

# Dual-Axis Tilting Quadrotor Aircraft

An investigation into the overactuatedness and control thereof



**Nicholas Von Klemperer**

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

**April 2017**

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

*Keywords:* Non-linear, control, allocation, quadrotor, UAV



*”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 or cited, any and all illustrations or diagrams demonstrated in this work are my own productions.
  - (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>
    - 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>
    - All CAD design files & assemblies: <https://grabcad.com/nick.vk-1>

---

Nicholas Von Klemperer  
Department of Electrical Engineering  
University of Cape Town  
Wednesday 5<sup>th</sup> April, 2017

# Abstract

## Dual-Axis Tilting Quadrotor Aircraft

Nicholas Von Klemperer

*Wednesday 5<sup>th</sup> April, 2017*

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 and 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 dissertation, with unique 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 later include filtering state estimation effects on those dynamics*). 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 hobbyist components, thereafter conclusions are drawn on the success or failure of the design.

The purpose of the investigation is evaluation of practicality and feasibility for such a design, most importantly whether the complexity of the mechanical design is an acceptable compromise for the additional degrees of control actuation introduced. The outcome of the build is the assertion of economic feasibility (controller cost and effort) of such a prototype to expand the range of a quadrotor's motion. The design and control derivation 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 are listed here...*

# Acknowledgements

# Nomenclature

In order of appearance:

DOF - Degree of Freedom(s)

$\mu$ C - micro-controller

UAV - Unmanned aerial vehicle

SISO - Single input single output, control loop

MEMS - Micro-electromechanical system

DIY - Do it yourself

VTOL - Vertical takeoff/landing

IMU - Inertial measurement unit

BLDC - Brushless direct current, motor type

KV - Kilo-volt, BLDC motor rating

$\mu$ C - Micro-controller shorthand

PWM - Pulse width modulation

CH - Channel, radio control & PWM signals typically

RC - Radio control

OAT - Opposed active tilting

dOAT - Dual axis opposed active tilting

PD - Proportional derivative, control law

PID - Proportional integral derivative, control law

IBC - Ideal backstepping control

ABC - Adaptive backstepping control

PSO - Particle swarm optimization, gradient free genetic algorithm

BEM - Blade element theory

ESC - Electronic speed controller

MPC - Model predictive control

LQR - Linear quadratic regulator

LCF - Lyupanov candidate function

ITAE - Integral time additive error

TSK - Takagi-Sugeno-kang

I/O - Input/Output

RPM - Revolution Per Minute

RPS - Revolution Per Second

W.R.T - With respect to

LCF - Lyupanov Candidate Function

*iff* - If and only if

P.D - Positive definite, NOT proportional derivative

S.T - such that

FTC - Fault Tolerant Control

# Symbols

Propeller Rotational Speed:  $\Omega_i$  [rpm] for motors:  $i \in [1, 2, 3, 4]$

*Rotational speeds in [RPS] are used for Blade Element Theory Calculations in Chapter:3*

Net body torque:  $\mu \vec{\tau} = [\tau_\phi \ \tau_\theta \ \tau_\psi]^T \in \mathcal{F}^b$

Net body thrust:  $\mu \vec{T} = [T_x \ T_y \ T_z]^T \in \mathcal{F}^b$

Body Position:  $\vec{E} = [x \ y \ z]^T \in \mathcal{F}^I$

Euler Angles:  $\vec{\mathcal{E}} = [\phi \ \theta \ \psi]^T \in \mathcal{F}^{I,v1,b}$

Servo 1 Position:  $\lambda_i$  [rad]

Servo 2 Position:  $\alpha_i$  [rad]

Motor module actuator positions:  $[\Omega_i \ \lambda_i \ \alpha_i]^T \in \mathcal{F}^{M_i}$

Actuator matrix:  $u = [M_1 \ \dots \ M_4]^T \in \mathbb{U}^{12}$

Motor module displacement arm:  $\vec{L}_{arm} = 195.16$  [mm]

Euler Rates:  $\frac{d}{dt} \vec{\eta} = \dot{\vec{\eta}} = \Phi(\eta) \dot{\omega}_b = [\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T \in \mathcal{F}^{v1,v2,I}$

Angular Velocity:  $\omega = [p \ q \ r]^T \in \mathcal{F}^b$

Linear Velocity:  $\nu = [u \ v \ w]^T \in \mathcal{F}^b$

# Contents

<b>Declaration</b>	ii
<b>Abstract</b>	iii
<b>Acknowledgements</b>	iv
<b>Nomenclature</b>	v
<b>Symbols</b>	vi
<b>1 Introduction</b>	1
1.1 Foreword . . . . .	1
1.1.1 A Brief Background to the Study . . . . .	1
1.1.2 Research Questions & Hypotheses . . . . .	2
1.1.3 Significance of Study . . . . .	3
1.1.4 Scope and Limitations . . . . .	4
1.2 Literature Review . . . . .	7
1.2.1 Existing & Related Work . . . . .	7
1.2.2 Notable Quadrotor Control Implementations . . . . .	10
<b>2 Prototype Design</b>	15
2.1 Design . . . . .	15
2.1.1 Actuation Functionality . . . . .	16
2.2 Reference Frames Used . . . . .	18
2.2.1 Reference Frames Convention . . . . .	18
2.2.2 Motor Axis Layout . . . . .	21
2.3 Inertial Matrices & Masses . . . . .	24

2.4 Electronics . . . . .	31
2.4.1 Actuator Transfer Functions . . . . .	35
<b>3 Kinematics &amp; Dynamics</b>	<b>40</b>
3.1 Rigid Body Dynamics . . . . .	40
3.1.1 Lagrange Derivation . . . . .	40
3.1.2 Rotation Matrix Singularity . . . . .	43
3.1.3 Quaternion Dynamics . . . . .	44
3.1.4 Quaternion Unwinding . . . . .	46
3.2 Multibody Nonlinearities . . . . .	47
3.2.1 Relative Rotational Gyroscopic & Inertial Torques . . . . .	47
3.3 Aerodynamics . . . . .	52
3.3.1 Propeller Torque and Thrust . . . . .	52
3.3.2 Hinged Propeller Conning & Flapping . . . . .	58
3.3.3 Drag . . . . .	59
3.4 Consolidated Model . . . . .	60
<b>4 Controller Development</b>	<b>61</b>
4.1 Stability . . . . .	61
4.2 Control Loop . . . . .	61
4.3 Control Plant Inputs . . . . .	63
4.4 Lyapunov Stability Theorem . . . . .	65
4.5 Attitude Control . . . . .	66
4.5.1 The Attitude Control Problem . . . . .	66
4.5.2 Linear Controllers . . . . .	66
4.5.3 Non-linear Controllers . . . . .	71
4.6 Position Control . . . . .	74
4.6.1 PD Controller . . . . .	75
4.6.2 Adaptive Backstepping Controller . . . . .	76
4.7 Controller Allocation . . . . .	78
4.7.1 Pseudo Inverse Allocator . . . . .	81
4.7.2 Weighted Pseudo Inverse Allocator . . . . .	83

4.7.3	Priority Norm Inverse Allocator . . . . .	84
4.7.4	Non-linear Plant Control Allocation . . . . .	86
4.7.5	Online Optimized Secondary Goal Allocator . . . . .	86
<b>5</b>	<b>Simulations &amp; Results</b>	<b>87</b>
5.1	Controller Tuning . . . . .	88
5.1.1	Partical Swarm Based Optimization . . . . .	88
5.1.2	Performance Metric . . . . .	88
5.1.3	Global & Local Minima . . . . .	88
5.1.4	Fmincon Differences . . . . .	88
5.1.5	PD . . . . .	88
5.1.6	XPD . . . . .	88
5.1.7	IBC . . . . .	88
5.1.8	ABC . . . . .	88
5.2	Allocator Optimization . . . . .	88
5.3	Simulation Block . . . . .	88
5.4	Optimized Controller Results . . . . .	88
5.4.1	Attitude Control . . . . .	88
5.4.2	Position Control . . . . .	88
5.4.3	Allocator Performance . . . . .	88
5.5	State Estimation . . . . .	88
5.5.1	Attitude Control Results . . . . .	88
5.5.2	Autopilot Outcome . . . . .	88
<b>6</b>	<b>Prototype Flight Results</b>	<b>89</b>
<b>7</b>	<b>Conclusion</b>	<b>90</b>
<b>A</b>	<b>Expanded Equations</b>	<b>91</b>
A.1	Standard Quadrotor Dynamics . . . . .	91
A.2	Blade-Element Momentum Expansion . . . . .	93
A.3	Induced Torques . . . . .	93
A.4	Euler-Angles from Quaternions . . . . .	95

<b>B Design Bill of Materials</b>	<b>96</b>
B.1 Parts List . . . . .	96
B.2 F3 Deluxe Schematic Diagram . . . . .	101
<b>C System ID Test Data</b>	<b>102</b>
C.1 Thrust and Torque Test Data . . . . .	102
C.2 Cobra CM2208-200KV Thrust Data . . . . .	103

# List of Figures

1.1	Bell/Boeing V22 Osprey actuation, notations pertinent to patent [95]	2
1.2	Mechanical actuators	5
1.3	General structure for opposed tilting platform, taken from [44]	7
1.4	DJI Inspire1, the notations are with regards to the DJI patent [130]	8
1.5	Tilt-rotor mechanisms	9
1.6	Dual-axis tilt-rotor mechanism used in [38]	9
1.7	ArduCopter PI control structure for pitch angle $\theta$ ; from [74]	11
2.1	Isometric view of the prototype design	15
2.2	Tilting rotor design	16
2.3	Difference between propeller and motor planes	17
2.4	Motor module assembly	17
2.5	Digital and analogue servo timing	17
2.6	Inertial and body reference frames	18
2.7	Aligned motor frame axes	21
2.8	Intermediate motor frames	22
2.9	Body frame axes layout	23
2.10	Inertial measurement references	24
2.11	Inertial measurement references cont.	26
2.12	Body structure inertias	27
2.13	Inertial, mass and motor modules respective centers	28
2.14	Hardware schematic diagram	31
2.15	SPRacing F3 deluxe layout	32
2.16	SBUS converter & 6CH receiver	32
2.17	S.BUS data stream	33

2.18 BLDC electronic speed controllers . . . . .	34
2.19 RPM sensor calibration plots . . . . .	35
2.20 Servo transfer function test rig . . . . .	35
2.21 Unloaded servo transfer characteristics . . . . .	36
2.22 Servo block diagram . . . . .	37
2.23 Inner ring servo characteristics . . . . .	37
2.24 Middle ring servo characteristics . . . . .	37
2.25 BLDC rpm speed calibration and transfer function rig . . . . .	38
2.26 BLDC motor characteristics . . . . .	39
3.1 Mechanical gimbal lock . . . . .	43
3.2 Exploded inertial bodies for $\vec{\tau}_\lambda$ . . . . .	48
3.3 Exploded inertial bodies for $\vec{\tau}_\alpha$ . . . . .	50
3.4 Propeller types . . . . .	53
3.5 Disc Actuator Propeller Planar Flow . . . . .	53
3.6 Blade element profile at radius r . . . . .	54
3.7 Power & thrust coefficients . . . . .	56
3.8 Static propeller tests . . . . .	57
3.9 Propeller blade flapping; from [55] . . . . .	58
3.10 Propeller coning . . . . .	59
4.1 Generalized control loop with allocation . . . . .	63
4.2 Abstracted control block . . . . .	64
4.3 Weighting matrix biasing . . . . .	83
4.4 Hover conditions W.R.T the inertial frame $\mathcal{F}^I$ . . . . .	84
4.5 Hover conditions W.R.T the body frame $\mathcal{F}^b$ . . . . .	85
4.6 Allocation loop iteration . . . . .	86
A.1 Torque response rig . . . . .	94
A.2 Torque responses for inner and middle rings . . . . .	94
B.1 Bearing Bracket Inner Ring Assembly . . . . .	98
B.2 Servo Bracket Inner Ring Assembly . . . . .	98

B.3	Servo Bracket Middle Ring Assembly . . . . .	98
B.4	Bearing Holder Middle Ring Assembly . . . . .	98
B.5	Servo Mount Middle Ring Assembly . . . . .	98
B.6	Bearing Shaft Middle Ring Assembly . . . . .	98
B.7	Bearing Holder Damping Assembly . . . . .	99
B.8	Servo Mount Damping Assembly . . . . .	99
B.9	Servo Mount Damping Bracket . . . . .	99
B.10	Bearing Holder Damping Bracket . . . . .	99
B.11	Arm Mount Damping Bracket . . . . .	99
B.12	Frame Brackets . . . . .	99
B.13	F3 Deluxe Flight Controller Hardware Schematic . . . . .	101
C.1	Clockwise and counterclockwise rotation tests . . . . .	102
C.2	Official Test Results for Cobra Motors . . . . .	103

# List of Tables

1.1	A breakdown of common attitude controllers . . . . .	11
B.1	Parts List . . . . .	96
B.2	3D Printed Parts . . . . .	97
B.3	Inner & Middle Ring Assemblies . . . . .	98
B.4	Damping Assemblies . . . . .	99
B.5	Laser Cut Damping Brackets . . . . .	99
B.6	Laser Cut Parts . . . . .	100

# Chapter 1

## Introduction

### 1.1 Foreword

#### 1.1.1 A Brief Background to the Study

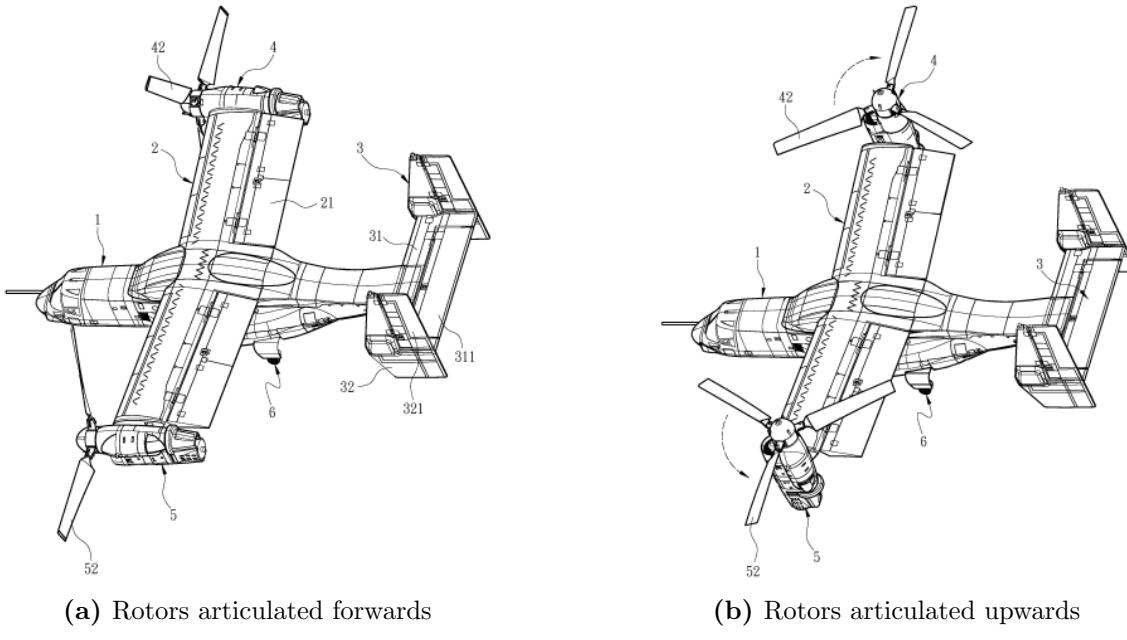
A popular topic for current control and automation research is that of quadrotor UAVs. Attitude control of a quadrotor poses a unique 6-DOF control problem, to be solved with an under-actuated 4-DOF system. As a result the pitch,  $\phi$ , and roll,  $\theta$ , plants are not directly controllable. The 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 translational and angular motions arise from gyroscopic torques and Coriolis accelerations (Sec: 3.2.1). Such effects are mostly negligible around the origin, hence the origin trim point decouples the system's nonlinearities. The control system can therefore reduce each state variable,  $\vec{x}_b = [x \ y \ z \ \phi \ \theta \ \psi]^T$ , to independent single-input single-output (*SISO*) plants. Those simplifications are derived in the Appendix:A.1.

As almost every quadrotor research paper mentions, the recent interest in the platform is due to increased availability of micro-electromechanical systems (*MEMS*) and low-cost microprocessor systems. These technical advancements accomodate onboard state estimation and control algorithm processes in real time. Developmental progress in quadrotors and, to a lesser extent unmanned aerial vehicles (*UAVs*) in general, has led to rapidly growing enthusiast communities. For example; HobbyKing [54] is now a name synonymous with providing custom DIY hobbyist quadrotor assembly kits and frames, no longer retailing only prebuilt commercial products like DJI Phantom [34] or ParrotAR [1] drones.

The avenue for potential application of both fixed wing and vertical take-off and landing (*VTOL*) UAVs is expansive; supporting civil [89], agricultural [94] and security [70] industries and not just recreational hobbyists. The quadrotor design 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<sup>th</sup> amendment of civil aviation laws [98], commercial use of UAVs is now both legalized and regulated. Research into any non-trivial aspect of the field will therefore be extremely valuable to the field as a whole. Large scale quadrotor, hexrotor and even octocopter UAVs are popular intermediate choices for aerial cinematography and other high payload capacity applications. The cost of a commercial drones such as the SteadiDrone Maverik [77] are significantly less than a chartered helicopter, used to achieve 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 aerial structures. When increasing the size of any vehicle, its performance is adversely affected if actuation rates are not proportionately increased.

### 1.1.2 Research Questions & Hypotheses

The difficulty with quadrotor control is that fundamentally, from their uncertainty and underactuation, they are ill-posed for 6 degree of freedom (*DOF*) setpoint tracking. A quadrotor inherently has only four controllable inputs; each propeller's rotational speed,  $\Omega_{1,2,3,4}$ , which are then abstracted to a net virtual control input net torque,  $\vec{\tau}_\mu = [\tau_\phi \ \tau_\theta \ \tau_\psi]^T$ , and a perpendicular heave thrust  $\vec{T}_\mu = \sum_{i=1}^4 T(\Omega_i)$  in the  $\hat{z}_b$  direction. Those four inputs are then used to effect both the translational X-Y-Z positions,  $\vec{x} = [x \ y \ z]^T$ , and angular pitch, roll and yaw attitude rotations,  $\vec{\eta} = [\phi \ \theta \ \psi]^T$ . Pitch and roll torques,  $\tau_\phi$  and  $\tau_\theta$  respectively, are produced from differential thrusts of each opposing propellers. Yawing torque,  $\tau_\psi$ , is induced only by the sum of aerodynamic drag torques about each propeller's rotational axis. Aerodynamic drag and thrust responses are highly non-linear (detailed later in Sec:3.3.1) and difficult to approximate as sources of control torque. As a result the body's yaw channel control is depreciated. Stemming from the system's under-actuation, the attitude control problem becomes a zero set point problem because any other attempt to track attitude is ill-posed.



**Figure 1.1:** Bell/Boeing V22 Osprey actuation, notations pertinent to patent [95]

The aim of this dissertation is to implement attitude and position dynamic set point tracking on a quadrotor UAV by solving the problem of its inherent under-actuation. Inspired by Boeing/Bell Helicopter's V22 Osprey (Fig:1.1) and the tilting articulation of its propellers, the prototype design proposed here (detail in Sec:2.1) introduces two additional actuators for each of the quadrotor's four lift propellers. Specifically, adding rotations about the  $\hat{X}$  and  $\hat{Y}$  axes for each motor/propeller pair, the resultant are four separately articulated 3-dimensional thrust vectors instead of a bound perpendicular net heave force. 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,  $\vec{x}_d(t) = [x \ y \ z \ \psi \ \theta \ \phi]^T$ , the error state  $\vec{x}_e(t) = \vec{x}_d(t) - \vec{x}_b(t)$  is asymptotically stable. Trajectory stability is explicitly defined later in the context of Lyapunov stability analysis in Sec:4.4.

$$\lim_{t \rightarrow \infty} \vec{x}_e(t) = \vec{0} \quad \forall \vec{x} \in \mathbb{R}^n \quad (1.1)$$

Where  $n$  is number of the degrees of freedom the system has, typically a 6-DOF plant for rigid bodies. The over-actuation brings about the need for a control allocation scheme, one 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 which the project could 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 dynamic modeling of the system; making as few assumptions as possible to the non-linear dynamics involved in the quadrotor's motion and its operational conditions. Common linearizations often applied to the quadrotor's control plant will not hold true for the more aggressive maneuvers; they are dependent on small angle approximations and neglect 2<sup>nd</sup> or higher order effects. To produce a stabilizing control law solution there first needs to be a dynamic model that incorporates both multibody and actuator dynamics, against which the controller efficacy can be tested. The final key outcomes for the project are; the prototype design, its dynamic model and simulation analysis, the resultant control law produced and finally conclusions drawn on all of the above.

For a rigidly connected multibody system with rotational joints between sub-bodies, the induced relative motion between those sub-bodies will produce a lot of unwanted dynamics like inertial and gyroscopic responses, amongst others... A rotating propeller will respond to pitching or rolling much like a Control Moment Gyroscope [129] or a flywheel, producing a precipitating torque cross product. A less trivial aspect which is occasionally considered are the aerodynamic effects produced from the propeller's aerofoil profile. Such induced responses manifest normal to the propeller's rotational axis. Those aspects are not typically compensated for due to a quadrotor's fundamental co-planar propeller counter-rotating pairs which mostly negate such effects. A plant dependent control law is needed for dynamic compensation which reduces uncertainty associated with the subsequent stability proof.

### 1.1.3 Significance of Study

Owing to the huge popularity of quadrotor platforms as research tools (i.e [10,18,47], etc...), any work that builds on UAV and quadrotor fundamentals will prove to be valuable. With that being said, there is already a plethora of research on the subject of linear and non-linear control techniques for quadrotor platforms (surveyed in Table:1.1). Attitude control loops are the most common topic for research, requiring a unique under-actuated solution and mostly linearized around the origin (Appendix:A.1). Far less common is the application of optimal flight path and trajectory planning to a quadrotor's (*augmented*) autopilot system. The difficulty and ill-posed aspect of a quadrotor's attitude control does not hold true for its position plant, so standard techniques can be applied for waypoint and trajectory planning once the attitude control problem has been addressed.

The most significant aspect of this project is the attitude control, discussed later in Sec:4.5. The over-actuation of the proposed design and, more critically, the manner in which the controller's commanded (virtual) output is distributed among those control effectors would, at the time of writing, appear to be the first of its kind. Otherwise known as control allocation, the requirements of the distribution algorithm(s) are outlined in Sec:4.7. Dynamic setpoint attitude control for aerospace bodies is not a subject heavily researched outside the field of satellite attitude control. Even papers that propose similarly complicated mechanical over-actuation (expanded upon in next in the literature review, Sec:1.2) hardly broach the topic of tracking attitude set points away from the origin.

The control plant presented in this dissertation, developed in Chapter:4, does indeed close both the position and attitude control loops. There is, however, no consideration of trajectory generation nor flight path planning as such topics are well discussed elsewhere. Once closed loop position and attitude control have been achieved, the control algorithms can be adjusted to incorporate higher order state derivative (acceleration, jerk and jounce) tracking needed for nodal waypoint planning. The heuristics involved with flight path planning are well documented and their application is an easily implemented task [48, 78, 110].

Where possible, the system identification and control (*design* and *allocation*) for this project is kept modular and generally applicable. The intention is that its pertinence falls not only within the UAV field but also to any aerospace attitude control plant, rigid or otherwise. Hopefully the investigation can be expanded upon with more focused research on one of the subsystems without compromising

the stability of the whole plant. Provisionally, an obvious outcome which the project 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...

Moreover, this dissertation could provide greater insight into higher bandwidth actuation and hence faster control responses for larger aerospace bodies. Any standard quadrotor uses differential thrusts to develop a torque about its body. Such actuation suffers a second order inertial response when the propellers accelerate or decelerate;  $\vec{\tau}_p = \mathbb{I}_p \dot{\Omega}_i$  for  $i \in [1 : 4]$ . Prioritizing pitching the propeller's principle axis of rotation in lieu of changing the rotational speed could potentially improve the actuator plant rate response. This is entirely dependent on how the allocator block is prioritized (presented in Sec:4.7). The exact effects of different actuator prioritization and distribution in the context of aerospace control are, at the time of writing, unique to this research.

### 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 control research question is to apply dynamic attitude setpoint control to the quadrotor platform. Stemming from this is an investigation into the kinematics that are potentially influenced by the design changes and the structure's relative motion. In order to apply correct control theory to achieve the attitude tracking on the physical prototype, plant dynamics must first be identified for the controller to be optimized correctly. Aspects of the mechanical design are detailed in the next chapter; Ch:2.1. There is no scope beyond the cursory investigation for materials analysis or stress testing of the design. This dissertation's scope focuses mainly on the control derivation and embedded systems design/implementation and 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 Sec:1.1.3; trajectory and flight path planning are not ubiquitous with this dissertation. Derivations for the differential equations of motion which dictate a 6-DOF body's movement are applicable to any aerospace body, rigid or otherwise. Some particular standards are used, like Z-Y-X Euler Aerospace rotational matrix sequences, all of which are covered in Sec:2.2. The control plant is stabilized with non-linear state-space control techniques in the time domain, aided and justified by Lyapunov stability theorem [15, 83, 103]. Alternative solutions using model predictive control (*MPC*) or quantitative feedback theory (*QFT*) could provide more refined or effective controllers, however they are not discussed here 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 App:A.1 but beyond that there are no further discussions. Any comparisons between non-zero and zero set-point attitude controller efficacy for quadrotors are difficult as the fundamental objectives are in stark contrast with one another.

Arguably the most important and potentially 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 an entire set of compatible actuator solutions,  $u \in \mathbb{U} \in \mathbb{R}^{12}$ , 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 combined to achieve the desired 6 control plant inputs, specifically net force  $\vec{F}_{net}$  and net torque  $\vec{\tau}_{net}$  acting on the system. Appropriate techniques are first derived in Sec:4.7 then simulated and compared before a final solution is implemented in Section:5.4. It is not a comprehensive survey 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.

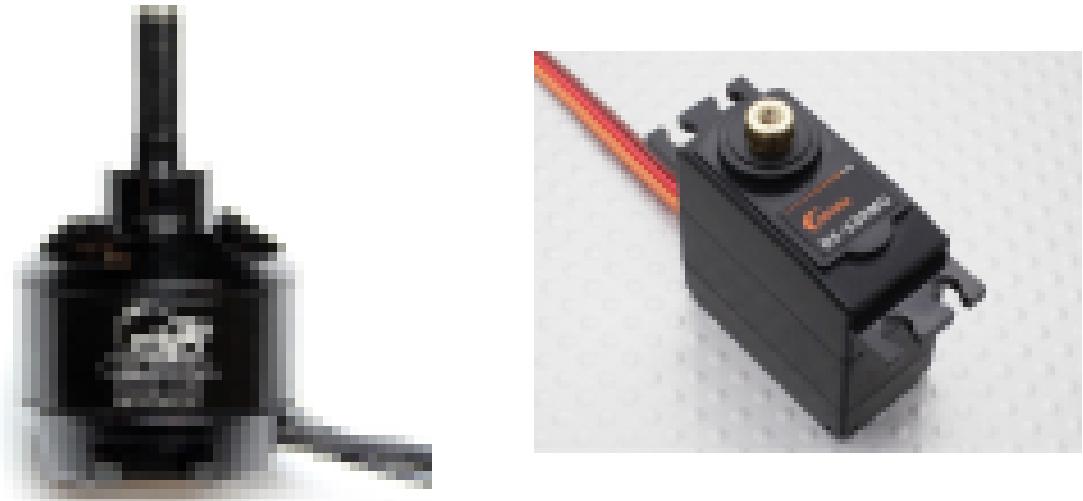
With regards to the prototype design, in Sec2.1, it is assumed that certain aspects are readily available and require no design/development. Particularly the state estimation, updated through a 4-camera positioning system fused with a 6-axis IMU through Kalman Filtering (Sec:5.5), is assumed to accurate 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 thrusts which are required to keep an aircraft aloft and oppose the net gravitational force are obviously dependent on the body's net weight. The steady state actuator rates ought to be far less than saturation conditions to ensure sufficient actuator headroom to implement control actuators. 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 effectiveness 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 is 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 design 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 bladed  $6 \times 4.5$  inch small diameter propellers constrained the maximal overall dimensions of the prototype, but as a consequence required very high RPM motors. Specifically a set of four Cobra-2208/2000KV [30] brushless direct current (*BLDC*) motors are used for lift actuation (Fig:1.2a).

A direct consequence of that decision is, provisionally based on thrust tests, the net thrust actuation disposable to the control loop is limited to around  $950 \text{ [g]} \approx 9.5 \text{ [N]}$ , per motor at  $14.1 \text{ [V]}$ . That thrust test data is provided from the official Cobra motor's website, [30], included in App:C.2 and verified independently through testing in Sec:3.3.1. It is critical to ensure the control block does not induce saturation of those BLDC motors. The frame weight needs to be under 50% of the maximum available thrust; or roughly below  $2 \text{ [kg]}$ . Saturation conditions are detailed later in Sec:4.7.



(a) Cobra CM2208/2000KV BLDC motor [40]

(b) Corona DS-339MG digital servo [54]

**Figure 1.2:** Mechanical actuators

Another aspect of limitations produced by design decisions made, mostly to reduce prototype costs and weight, is to use of  $180^\circ$  rotation servo motors. Here Corona DS-339MG metal gear digital servos (Fig:1.2b) are used. The servos are for each individual motor's  $\hat{X}_{M_i}$  and  $\hat{X}_{M_i}$  axial pitch and roll actuators respectively, terms  $\lambda_i$  and  $\alpha_i$  to differentiate from body pitch  $\phi$  and roll  $\theta$ . Servos act in place of either BLDC gimbal or stepper motors with closed loop position control. The latter pair would both accommodate for continuous ( $> 2\pi$ ) rotation but would need their own control design. Continuous rotation (velocity controlled) servos could otherwise be used but would similarly require angular rotation feedback. Any rotations beyond  $2\pi$  would similarly require slip rings to transmit power throughout rotational movement to avoid mechanical interference from connection lines. Implementing such a design whilst maintaining an acceptable weight would prove too costly nor would it provide additional insight attained from experimental testing. The effect of servo rotation limits can be evaluated in simulation and if it proves to be significant, continuous rotation can be implemented...

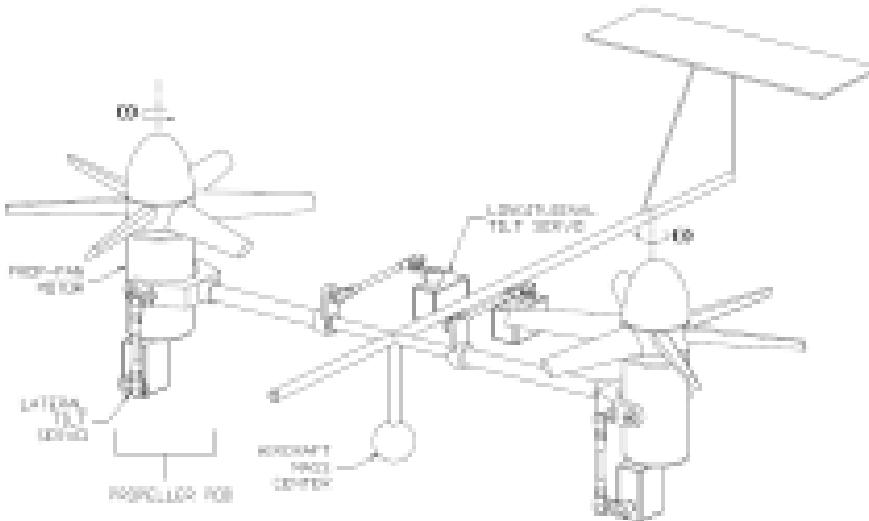
Discrete elements for the whole system could potentially limit performance but are mitigated where possible. For example analogue servos have an associated 1 [ms] dead time from their 50 [Hz] refresh rate. That can be addressed by using faster, albeit more expensive, digital servos which samples at 330 [Hz]. The prototype's flight controller needs to provide 12 pulse-width modulated (*PWM*) output compare channels for the 8 servos and 4 BLDC speed controllers. State updates from a ground control station and a fail safe 6-channel (*CH*) radio control receiver module also needs to be processed by the micro-controller ( $\mu C$ ) system. Particular attention is paid to the embedded system design and layout in Sec:2.4.

## 1.2 Literature Review

### 1.2.1 Existing & Related Work

The field of transformable aerospace frames is not new, with many commercial examples seeing successes over their operational life span. The most notable tilting-rotor vehicle is the Boeing/Bell V22 Osprey [39] 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 1.3:** General structure for opposed tilting platform, taken from [44]

Research into birotor vehicles (Fig:1.3) with ancillary lift propeller actuation is oft termed *Opposed Active Tilting* or *OAT*. Such a rotorcraft's mechanical design applies either a single *oblique*  $45^\circ$  tilting axis relative to the body; [13, 45, 65], or a *lateral* tilting axis, adjacent to the body; [26, 67, 93, 112]. 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 independent propeller pitch and roll motions to further expand the actuation suite, [3, 44]. A birotor is sometimes considered preferable to higher degree of freedom multirotor platforms due to their reduced controller effort. However the controller plant derivation, typically requiring feedback linearization and virtual plant abstraction, often detracts from the quality and effectiveness of its stability solution as a result of the birotor's underactuation.

Birotor attitude control mostly introduces plant independent PD [13] and PID [93] stabilizing controller schemes. Sometimes more computationally intensive and plant dependent *ideal* or *adaptive* backstepping controllers are implemented, presented in [65, 112] and [67] respectively. The gyroscopic response of a birotor vehicle's attitude system is more pronounced than that of a quadrotor, derived in Sec:3.2, and so feedback linearisation is almost always used. In an interesting progression from the norm, [72] proposed a unique PID co-efficient selection algorithm for a bi-rotor control block. Using a particle swarm optimization (*PSO*) technique, similar to [132], the coefficients were globally optimized around a given performance metric. However their performance criterion is a standard integral time-weighted absolute error (*ITAE*) term and nothing more appropriate involving effects unique to flight systems was used. *PSO* algorithms iteratively search for a globally optimized solution and offer independent, gradient free based optimization. In subsequent chapters, controller coefficients are optimized using *PSO* algorithms, shown later in Sec:5.1.

## Quadruped

Expanding on bi-rotor vehicles, the quadrotor UAV is a popular and well researched multirotor platform due to its mechanical simplicity. The current popularity of quadrotors as research platforms started in 2002, with a control algorithm implemented on what is now known as the X4-Flyer quadrotor [47, 100]. Alternative iterations then followed; like the Microraptor [105] and STARMAC [55] quadcopters which have subsequently been built and tested. A multitude of literature exists around quadrotor kinematics and their control [5, 18, 29, 78, 104], however dedicated rigid body 6-DOF dynamic papers [80, 96] offer 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 (App:A.1). Lately research projects have begun to incorporate non-linear aerodynamic effects like drag and propeller blade-element momentum (*BEM*) theory into the plant model [22, 55, 107]. The higher fidelity models for thrust and propeller responses offer more precision by making less linearisations and assumptions; [7, 55].



(a) Inspire1 articulated upwards

(b) Inspire1 articulated downwards

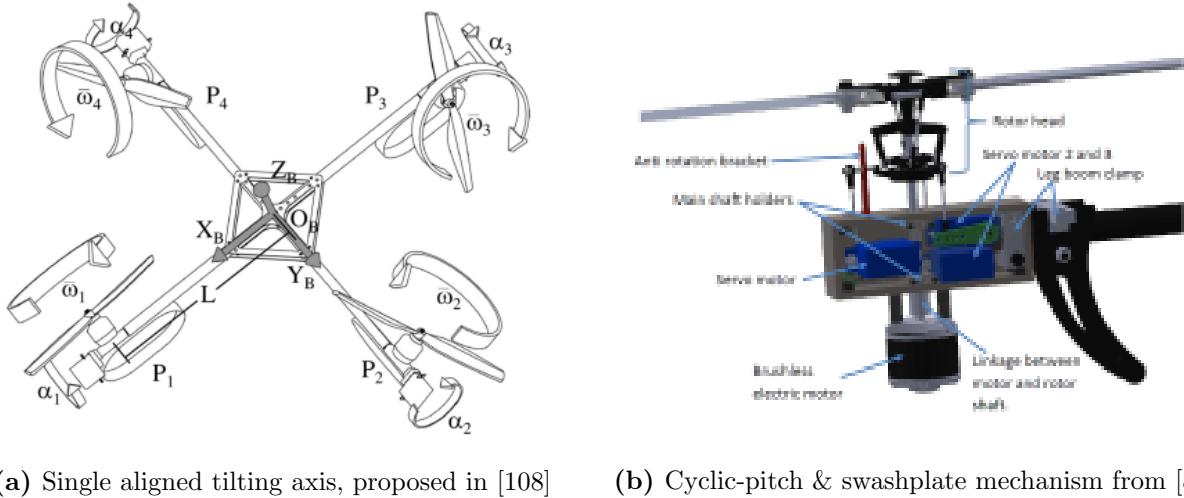
**Figure 1.4:** DJI Inspire1, the notations are with regards to the DJI patent [130]

At the time of writing, the only commercial UAV multirotor capable of structural transformation is the DJI Inspire1 quadrotor [33], manufactured by Shenzhen DJI Technologies. DJI are better known for their hugely successful DJI Phantom commercial quadrotor [34]. The Inspire1 can articulate its supporting arms up and down as shown in Fig:1.4, both images were sourced from the drone's patent; held by SZ DJI Tech Co [130]. The purpose of such transformations is to both alter the center of gravity and to further expose a belly mounted camera gimbal for panoramic viewing angles. This changes the bodies inertial tensor about its center of gravity, affecting the second order inertial response opposed to changes in angular velocity;  $\vec{\tau} = J\Delta\vec{\omega}_b$ . That variable inertial matrix is a detrimental effect which makes researchers apprehensive of transformable aerospace frames. The range of transformations which the Inspire1 frame can undergo is limited to just articulating its 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 propeller module tilting articulations. The extra actuation scheme(s) were first conceptualized and implemented on a prototype related to an ongoing project covered in two reports; [108, 109]. Those authors modified and tested a QuadroXL four rotor helicopter, produced by MikroKopter [41], to actuate a single axis of tilting aligned with the frame's arms (Fig:1.5a). Their proposed control solution, detailed next in Sec:1.2.2, assumes no nominal linearised conditions around hover flight, unlike a similar single axis tilting quadrotor prototype designed by Nemati in [86]. The latter is *simulated* but remains as yet untested.

One approach to improving quadrotor flight response is to alter the manner in which the thrust is mechanically actuated, potentially improving actuator bandwidth (demonstrated in [2, 38]). Drawing from helicopter design, [85] purported a novel quadrotor UAV prototype that used swashplates for varying the propeller pitch and generating torque moments. The aim was a design which was not dependent on rotational speed controlling power electronics (*ESCs*) 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, [85] made use of existing off-the-shelf hobbyist helicopter components to design a rotor actuation bracket (Fig:1.5b). The cyclic-pitch swashplates used could apply pitching and rolling torques [87],  $\tau_\phi$  and  $\tau_\theta$ , about each propeller's hub, its *principle axis of rotation*. The torques were induced by altering the blades angle of attack throughout the propeller's rotational cycle. The actuation rate of such a configuration is far greater than that of a differential torque produced rolling/pitching motion.



**Figure 1.5:** Tilt-rotor mechanisms

Irrespective of the strong initial design in the early stages of his project, it would appear that the research in [85] 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 severely lacking. A brief introductory proposal of an MPC attitude control system detracted from the comprehensive dynamics discussed. The project ended before testing, simulation or results could be obtained. Unfortunately, despite the novel over-actuated design, there was no discussion given on how that actuator allocation, being the most unique aspect of the project, would be achieved.



**Figure 1.6:** Dual-axis tilt-rotor mechanism used in [38]

Finally, the most crucial research to mention is a project completed by Pau Segui Gasco in [38], which was a dual presented MSc project with Yazan Al-Rihani whose respective research was presented in [2]. At the time of writing, this would appear to be the only project published pertaining to *over-actuation* in aerospace bodies implemented and tested on a quadrotor platform. The research was split between the two authors who completed the electronic/control design and the mechanical design for their respective MSc dissertations. Shown in Fig:1.6, the dual-axis articulation is achieved using an RC helicopter tail bracket and servo push-rod mechanism; reducing the mass of the articulated components but limiting the range of its possible actuation. Considering the propellers as energy

storing flywheels, the induced gyroscopic response was then treated as an additional controllable actuator plant. Their commanded virtual control is distributed by weighted inversion amongst the actuator set, Sec:1.2.2. The whole project justifies the extra actuation as fault tolerance redundancy (*FTC*) but does not necessarily prove how such a redundancy could be beneficial.

### 1.2.2 Notable Quadrotor Control Implementations

#### Quadcopter Attitude Control

Note that here  $\vec{\eta}$  is not necessarily an Euler angle set but any attitude representative state variable.

Attitude control of a 6-DOF aerospace body, quadrotor or otherwise, is best described by [124] and referred to as *the attitude control problem*. For a rigid body that has an instantaneous Euler attitude state  $\vec{\eta}_b$  and a desired state  $\vec{\eta}_d$ , the problem is to then find a stabilizing torque control  $\vec{\tau}_\mu$ . The control law is dependent on some feedback error state  $\vec{\eta}_e$ . Quaternion attitude states later replace Euler angles for attitude representation,  $\vec{\eta}_b \Rightarrow Q_b$ . The general control law is defined as:

$$\vec{\tau}_\mu = h(\vec{\eta}_d, \dot{\vec{\eta}}_d, \vec{\eta}_b, \dot{\vec{\eta}}_b, t) \in \mathcal{F}^b \quad (1.2a)$$

$$\rightarrow \vec{\tau}_\mu = h(\vec{\eta}_e, \dot{\vec{\eta}}_e, t) \text{ given some error state } \vec{\eta}_e \quad (1.2b)$$

Where the control law designs a net torque such that both the angular position and rates are stabilized with the bounded limits;  $\lim \vec{\eta}_b \rightarrow \vec{\eta}_d$  and  $\lim \dot{\vec{\eta}}_b \rightarrow \dot{\vec{\eta}}_d$  respectively as  $t \rightarrow \infty$ . A distinction must be made between euler angular rate vector,  $\dot{\vec{\eta}}_b = [\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T$  and the angular velocity vector  $\vec{\omega}_b = [p \ q \ r]^T$ . Depending on how the attitude is posed; with rotation matrices [69, 80, 96], quaternions [37, 43, 46, 69] or otherwise (direct cosine matrix etc ...) the error state  $\vec{\eta}_e = \vec{\eta}_d - \vec{\eta}_b$  could then differ to a (Hamilton) multiplicative relationship. [124] describes these conventionally different error states.

Simulation and modelling papers often rely on Euler angle based rotation matrices for attitude representation, [16, 18, 79, 86, 106], without addressing the inherent singularity associated with such an attitude representation (known as gimbal lock, [115], Sec:3.1.2). The alternative quaternion attitude representation, first implemented in 2006 on a quadrotor UAV platform in [121], is often used in lieu of rotation matrices. Quaternions do have their own caveat of *unwinding* as a result of the dual-coverage in  $\mathbb{R}^3$  space, discussed in [82] and derived mathematically later in Sec:3.1.4. Quaternions are  $\in \mathbb{R}^4$  variables for attitude representations in  $\mathbb{R}^3$  and so a mapping  $\mathbb{R}^4 \rightarrow \mathbb{R}^3$  produces an infinite coverage set for each unique attitude state in  $\mathbb{R}^3$ .

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 cross-product gyroscopic and Coriolis terms are both neglected when the angular velocity is small,  $\vec{\omega}_b \approx \vec{0}$ , and the inertial matrix  $J_b$  is approximately diagonal,  $\text{rank}(J_b) = x$  for  $x \in \mathbb{R}^x$ . The consequence of such simplifications is the deterioration of both the gyroscopic torque term,  $\vec{\tau}_{gyro} = -\vec{\omega}_b \times J_b \vec{\omega}_b \approx \vec{0}$  and the Coriolis force term,  $\vec{F}_{cor} = -\vec{\omega}_b \times m\vec{v}_b \approx \vec{0}$  in the body's dynamics (Ch:3 for context). Once the coupled cross-product terms are no longer of consequence, the 6 DOF trajectory,  $\vec{x} = [x \ y \ z \ \phi \ \theta \ \psi]^T$ , can be treated as a series of independent 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 Sec:3.1.3). So a quaternion combined four variable attitude state-space vector is then used,  $Q_b = [q_0 \ \vec{q}]^T$ , for the major loop trajectory plant of  $\vec{x}(t)$ .

Opensource and hobbyist flight controller's software (Arducopter [4], Openpilot [73] whose firmware stack is now maintained by LibrePilot, CleanFlight [27], BetaFlight [11], 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* [74] summarizes the control structures implemented on a range of popular flight controllers.



**Figure 1.7:** ArduCopter PI control structure for pitch angle  $\theta$ ; from [74]

The most popular of which, ArduCopter, implements a feed-forward PI compensation controller, whose single channel control loop for an attitude roll channel  $\theta$  is shown in Fig:1.7. PI, PD and PID controllers are all popular and effective plant independent control solutions for general attitude plants. Table:1.1 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. One ideal backstepping controller listed in Table:1.1, presented in [112], applies an algorithm derived through Hurwitz polynomials unlike the Lyapunov based backstepping control law(s) derived later in Ch:4.

Controller Type	Independent	Dependent	Total Examples
PI	[124]	[124]	2
PD	[2, 78]	[37, 86]	4
PID	[16, 19, 104, 108, 124]	[55, 106, 124]	8
Lead	[29, 100]	N/A	2
LQR	[19]	N/A	1
Backstepping controllers			
Ideal	[79, 112]	[79]	3
Adaptive	[10, 32, 67, 84]		4

**Table 1.1:** A breakdown of common attitude controllers

In a collection of papers, written by the most prolific early quadrotor author(s) S. Boubdallah and R. Siegward [17–19]; a range of different attitude control implementations are surveyed and tested on the OS4 platform. The final paper, [18], derived and practically tested an integral backstepping attitude controller on the OS4 quadrotor platform. It builds on their research presented earlier in [19] which provides an analysis of PID vs linear quadratic regulator (*LQR*) attitude controllers, specifically in the context of underactuated quadrotor attitude control. LQR controllers aim to optimize the controller effort with actuator inputs  $u \in \mathbb{U}$ ; controller effort is then  $\|u\|_2$  or the  $L_2$  norm (magnitude) of the plant input. Although, in theory, solving the associated Riccati cost function may produce a cost optimal, stable and efficient control law it needs exact plant matching. In reality, exact plant matching is difficult to achieve for a quadcopter or any aerospace body for that matter. The resultant controller in [19] achieved asymptotic stability but had poor steady state performance due to low accuracy of the identified actuator dynamics and poor confidence inertial measurements.

Adaptive Backstepping Control (in [128] or any other example in Table:1.1) expands on nominal ideal backstepping fundamentals by introducing disturbance and plant uncertainty terms into the Lyapunov energy function to be used for the backstepping suppression. For Lyapunov iteration the adaptive backstepping process requires a disturbance estimate derivative or *update law* which is often difficult to quantify. Approximation of plant disturbances without *a priori* information is a complex subject. At some point in the design an approximation heuristic must be adopted and that typically involves some compromise of performance or accuracy. One example of disturbance approximation in [32] proposes using a statistical projection operator (or *proj(.)*, [24]). When used in adaptive control, presented similarly in [25], the projection operator *proj(.)* ensures a derivative based estimator can be bound for adaptive regression approximation [99].

Although the control implementation is not explicitly backstepping, in [133] a sliding mode controller was used to compensate for the disturbances in an Unmanned Submersible Vehicle attitude plant. The underwater current disturbances were approximated using a fuzzy logic system, specifically a *zero-order Takagi-Sugeno-kang* fuzzy approximator. The TSK system has been shown in [81] to mimic an artificial neural network approximator; where the fuzzy TSK system is more comprehensible than the latter. Statistical analysis and investigation of approximators without *a priori* knowledge of a system are well beyond the scope of this research but are worth mentioning.

### Single/Dual Axis Control & Allocation

The steady state attitude control actuation introduced with either single or dual axis articulation provides room for more control goals to be achieved. Of the few papers published on tilting-axis quadrotors, PD controllers (used in [86] and again in [2,38]) and PID controllers (collectively [108,109]) are the standard fare for attitude control blocks. For either of these systems, there needs to be an allocation rule to distribute a commanded input amongst the actuator set. In the control allocation survey [60] the author describes the control allocation problem for a dynamic plant:

$$\dot{\vec{x}} = f(\vec{x}, t) + g(\vec{x}, \vec{\nu}, t) \quad \vec{x} \in \mathbb{R}^n, \vec{\nu} \in \mathbb{R}^m \quad (1.3a)$$

$$\vec{y} = c(\vec{x}, t) \quad (1.3b)$$

*State variables of [60] were changed to match this dissertation's conventions. In the state space equation Eq:1.3a, it is assumed the plant input,  $\vec{\nu}$ , has a linear multiplicative relationship with the input response,  $g(\vec{x}, t, \vec{\nu}) \Rightarrow g(\vec{x}, t)\vec{\nu}$ . That linear relationship is a prerequisite for most allocation inversion rules but is not a necessity...*

In Eq:1.3a the state  $\vec{x} \in \mathbb{R}^n$  has associated plant dynamics  $f(\vec{x}, t)$  and a input response  $g(\vec{x}, \vec{\nu}, t)$ . Setpoint tracking control equates the output variable with the state, in practice only state estimate is available:

$$\vec{y} = c(\vec{x}, t) = h(\vec{x}) = \hat{\vec{x}} \quad (1.4)$$

Therefore the output  $\vec{y}$  has the same dimension has the state  $\vec{x}$ ; or rather  $\vec{x}, \vec{y} \in \mathbb{R}^n$ . In an ideal, well posed system the number of actuator inputs equals the number of outputs; that being  $\dim(\vec{x}) = \dim(\vec{\nu}) \in \mathbb{R}^n$ . In the case where the control input  $\vec{\nu} \in \mathbb{R}^m$ , if  $m > n$  the problem is then over-actuated and an abstraction is needed. The system mechanically commands a physical control input  $\vec{\nu}_c$ , dependent on explicit actuator positions  $u \in \mathbb{U} \in \mathbb{R}^m$  as per some *effectiveness* function derived from the actuator plant's dynamics:

$$\vec{\nu}_c = B(\vec{x}, u, t) \quad \in \mathbb{R}^n \quad (1.5)$$

Assuming that some higher level control law designs well a satisfactory stabilizing virtual control input from the error state(s)  $\vec{\nu}_d = h(\vec{x}_d, \dot{\vec{x}}_d, \vec{x}_b, \dot{\vec{x}}_b, t) \in \mathbb{R}^n$ . The allocation rule then aims to solve for an explicit actuator position  $u \in \mathbb{U} \in \mathbb{R}^m$  derived from  $\vec{\nu}_d$  which actuates the physically commanded control input  $\vec{\nu}_c$ , minimizing the deviation or slack  $\vec{s}$  between virtual desired and physical commanded inputs  $\vec{\nu}_d$  and  $\vec{\nu}_c$  respectively. Allocation is effectively a paradigm which transforms dimensions  $\mathbb{R}^m \rightarrow \mathbb{R}^n$  using a commanded actuator matrix position  $u \in \mathbb{R}^m$ . An over-actuated plant can be summarized in non-linear state space as:

$$\dot{\vec{x}} = f(\vec{x}, t) + g(\vec{x}, \vec{\nu}_c, t) \quad \vec{x} \in \mathbb{R}^n \quad (1.6a)$$

$$\vec{\nu}_c = B(\vec{x}, u, t) \quad \vec{\nu}_c \in \mathbb{R}^n \quad (1.6b)$$

$$\text{with } u \in \mathbb{U}^m \text{ subject to some } \min(\vec{s}) \text{ such that } \vec{s} = \vec{\nu}_d - \vec{\nu}_c \quad (1.6c)$$

$$\vec{\nu}_d = h(\vec{x}_d, \dot{\vec{x}}_d, \vec{x}_b, \dot{\vec{x}}_b, t) \quad \vec{\nu}_d \in \mathbb{R}^n \quad (1.6d)$$

$$\vec{y} = c(\vec{x}, t) = \hat{\vec{x}} \quad (1.6e)$$

The effectiveness function  $B(\vec{x}, u, t)$  quantifies how actuator inputs  $u \in \mathbb{U}$  correlate to the physically commanded plant input  $\vec{\nu}_c$ . Inversion based allocation rules which solve for explicit actuator solutions (Sec:4.7) require that  $B(\vec{x}, u, t)$  can be abstracted to a linear multiplicative relationship  $B(\vec{x}, t)u$  with  $B(\vec{x}, t) \in \mathbb{R}^{n \times m}$ , such that a generalized inverse of  $B(\vec{x}, t)$  can be found. For generic set point tracking the control law will design a desired virtual control input  $\vec{\nu}_d$ , the allocation rule then has to solve  $u$  for  $\vec{\nu}_c$  such that for some slack variable  $s = \vec{\nu}_c - \vec{\nu}_d$  is minimized:

$$\min_{u \in \mathbb{R}^m, s \in \mathbb{R}^n} \|Q_s\| \text{ subject to } B(\vec{x}, u, t) - h(\vec{x}_e, t) = \vec{\nu}_c - \vec{\nu}_d = s \quad u \in \mathbb{U} \quad (1.7)$$

Which ensures the commanded input  $\vec{\nu}_c$  tracks the desired control input  $\vec{\nu}_d$ ;  $\vec{\nu}_c \rightarrow \vec{\nu}_d$  as per some cost function of the slack variable  $Q_s$ . Mostly the L<sub>2</sub> norm,  $Q_s = \|s\|_2$ , is used. In an over-actuated system it then follows that there is a whole set of possible inputs for each  $\vec{\nu}_c$ . A unique actuator solution (rather than a family of solutions) to Eq:1.7 needs a secondary objective function,  $j(\vec{x}, u, t)$ . Eq:1.7 then becomes;

$$\min_{u \in \mathbb{R}^m, s \in \mathbb{R}^n} (\|Q_s\| + j(\vec{x}, u, t)) \text{ subject to } \vec{\nu}_c - \vec{\nu}_d = s \quad u \in \mathbb{U} \quad (1.8)$$

Those same authors, Johansen and Tjønnås in [60–62], proposed multiple control allocation solutions to a variety of systems. Following [60]; in a subsequent paper [61], the authors introduced a secondary cost function, driving the solution away from the typical linear quadratic programming pseudo and weighted inversion solutions. Aiming for actuator efficiency and not just input saturation, a subsequent paper [62] proposed adaptively allocating actuator positions online. Using a Lyapunov energy equation as the online cost function, the minimization adaptive law was ensured to always settle on a feasible solution.

Over-actuation is not often applied to quadrotors and rather than providing a comprehensive literature review of associated papers here (which are all mostly theoretical derivation), the contextual application and solutions are expanded upon later in Sec:4.7. The only overactuated quadrotor literature which covers allocation of the extra actuators is [2, 38], where the authors apply a weighted pseudo inverse (otherwise known as the Moore-Penrose Inverse [71]) allocation rule. Birotor dual-axis tilting, detailed earlier, results in a critically actuated system and so requires no allocation. As mentioned before, a prerequisite for (*pseudo*) inversion is a multiplicative *linear* control effectiveness relationship for Eq:1.6b.

The only overactuated quadcopter paper which addressed its required control allocation was that of Gasco and Rihani in [2, 38]. Their solution applied weighted inversion, relying on some very specific assumptions to achieve that required input linearity for the system in Eq:1.6b. For the gyroscopic torque response to extra actuator  $\eta$  or  $\gamma$  pitching and rolling movement:

$$\tau = J\Omega\dot{\eta} \in \mathcal{F}^b \quad (1.9)$$

With  $\Omega$  being that propellers rotational speed and  $\dot{\eta}$  being the inducing servos rate. The authors assumed the extra actuators pitch and roll angular rates;  $\dot{\eta}$  and  $\dot{\gamma}$  respectively, were both proportionally related to their positions  $\eta$  and  $\gamma$  as follows:

$$\dot{\eta} \approx \frac{1}{t_{settle}}\eta \text{ and } \dot{\gamma} \approx \frac{1}{t_{settle}}\gamma \quad (1.10)$$

Where  $t_{settle}$  is a constant derived the actuator transfer function's settling time from a unit input step. Such an assumption holds true so long as  $\Delta\eta$  or  $\Delta\gamma$  is smaller than the initial step used to evaluate  $t_{settle}$ , a restrictive and unrealistic assumption but implemented nonetheless. It then follows that the gyroscopic first order torque  $\tau = -\vec{\omega}_b \times J_b \vec{\omega}_b$  and second order inertial torque  $\tau = J_b \dot{\vec{\omega}}_b$  responses are both functions of their associated servo positions  $\eta$  and  $\gamma$ , not respective their derivatives. The extent of that consequence is contrasted with the allocation solution in Sec:4.7.

## Satellite Attitude Control

Unconstrained attitude set-point tracking for 6-DOF bodies, quaternion based or otherwise, is a topic well covered in the field of satellite attitude control; [59, 68, 126]. The *status quo* for recent research is on non-linear adaptive attitude backstepping control systems, wherein the adaptive update rule is the novel contribution. Plant uncertainty always adversely affects the confidence in inertial tensor measurements critical to the attitude control of a satellite. In [59] the authors proposed applying adaptive backstepping to compensate for steady state plant uncertainty errors of the (asymmetric) inertial estimations.

Alternatively, instead of deliberating on costly non-orbital prelaunch inertial measurements, [14] developed an algorithm for estimating the inertial tensor using 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 tolerance, and thus require control allocation. Often the extra allocators are control moment gyroscopic actuators, flywheels driven by DC motors, to produce rotational torques. Thrusters have a limited amount of fuel and can actuate the system a finite number of times. The thrusters can then be scheduled with a lower priority, biasing electronic CMG actuators. In [68] the authors address the over-actuation with direct pseudo inversion before applying quaternion based backstepping for attitude control. Such an inversion solves for Eq:1.8:

$$u = B^\dagger \vec{\nu}_d \quad (1.11a)$$

$$B^\dagger = B^T (BB^T)^{-1} \quad (1.11b)$$

$$u \in \mathbb{R}^m, \vec{\nu}_d \in \mathbb{R}^n, B \in \mathbb{R}^{m \times n}, B^\dagger \in \mathbb{R}^{n \times m} \quad (1.11c)$$

Where  $B$  is the effectiveness matrix, a static effector form of the effectiveness function  $B(\vec{x}, u, t)$ . The generalized inverse  $B^\dagger$  is such that  $BB^\dagger = \mathbb{I}$ . Specifically  $B^\dagger$  is the general *pseudo* inverse of  $B$  (more on inversions in Sec:4.7). Moreover there is an assumed linear multiplicative relationship between the input,  $u \in \mathbb{U}$ , and the input effectiveness matrix in Eq:1.6b. The controller designed actuator torque  $\vec{\nu}_d$  then dictates the input  $u$  as per Eq:1.11a. Much like the over-actuation previously discussed with respect to quadcopters; the pseudo inversion method of actuator distribution applies linear quadratic optimization to the allocation slack cost function, Eq:1.7.

The resultant quaternion attitude backstepping controller developed in [68] demonstrated global uniform asymptotic stability. The strength of that backstepping stability lies in the choice of trajectory aiming to be stabilized;  $z \rightarrow \vec{0}$ . The first Lyapunov trajectory was defined as:

$$z_1 = \begin{bmatrix} 1 - |q_0| \\ \vec{q}_e \end{bmatrix} \quad (1.12a)$$

Such that the Lyapunov energy function candidate is always positive definite and its derivative is positive definite decrescent. The particulars of that stability proof are omitted but it is worth detailing their chosen candidate function;

$$V_1(z) = z_1^T z_1 > 0 \quad \forall [q_0, \vec{q}_e] \quad (1.12b)$$

The absolute quaternion error scalar in Eq:1.12a ensures a global trajectory's asymptotic stability (Sec:4.5.3), not just local stability that would otherwise be gained. The stable equilibrium points at  $Q_e = [\pm 1 \quad \vec{0}]^T$  apply settling of the trajectory's *error*, allowing the satellite to track its setpoint. However considering that the controller is an ideally compensating controller, the disturbance rejection and uncertainty compensation of the attitude controller could potentially disrupt that achieved stability. Something which was not discussed in the original paper.

# 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, aimed at optimizing engineering time spent on construction and reducing the associated component costs. Significant consideration for the design process was the net weight whose upper limit is inherently limited by the thrust produced from lift motors. Some of the more important design factors, like inertial matrices and associated masses (Sec:2.3), are discussed here in order to give context for the dynamics derived later in Ch:3. The reference frame orientations (which those dynamics are developed with respect to) are detailed here. A brief overview of the electrical systems layout is then given with the components associated and their electrical characteristics included. Finally the actuator suite's functionality and transfer characteristics are quantified. A review of the physical prototype realized and control loop(s) implemented is detailed in Ch:6 along with actual flight test results.

### 2.1.1 Actuation Functionality

The most important component of the design is the manner of articulation for each concentric gimbal ring which forms the four motor module structures. The control 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 net 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 later in Sec:2.3.



**Figure 2.2:** Tilting rotor design

Each motor module is positioned such that its produced thrust vector coincides with the intersection of its two rotational axes (Fig:2.2a). As a result there is only a perpendicular displacement,  $L_{arm} = 195.16$  mm, co-planar to the body frame's X-Y-Z origin  $\vec{O}_b$  (see subsequent Fig:2.8). That length directly affects the differential torque plant;  $\vec{\tau}_{diff} = \sum \vec{L}_i \times \vec{T}_i$ . An eccentric thrust vector line would make the torque 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 is more prudent 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 of modeling a propeller's aerodynamic thrust (Sec:3.3.1).

The primary body structure is similar to a traditional quadcopter '+' configuration with adjacent propellers spinning in opposite directions. Each motor module's rotational assembly is suspended by silicone damping balls (Fig:2.2b). A smaller damping assembly in the center of the frame houses all the electronics and power distribution circuitry. All the mounting brackets affixing the motor module rings are 3D printed from CAD models using an Ultimaker V2+ [127]. A complete bill of materials for all parts used, including working drawings for each 3D printed bracket and the laser cut frame(s), is presented in Appendix:B.

The propeller's 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 23.0 [mm] and must be considered when evaluating pitch/roll inertial and gyroscopic torque responses later in Sec:3.2.1. The propellers are 6 inch ( $6 \times 4.5$ ) 3-Blade plastic Gemfam propellers, powered by Cobra CM2208-2000KV Brushless DC motors (Fig:2.4a). The thrust produced as a function of angular velocity (in RPS) for the propellers is derived in Sec:3.3.1.



**Figure 2.3:** Difference between propeller and motor planes

The BLDC motors are controlled with LDPower 20A ESC modules with an in-line OrangeRx RPM Sensor. The ESCs were reflashed with BLHeli [12] firmware. The default firmware on the speed controllers had an unsatisfactory exponential approaching (not linear) input speed curve, in contrast with the linear (unloaded) speed curve in Fig:2.19. The net transfer functions for both ESC modules and the servos are detailed later in Sec:2.4.1. Power for the quadrotor is supplied from a power tether (not from a battery bank). 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 through conventional wiring, however an ideal and more flexible design would see slip-rings for each module's power supply.

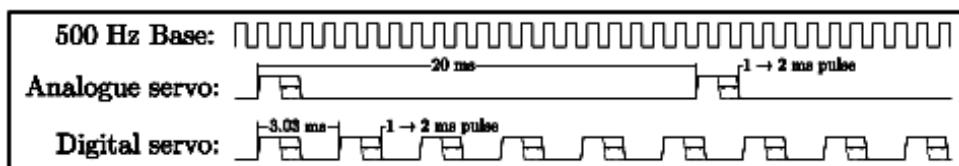


(a) Cobra CM2208-2000KV BLDC motor module

(b) Corona DS-339MG servo bracket

**Figure 2.4:** Motor module assembly

Metal gear Corona DS-339MG digital servos are used for the two axes of rotation (Fig:2.4b). Each servo has a rotational range of  $\approx \pi$ , positioned such that a zero<sup>th</sup> offset aligns the motor modules, adjacent to the body frame, and has a  $\pm\pi/2$  rotational range. A digital servo updates at 330 Hz, faster than a 50 Hz analogue servo equivalent (Fig:2.5). This means the otherwise 20 ms zero-order "analogue" sampling effect is a less significant 3.30 ms zero-order holding time. Both the  $\hat{X}_{Mi}$  and  $\hat{Y}_{Mi}$  axis servos will be rotating a larger inertial body, as such the open loop plant dynamics are determined empirically in Sec:2.4.1.

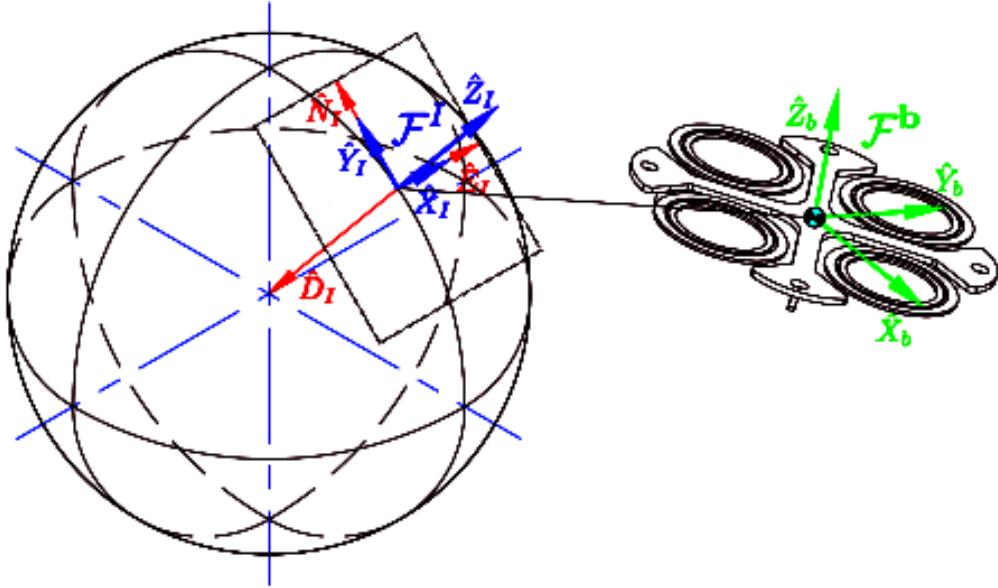


**Figure 2.5:** Digital and analogue servo timing

## 2.2 Reference Frames Used

Attitude conventions used for deriving the system's dynamics, in Ch:3, are first discussed here. Often these aspects are assumed to be obvious enough that they are omitted. It is 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 rotation and transformation operations. Both [46] and [96] provide an in-depth and thorough explanation of rotation matrices and direct cosine matrix attitude representation, if such concepts are unfamiliar to the reader. Quaternions are introduced to replace rotation matrix notation for the dynamics in Sec:3.1.3.

### 2.2.1 Reference Frames Convention



**Figure 2.6:** Inertial and body reference frames

NASA aerospace frames are used for principle Cartesian inertial and body coordinate representation (Fig:2.6). The inertial frame,  $\mathcal{F}^I$ , is aligned such that the  $\hat{Y}_I$  axis is in the  $\hat{N}$ orth direction,  $\hat{X}_I$  is in the  $\hat{E}$ ast direction and  $-\hat{Z}_I$  is in the  $\hat{D}$ ownward direction. In Euler orbital sequences the  $\hat{Z}$  direction would be toward the Earth's center, sometimes referred to as the NED convention which differs from the NASA frames used here. 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 upward direction (illustrated in Fig:2.9). The body frame's axes and center of motion relative to the prototype design's center of mass is related as detailed next in Sec: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.

$$\vec{v}_I = R_b^I(\eta)\vec{v}_b \quad \vec{v}_b \in \mathcal{F}^b, \vec{v}_I \in \mathcal{F}^I \quad (2.1)$$

Relative angular displacement between two frames is commonly measured by the three angle Euler set. The Euler angle set  $\vec{\eta} = [\phi \ \theta \ \psi]^T$  represents pitch  $\phi$ , roll  $\theta$  and yaw  $\psi$  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 general rotation equation to *rotate* some vector  $\vec{v}$  about a normalized unit axis  $\hat{u}$  through a rotation angle  $\theta$  is given by the formula, proven in [31]:

$$\vec{v}' = (1 - \cos(\theta))(\vec{v} \cdot \hat{u})\hat{u} + \cos(\theta)\vec{v} + \sin(\theta)(\hat{u} \times \vec{v}) \quad (2.2)$$

In Eq:2.2, when  $\hat{u}$  is a unit vector in the direction of either  $\hat{X}, \hat{Y}$  or  $\hat{Z}$  axes, the equation can be simplified to produce the three fundamental rotation matrices;  $R_x(\phi)$ ,  $R_y(\theta)$  and  $R_z(\psi)$ . That set of three principle rotation matrices about the Cartesian X-Y-Z axes are defined as:

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \quad (2.3a)$$

$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (2.3b)$$

$$R_z(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.3c)$$

The notation for a rotation matrix operation is multiplication of the matrix  $R_n(\theta)$ , applying a left-handed *rotation* operator about some axis  $\hat{n}$  by  $\theta$ . The resultant vector of a rotation operation still exists in the same reference frame. An  $\hat{X}$  axis rotation by  $\phi$  of some vector  $\vec{v}$  is given by;

$$\vec{v}' = R_x(\phi)\vec{v} \quad \vec{v}', \vec{v} \in \mathcal{F}^1 \quad (2.4a)$$

No subscripts are used in Eq: 2.4 to indicate reference frame ownership because all vectors are in the same frame. The time derivative of a rotation matrix about some axis  $\hat{n}$  by a rotation  $\theta$ ,  $R_n(\theta)$  is shown in [104] to be:

$$\frac{d}{dt}(R_n(\theta)) = (\dot{\theta} \cdot \hat{n}) \times R_n \Rightarrow [\dot{\theta} \cdot \hat{n}]_x R_n \quad (2.5a)$$

Where, for some vector  $\vec{a}$ , the operator  $[\vec{a}]_x$  denotes the cross-product matrix or *skew* matrix. The symmetric skew matrix is a matrix multiplication to replace the cross-product operator, for some other vector  $\vec{b}$ ;

$$\vec{a} \times \vec{b} = [\vec{a}]_x \vec{b} \quad (2.5b)$$

$$[\vec{a}]_x \triangleq \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \quad (2.5c)$$

A vector *transformation* changes the resultant vector's reference frame. The 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  $\phi$  about the  $\hat{X}$  axis is then a negative rotation operation:

$$\vec{v}_2 = R_x(-\phi)\vec{v}_1 \quad (2.6a)$$

$$\vec{v}_2 \in \mathcal{F}^2 \text{ and } \vec{v}_1 \in \mathcal{F}^1 \quad (2.6b)$$

The distinction between Eq:2.4 and Eq:2.6 is the directional sense of the angular operand  $\phi$ , and hence the effect it has on the argument vector. The transformation or rotation of a vector from the inertial frame  $\mathcal{F}^I$  to the body frame  $\mathcal{F}^b$  is the product of three sequential operations about each principle axis. Each subsequent rotation is applied relative to a new intermediate frame; hence each Euler angle is taken relative to a specific intermediate frame and not a global one. The order of those axial rotation operations does affect the Euler set, any consequences of which are detailed in [69]. In this dissertation the Z-Y-X or yaw, pitch, roll rotation sequence is used. A rotation of the vector  $\vec{v}$  from the inertial to the body frame,  $\mathcal{F}^I \rightarrow \mathcal{F}^b$ , is then applied by sequential yaw,  $\psi$ , pitch,  $\theta$ , and roll  $\phi$  operations about the  $\hat{Z}$ ,  $\hat{Y}$  and  $\hat{X}$  axes respectively:

$$R_I^b(\eta) = R_I^b(\phi, \theta, \psi) \triangleq R_z(\psi)R_y(\theta)R_x(\phi) \quad (2.7a)$$

$$\vec{v}' = R_I^b(\phi, \theta, \psi)\vec{v} \quad (2.7b)$$

$$\rightarrow \vec{v}' = R_z(\psi)R_y(\theta)R_x(\phi)\vec{v} \quad (2.7c)$$

It is important to note that in Eq:2.7 both operand and output vectors are *both* in the inertial frame,  $\vec{v}' , \vec{v} \in \mathcal{F}^I$ . A *transformation* of a vector from the inertial to the body frame is the negative counterpart of Eq:2.7, a distinction that is not always explicitly stated.

$$\vec{v}_b = R_I^b(-\eta)\vec{v}_I = R_I^b(-\phi, -\theta, -\psi)\vec{v}_I \quad (2.8a)$$

$$\text{for } \vec{v}_b \in \mathcal{F}^b \text{ and } \vec{v}_I \in \mathcal{F}^I \quad (2.8b)$$

$$\rightarrow \vec{v}_b = R_z(-\psi)R_y(-\theta)R_x(-\phi)\vec{v}_I \quad (2.8c)$$

$$= R_x(\phi)R_y(\theta)R_z(\psi)\vec{v}_I = R_b^I\vec{v}_I \quad (2.8d)$$

$$R_I^b = (R_b^I)^{-1} = (R_b^I)^T \quad (2.8e)$$

The relationship in Eq:2.8e is an inversion property (*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 whilst the angular sign differentiates the rotation or transformation operations.

The body frame's angular velocity is taken relative to the inertial frame, represented by  $\vec{\omega}_{b/I}$  mostly just simplified to  $\vec{\omega}_b$ . Seeing that each Euler angle is measured with respect to an intermediary frame, a distinction must then be made between  $d\vec{\eta}/dt$  and  $\vec{\omega}_b$ . All three Euler angles need to be transformed to a shared frame to define the relationship between Euler and angular rates. Exploiting vehicle frames 1 & 2, or rather  $\mathcal{F}^{v1}$  &  $\mathcal{F}^{v2}$ , as intermediary frames to respectively describe frames after  $R_x(\phi)$  and  $R_y(\theta)$  operations and using the rotation matrix derivative from Eq:2.5. The angular velocity  $\vec{\omega}_b$  is the time derivative of Euler angles in the body frame:

$$\vec{\omega}_b = [p \quad q \quad r]^T = \frac{d}{dt}\vec{\eta} = \frac{d}{dt}\vec{\eta}_b \quad \in \mathcal{F}^b \quad (2.9a)$$

$$\vec{\eta}_b = R_{v_2}^b(\phi)\vec{\phi} + R_{v_2}^b(\phi)R_{v_1}^{v_2}(\theta)\vec{\theta} + R_{v_2}^b(\phi)R_{v_1}^{v_2}(\theta)R_I^{v_1}(\psi)\vec{\psi} \quad \in \mathcal{F}^b \quad (2.9b)$$

$$\therefore \vec{\omega}_b = \left[ \dot{\vec{\phi}} \right]_x R_{v_2}^b(\phi) + R_{v_2}^b(\phi) \left[ \dot{\vec{\theta}} \right]_x R_{v_1}^{v_2}(\theta) + R_{v_2}^b(\phi)R_{v_1}^{v_2}(\theta) \left[ \dot{\vec{\psi}} \right]_x R_I^{v_1}(\psi) \quad \in \mathcal{F}^b \quad (2.9c)$$

With Euler vectors  $\vec{\phi}$ ,  $\vec{\theta}$  and  $\vec{\psi}$  being axis projections onto X-Y-Z axes respectively;  $\phi \cdot \hat{i}$ ,  $\theta \cdot \hat{j}$  and  $\psi \cdot \hat{k}$ . The vehicle frames used for Eq:2.9a 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, here Z-Y-Z rotation sequences were used. The Euler rates equation then simplifies to the formal relationship between two rotating frames, with  $\vec{\omega}_b = [p \ q \ r]^T$  in  $\text{rad.s}^{-1}$ :

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \sin(\phi)\cos(\theta) \\ 0 & -\sin(\theta) & \cos(\phi)\sin(\theta) \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (2.9d)$$

$$\Rightarrow \vec{\omega}_b = \Psi(\eta)\dot{\vec{\eta}} \quad \in \mathcal{F}^b \quad (2.9e)$$

$$\Psi(\eta) = \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \sin(\phi)\cos(\theta) \\ 0 & -\sin(\theta) & \cos(\phi)\sin(\theta) \end{bmatrix} \quad (2.9f)$$

$$\Rightarrow \dot{\vec{\eta}} = \Psi^{-1}(\eta)\vec{\omega}_b = \Phi(\eta)\vec{\omega}_b \quad \in \mathcal{F}^{v1,v2,I} \quad (2.9g)$$

$$\Phi(\eta) = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)\sec(\theta) & \cos(\phi)\sec(\theta) \end{bmatrix} \quad (2.9h)$$

The *Euler* matrix,  $\Psi(\eta)$ , contains a well known and problematic singularity; at  $\theta = \pm\pi/2$  the determinate of the transformation matrix is zero. The mathematical manifestation of that singularity and its physical consequences are expanded on in Sec:3.1.2. The singularity is present in the middle roll angle  $\theta$ , which is a direct consequence of the chosen Z-Y-X rotation sequence adopted. Each Euler angle is potentially singular depending on the rotation order used. In later dynamics quaternions are used in lieu of Euler angles (Sec:3.1.3). Attitude in  $\mathbb{R}^3$ , or  $SO(3)$ , is intuitive and well suited to the conventions defined here.

Quaternions, despite being in  $\mathbb{R}^4$ , are similarly constructed in the Z-Y-X order following a three rotation sequence. Combined quaternion operations are additive but non-commutative, as such the order is important. The constructed attitude quaternion order will produce the same resultant frame orientation however the quaternion, and its rotation path, will differ. A quaternion  $Q_b$ , representing the body's attitude, a vector  $\vec{v}_I$  in the inertial frame is related to the body frame  $\mathcal{F}^b$  as follows:

$$\vec{v}_b = R_I^b(-\eta)\vec{v}_I \iff Q_b \otimes [0 \ \vec{v}_I]^T \otimes Q_b^* \quad (2.10a)$$

$$Q_b \triangleq Q_z Q_y Q_x \text{ and it's inverse } Q_b^* \triangleq Q_x^* Q_y^* Q_z^* \quad (2.10b)$$

The symbol  $\otimes$  represents the Hamilton product, or quaternion multiplication operator. Later the Hamilton product is used again for inertial tensor transformations in Sec:2.3. Each quaternion,  $Q_i$ , is always the *unit* quaternion about the  $i^{th}$  axis. For the body quaternion,  $Q_b$ , it is the unit quaternion rotation about the body's Euler axis [69]. 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:2.10a is equivalent to;

$$Q_{\vec{v}'} = Q \otimes (Q_{\vec{v}}) \otimes Q^* \quad (2.11a)$$

$$\text{Where } Q_{\vec{v}} = [0 \ \vec{v}]^T, \quad Q_{\vec{v}'} = [0 \ \vec{v}']^T \quad (2.11b)$$

Quaternion representation in Eq:2.11b ensures that the operation is entirely in  $\mathbb{R}^4$  space. However it is typically omitted, despite  $\mathbb{R}^4$  being implied and as such, Eq:2.11a is then simply:

$$\vec{v}' = Q \otimes (\vec{v}) \otimes Q^* \quad (2.12)$$

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

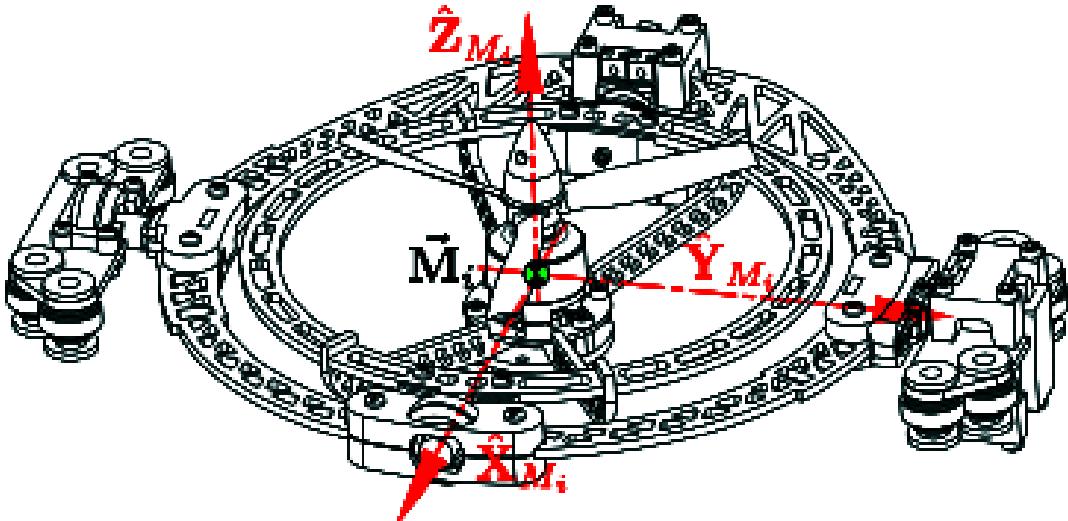
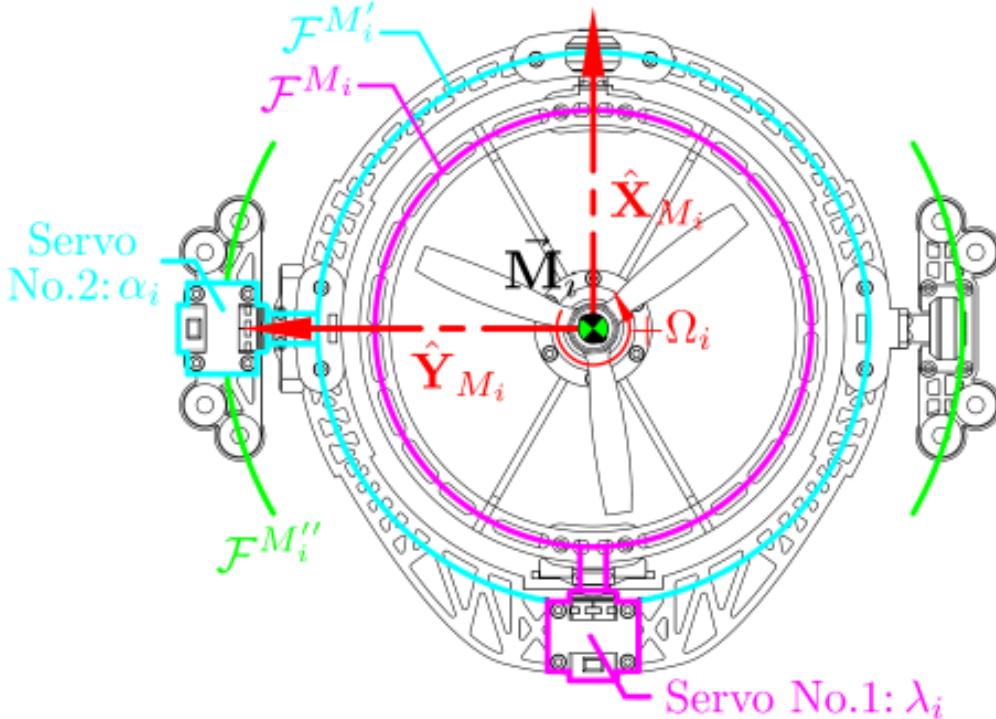


Figure 2.7: Aligned motor frame axes

The whole structure (Fig:2.1) consists of multiple rigidly connected bodies with only relative rotations between each body permitted by its joints, illustrated previously in the design description in Sec:2.1. Those rigid bodies are categorized into four inter-connected motor modules,  $\mathbf{M}_{1 \rightarrow 4}$ , and a single body structure,  $\mathbf{B}$  (*frame* structure, not reference frame). Each module contains two sequential gimbal rings, where each ring has one degree of relative rotation, actuated by a servo, between itself and the subsequent ring. There needs to be distinct nomenclature used for describing these motor modules such that the dynamic derivations later are clear and logical despite the complicated multibody system...



**Figure 2.8:** Intermediate motor frames

Every propeller/motor is actuated by a pair of two servos about two subsequent rotational axes (Fig:2.7) in a similar fashion to an Euler rotation sequence. The  $i^{th}$  propeller, attached to the BLDC motor's rotor, rotates in frame  $\mathcal{F}^{M_i}$  with a rotational speed  $\Omega_i$  in revolutions per second, or [RPS], about the  $\hat{Z}_{M_i}$  stator axis. The motor's stator is affixed to the inner ring assembly which rotates about its  $\hat{X}_{M_i}$  axis by  $\lambda_i$  from the module's first servo. The first servo is attached to the middle ring assembly with the frame  $\mathcal{F}^{M'_i}$ . The middle ring assembly, frame  $\mathcal{F}^{M'_i}$ , rotates about its  $\hat{Y}_{M'_i}$  axis actuated by the second  $\alpha_i$  servo. That second servo is affixed to an intermediate  $\mathcal{F}^{M''_i}$  frame, finally there's an 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 the rotational speed, both servo positions and all their respective rates;  $[\Omega_i \lambda_i \alpha_i \dot{\Omega}_i \dot{\lambda}_i \dot{\alpha}_i]^T$  for  $i \in [1 : 4]$ .

Fig:2.9 shows how the axes of each motor module align with the body frame's axes at rest. The body frame  $\mathcal{F}^b$  has the origin  $\vec{O}_b$  at the X-Y center of the structure, co-planar to each motor modules' center. *Neither* the body frame's origin *nor* each modules center of rotation is coincidental with either body's center of mass. The exact disparity between the origin(s) of motion and the respective body's center of mass are detailed subsequently in Sec:2.3.

The motor module pair 1 and 3 have their  $\hat{X}$ -axes in the positive and negative  $\hat{X}_b$  directions of the body frame respectively. Similarly Modules 2 and 4 have their  $\hat{X}$ -axes in the positive and negative  $\hat{Y}_b$  directions of the body frame. Motor modules 1 & 3 have clockwise rotating propellers; denoted by a positive super script or  $\Omega_{[1,3]}^+$ . Conversely modules 2 & 4 have counter-clockwise rotations; denoted by a negative super script or  $\Omega_{[2,4]}^-$ .

Not shown in Fig:2.9 is the relative  $\hat{Z}_b$  origin position of  $\vec{O}_b$  with respect to the entire assembly. The  $\Delta Z$  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 not coincidental with the center of mass.

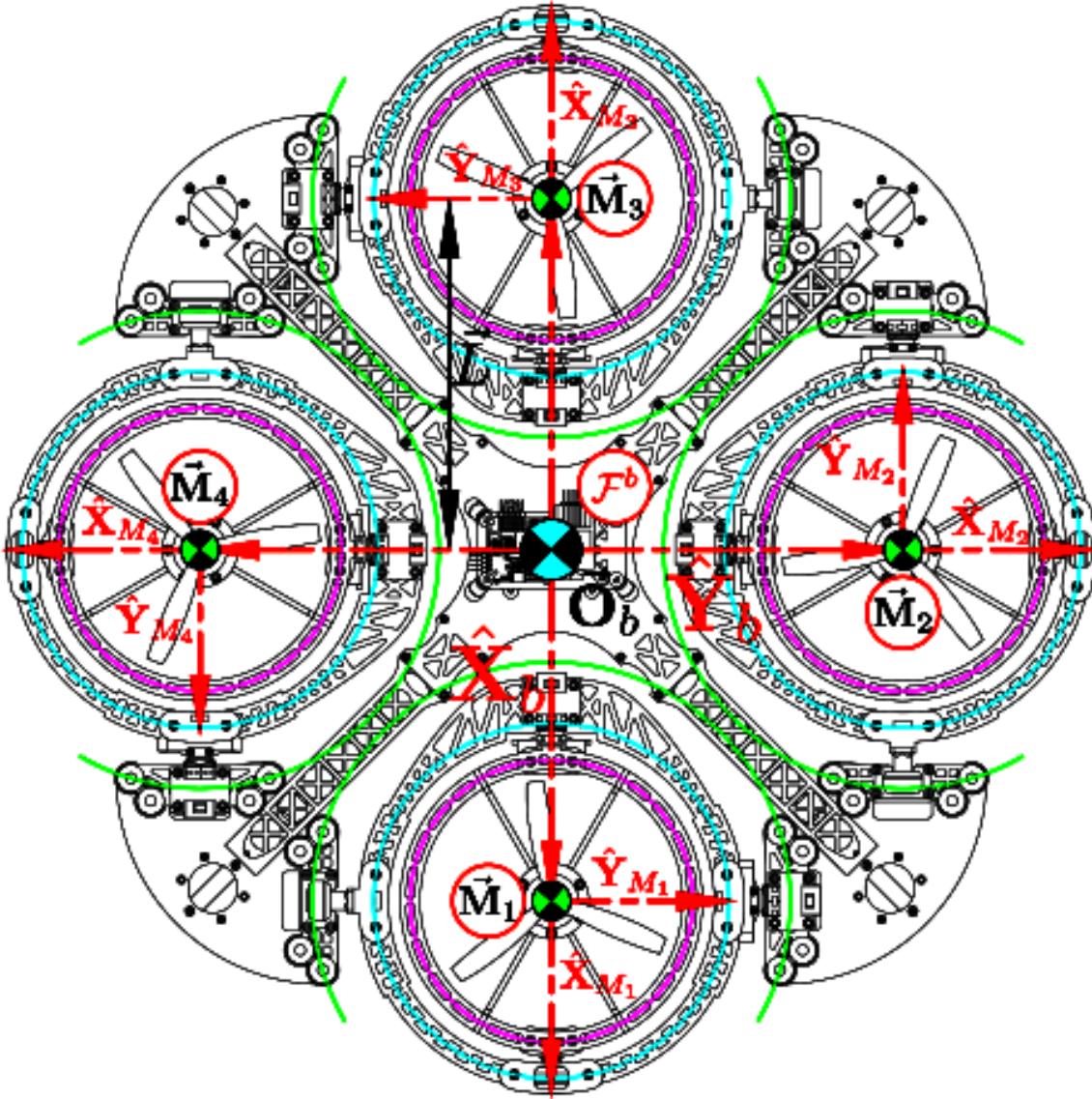


Figure 2.9: Body frame axes layout

Each motor is displaced from the body frame origin by  $L_{arm} = 195.16$  [mm]. Transforming some vector  $\vec{v}_{M_i}$  in the motor frame  $\mathcal{F}^{M_i}$  to the body frame is given as:

$$\vec{v}_b = R_{M_i}^b \vec{v}_{M_i} = R_z(-\sigma_i)R_y(-\alpha_i)R_x(-\lambda_i)\vec{v}_{M_i} \quad \text{for } \sigma_i \in [0 \quad \frac{\pi}{2} \quad \pi \quad \frac{2\pi}{3}] \quad (2.13a)$$

The orthogonal  $\sigma_i$  rotations about  $\hat{Z}_{M_i''}$  are independent of actuator positions,  $\sigma_i$  is a constant determined by the motor module's location. The rotation matrices  $R_z(\sigma_i)$  for  $\sigma_i = (i - 1)\pi/2$  are:

$$R_z = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{for } i \in [1, 2, 3, 4] \text{ respectively} \quad (2.13b)$$

The actuator space, including propeller speed  $\Omega_i$  [RPS], is then  $\in \mathbb{R}^{12}$ , or rather  $\mathbb{U} \in \mathbb{R}^{12}$ , in contrast with  $\mathbb{U} \in \mathbb{R}^4$  for a standard quadrotor. The actuator input set  $u \in \mathbb{U}$  is then structured as:

$$u_{\mathbb{U}} = [\Omega_1^+ \quad \lambda_1 \quad \alpha_1 \quad \dots \quad \Omega_4^- \quad \lambda_4 \quad \alpha_4]^T \quad (2.14)$$

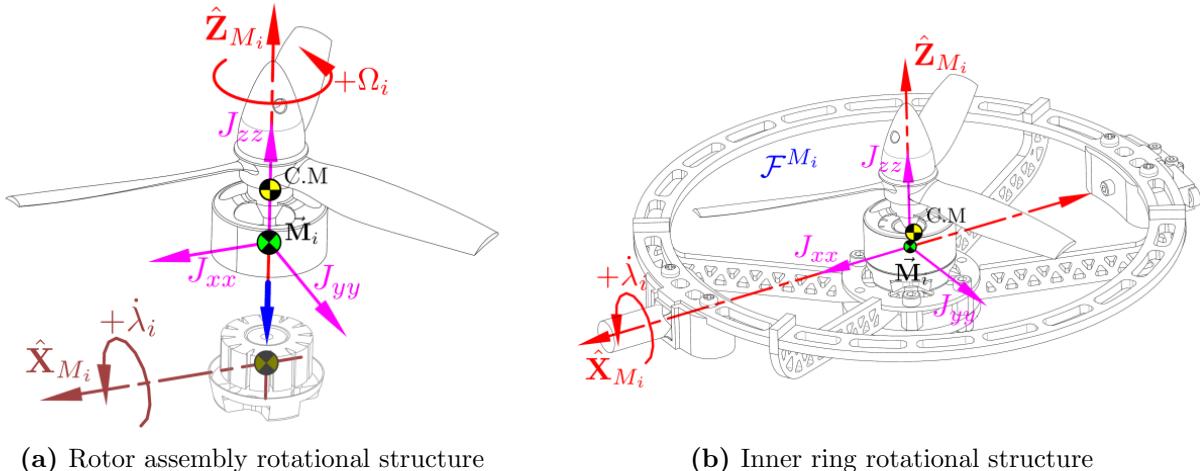
## 2.3 Inertial Matrices & Masses

In some cases when transforming inertias it is more appropriate to use rotation matrices to apply the transformation and not quaternions. Spatial rotation of inertial tensors are ill suited to quaternion parametrization.

An undesirable side effect of the relative rotations within a non-rigid body are the inertial responses 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 body. Similarly a gyroscopic cross product from rotational velocities is also present when rotating bodies that have their own relative rotation. Typically for most rigid body dynamics, such first and second order effects are negligible given that the angular rates which they depend on are 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 non-zero time varying body angular velocities and accelerations.

The dynamic effects of those torque responses are derived next in Sec:3.2.1. Both inertial and gyroscopic effects are dependent on the considered body's inertia about each respective rotational axis. The magnitude of those inertias are obviously a by-product of the structure's design. The following inertias presented are all calculated from a SolidWorks model with masses to match physical prototype measurements. Each rigidly connected body affected by the same angular velocity is grouped together. Every motor module then contains 3 connected bodies; the propeller/rotor body, the inner ring and finally middle ring assemblies, each of which are now described and expanded.

The first rotational body to consider is that of the propeller and rotor assembly (Fig:2.10a, excluding the motors stator). The "rotor" assembly, with subscript r, has a net mass  $m_r = 27$  g with a center of mass C.M<sub>r</sub> = [0.0 0.0 15.5]<sup>T</sup> [mm] relative to the entire motor modules center of rotation. The propeller's plane of rotation is [0.0 0.0 23.0]<sup>T</sup> [mm] relative to the entire body's center of rotation (previously in Fig:2.3). At high speeds the propeller's inertial contribution to the rotor assembly is simplified to a solid disc.



**Figure 2.10:** Inertial measurement references

The entire rotor assembly then has an inertia  $J_r$ , with principle inertial axes aligned as in Fig:2.10a:

$$J_r = \begin{bmatrix} 105.5 & 0.0 & 0.0 \\ 0.0 & 105.5 & 0.0 \\ 0.0 & 0.0 & 41.8 \end{bmatrix} \quad [\text{g.cm}^2] \quad (2.15)$$

The net angular velocity of the rotor assembly  $\vec{\omega}_r$  relative to the body frame is produced by the BLDC motor's own rotational velocity  $\Omega_i$  and both servo rates;  $\lambda_i$  and  $\dot{\alpha}_i$ . Each servo velocity is *transformed* onto the motor frame  $\mathcal{F}^{M_i}$ .

$$\vec{\omega}_r = \begin{bmatrix} 0 \\ 0 \\ \Omega_i \end{bmatrix} + \frac{d\lambda_i}{dt} R_x(-\lambda_i) \begin{bmatrix} \lambda_i \\ 0 \\ 0 \end{bmatrix} + \frac{d\alpha_i}{dt} R_y(-\alpha_i) R_x(-\lambda_i) \begin{bmatrix} 0 \\ \alpha_i \\ 0 \end{bmatrix} \in \mathcal{F}^{M_i} \quad (2.16)$$

*Eq:2.16 is later replaced with a quaternion operator. That equation and the remaining angular velocity equations for each body derived here are therefore not expanded further in their current rotation matrix form(s)...*

The next assembly, on which the motor frame  $\mathcal{F}^{M_i}$  is attached, is the inner ring assembly with subscript n. The inner ring structure has a mass  $m_n = 92$  [g], including the rotor assembly in that calculation. The center of mass is positioned  $C.M_n = [-1.44 \ 00.0 \ 5.14]^T$  [mm] relative to the center of rotation  $\vec{M}_i$ . The inner ring, being rotated by the  $\lambda_i$  servo about the  $\hat{X}_{M_i}$  axis, then has an inertial matrix (centered and aligned with axes as in Fig:2.10b):

$$J_n = J_{M_i} = \begin{bmatrix} 520.9 & -31.7 & -0.3 \\ -31.7 & 1826.3 & 0.0 \\ -0.3 & 0.0 & 2050.8 \end{bmatrix} \text{ [g.cm}^2\text{]} \quad (2.17)$$

The effect of rapidly spinning propellers, relative to the servo rotational velocity, on the inertia in Eq:2.17 is approximated well by a solid disc. The inner ring's inertial components are then regarded as constant with respect to  $\Omega_i$ ; moreover its center of mass is independent of that propeller's rotation. The rotational velocity of the collective inner ring assembly,  $\vec{\omega}_n$  or  $\vec{\omega}_{M_i}$  for the angular velocity of the frame  $\mathcal{F}^{M_i}$ , is similar to that of Eq:2.16 and occurs in the same frame but without the added rotor rotation of  $\Omega_i$ :

$$\vec{\omega}_n = \vec{\omega}_{M_i/b} = \frac{d\lambda_i}{dt} R_x(-\lambda_i) \begin{bmatrix} \lambda_i \\ 0 \\ 0 \end{bmatrix} + \frac{d\alpha_i}{dt} R_y(-\alpha_i) R_x(-\lambda_i) \begin{bmatrix} 0 \\ \alpha_i \\ 0 \end{bmatrix} \in \mathcal{F}^{M_i} \quad (2.18)$$

That first actuating servo for  $\lambda_i$  and its coaxial support bearing are both affixed to the intermediate middle ring assembly, with subscript m (Fig:2.11a). The intermediate frame  $\mathcal{F}^{M'_i}$  is attached to the middle ring body with a mass  $m_m = 98$  [g], excluding the inner most ring's contribution. That middle ring body alone has a center of mass  $C.M_m = [-4.70 \ 0.37 \ -0.36]^T$  [cm] relative to  $\vec{M}_i$ .

Together the inner and middle rings make the whole motor module assembly (Fig:2.11b), with a subscript  $\zeta$ , has a mass  $m_\zeta = 190$  [g]. That center of mass for the motor module varies from the inner ring's rotational position  $\lambda_i$ . The module's center of mass  $C.M_\zeta$  is calculated, with attention paid to units of measurement used, as follows:

$$C.M'_n = R_x(\lambda)(C.M_n) \quad (2.19a)$$

$$C.M_\zeta = \frac{m_m(C.M_m) + m_n(C.M'_n)}{m_m + m_n} \quad (2.19b)$$

$$C.M_\zeta(\lambda) = \frac{98[-4.70 \ 0.37 \ -0.36]^T \times 10^{-7} + 92R_x(\lambda)[-1.44 \ 0.00 \ 3.06]^T \times 10^{-8}}{190 \times 10^{-3}} \quad (2.19c)$$

$$C.M_\zeta(0) = [-2.49 \ 0.19 \ 0.04]^T \Big|_{\lambda_i=0} \text{ [cm]} \quad (2.19d)$$

The complete motor module is rotated by the  $\alpha_i$  servo about its  $\hat{Y}_{M'_i}$  axis. The module's compound body inertia,  $J_\zeta$ , is a combination of the middle ring's inertia  $J_m$  and the inner ring's inertia  $J_n$  rotated by  $\lambda_i$  about  $\hat{X}_{M_i}$  (aligned as per Fig:2.11b). The latter's contribution is dependant on the *rotation* (not transformation) angle  $\lambda_i$  which from the conservation of angular momentum theory, detailed concisely in [97], produces the net inertia of that frame;  $J_{M'_i}$  or  $J_\zeta$ :

$$\text{With } J_m = \begin{bmatrix} 2905.7 & 0.0 & 390.9 \\ 0.0 & 8446.4 & 0.0 \\ 390.9 & 0.0 & 11125.7 \end{bmatrix} \text{ [g.cm}^2\text{]} \quad (2.20a)$$

$$J_\zeta(\lambda_i) = J_{M'_i} = J_m + R_x(\lambda_i)(J_n)R_x^{-1}(\lambda_i) \quad (2.20b)$$

And an inertial matrix, for reference, at rest:

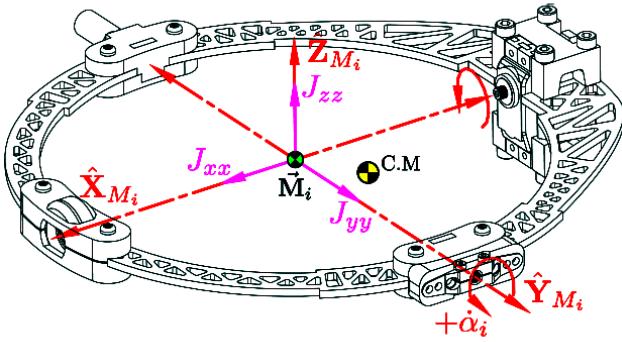
$$J_\zeta(0) = \begin{bmatrix} 3365.4 & -0.1 & 390.6 \\ -0.1 & 10210.1 & 0.0 \\ 390.6 & 0.0 & 13118.0 \end{bmatrix} \Big|_{\lambda_i=0} \quad [\text{g.cm}^2] \quad (2.20c)$$

Because  $R_x$  is a full rank square matrix, its inverse  $R_x^{-1}$  used in Eq:2.20b always exists. The modules inertia could be further divided into constant and variable components which, under certain conditions, could be simplified and neglected...

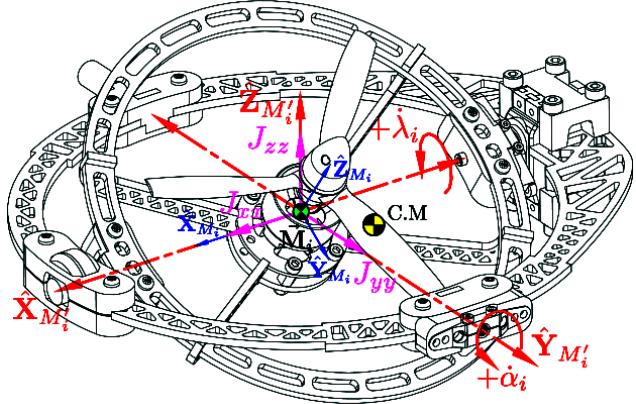
$$J_\zeta(\lambda_i) = J_{const} + J_{M_i}(\lambda_i) \quad (2.20d)$$

Noting that  $J_{M_i} = J_n$  is the inner ring's inertia from Eq:2.17, but re-orientated through a rotation  $R_x(\lambda_i)$ . Finally, the angular velocity experienced by the net motor assembly,  $\vec{\omega}_\zeta$  in frame  $\mathcal{F}^{M'_i}$ , is solely as a result of the  $\alpha_i$  servo rotation:

$$\vec{\omega}_\zeta = \vec{\omega}_{M'_i/b} = \frac{d\alpha_i}{dt} R_y(-\alpha_i) \begin{bmatrix} 0 \\ \alpha_i \\ 0 \end{bmatrix} \in \mathcal{F}^{M'_i} \quad (2.20e)$$



(a) Middle ring rotational structure



(b) Module assembly rotational structure

**Figure 2.11:** Inertial measurement references cont.

The variable inertias dependent on state inputs are one of many non-trivial aspects unique to the multibody design. Similarly the center of mass for each motor module's compound assembly coincides with neither rotational axes' intersections. As a result the effective center of mass for each module, and hence the entire body, is going to be a time varying function each motor module's angular rotational position.

The second servo which actuates  $\alpha_i$  connects the motor module (both the inner and middle ring assemblies) to the body structure. The inertial volume of that second servo and its coaxial bearing support contributes then to the body structure's inertia; whose value excludes any of the four motor modules (Fig:2.12a). Consisting of servo and bearing damping brackets, each "damping" assembly collectively weighs  $m_d = 84$  [g] and suspends the motor modules from the body frame with a set of silicone damping balls. The body structure, with a subscript  $y$ , has center of mass C.M<sub>y</sub> (without motor modules attached, Fig:2.12b). The center of mass coincides with the  $\hat{X}_b$  and  $\hat{Y}_b$  directional axes but lies  $\Delta Z = -9.52$  [mm] below the Body Frame's origin of motion;  $\vec{\mathbf{O}}_b \in \mathcal{F}^b$ .

*Note: that body frame origin  $\vec{\mathbf{O}}_b$  which all motion is calculated with respect to is co-planar to the motor module's rotational centers, not the net center of mass.*

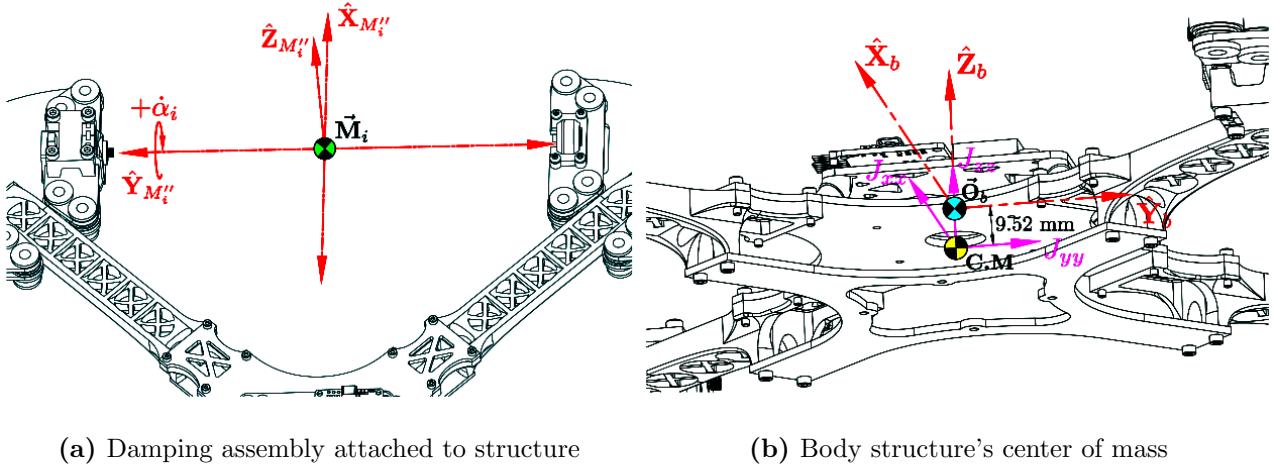


Figure 2.12: Body structure inertias

The body's weight, including all four damping assemblies and electronics, totals to  $m_y = 814.70$  [g]. Similarly the body's net inertia (*sans* motor modules)  $J_y$ , about its center of mass (Fig:2.12b), is:

$$J_y = \begin{bmatrix} 181569.7 & 0.4 & -19.4 \\ 0.4 & 181692.2 & 8.9 \\ -19.4 & 8.8 & 360067.2 \end{bmatrix} \times 10^{-7} \text{ [kg.m}^2\text{]} \quad (2.21a)$$

Using the Parallel Axis theorem to translate that inertia to the origin of motion by  $\Delta Z = +9.52$  [mm], the inertia about the origin,  $\vec{\mathbf{O}}_b$ , is:

$$J' = J + m(\vec{d} \cdot \vec{d} - \vec{d} \otimes \vec{d}) \approx J + md^2 \quad (2.21b)$$

For the general parallel axis transformation in Eq:2.21b,  $\otimes$  represents the Hamilton product of two  $[3 \times 1]$  matrices. It is used later to indicate quaternion multiplication. The vector  $\vec{d}$  is the difference between the center of mass C.M and the body frame origin  $\vec{\mathbf{O}}_b$ .

$$\therefore J' = \frac{J_y}{C.M} + m_y(\Delta \vec{Z} \cdot \Delta \vec{Z} - \Delta \vec{Z} \otimes \Delta \vec{Z}) \quad (2.21c)$$

$$\rightarrow J'_y = \begin{bmatrix} 182307.7 & 0.4 & -14.5 \\ 0.4 & 182430.1 & 6.5 \\ -14.5 & 6.5 & 360067.2 \end{bmatrix} \times 10^{-7} \text{ [kg.m}^2\text{]} \quad (2.21d)$$

Net inertia for the complete multibody vehicle,  $J_b$ , about the origin  $\vec{\mathbf{O}}_b$  is a combination of all the relative attached bodies. The assembly's inertia  $J_b$  is the *net* body frame's inertia, different from  $J_y$  which is the inertia for *only* the body structure. That collective assembly being; the four motor modules, rotated by  $\alpha_i$  and then translated to the body frame origin, and the body structure's contribution itself. Those inertial matrices' relative axial transformations from their respective centers of rotation, in frames  $\mathcal{F}^{M_i}$  for  $i \in [1 : 4]$ , to the body frame  $\mathcal{F}^b$  is analogous to that of Eq:2.13. Reiterating that  $\vec{\mathbf{O}}_b$  is *co-planar* to each module's center of rotation; each motor module's inertia,  $J_\zeta(\lambda_i)$  or  $J_{M'_i}$  defined in Eq:2.20b, is further rotated by  $\alpha_i$  about the  $\hat{Y}_{M'_i}$  axis and finally an orthogonal  $\hat{Z}_{M''_i}$  axis rotation (aligned with  $\hat{Z}_b$ ) onto  $\mathcal{F}^b$ .

For the entire body's net inertia each contributing inertial matrix must be defined with respect to the body's origin; first aligned parallel to the common set of body frame axes  $\hat{X}_b$ ,  $\hat{Y}_b$  and  $\hat{Z}_b$  and then translated to the origin  $\vec{\mathbf{O}}_b$ . Each motor module's inertia, still centered relative to each individual rotational centers  $\vec{\mathbf{M}}_i$  in Fig:2.13, but re-orientated to align parallel with  $\vec{\mathbf{O}}_b$  with rotations about axes  $\hat{X} \in \mathcal{F}^{M_i}$ ,  $\hat{Y} \in \mathcal{F}^{M'_i}$ ,  $\hat{Z} \in \mathcal{F}^{M''_i}$ .

$$J_{\vec{\mathbf{M}}_i} = R_z(\sigma_i)R_y(\alpha_i)(J_\zeta(\lambda_i))R_y^{-1}(\alpha_i)R_z^{-1}(\sigma_i) \quad (2.22a)$$

$R_z(\sigma_i)$  was defined as relative orthogonal  $\hat{Z}_b$  rotations earlier in Eq:2.13b. Expanding each module's inertia to individual inner and middle ring inertial contributions then yields:

$$\therefore J_{\vec{M}_i} = R_z R_y(\alpha_i) (J_m) R_y^{-1}(\alpha_i) R_z^{-1} + R_z R_y(\alpha_i) R_x(\lambda_i) (J_n) R_x^{-1}(\lambda_i) R_y^{-1}(\alpha_i) R_z^{-1} \quad (2.22b)$$

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

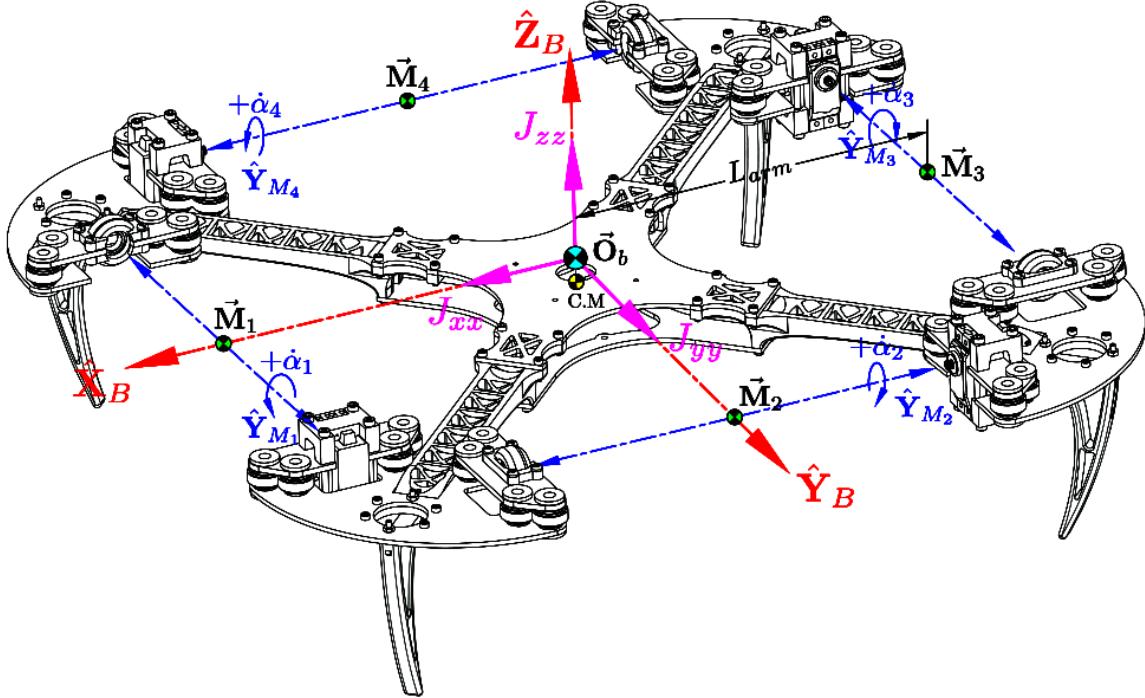


Figure 2.13: Inertial, mass and motor modules respective centers

Each module's rotational center  $\vec{M}_{1 \rightarrow 4}$  are spaced equally relative to the origin of motion,  $\vec{O}_b$ , with a parallel axis arm  $L_{arm} = 195.16$  [mm] (illustrated in Fig:2.13). To avoid notational confusion the term  $\vec{L}_i = [\pm 195.16 \ 0 \ 0]^T$  or  $[0 \ \pm 195.16 \ 0]^T$  is used to represent the vector displacement between the origin  $\vec{O}_b$  and each motor modules center of rotation  $\vec{M}_{1 \rightarrow 4}$ . The net inertial equation  $J_b(u)$ , about the origin  $\vec{O}_b$  and dependent on the actuator position matrix  $u \in \mathbb{U}$ , can be calculated as:

$$J_b(u) = J_y + \sum_{i=1}^4 J'_{\vec{M}_i} \quad [\text{kg.m}^2], \quad u \in \mathbb{U} \quad (2.23a)$$

Where  $J'_{\vec{M}_i}$  is the motor module inertia from Eq:2.22 but translated to the origin  $\vec{O}_b$  using a parallel axis theorem with  $m_\zeta = 190$  [g]:

$$J'_{\vec{M}_i} = J_{\vec{M}_i} + m_\zeta (\vec{L}_i \cdot \vec{L}_i - \vec{L}_i \otimes \vec{L}_i) \quad (2.23b)$$

Although Eq:2.23 produces the entire body's inertia, each equation to calculate  $J_{\vec{M}_i}$  involves cascaded transformations which may deteriorate the resultants certainty. Each motor module's inertia is first translated to their respective centers of rotation then rotated as per the two servos and then finally translated again back to the body frame's origin. Alternatively considering inertias about each sub-body's center of mass and then directly translating to the body frame's origin will improve the certainty of the produced inertial equations. It is also perhaps more intuitive to consider each sub-body's contribution individually, despite having been derived as combined inertial bodies in the above equations.

$$J_b(u) = J'_y + \sum_{i=1}^4 J'_n + \sum_{i=1}^4 J'_m \quad u \in \mathbb{U} \quad (2.24)$$

Isolating each body and independently considering each inertia; starting with the inner rings, each having an inertia  $J_n$  with respect to its *center of mass* (and not center of rotation) measured *relative* to its center of rotation. The following is then fundamentally different from the process in Eq:2.17, calculating the inner ring's inertial contribution about the origin  $\vec{O}_b$ .

For the inner ring, with a mass  $m_n$  and center of mass  $C.M_n$  relative to the center of rotation  $\vec{M}_i$  respectively:

$$m_n = 92 \text{ [g]} \quad (2.25a)$$

$$C.M_n = [-1.44 \ 0.0 \ 5.14]^T \text{ [mm]} \quad (2.25b)$$

The inner ring's inertial matrix at that center of mass was found to be:

$$\begin{matrix} J_n \\ C.M \end{matrix} = \begin{bmatrix} 496.6 & -31.7 & 6.6 \\ -31.7 & 1800.1 & 0.0 \\ 6.6 & 0 & 2048.9 \end{bmatrix} \text{ [g.cm}^2\text{]} \quad (2.25c)$$

The inner ring's center of mass, rotated by  $\lambda_i$  and  $\alpha_i$  servos about their respective axes, is then:

$$C.M'_n = R_z R_y(\alpha_i) R_x(\lambda_i) (C.M_n) \quad (2.25d)$$

So transforming the inertia from Eq:2.25c, still about the center of mass  $C.M_n$ , but with axes aligned parallel with the body frame, or using the shorthand  $\parallel \vec{O}_b$ :

$$\begin{matrix} J'_n(\lambda_i, \alpha_i) \\ \parallel \vec{O}_b \end{matrix} = R_z R_y(\alpha_i) R_x(\lambda_i) (J_n) R_x^{-1}(\lambda_i) R_y^{-1}(\alpha_i) R_z^{-1} \quad (2.25e)$$

The vector difference between the new, rotated center of mass to the body origin  $\vec{O}_b$  is given by:

$$\Delta L = \vec{L}_i - C.M'_n \quad (2.25f)$$

Then using the above in a parallel axis translation, adapted from Eq:2.21b, to move the rotated inertia  $J'_n$  to the center of the body frame  $\vec{O}_b$ :

$$\begin{matrix} J_n \\ \vec{O}_b \end{matrix} = \begin{matrix} J'_n \\ \parallel \vec{O}_b \end{matrix} + m_n ((\Delta L \cdot \Delta L) \mathbb{I}_{3 \times 3} - \Delta L \otimes \Delta L) \quad (2.25g)$$

And for reference, when both  $\lambda_i = 0$  and  $\alpha_i = 0$ , the inner ring's inertia is:

$$\begin{matrix} J_n \\ \vec{O}_b \end{matrix} = \begin{bmatrix} 520.9 & -31.0 & 922.6 \\ -31.0 & 36348.5 & 0.0 \\ 922.6 & 0.0 & 36573.0 \end{bmatrix} \times 10^{-7} \Big|_{\lambda_i, \alpha_i=0} \text{ [kg.m}^2\text{]}, \in \mathcal{F}^b \quad (2.25h)$$

Similarly, the same process is applied for the middle ring's rotated and translated inertia. The middle ring (*only*) has a mass and center of mass relative to the center of rotation respectively:

$$m_m = 98 \text{ [g]} \quad (2.26a)$$

$$C.M_m = [-47.00 \ 3.74 \ -3.63]^T \text{ [mm]} \quad (2.26b)$$

The inertial matrix of the middle ring body, excluding the inner ring, about its center of gravity is:

$$\begin{matrix} J_m \\ C.M \end{matrix} = \begin{bmatrix} 2879.1 & 172.3 & 223.6 \\ 172.3 & 6269.0 & 13.3 \\ 223.6 & 13.3 & 8947.5 \end{bmatrix} \text{ [g.cm}^2\text{]} \quad (2.26c)$$

The center of mass is rotated only by the  $\alpha_i$  servo about the  $\hat{Y}_{M'_i}$  axis:

$$C.M'_m = R_z R_y(\alpha_i) (C.M_m) \quad (2.26d)$$

Then the rotated inertial matrix, aligned with axes parallel to the body frame origin  $\vec{\mathbf{O}}_b$ , follows:

$$\begin{matrix} J_m \\ \parallel \vec{\mathbf{O}}_b \end{matrix} = R_z R_y(\alpha_i)(J_m) R_y^{-1}(\alpha_i) R_z^{-1} \quad (2.26e)$$

The vector difference from the rotated center of mass to the body frame origin is calculated:

$$\Delta L = \vec{L}_i - C.M'_m \quad (2.26f)$$

Which then leads to the parallel axis translation of the middle ring's inertia to the body origin:

$$\begin{matrix} J_m \\ \parallel \vec{\mathbf{O}}_b \end{matrix} = J_m + m_m((\Delta L \cdot \Delta L)\mathbb{I}_{3x3} - \Delta L \otimes \Delta L) \quad (2.26g)$$

Then, for reference at  $\alpha_i = 0$  the middle ring's inertial contribution at  $\vec{\mathbf{O}}_b$  is:

$$\begin{matrix} J_m \\ \parallel \vec{\mathbf{O}}_b \end{matrix} = \begin{bmatrix} 2905.7 & 715.4 & -303.9 \\ 715.4 & 27795.7 & 0.0 \\ -303.9 & 0.0 & 30475.0 \end{bmatrix} \times 10^{-7} \Big|_{\alpha_i=0} \quad [\text{kg.m}^2], \in \mathcal{F}^b \quad (2.26h)$$

Then, reiterating Eq:2.24, the instantaneous inertia of the entire body in motion is calculated as the contribution of the connected sub-bodies depending on the actuator matrix  $u \in \mathbb{U}$ .

$$\begin{matrix} J_b(u) \\ \parallel \vec{\mathbf{O}}_b \end{matrix} = J'_y + \sum_{i=1}^4 \begin{matrix} J_n \\ \parallel \vec{\mathbf{O}}_b \end{matrix} + \sum_{i=1}^4 \begin{matrix} J_m \\ \parallel \vec{\mathbf{O}}_b \end{matrix} \quad u \in \mathbb{U} \quad (2.27a)$$

The mass for the whole assembly is  $m_b = 1574.7$  [g]. For reference, according to Eq:2.27a, the inertial matrix for the assembly at the actuator rest conditions,  $u = \vec{0}$ , about the origin  $\vec{\mathbf{O}}_b$  is:

$$J_b(\vec{0}) = \begin{bmatrix} 317448.2 & 0.4 & -14.5 \\ 0.4 & 317570.7 & 6.5 \\ -14.5 & 6.5 & 628257.5 \end{bmatrix} \times 10^{-7} \Big|_{u=\vec{0}} \quad [\text{kg.m}^2], \in \mathcal{F}^b \quad (2.27b)$$

Unless otherwise specified; any inertia  $J_b(u)$  indicates an instantaneous calculated solution to Eq:2.27a given a particular  $u(t) \in \mathbb{U}$ . The purpose of the derivations for rotated centers of mass in Eq:2.25 and Eq:2.26 is twofold; highlighting both the inertial contributions *and* the variable center of masses for each sub-body.

Because the origin of motion in the body frame  $\mathcal{F}^b$  and the net body's center of mass are not coincidental, it is important to calculate the net center of mass position, varying with actuator positions  $u \in \mathbb{U}$ . In the general case for a collection of  $n$  bodies, with each body's center of mass at some position  $\vec{X}_i$  and each having a mass  $m_i$ , resultant center of mass is:

$$C.M = \frac{\sum_{i=1}^n m_i \cdot \vec{X}_i}{\sum_{i=1}^n m_i} \quad (2.28a)$$

Using  $C.M'_n$  and  $C.M'_m$  as rotated centers of mass defined in Eq:2.25d and Eq:2.26d respectively, the entire assembly has a variable center of mass:

$$C.M(u) = \frac{m_y C.M_y + \sum_{i=1}^4 m_n C.M'_n + \sum_{i=1}^4 m_m C.M'_m}{m_y + \sum_{i=1}^4 m_n + \sum_{i=1}^4 m_m} \quad (2.28b)$$

So the center of gravity when all actuators are at their zero positions is:  $C.M = [0 \ 0 \ -4.94]^T$  [mm]. Using a gravity vector  $\vec{G}_b = R_I^b \vec{F}_g = R_I^b [0 \ 0 \ -9.81(m_b)]^T$  [N] in the body frame  $\mathcal{F}^b$ , the resultant gravitational torque about the origin  $\vec{\mathbf{O}}_b$  in the body frame  $\mathcal{F}^b$  is:

$$\Delta C.G = \vec{\mathbf{O}}_b - C.M(u) \quad (2.28c)$$

$$\vec{\tau}_g = \Delta C.G \times \vec{G}_b \quad [\text{N.m}], \tau_g \in \mathcal{F}^b \quad (2.28d)$$

## 2.4 Electronics

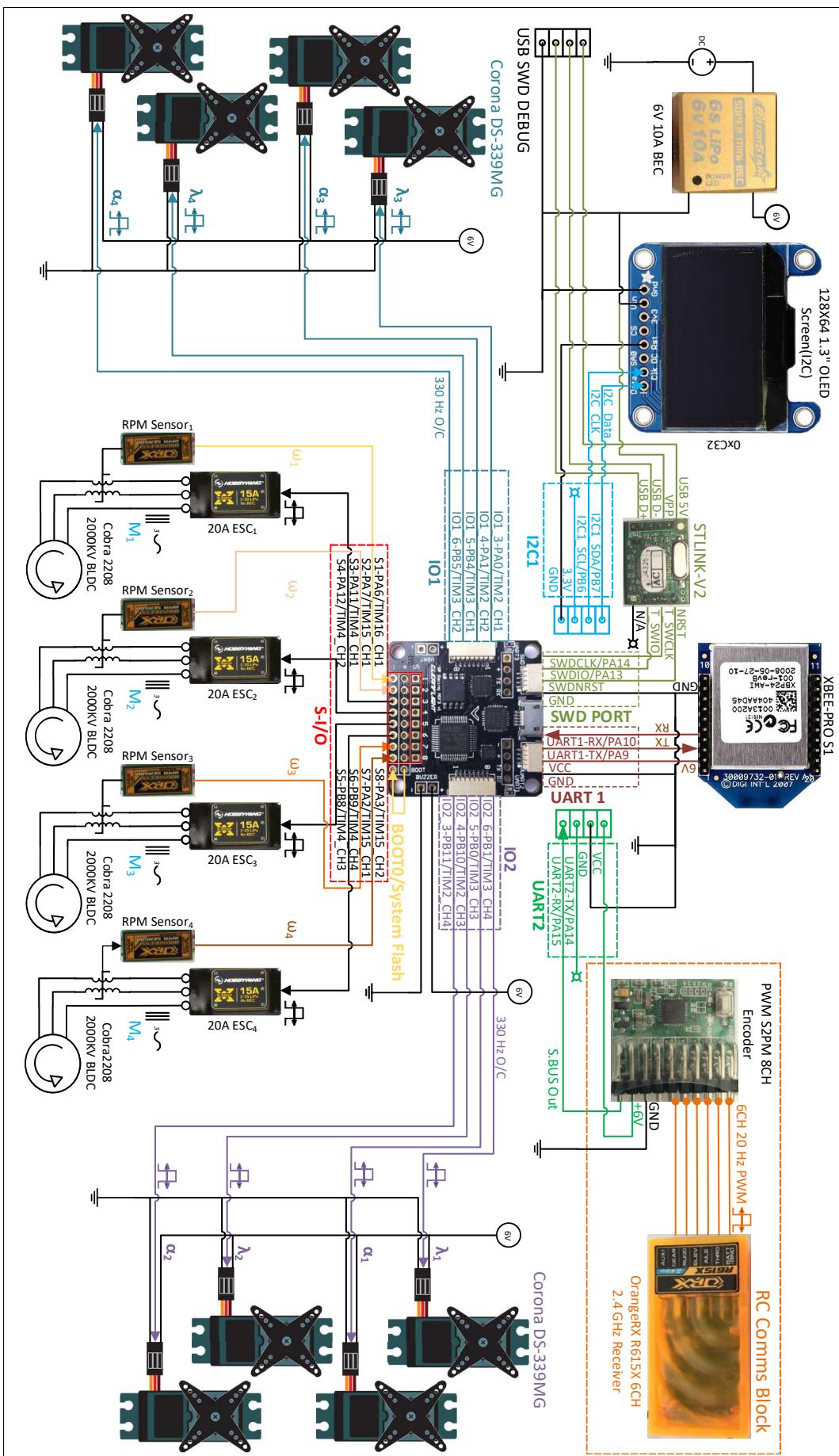
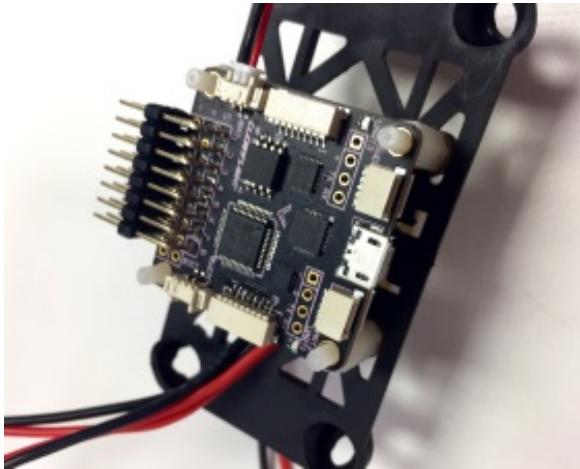
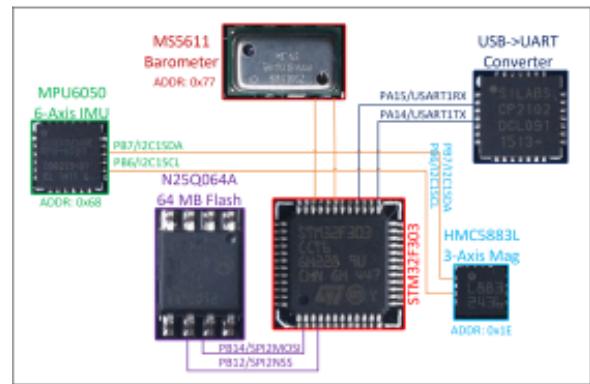


Figure 2.14: Hardware schematic diagram

An abstracted hardware diagram for the (electronic) system layout is shown in Fig:2.14. It is an illustration for the connection of different electronic peripherals used to aid the on-board control system. The structure of the implemented autopilot system and control loops are addressed later. This section aims to provide a brief overview of the specific modules used, their purpose and a description of how they are interfaced. No control loops or code structure are considered yet, those are detailed in Ch:4 and Ch:6 respectively.



(a) SPRacing F3 deluxe flight controller



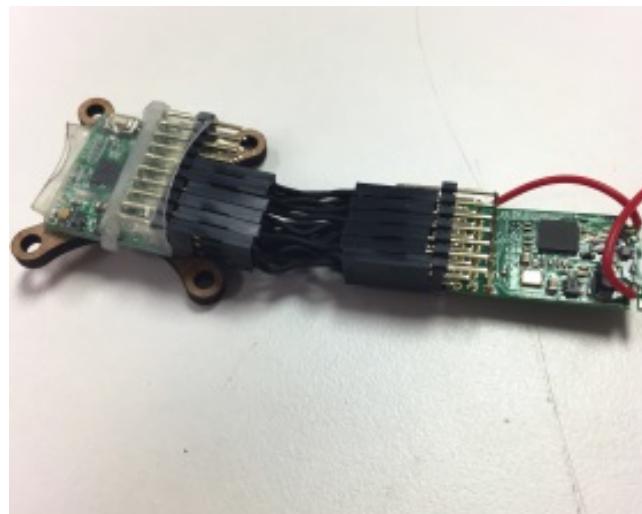
(b) F3 Deluxe on-board connections

**Figure 2.15:** SPRacing F3 deluxe layout

The embedded system is constructed around an ARM STM32F303 [119] based microcontroller. The micro-processor board is a commercial flight control board, specifically an SPRacing F3 Deluxe [28] which has had its bootloader removed and custom firmware, unique to this project, developed for it. CleanFlight or BetaFlight opensource software ([27] or [11] respectively) are typically used for the SPRacing F3; despite open source software its hardware specifications are however not openly available. The reverse engineered electrical schematic for the board is included in App:B.2 but a simplified overview of its internal connections is shown in Fig:2.15b.

The embedded software implemented is described in Ch:6; the I/O for all the peripherals are however detailed here. The flight-controller has the following onboard peripherals: an I2C MPU-6050 6-axis gyroscope & accelerometer [57] with an I2C connected HMC5883 magnetometer compass [35], an I2C MS5611 barometer [117] and finally 64 Mb of SPI flash memory.

*The combination of above sensors fused for state estimation and their associated filtration algorithms are dealt with in Sec:5.5 of Ch:5.*

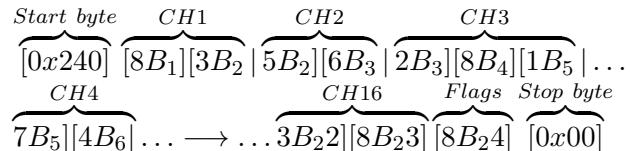
**Figure 2.16:** SBUS converter & 6CH receiver

Two separate wireless communication loops are used. First; the system relays full state information for a complete 6-DOF XYZ position and orientation autopilot system. Sent from an independent ground control station (*GCS*) using 2.4 GHz XBEE S1 module(s) [58] which is connected to the flight controller via USART. Full state-estimation, using a multi-camera system (??), and basic trajectory generation is performed on the GCS for the vehicle to track that trajectory.

Secondly; a partial trajectory (basic orientation) augmented pilot control input system, fail safe and secondary to the autopilot loop, is transmitted through a 6 channel 2.4 GHz radio frequency module. The secondary system allows for physical control without the need of a trajectory generation loop. The 6 CH received signals, otherwise permeated as six individual 20 kHz PWM signals via an OrangeRx R615x receiver [91], are encoded into a single proprietary S.BUS data stream (Fig:2.16).

The need for a serial bus (S.BUS) encoder, specifically [52], comes about as a consequence of the introduction of the 8 additional servos. As a result, there are no longer 6 free additional timer input/output channels which can be dedicated to input capture of those RC channels. Encoding the received data to a serial data line means the 6CH commands can be processed with a single RX channel. The encoder implements a USART derivative communications standard called S.BUS. Shown in Fig:2.17 the S.BUS data, captured with a logic analyzer [111], was used to ascertain the data stream's following parameters:

- 25 Bytes per packet
- 8-Bit byte length
- 1 Start byte 0x240
- 1 Byte of state flags
- 1 Stop byte 0x0
- Bytes are:
  - MSB First
  - 1 start & 2 stop bits
  - Even parity bit
  - Inverted
  - 100000 baud ( $b.s^{-1}$ )
- 22 total 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:



**Figure 2.17:** S.BUS data stream

The received information from the transmitted 6 channels is smoothed with a digital filter, using an infinite impulse response moving average filter. The filters difference equation can be as follows:

$$y_n = \left(1 - \frac{1}{N}\right)y_{n-1} + \frac{1}{N}x_n \quad (2.29)$$

Moving over an average of  $N = 5$  samples which, each with a propagation delay of 14 ms due to S.BUS transmission, the filter has a 70 ms zero order holding time. The signal's sampling delays are sufficiently faster than the transfer times of the signals are not of consequence.

Similarly all the measured RPM signals measured by the OrangeRx RPM speed sensors are filtered over 5 samples as well. Any received signals referred to are all post filtration. Filtering for state estimation made without using the inertial-measurement unit (using the camera system) is separately performed on the Ground Control Station computer.

Each of the eight digital servo actuators are controlled individually from 330 Hz center aligned PWM timer output compare channels (TIM2:CH1→CH4 & TIM3:CH1→CH4). Output pulses typically range from 1 – 2 [ms] to linearly control the rotational position. The servos exact range and transfer function(s) is empirically determined next in Sec:2.4.1. The four 20 [A] brushless DC electronic speed controllers (*ESCs*) are each driven from a 20 Hz PWM output (TIM4:CH1→CH4), similarly with 1 – 2 [ms] input pulse widths.

There is a total of 12 PWM output compare signals drawn from the flightcontroller, 8 for the servos and 4 for the ESCs. The servos are powered by a regulated 6 [V] DC 10 [A] power supply [51] whilst the ESCs switch unregulated 14.1 [V] DC supplied from an external power tethered. The DC supply could be drawn from a bank, which would adversely affect the weight.

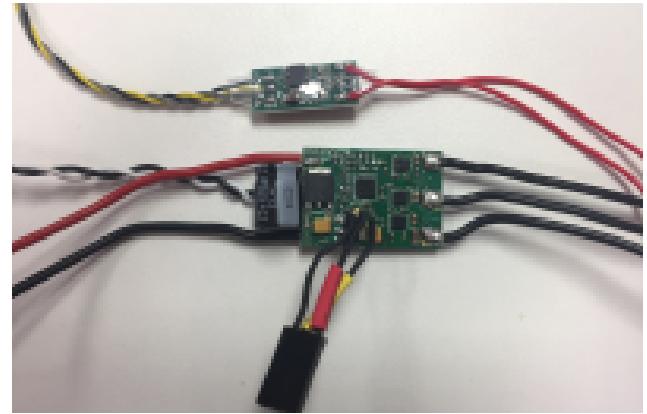
There is no integrated feedback for instantaneous RPM values available from the ESCs. Dedicated OrangeRX BLDC RPM sensors, [50], are used to measure each of the four motor's rotational speeds. Despite being termed *brushless DC motors*, the motors are actually 3-phase motors which, but when used with an ESC they behave like closed loop DC motors. The RPM sensors physically measure switching phases across two of the three motor phases, following that exact RPM can be ascertained. In general, the switching signal of a 3-Phase induction motor is shown by [76] to be proportional to the rotational velocity:

$$F_{rps} = \frac{2 \times F_{poles}}{\text{No. of rotor poles}} \quad [\text{Hz}] \quad (2.30)$$

The output signal generated by the OrangeRx RPM sensors varies the period of a 50% duty cycle square wave, that wave frequency is directly proportional to the motor's pole switching frequency. The sensor output signal has a gain of 7 for the 14 pole BLDC Cobra motors. That gain is verified with the linear relationship(s) physically measured using an optical rotation sensor, plotted in Fig:2.19. Knowing exact RPM rates means the subsequent thrust and aerodynamic torques for the control plant inputs can be calculated with greater certainty.



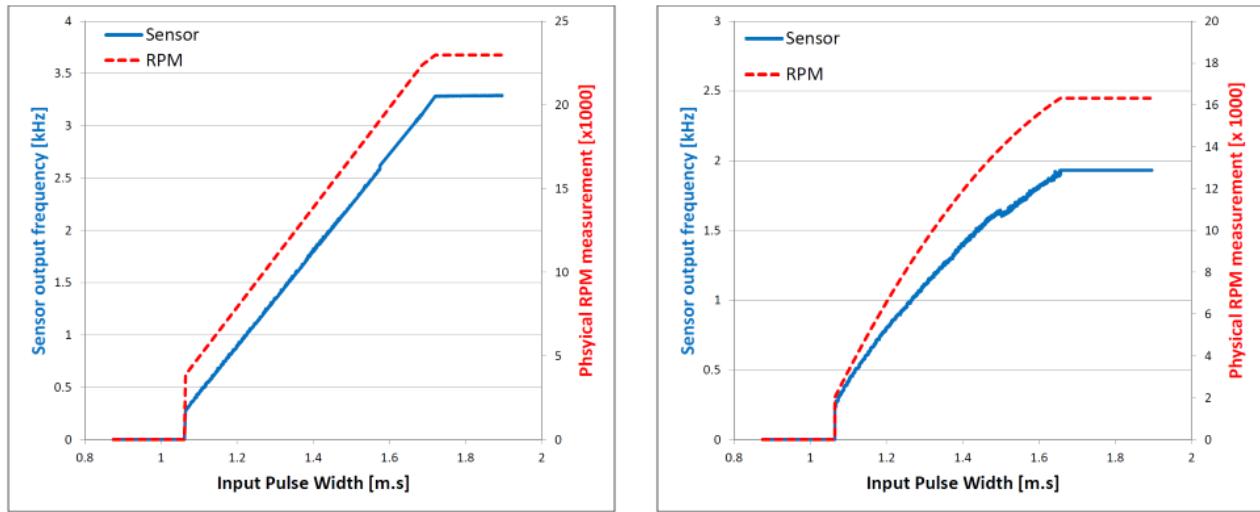
(a) XRotor 20A ESC connection guide [49]



(b) LDPower 20A ESC with RPM sensor

**Figure 2.18:** BLDC electronic speed controllers

The speed controllers, although LDPower 20A devices, are all re-flashed with BLHeli firmware [12]. The LDPower ESCs (Fig:2.18b) match Hobbywing Xrotor 20A ones (Fig:2.18a) which both use SiLabs F396 microcontrollers; the same firmware can be flashed onto the MCUs. The custom BLHeli software provides greater refinement over parameter configuration like the deflection range of inputs, however, default values were used. The plot in Fig:2.19a shows the rotation per second, or otherwise frequency in Hz, speed curve for an unloaded motor; similarly in Fig:2.19b shows the speed curve when loaded for a  $6 \times 4.5$  prop.



(a) RPM sensor plot - no load

(b) RPM sensor plot - 6X4.5 prop

Figure 2.19: RPM sensor calibration plots

It is worth noting the loaded speed plot in Fig:2.19b is slightly quadratic; the loaded response is from the second order aerodynamic torque term, quadratic with respect to RPS, expanded on in Sec:3.3.1. Moreover, when the motor is torque loaded by the propeller, the ESC current limits rotational speeds at just over 16 kRPM.

Timers channels are used to measure the varying frequency output from 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 communication line is an I2C O-LED display for debugging and status update purposes.

Any STM32  $\mu$ controller is programmed through a dedicated debugging device. The ST-Link V2 [118] is the current proprietary device which, itself, is a specially programmed STM32F10 chip. The chip connects to the dedicated Serial Wire Debugging ports of the target STM (*SWD-CLK*, *SWD-IO* & *SWD-NRST*) and is interfaced via regular USBD+ and USBD- data lines.

#### 2.4.1 Actuator Transfer Functions

##### Servo Transfer Functions

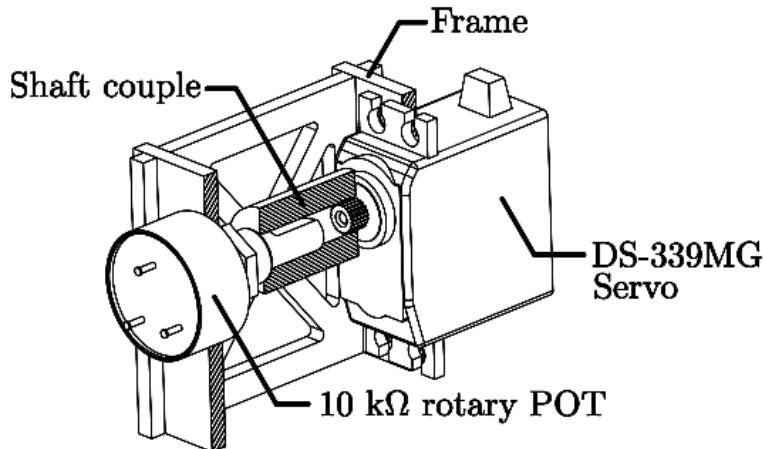
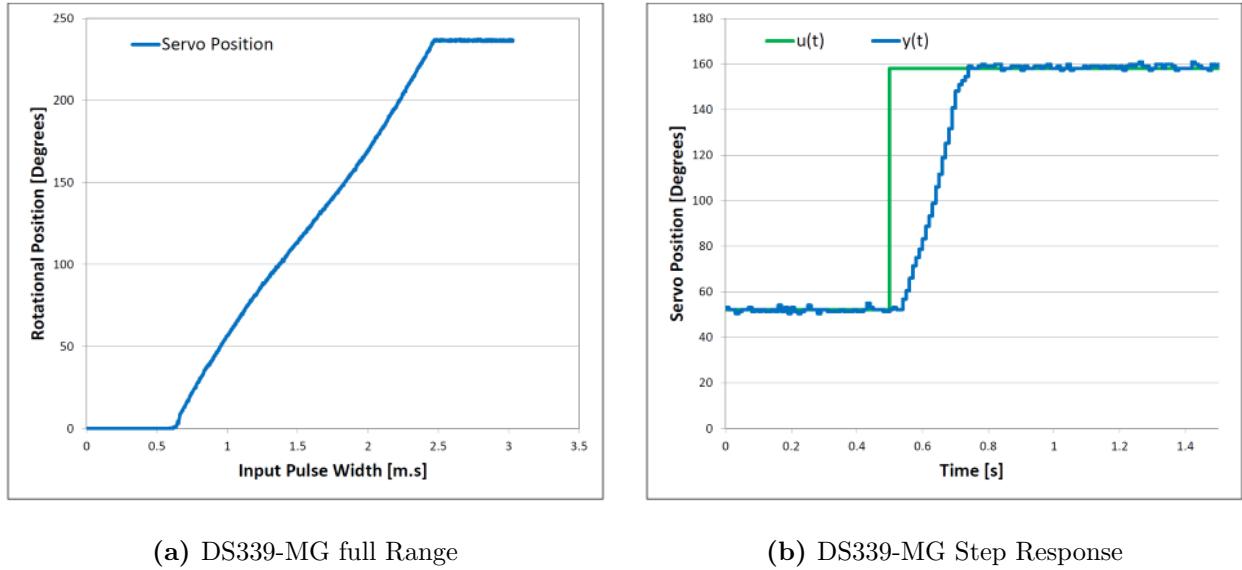


Figure 2.20: Servo transfer function test rig

The full scale deflection for digital servos are in fact greater than their quoted 180° range. Each servo has a rotational input range of around 230° (Fig:2.21a). In the prototype control loop for 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 to be validated the simulation would need to represent the servo's response accurately.

Seeing that the 180° limitation was imposed as a design decision, one of the first points of contention is the effect such a constraint would have on the feasible operating trajectories. The control algorithms derived in Ch:4 are first tested with an ideal, continuous rotation servo actuator with similar rate limits and transfer characteristics. Following that servo saturation limitations are introduced and the constraints to feasibly achievable trajectories are investigated.



**Figure 2.21:** Unloaded servo transfer characteristics

For the servos rotational range and step response are shown in Fig:2.21, the relationship between the input pulse-width  $x$  [m.s] and the rotational output position  $y$  [°] is given by:

$$y(x) = \begin{cases} 0^\circ & x < 0.65 \text{ ms} \\ 129.12x - 82.64 & 0.64 \text{ ms} \leq x \leq 2.46 \text{ ms} \\ 230^\circ & x > 2.46 \text{ ms} \end{cases} \quad (2.31)$$

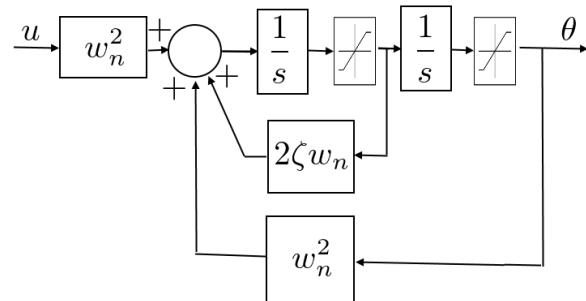
In practice the equation Eq:2.31 is changed such that 0° offset is taken at around a 50% input, making its operational range  $\pm 90^\circ$ . Each servo is mechanically rate limited to 60°/0.15s or 400 RPS with a dead time of  $t_d \approx 1.2$  [m.s] and a (*negligible*) mechanical deadband of 4 [μs]. Each servo has an approximate (*critically damped*) second order transfer function

$$G_{servo}(s) = e^{-t_{ds}} \frac{w_n^2}{s^2 + 2\zeta w_n s + w_n^2} = e^{-0.012s} \frac{(15.717)^2}{s^2 + 2(1)(15.717)s + (15.717)^2} \quad (2.32a)$$

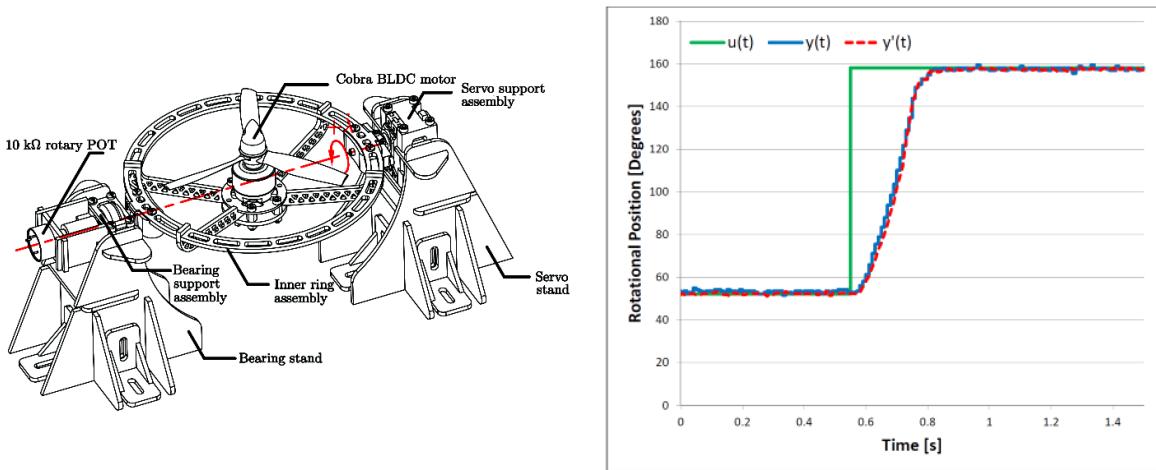
With saturation limits:

$$Y_{servo}(s) = \begin{cases} 0^\circ & |U(s)| < 0.65 \\ G(s) & 0.65 \leq |U(s)| \leq 2.46 \\ 230^\circ & |U(s)| > 2.46 \end{cases} \quad (2.32b)$$

The net transfer block for the servo is shown in Fig:2.22, including saturating non-linearities but neglecting the mechanical deadband...

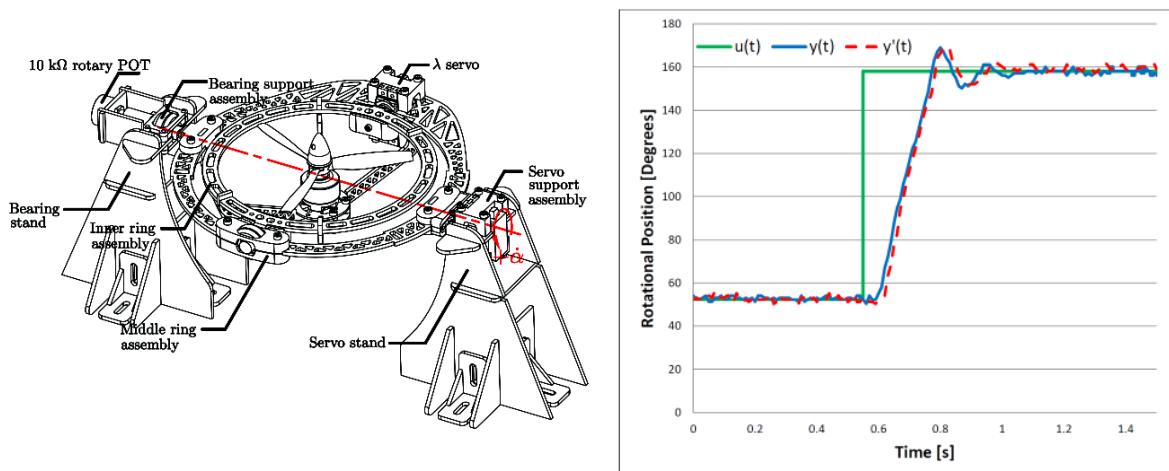
**Figure 2.22:** Servo block diagram

The plot previously in Fig:2.21b shows the transfer characteristics at the shaft output of an unloaded servo. When rotating the inertial body of the inner ring assembly, arranged as in Fig:2.23, the plant response plot of  $y(t)$  is consistent with the transfer function in Eq:2.32. Despite rotating a load and hence requiring a greater torque, the servo's characteristics remains unchanged, even when the BLDC motor (with a  $6 \times 4.5$  prop) with a rotational velocity of 6500 RPM is introduced, plotted  $y'(t)$ , further increasing the torque load of the assembly as a result of gyroscopic resistance, Eq:2.15.



(a) Inner ring servo rig

(b) Servo response plot

**Figure 2.23:** Inner ring servo characteristics

(a) Middle ring servo test rig

(b) Servo response plot

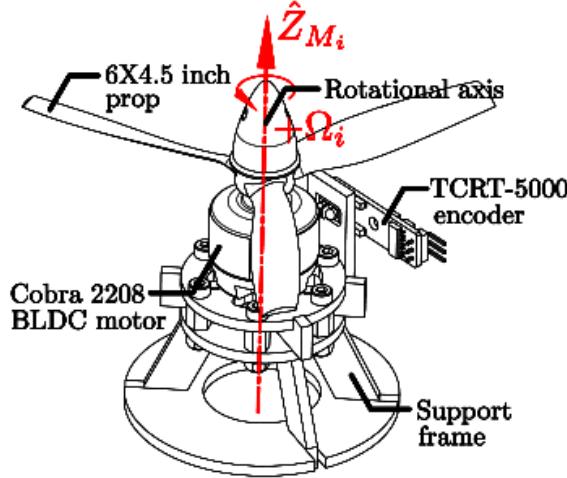
**Figure 2.24:** Middle ring servo characteristics

In Fig:2.24 the response for the middle ring servo is shown. Its rate transients remain the same but oscillations occur at the settling point, introduced by the larger inertia being driven, which has flex within its structure. The rotational position of the assembly was measured with respect to the

bearing supported output shaft, coaxial to the servos, and *not* the servo's output shaft (Fig:2.24a). The oscillations are still present under load, plotted in  $y'(t)$ , despite the frame being tensioned by the thrust vector. The mechanical harmonics can be accounted for by either introducing a more rigid sub-frame, limiting the maximum angular rate or simply adding a state dependent uncertainty term to the control plant.

### BLDC Transfer Functions

Each Cobra 2208 BLDC motor, when loaded with a  $6 \times 4.5$  inch propeller has a quadratic speed curve, plotted in Fig:2.26a. This is as a result of the propeller's opposing aerodynamic drag, *approximately* proportional to the square of the propellers angular velocity (more on propeller aerodynamics in Sec:3.3.1).



**Figure 2.25:** BLDC rpm speed calibration and transfer function rig

Through the BLHeli interface, the input range for the motor's speed controllers can be adjusted, but for the purposes of this project were left unchanged. That input relationship between pulse-widths to the ESC and output RPM sensor signal is given by the hybrid state equations for input range limits:

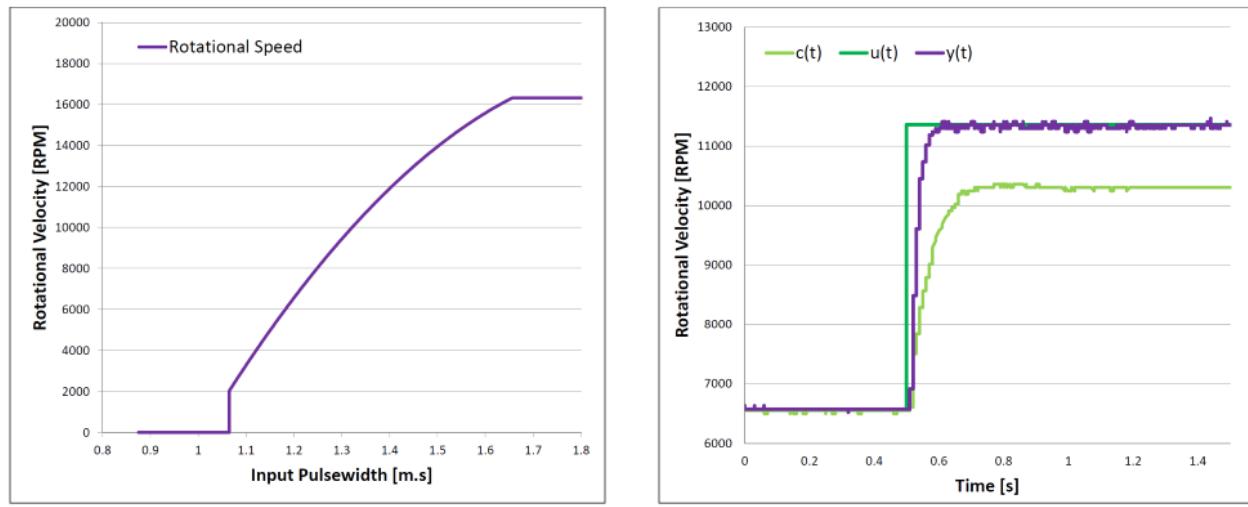
$$y(x) = \begin{cases} 0 & x < 1.065 \text{ ms} \\ -20593x^2 + 80187x - 60004 & 1.065 \text{ ms} \leq x \leq 1.655 \text{ ms} \\ 16300 & x > 1.655 \text{ ms} \end{cases} \quad [\text{RPM}] \quad (2.33)$$

The upper limit in Eq:2.33 and the motor's step response are both governed by the ESC's maximum current limit; in this case 20 [A]. Imposing 10 [A] current limiting, a consequence of using lower power ESCs is plotted  $c(t)$  in Fig:2.26b, significantly restricts the motor's transient and steady-state performance. The motor's step response,  $y(t)$ , has a negligible dead time and 2<sup>nd</sup> order dynamics, with a transient time constant far faster than the servo's plant. The motor's transfer function for speed in RPM is:

$$G_{BLDC}(s) = \frac{1}{(1 + 1.7583s \times 10^{-3})(1 + 1.7494s \times 10^{-3})} \quad [\text{RPM}] \quad (2.34a)$$

And saturation limits:

$$Y_{BLDC}(s) = \begin{cases} 0 & |U(s)| < 1.065 \\ G(s) & 1.065 \leq |U(s)| \leq 1.655 \\ 16300 & |U(s)| > 1.655 \end{cases} \quad (2.34b)$$



(a) BLDC RPM range

(b) Cobra BLDC step response

Figure 2.26: BLDC motor characteristics

# Chapter 3

## Kinematics & Dynamics

The following generally applicable, rigid body dynamics are first derived with respect to net forces and torques. Thereafter, those dynamics are adapted to the non-linear, multibody case where constrained relative rotational motion between bodies is permitted. Following that, aerodynamic effects are incorporated into the plant's model. Finally a consolidated, quaternion based plant model is presented which is used for the later control plant development in Ch:4.

### 3.1 Rigid Body Dynamics

#### 3.1.1 Lagrange Derivation

Fundamentally any body, rigid or otherwise, can undergo two kinds of motion, namely rotational and translation. Often a Lagrangian approach for combined angular and translational movements is used to derive the differential equations of motion for each degree of freedom [101, 123]. The Lagrangian principle ensures that (translational and rotational) kinematic energies and potential energy are conserved throughout the system's trajectory progression. When combined with Euler-Rotational equations, the Euler-Lagrangian [125] formulation fully defines the aerospace 6-DOF equation set.

Lagrangian formalism is regarded as especially useful in non-Cartesian (*spherical etc...*) co-ordinate frames or multi-body systems. With that being said, Cartesian co-ordinates were defined in Sec:2.2.2 for this system. An alternative relative co-ordinate system could be used for implicit-Euler based dynamics [88]. Rigid body dynamics in a Cartesian co-ordinate frame do lend themselves to Newtonian mechanics. The Newton-Euler and Euler-Lagrange formulations both result in the same differential equations of motion, but follow different derivations. The Lagrangian operator,  $\mathcal{L}$ , is a term consisting of the difference between kinetic and potential energies,  $T$  and  $U$  respectively. Considering some generalized path co-ordinates  $\vec{\mathbf{r}}(t)$ , for both position  $\vec{\xi}$  and attitude  $\vec{H}$  co-ordinates;

$$\vec{\mathbf{r}}(t) = \begin{bmatrix} \vec{\xi} & \vec{H} \end{bmatrix}^T \quad \in \mathcal{F}^\Lambda \quad (3.1)$$

The co-ordinates in Eq:3.1 are generalized and taken with respect to some shared frame  $\Lambda$ . Those generalized co-ordinates are later refined to Cartesian body co-ordinates with respect to the inertial frame. The Lagrangian is, by definition:

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

With the trajectory's kinetic and potential energy function(s) are  $T$  and  $U$  respectively. Introducing a rigid body's general (translational and rotational) kinetic and potential energies, together in the shared reference frame  $\mathcal{F}^\Lambda$ .

There is no angular contribution of potential stored energy, so  $U(\vec{r}, \dot{\vec{r}})$  depends only on gravitational potential energy. With gravity defined as  $G_I = [0 \ 0 \ -9.81]^T$  [ $\text{m.s}^{-2}$ ] in the Inertial frame,  $\in \mathcal{F}^I$ , the Lagrangian scalar is then:

$$\mathcal{L}(\vec{r}, \dot{\vec{r}}, t) = \frac{1}{2} \dot{\vec{\xi}}^T (m_b) \dot{\vec{\xi}} + \frac{1}{2} \dot{\vec{H}}^T (J_b) \dot{\vec{H}} - m_b \vec{G}_\Lambda z \quad (3.2b)$$

Noting that  $J_b$  is body's inertial matrix aligned and translated with respect to the common frame  $\mathcal{F}^\Lambda$ . The Euler-Lagrange formulation equates partial derivatives of the Lagrangian to any generalized forces,  $\vec{V}$ , acting on the system in  $\Lambda$ . In the rigid body motion case the generalized forces are net forces  $\vec{F}_{net}$  torques  $\vec{\tau}_{net}$  in the shared frame  $\in \mathcal{F}^\Lambda$ .

$$\frac{d}{dt} \left( \frac{\delta \mathcal{L}}{\delta \dot{\vec{r}}} \right) - \frac{\delta \mathcal{L}}{\delta \vec{r}} = \vec{V} = \begin{bmatrix} \vec{F}_{net} \\ \vec{\tau}_{net} \end{bmatrix} \in \mathcal{F}^\Lambda \quad (3.3)$$

Then taking the partial derivatives of Eq:3.2b with respect to the path co-ordinates  $\vec{r}(t)$ :

$$\frac{\delta \mathcal{L}}{\delta \vec{r}} = \begin{bmatrix} m_b \vec{G}_\Lambda \\ 0 \end{bmatrix} \in \mathcal{F}^\Lambda \quad (3.4a)$$

$$\frac{d}{dt} \left( \frac{\delta \mathcal{L}}{\delta \dot{\vec{r}}} \right) = \left[ \frac{d}{dt} m_b \dot{\vec{\xi}} \quad \frac{d}{dt} J_b \dot{\vec{H}} \right]^T \in \mathcal{F}^\Lambda \quad (3.4b)$$

Where  $\vec{G}_\Lambda$  is the gravitation force in the common frame  $\mathcal{F}^\Lambda$  which  $\mathcal{L}(\vec{r}, \dot{\vec{r}})$  is with respect to. The body mass  $m_b$  and inertia  $J_b$  could potentially have some non-zero time derivative, but here are considered to be constant. Time varying inertias are addressed later in Sec:3.2.1. Any vector in a rotating generalized coordinate system has time derivative relative to another reference frame as per the Reynolds Transportation Theorem [120]:

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

So applying Eq:3.5 to the partial derivatives in Eq:3.4b and further defining the generalized coordinates  $[\vec{\xi}, \vec{H}]^T$  as 6-DOF Cartesian body coordinates with respect to an inertial frame (the body frame  $\mathcal{F}^b$  and inertial frame  $\mathcal{F}^I$ ). Reiterating that the angular orientations  $\vec{\xi}$  are with respect to a common frame  $\mathcal{F}^\Lambda$ , unlike Euler angles  $\vec{\eta} \in \mathcal{F}^{v2,v1,I}$ . Recalling the definition of attitude in a common frame  $\vec{\eta}_b$  from Eq:2.9d, where  $\vec{\omega}_b = \vec{\eta}_b$  and  $\vec{\eta}_b \in \mathcal{F}^b$ , the trajectory is refined:

$$\vec{r} = [\vec{\xi} \ \vec{H}]^T \triangleq \begin{bmatrix} \vec{\mathcal{E}} \\ \vec{\eta}_b \end{bmatrix} \rightarrow \dot{\vec{r}} = [\dot{\vec{\xi}} \ \dot{\vec{H}}]^T \triangleq \frac{d}{dt} \begin{bmatrix} \vec{\mathcal{E}} \\ \vec{\eta}_b \end{bmatrix} = \begin{bmatrix} \vec{\nu} \\ \vec{\omega} \end{bmatrix} \in \mathcal{F}^b \quad (3.6)$$

It then follows that the Lagrangian in Eq:3.2b changes to:

$$\mathcal{L} = \frac{1}{2} \vec{\nu}^T (m_b) \vec{\nu} + \frac{1}{2} \vec{\omega}^T (J_b) \vec{\omega} - m_b \vec{G}_b z \in \mathcal{F}^b \quad (3.7a)$$

$$\frac{d}{dt} \left( \frac{\delta \mathcal{L}}{\delta \dot{\vec{r}}} \right) = \left[ m_b \frac{d}{dt} \vec{\nu} \quad J_b \frac{d}{dt} \vec{\omega} \right]^T \quad (3.7b)$$

$$\rightarrow m_b \frac{d}{dt} \vec{\nu} = m_b \dot{\vec{\nu}} + \vec{\omega}_{I/b} \times \vec{\nu} \quad (3.7c)$$

$$\rightarrow J_b \frac{d}{dt} \vec{\omega} = J_b \dot{\vec{\omega}} + \vec{\omega}_{I/b} \times J_b \vec{\omega} \quad (3.7d)$$

Which, when substituted back into the Euler-Lagrange formulation Eq:3.3, produces the familiar Newton-Euler rigid body equations for translational and rotational motion;

$$\vec{F}_{net} = m_b \dot{\vec{\nu}} + \vec{\omega}_b \times m_b \vec{\nu} - m_b R_I^b(-\eta) \vec{G}_I \in \mathcal{F}^b \quad (3.8a)$$

$$\vec{\tau}_{net} = J_b \dot{\vec{\omega}}_b + \vec{\omega}_b \times J_b \vec{\omega}_b \in \mathcal{F}^b \quad (3.8b)$$

Reiterating that  $\vec{\eta}_b \neq \vec{\eta}$  because each Euler Angle is defined in a sequentially rotated reference frame. Four equations then completely describe a body position's and attitude's state dynamics:

$$\dot{\vec{\mathcal{E}}} = R_b^I(-\eta)\vec{\nu} \quad \in \mathcal{F}^I \quad (3.9a)$$

$$\vec{F}_{net} = m_b\dot{\vec{\nu}} + \vec{\omega}_b \times m_b\vec{\nu} - m_bR_I^b(-\eta)\vec{G}_I \quad \in \mathcal{F}^b \quad (3.9b)$$

$$\dot{\vec{\eta}} = \Phi(\eta)\vec{\omega}_b \quad \in \mathcal{F}^{v2,v1,I} \quad (3.9c)$$

$$\vec{\tau}_{net} = J_b\dot{\vec{\omega}}_b + \vec{\omega}_b \times J_b\vec{\omega}_b \quad \in \mathcal{F}^b \quad (3.9d)$$

Where  $\Psi(\eta)$  is the Euler matrix defined previously in Eq:2.9e. State differentials from Eq:3.9 can be simplified to a set of two equations defined in the reference frames of the state variables which they represent. The non-linear form of those equations substitutes  $d\vec{\eta}/dt = \Phi(\eta)\vec{\omega}_b$  into the Lagrangian derivative, Eq:3.4b.

$$\frac{d}{dt}\left(\frac{\delta\mathcal{L}}{\delta\dot{\mathbf{r}}}\right) = \begin{bmatrix} m_b\frac{d}{dt}\vec{\nu} & J_b\frac{d}{dt}\dot{\vec{\eta}}_b \end{bmatrix}^T \Rightarrow \begin{bmatrix} m_b\frac{d}{dt}\vec{\nu} & J_b\frac{d}{dt}\Phi(\eta)\vec{\omega}_b \end{bmatrix}^T \quad (3.10)$$

This only affects the angular component as the two kinetic energies are independent of one another. Applying the differential chain rule yields:

$$J_b\frac{d}{dt}\Phi(\eta)\vec{\omega}_b = J_b(\Phi(\dot{\eta})\vec{\omega}_b + \Phi(\eta)\dot{\vec{\omega}}_b) \quad (3.11)$$

Drawing from [88] and recognizing that  $J_b$  must be transformed to the common intermediate Euler axes,  $\mathbb{J} = \Psi(\eta)^T J_b \Psi(\eta)$ . The state differential equation for angular acceleration in Eq:3.8b, then defined in intermediate (non-inertial) Euler frames for each angle, becomes:

$$M(\eta)\ddot{\vec{\eta}} + C(\eta, \dot{\eta})\dot{\vec{\eta}} = \Psi(\eta)\vec{\tau}_{net} \quad \in \mathcal{F}^{v2,v1,I} \quad (3.12a)$$

$$M(\eta) = \Psi(\eta)^T J_b \Psi(\eta) \quad (3.12b)$$

$$C(\eta, \dot{\eta}) = -\Psi(\eta)J_b\Psi(\dot{\eta}) + \Psi(\eta)^T[\Psi(\eta)\dot{\vec{\eta}}]_x J_b\Psi(\eta) \quad (3.12c)$$

The relationship  $\dot{\Phi} = \Phi\dot{\Psi}\Phi$  was used to simplify Eq:3.12, the singularity in  $\Phi$  still remains. The equation in Eq:3.12a fully describes the state derivative  $\ddot{\vec{\eta}}$  in its own reference frame(s),  $\mathcal{F}^{v2,v1,I}$ . The two differential equations which fully describe the entire body's motion are then:

$$\vec{F}_{net} = m_b\dot{\vec{\mathcal{E}}} + R_b^I(-\eta)\vec{\omega}_b \times m_b\dot{\vec{\mathcal{E}}} - m_b\vec{G}_I \quad \in \mathcal{F}^I \quad (3.13a)$$

$$\vec{\tau}_{net} = \Psi(\eta)^{-1}M(\eta)\ddot{\vec{\eta}} + \Psi(\eta)^{-1}C(\eta, \dot{\eta}) \quad \in \mathcal{F}^{v2,v1,I} \quad (3.13b)$$

The generalized net forces and torques acting on the body,  $\vec{F}_\mu$  and  $\vec{\tau}_\mu$  respectively, are the system's controllable inputs which are directly affected by the system's actuators. How the actuator suite affects the net forces and torques depend on the actuator's associated effectiveness function. In the general case, which is refined in Sec:3.3, the control inputs for a quadrotor are typically as follows. The net force acting on the system is simply the sum of all thrust forces produced by rotating propellers  $\vec{T}(\Omega_i)$ ;

$$\vec{F}_\mu = \sum_{i=1}^4 \vec{T}(\Omega_i) \quad [\text{N}], \quad \in \mathcal{F}^b \quad (3.14a)$$

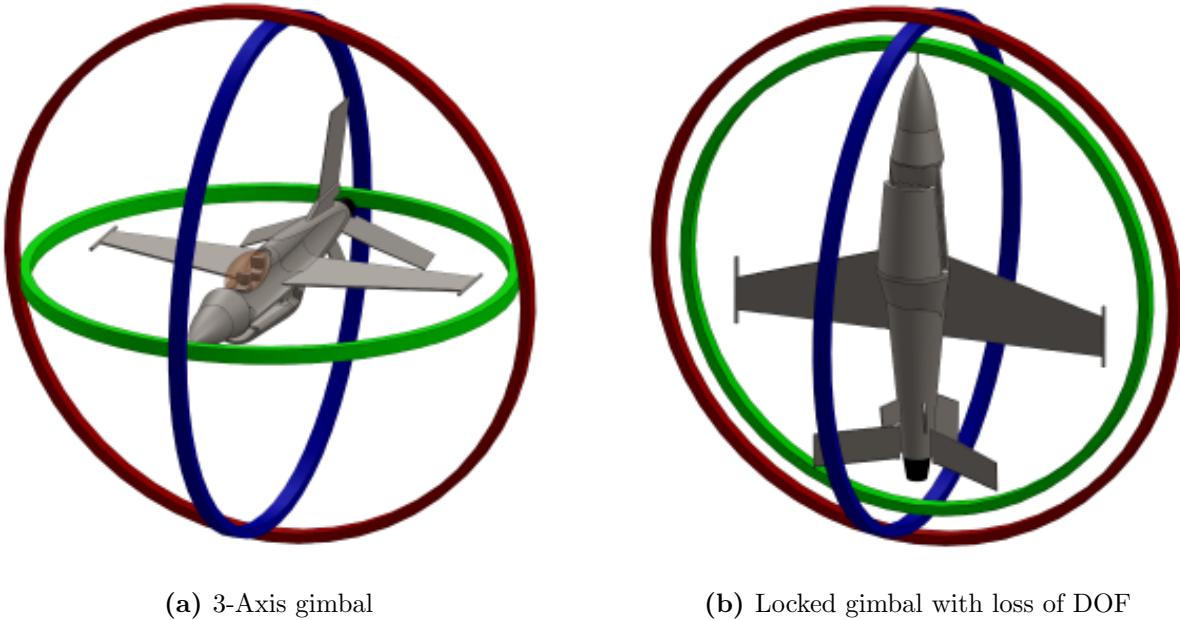
Similarly the net torque is the sum of all *differential* torque arms produced from opposing propeller thrust vectors.

$$\vec{\tau}_\mu = \sum_{i=1}^4 \vec{l}_i \times \vec{T}(\Omega_i) \quad [\text{N.m}], \quad \in \mathcal{F}^b \quad (3.14b)$$

Where  $\vec{T}(\Omega_i)$  is the  $i^{th}$  motor's thrust vector, fixed in the  $\hat{Z}_b$  direction. The thrust vectors could potentially be  $\in \mathbb{R}^3$ . Similarly  $\vec{l}_i$  is that thrust vector's perpendicular displacement from the origin  $\mathbf{O}_b$ . The above equations are still applicable to any 6 DOF body, common simplifications applied to the system(s) for quadrotor control are explored in App:A.1. Aspects unique to (multibody) aerospace frames are now introduced. Obviously the contextual focus is on quadrotor and tilting quadrotor platforms...

### 3.1.2 Rotation Matrix Singularity

The singularity inherent to Euler Angle parametrization is often mentioned but far less common is the mathematical demonstration of how that singularity manifests itself. A singularity occurs for some matrix  $A$  in  $\vec{y} = A\vec{x}$  has a zero determinate, losing rank and differentiability. The combined rotation matrix from the  $\mathcal{F}^I$  to  $\mathcal{F}^b$  is the singular component of an Euler parametrized sequence. Considering the case of a rotational 3-axis gimbal system(Fig:3.1a) which mimics the sequential nature of the Euler set. When the intermediary sequenced rotational angle is at  $\pi/2$ , the remaining two axes become collinear (Fig:3.1b). In Z-Y-X rotation sequence adopted in this work, the singularity occurs from the rolling angle  $\theta$  about  $\hat{Y}$ . Both the pitch  $\phi$  or yaw  $\psi$  rotations will subsequently have the same rotational effect. Such a situation results in as a loss of a degree of freedom.



**Figure 3.1:** Mechanical gimbal lock

What is clear physically is not necessarily as obvious mathematically. A loss of rank occurs in the Euler Matrix  $\Psi(\eta)$ , defined previously in Eq:2.9g from Sec:2.2.1. That relation between angular velocity, in the inertial frame or inversely in the body frame, and the angular rates of the Euler Angles has a determinant:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)\sec(\theta) & \cos(\phi)\sec(\theta) \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \Phi(\eta)\omega_b \quad (3.15)$$

$$\rightarrow \det(\Phi(\eta)) = \cos(\phi)(\cos(\phi)\sec(\theta)) + \sin(\phi)(\sin(\phi)\sec(\theta)) = \sec(\theta) \quad (3.16)$$

$$\therefore \lim_{\theta \rightarrow \pi/2} |\Phi(\eta)| = \sec(\theta) \rightarrow \infty \quad (3.17)$$

The Euler matrix  $\Phi(\eta)$  loses rank as  $\theta \rightarrow \pi/2$ , loosing differentiability as well. The physical consequence of this is the loss of a degree of freedom. More specifically, if one looks at how the Z-Y-X rotation (or transformation) matrices are formulated, from Eq:2.3.

$$R_I^b = R_z R_y R_x = \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi & c_\phi \end{bmatrix} \quad (3.18a)$$

$$R_I^b(\eta) = \begin{bmatrix} c_\psi c_\theta & c_\psi s_\theta s_\phi - s_\psi c_\phi & c_\psi s_\theta c_\phi + s_\psi s_\phi \\ s_\psi c_\theta & s_\psi s_\theta s_\phi + c_\psi c_\phi & s_\psi s_\theta c_\phi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\phi c_\theta \end{bmatrix} \quad (3.18b)$$

In the case where  $\theta = \pi/2$ , and using trigonometric double angles, the following can be reduced;

$$R_I^b(\eta) = \begin{bmatrix} 0 & c_\psi s_\phi - s_\psi c_\phi & c_\psi c_\phi + s_\psi s_\phi \\ 0 & s_\psi s_\phi + c_\psi c_\phi & s_\psi c_\phi - c_\psi s_\phi \\ -1 & 0 & 0 \end{bmatrix} \Big|_{\theta=\pi/2} \quad (3.18c)$$

$$= \begin{bmatrix} 0 & s(\phi - \psi) & c(\phi - \psi) \\ 0 & c(\phi - \psi) & s(\phi - \psi) \\ -1 & 0 & 0 \end{bmatrix} \quad (3.18d)$$

$$= R_{x'}(\phi - \psi) \quad (3.18e)$$

Where the resultant in Eq:3.18e represents an  $\hat{X}'$ -axis rotation in a new intermediate frame, post a  $\pi/2$  rotation about the  $\hat{Y}$ -axis. Through trigonometric double angles a degree of freedom is lost at  $\theta = \pi/2$ , when  $\phi$  &  $\psi$  effect the same angle.

### 3.1.3 Quaternion Dynamics

An algorithm proposed in [115] suggested a solution to avoid Euler Angle singularities. The heuristic proposed involved switching between sequence 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* attitude representations in  $\mathbb{R}^4$  ([43,46,69] amongst others... most notably made popular by [114] for use in animation).

A quaternion is analogous to a rotation matrix because it represents an attitude difference between two reference frames. An  $\mathbb{R}^3$  attitude is parameterized as one rotation  $\theta$  about a single unit *Euler* axis  $\hat{u}$  (demonstrated using the Rodriguez Formula in [82]). In brief a quaternion consists of a scalar component,  $q_0$ , and complex vector component,  $\vec{q} \in \mathbb{C}^3$ , such that;

$$Q \triangleq \begin{bmatrix} q_0 \\ \vec{q} \end{bmatrix} \in \mathbb{R}^4 \quad (3.19)$$

The relationship between an Euler Angles rotation matrix  $R_I^b(\eta)$  and a quaternion attitude  $Q_b$  is given by the Rodriguez formula:

$$R_I^b(\eta) = R(Q_b) = \mathbb{I} + 2q_0[\vec{q}]_\times + 2[\vec{q}]_\times^2 \quad (3.20)$$

Where  $[.]_\times$  is the cross-product matrix defined earlier in Eq:2.5c. All quaternions, unless otherwise specified, are unit quaternions  $Q \in \mathbb{Q}_u$ . Quaternion with a unity magnitude ensure that rotational operations maintain the vector operand's magnitude. A unit quaternion is defined:

$$\|Q\| = \sqrt{q_0^2 + \vec{q}^2} = 1 \quad (3.21)$$

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

$$Q \otimes P = \begin{bmatrix} q_0 \\ \vec{q} \end{bmatrix} \otimes \begin{bmatrix} p_0 \\ \vec{p} \end{bmatrix} \quad (3.22a)$$

$$= \begin{bmatrix} q_0 p_0 - \vec{q} \cdot \vec{p} \\ q_0 \vec{p} + p_0 \vec{q} + \vec{q} \times \vec{p} \end{bmatrix} \quad (3.22b)$$

$$= q_0 p_0 - \vec{q} \cdot \vec{p} + p_0 \vec{q} + q_0 \vec{p} + \vec{q} \times \vec{p} \quad (3.22c)$$

Because the vector component of a quaternion is complex valued, it is natural that there exists a quaternion complex conjugate property, defined as:

$$Q^* = \begin{bmatrix} q_0 \\ -\vec{q} \end{bmatrix} \quad (3.23)$$

It then follows that the fundamental quaternion identity is:

$$Q \otimes Q^* = \mathbb{I}_{4 \times 4} \quad (3.24)$$

A right handed quaternion rotation applied to a vector  $\vec{v} \in \mathbb{R}^3$  involves multiplication by two unit quaternions.

$$\begin{bmatrix} 0 \\ \vec{v}' \end{bmatrix} = Q \otimes \begin{bmatrix} 0 \\ \vec{v} \end{bmatrix} \otimes Q^* \quad (3.25)$$

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

$$\vec{v}' = Q \otimes (\vec{v}) \otimes Q^* \quad (3.26)$$

In the case of rigid body attitude parameterization with quaternions,  $Q_b$  is the quaternion which represents the difference between  $\mathcal{F}^b$  and  $\mathcal{F}^I$ . A quaternion operator is equivalent to a rotation matrix operation, for some vector  $\vec{v}_I \in \mathcal{F}^I$ ;

$$\vec{v}_b = R_I^b(\eta) \vec{v}_I \xrightarrow{Q} Q_b \otimes (\vec{v}_I) \otimes Q_b^* \quad (3.27)$$

Since quaternions are non-commutative, the construction of a body quaternion  $Q_b$  from an Euler angle set  $\vec{\eta}$  is sequence dependent. Euler angles, despite being singular, are conceptually simple terms for describing a body's orientation. A Z-Y-X sequenced body quaternion,  $Q_b$ , can be constructed from Euler angles as:

$$Q_b = Q_z \otimes Q_y \otimes Q_x = \begin{bmatrix} \cos(\psi/2) \\ 0 \\ 0 \\ \sin(\psi/2) \end{bmatrix} \otimes \begin{bmatrix} \cos(\theta/2) \\ 0 \\ \sin(\theta/2) \\ 0 \end{bmatrix} \otimes \begin{bmatrix} \cos(\phi/2) \\ \sin(\phi/2) \\ 0 \\ 0 \end{bmatrix} \quad (3.28)$$

A quaternion time derivative, defined in [43], with  $Q_\omega$  being a quaternion with a vector component equal to angular velocity  $\vec{\omega}_{b/I}$  and a zero scalar component, is:

$$\frac{d}{dt} Q_b = \frac{1}{2} Q_b \otimes Q_\omega = \begin{bmatrix} -\frac{1}{2} \vec{q}^T \vec{\omega}_b \\ \frac{1}{2} ([\vec{q}]_\times + q_0 \mathbb{I}) \vec{\omega}_b \end{bmatrix} \quad (3.29)$$

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 inertial frame with respect to the body frame or inversely so. The first quaternion time derivative replaces angular velocity rate differentials Eq:3.9a and Eq:3.9c;

$$\dot{Q} = R_b^I(-\eta) \vec{v} \xrightarrow{Q} Q_b(-\eta) \otimes \vec{v} \otimes Q_b(-\eta)^* = Q_b^* \otimes \vec{v} \otimes Q_b \in \mathcal{F}^I \quad (3.30a)$$

$$\dot{\eta} = \Phi(\eta) \vec{\omega}_b \in \mathcal{F}^{v2,v1,I} \xrightarrow{Q} \dot{Q}_b = \frac{1}{2} Q_b \otimes Q_\omega \in \mathcal{F}^I \quad (3.30b)$$

Second order time derivatives for quaternion acceleration aren't as useful or concise as their higher order velocity counterparts. The second order derivative is provided here however it's only relevant to quaternion backstepping later in the control chapter. If possible, quaternion accelerations are mostly avoided due to their complexity of their evaluation;

$$\ddot{Q}(\dot{Q}, Q, t) = \dot{Q} \otimes Q^* \otimes \dot{Q} + \frac{1}{2} Q \otimes [J_b^{-1}(\tau - 4(Q^* \otimes \dot{Q}) \times (J_b(Q^* \otimes \dot{Q})))] \quad (3.31)$$

An Euler angle attitude error state used for control plants is defined as the subtracted error between desired and existing attitude orientations  $\vec{\eta}_{ad}$  and  $\vec{\eta}_{ab}$  respectively. Where  $\vec{\eta}_d$  is some attitude produced from a trajectory generation loop.

$$\vec{\eta}_e = \vec{\eta}_d - \vec{\eta}_b \quad (3.32)$$

In contrast with Eq:3.32, a quaternion attitude error is a multiplicative term defined as the difference between two quaternions  $Q_d$  and  $Q_b$ ;

$$Q_e = Q_d^* \otimes Q_b \quad (3.33)$$

Quaternion attitude control and its stability goals are expanded upon later in Sec:4.5.1.

### 3.1.4 Quaternion Unwinding

Although quaternions are indeed better than Euler angles, lacking the associated singularity, they do contain one caveat. Because a quaternion  $Q = [q_0 \vec{q}]^T$  represents a body's attitude in  $\mathbb{R}^3$  using  $\mathbb{R}^4$  there is an infinite coverage of attitude states, [82]. Each unit quaternion, stemming from Euler-Rodriguez theorem, represents a single Euler-axis rotation of  $\theta$  about a unit axis  $\hat{u}$  such that:

$$Q = \begin{bmatrix} q_0 \\ \vec{q} \end{bmatrix} = \begin{bmatrix} \cos(\theta/2) \\ \sin(\theta/2)\hat{u} \end{bmatrix} \quad (3.34)$$

That rotation is applied with a quaternion operator, Eq:3.26. For every attitude state in 3-Dimensions there exist two unique quaternions which correspond to the same orientation, differing by their rotational direction about the Euler-axis. The rotation angle  $\theta$  about the Euler-axis  $\hat{u}$  is reciprocal in that  $\theta = \theta + 2k\pi$ ,  $k \in \mathbb{N}$ , there are then two definitions for  $Q_b$ ;

$$Q_b = \begin{bmatrix} q_0 \\ \vec{q} \end{bmatrix} = \begin{bmatrix} \cos(\theta/2) \\ \sin(\theta/2)\hat{u} \end{bmatrix} \quad (3.35a)$$

$$Q_b = \begin{bmatrix} \cos(\pi - \theta/2) \\ \sin(\pi - \theta/2)\hat{u} \end{bmatrix} = \begin{bmatrix} -\cos(\theta/2) \\ \sin(\theta/2)\hat{u} \end{bmatrix} \quad (3.35b)$$

$$\eta \in \mathbb{R}^3 \iff \begin{bmatrix} \pm q_0 \\ \vec{q} \end{bmatrix} \in \mathbb{R}^4 \quad (3.35c)$$

Eq:3.35c asserts that for each attitude in  $\mathbb{R}^3$  there are *two* corresponding quaternions in  $\mathbb{R}^4$ ;  $[\pm q_0 \vec{q}]^T$ . A consequence of this is two possible error state trajectories exist for every attitude difference. Both a clockwise,  $\theta$ , and an anticlockwise,  $2\pi - \theta$ , rotations which point to the same quaternion attitude error state. This could lead to an erroneous and unnecessary "unwinding" of a complete counter revolution. As such for attitude controllers the requirement is that for positive and negative scalars the control input is consistent:

$$\vec{\nu}_d = h([q_0 \vec{q}]^T, t) = h([-q_0 \vec{q}]^T, t) \quad (3.36)$$

Or more simply that  $Q_e = [|q_0| \vec{q}]^T$ . The most simple solution adhering to that constraint, which is often used, is to neglect the quaternion scalar component altogether. Using a reduced error state, only the quaternion error vector as an argument for the control law;  $h(\vec{q}_e, t)$ . Such a solution is an oversimplification and would only ever be locally stable.

An alternative is to use only the absolute quaternion scalar, which ensures the error state represents a right-handed (clockwise) rotation and not necessarily the shortest path. If the resolution of trajectory co-ordinates generated is sufficiently fine the control plant won't encounter a problem. One proposal presented in [23] suggested using a *signum* operator to design the controller coefficient sign for the virtual angular velocity,  $\vec{\omega}_d$  control plant input.

$$\vec{\omega}_d = \frac{2}{\Gamma_1} sgn(q_0) \vec{q} \quad (3.37a)$$

$$sgn(q_0) = \begin{cases} 1 & q_0 \geq 0 \\ -1 & q_0 < 0 \end{cases} \quad (3.37b)$$

Eq:3.37 was shown to be asymptotically stable but only locally in the case where the Euler-axis angle is constrained;  $\theta \leq \pm\pi$ . That control law would still need the control torques to be calculated from that angular velocity  $\vec{\omega}_d$  setpoint.

In [10], the authors used a backstepping controller with a trajectory using the absolute quaternion scalar. The resultant was a global asymptotically stable control law which tracked quaternion setpoints for a satellite's attitude. Controllers presented in Sec:4.5 all incorporate the signed quaternion scalars into the control law; hence relying on the trajectory generation to provide the desired direction of the rotation path.

## 3.2 Multibody Nonlinearities

Typically multibody dynamics are solved (and simulated) as a series of constrained torque and force responses. There are different schools of thought on the subject which each propose methodologies for stepping through the systems dynamics (*e.g* Implicit Euler integration; [64, 131]...). For the prototype under consideration, only relative rotational motion is mechanically permissible between the connected rigid bodies. Each body is considered independently whose constraint torques induced from excitation are imposed onto sequential rotational joints. Opposing those torques are *gyroscopic* and *inertial* Newtonian responses. Those responses are now quantified and introduced to the dynamic model derived in Sec:3.1.1. The design has no translational degrees of freedom between each body and is treated as rigid.

A distinction must be made between torque responses here and those previously in Eq:3.9d. The torque term(s) previously regard the entire body as rigid, the induced responses are then as a result of the entire multibody's collective motion. The following is with regards to the internal relative movements between each body. The analysis here aims to approximate the highly non-linear dynamic system as clear responses induced from servo rotations of inner and middle ring bodies,  $\Delta\lambda_i$  and  $\Delta\alpha_i$  respectively.

Alternatively the net dynamics could be derived from a Lagrangian for the *entire* 13 body dynamic system. Where those connected bodies are; four rotor/propeller bodies (Fig:2.10a), four inner ring bodies (Fig:2.10b), four middle ring bodies (Fig:2.11a) and finally the frame structure (Fig:2.13) each with 6 degrees of freedom. The constraints on the assembly's joints would eventually reduce the degrees of freedom and simplify solving for net responses. However such an approach is not used here as complete 6-DOF equations for every rigid body first need to be quantified before they can be reduced, making the kinematic derivation dramatically more cumbersome.

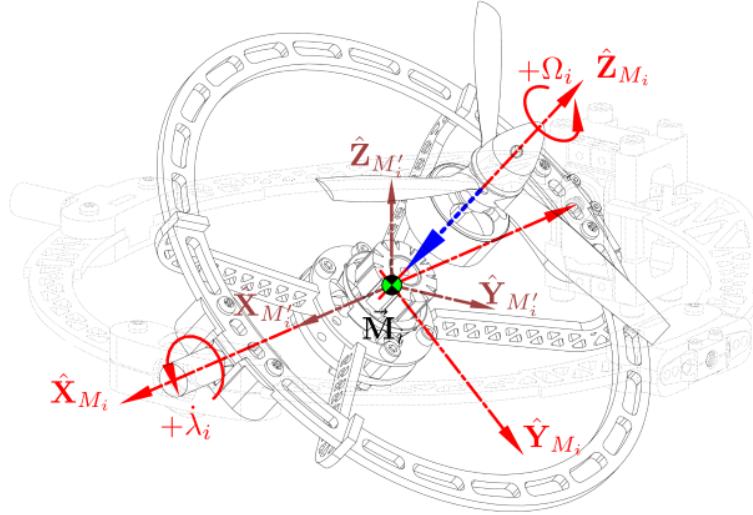
*Rotation matrices are used in the following derivations owing to the fact that induced torque responses are dependent on transformed rotational inertias. Quaternions, as mentioned in Sec:2.3, are ill-suited to inertia transformations. Moreover each actuating servo has a range of  $\pm\pi/2$  so rotation matrix singularities won't be a concern...*

### 3.2.1 Relative Rotational Gyroscopic & Inertial Torques

The torque responses induced by relative rotations, the only permissible intra-body movement, are transferred from the interacting bodies as a result of Newton's second law applied to rotational motion. For each of the motor modules' pitching or rolling motion the respective servo motors apply some torque to invoke that rotation. Opposed to that angular acceleration are both inertial and gyroscopic torque responses of the body being rotated.

Each of the four motor modules are symmetrical and as such the induced torque response characteristics for one module can be extrapolated through a  $\hat{Z}_b$  reference frame rotation. Each motor module is positioned relative to the body frame center of motion  $\vec{\mathbf{O}}_b$  as in Fig:2.9. Because each relative rotation from the actuator set,  $u \in \mathbb{U}$ , is actuated separately and upon a different body, their responses are calculated independently too.

Drawing again from Lagrangian theory and considering only the angular kinetic for the inner ring assembly  $\mathcal{F}^{M_i}$ . There is no relative translational motion between each body and so there is no translational kinetic energy contribution. The translational kinetic energy for each module is an extension of body's kinetic energy in Eq:3.7a and independent of any actuator positions. The motor module's translational motion is accounted for in Eq:3.13a.



**Figure 3.2:** Exploded inertial bodies for  $\vec{\tau}_\lambda$

Deriving dynamic responses for changes in  $\lambda_i$  on the inner ring frame  $\mathcal{F}^{M_i}$  relative to the middle ring frame  $\mathcal{F}^{M'_i}$  requires a relative path co-ordinate to be defined. Seeing that the only path variable between the two frames is  $\lambda_i$ ; the path co-ordinates  $\vec{u}(t) = [\lambda_i \ 0 \ 0]^T$  for the servo position  $\lambda$  are used to form the Lagrangian for the inner ring's motion in the middle ring frame;  $\mathcal{L}_{M'_i} \in \mathcal{F}^{M'_i}$ . The inner ring assembly consists of two separate bodies (Fig:3.2) with relative rotational motion and independent kinetic energies; the rotor assembly with an inertia  $J_r$  (defined earlier in Eq:2.15) and the inner ring (*sans* rotor assembly) which has an inertia  $J_{ir}$ :

$$J_{ir} = J_n - J_r = J_{M_i} - J_r \quad \in \mathcal{F}^{M_i} \quad (3.38)$$

Where  $J_{M_i}$  is the net inertia for the inner ring assembly from Eq:2.17. The rotor assembly has an angular velocity  $\vec{\omega}_{r/M'_i}$  relative to the middle ring frame  $\mathcal{F}^{M'_i}$ :

$$\vec{\omega}_{r/M'_i} = R_x(-\lambda)\vec{\Omega}_i + \frac{d}{dt}(\vec{\lambda}_i) \quad \in \mathcal{F}^{M'_i} \quad (3.39a)$$

$$= R_x(-\lambda)\vec{\Omega}_i + \dot{\vec{\lambda}}_i \quad (3.39b)$$

With the propeller's angular velocity vector  $\vec{\Omega}_i = [0 \ 0 \ \Omega_i]^T$  in  $[\text{rad.s}^{-1}]$ , not in revolutions per second, and the servo position  $\vec{\lambda}_i = [\lambda_i \ 0 \ 0]^T$ . Similarly the inner ring's angular velocity  $\vec{\omega}_{M_i/M'_i}$  relative to  $\mathcal{F}^{M'_i}$  is only as a result of  $\Delta\lambda_i$ :

$$\vec{\omega}_{M_i/M'_i} = \frac{d}{dt}(\vec{\lambda}_i) = \dot{\vec{\lambda}}_i \quad \in \mathcal{F}^{M'_i} \quad (3.40)$$

The Lagrangian for the inner ring's energy  $\mathcal{L}_{M'_i}$ , relative to the middle ring frame  $\mathcal{F}^{M'_i}$ , consists purely of rotational kinetic energy from Eq:3.39 and Eq:3.40. Both inertias for the rotor and inner ring bodies,  $J_r$  and  $J_{ir}$  respectively, are transformed onto the middle ring frame  $\mathcal{F}^{M'_i}$ :

$$\mathcal{L}_{M'_i} = \frac{1}{2}\vec{\omega}_{r/M'_i}^T(J'_r)\vec{\omega}_{r/M'_i} + \frac{1}{2}\vec{\omega}_{M_i/M'_i}^T(J'_{ir})\vec{\omega}_{M_i/M'_i} \quad (3.41a)$$

$$= \frac{1}{2}\left(R_x(-\lambda)\vec{\Omega}_i + \dot{\vec{\lambda}}_i\right)^T\left(R_x(\lambda)(J_r)R_x^{-1}(\lambda)\right)\left(R_x(-\lambda)\vec{\Omega}_i + \dot{\vec{\lambda}}_i\right) + \frac{1}{2}\dot{\vec{\lambda}}_i^T\left(R_x(\lambda)(J_{ir})R_x^{-1}(\lambda)\right)\dot{\vec{\lambda}}_i \quad (3.41b)$$

Noting that the inner ring's inertia  $J_{ir}$  is now a separate body from the rotor  $J_r$ . Recalling the Euler-Lagrange formulation from Eq:3.3 with path co-ordinates  $\vec{u}(t)$  for the inner ring frame,  $\mathcal{F}^{M_i}$ , relative to the middle ring frame,  $\mathcal{F}^{M'_i}$ :

$$\frac{d}{dt}\left(\frac{\delta\mathcal{L}_{M'_i}}{\delta\dot{\vec{u}}}\right) - \frac{\delta\mathcal{L}_{M'_i}}{\delta\vec{u}} = \vec{U} = \vec{\tau}_\lambda \quad \in \mathcal{F}^{M'_i} \quad (3.42)$$

From [9] the partial derivative of a rotation matrix  $R_x(\lambda)$  (and by extension the *transformation matrix*  $R_x(-\lambda)$ ) is linearized using Taylor series expansion. It follows that for some small perturbation  $\delta\theta$  away from the nominal  $\bar{\theta}$ , a generalized rotation matrix about an axis  $\hat{u}$  by that angle  $\theta$  becomes a first order approximation:

$$R_u(\bar{\theta} + \delta\theta) \approx \underbrace{(1 - [\Psi_u(\bar{\theta})\delta\theta]_{\times})}_{\text{infinitesimal rot}} R_u(\bar{\theta}) \quad (3.43)$$

Where  $\Psi_u(\theta)$  is a generalized Euler matrix derivative analogous to Eq:2.9e. The consequence of Eq:3.43 is that, for the transformed rotational inertias in Eq:3.41a, both  $R_x(\lambda)^T(J_r)R_x(\lambda)$  and  $R_x(\lambda)^T(J_{ir})R_x(\lambda)$  can be approximated using their instantaneous transformation with no partial rate derivatives.

$$R_x(\lambda)(J_r)R_x^{-1}(\lambda) = J'_r \rightarrow \frac{\delta}{\delta \vec{u}} J'_r = \frac{\delta}{\delta \lambda_i} J'_r \approx 0 \quad (3.44a)$$

Similarly for the inner ring without the rotor body's contribution:

$$R_x(\lambda)(J_{ir})R_x^{-1}(\lambda) = J'_{ir} \rightarrow \frac{\delta}{\delta \vec{u}} J'_{ir} = \frac{\delta}{\delta \lambda_i} J'_{ir} \approx 0 \quad (3.44b)$$

It follows that the partial derivatives of the Lagrangian Eq:3.41a with respect to  $\vec{u}$  are negligible;  $\delta\mathcal{L}_{M'_i}/\delta\vec{u} = 0$ . Only the partial derivative with respect to the path rate  $\dot{\vec{u}}$  remains. That partial derivative is then:

$$\therefore \frac{\delta\mathcal{L}_{M'_i}}{\delta \dot{\vec{u}}} = (J'_r) \left( R_x(-\lambda) \vec{\Omega}_i + \dot{\vec{\lambda}}_i \right) + (J'_{ir}) \dot{\vec{\lambda}}_i \quad (3.45a)$$

Inserting the Reynolds transportation theorem, Eq:3.5 for a vector's derivative in a rotating reference frame, into Eq:3.45a yields:

$$\begin{aligned} \rightarrow \frac{d}{dt} \left( \frac{\delta\mathcal{L}_{M'_i}}{\delta \dot{\vec{u}}} \right) &= \left( (J'_r) R_x(-\lambda) \dot{\vec{\Omega}}_i + \vec{\omega}_{M_i/M'_i} \times (J'_r) R_x(-\lambda) \vec{\Omega}_i + (J'_r) \ddot{\vec{\lambda}}_i + \vec{\omega}_{M_i/M'_i} \times (J'_r) \dot{\vec{\lambda}}_i \right) \\ &\quad + \left( (J'_{ir}) \ddot{\vec{\lambda}}_i + \vec{\omega}_{M_i/M'_i} \times (J'_{ir}) \dot{\vec{\lambda}}_i \right) \end{aligned} \quad (3.45b)$$

Recombining inertial contributions with the same angular velocity ( $J'_r + J'_{ir} = J'_n$ ) and recognizing that, from Eq:3.40  $\vec{\omega}_{M_i/M'_i} = \dot{\vec{\lambda}}$ ; the generalized net torque encountered by a rotation of  $\lambda$  is:

$$\vec{\tau}_{\lambda} = \frac{d}{dt} \left( \frac{\delta\mathcal{L}_{M'_i}}{\delta \dot{\vec{u}}} \right) = (J'_r) \dot{\vec{\Omega}}_i' + \dot{\vec{\lambda}}_i \times (J'_r) \vec{\Omega}'_i + (J'_n) \ddot{\vec{\lambda}}_i + \dot{\vec{\lambda}}_i \times (J'_n) \dot{\vec{\lambda}}_i \in \mathcal{F}^{M'_i} \quad (3.46a)$$

Using  $\vec{\Omega}'_i$  and  $\dot{\vec{\Omega}}'_i$  as the respective transformed angular velocity and acceleration of the propeller in the middle ring frame;

$$\vec{\Omega}'_i = R_x(-\lambda) \vec{\Omega}_i \in \mathcal{F}^{M'_i} \quad (3.46b)$$

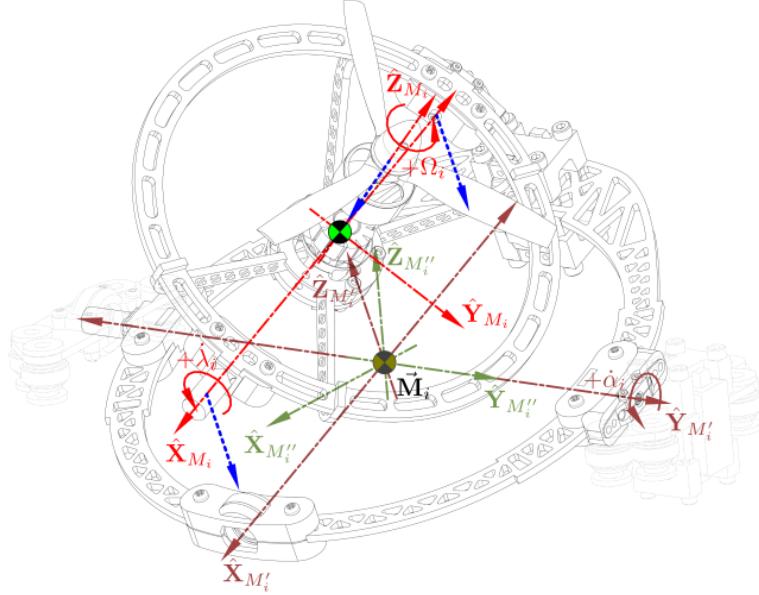
$$\dot{\vec{\Omega}}'_i = R_x(-\lambda) \dot{\vec{\Omega}}_i \in \mathcal{F}^{M'_i} \quad (3.46c)$$

The net torque response from a  $\lambda_i$  rotation, induced in the middle ring frame  $\mathcal{F}^{M'_i}$ , can be grouped into second order *inertial* and first order *gyroscopic* components;

$$\vec{\tau}_{\lambda} = \underbrace{(J'_r) \dot{\vec{\Omega}}'_i + (J'_n) \ddot{\vec{\lambda}}_i}_{\text{Inertial}} + \underbrace{\dot{\vec{\lambda}}_i \times (J'_r) \vec{\Omega}'_i + \dot{\vec{\lambda}}_i \times (J'_n) \dot{\vec{\lambda}}_i}_{\text{Gyroscopic}} \in \mathcal{F}^{M'_i} \quad (3.47)$$

Taking special care not to confuse  $J'_n$  and  $J_m$ . The former being the inner ring's inertia transformed into the middle ring frame  $\mathcal{F}^{M'_i}$  and the latter is the middle ring's inertia in that same reference frame.

Similarly for the middle ring frame  $\mathcal{F}^{M'_i}$  relative to the intermediary frame  $\mathcal{F}^{M''_i}$ , the only relative path variable is  $\vec{v}(t) = [0 \ \alpha_i \ 0]^T$ . The entire motor module structure consists of three separate rotating bodies each with their own relative angular velocities; the rotor assembly, inner ring structure and middle ring structure (Fig:3.3).



**Figure 3.3:** Exploded inertial bodies for  $\vec{\tau}_\alpha$

Following the same process to evaluate the middle ring's response, but with respect to the intermediary frame  $\mathcal{F}^{M''_i}$ . The rotor assembly has a transformed inertia:

$$J''_r = R_y(\alpha)(J'_r)R_y^{-1}(\alpha) = R_y(\alpha)R_x(\lambda)(J_r)R_x^{-1}(\lambda)R_y^{-1}(\alpha) \quad \in \mathcal{F}^{M''_i} \quad (3.48a)$$

The inner ring has an inertia, still *without* including the rotor assembly:

$$J''_{ir} = R_y(\alpha)(J'_{ir})R_y^{-1}(\alpha) = R_y(\alpha)R_x(\lambda)(J_{ir})R_x^{-1}(\lambda)R_y^{-1}(\alpha) \quad \in \mathcal{F}^{M''_i} \quad (3.48b)$$

Lastly the middle ring assembly's inertia from Eq:2.20a, with neither the rotor's nor the inner ring's contributions:

$$J'_m = R_y(\alpha)(J_\zeta)R_y^{-1}(\alpha) - R_y(\alpha)R_x(\lambda)(J_n)R_x^{-1}(\lambda)R_y^{-1}(\alpha) \quad (3.48c)$$

$$= J'_\zeta - J''_n = J'_\zeta - (J''_{ir} + J''_r) \quad \in \mathcal{F}^{M''_i} \quad (3.48d)$$

Where  $J_\zeta$  is the collective motor module inertia from Eq:2.20b. Each body then has its own relative angular velocity with respect to the frame  $\mathcal{F}^{M''_i}$ . For the rotor;  $\vec{\omega}_{r/M''_i}$  is the relative angular velocity of that assembly:

$$\vec{\omega}_{r/M''_i} = R_y(-\alpha)R_x(-\lambda)\vec{\Omega}_i + \frac{d}{dt}(R_y(-\alpha)\vec{\lambda}_i) + \frac{d}{dt}(\vec{\alpha}_i) \quad \in \mathcal{F}^{M''_i} \quad (3.49a)$$

$$= R_y(-\alpha)R_x(-\lambda)\vec{\Omega}_i + R_y(-\alpha)\dot{\vec{\lambda}}_i + \dot{\vec{\alpha}}_i \quad (3.49b)$$

$$\rightarrow \vec{\omega}_{r/M''_i} = \vec{\Omega}''_i + \dot{\vec{\lambda}}'_i + \dot{\vec{\alpha}}_i \quad (3.49c)$$

Next, the inner ring has an angular velocity  $\vec{\omega}_{M_i/M''_i}$  relative to the intermediate frame  $\mathcal{F}^{M''_i}$ :

$$\vec{\omega}_{M_i/M''_i} = \frac{d}{dt}(R_y(-\alpha)\vec{\lambda}_i) + \frac{d}{dt}(\vec{\alpha}_i) \quad \in \mathcal{F}^{M''_i} \quad (3.50a)$$

$$= R_y(\alpha)\dot{\vec{\lambda}}_i + \dot{\vec{\alpha}}_i = \dot{\vec{\lambda}}'_i + \dot{\vec{\alpha}}_i \quad (3.50b)$$

Finally the middle ring has an angular velocity  $\vec{\omega}_{M'_i/M''_i}$  relative to the intermediary frame:

$$\vec{\omega}_{M'_i/M''_i} = \frac{d}{dt}(\vec{\alpha}_i) = \dot{\vec{\alpha}}_i \quad \in \mathcal{F}^{M''_i} \quad (3.51)$$

Then using the relative path co-ordinate  $\vec{v}(t)$  the Lagrangian  $\mathcal{L}_{M_i''}$  can be constructed for the motor module relative to the frame  $\mathcal{F}^{M_i''}$ ;

$$\mathcal{L}_{M_i''} = \frac{1}{2} \vec{\omega}_{r/M_i''}^T (J_r'') \vec{\omega}_{r/M_i''} + \frac{1}{2} \vec{\omega}_{M_i'/M_i''}^T (J_{ir}'') \vec{\omega}_{M_i'/M_i''} + \frac{1}{2} \vec{\omega}_{M_i'/M_i''}^T (J_m') \vec{\omega}_{M_i'/M_i''} \quad (3.52)$$

Therefore the Langragian from Eq:3.52 expands to:

$$\begin{aligned} \mathcal{L}_{M_i''} = & \frac{1}{2} \left[ R_y(-\alpha) R_x(-\lambda) \vec{\Omega}_i + R_y(-\alpha) \dot{\vec{\lambda}}_i + \dot{\vec{\alpha}}_i \right]^T (J_r'') \left[ R_y(-\alpha) R_x(-\lambda) \vec{\Omega}_i + R_y(-\alpha) \dot{\vec{\lambda}}_i + \dot{\vec{\alpha}}_i \right] \\ & + \frac{1}{2} \left[ R_y(-\alpha) \dot{\vec{\lambda}}_i + \dot{\vec{\alpha}}_i \right]^T (J_{ir}'') \left[ R_y(\alpha) \dot{\vec{\lambda}}_i + \dot{\vec{\alpha}}_i \right] + \frac{1}{2} \dot{\vec{\alpha}}_i^T (J_m') \dot{\vec{\alpha}}_i \end{aligned} \quad (3.53)$$

Again, justifying the rotation matrix linearization using Eq:3.43, matrices  $J_r''$ ,  $J_{ir}''$  and  $J_m'$  are all instantaneous transformed inertias. The Euler-Lagrange formulation then simplifies, with the partial derivative  $\delta \mathcal{L}_{M_i''} / \delta \vec{v} = 0$ , to:

$$\frac{d}{dt} \left( \frac{\delta \mathcal{L}_{M_i''}}{\delta \dot{\vec{v}}} \right) - \frac{\delta \mathcal{L}_{M_i''}}{\delta \vec{v}} = \frac{d}{dt} \left( \frac{\delta \mathcal{L}_{M_i''}}{\delta \dot{\vec{v}}} \right) = \vec{V} = \vec{\tau}_\alpha \quad (3.54)$$

Finding the partial derivative of the Lagrangian  $\mathcal{L}_{M_i''}$  with respect to  $\dot{\vec{v}}$  is then:

$$\frac{\delta \mathcal{L}_{M_i''}}{\delta \dot{\vec{v}}} = (J_r'') \left[ R_y(-\alpha) R_x(-\lambda) \vec{\Omega}_i + R_y(-\alpha) \dot{\vec{\lambda}}_i + \dot{\vec{\alpha}}_i \right] + (J_{ir}'') \left[ R_y(-\alpha) \dot{\vec{\lambda}}_i + \dot{\vec{\alpha}}_i \right] + (J_m') \dot{\vec{\alpha}}_i \quad (3.55a)$$

$$\begin{aligned} \rightarrow \frac{d}{dt} \left( \frac{\delta \mathcal{L}_{M_i''}}{\delta \dot{\vec{v}}} \right) = & \left( (J_r'') \ddot{\vec{\Omega}}_i'' + \vec{\omega}_{M_i/M_i''} \times (J_r'') \vec{\Omega}_i'' + (J_r'') \ddot{\vec{\lambda}}_i' + \vec{\omega}_{M_i/M_i''} \times (J_r'') \dot{\vec{\lambda}}_i' + (J_r'') \ddot{\vec{\alpha}}_i \right. \\ & + \vec{\omega}_{M_i'/M_i''} \times (J_r'') \dot{\vec{\alpha}}_i \Big) + \left( (J_{ir}'') \ddot{\vec{\lambda}}_i' + \vec{\omega}_{M_i/M_i''} \times (J_{ir}'') \dot{\vec{\lambda}}_i' + (J_{ir}'') \ddot{\vec{\alpha}}_i + \vec{\omega}_{M_i'/M_i''} \times (J_{ir}'') \dot{\vec{\alpha}}_i \right) + \left( (J_m') \ddot{\vec{\alpha}}_i \right. \\ & \left. + \vec{\omega}_{M_i'/M_i''} \times (J_m') \dot{\vec{\alpha}}_i \right) \end{aligned} \quad (3.55b)$$

Which is an ominous and seemingly complicated result to try expand and make sense of. However it can be simplified; recognizing that Eq:3.55b contains kinetic energies already introduced in Eq:3.47, but transformed to the frame  $\mathcal{F}^{M_i''}$ . After some mathematics, Eq:3.55b can be simplified with responses pertinent to  $\Delta \alpha_i$  and then the generalized force transformed response  $R_y(-\alpha) \vec{\tau}_\lambda$ :

$$\begin{aligned} \vec{V} = & \ddot{\vec{\alpha}}_i (J_r'') + \dot{\vec{\alpha}}_i \times (J_r'') [R_y(-\alpha) R_x(-\lambda) \vec{\Omega}_i] + \dot{\vec{\alpha}}_i \times (J_r'') [R_y(-\alpha) \vec{\lambda}] + \dot{\vec{\alpha}}_i \times (J_r'') \dot{\vec{\alpha}}_i + \ddot{\vec{\alpha}}_i (J_{ir}'') \\ & + \dot{\vec{\alpha}}_i \times (J_{ir}'') [R_y(-\alpha) \dot{\vec{\lambda}}_i] + \dot{\vec{\alpha}}_i \times (J_{ir}'') \dot{\vec{\alpha}}_i + (J_m') \ddot{\vec{\alpha}}_i + \dot{\vec{\alpha}}_i \times (J_m') \dot{\vec{\alpha}}_i + R_y(-\alpha) \vec{\tau}_\lambda \end{aligned} \quad (3.55c)$$

Which can be further simplified by introducing angular velocities;  $\vec{\omega}_{r/M_i''}$  of the rotor relative to the intermediary frame from Eq:3.49,  $\vec{\omega}_{M_i/M_i''}$  of the inner ring relative to that intermediary frame  $\mathcal{F}^{M_i''}$  from Eq:3.50 and finally  $\vec{\omega}_{M_i'/M_i''}$  of the middle ring relative to the fixed intermediate frame from Eq:3.51.

$$\begin{aligned} = & (J_r'') \ddot{\vec{\alpha}}_i + \dot{\vec{\alpha}}_i \times (J_r'') \vec{\omega}_{r/M_i''} + \ddot{\vec{\alpha}}_i (J_{ir}'') + \dot{\vec{\alpha}}_i \times (J_{ir}'') \vec{\omega}_{M_i/M_i''} + (J_m') \ddot{\vec{\alpha}}_i + \dot{\vec{\alpha}}_i \times (J_m') \vec{\omega}_{M_i'/M_i''} \\ & + R_y(-\alpha) \vec{\tau}_\lambda \in \mathcal{F}^{M_i''} \end{aligned} \quad (3.55d)$$

Isolating the torque response from  $\Delta \alpha$ , and again grouping inertial bodies with shared angular velocities together. The first and second order *gyroscopic* and *inertial* responses are then:

$$\vec{\tau}_\alpha(\lambda_i) = \underbrace{(J_\zeta'(\lambda)) \ddot{\vec{\alpha}}_i}_{\text{Inertial}} + \underbrace{\dot{\vec{\alpha}}_i \times (J_r'') \vec{\omega}_{r/M_i''} + \dot{\vec{\alpha}}_i \times (J_{ir}'') \vec{\omega}_{M_i/M_i''} + \dot{\vec{\alpha}}_i \times (J_m') \vec{\omega}_{M_i'/M_i''}}_{\text{Gyroscopic}} \in \mathcal{F}^{M_i''} \quad (3.56)$$

Both Eq:3.47 and Eq:3.56 are obvious results stemming from a clear second order inertial response to accelerations  $\ddot{\lambda}_i$  and  $\ddot{\alpha}_i$  respectively. The remaining gyroscopic components of those torque responses results from the cross product of those angular velocities  $\dot{\lambda}_i$  and  $\dot{\alpha}_i$  and the internal bodies relative angular velocities. Careful inspection of the multi-body system could have produced the same equation(s). The assumption in Eq:3.43 that rotated inertias can be linearised is shown to hold true in Appendix:A.3 where simulations and physical tests corroborating the above models.

Both servo's respective induced torques,  $\vec{\tau}_\lambda$  and  $\vec{\tau}_\alpha(\lambda_i)$ , occur in sequential gimbal-like frames. The opposing negative responses to induced relative rotations effect the angular state dynamics in Eq:3.9d, and must be transformed to the common body frame.

$$\vec{\tau}_Q(u) = - \sum_{i=1}^4 R_z(-\sigma)R_y(-\alpha)\vec{\tau}_{\lambda_i}(u) - R_z(-\sigma)\vec{\tau}_{\alpha_i}(u) \in \mathcal{F}^b \quad (3.57)$$

The final non-linear response as a result of relative rotations is the entire bodies inertial and gyroscopic response to net angular velocities of the multibody system relative to the inertial frame. Within the basic 6-DOF differential equations, an inertial and gyroscopic term is included, reiterating that equation:

$$\dot{\vec{\omega}}_b = (J_b^{-1}) \left( -\vec{\omega}_b \times (J_b)\vec{\omega}_b + \vec{\tau}_{net} \right) \in \mathcal{F}^b \quad (3.58a)$$

From inspection, drawing from both Eq:3.56 and Eq:3.47, the relative rotations induce a response as a result of the net body angular velocity  $\vec{\omega}_b$  is given by:

$$\vec{\tau}_b = \sum_{i=1}^4 \vec{\omega}_b \times (J_r''')\vec{\omega}_{r/b} + \vec{\omega}_b \times (J_{ir}''')\vec{\omega}_{M_i/b} + \vec{\omega}_b \times (J_m'')\vec{\omega}_{M'_i/b} \quad (3.58b)$$

$$\rightarrow \dot{\vec{\omega}}_b = (J_b^{-1}) \left( -\vec{\omega}_b \times (J_b)\vec{\omega}_b - \vec{\tau}_b + \vec{\tau}_{net} \right) \quad (3.58c)$$

Uncertainties associated with the above inertial models or their particular values can easily be compensated for as disturbances. More specifically uncertainties of the above equations system are modelled as plant dependent state uncertainties; and can be adaptively compensated for accordingly...

### 3.3 Aerodynamics

The relationship between a propeller's rotational speed,  $\Omega_i$  in [RPS], and its perpendicular thrust vector,  $\vec{T}(\Omega_i)$ , is more complicated than the quadratic simplification taken at static conditions which most papers suggest (e.g [78, 100] etc...). The thrust produced is mostly dependent on the incident air stream flowing through the propeller's rotational plane; typically being the component of the body velocity normal to that propeller's plane. Fluid flowing *tangentially* across the propeller's plane contributes toward in-plane aerodynamic drag (and hence torque).

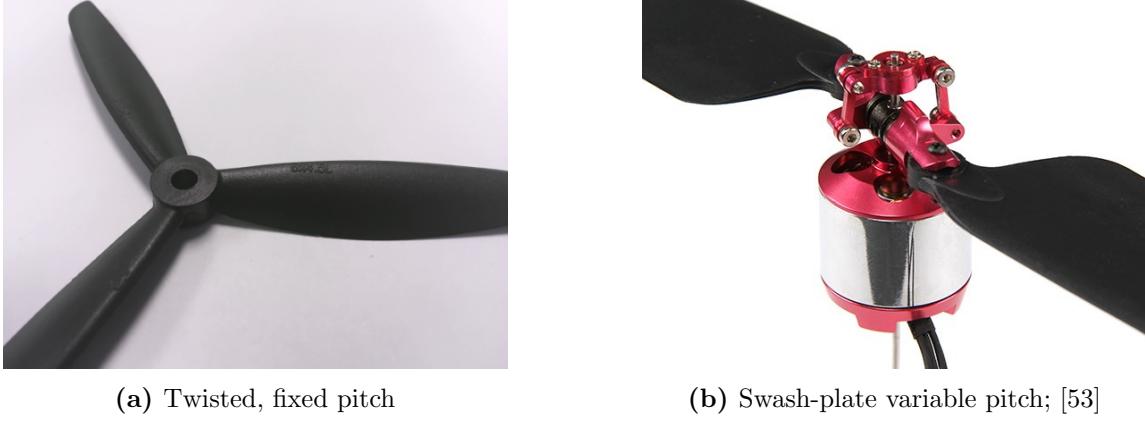
The combination of aerodynamic blade-element [92, 107] and fluid-dynamics momentum (*disc actuator*) theories equate an integral term generated across the propeller's length with the produced thrust or torque. A schedule of all aerodynamic effects encountered by a quadrotor's propellers is thoroughly detailed in both [8] and [7]. The following is a review of pertinent aerodynamic theories; vortex ring state and parasitic drag effects are not included as they will be approximately negligible given the aircraft's proposed flight envelope.

#### 3.3.1 Propeller Torque and Thrust

*A possible situation which the prototype could encounter is where an upstream propeller provides the incident fluid flow to another downstream propeller. Such a situation presents a complicated fluid*

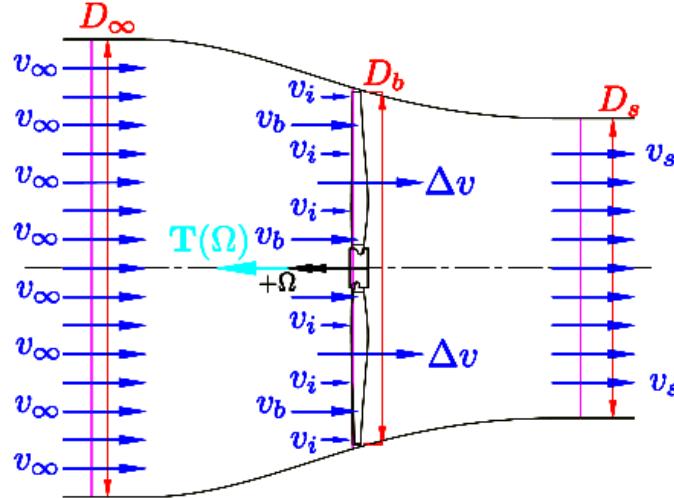
dynamics and vortex wake effect problem. Propeller overlapping effects are discussed in [122] but remain open to further research in the context of the aircraft considered here.

To expedite the system identification process some simplifications are made on the aerodynamics to construct an approximate model; specifically using coefficients in place of complete local chord and pitch based integrals. Such an assumption holds true given that twisted, fixed pitch propellers are used (Fig:3.4a) and not variable pitch swash-plate actuated propellers (Fig:3.4b).



**Figure 3.4:** Propeller types

A propeller's profile applies a perpendicular thrust force,  $T$ , onto the fluid in which it rotates. To build the following theoretical explanation propellers are first considered in terms of momentum theory; only perpendicular fluid flow through the propeller's plane is regarded. That fluid stream (Fig:3.5) has an incident upstream velocity,  $v_\infty$ , and a resultant slip velocity,  $v_s$ , downstream relative to the rotational plane.



**Figure 3.5:** Disc Actuator Propeller Planar Flow

The change of fluid flow as a result of the propeller's rotation can be given as:

$$v_s = \Delta v + v_\infty \quad (3.59)$$

Where  $\Delta v$  is the net change in fluid velocity caused by the propeller blade's rotating aerofoil profile. The propeller induces a velocity directly in front of its rotational plane,  $v_i$ , such that the net fluid flow into the plane is  $v_b = v_i + v_\infty$ . That induced inflowing fluid velocity is different to the net velocity contribution of the propeller;  $v_i \neq \Delta v$ . It is shown in [8] that, using Bernoulli's pressure theorem, the net fluid flow through the propeller's plane is:

$$v_b = \frac{1}{2}(v_s - v_\infty) = \frac{1}{2}\Delta v = \frac{1}{2}v_s|_{v_\infty=0} \quad (3.60)$$

Stemming from classical disc actuator (fluid *momentum*) theory, [102], the scalar force  $T(\Omega)$  acting on the fluid is calculated as a function of mass flow rate with respect to the change in fluid velocity (pressure differential).

$$T = (A_b v_b) \Delta v = \rho \pi R_b^2 v_b \Delta v = \rho \pi R_b^2 (v_i + v_\infty) \Delta v = \frac{1}{2} \rho \pi R_b^2 \Delta v^2 \quad (3.61)$$

Where  $R_b$  is the disc (propeller) radius in [m] for the fluid stream under consideration. The fluid density of that stream,  $\rho$ , is typically  $1.225$  [kg.m<sup>-3</sup>] at standard temperature and pressure (*stp*). The desired form of thrust generated is as a function of propeller rotational velocity  $\Omega_i$ , so Eq:3.61 is not satisfactory.

Eq:3.61 can however be solved as a function of aerodynamic propulsive power expended,  $\Delta P = T \Delta v$ . Rotational kinetic energy of a propeller and its transferred propulsive power is difficult to quantify, with compound parasitic losses deteriorating the efficiency of the propeller. Furthermore, the local fluid velocity through the propeller is not purely normal to the propeller plane.

The fluid flow induced by the propeller's rotation directly in front of its plane of rotation is not purely perpendicular but has axial and tangential induced components, termed  $a$  and  $a'$  respectively. Those induced components for the fluid velocity can be abstracted to induction factors dependent on the incident fluid velocity to the propeller's plane of rotation:

$$v_i = av_\infty \text{ in the axial direction} \quad (3.62a)$$

$$v_\theta = a'\Omega r \text{ in the tangential direction} \quad (3.62b)$$

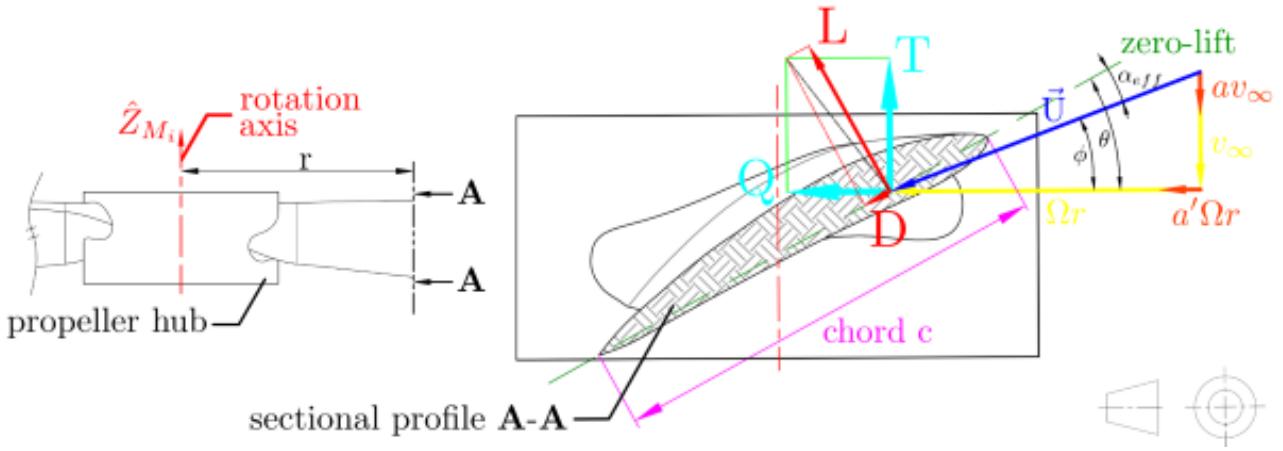
From induction factors defined Eq:3.62, the velocity components can be written as factors of upstream velocity  $v_\infty$ .

$$v_b = (1 + a)v_\infty \quad (3.63a)$$

$$v_s = (1 + 2a)v_\infty \quad (3.63b)$$

A consequence of the tangential fluid flow is that an angular momentum flow rate exists across the propeller plane. This produces a torque opposing the rotational motion about the propeller's axis, analogous but perpendicular to Eq:3.61.

$$\vec{Q} = \rho \pi R_b^3 (v_\theta - v_\infty) v_b \quad (3.64)$$



**Figure 3.6:** Blade element profile at radius  $r$

Together, Eq:3.61 and Eq:3.64 comprise propeller momentum theory but cannot be solved on their own. Blade-element theory analyses incremental aerofoil sections of width  $dr$  of the propeller profile (Fig:3.6) at some radius  $r$ . Each aerofoil element has a net local fluid velocity  $\vec{U}$  across its profile, calculated as:

$$\vec{U} = \sqrt{(v_\infty + v_i)^2 + (v_\Omega + v_\theta)^2} \quad (3.65)$$

Where each profile has a chord length  $c$  and an inclination (or *pitch*)  $\theta$  of the aerofoil *zero-lift line* relative to the horizontal. Local fluid velocities incident to the propeller profile (shown in Fig:3.6) make their own angle of attack  $\phi$  such that:

$$\phi = \theta - \alpha_{\text{effective}} \quad (3.66)$$

That local angle of attack varies with the incident fluid flow magnitude  $v_\infty$  and the induced axial velocity  $v_i$ . That trigonometric ratio between the two is given as:

$$\phi = \tan^{-1} \left( \frac{v_\infty + v_i}{v_\Omega + v_\theta} \right) = \tan^{-1} \left( \frac{v_\infty(1+a)}{\Omega r(1+a')} \right) \quad (3.67)$$

In-plane fluid flow  $\vec{U}(r, \phi)$ , for an element at radius  $r$  with a local angle of attack  $\phi$ , then contributes towards elemental lift and drag forces as a function of aerofoil's dimensionless lift,  $C_L$ , and drag,  $C_D$ , coefficients. Those coefficients are determined by the aerofoil's characteristics, but would be constant across the length of a variable pitch, hinged and untwisted hinged prop (Fig:3.4b).

$$\Delta L = \frac{1}{2} \rho \vec{U}(r, \phi)^2 c C_L \quad (3.68a)$$

$$\Delta D = \frac{1}{2} \rho \vec{U}(r, \phi)^2 c C_D \quad (3.68b)$$

With air density  $\rho$  is typically taken at stp. Lift and drag forces, when taken parallel and perpendicular to the plane of rotation, are thrust  $T$  and torque  $F_q$  forces (Fig:3.6). The in-plane force applies an aerodynamic torque  $Q$  at the propellers hub because the force  $F_q$  acts at a radius  $r$ , [55].

$$dT = \frac{1}{2} \rho \vec{U}(r, \phi)^2 c (C_L \cos(\phi) + C_D \sin(\phi)) . dr \quad (3.69a)$$

$$dF_q = \frac{1}{2} \rho \vec{U}(r, \phi)^2 c (C_L \sin(\phi) + C_D \cos(\phi)) . dr \quad (3.69b)$$

$$\rightarrow dQ = \frac{1}{2} \rho \vec{U}(r, \phi)^2 c (C_L \sin(\phi) + C_D \cos(\phi)) r . dr \quad (3.69c)$$

$$\rightarrow dP = \Omega r dF_x . dr \quad (3.69d)$$

Rotational power expended is a product of angular velocity and opposing torque; Eq:3.69d, and is used in lieu of torque or drag terms, Eq:3.69c or Eq:3.69b. Then calculating forces and power terms as per momentum theory for each element, in terms of axial and tangential induction factors:

$$dT = \rho 4\pi r^2 v_\infty (1+a) a . dr \quad (3.70a)$$

$$dP = \rho 4\pi r^2 v_\infty (1+a) \Omega r (1+a') . dr \quad (3.70b)$$

Equating momentum and element terms produces the blade-element momentum equation(s) for aerodynamic thrust and power from a propeller. Following a few assumptions; most importantly that the lift coefficient  $C_L$  is a linear function of the effective angle of attack  $\alpha_{\text{eff}}$ . The lift curve gradient,  $a_L$ , is shown in [56] for an ideally twisted blade, like the fixed pitch propellers under consideration here, to be  $2\pi$ .

$$C_L = 2\pi(\theta - \phi) \quad (3.71)$$

Assuming tangentially induced velocities  $v_\theta$  are small (or the tangential induction factor  $a' \ll 1$ ) when compared to the propeller's speed translational speed at radius  $r$ ,  $v(r) = \Omega r$ . Small angle approximations then apply to Eq:3.69a-3.69c;  $\cos(\phi + \alpha_{\text{eff}}) \approx 1$  and  $\sin(\phi + \alpha_{\text{eff}}) \approx \phi + \alpha_{\text{eff}}$ . Similarly net inflow and axial velocities are  $v_\infty + v_i \ll \Omega r$ , the following integrals are then found:

$$T = \int_{r=0}^R \frac{1}{2} a_L b c \rho (\Omega r)^2 \left[ \theta - \frac{v_\infty + v_i}{\Omega r} \right] . dr \quad (3.72a)$$

$$P = \int_{r=0}^R \frac{1}{2} a_L b c \rho (\Omega r)^3 \left[ \left( \theta - \frac{v_\infty + v_i}{\Omega r} \right) \left( \frac{v_\infty + v_i}{\Omega r} \right) + C_d \right] dr \quad (3.72b)$$

Where  $b$  is the number of propeller blades. Knowing exact pitch and chord values as a function of  $r/R$  is difficult and calculating integrals at each process step is cumbersome. Both Eq:3.72a and Eq:3.72b can be solved by equating element and momentum terms (a full expansion is given in Appendix:A.2). Often dimensionless thrust, torque and power coefficients are defined across the entire blade's length:

$$C_T(J) = \frac{T}{\rho \Omega^2 D^4} \quad (3.73a)$$

$$C_P(J) = \frac{P}{\rho \Omega^3 D^5} \quad (3.73b)$$

Where  $\Omega$  is the propeller's rotational speed in revolutions per second (*RPS*) and different to Eq:3.39 with units [ $\text{rad.s}^{-1}$ ],  $D$  is the propeller's diameter in [m]. For fixed pitch propellers the thrust and power coefficients are easily determined and remain consistent. Both Eq:3.73a and Eq:3.73b vary due to the dimensionless *advance ratio*  $J$ .

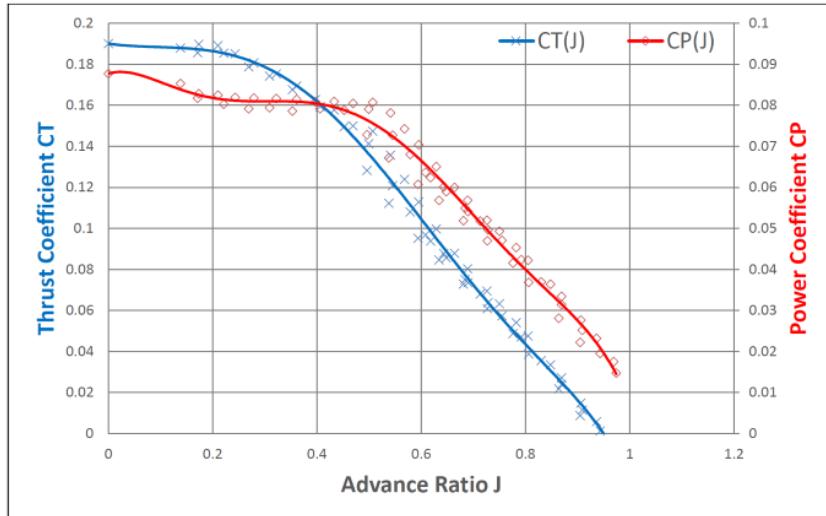
$$J = \frac{v_\infty}{\Omega R} \quad (3.74)$$

Typically the net upstream velocity  $v_\infty$  is simply the perpendicular component (projected onto the plane's normal vector  $\hat{n}$ , Eq:3.76) of the vehicle's translational velocity in the body frame,  $\vec{v}_b \perp \hat{n}$ . For the case of a zero advance ratio,  $J = 0$ , the conditions are regarded as static. Static thrust and power coefficients are nominal in their values.

Propeller databases like [21] provide comprehensive coefficient values for a range of small and medium diameter propeller types at different advance ratios. Included in the database are blade profiles, pitch and chord lengths; all the results are outcomes of the investigation [22]. The introduction of those coefficients greatly simplifies the thrust estimation process. For a typical 6X4.5 inch propeller the coefficients used were linearly interpolated from similar pitched database results to match subsequent physical test values. The static thrust and power coefficients are respectively:

$$C_{T0} = 0.191 \quad (3.75a)$$

$$C_{P0} = 0.0877 \quad (3.75b)$$



**Figure 3.7:** Power & thrust coefficients

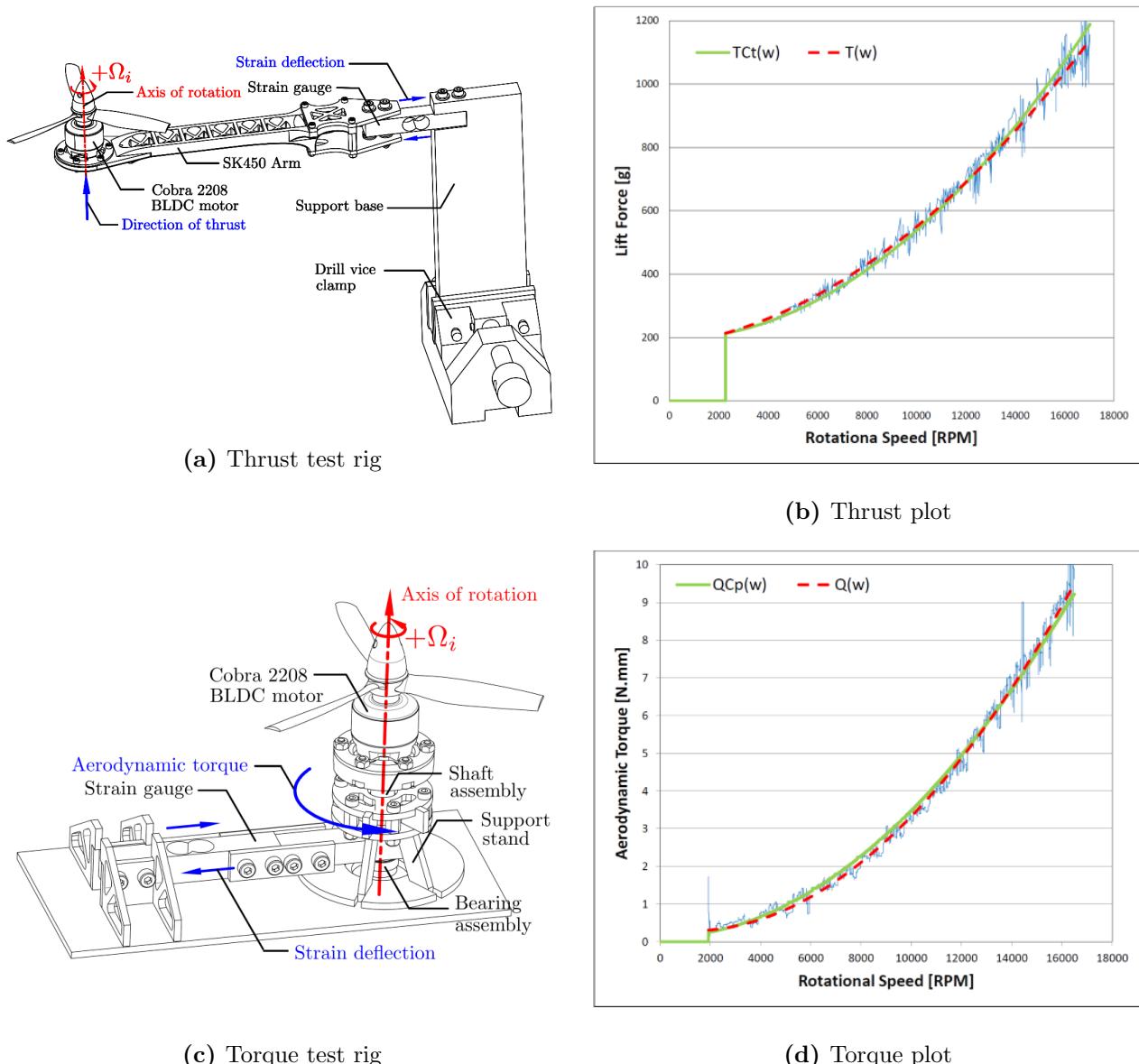
Fig:3.7 shows plots of coefficients for thrust,  $C_T$ , and power,  $C_P$ , as a function of the advance ratio  $J$ . As the incident upstream fluid velocity,  $v_\infty$ , increases, the thrust coefficient decreases. So too does the power coefficient and hence the aerodynamic torque. The thrust and power coefficients can be assumed constant for low advance ratios, or in the case considered here, translational velocities.

In Fig:3.8, the thrust and torque test rigs and the results of both static (thrust and torque) tests are shown. Measured values for each test are plotted;  $T(\Omega)$  in Fig:3.8b for thrust and  $Q(\Omega)$  in Fig:3.8d for torque. The physically tested values are fitted with parabolic trend-lines and plotted against static coefficient estimates using Eq:3.73a for thrust  $\hat{TC}_t(\Omega)$  and Eq:3.73b for calculated torque  $\hat{QC}_p(\Omega)$ . Results from Fig:3.7 are used as a lookup table and values from Eq:3.73 are calculated, induced propeller thrust and torques can be accurately modeled parabolically, the power term is cubic with respect to its rotational velocity.

Advance ratios, or rather the propeller incident fluid flow(s), are dependent on the vehicle's net transnational and angular velocity. Such that the fluid velocity's normal component to the propeller plane is given by:

$$v_\infty = (\vec{v}_b + \vec{L}_{\text{arm}} \times \vec{\omega}_b) \cdot \hat{n} \in \mathcal{F}^{M_i} \quad (3.76)$$

Where  $\vec{v}_b$  [ $\text{m.s}^{-1}$ ] is the body's transnational velocity and  $\vec{\omega}_b$  [ $\text{rad.s}^{-1}$ ] is the body's angular velocity, both transformed to the propeller's frame,  $\in \mathcal{F}^{M_i}$ . Furthermore  $\hat{n}(\lambda_i, \alpha_i)$  is the unit vector normal to the propeller's rotational plane, relative to the body velocity. Then  $J$  is calculated from Eq:3.74.



**Figure 3.8:** Static propeller tests

It's worth reiterating that the above static coefficients are indeed calculated from physical static tests. However advance ratio coefficient dependencies are linearly interpolated from the closest available matching data (APC Thin-Electric 8X6 propellers) cited from [21].

Clockwise and counterclockwise propellers and rotations were used for both thrust and torque tests. Despite the thrust and test rigs having been designed to isolate each respective response, bi-directional test results were averaged in the hopes that unwanted opposing effects were canceled. Both positive and negative rotational test results for thrust and torque measurements are included in Appendix:C.1

*Discrepancies which emerge between the model or coefficient values derived can be accounted for with lumped uncertainty disturbance term(s). Model uncertainty compensation can easily be incorporated into adaptive backstepping or  $H_\infty$  control algorithms. The deviation of the modeled thrust or torques from their true values would be simple to incorporate into a plant dependent Lyapunov candidate function; Sec:4.5.3.*

### 3.3.2 Hinged Propeller Conning & Flapping

Other non-linear effects which adversely effect a propeller's performance have all been well documented in the helicopter aerodynamic and propeller fields [20,113]. Typically such effects are more pronounced when observing hinged variable pitch are diminished. Moreover, low translational velocities suppress such responses but they're worth mentioning. Conning and flapping are the two most significant aerodynamic responses produced by a propeller. Other phenomenon like cyclic vortex ring states are deemed to not be applicable here and fall outside the scope of the investigation.

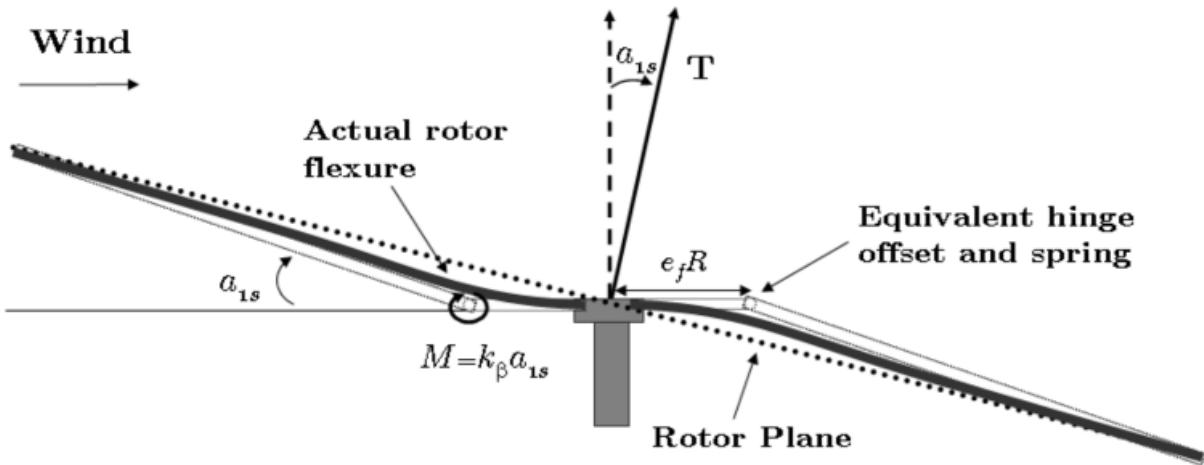


Figure 3.9: Propeller blade flapping; from [55]

In translational flight, for a propeller without shrouding or a duct, each blade encounters varying incident fluid flow. The advancing blade relative to the body's translational direction encounters a greater fluid flow than the retreating blade. The effective local angle(s) of attack for the opposing advancing and retreating propeller blades are not symmetrical from the inflows variation. Those unbalanced angles of attack produce a dissymmetry of lift across the propeller blade's surface.

Throughout each rotation the blade is forced up and down as it cycles through a varying fluid velocity field, applying a torque moment about the propeller's hub. That torque's magnitude is a function of the body's net translational velocity and the propeller material's stiffness (hence its susceptibility to deflection). The flapping pitches the effective propeller plane (*tip-path plane*), and hence the thrust vector line, away from its principle axis (Fig:3.9).

The propeller's resultant thrust vector is pitched away from its nominal perpendicular state by some deflection angle,  $\alpha_{1s}$  in Fig:3.9, toward the direction of translational movement or wind disturbance. Propeller flapping is diminished at low translational velocities with small wind disturbances relative to propeller rotational speed. As such flapping is not applicable to the feasible flight envelope envisaged for the prototype here.

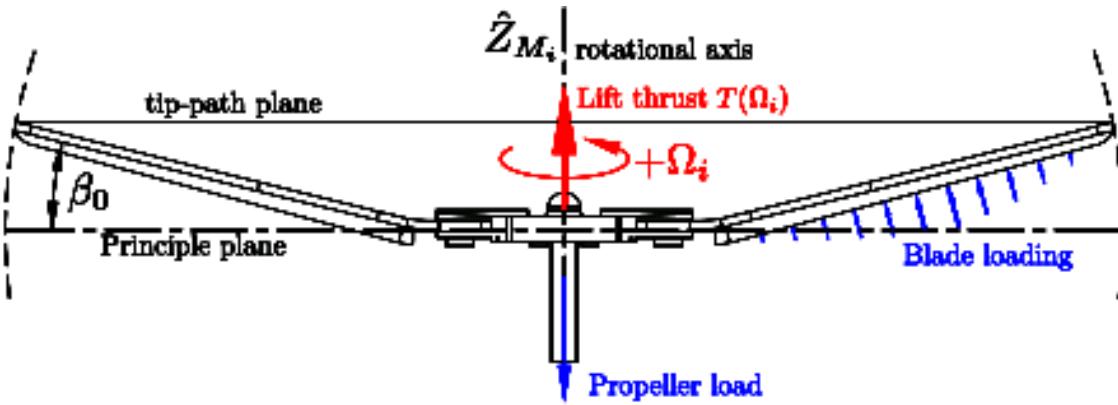


Figure 3.10: Propeller coning

Coning (illustrated in Fig.3.10) is another form of propeller deflection, which is again dependent on the blade's material stiffness. Coning causes both advancing and retreating propeller blades to both deflect upward. Distributed loading on the propeller surface from supporting a body's weight causes the upward deflection. The coning reduces the effective propeller disc's radius, adversely affecting thrust produced, Eq:3.72a. Increased loading accentuates the coning angle experienced by the propellers and as such reduces the tip-path-plane.

Both aerodynamic induced propeller deflection effects can be quantified numerically. Their derivation and resultant equations are cumbersome however. In due course both effects on the produced prototype are not significant enough to produce instability if neglected. The frame could potentially be affected in more adverse ways given certain flight conditions with higher translational velocities or incident wind and fluid flow disturbances...

### 3.3.3 Drag

For any solid body with some non-zero relative translational velocity motion within a fluid, there is a second order damping response opposing translational velocity. The net drag force,  $\vec{D}_{net}$ , although locally dependent on individual component cross-sections can be abstracted to a drag coefficient matrix representing the whole body.

$$\vec{D}_{net}(\vec{v}) = \begin{bmatrix} D_{ii} & D_{ij} & D_{ik} \\ D_{ji} & D_{jj} & D_{jk} \\ D_{ki} & D_{kj} & C_{kk} \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}^2 \quad [N], \quad \in \mathcal{F}^b \quad (3.77)$$

The drag coefficients subscripts,  $\hat{i}, \hat{j}$  &  $\hat{k}$ , are determined by the frames directional cross-section areas for each  $\hat{X}_b, \hat{Y}_b, \hat{Z}_b$  axis respectively. Given a well designed and symmetrical frame, it can be assumed the off-diagonal elements aren't of consequence and as such the drag equation can be simplified to:

$$\vec{D}_{net}(\vec{v}) \approx diag(D_{ii}, D_{jj}, D_{kk})\vec{v}^2 \quad \in \mathcal{F}^b \quad (3.78)$$

Given the second order dependency on translational velocity of the opposing drag forces, such terms are relegated to lumped disturbance terms to be adaptively compensated for in the control loop, Sec:4.5.3. The time scale separation between velocity and wind drag effects with the control loop allow for such a relegation. Analogous drag-like effects opposing angular rotation rates exist but, for the intents and purposes of most practical flight envelopes, can be disregarded.

In simulation; if the plant has sufficient disturbance rejection then the drag term in Eq:3.77 would be easily accounted for an adaptive backstepping algorithm. Furthermore it is possible to physically test for the drag coefficients to attain a higher certainty model but, given the flight conditions for this research, such effects will be small if not negligible. As such those tests are outside the scope of investigation here...

### 3.4 Consolidated Model

Reiterating the different aspects detailed above and consolidating the state equations from Eq:3.9a-3.9d. Then lifting the attitude states to  $\mathbb{R}^4$  space with the use of quaternions. Also introducing the non-linear inertial and gyroscopic responses to induced perturbations,  $\vec{\tau}_\lambda$  and  $\vec{\tau}_\alpha$  from Eq:3.47 and Eq:3.56 respectively, with non-linear inertial matrix terms  $J_b(u)$  from Section:2.3. Finally replacing net virtual plant inputs,  $\vec{\tau}_\mu$  and  $\vec{F}_\mu$ , with higher fidelity thrust models. The exact effectiveness actuator relationships are explored in Section:4.3. The following are state differential equations then used for subsequent control system development and simulation...

$$\dot{\mathcal{E}} = Q_b^* \otimes \vec{v}_b \otimes Q_b \in \mathcal{F}^I \quad (3.79a)$$

$$\dot{\vec{v}}_b = m^{-1} (-\vec{\omega}_b \times m\vec{v}_b + Q_b \otimes m\vec{G}_I \otimes Q_b^* - \vec{D}_{net}(\vec{v}_b) + \vec{F}_\mu(u)) \in \mathcal{F}^b \quad (3.79b)$$

$$\dot{Q}_b = \frac{1}{2} Q_b \otimes \vec{\omega}_b \in \mathcal{F}^I \quad (3.79c)$$

$$\dot{\vec{\omega}}_b = \mathbb{I}_b(u)^{-1} (-\vec{\omega}_b \times \mathbb{I}_b(u)\vec{\omega}_b - \vec{\tau}_Q(u) + \vec{\tau}_g(u) + \sum_{i=1}^4 \vec{Q}(\Omega, \lambda, \alpha) + \mu \vec{\tau}(u)) \in \mathcal{F}^b \quad (3.79d)$$

$$u = [\Omega_1^+, \lambda_1, \alpha_1, \dots, \Omega_4^-, \lambda_4, \alpha_4] \in \mathbb{U} \in \mathbb{R}^{12} \quad (3.79e)$$

With net force and torque control plant inputs,  $\vec{F}_\mu$  and  $\vec{\tau}_\mu$  respectively. Both are later abstracted to virtual control inputs next in Ch:4, motor number subscripts  $i \in [1 : 4]$  are implied in the following.

$$\vec{F}_\mu(u) = \sum_{i=1}^4 \vec{T}(\Omega, \lambda, \alpha) = \sum_{i=1}^4 Q_{M_i}^* \otimes T(\Omega) \otimes Q_{M_i} \in \mathcal{F}^b \quad (3.80a)$$

$$\vec{\tau}_\mu(u) = \sum_{i=1}^4 \vec{l} \times \vec{T}(\Omega, \lambda, \alpha) = \sum_{i=1}^4 \vec{l} \times (Q_{M_i}^* \otimes T(\Omega) \otimes Q_{M_i}) \in \mathcal{F}^b \quad (3.80b)$$

Scalar thrust  $T(\Omega)$  is a function of the propeller's rotational velocity whereas  $\vec{T}(\Omega, \lambda, \alpha)$  is a vector in  $\mathbb{R}^3$ . The thrust vector is redirected similarly to Eq:2.13a and transformed to the frame  $\mathcal{F}^b$ . Equivalently  $Q(\Omega)$  is the scalar aerodynamic torque in  $\mathcal{F}^{M_i}$  about each motor's rotor  $\hat{Z}$ -axis,  $\vec{Q}(\Omega, \lambda, \alpha)$  is the torque vector counterpart in  $\mathcal{F}^b$ . Both thrust and aerodynamic propeller torque terms are calculated from their respective coefficients (plotted in Fig:3.7):

$$T(\Omega) = C_T(J) \rho \Omega^2 D^4 \quad [\text{N}] \quad (3.81a)$$

$$Q(\Omega) = C_P(J) \rho \Omega^3 D^5 (1/R\Omega) \quad [\text{Nm}] \quad (3.81b)$$

Noting that  $\Omega$  for aerodynamic calculations in Eq:3.81a and Eq:3.81b has units [RPS]. Inertial torque responses from actuator input rates from Eq:3.57 are introduced as feedback compensation terms;

$$\tau_Q(u) = \sum_{i=1}^4 -Q_{M_i} \otimes \tau_{\lambda_i}(u) \otimes Q_{M_i}^* - Q_{M'_i} \otimes \tau_{\alpha_i}(u) \otimes Q_{M'_i}^* \in \mathcal{F}^b \quad (3.82)$$

Then including variable gravitational torque arm from Eq:3.83, dependent on net actuator positions  $u$ :

$$\vec{\tau}_g(u) = \Delta C.G \times \vec{G}_b \in \mathcal{F}^b \quad (3.83)$$

And finally, the entire vehicles instantaneous inertia, taken from Eq:2.24, is given as:

$$J_b(u) = J_y' + \sum_{i=1}^4 J_n + \sum_{i=1}^4 J_m \quad u \in \mathbb{U} \quad (3.84)$$

It is possible to bundle both attitude states (either euler angles  $\vec{\eta}$  or quaternions  $Q_b$ ) together with the linear translational position  $\vec{\mathcal{E}}$  into a single state  $\vec{x}$ . Which then has its own combined control law. This could potentially exploit the cross-product coupling terms between angular and linear displacements for control benefits.

# Chapter 4

## Controller Development

### 4.1 Stability

Stables

### 4.2 Control Loop

The control problem for this dissertation is, as outlined in Chater:1; to achieve dynamic (*attitude*) set point tracking on a quadrotor by solving the problem of its inherent underactuation. For the purposes of the subsequent controller development, the plant is described in the following non-linear state space form:

$$\dot{\mathbf{x}} = f(\mathbf{x}, t) + g(\mathbf{x}, \vec{\nu}, t) \quad (4.1a)$$

$$y = c(\mathbf{x}, t) + d(\mathbf{x}, \vec{\nu}, t) \quad (4.1b)$$

Where the plant dynamics are governed by  $f(\mathbf{x}, t)$  and the plant's input response by  $g(\mathbf{x}, \vec{\nu}, t)$ , for a given control input  $\vec{\nu}$ . The latter is not necessarily a function based relationship and could take the multiplicative form;  $g(\mathbf{x}, t)\vec{\nu}$ . The objective for setpoint tracking is for the output to track the state; namely  $y = c(\mathbf{x}, t) = \mathbf{x}$ . As such, the control problem is to design a stabilizing control law for an error state  $\mathbf{x}_e$ <sup>1</sup>:

$$\vec{\nu}_d = h(\mathbf{x}_e, t) \quad (4.2)$$

Such that the control plant is globally asymptotically stable or that  $\lim_{t \rightarrow \infty} \mathbf{x}_e = 0$ . It is possible to combine attitude and position states into a single common trajectory state such that:

$$\mathbf{x} = \begin{bmatrix} \vec{\mathcal{E}} \\ Q_b \end{bmatrix} \quad (4.3)$$

The body's trajectory is then fully described by  $\mathbf{x}(t)$ . Separate control laws are developed for attitude and position tracking and hence those states aren't combined. However for the purposes of discussing the control plant, a single major loop is considered. The designed control input,  $\vec{\nu}_d$ , is then implemented by actuator suite  $u \in \mathbb{U}$  through its effectiveness function:

$$\nu_c = B(\mathbf{x}, u, t) \quad (4.4)$$

The exact relationship of the virtual control input and commanded input,  $\nu_c \rightarrow \nu_d$ , is governed by the allocation algorithm. That allocation function,  $B^\dagger$ , can be *roughly* referred to as the effectiveness

---

<sup>1</sup>Ignoring how the state error is formulated for the time being...

inverse<sup>2</sup>. The actuator positions are then solved for, avoiding saturation, subject to some constraint:

$$\underset{\in \mathbb{U}}{u} = B^\dagger(\mathbf{x}, \nu_d, t) \quad (4.5)$$

The control allocation requirements and schemes are expanded upon subsequently in Sec:4.7. Multiple attitude controllers are presented whose stability is proven with Lyapunov<sup>†</sup> stability theorem. Each controller is compared in the context of an over actuated quadrotor plant. Similarly a series of proposed allocation schemes are evaluated too. Those comparisons, their details and how controller efficacy and stability are evaluated is all presented next in Chapter:5.

---

<sup>2</sup>Direct (*pseudo*) inversion is the typical allocation scheme.

A generalized over-actuated control loop is split into a series of cascaded control blocks, each with an individual function, as illustrated in Fig:4.1. From the error state of the generated trajectory,  $\mathbf{x}_e$ , the control law designs a virtual control input,  $\vec{\nu}_d$ , which is cast as the argument to the allocation block. From the allocation law,  $B^\dagger(\mathbf{x}, \vec{\nu}_d, t)$ , physical actuator positions are obtained;  $u \in \mathbb{U}$ . Those actuator positions effect a virtual plant input,  $\vec{\nu}_c = B(\mathbf{x}, u, t)$ , which is an input to the state function's dynamics, Eq:3.79. Not shown, but implied in Fig:4.1, is the state derivative feedback of  $\dot{\mathbf{x}}$  to the plant transfer function. Finally the output tracking state is estimated with some filtration paradigm,  $\hat{\mathbf{x}} = A(\mathbf{x}, t)$ , and fed back to the error state.

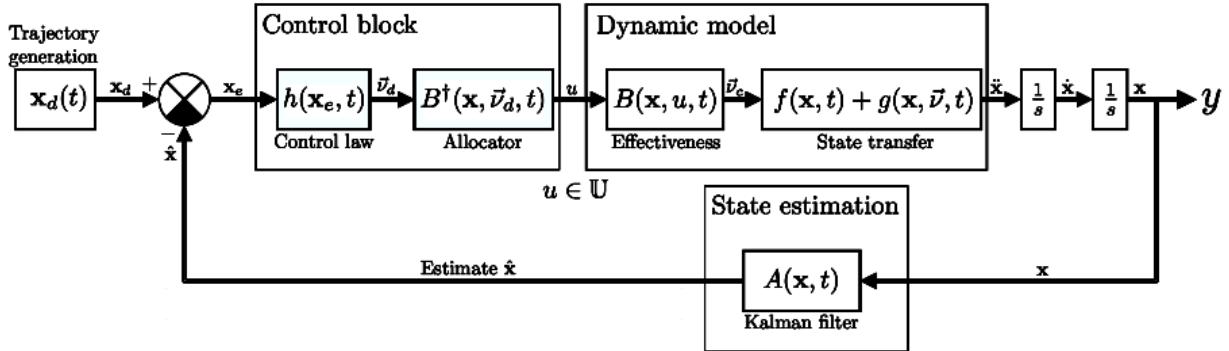


Figure 4.1: Generalized control loop with allocation

### 4.3 Control Plant Inputs

Thus far control plant inputs for the set of differential state equations, from Eq:3.79, have mostly been described with net forces and torques;  $\mu\vec{F}$  &  $\mu\vec{\tau}$ . The relationship between each propeller's rotational speed & servo positions and the its resultant output thrust direction is calculated as a quaternion transformation of produced lift force, as in Eq:3.80.

$$\mu\vec{F}(u) = \sum Q_{M_i}^*(\lambda_i, \alpha_i) \otimes T(\Omega_i) \otimes Q_{M_i}(\lambda_i, \alpha_i) \quad \in \mathcal{F}^b \quad (4.6a)$$

$$\mu\vec{\tau}(u) = \sum \vec{l} \times (Q_{M_i}^*(\lambda_i, \alpha_i) \otimes T(\Omega_i) \otimes Q_{M_i}(\lambda_i, \alpha_i)) \quad \in \mathcal{F}^b \quad (4.6b)$$

To accommodate comparison of each controller and allocation scheme, the error state control law(s) design net plant inputs  $\mu\vec{F}$  and  $\mu\vec{\tau}$ . The allocation rule then takes both net inputs as an argument to find actuator positions to effect those net inputs. As such each control law can be tested against various allocation rules and *vise versa*. However typical allocation algorithms, like pseudo-inversion, require a multiplicative relationship between plant and control inputs...

The actuator effectiveness functions in Eq:4.6 aren't readily reducible to a single multiplicative relationship with the actuator matrix  $u \in \mathbb{U}$ . Thusly the effectiveness function needs an extra layer of abstraction to incorporate a multiplicative relationship. Rather than calculating actuator positions directly from  $\vec{\nu}_d$ , a set of four 3-dimensional thrust vectors,  $\vec{T}_{1 \rightarrow 4}$  for each motor module, are calculated first.

$$\vec{\nu}_d = \begin{bmatrix} \mu\vec{F} \\ \mu\vec{\tau} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ [\vec{l}_1]_\times & [\vec{l}_2]_\times & [\vec{l}_3]_\times & [\vec{l}_4]_\times \end{bmatrix} \begin{bmatrix} \vec{T}_1 \\ \vec{T}_2 \\ \vec{T}_3 \\ \vec{T}_4 \end{bmatrix} \quad (4.7)$$

Where  $[\vec{l}_i]_\times$  is the cross product vector of the  $i^{th}$  torque arm. Individual actuator positions for each module,  $[\Omega_i, \lambda_i, \alpha_i]^T$ , can be calculated from those thrust vectors  $\vec{T}_i$  for  $i \in [1 : 4]$  with some trigonometry, ensuring that they only adhere to Eq:4.6. That trigonometric inversion<sup>3</sup> can be described as the function  $R^\dagger$ :

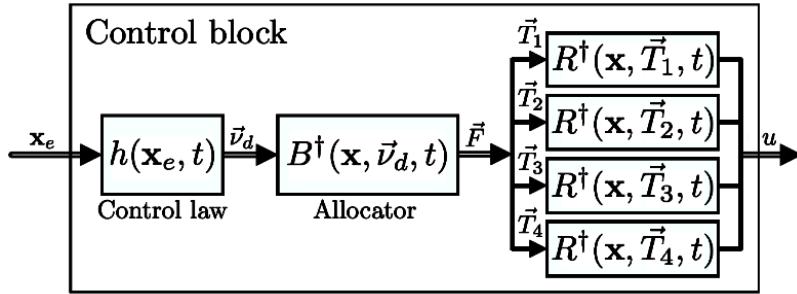
$$[\Omega_i, \lambda_i, \alpha_i]^T = R^\dagger(\mathbf{x}, \vec{F}_i, t) \quad \text{for } i \in [1 : 4] \quad (4.8)$$

<sup>3</sup>Inverting either rotation matrix operations or quaternions to solve for angular servo positions, Eq:4.109 in Sec:4.7.

To summarize; each allocation rule decomposes net force and torque vectors into four directional thrust vectors for each, or 12 directional components. The force components are an abstracted allocation layer in place of explicit actuator positions, which are subsequently solved for...

$$B^\dagger(\mathbf{x}, \vec{\nu}_d, t) = [T_{1x}, T_{1y}, T_{1z}, \dots, T_{4x}, T_{4y}, T_{4z}]^T \quad (4.9)$$

The control block in the loop (Fig:4.1) is then modified to incorporate the extra allocation abstraction level, shown in Fig:4.2. The output from that control block is still the same actuator matrix  $u \in \mathbb{U}$ . The block merely accommodates for comparison of various  $B^\dagger(\mathbf{x}, \vec{\nu}_d, t)$  allocation rules without having to redesign the remainder of the loop's structure.



**Figure 4.2:** Abstracted control block

Al allocation algorithms proposed follow the same input/output structure described in Fig:4.2. Only one allocation algorithm does, however, circumvent the virtual abstraction level of thrust vector's for each module to directly calculate actuator positions, Section:4.7.5.

Each control law is co-dependent on an accompanying allocation algorithm. Traditional control loops (under-actuated or well matched) typically have a unity allocation rule and as such require no consideration so they're mostly disregarded. Separate control laws for attitude ad position control are presented next in Section:4.5 and 4.6 respectively. Thereafter a series of allocation rules are proposed in Section:4.7. Although presented independently, the controller and allocation laws are mutually inclusive. The stability of each control law is proven objectively but actual controller tuning and optimization takes place only in the following Chapter:5, in Sec:5.1.

## Model Dependent & Independent Controllers

Two classes of controllers are presented, attitude and position control laws. The former being the primary focus of this research project and containing a more complete schedule of control treatment and controller comparison. Both control categories consider MIMO state vector loops for attitude and position states  $\mathcal{E}$  &  $Q_b$ . The allocation algorithm combines both virtual control inputs  $\vec{\nu}_d = [\mu \vec{F} \mu \vec{\tau}]^T$  generated from the two control categories to calculate actuator positions.

The control dependency on the system plant is as a consequence of the prominent actuator response dynamics, as derived previously in Sec:3.2.1. Whilst not a prerequisite for stability, plant dependent compensation certainly improves controller performances. Independent and dependent cases are only considered for one type of controller; the most basic case PD controller in Section:4.5.2. It's shown that for an independent (PD) controller to achieve global stability some stringent assumptions must first be met.

Inherent plant dependency makes backstepping controllers an attractive control paradigm in this dissertation's context. The proposed plant dependent control laws compensate for undesirable dynamics their design, basic PD & PID control structures (*and the like*) will not. The first and most basic control solution, used as a reference case, is a PD controller for attitude and position with direct-inversion<sup>4</sup> allocation, both plant dependent and independent PD controllers are compared.

<sup>4</sup>Pseudo-inversion or Moore-Penrose inversion

## 4.4 Lyapunov Stability Theorem

Lyapunov's stability theory is a critical aspect of non-linear controller design. An abundance of literature has been written on the subject<sup>5</sup> spanning through the progression of control engineering. Typically linear systems are proven<sup>†</sup> to be stable using the frequency domain with Laplace transforms, the same is not true for non-linear systems. Lyapunov's stability theorem proves (*global*) asymptotic stability for continuous time invariant systems, linear or otherwise.

The theorem applies analysis of a generalized energy function representative of a system's autonomous trajectory. A negative trajectory energy derivative will ensure the system's energy is always dissipating toward a stable settling point. Lyapunov analysis is a popular method for stability verification because system's trajectory itself needn't be explicitly defined for stability to be ascertained. Proof of Lyapunov's theorem is done with a contradiction disproof and, as such, the theoretical underpinning is somewhat cumbersome.

Despite the conceptually difficult proof, it's worth reiterating its fundamentals given that backstepping controllers are proposed later in Sec:4.5.3 for attitude control. A backstepping controller iteratively enforces Lyapunov stability criterion onto the system through the control structure. In general, given a non-linear time invariant system that follows some continually differentiable trajectory  $\mathbf{x}(t)$ , typically the trajectory is going to progress subject to some rule:

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t)) \quad (4.10)$$

Then, constructing a generalized positive-definite function (generalized energy or *Lyapunov candidate* function)  $V(x)$  for a trajectory  $x = \mathbf{x}(t)$ . A positive definite matrix,  $M$ , is defined such that  $z^T M z \geq 0 \forall z$ . As such an LCF typically has the form:

$$V = \mathbf{x}^T P \mathbf{x} \quad (4.11)$$

Given that, by its definition, the trajectory is continually differentiable; there is a partial gradient matrix for each component of  $V(x)$  in the form:

$$\nabla V(x) = \left[ \frac{\delta V(x)}{\delta x_1} \frac{\delta V(x)}{\delta x_2} \dots \frac{\delta V(x)}{\delta x_n} \right] \quad x \in \mathbb{R}^n \quad (4.12)$$

The energy function's derivative, otherwise referred to as the *Lie derivative*<sup>†</sup>, is calculated as follows:

$$\dot{V}(x) = \nabla V(x)^T f(x) = \frac{\delta V(x)}{\delta x_1} f_1(x) + \frac{\delta V(x)}{\delta x_2} f_2(x_2) + \dots + \frac{\delta V(x)}{\delta x_n} f_n(x) \quad (4.13)$$

Lyapunov's theorem states that *iff* the candidate function  $V(x)$  is positive definite with  $\dot{V}(0) = 0$  and its derivative is negative definite;  $\dot{V}(x) < 0 \forall x \neq 0$ , the system is then globally asymptotically stable. Mathematically that means, for any  $\mathbf{x}(t) \geq 0$ :

$$V(\mathbf{x}(t)) = V(\mathbf{x}(0)) + \int_0^t \dot{V}(\mathbf{x}(t)).dt \leq V(\mathbf{x}(t)) \quad (4.14)$$

Which can be physically interpreted as the system's generalized energy function always dissipating, irrespective of trajectory path taken. With a continually decreasing energy function, the system will inevitably settle to some stable point, hence the trajectory exists in some bounded  $\{x|V(x) \leq V(\mathbf{x}(t))\}$ , which is defined as global asymptotic stability. Every trajectory of  $\dot{\mathbf{x}}(t) = f(\mathbf{x}(t))$  converges to the zero<sup>6</sup> setpoint as  $t \rightarrow \infty$ .

The asymptotic stability proof can be extended to exponential stability boundedness, such that *iff* the same conditions are met and there exists some positive coefficient  $\alpha > 0$  such that  $\dot{V}(x) \leq -\alpha V(x)$ . That implies the system is globally exponentially stable as is bound in such a way that:

$$\|\mathbf{x}(t)\| \leq M e^{-\alpha t/2} \|\mathbf{x}(0)\| \quad (4.15)$$

---

<sup>5</sup>Included in almost every meritable textbook and papers; [83, 103], amongst others...

<sup>6</sup>Adapted to zero error state tracking in lieu or zero set point settling.

## 4.5 Attitude Control

### 4.5.1 The Attitude Control Problem

Set point tracking control of the attitude plant is to then design a stabilizing control torque  $\mu\vec{\tau} = h(\mathbf{x}_e, t)$  such that; for any desired attitude quaternion,  $\forall Q_d \in \mathbb{Q}$ , and an instantaneous attitude body quaternion, similarly  $\forall Q_b \in \mathbb{Q}$ , the error state asymptotically stabilizes to 0;  $Q_e \rightarrow [1 \vec{0}]^T$ . Or that:

$$\mu\vec{\tau} = h(Q_e, \dot{Q}_e) \text{ such that } \lim_{t \rightarrow \infty} Q_e = \begin{bmatrix} 1 \\ \vec{0} \end{bmatrix} \quad (4.16)$$

Quaternion error states are defined as the Hamilton product (*difference*) between the desired and instantaneous quaternion attitude states. Quaternion error states are in contrast with the subtractive relationship for Euler angle error states. The attitude error state is calculated as:

$$Q_e = Q_d^* \otimes Q_b \quad (4.17)$$

The relative angular velocity error between the body frame,  $\mathcal{F}^b$ , and the trajectory's desired frame,  $\mathcal{F}^d$ , is given as  $\vec{\omega}_e$ . The body angular velocity,  $\vec{\omega}_b$  is subject to the differential Eq:3.79d. As such there's an angular rate error:

$$\vec{\omega}_e = \vec{\omega}_d - \vec{\omega}_b \quad (4.18a)$$

The desired angular rate is taken with respect to the desired angular attitude frame, and so it must be transformed back onto the existing body frame.

$$\vec{\omega}_e = Q_e^* \otimes \vec{\omega}_d \otimes Q_e - \vec{\omega}_b \quad (4.18b)$$

Typically for the trajectories generated here the desired angular velocity is zero;  $\vec{\omega}_d = \vec{0}$ . It follows that the angular rate error is then simply the negative body angular velocity. It would be easy to incorporate a non-zero angular velocity setpoint to accommodate for higher order state derivative tracking trajectories.

$$\vec{\omega}_e = -\vec{\omega}_b \Big|_{\vec{\omega}_d=\vec{0}} \quad (4.18c)$$

The time derivative of the quaternion error state is given by Eq:3.29. The derivative  $\dot{Q}_e$  is then dependent on the angular velocity error and calculated as follows:

$$\dot{Q}_e = \frac{1}{2}Q_e \otimes \vec{\omega}_e = -\frac{1}{2}Q_e \otimes \vec{\omega}_b \Big|_{\vec{\omega}_d=\vec{0}} \quad (4.19)$$

### 4.5.2 Linear Controllers

#### PD Controller

The control law which is used as a basic reference for comparison is a simple Proportional-Derivative structured attitude controller. Specifically, a stability proof derived from the one presented *The Attitude Control Problem* [124] is used for asymptotic stability verification. An attitude PD control law, proportional to the vector quaternion error only<sup>7</sup> and angular rate error, designs the control torque as:

$$\mu\vec{\tau}_{PD} = K\vec{\omega}_e + \alpha\vec{q}_e \quad (4.20)$$

Where both  $K$  and  $\alpha$  are positive definite symmetrical  $3 \times 3$  coefficient matrices still to be determined. This control law neglects the quaternion scalar error and is susceptible to unwinding. Then using a candidate Lyapunov energy function  $V_{PD}$ :

$$V_{PD}(\vec{q}_e, \vec{\omega}_e) = \alpha\vec{q}_e^T \vec{q}_e + \alpha(q_0 - 1)^2 + \frac{1}{2}\vec{\omega}_e^T \mathbb{I}_b \vec{\omega}_e \quad (4.21)$$

---

<sup>7</sup>Such that the error is  $\in \mathbb{R}^3$ .

And recalling from Eq:3.79d that body's the angular velocity differential  $\dot{\vec{\omega}}_b$  is:

$$\dot{\vec{\omega}}_b = \mathbb{I}_b^{-1}(-\vec{\omega}_b \times \mathbb{I}_b \vec{\omega}_b + \vec{\tau}_Q + \vec{\tau}_g + \vec{Q} + \mu \vec{\tau}) \in \mathcal{F}^b \quad (4.22)$$

With actuator inputs  $u \in \mathbb{U}$  implied and  $\vec{Q}$  being a simplified representation of the net aerodynamic torque experienced by the body from the rotating propellers, drawn from Eq:3.81b. Then, exploiting a unit quaternion's inherent property, it follows that:

$$\|Q\| = \vec{q}^T \vec{q} + q_0^2 = \vec{q}^2 + q_0^2 = 1 \quad (4.23)$$

Substituting the angular velocity error state,  $\vec{\omega}_e = -\vec{\omega}_b$ , the proportional derivative LCF in Eq:4.21 is simplified<sup>8</sup> to:

$$V_{PD} = \alpha \vec{q}_e^2 + \alpha q_0^2 - 2q_0 + 1 + \frac{1}{2} \vec{\omega}_e^T \mathbb{I}_b \vec{\omega}_e \quad (4.24a)$$

$$= 2\alpha(1 - q_0) + \frac{1}{2} \vec{\omega}_b^T \mathbb{I}_b \vec{\omega}_b \quad (4.24b)$$

Similarly, using the fact that for a quaternion's derivative:

$$\dot{Q} = \begin{bmatrix} -\frac{1}{2} \vec{q}^T \vec{\omega} \\ \frac{1}{2} (\vec{q} \times q_0 \mathbb{I}) \vec{\omega} \end{bmatrix} \quad (4.25)$$

Then, substituting the above into the derivative of the LCF,  $\dot{V}_{PD}$ , yields:

$$\dot{V}_{PD} = 2\alpha \frac{1}{2} \vec{q}_e^T \vec{\omega}_e + \frac{1}{2} \dot{\vec{\omega}}_b^T \mathbb{I}_b \vec{\omega}_b + \frac{1}{2} \vec{\omega}_b \mathbb{I}_b \dot{\vec{\omega}}_b^T \quad (4.26a)$$

$$= -\alpha \vec{q}_e^T \vec{\omega}_b + \vec{\omega}_b^T \mathbb{I}_b \dot{\vec{\omega}}_b \quad (4.26b)$$

Simplifying the angular acceleration  $\dot{\vec{\omega}}_b$  and introducing the PD control law Eq:4.20,  $\mu \vec{\tau}_{PD}$ :

$$\vec{\omega}_b^T \mathbb{I}_b \dot{\vec{\omega}}_b = \vec{\omega}_b^T (-\vec{\omega}_b \times \mathbb{I}_b \vec{\omega}_b + \vec{\tau}_Q + \vec{\tau}_g + \vec{Q} - K \vec{\omega}_b + \alpha \vec{q}_e) \quad (4.27a)$$

$$\rightarrow \dot{V}_{PD} = -\alpha \vec{q}_e^T \vec{\omega}_b + \vec{\omega}_b^T (-\vec{\omega}_b \times \mathbb{I}_b \vec{\omega}_b + \vec{\tau}_Q + \vec{\tau}_g + \vec{Q} - K \vec{\omega}_b + \alpha \vec{q}_e) \quad (4.27b)$$

$$= -\alpha \vec{q}_e^T \vec{\omega}_b + \alpha \vec{\omega}_b^T \vec{q}_e - \vec{\omega}_b^T K \vec{\omega}_b + \vec{\omega}_b^T (-\vec{\omega}_b \times \mathbb{I}_b \vec{\omega}_b + \vec{\tau}_Q + \vec{\tau}_g + \vec{Q}) \quad (4.27c)$$

It follows that the transpose term  $\vec{q}_e^T \vec{\omega}_b \iff \vec{\omega}_b^T \vec{q}_e$  is interchangeable as its resultant product is the same. The LCF derivative then simplifies to:

$$\dot{V}_{PD} = -\vec{\omega}_b^T K \vec{\omega}_b + \vec{\omega}_b^T (-\vec{\omega}_b \times \mathbb{I}_b \vec{\omega}_b + \vec{\tau}_Q + \vec{\tau}_g + \vec{Q}) \quad (4.27d)$$

Then, under specific circumstances the following assumptions can be made to ensure the asymptotic stability proof can be applied. The stability obviously breaks down if any of the assumptions fail, as such the stability is not global...

1. The inertial matrix,  $\mathbb{I}_b$ , is approximately diagonal. Which, given the symmetrical design and similarly that the angular rate can be made small with appropriately slow trajectory updates, is a fair assumption then:

$$\vec{\omega}_b^T (\vec{\omega}_b \times \mathbb{I}_b \vec{\omega}_b) \approx \vec{0}$$

2. The actuator rate torque responses,  $\vec{\tau}_Q$ , are all second order effects dependent on  $\dot{u}$ . Typically the actuator rates are going to be kept small and so any of the inertial responses to those position changes are small enough to be considered negligible. The approximation is made:

$$\vec{\tau}_Q \approx \vec{0}$$

3. Finally, for the sake of the stability proof, the eccentric gravitational torque arm is neglected,  $\vec{\tau}_g \approx \vec{0}$ . Such a situation only holds true if  $u \approx \vec{0}$  or that servo actuator positions<sup>9</sup> are close to their zero positions.

<sup>8</sup>The quaternion scalar  $q_0$  in Eq:4.24 is implied to be the quaternion error state scalar

<sup>9</sup>Excluding propeller rotational speeds, considering only the servo positions

All of these assumptions are made under extraneous circumstances and can't be assumed for almost all of the prototype's flight envelope. The plant independent case is considered and simulated purely for contrition; mainly to demonstrate the need for plant dependent compensation. All subsequent control laws compensate for the plant dynamic response torques introduced in Section:3.2.

If each of the assumptions made hold true, then the Lyapunov energy function's derivative is approximately negative definite.

$$\dot{V}_{PD} \approx -\alpha \vec{q}_e^T \vec{\omega}_b + \vec{\omega}_b^T (-K \vec{\omega}_b + \alpha \vec{q}_e) \quad (4.28a)$$

$$\Rightarrow \dot{V}_{PD} = -\vec{\omega}_b^T K \vec{\omega}_b = -K \|\vec{\omega}_b\|^2 < 0 \quad \forall (\vec{\omega}_e, Q_b) = \mathbf{z}(t) \quad (4.28b)$$

Where  $\mathbf{z}(t)$  is a generalized attitude trajectory which includes  $\vec{\omega}_b$  &  $Q_b$  and  $K$  is a symmetrical<sup>10</sup> positive (*definite*) 3X3 coefficient matrix. Then from Lyapunov stability theorem the limits exist;  $\lim_{t \rightarrow \infty} \vec{\omega}_e = \vec{0}$ ,  $\lim_{t \rightarrow \infty} \vec{q}_e = 0$  and  $\lim_{t \rightarrow \infty} (1 - q_0) = 0$ . Hence  $Q_e \rightarrow [1 \ \vec{0}]^T$  as  $t \rightarrow \infty$ , asymptotically stabilizing the attitude error state.

Introducing model dependent compensation to the PD control law in Eq:4.20 alleviates the stringent requirements on assumptions 1 through 3.

$$\mu \vec{\tau}_{PD} = \vec{\omega}_b \times \mathbb{I}_b \vec{\omega}_b - \vec{\tau}_Q - \vec{\tau}_g - \vec{Q} + K \vec{\omega}_b + \alpha \vec{q}_e \quad (4.29)$$

The resultant stability proof for Eq:4.29 is much the same as that for the independent case, Eq:4.20, and uses the identical LCF from Eq:4.21. The resultant dependent control law is no longer reliant on the very broad assumptions needed for independent stability to be achieved. The dynamic compensation in Eq:4.29 improves control response, especially considering the form of unwanted dynamics which have already quantified previously and modelled with *relative* confidence.

### Auxiliary Plant Controller

Expanding on what has, in practice<sup>11</sup>, proven to be a very popular and effective control law for attitude stabilization, McGilvray et al. [2006] [121] suggested introducing an auxiliary plant term to a Proportional-Derivative structure. Most significantly, their altered PD controller adds auxiliary terms proportional to the quaternion time derivative error. The critical component of that change is the part of the auxiliary plant proportional to the quaternion scalar. The scalar term is otherwise neglected in the previous PD control law (Sec:4.5.2) and prevents unwinding if incorporated.

The modified (*auxilliarily*) PD control torque is a function of errors states for quaternions, angular rates and quaternion rates. The compensating plant dependent control law is given as:

$$\mu \vec{\tau}_{XPD} = \underbrace{-\Gamma_2 \tilde{\Omega} - \Gamma_3 \vec{q}_e + \mathbb{I}_b \dot{\tilde{\Omega}}}_{Independent} + \underbrace{\vec{\omega}_b \times \mathbb{I}_b \vec{\omega}_b + \vec{\tau}_Q + \vec{\tau}_g + \vec{Q}}_{Compensation} \quad (4.30)$$

In which case the coefficients<sup>12</sup>  $\Gamma_2$  &  $\Gamma_3$  are both diagonal positive definite coefficient matrices and  $\Gamma_1$ , introduced subsequently in Eq:4.32, is a p.d symmetrical coefficient matrix. The auxiliary plants  $\tilde{\Omega}$  &  $\dot{\tilde{\Omega}}$  are defined as follows and draw on Eq:4.25 for definition of some aspects. For the first auxiliary plant  $\tilde{\Omega}$  is proportional to the quaternion error and hence  $\dot{\tilde{\Omega}}$  is a quaternion derivative term:

$$\tilde{\Omega} = -\Gamma_1 \vec{q}_e \Rightarrow \dot{\tilde{\Omega}} = -\Gamma_1 \dot{\vec{q}}_e \quad (4.31a)$$

$$\dot{\tilde{\Omega}} = -\frac{1}{2} \Gamma_1 (q_0 \mathbb{I}_{3X3} + [\vec{q}_e]_\times) \vec{\omega}_e \quad (4.31b)$$

$$= \frac{1}{2} \Gamma_1 (q_0 \mathbb{I}_{3X3} + [\vec{q}_e]_\times) \vec{\omega}_b \quad (4.31c)$$

<sup>10</sup>Symmetry, unlike the subsequent Auxiliary controller, is not a prerequisite for stability...

<sup>11</sup>Practical examples of various quadrotor attitude PD controllers listed in Table:1.1 from Sec:1.2.1.

<sup>12</sup>Reiterating that exact coefficient values are determined in Chapter:5...

The second auxiliary plant,  $\tilde{\Omega}$ , is a term proportional to a combined quaternion vector and angular velocity error state.

$$\tilde{\Omega} = \vec{\omega}_e - \bar{\Omega} = \vec{\omega}_e + \Gamma_1 \vec{q}_e \quad (4.32a)$$

$$= -\vec{\omega}_b + \Gamma_1 \vec{q}_e \quad (4.32b)$$

Using an LCF similar to the basic one  $V_{PD}$  from Eq:4.21, but introducing an auxiliary term  $\tilde{\Omega}$  into the candidate function  $V_{XPD}$ :

$$V_{XPD}(\vec{q}_e, \tilde{\Omega}) = \vec{q}_e^T \vec{q}_e + (q_0 - 1)^2 + \frac{1}{2} \tilde{\Omega}^T (\Gamma_3^{-1} \mathbb{I}_b) \tilde{\Omega} \quad (4.33)$$

Again using the simplification from a quaternion's inherent properties in Eq:4.23, the LCF from Eq:4.33 then simplifies with the following derivative:

$$V_{XPD} = 2(1 - q_0) + \frac{1}{2} \tilde{\Omega}^T (\Gamma_3^{-1} \mathbb{I}_b) \tilde{\Omega} \quad (4.34a)$$

$$\dot{V}_{XPD} = 2 \frac{1}{2} \vec{q}_e^T \vec{\omega}_e + \frac{1}{2} \dot{\tilde{\Omega}}^T (\Gamma_3^{-1} \mathbb{I}_b) \tilde{\Omega} + \frac{1}{2} \tilde{\Omega}^T (\Gamma_3^{-1} \mathbb{I}_b) \dot{\tilde{\Omega}} \quad (4.34b)$$

$$\dot{V}_{XPD} = -\vec{q}_e^T \vec{\omega}_b + \frac{1}{2} \dot{\tilde{\Omega}}^T (\Gamma_3^{-1} \mathbb{I}_b) \tilde{\Omega} + \frac{1}{2} \tilde{\Omega}^T (\Gamma_3^{-1} \mathbb{I}_b) \dot{\tilde{\Omega}} \quad (4.34c)$$

It follows that from Eq:4.32 then the auxiliary plant derivative  $\dot{\tilde{\Omega}}$  is:

$$\dot{\tilde{\Omega}} = -\dot{\vec{\omega}}_b + \Gamma_1 \dot{\tilde{\Omega}} \Rightarrow \dot{\vec{\omega}}_b = \mathbb{I}_b^{-1} (-\vec{\omega}_b \times \mathbb{I}_b \vec{\omega}_b + \vec{\tau}_Q + \vec{\tau}_g + \vec{Q} + \mu \vec{\tau}) \quad (4.35a)$$

$$\therefore \dot{\tilde{\Omega}} = -\mathbb{I}_b^{-1} (-\vec{\omega}_b \times \mathbb{I}_b \vec{\omega}_b + \vec{\tau}_Q + \vec{\tau}_g + \vec{Q} + \mu \vec{\tau}) - \Gamma_1 \dot{\tilde{\Omega}} \quad (4.35b)$$

Substituting the auxiliary PD control law,  $\mu \vec{\tau}_{XPD}$  from Eq:4.30, into the auxiliary plant derivative then yields:

$$\rightarrow \dot{\tilde{\Omega}} = \mathbb{I}_b^{-1} (\mathbb{I}_b \dot{\tilde{\Omega}} - \Gamma_2 \tilde{\Omega} - \Gamma_3 \vec{q}_e) - \dot{\tilde{\Omega}} \quad (4.35c)$$

$$= \mathbb{I}_b^{-1} (-\Gamma_2 \tilde{\Omega} - \Gamma_3 \vec{q}_e) \quad (4.35d)$$

From the positive symmetric (or *diagonal*) properties of the coefficient matrices  $\Gamma_1, \Gamma_2$  &  $\Gamma_3$ , the auxiliary plant's transpose is then:

$$\dot{\tilde{\Omega}}^T = \mathbb{I}_b^{-1} (-\Gamma_2 \tilde{\Omega}^T - \Gamma_3 \vec{q}_e^T) \quad (4.36)$$

It then follows that the P.D auxiliary plant component in the LCF, Eq:4.33, simplifies:

$$\frac{1}{2} \dot{\tilde{\Omega}}^T (\Gamma_3^{-1} \mathbb{I}_b) \tilde{\Omega} = \frac{1}{2} (-\Gamma_2 \tilde{\Omega}^T - \Gamma_3 \vec{q}_e^T) \Gamma_3^{-1} \tilde{\Omega} \quad (4.37a)$$

$$= \frac{1}{2} (-\tilde{\Omega}^T \Gamma_2 \Gamma_3^{-1} \tilde{\Omega} - \vec{q}_e^T \tilde{\Omega}) \quad (4.37b)$$

And substituting Eq:4.32,  $\vec{q}_e^T \tilde{\Omega} = -\vec{q}_e^T \vec{\omega}_b + \Gamma_1 \vec{q}_e^T$ :

$$\frac{1}{2} (-\tilde{\Omega}^T \Gamma_2 \Gamma_3^{-1} \tilde{\Omega} + \vec{q}_e^T \vec{\omega}_b - \vec{q}_e^T \Gamma_1 \vec{q}_e) \quad (4.37c)$$

Similarly, for the transposed energy function counterpart:

$$\frac{1}{2} \tilde{\Omega}^T (\Gamma_3^{-1} \mathbb{I}_b) \dot{\tilde{\Omega}} = \frac{1}{2} (-\tilde{\Omega} \Gamma_2 \Gamma_3^{-1} \tilde{\Omega}^T + \vec{q}_e^T \vec{\omega}_b - \vec{q}_e^T \Gamma_1 \vec{q}_e) \quad (4.37d)$$

Which, when substituted back into Eq:4.34c, then simplifies the LCF derivative to negative definite:

$$\Rightarrow \dot{V}_{XPD} = -\vec{q}_e^T \Gamma_1 \vec{q}_e - \tilde{\Omega} \Gamma_2 \Gamma_3^{-1} \tilde{\Omega}^T < \vec{0} \quad \forall (\vec{q}_e, \tilde{\Omega}) \quad (4.38)$$

As such, the control law  $\mu \vec{\tau}_{XPD}$  asymptotically stabilizes the attitude plant. Both  $\tilde{\Omega}$  and  $\vec{q}_e$  tend to  $\vec{0}$ , or more specifically the following limits exist:

$$\lim_{t \rightarrow \infty} \vec{q}_e = 0 \text{ and } \lim_{t \rightarrow \infty} \tilde{\Omega} = 0 \quad (4.39a)$$

Then, from the auxiliary plant definition in Eq:4.32, the extended limits present themselves;

$$\lim_{t \rightarrow \infty} \vec{\omega}_b = 0 \quad \text{and} \quad \lim_{t \rightarrow \infty} \tilde{\Omega} = 0 \quad (4.39b)$$

The stability proof for  $V_{XPD}$  can then be extended to a stable exponentially bounded trajectory. From a quaternion's inherent definition it follows that  $0 \leq q_0 \leq 1$ . It can then be stated that:

$$1 - q_0 \leq 1 - q_0^2 = \|\vec{q}_e\|^2 \quad (4.40)$$

Seeing that exponential stability is a maximal boundedness proof, the relationship Eq:4.40 can then replace the quaternion scalar term  $2(1 - q_e)$  in  $V_{XPD}$ . For the stability proof the LCF is rewritten in terms of its component's norm(s) to produce the inequality:

$$V_{XPD} = \vec{q}_e^T \vec{q}_e + (q_0 - 1)^2 + \frac{1}{2} \tilde{\Omega}^T (\Gamma_3^{-1} \mathbb{I}_b) \tilde{\Omega} \quad (4.41a)$$

$$\rightarrow V_{XPD} \leq 2 \|\vec{q}_e\|^2 + \frac{1}{2} \Gamma_3^{-1} \mathbb{I}_b \|\tilde{\Omega}\|^2 \quad (4.41b)$$

Similarly the LCF derivative can be written in terms of its norms as:

$$\dot{V}_{XPD} = -\Gamma_2 \Gamma_3^{-1} \|\tilde{\Omega}\|^2 - \Gamma_1 \|\vec{q}_e\|^2 \quad (4.41c)$$

$V_{XPD}$  then gains a maximum such that:

$$V_{XPD} \leq \max \left\{ 2, \frac{\lambda_{\max}(\Gamma_3^{-1} \mathbb{I}_b)}{2} \right\} (\|\vec{q}_e\|^2 + \|\tilde{\Omega}\|^2) \quad (4.42)$$

Where the function  $\lambda_{\max}$  represents the maximum eigenvalue of its argument, in this case  $\Gamma_3^{-1} \mathbb{I}_b$ . Similarly the *negative definite* LCF derivative is bound by the minimum:

$$\dot{V}_{XPD} \leq -\min \{ \lambda_{\min}(\Gamma_1), \lambda_{\min}(\Gamma_2 \Gamma_3^{-1}) \} (\|\vec{q}_e\|^2 + \|\tilde{\Omega}\|^2) \quad (4.43)$$

Therefore there exists some ratio  $\alpha > 0$  that satisfies the relationship requirement between the LCF and its derivative;  $\dot{V}_{XPD} \leq -\alpha V_{XPD}$ , where  $\alpha$  is defined as the ratio of those maxima<sup>13</sup>:

$$\alpha = \frac{\min \{ \lambda_{\min}(\Gamma_1), \lambda_{\min}(\Gamma_2 \Gamma_3^{-1}) \}}{\max \{ 2, \frac{\lambda_{\max}(\Gamma_3^{-1} \mathbb{I}_b)}{2} \}} \quad (4.44)$$

The attitude trajectory  $(\vec{q}_e(t), \tilde{\Omega}(t))$  is then exponentially bounded by:

$$(\|\vec{q}_e(t)\|, \|\tilde{\Omega}(t)\|) \leq M e^{-\alpha t/2} (\|\vec{q}_e(0)\|, \|\tilde{\Omega}(0)\|) \quad (4.45)$$

The above stability proof for the auxiliary attitude controller was expanded upon and derived from McGilvray et al. [2006] [121] and adapted to fit attitude setpoint tracking. The fact that the auxiliary plant controller introduces the quaternion error, which is dependent on the quaternion scalar, dramatically improves controller performance. The exponential stability notably improves settling times and overshoot errors, seen next in Chapter:5.

Interestingly an earlier paper by Joshi, et al. [1995] [63] was the precursor for PD based attitude plants with asymptotic exponential stability. Joshi's control law first proposed didn't make use of any defined auxiliary plants, unlike [121], but equivalent terms were effectively incorporated. That control law developed for spacecraft attitude tracking proposed a very similar exponentially stabilizing control scheme to that of  $\mu \vec{\tau}_{XPD}$ . That controller, when changed to the notational convention used above, designs body torque as:

$$\mu \vec{\tau}'_{XPD} = -\frac{1}{2} \left[ ([\vec{q}_e]_\times + q_0 \mathbb{I}_{3 \times 3}) \Gamma_1 + \alpha (1 - q_0 \mathbb{I}) \right] \vec{q}_e - \Gamma_2 \vec{\omega}_b \quad (4.46)$$

<sup>13</sup>A maximum and minimum

### 4.5.3 Non-linear Controllers

Backstepping controllers(presented in [10, 66, 68],etc...) are a popular choice for non-linear attitude control plants. The process, through iterative design, enforces Lyapunov stability criteria to ensure asymptotic stability. In a report [128] Van Kampen, et al. [2008] describes fundamental backstepping algorithms. Ideal backstepping control is a precise control solution which requires exact plant matching, something that is difficult to achieve in practice. Another caveat of IBC control is poor disturbance rejection, being especially susceptible to plant uncertainty. The ideal backstepping algorithm can then be extended to incorporate non-idealities. The disturbance and uncertainty (*estimate error*) terms are incorporated into the LCF energy function. By Lyapunov's theorem their respective estimation error terms are stabilized.

#### Ideal Backstepping Controller

Starting with the ideal case for the first proposed backstepping controller; it's assumed the attitude plant described in Eq:3.79d, from the consolidated model in Sec:3.4, absolutely matches the dynamics of the physical prototype. The ideal backstepping controller aims to perfectly compensate for the plant's dynamic response to trajectory inputs. Ignoring any uncertainty associated with the dynamic equation, the aim here is to apply a stabilizing torque design. Recalling the quaternion tracking error from Eq:4.17;  $Q_e = Q_d^* \otimes Q_b$ . Then considering the first LCF proposal:

$$V_1(\vec{q}_e) = \vec{q}_e^T \vec{q}_e + (q_0 - 1)^2 \quad (4.47)$$

Which, after substituting in the quaternion derivatives and *without* using the quaternion simplification in Eq:4.24, has a Lie derivative:

$$\dot{V}_1 = 2\vec{q}_e^T \dot{\vec{q}}_e + 2(q_0 - 1)\dot{q}_0 \quad (4.48a)$$

$$= 2\vec{q}_e^T \frac{1}{2} ([\vec{q}_e]_{\times} + q_0 \mathbb{I}_{3 \times 3}) \vec{\omega}_e - 2(q_0 - 1) \frac{1}{2} \vec{q}_e^T \vec{\omega}_e \quad (4.48b)$$

$$= \vec{q}_e^T ([\vec{q}_e]_{\times} + q_0 \mathbb{I}_{3 \times 3}) \vec{\omega}_e - q_0 \vec{q}_e^T \vec{\omega}_e + \vec{q}_e^T \vec{\omega}_e \quad (4.48c)$$

$$= \vec{q}_e^T [\vec{q}_e]_{\times} \vec{\omega}_e + \vec{q}_e^T \vec{\omega}_e \quad (4.48d)$$

$$= -\vec{q}_e^T [\vec{q}_e]_{\times} \vec{\omega}_b - \vec{q}_e^T \vec{\omega}_b \quad (4.48e)$$

Then choosing the first stabilizing function,  $z_1$ , with a virtual backstepping control input  $\Omega_d$ . It's important to note that  $\Omega_d$  is used here to differentiate the backstepping *desired* value from the trajectory instructed  $\vec{\omega}_d$  from Eq:4.18a, or any auxiliary plants defined previously for the Auxiliary PD controller in Sec:4.5.2.

$$\vec{\omega}_b \Rightarrow \Omega_d = \Gamma_1 \vec{q}_e \quad (4.49)$$

Where  $\Gamma_1$  is the first symmetric positive definite coefficient matrix, a fact that is important to stress due to positive definite matrix's invertability. That stabilizing law then simplifies the LCF derivative  $\dot{V}_1$  to the negative definite term:

$$\dot{V}_1 = -\vec{q}_e^T [\vec{q}_e]_{\times} \Omega_d - \vec{q}_e^T \Omega_d \quad (4.50a)$$

$$= -\vec{q}_e^T [\vec{q}_e]_{\times} \Gamma_1 \vec{q}_e - \vec{q}_e^T \Gamma_1 \vec{q}_e \quad (4.50b)$$

$$= -\vec{q}_e^T \Gamma_1 \vec{q}_e \quad (4.50c)$$

A vector cross product with itself has a zero resultant. However, that stabilizing virtual plant input  $\Omega_d$  has its own associated error,  $z_1$ , which then needs to be stabilized as well:

$$z_1 = \vec{\omega}_b - \Omega_d = \vec{\omega}_b - \Gamma_1 \vec{q}_e \quad (4.51a)$$

$$\rightarrow \vec{\omega}_b = z_1 - \Gamma_1 \vec{q}_e \quad (4.51b)$$

Introducing that error  $z_1$  into a second LCF, which expands the first proposed  $V_1$ . Here is where it's important that  $\Gamma_1$  is p.d & symmetrical:

$$V_2(\vec{q}_e, z_1) = V_1(\vec{q}_e) + \frac{1}{2} z_1^T \Gamma_1^{-1} z_1 \quad (4.52a)$$

$$= \vec{q}_e^T \vec{q}_e + (q_0 - 1)^2 + \frac{1}{2} z_1^T \Gamma_1^{-1} z_1 \quad (4.52b)$$

That first error  $z_1$  has its own derivative, and recalling  $\dot{\vec{\omega}}_b$  from earlier with an as yet undefined controllable input  $\mu\vec{\tau}$ , which still has plant dependency compensation.

$$\dot{z}_1 = \dot{\vec{\omega}}_b - \Gamma_1 \dot{\vec{q}}_e \quad (4.53a)$$

$$= \dot{\vec{\omega}}_b - \frac{\Gamma_1}{2} ([\vec{q}_e]_{\times} + q_0 \mathbb{I}_{3 \times 3}) \vec{\omega}_e \quad (4.53b)$$

$$= \mathbb{I}_b^{-1} (-\vec{\omega}_b \times \mathbb{I}_b \vec{\omega}_b + \vec{\tau}_Q + \vec{\tau}_g + \vec{Q} + \mu \vec{\tau}) + \frac{\Gamma_1}{2} ([\vec{q}_e]_{\times} + q_0 \mathbb{I}_{3 \times 3}) \vec{\omega}_b \quad (4.53c)$$

So then, following from Eq:4.53c, finding the derivative  $\dot{V}_2$ :

$$\begin{aligned} \dot{V}_2(\vec{q}_e, z_1) &= \vec{q}_e^T (z_1 - \Gamma_1 \vec{q}_e) + z_1^T \Gamma_1^{-1} \left( \mathbb{I}_b^{-1} (-\vec{\omega}_b \times \mathbb{I}_b \vec{\omega}_b + \vec{\tau}_Q + \vec{\tau}_g + \vec{Q} + \mu \vec{\tau}) \right. \\ &\quad \left. + \frac{\Gamma_1}{2} ([\vec{q}_e]_{\times} + q_0 \mathbb{I}_{3 \times 3}) \vec{\omega}_b \right) \end{aligned} \quad (4.54a)$$

$$\begin{aligned} &= -\vec{q}_e^T \Gamma_1 \vec{q}_e + z_1^T \Gamma_1^{-1} \left( \Gamma_1 \vec{q}_e + \mathbb{I}_b^{-1} (\vec{\omega}_b \times \mathbb{I}_b \vec{\omega}_b + \vec{\tau}_Q + \vec{\tau}_g + \vec{Q} + \mu \vec{\tau}) \right. \\ &\quad \left. + \frac{\Gamma_1}{2} ([\vec{q}_e]_{\times} + q_0 \mathbb{I}_{3 \times 3}) \vec{\omega}_b \right) \end{aligned} \quad (4.54b)$$

Then proposing the compensated stabilizing backstepping control law:

$$\mu \vec{\tau}_{IBC} = \vec{\omega}_b \times \mathbb{I}_b \vec{\omega}_b - \vec{\tau}_Q - \vec{\tau}_g - \vec{Q} - \mathbb{I}_b \Gamma_1 \vec{q}_e - \frac{\mathbb{I}_b \Gamma_1}{2} ([\vec{q}_e]_{\times} + q_0 \mathbb{I}_{3 \times 3}) \vec{\omega}_b - \mathbb{I}_b \Gamma_2 z_1 \quad (4.55)$$

With  $\Gamma_2$  being another positive definite symmetric coefficient matrix. Then with the control law  $\mu \vec{\tau}_{IBC}$  introduced into the LCF derivative  $\dot{V}_2$  simplifies to negative definite:

$$\dot{V}_2 = -\vec{q}_e^T \Gamma_1 \vec{q}_e - z_1^T \Gamma_2 z_1 \leq 0 \quad \forall (\vec{q}_e, z_1) \quad (4.56)$$

As such  $\vec{q}_e \rightarrow 0$  &  $z_1 \rightarrow 0$  as  $t \rightarrow \infty$ . Similarly  $z_1 \rightarrow 0$ , which leads to the limit:

$$\lim_{t \rightarrow \infty} (\vec{\omega}_b - \Gamma_1 \vec{q}_e) = 0 \quad (4.57)$$

Because the quaternion error vector already tends to 0;  $\vec{q}_e \rightarrow 0$ , it follows that  $\vec{\omega}_b \rightarrow 0$ . It can also be said that, from the definition of  $\vec{\omega}_e$ , that the angular velocity error stabilizes too. There is a distinct similarity in the structure of  $\mu \vec{\tau}_{IBC}$  from Eq:4.55 and that of the auxiliary PD controller presented in Eq:4.30. Furthermore, using the same reasoning from Eq:4.41, the exponential stability proof then follows:

$$V_{IBC} \leq V_2 = 2 \|\vec{q}_e\|^2 + \frac{\Gamma_1^{-1}}{2} \|z_1\|^2 \quad (4.58a)$$

$$\dot{V}_{IBC} \leq \dot{V}_2 = -\Gamma_1 \|\vec{q}_e\|^2 - \Gamma_2 \|z_1\|^2 \quad (4.58b)$$

Then both the energy function and its derivative are bound respectively by:

$$V_{IBC} \leq \max \left\{ 2, \frac{\lambda_{\max}(\Gamma_1^{-1})}{2} \right\} (\|\vec{q}_e\|^2 + \|z_1\|^2) \quad (4.59a)$$

$$\dot{V}_{IBC} \leq \min \{ \lambda_{\min}(\Gamma_1), \lambda_{\min}(\Gamma_2) \} (\|\vec{q}_e\|^2 + \|z_1\|^2) \quad (4.59b)$$

Which then leads to a similar exponential stability trajectory boundedness such that:

$$\dot{V}_{IBC} \leq \alpha V_{IBC} \quad (4.60a)$$

$$\therefore (\|\vec{q}_e(t)\|, \|z_1(t)\|) \leq M e^{-\alpha t/2} (\|\vec{q}_e(0)\|, \|z_1(0)\|) \quad (4.60b)$$

## Adaptive Backstepping Controller

As effective as the control law defined above in Section:4.5.3 may be, it lacks suitable disturbance rejection properties. Any plant uncertainties or disturbances encountered would adversely affect the controller in a dramatic manner (Sec:5.4). Introducing a term for lumped uncertainty/disturbance torques,  $\vec{L}$ , into the dynamic equations leads to:

$$\dot{\vec{\omega}}_b = \mathbb{I}_b^{-1}(-\vec{\omega}_b \times \mathbb{I}_b \vec{\omega}_b + \vec{\tau}_Q + \vec{\tau}_g + \vec{Q} + \vec{L} + \mu \vec{\tau}) \quad (4.61)$$

It would obviously be easy to simply introduce a compensation term for  $-\vec{L}$  into the control law. In practice, however, it is very difficult to approximate a disturbance term without *apriori* knowledge about any of its properties. Noise compensation in sensors can be done easily due to the known frequency bandwidth which that noise occurs in, the same cannot be said for wind disturbances and the like.

An approximate estimation term  $\hat{L}$  has to be used for that disturbance compensation in the designed control torque  $\mu \vec{\tau}$ . That estimate term is then going to have its own error from the physical disturbance affecting the system:

$$\tilde{L} = \vec{L} - \hat{L} \quad (4.62)$$

The purpose of adaptive backstepping is to introduce that estimate error term into an LCF and develop a derivative term for  $\dot{\hat{L}}$ , or a disturbance update law, such that even the estimate error asymptotically stabilizes. Typically, that disturbance update rule is the contribution of satellite and general attitude control papers. Similar terms can be introduced for plant uncertainty which can similarly be adapted for but are not included here...

The estimate error is then introduced into the LCF from an ideal backstepping control, in order for it to be dissipated as per Lyapunov theorem.

$$V_{ABC}(\vec{q}_e, z_1, \tilde{L}) = V_{IBC}(\vec{q}_e, z_1) + \frac{1}{2} \tilde{L}^T \Gamma_L^{-1} \tilde{L} \quad (4.63a)$$

$$= \vec{q}_e^T \vec{q}_e + (q_0 - 1)^2 + \frac{1}{2} z_1^T \Gamma_1^{-1} z_1 + \frac{1}{2} \tilde{L}^T \Gamma_L^{-1} \tilde{L} \quad (4.63b)$$

Where the positive symmetric matrix  $\Gamma_L \geq 0 \in \mathbb{R}^{3 \times 3}$  is termed as the adaptation gain coefficient matrix. Those particular coefficients determine how responsive the system is to disturbances and the rate at which it adapts to compensate for them. Then, to prove stability one starts with the Lie derivative  $\dot{V}_{ABC}$ :

$$\dot{V}_{ABC}(\vec{q}_e, z_1, \tilde{L}) = \dot{V}_{IBC}(\vec{q}_e, z_1) + \frac{1}{2} \dot{\tilde{L}}^T \Gamma_L^{-1} \tilde{L} + \frac{1}{2} \tilde{L}^T \Gamma_L^{-1} \dot{\tilde{L}} \quad (4.64)$$

Recalling the definition of  $\tilde{L}$  from Eq:4.62. For its derivative  $\dot{\tilde{L}}$  it's reasonable to assume the dynamics of the physical disturbance  $\vec{L}$  are far slower than the time constant of the control system, or that  $\dot{\vec{L}} \ll \dot{\hat{L}}$ . Then it follows:

$$\dot{\tilde{L}} = \dot{\vec{L}} - \dot{\hat{L}} \approx \vec{0} - \dot{\hat{L}} = -\dot{\hat{L}} \quad (4.65)$$

Substituting that estimation error rate back into the derivative  $\dot{V}_{ABC}$ , which expands upon Eq:4.54, yields:

$$\begin{aligned} \dot{V}_{ABC}(\vec{q}_e, z_1, \tilde{L}) &= \vec{q}_e^T (z_1 - \Gamma_1 \vec{q}_e) + z_1^T \Gamma_1^{-1} \left( \mathbb{I}_b^{-1} (-\vec{\omega}_b \times \mathbb{I}_b \vec{\omega}_b + \vec{\tau}_Q + \vec{\tau}_g + \vec{Q} + \vec{L} + \mu \vec{\tau}) \right. \\ &\quad \left. + \frac{\Gamma_1}{2} ([\vec{q}_e]_\times + q_0 \mathbb{I}_{3 \times 3}) \vec{\omega}_b \right) - \tilde{L}^T \Gamma_L^{-1} \dot{\hat{L}} \end{aligned} \quad (4.66a)$$

And using a similar control law to  $\mu \vec{\tau}_{IBC}$ , which has a disturbance estimate compensation term:

$$\mu \vec{\tau}_{ABC} = \vec{\omega}_b \times \mathbb{I}_b \vec{\omega}_b - \vec{\tau}_Q - \vec{\tau}_g - \vec{Q} - \hat{L} - \mathbb{I}_b \Gamma_1 \vec{q}_e - \frac{\Gamma_1 \mathbb{I}_b}{2} ([\vec{q}_e]_\times + q_0 \mathbb{I}_{3 \times 3}) \vec{\omega}_b - \mathbb{I}_b \Gamma_2 z_1 \quad (4.67a)$$

Which reduces the energy function's derivative to:

$$\dot{V}_{ABC} = -\vec{q}_e^T \Gamma_1 \vec{q}_e - z_1^T \Gamma_2 z_1 + z_1^T \Gamma_1^{-1} \left( \mathbb{I}_b^{-1} (\vec{L} - \hat{L}) \right) - \tilde{L}^T \Gamma_L^{-1} \dot{\hat{L}} \quad (4.67b)$$

$$= -\vec{q}_e^T \Gamma_1 \vec{q}_e - z_1^T \Gamma_2 z_1 + z_1^T (\Gamma_1^{-1} \mathbb{I}_b^{-1}) \tilde{L} - \tilde{L}^T \Gamma_L^{-1} \dot{\hat{L}} \quad (4.67c)$$

$$= -\vec{q}_e^T \Gamma_1 \vec{q}_e - z_1^T \Gamma_2 z_1 + \tilde{L}^T \Gamma_L^{-1} (\Gamma_1^{-1} \Gamma_L \mathbb{I}_b^{-1} z_1 - \dot{\hat{L}}) \quad (4.67d)$$

The decision then needs to be made as to how the disturbance estimate is going to be updated, or what  $\dot{\hat{L}}$  is defined as. The clear choice would be to compensate for the final term in the LCF, making it purely negative definite:

$$\dot{\hat{L}} = \Gamma_1^{-1} \Gamma_L \mathbb{I}_b^{-1} z_1 = \Gamma_1^{-1} \Gamma_L \mathbb{I}_b^{-1} \vec{\omega}_b - \Gamma_L \mathbb{I}_b^{-1} \vec{q}_e \quad (4.68)$$

The disturbance is therefore compensated for and the estimate error is ensured to have asymptotic stability seeing that  $V_{ABC}$  is positive definite.

$$\dot{V}_{ABC} = -\vec{q}_e^T \Gamma_1 \vec{q}_e - z_1^T \Gamma_2 z_1 < \vec{0} \quad \forall (\vec{q}_e, z_1, \tilde{L}) \quad (4.69)$$

Exponential stability for the plant however cannot be proven with the above control and disturbance laws, there is no non-zero estimate error coefficient in the LCF derivative. A lot of work has been done on the statistical nature of disturbance approximation and how best to adapt a non-linear control system to the influence of unwanted disturbances. An interesting approach would be to use the previous disturbance estimate,  $\vec{L} = \hat{L}_{n-1}$ , such that:

$$\tilde{L}' = \vec{L} - \hat{L} = (\hat{L}_{n-1} - \hat{L}_n) \quad (4.70a)$$

$$\dot{\tilde{L}'} = \Gamma_1^{-1} \Gamma_L \mathbb{I}_b^{-1} z_1 + \tilde{L}' \quad (4.70b)$$

$$\dot{\hat{L}} = \Gamma_1^{-1} \Gamma_L \mathbb{I}_b^{-1} \vec{\omega}_b - \Gamma_L \mathbb{I}_b^{-1} \vec{q}_e + (\hat{L}_{n-1} - \hat{L}_n) \quad (4.70c)$$

$$\therefore \dot{V}'_{ABC} = -\vec{q}_e^T \Gamma_1 \vec{q}_e - z_1^T \Gamma_2 z_1 - \tilde{L}'^T \Gamma_L^{-1} \tilde{L}' \quad (4.70d)$$

Given that the starting estimate  $\hat{L}_0 = \vec{0}$  and that the change of disturbance over a single control cycle is going to be small once the approximator has settled, its fair to assume the following:

$$\lim_{t \rightarrow \infty} \tilde{L}' = (\hat{L}_{n-1} - \hat{L}_n) \rightarrow \tilde{L} \quad (4.71)$$

Then, it leads to the following LCF derivative which can then prove exponential stability. It clear that a coefficient  $\dot{V}_{ABC} \leq \alpha V_{ABC}$  exists and can be found:

$$\dot{V}_{ABC}(\vec{q}_e, z_1, \tilde{L}) = -\vec{q}_e^T \Gamma_1 \vec{q}_e - z_1^T \Gamma_2 z_1 - \tilde{L}^T \Gamma_L^{-1} \tilde{L} \quad (4.72)$$

The assumption in Eq:4.71 is going to need to be tested in simulation later in Chapter:5; the adaptive gain matrix  $\Gamma_L$  is something that will similarly need to be designed. For control coefficients a separate optimization loop will be run, later disturbance will be introduced and the adaptive gain will independently be attained and optimized.

## 4.6 Position Control

Only two control laws for position control are proposed. Due to the nature of Coriolis cross-coupling, an attitude plant can be stabilized independently from the position plant, the converse is however not true. A basic Proportional-Derivative control structure is presented as the reference case, thereafter a more complicated adaptive backstepping control algorithm is derived...

The dynamics for position control, Eq:3.79b, include a coupled angular velocity element.

$$\dot{\vec{v}}_b = m^{-1}(-\vec{\omega}_b \times m\vec{v}_b + Q_b^* \otimes m\vec{G}_I \otimes Q_b + \mu\vec{F}) \in \mathcal{F}^b \quad (4.73)$$

Typically, given the standard operating conditions of a quadrotor, it's assumed that  $\vec{\omega}_b \approx \vec{0}$ . As such the inherent angular velocity coupled dynamics are negligible;  $\vec{\omega}_b \times m\vec{v}_b \approx 0$ . If the entire state vector, both attitude and position  $\mathbf{x}(t) = [\mathcal{E}, Q_b]^T$ , of the plant is known then it's easy to compensate for those dynamics rather than making assumptions about their influence on the system given particular operating conditions. That plant dependency can be introduced in the control force  $\mu\vec{F}$ .

The translational velocity,  $\vec{v}_b$ , defined in the body frame is related to the inertial frame through a quaternion transformation:

$$\dot{\mathcal{E}} = Q_b \otimes \vec{v}_b \otimes Q_b^* \in \mathcal{F}^I \quad (4.74)$$

The difference in reference frames is an important distinction between the position and attitude control loops. Position error is calculated purely as a subtractive term:

$$\mathcal{E}_e = \mathcal{E}_d - \mathcal{E}_b \in \mathcal{F}^I \quad (4.75)$$

With  $\mathcal{E}_d(t)$  being some desired position designed by the trajectory generation block. The translational velocity error can be similarly calculated but, in the same way angular velocity  $\vec{\omega}_d = \vec{0}$ , the desired translational velocity is zero.

$$\dot{\mathcal{E}}_e = \dot{\mathcal{E}}_d - \dot{\mathcal{E}}_b = -\dot{\mathcal{E}}_b \Big|_{\dot{\mathcal{E}}_d=\vec{0}} \quad (4.76)$$

The objective for position setpoint tracking is analogous to that of the attitude setpoint tracking. In particular the aim is to produce a stabilizing control law that ensures the position tracking error asymptotically tends to  $\vec{0}$ :

$$\mu\vec{F} = g(\mathcal{E}_e, \dot{\mathcal{E}}_e) \text{ such that } \lim_{t \rightarrow \infty} \mathcal{E}_e = \vec{0} \quad (4.77)$$

Where  $\mu\vec{F}$  is the control force to effect Eq:4.73  $\in \mathcal{F}^b$ .

#### 4.6.1 PD Controller

Starting with a simple PD structure to use as a reference case. A plant dependent controller designs the net force proportional to both the position error and the first derivative velocity error<sup>14</sup>.

$$\mu\vec{F}_{PD} = K\dot{\mathcal{E}}_e + \alpha\mathcal{E}_e + \vec{\omega}_b \times m\vec{v}_b - m\vec{G}_b \in \mathcal{F}^b \quad (4.78)$$

For the stability proof the error states must be transformed to the body frame  $\mathcal{F}^b$  such that the control input and error states all act in a common frame. So defining an error state in the body frame  $X_e$ :

$$X_e = Q_b^* \otimes (\mathcal{E}_d - \mathcal{E}_b) \otimes Q_b = X_d - X_b \quad (4.79a)$$

$$\dot{X}_e = Q_b^* \otimes (\dot{\mathcal{E}}_d - \dot{\mathcal{E}}_b) \otimes Q_b = -Q_b^* \otimes \dot{\mathcal{E}}_b \otimes Q_b = -\vec{v}_b \Big|_{\dot{\mathcal{E}}_d=\vec{0}} \quad (4.79b)$$

As such the control law from Eq:4.78, despite being  $\in \mathcal{F}^b$  has arguments  $\mathcal{E}, \dot{\mathcal{E}} \in \mathcal{F}^I$ , which must similarly transform to:

$$\mu\vec{F}_{PD} = K\dot{X}_e + \alpha X_e + \vec{\omega}_b \times m\vec{v}_b - m\vec{G}_b \quad (4.80a)$$

$$= -K\vec{v}_b + \alpha X_e + \vec{\omega} \times m\vec{v}_b - m\vec{G}_b \quad (4.80b)$$

Then using a p.d Lyapunov candidate function:

$$V_{PD}(X_e, \dot{X}_e) = \frac{\alpha}{2} X_e^T X_e + \frac{m}{2} \dot{X}_e^T \dot{X}_e = \frac{\alpha}{2} X_e^T X_e + \frac{m}{2} \vec{v}_b^T \vec{v}_b \quad (4.81)$$

<sup>14</sup>The same P and D coefficient symbols are used for continuity.

Then calculating the LCF derivative with the PD control law substituted:

$$\dot{V}_{PD} = \alpha X_e^T \dot{X}_e + \vec{v}_b^T m \dot{\vec{v}}_b = -\alpha X_e^T \vec{v}_b + \vec{v}_b^T m \dot{\vec{v}}_b \quad (4.82a)$$

$$= -\alpha X_e^T \vec{v}_b + \vec{v}_b^T (-\vec{\omega}_b \times m \vec{v}_b + m \vec{G}_b + \mu \vec{F}_{PD}) \quad (4.82b)$$

$$= -\alpha X_e^T \vec{v}_b + \vec{v}_b^T (-K \vec{v}_b + \alpha X_e) \quad (4.82c)$$

$$\Rightarrow \dot{V}_{PD} = -K \vec{v}_b^T \vec{v}_b < \vec{0} \quad \forall (X_e, \dot{X}_e) \quad (4.82d)$$

It then follows that the following global asymptotically<sup>15</sup> stabilizing limits exist:

$$\lim_{t \rightarrow \infty} X_e = Q_b^* \otimes (\mathcal{E}_d - \mathcal{E}_b) \otimes Q_b = \vec{0} \quad (4.83a)$$

$$\therefore \lim_{t \rightarrow \infty} \mathcal{E}_b = \mathcal{E}_d \quad (4.83b)$$

$$\lim_{t \rightarrow \infty} \dot{X}_e = Q_b^* \otimes (\dot{\mathcal{E}}_d - \dot{\mathcal{E}}_b) \otimes Q_b = -\vec{v}_b \Big|_{\dot{\mathcal{E}}_e=0} = 0 \quad (4.83c)$$

#### 4.6.2 Adaptive Backstepping Controller

An adaptive backstepping algorithm, similar the attitude controller derived previously in Sec:4.5.3, is now applied to position control. The disturbance term,  $\vec{D} \in \mathcal{F}^b$ , introduced to the differential Eq:4.73 represents any lumped drag and wind forces encountered by the body which weren't quantified numerically in Sec:3.3.3. The backstepping iterations of the position control loop first need to stabilize the position error and then compensate for those disturbances...

$$\dot{\vec{v}}_b = m^{-1} (-\vec{\omega}_b \times m \vec{v}_b + m \vec{G}_b + \vec{D}_b + \mu \vec{F}) \in \mathcal{F}^b \quad (4.84)$$

Obviously the compensation for  $\vec{D}$  is going to be an approximation of that physical disturbance term;  $\hat{D}$ . Beginning the backstepping process for position with the position state tracking error:

$$z_1 = \mathcal{E}_d - \mathcal{E}_b \quad (4.85)$$

Which then has its own derivative:

$$\dot{z}_1 = \dot{\mathcal{E}}_d - \dot{\mathcal{E}}_b = Q_b \otimes (\vec{0} - \vec{v}_b) \otimes Q_b^* = -Q_b \otimes \vec{v}_b \otimes Q_b^* \quad (4.86)$$

Transforming that error,  $z_1$ , to the body frame  $\mathcal{F}^b$ , in the same way as Eq:4.79a, makes the stability proof more concise. That reference frame transformation doesn't affect the Lie derivative as the energy function's gradient depends on its partial derivative w.r.t it's positional trajectory only, namely  $\mathcal{E}_e(t)$ .

$$\hat{z}_1 = X_e = Q_b^* \otimes z_1 \otimes Q_b = Q_b^* \otimes (\mathcal{E}_d - \mathcal{E}_b) \otimes Q_b \quad (4.87a)$$

$$\therefore \dot{\hat{z}}_1 = Q_b^* \otimes \dot{z}_1 \otimes Q_b = Q_b^* \otimes (\dot{\mathcal{E}}_d - \dot{\mathcal{E}}_b) \otimes Q_b = -\vec{v}_b \quad (4.87b)$$

Then proposing the first positive definite LCF,  $V_1(\hat{z}_1)$ , in terms of that tracking error:

$$V_1(\hat{z}_1) = \frac{1}{2} \hat{z}_1^T \hat{z}_1 \quad (4.88a)$$

$$\Rightarrow \dot{V}_1 = \hat{z}_1^T \dot{\hat{z}}_1 = -\hat{z}_1^T \vec{v}_b \Big|_{\dot{\mathcal{E}}_d=\vec{0}} \quad (4.88b)$$

The first stabilizing velocity function<sup>16</sup>,  $\Omega_d$ , and its associated error,  $\hat{z}_2$ , can be defined as:

$$\vec{v}_b \Rightarrow \Omega_d = \Gamma_1 \hat{z}_1 \quad (4.89a)$$

---

<sup>15</sup>Not exponentially stabilizing however.

<sup>16</sup>Using  $\Omega_d$  to differentiate from  $\vec{v}_d$  which would otherwise be the translational velocity produced from the desired trajectory...

$$\hat{z}_2 = \Omega_d - \vec{v}_b = \Gamma_1 \hat{z}_1 - \vec{v}_b \quad (4.89b)$$

$$\therefore \vec{v}_b = \Gamma_1 \hat{z}_1 - \hat{z}_2 \quad (4.89c)$$

So that second error state  $\hat{z}_2$  has a derivative:

$$\dot{\hat{z}}_2 = \dot{\Omega}_d - \dot{\vec{v}}_b = \Gamma_1 \dot{\hat{z}}_1 - m^{-1}(-\vec{\omega}_b \times m\vec{v}_b + m\vec{G}_b + \vec{D}_b + \mu\vec{F}) \quad (4.90a)$$

$$= -\Gamma_1 \vec{v}_b - m^{-1}(-\vec{\omega}_b \times m\vec{v}_b + m\vec{G}_b + \vec{D}_b + \mu\vec{F}) \quad (4.90b)$$

Introducing that second error  $\hat{z}_2$  into a new LCF  $V_2$ :

$$V_2(\hat{z}_1, \hat{z}_2) = V_1(\hat{z}_1) + \frac{1}{2} \hat{z}_2^T \hat{z}_2 = \frac{1}{2} \hat{z}_1^T \hat{z}_1 + \frac{1}{2} \hat{z}_2^T \hat{z}_2 \quad (4.91)$$

Which has a derivative:

$$\dot{V}_2 = \hat{z}_1^T \dot{\hat{z}}_1 + \hat{z}_2^T \dot{\hat{z}}_2 = -\hat{z}_1^T \vec{v}_b + \hat{z}_2^T \dot{\hat{z}}_2 \quad (4.92a)$$

$$= -\hat{z}_1^T \vec{v}_b + \hat{z}_2^T \left( -\Gamma_1 \vec{v}_b - m^{-1}(-\vec{\omega}_b \times m\vec{v}_b + m\vec{G}_b + \vec{D}_b + \mu\vec{F}) \right) \quad (4.92b)$$

And substituting Eq:4.89c for  $\vec{v}_b$  into only the first energy term of the LCF derivative. Specifically;  $-\hat{z}_1^T \vec{v}_b = -\hat{z}_1^T (\Gamma_1 \hat{z}_1 - \hat{z}_2)$ . The remaining terms for  $\vec{v}_b$  are left unchanged:

$$= -\hat{z}_1^T (\Gamma_1 \hat{z}_1 - \hat{z}_2) + \hat{z}_2^T \left( -\Gamma_1 \vec{v}_b - m^{-1}(-\vec{\omega}_b \times m\vec{v}_b + m\vec{G}_b + \vec{D}_b + \mu\vec{F}) \right) \quad (4.92c)$$

$$= -\hat{z}_1^T \Gamma_1 \hat{z}_1 + \hat{z}_2^T \left( -\hat{z}_1 - \Gamma_1 \vec{v}_b - m^{-1}(-\vec{\omega}_b \times m\vec{v}_b + m\vec{G}_b + \vec{D}_b + \mu\vec{F}) \right) \quad (4.92d)$$

An ideal backstepping control law, with the assumption that  $\vec{D}_b$  is precisely known, is then:

$$\mu\vec{F}_{IBC} = \vec{\omega}_b \times m\vec{v}_b - m\vec{G}_b - \vec{D}_b - m\hat{z}_1 - m\Gamma_1 \vec{v}_b + m\Gamma_2 \hat{z}_2 \quad (4.93a)$$

$$= \vec{\omega}_b \times m\vec{v}_b - m\vec{G}_b - \vec{D}_b + (\Gamma_1 \Gamma_2 - m)\hat{z}_1 - m(\Gamma_1 + \Gamma_2)\vec{v}_b \quad (4.93b)$$

$$\Rightarrow \dot{V}_{IBC} = \dot{V}_2 = -\hat{z}_1^T \Gamma_1 \hat{z}_1 - \hat{z}_2^T \Gamma_2 \hat{z}_2 < 0 \quad \forall (\hat{z}_1, \hat{z}_2) \& \forall (z_1, z_2) \quad (4.93c)$$

Which clearly leads to asymptotic (*extended to exponential next*) stability under the assumption that the disturbance term  $\vec{D}_b$  is known and can be compensated for well. In the controller both  $\Gamma_1$  &  $\Gamma_2$  are symmetric positive definite control coefficient matrices to be determined later...

Adjusting the backstepping rule and proposed LCF to incorporate an adaptive disturbance approximation term  $\hat{D}$ , similar to the adaptive backstepping attitude controller previously in Sec:4.5.3. That approximation leads to an estimation error  $\tilde{D}$ , once again assuming the physical disturbance dynamics  $\vec{D}_b$  are far slower than the control dynamics;  $\dot{\vec{D}}_b \ll \dot{\hat{D}}$ .

$$\tilde{D} = \vec{D}_b - \hat{D} \in \mathcal{F}^b \quad (4.94a)$$

$$\dot{\tilde{D}} = \dot{\vec{D}}_b - \dot{\hat{D}} \approx \vec{0} - \dot{\hat{D}} = -\dot{\hat{D}} \quad (4.94b)$$

$$\rightarrow \mu\vec{F}_{ABC} = \vec{\omega}_b \times m\vec{v}_b - m\vec{G}_b - \hat{D} - m\hat{z}_1 - m\Gamma_1 \vec{v}_d + m\Gamma_2 \hat{z}_2 \quad (4.94c)$$

$$= \vec{\omega}_b \times m\vec{v}_b - m\vec{G}_b - \hat{D} + (\Gamma_1 \Gamma_2 - m)\hat{z}_1 - m(\Gamma_1 + \Gamma_2)\vec{v}_b \quad (4.94d)$$

Then proposing an LCF which includes that disturbance estimate error  $\tilde{D}$  and finding its derivative:

$$V_{ABC} = \frac{1}{2} \hat{z}_1^T \hat{z}_1 + \frac{1}{2} \hat{z}_2^T \hat{z}_2 + \frac{1}{2} \tilde{D}^T \Gamma_D^{-1} \tilde{D} \quad (4.95a)$$

$$\Rightarrow \dot{V}_{ABC} = \hat{z}_1^T \dot{\hat{z}}_1 + \hat{z}_2^T \dot{\hat{z}}_2 + \tilde{D}^T \Gamma_D^{-1} \dot{\tilde{D}} \quad (4.95b)$$

$$= -\hat{z}_1^T \Gamma_1 \hat{z}_1 + \hat{z}_2^T \left( -\hat{z}_1 - \Gamma_1 \vec{v}_b - m^{-1}(-\vec{\omega}_b \times m\vec{v}_b + m\vec{G}_b + \vec{D}_b + \mu\vec{F}_{ABC}) \right) - \tilde{D} \Gamma_D^{-1} \dot{\tilde{D}} \quad (4.95c)$$

$$= -\hat{z}_1^T \Gamma_1 \hat{z}_1 + \hat{z}_2^T \left( -\Gamma_2 \hat{z}_2 - m^{-1} (\vec{D}_b - \hat{D}) \right) - \tilde{D}^T \Gamma_D^{-1} \dot{\hat{D}} \quad (4.95d)$$

$$= -\hat{z}_1^T \Gamma_1 \hat{z}_1 - \hat{z}_2^T \Gamma_2 \hat{z}_2 - \hat{z}_2^T m^{-1} \tilde{D} - \tilde{D}^T \Gamma_D^{-1} \dot{\hat{D}} \quad (4.95e)$$

$$= -\hat{z}_1^T \Gamma_1 \hat{z}_1 - \hat{z}_2^T \Gamma_2 \hat{z}_2 - \tilde{D}^T \Gamma_D^{-1} (m^{-1} \Gamma_D \hat{z}_2 + \dot{\hat{D}}) \quad (4.95f)$$

Then, a self-evident choice for the disturbance update law would be;  $\dot{\hat{D}} = -m^{-1} \Gamma_D \hat{z}_2$ , which would ensure asymptotic stability. Expanding on that, an interesting solution which could potentially enforce exponential stability would be to use  $\hat{D}_{n-1}$  as a disturbance estimate:

$$\dot{\hat{D}} = -m^{-1} \Gamma_D \hat{z}_2 + (\hat{D}_{n-1} - \hat{D}) \quad (4.96a)$$

$$\therefore \dot{V}_{ABC} = -\hat{z}_1^T \Gamma_1 \hat{z} - \hat{z}_2^T \Gamma_2 \hat{z}_2 - \tilde{D}^T \Gamma_D^{-1} (\hat{D}_{n-1} - \hat{D}_n) \quad (4.96b)$$

$$\approx -\hat{z}_1^T \Gamma_1 \hat{z}_1 - \hat{z}_2^T \Gamma_2 \hat{z}_2 - \tilde{D}^T \Gamma_D^{-1} \tilde{D} < \vec{0} \quad \forall (\hat{z}_1, \hat{z}_2, \tilde{D}) \quad (4.96c)$$

Similarly for the suggested exponentially stable adaptive backstepping controller for the attitude plant, using  $\hat{D}_{n-1}$  for an existing disturbance estimate will need to be simulated in order to ascertain if it's a suitable conjecture. Note here that adaptive control laws refer to adaptability to unknown disturbances and not plant model uncertainty which would take a different approach to the backstepping algorithm.

## 4.7 Controller Allocation

Following the higher level control laws, a distribution algorithm is needed to *allocate* out the desired virtual control inputs,  $\vec{\nu}_d = [\mu \vec{F} \ \mu \vec{r}]^T$ , to commanded actuator positions,  $u_c \in \mathbb{U}$ . The allocation block,  $B^\dagger(\mathbf{x}, \vec{\nu}_d, t)$ , from the control loop, Fig:4.1, constructs physical actuator positions from the virtual control input. For regular, unconstrained control allocation the solution is posed as an optimization<sup>17</sup> problem; aiming to minimize deviation between the virtual and commanded control inputs,  $\vec{\nu}_d$  &  $\vec{\nu}_c$  respectively.

$$\min_{u \in \mathbb{U}^{12}, s \in \mathbb{R}^6} (\|Q_s\|) \text{ such that } \vec{\nu}_d - \vec{\nu}_c = h(\mathbf{x}_e, t) - B(\mathbf{x}, t, u) = s \quad (4.97)$$

Where  $Q_s$  is some cost function to prioritize the slack variable,  $s$ , requirements. Typically that cost function will just be the  $L_2$  norm of the slack. In Eq:4.97 a generalized controller,  $h(\mathbf{x}_e, t)$ , is used; in the context of a 6-DOF control loop. That controller would combined virtual control inputs  $\mu \vec{F}$  and  $\mu \vec{r}$  for position and attitude control respectively. In the over-actuated case, there exists an entire family of suitable actuator positions  $u$  which are all solutions to Eq:4.97. The over-allocation solution is to introduce a secondary cost function<sup>18</sup>,  $J(\mathbf{x}, t, u)$ , into the optimization problem of Eq:4.97.

$$\min_{u \in \mathbb{U}^{12}, s \in \mathbb{R}^6} (\|Q_s\| + J(\mathbf{x}, u, t)) \text{ such that } h(\mathbf{x}_e, t) - B(\mathbf{x}, u, t) = s \quad (4.98)$$

That secondary control objective  $J(\mathbf{x}, t, u)$  and its associated *explicit* solution to Eq:4.98 is the subject of control allocation research. Not much work has been done on over-allocation for aerospace vehicles outside the field of satellite attitude control (Section:1.2.2 for examples). Often satellites are over actuated for the sake of fault tolerance and redundancy [6, 75]. Actuator rate constraints can be further introduced such that  $u$  is limited by  $\Delta u$ , constraining sequential actuator position changes.

$$\begin{aligned} \therefore \min_{u \in \mathbb{U}^{12}, s \in \mathbb{R}^6} (\|Q_s\| + J(\mathbf{x}, u, t)) \text{ s.t. } h(\mathbf{x}_e, t) - B(\mathbf{x}, u, t) = s \\ \text{subject to } u = u_{n-1} + \Delta u, \Delta u \in \mathbb{C} \end{aligned} \quad (4.99)$$

<sup>17</sup>Control Allocation by Johansen, et al. [2012] [60] and Control allocation for over actuated systems by Oppenheimer, et al. [2006] [90] both detail the nature of generalized nonlinear allocation loops.

<sup>18</sup>Or control objective.

Most control allocation paradigms assume a linear, multiplicative relationship with the effectiveness function, hence the abstraction layer which was introduced previously in Eq:4.7. The allocator effectiveness function, when abstracted to a linear matrix multiplication, reduces to:

$$\vec{\nu}_d = h(\mathbf{x}_e, t) \iff B(\mathbf{x}, u, t) = B(\mathbf{x}, t)u = \vec{\nu}_c \quad (4.100)$$

With  $\vec{\nu}_d$  &  $\vec{\nu}_c \in \mathbb{R}^n$ ,  $u \in \mathbb{U} \in \mathbb{R}^m$ ,  $B \in \mathbb{R}^{m \times n}$ . That assumption makes addressing the allocation conceptually simpler, accommodating the use of inversion based allocation laws (Sec:4.7.1,4.7.2,4.7.3). That abstraction to a multiplicative relationship with decomposed thrust components was suggested in Eq:4.7. The subsequent rotation "inversion" function,  $R^\dagger(\mathbf{x}, \vec{F}_i, t)$ , to solve for physical actuator positions  $(\Omega_i, \lambda_i, \alpha_i)$ , is however as yet undefined.

Assuming for the moment there is some allocation rule that, from  $\vec{\nu}_d$ , designs well four decomposed stabilizing 3-dimensional thrust vectors  $\vec{T}_{1 \rightarrow 4}$  to be produced by each motor module. It then follows that each of those four thrust vectors then relate<sup>19</sup> to their individual associated actuator positions through a quaternion *rotation*:

$$\vec{T}_i = Q_{M_i} \otimes \vec{T}(\Omega_i) \otimes Q_{M_i}^* \in \mathcal{F}^b \quad (4.101a)$$

$$= Q_z(\sigma_i)Q_y(\alpha_i)Q_x(\lambda_i) \otimes \vec{T}(\Omega_i) \otimes Q_x^*(\lambda_i)Q_y^*(\alpha_i)Q_z^*(\sigma_i) \quad (4.101b)$$

Where each motor thrust vector,  $\vec{T}(\Omega_i)$ , is:

$$\vec{T}(\Omega_i) = \begin{bmatrix} 0 \\ 0 \\ T(\Omega_i) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ C_T(J)\rho\Omega_i^2 D^4 \end{bmatrix} \in \mathcal{F}^{M_i} \quad (4.101c)$$

In the transformation Eq:4.101a the angle  $\sigma_i$  is an orthogonal  $\hat{Z}$  transformation from  $\mathcal{F}^b \rightarrow \mathcal{F}^{M_i''}$  from Eq:2.13a. The thrust function  $T(\Omega_i)$  is the BEM theory equation using thrust coefficients, Eq:3.81a, in the direction of the rotor shaft's axis of rotation, bound to  $\hat{Z}_{M_i}$ . Seeing that quaternion rotation (*transformation*) operators change the reference frame whilst retaining the vector operand's magnitude, it follows that  $T(\Omega_i)$ , and by extension the propeller speed  $\Omega_i$ , can be solved for:

$$|\vec{T}_i| = \sqrt{\|[T_x \ T_y \ T_z]\|} = \sqrt{T_x^2 + T_y^2 + T_z^2} = |T(\Omega_i)| = |C_T(J)\rho\Omega_i^2 D^4| \quad (4.102a)$$

$$\rightarrow \Omega_i = \sqrt{\frac{|\vec{T}_i|}{C_T(J)\rho D^4}} = \sqrt{\frac{\sqrt{T_x^2 + T_y^2 + T_z^2}}{C_T(J)\rho D^4}} \quad (4.102b)$$

Then reversing (or *undoing*) the transformation from motor module to body frame in Eq:4.101a:

$$\vec{T}(\Omega_i) = Q_z^*(\sigma_i)Q_y^*(\alpha_i)Q_x^*(\lambda_i) \otimes \vec{T}_i \otimes Q_x(\lambda_i)Q_y(\alpha_i)Q_z(\sigma_i) \in \mathcal{F}^{M_i} \quad (4.103a)$$

$$\rightarrow \vec{T}(\Omega_i) = Q_{M_i}^* \otimes \vec{T}_i \otimes Q_{M_i} \in \mathcal{F}^{M_i} \quad (4.103b)$$

Knowing only  $\vec{T}(\Omega_i)$  and  $\vec{T}_i$  in the motor frame and body frame respectively requires solving for a quaternion which relates the two. If both vectors are of unit length,  $\hat{T}_i$  &  $\hat{T}(\Omega_i)$ ; then the following relationship can be exploited to find a relative quaternion:

$$\hat{T}_i = \frac{\vec{T}_i}{|\vec{T}_i|} = \frac{\vec{T}_i}{\sqrt{T_x^2 + T_y^2 + T_z^2}} \in \mathcal{F}^b \quad (4.104a)$$

$$\hat{T}(\Omega_i) = \frac{\vec{T}(\Omega_i)}{|\vec{T}(\Omega_i)|} = \frac{\vec{T}(\Omega_i)}{|C_T(J)\rho\Omega^2 D^4|} = [0 \ 0 \ 1]^T \in \mathcal{F}^{M_i} \quad (4.104b)$$

$$Q_{M_i} = \begin{bmatrix} q_0 \\ \vec{q} \end{bmatrix} = \begin{bmatrix} 1 + \hat{T}_i \cdot \hat{T}(\Omega_i) \\ -\hat{T}_i \times \hat{T}(\Omega_i) \end{bmatrix} \quad (4.104c)$$

<sup>19</sup>Using the quaternion analogue of rotation  $\mathcal{F}^{M_i} \rightarrow \mathcal{F}^b$  from Eq:2.13a.

Where Eq:4.104c is an extension of the inherent quaternion definition which rotates a vector around a single Euler axis, Eq:3.22c, when applied to two unit vectors. That quaternion can indeed be used to solve for relative pitch, roll and yaw Euler angles (Appendix:A.4). The problem is that Eq:4.104c solves for the **most direct, shortest path** rotation from one vector to another. In most cases, a sequenced Z-Y-X rotation is by no means the shortest possible path. As a result solutions for  $[\phi, \theta, \psi]^T$  from Eq:4.104c won't be meaningful for trying to resolve suitable servo positions  $\lambda_i$  and  $\alpha_i$ .

The associated  $[\phi, \theta, \psi]^T$  solutions to Eq:A.18 are then of no consequence in trying to solve for sequence of rotation angles<sup>20</sup>  $[\lambda_i, \alpha_i, \sigma_i]^T$ . Furthermore, when considering a sequenced Z-Y-X quaternion, no further insight can be extracted without applying cumbersome trigonometric inversions;

$$Q_b = \begin{bmatrix} \cos \frac{\psi}{2} \\ 0 \\ 0 \\ \sin \frac{\psi}{2} \end{bmatrix} \otimes \begin{bmatrix} \cos \frac{\theta}{2} \\ 0 \\ \sin \frac{\theta}{2} \\ 0 \end{bmatrix} \otimes \begin{bmatrix} \cos \frac{\phi}{2} \\ \sin \frac{\phi}{2} \\ 0 \\ 0 \end{bmatrix} \quad (4.105a)$$

$$= \begin{bmatrix} c \frac{\psi}{2} c \frac{\theta}{2} c \frac{\phi}{2} + s \frac{\psi}{2} s \frac{\theta}{2} s \frac{\phi}{2} \\ c \frac{\psi}{2} c \frac{\theta}{2} s \frac{\phi}{2} - s \frac{\psi}{2} s \frac{\theta}{2} c \frac{\phi}{2} \\ c \frac{\psi}{2} s \frac{\theta}{2} c \frac{\phi}{2} + s \frac{\psi}{2} c \frac{\theta}{2} s \frac{\phi}{2} \\ s \frac{\psi}{2} c \frac{\theta}{2} c \frac{\phi}{2} - c \frac{\psi}{2} s \frac{\theta}{2} s \frac{\phi}{2} \end{bmatrix} = \begin{bmatrix} q_0 \\ q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} q_0 \\ \vec{q} \end{bmatrix} \quad (4.105b)$$

$$\rightarrow \vec{T}_i = \begin{bmatrix} c \frac{\psi}{2} c \frac{\theta}{2} c \frac{\phi}{2} + s \frac{\psi}{2} s \frac{\theta}{2} s \frac{\phi}{2} \\ c \frac{\psi}{2} c \frac{\theta}{2} s \frac{\phi}{2} - s \frac{\psi}{2} s \frac{\theta}{2} c \frac{\phi}{2} \\ c \frac{\psi}{2} s \frac{\theta}{2} c \frac{\phi}{2} + s \frac{\psi}{2} c \frac{\theta}{2} s \frac{\phi}{2} \\ s \frac{\psi}{2} c \frac{\theta}{2} c \frac{\phi}{2} - c \frac{\psi}{2} s \frac{\theta}{2} s \frac{\phi}{2} \end{bmatrix} \otimes \vec{T}(\Omega_i) \otimes \begin{bmatrix} s \frac{\psi}{2} s \frac{\theta}{2} s \frac{\phi}{2} + c \frac{\psi}{2} c \frac{\theta}{2} c \frac{\phi}{2} \\ s \frac{\psi}{2} s \frac{\theta}{2} c \frac{\phi}{2} - c \frac{\psi}{2} c \frac{\theta}{2} s \frac{\phi}{2} \\ -c \frac{\psi}{2} s \frac{\theta}{2} c \frac{\phi}{2} - s \frac{\psi}{2} c \frac{\theta}{2} s \frac{\phi}{2} \\ c \frac{\psi}{2} s \frac{\theta}{2} s \frac{\phi}{2} - s \frac{\psi}{2} c \frac{\theta}{2} c \frac{\phi}{2} \end{bmatrix} \quad (4.105c)$$

Instead; returning to rotation matrices for the inverse transformation and reiterating that Euler angle equivalents for the servos are;  $[\phi, \theta, \psi]^T \iff [\lambda_i, \alpha_i, \sigma_i]^T$ . It then follows (where  $i^{th}$  motor subscripts  $1 \rightarrow 4$  are implied):

$$\vec{T}_i = \begin{bmatrix} c\sigma & -s\sigma & 0 \\ s\sigma & c\sigma & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\alpha & 0 & s\alpha \\ 0 & 1 & 0 \\ -s\alpha & 0 & c\alpha \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\lambda & -s\lambda \\ 0 & s\lambda & c\lambda \end{bmatrix} \vec{T}(\Omega_i) \quad (4.106a)$$

$$\Rightarrow \vec{T}_i = \begin{bmatrix} c\sigma c\alpha & c\sigma s\alpha s\lambda - s\sigma c\lambda & c\sigma s\alpha c\lambda + s\sigma s\lambda \\ s\sigma c\alpha & s\sigma s\alpha s\lambda + c\sigma c\lambda & s\sigma s\alpha c\lambda - c\sigma s\lambda \\ -s\alpha & cas\lambda & cac\lambda \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ T(\Omega_i) \end{bmatrix} \quad (4.106b)$$

$$\Rightarrow \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} = \begin{bmatrix} s\sigma s\lambda + c\sigma s\alpha c\lambda \\ s\sigma s\alpha c\lambda - c\sigma s\alpha \\ cac\lambda \end{bmatrix} T(\Omega_i) \quad (4.106c)$$

Where  $\sigma$  is an orthogonal multiple which rotates the vector about the  $\hat{Z}_b$  axis. The fact that the principle thrust vector  $\vec{T}(\Omega_i)$  has only a  $\hat{Z}_{M_i}$  component in the motor frame makes the solution for servo angles dramatically less complex to solve. Then Eq:4.106c simplifies even further to the following four trigonometric relations respectively for each motor module:

$$\begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} = \left[ \begin{bmatrix} sac\lambda \\ -s\lambda \\ cac\lambda \end{bmatrix}, \begin{bmatrix} s\lambda \\ sac\lambda \\ cac\lambda \end{bmatrix}, \begin{bmatrix} -sac\lambda \\ s\lambda \\ cac\lambda \end{bmatrix}, \begin{bmatrix} -s\lambda \\ -sac\lambda \\ cac\lambda \end{bmatrix} \right] T(\Omega_i) \quad \text{for } i \in [1, 2, 3, 4] \quad (4.107)$$

It then becomes a simple case of inverse trigonometry to solve for both  $\lambda_i$  and  $\alpha_i$  respectively, for the example case of  $i = 1$ , the following holds true and can be similarly applied to the remaining motor modules.

<sup>20</sup> $\sigma_i$  is already known to be an orthogonal multiplicate...

Using  $T(\Omega_i) = \|\vec{T}_i\|$  and the four quadrant secondary arctangent2 function<sup>21</sup>,  $\text{arctan2}(x, y)$ , for both inversion solutions to get full quadrature<sup>22</sup> results:

$$\lambda = \text{arctan2}\left(-T_y, \sqrt{\|\vec{T}_i\|^2 - T_y^2}\right) \quad (4.108a)$$

$$\alpha_i = \text{arctan2}(T_x, T_z) \quad (4.108b)$$

Therefore, the secondary component of the control allocation block,  $R^\dagger(\mathbf{x}, \vec{T}_i, t)$  from Fig:4.2 is then summarized as a single rotation inversion function (for motor module  $i = 1$ ):

$$\begin{bmatrix} \Omega_i \\ \lambda_i \\ \alpha_i \end{bmatrix} = R^\dagger(\mathbf{x}, \vec{T}_i, t) = \begin{bmatrix} \left(\sqrt{T_x^2 + T_y^2 + T_z^2}/C_T(J)\rho D^4\right)^{\frac{1}{2}} \\ \text{atan2}(T_x^2, \|\vec{T}_i\| \sqrt{\|\vec{T}_i\|^2 - T_x^2}) \\ -\text{atan2}(T_x, T_z \|\vec{T}_i\|) \end{bmatrix} \quad (4.109)$$

Further simplifications could be drawn from the definitions of each element included in Eq:4.109, but it would just be superfluous as each servo angle can be solved for. At this stage the only remaining undefined component of the entire control block is the abstracted allocation algorithms,  $B^\dagger(\mathbf{x}, \vec{\nu}_d, t)$ , which are now addressed...

#### 4.7.1 Pseudo Inverse Allocator

Conceptually the simplest control allocation solutions to Eq:4.98 stems from what are categorized as "inversion" based cost optimizations (the first three proposed allocators in Sec:4.7.1, 4.7.2 & 4.7.3). As alluded to previously, the requirements for inversion based allocation is that the effectiveness function  $B(\mathbf{x}, u, t)$  is a linear relationship which can be abstracted to  $B(\mathbf{x}, t)u$ . The objective of inversion is that for the control problem  $\vec{\nu}_c = B(\mathbf{x}, t)u$  to find some matrix  $B^\dagger(\mathbf{x}, t)$  such that for some  $\vec{\nu}_d$ :

$$\vec{\nu}_c = \vec{\nu}_d = B(\mathbf{x}, t)u \Rightarrow B^\dagger(\mathbf{x}, t)\vec{\nu}_d = B^\dagger(\mathbf{x}, t)B(\mathbf{x}, t)u \quad (4.110a)$$

With the inversion identity:

$$B(\mathbf{x}, t)B^\dagger(\mathbf{x}, t) = \mathbb{I}_{m \times m} \quad (4.110b)$$

$$\rightarrow u = B^\dagger(\mathbf{x}, t)\vec{\nu}_d \quad (4.110c)$$

Or more generally, without the dependency of linearity:

$$u = B^\dagger(\mathbf{x}, \vec{\nu}_d, t) \quad (4.110d)$$

Where  $B(\mathbf{x}, t) \in \mathbb{R}^{m \times n}$ . When the  $B$  matrix has full rank; that being  $m > n$ , the inversion of  $B^\dagger$ , short of online iterative techniques to solve for the inversion, is not so trivial. A linear least squares technique is the basis of the direct inversion allocation schemes. The secondary control objective,  $J(\mathbf{x}, u, t)$ , is chosen to be a quadratic cost function that can be solved as an explicit least squares problem. The net effect of which aims to minimize controller effort (*magnitude*), such that:

$$J(\mathbf{x}, u, t) = \min_{u \in \mathbb{U}} \frac{1}{2} (u - u_p)^T W (u - u_p) \text{ such that } \vec{\nu}_c = B(\mathbf{x}, u, t) \quad (4.111)$$

Its worth mentioning that Eq:4.111 has no slack variable to be optimized, unlike Eq:4.98. The matrix  $W$  is a positive definite matrix of weighting<sup>23</sup> coefficients that prioritises different actuators in the actuator matrix  $u$  higher than others. Similarly  $u_p$  is the preferred actuator position matrix<sup>24</sup> which the system tends towards.

<sup>21</sup>More on the  $\text{arctan2}$  function in Appendix:A.4.

<sup>22</sup>Exploiting the fact that  $\text{arctan}(x) = \text{arcsin}(x/\sqrt{1-x^2})$ .

<sup>23</sup>Discussed in full next in Sec:4.7.2.

<sup>24</sup>Priority acutator positions are detailed in Sec:4.7.3.

The least squares solution [42] to Eq:4.111 for that inversion matrix  $B^\dagger(\mathbf{x}, t)$  is then:

$$u \in \mathbb{U} = (\mathbb{I} - CB(\mathbf{x}, t))u_p + C\vec{\nu}_d \quad (4.112a)$$

$$C = W^{-1}B^T(\mathbf{x}, t)(B(\mathbf{x}, t)W^{-1}B^T(\mathbf{x}, t))^{-1} \quad (4.112b)$$

The solution in Eq:4.112 is referred to as the generalized inverse with weighted and preferred actuators positions. In the special case where there are no weightings,  $W = \mathbb{I}_{n \times n}$ , and no preferred actuator values are given,  $u_p = \vec{0}$ , the solution reduces to:

$$u = B^T(\mathbf{x}, t)(B(\mathbf{x}, t).B^T(\mathbf{x}, t))^{-1}\vec{\nu}_d \quad (4.113a)$$

$$= B^\dagger(\mathbf{x}, t)\vec{\nu}_d \quad (4.113b)$$

The simplified unique case of Eq:4.113 is termed a Moore-Penrose or pseudo-inversion of the actuator effectiveness matrix  $B(\mathbf{x}, t)$ . The pseudo-inversion is the most basic of allocation techniques, with a least squares minimization of controller effort. For an effectiveness  $B(\mathbf{x}, t)$  matrix from Eq:4.8 relating to the layout described in Fig:2.9, the pseudo-inversion allocator is:

$$B(\mathbf{x}, t) = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & L & 0 & 0 & 0 & 0 & 0 & -L \\ 0 & 0 & -L & 0 & 0 & 0 & 0 & 0 & L & 0 & 0 & 0 \\ 0 & L & 0 & -L & 0 & 0 & 0 & -L & 0 & L & 0 & 0 \end{bmatrix} \in \mathbb{R}^{12 \times 6} \quad (4.114a)$$

$$\Rightarrow B^\dagger(\mathbf{x}, t) = B^T(B.B^T)^{-1} \quad (4.114b)$$

$$= \begin{bmatrix} \frac{1}{4} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{4} & 0 & 0 & 0 & \frac{1}{4L} \\ 0 & 0 & \frac{1}{4} & 0 & \frac{-1}{2L} & 0 \\ \frac{1}{4} & 0 & 0 & 0 & 0 & \frac{-1}{4L} \\ 0 & \frac{1}{4} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{4} & \frac{1}{2L} & 0 & 0 \\ \frac{1}{4} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{4} & 0 & 0 & 0 & \frac{-1}{4L} \\ 0 & 0 & \frac{1}{4} & 0 & \frac{1}{2L} & 0 \\ \frac{1}{4} & 0 & 0 & 0 & 0 & \frac{1}{4L} \\ 0 & \frac{1}{4} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{4} & \frac{-1}{2L} & 0 & 0 \end{bmatrix} \in \mathbb{R}^{6 \times 12} \quad (4.114c)$$

Such that the pseudo-inversion allocation rule  $u = B^\dagger(\mathbf{x}, t)\vec{\nu}_d$  produces a feasible set of control thrust vectors,  $\vec{T}_{1 \rightarrow 4}$ , for some virtual control input  $\vec{\nu}_d = h(\mathbf{x}_e, t)$ . Those thrust vectors, numbered  $1 \rightarrow 4$ , are then solved for as explicit actuator positions  $[\Omega_i, \lambda_i, \alpha_i]^T = R^\dagger(\mathbf{x}, \vec{T}_i, t)$  to construct the actuator matrix  $u \in \mathbb{U} \in \mathbb{R}^{12}$ . Noting that  $B(\mathbf{x}, t)$  does not necessarily have to be static with respect to either the state vector  $\mathbf{x}$  or time  $t$ . The allocation rule in Eq:4.114 is the most simplified case of the least squares quadratically optimized equation for Eq:4.98 and is used as the base reference allocation law to which all other proposed rules are compared against.

The direct (*pseudo*) inversion solution guarantees the commanded virtual control input is met and that actuators aren't necessarily saturated. In certain cases it may be desired to completely saturate certain actuators before exploiting other actuator plant inputs. That would entail an iterative "daisy chaining" [60] allocation to be performed numerically online, enforcing saturation for atleast some actuators and achievement of control objectives. That particular approach is avoided here as completely saturating an actuator isn't desirable; in the context of thrust generation (or vectoring) with propeller's saturation is something to be avoided...

### 4.7.2 Weighted Pseudo Inverse Allocator

A weighted inversion still treats the preferred actuator positions as negligible, or that  $u_p = \vec{0}$  in Eq:4.112, but adds priority to different actuators in the form of a  $W$  matrix. The positive definite (*symmetrical*) weighting matrix is square with respect to the actuator dimension,  $W \in \mathbb{R}^{12 \times 12}$  (*or more generally*  $W \in \mathbb{R}^{m \times m}$ ). The Moore-Penrose inversion assumes that all actuators are equally weighted and purely diagonal,  $W = \mathbb{I}$ . A time dependent adaptive weighting matrix could prioritize actuators following control faults or actuator deterioration. The control objective of a weighted inversion is to design the explicit weighting coefficients as per some preferred heuristic or optimization<sup>25</sup>.

Each weighting coefficient determines how the least squares solution to Eq:4.98 preferentially biases a particular actuator, in this case the weighting matrix's divisions correlate to mixed actuator thrust vector values. The  $3 \times 3$  diagonal groupings  $W_{1 \rightarrow 4}$  relate to individual thrust component biasing ( $T_{ix}, T_{iy}, T_{iz}$ ) whilst off-centre  $3 \times 3$  groupings mix separate thrust terms  $\vec{T}_{1 \rightarrow 4}$ .

Pseudo-inversion, previously, will exactly match the virtual control input  $\vec{\nu}_d = B(\mathbf{x}, u, t) = \vec{\nu}_c$  so long as the actuators aren't saturated. Biasing actuators with an explicit weighting matrix could otherwise introduce a slack between the desired control requirements and their commanded counterparts. Such a case could result in instability given that trajectory tracking is stabilized through Lyapunov's theorem in the design of  $\vec{\nu}_d$ , not solving for allocated actuator positions. Short of iteratively<sup>26</sup> processing variable weights until a viable solution is found, a constraint on the nature of the weighting matrix needs to be introduced.

$$\begin{aligned} \vec{T}_1 &\Downarrow & \vec{T}_2 &\Downarrow & \vec{T}_3 &\Downarrow & \vec{T}_4 &\Downarrow \\ \vec{T}_1 &\Rightarrow \begin{bmatrix} W_{1:1}W_{1:2}W_{1:3} \\ W_{1:4}W_{1:5}W_{1:6} \\ W_{1:7}W_{1:8}W_{1:9} \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} W_{5:1}W_{5:2}W_{5:3} \\ W_{5:4}W_{5:5}W_{5:6} \\ W_{5:7}W_{5:8}W_{5:9} \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ \vec{T}_2 &\Rightarrow \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} W_{2:1}W_{2:2}W_{2:3} \\ W_{2:4}W_{2:5}W_{2:6} \\ W_{2:7}W_{2:8}W_{2:9} \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} W_{6:1}W_{6:2}W_{6:3} \\ W_{6:4}W_{6:5}W_{6:6} \\ W_{6:7}W_{6:8}W_{6:9} \end{bmatrix} \\ \vec{T}_3 &\Rightarrow \begin{bmatrix} W_{5:1}W_{5:2}W_{5:3} \\ W_{5:4}W_{5:5}W_{5:6} \\ W_{5:7}W_{5:8}W_{5:9} \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} W_{3:1}W_{3:2}W_{3:3} \\ W_{3:4}W_{3:5}W_{3:6} \\ W_{3:7}W_{3:8}W_{3:9} \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ \vec{T}_4 &\Rightarrow \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} W_{6:1}W_{6:2}W_{6:3} \\ W_{6:4}W_{6:5}W_{6:6} \\ W_{6:7}W_{6:8}W_{6:9} \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} W_{4:1}W_{4:2}W_{4:3} \\ W_{4:4}W_{4:5}W_{4:6} \\ W_{4:7}W_{4:8}W_{4:9} \end{bmatrix} \end{aligned}$$

**Figure 4.3:** Weighting matrix biasing

So long each horizontal and vertical weighting groups contributing to each thrust vector,  $W_{T_i} \in \mathbb{R}^{3 \times 12}$ , each have a unit norm, the designed control torque and force inputs will be met. Physically the resultant thrusts and torque (thrust differentials) would be balanced amongst similarly directed components. Furthermore, an additional restraint is that only permissible thrust vector mixings are between opposing pairs;  $\vec{T}_1 \& \vec{T}_3$  and  $\vec{T}_2 \& \vec{T}_4$ . Such a constraint simplifies the time spent optimizing weighting coefficients in Sec:5.2.

<sup>25</sup>Not considered or discussed are adaptive weightings as those are out of the scope of this work and pertain more to FTC [6].

<sup>26</sup>Online iterative solutions are avoided given their increased computational complexity and the possibility that, given an infinite processing time, a solution may not necessarily be found.

The physical consequences of giving priority biasing to thrust vector components in the  $\hat{X}_b$  &  $\hat{Y}_b$ <sup>27</sup> directions is that the allocation block prioritizes using pitch or roll servos,  $\lambda_i$  &  $\alpha_i$  respectively, before changing the propeller's rotational speed  $\Omega_i$ . Similarly balancing the off-diagonal thrust vector mixing blends controller effort amongst opposing actuators.

The explicit weighting coefficients are to be optimized iteratively in simulation, Sec:5.2; aiming to minimize some performance metric. That metric, which evaluates relative performance of a proposed set of weighting coefficients, is penalized<sup>28</sup> from actuator slew rate times and a slack variable norm;

$$\int (a \|t^{\nu_d - \nu_c} - 1\| + b \|s\|) dt \quad (4.115)$$

Where the integral is run until  $t \rightarrow \infty$  over the length of a single simulation cycle. As such, the weighting matrix coefficients try to reduce the transient time taken for the actuator block to settle whilst ensuring stability isn't compromised. Optimization iterations for the weight coefficients are completely independent from the controller coefficient loops to be run in Sec:5.1...

#### 4.7.3 Priority Norm Inverse Allocator

The last allocator based on typical inversion applies a non-zero preferred actuator position from Eq:4.98; or that  $u_p \neq \vec{0} \in \mathbb{U}$ . The preferred actuator position is the matrix value of  $u$  which the allocator naturally tends toward. An obvious choice for that value are the conditions required for stable hovering, those which simply keep the quadcopter airborne. There are however some intricacies which must be discussed with respect to what hovering conditions are.

For a generalized body of weight  $m$ , a net gravitational force opposes upward movement in the inertial frame;  $\vec{M} = [0, 0, -G.m]^T \in \mathcal{F}^I$ . At the hover state there are no net forces or torques<sup>29</sup> acting on the system, all dynamics are balanced. As such the hovering conditions are then simply:

$$\begin{bmatrix} \mu \vec{F}_p \\ \mu \vec{\tau}_p \end{bmatrix} = \begin{bmatrix} \vec{M} \\ \vec{0} \end{bmatrix} \in \mathcal{F}^I \quad (4.116)$$

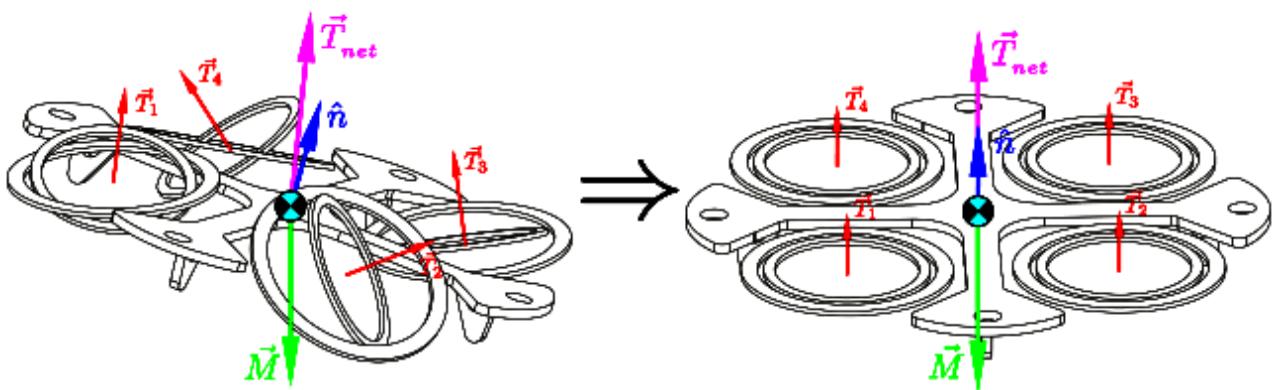


Figure 4.4: Hover conditions W.R.T the inertial frame  $\mathcal{F}^I$

<sup>27</sup>Recalling that the allocator block designs  $\vec{T}_{1-4}$  in the body frame,  $\in \mathcal{F}^b$ . Then the rotation inversion block  $R^\dagger(\mathbf{x}, \vec{T}_i, t)$  from Eq:4.109 finds  $(\Omega_i, \lambda_i, \alpha_i)$  to transform  $\vec{T}(\Omega_i)$  to the body frame; effectively mapping  $\mathcal{F}^{M_i} \rightarrow \mathcal{F}^b$ .

<sup>28</sup>More on simulations and optimizations next in Chapter:5-Simulations & Results.

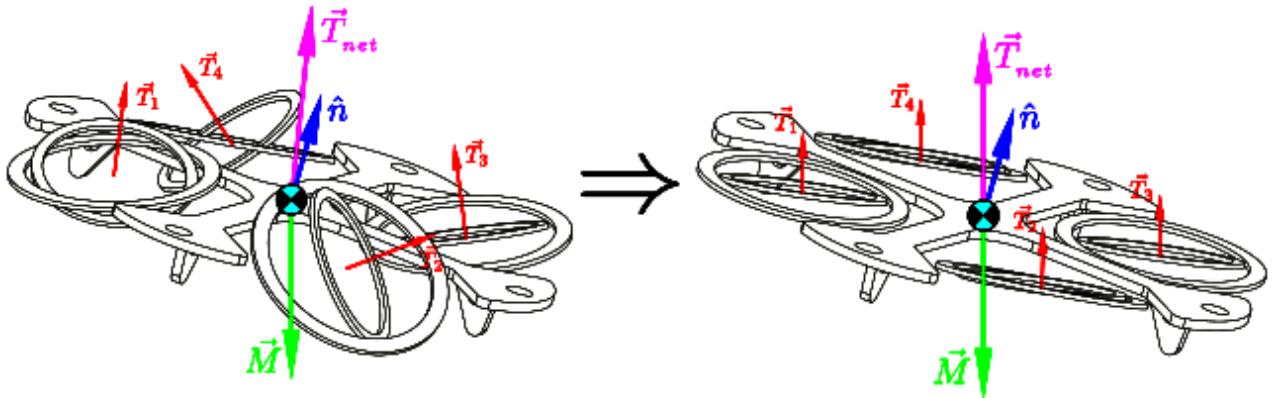
<sup>29</sup>Unwanted system dynamics like torques from an eccentric gravitational center or disturbances are compensated for in a plant dependent control law  $\mu \vec{\tau} = h(\mathbf{x}_e, t)$ .

However, calculating hover conditions purely in the inertial frame give no indication on what attitude the body has. Two options present themselves on how to solve for hover values. First; take hover conditions with respect to the inertial frame, such that the body's attitude tends to the origin always. The free body diagram in Fig:4.4 illustrates this.

$$\vec{\nu}_I = \begin{bmatrix} \mu \vec{F}_p \\ \mu \vec{\tau}_p \end{bmatrix} = \begin{bmatrix} \vec{M} \\ \vec{0} \end{bmatrix} \in \mathcal{F}^b \quad (4.117)$$

Conversely, the second option is to take hover conditions with respect to the body frame (Fig:4.5). The difference is that the body's preferred actuator positions are dependent on each instantaneous orientation. That attitude stays constant whilst the actuators are redirected to produce inertial hovering conditions; irrespective of the attitude. The preferred hovering conditions are then always dependent on the commanded attitude trajectory.

$$\vec{\nu}_b = \begin{bmatrix} \mu \vec{F}_p \\ \mu \vec{\tau}_p \end{bmatrix} = \begin{bmatrix} Q_b^* \otimes \vec{M} \otimes Q_b \\ \vec{0} \end{bmatrix} \in \mathcal{F}^b \quad (4.118)$$



**Figure 4.5:** Hover conditions W.R.T the body frame  $\mathcal{F}^b$

Explicit actuator positions are then solved for Eq:4.117 & Eq:4.118 using pseudo inversion from Eq:4.113. The two solutions are then as follows:

$$u_p^I = R^\dagger(\mathbf{x}, (B^\dagger(\mathbf{x}, \vec{\nu}_I, t)), t) \quad (4.119a)$$

$$u_p^b = R^\dagger(\mathbf{x}, (B^\dagger(\mathbf{x}, \vec{\nu}_b, t)), t) \quad (4.119b)$$

Where the inverse rotation operator,  $R^\dagger$  from Eq:4.119, is applied to all four thrust vectors produced by the allocation operator  $B^\dagger$ . Both actuator matrices are then applied to Eq:4.112 and could be combined with a non-diagonal weighting matrix.

$$u_{\in \mathbb{U}} = (\mathbb{I} - CB(\mathbf{x}, t))u_p + C\vec{\nu}_d \quad (4.120a)$$

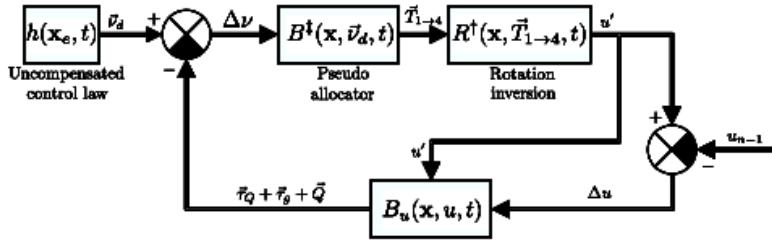
$$C = W^{-1}B^T(\mathbf{x}, t)(B(\mathbf{x}, t)W^{-1}B^T(\mathbf{x}, t))^{-1} \quad (4.120b)$$

The physical consequences of either preferred actuator positions are demonstrated in simulation in Sec:5.4.3. Priority actuator positions aren't simulated together with weighting matrices, the two are compared independently...

#### 4.7.4 Non-linear Plant Control Allocation

Despite the added actuation, each complex dynamic response from an actuator excitation is not fully exploited. The dynamics of an actuator's motor module, Sec:3.2.1, until now has been treated as an element to be compensated for in feedback structure. An alternative approach, seen in Gasco, et al. [2012] [2, 38], is to use the actuator reactions as additional non-linear actuator plants. In [2, 38] the actuator plants and their resultant dynamics were introduced as additional dimensions to the actuator matrix  $u \in \mathbb{U}$ .

Such an approach was achievable because the authors, despite adding two extra degrees of freedom for each propeller, hadn't vectored the propeller thrust. The non-linear proposal here is to first calculate a Pseudo-inversion actuator solution *without* plant compensation<sup>30</sup>, then introducing those induced actuator responses from such an excitation to alleviate the control plant requirement. A subsequent revised virtual control plant input is used iteratively to find a subsequent pseudo-inversion solution; the process is cycled until the control requirements are met.



**Figure 4.6:** Allocation loop iteration

In Fig:4.6 the iteration loop is shown, each iteration is run online and settles at a balance point. In the loop, the block  $B_u(\mathbf{x}, u, t)$  is a combination of non-linear actuator response terms from Eq:3.82,3.83 & 3.81b; those being  $\vec{\tau}_Q$ ,  $\vec{\tau}_g$  &  $\vec{Q}$  respectively. The settling point, where possible, a portion of the commanded control input  $\vec{v}_d$  is achieved from the otherwise compensated for actuator response dynamics.

#### 4.7.5 Online Optimized Secondary Goal Allocator

<sup>30</sup>Disregarding  $\vec{\tau}_Q$ ,  $\vec{\tau}_g$  &  $\vec{Q}$ .



# 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

#### 5.1.5 PD

#### 5.1.6 XPD

#### 5.1.7 IBC

#### 5.1.8 ABC

### 5.2 Allocator Optimization

### 5.3 Simulation Block

### 5.4 Optimized Controller Results

#### 5.4.1 Attitude Control

#### 5.4.2 Position Control

#### 5.4.3 Allocator Performance

### 5.5 State Estimation

#### 5.5.1 Attitude Control Results

#### 5.5.2 Attitude Outputs

## Chapter 6

# Prototype Flight Results

# Chapter 7

## Conclusion

- Lagrange dynamics for multibody system could have produced a more concise model etc . . .
- Particle multibody dynamics with interactions could provide a more verbose simulation environment rather than the very newtonian simulation loop constructed
- Implicit equation dynamics in simulation may improve optimization loops in PSO algorithm - Firmware changes to ESC drastically improved transfer function time constant, made assumption that servos would improve actuator response times redundant.

# Appendix A

## Expanded Equations

### A.1 Standard Quadrotor Dynamics

Following the 6-DOF fundamental derivations in Section 3.1.1, the common reductions typically applied for a generic "+" configuration quadrotor are now presented. Reiterating those four differential equations, Eq:3.9, which describe a rigid body's motion (using rotation matrices and not quaternions):

$$\dot{\vec{\mathcal{E}}} = \mathbb{R}_b^I(-\eta) \vec{v}_b \quad \in \mathcal{F}^I \quad (\text{A.1a})$$

$$\dot{\vec{v}}_b = m^{-1} [ -\vec{\omega}_b \times m\vec{v} + m\mathbb{R}_I^b(-\eta) \vec{G}_I + \vec{F}_{net} ] \quad \in \mathcal{F}^b \quad (\text{A.1b})$$

$$\dot{\vec{\eta}} = \Psi(\eta) \vec{\omega}_b \quad \in \mathcal{F}^{v2}, \mathcal{F}^{v1}, \mathcal{F}^I \quad (\text{A.1c})$$

$$\dot{\vec{\omega}}_b = \mathbb{I}_b^{-1} [ -\vec{\omega}_b \times \mathbb{I}_b \vec{\omega}_b + \vec{\tau}_{net} ] \quad \in \mathcal{F}^b \quad (\text{A.1d})$$

With the Euler matrix,  $\Psi(\eta)$ , defined in Eq:2.9f. The net lift force produced by motors  $i = [1 : 4]$ , bound perpendicularly to the  $\hat{Z}_b$  axis, is given by:

$$\vec{T} = \sum_{i=1}^4 F_i \hat{k} \quad \in \mathcal{F}^b \quad (\text{A.2a})$$

The simplified relationship between the thrust force  $\vec{F}_i$  and the propellers rotational speed  $\Omega_i$  is approximately quadratic:

$$\vec{F}_i = k_1 \Omega_i^2 \quad (\text{A.2b})$$

Similarly the aerodynamic torque opposing each rotating propeller is:

$$Q = k_2 \Omega_i^2 \quad (\text{A.3})$$

Coefficients  $k_1$  &  $k_2$  are typically determined from physical test results. The control pitch and roll torques,  $\tau_\phi$  &  $\tau_\theta$  respectively, are generated by opposing differential lift forces. Lastly the yaw torque is generated as a net response to the rotational aerodynamic propeller torques. The control torque inputs are then defined as:

$$\tau_\phi = \vec{L}_{arm}(\vec{F}_1 - \vec{F}_3) \quad (\text{A.4a})$$

$$\tau_\theta = \vec{L}_{arm}(\vec{F}_2 - \vec{F}_4) \quad (\text{A.4b})$$

$$\tau_\psi = \sum_{i=1}^4 (-1)^i k_2 \Omega_i \quad (\text{A.4c})$$

Then expanding the linear translational position and attitude differentials, Eq:A.1b & Eq:A.1d, to their component forms:

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} = \begin{pmatrix} rv - qw \\ pw - ru \\ qu - pv \end{pmatrix} + \begin{pmatrix} -g\sin(\theta) \\ g\cos(\theta)\sin(\phi) \\ g\cos(\theta)\cos(\phi) \end{pmatrix} + \frac{1}{m} \begin{pmatrix} 0 \\ 0 \\ T \end{pmatrix} \in \mathcal{F}^b \quad (\text{A.5a})$$

$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} \frac{\mathbb{I}_y - \mathbb{I}_z}{\mathbb{I}_x} qr \\ \frac{\mathbb{I}_z - \mathbb{I}_x}{\mathbb{I}_y} pr \\ \frac{\mathbb{I}_x - \mathbb{I}_y}{\mathbb{I}_z} pq \end{pmatrix} + \mathbb{I}_b^{-1} \begin{pmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{pmatrix} \in \mathcal{F}^b \quad (\text{A.5b})$$

Considering the size of the average angular velocity  $\omega_b \approx \vec{0}$ , the gyroscopic and Coriolis effects on the body (namely both cross product terms) are depreciated and can be regarded as negligible. Assuming too that the body has a (*roughly*) diagonal inertial matrix. Then:

$$\begin{pmatrix} rv - qw \\ pw - ru \\ qu - pv \end{pmatrix} \approx \vec{0} \quad \text{and} \quad \begin{pmatrix} \frac{\mathbb{I}_y - \mathbb{I}_z}{\mathbb{I}_x} qr \\ \frac{\mathbb{I}_z - \mathbb{I}_x}{\mathbb{I}_y} pr \\ \frac{\mathbb{I}_x - \mathbb{I}_y}{\mathbb{I}_z} pq \end{pmatrix} \approx \vec{0} \quad (\text{A.6})$$

As a result, equations from Eq:A.5 then simplify to:

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} = \begin{pmatrix} -g\sin(\theta) \\ g\cos(\theta)\sin(\phi) \\ g\cos(\theta)\cos(\phi) \end{pmatrix} + \frac{1}{m} \begin{pmatrix} 0 \\ 0 \\ T \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} \frac{1}{\mathbb{I}_x} \tau_\phi \\ \frac{1}{\mathbb{I}_y} \tau_\theta \\ \frac{1}{\mathbb{I}_z} \tau_\psi \end{pmatrix} \quad (\text{A.7})$$

Similarly, around the origin and at hovering conditions,  $\Psi(\eta) \approx \vec{1}$  for  $\eta \approx \vec{0}$  and so from Eq:A.1c, then  $\dot{\eta} \approx \omega_b$ . Or in component form:

$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} \approx \begin{pmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{pmatrix} \quad (\text{A.8})$$

As such, the differential equations Eq:A.5 are then simplified to the following six SISO controllable plants:

$$\ddot{x} = (-\cos(\phi)\sin(\theta)\cos(\psi) - \sin(\phi)\sin(\psi)) \frac{1}{m} T \quad (\text{A.9a})$$

$$\ddot{y} = (-\cos(\phi)\sin(\theta)\sin(\psi) + \sin(\phi)\cos(\psi)) \frac{1}{m} T \quad (\text{A.9b})$$

$$\ddot{z} = g - (\cos(\phi)\cos(\theta)) \frac{1}{m} T \quad (\text{A.9c})$$

$$\ddot{\phi} = \frac{1}{\mathbb{I}_x} \tau_\phi \quad (\text{A.9d})$$

$$\ddot{\theta} = \frac{1}{\mathbb{I}_y} \tau_\theta \quad (\text{A.9e})$$

$$\ddot{\psi} = \frac{1}{\mathbb{I}_z} \tau_\psi \quad (\text{A.9f})$$

Typically, the simplified states in Eq:A.9 are abstracted to what is termed as an "augmented pilot control system". As such the controllable inputs are abstracted to  $T$ ,  $\dot{\phi}$ ,  $\dot{\theta}$ ,  $\dot{\psi}$ . Wherein the pilot can dictate the attitude and net heave thrust of the quadrotor, mostly with various flavours of PID control.

## A.2 Blade-Element Momentum Expansion

Expanding on the Blade-Element Momentum equations from Eq:3.70 & Eq:3.72a. Reiterating the equations, they are:

$$dT = \rho 4\pi r v_\infty (1+a) a dr \quad (\text{A.10a})$$

$$dT = \frac{1}{2} a_L b c \rho (\Omega r)^2 \left( \theta - \frac{v_\infty + v_i}{\Omega r} \right) dr \quad (\text{A.10b})$$

Equating the two and defining an inflow ratio term  $\lambda = \frac{v_\infty + v_i}{\Omega r} = \frac{v_\infty (1+a)}{\Omega r}$  yields the following quadratic equation:

$$\lambda^2 + \left( \frac{\sigma a_L}{8} + \lambda_c \right) \lambda - \frac{\sigma a_L}{8} \theta \frac{r}{R} = 0 \quad (\text{A.11})$$

Where  $\lambda_c$  is the nominal free-stream inflow ratio when  $v_i = 0$ . Another term,  $\sigma$ , is defined as the propeller solidity and is given by:

$$\sigma = \frac{bc}{\pi R} \quad (\text{A.12})$$

Then, solving Eq:A.11 for  $\lambda$ :

$$\lambda = \sqrt{\left( \frac{\sigma a_L}{16} - \frac{\lambda_c}{2} \right)^2 + \frac{\sigma a_L}{8} \theta \frac{r}{R}} - \left( \frac{\sigma a_L}{16} - \frac{\lambda_c}{2} \right) \quad (\text{A.13})$$

So then the inflow ratio can be solved as a function of the propeller element's aerofoil profile and its static inflow factor. In static conditions, the inflow factor is:

$$\lambda = \frac{v_i}{\Omega r} = \sqrt{\frac{C_{T0}}{2}} \quad (\text{A.14})$$

Then substituting  $\lambda$  back into Eq:3.72a and solving the integral produces an instantaneous thrust value. The difficulty of solving the blade-element momentum integrals is knowing the exact chord profile and local angle of attack.

## A.3 Induced Torques

In Ch:3.2.1 the proposed Lagrangian equations for both  $\Delta\lambda$  and  $\Delta\alpha$  servo movements is presented. Reiterating those equations, the inner ring torque response from Eq:3.41a:

$$\mathcal{L}_{M'_i} = \frac{1}{2} \vec{\omega}_{r/M'_i}^T (J'_r) \vec{\omega}_{r/M'_i} + \frac{1}{2} \vec{\omega}_{M_i/M'_i}^T (J'_{ir}) \vec{\omega}_{M_i/M'_i} \quad (\text{A.15a})$$

And for the middle ring response from Eq:3.52

$$\mathcal{L}_{M''_i} = \frac{1}{2} \vec{\omega}_{r/M''_i}^T (J''_r) \vec{\omega}_{r/M''_i} + \frac{1}{2} \vec{\omega}_{M'_i/M''_i}^T (J''_{ir}) \vec{\omega}_{M'_i/M''_i} + \frac{1}{2} \vec{\omega}_{M'_i/M''_i}^T (J'_m) \vec{\omega}_{M'_i/M''_i} \quad (\text{A.15b})$$

The Euler-Lagrange formulation of generalize path co-ordinates  $\vec{v}$  solves for generalized forces  $\vec{V}$  acting on the system as follows:

$$\frac{d}{dt} \left( \frac{\delta \mathcal{L}}{\delta \dot{\vec{v}}} \right) - \frac{\delta \mathcal{L}}{\delta \vec{v}} = \vec{V} \quad (\text{A.16})$$

In the same shared frame which  $\vec{v}$  is defined with respect to. Solutions for generalized forces for both Eq:A.15a and Eq:A.15b are given in Eq:3.47 and Eq:3.56 respectively. That solution exploits the assumption, detailed in Eq:3.43, that partial derivatives of rotation matrices can be linearized. That assumption leads to the simplification that, for a transformed inertia  $J'_r$  of the inner ring in Eq:A.15a for example, its partial derivative is given:

$$\frac{\delta}{\delta \lambda} (J'_r) = \frac{\delta}{\delta \lambda} \left( R_x(\lambda) (J_r) R_x^{-1}(\lambda) \right) \quad (\text{A.17a})$$

Which, from the product rule, leads to:

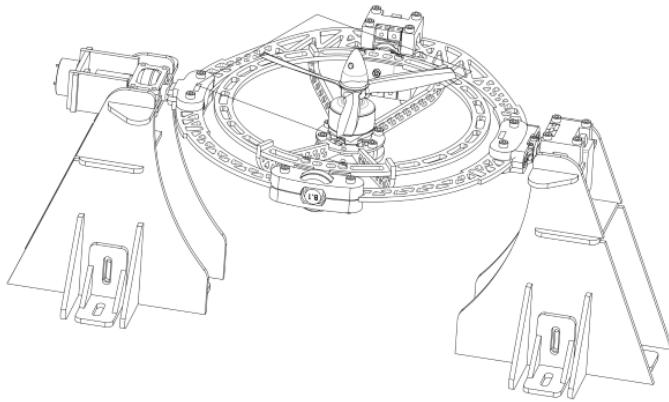
$$= \frac{\delta}{\delta \lambda} (R_x(\lambda)) (J_r) R_x^{-1}(\lambda) + R_x(\lambda) (J_r) \frac{\delta}{\delta \lambda} (R_x^{-1}(\lambda)) \quad (\text{A.17b})$$

If a rotation matrix's partial derivative can be linearized for small  $\Delta \lambda$ , as in [9], it then follows:

$$\frac{\delta}{\delta \lambda} (R_x(\lambda)) \approx 0 \quad (\text{A.17c})$$

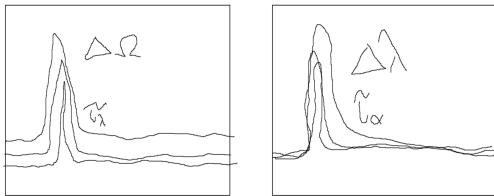
$$\rightarrow \frac{\delta}{\delta \lambda} (J'_r) \approx 0 \quad (\text{A.17d})$$

That simplification results in a reduction of the lagrangian



**Figure A.1:** Torque response rig

The torque response equations were tested using the rig in Fig:A.1. The first plot in Fig:A.2a shows the induced torque for  $\tau_\lambda$  measured purely about the  $\hat{X}_{M_i}$  axis. The plot changes with increased rotation rates of  $\Omega_i$ , motors 1 & 3 have clockwise rotations ('+'), motors 2 & 4 are counter-clockwise ('-'), illustrating the gyroscopic torque effect from the propeller's rotation. Plotted against measured values are  $\hat{\tau}_\lambda$  estimates from Eq:3.47. Similarly the second plot in Fig:A.2b shows the middle ring induced torque,  $\tau_\alpha \in \mathcal{F}^{M'_i}$ . Detailing variations with respect to changing  $\lambda_i$  positions. The changes in  $\mathbb{I}_\alpha(\lambda_i)$  alter the magnitude of torque responses inline with estimates of  $\hat{\tau}_\alpha$ .



(a)  $\tau_\lambda$  variations with propeller speeds  $\Omega_i$       (b)  $\tau_\alpha$  variations with changing  $\mathbb{I}_\alpha(\lambda_i)$

**Figure A.2:** Torque responses for inner and middle rings

The above responses are pertinent to simulation and plant dependent feedback compensation. The simulation environment is structured such that the torques are produced as responses from Newtonian movement at every step interval. In due course it would be more efficient (and less stiff) for the simulation to exploit an implicit Euler [64, 131] coordinate system in lieu of the cartesian response equations developed above. However this was not implemented in Chapter:5 and remains open to further testing and simulation...

## A.4 Euler-Angles from Quaternions

The solution for Euler angles from an attitude quaternion is an easy trigonometric inversion. Noting that the transformation from the body frame to each motor frame follows the Z-Y-X sequence, and using an inversion solution adapted from [116], where the transformation to quaternions is based on Shoemake's [114] definition. Each quaternion can be constructed from sequenced Euler angles, as in Eq:3.28. Then, solving for each euler angle using simultaneous solutions and inverse trigonometry:

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \arctan2(2(q_0q_x + q_yq_z), 1 - 2(q_x^2 + q_y^2)) \\ \arcsin(2(q_0q_y - q_xq_z)) \\ \arctan2(2(q_0q_z + q_xq_y), 1 - 2(q_y^2 + q_z^2)) \end{bmatrix} \quad (\text{A.18})$$

Where  $\arctan2(x, y)$  is the four-quadrant tangent inverse [36], producing the principle argument of the complex operands;

$$\arctan2(x, y) = PR \arg(x + yi) = Arg(x + yi) \quad (\text{A.19})$$

The use of an full quadrature arctangent function is to find solutions for Euler angles that are not only acute.

## 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 Brushless DC	4	44.2
Frame Components		
APM Flight Controller Damping Platform	1	7
HobbyKing SK450 Replacement Arm (2 pcs)	2	51
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

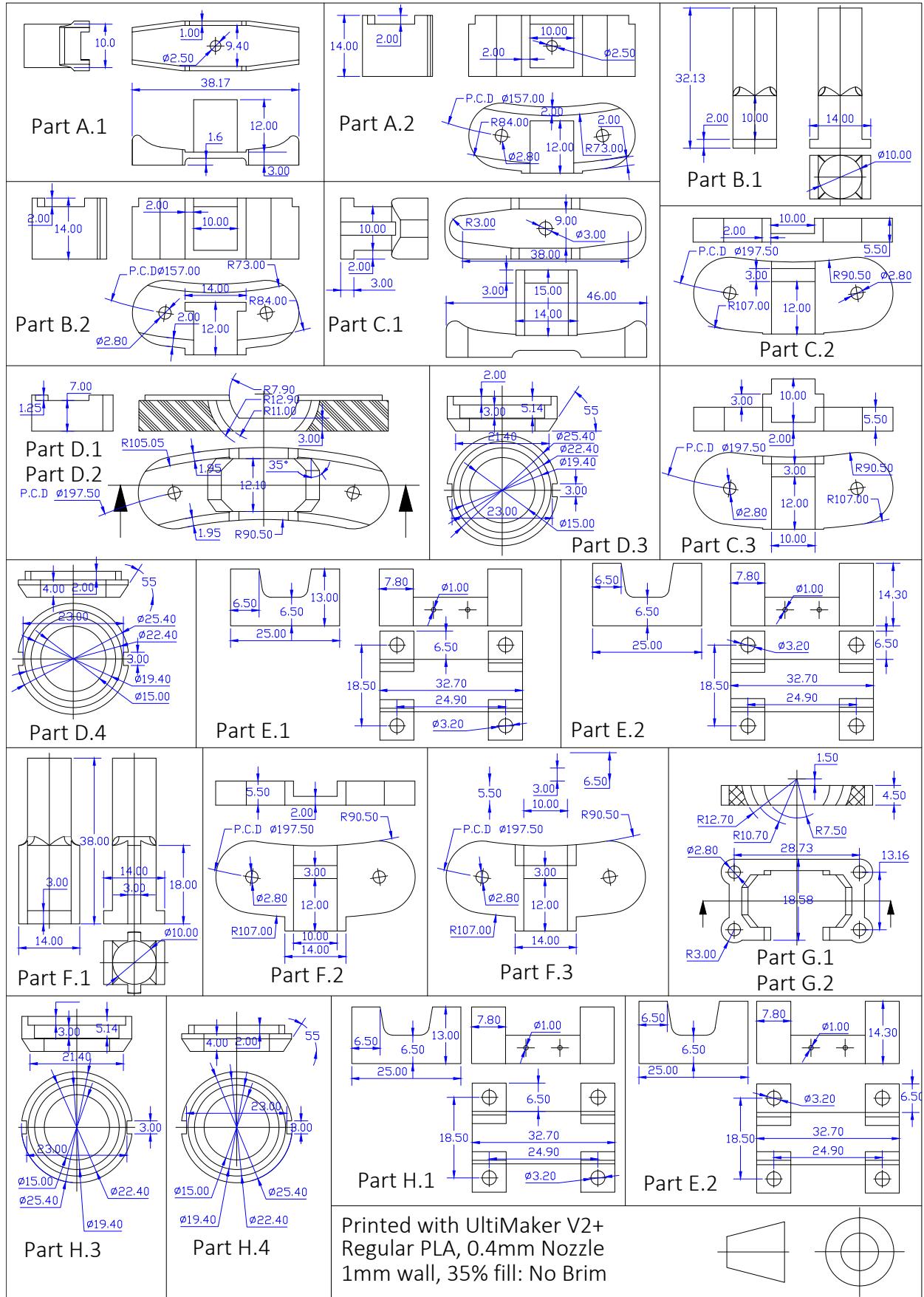
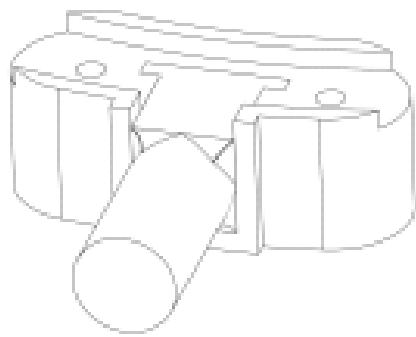
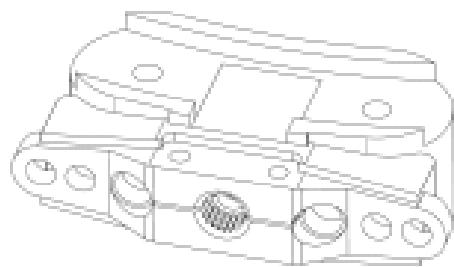


Table B.2: 3D Printed Parts

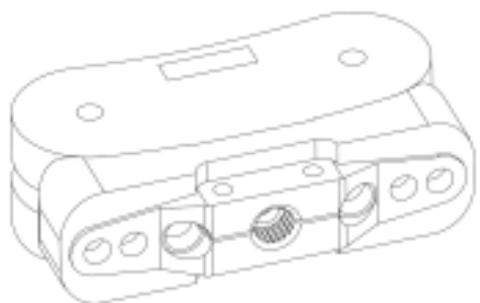
## Bracket Assemblies 2



**Figure B.1:** Bearing Bracket Inner Ring Assembly  
Parts: A.1, A.2



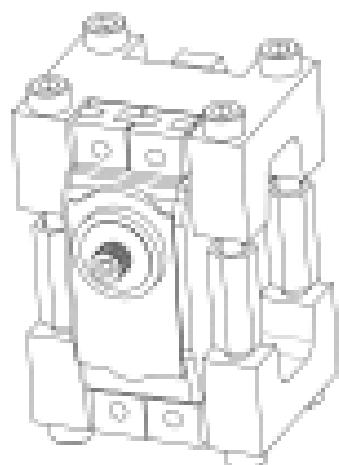
**Figure B.2:** Servo Bracket Inner Ring Assembly  
Parts: B.1, B.2, M3 Servo Horn



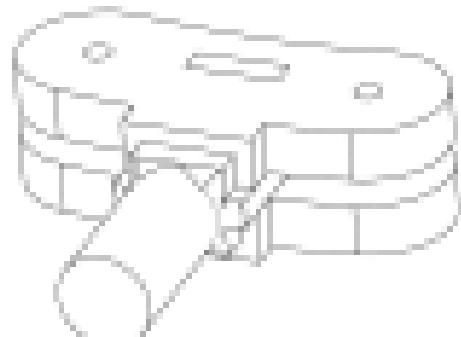
**Figure B.3:** Servo Bracket Middle Ring Assembly  
Parts: C.1, C.2, C.3, M3 Servo Horn



**Figure B.4:** Bearing Holder Middle Ring Assembly  
Parts: D.1, D.2, D.3, D.4, 18-10 Bearing



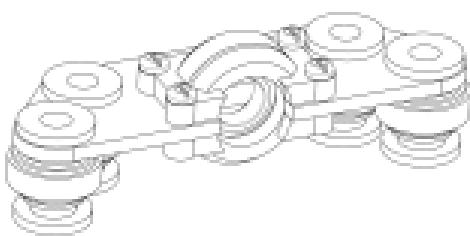
**Figure B.5:** Servo Mount Middle Ring Assembly  
Parts: E.1, E.2, Corona Servo & Fasteners



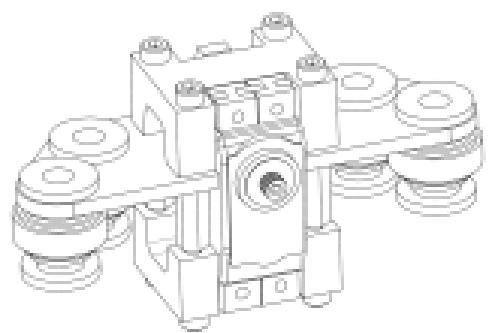
**Figure B.6:** Bearing Shaft Middle Ring Assembly  
Parts: F.1, F.2, F.3

**Table B.3:** Inner & Middle Ring Assemblies

## Bracket Assemblies 2



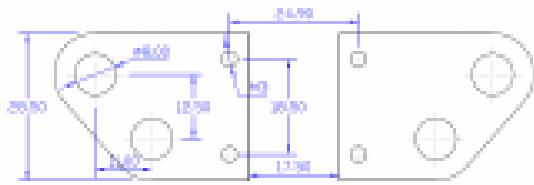
**Figure B.7:** Bearing Holder Damping Assembly  
Parts: G.1, G.2, G.3, G.4, 18-10 Bearing, 80g Damping Balls, Bearing Holder Damping Bracket



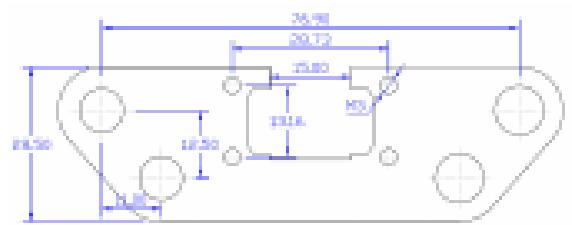
**Figure B.8:** Servo Mount Damping Assembly  
Parts: H.1, H.2, Corona Servo & Fasteners, 80g Damping Balls, Servo Mount Damping Bracket

**Table B.4:** Damping Assemblies

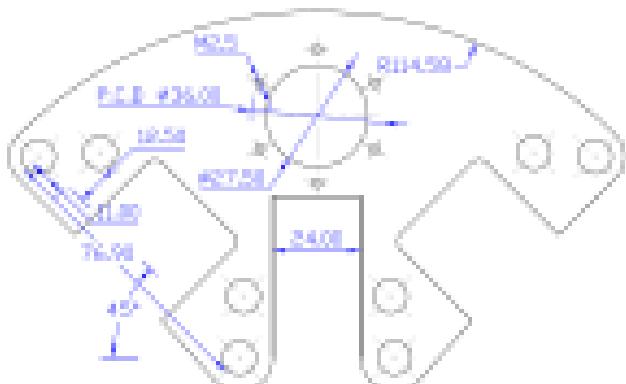
## Laser Cut Brackets



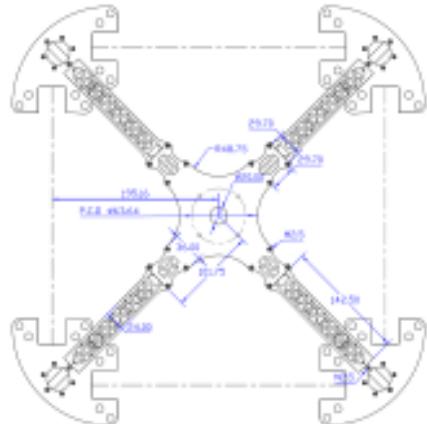
**Figure B.9:** Servo Mount Damping Bracket



**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

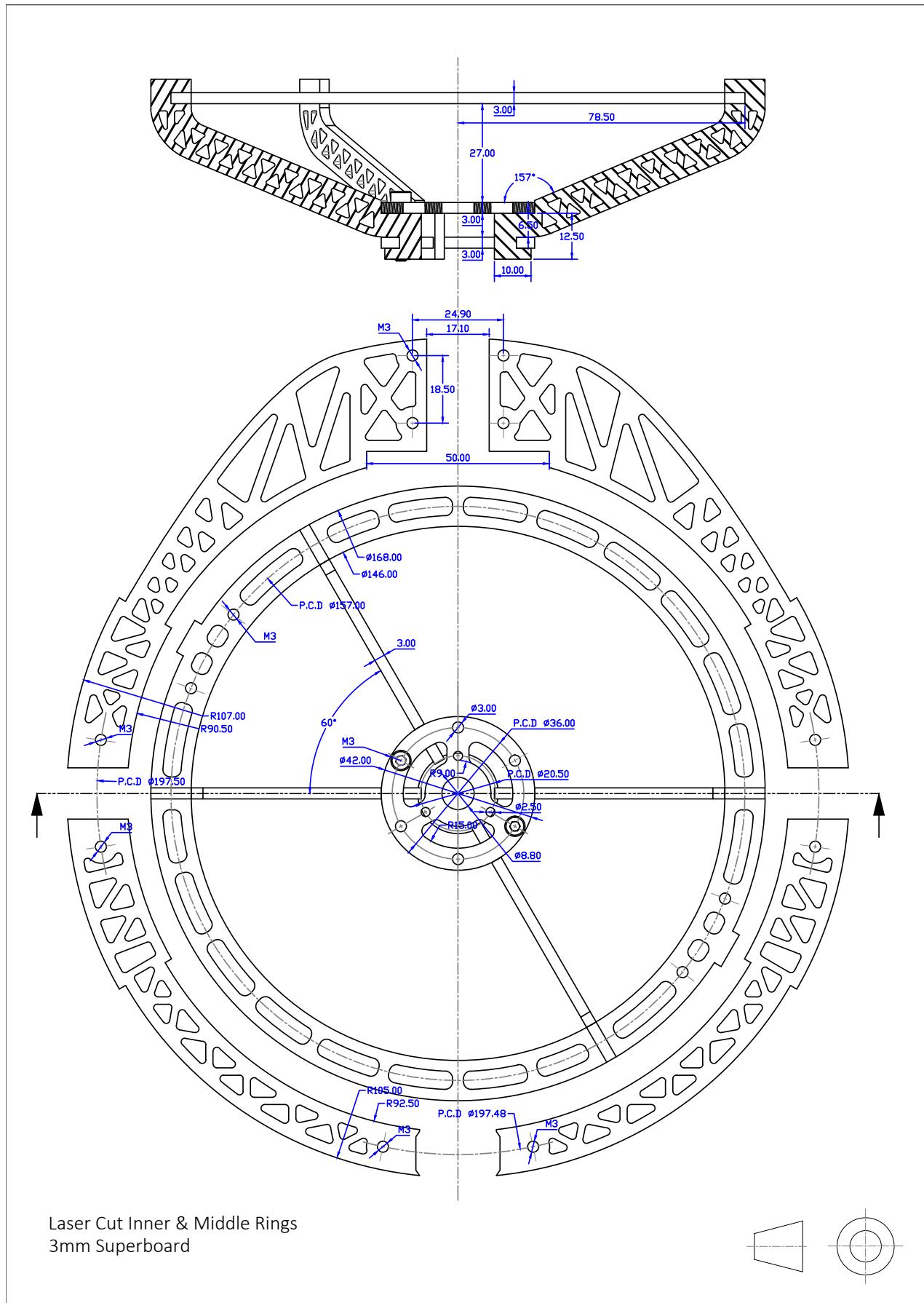


Table B.6: Laser Cut Parts

## B.2 F3 Deluxe Schematic Diagram

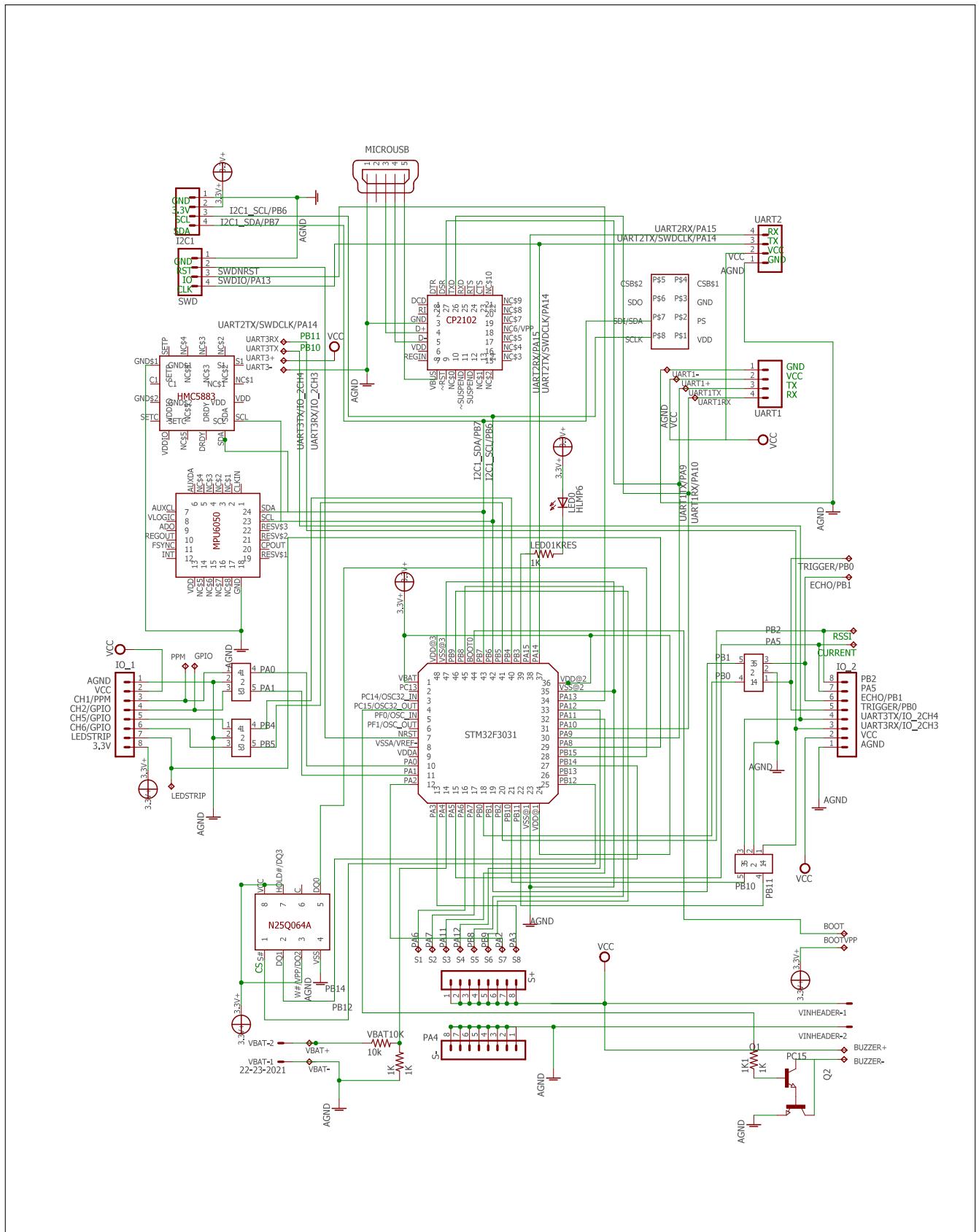
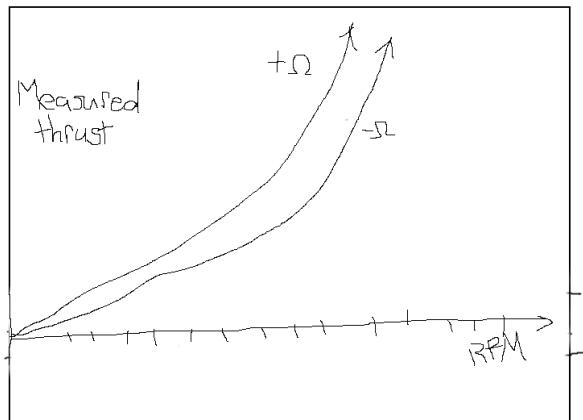


Figure B.13: F3 Deluxe Flight Controller Hardware Schematic

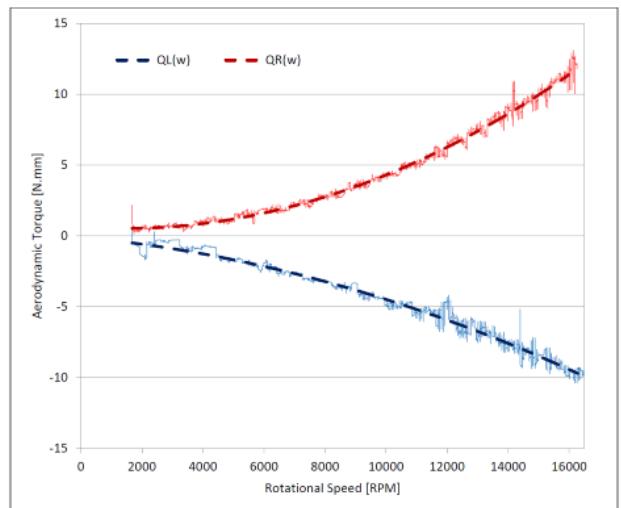
## Appendix C

# System ID Test Data

### C.1 Thrust and Torque Test Data



(a) Thrust tests



(b) Torque tests

**Figure C.1:** Clockwise and counterclockwise rotation tests

## C.2 Cobra CM2208-200KV Thrust Data

Cobra CM-2208/20 Motor Propeller Data										
Magnets 14-Pole	Motor Wind 20-Turn Delta	Motor Kv 2000 RPM/Volt		No-Load Current $I_o = 0.77$ Amps @ 10v	Motor Resistance $R_m = 0.076$ Ohms		I Max 20 Amps	P Max (3S) 220 W		
Stator 12-Slot	Outside Diameter 27.7 mm, 1.091 in.	Body Length 24.0 mm, 0.945 in.		Total Shaft Length 45.2 mm, 1.780 in.	Shaft Diameter 3.17 mm, 0.125 in.		Motor Weight 44.2 gm, 1.56 oz			
Test Data From Sample Motor		Input	6.0 V	8.0 V	10.0V	12.0V	Measured Kv value	Measured Rm Value		
		$I_o$ 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.2: Official Test Results for Cobra Motors

# Bibliography

- [1] Parrot AG. Parrot ar drone 2.0. <https://www.parrot.com/fr/drones/parrot-ardrone-20-elite-edition#technical>, 2016. [Accessed:2017-01-16].
- [2] Yazan Al-Rihani. *Development of a dual axis tilt rotorcraft uav: Design, prototyping and control.*, volume 1. Cranfield University: School of Engineering, 2012.
- [3] 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.
- [4] APMCopter. Arducopter main page. Website: <http://www.arducopter.co.uk/>, 6 2016. Arducopter (APM) Official Website.
- [5] E. Balasubramanian and R. Vasantharaj. Dynamic modelling and control of quadrotor. *International Journal of Engineering and Technology*, pages 63–39, February 2013.
- [6] P. Baldi, B. Mogens, P. Castaldi, N Mimmo, and S. Simani. Adatpive ftc based on control allocation and fault accomodation for satellite reaction wheels. In *Conference on Control and Fault-Tolerant Systems*, volume 3, pages 1–6, 9 2016.
- [7] 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 University of Wellington, 12 2012. Published in Conference Proceedings.
- [8] M. Bangura, M Melega, R. Naldi, and R. Mahony. Aerodynamics of rotorblades for quadrotors. Report, Colaboration: Australian National University & University of Bologna, 1 2016. ArXiv Published:<https://arxiv.org/abs/1601.00733>.
- [9] T. Barfoot, J. Forbes, and P. Furgale. Pose estimation using linearized rotations and quaternion algebra. Report, University of Toronto, Institute for Aerospace Studies, 6 2011.
- [10] 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.
- [11] BetaFlight. Betaflight fc4 repo. Forked from the CleanFlight repo,<https://github.com/betaflight/betaflight>, 2016. [Accessed:2016-9-17].
- [12] BLHeli. Blheli master branch (silabs). <https://github.com/bitdump/BLHeli/tree/master/SiLabs>, 2016. [Accessed:2016-11-05].
- [13] 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.
- [14] 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.

- [15] Edward Boje. Lyapunov stability analysis. Report, University of Kwazulu-Natal: School of Electrical, Electronic and Computer Engineering, 1 2005. Course notes for: Control Systems ENEL4CN.
- [16] 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.
- [17] S. Bouabdallah and R. Siegward. Backstepping and sliding mode techniques applied to an indoor micro quadrotor. *IEEE International Conference on Intelligent Robots and Systems*, pages 2247–2252, 4 2005.
- [18] 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.
- [19] Samir Bouabdallah, Andre Noth, and Roland Siegward. Pid vs lq control techniques applied to an indoor micro quadrotor. *IEEE International Conference on Intelligent Robots and Systems*, pages 2451–2456, 9 2004.
- [20] A.R.S Bramwell, D. Balmford, and G. Done. *Bramwell's Helicopter Dynamics*, chapter 1-3, pages 1–144. Elsevier Ltd, 2 edition, 1999.
- [21] J. Brandt, R. Deters, G. Ananda, and M. Selig. Uiuc propeller data site. Univesity of Illinois Urbana-Champaign; Department of Aerospace Engineering: <http://www.steadidrone.com/>, 2008. [Accessed:2016-13-12].
- [22] 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.
- [23] D. Brescianini, M. Hehn, and R D'Andrea. Nonlinear quadrocopter attitude control. Technical report, Institute for Dynamic Systems and Control, ETH Zurich, 10 2013.
- [24] Z. Cai and M. S. de Queiroz. Asymptotic adaptive regulation of parametric strict-feedback systems with additive disturbance. In *ME Graduate Student Conference*, pages 3–5, Baton Rouge, Louisiana, 4 2005. LSU Graduate School.
- [25] 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.
- [26] 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.
- [27] CleanFlight. Cleanflight repo. <https://github.com/cleanflight/cleanflight>, 2016. [Accessed:2016-11-13].
- [28] Dominic Clifton. Spracing f3 deluxe flight controller. <http://seriouslypro.com/spracingf3>, 2015. [Accessed:2016-10-04].
- [29] 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.
- [30] Innov8tive Designs. Cobra cm2208/2000 motors. <http://innov8tivedesigns.com/cobra-cm-2208-20-motor-kv-2000>, 2016. [Accessed:2016-06-10].

- [31] D. DGaranin. Rotational motion of rigid bodys. Technical report, City University of New York, Department of Engineering, 11 2008. Course Notes cited from: [http://www.lehman.edu/faculty/dgaranin/teaching-analytical\\_mechanics-Fall-2008.php](http://www.lehman.edu/faculty/dgaranin/teaching-analytical_mechanics-Fall-2008.php).
- [32] 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.
- [33] DJI Drones. Dji inspire one. <http://www.dji.com/product/inspire-1>, 2016. [Accessed:2016-07-10].
- [34] DJI Drones. Dji phantom. <http://www.dji.com/products/phantom>, 2016. [Accessed:2016-06-12].
- [35] Honeywell Solid State Electronics. Hmc5833 magnetometer datasheet. Advanced Information Data Sheet, 10 2010. Available From:<https://strawberry-linux.com/pub/HMC5833L.pdf>.
- [36] Internation Organization for Standardization/International Electrotechnical Commission. Programming language c. Open PDF for Library:<http://www.open-std.org/jtc1/sc22/wg14/www/docs/n1124.pdf>, 2005. Section 7.12.4.4:The atan2 functions, Page 219.
- [37] Emil Fresk and George Nikolakopoulos. Full quaternion based attitude control for a quadrotor. *European Control Conference*, pages 3864–3869, 6 2013.
- [38] Pau. S Gasco. *Development of a Dual Axis Tilt Rotorcraft UAV: Modelling, Simulation and Control*, volume 1. Cranfield University: School of Engineering, 2012.
- [39] J. Gertler. V-22 osprey tilt-rotor aircraft: Background and issues for congress. Report, Congressional Research Service, 3 2011.
- [40] GetFPV.com. Cobra 2208 2000kv bldc motor product page. <http://www.getfpv.com>, 2016. [Accessed:2017-02-19].
- [41] HiSystems GmbH. Mikrokopter quadroxl. <http://www.mikrokopter.de/en/products/kits>, 2016. [Accessed:2016-06-13].
- [42] G. Golub and C. Van Loan. *Matrix Computations*, volume 3. Johns Hopkins University Press., 1996.
- [43] Basile Graf. Quaternions and dynamics. Publication for Mathematics - Dynamical Systems, 2 2007.
- [44] 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.
- [45] Gary R. Gress. *Passive Stabilization of VTOL Aircraft Having Obliquely Tilting Propellers*. University of Calgary, Department of Mechanical Engineering, Calgary, Alberta, 2014.
- [46] Karsten Groekatthfer and Zizung Yoon. Introudction into quaternions for spacecraft attitude representation. *Technical University of Berlin: Department of Astronautics and Aeronautics*, pages 1–16, May 2012.
- [47] 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.

- [48] M. Hehn and R. D'Andrea. Quadrocopter trajectory generation and control. In *International Federation of Automatic Control Conference*. Institute for Dynamic Systems and Control, Swiss Federal Institute of Technology; Zurich, 2011. Published in Conference Proceedings.
- [49] Drone HiTech. Xrotor 20a esc. [http://dronehitech.com/wp-content/uploads/2016/04/IMG\\_0524.jpg](http://dronehitech.com/wp-content/uploads/2016/04/IMG_0524.jpg), 2016. [Accessed:2016-11-05].
- [50] HobbyKing. Orangerx rpm sensor. [http://www.hobbyking.com/hobbyking/store/\\_61511\\_Orange\\_RPM\\_Sensor.html](http://www.hobbyking.com/hobbyking/store/_61511_Orange_RPM_Sensor.html), 2016. [Accessed:2016-10-09].
- [51] HobbyKing. Rotorstar super mini s-bec. [http://www.hobbyking.com/hobbyking/store/\\_33987\\_RotorStar\\_Super\\_Mini\\_S\\_BEC\\_6S\\_10A.html](http://www.hobbyking.com/hobbyking/store/_33987_RotorStar_Super_Mini_S_BEC_6S_10A.html), 2016. [Accessed:2016-10-08].
- [52] HobbyKing. Signal converter module sbus-ppm-pwm. [http://www.hobbyking.com/hobbyking/store/\\_88384\\_Signal\\_Converter\\_Module\\_SBUS\\_PPM\\_PWM\\_S2PW\\_.html](http://www.hobbyking.com/hobbyking/store/_88384_Signal_Converter_Module_SBUS_PPM_PWM_S2PW_.html), 2016. [Accessed:2016-10-09].
- [53] HobbyKing.com. A281 920kv brushless outrunn with variable pitch prop assembly. [https://hobbyking.com/en\\_us/a281-920kv-brushless-outrunner-w-variable-pitch-prop-assembly.html?store=en\\_us](https://hobbyking.com/en_us/a281-920kv-brushless-outrunner-w-variable-pitch-prop-assembly.html?store=en_us), 2016. [Accessed:2017-19-0].
- [54] HobbyKing.com. Hobby king: The ultimate hobby experience. <http://www.hobbyking.com/hobbyking/store/index.asp>, 2016. [Accessed:2016-06-12].
- [55] 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.
- [56] E.L Houghton and E.W Carpenter. *Aerodynamics for Engineering Students*, volume 5. Butterworth-Heinemann, 2003.
- [57] 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](https://www.cdiweb.com/datasheets/invensense/MPU-6050_DataSheet_V3%204.pdf).
- [58] Digi International. Xbee/xbee pro rf modules. Technical Data Sheet, 9 2009. Available From:<https://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-Datasheet.pdf>.
- [59] 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.
- [60] Tor A. Johansen and Thor I. Fossen. Control allocation - a survey. *Automatica*, 45:10871103, 11 2012. Prepared for: Department of Engineering Cybernetics - Norwegian University of Science and Technology.
- [61] 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, N/A 2005.
- [62] Tor A. Johansen and Johannes Tjnnns. Adaptive control allocation. White paper, Department of Engineering Cybernetics, Nowegian University of Science and Technology, N/A 2008.

- [63] S.M Joshi, A.G Keklar, and J.T Wen. Robust attitude stabilization of spacecraft using nonlinear feedback. *IEEE Transactions on Automatic Control*, pages 1800–1803, October 1995.
- [64] C. Karen Liu and S. Jain. Tutorial on multibody dynamics. Georgia Institute of Technology: Online Course Content, N/A, 10 2012. Available at: [http://www.cc.gatech.edu/~karenliu/Home\\_files/dynamics\\_1.pdf](http://www.cc.gatech.edu/~karenliu/Home_files/dynamics_1.pdf).
- [65] 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.
- [66] A. Koshkouei, A. Zinober, and K. Burnham. Adaptive sliding mode backstepping control of nonlinear systems with unmatched uncertainty. *Asian Journal of Control*, pages 447–453, 12 2004.
- [67] 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.
- [68] Raymond Kristiansen and Per J. Nicklasson. Satellite attitude control by quaternion-based backstepping. *American Control Conference*, N/A:907–912, 6 2005. Published by Department of Computer Science, Electrical Engineering and Space Technology; Narvik University College.
- [69] Jack B. Kuipers. *Quaternions and Rotation Sequences: A Primer with Application to Orbital Aerospace and Virtual Reality*, pages 127–143. Princeton University Press, September 2002. Used for Quaternion and Rotation Matrix reference.
- [70] Peter Lambert. Nakazawa, banton and jin, bai x. Technical Report N/A, Computer and Electrical Engineering: University of Victoria, Victoria, Canada, 12 2013.
- [71] 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>.
- [72] 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.
- [73] LibrePilot. Openpilot/librepilot wiki. Website: <http://opwiki.readthedocs.io/en/latest/index.html>, 5 2016. Information wiki page for LibrePilot/OpenPilot firmware.
- [74] 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.
- [75] C. Liu, B. Jiang, X. Song, and S. Zhang. Fault-tolerant control allocation for over-actuated discrete-time systems. *Journal of The Franklin Institute*, 352:2297–2313, 3 2015. Research output by Nanjing College of Automation Engineering, University of Aeronautics and Astronautics.
- [76] Kilowatt Classroom LLC. Vfd fundamentals. Report, Kilowatt Classroom LLC, 2 2003.
- [77] SteadiDrone PTY LTD. Steadidrone home. <http://www.steadidrone.com/>, 2016. [Accessed:2016-06-08].
- [78] 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.
- [79] Tarek Madani and Abdelaziz Benallegue. Backstepping control for a quadrotor helicopter. *International Conference on Intelligent Robots and Systems*, pages 3255–3260, October 2006.

- [80] I. Mandre. Rigid body dynamics using euler's equations, rungekutta and quaternions. Unpublished, 2 2006.
- [81] 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.
- [82] 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.
- [83] A. Megretski. Dynamics on nonlinear systems: Finding lyapunov functions. Technical report, Department of Electrical Engineering and Computer Sciences: Massachusetts Institute of Technology, 7 2003.
- [84] 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.
- [85] Svein Rivli Napsholm. Prototype of a tiltrotor helicopter. Master's thesis, Norwegian University of Science and Technology: Department of Engineering Cybernetics, Norway, 1 2013.
- [86] A. Nemati and M. Kumar. Modeling and control of a single axis tilting quadcopter. *American Control Conference*, pages 3077–3082, June 2014.
- [87] 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.
- [88] 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.
- [89] Gustavo P. Oliveira. Quadcopter civil applications. Master's thesis, Informatics and Computer Engineering: University of Portugal, Portugal, 2 2014.
- [90] M. Oppenheimer, D. Doman, and M. Bolender. Control allocation for over-actuated systems. In *Mediterranean conference on Control and Automation*, volume 14, page 1, 6 2006.
- [91] OrangeRx. Orangerx r615x receiver. User Manual, 10 2014. Available From:<http://www.hobbyking.com/hobbyking/store/uploads/672761531X1606554X18.pdf>.
- [92] M. Orsag and S. Bogdan. Influence of forward and descent flight on quadrotor dynamics. Report, Department of Control and Computer Engineering, University of Zagreb, Croatia, 2 2012.
- [93] 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.
- [94] Parth N. Patel, Malav A. Patel, Rahul M. Faldu, and Yash R. Dave. Quadcopter for agricultural surveillance. In *Advance in Electronic and Electrical Engineering*, volume 3, India, 2013.
- [95] Google Patents. V22 osprey paten. <https://www.google.com/patents/US20110177748>, 2010. [Accessed:2017-03-03].
- [96] J. Peraire and S. Widnall. 3d rigid body dynamics: Euler angles. Lecture notes for Dynamics Course, 2009. Dynamics course notes, fall 2007.
- [97] J. Peraire and S. Widnall. 3d rigid body dynamics: The inertia tensor. Lecture notes for Dynamics Course, 2009. Dynamics course notes, fall 2007.

- [98] D. Peters. Eighth amendment of the civil aviation regulations. Goverment Gazette Notice, 5 2015. In Amendment to the Civil Aviation Act, 2009 (Act No.13 of 2009).
- [99] 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.
- [100] 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.
- [101] Dmitry Prof. Garanin. Rotational motion of rigid bodies. Analytical Dynamics Course Notes, 11 2008. Content for City University of New York.
- [102] G Rajagopalan. Froudes momentum theory: (actuator disk theory). Technical report, Iowa State University, Department of Engineering, 7 2002. Course Notes cited from: <http://www.public.iastate.edu/~aero442/unit2.pdf>.
- [103] F. Ramponi and J. Lygeros. Lecture notes on linear system theory. Technical report, Department of Information Engineering, University of Brescia, 1 2015. Course Notes cited from: [http://home.mit.bme.hu/~virosztek/docs/mt\\_literature/LectureNotes.pdf](http://home.mit.bme.hu/~virosztek/docs/mt_literature/LectureNotes.pdf).
- [104] Beard Randal. Quadrotor dynamics and control. Report, Brigham Young University, 2 2008. Part of the Electrical and Computer Engineering Commons.
- [105] 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.
- [106] 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.
- [107] M.K. Rwigema. Propeller blade element momentum theory with vortex wake deflection. *International Congress of the Aeronautical Sciences, 27th*, pages 1–9, January 2010.
- [108] 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.
- [109] 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.
- [110] S. Salazar-Cruz, A. Palomino, and R. Lozano. Trajectory tracking for a four rotor mini-aircraft. In *IEEE Conference on Decision and control, the European Control Conference*, pages 2505–2510, Seville, Spain, 12 2005. IEEE, IEEE 2005.
- [111] Inc. Saleae. Logic8. <https://www.saleae.com/>, 2017. [Accessed:2017-03-12].
- [112] 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.
- [113] J. Seddon. *Basic Helicopter Aerodynamics*, chapter 2-5, pages 4–66. BSP Professional Books, 1 edition, 1990.
- [114] K. Shoemake. Quaternions. Siggraph course lecture notes, Department of Computer and Information Science; University of Pennsylvania, N/A 1987.

- [115] 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.
- [116] G. Slabaugh. Computing euler angles from a rotation matrix. Lecture notes, City University, London, 5 1999.
- [117] Measurement Speacialties. Ms5611 barometric pressure sensor. Technical Data Sheet, 10 2012. Available From:[http://www.amsys.info/sheets/amsys.en.ms5611\\_01ba03.pdf](http://www.amsys.info/sheets/amsys.en.ms5611_01ba03.pdf).
- [118] STMicroElectronics. St-link/v2 in circuit debugger/programmer for stm32.
- [119] 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](http://www.st.com/content/st_com/en/products/microcontrollers/stm32-32-bit-arm-cortex-mcus/stm32f3-series/stm32f303.html?querycriteria=productId=LN1531).
- [120] 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](http://web.mit.edu/1.63/www/Lec-notes/chap1_basics/1-3trans-thm.pdf).
- [121] Abdelhamid Tayebi and Stephen McGilvray. Attitude stabilization of a vtol quadrotor aircraft. *IEEE Transactions on Control Systems Technology*, pages 562–571, May 2006.
- [122] B. Theys, G. Dimitriadis, P. Hendrick, and J. De Schutter. Influence of propeller configuration on propulsion system efficiency of multi-rotor unmanned aerial vehicles. In *International Conference on Unmanned Aircraft Systems*, pages 195–201, Arlington, Virginia, 6 2016. IEEE, IEEE 2016.
- [123] Stephen T. Thornton and Jerry B. Marion. *Classical Dynamics of Particles and Systems*, chapter 7, pages 228–289. Thompson Brooks/Cole, 5 edition, 2003.
- [124] John Ting-Yung Wen and Kenneth Kreutz-Delgado. The attitude control problem. *IEEE Transactions on Automatic Control*, pages 1148–1162, October 1991.
- [125] David Tong. Lagrange formalism. Lectures of Classic Dynamics, Course Notes, 2005. Classical Mechanics Notes.
- [126] P. Tsotras, 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.
- [127] Ultimaker. Ultimaker v2+ product page. <https://ultimaker.com/en/products/ultimaker-2-plus#volume>, 2016. [Accessed:2016-9-11].
- [128] 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>.
- [129] 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.
- [130] Tao Wang, Tao Zhao, Du Hao, and Mingxi Wang. Transformable aerial vehicle, 09 2014.
- [131] A. Witkin and D. Baraff. Physically based modeling: Principles and practice. CMU: Online Siggraph Course notes, N/A, 9 1997. Course Notes cited from <http://www.cs.cmu.edu/~baraff/sigcourse/>.
- [132] 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.

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