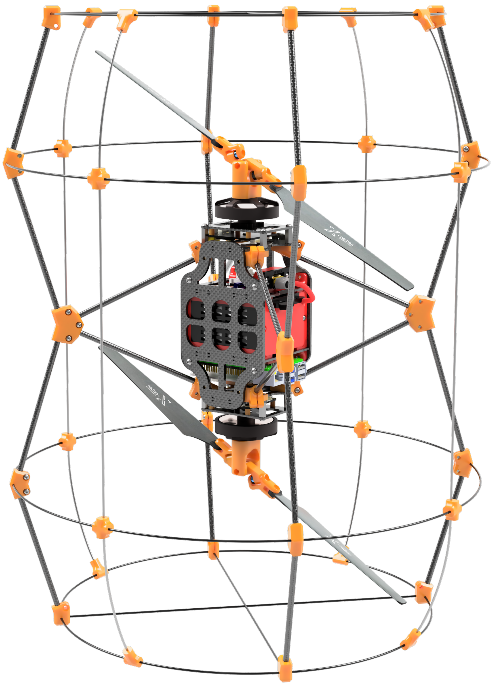
Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

Håvard Brenne

Design, Modeling, Control and Integration of Thrust Vectoring Rotors in Micro Aerial Vehicles

Master's thesis in Cybernetics and Robotics
Supervisor: Prof. Dr. Kostas Alexis
Co-supervisor: Paolo De Petris
June 2023



NTNU
Norwegian University of
Science and Technology

Håvard Brenne

Design, Modeling, Control and Integration of Thrust Vectoring Rotors in Micro Aerial Vehicles

Master's thesis in Cybernetics and Robotics
Supervisor: Prof. Dr. Kostas Alexis
Co-supervisor: Paolo De Petris
June 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

Preface

This report contains my Master's thesis, carried out as a requirement for the Masters degree program in Cybernetics and Robotics at the Norwegian University of Science and Technology. The project was completed in association with the Autonomous Robots Lab at the same institution. This lab is dedicated to the development of resilient and autonomous robotics systems operating in complex environments over extended periods.

The project carried out for my Master's thesis is a continuation of my earlier specialization project. The total goal of both of these parts have been to investigate and develop required systems for thrust vectoring rotors in micro aerial vehicles. In the specialization project, I spent effort to develop and analyze a basic thrust vectoring rotor and the theory behind its operation. In this Master's thesis, results from the specialization project were used to improve the rotor system, as well as design a micro aerial vehicle where it is integrated. Some sections of this report overlap with this previous work. These sections have been indicated accordingly.

Acknowledgements

I would like to express my gratitude to my supervisor Professor Kostas Alexis, and co-supervisor Ph.D. candidate Paolo De Petris for the guidance, support, and encouragement throughout the duration of this project. I am thankful for the opportunity to work on such an exciting project and for the resources provided to me to implement it. I wish to extend special thanks to co-supervisor Paolo De Petris for his dedication to the project, assisting in late-hour rotorcraft flights, as well as keeping track of a large amount of orders. I am also grateful for the help gotten from members of the Autonomous Robots Lab at NTNU throughout the project. Many discussions were had with members of the lab, which helped my grasp of underlying theory as well as critical evaluation of the methodologies I employed. I would like to specifically thank Ph.D. candidate Jørgen Anker Olsen for assistance with setting up sensors for experiments.

Abstract

This thesis covers the modeling, control and implementation of a torque modulated, thrust vectoring rotor, from simulation, to rotor design and control, to integration in micro aerial vehicles. The thrust vectoring rotor concept allows control of the thrust vector by tilting the tip path plane of the rotor, using only a single actuator. This control is achieved through an articulated rotor with passive hinges, which are exploited by harmonic motor torque modulation to control the tip path plane of the rotor.

The rotor model developed in this work accounts for non-linear aerodynamic effects as well as accurate dynamics of the rotor articulation. Its ability to predict thrust vectoring with harmonic motor control is demonstrated by comparing simulations to measurements from a physical implementation of the rotor. Given the predictive accuracy, the model has potential for future advancements in rotor design and control.

A field oriented motor control approach is used to implement a torque modulation control law in mechanical terms, compared to modulation in electrical terms in previous work. Implementing sinusoidal torque modulation on custom motor driver firmware allowed high rate harmonic modulation. A notable benefit here is a decoupling of the update rate neccesary for motor commands, and the control rate neccesary for the harmonic control. This facilitates slower update rates in motor commands, and could be especially useful for smaller rotors that operate at higher rotational speeds.

The physical rotor was designed with geometry and materials capable of large thrust vectoring angles. The final design, machined in high performance plastic, combined with the motor control strategy, enabled demonstration of thrust vector elevation angles up to $30°$, surpassing results of previous work.

A simplified model suitable for control allocation is derived. The simplified model is used to develop unconstrained control allocation for a general rotorcraft with an arbitrary number of thrust vectoring rotors. This framework has been successfully implemented in open-source autopilot software, where rotor arrangements can be configured in a user interface.

A coaxial rotorcraft with two thrust vectoring rotors was designed to demonstrate the thrust vectoring rotor in flight. Basic hovering, flight control and great efficiency were demonstrated with the implemented systems. As further flight experiments and control strategies should be explored, this rotorcraft and thrust vectoring system could serve as a platform for future investigation and applications of thrust vectoring rotors and fully-actuated micro aerial vehicles.

Sammendrag

Denne masteroppgaven omhandler modellering, regulering og implementering av en rotor der retningen på løftekraften kan styres med dreiemomentmodulering. Videre tar oppgaven for seg simulering, rotor design og styring, og integrering i små luftfartøy. Rotorkonseptet tillater styring av rotorens stilling og rotasjonshastighet ved hjelp av kun en enkelt aktuator. Denne styringen oppnås gjennom harmonisk motorstyring med en rotor som har strategisk plasserte passive hengsler. Dette kan være svært nyttig for å oppnå bedre effektivitet og manøvreringsegenskaper for luftfartøy.

En modell av denne rotoren er utviklet i denne oppgaven. Den tar hensyn til ulineær aerodynamikk og en nøyaktig modell av rotorens bevegelseslikninger. Dens evne til å forutsi retningen til løftekraften med harmonisk motorstyring demonstreres ved å sammenligne simuleringer med målinger av en fysisk implementering av rotoren. Gitt den prediktive nøyaktigheten, har modellen potensial for fremtidige fremskritt innen rotordesign og regulering.

En feltorientert motorstyringsmetode brukes for å regulere harmonisk modulering av motorens dreiemoment. Egenutviklet fastvare for motordriveren tillot høyfrekvent harmonisk regulering. En fordel ved å implementer harmonisk modulering på motordriveren er at oppdateringsfrekvensen til reguleringskommandoer kan settes uavhenging av ønsket oppdateringsfrekvens for harmonisk modulering, noe som kan være spesielt nyttig for små rotorer som opererer ved høye rotasjonshastigheter.

Den fysiske rotoren ble designet med geometri og materialer som er i stand til å oppnå store vinkler i styring av løftekraften. Det endelige designet ble maskinert i høytytende plast. Sammen med motorstyringsstrategien muliggjorde dette en demonstasjon av regulering av løftekraftretningen i opp til 30 grader i elevasjonsvinkel.

En forenklet modell egnet for kontrollallokasjon er utledet. Denne modellen brukes for å utvikle et rammeverk for ubegrenset kontrollallokering for et generelt luftfartøy med et vilkårlig antall rotorer med styrbar løftekraftretning. Dette rammeverket ble implementert i autopilotprogramvare, der rotoroppsettet kan konfigureres i et brukergrensesnitt.

Et luftfartøy med to rotorer med styrbar løftekraftretning ble designet for å demonstrere den styrbare rotoren under flyvning. Grunnleggende flyveegenskaper og høy effektivitet ble demonstrert med de implementerte systemene. Luftfartøyet og systemene for styrbar løftekraftretning utviklet i denne oppgaven kan derfor brukes til videre undersøkelser innenfor små luftfartøy.

Table of Contents

List of Tables

List of Figures

Abbreviations

Abbreviation	Description
MAV	Micro Aerial Vehicle
BEMT	Blade Element Momentum Theory
BET	Blade Element Theory
MT	Momentum Theory
CFD	Computational Fluid Dynamics
DOF	Degree(s) of Freedom
URDF	Unified Robotics Description Format
FOC	Field oriented control
PMSM	Permanent magnet synchronous motor
BLDCM	Brushless direct current motor
PWM	Pulse width modulation
EMF	Electromotive force
CAD	Computer Aided Design
PI-controller	Proportional integral controller
CAN-FD	Controller area network flexible data-rate
RTOS	Real Time Operating System
FT	Force torque sensor
IMU	Inertial measurement unit
CW	Clockwise
CCW	Counter Clockwise

1

Introduction

This introduction derives from the introduction of the report for the associated specialization project.

The quadcopter configuration is a widely used micro aerial vehicle (MAV) configuration, both in industry and research. The simple mechanical design of only four moving parts is advantageous, especially compared to configurations such as the traditional helicopter with a mechanically complex swashplate mechanism. As MAVs are often low-cost, high-volume and used in less controlled environments than helicopters, the low maintenance and cost of the simple mechanical design is crucial. Other MAV configurations can offer advantages over the quadcopter, especially for specific applications. Adapting a cyclical pitch controllable rotor such as in the helicopter enables configurations with fewer, and comparatively larger rotors than that of the quadcopter. Designing a miniaturized swashplate mechanism is difficult. It requires extra actuators to control the blade pitch, in addition to miniaturized coupling and swashplate mechanism. This increases the mechanical complexity substantially, as well as the cost, weight and maintenance requirements. For a typical MAV helicopter configuration, even with fewer rotors, the total number of actuators will be at least four: One main motor, a tail rotor and at minimum two servo actuators to control the cyclical pitch of the main rotor. This does not include an additional servo actuator, which is required if control over cyclical pitch is desired.

An alternative mechanism exists for cyclical pitch control. Rather than using a mechanical swashplate with additional actuators, the blade pitch is cyclically controlled by the main motor. A specific hinged rotor articulation enables once-per-revolution harmonic motor torque modulation to induce a cyclical pitch angle. It is an underactuated system with no additional actuators or swashplate required, just simple, passive hinges and a sophisticated software control algorithm. The main concept consists of kinematically coupling blade lag and blade pitch through a skewed lag-pitch hinge. Acceleration of the motor causes blade lag, which couples to blade pitch. Having a positive lag-pitch coupling on one blade, and a negative lag-pitch coupling on the opposite blade allows a blade pitch difference. A desired blade pitch difference at positions within a rotation can be controlled by harmonically modulating the motor torque.

Using this cyclical blade pitch control mechanism could offer several advantages for MAVs. For a helicopter MAV configuration, the mechanism can replace the swashplate and the need of servo actuators, simplifying the mechanical system.

Other MAV configurations could also benefit from cyclical pitch control. Most multirotor configurations have all actuators in the same plane, such that a force can only be applied in one direction. Such underactuated rotorcraft can therefore only linearly accelerate in this direction, and needs to orient the whole rotorcraft in the direction of acceleration. With a specific hinged rotor and cyclical pitch control, a thrust vectoring rotor is possible. With two such rotors, it can be shown that a fully-actuated rotorcraft is possible. Actuation of force and moment can be done independently, such that acceleration and orientation can be decoupled. A fully-actuated rotorcraft can hover at non-zero angles and accelerate in any direction.

Previous fully-actuated MAVs have typically used servo motors to tilt the rotors. These servo motors could be replaced by the mentioned thrust vectoring rotors. Additionally, rotor tilting by a servo has been shown to have the inherent undesired property of non-minimum phase dynamics, fundamentally limiting the controller bandwidth[1]. The physical explanation given for this phenomenon is that the servo motor torque applied to tilt the entire rotor assembly causes an undesired moment in the opposite direction of the desired motion, regardless of the control method used. This undesirable moment should be reduced by only tilting the rotor blades.

To introduce the thrust vectoring concept studied in this thesis, a more intuitive overview of the concept is presented in the following section.

Additionally, a video demonstrating the thrust vectoring rotor as well as rotorcraft flight is provided here: `https://youtu.be/HBLV1LE9DL0`.

1.1 Rotor Thrust Vector Control Concept

This section has been included from the author's previous work in the related specialization project.

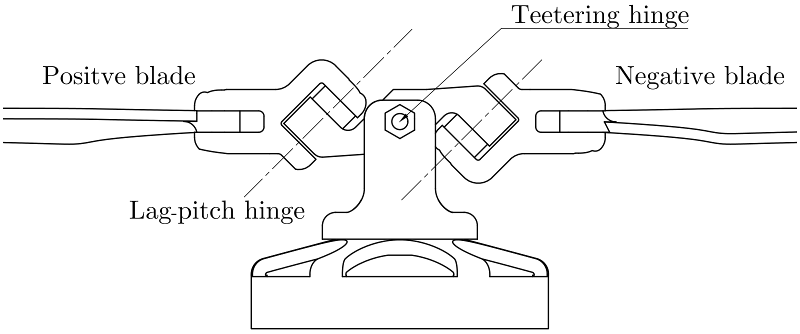
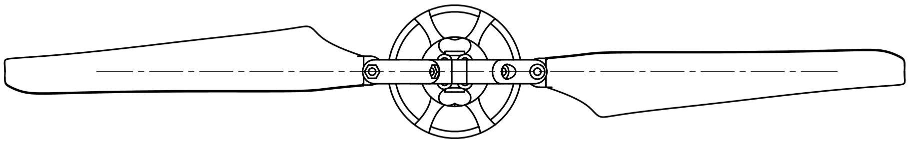
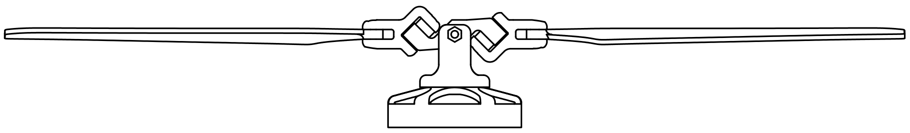


Figure 1.1: Rotor overview

Figure 1.1 shows the rotor, consisting of a motor, two blades, a central teetering hinge and two skewed lag-pitch hinges. The orientation of the skewed lag-pitch hinges is such that the blade marked as ”positive” obtains a positive coupling between blade lag and blade pitch, while the blade marked ”negative” obtains a negative coupling between blade lag and blade pitch. To illustrate the concept, a simplified thrust vector control sequence



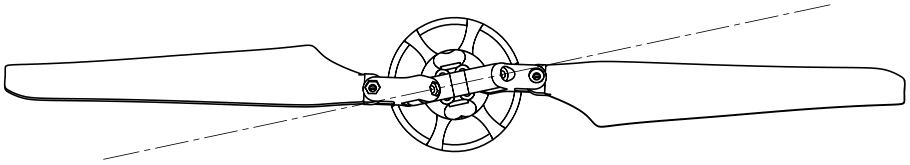
(a) Top view



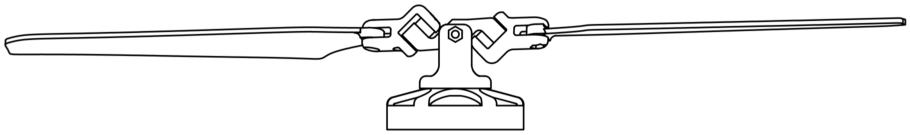
(b) Side view

(c) Front view with positive lag-pitch coupling blade**(d)** Back view with negative lag-pitch coupling blade

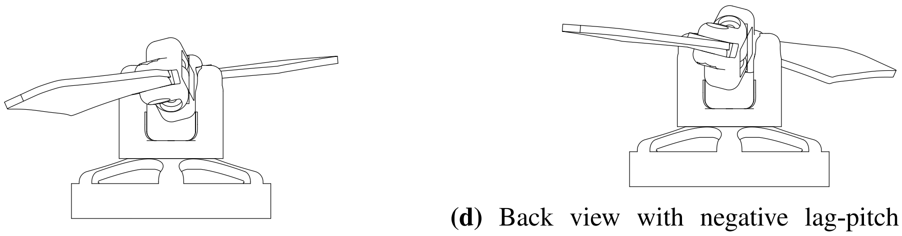
Figure 1.2: Overview of rotor configuration with hinge angles equal zero. This corresponds to a conventional rotor configuration without hinges



(a) Top view

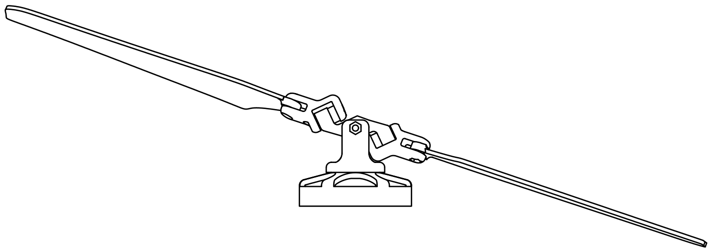


(b) Side view



(d) Back view with negative lag-pitch coupling
(c) Front view with positive lag-pitch coupling blade.blade.

Figure 1.3: Overview of rotor configuration with counter clockwise motor acceleration. The moment of inertia of the blades causes them to lag behind with a lag-pitch hinge angle. The positive blade obtains a higher pitch, while the negative blade obtains a lower pitch



(a) Side view

Figure 1.4: Side view of rotor configuration with tilted thrust vector. The difference in blade pitch causes a difference in aerodynamic force between the blades. The side hub assembly rotates along the teetering hub, tilting the thrust disk.

is presented.

Step 1: For simplicity, Figure 1.2 shows the rotor with hinge angles equal to zero, under the assumption that this is an equilibrium when rotating at a constant speed.

Step 2: The motor accelerates counter clockwise in Figure 1.3, causing both blades to lag behind the motor position due to their moment of inertia. This lag can be seen in Figure 1.3(a). As the blade lag is coupled to the blade pitch, a resulting increased blade pitch can be seen for the positive blade in Figure 1.3(c) and the corresponding left blade in Figure 1.3(b). A decrease in blade pitch can be seen for the negative blade in Figure 1.3(d) and the corresponding right blade in Figure 1.3(b).

Step 3: The aerodynamic lift is proportional to the blade pitch, such that the thrust generated by the positive blade increases, and opposite for the negative blade. As the blades can freely rotate about the teetering hinge, this difference in thrust causes the blades to tilt around the teetering hinge, as seen in Figure 1.4.

Let Figure 1.4 illustrate the maximum teetering angle. To control a constant thrust vector direction, the teetering angle needs to follow a sinusoidal trajectory with respect to the motor position. By decelerating the motor, the blades will lead the motor position due to their moment of inertia. Thus, the opposite action to what was explained previously occurs, the positive blade pitch decreases, while the negative blade pitch increases.

1.2 Related Work

Actuating forces and moments in a rotor without a swashplate have been studied with different concepts. Wu, Tang, Quan, *et al.* [2] uses a magnetic coil to actuate cyclical pitch. The focus of this thesis is on torque modulated cyclical pitch. Paulos and Yim [3] derives a linear model verified with experimental results of a rotor with cyclical pitch control through torque modulation. Here, individual blade flap hinges are used instead of the teetering hinge shown in Section 1.1. This design and the linear model are demonstrated for multiple rotor scales, for a rotor with a diameter of 0.1 m and 1 m in [4]. In later work, the hinge design described in Section 1.1 was demonstrated [5]. Both the teetering hinge as well as the flapping hinge help relieve bending moments in the blade, allowing a degree of freedom. With the teetering hinge, moment transfers in the hub are nearly eliminated compared to the flapping hinge. This motivates the total contribution of the rotor to be approximated as a thrust vector perpendicular to the tip path plane without direct moment transfers.

Rotorcraft control has been demonstrated with this concept. Paulos, Caraher, and Yim [5] emulates a fully actuated rotorcraft by employing two thrust vectoring, torque modulated rotors in a coaxial configuration. Qin, Chen, Cai, *et al.* [6] demonstrates a bicopter design where thrust vectoring is constrained to only the roll axis to constrain the degrees of freedom similarly to standard multirotors. Chen, Kong, Xu, *et al.* [7] demonstrates flight with only a single torque-modulated rotor by giving up yaw authority. As a consequence, the rotorcraft starts rotating about the yaw axis, which is exploited for sensor field of view enhancement. In industry, torque modulated rotor control has also seen recent usage [8], [9].

Fully actuated rotorcraft have been investigated in many rotorcraft concepts. One solution is to place multiple motors at different angles to achieve the required degrees of freedom, as in [10]. Some have used servos to individually tilt rotors [11], [1], [12], [13]. Other designs have relied on synchronously tilting multiple rotors with each servo, reducing the number of servos neccesary, while losing degrees of freedom [14],[15].

In industry, fully actuated rotorcraft have been used in non destructive testing and contact inspection [16], [17].

1.3 Objectives

The main objectives of this thesis are:

- Developing a comprehensive model of the thrust vectoring rotor that accounts for non-linear aerodynamic forces as well as the non-linear rigid body dynamics and kinematics of the articulated rotor.
- Designing and implementing a high performance thrust vectoring rotor and all necessary systems for their integration in MAVs.
- Evaluating the thrust vectoring rotors in flight.

Developing a non-linear model of the thrust vectoring rotor is motivated by acquiring a better understanding of the fundamental properties of the system. The scope of this thesis is reduced to analyzing properties of this model, and comparing the results to a physical rotor.

In developing high-performance thrust vectoring rotors, relying on once-per-revolution torque modulation, accurate and performant motor control is important. This motivates the additional focus on motor control techniques in this thesis.

Integration and testing of the thrust vectoring rotor in an actual rotorcraft is critical, as this process may reveal effects that are not captured by test stands or simulations. This is the main motivation for developing a custom rotorcraft. As the undertaking of designing both the thrust vectoring rotor system and a suitable rotorcraft is substantial, there are limits in the scope of rotorcraft control. Primary focus is put towards the thrust vectoring rotor system, as well as its integration in an arbitrary rotorcraft configuration. Advanced rotorcraft control is not a main focus and rotorcraft testing is performed to evaluate basic functionality of the underlying rotor system.

1.4 Contributions

In this work, the author has made the following main contributions:

- Developed a model of a teetering, lag-pitch coupled rotor configuration with thrust vectoring.
- Developed a numerical implementation of the model. Simulated, analyzed and compared the model to a physical rotor.
- Designed and produced a thrust vectoring rotor, reaching higher maximum angles than previous implementations.
- Integrated a high performance (40kHz) torque modulation controller on custom motor driver firmware.
- Developed software to analyze, calibrate and control thrust vectoring.
- Derived both the actuator model and control allocation of general thrust vectoring rotors, and implemented this in open-source autopilot software.
- Designed, produced and flight tested a coaxial rotorcraft utilizing the aforementioned contributions.

The software source code as well as design and production files of the rotor and rotorcraft are available here:
`https://github.com/h-brenne/thrust_vector_control`

1.5 Mathematical Notation

The mathematical notation in this thesis follow these conventions:

- Vectors are written in bold lowercase, such as vector $\mathbf{v}$.
- Matrices are written in bold uppercase, such as matrix $\mathbf{A}$.
- Reference frames are written in calligraphy. An exception to this is the rotation matrix $\mathcal{R}$, where $\mathcal{R}_{\mathcal{A}}^{\mathcal{B}}$ describes the rotation from frame $\mathcal{A}$ to frame $\mathcal{B}$.

2

Theory

2.1 Rotor Dynamics

This section builds upon the aerodynamic theory presented in the authors previous work in the related specialization project. The theory has been significantly extended in this work to include a more comprehensive overview of rotor theory, aerodynamic effects as well as the methods of solving the resulting systems.

Rotor flight systems has had a rich history of use and extensive research and engineering. This field and especially helicopter main rotor systems, is very relevant when it comes to modeling and understanding the miniaturized hinged rotor studied in this thesis.

2.1.1 Rotor Systems

Helicopter rotor systems are usually divided into three main mechanical classifications: teetering rotor, articulated rotor and hingeless rotor [18]. The teetering rotor consists of a central teetering hinge, allowing typically two blades to flap synchronously. Lead lag hinges are employed to allow degrees of freedom for individual blade lead and lag. The articulated rotor trades the teetering hinge for individual blade flap hinges. The hingeless rotor uses no mechanical hinges and relies on the blade and structural elasticity to relive the generated bending moments.

The rotor system analyzed in this thesis is most similar to the teetering helicopter rotor system. The main difference lies in the skewed lag-pitch hinge, kinematically coupling lead-lag and pitch. Additionally, the actuator-controlled collective and cyclical pitch through a swashplate is eliminated.

Modeling of rotor systems can range from involving complex computational fluid dynamics simulations to simple analytical models that show basic properties.

The teetering dynamics are especially important for flight control. From an averaged model, it can be shown that for a perfectly hinged teetering rotor, blade pitch input causes a maximum flapping response with a 90 degree phase offset [19].

2.1.2 Aerodynamic flight

Aerodynamic forces are generated by an object moving relative to the air. In this rotor case, the moving object we wish to analyze in terms of aerodynamics is simplified to the blades, while other parts of the rotor are neglected in terms of aerodynamic forces. To model an object moving through air, we need to account for the properties of the atmosphere, which can be summarized by the ideal gas law,

$$p = \rho R T_{\text{air,}} \tag{2.1}$$

where p is the pressure, ρ is the air density, R is the ideal gas constant of air and T_{air} is the air temperature. The density and temperature of the atmosphere will in general vary with the altitude, and the pressure will vary accordingly as well. These variables will be important for the aerodynamic effects to be modeled.

2.1.3 Blade Element Momentum Theory

A common method used to model a spinning propeller blade has been to use blade element momentum theory(BEMT). This combines momentum theory (MT) and blade element theory (BET). Although BEMT does not capture all aerodynamic effects, the main effects are captured and results are satisfyingly close to reality.

Leutenegger, Hürzeler, Stowers, *et al.* [19] has been used as the main reference in this section. Some additional derivations are shown to allow for further understanding.

The most basic assumption of BEMT is a rotor with radius R that spins at a constant speed Ω.

Momentum Theory

In momentum theory the rotor is simplified to an infinitely thin disk that accelerates air. A tubular control volume is defined around the slipstream that passes through the propulsion disk, reaching far above and below the propulsion disk. This slipstream is visualized in Figure 2.1. The external inflow velocity at a far distance above the propulsion disk is given by v_∞. The external inflow velocity is in general given by the airspeed perpendicular to the propulsion disk. The propulsion disk, a boundary layer, is then assumed to produce a thrust force T that accelerates the air passing through the boundary layer to the sum of the external inflow velocity and the inflow velocity induced by the propeller v_i. At the far wake below the propulsion disk, the air velocity has a higher value v_w. The laws off mass, momentum and energy balance can be expressed for this control volume.

The equation for the mass flow through a cylinder [20, p. 419] is used. The mass flow through the infinitely thin propulsion disk, where the velocity is $v_\infty + v_i$, is given by

$$\dot{m}_2 = \rho v_2 A_2 = \rho(v_\infty + v_i)\pi R^2, \tag{2.2}$$

where w is the mass flow. The air density is considered constant, as the air is considered incompressible. By conservation of mass, there are equal mass flows $\dot{m}$ at each of the three boundaries considered

$$\dot{m} = \dot{m}_1 = \dot{m}_2 = \dot{m}_3 \tag{2.3}$$

The momentum balance equation of the control volume can be reduced to

$$f = \dot{m}_3 v_w - \dot{m}_1 v_\infty = \dot{m}(v_w - v_\infty) = T, \tag{2.4}$$

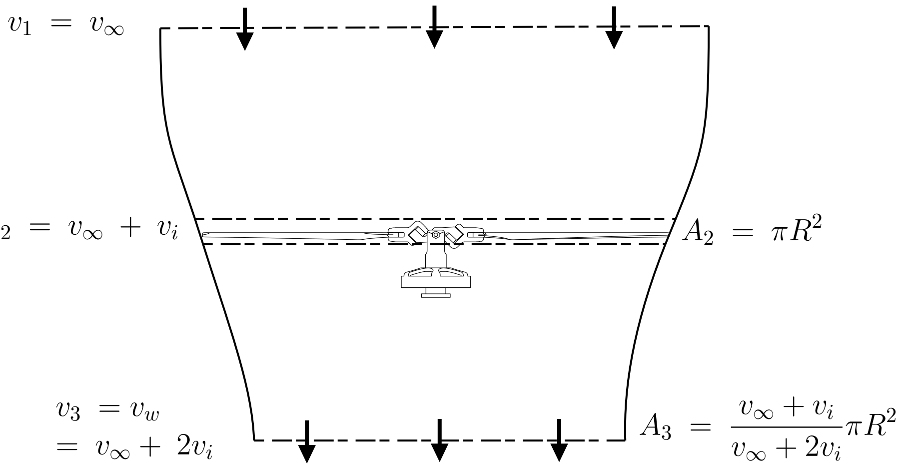


Figure 2.1: Control volume of for the slipstream produced by the rotor by Momentum Theory.

where T is the total thrust force produced by the propulsion disk, causing an increase v_i of air velocity. The power produced by the thrust force is equal to the change in kinetic energy, as other losses are neglected. This power conservation is written as

$$P = fv = Tv_i = \frac{1}{2}\dot{m}(v_w - v_\infty)^2 \tag{2.5}$$

Inserting Equation (2.4) into Equation (2.5), the far wake velocity is obtained

$$v_w = v_\infty + 2v_i \tag{2.6}$$

Using Equation (2.2), the thrust force can now be expressed as

$$T = 2\pi\rho v_i(v_\infty + v_i)R^2 \tag{2.7}$$

The area of the slipstream along the tubular control volume will change as the velocity changes due to the balance laws shown above. The area of the outlet A_3 can be expressed by evaluating Equation (2.2), Equation (2.3) and Equation (2.6).

$$A_3 = \frac{v_\infty + v_i}{v_\infty + 2v_i}\pi R^2 \tag{2.8}$$

Annular sections at radius r and width dr of the propulsion disk can be constructed. Thrust force for each annular ring, dT, can be calculated, and this will become useful in combing MT and BET. Using the same procedure and conservation laws as above, it can be shown that

$$dT = 4\pi\rho v_i(v_\infty + v_i)rdr. \tag{2.9}$$

Blade Element Theory

The blade element theory method consists of analysis of a single revolving propeller blade. Blade elements are constructed by incrementing sections along the length of the blade, which can be seen in Figure 2.3. Further analysis of a single blade element section shown in Figure 2.2 can be performed.

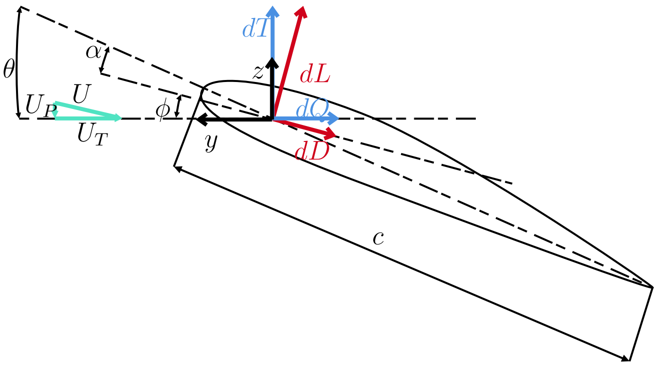


Figure 2.2: Section view of a blade element section.

The airflow acting on an airfoil section will be assumed to be locally two dimensional, such that analysis can be done locally for infinitesimal airfoil sections.

In each infinitesimal, two dimensional blade element section at radial distance r, the aerodynamic force is split into lift force dL, drag force dD and moment dM. These are reduced to the point defined $0.25c$ from the leading edge along the chord line, where the forces and moments are more suited to be modeled with linear constants. Lift forces are defined as the aerodynamic force perpendicular to the air inflow given by the local inflow angle ϕ, while the drag forces are defined parallel to the air inflow direction, as seen in Figure 2.2. The chord length c of the blade is defined as the distance between the leading and trailing edge of the blade. The line through these two points defines the chord line.

The angle of attack a is defined as the angle between the air inflow and the zero-lift line. The zero-lift line is aligned with the direction of inflow that causes zero lift. For the symmetrical airfoil presented in Figure 2.2, the zero-lift line is equal to the chordline, while for non-symmetrical airfoils this is generally not true.

Using a simple drag model, the section lift, drag and moment for a blade section can be expressed as

$$\begin{aligned} dL &= \frac{1}{2}\rho U^2 c_l c dr, \\ dD &= \frac{1}{2}\rho U^2 c_d c dr, \\ dM &= \frac{1}{2}\rho U^2 c_m c^2 dr, \end{aligned} \tag{2.10}$$

where U is the inflow velocity, c_l is the section lift coefficient, c_d is the section drag coefficient and c_m is the section moment coefficient. c_l, c_d and c_m are dependent on the local airfoil shape, the Reynolds number, the Mach number and the local angle of attack.

The inflow velocity U is defined in terms of components of tangential and perpendicular inflow velocities U_T and U_P.

The tangential component will be dominated by the blade angular speed at the local section radius. The perpendicular component is often approximated by the external inflow velocity v_∞ and the inflow velocity induced by the propeller v_i.

$$U_P \approx v_i + v_\infty \tag{2.11}$$

The local inflow angle is defined as ϕ in Figure 2.2, defined by U_P and U_T and thus $\phi = tan^{-1}(U_P/U_T)$.

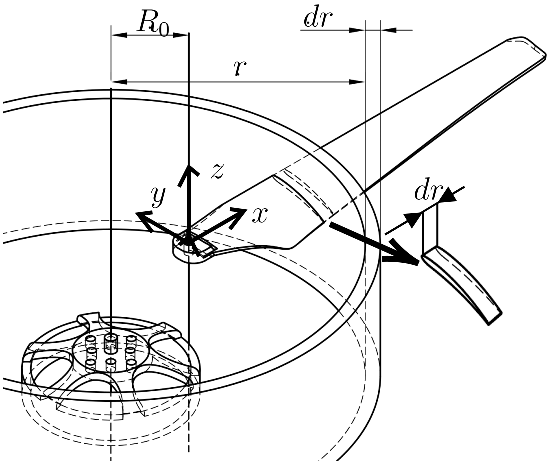


Figure 2.3: Blade element sections constructed along rotor blade. Blade sections with thickness dr are constructed at radius r from the starting radius R_0.

The local angle of attack a is then found as $a = \theta - \phi = \theta - tan^{-1}(U_P/U_T)$, where θ is the blade pitch angle. We want to express the sectional aerodynamic forces in the z- and y-axis of the blade frame.

$$\begin{aligned} dF_z &= dL\cos\phi - dD\sin\phi \\ dF_y &= -(dL\sin\phi + dD\cos\phi) \end{aligned} \tag{2.12}$$

Aerodynamic forces are conventionally expressed in terms of incremental thrust and torque, dT and dQ, aligned with the y- and z-axis.

$$\begin{aligned} dT &= dF_z \\ dQ &= dF_y r \end{aligned} \tag{2.13}$$

Aerodynamic forces are conventionally expressed in terms of incremental thrust and torque, dT and dQ, aligned with the y- and z axes. Similarly to MT, the results of BET can be summarized

$$\begin{aligned} dT =& dF_z = dL\cos\phi - dD\sin\phi \\ dQ =& -dF_y r = (dL\sin\phi + dD\cos\phi)r \end{aligned} \tag{2.14}$$

Nondimensionalization

In rotorcraft modeling, nondimensionalization during modeling and analysis is common [18]. This allows constants to be removed from the model, while the important dynamics remain. In rotorcraft analysis, this enables some fundamental relationships to be shown that do not depend on the scale of the rotorcraft.

The radial distance and increment of a sectional element in BET or annular ring in MT can be nondimensionalized by the tip speed

$$\begin{aligned} \bar{r} =& \frac{r}{R} \\ d\bar{r} =& \frac{dr}{R}. \end{aligned} \tag{2.15}$$

Nondimensional thrust and torque coefficients C_T and C_Q[18] are defined as

$$\begin{aligned} C_T =& \frac{T}{\rho(\pi R^2)(\Omega R)^2} \\ C_Q =& \frac{Q}{\rho(\pi R^2)(\Omega R)^2 R}. \end{aligned} \tag{2.16}$$

Inflow velocities v_i and v_∞ can be normalized by the propeller tip speed ΩR

$$\begin{aligned} \lambda_i =& \frac{v_i}{\Omega R} \\ \lambda_\infty =& \frac{v_\infty}{\Omega R} \end{aligned} \tag{2.17}$$

Rotor solidity σ gives a rough dimensionless approximation of the blade area compared to propulsion disk area.

$$\sigma = \frac{c}{\pi R} \tag{2.18}$$

In terms of nondimensional thrust coefficient, the result of MT becomes

$$dC_T = 4(\lambda_i + \lambda_\infty)^2 \bar{r} d\bar{r}. \tag{2.19}$$

For BET, the nondimensional result is given as

$$\begin{aligned} dC_T =& \frac{dL\cos\phi - dD\sin\phi}{\rho(\pi R^2)(\Omega R)^2} \\ dC_Q =& \frac{(dL\sin\phi + dD\cos\phi)\bar{r}}{\rho(\pi R^2)(\Omega R)^2 R} \end{aligned} \tag{2.20}$$

BEMT: Combining BET and MT

MT can model the induced inflow given the thrust and vice versa. BET can model sectional aerodynamic forces, but depends on the induced inflow velocity. To solve these equations, BET and MT must be combined. As seen above, this will in general lead to a system of non-linear equations. There are many different methods developed to solve this system either analytically or numerically, depending on the amount of assumptions and simplifications afforded.

2.1.4 BEMT with Small Angles Approximations

A very common method of solving BEMT is to arrive at an analytical solution by incorporating several small angle approximations. This method is presented in [18]. The following assumptions are made:
The inflow angle ϕ is assumed small, such that

$$\phi \approx \frac{U_P}{U_T}, \qquad \sin\phi \approx \phi, \qquad \cos\phi \approx 1.$$

With this assumption, the BET result becomes

$$\begin{aligned} dC_T =& \frac{dL - dD\phi}{\rho(\pi R^2)(\Omega R)^2} \\ dC_Q =& \frac{(dL\phi + dD)\bar{r}}{\rho(\pi R^2)(\Omega R)^2 R}. \end{aligned} \tag{2.21}$$

For low angles of attack, the drag coefficient is significantly smaller than the lift coefficient, justifying

$$dC_T = \frac{dL}{\rho(\pi R^2)(\Omega R)^2} \tag{2.22}$$

Assuming that the tangential inflow U_T is much larger than the perpendicular

$$U \approx \bar{r}\Omega R, \tag{2.23}$$

nondimensional thrust and torque coefficients can be greatly simplified.

$$\begin{aligned} dC_T =& \frac{\sigma}{2} c_l \bar{r}^2 d\bar{r} \\ dC_Q =& \frac{\sigma}{2}(\phi c_l + c_d)\bar{r}^3 d\bar{r}. \end{aligned}$$

For the remaining lift coefficient, a linear approximation is used in terms of angle of attack

$$c_l = c_{l1}\alpha + c_{l0} \tag{2.24}$$

From a specific blade section profile, the section lift curve $c_l(\alpha)$ can be generated from experimental data or CFD simulations. c_{l1} is often referred to as the section lift curve slope, while c_{l0} is the section lift curve offset [21].

$$c_{l1} = \frac{dc_l(\alpha)}{d\alpha} \qquad\qquad c_{l0} = c_l(0)$$

With these assumptions, the result from BET in Equation (2.20) can be simplified after inserting Equation (2.10).

$$dC_T = \frac{\sigma}{2}\theta' \bar{r}^2 d\bar{r}, \tag{2.25}$$

where $\theta' = \theta + \frac{c_{l0}}{c_{l1}}$ is the virtual pitch. Setting the result of MT in Equation (2.19) equal to BET in Equation (2.25):

$$4(\lambda_i + \lambda_\infty)^2 \bar{r} d\bar{r} = \frac{\sigma}{2}\theta' \bar{r}^2 d\bar{r}, \tag{2.26}$$

the nondimensional induced inflow can be calculated by a single equation

$$\lambda_i(r, \lambda_\infty) = \sqrt{\left(\frac{\sigma c_{l1}}{16} - \frac{\lambda_\infty}{2}\right)^2 + \frac{\sigma c_{l1}}{8}\theta' \bar{r}} - \frac{\sigma c_{l1}}{16} - \frac{\lambda_\infty}{2} \tag{2.27}$$

Knowing λ_i all other expressions can be evaluated. This method thus enables a very simple and computationally effective way to evaluate BEMT. The cost is the approximations introduced.

2.1.5 BEMT without Additional Assumptions

Solving numerically for inflow in BEMT is possible without including additional small angle approximations. Stahlhut and Leishman [22] derived a large angle approach to BEMT, where no small angle approximations are made. To solve the resulting set of nonlinear equations, they are combined into a single transcendental equation of inflow angle

ϕ. Without showing the full derivation, the results can be summarized to show the form. [22] shows that the transcendental equation of inflow can be expressed as

$$g(\phi) = (\Omega r \sin\phi - V_\infty \cos\phi)\sin\phi - \mathrm{sgn}(\phi)\frac{\sigma C_l \sec\gamma}{8r}\left[\frac{\Omega y}{K_T}\cos(\phi+\gamma) + \frac{V_\infty}{K_P}\sin(\phi+\gamma)\right], \tag{2.28}$$

where $\gamma = \tan^{-1}(\frac{c_d}{c_l})$, K_T and K_P capture tangential and perpendicular loss effects.

Root finding methods can be used to solve for ϕ. Stahlhut and Leishman [22] discusses common methods which include fixed-point iteration and gradient based methods, where there are problems with convergence. A bracketed bisection method is proposed, which has guaranteed convergence, but is quite slow. [22] is referred to for further details.

2.1.6 Disk Loading and Scaling

A possibility with a rotor pitch controlled MAV is to use fewer, larger rotor blades compared to a configuration such as the quadcopter. By employing momentum theory, it is possible to establish a relationship between the ideal power required for hover P, the thrust T and disk area $A = \pi R^2$. At hover, v_∞ is zero. By then inserting Equation (2.2) into Equation (2.4)

$$T = 2\rho A v_i^2 \implies v_i = \sqrt{\frac{T}{2\rho A}} \tag{2.29}$$

Inserting this into Equation (2.5)

$$P = T v_i = T\sqrt{\frac{T}{2\rho A}} \tag{2.30}$$

$$\frac{P}{T} = \sqrt{\frac{T}{2\rho A}}, \tag{2.31}$$

which gives the power per thrust ratio. Disk loading is then defined as T/A. The insight from this relation is that the lower the disk loading, the more efficient hover can be obtained, requiring less power per thrust. Since the disk loading scales inversely with disk area, a larger disk area compared to the mass of the vehicle will in general give higher hovering efficiency. This relationship can be used for insight about the scale of the vehicle, but it can also be used to compare the efficiency of different rotor configurations. Qin, Xu, Lee, *et al.* [23] compares disk loading and hovering efficiency for rotorcraft with the same minimum width, accounting only for the rotor disks, using Equation (2.31). A quadcopter or helicopter configuration of the same minimum width will have the same disk area when the quadcopter rotor spacing is at the minimum. For a bicopter or coaxial rotorcraft with the same minimum width, the disk area will be twice that of the single rotor helicopter. According to momentum theory in Equation (2.31), the hovering efficiency with twice the disk area increases by $\sqrt{2} - 1 = 41\%$, assuming the same rotorcraft weight. Momentum theory captures the ideal, maximum efficiency given a disk area. Aerodynamic losses will decrease the efficiency, and depend on elements such as the geometry and size of the rotor blades, operating speed of the rotors as well as inflow altering effects such as nearby rotors or the rotorcraft frame.

2.2 Rigid Multibody Systems

A model of both the rotor and the rotorcraft will be developed. For the rotor, a multibody system captures the articulated rotor with hubs and blades connected by hinges. The rotorcraft itself will also be modeled as a single rigid body system. This section aims to cover the basic principles that will be used for further modeling, design and control. The modeling of both the rotor and the rotorcraft will remain quite simple, such that this section aims to introduce the principles used, rather than provide extensive coverage of multibody systems. It is assumed that the reader has knowledge of this theory.

For kinematics, the theory is based on Waldron and Schmiedeler [24]. For dynamics, the theory is based on Featherstone and Orin [25].

2.2.1 Kinematics

Rotation Matrices

Rotation matrices are orthogonal matrices used to perform rotations in Euclidean space. Direction cosines between the unit vectors of the two frames form the elements of these matrices. The rotation of the coordinate frame i to the coordinate frame j can be expressed as

$$\mathcal{R}_i^j = \begin{bmatrix} \hat{\boldsymbol{x}}_i \cdot \hat{\boldsymbol{x}}_j & \hat{\boldsymbol{y}}_i \cdot \hat{\boldsymbol{x}}_j & \hat{\boldsymbol{z}}_i \cdot \hat{\boldsymbol{x}}_j \\ \hat{\boldsymbol{x}}_i \cdot \hat{\boldsymbol{y}}_j & \hat{\boldsymbol{y}}_i \cdot \hat{\boldsymbol{y}}_j & \hat{\boldsymbol{z}}_i \cdot \hat{\boldsymbol{y}}_j \\ \hat{\boldsymbol{x}}_i \cdot \hat{\boldsymbol{z}}_j & \hat{\boldsymbol{y}}_i \cdot \hat{\boldsymbol{z}}_j & \hat{\boldsymbol{z}}_i \cdot \hat{\boldsymbol{z}}_j \end{bmatrix} \tag{2.32}$$

The following matrices can be shown to represent an elementary rotation θ around the x, y, and z coordinate axes of a reference frame:

$$\begin{aligned} \mathcal{R}_Z(\theta) &= \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ \mathcal{R}_Y(\theta) &= \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \\ \mathcal{R}_X(\theta) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \end{aligned} \tag{2.33}$$

Euler Angles

Euler angles are used to represent frame orientation by three consecutive rotations about a set of coordinate axes. This allows a rotation to be expressed in terms of three variables instead of nine as in a rotation matrix. There exist different conventions to choose the axes, their order as well which coordinate frame is used. The two main conventions are proper Euler angles and the Tait-Bryan convention. In the proper Euler angle convention, one axis of rotation is repeated. In the Tait-Bryan convention, all three coordinate frame axes are used. A common order of rotations with Tait-Bryan is the Z-Y-X convention, with roll-pitch-yaw terminology originating from aeronautics. When performing the rotation sequence, the body frame is initially aligned with the inertial reference frame used. Subsequent rotations occur about the body coordinate axis that follows the rotation steps. The Z-Y-X sequence consists of a yaw rotation ψ about the body z-axis, a pitch rotation θ about the now rotated body y-axis, and then a roll rotation ϕ about the body x-axis. This Z-Y-X rotation sequence of a body frame $\mathcal{B}$ attached to a rigid body in relation to a reference frame $\mathcal{I}$ can be expressed as a rotation matrix:

$$\mathcal{R}_{\mathcal{B}}^{\mathcal{I}} = \mathcal{R}_Z(\psi)\mathcal{R}_Y(\theta)\mathcal{R}_X(\phi) \tag{2.34}$$

In rotorcraft control, the Tait-Bryan convention is common, and will be used in this thesis when referring to Euler angles.

Joint Kinematics Representation

A kinematic chain can be formed by a series of rigid bodies, sometimes called links, connected by perfect joints to form a chain, tree or loop structure. This is an idealization of the real system, where the bodies are not rigid, and the joints will have non ideal properties. In the rotor modeling case, each hinge represents a revolute joint, and each blade or hub represents a rigid body, forming a kinematic chain.

Defining the geometrical representation of the kinematic structure can be done with different conventions. In general, a reference frame for each rigid body needs to be defined, defining the pose of the body in relation to the other bodies and joints. Common conventions such as the Denavit-Hartenberg convention reduce the number of parameters necessary to define each reference frame in the chain from the full six parameters to four parameters.

A consequence with these four parameter representations is that the coordinate frames of each body will have a unique orientation for a given structure, the body z-axis has to align with the joint axis. An alternative is to express each coordinate frame with the full six parameters of rotation and translation. The advantage in this case is that the body coordinate frame can be set according to what is convenient. Such representations are common in modular and implementation oriented systems such as the Unified Robot Description Format (URDF).

Differential Kinematics

Differential kinematics describes the relationship between joint velocities and the velocities of end-effectors in the kinematic chain of a mechanical system. This will depend on the current position of all the bodies in the chain. The relationship is commonly expressed using a geometric Jacobian matrix, $\boldsymbol{J}$, which is in general a non-linear function of the joint variables q. The Jacobian consists of the first-order partial derivatives of the kinematic model relating the position of the associated end-effector in terms of joint variables.

The geometric Jacobian can be used to solve the instantaneous forward kinematics problem, computing the velocity of the associated end-effector when the joint velocities are known.

$$\mathrm{v} = \boldsymbol{J}(\mathrm{q})\dot{\mathrm{q}} \tag{2.35}$$

where $\mathbf{v}$ is the velocity of the end-effector and $\dot{\mathbf{q}}$ is the vector of joint velocities. In a system with multiple end-effectors, each end-effector will have its own Jacobian matrix, reflecting its specific relationship between joint velocities and its own velocity. Typically in robotics, there is a single end-effector which results in a single geometric Jacobian. In the rotor modeling effort in this thesis, there is a kinematic tree structure, where there are two branches, with two end-effectors of concern. These are the two blades.

Statics

Statics analysis concerns forces and torques acting on rigid bodies. A wrench $\boldsymbol{w} \in \mathbb{R}^6$ represents the generalized force and moment acting on a rigid body, given in some coordinate frame. The wrench $\boldsymbol{w}_b^{\mathcal{B}}$ acting on body b given in frame $\mathcal{B}$ can be assembled as

$$\boldsymbol{w}_b^{B} = \begin{bmatrix} \boldsymbol{\tau}_b^{B} \\ \boldsymbol{f}_b^{B}, \end{bmatrix} \tag{2.36}$$

where $\boldsymbol{\tau}_b^{\mathcal{B}}$ is the moment acting on b, and $\boldsymbol{f}_b^{\mathcal{B}}$ is the force acting on b, both given in frame $\mathcal{B}$. The wrench applied to a body in a kinematic chain can be related to the joint space forces and torques $\boldsymbol{\tau}$ by the geometric Jacobian.

$$\boldsymbol{\tau} = \boldsymbol{J}^T \boldsymbol{w} \tag{2.37}$$

Both the geometric Jacobian and the wrench needs to be expressed in the same coordinate frame here.

When there are multiple wrenches on multiple end-effectors in the kinematic chain, the resulting total joint space forces and torques can be summed. In the case such as the rotor where there are two end effectors, the total generalized forces and moments can be expressed as

$$\boldsymbol{\tau} = \boldsymbol{J}_1^T \boldsymbol{w}_1 + \boldsymbol{J}_2^T \boldsymbol{w}_2 \tag{2.38}$$

where $\boldsymbol{J}_1$ and $\boldsymbol{J}_2$ are the respective geometric Jacobians for the two end-effectors where the wrench $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ are applied in their respective frames.

2.2.2 Dynamics

The dynamics of rigid body systems consists of relating forces and moments acting on the system to the resulting accelerations and motions of the system.

Equations of Motion

The canonical equations of motion in joint space is given as:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} \tag{2.39}$$

The terms in this equation can be explained in terms of their physical meaning:

- $\boldsymbol{M}(\boldsymbol{q})$ is the non-linear inertia matrix which enters as a product with joint accelerations.
- $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is the velocity product that captures Coriolis and centrifugal terms.
- $\boldsymbol{g}(\boldsymbol{q})$ contains the terms due to the gravity force acting on each body.
- $\boldsymbol{\tau}$ contains external forces or moments.

Computing Forward Dynamics of Rigid Multibody Systems

In this thesis, the forward dynamics is computed when simulating the rotor system. Forward dynamics concerns the calculation of joint accelerations from applied joint forces and torques. From the equations of motion in Equation (2.39), the joint accelerations can be calculated as

$$\ddot{\boldsymbol{q}} = \boldsymbol{M}(\mathrm{q})^{-1}\left(-\boldsymbol{C}(\mathrm{q},\dot{\mathrm{q}})\dot{\mathrm{q}} - \boldsymbol{g}(\mathrm{q}) + \boldsymbol{\tau}\right) \tag{2.40}$$

There exist efficient recursive algorithms that exploit the kinematic tree structure explained in Section 2.2.1 to compute the forward dynamics. One such algorithm is the articulated-body algorithm, covered in detail by Featherstone and Orin [25].

2.3 Field Oriented Motor Control

This section is taken from the previous work of the author in the related specialization project. The motor modeling and control techniques described in this section are necessary for further design and analysis of the thrust vectoring rotor.

2.3.1 AC Synchronous Motors

For MAV propulsion, AC synchronous motors with permanent rare-earth magnets are commonly used, while brushed DC motors have been used primarily for extremely low cost vehicles. Compared to brushed DC motors, AC synchronous motors eliminate the mechanical brush commutator, achieving higher power density, better dynamical performance and efficiency [26].

Two different designs of AC synchronous motors with permanent magnets will be analyzed, permanent magnet synchronous motors (PMSM), and brush-less DC motors (BLDCM). The main difference between PMSM and BLDCM is the shape of the back-electromotive force (EMF) waveform. PMSM has windings designed for a sinusoidal back-EMF, and BLDCM has windings designed for a trapezoidal back-EMF waveform [26]. Off-the-shelf motors typically used for propeller driven MAVs are commonly BLDCM. The reason for the back-EMF waveform of BLDCM is that it is suitable for a simple, more economical six-step control method [26]. For PMSM, FOC is a common control technique, but BLDCM can still benefit from FOC. As the electrical and mechanical dynamical models are otherwise identical between these motor types, PMSM will be considered for modeling, before discussing differences between PMSM and BLDCM.

2.3.2 PMSM Model

A PMSM with three stator windings, corresponding to phases a, b, c will be considered. The stator phases have corresponding currents $\boldsymbol{i}_{abc} = \begin{bmatrix} i_a & i_b & i_c \end{bmatrix}^T$, voltages $\boldsymbol{u}_{abc} = \begin{bmatrix} u_a & u_b & u_c \end{bmatrix}^T$ and fluxes $\boldsymbol{\varphi}_{abc} = \begin{bmatrix} \varphi_a & \varphi_b & \varphi_c \end{bmatrix}^T$. An illustration of a single pole pair PMSM with external rotor is given in Figure 2.4. The permanent magnets lie in the rotor, while the phase windings lie in the stator. Interior rotor PMSMs are most common in industry, but MAVs usually have exterior rotors. It does not present a difference for a simplified model. The rotor position ψ is given in relation to the a-phase axis, and the

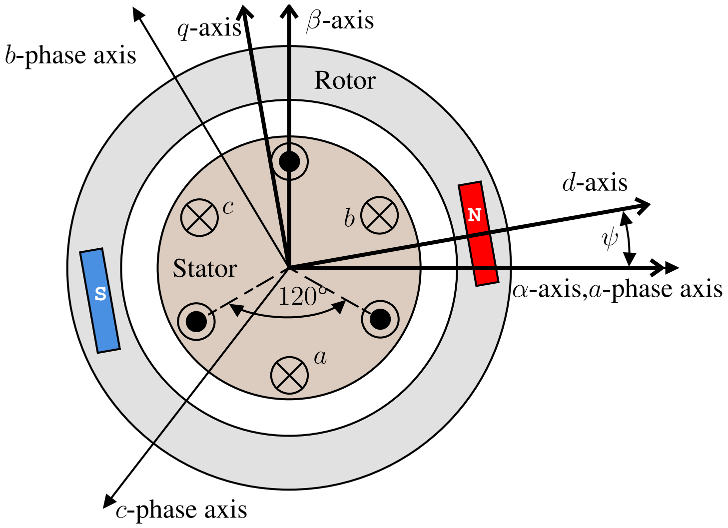


Figure 2.4: Exterior rotor PMSM with single pole pair.

mechanical rotor speed $\omega = \dot{\psi}$ is assumed to be positive to simplify modeling. The phase quantities are assumed to be balanced due to symmetry [26], such that

$$\begin{aligned} i_a + i_b + i_c &= 0 \\ u_a + u_b + u_c &= 0 \\ \varphi_a + \varphi_b + \varphi_c &= 0 \end{aligned} \tag{2.41}$$

The rotor electrical angle is given as $\psi_e = N_p \psi$, where N_p is the number of pole pairs. The number of poles of the PMSM will typically be equal to the number of permanent magnets in the rotor. Figure 2.4 shows a PMSM with a single pole pair, such that the rotor mechanical angle ψ is aligned with the electrical angle ψ_e.

A general AC synchronous motor stator voltage laws are given by [26]

$$\frac{d\boldsymbol{\varphi}_{abc}}{dt} = \boldsymbol{u}_{abc} - R_{ohm}\boldsymbol{i}_{abc} \tag{2.42}$$

where R_{ohm} is the stator winding resistance, such that the voltage is dissipated as resistive winding losses and contributions to the stator fluxes. The rotor flux is given by the permanent magnets in the rotor. The rotor flux components $\boldsymbol{\varphi}_r$, written in the stationary stator frame, will rotate with the rotor electrical angle ψ_e. $\boldsymbol{\varphi}_r$ can be written as [27]

$$\boldsymbol{\varphi}_r = \varphi_e \begin{bmatrix} \cos(\psi_e) \\ \cos(\psi_e - 2\pi/3) \\ \cos(\psi_e + 2\pi/3) \end{bmatrix} \tag{2.43}$$

where φ_e is the nominal rotor flux amplitude. The stator flux $\boldsymbol{\varphi}_{abc}$ consists of two components, a stator current component and a rotor flux component.

$$\boldsymbol{\varphi}_{abc} = \boldsymbol{\varphi}_r + L\boldsymbol{i}_{abc} \tag{2.44}$$

By time differentiating Equation (2.44)

$$\frac{d\boldsymbol{\varphi}_{abc}}{dt} = -\varphi_e \begin{bmatrix} \sin(\psi_e) \\ \sin(\psi_e - 2\pi/3) \\ \sin(\psi_e + 2\pi/3) \end{bmatrix} \dot{\psi}_e + L\frac{d\boldsymbol{i}_{abc}}{dt} \tag{2.45}$$

the current dynamics are obtained by inserting Equation (2.42) in Equation (2.45).

$$\begin{aligned} \frac{d\boldsymbol{i}_{abc}}{dt} &= \frac{\boldsymbol{u}_{abc}}{L} - \frac{R_{ohm}\boldsymbol{i}_{abc}}{L} + \frac{\varphi_e}{L} \begin{bmatrix} \sin(\psi_e) \\ \sin(\psi_e - 2\pi/3) \\ \sin(\psi_e + 2\pi/3) \end{bmatrix} \dot{\psi}_e \\ \dot{\psi}_e &= \omega_e \end{aligned} \tag{2.46}$$

2.3.3 Coordinate Transformations

The stator winding currents, voltages and fluxes will be represented in different frames, which will be convenient for control.

Clarke Transformation

The abc quantities can be transformed to a stator fixed, orthogonal two-coordinate frame $\alpha\beta$ by the Clarke transform [28]. The Clarke transform is given by

$$\boldsymbol{T}_C = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \tag{2.47}$$

This transformation preserves amplitudes as its determinant is 1. Due to the balanced quantities from Equation (2.41), we get the two-coordinate $\alpha\beta$ frame by noticing that the last coordinate after applying the Clarke transform always equals zero. By using this simplification, the i_α and i_β currents are expressed as

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} \tag{2.48}$$

As seen in Figure 2.4, the α-axis is stator fixed and aligned to the a-phase axis.

Park Transformation

A rotor fixed, orthogonal two coordinate direct-quadrature frame dq is defined, with the d-axis aligned with the rotor electrical angle ψ_e [28]. The transformation from the $\alpha\beta$-frame to the dq reference frame is then simply a 2D rotation by ψ_e. The transformation can be expressed with a 2D rotation matrix.

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \begin{bmatrix} \cos(\psi_e) & \sin(\psi_e) \\ -\sin(\psi_e) & \cos(\psi_e) \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} \tag{2.49}$$

This rotation will be referenced as the Park transformation in the following sections, however, the total transformation from abc quantities to dq frame is sometimes referenced as the Park transform [28].

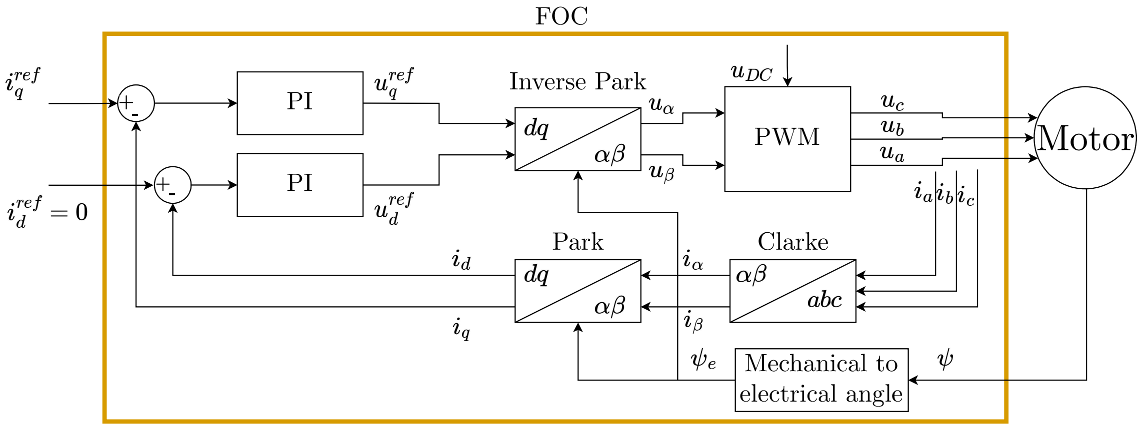


Figure 2.5: Block diagram of Field oriented control algorithm

2.3.4 Field Oriented Control

The main idea of FOC is to do feedback current control in the rotor fixed frame dq. The current dynamics in Equation (2.46) can be expressed in the dq-frame by applying the Clarke and Park transform. After applying some trigonometric identities, the result of this transformation can be written as

$$\begin{aligned} \frac{di_d}{dt} &= \frac{u_d}{L} - \frac{R_{ohm} i_d}{L} + \dot{\psi}_e i_q \\ \frac{di_q}{dt} &= \frac{u_q}{L} - \frac{R_{ohm} i_q}{L} + \frac{\varphi_e}{L} + \dot{\psi}_e i_d \end{aligned} \tag{2.50}$$

In this frame, currents and voltages vary slowly and can be controlled at a much lower bandwidth compared to the stator frame due to the independence of rotor electrical position. Another convenient feature of the dq frame is that d and q currents will independently contribute to the flux and torque, respectively. This is similar to the dynamics of a DC motor. The electrical torque in a PMSM in the dq-frame is given as ([27])

$$\tau_m = \frac{3}{2} i_q N_p \varphi_e \tag{2.51}$$

This relation can be expressed in terms of a torque constant $k_T = N_p \varphi_e$ with SI unit $[\frac{Nm}{A}]$, or in terms of the back-EMF constant $k_e = N_p \varphi_e$ with unit $[\frac{V}{rad/s}]$. k_e is also called the motor electrical constant.

$$\tau_m = \frac{3}{2} i_q k_e \tag{2.52}$$

This result shows that the torque dynamics in terms of the quadrature current i_q is similar to that of a DC-motor.

FOC algorithm

The remaining problem is to find and apply the phase voltages $\boldsymbol{u}_{abc}$ that will result in the dq reference current. Figure 2.5 shows an overview of the FOC algorithm. The current controller of the motor operates in rotor frame currents i_q and i_d. A control objective of FOC is to maximize the torque per ampere. This can be achieved by controlling the direct current i_d to zero, and using the quadrature current i_q as the reference. A quadrature current reference is given as input to the current controller, which outputs voltage references in the dq rotor fixed frame. To perform feedback current control, the current needs to be measured or estimated. By directly measuring phase currents i_a, i_b, i_c, as well as measuring or estimating ψ_e, Clark and Park transformations can be applied to reconstruct measured dq currents for the current loop. The output from the current controller, quadrature and direct voltage references u_q^{ref} and u_d^{ref} are transformed to the $\alpha\beta$-frame. Pulse width modulation (PWM) of the DC supply voltage u_{DC} is used to generate the phase voltages to be applied to the motor phases.

Torque control is then obtained by the relation

$$i_q^{ref} = \frac{2}{3N_p\varphi_e}\tau_m^{ref} \tag{2.53}$$

such that a reference torque τ_m^{ref} can be tracked by calculating the equivalent quadrature current reference i_q^{ref} for the current loop, while keeping the direct current reference i_d^{ref} zero.

2.3.5 Six-step BLDCM Control

The six-step algorithm, also called trapezoidal control, exploits the specifically designed back-EMF characteristic of BLDCM and is widely used for MAV motor control. Some variation of this algorithm is used by the motor controller in [6] and [7]. For the six-step algorithm, one motor phase is given a PWM-driven voltage, one phase is driven to 0, and one phase is kept floating at any time moment ([27]). Current only flows between the two non-floating phases, and Equation (2.41) conveniently restricts the control problem. The motor is divided into a sextant, where the controlled phase is updated at every crossing. The crossings can be detected by measuring the back-EMF such that a crossing happens when the open phase voltage crosses zero compared to neutral. The duty-cycle of the PWM signal then gives the back-EMF oriented current vector, which is used to control the motor speed. In this technique, the mechanical angle of the motor is not needed, the electrical angle is rather estimated based on the zero-crossings. Another advantage of the six-step algorithm is that it is completely parameter free, making it seamless to apply to different motors.

2.3.6 FOC for BLDCM

Field oriented control has been shown for PMSM, however, the motor that will be used is a BLDCM. FOC can be applied in the exact same manner for BLDCM. The main difference will be torque ripple due to the mismatch between the sinusoidal drive current from FOC and the trapezoidal back-EMF of BLDCM. [26, p. 342] states that the six-step algorithm for BLDCM suffers from torque ripple due to the difficulty of manufacturing a BLDCM with a perfect trapezoidal back-EMF. [27] investigates the back-EMF of two different BLDCMs designed for MAV use. It is apparent that the back-EMF waveforms of the tested motors are closer to sinusoidal than to trapezoidal. [27] then validates experimentally that for a specific BLDCM, T-Motor Antigravity 4006, a FOC approach outperforms a six-step algorithm in efficiency by a small margin. The tests were performed by operating the motor with a 13 inch propeller at different constant speeds. Efficiency was measured by considering the total power compared to the resistive losses in the windings, computed from the measured winding resistance. In particular, the efficiency gains were measured to be 0.4%, 0.77% and 0.98% respectively for steady propeller spinning at 3000, 4500 and 6000 rpm.

3

Methods

3.1 Modeling the Thrust Vectoring Rotor

A model of the dynamics could be useful for many applications. One of the key motivations of this model is to better understand the fundamental properties of the thrust vectoring rotor. Here, being able to accurately predict the coupling between torque modulation and teetering response will be a key factor in the quality of the model. Other motivations are beyond the scope of this thesis. A good model will allow for design experimentation and optimization of rotor designs. Additionally, a sufficiently computationally efficient rotor model could be useful for model-based control.

Paulos and Yim [3] developed a model of a similar rotor, with two offset flapping hinges compared to the single central teetering hinge in this case. A similar open-chain dynamics formulation will be used here to model the teetering rotor. The model of Paulos and Yim [3] assumed symmetric motion of the blades and derived closed-form analytical equations of a linearized system. In the modeling effort in this thesis, the blade motion is not assumed to be symmetrical. A non-linear model is considered to capture non-linear relations in motions and aerodynamics.

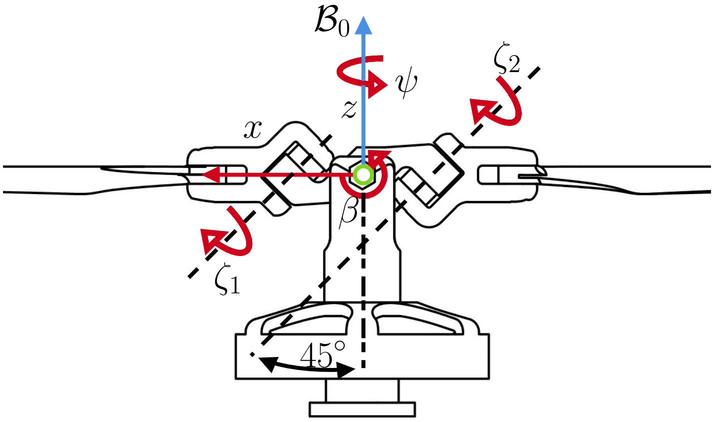


Figure 3.1: Definition of rotor joint parameters and base frame $\mathcal{B}_0$. The rotor is illustrated with joint parameters at zero, as well as the fixed lag-pitch skew angle of $45°$.

3.1.1 Rotor Rigid Body Kinematic Chain

The rotor consists of four moving bodies, labeled B_1 to B_4 in Figure 3.2. B_1 consists of the motor rotor as well as the attached central hub. B_2 consists of the teetering hub. B_3 and B_4 are the two bodies at the end of the chain, where each of them consists of the rotor side hubs with a blade rigidly attached. B_3 will be referred to as the positive blade body, as there is a positive coupling between lag and pitch. B_4 will be referred to as the negative blade body. There are four revolute joints, the motor position ψ, the teetering hinge angle β, ζ_1 and ζ_2. ζ_1 is the skewed lag-pitch hinge on the positive blade side and ζ_2 is the skewed lag-pitch hinge on the negative blade side. The joint definitions can be seen in Figure 3.1. Generalized coordinates $\mathbf{q}$ are chosen as the joint parameters.

$$\mathbf{q} = \begin{bmatrix} \psi \\ \beta \\ \zeta_1 \\ \zeta_2 \end{bmatrix} \tag{3.1}$$

With a general open-chain formulation as in Section 2.2.2, the equations of motion can be written as

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}_{\mathrm{m}} - \boldsymbol{n}_{aero} \tag{3.2}$$

The external forces and moments in the system are the aerodynamic moments which enters in the vector $\boldsymbol{n}_{aero}$, and the motor torque which enters in the vector $\boldsymbol{\tau}_m$.

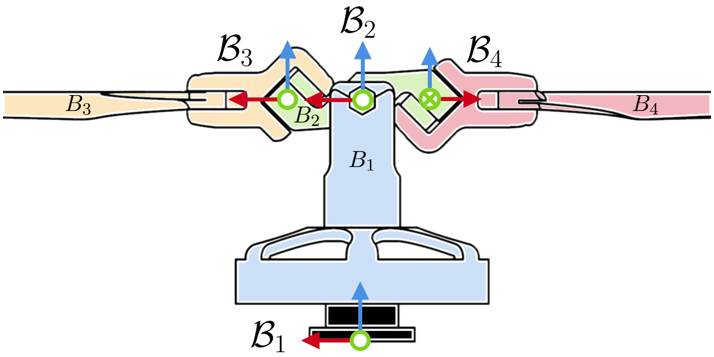


Figure 3.2: Definition of rotor bodies B_i and body frames $\mathcal{B}_i$ for each of the four rigid bodies modeled. The frame orientations are illustrated with all joint parameters at zero.

3.1.2 Aerodynamics

To apply BEMT for this rotor, some modeling choices have been made. There are two main formulations that could be made regarding the choice of reference frame for the blade. One approach is to fix the z-axis of the blade frame to the base frame $\mathcal{B}_0$ z-axis, regardless of joint angles. Then, the blade pitch angle can be calculated in terms of joint angles, and joint velocities can enter as terms in the sectional velocities. This approach was used by the modeling effort of Paulos and Yim [3], and was suited for deriving a linearized model.

In this work, a different approach is taken. The blade frame is chosen to be fixed to the blade rigid body. This frame follows joint angles, such that the blade pitch is fixed. By calculating sectional velocities along the blade in this frame, all effects of joint angles and joint velocities are captured in the inflow angle ϕ.

This choice of reference frame is further motivated by modeling the blade induced inflow from the MT propulsion disk. When a phase-locked teetering response is induced, the propulsion disk will tilt. In a steady-state operation, the tilting angle of the propeller tip path plane can be defined by the once per revolution maximum angle of teetering, and azimuth angle of maximum teetering. This tilting is assumed to define the propulsion disk. Modeling the propulsion disk dynamics will in general require a subsystem with a set of new states. To approximate this tilting in a simple way, an observation of the simulated system is used. At steady-state propulsion disk tilting, the phase-locked teetering and lag-pitch joint response causes the z-axis of the blade frames to be close to perpendicular to the propulsion disk over the whole azimuth. If the blade is assumed to move in the tilted propulsion disk equivalent to the non-hinged rotor, this alignment is necessary. With these arguments, the inflow induced by MT is assumed to be a perpendicular component in blade frame at any point in time. This is convenient as it makes this formulation equal to standard MT, just with a different reference frame.

The inflow velocity U, now expressed in blade frame, can be modified from the standard BEMT formulation to account for both tangential and perpendicular blade induced inflows.

$$\begin{aligned} U_P &= v_{i_P} + v_{\infty_P}, \\ U_T &= v_{i_T} + v_{\infty_T}, \end{aligned} \tag{3.3}$$

where v_{i_P} and v_{i_T} are the perpendicular and tangential induced inflows, and v_{∞_P} and v_{∞_T} are the perpendicular and tangential external inflows. The frames defined in Figure 3.2 are convenient in choosing blade body fixed frames, $\mathcal{B}_3$ is used for the positive blade, and $\mathcal{B}_4$ is used for the negative blade. In the blade frame defined, differential thrust $dT = dF_z$ will be a perpendicular component in both BET and MT, and the standard MT result in equation (2.9) can be used. The average rotor speed Ω is used for MT. For BET, the modified inflow velocity in Equation (3.3) is used with terms from joint velocities when calculating differential sectional forces dF_z and dF_y in Equation (2.12).

For the two blade bodies B_b and frames $\mathcal{B}_b$, $b = \{3, 4\}$, the blade sectional velocity components v_{i_T} and v_{i_P} at radial position r can be calculated as

$$\begin{aligned} v_{i_T} &= v_{i,y}^{\mathcal{B}_b}, \\ v_{i_P} &= v_{i,z}^{\mathcal{B}_b}, \end{aligned} \tag{3.4}$$

where $\mathbf{v}_i^{\mathcal{B}_b} = \begin{bmatrix} v_{i,x}^{\mathcal{B}_b} & v_{i,y}^{\mathcal{B}_b} & v_{i,z}^{\mathcal{B}_b} \end{bmatrix}^T$, given by

$$\mathbf{v}_i^{\mathcal{B}_b} = \mathcal{R}_{\mathcal{B}_0}^{\mathcal{B}_b}(\boldsymbol{J}_{B_b,v}(\mathbf{q})\dot{\mathbf{q}}) + (\mathcal{R}_{\mathcal{B}_0}^{\mathcal{B}_b}(\boldsymbol{J}_{B_b,\omega}(\mathbf{q})\dot{\mathbf{q}})) \times \mathbf{r}, \tag{3.5}$$

where $\boldsymbol{J}_{B_b} = \begin{bmatrix} \boldsymbol{J}_{B_b,v} & \boldsymbol{J}_{B_b,\omega} \end{bmatrix}^T$ and $\boldsymbol{r} = \begin{bmatrix} r & 0 & 0 \end{bmatrix}^T$. Equation (3.5) is derived in Appendix A.1.

By using Equation (3.3) in Equation (2.12), the aerodynamic forces in the blade reference frame acting on the blade body can be calculated. This formulation is illustrated for the positive blade in Figure 3.3. For convenience, forces are expressed in their dimensioned version rather than their dimensionless equivalent.

$$\begin{aligned} F_z &= \int_{R_0}^{R} dF_z dr \\ F_y &= \int_{R_0}^{R} dF_y dr \end{aligned} \tag{3.6}$$

Calculating the moments acting on the origin of the blade reference frame due to the aerodynamic forces involves multiplying by the blade radial position r, the one-dimensional force displacement in the blade frame.

$$\begin{aligned} M_z &= \int_{R_0}^{R} dF_z r dr \\ M_y &= \int_{R_0}^{R} dF_y r dr \end{aligned} \tag{3.7}$$

With these general results for a blade, the two wrenches $\boldsymbol{w}_{B_3}^{\mathcal{B}_3}$ and $\boldsymbol{w}_{B_4}^{\mathcal{B}_4}$, acting on bodies B_3 and B_4, given in blade frames $\mathcal{B}_3$ and $\mathcal{B}_4$, can be assembled.

$$\boldsymbol{w}_{B_3}^{\mathcal{B}_3} = \begin{bmatrix} 0 \\ M_{3,y} \\ M_{3,z} \\ 0 \\ F_{3,y} \\ F_{3,z} \end{bmatrix} \quad \boldsymbol{w}_{B_4}^{\mathcal{B}_4} = \begin{bmatrix} 0 \\ M_{4,y} \\ M_{4,z} \\ 0 \\ F_{4,y} \\ F_{4,z} \end{bmatrix} \tag{3.8}$$

The final aerodynamic joint space torque $\boldsymbol{n}_{aero}$ enters the equations of motion in Equation (3.2). By statics, this is computed by the wrenches and their corresponding geometric Jacobians.

$$\boldsymbol{n}_{aero} = \boldsymbol{J}_{B_3}^T(\mathrm{q})\boldsymbol{w}_{B_3}^{\mathcal{B}_3} + \boldsymbol{J}_{B_4}^T(\mathrm{q})\boldsymbol{w}_{B_4}^{\mathcal{B}_4} \tag{3.9}$$

3.1.3 Motor Model and Harmonic Control

It is assumed for motor modeling purposes that the electrical dynamics are significantly faster than the mechanical dynamics. As explained in Section 2.3.6, this motivates the separation of mechanical and electrical control loops. With this assumption, the motor is assumed to be able to achieve an instantaneous phase current, neglecting the dynamics of the current control loop. With this assumption, an instantaneous torque is achieved.

The quadrature current i_q^{ref} input to the idealized FOC subsystem is calculated from the motor control law. In anticipation of harmonic torque modulation, two reference signals are adopted. The motor controller consists of a PI speed controller with motor speed reference ω_{ref}, and an optional torque feedforward τ_m^{ff}. Two different control laws will be considered, one which uses harmonic speed reference and one which uses harmonic torque feedforward. In the case where harmonic torque is modulated through a harmonic speed reference, torque feedforward is not used. The harmonic speed reference is in that case expressed as

$$\omega_{ref} = \Omega + \tilde{\Omega}, \tag{3.10}$$

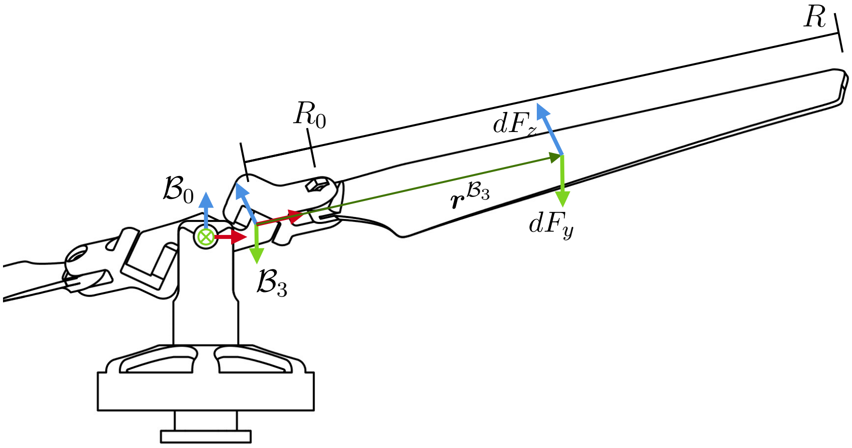


Figure 3.3: Blade section differential forces dF_z and dF_y for the positive blade at radial position r. $\mathcal{B}_0$ is the base frame, $\mathcal{B}_3$ is the body frame attached to body B_3, the positive blade. R_0 to R illustrates the part of the rotor where aerodynamic forces are considered.

where Ω is the average rotor speed setpoint, and $\tilde{\Omega}$ is the harmonic speed component. For the torque feedforward modulation control law, the speed reference is given as

$$\omega_{ref} = \Omega. \tag{3.11}$$

These control laws are further explained in Section 3.2.3. With these control laws, the FOC input dq currents can be expressed as

$$\begin{aligned} i_q^{ref} &= K_{P_v}(\omega_{ref} - \omega) - K_{I_v}\int(\omega_{ref} - \omega)\mathrm{d}t + \frac{2}{3K_e}\tau_m^{ff}, \\ i_d^{ref} &= 0, \end{aligned} \tag{3.12}$$

where K_{P_v} and K_{I_v} are constants for the speed PI-controller. The torque feedforward τ_m^{ff} is expressed in i_q by Equation (2.53). From the relation between motor torque and quadrature current in Equation (2.52), the resulting motor torque can be expressed.

$$\begin{aligned} \tau_m =& \frac{3}{2}K_e\left[K_{P_v}(e(t)) + K_{I_v}\int_0^t(e(\tau))\mathrm{d}\tau\right] + \tau_m^{ff} \\ e(t) =& \omega^{ref} - \omega \end{aligned} \tag{3.13}$$

The motor torque τ_m enters the equations of motion in vector $\boldsymbol{\tau}_m$:

$$\boldsymbol{\tau}_m = \begin{bmatrix} \tau_m \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.14}$$

3.1.4 State-Space Formulation

The equations of motion in Equation (3.2) can be described as a first order differential equation of state space $\bar{\boldsymbol{x}}$.

$$\bar{\mathbf{x}} = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} \tag{3.15}$$

An extra error state is necessary to handle the integral term from Equation (3.13).

$$\begin{aligned} x_e &= \int_0^t e(\tau) d\tau \\ \dot{x_e} &= e \end{aligned} \tag{3.16}$$

The augmented state-space model is defined with states $\mathbf{x}$:

$$\mathbf{x} = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \\ x_e \end{bmatrix} = \begin{bmatrix} \psi \\ \beta \\ \zeta_1 \\ \zeta_2 \\ \omega \\ \dot{\beta} \\ \dot{\zeta_1} \\ \dot{\zeta_2} \\ x_e \end{bmatrix} \tag{3.17}$$

The non-linear system dynamics in state-space can now be formulated by considering the equations of motion in Equation (3.2):

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \\ \dot{x_e} \end{bmatrix} = \begin{bmatrix} \mathbf{q} \\ \boldsymbol{M}(\mathbf{q})^{-1} \left(-\boldsymbol{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \boldsymbol{g}(\mathbf{q}) + \boldsymbol{\tau}_m + \boldsymbol{N} \right) \\ e \end{bmatrix} \tag{3.18}$$

3.1.5 Numerical Implementation and Simulation

A numerical solver is used to solve the non-linear differential equations in Equation (3.18). The MATLAB computing platform [29], with the default solver ODE45 for non-stiff differential equations is used.

The parameters used for simulation are modeled after the physical rotor design, presented in Section 3.2. Some of the methodology in this numerical implementation is motivated by choices in that design process, such as the usage of Computer Aided Design (CAD) data as well as the blade aerodynamic modeling.

Pseudo-code of how Equation (3.18) is calculated at each time step can be found in Algorithm 1. In the rest of this section, the steps involved in this calculation are described.

Kinematics and Dynamics

The relevant kinematics and dynamics operations necessary to implement this system are the forward dynamics, computing Jacobians, as well as frame transforms. For all these tasks, there exist computationally efficient algorithms for rigid body chain formulations. MATLAB Robotics System Toolbox [30] is used for joint kinematics and dynamics.

The inertia and mass of each body of the rotor in their respective frame has been calculated from CAD data using CAD software Autodesk Fusion 360. The exact geometry of the rotor hubs is available from the design files, developed in Section 3.2.1. The geometry of the motor stator, motor rotor, bearings and rotor blade were reverse engineered. Each physical body modeled was assumed to be of uniform mass density, calculated from CAD volume divided by measured weight. The revolute joints were specified in CAD according to Figure 3.1. The tool Fusion2URDF [31] was used to specify body frames, joint frames, joint type, as well as mass and inertia for each body in respective frames as a URDF file. This format is supported in MATLAB Robotics Toolbox.

Aerodynamics

Numerical integration is used to solve the integrals in Equation (3.6) and Equation (3.7). The blade is divided into a discrete number of sections. For every section i, the differential sectional forces dF_z and dF_y are solved, either with the small angle approximations BEMT method, or with the full non-linear BEMT method. Both integrals are then approximated with the trapezoidal rule. The blade is modeled with varying geometry over the radial position r. In particular, the chord length c, the pitch θ, the lift coefficient c_l and the drag coefficient c_d are considered to vary over r. Values for these parameters are found for a few sections by measurements and simulation of the airfoil. This method is further elaborated in Appendix A.2. To calculate the value of these blade geometry parameters at any section i, interpolation is performed. 50 blade sections with interpolation are visualized in Figure 3.4

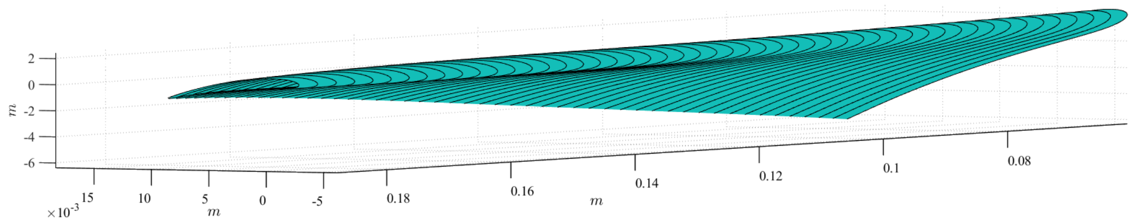


Figure 3.4: 50 Blade sections modeled for the MF1302 blade. Interpolation is performed between the geometric values found in Appendix A.2.

Open source BEMT implementation Rotare [32] is used for all BEMT related calculations. Rotare implements both the small angle approximations BEMT method described in Section 2.1.4, as well as the method described in Section 2.1.5. Implementing these methods does not necessarily require too much effort. A major reason to use Rotare is rather the extensive support of radial blade geometry specification and interpolation, as well as the provided validation of the implementation, increasing confidence in the correctness.

The software were modified to allow custom tangential and axial blade induced velocities for the BET part. Additionally, some modifications were made to handle negative axial inflow velocities.

Algorithm 1 Derivative of Rotor Equations of Motion State Space $\mathbf{x}$

```
    Input: $\mathbf{x} = \begin{bmatrix} \mathbf{q} & \dot{\mathbf{q}} & e \end{bmatrix}^T, \Omega$
    Output: $\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{q}} & \ddot{\mathbf{q}} & \dot{e} \end{bmatrix}^T$
$\psi \leftarrow \mathbf{q}(0)$                                        ▷ First element of $\mathbf{q}$
$\omega \leftarrow \dot{\mathbf{q}}(0)$                                        ▷ First element of $\dot{\mathbf{q}}$
$\omega^{ref} \leftarrow sinusoidalControlLaw(\psi, \Omega)$
$Q \leftarrow PIController(\omega, \omega^{ref}, e)$                              ▷ Motor Torque
for each blade $b$ do
    $\mathbf{J}_b \leftarrow geometricJacobian(\mathbf{q}, b)$
    $\mathcal{R}^b_{base} \leftarrow getRotation(\mathbf{q}, b)$
    $\mathbf{v}^{base}_b, \boldsymbol{\omega}^{base}_b \leftarrow \boldsymbol{J}_b(\mathbf{q})\dot{\mathbf{q}}$
    $\mathbf{v}_b \leftarrow \mathcal{R}^b_{base}\mathbf{v}^{base}_b$
    $\boldsymbol{\omega}_b \leftarrow \mathcal{R}^b_{base}\boldsymbol{\omega}^{base}_b$
    for each blade section $i$ do
        $\mathbf{v}_i \leftarrow \mathbf{v}_b + \boldsymbol{\omega}_b \times \mathbf{r}_i$
        $dF_{y_i}, dF_{z_i} \leftarrow BEMT(\Omega, \mathbf{v}_i)$
        $d\mathbf{f}_{aero} \leftarrow append(d\mathbf{f}_{aero}, [dF_{y_i}, dF_{z_i}])$
    end for
    $\mathbf{f}_{aero} \leftarrow trapezoidalIntegration(d\mathbf{f}_{aero})$
    $\boldsymbol{\tau}_{aero} \leftarrow trapezoidalIntegration(d\mathbf{f}_{aero}\mathbf{r})$
    $wrench \leftarrow wrench + composeWrench(\mathbf{q}, b, \boldsymbol{f}_{aero}, \boldsymbol{\tau}_{aero})$
end for
$\dot{\mathbf{q}} \leftarrow \dot{\mathbf{q}}$
$\ddot{\mathbf{q}} \leftarrow forwardDynamics(\mathbf{q}, \dot{\mathbf{q}}, Q, wrench)$
$\dot{e} \leftarrow \omega^{ref} - \omega$
```

3.2 Thrust Vectoring Rotor Implementation

In the associated specialization project, a rotor allowing up to 20 degrees was designed and 3D-printed. Thrust vector control was achieved for low speeds in a Python implementation, where sinusoidal modulation was running on a laptop, communicating at 400 Hz to the motor driver. This section builds on this work with improvements in design and implementation, while some previous findings such as the control laws remain.

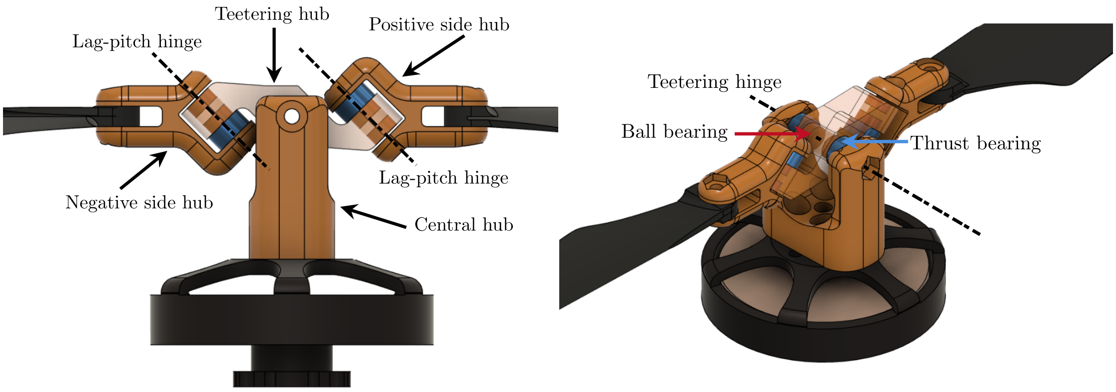


Figure 3.5: Annotated rotor design. Ball bearings are colored red, thrust bearings are colored blue.

3.2.1 Rotor Design

The designed thrust vector controllable rotor derives closely from the rotor design of Qin, Chen, Cai, *et al.* [6]. The rotor configuration in Figure 3.5 consists of a single teetering hinge and positive and negative 45° skewed lag-pitch hinges. A central hub connects the mechanism to the motor and acts as a mount for the teetering hinge. The teetering hub revolves around the teetering hinge and connects to the two side hubs. The side hubs are connected to the teetering hub through the skewed lag-pitch hinges.

Qin, Chen, Cai, *et al.* [6] uses two different types of bearings to achieve low friction, and these are used here. Each of the three hinges has two ball bearings to support radial load. For the teetering hinge, two thrust bearings clasp the teetering hub such that axial load can be supported between the central hub and the teetering hub. For the side hubs, only one thrust bearing is used for axial load. The argument given by [6] is that during

Figure 3.6: Iterative design process of the central hub

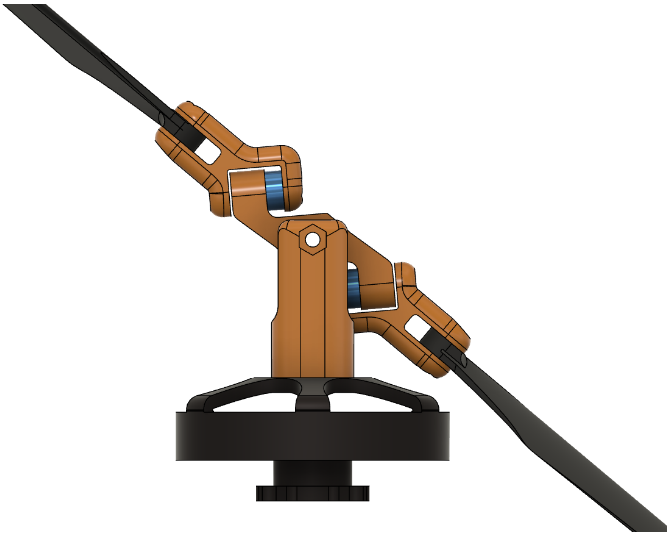


Figure 3.7: Rotor at maximum teetering angle of 40°. The maximum angle is limited by the central hub geometry, which is neccesary to limit angles at rest.

Figure 3.8: Geometrical design to allow large joint angles. The skewed lag-pitch joint is allowed up to 90° degree angles at all teetering angles up to the maximum. The teetering angle is shown at the maximum designed angle of 40°.

Table 3.1: Dimensional specification of mechanical parts for the thrust vectoring rotor. Bearing dimensions are given in (inner diameter) x (outer diameter) x (heigth)

Part	Dimension
Rotor radius R	0.19 m
Thrust bearing	3mm x 8mm x 3.5mm
Ball bearing	3mm x 6mm x 2.5mm
Bolts	M3

rotation, the lateral component of the centrifugal force will act on the side hubs, such that they are forced against the single thrust bearing.

The design choices consists of choice of hardware such as hinges and bolts, as well as geometry that connects these parts and the motor together.

Mechanical design specifications

To develop design specifications for the rotor, potential constraints were first identified. The first constraint was choosing the teetering, lag-pitch skewed rotor geometry. The second constraint consists of selecting a suitable motor and rotor blades sets the scale of forces and rotational speed. The T-Motor MF1302 propeller and MN5006 motor were selected. This is the same motor and propeller blade combination as used by [6] and [7]. Chen, Kong, Xu, *et al.* [7] provides an extensive comparison of motors and propeller blades. The motor and propeller combination mentioned above was selected by [7] for efficiency, suitability for a reasonable MAV size, and for motor acceleration properties. [7] is referred to for further details of these properties. Additionally, a similar motor and the same propeller size were shown to be more efficient with FOC compared to the standard six-step algorithm [27], as described in Section 2.3.6. A second constraint is the availability of miniaturized bearings. Small bearings that were readily available were chosen. The chosen mechanical parts and their dimensions are listed in Table 3.1. This provides geometrical specifications for hardware interfaces in the 3D-modeled parts.

The geometry of the 3D-modeled parts will limit the maximum hinge angles. Given a desired maximum thrust vector angle, the maximum teetering hinge angle is equal to this angle. To allow the possibility of large thrust vectoring elevation angles, the geometry is designed for teetering angles up to $40°$. The maximum skewed lag-pitch angle will depend on the dynamics of the rotor, and in this design process it is assumed to be up to $90°$. Figure 3.7 and Figure 3.8 show the geometry that was designed to allow these maximum angles.

Design Process

An iterative design process was used to design the 3D-modeled parts. Rapid prototyping with additive manufacturing allowed several 3D-printed parts to be evaluated for strength and acceptable geometry. After evaluation, design changes were made based on the results. Some produced versions of the central hub from this iterative process can be seen in Figure 3.6. 3D-printed rotor hubs are also used in [6], [5] and [7].

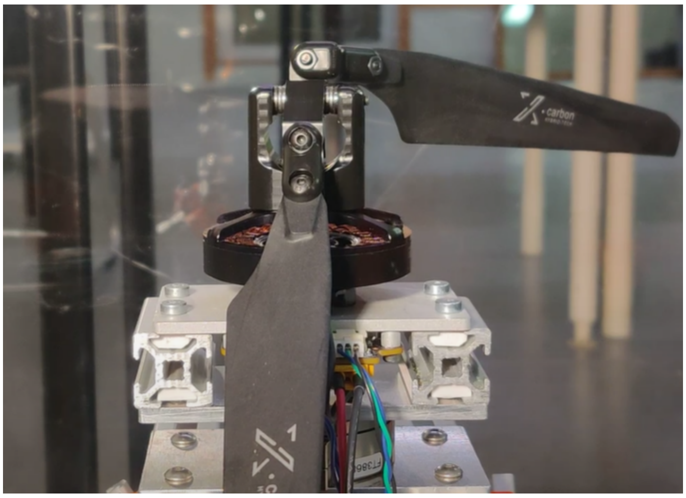


Figure 3.9: Rotor at rest position.

Materials and Production

Evaluating several 3D-printed parts, the strength and quality were found to not be sufficient. Some experimentation was made in optimizing the 3D-printed layer direction for maximum strength, as the 3D-printing process introduces non-isotropic properties. This was done due to early prototypes breaking along the layer direction when accelerating the rotor. Layer direction optimized parts broke in identified weak spots in the geometry, not across the layer direction. This could mean that the strength of the parts was limited by the material strength. The imperfections and variability in the 3D-printing process caused some parts to fail under normal operation as well.

Due to these limitations, the final geometry was machined in polyoxymethylene (POM), a high-strength, high-stiffness and low-friction plastic. This subtractive machining process produced higher quality parts with less imperfections and variability, better surface quality, and isotropic strength properties compared to the 3D-printing method. The low-friction properties of the plastic and the high surface quality from the machining process allowed tighter tolerances for the interface between the lag-pitch hinge and the teetering hinge, seen in Figure 3.5. During testing, none of the machined parts broke under normal operation.

At rest behaviour

Another consideration for the rotor design is the hinge configuration at rest, which can be seen in Figure 3.9. For the negative lag-pitch coupled blade, the center of mass of the blade and side hub lies below the lag-pitch hinge axis. A zero skewed lag-pitch angle is a stable equilibrium in this case, and the blade will rest with zero skewed lag-pitch angle. For the positive skewed lag-pitch coupled blade, the center of mass of the blade and side hub lies above the lag-pitch hinge axis. In this case, the zero skewed lag-pitch angle is an unstable equilibrium, causing the blade to rotate to one of the end positions of the lag-pitch hinge. By doing so, the blade moves closer to the center of the rotor, shifting the center of mass of the total assembly away from the center. This causes the rotor to rotate about the teetering hinge towards the negative lag-pitch coupled blade. The problem with this at rest behavior is that the rotor has to start from the most extreme teetering angle.

Thus, it will need full clearance from potential obstructions while rotating at this extreme.

3.2.2 Idealized Thrust Vectoring Model

Using the model provided in Equation (3.18) is not practical for the low-level torque modulation control. The numerical solver is not fast enough for the bandwidth required to modulate sinusoidal torque within each revolution at several thousand rpm. In anticipation of a simplified model for control, a steady state thrust vectoring definition is provided in Figure 3.10. The rotor is approximated to achieve a steady state propulsion disk direction and a steady state thrust force vector $\boldsymbol{f_T}$ given a set of inputs. Consider the reference frame in Figure 3.10, which is fixed to the motor base, not rotating with the rotor. The thrust vector $\boldsymbol{f_T}$ is parameterized by the force magnitude f, the elevation angle β_c and the azimuth angle ψ_c. The elevation angle β_c is defined as the maximum angle of teetering within each rotation. The azimuth angle ψ_c is given as the azimuth angle of maximum teetering, measured as right hand rotation about the z-axis, starting at the x-axis. Additionally, a drag induced torque $\boldsymbol{\tau_D}$ with magnitude τ_D is modeled, collinear with the thrust force vector.

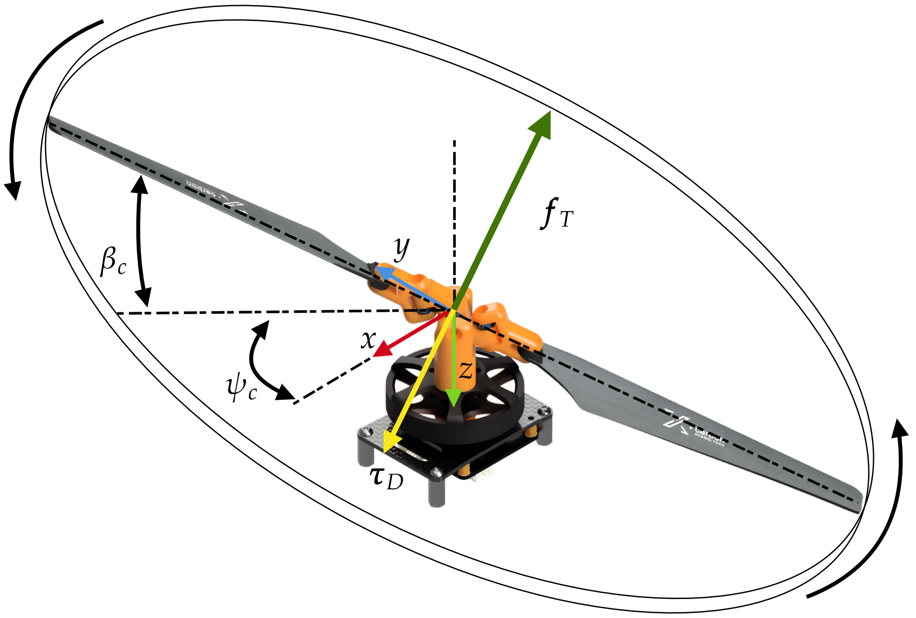


Figure 3.10: Definition of thrust generating propulsion disk. The elevation angle β_c is the maximum teetering angle within one revolution, with the xy-plane as reference. The azimuth angle ψ_c is the azimuth angle of maximum teetering, measured as right hand rotation about the z-axis, with the x-direction as zero. These angles defines the direction of the thrust force vector $\boldsymbol{f_T}$, and drag induced moment $\boldsymbol{\tau_D}$

3.2.3 Harmonic Motor Control Laws

Previous implementations of cyclical pitch with harmonic motor control such as [3], [5], [6] and [7] have used a PI speed controller with voltage as output, and feedforward voltage as harmonic modulator. It would be possible to implement this scheme using FOC by removing the current controller and directly controlling quadrature voltage u_q. There are potential advantages using the current controller. With FOC, we want to achieve smooth and maximum torque, which is given by the current in Equation (2.52). This allows controlling torque within bounds, and could potentially be used to generate smoother trajectories. Another advantage of using a current controller is that control of electrical and mechanical dynamics is decoupled and can be tuned separately.

The control objective is to control the thrust vector force magnitude f and direction β_c and ψ_c. [3] finds that there is an approximately linear relationship between the amplitude of a modulated sinusoidal voltage and maximum tilt angle β_c. [5] and [6] uses this fact to establish a linear relationship that can be calibrated. For [5], a minimum amplitude was necessary to overcome static friction of the pin hinges used. [6] uses a slightly different control law. As radial and axial low friction bearings are introduced, no minimum amplitude is assumed to be needed for blade tilt response. Similar low friction bearings have been implemented in Section 3.2.1 Rotor Design, and no minimum amplitude will be used. For the azimuth angle response, [3] finds that ψ_c lags the modulated input voltage by an angle. In [5], a constant offset angle is used to compensate, while in [6] a function dependent on the average motor speed Ω is used.

Motor Direction Definition

Depending on the rotorcraft configuration, rotors rotating both clockwise (CW) and counter clockwise (CCW) are needed if torque balance is required due to $\boldsymbol{\tau}_D$. CW motor rotation will be defined as right hand rotation about a downward facing z-axis. CCW rotation will be defined as negative right hand rotation about a downward facing z-axis. A CCW rotor is shown in Figure 3.10. For both cases, motor position is given by $\psi \in [-\pi, \pi]$, and motor speed by $\omega > 0$. Motor position zero, $\psi = 0$ is defined as the motor position where the positive blade is aligned with the x-axis of the reference frame in Figure 3.10. The defined positive direction of rotation is defined such that CW rotation is positive for a CW rotor, and CCW rotation is positive for a CCW rotor.

Harmonic Speed Control Law

The principle of operation of the mechanism relies on a harmonic acceleration of the rotor. It is possible to formulate a sinusoidal motor speed reference that results in sinusoidal rotor acceleration. The input reference to the speed controller is proposed as

$$\begin{aligned} \omega_{ref} &= \Omega + \tilde{\Omega} \\ \tilde{\Omega} &= A\Omega \sin(\psi + \psi_k) \end{aligned} \tag{3.19}$$

where ψ is the motor angle, and ψ_k is a phase offset compensation. The speed reference sinusoidal term $\tilde{\Omega}$ is given by the input $A \in [0, 1]$ multiplied by the average speed Ω. This choice is motivated by the anticipated linear relationship between A and β_c, in addition to having an intuitive amplitude command in terms of speed. With this control law, the following approximations are proposed:

$$\begin{aligned} f &= k_T \Omega^2 \\ \tau_D &= k_M f \\ \beta_c &= k_A A \\ \psi_c &= d(\psi_k - \psi_a) \end{aligned} \tag{3.20}$$

where ψ_a is a constant azimuth phase offset angle. $d = 1$ for a CCW rotor and $d = -1$ for a CW rotor. For convenience and implementation reasons, the directional sign d is encoded in k_M, such that k_M is negative for a CCW rotor, and positive for a CW rotor. To analyze the torque of this control law, the motor speed reference can be inserted in the motor torque expression from Equation (3.13).

$$\tau_m = \frac{3}{2} K_e \left[K_{P_v}(\Omega - \omega) + K_{I_v} \int (\Omega - \omega) \mathrm{d}t + K_{P_v}(\tilde{\Omega}) + K_{I_v} \int (\tilde{\Omega}) \mathrm{d}t \right] \tag{3.21}$$

Harmonic torque feedforward

An alternative control law with harmonic motor torque modulation can enter through a torque feedforward term. The harmonic torque feedforward control law is proposed as

$$\begin{aligned} \omega_{ref} &= \Omega \\ \tau_m^{ff} &= A_\tau \sin(\psi + \psi_k) \end{aligned} \tag{3.22}$$

where ψ_k is given in Equation (3.20). Similarly to Equation (3.21), τ_m can be calculated for this control law.

$$\tau_m = \frac{3}{2} K_e \left[K_{P_v}(\Omega - \omega) + K_{I_v} \int (\Omega - \omega) \mathrm{d}t \right] + \tau_m^{ff} \tag{3.23}$$

The integral term with modulated sinusoidal signal apparent in Equation (3.21) does not appear with this feedforward modulation, as the speed PI controller is bypassed. This could improve control performance in terms of phase delay. The voltage control law in [3] has sinusoidal voltage bypass the speed controller as well.

In terms of u_q voltage, an integral term will still appear due to the PI current controller. It is assumed that the electrical dynamics are much faster than the mechanical dynamics, such that the current controller can be tuned to a significantly higher bandwidth than the speed controller.

3.2.4 Motor Control Implementation

A field oriented controller within design specifications of the chosen motor and propeller is chosen. Unlike the six-step control algorithm implemented in most off-the-shelf BLDCM drivers, FOC requires certain parameters of the motor, as well as tuning. For this reason, FOC drivers are often designed for a specific motor. A configurable field oriented controller with open source firmware were chosen to have control over parameters, and not rely on the manufacturer for motor integration. The open source firmware is also exploited for custom control laws. The mjbots moteus r4.11 controller was chosen. It is reasonably electrically and mechanically dimensioned for the motor and load, weighing 14.2 g with a peak phase current rating of 100 A and a peak electrical power rating of 500 W. The circuit board has an integrated hall effect absolute magnetic encoder for measuring the rotor position, such that the whole unit is mounted below the motor. Interfacing is done through the Controller Area Network Flexible Data-rate (CAN-FD) protocol, capable of up to 5 Mbps.

The FOC algorithm is not computationally cheap and should run at a high rate. The moteus controller uses a 32-bit STM32G4 microprocessor in the main microcontroller. The internal firmware runs a FOC loop with similar structure as shown in Figure 2.5. Additionally an outer configurable PID position or speed loop is implemented on the controller. The whole control system runs at a configurable update rate of up to 40 kHz.

Custom Sinusoidal Modulation Firmware

Performing sinusoidal modulation from Section 3.2.3 to hold a commanded thrust vector can be decomposed from generating the commanded thrust vector. In general, it can be assumed that this once per revolution modulation needs to run at a higher bandwidth than the controller outputting thrust vector commands.

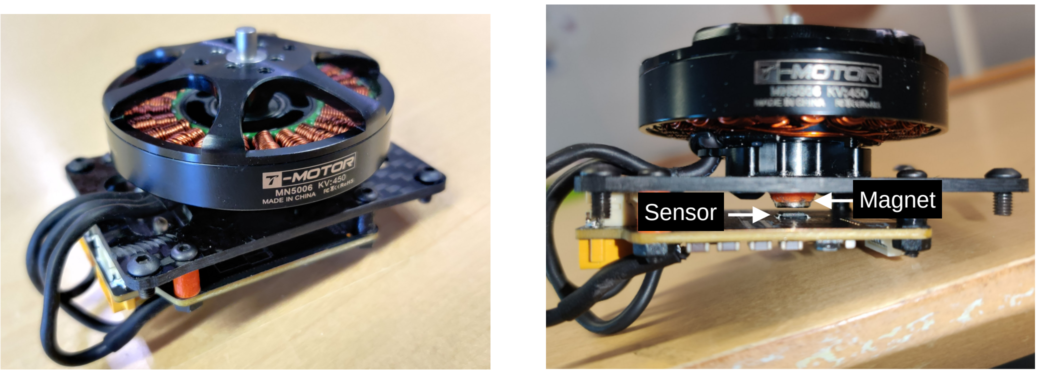
This argument is further motivated by the open source moteus driver. The driver runs high-rate cascaded motor control with accurate motor position feedback, with even stricter demands than expected for the sinusoidal modulation. If there are enough processor cycles available for the additional sinusoidal modulation, implementation here would be possible.

With this motivation, a custom motor driver firmware is implemented. Firmware modification is supported by a well written modular interface in the open source firmware. Two additional control modes are implemented for the two control laws in Section 3.2.3. Each of these can be enabled through the moteus configuration system. In these modes, the motor speed PI controller is modified to Equation (3.19), and Equation (3.22), respectively for the two modes. The motor command interface of the driver is extended to include A and ψ_c. Standard speed command in the driver is used for Ω in these modes. Figure 3.12 shows an overview of controllers, communication and hardware used for the harmonic speed control law.

Rotor position measurement

An AS5047P magnetic encoder with an angular resolution of $2\pi/4096$ is integrated in the moteus controller, which communicates through SPI to the main microcontroller at 12 MHz and is sampled at every iteration. In comparison, [6] and [7] use the AS5600 magnetic encoder with a slower I2C protocol, where they achieved a sampling rate of position measurements of 920 Hz, and 910 Hz, respectively.

To measure the position of the rotor, a diametrically magnetized magnet is used. A magnetization across the diameter allows the direction of the magnetic field to be used as a measure of the rotor position. The drive shaft of the motor extends through the center of the stator to the mounting base of the motor, where such a magnet can be mounted. For optimal performance, the magnet should be placed accurately in relation to the encoder, which resides in a surface mounted chip package on the moteus controller board. A 3D-



(a) Motor and motor driver module with designed carbon fiber mounting solution.

(b) Magnetic encoder, with 3D-printed magnet holder jig in orange.

Figure 3.11: Module with motor and motor driver. The diametrically magnetized magnet used for rotor position measurement is shown in (b).

Table 3.2: Measured T-Motor MN5006 motor parameters

Parameter	Value
Motor electrical constant K_e	0.014 V/(rad/s)
Stator resistance R_{ohm}	0.108 ohm
Stator inductance L	1.409×10^{-5} H
Number of pole pairs N_p	14

printed magnet guide jig was designed and used to achieve accurate positioning. See Figure 3.11. Initial testing with a manual, off-center magnet mount resulted in excessive audible noise from the motor.

As the encoder measurement is noisy, a filter is used for mechanical position, and this filtered position is used to determine mechanical speed ω to be used in velocity control. A computationally efficient all-digital phase-locked loop filter is implemented by the moteus driver, and the bandwidth can be configured.

An offset position is defined such that the positive blade aligns with azimuth angle zero as defined in Section 3.2.2.

Parameter estimation

Several motor parameters are needed for the FOC algorithm, and are in this case not given by the manufacturer. Estimation of stator inductance L, stator resistance R and back EMF constant K_e was performed. To estimate these parameters, a calibration script by mjbots bundled with the moteus controller were used. The details of this script are described below.

Stator resistance is measured by applying different u_d voltages, measuring the corresponding current and using Ohm's law to calculate the average resistance. Stator inductance is measured by applying a u_d reference and measuring the rate of change of current. Then the definition of inductance, $L = u(t)/\frac{di}{dt}$ can be used. d-axis voltage is used in both cases, such that there is no torque applied to the motor, resulting in a zero rotor speed. In the case of resistance measurement, there are no mechanical losses. In the case of inductance measurements, there are no mutual inductances.

The motor velocity constant k_v, defined as $k_v = \omega_{noload}/u$ is estimated by the calibration script by applying a quadrature voltage u_q and measuring the motor mechanical speed ω. This was performed without any propeller or similar connected to the motor.

The back-EMF constant k_e appearing in Equation (2.52) is estimated as $k_e = 1/k_v$ under the assumption of no mechanical losses. As there are clearly losses when spinning the motor, this method is not perfectly accurate.

Alternatively to how k_e was estimated in this project, a more direct method is possible. By spinning the motor externally, operating it as a generator, a relation between rotational velocity and back-EMF can be established, circumventing the mechanical losses. This method is used by [27].

Current Controller Tuning

The current feedback loop in dq-frame implemented in the moteus controller can be written as

$$\begin{aligned} u_q &= K_{P_i}(i_q^{ref} - i_q) + K_{I_i}\int(i_q^{ref} - i_q)\mathrm{d}t \\ u_d &= K_{P_i}(i_d^{ref} - i_d) + K_{I_i}\int(i_d^{ref} - i_d)\mathrm{d}t \end{aligned} \tag{3.24}$$

where K_{P_i} and K_{I_i} are constants for the current PI-controller.

Tuning of the current controller is performed by pole placement. This is a part of the calibration script of the moteus controller. A desired torque bandwidth is selected, this is used to set the bandwidth of the current controller. The poles are then placed based on the chosen bandwidth. The phase-locked loop filter for motor position is set by the driver to have twice the bandwidth of selected torque bandwidth. More filtering will result in smoother torque signal, while excessive filtering will limit torque bandwidth. Ideally, the mechanical dynamics of the motor should be evaluated to see what torque bandwidth is physically possible. The torque bandwidth was set to 800 Hz after evaluating several bandwidths for dynamical performance versus vibrations. A higher torque bandwidth were neccesary to allow higher gains in the speed controller at higher speeds.

Speed Controller Tuning

The speed PI-controller implemented in the moteus controller can be written as

$$\begin{aligned} i_q^{ref} &= K_{P_v}(\omega_{ref} - \omega) - K_{I_v}\int(\omega_{ref} - \omega)\mathrm{d}t + \frac{2}{3K_e}\tau_m^{ff} \\ i_d^{ref} &= 0 \end{aligned} \tag{3.25}$$

where K_{P_v} and K_{I_v} are constants for the speed PI-controller and τ_m^{ff} is an optional torque feedforward. τ_m^{ff} is expressed in i_q by Equation (2.53). From the relation between motor torque and quadrature current in Equation (2.52), the resulting motor torque can be expressed.

A manual tune were performed for the speed PI controller. An initial tune were determined for the motor with no rotor attached. With the rotor mounted to the motor, higher gains were possible with the increased rotational inertia. A high gain were chosen, allowing a minimal amount of overshoot to maximize rise time. The integral term were chosen by setting a constant velocity setpoint to achieve a zero steady state error, where a mainly speed dependent torque disturbance needs to be integrated. This well known disturbance could have been handled with a torque feedforward, potentially improving performance.

3.2.5 Thrust Vector Command Implementation

Thrust vector commands need to be generated or handled for both flight and calibration. Then, calculated motor commands needs to be sent to the motor driver. A C++ executable project to handle thrust vector control was developed for the single-board computer Raspberry Pi 4B+ with the mjbots pi3hat CAN-FD controller board.

Implementation Details

A basic motor driver interface example over CAN-FD supplied by mjbots was used as a starting point for implementation. This includes CAN-FD drivers and an interface to the default motor driver firmware. The motor driver interface was modified to include the new motor commands A and ψ_k.
For the overall design, an object-oriented pattern were implemented, and key implementation details are summarized here. An abstract *Controller* class calculates the thrust vector command at a given time instance. This abstract class has two implementations. The *CalibrationController* class implements thrust vector command generation based on input vectors of motor command sequences and their duration. Such a sequence of motor command vectors can be generated by a function which calculates a sequence given minimum, maximum and number of commands. Separate values are accepted for each motor command, and the combination of all possibilities is returned. The *ThrustVectorController* class reads input thrust vector commands and maps them to motor commands. The mapping is defined in Equation (3.20).

A main command loop runs the chosen *Controller* class implementation. At each timestep, motor commands are calculated and sent over CAN-FD, and motor telemetry data is received. Both motor commands and motor telemetry data are saved to memory, used for analysis.

Reading Thrust Vector Commands with PWM

Thrust vector commands are sent from the flight controller with pulse width modulation(PWM), where the signal is mapped to pulse widths between 1000 µs and 2000 µs. One PWM channel is used for each command in terms of thrust magnitude, azimuth and elevation. The PWM signals are connected to the general input output pins of the Raspberry Pi. Decoding of the PWM signal is implemented as a part of the thrust vector command software, using the *pigpio* C library for pin interrupts. Each PWM input is assigned an object of a *PWMReader* class. A callback function is registered on level change of the respective input pin. The number of system clock ticks in microseconds is measured from a rising edge to a falling edge. This is the pulse width. The number of clock ticks is stored as a 32-bit unsigned integer by *pigpio*. Every 2^{32} µs, roughly every 1

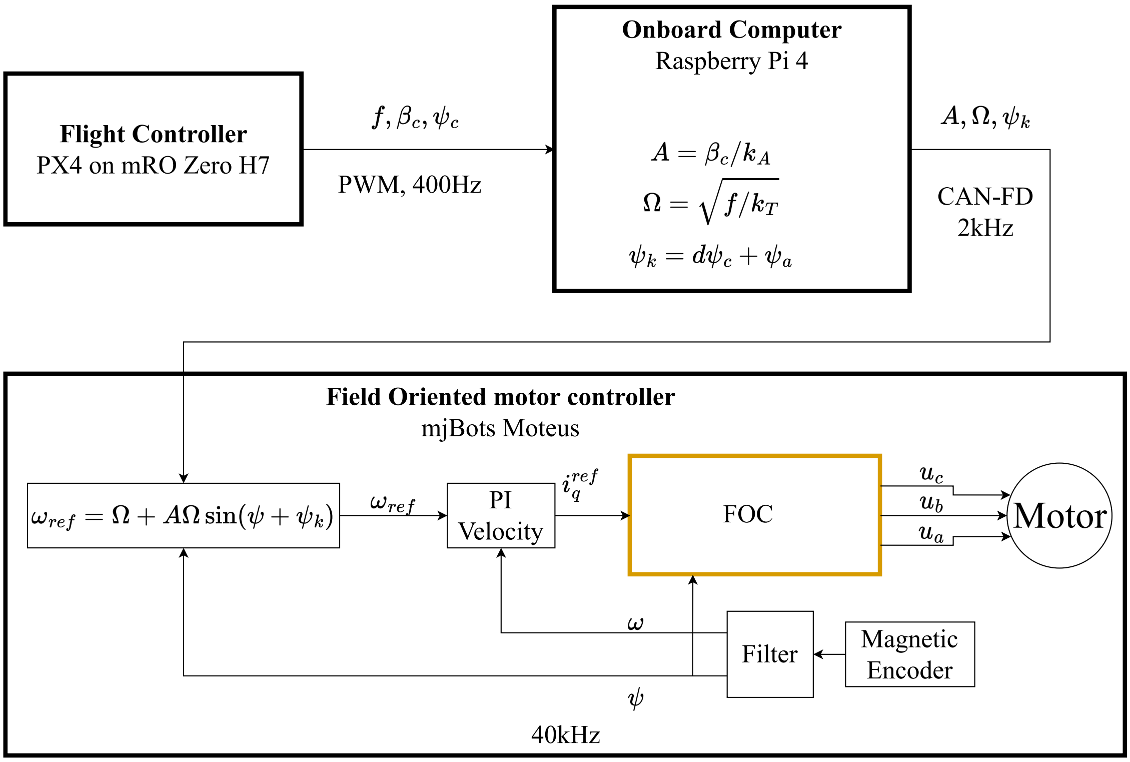


Figure 3.12: Thrust vector control pipeline, from thrust vector command at flight controller, to motor control. The flight controller outputs thrust vector commands as magnitude f and angles β_c and ψ_c. The companion computer calculates motor commands in terms of sinusoidal amplitude A, average motor speed Ω and phase ψ_c. These are sent to the motor controller over CAN-FD, where motor speed setpoints are calculated by the sinusoidal control law.

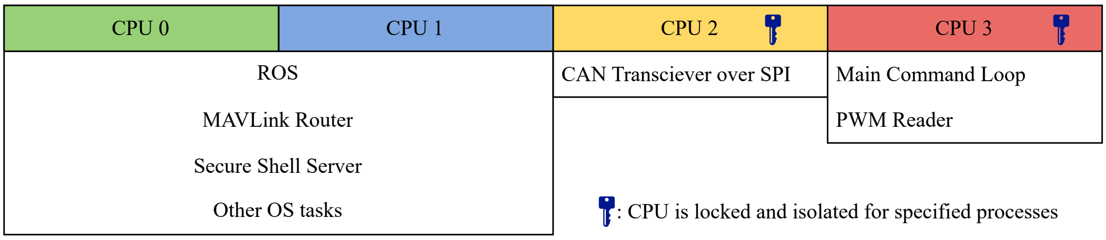


Figure 3.13: Overview of the most important tasks and processes running on the Raspberry Pi companion computer

hour and 12 minutes, the number of ticks wraps around to zero. This wrap around is handled by calculating the pulse width as the modulus of the tick counts and the maximum number of ticks. This operation ensures that the correct pulse width is always computed, even when the tick count overflows and wraps around to zero.

Running Near Real-Time on the Raspberry Pi

The task of receiving PWM signals and sending the corresponding motor commands over CAN-FD has real-time demands. To measure the PWM pulse widths, microsecond precision is necessary. When sending motor commands, the latency has to be bounded to some maximum. If the latency is too high, the drone will become unstable and cause catastrophic failure.

These demands require a real-time system. The flight controller with autopilot software PX4[33] runs the NuttX Real Time Operating System (RTOS), which would be well suited for the tasks above. The Raspberry Pi does not run a real-time system, using the Linux operating system without a real-time kernel. Processes can be unbounded by being interrupted by other unbounded tasks.

The reason for not implementing thrust vector command handling on the flight controller is that there does not exist a compatible driver between the flight controller and the motor driver used. The ideal way to solve this would be to implement a CAN-FD interface between the flight controller and the motor driver. In this project, this was not attempted. This is mostly due to the effort and time usage necessary for this driver development. CAN-FD support has been very recently added in PX4 and its RTOS. There is also an additional effort required to make a driver compatible with the PX4 output driver interface.

To better handle real-time tasks on the Raspberry Pi, several techniques were used. In result, the scheduling can in practice be almost assured to be bounded. These techniques are described in detail by Fairhead [34], and motivated by mjbots pi3hat example implementation. The techniques used are summarized below.

- Using the performance governor for critical CPUs, preventing clock speed reduction.
- Isolating critical CPUs with the $isolcpus$ kernel parameter, preventing other processes and interrupt routines from running on these CPUs.
- Set a high real-time scheduling attribute with $chrt$.

- Lock current and future address space of the process with the Linux kernel function $mlockall$. This prevents having to page into memory.

The chosen utilization of the four CPU cores on the Raspberry Pi can be seen in Figure 3.13. The Raspberry Pi is responsible for a range of higher-level tasks that run according to the Linux scheduler on the first two cores. The two last cores are isolated, and only specific processes are allowed to run. CPU 2 is used for the CAN transceiver developed by mjbots. CPU 3 runs both the main command loop, as well as the *PWM-Reader* callbacks on level change. Having these processes on separate cores did not yield a measurable performance increase compared to having them use the same core.

3.2.6 Thrust Vectoring Calibration and Measurement

The mechanical test stand was designed and produced during the specialization project. In this work, the calibration and software described here were developed.

The goal of the calibration is to model the Section 3.2.2 Idealized Thrust Vectoring Model as dependent on motor commands Ω, β_c and ψ_c.

The model depends on the motor control law. Only the harmonic speed control law will be considered. The model for the thrust vector and the drag moment is given in Equation (3.20). Calibration is performed experimentally by performing a calibration routine, measuring force and moment. The software described in Section 3.2.5 is used to generate sequences of motor commands that are executed.

Measurement Setup

The experimental test setup consists of the test rig seen in Figure 3.14 with the rotor from Section 3.2.1 and motor driver running the custom sinusoidal firmware. A 6-axis force torque (FT) sensor and an inertial measurement unit (IMU) are used. The FT sensor is used for thrust vector calibration and measurement, while the IMU is used to look at vibrations. Figure 3.14 illustrates the IMU, force/torque sensor and rotor mounted on the test rig.

The IMU used is the VectorNav VN-100. Accelerometer and gyroscope data is sampled at 800 Hz. The default sensor coordinate frame is used. Accelerations are measured in the x, y, z axes as defined in Figure 3.14. Gyroscope yaw, pitch, roll angular rate measurements are defined as right-handed rotations around z, y, x, respectively.

The 6-axis FT sensor used is the ATI Mini45. A data acquisition system reads analog strain gauge voltage measurements and outputs processed values for force and torque in 6 axes. The system was configured to acquire analog samples at 10 kHz, averaging 10 samples and outputting 1 kHz FT data. The reference frame for the FT measurements was transformed from the sensors default to measure the thrust vector with frames as in figure 3.10. This transformed frame is ground-fixed at the center of the teetering hub, and consists of a translation of 77.5 mm in positive z direction and a negative $60°$ rotation about z. See Figure 3.14. The FT sensor has high uncertainty for the forces and torques that will be measured. The 95$\%$ confidence level measurement uncertainty for torques are in the order of magnitude of 0.1 Nm in the default coordinate system. For the transformed frame, this corresponds to 95$\%$ confidence level measurement uncertainty in the order of magnitude of 1N for x and y forces.

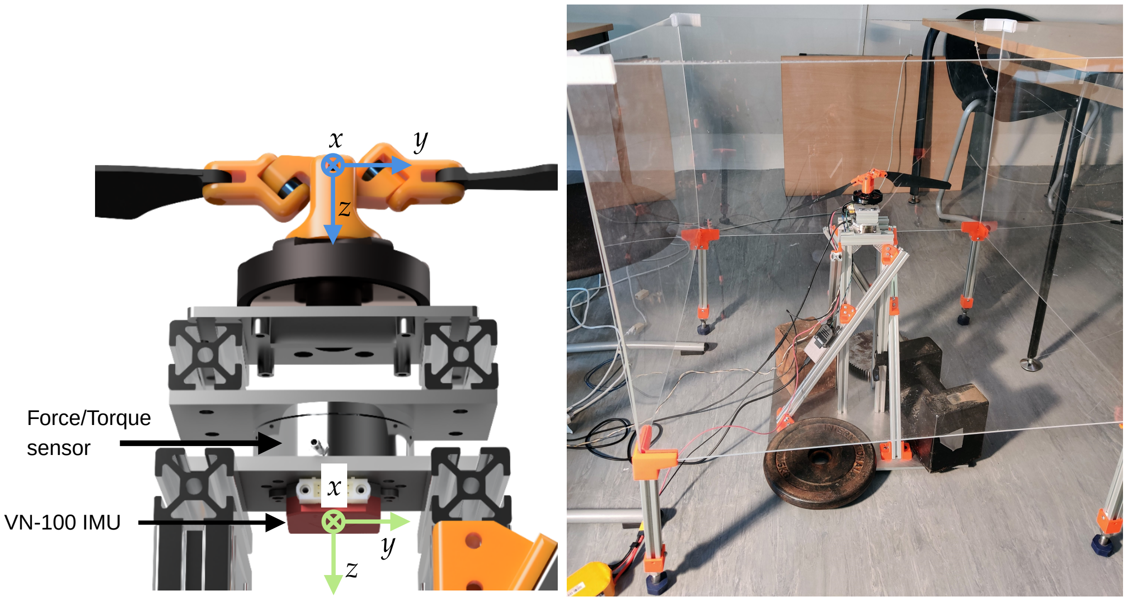


Figure 3.14: Experimental measurement setup. Force and moment sensor axes are marked in blue. IMU acceleration axes are marked in green. Both frames are ground-fixed and do not rotate with the rotor.

Mapping from Force Measurements to Azimuth-Elevation Angles

Let F_x, F_y, and F_z be the measured FT sensor force components in the x, y, and z directions, respectively, in the frame given in Figure 3.14. The thrust vector azimuth ψ_c and elevation β_c angles can be calculated as:

$$\psi_c = arctan2(F_y, F_x) \tag{3.26}$$

$$\beta_c = arctan2(\sqrt{F_x^2 + F_y^2}, -F_z) \tag{3.27}$$

where arctan2 is the routine which returns the arc tangent of the two inputs, considering the sign of both arguments to determine the correct quadrant of the result.

Regression technique

Regression has been used to determine the constants. In all the models used, curve fitting an arbitrary model is achieved with non-linear least squares. Specifically, the *optimize.curve_fit* method of the Python package *SciPy* [35] is utilized. Even though ordinary least squares could be used for the linear models developed, having a single method that handles all models used in the calibration software makes a cleaner interface with no practical downside.

Calibration and Analysis Software

To accurately perform the regression, clean and time synchronized data are needed for both motor commands and measured force and torque. The force and moment data is

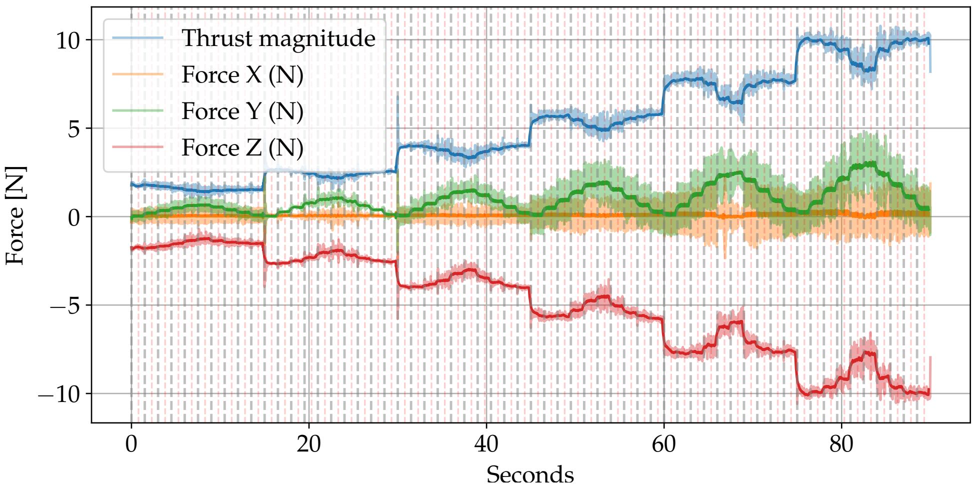


Figure 3.15: Example calibration routine force measurements. Raw data are shown in lighter color, and an exponential moving average is shown in darker color. The vertical gray dotted lines mark the automated motor command section endpoint timestamps, the red marks the section midpoints.

noisy, and not time synchronized with the motor commands. Additionally, there are transient responses when changing motor commands, while the idealized model only accounts for the steady-state response. A set of Python scripts was developed to handle these tasks.

The time series of motor command data is divided into sequences where each sequence represents a unique combination of motor commands. The average force and moment is then calculated within each sequence, such that each unique motor command combination results in one data point. A small time period of the first and last part of each sequence is not used for averaging, to account for transients and small time misalignment.

The time synchronization between motor commands and FT data is done manually, with visual guidance. When initially running the script, the time offset between the two datasets is set to zero. A new field is generated for the FT data, where a manual offset can be set. Visual guidance of the start, center, and end time of the motor command sequence is overlaid on the FT data to aid and verify alignment, shown in Figure 3.15.

3.3 Rotorcraft Design

3.3.1 Integrating Thrust Vectoring Rotors in Rotorcraft

Choice of Rotorcraft Configurations

Integrating the thrust vectoring rotor in a rotorcraft opens some interesting discussions about which configuration would be the most performant or suitable. Configuration here relates to the number of rotors as well as their orientation and size. Common configurations would include the helicopter, coaxial rotorcraft with two rotors as well as multirotors with a number of rotors typically positioned in the same plane. Additionally, a range of vehicles with wings for lift could also be potential use cases for the thrust vectoring rotor, however they will not be considered here.

The coaxial rotorcraft configuration with two thrust vectoring rotors has been chosen for this thesis. It offers a large potential for disk loading and thus efficiency, compared to multirotors and helicopters of the same footprint. An additional reason to choose to use two thrust vectoring rotors is the emulation of a fully actuated rotorcraft, where force and moment can be generated in all directions.

3.3.2 Coaxial Rotorcraft System Overview

To evaluate the thrust vectoring rotor in flight, a rotorcraft has been designed. One approach could have been to integrate the thrust vectoring rotor into an off-the-shelf rotorcraft, but finding a suitable model proved challenging. A rotorcraft with thrust vectoring rotors will have different design considerations compared to common configurations such as a quadcopter or a helicopter. To keep weight and size down, a high level of integration is required for all parts as well, which can be achieved by designing a custom rotorcraft from the ground up for this concept. Design choices can be optimized with the total design in mind. With an off-the-shelf rotorcraft, modeling certain effects often requires reverse engineering. Such effects could be CAD data for rotor placement, mass distribution and inertia. Using a forward engineering approach like here simplifies further modeling, some of which are considered in this thesis.

The design of the rotorcraft is here referred to as designing the frame and the placement of propulsion and components. These are the most specialized parts for the thrust vectoring rotorcraft, while other parts can be selected off-the-shelf and integrated.
An overview of the rotorcraft design can be seen in Figure 3.16.

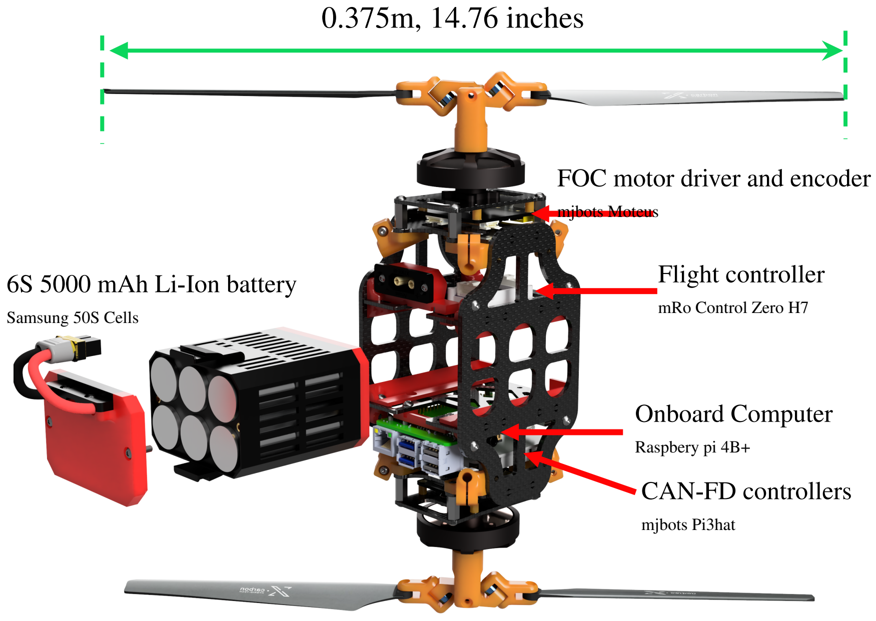


Figure 3.16: Overview of the designed coaxial rotorcraft and the main components. The landing gear and protective cage are not included here

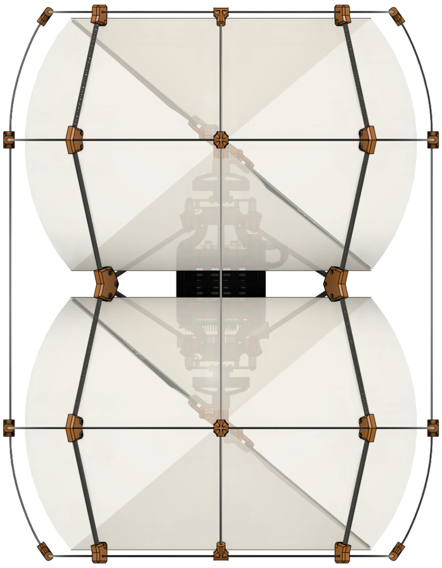


Figure 3.17: Coaxial rotorcraft with operational space of each rotor illustrated in the gray volume.

Design Specifications

The main design specifications that will drive design decisions for the coaxial rotorcraft are the rotor size and the rotor spacing. The rotor size, $R = 0.19$ m, were decided in Section 3.2.1 Rotor Design. Choosing the rotor spacing, referred to as l_d, could rely on simulation for maximum efficiency. Here, other practical considerations drive the rotor spacing. Consider the maximum teetering angle of $40°$ of the thrust vectoring rotor in Section 3.2.1, chosen to investigate the possibility of large angles. The rotors do not tilt synchronously, as this would limit the degrees of freedom in the system. Therefore, they cannot have any overlap in operational space. From a geometric consideration, the minimum spacing $l_{d,min}$ for the rotors is

$$l_{d,min} = 2R\sin\left(\max(\beta_c)\right) \tag{3.28}$$

where R is the radius of each rotor. The total vertical height of the operational space of the rotors can be calculated as $l_h = l_d + 2R\sin\left(\max(\beta_c)\right)$. Thus, the minimum height is $l_{h,min} = 2l_{d,min}$. With $R = 0.19$ m and $\max(\beta_c) = 40°$, $l_{d,min} = 0.24$ m and $l_{h,min} = 0.48$ m. Compared to R, both values are quite high. Given the footprint the rotorcraft will occupy, a value close to the minimum is desirable.

There is an additional consideration for $l_{d,min}$. With off-the-shelf motors for MAVs, it is not possible to extend non-rotating structure through the rotor. Any structure extending above the top rotor or below the bottom rotor must pass through a gap, with a height of $l_{gap} = l_d - l_{d,min}$, between the two rotors. Such structure would include the landing gear. Thus, the final rotor-rotor distance can be given by $l_d = l_{d,min} + l_{gap}$.

With these considerations, the operational space of each rotor, the gap between them and the final configuration can be seen in Figure 3.17. The final parameters chosen are $l_d = 0.27$ m and $l_{gap} = 0.03$ m.

Frame

The coaxial propulsion system drives the main design decisions in the frame design. The main structure is contained under the hub of the rotors when the thrust vectoring elevation angle is zero. Airflow, especially further out along the blade, where the induced velocities are higher, is minimally obstructed in this configuration. Six carbon fiber sheets are used as the main structural elements of the central frame in Figure 3.16. The resulting midsection, containing all electronics and rotors, is stiff and lightweight. Four carbon fiber sheets of thickness 2 mm are contained between two side carbon fiber sheets of thickness 3 mm. These four carbon fiber sheets slots into the side carbon sheets, and contains mounting geometry for the electronics.

An outer cage was designed. This functions both as necessary landing gear, as well as collision protection.

Easy repair and modification have been a goal of the design. By removing one of the side carbon fiber sheets, as well as two additional screws, the entire midsection can be removed from the cage and propulsion units for maintenance.

Battery

A custom battery mounting solution and a custom battery pack was designed for the rotorcraft. The custom mounting solution is motivated by having the battery, a significant contributor to weight and inertia, securely fastened to the center of the frame. A compliant mechanism in a 3D-printed battery case was designed to allow easy battery replacement between flights. The final design of the mechanism is shown in Figure 3.18 was tested with more than 100 insertions, without signs of degradation.

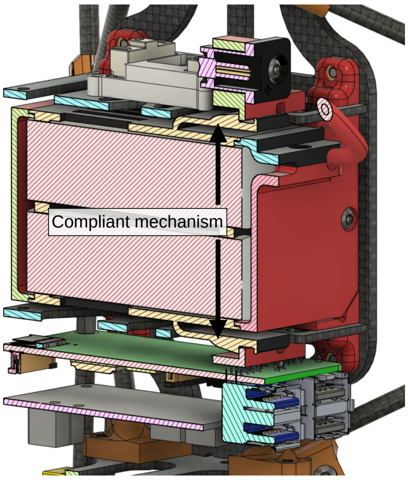
A custom battery pack is motivated by flight efficiency. Off-the-shelf batteries for MAVs are typically high discharge rate lithium ion polymer packs, as many MAVs have a high current draw. Considering the current draw of the propulsion system, a battery pack consisting of six series connected Samsung 50S 21700 lithium ion cylindrical cells were made. Each cell has a manufacturer rated maximum discharge rate of up to 45 A with cooling, and a capacity of 5000 mAh. The total six series pack has a rated energy of 111 Wh at nominal current draw, with a graviometric energy density of 260 Wh/kg. An off-the-shelf lithium ion polymer battery that fits the custom battery case is also used. This battery has a rated total energy of 33.3 Wh and a graviometric energy density of 138 Wh/kg.

Electrical Design Considerations

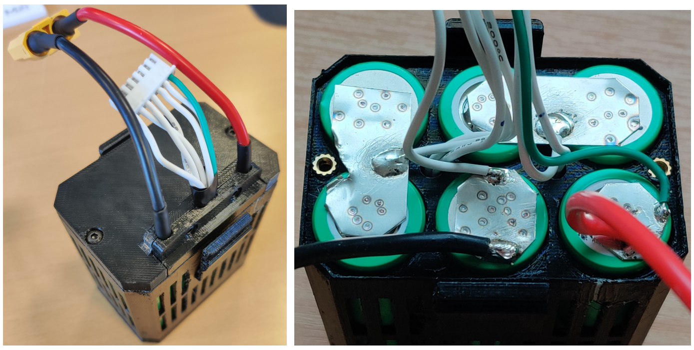
The CAN drivers are particularly sensitive to voltage transients and input over-voltage. This caused several CAN driver failures in the first electrical design of rotorcraft. Intermittent or bad power connections can cause voltage arcs resulting in high voltage transients. If these transients enter the CAN driver through ground, they can cause damage.

To combat this, the rotorcraft electrical design was redesigned to include a two-step connector and strain-relieved power cables. The first stage of the two-step connector positive lead contains a resistor. This limits current flow and reduces the possibility of arching. Full insertion of the connector bypasses the resistance.

Strain-relieving the power cables reduces stresses on the motor driver power connectors. There is less chance of intermittent connection, increasing stability in the system.



(a) Section view of rotorcraft showing battery pack compliant mechanism



(b) Assembled battery. **(c)** Series cell construction.

Figure 3.18: Custom battery pack with six lithium ion series cells.

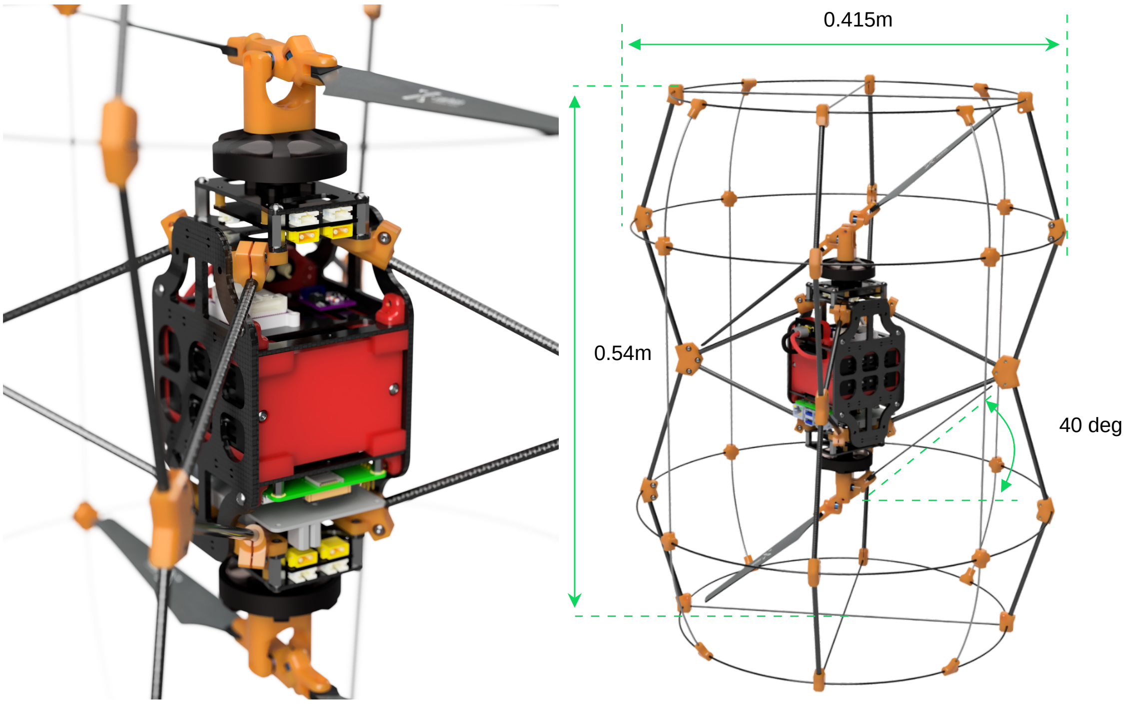


Figure 3.19: Designed coaxial rotorcraft with protective cage. The image on the left shows the defined rear of the rotorcraft. The $40°$ angle illustrates the maximum thrust vectoring actuation that the frame geometry is designed for.

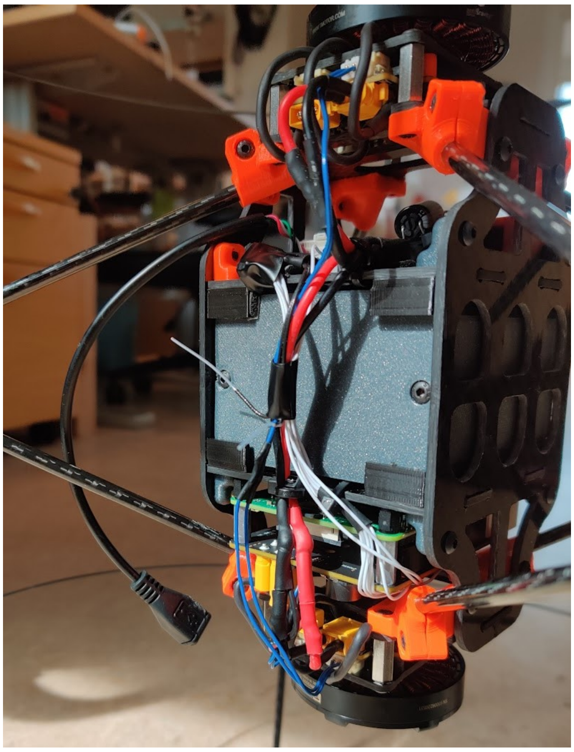


Figure 3.20: Rotorcraft rear side with wiring.

Another design consideration in this case is safety. The power connectors to the individual motor drivers were disconnected when the rotorcraft was not in operation to prevent any unexpected motor activation. However, these connectors have a finite number of reinsertion cycles, in addition to voltage transient risks due to the capacitive load of the motor and driver.

To maintain safety while the motor drivers are powered, removable propeller blockers were designed and used. A simple design of four carbon tubes is inserted over the rotorcraft body, blocking the rotor blades in the event of unintended rotor spin-up. The blocker is removed before flight.

3.4 Rotorcraft Modeling and Control

This section introduces modeling and unconstrained control allocation for general rotorcraft with n_τ thrust vectoring rotors. A model accounting for the interactions between the rotors and the rotorcraft is introduced in Section 3.4.1, while an idealized model is introduced in Section 3.4.2. Modeling the specific coaxial rotorcraft configuration is shown with the idealized model in Section 3.4.4.

3.4.1 Rotorcraft - Rotor Coupled Model

Modeling the full dynamics of a rotorcraft with n_τ rotors as modeled in section 3.1 includes the coupled dynamics of the rotorcraft and rotor systems. One way to model this system is to assume that the rotorcraft has a central rigid body where all rotor bases are rigidly attached. This rigid multibody system can be modeled as a kinematic chain. It has a floating base, as the central body of the rotorcraft can move freely in translation and rotation in six degrees of freedom. A minimal set of generalized coordinates can be constructed for this kinematic system. If the rotor with generalized coordinates from Equation (3.1) is chosen, the total system with n_τ rotors will be $\boldsymbol{q}_p \in \mathbb{R}^{6+4n_\tau}$. By considering the central body position and orientation in relation to an inertial frame $\mathcal{I}$, one such choice can be:

$$\boldsymbol{q}_p = \begin{bmatrix} x_p \\ y_p \\ z_p \\ \phi_p \\ \theta_p \\ \psi_p \\ \psi_1 \\ \beta_1 \\ \zeta_{1_1} \\ \zeta_{2_1} \\ \vdots \\ \psi_{n_\tau} \\ \beta_{n_\tau} \\ \zeta_{1_{n_\tau}} \\ \zeta_{2_{n_\tau}} \end{bmatrix} \tag{3.29}$$

where the six floating base generalized coordinates are marked in blue, and the generalized coordinates of the $4n_\tau$ rotors are marked in green. p_x, p_y, p_z are the translation coordinates of the central body in $\mathcal{I}$, and p_ϕ, p_θ, p_ψ are the orientation of the central body given by Euler angles in frame $\mathcal{I}$. Assuming that the external forces and moments are the per rotor aerodynamic forces, motor torque and a rotorcraft drag force and torque entering in $\boldsymbol{\tau}_D$, the equations of motion of this system can be written as

$$\boldsymbol{M}_p(\boldsymbol{q}_p)\ddot{\boldsymbol{q}}_p + \boldsymbol{C}_p(\boldsymbol{q}_p, \dot{\boldsymbol{q}})_p\dot{\boldsymbol{q}}_p + \boldsymbol{g}_p(\boldsymbol{q}_p) = \sum_{i=1}^{n_\tau}(\boldsymbol{\tau}_{m_i} - \boldsymbol{n}_{aero,i}) + \boldsymbol{\tau}_D \tag{3.30}$$

where $\boldsymbol{\tau}_{m_i}$ and $\boldsymbol{n}_{aero,i}$ must be defined such that they have terms for rotor i in joint space. The open loop dynamical model in state space, neglecting the error state in Equation (3.18), can be written as

$$\dot{\boldsymbol{x}}_p = \begin{bmatrix} \dot{\mathbf{q}}_p \\ \ddot{\mathbf{q}}_p \end{bmatrix} = \begin{bmatrix} \mathbf{q}_p \\ \boldsymbol{M}_p(\mathbf{q})_p^{-1} \left(-\boldsymbol{C}_p(\mathbf{q}_p, \dot{\mathbf{q}})_p \dot{\mathbf{q}}_p - \boldsymbol{g}_p(\mathbf{q}_p) + \sum_{i=1}^{n_\tau}(\boldsymbol{\tau}_{m_i} - \boldsymbol{n}_{aero,i}) + \boldsymbol{\tau}_D\right) \end{bmatrix} \tag{3.31}$$

A model of the aerodynamic forces of rotor i, $\boldsymbol{N}_i$, will in general need to include the rotorcraft body velocities, blade orientation and velocity with the form $N_i(\dot{x}_p, \dot{y}_p, \dot{z}_p, \dot{\phi}_p, \dot{\theta}_p, \dot{\psi}_p, \psi_i, \beta_i, \zeta_{1_i}, \zeta_{2_i}, \dot{\psi}_i, \dot{\beta}_i, \dot{\zeta}_{1_i}, \dot{\zeta}_{2_i})$.

3.4.2 Idealized Model

The idealized thrust vector model from Section 3.2.2 is used to arrive at a simplified model of external forces and moments. Assuming that the steady state thrust force and drag induced moment for each rotor are the only external forces and moments acting on the system introduces several assumptions:

- Each rotor produces an instantaneous thrust vector with a well defined mapping.
- Dynamic effects of the thrust vectoring such as gyroscopic torque are neglected.
- No rotorcraft-rotor coupling effects.
- No rotor-rotor coupling effects.

Further, the equations of motion from Equation (3.30) are approximated as a single floating base rigid body, a well-known equation. This approximation reduces the generalized forces and moments to $\mathbb{R}^6$.

$$\boldsymbol{\tau} = \begin{bmatrix} \boldsymbol{\tau}_{\mathcal{B}} \\ \boldsymbol{f}_{\mathcal{B}} \end{bmatrix}, \tag{3.32}$$

where $\boldsymbol{f}_{\mathcal{B}}$ is the body frame force vector and $\boldsymbol{\tau}_{\mathcal{B}}$ is the body frame moment vector, both acting on the single body modeled.

Coordinate Frames

An inertial, world fixed frame $\mathcal{I}$ is considered. A rotorcraft body-fixed frame $\mathcal{B}$ is considered, with origin in the center of mass. This frame follows the Forward-Right-Down convention, with the x-axis pointing at the forward direction, y-axis to the right and z-axis down.

Each thrust vectoring rotor has its own thrust vector frame. These rotor frames are labeled as $\mathcal{T}_i, i = \{1, ..., n_\tau\}$, where n_τ are the number of rotors. The thrust vector of each rotor i is assumed to be given by three inputs, force magnitude f_i, elevation direction β_i and azimuth direction ψ_i. The origin of this frame is the same as $\mathcal{B}_0$ in Figure 3.2. The vector $\boldsymbol{r}_i$ defines the position of the origin of $\mathcal{T}_i$, given in body frame $\mathcal{B}$. Zero azimuth and elevation angle are defined with the thrust vector frame aligned with the body frame.

The rotation from $\mathcal{T}_i$ to $\mathcal{B}$ can be expressed as a sequence of two rotations similar to Euler angles. This sequence is defined by first a right handed rotation ψ_i about $\mathcal{B}_z$. The second rotation β_i is defined as right handed rotation about the y-axis of the ψ_i rotated frame.

$$\mathcal{R}_{\mathcal{T}_i}^{\mathcal{B}} = R_Z(\psi_i) R_Y(\beta_i) \tag{3.33}$$

From Equation (2.33), this rotation matrix can be written as

$$\mathcal{R}^{\mathcal{B}}_{\mathcal{T}_i} = \begin{bmatrix} C_{\psi_i}C_{\beta_i} & -S_{\psi_i} & C_{\psi_i}S_{\beta_i} \\ S_{\psi_i}C_{\beta_i} & C_{\psi_i} & S_{\psi_i}S_{\beta_i} \\ -S_{\beta_i} & 0 & C_{\beta_i} \end{bmatrix}. \tag{3.34}$$

This sequence is equivalent to a rotation β_i about $\mathcal{B}_y$ followed by a rotation ψ_i about $\mathcal{B}_z$.

Forces and Moments in Body Frame

As defined above, the thrust force vector $\boldsymbol{f}_{T_i}$ is aligned to the rotor frame z-axis. A rotor drag induced moment $\boldsymbol{\tau}_{D_i}$ is considered collinear with the thrust force vector as defined in section 3.2.2.

$$\boldsymbol{f}_{T_i} = \begin{bmatrix} 0 \\ 0 \\ f_i \end{bmatrix}, \boldsymbol{\tau}_{D_i} = \begin{bmatrix} 0 \\ 0 \\ k_{M_i} f_i \end{bmatrix} \tag{3.35}$$

where k_{M_i} is the constant that provides a linear mapping between force magnitude for rotor i, and drag induced moment for rotor i. The total body frame forces $\boldsymbol{F}_{\mathcal{B}}$ and moments $\boldsymbol{\tau}_{\mathcal{B}}$ is expressed as

$$\begin{aligned} \boldsymbol{f}_{\mathcal{B}} &= \sum_{i=1}^{n_{\mathcal{T}}} \mathcal{R}^{\mathcal{B}}_{\mathcal{T}_i} \boldsymbol{f}_{T_i} \\ \boldsymbol{\tau}_{\mathcal{B}} &= \boldsymbol{\tau}^{\boldsymbol{f}}_{\mathcal{B}} + \boldsymbol{\tau}^{\boldsymbol{D}}_{\mathcal{B}} \\ \boldsymbol{\tau}^{\boldsymbol{f}}_{\mathcal{B}} &= \sum_{i=1}^{n_{\mathcal{T}}} \boldsymbol{r}_i \times \mathcal{R}^{\mathcal{B}}_{\mathcal{T}_i} \boldsymbol{f}_{T_i} \\ \boldsymbol{\tau}^{\boldsymbol{D}}_{\mathcal{B}} &= \sum_{i=1}^{n_{\mathcal{T}}} \mathcal{R}^{\mathcal{B}}_{\mathcal{T}_i} \boldsymbol{\tau}_{D_i} \end{aligned} \tag{3.36}$$

where $\boldsymbol{\tau}^{\boldsymbol{F}}_{\mathcal{B}}$ is the moment due to the displaced thrust vector and $\boldsymbol{\tau}^{\boldsymbol{D}}_{\mathcal{B}}$ is the drag induced moment.

3.4.3 Unconstrained Control Allocation

The control allocation problem concerns the coordination of control actuators to achieve a desired generalized forces and moments. Johansen and Fossen [36] provides an overview of this topic. The mapping from input $\boldsymbol{u}$ to generalized force and moment given by the equations of motion can be written as

$$\boldsymbol{\tau} = \boldsymbol{B} u, \tag{3.37}$$

where $\boldsymbol{B}$ is referred to as the actuator effectiveness matrix. In the idealized model presented in Section 3.4.2, the generalized forces and moments are the the body forces and moments

$$\boldsymbol{\tau} = \begin{bmatrix} \boldsymbol{f}_{\mathcal{B}} \\ \boldsymbol{\tau}_{\mathcal{B}} \end{bmatrix} \tag{3.38}$$

Then, , $\boldsymbol{B} \in \mathbb{R}^{6 \times 3n_{\mathcal{T}}}, \boldsymbol{u} \in \mathbb{R}^{3n_{\mathcal{T}}}$. From this it is possible to conclude that the system is under-actuated for $n_{\mathcal{T}} = 1$, over-actuated for $n_{\mathcal{T}} > 2$, and neither for $n_{\mathcal{T}} = 2$. In the following equations, terms relating to rotor 1 will be marked in red and the terms relating to rotor n_τ will be marked in yellow for readability. The control input vector $\boldsymbol{u}$ in terms of per rotor thrust vector force magnitude, azimuth angle and elevation angle is

$$\boldsymbol{u} = \begin{bmatrix} f_1 & \psi_1 & \beta_1 & \dots & f_i & \psi_i & \beta_i \end{bmatrix}^T \tag{3.39}$$

With this choice of $\boldsymbol{u}$, the $\boldsymbol{B}$ matrix can be calculated by evaluating Equation (3.36).

$$\boldsymbol{B} =$$

$$\begin{bmatrix} c_{\psi_1} s_{\beta_1} & 0 & 0 & \dots & c_{\psi_{n_\tau}} s_{\beta_{n_\tau}} & 0 & 0 \\ s_{\beta_1} s_{\psi_1} & 0 & 0 & \dots & s_{\beta_{n_\tau}} s_{\psi_{n_\tau}} & 0 & 0 \\ c_{\beta_1} & 0 & 0 & \dots & c_{\beta_{n_\tau}} & 0 & 0 \\ \begin{array}{c} r_{1,y} c_{\beta_1} + k_{M_1} c_{\psi_1} s_{\beta_1} - \\ r_{1,z} s_{\beta_1} s_{\psi_1} \end{array} & 0 & 0 & \dots & r_{n_\tau,y} c_{\beta_{n_\tau}} + k_{M_i} c_{\psi_{n_\tau}} s_{\beta_{n_\tau}} - r_{n_\tau,z} s_{\beta_{n_\tau}} s_{\psi_{n_\tau}} & 0 & 0 \\ \begin{array}{c} k_{M_1} s_{\beta_1} s_{\psi_1} - r_{1,x} c_{\beta_1} + \\ r_{1,z} c_{\psi_1} s_{\beta_1} \end{array} & 0 & 0 & \dots & \begin{array}{c} k_{M_i} s_{\beta_{n_\tau}} s_{\psi_{n_\tau}} - r_{n_\tau,x} c_{\beta_{n_\tau}} + \\ r_{n_\tau,z} c_{\psi_{n_\tau}} s_{\beta_{n_\tau}} \end{array} & 0 & 0 \\ \begin{array}{c} k_{M_1} c_{\beta_1} - r_{1,y} c_{\psi_1} s_{\beta_1} + \\ r_{1,x} s_{\beta_1} s_{\psi_1} \end{array} & 0 & 0 & \dots & \begin{array}{c} k_{M_i} c_{\beta_{n_\tau}} - r_{n_\tau,y} c_{\psi_{n_\tau}} s_{\beta_{n_\tau}} + \\ r_{n_\tau,x} s_{\beta_{n_\tau}} s_{\psi_{n_\tau}} \end{array} & 0 & 0 \end{bmatrix} \tag{3.40}$$

This matrix has terms coupled in $\boldsymbol{u}$ and is non-linear. In this case, the problem can be transformed into a linear problem with a new choice of input. The thrust force vector of rotor i can be expressed as decomposed body frame forces.

$$\begin{aligned} f_{i,x} &= f_i \cos(\psi_i) \sin(\beta_i) \\ f_{i,y} &= f_i \sin(\psi_i) \sin(\beta_i) \\ f_{i,z} &= f_i \cos(\beta_i) \end{aligned} \tag{3.41}$$

If the control input, now $\hat{\boldsymbol{u}}$, is expressed in terms of decomposed body frame forces, the new $\hat{\boldsymbol{B}}$ becomes linear.

$$\boldsymbol{\tau} = \hat{\boldsymbol{B}} \hat{\boldsymbol{u}} \tag{3.42}$$

$$\hat{\boldsymbol{u}} = \begin{bmatrix} f_{1,x} & f_{1,y} & f_{1,z} & \dots & f_{i,x} & f_{i,y} & f_{i,y} \end{bmatrix}^T \tag{3.43}$$

$$\hat{\boldsymbol{B}} = \begin{bmatrix} 1 & 0 & 0 & \dots & 1 & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 1 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 & 1 \\ k_{M_1} & -r_{1,z} & r_{1,y} & \dots & k_{M_i} & -r_{i,z} & r_{i,y} \\ r_{1,z} & k_{M_1} & -r_{1,x} & \dots & r_{1,z} & k_{M_i} & -r_{1,x} \\ -r_{1,y} & r_{1,x} & k_{M_1} & \dots & -r_{1,y} & r_{1,x} & k_{M_i} \end{bmatrix} \tag{3.44}$$

This force decomposition is a common solution for vectoring thrusters [36]. This method is used for one-DOF vectoring rotors in [11], [12], and for two-DOF vectoring rotors in [13] and [5].

The challenge of unconstrained control allocation is then to invert this linear model. This will solve the unconstrained allocation problem. For $n_\tau = 2$, $\hat{\boldsymbol{B}}$ is a square matrix, and if the determinant is non-zero, it is invertible. For $n_\tau \neq 2$, $\hat{\boldsymbol{B}}$ is not square and not invertible. A common solution, especially for over-actuated systems, is to use a generalized inverse [36]. The Moore-Penrose pseudo-inverse is defined as [36]

$$\hat{\boldsymbol{B}}^+ = \hat{\boldsymbol{B}}^T (\hat{\boldsymbol{B}} \hat{\boldsymbol{B}}^T)^{-1} \tag{3.45}$$

such that the solution for $\hat{\boldsymbol{u}}$ can be defined as

$$\hat{\boldsymbol{u}} = \hat{\boldsymbol{B}}^{+}\boldsymbol{\tau}_c \tag{3.46}$$

where $\boldsymbol{\tau}_c \in \mathbb{R}^6$ are the commanded body frame forces and moments. The Moore-Penrose pseudo-inverse is especially suitable, as it can be shown to be the minimum norm solution of the inverse problem [36], and in the case of an invertible $\hat{\boldsymbol{B}}$, it coincides with the inverse. The square of the norm of $\hat{\boldsymbol{u}}$ is

$$||\hat{\boldsymbol{u}}||^2 = (\sum_{i=1}^{n_\tau} f_i)^2 \sum_{i=1}^{n_\tau} (\cos(\psi_i)\sin(\beta_i))^2 + (\sin(\psi_i)\sin(\beta_i))^2 + (\cos(\beta_i))^2 \tag{3.47}$$

Kamel, Verling, Elkhatib, *et al.* [11] shows that a proportionality to the sum of squared force magnitude leads to consistent and power efficient solutions for a similar problem, assuming the same thrust coefficient for all rotors.

For $n_\tau = 1$, $\hat{\boldsymbol{B}}$ is rank-deficient, and actuating force and moment is not possible in all $\mathbb{R}^6$. Rank-deficiency is also possible for other n_τ.

3.4.4 Coaxial Rotorcraft

Modeling and unconstrained control allocation of the coaxial rotorcraft designed in Section 3.3 can be performed with the methods described in this section. Two rotors, $n_\tau = 2$, with rotor frames $\mathcal{T}_i, i = \{1, 2\}$ are considered. The body frame $\mathcal{B}$ and the external forces $\boldsymbol{f}_{T_i}$ and moments $\boldsymbol{f}_{D_i}$ are visualized in Figure 3.21. For this configuration, the body frame forces and moments become

$$\begin{aligned}
\boldsymbol{F}_{\mathcal{B}} &= \mathcal{R}^{\mathcal{B}}_{\mathcal{T}_1}\boldsymbol{f}_{T_1} + \mathcal{R}^{\mathcal{B}}_{\mathcal{T}_2}\boldsymbol{f}_{T_2} \\
\boldsymbol{\tau}_{\mathcal{B}} &= \boldsymbol{\tau}^{\boldsymbol{f}}_{\mathcal{B}} + \boldsymbol{\tau}^{\boldsymbol{D}}_{\mathcal{B}} \\
\boldsymbol{\tau}^{\boldsymbol{D}}_{\mathcal{B}} &= \boldsymbol{r}_1 \times \mathcal{R}^{\mathcal{B}}_{\mathcal{T}_1}\boldsymbol{f}_{T_1} + \boldsymbol{r}_2 \times \mathcal{R}^{\mathcal{B}}_{\mathcal{T}_2}\boldsymbol{f}_{T_2} \\
\boldsymbol{M}^{\boldsymbol{D}}_{\mathcal{B}} &= \mathcal{R}^{\mathcal{B}}_{\mathcal{T}_1}\boldsymbol{\tau}_{D_1} + \mathcal{R}^{\mathcal{B}}_{\mathcal{T}_2}\boldsymbol{\tau}_{D_2}
\end{aligned} \tag{3.48}$$

The control input vector is now $\hat{\boldsymbol{u}} \in \mathbb{R}^6$ and the actuator effectiveness matrix is now $\hat{\boldsymbol{B}} \in \mathbb{R}^{6\times 6}$.

$$\hat{\boldsymbol{u}} = \begin{bmatrix} f_{1,x} & f_{1,y} & f_{1,z} & f_{2,x} & f_{2,y} & f_{2,y} \end{bmatrix}^T \tag{3.49}$$

$$\hat{\boldsymbol{B}} = \begin{bmatrix}
1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 \\
k_{M_1} & -r_{1,z} & r_{1,y} & k_{M_2} & -r_{2,z} & r_{2,y} \\
r_{1,z} & k_{M_1} & -r_{1,x} & r_{2,z} & k_{M_2} & -r_{1,x} \\
-r_{1,y} & r_{1,x} & k_{M_1} & -r_{2,y} & r_{2,x} & k_{M_2}
\end{bmatrix} \tag{3.50}$$

Since this matrix is square, it is invertible if the determinant is non-zero. If it is assumed that each rotor only has a z-displacement in body frame, $r_{1,y} = r_{2,y} = r_{1,x} = r_{2,x} = 0$, the determinant of $\hat{\boldsymbol{B}}$ is

$$\det \hat{\boldsymbol{B}} = -(k_{M_1} - k_{M_2})^3 \tag{3.51}$$

which is invertible when $k_{M_1} \neq k_{M_2}$. The Moore-Penrose pseudo-inverse can be used in any case, however, with a zero determinant $\hat{\boldsymbol{B}}$ is rank-deficient, and full actuation of

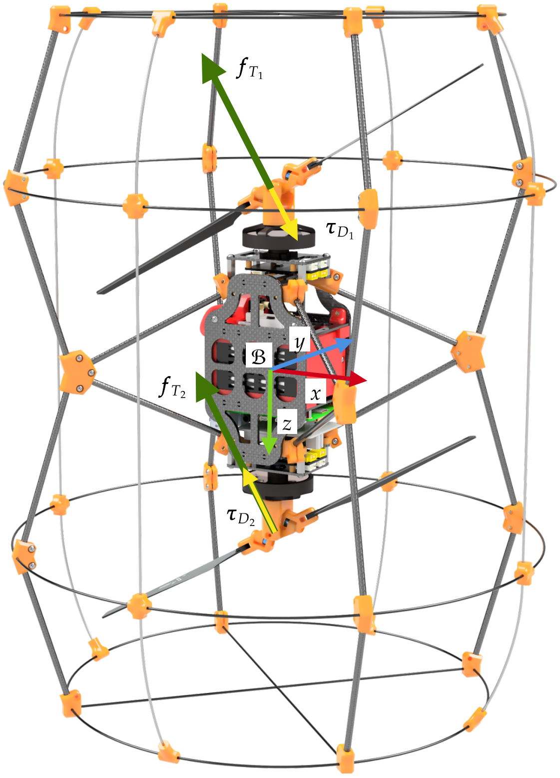


Figure 3.21: Coaxial rotorcraft external forces, external moments and body frame $\mathcal{B}$.

force and torque in $\mathbb{R}^6$ is not possible. Note that this does not account for the constraints in force magnitude, thrust vectoring angles and moment magnitude.

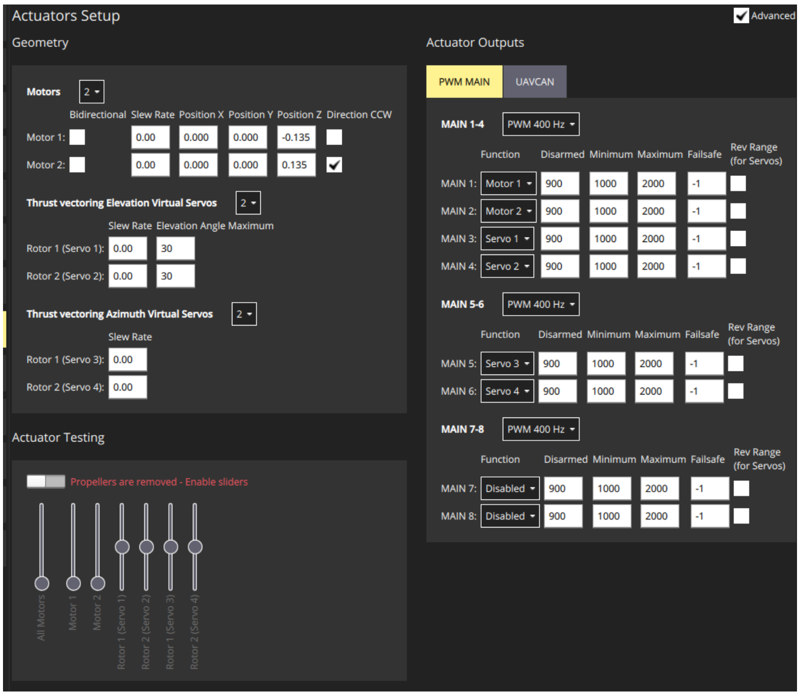
There are additional requirements for operation that are not captured in unconstrained control allocation. For the coaxial rotorcraft, the two rotors are required to spin in opposite directions to control body z-axis moment in a range around zero. In Figure 3.21, it can be seen that the upper rotor, $i = 1$, spins CCW, while the lower rotor, $i = 2$ spins CW.

Paulos, Caraher, and Yim [5] presents an actuator effectiveness matrix for a similar thrust vectoring coaxial rotorcraft. Here, the drag induced moment is only modeled as a factor of $f_{i,z}$. In that work, a maximum thrust vectoring elevation angle of $9°$ was possible, which could have motivated this approximation.

3.4.5 Rotorcraft Control - High Level View

Performing trajectory and attitude control for rotorcraft typically consists of a cascade of controllers. For the conventional under-actuated rotorcraft, translation and orientation are coupled. A typical control strategy starts with a desired rotorcraft acceleration vector. A rotorcraft attitude reference is then calculated according to the acceleration vector, such that body-frame force can be actuated in the direction of the desired acceleration. An angle controller calculates desired body rates based on the current rotorcraft orientation and desired orientation. An inner rate controller calculates desired body frame moments $\boldsymbol{\tau}_{\mathcal{B}}$ based on the current body rates and desired body rates. These are then used in the control allocation module.

For a fully-actuated rotorcraft, where body-frame force $\boldsymbol{F}_{\mathcal{B}}$ and moment $\boldsymbol{\tau}_{\mathcal{B}}$ can be actuated in any direction, the coupling between rotorcraft orientation and translation is



Motors 2

	Bidirectional	Slew Rate	Position X	Position Y	Position Z	Direction CCW
Motor 1:	☐	0.00	0.000	0.000	-0.135	☐
Motor 2:	☐	0.00	0.000	0.000	0.135	☑

Thrust vectoring Elevation Virtual Servos 2

	Slew Rate	Elevation Angle Maximum
Rotor 1 (Servo 1):	0.00	30
Rotor 2 (Servo 2):	0.00	30

Thrust vectoring Azimuth Virtual Servos 2

	Slew Rate
Rotor 1 (Servo 3):	0.00
Rotor 2 (Servo 4):	0.00

MAIN 1-4 PWM 400 Hz

	Function	Disarmed	Minimum	Maximum	Failsafe	Rev Range (for Servos)
MAIN 1:	Motor 1	900	1000	2000	-1	☐
MAIN 2:	Motor 2	900	1000	2000	-1	☐
MAIN 3:	Servo 1	900	1000	2000	-1	☐
MAIN 4:	Servo 2	900	1000	2000	-1	☐

MAIN 5-6 PWM 400 Hz

	Function	Disarmed	Minimum	Maximum	Failsafe	Rev Range (for Servos)
MAIN 5:	Servo 3	900	1000	2000	-1	☐
MAIN 6:	Servo 4	900	1000	2000	-1	☐

MAIN 7-8 PWM 400 Hz

	Function	Disarmed	Minimum	Maximum	Failsafe	Rev Range (for Servos)
MAIN 7:	Disabled	900	1000	2000	-1	☐
MAIN 8:	Disabled	900	1000	2000	-1	☐

Figure 3.22: Actuator setup user interface in QGroundControl. The number of thrust vectoring rotors, their position, their rotation direction as well as the maximum elevation angle is configurable here.

not neccesary. A desired acceleration can be given regardless of the desired orientation. Paulos, Caraher, and Yim [5] uses such a control strategy for a fully actuated rotorcraft.

If constraints and non-ideal systems are considered, there is still coupling between the rotorcraft orientation and actuation of forces and moments in world-frame. Rashad, Goerres, Aarts, *et al.* [37] covers techniques for fully-actuated rotorcraft with constraints, considering rotorcraft with laterally bounded force actuation. The thrust vectoring rotors in this work are constrained in maximum elevation angle, which constrains the maximum lateral force to be lower than the maximum vertical force. This is not accounted for in the unconstrained control allocation implementation of this work.

3.4.6 Autopilot Integration

Implementation of control allocation for thrust vectoring rotors with n_τ rotors, limited by hardware outputs, has been developed for the open source PX4 autopilot software [33]. The dynamic control allocation module of PX4 has been used, where many of the required operations and logic are implemented in a modular and object-oriented interface.

This interface was extended to accommodate the force decomposition described in Section 3.4.3. Specifically, an abstract class *ActuatorEffectiveness* is extended with a virtual function *transformActuatorControls*. This function allows for a variable change in the control allocation input vector with a default implementation of no transform. An implementation of *ActuatorEffectiveness* is made for thrust vectoring rotorcraft, overriding this function as well as implementing the actuator effectiveness matrix $\hat{\boldsymbol{B}}$ from Section 3.4.3. Additionally, a custom airframe class was implemented, as well as custom parameters that were made available through the QGroundControl user interface, seen in Figure 3.22.

4

Results and Discussion

Experiments with the developed methods, hardware and software are presented and discussed in this chapter.

4.1 Thrust Vectoring Rotor System

The experimental results of the rotor system consists of simulations of the rotor model from Section 3.1, and measured data from the real rotor as described in Section 3.2.6. In both these cases, it is important to note that this is a fixed-base model, and the measured rotor is fixed-base in the sense that it is rigidly attached to the measurement test stand. This allows analysis of the decoupled rotor system from the rotorcraft, which is convenient. It will not account for all effects experienced on a floating-base rotorcraft.

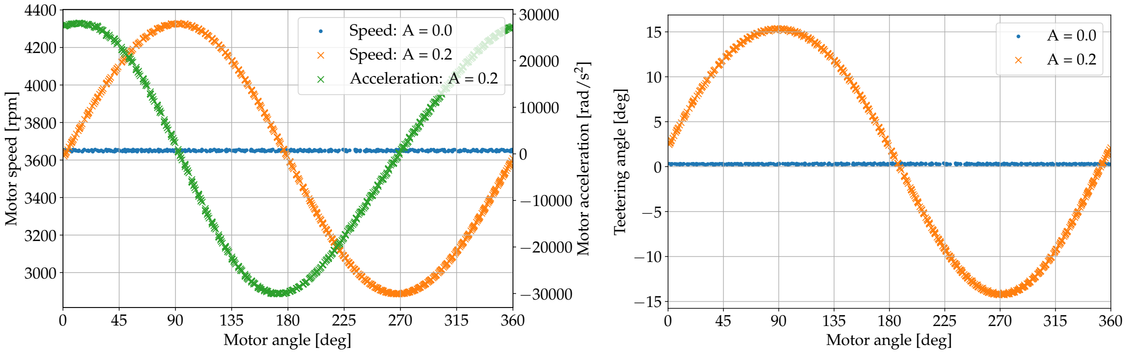
For the real system, only the motor angle and velocity are measured, not the other joint angles. The motor angle together with the six-axis FT data are available. In the analysis of the other joint angles, only simulated data can be used. This means that the simulation can only be validated in terms of motor angle ψ and the force and torque produced. There is a strong coupling between the joint angles and the resulting forces and torques. It could be argued that showing correspondence between parts of the response in both the simulated and real system would strengthen the confidence in all joint angle responses in the simulation.

The main experiments conducted for the thrust vectoring rotor system were performed with the calibration routine described in Section 3.2.6. Each combination of motor commands is set constant for a time period, such that the average response can be calculated. Similar motor commands were used as input to the numerical simulation of the rotor system. For all these experiments, the harmonic speed control law from Section 3.2.3 is used, for the following motor commands:

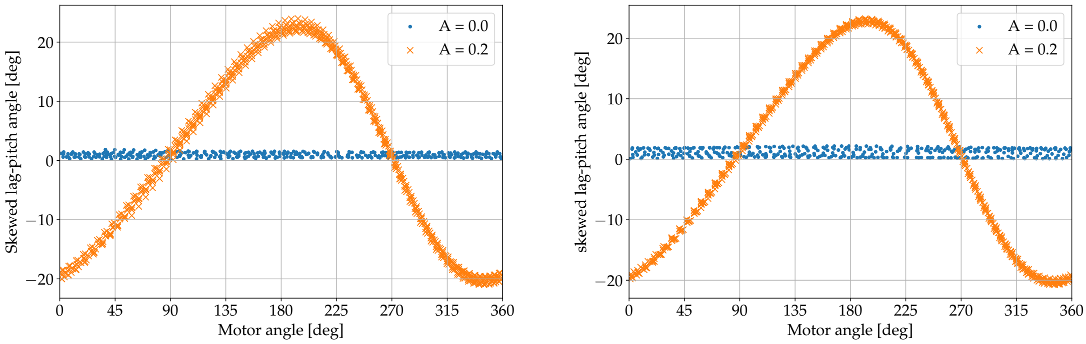
- Average motor speed from $\Omega = 2400$ rpm to $\Omega = 5400$ rpm in steps of 600 rpm.
- Sinusoidal amplitude from $A = 0.0$ to $A = 0.35$ for the measured rotor, and from $A = 0.0$ to $A = 0.40$ in simulation.
- Sinusoidal phase offset $\psi_k = 0$.

4.1.1 Simulated Rotor Dynamics

Simulated per motor revolution response of joint angles for the rotor system is shown in Figure 4.1, with the low angle approximation BEMT method. Additionally, per motor revolution motor speed and acceleration are shown. Data for multiple revolutions are plotted. Responses are simulated with the harmonic speed control law at sinusoidal amplitude $A = 0.2$, and with $A = 0.0$, both with average speed $\Omega = 3600$ rpm.



(a) Motor Speed and Acceleration. **(b)** Teetering angle.



(c) Positive blade skewed lag-pitch angle. **(d)** Negative blade skewed lag-pitch angle.

Figure 4.1: Simulated motor speed, acceleration and hinge angles over motor angle at steady state operation. BEMT with small angles approximation is used. Two sinusoidal speed control commands with amplitude A and phase zero are shown, with average motor speed $\Omega = 3600$ rpm.

Teetering Response

The response of the teetering joint can be seen in Figure 4.1(b). For the control input with a sinusoidal amplitude of $A = 0.2$, a sinusoidal response that is phase-locked to the motor angle is observed. The maximum teetering occurs when the motor angle is $90°$. Looking at the motor speed and acceleration over motor angle in Figure 4.1(a), the teetering and motor speed show in-phase sinusoidal responses. Considering the harmonic speed control law in Equation (3.19), the maximum motor speed command occurs at motor angle $90°$, when the sinusoidal phase offset is zero. Thus, the motor speed response shown in Figure 4.1(a) does not lag the reference command noticeably. Since the velocity lags the acceleration by $90°$ due to the derivative, the teetering response lags the acceleration by $90°$, which can be seen in Figure 4.1.

This teetering response can be compared to the teetering response of a conventional helicopter rotor, where blade pitch is controlled with a swashplate. In a perfectly hinged teetering rotor system, the teetering response lags the blade pitch input by $90°$ [19]. Similarly for this system, the teetering response lags the acceleration by $90°$. Motor acceleration, through motor torque, can then be seen as the input driving the sinusoidal teetering response through the lag-pitch kinematic coupling. The same argument is given in Section 1.1.

Lag-pitch Response

The response of the positive lag-pitch joint can be seen in Figure 4.1(c), and the response of the negative lag-pitch joint can be seen in Figure 4.1(d). They show in-phase sinusoidal responses. This is expected, as blade leading and blade lagging cause equal angles due to the lag-pitch joint definitions in Figure 3.1. Since there is a positive kinematic coupling between the blade pitch and the lag-pitch joint for the positive lag-pitch joint, and a negative coupling for the negative lag-pitch joint, the blade pitch responses for the two blades are $180°$ out of phase with each other. Another observation that can be made about the lag-pitch response is that the amplitude, and thus the maximum angles, are higher than those of the teetering joint for this specific sinusoidal input. This is important for the thrust vector rotor hub design, the geometry has to account for the maximum angles to be collision-free. For the design in Section 3.2.1, $90°$ maximum lag-pitch angles were taken into account, as the maximum possible lag-pitch angle was not known at design time.

Dynamics at Zero Sinusoidal Amplitude

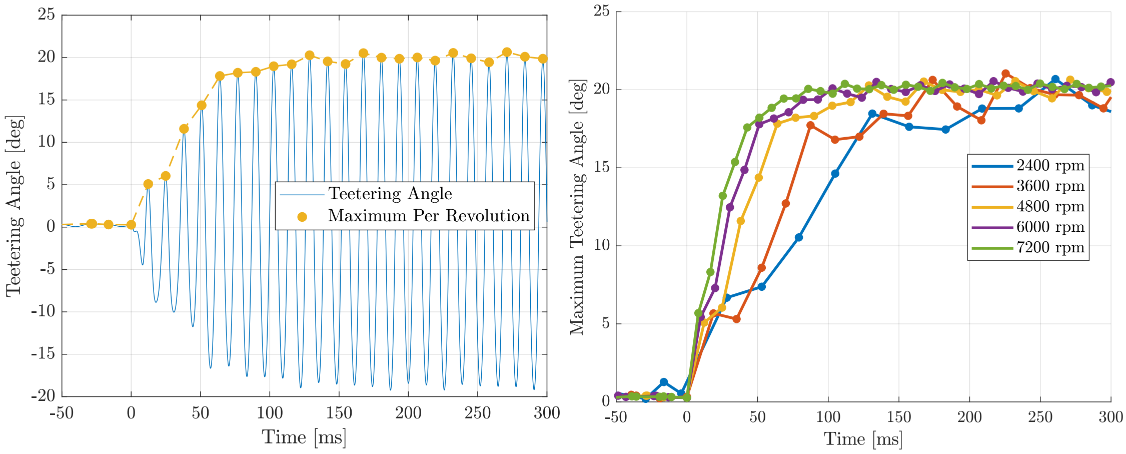
The simulated response without a sinusoidal speed component, $A = 0$, is shown in Figure 4.1. This corresponds to a constant motor speed reference. It can be seen that there is a small positive angle for the teetering and lag-pitch joints over all motor angles. The positive angles of the lag-pitch joints can be explained by the lag of both joints due to the drag force of the rotor blade. Since both lag-pitch joints have a positive angle, the positive side has an increased pitch, while the negative side has a decreased pitch. This can explain the positive teetering angle over all motor angles. In effect, this could cause a small, motor frequency phase-locked oscillation of the thrust vector at this constant speed reference.

Input Step Response

A simulated step response in sinusoidal amplitude A is shown in Figure 4.2. The sinusoidal amplitude is increased from $A = 0$, to $A = 0.26$ at $t = 0$. $A = 0.26$ is chosen as it was found to result in a steady-state maximum teetering angle of about $20°$, which can be compared with other systems. The simulations were performed with the low-angle approximation BEMT method. The step in sinusoidal amplitude in the harmonic speed control law occurs at time $t = 0$. The resulting response in teetering angle is shown in Figure 4.2(c) for average motor speed 4800 rpm. The teetering angle starts to oscillate after $t = 0$, and this oscillation reaches a steady maximum of $20°$ at around $t = 130$ ms. This oscillation is phase-locked to the motor angle similar to Figure 4.1(b). The maximum teetering angle per motor revolution is further indicated in Figure 4.2(c) by the dots. The propulsion disk tilting effect of a motor angle phase-locked teetering response is illustrated by Figure 4.2(a) and Figure 4.2(b). At Figure 4.2(a), the teetering angle is close to constant zero, such that the propulsion disk does not tilt. At Figure 4.2(b), the teetering angle has a maximum of around 20°, which corresponds to a propulsion disk elevation angle of 20°.

The step response simulated for multiple average rotor speeds Ω, from $\Omega = 2400$ to $\Omega = 7200$, is shown in Figure 4.2(d). There are two interesting observations. First, the same maximum teetering angle of 20° is reached for all the average motor speeds simulated. Second, as the average motor speed increases, the rise time to reach the maximum teetering angle appears to decrease. Equivalently, as the average motor speed increases, the dynamics of the propulsion disk elevation becomes faster.

(a) Simulation at $t < 0$ **(b)** Simulation at $t >> 0$



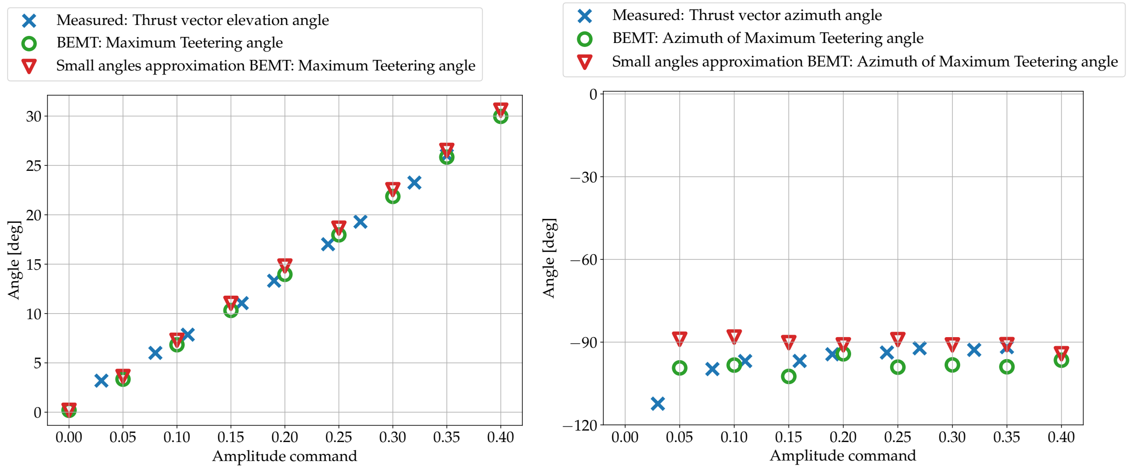
(c) Simulated step response at average speed $\Omega = 4800$ rpm

(d) Simulated step response at different average speeds Ω

Figure 4.2: Rotor maximum teetering angle step responses simulated with the small angles approximation BEMT method. Input sinusoidal amplitude A is stepped from $A = 0$ to $A = 0.26$ at $t = 0$. A is chosen to cause approximately 20 degree steady state maximum teetering angle. (c) shows how the maximum teetering angle per revolution has been found from the time response for $\Omega = 4800$ rpm. (d) shows the same methodology with the time response of maximum teetering angle per revolution for multiple different Ω. (a) and (b) shows frame blended recordings of the simulated response and visualizes the step response effect of a 20° rotor disk tilt. A video of the step response for $\Omega = 4800$ rpm slowed down 100 times is available here: `https://youtu.be/HBLV1LE9DL0?t=79`

4.1.2 Simulated and Measured Thrust Vectoring

The thrust force vector given a specific motor input will now be compared between simulation and measurement. The definition of the thrust force azimuth and elevation angle are given in Section 3.2.2.



(a) Thrust Vector Elevation angle **(b)** Thrust Vector Azimuth angle

Figure 4.3: Comparison of thrust vectoring between simulation and real measured rotor tests. All responses are shown at average motor speed $\Omega = 3600$ rpm. In simulations, maximum teetering angle and azimuth of maximum teetering angle are calculated as averages at steady state. This is compared to the measured thrust vector force elevation and azimuth angle of the real rotor. Small angles approximation BEMT refers to the method in Section 2.1.4, while BEMT refers to the method in Section 2.1.5

Azimuth and Elevation Angle

Figure 4.3 compares thrust force vector averaged angles for the simulated and real system when increasing the sinusoidal motor amplitude. All results shown correspond to a single average rotor speed of $\Omega = 3600$ rpm. For the real system, the thrust vector elevation and azimuth angles are averaged from force data. For the simulated system, the thrust vector angles are given by averaged per revolution joint angles. This argument is used when modeling aerodynamics in Section 3.1.2, and is repeated here. With the assumption of the blades moving in a steady state propulsion disk, the maximum per revolution teetering angle defines the elevation, while the azimuth is given by the azimuth angle of maximum teetering.

Both the simulation and the measurements show a linear trend between the elevation angle of the thrust vector and the amplitude command in Figure 4.3. The simulation predicts the elevation angle of the measured rotor well. For the azimuth angle, both simulation and measurements show a negative phase lag between the defined zero azimuth and the thrust vectoring azimuth. Since the rotor rotates CCW in both simulation and measurements in this experiment, the thrust vectoring azimuth decreases as the motor angle increases. Thus, this lag is equivalent to the 90° lag seen in Figure 4.1(b). There is more discrepancy between the measured and simulated data here, especially at low amplitude commands. This could be partly due to measurement error, as the accuracy of the force measurements are quite low as explained in Section 3.2.6. The azimuth angle is calculated as $arctan2(F_y, F_x)$, and for small amplitudes A, the x- and y-components of the thrust force are small and will have a lower signal to noise ratio.

Force and Drag Induced Moment

Simulated and measured force and torque are shown in Figure 4.4. The simulated force and drag induced torque contains only the aerodynamic forces from Section 3.1.2, neglecting other forces apparent from the equations of motion in Equation (3.2). The figure illustrates measurements from two slightly different rotors. The rotor labeled ”inverted” is the lower rotor in the coaxial rotorcraft, which is different from the rotor analyzed in simulation and measurements so far. The blades of the ’inverted’ rotor are flipped due to the entire rotor assembly being mounted in an inverted position, which can be observed for the lower rotor in Figure 3.16.

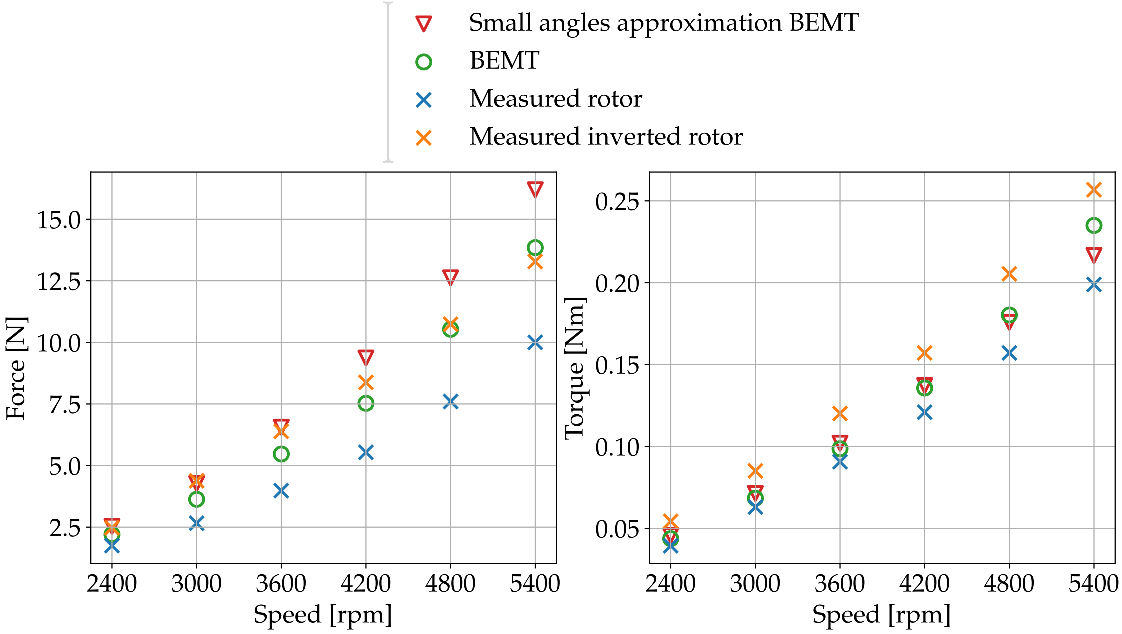


Figure 4.4: Force and torque of the simulated and measured rotor. Small angles approximation BEMT refers to the method in Section 2.1.4, while BEMT refers to the method in Section 2.1.5

There is a large measured difference in force and drag induced torque between the upper and the lower inverted rotor. The inverted rotor produces more thrust force and more drag induced torque than the upper rotor. The measurement process differs slightly between these rotors. The rotor depicted in section 3.2.6 directs its airflow towards the ground, while the airflow from the inverted rotor is directed upwards. This might not explain this difference, as the same effect of more thrust force for the lower rotor were seen in flight.

The model seems to over-estimate the amount of thrust force, especially at higher rotor speeds. The drag induced torque is close to the mean of the upper and lower rotor for all rotor speeds evaluated. In general, the simulation does not match as well with the measurements as in the case with the thrust vector angles. In calculating the aerodynamic forces, a significant number of assumptions and simplifications are made. The model of the blade geometry shown in Appendix A.2 also has clear limitations. In comparison, the forward dynamics carried out during the simulation is calculated to numerical precision, with minimal assumptions and simplifications in the model. This could be crucial for accurate joint angle simulations.

4.1.3 Harmonic Torque Feedforward Modulation

The harmonic torque control law from Section 3.2.3 was implemented on the custom motor driver and is evaluated with one experiment here. Note that this control law is only used in this experiment, while all other experiments use the harmonic speed control law. Figure 4.5 shows the elevation response for several average rotor speeds and sinusoidal torque feedforward amplitudes. In contrast to the harmonic speed modulation law, the amplitude is here given directly as the torque amplitude, and not as a multiple of average rotor speed. Compared to the harmonic speed control law, the elevation response varies greatly with average rotor speed. A linear model without offset has been fit to each data set corresponding to a specific rotor speed to demonstrate the approximate linear relationship between thrust vector elevation angle and sinusoidal torque feedforward amplitude. To use this model for rotorcraft flight, a model would need to account for the dependence on average rotor speed. Observing the slopes of the linear models in Figure 4.5, it is evident that this relationship is not linearly dependent on average rotor speed, unlike the case with the harmonic speed control law. Establishing a suitable model would be necessary to evaluate this control law for flight. As discussed in Section 3.2.3, this control law could have advantages in terms of phase delay compared to the harmonic speed control law, but further investigation is necessary to evaluate this control law fully.

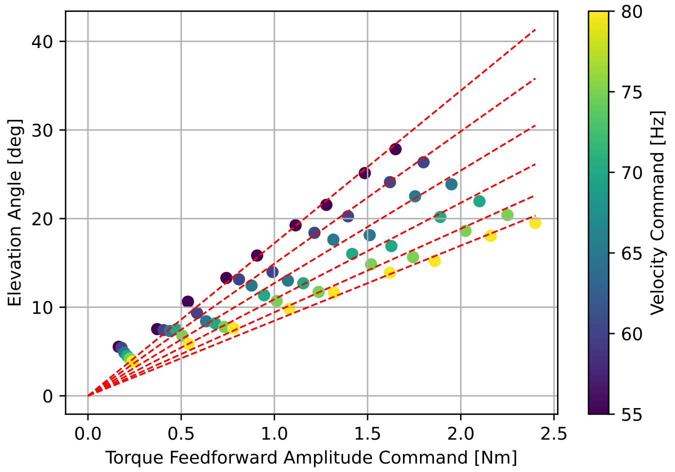


Figure 4.5: Measured thrust vector elevation data with torque feedforward control law. The calibration routine has been run for a set of average rotor speeds and sinusoidal torque feedforward amplitudes.

4.1.4 Thrust Vectoring Calibration and Performance

The thrust vector calibration involves creating a model that relates the motor commands Ω, ψ_k, and A to the magnitude of the thrust force vector, its elevation and azimuth angles, and the magnitude of the drag-induced moment. This model, which is used in the harmonic speed control law, is explained in detail in Section 3.2.3. The calibration method is explained in section 3.2.6. The results from calibration are discussed here, as well as the overall performance of the thrust vectoring rotor.

Elevation Angle

The observed linear trend between thrust vector elevation angle and amplitude command, as seen in Figure 4.3, suggests a linear model for the elevation angle as a function of amplitude command: $\beta_c = k_A A$, as used in Equation (3.20). Calibration data, as well as the final linear constant k_A are shown in Appendix A.3. Here, multiple average rotor speeds are plotted. There is some dependence on rotor speed, but the model of elevation as linear in amplitude command was considered sufficiently accurate. The dependence on rotor speed could be caused by the motor speed PI-controller.

Maximum elevation angles between 21° and 29° were achieved for the maximum sinusoidal amplitude tested, $A = 0.35$. Since this amplitude modulates the speed reference signal, the maximum speed of the motor is important to keep in mind, which is limited by the maximum voltage the motor driver can supply given its supply voltage. In this case, using $A = 0.35$ is only possible when the average motor speed $\Omega \geq \omega_{max}/(1 + 0.35) = 0.74\omega_{max}$, where ω_{max} is the maximum motor speed. During some early experimentation, elevation angles above 30° were achieved, but only for lower rotor speeds. These results could motivate constraints in elevation angles that depend on motor speed, approaching zero as the rotor speed approaches the maximum. Since constrained control allocation is not considered in this thesis, this is not explored further.

Azimuth Angle

The thrust vector azimuth model is proposed in Equation (3.20): $\psi_c = d(\psi_k - \psi_a)$, where ψ_c is the commanded thrust vector angle, and ψ_k is the sinusoidal phase offset sent to the sinusoidal motor control law. ψ_a is a phase offset that was chosen to be 90°. Looking at both Figure 4.3 and Appendix A.3, this offset is not perfectly constant in terms of motor commands. It does however not show an increasing lag with increasing rotor speeds. Such phase lag could be caused by a low bandwidth sinusoidal modulation and time delay. In the associated specialization project, time delay due to a 400 Hz sinusoidal modulation with additional communication delay caused azimuth phase lag, increasing with rotor speed. Here, sinusoidal modulation is performed on the motor driver at 40 kHz, resulting in low time delay compared to the maximum rotor speed tested of 5400 rpm $=$ 90 Hz.

Force and Drag Induced Moment

The calibration results for the force and drag induced moment can be found in Appendix A.3.

4.1.5 Comparison of Simulated BEMT Methods

Simulation of the rotor was performed with both BEMT methods described, the small angles approximations method from Section 2.1.4, and the large angles method from Section 2.1.5. In terms of simulated joint angles seen in Figure 4.3, the methods produce similar results. There is a greater difference between the methods in the aerodynamic force and moment produced, seen in Figure 4.4. Here, the small angles approximations method produces a higher thrust force at a given rotor speed. This could be due to the difference in the tip loss models used. In terms of computation time, the numeric simulation using the small angles approximations method is significantly faster. For the large angles method, the zero of the non-linear inflow function in Equation (2.28) must be computed individually for each blade section, utilizing the bisection method. For the small angles approximation method, the inflow from Equation (2.27) can be evaluated for all blade sections in one efficient vector operation. As the prediction accuracy of both methods was similar, the low angles approximation method was used in the experiments in Figure 4.1 and Figure 4.2.

4.2 Coaxial Rotorcraft

4.2.1 Motor Command Software Performance

The developed software running on the onboard Raspberry Pi is important for flight, as the motor commands from the flight controller enters here. As explained in Section 3.2.5, there is concern about the real-time capabilities of the implementation.

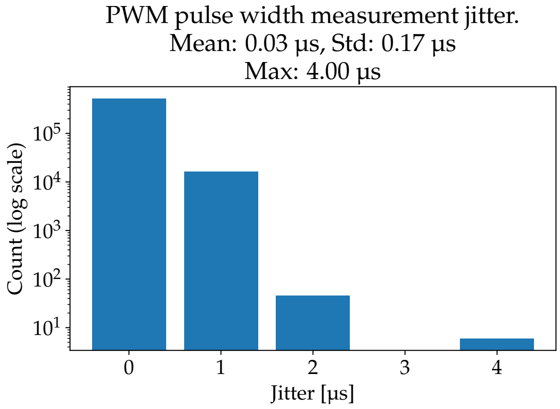


Figure 4.6: PWM measurement performance on the Raspberry Pi while running all systems.

PWM Pulse Width Measurements

Figure 4.6 shows the performance of PWM pulse width measurements. Constant PWM signals with a pulse width of 900 µs were transmitted through the six PWM channels of the flight controller for a time period at a rate of 400 Hz for each channel. The jitter in Figure 4.6 refers to the difference between the PWM pulse widths sent from the flight controller and the pulse widths measured on the Raspberry Pi. It is assumed that the flight controller sends a perfect PWM-signal. While the majority of pulse widths are measured correctly, there is a large amount of jitter of 1 µs. Considering the clock resolution of 1 µs used to measure the pulse width on the Raspberry Pi, this is an expected limitation. As the signal is modulated with pulse widths between 1000 µs and 2000 µs, a 1 µs jitter constitutes to 0.1% of the full-scale range. Importantly, there are no large deviations in measurements that could have occurred if the pulse width measurement were interrupted by the operation system.

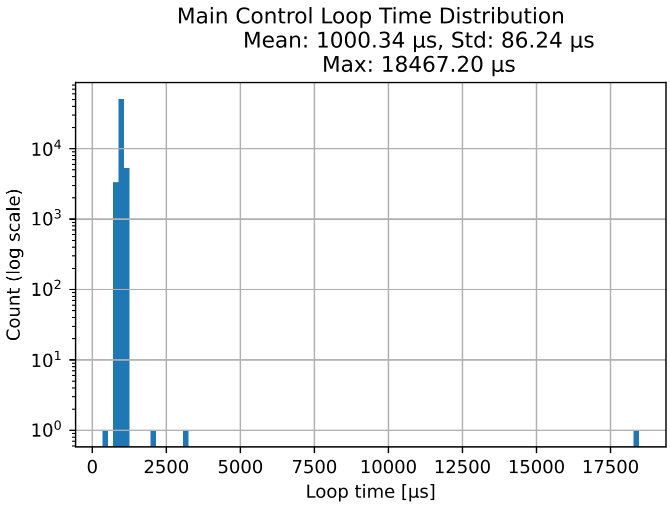


Figure 4.7: Distribution of loop time for the main control loop in the thrust vector control software running on the onboard Raspberry Pi 4 computer. The data is captured from a 100 second flight.

Main Thrust Vector Command

The loop time of the main motor command loop running on the Raspberry Pi was measured during flight. This is the process that takes thrust vector commands from the *PWMReader* class, calculates appropriate motor commands and sends them over CAN-FD. This process is set to run at a rate of 1 kHz. The distribution of measured loop time for this process during flight can be seen in Figure 4.7. The majority of loop times are close to 1 kHz. From this experiment there is especially one measurement showing a large delay in the loop of 18467 $\mu s \approx 0.018$ s. While this single delay did not have any noticeable effect on flight, it raises concerns about the real-time performance and stability of this implementation. Running the thrust vectoring commands through the Raspberry Pi companion computer is possible, but a proper RTOS should be used for critical applications.

4.2.2 Flight

Some IMU filtering effort was required to achieve flight due to the outer cage of the rotorcraft lacking stiffness. This effort is covered in Appendix A.4.

Simple flight tests were performed with the coaxial rotorcraft. This combines several of the developed systems:

- The thrust vectoring rotors, their control and calibration described in Section 3.2.
- The designed coaxial rotorcraft described in Section 3.3
- The control allocation and actuator model described in Section 3.4.

The goal of the flight testing is to demonstrate that these systems work sufficiently for flight. Extensive flight testing and analysis of the control of thrust vectoring rotorcraft are beyond the scope of this thesis. Two main flight characteristics are analyzed, hovering and attitude control.

Table 4.1: Hovering statistics of coaxial rotorcraft.

Hovering Metric	Value
Total weight	1.20 kg
Average power usage	155 W
Efficiency	7.74 g/W
Upper rotor average speed	4600 rpm
Lower rotor average speed	3700 rpm

Figure 4.8: Rotorcraft hovering

Hovering

The main metrics measured from rotorcraft hovering can be seen in Table 4.1. Average power usage was measured from the onboard power module and averaged over a period of 10 seconds. This includes power usage by the two rotors, as well as all the electronics such as the Raspberry Pi computer. The same 10 second period was used to find the average rotor speed for the upper and lower rotor.

The efficiency of this hovering sequence was 7.74 g/W, where the off-the-shelf lithium ion polymer battery was used. With this efficiency and battery, a total flight time of nearly 13 minutes is possible. The custom, higher energy density battery pack is 0.16 kg heavier than the battery used for the hovering test. Assuming the same 7.74 g/W efficiency and a rotorcraft weighing 1.36 kg with the custom battery pack, a total flight time of nearly 38 minutes is possible. Note that the efficiency of hovering will be lower with the heavier rotorcraft, as the rotors would need to operate at higher speeds. These calculations should be confirmed by actual flight tests, but they still provide an indication of the possible flight time of this system.

Efficiency is now compared to that of similar rotorcraft. [7] achieves a hovering efficiency of 6.7 g/W with a single-rotor rotorcraft weighing 1.23 kg, with a similar rotor to this work. The single-rotor rotorcraft is further demonstrated to have an efficiency 17.5% higher than a quadcopter with equal disk area [7]. The coaxial rotorcraft in this work has approximately twice the disk area and demonstrates a 15.5% increase in efficiency compared to the single-rotor rotorcraft. According to momentum theory, discussed in Section 2.1.6, doubling the disk area and keeping the weight equal would suggest a 41% increase in efficiency, significantly higher than seen here. However, momentum theory does not account for losses such as those arising from the induced inflow of the upper rotor entering the lower rotor.

It is apparent that the upper rotor has to operate at a higher average rotor speed compared to the lower rotor at hover. The difference in rotor speeds cannot be explained by the controller, calibration chosen or the control allocation, and has to be inherent difference between the rotors. An argument will be given for this claim. Since the rotorcraft at hover had close to zero rotational velocity about any of its body axes, a net zero moment is produced by the two rotors. The force produced by the two rotors during hover has to be equal in magnitude and opposite in direction to the gravity force vector. This defines all external forces and moments, which there only exists a single solution to in terms of average rotor speed for the two rotors.

Aerodynamic effects could explain a difference in the force produced by the two rotors. The lower rotor operates in the induced airflow of the upper rotor. This does not explain that the same difference in force produced by the two rotors appeared when individually testing the rotors in Figure 4.4.

The difference in rotor speed is not necessarily a problem. Figuring out the cause of this effect would provide more insight into the consequences and potential mitigation that could be performed in the design of the rotor or rotorcraft.

Control

A simple flight was performed using the multicopter position controller of the PX4 autopilot [33]. Here, actuation of rotorcraft body-frame forces is limited to the z-axis, similarly to conventional, under-actuated MAVs as described in Section 3.4.5.

The tracking performance of rotorcraft velocity control, angle control and angular rate control is demonstrated in Figure 4.10. Photos of the rotorcraft in motion during these experiments can be seen in Figure 4.9. The angles and angular rates are displayed in Euler angles. Manual control input was used to generate rotorcraft acceleration setpoints. For pitch and roll angle, setpoints of up to $\pm 30°$ and $\pm 20°$ were tracked, respectively. The yaw control results show that yaw angular rates of up to $100°$/s can be tracked with only minor deviations in the vertical velocity of the rotorcraft.

With this limited experiment, it is not possible to quantify the limits of performance of this rotorcraft, while it shows a basic level of performance. The system is not pushed to its limits, both in terms of maximum thrust vectoring angles, as well as controller tuning. The maximum thrust vector elevation angle never exceeded $10°$ for any of the rotors in this experiment.

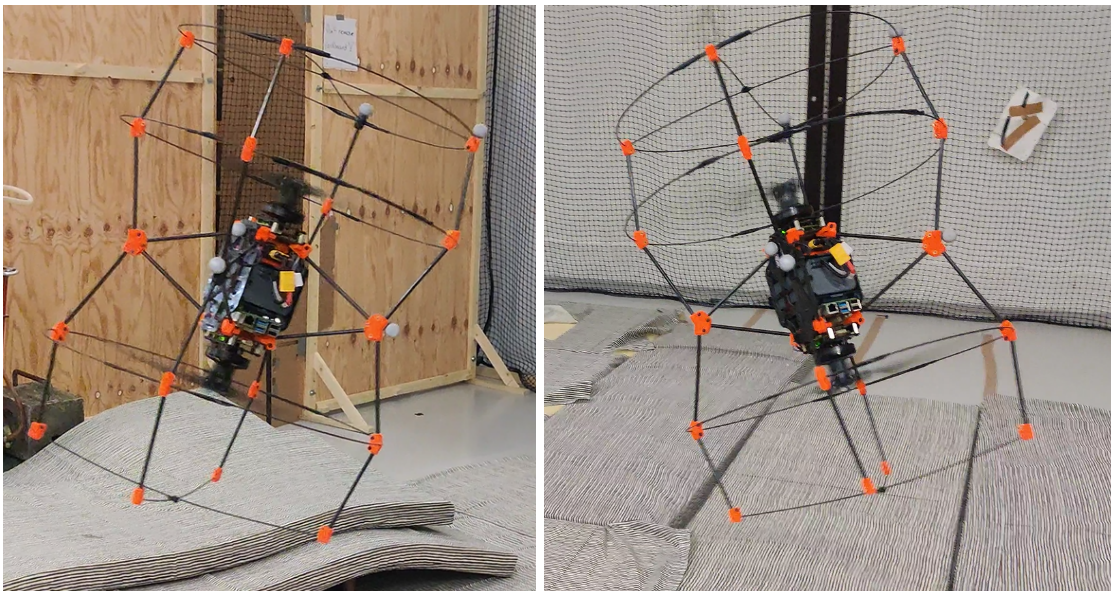


Figure 4.9: Rotorcraft during attitude control experiment. The rotorcraft is moving laterally.

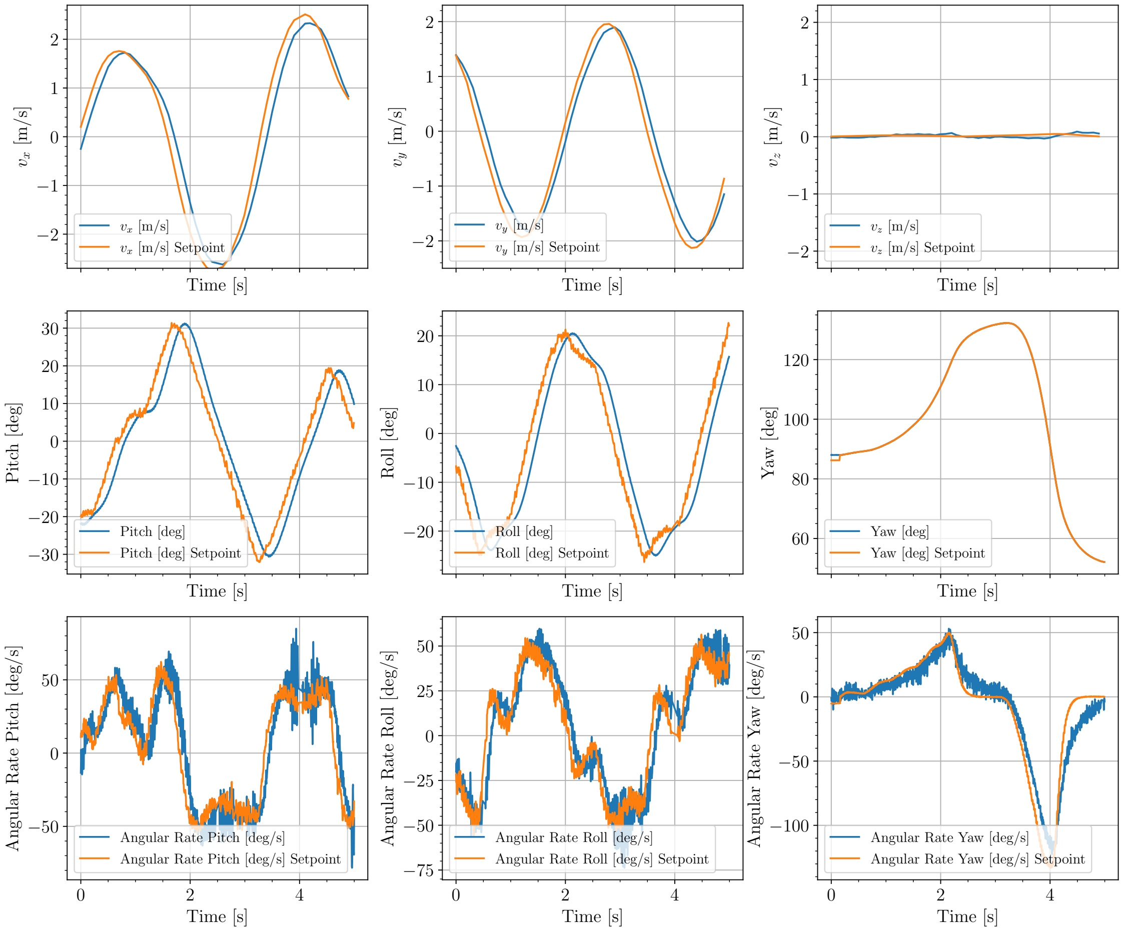


Figure 4.10: Experiment with large manual control setpoints for coaxial rotorcraft in flight. Three distinct time segments are shown column wise. World-frame velocities and velocity setpoints have been rotated by the heading direction of the rotorcraft, and are shown as v_x, v_y, v_z. A video of this flight can be found here: `https://youtu.be/HBLV1LE9DL0?t=48`

4.3 Future Work

Only the single rotor model was simulated in this work. The full rotor and rotorcraft model should be simulated and evaluated for its ability to predict rotorcraft flight dynamics.

More computationally efficient modeling efforts could be explored and should be benchmarked against the predictive ability of the model provided here.

Only basic flight without actuating the lateral body-frame forces was shown. While the initial results in attitude control are promising, more testing should be performed. With maximum thrust vectoring angles of more than 25 degrees shown on a test rig, testing the limits of rotorcraft control with this performance would show how performant this system can be. Additional flight and test rig experimentation could be performed on the achievable control bandwidth possible due to the dynamics of the thrust vectoring, where only simulation results have been analyzed in this thesis.

For the control allocation model developed, considering and implementing a constrained control allocation method is necessary to handle operation at limits. This should be analyzed with the specific use of the thrust vectoring rotors studied here.

While the FOC approach and the implementation of the motor driver performed well, there are many areas that could be further researched here. This system is integral in thrust vectoring performance and needs both optimal tracking performance and smooth operation. It is suggested to look into the following for improvements in the thrust vectoring implementation investigated in this thesis:

- Compare voltage control as seen in previous implementations[5], [6], to current control with FOC used in this work with the same rotor.
- Implement speed dependent rotor drag as torque feedforward.
- Investigate smoother and higher performance torque control tracking. This could include improving current controller bandwidth and noise, as well as further investigating the proposed sinusoidal torque feedforward control law.
- Implement a CAN-FD driver for the thrust vectoring firmware on the flight controller. This allows elimination of the companion computer in the low level actuator control, reducing complexity, hardware reliance and assures real-time capability.

5

Conclusion

The work presented in this thesis has outlined a comprehensive and in-depth investigation into the modeling, control, and implementation of a torque modulated thrust vectoring rotor.

The flight of the coaxial rotorcraft has demonstrated the ability of all the systems developed. A rotorcraft with only two actuators handles attitude control and can hover with great efficiency. Coupled with the custom, high energy density battery designed for the rotorcraft, long endurance flight in the range of 30 minutes is possible with this platform.

The designed and produced thrust vectoring rotors were demonstrated with control of the thrust vector elevation angle up to $30°$. Achieving this requires extra care in the geometry and strength of the design, as well as the motor control performance. It is seen that elevation angle remains well approximated by a model linear in terms of harmonic speed amplitude, even for high angles.

The model and numerical simulation developed demonstrated great ability in predicting thrust vectoring direction. Larger deviations between simulation and measurements were seen in predicting the aerodynamic forces, which could be improved by defining the blade geometry more accurately.

High performance software and firmware implementation was attempted in all parts of this project. In terms of thrust vector control, this objective was achieved, with 40 kHz sinusoidal torque modulation running on custom motor driver firmware. Sending commands between the flight controller and motor driver proved to be more challenging. Although the solution was not implemented on a real-time operating system, and thrust vector commands are sent through the Raspberry Pi companion computer, the final results were adequate for flight.

A complete solution to torque modulated thrust vectoring rotors is provided, enabling integration in rotorcraft. This enables further exploration of this rotor system, efficient flight and fully-actuated rotorcraft.

Bibliography

[1] Y. Li, Y. Qin, W. Xu, and F. Zhang, "Modeling, identification, and control of non-minimum phase dynamics of bi-copter UAVs," in *2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, IEEE, Jul. 2020. DOI: `10.1109/aim43001.2020.9158910`.

[2] Q. Wu, D. Tang, Q. Quan, K. Zhu, Y. Wu, and Z. Deng, "A directly coil-actuated cyclic pitch control based rotor: Design, modeling, and analysis," *Aerospace Science and Technology*, vol. 140, p. 108 423, 2023, ISSN: 1270-9638. DOI: `https://doi.org/10.1016/j.ast.2023.108423`.

[3] J. Paulos and M. Yim, "Cyclic blade pitch control without a swashplate for small helicopters," *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 3, pp. 689–700, 2018. DOI: `10.2514/1.G002683`.

[4] J. J. Paulos and M. Yim, "Scalability of cyclic control without blade pitch actuators," in *2018 AIAA Atmospheric Flight Mechanics Conference*. DOI: `10.2514/6.2018-0532`.

[5] J. Paulos, B. Caraher, and M. Yim, "Emulating a fully actuated aerial vehicle using two actuators," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 7011–7016. DOI: `10.1109/ICRA.2018.8462975`.

[6] Y. Qin, N. Chen, Y. Cai, W. Xu, and F. Zhang, "Gemini ii: Design, modeling, and control of a compact yet efficient servoless bi-copter," *IEEE/ASME Transactions on Mechatronics*, pp. 1–12, 2022. DOI: `10.1109/TMECH.2022.3153587`.

[7] N. Chen, F. Kong, W. Xu, *et al.*, "A self-rotating, single-actuated uav with extended sensor field of view for autonomous navigation," *Science Robotics*, vol. 8, no. 76, eade4538, 2023. DOI: `10.1126/scirobotics.ade4538`.

[8] Flybotix, *Flybotix Asio Drone*, 2022. [Online]. Available: `https://flybotix.com/asio-drone/` (visited on 11/28/2022).

[9] Vertiq, *Underactuated system*, 2022. [Online]. Available: `https://www.vertiq.co/underactuated-system` (visited on 11/28/2022).

[10] D. Brescianini and R. D'Andrea, "Design, modeling and control of an omni-directional aerial vehicle," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 3261–3266. DOI: `10.1109/ICRA.2016.7487497`.

[11] M. Kamel, S. Verling, O. Elkhatib, *et al.*, “The voliro omniorientational hexacopter: An agile and maneuverable tiltable-rotor aerial vehicle,” *IEEE Robotics & Automation Magazine*, vol. 25, no. 4, pp. 34–44, 2018. DOI: `10.1109/MRA.2018.2866758`.

[12] X. Xu, K. Watanabe, and I. Nagai, “Feedback linearization control for a tandem rotor uav robot equipped with two 2-dof tiltable coaxial-rotors,” *Artificial Life and Robotics*, vol. 26, pp. 259–268, 2020.

[13] V. L. Salvatore Marcellini Jonathan Cacace, “A PX4 Integrated Framework for Modeling and Controlling Multicopters with Tiltable Rotors,” pp. 1–8, 2023.

[14] P. Zheng, X. Tan, B. B. Kocer, E. Yang, and M. Kovac, “Tiltdrone: A fully-actuated tilting quadrotor platform,” *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6845–6852, 2020. DOI: `10.1109/LRA.2020.3010460`.

[15] J. Buzzatto and M. Liarokapis, “The Omnirotor Platform: A Versatile, Multi-Modal, Coaxial, All-Terrain Vehicle,” *IEEE Access*, vol. 11, pp. 27 928–27 941, Jan. 1, 2023, published.

[16] Voliro, *Voliro T*, 2023. [Online]. Available: `https://voliro.com/#product` (visited on 05/20/2023).

[17] Skygauge, *The Skygauge*, 2023. [Online]. Available: `https://www.skygauge.co/the-skygauge` (visited on 06/11/2023).

[18] G. J. Leishman, *Principles of Helicopter Aerodynamics*, 2nd ed. Cambridge, England: Cambridge University Press, Dec. 2016.

[19] S. Leutenegger, C. Hürzeler, A. K. Stowers, *et al.*, “Flying robots,” in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds., Springer International Publishing, 2016, ch. 26. DOI: `10.1007/978-3-319-32552-1`.

[20] O. Egeland and J. T. Gravdahl, *Modeling and simulation for automatic control.* Marine Cybernetics Trondheim, Norway, 2002.

[21] S. Gudmundsson, “Chapter 9 - the anatomy of the wing,” in *General Aviation Aircraft Design*, S. Gudmundsson, Ed., Boston: Butterworth-Heinemann, 2014, pp. 299–399, ISBN: 978-0-12-397308-5. DOI: `10.1016/B978-0-12-397308-5.00009-X`.

[22] C. W. Stahlhut and J. G. Leishman, “Aerodynamic design optimization of proprotors for convertible-rotor concepts,” *AHS International*, 2012.

[23] Y. Qin, W. Xu, A. Lee, and F. Zhang, “Gemini: A Compact Yet Efficient Bi-Copter UAV for Indoor Applications,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3213–3220, 2020. DOI: `10.1109/LRA.2020.2974718`.

[24] K. Waldron and J. Schmiedeler, “Kinematics,” in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds., Springer International Publishing, 2016, ch. 1. DOI: `10.1007/978-3-319-32552-1`.

[25] R. Featherstone and D. E. Orin, “Dynamics,” in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds., Springer International Publishing, 2016, ch. 1. DOI: `10.1007/978-3-319-32552-1`.

[26] W. Leonhard, *Control of Electrical Drives*. Springer Berlin Heidelberg, 2001. DOI: `10.1007/978-3-642-56649-3`.

[27] A. Bosso, C. Conficoni, D. Raggini, and A. Tilli, “A computational-effective field-oriented control strategy for accurate and efficient electric propulsion of unmanned aerial vehicles,” *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 3, pp. 1501–1511, 2021. DOI: `10.1109/TMECH.2020.3022379`.

[28] C. J. O’Rourke, M. M. Qasim, M. R. Overlin, and J. L. Kirtley, “A geometric interpretation of reference frames and transformations: Dq0, clarke, and park,” *IEEE Transactions on Energy Conversion*, vol. 34, no. 4, pp. 2070–2083, 2019. DOI: `10.1109/TEC.2019.2941175`.

[29] T. M. Inc., *MATLAB version: 9.12.0 (R2022a)*, Natick, Massachusetts, United States, 2022. [Online]. Available: `https://www.mathworks.com`.

[30] T. M. Inc., *Robotics System Toolbox version: 4.0 (R2022a)*, Natick, Massachusetts, United States, 2022. [Online]. Available: `https://www.mathworks.com`.

[31] T. Kitamura, *Fusion2urdf*, `https://github.com/syuntoku14/fusion2urdf`, 2020.

[32] T. Lambert, *Rotare, A Matlab BEMT implementation. Version 0.1.2*, 2023. [Online]. Available: `https://gitlab.uliege.be/rotare/rotare`.

[33] L. Meier, D. Honegger, and M. Pollefeys, “PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 6235–6240. DOI: `10.1109/ICRA.2015.7140074`.

[34] H. Fairhead, *Raspberry Pi IoT In C*. I/O Press, Oct. 2020.

[35] P. Virtanen, R. Gommers, T. E. Oliphant, *et al.*, “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python,” *Nature Methods*, vol. 17, pp. 261–272, 2020. DOI: `10.1038/s41592-019-0686-2`.

[36] T. A. Johansen and T. I. Fossen, “Control allocation—A survey,” *Automatica*, vol. 49, no. 5, pp. 1087–1103, 2013, ISSN: 0005-1098. DOI: `https://doi.org/10.1016/j.automatica.2013.01.035`.

[37] R. Rashad, J. Goerres, R. Aarts, J. B. C. Engelen, and S. Stramigioli, “Fully actuated multirotor uavs: A literature review,” *IEEE Robotics & Automation Magazine*, vol. 27, no. 3, pp. 97–107, 2020. DOI: `10.1109/MRA.2019.2955964`.

[38] M. Selig, *UIUC Airfoil Data Site*. Department of Aeronautical and Astronautical Engineering University of Illinois at Urbana-Champaign, 1996.

[39] M. Drela, “XFOIL: An Analysis and Design System for Low Reynolds Number Airfoils,” in *Low Reynolds Number Aerodynamics*, T. J. Mueller, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 1989, pp. 1–12, ISBN: 978-3-642-84010-4.

A

Appendix

A.1 Blade Section Velocity from Joint Space Parameters

The blade induced inflow $\mathbf{v}_i$ at radial position r needs to be calculated for each blade. The two blade bodies blade bodies are given as B_b and frames $\mathcal{B}_b$ with $b = \{3, 4\}$.

The first step is to find the linear and angular velocity of the blade body in Cartesian Space. This will be the velocity of origin $\mathcal{B}_b$. The relationship between joint velocities in the Joint Space and the corresponding velocities in Cartesian Space for any body in the kinematic chain is given by the respective geometric Jacobian. In this case, the geometric Jacobian is $\boldsymbol{J}_{B_b}$.

The Cartesian Space velocities in base frame $\mathcal{B}_0$ for blade b can be computed from the Jacobian $\boldsymbol{J}_{B_b}$.

$$\begin{bmatrix} \mathbf{v}_{B_b}^{\mathcal{B}_0} \\ \boldsymbol{\omega}_{B_b}^{\mathcal{B}_0} \end{bmatrix} = \boldsymbol{J}_{B_b}(\mathbf{q})\dot{\mathbf{q}}, \tag{A.1}$$

where $\begin{bmatrix} \mathbf{v}_{B_b}^{\mathcal{B}_0} & \boldsymbol{\omega}_{B_b}^{\mathcal{B}_0} \end{bmatrix}^T$ are the linear and angular velocities for the positive blade body B_b in base frame $\mathcal{B}_0$. To express these velocities in $\mathcal{B}_b$, the rotation matrix from base frame to $\mathcal{B}_b$ is used.

$$\begin{bmatrix} \mathbf{v}_{B_b}^{\mathcal{B}_b} \\ \boldsymbol{\omega}_{B_b}^{\mathcal{B}_b} \end{bmatrix} = \begin{bmatrix} \mathcal{R}_{\mathcal{B}_0}^{\mathcal{B}_b} \mathbf{v}_{B_b}^{\mathcal{B}_0} \\ \mathcal{R}_{\mathcal{B}_0}^{\mathcal{B}_b} \boldsymbol{\omega}_{B_b}^{\mathcal{B}_0} \end{bmatrix} \tag{A.2}$$

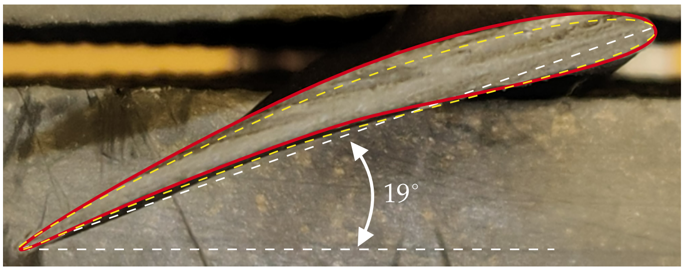
With the blade frame velocities, the local linear velocity of a blade section at radial position r, $\mathbf{v}_i^{\mathcal{B}_b}$, can be expressed. The blade is assumed rigid, such that there are no bending induced velocities. An additional term appears from the angular rotation induced velocity.

$$\mathbf{v}_i^{\mathcal{B}_b} = \mathbf{v}_{B_b}^{\mathcal{B}_b} + \boldsymbol{\omega}_{B_b}^{\mathcal{B}_b} \times \mathbf{r}, \tag{A.3}$$

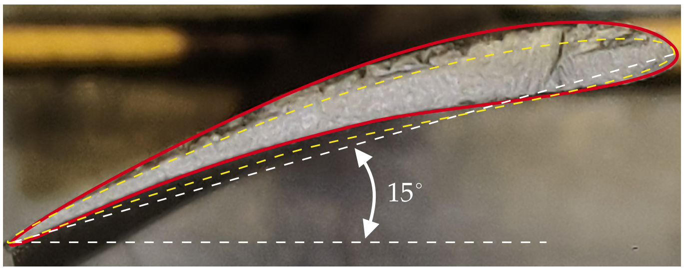
where $\boldsymbol{r} = \begin{bmatrix} r & 0 & 0 \end{bmatrix}^T$. By decomposing the Jacobian into linear and angular velocity components, $\boldsymbol{J}_{B_b} = \begin{bmatrix} \boldsymbol{J}_{B_b,v} & \boldsymbol{J}_{B_b,\omega} \end{bmatrix}^T$, blade b section velocity at radial position r, due to joint space velocities, can be summarized as

$$\mathbf{v}_i^{\mathcal{B}_b} = \mathcal{R}_{\mathcal{B}_0}^{\mathcal{B}_b}(\boldsymbol{J}_{B_b,v}(\mathbf{q})\dot{\mathbf{q}}) + (\mathcal{R}_{\mathcal{B}_0}^{\mathcal{B}_b}(\boldsymbol{J}_{B_b,\omega}(\mathbf{q})\dot{\mathbf{q}})) \times \mathbf{r}, \tag{A.4}$$

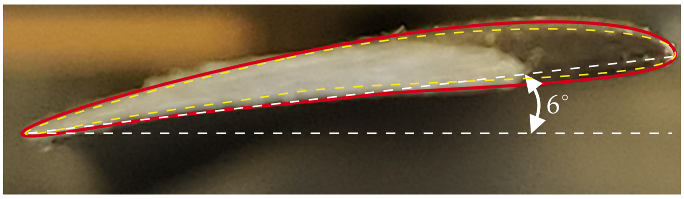
A.2 Modeling MF1302 Propeller Blade



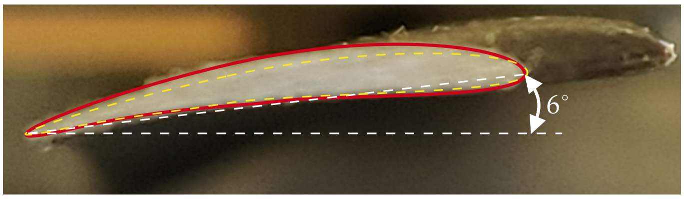
(a) Section 1: $\theta = 19°$, $r_b = 25.0\text{mm}$, $c = 26.0$ mm



(b) Section 2: $\theta = 15°$, $r_b = 62.0\text{mm}$, $c = 23.5$ mm



(c) Section 3: $\theta = 6°$, $r_b = 143.5\text{mm}$, $c = 14.8$ mm



(d) Section 4: $\theta = 6°$, $r_b = 155.0\text{mm}$, $c = 11.0$ mm

Figure A.1: Local pitch angle θ, radial distance r_b and chord length c for four blade sections of the MF1302 blade. The radial distance is measured from the blade mounting hole. The yellow airfoil shape is the Archer A18 airfoil coordinates [38], which is used as an estimate for all sections.

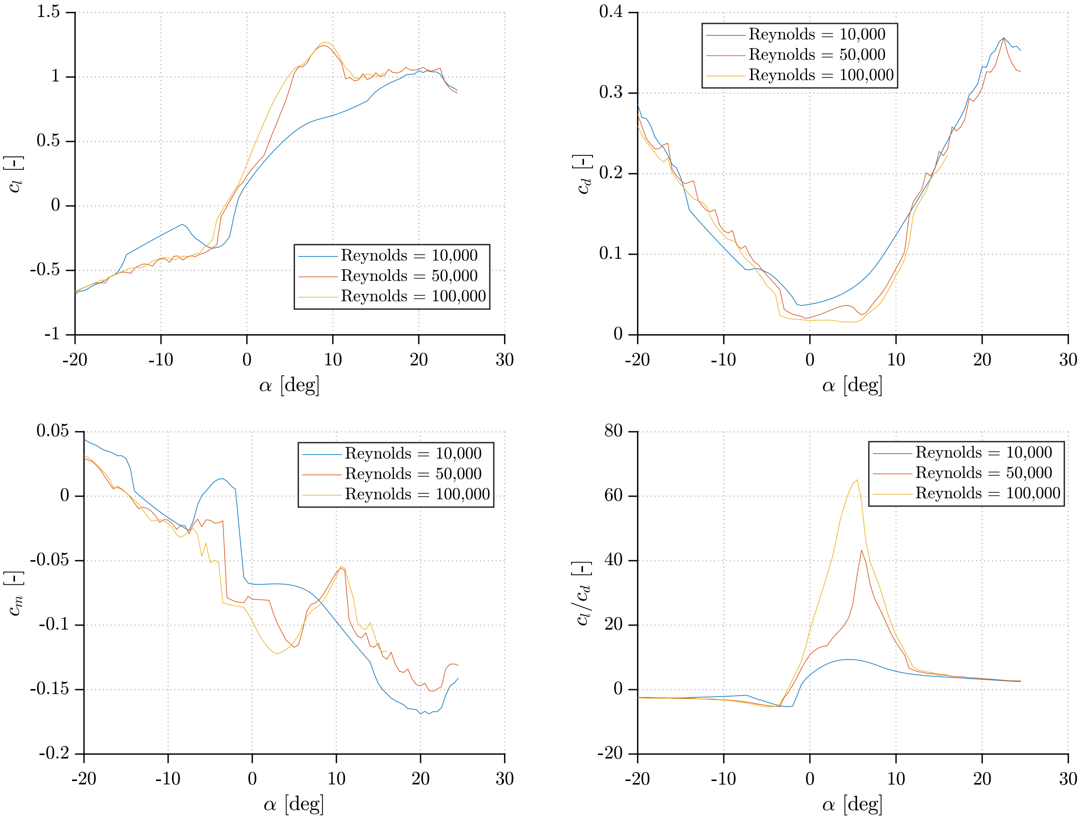


Figure A.2: Simulated lift, drag and moment coefficients for Archer A18 airfoil[38] for some angle of attacks α and Reynolds numbers. Simulated with XFOIL [39] at Mach number equal zero using wrapper from Rotare [32].

A.3 Thrust Vector Calibration Data

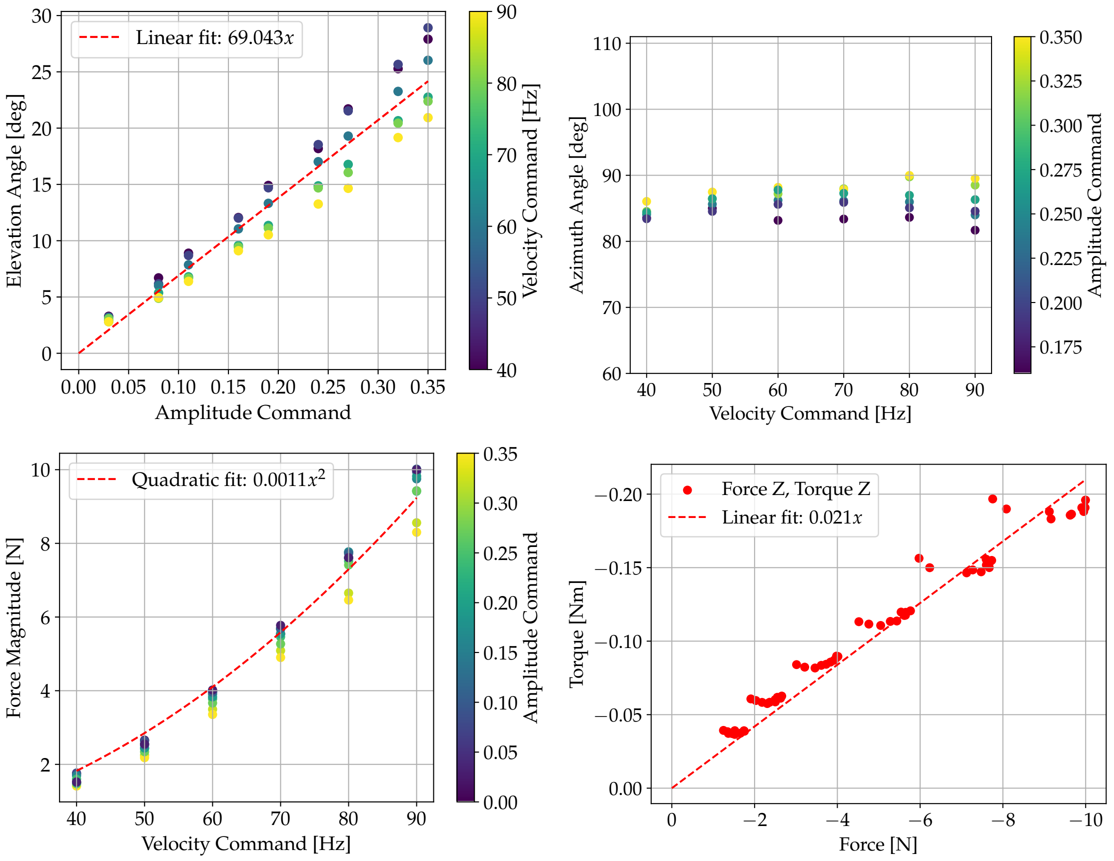


Figure A.3: Calibration data for rotor measured on test stand.

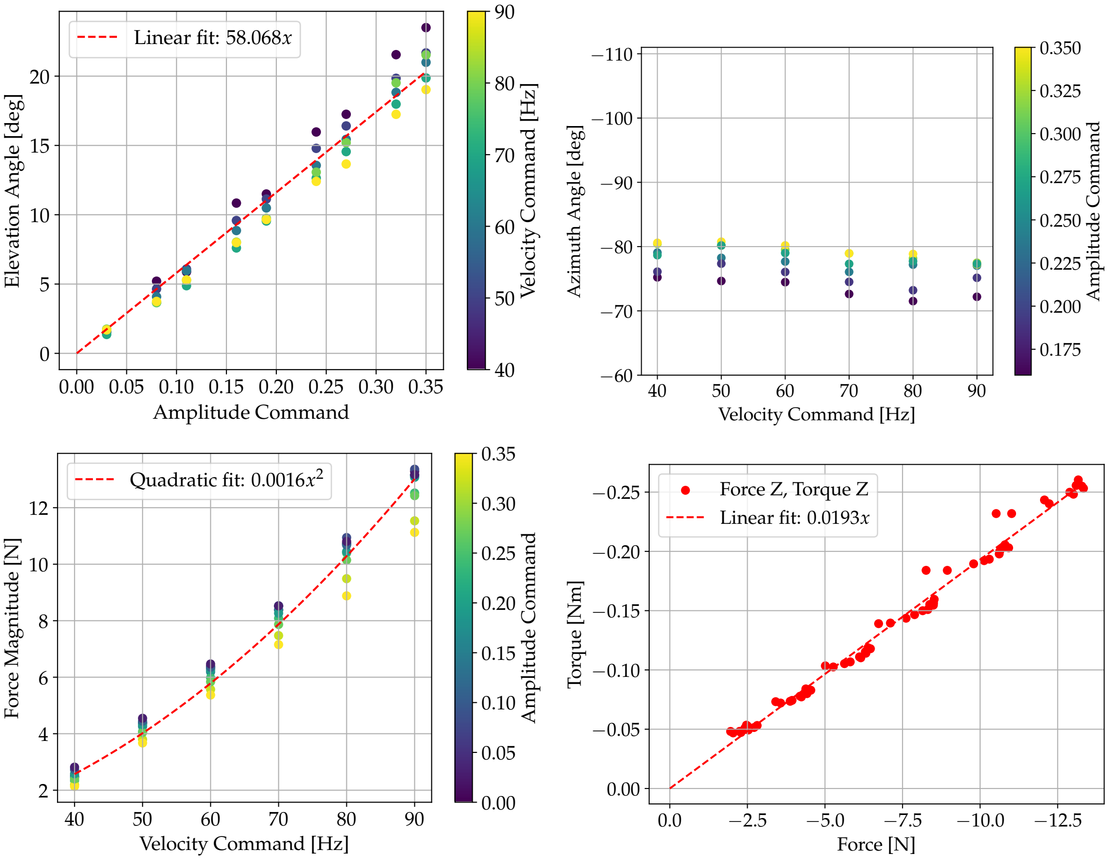


Figure A.4: Calibration data for inverted rotor measured on test stand.

A.4 Filtering Problematic Frame Resonant Frequency for Improved Control

Initial flights of the coaxial rotorcraft were unsuccessful due to large amounts of vibrations, which made controller tuning difficult to impossible. The root cause of this vibration seemed to come from the outer cage, which is too flexible, causing unwanted frequencies in the range of rotorcraft control.

The vibration of this outer cage are likely especially driven by the inherent once per revolution harmonic torque modulation. In the associated specialization project, a large, once per revolution frequency component equal to the average motor speed Ω, with associated higher order harmonics and non-linear modes were observed. In the spectogram of the sum of measured force components in Figure A.5 an experiment were performed by increasing sinusoidal A in steps. It shows that when increasing the sinusoidal amplitude, the general noise level, as well as the frequency components of harmonic frequencies, is increasing. The first hovering flight were possible after applying the in-flight fast Fourier transform routine available in PX4[33], which significantly reduced vibrations. This routine analyzes the frequency components of the IMU gyro data with a fast Fourier transform, detecting the largest frequency components and filters them with a narrow stop band filter, often called notch filter. Analyzing this data, it was apparent that the non-constant average motor frequency Ω were detected and filtered by the routine.

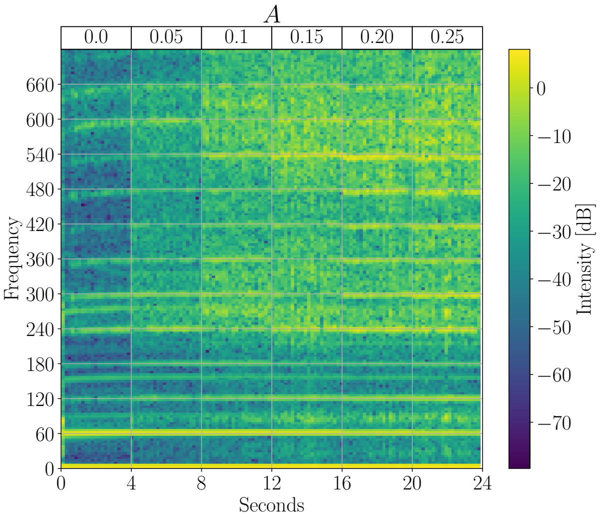
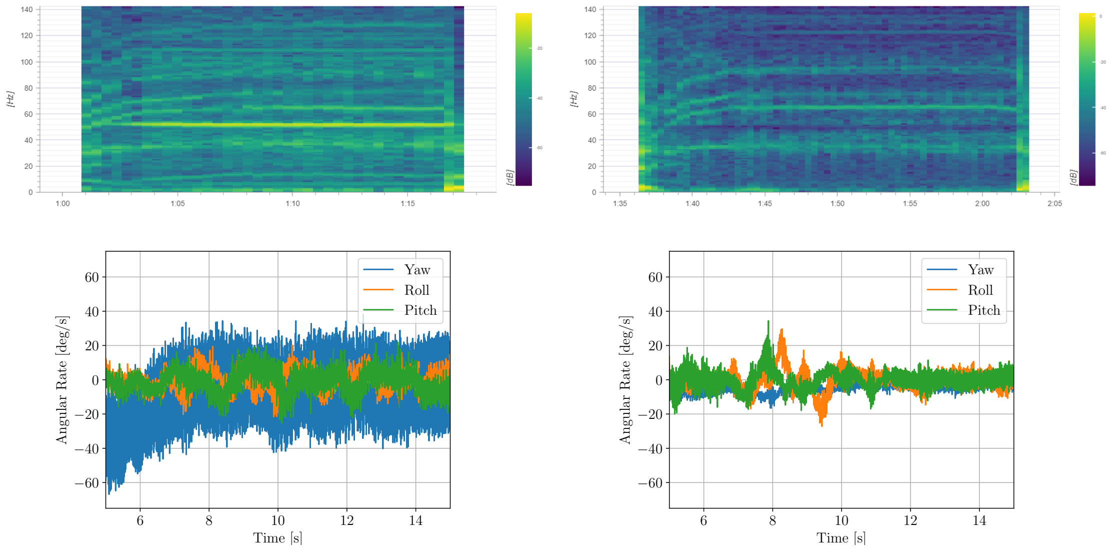


Figure A.5: Spectogram of sum of measured F_x, F_y, F_z at average motor speed Ω = 60 Hz = 3600 rpm while increasing sinusoidal amplitude A. Data from associated specialization project.

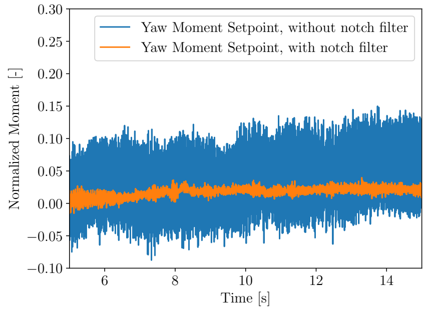
The flight performance were still not satisfactory, with severe and visible vibrations in the outer cage. The vertical cage carbon fiber tubes were seen to vibrate by twisting along the body z-axis (yaw). Figure A.6(a) shows IMU angular velocities, and the spectogram of their sum, during the first hover flight. A large frequency component of around 50 Hz shows up during the whole flight, independent of the non-constant average rotor speeds. The yaw angular rate has especially bad oscillation. The same hovering sequence with a notch filter at 50 Hz is shown in Figure A.6(b). The oscillations in measured yaw angular rate is significantly reduced. Visually observing the rotorcraft, the twisting and vibration of the outer cage were no longer visible by eye. Figure A.6(c) shows that the vibration were amplified by angular rate control in the yaw-axis before adding the notch filter.

Adding this filtering has drawbacks. It can be computationally straining as well as limiting the possible control bandwidth. Addressing the root cause of the issue, lack of frame stiffness, would be a more effective frequency. The vertical carbon fiber tubes are not well supported in the body z-axis twist direction, which could be mitigated by adding more horizontal support structure in the cage. Even with a stiffer frame, it is assumed that there will be unwanted frequency components due to the frame, however these would likely be at a higher, more manageable frequency.



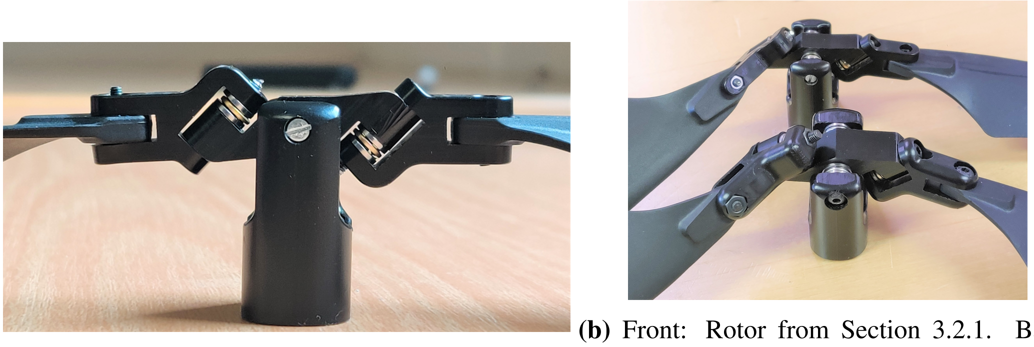
(a) Without notch filter.
Top: Spectogram of sum of angular velocities.
Bottom: Body-frame angular velocities

(b) With notch filter at 50 Hz.
Top: Spectogram of sum of angular velocities.
Bottom: Body-frame angular velocities



(c) Moment setpoint $\tau_{\mathcal{B},z}$ before and after filtering.

Figure A.6: Data from rotorcraft hovering flight, before and after applying a notch filter on IMU gyroscope measurements at 50 Hz.



(a) Smaller rotor.

(b) Front: Rotor from Section 3.2.1. Back: Proposed smaller rotor.

Figure A.7: Overview of the smaller rotor design. Compared to the rotor design from Section 3.2.1

A.5 Smaller Rotor Design

A smaller rotor design were developed, and is presented here. The main goal of this undertaking were to allow a smaller hub size, reduce inertia and save weight. This could be especially helpful if it were to be used for even smaller rotorcraft than the one designed in this thesis.

The size reduction of this rotor is mostly contributed to smaller thrust and radial bearings, which were discovered during later parts of the project. These bearings have a 2mm inner diameter suitable for M2 screws. The dimensions of the rotor and bearings can be seen in Table A.1. With less geometric space being taken by the bearings and screws, a smaller hub design is possible. The smaller hardware also contributes to a lower weight. In total, this rotor weighs 33% less than the larger version.

The design seemed to work well under test stand experiments. In attempting flight testing, one of the side hubs as well as the teetering hub broke. The POM material is not strong enough for this small geometry. Additionally there were some flexing in the plastic hubs, causing a larger coning angle. Machining the hubs in aluminium could be a potential mitigation, otherwise a slightly larger version could be designed. [htb!]

Table A.1: Dimensional specification of mechanical parts for the smaller rotor. Bearing dimensions are given in (inner diameter) x (outer diameter) x (height)

Part	Dimension
Rotor radius R	0.183 m
Thrust bearing	2mm x 6mm x 3mm
Ball bearing	2mm x 5mm x 2.3mm
Bolts	M2

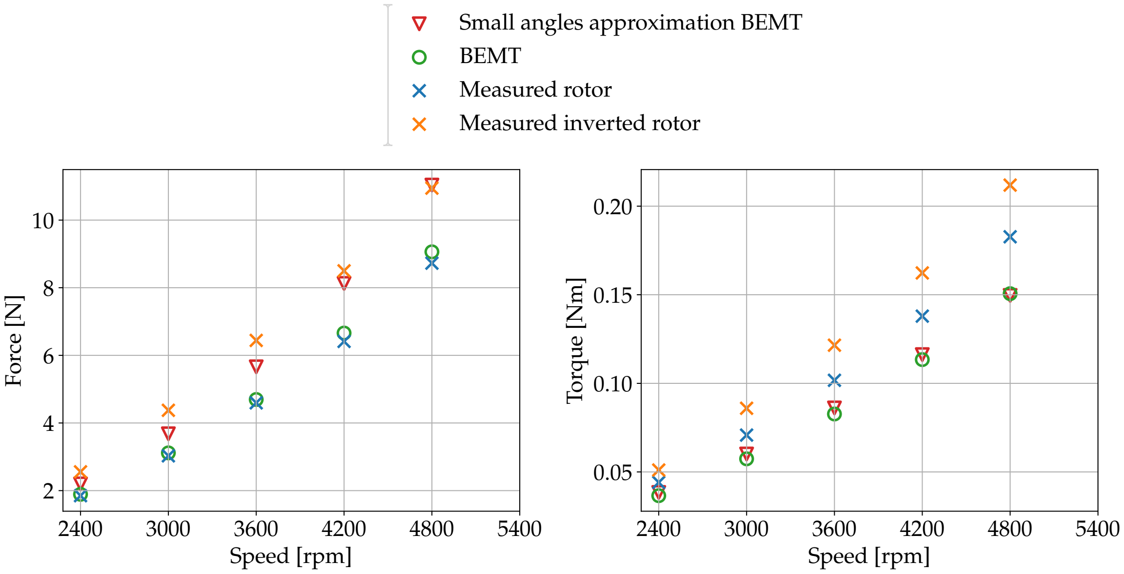


Figure A.8: Force and torque of the BEMT simulated and measured small rotor.

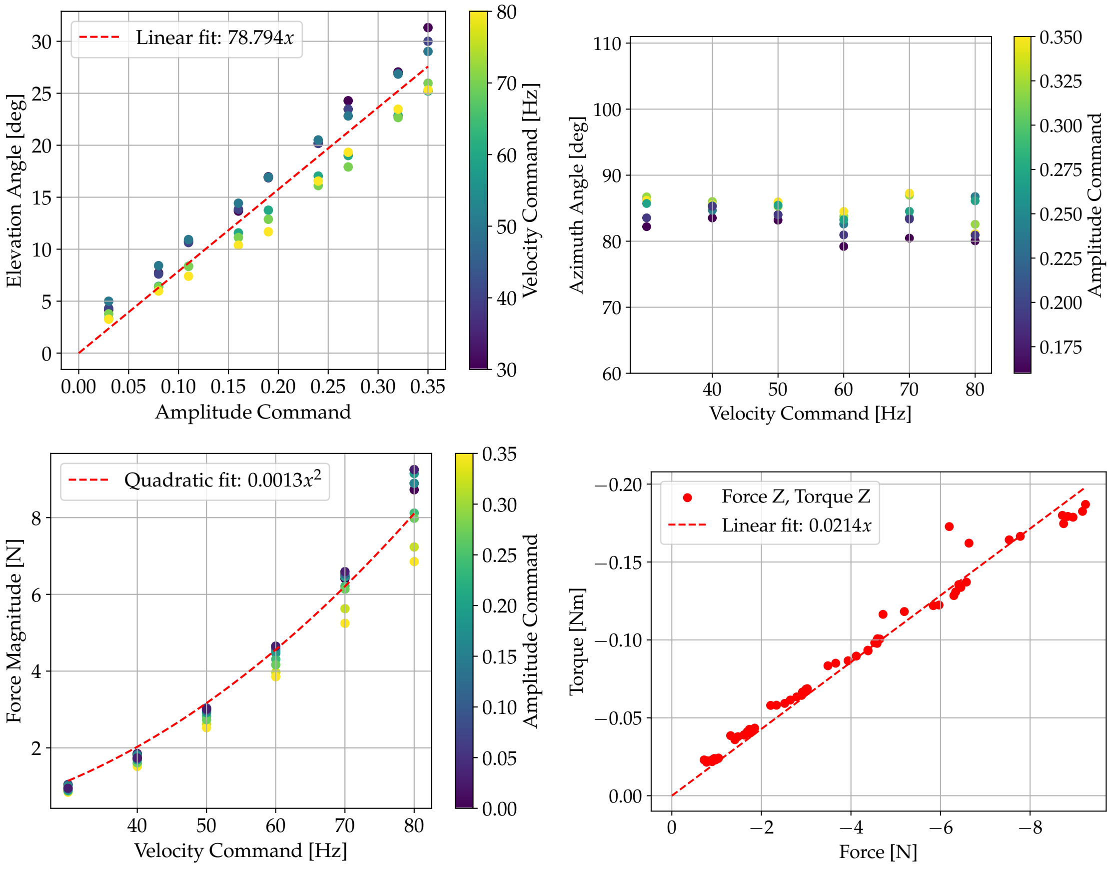


Figure A.9: Calibration data for smaller rotor measured on test stand.

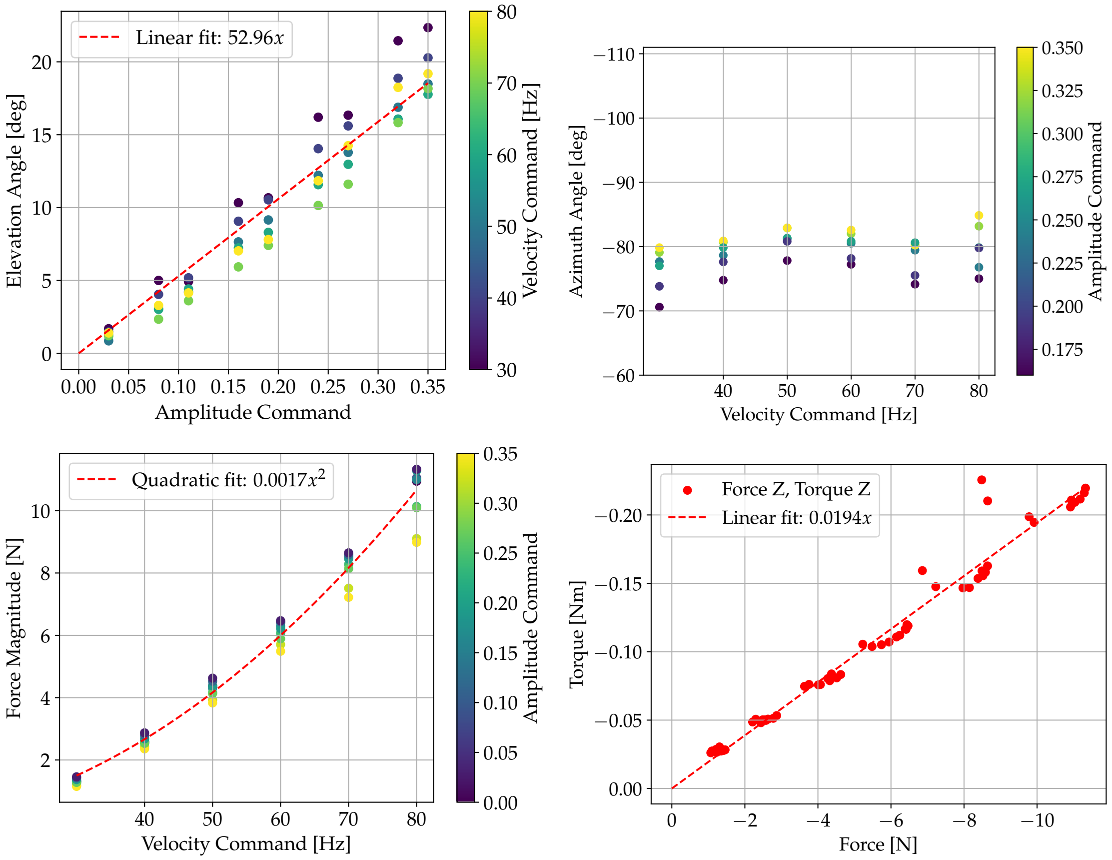


Figure A.10: Calibration data for inverted smaller rotor measured on test stand.

NTNU

Norwegian University of
Science and Technology

AU
RO
LA