

Thrust Vector Control of a Coaxial UAV Rocket Analog

Bianca Diana Turneanu (s193422)
Master of Science in Engineering
2021

Thrust Vector Control of a Coaxial UAV Rocket Analog

Report written by:

Bianca Diana Turneanu (s193422)

Advisor(s):

Evangelos Boukas, Associate Professor at the Electrical Engineering Department of DTU
Ilmar Santos, Professor, Dr.-Ing. dr.techn. livre-docente, Mechanical Engineering Department of DTU

DTU Electrical Engineering

Technical University of Denmark
2800 Kgs. Lyngby
Denmark

elektro@elektro.dtu.dk

Project period: 25. September- 10. April

ECTS: 30

Education: M.Sc.

Field: Autonomous Systems

Class: Confidential

Edition: 1. edition

Remarks: This report is submitted as partial fulfillment of the requirements for graduation in the above education at the Technical University of Denmark.

Copyrights: ©Bianca Diana Turneanu, 2021

Summary

Copenhagen Suborbitals changes the type of thrust vector control for the upcoming rocket Spica, from the previous jet vanes solution, to gimbaling the engine. While this is a common solution for spacecraft, it is the first time this system will be used by the organisation and it requires understanding of what the change involves. The goal of this project is to study the overall principles of this new type of thrust vector control, then create a system consisting of control and navigation elements, taking into consideration flight elements, for the duration of the powered flight.

The navigation tests carried on a robotic platform show that the best results, in terms of minimal offset, noise and drift over time, were given by the Mahony filter, seconded by Madgwick filter (provided steady calibrated measurements). The approach chosen for the project is Newton-Euler dynamics modelling in order to ensure reliable mechatronic design, as well as model-based control due to scalability and reliability, followed by simulation, experimental validation and analysis. The controller was chosen to be a Proportional Integral Derivative controller, tuned model-based, in order to critically dampen the system.

Since the Spica rocket is in its initial production phases, the control system developed in this project will be tested on a simplified hardware prototype, simulating the rocket. The reason for this was to insure a testing platform of the model-based approach and assess its performance. This prototype (drone) consists of two contra rotating coaxial propellers representing the thrust and a rudder which simulates the angle deflection produced by a gimbal thrust system. The systems created in this project consist of the navigation system based on an IMU with 9 DoF with sensor fusion and a control system based on the kinematics and dynamics (Newton-Euler) of the drone prototype.

In conclusion, quaternions based navigation filters provide reliable IMU measurements fit for lower processing micro-controllers and model-based tuning of controller allows for a more complete understanding of the system and its limitations, as well as better tuned algorithms for the system.

Preface

This thesis was prepared at the department of Automation and Control at the Technical University of Denmark in fulfillment of the requirements for acquiring a Master of Science in Engineering degree in Autonomous Systems.

Acknowledgements

I would like to thank supervisors Prof. Evangelos Boukas and Prof. Ilmar Santos for their invaluable guidance, support, considerable dedication and effort; Prof. Matteo Fumagalli for the courage to put the foundation of a challenging project, and paving the way; Prof. Jens Christian Andersen for support during uncertain and challenging times. Kim Jensen for his support, encouragement and kindly lending propellers and RC equipment for this project. Jacob Larsen, Flemming Nyboe, Peter Scott for insightful discussions and sharing of their invaluable experience in rocket engineering, as well as Martin H. Pedersen, Bo Brændstrup, Uffe Ravn, Niels Foldager and so many others in Copenhagen Suborbitals.

Contents

Summary	i
Preface	iii
Acknowledgements	v
Contents	vii
1 Introduction	1
1.1 Copenhagen Suborbitals	1
1.2 Motivation and Project scope	1
1.3 Requirements	3
1.4 Comparison between hardware prototype and rocket	3
2 Spica rocket mission	7
2.1 Spica rocket	7
2.2 Rocket aerodynamics and stability	9
2.3 Thrust vector control	12
3 UAV prototype - system description	15
3.1 Overview	15
3.2 Drone building and assembly process	15
3.3 Electrical components	18
3.4 Servo motor	18
3.5 Rudder	20
3.6 Electronics IMU, Arduino	22
4 Inertial Navigation	25
4.1 Investigations of inertial navigation	25
4.2 Rotation representation	25
4.3 Heading	26
4.4 The Inertial Measurement Unit sensor	26
4.5 IMU sensor errors	27
4.6 IMU calibration process	28
4.7 Sensor fusion	31
4.8 Hardware used: IMU and Arduino	33

5 Attitude dynamics: modelling and control	35
5.1 Model-based design approach	35
5.2 Model: kinematics and dynamics - Newton-Euler approach for rigid multi-body systems	36
5.3 Control	53
6 Experimental validation and results	59
6.1 Experimental setup for sensor fusion methods	59
6.2 Experimental setup for control algorithm	66
7 Discussion and Future Work	69
7.1 Discussion	69
7.2 Future work	71
8 Conclusion	73
Appendices	75
A Appendix	77
Bibliography	103

List of Figures

1.1	Timeline of rockets launched and planned by Copenhagen Suborbitals	2
1.2	Comparison between rocket and drone	4
2.1	Description of Spica rocket mission	8
2.2	Rockets launch site in Baltic Sea	8
2.3	Forces acting on a rocket. Source: "Introduction to Rocket Science and Engineering" by : Travis Taylor, 2017	9
2.4	Flight angle. Source: "Introduction to Rocket Science and Engineering" by : Travis Taylor, 2017	10
2.5	Capture of jet vanes function from a previous rocket	11
2.6	Gimbaled thrust. Source: NASA	12
3.1	UAV frame CAD	16
3.2	CNC plasma cutting of the aluminium frame	16
3.3	Drone assembly	17
3.4	3D printed motor holders drawing	17
3.5	3D printed motor holders and propeller servo motor	18
3.6	Mounted servo motor	19
3.7	Welding of rudder	20
3.8	TIG welding of rudder	21
3.9	Rudder mount and fins	21
3.10	Rudder - servo motor assembly	22
3.11	Arduino Nano	23
3.12	Assembled drone	24
4.1	Axes difference on accelerometer-gyro and magnetometer	28
4.2	Magnetometer samples fit to ideal sphere	30
4.3	Hard offset effects calibration	30
4.4	Strong effects of hard iron and soft iron distortions in uncalibrated magnetometer measurements vs hard iron calibrated measurements	31
4.5	BNO055 IMU sensor	33
5.1	The servo-rudder mechanical system	36
5.2	Free-body diagram of the drone body frame	38
5.3	Free-body diagram of the servo-rudder mechanical system	39
5.4	Servo	40

5.5	Link	40
5.6	Rudder arm	40
5.7	Constraint Figure	41
5.8	Model validation of the constraint equation for body 1	48
5.9	Model validation of the constraint equation for body 2	49
5.10	Model validation of the constraint equation for body 3	50
5.11	Model validation with zero forces	51
5.12	Model validation with gravity only	52
5.13	Model validation with forces applied	53
5.14	Feedback control diagram	54
5.15	Pole-Zero Map	55
5.16	Root Locus Map	56
5.17	Simulink model used to test PID	57
5.18	Oscilloscope graph of the angular position of UAV and servo angle	58
6.1	IMU testbed on UR3 and Optitrack motion capture	60
6.2	IMU static testing	61
6.3	IMU inclination testing	61
6.4	Comparison of Filters on Pitch Axis	62
6.5	Comparison of Filters on Pitch Axis in Motion	64
7.1	Comparison of moving average filters	70
7.2	Comparison of moving average filters with removed delay	71
A.1	UAV frame front	78
A.2	Propeller Motor	79
A.3	Median filter effects on data	79
A.4	Stacked axes of reference Optitrack signal for static test	80
A.5	Stacked axes of integrated gyro signal for static test	81
A.6	Stacked axes of complementary filter signal for static test	82
A.7	Stacked axes of Mahony filter signal for static test	83
A.8	Stacked axes of Madgwick filter signal for static test	84
A.9	Tiled plot of roll axes for static test	85
A.10	Tiled plot of pitch axes for static test	86
A.11	Tiled plot of yaw axes for static test	87
A.12	Combined plot of roll axes for static test	88
A.13	Combined plot of yaw axes for static test	89
A.14	Stacked axes of reference Optitrack signal for motion test	90
A.15	Stacked axes of integrated gyro signal for motion test	91
A.16	Stacked axes of complementary filter signal for motion test	92
A.17	Stacked axes of Mahony filter signal for motion test	93
A.18	Stacked axes of Madgwick filter signal for motion test	94
A.19	Tiled plot of roll axes for motion test	95
A.20	Tiled plot of pitch axes for motion test	96

A.21 Tiled plot of yaw axes for motion test	97
A.22 Combined plot of roll axes for motion test	98
A.23 Combined plot of yaw axes for motion test	99
A.24 Model validation applied forces: body 0 angle	100
A.25 Model validation applied forces: body 1 angle	101
A.26 Model validation applied forces: body 0	102

List of Tables

6.1 Table of navigation filters per axis for static test	63
6.2 Table of navigation filters per axis for motion test	65
6.3 Table comparing RMSE performance of navigation filters	65

CHAPTER 1

Introduction

1.1 Copenhagen Suborbitals

Copenhagen Suborbitals is a space organisation located in Denmark. Since 2011, they have developed and flown six internal built rockets and space capsules, launching rockets from a sailing platform in international waters (Baltic Sea). The rockets are built in a workshop in Copenhagen, Denmark. The budget is minimal compared to professional space programs, therefore the guiding principle is building spacecraft with focus on functionality and cost-efficiency. This requires use of self-built production methods in the workshop.

The goal of the organisation is to develop technologies for flying crewed into space and return safely, which will happen with the Spica mission, which is the next phase in Copenhagen Suborbitals's space program. The previous rockets have been developed for testing technologies necessary for this mission, with considerations for human flight. The first Spica rocket will combine previous knowledge as well as new technologies, at a higher scale. It will be a reusable vehicle with a different thrust vector control system compared to previous rockets. It is necessary to be studied and understand the type of control systems are suitable for this rocket.

The navigation system of this rocket will be inertial based, involving inertial measurement units with 9 degrees of freedom: accelerometer, gyroscope, magnetometer. Since the rocket is still in early stages of development, a hardware prototype simulating the rocket will be developed, which will allow to test control algorithms that will build the foundation for Spica control strategies.

1.2 Motivation and Project scope

The project has two parts: navigation and control. These parts will be investigated in regards to the challenges they experience in flight, as well investigating the workflow of building such a system. The aim of this project is to study the control system involved in gimbaled nozzle thrust vector control and develop a workflow for control of a gimbaled engine for the duration of the powered flight, with measurements from an inertial navigation system. The rocket will have a sensor suite, however, the requirement



Figure 1.1: Timeline of rockets launched and planned by Copenhagen Suborbitals.

for this project is to base the navigation of the system on an inertial measurement unit sensor only. It is also required to research options for achieving absolute heading (yaw).

The main tasks are: understanding what is involved in rocket stability in flight and gimbaled engine TVC; building a technology demonstrator prototype simulating the rocket to test the control systems; developing a inertial state estimation system based on an inertial measurement unit; designing a controller that will use the inertial sensor data to control a system suitable for a gimbaled engine and testing the controller on a technology demonstrator prototype as a substitute for the rocket.

Technology demonstrator was settled to be a coaxial contra rotating dual copter that simulates the thrust of a rocket and the deflection within a few degrees through a rudder, simulating the gimbal engine in one axis. Therefore, the drone will be tested in one axis (pitch axis). It is outside the scope of this project to fly the drone freely, as that would require focus on additional hardware design and building of a thrust vector control system in two axes that could sustain control in flight. It is also outside of the scope of the project to control the thrust generated by the propellers as part of the control system - as a similarity to the rocket, where the thrust is controlled by the engine control system, separated from the guidance, navigation, control.

Another main objective of this project was interacting with hardware as part of the process, as opposed to running simulations only. This approach, although more challeng-

ing, time consuming and error-prone, was preferred because it provides the realistic and practical view of the results, along with places to improve. Hardware implementation also allows for understanding physical limitations of the systems that must be taken into consideration in the design and control process, such as actuator delays, slack and limits, behavior of electronic devices under different types of stress, signal processing challenges, physical misalignment issues, constraints to be taken into consideration.

The modelling of the system was based on first principles, Newton-Euler as opposed to Lagrange - as the focus of this project is understanding the interplay of forces acting on the system. The system is modeled as a multi-body, with modeled dynamics forces and reaction forces, along with their physical constraints. The reasoning was to allow understanding of the dynamics of the system, the separate parts acting on each other, the physical constraints and to allow mechatronic design work for the rocket based on understanding of these interactions and real-life challenges.

1.3 Requirements

1. investigating sources of errors in the inertial sensor, particularly the gyro drift in flight.
2. deciding on an absolute heading (yaw) sensor to correct the gyro orientation.
3. navigational attitude representation through a method avoiding singularities.
4. investigating navigation filters fit for lower processing power, Arduino class microcontrollers.
5. navigation filters with a average deviation of maximum 1.5 degrees from the true value.
6. 1 axis control that can be extended to 2 or 3 axes
7. Controller with max 30 percent overshoot and 1.5 sec settling time

1.4 Comparison between hardware prototype and rocket

The reason for using a testing platform in the form is that the rocket is not built yet, it currently at the start of its production phase and therefore, cannot be tested upon. However, the controller for such a rocket takes a long time to develop and should be developed alongside the hardware, not afterwards. Often cases, rocket attitude control systems cannot be tested on the real hardware prior to launch, therefore there is a need to be able to test on a technology demonstrator prototype instead, to assess the

capabilities and performance of the controller and familiarize with hardware challenges. This is the advantage of having the drone prototype, as opposed to purely running models in simulation: it allows for testing of control algorithms and assessing the challenges that arise when working with real hardware, which simulations can only partially replicate.

The drone and the rocket are comparable due to their similarities, which allowed for reduced scale and scope process replication for a first iteration of the prototype. Both drone and rocket are modelled as a rigid body: drone has a rectangular frame for stability purposes when static, whereas the rocket is cylindrical for aerodynamic purposes. The navigation is based off an IMU in both bodies: the IMU of the drone is a low-cost MEMS (micro-electro-mechanical system) technology, while the IMU used in the rocket will be a high precision sensor. Thrust force is an important component of the thrust vector control in both bodies: in the case of the rocket, it is provided through propellant combustion, whereas the drone has air flow pushed by its two propellers. The drone has the advantage of being less dangerous to operate due to avoiding the combustion process and makes tests easier, safer, quicker to carry.

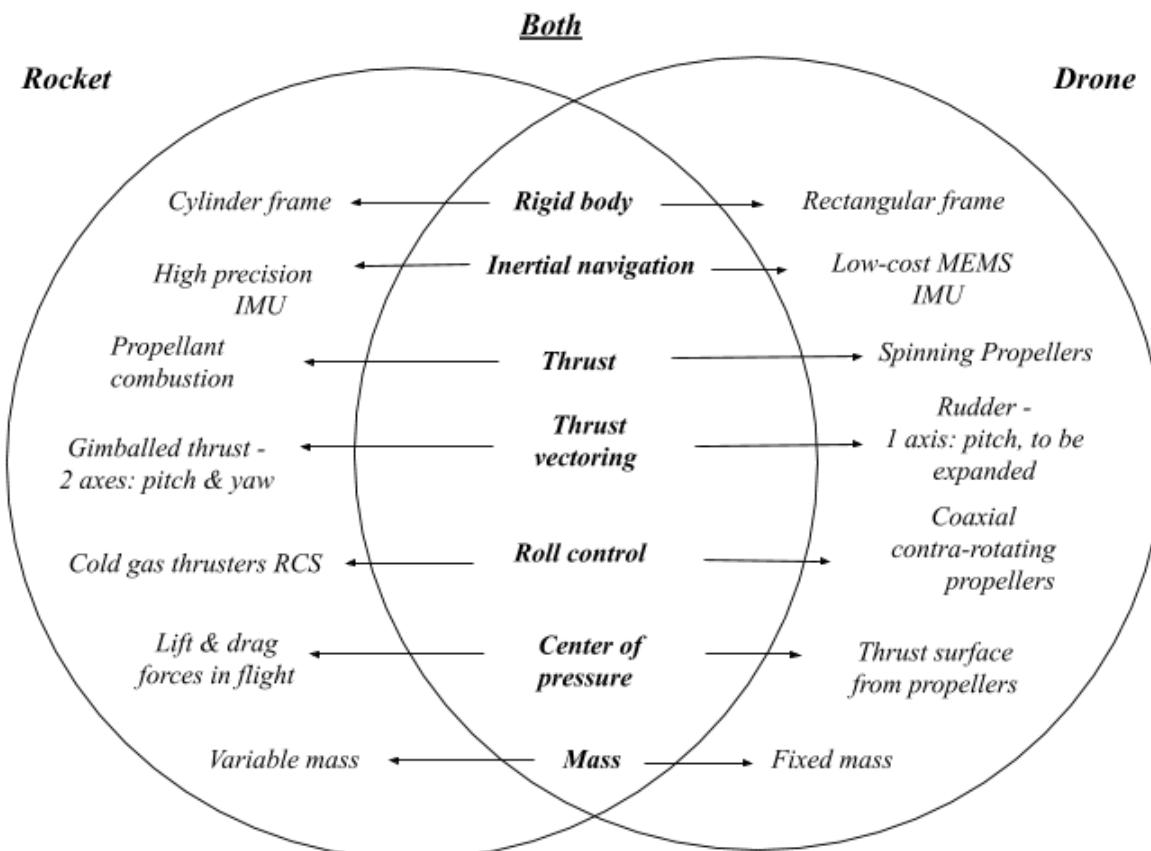


Figure 1.2: Comparison between rocket and drone.

The other component of the thrust vector control is the thrust deflection, carried in the rocket by its gimbaled engine, in two axis: pitch and yaw. In the case of the drone, the metal rudder handles the deflection of the propeller air flow, for now in one axis (pitch). There are three axes to both the rocket and the drone: roll, pitch and yaw. The roll axis (longitudinal) is controlled in both bodies by a separate control system. In the case of the rocket, the system is cold gas thrusters placed in the upper body of the rocket. The drone has roll control ensured by its two contra rotating propellers, which cancel each other's introduced moment about the longitudinal axis.

An important part of aerial stability is the location of the center of pressure on the rocket, relative to the center of mass. The center of pressure is the place where the aerodynamic forces act in flight. Although currently the drone will not be flown freely, the same principles apply for its aerial stability: it is necessary that the drone be top-heavy, like the rocket, in order to maintain the stability margin between the center of mass located above the center of pressure. Rocket has its center of pressure located towards bottom, near the location of its fins. The drone has its center of pressure on the area the propellers act on the rudder. The last important aspect in comparing these two systems is their mass. There is a considerable difference in their size and mass, however, they are not built to scale of each other. One of the key properties of a rocket is that the depletion of its fuel in flight leads to variable mass and moment of inertia. This is something that is not replicated by the drone at this time, drone having instead fixed mass and moment of inertia.

Similarities:

- Inertial navigation
- Drone and rocket have thrust, which allows for thrust vector control through thrust deflection
- Both have center of mass above the center of pressure
- Rocket has roll control handled by the reaction control system, drone also has roll control handled by its contra rotating propellers

Differences:

- Rocket is combustion powered, drone's thrust is propeller based
- Rocket has 2 axis of control on gimbal, drone currently has 1 for its first iteration, to be expanded to 2 in future work
- Size/scale difference
- Drone does not currently have variable mass to affect its moment of inertia, a common property of rockets.

Next chapter describes the rocket in more detail, as well as the considerents preceding the work on the control system.

CHAPTER 2

Spica rocket mission

This chapter describes the rocket, elements of aerodynamics stability, the type of thrust vector control for the rocket, which is to be studied and controlled. This is to understand what elements precede control and which influences are to be expected and what does the controller have to compensate for.

2.1 Spica rocket

Spica is the next rocket developed and built by Copenhagen Suborbitals. The rocket aims to be the first crewed vehicle developed by the organisation and the first crewed amateur rocket in the world. Spica will be powered by a 100 kN liquid bi-propellant engine running on liquid oxygen and ethanol. Spica will have a diameter of 0.955 m, a total height of about 13 meters and a Gross Lift Off Weight of 4000 kg of which 2600 kg will be propellant.

From the Copenhagen Suborbitals website on Spica mission description: "Spica will be launched from an upgraded Sputnik platform capable of supporting the weight and space required. With a Gross Lift-Off Weight (GLOW) of 4000 kg and a thrust of 100 kN Spica will climb out with an initial Thrust To Weight Ratio (TTWR) of 2.55. At T+20 seconds it will go super sonic and at T+90 seconds the engine will shut off at a velocity of about 3600 km/h and an altitude of 50 km. From here Spica will coast to apogee at 105 km at T+190 seconds. A few seconds later the ballute will be deployed in order to stabilize the capsule through the thin part of the atmosphere. Approximately 9 minutes after launch at an altitude of 4 km the parachutes will unfold and provide a gentle landing in the Baltic Ocean where the recovery team will be ready to secure the capsule and the astronaut."

Rockets are launched from the military firing practice area ESD 139 in the Baltic Sea, 20 km east of the Danish island of Bornholm. It spans 70×35 km, and it is opened for launch by the Royal Danish Navy for the launch time window. The Danish and Swedish authorities close the airspace for air traffic in the hours prior to launch. The mission base is the seaside town Nexø on the east coast of Bornholm, also named "Spaceport Nexø". The launch platform is MLP-Sputnik, a metal platform powered by diesel engines.

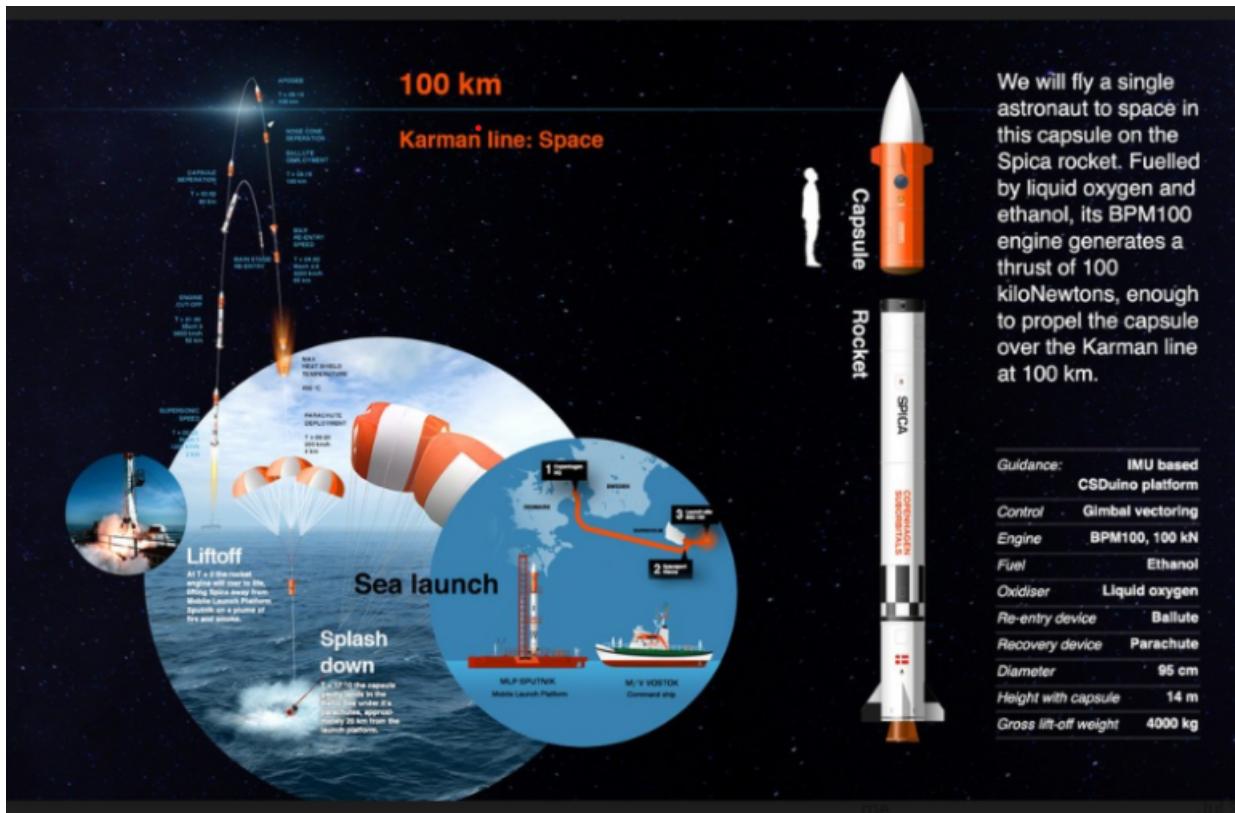


Figure 2.1: Description of Spica rocket mission .



Figure 2.2: Rockets launch site in Baltic Sea.

2.2 Rocket aerodynamics and stability

This section describes theoretical elements of rocket aerodynamics and aerial stability. The motivation is to fulfill the requirement of understanding elements affecting the flight significantly, preceding the control system design. This section aims to answer the following questions: What affects rocket aerodynamic stability? What needs to be known before thrust vector control acts?

Flight stability is important for the vehicle in order to avoid tumbling, oscillations which cause extra drag and sloshing of the liquid propellants [19]. A primary concern is to prevent large-amplitude transient oscillations. There are two components to stability: inherent stable design and active stabilisation from control systems. Inherent stability is specific to the rocket's design, that is, aerodynamic design meant to minimize drag. Such an example is the design of the nose cone and thrust alignment with body center of gravity. Fin stabilization is another important aspect of passive stability.

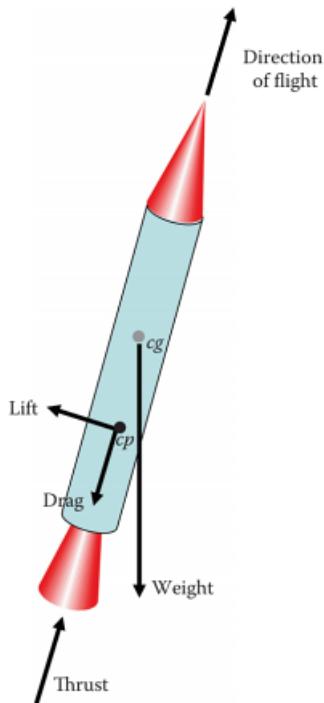


Figure 2.3: Forces acting on a rocket. Source: "Introduction to Rocket Science and Engineering" by : Travis Taylor, 2017.

The aerodynamic forces acting on a rocket in flight are thrust, weight, lift and drag, dependant on the vehicle's shape, size, angle of attack. Thrust and weight act on the rocket's center of gravity (center of mass), while lift and drag forces act on the center of pressure. One of the main challenges in stabilizing a rocket comes from the shift in rocket's center of gravity as propellant is consumed during flight. This shift is more

significant in large rockets, such as Spica, where the difference between (Hobbs, 2010) [9]. It is necessary to know the absolute position of the two centers and the position relative to each other - their distance is termed stability margin. In order for the rocket to be stable in flight, the center of gravity needs to be above center of pressure [20].

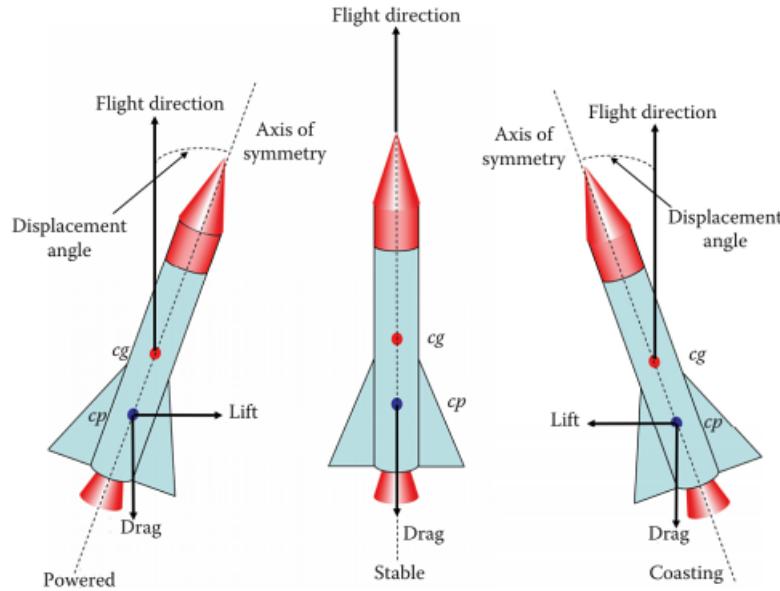


Figure 2.4: Flight angle. Source: "Introduction to Rocket Science and Engineering" by : Travis Taylor, 2017.

During powered flight, perturbations might occur which deviate the vehicle from its desired path, displacing its angle of attack. The aerodynamic forces will cause a torque about the center of gravity. During coasting (no engine thrust), the torque will be created in the opposite direction (Taylor, 2017). Having the proper stability margin, a rocket built top-heavy, allows the torques created by perturbations to self correct and stabilize the vehicle in flight - a phenomenon called restoring force. The restoring force is laminar flow of air occurring at high vehicle velocity. However, from launch and until achieving the necessary velocity, being top heavy, the rocket is highly unstable as it lifts off the launch rail [20]. It is not efficient to design a highly complex controller that can handle the large instability margins of the rocket launching right off the pad.

The preferred approach is a controller that activates once the rocket is just off its rail, with enough velocity to take advantage of the restoring force stabilization and having its controller perform small adjustments. To sum up, in flight, the rocket is subjected to disturbance torques which can be due to environmental torques, produced by the aerodynamic forces [22] or axial misalignments about the rocket's center of mass. Passive stability is necessary but insufficient for large rockets, therefore active guidance and attitude control is necessary due to the inevitability of disruptions occurring in flight. Active guidance and stabilization can be achieved by deflecting thrust so that a

rotational effect is imparted to the body in flight. The chosen attitude control type for Spica is thrust vector control through gimbaled engine.

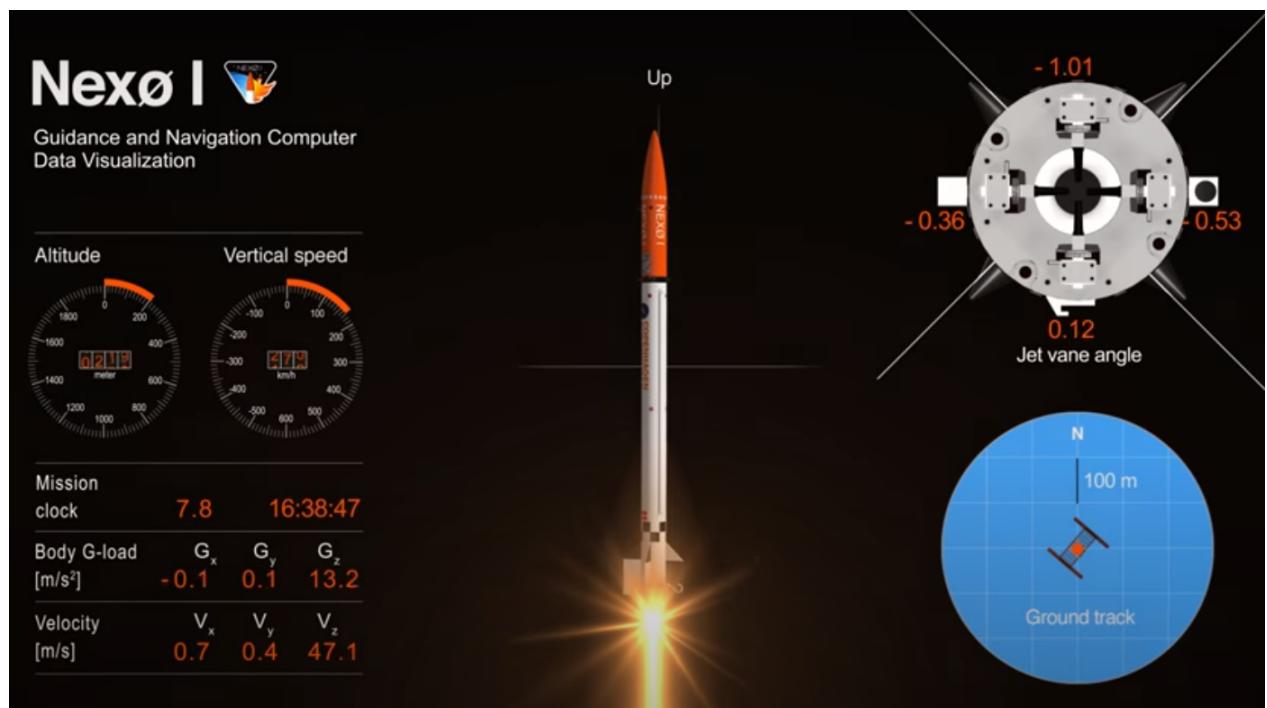


Figure 2.5: Capture of jet vanes function from a previous rocket.

2.3 Thrust vector control

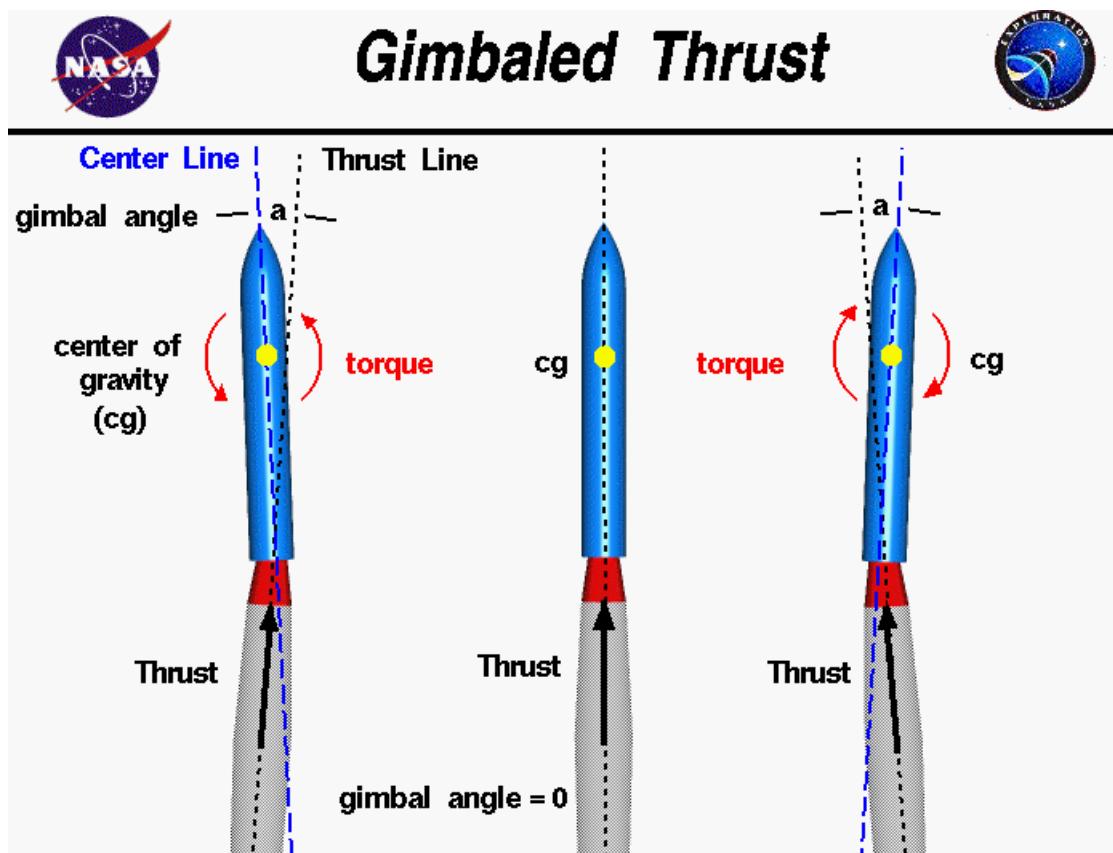


Figure 2.6: Gimbaled thrust. Source: NASA.

All actively controlled Copenhagen Suborbitals rockets have featured jet vane based attitude and control systems. Jet vanes are made of graphite and allow deflection of the flow of the nozzle. While the system had great performance in previous rockets and insured stable control in flight, the next rocket cannot use jet vanes. Due to their placement, jet vanes incur a drag (loss) of approximately 10 percent of the thrust - a significant loss for Spica-sized vehicles, which makes vanes not a feasible solution for this mission. Graphite jet vanes are challenging to produce in Denmark, with the added difficulty of producing them in the size necessary for Spica. Since jet vanes are not a suitable option for a Spica-sized rocket, there had to be a change towards another type of TVC. Out of the TVC types available for rockets, the organisation decided on gimbaled thrust. Spica will be the first of Copenhagen Suborbitals rockets to feature a gimbal engine type of thrust vector control, as opposed to the previous solution of exhaust graphite vanes (jet vanes).

Spica will employ the more efficient gimbal system where the thrust chamber will be tilted, relative to the rocket's center of gravity, to provide thrust vectoring and correct

pitch and yaw. The advantages of the gimbal solution are that there will be no loss incurred due to drag as was the case in jet vanes, due to their placement. Another advantage is that the organisation is able to manufacture the gimbal system in-house. Since this is a major design change compared to the previous rockets the gimbal system will be tested on several unmanned flights prior to the first flight of crewed Spica. The chosen TVC of gimbaled engine allows for four degrees of freedom, pitch and yaw control axes. The roll control will be handled by a separate control system, out of the scope of this project - a reaction control system consisting of air thrusters. The TVC in this project will attempt to simulate gimbal working principles, that is, deflection of thrust in order to stabilize the rocket about its center of mass, with the difference of being tested in one axis.

This chapter introduced a summary of the technical and theoretical considerations preceding the work on navigation and control. Following content will address the building process of the prototype and its constituting elements.

CHAPTER 3

UAV prototype - system description

This chapter describes the working principle, the physical prototype of the drone, its electrical electronics parts.

3.1 Overview

The prototype is a coaxial contra-rotating copter drone. For thrust, it features two propellers placed on z axis of the drone (longitudinal), rotating in the opposite direction to each other. This concept was adopted because a single propeller would introduce a moment in the frame about its axis in the opposite direction of its rotation, while having two contra rotating propellers cancel each other's axial introduced moments. The advantages of using such a machine are its reduced scale, thrust-to-weight ratio, and aerodynamic symmetry, which removes the yaw moment and side forces commonly seen in single-rotor copters. [21] The main components of the drone are the metal frame, the two propellers, a metal rudder, the Arduino and IMU.

3.2 Drone building and assembly process

The building of the drone required considerable machining and production work. It was decided on aluminium rectangular plates and four aluminium legs for the drone in order to be statically stable. The material was chosen in order to be light. The frame plates were measured and designed in CAD drawing (Figures 3.1, A.1, ??). A CNC plasma cutter custom built in the workshop cut the frame out of aluminium plates.

The plasma cut aluminium plates had holes drilled, countersunk to allow the motor screws to sit flush. For stability, they were fastened with rivets to aluminium angle profiles, which constituted the legs and the supports for the plates.

For the mounting of propellers and their motors on the plates, two green 3D printed holders that were measured to fit the motors diameter and the opening required to mount the propeller servo motor.

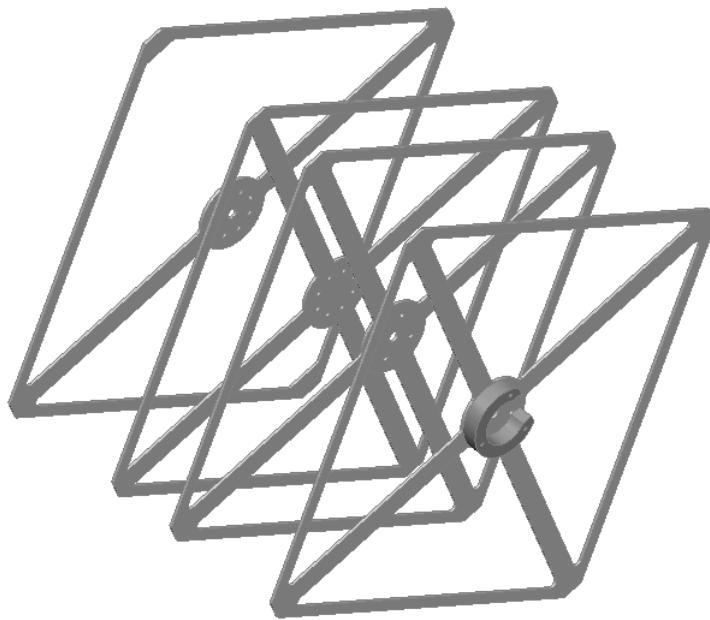


Figure 3.1: UAV frame CAD.



Figure 3.2: CNC plasma cutting of the aluminium frame.

The holders were drilled manually to avoid breaking the material and mounted such that it allowed for the correct installation and movement of the propellers, that is, allowed their servo motor (pictured) to adjust the pitch of the propellers. Once the

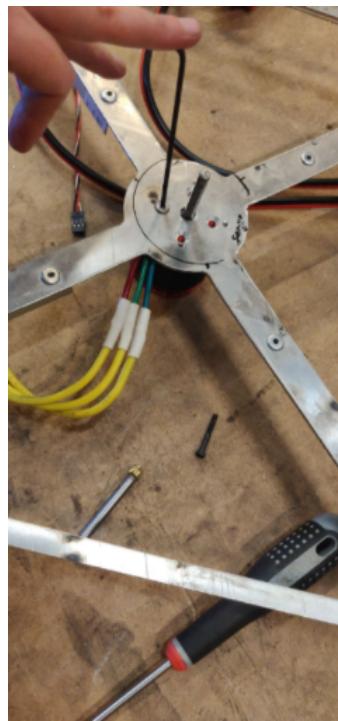


Figure 3.3: Drone assembly.

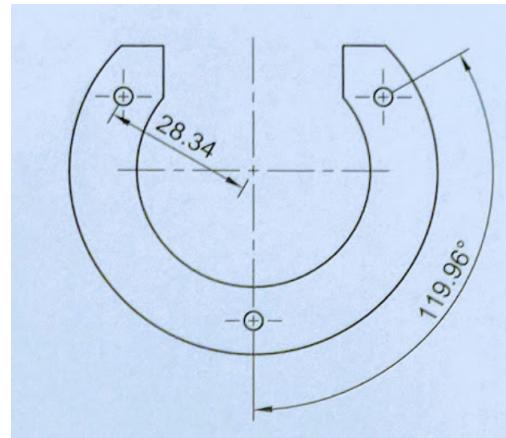


Figure 3.4: 3D printed motor holders drawing.

servo was mounted, the motors and propellers were mounted with screws on the two top levels of the aluminium frame.



Figure 3.5: 3D printed motor holders and propeller servo motor.

3.3 Electrical components

The setup has electric supply from a 12V LiPo battery, powering the propeller motors and its transmitters. There is an RC controller commanding the propellers' motors through the transmitters, with cables connecting the motors to the battery. The Arduinos, servo and the IMU are powered by a power source capable of outputting 5 V to the breadboard. This is due to the Arduinos providing insufficient power for running all these components.

3.4 Servo motor

A servo motor is used to move the rudder, actuating the control system through a middle plastic link connecting the servo arm to the rudder arm. Some behaviors have been observed in lower cost servos, which negatively affected the system performance and introduced delays or challenges for the controller. An important consideration was reducing the servo play observed in some models (backlash), that is, the servo arm not being fastened to its position by its gearbox, but allowing for a sway movement of some degrees and resulting in a deviation at the end of the servo arm.

This was problematic considering that the rudder is a heavier object that can exert a force strong enough to affect the servo's arm position, with potential of overpowering a weaker gearbox and taking advantage of this play, thus impacting the servo's true position due to its mass during actuation. This effect would introduce delays and disrupt the performance of the control system by producing an output some degrees away from the commanded one, under the force of the rudder pulling and pushing on the servo

arm.

Another observed behaviour in these servos was imprecise position control from the servo's gearbox controller, which made it noticeable that after sending a position command from Arduino, for example 90 degrees, the servo would overshoot, reaching some degrees over 90, then correcting itself to 90. This would pose a problem for the controller, as the angular position it would command would not be the one produced by the servo at the specified time, but rather first overshooting then correcting itself, adding instability and delay to the system which the controller would have to compensate for. The effect of this overshoot on the drone's pitch angle would be magnified under the force of propellers, causing unnecessary oscillations.

These behaviors were observed in different servos that have been installed and tried for the purpose, with some failing (burning out) after repetitive movement. The final choice was a Futaba servo, which did not exhibit the previously mentioned challenges and which proved to be durable, reliable, more precise and has a demonstrated history for sound aerial performance from previous rocket flights, used for actuating the exhaust vanes against the engine thrust.

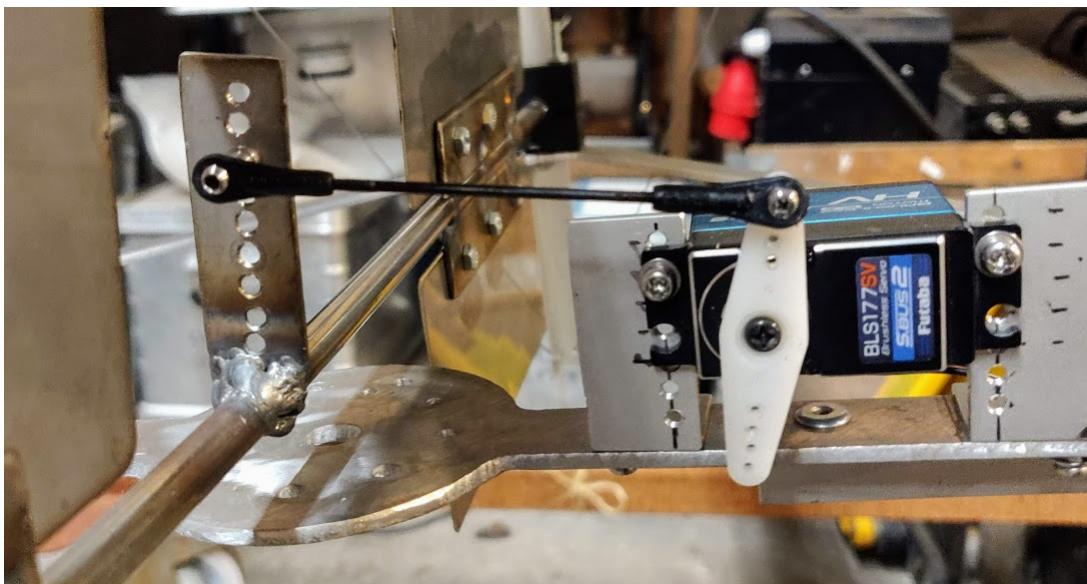


Figure 3.6: Mounted servo motor.

The mounting of the servo involved prior trial and error in order to find the position situated at the midrange of the servo. This was necessary in order to ensure roughly equal movement back and forth. In order to find the servo midrange, its position was commanded to 0 degrees, assumed to be the minimum, and to 180, assumed to be the maximum. Initially, the servo was tested at 0 degrees, however, some servos were not responding (moving) to set position to 0 degrees, while responding by moving to 90 degrees or higher values.

It was found that while some servos have their starting point from 0 degrees, it was not a general rule. The Futaba one did not respond to commands of setting position to 0 degrees, as it was discovered to have working range from 30-160 deg. Due to having a range of motion of 130 degrees, its midpoint was found at $30+130/2=95$ degrees. The servo was set to its midpoint as initial condition (starting position), then mounted on the frame for future tests.

3.5 Rudder

The rudder serves the purpose of providing thrust vector control by deflecting the flow from the propellers by an angle. It was decided to provide control in one degree of freedom only (pitch angle) as a technology demonstrator.



Figure 3.7: Welding of rudder.

A steel rod was cut to fit the drone profile and serve as mount and axis for the rudder fins. After measuring for the proper placement of the fins, four rectangular steel pieces were cut, holes machined and TIG welded on the steel rod to support the fins. TIG welding was preferred rather than MIG due to the small size of the pieces and the intent on joining them together through melting as opposed to adding extra material, to keep the mass of the rudder reduced.



Figure 3.8: TIG welding of rudder.

There was a need for the rudder to have a connection to the servo motor. For this purpose, another thin piece of metal is attached to the rod perpendicularly to act as an arm. The metal piece was designed, cut and had holes drilled 5 mm apart in order to allow for multiple placements for the servo link. The piece was then MIG welded on to the rudder.



Figure 3.9: Rudder mount and fins.

The rod was now assembled ready to be placed on the frame. For its placement, two sets of black 3D printed brackets were designed. The brackets were designed to allow the rudder to be mounted and removed easily, by having a detachable top bracket. After being produced, and drilled through, the bottom bracket was attached to the bottom aluminium plate and the top one fastens with screws on top. The rod with the supporting pieces was then mounted in the two 3D printed brackets, allowing it to move freely along its axis.

The rudder fins were the last pieces to be added to the setup. The fins were produced by having four metal plates were cut, surface polished for easier welding, then drilled,

countersunk and mounted with screws on the welded supports - subsequent to installing the rod in the frame. Thus, the rudder was completely assembled. The rudder is linked to the servo arm through a thin rigid link, detailed in the modelling section.

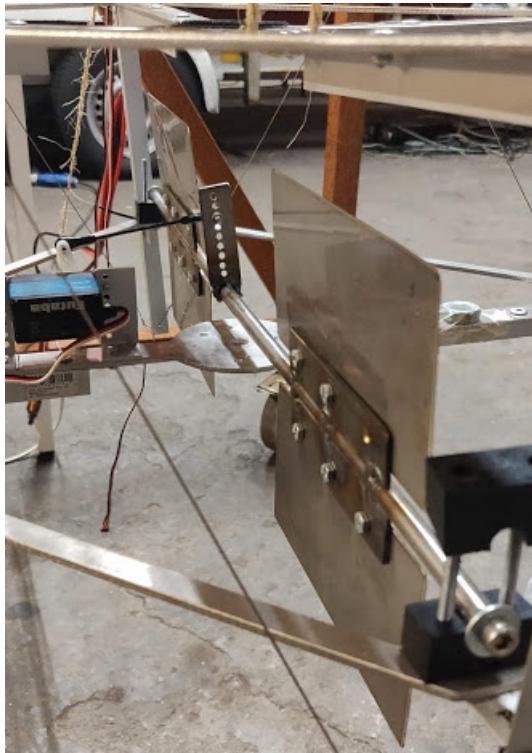


Figure 3.10: Rudder - servo motor assembly.

3.6 Electronics |IMU, Arduino

The IMU is mounted on the plate above the rudder, because that is where the center of mass is located on the drone. It sits on a breadboard, attached with Velcro adhesive which serves a double purpose: to mount the IMU in a way that allows for taking it off the drone for testing and calibration, and to dampen vibrations from the propellers when they are running. The IMU is connected to the Arduino, sitting on the table two meters away, through an extended cable. The Arduinos, IMU and servo are connected to a breadboard, powered by the power supply.

The extension cable was necessary in order to connect the IMU to the microcontroller on the table. The cable was made by removing isolation off cable ends, soldering rigid metal piece to the ends in order to ensure it stays fixed on the breadboard; then plastic separator was added to help isolate, with heat gun. The setup was tested by reading the data coming out of the IMU, to ensure continuity and correct transmission. This chapter described the constituent parts of the drone to provide an overview of the “plant”

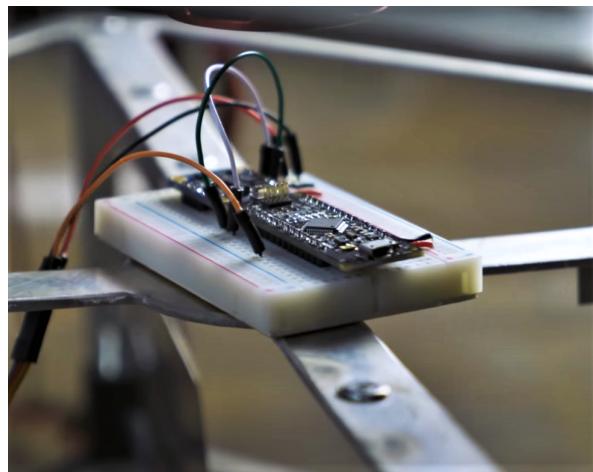


Figure 3.11: Arduino Nano.

to be controlled, along with introducing the electronics and IMU sensor. Next chapter explains the IMU sensor more in depth in connection with the Arduinos for the purpose of state estimation.



Figure 3.12: Assembled drone.

CHAPTER 4

Inertial Navigation

This chapter describes inertial navigation, as well as how the together in sensor fusion provide navigation data. Here is documented the work on calibrating the sensor, types of errors encountered.

4.1 Investigations of inertial navigation

Inertial navigation uses measurements by processing signals from accelerometers and gyroscopes to track the position and orientation of an object [23]. Navigation is an important part of controlled flight and due to errors being observed in previous rocket flights, particularly in the gyro, as it was observed that it drifts in flight. The intention in this chapter is to study and understand most sources of errors (bias, drift, etc.) that can be encountered in inertial navigation and methods for compensation.

Project requirements state the need for investigating methods of providing absolute heading (yaw) measurements to correct gyro drift. Another requirement is representing navigation attitude through a method that avoids singularities. Another requirement was investigating sources of errors in the inertial sensor, particularly the gyro drift in flight. The final requirement was to investigate filters fit for lower processing power, Arduino class micro controllers. These requirements will be addressed in this chapter.

4.2 Rotation representation

In order to fulfill the requirement concerning rotation representation, three rotation parametrizations have been considered: Euler angles, rotation matrices and quaternions. Out of the aforementioned, quaternions are chosen as the preferred method. Quaternions have the advantage of avoiding both the pitfalls of singularity of Euler angles [2] and the overparametrization of rotation matrices, of nine numbers to represent rotation which, in some circumstances, would run into gimbal lock [6].

$$\mathbf{q} = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k} \quad (4.1)$$

A quaternion (4.1) is a hyper complex number of rank 4. q_0 is the scalar part of the quaternion, with the other three units representing the vectors in ijk [8]. The navigation

calculations will be done by the algorithms in quaternions. Since the quaternion is a complex number [8], for readability, the output data will be converted from quaternions into Euler angles to facilitate translation and intuitive understanding of the result.

4.3 Heading

The system requires a drift-free or absolute heading output, which requires a constant reference point to measure against. Gyro and accelerometer, used so far in flights, provide relative heading and are insufficient for the purpose due to their lack of absolute input for the yaw (heading). With relative heading, the output can drift over time, which reduces accuracy. Having an absolute heading can help solve the gyro drift problem. There are a few choices for absolute heading: sun sensor, star trackers, horizon tracker, cameras, GPS, compass (magnetometer) [10] - out of these, the most suitable and readily available was the magnetometer. The magnetometer has the advantage of being a lightweight, low power vector sensor, measuring magnitude and direction of the magnetic field (Wertz, 2012, p. 180) [22]. GPS could have been a solution, however, here it is not to be considered for input due to previous reported experience of the GPS not maintaining fix in flight, therefore its data not being usable.

The final decision to solve absolute heading demand is to use a magnetometer, specifically a 9 DoF IMU as opposed to a discrete device. This is because the magnetometer is already available in the current and past IMUs used in the rocket flights - however, it was not used before in navigation data for these reasons: it was difficult to calibrate against the metal launch platform and the metal in the components, as well as being unclear which are the factors influencing magnetometer readings to be calibrated against. Despite challenges, it presents the most accessible option and viable option. The goal in this chapter is to investigate its sources of distortion, as well as methods to counter them. For the purpose of fulfilling the next requirement of examining sources of errors in the inertial navigation, the inertial sensor will be examined, along with its characteristic measurement disturbances.

4.4 The Inertial Measurement Unit sensor

Inertial measurement units (IMU) typically contain three orthogonal rate-gyroscopes and three orthogonal accelerometers, measuring angular velocity and linear acceleration. [23], while a MARG system (Magnetic, Angular Rate, and Gravity) is an IMU with added magnetometer. The IMU is restricted to outputting orientation measurements relative to the direction of gravity, whereas MARG systems provide orientation relative to the direction of gravity and the earth's magnetic field [13]. For simplicity, the sensors used is referred to as IMU throughout this project, despite it being a MARG system. Thus, the sensors used in this project were MEMS (microelectromechanical system) IMU consisting of tri-axis:

- Accelerometer
- Gyroscope
- Magnetometer

The accelerometer measures the external specific force acting on the sensor, consisting on both the sensor's acceleration and gravity [11]. The gyroscope measures the sensor's angular velocity, that is, the rate of change of the sensor's orientation, in the body frame. Magnetometer measures the earth's magnetic field.

4.5 IMU sensor errors

Accurate acceleration, angular rate and magnetic field measurements are essential for sensor fusion algorithms and the correct estimation of orientation. It has been noted that without error compensation, the devices experience significant distortions; the gyroscope, in particular, has a drift of 5-15 degrees/minute [3], but with proper error calibration, the output measurements can be significantly improved. For that purpose, common types of errors occurring in a MARG device will be investigated, as well as methods of compensation for such errors.

Accelerometer distortions consist of: noise, undesired high frequency oscillations; bias (offset), deviation from ideal output and vibration, read by the sensor as linear acceleration, adding to the high frequency noise. Thus, the accelerometer sensor model is described as:

$$a(t) = a_b(t) + g + b_{acc}(t) + n_{acc}(t) \quad (4.2)$$

where: $a(t)$ is the acceleration reported by the sensor, a_b is the true acceleration of the body, g is the gravitational acceleration, $b_{acc}(t)$ is the offset (bias) of the sensor and $n_{acc}(t)$ is the signal noise.

Gyroscope distortions are its bias (offset), deviation from ideal output and random drift (walk), observed in the output deviation over time due to disturbances in the sensor [12]. Therefore, the Sensor gyro model is described by the equation:

$$\omega(t) = \omega_b(t) + b_{gyro}(t) + n_{gyro}(t) \quad (4.3)$$

where: $\omega(t)$ is the angular rate reported by the sensor, ω_b is the true angular rate of the body, $b_{gyro}(t)$ is the bias of the sensor and $n_{gyro}(t)$ is the signal noise (drift). [7]

Magnetometer distortions are: hard offset, strong effects caused by manufacturing defects in the sensor or environment distortions; soft offset, produced by less pronounced magnetic objects in the sensor's environment which distort the surrounding

magnetic field; and magnetic declination, the position of the magnetic north relative to the geographic north, depending on location coordinates. Following, the magnetometer sensor model described as:

$$m(t) = m_b(t) + b_m(t) + n_m(t) + d_m \quad (4.4)$$

where: $m(t)$ is the magnetic field measurement reported by the sensor, m_b is the true magnetic field measurement of the body, $b_m(t)$ is the offset (bias) of the sensor, hard iron distortion in the case of the magnetometer and $n_m(t)$ is the signal noise. d_m is the magnetic declination, dependant on location.

Cross-coupling has been observed particularly in the lower quality IMUs, meaning some data was passed between axes of the same device (X readings read in Y) or accelerometer sending readings to the gyro (vibration being picked by the gyro, a clear error, as vibration is linear acceleration which should only be measured by the accelerometer).

4.6 IMU calibration process

Due to sensor readings distortions, pre-processing is required in order for the measurements to provide useful estimation. The calibration and alignment process has followed the same steps for each IMU device tested in this project. One of the components, usually the magnetometer, has a different set of axes and reference frame than the others and so the axes need to be consulted after reading the datasheet and the output data in order to find the reference frame and align them accordingly.

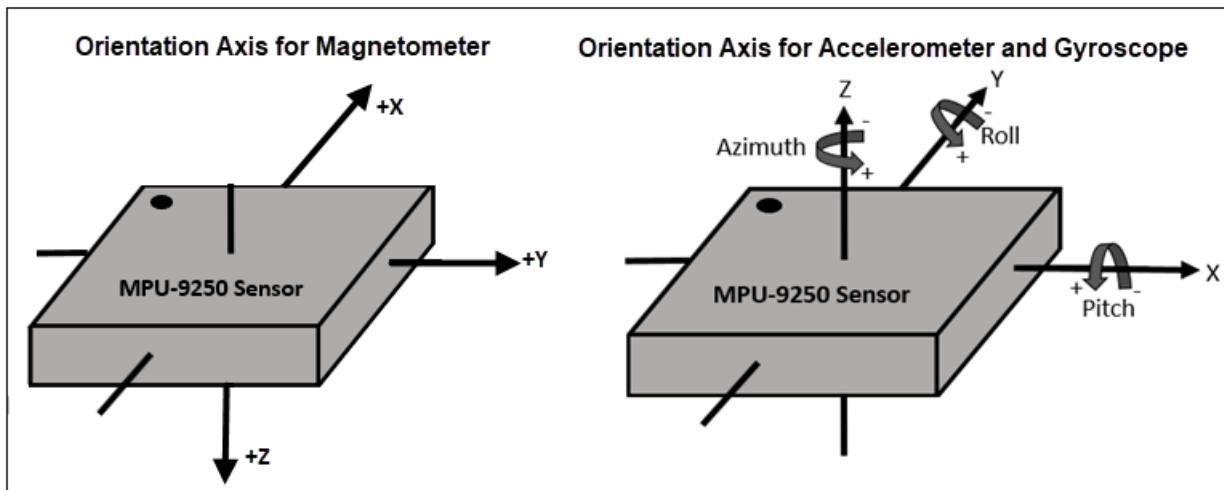


Figure 4.1: Axes difference on accelerometer-gyro and magnetometer.

In this project, the linear axes were defined as positive NED (north, east, down), and the rotational (gyro) as positive counterclockwise. Depending on the IMU and the chosen frame of reference, the accelerometer and the gyroscope axes will be swapped and/or inverted to match the magnetometer axes. For example, in the case of the MPU-9250 (figure 4.1), in order to align to NED convention, the the x and y axes of the accelerometer-gyro were swapped and the z axis was inverted, which is in alignment with the existing axes of the magnetometer device.

Then it is necessary to determine polarity values for the accelerometer, and gyroscope, by reading their data and determining which is the positive and negative direction of each axis. The accelerometer polarity can be determined by reading the gravity values, whether positive or negative. The gyroscope readings display whether clockwise or counterclockwise positive for each axis and can be inverted in case of non-alignment with the chosen convention, in this case, counterclockwise positive. The rotation axes were defined as [5]: roll is the rotation about the x axis, pitch is the rotation about the y axis, yaw is the rotation about the z axis.

Calibration of the sensor is important before fusing, filtering and using the output. The calibration process of the accelerometer, gyroscope and magnetometer has some similarities, in the sense that all three require the removal of offsets (biases); then the noise, drift, and vibration characteristic to each component is handled by the sensor fusion filters chosen, described below.

Accelerometer offsets are removed by reading the gravity vector in each axis while on a flat surface and compensating accordingly until the reading shows 9.8 or 1g, depending on the chosen unit of measurement. **Gyroscope** bias is removed in a similar fashion, by what is known as the tombstone test, where the sensor is placed flat on a surface, and compensations are applied to level the values to 0. **Magnetometer** is affected by hard iron effects, causing deviation from origin and soft iron offsets, causing skew, stretching the magnetic sphere towards an ellipse shape. In order to determine the offsets, the magnetometer must be moved in a eight-shape or to cover as many orientations as possible during the measurements, to fully capture environment distortion. Figure 4.2 shows the deviation caused by hard iron effects and the ideal sphere that represents calibrated magnetometer data.

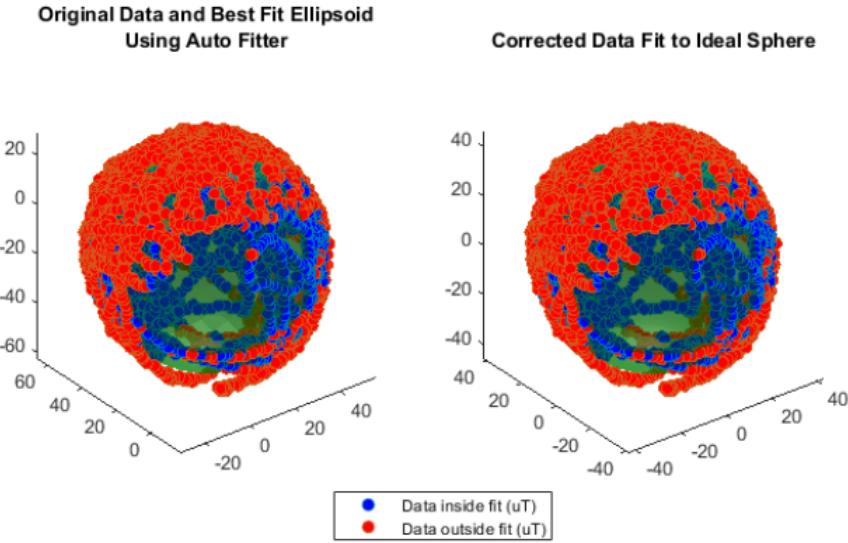


Figure 4.2: Magnetometer samples fit to ideal sphere.

Figure 4.5 shows the measurements from figure 4.2, as example of hard offset calibration, with little noticeable skew from soft iron effects.

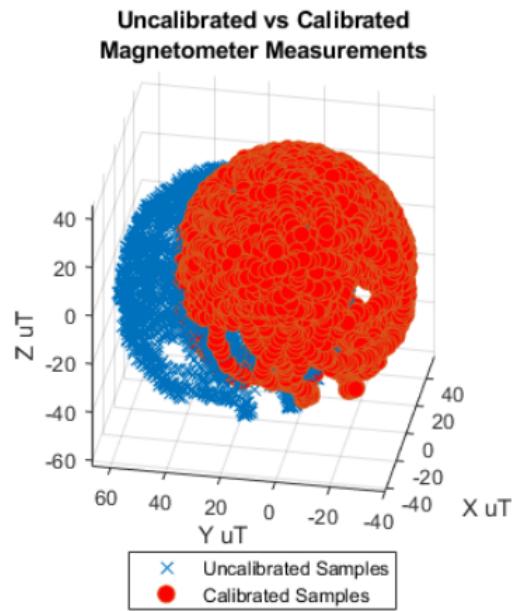


Figure 4.3: Hard offset effects calibration.

This process has been performed on each IMU tested. Figures 4.4, 4.5 and 4.2 show data collected from an IMU during calibration process. In figure 4.4, hard iron the offset

can be observed in the uncalibrated readings by the deviation from (0,0,0) origin; the soft iron effect can be noticed in the ellipsoid shape of the samples, as opposed to a spherical shape, expected in the case of calibrated data.

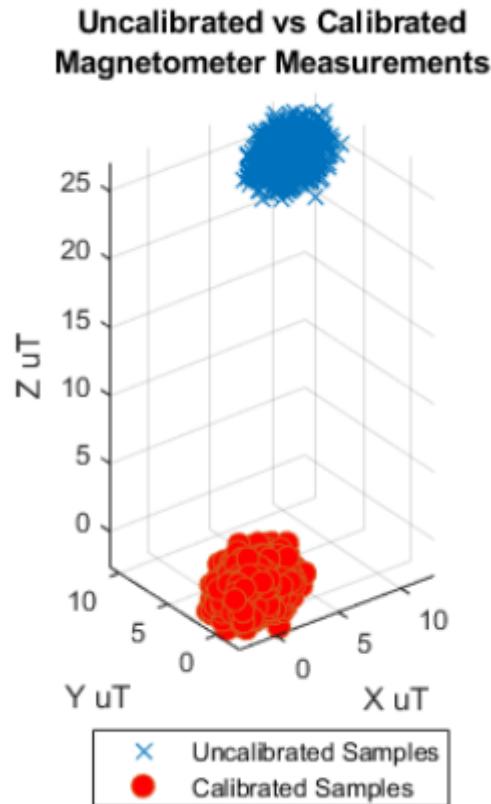


Figure 4.4: Strong effects of hard iron and soft iron distortions in uncalibrated magnetometer measurements vs hard iron calibrated measurements.

These figures have covered the cases of offset distortion in magnetometer readings. The last distortion experienced by the magnetometer, magnetic declination, is to be determined based on location - in Copenhagen, Denmark, the declination is $+4.37^\circ$, which is added as a correction to the heading readings. Since the magnetometer picks up magnetic fields, any slight change in the environment requires a new calibration.

4.7 Sensor fusion

The calibrated measurements of the sensor are not directly usable, due to sensor noise, picking up vibration or other mechanical disturbances even when static. [14] After calibration of the sensor, the measurements can be combined through sensor filtering

to produce more accurate, usable estimation of position and orientation. The main consideration for choosing filters has been their efficiency and computational cost.

The following filters have been tested in this project:

- Complementary
- Mahony
- Madgwick

The complementary filter is a common, fast, efficient solution for inertial measurements. It acts by low-pass filtering accelerometer data, and high-pass filtering by direct integration of gyroscope data, then fusing these estimates together [7]. It uses the fact that the measurements from accelerometer and magnetometer are noisy yet steady over time, while gyroscope measurements are accurate in short periods, but drift over time. It results that in frequency domain, the gyroscope has desirable properties at high frequencies, therefore it requires a high-pass filter; while accelerometer and magnetometer have useful properties at low frequencies, which can be exploited through a low-pass filter. [11]. By combining the advantages of each group, they complement for each other's deficiencies.

The following two are complementary-based filters. Mahony or Explicit Complementary Filter, is a non-linear complementary filter, designed specifically for low-cost inertial measurement units and capable of achieving the same results as an extended Kalman filter with GPS and inertial data. This filter has been validated only for inertial data from accelerometer-gyro. [7] Madgwick filter has been proven to produce similar to [4] or improved accuracy over Kalman-based algorithms [13], with the advantage of having low computation cost, suitable for Arduino class micro-controllers. Another advantage over Mahony filter is its suitability for IMU and MARG systems, therefore, including magnetometers.

Madgwick filter has a tunable parameter, β , related to the gyroscope measurement error. A lower value for the parameter will decrease the influence of the accelerometer and magnetometer, allowing the gyroscope measurements to have a higher weight[13]. This project uses the library implementation of [17] for Mahony and Madgwick filters. There was testing with the value of β for Madgwick, initially tested with 0.1 and with values ranging to 0.9. The best outcome was found with the value of 0.5, as it converges quicker than having lower values and it balances the drift from the gyroscope. However, this is assuming usable, calibrated measurements from the accelerometer and magnetometer.

4.8 Hardware used: IMU and Arduino

There have been several IMUs tried over the course of this project, from the least quality to highest:

- GY-521
- MPU6050
- Razor 9 DoF
- LSM9DS1
- BNO055

First two IMUs were 6 DoF (accelerometer and gyro), without magnetometer, which was insufficient for the purpose of project after deciding on magnetometer as absolute heading sensor. In addition, their gyroscopes were picking up vibration, which signifies axes and device cross-talk - an error, as the gyro should not pick up vibration, since it is linear acceleration. Next two devices were 9 DoF sensors, with different challenges due to their magnetometer in particular. However, the primary challenge was they turned out overall unsuitable for aerial devices, as the noise was significant, it required extensive calibration and filtering, which introduced serious delays. The final choice was BNO055, which proved to be the most reliable device out of the list and suitable for the purpose.

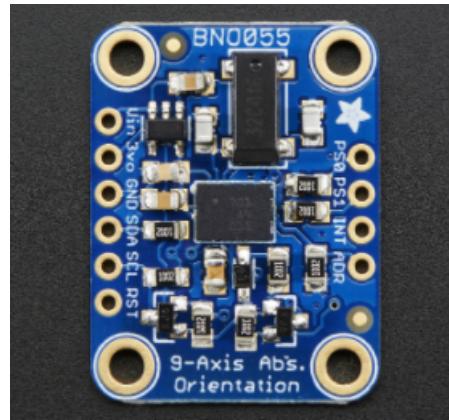


Figure 4.5: BNO055 IMU sensor.

Several Arduino class microcontrollers were used, due to their different capabilities: Mega, Uno, Nano, as well as a variant of Nano with SAMD21 processor. The processing power of the Nano, atm328, turned out to be insufficient for the Madgwick filter so the original Nano was replaced it with a SAMD21 processor version of the Nano. The final code was settled on Nano/SAMD21 because of the superior processor, smaller size, made

with code for Madgwick/Mahony/complementary filter plus a PID controller. However, since there was intent to use Simulink, which does not support the Nano/SAMD21, another Arduino Mega is used for the code generated from Simulink.

This chapter discussed the different types of challenges met in inertial navigation, for the purpose of investigating and finding ways of countering them to ensure accurate navigation. The next chapters will discuss the modelling and control of the system.

CHAPTER 5

Attitude dynamics: modelling and control

This chapter introduces the dynamics modelling principles used for this project, the reason for model-based approach and other modelling decisions, as well as the control system as following the previous chapters

5.1 Model-based design approach

The motivation for model-based design of the controller is that this can be useful on a system even before the physical system exists, as it is the case for the Spica rocket. Model-based tuned controller can be more efficient, more precise and less error prone, therefore requiring less controller corrections needed, which translates into more precise movements and conservation of energy and resources. Model-based also implies that scalability towards the rocket is inherent.

In this sense it is useful to have an abstraction of the physical system, which can be very complex and would be overwhelming to deal with all its structural parts. It is easier to make changes but also a model-based approach allows the controller to be more precise and suited for that particular type of system.

The approach chosen in this thesis is derivation of the dynamics model from first principles. This approach provides understanding of all aspects of the model - as opposed to choosing system identification from empirical data. This would run the risk of having an oversimplified model. The model is then validated through empirical data - from tests - and also as a means to determine its parameters for accuracy of the model. The model's aim is not a high degree of accuracy but having a reliable starting point. This model would be used for analysis for the system as opposed to design.

Having a model helps map more precisely the input to output and to see how the different parameters (inertia etc) affect the response of the system. The model does not include an important part: the point of saturation of the actuators of components. Another simplification consists of the model assuming its parameters are time invariant. They may in fact change with temperature etc. It is necessary to take into consideration actuator delays and other interactions between the links of the rocket; it is not a static

system, it's dynamic: it will take time to react. It varies with time, depending on initial conditions and past inputs.

The more complex model which includes the non-linear, time-varying parameters can be tested through simulation, achieving higher accuracy. Simulation allows for varying different parameters and observing their response in real time. Since there is interest in the transient response of the system as opposed to purely steady-state response, the system will be modeled as a dynamic system.

5.2 Model: kinematics and dynamics - Newton-Euler approach for rigid multi-body systems

The aim of this section is to analyze the kinematic and dynamic behavior of the dual propeller system described above. For this project, the approach is Newton-Euler for multi-body systems, where bodies are separated at the joints, each body considered an individual unit, considered rigid bodies. This modelling strategy used in this project section was introduced in (Santos, 2003) [16].

Newton-Euler was preferred as opposed to Lagrange due to the focus being on understanding the interplay of forces as opposed to the energy in the mechanical system, relevant for the drone as well as for the rocket. The mechanical system consists of a static structure (drone frame, propellers) and three moving rigid bodies, approached in 2D. The moving bodies form the servo-rudder system, with three bodies and one degree of freedom, the servo arm angle.

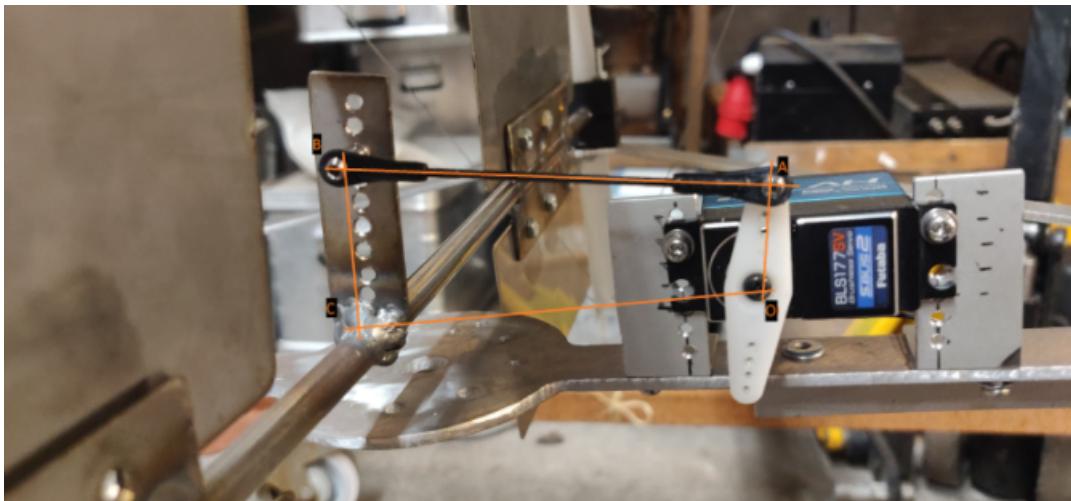


Figure 5.1: The servo-rudder mechanical system.

Nomenclature

Symbol	Meaning
a	absolute acceleration
t	time
m	mass
g	acceleration due to gravity
θ	angular displacement (scalar)
$\dot{\theta}$	angular velocity (scalar)
$\ddot{\theta}$	angular acceleration (scalar)
ω	angular velocity (vector)
CM	center of mass
r	position vector
F	force
B_x	body number x
T	transformation matrix
I	moment of inertia

Parameters List

Parameters List			
Symbol	Parameter	Quantity	Units
m_0	mass body 0	2.75	kg
m_1	mass body 1	0.05	kg
m_2	mass body 2	0.00005	kg
m_3	mass body 3	0.54	kg
I_{zz0}	rotational inertia about Z axis body 0	0.1	$kg \times m^2$
I_{zz1}	rotational inertia about Z axis body 1	0.625	$kg \times m^2$
I_{zz2}	rotational inertia about Z axis body 2	0.2408×10^{-25}	$kg \times m^2$
I_{zz3}	rotational inertia about Z axis body 3	0.0022	$kg \times m^2$
F_{prop}	thrust force	30	N
g	acceleration due to gravity	9.8	m/s^2

The prototype is modelled as a fixed-axis rotation body. The dynamical equations of motion and the dynamic reaction forces are derived. Translational equations are written in the inertial frame, rotational equations are written in the moving body frame. Defining the inertial (**I**) and the moving coordinate systems for the bodies (**B0**, **B1**, **B2**, **B3**) in this mechanism and the free-body diagram of the mechanical system defining each of the three bodies and the interplay of reaction forces (Santos, 2003) [16]:

5.2.1 Reference frames

I - inertial X, Y, Z; **i**, **j**, **k**

B0- moving body 0: **i0**, **j0**, **k0**

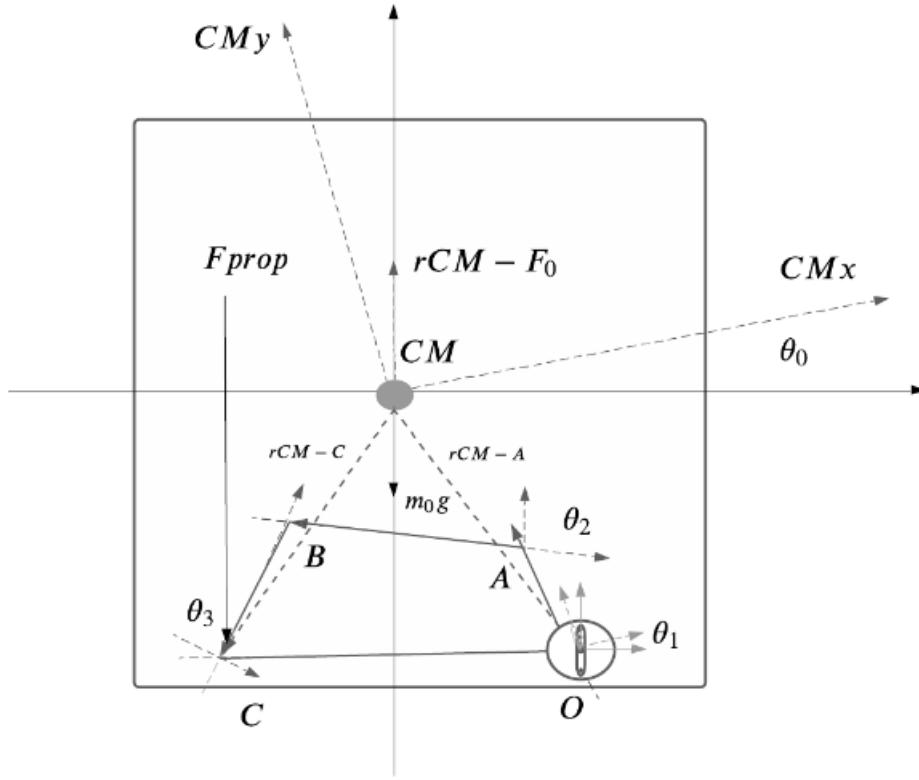


Figure 5.2: Free-body diagram of the drone body frame.

B1- moving body 1: $\mathbf{i}_1, \mathbf{j}_1, \mathbf{k}_1$

B2- moving body 2: $\mathbf{i}_2, \mathbf{j}_2, \mathbf{k}_2$

B3- moving body 3: $\mathbf{i}_3, \mathbf{j}_3, \mathbf{k}_3$

Defining the transformation matrix from the inertial coordinate system (**I**) to body coordinate systems (**B1**, **B2**, **B3**) [16].

$$T_\theta = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.1)$$

Defining the transformation from inertial frame to body frame as

$${}_B r = T_{\theta I} r \quad (5.2)$$

and from body frame to inertial frame as

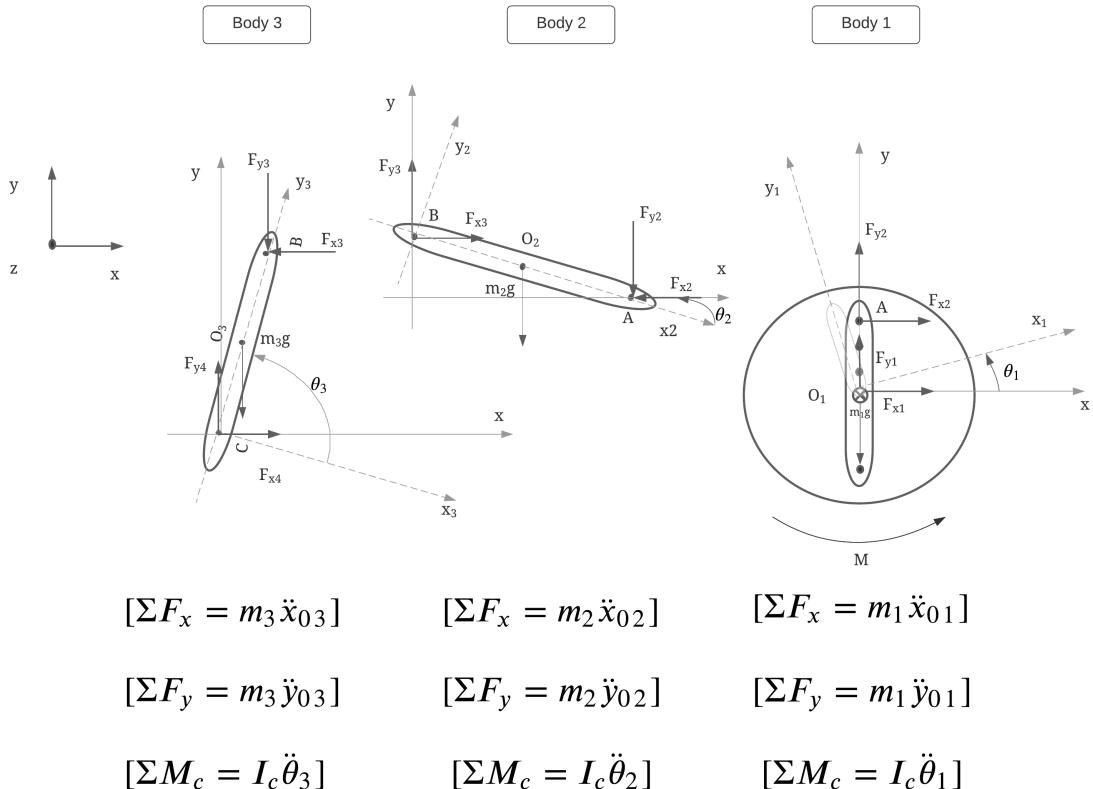


Figure 5.3: Free-body diagram of the servo-rudder mechanical system.

$${}_I r = T_{\theta} {}_B r \quad (5.3)$$

Writing the displacement vectors for each of the three bodies of the mechanism in the body frame and the distance from point C to O, closing the constraint equation, in the inertial frame

$${}_{B1} \vec{r}_{OA} = \begin{bmatrix} 0 \\ 0.02 \\ 0 \end{bmatrix} [m] \quad (5.4)$$

$${}_{B2} \vec{r}_{AB} = \begin{bmatrix} -0.08 \\ 0 \\ 0 \end{bmatrix} [m] \quad (5.5)$$

$$B_3 \vec{r}_{BC} = \begin{bmatrix} 0 \\ -0.036 \\ 0 \end{bmatrix} [m] \quad (5.6)$$

$$I \vec{r}_{CO} = \begin{bmatrix} 0.078 \\ 0 \\ 0 \end{bmatrix} [m] \quad (5.7)$$

The individual bodies are presented with their properties: mass, mass moment of inertia and displacement vector from the center of mass. The bodies are thin, with negligible thickness and their mass moment of inertia was calculated with the bodies modeled as rods.

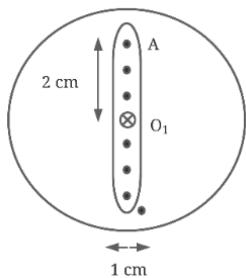


Figure 5.4: Servo.

Body 1: Servo arm, rotating about its center pivot. The servo arm has a length of 0.02 m from pivot point to end of servo arm, coinciding with center of mass of the body, mass = 0.003 kg and a calculated mass moment of inertia around Z axis of $I_{zz1} = 6.25e-7 \text{ kg/m}^2$. The servo arm has negligible thickness.

Body 2 has mass of $m = 0.001 \text{ kg}$ and mass moment of inertia around Z axis of $I_{zz2} = 2.408333e-15 \text{ kg/m}^2$. The center of mass of the body AB is at the middle of the length between the points.

The link has negligible thickness.

Figure 5.4: Servo.

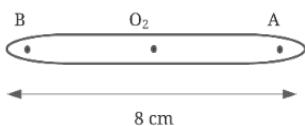
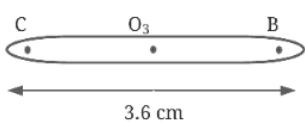


Figure 5.5: Link.

Body 3: rudder has a mass $m = 0.54 \text{ kg}$ and mass moment of inertia around Z axis of $I_{zz3} = 0.0022 \text{ kg/m}^2$. It has its center of mass at the point of contact with the rudder (labeled C)

Further the connection points between the three bodies

Figure 5.6: Rudder arm. are shown, forming the constraint equation.



$$I \vec{r}_{0A} + I \vec{r}_{AB} + I \vec{r}_{BC} + I \vec{r}_{C0} = 0 \quad (5.8)$$

The constraint equation is formed by having the displacement vectors transformed from body frame to inertial frame and equaled to 0.

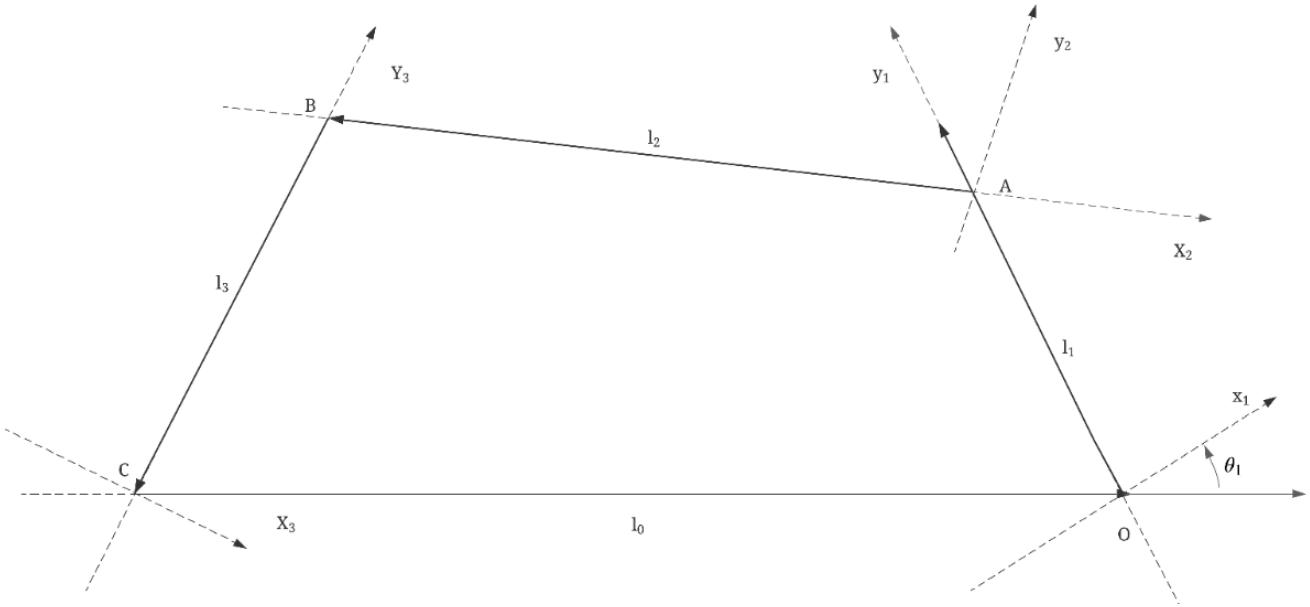


Figure 5.7: Constraint Figure.

$${}_I\vec{r}_{0A} + {}_I\vec{r}_{AB} + {}_I\vec{r}_{BC} + {}_I\vec{r}_{C0} = 0 \Rightarrow T_{\theta_1 B}^T \vec{r}_{0A} + T_{\theta_2 B}^T \vec{r}_{AB} + T_{\theta_3 B}^T \vec{r}_{BC} + {}_I\vec{r}_{C0} = 0 \quad (5.9)$$

Therefore, results the following constraint equation in inertial reference frame.

$$\begin{bmatrix} -0.02 \sin \theta_1 \\ 0.02 \cos \theta_1 \\ 0.0 \end{bmatrix} + \begin{bmatrix} -0.08 \cos \theta_2 \\ -0.08 \sin \theta_2 \\ 0.0 \end{bmatrix} + \begin{bmatrix} 0.036 \sin \theta_3 \\ -0.036 \cos \theta_3 \\ 0.0 \end{bmatrix} + \begin{bmatrix} 0.078 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (5.10)$$

The coordinate θ_1 [rad] is the master coordinate which describes the servo's angular position (body 1). θ_2 [rad] and θ_3 [rad] are “slave” coordinates, responsible for describing the angular position of bodies 2 and 3 depending on the motion of body 1, therefore θ_2 and θ_3 are described as a function of θ_1 . Results a system composed of two nonlinear algebraic equations which will be numerically solved through a method capable of solving for roots of non-linear equations. In this project, the Newton-Raphson method has been used to iteratively arrive at the roots, describing the two slave coordinates. By differentiating the constraint equation (5.8), results the angular velocity of the bodies. By using Cramer’s rule, results the angular velocities $\dot{\theta}_2$ and $\dot{\theta}_3$ as a function of the servo angular rate $\dot{\theta}_1$

$$\dot{\theta}_2 = \frac{0.00072 \left(\frac{d}{dt} \theta_1(t) \right) \cos(\theta_1(t)) \sin(\theta_3(t)) - 0.00072 \cos(\theta_3(t)) \left(\frac{d}{dt} \theta_1(t) \right) \sin(\theta_1(t))}{0.00288 \sin(\theta_2(t)) \sin(\theta_3(t)) + 0.00288 \cos(\theta_3(t)) \cos(\theta_2(t))} \quad (5.11)$$

$$\dot{\theta}_3 = \frac{0.0016 \sin(\theta_2(t)) \left(\frac{d}{dt} \theta_1(t) \right) \sin(\theta_1(t)) + 0.0016 \left(\frac{d}{dt} \theta_1(t) \right) \cos(\theta_1(t)) \cos(\theta_2(t))}{0.00288 \sin(\theta_2(t)) \sin(\theta_3(t)) + 0.00288 \cos(\theta_3(t)) \cos(\theta_2(t))} \quad (5.12)$$

Differentiating the constraint equation (5.8) twice, results the angular acceleration of bodies of bodies 2 and 3. By using Cramer's rule, results the angular acceleration $\ddot{\theta}_2$ and $\ddot{\theta}_3$ as a function of the servo angular acceleration $\ddot{\theta}_1$. The equations are larger than can be displayed here, therefore they are available in scaled down form in the appendix, (A.1) and (A.2).

5.2.2 Absolute linear acceleration

The next step is calculating the absolute linear acceleration vectors of the center of mass of the moving bodies. The absolute linear acceleration at the center of mass of each body is composed of normal and tangential acceleration.

The absolute velocity vector is defined as the derivative in time of the position vector and the absolute acceleration vector is the derivative in time of the position vector.

The calculations are done in inertial frame, in order to obtain the absolute position vectors and their absolute derivatives, without losing information. [16].

Body 1 - Servo

$${}_I\vec{a}_1^* = {}_I\vec{a}_0 + {}_I\vec{\omega}_1 \times ({}_I\vec{\omega}_1 \times {}_I\vec{r}_{O-CM}) + \underbrace{{}_I\dot{\vec{\omega}}_1 \times {}_I\vec{r}_{O-CM}}_{\text{coriolis}=0} + \underbrace{2{}_I\vec{\omega}_1 \times {}_I\vec{v}_{rel}}_{=0} + {}_I\vec{a}_{rel} \quad (5.13)$$

where

$${}_{B1}\vec{r}_{O-CM} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad {}_I\vec{\omega}_1 = \begin{bmatrix} 0 \\ 0 \\ \frac{d}{dt} \theta_1(t) \end{bmatrix} \quad {}_I\dot{\vec{\omega}}_1 = \begin{bmatrix} 0 \\ 0 \\ \frac{d^2}{dt^2} \theta_1(t) \end{bmatrix}$$

Resulting in 0 absolute acceleration of the body 1 due to point O coinciding with the center of mass of body 1 (servo) and no translational motion in body 1. **This is valid**

in the case of static drone, however it would change with the addition of the 3 DoF of the drone body.

Body 2 - Link

Using the equation for absolute acceleration around the center of mass of body 2

$${}_I\vec{a}_2^* = {}_I\vec{a}_A + {}_I\vec{\omega}_2 \times ({}_I\vec{\omega}_2 \times {}_I\vec{r}_{A-CM}) + {}_I\dot{\vec{\omega}}_2 \times {}_I\vec{r}_{A-CM} \quad (5.14)$$

$${}_I\vec{a}_A = {}_I\vec{\omega}_1 \times ({}_I\vec{\omega}_1 \times {}_I\vec{r}_{OA}) + {}_I\dot{\vec{\omega}}_1 \times {}_I\vec{r}_{OA} \quad (5.15)$$

where:

$${}_{B1}\vec{r}_{OA} = \begin{bmatrix} -0.02 \sin \theta_1 \\ 0.02 \cos \theta_1 \\ 0.0 \end{bmatrix} \quad {}_{B2}\vec{r}_{A-CM} = \begin{bmatrix} -0.04 \cos \theta_2 \\ -0.04 \sin \theta_2 \\ 0.0 \end{bmatrix} \quad {}_I\vec{\omega}_2 = \begin{bmatrix} 0 \\ 0 \\ \frac{d}{dt}\theta_2(t) \end{bmatrix} \quad {}_I\dot{\vec{\omega}}_2 = \begin{bmatrix} 0 \\ 0 \\ \frac{d^2}{dt^2}\theta_2(t) \end{bmatrix}$$

Results the following:

$${}_I\vec{a}_2^* = \begin{bmatrix} 0.02 \left(\frac{d}{dt}\theta_1(t)\right)^2 \sin \theta_1 - 0.02 \left(\frac{d^2}{dt^2}\theta_1(t)\right) \cos \theta_1 + 0.04 \left(\frac{d}{dt}\theta_2(t)\right)^2 \cos \theta_2 + 0.04 \left(\frac{d^2}{dt^2}\theta_2(t)\right) \sin \theta_2 \\ -0.02 \left(\frac{d}{dt}\theta_1(t)\right)^2 \cos \theta_1 - 0.02 \left(\frac{d^2}{dt^2}\theta_1(t)\right) \sin \theta_1 + 0.04 \left(\frac{d}{dt}\theta_2(t)\right)^2 \sin \theta_2 - 0.04 \left(\frac{d^2}{dt^2}\theta_2(t)\right) \cos \theta_2 \\ 0.0 \end{bmatrix} \quad (5.16)$$

Body 3 - Rudder

Finally, following the same procedure for body 3

$${}_I\vec{a}_3^* = {}_I\vec{a}_C + {}_I\vec{\omega}_3 \times ({}_I\vec{\omega}_3 \times {}_I\vec{r}_{C-CM}) + {}_I\dot{\vec{\omega}}_3 \times {}_I\vec{r}_{C-CM} \quad (5.17)$$

where

$${}_{B3}\vec{r}_{C-CM} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad {}_I\vec{\omega}_3 = \begin{bmatrix} 0 \\ 0 \\ \frac{d}{dt}\theta_3(t) \end{bmatrix} \quad {}_I\dot{\vec{\omega}}_3 = \begin{bmatrix} 0 \\ 0 \\ \frac{d^2}{dt^2}\theta_3(t) \end{bmatrix}$$

Which results in 0 linear acceleration for body 3 (rudder) since point C coincides with the center of mass of body 3 and there is no translational movement in body 3. This concludes the kinematics section of the model. The next section models the dynamics of the system.

5.2.3 Dynamic equilibrium (Newton-Euler)

Deriving the dynamics of the system, summing all forces and moments acting on the bodies

$$\begin{array}{lll} [\Sigma F_x = m_3 \ddot{x}_{03}] & [\Sigma F_y = m_3 \ddot{y}_{03}] & [\Sigma M_c = I_c \ddot{\theta}_3] \\ [\Sigma F_x = m_2 \ddot{x}_{02}] & [\Sigma F_y = m_2 \ddot{y}_{02}] & [\Sigma M_c = I_c \ddot{\theta}_2] \\ [\Sigma F_x = m_1 \ddot{x}_{01}] & [\Sigma F_y = m_1 \ddot{y}_{01}] & [\Sigma M_c = I_c \ddot{\theta}_1] \end{array}$$

Following the Newton-Euler method for achieving dynamical equilibrium between the rigid bodies, with the forces and reaction forces exerted on each body equal to mass times the absolute acceleration at the center of mass described in (5.13), (5.14) and (5.17):

Newton - Body 1

$$\begin{bmatrix} Ox \\ Oy \\ 0 \end{bmatrix} + \begin{bmatrix} Ax \\ Ay \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ -m_1 g \\ 0 \end{bmatrix} = m_{1I} \vec{a}_1^* \quad (5.18)$$

Newton - Body 2

$$\begin{bmatrix} -Ax \\ -Ay \\ 0 \end{bmatrix} + \begin{bmatrix} Bx \\ By \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ -m_2 g \\ 0 \end{bmatrix} = m_{2I} \vec{a}_2^* \quad (5.19)$$

Newton - Body 3

$$\begin{bmatrix} -Bx \\ -By \\ 0 \end{bmatrix} + \begin{bmatrix} Cx \\ Cy \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ -m_3 g \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ -F_{prop_1} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ -F_{prop_2} \\ 0 \end{bmatrix} = m_{3I} \vec{a}_3^* \quad (5.20)$$

Body 1 (servo) and 2 (link) have only the forces and reaction forces in x and y acting on each of their two respective connection points plus their weight, while body 3 (rudder) has the forces acting from the two propellers. Following, the rotational equations for each body, in the body frame:

Euler - Body 1

$${}_{B1}r_{CM-A} \times (T_{\theta1} {}_I A) + {}_{B1}r_{CM-O} \times (T_{\theta1} {}_I O) + {}_{B1}T = I \frac{d}{dt}({}_{B1}\omega) \quad (5.21)$$

where

$${}_{B1}r_{CM-O} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad {}_{B1}r_{CM-A} = \begin{bmatrix} 0 \\ 0.02 \\ 0 \end{bmatrix} \quad {}_{B1}T = \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix}$$

Resulting in :

$$0(-Ox \cos\theta_1 - Oy \sin\theta_1) - 0.02(Ax \cos\theta_1 + Ay \sin\theta_1) + T = Iz z_1 \ddot{\theta}_1 \quad (5.22)$$

Euler - Body 2

$${}_{B2}r_{CM-A} \times (-T_{\theta2} {}_I A) + {}_{B2}r_{CM-B} \times (T_{\theta2} {}_I B) = I \frac{d}{dt}({}_{B2}\omega) \quad (5.23)$$

where

$${}_{B2}r_{CM-A} = \begin{bmatrix} 0.04 \\ 0 \\ 0 \end{bmatrix} \quad {}_{B2}r_{CM-B} = \begin{bmatrix} -0.04 \\ 0 \\ 0 \end{bmatrix}$$

Resulting in :

$$0.04(Ax \sin\theta_2 - Ay \cos\theta_2) + 0.04(Bx \sin\theta_2 - By \cos\theta_2) = Iz z_2 \ddot{\theta}_2 \quad (5.24)$$

Euler - Body 3

$$\begin{aligned} {}_{B3}r_{CM-B} \times (-T_{\theta3} {}_I B) + {}_{B3}r_{CM-C} \times (T_{\theta3} {}_I C) + {}_{B3}r_{CM-Fprop1} \times (-T_{\theta3} {}_I F_{prop1}) \\ + {}_{B3}r_{CM-Fprop2} \times (-T_{\theta3} {}_I F_{prop2}) = I \frac{d}{dt}({}_{B3}\omega) \end{aligned} \quad (5.25)$$

where

$$B3r_{CM-B} = \begin{bmatrix} 0 \\ 0.036 \\ 0 \end{bmatrix} \quad B3r_{CM-C} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad B3r_{CM-F_{prop1}} = \begin{bmatrix} 0 \\ -0.055 \\ 0 \end{bmatrix} \quad B3r_{CM-F_{prop2}} = \begin{bmatrix} 0 \\ -0.035 \\ 0 \end{bmatrix}$$

Resulting in :

$$0.036(Bx \cos\theta_3 + By \sin\theta_3) - 0(Cx \cos\theta_3 + Cy \sin\theta_3) + 0.055 F_{prop1} \sin(\theta_3) + 0.035 F_{prop2} \sin(\theta_3) = Izz_3 \ddot{\theta}_3 \quad (5.26)$$

The dynamical equilibrium can now be represented in matrix form as (5.28), by writing Newton equations (5.18), (5.19), (5.20), Euler equations (5.22), (5.24), (5.26), along with the second derivative of the constraint equation, into a system of order 11, composed of 9 equations from dynamical equilibrium and 2 from constraint equations. (Santos, 2003) [16]

$$A_{s11x11} \cdot x_{s11x1} = b_{s11x1} \text{ where } x_s = \{O_x \ O_y \ A_x \ A_y \ B_x \ B_y \ C_x \ C_y \ \ddot{\theta}_1 \ \ddot{\theta}_3 \ \ddot{\theta}_3\}^T \quad (5.27)$$

From matrices A (5.28), x (5.29) and b (5.30), through Cramer's rule, can be extracted $\ddot{\theta}_1$, which is the plant to be controlled through the control system.

Matrix A:

$$\left[\begin{array}{ccccccccccccc} 1 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & -0.195m_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & -0.078m_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & -m_2(0.195 - 0.02 \cos(\theta_1) + 0.08 \cos(\theta_2) \sin(\theta_2)) & 0.02m_2 \cos(\theta_1) & -0.08m_2 \cos(\theta_2) \sin(\theta_2) & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & -m_2(0.078 - 0.02 \sin(\theta_1) + 0.04 (\sin(\theta_2))^2 - 0.04 (\cos(\theta_2))^2) & 0.02m_2 \sin(\theta_1) & -m_2(0.04 (\sin(\theta_2))^2 - 0.04 (\cos(\theta_2))^2) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & -0.195m_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.195 & -0.078 & 0 & 0 & 0 & 0 & -0.195 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.02 \cos(\theta_1) & -0.02 \sin(\theta_1) & 0 & 0 & 0 & 0 & -Izz0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.04 \sin(\theta_2) & -0.04 \cos(\theta_2) & 0.04 \sin(\theta_2) & -0.04 \cos(\theta_2) & 0 & 0 & -Izz1 & -Izz1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.036 \cos(\theta_3) & 0.036 \sin(\theta_3) & 0 & 0 & -Izz2 & 0 & -Izz2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -Izz3 & 0 & 0 & -Izz3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.02 \cos(\theta_1) & 0.08 \sin(\theta_2) & 0.036 \cos(\theta_3) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.02 \sin(\theta_1) & -0.08 \cos(\theta_2) & 0.036 \sin(\theta_3) \end{array} \right] \quad (5.28)$$

Matrix X

$$x = \begin{bmatrix} F_{cmx} \\ F_{cmy} \\ Ox \\ Oy \\ Ax \\ Ay \\ Bx \\ By \\ Cx \\ Cy \\ \frac{d^2}{dt^2}\theta_0(t) \\ \frac{d^2}{dt^2}\theta_1(t) \\ \frac{d^2}{dt^2}\theta_2(t) \\ \frac{d^2}{dt^2}\theta_3(t) \end{bmatrix} \quad (5.29)$$

Matrix b

$$b = \begin{bmatrix} 0 \\ gm_0 + F_{prop} \\ -0.078 m_1 \left(\frac{d}{dt} \theta_0(t) \right)^2 \\ m_1 \left(g + 0.195 \left(\frac{d}{dt} \theta_0(t) \right)^2 \right) \\ 0.04 m_2 \left(-2.95 \left(\frac{d}{dt} \theta_0(t) \right)^2 + 0.5 \left(\frac{d}{dt} \theta_0(t) \right)^2 \sin(\theta_1) + \left(\frac{d}{dt} \theta_0(t) \right) \left(\frac{d}{dt} \theta_1(t) \right) \sin(\theta_1) + 0.5 \left(\frac{d}{dt} \theta_1(t) \right)^2 \sin(\theta_1) \right. \\ \left. + 2.0 (\cos(\theta_2))^2 \left(\frac{d}{dt} \theta_0(t) \right)^2 + 4.0 (\cos(\theta_2))^2 \left(\frac{d}{dt} \theta_0(t) \right) \frac{d}{dt} \theta_2(t) + 2.0 (\cos(\theta_2))^2 \left(\frac{d}{dt} \theta_2(t) \right)^2 - 2.0 \left(\frac{d}{dt} \theta_0(t) \right) \frac{d}{dt} \theta_2(t) - \left(\frac{d}{dt} \theta_2(t) \right)^2 \right. \\ \left. m_2 \left((0.195 - 0.02 \cos(\theta_1) + 0.08 \cos(\theta_2) \sin(\theta_2)) \left(\frac{d}{dt} \theta_0(t) \right)^2 + (-0.04 \left(\frac{d}{dt} \theta_1(t) \right) \cos(\theta_1) \right. \right. \\ \left. \left. + 0.16 \sin(\theta_2) \cos(\theta_2) \frac{d}{dt} \theta_2(t) \frac{d}{dt} \theta_0(t) + g - 0.02 \left(\frac{d}{dt} \theta_1(t) \right)^2 \cos(\theta_1) + 0.08 \sin(\theta_2) \cos(\theta_2) \left(\frac{d}{dt} \theta_2(t) \right)^2 \right. \right. \\ \left. \left. - 0.195 F_{prop} \cos(\theta_3) \sin(\theta_3) \right. \right. \\ \left. m_3 g - 0.195 F_{prop} (\cos(\theta_3))^2 + 0.195 m_3 \left(\frac{d}{dt} \theta_0(t) \right)^2 \right. \\ \left. 0.05173333333 V(t) \right. \\ \left. - 0.05173333333 V(t) \right. \\ 0 \\ 0 \\ 0.02 \left(\frac{d}{dt} \theta_1(t) \right)^2 \sin(\theta_1) + 0.08 \left(\frac{d}{dt} \theta_2(t) \right)^2 \cos(\theta_2) + 0.036 \left(\frac{d}{dt} \theta_3(t) \right)^2 \sin(\theta_3) \\ - 0.02 \left(\frac{d}{dt} \theta_1(t) \right)^2 \cos(\theta_1) + 0.08 \left(\frac{d}{dt} \theta_2(t) \right)^2 \sin(\theta_2) + 0.036 \left(\frac{d}{dt} \theta_3(t) \right)^2 \cos(\theta_3) \end{bmatrix} \quad (5.30)$$

5.2.4 Model Validation

In order for the model to be usable, it first had to be validated through integration over time to observe if the variables (primarily angular positions over time of the four bodies) behave as expected. Before testing the entire system, an integration over time

of the constraint equation (the servo-rudder system) was performed in order to verify the geometry between the three bodies during movement. The model was integrated over 3 seconds of simulation time, with a small integration step of $\delta t = 0.0001$, in order to capture good resolution of the signal.

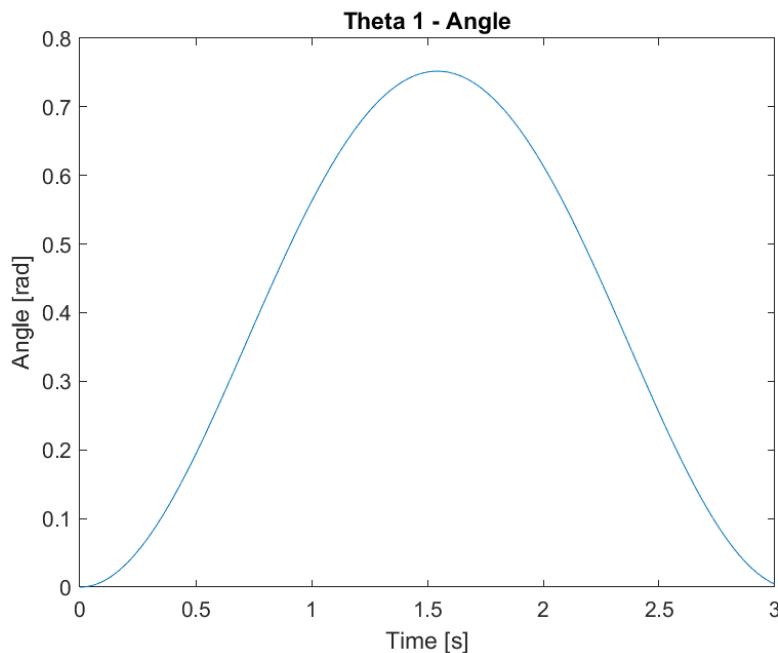


Figure 5.8: Model validation of the constraint equation for body 1.

Figure 5.8 shows the movement of the servo arm in radians, reaching its limit in one direction and returning to starting position.

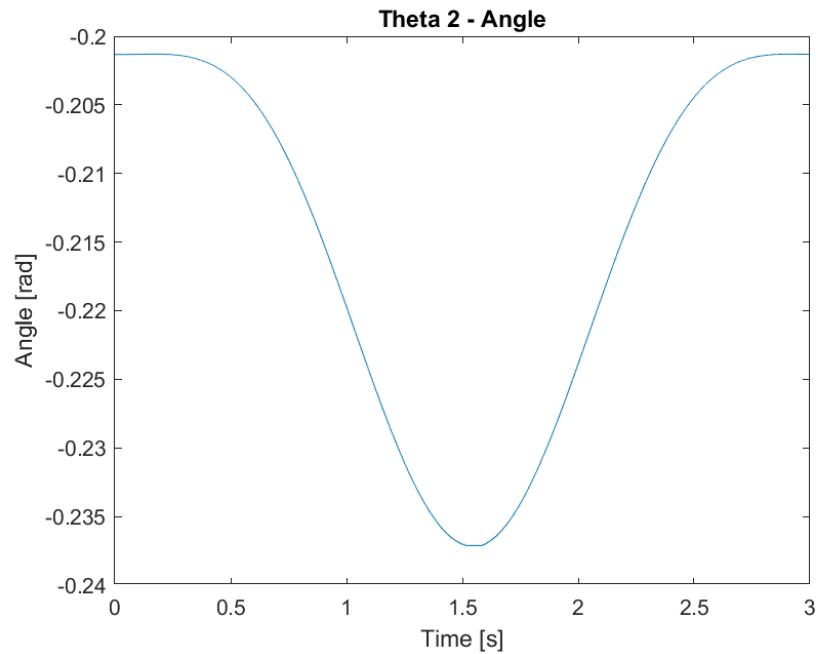


Figure 5.9: Model validation of the constraint equation for body 2.

Figure 5.9 shows the movement of the link in radians, reaching its limit in one direction and returning to starting position. The movement of the link is indeed very restrained.

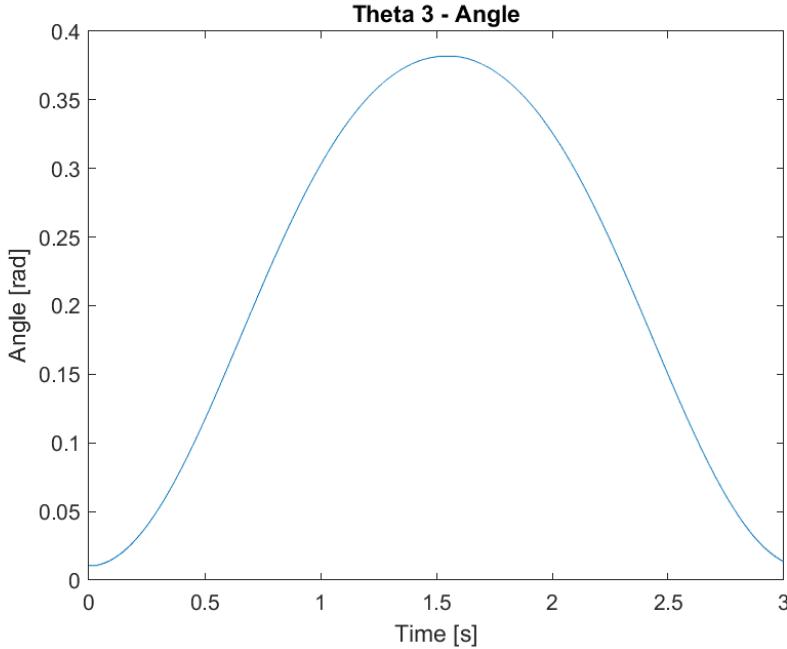


Figure 5.10: Model validation of the constraint equation for body 3.

Figure 5.10 shows the movement of the rudder in radians, reaching a limit in one direction and returning to starting position.

The limits displayed by the simulation are consistent with the real-life behavior of the bodies, synchronized with each other. This concludes the constraint equation validation of the model. The next step in validating the model is including the UAV body into the simulation, expanding all equations to include the interactions with the other three bodies and removing all forces from the system.

Therefore, the first method of model validation was done by setting parameters to zero forces and moments to verify that it produces the expected output of zero accelerations and zero motion, with initial conditions set to equilibrium position for all bodies. In fig. 5.11, Theta0 represents angular position of the UAV with respect to the inertial frame. Theta1 represents servo angle, Theta2 is the angle of the Body 2 (link between servo and rudder arm), Theta3 is the angle of the rudder. The model was being integrated over time, with masses and moments of inertia included but no gravitational acceleration or input force. In the case of correct model, the expected output was no motion over time. As seen in fig.5.11, this was correct.

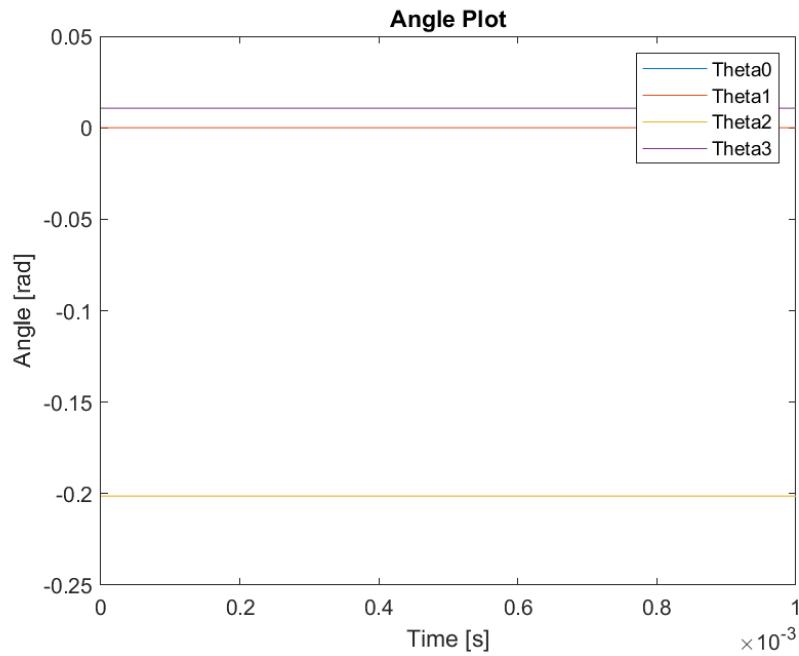


Figure 5.11: Model validation with zero forces.

The second test was performed by adding gravitational acceleration as the only force acting on the system, with all bodies expected to display a slow starting but accelerating movement, simulating the falling rotation of the UAV around its axis, with the other bodies attached to its frame following this motion. Fig.5.12 shows this test, with the UAV body (Theta0) increasing its rotation about its axis unbounded. Servo (Theta1) is following UAV frame movement, without displaying its kinematic constraints specific to having had voltage input, as it does during the next test. Body 2 and Body 3 are also following the "falling" motion.

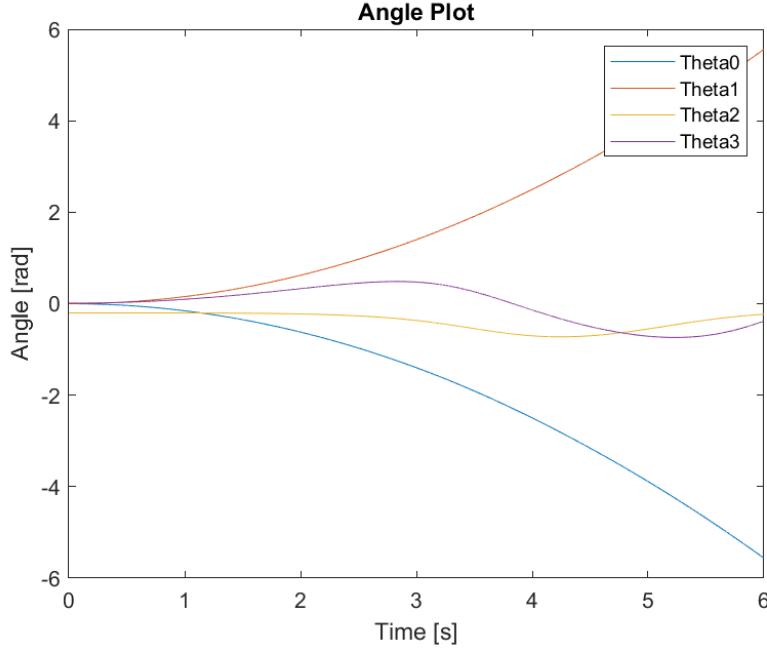


Figure 5.12: Model validation with gravity only.

In the third test case, the gravity parameter was kept and the propellers thrust force and servo moment were added. The expected behavior was to observe the kinematic constraints of the servo-rudder construction and the effect of the rudder on the UAV body. These are observed in fig. 5.13, where the UAV body 0 is inclined towards an angular position of 0.4 rad, then rapidly rotating in the opposite direction, since the system is not controlled and will therefore keep tumbling. Servo (body 1) is seen to reach its kinematic limit of nearly 0.8 rad (45 degrees), moving along with the rudder (body 3) that moves in the same direction but with lower angles amplitude, to which it is rigidly linked to (body 2).

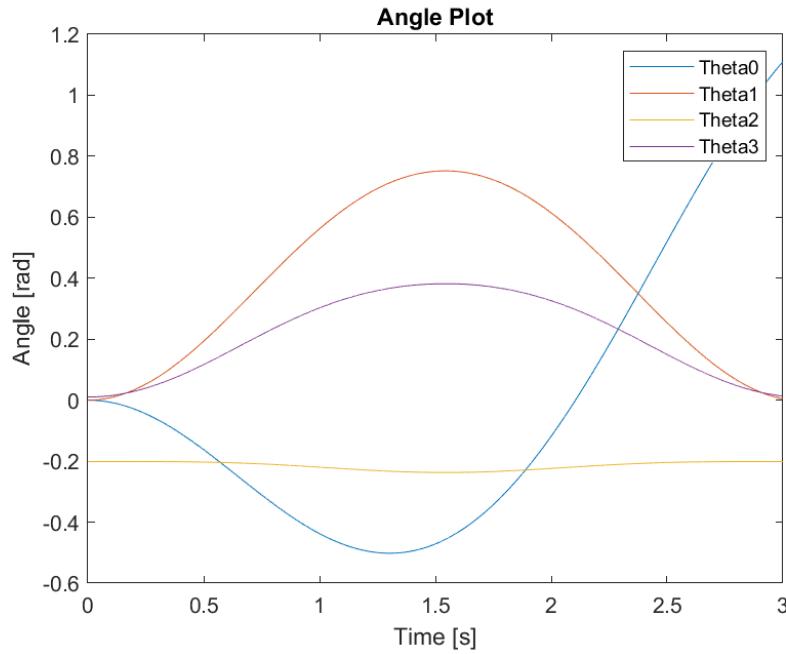


Figure 5.13: Model validation with forces applied.

Before moving into designing the controller based on the model, the model had to be correct. In this section, the model has proven to be correct. This concludes the model validation section and it can now be advanced towards the control section.

5.3 Control

Attitude control is the process of achieving and maintaining an orientation of the rocket. The advantage of model based PID controller tuning is that it provides robustness with respect to sensor noise, thrust control accuracy or unmodeled plant dynamics [18] . Likewise, having a dynamic model of the system allows for scalability, as well as tracking of system forces and their interactions.

Feedback control The feedback controller is a closed-loop servomechanism TVC system, described in the diagram below. The navigation system continuously updates the rocket with its current location, therefore the system knows where it is at all times; it knows this because it knows where it isn't - within reason. By subtracting where it is from where it isn't, or where it isn't from where it is (whichever is greater), it obtains a difference, or deviation. (Ground Launched Cruise Missile, 1997, p. 5) [1]

The feedback controller compensates for the deviation (measured error) in order to correct the current position of the body towards the desired position (reference input point). The controller sends the signal towards the actuator, which in turn adjusts the position of the thrust vector control system, thus correcting the orientation of the plant

(system). The guiding principles of this process are valid for both rocket system and UAV system.

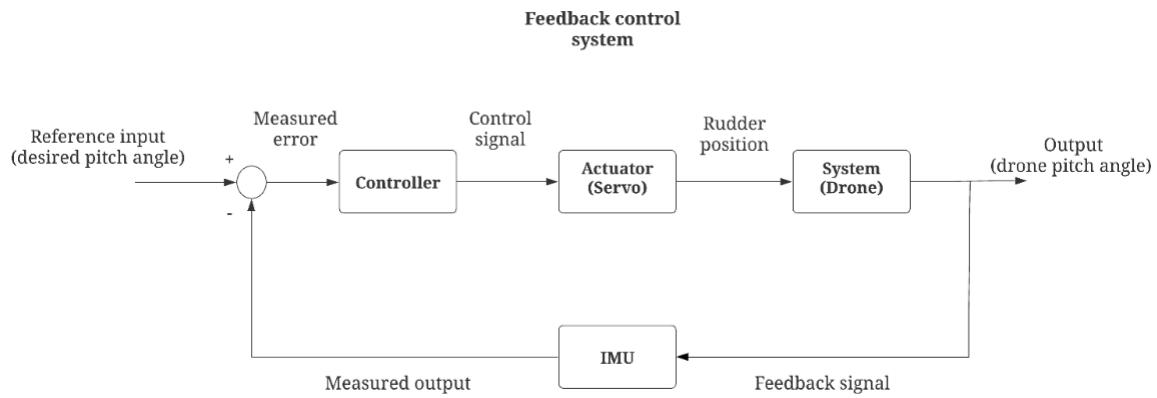


Figure 5.14: Feedback control diagram.

The role of the controller is to minimize the error. In designing the controller, it is aimed to adjust transient and steady-state response of a control system to meet the performance requirements and adjust the parameters of the system to the desired state. [24]

Controller design

In the modelling section chapter, the UAV was modeled using first principles (non-linear physical laws).

The system has voltage to servo as input, with drone angular position as output. The thrust force of the propellers is considered part of the model as opposed to an input due to the force being a constant. Therefore, the system is considered a SISO (single input single output), modelled as state space model.

Linearization - the model obtained from first principles was highly non-linear.

While this non-linearity gives higher precision, it is resource intensive and especially so for Arduino-class micro controllers. Linearization helps speed up this process and would facilitate control system design.

This is achieved by the first order partial derivatives of the model at a steady state operating point. (Jacobian matrix)

A is the Jacobian of the function evaluated at the steady state (equilibrium position).

Input equation

$$\dot{\mathbf{x}}_z = \mathbf{A}_z \mathbf{x}_z + \mathbf{B}_z \mathbf{u}_z$$

Output equation

$$\mathbf{y}_z = \mathbf{C}_z \mathbf{x}_z + \mathbf{D}_z \mathbf{u}_z$$

A – System Matrix	u -Vector 1
B – Input Matrix	z -Vector 2
C – Output Matrix	y -Vector 2
D – Output Matrix	

Linearized stability analysis

The stability is expressed with respect to the system response as a function of time.

In order to determine the stability of the system, its eigenvalues (poles) are analyzed, pictured in (5.15), Where the poles of the system are marked with an **x** and zeros are marked with **o**. It can be seen in (5.15) that the poles of the system have zero real part - indicating that the system is marginally stable.

On the imaginary scale, two poles having zero imaginary parts - indicating no oscillations - and two poles placed on the axis, indicating oscillations in the system. Therefore, the purely imaginary eigenvalues describe the system as a marginally stable, undamped oscillator.

The two zeroes of the system are placed on the real axis, at $+/-0.00808$. The zero in the left plane is a good solution for controller design, having stability and no oscillations.

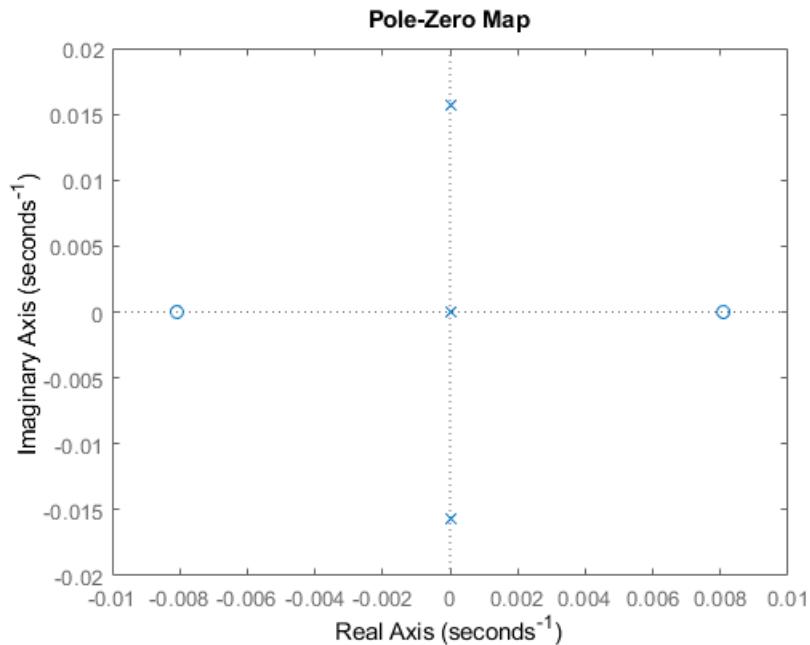


Figure 5.15: Pole-Zero Map.

The poles and zeroes are then analyzed in combination with varying gains for the feedback system, in order to determine viable solutions for the controller. The plot (5.16)

shows the grid on the left hand side of the real-imaginary plane, for the values where the system is stable.

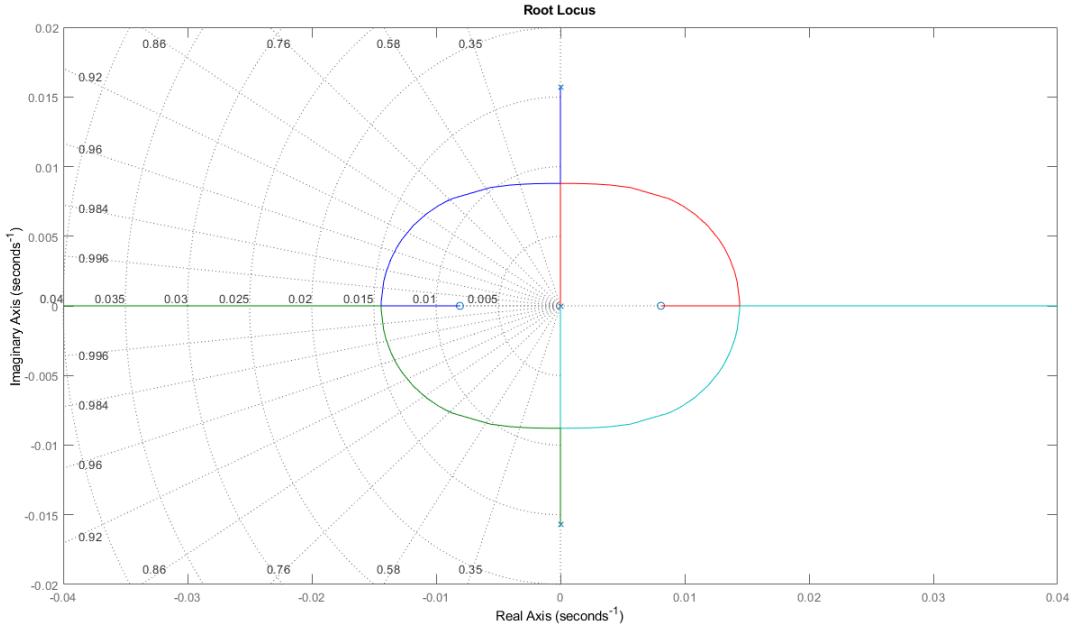


Figure 5.16: Root Locus Map.

Controllability matrix of the state-space LTI system

Controllability is a property of a system of a particular actuator configuration to control all the states of the system; in a similar manner, observability returns the property of the particular sensor configuration to estimate all the states of the system. [15]

The system returns full state observability.

Analysis of controllability matrix reveals that it has rank equal to the dimension of the state space, therefore the system is full rank controllable.

Control

The chosen control algorithm was a PID (Proportional Integral Derivative) controller, since literature revealed that typical control laws for rockets are variations of the PID controller. [5] [20]

An advantage of the PID is that it can fulfill the requirement of extending to additional axes because it can be decoupled - the current 1 axis control can be extended to 2 or 3 axes, which was one of the requirements. Another advantage of the PID is that a different set of gains can be calculated for different moments of inertia - in the case of

the rocket, it has a different moment of inertia due to exhausting the propellants, from the moment of launch to the moment of emptying the tanks.

The controller has a number of time-domain specifications:

- Rise time - time required to reach 90 percent of target
- Maximum overshoot - exceeding target
- Settling time - time required to reach steady state ([5]), page 318

The performance of the controller will be assessed against the requirements mentioned in section 1.5: max. 30 percent overshoot and 1.5 sec. settling time for the controller.

Simulation

A model was prepared in Simulink for testing the control algorithm for the UAV.

The control input is the actuator torque.

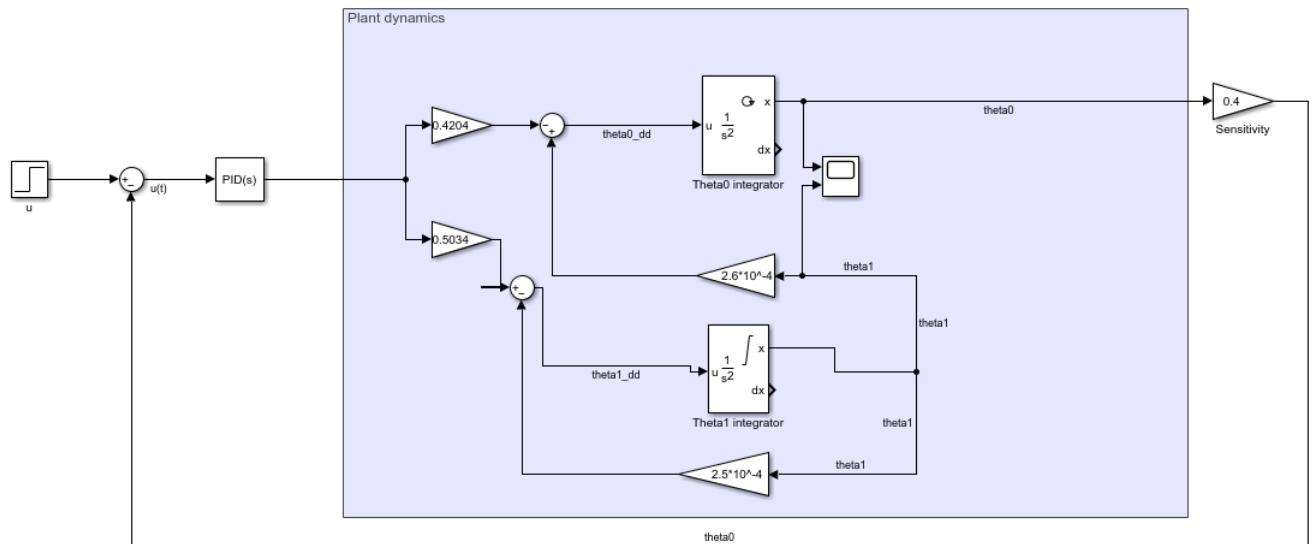


Figure 5.17: Simulink model used to test PID.

The controller was initially tested with Proportional gain (k_P) only, however, it was oscillating without stabilizing for any input gain. With the addition of the Derivative gain (k_D), the system stabilized. The gains were tuned in order to find the set of gains which minimized overshoot, rise time and settling time of the signal. The Integral gain was found not necessary, since the system reaches reference input without any

significant steady state error to be corrected for - and the Integral (kI) gain was found to be otherwise detrimental to the performance, introducing overshoot, oscillation and increasing the settling time, as well as the potential of wind-up - and was therefore set to 0.

In the Simulink model, the initial condition for UAV angular position was set to 0.2 rad, corresponding to a starting inclination that the controller will have to correct for in order to reach angular position of 0 rad. The velocity of the UAV was set to 0.2 rad/s initial condition. With these conditions, the best performing set of gains was found to be $kP = -2$ and $kD = -8$.

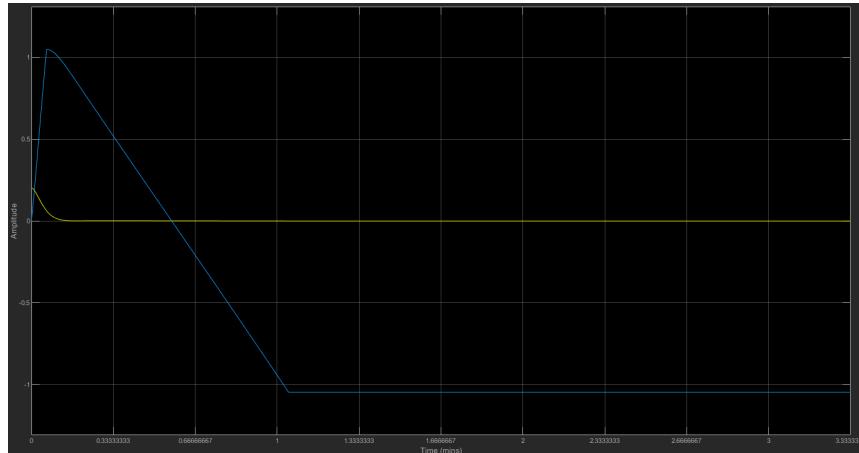


Figure 5.18: Oscilloscope graph of the angular position of UAV and servo angle.

In 5.18, θ_0 represents the angular position of the UAV and θ_1 represents the servo (actuator) position. With the mentioned gains, the system was found to have no overshoot, no necessary rising time and a settling time of 0.2 seconds. The controller thus sets the system, a harmonic oscillator, into a critically damped oscillator, reaching the equilibrium position of 0 rad quickly with no overshoot or oscillations. With this controller, the system remains critically damped with any initial condition ranging from 0 to ± 0.8 radians (45 degrees). This fulfills the requirements of having under 30 percent overshoot and under 1.5 settling time.

This chapter discussed the modelling and control decisions for the project, approaches which are to be demonstrated in the following chapter in tests.

CHAPTER 6

Experimental validation and results

6.1 Experimental setup for sensor fusion methods

This chapter describes the test facility and experiments done in order to evaluate the performance of the complementary, Mahony and Madgwick sensor fusion algorithms.

6.1.1 Experimental setup - navigation tests

The tests were performed with a the IMU and Arduino placed on a UR3 robotic arm and aided by the Optitrack motion capture system of the DTU robotic lab as "ground truth" due to measurements accuracy. Before each test, the IMU is calibrated, through the methods described in the navigation chapter until full calibration is achieved. However, it has been observed that under motion, the calibration of this device is affected, particularly on the accelerometer and magnetometer side, which negatively affected the filters relying on both these devices for updates.

The data from Optitrack was collected through its network broadcast address. The broadcast data is transmitted and received through UDP protocol, therefore there is loss of packets and the data appears noisy. The filters data was collected through an USB cable connected to the Arduino Nano with SAMD21 processor and the IMU sensor. The measurements are represented as a Gaussian distribution having a mean (offset) and a standard deviation (noise).

With a filter update rate of 1 kHz, the integrated gyro was found to drift while static with 1 deg/s. With an update rate of 1 mHz, the drift was reduced to 0.1 deg/s. This update rate was selected in order to provide comparable measurements with the fusion filters update rates: complementary, Mahony and Madgwick. The former two filters are iterative, having to iterate over several samples in order to converge.

How each RPY is calculated

The filters were aligned to match same IMU axes and polarities and ranges. The Madgwick filter was tuned to beta parameter of 0.5, which gives equal weight to gyro

and to the accelerometer-magnetometer pair. The complementary filter uses 0.95 weight from gyro and 0.05 from accelerometer for the roll and pitch axes and a compensated magnetometer calculation for the yaw.

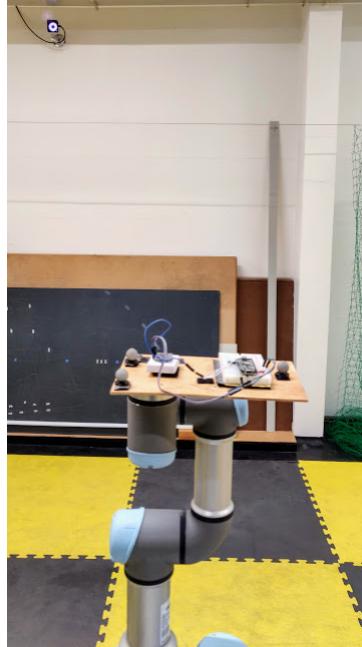


Figure 6.1: IMU testbed on UR3 and Optitrack motion capture.

In order to evaluate filters performance, two experiments were conducted for each algorithm:

Experiment 1

The first experiment evaluates the drift over time of the gyroscope without motion. The testbed is maintained static for 10 minutes for each filter; the data from all filters has been resampled to 100 Hz. Based on the measurements, the algorithms are evaluated with respect to the root mean square error and standard deviation from the reference point, compared to the raw integrated gyro data. The objective of this experiment is to compare the drift over time of the raw gyro measurements with the performance of the filters and selecting the algorithm that best minimizes this effect.

Experiment 2

The second experiment evaluates the performance of the filters in a motion ranging the IMU pitch angle from -45 to +45 degrees, involving all three components of the IMU in sensor fusion and comparing their performance against raw integrated gyro data and ground truth provided by the Optitrack motion capture system. The testbed performs three cycles of the full motion. The objective of this experiment is to compare the performance of the filters against raw integrated gyro data and select the best performing algorithm in term of least deviation from the reference point from Optitrack.

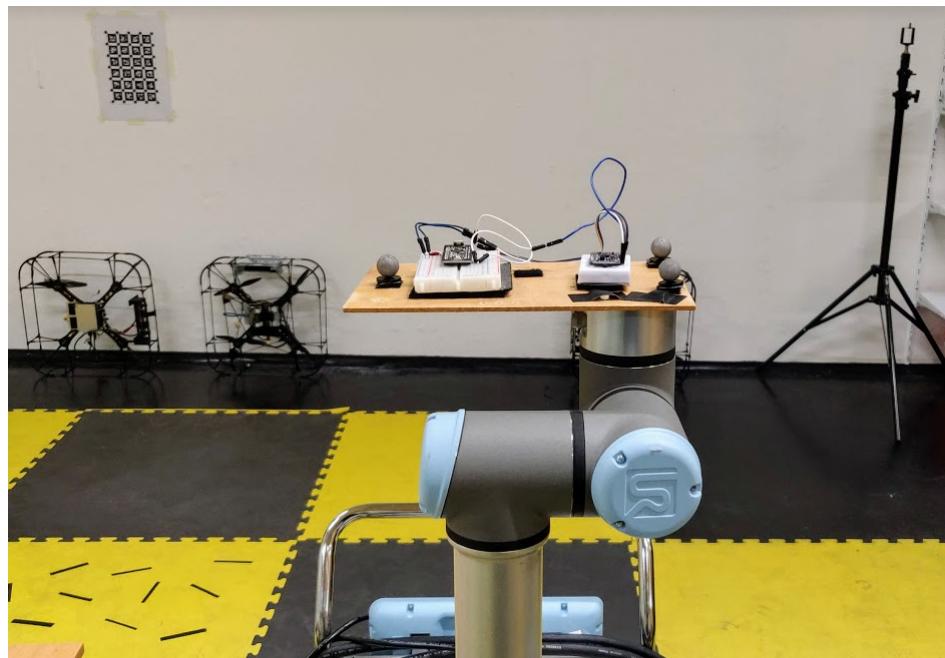


Figure 6.2: IMU static testing.



Figure 6.3: IMU inclination testing.

6.1.2 Analysis of navigation performance

Optitrack, Complementary filter and integrated gyro provide a relative measurement of yaw from their initial point, considered 0. Mahony and Madgwick filter calculate absolute heading as measured by the magnetometer, based on the Earth's magnetic North. Due to the objective of this test is to measure the drift while static, all filters have been aligned to the absolute yaw measurement provided by the former two filters - then measuring deviations while the sensor is maintained static.

Experiment 1: Static test for comparing drift over time

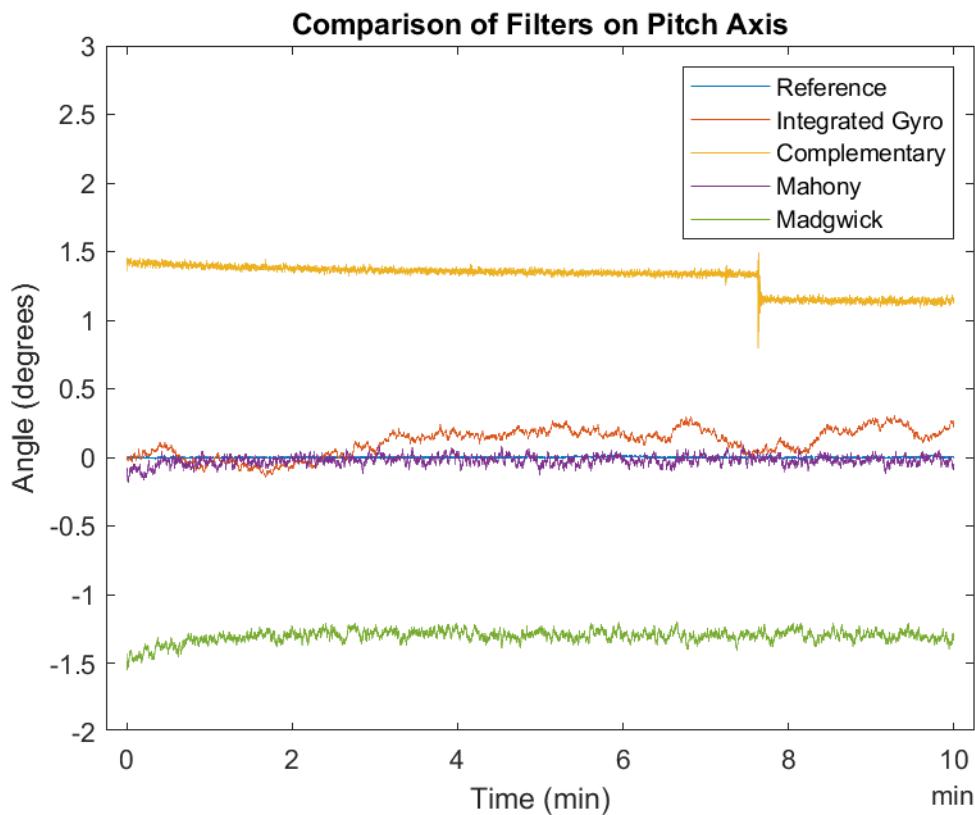


Figure 6.4: Comparison of Filters on Pitch Axis.

Figure 6.4 compares the performance of the chosen fusion filters against the reference signal from Optitrack. Pitch axis has been selected because it coincides with the axis that the UAV can be tested on, with the current setup. Plots of the roll and yaw axes available in the appendix A.12, A.13.

This is showing how each filter performs while static, to investigate the method that best minimizes drift over time and signal noise.

The gyro is known to be better suited for motion rather than static applications, as it has low noise incidence but presents drift over time. The accelerometer is known to be

stable over time, but has higher noise occurrences. In the case of the complementary filter in 6.2, the platform was not moved, therefore the most probable incident around 8:00 minute time that led to the spike was likely higher frequency accelerometer noise. This plot shows that the complementary filter does not compensate for such sudden spikes. The integrated gyro remained close to the reference 0 degrees for 3 minutes, after which an increase in drift is visible. Mahony filter performed best in this graph, closest to the reference point. The Madgwick filter shows steadiness in signal once converged, which seems to have taken about one minute with this beta parameter setting. The -1.3 offset is likely due to loss of calibration in the accelerometer and magnetometer data, showing it is more sensitive to this error than the Mahony filter.

Filters	Mean Roll	StdDev Roll	Mean Pitch	StdDev Pitch	Mean Yaw	StdDev Yaw
<i>Integrated Gyro</i>	-0.191	0.086	0.109	0.100	197.02	1.781
<i>Complementary Filter</i>	0.478	0.207	1.311	0.096	200.51	3.128
<i>Mahony Filter</i>	0.086	0.071	-0.022	0.033	200.26	0.268
<i>Madgwick Filter</i>	0.175	0.125	-1.304	0.043	200.54	0.624

Table 6.1: Table of navigation filters per axis for static test.

Table 6.1 shows the performance of the filters for each separate axis. The mean value for roll and pitch is desired to be closer to 0, whereas for yaw it is desired to be closer to 200, the absolute yaw value as indicated by the sensor compass, while standard deviation is desired to be minimized. Based on these parameters, gyro integrated had a mean absolute offset of 0.1-0.2 degrees for roll and pitch and the largest yaw deviation of all filters. The other filters had similar yaw performance in static conditions. Complementary filter presented the largest roll and pitch offsets, as well as standard deviation overall. This is possibly due to the large weight given to gyro measurements in the filter, which tends to drift over time while static, possibly amplified by the incidence of accelerometer noise. It is expected that under movement, the adapting properties of the gyro will provide the filter with better performance. Mahony performed best both in terms of offset and noise, consistently across all axes. Madgwick measured slightly less well, unexpectedly, with a large deviation on pitch mean, but low noise; low roll offset but slightly higher noise than Mahony and gyro integrated and finally, good performance on yaw data. Madgwick is likely to have been more impacted than Mahony by calibration errors.

Experiment 2: Motion test for comparing performance under movement

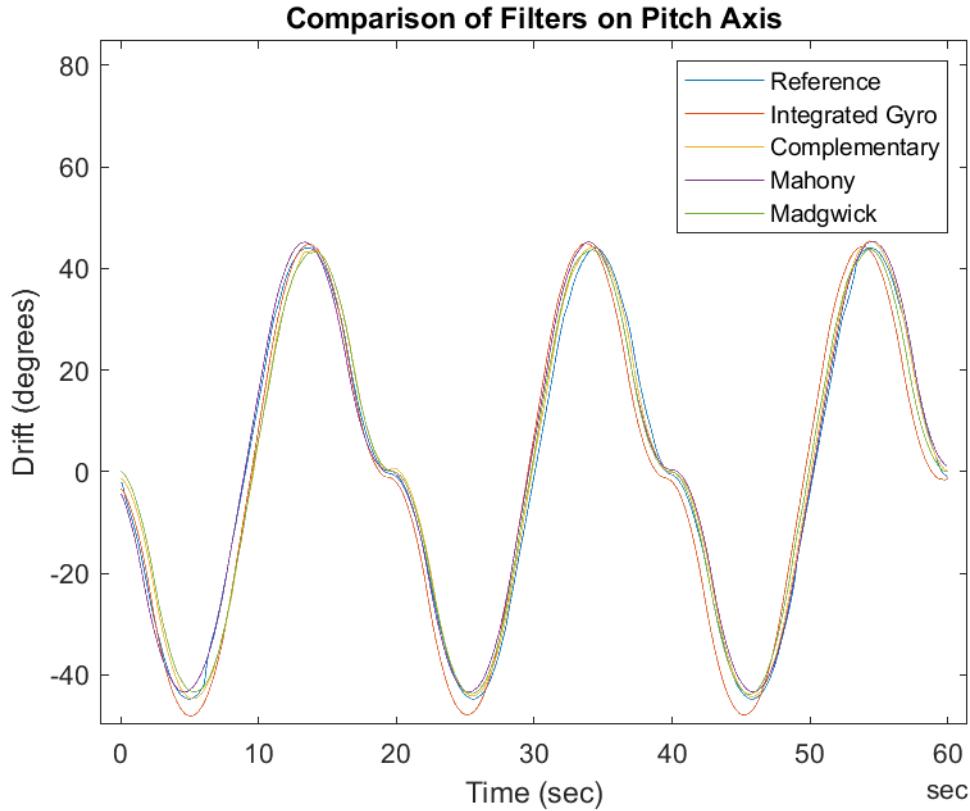


Figure 6.5: Comparison of Filters on Pitch Axis in Motion.

Figure 6.5 compares the performance of the chosen fusion filters against the reference signal from Optitrack, during a motion of approximately ± 45 degrees angle on the pitch axis. All filters performed well in motion, following the reference signal closely; the most significant being the deviation of 1.598 for the integrated gyro. However, on the roll (A.22) and yaw (A.23) axes, some larger differences emerged. While it represented the trend of the motion well, the gyro shows a significant offset of 5.011 degrees from the reference. This is due to the fact that it lacks update from an absolute measurement, as the other filters do. It follows that integrating gyro is not reliable data on its own, due to not being able to detect the true angular position - but only the motion relative to its current (drifted) angle. This drift was particularly apparent on the yaw axis (A.23), where there is an upward drifting trend that does not follow reference well. The other outlier was complementary filter, having also challenges in following the motion reference on roll - and particularly so on yaw axis. The complementary filter is a suitable choice as far as roll and pitch is involved, but it is not reliable for yaw, as the deviation is too high compared to the other filters. Mahony had the smoothest and most accurate signal on roll but having flatter output on yaw than expected, with Madgwick having less smooth output on roll yet displaying a more reliable capture of the signal trend on yaw axis.

Filters	Mean Roll	StdDev Roll	Mean Pitch	StdDev Pitch	Mean Yaw	StdDev Yaw
<i>Integrated Gyro</i>	5.011	1.657	-1.598	30.641	178.86	0.770
<i>Complementary Filter</i>	0.995	1.445	0.041	29.496	185.51	3.940
<i>Mahony Filter</i>	-0.060	1.805	0.575	29.547	178.46	0.389
<i>Madgwick Filter</i>	-0.605	0.652	0.026	28.882	181.3	1.124

Table 6.2: Table of navigation filters per axis for motion test.

The integrated gyro performs better in static than movement tests, in terms of both offset (mean) and noise measurement (std dev). While its yaw std.dev. is low, that measurement cannot be considered conclusive since it did not report well on the true motion. This difference between static and motion could potentially be accounted for by the additional drift from reference it experiences under movement, not having information about absolute measurements to correct this drift, thus leading to deviations wider in amplitude. The complementary filter encountered similar challenges, being 95 percent based on gyro measurements - it performed better in the static test. While its pitch and roll measurements were fairly similar to Mahony and Madgwick filters, its yaw estimation is not usable, having widely deviated and provided particularly noisy output. Since it is a requirement in this project to have accurate yaw data, the integrated gyro and complementary filter prove to not be suitable.

Mahony performed similarly to the static test, with reliable and consistent, measurements for both pitch and roll axis, with reliable signal trend capture. Yaw output was satisfactory, considering that the magnetometer lost and regained calibration during the motion, which impacted this measurement for both Mahony and Madgwick filters. It is expected that with more consistent sensor calibration, yaw output will improve. Madgwick filter also performed well - with best results on pitch axis and best trend capture on yaw axis of all filters. Roll measurement was its most affected, but it still provided good data compared to integrated gyro and complementary - and overall a comparable performance to Mahony.

Combined measurements

Filters	RMSE Static	RMSE Motion
<i>Integrated Gyro</i>	3.4796	8.557
<i>Complementary Filter</i>	3.4712	8.057
<i>Mahony Filter</i>	0.3898	3.3193
<i>Madgwick Filter</i>	1.5579	4.6223

Table 6.3: Table comparing RMSE performance of navigation filters.

The RMSE (Root Mean Squared Error) for each test was evaluated by taking the square root of the sum of squared differences between each filter axis and the reference (Optitrack). This metric was chosen because it provides an overview of each filter's deviation, combined across all axes. This deviation is in units of degrees, representing the offset compared to reference.

Performance is measured by having a smaller RMSE value, closer to 0. The RMSE during the static tests show the iterative Mahony and Madgwick filters with a significantly improved performance over integrated gyro and complementary filters. This case holds for both static and motion tests. The integrated gyro shows that indeed it is the least desirable option, with the complementary filter showing marginal improvement over it.

During the tests, particularly during the motion test, calibration of the IMU was affected, especially so the accelerometer and magnetometer. Therefore, filters Mahony and Madgwick relying on these two devices had their performance impacted more than the other two filters, relying on gyro only (integrated gyro) and accelerometer plus gyro (complementary filter).

Even so, the two former iterative filters displayed improved performance and robustness over the integrated gyro and complementary. Mahony and Madgwick both suffered the same deterioration of calibration - out of which, Madgwick proved more sensitive to such impact, having a higher deviation (offset) as seen in the mean value per axis (absolute values of 0.175, 1.304, 0.624 for static test, and 0.026, 0.605, 18.7 for motion - compared to expected values of 0 roll, 0 pitch, 200 yaw).

The calibration of the sensor is potentially affected by vibration of the robotic platform (linear acceleration read by the accelerometer), issues in initial calibration of the accelerometer, hard iron offsets due to the presence of metal surfaces in the robotic lab. Another point of impact on the performance of the filters is the hardware - the IMU is a low-cost device.

6.2 Experimental setup for control algorithm

6.2.1 Experimental setup - control test

The objective was to test the controller on pitch axis, as designed, without interference from additional forces on the system. The initial setup had the drone oscillating wildly in its enclosure, therefore some changes were performed. The drone was reinforced with steel wire and adjusted into a rigid body, to remove side wobbling and products of inertia. In the previous setup, the propeller thrust was hitting the floor and influencing drone oscillations - therefore it was moved at a higher above floor level, hanged about its center of mass and restricted its motion to only allowing the pitch axis degree of freedom.

It is outside of the scope of the project to control the thrust of the propellers - in

the same manner that the rocket thrust is not handled by the guidance system, but given as a constant. Therefore, the testing is restricted to the drone's angular position, as influenced by the rudder deflecting the propeller thrust. Testing is performed with the drone tethered for safety, with its body frame placed at some inclination from its equilibrium point and measuring for the rise time, overshoot and settling time of the controller.

6.2.2 Analysis of control performance

This chapter demonstrated the use of the model-based control, with the observed outcome; following this, it will be the discussion of the results and the tests to be carried in future work.

CHAPTER 7

Discussion and Future Work

This chapter further discusses the results from the testing phase, along with plans for future work

7.1 Discussion

The goal of having the thrust vector control restricted to one axis was to allow for the clear understanding of the working principles and process workflow of designing and controlling a rigid body. From this understanding, the same principles can be used further, by just extending the rudder to a two-axis control platform instead and adding a decoupled control for the additional axis (yaw).

The multitude of IMUs tried and tested helped understand the process from the ground up of navigation. From converting the voltage readings into the respective units of each device, to reading and aligning reference frames, to experiencing the limitations of each and relative to each other, such as the errors found in the cheaper versions (crosstalk) were significantly reduced in the higher end ones (BNO055). It was not the initial plan to test multiple IMUs, however as more sensors became available, it warranted comparison for finding the better fit for the project. The initial recommendation from the organisation was to limit the scope of this project by simulating a perfect IMU sensor. This was to avoid dealing with the significant challenges of real-world hardware. However, the navigation has proven to be a highly rewarding phase in terms of understanding the process, real life challenges in inertial sensors and the interaction with hardware.

Before finding the Mahony/Madgwick filters, there were other filters tried, which were finally deemed unnecessary after these filters, as they handle vibration. Methods for removing noise were tried, such as Moving average with different weight for older versus newer data, Weighted average, Median filter, Savitzky-Golay Filter, out of which moving/weighted average showed the most promising results, as they have an effect of smoothing the errors, whereas Savitzky-Golay proved to follow outliers more closely as opposed to removing them. Moving average is a Finite Impulse Response (FIR) filter, taking N data samples and averaging them over a given window of time. Exponential moving average filter is an Infinite Impulse Response (IIR) filter. It has a parameter (weight) ranging from 0 to 1, a higher value giving more weight to older data. In figure 7.1, this parameter was set to 0.9.

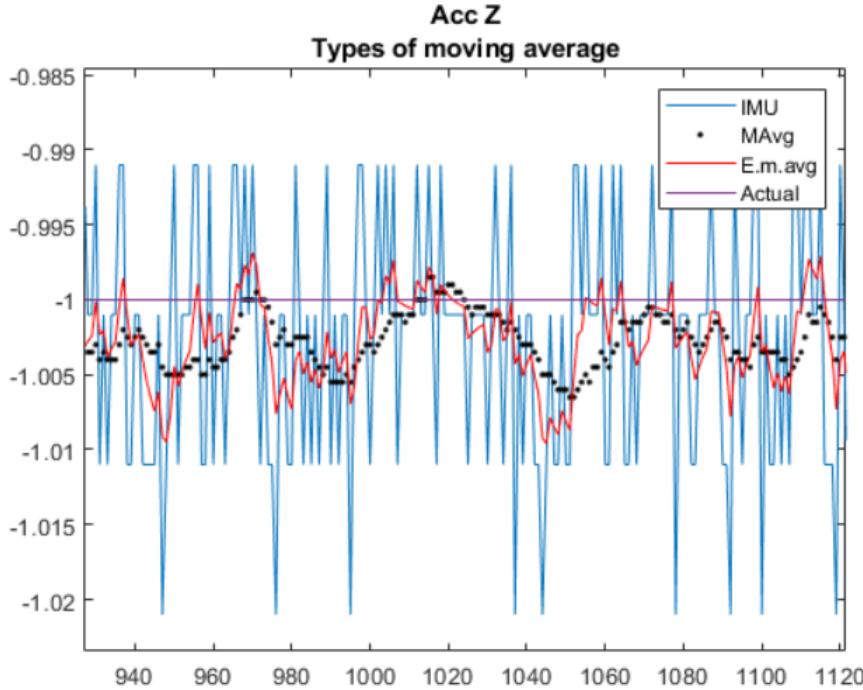


Figure 7.1: Comparison of moving average filters.

Figure 7.1 shows the noise of an accelerometer in one axis (z), measured in units of g (9.8 m/s^2). This graph plots the acceleration (y axis) measured while the sensor is maintained static, over a subset of samples taken (x axis). The red line ("Actual"), represents the true value of the gravitational acceleration, 1 g, used as reference. The blue "IMU" signal represents the noisy IMU measurement of the gravitational acceleration. The signals plotted in red and black show the two types of moving average filters, sliding window and exponential weighting.

While the exponential method follows the changes in signal more precisely, sliding window moving mean provides a smoother signal, that is more suitable for passing the data to a controller, as is the case in this project. Therefore, the sliding window method would be preferred. It can be noticed that the moving average introduces a delay, significant compared to the exponential moving average. The moving mean filter of length N has a sample delay of:

$$(N - 1)/2 \quad (7.1)$$

An attempt was made to compare simple moving average and exponential weighted moving average filters from figure 7.1 between themselves, with delay removed from the simple moving average. In that effect, it can be seen in figure 7.2 that the smoothing effect is slightly improved over the exponential moving filter, achieving lower peaks and more cohesive data. The moving average filter can be a useful alternative to the

complementary filter-based solutions presented in the fusion chapter, if the delay is taken into consideration. Measurement data was found to be normally distributed, therefore the IMU could also benefit from a probabilistic filter such as Kalman filter.

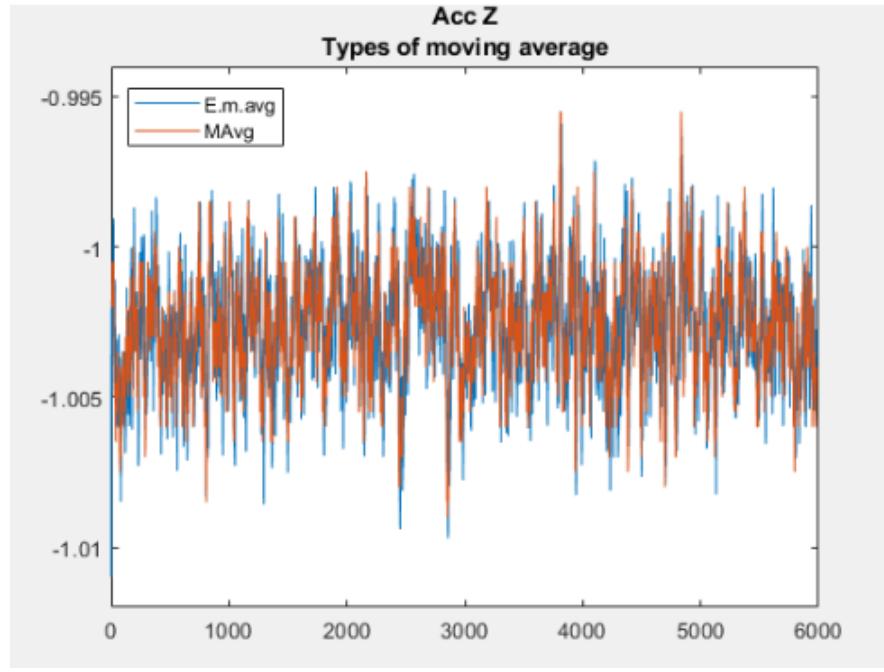


Figure 7.2: Comparison of moving average filters with removed delay.

Navigation

Complementary isn't suitable for singularities because of way it uses Euler angles. On top of it Mahony performing best anyway.

Control considerents: robustness

Calibration of the errors was important because the derivative component of the PID amplifies high-frequency signals (noise) in the sensor measurements.

In case of using the Integral part of PID, anti-windup measures are also necessary.

7.2 Future work

This chapter rounded up the remaining discussion of the project. Subsequent iterations of the hardware prototype would expand its capabilities to two axis control surface, which would require designing and building another deflection mechanism.

The navigation will be tested on a small scale rocket built by former DanSTAR students of DTU, in collaboration with Copenhagen Suborbitals, planned to be launched in Portugal in the fall of 2021. Prior to this, there will be conducted additional vibration

tests of the filters, along with performance under acceleration, jerky movements and heavy vibration.

The controller will have added features with focus on robustness and adding variable mass and moment of inertia as opposed to the fixed bodies moment of inertia in the current iteration. Tests should be done on variable mass drone to closer simulate the rocket.

Another possibility for control architecture is gain scheduling or cascaded control.

CHAPTER 8

Conclusion

The main objective of this project was to design, implement and test a control system and navigation approach. The main tasks were: a) understanding what is involved in rocket stability in flight and gimbaled engine TVC b) building a technology demonstrator prototype simulating the rocket to test the control systems on c) developing a inertial state estimation system based on an inertial measurement unit d) designing a controller that will use the inertial sensor data to control a system suitable for a gimbaled engine e) testing controller on a technology demonstrator prototype as a substitute for the rocket. The requirements were: a) investigating sources of errors in the inertial sensor, particularly the gyro drift in flight; b) deciding on an absolute heading (yaw) sensor to correct the gyro orientation; c) navigational attitude representation through a method avoiding singularities; d) investigating navigation filters fit for lower processing power, Arduino class micro-controllers; e) navigation filters with a average deviation of maximum 1.5 degrees from the true value, for each axis; f) 1 axis control that can be extended to 2 or 3 axes, g) controller with max 30 percent overshoot and 1.5 sec settling time.

The first task was fulfilled in chapter 2, Second task in chapter 3, third task in chapter 4 and last task in chapter 6. Requirement 1 to 4 were discussed in chapter 4, discussing the various types of errors specific to each device, choosing magnetometer as absolute heading measurement, filters based on quaternions which avoids singularities and fit for lower processing power microcontrollers. Requirement 5 was validated in chapter 6, during testing, where Mahony filter emerged as the better performing filter, with Madgwick a close second, given steady calibrated data. The control requirements were fulfilled by using a PID controller, which can be extended for all axes and having found sets of gains which stabilize the system with no overshoot and returning it to equilibrium in the least possible time. Therefore, all tasks and all requirements were met in full. The objective of understanding and developing a workflow necessary for the concept of an inertial navigation based thrust vector control was achieved. Due to the meeting of the agreed objectives, the project was successful.

Appendices

APPENDIX A

Appendix

A.0.1 Project code

The code for the project can be found at:

<https://github.com/BiancaDT/ArduinoIMU>

A.0.2 Equations

$$\ddot{\theta}_2 = \frac{\left((\sin(\theta_3(t)) \sin(\theta_1(t)) + \cos(\theta_1(t)) \cos(\theta_3(t))) \left(\frac{d}{dt} \theta_2(t) \right)^2 + \cos(\theta_1(t)) \sin(\theta_3(t)) - \sin(\theta_1(t)) \cos(\theta_3(t)) \right) \frac{d^2}{dt^2} \theta_1(t) + \left(\frac{d}{dt} \theta_1(t) \right)^2 \left((\cos(\theta_1(t)) \sin(\theta_2(t)) - \sin(\theta_1(t)) \cos(\theta_2(t))) \left(\frac{d}{dt} \theta_3(t) \right)^2 - \sin(\theta_2(t)) \sin(\theta_1(t)) - \cos(\theta_1(t)) \cos(\theta_2(t)) \right)}{\left((4.0 \cos(\theta_3(t)) \cos(\theta_2(t)) + 4.0 \sin(\theta_2(t)) \sin(\theta_3(t))) \left(\frac{d}{dt} \theta_2(t) \right)^2 - 4.0 \sin(\theta_2(t)) \cos(\theta_2(t)) + 4.0 \sin(\theta_2(t)) \cos(\theta_3(t)) + 4.0 \cos(\theta_2(t)) \cos(\theta_3(t)) \right) \left(\frac{d}{dt} \theta_3(t) \right)^2 + (4.0 \sin(\theta_1(t)) \cos(\theta_2(t)) - 4.0 \sin(\theta_2(t)) \cos(\theta_3(t))) \left(\frac{d}{dt} \theta_1(t) \right)^2 + 4.0 \cos(\theta_3(t)) \cos(\theta_2(t)) + 4.0 \sin(\theta_2(t)) \sin(\theta_3(t))} \quad (A.1)$$

$$\ddot{\theta}_3 = \frac{\left((5.0 \sin(\theta_1(t)) \cos(\theta_2(t)) - 5.0 \sin(\theta_2(t)) \cos(\theta_1(t))) \left(\frac{d}{dt} \theta_2(t) \right)^2 + 5.0 \cos(\theta_2(t)) \cos(\theta_1(t)) + 5.0 \sin(\theta_2(t)) \sin(\theta_1(t)) \right) \frac{d^2}{dt^2} \theta_1(t) + (5.0 \cos(\theta_2(t)) \cos(\theta_1(t)) + 5.0 \sin(\theta_2(t)) \sin(\theta_1(t))) \left(\frac{d}{dt} \theta_1(t) \right)^2 + (-5.0 \sin(\theta_1(t)) \cos(\theta_2(t)) + 5.0 \sin(\theta_2(t)) \cos(\theta_1(t))) \left(\frac{d}{dt} \theta_3(t) \right)^2 + (-9.0 \sin(\theta_3(t)) \cos(\theta_2(t)) + 9.0 \sin(\theta_2(t)) \cos(\theta_3(t))) \left(\frac{d}{dt} \theta_2(t) \right)^2 + (-9.0 \sin(\theta_3(t)) \cos(\theta_2(t)) + 9.0 \sin(\theta_2(t)) \cos(\theta_3(t))) \left(\frac{d}{dt} \theta_1(t) \right)^2 + 9.0 \cos(\theta_3(t)) \cos(\theta_2(t)) + 9.0 \sin(\theta_2(t)) \sin(\theta_3(t))}{\left((9.0 \cos(\theta_3(t)) \cos(\theta_2(t)) + 9.0 \sin(\theta_2(t)) \sin(\theta_3(t))) \left(\frac{d}{dt} \theta_2(t) \right)^2 + 9.0 \sin(\theta_3(t)) \cos(\theta_2(t)) - 9.0 \sin(\theta_2(t)) \cos(\theta_3(t)) \right) \left(\frac{d}{dt} \theta_3(t) \right)^2 + (-9.0 \sin(\theta_3(t)) \cos(\theta_2(t)) + 9.0 \sin(\theta_2(t)) \cos(\theta_3(t))) \left(\frac{d}{dt} \theta_2(t) \right)^2 + 9.0 \cos(\theta_3(t)) \cos(\theta_2(t)) + 9.0 \sin(\theta_2(t)) \sin(\theta_3(t))} \quad (A.2)$$

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & m_1(0) & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & m_1(0) & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & -m_2(0.02 \cos(\theta_1)) & m_2(0.04 \sin(\theta_2)) & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & -m_2(0.02 \sin(\theta_1)) & -m_2(0.04 \cos(\theta_2)) & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & m_3(0) \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & m_3(0) \\ 0.0 & 0.0 & 0.02 \cos(\theta_1) & 0.02 \sin(\theta_1) & 0 & 0 & 0 & 0 & -Izz_1 & 0 & 0 \\ 0 & 0 & -0.04 \sin(\theta_2) & 0.04 \cos(\theta_2) & -0.04 \sin(\theta_2) & 0.04 \cos(\theta_2) & 0 & 0 & 0 & -Izz_2 & 0 \\ 0 & 0 & 0 & 0 & -0.036 \cos(\theta_3) & -0.036 \sin(\theta_3) & 0.000 & 0 & 0 & -Izz_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.02 \cos(\theta_1) & 0.08 \sin(\theta_2) & 0.036 \cos(\theta_3) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.02 \sin(\theta_1) & -0.08 \cos(\theta_2) & 0.036 \sin(\theta_3) \end{bmatrix} \quad (A.3)$$

A.0.3 Drawings and schematics

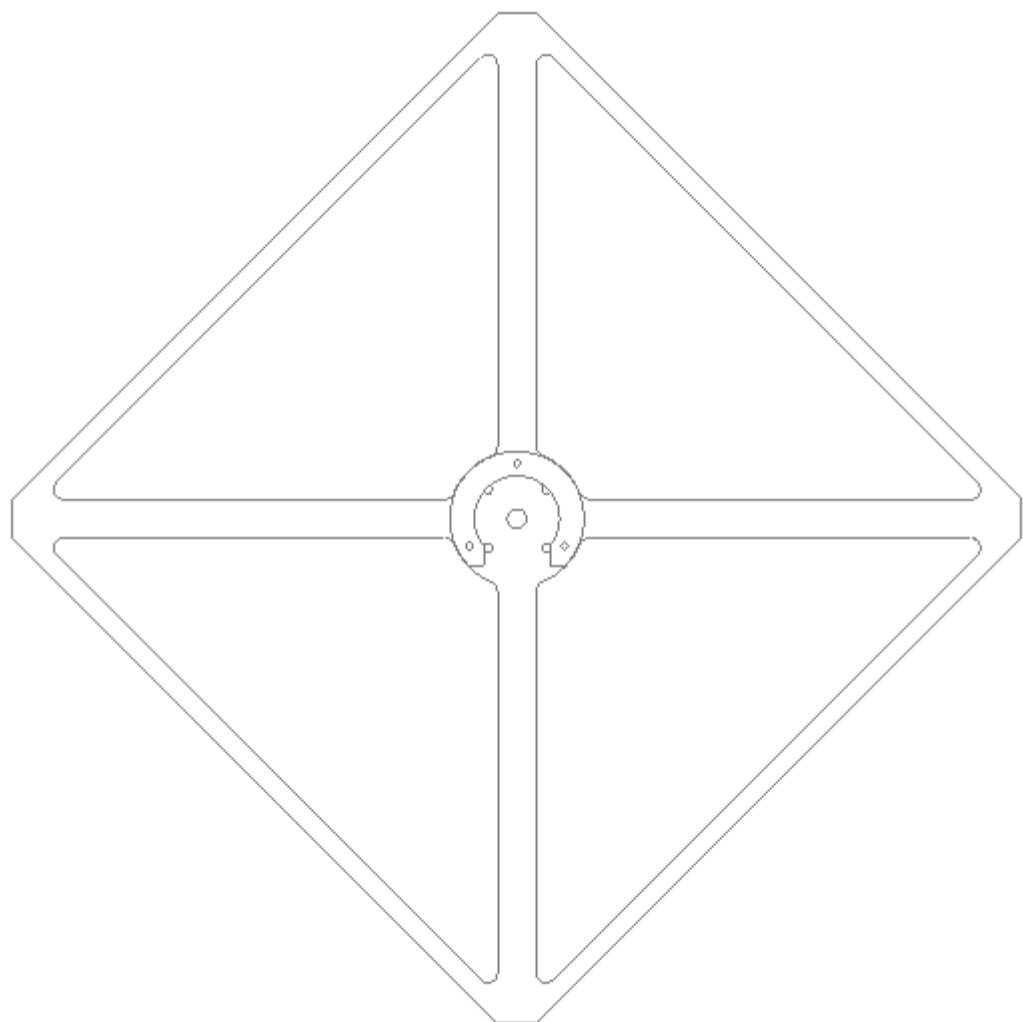


Figure A.1: UAV frame front.

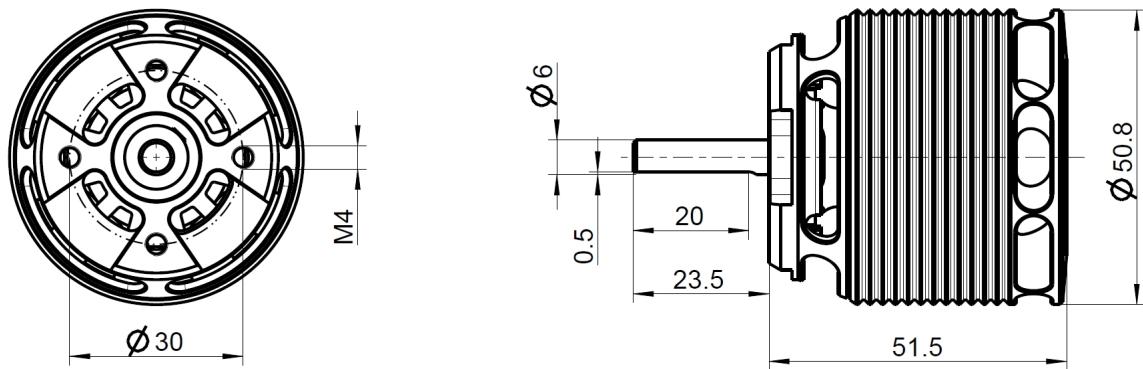


Figure A.2: Propeller Motor.

A.0.4 Graphs

Median filter effect on IMU data

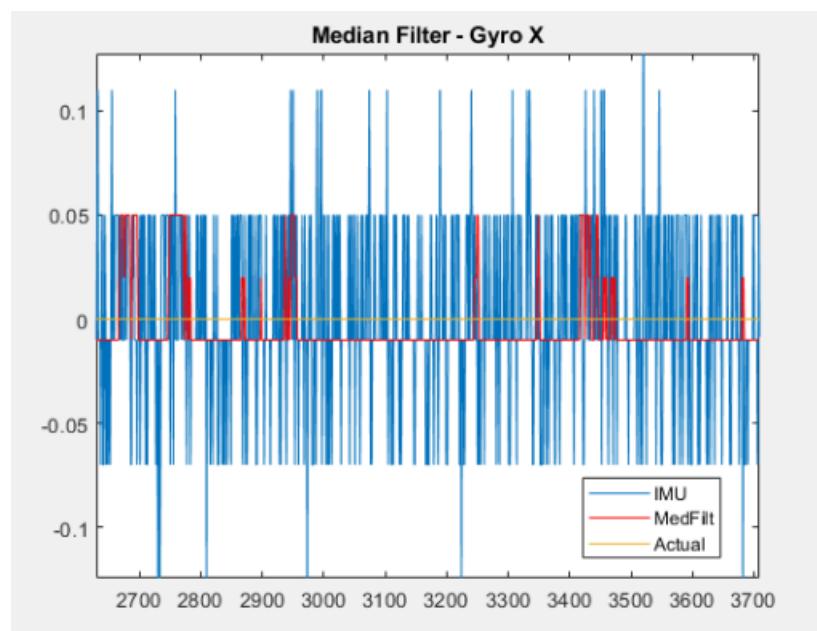


Figure A.3: Median filter effects on data.

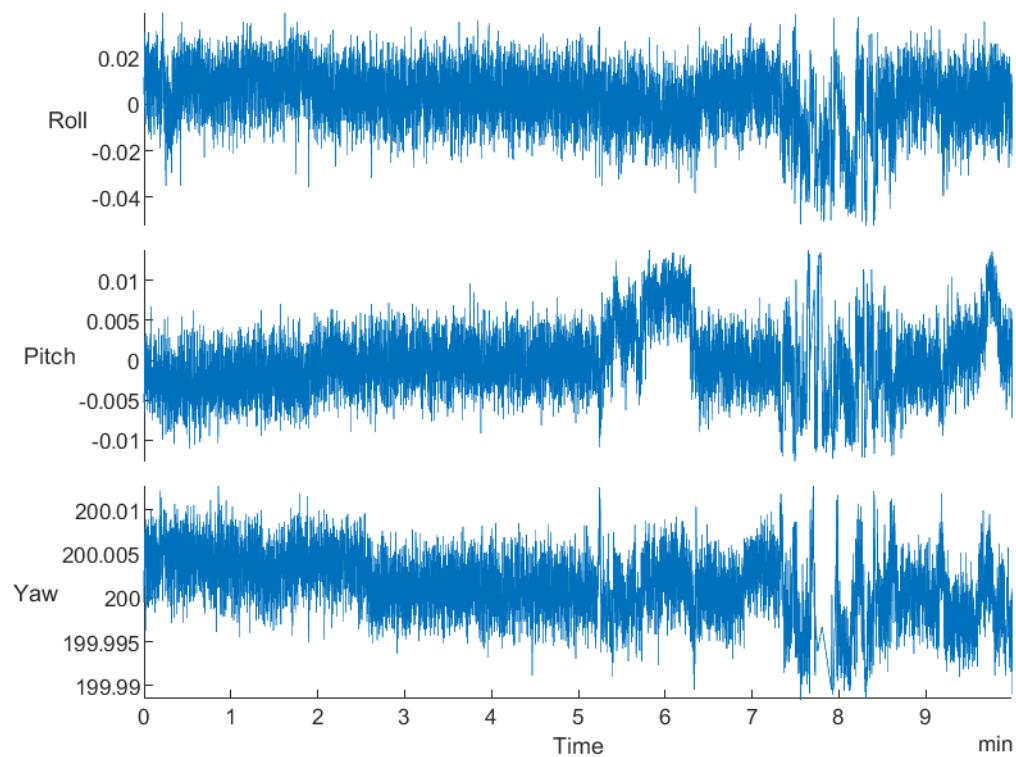


Figure A.4: Stacked axes of reference Optitrack signal for static test.

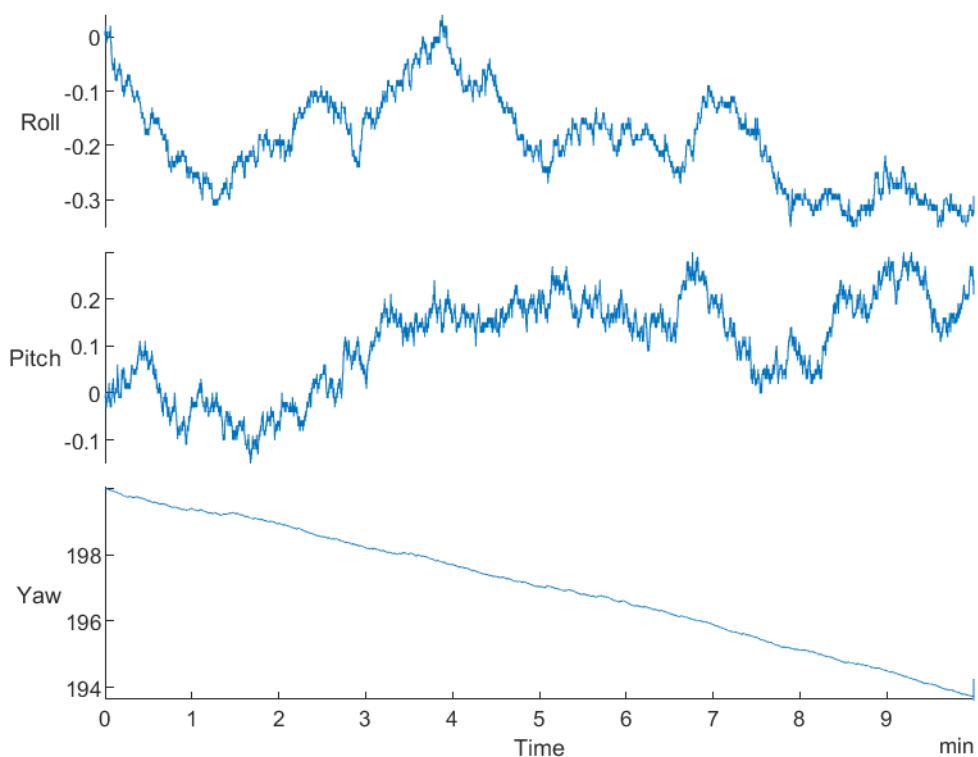


Figure A.5: Stacked axes of integrated gyro signal for static test.

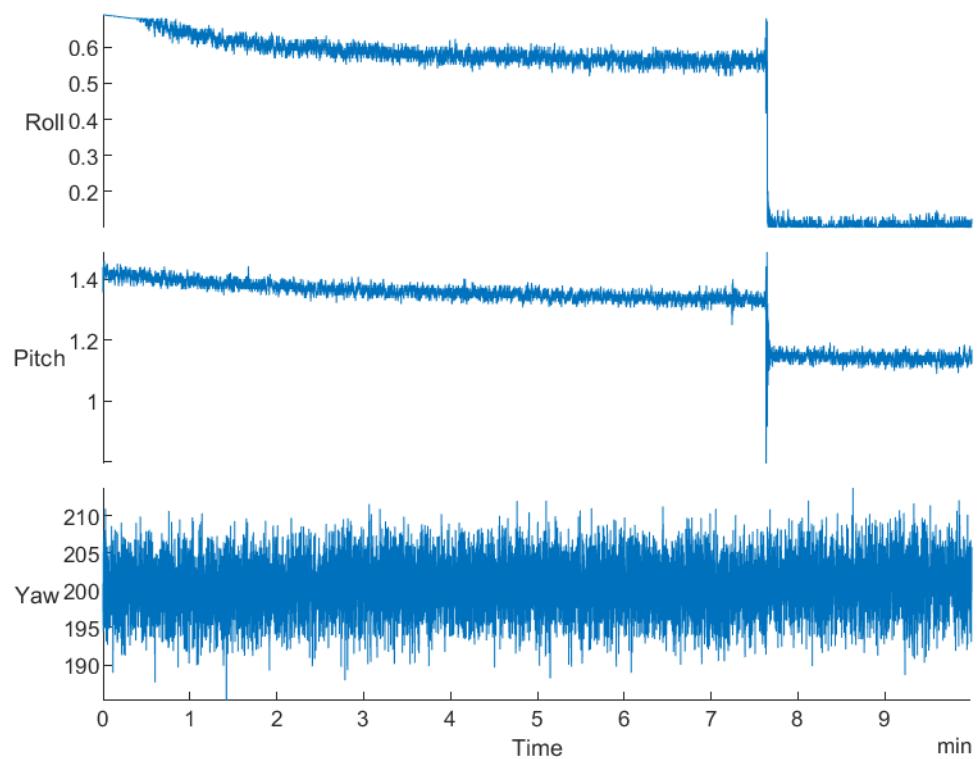


Figure A.6: Stacked axes of complementary filter signal for static test.

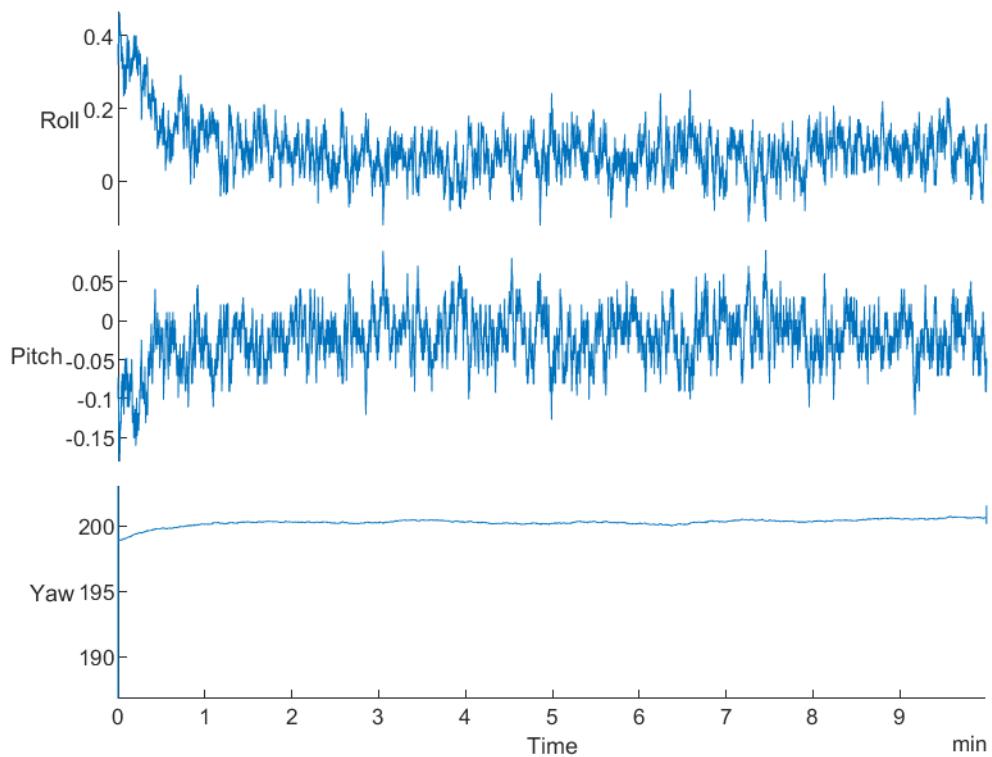


Figure A.7: Stacked axes of Mahony filter signal for static test.

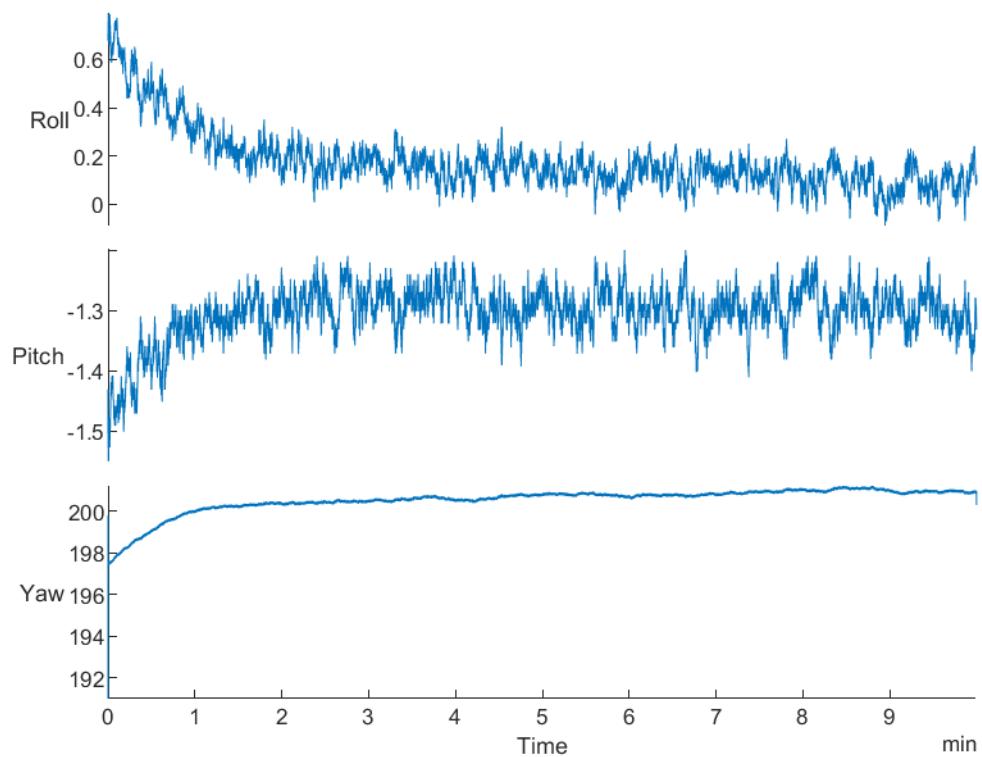


Figure A.8: Stacked axes of Madgwick filter signal for static test.

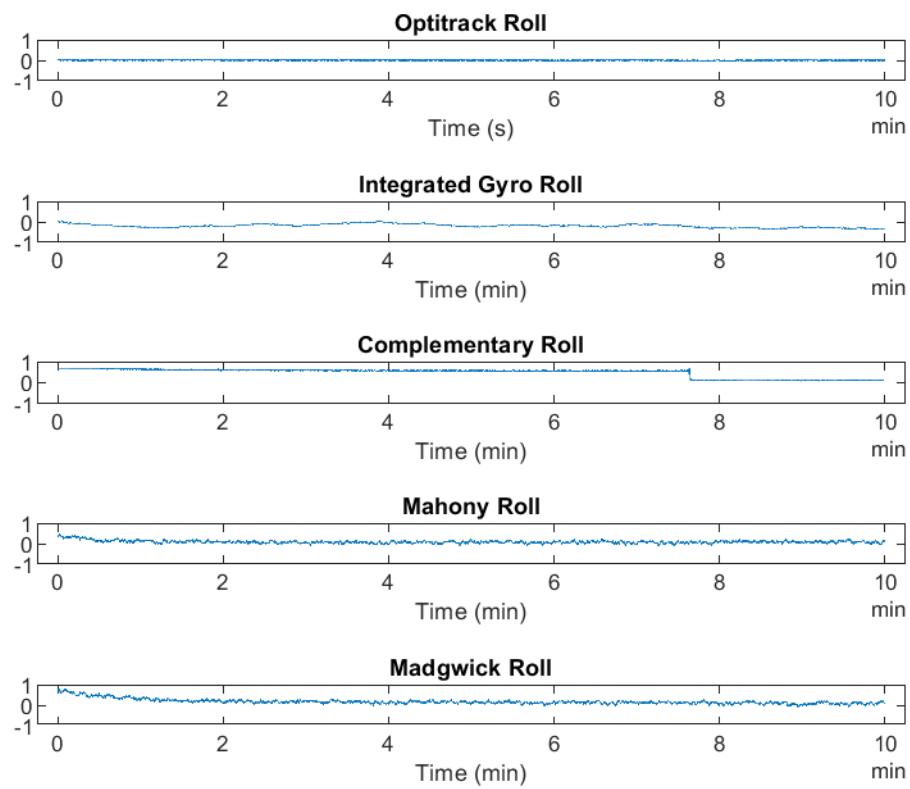


Figure A.9: Tiled plot of roll axes for static test.

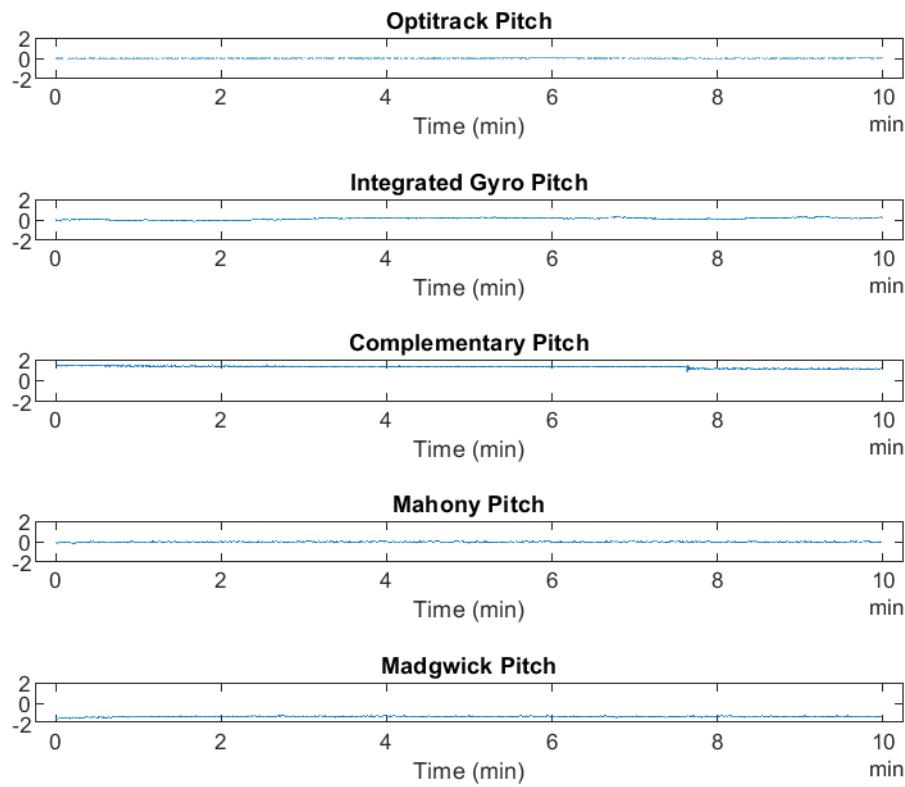


Figure A.10: Tiled plot of pitch axes for static test.

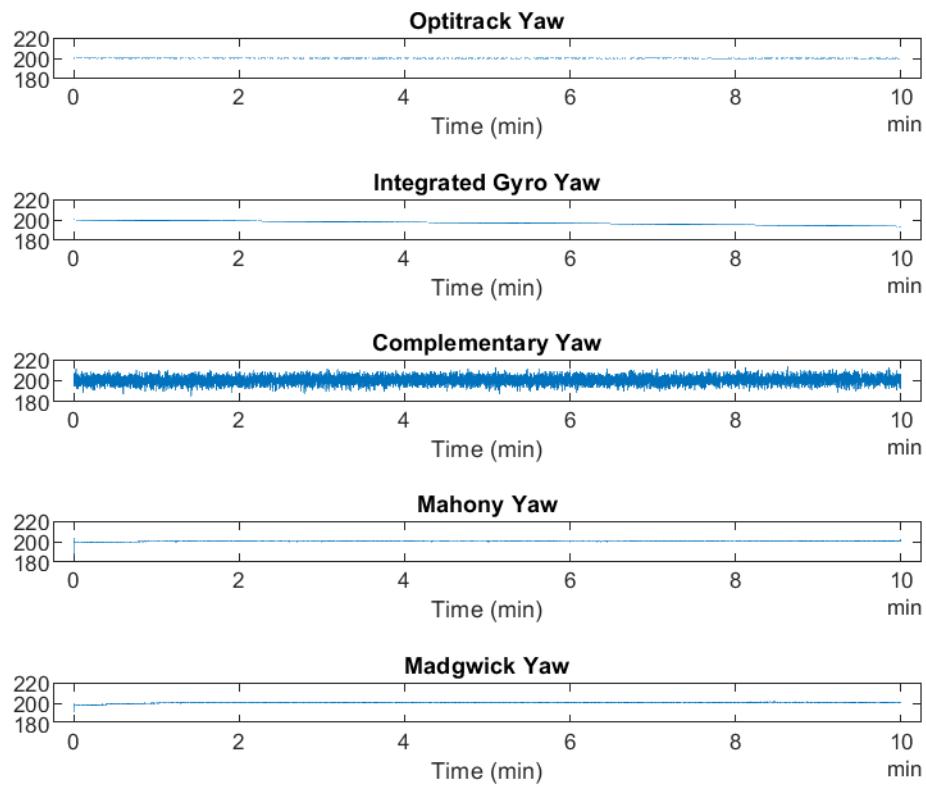


Figure A.11: Tiled plot of yaw axes for static test.

Static navigation tests: combined plots

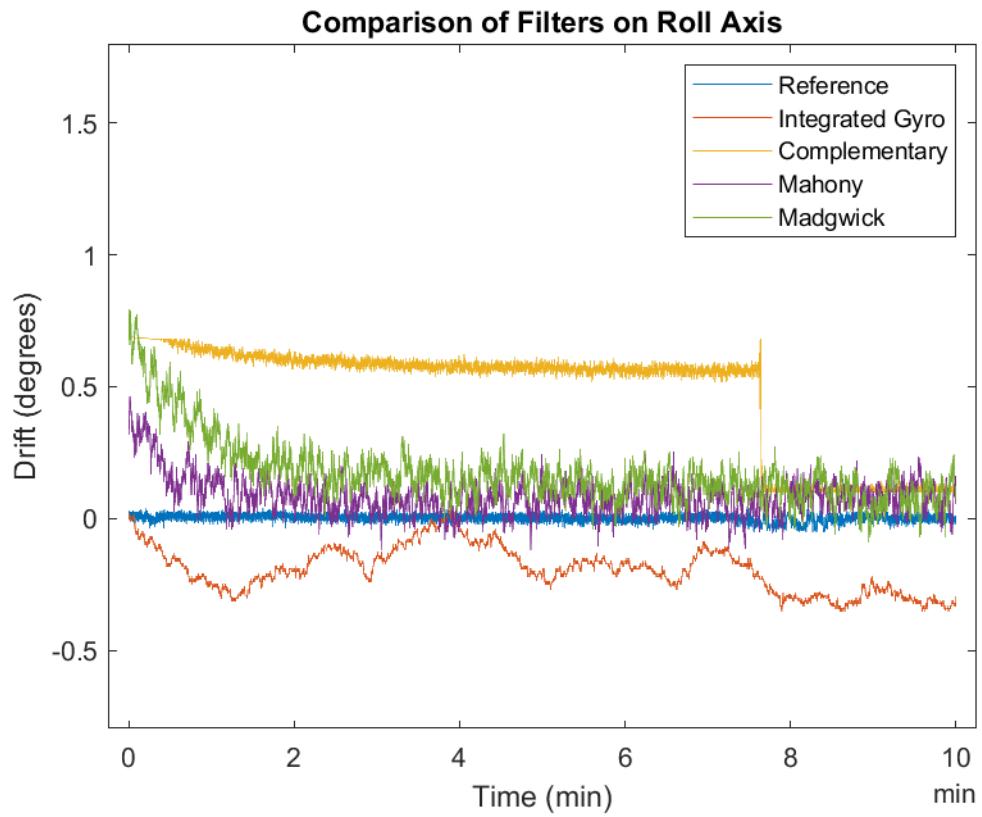


Figure A.12: Combined plot of roll axes for static test.

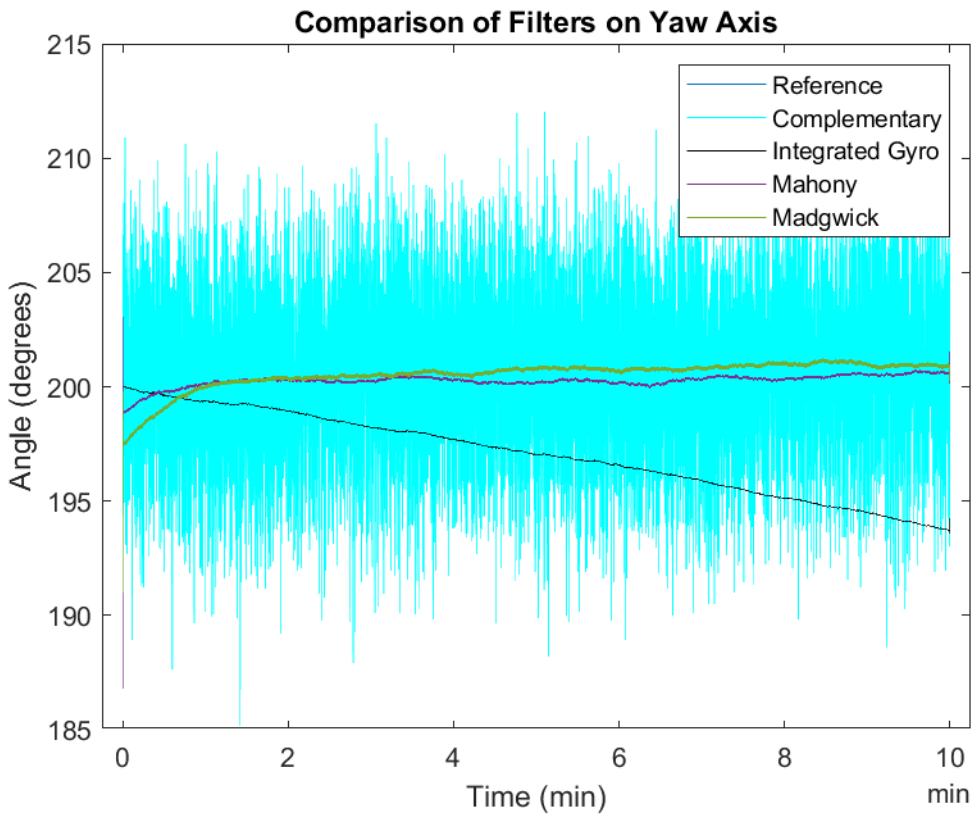


Figure A.13: Combined plot of yaw axes for static test.

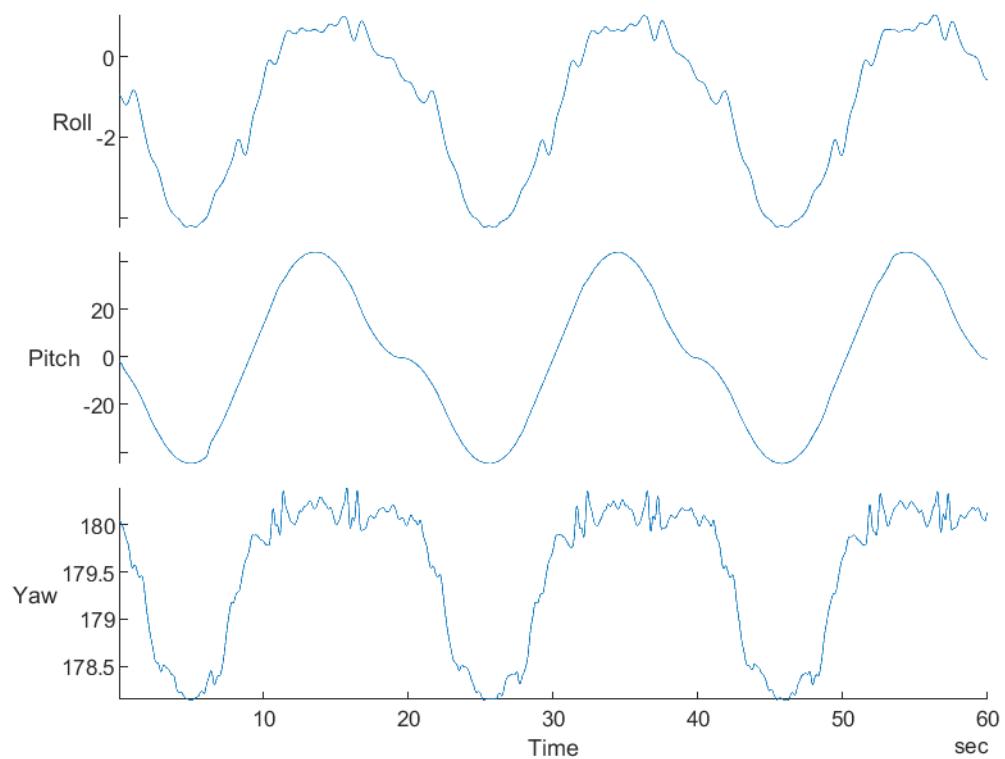


Figure A.14: Stacked axes of reference Optitrack signal for motion test.

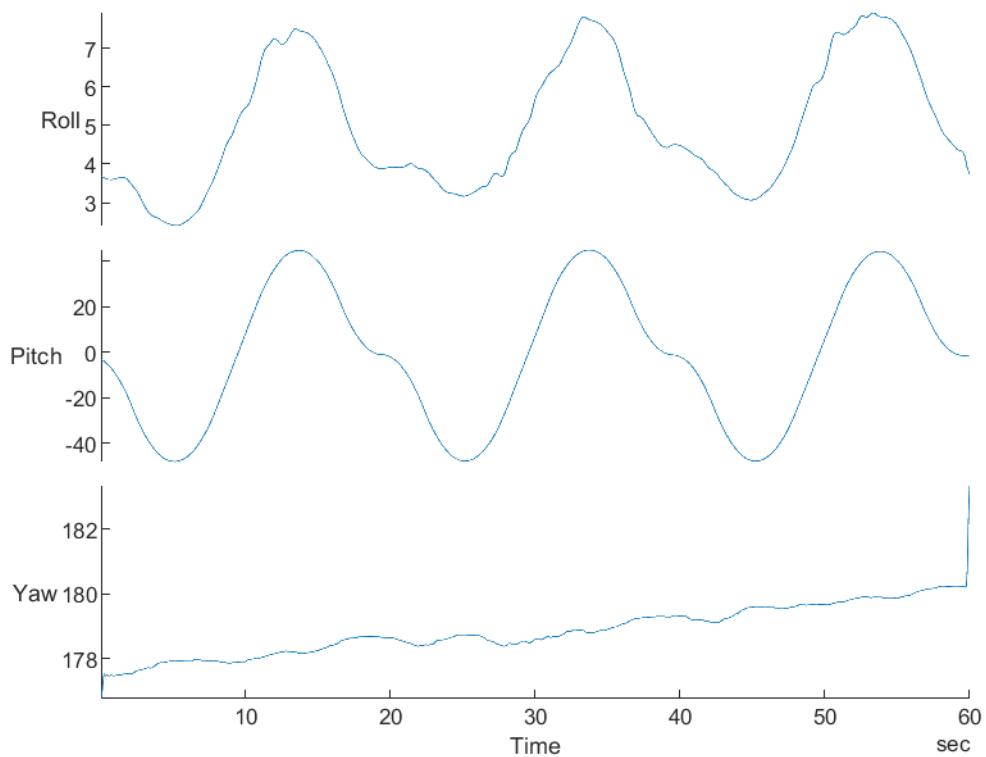


Figure A.15: Stacked axes of integrated gyro signal for motion test.

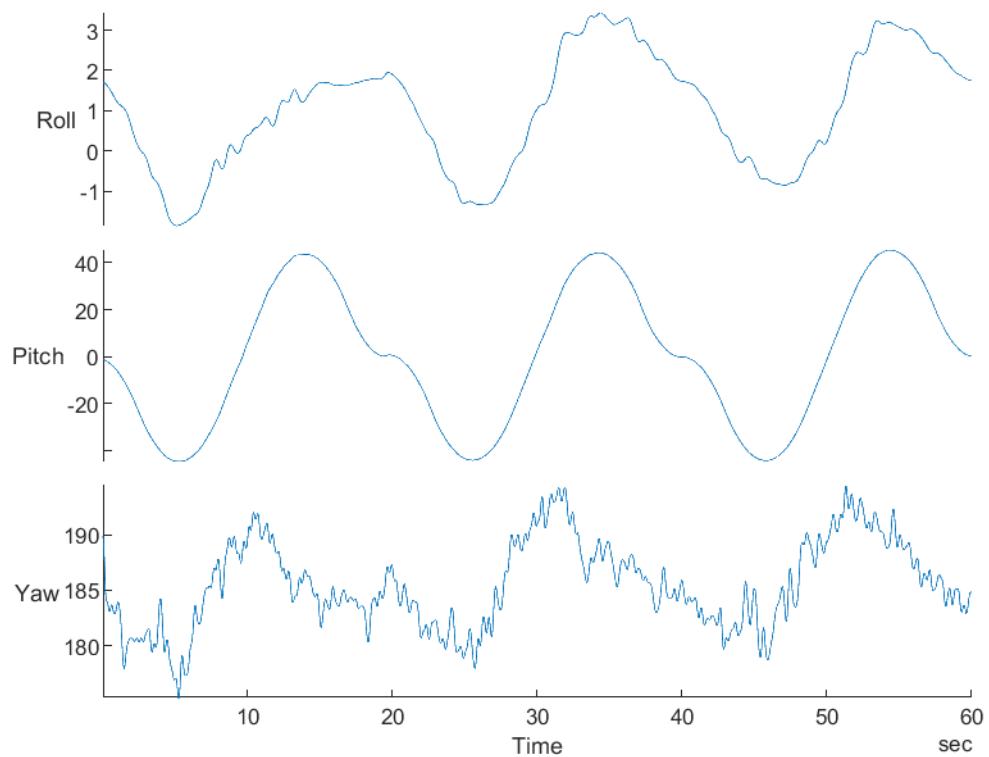


Figure A.16: Stacked axes of complementary filter signal for motion test.

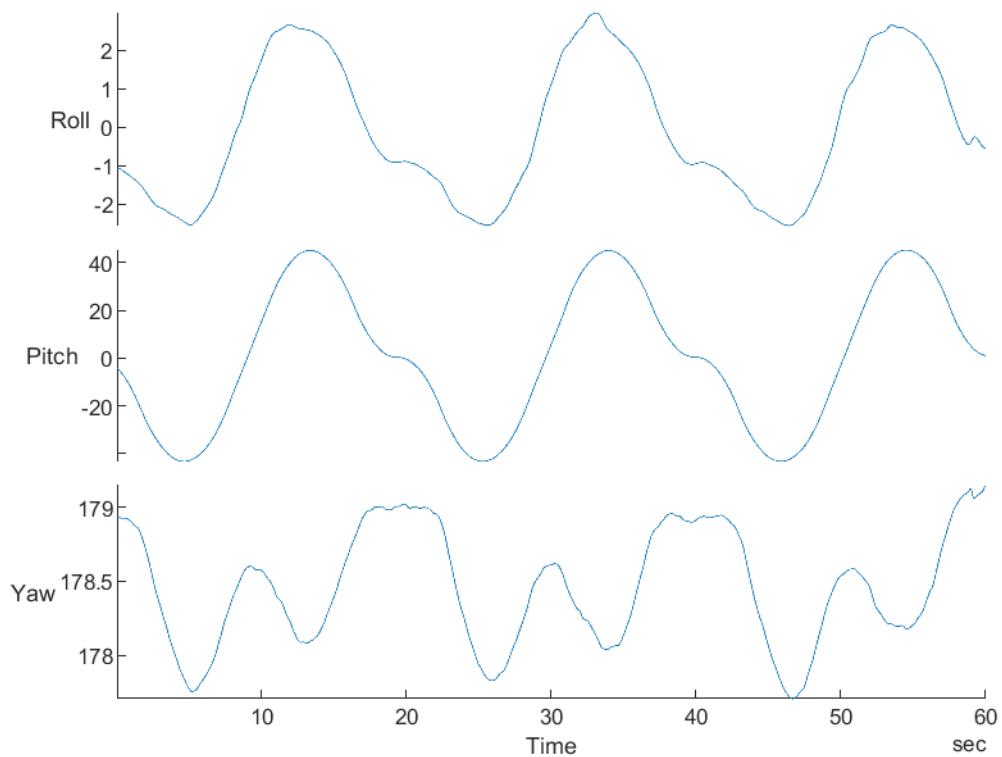


Figure A.17: Stacked axes of Mahony filter signal for motion test.

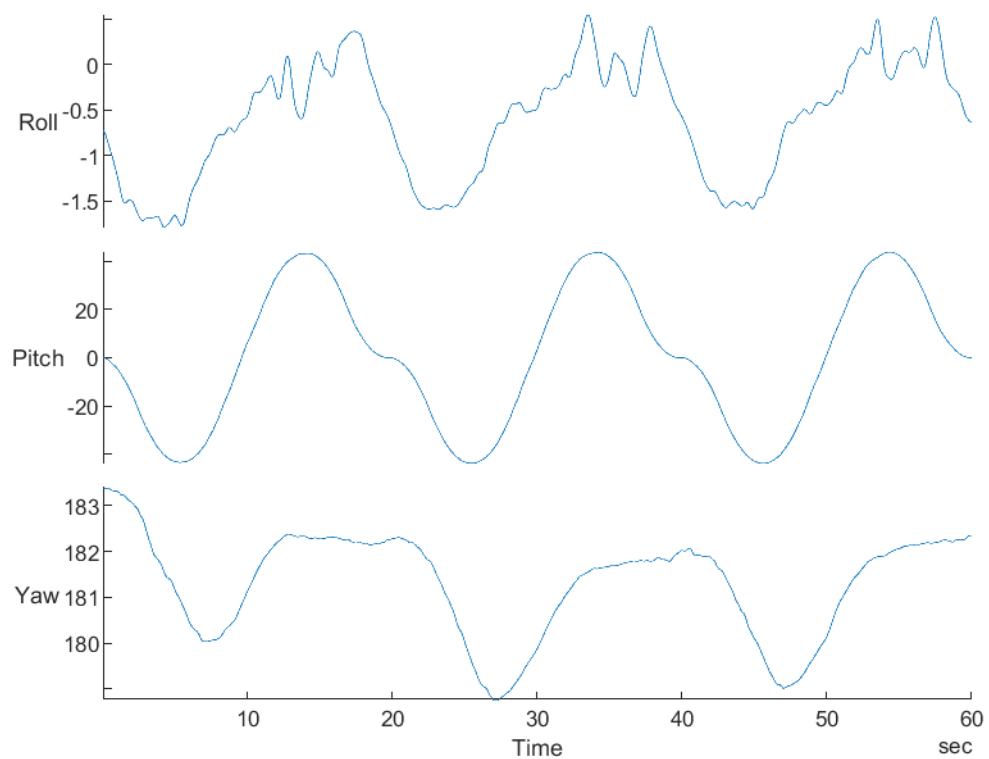


Figure A.18: Stacked axes of Madgwick filter signal for motion test.

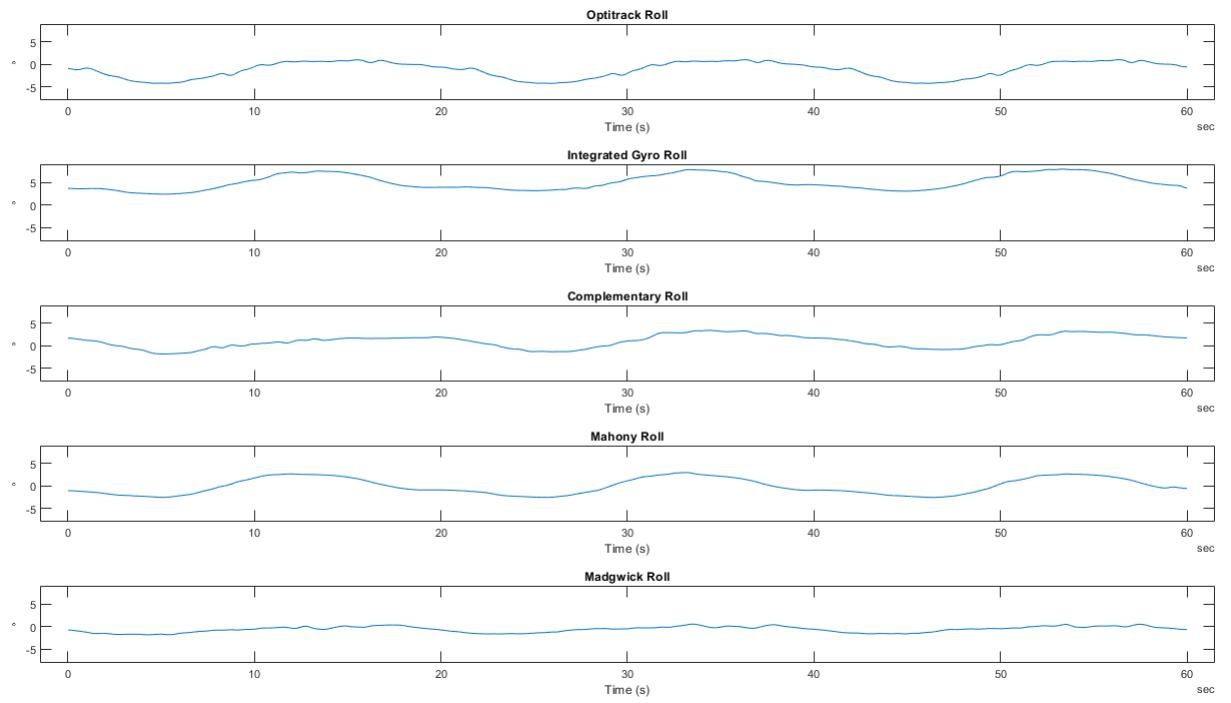


Figure A.19: Tiled plot of roll axes for motion test.

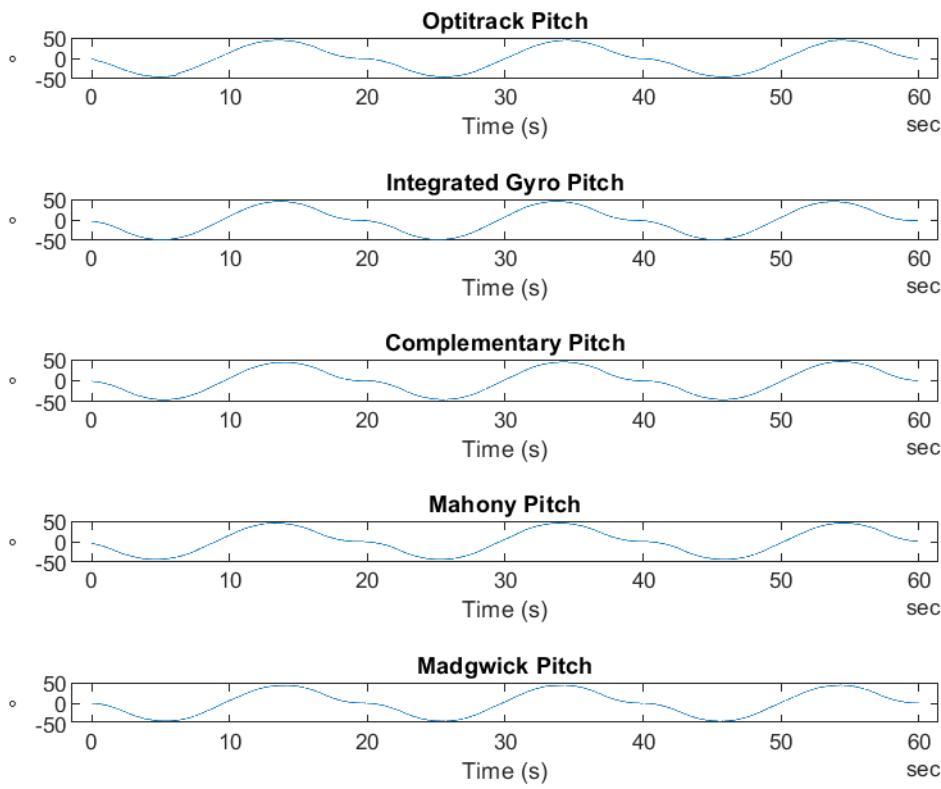


Figure A.20: Tiled plot of pitch axes for motion test.

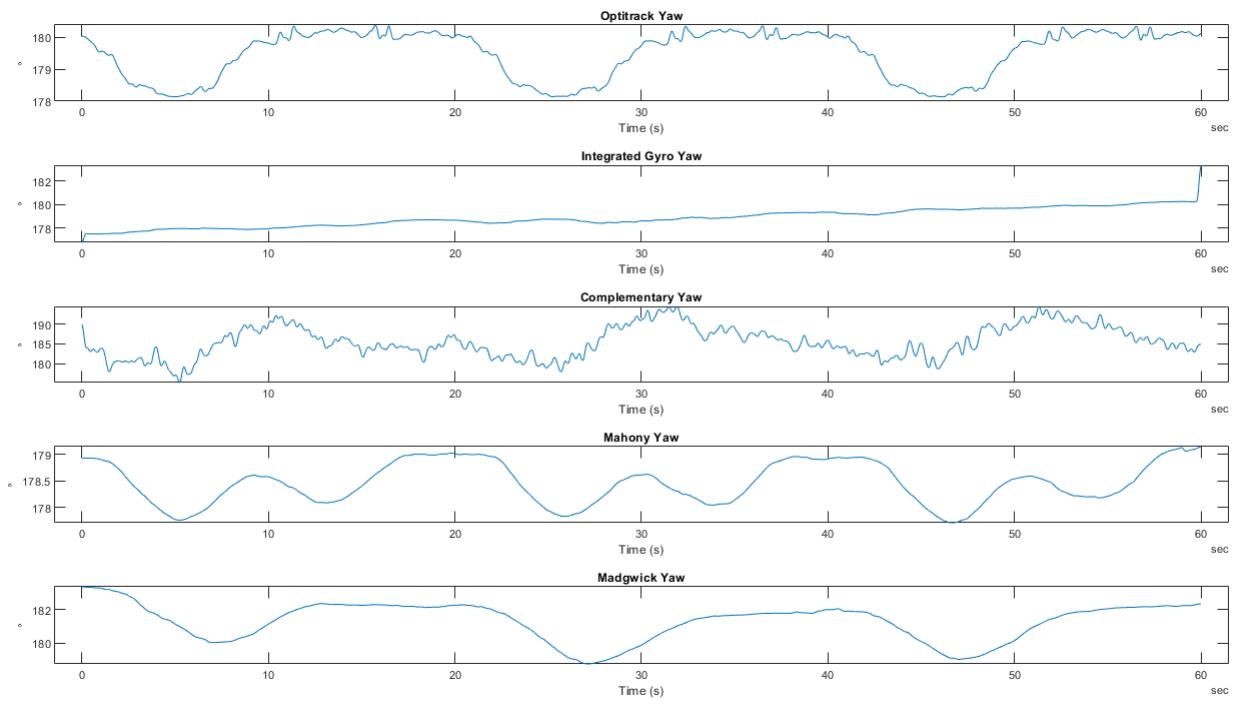


Figure A.21: Tiled plot of yaw axes for motion test.

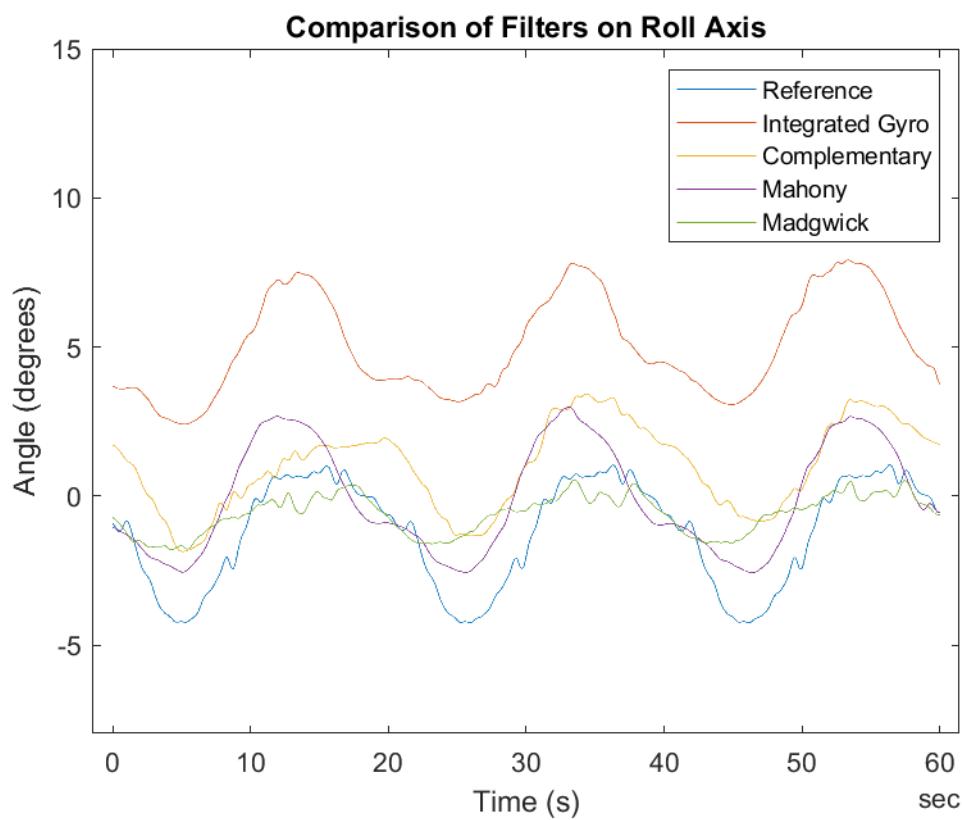


Figure A.22: Combined plot of roll axes for motion test.

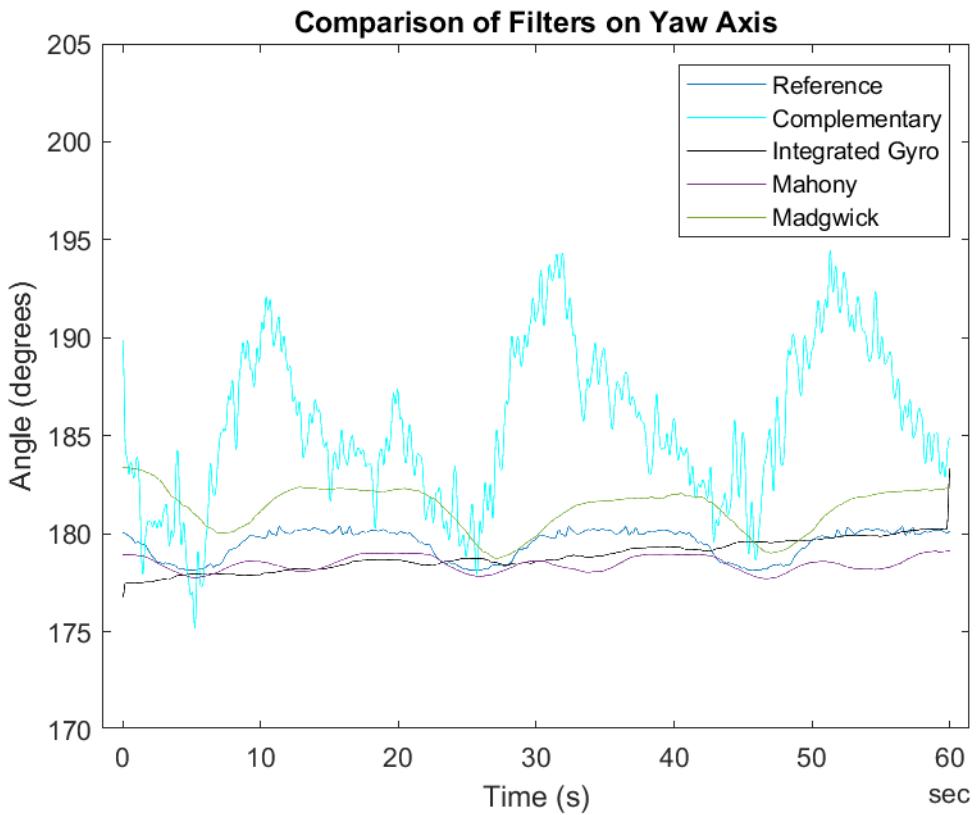


Figure A.23: Combined plot of yaw axes for motion test.

A.0.5 Model Validation

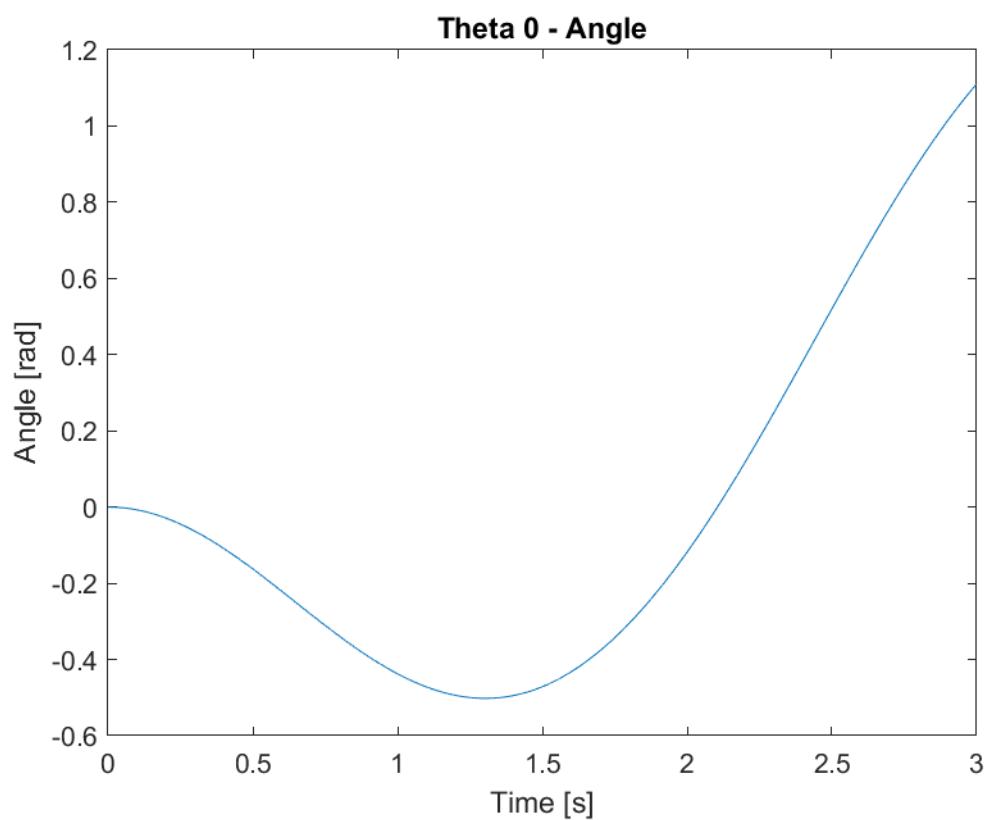


Figure A.24: Model validation applied forces: body 0 angle.

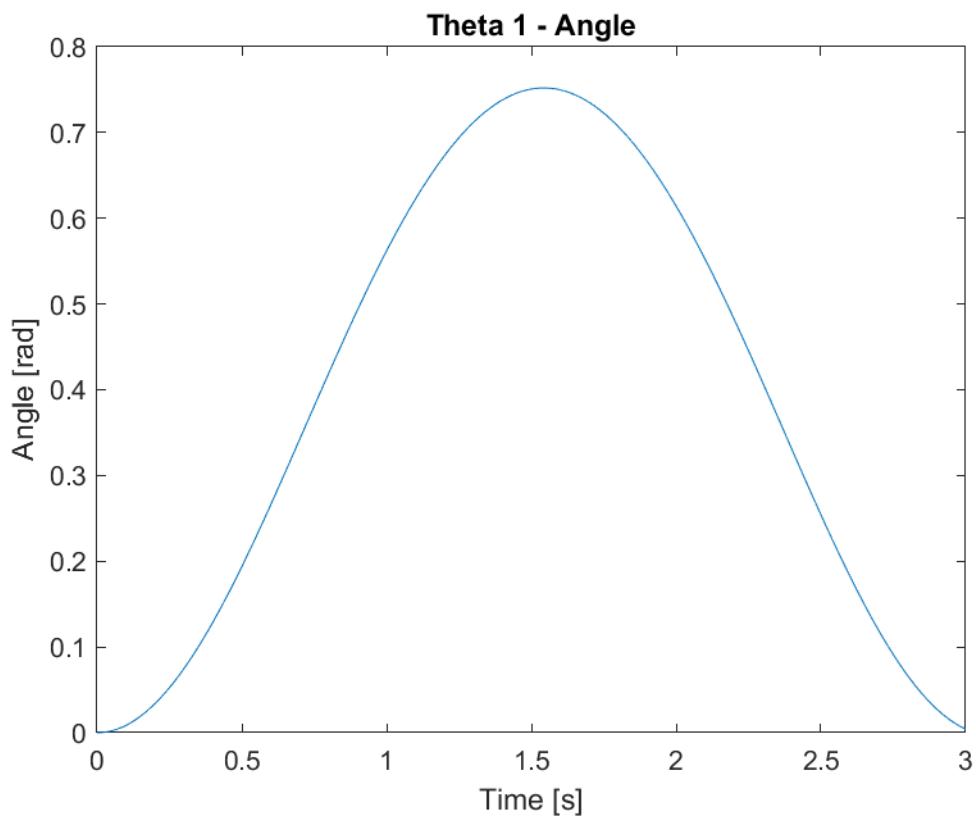


Figure A.25: Model validation applied forces: body 1 angle.

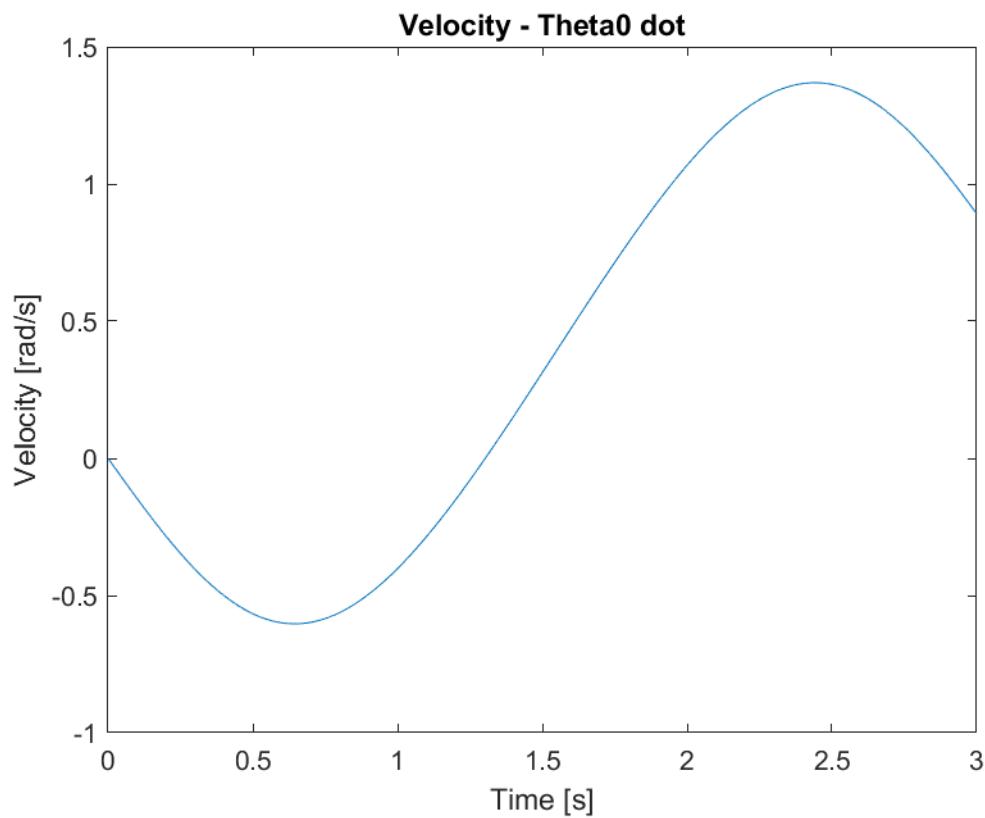


Figure A.26: Model validation applied forces: body 0.

Bibliography

- [1] In: *Association of Air Force Missileers* Volume 5, Number 4 (1997), page 6. URL: <http://afmissileers.com/newsletters/NL1997/Dec97.pdf>.
- [2] M. H. Ang and V. D. Tournassis. “Singularities of Euler and Roll-Pitch-Yaw Representations.” In: *IEEE Transactions on Aerospace and Electronic Systems* AES-23.3 (1987), pages 317–324. DOI: [10.1109/TAES.1987.310828](https://doi.org/10.1109/TAES.1987.310828).
- [3] Billur Barshan and Hugh F Durrant-Whyte. “Inertial navigation systems for mobile robots.” In: *IEEE transactions on robotics and automation* 11.3 (1995), pages 328–342.
- [4] A Cirillo et al. “A comparison of multisensor attitude estimation algorithms.” In: *Multisensor Attitude Estimation: Fundamental Concepts and Applications* (2016), pages 529–539.
- [5] Anton H De Ruiter, Christopher Damaren, and James R Forbes. *Spacecraft dynamics and control: an introduction*. John Wiley & Sons, 2012.
- [6] Alan A Desrochers. *Intelligent robotic systems for space exploration*. Volume 168. Springer Science & Business Media, 2012.
- [7] Mark Euston et al. “A complementary filter for attitude estimation of a fixed-wing UAV.” In: *2008 IEEE/RSJ international conference on intelligent robots and systems*. IEEE. 2008, pages 340–345.
- [8] Emil Fresk and George Nikolakopoulos. “Full quaternion based attitude control for a quadrotor.” In: *2013 European control conference (ECC)*. IEEE. 2013, pages 3864–3869.
- [9] Marvin Hobbs. *Basics of missile guidance and space techniques*. Wildside Press LLC, 2010.
- [10] Weiduo Hu. *Fundamental spacecraft dynamics and control*. John Wiley & Sons, 2015.
- [11] Manon Kok, Jeroen D Hol, and Thomas B Schön. “Using inertial sensors for position and orientation estimation.” In: *arXiv preprint arXiv:1704.06053* (2017).
- [12] Anthony Lawrence. “Gyro and Accelerometer Errors and Their Consequences.” In: *Modern Inertial Technology*. Springer, 1998, pages 25–42.

- [13] Sebastian Madgwick. “An efficient orientation filter for inertial and inertial/magnetic sensor arrays.” In: *Report x-io and University of Bristol (UK)* 25 (2010), pages 113–118.
- [14] Photis Patonis et al. “A fusion method for combining low-cost IMU/magnetometer outputs for use in applications on mobile devices.” In: *Sensors* 18.8 (2018), page 2616.
- [15] André Preumont. *Vibration control of active structures: an introduction*. Volume 246. Springer, 2018.
- [16] Ilmar Ferreira Santos. *Dinâmica de sistemas mecânicos: modelagem, simulação, visualização, verificação*. Makron, 2001.
- [17] Robson Oliveira dos Santos. *SensorFusion*. <https://github.com/aster94/SensorFusion>. 2020.
- [18] D. A. Smallwood and L. L. Whitcomb. “Model-based dynamic positioning of underwater robotic vehicles: theory and experiment.” In: *IEEE Journal of Oceanic Engineering* 29.1 (2004), pages 169–186. DOI: 10.1109/JOE.2003.823312.
- [19] George P Sutton and Oscar Biblarz. *Rocket propulsion elements*. John Wiley & Sons, 2016.
- [20] Travis S Taylor. *Introduction to rocket science and engineering*. CRC Press, 2017.
- [21] Fei Wang et al. “Flight dynamics modeling of coaxial rotorcraft UAVs.” In: *Handbook of Unmanned Aerial Vehicles* (2014), pages 1217–1256.
- [22] James R Wertz. *Spacecraft attitude determination and control*. Volume 73. Springer Science & Business Media, 2012.
- [23] Oliver J. Woodman. *An introduction to inertial navigation*. Technical report UCAM-CL-TR-696. University of Cambridge, Computer Laboratory, August 2007. URL: <https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-696.pdf>.
- [24] Rafael Yanushevsky. *Modern missile guidance*. CRC Press, 2018.

DTU Electrical Engineering
Department of Electrical Engineering
Technical University of Denmark
Ørsted Plads
Building 348
DK-2800 Kgs. Lyngby
Denmark
Tel: (+45) 45 25 38 00
www.elektro.dtu.dk