench. Figure 2.12 shows the static structural analysis followed in the software, where the testing bench geometry was imported and all the simulation parameters were established.

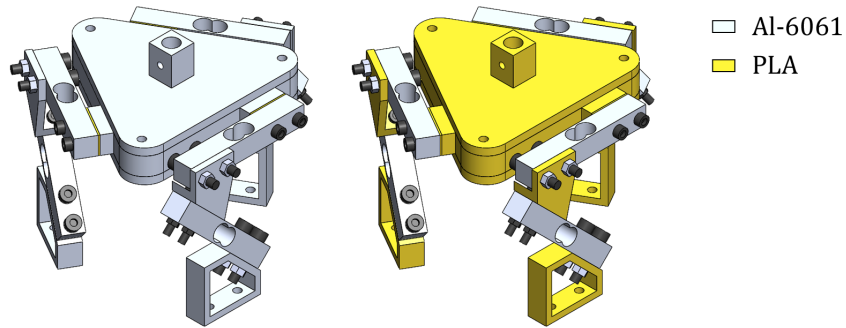


Figure 2.11: Testing bench fixing pieces with different materials

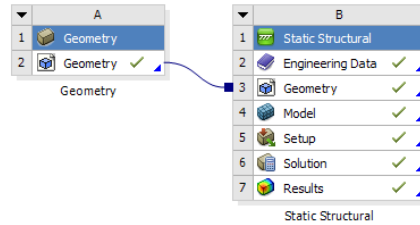


Figure 2.12: ANSYS® Workbench static structural analysis

### 2.5.1 PARAMETERS SETTING

Al-6061 and PLA mechanical properties were imported into the software in the Engineering Data section of the static structural analysis. Table 2.4 shows the mechanical properties of the proposed materials for the testing bench. PLA mechanical properties were obtained experimentally by means of tensile tests; Subsection 2.6.4 describes this procedure in more detail.

Table 2.4: Mechanical properties of Al-6061 and PLA

Property	Al-6061	PLA
Density [kg/m <sup>3</sup> ]	2770	1252
Elastic Modulus (E) [GPa]	68.9	1.9482
Ultimate Tensile Strength (UTS) [MPa]	310	41.388

Once the mechanical properties of the materials were imported, the corresponding materials were assigned to each element of the testing bench in the Model section, as seen in Figure 2.13. In the same way, the mesh of the whole testing bench was created adding face sizing on various surfaces in order to obtain a more refined result. The final mesh resulted in 724,698 nodes and 211,991 elements (see Figure 2.14).

Continuing with the setting parameters for the FEM simulations, the boundary conditions are then allocated according to the functioning of the testing bench. Figure 2.15a shows the fixed supports which are assigned to the inferior surfaces

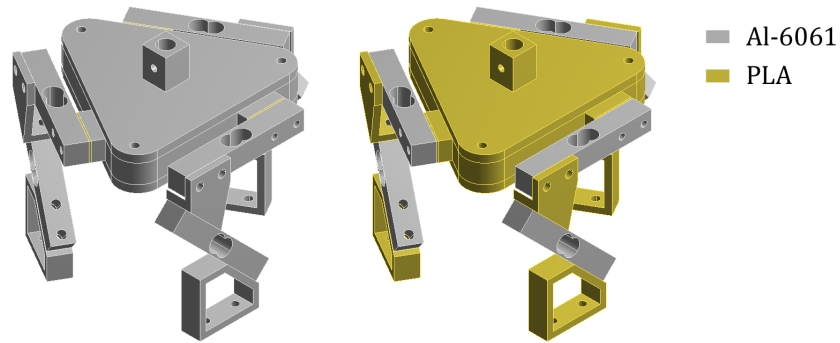


Figure 2.13: Testing bench materials assignment for FEM simulations

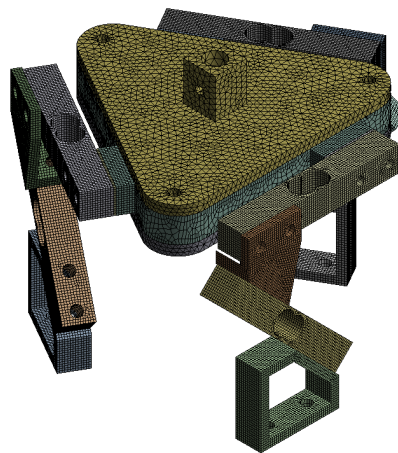


Figure 2.14: Testing bench mesh for FEM simulations

of each Piece A whereas Figure 2.15b shows the applied load (pressure) over the smallest surface at the top of Piece D. The magnitude of such load was varied from 1.9 KPa to 52.9 KPa which corresponds to a force range from 35.7 gf to 957.5 gf. Forces are originally considered as the loads for experimentation, nevertheless, they were converted into pressures for the FEM simulations in order to get uniformity in the results. These chosen loads are known forces (weights) that have been selected for the calibration of the testing bench. Subsection 2.7.1 describes these calibration weights in more detail.

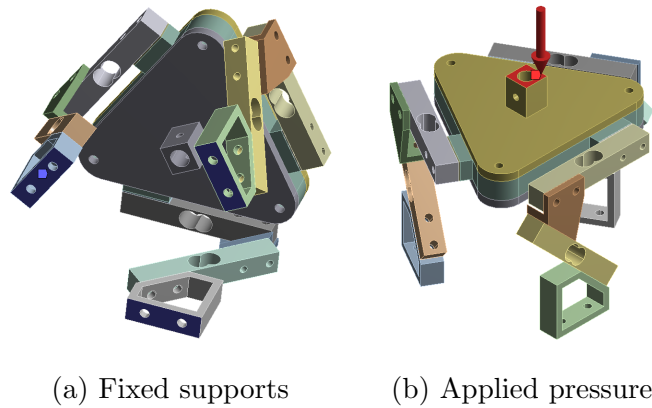


Figure 2.15: Testing bench boundary conditions for FEM simulations

## 2.5.2 SIMULATIONS RESULTS

As mentioned at the beginning of this section, the objective of the numerical analysis is to obtain the testing bench deformations and stresses under the application of different loads. Figure 2.16 shows the testing bench total deformation result for both Al-6061 and PLA materials. Red areas show greater deformation, whilst blue ones least deformation. It is important to mention that these figures only represent the zones where they present such deformations, but magnitudes are different. In the case of the Al-6061 testing bench, one can see that the magnitude of the deformation is carried out equally throughout the entire piece D, on the other hand, in terms of the PLA testing bench, the deformation is highly concentrated in the center of piece D.

The stress analysis was carried out according to the von Mises stress theory. Figure 2.17 shows the stress distribution throughout all the testing bench taking into account both Al-6061 and PLA materials. One can see that the total maximum stresses are located in different sites, whereas the surface of local maximum stresses (located in the narrow section of one load cell) are greater on the Al-6061 testing bench.

Figure 2.18 shows the magnitudes of both deformation and stresses analysis

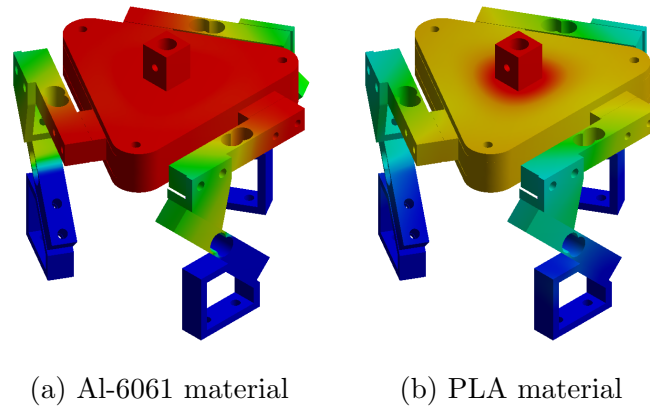


Figure 2.16: Testing bench FEM total deformation

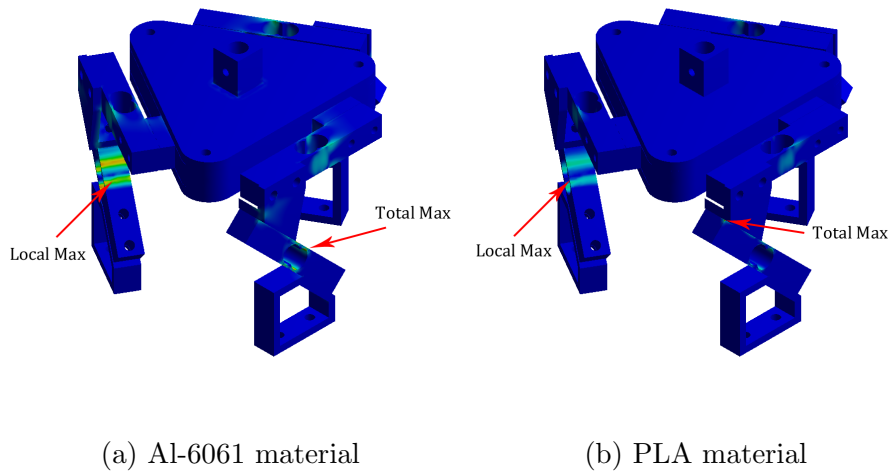
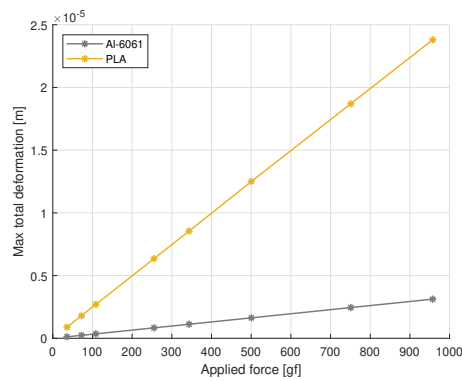


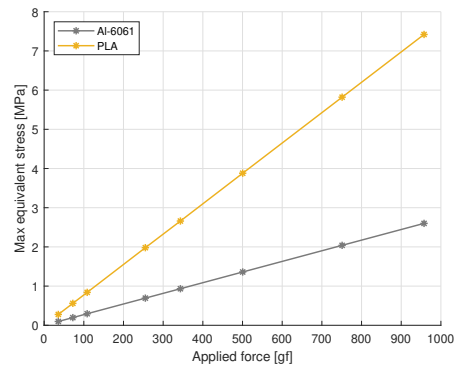
Figure 2.17: Testing bench FEM equivalent stress

carried out in the FEM simulations. Note in Figure 2.18a that as the applied force increases, the total deformation also does for both Al-6061 and PLA materials. Nonetheless, with the same applied force, the PLA deforms more highly than Al-6061, which is an expected result because of the mechanical properties of these materials. This behavior can also be observed in the stress analysis of Figure 2.18b where there is a big difference between the Al-6061 and PLA maximum equivalent stresses as the applied force increases. Nevertheless, these stress magnitudes do not correspond to the same point in the testing bench, reason why a local analysis had to be carried out in order to compare. Figure 2.18c shows the maximum stresses located

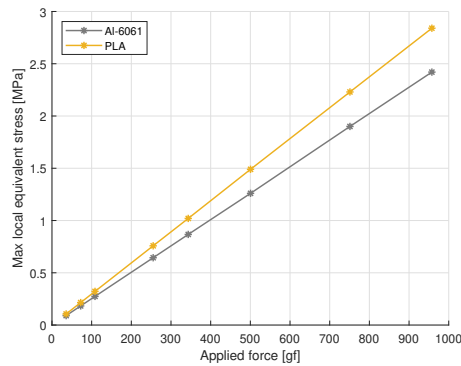
in the narrow section of a specific load cell as previously indicated in Figure 2.17. In this case, the magnitudes of such stresses are very close to each other, meaning that changing material does not highly affect the stresses experienced by the load cells and therefore, the results thrown by the strain gauges would not vary much. However, an important point to mention is that there is more area of stress in the Al-6061 testing bench than in the PLA as previously seen. This fact could also affect the values thrown by the strain gauges, but this is uncertain until experimentation.



(a) Maximum total deformation



(b) Maximum equivalent stress



(c) Maximum local equivalent stress

Figure 2.18: Testing bench FEM simulations results

## 2.6 FABRICATION

The low-cost UAV testing bench is doubly fabricated. One testing bench is fabricated with Al-6061 and another one with PLA, using machining and additive manufacturing techniques respectively.

### 2.6.1 ELECTRONICS ENCLOSURE

Before the fabrication of the testing bench fixing pieces, the load cells, ADS 1256, and Arduino<sup>®</sup> UNO were all connected together according to the diagrams for their correct operation (see Figure 2.19).

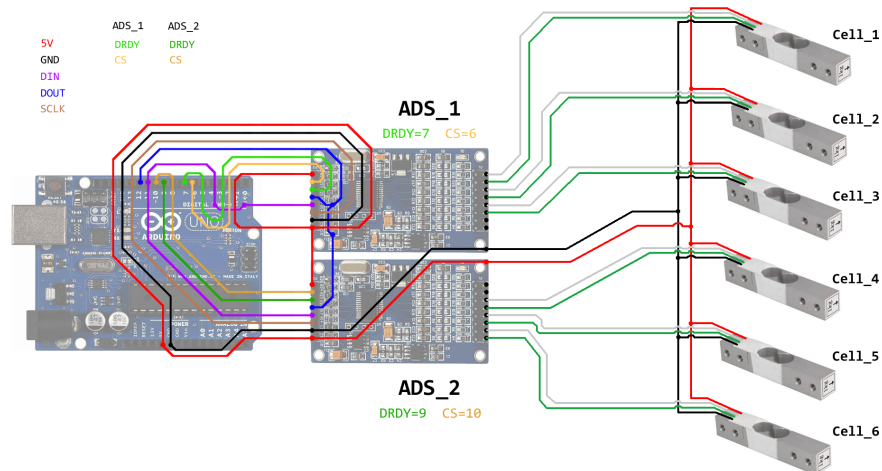


Figure 2.19: Testing bench electronics connection diagram

The two ADS 1256 and Arduino<sup>®</sup> UNO were soldered on a PCB (Printed Circuit Board) copper-clad board where the routing was previously performed. The result of this procedure can be observed in Figure 2.20. Once these devices were soldered into the PCB, an electronics enclosure was fabricated, which is a box-like structure designed to protect, contain, and enclose electronic devices. This box was created by means of additive manufacturing using a Creality<sup>®</sup> Ender-3 3D printer.



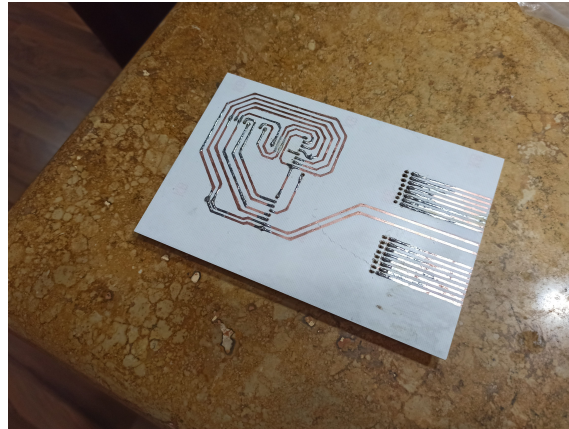


Figure 2.20: Testing bench PCB soldered

Figure 2.21 shows the 3D printed electronics enclosure as well as the PCB with its electronics together.

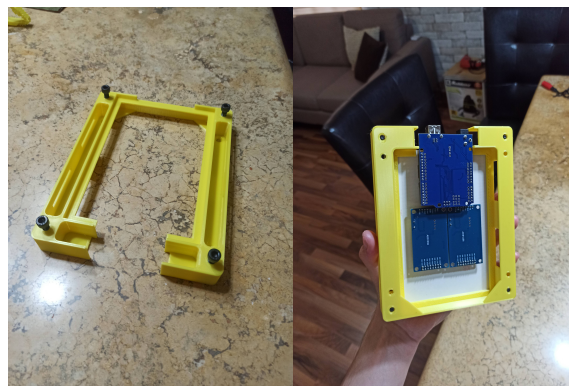


Figure 2.21: Testing bench electronics enclosure

Additionally to the electronics enclosure, three mini boxes were also 3D printed in order to protect the routing that connects the six load cells to the two ADS 1256. The connection between the routing and the ADS 1256 is carried out by means of a simple Ethernet cable. Figure 2.22 shows all the connected electronics including all the six load cells.

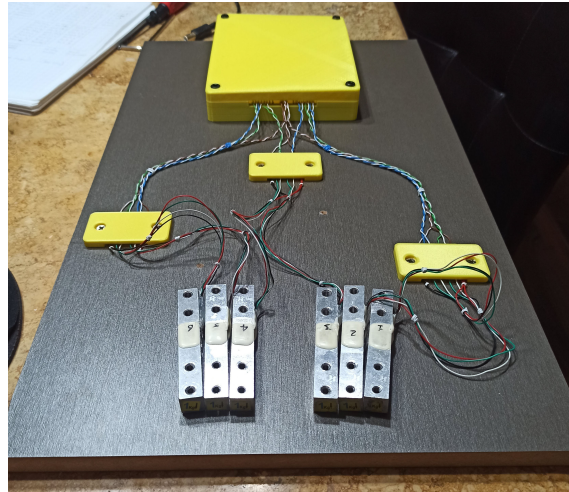


Figure 2.22: All testing bench electronics connected

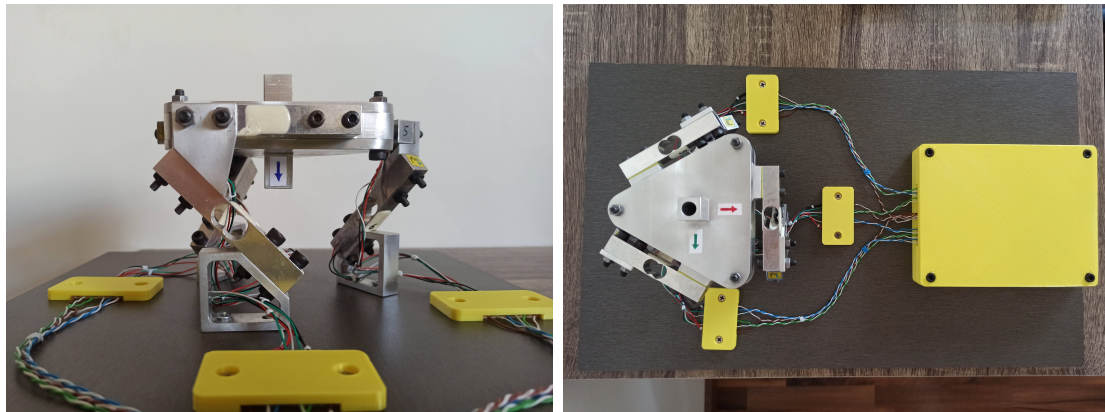
### 2.6.2 AL-6061 TESTING BENCH

The Al-6061 testing bench is manufactured by means of machining techniques, where fixing pieces A, B, C, and D were created from Al-6061 alloy blocks. These pieces are shown in Figure 2.23.



Figure 2.23: Testing bench Al-6061 fixing pieces

Once the Al-6061 fixing pieces were fabricated, the next step was to assemble them together with the load cells according to the testing bench design. For this, M4 and M5 screws and nuts were used. The final assembly of the Al-6061 testing bench is presented in Figure 2.24.



(a) Front view

(b) Top view

Figure 2.24: Al-6061 low-cost UAV testing bench

### 2.6.3 PLA TESTING BENCH

The PLA testing bench is manufactured by means of additive manufacturing, where fixing pieces A, B, C, and D were printed using a Creality<sup>®</sup> Ender-3 3D printer. These pieces are shown in Figure 2.25.

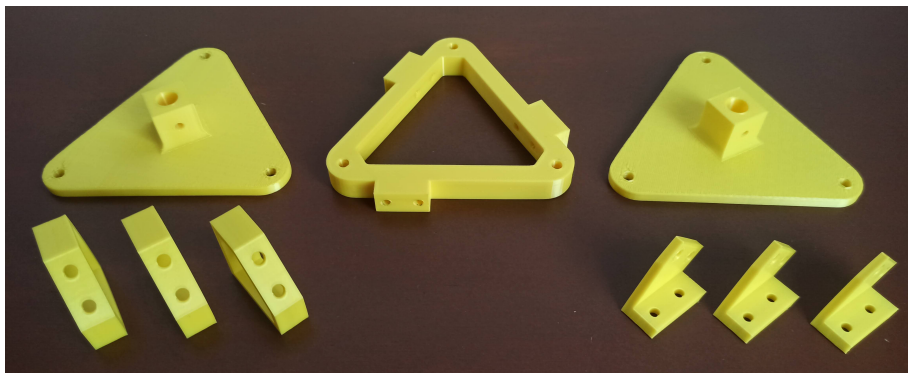
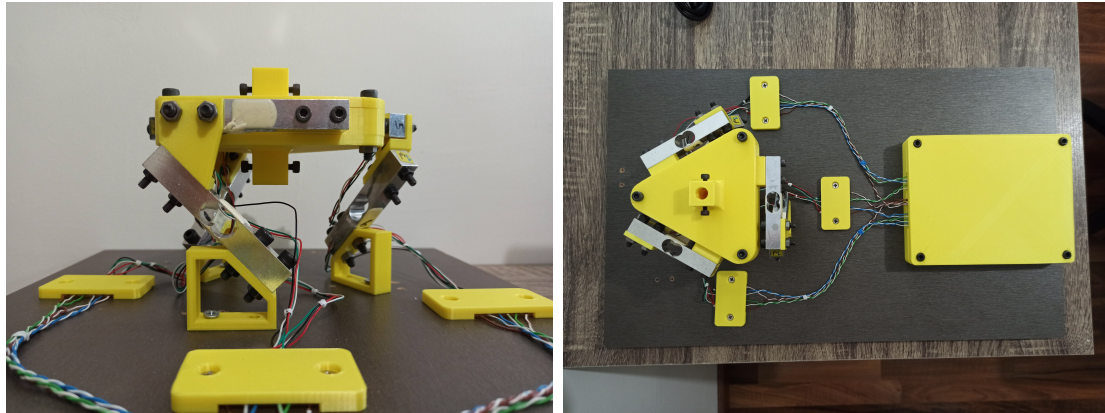


Figure 2.25: Testing bench PLA fixing pieces

Once the PLA fixing pieces were printed, the next step was to assemble them together with the load cells according to the testing bench design. Same as with the Al-6061 testing bench, M4 and M5 screws and nuts were used. The final assembly of the PLA testing bench is presented in Figure 2.26.



(a) Front view

(b) Top view

Figure 2.26: PLA low-cost UAV testing bench

#### 2.6.4 PLA SPECIMENS

To obtain the mechanical properties of PLA there were created type IV tensile test specimens. Four kinds of specimens were fabricated varying their infill density of 25, 50, 75, and 100 percent. These specimens were also created using a Creality® Ender-3 3D printer. Figure 2.27 shows each type of specimen and its quantity. Note that for each kind of specimen, there is a minimum of three specimens; this is to ensure reliability in the results of the mechanical tests.

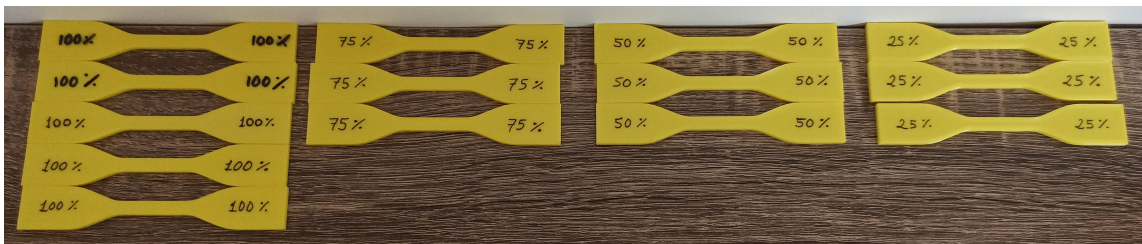


Figure 2.27: Type IV tensile test specimens

All the specimens were subjected to tensile tests using a SHIMADZU® AGS-X tensile tester with a speed of 10mm/min, 10 samples per second, and a distance between grips of 4cm. Figure 2.28 shows the procedure of the mechanical tests with the printed specimens.

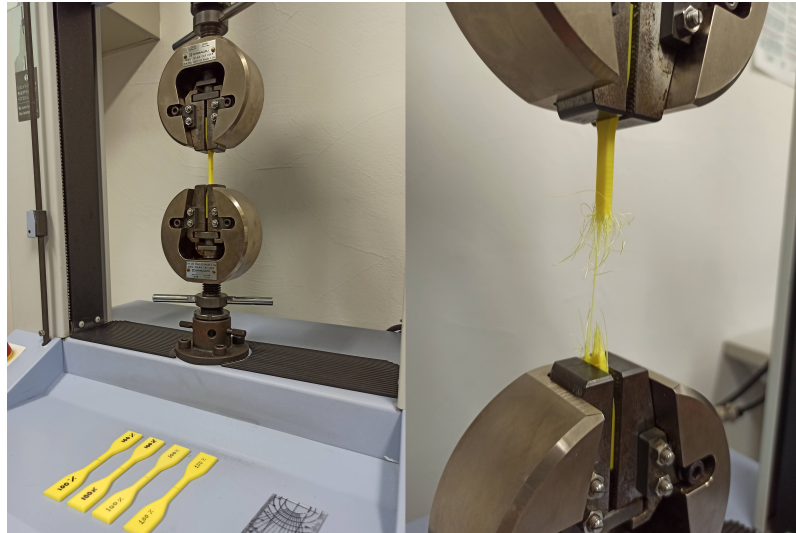


Figure 2.28: Type IV tensile test specimens under tension

Thanks to the tensile tests, the stress-strain curves of all four kinds of specimens could be obtained, and therefore, their mechanical properties. Figure 2.29 shows the stress-strain curves of PLA and Table 2.5 presents the mechanical properties data obtained from such curves.

Table 2.5: PLA mechanical properties obtained by tensile tests

Property	Infill density			
	25%	50%	75%	100%
Elastic Modulus (E) [GPa]	1.6099	1.6907	1.8487	1.9482
Ultimate Tensile Strength (UTS) [MPa]	36.811	38.591	41.233	41.388
Yield Strength [MPa]	12.75	13.2	15.16	16.02

## 2.7 CALIBRATION AND PROGRAMMING

The calibration of the testing bench is carried out in order for it to deliver reliable results. The programming is carried out in the MATLAB<sup>®</sup> software. The communication between the Arduino<sup>®</sup> UNO and the PC is via COM port, and the testing bench software is used to process all the data obtained from the load cells

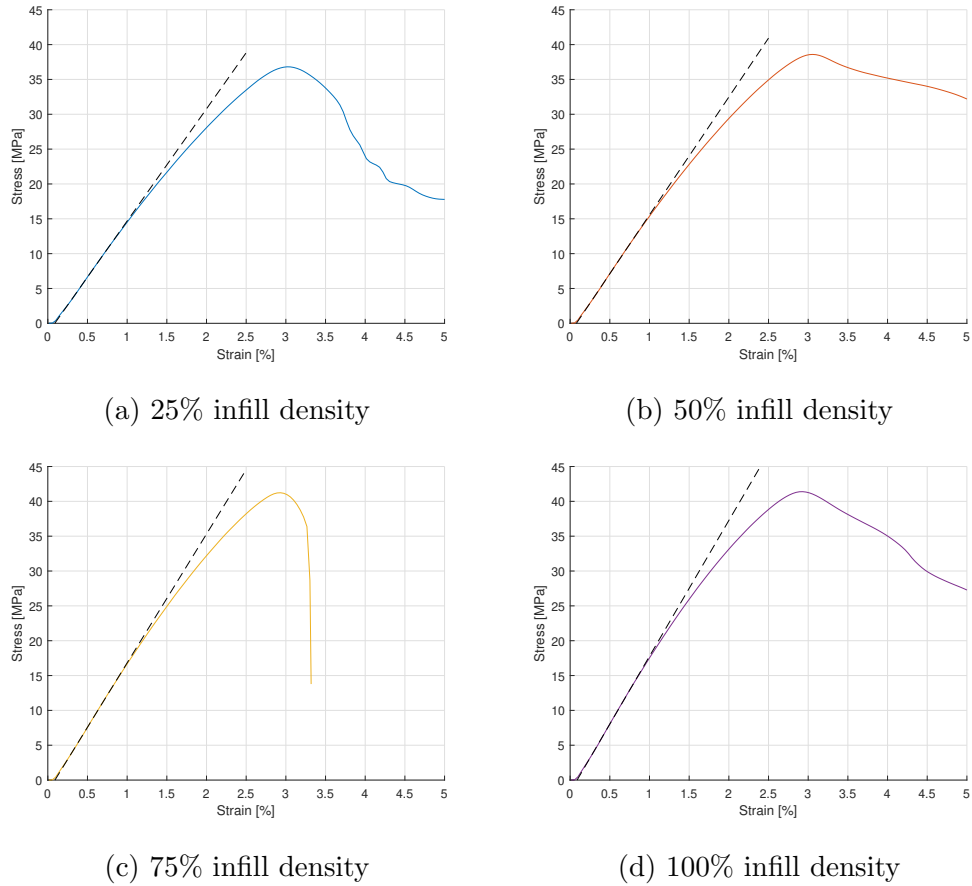


Figure 2.29: Stress-strain curves of PLA

and transform it into the desired data thanks to the mathematical model.

### 2.7.1 CALIBRATION WEIGHTS

The calibration process of the testing bench consists of registering the voltage differentials thrown by all the six load cells when applying certain known weights to each one of them. Eight different aluminum and carbon fiber pieces were used as weights for the calibration of the load cells (see Figure 2.30). These pieces are referred to as calibration weights for convenience. The calibration weights data was obtained using an OHAUS<sup>®</sup> Pioneer<sup>™</sup>precision scale and a METTLER TOLEDO<sup>®</sup> balance. Figure 2.31 shows both balances used for the calibration weights registration and

Table 2.6 presents the results of all the measurements.

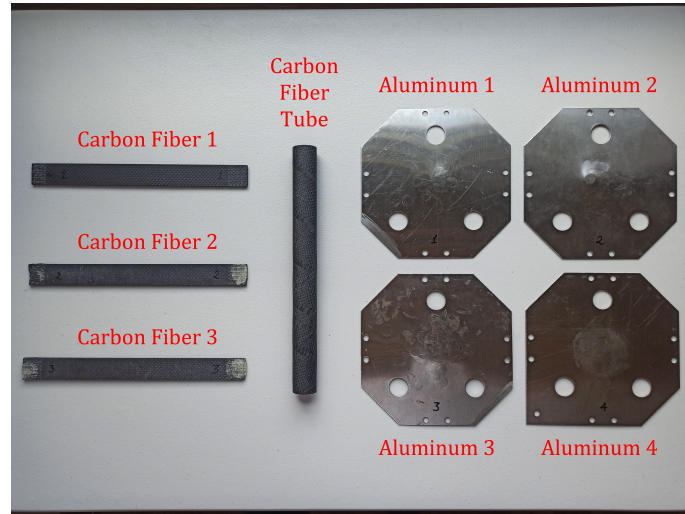


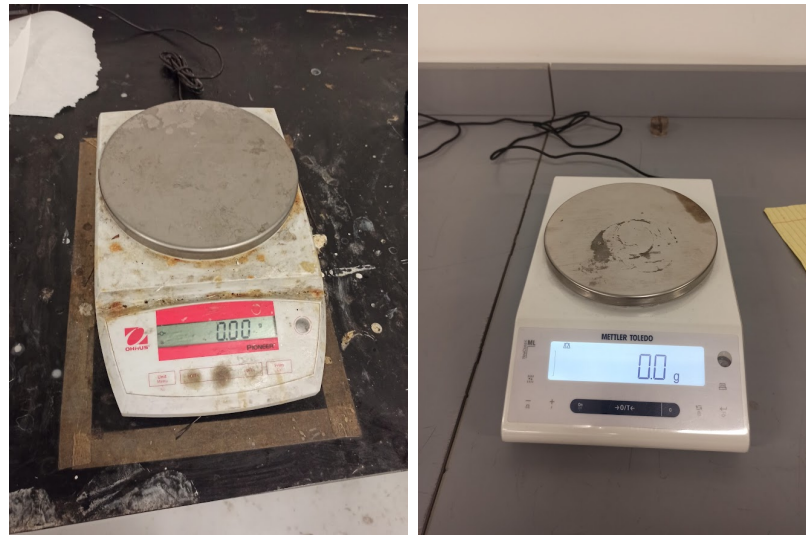
Figure 2.30: Testing bench calibration weights

Table 2.6: Calibration weights measurements data

Element	OHAUS PIONEER [gf]	METTLER TOLEDO [gf]	Average [gf]
Carbon Fiber 1	35.73	35.7	35.715
Carbon Fiber 2	36.63	36.6	36.615
Carbon Fiber 3	36.2	36.1	36.15
Carbon Fiber Tube	343.445	-	343.445
Aluminum 1	249.75	249.3	249.525
Aluminum 2	251.21	250.7	250.955
Aluminum 3	250.805	250.3	250.5525
Aluminum 4	255.34	254.8	255.07

## 2.7.2 LOAD CELLS CALIBRATION

Since the testing bench load cells have a permissible weight limit of 1 kgf, a combination of the calibration weights was carried out in order to vary the weight in the range from 0 gf to 1000 gf when calibrating each load cell separately. Figure 2.32a shows the load cells clamping so that they stay in a cantilever way. Figure 2.32b shows a 3D-printed weight-bearing surface attached to the load cell so that the

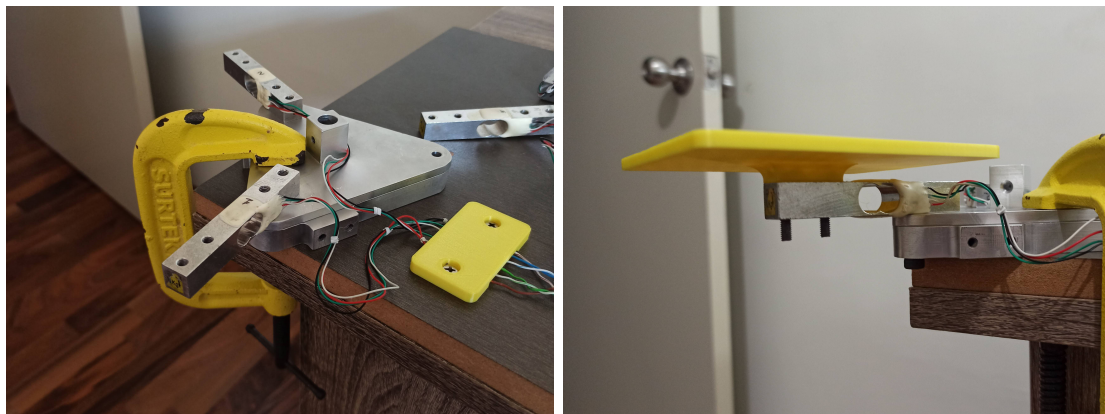


(a) OHAUS® Pioneer™

(b) METTLER TOLEDO®

Figure 2.31: Balances used to measure the calibration weights

weighting process is easier to carry out. Figure 2.33 shows some of the calibration weights being measured by the load cells in the calibration process.



(a) Load cells clamping

(b) Load cell weight bearing surface

Figure 2.32: Load cells calibration scenario

Table 2.7 presents the results of each combination of the calibration weights measurements, where all the six load cells were tested and all their respective DV were registered. For each combination and load cell, there were carried out three tests to ensure repeatability in measurements.





(a) Measuring Carbon Fiber 1, 2, and 3

(b) Measuring Aluminum 1 and 2

Figure 2.33: Load cells calibration process using calibration weights

With all the information from the calibration weights combinations and their respective load cells DV, there could be observed a linear relationship between the calibration weights magnitudes and the load cells DV. For this reason, a least-squares linear regression was proposed in order to create six models (one for each load cell) so that each model uses the load cell DV as input data and outputs the actual force experienced by the load cell. The equations of the mathematical model are obtained from reference [60] and presented below (consider  $\sum = \sum_{i=1}^n$ ).

$$y = a_0 + a_1x \quad (2.10)$$

$$a_1 = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{n \sum x_i^2 - (\sum x_i)^2} \quad (2.11)$$

$$a_0 = \bar{y} - a_1 \bar{x} \quad (2.12)$$

$$r = \frac{n \sum x_i y_i - (\sum x_i) (\sum y_i)}{\sqrt{n \sum x_i^2 - (\sum x_i)^2} \sqrt{n \sum y_i^2 - (\sum y_i)^2}} \quad (2.13)$$

Equation 2.10 represents the mathematical expression for the straight line, where  $a_0$  and  $a_1$  are coefficients representing the intercept and the slope, respectively

Table 2.7: Calibration weights combinations and load cells DV

Element	Weight [gf]	Load cells DV [ $\times 10^{-5}$ V]					
		Cell 1	Cell 2	Cell 3	Cell 4	Cell 5	Cell 6
Carbon Fiber 1	35.715	0.9019	0.9689	0.8713	0.8939	0.8389	0.9015
Carbon Fiber 1 Carbon Fiber 2	72.33	1.8199	1.9451	1.7646	1.8101	1.6966	1.8286
Carbon Fiber 1 Carbon Fiber 2 Carbon Fiber 3	108.48	2.7299	2.9194	2.6510	2.7182	2.5499	2.7450
Aluminum 4	255.07	6.4146	6.8518	6.2363	6.3881	6.0094	6.4659
Carbon Fiber Tube	343.445	8.6429	9.2315	8.3877	8.5980	8.0798	8.6953
Aluminum 1 Aluminum 2	500.48	12.5890	13.4433	12.2377	12.5358	11.7841	12.6807
Aluminum 1 Aluminum 2 Aluminum 3	751.0325	18.9003	20.1638	18.3632	18.8121	17.6844	19.0292
Aluminum 3 Aluminum 4 Carbon Fiber 1 Carbon Fiber 2 Carbon Fiber 3 Carbon Fiber Tube	957.5475	24.0822	25.7006	23.4074	23.9806	22.5394	24.2537

(Equations 2.12 and 2.11) and  $r$  is the correlation coefficient (Equation 2.13) which evaluates how well the model fits the data. For a perfect fit,  $r = 1$ .

The least-squares linear regression model was applied with all the data presented in Table 2.7 (load cells DV and their respective weight force data) so that, the coefficients  $a_0$  and  $a_1$  could be computed for each load cell. These coefficients are presented in Table 2.8.

The load cell calibration process is complete with the linear model coefficients obtained for each load cell. The coefficients are key to the functioning of the whole

Table 2.8: Coefficients  $a_0$  and  $a_1$  for load cells linear regression models

Coefficient	Cell 1	Cell 2	Cell 3	Cell 4	Cell 5	Cell 6
$a_0$	-0.0829	-0.3379	0.1014	0.0289	0.1096	0.0771
$a_1$	39.7571	37.2649	40.9	39.9259	42.4717	39.4705

testing bench and they might change according to the atmospheric pressure.

### 2.7.3 TESTING BENCH SOFTWARE

There were created several scripts in the whole testing bench project, however, only the most relevant ones are considered in this subsection. These are the load cells calibration and main scripts.

The load cells calibration script involves the load cells calibration process as presented in the previous subsection. The script is simple and it only consists of measuring the DV from a load cell when a calibration weight is placed over it. Figure 2.34 shows the load cells calibration script when running. Note that a calibration weight was placed over a load cell shortly after sample 50. The black curve indicates the raw data from the load cell and the red one is the filtered data. The load cells DV data is filtered using a low-pass filter from reference [61]. Once the total required samples have been obtained, an average of the filtered measurements is made from the sample where the filtered data is stabilized until the last sample. Overall, a total of 200 samples are considered for measuring the load cell DV, generally from sample 200 until sample 400. The script ends by showing the average of the load cell DV, meaning that such load cell DV corresponds to the calibration weight placed to measure. This process was carried out with all six load cells and all eight combinations of calibration weights. In this way, Table 2.7 was filled.

On the other hand, the testing bench main script is the one used in the operation of the testing bench. The testing bench operation is graphically represented

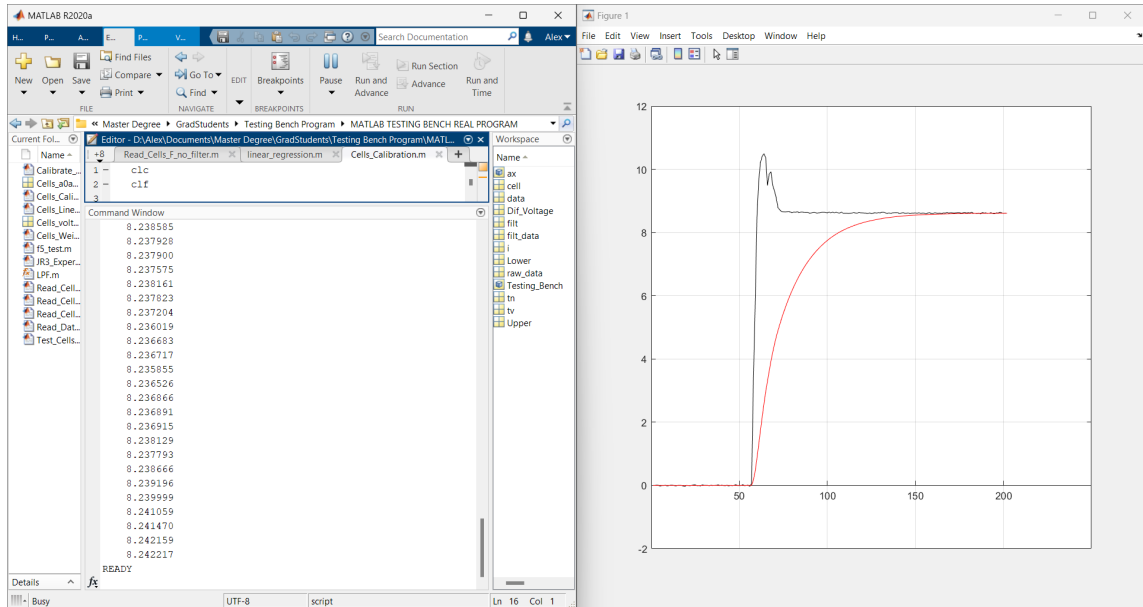


Figure 2.34: Testing bench software: load cells calibration

in the flowchart from Figure 2.35. It starts with the serial port communication with the Arduino<sup>®</sup> UNO, then the software receives 10 data samples from the load cells to check communication and let possible voltage spikes dissipate. Next, the software reads data and filters it in real-time using the same low-pass filter as in the load cells calibration process (see Figure 2.36). This action is repeated until the filtered data is completely steady.

Once the data is steady, 25 data samples are received and used to tare the load cells DV. At this point, the load cells are tared and ready to provide data, as shown in Figure 2.37. The software now receives the load cells DV data and then applies the linear regression models in order to convert the load cells DV into their respective forces in [gf]. Once all the forces of the load cells are known, the testing bench mathematical model is applied so that the values  $F_x$ ,  $F_y$ , and  $F_z$  in the UAV body frame  $\{B\}$  are delivered and stored, as shown in Figure 2.38.

This process is repeated until all the needed samples are already received; if not, the process repeats from receiving data followed by applying the linear regression models, and so on. If all the needed samples are already received, the testing bench

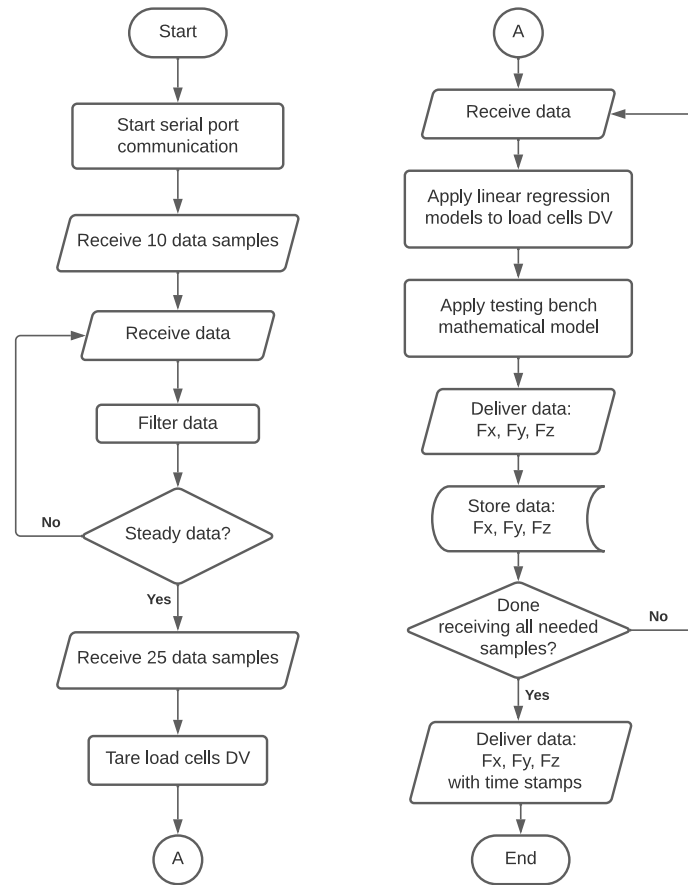


Figure 2.35: Testing bench operation flowchart

software ends with delivering the stored data  $F_x$ ,  $F_y$ , and  $F_z$  with time stamps which are useful for knowing the exact time where the forces were exerted on the testing bench.

Figures 2.39 and 2.40 show all the registered data with time stamps in the MATLAB<sup>®</sup> interface and once the data is exported, respectively. In this way, the low-cost UAV testing bench software ends its operation.

## 2.8 STATIC AND DYNAMIC TESTS

Static and dynamic tests are carried out so that the low-cost UAV testing bench functioning can be validated according to the results provided by itself and using

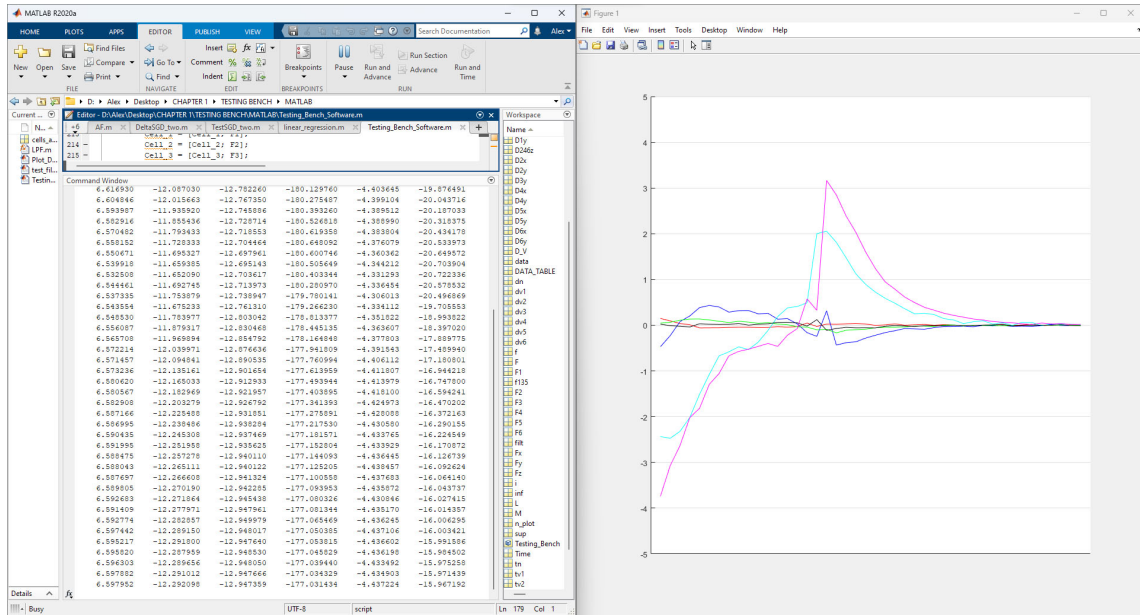


Figure 2.36: Testing bench software: reading load cells DV data and waiting for stabilization

existing high-cost sensors as well. The results are presented taking into account both AI-6061 and PLA testing benches. Finally, precision and accuracy results are shown to give a final general approval to the testing bench.

### 2.8.1 STATIC TESTS RESULTS

In order to validate the output data from the testing bench under the application of static loads in the  $z$  axis, a series of static tests were carried out. These static tests consisted of measuring the already known calibration weights according to the combinations from Table 2.7. The tests were performed with the AI-6061 and PLA testing benches. In addition, a force-torque sensor from ©JR3 multi-axis load cell systems was used for reference and comparison of results. Figure 2.41 shows all three platforms carrying out the static force tests over the  $z$  axis.

The results from all the static force measurements over the  $z$  axis with the three platforms are presented in Figure 2.42a, where the outputs are compared to

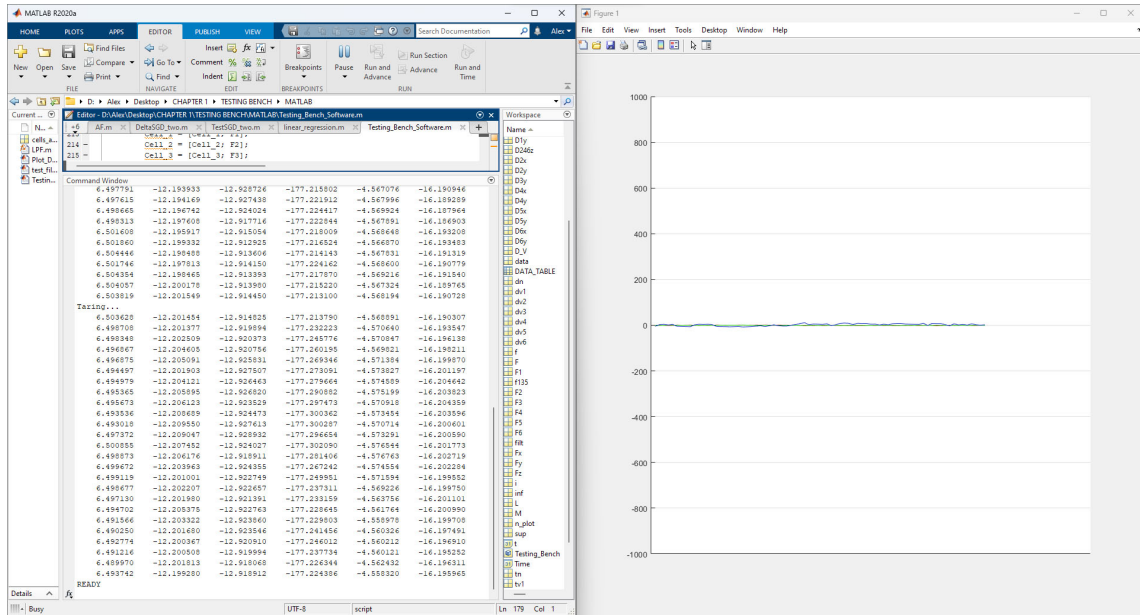


Figure 2.37: Testing bench software: load cells DV data stabilized and taring process completed

the target forces. Note that all three platforms show good performance since they all tend to match the target force, nevertheless, as the force increases the PLA measurements tend to deviate slightly from the target force. On the other hand, Figure 2.42b shows the error force when varying the applied force on the platforms. The error from the JR3 force-torque sensor varies stochastically at lower forces, but tends to decrease at higher forces while Al-6061 and PLA testing benches errors tend to increase as the force increases; this to a lesser extent with the Al-6061 testing bench.

Overall, the error delivered by the PLA testing bench is greater than the Al-6061 when applying static forces and both testing bench errors present an increasing behavior as the applied force increases over the  $z$  axis.

On the other hand, there were also carried out static tests on the testing bench  $x$  and  $y$  axes, as shown in Figures 2.43 and 2.44 respectively. These tests were performed only considering the Al-6061 testing bench. The results of such tests are presented in Figure 2.45, where all three graphs show good performance in

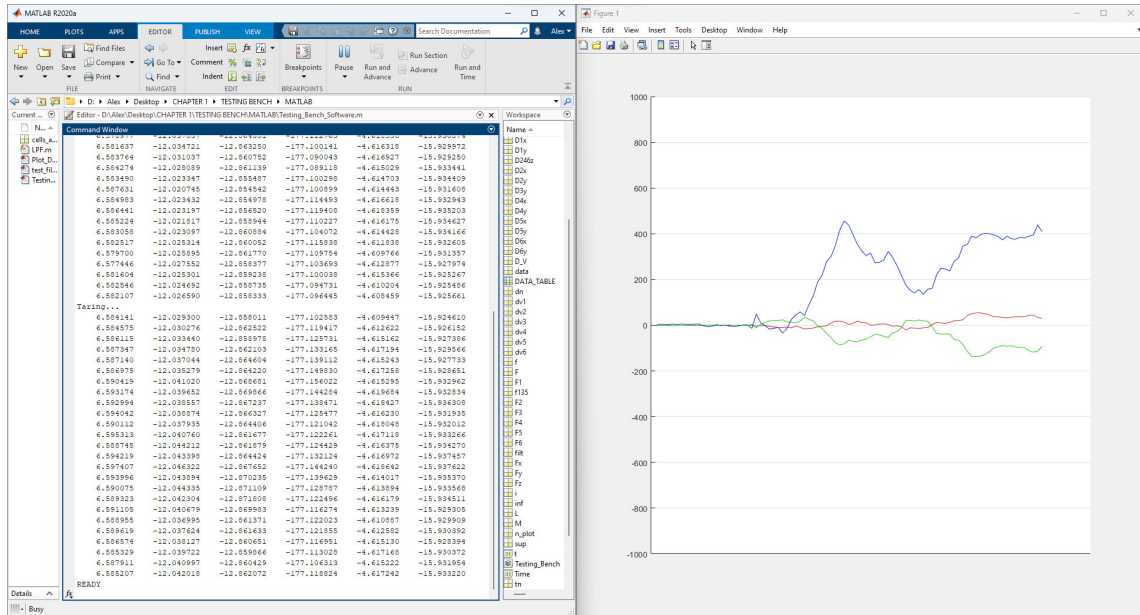


Figure 2.38: Testing bench software: reading forces  $F_x$ ,  $F_y$ , and  $F_z$  in UAV body frame  $\{B\}$

the measurements, according to the expected results. Note that 10 measurements were made for each calibration weight combination. Moreover, linear regression was applied on each axis results and their respective correlation coefficients  $r$  are presented, showing an adequate correlation between the measurements.

## 2.8.2 DYNAMIC TESTS RESULTS

Dynamic forces are essential in the validation of the low-cost UAV testing bench since the latter is actually made to measure forces exerted by UAVs, which are evidently dynamic types. Therefore, dynamic force tests were carried out with the Al-6061 and PLA testing benches, as well as with the JR3 force-torque sensor. The dynamic forces were generated by a DJI® Tello UAV which was fixed on the three platforms in separate tests, just as Figure 2.46 illustrates. These dynamic tests only considered the  $z$  axis for validation. The results from the dynamic tests are presented in Figure 2.47. Note that three different measurement sets were carried out for repeatability



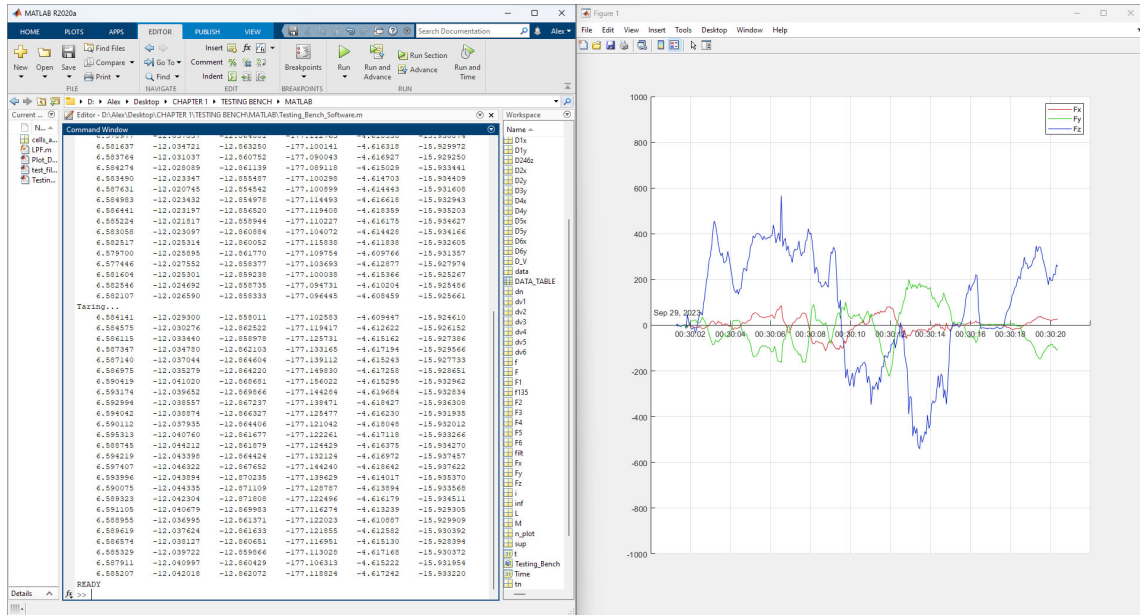


Figure 2.39: Testing bench software: all read forces  $F_x$ ,  $F_y$ , and  $F_z$  in UAV body frame  $\{B\}$

purposes. In all the measurement sets the JR3 force-torque sensor shows a large amount of noise so it is difficult to compare with the other results, even though all data was filtered. Meanwhile, Al-6061 and PLA testing bench results show similar maximum force exertion over the  $z$  axis. In this case, the PLA testing bench shows slightly higher force exertion than the Al-6061. This latter result confirms the stress results in the numerical analysis from the FEM simulations, where the PLA testing bench showed a greater amount of stress in the narrow sections of the load cells, meaning that they are in fact experiencing greater stress and therefore throwing higher values of exertion forces.

In this way, the dynamic tests are concluded and the low-cost UAV testing bench is validated under the established operating conditions.

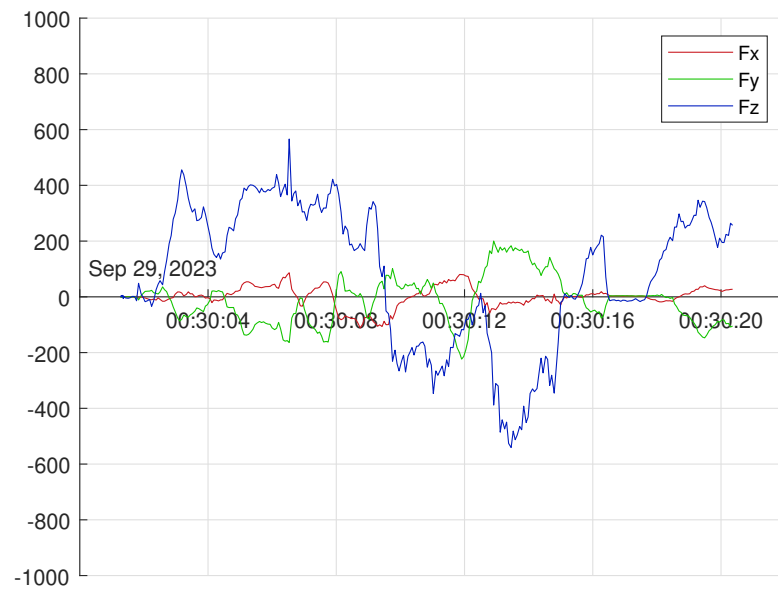
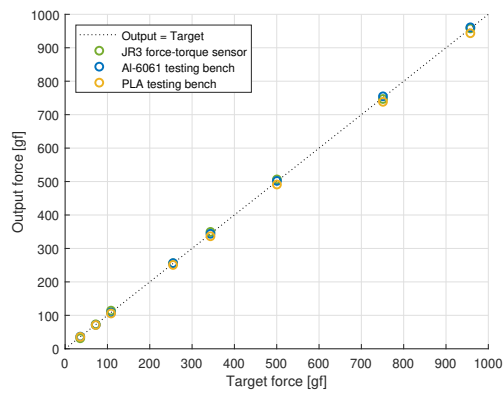


Figure 2.40: All exported forces  $F_x$ ,  $F_y$ , and  $F_z$  read in UAV body frame  $\{B\}$

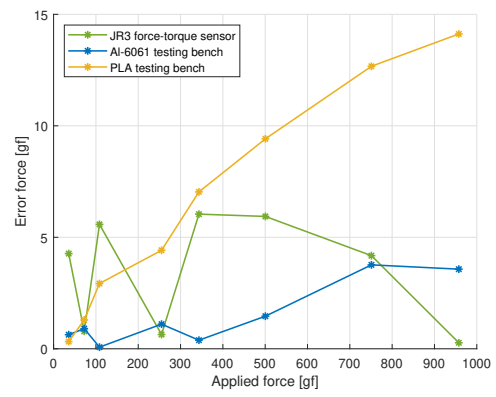


(a) Al-6061 testing bench (b) PLA testing bench (c) JR3 force-torque sensor

Figure 2.41: Static force tests with different platforms using calibration weights over  $z$  axis



(a) Output force vs target force



(b) Error force vs applied force

Figure 2.42: Static force tests results over  $z$  axis

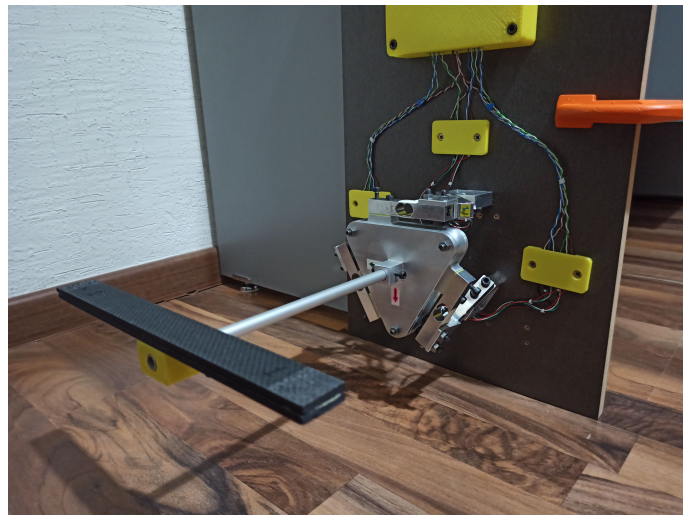


Figure 2.43: Testing bench static force tests using calibration weights over  $x$  axis

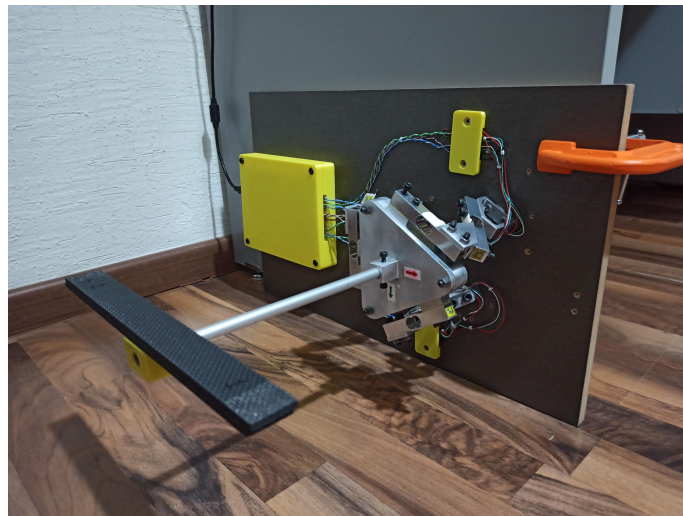


Figure 2.44: Testing bench static force tests using calibration weights over  $y$  axis

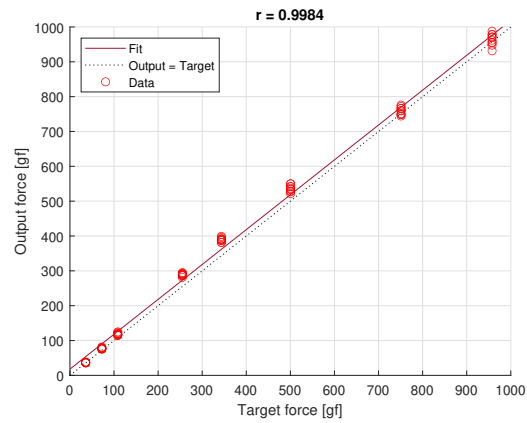
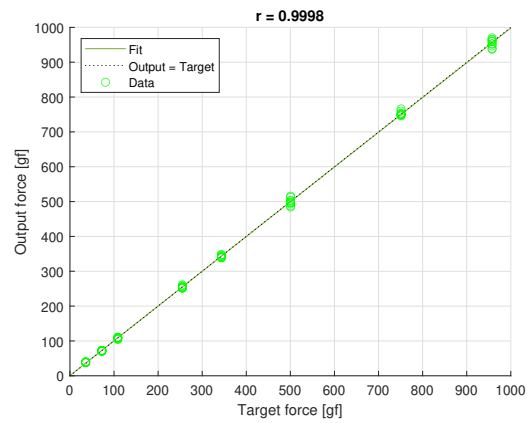
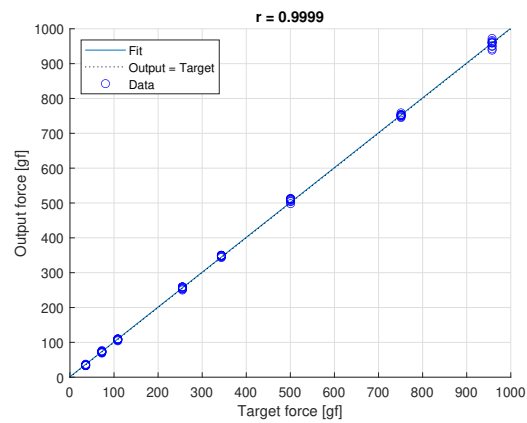
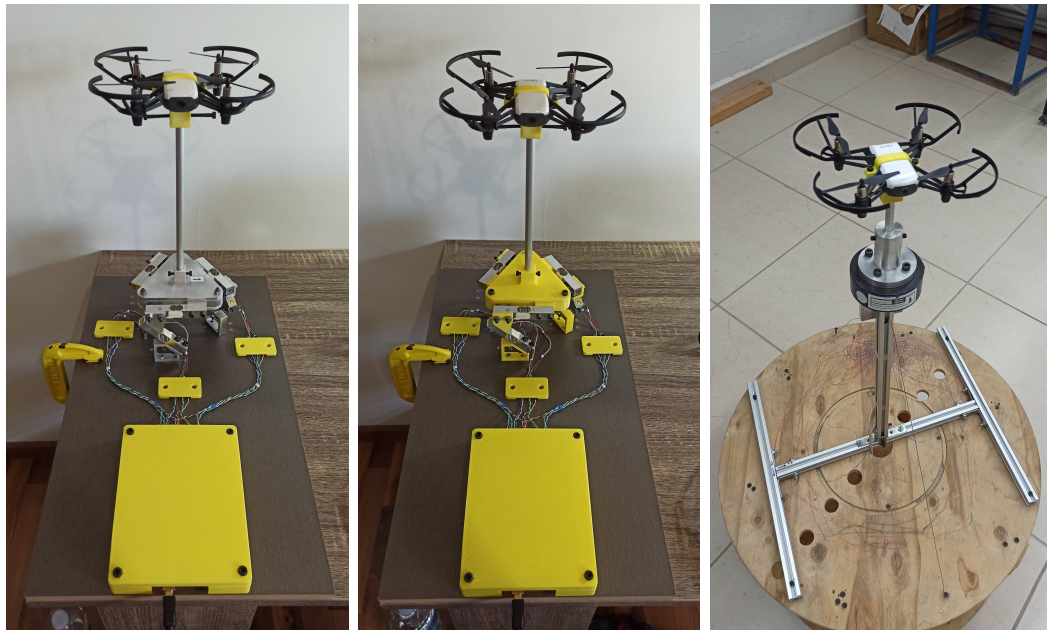
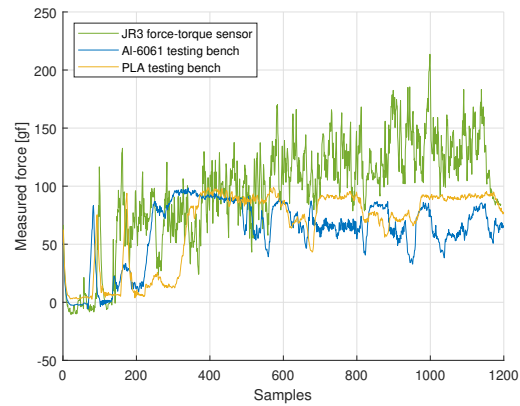
(a) Output force vs target force over  $x$  axis(b) Output force vs target force over  $y$  axis(c) Output force vs target force over  $z$  axis

Figure 2.45: Static force tests results over all axis and correlation coefficients

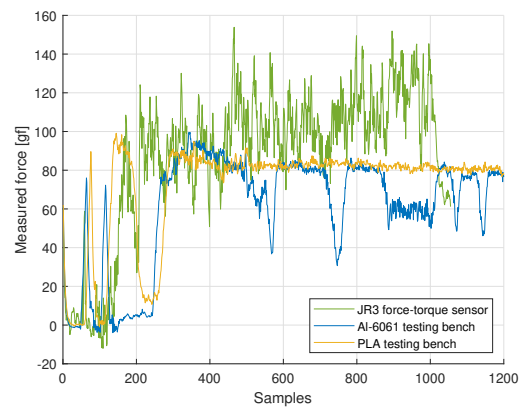


(a) Al-6061 testing bench    (b) PLA testing bench    (c) JR3 force-torque sensor

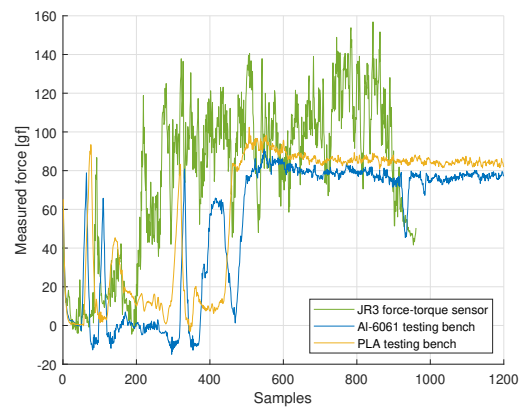
Figure 2.46: Dynamic force tests with different platforms using a DJI® Tello UAV



(a) First set of tests



(b) Second set of tests



(c) Third set of tests

Figure 2.47: Dynamic force tests results over  $z$  axis

## CHAPTER 3

# QUADROTOR DESIGN AND CONSTRUCTION

---

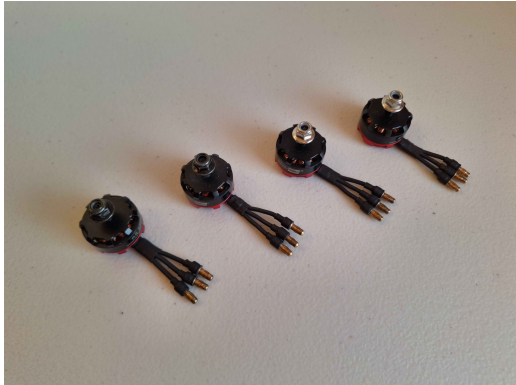
The design and construction of an actual quadrotor with its four independent and completely controlled motors are addressed. All the needed components were obtained and assembled together to finally have a quadrotor entirely created from scratch and able to control its four motors independently. In the next sections, all the design and construction process of the quadrotor is presented, as well as its general operation and the experimental tests that were carried out to later employ the data in the system identification process.

### 3.1 AVIONICS

As every UAV, a quadrotor possesses electronic components, some more essential than others depending on the complexity of the vehicle. The electronic components that were chosen to create the quadrotor presented in this work are listed in Table 3.1.

The components from Table 3.1 were acquired (except for the frame which is designed from scratch) and these are presented in Figure 3.1.

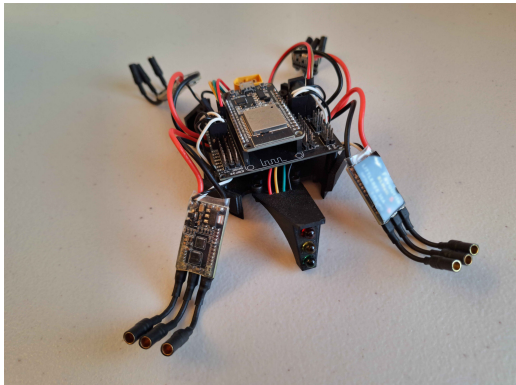




(a) Motors



(b) Battery



(c) Electronics misc.



(d) Propellers

Figure 3.1: Quadrotor components

Table 3.1: Selected electronic components for the quadrotor

Component	Functioning
4 Brushless motors 2300 KV	Machine that converts electrical energy into mechanical energy without using brushes
4 Electronic speed controllers 30A	Electronic circuit that controls the speed of an electric motor
LiPo battery 4 cells	Rechargeable battery of lithium-ion
Power distribution board 12V	Distributes the energy from the battery to all other parts of the quadrotor
ESP32	Integrated circuit used to record instructions written in a programming language, it also provides Wi-Fi and Bluetooth connectivity for embedded systems
ESP32 shield	Expansion board for ESP32
3 LEDs	Semiconductor device that emits light when an electric current is passed through it
4 Propellers 3-blade 5040	Device with three blades that spin around to produce a force
Frame	Suit of armor for all the electronic components that constitute the quadrotor

## 3.2 DESIGN

The quadrotor is designed to be a First Person View (FPV) type. In the next subsections, the conceptual, preliminary, and detailed design of the quadrotor are presented.

### 3.2.1 CONCEPTUAL DESIGN

As a starting point for the design, the quadrotor is proposed as an X configuration. The conceptual design is illustrated in Figure 3.2. Note that it is shown from a top view and heading to the north. All its four motors are numbered in the direction of quadrants and their spin direction is proposed as shown. Moreover, notice that the distance between the motors is 250mm, a value that was determined as a function of the size of the motors and the type of quadrotor.

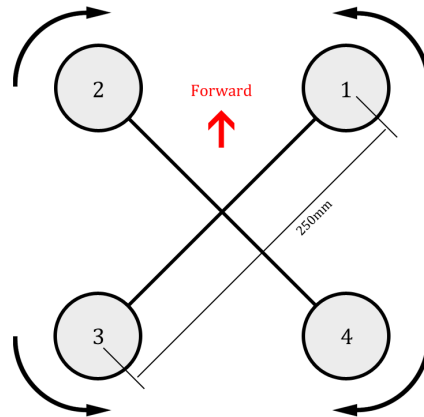


Figure 3.2: Quadrotor conceptual design

### 3.2.2 PRELIMINARY DESIGN

The preliminary design phase consisted of designing different possible frames. In order to do this, the parts of all the quadrotor components were created by means of Computer-Aided Design (CAD) software Solidworks® (see Figure 3.3). This is to consider sizes and facilitate the arrangement of all components over the frame.

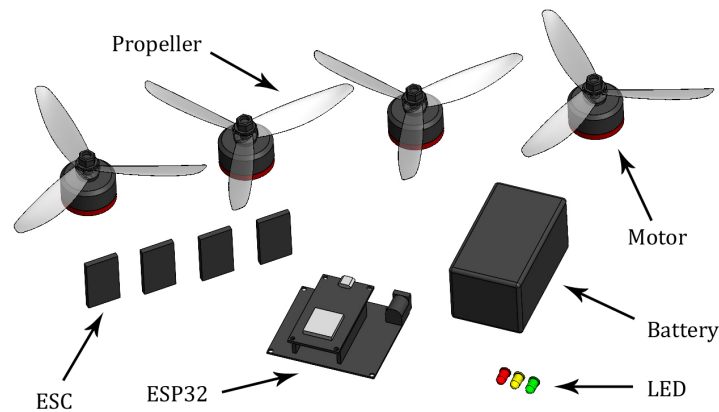


Figure 3.3: Quadrotor CAD components

Once all the CAD components were created, iterations of the preliminary design began. It started with the design of a frame and then the components were arranged in different ways. The goal was to come up with a design that takes up as little space as possible and with all the components correctly positioned. Figure 3.4

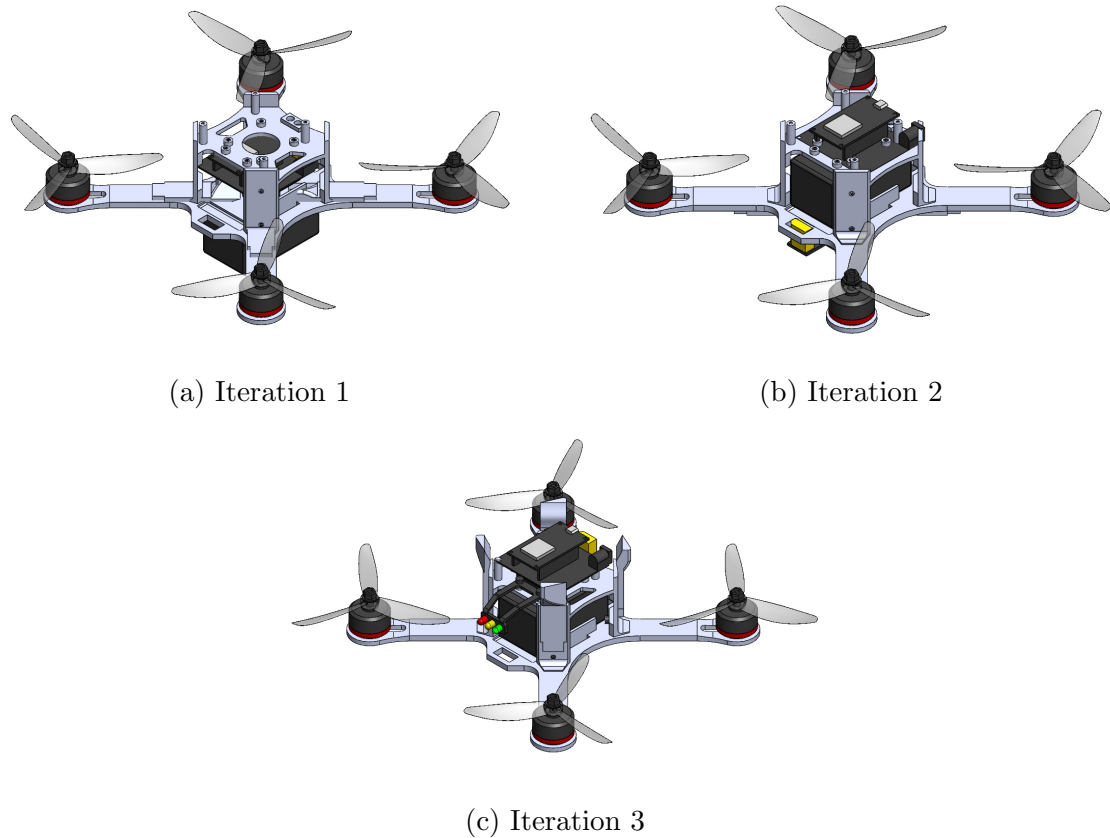


Figure 3.4: Quadrotor preliminary design phase

shows some of the iterations carried out in the preliminary design phase.

### 3.2.3 DETAILED DESIGN

After selecting the more suitable arrangement for the quadrotor components and designing its frame, the detailed design was reached (see Figure 3.5). The quadrotor design includes a reinforced frame to prevent buckling; propeller protectors; a base in the center to hold all electronics and a head to hold the LED.



Figure 3.5: Quadrotor detailed design

### 3.3 CONSTRUCTION

Once all the components were acquired and the design was stated, the construction of the quadrotor was carried out. The frame was created by means of additive manufacturing using a Creality<sup>®</sup> Ender-3 3D printer employing PLA as the material with 100% infill density. Figure 3.6 shows this procedure.

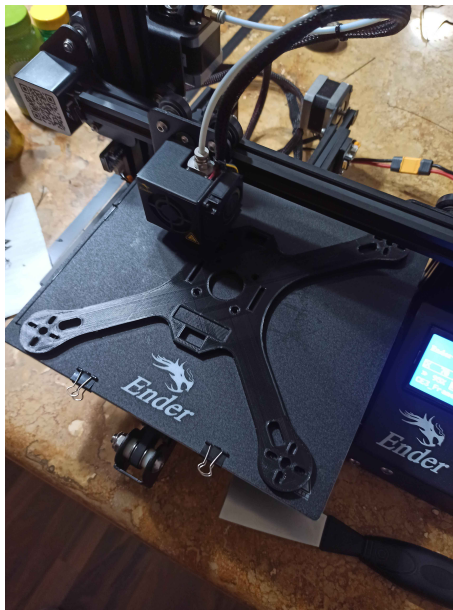


Figure 3.6: Printing quadrotor frame

The same strategy was carried out for the propeller protectors, the center base, and the LED head. The quadrotor frame with its 3D-printed elements assembled is shown in Figure 3.7.



Figure 3.7: Quadrotor frame

Later, all the electronics and remaining components were installed into the frame, thus having the complete constructed quadrotor (see Figure 3.8).

### 3.4 OPERATION AND CONNECTIVITY

Recalling the main objective of this work, which is to identify the mathematical model of a quadrotor, the operation of the constructed quadrotor is planned to be simple since it will not have to take off and fly, reason why an IMU (Inertial Measurement Unit) was not included in the components. Therefore, in this work, the operation of the quadrotor is simply to activate the motors in certain sequences to recreate its dynamics: roll, pitch, yaw, hover, and bounce. In the same way, to recreate its displacements: forward, backward, left, and right at different angles of inclination. Figures 3.9 and 3.10 illustrate the quadrotor dynamics and the motors spin intensity, respectively.

In order to recreate the dynamics, the motors had to be properly calibrated according to the ESC protocols and a wireless connection had to be established.



(a) Top view



(b) Isometric view



(c) Front view



(d) Side view

Figure 3.8: The quadrotor

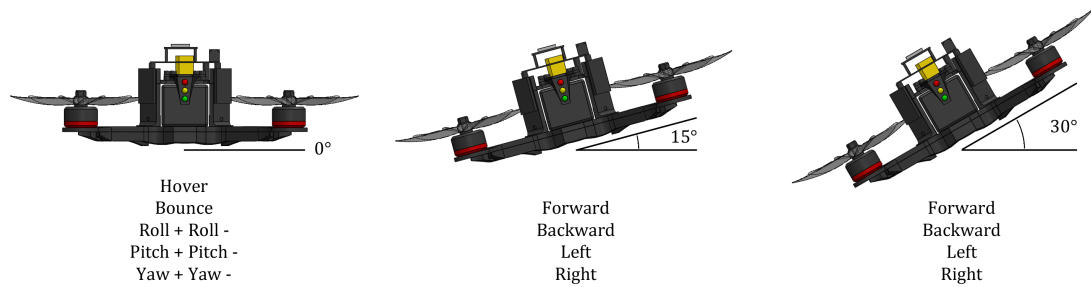


Figure 3.9: Quadrotor dynamics

The quadrotor commands for the motors activation sequences were programmed in Arduino<sup>®</sup> and then downloaded to the ESP32. The connection between the ESP32 and the computer is via Bluetooth and the MATLAB<sup>®</sup> software is used for this process. For simplicity, the Arduino<sup>®</sup> and MATLAB<sup>®</sup> codes are represented as pseudo codes, where Algorithm 1 is the Arduino<sup>®</sup> procedure for the activation of the quadrotor motors and Algorithm 2 is the MATLAB<sup>®</sup> procedure for the communication with the quadrotors ESP32.

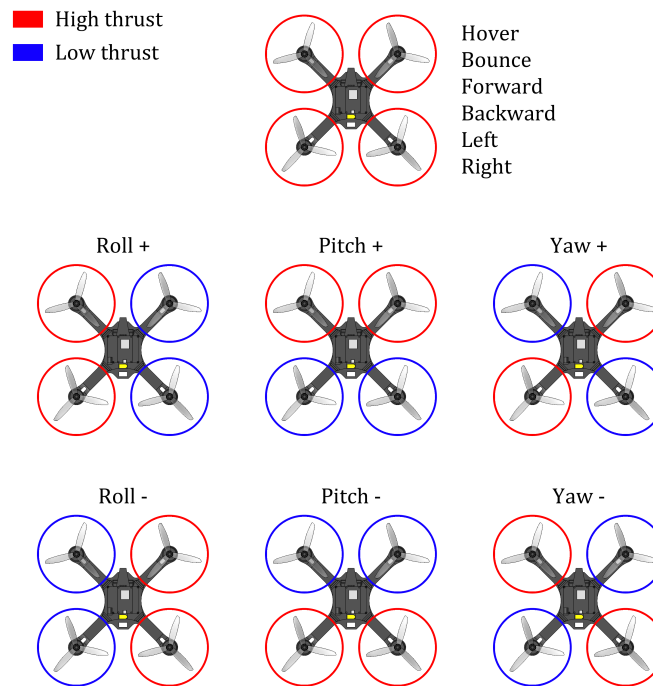


Figure 3.10: Quadrotor dynamics according to their motors spin intensity



---

**Algorithm 1:** Arduino<sup>®</sup> procedure for the activation of motors

---

**Result:** Activate motors and send data to MATLAB<sup>®</sup> software

```
while Begin is true do
    turn on all LEDs;
    set Servo Motors to initial PWM value;
    set Begin to false;
end

turn on the red LED;

if Bluetooth data is available then
    read the incoming character;
    process the character and set PWM and Run motors sequences
    accordingly;
    while Run is true do
        turn on the yellow LED;
        set Servo Motors and send data based on PWM values;
        for i from 1 to PWM do
            set Servo Motors and send data based on the current PWM
            value;
        end
        turn on the yellow LED;
        for i from PWM to 0 do
            set Servo Motors and send data based on the current PWM
            value;
        end
        terminate data transmission;
        turn on the red LED;
        set Run to false;
    end
end

end
```

---

**Algorithm 2:** MATLAB<sup>®</sup> procedure for receiving motors data**Result:** Receive, plot, and export motors data from Arduino<sup>®</sup>**while** *Start is true* **do**

establish Bluetooth connection with ESP32;

    Start  $\leftarrow$  false;**end**

prompt user for experiment number;

prompt user for command;

send command to ESP32;

read data from ESP32;

process and store data in arrays  $Y$ ,  $D$ , and  $T$ ;

create subplots for each motor;

plot data for each motor;

customize plot appearance;

create a table (OUTPUT) using time, motor 1, motor 2, motor 3, motor 4;

generate a filename based on experiment number;

save the table (OUTPUT) to a CSV file with the generated filename;

A simplified way to represent the communication between the quadrotor and the computer is shown in Figure 3.11.

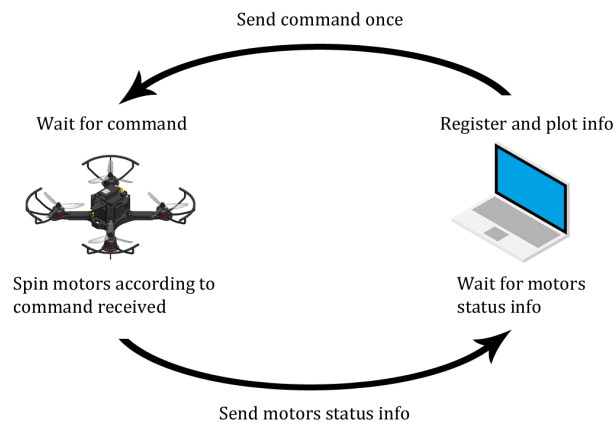


Figure 3.11: Communication process between computer and quadrotor

## 3.5 EXPERIMENTAL TESTS

In this section the experimentation with the quadrotor is carried out. A six degree of freedom force-torque sensor was needed for data measurement and an experimental bench had to be designed and built in order to attach the quadrotor to the force-torque sensor in an area raised to the ground to avoid ground effect caused by the propellers when spinning. All the proposed dynamics were performed by the quadrotor, showing good results to be used in the system identification process.

### 3.5.1 FORCE-TORQUE SENSOR

Although in Chapter 1 a testing bench was developed to measure force and torque data from a UAV, as discussed, the testing bench cannot provide data from all the degrees of freedom needed for the quadrotor system identification. Therefore, a six-degree-of-freedom force-torque sensor had to be considered to successfully carry out the experimental tests.

A commercial force-torque sensor is considered in this work for the quadrotor experimental tests (see Figure 3.12). It is a miniature digital force and torque sensor which measures in three orthogonal axes. The six-axis force-torque sensor comes with a conversion board, an evaluation board, and an interconnecting cable as shown in Figure 3.13, elements which are essential for its operation. The outline specifications of this sensor are presented in Table 3.2.

The miniature force-torque sensor also comes with a friendly interface for its operation (see Figure 3.14). One can set the communication port; power on/off the sensor; initialize according to the established parameters such as sampling interval, restart times, measuring times, and plotting limits; start and stop reading data; offset data; and save log. Once initialized and the start button has been pressed, the sensor will immediately start reading data according to the parameters previously

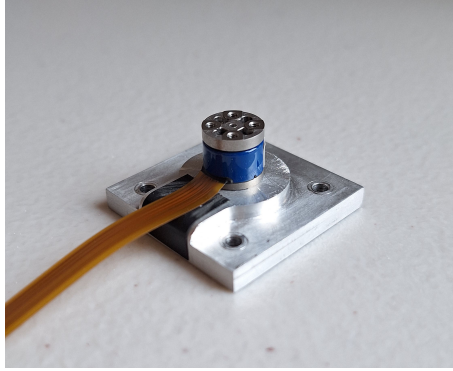


Figure 3.12: Miniature force-torque sensor

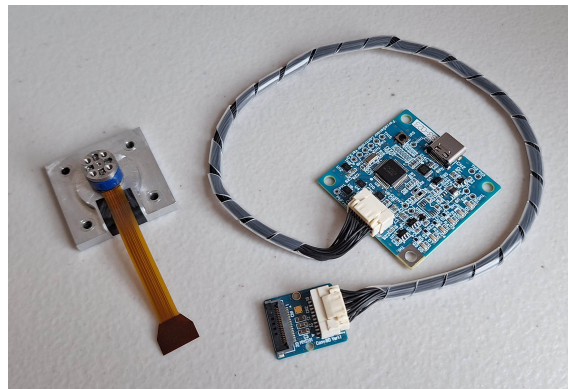


Figure 3.13: Miniature force-torque sensor evaluation kit

established and the force and moment graphs will begin to show the data being read by the sensor according to each axis.

### 3.5.2 EXPERIMENTAL BENCH

In order to use the miniature force-torque sensor for the application with the quadrotor, an experimental bench had to be designed and constructed. It was proposed to be created by means of aluminum profiles, hinges, and connectors. There was also the need to design and manufacture additional pieces to hold the sensor and its components into the bench. Figure 3.15 shows the experimental bench design. Note that the miniature force-torque sensor is assembled on the experimental bench using two aluminum pieces. Additionally, two PLA pieces are considered to hold the

Table 3.2: Miniature force-torque sensor outline specifications

Size (Diameter $\times$ Height)	$\varnothing 9.6\text{mm} \times 9.0\text{mm}$
Weight	3.0g
Rated (Force, Moment)	40N, 0.4N·m
Load Capacity (Force, Moment)	200N, 1.8N·m
Resolution (Force, Moment)	0.1N, 0.001N·m
Interface	I <sup>2</sup> C, SPI
Power-supply voltage	3.3V

remaining components which are the conversion and evaluation boards.

Once all the aluminum profiles, hinges, and connectors were obtained, the assembly was carried out. In the same way, once the aluminum and PLA pieces were manufactured by means of machining and 3D-printing techniques respectively, they were assembled on the main aluminum profile as shown in Figure 3.16, having thus finished the construction of the experimental bench. Note that the two aluminum pieces were designed in order to put on and take off the miniature force-torque sensor for ease of handling. In this way, when using the experimental bench, the assembly of the sensor on the experimental bench is as shown in Figure 3.17.

### 3.5.3 QUADROTOR TESTS

In total, three large sets of tests were carried out; dynamics simulation, thrust, and torque tests. For the first set, the quadrotor had to be positioned as previously shown in Figure 3.9 and had to carry out the simulation of the different dynamics also illustrated in Figure 3.10. In order to measure the forces and moments exerted by the quadrotor while simulating its dynamics, the quadrotor had to be assembled over the miniature force-torque sensor in the experimental bench, as shown in Figure 3.18. Note that pieces had to be designed and 3D printed in order to hold the quadrotor

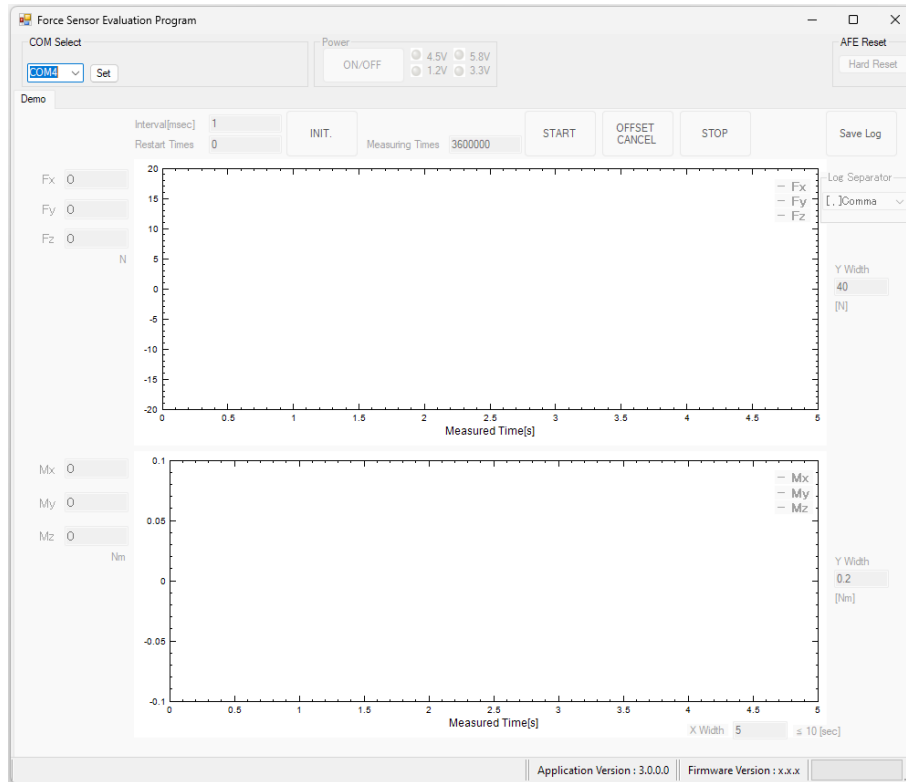


Figure 3.14: Miniature force-torque sensor software interface

with the miniature force-torque sensor at  $0^\circ$ ,  $15^\circ$ , and  $30^\circ$  degrees of inclination with respect to the ground plane.

Additionally, and for safety purposes, the quadrotor counts with three LEDs of red, yellow, and green colors, which were programmed to be activated according to the current state of the motors as shown in Figure 3.19.

The second set of tests consisted of measuring the thrust of the quadrotor propellers. Since all the propellers are equal, only the thrust measurement of the propeller of motor 1 was carried out. To do this, a motor thrust tester was obtained and the quadrotor motor 1 was assembled on it, as shown in Figure 3.20a. While carrying out the tests, the propeller thrust and RPM values were both registered. Fifty measurements were made varying the motors spin intensity from 1% to 50% of total thrust. On the other hand, the third and last set of tests consisted of measuring the propeller-induced torque. For this purpose, the miniature force-torque sensor was

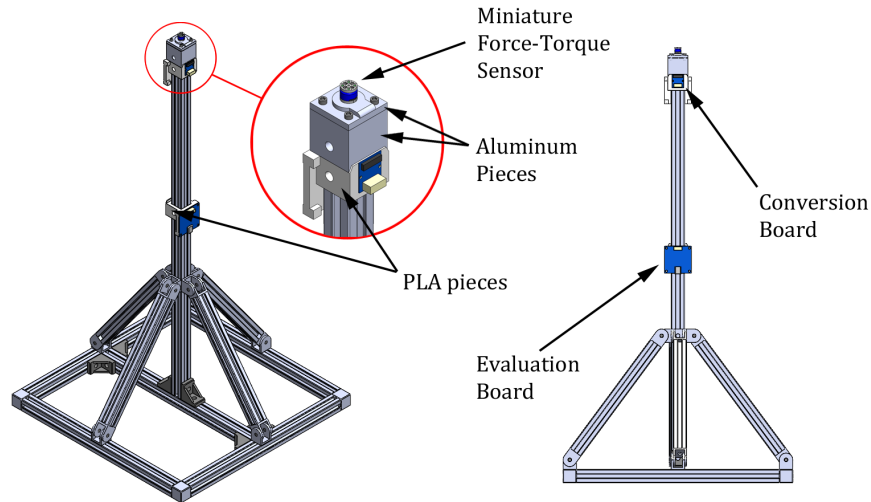


Figure 3.15: Experimental bench design

needed since the torque magnitudes are too small and precision is essential. Because the torque induced by the propeller of motor 1 is applied over all the quadrotor and its sought magnitude is with respect to the  $z$ -axis (parallel to the motors axial axis), it was fixed as shown in Figure 3.20b. Note that only motor 1 was activated so that the measured torque values over the  $z$ -axis correspond only to the propeller of motor 1. For this case, twenty measurements were made varying the motor spin intensity from 2% to 40% of total thrust.

#### 3.5.4 TESTS RESULTS

A total of 118 tests were carried out with the quadrotor, belonging 48 to the dynamics simulation, 50 to the propeller thrust, and 20 to the propeller torque. The results of the dynamics simulation tests were obtained as Figure 3.21 illustrates.

The input signals (thrust percentage of each motor) were registered as well as the output signals (quadrotor forces and moments) as shown in Figures 3.21a and 3.21b respectively. This pair of information was taken for all the 48 tests carried out

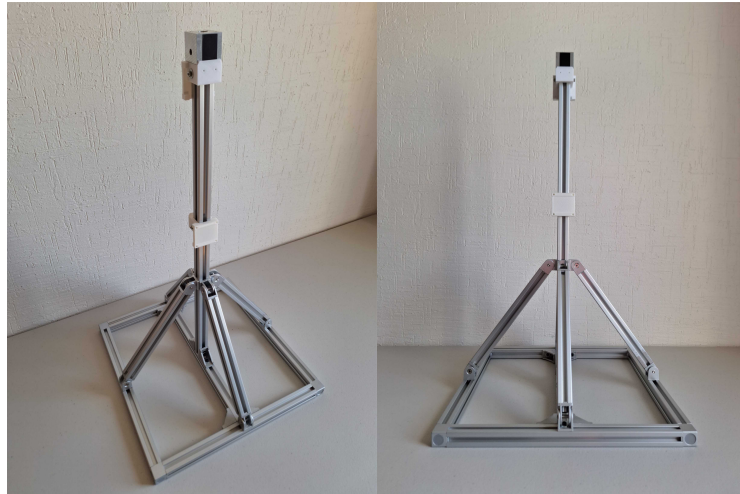


Figure 3.16: Experimental bench

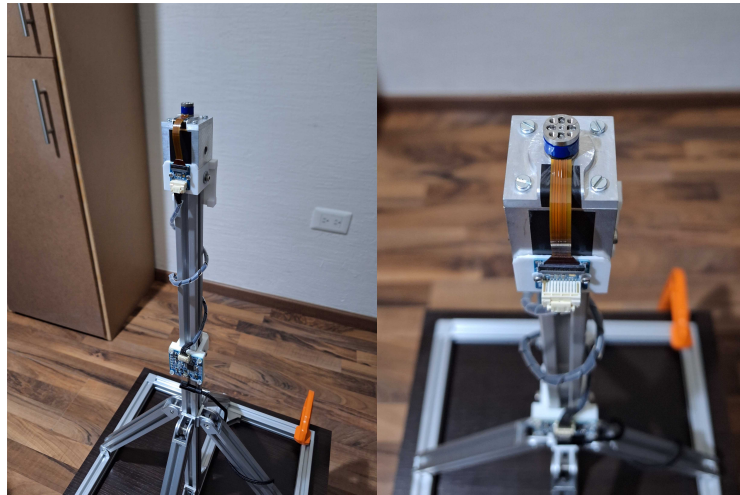
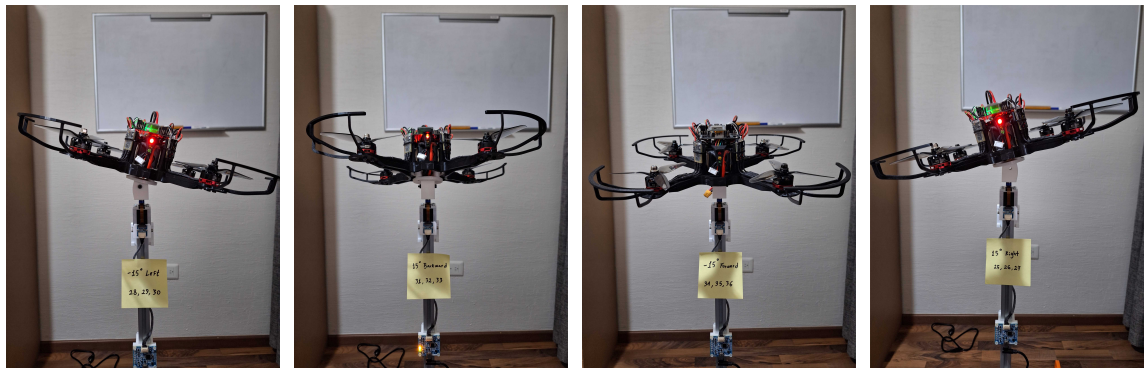


Figure 3.17: Miniature force-torque sensor assembled in experimental bench

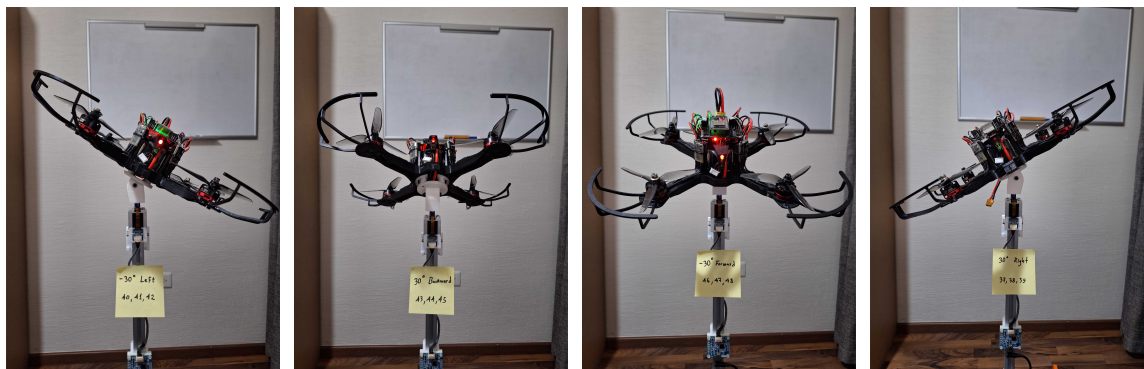
for the dynamics simulation tests. On the other hand, the results of the thrust and torque tests carried out on the quadrotor propeller are presented in Figures 3.22a and 3.22b respectively.

As seen, all of the results presented in this section were experimentally obtained and are useful for the quadrotor system identification process which is presented in the next chapter.





(a)  $-15^\circ$  roll  $\rightarrow$  left    (b)  $15^\circ$  pitch  $\rightarrow$  bwd    (c)  $-15^\circ$  pitch  $\rightarrow$  fwd    (d)  $15^\circ$  roll  $\rightarrow$  right



(e)  $-30^\circ$  roll  $\rightarrow$  left    (f)  $30^\circ$  pitch  $\rightarrow$  bwd    (g)  $-30^\circ$  pitch  $\rightarrow$  fwd    (h)  $30^\circ$  roll  $\rightarrow$  right



(i) Hover, Bounce, Yaw

Figure 3.18: Quadrotor dynamics simulation tests



(a) Red light: No spinning motors

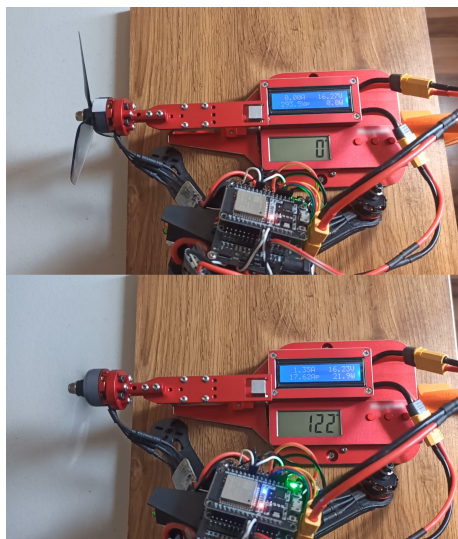


(b) Yellow light: Changing motors RPMs



(c) Green light: Constant motors RPMs

Figure 3.19: Rotation of quadrotor motors according to LED light



(a) Measuring propeller thrust



(b) Measuring propeller induced torque

Figure 3.20: Tests for the characterization of the quadrotor propellers

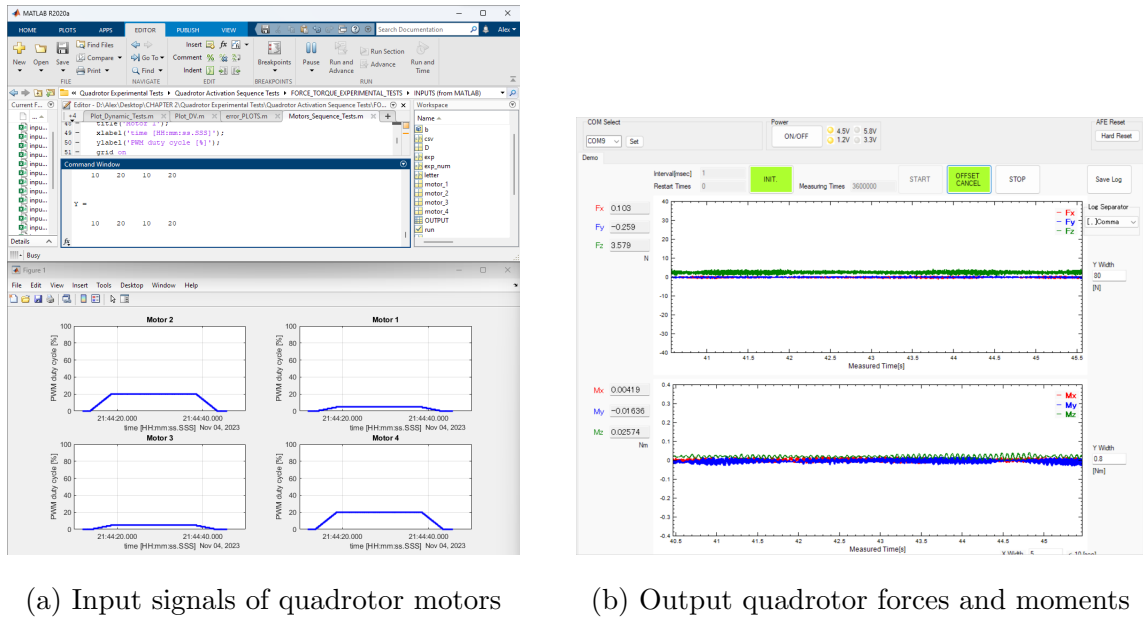


Figure 3.21: Obtaining quadrotor tests results from force-torque sensor

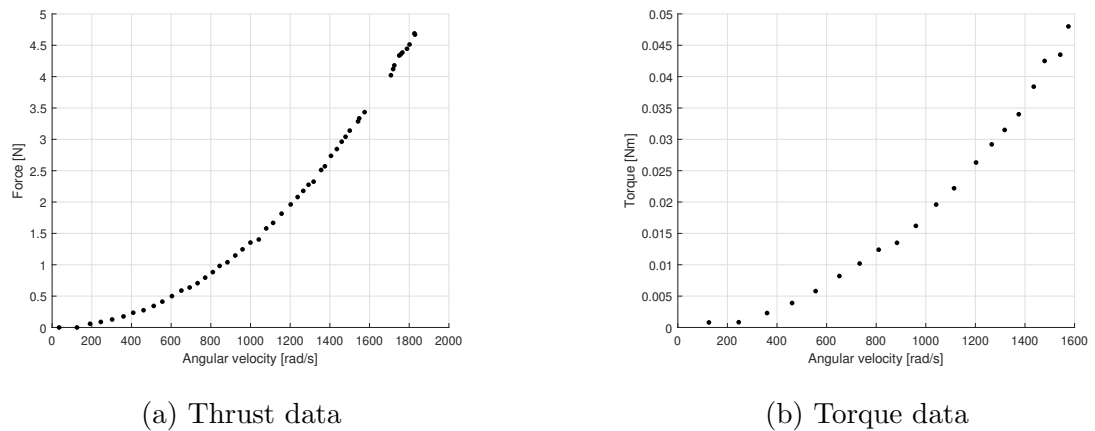


Figure 3.22: Quadrotor propellers data obtained from tests

## CHAPTER 4

# QUADROTOR SYSTEM IDENTIFICATION

---

The quadrotor system identification is addressed based on three different methods; white box, grey box, and black box approaches. The white box model considers the mathematical expressions of the quadrotor and all of its parameters are determined by analytical and computational strategies. The grey box model contemplates the mathematical expressions as well, however, most of the quadrotor parameters are unknown and they are estimated using Artificial Neural Networks (ANNs). Lastly, the black box model considers the quadrotor as a completely unknown system where there is no understanding of the relationship between inputs and outputs, therefore an ANN is trained to learn the non-linearities of the system and thus be able to predict the outputs accurately. The quadrotor white, grey, and black models are developed and presented in the following sections.

## 4.1 MATHEMATICAL MODEL (WHITE BOX APPROACH)

The quadrotor mathematical model is derived according to first principles and physical laws. All the model parameters are determined and a simulation of the model is executed incorporating an attitude control system.

### 4.1.1 EQUATIONS OF MOTION

The Newton-Euler formulation is employed to determine the mathematical model of the quadrotor UAV [62], [63]. This formulation consists of a set of differential equations that describe the quadrotor translational and rotational movements [62]. It contemplates two reference frames; one inertial frame fixed to the surface of the Earth which is defined as  $I = \{x_I, y_I, z_I\}$  and a body frame attached to the center of gravity of the quadrotor represented as  $B = \{x_B, y_B, z_B\}$ . Figure 4.1 shows graphically both frames.

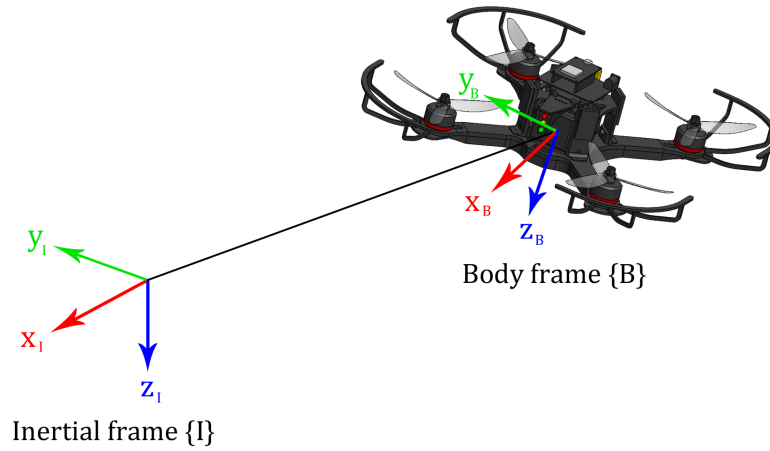


Figure 4.1: Inertial and body reference frames

The formulation assumes the quadrotor is a rigid body and its center of mass is attached and matches the body reference frame. The model is defined as:

$$\dot{\xi} = V \quad (4.1)$$

$$m\dot{V} = R(-T_T) + mge_3 \quad (4.2)$$

$$\dot{\eta} = W\Omega \quad (4.3)$$

$$J\dot{\Omega} = -\Omega \times J\Omega + \tau_a \quad (4.4)$$

where Equations 4.1 and 4.2 describe the translational dynamics while Equations 4.3 and 4.4 describe the rotational dynamics of the quadrotor.  $\xi = [x \ y \ z]^T \in \mathbb{R}^3$  represent the spatial coordinates with respect to the inertial frame and  $\eta = [\phi \ \theta \ \psi]^T \in \mathbb{R}^3$

denotes the rotation coordinates of the quadrotor UAV. The orientation of the quadrotor with respect to the inertial reference frame  $\{I\}$  is obtained by means of the rotation matrix  $R$ , which is parameterized by the Euler angles  $\phi$  roll,  $\theta$  pitch, and  $\psi$  yaw. These represent a rotation with respect to the  $x$ ,  $y$ , and  $z$  axis, respectively. In adherence to the  $zyx$  convention, the quadrotor orientation with respect to the inertial reference frame  $\{I\}$  can be described using the rotation matrix, which is deduced below.

First, a rotation by  $\psi$  about  $z_I$  is carried out (see Figure 4.2).

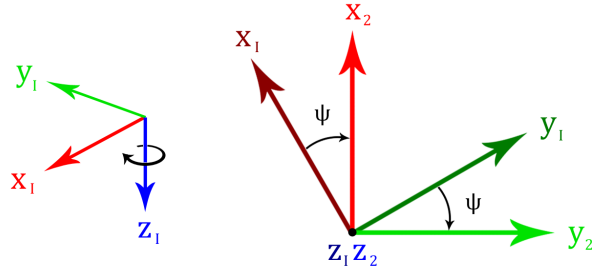


Figure 4.2: Rotation by  $\psi$  about  $z_I$

The rotation can be expressed in the following matrix form:

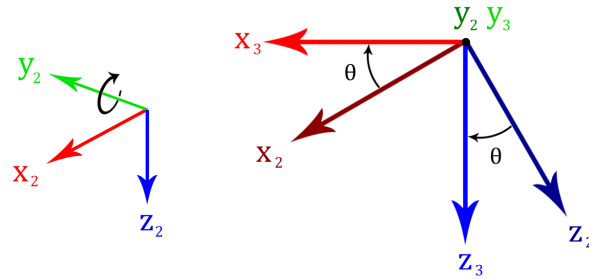
$$\begin{bmatrix} x_I \\ y_I \\ z_I \end{bmatrix} = \begin{bmatrix} C_\psi & -S_\psi & 0 \\ S_\psi & C_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = R_{2 \rightarrow I} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} \quad (4.5)$$

where  $R_{2 \rightarrow I}$  is the matrix to transform from position 2 to the inertial frame  $\{I\}$ .

Then, a rotation by  $\theta$  about  $y_2$  is performed (see Figure 4.3).

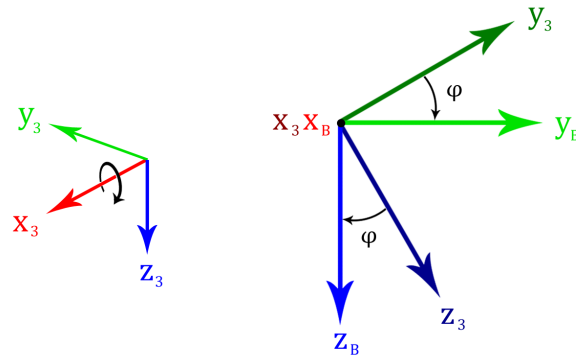
This rotation is expressed in the following matrix form:

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} C_\theta & 0 & S_\theta \\ 0 & 1 & 0 \\ -S_\theta & 0 & C_\theta \end{bmatrix} \begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix} = R_{3 \rightarrow 2} \begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix} \quad (4.6)$$

Figure 4.3: Rotation by  $\theta$  about  $y_2$ 

where  $R_{3 \rightarrow 2}$  is the matrix to transform from position 3 to position 2.

Finally, a rotation by  $\phi$  about  $x_3$  is performed (see Figure 4.4).

Figure 4.4: Rotation by  $\phi$  about  $x_3$ 

The rotation is represented by the following matrix form:

$$\begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_\phi & -S_\phi \\ 0 & S_\phi & C_\phi \end{bmatrix} \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} = R_{B \rightarrow 3} \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} \quad (4.7)$$

where  $R_{B \rightarrow 3}$  is the matrix to transform from the body frame  $\{B\}$  to position 3.

The expression below provides the orthogonal rotation matrix  $R \in SO(3)$  for

transitioning from the body frame  $\{B\}$  to the inertial frame  $\{I\}$ .

$$\begin{bmatrix} x_I \\ y_I \\ z_I \end{bmatrix} = R_{2 \rightarrow I} R_{3 \rightarrow 2} R_{B \rightarrow 3} \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} = R_{B \rightarrow I} \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} = R \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} \quad (4.8)$$

Replacing  $R_{2 \rightarrow I}$ ,  $R_{3 \rightarrow 2}$ , and  $R_{B \rightarrow 3}$  in the expression above:

$$\begin{bmatrix} x_I \\ y_I \\ z_I \end{bmatrix} = \begin{bmatrix} C_\psi & -S_\psi & 0 \\ S_\psi & C_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_\theta & 0 & S_\theta \\ 0 & 1 & 0 \\ -S_\theta & 0 & C_\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_\phi & -S_\phi \\ 0 & S_\phi & C_\phi \end{bmatrix} \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} \quad (4.9)$$

Solving matrix multiplication, the rotation matrix  $R \in SO(3)$  is obtained:

$$R = \begin{bmatrix} C_\theta C_\psi & S_\phi S_\theta C_\psi - C_\phi S_\psi & C_\phi S_\theta C_\psi + S_\phi S_\psi \\ C_\theta S_\psi & S_\phi S_\theta S_\psi + C_\phi C_\psi & C_\phi S_\theta S_\psi - S_\phi C_\psi \\ -S_\theta & S_\phi C_\theta & C_\phi C_\theta \end{bmatrix} \quad (4.10)$$

where  $C_\phi$ ,  $C_\theta$ ,  $C_\psi$ ,  $S_\phi$ ,  $S_\theta$ , and  $S_\psi$  are cosine and sine operations, correspondingly.

Continuing with the quadrotor mathematical model,  $\Omega = [p \ q \ r]^T \in \mathbb{R}^3$  represents the angular velocities in the body frame  $\{B\}$ .  $V = [v_x \ v_y \ v_z]^T \in \mathbb{R}^3$  is the translational velocity vector with respect to the inertial frame  $\{I\}$ .  $T_T = [0 \ 0 \ F_T]^T \in \mathbb{R}_{>0}^{3 \times 1}$  is the quadrotor total thrust with  $F_T$  acting as the entire vertical component of thrust in the body frame  $\{B\}$ .  $\tau_a = [\tau_\phi \ \tau_\theta \ \tau_\psi]^T$  is the vector of torques produced by the actuators. In addition, vectors  $e_1$ ,  $e_2$ , and  $e_3$  represent the canonical basis vectors of  $\mathbb{R}^3$ . The terms  $m \in \mathbb{R}$  and  $g = 9.81m/s^2$  represent the quadrotor mass and the Earth standard gravity, respectively. The matrix  $J = \text{Diag}[J_{xx} \ J_{yy} \ J_{zz}] \in \mathbb{R}^{3 \times 3}$  contains the quadrotor principal moments of inertia [64].

The quadrotor vertical component of thrust in the body frame is expressed as  $F_T = \sum_{i=1}^4 F_i$  where  $F_i$  indicates the force exerted by each motor. Both forces  $F_i$  and torques  $Q_i$  are expressed as a function of the angular velocity of the motors.  $F_i = C_F \omega_i^2$  represent the force and  $Q_i = C_Q \omega_i^2$  the reactive torque of the motor,



where  $\omega_i$  is the angular velocity and  $C_F$  and  $C_Q$  are the thrust and torque coefficients respectively.

Furthermore, the roll dynamics produced by the actuators is expressed as  $\tau_\phi = (-F_1 + F_2 + F_3 - F_4) d$ , for pitch dynamics is  $\tau_\theta = (F_1 + F_2 - F_3 - F_4) d$ , and for yaw dynamics  $\tau_\psi = Q_1 - Q_2 + Q_3 - Q_4$ , where  $d = l \cos 45$ . Figure 4.5 shows the quadrotor forces and moments convention.

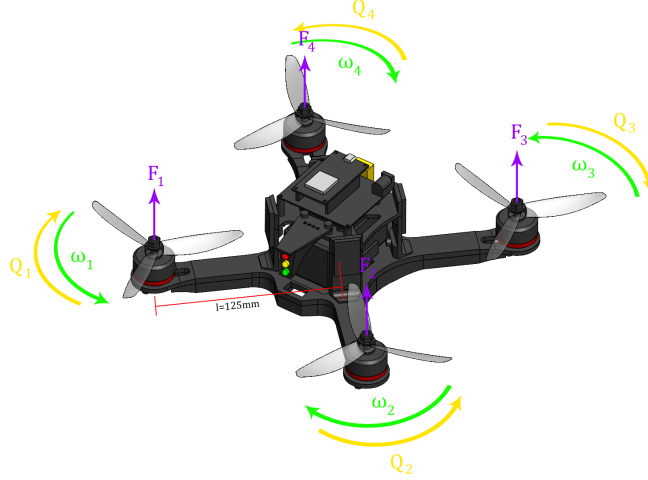


Figure 4.5: Quadrotor forces and moments convention

The quadrotor angular velocity in the body frame is written as:

$$\Omega = R^{3 \rightarrow B} R^{2 \rightarrow 3} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + R^{3 \rightarrow B} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} \quad (4.11)$$

where  $R^{3 \rightarrow B} = (R^{B \rightarrow 3})^T$  and  $R^{2 \rightarrow 3} = (R^{3 \rightarrow 2})^T$ .

Replacing  $R^{3 \rightarrow B}$  and  $R^{2 \rightarrow 3}$  in the above expression:

$$\Omega = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_\phi & S_\phi \\ 0 & -S_\phi & C_\phi \end{bmatrix} \begin{bmatrix} C_\theta & 0 & -S_\theta \\ 0 & 1 & 0 \\ S_\theta & 0 & C_\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_\phi & S_\phi \\ 0 & -S_\phi & C_\phi \end{bmatrix} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} \quad (4.12)$$

Solving matrix multiplication:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \sin \phi \cos \theta \\ 0 & -\sin \phi & \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (4.13)$$

Applying the inverse matrix, it is obtained:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (4.14)$$

where

$$W = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix} \quad (4.15)$$

is the matrix to convert the quadrotor angular velocity from the body frame  $\{B\}$  to the inertial frame  $\{I\}$ .

Considering all the mathematical statements previously developed and according to Equations 4.1 to 4.4, the expanded form of the quadrotor mathematical model is presented, where Equations 4.16 and 4.17 describe the quadrotor translational dynamics while Equations 4.18 and 4.19 describe its rotational dynamics, assuming all parameters are known.

$$\begin{aligned} \dot{x} &= v_x \\ \dot{y} &= v_y \\ \dot{z} &= v_z \end{aligned} \quad (4.16)$$

$$\begin{aligned} \dot{v}_x &= -\frac{F_T}{m} (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) \\ \dot{v}_y &= -\frac{F_T}{m} (\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi) \\ \dot{v}_z &= -\frac{F_T}{m} (\cos \phi \cos \theta) + g \end{aligned} \quad (4.17)$$

$$\begin{aligned}
\dot{\phi} &= p + q \sin \phi \tan \theta + r \cos \phi \tan \theta \\
\dot{\theta} &= q \cos \phi - r \sin \phi \\
\dot{\psi} &= q \sin \phi \sec \theta + r \cos \phi \sec \theta
\end{aligned} \tag{4.18}$$

$$\begin{aligned}
\dot{p} &= \left( \frac{J_{yy} - J_{zz}}{J_{xx}} \right) qr + \left( \frac{1}{J_{xx}} \right) \tau_{\phi} \\
\dot{q} &= \left( \frac{J_{zz} - J_{xx}}{J_{yy}} \right) pr + \left( \frac{1}{J_{yy}} \right) \tau_{\theta} \\
\dot{r} &= \left( \frac{J_{xx} - J_{yy}}{J_{zz}} \right) pq + \left( \frac{1}{J_{zz}} \right) \tau_{\psi}
\end{aligned} \tag{4.19}$$

#### 4.1.2 PARAMETER DETERMINATION

The quadrotor mathematical model previously presented is applicable to any type of quadrotor whose geometry is symmetrical in its three main axes. However, what makes one quadrotor different from another (in the context of its dynamics) are the model parameters, which are the mass, lever arm distance, propeller aerodynamic coefficients, and moments of inertia. The model parameters corresponding to the quadrotor constructed in Chapter 3 are determined below.

The mass ( $m$ ) is easily obtained by putting the quadrotor on a scale, thus measuring the quantity.

The lever arm distance ( $d$ ) is easily calculated as well, by measuring the quadrotor arm length ( $l$ ) and then multiplying it by  $\cos 45$ .

The aerodynamic coefficients ( $C_F$  and  $C_Q$ ) are parameters that cannot be measured directly. In order to determine these values, the propellers data obtained from the experimental tests is used and a least-squares polynomial regression model is applied to each set of data. Since both data can perfectly fit on a second-order

polynomial starting at the origin, the model is represented as:

$$y = a_0x^2 + e \quad (4.20)$$

solving for the error  $e$ :

$$e = y - a_0x^2 \quad (4.21)$$

carrying out the sum of the square of the residuals  $S_r$ :

$$S_r = \sum (y_i - a_0x_i^2)^2 \quad (4.22)$$

results in the equation that has to be minimized, hence deriving  $S_r$  with respect to the leading coefficient  $a_0$  and setting equal to zero:

$$-2 \sum (y_i - a_0x_i^2) x_i^2 = 0 \quad (4.23)$$

solving for  $a_0$ :

$$a_0 = \frac{\sum x_i^2 y_i}{\sum x_i^4} \quad (4.24)$$

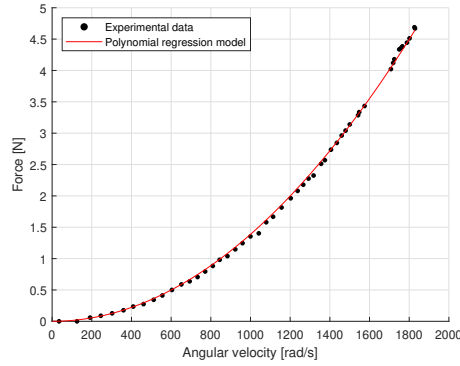
where  $\sum = \sum_{i=1}^n$  and  $a_0$  is the parameter that best fits the data with quadratic behavior and which starts at the origin. The polynomial regression is based on reference [60].

Thus,  $a_0$ ,  $x_i$  and  $y_i$  from Equation 4.24 can be rewritten as:

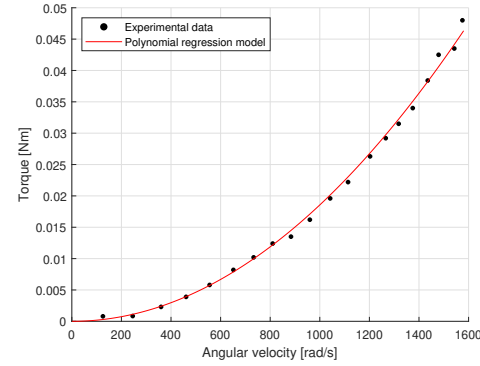
$$C_F = \frac{\sum \omega_i^2 F_i}{\sum \omega_i^4} \quad (4.25)$$

$$C_Q = \frac{\sum \omega_i^2 Q_i}{\sum \omega_i^4} \quad (4.26)$$

where Equations 4.25 and 4.26 correspond to the propeller thrust and torque coefficients, respectively. Recall that all summations are from  $i = 1$  through  $n = 50$  for the thrust coefficient and from  $i = 1$  through  $n = 20$  for the torque coefficient, according to the number of experimental tests addressed in Chapter 3. The comparison between the experimental data and the polynomial regression models of the quadrotor propellers is shown in Figure 4.6.



(a) Thrust model



(b) Torque model

Figure 4.6: Polynomial regression models of quadrotor propellers

The principal moments of inertia ( $J_{xx}$ ,  $J_{yy}$ , and  $J_{zz}$ ) can be calculated according to Equation 4.27 which is the integral of the squared distance from the axis of rotation ( $r^2$ ) multiplied by the differential mass ( $dm$ ) over the total mass distribution of a rotating body.

$$J = \int_m r^2 dm \quad (4.27)$$

Since the mass distribution of the quadrotor is complex to find by direct measuring, the CAD software Solidworks<sup>®</sup> was utilized for greater ease and thus to estimate the moments of inertia. This was done by directly measuring the mass of all the quadrotor components on a scale and then assigning these mass values to the 3D models in the software. Then, when assembling all the quadrotor components with their corresponding masses, the moments of inertia of the whole quadrotor can be found by accessing the mass properties, as shown in Figure 4.7.

In this way, all the quadrotor parameters are determined and this implies that when replaced in the mathematical model, they completely and accurately describe the quadrotor dynamics. The parameter values are presented in Table 4.1.

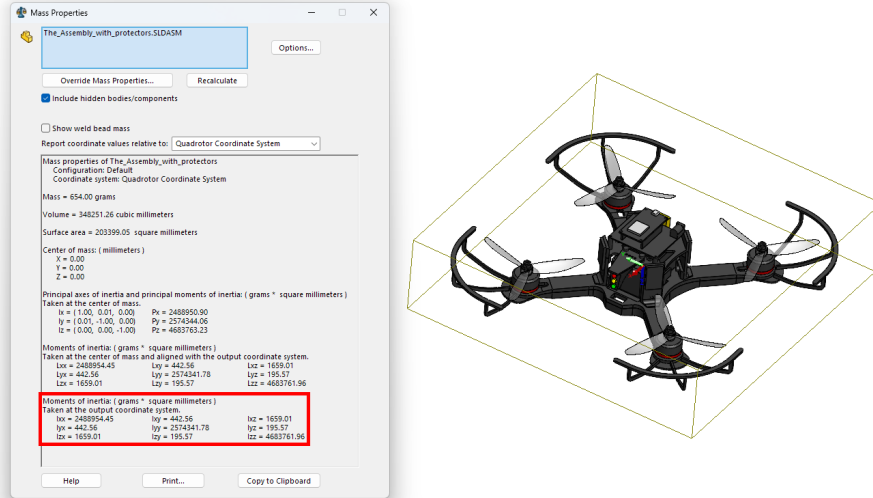


Figure 4.7: Quadrotor moments of inertia matrix obtained in CAD software

Table 4.1: Quadrotor parameters (white box approach)

Parameter	Value	Unit	Technique for determining
$m$	0.652	kg	Direct measuring
$d$	0.0884	m	Direct measuring
$C_F$	$1.3877 \times 10^{-6}$	$\text{N}\cdot\text{s}^2$	Least-Squares polynomial regression
$C_Q$	$1.8558 \times 10^{-8}$	$\text{N}\cdot\text{m}\cdot\text{s}^2$	Least-Squares polynomial regression
$J_{xx}$	$2.4890 \times 10^{-3}$	$\text{kg}\cdot\text{m}^2$	CAD software
$J_{yy}$	$2.5743 \times 10^{-3}$	$\text{kg}\cdot\text{m}^2$	CAD software
$J_{zz}$	$4.6838 \times 10^{-3}$	$\text{kg}\cdot\text{m}^2$	CAD software

### 4.1.3 MODEL SIMULATION

The quadrotor mathematical model is tested using MATLAB<sup>®</sup> Simulink<sup>®</sup> software. The simulation consists of an attitude control system (see Figure 4.8). The orientation and total force are the system references and the angular velocities, angular accelerations, and torques are the output variables. Note that disturbance and noise are added to make the simulation as close to reality as possible. The reference signals for the total force and orientation angles are presented in Equation 4.28, where the force and angles are measured in Newtons and radians, respectively.

$$F_{T_d} = 5$$

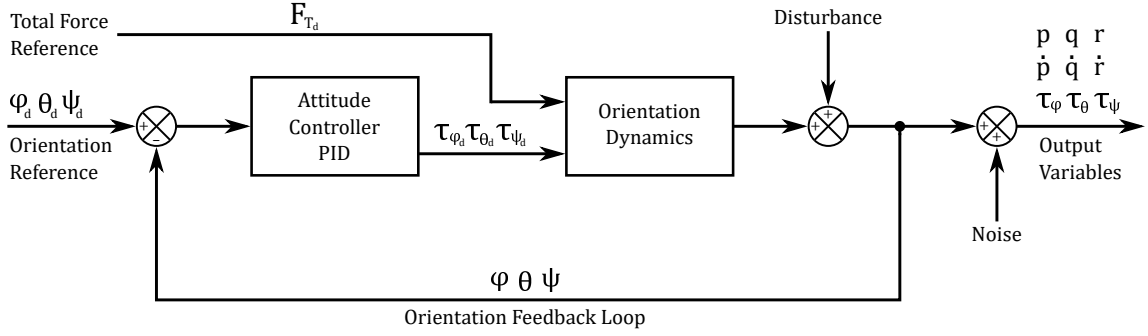


Figure 4.8: Quadrotor attitude control system

$$\begin{aligned}
 \phi_d(t) &= \frac{\pi}{36} \sin(4t) \\
 \theta_d(t) &= \frac{\pi}{36} \sin\left(4t + \frac{\pi}{2}\right) \\
 \psi_d(t) &= \frac{\pi}{9}t
 \end{aligned} \tag{4.28}$$

The attitude controller is chosen as PID where the inputs are the errors between the desired and actual orientation angles, as shown below.

$$\begin{aligned}
 \tilde{\phi}(t) &= \phi_d(t) - \phi(t) \\
 \tilde{\theta}(t) &= \theta_d(t) - \theta(t) \\
 \tilde{\psi}(t) &= \psi_d(t) - \psi(t)
 \end{aligned} \tag{4.29}$$

The quadrotor torques are the controller output variables which are defined in accordance with the proposed law control shown in Equation 4.30, where gain factors  $k_p$ ,  $k_i$ , and  $k_d$  from each orientation angle are manually defined so that the attitude response is as expected. The PID gain factors are presented in Table 4.2.

$$\begin{aligned}
 \tau_{\phi_d} &= k_{p_\phi} \tilde{\phi} + k_{i_\phi} \int \tilde{\phi} dt + k_{d_\phi} \dot{\tilde{\phi}} \\
 \tau_{\theta_d} &= k_{p_\theta} \tilde{\theta} + k_{i_\theta} \int \tilde{\theta} dt + k_{d_\theta} \dot{\tilde{\theta}} \\
 \tau_{\psi_d} &= k_{p_\psi} \tilde{\psi} + k_{i_\psi} \int \tilde{\psi} dt + k_{d_\psi} \dot{\tilde{\psi}}
 \end{aligned} \tag{4.30}$$

Consider the vector  $\Gamma = [F_T \ \tau_\phi \ \tau_\theta \ \tau_\psi]^T$  as the vector containing the total force and torques, and the vector  $U = [u_1 \ u_2 \ u_3 \ u_4]^T = [\omega_1^2 \ \omega_2^2 \ \omega_3^2 \ \omega_4^2]^T$  as the vector that

Table 4.2: PID attitude controller gain factors

Gain Factor	Value	Gain Factor	Value	Gain Factor	Value
$k_{p_\phi}$	0.2	$k_{i_\phi}$	3	$k_{d_\phi}$	0.1
$k_{p_\theta}$	0.2	$k_{i_\theta}$	3	$k_{d_\theta}$	0.1
$k_{p_\psi}$	4	$k_{i_\psi}$	0.1	$k_{d_\psi}$	0.1

contains the inputs to the quadrotor mathematical model, which are the propeller quadratic velocities. According to the mathematical model, the total force and torques can be expressed in the form  $\Gamma = C_A U$ , where  $C_A$  contains the parameters  $C_F$ ,  $C_Q$ , and  $d$ . The expanded expression is shown below.

$$\begin{bmatrix} F_T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} C_F & C_F & C_F & C_F \\ -C_F d & C_F d & C_F d & -C_F d \\ C_F d & C_F d & -C_F d & -C_F d \\ C_Q & -C_Q & C_Q & -C_Q \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \quad (4.31)$$

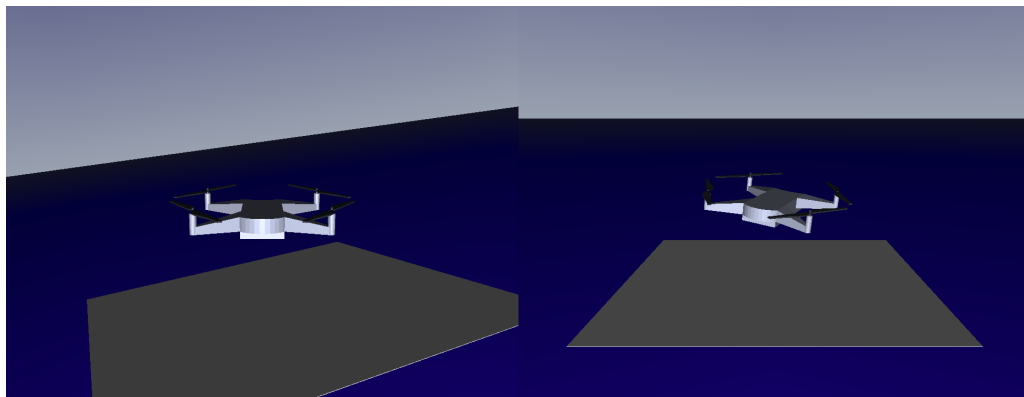
The input vector  $U$  is obtained by the expression  $U = C_A^{-1} \Gamma$  where  $\Gamma$  becomes  $\Gamma_d$ , that is  $\Gamma_d = [F_{T_d} \ \tau_{\phi_d} \ \tau_{\theta_d} \ \tau_{\psi_d}]^T$ , hence:

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} \frac{1}{C_F} & -\frac{1}{C_F d} & \frac{1}{C_F d} & \frac{1}{C_Q} \\ \frac{1}{C_F} & \frac{1}{C_F d} & \frac{1}{C_F d} & -\frac{1}{C_Q} \\ \frac{1}{C_F} & \frac{1}{C_F d} & -\frac{1}{C_F d} & \frac{1}{C_Q} \\ \frac{1}{C_F} & -\frac{1}{C_F d} & -\frac{1}{C_F d} & \frac{1}{C_Q} \end{bmatrix} \begin{bmatrix} F_{T_d} \\ \tau_{\phi_d} \\ \tau_{\theta_d} \\ \tau_{\psi_d} \end{bmatrix} \quad (4.32)$$

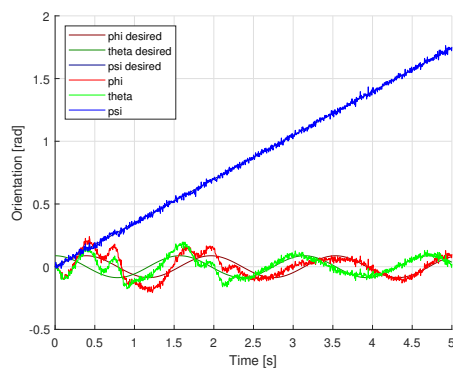
is the expression utilized to find the quadratic velocities (desired velocities) which are the inputs to the quadrotor mathematical model that consequently produces its orientation dynamics following the reference signals.

The simulation is performed for 5 seconds and the variables of interest are extracted from the system response. The results are presented in Figure 4.9. Note that a 3D virtual environment is created so that the quadrotor attitude response is also graphically seen.

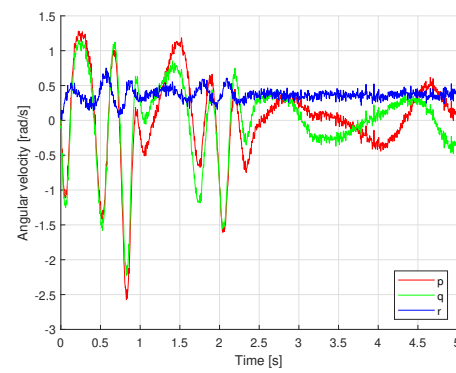




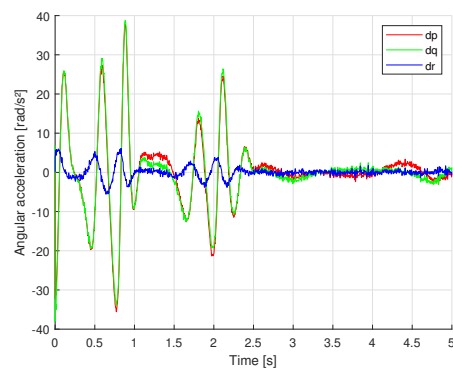
(a) 3D virtual environment



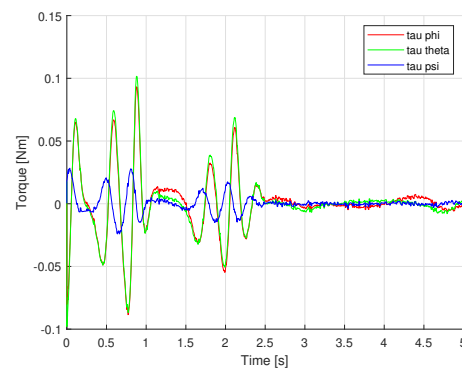
(b) Orientation



(c) Velocity



(d) Acceleration



(e) Torque

Figure 4.9: Quadrotor attitude control simulation results

## 4.2 ARTIFICIAL NEURAL NETWORKS

The operating principle of an Artificial Neural Network (ANN) is addressed in this section as well as the learning algorithms used for the quadrotor system identification. All ANNs developed in this work are based on references [65], [66].

### 4.2.1 NODES AND LAYERS

An ANN is composed of neurons (nodes) that are connected to other neurons by weighted connectors. The function of a neuron is to transmit signals from one to another. Figure 4.10 shows a neuron which receives three input signals ( $x_1$ ,  $x_2$ , and  $x_3$ ) and outputs one signal ( $y$ ). The circle refers to the neuron, the squares to the inputs, and the arrows indicate the signal flow direction. Variables  $w_1$ ,  $w_2$ , and  $w_3$  are called the weights of the neural network and they store information. The variable  $b$  is the bias and it also stores information. Thus, the neural network stores information in the form of weights and biases [65].

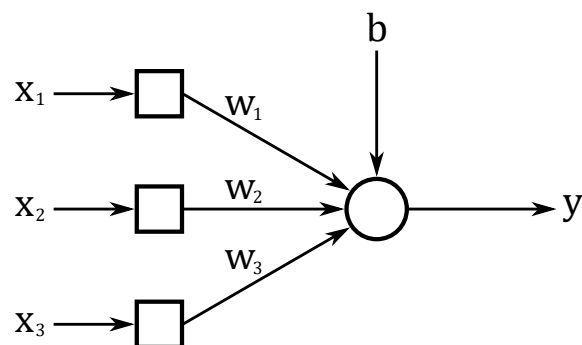


Figure 4.10: A node that receives three inputs

In the example of Figure 4.10, the input signals are multiplied by their corresponding weights before reaching the node. Once the weighted signals reach the node, they are added and become the weighted sum including the sum of the bias.

The above is calculated as follows:

$$v = w_1x_1 + w_2x_2 + w_3x_3 + b \quad (4.33)$$

where  $v$  is the weighted sum of the node. This equation implies that signals with higher weight will have a more significant impact on the output than the others. This equation can also be written with matrices as:

$$v = wx + b \quad (4.34)$$

where

$$w = \begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix}$$

$$x = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T$$

Lastly, the node inputs the weighted sum into the activation function, resulting in the node output:

$$y = \varphi(v) = \varphi(wx + b) \quad (4.35)$$

where  $\varphi(\cdot)$  is the activation function.

There exist different activation functions that can be implemented depending on the application of the neural network, such as the ones shown in Figure 4.11. In this work, linear and sigmoid activation functions are employed for the quadrotor system identification.

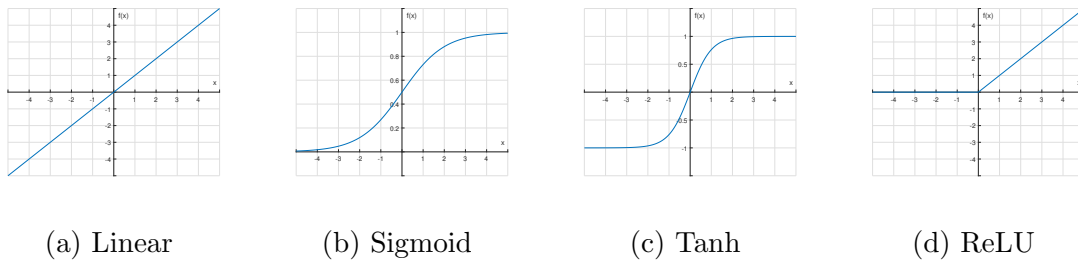


Figure 4.11: Commonly used activation functions

Different types of neural networks can be created based on how the nodes are connected to each other. One of the most utilized types of neural networks is characterized by having a layered structure of nodes as shown in Figure 4.12.

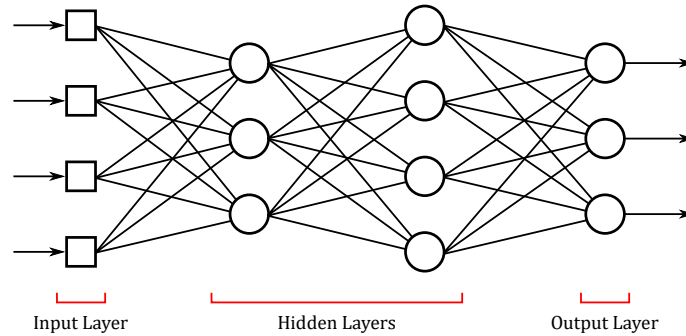


Figure 4.12: Neural network structure composed of layers of nodes

The input layer consists of a group of square nodes that directly transmit the input signals without calculating any weighted sum or activation function. On the other hand, the output layer consists of the nodes that calculate the final result of the neural network. The hidden layers are the ones located between the input and output layers. Neural networks that only consist of input and output layers are called *single-layer* neural networks. Neural networks that consist of input, output, and one or more hidden layers are referred to as *multi-layer* neural networks, where a shallow neural network has only one hidden layer and a deep neural network possesses two or more hidden layers.

## 4.2.2 LEARNING METHODOLOGY

Machine learning can be divided into three main learning methodologies; supervised learning, unsupervised learning, and reinforcement learning. Supervised learning refers to when instances are given with known labels while in unsupervised learning, the instances are unlabeled. Reinforcement learning refers to when the learner is not instructed on which action to take but instead must discover by itself which actions produce the highest reward by trying them [67].

Neural networks can be trained using supervised, unsupervised, and reinforcement learning techniques, however, this work considers supervised learning as the learning methodology of the neural network (see Figure 4.13).

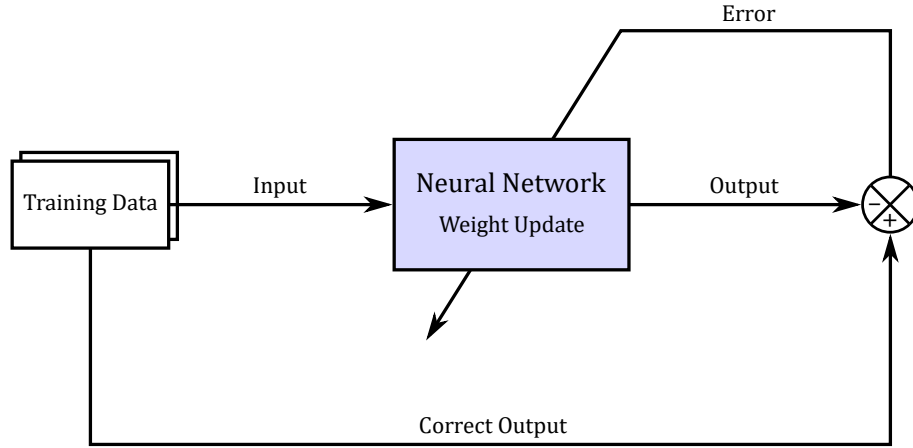


Figure 4.13: Supervised learning scheme of a neural network

In supervised learning, each data point from the training data consists of an  $\{input - correct\ output\}$  pair, where the input enters the neural network and produces an output which is then compared to the correct output. The error between the output and the correct output goes into the neural network and updates its weights in order to minimize the error according to the cost function, which is the measure of the neural network error. This process repeats for each pair of data.

### 4.2.3 GENERALIZED DELTA RULE

The delta rule is a type of numerical approach referred to as *gradient descent*, which is the characteristic learning rule of the single-layer neural network [65]. For any activation function, the delta rule is expressed as:

$$w_{ij} \leftarrow w_{ij} + \alpha \delta_i x_j \quad (4.36)$$

where  $w_{ij}$  is the weight between the output node  $i$  and input node  $j$ ,  $\alpha$  is the learning rate whose value can be between 0 and 1,  $x_j$  is the output from the input node  $j$  and  $\delta_i$  is defined as:

$$\delta_i = \dot{\varphi}(v_i) e_i \quad (4.37)$$

where  $\dot{\varphi}$  is the derivative of the activation function  $\varphi$  of the output node  $i$ ,  $v_i$  is the weighted sum of the output node  $i$  and  $e_i$  is the error of the output node  $i$ . The

error is expressed as follows:

$$e_i = d_i - y_i \quad (4.38)$$

where  $d_i$  is the correct output and  $y_i$  the neural network output.

For a linear activation function:

$$\varphi(v_i) = v_i \quad (4.39)$$

its derivative is expressed as:

$$\dot{\varphi}(v_i) = 1 \quad (4.40)$$

replacing  $\dot{\varphi}(v_i)$  on Equation 4.37,

$$\delta_i = e_i \quad (4.41)$$

substituting for  $\delta_i$  on Equation 4.36,

$$w_{ij} \leftarrow w_{ij} + \alpha e_i x_j \quad (4.42)$$

is the rule used to update the weights for a linear activation function.

For a sigmoid activation function:

$$\varphi(v_i) = \frac{1}{1 + e^{-v_i}} \quad (4.43)$$

its derivative is expressed as:

$$\dot{\varphi}(v_i) = \varphi(v_i) (1 - \varphi(v_i)) \quad (4.44)$$

replacing  $\dot{\varphi}(v_i)$  on Equation 4.37,

$$\delta_i = \varphi(v_i) (1 - \varphi(v_i)) e_i \quad (4.45)$$

substituting for  $\delta_i$  on Equation 4.36,

$$w_{ij} \leftarrow w_{ij} + \alpha \varphi(v_i) (1 - \varphi(v_i)) e_i x_j \quad (4.46)$$

is the rule used to update the weights for a sigmoid activation function.

To simplify, the weights update for the delta rule can also be expressed as:

$$w_{ij} \leftarrow w_{ij} + \Delta w_{ij} \quad (4.47)$$

where  $\Delta w_{ij} = \alpha \delta_i x_j$  is the expression that contains the calculations for learning.

An example of the learning process of a single-layer neural network employing the delta rule is presented. Consider the neural network shown in Figure 4.14.

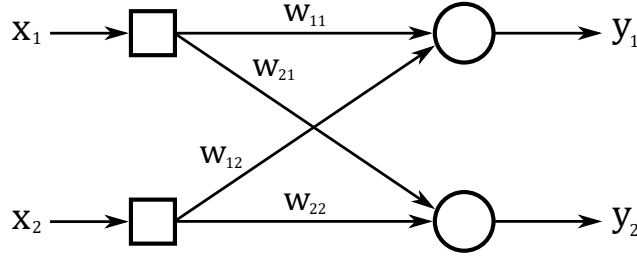


Figure 4.14: Neural network that consists of two input and two output nodes

The weighted sums of the output nodes are expressed as:

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (4.48)$$

The outputs of the neural network are then:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \varphi(v_1) \\ \varphi(v_2) \end{bmatrix} \quad (4.49)$$

Considering the delta rule as the neural network learning algorithm, the update of the weights is calculated as follows:

$$\begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} \leftarrow \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} + \alpha \begin{bmatrix} \delta_1 \\ \delta_2 \end{bmatrix} \odot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (4.50)$$

where symbol  $\odot$  refers to the *Hadamard product* and the deltas  $\delta_i$  are:

$$\begin{bmatrix} \delta_1 \\ \delta_2 \end{bmatrix} = \begin{bmatrix} \dot{\varphi}(v_1) \\ \dot{\varphi}(v_2) \end{bmatrix} \odot \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}$$

In this way, the neural network updates its weights in order to minimize the error between its output and the correct output.

#### 4.2.4 TRAINING DATA PROCESSING

There are several approaches in which a neural network processes the training data. Three typical methods for processing training data in supervised learning neural networks are the *stochastic gradient descent*, *batch gradient descent*, and *mini-batch gradient descent*.

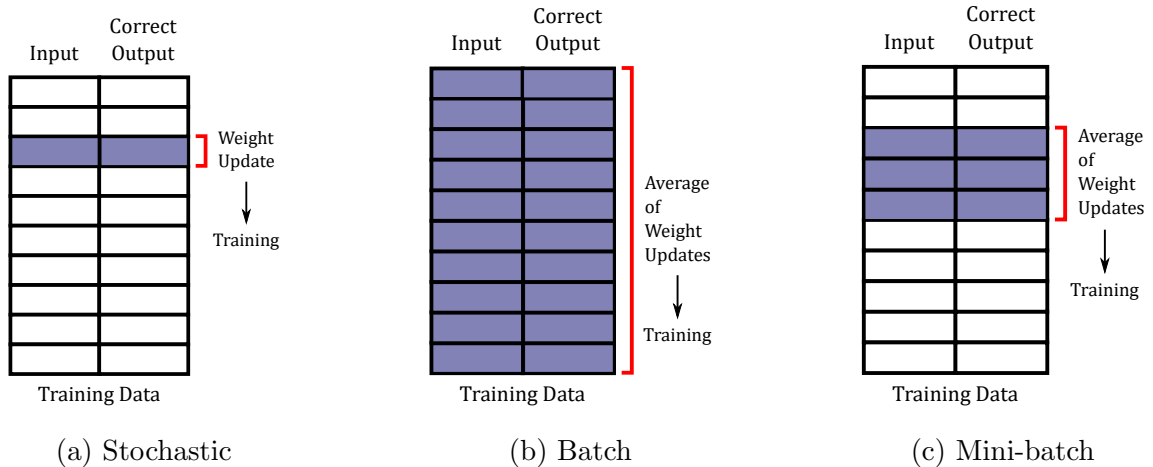


Figure 4.15: How the different methods select training data and update the weights

The stochastic gradient descent is a method where the neural network is trained with each weight update (see Figure 4.15a). On the other hand, the batch gradient descent uses the average of all weight updates for the training (see Figure 4.15b). The mini-batch gradient descent is a combination of the stochastic and batch methods, where a part of the training data is selected and uses the average of the weight updates from the picked dataset to train the neural network (see Figure 4.15c).

When all the training data has been used for the weight updates, an *epoch* has taken place. For the case of the stochastic gradient descent approach shown in Figure 4.15a, ten pieces of training equal an epoch. On the other hand, the batch gradient descent approach from Figure 4.15b means that one training equals an epoch. Lastly, in the case of the mini-batch gradient descent approach shown in Figure 4.15c, four pieces of training are carried out per epoch.



## 4.2.5 BACK-PROPAGATION ALGORITHM

The back-propagation algorithm has become the standard algorithm for training multi-layer neural networks. It learns by calculating the errors of the output layer back-propagating them to determine the errors in the hidden layers [68].

An example of the learning process of a multi-layer neural network employing the back-propagation algorithm is presented. Consider the multi-layer (shallow) neural network from Figure 4.16.

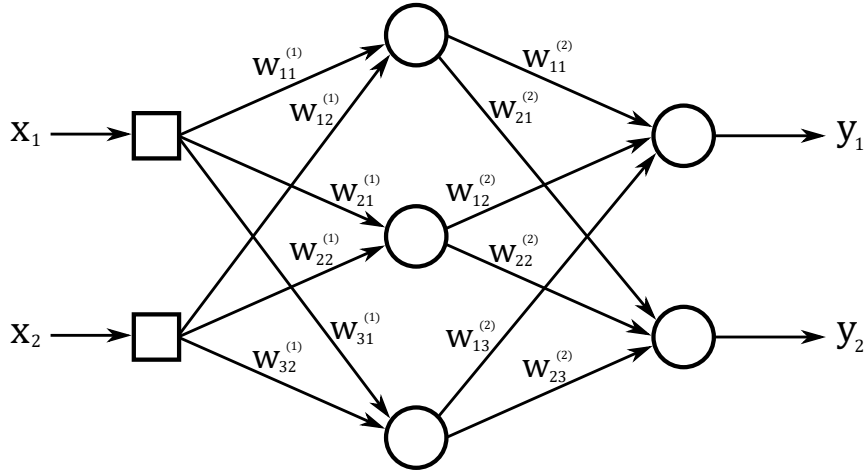


Figure 4.16: Neural network that consists of two input and two output nodes and a hidden layer with three nodes

The weighted sums of the nodes from the hidden layer are expressed as:

$$\begin{bmatrix} v_1^{(1)} \\ v_2^{(1)} \\ v_3^{(1)} \end{bmatrix} = \begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} \\ w_{31}^{(1)} & w_{32}^{(1)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (4.51)$$

The outputs of the hidden layer are then:

$$\begin{bmatrix} y_1^{(1)} \\ y_2^{(1)} \\ y_3^{(1)} \end{bmatrix} = \begin{bmatrix} \varphi(v_1^{(1)}) \\ \varphi(v_2^{(1)}) \\ \varphi(v_3^{(1)}) \end{bmatrix} \quad (4.52)$$

consequently, these nodes become the input nodes to the output layer. Thus, the weighted sums of the output nodes are:

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} w_{11}^{(2)} & w_{12}^{(2)} & w_{13}^{(2)} \\ w_{21}^{(2)} & w_{22}^{(2)} & w_{23}^{(2)} \end{bmatrix} \begin{bmatrix} y_1^{(1)} \\ y_2^{(1)} \\ y_3^{(1)} \end{bmatrix} \quad (4.53)$$

The outputs of the neural network are then:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \varphi(v_1) \\ \varphi(v_2) \end{bmatrix} \quad (4.54)$$

Now the back-propagation algorithm is applied, which is based on the delta rule previously presented. The update of the weights that are situated between the hidden and output layers are calculated as follows:

$$\begin{bmatrix} w_{11}^{(2)} & w_{12}^{(2)} & w_{13}^{(2)} \\ w_{21}^{(2)} & w_{22}^{(2)} & w_{23}^{(2)} \end{bmatrix} \leftarrow \begin{bmatrix} w_{11}^{(2)} & w_{12}^{(2)} & w_{13}^{(2)} \\ w_{21}^{(2)} & w_{22}^{(2)} & w_{23}^{(2)} \end{bmatrix} + \alpha \begin{bmatrix} \delta_1 \\ \delta_2 \end{bmatrix} \odot \begin{bmatrix} y_1^{(1)} \\ y_2^{(1)} \\ y_3^{(1)} \end{bmatrix}^T \quad (4.55)$$

where the deltas  $\delta_i$  are:

$$\begin{bmatrix} \delta_1 \\ \delta_2 \end{bmatrix} = \begin{bmatrix} \dot{\varphi}(v_1) \\ \dot{\varphi}(v_2) \end{bmatrix} \odot \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}$$

The update of the weights that are situated between the input and hidden layers are calculated as follows:

$$\begin{bmatrix} w_{11}^{(1)} & w_{21}^{(1)} & w_{31}^{(1)} \\ w_{12}^{(1)} & w_{22}^{(1)} & w_{32}^{(1)} \end{bmatrix} \leftarrow \begin{bmatrix} w_{11}^{(1)} & w_{21}^{(1)} & w_{31}^{(1)} \\ w_{12}^{(1)} & w_{22}^{(1)} & w_{32}^{(1)} \end{bmatrix} + \alpha \begin{bmatrix} \delta_1^{(1)} \\ \delta_2^{(1)} \\ \delta_3^{(1)} \end{bmatrix}^T \odot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (4.56)$$

where the deltas  $\delta_i^{(1)}$  are:

$$\begin{bmatrix} \delta_1^{(1)} \\ \delta_2^{(1)} \\ \delta_3^{(1)} \end{bmatrix} = \begin{bmatrix} \dot{\varphi}(v_1^{(1)}) \\ \dot{\varphi}(v_2^{(1)}) \\ \dot{\varphi}(v_3^{(1)}) \end{bmatrix} \odot \begin{bmatrix} e_1^{(1)} \\ e_2^{(1)} \\ e_3^{(1)} \end{bmatrix}$$

Note that the errors  $e_i^{(1)}$  cannot be calculated by comparing the output with the correct output since these are situated in the hidden layer. Therefore, they are calculated as follows:

$$\begin{bmatrix} e_1^{(1)} \\ e_2^{(1)} \\ e_3^{(1)} \end{bmatrix} = \begin{bmatrix} w_{11}^{(2)} & w_{21}^{(2)} \\ w_{12}^{(2)} & w_{22}^{(2)} \\ w_{13}^{(2)} & w_{23}^{(2)} \end{bmatrix} \begin{bmatrix} \delta_1 \\ \delta_2 \end{bmatrix} \quad (4.57)$$

In this way, the neural network updates its weights in order to minimize the error between its output and the correct output. Note that the process of updating goes backward, back-propagating the errors from the outputs to the inputs, hence the name of the algorithm.

### 4.3 PARAMETER ESTIMATION (GREY BOX APPROACH)

The quadrotor mathematical model, as discussed earlier in this chapter, possesses parameters that characterize the behavior of the vehicle. In this section, the aerodynamic and inertial parameters of the quadrotor mathematical model are estimated through the application of artificial neural networks.

#### 4.3.1 NEURAL NETWORK FOR ESTIMATION OF AERODYNAMIC PARAMETERS

The aerodynamic parameters  $C_F$  and  $C_Q$  from the quadrotor mathematical model are estimated using the experimental data from Chapter 3 through the training of a neural network for each parameter. The proposed neural network architecture for this application is shown in Figure 4.17. Note that the input node receives a quadratic value  $x^2$  which represents the propeller quadratic velocities. One output node  $y$  is proposed, which outputs the force/torque values. Furthermore, a hidden layer of  $n$  number of nodes is considered.

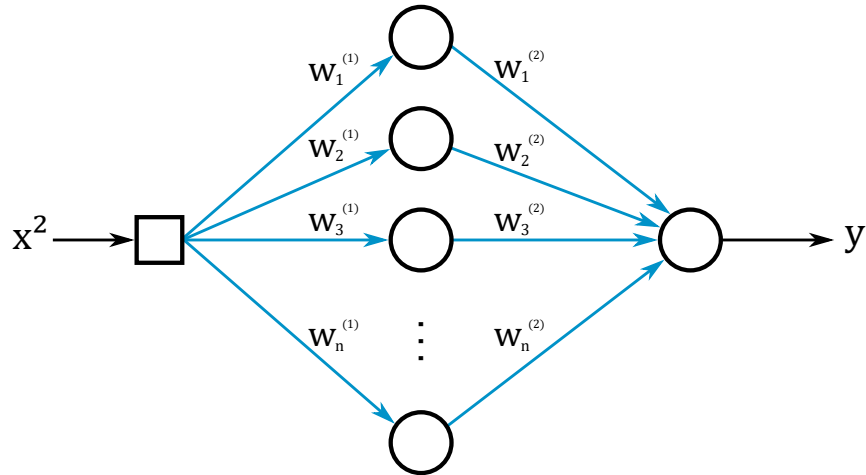


Figure 4.17: Neural network that consists of one input and one output nodes and a hidden layer of  $n$  nodes

The weights are represented as follows:

$$w^{(1)} = [w_1^{(1)} \quad w_2^{(1)} \quad w_3^{(1)} \quad \dots \quad w_n^{(1)}]^T \quad (4.58)$$

$$w^{(2)} = [w_1^{(2)} \quad w_2^{(2)} \quad w_3^{(2)} \quad \dots \quad w_n^{(2)}]$$

The weighted sum of the nodes from the hidden layer are:

$$v^{(1)} = w^{(1)}x^2 \quad (4.59)$$

Then, the output of the hidden layer is calculated as:

$$y^{(1)} = \varphi(v^{(1)}) \quad (4.60)$$

The weighted sum of the output node is expressed as:

$$v = w^{(2)}y^{(1)} \quad (4.61)$$

Lastly, the output from the neural network is:

$$y = \varphi(v) \quad (4.62)$$

For linear activation functions in the hidden and output layers:

$$y^{(1)} = v^{(1)} \quad (4.63)$$

$$y = v \quad (4.64)$$

Thus, the output of the neural network can be re-written as:

$$y = w^{(2)}w^{(1)}x^2 \quad (4.65)$$

Note that Equation 4.65 has the same structure of a second-order polynomial starting at the origin, where the weights multiplication  $w^{(2)}w^{(1)}$  represent the leading coefficient. In the same way, this equation is equivalent to the quadrotor force and torque expressions from the mathematical model:

$$F = C_F\omega^2 \quad (4.66)$$

$$Q = C_Q\omega^2$$

therefore, the proper adjustment of weights  $w^{(1)}$  and  $w^{(2)}$  on Equation 4.65 leads to finding the aerodynamic parameters  $C_F$  and  $C_Q$  by employing the force/torque experimental data previously shown in Figure 3.22. Nevertheless, since the data contains quantities greater than 1, data normalization has to be carried out before training so the neural network works properly, that is handling values from 0 to 1.

Input  $x$  and correct output  $d$  from the experimental tests are scaled according to the following expressions:

$$x_{scaled} = k_x x \quad (4.67)$$

$$d_{scaled} = k_d d$$

where  $k_x$  and  $k_d$  are the scaling factors, which are calculated as follows:

$$k_x = \frac{1}{\max(|x|)} \quad (4.68)$$

$$k_d = \frac{1}{\max(|d|)}$$

Since the correct output  $d$  is compared to the neural network output  $y$  in order to minimize the cost function, and considering that the correct output is now scaled by the factor  $k_d$ , this implies that the neural network output is also scaled by the same factor. Therefore, the neural network scaled output is expressed as:

$$y_{scaled} = k_d y \quad (4.69)$$

Replacing the scaled values  $x_{scaled}$  and  $y_{scaled}$  into Equation 4.65, it is obtained:

$$k_d y = w^{(2)} w^{(1)} (k_x x)^2 \quad (4.70)$$

developing the quadratic term and solving for  $y$ , it becomes:

$$y = w^{(2)} w^{(1)} \left( \frac{k_x^2}{k_d} \right) x^2 \quad (4.71)$$

where  $x$  and  $y$  are the input and neural network output without scaling, respectively. The weights multiplication  $w^{(2)} w^{(1)}$  multiplied by the scaling expression  $\left( \frac{k_x^2}{k_d} \right)$  represents the leading coefficient.

Aerodynamic parameters  $C_F$  and  $C_Q$  are estimated by training two separate neural networks, according to each parameter. The training data for both forces and torques are scaled and then the back-propagation algorithm is applied as the learning rule so the weights of both neural networks are adjusted to minimize the cost functions. Once the cost functions are minimized and the errors converge to zero, it is said that the weights of both neural networks have been properly updated. Nonetheless, since the weights are updated using scaled data, these possess scaled values and therefore the outputs of both neural networks are also scaled. Then, the expression from Equation 4.71 is performed in order to descale the values of the weights and therefore be able to handle all normal data, that is without scale.

Force and torque equations are equivalent to Equation 4.71 as follows:

$$\begin{aligned} F = C_F \omega^2 & \leftrightarrow y = w^{(2)} w^{(1)} \left( \frac{k_x^2}{k_d} \right) x^2 \\ Q = C_Q \omega^2 & \leftrightarrow y = w^{(2)} w^{(1)} \left( \frac{k_x^2}{k_d} \right) x^2 \end{aligned} \quad (4.72)$$

therefore, the aerodynamic coefficients are calculated based each neural networks descaled weights:

$$C_F = w^{(2)}w^{(1)} \begin{pmatrix} \frac{k_x^2}{k_d} \end{pmatrix} \quad (4.73)$$

$$C_Q = w^{(2)}w^{(1)} \begin{pmatrix} \frac{k_x^2}{k_d} \end{pmatrix}$$

Using the thrust and torque data from Chapter 3 shown in Figure 3.22, two neural networks are trained as previously outlined, considering four neurons in the hidden layer of each. The data is processed according to the Batch method, that is, updating the weights once for all training data. Then, the quadrotor parameters  $C_F$  and  $C_Q$  are estimated by means of Equations 4.73. The values of the parameters are presented at the end of this section. For additional data such as weights, scaling factors, and learning rates refer to Appendix A.1.

Figures 4.18 and 4.19 show the neural network thrust and torque models of the rotor, respectively. Note in Figures 4.18a and 4.19a the evolution of the cost function (Mean Squared Error) through the training epochs, where it can be seen that in both cases the error converged to zero around epoch 500. On the other hand, Figures 4.18b and 4.19b show the output of the neural network (thrust and torque models) in comparison with the experimental data, according to each case.

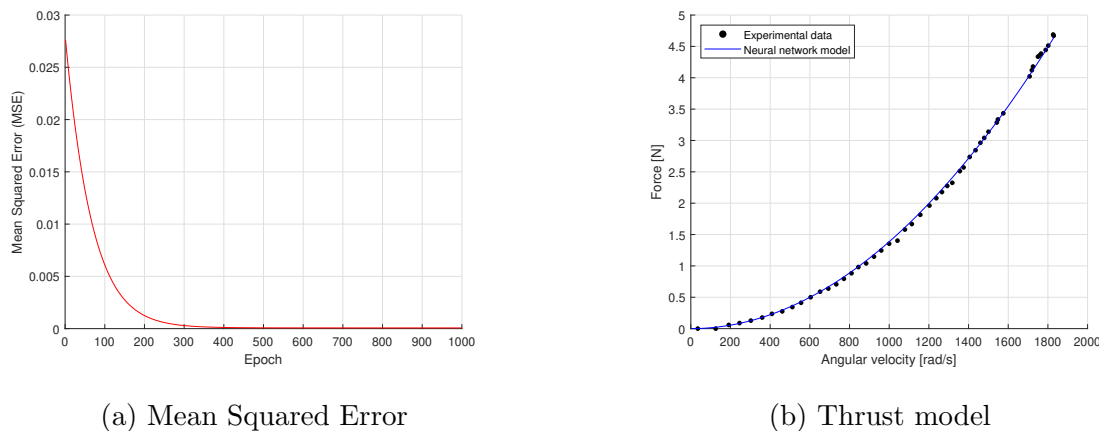


Figure 4.18: Neural network thrust rotor model

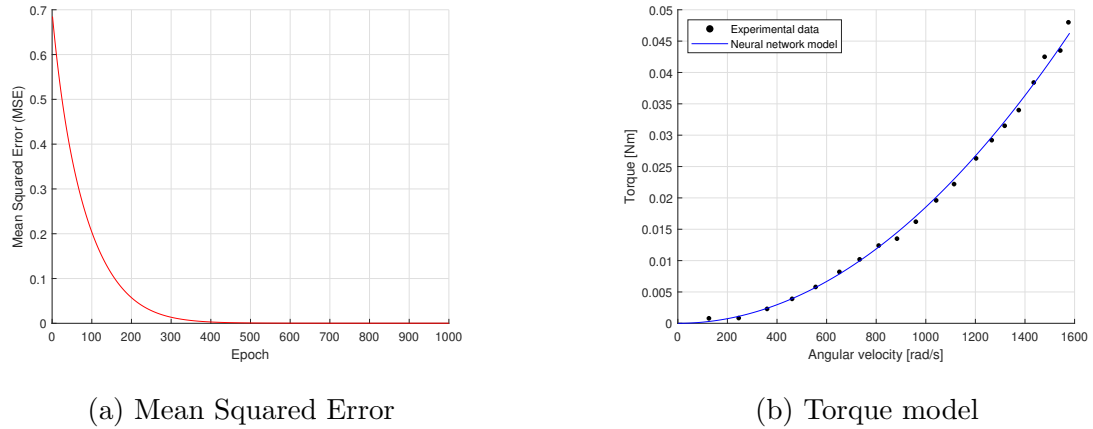


Figure 4.19: Neural network torque rotor model

As additional information, Figure 4.20 shows how the number of neurons in the hidden layer affects the learning process of the neural network. Note that as the number of neurons increases, the Mean Squared Error (MSE) converges to zero faster, that is, requiring fewer training epochs. This is a general behavior since the initial conditions of the weights (which are randomly assigned) also affect the learning process.

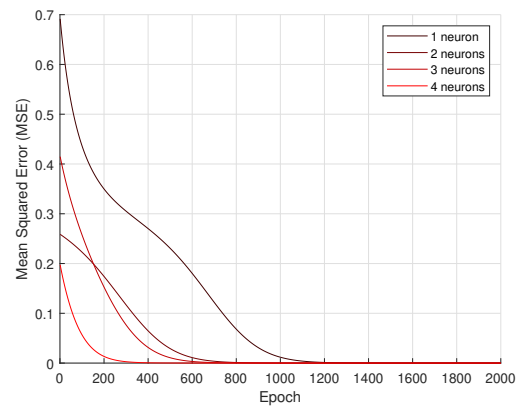


Figure 4.20: Effect of neurons in the hidden layer over MSE



### 4.3.2 NEURAL NETWORK FOR ESTIMATION OF INERTIAL PARAMETERS

The proposed neural network architecture to estimate the quadrotor inertial parameters ( $J_{xx}$ ,  $J_{yy}$ , and  $J_{zz}$ ) is shown in Figure 4.21. Note that it is a single-layer neural network that receives six inputs that represent the quadrotor angular velocities multiplications ( $qr$ ,  $pr$ , and  $pq$ ) and the actuators torques ( $\tau_\phi$ ,  $\tau_\theta$ , and  $\tau_\psi$ ) and outputs the angular accelerations ( $\dot{p}$ ,  $\dot{q}$ , and  $\dot{r}$ ). Also note that for this scenario, not all the neural network nodes are connected to each other.

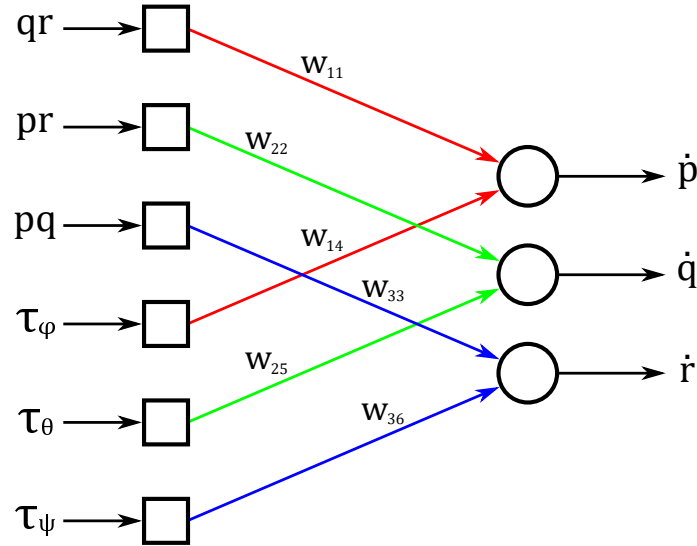


Figure 4.21: Neural network that consists of six input and three output nodes

The input vector  $x$ , the output vector  $y$ , as well as the weights matrix  $w$  are defined as follows:

$$\begin{aligned}
 x &= [qr \quad pr \quad pq \quad \tau_\phi \quad \tau_\theta \quad \tau_\psi]^T \\
 y &= [\dot{p} \quad \dot{q} \quad \dot{r}]^T \\
 w &= \begin{bmatrix} w_{11} & 0 & 0 & w_{14} & 0 & 0 \\ 0 & w_{22} & 0 & 0 & w_{25} & 0 \\ 0 & 0 & w_{33} & 0 & 0 & w_{36} \end{bmatrix}
 \end{aligned} \tag{4.74}$$

The weighted sums of the output nodes are calculated as:

$$v = wx \quad (4.75)$$

The neural network output is:

$$y = \varphi(v) \quad (4.76)$$

For linear activation function:

$$y = v \quad (4.77)$$

Therefore, the neural network output is expressed as:

$$y = wx \quad (4.78)$$

The training of the neural network is carried out using the data from the quadrotor attitude control simulation previously shown in Figure 4.9. The data is selected for training according to the Batch method. Then, the generalized delta rule is applied as the learning algorithm in order to minimize the cost function. Figure 4.22 shows the MSE cost function through the epochs. Note that for  $\dot{p}$  and  $\dot{q}$  the error converged to zero in a more prompt manner compared to  $\dot{r}$  which converged at approximately epoch 20,000.

Recalling the expressions of the quadrotor angular accelerations from Equation 4.19 and according to the structure of the proposed neural network, the following equivalences are stated:

$$\begin{aligned} \dot{p} = w_{11}qr + w_{14}\tau_\phi &\leftrightarrow \dot{p} = \left(\frac{J_{yy} - J_{zz}}{J_{xx}}\right)qr + \left(\frac{1}{J_{xx}}\right)\tau_\phi \\ \dot{q} = w_{22}pr + w_{25}\tau_\theta &\leftrightarrow \dot{q} = \left(\frac{J_{zz} - J_{xx}}{J_{yy}}\right)pr + \left(\frac{1}{J_{yy}}\right)\tau_\theta \\ \dot{r} = w_{33}pq + w_{36}\tau_\psi &\leftrightarrow \dot{r} = \left(\frac{J_{xx} - J_{yy}}{J_{zz}}\right)pq + \left(\frac{1}{J_{zz}}\right)\tau_\psi \end{aligned} \quad (4.79)$$

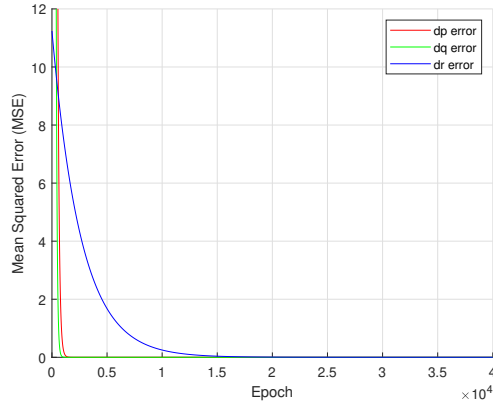


Figure 4.22: Mean Squared Error through epochs on inertial parameters estimation

The principal moments of inertia are then estimated as:

$$\begin{aligned}
 J_{xx} &= \frac{1}{w_{14}} \\
 J_{yy} &= \frac{1}{w_{25}} \\
 J_{zz} &= \frac{1}{w_{36}}
 \end{aligned} \tag{4.80}$$

The neural network weights and learning rate are presented in Appendix A.1.

### 4.3.3 ESTIMATED PARAMETERS AND EVALUATION

The aerodynamic and inertial parameters estimated with the application of neural networks are presented in Table 4.3.

In order to evaluate the results, a simulation of the quadrotor mathematical model with the estimated parameters is carried out using MATLAB<sup>®</sup> Simulink<sup>®</sup> software. The simulation consists of the same attitude control system from Figure 4.8, nevertheless, the reference signals are changed to Equation 4.81. The results are presented in Figure 4.23.

Table 4.3: Quadrotor parameters (grey box approach)

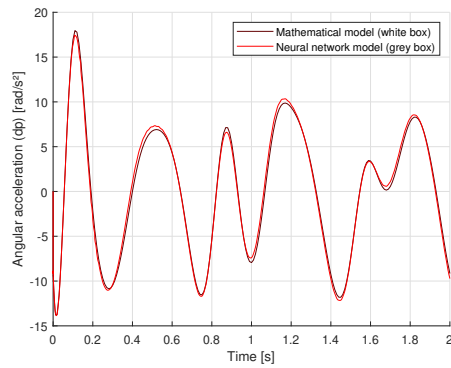
Parameter	Value	Unit	Technique for determining
$m$	0.652	kg	Direct measuring
$d$	0.0884	m	Direct measuring
$C_F$	$1.3859 \times 10^{-6}$	N·s <sup>2</sup>	Neural network parameter estimation
$C_Q$	$1.8534 \times 10^{-8}$	N·m·s <sup>2</sup>	Neural network parameter estimation
$J_{xx}$	$2.4903 \times 10^{-3}$	kg·m <sup>2</sup>	Neural network parameter estimation
$J_{yy}$	$2.5756 \times 10^{-3}$	kg·m <sup>2</sup>	Neural network parameter estimation
$J_{zz}$	$4.5862 \times 10^{-3}$	kg·m <sup>2</sup>	Neural network parameter estimation

$$\begin{aligned}
F_{T_d} &= 5 \\
\phi_d(t) &= \frac{\pi}{45} \sin(10t) \\
\theta_d(t) &= \frac{\pi}{45} \sin\left(10t + \frac{\pi}{2}\right) \\
\psi_d(t) &= \frac{2\pi}{45} \sin(2t)
\end{aligned} \tag{4.81}$$

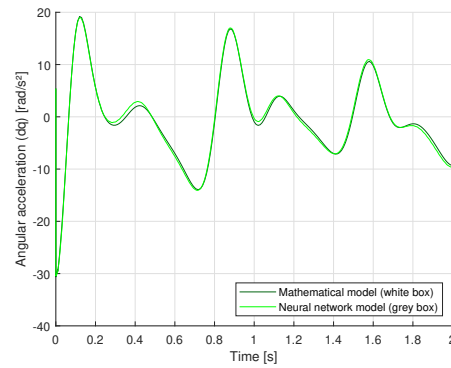
## 4.4 DATA-DRIVEN IDENTIFICATION (BLACK BOX APPROACH)

In the black box approach, the quadrotor is treated as an entirely unknown system where it is assumed that there is no direct relation between its inputs and outputs, reason why it is called black box since it cannot be seen what the box contains that causes the outputs with certain inputs.

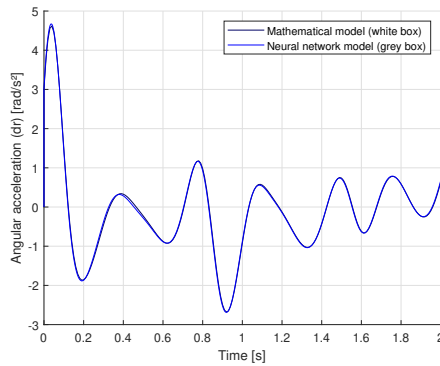
This section attempts to predict the forces and moments exerted by the quadrotor according to the thrust force of each motor and the attitude of the aircraft, without any knowledge of the internal system architecture. This is done by means of a data-driven method, that employs the quadrotor experimental data for training a neural network and thus to predict the desired values.



(a) Angular acceleration (dp)



(b) Angular acceleration (dq)



(c) Angular acceleration (dr)

Figure 4.23: Comparison of quadrotor angular accelerations between white box and grey box approaches

## 4.4.1 NEURAL NETWORK FOR DATA PREDICTION

The neural network proposed for the black box approach intends to predict the constant values of forces and moments exerted by the quadrotor. The architecture of such neural network is shown in Figure 4.24, which has six input nodes that represent the motor forces ( $F_1$ ,  $F_2$ ,  $F_3$ , and  $F_4$ ) and the aircraft roll ( $\phi$ ) and pitch ( $\theta$ ) angles; it also possesses six output nodes that represent the forces ( $F_x$ ,  $F_y$ , and  $F_z$ ) and moments ( $M_x$ ,  $M_y$ , and  $M_z$ ) exerted by the quadrotor; lastly, two hidden layers are proposed with  $n$  and  $m$  nodes, according to each.

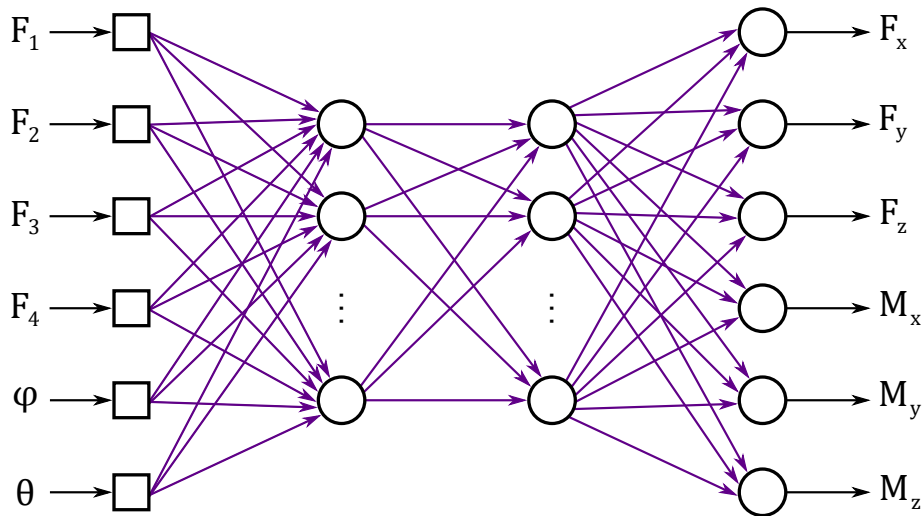


Figure 4.24: Neural network that consists of six input and output nodes and two hidden layers of  $n$  and  $m$  nodes

The training of the neural network is carried out employing the back-propagation algorithm and its functional process is just like the one already detailed in subsection 4.2.5, but with slight differences in the number of input and output nodes, as well as in the number of hidden layers. For this case, the activation functions in the hidden layers are specified as sigmoid and in the output layer as linear. Moreover, the training data is processed as the Batch method, which updates all the neural network weights once for all training data. The training of the neural network employs the data obtained from the experiments presented in Chapter 3. The signals of the motors used for the experimentation are presented in Appendix A.2.

#### 4.4.2 DATA PREDICTION EVALUATION

During the training process of the neural network the number of neurons in both hidden layers is changed and satisfactory results are obtained with a number of 20 neurons in each hidden layer. The learning rate is set to 0.1 and a number of 50,000 epochs are performed. Figure 4.25 shows the MSE of all six output variables through training epochs.

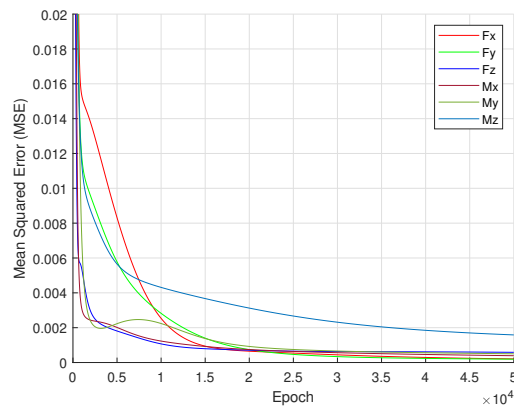
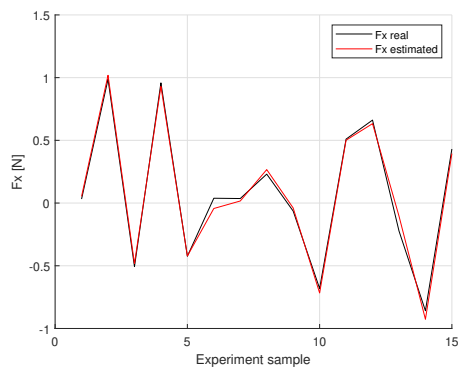


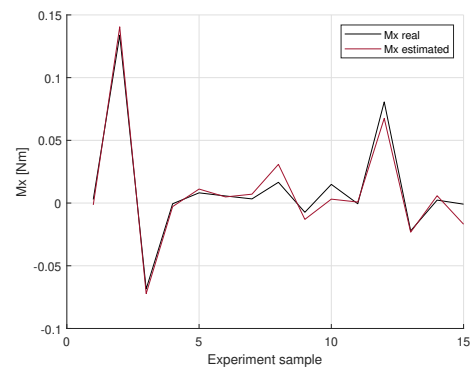
Figure 4.25: Mean Squared Error through epochs on data-driven identification

Once the neural network is properly trained, the evaluation of its performance is carried out. A number of 15 experiment samples are collected from the experimental tests (not including the training data) and the evaluation is conducted. The results are presented in Figure 4.26, where it can be observed the comparison between the real data and the neural network outputs, the latter showing satisfactory results in most of the data samples.

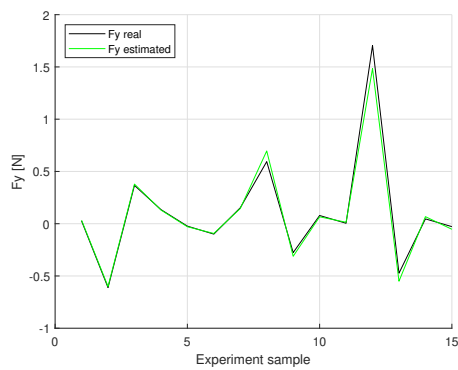
It is important to highlight that this neural network only predicts the static forces and moments exerted by the quadrotor. Therefore, the quadrotor dynamics remain uncertain since the current neural network architecture is not designed to capture it. Nonetheless, the results presented in this subsection provide a promising overview of the quadrotor behavior, even without any prior knowledge of the system.



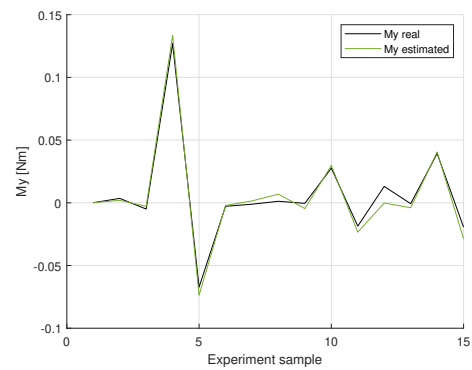
(a)  $F_x$



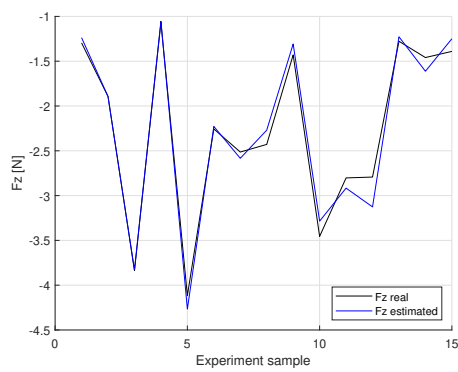
(b)  $M_x$



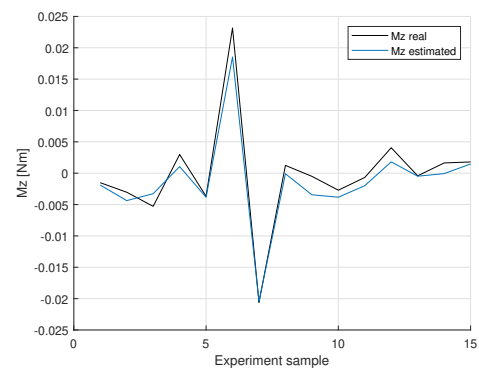
(c)  $F_y$



(d)  $M_y$



(e)  $F_z$



(f)  $M_z$

Figure 4.26: Comparison between real and estimated forces and moments from data-driven identification



## CHAPTER 5

# CONCLUSIONS

---

This research work addressed multiple subjects in diverse areas of engineering, from which the conclusions are presented below.

A low-cost UAV testing bench was designed and developed to measure forces and moments exerted by a UAV, nevertheless, the results showed reliable accuracy in one degree of freedom only, which is the force in the  $z$  axis. For this reason, the first hypothesis of this research work was not achieved, therefore, a mini force-torque sensor was acquired and an experimental base was created in order to meet the objectives.

The design and construction of a quadrotor from scratch was carried out and its four motors could successfully be characterized and controlled independently. Consequently, experimental tests with the quadrotor and experimental base were performed effectively. Thus, forces and moments data were acquired and stored for the system identification process.

The quadrotor system identification was carried out according to each proposed method which are; white, grey, and black box approaches. The white box approach consisted of deriving the quadrotor equations of motion and simulating its dynamics. Conversely, in the grey box approach the parameters from the mathematical model were obtained by training a neural network with experimental and simulation data.

---

Lastly, the black box approach focused only on the experimental data in order to train a neural network and thus identify the quadrotor system. In all three methods, the obtained results showed good precision and accuracy. Thus, this research work was successfully concluded.

## APPENDIX A

# APPENDIX

---

### A.1 NEURAL NETWORKS DATA

In this appendix, the quadrotor neural network parameters from the thrust, torque, and inertial models obtained in Chapter 4 are presented.

#### A.1.1 THRUST MODEL

$$\alpha = 0.01$$

$$k_d = 0.2133$$

$$k_x = 5.4669 \times 10^{-4}$$

$$w^{(1)} = \begin{bmatrix} -0.5311 & 0.1014 & -1.0462 & -0.1708 \end{bmatrix}^T$$

$$w^{(2)} = \begin{bmatrix} -0.4342 & -0.6602 & -0.7606 & -0.1731 \end{bmatrix}$$

### A.1.2 TORQUE MODEL

$$\alpha = 0.01$$

$$k_d = 20.8333$$

$$k_x = 6.3486 \times 10^{-4}$$

$$w^{(1)} = \begin{bmatrix} -0.4279 & 1.0063 & -0.8083 & -0.4139 \end{bmatrix}^T$$

$$w^{(2)} = \begin{bmatrix} -0.3693 & 0.5336 & -0.0600 & -0.5183 \end{bmatrix}$$

### A.1.3 INERTIAL MODEL

$$\alpha = 0.8$$

$$w = \begin{bmatrix} -1.7592 & 0 & 0 & 401.5602 & 0 & 0 \\ 0 & 0.0076 & 0 & 0 & 388.2557 & 0 \\ 0 & 0 & 4.0810 & 0 & 0 & 218.0455 \end{bmatrix}$$

## A.2 QUADROTOR EXPERIMENTS DATA

In this appendix, all the experiments that were performed with the quadrotor and presented in Chapter 3 are listed.

Table A.1: Quadrotor motors signals for dynamics simulation tests

Experiment	Motor 1			Motor 2			Motor 3			Motor 4			Quadrotor Dynamics
	Signal	Max Value	Frequency	Signal	Max Value	Frequency	Signal	Max Value	Frequency	Signal	Max Value	Frequency	
1	Trapezoidal	0.1	N/A	Trapezoidal	0.1	N/A	Trapezoidal	0.1	N/A	Trapezoidal	0.1	N/A	Hover
2	Trapezoidal	0.15	N/A	Trapezoidal	0.15	N/A	Trapezoidal	0.15	N/A	Trapezoidal	0.15	N/A	Hover
3	Trapezoidal	0.2	N/A	Trapezoidal	0.2	N/A	Trapezoidal	0.2	N/A	Trapezoidal	0.2	N/A	Hover
4	Trapezoidal	0.05	N/A	Trapezoidal	0.2	N/A	Trapezoidal	0.2	N/A	Trapezoidal	0.05	N/A	Roll +
5	Trapezoidal	0.1	N/A	Trapezoidal	0.2	N/A	Trapezoidal	0.2	N/A	Trapezoidal	0.1	N/A	Roll +
6	Trapezoidal	0.15	N/A	Trapezoidal	0.2	N/A	Trapezoidal	0.2	N/A	Trapezoidal	0.15	N/A	Roll +
7	Trapezoidal	0.2	N/A	Trapezoidal	0.05	N/A	Trapezoidal	0.05	N/A	Trapezoidal	0.2	N/A	Roll -
8	Trapezoidal	0.2	N/A	Trapezoidal	0.1	N/A	Trapezoidal	0.1	N/A	Trapezoidal	0.2	N/A	Roll -
9	Trapezoidal	0.2	N/A	Trapezoidal	0.15	N/A	Trapezoidal	0.15	N/A	Trapezoidal	0.2	N/A	Roll -
10	Trapezoidal	0.2	N/A	Trapezoidal	0.2	N/A	Trapezoidal	0.05	N/A	Trapezoidal	0.05	N/A	Pitch +
11	Trapezoidal	0.2	N/A	Trapezoidal	0.2	N/A	Trapezoidal	0.1	N/A	Trapezoidal	0.1	N/A	Pitch +
12	Trapezoidal	0.2	N/A	Trapezoidal	0.2	N/A	Trapezoidal	0.15	N/A	Trapezoidal	0.15	N/A	Pitch +
13	Trapezoidal	0.05	N/A	Trapezoidal	0.05	N/A	Trapezoidal	0.2	N/A	Trapezoidal	0.2	N/A	Pitch -
14	Trapezoidal	0.1	N/A	Trapezoidal	0.1	N/A	Trapezoidal	0.2	N/A	Trapezoidal	0.2	N/A	Pitch -
15	Trapezoidal	0.15	N/A	Trapezoidal	0.15	N/A	Trapezoidal	0.2	N/A	Trapezoidal	0.2	N/A	Pitch -
16	Trapezoidal	0.2	N/A	Trapezoidal	0.05	N/A	Trapezoidal	0.2	N/A	Trapezoidal	0.05	N/A	Yaw+
17	Trapezoidal	0.2	N/A	Trapezoidal	0.1	N/A	Trapezoidal	0.2	N/A	Trapezoidal	0.1	N/A	Yaw+
18	Trapezoidal	0.2	N/A	Trapezoidal	0.15	N/A	Trapezoidal	0.2	N/A	Trapezoidal	0.15	N/A	Yaw+
19	Trapezoidal	0.05	N/A	Trapezoidal	0.2	N/A	Trapezoidal	0.05	N/A	Trapezoidal	0.2	N/A	Yaw-
20	Trapezoidal	0.1	N/A	Trapezoidal	0.2	N/A	Trapezoidal	0.1	N/A	Trapezoidal	0.2	N/A	Yaw-
21	Trapezoidal	0.15	N/A	Trapezoidal	0.2	N/A	Trapezoidal	0.15	N/A	Trapezoidal	0.2	N/A	Yaw-
22	Sine wave	0.2	0.25 Hz	Sine wave	0.2	0.25 Hz	Sine wave	0.2	0.25 Hz	Sine wave	0.2	0.25 Hz	Bounce
23	Sine wave	0.2	0.4 Hz	Sine wave	0.2	0.4 Hz	Sine wave	0.2	0.4 Hz	Sine wave	0.2	0.4 Hz	Bounce
24	Sine wave	0.2	0.5 Hz	Sine wave	0.2	0.5 Hz	Sine wave	0.2	0.5 Hz	Sine wave	0.2	0.5 Hz	Bounce
25	Trapezoidal	0.1	N/A	Trapezoidal	0.1	N/A	Trapezoidal	0.1	N/A	Trapezoidal	0.1	N/A	15° Right
26	Trapezoidal	0.15	N/A	Trapezoidal	0.15	N/A	Trapezoidal	0.15	N/A	Trapezoidal	0.15	N/A	15° Right
27	Trapezoidal	0.2	N/A	Trapezoidal	0.2	N/A	Trapezoidal	0.2	N/A	Trapezoidal	0.2	N/A	15° Right
28	Trapezoidal	0.1	N/A	Trapezoidal	0.1	N/A	Trapezoidal	0.1	N/A	Trapezoidal	0.1	N/A	-15° Left
29	Trapezoidal	0.15	N/A	Trapezoidal	0.15	N/A	Trapezoidal	0.15	N/A	Trapezoidal	0.15	N/A	-15° Left
30	Trapezoidal	0.2	N/A	Trapezoidal	0.2	N/A	Trapezoidal	0.2	N/A	Trapezoidal	0.2	N/A	-15° Left
31	Trapezoidal	0.1	N/A	Trapezoidal	0.1	N/A	Trapezoidal	0.1	N/A	Trapezoidal	0.1	N/A	15° Backward
32	Trapezoidal	0.15	N/A	Trapezoidal	0.15	N/A	Trapezoidal	0.15	N/A	Trapezoidal	0.15	N/A	15° Backward
33	Trapezoidal	0.2	N/A	Trapezoidal	0.2	N/A	Trapezoidal	0.2	N/A	Trapezoidal	0.2	N/A	15° Backward
34	Trapezoidal	0.1	N/A	Trapezoidal	0.1	N/A	Trapezoidal	0.1	N/A	Trapezoidal	0.1	N/A	-15° Forward
35	Trapezoidal	0.15	N/A	Trapezoidal	0.15	N/A	Trapezoidal	0.15	N/A	Trapezoidal	0.15	N/A	-15° Forward
36	Trapezoidal	0.2	N/A	Trapezoidal	0.2	N/A	Trapezoidal	0.2	N/A	Trapezoidal	0.2	N/A	-15° Forward
37	Trapezoidal	0.1	N/A	Trapezoidal	0.1	N/A	Trapezoidal	0.1	N/A	Trapezoidal	0.1	N/A	30° Right
38	Trapezoidal	0.15	N/A	Trapezoidal	0.15	N/A	Trapezoidal	0.15	N/A	Trapezoidal	0.15	N/A	30° Right
39	Trapezoidal	0.2	N/A	Trapezoidal	0.2	N/A	Trapezoidal	0.2	N/A	Trapezoidal	0.2	N/A	30° Right
40	Trapezoidal	0.1	N/A	Trapezoidal	0.1	N/A	Trapezoidal	0.1	N/A	Trapezoidal	0.1	N/A	-30° Left
41	Trapezoidal	0.15	N/A	Trapezoidal	0.15	N/A	Trapezoidal	0.15	N/A	Trapezoidal	0.15	N/A	-30° Left
42	Trapezoidal	0.2	N/A	Trapezoidal	0.2	N/A	Trapezoidal	0.2	N/A	Trapezoidal	0.2	N/A	-30° Left
43	Trapezoidal	0.1	N/A	Trapezoidal	0.1	N/A	Trapezoidal	0.1	N/A	Trapezoidal	0.1	N/A	30° Backward
44	Trapezoidal	0.15	N/A	Trapezoidal	0.15	N/A	Trapezoidal	0.15	N/A	Trapezoidal	0.15	N/A	30° Backward
45	Trapezoidal	0.2	N/A	Trapezoidal	0.2	N/A	Trapezoidal	0.2	N/A	Trapezoidal	0.2	N/A	30° Backward
46	Trapezoidal	0.1	N/A	Trapezoidal	0.1	N/A	Trapezoidal	0.1	N/A	Trapezoidal	0.1	N/A	-30° Forward
47	Trapezoidal	0.15	N/A	Trapezoidal	0.15	N/A	Trapezoidal	0.15	N/A	Trapezoidal	0.15	N/A	-30° Forward
48	Trapezoidal	0.2	N/A	Trapezoidal	0.2	N/A	Trapezoidal	0.2	N/A	Trapezoidal	0.2	N/A	-30° Forward

# REFERENCES

---

- [1] DJI. DJI Inspire 3. Available on [www.dji.com](http://www.dji.com). 2024.
- [2] Autel Robotics. DEVO Lite Series. Available on [www.autelrobotics.com](http://www.autelrobotics.com). 2023.
- [3] Hubsan. H107D+ X4 FPV PLUS. Available on [www.hubsan.com](http://www.hubsan.com).
- [4] Fernando H. C. T. E., De Silva A. T. A., De Zoysa M. D. C., Dilshan K. A. D. C., and Munasinghe S. R. “Modelling, simulation and implementation of a quadrotor UAV”. IEEE 8th International conference on industrial and information systems, pp. 207-212. IEEE. 2013.
- [5] Hoffmann G., Waslander S., and Tomlin C. “Distributed cooperative search using information-theoretic costs for particle filters, with quadrotor applications”. AIAA Guidance, Navigation, and Control Conference and Exhibit, p. 6576. 2006.
- [6] Roldão V., Cunha R., Cabecinhas D., Silvestre C., and Oliveira P. “A leader-following trajectory generator with application to quadrotor formation flight”. Robotics and Autonomous Systems, 62(10), pp. 1597-1609. 2014.
- [7] Allen R. and Pavone M. “A real-time framework for kinodynamic planning with application to quadrotor obstacle avoidance”. AIAA Guidance, Navigation, and Control Conference, p. 1374. 2016.
- [8] Elmokadem T. “Distributed coverage control of quadrotor multi-UAV systems for precision agriculture”. IFAC-PapersOnLine, 52(30), pp. 251-256. 2019.

- 
- [9] Tellez-Belkotosky P. A., Cabriaes-Ramirez L. E., Gutierrez-Martinez M. A., and Ollervides-Vazquez E. J. “Intelligent PIV fuzzy navigation and attitude controller for an octorotor Mini-UAV”. *Machines*, 11(2), p. 266. 2023.
- [10] Ollervides-Vazquez E. J., Tellez-Belkotosky P. A., Santibañez V., Rojo-Rodriguez E. G., Reyes-Osorio L. A., and Garcia-Salazar O. “Modeling and Simulation of an Octorotor UAV with Manipulator Arm”. *Drones*, 7(3), p. 168. 2023.
- [11] Tellez-Belkotosky P. A., Ollervides-Vazquez E. J., Rojo-Rodriguez E. G., Santillan-Avila J. L., Cabriaes-Ramirez L. E., Gutierrez-Martinez M. A., and Garcia-Salazar O. “Nonlinear flight navigation controller for an octorotor unmanned aerial vehicle”. In *2021 International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE)*, pp. 52-59. IEEE. 2021.
- [12] Liang Q., Zhang D., Coppola G., Wang Y., Wei S., and Ge Y. “Multi-dimensional MEMS/micro sensor for force and moment sensing: A review”. *IEEE Sensors Journal*, 14(8), pp. 2643-2657. 2014.
- [13] Tavakolpour-Saleh A. R. and Sadeghzadeh M. R. “Design and development of a three-component force/moment sensor for underwater hydrodynamic tests”. *Sensors and Actuators A: Physical*, 216, pp. 84-91. 2014.
- [14] Liu T. “Design of a three-dimensional capacitor-based six-axis force sensor for human-robot interaction”. *Sensors and Actuators A: Physical*, 331, p. 112939. 2021.
- [15] Akbari H. and Kazerooni A. “Improving the coupling errors of a Maltese cross-beams type six-axis force/moment sensor using numerical shape-optimization technique”. *Measurement*, 126, pp. 342-355. 2018.
- [16] Fries F., Win S. K. H., Tang E., Low J. E., Win L. S. T., y Alvarado P. V., and Foong S. “Design and implementation of a compact rotational speed and air

- flow sensor for unmanned aerial vehicles”. *IEEE Sensors Journal*, 19(22), pp. 10298-10307. 2019.
- [17] Zhang H., Jia Y., Guo Y., Qian K., Song A., and Xi N. “Online sensor information and redundancy resolution based obstacle avoidance for high dof mobile manipulator teleoperation”. *International Journal of Advanced Robotic Systems*, 10(5), p. 244. 2013.
- [18] Payo I., Adánez J. M., Rosa D. R., Fernandez R., and Vazquez A. S. “Six-axis column-type force and moment sensor for robotic applications”. *IEEE Sensors Journal*, 18(17), pp. 6996-7004. 2018.
- [19] Jacobs D. A. and Ferris D. P. “Estimation of ground reaction forces and ankle moment with multiple, low-cost sensors”. *Journal of neuroengineering and rehabilitation*, 12(1), pp. 1-12. 2015.
- [20] Tlatelpa-Osorio Y. E., Corona-Sánchez J. J., and Rodríguez-Cortés H. “Quadrotor control based on an estimator of external forces and moments”. *International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 957-963. IEEE. 2016.
- [21] Papachristos C., Alexis K., and Tzes A. “Efficient force exertion for aerial robotic manipulation: Exploiting the thrust-vectoring authority of a tri-tiltrotor UAV”. *IEEE international conference on robotics and automation (ICRA)*, pp. 4500-4505. IEEE. 2014.
- [22] Strachan R., Knowles K., Lawson N. J., and Finnis M. V. “Force and moment measurements for a generic car model in proximity to a side wall”. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of automobile engineering*, 226(10), pp. 1352-1364. 2012.
- [23] Tyto Robotics. Flight Stand 50. Available on [www.tytorobotics.com](http://www.tytorobotics.com). 2024.
- [24] Deshpande M. S., Jawale H. P., and Thorat H. T. “Development, calibration and testing of three axis force sensor”. *International Conference on Mechanical and Aerospace Engineering (ICMAE)*, pp. 285-289. IEEE. 2016.



- [25] Yuan C., Luo L. P., Yuan Q., Wu J., Yan R. J., Kim H., Shin K. S., and Han C. S. “Development and evaluation of a compact 6-axis force/moment sensor with a serial structure for the humanoid robot foot”. *Measurement*, 70, pp. 110-122. 2015.
- [26] Park J. Y., Shim H., Jun B. H., Lee P. M., Yoo S. Y., and Baek H. “Measurement of hydrodynamic forces and moment acting on Crabster, CR200 using model tests”. *IEEE Underwater Technology (UT)*, pp. 1-5. IEEE. 2017.
- [27] Huang B., Tao J., Yi J., Wang X., Li C., and Chen S. “Improvement for the stability of an air-lubricated six-axis force/moment sensor”. *The International Journal of Advanced Manufacturing Technology*, 92(1), pp. 715-721. 2017.
- [28] Kim C. and Lee C. H. “Development of a 6-DoF FBG force–moment sensor for a haptic interface with minimally invasive robotic surgery”. *Journal of Mechanical Science and Technology*, 30(8), pp. 3705-3712. 2016.
- [29] Zhang W., Lua K. B., Senthil K. A., Lim T. T., Yeo K. S., and Zhou G. “Design and characterization of a novel T-shaped multi-axis piezoresistive force/moment sensor”. *IEEE Sensors Journal*, 16(11), pp. 4198-4210. 2016.
- [30] Ghani J. A., Jye P. S., Haron C. H. C., Rizal M., and Nuawi M. Z. “Determination of sensor location for cutting tool deflection using finite element method simulation”. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 226(9), pp. 2373-2377. 2012.
- [31] Kim J. H. “Multi-axis force-torque sensors for measuring zero-moment point in humanoid robots: A review”. *IEEE Sensors Journal*, 20(3), pp. 1126-1141. 2019.
- [32] Lin G., Pang H., Zhang W., Wang D., and Feng L. “A self-decoupled three-axis force sensor for measuring the wheel force”. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 228(3), pp. 319-334. 2014.

- 
- [33] Yu Y. and Ding X. “A quadrotor test bench for six degree of freedom flight”. *Journal of Intelligent & Robotic Systems*, 68(3), pp. 323-338. 2012.
- [34] Legowo A., Sulaeman E., and Rosli D. “Review on system identification for quadrotor unmanned aerial vehicle (UAV)”. *Advances in Science and Engineering Technology International Conferences (ASET)*, pp. 1-8. IEEE. 2019.
- [35] Tan K. C. and Li Y. “Evolutionary system identification in the time domain”. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 211(5), pp. 319-323. 1997.
- [36] Tischler M. B. and Remple R. K. *Aircraft and rotorcraft system identification*. American Institute of Aeronautics and Astronautics. 2006.
- [37] MathWorks. *Mathematical Modeling*. Available on [www.mathworks.com/solutions/mathematical-modeling.html](http://www.mathworks.com/solutions/mathematical-modeling.html). 2024.
- [38] Sjöberg J., Zhang Q., Ljung L., Benveniste A., Delyon B., Glorennec P. Y., Hjalmarsson H., and Juditsky A. “Nonlinear black-box modeling in system identification: a unified overview”. *Automatica*, 31(12), pp. 1691-1724. 1995.
- [39] B. Douglas. *Capabilities for Modeling Dynamic Systems*. Available on [www.mathworks.com/solutions/control-systems/modeling-dynamic-systems.html](http://www.mathworks.com/solutions/control-systems/modeling-dynamic-systems.html). 2023.
- [40] Nugroho L. and Akmelawati R. “Comparison of black-grey-white box approach in system identification of a flight vehicle”. *Journal of Physics: Conference Series*, vol. 1130, no. 1, p. 012024. IOP Publishing. 2018.
- [41] Pashayev A. and Sabziev E. “Refinement of the parameters of a mathematical model of quadcopter dynamics”. *Scientific Journal of Silesian University of Technology. Series Transport*. 109, pp. 141-151. 2020.
- [42] Amezcuita-Brooks L., Liceaga-Castro E., Gonzalez-Sanchez M., Garcia-Salazar O., and Martinez-Vazquez D. “Towards a standard design model for quad-

- rotors: A review of current models, their accuracy and a novel simplified model”. *Progress in Aerospace Sciences*, 95, pp. 1-23. 2017.
- [43] Abas N., Legowo A., Ibrahim Z., Rahim N., and Kassim A. M. “Modeling and system identification using extended kalman filter for a quadrotor system”. *Applied Mechanics and Materials*, 313, pp. 976-981. 2013.
- [44] Yuksek B., Saldiran E., Cetin A., Yeniceri R., and Inalhan G. “System identification and model-based flight control system design for an agile maneuvering quadrotor platform”. *AIAA Scitech 2020 Forum*, p. 1835. 2020.
- [45] Sun S., de Visser C. C., and Chu Q. “Quadrotor gray-box model identification from high-speed flight data”. *Journal of Aircraft*, 56(2), pp. 645-661. 2019.
- [46] Amezcua-Brooks L., Hernandez-Alcantara D., Santana-Delgado C., Covarrubias-Fabela R., Garcia-Salazar O., and Ramirez-Mendoza A. M. “Improved model for micro-UAV propulsion systems: Characterization and applications”. *IEEE Transactions on Aerospace and Electronic Systems*, 56(3), pp. 2174-2197. 2019.
- [47] Bishop C. M. and Chris M. “Neural networks and their applications”. *Review of Scientific Instruments*. 1994.
- [48] Yegnanarayana B. *Artificial neural networks*. PHI Learning Pvt. Ltd. 2009.
- [49] Pramoditha R. The Concept of Artificial Neurons (Perceptrons) in Neural Networks. Available on [www.medium.com](http://www.medium.com). 2021.
- [50] Tutunji T. A. “Parametric system identification using neural networks”. *Applied Soft Computing*, 47, pp. 251-261. 2016.
- [51] Mendoza A. M. E. R., Fabela J. R. C., Amezcua-Brooks L. A., and Alcántara D. H. “Parameter identification using fuzzy neurons: Application to drones and induction motors”. *Dyna (Spain)*, 93(1), pp. 75-81. 2018.

- 
- [52] Li Q., Qian J., Zhu Z., Bao X., Helwa M. K., and Schoellig A. P. “Deep neural networks for improved, impromptu trajectory tracking of quadrotors”. IEEE International Conference on Robotics and Automation (ICRA), pp. 5183-5189. IEEE. 2017.
- [53] Amiruddin B. P., Iskandar E., Fatoni A., and Santoso A. “Deep learning based system identification of quadcopter unmanned aerial vehicle”. 3rd International Conference on Information and Communications Technology (ICOIACT), pp. 165-169. IEEE. 2020.
- [54] Pairan M. F. and Shamsudin S. S. “System identification of an unmanned quadcopter system using MRAN neural”. IOP conference series: materials science and engineering, vol. 270, no. 1, p. 012019. IOP Publishing. 2017.
- [55] Fabela R., Santana C., Naranjo A., Amezcua-Brooks L., Liceaga-Castro E., and Torres-Reyna M. “Experimental characterization of a small and micro unmanned aerial vehicle propulsion systems”. AIAA Atmospheric Flight Mechanics Conference, p. 1530. 2016.
- [56] Muller I., Machado R., Pereira C. E., and Brusamarello V. “Load Cells in Force Sensing Analysis - Theory and a Novel Application”. IEEE Instrumentation & Measurement Magazine. 2010.
- [57] Newton Weighing. Load Cell Column Type. Available on [www.newtonweighing.net/column-load-cell/](http://www.newtonweighing.net/column-load-cell/). 2019.
- [58] A-Tech Instruments Ltd. Load Cell and Force Sensor. Available on [www.a-tech.ca](http://www.a-tech.ca).
- [59] ANYLOAD. How Does a Load Cell Work?. Available on [www.anyload.com/how-does-a-load-cell-work/](http://www.anyload.com/how-does-a-load-cell-work/). 2024.
- [60] Chapra S. C. and Canale R. P. *Numerical methods for engineers*. McGraw-hill Education-Europe. 2014.

- 
- [61] Kim P. and Huh L. *Kalman filter for beginners: with MATLAB examples*. s.n. 2011.
- [62] Ollervides-Vazquez E. J., Rojo-Rodriguez E. G., Rojo-Rodriguez E. U., Cabriaes-Ramirez L. E., and Garcia-Salazar O. “Two-layer saturated PID controller for the trajectory tracking of a quadrotor UAV”. International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE), pp. 85-91. IEEE. 2020.
- [63] Tagay A., Omar A., and Ali M. H. “Development of control algorithm for a quadcopter”. *Procedia Computer Science*, 179, pp. 242-251. 2021.
- [64] Yedavalli R. K. *Flight dynamics and control of aero and space vehicles*. John Wiley & Sons. 2020.
- [65] Kim P. *Matlab deep learning: With machine learning, neural networks and artificial intelligence*. Apress. 2017.
- [66] Suryana L. Parameter Identification with Neural Network — Gray-Box Approach. Available on [www.lucaselbert.medium.com/parameter-identification-with-neural-network-grey-box-approach-41e72d76151](http://www.lucaselbert.medium.com/parameter-identification-with-neural-network-grey-box-approach-41e72d76151). 2022.
- [67] Muhammad I. and Yan Z. “Supervised Machine Learning Approaches: A Survey”. *ICTACT Journal on Soft Computing*, 5(3). 2015.
- [68] Nawi N. M., Khan A., and Rehman M. Z. “A new back-propagation neural network optimized with cuckoo search algorithm”. *Computational Science and Its Applications–ICCSA 2013: 13th International Conference, Ho Chi Minh City, Vietnam, June 24-27, 2013, Proceedings, Part I 13*, pp. 413-426. Springer Berlin Heidelberg. 2013.

# AUTOBIOGRAPHICAL SUMMARY

---

Alejandro Jiménez Flores

Candidate to obtain the degree of  
Master of Science in Aeronautical Engineering  
with Orientation in Flight Dynamics

Universidad Autónoma de Nuevo León  
Facultad de Ingeniería Mecánica y Eléctrica

Thesis:

APPLICATION OF AN INTELLIGENT ALGORITHM IN THE SYSTEM  
IDENTIFICATION OF A QUADROTOR

Alejandro Jiménez Flores was born in the city of Chihuahua, Chih. México on November 16th, 1997. In the year of 2016, he began his studies in aeronautical engineering at the Polytechnic University of Chihuahua and got his bachelor degree in 2020. He has experience in unmanned aerial vehicles research and has worked on projects in the areas of flight dynamics, control systems, computer-aided design and engineering, programming, additive manufacturing, and machine learning.