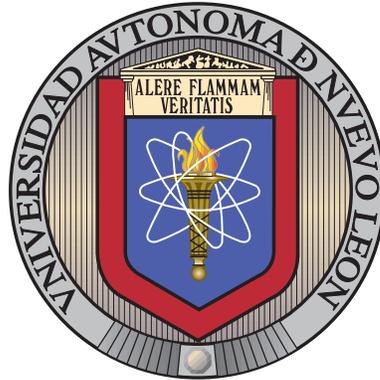


UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

SUBDIRECCIÓN DE ESTUDIOS DE POSGRADO



DESIGN, MODELING, AND CONTROL OF A
MICRO COAXIAL DRONE USING OPTIMAL
ADAPTIVE CONTROL BASED ON
REINFORCEMENT LEARNING

BY

ING. VÍCTOR HUGO DOMÍNGUEZ DE LA CRUZ

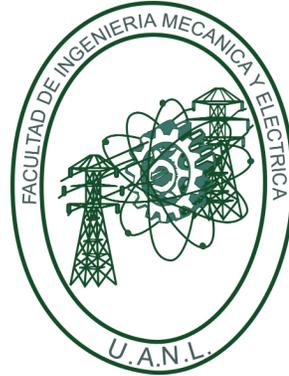
AS A PARTIAL REQUIREMENT TO OBTAIN THE DEGREE OF
MAESTRÍA EN INGENIERÍA EN AERONÁUTICA
CON ORIENTACIÓN EN DINÁMICA DE VUELO

DECEMBER 2022

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

SUBDIRECCIÓN DE ESTUDIOS DE POSGRADO



DESIGN, MODELING, AND CONTROL OF A
MICRO COAXIAL DRONE USING OPTIMAL
ADAPTIVE CONTROL BASED ON
REINFORCEMENT LEARNING

BY

ING. VÍCTOR HUGO DOMÍNGUEZ DE LA CRUZ

AS A PARTIAL REQUIREMENT TO OBTAIN THE DEGREE OF
MAESTRÍA EN INGENIERÍA EN AERONÁUTICA
CON ORIENTACIÓN EN DINÁMICA DE VUELO

DECEMBER 2022

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

Los miembros del Comité de Evaluación de Tesis recomendamos que la Tesis “Design, Modeling, and Control of a Micro Coaxial Drone Using Optimal Adaptive Control Based on Reinforcement Learning”, realizada por el estudiante Víctor Hugo Domínguez de la Cruz, con número de matrícula 1602046, sea aceptada para su defensa como requisito parcial para obtener el grado de Maestría en Ingeniería Aeronáutica con orientación en: Estructuras, Materiales, Dinámica de Vuelo.

El Comité de Evaluación de Tesis

Dr. Octavio García Salazar
Director

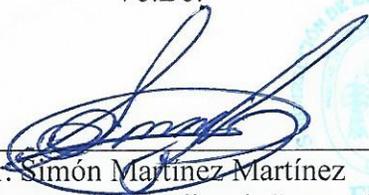
Dr. Luis Arturo Reyes Osorio
Co-director

Dr. Luis Antonio Amézquita Brooks
Revisor

Dr. Carlos Antonio Santana Delgado
Revisor

Dr. Efraín Ibarra Jiménez
Revisor

Vo.Bo.


Dr. Simón Martínez Martínez
Subdirector de Estudios de Posgrado

Institución 190001

Programa 563510

Acta Núm. 4169

Ciudad Universitaria, a 23 de Mayo del 2023

To my parents for the thrust ...

To my couple for the support ...

To my mentors for the inspiration ...

To me for the courage ...

CONTENTS

Abstract	xiv
1 Introduction	1
1.1 General Objectives	6
1.2 Specific Objectives	7
1.3 Scientific contributions	7
2 Design of Coaxial MUAV	8
3 Mathematical Model	14
3.1 Kinematics	14
3.2 Forces	16
3.3 Moments	17
3.4 Equations of Motion	19
4 Integral Reinforcement Learning (IRL) approach	24
4.1 Continuous-time adaptive critic solution for the linear quadratic reg- ulator	25

4.2	Integral Reinforcement Learning	26
4.3	Online implementation of IRL adaptive optimal control for MCR UAV V3.0	28
4.3.1	Recursive Least Squares	31
4.4	Observer for Non Full-State Feedback	36
4.5	Output disturbance for observer	40
5	Experimental Characterization	45
5.1	Manufacturing of specimens	45
5.2	Mechanical testing	47
5.3	Static aerodynamic characterization	49
5.3.1	Brushless rotor characterization	51
5.3.2	Ailerons characterization	52
6	Computational Procedure	55
6.1	Computational Fluid Dynamics (CFD)	55
6.2	Finite Element Analysis (FEA)	59
6.3	Optimal adaptive IRL	62
6.3.1	Optimal adaptive IRL considering output disturbance	67
7	MCR UAV V3.0 Manufacture and Flight Testing	73
7.1	Additive manufacturing	73
7.2	Vehicle assembly	75

7.3	Avionics assembly	76
7.4	Flight computer programming	77
7.5	Flight testing	79
8	Results and Discussion	81
8.1	Tensile test results	81
8.2	CFD results	84
8.3	FEM structural results	86
8.4	Optimal adaptive IRL results	91
8.4.1	IRL considering output disturbance results	92
8.4.2	IRL considering output perturbation (4 iters.) results	98
9	Conclusions	103
A	IRL Matlab Programming	106
B	Disturbed IRL Matlab Programming	113
C	Disturbed IRL Matlab Programming (4 iters.)	122

LIST OF FIGURES

1.1	Schematic of fuse deposition modeling (FDM) technology.	3
2.1	MCR UAV V3.0 main objective.	8
2.2	Flowchart used to MCR UAV V3.0 design.	9
2.3	MCR UAV components. a) 30A ESC b) control surfaces c) flight computer d) LiPo battery e) servomechanism f) coaxial blades g) control surfaces links.	9
2.4	Control surfaces topologies (units in mm). a) Concave deflected shape, b) optimized concave deflected shape, c) convex deflected surface and d) optimized convex deflected surface.	12
2.5	MCR UAV V3.0 Dimensions.	13
3.1	MCR UAV V3.0.	15
3.2	MCR UAV V3.0 forces.	17
4.1	IRL stages.	27
4.2	IRL stages considering formulations.	28
4.3	Flowchart for online IRL algorithm for CT linear systems.	33

4.4	Time representation of the IRL online policy iteration algorithm. . . .	35
4.5	Block diagram of the IRL online policy iteration algorithm considering a full-state observer.	37
4.6	Block diagram of the IRL online policy iteration algorithm considering a full-state observer and disturbance.	41
5.1	Type IV specimens for tensile testing (units in mm).	45
5.2	a) Specimens on edge at different building orientations and b) printed specimens with grid support pattern.	47
5.3	Tensile testing done according to ASTM D638 standard.	49
5.4	Sensors used for MCR UAV V3.0 static characterization.	51
5.5	CRM2409 characterization results.	52
5.6	Reactive torque generated by both rotors individually.	53
5.7	Forces generated for first control surface.	54
5.8	Forces generated for second control surface.	54
6.1	Flow diagram of the numerical study of MCR UAV V3.0.	56
6.2	Control volumes and model simplification of CFD study of MCR UAV V3.0. 1 and 2- blades and 3-bullet shape and control surfaces.	57
6.3	Control volume mesh. a) Frontal and superior view of entire control volumes. b) Inflation layers around deflected control surface.	59
6.4	CFD imported pressure mapped over the control surface.	61
6.5	Block diagram used to export disturbance signals.	68

6.6	Disturbances at different frequencies.	69
7.1	Components of MUAV printed by FDM, a) base supports b) control surfaces.	74
7.2	a) Upper plate mass, b) lower plate mass, a) flight computer plate and joints mass, d) ailerons plate mass.	75
7.3	MUAV Structural assembly progress.	76
7.4	MUAV Avionics assembly progress.	78
7.5	PID Tuning on Librepilot software.	79
7.6	MUAV outdoor flight test.	80
8.1	Stress vs strain plot for specimens with a 10% infill.	82
8.2	Stress vs strain curve of specimens with 15% infill.	83
8.3	Velocity contour around MCR UAV V3.0. a) A-convex shape deflected surface and b) B-optimized convex shape deflected surface. . .	84
8.4	Velocity contour around the UAV, a) scenario C with concave deflected surface and b) scenario D with optimized concave deflected surface.	85
8.5	Finite element results of control surface of scenario A. a) Total deformation (mm) and b) von Mises stress (MPa).	87
8.6	Finite element results of scenario B for an optimized control surface. a) Total deformation (mm) and b) von Mises stress (MPa).	87
8.7	Finite element results of condition C with an optimized control surface. a) Total deformation (mm) and b) von Mises stress (MPa). . . .	88

8.8	Finite element results of condition D with an optimized control surface. a) Total deformation (mm) and b) von Mises stress (MPa). . . .	89
8.9	Numerical results of control surfaces. a) case A and B, b) case C and D.	89
8.10	Least squares variables for no disturbance scenario.	92
8.11	States obtained for no disturbance scenario.	93
8.12	Least squares variables for 75Hz disturbance scenario.	94
8.13	States obtained for 75Hz disturbance scenario.	95
8.14	Least squares variables for 100Hz disturbance scenario.	96
8.15	States obtained for 100Hz disturbance scenario.	96
8.16	Least squares variables for 200Hz disturbance scenario.	97
8.17	States obtained for 200Hz disturbance scenario.	98
8.18	Least squares variables for 100Hz disturbance scenario.	99
8.19	States obtained for 100Hz disturbance scenario.	100
8.20	Least squares variables for 150Hz disturbance scenario.	101
8.21	States obtained for 150Hz disturbance scenario.	102

LIST OF TABLES

2.1	LiPo Calculation results	11
2.2	Avionics used in MCR UAV V3.0.	13
5.1	Printing parameters selected in Creality Ender-3 FDM printer.	46
6.1	Control volume dimensions of MCR UAV V3.0.	58
6.2	CFD Computed forces.	60
6.3	FEA Computed mapped force.	61
8.1	Mechanical properties for specimens with 10% infill density.	82
8.2	Mechanical properties for specimens with 15% infill density.	83
8.3	Stress and deformation results for control surfaces; A, B, C and D.	90
8.4	Output parameters for surface topologies; A, B, C and D.	90
8.5	Obtained controllers for undisturbed simulation.	93
8.6	Obtained controllers for 75Hz disturbance simulation.	95
8.7	Obtained controllers for 100Hz disturbance simulation.	97
8.8	Obtained controllers for 200Hz disturbance simulation.	98

8.9	Obtained controllers for 4 iterations 100Hz disturbance simulation. . .	100
8.10	Obtained controllers for 4 iterations 150Hz disturbance simulation. . .	102

ABSTRACT

Ing. Víctor Hugo Domínguez de la Cruz.

Candidate for the degree of Maestría en Ingeniería en Aeronáutica con orientación en Dinámica de Vuelo.

Universidad Autónoma de Nuevo León.

Facultad de Ingeniería Mecánica y Eléctrica.

Title of the study: DESIGN, MODELING, AND CONTROL OF A MICRO COAXIAL DRONE USING OPTIMAL ADAPTIVE CONTROL BASED ON REINFORCEMENT LEARNING.

Amount of pages: 138.

OBJECTIVES: Design an optimal adaptive control approach based on reinforcement learning scheme that allows calculating and updating the control policy for the dynamics of the MCR UAV V3.0 in the presence of disturbances. Moreover, the optimization of the MCR UAV V3.0 is planned to overcome the lack of structural stability of previously manufactured vehicles, and in consequence increase the flight endurance and reduction of the resources and manufacturing time.

The realization of the main objective implies, building and coding on Matlab software the algorithm, quantify the operating range of disturbances of the proposed control scheme, building on 3D CAD software the MCR V3.0 UAV including the specific avionic elements and movements of the vehicle, manufacture and test of the optimized vehicle using Fused Deposition Modeling (FDM).

CONTRIBUTIONS AND CONCLUSION: The main contributions are the design, modeling, and control of a micro coaxial drone employing an optimal adaptive control scheme based on reinforcement learning algorithm, including the quantification of

the disturbances frequencies allowed by the algorithm. Moreover, the optimization of the MCR UAV V3.0 is done, reducing considerably the net mass of the vehicle and in consequence, the increase of the endurance with respect to previous models. Lastly, the correct 3D design and manufacture of the prototype is done and the vehicle is tested using a conventional PID controller. This is done to assure the correct performance of the new design for further research focused on implementing of new control algorithms.

The mathematical model, computation, and disturbance characterization of an optimal adaptive IRL control scheme were presented in this work. The conventional modeling of the MCR UAV V3.0 in state-space was included adding a full-state observer, including disturbances. Moreover, the IRL algorithm coding was explained and posted in the present work. The FDM manufacture and numerical CFD/FEA procedure of MCR UAV V3.0 with a coaxial counter-rotating configuration were developed in this work. Optimization of two different surface topologies were evaluated: concave deflected surface, and convex deflected surface, including two more optimized surfaces.

Adviser signature: _____
Dr. Octavio Garcia Salazar

CHAPTER 1

INTRODUCTION

Unmanned aerial vehicles (UAVs) have proven to perform complex navigation tasks for humans, as well as surveillance, exploration in difficult access environments, photogrammetry, and aerial shots for various purposes. However, the intrinsic meteorological uncertainties of any external and internal environment are a current constraint for these vehicles, placing these situations as the main causes of which UAVs have incidents. Due to the exponential growth of applications that these vehicles currently have, it is important to have containment systems for phenomena caused by an unpredictable environment. Such systems must have the ability to stabilize the vehicle in a safe position that does not compromise the UAV or its environment, regardless of the initial position of the vehicle when the system is executed or the environment disturbances present at the moment.

Adaptive control schemes based on Reinforcement Learning are an interesting proposal that has generated abundant interest within the scientific community that investigates UAVs since this control technique allows vehicles to deal with unpredictable environmental situations, Lin et al. (2020) developed an adaptive algorithm that allowed the control of a quadrotor, making use of neural networks for the calculation and updating of the control policy, the algorithm was executed only under a situation of risk previously established by the author. Moreover, Zhang Sitong and Qianhui (2022) used as the main premise of his work the fact that most current

publications of path planning use known or controlled environments, Sitong Zhang proposed a control methodology based on deep reinforced learning (DRL) by combining the use of two neural networks to update and calculate control policy, the algorithm is based on Twin Delayed Deep Deterministic Policy Gradients (TD3).

Other advances within the DRL include modeling the environment using the Markov Decision Process (MDP) process. He et al. (2021) proposed the use of a neural network trainable by DRL that allows the navigation of a quadrotor in an unknown environment. The combination of neural networks in conjunction with DRL has become one of the main objects of interest for researchers in the area. Likewise, HU et al. (2021) proposed a control scheme that is based on a Deep Deterministic Policy Gradient (DDPG) in conjunction with a bank of previously recorded experiences. The algorithm has a neural mechanism that allows it to find on similar experience to the current vehicle situation.

The research field of unmanned aerial vehicles (UAVs) has been increasing during the last decade due to the requirements of modern society for UAVs with complex tasks adapted for specific environments. New vehicle configurations are developed to operate the vastly extended field of missions more efficiently. The development of micro unmanned aerial vehicles (MUAV) was an immediate response to these complex missions. Most MUAVs such as MCR UAV V3.0 were designed to accomplish low-speed missions in a specified place; these can be reached by a vehicle with a rotating propulsion system, such as quadrotors, tilt rotors, helicopters, and certain flapping wing vehicles. Unfortunately, since these vehicles operate at low-speed environments with relatively low external forces actuating on the vehicle, most of the implementations are focused on control procedures for the dynamic systems and do not consider any further analysis to assure structural safe or even a more efficient structure in terms of weight and dimensions. Richards et al. (2020) presents a brief description of the required considerations for the design and manufacture of a UAVs using additive manufacturing. The current efforts on MUAV with a hybrid configuration are based on the design strategy and the dynamic characteristics

(Saeed et al., 2018).

Fuse deposition modeling (FDM) is the additive manufacturing technology mostly used to manufacture lightweight structures for aerospace, sport, and medical industries Dikshit et al. (2020). The methodology used by FDM consists of the extrusion of a polymeric material and layer deposition, see Figure 1.1. The printing parameters such as printing orientation and infill percentage, induce variability of the mechanical behavior depending on the printer and the material used (Ravindrababu et al., 2018; Uddin et al., 2017) . Thus, it is important to evaluate the effect of printing orientation and infill percentage on the stiffness and strength of printed components. FDM technology has been utilized to design and manufacture small UAV due to its low cost of printers and materials (Pascariu and Zaharia, 2020)

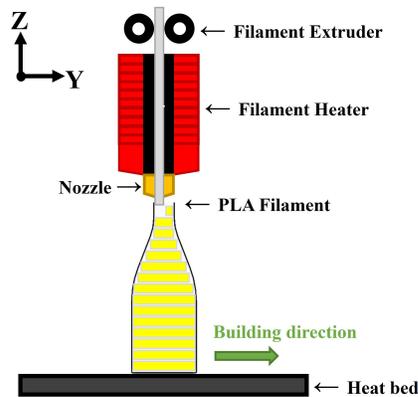


FIGURE 1.1: Schematic of fuse deposition modeling (FDM) technology.

Material extrusion additive manufacturing technique has been widely adopted due to simplicity and low-cost production, the author remarks on the necessity to perform several changes on both hardware and software to fabricate continuous fiber-reinforced polymer CFRP or other non conventional materials, recommending to use of commercial and well-studied thermoplastics filaments such as PLA or ABS (Liu et al., 2021). In the fast manufacture prototypes research, the material most utilized is polylactic acid (PLA) due to its simplicity and printing agility, moreover, it is also a biodegradable thermoplastic polyester material, with a lower amount of hazardous

components and printing temperatures, which saves energy (Kam et al., 2022). A review of articles that characterize the tensile behavior of PLA specimens is presented by Suteja and Soesanti (2020). The main observations are directed to the variation of the layer thickness, building orientation, infill pattern and density. Chacón et al. (2017) presents a tensile and bending analysis of PLA specimens manufactured by FDM technique, considering constant raster angle and printing temperature. It was observed that the mechanical properties decrease as layer thickness decreases, except for the upright printing direction. Yao et al. (2019) presents a methodology to predict the ultimate tensile strength based on Hill-Tsai anisotropic yield criterion. Experiments were conducted using specimens manufactured at different orientation angles and layer thickness. The analytical model agreed with the experimental data.

The variation of temperature during printing operation has been an interesting topic of research, Morteza BehzadnasabAli (2016) evaluated the nozzle printing temperature of PLA specimens manufactured by FDM. The authors performed tensile characterization to evaluate the effect of the temperature. Johansson (2016) developed a study aimed to increase the manufacturing layer bonding of different materials including PLA, the author varies different printing parameter such as extruder temperature, flow rate, layer thickness and printing speed. Tensile testing was performed to find out the most suitable configuration.

The variation of the layer thickness has a considerable effect on the mechanical properties of the printed specimens (Suteja and Soesanti, 2020). Atakok et al. (2022) studied the feasibility of using recycled PLA (Re-PLA). Three different layer thickness and infill percentages were evaluated. The mechanical properties showed an increasing tendency when layer thickness and infill percentage were increased. An evaluation of the ultimate tensile strength and break strain of PLA specimens built by FDM was performed at Antonio Lanzotti (2015). The work was oriented to vary layer thickness, infill, and the number of perimeters. The mechanical properties showed an increasing tendency as the number of perimeters was raised. However, as the layer thickness increase, the tensile strength is reduced. A maximum layer

thickness of 0.15 mm was selected.

Mechanical characterization of five printing materials, including PLA, was developed by Abeykoon et al. (2020). FDM parameters such as infill percentage, infill printing speed, infill patterns and printing temperature were evaluated to find out the best configuration. For PLA, the most suitable speed is reached at 90 mm/s, and the best printing temperature is observed at 115°. Harpool et al. (2021) evaluated the mechanical behavior of PLA specimens manufactured by FDM varying the infill percentage and pattern. It was found that the maximum tensile stress is present on hexagonal specimens (Harpool, 2016).

Recently, diverse studies have been focused on the structural analysis of MUAV with different purposes. For example, Li et al. (2019) developed a static and dynamic analysis using ANSYS to evaluate a maximum payload for fire extinguishing applications. Negrello F. (2016) performed a static and durability analysis to guarantee the number of cycles of a system created for a flapping wing MUAV. Chen et al. (2018) developed a static and modal analysis for a solar-powered high aspect ratio UAV to analyze flutter conditions for a fixed-wing vehicle. Henderson et al. (2017) developed a study for a non-conventional vehicle that deploys six arms, each one with one rotor attached. An experimental procedure was performed to calculate the forces actuating on the arms. A morphological improvement was obtained on the moving elements by finite element analysis, reducing the maximum deflection and increasing the stiffness. Tavares et al. (2017) performed a study to reduce weight and generate an improvement in the reliability of a morphing wing. Zhu et al. (2019) performed a numerical analysis to optimize the aerodynamic characteristics of a deployable UAV considering control capabilities.

A complete design methodology for a fixed-wing UAV was developed by Aktas Y.O. (2016). The study includes fluid-structure analysis, flight dynamics and material selection. The fluid study was used to characterize the aerodynamics of the vehicle at different flight regimes and obtain the pressure gradients. The static

structural analysis considers the weight and rotor thrust to find the critical zones. Panagiotou et al. (2014) presents an aerodynamic optimization study for a UAV winglet. The authors evaluate different winglets configurations and different angles of attack. It was reported that the aerodynamic efficiency is at least 10% higher than the vehicle with no winglets. A methodology to design propellers using high-fidelity analytical tools was developed by Cruzatty et al. (2022). A fluid-structure analysis was performed to evaluate the structure and aerodynamic characteristics. The results showed a maximum error of 9% between the computational and analytical approaches. The aerodynamic optimization of a tactical blended-wing-body UAV was performed by Kapsalis et al. (2021). The authors evaluated several parameters (aspect ratio, taper ratio and sweep angle) to determine the maximum velocity, takeoff runaway and gross take-off weight. De Vivo et al. (2019) designed a Prandtl box-wing UAV based on fluid-structure analysis. The fluid analysis was performed to obtain the highest aerodynamic efficiency of the wing, winglet, and fuselage. The structural analysis includes ribs, spars, skin, wing box and fuselage. The study allowed vehicle weight reduction and material distribution.

In the present work, a numerical design and manufacture of an unmanned aerial vehicle with a coaxial configuration (MCR UAV v3.0) are developed. Fused deposition modeling is utilized to fabricate critical components of the vehicle. Mechanical properties are evaluated at different infill percentages and orientations. A fluid-structure analysis is run mapping pressures acquired from the computational fluid dynamic (CFD) study and used as a load over the control surfaces in the finite element model. A parametric study was performed to evaluate the mechanical response of the structure under different control surface topologies.

1.1 GENERAL OBJECTIVES

The principal objective of the research is to design an optimal adaptive control scheme based on reinforcement learning that allows calculating and updating the

controller for the dynamics of the MCR UAV V3.0 in the presence of disturbances. However, prior to design the control scheme is required to develop the optimization of the MCR UAV V3.0 to overcome the lack of structural stability of the prototype previously manufactured (Dominguez et al., 2022), and in consequence increase the flight endurance and reduction of the resources and manufacturing time.

1.2 SPECIFIC OBJECTIVES

The realization of the main objective implies, building and coding on Matlab software the algorithm, quantify the operating range of disturbances of the proposed control scheme, building on 3D CAD software the MCR UAV V3.0 including the specific avionic elements and movements of the vehicle, and to achieve a correct optimization of the vehicle is required to develop a computational procedure that includes Computational Fluid Dynamics (CFD) and Finite Element Analysis (FEA), material characterization and morphological optimization of control surfaces based on computational simulations.

1.3 SCIENTIFIC CONTRIBUTIONS

The present research resulted out-standing in the UAV research since it intrinsically allowed to produce scientific divulgation into international journal and congress, the first publication remarks the design of the vehicle based on the desired dynamical behavior of it and contemplating simple computational procedure during the development of this publication the first prototype of MCR V3.0 UAV was build (Dominguez et al., 2022), secondly, once the design was done and the vehicle manufactured, a control technique based on a saturated PD controller approach was developed employing computational simulation (Cabriales-Ramírez et al., 2021).

CHAPTER 2

DESIGN OF COAXIAL MUAV

The main characteristic of this aerial vehicle is that it is capable of performing vertical and hover flight as well as horizontal flight. The primary purpose of the MCR UAV V3.0 is to be deployed from a fixed-wing as a shuttle operation, for a subsequent and autonomous hovering flight, once, it arrives at the desired position, it there exists a risk for the human being, it will perform monitoring and observation acquiring, sending the data set to the ground station considering a camera and advanced telemetry, see Fig. 2.1. For the development of this micro UAV, the proposed methodology is considered in the following flowchart, see Figure 2.2.

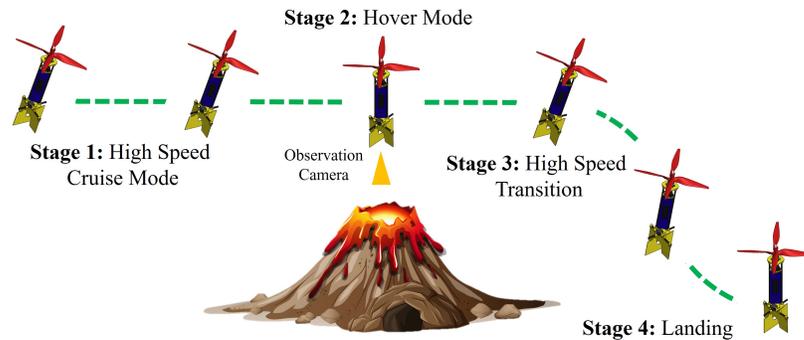


FIGURE 2.1: MCR UAV V3.0 main objective.

The MCR UAV consists of two main contra-rotating rotor which provides the vehicle thrust vector and both are capable of regulating yaw and height motion, a pair of different control surfaces are positioned at the last part of the coaxial vehicle

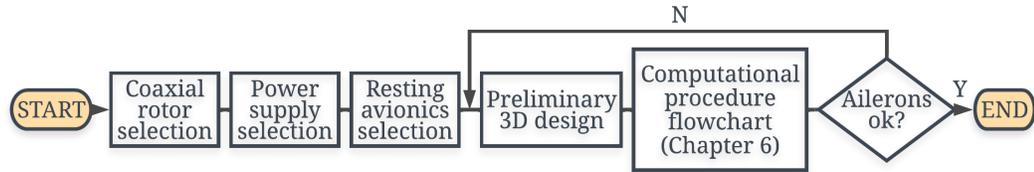


FIGURE 2.2: Flowchart used to MCR UAV V3.0 design.

to deflect the airstream produced by the main rotor and function as a landing gear, these ailerons allow the vehicle to rotate over the center of gravity in X and Y axis. see Figure 2.3.

The coaxial rotor selection is the first component to consider. The rotor selected is the CR23L KV1100 with a propeller length of 0.2032 m and a maximum carry capability (T_{max}) of 1.432 kg. The dimensions of the vehicle are established considering the location of the electronics and the propulsion system.

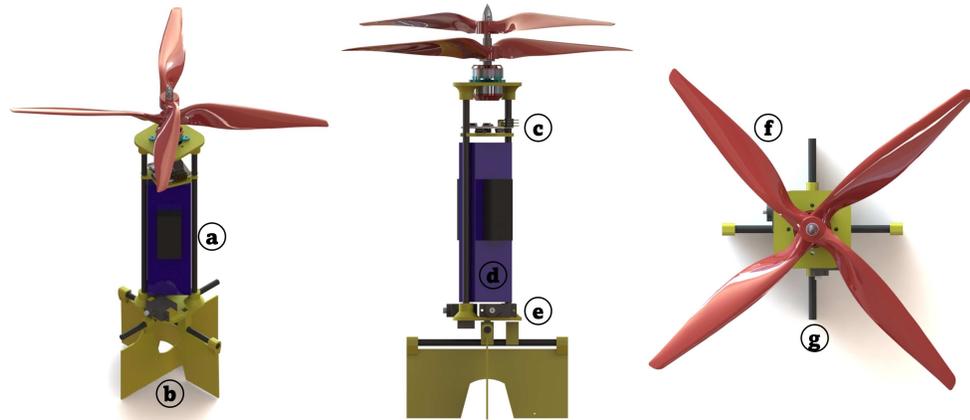


FIGURE 2.3: MCR UAV components. a) 30A ESC b) control surfaces c) flight computer d) LiPo battery e) servomechanism f) coaxial blades g) control surfaces links.

The vehicle must be able to carry out all the avionics and a payload of at least 0.2 kg, defined as the maximum takeoff weight ($MTOW$). The vehicle departure from the ground should reach an $MTOW$ of 50% with the maximum capabilities of the rotor [13]. This will assure adequate battery longevity and additional thrust

available for maneuvers. The net weight of the vehicle is defined by (2.1). This equation relates the weight (W_{UAV}) and the maximum carry capability (T_{max}).

$$W_{UAV} = \frac{T_{max}}{2} - 0.2 \quad (2.1)$$

The resulting weight of the vehicle is 0.516 kg, then the rest of the avionics must be selected to reach this weight. Another critical aspect of the design is the battery used to power the propulsion system and the rest of the avionics. According to the selected coaxial rotor, the power consumption at 50% of the thrust is 361 W (P_{test}) with a voltage of 11.1 V (V_{bat}) using a 3S LiPo battery. Two different batteries (I_{bat}) are analyzed: 2200 mAh, 3000 mAh, and 3300 mAh LiPo battery. The formulation used to calculate the estimated power supplied by the battery is defined by (2.2).

$$P_{bat} = I_{bat} \cdot V_{bat} \quad (2.2)$$

The estimated flight time (FT) for each proposal is calculated according to (2.3), all the results presented in the formulations are described in the Table 2.1.

$$FT = \frac{P_{bat}}{P_{test}} \cdot 60 \quad (2.3)$$

In the present document, four control surfaces topologies are evaluated, these are named as: A-concave deflected shape, B-optimized concave deflected shape, C-convex deflected surface and D- optimized convex deflected surface, illustrated in Figure 2.4.

The pretended environmental conditions for the MCR UAV are those where the arriving/departing speed is crucial for the vehicle integrity and data collection, the vehicle can perform this task since two ailerons with a considerable size (30% of the vehicle height) are used to generate pitch and roll movement. The maximum

operating range of the vehicle is limited to 2 km at sea level air conditions based on the International Standard Atmosphere (ISA). Based on the rotors capabilities, the propellers must have a maximum elongation of 9 in.

TABLE 2.1: LiPo Calculation results

LiPo Battery Selection			
-	LiPo 2200 mAh	LiPo 3000 mAh	LiPo 3300 mAh
Weight (kg)	0.381	0.020	0, 0, 0.025
Power Consumption (W)	0.381	0.020	0, 0, 0.025
Flight Time (min)	3.048	4.080	0, 0, 0.25
Time Increment (%)	0	36.54	50.12

The discussed ailerons are responsible for the fast displacement during the mission. Nevertheless, the link between the body and control surface is a critical element of the vehicle, this can be stated since in Dominguez et al. (2022) the prototype built for control and navigation purposes presented a lack of structural stability in the mentioned part. The first prototype of MCR UAV V3.0 consisted of a pair of control surfaces fabricated from polyurethane foam. However, during flight at an elevated rotational speed of coaxial blades, foam surfaces become unstable generating parasite vibrations, thus, affecting notably the inertial measurement and navigation. In consequence, the FDM technique is considered for both surfaces, reaching a stable aerodynamic behavior. The presented document is focused on addressing this specific problem, that is, to study the printing parameters taking into consideration the strong response of the component.

Based on the propulsion system capabilities, a contra-rotating brushless motor is used to maintain the micro UAV in a vertical pose and the autopilot is selected based on the vehicle configuration. The body of the vehicle from the control surfaces to the tip measures 274.20 mm. The dimensions of the ailerons chord are about 61 and 60 mm, see Figure 2.5,

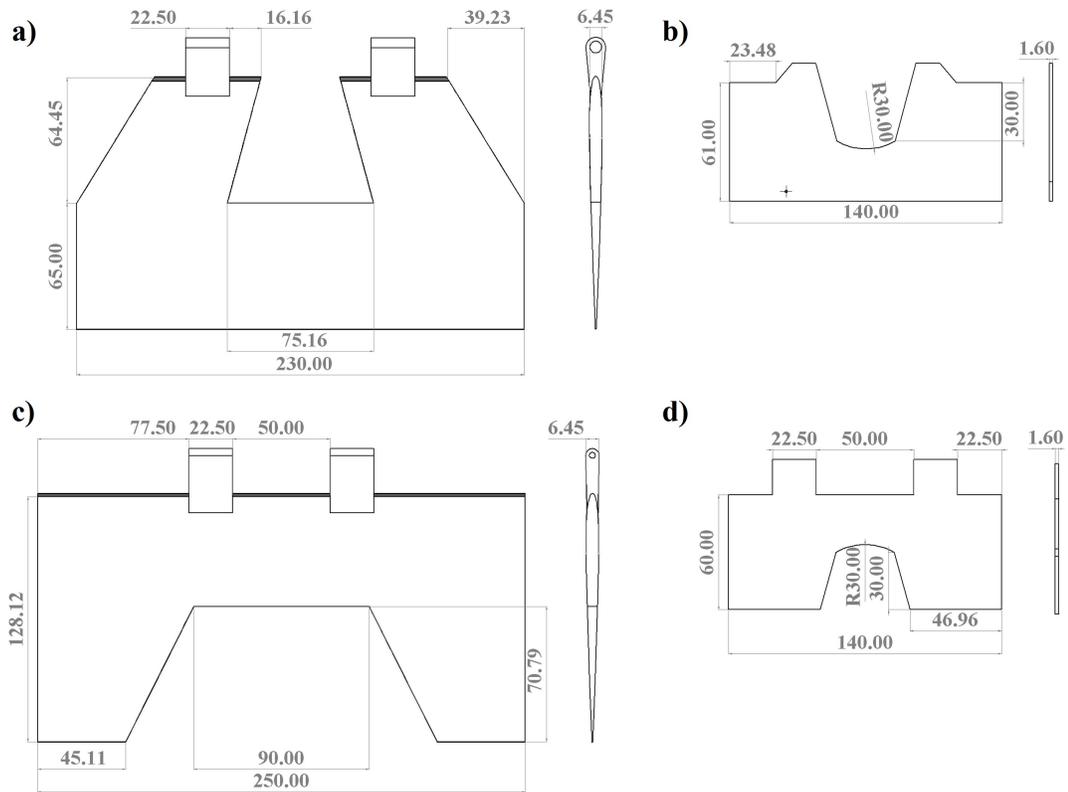


FIGURE 2.4: Control surfaces topologies (units in mm). a) Concave deflected shape, b) optimized concave deflected shape, c) convex deflected surface and d) optimized convex deflected surface.

The autopilot selection is based on the propulsion system selection and the power system: CR23L KV1100 with a propeller 9047 CR/CCR and Zippy 3300mAh LiPo. Then, the rotor requires ESCs (Electronic Speed Controllers) capable of holding constant 30 Amperes, thus one AEO E-Power 30A is used for each rotor, according to the required net weight for the vehicle a conventional flight computer is used: CC3D revolution, it is configured for a coaxial helicopter with a swashplate controlled by two servos. For ailerons two Hitec HS65-MG micro servos are implemented. The battery selected is a Zippy Compact 3300 mAh, all the system is controlled by a Spektrum DX8 Radio controller, see Table 2.2.

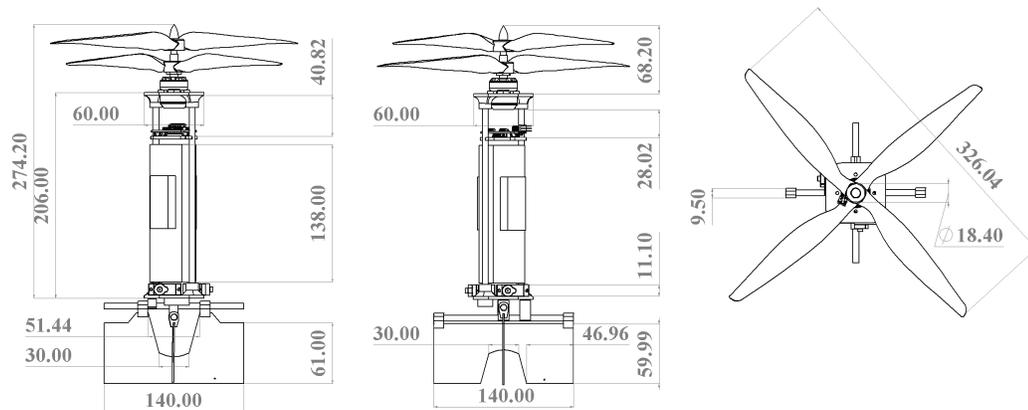


FIGURE 2.5: MCR UAV V3.0 Dimensions.

TABLE 2.2: Avionics used in MCR UAV V3.0.

MCR UAV V3.0 AVIONICS	
Motor	CR23L KV1100
ESC	AEO E-Power 30A
Propeller	CR/CCR 9047
Flight Controller	CC3D Revolution
IMU	MPU 6050
Servo	Hitec HS65-MG
Battery	LiPo 3300 mAh
Radio	Spektrum DX8

CHAPTER 3

MATHEMATICAL MODEL

3.1 KINEMATICS

The model of the MCR UAV V3.0 considers an inertial fixed frame as $\mathcal{I}=\{x_{\mathcal{I}}, y_{\mathcal{I}}, z_{\mathcal{I}}\}$ and a body frame fixed attached to the center of gravity of the aircraft as $\mathcal{B}=\{x_{\mathcal{B}}, y_{\mathcal{B}}, z_{\mathcal{B}}\}$. The wind frame $\mathcal{W}=\{x_{\mathcal{W}}, y_{\mathcal{W}}, z_{\mathcal{W}}\}$ is considered during the cruise of the airplane, Stengel (2022), Stevens et al. (2015), see Fig. 3.1

The equations of motion are obtained by using the Newton-Euler formulation and are given by the following expressions:

$$\dot{\xi} = V \tag{3.1}$$

$$\dot{V} = RF + mge_3 \tag{3.2}$$

$$\dot{R} = R\hat{\Omega} \tag{3.3}$$

$$J\dot{\Omega} = -\Omega \times J\Omega + \Gamma \tag{3.4}$$

where $\xi = (x, y, z)^\top \in \mathbb{R}^3$ and $V = (\dot{x}, \dot{y}, \dot{z})^\top \in \mathbb{R}^3$ are the position coordinates and translational velocity relative to the inertial frame. $m \in \mathbb{R}$ denotes the MCR UAV V3.0 mass, and e_1, e_2 , and e_3 are the vectors of the canonical basis of \mathbb{R}^3 in \mathcal{I} . e_{b_1} ,

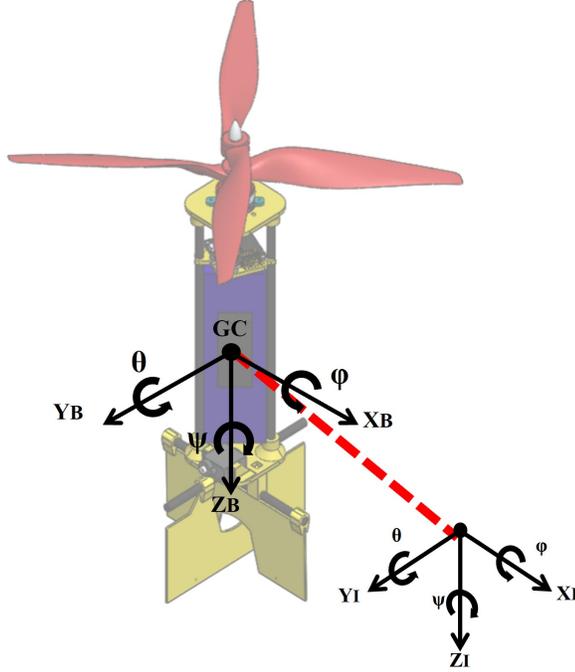


FIGURE 3.1: MCR UAV V3.0.

e_{b_2} , and e_{b_3} are the vectors of the canonical basis of \mathbb{R}^3 in \mathcal{B} . $\eta = (\phi, \theta, \psi)^\top \in \mathbb{R}^3$ describes the rotation coordinates where ϕ , θ and ψ represent the roll, pitch and yaw or heading, respectively. The rotation matrix, $R \in SO(3) : \mathcal{B} \rightarrow \mathcal{I}$, satisfies the $SO(3) = \{R | R \in \mathbb{R}^{3 \times 3}, \det[R] = 1, RR^\top = R^\top R = I\}$ and is parameterized by the Euler angles ϕ , θ and ψ . The rotation matrix R is written as:

$$R = \begin{pmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{pmatrix} \quad (3.5)$$

$\Omega = (p, q, r)^\top \in \mathbb{R}^3$ is the angular velocity in \mathcal{B} and $\hat{\Omega}$ denotes the skew-symmetric matrix of the vector which is given by:

$$\hat{\Omega} = \begin{pmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{pmatrix} \quad (3.6)$$

$J \in \mathbb{R}^{3 \times 3}$ contains the moments of inertia and $\Gamma = (\Gamma_x, \Gamma_y, \Gamma_z)$ are the moments acting on the drone.

3.2 FORCES

The force acting on the MCR UAV V3.0 is defined as propulsion, aerodynamics forces and the weight. The total force results:

$$F = \begin{pmatrix} F_x & F_y & F_z \end{pmatrix}^\top = F_p + F_a + F_w \quad (3.7)$$

Propulsion force of the MCR UAV V3.0 consists of two forces T_1 and T_2 generated by two rotors in coaxial configuration as shown in Figure 3.2

$$F_p = \begin{pmatrix} 0 & 0 & -T_T \end{pmatrix}^\top \quad (3.8)$$

The propulsion thrust force $T_T = T_1 + T_2$, Then, the vehicle thrust, in \mathcal{B} , is given by $F_p = -T_T e_{b_3}$ by assuming that the performance of the upper rotor is not influenced by the lower the rotor, that the rotor planes are sufficiently close together and that each rotor provides an equal fraction of the total system, Leishman (2006).

Aerodynamic forces the MCR UAV V3.0 is designed with two control surfaces which produce lift L_1, L_2 and drag D_1, D_2 forces. It results

$$F_a = W^\top \begin{pmatrix} L_2 \\ L_1 \\ D_1 + D_2 + D_v \end{pmatrix} \quad (3.9)$$

with the rotation matrix $W \in SO(3) : \mathcal{B} \rightarrow \mathcal{A}$ that transforms a force from the body frame to aerodynamic frame is described as

$$W = \begin{pmatrix} c_\alpha c_\beta & s_\beta & s_\alpha c_\beta \\ -c_\alpha s_\beta & c_\beta & -s_\alpha s_\beta \\ -s_\alpha & 0 & c_\alpha \end{pmatrix} \quad (3.10)$$

where α is the angle of attack and β are the sideslip angle, Leishman (2006), Kermode et al. (2012).

Weight force The force due the gravity is given by.

$$F_w = \begin{pmatrix} 0 & 0 & mg \end{pmatrix}^\top \quad (3.11)$$

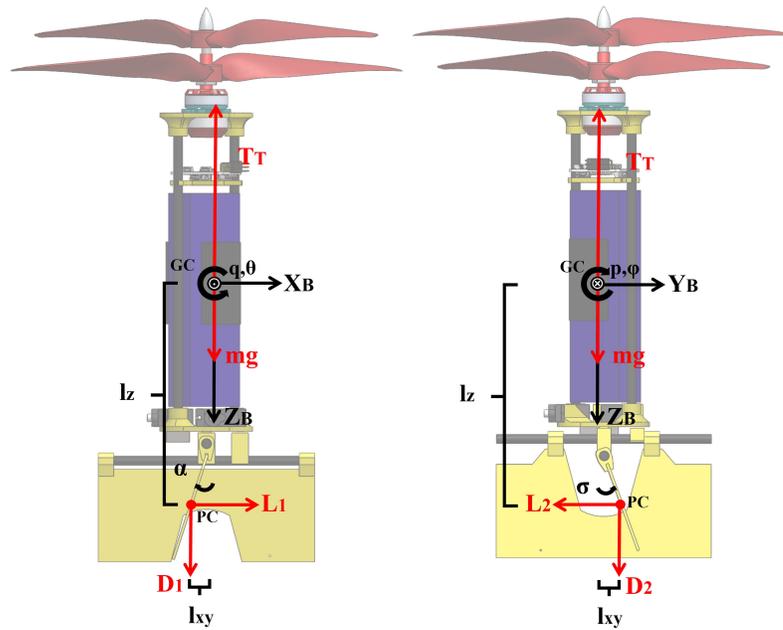


FIGURE 3.2: MCR UAV V3.0 forces.

3.3 MOMENTS

The moments acting on the aerial vehicle are described with propulsion and aerodynamic moments.

$$\Gamma = \begin{pmatrix} \Gamma_x & \Gamma_y & \Gamma_z \end{pmatrix}^\top = \Gamma_r + \Gamma_g + \Gamma_{ac} + \Gamma_{ae} \quad (3.12)$$

The *reaction moment*, due to the rotor shaft, is given by the following mathematical expression:

$$\begin{aligned} \Gamma_r &= \sum_{i=1}^2 Q_i (-1)^{i+1} e_3 \\ &= \begin{pmatrix} 0 \\ 0 \\ Q_T \end{pmatrix} \end{aligned} \quad (3.13)$$

with $Q_T = (Q_1 - Q_2)$ for $Q_i = \rho A_i r_i^3 c_{Q_i} \omega_i^2$, where ρ is the air density, A_i is the rotor disk area, r_i is the rotor radius, c_{Q_i} denotes the rotor shaft moment coefficient and ω_i denotes the angular velocity of the rotor i with $i=1, 2$. The *gyroscopic moment* generated by the rotation of the airframe and the two rotors, it is described by

$$\begin{aligned} \Gamma_g &= \sum_{i=1}^2 I_{r_i} (\Omega \times e_z) (-1)^{i+1} \omega_i \\ &= \begin{pmatrix} \Omega_2 (I_{r_1} \omega_1 - I_{r_2} \omega_2) \\ \Omega_1 (-I_{r_1} \omega_1 + I_{r_2} \omega_2) \\ 0 \end{pmatrix} \end{aligned} \quad (3.14)$$

where I_{r_i} is the moment of inertia of the rotor i and ω_i denotes the angular velocity of the rotor i , with $i=1, 2$.

The *actuator moments* are generated by the control surface forces and are described as

$$\Gamma_{ac} = \begin{pmatrix} L_1 l_z + D_1 l_{xy} & L_1 l_z + D_1 l_{xy} & 0 \end{pmatrix}^\top \quad (3.15)$$

where l_z denotes the distance from the center of gravity to control surfaces through z axis, l_{xy} denotes the distance between the center of gravity and control surfaces

through x or y axes. Note that the vehicle is symmetric, l_{xy} and l_z have the same distance.

Finally, the *aerodynamic moments* presented on the airframe are described as

$$\Gamma_{ae} = \left(\mathcal{L} \quad \mathcal{M} \quad \mathcal{N} \right)^\top \quad (3.16)$$

where \mathcal{L} , \mathcal{M} and \mathcal{N} are the aerodynamic rolling, pitching and yawing moments respectively.

That is, the 4 independent control inputs, $U \in \mathbb{R}^4$ in matrix form, provide the following well-posed actuation mapping

$$U = A_a f \quad (3.17)$$

$$U = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & l_z & 0 & 0 \\ 0 & 0 & l_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} T_T \\ L_1 \\ L_2 \\ Q_T \end{bmatrix} \quad (3.18)$$

being the mapping $A_a \in \mathbb{R}^{4 \times 4}$ invertible.

3.4 EQUATIONS OF MOTION

Taking (3.1)-(3.4), a nonlinear set of equations, based on computed-torque, can be described as

$$\ddot{X} = \frac{F_x}{m} c_\theta c_\psi + \frac{F_y}{m} (s_\phi s_\theta c_\psi - c_\phi s_\psi) + \frac{F_z}{m} (c_\phi s_\theta c_\psi + s_\phi s_\psi) \quad (3.19)$$

$$\ddot{Y} = \frac{F_x}{m} c_\theta s_\psi + \frac{F_y}{m} (s_\phi s_\theta s_\psi + c_\phi c_\psi) + \frac{F_z}{m} (c_\phi s_\theta s_\psi - s_\phi c_\psi) \quad (3.20)$$

$$\ddot{Z} = -\frac{F_x}{m} s_\theta + \frac{F_y}{m} s_\phi c_\theta + \frac{F_z}{m} c_\phi c_\theta - g \quad (3.21)$$

$$\begin{aligned} \ddot{\phi} = & \frac{\dot{\theta}\dot{\psi}}{c_\theta} + \frac{\dot{\theta}\dot{\phi}s_\theta}{c_\theta} + \frac{1}{J_{xx}} [\Gamma_x + qr(J_{yy} - J_{zz})] \\ & + \frac{c_\phi s_\theta}{c_\theta J_{zz}} [\Gamma_z + pq(J_{xx} - J_{yy})] \\ & + \frac{s_\phi s_\theta}{c_\theta J_{yy}} [\Gamma_y + pr(J_{xx} - J_{zz})] \end{aligned} \quad (3.22)$$

$$\begin{aligned} \ddot{\theta} = & -\dot{\phi}\dot{\psi}c_\theta + \frac{c_\phi}{J_{yy}} [\Gamma_y - pr(J_{xx} - J_{zz})] \\ & + \frac{s_\phi}{J_{zz}} [-\Gamma_z - pq(J_{xx} - J_{yy})] \end{aligned} \quad (3.23)$$

$$\begin{aligned} \ddot{\psi} = & \frac{\dot{\theta}\dot{\phi}}{c_\theta} + \frac{\dot{\theta}\dot{\psi}s_\theta}{c_\theta} + \frac{c_\phi}{c_\theta J_{zz}} [\Gamma_z + pq(J_{xx} - J_{yy})] \\ & + \frac{s_\phi}{c_\theta J_{yy}} [\Gamma_y - pr(J_{xx} - J_{zz})] \end{aligned} \quad (3.24)$$

In the aerodynamic analysis, the propeller is considered as a disk, and the flow is incompressible. Then, the velocity is constant and the pressure is uniform over the disk.

The propulsion thrust force $T_c = T_1 + T_2$, for $T_1 = \kappa w_{r_1}^2$ is the thrust of the motor up and $T_2 = \kappa w_{r_2}^2$ for the motor down, is generated by the coaxial rotors, for an aerodynamic constant $\kappa > 0$. Then, the vehicle thrust, in \mathcal{B} , is given by $F_p = T_T e_{b_3}$ by assuming that the performance of the upper rotor is not influenced by the lower rotor, that the rotor planes are sufficiently close together and that each rotor provides an equal fraction of the total system.

Considering an approximation of zero Euler angles $\phi = 0$, $\theta = 0$ and $\psi = 0$ a simplification of the equations of motion could be performed using Jacobian Linearization Chen (1984), it means to consider $c\phi = 1$, $c\theta = 1$, $c\psi = 1$, $s\phi = \phi$, $s\theta = \theta$ and $s\psi = 0$, Moreover, since the mandatory control is to be in hover mode, is congruent to establish that in this flight mode all the rotational speed on the body frame are zero, then $p = 0$, $q = 0$, $r = 0$, $F_x = 0$ and $F_y = 0$ the resulting model can be described as:

$$\ddot{X} = \theta g \quad (3.25)$$

$$\ddot{Y} = -\phi g \quad (3.26)$$

$$\ddot{Z} = \frac{F_z}{m} - g \quad (3.27)$$

$$\ddot{\phi} = \frac{\Gamma_x}{J_{xx}} \quad (3.28)$$

$$\ddot{\theta} = \frac{\Gamma_y}{J_{yy}} \quad (3.29)$$

$$\ddot{\psi} = \frac{\Gamma_x}{J_{zz}} \quad (3.30)$$

Based on the previous equations is possible to write state differential equations considering a jacobian linearization, since the purpose of this document is only to control lateral and longitudinal dynamics only (3.25), (3.26), (3.28) and (3.29) are employed resulting in the following set of differential equations, considering $x_1 = X$, $x_2 = \dot{X}$, $x_3 = Y$, $x_4 = \dot{Y}$, $x_5 = \phi$, $x_6 = \dot{\phi}$, $x_7 = \theta$, $x_8 = \dot{\theta}$, then in order to create a linear system is possible to write all in first order equations, resulting in:

$$\dot{x}_1 = \dot{X} \quad (3.31)$$

$$\dot{x}_2 = \theta g \quad (3.32)$$

$$\dot{x}_3 = \dot{Y} \quad (3.33)$$

$$\dot{x}_4 = -\phi g \quad (3.34)$$

$$\dot{x}_5 = \dot{\phi} \quad (3.35)$$

$$\dot{x}_6 = \frac{\Gamma_x}{J_{xx}} \quad (3.36)$$

$$\dot{x}_7 = \dot{\theta} \quad (3.37)$$

$$\dot{x}_8 = \frac{\Gamma_y}{J_{yy}} \quad (3.38)$$

This specific form of representation allows to write all the equations of motion using state-space form, of the linear time invariant (LTI) system, to restate the previous equations is necessary to use:

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (3.39)$$

where $x(t) \in \mathbb{R}^n$ is the state vector space with a column form of n state variables, $u(t) \in \mathbb{R}^m$ is the vector with column form with m input variables called the input vector, $A \in \mathbb{R}^{n \times n}$ is the state matrix with $(n \times n)$ dimensions, $B \in \mathbb{R}^{n \times m}$ is the input matrix with $(n \times m)$ dimensions. Since only lateral and longitudinal dynamics are evaluated in this document, $n = 4$ and $m = 1$ for each decoupled dynamics.

Since for a number of systems some of the state variables may be inaccessible or their values. Thus a second equation is necessary to determine the system output signals. The output equation is written in the general form

$$y(t) = Cx(t) + Du(t) \quad (3.40)$$

where $y(t)$ is the vector with column form of r output variables called the output vector, C is the output matrix with $(r \times n)$ dimensions and D is the direct transition matrix with $(r \times m)$ dimensions,

Since the scope of this document is to control the longitudinal and lateral dynamics of the system the linearized equations can be splitted in to two different state-space systems, Restating (3.31) - (3.38) using state space form described at (3.39) and (3.40)

For lateral dynamics:

$$\begin{bmatrix} \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -g & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{J_{xx}} \end{bmatrix} \Gamma_x \quad (3.41)$$

For longitudinal dynamics:

$$\begin{vmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_7 \\ \dot{x}_8 \end{vmatrix} = \begin{vmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & g & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{vmatrix} \begin{vmatrix} x_1 \\ x_2 \\ x_7 \\ x_8 \end{vmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{J_{yy}} \end{bmatrix} \Gamma_y \quad (3.42)$$

INTEGRAL REINFORCEMENT LEARNING (IRL) APPROACH

Applications of Reinforcement Learning (RL) in feedback control for continuous-time (CT) dynamical systems $\dot{x} = f(x) + g(x)u$ has a lack. Since Bellman equation for discrete-time (DT) systems $V(x_k) = r(x_k, u_k) + \gamma V(X_{K+1})$ does not depend on the system dynamics, however, for CT systems the Bellman equation actually depends on the dynamics.

The control of the MCR UAV V3.0 is based on an Integral Reinforcement Learning (IRL) algorithm to online solve the linear quadratic regulator (LQR) problem. This algorithm is based on policy iterations that provide a solution procedure executed online for the optimal control problem CT, linear time-invariant systems based on RL.

One of the most important characteristics of IRL algorithm is that partially model-free of this, which means that it does not require full knowledge of the system dynamics. The dynamics of system matrix A is not required, however, the input-coupling matrix B must be well known.

4.1 CONTINUOUS-TIME ADAPTIVE CRITIC SOLUTION FOR THE LINEAR QUADRATIC REGULATOR

This section develops the IRL algorithm to online solve LQR problem without using knowledge of the MCR UAV 3.0 system (matrix A). This procedure solves the algebraic Riccati equation (ARE) online in real-time by measuring data from states $x(t)$ and control scheme input $u(t)$ along the system trajectories.

Considering the CT linear time-invariant dynamical system described by:

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (4.1)$$

with state $x(t) \in \mathbb{R}^n$ control input $u(t) \in \mathbb{R}^m$, and state input matrix A B stabilizable. To this system can be written an associate infinite-horizon quadratic cost function

$$V(x(t_0), t_0) = \int_{t_0}^{\infty} (x^T(\tau)Qx(\tau) + u^T(\tau)Ru(\tau)) (d\tau) \quad (4.2)$$

where $Q > 0$ and $R > 0$ are positive matrices, also, $(Q^{1/2}, A)$ is detectable. The LQR optimal control problem requires finding the control policy that minimizes the cost, this means

$$u(t) = \underset{\substack{u(t) \\ t_0 \leq t \leq \infty}}{\operatorname{argmin}} V(t_0, x(t_0), u(t)) \quad (4.3)$$

The solution to this optimal control problem can be determined by Bellman's optimality principle, which is given by the state feedback law $u(t) = -Kx(t)$ given by

$$K = R^{-1}B^T P \quad (4.4)$$

where the matrix P is the unique positive definite solution of the ARE given by

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \quad (4.5)$$

Accomplishing the detectability condition for $(Q^{1/2}, A)$ the unique positive semi-definite solution of the ARE determines a stable closed-loop controller given by (4.4).

It is remarkable to mention that to solve (4.5), it is required complete knowledge of the model of the system, it implies states matrix A and the control input matrix B . Thus, before the implementation is required a system identification, however, this procedure most often ends with a proximate of the real dynamics. Indeed, IRL algorithm does not require the prior system identification procedure, this particular property does an interesting algorithm from the control systems point of view.

4.2 INTEGRAL REINFORCEMENT LEARNING

This section presents IRL algorithm that solves online for the optimal control gain (4.4) without having knowledge of the system state A from (4.1), it implies solving (4.5). The algorithm result is an adaptive controller that converges to the state-feedback optimal controller. The algorithm is based on an actor-critic structure and consists of a two-step procedure: see Figure 4.1

- critic update stage: results in calculating the infinite-horizon cost associated with a given stabilizing controller.
- actor update stage: this represents the feedback gain matrix K update with the purpose of reducing the cost compared to the present controller

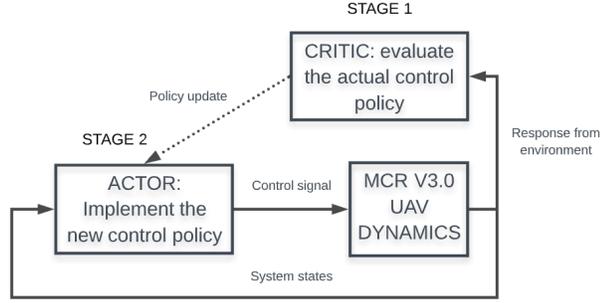


FIGURE 4.1: IRL stages.

Knowing that K is a stabilizing gain for linear CT MCR UAV V3.0 dynamics (3.41) - (3.42) such that $\dot{x} = (A - BK)x$ is closed loop stable. Then the corresponding infinite-horizon quadratic cost equation can be described by:

$$V(x(t)) = \int_{t_0}^{\infty} x^T(\tau)(Q + K^T RK)x(\tau)(d\tau) = x^T(t)Px(t) \quad (4.6)$$

where P is a real symmetric and positive matrix, this represents the solution of the Lyapunov equation.

$$(A - BK)^T P + P(A - BK) = -(K^T RK + Q) \quad (4.7)$$

Now $V(x(t))$ can be used as a Lyapunov function for (4.1) with a gain K . Then infinite-horizon quadratic cost equation can be written as:

$$V(x(t)) = \int_t^{t+T} x^T(\tau)(Q + K^T RK)x(\tau)(d\tau) + V(x(t+T)) \quad (4.8)$$

This new formulation is called CT Bellman equation for LQR problem. Based on (4.6), is possible to modify the equation for convenience of the solution to:

$$x_t^T P_i x_t = \int_t^{t+T} x_t^T (Q + K^T RK)x(\tau)(d\tau) + x_{t+T}^T P_i x_{t+T} \quad (4.9)$$

Moreover, since infinite-horizon quadratic cost equation is modified, the control policy is modified:

$$K_{i+1} = R^{-1}B^T P_i \tag{4.10}$$

Equations (4.9) and(4.10) formulate a IRL algorithm see Figure 4.2, this is a policy iteration procedure for CT systems. It is remarkable to mention that an initial stabilizing control gain K_1 is required. Note that the implementation of this algorithm does not require the states matrix A .

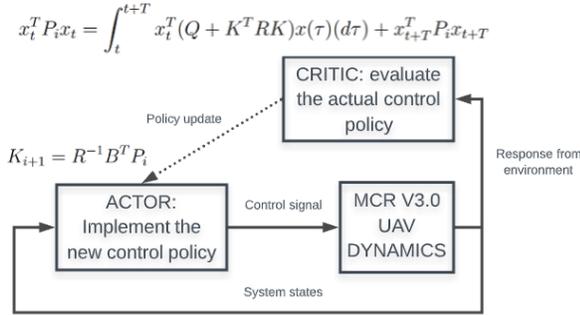


FIGURE 4.2: IRL stages considering formulations.

4.3 ONLINE IMPLEMENTATION OF IRL ADAPTIVE OPTIMAL CONTROL FOR MCR UAV V3.0

This section is focused on the implementation of IRL adaptive optimal control for MCR UAV V3.0, this can be done using the discussed two stages procedure; critic and actor update, where formulations (4.9) and(4.10) are used respectively. The presented algorithm is capable of developing the IRL iterations in real-time only by measuring the current states $x(t)$, future states $x(t+T)$, and measuring or computing the control signal $u(t)$. For this procedure, there is only required knowledge of the input B matrix because it explicitly appears in the policy update formulation (4.10)

Based on (4.6) is possible to claim:

$$V_i(x(t)) = x^T(t)P_i x(t) \quad (4.11)$$

where i refers for each iteration of each IRL algorithm, it is possible to claim that the parameters of the value function $V_i(x(t))$ include the independent values of the symmetric P matrix at the same iteration. The value function can be calculated by measuring the data of the current states $x(t)$, future states $x(t + T)$, and control signal $u(t)$. To compute these parameters, the term $x^T(t)P_i x(t)$ is written as:

$$x^T(t)P_i x(t) = \bar{p}^T \bar{x}(t) \quad (4.12)$$

where $\bar{x}(t)$ is the Kronecker product quadratic polynomial basis vector, where the MCR UAV V3.0 states are inherently included. The vector \bar{p}^T contains only the independent elements of the matrix P ordered by columns, this means that the elements below the diagonal are removed, also, the off-diagonal elements are considered as $2P_{ij}$ considering ij as row-column respectively of the P matrix.

Considering MCR UAV V3.0 lateral dynamics described at (3.41) is possible to restate the mathematical expression (4.12) in the following form:

$$\begin{array}{c} \left| \begin{array}{cccc} x_1 & x_2 & x_7 & x_8 \end{array} \right| \end{array} \begin{array}{c} \left[\begin{array}{cccc} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \\ P_{41} & P_{42} & P_{43} & P_{44} \end{array} \right] \begin{array}{c} \left[\begin{array}{c} x_1 \\ x_2 \\ x_7 \\ x_8 \end{array} \right] \end{array} = \begin{array}{c} \left[\begin{array}{c} P_{11} \\ 2P_{12} \\ 2P_{13} \\ 2P_{14} \\ P_{22} \\ 2P_{23} \\ 2P_{24} \\ P_{33} \\ 2P_{34} \\ P_{44} \end{array} \right]^T \begin{array}{c} \left[\begin{array}{c} \dot{x}_1^2 \\ x_1x_2 \\ x_1x_7 \\ x_1x_8 \\ \dot{x}_2^2 \\ x_2x_7 \\ x_2x_8 \\ \dot{x}_7^2 \\ x_7x_8 \\ \dot{x}_8^2 \end{array} \right] \end{array} \end{array} \quad (4.13)$$

Combining (4.12) into first stage algorithm formulation: CT Bellman equation for LQR problem (4.9), the next formulation can be derived:

$$\bar{p}_i^T(\bar{x}(t) - \bar{x}(t+T)) = \int_t^{t+T} x_t^T(Q + K^T RK)x(\tau)(d\tau) \quad (4.14)$$

Now the only unknown variable is oriented to \bar{p}_i , remembering that this vector contains the unique values of kernel P matrix necessary to solve the second stage algorithm formulation: Policy update (4.9), also $(\bar{x}(t) - \bar{x}(t+T))$ acts as a regression vector, knowing this:

$$d(\bar{x}(t), K_i) = \int_t^{t+T} x_t^T(Q + K^T RK)x(\tau)(d\tau) \quad (4.15)$$

From the last derivation is possible to claim that the correct values of the right side equation give in essence a mathematical expression that contains the unique values of kernel P matrix, which means, it contains the \bar{p} vector. To compute, it efficiently, define a controller state $V(t)$ and add the state equation to the controller scheme.

$$\dot{V} = x^T(t)Qx(t) + u^T(t)Ru(t) \quad (4.16)$$

The necessary regression vector needed is explicitly described by:

$$d(\bar{x}(t), K_i) = V(t + T) + V(t) \quad (4.17)$$

where $V(t)$ is the simple integration of the new state established at (4.16) there is only one unknown parameter vector \bar{p}_i , however in order to solve there is necessary to use (4.17) formulation since this is expressed as a regression vector. As consequence is possible to define a standard form in adaptive control to solve this unknown; using methods such as recursive least-squares (RLS) or a more complex method such as gradient descent. Then, a persistence of the excitation condition is required. The presented documentation it is used RLS algorithm.

4.3.1 RECURSIVE LEAST SQUARES

RLS technique is used to solve (4.17), this can only be employed after a sufficient number of state-trajectory points which are collected using the same control policy K generated using (4.10) a least squares (LS) method can be employed to solve for the parameters of \bar{p}_i , this vector contains the independent values of kernel matrix P .

The vector \bar{p}_i can be found by minimizing in the least-squares sense, the error between the target function, $d(\bar{x}(t), K_i)$ described at (4.17) and the parameterized left side of (4.14). Considering MCR UAV V3.0 dynamics, P matrix has 10 independent elements.

Therefore, the left side of (4.14) can be computed using batch least squares, however, in order to solve correctly it is necessary to execute LS procedure with at least $N \geq n(n + 1)/2$ number of iterations N where n is the number of the states

of state-space system over time intervals T . A flowchart of this online adaptive IRL algorithm is presented in Figure 4.3. Then, the LS solution is obtained as:

$$\bar{p}_i = (XX^T)^{-1}XY \quad (4.18)$$

where:

$$X = [\bar{x}_\Delta^1 \quad \bar{x}_\Delta^2 \quad \bar{x}_\Delta^3 \dots \quad \bar{x}_\Delta^N] \quad (4.19)$$

$$\bar{x}_\Delta^i = \bar{x}^i(t) - \bar{x}^i(t+T) \quad (4.20)$$

$$Y = [d(\bar{x}^i, K_i) \quad d(\bar{x}^2, K_i) \dots \quad d(\bar{x}^N, K_i)] \quad (4.21)$$

The LS problem can be solved after there are N data points collected along all state trajectories, under the presence of an excitation requirement, the presence of excitation is an intrinsic characteristic of adaptable control algorithms.

In order to solve LS algorithm it is necessary to create an array of data for equation (4.19) and other for (4.21) based on the MCR UAV V3.0 dynamics described at (3.41) and (3.42), for instance since a symmetric vehicle has been considered, longitudinal and lateral dynamics are mathematically equal. Thus, as in (4.13) for convenience of the documentation only lateral dynamics are mathematically reported.

Firstly, at (4.19) is in essence an array created with several N elements where each element corresponds to a difference of the current Kronecker product vector $\bar{x}^i(t)$ and the future vector in time $\bar{x}^i(t+T)$ where T is the limit time to calculate a new control policy and update for convenience of the algorithm it is established as 0.5 seconds, all this can be described at (4.20).

Based on (4.13) is possible to define the Kronecker product vector as:

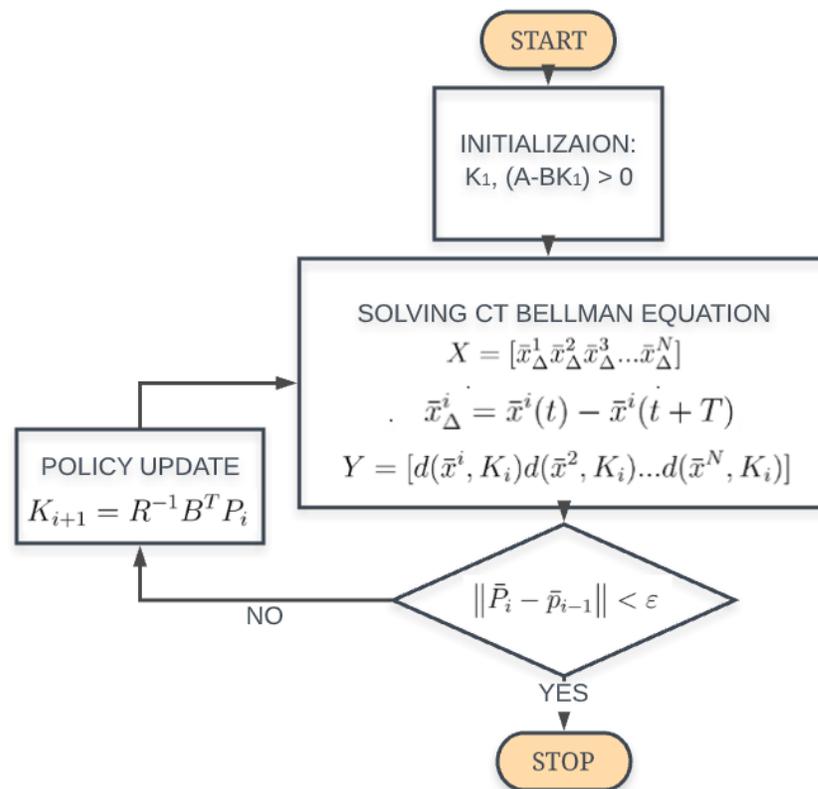


FIGURE 4.3: Flowchart for online IRL algorithm for CT linear systems.

$$\bar{x} = \begin{bmatrix} x_1^2 & x_1x_2 & x_1x_7 & x_1x_8 & x_2^2 & x_2x_7 & x_2x_8 & x_7^2 & x_7x_8 & x_8^2 \end{bmatrix} \quad (4.22)$$

Thus, substituting (4.22) on (4.20)

$$\bar{x}_\Delta = \begin{bmatrix} x_1^2(t) - x_1^2(t+T) \\ x_1x_2(t) - x_1x_2(t+T) \\ x_1x_7(t) - x_1x_7(t+T) \\ x_1x_8(t) - x_1x_8(t+T) \\ x_2^2(t) - x_2^2(t+T) \\ x_2x_7(t) - x_2x_7(t+T) \\ x_2x_8(t) - x_2x_8(t+T) \\ x_7^2(t) - x_7^2(t+T) \\ x_7x_8(t) - x_7x_8(t+T) \\ x_8^2(t) - x_8^2(t+T) \end{bmatrix} \quad (4.23)$$

This is the actual mathematical representation of (4.20) for MCR UAV V3.0 dynamics in order to solve LS algorithm and solve the policy iteration procedure. However, this vector calculation is modified every iteration i of the simulation, since $x(t)$ represents the current state of the iteration i evaluated. It means that, these values are changing during time, unlike $x(t+T)$ that state values are completely static since T is defined as 0.5 seconds, moreover, time-step of the simulation is established as 0.005 seconds, then, doing basic mathematics is possible to calculate that 100 time-step are required to reach 0.5 simulation time. Thus $N = 100$. See Figure 4.4

Based on the last time representation of the IRL online policy iteration algorithm is possible to restate (4.23) vector in the following form:

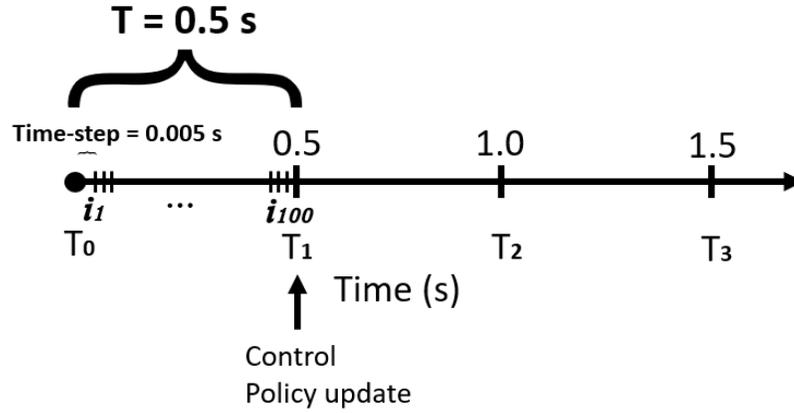


FIGURE 4.4: Time representation of the IRL online policy iteration algorithm.

$$\bar{x}_{\Delta}^{i=1} = \begin{bmatrix} x_1^2(0) - x_1^2(0.5) \\ x_1x_2(0) - x_1x_2(0.5) \\ x_1x_7(0) - x_1x_7(0.5) \\ x_1x_8(0) - x_1x_8(0.5) \\ x_2^2(0) - x_2^2(0.5) \\ x_2x_7(0) - x_2x_7(0.5) \\ x_2x_8(0) - x_2x_8(0.5) \\ x_7^2(0) - x_7^2(0.5) \\ x_7x_8(0) - x_7x_8(0.5) \\ x_8^2(0) - x_8^2(0.5) \end{bmatrix} \quad (4.24)$$

It is essential to clarify that the presented vector only corresponds to the calculation of the first iteration $i = 1$. To correctly represent (4.19) formulation and solve LS algorithm it is necessary to calculate the vector (4.24) for each iteration i , thus is necessary to calculate the vector N times. Then:

$$X = \begin{bmatrix} \begin{bmatrix} x_1^2(t) - x_1^2(t+T) \\ x_1x_2(t) - x_1x_2(t+T) \\ x_1x_7(t) - x_1x_7(t+T) \\ x_1x_8(t) - x_1x_8(t+T) \\ x_2^2(t) - x_2^2(t+T) \\ x_2x_7(t) - x_2x_7(t+T) \\ x_2x_8(t) - x_2x_8(t+T) \\ x_7^2(t) - x_7^2(t+T) \\ x_7x_8(t) - x_7x_8(t+T) \\ x_8^2(t) - x_8^2(t+T) \end{bmatrix} & \begin{bmatrix} x_1^2(t+t) - x_1^2(t+T) \\ x_1x_2(t+t) - x_1x_2(t+T) \\ x_1x_7(t+t) - x_1x_7(t+T) \\ x_1x_8(t+t) - x_1x_8(t+T) \\ x_2^2(t+t) - x_2^2(t+T) \\ x_2x_7(t+t) - x_2x_7(t+T) \\ x_2x_8(t+t) - x_2x_8(t+T) \\ x_7^2(t+t) - x_7^2(t+T) \\ x_7x_8(t+t) - x_7x_8(t+T) \\ x_8^2(t+t) - x_8^2(t+T) \end{bmatrix} & \dots & \bar{x}_{\Delta}^{i=N} \end{bmatrix} \quad (4.25)$$

Since the time step for the simulation is declared as 0.005 seconds and the policy update interval T as 0.5 seconds there are 100 iterations necessary to reach T in time; as consequence, (4.25) formulation is a matrix with $(10 \times N)$ columns. This vector is declared as the independent variable to execute LS algorithm.

Secondly, the dependent variable is missing to calculate, this is represented in (4.21), this formulation can be rewritten using (4.17), since the Kroneleker quadratic product is known that this row vector has $1 \times N$ dimensions.

$$Y = [V(t+T) + V(t) \quad V(t+T) + V(t+t) \dots \quad d(\bar{x}^N(t), K_i)] \quad (4.26)$$

4.4 OBSERVER FOR NON FULL-STATE FEEDBACK

In order to think about a real implementation, it is possible to include a simple flight controller to the MCR UAV V3.0 since there is not enough margin of weight to include avionics. Often, this simple type of flight controller only includes an Inertial

if there is any error of estimation at the output, it asymptotically tends to zero.

From the system representation in Figure 4.5 it is possible to illustrate the equation of the observer states \hat{x} .

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L(Cx(t) - C\hat{x}(t)) \quad (4.29)$$

It is possible to rewritten the previous formulation by making a combination of the output estimation error $\tilde{y}(t)$ equations described at (4.27) - (4.28)

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L(y(t) - C\hat{x}(t)) \quad (4.30)$$

$$\dot{\hat{x}}(t) = (A - LC)\hat{x}(t) + Bu\hat{x}(t) + Ly(t) \quad (4.31)$$

Moreover, it is possible to feedback the observer states $\dot{\hat{x}}$ on the complete system illustrated in Figure 4.5 based on the analysis of the derivative of the states estimation error $e(t)$

$$\dot{e}(t) = \dot{x}(t) - \dot{\hat{x}}(t) \quad (4.32)$$

The previous formulation can be modified using the state representation in (3.39) and (4.31)

$$\dot{e}(t) = Ax + Bu - (A - LC)\hat{x}(t) - Bu\hat{x}(t) - Ly(t) \quad (4.33)$$

$$\dot{e}(t) = (A - LC)x(t) - (A - LC)\hat{x}(t) \quad (4.34)$$

$$\dot{e}(t) = (A - LC)(x(t) - \hat{x}(t)) \quad (4.35)$$

$$\dot{e}(t) = (A - LC)e(t) \quad (4.36)$$

The formulation (4.36) states that if all the eigenvalues of the $(A - CL)$ product are stable; thus, the estimation error $\dot{e}(t)$ is asymptotically tending to zero, in

this form the measured states $\hat{x}(t)$ can be approximated to the real states $x(t)$; as consequence, the control law can be modified to $u = -K\hat{x}(t)$

Considering the rewritten control law to $u = -K\hat{x}(t)$ it is possible to build a complete model using the main state-space plant and the observer system illustrated at Figure 4.5. This new system describes the entire control procedure:

$$\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} A & -BK \\ LC & A - LC - BK \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} + \begin{bmatrix} B \\ B \end{bmatrix} r \quad (4.37)$$

$$y = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} \quad (4.38)$$

Considering the following transformation of the equivalence:

$$\begin{bmatrix} x \\ x - \hat{x} \end{bmatrix} = \begin{bmatrix} I & 0 \\ I & -I \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} \quad (4.39)$$

The transformation allows to rewritten the entire system as the following form:

$$\begin{bmatrix} \dot{x} \\ \dot{e} \end{bmatrix} = \begin{bmatrix} A - BK & BK \\ 0 & A - LC \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} r \quad (4.40)$$

$$[y] = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix} \quad (4.41)$$

This last formulation is used to obtain the response of the system considering an observer for the entire system. It is remarkable to mention that the response of this system involves the measures states derivatives \dot{x} and the derivative of the states estimation error $\dot{\hat{x}}(t)$.

For lateral dynamics described in (3.41) the previous formulation can be rewritten as:

Output perturbations are produced by uncertainties in the environment. For flying vehicles, a wind gust can be a suitable example, or any human impulse intentionally executed to measure the robustness of the control scheme.

In the present computational procedure, it is preferred to analyze an output perturbation since these are considered as most probable to appear in a real-world implementation; thus, the state-space mathematical model must be modified:

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{4.44}$$

$$y(t) = Cx(t) + CEp(t) \tag{4.45}$$

The proposed mathematical model can be obtained using the block diagram represented in Figure 4.6 it is possible to visualize that a perturbation $d(t)$ is implemented and a deputation matrix (E) is also employed to decide the states that are perturbed.

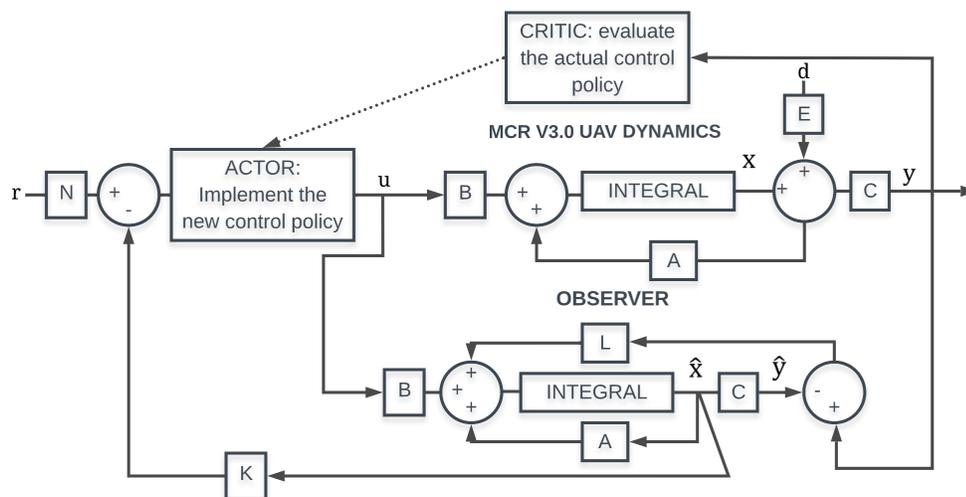


FIGURE 4.6: Block diagram of the IRL online policy iteration algorithm considering a full-state observer and disturbance.

Considering the proposed control algorithm is required to include the control input to the stabilization of the vehicle. The control input obtained is (4.46):

$$u = Nr(t) - K(x(t) + Ep(t)) \quad (4.46)$$

Substituting the presented control law described in (4.46) in to the mathematical model state equation (4.44) the following expression can be obtained and simplified:

$$\dot{x}(t) = Ax(t) + B[Nr(t) - K(x(t) + Ep(t))] \quad (4.47)$$

$$\dot{x}(t) = Ax(t) + BNr(t) - BKx(t) - BKEp(t) \quad (4.48)$$

$$\dot{x}(t) = (A - BK)x(t) + BNr(t) - BKEp(t) \quad (4.49)$$

Based on the expression before it is possible to obtain the A and B matrices for the simulation, corresponding to the new mathematical model can be described:

$$[\dot{x}] = [A - BK][x] + [BN - BKE] \begin{bmatrix} r \\ p \end{bmatrix} \quad (4.50)$$

According to the control scheme presented in Figure 4.6 it is possible to obtain the equation of state for the observer $\hat{x}(t)$:

$$\dot{\hat{x}} = A\hat{x}(t) + Bu(t) + L(Cx(t) + CEp(t) - C\hat{x}(t)) \quad (4.51)$$

$$\dot{\hat{x}} = A\hat{x}(t) + Bu(t) + LCx(t) + LCEp(t) - LC\hat{x}(t) \quad (4.52)$$

$$\dot{\hat{x}} = (A - LC)\hat{x}(t) + Bu(t) + LCx(t) + LCEp(t) \quad (4.53)$$

where in the presented system is possible to calculate the error between both state signals; states of the controller $x(t)$ and observer states $\hat{x}(t)$, this can be:

$$e(t) = x(t) - \hat{x}(t) \quad (4.54)$$

Based on the previous error equation, and substituting (4.53) and (4.45) it is possible to obtain the following expression based on the error of the states $e(t)$.

$$\dot{e}(t) = Ax(t) + Bu(t) - [(A - LC)\hat{x}(t) + Bu(t) + LCx(t) + LCEp(t)] \quad (4.55)$$

$$\dot{e}(t) = Ax(t) + Bu(t) - (A - LC)\hat{x}(t) - Bu(t) - LCx(t) - LCEp(t) \quad (4.56)$$

$$\dot{e}(t) = (A - LC)x(t) - (A - LC)\hat{x}(t) - LCEp(t) \quad (4.57)$$

$$\dot{e}(t) = (A - LC)e(t) - LCEp(t) \quad (4.58)$$

This last formulation allows building the new mathematical model that includes the perturbation elements combined with the principal dynamics of the vehicle,

$$\begin{bmatrix} \dot{x} \\ \dot{e} \end{bmatrix} = \begin{bmatrix} A - BK & BK \\ 0 & A - LC \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix} + \begin{bmatrix} BN & -NKE \\ 0 & -LCE \end{bmatrix} \begin{bmatrix} r \\ p \end{bmatrix} \quad (4.59)$$

$$\begin{bmatrix} y \end{bmatrix} = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix} + \begin{bmatrix} 0 & CE \end{bmatrix} \begin{bmatrix} r \\ p \end{bmatrix} \quad (4.60)$$

For lateral dynamics described in (3.41), the previous formulation can be rewritten as:

$$\begin{aligned}
\begin{bmatrix} \dot{x}_1 \\ \ddot{x}_2 \\ \dot{x}_7 \\ \ddot{x}_8 \\ \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{x}_7 \\ \tilde{x}_8 \end{bmatrix} &= \begin{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & g & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix} \end{bmatrix} - \begin{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{J_{yy}} \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix} \end{bmatrix} \begin{bmatrix} \begin{bmatrix} K_{11} \\ K_{12} \\ K_{13} \\ K_{14} \end{bmatrix}^T \\ \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & g & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix} \end{bmatrix}^T + \begin{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{J_{yy}} \end{bmatrix} \\ \begin{bmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \\ L_{31} & L_{32} \\ L_{41} & L_{42} \end{bmatrix} \\ \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}^T \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_7 \\ x_8 \\ \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{x}_7 \\ \tilde{x}_8 \end{bmatrix} + \\
\begin{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{J_{yy}} \end{bmatrix} \\ \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \end{bmatrix} \begin{bmatrix} \begin{bmatrix} N_{11} \\ N_{12} \\ N_{13} \\ N_{14} \end{bmatrix}^T \\ \begin{bmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \\ L_{31} & L_{32} \\ L_{41} & L_{42} \end{bmatrix} \\ \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}^T \end{bmatrix} \begin{bmatrix} E_{11} \\ E_{12} \\ E_{13} \\ E_{14} \end{bmatrix} \left| \begin{array}{l} \Gamma_y \\ p \end{array} \right|
\end{aligned} \tag{4.61}$$

$$\begin{aligned}
\begin{bmatrix} x_1 \\ x_7 \end{bmatrix} &= \begin{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 & x_2 & x_7 & x_8 & \tilde{x}_1 & \tilde{x}_2 & \tilde{x}_7 & \tilde{x}_8 \end{bmatrix}^T + \\
\begin{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix} \end{bmatrix} \begin{bmatrix} \begin{bmatrix} E_{11} \\ E_{12} \\ E_{13} \\ E_{14} \end{bmatrix} \end{bmatrix} \left| \begin{array}{l} \Gamma_y \\ p \end{array} \right|
\end{aligned} \tag{4.62}$$

CHAPTER 5

EXPERIMENTAL CHARACTERIZATION

5.1 MANUFACTURING OF SPECIMENS

The fast prototyping of the specimens is developed to determine the mechanical response of the Creality CR Series PLA. Samples dimensions are implemented according to ASTM D638 standard using the type IV morphology for specimens, Figure 5.1.

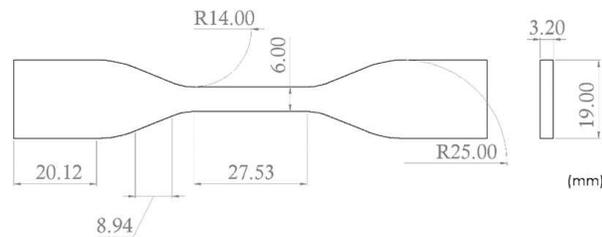


FIGURE 5.1: Type IV specimens for tensile testing (units in mm).

The manufacturing process employs a Creality Ender-3 FDM printer with a printing volume of 220mm x 220mm x 250 mm and a resolution of 0.1mm. Creality CR Series polylactic acid (PLA) filament is used as a material due to its high mechanical properties, such as good adherence to heat bed, no warping or deformation during manufacturing, and no need of printing environment isolation. The slicer se-

lection is selected based on the author experience, Matter-Control is used to create the machine code from the 3D model file. The specimens printing parameters are shown in Table 5.1.

TABLE 5.1: Printing parameters selected in Creality Ender-3 FDM printer.

FDM Manufacture Parameters	
Infill pattern	Hexagonal
Infill density	10 % - 15%
Printing orientation	0° - 90°
Shell thickness	3 layers
Nozzle size	0.4 mm
Extrusion temperature	200°C
Heat Bed temperature	60°C
Deposition speed	40 mm/s

The temperatures of the printing process are one of the most harmful properties if these are not selected correctly. The extruder and heat bed temperatures are settled at 200°C and 60°C respectively, these are selected based on filament supplier recommendations. The PLA is melted and extruded using a step motor to push the material and a nozzle with a diameter of 0.4 mm. The samples are manufactured with a layer thickness of 0.2 mm and a raster angle of +45° / -45° with the purpose of obtaining parts with improved mechanical properties. The entire MUAV components are designed to avoid the usage of supports in the manufacturing process, this is important since it allows material to save and improve manufacturing time. During the first layer of material deposition, the nozzle cooling is disabled, improving the adherence of the specimen and avoiding detachment of the warm bed.

In aeronautical manufacturing and material selection, the density of the net weight of the components is one of the most considerable properties to consider in usage; in UAV prototyping manufacture this is not the exception, thus, the specimens

are manufactured considering two low infill densities (10% and 15%) and eight different building directions (0° flat, 0° edge, 15° edge, 30° edge, 45° edge, 60° edge, 75° edge, 90° edge), Figure 5.2. The infill pattern is settled as hexagonal since it is considered the most suitable for structural purposes Harpool et al. (2021).

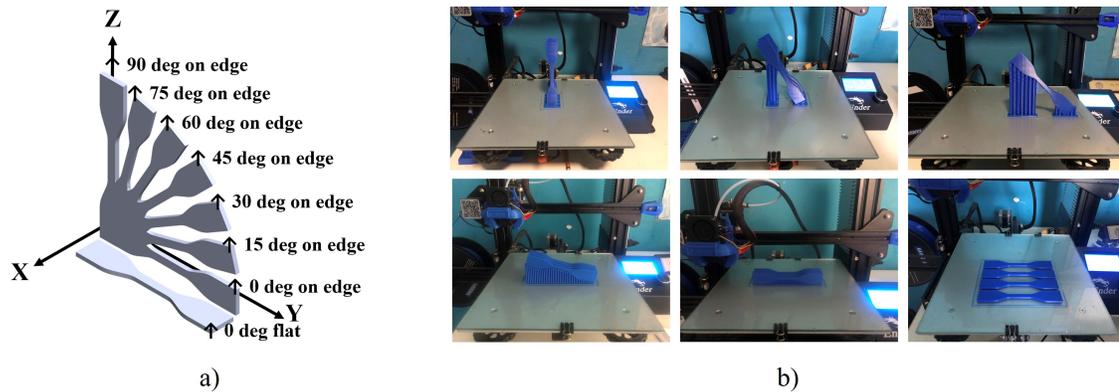


FIGURE 5.2: a) Specimens on edge at different building orientations and b) printed specimens with grid support pattern.

Several inconveniences arise from the specimens with a building angle higher than 15° edge, these have a lack of adherence to the heat bed, and even the first layer deposition of material cooling is disabled; thus, rafts are used for these specific scenarios. Raft parameters selection is based on the Matter Control recommendations, these include an expansion distance of 5 mm from the specimen and an air gap of 0.4 mm. Supports are needed for specimens from 15° to 75° since the geometry is not stable. Part b) shows the printed specimens, supports settled and grid support pattern, Figure 5.2.

5.2 MECHANICAL TESTING

The tensile testing is developed based on the ASTM D638 standard for polymers, the procedure preserves the desired specimen dimensions for specific purposes, permitted displacement speeds and machinery resolution, suitable atmospheric conditions also,

calculation of material properties is done based only on the equipment used. The procedure is done employing a universal testing machine with MTS Landmark brand with model 64,725, the software used for data acquisition of the axial extensometer and manipulation of the universal machine belongs to MTS Systems company: MTS Flextest 40, Station Manager. (Fig. 5.3) An axial extensometer allocated at the center of the specimen is used to have appropriate data on the elastic region, this uni axial piezoelectric sensor does not allow to calculating the poisson ratio of the material, only elastic modulus, however, for purposes of the present document, this is enough information.

Due to the thin layer thickness and low infill density used, the pressure of the grips is necessary to be quantified to avoid any undesired deformation or even cracking of the specimens before tensile testing. Specimens are clamped inside the load frame using standard grips provided by the supplier with no additions, the process is done with a gripper pressure of 800 psi, this value is carefully obtained using trial-error testing employing 10% of infill specimens manufactured at 45° , since these are considered as one of the most fragile of all the entire procedure.

All testing are conducted under ambient conditions with a temperature of 25°C and 30% of humidity and 101.4 kPa, these environmental variables are controlled using the specific cooling control system of the CIHA Testing laboratory. Moreover, a constant displacement velocity of 2 mm/min is used to separate the lower and upper load frame of the universal machine. A replication of four specimens for each building direction are tested to acquire more reliable data, this is also done since the PLA most often tends to break avoiding obtaining reliable data, this is caused when the universal machine grip pressure is not precisely defined or the equipment is not calibrated correctly.

The universal machine and uni axial extensometer from MTS Systems software provide several data, however, only specific parameters are required for the present document, from the extensometer only deformation and force on the load frame

are necessary data, moreover, it is necessary to measure the transversal length and thickness of the specimens with the purpose of calculating the transversal net area for future stress-strain curves plotting and analysis, see Figure 5.3.

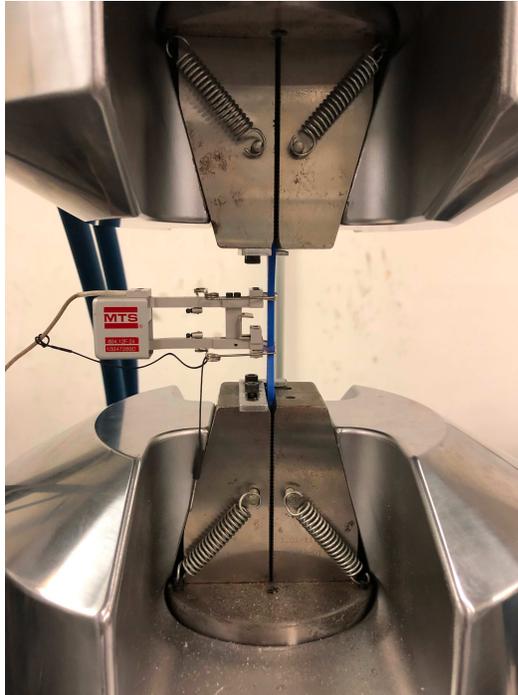


FIGURE 5.3: Tensile testing done according to ASTM D638 standard.

5.3 STATIC AERODYNAMIC CHARACTERIZATION

The MCR UAV V3.0 has a coaxial brushless motor attached at the top of the its fuselage, a coaxial rotor consists of two counter-rotating blades. The rotor is capable of counteracting the net weight of the vehicle and generate enough thrust to some required altitude, longitude, or latitude maneuver, moreover, the reactive torque between both rotors must be ideally zero. This means, that the rotational speed in hover must be equal, however, it is possible to find some relative low torque in $[Z]$ direction that can affect the altitude and yaw $[\psi]$ control, and it is necessary to characterize the magnitude of the mentioned low parasite torque.

Moreover, the vehicle has two control surfaces attached at the bottom, these are used to change the position over the axis $[XY]$ in a three-dimensional euclidean space, the ailerons generate torque over the vehicle gravity center. This is caused when the air produced by the coaxial rotor impacts the area of the mentioned surfaces. However, the low-weight servomechanism used to activate both control surface allows minimum vibrations when high air speeds are produced by the rotors, these vibrations can generate low parasite torques over the gravity center that results in $[XY]$ perturbations. It is quite important to characterize the net forces actuating since these can be used to refine the mathematical model proposed in Section chapter 3.

The experimental procedure is divided in two different parts:

- Brushless rotor characterization.
- Ailerons characterization.

The entire static rotor and vehicle characterization procedure are performed using a set of two non-invasive sensing systems. Firstly, an infrared laser generator in conjunction with a proximity sensor is used for RPM sensing, this is attached at the top of the rotor including a non-reflective tape. This is used since it allows the variation in the sensing reflection, this specific change in the reflection intensity allows predicting the RPM.

Secondly, a non-invasive PWM pulse reading works using the RC receiver and analog input pins of the Arduino Nano, then all the readings are packed and sent to the computer using a Bluetooth module to avoid cable interference with the air flux. See Figure 5.4

The experimental forces/moments are measured by a piezoelectric Sensor 100M from JR3 Multi-Axis Load Cell Technologies with their respective power supply of 120VAC and data acquisition bank supplied by the company, all this equipment is

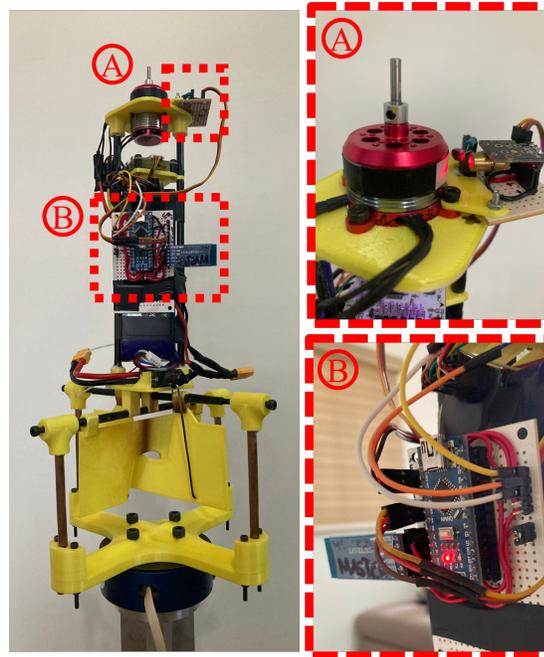


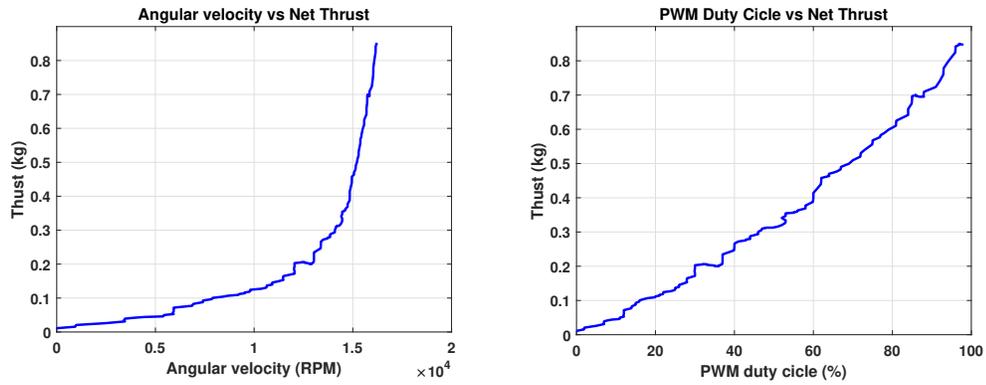
FIGURE 5.4: Sensors used for MCR UAV V3.0 static characterization.

connected to a Dell Precision T7500 with Intel Xeon, 32GB of RAM memory DDR3, the experimental acquisition process is done at a frequency of 50Hz.

5.3.1 BRUSHLESS ROTOR CHARACTERIZATION

In this section, the coaxial rotor is characterized, to experimentally obtain the PWM signal and RPM required to overcome the weight of the vehicle simulating a hover maneuver, it is important to clarify that the vehicle weight is 0.578 Kg. Moreover, the net reactive torque must be calculated for hover missions where no torque difference is required or no yaw angle is required.

The coaxial rotor is placed over the JR3 piezoelectric electric sensor to obtain the thrust and moment properties of the coaxial rotor. In Figure 5.4 it is possible to visualize the non-invasive RPM sensor attached to the rotor, this element is possible to be used without the vehicle structure. However, the PWM sensing com-



(a) Angular velocity against thrust generated response. (b) PWM signal against thrust generated response.

FIGURE 5.5: CRM2409 characterization results.

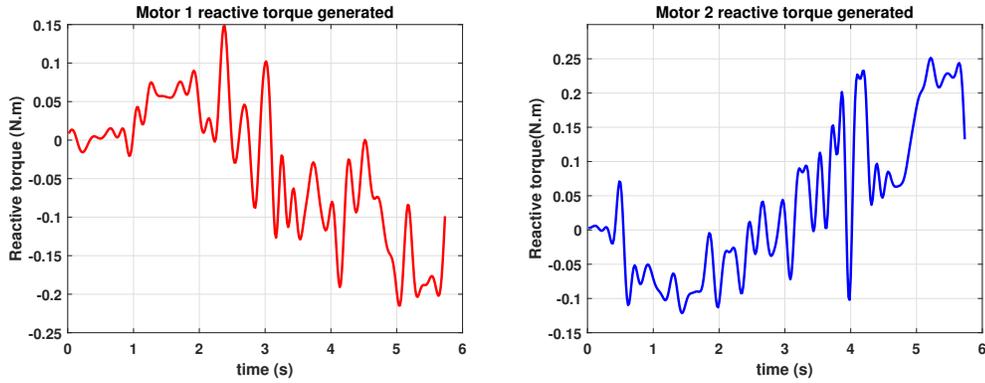
ponents must be reallocated over the JR3 aluminum structure to avoid the ground aerodynamic effect.

Based on the data obtained in this procedure, the net MCR UAV V3.0 weight can be overcome at %73 of the total PWM signal produced by the radio controller Figure 5.6b and 15,500 RPM of the brushless rotor. Figure 5.6a. The net weight of the vehicle is remarked with a red dot over the response.

At the same time that the lectures of the generated coaxial rotor forces are taken, reactive moments of each rotor can be also obtained, however, since the brushless rotor is coaxial, it is not possible to perform this specific procedure at the same moment for each rotor individually, it is necessary to perform the process one more time. Figure 5.6

5.3.2 AILERONS CHARACTERIZATION

The ailerons characterization can be performed after the simple brushless rotor procedure, this is, since is required to know the exact PWM or angular velocity in RPM values required to overcome the vehicle net weight (0.578 Kg). The justification



(a) Rotor 1 reactive torque generated. (b) Rotor 2 reactive torque generated.

FIGURE 5.6: Reactive torque generated by both rotors individually.

for this is that is essential to carry out the aerodynamic characterization for both control surfaces considering ideal hover conditions.

The procedure is performed using a pre-build PWM signal, this signal ensures the deflection of both ailerons from 0 to 20 degrees in a total time of 2 seconds, the logic of the signal is as follows: it must vary every 0.5 seconds the aileron 5 degrees, this must be repeated 4 times for each aileron independently, thus, this procedure is repeated twice, one for each control surface.

The control surfaces results for the first procedure over X axis are almost zero, this response is expected since the aileron over this axis is not deflected, nevertheless, the peak of this signal is 0.05 N which can be used for the controller as a saturated perturbation. Figure 5.7a Moreover, for Y axis the peak response is 0.4 N with an increment from 10 degrees. Figure 5.7b, in this axis the aileron is deflected.

The second procedure response over X axis is increasing uniformly unlike the first procedure, the peak response is 0.5 N. Figure 5.8a. For Y axis the peak response is 0.2 N unlike the first procedure, the variance at zero degrees is higher, this can be attributed to the morphology of the first aileron. The passive response over the Y axis can be used as disturbance in the mathematical model. Figure 5.8b it is important to mention that a not exact behavior is expected since both ailerons have

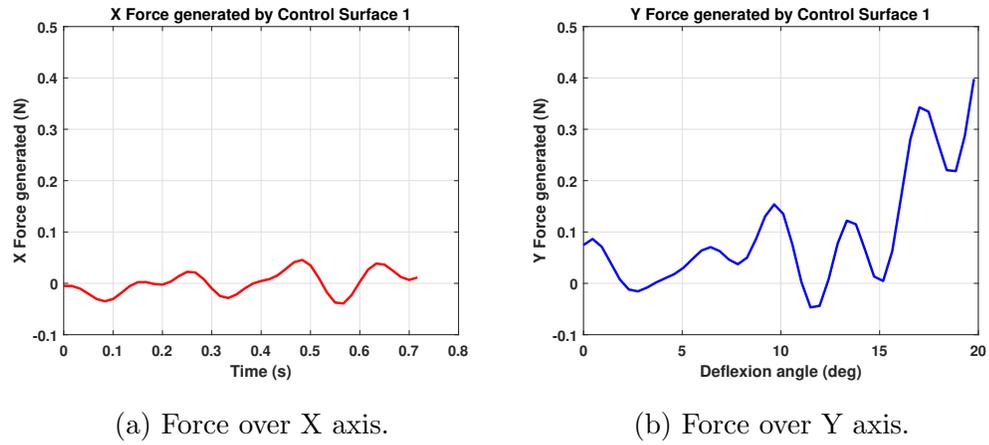


FIGURE 5.7: Forces generated for first control surface.

a different shape configuration.

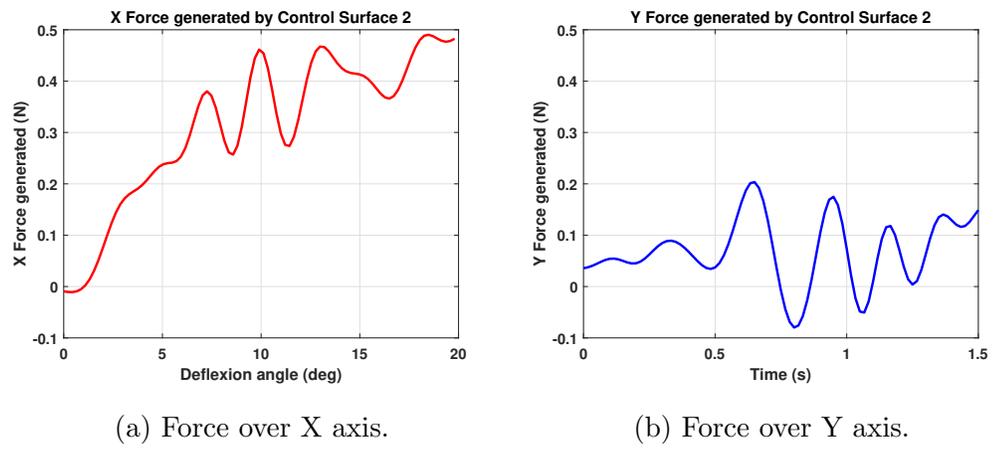


FIGURE 5.8: Forces generated for second control surface.

CHAPTER 6

COMPUTATIONAL PROCEDURE

6.1 COMPUTATIONAL FLUID DYNAMICS (CFD)

The computational fluid dynamic (CFD) analysis of MCR UAV V3.0 is performed employing ANSYS, Fluent. Then, a finite element structural analysis is done making usage of the static structural module of ANSYS. The CFD simulation is carried out to find out the pressure distribution around both ailerons areas, then, the structural analysis is used to quantify the total deformation and maximum Von-Mises Stress over the structure. Moreover, an authors previous work is based on the calculation of lift and drag coefficients to evaluate the aerodynamic effect of the ailerons (Dominguez et al., 2022). Figure 6.1 presents a flow diagram of the methodology utilized in this work during different stages of design and numerical model.

The first two tasks developed on the flow diagram are entirely dedicated to the Computer Aid Design (CAD) creation and simplification of it, this last step is necessary before performing any computational and numerical study, this is done with the purpose of saving computational resources in the numerical procedure, next sections are dedicated to CFD analysis performing, this preserving as much as possible normal environmental conditions based on ISA and MCR UAV V3.0 hovering propulsion, is extremely important to mention that before to finish CFD

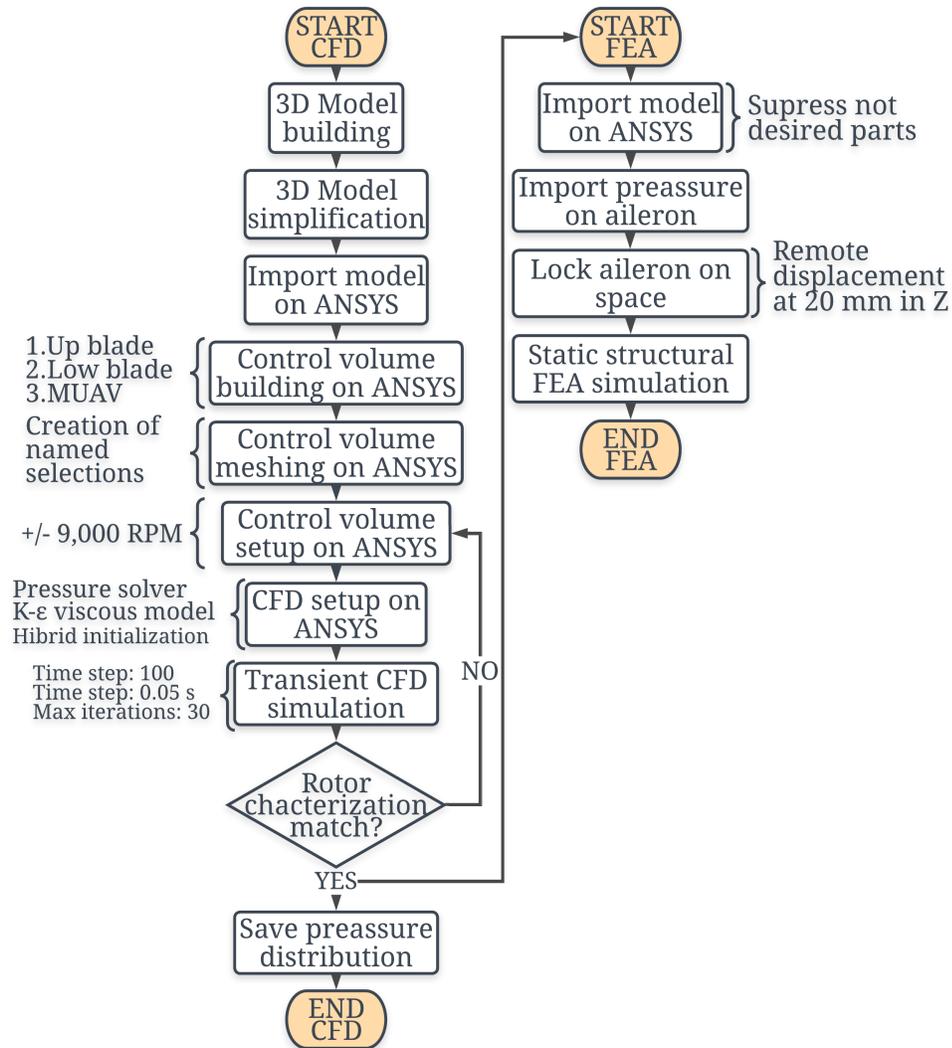


FIGURE 6.1: Flow diagram of the numerical study of MCR UAV V3.0.

calculations is necessary to verify that upper and lower blades are spinning on the correct direction, in the center of rotation and over the center of the control volumes. The model simplification stage also tries to preserve only the aerodynamic elements that could affect the behavior of the surfaces or consume too many computational resources, at the end of this step the model consists of only two coaxial counter-rotating blades 8040, the 3300 mAh LiPo shape fuselage and both control surfaces. Three control volumes are defined for the CFD study, as depicted in Figure 6.2.

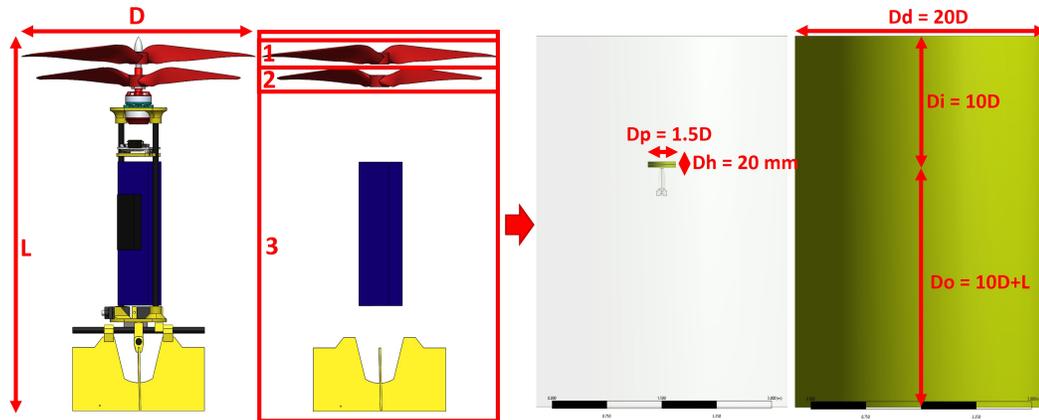


FIGURE 6.2: Control volumes and model simplification of CFD study of MCR UAV V3.0. 1 and 2- blades and 3- bullet shape and control surfaces.

In total three control volumes are defined for the present CFD analysis, two rotating volumes are defined for the coaxial blades; upper and lower, these are immersed in a bigger control volume, and an additional volume is defined for the entire vehicle, see Figure 6.2, A cylindrical morphology is used for all of the presented elements, is important to mention that the vehicle possesses a symmetric morphology in X and Y axis, and the propulsion system is allocated at the top affecting both axes equally, a cylindrical shape guarantees a homogeneous distance between all the vehicle streams, this is necessary since the hovering mode is analyzed with a free stream. (Mo et al., 2021)

The dimensions of the described control volumes are presented in Table 6.1. The values are selected based on the vehicle length (L) and the 8040 propellers diameter (D) which corresponds to 365 mm and 203.2 mm respectively. The distance from the rotating elements to the inlet wall (D_i) is considered as ten times the propeller diameter ($10D$), the distance from rotating elements to the outlet wall (D_o) is considered ten times de propeller diameter plus the length of the vehicle ($10D + L$), the diameter of the vehicle fluid domain (D_d) is considered as twenty times the propeller diameter ($20D$). Finally, for the propeller, the diameter of the fluid domain (D_p) is defined as $1.5D$ the diameter of the propeller and a high (D_h)

of 20 mm, this last dimension is selected based on the coaxial rotor dimensions. (Mo et al., 2021) Figure 6.2.

TABLE 6.1: Control volume dimensions of MCR UAV V3.0.

Control Volume Dimensions			
Component	Diameter (m)	Height (m)	Center location(m)
Low blade	0.381	0.020	0, 0, 0.025
High blade	0.381	0.020	0, 0, 0.025
Vehicle	3.048	4.080	0, 0, 0.25

The mesh is created contemplating a sphere of influence with a diameter of 0.25 m, which is positioned at -0.460 mm in Y direction, this is generated to create a more refined mesh in this specific zone where both ailerons are allocated, moreover, inflation layers are used over the control surface area these with a 3 mm of the total thickness, see Figure 6.3. The mesh created for Scenario A contains 159,923 nodes and 894,947 elements in total. The mesh quality obtained shows an acceptable value with an element metric of 0.827. Moreover, with the objective of assuring optimal mesh quality, several simulations are conducted before defining a final methodology to build the mesh, the simulation is performed and monitoring the maximum pressure obtained over both control surfaces and their effect of them. As a result, it is possible to obtain optimal element sizes for all the control volumes.

Transient simulation is performed with pressure-based solver including gravity magnitude of 9.81 m/s. A $\kappa - \epsilon$ realizable turbulence model with scalable wall function is selected in the presented numerical study. Kutty and Rajendran (2017) presented evidence that this turbulence model results more efficiently than $\kappa - \omega$ when the Reynold number of the propellers is higher than 100,000. A hybrid initialization is performed to obtain suitable convergence from the inlet, the transient solution parameters include a time step size of 0.005 seconds, 100-time steps, and a maximum of 30 iterations per time step, these parameters are selected based on an

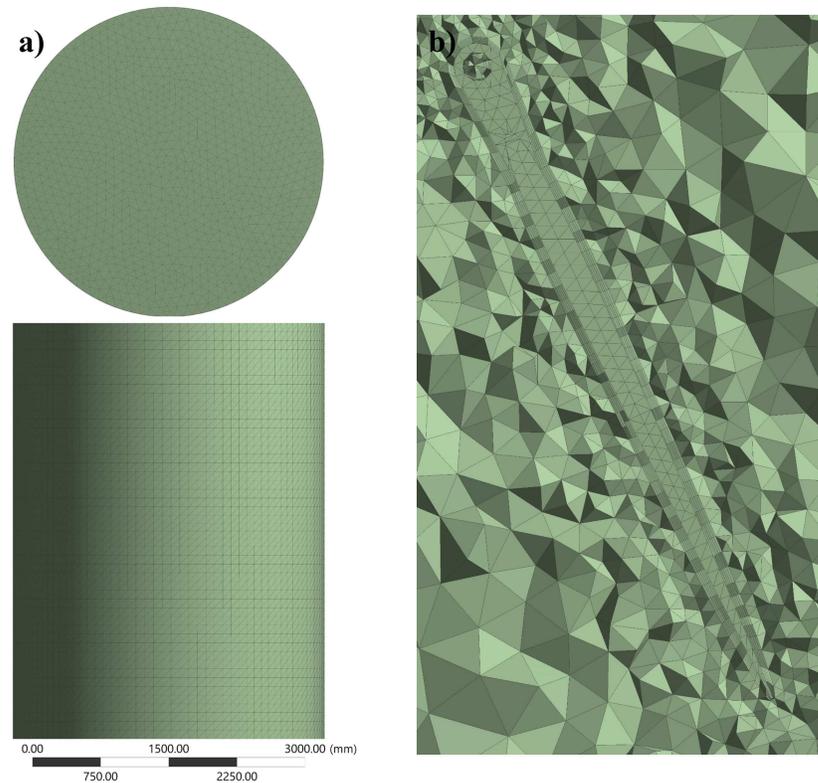


FIGURE 6.3: Control volume mesh. a) Frontal and superior view of entire control volumes. b) Inflation layers around deflected control surface.

iterative process done, where the settlement time for the fluid around the vehicle is at 2.5 seconds of simulation, however, in order to assure a more convenient response, the simulation is settled to a maximum time of 5 seconds.

6.2 FINITE ELEMENT ANALYSIS (FEA)

The numerical model is focused on the critical loading scenarios over the vehicle presented. A parametric study is carried out for both ailerons individually with a variation of the wall thickness values from 0.2 to 1.2 mm. The mechanical properties used are those corresponding to Creality CR Series polylactic acid (PLA) with an infill of 15% obtained from the mechanical testing. The three-dimensional model is simplified the same as in the CFD analysis, this is to evaluate only the deflected con-

trol surface. It is important to mention that this component is a three-dimensional model not as a solid element but as a surface, this can vary the thickness of all the walls on the static analysis. Most of the elements built on the meshing process are quadratic cell, this is established as the main dominant element for the FEA process, a face sizing is included for areas overall for all the scenarios discussed, this consist of the definition of an element size of 3 mm with a hard behavior. The mesh generated for scenario A consists of 7107 elements with 6981 nodes.

The restrictions for ailerons displacement are assigned in X and Z axis; thus, the surface must be fixed on these axes while Y axis deformations are allowed, this is done since the most critical failure of a previously built prototype was presented in this specific direction. (Dominguez et al., 2022), the length between the aileron and a remote displacement, that restricts the deformation of the surface in X and Y axis, is 20 mm in the positive Y direction (on the top of the control surface). A remote displacement is selected as the most suitable restriction element to simulate this condition. The FEA model map the pressure obtained from the CFD analysis, this is selected as a boundary condition. The imported pressure is mapped over the entire area of the aileron analyzed in the respective scenario as shown in Figure 6.4. The obtained pressures from CFD and the mapped values on FEA are described Table 6.2 and Table 6.3 respectively.

TABLE 6.2: CFD Computed forces.

Obtained Forces From CFD			
Control surface	X-Component (N)	Y-Component (N)	Z-Component (N)
A	-0.9343	-0.4528	-1.27E-03
B	-0.2649	-9.0923E-02	-5.0185E-04
C	5.5128E-04	-0.1177	-0.2427
D	8.3823E-04	1.1462E-02	-3.4074E-02

For A and B scenarios is possible to visualize that the results obtained in Y

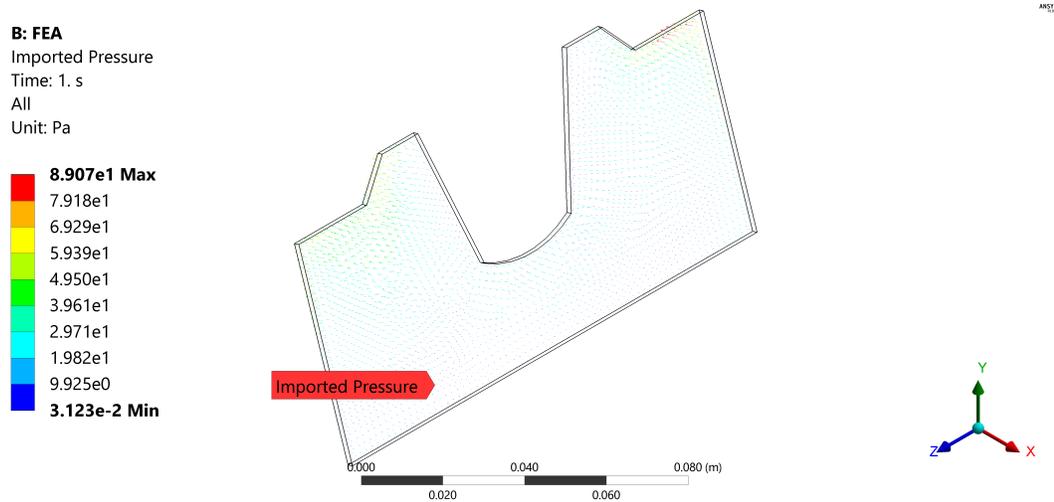


FIGURE 6.4: CFD imported pressure mapped over the control surface.

TABLE 6.3: FEA Computed mapped force.

Obtained Forces From FEA			
Control surface	X-Component (N)	Y-Component (N)	Z-Component (N)
A	-0.9016	-0.4528	-1.27E-03
B	-0.2507	-9.0923E-02	-5.0185E-04
C	5.5281E-04	-0.1142	-0.2382
D	5.0111E-04	8.9910E-03	-2.2630E-02

and Z directions are near to zero compared with X axis, this confirms that the remote displacement used correctly restricts the deformations on X and Z axis, thus, is the most suitable option for the analysis, also, for scenario C and D, the forces are almost neglectable on X and Y direction. It is visualized that the vectors with higher magnitude are in the predicted direction, this since the air streams impact the surface from Y direction. It is also noted that the results obtained on X axis are near to zero since the aileron is not impacted by any fluid in this direction.

6.3 OPTIMAL ADAPTIVE IRL

The computational procedure used for the generation of the control scheme for the Optimal adaptive IRL is elaborated in Matlab from MathWorks, this software is preferred to analyze complicated control systems used and build for scientist. The presented Optimal adaptive IRL algorithm is programmed following a VTOL mathematical model, the main objective is to iterate the gain K produced to stabilize the vehicle from random states $[x\dot{x}\theta p]$ initial conditions, the new iterated must be more optimal than the previous one, in the sense of expense less energy to keep the vehicle in hover position.

Firstly, is necessary to declare the basic state-space model of the vehicle, this is done without considering any disturbance, observer or any addition, moreover, define the states regulator matrix Q and output regulator matrix R , this help to calculate an initial K gain that is used to iterate a new controller.

```
%ONLINE IRL of a QUADROTOR considering only DoF axis (X PHI)
clc
clear all
g = 9.81;
%-----STATE SPACE-----
A = [0 1 0 0;0 0 -g 0;0 0 0 1;0 0 0 0]; %[x u phi p]
B = [0 0 0 1]'; %Mx
C = [1 0 0 0;0 0 1 0]; %[x;phi]
D = [0 0]';
G = ss(A,B,C,D);
%Controlability
co = ctrb(G);
controllability = rank(co);
%DEFINIR Q & R
```

```

Q = C'*C*7000;
R = 1;
%Define initial controller
P = care(A,B,Q,R);
p0 = [P(1,1) P(1,2)*2 P(1,3)*2 P(1,4)*2 P(2,2)...
      P(2,3)*2 P(2,4)*2 P(3,3) P(3,4)*2 P(4,4)]';
K = lqr(A,B,Q,R)

```

Then is required to calculate the reference gain used to eliminate the steady state error produced in a LQR controller, since this type of scheme functions similarly as a proportional controller, then, the controlled plant in closed-loop is computed.

```

%REFERNCE GAIN
Cn = [10 0 10 0];
sys_ss = ss(A,B,Cn,0);
Nbar = rscale(sys_ss,K);
GC = ss(A-B*K,B*Nbar,C,D);

```

Once the normal plant is defined is required to compute the state-space plant considering the observer, this is defined in (4.42) and (4.43), however, before to performing this task the L gain that assures the convergence of the estimates stated must be defined, this is done based on the slowest pole obtained by the closed-loop A matrix $(A-BK)$, for this algorithm, the slowest pole corresponds to $-2.2+/-7.35j$, thus, L matrix elements must be 10 times faster in a root-locus analysis.

```

%OBSERVER
ob = obsv(G);
observability = rank(ob);
poles = eig(A-B*K);
P = [-22 -23 -24 -25];

```

```

L = place(A',C',P)';
A_0 = [(A-B*K) (B*K);
        zeros(size(A)) (A-L*C)];
B_0 = [B*Nbar;
        zeros(size(B))];
C_0 = [C zeros(size(C))];
D_0 = [0;0];
GC_0 = ss(A_0,B_0,C_0,D_0);

```

The previous state-space mathematical model employing a full-state observer is simulated using *lsim* Matlab command and employing random initial conditions x_0 , this is the purpose of the controller, it must be capable to stabilize the vehicle without any impediment of these conditions, which is remarkable to mention that this process is done with a time-step of 0.005 seconds, these first results will be used to calculate the new K controller.

```

%CLOSED LOOP SIMULATION
t = 0:0.005:.5;
r = .4 + 0*ones(size(t));
xo = [1 0 0.1 0];
[y,t,XX] = lsim(GC_0,r,t,[xo 0 0 0 0]);
X = [XX(:,1) XX(:,2) XX(:,3) XX(:,4)];

```

This last computation produces the states and outputs of the algorithm for a certain time interval (0-0.5) seconds, the obtained states are used to compute the control input u and in addition to the outputs y the cost function V is calculated:

```

%CALCULATION U
u = (- K*X')';
n = length(t);

```

```

%CALCULUS V
for i = 1:n
Vdot(i) = X(i,:)*Q*X(i,:)' + u(i,:)'*R*u(i,:);
end
Vdot = Vdot';
V = cumtrapz(t,Vdot);
T = n;

```

Now, the variables used for the least square algorithm must be computed, in this algorithm, there are two types of variables: dependent Y and independent x , the dependency is around the previously calculated V cost function, this is required to calculate Y variable and x only required of the generates state itself. Both processes are done based on an iterative process employing a *for* command. For the convenience of the document elongation the entire iteration is not included, however, is possible to visualize the complete iteration in the Appendix A.

```

%CALCULATION Y
for k = 1:n
Y(k) = V(T,1)-V(k,1);
end
Y = Y';
%CALCULATION x
for k = 1:n
x1(k) = X(k,1)^2-X(T,1)^2;
end
for k = 1:n
x2(k) = X(k,1)*X(k,2)-X(T,1)*X(T,2);
end
...
for k = 1:n

```

```

x10(k) = X(k,4)^2-X(T,4)^2;
end
x = vertcat(x1,x2,x3,x4,x5,x6,x7,x8,x9,x10);

```

The independent must be optimized in the least square sense in comparison with the dependent variable, is remarkable to mention that this algorithm considers a linear approximation, which results in the vector p that contains the unique non-repeated elements of the Riccati matrix P , thus, the actualization of the control police K can be effectuated.

```

%CALCULATION p
p = inv(x*x')*x*Y; %LEAST SQUARES
%CALCULATION P
P = [p(1,1) p(2,1)/2 p(3,1)/2 p(4,1)/2;
     p(2,1)/2 p(5,1) p(6,1)/2 p(7,1)/2;
     p(3,1)/2 p(6,1)/2 p(8,1) p(9,1)/2;
     p(4,1)/2 p(7,1)/2 p(9,1)/2 p(10,1)];
K = inv(R)*B'*P

```

The new control policy K assures to be more optimal in the sense of energy expense, this step is the last of the iterative process. From this point the calculation of the reference gain N must be calculated again with the new gain K , thus, perform another closed-loop simulation from the 0.5 seconds, is important that the closed-loop simulation contain more than 10 time-steps, this is required since the Riccati matrix has 10 independent elements. The entire algorithm can be visualized in the Appendix A.

6.3.1 OPTIMAL ADAPTIVE IRL CONSIDERING OUTPUT DISTURBANCE

Same as in the previous explained algorithm, the first task to do is to declare the basic state-space model of the vehicle, this is done without considering any disturbance, observer or any addition, moreover, define the states regulator matrix Q and output regulator matrix R , this help to calculate an initial K gain that is used to iterate a new controller.

```
%ONLINE IRL of a QUADROTOR considering only DoF axis (X PHI)
clc
clear all
g = 9.81;
%-----STATE SPACE-----
A = [0 1 0 0;0 0 -g 0;0 0 0 1;0 0 0 0]; %[x u phi p]
B = [0 0 0 1]'; %Mx
C = [1 0 0 0;0 0 1 0]; %[x;phi]
D = [0 0]';
G = ss(A,B,C,D);
%Controlability
co = ctrb(G);
controllability = rank(co);
%DEFINIR Q & R
Q = C'*C*7000;
R = 1;
%Define initial controller
P = care(A,B,Q,R);
p0 = [P(1,1) P(1,2)*2 P(1,3)*2 P(1,4)*2 P(2,2)...
      P(2,3)*2 P(2,4)*2 P(3,3) P(3,4)*2 P(4,4)]';
K = lqr(A,B,Q,R)
```

Then is required to calculate the reference gain used to eliminate the steady state error produced in a LQR controller, since this type of scheme functions similarly to a proportional controller, then, the controlled plant in a closed-loop is computed.

```
%REFERNCE GAIN
Cn = [10 0 10 0];
sys_ss = ss(A,B,Cn,0);
Nbar = rscale(sys_ss,K);
GC = ss(A-B*K,B*Nbar,C,D);
```

The perturbation signals are obtained using Simulink from Matlab, these are modeled using white gaussian noise and a transfer function of the first order that helps to determine the frequency of the disturbance, then, these signals are exported to a .mat file. The block diagram used is represented in Figure 6.5.

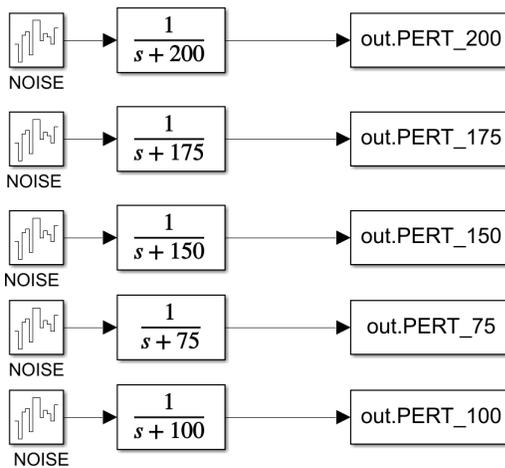


FIGURE 6.5: Block diagram used to export disturbance signals.

The obtained graphs that represent the perturbations used in the proposed algorithm are depicted in Fig. 6.6.

Once the Simulink program calculates the perturbations in time and export them in a .mat file, this must be imported to the optimal adaptive IRL script, this can be done following:

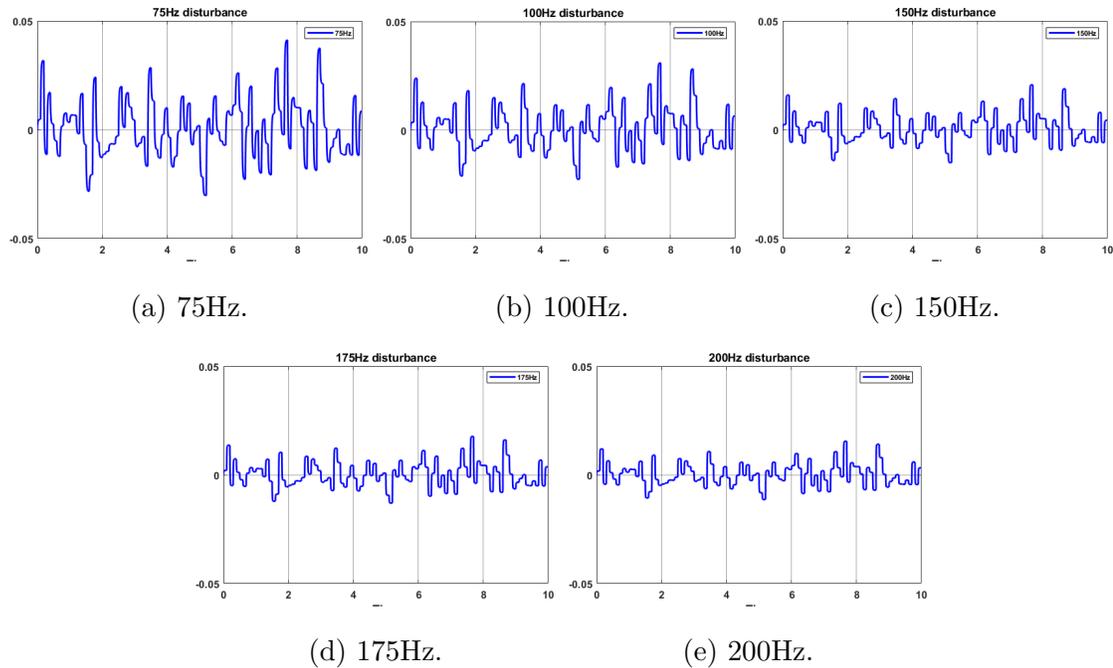


FIGURE 6.6: Disturbances at different frequencies.

```

%PERTURBTIONS
Perturbations = importdata('Perturbations.mat');
P_OUT = Perturbations.PERT_200';
F = [1 0 1 0]'; %[x u phi p]

```

Once the normal plant is defined and the perturbations saved in the workspace of Matlab, is required to compute the state-space plant considering the observer, this is defined in (4.61) and (4.62), same as in the adaptable optimal IRL algorithm, the L gain must be defined, this is done based on the slowest pole obtained by the closed-loop A matrix ($A - BK$). Note that the B matrix is now modified, including the terms of the disturbance and control input.

```

%OBSERVER
poles = eig(A-B*K); %-2.22+/-2.22j
P = [-22 -23 -24 -25];
L = place(A',C',P)';

```

```

A_0 = [(A-B*K) (B*K);
        zeros(size(A)) (A-L*C)];
B_0 = [B*Nbar -B*K*F;
        zeros(size(B)) -L*C*F];
C_0 = [C zeros(size(C))];
D_0 = 0;
GC_0 = ss(A_0,B_0,C_0,D_0);

```

The previous state-space mathematical model employing a full-state observer is simulated using *lsim* Matlab command and employing random initial conditions x_0 , this process is done with a time-step of 0.005 seconds, these first results will be used to calculate the new K controller. Note that at the end of the output the term CEp is added according to (4.62).

```

%CLOSED LOOP SIMULATION
t = 0:0.005:.25;
r = 0*ones(size(t));
xo = [1 0 .1 0];
[y,t,XX] = lsim(GC_0,[r; P_OUT(1,1:51)],t,[xo 0 0 0 0]);
X = [XX(:,1) XX(:,2) XX(:,3) XX(:,4)];
y = y + (C*F*P_OUT(1,1:51))';

```

This last computation produces the states and outputs of the algorithm for certain time interval (0-0.25) seconds, the obtained states are used to compute the control input u , also, as it is proposed in (4.46) the control law is modified when an output perturbation is considered the term KEp is added to the equation. Moreover, employing the outputs y of the system, the cost function V is calculated:

```

%CALCULATION U
u = (- K*X')' - (K*F*P_OUT(1,1:51))';

```

```

n = length(t);
for i = 1:n
Vdot(i) = X(i,:)*Q*X(i,:) + u(i,:)'*R*u(i,:);
end
Vdot = Vdot';
V = cumtrapz(t,Vdot);
T = n;

```

Now, the variables used for the least square algorithm must be computed, in this algorithm, there are two types of variables: dependent Y and independent x , the dependency is around the previously calculated V cost function, this is required to calculate Y variable and x only required of the generates state itself. Both processes are done based on an iterative process employing a *for* command. For the convenience of the document elongation the entire iteration is not included, however, is possible to visualize the complete iteration in the Appendix B

```

%CALCULATION Y
for k = 1:n
Y(k) = V(T,1)-V(k,1);
end
Y = Y';
%CALCULATION x
for k = 1:n
x1(k) = X(k,1)^2-X(T,1)^2;
end
for k = 1:n
x2(k) = X(k,1)*X(k,2)-X(T,1)*X(T,2);
end
...
for k = 1:n

```

```

x10(k) = X(k,4)^2-X(T,4)^2;
end
x = vertcat(x1,x2,x3,x4,x5,x6,x7,x8,x9,x10);

```

The independent variable must be optimized in the least square sense in comparison with the dependent variable, this results in the vector p that contains the unique non-repeated elements of the Riccati matrix P , thus, the actualization of the control police K can be effectuated.

```

%CALCULATION p
p = inv(x*x')*x*Y; %LEAST SQUARES
%CALCULATION P
P = [p(1,1) p(2,1)/2 p(3,1)/2 p(4,1)/2;
     p(2,1)/2 p(5,1) p(6,1)/2 p(7,1)/2;
     p(3,1)/2 p(6,1)/2 p(8,1) p(9,1)/2;
     p(4,1)/2 p(7,1)/2 p(9,1)/2 p(10,1)];
K = inv(R)*B'*P

```

The new control policy K assures to be more optimal in the sense of energy expense, this step is the last of the iterative process. From this point the calculation of the reference gain N must be calculated again with the new gain K , thus, perform another closed-loop simulation from the 0.25 seconds. The entire algorithm can be visualized in the Appendix B.

This methodology is repeated twice until getting two more control policies K , this assures the correct functioning of the algorithm; however, the iteration process could continue generating more controllers. If the plant is already stabilized and considering that as most adaptive control algorithms excitation signals are required, the controllers generated will have high gains that could result in a noisy results. This process is evaluated, the control algorithm is described in the Appendix C.

CHAPTER 7

MCR UAV V3.0 MANUFACTURE AND FLIGHT TESTING

7.1 ADDITIVE MANUFACTURING

The fuselage and control surfaces are manufactured employing FDM technique. The printer used is an Ender 3 from Creality brand with a printing volume of 220 mm x 220 mm x 250mm. The vehicle components are manufactured using Creality PLA CR Series with a nominal diameter of 1.75 mm +/- 0.2 mm, the layer thickness is settled to a resolution of 0.1 mm, the hexagonal infill is selected as the most suitable according to the high mechanical properties reported in literature (Harpool, 2016) and an infill of 15%. Temperatures are one the most important characteristic to carry out in FDM process, the nozzle and bed temperatures are settled to 200°C and 60°C respectively. Another highly important element in this process is the slices, which is the software used to keep a correct material deposition in the heat bed, the slicer used to generate the layer code is Matter Control from MatterHackers, this is selected based on the author personal experience.

It is important to mention that special consideration is taken in to account in the 3D modeling process with the purpose of avoiding a complicated and late

FDM procedure. Firstly, all the components are designed to avoid the need of complicated supports, this can be achieved using fillets, divided components and increasing the deposition speed on a bridging section. Secondly, all the components that are necessary to be jointed are designed considering a gap of 0.2 mm in each joint, to generate a thin cavity to add two-components epoxy adhesive. The vehicle structural elements and ailerons printed can be depicted in Figure 7.1.



FIGURE 7.1: Components of MUAV printed by FDM, a) base supports b) control surfaces.

The optimized ailerons take 2 hours of printing time with a net weight of 0.029 kg; in comparison with the previous robust control surfaces a considerable reduction is done. Since the manufacturing time is near to 20 hours and a net mass of 0.067 kg, the net manufacturing time and resources are reduced considerably. The upper and lower plates are elements designed to hold the coaxial brushless motor and servomechanism respectively, the net mass of each element corresponds to 0.009 kg and 0.012 kg; these components are linked using two carbon fiber rods. The flight computer plate is designed to firmly hold the mentioned component, this must be stable and rotated to allow USB connection with the electronic device, the total mass resulted in 0.009 kg in conjunction with the elements used to attach the ailerons to another two carbon fiber rods Figure 7.2.

The entire mass of the MCR UAV V3.0 results in 0.578 kg; this value considers two carbon fiber rods with a longitude of 200 mm used to link the upper and lower

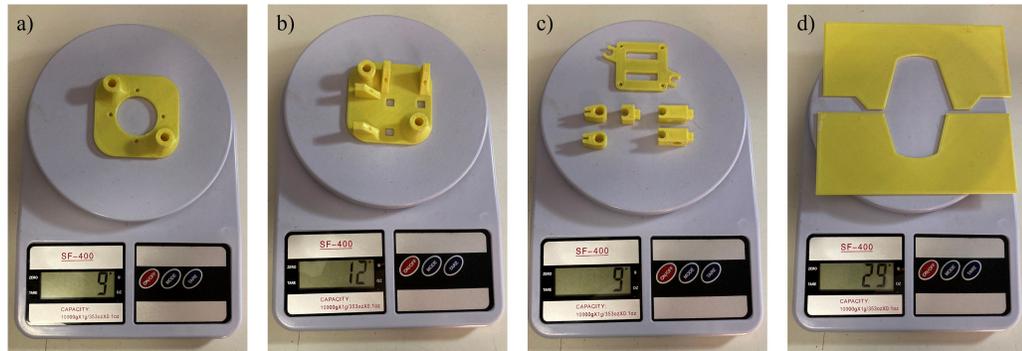


FIGURE 7.2: a) Upper plate mass, b) lower plate mass, c) flight computer plate and joints mass, d) ailerons plate mass.

PLA components (Upper and lower plates) that constitute the fuselage of the vehicle. Moreover, two more rods of 150 mm are required to attach the control surfaces to the body of the fuselage, these elements are used to deflect the ailerons over one simple node in the structure; the entire structure without any avionics included weights a mass of 0.072 kg.

7.2 VEHICLE ASSEMBLY

The components are assembled following an established path to assure the quality of the final vehicle. Firstly, the top and bottom plates are linked by employing a set of two carbon fiber rods with an elongation of 200 mm. These are jointed using Devcon two-component epoxy glue, this specific adhesive assures a heat resistance of 100°C and high mechanical resistance; this can be used even to fix metal cracks. It is highly important to mention that this process is relevant since the entire balance of the vehicle depends on the correct connection of both plates; a level is used to perform this task. The curing time of the adhesive rounds is about 2 hours.

Secondly, the ailerons are attached by pressure and epoxy adhesive; the ailerons stuck are carefully allocated to the bottom plate despite that this component includes guide holes to avoid misalignment. Moreover, two carbon fiber rods with an elonga-

tion of 140 mm are attached to the previously stuck aileron links. These elements are positioned on the structure by pressure, however, the rods are able to be displaced if any force is applied in Y or X direction. These are fixed using insulating tape, rounding twice the rods into the edge of stuck aileron links.

Lastly, to conclude the assembly, both control surfaces are joined to the aileron links previously positioned. These components are designed to allow the link by pressure, however, using hot silicone is also included to assure a rigid attachment. These elements are not linked using epoxy adhesive since it is necessary to be precise with the allocation of the ailerons. Some movement in their position of them could affect the desired behavior, thus, a fast-curing adhesive must be selected; the curing time of the hot silicone is rounding 5-10 s. Once the control surfaces are attached, these elements are now restricted in horizontal displacement, see Figure 7.3.

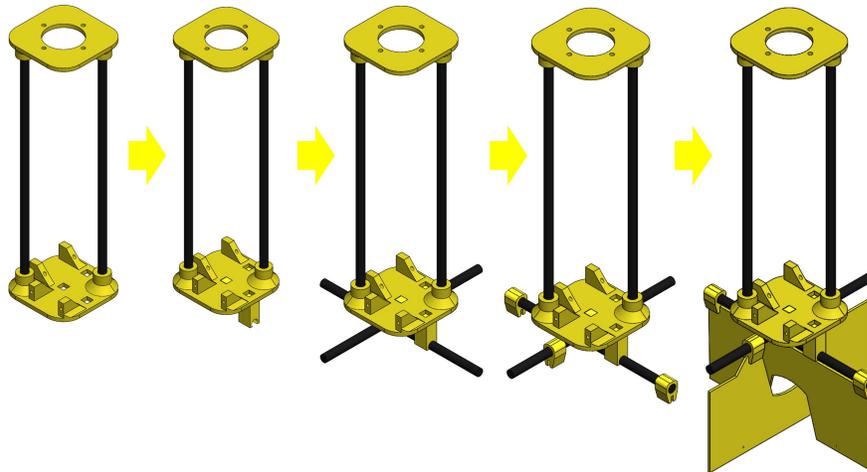


FIGURE 7.3: MUAV Structural assembly progress.

7.3 AVIONICS ASSEMBLY

The servomotors are positioned over the lower plate employing four 2M screws with a diameter of 2 mm and a chord elongation of 7 mm, these are placed to manipulate the control surfaces. Over both servos, the Zippy 3300mAh LiPo battery is firmly

located and fixed using black insulating tape to avoid any movement; this is done since including another PLA results in an increase in the vehicle weight. This step is carefully done since the battery connection cables for the power supply and cell balance must be physically liberated for future charging of the battery. At both sides of the battery two 30A Turnigy ESC are allocated, one on each side to keep in Z axis the gravity center of the vehicle, the ESC allows to manipulate of the PWM signal from the transmitter to control the brushless rotor. This specific procedure described of the process is named as the power stage assembly, where the voltage supply components and servos are attached, see Figure 7.4.

The next stage of the assembly procedure is called as the control stage; the CC3D flight computer is positioned over the battery using the flight controller plate, this is done by adding 10 mm between both elements. The computer is fixed to the plate using four rubber vibration insulators with M3 screws. The radio receiver is placed over the aileron links, to avoid a displacement of the vehicle gravity center. Lastly, the coaxial rotor is placed and fixed over the upper plate at the top of the vehicle using four M3 screws with a diameter of 3 mm and a chord elongation of 7 mm. The rotor is placed with two 8060 CW/CCW blades, these are fixed employing two 2M screws for each blade and a set of machined aluminum pieces provided by the rotor supplier, see Figure 7.4.

7.4 FLIGHT COMPUTER PROGRAMMING

A flight computer CC3D revolution is programmed using Librepilot; the software includes a coaxial helicopter vehicle with two servomechanisms attached to the swashplate to manipulate roll and pitch displacement. This configuration results highly similar to the MCR UAV V3.0, thus, this configuration is selected, however, the range of manipulation for the servomotors is manipulated, double direction is allowed to the mechanism.

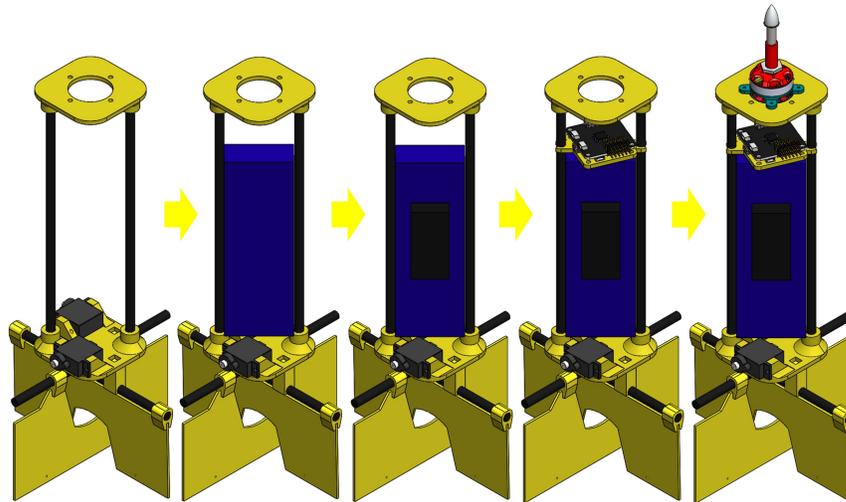


FIGURE 7.4: MUAV Avionics assembly progress.

Although the procedure could seem straightforward, the challenge is to change and test the values of the PID controller implemented in the flight computer from the factory, achieving a stable vehicle flight. Firstly, the values of the integral and derivative gains are defined as zero, this with the purpose of only tuning the proportional gain without the effect of the other gains. Thereafter, the proportional gain is increased in steps of 5% of the maximum value allowed in Librepilot, this is done until a vibrating response is reached on the vehicle; once an oscillating response is achieved, the proportional gain is reduced by 5% to avoid the generated vibrations. Secondly, the integral gain value is found, and the final value and the increment of this variable must be controlled to reduce the stationary state error visible with a proportional controller alone. This value is incremented in steps of 3% of the maximum value allowed in Librepilot, until a moderate oscillating response is obtained that does not affect the flight, in summary, it is remarkable to mention that if this value is too high can cause the crash of the vehicle. Derivative gain is settled to zero, but it is also incremented in steps of 3% of the maximum value allowed in Librepilot, this is used to eliminate all the vibrating behavior generated by the integral controller. This value is incremented until any oscillation in hover mode is not visually present, see Figure 7.5.

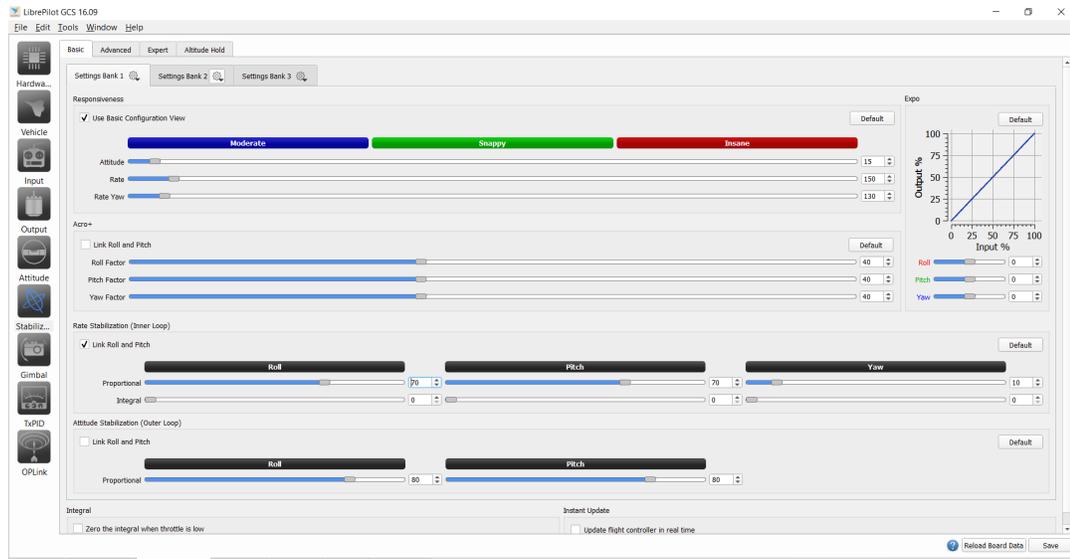


FIGURE 7.5: PID Tuning on LibrePilot software.

7.5 FLIGHT TESTING

The flight test is performed using the new optimized control surfaces, these new components reduced the total net mass of the vehicle from 0.843 kg to 0.578 kg. All flight tests are done in an outdoor environment under basic atmospheric conditions according to IEA, preserving normal values for air density, pressure, and temperature, this stage is done in a sunny day with no high-speed wind conditions, and low humidity. The test is performed with two human elements, one pilot; who assures the vehicle stabilization and correct path, and one viewer, who must be able to protect the MUAV in case of an unstable or fast landing on the floor. The maximum longitudinal/lateral speed reached by the vehicle is near of the 4 m/s, the maximum elevation speed achieved is rounding 2.5 m/s, and the autonomy of 5 minutes considering hover flight with no high-speed wind conditions and no aggressive trajectories. Including several maneuvers and displacements the endurance of the vehicle is reduced to 3.45 minutes. All testing is done including a cargo capacity mass of 0.2 kg, under these conditions and considerable displacements the vehicle is capable of flying for 3 minutes with a maximum longitudinal/lateral speed of 4 m/s and an

elevation speed of 2.5 m/s, see Figure 7.6.



FIGURE 7.6: MUAV outdoor flight test.

The new optimized control surfaces reduced considerably the spent time and used material. These two benefits can be attributed to the printing orientation; it is changed from 90° (robust ailerons) on edge to 0° flat (robust ailerons) see Figure 5.2. On both scenarios, the generated surface shows a smooth material deposition without any visible stair-stepping phenomena; thus, the roughness effects are considered neglectable in the analysis. Moreover, the post-processing of the optimized surfaces such as acetone/resin layers or heat treatment is not considered in the present document since these procedures are performed to increase the mechanical properties of the PLA, this could be a possible solution for the presented problem, however, the failure is aborded in the change of the morphology not the properties of the material.(Wach et al., 2018; Kozior et al., 2020)

CHAPTER 8

RESULTS AND DISCUSSION

8.1 TENSILE TEST RESULTS

The stress-strain plots for specimens in all building angles direction with 10% infill are depicted in Figure 8.1. The plot reveals that the edge specimens at 0° possess the highest stiffness and strength properties (37.08 MPa) in comparison to 90° specimens. The maximum strength in tensile varies from 6 MPa to 37 MPa; this specific property includes a decreasing tendency as the printing angle increases; thus, at higher printing angles the mechanical properties in the tensile direction decrease. Specimens manufactured at 30° tested fail in a brittle manner; this is caused by the poor adhesion generated in these specific specimens and the orientation between deposited layers; these specific specimens are not oriented in the same direction that the layers are stacked. Individual raster and a concentrated area carry the load during tensile testing. (Saeed et al., 2018).

It is shown that the build orientation directly impacts the mechanical properties of the specimens; although, in some cases, it could not be critical. For example, the results of the elastic modulus varied from 2.46 GPa to 3.92 GPa, these are illustrated in Table 8.1. The elastic modulus follows the same tendency as the printing angle; both properties increase. This reduction can be caused to the not homogeneous

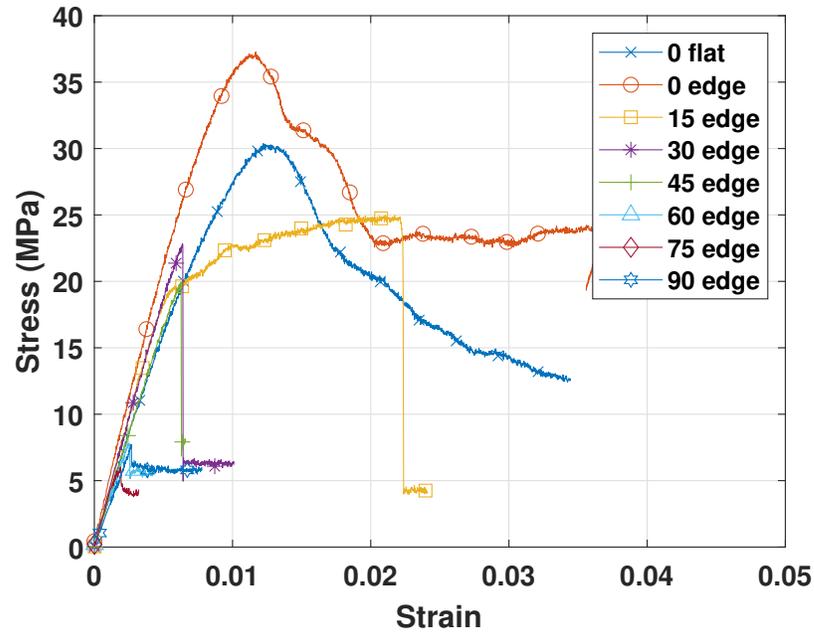


FIGURE 8.1: Stress vs strain plot for specimens with a 10% infill.

load distribution over the area of the specimen, this is caused by a deficient layer adherence during the extrusion of the material.

TABLE 8.1: Mechanical properties for specimens with 10% infill density.

10% Infill mechanical properties								
-	0°F	0 °E	15°E	30°E	45°E	60°E	75°E	90°E
Tensile strength (MPa)	30.01	37.08	24.82	22.66	19.96	7.73	6.02	7.7
Elastic modulus (GPa)	2.41	3.46	3.92	2.94	2.98	2.89	2.79	2.98
Shear modulus (GPa)	0.88	1.27	1.44	1.04	1.09	1.06	1.02	1.09

The specimens with 15% infill possess a tensile strength that varies from 7.28 MPa to 43MPa, Figure 8.2. A decreasing tendency is observed; while the printing angle is incremented this property tends to decrease. The elastic modulus is in the range from 2.08 to 3.59 GPa, this is well illustrated in Table 8.2. It is remarkable to mention that the results obtained for 15% infill specimens show a bigger tensile

strength than 10% infill results; it is observed that the maximum property is allocated at 42.83 MPa for 0°on-edge specimens. An averaged modulus of elasticity is considered to simulate FDM parts in a specific build orientation.

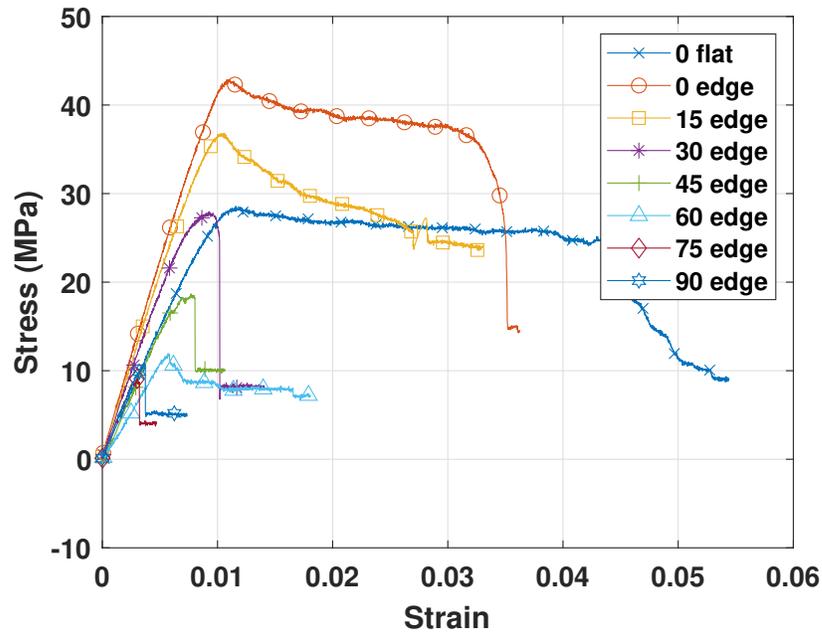


FIGURE 8.2: Stress vs strain curve of specimens with 15% infill.

TABLE 8.2: Mechanical properties for specimens with 15% infill density.

10% Infill mechanical properties								
-	0°F	0 °E	15°E	30°E	45°E	60°E	75°E	90°E
Tensile strength (MPa)	32.64	42.83	19.16	14.61	12.86	7.77	7.28	7.32
Elastic modulus (GPa)	2.43	3.59	3.51	3.38	3.06	3.17	2.08	2.78
Shear modulus (GPa)	0.89	1.31	1.29	1.24	1.12	1.16	0.76	1.02

The obtained average modulus of elasticity for specimens at 10% and 15% infill is 3.00 GPa and 3.04 GPa respectively. Ravindrababu et al. (2018) reported quite similar results, even though the author did not consider the same infill percentage, however, the elasticity modulus presented higher values, which can be attributed to

the infill percentage utilized in this publication. The influence of build angle in some situations could not be considered crucial, thus, numerical analysis to evaluate the stress state in the vehicle is fundamental before building FDM components.

8.2 CFD RESULTS

Fluid simulations are performed to predict the effect of air streams surrounding the MCR UAV V3.0. Then, a finite element study is performed to obtain the von-Mises stress and maximum deformation of the structure. Four control surface topologies are evaluated: A-concave deflected surface, B-optimized concave shape, C-convex deflected surface, and D-optimized convex deflected surface. Each topology is evaluated considering a critical deflecting angle of 25° . Fig. 8.3. shows the velocity contour for the ailerons scenarios A and B.

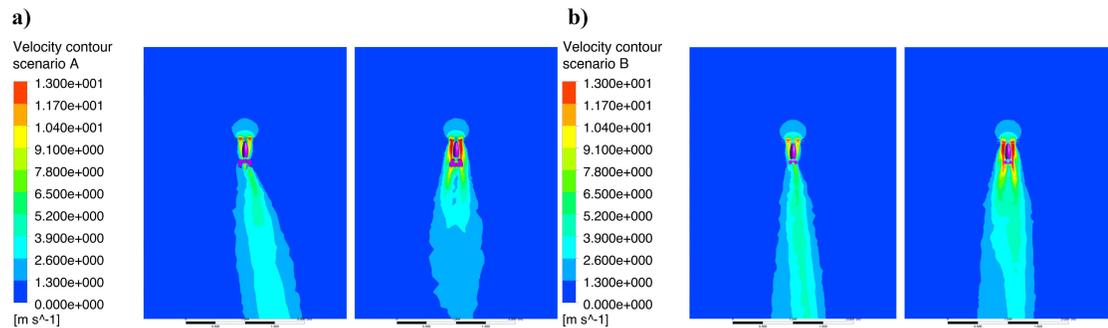


FIGURE 8.3: Velocity contour around MCR UAV V3.0. a) A-convex shape deflected surface and b) B-optimized convex shape deflected surface.

It is possible to visualize that the maximum velocity is reached at the tip of the propellers, where the red color is quite visible. For a better visual representation, the maximum velocity is limited in ANSYS software to 13 m/s, this is done since the maximum tip blade velocity is not further explored/analyzed in the presented document. The velocity contour of topologies A and B is observed in the interaction between the ailerons, fuselage and the airflow surrounding the vehicle. The contour

tool enables the prediction of flow properties, such as the pressures around the area of the control surfaces, for future static structural analysis (Seeni, 2019).

The velocity contour of Figure 8.3 a) presents a notable deviation of the air stream. The velocity magnitude behind and ahead of the deflected aileron varies from 2.6 m/s to 9.1 m/s. The air distribution around the bullet fuselage is considered homogeneous. It is observed that the air stream impacts only on the certain specific central zone of the concave not-deflected surface, this generates a bifurcation of the air streams, which is quite visible on Figure 8.3 a). Considering the reshaping of the surface; reducing the longitude (condition B), the velocity surrounding the main fuselage is also homogeneous for the topology B as illustrated in Figure 8.3 b). Additionally, the convex aileron or not-deflected surface is directly impacting the air stream from the propellers, avoiding a divergence of the stream. This result is quite suitable since it is desired to concentrate the air in the central zone of the control surface. Thus, the not-deflected surface is also modified to reduce the dimension of the surface; this modification is illustrated in scenario C shown in Figure 8.4 a). The simplification of the concave control surface of scenario D is depicted in Figure 8.4 b).

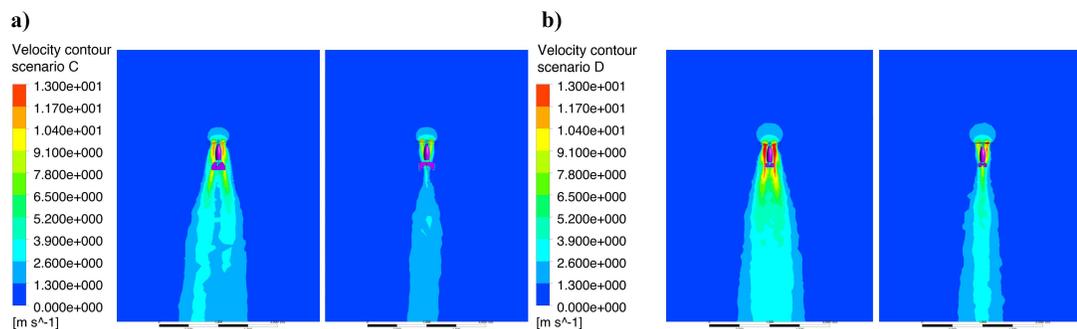


FIGURE 8.4: Velocity contour around the UAV, a) scenario C with concave deflected surface and b) scenario D with optimized concave deflected surface.

The entire area including the fuselage is impacted by a non-zero stream. The main difference between scenarios B and D (optimized surfaces) occurs on the convex surface (scenario D), where the modification of the morphology avoids the air stream

divergence; instead, two parallel air paths are visible with even higher velocities than the robust convex surface. Scenario D presents the optimized deflected concave surface. Figure 8.4 b) It is seen a quite little deviation of the air stream; however, since the size of the surface is smaller than in the C scenario, the deviation is decreased. Moreover, a higher velocity (9.1 m/s to 11.7 m/s) is obtained behind the control surface, thus generating more moments over the vehicle gravity center. After the CFD procedure is done, a structural design based on the finite element method is performed on the MCR UAV V3.0. The CFD pressure distribution results are used as a main boundary condition for the finite element approach.

8.3 FEM STRUCTURAL RESULTS

Finite element structural studies are performed considering a variation in the thickness of the walls. The wall thickness for surface A and surface C is varied from 0.4 mm to 1.2 mm, these ranges are higher since these ailerons are thicker and more robust than optimized ailerons. The thickness for surfaces B and D is varied from 0.2 mm to 1.2 mm. It is important to mention that the optimized surfaces (B and D) are printed over the flat surface of the printer (0° flat); thus, the wall thickness of the part can be as minimum as the layer thickness permitted by the printer (0.1 mm), and robust surfaces are printed on (90° edge), thus the thickness of the walls is measured by the number of perimeters and the nozzle used (0.4 mm), thus, the minimum thickness allowed is 0.4 mm considering 1 layer in the perimeter. The infill percentage is evaluated at 10% and 15%. Figure 8.5 a) presents the total deformation and Fig. 8.5 the b) von Mises stress of the control surface for surface topology A (with a thickness of 0.4 mm). The maximum deformation is obtained at the right bottom of the aileron with a maximum value of 0.79 mm.

A maximum strain of 0.012 is obtained for the condition at 0° on the edge. The von Mises stress of the control surface A is illustrated in Figure 8.5 b), with a maximum stress value of 2.74 MPa. The stress distribution is concentrated at the

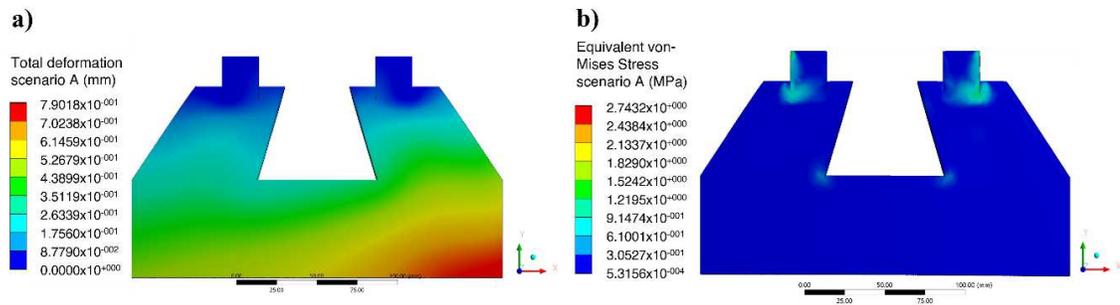


FIGURE 8.5: Finite element results of control surface of scenario A. a) Total deformation (mm) and b) von Mises stress (MPa).

sides of the connecting section between the fuselage and the control surface, this is the specific problem with a previously manufactured vehicle where cracks were allocated. (Dominguez et al., 2022)

Figure 8.6 a) illustrates the total deformation for aileron B and Figure 8.6 b) represents the von-Mises stress of topology B. The maximum deformation is concentrated at the bottom of the surface, this with a value of 0.63 mm; it is important to mention that the maximum deformation is uniformly distributed around the lower edge.

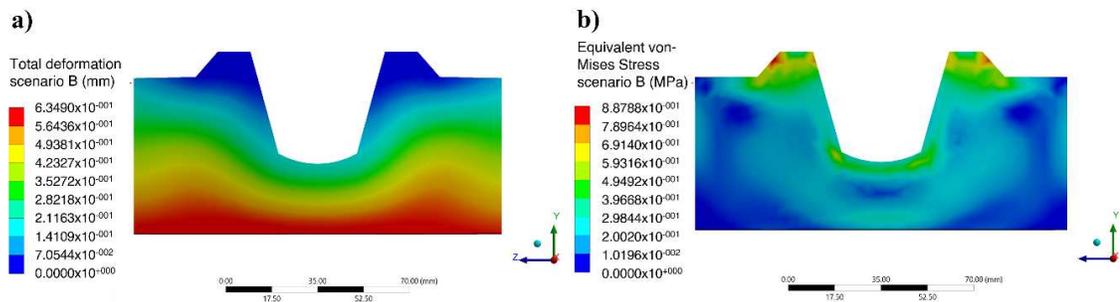


FIGURE 8.6: Finite element results of scenario B for an optimized control surface. a) Total deformation (mm) and b) von Mises stress (MPa).

In Figure 8.6 b), the von-Mises stress presents a maximum value located at the top of the surface; this is caused to the smaller size of the modified surface.

The numerical results of scenario C are illustrated in Figure 8.7 a) which presents a maximum deformation of 0.62 mm located at the bottom of the surface. This condition presented a wall thickness of 0.4 mm.

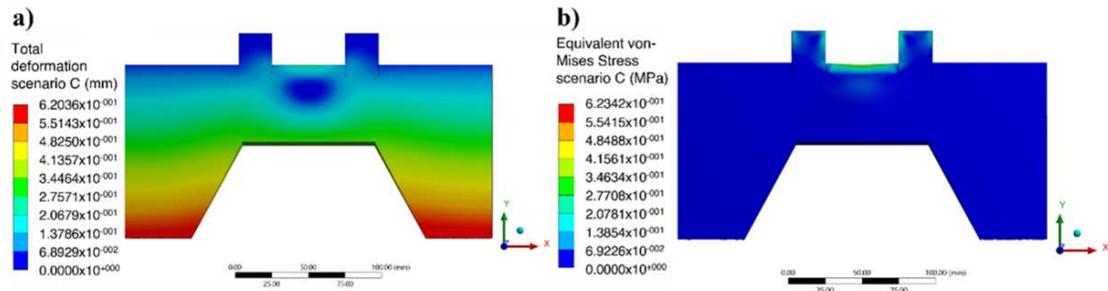


FIGURE 8.7: Finite element results of condition C with an optimized control surface. a) Total deformation (mm) and b) von Mises stress (MPa).

Figure 8.7 b) presents the von Mises stress of topology C with a maximum value of 6.23 MPa. The maximum value is located at the support section, where moments generate from weight and forces acting on the surface.

Optimization of the convex control surface (topology D) is illustrated in Figure 8.8. Figure 8.8 a) depicts a maximum deformation of 0.25 mm (thickness of 0.2 mm), this is concentrated in the center of the aileron. This value decreased in comparison with scenario C, this phenomenon can be attributed to the morphology of the aileron, which is smaller. According to strain-stress curves reported in Figure 8.2 the maximum deflection does not surpass the plastic region. Figure 8.8 b) shows the von-Mises stress distribution of the control surface, unlike scenario C, the stress distribution is homogeneous and is not concentrated in the link section.

The global results for scenarios A to D are plotted in Figure 8.9. It is easy to visualize that the maximum deformation obtained decreases the same as the thickness of the walls increases; however, the stress maximum value does not vary considerably for the optimized surface (B and D) in comparison with the robust.

Table 8.3 presents the maximum and minimum data for the stress and de-

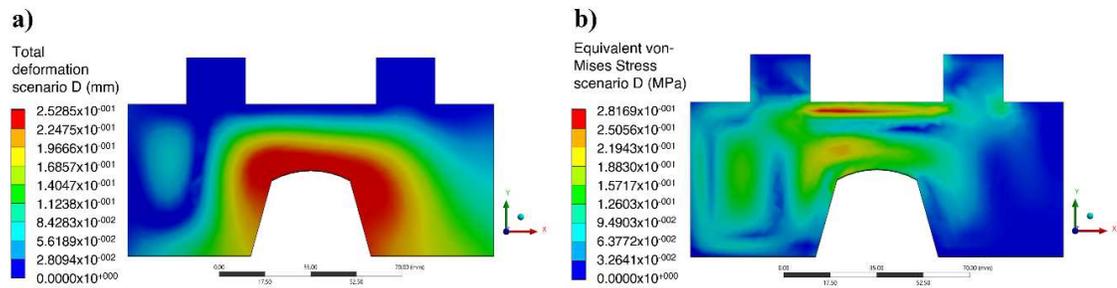


FIGURE 8.8: Finite element results of condition D with an optimized control surface.

a) Total deformation (mm) and b) von Mises stress (MPa).

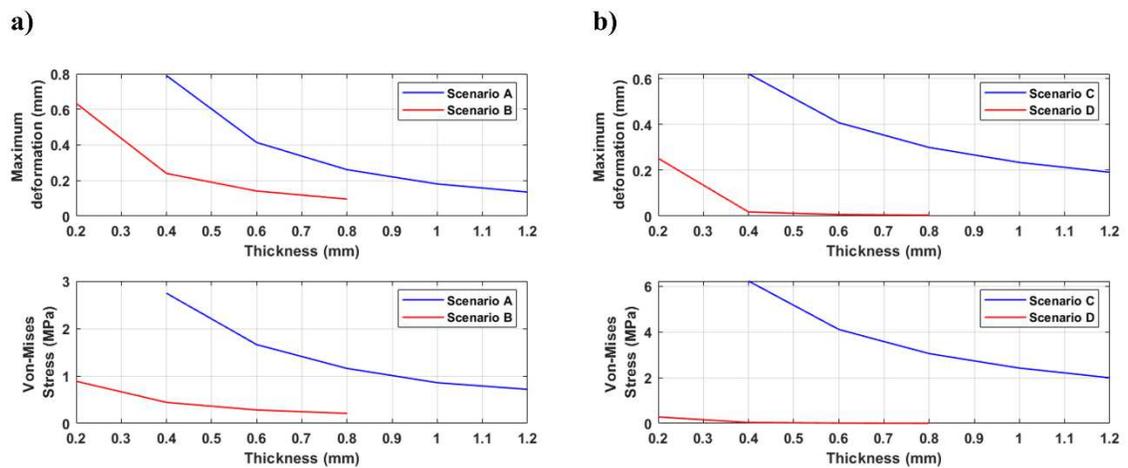


FIGURE 8.9: Numerical results of control surfaces. a) case A and B, b) case C and D.

formation for each surface topology. The best-obtained parameters are those with optimized morphology, this can be accredited to the better stress distribution obtained over the control surface area. These specific results easily contrast with the obtained results for A and C scenarios, where concentrated stress is observed, at the linking region between the bullet fuselage and the control surfaces. Moreover, the stress distribution in the optimized ailerons is significantly reduced; this can be attributed to the smaller size of the control surfaces, these specific scenarios present less air stream, generating lower stress conditions.

TABLE 8.3: Stress and deformation results for control surfaces; A, B, C and D.

Minimum, maximum deformation and stress			
Control surface	Wall thickness (mm)	Deformation (mm)	Stress (MPa)
A	0.4	0.79	2.73
A	1.2	0.13	0.72
B	0.4	0.62	6.23
B	1.2	0.19	1.99
C	0.2	0.63	0.88
C	0.8	0.09	0.21
D	0.2	0.17	0.29
D	0.8	0.01	0.03

The infill percentages selected for the structural evaluation of von-Mises stress and total deformation are considered as the most critical for the UAVs with low-speed applications (Cruzatty et al., 2022). The control surfaces are analyzed using FDM slicer to quantify the net impact area and the total mass considering support elements if these are required. All these parameters are represented in Table 8.4

TABLE 8.4: Output parameters for surface topologies; A, B, C and D.

FDM Output Parameters			
Control surface	Area (m ²)	Mass (Kg)	FDM Time (hrs.)
A	0.0228	0.054	23.10
B	0.0080	0.016	1.90
C	0.0231	0.056	23.10
D	0.0075	0.015	1.80

It is noticeable a considerable reduction of impact area for optimized surfaces B and D. A mass reduction of around 70% is attained for both optimized ailerons.

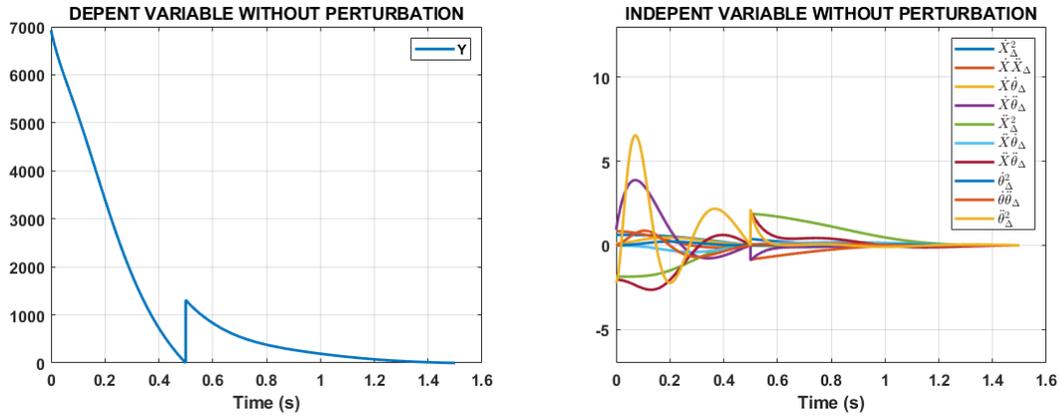
The low-cost prototyping methodology permits to modify the final FDM building direction: from 90° on edge to 0° flat, this reduces the manufacturing time in both ailerons. The approach presented, involves an FDM strategy to quantify the effect of building orientation and minimum printing thickness, to reduce the prototyping cost in the first stage of the structural design. Moreover, the wall thickness selection is vital for the buckling evaluation of the MCR UAV V3.0 under bending conditions.

8.4 OPTIMAL ADAPTIVE IRL RESULTS

The optimal adaptive IRL algorithm is mathematically presented in chapter 4 and the algorithm simulation on Matlab software is described in chapter 6, it was mentioned that the control scheme is simulated considering output disturbances, however, it is also necessary to simulate the scheme with no-disturbances. This with the purpose of quantifying the effects of the disturbances in the algorithm, thus, this section explains the results obtained for the algorithm described in section 6.3. All the simulations were carried out employing an initial state of 1 meter in the longitudinal axis X and 0.1 radians ($\approx 6deg$) in the pitch θ euler angle.

As it was mentioned previously in section 6.3, the algorithm tries to minimize the error in the least square sense using an independent and a dependent variable. These must be minimized in time, this means that the algorithm is working correctly, the representation of these data is illustrated in Fig. 8.10

Figure 8.10a illustrates the Y variable, this acts as dependent variable since it depends on the cost function Y which is intrinsically dependent of the system output y and systems states x . Figure 8.10b shows the independent variable, these are the states described in the equation (4.23). It is quite simple to visualize that the reduction the variables is effective in both scenarios, these represent the error respect to the past value of the same variable, thus, the controller is successfully reducing the error in the least square sense.



(a) Dependent variable.

(b) Independent variable.

FIGURE 8.10: Least squares variables for no disturbance scenario.

The states of the system x are quite important, these are the proof that the algorithm correctly works. Figure 8.11 shows the states obtained from the algorithm without any consideration of disturbances, it is possible to visualize that the most affected variable is the angular velocity p since it tends to oscillate, however, the position X and pitch angle θ returns to equilibrium from the initial value designated [10.1]. In the results the settling time corresponds to 1.35 seconds to the position and 1.67 to the angle, it is remarkable to mention that the angular velocity obtained seem to be elevated in comparison with the other variables, nevertheless, the units of this are different and the obtained results are inside of the normal expected range.

The control policies K obtained from the algorithm without some disturbance present are described in Table 8.5,

8.4.1 IRL CONSIDERING OUTPUT DISTURBANCE RESULTS

The simulation without any consideration of disturbance obtained accurate results for the stabilization of the vehicle from a non-zero initial position [10.1] with low settlement time, however, this scenario for a real implementation does not represent

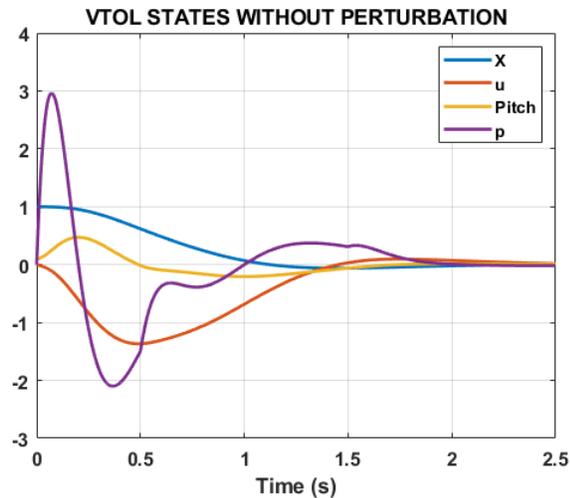


FIGURE 8.11: States obtained for no disturbance scenario.

TABLE 8.5: Obtained controllers for undisturbed simulation.

No-Disturbance Iterated Control Policy				
Iteration	K_{11}	K_{12}	K_{13}	K_{14}
1	-122.45	-70.48	201.24	20.06
2	-125.48	-61.46	172.73	18.07
3	-121.96	-72.30	204.56	20.33

the real phenomena, thus, the inclusion of output disturbances is required for the validation of the algorithm, moreover, the quantification of the operation range is also required for correct algorithm validation.

The range of characterization for the algorithm was defined from 75Hz to 200Hz as is presented in Fig. 6.6, the inclusion to the algorithm required a modification of the state-space model and control input, these changes are depicted in Figure 4.6.

The dependent and independent variables used for the least squares iteration are depicted in Figure 8.12, notice that the Y variable decreases fastly, however, it did slower than the not disturbed scenario, this is a consequence that the error

asymptotically tends to zero slower than other simulation Figure 8.12a. Moreover, the x states are affected since the response includes a peak signal, this effect also can be attributed directly to the perturbation since the states take more time to tend zero Figure 8.12b.

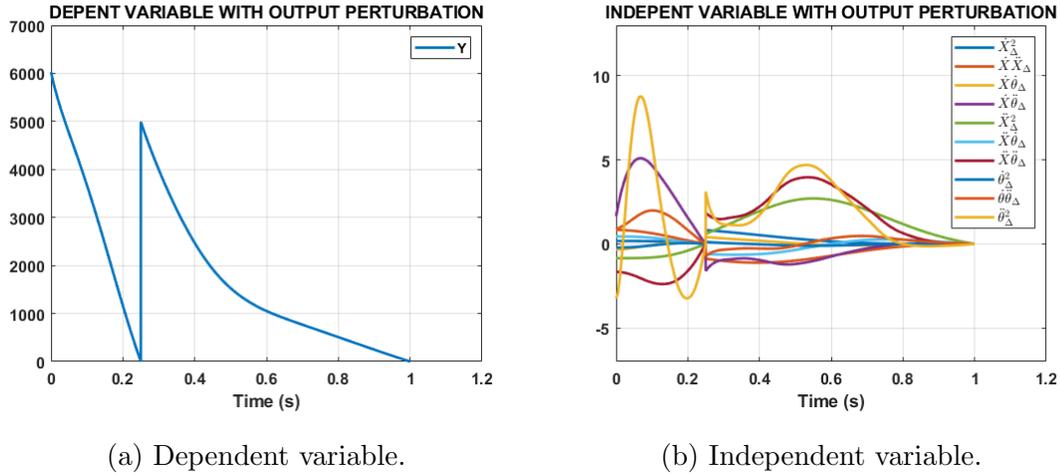


FIGURE 8.12: Least squares variables for 75Hz disturbance scenario.

The states response obtained was quite different in comparison to the not disturbed scenario, this includes a remarkable oscillation in all the states and a peak response of 3.42 for the angular velocity p , and the settlement time was increased to 5.14 seconds for the position of the vehicle X and to 5.96 seconds for the pitch angle θ . This can be attributed that the last controller K obtained values were substantially reduced respect to the original values, thus, this generated an oscillated response in all the states. Nevertheless, the reduction was not affected in the angular velocity gain preserving similar value with respect to the initial controller, this lack of variance generated a noisy response for this variable Fig. 8.13.

The control policies K obtained from the algorithm with 75Hz disturbance present are described in Table 8.6.

The scenario with a disturbance with a frequency of 100Hz was simulated, the results of the least square variables are depicted in Figure 8.14. The comparison in the response of independent and dependent variables with respect to the scenario of

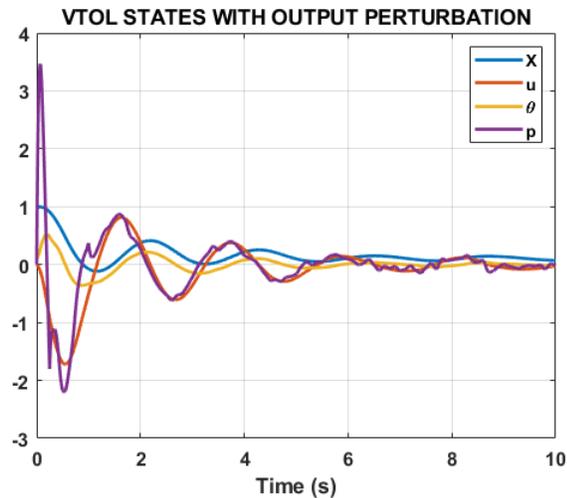


FIGURE 8.13: States obtained for 75Hz disturbance scenario.

TABLE 8.6: Obtained controllers for 75Hz disturbance simulation.

75Hz Disturbance Iterated Control Policy				
Iteration	K_{11}	K_{12}	K_{13}	K_{14}
1	-141.42	-80.61	225.40	21.23
2	-153.74	-70.47	181.88	24.67
3	-2.16	-19.20	27.46	21.71

75Hz is neglectable, a quite similar response was obtained; it is important to mention that all the variables and stated tended to zero successfully.

The states x obtained from the 100Hz disturbance scenario simulation changed with respect to the last presented simulation, firstly, the settlement time was reduced. The position X resulted in 4.24 seconds and 4.37 seconds for the pitch angle θ , moreover, the fast oscillating response obtained at the steady state for the angular velocity p was considerably reduced; this can be observed in Figure 8.15.

The control policies K obtained from the algorithm with 100Hz disturbance present are described in Table 8.7.

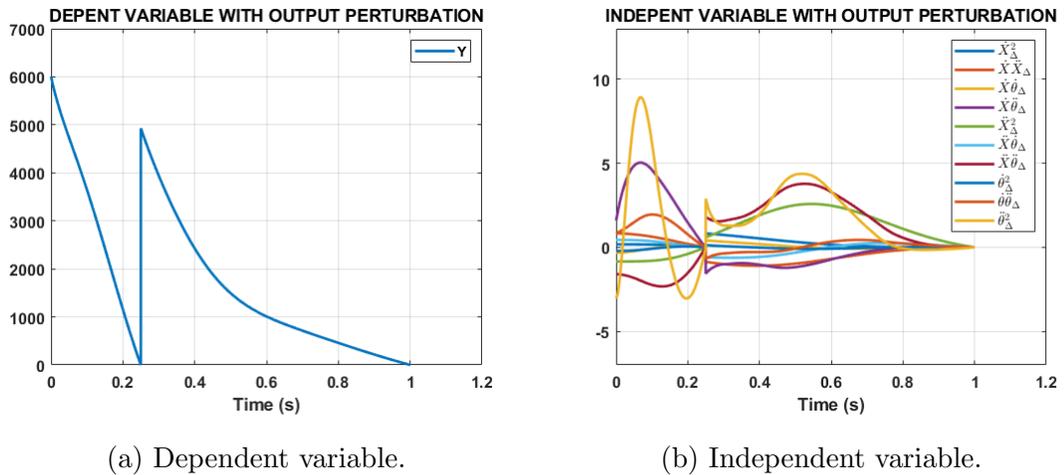


FIGURE 8.14: Least squares variables for 100Hz disturbance scenario.

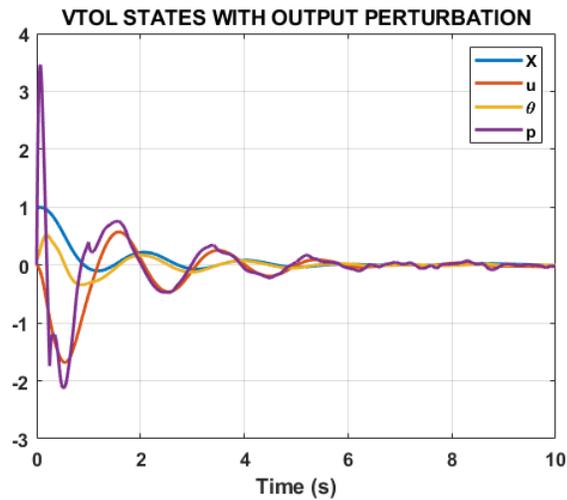


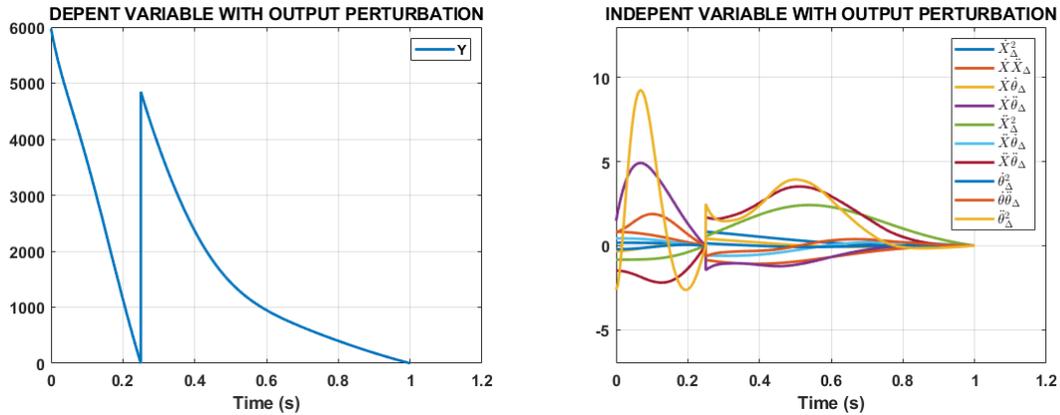
FIGURE 8.15: States obtained for 100Hz disturbance scenario.

A disturbance with a frequency of 200Hz was also simulated as the final characterization of the disturbances, the results of the least square variables are depicted in Figure 8.16. The comparison in the response of independent and dependent variables with respect to the scenario of 75Hz and 100Hz is neglectable, a quite similar response was obtained; then, the responses followed the same tendency to reach zero error in a limited time.

The states x obtained from the 200Hz disturbance scenario simulation changed

TABLE 8.7: Obtained controllers for 100Hz disturbance simulation.

100Hz Disturbance Iterated Control Policy				
Iteration	K_{11}	K_{12}	K_{13}	K_{14}
1	-141.42	-80.61	225.40	21.23
2	-147.80	-68.61	179.29	22.78
3	-14.96	-22.29	43.72	20.27



(a) Dependent variable.

(b) Independent variable.

FIGURE 8.16: Least squares variables for 200Hz disturbance scenario.

abruptly with respect to 75Hz and 100Hz simulation, firstly, the settlement time was reduced, and the position X resulted in 2.31 seconds and 2.43 seconds for the pitch angle θ . Moreover, for all the states the oscillating response was near deducted to zero, obtaining a response similar to the scenario with no-disturbances, this can be observed in Figure 8.17.

The control policies K obtained from the algorithm with 200Hz disturbance present are described in Table 8.8.

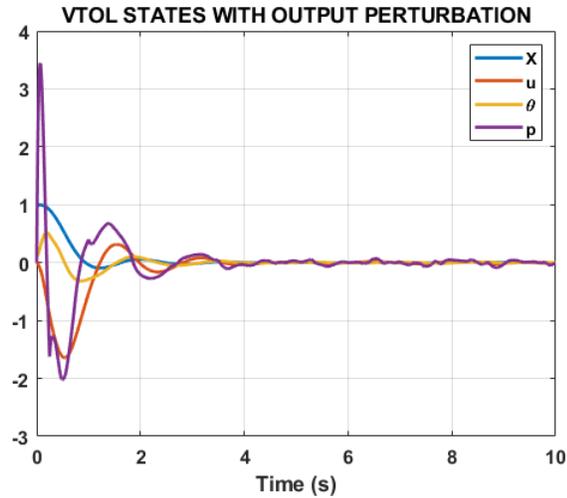


FIGURE 8.17: States obtained for 200Hz disturbance scenario.

TABLE 8.8: Obtained controllers for 200Hz disturbance simulation.

200Hz Disturbance Iterated Control Policy				
Iteration	K_{11}	K_{12}	K_{13}	K_{14}
1	-141.42	-80.61	225.40	21.23
2	-139.52	-64.65	179.01	19.76
3	-43.71	-30.45	74.37	17.19

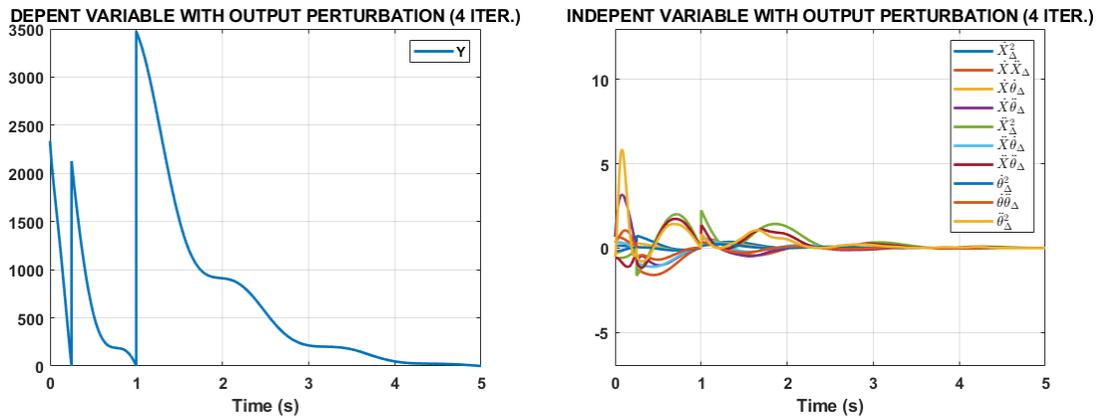
8.4.2 IRL CONSIDERING OUTPUT PERTURBATION (4 ITERS.)

RESULTS

The simulations presented in the last section included three iterations for the generation of a new control policies K successfully stabilizing the system and reaching asymptotically the zero states, nevertheless, the presented algorithm is adaptive. This type of control scheme requires enough signal excitation to work properly. In this section the results of the evaluation of an additional calculation of the controller are presented, this involves to develop the calculation employing the already stabilized signals near to the zero position without any considerable excitation.

The scenario with a disturbance of 75Hz obtained an unstable response, where all the states tend to the infinite and in consequence; the variables used to perform least square computation never tend to zero. The simulation with a stable response was 100Hz onward up to 150Hz. Firstly, the dependent variable shown a considerable increment with respect to the previous simulation, this phenomenon can be attributed to the lack of signal excitation. However, the variable tends to zero after a considerable time (4 seconds), it was required to take more time-steps since the error was inherently near to zero thanks to the stabilization of the states, the result is shown in Figure 8.18a.

In comparison with the three iterations simulations presented in the last section, the independent variable related to the system states presents near of half in magnitude with respect to the maximum point. Considering more time-steps and a more stabilized response, the difference between the previous signal is lower and as a result the plot presented lower values, Figure 8.18b.



(a) Dependent variable.

(b) Independent variable.

FIGURE 8.18: Least squares variables for 100Hz disturbance scenario.

In comparison with the three iterations considering the same disturbance frequency illustrated in Figure 8.15. This simulation presents a more damped response with fewer oscillations for the velocity variables $[u p]$. Nevertheless, the settlement time of both outputs; position X and pitch angle θ presents a variance lower than

the 2% respect to the three iteration simulation. Moreover, at the steady state of angular velocity p a fast oscillating response was obtained, this can be attributed to the last high gain values obtained for the last iteration, these results are depicted in Figure 8.19.

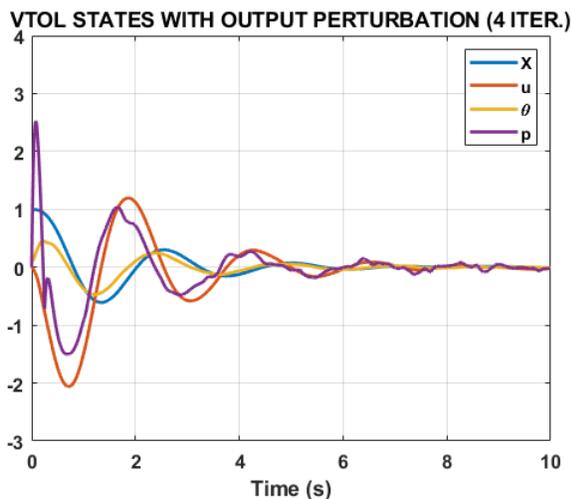


FIGURE 8.19: States obtained for 100Hz disturbance scenario.

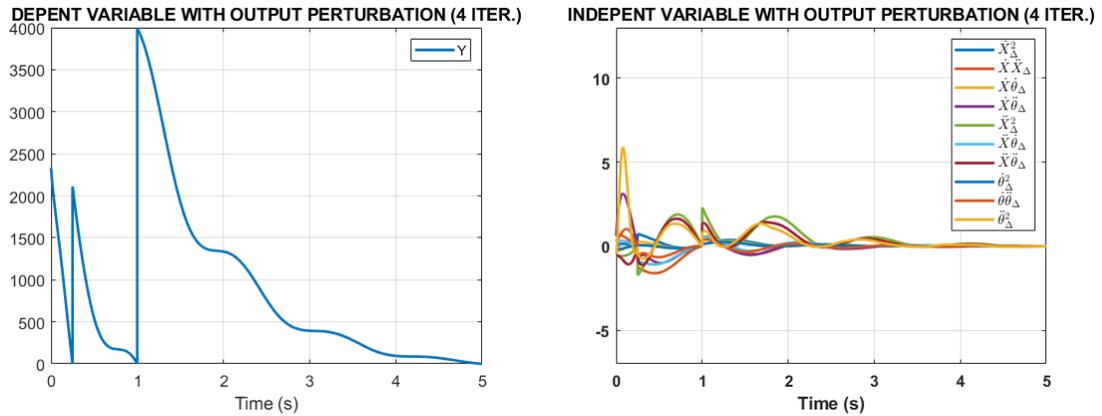
The control policies K obtained from the four iterations algorithm with 100Hz disturbance present are described in Table 8.9.

TABLE 8.9: Obtained controllers for 4 iterations 100Hz disturbance simulation.

100Hz Disturbance Iterated Control Policy				
Iteration	K_{11}	K_{12}	K_{13}	K_{14}
1	-89.44	-53.67	158.01	17.77
2	-97.91	-25.07	149.66	21.77
3	-135.87	-39.77	224.35	26.96
4	-887.40	-466.50	1626.80	370.20

Lastly, the last stable scenario considering four controller iterations was employing a 150Hz disturbance gain, the dependent variable presented a similar behavior to the previous simulation, however, the magnitude of the maximum peak

was increased by 500 units but following a decreasing tendency to zero, this result is depicted in Figure 8.20a. For the independent variable, the behavior was quite similar to the 100Hz simulation without any considerable change. Figure 8.20b



(a) Dependent variable.

(b) Independent variable.

FIGURE 8.20: Least squares variables for 150Hz disturbance scenario.

The states obtained were more oscillating than the 100Hz disturbance response, the settlement time for the position X and pitch angle θ was increased, for both states the result was 5.92 seconds. These results can be attributed to the final high gains obtained in the fourth iteration, it is important to remark that this frequency was the maximum capacity to be handled by the algorithm.

The control policies K obtained from the four iterations algorithm with 150Hz disturbance present are described in Table 8.10,

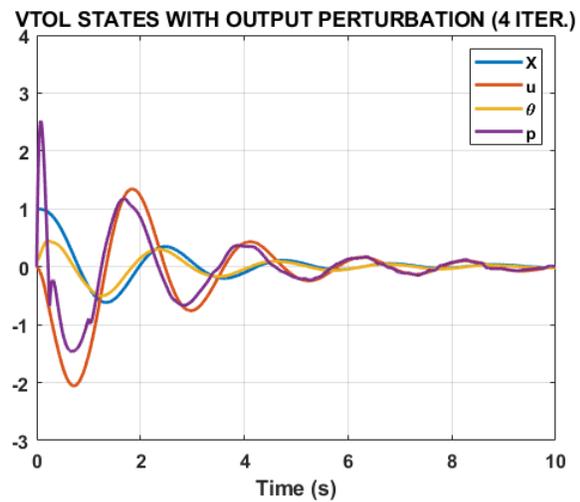


FIGURE 8.21: States obtained for 150Hz disturbance scenario.

TABLE 8.10: Obtained controllers for 4 iterations 150Hz disturbance simulation.

150Hz Disturbance Iterated Control Policy				
Iteration	K_{11}	K_{12}	K_{13}	K_{14}
1	-89.44	-53.67	158.01	17.77
2	-91.63	-21.97	145.78	19.41
3	-108.96	-35.37	170.08	28.36
4	-166.58	-364.15	437.82	354.04

CHAPTER 9

CONCLUSIONS

The design, development, modeling, computation, and disturbance characterization of an optimal adaptive IRL control scheme were presented in this work. The conventional modeling of the MCR UAV V3.0 in state-space included a full-state observer with disturbances. Moreover, the IRL algorithm coding was explained and posted in the present work. The principal findings are pointed out:

- IRL algorithm successfully stabilizes the vehicle parting from a random initial condition for the most important variables; position and inclination angle. This value can be changed; however, the settlement time is affected until it is enough large to be undesired for aeronautical purposes.
- The mathematical model proposed in the present document allows the generation of convergence between the observer states and real vehicle dynamics states considering a simple proportional gain.
- IRL algorithm is suitable to iterate a maximum of three times to obtain a new control policy, more iterations could generate an unstable or gains with high values that can produce a fast oscillating response.
- The frequency range of output disturbances, that the algorithm can handle without being unstable and generate suitable settlement time for aeronautical

purposes, can vary from 75Hz to 200Hz considering a procedure where the control policy is only calculated/updated three times.

- The frequency range of output disturbances, that the algorithm can handle without being unstable and generate suitable settlement time for aeronautical purposes, can vary from 100Hz to 150Hz considering a procedure where the control policy is calculated/updated four times.

The FDM manufacture and numerical CFD/FEA procedure of MCR UAV V3.0 with a coaxial counter-rotating configuration were developed in this work. Optimization of two different surface topologies was evaluated: concave deflected surface, and convex deflected surface, including two more optimized surfaces. The principal findings are pointed out:

- Infill densities of 10% and 15% and building angles from 0° to 90° were parameters for specimens to mechanical characterization. The PLA showed an ultimate tensile strength of 37 MPa considering 10% infill and 43 MPa for 15% infill. The modulus of elasticity average for specimens at 10% was 3.00 GPa and 3.04 GPa for 15%.
- The most critical zone of the control surface was located at the superior zone of the ailerons since the air streams impact this specific zone of the component. Scenario B shows a simplification of the concave control surface, for this procedure the maximum deformation obtained was 0.17 mm considering a thickness of 0.2 mm. This value decreased compared with the robust component presented in scenario A which presented a deformation of 0.62 mm considering a thickness of 0.4 mm.
- The results obtained for A to D scenarios show that the total deformation follows a decreasing tendency as the thickness is increased. For the surface described in scenario A, the maximum deformation obtained was 0.62 mm considering a thickness 0.4 mm, and for the component of the scenario D, the

value decreased to 0.25 mm considering a thickness of 0.2 mm; nevertheless, the stress magnitude does not vary considerably for each condition.

- The von-Mises stress around both newly optimized ailerons was distributed homogeneously over the area, to avoid stress concentrations. These results were carried out through morphology modification of edges and connecting components of the vehicle.
- The net resulting weight was reduced to near of 70% for both new optimized ailerons. The total weight of the MCR UAV V3.0 was reduced by 32% based on a previously manufactured prototype (Dominguez et al., 2022).
- The presented work successfully introduces a FDM methodology to evaluate the effect of building orientation and printing thickness of FDM manufactured components, with the aim to reduce the prototyping material cost and design utilized resources.

APPENDIX A

IRL MATLAB PROGRAMMING

```
%ONLINE IRL of a QUADROTOR considering only DoF axis (X PHI)
clc
clear all
g = 9.81;
%-----STATE SPACE-----
A = [0 1 0 0;0 0 -g 0;0 0 0 1;0 0 0 0]; %[x u phi p]
B = [0 0 0 1]'; %Mx
C = [1 0 0 0;0 0 1 0]; %[x;phi]
D = [0 0]';
G = ss(A,B,C,D);
co = ctrb(G);
controllability = rank(co);
Q = C'*C*7000;
R = 1;
P = care(A,B,Q,R);
p0 = [P(1,1) P(1,2)*2 P(1,3)*2 P(1,4)*2 P(2,2)...
      P(2,3)*2 P(2,4)*2 P(3,3) P(3,4)*2 P(4,4)]';
K = lqr(A,B,Q,R)
%REFERNCE GAIN
```

```

Cn = [10 0 10 0];
sys_ss = ss(A,B,Cn,0);
Nbar = rscale(sys_ss,K);
GC = ss(A-B*K,B*Nbar,C,D);
%OBSERVER
ob = obsv(G);
observability = rank(ob);
poles = eig(A-B*K);
P = [-22 -23 -24 -25];
L = place(A',C',P)';
A_0 = [(A-B*K) (B*K);
        zeros(size(A)) (A-L*C)];
B_0 = [B*Nbar;
        zeros(size(B))];
C_0 = [C zeros(size(C))];
D_0 = [0;0];
GC_0 = ss(A_0,B_0,C_0,D_0);
%CLOSED LOOP SIMULATION
t = 0:0.005:.5;
r = .4 + 0*ones(size(t));
xo = [1 0 0.1 0];
[y,t,XX] = lsim(GC_0,r,t,[xo 0 0 0 0]);
X = [XX(:,1) XX(:,2) XX(:,3) XX(:,4)];
%-----FIRST ITERATION USING K INITIAL-----
%CALCULATION u
u = (- K*X')';
n = length(t);
% CALCULATION V
for i = 1:n
Vdot(i) = X(i,:)*Q*X(i,:) + u(i,:)'*R*u(i,:);

```

```
end
Vdot = Vdot';
V = cumtrapz(t,Vdot); %Integer Vdot
T = n;
%CALCULATION Y
for k = 1:n
Y(k) = V(T,1)-V(k,1);
end
Y = Y';
%CALCULATION x
for k = 1:n
x1(k) = X(k,1)^2-X(T,1)^2;
end
for k = 1:n
x2(k) = X(k,1)*X(k,2)-X(T,1)*X(T,2);
end
for k = 1:n
x3(k) = X(k,1)*X(k,3)-X(T,1)*X(T,3);
end
for k = 1:n
x4(k) = X(k,1)*X(k,4)-X(T,1)*X(T,4);
end
for k = 1:n
x5(k) = X(k,2)^2-X(T,2)^2;
end
for k = 1:n
x6(k) = X(k,2)*X(k,3)-X(T,2)*X(T,3);
end
for k = 1:n
x7(k) = X(k,2)*X(k,4)-X(T,2)*X(T,4);
```

```

end
for k = 1:n
x8(k) = X(k,3)^2-X(T,3)^2;
end
for k = 1:n
x9(k) = X(k,3)*X(k,4)-X(T,3)*X(T,4);
end
for k = 1:n
x10(k) = X(k,4)^2-X(T,4)^2;
end
x = vertcat(x1,x2,x3,x4,x5,x6,x7,x8,x9,x10);
%CALCULATION p
p = inv(x*x')*x*Y; %LEAST SQUARES
%CALCULATION P
P = [p(1,1) p(2,1)/2 p(3,1)/2 p(4,1)/2;
      p(2,1)/2 p(5,1) p(6,1)/2 p(7,1)/2;
      p(3,1)/2 p(6,1)/2 p(8,1) p(9,1)/2;
      p(4,1)/2 p(7,1)/2 p(9,1)/2 p(10,1)];
K = inv(R)*B'*P
%CLOSED LOOP SIMULATION
Cn = [1 1 1 1];
sys_ss = ss(A,B,Cn,0);
Nbar = rscale(sys_ss,K);
GC = ss(A-B*K,B*Nbar,C,D);
%Calculate controlled-observed matrices
A_0 = [(A-B*K) (B*K);
        zeros(size(A)) (A-L*C)];
B_0 = [B*Nbar;
        zeros(size(B))];
C_0 = [C zeros(size(C))];

```

```

D_0 = [0;0];
GC_0 = ss(A_0,B_0,C_0,D_0);
t2 = 0.5:0.005:1.5;
r2 = 0*ones(size(t2));
  xo = [X(end,1) X(end,2) X(end,3) X(end,4)];
[y2,t2,XX2] = lsim(GC_0,r2,t2,[xo 0 0 0 0]);
X2 = [XX2(:,1) XX2(:,2) XX2(:,3) XX2(:,4)];
%-----SECOND ITERATION USING K OBTAINED-----
u2 = (- K*X2')';
n = length(t2);
for i = 1:n
V2dot(i) = X2(i,:)*Q*X2(i,:)' + u2(i,:)'*R*u2(i,:);
end
V2dot = V2dot';
V2 = cumtrapz(t2,V2dot); %INTEGRAL Vdot
T = n;
for k = 1:n
Y2(k) = V2(T,1)-V2(k,1);
end
Y2 = Y2';
for k = 1:n
x1(k) = X2(k,1)^2-X2(T,1)^2;
end
for k = 1:n
x2(k) = X2(k,1)*X2(k,2)-X2(T,1)*X2(T,2);
end
for k = 1:n
x3(k) = X2(k,1)*X2(k,3)-X2(T,1)*X2(T,3);
end
for k = 1:n

```

```

x4(k) = X2(k,1)*X2(k,4)-X2(T,1)*X2(T,4);
end
for k = 1:n
x5(k) = X2(k,2)^2-X2(T,2)^2;
end
for k = 1:n
x6(k) = X2(k,2)*X2(k,3)-X2(T,2)*X2(T,3);
end
for k = 1:n
x7(k) = X2(k,2)*X2(k,4)-X2(T,2)*X2(T,4);
end
for k = 1:n
x8(k) = X2(k,3)^2-X2(T,3)^2;
end
for k = 1:n
x9(k) = X2(k,3)*X2(k,4)-X2(T,3)*X2(T,4);
end
for k = 1:n
x10(k) = X2(k,4)^2-X2(T,4)^2;
end
x = vertcat(x1,x2,x3,x4,x5,x6,x7,x8,x9,x10);
p = inv(x*x')*x*Y2;
P = [p(1,1) p(2,1)/2 p(3,1)/2 p(4,1)/2;
      p(2,1)/2 p(5,1) p(6,1)/2 p(7,1)/2;
      p(3,1)/2 p(6,1)/2 p(8,1) p(9,1)/2;
      p(4,1)/2 p(7,1)/2 p(9,1)/2 p(10,1)];
K = 1*B'*P
Cn = [1 0 0 0];
sys_ss = ss(A,B,Cn,0);
Nbar = rscale(sys_ss,K);

```

```
GC = ss(A-B*K,B*Nbar,C,D);
A_0 = [(A-B*K) (B*K);
        zeros(size(A)) (A-L*C)];
B_0 = [B*Nbar;
        zeros(size(B))];
C_0 = [C zeros(size(C))];
D_0 = [0;0];
GC_0 = ss(A_0,B_0,C_0,D_0);
t3 = 1.5:0.005:2.5;
r3 = 0*ones(size(t3));
xo = [X2(end,1) X2(end,2) X2(end,3) X2(end,4)];
[y3,t3,XX3] = lsim(GC_0,r3,t3,[xo 0 0 0 0]);
X3 = [XX3(:,1) XX3(:,2) XX3(:,3) XX3(:,4)];
times = vertcat(t,t2,t3);
states = vertcat(X,X2,X3);
outputs = vertcat(y,y2,y3);
plot(times,states)
title('VTOL STATES WITH INPUT PERTURBATION')
legend('X','u','Pitch','p')    %[x u phi p]
grid on
xlabel('Time (s)')
%Matlab Exportation
output1 = timeseries(outputs(:,1),times);
output2 = timeseries(outputs(:,2),times);
```

APPENDIX B

DISTURBED IRL MATLAB PROGRAMMING

```
clc
clear all
g = 9.81;
%-----STATE SPACE-----
A = [0 1 0 0;0 0 -g 0;0 0 0 1;0 0 0 0]; %[x u phi p]
B = [0 0 0 1]'; %Mx
C = [1 0 0 0;0 0 1 0]; %[x;phi]
D = [0 0]';
G = ss(A,B,C,D);
%DEFINE INITIAL CONTROLLER
Q = C'*C*20000;
R = 1;
K = lqr(A,B,Q,R)
%REFERENCE-GAIN
Cn = [1 0 1 0];
sys_ss = ss(A,B,Cn,0);
Nbar = rscale(sys_ss,K);
```

```

%PERTURBTIONS

Perturbations = importdata('Perturbations.mat');
P_OUT = Perturbations.PERT_200';
F = [1 0 1 0]'; %[x u phi p]

%OBSERVER

poles = eig(A-B*K); %-2.22+/-2.22J
P = [-22 -23 -24 -25];
L = place(A',C',P)';
A_0 = [(A-B*K) (B*K);
        zeros(size(A)) (A-L*C)];
B_0 = [B*Nbar -B*K*F;
        zeros(size(B)) -L*C*F];
C_0 = [C zeros(size(C))];
D_0 = 0;
GC_0 = ss(A_0,B_0,C_0,D_0);

%CLOSED LOOP SIMULATION

t = 0:0.005:.25;
r = 0*ones(size(t));
xo = [1 0 .1 0];
[y,t,XX] = lsim(GC_0,[r; P_OUT(1,1:51)],t,[xo 0 0 0 0]);
X = [XX(:,1) XX(:,2) XX(:,3) XX(:,4)];
y = y + (C*F*P_OUT(1,1:51))';

%-----FIRST ITERATION-----
u = (- K*X')' - (K*F*P_OUT(1,1:51))';
n = length(t);
for i = 1:n
Vdot(i) = X(i,:)*Q*X(i,:)' + u(i,:)'*R*u(i,:);
end
Vdot = Vdot';
V = cumtrapz(t,Vdot);

```

```
T = n;
for k = 1:n
Y(k) = V(T,1)-V(k,1);
end
Y = Y';
for k = 1:n
x1(k) = X(k,1)^2-X(T,1)^2;
end
for k = 1:n
x2(k) = X(k,1)*X(k,2)-X(T,1)*X(T,2);
end
for k = 1:n
x3(k) = X(k,1)*X(k,3)-X(T,1)*X(T,3);
end
for k = 1:n
x4(k) = X(k,1)*X(k,4)-X(T,1)*X(T,4);
end
for k = 1:n
x5(k) = X(k,2)^2-X(T,2)^2;
end
for k = 1:n
x6(k) = X(k,2)*X(k,3)-X(T,2)*X(T,3);
end
for k = 1:n
x7(k) = X(k,2)*X(k,4)-X(T,2)*X(T,4);
end
for k = 1:n
x8(k) = X(k,3)^2-X(T,3)^2;
end
for k = 1:n
```

```

x9(k) = X(k,3)*X(k,4)-X(T,3)*X(T,4);
end
for k = 1:n
x10(k) = X(k,4)^2-X(T,4)^2;
end
x = vertcat(x1,x2,x3,x4,x5,x6,x7,x8,x9,x10);
p = pinv(x*x')*x*Y;
P = [p(1,1) p(2,1)/2 p(3,1)/2 p(4,1)/2;
      p(2,1)/2 p(5,1) p(6,1)/2 p(7,1)/2;
      p(3,1)/2 p(6,1)/2 p(8,1) p(9,1)/2;
      p(4,1)/2 p(7,1)/2 p(9,1)/2 p(10,1)];
K = 1*B'*P
%REFERENCE-GAIN
Nbar = rscale(sys_ss,K);
%OBSERVER
poles = eig(A-B*K);
P = [-22 -23 -24 -25];
L = place(A',C',P)';
A_0 = [(A-B*K) (B*K);
        zeros(size(A)) (A-L*C)];
B_0 = [B*Nbar -B*K*F;
        zeros(size(B)) -L*C*F];
C_0 = [C zeros(size(C))];
D_0 = 0;
GC_0 = ss(A_0,B_0,C_0,D_0);
t2_f = 1;
t2_f_p = 201;
%CLOSED LOOP SIMULATION
t2 = 0.25:0.005:t2_f; %
r2 = 0*ones(size(t2));

```

```

xo = [X(end,1) X(end,2) X(end,3) X(end,4)];
[y2,t2,XX2] = lsim(GC_0,[r2; P_OUT(1,51:t2_f_p)],t2,[xo 0 0 0 0]);
X2 = [XX2(:,1) XX2(:,2) XX2(:,3) XX2(:,4)];
y2 = y2 + (C*F*P_OUT(1,51:t2_f_p))';
%-----SECOND ITERATION-----
u2 = (- K*X2')' - (K*F*P_OUT(1,51:t2_f_p))';
n = length(t2);
for i = 1:n
V2dot(i) = X2(i,:)*Q*X2(i,:)' + u2(i,:)'*R*u2(i,:);
end
V2dot = V2dot';
V2 = cumtrapz(t2,V2dot);
T = n;
for k = 1:n
Y2(k) = V2(T,1)-V2(k,1);
end
Y2 = Y2';
clear x1
for k = 1:n
x1(k) = X2(k,1)^2-X2(T,1)^2;
end
clear x2
for k = 1:n
x2(k) = X2(k,1)*X2(k,2)-X2(T,1)*X2(T,2);
end
clear x3
for k = 1:n
x3(k) = X2(k,1)*X2(k,3)-X2(T,1)*X2(T,3);
end
clear x4

```

```
for k = 1:n
x4(k) = X2(k,1)*X2(k,4)-X2(T,1)*X2(T,4);
end
clear x5
for k = 1:n
x5(k) = X2(k,2)^2-X2(T,2)^2;
end
clear x6
for k = 1:n
x6(k) = X2(k,2)*X2(k,3)-X2(T,2)*X2(T,3);
end
clear x7
for k = 1:n
x7(k) = X2(k,2)*X2(k,4)-X2(T,2)*X2(T,4);
end
clear x8
for k = 1:n
x8(k) = X2(k,3)^2-X2(T,3)^2;
end
clear x9
for k = 1:n
x9(k) = X2(k,3)*X2(k,4)-X2(T,3)*X2(T,4);
end
clear x10
for k = 1:n
x10(k) = X2(k,4)^2-X2(T,4)^2;
end
xx = vertcat(x1,x2,x3,x4,x5,x6,x7,x8,x9,x10);
p = pinv(xx*xx')*xx*Y2;
P = [p(1,1) p(2,1)/2 p(3,1)/2 p(4,1)/2;
```

```

    p(2,1)/2 p(5,1) p(6,1)/2 p(7,1)/2;
    p(3,1)/2 p(6,1)/2 p(8,1) p(9,1)/2;
    p(4,1)/2 p(7,1)/2 p(9,1)/2 p(10,1)];
K = 1*B'*P
%REFERENCE-GAIN
Nbar = rscale(sys_ss,K);
%OBSERVER
poles = eig(A-B*K);
P = [-22 -23 -24 -25];
L = place(A',C',P)';
A_0 = [(A-B*K) (B*K);
        zeros(size(A)) (A-L*C)];
B_0 = [B*Nbar -B*K*F;
        zeros(size(B)) -L*C*F];
C_0 = [C zeros(size(C))];
D_0 = 0;
GC_0 = ss(A_0,B_0,C_0,D_0);
t3_f = 10;
t3_f_p = 2001;
%CLOSED LOOP SIMULATION
t3 = t2_f:0.005:t3_f;
r3 = 0*ones(size(t3));
xo = [X2(end,1) X2(end,2) X2(end,3) X2(end,4)];
[y3,t3,XX3] = lsim(GC_0,[r3; P_OUT(1,t2_f_p:t3_f_p)],t3,[xo 0 0 0 0]);
X3 = [XX3(:,1) XX3(:,2) XX3(:,3) XX3(:,4)];
y3 = y3 + (C*F*P_OUT(1,t2_f_p:t3_f_p))';
%PLOTING
times = vertcat(t,t2,t3);
states = vertcat(X,X2,X3);
outputs = vertcat(y,y2,y3);

```

```

figure(1)
plot(times,states,'LineWidth',2)
title('VTOL STATES WITH OUTPUT PERTURBATION')
legend('X','u','\theta','p')    %[x u phi p]
grid on
xlabel('Time (s)')
ylim([-3 4])
%%%%%
figure(2)
times = vertcat(t,t2);
IND = horzcat(x,xx);
DEP = vertcat(Y,Y2);
plot(times,IND,'LineWidth',2)
title('INDEPENDENT VARIABLE WITH OUTPUT PERTURBATION')
legend('$\dot{X}_\Delta^2$', '$\dot{X}\ddot{X}_\Delta$', ...
'$\dot{X}\dot{\theta}_\Delta$', '$\dot{X}\ddot{\theta}_\Delta$', ...
'$\ddot{X}_\Delta^2$', '$\ddot{X}\dot{\theta}_\Delta$', ...
'$\ddot{X}\ddot{\theta}_\Delta$', '$\dot{\theta}_\Delta^2$', ...
'$\dot{\theta}\ddot{\theta}_\Delta$', '$\ddot{\theta}_\Delta^2$', ...
'Interpreter','latex');
grid on
xlabel('Time (s)')
ylim([-7 13])
%%%%%
figure(3)
plot(times,DEP,'LineWidth',2)
title('DEPENT VARIABLE WITH OUTPUT PERTURBATION')
legend('Y')
grid on
xlabel('Time (s)')

```

```
%MATLAB TO SIMULINK EXPORTATION  
output1 = timeseries(outputs(:,1),times);  
output2 = timeseries(outputs(:,2),times);
```

APPENDIX C

DISTURBED IRL MATLAB PROGRAMMING (4 ITERS.)

```
clc
clear all
g = 9.81;
%-----STATE SPACE-----
A = [0 1 0 0;0 0 -g 0;0 0 0 1;0 0 0 0]; %[x u phi p]
B = [0 0 0 1]';
C = [1 0 0 0;0 0 1 0];
D = [0 0]';
G = ss(A,B,C,D);
%DEFINE INITIAL CONTROLLER
Q = C'*C*8000;
R = 1;
K = lqr(A,B,Q,R)
%REFERENCE-GAIN
Cn = [1 0 1 0];
sys_ss = ss(A,B,Cn,0);
Nbar = rscale(sys_ss,K);
```

```

%PERTURBTIONS

Perturbations = importdata('Perturbations.mat');
P_OUT = Perturbations.PERT_100';
F = [1 0 1 0]'; %[x u phi p]
%OBSERVER
poles = eig(A-B*K);
P = [-22 -23 -24 -25];
L = place(A',C',P)';
A_0 = [(A-B*K) (B*K);
        zeros(size(A)) (A-L*C)];
B_0 = [B*Nbar -B*K*F;
        zeros(size(B)) -L*C*F];
C_0 = [C zeros(size(C))];
D_0 = 0;
GC_0 = ss(A_0,B_0,C_0,D_0);
%CLOSED LOOP SIMULATION
t = 0:0.005:.25;
r = 0*ones(size(t));
xo = [1 0 .1 0]; %INITIAL RANDOM STATES
[y,t,XX] = lsim(GC_0,[r; P_OUT(1,1:51)],t,[xo 0 0 0 0]);
X = [XX(:,1) XX(:,2) XX(:,3) XX(:,4)];
y = y + (C*F*P_OUT(1,1:51))';
%-----FIRST ITERATION-----
u = (- K*X')' - (K*F*P_OUT(1,1:51))';
n = length(t);
for i = 1:n
Vdot(i) = X(i,:)*Q*X(i,:)' + u(i,:)'*R*u(i,:);
end
Vdot = Vdot';
V = cumtrapz(t,Vdot);

```

```
T = n;
for k = 1:n
Y(k) = V(T,1)-V(k,1);
end
Y = Y';
for k = 1:n
x1(k) = X(k,1)^2-X(T,1)^2;
end
for k = 1:n
x2(k) = X(k,1)*X(k,2)-X(T,1)*X(T,2);
end
for k = 1:n
x3(k) = X(k,1)*X(k,3)-X(T,1)*X(T,3);
end
for k = 1:n
x4(k) = X(k,1)*X(k,4)-X(T,1)*X(T,4);
end
for k = 1:n
x5(k) = X(k,2)^2-X(T,2)^2;
end
for k = 1:n
x6(k) = X(k,2)*X(k,3)-X(T,2)*X(T,3);
end
for k = 1:n
x7(k) = X(k,2)*X(k,4)-X(T,2)*X(T,4);
end
for k = 1:n
x8(k) = X(k,3)^2-X(T,3)^2;
end
for k = 1:n
```

```
x9(k) = X(k,3)*X(k,4)-X(T,3)*X(T,4);
end
for k = 1:n
x10(k) = X(k,4)^2-X(T,4)^2;
end
x = vertcat(x1,x2,x3,x4,x5,x6,x7,x8,x9,x10);
p = pinv(x*x')*x*Y;
P = [p(1,1) p(2,1)/2 p(3,1)/2 p(4,1)/2;
      p(2,1)/2 p(5,1) p(6,1)/2 p(7,1)/2;
      p(3,1)/2 p(6,1)/2 p(8,1) p(9,1)/2;
      p(4,1)/2 p(7,1)/2 p(9,1)/2 p(10,1)];
K = 1*B'*P
%REFERENCE-GAIN
Nbar = rscale(sys_ss,K);
%OBSERVER
poles = eig(A-B*K);
P = [-22 -23 -24 -25];
L = place(A',C',P)';
A_0 = [(A-B*K) (B*K);
        zeros(size(A)) (A-L*C)];
B_0 = [B*Nbar -B*K*F;
        zeros(size(B)) -L*C*F];
C_0 = [C zeros(size(C))];
D_0 = 0;
GC_0 = ss(A_0,B_0,C_0,D_0);
t2_f = 1;
t2_f_p = 201;
%CLOSED LOOP SIMULATION
t2 = 0.25:0.005:t2_f; %
r2 = 0*ones(size(t2));
```

```

xo = [X(end,1) X(end,2) X(end,3) X(end,4)];
[y2,t2,XX2] = lsim(GC_0,[r2; P_OUT(1,51:t2_f_p)],t2,[xo 0 0 0 0]);
X2 = [XX2(:,1) XX2(:,2) XX2(:,3) XX2(:,4)];
y2 = y2 + (C*F*P_OUT(1,51:t2_f_p))';
%-----SECOND ITERATION-----
u2 = (- K*X2')' - (K*F*P_OUT(1,51:t2_f_p))';
n = length(t2);
for i = 1:n
V2dot(i) = X2(i,:)*Q*X2(i,:)' + u2(i,:)'*R*u2(i,:);
end
V2dot = V2dot';
V2 = cumtrapz(t2,V2dot);
T = n;
for k = 1:n
Y2(k) = V2(T,1)-V2(k,1);
end
Y2 = Y2';
clear x1
for k = 1:n
x1(k) = X2(k,1)^2-X2(T,1)^2;
end
clear x2
for k = 1:n
x2(k) = X2(k,1)*X2(k,2)-X2(T,1)*X2(T,2);
end
clear x3
for k = 1:n
x3(k) = X2(k,1)*X2(k,3)-X2(T,1)*X2(T,3);
end
clear x4

```

```
for k = 1:n
x4(k) = X2(k,1)*X2(k,4)-X2(T,1)*X2(T,4);
end
clear x5
for k = 1:n
x5(k) = X2(k,2)^2-X2(T,2)^2;
end
clear x6
for k = 1:n
x6(k) = X2(k,2)*X2(k,3)-X2(T,2)*X2(T,3);
end
clear x7
for k = 1:n
x7(k) = X2(k,2)*X2(k,4)-X2(T,2)*X2(T,4);
end
clear x8
for k = 1:n
x8(k) = X2(k,3)^2-X2(T,3)^2;
end
clear x9
for k = 1:n
x9(k) = X2(k,3)*X2(k,4)-X2(T,3)*X2(T,4);
end
clear x10
for k = 1:n
x10(k) = X2(k,4)^2-X2(T,4)^2;
end
xx = vertcat(x1,x2,x3,x4,x5,x6,x7,x8,x9,x10);
p = pinv(xx*xx')*xx*Y2;
P = [p(1,1) p(2,1)/2 p(3,1)/2 p(4,1)/2;
```

```

    p(2,1)/2 p(5,1) p(6,1)/2 p(7,1)/2;
    p(3,1)/2 p(6,1)/2 p(8,1) p(9,1)/2;
    p(4,1)/2 p(7,1)/2 p(9,1)/2 p(10,1)];
K = 1*B'*P
%REFERENCE-GAIN
Nbar = rscale(sys_ss,K);
%OBSERVER
poles = eig(A-B*K);
P = [-22 -23 -24 -25];
L = place(A',C',P)';
A_0 = [(A-B*K) (B*K);
        zeros(size(A)) (A-L*C)];
B_0 = [B*Nbar -B*K*F;
        zeros(size(B)) -L*C*F];
C_0 = [C zeros(size(C))];
D_0 = 0;
GC_0 = ss(A_0,B_0,C_0,D_0);
t3_f = 5;
t3_f_p = 1001;
%CLOSED LOOP SIMULATION
t3 = t2_f:0.005:t3_f;
r3 = 0*ones(size(t3));
xo = [X2(end,1) X2(end,2) X2(end,3) X2(end,4)];
[y3,t3,XX3] = lsim(GC_0,[r3; P_OUT(1,t2_f_p:t3_f_p)],t3,[xo 0 0 0 0]);
X3 = [XX3(:,1) XX3(:,2) XX3(:,3) XX3(:,4)];
y3 = y3 + (C*F*P_OUT(1,t2_f_p:t3_f_p))';
%-----THIRD ITERATION-----
u3 = (- K*X3')' - (K*F*P_OUT(1,t2_f_p:t3_f_p))';
n = length(t3);
for i = 1:n

```

```
V3dot(i) = X3(i,:)*Q*X3(i,:)' + u3(i,:)'*R*u3(i,:);
end
V3dot = V3dot';
V3 = cumtrapz(t3,V3dot);
T = n;
for k = 1:n
Y3(k) = V3(T,1)-V3(k,1);
end
Y3 = Y3';
clear x1
for k = 1:n
x1(k) = X3(k,1)^2-X3(T,1)^2;
end
clear x2
for k = 1:n
x2(k) = X3(k,1)*X3(k,2)-X3(T,1)*X3(T,2);
end
clear x3
for k = 1:n
x3(k) = X3(k,1)*X3(k,3)-X3(T,1)*X3(T,3);
end
clear x4
for k = 1:n
x4(k) = X3(k,1)*X3(k,4)-X3(T,1)*X3(T,4);
end
clear x5
for k = 1:n
x5(k) = X3(k,2)^2-X3(T,2)^2;
end
clear x6
```

```
for k = 1:n
x6(k) = X3(k,2)*X3(k,3)-X3(T,2)*X3(T,3);
end
clear x7
for k = 1:n
x7(k) = X3(k,2)*X3(k,4)-X3(T,2)*X3(T,4);
end
clear x8
for k = 1:n
x8(k) = X3(k,3)^2-X3(T,3)^2;
end
clear x9
for k = 1:n
x9(k) = X3(k,3)*X3(k,4)-X3(T,3)*X3(T,4);
end
clear x10
for k = 1:n
x10(k) = X3(k,4)^2-X3(T,4)^2;
end
xxx = vertcat(x1,x2,x3,x4,x5,x6,x7,x8,x9,x10);
p = pinv(xxx*xxx')*xxx*Y3;
P = [p(1,1) p(2,1)/2 p(3,1)/2 p(4,1)/2;
      p(2,1)/2 p(5,1) p(6,1)/2 p(7,1)/2;
      p(3,1)/2 p(6,1)/2 p(8,1) p(9,1)/2;
      p(4,1)/2 p(7,1)/2 p(9,1)/2 p(10,1)];
K = 1*B'*P
%REFERENCE-GAIN
Nbar = rscale(sys_ss,K);
%OBSERVER
poles = eig(A-B*K);
```

```

P = [-22 -23 -24 -25];
L = place(A',C',P)';
A_0 = [(A-B*K) (B*K);
        zeros(size(A)) (A-L*C)];
B_0 = [B*Nbar -B*K*F;
        zeros(size(B)) -L*C*F];
C_0 = [C zeros(size(C))];
D_0 = 0;
GC_0 = ss(A_0,B_0,C_0,D_0);
t4_f = 10;
t4_f_p = 2001;
%CLOSED LOOP SIMULATION
t4 = t3_f:0.005:t4_f;
r4 = 0*ones(size(t4));
xo = [X3(end,1) X3(end,2) X3(end,3) X3(end,4)];
[y4,t4,XX4] = lsim(GC_0,[r4; P_OUT(1,t3_f_p:t4_f_p)],t4,[xo 0 0 0 0]);
X4 = [XX4(:,1) XX4(:,2) XX4(:,3) XX4(:,4)];
y4 = y4 + (C*F*P_OUT(1,t3_f_p:t4_f_p))';
%PLOTING
times = vertcat(t,t2,t3,t4);
states = vertcat(X,X2,X3,X4);
outputs = vertcat(y,y2,y3,y4);
figure(1)
plot(times,states,'LineWidth',2)
title('VTOL STATES WITH OUTPUT PERTURBATION (4 ITER.)')
legend('X','u','\theta','p') % [x u phi p]
grid on
xlabel('Time (s)')
ylim([-3 4])
%%%%

```

```

figure(2)
times = vertcat(t,t2,t3);
IND = horzcat(x,xx,xxx);
DEP = vertcat(Y,Y2,Y3);
plot(times,IND,'LineWidth',2)
title('INDEPENDENT VARIABLE WITH OUTPUT PERTURBATION')
legend('$\dot{X}_\Delta ^{2}$','$\dot{X}\ddot{X}_\Delta$', ...
'$\dot{X}\dot{\theta}_\Delta$', '$\dot{X}\ddot{\theta}_\Delta$', ...
'$\ddot{X}_\Delta ^{2}$','$\ddot{X}\dot{\theta}_\Delta$', ...
'$\ddot{X}\ddot{\theta}_\Delta$', '$\dot{\theta}_\Delta ^{2}$$', ...
'$\dot{\theta}_\Delta\ddot{\theta}_\Delta$', '$\ddot{\theta}_\Delta^2$', ...
'Interpreter','latex');
grid on
xlabel('Time (s)')
ylim([-7 13])
%%%%
figure(3)
plot(times,DEP,'LineWidth',2)
title('DEPENT VARIABLE WITH OUTPUT PERTURBATION (4 ITER.)')
legend('Y')
grid on
xlabel('Time (s)')
%MATLAB TO SIMULINK EXPORTATION
output1 = timeseries(outputs(:,1),times);
output2 = timeseries(outputs(:,2),times);

```

BIBLIOGRAPHY

- Abeykoon, C., Sri-Amphorn, P., and Fernando, A. (2020). Optimization of fused deposition modeling parameters for improved pla and abs 3d printed structures. *International Journal of Lightweight Materials and Manufacture*, 3(3):284–297.
- Aktas Y.O., Ozdemir U., D. Y. e. a. (2016). Rapid prototyping of a fixed-wing vtol uav for design testing. *J Intell Robot Syst*, 84:639–664.
- Antonio Lanzotti, Marzio Grasso, G. S. e. a. (2015). The impact of process parameters on mechanical properties of parts fabricated in pla with an open-source 3-d printer. *Rapid Prototyping Journal*, 21:604–617.
- Atakok, G., Kam, M., and Koc, H. B. (2022). Tensile, three-point bending and impact strength of 3d printed parts using pla and recycled pla filaments: A statistical investigation. *Journal of Materials Research and Technology*, 18:1542–1554.
- Cabrales-Ramírez, L. E., Dominguez, V. H., Gutierrez-Martinez, M. A., Javier Ollervides-Vazquez, E., Rojo-Rodriguez, E. G., Tellez-Belkotosky, P. A., and Garcia-Salazar, O. (2021). Trajectory tracking based on a saturated pd controller on $se(3)$ for a micro coaxial drone. In *2021 International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE)*, pages 146–153.
- Chacón, J., Caminero, M., García-Plaza, E., and Núñez, P. (2017). Additive manufacturing of pla structures using fused deposition modelling: Effect of process parameters on mechanical properties and their optimal selection. *Materials and Design*, 124:143–157.

- Chen, C.-T. (1984). *Linear system theory and design*. Saunders college publishing.
- Chen, R., Zhu, X., Zhou, Z., Wang, Z., and Zhang, T. (2018). Study on the structure design of solar powerd uav. In *2018 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pages 1–7.
- Cruzatty, C., Sarmiento, E., Valencia, E., and Cando, E. (2022). Design methodology of a uav propeller implemented in monitoring activities. *Materials Today: Proceedings*, 49:115–121. Advances in Mechanical Engineering Trends.
- De Vivo, L., Tran, D., and Kuester, F. (2019). Towards design of a 3d printable prandtl box-wing unmanned aerial vehicle. In *2019 IEEE Aerospace Conference*, pages 1–17.
- Dikshit, V., Goh, G. D., Nagalingam, A. P., Goh, G. L., and Yeong, W. Y. (2020). Chapter 17 - recent progress in 3d printing of fiber-reinforced composite and nanocomposites. In Han, B., Sharma, S., Nguyen, T. A., Longbiao, L., and Bhat, K. S., editors, *Fiber-Reinforced Nanocomposites: Fundamentals and Applications*, Micro and Nano Technologies, pages 371–394. Elsevier.
- Dominguez, V. H., Garcia-Salazar, O., Amezcua-Brooks, L., Reyes-Osorio, L. A., Santana-Delgado, C., and Rojo-Rodriguez, E. G. (2022). Micro coaxial drone: Flight dynamics, simulation and ground testing. *Aerospace*, 9(5).
- Harpool, T. D. (2016). *Observing the effect of infill shapes on the tensile characteristics of 3D printed plastic parts*. PhD thesis, Wichita State University.
- Harpool, T. D., Alarifi, I. M., Alshammari, B. A., Aabid, A., Baig, M., Malik, R. A., Mohamed Sayed, A., Asmatulu, R., and EL-Bagory, T. M. A. A. (2021). Evaluation of the infill design on the tensile response of 3d printed polylactic acid polymer. *Materials*, 14(9).
- He, L., Aouf, N., and Song, B. (2021). Explainable deep reinforcement learning for uav autonomous path planning. *Aerospace Science and Technology*, 118:107052.

- Henderson, L., Glaser, T., and Kuester, F. (2017). Towards bio-inspired structural design of a 3d printable, ballistically deployable, multi-rotor uav. In *2017 IEEE Aerospace Conference*, pages 1–7.
- HU, Z., GAO, X., WAN, K., ZHAI, Y., and WANG, Q. (2021). Relevant experience learning: A deep reinforcement learning method for uav autonomous motion planning in complex unknown environments. *Chinese Journal of Aeronautics*, 34(12):187–204.
- Johansson, F. (2016). Optimizing fused filament fabrication 3d printing for durability : Tensile properties and layer bonding.
- Kam, M., Ipekci, A., and Sengul, O. (2022). Taguchi optimization of fused deposition modeling process parameters on mechanical characteristics of pla+ filament material. *Scientia Iranica*, 29(1):79–89.
- Kapsalis, S., Panagiotou, P., and Yakinthos, K. (2021). Cfd-aided optimization of a tactical blended-wing-body uav platform using the taguchi method. *Aerospace Science and Technology*, 108:106395.
- Kermode, A., Barnard, R., and Philpott, D. (2012). *Mechanics of Flight*. Pearson.
- Kozior, T., Mamun, A., Trabelsi, M., Sabantina, L., and Ehrmann, A. (2020). Quality of the surface texture and mechanical properties of fdm printed samples after thermal and chemical treatment. *Strojniški vestnik - Journal of Mechanical Engineering*, 66(2):105–113.
- Kutty, H. A. and Rajendran, P. (2017). 3d cfd simulation and experimental validation of small apc slow flyer propeller blade. *Aerospace*, 4(1).
- Leishman, G. J. (2006). *Principles of helicopter aerodynamics with CD extra*. Cambridge university press.
- Li, Z.-k., Guo, J., Li, F.-m., Yan, Y.-d., Yang, Y., and Liu, Y. (2019). Structure modelling and simulation analysis of six-rotor uav. In Zhang, X., editor,

- The Proceedings of the 2018 Asia-Pacific International Symposium on Aerospace Technology (APISAT 2018)*, pages 2811–2821, Singapore. Springer Singapore.
- Lin, X., Liu, J., Yu, Y., and Sun, C. (2020). Event-triggered reinforcement learning control for the quadrotor uav with actuator saturation. *Neurocomputing*, 415:135–145.
- Liu, G., Xiong, Y., and Zhou, L. (2021). Additive manufacturing of continuous fiber reinforced polymer composites: Design opportunities and novel applications. *Composites Communications*, 27:100907.
- Mo, Q., Guan, H., He, S., Liu, Y., and Guo, R. (2021). Guidelines for the computational domain size on an urban-scale VAWT. *Journal of Physics: Conference Series*, 1820(1):012177.
- Morteza BehzadnasabAli, Y. A. Y. (2016). Effects of 3d printer nozzle head temperature on the physical and mechanical properties of pla based product. *Proceedings of the 12th International Seminar on Polymer Science and Technology*.
- Negrello F., Silvestri P., L. A. (2016). Preliminary design of a small-sized flapping uav: Ii. kinematic and structural aspects. *Meccanica*, 51:1369–1385.
- Panagiotou, P., Kaparos, P., and Yakinthos, K. (2014). Winglet design and optimization for a male uav using cfd. *Aerospace Science and Technology*, 39:190–205.
- Pascariu, I. S. and Zaharia, S. M. (2020). Design and testing of an unmanned aerial vehicle manufactured by fused deposition modeling. *Journal of Aerospace Engineering*, 33(4):06020002.
- Ravindrababu, S., Govdeli, Y., Wong, Z. W., and Kayacan, E. (2018). Evaluation of the influence of build and print orientations of unmanned aerial vehicle parts fabricated using fused deposition modeling process.
- Richards, N., Ouwerkerk, J., Brown, B., and Cohen, K. (2020). 10 - design, fabrication and flight testing of small uavs using additive manufacturing. In Cazaurang,

- F., Cohen, K., and Kumar, M., editors, *Multi-Rotor Platform-based UAV Systems*, pages 203–215. ISTE.
- Saeed, A. S., Younes, A. B., Cai, C., and Cai, G. (2018). A survey of hybrid unmanned aerial vehicles. *Progress in Aerospace Sciences*, 98:91–105.
- Seeni, A. (2019). Aerodynamic performance characterization and static structural analysis of slotted propeller: Part a effect of position. *Mathematical Modelling of Engineering Problems*, 6:611–624.
- Stengel, R. F. (2022). *Flight dynamics*. Princeton university press.
- Stevens, B. L., Lewis, F. L., and Johnson, E. N. (2015). *Aircraft control and simulation: dynamics, controls design, and autonomous systems*. John Wiley & Sons.
- Suteja, T. J. and Soesanti, A. (2020). Mechanical properties of 3d printed polylactic acid product for various infill design parameters: A review. *Journal of Physics: Conference Series*, 1569(4):042010.
- Tavares, S. M. O., Moreira, S. J., de Castro, P. M. S. T., and Gamboa, P. V. (2017). Morphing aeronautical structures: A review focused on uavs and durability assessment. In *2017 IEEE 4th International Conference Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD)*, pages 49–52.
- Uddin, M. S., Sidek, M. F. R., Faizal, M. A., Ghomashchi, R., and Pramanik, A. (2017). Evaluating Mechanical Properties and Failure Mechanisms of Fused Deposition Modeling Acrylonitrile Butadiene Styrene Parts. *Journal of Manufacturing Science and Engineering*, 139(8). 081018.
- Wach, R. A., Wolszczak, P., and Adamus-Wlodarczyk, A. (2018). Enhancement of mechanical properties of fdm-pla parts via thermal annealing. *Macromolecular Materials and Engineering*, 303(9):1800169.
- Yao, T., Deng, Z., Zhang, K., and Li, S. (2019). A method to predict the ultimate tensile strength of 3d printing polylactic acid (pla) materials with different printing orientations. *Composites Part B: Engineering*, 163:393–402.

Zhang Sitong, L. Y. and Qianhui, D. (2022). Autonomous navigation of uav in multi-obstacle environments based on a deep reinforcement learning approach. *Appl. Soft Comput.*, 115(C).

Zhu, Z., Guo, H., and Ma, J. (2019). Aerodynamic layout optimization design of a barrel-launched uav wing considering control capability of multiple control surfaces. *Aerospace Science and Technology*, 93:105297.

AUTOBIOGRAPHY

Ing. Víctor Hugo Domínguez de la Cruz

Candidate to obtain the degree of
Maestría en Ingeniería en Aeronáutica
con orientación en Dinámica de Vuelo

Universidad Autónoma de Nuevo León
Facultad de Ingeniería Mecánica y Eléctrica

Thesis:

DESIGN, MODELING, AND CONTROL OF A MICRO COAXIAL
DRONE USING OPTIMAL ADAPTIVE CONTROL BASED ON
REINFORCEMENT LEARNING

Victor Hugo Dominguez de la Cruz, born in Mexico City, Mexico in July 10th, 1993. With parents Julian Dominguez Santos as father and Angelica de la Cruz Cordova as mother. Graduated in December 2018 from Facultad de Ingenieria Mecanica y Electrica (FIME) at the Universidad Autonoma de Nuevo Leon (UANL) with a bachelor degree in aeronautical engineering.