Dual-Axis Tilting Quadrotor Aircraft

An investigation into the overactuatedness thereof

Nicholas Von Klemperer

Department of Electrical Engineering
University of Cape Town
Rondebosch, Cape Town
South Africa

November 2016

MSc thesis submitted in fulfilment of the requirements for the degree of Masters of Science in the Department of Electrical Engineering at the University of Cape Town

Keywords: Control, Allocation, Non-linear, Autopilot

"*We're gonna have a superconductor turned up full blast and pointed at you for the duration of this next test. I'll be honest, we're throwing science at the wall here to see what sticks. No idea what it'll do. Probably nothing. Best-case scenario, you might get some superpowers...*"

Cave Johnson -Founder & CEO of Aperture Science

Declaration

I, Nicholas Von Klemperer, hereby:

1. grant the University of Cape Town free license to reproduce the above thesis in whole or in part, for the purpose of research only;
2. declare that:
   (a) This thesis is my own unaided work, both in concept and execution, and apart from the normal guidance from my supervisor, I have received no assistance except as stated below:
   (b) Neither the substance nor any part of the above thesis has been submitted in the past, or is being, or is to be submitted for a degree at this University or at any other university, except as stated below.
   (c) Unless otherwise stated, any and all illustrations or diagrams demonstrated in this work are productions of my own.
   (d) All the content used to compile this report and complete the investigation revolving around the whole project is collectively hosted on the following GIT repositories:
      - LaTeXreport: `https://github.com/nickvonklemp/Masters-Report`
      - STM32F303 Projects: `https://github.com/nickvonklemp/Code`
      - Design Files & Blue Prints: `https://github.com/Design`
      - Hardware Schematics: `https://github.com/nickvonklemp/visio` &
      - EagleCad Schematics `https://github.com/nickvonklemp/Eagle`
      - MatLab Simulink Code: `https://github.com/nickvonklemp/Simulink`
      - Results & Simulation Data: `https://github.com/nickvonklemp/results`

Nicholas Von Klemperer

Department of Electrical Engineering
University of Cape Town

Friday 18th November, 2016

Abstract

Dual-Axis Tilting Quadrotor Aircraft

Nicholas Von Klemperer

Friday 18th November, 2016

The aim of this project is to design, simulate and control a novel quadrotor platform which can articulate all 6 Degrees of Freedom by vectoring each propeller's directional thrust. To achieve this the structure of the air-frame must redirect those thrust vectors to any desired orientation. This means it has to transform its configuration during flight, redirecting lift actuators whilst still maintaining stable attitude & position control, despite of such relative motion. In view of this required articulation the proposal is to add 2 axes (degrees) of extra actuation to each propeller. As a result each lift propeller can then be pitched or rolled relative to the body frame. This adaptation, to what is an otherwise well covered and highly researched platform, produces an over-actuated control problem. Actuator control allocation in the context of aerospace platforms is the primary contribution of this paper, with novel elements of non-linear (*state-space*) attitude control and plant uncertainty compensation.

The structure of the dissertation first presents the design which the subsequent dynamics and control are derived with respect to. Following that, the kinematics associated with rigid bodies are derived. Any unique effects that could apply to the design like gyroscopic, inertial and aerodynamic responses are investigated and then incorporated into the dynamics. Position and attitude control algorithms are first derived, then simulated and compared based on the plant's dynamics (*which include discretionary effects on the system*). The relative performance of the controllers are evaluated but regular performance metrics for attitude and position control are ill-suited for such a system. Some time is spent discussing the consequence of this and how the controllers are actually evaluated. Finally the design is built and tested using readily available RC components and conclusions drawn on the success or failure of the design.

The purpose of the investigation is the practicality and feasibility of such a design, most importantly whether the complexity of the mechanical design is a decent compromise for the added degrees of control actuation. The outcome of the build is to ascertain if it's economically feasible (cost and controller effort) to use such a prototype to expand the range of a quadrotor's motion. The design and control treatment presented here are by no means optimal nor the most exhaustive solutions, focus is placed on the system as a whole and not just one aspect of it.

This dissertation report is presented in a logical progression of concepts and information. In some cases the research and results were completed in a different order from how they're listed.

Acknowledgements

Nomenclature

DOF - Degree of Freedom(s)
μC - micro-controller
UAV - Unmanned Aerial Vehicle
MAV - Manned Aerial Vehicle
SISO - Single Input Single Output
MEMS - Microelectromechanical Systems
DIY - Do It Yourself
BLDC - Brushless Direct Current
VTOL - Vetical Takeoff/Landing
OAT - Opposed Active Tilting
dOAT - Dual axist Opposed Active Tilting
PI - Proportional Integral, control strucutre
PD - Proportional Derivative, control structure
PID - Proportional Integral Derivative, control structure
IBC - Ideal Backstepping Control
ABC - Adaptive Backstepping Control
LQR - Linear Quadratic Regulator
LCD - Lyupanov Candidate Function
PSO - Particle Swarm Optimization
ITAE - Integral Time Addative Error
BEM - Blade-Element Momentum
ESC - Electronic Speed Controller
RC - Radio Control
CH - Channel
MPC - Model Predictive Control
TSK - Takagi-Sugeno-kang
I/O - Input/Output
RPM - Revolution Per Minute
RPS - Revolution Per Second
W.R.T - With respect to

Symbols

Propeller Rotational Speed: $\Omega_i \ \ [rpm]$
Rotational speed in RPS is used for Blade Element Theory Calculations in Chapter:3
Servo 1: λ_i
Servo 2: α_i
Inertial Position: $\vec{P} = \begin{bmatrix} X_I & Y_I & Z_I \end{bmatrix}^T \ \in \mathcal{F}^I$
Body Position: $\vec{E} = \begin{bmatrix} x & y & z \end{bmatrix}^T \ \in \mathcal{F}^b$
Euler Angles: $\vec{\mathcal{E}} = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T$
Euler Rates: $\frac{d}{dt}\vec{\eta} = \dot{\vec{\eta}} = \Phi(\eta)\dot{\omega}_b$
Angular Velocity: $\omega = \begin{bmatrix} p & q & r \end{bmatrix}^T \ \in \mathcal{F}^b$
Linear Velocity: $\nu = \begin{bmatrix} u & v & w \end{bmatrix}^T \ \in \mathcal{F}^b$

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Foreword

1.1.1 A Brief Background to the Study

A popular topic for current control and automation research is that of quadrotor UAVs. Attitude control of a quadrotor poses a unique 6-DOF control problem, to be solved with an under-actuated 4-DOF system. As a result the ϕ pitch and θ roll plants aren't directly controllable. The attitude plant is often simplified around a stable operating point. The trimmed operating region is always at the inertial frame's origin; resulting in a zero-set point tracking problem. The highly coupled non-linear dynamics of a rigid body's transnational and angular motions arise from gyroscopic torques [Section: 3.2.1] and Coriolis accelerations [Section: 3.2.2]. These effects are negligible around the origin, hence the origin trim point removes the system's nonlinearities. The control system can then reduce[1] each state variable, $\vec{X}_b = \begin{bmatrix}\phi \ \theta \ \psi \ x \ y \ z\end{bmatrix}^T$, to individual SISO plants.

As almost every recent quadrotor research paper mentions, the late interest in the platform is from recent emergences in availability of MEMS systems and low-cost microprocessors. Such advancements accomodate onboard state estimation and control algorithm processing in real time. Developmental progress in quadrotors and, to a lesser extent, UAVs in general has led to rapidly growing enthusiast communities. HobbyKing [37] is now synonymous with providing custom DIY hobbyist quadrotor kits, no longer just a reailter for prebuilt commercial products like the DJI Phantom [23].

The avenue for potential application of both fixed wing and VTOL UAVs is expansive, supporting civil [65], agricultural [68] and security [49] industries. The quadrotor platform provides a mechanically simple platform on which to test advanced aerospace control algorithms. Commercial drone usage in industry is already emerging as a prolific sector; especially in Southern Africa. Subsequently following the 8^{th} amendment of civil aviation laws [71], commercial use of UAVs has been both legalized and regulated. Research into any non-trivial aspect of the field will therefore be to extremely valuable to the field as a whole.

Large scale quadrotor, hexrotor and even octorotor UAVs are popular intermediate choices for aerial cinematography due to their high payload capacity. The cost of a commercial drone like the SteadiDrone Maverik [54] is far less than a chartered helicopter used for the same panoramic aerial scenes or on-site inspections. One foreseeable issue which may hinder commercial drone progress in the agricultural and civil sectors is the consequential inertial effects from scaling up the aerospace bodies. When scaling up any vehicle, its performance is adversely affected if actuation rates aren't proportionately increased.

[1] Expanded upon in Appendix:A

1.1.2 Research Questions & Hypotheses

The difficulty with quadrotor control is that fundamentally it's unstable and under-actuated, *empirically proven later with Layupanov Theorem in Chapter:4*. A quadrotor has only four controllable inputs, namely propeller rotational speeds, $\Omega_{1,2,3,4}$, which are then abstracted[2] to virtual control inputs net torque, $\vec{\tau}_{net} = [\tau_\phi \ \tau_\theta \ \tau_\psi]^T$, and a perpendicular heave thrust $\vec{T}_{net} = \sum_{i=1}^{4} \ T(\Omega_i)$. Those four inputs have to affect both the linear X-Y-Z positions, $\mathcal{E} = [x \ y \ z]^T$, and angular pitch, roll and yaw rotations, $\eta = [\phi \ \theta \ \psi]^T$. Pitch and roll torques, τ_ϕ & τ_θ, are induced from differentials thrusts of each opposing propeller. Yaw torque, τ_ψ, is dependent on net aerodynamic torque about the rotational axes of each propeller (See Section:3.3.1). Aerodynamic responses are non-linear and fluctuating sources of control torques and as such the body's yaw control is depreciated. A result of the under-actuation is that the attitude control problem then becomes a zero set point problem, any other attempt to track attitude is ill-posed.

The aim of this project is to implement quadrotor attitude and position set point tracking by solving the problem of its inherent under-actuation. Inspired by Boeing/Bell Helicopter's V22 Osprey and the tilting articulation of its propellers, the prototype design proposed here introduces two additional actuators for each of the quadrotor's lift propellers. Specifically, adding rotations about the $\hat{X}$ and $\hat{Y}$ axes for each motor/propeller pair. The resultant is a vectored 3 dimensional thrust force rather than a bound perpendicular heave thrust. The control problem is then posed as the design and allocation of net forces, $\vec{F}_{net} = [F_x \ F_y \ F_z]^T$, and torques, $\vec{\tau}_{net} = [\tau_\phi \ \tau_\theta \ \tau_\psi]^T$, for a general 6-DOF body such that for any given trajectory, $X_d(t) = [x \ y \ z \ \psi \ \theta \ \phi]^T$, the error state $X_e(t) = X_d(t) - X_b(t)$ asymptotically tends to $\vec{0}$.

$$lim_{t\to\infty} X_e(t) = \vec{0} \quad \forall X \in \mathbb{R}^n \tag{1.1}$$

Where n is the degree of freedom the system has. The over-actuation brings about the need for a control allocation scheme which distributes the 6 commanded system inputs (net torques and forces) among the actuator set (12 actuators) in order to optimize some objective function secondary to that of Eq:1.1. The potential improvement(s) for exploiting those over-actuated elements is the most novel outcome that this project will yield. A cost function aimed at optimizing some aspect unique to aerospace bodies is going to be a completely unique contribution.

Part of the control research question is the multivariable treatment of the system, making no assumptions or simplifications to the non-linear dynamics involved in the quadrotors motion and its operational conditions. Standard linearizations applied to the quadrotor's control plant won't hold true for the more aggressive manoeuvres; they're dependent on small angle approximations and negligible 2^{nd} order effects. A stabilizing control law solution will need to expand and simulate the existing kinematic model of an aerial body and apply it to a quadrotor's motion. Following this there must be design, development and control of the new actuator suite which is to be implemented on a quadrotor platform. Final key outcomes for the project are the simulation analysis and prototype construction for the proposed design and the conclusion drawn thereon.

Introducing relative motion within an unconstrained body will produce a lot of unwanted dynamics like inertial and gyroscopic responses, amongst others. A rotating propeller will respond to pitching much like a Control Moment Gyroscope [94] or a flywheel, producing a precipitation torque. A less trivial aspect to consider is the aerodynamic torque produced from the propeller's aerofoil profile. Such induced responses occur in planes perpendicular to whatever the propeller's rotation exists in. These aspects aren't normally compensated for due to a quadrotor's fundamental co-planar propeller rotation. It's anticipated that a plant dependent control solution will have to compensate for these dynamics, which if left unaccounted for could potentially cause instability.

[2] The abstraction of which is explored in Appendix:A

1.1.3 Significance of Study

Due to the huge popularity of quadrotor platforms as research tools, any work that expands the UAV & quadrotor general body of knowledge will prove to be valuable. With that being said, there is already a vast amount of existing research on linear and non-linear control techniques for regular quadrotor platforms. The attitude loop is the most common topic for control research, requiring an under-actuated solution and mostly linearized around the origin (See Appendix:A). Far less common is the application of optimal flight path and trajectory planning to quadrotor control. The uniqueness and difficulty of the quadrotor attitude control does not hold true for its position control, so standard techniques can be used for way point planning and the like once the attitude control problem has been answered.

The most significant aspect of this project is the attitude control, discussed later in Section:4.1. The over-actuation of the proposed design and, more critically, the manner in which the controller's (virtual) output is distributed among those control effectors would appear to be the first of its kind. Otherwise known as control allocation, the requirements of the distribution algorithm(s) are outlined in Section:4.3. Dynamic set point attitude control for aerospace bodies is not a subject heavily researched outside the field of satellite attitude control. Even papers which propose similarly complicated mechanical over-actuation (expanded upon in next in the literature review, Section:1.2) hardly broach the topic of tracking attitude set points away from the origin.

Whilst the control plant (developed in Chapter:4) does indeed close both the position and attitude control loops, there is no consideration of trajectory generation nor flight path planning. Such topics are well discussed elsewhere in a far more concise and deliberate way than this project could ever hope to achieve. Once closed loop position and attitude control have been achieved, the control algorithms can be adjusted to account for higher order state derivative (acceleration, jerk and jounce) tracking needed for nodal way point planning. The heuristics involved with flight path planning are well documented and their implementation is a relatively academic task.

Where possible the system identification and control (design and allocation) for this project is kept both modular and generically applicable. The intention here is that its pertinence falls not only within the UAV field but to any aerospace or free body attitude control. Hopefully this investigation can be expanded upon with more in-depth research on one of the subsystems without compromising the stability of the whole plant. Provisionally, an obvious outcome which the investigation could yield is improved yaw control of a quadcopter's attitude. However, if the express purpose was just to improve yaw control, it could be done with a dramatically less complicated design.

Furthermore, the project could provide greater insight into high bandwidth actuation and thus a faster control response for larger aerospace bodies. Any standard quadrotor uses differential thrust to develop a torque about its body. Such actuation suffers a second order inertial response when the propellers accelerate or decelerate, $\tau_{simplified} = \mathbb{I}_f \; \dot{\omega}_i$. Prioritizing pitching the propeller's principle axis of rotation rather than changing the propeller's speed could potentially improve the virtual control response. This is entirely dependent on how the allocator block is prioritized (presented in Section:4.3). The exact effects of different actuator prioritization and distribution in the context of aerospace control are wholly unique to this dissertation.

1.1.4 Scope and Limitations

Scope

Critical to this project is the conceptualized design and prototyping of a novel actuation suite to be used on a quadrotor platform. The express purpose of which is to apply dynamic set point attitude control to the body. Stemming from this is an investigation into the kinematics that are potentially influenced by the design and the structure's relative motion. In order to apply correct control theory to achieve the attitude tracking on a physical prototype, the plant dynamics must first be identified for input responses to be approximated with confidence. Aspects of the mechanical design are covered next in Section:2.1 but, beyond the cursory investigation, there is no scope for materials analysis or stress testing of the design. To the detriment of the project, the design will either produce an over-engineered or catastrophically under-engineered solution. The scope focuses mainly on the control application and embedded systems design, not the structural integrity of a proposed frame given the forces it may undergo. Physical measurements are only made for critical kinematics, such as inertial measurements for the second order gyroscopic and inertial dynamic responses.

As mentioned in the antecedent Section:1.1.3, trajectory & flight path planning are not ubiquitous with this dissertation. Derivations for the differential equations which dictate a 6-DOF body's movement are wholly applicable to any dynamic (rigid or otherwise) aerospace body, although some particular standards are used (sic Z-Y-X Euler Aerospace Sequence, Section:2.2). Similarly the control plant is stabilized with non-linear state space control techniques, aided and justified by Lyupanov theorem. Alternative solutions using Model Predictive Control or Quantitative Feedback Theory could provide more refined or effective controllers, however they aren't presented and remain open to further investigation. Quadrotor attitude control is commonly stabilized with feedback linearizations, decoupling the plant around a trim point so that SISO techniques can be applied. A derivation of such a linearization is included in Appendix:A but beyond that there are no further discussions. Any comparison between non-zero and zero-set point attitude control of quadrotor is difficult as the fundamental objectives are in stark contrast with one another.

Arguably the most important and indeed novel aspect of this project is the control allocation. The system has 12 plant inputs and 6 output variables to be controlled. There is then a family of actuator set solutions, $u \in \mathbb{U}$, which exist for each commanded input. Such a plant is classified as over-actuated. Ergo, there must be some logical process as to how those 12 inputs are articulated to achieve the desired 6 control plant inputs. Appropriate techniques are first investigated in Section:4.3 and compared before a final solution is implemented in Section:5.4. It is by no means a comprehensive investigation of every possible allocation scheme but rather an analysis of the sub-set of problems and design of what is regarded as a logical and pertinent approach.

With regards to the actual prototype design, in Section 2.1, it's assumed that certain aspects are a given certainty. Particularly the state estimation, updated through a 4-camera positioning system fused with a 6-axis IMU through Kalman Filtering, is assumed to precise and readily disposable at a consistent 50 Hz. Hence state estimation is included but is bereft of intricate detail, this is another topic which remains open to further investigation.

Limitations

The biggest constraint faced by the design is the net weight of the assembled frame. Lift forces required to keep the body aloft are obviously dependent on the all up weight. Conventional wisdom has it that steady state actuator rates ought to be far less that saturation conditions. For stability to be guaranteed at all feasible operating conditions, the actuators must have sufficient headroom to still effect the desired control inputs. Conversely the structure's net weight is mostly dependent on the lift motors, often being the heaviest part of the vehicle (*batteries included*). A trade-off between net weight and actuator efficacy makes designing the prototype a balancing act of compromise; added actuation is needed to produce the desired thrust vectoring. That added actuation is going to increase the weight which then requires more thrust force to ensure the vehicle remains airborne. Larger motors then need stronger actuators to effect the relative motion and overcome the bodies inertial response. It's a compromise between the weight of the body and the strength/quality of the actuation.

To forego the deliberation detailed above, reducing the possibility of unbounded scope creep, a limitation is self-imposed on the prototype design. Restricting the propeller diameter, and hence maximum thrust/frame size, will provide a constraint upon which all other design considerations must adhere to. Smaller propellers require far greater rotational speeds to produce similar levels of thrust that their larger diameter counterparts could provide. Electing to use 3 blade $6X4.5$ inch small diameter propellers is going to reduce the overall dimensions of the prototype, but as a consequence will require very high RPM motor. Specifically a set of four Cobra-2208/2000KV [20] Brushless DC motors are to be used for lift actuation (Fig:1.1a)[3]. A direct consequence of this decision is that, provisionally based upon test data[4], the net thrust disposable for actuation is limited to around 950g, $\approx$ 9.5 N, per motor (see Section:3.3.1). It's critical to ensure the control block doesn't induce over-saturation of the motor actuation, so the frame weight needs to be around 50% of the maximum available thrust, or roughly 2 Kg. Saturation conditions are detailed later in Section: 4.3.

Another aspect of limitations produced by design decisions made, mostly to reduce prototype costs and weight, is to use of 180° rotation servo motors. Here Corona DS-339MG metal gear digital servos (Fig:1.1b)[5] are used. The servos are for each individual motor's $\hat{X}_{M_i}$ and $\hat{Y}_{M_i}$ axial pitch and roll actuations respectively. The servos act in lieu of either continuous BLDC or stepper motors. Any non-servo rotations beyond 360° will require closed loop position control and, unlike servos, would need slip rings to transmit power throughout rotational movement. However the logistics of implementing such a design whilst maintaining an acceptable weight is almost impossible. Such an implementation is going to dramatically increase the size of the prototype to accommodate for weight increases. Commercial camera stabilizing gimbals already make use of similar configurations but the I/O requirements from the flight controller μC already constricts the amount of expansion available.

(a) Cobra CM2208/2000KV BLDC motor

(b) Corona DS-339MG digital servo

Figure 1.1

Discrete elements for the whole system can potentially limit performance but are going to be mitigated if possible. For example analogue servos have an associated $1ms$ deadband from their $50Hz$ refresh rate. That can be addressed by using faster, albeit more expensive, digital servos which samples at $330Hz$. The prototype's flight controller needs to provide 12 PWM output compare channels for the 8 servos and 4 BLDC speed controllers. State updates from a ground control station and a fail safe 6CH RC receiver module also needs to be processed by the μC system. Particular attention is paid to the embedded system design and layout in Section:2.3.

[3] Image cited from: `http://www.getfpv.com/cobra-cm-2208-2000kv-motor.html`
[4] Official test data from [20] included in Appendix:C.2 and verified independently in Section:3.3.1
[5] From the DS-339MG product page, HobbyKing [37]

1.2 Literature Review

1.2.1 Existing & Related Work

The field of transformable aerospace frames is not necessarily a new one, with many commercial examples having seen a lot of success over their operational life span. The most notable tilting-rotor vehicle is that of the Boeing/Bell V22 Osprey [?] aircraft. First introduced into the field in 2007, the Osprey has the ability to pitch its two lift propellers forward to aid translational flight after vertically taking off or landing. In addition to this there have been many papers published on similar tilting bi-rotor UAVs for research purposes.

Birotors

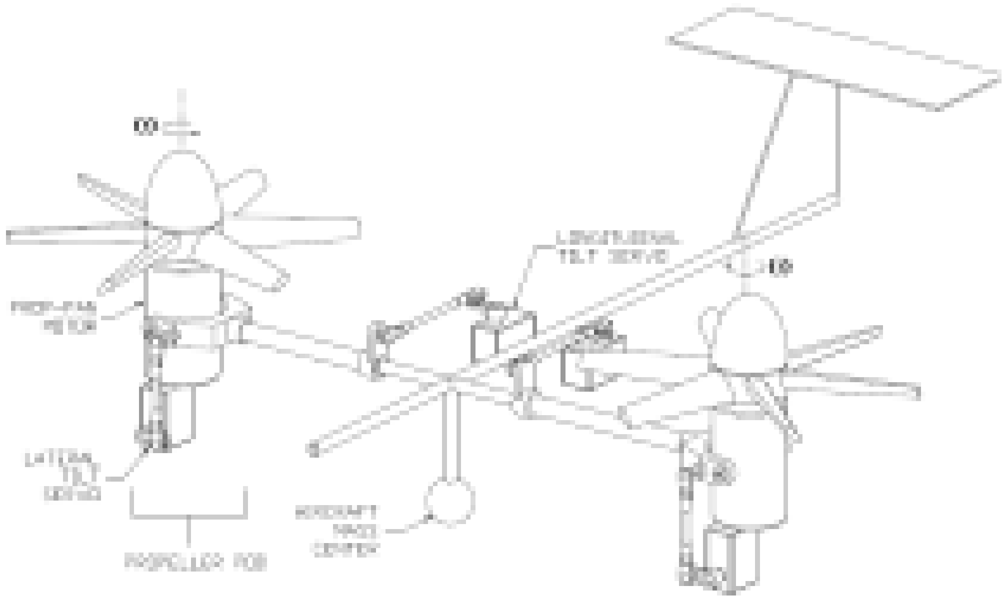


Figure 1.2: General structure for opposed tilting platform

Research into birotor vehicles (Fig:1.2)[6] with ancilliary lift propeller actuation is oft termed Opposed Active Tilting or *OAT*. Such a rotorcraft's mechanical design applies either a single *oblique* 45° tilting axis relative to the body; [9, 30, 45], or a *lateral* tilting axis, adjacent to the body; [16, 46, 67, 81]. Leading research is currently focussed on applying doubly actuated tilting axes to birotor UAVs. Dual axis Opposed Active Tilting or *dOAT* introduces vectored thrust with propeller pitch and roll motions to further expand the actuation suite, [2, 29]. A birotor is sometimes considered preferable to higher order multirotor platforms due to their reduced controller effort. However the controller plant abstraction often detracts from the quality and effectiveness of its stability solution as a result of the birotor's underactuation.

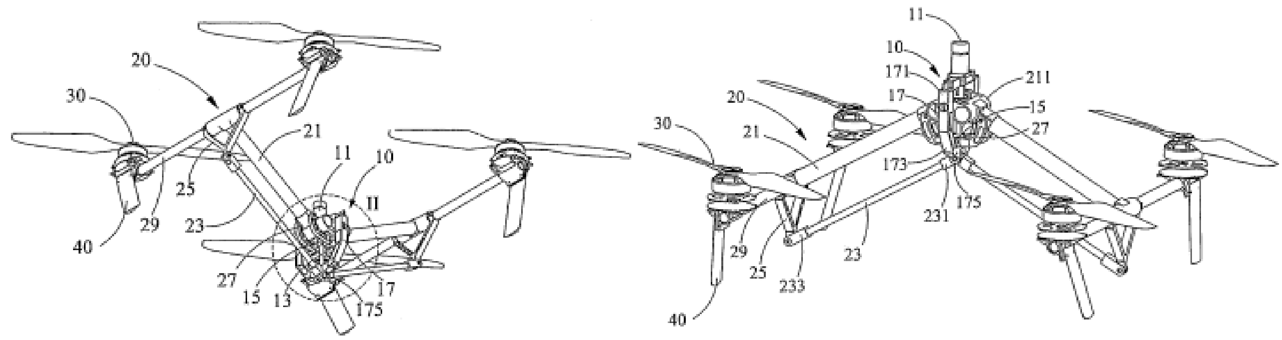
Birotor attitude control typically incorporates plant independent PD [9] and PID [67] controller schemes. Occasionally more computationally intensive and plant dependent Ideal and Adaptive back-stepping controllers (*IBC* or *ABC*) are implemented, presented in [45, 81] and [46] respectively. The cross-coupling of a birotor vehicle's attitude system is more pronounced than that of a quadrotor, derived in Section:3.2, and so feedback linearisation is almost always used. In an interesting progression from the norm, Lee et al, [51], proposed a PID co-efficient selection algorithm for a bi-rotor control block. Using a Particle Swarm Optimization techinque, similar to [96], the coefficients were globally optimized around a given performance metric. However their performance criterion is a basic

[6]Image from G. Gress: [29]

ITAE† term and nothing more appropriate involving effects unique to flight systems. *PSO* algorithms iteratively search for a globally optimized solution and offer independent, derivative free optimization. Later on non-linear controller coefficient are also optimized in this paper using a PSO algorithm, shown in Section:5.1.

Quadrotors

Expanding on multirotor vehicles, the quadrotor UAV is a popular and well researched multirotor platform due to its mechanical simplicity. What would appear to be one of the first quadrotor research implementations, in 2002, is the X4-Flyer quadrotor, [32, 73]. Alternative iterations like the Microraptor [76] and STARMAC [38] quadcopters have subsequently been built and tested. A plethora of literature exists around quadrotor kinematics & control [4, 12, 19, 55, 75], however dedicated rigid body 6-DOF dynamic papers [57, 69] provide better explanations of the kinematics. Often the plant's dynamics are simplified around an origin trim point and assumed to reduce into 6 SISO plants for each degree of freedom (Appendix:A). Lately research projects have begun to incorporate aerodynamic effects like drag and propeller BEM theory into the plant model [14, 38, 78]. Although mostly negligible under standard opperating conditions, the higher fidelity models offer more precision without making any linearisations or assumptions, [5, 38].



(a) Inspire1 articulated upwards **(b)** Inspire1 articulated downwards

Figure 1.3: DJI Inspire1

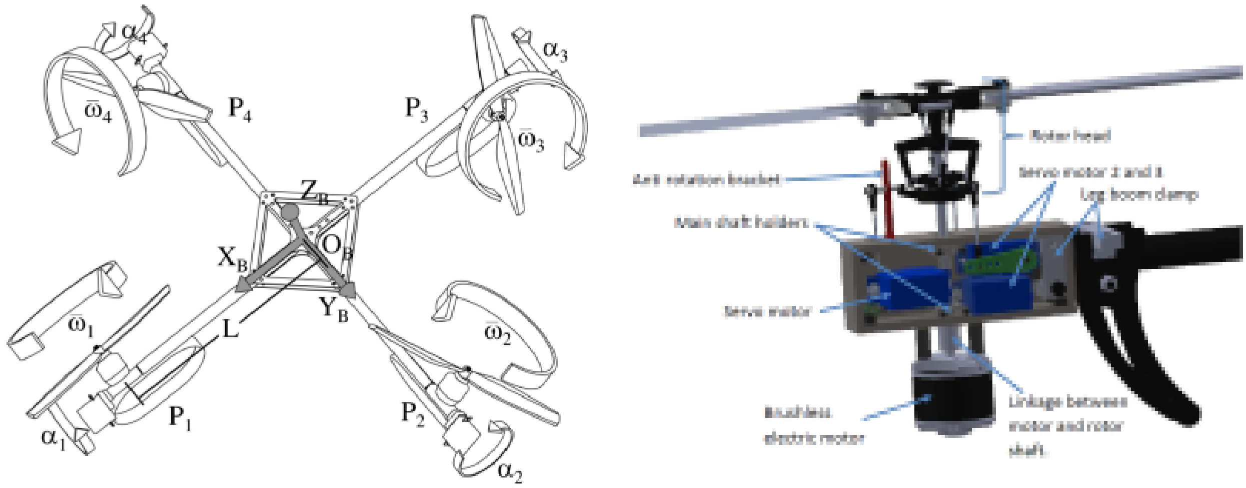
At the time of writing, the only commercial UAV multirotor capable of structural transformation is the DJI Inspire1 quadrotor [22], made by Shenzen DJI Technologies (better known for the hugely successful DJI Phantom drone [23]). The Inspire1 can articulate its supporting arms up and down as shown in Fig:1.3 [7]. The aim of such movements is to both alter the center of gravity and further expose a belly mounted camera gimbal for panoramic viewing angles. This transformation changes the moment of inertia about the body's center of gravity, in turn changing the inertial torque response induced by angular velocities, an otherwise detrimental effect which makes researchers apprehensive of transformable aerospace frames. The range of transformations which the frame can undergo is limited to just articulating the arms up and down.

In a similar fashion to the progression seen in birotor state-of-the-art, quadrotor research is engaging the topics of single and dual axis tilting articulations. First conceptualized and implemented on a prototype related to an ongoing project covered in two reports, [79, 80]. The authors M. Ryll et al.(2012, 2013) modified and tested a QuadroXL four rotor helicopter, propduced by MikroKopter [27], to actuate only a single axis of tilting aligned with the frame's arms (Fig:1.4a)[8]. Their proposed control solution, discussed in detail next in Section:1.2.2, assumes no nominal linearised conditions around hover flight, unlike a similar single axis tilting quadrotor prototype designed by Nemati, et al. (2012) [62]. The latter remains simulated but as yet untested.

[7]Both images were sourced from the drone's patent, held by SZ DJI Tech Co [95]

[8]Image sourced from Modelling and Control of a Quadrotor UAV with tilting propellers, [79]

One approach to improving quadrotor flight response is to alter the manner in which the thrust is mechanically actuated, potentially improving the actuator bandwidth. Drawing from helicopter design, a project by Napsholm, (2013) [61], purported a quadrotor UAV prototype that used swashplates for varying the propeller pitch and generating torque moments. The aim was a design which wasn't dependent on speed control (*ESC*) power electronics to actuate variable thrust forces. Petrol motors were intended for use in place of BLDC motors. Furthermore, the design proposed a single axis of tilt actuation to each of the four motor modules. Whilst mechanically complex, Napsholm made use of existing RC helicopter components to design a rotor actuation bracket (Fig:1.4b). The cyclic-pitch swashplate used [63] could apply torques, τ_ϕ and τ_θ, about each propeller's hub, *principle axis of rotation*, by altering the blades angle of attack throughout its rotational cycle. The actuation rate of such a configuration is far faster than that of a differential torque produced rolling/pitching motion.



(a) Single rotation axis aligned with the frames arm **(b)** Cyclic-pitch & swashplate mechanism

Figure 1.4

Irrespective of the strong initial design in the early stages of his project, it would appear that Napsholm's research suffered due to time constraints. The introductory derivation on aerodynamic effects and deliberation over the design provide clear insight into the projects goals. However the control solution and system architecture, electronic and software, are significantly lacking. An introductory proposal of an MPC attitude control system detracted from the comprehensive dynamics discussed. The project ended before testing, simulation and results could be obtained. Unfortunately, despite the novel over-actuated design, there was no discussion given on how the allocation, being the most unique aspect, would be achieved.

Finally, the most crucial research to mention is a project completed by Pau Segui Gasco [26], which was a dual presented MSc project with Yazan Al-Rihani [1]. At the time of writing, this would appear to be the only project published pertaining to *over-actuation* in aerospace bodies implemented on a quadrotor platform. The research was split between the two authors who completed the control/electronic design and the mechanical design for their respective MSc dissertations. Shown in Fig:1.5[9], the dual-axis articulation is achieved using an RC helicopter tail bracket and servo push-rod mechanism; reducing the mass of the articulated component but limiting the range of its actuation. Considering the propellers as a spinning flywheel, the induced gyroscopic response was then be treated as a controllable actuator plant. Their commanded virtual control is distributed by weighted inversion among the actuator set, Section: 1.2.2. The whole project justifies the extra actuation as fault tolerance redundancy but doesn't necessarily prove how such a redundancy could be beneficial.

[9]Image from Development of a Dual Axis Tilt Rotorcraft UAV: Modelling, Simulation and Control [26]

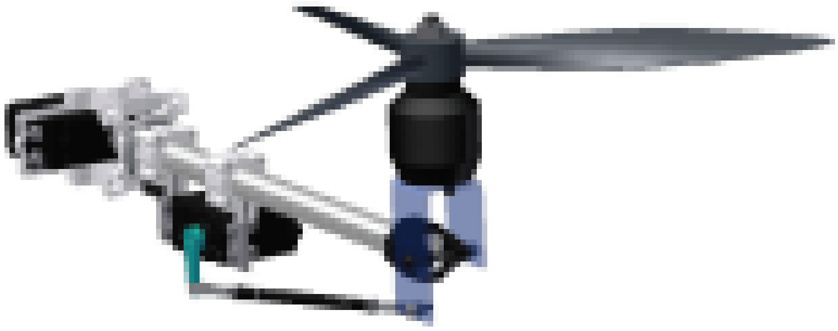


Figure 1.5: Dual-axis tilt-rotor mechanism

1.2.2 Notable Quadrotor Control Implementations

Quadcopter Attitude Control

Attitude control of a 6-DOF body is best described by *The Attitude Control Problem* [89]. A rigid body that currently has an attitude state[10] $\vec{\eta}_s$ and a desired state $\vec{\eta}_d$, the problem is to then find a torque control law:

$$\mu\tau = H(\vec{\eta}_s, \vec{\eta}_d, \dot{\vec{\eta}}_s, \dot{\vec{\eta}}_d) \tag{1.2}$$

Such that both the angular position $lim\ \vec{\eta}_s \to \vec{\eta}_d$ and that angular rates $lim\ \dot{\vec{\eta}}_s \to \dot{\vec{\eta}}_d$ asymptotically stabilize as $t \to \infty$. A distinction must be made between angular rate vector, $\dot{\vec{\eta}} = [\dot{\phi}\ \dot{\theta}\ \dot{\psi}]^T$ and the angular velocity vector $\vec{\omega}_b = [p\ q\ r]^T$. Depending on how the attitude is posed; with rotation matrices [48, 57, 69], quaternions [25, 28, 31, 48] or otherwise (Direct Cosine Matrix etc ...) the error sate[11] $\Delta\vec{E} = \vec{\eta}_d - \vec{\eta}_s$ could then differ to a (hamilton) multiplicative relationship.
Note that here $\vec{\eta}$ is not necessarily an Euler set but any attitude representative state variable.

Simulation and modelling papers often rely on Euler angle based rotation matrices for attitude representation, [11,12,56,62,77] without addressing the inherent singularity associated with such an attitude representation (sic Gimbal Lock, [82], Section:3.1.2). The alternative quaternion attitude representation, first implemented on a quadrotor UAV in 2006 [87], is often used in lieu of rotation matrices but has its own caveat of *unwinding*, (Section:3.1.4), as a result of quaternions dual-coverage [59] in $\mathbb{R}^3$ space. Quaternions are $\in \mathbb{R}^4$ variables for attitude representations and so a mapping $\to \mathbb{R}^3$ produces a dual coverage for each attitude state.

Quadrotor plant dynamics, as mentioned previously, are often simplified; especially when represented with a 3-variable Euler angle set, $\vec{\eta} = [\phi\ \theta\ \psi]^T$. The coupled gyroscopic and Coriolis responses are both neglected when the angular velocity[12] is small, $\vec{\omega}_b \approx 0$, and the inertial matrix is diagonal, $rk(\mathbb{I}_f) = x$ for $\in \mathbb{R}^x$. The consequence of which is the ineffectual deterioration of both the gyroscopic term, $\vec{\tau}_{gyro} = -\vec{\omega}_b \times \mathbb{I}_b\vec{\omega}_b \approx 0$ and the Coriolis force term, $\vec{F}_{cor} = -\vec{\omega}_b \times \vec{a}_b \approx 0$ in the bodies dynamics (Chapter:3 for context). Once the coupled cross-product terms are no longer of consequence, the 6 DOF, $[x\ y\ z\ \phi\ \theta\ \psi]^T$, can be treated as a series of individual SISO plants each controlled with an appropriate technique. Quaternion represented attitude plants cannot easily be decomposed into individual single-input-single-output systems (quaternion dynamics in Section:3.1.3). So a quaternion (combined four variable attitude state vector) is then used, $Q_b = [q_0\ \vec{q}\,]^T$ for the abstracted major loop plant.

[10]Quaternion attitude states will later replace Euler angles
[11]*The Attitude Control* [89] describes these conventionally different error states
[12]Angular velocity and angular rates are fundamentally different, $\vec{\omega}_b \neq \dot{\vec{\eta}}$

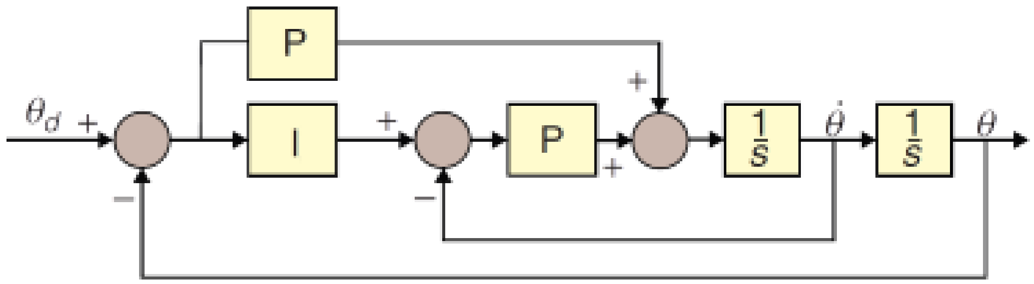


Figure 1.6: ArduCopter PI Euler angle attitude control loop

Commercial flight controllers' software (Arducopter [3], Openpilot [52][13], CleanFlight [17], BetaFlight [7], etc ...) for custom fabricated UAV platforms all apply their own flavour of structured attitude controllers and state estimation algorithms, based on onboard hardware sensor fusion. The article *Build Your Own Quadrotor* [53] summarizes the control structures implemented on a range of popular flight controllers. The most popular of which, ArduCopter, implements a feed-forward PI compensation controller (Fig:1.6)[14]. PI, PD and PID controllers are all easy and effective plant independent control solutions for general attitude plants. Table:1.1 collectively lists the common attitude control blocks (not exclusively quadrotors UAVs but MAVs too) and which projects they've been implemented in, after which a critique on the more unique adaptations is given.

Controller Type	Independent	Dependent	Total
PI	[89]	[89]	2
PD	[1,55]	[25,62]	4
PID	[11,13,75,79,89]	[38,77,89]	8
Lead	[19,73]	lead	2
IBC	[56,81][15]	[56]	3
ABC	[6,21,46,60]		4
LQR	[13]	LQR	1

Table 1.1: A breakdown of common attitude controllers

In a collection of papers, written by Bouabdallah et al ... (2003,2004,2007) arguably the most prolific early quadrotor authors, a range of different control implementations are derived and reviewed. Their last paper (2007) [12] derived and pratically tested an Integral Backstepping attitude controller on an OS4 quadrotor platform. It builds on their research from an earlier paper (2003) [13] wherein an analysis of PID vs LQR attitude controllers in the context of quadrotors is posed. LQR controllers aim to optimize the controller effort (with $u \in \mathbb{U}$, controller effort is then $||u||$ or the L_2 norm of the plant input). Although, in theory, solving the assocaited Ricatti[†] cost function may produce an optimial, stable and efficient control law it needs exact plant matching. In practice, exact plant matching is difficult to achieve for a quadcopter or any aerospace body for that matter. The resultant controller in [13] achieved asymptotic stability but had poor steady state performance due to low accuracy of the identified actuator dynamics and poor inertial measurements.

Adaptive Backstepping Control [93](any of the examples in Table:1.1) builds on nominal IBC fundamentals by introducing an aditional disturbance state term in the LCF used for the backstepping iteration. The drawback with this form of Backstepping approach is that, from the Lyupanov control theorem, a time derivative for the estimated disturbance (or an *update law*) is needed. Disturbance approximation has been investigated thoroughly but, for a signal without *apriori* information, some heuristic needs to be adopted with the approximation which usually involves some compromise.

[13] NOTE: OpenPilot's firmware stack is now maintained by LibrePilot
[14] Image sourced from *Build your own Quadrotor* [53]

In one example, [21], the authors implemented a statistical $proj(.)$ operator based technique. Which, when used in adaptive control, the projection operator [15], $proj(.)$, ensures a derivative based estimator is bounded for adaptive regression approxmation [72].

Although the control implementation isn't backstepping based, in [97], a sliding mode controller was used to compensate for the disturbances in an Unmanned Submersible Vehicle atttiude plant. The underwater current disturbances were approximated using a fuzzy logic system, specifically a *zero-order TSK* fuzzy controller. The TSK system has been proven to act in the same way as an Artificial Neural Network approximator [58]; where the TSK system is more comprehensible than the latter. Statistical analysis and investigation of approximators without *apriori* knowledge of a system are well beyond the scope of this research but are worth mentioning.

Single/Dual Axis Control & Allocation

The extra actuation introduced with single and dual axis articulation provides room for more control goals to be achieved as the order of actuation increases. Of the few papers published on tilting-axis quadrotors, PD controllers (Nemati et al.[2014] [62] and again in Gasco & Rihani [1, 26]) and PID controllers (Ryll et al.[2012,2013] [79,80]) are the norm for control blocks. For either of these systems there needs to be an allocation rule to distribute a commanded input amongst the actuator set. In [42], Johansen et al.[2012] describes the control allocation problem for a dynamic plant:

$$\dot{x} = f(x,t) + g(x,t)\tau \tag{1.3a}$$
$$y = l(x,t) \tag{1.3b}$$

Note in state space Equation:1.3a, it's assumed the plant input[16], τ, has a linear multiplicative relationship with the input response, $g(x,t,\tau) \iff g(x,t)\tau$.

With a state $x \in \mathbb{R}^n$ and $f(x,t)$ & $g(x,t)$ being the plant's dynamics and input response respectively. In set point tracking, the output is then *tracking* the state $y = x$, and hence $y \in \mathbb{R}^n$. In an ideal well posed system the number of actuator inputs equals the number of controllable variable outputs; that being $dim(x) = dim(\tau) \in \mathbb{R}^n$. In the case where the control input $\tau \in \mathbb{R}^m$, if $m > n$ the problem is then overactuated and a level of abstraction is needed; an asymptotically stabilizing virtual control input ν_d is designed by a control law $\nu_d = h(x_e, t)$ to affect dynamics. The goal is to then find a function that maps $\mathbb{R}^m \to \mathbb{R}^n$ for an actuator matrix $u \in \mathbb{U}^m$. An overactuated plant can be described in state-space as:

$$\dot{x} = f(x,t) + g(x,t)\nu_d,\ \nu_d \in \mathbb{R}^n \tag{1.4a}$$
$$\nu_c = B(x,t,u) \approx B(x,t)u,\ u \in \mathbb{U}^m,\ \nu_c \in \mathbb{R}^n \tag{1.4b}$$
$$y = x \tag{1.4c}$$

$B(x,t,u)$ is the effectiveness function which quantifies how the actuator inputs u relate to the virtual commanded input ν_c. $B(x,t,u)$ can be abstracted to a multiplicative relationship $B(x,t)u$ if the plant's dynamics permit it, such that; $B(x,t) \in \mathbb{R}^{n\times m}$. For generic setpoint tracking the control law will design a desired virtual control input ν_d, the allocation rule then has to solve u for ν_c such that a slack variable $s = \nu_c - \nu_d$ is minimized:

$$\min_{u\in\mathbb{R}^m, s\in\mathbb{R}^n} \|Q_s\| \text{ subject to } B(x,t,u) - h(x_e,t) = \nu_c - \nu_d = s,\ u \in \mathbb{U} \tag{1.5}$$

Which ensures the commanded input ν_c tracks the desired control input ν_d; $\nu_c \to \nu_d$ as per some cost function of the slack variable Q_s. Mostly the L2 norm, $\|Q_s\|$, is used. In an overactuated system it then follows that there is a set of possible inputs for each ν_c. A unique actuator solution (rather than a family solution set) to Eq:1.5 needs a secondary objective function, $J(x,t,u)$. Eq:1.5 then becomes;

$$\min_{u\in\mathbb{R}^m, s\in\mathbb{R}^n} (\|Q_s\| + J(x,t,u)) \text{ subject to } \nu_c - h(x_e,t) = s, u \in \mathbb{U} \tag{1.6}$$

[16]Disambiguation: τ is not necessarily the torque input.

Those same authors Johansen and Tjnns [2004,2005,2008] proposed multiple control allocation solutions to a variety of systems. Following [42], in a subsequent paper [43], Johansen and Tjnns [2005] introduce a secondary cost function, driving the solution away from the typical quadratic programming direct or weighted inversion solution. Aiming for optimal efficiency and not just actuator saturation. In a followup paper [44], the same authors proposed an online adaptive algorithm approach, using a Lyupanov energy function, to ensure the minimization adaptive law settles to a feasible solution.

Over-actuation is not something often applied to quadrotors and as a result rather than providing a comprehensive literature review of associated papers here (which are all mostly theoretical derivation), the contextual application and solutions to the above posed problems are expanded later in Section:4.3.1. The only overactuated quadrotor (birotor dual-axis tilting makes the system critically actuated and so requires no allocation) literature which covers allocation of the extra actuators is [1,26], where the authors apply a weighted pseudo inverse (sic Moore Penrose Inverse [50]) allocation rule. A prerequisite for pseudo inversion is a multiplicative *linear* control effectiveness relationship for Eq:1.4b.

Segui et al. [2012] applied weighted inversion, relying on some very specific assumptions to achieve that linearity relationship in Eq:1.4b. For the net torque response the authors assumed the extra actuators pitch and roll angular rates, $\dot{\phi}$ and $\dot{\theta}$ respectively, were proportionally related as follows:

$$\dot{\phi} \approx \frac{\phi}{t_{rise}} \tag{1.7}$$

In which t_{rise} is the actuators rise time to a set-point. As a result the gyroscopic first order torque $\tau_{gyro} = -\omega \times \mathbb{I}_f \omega$ and second order inertial torque $\tau = \mathbb{I}\dot{\omega}$ are then functions of position ϕ or θ and not their derivatives. The extent of that consequence is contrasted with the allocation solution in Section:4.3.

Satellite Attitude Control

Unconstrained attitude set-point tracking for 6-DOF bodies, quaternion represented or otherwise, is a topic well covered in the field of satellite attitude control; [41,47,91]. The *status quo* for recent research is on non-linear adaptive attitude back-stepping control systems, wherein the adaptive update rule is the novel contribution. Often plant uncertainty affects the inertial tensor of a satellite. In [41], the authors Wang Jia, et al. [2010], proposed applying adaptive back-stepping to compensate for steady state errors of (asymmetric) inertial estimations. Alternatively, instead of deliberating on costly non-orbital prelaunch inertial measurements Bodrany, et al.[2000] [10] developed an algorithm for estimating the inertia tensor based on controlled single axis perturbations. Such an approach does assume any initial estimates are sufficiently close to true body values such that they will settle and stability can be ensured, irrespective of how unacceptable the transient performance may be.

Satellite actuator suites mostly include additional redundant effectors, to ensure fault tollerance, and thus require control allocation. Often the extra allocators are CMG actuators, driven by DC motors, to produce rotational torques. Fuel burning can only actuate for a certain period of time and so thrusters are scheduled to have a lower priority. Seen in the paper [47]; the authors, Kristiansen et al. [2005], address the over-actuation with direct and well-matched inversion before applying quaternion based back-stepping for attitude control. A direct inversion solves to Eq:1.6 such that:

$$u = B^{\dagger}({\tau_a}^b - D{\omega_{ib}}^b) \tag{1.8a}$$

$$B^{\dagger} = B^T(BB^T)^{-1} \tag{1.8b}$$

Where B is the effectiveness matrix and $B^{\dagger}$ is such that $BB^{\dagger} = \mathbb{I}$. Specifically $B^{\dagger}$ is the general *pseudo* inverse of B (more on inversions in Sec:4.3). It's assumed there's a multiplicative relationship between the input, $u \in \mathbb{U}$, and the input effectiveness matrix in Eq:1.4b. The controller designed actuator torque ${\tau_a}^b$ then dictates the input u as in Eq:1.8a. Much like the over-actuation previously discussed W.R.T quadcopters; the pseudo inversion method of actuator distribution applies quadratic optimization to the allocation slack cost function, Eq:1.5.

Chapter 2

Prototype Design

2.1 Design

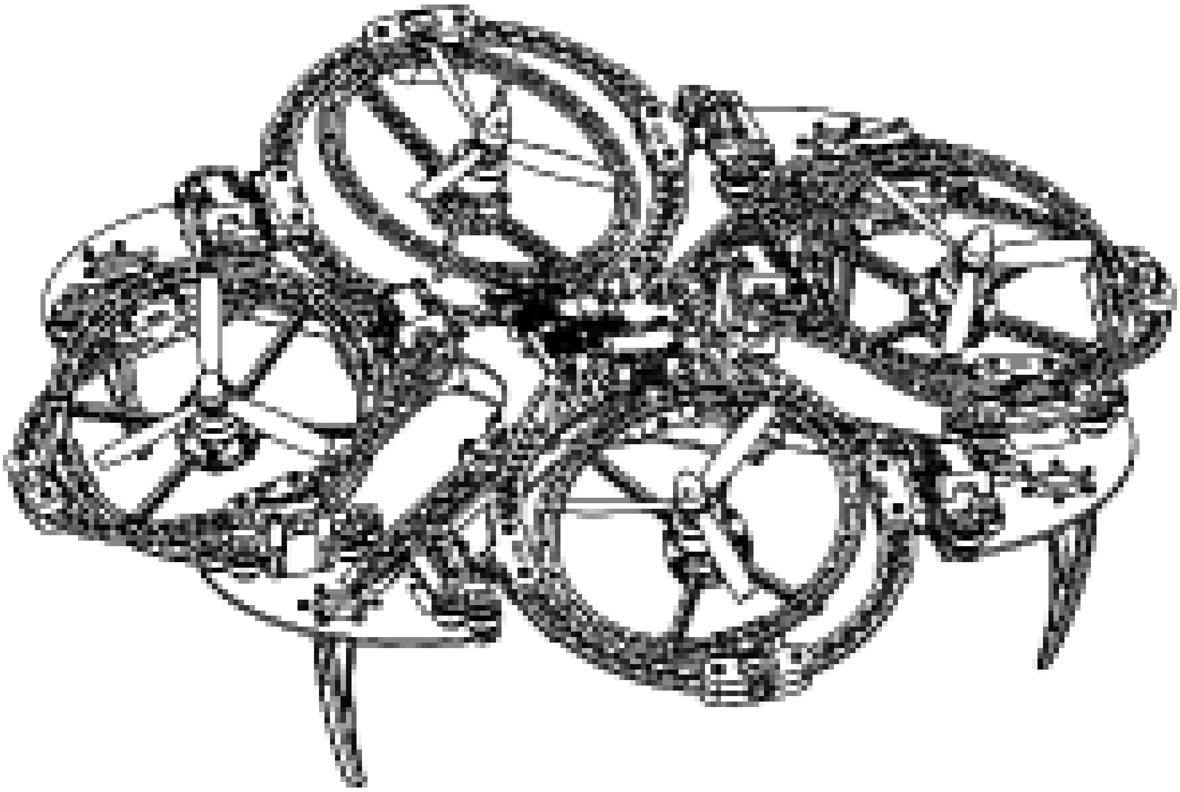
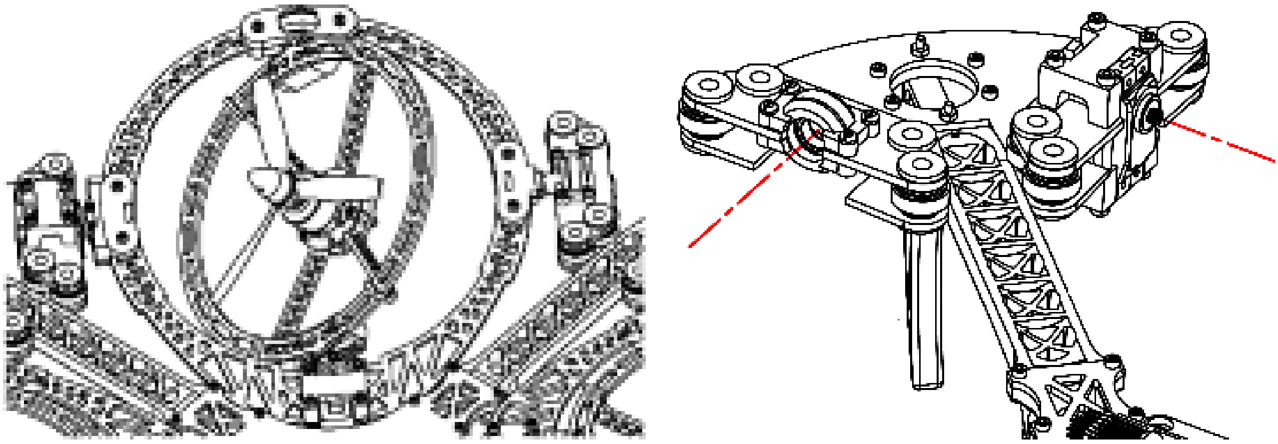


Figure 2.1: Isometric view of the prototype design

The final prototype (Fig:2.1) went through a series of different design iterations, all aimed at optimizing engineering time spent on construction and reducing the associated component costs thereof. A significant aspect of consideration for the design process was the net weight whose upper limit, as mentioned before, is inherently limited by the thrust produced from lift motors. Some of the more important design factors, like inertias & mass centers (Section:2.2.3), are discussed here in order to give context for the dynamics derived in the next chapter. The reference frame orientations which those dynamics are developed with respect to is then detailed as well. A brief overview of the electrical systems layout is given with the components associated and their electrical characteristics listed. Finally the actuator suite's functionality and transfer characteristics are also quantified. A review of the physical prototype realized and control loop implemented is detailed in Chapter:6 along with actual flight test results.

2.1.1 Actuation

The novel component of the design is the manner of articulation for each concentric gimbal ring which forms the motor modules. The design objective is to produce a thrust vectoring actuation set for a quadrotor's control plant. The outcome was a module which independently redirects the thrust generated by the lift propellers (Fig:2.2a). Within each module are servos affixed onto sequential support rings to pitch and roll the substructure's axes. The gyroscope-like frame that surrounds each motor/propeller pair accommodates that relative movement. Aligned with each servo is a coaxial support bearing. The bearing and actuator servos have a mass disparity which results in an eccentric center of mass, producing a gravitational torque arm. Unfortunately, due to weight constraints, counter balance measures cannot be introduced. Consequences from the center of mass variations must be either compensated for (*plant dependent solution*) or exploited in the dynamics (*additional non-linear actuator plants*). The precise effects are quantified numerically next in Section:2.2.3.



(a) Motor module assembly **(b)** Motor frame damping support assemblies

Figure 2.2

Each motor module is positioned such that its produced thrust vector coincides with the intersection of its two rotational axes. As a result, there's only a perpendicular displacement co-planar to the body frames X-Y-Z origin (See Fig:2.7), $\vec{L}_{arm}$. That length directly affects the differential thrust torque $\tau_{diff} = \vec{L}_{arm} \times \vec{T}$. An off-center thrust vector line would make that arm displacement a non-orthogonal vector. The center of gravity for each module is time varying and depends on its two servo rotational positions. It's more practical to ensure intersection of the thrust vector with the rotational center than to balance the masses undergoing rotation. A thrust varying torque is harder to approximate and hence compensate for than a gravitational torque, given the complexity with modeling a propeller's aerodynamic thrust (Section:3.3.1).

The primary body structure, similar to a traditional quadcopter '+' configuration, suspends each motor rotational assembly with silicon damping balls (Fig:2.2b). For damping to be effective there has to be roughly equatable relative masses between the two damped bodies. A smaller damping assembly in the center of the frame houses all the electronics and power distribution circuitry. All the mounting brackets which affix the motor module rings are 3D printed from CAD models using an Ultimaker V2+ [92]. There is a complete bill of materials for all parts used, including working drawings for each 3D printed bracket and the laser cut frame(s), in Appendix:B.

The propellers rotational plane is not aligned exactly with the plane made by the $\hat{X}_{M_i}$ and $\hat{Y}_{M_i}$ rotational servo axes (Fig:2.3). The offset is approximately 23.39 mm and must be considered when evaluating pitch/roll gyroscopic torque responses later in Section:3.2.1. The propellers are 6 inch (6×4.5) 3-Blade plastic Gemfam propellers, powered by Cobra CM2208-2000KV Brushless DC motors. The thrust produced as a function of angular velocity (in RPS) for the propellers is derived in Section:3.3.1.

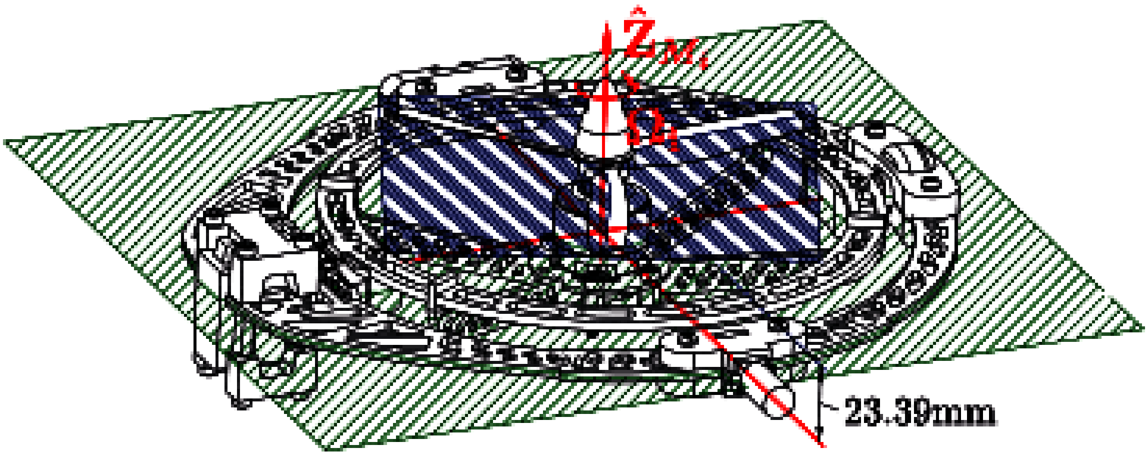
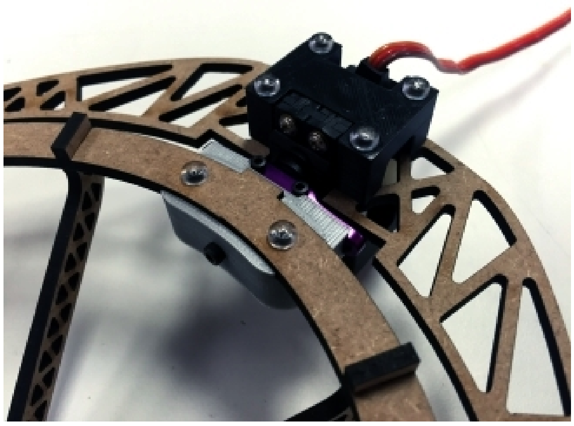


Figure 2.3: Difference between propeller and motor planes

The BLDC motors are controlled with LDPower 20A ESC[1] modules with an inline OrangeRx RPM Sensor. The transfer function for the combined unit is presented subsequently in Section:2.3.1. Power for the quadrotor is supplied not from a battery bank but from a power tether. Tethered power will ensure consistent flight time and reduce the concern of payload restriction on the available lift actuation. Power lines to both the BLDC motors and servos are supplied conventionally, however an ideal construction would see slip-rings for each module's power supply.

(a) Cobra CM2208-2000KV BLDC Motor Module



(b) Corona DS-339MG Servo Bracket

Metal gear Corona DS-339MG digital servos are used for the two axes of rotation (Fig:2.4b). Each servo has a range of 180°, positioned such that a zeroth offset aligns the motor modules, adjacent to the body frame, and has a ±90°range. A digital servo updates at 330 Hz, faster than a 50 Hz analogue servo equivalent (Table:2.1). This means the otherwise 20ms zero-order analogue sampling becomes a less significant 3.30ms zero-order holding time. Both the $\hat{X}_{M_i}$ and $\hat{Y}_{M_i}$ axis servos will be rotating a large loading mass and so their *open loop* plant dynamics are determined empirically in Section:2.3.1 using test data included in Appendix:C.

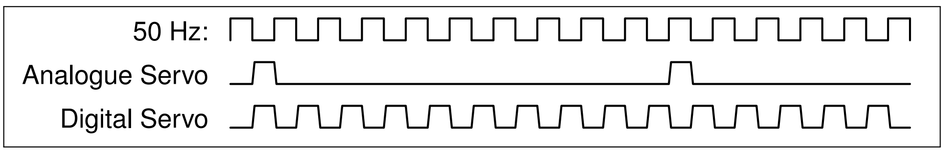


Table 2.1: Analogue & Digital Timing Signals

[1]Flashed with BLHeli [8] firmware

2.2 Conventions Used

The attitude conventions used for the system's dynamic derivations, in the following Chapter:3, are first briefly discussed here. Often these aspects are assumed to be obvious enough that they're omitted. It's important to clearly and unambiguously define a standard set of framing conventions to avoid uncertainty later. Rotation matrices are included but the focus is on the *contrast* between a rotation and transformation operation. Both [31] and [69] provide an in depth and thorough explanation of rotation matrices and DCM attitude representation if such concepts are unfamiliar to the reader. Quaternions are introduced in Section:3.1.3.

2.2.1 Reference Frames Convention

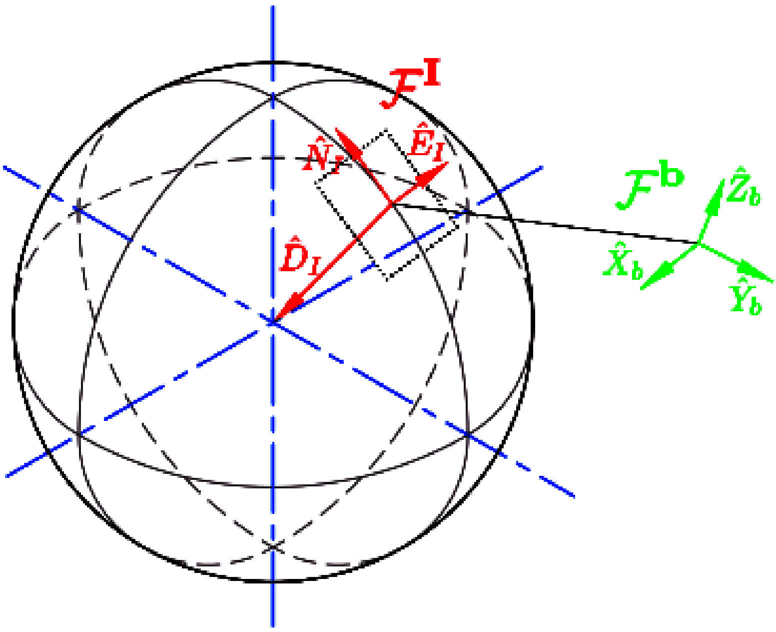


Figure 2.5: Inertial and Body Reference Frames

Euler (aerospace) frames are used for principle inertial and body coordinate representation (Fig:2.5). The inertial frame, $\mathcal{F}^I$, is aligned such that the $\hat{X}_I$ axis is in the $\hat{N}$orth direction, $\hat{Y}_I$ is in the $\hat{E}$ast direction and $\hat{Z}_I$ is in the $\hat{D}$ownward direction[2]. The body frame, $\mathcal{F}^b$, then has both $\hat{X}_b$ and $\hat{Y}_b$ aligned obliquely between two perpendicular arms of the quadrotor's body and the $\hat{Z}_b$ axis in the body's normal direction (Fig:2.8). The body frame's axes and their relation to the prototype design are highlighted next in Section:2.2.2. Frame superscripts I and b represent inertial and body frames respectively whilst vector subscripts imply the reference frame in which the vector's coordinates exists or taken relative to.

Relative angular displacement between two frames is commonly measured by the three angle Euler set. The Euler angles $\vec{\eta} = [\phi \;\; \theta \;\; \psi]^T$ represents rotations about the $\hat{X}$,$\hat{Y}$ and $\hat{Z}$ axes respectively. Depending on how the rotation sequence is formulated, those angles can be used to construct rotation matrices which give relation to vectors or can transform coordinates. The generic equation to *rotate* a vector $\vec{v}$ about a (normalized) axis $\hat{n}$ by some angle μ is given by[3]:

$$\vec{v}\,' = \big(1 - cos(\mu)\big)\big(\vec{v} \cdot \hat{n}\big)\hat{n} + cos(\mu)\vec{v} + sin(\mu)\big(\hat{n} \times \vec{v}\big) \tag{2.1}$$

Which, when $\hat{n}$ is either $\hat{X}$,$\hat{Y}$ or $\hat{Z}$ axes, can be simplified to produce the fundamental rotation matrices $\mathbb{R}_x(\phi)$,$\mathbb{R}_y(\theta)$ and $\mathbb{R}_z(\psi)$.

[2] In orbital sequences this would be toward the Earth's center. Sometimes referred to as the NÊD convention

[3] Derived and proven in *Quadrotor Dynamics and Control* [75]

Multiplication by a rotation matrix $\mathbb{R}(\cdot)$ applies a left-handed *rotation* operator, the resultant vector still exists in the same reference frame. An $\hat{X}$ axis rotation by ϕ is;

$$\vec{v}\,' = \mathbb{R}_x(\phi)\vec{v} \tag{2.2a}$$

$$\vec{v}\,', \vec{v} \in \mathcal{F}^1 \tag{2.2b}$$

No subscripts are used in Eq: 2.2 to indicate reference frame ownership because all vectors are in the same frame

A vector *transformation* changes the resultant vector's reference frame. Transformation is then a rotation by an angle of the *difference* between the resulting and principle reference frames. A transformation from frame $\mathcal{F}^1$ to $\mathcal{F}^2$, differing by an angle of ϕ about the $\hat{X}$ axis is then:

$$\vec{v}_2 = \mathbb{R}_x(-\phi)\vec{v}_1 \tag{2.3a}$$

$$\vec{v}_2 \in \mathcal{F}^2 \text{ and } \vec{v}_1 \in \mathcal{F}^1 \tag{2.3b}$$

The distinction between Eq:2.2 and Eq:2.3 is the directional sense of the angular operand ϕ, and hence the effect it has on the argument vector. The transformation or rotation of a vector from $\mathcal{F}^I$ to $\mathcal{F}^b$ is the product of three sequential operations about each axis. Each subsequent rotation is applied relative to a new intermediate frame; hence each Euler angle is taken relative to a specific intermediate frame. The sequence of axial rotation operations does indeed effect the Euler set. Any consequences of that chosen order is something discussed indepth in *Quaternions and Rotation Sequence*, [48]. In this dissertation the Z-Y-X sequence is used. A transformation of the vector $\vec{v}$ from the inertial to the body frame is then applied by:

$$\mathbb{R}_I^b \triangleq \mathbb{R}_z(\psi)\mathbb{R}_y(\theta)\mathbb{R}_x(\phi) \tag{2.4a}$$

$$\vec{v}_b = \mathbb{R}_I^b(-\psi, -\theta, -\phi)\vec{v}_I \tag{2.4b}$$

$$\Rightarrow \vec{v}_b = \mathbb{R}_z(-\psi)\mathbb{R}_y(-\theta)\mathbb{R}_x(-\phi)\vec{v}_I \tag{2.4c}$$

$$\mathbb{R}_z(-\psi)\mathbb{R}_y(-\theta)\mathbb{R}_x(-\phi) \iff \mathbb{R}_x(\phi)\mathbb{R}_y(\theta)\mathbb{R}_z(\psi) = \mathbb{R}_b^I \tag{2.4d}$$

$$\mathbb{R}_I^b = \left(\mathbb{R}_b^I\right)^{-1} = \left(\mathbb{R}_b^I\right)^T \tag{2.4e}$$

The relationship in Eq:2.4d is an inversion (*transpose*) of the rotation matrix. A rotation matrix's inverse can be used interchangeably with its negative counterpart to maintain a positive sense of the argument angle. To ensure clarity throughout this dissertation's mathematics, a negative angular sense implies a *transformation* to a different reference frame. Where applicable, the order of rotation will indicate the sequence direction and an angular sign differentiates the rotation and transformation operators.

The body frame's angular velocity is taken relative to the inertial frame, represented by $\vec{\omega}_{b/I} \Rightarrow \vec{\omega}_b$. Seeing that each Euler angle is measured with respect to an intermediary frame, a distinction must then be made between $\dot{\eta}$ and $\vec{\omega}_b$. All three Euler angles need to be transformed to one common frame. Exploiting vehicle frames 1 & 2, or rather $\mathcal{F}^{v1}$ & $\mathcal{F}^{v2}$, as intermediate frames to respectively describe post $\mathbb{R}_x(\phi)$ and $\mathbb{R}_y(\theta)$ operations.

$$\vec{\omega}_b = \frac{d}{dt_b}\eta = \frac{d\phi}{dt}\mathbb{R}_{v2}^b(\phi)\begin{bmatrix}\phi\\0\\0\end{bmatrix} + \frac{d\theta}{dt}\mathbb{R}_{v2}^b(\phi)\mathbb{R}_{v1}^{v2}(\theta)\begin{bmatrix}0\\\theta\\0\end{bmatrix} + \frac{d\psi}{dt}\mathbb{R}_{v2}^b(\phi)\mathbb{R}_{v1}^{v2}(\theta)\mathbb{R}_I^{v1}(\psi)\begin{bmatrix}0\\0\\\psi\end{bmatrix} \tag{2.5a}$$

The vehicle frames in Eq:2.5a and the subsequent rotations between each frame don't necessarily have to be in that order. The equation could change depending on what rotation sequence was used.

Which then simplifies to the formal relationship between two rotating frames, with $\vec{\omega}_b = [p\ q\ r]^T$ in $rad.s^{-1}$:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -sin(\theta) \\ 0 & cos(\phi) & sin(\phi)cos(\theta) \\ 0 & -sin(\theta) & cos(\phi)sin(\theta) \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{2.5b}$$

$$\Rightarrow \vec{\omega}_b = \Psi(\eta)\dot{\eta} \tag{2.5c}$$

$$\Psi(\eta) = \begin{bmatrix} 1 & 0 & -sin(\theta) \\ 0 & cos(\phi) & sin(\phi)cos(\theta) \\ 0 & -sin(\theta) & cos(\phi)sin(\theta) \end{bmatrix} \tag{2.5d}$$

$$\Rightarrow \dot{\eta} = \Psi^{-1}(\eta)\vec{\omega}_b = \Phi(\eta)\vec{\omega}_b \tag{2.5e}$$

$$\Phi(\mathcal{E}) = \begin{bmatrix} 1 & sin(\phi)tan(\theta) & cos(\phi)tan(\theta) \\ 0 & cos(\phi) & -sin(\phi) \\ 0 & sin(\phi)sec(\theta) & cos(\phi)sec(\theta) \end{bmatrix} \tag{2.5f}$$

The termed *Euler* matrix, $\Phi(\eta)$, contains a well known and problematic singularity at $\theta = \pm\pi/2$; because $sec(\theta) \to \infty$ as $\theta \to \pi/2$. The effect of the rotation matrix singularity is further explored later in Section:3.1.2. It's manifestation in the θ angle here is a direct consequence of the Z-Y-X sequence used. Each Euler angle can potentially suffer a singularity depending on how the rotations are sequenced. Indeed quaternions are used for kinematics later in lieu of Euler angles. Euler angular attitude representation is, however, easily understood and well suited to the conventional distinctions made in this Chapter.

Quaternion operations are similarly sequenced in the Z-Y-X order:

$$\mathbb{R}_I^b \iff Q_b \otimes (.) \otimes Q_b^* \tag{2.6a}$$

$$Q_b \triangleq Q_z Q_y Q_x \text{ and } Q_b \triangleq Q_x^* Q_y^* Q_z^* \tag{2.6b}$$

With $\otimes$ being the Hamilton product (or quaternion multiplication). Each quaternion, Q_i, is a unit quaternion about that $\hat{i}^{th}$ axis. It is important to note that a quaternion rotation operates on an argument vector with a zero quaternion scalar component. So then for some vector $\vec{v}$, the quaternion rotation operation in Eq:2.6a is equivalent to;

$$Q_{\vec{v}}\ ' = Q \otimes (Q_{\vec{v}}) \otimes Q^* \tag{2.7a}$$

$$\text{Where } Q_{\vec{v}} = \begin{bmatrix} 0 \\ \vec{v} \end{bmatrix},\ Q_{\vec{v}}\ ' = \begin{bmatrix} 0 \\ \vec{v}\ ' \end{bmatrix} \tag{2.7b}$$

The quaternion representation in Eq:2.7b ensures that the operation is entirely in $\mathbb{R}^4$ space. However it is usually omitted, despite $\mathbb{R}^4$ being implied and as such, Eq:2.7a is then simply:

$$\vec{v}\ ' = Q \otimes (\vec{v}) \otimes Q^* \tag{2.8}$$

Quaternion dynamics, and the quaternion operator, are later expanded upon to replace the use of Euler angles and Rotation matrices as a convention for attitude representation later in Chapter:3

2.2.2 Motor Axis Layout

Fundamentally the whole structure, although treated as fixed and rigid in the kinematics, consists of multiple rigid bodies with relative rotations to one another, illustrated previously in the design description in Section:2.1. Those rigid bodies are divided into four inter-connected motor modules and a single body structure. Each module consists of two sequential gimbal rings, each with one degree of relative rotation between itself and the next subsequent ring. There needs to be distinct nomenclature used for describing these motor modules.

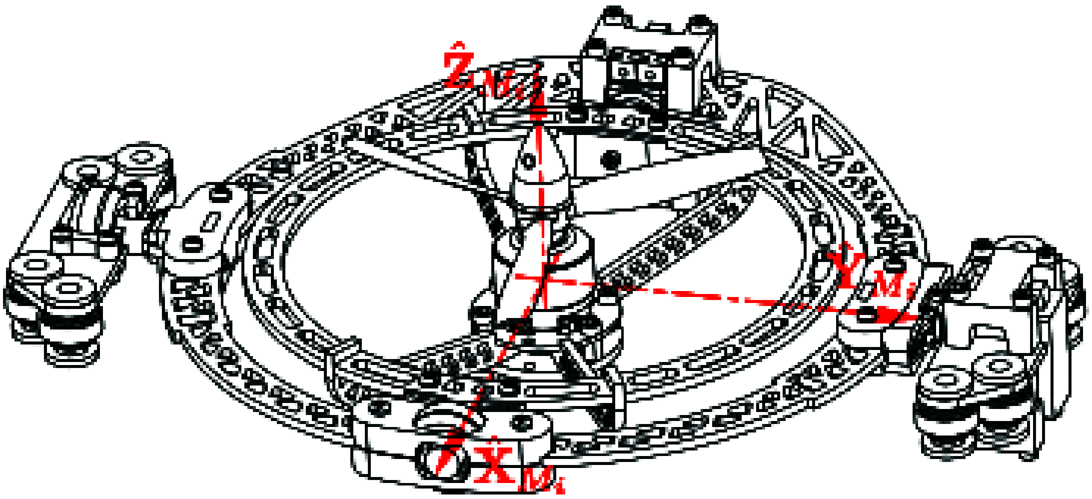


Figure 2.6: Aligned Motor Frame Axes

Every propeller/motor pair is actuated by two servos. The i^{th} propeller, directly driven by the motor's rotor, has a rotational speed Ω_i $[RPS]$ about the $\hat{Z}$ stator axis. Two servos are aligned *at rest* with $\hat{Y}$ and $\hat{X}$ axes to pitch and roll the propeller away from its principle rotational axis. Each motor has its own reference frame, $\mathcal{F}^{M_i}$, aligned in Fig:2.6 and highlighted with the rotational rings in Fig:2.7.

Motor frames, numbered $1-4$, transform to the body frame first by an angle of $\lambda_i°$ about the $\hat{X}_{M_i}$ axis. Then by $\alpha_i°$ about the $\hat{Y}_{M_i'}$ axis in an intermediate M_i' frame. The first servo actuates λ_i, rotating $\mathcal{F}^{M_i}$ to an intermediate $\mathcal{F}^{M_i'}$ frame. Secondly, the next servo actuates α_i to produce a second intermediate frame M_i''. That second servo is affixed in the M_i'' frame. Lastly there's a relative orthogonal rotation about $\hat{Z}_{M_i''}$ between $\mathcal{F}^b$ and $\mathcal{F}^{M_i''}$. Each module's actuation state is fully described by $[\Omega_i,\ \lambda_i,\ \alpha_i]^T$ for $i \in [1 : 4]$. The four motor modules are aligned relative to the body's XYZ axes as shown in Fig:2.8. Modules 1 and 3 have their X-axes in the positive and negative $\hat{X}$ directions of the body frame respectively. Similarly Modules 2 and 4 have their X-axes in the positive and negative $\hat{Y}$ directions of the body frame.

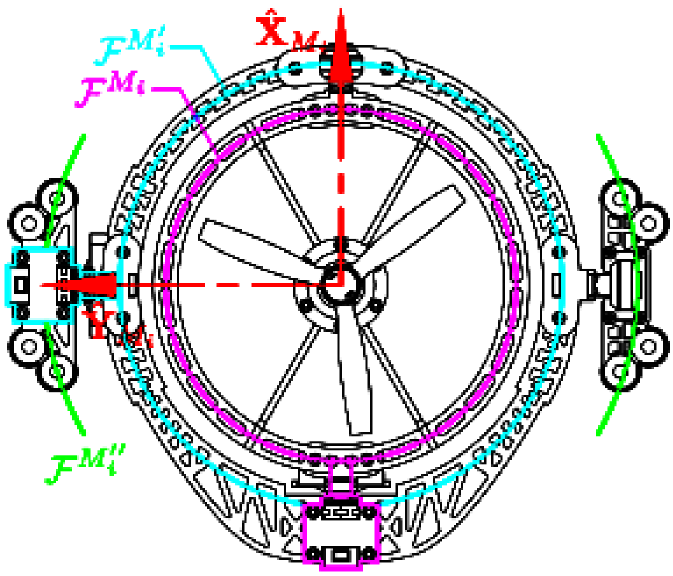


Figure 2.7: Intermediate Motor Frames

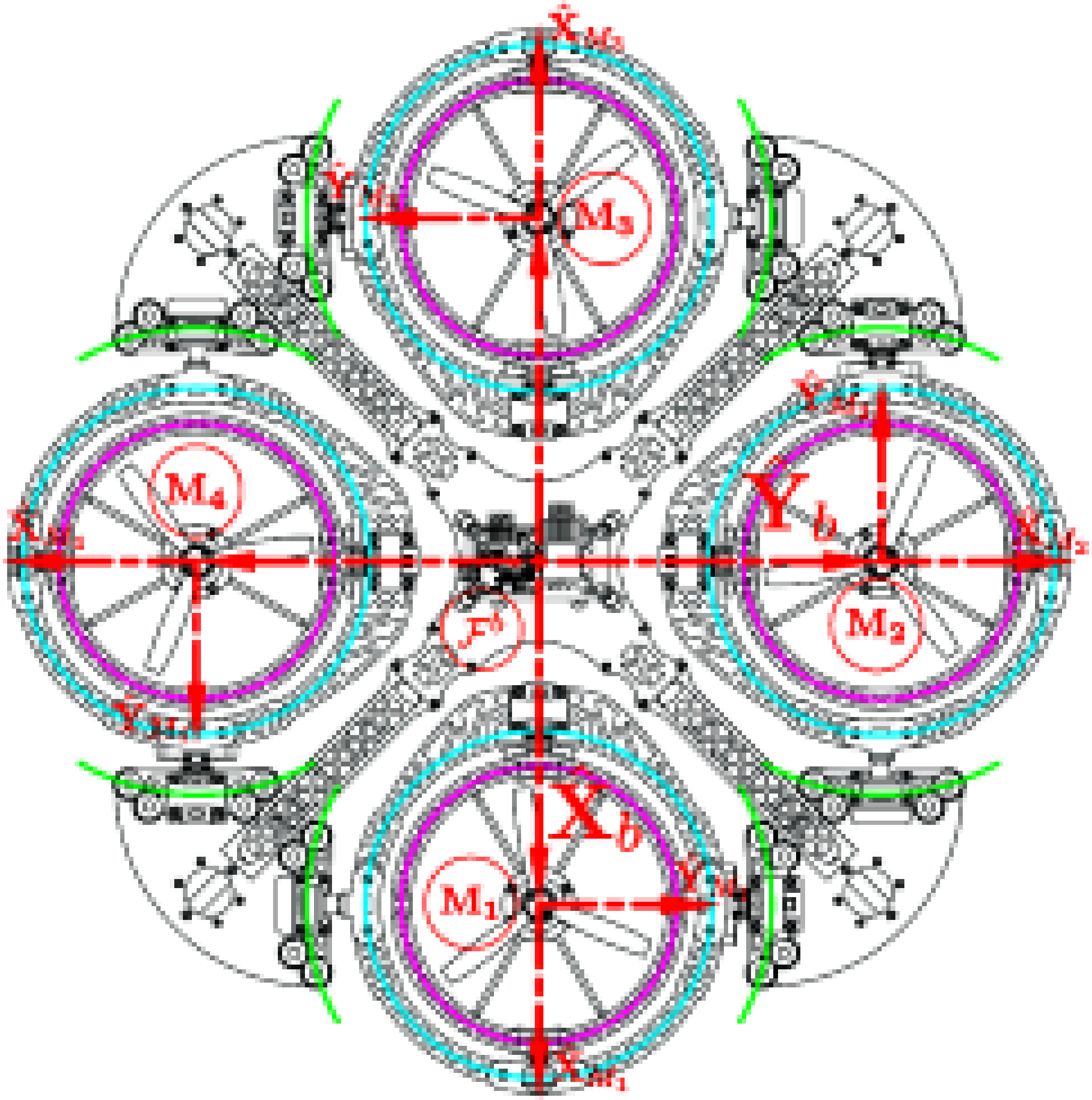


Figure 2.8: Body Frame Axes Layout

Not shown in Fig:2.8 is the relative $\hat{Z}$ axis position with respect to the structure. The $\hat{Z}$ height of the body's motion centroid is such that its origin is co-planar with the four motor modules rotational centers. The center of motion is <u>*not*</u> *the center of mass, an aspect which is investigated next in Section:2.2.3.*

Vector transformations from each of the motor frames to the body frame are characterized as:

$$\vec{v}_b = \mathbb{R}_z(-\sigma_i)\mathbb{R}_y(-\alpha_i)\mathbb{R}_x(-\lambda_i)\vec{v}_{M_i},\ \ \sigma_i \in [0, 90°, 180°, 270°] \tag{2.9a}$$

With orthogonal rotation matrices $\mathbb{R}_z$:

$$\mathbb{R}_z = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ for } i \in [1, 2, 3, 4] \text{ respectively} \tag{2.9b}$$

The entire actuator space, including propeller speed Ω_i $[RPS]$, is then $\in \mathbb{R}^{12}$, or rather $\mathbb{U} \in \mathbb{R}^{12}$. The actuator input set $u \in \mathbb{U}$ is then structured as:

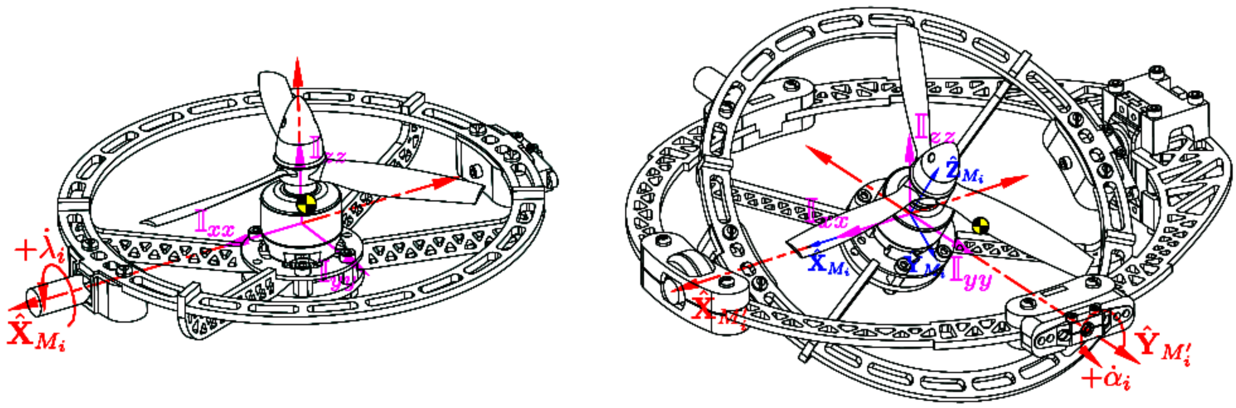
$$u_{\in\mathbb{U}} = \begin{bmatrix}\Omega_1\ \lambda_1\ \alpha_1\ \ldots\ \Omega_4\ \lambda_4\ \alpha_4\end{bmatrix}^T \tag{2.10}$$

2.2.3 Inertial Matrices & Mass

Although inertias are presented here rounded to either 2 or 0 decimal places, full floating point numbers are used in simulation and prototype software. Un-rounded inertias are included in Appendix:D. Similarly rotation matrices produce a more cumbersome results for Eq:2.12,2.15,2.17, which are all susceptible to singularities. Quaternion transformations between rotating reference frames are used in practice.

Inertias

An undesirable side effect of the relative rigid body rotations within the structure are the inertial responses produced associated with such movements. Given Newton's Second Law of Rotational Motion†, each applied rotation is going to produce an equal but opposite reaction onto the principle inducing frame. Similarly a gyroscopic cross product from rotational velocities is also present. Such first and second order effects are often neglected given that the angular rates which they're dependent on are mostly small enough to approximate as zero, $\vec{\omega}_b \approx \vec{0}$. A dynamic set-point (non-zero) attitude tracking plant is, however, going to produce sizeable time varying body angular velocities and accelerations. Unlike a traditionally actuated quadrotor, such effects have to be accounted for.



(a) Inner Ring Rotational Structure **(b)** Middle Ring Rotational Structure

Figure 2.9: Inertial Measurement References

The manifestation of the aforementioned torques are explored in thorough detail in Section:3.2. Those effects are both dependent on the rotational body's inertial tensor[4] about each respective rotational axis. The magnitude of those inertias are obviously a by-product of the structure's design. Starting with the innermost assembly, in each Motor Frame $\mathcal{F}^{M_i}$, the inner ring structure is a 92g assembly (all components incorporated). The rotational center *roughly* coincides with the center of its mass ($C.M = [-1.44\ 0\ 5.81]^T\ \ [mm]$ relative to its rotational center). The inner ring being rotated by $\lambda_i{}^\circ$ about the $\hat{X}_{M_i}$ axis then has an inertial matrix (centered and aligned with axes as in Fig:2.9a):

$$\mathbb{I}_{M_i} = \begin{bmatrix} 561.96 & -32.29 & -0.26 \\ -32.29 & 1888.74 & 0.00 \\ -0.26 & 0.00 & 2090.97 \end{bmatrix} \quad [g.cm^2] \tag{2.11a}$$

$$\approx diag\big(562,\ 1889,\ 2091\big) \times 10^{-7} \quad [kg.m^2] \tag{2.11b}$$

The effect of rapidly spinning propellers on the inertia in Eq:2.11a is approximated well by a solid disc, hence the inner ring's inertial components are regarded as constant. The moment of inertia about that $\hat{X}_{M_i}$ rotational axis, pertinent to a λ_i rotation, is then $\mathbb{I}_\lambda \approx 531 \times 10^{-7}\ \ [kg.m^2]$.

[4]All inertias are assumed symmetrical and calculated in Solidworks with overridden masses to match physical prototype measurements, all those values are included in Appendix:B

The first λ_i actuating servo and bearing supports are affixed to the intermediate middle ring assembly (Fig:2.9b). The middle ring frame, $\mathcal{F}^{M_i'}$, is a $98g$ structure, excluding the inner most ring. Collectively the mass for both the inner and middle rings structures is $m_{module} = 190g$. The middle ring is rotated by $\alpha_i°$ about its $\hat{Y}_{M_i'}$ axis. The compound body's inertia about that axis of rotation, $\hat{Y}_{M_i}$, is a combination of both the middle ring's inertia and the inner ring's. The latter contribution being a function of the *rotation* (not transformation) angle $\lambda_i°$ which, from the conservation of angular momentum theory [86][5], is:

$$\text{If } \mathbb{I}_{middle} = \begin{bmatrix} 2905.70 & 0.02 & 390.89 \\ 0.02 & 8446.41 & 0.01 \\ 390.89 & 0.01 & 11125.74 \end{bmatrix} \quad [g.cm^2] \tag{2.12a}$$

$$\mathbb{I}_{M_i'} = \mathbb{I}_{middle} + \mathbb{R}_x(\lambda_i)\big(\mathbb{I}_{inner}\big)\mathbb{R}_x^{-1}(\lambda_i) \tag{2.12b}$$

$$\mathbb{I}_{M_i'}(\lambda_i) = \mathbb{I}_{const} + \mathbb{I}_{M_i}(\lambda_i) \tag{2.12c}$$

$$\approx \begin{bmatrix} 3468 & 0 & 391 \\ 0 & 10436 & 0 \\ 391 & 0 & 13155 \end{bmatrix} + \begin{bmatrix} 0 & -32c_\lambda & -32s_\lambda \\ -32c_\lambda & -101c_{2\lambda} & 101s_{2\lambda} \\ -32s_\lambda & 101s_{2\lambda} & 101c_{2\lambda} \end{bmatrix} \times 10^{-7} \quad [kg.m^2] \tag{2.12d}$$

With $\mathbb{I}_{inner} = \mathbb{I}_{M_i}$ being the inertia from Eq:2.11a, re-orientated with a rotation $\mathbb{R}_x(\lambda_i)$. The net inertia is then a function of the rotation angle λ_i and a constant inertia (Eq:2.12c) which is then simplified[6] to Eq:2.12d. It's important to note the two non-zero products of inertia, $\mathbb{I}_{yx}$ and $\mathbb{I}_{yz}$, which are going to result in a vector $\vec{\tau}_\eta$ response. The inertia then encountered by an α_i rotation is:

$$\mathbb{I}_\eta(\lambda) \approx [-32c_\lambda,\ 10436 - 101c_{2\lambda},\ 101s_{2\lambda}]^T \times 10^{-7} \quad [kg.m^2] \tag{2.13}$$

Variable inertias dependent on state input variables are the first of many non-trivial aspects unique to this aircraft's design. The resultant control solutions are thus decidedly plant dependent in their formulation. Secondly, the center of mass for the motor module's compound assembly isn't coincidental with either rotational axes intersection. As a result the effective center of mass for the entire structure is going to be a function of the angular rotational position of each motor module and time varying.

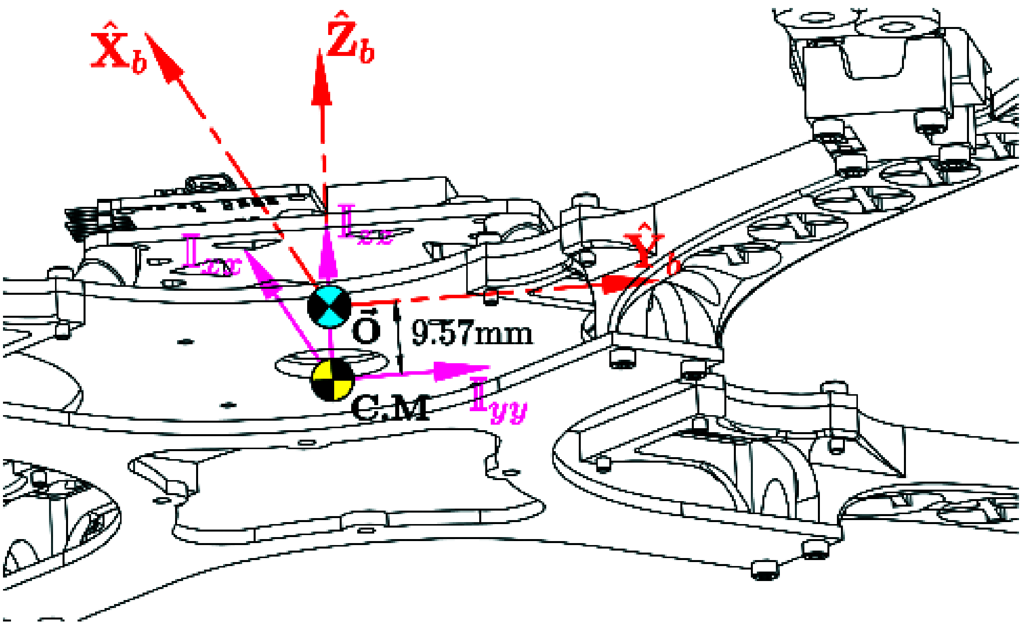


Figure 2.10: Body Frame Center of Mass

[5] $\mathbb{R}_x$ is a full rank and square, so an inverse $\mathbb{R}_x^{-1}$ always exists

[6] Eq:2.12d is rounded to no decimal places, seeing that its units are already $\times 10^{-7}$ and thus some products of inertia are omitted.

The second α_i rotating servo adjoins the complete motor module (both the inner and middle ring assemblies) to the body structure. The inertial volume of the servo and bearing supports contribute then to the body structure's inertia, whose value excludes any of the four motor modules. Consisting of servo and bearing damping brackets, each "damping" assembly collectively weighs $84g$ and suspends the motor modules from the body frame with a set of silicon damping balls. The body structure's center of mass (without motor modules, Fig:2.10) coincides with the XY directional axes and lies $\Delta Z = -9.57\ mm$ below the Body Frame's origin of motion, $\vec{O} \in \mathcal{F}^b$.
Note: the origin which all motion is calculated with respect to is co-planar to the motor module's rotational centers, not the net center of mass.
The body's weight, including all four damping assemblies and electronics, totals to $814.7\ g$. The body's net inertia (*sans* motor modules) $\mathbb{I}_{body}$, about its center of mass is (Fig:2.10)

$$\mathbb{I}_{body} = \begin{bmatrix} 181689.67 & -0.44 & -8.86 \\ -0.44 & 181567.22 & -19.44 \\ -8.86 & -19.44 & 360077.58 \end{bmatrix} \times 10^{-7} \quad [kg.m^2] \tag{2.14a}$$

Using the Parallel Axis theorem†, that same net body inertia about the body frame's origin, $\vec{O}_b$, is:

$$\mathbb{I}_{body}' = \mathbb{I}_{body} + m(\vec{d} \cdot \vec{d} + \vec{d} \otimes \vec{d}) \approx \mathbb{I}_{body} + md^2 \tag{2.14b}$$

Here $\otimes$ represents the Hamilton product of two 3X3 matrices, it's used again later in Chapter:3 to indicate the quaternion multiplication operator.

$$\underset{\vec{\mathbf{O}}}{\mathbb{I}_{body}'} = \begin{bmatrix} 182435.66 & -0.42 & -6.46 \\ -0.42 & 182313.18 & -14.52 \\ -6.46 & -10.41 & 360077.62 \end{bmatrix} \times 10^{-7}\ [kg.m^2] \tag{2.14c}$$

Net inertia for the compound assembly, $\mathbb{I}_b$[7], about the origin $\vec{O}_b$ is a combination of all the relative attached bodies. That being; the four motor modules, transformed and then translated to the center of motion, and the body structure itself. That transformation is analogous to that of Eq: 2.9. Reiterating that the the origin is co-planar to the module's center of rotation, each motor module's inertia, $\mathbb{I}_{M_i'}$[8], is further rotated by $\alpha_i°$ about $\hat{Y}_{M_i'}$ and finally an orthogonal $\hat{Z}_{M_i''}$ rotation (aligned with $\hat{Z}_b$) onto $\mathcal{F}^b$. Still measured with respect to their individual rotational centers, $\vec{\mathbf{M}}_i$, but re-orientated to align with $||\vec{\mathbf{O}}_b$. Contribution of each motor module's inertia, with $\mathbb{R}_z$ being the same as Eq:2.9b, is then:

$$\mathbb{I}_{i^{th}motor} = \mathbb{R}_z(\sigma_i)\mathbb{R}_z(\alpha_i)\big(\mathbb{I}_{M_i'}(\lambda_i)\big)\mathbb{R}_y^{-1}(\alpha_i)\mathbb{R}_z^{-1}(\sigma_i) \tag{2.15a}$$

Expanding to Inner and Middle Ring components:

$$= \mathbb{R}_z\mathbb{R}_y(\alpha_i)\big(\mathbb{I}_{middle}\big)\mathbb{R}_y^{-1}(\alpha_i)\mathbb{R}_z^{-1} + \mathbb{R}_z\mathbb{R}_y(\alpha_i)\mathbb{R}_x(\lambda_i)\big(\mathbb{I}_{inner}\big)\mathbb{R}_x^{-1}(\lambda_i)\mathbb{R}_y^{-1}(\alpha_i)\mathbb{R}_z^{-1} \tag{2.15b}$$

$$\text{With axes } \hat{X} \in \mathcal{F}^{M_i},\quad \hat{Y} \in \mathcal{F}^{M_i'},\quad \hat{Z} \in \mathcal{F}^{M_i''} \tag{2.15c}$$

It's at this stage that, despite simplifications, the symbolic inertial equation becomes overly cumbersome to include with numeric values... For the sake of brevity, exact calculated inertial values for the input dependent plant are omitted.

Each module's rotational center ([±195.16 0 0] & [0 ± 195.16 0] recalling Fig:2.8) are spaced equally relative to $\vec{\mathbf{O}}_b$ with a parallel axis arm $\vec{L}_{arm} = \begin{bmatrix}195.16 & 0 & 0\end{bmatrix}^T$ $[mm]$ (Fig:2.11). The net inertial equation about $\vec{\mathbf{O}}_b$, dependent on the actuator suite $\mathbb{U}$ positions, can be calculated as:

$$\underset{u \in \mathbb{U}}{\mathbb{I}_b(u)} = \mathbb{I}_{body} + \sum_{i=1}^{4} \mathbb{M}_i \quad [kg.m^2] \tag{2.16a}$$

[7]Disambiguation: $\mathbb{I}_b$ is *net* body frame's inertia, different from $\mathbb{I}_{body}$ which is the inertia for *just* the body structure
[8]As defined in Eq:2.12d

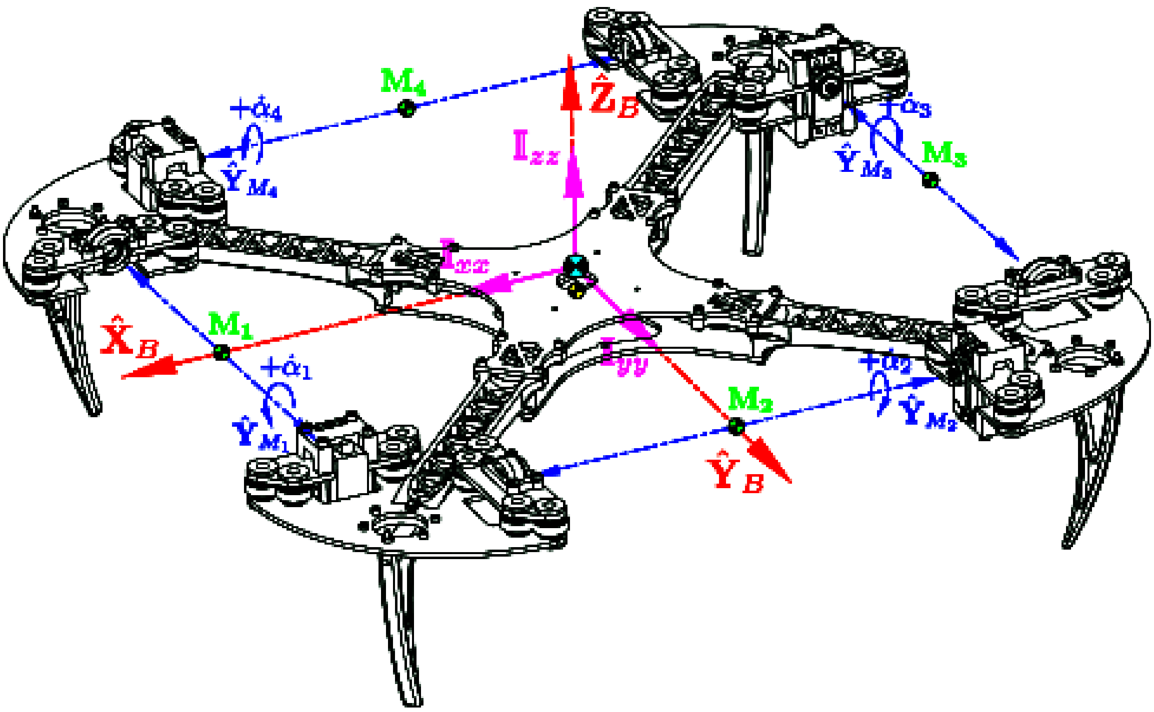


Figure 2.11: Inertial Center & Mass Center

$$\mathbb{M}_i = \mathbb{I}_{i^{th}motor} + m_{module}\big(\vec{L}\cdot\vec{L} - \vec{L}\otimes\vec{L}\big) \tag{2.16b}$$

Although Eq:2.16 does indeed produce the net body's inertia, the transformations to calculate $\mathbb{M}_i$ are compounded. Motor module inertias are first translated to their centers of rotation from their respective center of masses and then finally to the body frame's origin. Subsequent transformations are successively going to deteriorate the floating point precision of the resultant inertial tensor. Transforming inertial tensors about each sub-body's center of mass directly to the body frame origin will improve the reliability of the produced inertial equations. It is perhaps more intuitive to consider each sub-body's contribution individually, despite having been derived as combined inertial systems previously.

$$\underset{u\in\mathbb{U}}{\mathbb{I}_b(u)} = \mathbb{I}_{body} + \sum_{i=1}^{4}\mathbb{M}_{inner} + \sum_{i=1}^{4}\mathbb{M}_{middle} \tag{2.17}$$

The relative movement pertinent to Eq:2.11 and Eq:2.12 are separate from those affecting Eq:2.16. For each inner ring, W.R.T its center of mass measured relative to its center of rotation, different from Eq:2.11a, the inner ring's inertia is calculated as;

$$m_{inner} = 92 \quad [g] \tag{2.18a}$$

$$\underset{C.M}{\mathbb{I}_{inner}} = \begin{bmatrix} 530.88 & -32.29 & 7.46 \\ -32.29 & 1855.74 & 0 \\ 7.46 & 0 & 2088.87 \end{bmatrix} \quad [g.cm^2] \tag{2.18b}$$

$$C.M_{inner} = \begin{bmatrix} -1.44 & 0 & 5.81 \end{bmatrix}^T \quad [mm] \tag{2.18c}$$

$$C.M_{inner}{}' = \mathbb{R}_z\mathbb{R}_y(\alpha_i)\mathbb{R}_x(\lambda_i)\big(C.M_{inner}\big) \tag{2.18d}$$

$$\underset{\|\,\vec{\mathbf{O}}}{\mathbb{I}_{inner}} = \mathbb{R}_z\mathbb{R}_y(\alpha_i)\mathbb{R}_x(\lambda_i)\big(\mathbb{I}_{inner}\big)\mathbb{R}_x^{-1}(\lambda_i)\mathbb{R}_y^{-1}(\alpha_i)\mathbb{R}_z^{-1} \tag{2.18e}$$

$$\Delta L = \vec{L}_{arm} - C.M_{inner}{}' \tag{2.18f}$$

$$\mathbb{M}_{inner} = \underset{\vec{\mathbf{O}}}{\mathbb{I}_{inner}} = \underset{\|\,\vec{\mathbf{O}}}{\mathbb{I}_{inner}} + m_{inner}\big((\Delta L\cdot\Delta L)\mathbb{I}_{3x3} - \Delta L\otimes\Delta L\big) \tag{2.18g}$$

Similarly for the middle rings:

$$m_{middle} = 98 \quad [g] \tag{2.19a}$$

$$\underset{C.M}{\mathbb{I}_{middle}} = \begin{bmatrix} 2879.06 & 172.29 & 223.58 \\ 172.29 & 6268.97 & 13.33 \\ 223.58 & 13.33 & 8947.52 \end{bmatrix} \quad [g.cm^2] \tag{2.19b}$$

$$C.M_{middle} = \begin{bmatrix} -47.00 & 3.74 & -3.63 \end{bmatrix}^T \quad [mm] \tag{2.19c}$$

$$C.M_{middle}{}' = \mathbb{R}_z \mathbb{R}_y(\alpha_i)\big(C.M_{middle}\big) \tag{2.19d}$$

$$\underset{||\vec{\mathbf{O}}}{\mathbb{I}_{middle}} = \mathbb{R}_z \mathbb{R}_y(\alpha_i)\big(\mathbb{I}_{middle}\big)\mathbb{R}_y^{-1}(\alpha_i)\mathbb{R}_z^{-1} \tag{2.19e}$$

$$\Delta L = \vec{L}_{arm} - C.M_{middle}{}' \tag{2.19f}$$

$$\mathbb{M}_{middle} = \underset{\vec{\mathbf{O}}}{\mathbb{I}_{middle}} = \underset{||\vec{\mathbf{O}}}{\mathbb{I}_{middle}} + m_{middle}\big((\Delta L \cdot \Delta L)\mathbb{I}_{3x3} - \Delta L \otimes \Delta L\big) \tag{2.19g}$$

Unless otherwise specified; any inertia $\mathbb{I}_b(u)$, irrespective of arguments, will refer to an instantaneous calculated solution to Eq:2.17 given a particular $u(t) \in \mathbb{U}$. The purpose of the derivations Eq:2.18 & Eq:2.19 is twofold; highlighting both the inertial contributions and the variable center of masses for each sub-body. Seeing that the origin of the motion frame $\mathcal{F}^b$ and the net body's center of mass aren't coincidental, it's important to quantify the equation for the varying center of mass. If, for a collection of n bodies, with each body's center $\vec{X}_i$ and a mass m_i, the net center of mass is:

$$C.M = \frac{\sum_{i=1}^{n} m_i.\vec{X}_i}{\sum_{i=1}^{n} m_i} \tag{2.20a}$$

Such that, with $\vec{X}_{inner}$ & $\vec{X}_{middle}$ being rotated centers of mass defined in Eq:2.18d & Eq:2.19d respectively, the entire assembly has a center of mass†:

$$C.M(u) = \frac{m_{body}.\vec{X}_{body} + \sum m_{inner}.\vec{X}_{inner} + \sum m_{middle}.\vec{X}_{middle}}{m_{body} + \sum m_{inner} + \sum m_{middle}} \tag{2.20b}$$

Making the resultant gravitational torque[9] about the origin $\vec{\mathbf{O}}$ at any given moment:

$$\Delta C.G = \vec{\mathbf{O}} - C.M \tag{2.20c}$$

$$\tau_g = \Delta C.G \times \vec{G}_b \quad [N.m], \tau_g \in \mathcal{F}^b \tag{2.20d}$$

The net mass for the whole assembly is 1574 g. For reference the center of gravity when all actuators are at their zero positions is: $[-0.02 \;\; -0.03 \;\; -4.5]$ $[mm]$. Then, according to Eq:2.17, the inertial tensor for the net assembly at the rest conditions, $u = \vec{0}$, about the origin $\vec{\mathbf{O}}$ is:

$$\mathbb{I}_b(\vec{0}) = \begin{bmatrix} 317784.78 & -0.42 & -6.46 \\ -0.42 & 317662.31 & -14.52 \\ -6.46 & -14.52 & 628430.75 \end{bmatrix} \quad [g.cm^2] \tag{2.21}$$

[9] With $\vec{G}_b = \mathbb{R}_I^b \vec{F}_g \quad [N]$

2.3 Electronics

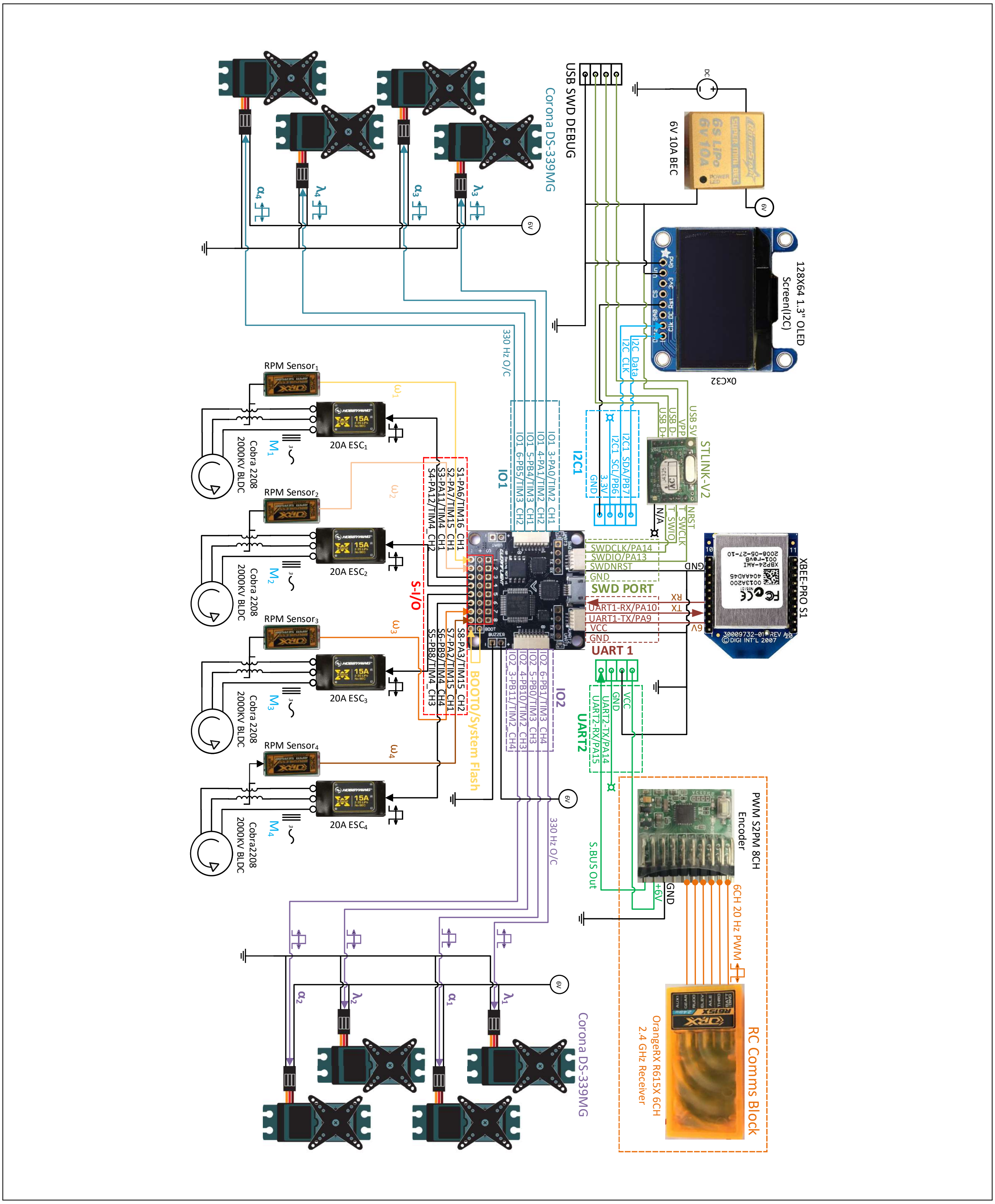
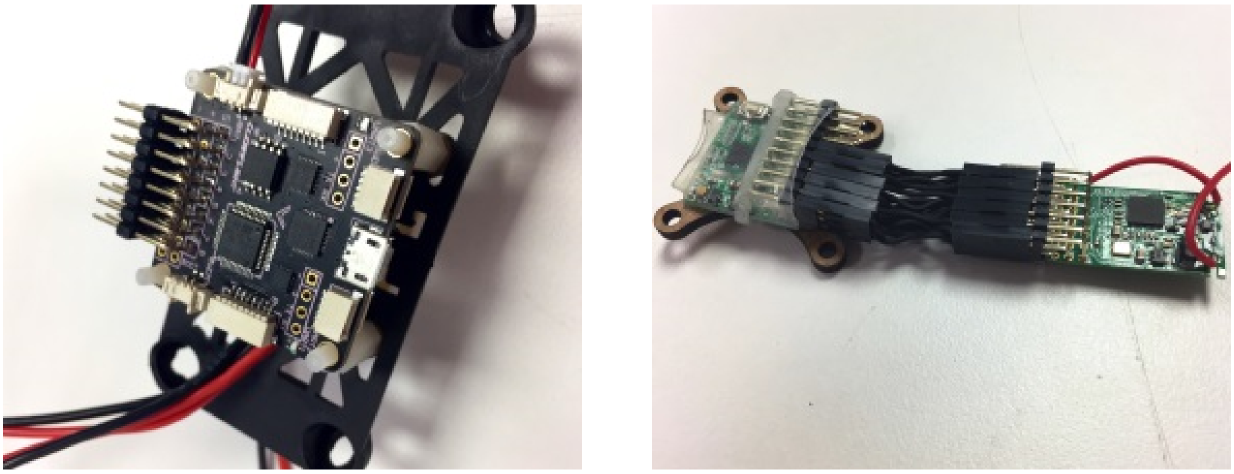


Figure 2.12: Hardware Schematic Diagram

An abstracted hardware diagram for the (electronic) system layout is shown in Fig:2.3. It's an illustration for the connection of different electronic peripherals used to aid the on-board control system. The structure of the autopilot system and control loops are addressed later in Chapter:6. This description aims to provide a brief overview of the specific modules used, their purpose and how they're interfaced. No code structure or control loops are considered yet...



(a) SPRacing F3 Deluxe Flight Controller **(b)** SBUS Converter & 6CH Receiver Modules

Figure 2.13

The entire system is centered around an ARM STM32F303 [85] based μC. The micro-processor board is a commercial flight control board, specifically an SPRacing F3 Deluxe [18][10], which has had its bootloader removed and custom firmware, unique to this project, developed for it. That software is later described in Chapter:6; the I/O for all the peripherals are detailed here however. The flight-controller has the following onboard peripherls: an I2C MPU-6050 [39] 6-axis gyroscope & accelerometer with a connected HMC5883 [24] magnetometer compass, an SPI MS5611 [83] barometer and similarly 64 Mb of SPI flash memory. The electrical schematic of those peripherals and the core STM32F303 is detailed in Appendix:B.2 but their connection(s) are shown in Fig:2.3.
The combination of above sensors fused for state estimation and their associated algorithms are dealt with in Section:5.3 in Chapter:5.

Figure 2.14: S.BUS Data Stream

Two wireless communication peripherals are used. First the system relays full state information, for a complete 6-DOF autopilot system, from a ground control station using 2.4 GHz XBEE S1 module(s) [40], USART connected. Secondly, an augmented pilot control input system, fail safe and secondary to the autopilot loop, is transmitted through 6 Channel 2.4 GHz R/F comms. The 6 CH received signals, otherwise permeated as six individual 20 KHz PWM signals via an OrangeRx R615x [66] receiver, are encoded into a single proprietary S.BUS data stream.

[10] CleanFlight opensource software is regularly used for the F3 but its hardware specifications are not openly avaiable. The reverse engineered electrical schematic for the board is included in Appendix:B.2

The need for an S.BUS encoder [36] comes about as a consequence of the introduction of the 8 additional servos. As a result, there are no longer 6 free additional timer I/O channels which can be allocated for input capture. Encoding the received data to a serial communication protocol means the 6CH data can be processed on a single serial RX line. The S.Bus encoder implements a USART derivative communications standard, Fig:2.14 shows the sampled data stream used to ascertain the standard's following parameters:

- 25 Bytes per packet
- 8-Bit byte length
- 1 Start byte 0x240
- 1 Byte of flags
- 1 Stop byte 0x0
- Bytes are:
  - MSB First
  - 1 start & 2 stop bits
  - Even parity bit
  - Inverted
  - 100000 baud (bps)
- 22 bytes of CH data
- Each channel's data is 11 bits long
- 16CH encoded
- Channel data is little endian prioritized
- 14 ms idle time between packets
- Packets are arranged:

$$\overbrace{[0x240]}^{Start\ byte}\ \overbrace{[8B_1][3B_2}^{CH1}|\overbrace{5B_2][6B_3}^{CH2}|\overbrace{2B_3][8B_4][1B_5}^{CH3}|\ldots$$

$$\overbrace{7B_5][4B_6]}^{CH4} \longrightarrow \overbrace{3B_22][8B_23]}^{CH16}\overbrace{[8B_24]}^{Flags}\ \overbrace{[0x00]}^{Stop\ byte}$$

The received information from the transmitted 6 channels is filtered through a moving average IIR† filter. The filters difference equation is as follows:

$$y_n = (1 - \frac{1}{N})y_{n-1} + \frac{1}{N}x_n \tag{2.22}$$

Moving over an average of $N = 5$ samples. The signal's sample rate is sufficiently fast enough such that the digital filter's frequency response isn't of consequence. Similarly all the measured RPM signals are filtered as well. Any received signals referred to are all post filtration. Filtering for state estimates is separately performed on the Ground Control Station computer.

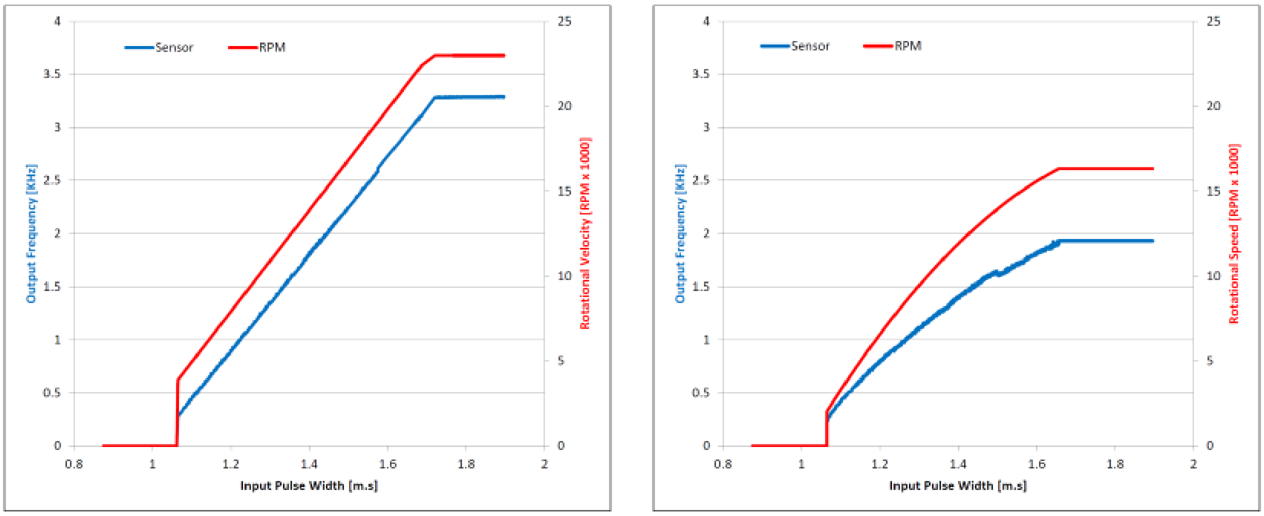
Each of the eight digital servo actuators are driven individually from 330 Hz PWM timer output compare channels (TIM2:CH1→CH4 & TIM3:CH1→CH4). Output pulses typically range from 1ms - 2ms to linearly control the rotary position. The exact range and transfer function is empirically determined next in Subsection:2.3.1. The four 20A brushless DC speed controllers (ESCs) are each driven from a 20 Hz PWM output (TIM4:CH1→CH4), similarly with 1ms - 2ms pulse widths. There are a total of 12 PWM output compare signals drawn from the μC. Servos are powered by a regulated 6V DC 10A power supply [35] whilst the ESCs switch unregulated 15.1 V DC from an externally tethered power supply. The DC supply could potentially be drawn from an on-board battery bank but that would add significant weight to an already heavy platform.

There's no integrated feedback for instantaneous RPM values from the ESCs. Using discrete OrangeRX BLDC RPM sensors [34], that measure switching phases across two of the three motor phases, the exact RPM can be ascertained. The switching signal of a 3-Phase [11] is [**?**]:

$$F_{rps} = \frac{2 \times F_{poles}}{\text{No. of rotor poles}} \quad [Hz] \tag{2.23}$$

The signal produced by the RPM sensors varies a 50% duty cycle square wave's period, directly proportional to that pole switching frequency. The RPM sensor's output signal is then calibrated to a gain of 7 for the 14 pole BLDC Cobra motors used. That gain is verified with the linear relationship(s) is shown in Figs:2.15. Knowing exact RPM rates means the subsequent thrust and aerodynamic torques for the control plant inputs can be calculated.

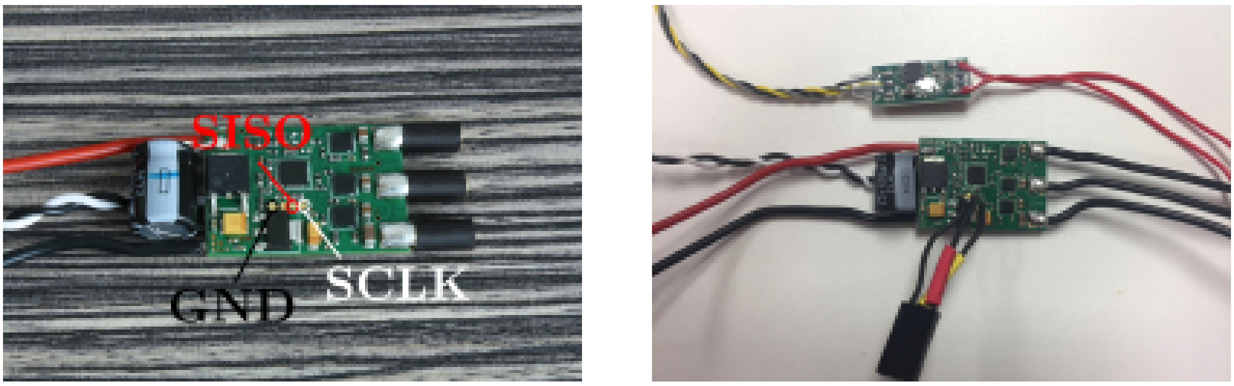
[11] Although called BLDC motors, the motors are actually 3-Phase IM motors which, when combined with an ESC, behave in closed loop like BLDC motors.



(a) RPM sensor plot - no load **(b)** RPM sensor plot - 6" prop

Figure 2.15

The speed controllers, although LDPower 20A devices, are all re-flashed with BLHeli[12] [8] firmware. The custom software on the ESC's μcontroller can provide greater refinement over parameter configuration like the deflection range of inputs, however, default values were used. The plot in Fig:2.15a shows the linear RPS (in Hz) speed line for an unloaded motor, similarly in Fig:2.15b shows the speed curve when loaded for a 6-inch prop. It's interesting to note that the loaded speed curve is slightly parabolic, this is from the aerodynamic drag term which is quadratic with respect to rotational velocity, Section:3.3.1. Moreover, when the motor is torque loaded by the propeller, the ESC current limits rotational speeds at just over 13 000 RPM. The sensor feedback is used for minor loop RPM control.



(a) XRotor 20A ESC connection guide [33] **(b)** LDPower 20A ESC

Figure 2.16

Timers channels are used to measure the varying frequency of the RPM sensors. General purpose Timers 15 (TIM15:CH1→CH2), 16 (TIM16:CH1) and 17 (TIM17:CH1) are configured to capture the input PWM signal generated by the speed sensors. Included on the I2C communciation line is an I2C O-LED display for debugging and status update purposes.

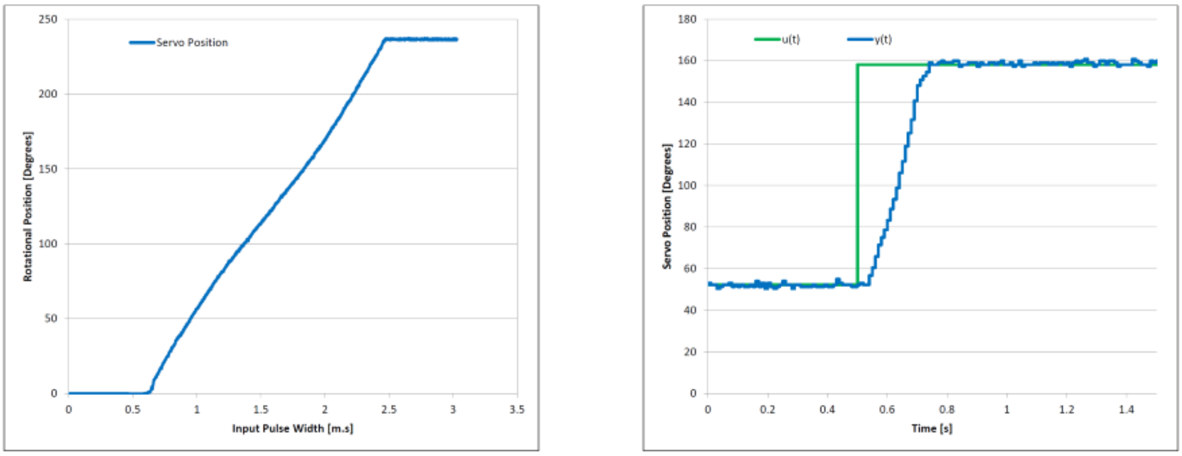
Any STM32 μcontroller is programmed through a dedicated debugging device. The ST-Link V2 [84] is the current proprietary device which, itself, is a specially programmed STM32F10 chip. The chip connects to the dedicated **S**erial **W**ire **D**ebugging ports of the target STM (*SWD-CLK, SWD-IO* & *SWD-NRST*) and is interfaced via regular USBD+ and USBD- data lines.

[12] LDPower 20A ESCs(Fig:2.16b) match Hobbywing Xrotor 20A speed controllers (Fig:2.16a), they both use SiLabs F396 MCUs. Physical rotational values in the plots Fig:2.15 were measured with optical encoders.

2.3.1 Actuator Transfer Functions

Servo Transfer Functions

The full scale deflection of each digital servo is in fact greater than its quoted 180° range. Each servo has a rotational range of around 230° (Fig:2.17a). The exact characteristics for every servo differ slightly and so individual transfer functions for each of the 8 servos are used. In the prototype control loop the servos are left in open loop; the major loop controller coefficients are expected to account for minor loop actuator dynamics. With that being said, for such an expectation the simulation would need to accurately represent the servo's response. Seeing that the 180° limitation was a design decision imposed, one of the first points of comparison is the effect such a constraint would have on the feasible operational regions. The control algorithms developed in Chapter:4 are first tested with an ideal, continuous rotation servo actuator with similar rate limits and transfer characteristics. For the servo



(a) DS339-MG Full Range

(b) DS339-MG Step Response

Figure 2.17

[13] whose data range and response are shown in Fig:2.17, the relationship the input pulse-width x $[m.s]$ and the rotational output position y *degrees* is given by:

$$y(x) = \begin{cases} 0° & x < 0.65\ m.s \\ 129.12x - 82.64 & 0.64\ m.s \leq x \leq 2.46\ m.s \\ 230° & x > 2.46\ m.s \end{cases} \tag{2.24}$$

Although, in practice, the equation Eq:2.24 is changed such that 0° is taken at around a 50% input and the operational range is ±90°. Each servo is mechanically rate limited to 60°/0.15s or 400RPS with a dead time of 1.2 $m.s$ and a mechanical deadband of $\leq 4\mu s$. The net transfer block for the servo then has the following structure, including non-linearities but neglecting the deadband;

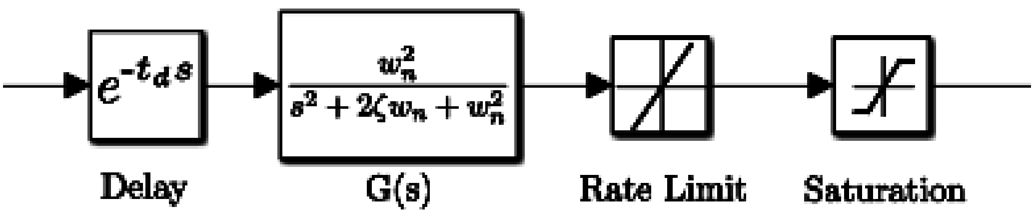


Figure 2.18: Servo block diagram

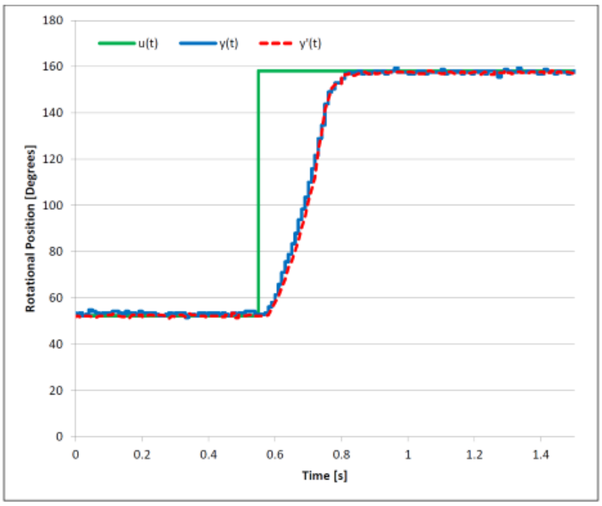
[13] Servo λ_1

Each servo has an approximate (*critically damped*) second order transfer function†:

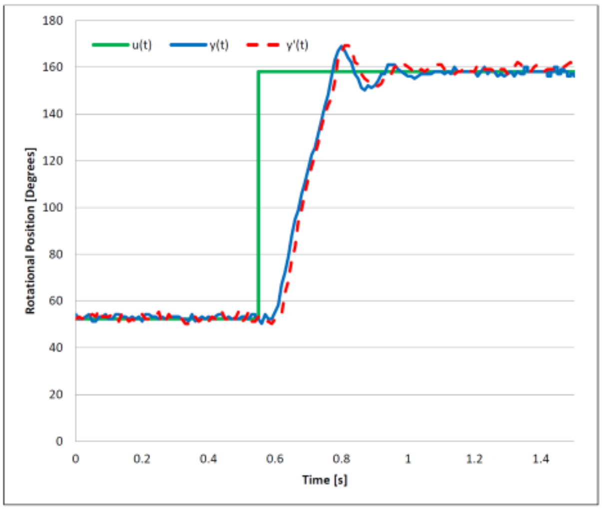
$$G(s) = e^{-t_d s}\frac{w_n^2}{s^2 + 2\zeta w_n s + w_n^2} = e^{-0.012s}\frac{(15.717)^2}{s^2 + 2(1)(15.717) + (15.717)^2} \tag{2.25a}$$

$$Y(s) = \begin{cases} 0^\circ & |U(s)| < 0.65 \\ G(s) & 0.65 \leq |U(s)| \leq 2.46 \\ 230^\circ & |U(s)| > 2.46 \end{cases} \tag{2.25b}$$

The plots in Fig:2.17b are of an unloaded servo's response. When loaded by an inner ring assembly (Fig:2.19a) the plant response $y(t)$ is consistent with Eq:2.25. Despite rotating a load mass and hence requiring a larger torque, the servo remains unchanged, even when the BLDC motor (with a 6×4.5" prop) is spun an average rate of 5000 RPM, $y'(t)$, further loading the servo.



(a) Inner Ring Servo Response



(b) Middle Ring Servo Response

However, in Fig:2.19b the mechanical response remains the some but harmonics are introduced from the larger mass being driven. These are product of the structure flex within the middle ring assembly[14], *not from the servo plant*. The oscillations still exist when under load, $y'(t)$, but can be reduced by either introducing a more rigid sub-frame, limiting the maximum angular rate or applying a damping minor loop controller. The latter would be in *virtual* closed loop with an approximated error rate as the prototype doesn't support position feedback for each motor module.

Motor RPM Transfer Functions

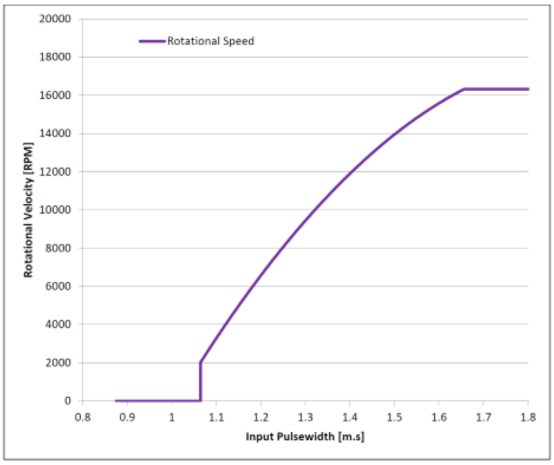
Each Cobra 2208 BLDC motor, when loaded with a 6×4.5 propeller has a quadratic speed curve, Fig:2.20a. This is as a result of the propeller's aerodynamic drag, *appromixately* proportional to the square of the propellers angular velocity (Section:3.3.1). The relationship[15] between input pulse-width to the ESCs and output RPM sensor signal is:

$$y(x) = \begin{cases} 0^\circ & x < 1.065\ m.s \\ -20593x^2 + 80187x - 60004 & 1.065\ m.s \leq x \leq 1.655\ m.s \\ 230^\circ & x > 1.655\ m.s \end{cases} \tag{2.26}$$

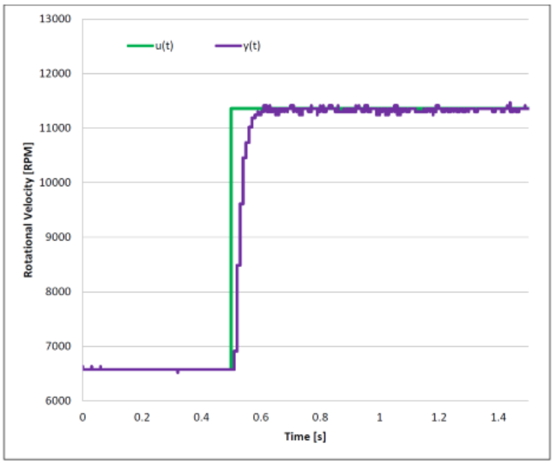
[14] rotational position was measured with respect to the bearing supported output shaft that is coaxial to the servos, not the servo output shaft

[15] The input range can be altered in ESC software

The upper limit in Eq:2.26 is goverened by the maximum ESC current limit, which is 20A. The step response has a negligible dead time



(a) BLDC RPM range



(b) Cobra BLDC step response

Chapter 3

Kinematics & Dynamics

The body's dynamics are first solved as rigid, with appropriate equations derived for generic 6-DOF motion. There after, non-linear aerodynamic and inertial effects, unique to multi-body relative rotations, are investigated and incorporated into the plant's model. Finally a consolidated, quaternion based plant model is presented which is used for the later control plant development next in Chapter:4.

3.1 Rigid Body Dynamics

3.1.1 Lagrange Derivation

Fundamentally any body, rigid or otherwise, can undergo two kinds of movements, namely rotational and translation motions. Often a Lagrangian [74,88] approach for combined angular and translational movements is used to derive the differential equations of motion for each degree of freedom. The Lagrangian principle ensures that translational and rotational kinematic energies and potential energy are conserved all throughout the system's trajectory progression. When combined with Euler-Rotational equations, the Euler-Lagrangian [90] formulation fully defines the aerospace 6-DOF equation set.

Lagrangian formalism is regarded as especially useful in non-cartesian (*spherical etc...*) co-ordinate frames and multi-body systems. With that being said, a cartesian co-ordinate system was already defined in Section:2.2.2, rigid body dynamics in a cartesian co-ordinate frame do lend themselves to Newtonian mechanics. The Newtonain-Euler or Euler-Lagrange formulations both stipulate the same resultant differential equations of motion. The Lagrangian operator, $\mathcal{L}$, is a term made up of the difference between kinetic and potential energies, T and U respectively. Considering some generalized path co-ordinates $\mathbf{r}(t)$, for both linear $\mathcal{E}$ and angular η relative positions;

$$\mathbf{r}(t) = \begin{bmatrix}\mathcal{E} & \eta\end{bmatrix}^T \tag{3.1}$$

The co-ordinates in Eq:3.1 are generalized here, despite being symbols commonly used to represent linear and angular positions. The generalized co-ordinates are later be refined as Cartesian body co-ordinates with respect to the inertial frame. The Lagrangian, by definition, is then:

$$\mathcal{L}(\mathbf{r}, \dot{\mathbf{r}}, t) = T(\mathbf{r}, \dot{\mathbf{r}}) - U(\mathbf{r}, \dot{\mathbf{r}}) \tag{3.2a}$$

Introducing generic kinetic (angular & linear) and potential energies, the latter being only gravitational potential energy in this case;

$$\mathcal{L} = \frac{1}{2}\dot{\mathcal{E}}^T(m)\dot{\mathcal{E}} + \frac{1}{2}\dot{\eta}^T(\mathbb{I})\dot{\eta} - mgz \tag{3.2b}$$

Noting that $\mathbb{I}$ is the inertial tensor aligned w.r.t to whichever generalized coordinates are used. The Euler-Lagrange formulation equates partial derivatives of the Lagrangian to any generalized forces, $\mathbf{V}$, acting on the system. In this case the generalized forces or a net force, F $[N]$, and a net torque, τ $[N.m]$.

$$\frac{d}{dt}\left(\frac{\delta L}{\delta \dot{\mathbf{r}}}\right) - \frac{\delta L}{\delta \mathbf{r}} = \mathbf{V} = \begin{bmatrix} F \\ \tau \end{bmatrix} \tag{3.3}$$

Taking the partial derivatives of Eq:3.2b with respect to its path co-ordinates $\mathbf{r}$:

$$\frac{\delta L}{\delta \mathbf{r}} = \begin{bmatrix} mG \\ 0 \end{bmatrix} \tag{3.4a}$$

$$\frac{d}{dt}\left(\frac{\delta L}{\delta \dot{\mathbf{r}}}\right) = \begin{bmatrix} m\frac{d}{dt}\dot{\mathcal{E}} & \mathbb{I}\frac{d}{dt}\dot{\eta} \end{bmatrix}^T \tag{3.4b}$$

In any generalized coordinate system a rotating vector's time derivative, according to the Reynolds Transportation Theorem [**?**, 70], is given by:

$$\frac{d\vec{f}}{dt_a} = \frac{d\vec{f}}{dt_b} + \vec{\omega}_{a/b} \times \vec{f} \tag{3.5}$$

So applying that theorem (Eq:3.5) to the partial derivatives in Eq:3.4b and further defining the generalized co-ordinates as cartesian body coordinates with respect to an inertial origin. Noting that in Eq:3.4b the place holders used for linear and angular positions are in a common shared frame[1], and hence $[\dot{\mathcal{E}} \;\; \dot{\eta}]^T \equiv [\nu \;\; \omega]^T \in \mathcal{F}^b$. It then follows that Lagrangian will change:

$$\mathcal{L} = \frac{1}{2}\nu^T(m)\nu + \frac{1}{2}\omega^T(\mathbb{I})\omega - mG_b z \tag{3.6a}$$

$$\frac{d}{dt}\left(\frac{\delta L}{\delta \dot{\mathbf{r}}}\right) = \begin{bmatrix} m\frac{d}{dt}\nu & \mathbb{I}\frac{d}{dt}\omega \end{bmatrix}^T \tag{3.6b}$$

$$\rightarrow m\frac{d}{dt}\nu_b = m\dot{\nu} + \vec{\omega}_{I/b} \times \nu \tag{3.6c}$$

$$\rightarrow \mathbb{I}_b\frac{d}{dt}\omega_b = \mathbb{I}_b\dot{\omega} + \vec{\omega}_{I/b} \times \mathbb{I}_b\omega \tag{3.6d}$$

Which, when reintroduced to the Euler-Langrage formulation Eq:3.3, results in the familiar Newton-Euler equations for linear and angular movements, both in the body frame;

$$F = m\dot{\nu} + \omega_b \times m\nu - m\mathbb{R}_I^b(-\eta)G \tag{3.7a}$$

$$\tau = \mathbb{I}_b\dot{\omega}_b + \omega_b \times \mathbb{I}_b\omega \tag{3.7b}$$

It's important to recall that $\omega_b \neq \dot{\eta}$ where $\eta = [\phi \; \theta \; \psi]^T$, seeing that Euler Angles are defined in sequentially rotated reference frames. So the four differential equations often used to completely describe the entire state derivatives are:

$$\dot{\mathcal{E}} = \mathbb{R}_b^I(-\eta)\nu \qquad \in \mathcal{F}^I \tag{3.8a}$$

$$F = m\dot{\nu} + \omega_b \times m\nu - m\mathbb{R}_I^b(-\eta)G \qquad \in \mathcal{F}^b \tag{3.8b}$$

$$\dot{\eta} = \Psi(\eta)\omega_b \qquad \in \mathcal{F}^{v2}, \mathcal{F}^{v1}, \mathcal{F}^I \tag{3.8c}$$

$$\tau = \mathbb{I}_b\omega_b + \omega_b \times \mathbb{I}_b\omega \qquad \in \mathcal{F}^b \tag{3.8d}$$

[1] In this case $\eta \neq [\phi \; \theta \; \psi]^T$ seeing that it's defined in a common frame

The set of state differentials in Eq:3.8 can be reduced to a set of two equations, defined only in their respective reference frames of the state variables which they describe. The non-linear form of those equations substitutes[2] $\dot{\eta} = \Phi(\eta)\omega_b$ in the Lagrangian derivative, Eq:3.4b.

$$\frac{d}{dt}\left(\frac{\delta L}{\delta \dot{\mathbf{r}}}\right) = \begin{bmatrix} m\frac{d}{dt}\nu & \mathbb{I}\frac{d}{dt}\dot{\eta} \end{bmatrix} = \begin{bmatrix} m\frac{d}{dt}\nu & \mathbb{I}\frac{d}{dt}\Phi(\eta)\omega_b \end{bmatrix} \tag{3.9}$$

This only changes the angular component, and so applying the differential chain rule:

$$\mathbb{I}\frac{d}{dt}\Phi(\eta)\dot{\omega}_b = \mathbb{I}\big(\Phi(\dot{\eta})\omega_b + \Phi(\eta)\dot{\omega}_b\big) \tag{3.10}$$

Drawing from [64] and recognizing that $\mathbb{I}$ must be transformed to common axes, $\mathbb{J} = \Psi(\eta)^T\mathbb{I}\Psi(\eta)$. The controllable differential equations in Eq:3.7, then in the inertial frame for force and intermediate Euler frames for each angle becomes[3]:

$$M(\eta)\ddot{\eta} + C(\eta,\dot{\eta})\dot{\eta} = \Psi(\eta)\tau \tag{3.11a}$$

$$M(\eta) = \Psi(\eta)^T\mathbb{I}\Psi(\eta) \tag{3.11b}$$

$$C(\eta,\dot{\eta}) = -\Psi(\eta)\mathbb{I}\Psi(\dot{\eta}) + \Psi(\eta)^T sk(\Psi(\eta)\dot{\eta})\mathbb{I}\Psi(\eta) \tag{3.11c}$$

Equation 3.11 fully describes the state derivative $\ddot{\eta}$ in its own frame(s). The two differential equations which describe the entire bodies motion are then:

$$F = m\dot{\mathcal{E}} + \mathbb{R}_b^I(-\eta)\omega_b \times m\dot{\mathcal{E}} - mG \qquad \in \mathcal{F}^I \tag{3.12a}$$

$$\Psi(\eta)\tau = M(\eta)\ddot{\eta} + C(\eta,\dot{\eta}) \qquad \in \mathcal{F}^{v2,v1,I} \tag{3.12b}$$

The generalized forces effecting the system, $F(u)$ and $\tau(u)$, are the system's controllable inputs and are going to be affected directly the systems actuators and their associated effectiveness function. In the general case, which is expanded upon in Section:3.3, the control inputs are typically as follows: The net force acting on the system is just the sum of all thrust vectors produced by rotating propellers, $T(\Omega_i)$.

$$\mu F = \sum \vec{T_i} \tag{3.13a}$$

Secondly the net torque is the sum of all torque arms produced from those propeller thrust vectors.

$$\mu\tau = \sum \vec{l_i} \times \vec{T_i} \tag{3.13b}$$

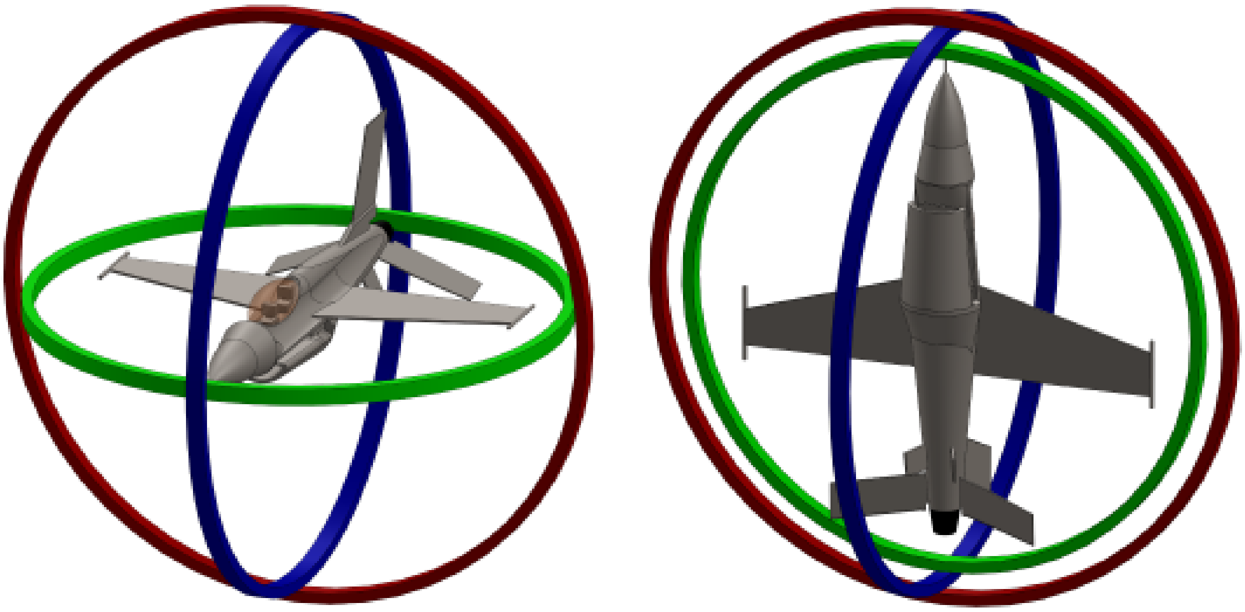
Where $\vec{T_i}$ is the i^{th} motor's thrust vector, not necessarily in 3 dimensions and typically bound to the $\hat{Z}_b$ axis, $\vec{l_i}$ is that motor's torque arm. The above equations are still applicable to any 6 DOF body, common simplifications applied to the system(s) for quadrotor control are explored in Appendix:A. Aspects unique to the quadrotor and tilting quadrotor platforms are now introduced...

3.1.2 Rotation Matrix Singularity

The Euler Angle singularity is often noted but infrequently proven mathematically, far less common is the demonstration of exactly how that singularity *mathematically* manifests itself. By definition, a singularity occurs when a loss of differentiability exists. In the case of an affixed 3-axis gimbal, when an intermediate axis, for example the rolling angle θ, is at $\pi/2$ then the remaining two axes become co-linear. That being a pitch ϕ or yaw ψ rotations will subsequently have the same effect. Such a situation results in the loss of a degree of freedom.

[2]Originally introduced in Eq:2.5e

[3]The relationship $\dot{\Phi} = \Phi\dot{\Psi}\Phi$ was used to simplify Eq:3.11, the singularity in Φ still remains...



(a) 3-Axis gimbal **(b)** Locked gimbal with loss of DOF

Figure 3.1: Gimbal lock

What is obvious in a physical system is not necessarily as clear mathematically. A clear loss of differentiability is manifested in the Euler Matrix $\Psi(\eta)$, Eq:2.5e from Section:2.2.1. The relation between angular velocity, in the inertial frame or inversely in the body frame, and the angular rates of the Euler Angles.

$$\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} = \begin{bmatrix}1 & sin(\phi)tan(\theta) & cos(\phi)tan(\theta)\\ 0 & cos(\phi) & -sin(\phi)\\ 0 & sin(\phi)sec(\theta) & cos(\phi)sec(\theta)\end{bmatrix}\begin{bmatrix}p\\ q\\ r\end{bmatrix} = \Phi(\eta)\omega_b \tag{3.14}$$

$$\text{As } \lim_{\theta\to\pi/2} sec(\theta) \to \infty \tag{3.15}$$

Or that $\Phi(\eta)$ is undefined at $\theta = \pi/2$. It's clear to see that in Eq:3.14 there exists an undefined singularity as $\theta \to \pi/2$. The physical consequence of this is the loss of a degree of freedom. More specifically, if one looks at how the rotation (or transformation) matrices are formulated:

$$\mathbb{R}_I^b = \mathbb{R}_z\mathbb{R}_y\mathbb{R}_x = \begin{bmatrix}c_\psi & -s_\psi & 0\\ s_\psi & c_\psi & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}c_\theta & 0 & s_\theta\\ 0 & 1 & 0\\ -s_\theta & 0 & c_\theta\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ 0 & c_\phi & -s_\phi\\ 0 & s_\phi & c_\phi\end{bmatrix} \tag{3.16a}$$

$$\mathbb{R}_I^b = \begin{bmatrix}c_\psi c_\theta & c_\psi s_\theta s_\phi - s_\psi c_\phi & c_\psi s_\theta c_\phi + s_\psi s_\phi\\ s_\psi c_\theta & s_\psi s_\theta s_\phi + c_\psi c_\phi & s_\psi s_\theta c_\phi - c_\psi s_\phi\\ -s_\theta & c_\theta s_\phi & c_\phi c_\theta\end{bmatrix} \tag{3.16b}$$

In the case where $\theta = \pi/2$;

$$= \begin{bmatrix}0 & c_\psi s_\phi - s_\psi c_\phi & c_\psi c_\phi + s_\psi s_\phi\\ 0 & s_\psi s_\phi + c_\psi c_\phi & s_\psi c_\phi - c_\psi s_\phi\\ -1 & 0 & 0\end{bmatrix} = \begin{bmatrix}0 & s(\phi-\psi) & c(\phi-\psi)\\ 0 & c(\phi-\psi) & s(\phi-\psi)\\ -1 & 0 & 0\end{bmatrix} = \mathbb{R}_{x'}(\phi-\psi) \tag{3.16c}$$

Where the outcome in Eq:3.16c represents an x-axis rotation in a new intermediate frame, post a 90° rotation in the y-axis. Through trigonometric double angles a degree of freedom is lost at $\theta = \pi/2$, when ϕ & ψ effect the same angle.

3.1.3 Quaternion Dynamics

An algorithm proposed in *How To Avoid a Singularity When Using Euler Angles?* [82] suggested a solution the problem of the Euler Angle singularity. The proposed heuristic was to switch between sequencing conventions (ZYX,ZYZ etc... there are 12 in total) such that the singularity is always avoided. However the implementation of such an algorithm is cumbersome and inefficient. Far more elegant is the use of *quaternion*, $\mathbb{R}^4$, attitude representations ([28, 31, 48] amongst others...).

A quaternion is analogous to a rotation matrix in that it represents an attitude difference between two reference frames. Without deliberating too much on their proof or details, a quaternion consists or a scalar component, q_0, and complex vector component, $\vec{q} \in \mathbb{C}^3$, such that;

$$Q \triangleq \begin{bmatrix} q_0 \\ \vec{q} \end{bmatrix} \quad \in \mathbb{R}^4 \tag{3.17}$$

Any and all quaternions, unless otherwise stated, in this dissertation are all unit quaternions[4], $Q \in \mathbb{Q}_u$. The need for quaternions with unity magnitude is such to ensure rotational operators don't affect the magnitude of the vector operand. A unit quaternion is defined as:

$$||Q|| = \sqrt{q_0{}^2 + \vec{q}^{\,2}} = 1 \tag{3.18}$$

Quaternion multiplication is distributive and associative, but not commutative. Specifically a quaternion multiplciation operation is equivalent to the Hamilton product. For two quaternions, Q & P:

$$Q \otimes P = \begin{bmatrix} q_0 & \vec{q} \end{bmatrix} \otimes \begin{bmatrix} p_0 & \vec{p} \end{bmatrix} \tag{3.19a}$$

$$= q_0 p_0 - \vec{q} \cdot \vec{p} + p_0 \vec{q} + q_0 \vec{p} + \vec{q} \times \vec{p} \tag{3.19b}$$

Seeing that the vector component of a quaternion is complex valued, it is natural that there exists a quaternion conjugate property. Namely:

$$Q^* = \begin{bmatrix} q_0 & -\vec{q} \end{bmatrix} \tag{3.20}$$

It then follows that[5]:

$$Q \otimes Q^* = \mathbb{I}_{4\times4} \tag{3.21}$$

To apply quaternion rotations to a vector ($\vec{v} \in \mathbb{R}^3$) involves multiplication by two unit quaternions.

$$\begin{bmatrix} 0 & \vec{v}\,' \end{bmatrix} = Q \otimes \begin{bmatrix} 0 & \vec{v} \end{bmatrix} \otimes Q^* \tag{3.22}$$

Mostly, the zero scalar components are omitted in a rotation (*or transformation*) operation, such that it is recognized the vector operands are substituted with quaternions.

$$\vec{v}\,' = Q \otimes (\vec{v}) \otimes Q^* \tag{3.23}$$

In the case of rigid body attitude representation, Q_b is the quaternion which represents the difference between $\mathcal{F}^I$ and $\mathcal{F}^b$. A quaternion operator is equivalent to a rotation matrix operation:

$$\mathbb{R}_I^b \underset{Q}{\iff} Q_b \otimes (.) \otimes Q_b^* \tag{3.24}$$

A quaternion time derivative, with Q_ω being a quaternion with a vector component equal to angular velocity $\omega \in \mathcal{F}^b$ and a zero scalar component, is given by:

$$\frac{d}{dt} Q_b = \frac{1}{2} Q_b \otimes Q_\omega = \begin{bmatrix} -\frac{1}{2}\vec{q}^{\,T} \vec{\omega}_b \\ \frac{1}{2}\left([\vec{q}\,]_\times + q_0 \mathbb{I}\right)\vec{\omega}_b \end{bmatrix} \tag{3.25}$$

[4]Unit quaternions are a subset of the quaternion space

[5]Disambigation:$\mathbb{I}$ in this context is a 4×4 identity matrix, not an inertial matrix

Using quaternions to represent attitudes negates the need for an Euler Matrix, $\Phi(\eta)$, to represent attitudes and their rates. A body quaternion is fully defined in the body frame. The first quaternion time derivative replaces Eq:3.8a& Eq:3.8c;

$$\dot{\mathcal{E}} = \mathbb{R}_b^I(-\eta)\vec{\nu} \underset{Q}{\Longleftrightarrow} Q_b \otimes \vec{\nu} \otimes Q_b^* \tag{3.26a}$$

$$\dot{\eta} = \Phi(\eta)\vec{\omega}_b \underset{Q}{\Longleftrightarrow} \dot{Q} = \frac{1}{2}Q_b \otimes Q_\omega \tag{3.26b}$$

Second order quaternion derivatives, quaternion accelerations aren't as useful as their velocity counterparts. The second order derivative is mentioned here however it's only relevant to quaternion backstepping in the control section. If possible, quaternion accelerations are avoided due to their complexity;

$$\ddot{Q}\big(\dot{Q}, Q, t\big) = \dot{Q} \otimes Q^* \otimes \dot{Q} + \frac{1}{2}Q \otimes \big[\mathbb{I}_b^- 1(\tau - 4(Q^* \otimes \dot{Q}) \times (\mathbb{I}_b(Q^* \otimes \dot{Q}))\big] \tag{3.27}$$

3.1.4 Quaternion Unwinding

Although quaternions are better than Euler angles and lack the associated singularity, they do contain one caveat. Seeing that a quaternion $Q = [q_0 \; \vec{q}]^T$ represents a rotation in $\mathbb{R}^3$ using $\mathbb{R}^4$ variables there is whats called dual coverage [59]. Each unit quaternion, stemming from Euler-Rodriguez theorem, is parameterised such that the quaternion operation represents a single rotation of θ about an axis $\hat{u}$ such that:

$$Q = \begin{bmatrix} q_0 & \vec{q} \end{bmatrix} = \pm \begin{bmatrix} cos(\theta/2) & sin(\theta/2)\hat{u} \end{bmatrix} \tag{3.28}$$

With a unit vector $\hat{u}$. Quaternions relate to an euler angle rotation matrix through the Rodriguez formula:

$$\mathbb{R}(Q) = \mathbb{I} + 2q_0[\vec{q}]_\times + 2[\vec{q}]_\times^2 \tag{3.29}$$

As a result its clear to see that for each unique attitude in 3-Dimensions there exist two quaternions which correlate to the same position. Namely:

$$Q = \begin{bmatrix} cos(\theta/2) & sin(\theta/2) \end{bmatrix}. \tag{3.30}$$

3.2 Non-linearities

3.2.1 Gyroscopic Torques

3.2.2 Coriolis Acceleration

3.2.3 Inertial Matrix

3.3 Aerodynamics

3.3.1 Thrust Forces & Propeller Torques

3.3.2 Drag

3.3.3 Conning & Flapping

3.3.4 Vortex Ring State

3.4 Consolidated Model

Chapter 4

Control Treatment

Control Plant & Discussion

Control Plant Inputs

Model Dependent & Independent Controllers

4.1 Attitude Control

4.1.1 The Attitude Control Problem

4.1.2 Quaternion Based Controllers

PD Controller

Auxilliary Plant Controller

PID Controller

4.1.3 Non-linear Controllers

Ideal Back-stepping Controller

Adaptive Back-stepping Controller

Lyupanov Derived Ideal Controller

4.2 Position Control

4.2.1 Backstepping Position Controller

4.3 Controller Allocation

4.3.1 Non-linear Plant Control Allocation

4.3.2 Pseudo Inverse Allocator

Chapter 5

Simulations & Results

5.1 Controller Tuning

5.1.1 Partical Swarm Based Optimization

5.1.2 Performance Metric

5.1.3 Global & Local Minima

5.1.4 Fmincon Differences

5.2 Simulation Block

5.3 State Estimation

5.4 Optimized Controller Comparisons

5.4.1 Allocator Performance

5.4.2 Attitude Control Results

5.4.3 Autopilot Outcome

Chapter 6

Prototype Flight Results

Chapter 7

Conclusion

- Lagrange dynmaics for multibody system could have produced a more concise model etc ...

Appendix A

Standard Quadrotor Dynamics

Appendix B

Design Bill of Materials

B.1 Parts List

Part Name	No. Used	Unit Weight[g]
Electronics		
SPRacing F3 Deluxe Flight Controller	1	8
OrangeRx 615X 2.4 GHz 6CH Receiver	1	9.8
Signal Converter SBUS-PPM-PWM	1	5.0
STLink-V2 Debugger	1	3
RotorStar Super Mini S-BEC 10A	1	30
128x96” OLED Display	1	7
XBee-Pro S1	2	4
HobbyWing XRotor 20A Opto ESC	4	15
OrangeRX RPM Sensor	4	2
HobbyKing Multi-Rotor Power Distribution Board	1	49
Motors		
Corona DS-339MG	8	32
Cobra 2208 2000KV Brushlesss DC	4	44.2
Frame Components		
APM Flight Controller Damping Platform	1	7
HobbyKing SK450 Replacement Arm (2 pcs)	2	N/A
SK450 Extended Landing Skid	1	23.25
Alloy Servo Arm (FUTABA)	8	4
10X18X6 Radial Ball Bearing	8	5
80g Damping Ball	32	≈ 0
Plastic Retainers for Damping Balls	32	≈ 0
3/5mm Aluminum Prop Adapter	4	≈ 1
6x4.5 Gemfam 3-Blade Propeller	4	6
M3 6mm Hex Nylon Spacer	8	≈ 0
M3 16mm Hex Nylon Spacer	32	≈ 0
M3 25mm Nylon Screw	128	≈ 0.08
M2.5x10mm Socket Head Cap Screw	36	≈ 0.2
M2.5x25mm Socket Head Cap Screw	20	≈ 0.6
M2.5 A-Lok Nut	16	≈ 0

Table B.1: Parts List

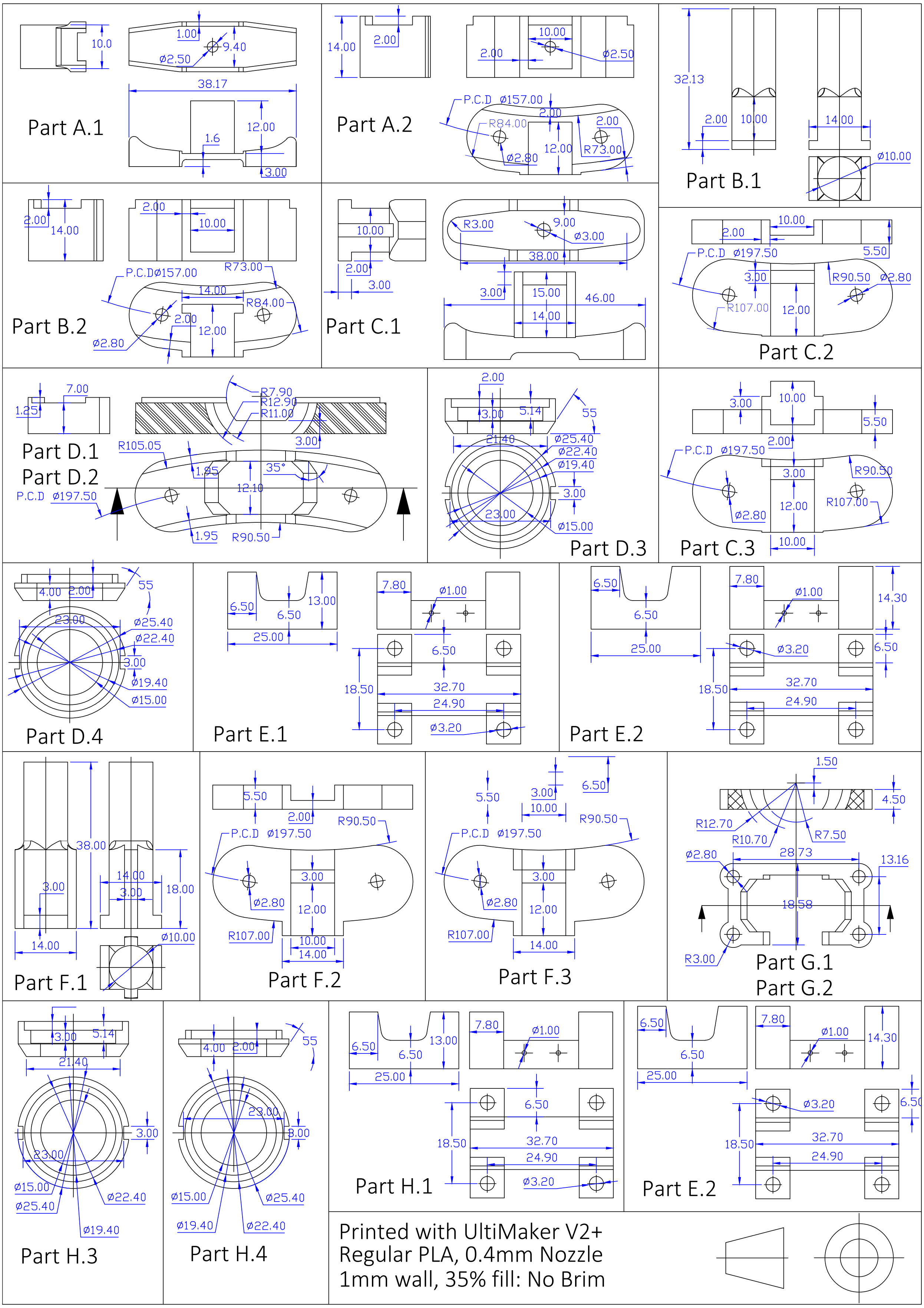


Table B.2: 3D Printed Parts

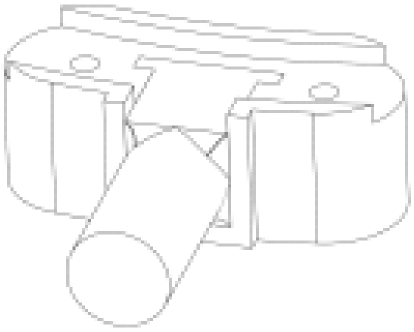
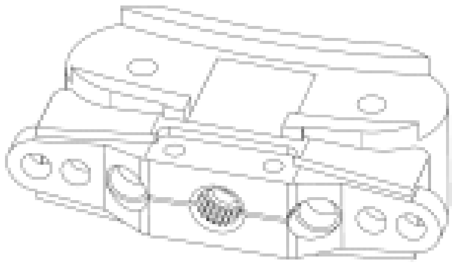
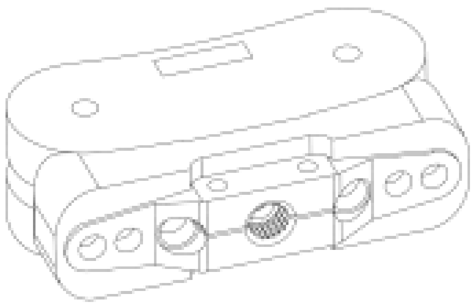
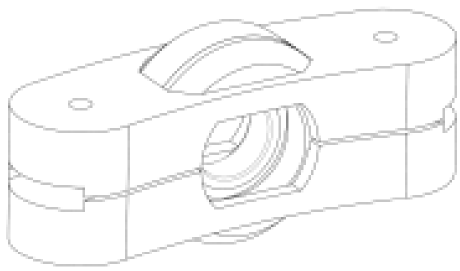
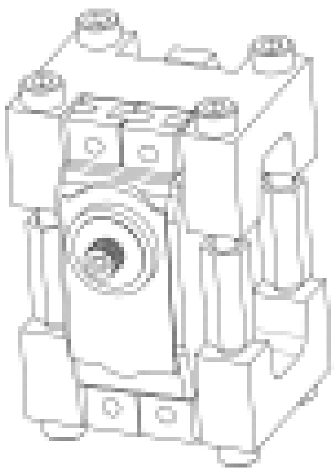
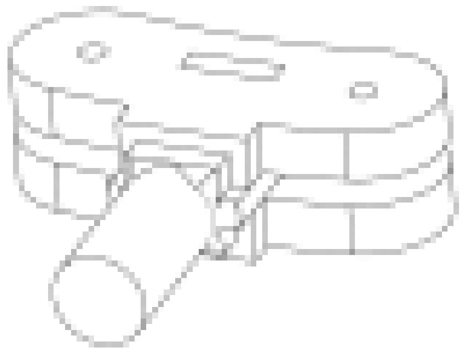
Bracket Assemblies 2	
 **Figure B.1:** Bearing Bracket Inner Ring Assembly Parts: A.1, A.2	 **Figure B.2:** Servo Bracket Inner Ring Assembly Parts: B.1, B.2, M3 Servo Horn
 **Figure B.3:** Servo Bracket Middle Ring Assembly Parts: C.1, C.2, C.3, M3 Servo Horn	 **Figure B.4:** Bearing Holder Middle Ring Assembly Parts: D.1, D.2, D.3, D.4, 18-10 Bearing
 **Figure B.5:** Servo Mount Middle Ring Assembly Parts: E.1, E.2, Corona Servo & Fasteners	 **Figure B.6:** Bearing Shaft Middle Ring Assembly Parts: F.1, F.2, F.3

Table B.3: Inner & Middle Ring Assemblies

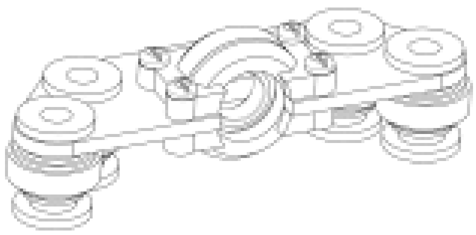
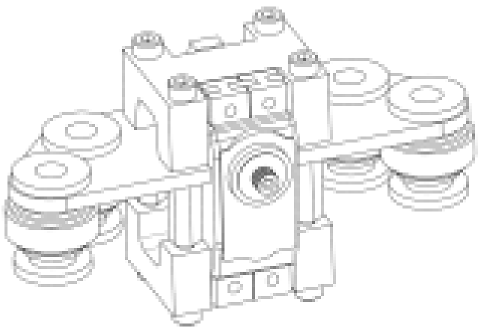
Bracket Assemblies 2	
 **Figure B.7:** Bearing Holder Damping Assembly Parts: G.1, G.2, G.3, G.4, 18-10 Bearing, 80g Damping Balls, Bearing Holder Damping Bracket	 **Figure B.8:** Servo Mount Damping Assembly Parts: H.1, H.2, Corona Servo & Fasteners, 80g Damping Balls, Servo Mount Damping Bracket

Table B.4: Damping Assemblies

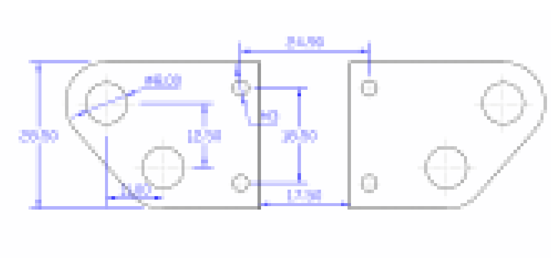
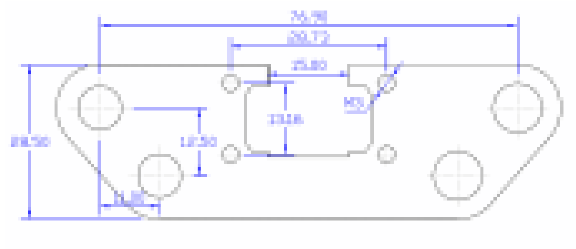
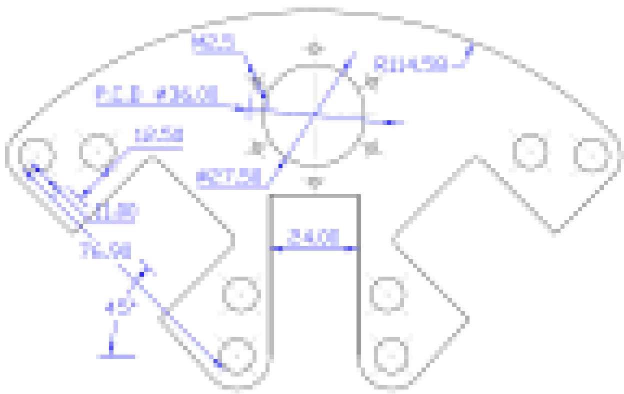
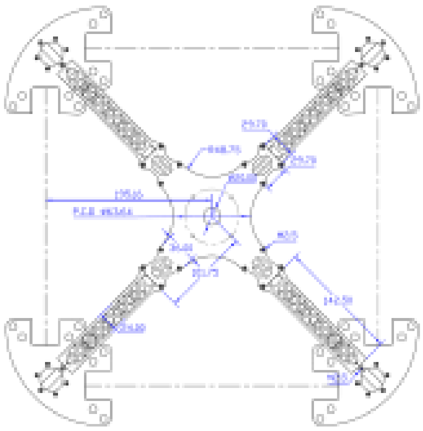
Laser Cut Brackets	
 **Figure B.9:** Servo Mount Damping Bracket	 **Figure B.10:** Bearing Holder Damping Bracket
 **Figure B.11:** Arm Mount Damping Bracket	 **Figure B.12:** Frame Brackets

Table B.5: Laser Cut Damping Brackets

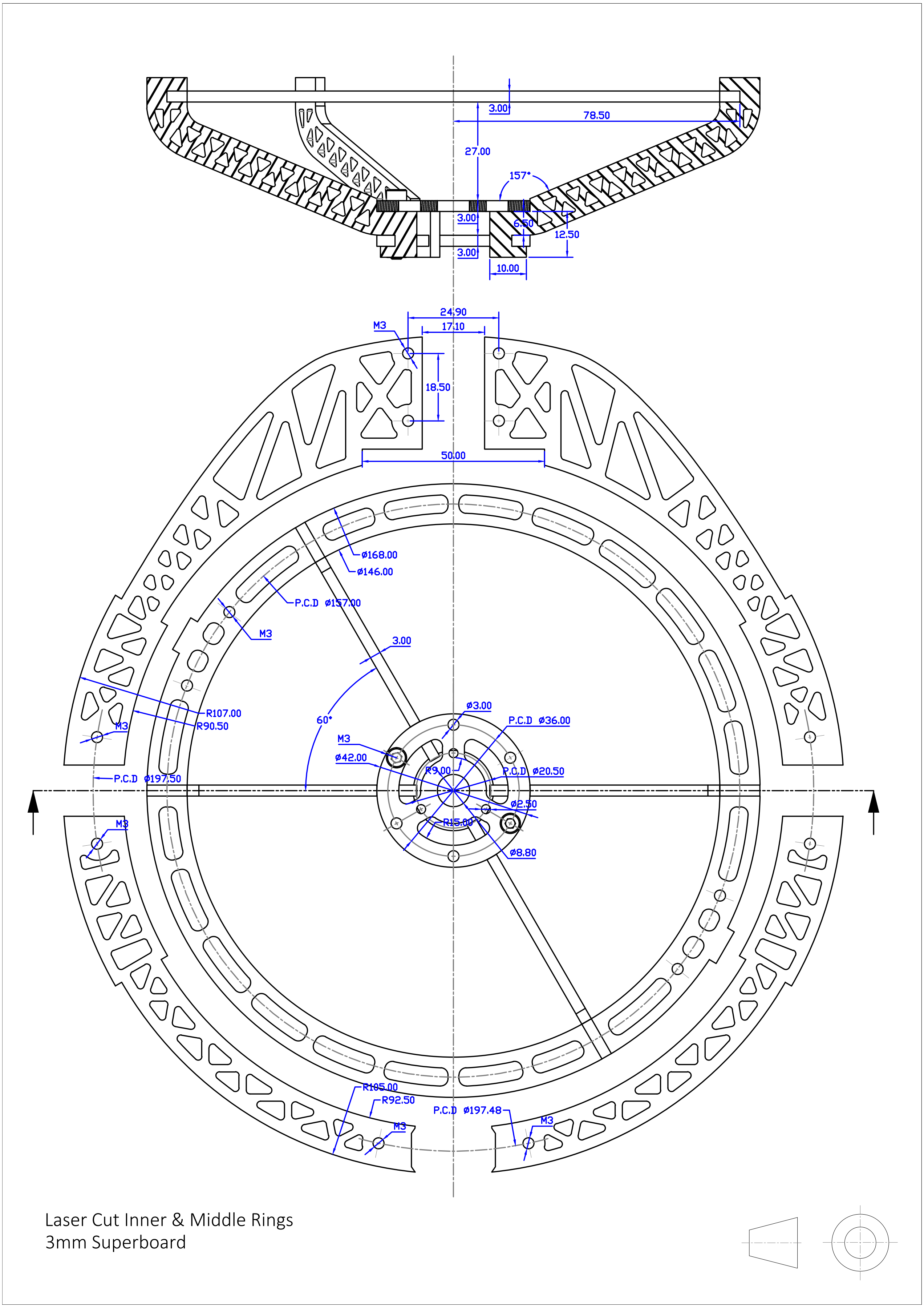


Table B.6: Laser Cut Parts

B.2 F3 Deluxe Schematic Diagram

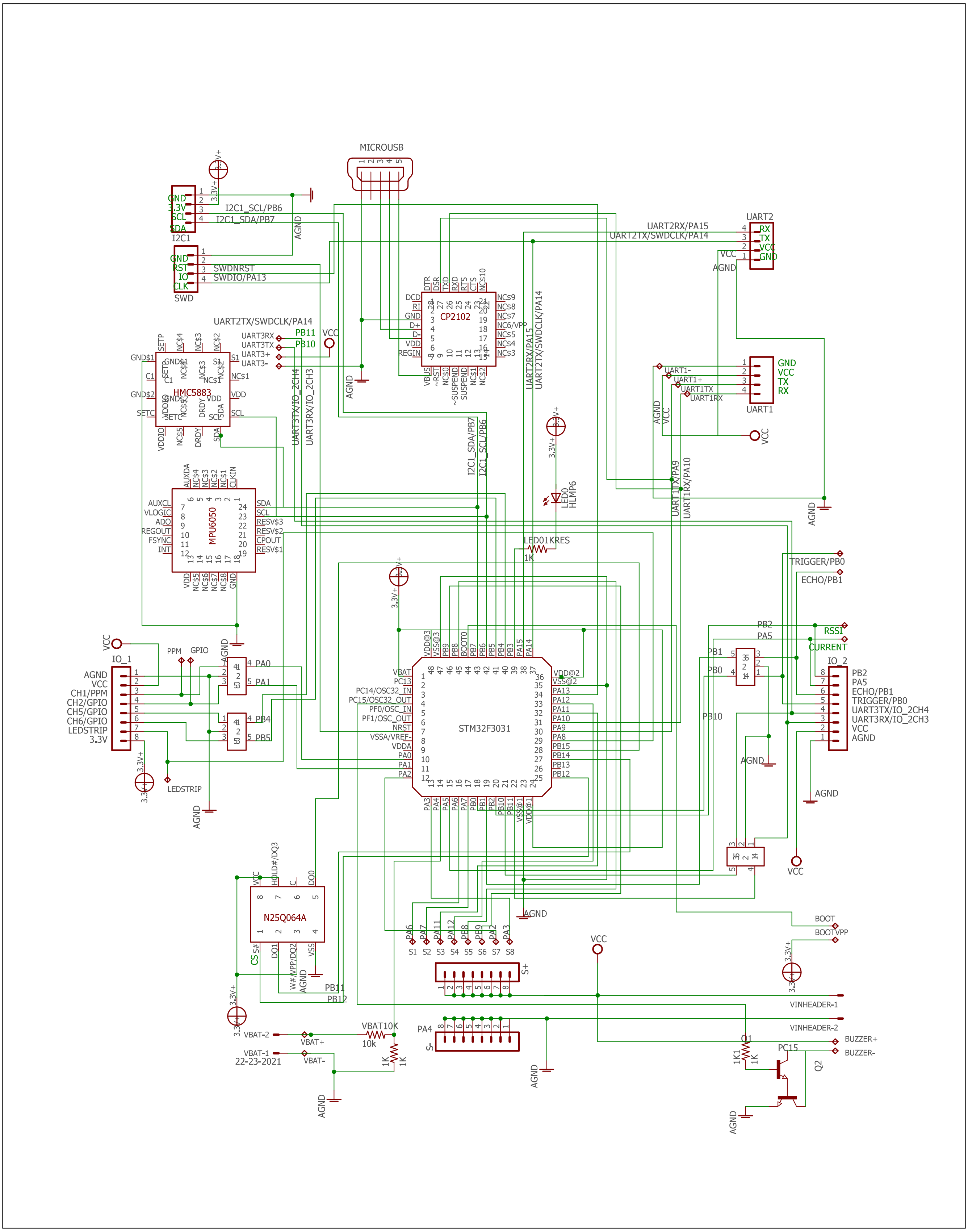


Figure B.13: F3 Deluxe Flight Controller Hardware Schematic

Appendix C

System ID Test Data

C.1 Servo Data

C.2 Cobra CM2208-200KV

Cobra CM-2208/20 Motor Propeller Data

Magnets	Motor Wind	Motor Kv	No-Load Current	Motor Resistance	I Max	P Max (3S)
14-Pole	20-Turn Delta	2000 RPM/Volt	Io = 0.77 Amps @ 10v	Rm = 0.076 Ohms	20 Amps	220 W

Stator	Outside Diameter	Body Length	Total Shaft Length	Shaft Diameter	Motor Weight
12-Slot	27.7 mm, 1.091 in.	24.0 mm, 0.945 in.	45.2 mm, 1.780 in.	3.17 mm, 0.125 in.	44.2 gm, 1.56 oz

Test Data From	Input	6.0 V	8.0 V	10.0V	12.0V	Measured Kv value	Measured Rm Value
Sample Motor	Io Value	0.59 A	0.67 A	0.77 A	0.87 A	1988 RPM/Volt @ 10v	0.076 Ohms

Prop Manf.	Prop Size	Li-Po Cells	Input Voltage	Motor Amps	Input Watts	Prop RPM	Pitch Speed in MPH	Thrust Grams	Thrust Ounces	Thrust Eff. Grams/W
APC	5.25x4.75-E	3	11.1	13.34	148.1	17,507	78.7	451	15.91	3.05
APC	5.5x4.5-E	3	11.1	13.67	151.7	17,388	74.1	456	16.08	3.01
APC	6x4-E	3	11.1	14.87	165.1	17,003	64.4	630	22.22	3.82
APC	7x4-SF	3	11.1	21.82	242.2	13,985	53.0	840	29.63	3.47
APC	7x5-E	3	11.1	24.02	266.6	13,272	62.8	797	28.11	2.99
FC	5x4.5	3	11.1	8.66	96.1	19,061	81.2	428	15.10	4.45
FC	5x4.5x3	3	11.1	12.38	137.4	17,825	76.0	534	18.84	3.89
FC	6x4.5	3	11.1	15.47	171.7	16,792	71.6	721	25.43	4.20
GemFan	5x3	3	11.1	6.67	74.0	19,801	56.3	374	13.19	5.05
HQ	5x4	3	11.1	7.13	79.1	18,182	68.9	373	13.16	4.71
HQ	5x4x3	3	11.1	9.25	102.7	17,401	65.9	449	15.84	4.37
HQ	5x4.5-BN	3	11.1	11.17	124.0	16,902	72.0	487	17.18	3.93
HQ	6x3	3	11.1	7.34	81.5	18,128	51.5	419	14.78	5.14
HQ	6x4.5	3	11.1	13.53	150.2	16,206	69.1	645	22.75	4.29
HQ	6x4.5x3	3	11.1	17.60	195.4	15,137	64.5	762	26.88	3.90
HQ	7x4	3	11.1	20.71	229.9	14,250	54.0	850	29.98	3.70
HQ	7x4.5	3	11.1	20.31	225.4	14,351	61.2	865	30.51	3.84

Prop Manf.	Prop Size	Li-Po Cells	Input Voltage	Motor Amps	Input Watts	Prop RPM	Pitch Speed in MPH	Thrust Grams	Thrust Ounces	Thrust Eff. Grams/W
APC	5.25x4.75-E	4	14.8	17.29	255.9	20,560	92.5	603	21.27	2.36
APC	5.5x4.5-E	4	14.8	17.87	264.5	20,436	87.1	635	22.40	2.40
APC	6x4-E	4	14.8	20.15	298.2	19,829	75.1	837	29.52	2.81
FC	5x4.5	4	14.8	10.89	161.2	22,511	95.9	588	20.74	3.65
FC	5x4.5x3	4	14.8	16.43	243.2	20,828	88.8	718	25.33	2.95
FC	6x4.5	4	14.8	20.09	297.3	19,809	84.4	998	35.20	3.36
HQ	4x4.5-BN	4	14.8	10.45	154.7	22,661	96.6	477	16.83	3.08
HQ	5x3	4	14.8	6.88	101.8	23,580	67.0	442	15.59	4.34
HQ	5x4	4	14.8	10.22	151.3	22,739	86.1	589	20.78	3.89
HQ	5x4x3	4	14.8	13.26	196.2	21,763	82.4	710	25.04	3.62
HQ	5x4.5-BN	4	14.8	16.10	238.3	20,899	89.1	744	26.24	3.12
HQ	6x3	4	14.8	11.06	163.7	22,512	64.0	679	23.95	4.15
HQ	6x4.5	4	14.8	19.62	290.4	19,948	85.0	982	34.64	3.38

Figure C.1: Official Test Results for Cobra Motors

Appendix D

Inertias

$$\mathbb{I}_{inner} \tag{D.1a}$$

$$\mathbb{I}_{middle} \tag{D.1b}$$

$$\mathbb{I}_{body} \tag{D.1c}$$

Bibliography

[1] Yazan Al-Rihani. *Development of a dual axis tilt rotorcraft uav: Design, prototyping and control.*, volume 1. Cranfield University: School of Engineering, 2012.

[2] N. Amiri, A. Ramirez-Serrano, and Davies R. Modelling of opposed lateral and longitudinal tilting dual-fan unmanned aerial vehicle. *International Federation of Automatic Control*, pages 2054–2059, September 2011.

[3] APMCopter. Arducopter main page. Website: http://www.arducopter.co.uk/, 6 2016. Arducopter (APM) Official Website.

[4] E. Balasubramanian and R. Vasantharaj. Dynamic modelling and control of quadrotor. *International Journal of Engineering and Technology*, pages 63–39, February 2013.

[5] M. Bangura and R. Mahony. Non-linear dynamic modelling for high performance control of a quadrotor. In *Australasian Conference on Robotics and Automation, Victoria University of Wellington*. Victoria Unversity of Wellington, 12 2012. Published in Conference Proceedings.

[6] Mohd Ariffanan Basri, Abdul R. Husain, and Kumeresan A. Danapalasingam. Intelligent adaptive backstepping control for mimo uncertain non-linear quadrotor helicopter systems. *Institute of Measurement Control Transactions*, pages 1–17, 2014.

[7] BetaFlight. Betaflight fc4 repo, 2016. Forked from the CleanFlight repo.

[8] BLHeli. Blheli master branch (silabs), 2016.

[9] Charles Blouin and Eric Lanteigne. Pitch control on an oblique active tilting bi-rotor. *International Conference on Unmanned Aircraft Systems*, pages 791–799, May 2014.

[10] R. Bodrany, W. Steyn, and M. Crawford. In-orbit estimation of the inertia matrix and thruster parameters of uosat-12. In *Conference on Small Satellites*, volume 14, pages 1–11. American Institute of Aeronautics and Astronautics, 2000.

[11] Hossein Bolandi, Mohammed Rezaei, Rezo Mohsenipour, Hossein Nemati, and Seed Majid Smailzadeh. Attitude control of a quadrotor with optimized pid. *Intelligent Control and Automation*, pages 335–342, August 2013.

[12] S. Bouabdallah and R. Siegwart. Full control of a quadrotor. *IEEE International Conference on Intelligent Robots and Systems*, pages 153–158, 11 2007. Written for Autonomous Systems Lab at Swiss Federal Institute of Technology.

[13] Samir Bouabdallah, Andre Noth, and Roland Siegward. Pid vs lq control technqiues applies to an indoor micro quadrotor. *IEEE International Conference on Intelligent Robots and Systems*, pages 2451–2456, 9 2004.

[14] J. Brandt and M. Selig. Propeller performance data at low reynolds numbers. *American Institute of Aeronautics and Astronautics Sciences Meeting, 49th*, pages 1–18, January 2011.

[15] Jian Chen, Aman Behal, and Darren M. Dawson. Adaptive output feedback control for a class of mimo nonlinear systems. In *Proceedings of the American Control Conference*, pages 5300–5306, Minneapolis, Minnesota, US, 6 2006. American Control Conference.

[16] Arindam B. Chowdhury, Anil Kulhare, and Guarav Raina. A generalized control method for tilt-rotor uav stabilization. *IEEE International COnference on Cyber Technology in Automation, Control and Intelligent Systems*, pages 309–314, May 2012.

[17] CleanFlight. Cleanflight repo, 2016.

[18] Dominic Clifton. Spracing f3 deluxe flight controller, 2015.

[19] R.F. de Olivera, F.T. de Salvi, and E.M. Belo. Dynamic modelling, simulation and control of an autonomous quadcopter aircraft. *International Congress of Mechanical Engineering*, pages 1–9, November 2009.

[20] Innov8tive Designs. Cobra cm2208/2000 motors, 2016.

[21] Chen Diao, Bin Xian, Qiang Yin, Wei Zeng, Haotao Li, and Yungao Yang. A nonlinear adaptive control approach for quadrotor uavs. In *Asian Control Conference Proceedings*, volume 8, pages 223–228, Kaohsiung, Taiwan, 5 2011. Asian Control Conference.

[22] DJI Drones. Dji inspire one, 2016.

[23] DJI Drones. Dji phantom, 2016.

[24] Honeywell Solid State Electronics. Hmc5833 magnetometer datasheet. Advanced Information Data Sheet, 10 2010. Available From:`https://strawberry-linux.com/pub/HMC5883L.pdf`.

[25] Emil Fresk and George Nikolakopoulos. Full quaternion based attitude control for a quadrotor. *European Control Conference*, pages 3864–3869, 6 2013.

[26] Pau. S Gasco. *Development of a Dual Axis Tilt Rotorcraft UAV: Modelling, Simulation and Control*, volume 1. Cranfield University: School of Engineering, 2012.

[27] HiSystems GmbH. Mikrokopter quadroxl, 2016.

[28] Basile Graf. Quaternions and dynamics. Publication for Mathematics - Dynamical Systems, 2 2007.

[29] Gary R. Gress. Lift fans as gyroscopes for controlling compact vtol air vehicles: Overview and development status of oblique active tilting. In *American Helicopter Society Annual Forum*, volume 63, Virginia Beach, 5 2007. American Helicopter Society, American Helicopter Society Inc. Forum Proceedings.

[30] Gary R. Gress. *Passive Stabilization of VTOL Aircraft Having Obliquely Tilting Propellers*. University of Calgary, Department of Mechanical Engineering, Calgary, Alberta, 2014.

[31] Karsten Groekatthfer and Zizung Yoon. Introudction into quaternions for spacecraft attitude representation. *Technical University of Berlin: Department of Astronautics and Aeronatuics*, pages 1–16, May 2012.

[32] N. Guenard, T. Hamel, and V. Moreau. Dynamic modelling and control strategy for an x4-flyer. *International Conference on Control and Automation*, pages 141–146, June 2005.

[33] Drone HiTech. Xrotor 20a esc, 2016.

[34] HobbyKing. Orangerx rpm sensor, 2016.

[35] HobbyKing. Rotorstar super mini s-bec, 2016.

[36] HobbyKing. Signal converter module sbus-ppm-pwm, 2016.

[37] HobbyKing.com. Hobby king: The ultimate hobby experience, 2016.

[38] G. Hoffmann, H. Huang, S. Waslander, and C. Tomlin. Quadrotor helicopter flight dynamics and control: Theory and experiment. In *Guidance, Navigation and Control Conference and Exhibit*, pages 1–19, Hilton Head, South Carolina, 8 2010. American Institute of Aeronautics and Astronautics, American Institute of Aeronautics and Astronautics. Derivation of advanced aerodynamic affects on STARMAC Quadrotor Prototype.

[39] InvenSense Inc. Mpu6050 6-axis gyroscope/accelerometer datasheet. Product Specification Data Sheet, 8 2013. Available From:`https://www.cdiweb.com/datasheets/invensense/MPU-6050_DataSheet_V3%204.pdf`.

[40] Digi International. Xbee/xbee pro rf modules. Technical Data Sheet, 9 2009. Available From:`https://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-Datasheet.pdf`.

[41] W. Jia, Z. Ming, Y. Zhiwei, and L. Bin. Adaptive back-stepping lpv control of satellite attitude maneuvers with sum of squares. In *World Congress on Intelligent Control and Automation*, volume 8, pages 1747–1752. IEEE, 7 2010.

[42] Tor A. Johansen and Thor I. Fossen. Control allocation - a survery. *Automatica*, 45:10871103, 11 2012. Prepared for: Department of Engineering Cybernetics - Norwegian University of Science and Technology.

[43] Tor A. Johansen, Thor I. Fossen, and Petter Tondel. Efficient optimal constraint control allocation via multi-parametric programming. White paper, Department of Engineering Cybernetics, Nowegian University of Science and Technology, 2005.

[44] Tor A. Johansen and Johannes Tjnns. Adaptive control allocation. White paper, Department of Engineering Cybernetics, Nowegian University of Science and Technology, 2008.

[45] Farid Kendoul, Isabelle Fantoni, and Rogelio Lozano. Modeling and control of a small autonomous aircraft having two tilting rotors. *IEEE Conference on Decision and COntrol*, pages 8144–8149, December 2005.

[46] P. Krishnamurthy and F. Khorrami. Adaptive backstepping and theta-d based controllers for a tilt-rotor aircraft. *Mediterranean Conference on Control and Automation*, pages 540–545, June 2011.

[47] Raymond Kristiansen and Per J. Nicklasson. Satellite attitude control by quaternion-based backstepping. *American Control Coference*, N/A:907–912, 6 2005. Published by Department of Computer Science, Electrical Engineering and Space Technology; Narvik University College.

[48] Jack B. Kuipers. *Quaternions and Rotation Sequences: A Prier with Aplication to Orbital Aerospace and Virtual Reality*, pages 127–143. Princeton University Press, September 2002. Used for Quaternion and Rotation Matrix reference.

[49] Peter Lambert. Nakazawa, banton and jin, bai x. Technical Report N/A, Computer and Electrical Engineering: University of Victoria, Victoria, Canada, 12 2013.

[50] Prof Allan J. Laub. The moore-penrose pseudo inverse. UCLA Math33A Course Content, UCLA, Los Angeles, 3 2008. Course Notes cited from `http://www.math.ucla.edu/~laub/33a.2.12s/mppseudoinverse.pdf`.

[51] Jang-Ho Lee, Byoung-Mun Min, and Eung-Tai Kim. Autopilot design of tilt-rotor uav using particle swarm optimization method. *International COnference on Control, Automation and Systems*, pages 1629–1633, October 2007.

[52] LibrePilot. Openpilot/librepilot wiki. Website: http://opwiki.readthedocs.io/en/latest/index.html, 5 2016. Information wiki page for LibrePilot/OpenPilot firmware.

[53] Hyon Lim, Jaemann Park, Daewon Lee, and H.J. Kim. Build your own quadrotor. *IEEE ROBOTICS & AUTOMATION MAGAZINE*, pages 33–45, 9 2012. Publication on Opensource Autopilot systems.

[54] SteadiDrone PTY LTD. Steadidrone home, 2016.

[55] Teppo Luukkonen. Modelling and control of a quadcopter. Master's thesis, Aalto University: School of Science, Eepso, Finland, 8 2011. Independent research project in applied mathematics.

[56] Tarek Madani and Abdelaziz Benallegue. Backstepping control for a quadrotor helicopter. *International COnference on Intelligent Robots and Systems*, pages 3255–3260, October 2006.

[57] I. Mandre. Rigid body dynamics using euler's equations, rungekutta and quaternions, 2 2006.

[58] Carlos J. Mantas and Jose M. Puche. Artificial neural networks are zero-order tsk fuzzy systems. In *IEE Transactions on Fuzzy Systems*, volume 16, pages 630–644, 6 2008.

[59] Christopher G. Mayhew, Ricardo G. Sanfelice, and Andrew R. Teel. On quaternion based attitude control and the unwinding phenomenon. *American Control Conference*, pages 299–304, June 2011.

[60] Ashfaq A. Mian and Wang Daoboo. Modelling and backstepping-based nonlinear control of a 6dof quadrotor helicopter. *Chinese Journal of Aeronautics*, 21:261–268, 3 2008. Simulated Backstepping Control.

[61] Svein Rivli Napsholm. Prototype of a tiltrotor helicopter. Master's thesis, Norwegian University of Science and Technology: Department of Engineering Cybernetics, Norway, 1 2013.

[62] A. Nemati and M. Kumar. Modeling and control of a single axis tilting quadcopter. *American Control Conference*, pages 3077–3082, June 2014.

[63] Kenzo Nonami, Farid Kendoul, Satoshi Suzuki, Wei Wang, and Daisuke Nakazawa. *Autonomous Flying Robots: Unmanned Aerial Vehicles and Micro Aerial Vehicles*, chapter 2, pages 44–48. Springer Japan, 1 edition, 2010. References to Cyclic-Pitch Control relevant subsections.

[64] Kenzo Nonami, Farid Kendoul, Satoshi Suzuki, Wei Wang, and Daisuke Nakazawa. *Autonomous Flying Robots: Unmanned Aerial Vehicles and Micro Aerial Vehicles*, chapter 8, page 166. Springer Japan, 1 edition, 2010.

[65] Gustavo P. Oliveira. Quadcopter civil applications. Master's thesis, Informatics and Computer Engineering: University of Portugal, Portugal, 2 2014.

[66] OrangeRx. Orangerx r615x receiver. User Manual, 10 2014. Available From:`http://www.hobbyking.com/hobbyking/store/uploads/672761531X1606554X18.pdf`.

[67] Christos Papachristos, Kostas Alexis, and Anthony Tzes. Design and experimental attitude control of an unmanned tilt-rotor aerial vehicle. *International Conference on Advanced Robotics*, pages 465–470, June 2011.

[68] Parth N. Patel, Malav A. Patel, Rahul M. Faldu, and Yash R. Dave. Quadcopter for agricultural surveilance. In *Advance in Electronic and Electrical Engineering*, volume 3, India, 2013.

[69] J. Peraire and S. Widnall. 3d rigid body dynamics: Euler angles. Lecture notes for Dynamics Course, 2009. Dynamics course notes, fall 2007.

[70] J. Peraire and S. Widnall. 3d rigid body dynamics: The inertia tensor. Lecture notes for Dynamics Course, 2009. Dynamics course notes, fall 2007.

[71] D. Peters. Eigth amendment of the civil aviation regulations. Goverment Gazette Notice, 5 2015. In Amendment to the Civil Aviation Act, 2009 (Act No.13 of 2009).

[72] Jean-Baptiste Pomet and Laurent Praly. Adaptive nonlinear regulation: Estimation from the lyupanov equation. In *IEEE Transactions on Automatic Control*, volume 37, pages 729–740. IEEE, 6 1992.

[73] P. Pounds, R. Mahony, P. Hynes, and J. Roberts. Design of a four-rotor aerial robot. *Australasian Conference on Robotics and Automation*, pages 145–150, November 2002.

[74] Dimitry Prof. Garanin. Rotational motion of rigid bodies. Analytical Dynamics Course Notes, 11 2008. Content for City University of New York.

[75] Beard Randal. Quadrotor dynamics and control. Report, Brigham Young University, 2 2008. Part of the Electrical and Computer Engineering Commons.

[76] O. Rawashdeh, H.C. Yang, R. AbouSleiman, and B. Sababha. Microraptor: A low cost autonomous quadrotor system. *International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, pages 1–8, August 2009.

[77] Anastasia Razinkove, Igor Gaponov, and Hyun-Chan Cho. Adaptive control over quadcopter uav under disturbances. *International Conference on Control, Automation and Systems*, pages 386–390, October 2014.

[78] M.K. Rwigema. Propeller blade element momentum theory with vortex wake deflection. *International Congress of the Aeronautical Sciences, 27th*, pages 1–9, January 2010.

[79] M. Ryll, H. Bulthoff, and P. Robuffo Giordano. Modelling and control of a quadrotor uav with tilting propellers. *IEEE International Conference on Robotics and Automation*, pages 4606–4613, May 2012.

[80] M. Ryll, H. Bulthoff, and P. Robuffo Giordano. First flight tests of a quadrotor uav with tilting propellers. *IEEE International Conference on Robotics and Automation*, pages 295–302, May 2013.

[81] A. Sanchez, J. Escareo, O. Garcia, and R. Lozano. Autonomous hovering of a noncyclic tiltrotor uav: Modeling, control and implementation. *The International Federation of Automatic Control*, pages 803–808, July 2008.

[82] Puneet Singla, Daniele Mortari, and John L. Junkins. How to avoid singularity when using euler angles? *Advances in the Astronautical Sciences*, pages 1409–1426, January 2005.

[83] Measurement Speacialties. Ms5611 barometric pressure sensor. Technical Data Sheet, 10 2012. Available From:`http://www.amsys.info/sheets/amsys.en.ms5611_01ba03.pdf`.

[84] STMicroElectronics. St-link/v2 in circuit debugger/programmer for stm32.

[85] STMicroElectronics. Rm0316 reference manual. Online Micro-Controller Reference Manual, 3 2016. Available From:`http://www.st.com/content/st_com/en/products/microcontrollers/stm32-32-bit-arm-cortex-mcus/stm32f3-series/stm32f303.html?querycriteria=productId=LN1531`.

[86] Prof S. Tavoularis. Reynolds transportation theorem. Course Notes on MCG3350 - Fluid Mechanics 1, 2008. Fluid Mechanics notes, fall 2008, `http://web.mit.edu/1.63/www/Lec-notes/chap1_basics/1-3trans-thm.pdf`.

[87] Abdelhamid Tayebi and Stephen McGilvray. Attitude stabilization of a vtol quadrotor aircraft. *IEEE Transactions on Control Systems Technology*, pages 562–571, May 2006.

[88] Stephen T. Thornton and Jerry B. Marion. *Classical Dynamics of Particles and Systems*, chapter 7, pages 228–289. Thompson Brooks/Cole, 5 edition, 2003.

[89] John Ting-Yung Wen and Kenneth Kreutz-Delgado. The attitude control problem. *IEEE Transactions on Automatic Control*, pages 1148–1162, October 1991.

[90] David Tong. Lagrange formalism. Lectures of Classic Dynamics, Course Notes, 2005. Classical Mechanics Notes.

[91] P. Tsiotras, M. Corless, and J.m Longuski. A novel approach to the attitude control of axisymmetric spacecraft. *Automatica*, 31:1099–1112, 3 1995. Control Automatica, Printed in Great Britan.

[92] Ultimaker. Ultimaker v2+ product page, 2016.

[93] E. van Kampen and M. M. van Paassen. Ae4301: Automatic flight control system design. Delft Centre for Systems and Control; MSc Notes, 1 2008. Course Notes cited from: `http://aerostudents.com/master/advancedFlightControl.php`.

[94] Ronny Votel and Doug Sinclair. Comparison of control moment gyros and reaction wheels for small earth-observing satellites. In *Conference on Small Satellites*, volume 26, pages 1–7. Utah State University, 8 2012. Open access on AIAA conference website.

[95] Tao Wang, Tao Zhao, Du Hao, and Mingxi Wang. Transformable aerial vehicle, 09 2014.

[96] X. Xiaozhu, L. Zaozhen, and C. Weining. Intelligent adaptive backstepping controller design based on the adaptive particle swarm optimization. *Chinese Control and Decision Conference*, pages 13–17, September 2009.

[97] Song Xin and Zou Zaojian. A fuzzy sliding mode controller with adaptive disturance approximation for an underwater robot. In *International Asia Conference on Informatics in Control, Automation and Robotics*, volume 2, pages 50–53, 10 2010.