Dual-Axis Tilting Quadrotor Aircraft

Dynamic modelling and control thereof

Nicholas Von Klemperer

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

February 2018

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

Friday 2nd February, 2018

Abstract

Dual-Axis Tilting Quadrotor Aircraft

Nicholas Von Klemperer

Friday 2nd February, 2018

This dissertation aims to apply non-zero attitude and position setpoint tracking to a quadrotor aircraft; achieved by solving the problem of a quadrotor's inherent under actuation. The introduction of extra actuation intends to mechanically accommodate for stable tracking of non-zero state trajectories. The requirement of the project is then to design, model, simulate and control a novel quadrotor platform which can articulate all six degrees of rotational and translational freedom (6-DOF) by redirecting and vectoring each propeller's individually produced thrust.

Considering the extended articulation, the proposal is to add an additional two axes (degrees) of actuation to each propeller on a traditional quadrotor frame. Each lift propeller can be independently pitched or rolled relative to the body frame. Such an adaptation, to what is an otherwise well understood aircraft, produces an over-actuated control problem. Being first and foremost a control engineering project, the focus of this work is plant model identification and control solution of the proposed aircraft design. A higher level setpoint tracking control loop designs a generalized plant input (net forces and torques) to act on the vehicle. An allocation rule then distributes that *virtual* input in solving for explicit actuator servo positions and rotational propeller speeds.

The dissertation is structured as follows; first a schedule of relevant existing works is reviewed in Ch:1 following an introduction to the project. Thereafter the prototype's design is detailed in Ch:2; only the final outcome of the design stage is presented. Following that, kinematics associated with generalized rigid body motion are derived in Ch:3 and subsequently expanded to incorporate any aerodynamic and multibody nonlinearities which may arise as a result of the aircraft's configuration (changes). Higher level state tracking control design is applied in Ch:4 whilst lower level control allocation rules are then proposed in Ch:5. Next, a comprehensive simulation is constructed in Ch:6; based on the plant dynamics derived in order to test and compare the proposed controller techniques. Finally a conclusion on the design(s) proposed and results achieved is presented in Ch:7.

Throughout the research, physical tests and simulations are used to corroborate proposed models or theorems. It was decided to omit flight tests of the platform due to time constraints; those aspects of the project remain open to further investigation. The subsequent embedded systems design stemming from the proposed control plant, however, is outlined in the latter of Ch:2, Sec:2.4. Implementations of which are not investigated here but design proposals are suggested. The primary outcome of the investigation is ascertaining the 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. Control derivations and the prototype design presented here are by no means optimal nor the most exhaustive solutions, focus is placed on the system and not just a single aspect of it.

Acknowledgements

Firstly, I'd like to thank Prof Edward Boje for the supervision he provided throughout this research. Secondly, without the support of everyone in the Mechatronics Lab (Room 6.00) at the University of Cape Town, this project would not have been possible. And last, but certainly not least, I would like to thank my parents, without whom none of this would have been possible.

Nomenclature

In order of appearance:

DOF - Degree of Freedom(s)
μC - 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
μC - 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: Ω_i $[rpm]$ for motors: $i \in \begin{bmatrix}1, & 2, & 3, & 4\end{bmatrix}$
Rotational speeds in [RPS] are used for Blade Element Theory Calculations in Chapter:3
Net body torque: $\mu\vec{\tau} = \begin{bmatrix}\tau_\phi & \tau_\theta & \tau_\psi\end{bmatrix}^T \quad \in \mathcal{F}^b$
Net body thrust: $\mu\vec{T} = \begin{bmatrix}T_x & T_y & T_z\end{bmatrix}^T \quad \in \mathcal{F}^b$
Body Position: $\vec{\mathcal{E}} = \begin{bmatrix}x & y & z\end{bmatrix}^T \quad \in \mathcal{F}^I$
Euler Angles: $\vec{\mathcal{E}} = \begin{bmatrix}\phi & \theta & \psi\end{bmatrix}^T \quad \in \mathcal{F}^{I,v1,b}$
Servo 1 Position: λ_i $[rad]$
Servo 2 Position: α_i $[rad]$
Motor module actuator positions: $\begin{bmatrix}\Omega_i & \lambda_i & \alpha_i\end{bmatrix}^T \quad \in \mathcal{F}^{M_i}$
Actuator matrix: $u = \begin{bmatrix}M_1 & \dots & M_4\end{bmatrix} \quad \in \mathbb{U}^{12}$
Motor module displacement arm: $\vec{L}_{arm} = 195.16$ $[mm]$
Euler Rates: $\frac{d}{dt}\vec{\eta} = \dot{\vec{\eta}} = \Phi(\eta)\dot{\omega}_b = \begin{bmatrix}\dot{\phi} & \dot{\theta} & \dot{\psi}\end{bmatrix}^T \quad \in \mathcal{F}^{v1,v2,I}$
Angular Velocity: $\omega = \begin{bmatrix}p & q & r\end{bmatrix}^T \in \mathcal{F}^b$
Linear Velocity: $\nu = \begin{bmatrix}u & v & w\end{bmatrix}^T \in \mathcal{F}^b$

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Foreword

1.1.1 A Brief Background to the Study

A popular topic for current control and automation research is that of quadrotor unmanned aerial vehicles (*UAVs*). Attitude control of a quadrotor poses an unique 6-DOF control problem, to be solved with an underactuated 4-DOF system. As a result the pitch, ϕ, and roll, θ, plants are not directly controllable. The attitude plant is often linearized around a stable operating point. The trimmed operating region is always at the inertial frame's origin; resulting in a zero setpoint tracking problem. The highly coupled nonlinear dynamics of a rigid body's translational and angular motions arise from gyroscopic torques and Coriolis accelerations (Sec: 3.4.1). 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 first order tracking state variable, $\vec{\mathbf{x}}_b = \begin{bmatrix} x \; y \; z \; \phi \; \theta \; \psi \end{bmatrix}^T$, to independent single-input single-output (*SISO*) plants. Those simplifications are derived in the App:A.1.

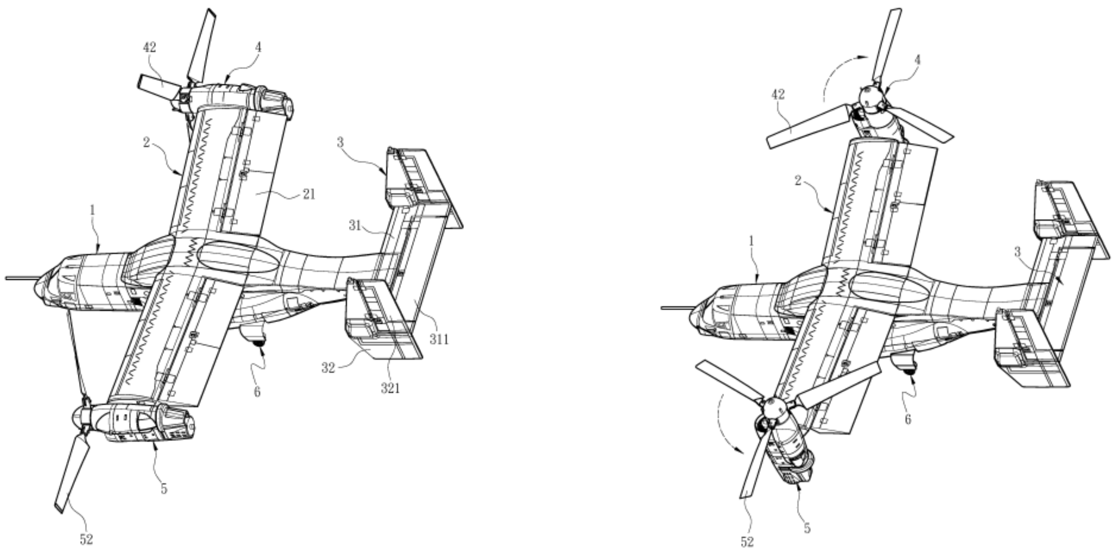
As almost every quadrotor research paper mentions, the recent interest in the platform is due to increased availability of micro-electromechanical systems (*MEMS*) and low-cost microprocessor systems. These technical advancements accomodate onboard state estimation and control algorithm processes in real time. Developmental progress in quadrotors and, to a lesser extent UAVs in general, has led to rapidly growing enthusiast communities. For example; HobbyKing [59] is now a name synonymous with providing custom DIY hobbyist quadrotor assembly kits and frames, no longer retailing only prebuilt commercial products like DJI's Phantom [37] or Parrot's AR [1] drones.

The avenue for potential application of both fixed wing and vertical take-off and landing (*VTOL*) UAVs is expansive; supporting civil [99], agricultural [104] and security [146] 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^{th} amendment of civil aviation laws [108], commercial use of UAVs is now both legalized and regulated. Research into any non-trivial aspect of the field will therefore be to extremely valuable to the field as a whole.

Large scale quadrotor, hexrotor and even octorotor UAVs are popular intermediate choices for aerial cinematography and other high payload capacity applications. The cost of commercial drones such as the SteadiDrone Maverik [86] 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 inertia damping effects from scaling up a vehicle's structure. 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-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 scalar perpendicular heave thrust projection $\vec{F}_\mu = \sum_{i=1}^{4}\ T(\Omega_i) \cdot \hat{z}_b$ in the $\hat{Z}_b$ direction. Those four inputs are then used to effect both the translational XYZ positions $\vec{\mathcal{E}}_I = [x\ y\ z]^T$ and angular pitch, roll and yaw attitude rotations $\vec{\eta}_b = [\phi\ \theta\ \psi]^T$. Pitch and roll torques, τ_ϕ and τ_θ respectively, are produced from differential thrusts of each opposing propellers. Yaw torque τ_ψ is induced from the net aerodynamic drag about each propeller's rotational axis. Aerodynamic drag and differential thrust responses are highly nonlinear (detailed later in Sec:3.2.1) and difficult to approximate as sources of control action. As a result the body's yaw channel control is depreciated. Stemming from the system's underactuation; the attitude control problem is a zero setpoint problem, attempting to track attitudes is ill-posed and will only ever be locally stable (in the Layupanov sense, Sec:4.4).



(a) Rotors articulated forwards **(b)** Rotors articulated upwards

Figure 1.1: Bell/Boeing V22 Osprey actuation, notations pertinent to patent [105]

The aim of this research is to implement non-zero attitude and position state setpoint tracking on a quadrotor by solving the problem of its inherent underactuation. Inspired by Boeing/Bell Helicopter's V22 Osprey (Fig:1.1) and the tilting articulation of its propellers, the prototype design proposed here (described 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 result is four individually articulated 3-D thrust vectors instead of a bound perpendicular net heave force. The control problem is then posed as the design and allocation of net input forces $\vec{F}_\mu = [F_x\ F_y\ F_z]^T$ and torques $\vec{\tau}_\mu = [\tau_\phi\ \tau_\theta\ \tau_\psi]^T$ to act on a general 6-DOF body, such that for any desired trajectory $\vec{\mathbf{x}}_d(t) = [x\ y\ z\ \psi\ \theta\ \phi]^T$ the error state $\vec{\mathbf{x}}_e(t) \triangleq \vec{\mathbf{x}}_d(t) - \vec{\mathbf{x}}_b(t)$ is asymptotically stable. Mathematically:

$$\lim_{t\to\infty} \vec{\mathbf{x}}_e(t) = \vec{0} \qquad \forall \vec{\mathbf{x}} \in \mathbb{R}^n \tag{1.1}$$

Where n is number of the degrees of freedom the system has, typically a 6-DOF plant for rigid bodies. Trajectory stability is explicitly defined later (Sec:4.3) in the context of Lyapunov stability analysis (Sec:4.4). The overactuation brings about the need for a control allocation scheme, one which distributes the six commanded system inputs (net torques and forces) among the twelve actuators in order to optimize some objective function secondary to that of Eq:1.1. The potential improvement(s) for exploiting those overactuated elements is the most novel outcome which the project could yield.

Part of the control research is the multivariable dynamic modelling of the system; making as few assumptions as possible about the nonlinear dynamics involved in the quadrotor's motion and its operational conditions. Common linearizations applied to the quadrotor's control plant will not hold true for more aggressive attitude maneuvers; they are dependent on small angle approximations and neglect second or higher order effects. To produce a stabilizing control solution there first needs to be a plant 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 mathematical plant 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 revolute joints between sub-bodies, the induced relative motion between those sub-bodies will produce complex dynamics like inertial and gyroscopic responses, amongst others. A rotating propeller will respond to pitching or rolling much like a Control Moment Gyroscope [140] or a flywheel, producing a precipitating torque cross product. A less trivial aspect, which is occasionally considered, is 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 strongly plant dependent control law is needed for dynamic compensation, reducing potential fragility associated with the subsequent stability proof. Unmodelled dynamics could push the plant out of the range of stability with regards to the Lyapunov proof.

1.1.3 Scope and Limitations

Scope

Critical to this project is the conceptualized design, prototyping and modelling of a novel actuation suite to be used on a quadrotor platform. The control research question is to apply dynamic setpoint control to the quadrotor platform. Stemming from this is an investigation into the kinematics that are potentially influenced by such a design and the structure's configuration changes. In order to apply correct control theory to achieve the state tracking on the physical prototype, plant dynamics must first be identified for the controller to be designed and 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 on deriving the vehicle's equations of motion and subsequent control design, not the structural integrity of the proposed frame given the forces it may undergo. No flight tests were performed but physical measurements were made on the platform for kinematic inertia measurements (Sec:2.3) and experiments were conducted for corroboration of second order gyroscopic and inertial dynamic responses relating to the novel actuation block (Sec:3.4.2).

Despite aiming to track first order trajectory setpoints, flight path planning and the trajectory generation thereof are not ubiquitous with this dissertation. Derivations for the differential equations for a 6-DOF body's motion, throughout Ch:3, are applicable to any aerospace body, rigid or otherwise. Some particular standards are used, like ZYX Euler Aerospace rotational matrix sequences, all of which are covered in Sec:2.2. The control plant is stabilized with nonlinear state-space control techniques in the time domain, aided and justified by Lyapunov stability theorem [19,93,114]. Alternative solutions using model predictive control (MPC) or quantitative feedback theory (QFT) could perhaps yield 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 setpoint 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 twelve plant inputs and six output variables to be controlled. There is then an entire set of compatible actuator solutions, $\vec{u} \in \mathbb{U} \in \mathbb{R}^{12}$, which satisfy each commanded virtual input. Such a plant is classified as overactuated. Ergo, there must be some logical process as to how those twelve actuators are combined to achieve the desired six control plant inputs; specifically input force $\vec{F}_\mu$ and torque $\vec{\tau}_\mu$ acting on the system.

Appropriate allocation rules are first derived in Ch:5 then simulated and compared in Ch:6 before the final solution is reviewed in Ch:7. It is not a comprehensive survey of every possible allocation or control 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 position and attitude state estimation; which is assumed to be updated through a five-camera positioning system and fused with an on-board 6-axis inertial measurement unit (IMU) using some discretized filtration, is assumed to be accurate and readily disposable at a consistent 50 Hz. Hence state estimation and its discretization effects are included in Sec:6.9 but are bereft of intricate detail. State estimation for quadrotors and aerial vehicles is a thoroughly researched subject, [9, 81, 111].

Limitations

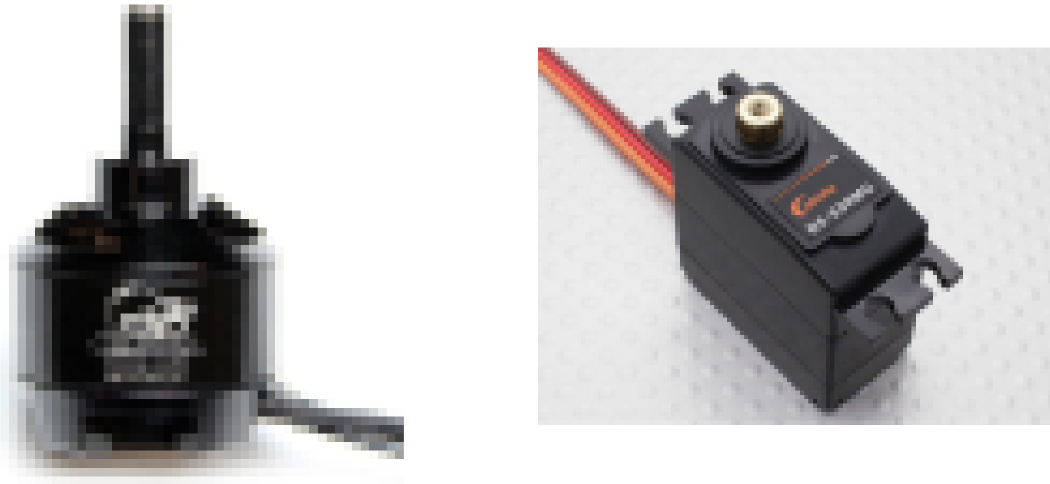
The biggest constraint faced by the design is the net weight of the assembled frame. Lift thrusts which are required to keep an aircraft aloft and oppose the net gravitational force are obviously dependent on the body's net weight. The steady state actuator positions and rates ought to be far less than their respective saturation limits to ensure sufficient actuator headroom to implement control actuations. 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 bandwidth/headroom 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 rotational motion applied to the sub-bodies to overcome their inertial responses. 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 than their larger diameter counterparts could provide. Electing to use three bladed 6×4.5 inch diameter propellers constrained the maximal overall dimensions of the prototype; but as a consequence required very high revolution per minute (RPM) motors. Specifically a set of four Cobra-2208/2000 KV [34] brushless direct current ($BLDC$) motors are proposed for lift actuation (Fig:1.2a).

A direct consequence of that decision is, provisionally based on official thrust tests of the motor included in App:C.2, the net thrust disposable to the control loop is limited to around 950 g $\approx$ 9.3 N, per motor at 14.1 V. That thrust test data is provided from the offical Cobra motor's website, [34], but further verification is done through physical testing in Sec:3.2.1. The frame weight should ideally remain below 50% of the maximum available thrust; or roughly below 2 kg.

Another aspect of limitations produced by design decisions made, mostly to reduce the prototype's cost, is to use of 180° rotation servo motors. Here Corona DS-339MG metal gear digital servos (Fig:1.2b) were selected as they were readily available from university stores. The servos are used for each individual motor's $\hat{X}_{M_i}$ and $\hat{Y}_{M_i}$ axial pitch and roll actuations respectively; terms λ_i and α_i represent those respective rotations for the i^{th} motor set to differentiate from body pitch ϕ and roll θ.



(a) Cobra CM2208/2000KV BLDC motor [45] **(b)** Corona DS-339MG digital servo [59]

Figure 1.2: Mechanical actuators

Servos act in place of either BLDC gimbal or stepper motors with closed loop position control to articulate actuator rotations. The latter pair could both accommodate for continuous ($> 2\pi$) rotations of the actuation modules (Sec:2.1.1) but would need their own control design which includes some element of position feedback. Continuous rotation (velocity controlled) servos could otherwise be used but would similarly require rotational feedback, making the design even more complex. Any rotations beyond 2π would similarly require slip rings to transmit power throughout rotational movement to avoid mechanical interference from connection lines.

Implementing such a design and maintaining an acceptable weight would prove too costly nor would it provide additional insight attained from experimental testing. The effect of servo rotational limits can be evaluated in simulation and if it proves to be significant, continuous rotation could be implemented. The initial design was constructed with flight tests in mind however subsequent dynamic and control derivations proved too time consuming; the project led to a close before final tests could be completed. Throughout the design stage in Ch:2 practical implementation was always considered. Certain elements of the whole system could potentially limit performance but were 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.

An important element of consideration was the prototype's proposed flight controller; needing to provide a total of twelve pulse-width modulated (PWM) output compare channels for the eight servos and four BLDC speed controllers. Moreover the system should have some form of primary state update from a ground control station and a secondary fail safe radio control receiver module, both to be processed by the micro-controller system. Particular attention is paid to the proposed embedded system design and layout in Sec:2.4.

1.1.4 Contributions of Study

Owing to the huge popularity of quadrotor platforms as research tools (i.e [11, 24, 50], 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 nonlinear control techniques for quadrotor platforms (surveyed in Table:1.1). Attitude control loops are the most common topic for research; requiring an unique underactuated solution and mostly linearized around the origin (App: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:4.6. The overactuation 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:5.1. 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 overactuation (expanded upon in next in the lit review, Sec:1.2) hardly broach the topic of tracking attitude setpoints away from the origin.

The control plant presented in this disseration, developed in Ch: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 [51, 53, 120]. Where possible, the system identification and control (both *design* and *allocation*) for this project is kept as generally applicable as possible. The intention is the project's pertinence falls not only within the UAV field but also to any aerospace attitude control plant, rigid or otherwise.

The primary contributions of this work, presented subsequently, start with a mechanical design for a novel quadrotor platform. Then a unique non-linear multibody dynamic model is derived for the vehicle's complex equations of motion. A series of presented control laws are (in some cases) designed and shown to be stable using Lyapunov stability analysis, thereafter higher level commanded control inputs are allocated to lower level actuator modules using derived static allocation laws. The controller coefficient selection exploits an iterative swarm algorithm to optimize each specific set of control coefficients and finally a simulation environment tests the efficacy of all of the above. Ideally the investigation can be expanded upon with more focused research on one of the above subsystems without compromising the stability of the remainder of the 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. For a propeller of rotational inertia J_p about its axis of rotation at an angular speed of Ω_i radians per second, the response torque from rotational accelerations induced in the propeller's frame M_i is given by:

$$\vec{\tau}_p = J_p \dot{\Omega}_i \cdot \hat{z}_{M_i} \qquad \in \mathcal{F}^{M_i} \tag{1.2}$$

Where $i \in [1:4]$ is for each of the four propeller speeds found on a quadrotor. A typical quadrotor helicopter has fixes propellers so each propeller's frame is shared with the body frame $\mathcal{F}^b$. Framing conventions are expanded on in Sec:2.2. Prioritizing pitching the propeller away from its 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 Ch:5). 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.

1.2 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 [44] 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

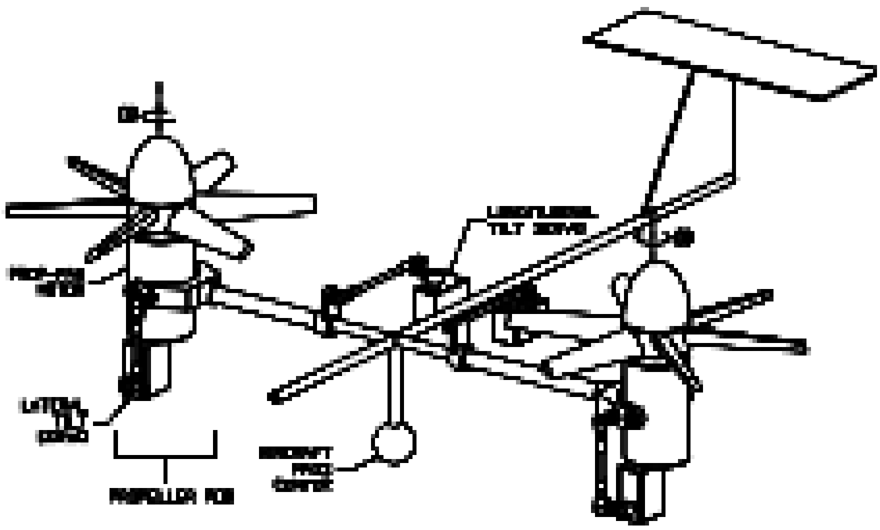


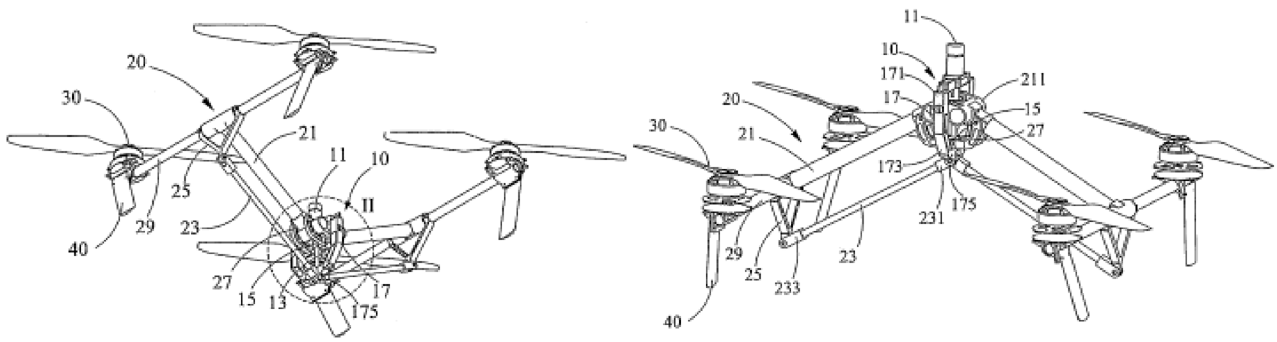
Figure 1.3: General structure for opposed tilting platform, taken from [48]

Research into birotor vehicles (Fig:1.3) with ancilliary lift propeller actuation is oft termed *Opposed Active Tilting* or *OAT*. Such a rotorcraft's mechanical design applies either a single *oblique* 45° tilting axis relative to the body; [17, 72], or a *lateral* tilting axis, adjacent to the body; [31, 74, 103, 122]. 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 actions to further expand the actuation suite, [3, 48]. 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 [17] and PID [103] stabilizing controller schemes. Sometimes more computationally intensive and plant dependent *ideal* or *adaptive* backstepping controllers are implemented, presented in [72, 122] and [74] respectively. The gyroscopic response of a birotor vehicle's attitude system is more pronounced than that of a quadrotor, derived in Sec:3.4, and so feedback linearisation is almost always used. In an interesting progression from the norm, [80] proposed an unique PID co-efficient selection algorithm for a bi-rotor control block. Using a particle swarm optimization (PSO) techinque, similar to [144], 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 for this project using PSO algorithms, shown later in Sec:6.2.

Quadrotors

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 algrithm implemented on what is now known as the X4-Flyer quadrotor [50, 110]. Alternative iterations then followed; like the Microraptor [115] and STARMAC [60] quadcopters which have subsequently been built and tested. A multitude of literature exists around quadrotor kinematics and their control [5, 13, 24], however dedicated rigid body 6-DOF mechanical derivations [89,106] offer better explanations of the kinematics. Often the plant's dynamics are simplified around an origin trim point and assumed to reduce into six SISO plants for each degree of freedom (App:A.1). Lately research projects have begun to incorporate nonlinear aerodynamic effects like drag and propeller blade-element momentum (*BEM*) theory into the plant model [7,27,60,117]. The higher fidelity models for thrust and propeller responses offer more precision by making less linearisations and assumptions.



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

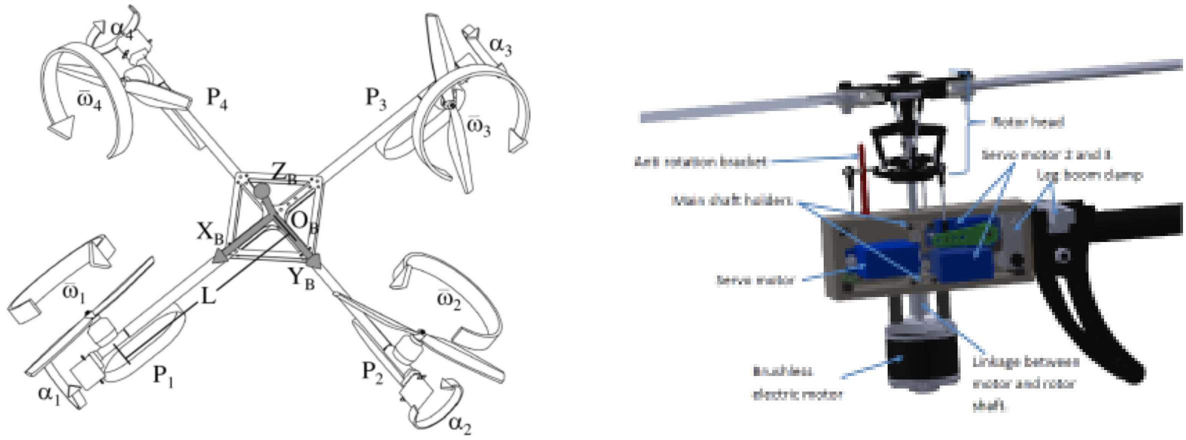
Figure 1.4: DJI Inspire1, the notations are with regards to the DJI patent [141]

At the time of writing, the only commercial UAV multirotor capable of structural transformation is the DJI Inspire1 quadrotor [36], manufactured by Shenzen DJI Technologies. DJI are better known for their hugely successful DJI Phantom commercial quadrotor [37]. The Inspire1 can articulate its supporting arms up and down as shown in Fig:1.4, the purpose of which 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 matrix about its center of gravity, affecting the second order inertial response opposed to changes in angular velocity; $\vec{\tau} = J\dot{\vec{\omega}}$. That variable inertia is a detrimental consequence which makes researchers apprehensive of reconfigurable 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 tilting articulations. The extra actuation scheme(s) were first conceptualized and implemented on a prototype related to an ongoing project covered in two reports; [118, 119]. Those authors modified and tested a QuadroXL four rotor helicopter, propduced by MikroKopter [46], 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 presented in [96]. 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, 43]). Drawing from helicopter design, [95] 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 independent of propeller rotational speed power electronics (*ESCs*) for thrust force actuation.

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, that prototype made use of existing off-the-shelf hobbyist helicopter components to design a rotor actuation bracket (Fig:1.5b). The cyclic-pitch swashplates [97] used could apply pitching and rolling torques, τ_ϕ and τ_θ, about each propeller's hub, its *principle axis of rotation.* The torques were induced by cycling the blade's 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.



(a) Single aligned tilting axis, proposed in [118] **(b)** Cyclic-pitch and swashplate mechanism from [95]

Figure 1.5: Tilt-rotor mechanisms

Irrespective of the strong initial design in the early stages of [95], it would appear that the research suffered due to time constraints. The introductory derivation on aerodynamic effects and deliberation over the design provide clear insight into the projects goals. However the control solution and system architecture, electronic and software, are 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 overactuated design, there was no discussion given on how that actuator allocation, being the most unique aspect of the project, would be achieved.

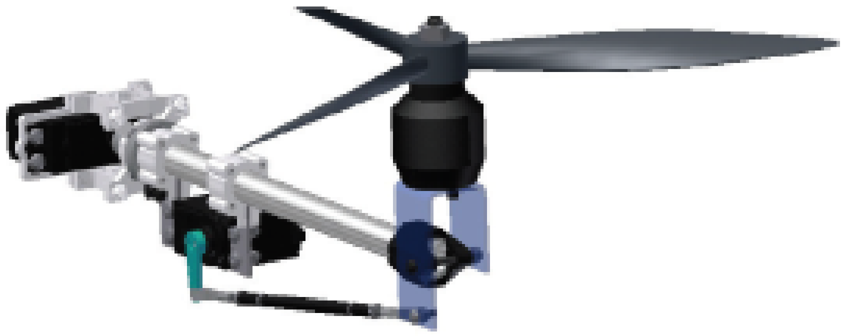


Figure 1.6: Dual-axis tilt-rotor mechanism used in [43]

Finally, the most crucial research to mention is [43], which was a dual presented masters dissertation together with [2]. Currently, this appears to be the only project published pertaining to overactuation 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 research projects. 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. The propellers are treated as energy storing flywheels whose induced gyroscopic response acts as a controllable actuator plant. Thrusts produced by the propellers were not vectored, but the controller's commanded virtual input is distributed to the actuator set by weighted pseudo-inversion, Sec:1.2.2. The extra actuation is justified as fault tolerance redundancy (FTC) but the project does not necessarily detail 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 [134] 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, $\vec{\eta}_b \Rightarrow Q_b$. A general attitude control law h designs an input torque $\vec{\tau}_\mu$:

$$\vec{\tau}_\mu \triangleq h(\vec{\eta}_d, \dot{\vec{\eta}}_d, \vec{\eta}_b, \dot{\vec{\eta}}_b, t) \qquad \in \mathcal{F}^b \tag{1.3a}$$

$$= h(\vec{\eta}_e, \dot{\vec{\eta}}_e, t) \;\text{ given some error state } \vec{\eta}_e \tag{1.3b}$$

Where the control law designs a net torque such that both the angular position and velocity rates are stabilized with the bounded limits; $\lim \vec{\eta}_b \to \vec{\eta}_d$ and $\lim \dot{\vec{\eta}}_b \to \dot{\vec{\eta}}_d$ respectively as $t \to \infty$. Stability definitions are expanded upon later in Sec:4.3. 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 [76, 89, 106], quaternions [41, 49, 76] or otherwise (direct cosine matrix etc ...) the error sate $\vec{\eta}_e = \vec{\eta}_d - \vec{\eta}_b$ could then differ to a (Hamilton) multiplicative relationship. [134] describes these conventionally different error states.

Simulation and modelling papers often rely on Euler angle based rotation matrices for attitude representation, [20, 24, 87, 96, 116], without addressing the inherent singularity associated with such an attitude representation (known as gimbal lock, [125], Sec:3.3.1). The alternative quaternion attitude representation, first implemented in 2006 on a quadrotor UAV platform in [131], 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 [92] and derived mathematically later in Sec:3.3.3. Quaternions are $\in \mathbb{R}^4$ variables for attitude representations in $\mathbb{R}^3$ and so a mapping $\mathbb{R}^4 \to \mathbb{R}^3$ 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, $\vec{\eta}_b = [\phi\ \theta\ \psi]^T$. The cross-coupled gyroscopic and Coriolis terms are both neglected when the body's angular velocity is small, $\vec{\omega}_b \approx \vec{0}$, and the inertial matrix J_b is approximately diagonal, $rank(J_b) = x$ for $\in \mathbb{R}^x$. The consequence of such simplifications is the depreciation 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 state trajectory, $\vec{\mathbf{x}}_b = [x\ y\ z\ \phi\ \theta\ \psi]^T$, 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 SISO channels (quaternion dynamics in Sec:3.3.2). So a quaternion combined four variable attitude state-space vector is then used, $Q_b \triangleq [q_0\ \vec{q}\,]^T$, for the major loop trajectory plant of $\vec{\mathbf{x}}_b(t)$.

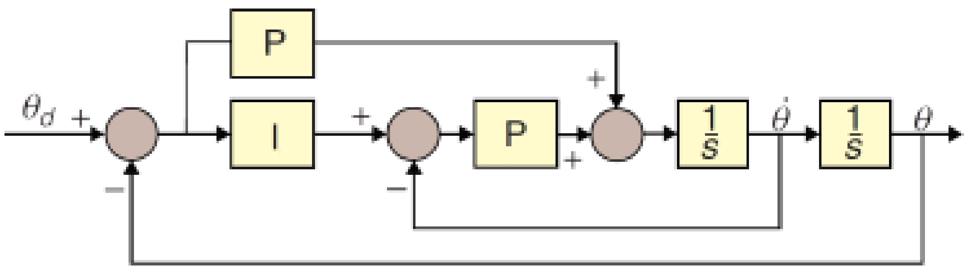


Figure 1.7: ArduCopter PI control structure for pitch angle channel θ; from [83]

Opensource and hobbyist flight controller software (Arducopter [4], Openpilot [82] whose firmware stack is now maintained by LibrePilot, CleanFlight [32], BetaFlight [15], 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. [83] summarizes the control structures implemented on a range of popular flight controllers.

The most popular of which, ArduCopter, implements a feed-forward PI compensation controller, whose single channel control loop for an attitude pitch channel θ 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 of the more unique adaptations is given. One ideal backsteping controller listed in Table:1.1, presented in [122], applies an algorithm derived through Hurwitz polynomials unlike the Lyapunov based backstepping control law(s) used here, derived later in Ch:4.

Controller Type	Independent	Dependent	Total Examples
PI	[134]	[134]	2
PD	[2]	[41, 96]	3
PID	[13, 20, 22, 118, 134]	[60, 116, 134]	8
Lead	[110]	N/A	1
LQR	[22]	N/A	1
Backstepping controllers			
Ideal	[87, 122]	[87]	3
Adaptive	[11, 35, 74, 94]		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 [22–24]; a range of different attitude control implementations are surveyed and tested on the OS4 platform. The final paper, [24], derived and pratically tested an integral backstepping attitude controller on the OS4 quadrotor platform. It builds on their research presented earlier in [22] 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 Euclidean norm (magnitutde) of the plant input. Although, in theory, solving the assocaited Ricatti cost function may produce a cost optimial, 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 [22] 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 [139] 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 *apriori* 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 over accuracy. One example of distrubance approximation in [35] proposes using a statistical projection operator (or $proj(.)$, [29]). When used in adaptive control, presented similarly in [30], the projection operator ensures a derivative based estimator can be bound for adaptive regression approxmation [109].

Although the control implementation is not explicitly backstepping, in [145] a sliding mode controller was used to compensate for the disturbances in an Unmanned Submersible Vehicle atttiude plant. The underwater current disturbances were approximated using a fuzzy logic system, specifically a *zero-order Takagi-Sugeno-kang* fuzzy approximator.

The TSK system has been shown in [90] 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 *apriori* knowledge of a system are well beyond the scope of this research but are worth mentioning.

Single/Dual Axis Control & Allocation

The addititonal control actuation introduced with either single or dual axis articulation provides room for secondary control goals to be achieved. Of the few papers published on tilting-axis quadrotors, PD controllers (used in [96] and again in both [2,43]) and PID controllers (collectively [118,119]) 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 a control allocation survery, [67] describes the control allocation problem for a dynamic plant:

$$\dot{\vec{\mathbf{x}}} = f(\vec{\mathbf{x}}, t) + g(\vec{\mathbf{x}}, \vec{\nu}, t) \qquad \vec{\mathbf{x}} \in \mathbb{R}^n, \vec{\nu} \in \mathbb{R}^m \tag{1.4a}$$

$$\vec{y} = c(\vec{\mathbf{x}}, t) \tag{1.4b}$$

State variables of [67] were changed to match this dissertation's conventions. In the state space equation Eq:1.4a, it is assumed the plant input, $\vec{\nu}$, has a linear multiplicative relationship with the input response, $g(\vec{\mathbf{x}}, t, \vec{\nu}) \Rightarrow g'(\vec{\mathbf{x}}, t)\vec{\nu}$. That linear relationship is a prerequisite for most allocation inversion rules but is not a necessity.

In Eq:1.4a the state $\vec{\mathbf{x}} \in \mathbb{R}^n$ has associated plant dynamics $f(\vec{\mathbf{x}}, t)$ and an input response $g(\vec{\mathbf{x}}, \vec{\nu}, t)$. Setpoint tracking control equates the output variable with the state, in practice only state estimates (denoted by a hat accent) are available:

$$\vec{y} = c(\vec{\mathbf{x}}, t) = A(\vec{\mathbf{x}}) = \hat{\mathbf{x}} \tag{1.5}$$

Therefore the output $\vec{y}$ has the same dimension as the state variable $\vec{\mathbf{x}}$; or rather both $\vec{\mathbf{x}}, \vec{y} \in \mathbb{R}^n$. In an ideal, well posed system the number of actuator inputs equals the number of outputs; that being $dim(\vec{\mathbf{x}}) = dim(\vec{\nu}) \in \mathbb{R}^n$. In the case where the control input $\vec{\nu}$ has a dimension m, for m different actuator plants $\vec{\nu} \in \mathbb{R}^m$. If $m > n$ the problem is then overactuated and a level of 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:

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

Assuming that some higher level control law designs well a satisfactory stabilizing virtual control input from the error state(s) $\vec{\nu}_d = h(\vec{\mathbf{x}}_d, \dot{\vec{\mathbf{x}}}_d, \vec{\mathbf{x}}_b, \dot{\vec{\mathbf{x}}}_b, t) \in \mathbb{R}^n$. 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 $\mathbb{R}^m \to \mathbb{R}^n$ using a commanded actuator matrix position $u \in \mathbb{R}^m$. An overactuated plant can be summarized into a nonlinear state space form as:

$$\dot{\vec{\mathbf{x}}} = f(\vec{\mathbf{x}}, t) + g(\vec{\mathbf{x}}, \vec{\nu}_c, t) \qquad \vec{\mathbf{x}} \in \mathbb{R}^n \tag{1.7a}$$

$$\vec{\nu}_c = B(\vec{\mathbf{x}}, u, t) \qquad \vec{\nu}_c \in \mathbb{R}^n \tag{1.7b}$$

$$\text{With } u \in \mathbb{U}^m \text{ subject to some } min(\vec{s}\,) \text{ such that } \vec{s} = \vec{\nu}_d - \vec{\nu}_c \tag{1.7c}$$

$$\text{Using a genaralized control law: } \vec{\nu}_d = \mathcal{H}(\vec{\mathbf{x}}_d, \dot{\vec{\mathbf{x}}}_d, \vec{\mathbf{x}}_b, \dot{\vec{\mathbf{x}}}_b, t) \qquad \vec{\nu}_d \in \mathbb{R}^n \tag{1.7d}$$

$$\vec{y} = c(\vec{\mathbf{x}}, t) = \vec{\mathbf{x}} \tag{1.7e}$$

The effectiveness function $B(\vec{\mathbf{x}}, u, t)$ 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:5.2) require that $B(\vec{\mathbf{x}}, u, t)$ can be abstracted to a linear multiplicative relationship $B'(\vec{\mathbf{x}}, t)u$ with $B'(\vec{\mathbf{x}}, t) \in \mathbb{R}^{n \times m}$, such that a generalized inverse of $B'(\vec{\mathbf{x}}, t)$ can be found. For generic setpoint tracking the control law $\mathcal{H}$ 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 $\vec{s} \triangleq \vec{\nu}_c - \vec{\nu}_d$ is minimized:

$$\underset{u \in \mathbb{R}^m,\ s \in \mathbb{R}^n}{min} \|\vec{s}\,\|_2 \text{ subject to } \vec{\nu}_c - \vec{\nu}_d = B(\vec{\mathbf{x}}, u, t) - \mathcal{H}(\vec{\mathbf{x}}_e, \dot{\vec{\mathbf{x}}}_e, t) = \vec{s} \quad u \in \mathbb{U} \tag{1.8}$$

Which ensures the commanded input $\vec{\nu}_c$ tracks the desired control input $\vec{\nu}_d$; $\vec{\nu}_c \to \vec{\nu}_d$ as per some cost function of the slack variable . Mostly the L_2 norm $\|s\|_2$ is used but could be some different cost metric. In an overactuated system it then follows that there is a whole set of possible inputs for each commanded $\vec{\nu}_c$. An unique actuator solution (rather than a family of solutions) to Eq:1.8 needs a secondary objective function, $j(\vec{\mathbf{x}}, u, t)$ to be solved explicitly. Eq:1.8 expands to:

$$\underset{u \in \mathbb{R}^m,\ s \in \mathbb{R}^n}{min} (\|\vec{s}\|_2 + j(\vec{\mathbf{x}}, u, t)) \text{ subject to } \vec{\nu}_c - \vec{\nu}_d = s \quad u \in \mathbb{U} \tag{1.9}$$

The same author from [67–69] proposed multiple control allocation solutions to a variety of systems. Following [67]; in a subsequent paper [68], the authors introduced a secondary cost function, driving the solution away from the typical linear quadratic programming pseudo and weighted inverse solutions. Aiming for actuator efficiency and not just input saturation, a subsequent paper [69] 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.

Overactuation 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 Ch:5. The only overactuated quadrotor literature which covers allocation of the extra actuators is [2, 43], where the authors apply a weighted pseudo inverse (otherwise known as the Moore-Penrose Inverse [78]) 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:1.7b.

The only overactuated quadcopter paper which addressed its required control allocation was that of the combined project in [2, 43]. That proposed solution applied weighted inversion, relying on some very specific assumptions to achieve the required input actuator linearity for the system in Eq:1.7b. For the gyroscopic torque response to extra actuator η pitching or γ rolling movement applied to each rotating propeller about the body's $\hat{X}_b$ and $\hat{Y}_b$ axes respectively:

$$\vec{\tau} = \big(\dot{\eta} \cdot \hat{X}_b\big) \times J\big(\Omega \cdot \hat{Z}_b\big) \quad \in \mathcal{F}^b \tag{1.10}$$

With Ω being that propellers rotational speed and $\dot{\eta}$ being the inducing servos rate. Projections onto body axes were used in Eq:1.10 seeing as the resultant thrust/responses were not vectored or assumed to be redirected. The authors assumed the extra actuators pitch and roll angular rates; $\dot{\eta}$ and $\dot{\gamma}$ respectively, were both proportionally related to their positions η and γ as follows:

$$\dot{\eta} \approx \frac{1}{t_{settle}} \Delta\eta \ \text{ and } \ \dot{\gamma} \approx \frac{1}{t_{settle}} \Delta\gamma \tag{1.11}$$

Where t_{settle} is a constant derived in 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 $\vec{\tau} = -\vec{\omega}_b \times J_b \vec{\omega}_b$ and second order inertial torque $\vec{\tau} = J_b \dot{\vec{\omega}}_b$ responses are both functions of their associated servo positions η and γ, not respective their derivatives. The extent of that consequence is contrasted with the allocation solution in proposed later in Ch:5.

Satellite Attitude Control

Unconstrained attitude setpoint tracking for 6-DOF bodies, quaternion based or otherwise, is a topic well covered in the field of satellite attitude control; [66,75,136]. The *status quo* for recent research is on nonlinear adaptive backstepping attitude control systems, wherein the adaptive update rule is the novel contribution. Plant uncertainty always adversly affects the confidence in inertial measurements critical to the attitude control of a satellite. In [66] 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, [18] suggested an algorithm for estimating the inertial matrix using controlled single axis perturbations. Such an approach does assume any initial values are sufficiently close to true body measurements such that estimates 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 only a finite number of times. The thrusters can then be scheduled with a lower priority, prefering bias of electronic CMG actuators. In [75] the authors address the overactuation with direct pseudo inversion before applying quaternion based backstepping for attitude control. Such an inversion solves for Eq:1.9 as follows:

$$u = B^{\dagger}\vec{\nu}_d \tag{1.12a}$$

$$B^{\dagger} = B^T(BB^T)^{-1} \tag{1.12b}$$

$$u \in \mathbb{R}^m,\ \vec{\nu}_d \in \mathbb{R}^n,\ B \in \mathbb{R}^{m\times n},\ B^{\dagger} \in \mathbb{R}^{n\times m} \tag{1.12c}$$

Where B is the effectiveness matrix which is a static effector form of the effectiveness function $B(\vec{\mathbf{x}}, u, t)$. The generalized inverse $B^{\dagger}$ is such that $BB^{\dagger} \equiv \mathbb{I}_{n\times n}$. Specifically $B^{\dagger}$ is the general *pseudo* inversion matrix of B (more on inversions in Ch:5). Moreover there is an assumed *affine* multiplicative relationship between the input, $u \in \mathbb{U}$, and the input effectiveness matrix from Eq:1.7b.

The higher level controller designs actuator torques, $\vec{\nu}_d$, which are then used to solve for explicit actuator positions u as per the inversion equation Eq:1.12a. Much like the overactuation previously discussed with respect to quadcopters; the pseudo inversion method of actuator distribution applies linear quadratic programming optimization to the allocation slack cost function, Eq:1.8. The resultant quaternion attitude backstepping controller developed in [75] demonstrated global uniform asymptotic stability. The strength of that backstepping stability lies in the choice of trajectory aiming to be stabilized; $z \to \vec{0}$.

The first candidate Lyapunov trajectory was defined as:

$$z_1 = \begin{bmatrix} 1 - |q_0| \\ \vec{q}_e \end{bmatrix} \tag{1.13a}$$

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

$$V_1(z) = {z_1}^T z_1 > 0 \quad \forall [q_0, \vec{q}_e] \tag{1.13b}$$

The absolute quaternion error scalar used in Eq:1.13a ensures a global trajectory's asymptotic stability (Sec:4.6.3), not just local stability that would otherwise be gained. The stable equilibrium points at $Q_e = \begin{bmatrix} \pm 1 & \vec{0} \end{bmatrix}^T$ 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

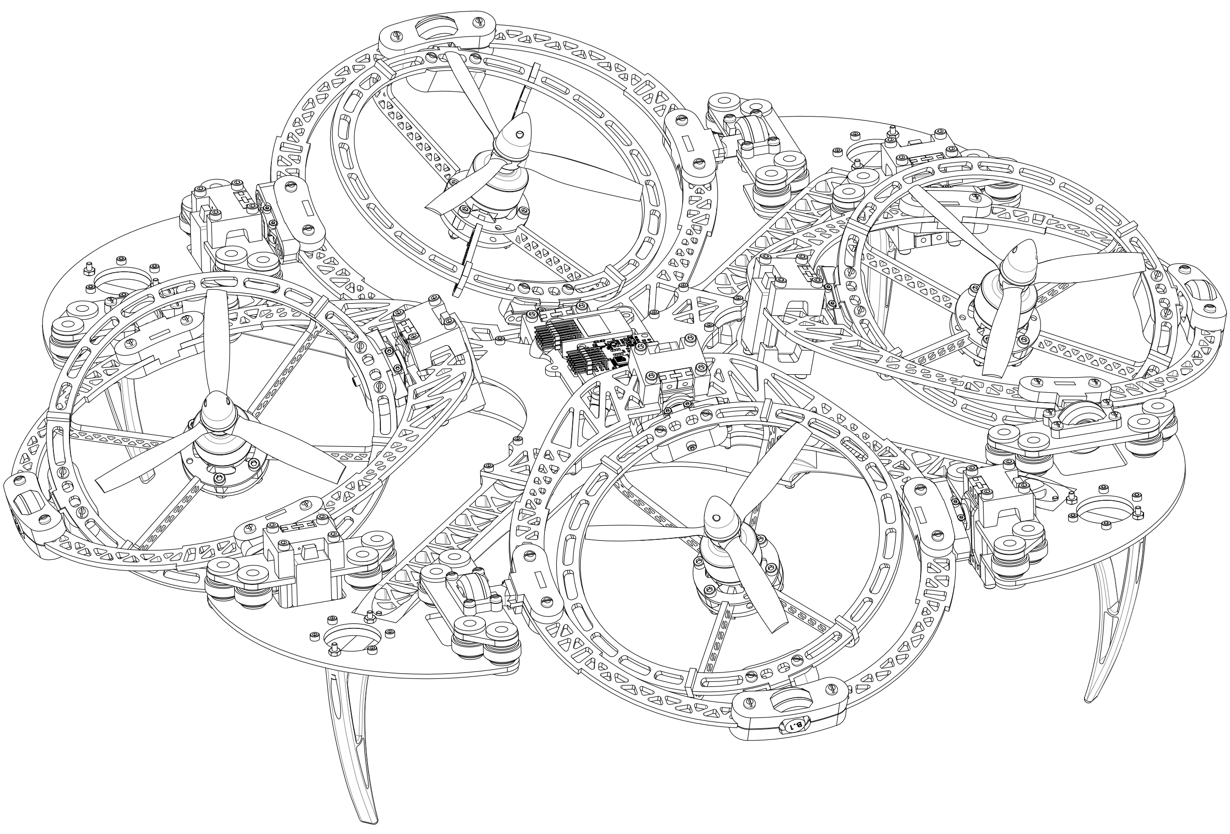
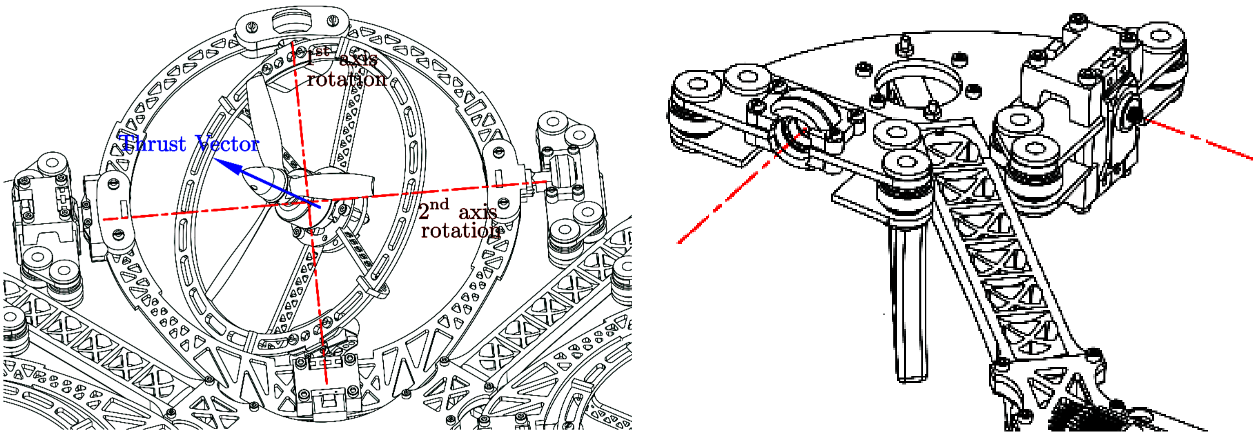


Figure 2.1: Isometric view of the prototype design

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 inertia 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.

2.1.1 Actuation Functionality

The most important component of the design is the articulation for each of the four vectored thrust forces. A concentric gimbal ring structure (Fig:2.2a) independently redirects each lift propeller/motor about two separate rotational axes. Within each module are servos affixed onto sequential gyroscope-like support rings to accommodate pitching and rolling of the propeller's direction. 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 nonlinear actuator plants*). The precise effects are quantified numerically later in Sec:2.3.



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

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 of the thrust vector, $L_{arm} = 195.16$ [mm], co-planar to the body frame's XYZ origin $\vec{\mathbf{O}}_b$ (see subsequent Fig:2.8). That length directly affects the differential torque plant; $\vec{\tau}_{diff} \triangleq \sum \vec{L}_i \times \vec{T}_i$. 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 the 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 modelling a propeller's aerodynamic thrust (Sec:3.2.1).

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+ [137]. 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 App: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:3.4.1. The propellers are six inch (6×4.5) three-bladed plastic Gemfam propellers, powered by Cobra CM2208-2000 KV Brushless DC motors (Fig:2.4a). The thrust produced as a function of angular velocity (in revolutions per second) for the propellers is derived later in Sec:3.2.1.

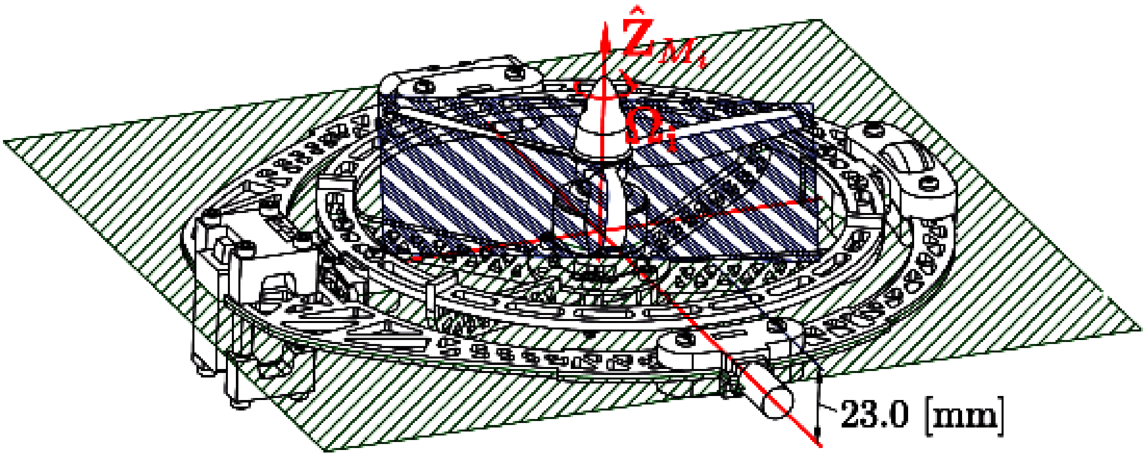
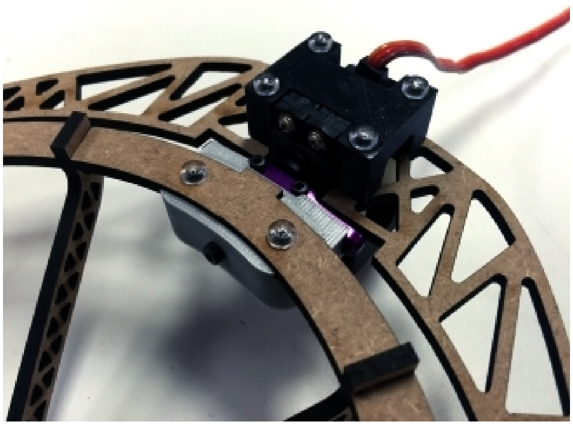


Figure 2.3: Difference between propeller and motor planes

The BLDC motors are controlled with LDPower 20A ESC modules with an in-line OrangeRx RPM Sensor. The ESCs were reflashed with BLHeli [16] firmware. The default firmware on the speed controllers had an unsatisfactory exponentially approaching, nonlinear input speed curve; in contrast with the linear unloaded speed curve in Fig:2.24. 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). 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 $\approx 180°$, positioned such that a zero$^{\text{th}}$ offset aligns the motor modules, adjacent to the body frame, and has a $\pm 90°$ 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 $\hat{X}_{M_i}$ and $\hat{Y}_{M_i}$ axis servos will be rotating differing inertial bodies; as such their open loop transfer functions are individually determined through testing in Sec:2.4.1.

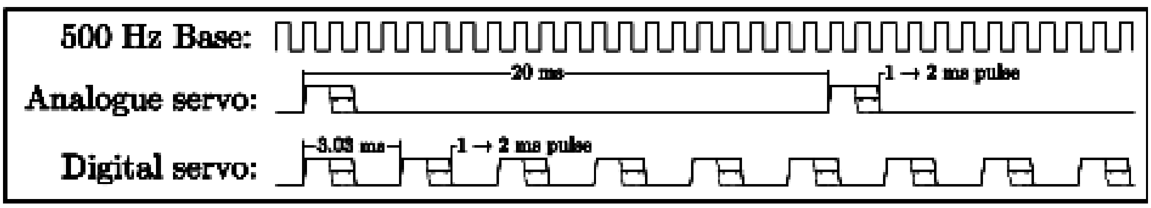


Figure 2.5: Digital and analogue servo timing

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 [49] and [106] provide an in-depth and thorough explanation of rotation matrices and direct cosine matrix attitude representation, if such concepts are unfamiliar to the reader. Later quaternions are used to replace rotation matrix notation for the dynamics in Sec:3.3.2.

2.2.1 Reference Frames Convention

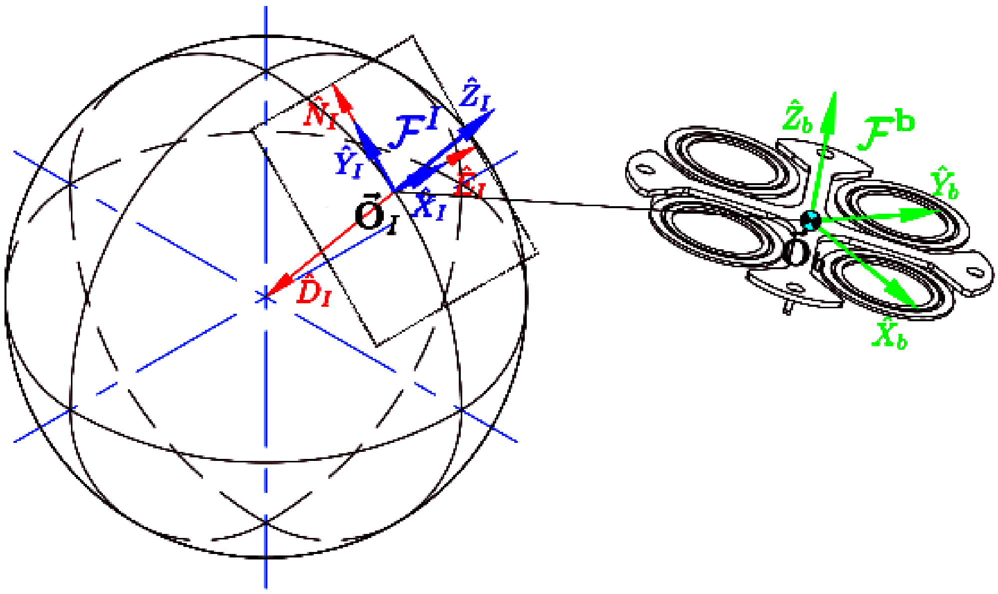


Figure 2.6: Inertial and body reference frames

NASA aerospace frames are used for principle Cartesian inertial and body coordinate representation (Fig:2.6). The inertial frame, $\mathcal{F}^I$ with an origin $\vec{\mathbf{O}}_I$, 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 NÊD convention which differs from the NASA frames used here. The body frame, $\mathcal{F}^b$ centered on the point $\vec{\mathbf{O}}_b$, 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 are both 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. The function $R_I^b(\eta)$ represents a rotation operator of the Euler set $\vec{\eta}$ (expanded on in Eq:2.11) rotating from subscript frame $\mathcal{F}^I$ to superscript frame $\mathcal{F}^b$.

A vector $\vec{\nu}$ has the relationship between the body and inertial frames:

$$\vec{\nu}_I \equiv R_b^I(\eta)\vec{\nu}_b \qquad \vec{\nu}_b \in \mathcal{F}^b,\ \vec{\nu}_I \in \mathcal{F}^I \tag{2.1}$$

Displacement between the inertial and body frames is given by $\vec{\mathcal{E}}_I$, defined in the inertial frame:

$$\vec{\mathcal{E}}_I \triangleq \begin{bmatrix} x & y & z \end{bmatrix}^T \qquad \in \mathcal{F}^I \tag{2.2}$$

An axial hat and upper case differentiates axis unit vectors $\hat{X}, \hat{Y}, \hat{Z}$ from inertial position quantities x, y, z in Eq:2.2. The body position's time derivative $\dot{\vec{\mathcal{E}}}_I$ refers to the *inertial frame* rate:

$$\frac{d}{dt}\vec{\mathcal{E}}_I = \begin{bmatrix}\dot{x} & \dot{y} & \dot{z}\end{bmatrix}^T \qquad \in \mathcal{F}^I \tag{2.3}$$

Whereas the body's translational velocity $\vec{v}_b$ is with respect to the body frame $\mathcal{F}^b$. Velocity and the inertial position time derivative are related as follows:

$$\vec{v}_b \triangleq R_I^b(\eta)\dot{\vec{\mathcal{E}}}_I \qquad \in \mathcal{F}^b \tag{2.4a}$$

$$= R_I^b(\eta)\begin{bmatrix}\dot{x} & \dot{y} & \dot{z}\end{bmatrix}^T \tag{2.4b}$$

Relative angular displacement between two frames is commonly measured by the three angle Euler set. The Euler angle set $\vec{\eta} \triangleq [\phi \ \theta \ \psi]^T$ represents pitch ϕ, roll θ and yaw ψ rotations about sequential $\hat{X}$,$\hat{Y}$ 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 $\vec{\nu}$ about a normalized unit axis $\hat{u}$ through a rotation angle θ is given by the rotation formula, derived in [42]:

$$\vec{\nu}\,' = \big(1 - cos(\theta)\big)\big(\vec{\nu}\cdot\hat{u}\big)\hat{u} + cos(\theta)\vec{\nu} + sin(\theta)\big(\hat{u}\times\vec{\nu}\big) \tag{2.5}$$

In Eq:2.5, when the unit vector $\hat{u}$ is in the direction of either $\hat{X}$; $\hat{Y}$ or $\hat{Z}$ axes the equation is 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 a Cartesian frame's XYZ axes are defined as:

$$R_x(\phi) \triangleq \begin{bmatrix}1 & 0 & 0\\ 0 & cos(\phi) & -sin(\phi)\\ 0 & sin(\phi) & cos(\phi)\end{bmatrix} \tag{2.6a}$$

$$R_y(\theta) \triangleq \begin{bmatrix}cos(\theta) & 0 & sin(\theta)\\ 0 & 1 & 0\\ -sin(\theta) & 0 & cos(\theta)\end{bmatrix} \tag{2.6b}$$

$$R_z(\psi) \triangleq \begin{bmatrix}cos(\psi) & -sin(\psi) & 0\\ sin(\psi) & cos(\psi) & 0\\ 0 & 0 & 1\end{bmatrix} \tag{2.6c}$$

The notation for a rotation matrix operation is multiplication of the matrix $R_u(\theta)$, applying a left-handed *rotation* operator about some axis $\hat{u}$ by θ. The resultant vector of a rotation operation still exists in the same reference frame. For example an $\hat{X}$ axis rotation by ϕ of some vector $\vec{\nu}$ is given by:

$$\vec{\nu}' = R_x(\phi)\vec{\nu} \qquad \vec{\nu}\,', \vec{\nu} \in \mathcal{F}^1 \tag{2.7a}$$

No subscripts are used in Eq:2.7 to indicate reference frame ownership because all vectors are in the same frame. The time derivative of a rotation matrix about some axis $\hat{u}$ by a rotation θ, $\dot{R}_u(\theta)$ is shown in [13] to be:

$$\frac{d}{dt}\big(R_u(\theta)\big) \triangleq \big(\dot{\theta}\cdot\hat{u}\big)\times R_u \equiv \big[\dot{\theta}\cdot\hat{u}\big]_\times R_u \tag{2.8a}$$

Where $\dot{\theta}\cdot\hat{u}$ is the projection of the angular rate $\dot{\theta}$ onto the $\hat{u}$ axis. Furthermore, for some vector $\vec{a}$, the operator $[\vec{a}]_\times$ 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 $\vec{b}$;

$$\vec{a}\times\vec{b} \equiv \big[\vec{a}\big]_\times\vec{b} \tag{2.8b}$$

$$\rightarrow \big[\vec{a}\big]_\times \triangleq \begin{bmatrix}0 & -a_3 & a_2\\ a_3 & 0 & -a_1\\ -a_2 & a_1 & 0\end{bmatrix} \tag{2.8c}$$

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

$$\vec{\nu}_2 = R_x(-\phi)\vec{\nu}_1 \tag{2.9a}$$

$$\vec{\nu}_2 \in \mathcal{F}^2 \text{ and } \vec{\nu}_1 \in \mathcal{F}^1 \tag{2.9b}$$

The distinction between Eq:2.7 and Eq:2.9 is the directional sense of the angular operand ϕ, and hence the effect it has on the argument vector. The transformation or rotation of a vector from the inertial frame $\mathcal{F}^I$ to the body frame $\mathcal{F}^b$ is the product of three sequential operations about each princple 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 indeed effects the Euler set, any consequences of which are detailed in [76]. This dissertation uses the ZYX or yaw, pitch, roll rotation sequence. A rotation of the vector $\vec{\nu}$ from the inertial to the body frame, $\mathcal{F}^I \to \mathcal{F}^b$, is then applied by sequential yaw, ψ, pitch, θ, and roll ϕ operations about the $\hat{Z}$, $\hat{Y}$ and $\hat{X}$ axes respectively:

$$R_I^b(\eta) = R_I^b(\phi,\theta,\psi) \triangleq R_z(\psi)R_y(\theta)R_x(\phi) \tag{2.10a}$$

$$\vec{\nu}\,' = R_I^b(\phi,\theta,\psi)\vec{\nu} \qquad \in \mathcal{F}^I \tag{2.10b}$$

$$= R_z(\psi)R_y(\theta)R_x(\phi)\vec{\nu} \tag{2.10c}$$

It is important to note that in Eq:2.10 both the operand $\vec{\nu}$ and output vector $\vec{\nu}\,'$ are both in the inertial frame. A *transformation* of a vector from the inertial to the body frame is the negative counterpart of Eq:2.10, a distinction which is not always explicitly specified.

$$\vec{\nu}_b = R_I^b(-\eta)\vec{\nu}_I \triangleq R_I^b(-\phi,-\theta,-\psi)\vec{\nu}_I \qquad \vec{\nu}_b \in \mathcal{F}^b,\quad \vec{\nu}_I \in \mathcal{F}^I \tag{2.11a}$$

$$\therefore \vec{\nu}_b = R_z(-\psi)R_y(-\theta)R_x(-\phi)\vec{\nu}_I \tag{2.11b}$$

$$= R_x(\phi)R_y(\theta)R_z(\psi)\vec{\nu}_I = R_b^I\vec{\nu}_I \tag{2.11c}$$

$$R_I^b = \left(R_b^I\right)^{-1} \equiv \left(R_b^I\right)^T \tag{2.11d}$$

The relationship in Eq:2.11d 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 common frame $\vec{\eta}_b \in \mathcal{F}^b$ to define the relationship between Euler and angular rates. Exploiting vehicle frames 1 and 2, or rather $\mathcal{F}^{v1}$ and $\mathcal{F}^{v2}$, as intermediary frames to retrospectively describe frames after $R_x(\phi)$ and $R_y(\theta)$ operations and using the rotation matrix derivative from Eq:2.8. The angular velocity $\vec{\omega}_b$ is the time derivative of Euler angles in the body frame:

$$\vec{\eta} = \begin{bmatrix}\phi & \theta & \psi\end{bmatrix}^T \qquad \in \mathcal{F}^{I,v_1,v_2} \tag{2.12a}$$

$$\vec{\omega}_b = \begin{bmatrix}p & q & r\end{bmatrix}^T \triangleq \frac{d}{dt_b}\vec{\eta} \equiv \frac{d}{dt}\vec{\eta}_b \qquad \in \mathcal{F}^b \tag{2.12b}$$

$$\vec{\eta}_b \triangleq R_{v_2}^b(\phi)\,\vec{\phi} + R_{v_2}^b(\phi)R_{v_1}^{v_2}(\theta)\,\vec{\theta} + R_{v_2}^b(\phi)R_{v_1}^{v_2}(\theta)R_I^{v_1}(\psi)\,\vec{\psi} \qquad \in \mathcal{F}^b \tag{2.12c}$$

$$\therefore \vec{\omega}_b = \left[\dot{\vec{\phi}}\,\right]_\times R_{v2}^b(\phi) + R_{v2}^b(\phi)\left[\dot{\vec{\theta}}\,\right]_\times R_{v1}^{v2}(\theta) + R_{v2}^b(\phi)R_{v1}^{v2}(\theta)\left[\dot{\vec{\psi}}\right]_\times R_I^{v1}(\psi) \qquad \in \mathcal{F}^b \tag{2.12d}$$

With Euler vectors $\vec{\phi}$, $\vec{\theta}$ and $\vec{\psi}$ being axis projections onto $\hat{X}$, $\hat{Y}$ and $\hat{Z}$ axes respectively; $\phi \cdot \hat{\imath}$, $\theta \cdot \hat{\jmath}$ and $\psi \cdot \hat{k}$. The vehicle frames used for Eq:2.12b 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 ZYX rotation sequences were used. The Euler rate Eq:2.12f then simplifies to the formal relationship between two rotating frames, with $\vec{\omega}_b = [p \; q \; r]^T$:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} \equiv \begin{bmatrix} 1 & 0 & -sin(\theta) \\ 0 & cos(\phi) & sin(\phi)cos(\theta) \\ 0 & -sin(\theta) & cos(\phi)sin(\theta) \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{2.12e}$$

$$\therefore \vec{\omega}_b = \Psi(\eta)\dot{\vec{\eta}} \qquad \in \mathcal{F}^b \tag{2.12f}$$

$$\Psi(\eta) \triangleq \begin{bmatrix} 1 & 0 & -sin(\theta) \\ 0 & cos(\phi) & sin(\phi)cos(\theta) \\ 0 & -sin(\theta) & cos(\phi)sin(\theta) \end{bmatrix} \tag{2.12g}$$

$$\therefore \dot{\vec{\eta}} = \Psi^{-1}(\eta)\vec{\omega}_b \equiv \Phi(\eta)\vec{\omega}_b \qquad \in \mathcal{F}^{v1,v2,I} \tag{2.12h}$$

$$\Phi(\eta) \triangleq \begin{bmatrix} 1 & sin(\phi)tan(\theta) & cos(\phi)tan(\theta) \\ 0 & cos(\phi) & -sin(\phi) \\ 0 & sin(\phi)sec(\theta) & cos(\phi)sec(\theta) \end{bmatrix} \tag{2.12i}$$

The *Euler* matrix $\Psi(\eta)$ contains a well known and problematic singularity at $\theta = \pm 90°$; where the determinant of the Euler transformation matrix is zero. The mathematical manifestation of that singularity and its physical consequences are expanded on in Sec:3.3.1. The singularity is present in the middle roll angle θ, which is a direct consequence of the chosen ZYX 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:3.3.2). Attitude in $\mathbb{R}^3$, or $SO(3)$, is intuitive and well suited to the conventions defined here.

Quaternions (Sec:3.3.2), despite being in $\mathbb{R}^4$, are similarly constructed in the ZYX 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, and some vector $\vec{\nu}_I$ in the inertial frame is related to the body frame $\mathcal{F}^b$ as follows:

$$\vec{\nu}_b = R_I^b(-\eta)\vec{\nu}_I \underset{Q}{\Longleftrightarrow} Q_b \otimes \begin{bmatrix} 0 & \vec{\nu}_I \end{bmatrix}^T \otimes Q_b^* \tag{2.13a}$$

$$Q_b \triangleq Q_z \otimes Q_y \otimes Q_x \;\; \text{and its inverse} \;\; Q_b^* \triangleq Q_x^* \otimes Q_y^* \otimes Q_z^* \tag{2.13b}$$

The symbol $\otimes$ represents the Hamilton product, or quaternion multiplication operator. Later the Hamilton product is used again for inertia tensor transformations (Sec:2.3). Each quaternion, $Q_{\hat{\imath}}$, is always the *unit* quaternion about the $\hat{\imath}^{th}$ axis. For the body quaternion, Q_b, it is the unit quaternion rotation about the body's Euler axis, [76]. A quaternion rotation operates on an argument vector with a zero quaternion scalar component. So then for some vector $\vec{\nu}$, the quaternion rotation operation in Eq:2.13a is equivalent to;

$$Q_{\vec{\nu}\,'} = Q \otimes (Q_{\vec{\nu}}) \otimes Q^* \tag{2.14a}$$

$$\text{Where } Q_{\vec{\nu}} \triangleq \begin{bmatrix} 0 & \vec{\nu} \end{bmatrix}^T \text{ and } Q_{\vec{\nu}\,'} \triangleq \begin{bmatrix} 0 & \vec{\nu}\,' \end{bmatrix} \tag{2.14b}$$

Quaternion representation in Eq:2.14b 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:2.14a is then simply:

$$\vec{\nu}\,' = Q \otimes (\vec{\nu}\,) \otimes Q^* \tag{2.15}$$

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 in Chapter:3.

2.2.2 Motor Axis Layout

The whole structure (previously in Fig:2.1) consists of multiple rigidly connected bodies with only relative rotations between each body permitted by its revolute joints, illustrated in the design description in Sec:2.1. Those rigid bodies are categorized into four inter-connected motor modules $\mathbf{M}_{1,2,3,4}$ or $\mathbf{M}_i,\ i \in [1:4]$ 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.

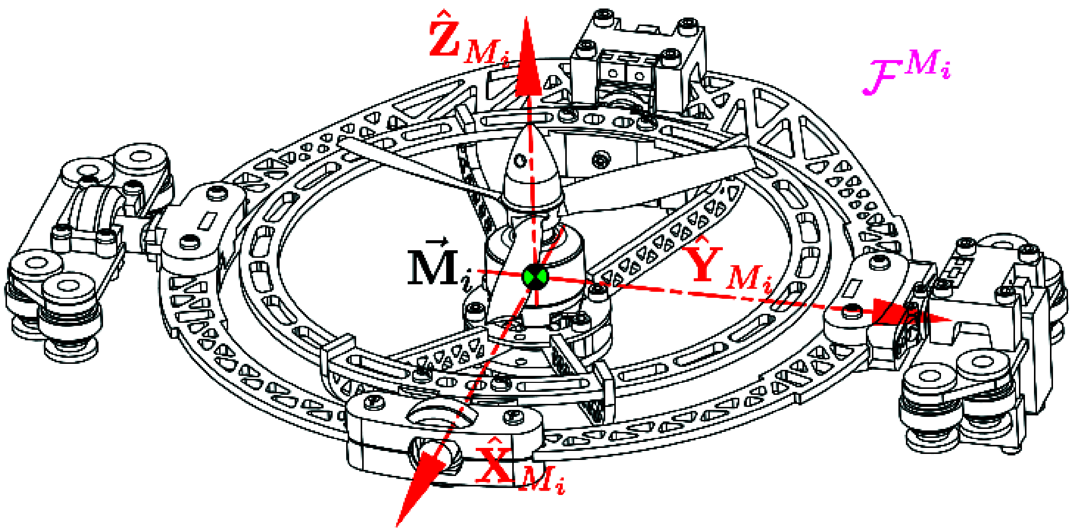


Figure 2.7: Aligned motor frame axes

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. A motor module frame $\mathcal{F}^{M_i}$ is attached to the innermost ring, the BLDC motor's stator is affixed to that frame and its rotor has a rotational velocity Ω_i about the $\hat{Z}_{M_i}$ stator axis. Fig:2.8 shows the sequential relative module frames.

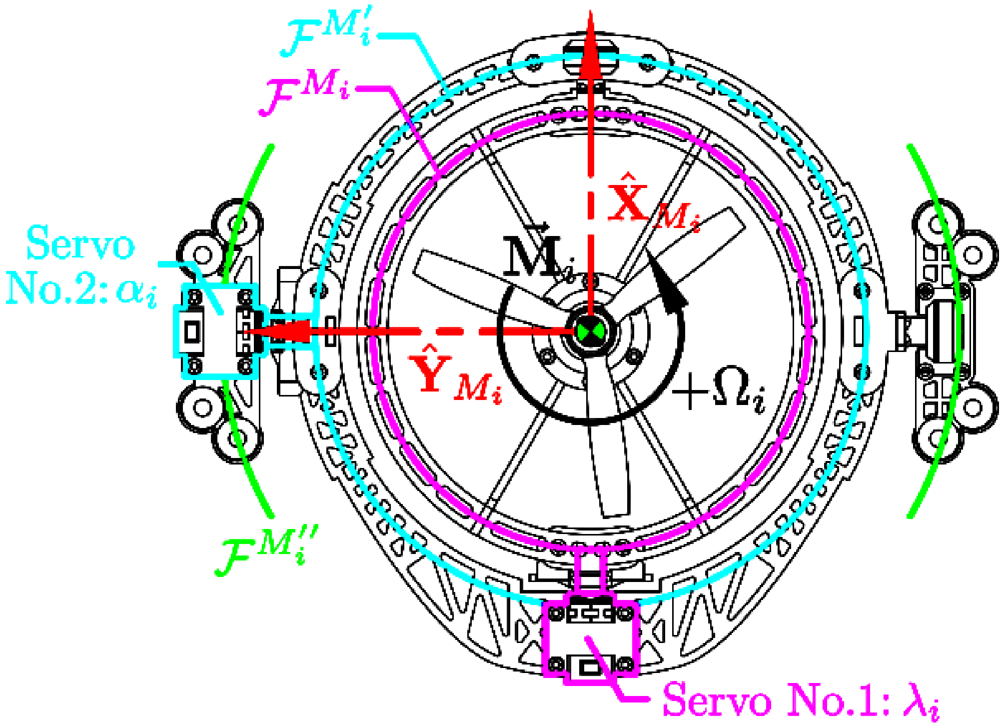


Figure 2.8: Intermediate motor frames

That inner ring frame rotates about its $\hat{X}_{M_i}$ axis by an angle λ_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 and frame then rotates by an angle α_i about its $\hat{Y}_{M_i'}$ axis actuated by the second servo. That second servo is affixed to an intermediate $\mathcal{F}^{M_i''}$ frame, finally there's an orthogonal rotation about that intermediate frame's $\hat{Z}_{M_i''}$ axis to the body frame $\mathcal{F}^b$. Each module's actuation state is fully described by the propeller's rotational speed Ω_i, both servo positions λ_i and α_i and all their respective rates; $u_i \triangleq [\Omega_i,\ \lambda_i,\ \alpha_i,\ \dot{\Omega}_i,\ \dot{\lambda}_i,\ \dot{\alpha}_i]^T$ 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 $\vec{\mathbf{O}}_b$ at the $\hat{X}_b$ and $\hat{Y}_b$ intersect of the structure, co-planar to each motor modules' center. *Neither* the body frame's origin *nor* each modules center of rotation are coincidental with body's center of mass. The exact disparity between the origin(s) of motion and the respective body's center of mass are quantified subsequently in Sec:2.3.

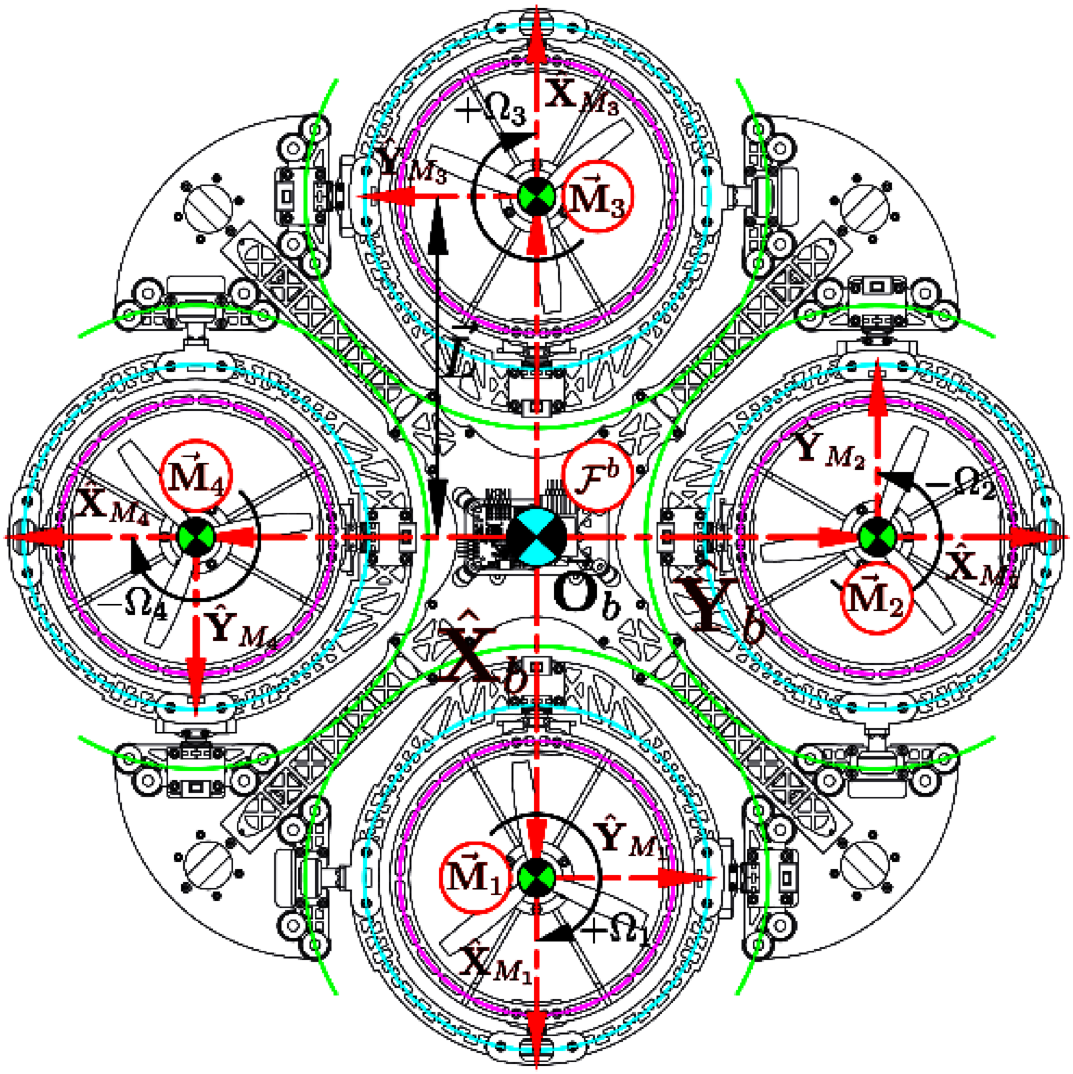


Figure 2.9: Body frame axes layout

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

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

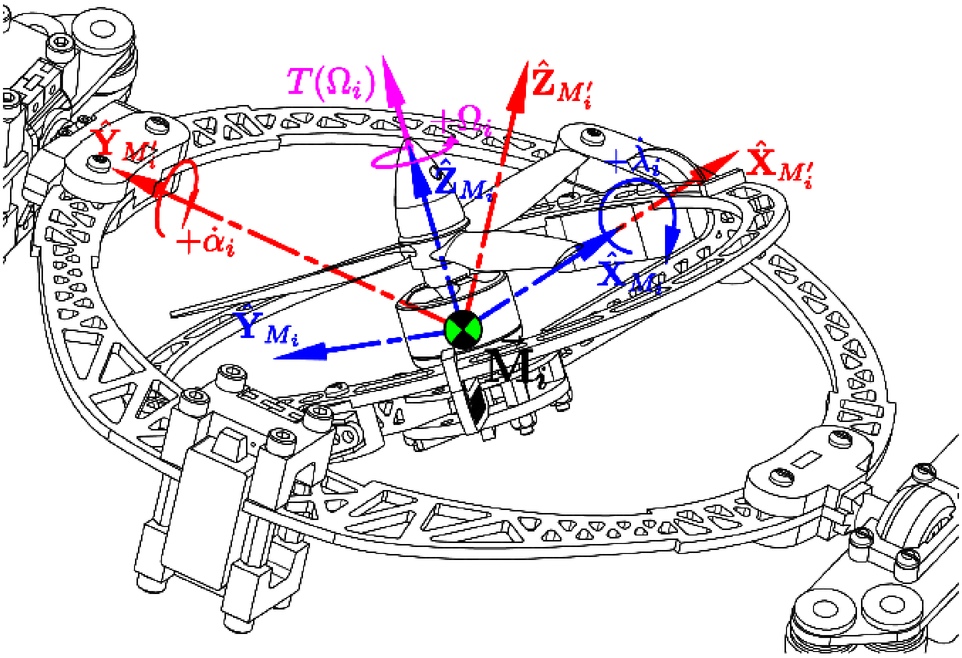


Figure 2.10: Motor thrust force

Each motor module's rotational center $\vec{\mathbf{M}}_i$ is displaced from the body frame origin $\vec{\mathbf{O}}_b$ by the distance $L_{arm} = 195.16$ [mm] (shown in Fig:2.9). Transformation of some vector $\vec{\nu}_{M_i}$ in the motor frame $\mathcal{F}^{M_i}$ to the body frame $\mathcal{F}^b$ is given as three sequential rotation operations:

$$\vec{\nu}_b = R^b_{M_i}\vec{\nu}_{M_i} = R_z(-\sigma_i)R_y(-\alpha_i)R_x(-\lambda_i)\vec{\nu}_{M_i} \qquad \in \mathcal{F}^b, \text{ for } \sigma_i \in \begin{bmatrix}0 & \frac{\pi}{2} & \pi & \frac{2\pi}{3}\end{bmatrix} \tag{2.16a}$$

The constant orthogonal σ_i rotations about $\hat{Z}_{M_i''}$ are independent of actuator positions, σ_i is determined by the motor module's location, illustrated in Fig:2.9. The rotation matrices $R_z(\sigma_i)$ for $\sigma_i = (i-1)\pi/2,\ i \in [1:4]$ are:

$$R_z = \begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix}, \begin{bmatrix}0 & -1 & 0\\1 & 0 & 0\\0 & 0 & 1\end{bmatrix}, \begin{bmatrix}-1 & 0 & 0\\0 & -1 & 0\\0 & 0 & 1\end{bmatrix}, \begin{bmatrix}0 & 1 & 0\\-1 & 0 & 0\\0 & 0 & 1\end{bmatrix} \text{ for } i \in [1:4] \text{ respectively} \tag{2.16b}$$

If the propeller's rotation Ω_i produces some thrust force $T(\Omega_i)$ in the motor module frame (Fig:2.10) which acts through the center of rotation $\vec{\mathbf{M}}_i$; that force is similarly transformed to the body frame through Eq:2.16a. A thrust vector for $\vec{T}_i \in \mathcal{F}^{M_i}$ in the body frame $\mathcal{F}^b$ is calculated:

$$\vec{T}_i = R_z(-\sigma_i)R_y(-\alpha_i)R_x(-\lambda_i)\begin{bmatrix}0 & 0 & T(\Omega_i)\end{bmatrix}^T \qquad \in \mathcal{F}^b \tag{2.17}$$

The actuator space, including propeller speed Ω_i, 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:

$$\underset{\in\mathbb{U}}{u} = \begin{bmatrix}\Omega_1^+ & \lambda_1 & \alpha_1 & \dots & \Omega_4^- & \lambda_4 & \alpha_4\end{bmatrix}^T \qquad \in \mathbb{R}^{12} \tag{2.18}$$

2.3 Inertial Matrices & Masses

When transforming inertias between reference frames it is more appropriate to use rotation matrices to apply the transformation and not quaternions. Spatial rotation of inertial matrices are ill suited to quaternion parametrization.

An undesirable consequence of 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 principally 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 (Sec:3.1), such first and second order effects are negligible given that the angular rates on which they depend are small enough to approximate as zero; $\vec{\omega}_b \approx \vec{0}$. A dynamic setpoint (non-zero) attitude tracking plant is, however, going to produce time varying body angular velocities and accelerations that must be accounted for.

The dynamic effects of those torque responses are derived later in Sec:3.4.1. Both inertial and gyroscopic effects are dependent on the considered body's rotational inertia about each respective axis. The magnitude of those inertias are ostensibly a by-product of the structure's design but also the vehicle's instantaneous configuration.

The following inertias presented are all calculated from a SolidWorks model with masses to match physical measurements taken of the constructed prototype. Each connected body affected by the same angular velocity is grouped together. Every motor module then contains 3 independent inertial bodies; the propeller/rotor body, the inner ring and finally middle ring assemblies, each of which are now described in detail.

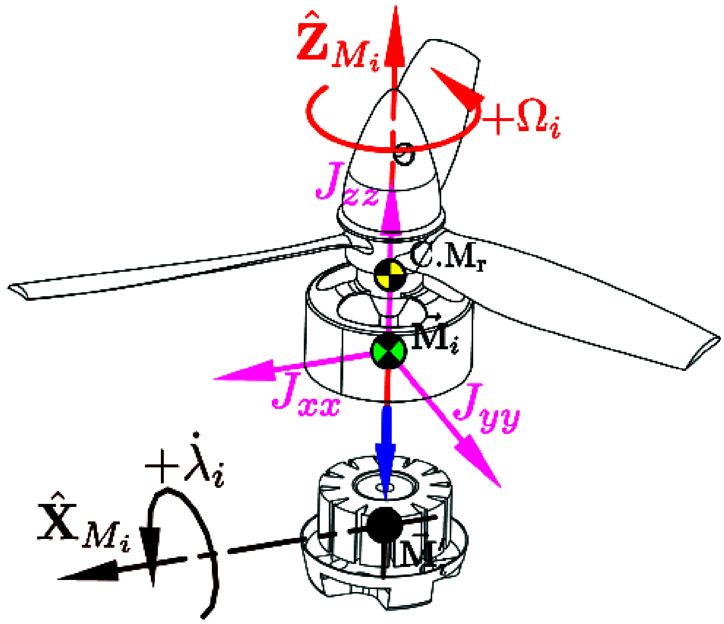


Figure 2.11: Rotor assembly rotational structure

The first rotational body to consider is that of the propeller and rotor assembly (Fig:2.11, excluding the motor's stator). The *rotor* assembly, with subscript r, has a net mass $m_{\mathrm{r}} = 27$ [g] with a center of mass $\mathrm{C_r} = \begin{bmatrix} 0.0 & 0.0 & 15.5 \end{bmatrix}^T$ [mm] relative to the entire motor modules center of rotation $\vec{\mathbf{M}}_i$. The propeller's rotation plane is similarly $\begin{bmatrix} 0.0 & 0.0 & 23.0 \end{bmatrix}^T$ [mm] relative to $\vec{\mathbf{M}}_i$ (previously illustrated in Fig:2.3).

At high speeds the propeller's inertia contribution to the rotor assembly can be approximated as a solid disc. It follows that the inner ring's inertial components can then be regarded as constant with respect to Ω_i; moreover its center of mass is independent of that propeller's rotation.

The entire rotor assembly then has a rotational constant inertia J_{r}, with principle inertial axes centered and aligned as in Fig:2.11:

$$J_{\mathrm{r}} = \begin{bmatrix} 105.5 & 0.0 & 0.0 \\ 0.0 & 105.5 & 0.0 \\ 0.0 & 0.0 & 41.8 \end{bmatrix} \quad [\mathrm{g.cm}^2] \tag{2.19}$$

The net angular velocity of the rotor assembly $\vec{\omega}_{r/b}$ relative to the body frame is produced by the BLDC motor's rotational velocity Ω_i and both servo rates; $\dot{\lambda}_i$ and $\dot{\alpha}_i$. Here Ω_i and both servo rates are measured in rad.s^{-1}, later Ω_i is used in rev.s^{-1} for Blade-element momentum theory thrust calculations (Sec:3.2.1). Each servo's angular velocity is *transformed* onto the motor frame $\mathcal{F}^{M_i}$.

$$\vec{\omega}_{r/b} = \begin{bmatrix} 0 \\ 0 \\ \Omega_i \end{bmatrix} + \frac{d\lambda_i}{dt} R_x(-\lambda_i) \begin{bmatrix} \lambda_i \\ 0 \\ 0 \end{bmatrix} + \frac{d\alpha_i}{dt} R_y(-\alpha_i) R_x(-\lambda_i) \begin{bmatrix} 0 \\ \alpha_i \\ 0 \end{bmatrix} \quad \in \mathcal{F}^{M_i} \tag{2.20}$$

Eq:2.20 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)...

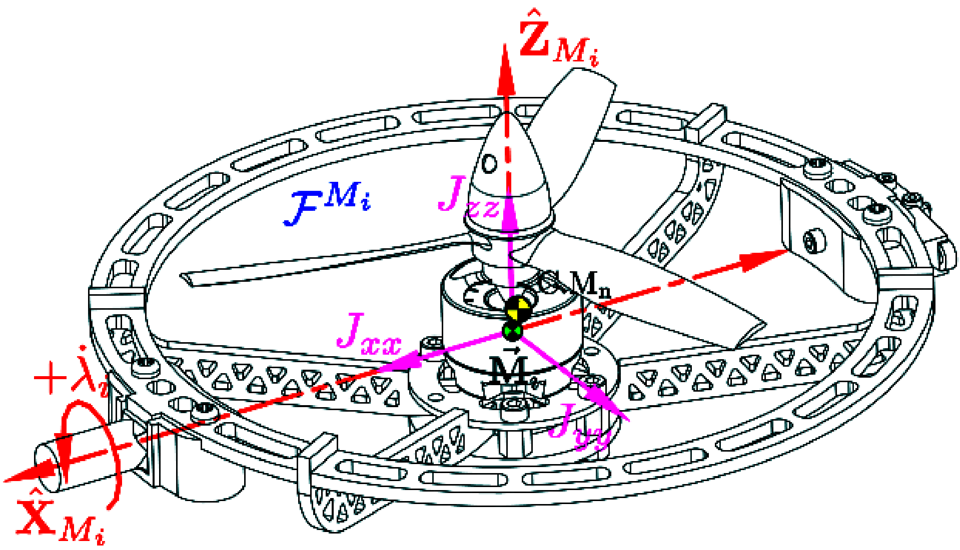


Figure 2.12: Inner ring rotational structure

The next assembly, to which the motor frame $\mathcal{F}^{M_i}$ is attached, is the *inner ring* assembly denoted with subscript n. The inner ring structure has a mass $m_{\mathrm{n}} = 92$ [g], <u>including</u> the rotor assembly in that calculation. The center of mass is positioned $\mathrm{C_n} = \begin{bmatrix} -1.44 & 00.0 & 5.14 \end{bmatrix}^T$ [mm] relative to the module's center of rotation $\vec{\mathbf{M}}_i$. The inner ring, being rotated by the λ_i servo about the $\hat{X}_{M_i}$ axis, then has an inertial matrix which <u>includes</u> J_{r} from Eq:2.19 centered and aligned with axes as in Fig:2.12:

$$J_{\mathrm{n}} = J_{M_i} = \begin{bmatrix} 520.9 & -31.7 & -0.3 \\ -31.7 & 1826.3 & 0.0 \\ -0.3 & 0.0 & 2050.8 \end{bmatrix} \quad [\mathrm{g.cm}^2] \tag{2.21}$$

The rotational velocity of the collective inner ring assembly $\vec{\omega}_{n/b}$ for the angular velocity of frame $\mathcal{F}^{M_i}$, is similar to that of Eq:2.20. They both occur in the same frame however the inner ring's angular velocity has no velocity contribution from Ω_i:

$$\vec{\omega}_{n/b} = \frac{d\lambda_i}{dt} R_x(-\lambda_i) \begin{bmatrix} \lambda_i \\ 0 \\ 0 \end{bmatrix} + \frac{d\alpha_i}{dt} R_y(-\alpha_i) R_x(-\lambda_i) \begin{bmatrix} 0 \\ \alpha_i \\ 0 \end{bmatrix} \quad \in \mathcal{F}^{M_i} \tag{2.22}$$

That first actuating servo for λ_i and its coaxial support bearing are both affixed to the intermediate *middle ring* assembly, with subscript m (middle ring only Fig:2.13). The intermediate frame $\mathcal{F}^{M_i'}$ is attached to the middle ring body with a mass $m_{\mathrm{m}} = 98$ [g], excluding the inner most ring's contribution. That middle ring body alone has a center of mass $\mathrm{C_m} = \begin{bmatrix}-4.70 & 0.37 & -0.36\end{bmatrix}^T$ [cm] relative to $\vec{\mathbf{M}}_i$.

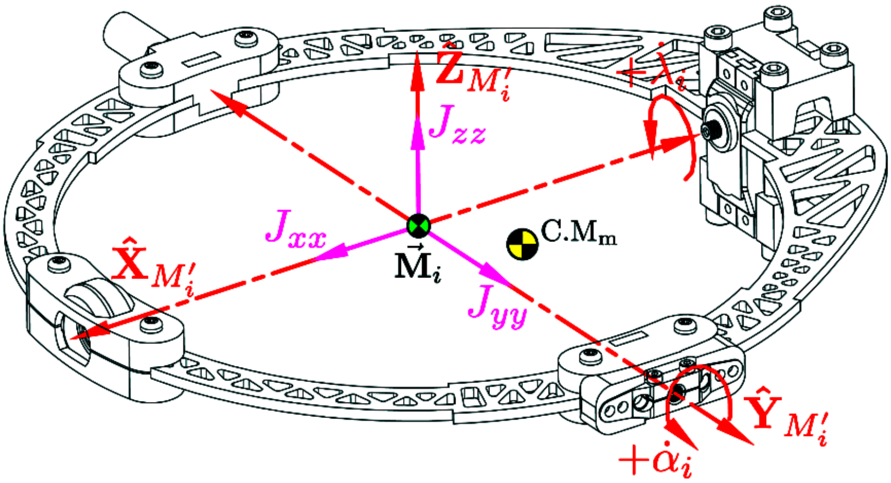


Figure 2.13: Middle ring rotational structure

Together the inner and middle rings make the whole motor module assembly (Fig:2.14), with a subscript p. The net module has a mass $m_{\mathrm{p}} = 190$ [g]. The center of mass for the entire module $\mathrm{C_p}$ is a function of the inner ring's rotational position λ_i relative to the middle frame $\mathcal{F}^{M_i'}$. That module's center of mass is calculated:

$$\mathrm{C_n'}(\lambda) \triangleq R_x(\lambda)\big(\mathrm{C_n}\big) \tag{2.23a}$$

$$\mathrm{C_p}(\lambda) \triangleq \frac{m_{\mathrm{m}}\big(\mathrm{C_m}\big) + m_{\mathrm{n}}\big(\mathrm{C_n'}(\lambda)\big)}{m_{\mathrm{p}}} \tag{2.23b}$$

Substituting physical values into Eq:2.23b for the inner and middle rings' center of masses respectively:

$$\mathrm{C_p}(\lambda) = \frac{98\begin{bmatrix}-4.70 & 0.37 & -0.36\end{bmatrix}^T \times 10^{-7} + 92R_x(\lambda)\begin{bmatrix}-1.44 & 0.00 & 3.06\end{bmatrix}^T \times 10^{-8}}{190 \times 10^{-3}} \tag{2.23c}$$

Which then has a value at rest, for reference, with the servo $\lambda_i = 0^\circ$ relative to the center of rotation $\vec{\mathbf{M}}_i$:

$$\mathrm{C_p}(0) = \begin{bmatrix}-2.49 & 0.19 & 0.04\end{bmatrix}^T\Big|_{\lambda_i=0} \quad [\mathrm{cm}] \tag{2.23d}$$

The complete motor module is finally rotated by the α_i servo about its $\hat{Y}_{M_i'}$ axis. The module's compound inertia J_{p} is a combination of the middle ring's inertia J_{m} and the inner ring's inertia J_{n} rotated by λ_i about $\hat{X}_{M_i}$ (Fig:2.14). The latter's contribution is dependent on the *rotation*, not transformation, angle λ_i which from the conservation of angular momentum theory, detailed in [107]. The motor module's net rotational inertial J_{p}, is then calculated from J_{m}:

$$\text{With} \quad J_{\mathrm{m}} = \begin{bmatrix}2905.7 & 0.0 & 390.9 \\ 0.0 & 8446.4 & 0.0 \\ 390.9 & 0.0 & 11125.7\end{bmatrix} \quad [\mathrm{g.cm^2}] \tag{2.24a}$$

$$J_{\mathrm{p}}(\lambda_i) \triangleq J_{\mathrm{m}} + R_x(\lambda_i)\big(J_{\mathrm{n}}\big)R_x^{-1}(\lambda_i) \tag{2.24b}$$

That net inertia for the complete motor module, with $\lambda_i = 0°$ and relative to the middle ring frame $\mathcal{F}^{M_i'}$, has a reference value:

$$J_{\mathrm{p}}(0) = \begin{bmatrix} 3365.4 & -0.1 & 390.6 \\ -0.1 & 10210.1 & 0.0 \\ 390.6 & 0.0 & 13118.0 \end{bmatrix}\Bigg|_{\lambda_i=0°} \quad [\mathrm{g.cm}^2] \tag{2.24c}$$

The rotation matrix R_x in Eq:2.24b is a full rank square matrix, its inverse R_x^{-1} always exists. The module's inertia could be further divided into constant and variable components; $J_{\mathrm{p}}(\lambda_i) = J_{const} + J_{M_i}(\lambda_i)$. The variable terms, if small enough or under certain conditions, could be simplified or neglected...

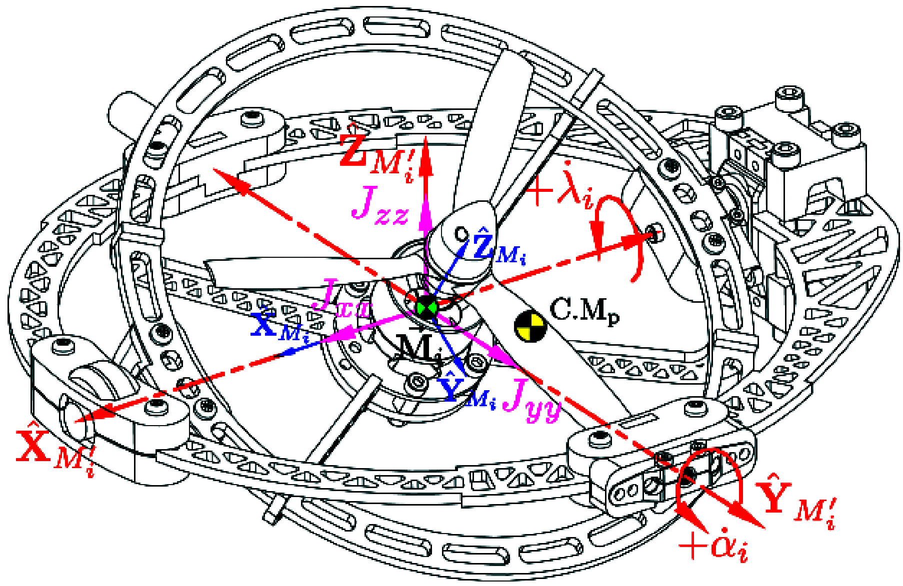


Figure 2.14: Module assembly rotational structure

Fig:2.15 shows how the complete motor module and its rotational axes (in Fig:2.14) are attached and centered relative to the body structure. The second α_i servo is fixed to the body structure and rotates the entire motor module about the $\hat{Y}_{M_i''}$ axis.

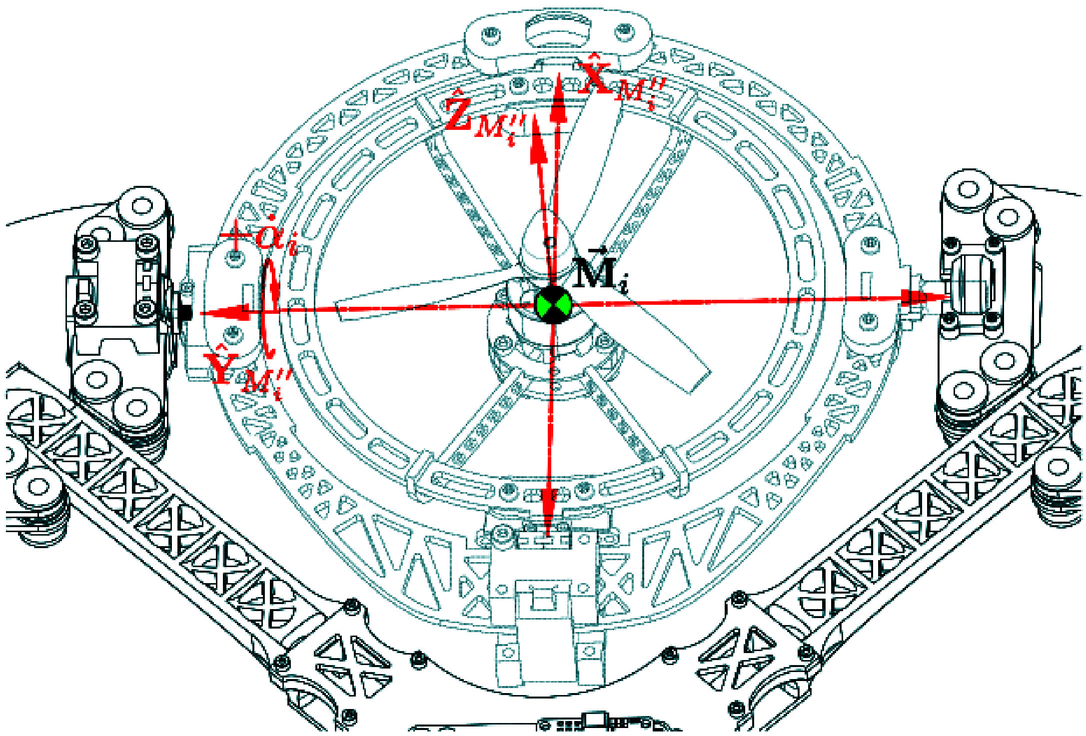


Figure 2.15: Complete motor module attached to the body structure

Finally, the angular velocity experienced by the net motor assembly relative to the body frame, $\vec{\omega}_{p/b}$ in frame $\mathcal{F}^{M_i'}$, is entirely as a result of the α_i servo actuation:

$$\vec{\omega}_{p/b} = \frac{d\alpha_i}{dt} R_y(-\alpha_i) \begin{bmatrix} 0 \\ \alpha_i \\ 0 \end{bmatrix} \qquad \in \mathcal{F}^{M_i'} \tag{2.24d}$$

That α_i servo is affixed to the body structure and so its inertial volume and that of the outer coaxial bearing support contributes then to the body structure's inertia; whose value excludes any of the four motor modules. Attached to that servo is an intermediate frame $\mathcal{F}^{M_i''}$ (Fig:2.15) which differs from the middle ring frame by an $R_y(-\alpha_i)$ transformation and differs from the body frame F^b by an orthogonal $R_z(\sigma_i)$ rotation.

The motor modules are suspended from the body frame with a set of silicone damping balls. The *body structure* which includes those connecting masses, with a subscript y, has center of mass C.M$_\mathrm{y}$ (without any motor modules attached, Fig:2.16). 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 $\vec{\mathbf{O}}_b \in \mathcal{F}^b$.

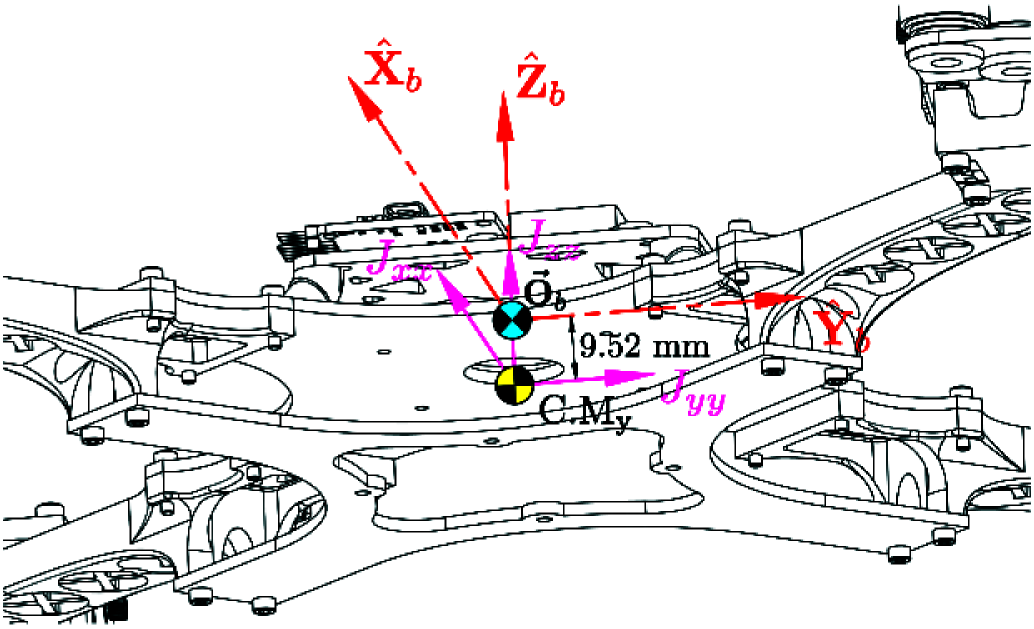


Figure 2.16: Body structure's center of mass

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.

The body structure's weight, including all four damping assemblies and electronics, totals to $m_\mathrm{y} = 814.70$ g. Similarly the body structure's net inertia (*sans* motor modules) J_y, about its center of mass (Fig:2.16), is:

$$\underset{C.M}{J_\mathrm{y}} = \begin{bmatrix} 181569.7 & 0.4 & -19.4 \\ 0.4 & 181692.2 & 8.9 \\ -19.4 & 8.8 & 360067.2 \end{bmatrix} \times 10^{-7} \quad \mathrm{kg.m^2} \tag{2.25a}$$

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:

$$J' = J + m\big(\vec{d}\cdot\vec{d} - \vec{d}\otimes\vec{d}\,\big) \approx J + md^2 \tag{2.25b}$$

For the general parallel axis transformation in Eq:2.25b, $\otimes$ 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 $\mathbf{C.M}_y$ and the body frame origin $\vec{\mathbf{O}}_b$.

$$\therefore J_y' = \underset{C.M}{J_\mathrm{y}} + m_\mathrm{y}\big(\Delta\vec{Z}\cdot\Delta\vec{Z} - \Delta\vec{Z}\otimes\Delta\vec{Z}\big) \tag{2.25c}$$

That body's constant inertia J_{y} at the origin $\vec{\mathbf{O}}_b$ and aligned with the body frame $\mathcal{F}^b$ is then:

$$\rightarrow \underset{\vec{\mathbf{O}}_b}{J'_{\text{y}}} = \begin{bmatrix} 182307.7 & 0.4 & -14.5 \\ 0.4 & 182430.1 & 6.5 \\ -14.5 & 6.5 & 360067.2 \end{bmatrix} \times 10^{-7} \quad \text{kg.m}^2 \tag{2.25d}$$

Net inertia for the complete multibody vehicle, $J_b(u)$ about the origin $\vec{\mathbf{O}}_b$, is a combination of all the relative attached bodies as a function of all actuator positions $u \in \mathbb{U}$. The entire assembly's inertia $J_b(u)$ 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, each rotated first by λ_i, then α_i and finally translated to the body frame origin; and the body structure's contribution itself.

Those motor modules' inertial transformations from their respective centers of rotation, in frames $\mathcal{F}^{M_i}$ for $i \in [1:4]$, to the body frame $\mathcal{F}^b$ are analogous to that of Eq:2.16. Reiterating that $\vec{\mathbf{O}}_b$ is *co-planar* to each module's center of rotation; each motor module's inertia $J_{\text{p}}(\lambda_i)$, defined in Eq:2.24b, is further rotated by α_i about the $\hat{Y}_{M'_i}$ axis and finally an orthogonal $\hat{Z}_{M''_i}$ axis rotation (aligned with $\hat{Z}_b$) onto $\mathcal{F}^b$.

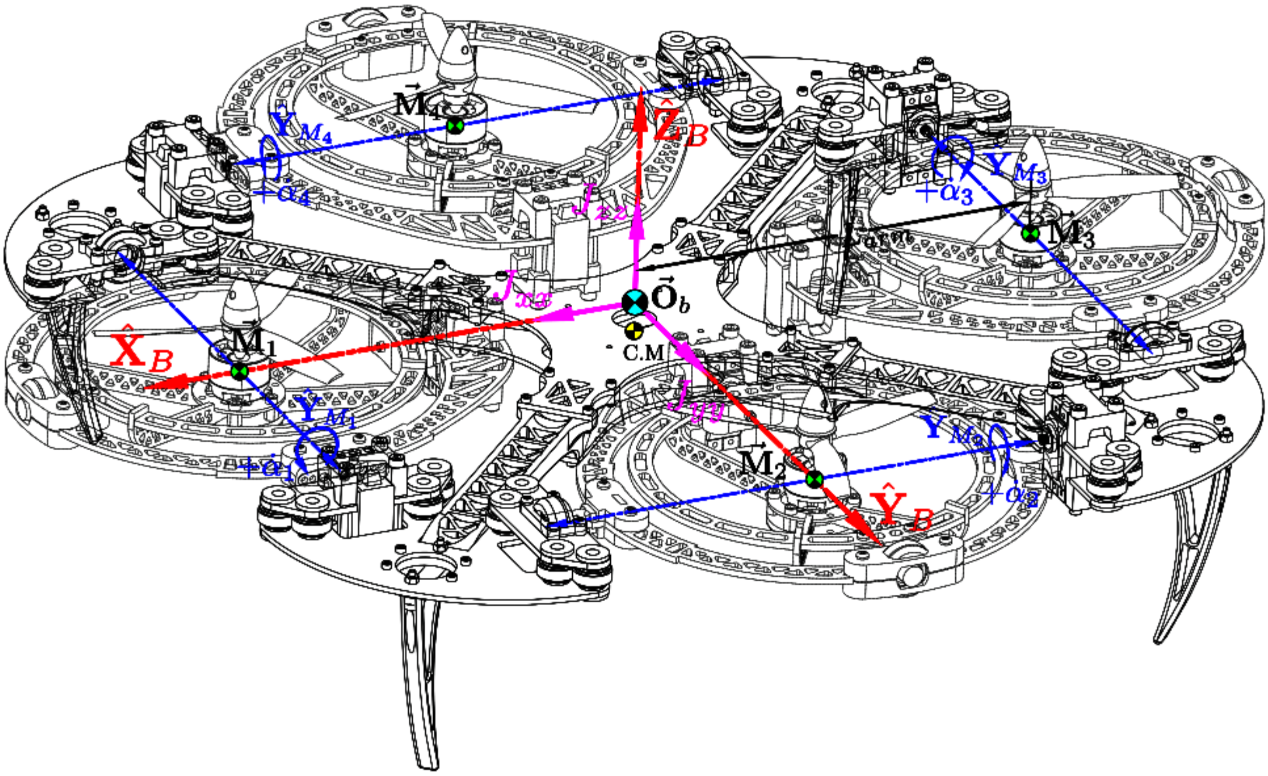


Figure 2.17: Inertial, mass and motor modules respective centers

For the entire body's net inertia each contributing assembly's inertia 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 center $\vec{\mathbf{M}}_i$ in Fig:2.17, but re-orientated to align parallel TO $\vec{\mathbf{O}}_b$ with rotations about axes $\hat{X} \in \mathcal{F}^{M_i}$, $\hat{Y} \in \mathcal{F}^{M'_i}$, $\hat{Z} \in \mathcal{F}^{M''_i}$, is calculated:

$$J_{\vec{\mathbf{M}}_i}(u \cdot i) = R_z(\sigma_i) R_y(\alpha_i) \big(J_p(\lambda_i)\big) R_y^{-1}(\alpha_i) R_z^{-1}(\sigma_i) \text{ for } i \in [1:4] \tag{2.26a}$$

The argument $(u \cdot i)$ in Eq:2.26a is the i$^{\text{th}}$ projection of the actuator space; that being $\begin{bmatrix}\Omega_i & \lambda_i & \alpha_i\end{bmatrix}^T$. Furthermore the rotation $R_z(\sigma_i)$ was defined as an orthogonal $\hat{Z}_b$ rotation previously in Eq:2.16b. Expanding each module's inertia to individual inner and middle ring inertial contributions then yields:

$$\therefore J_{\vec{\mathbf{M}}_i}(u \cdot i) = R_z R_y(\alpha_i) \big(J_{\text{m}}\big) R_y^{-1}(\alpha_i) R_z^{-1} + R_z R_y(\alpha_i) R_x(\lambda_i) \big(J_{\text{n}}\big) R_x^{-1}(\lambda_i) R_y^{-1}(\alpha_i) R_z^{-1} \tag{2.26b}$$

It is 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.

Each module's rotational center, vectors $\vec{\mathbf{M}}_{1\to4}$, are all equally spaced relative to the origin of motion, $\vec{\mathbf{O}}_b$, with a parallel axis arm $L_{arm} = 195.16$ mm (Fig:2.17). To avoid notational confusion the term $\vec{L}_i = \begin{bmatrix}\pm195.16 & 0 & 0\end{bmatrix}^T$ or $\begin{bmatrix}0 & \pm195.16 & 0\end{bmatrix}^T$ is used to represent the vector displacement between the origin $\vec{\mathbf{O}}_b$ and each motor modules center of rotation $\vec{\mathbf{M}}_{1\to4}$. The net inertial equation $J_b(u)$, about the origin $\vec{\mathbf{O}}_b$ and depending on the actuator position matrix $u \in \mathbb{U}$, can be calculated as:

$$\underset{\vec{\mathbf{O}}_b}{J_b}(u) = J_{\mathrm{y}} + \sum_{i=1}^{4} J'_{\vec{\mathbf{M}}_i}(u \cdot i) \quad \mathrm{kg.m}^2,\ u \in \mathbb{U} \tag{2.27a}$$

Where $J'_{\vec{\mathbf{M}}_i}(u \cdot i)$ is the motor module inertia from Eq:2.26 but translated to the origin $\vec{\mathbf{O}}_b$ using a parallel axis theorem with $m_{\mathrm{p}} = 190$ g and the displacement vector $\vec{L}_i$:

$$J'_{\vec{\mathbf{M}}_i}(u \cdot i) = J_{\vec{\mathbf{M}}_i}(u \cdot i) + m_{\mathrm{p}}\big(\vec{L}_i \cdot \vec{L}_i - \vec{L}_i \otimes \vec{L}_i\big) \tag{2.27b}$$

Although Eq:2.27 produces the net multi-body's inertia, each equation to calculate $J'_{\vec{\mathbf{M}}_i}$ involves cascaded transformations which may deteriorate the result's certainty. Each 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 the inertia contribution of each sub-assembly can be considered separately and translated directly to the body frame's origin from their respective mass centers. This will improve the accuracy of the produced inertial equations, each translation/rotation has with it an associated floating point concatenation. It is also perhaps more intuitive for the reader to consider each sub-body's contribution individually, despite having been derived as combined inertial bodies in the above. The vehicles net inertia can then be described as nine separate contributing bodies; four inner rings J_{n}, four middle rings J_{m} and one body structure J_{y}:

$$\underset{\vec{\mathbf{O}}_b}{J_b}(u) = \underset{\vec{\mathbf{O}}_b}{J'_{\mathrm{y}}} + \sum_{i=1}^{4} \underset{\vec{\mathbf{O}}_b}{J_{\mathrm{n}}}(u \cdot i) + \sum_{i=1}^{4} \underset{\vec{\mathbf{O}}_b}{J_{\mathrm{m}}}(u \cdot i) \quad u \in \mathbb{U} \tag{2.28}$$

Note that the rotor's inertia J_r is included in J_n. Then isolating each body and considering each inertia independently. Starting with the inner ring's contribution; 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:2.21, calculating the inner ring's inertial contribution about the origin $\vec{\mathbf{O}}_b$.

For the inner ring only, with a mass m_{n} and center of mass $C.M_{\mathrm{n}}$ relative to its center of rotation $\vec{\mathbf{M}}_i$. The inner ring's directly transformed inertial contribution then follows:

$$m_{\mathrm{n}} = 92 \quad \mathrm{g} \tag{2.29a}$$

$$C.M_{\mathrm{n}} = \begin{bmatrix}-1.44 & 0.0 & 5.14\end{bmatrix}^T \quad \mathrm{mm},\ \in \mathcal{F}^{M_i} \tag{2.29b}$$

The inner ring's inertial matrix about its center of mass (Fig:2.12) is the constant:

$$\underset{C.M}{J_{\mathrm{n}}} = \begin{bmatrix}496.6 & -31.7 & 6.6\\ -31.7 & 1800.1 & 0.0\\ 6.6 & 0 & 2048.9\end{bmatrix} \quad \mathrm{g.cm}^2 \tag{2.29c}$$

Relative to the body frame's origin $\vec{\mathbf{O}}_b$ the inner ring has a center of mass, rotated by λ_i and α_i servos about their respective axes with a relative orthogonal R_z rotation with σ_i for the i^{th} module implied:

$$C.M'''_{\mathrm{n}}(\lambda_i, \alpha_i) = R_z R_y(\alpha_i) R_x(\lambda_i)\big(C.M_{\mathrm{n}}\big) \quad \in \mathcal{F}^b \tag{2.29d}$$

Transforming the inertia from Eq:2.29c, still about the center of mass $C.M_{\mathrm{n}}'''$, but with axes aligned parallel with to body frame, or using the shorthand $||\vec{\mathbf{O}}_b$. The inner ring's inertia as a function of both servo angles λ_i and α_i aligned with $\mathcal{F}^b$ is:

$$\underset{||\vec{\mathbf{O}}_b}{J_{\mathrm{n}}'''}(\lambda_i, \alpha_i) = R_z R_y(\alpha_i) R_x(\lambda_i) \big(J_{\mathrm{n}}\big) R_x^{-1}(\lambda_i) R_y^{-1}(\alpha_i) R_z^{-1} \tag{2.29e}$$

The vector difference between the rotated center of mass $C.M_{\mathrm{n}}'''$ with the body origin $\vec{\mathbf{O}}_b$ is:

$$\Delta L = \vec{L}_i - C.M_{\mathrm{n}}'''(\lambda_i, \alpha_i) \tag{2.29f}$$

Then using the above with a parallel axis translation, adapted from Eq:2.25b, to move the rotated inertia J_{n}''' to the center of the body frame $\vec{\mathbf{O}}_b$:

$$\underset{\vec{\mathbf{O}}_b}{J_{\mathrm{n}}}(\lambda_i, \alpha_i) = \underset{||\vec{\mathbf{O}}_b}{J_{\mathrm{n}}'''}(\lambda_i, \alpha_i) + m_{\mathrm{n}}\big((\Delta L \cdot \Delta L)\mathbb{I}_{3\times 3} - \Delta L \otimes \Delta L\big) \tag{2.29g}$$

And for reference when both servos are at rest; $\lambda_i = 0$ and $\alpha_i = 0$, the inner ring's inertial contribution about the origin is explicitly:

$$\underset{\vec{\mathbf{O}}_b}{J_{\mathrm{n}}}(\lambda_i, \alpha_i) = \begin{bmatrix} 520.9 & -31.0 & 922.6 \\ -31.0 & 36348.5 & 0.0 \\ 922.6 & 0.0 & 36573.0 \end{bmatrix} \times 10^{-7} \Bigg|_{\lambda_i, \alpha_i = 0} \quad \mathrm{kg.m}^2, \ \in \mathcal{F}^b \tag{2.29h}$$

Similarly, the same process is applied for the middle ring's rotated and translated inertia. The middle ring *only* (Fig:2.13) has a mass and center of mass relative to the module's center of rotation respectively:

$$m_{\mathrm{m}} = 98 \quad \mathrm{g} \tag{2.30a}$$

$$C.M_{\mathrm{m}} = \begin{bmatrix} -47.00 & 3.74 & -3.63 \end{bmatrix}^T \quad \mathrm{mm}, \ \in \mathcal{F}^{M_i'} \tag{2.30b}$$

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

$$\underset{C.M}{J_{\mathrm{m}}} = \begin{bmatrix} 2879.1 & 172.3 & 223.6 \\ 172.3 & 6269.0 & 13.3 \\ 223.6 & 13.3 & 8947.5 \end{bmatrix} \quad \mathrm{g.cm}^2 \tag{2.30c}$$

Rotating the center of mass only by the α_i servo about the $\hat{Y}_{M_i'}$ axis yields the center of mass $C.M_{\mathrm{m}}'$ relative to $\vec{\mathbf{O}}_b$:

$$C.M_{\mathrm{m}}''(\alpha_i) = R_z R_y(\alpha_i)\big(C.M_{\mathrm{m}}\big) \quad \in \mathcal{F}^b \tag{2.30d}$$

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

$$\underset{||\vec{\mathbf{O}}_b}{J_{\mathrm{m}}''}(\alpha_i) = R_z R_y(\alpha_i)\big(J_{\mathrm{m}}\big) R_y^{-1}(\alpha_i) R_z^{-1} \tag{2.30e}$$

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

$$\Delta L = \vec{L}_i - C.M_{\mathrm{m}}''(\alpha_i) \tag{2.30f}$$

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

$$\underset{\vec{\mathbf{O}}_b}{J_{\mathrm{m}}}(\alpha_i) = \underset{||\vec{\mathbf{O}}_b}{J_{\mathrm{m}}''}(\alpha_i) + m_{\mathrm{m}}\big((\Delta L \cdot \Delta L)\mathbb{I}_{3x3} - \Delta L \otimes \Delta L\big) \tag{2.30g}$$

Again for reference; at rest with the middle ring servo $\alpha_i = 0$ the middle ring's inertial contribution at $\vec{\mathbf{O}}_b$ is:

$$\underset{\vec{\mathbf{O}}_b}{J_{\mathrm{m}}}(\alpha_i) = \begin{bmatrix} 2905.7 & 715.4 & -303.9 \\ 715.4 & 27795.7 & 0.0 \\ -303.9 & 0.0 & 30475.0 \end{bmatrix} \times 10^{-7} \Bigg|_{\alpha_i = 0} \quad \mathrm{kg.m}^2, \ \in \mathcal{F}^b \tag{2.30h}$$

Then, reiterating Eq:2.28, the instantaneous inertia of the entire body in motion is calculated as the contribution of each connected sub-body, depending on the actuator matrix $u \in \mathbb{U}$.

$$\underset{\vec{\mathbf{O}}_b}{J_b(u)} = \underset{\vec{\mathbf{O}}_b}{J'_{\mathrm{y}}} + \sum_{i=1}^{4} \underset{\vec{\mathbf{O}}_b}{J_{\mathrm{n}}(u \cdot i)} + \sum_{i=1}^{4} \underset{\vec{\mathbf{O}}_b}{J_{\mathrm{m}}(u \cdot i)} \qquad u \in \mathbb{U} \tag{2.31a}$$

The net mass for the entire multibody system is $m_b = 1574.7$ g. For reference and using Eq:2.31a; the inertial matrix for the assembly when actuators are at rest conditions, $u = \vec{0}$, about the origin $\vec{\mathbf{O}}_b$ is:

$$J_b(\vec{0}) = \begin{bmatrix} 317448.2 & 0.4 & -14.5 \\ 0.4 & 317570.7 & 6.5 \\ -14.5 & 6.5 & 628257.5 \end{bmatrix} \times 10^{-7} \Big|_{u=\vec{0}} \qquad \mathrm{kg.m}^2, \;\; \in \mathcal{F}^b \tag{2.31b}$$

The maximum variation of the body's net inertia is found from the maximum determinant of the inertial matrix in Eq:2.31a for some actuator state $max(det|J_b(u_\Lambda)|)$, $u_\Lambda \in \mathbb{U}$. A maximum $J_b(u_\Lambda)$, with a determinant $det|J_b(u_\Lambda)| = 1017.93 \times 10^{-7}$, is:

$$J_b(u_\Lambda) = \begin{bmatrix} 384695.4 & 0.4 & -14.5 \\ 0.4 & 384717.9 & 6.5 \\ -14.5 & 6.5 & 687970.7 \end{bmatrix} \times 10^{-7} \Big|_{u_\Lambda} \qquad \mathrm{kg.m}^2, \;\; \in \mathcal{F}^b \tag{2.32a}$$

With an actuator matrix, independent of propeller speeds $\Omega_{1 \to 4}$, as follows:

$$u_\Lambda = \begin{bmatrix} \Omega_1, & \lambda_1 = 178^\circ, & \alpha_1 = 260^\circ \ldots \\ \Omega_2, & \lambda_2 = 178^\circ, & \alpha_2 = 260^\circ \ldots \\ \Omega_3, & \lambda_3 = 178^\circ, & \alpha_3 = 0^\circ \ldots \\ \Omega_4, & \lambda_4 = 0^\circ, & \alpha_4 = 0^\circ \end{bmatrix} \tag{2.32b}$$

Conversely, the minimum net inertia for the body is from the smallest determinant of Eq:2.31a, for the actuator state $min(det|J_b(u_\Upsilon)|)$, $u_\Upsilon \in \mathbb{U}$. A minimum $J_b(u_\Upsilon)$, with a determinant $det|J_b(u_\Upsilon)| = 633.48 \times 10^{-7}$, is:

$$J_b(u_\Upsilon) = \begin{bmatrix} 317469.0 & 0.4 & -1219.0 \\ 0.4 & 317591.5 & 1195.3 \\ -1219.0 & 1195.3 & 628298.1 \end{bmatrix} \times 10^{-7} \Big|_{u_\Upsilon} \qquad \mathrm{kg.m}^2, \;\; \in \mathcal{F}^b \tag{2.33a}$$

When an actuator matrix for that minimum inertia is:

$$u_\Upsilon = \begin{bmatrix} \Omega_1, & \lambda_1 = 178^\circ, & \alpha_1 = 0^\circ \ldots \\ \Omega_2, & \lambda_2 = 0^\circ, & \alpha_2 = 260^\circ \ldots \\ \Omega_3, & \lambda_3 = 0^\circ, & \alpha_3 = 0^\circ \ldots \\ \Omega_4, & \lambda_4 = 0^\circ, & \alpha_4 = 0^\circ \end{bmatrix} \tag{2.33b}$$

The inclusion of Eq:2.32 and Eq:2.33 are used for maximum and minimum Eigenvalues of the body's inertial matrix at a later stage in the control derivation, Sec:4.6. It is interesting to note that both extremes of $J_b(u)$ are still symmetrical, and *roughly* diagonal. Actuator positions hardly affect the skew products of inertia in $J_b(u)$ but can vary the diagonal moments of inertia by almost 20% of their principle value.

Unless otherwise specified; any inertia $J_b(u)$ indicates an instantaneous calculated solution to Eq:2.31a given a particular $u(t) \in \mathbb{U}$. The purpose of the derivations for rotated centers of mass in Eq:2.29 and Eq:2.30 is twofold; highlighting both the inertial contributions *and* the variable center of mass for each sub-body. Seeing that the origin of motion $\vec{\mathbf{O}}_b$ in the body frame $\mathcal{F}^b$ and the body's effective center of mass $C.M_{\mathrm{b}}$ are not coincidental, it is important to quantify the net center of mass's variation 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:

$$C.M = \frac{\sum_{i=1}^{n} m_i.\vec{X}_i}{\sum_{i=1}^{n} m_i} \tag{2.34a}$$

Using $C.M_{\mathrm{n}}'''(\lambda_i, \alpha_i)$ and $C.M_{\mathrm{m}}''(\alpha_i)$ as rotated centers of mass defined in Eq:2.29d and Eq:2.30d respectively and $C.M_{\mathrm{y}}$ for the body structure, the vehicle has a variable center of mass $C.M_{\mathrm{b}}(u)$:

$$C.M_{\mathrm{b}}(u) = \frac{m_{\mathrm{y}}C.M_{\mathrm{y}} + \sum_{i=1}^{4} m_{\mathrm{n}}C.M_{\mathrm{n}}'''(u \cdot i) + \sum_{i=1}^{4} m_{\mathrm{m}}C.M_{\mathrm{m}}''(u \cdot i)}{m_{\mathrm{b}}} \tag{2.34b}$$

So, for reference, the net center of gravity for the entire multibody assembly, when all actuators are at their zero positions is: $C.M_b(\vec{0}) = \begin{bmatrix} 0 & 0 & -4.94 \end{bmatrix}^T$ mm. Using a gravity force vector $\vec{G}_b$ in the body frame as a result of gravitational acceleration $g = -9.81$ m.s^{-2}, acting on the vehicle's center of mass:

$$\vec{G}_b = R_I(\vec{\eta})^b \vec{G}_I \qquad \in \mathcal{F}^b \tag{2.35a}$$

$$= R_I^b(\vec{\eta}) \begin{bmatrix} 0 & 0 & -9.81(m_{\mathrm{b}}) \end{bmatrix}^T \quad \mathrm{N} \tag{2.35b}$$

Because Eq:2.35 acts through the body's center of gravity $C.M_{\mathrm{b}}(u)$, not its center of motion $\vec{\mathbf{O}}_b$, there exists a gravitational torque from the varying center of gravity. The resultant gravitational torque about the origin $\vec{\mathbf{O}}_b$ in the body frame $\mathcal{F}^b$ from that eccentric mass center for the vehicle is:

$$\Delta C.G = \vec{\mathbf{O}}_b - C.M_{\mathrm{b}}(u) \tag{2.35c}$$

$$\vec{\tau}_g = \Delta C.G \times m_b \vec{G}_b \qquad \in \mathcal{F}^b \tag{2.35d}$$

The prototype which was constructed is shown in Fig:2.18. The above mass centers and inertias were calculated from physical values measured on assembled components of the prototype. The listed values includes measurements of fasteners and electronics.

Figure 2.18: Final constructed prototype

Uncertainty with inertial measurements, proven to be destabilizing and detrimental to control efforts in [77, 142], can indeed be incorporated into state dependent plant uncertainty compenstaion like in [11]. Controllers with strong disturbance and uncertainty rejection, like a well designed H_∞ controller, would be ideally suited to controlling an attitude plant without having to explicitly specify all of the above inertias.

It is, however, worth the mathematical deliberation to detail each inertial equation given that Lagrange dynamics are later applied to determine the servo actuator dynamic responses (Sec:3.4). Such equations of motion will later need explicit terms defined for instantaneous transformed inertias…

2.4 Electronics

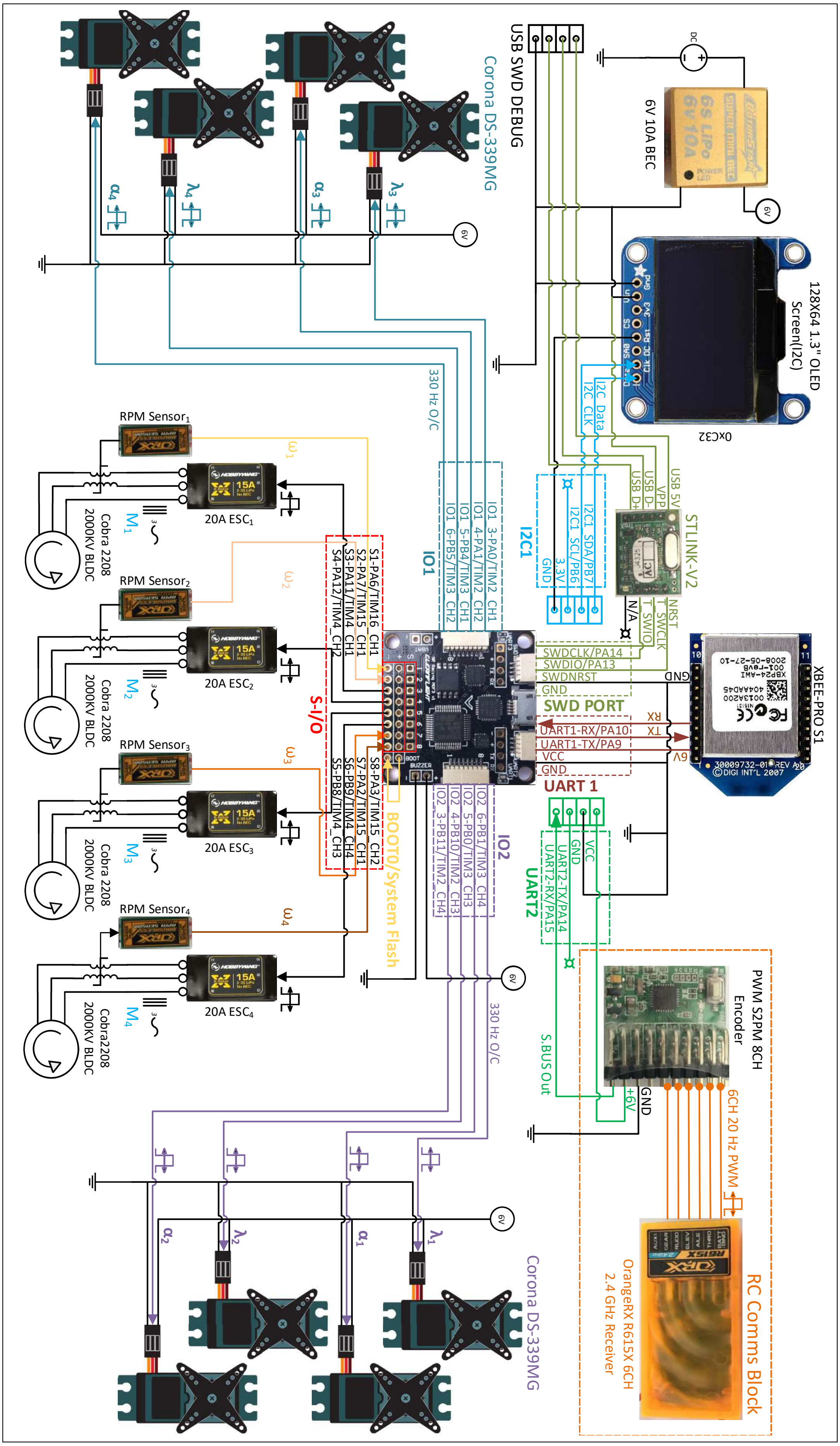
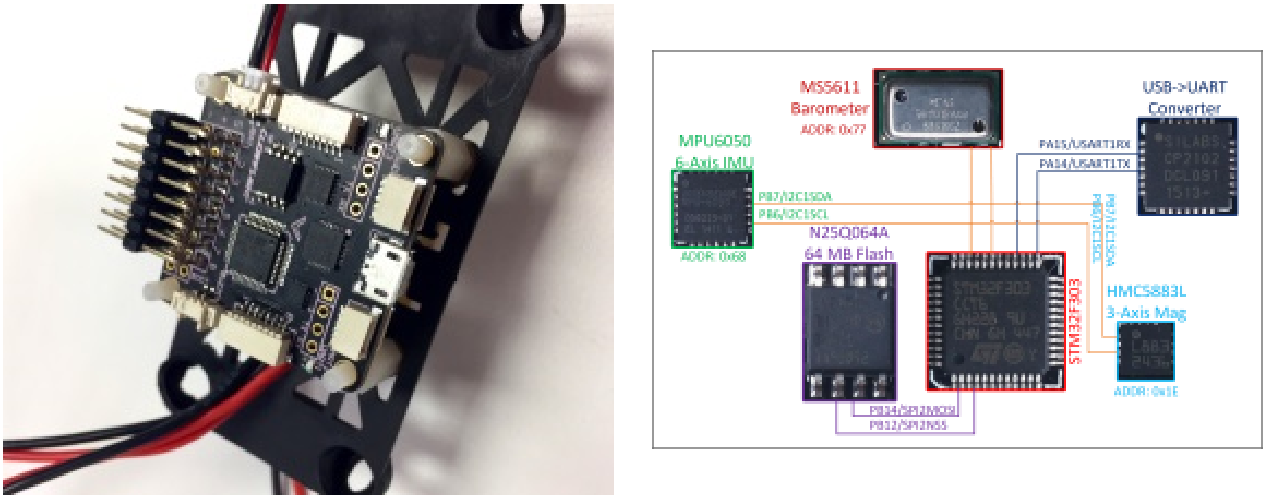


Figure 2.19: Hardware schematic diagram

An abstracted hardware diagram for the proposed (electronic) system layout is shown in Fig:2.19. It is an illustration for the connection of different electronic peripherals 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 intended for the flight controller, their purpose and a description of how they are interfaced. No control loops or code structures are discussed here.



(a) SPRacing F3 deluxe flight controller

(b) F3 Deluxe on-board connections

Figure 2.20: SPRacing F3 deluxe layout

The embedded system is constructed around an ARM STM32F303 [129] based microcontroller. The micro-processor board is a commercial flight control board, specifically an SPRacing F3 Deluxe [33]. CleanFlight or BetaFlight opensource software (from [32] and [15] respectively) are typically used for this SPRacing F3 board; but 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:2.20b.

The flight-controller has the following onboard peripherals; an I2C MPU-6050 6-axis gyroscope and accelerometer [63] with an I2C connected HMC5883 magnetometer compass [39]; an I2C MS5611 barometer [127] and finally 64 Mb of SPI flash memory. Consideration of sensor fusion effects of the above state-estimators is discussed subsequently in Sec:6.9. The caveats of Kalman filtering and discretized effects on the simulation loop are similarly discussed in that particular section.

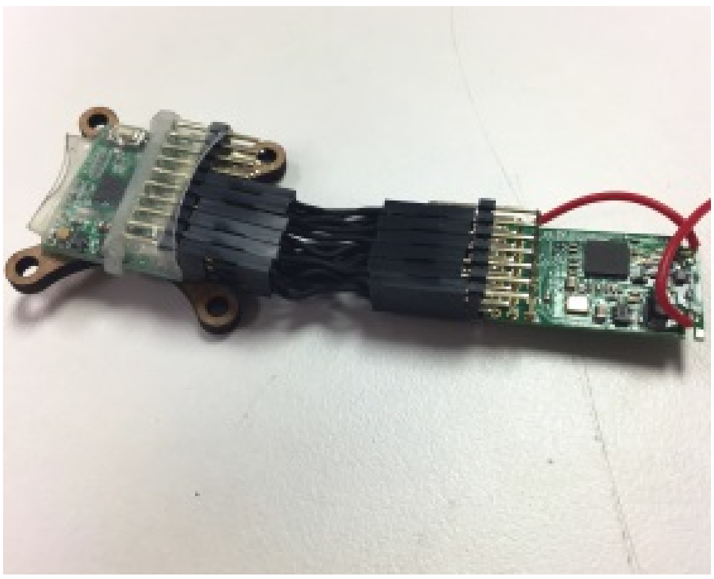


Figure 2.21: SBUS converter & 6CH receiver

Two separate wireless communication loops are to be used. First; the system relays full state information for a complete 6-DOF X-Y-Z position and $\phi - \theta - \psi$ orientation autopilot system. Sent from an independent ground control station (*GCS*) using 2.4 GHz XBEE S1 module(s) [65] which is connected to the flight controller via USART. Full state-estimation, using a multi-camera system ([111]), 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 six channel 2.4 GHz radio frequency module. The secondary system allows for phsyical control without the need of a trajectory generation loop. The six CH received signals, otherwise permeated as six individual 20 kHz PWM signals via an OrangeRx R615x receiver [101], are encoded into a single proprietary S.BUS data stream (Fig:2.21).

The need for a serial bus (S.BUS) encoder, specifically using [57], comes about as a consequence of the introduction of the eight additional servos. As a result, there are no longer six 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 six CH commands can be processed with a single RX channel by the microcontroller. The encoder implements a USART derivative communications standard called S.BUS. Shown in Fig:2.22 the S.BUS data, captured with a logic analyzer [121], 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:

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

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

Figure 2.22: S.BUS data stream

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

$$y_n = (1 - \frac{1}{N})y_{n-1} + \frac{1}{N}x_n \tag{2.36}$$

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

Similarly all the measured RPM signals measured by the OrangeRx RPM speed sensors are filtered over five 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 to be performed separately on the Ground Control Station computer.

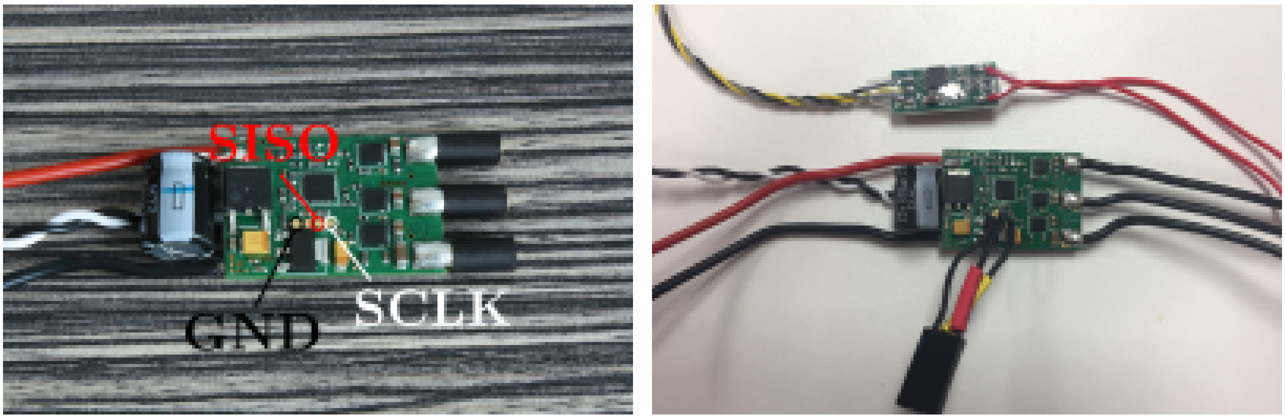
Each of the eight digital servo actuators are controlled individually from 330 Hz center aligned PWM timer output compare channels (TIM2:CH1→CH4 and TIM3:CH1→CH4). Output pulses range from $1-2$ ms to linearly control the rotational position. The servo's exact range and transfer function(s) is empirically determined next in Sec:2.4.1. The four 20 A brushless DC electronic speed controllers (*ESC*s) 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 twelve PWM output compare signals drawn from the flight controller, eight for the servos and four for the ESCs. The servos are powered by a regulated 6 V DC 10 A power supply [56] whilst the ESCs switch unregulated 14.1 V DC supplied from an external power tether. The DC supply could be drawn from a battery bank but that would adversely affect the weight of an already heavy platform.

There is no integrated feedback for instantaneous RPM values available from the ESCs. Dedicated OrangeRX BLDC RPM sensors, [55], 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, when used with an ESC, 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 [85] to be proportional to the rotational velocity:

$$F_{rps} = \frac{2 \times F_{poles}}{\text{No. of rotor poles}} \quad [\text{Hz}] \tag{2.37}$$

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:2.24. Knowing exact RPM rates means the subsequent thrust and aerodynamic torques for the control plant inputs can be calculated with greater certainty.

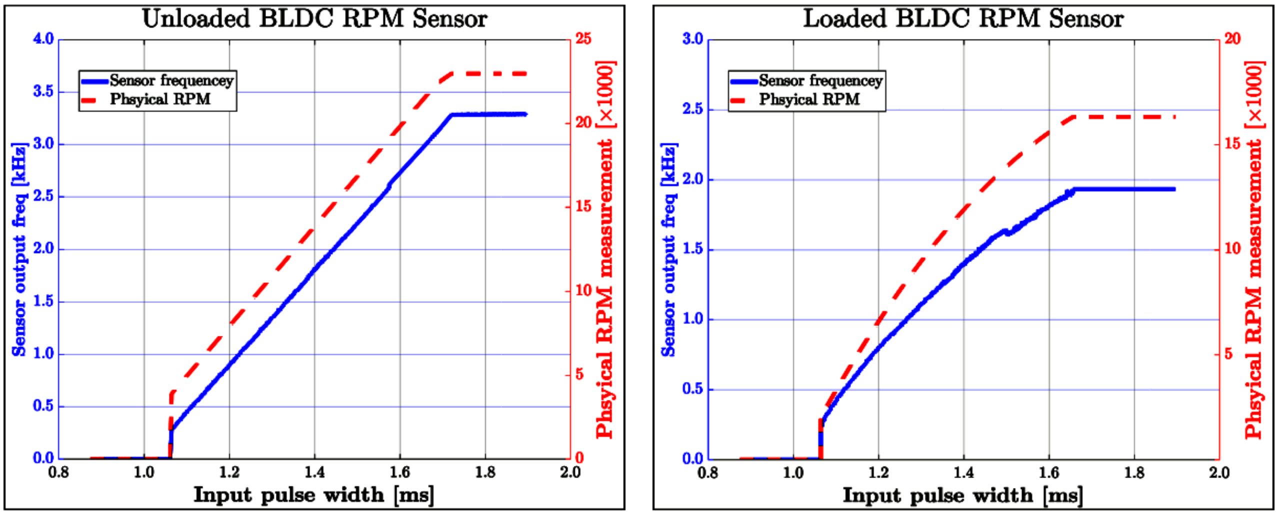


(a) XRotor 20A ESC connection guide [54]

(b) LDPower 20A ESC with RPM sensor

Figure 2.23: BLDC electronic speed controllers

The ESCs, although LDPower 20A devices, are re-flashed with BLHeli firmware [16]. The LDPower ESCs (Fig:2.23b) match Hobbywing Xrotor 20A ones (Fig:2.23a) which both use SiLabs F396 micro-controllers; the same firmware can be flashed onto both MCUs. Custom BLHeli software provides greater refinement over configurations like the deflection range of inputs, but default values were used. The plot in Fig:2.24a shows the rotation per second, or otherwise frequency in Hz, speed curve for an unloaded motor; similarly in Fig:2.24b shows the speed curve when loaded for a 6×4.5 prop.



(a) RPM sensor plot - no load

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

Figure 2.24: RPM sensor calibration plots

The loaded speed plot for a BLDC motor with an attached prop in Fig:2.24b is slightly quadratic; that response is due to second order aerodynamic drag, quadratic with respect to the propeller's rotational speed (expanded on in Sec:3.2.1). Moreover, when the motor is torque loaded by the propeller, the ESC current limits rotational speeds at just over 16×10^3 RPM.

Timer 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 microcontroller is programmed through a dedicated debugging device. The ST-Link V2 [128] is the current proprietary device which, itself, is a specially programmed STM32F10 chip. The chip connects to the dedicated **S**erial **W**ire **D**ebugging ports of the target STM (*SWD-CLK, SWD-IO* & *SWD-NRST*) and is interfaced via regular USBD+ and USBD- data lines.

2.4.1 Actuator Transfer Functions

Servo Transfer Functions

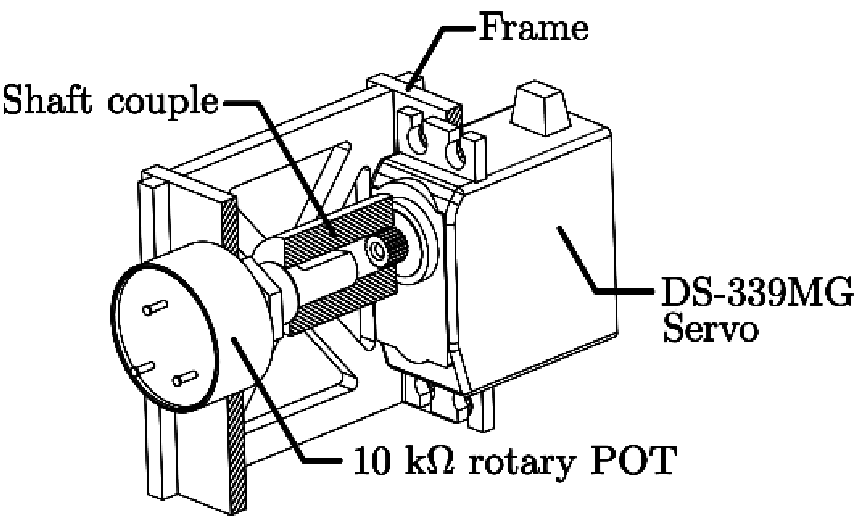
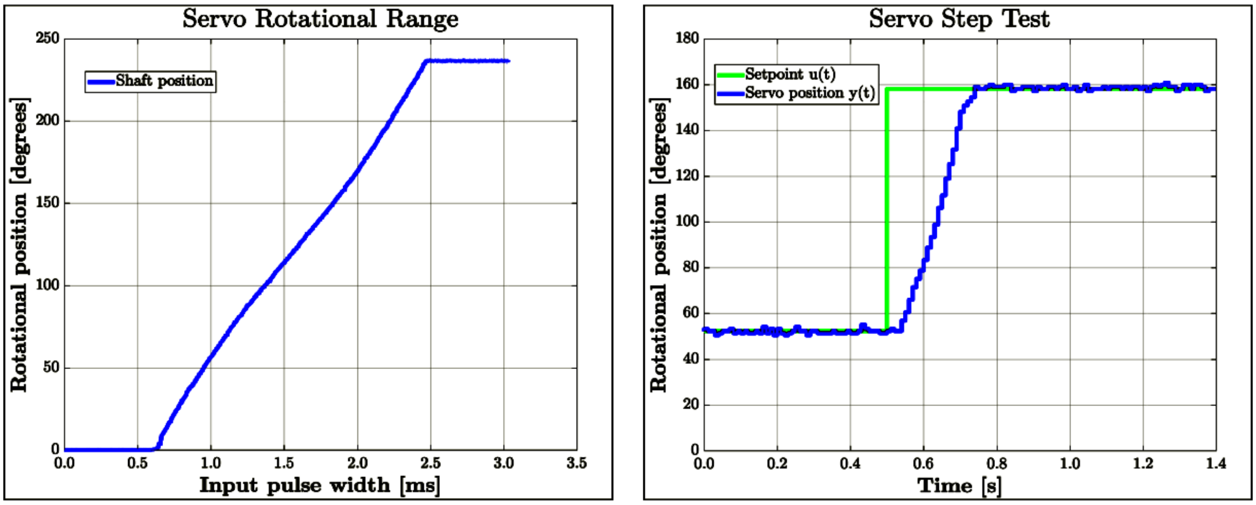


Figure 2.25: Servo transfer function test rig

The range and step transfer functions for an unloaded servo was evaluated with the test rig, illustrated in Fig:2.25. The servo's output shaft was mechanically coupled to a rotatory potentiometer which was sampled to measure the shaft's rotational position. Full scale deflection for the digital servos used are in fact greater than their quoted 180° range, each having an input range of around 230° plotted in Fig:2.26a. The prototype control loop commands each servo position in open loop; the major loop controller gains designed later in Sec:6.2 are expected to account for such minor loop actuator dynamics. However, the simulation must first accurately represent the servo's transfer characteristics for such an assumption to hold true.

Considering the servo's hard limit of 180° was a design imposed constraint, one point of contention is the effect such a restriction has on the feasible operating trajectories. The control algorithms derived in Ch:4 are first tested with an ideal, continuous rotation servo limited by the rate and transfer characteristics. Following that, the servo's rotational limitations are imposed on the system and the constraints to feasibly achievable trajectories are investigated in Sec:6.8.



(a) DS339-MG full Range **(b)** DS339-MG Step Response

Figure 2.26: Unloaded servo transfer characteristics

For the servo whose particular rotational range and step response is shown in Fig:2.26; the relationship between the input pulse-width x in m.s and the rotational output position y in degrees is governed by the hybrid system:

$$y(x) = \begin{cases} 0 & x < 0.65 \text{ ms} \\ 129.12x - 82.64 & 0.64 \text{ ms} \leq x \leq 2.46 \text{ ms} \\ 230 & x > 2.46 \text{ ms} \end{cases} \tag{2.38}$$

In practice Eq:2.38 is altered such that a 0° offset is taken at around 50% input, making its operational range ±90°. Each servo is mechanically rate limited to $60°/0.15\,s$ or 400 degrees per second with a dead time of $t_d \approx 1.2$ ms and a negligible mechanical deadband of 4 μs. Each servo has an approximate *critically damped* second order transfer function, determined from Fig:2.26b:

$$G(s)_{servo} = e^{-t_d s} \frac{w_n^2}{s^2 + 2\zeta w_n s + w_n^2} \tag{2.39a}$$

$$= \frac{e^{-0.012s}(14.869)^2}{s^2 + 2(1)(14.869)s + (14.869)^2} \tag{2.39b}$$

With input saturation limits for the PWM input magnitude $|U(s)|$ in ms:

$$Y(s)_{servo} = \begin{cases} 0° & |U(s)| < 0.65 \\ G(s) & 0.65 \leq |U(s)| \leq 2.46 \\ 230° & |U(s)| > 2.46 \end{cases} \tag{2.39c}$$

The net second order, critically damped transfer block for a servo is shown in Fig:2.27, including saturating nonlinearities but neglecting the afore mentioned mechanical deadband...

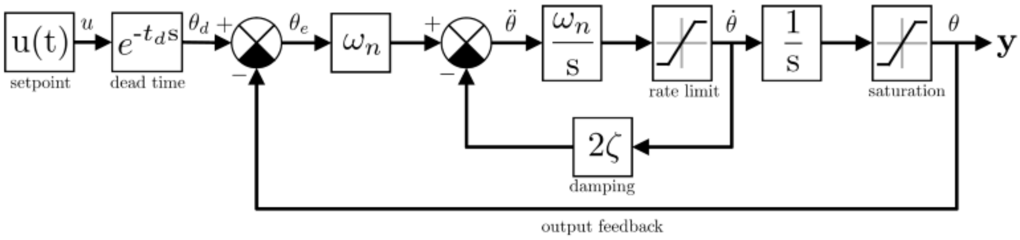
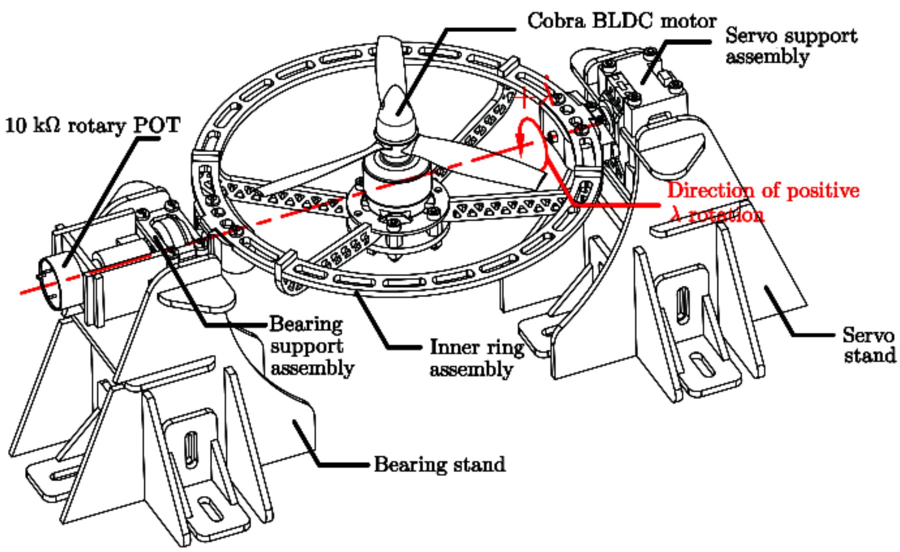
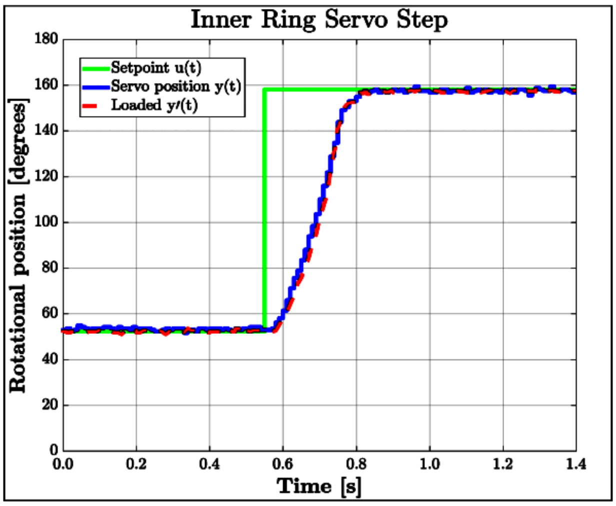


Figure 2.27: Servo block diagram

The plot in Fig:2.26b shows the step response, at the *shaft output*, of an unloaded servo. The servo's transfer characteristics when rotating the inner ring assembly (illustrated in Fig:2.12) are determined from the test rig in Fig:2.28a. Fig:2.28b shows the inner ring servo's step response $\mathbf{y(t)}$, which is unchanged from Eq:2.39. It then follows that for the inner ring's transfer function $\therefore G(s)_{inner} = G(s)_{servo}$. Even when actuating a loaded inner ring assembly with a propeller rotational velocity of $\Omega_i = 6000$ RPM, plotted $\mathbf{y'(t)}$; the transfer characteristics are the same in spite of a further increased load on the assembly due to the induced gyroscopic response, Eq:2.19.



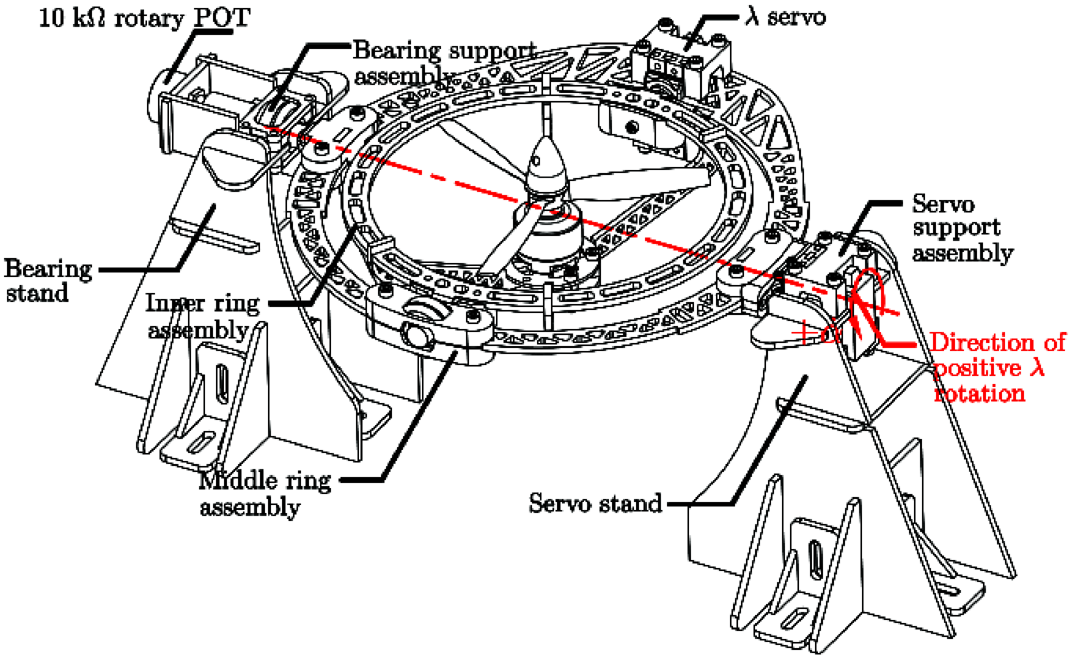
(a) Inner ring servo rig



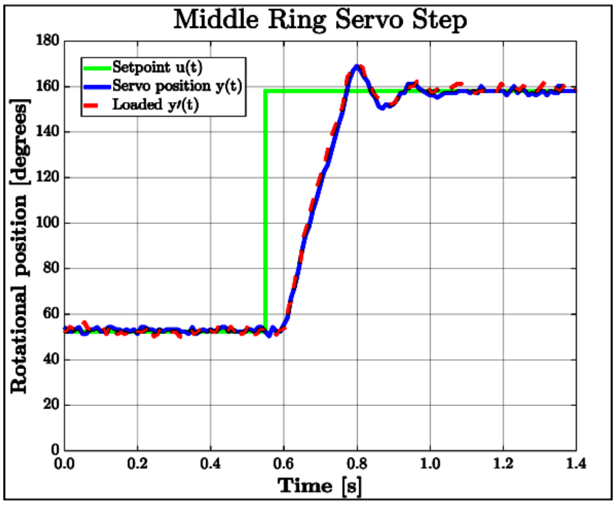
(b) Servo response plot

Figure 2.28: Inner ring servo characteristics

Fig:2.29b plots the step response for the servo actuating the middle ring assembly. Whilst its transients remain the same, oscillations are introduced at the settling point which demonstrates a second order under-damped plant. Those oscillations are as a result of the larger rotational inertia (Eq:2.24) and flexure within the frame structure. It is important to specify that the oscillations are not at the servo's output shaft; the rotational position was measured with respect to the bearing supported shaft, coaxial to the servos (Fig:2.29a). A separate, under-damped transfer fucnction is used for the middle ring's response, the rotational position α_i of the frame is to be used for thrust vectoring calculations in Eq:2.17. Those harmonics are still present under load, plotted in $\mathbf{y'(t)}$, despite the frame being tensioned by the thrust.



(a) Middle ring servo test rig



(b) Servo response plot

Figure 2.29: Middle ring servo characteristics

The mechanical structure could indeed be strengthened to reduce the oscillations present in Fig:2.29a. Strengthening the frame would, however, increase the mass of an already weight constrained system. Instead the under-damped transfer function is incorporated into the plant, that transfer function is:

$$G(s)_{middle} = \frac{e^{-0.012s}(12.591)^2}{s^2 + 2(0.454)(12.591)s + (12.591)^2} \tag{2.40}$$

BLDC Transfer Functions

Each Cobra 2208 BLDC motor, when loaded with a 6×4.5 propeller has a quadratic speed curve (plotted in Fig:2.31a). This is as a result of the propeller's opposing aerodynamic drag, *appromixately* proportional to the square of the propellers angular velocity. Propeller aerodynamics are expanded on further in Sec:3.2.1.

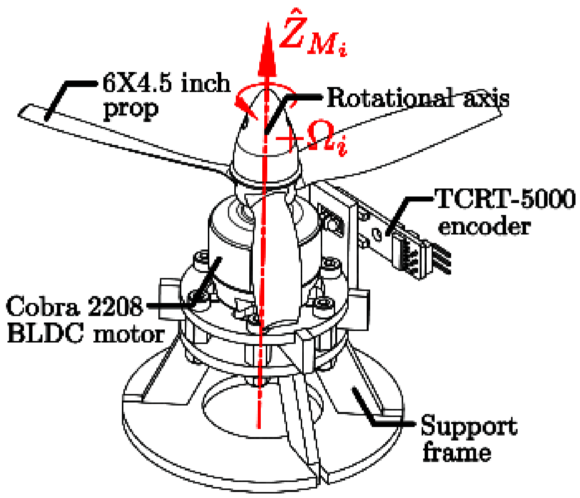
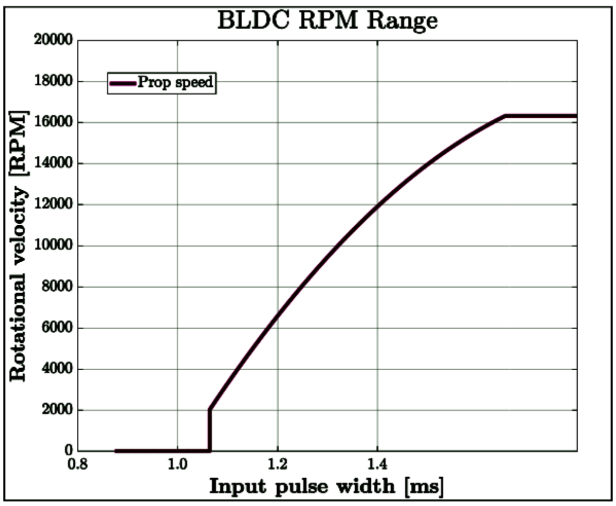


Figure 2.30: BLDC rpm speed calibration and transfer function rig

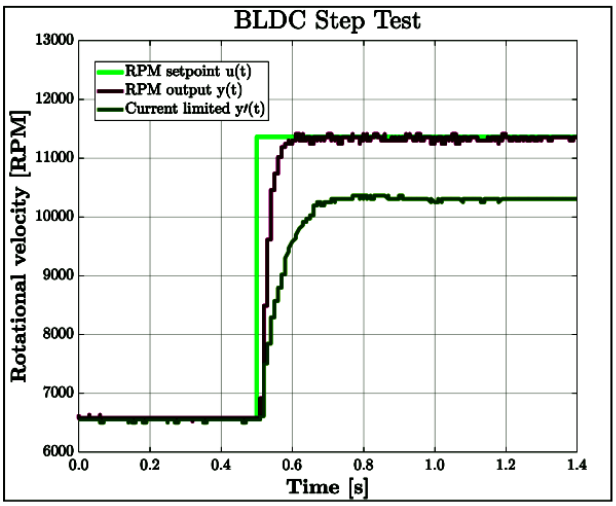
Using the BLHeli interface the input range for the motor's speed controllers can be adjusted, but for the purposes of this project were left unchanged. That relationship between input pulse-widths x in ms applied to the ESC and output sensor signal y in RPM is given by the hybrid state equations for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

$$y(x) = \begin{cases} 0 & x < 1.065 \text{ ms} \\ -20593x^2 + 80187x - 60004 & 1.065 \text{ ms} \leq x \leq 1.655 \text{ ms} \\ 16300 & x > 1.655 \text{ ms} \end{cases} \tag{2.41}$$

The upper limit in Eq:2.41 and the motor's step response are both governed by the ESC's maximum current limit; in this case 20 A. Artificially imposing 10 A current limit, a potential consequence of using lower power ESCs, is plotted $c(t)$ in Fig:2.31b. The current limit significantly restricts the motor's transient and steady-state performance.



(a) BLDC input RPM range



(b) Cobra BLDC step response

Figure 2.31: BLDC motor characteristics

The motor's step response, $\mathbf{y(t)}$, has a negligible dead time and second order dynamics, with a transient time constant far faster than the servo's plant. The motor's transfer function for speed in RPM is:

$$G_{BLDC}(s) = \frac{1}{\left(1 + 1.7583s \times 10^{-3}\right)\left(1 + 1.7494s \times 10^{-3}\right)} \tag{2.42a}$$

And saturating input limits on $|U(s)|$ for the PWM input magnitude in ms:

$$Y_{BLDC}(s) = \begin{cases} 0 & |U(s)| < 1.065 \\ G(s) & 1.065 \leq |U(s)| \leq 1.655 \\ 16300 & |U(s)| > 1.655 \end{cases} \tag{2.42b}$$

Combined Actuator Transfer

The net transfer characteristics for a complete motor module then combines Eq:2.39 for the inner ring step in λ_i, Eq:2.40 for changes in the middle ring α_i and finally Eq:2.42 for changes in the propeller's rotational speed Ω_i. A single module's transfer function is then bundled in the transfer block $C(s)$:

$$u_i \triangleq \begin{bmatrix} \Omega_i(s) \\ \lambda_i(s) \\ \alpha_i(s) \end{bmatrix} = \begin{bmatrix} B(s)_{BLDC} \\ N(s)_{inner} \\ M(s)_{middle} \end{bmatrix} = C(s) \tag{2.43a}$$

Furthermore, the actuator space for the i^{th} motor module $u_i \in \mathbb{U}$ is limited by input saturation conditions, where the inputs are *not* in pulse width magnitudes:

$$\mathbb{U}_i \triangleq \begin{bmatrix} 0 & : & 16300 \\ -90° & : & +90° \\ -90° & : & +90° \end{bmatrix} \tag{2.43b}$$

Where $\mathbb{U}_i$ is then extended to $\in \mathbb{R}^{12}$ for the entire actuator set. Later, instantaneous actuator positions are used to calculate response dynamics, in which case a commanded $u_c \in \mathbb{U}$ is applied, subject to the transfer functions and saturations of Eq:2.43. In the case of control design and feedback compensation, *error free* actuator estimates $\hat{u}$ are used, which represent sampled actuator states. A commanded actuator position u_c is the instantaneously set actuator value as per some control function, in practice there's an actuator transfer error:

$$u_e = u_c - u \approx u_c - \hat{u} \tag{2.44}$$

Which could lead to plant errors in inertia calculations dependent on those actuator positions, Sec:2.3. Estimates for instantaneous inertial positions $\hat{u}$ are further used for dynamic calculations, not the commanded actuator positions u_c. Moreover, the actuation error as a result of the minor loop transfer functions produce a deviation in expected actuation effort those being force and torque inputs. A robust controller with a well designed gain ought to account for those deviations and retain stability irrespective. . .

Chapter 3

Kinematics & Dynamics

The following generally applicable rigid body dynamics are first developed with respect to generalized net forces and torques acting on a rigid vehicle. Following that, dynamics are extended to the nonlinear multibody case wherein constrained relative rotational actuation between interconnected bodies is incorporated; representing the actuator action which the prototype can undergo. Propeller aerodynamic effects are subsequently included into the actuation input model. Finally a consolidated quaternion based model is presented which is used for the controller development next 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 translational movement. Often a Lagrangian approach for combined angular and translational movements is used to derive the differential equations of motion for each degree of freedom, [112,133]. The Lagrangian principle ensures that (translational and rotational) energies are conserved throughout the system's state progression. When combined with Euler-Rotation equations, the Euler-Lagrangian formulation from [135] fully defines the aerospace 6-DOF equations of motion.

Lagrangian formulation is regarded as especially useful in non-Cartesian (*spherical etc...*) coordinate frames and with multibody systems. With that being said, Cartesian coordinates were already defined in Sec:2.2.2 for the plant. Alternatively; relative coordinates could be used for implicit-Euler based dynamics as in [98]. Rigid body dynamics in Cartesian coordinates do lend themselves to Newtonian mechanics. Both Newton-Euler and Euler-Lagrange formulations produce the same resultant differential equations of motion, but follow conceptually different derivations. The Lagrangian operator $\mathcal{L}$ is a scalar term defined as the difference between a trajectory's kinetic and potential energies, T and U respectively. Considering some generalized path trajectory $\vec{\mathbf{r}}(t)$ for a body, with both position $\vec{\xi}$ and attitude $\vec{H}$ states:

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

Coordinates in Eq:3.1 are *generalized* and taken with respect to some hypothetical shared frame $\mathcal{F}^\Lambda$. The generalized coordinates are later refined to Cartesian body coordinates with respect to the inertial frame. The Lagrangian is the difference of the trajectory's kinetic and potential energies, by definition:

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

Where the trajectory's kinetic and potential energy functions are T and U respectively. Introducing a rigid body's general (translational and rotational) kinetic and potential energies, both defined with respect to that shared reference frame $\mathcal{F}^\Lambda$...

Noting first that there is no attitude contribution for stored potential energy, so $U(\vec{\mathbf{r}}, \dot{\vec{\mathbf{r}}})$ consists entirely of gravitational potential energy. Gravitational acceleration in the inertial frame $\in \mathcal{F}^I$ is:

$$\vec{G}_I = \begin{bmatrix} 0 & 0 & -9.81 \end{bmatrix}^T \quad \text{m.s}^{-2}, \ \in \mathcal{F}^I \tag{3.2b}$$

Where $\vec{G}_I$ acts in the negative $\hat{Z}_I$, downward, direction. Substituting translational kinetic and potential energies into the Lagrangian yields the following scalar term:

$$\mathcal{L}(\vec{\mathbf{r}}, \dot{\vec{\mathbf{r}}}, t) = \frac{1}{2}\dot{\vec{\xi}}^{\,T}(m_b)\dot{\vec{\xi}} + \frac{1}{2}\dot{\vec{H}}^{\,T}(J_b)\dot{\vec{H}} - m_b\vec{G}_\Lambda(h \cdot \hat{Z}_I) \tag{3.2c}$$

The vehicle's mass is m_b and its generalized inertia matrix is similarly J_b, 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, $\vec{\mathbf{V}}$, acting on the system in frame Λ. In the rigid body motion case those *generalized forces* are net forces $\vec{F}_\mu$ and net torques $\vec{\tau}_\mu$ in the shared frame $\in \mathcal{F}^\Lambda$.

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\vec{\mathbf{r}}}}\right) - \frac{\partial \mathcal{L}}{\partial \vec{\mathbf{r}}} = \vec{\mathbf{V}} = \begin{bmatrix} \vec{F}_\mu \\ \vec{\tau}_\mu \end{bmatrix} \qquad \in \mathcal{F}^\Lambda \tag{3.3}$$

Evaluating symbolic partial derivatives of Eq:3.2c with respect to the path coordinates $\vec{\mathbf{r}}(t)$ and path rates $\dot{\vec{\mathbf{r}}}(t)$ respectively produces the two following equations:

$$\frac{\partial \mathcal{L}}{\partial \vec{\mathbf{r}}} = \begin{bmatrix} m_b\vec{G}_\Lambda \\ 0 \end{bmatrix} \qquad \in \mathcal{F}^\Lambda \tag{3.4a}$$

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\vec{\mathbf{r}}}}\right) = \begin{bmatrix} \frac{d}{dt}m_b\dot{\vec{\xi}} & \frac{d}{dt}J_b\dot{\vec{H}} \end{bmatrix}^T \qquad \in \mathcal{F}^\Lambda \tag{3.4b}$$

Where $\vec{G}_\Lambda$ is the gravitation force transformed to the common frame $\mathcal{F}^\Lambda$ which $\mathcal{L}(\vec{\mathbf{r}}, \dot{\vec{\mathbf{r}}})$ is defined with respect to. The body mass m_b and inertia J_b could potentially have some non-zero time derivative, but for now are regarded as constants. Time varying inertias are later defined in Sec:2.3 and introduced to the dynamics subsequently in Sec:3.4.1. Here only the general rigid body case is considered...

Any vector in some non-Newtonian rotating reference frame $\mathcal{F}^a$ has a time derivative, relative to another frame $\mathcal{F}^b$ with an angular velocity $\vec{\omega}_{a/b}$, as per the Reynolds Transportation Theorem [130]:

$$\frac{d\vec{f}_b}{dt_a} = \frac{d\vec{f}_b}{dt_b} + \vec{\omega}_{a/b} \times \vec{f}_b \qquad \in \mathcal{F}^b \tag{3.5}$$

Applying Eq:3.5 to those partial derivatives in Eq:3.4b and further defining the generalized coordinates $\begin{bmatrix}\vec{\xi}, & \vec{H}\end{bmatrix}^T$ as 6-DOF Cartesian body coordinates with respect to the inertial frame $\mathcal{F}^I$ or the body frame $\mathcal{F}^b$, described in Sec:2.2.

Reiterating that the angular orientations $\vec{H}$ 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 an attitude in a shared frame $\vec{\eta}_b$ from Eq:2.12e, where $\vec{\omega}_b = \dot{\vec{\eta}}_b$ and $\vec{\eta}_b \in \mathcal{F}^b$, the trajectory's definition for $\vec{\mathbf{r}}$ is refined:

$$\vec{\mathbf{r}} = \begin{bmatrix} \vec{\xi} & \vec{H} \end{bmatrix}^T \triangleq \begin{bmatrix} \vec{\mathcal{E}}_b \\ \vec{\eta}_b \end{bmatrix} \qquad \in \mathcal{F}^b \tag{3.6a}$$

Note that the position $\vec{\mathcal{E}}_b$ in Eq:3.6a is the position in the *body frame*, unlike $\vec{\mathcal{E}}_I \in \mathcal{F}^I$ from Eq:2.2. The path rate $\dot{\vec{\mathbf{r}}}(t)$ is the defined as:

$$\therefore \dot{\vec{\mathbf{r}}} = \begin{bmatrix} \dot{\vec{\xi}} & \dot{\vec{H}} \end{bmatrix}^T \triangleq \frac{d}{dt}\begin{bmatrix} \vec{\mathcal{E}}_b \\ \vec{\eta}_b \end{bmatrix} = \begin{bmatrix} \vec{v}_b \\ \vec{\omega}_b \end{bmatrix} \qquad \in \mathcal{F}^b \tag{3.6b}$$

Substituting those changed path coordinates from Eq:3.6 into the Lagrangian Eq:3.2c yields a familiar Lagrangian scalar for a vehicle's energies for $\mathcal{F}^b$ relative to $\mathcal{F}^I$:

$$\mathcal{L} = \frac{1}{2}\vec{v}_b^{\,T}(m_b)\vec{v}_b + \frac{1}{2}\vec{\omega}_b^{\,T}(J_b)\vec{\omega}_b - m_b\vec{G}_b z_I \tag{3.7}$$

Where $\vec{G}_b$ is the gravitational force vector from Eq:3.2b transformed to $\mathcal{F}^b$ and z_I is the vertical height of the vehicle *in the inertial frame*. The time derivative of the substituted path coordinates in the partial derivative Eq:3.4b is then:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{r}}}\right) = \begin{bmatrix} m_b \frac{d}{dt}\vec{v}_b & J_b \frac{d}{dt}\vec{\omega}_b \end{bmatrix}^T \tag{3.8a}$$

With respective time derivatives of body frame vectors relative to the inertial frame, using the Reynolds transportation theorem:

$$\rightarrow m_b \frac{d}{dt}\vec{v}_b = m_b\dot{\vec{v}}_b + \vec{\omega}_{b/I} \times m_b\vec{v}_b \qquad \in \mathcal{F}^b \tag{3.8b}$$

$$\rightarrow J_b \frac{d}{dt}\vec{\omega}_b = J_b\dot{\vec{\omega}}_b + \vec{\omega}_{b/I} \times J_b\vec{\omega}_b \qquad \in \mathcal{F}^b \tag{3.8c}$$

Which, when substituted back into the Euler-Lagrange formulation in Eq:3.3, yields familiar Newton-Euler rigid body differential equations of translational and rotational motion for generalized net force and torque inputs; $\vec{F}_\mu$ and $\vec{\tau}_\mu$ respectively.

$$\begin{bmatrix} m_b\dot{\vec{v}}_b + \vec{\omega}_{b/I} \times m_b\vec{v}_b \\ J_b\dot{\vec{\omega}}_b + \vec{\omega}_{b/I} \times J_b\vec{\omega}_b \end{bmatrix} - \begin{bmatrix} m_b\vec{G}_b \\ 0 \end{bmatrix} = \vec{\mathbf{V}} = \begin{bmatrix} \vec{F}_\mu \\ \vec{\tau}_\mu \end{bmatrix} \qquad \in \mathcal{F}^b \tag{3.9a}$$

$$\therefore \vec{F}_\mu = m_b\dot{\vec{v}}_b + \vec{\omega}_b \times m_b\vec{v}_b - m_b R_I^b(-\eta)\vec{G}_I \tag{3.9b}$$

$$\therefore \vec{\tau}_\mu = J_b\dot{\vec{\omega}}_b + \vec{\omega}_b \times J_b\vec{\omega}_b \tag{3.9c}$$

Reiterating that $\vec{\eta}_b \neq \vec{\eta}$ because each Euler Angle is defined in sequentially rotated reference frames $\in \mathcal{F}^{v_2,v_1,I}$. Four separate equations are then needed to completely describe a body's position and attitude states:

$$\dot{\vec{\mathcal{E}}}_I = R_b^I(-\eta)\vec{v}_b \qquad \in \mathcal{F}^I \tag{3.10a}$$

$$\vec{F}_\mu = m_b\dot{\vec{v}}_b + \vec{\omega}_b \times m_b\vec{v}_b - m_b\vec{G}_b \qquad \in \mathcal{F}^b \tag{3.10b}$$

$$\dot{\vec{\eta}} = \Phi(\eta)\vec{\omega}_b \qquad \in \mathcal{F}^{v2,v1,I} \tag{3.10c}$$

$$\vec{\tau}_\mu = J_b\dot{\vec{\omega}}_b + \vec{\omega}_b \times J_b\vec{\omega}_b \qquad \in \mathcal{F}^b \tag{3.10d}$$

Where $\Phi(\eta)$ is the Euler matrix which relates Euler rates $\dot{\vec{\eta}}$ and angular velocity $\vec{\omega}_b$, defined previously in Eq:2.12f. State differentials from Eq:3.10 can be simplified to a pair of equations defined entirely in the reference frames of the variables which they represent. The nonlinear form of those equations substitutes $d\vec{\eta}/dt = \Phi(\eta)\vec{\omega}_b$ into the Lagrangian derivative in Eq:3.4b.

$$\frac{d}{dt}\left(\frac{\delta \mathcal{L}}{\delta \dot{\mathbf{r}}}\right) = \begin{bmatrix} m_b \frac{d}{dt}\vec{v}_b & J_b \frac{d}{dt}\dot{\vec{\eta}}_b \end{bmatrix}^T \Rightarrow \begin{bmatrix} m_b \frac{d}{dt}\vec{v}_b & J_b \frac{d}{dt}\Phi(\eta)\vec{\omega}_b \end{bmatrix}^T \tag{3.11}$$

Which only affects the angular component because the two kinetic energies are independent of one another. Applying the differential chain rule to the angular component of Eq:3.11 yields:

$$J_b \frac{d}{dt}\Phi(\eta)\vec{\omega}_b = J_b\big(\dot{\Phi}(\eta)\vec{\omega}_b + \Phi(\eta)\dot{\vec{\omega}}_b\big) \tag{3.12}$$

Drawing from [98] and recognizing that J_b must be transformed to the shared intermediate Euler axes, $J \triangleq \Psi(\eta)^T J_b \Psi(\eta)$. The state differential for the Euler angle acceleration counterpart of Eq:3.9c, defined in intermediate (non-inertial) Euler frames for each respective Euler angle, then becomes:

$$M(\eta)\ddot{\vec{\eta}} + C(\eta,\dot{\eta})\dot{\vec{\eta}} = \Psi(\eta)\vec{\tau}_\mu \qquad \in \mathcal{F}^{v2,v1,I} \tag{3.13a}$$

$$M(\eta) = \Psi(\eta)^T J_b \Psi(\eta) \tag{3.13b}$$

$$C(\eta,\dot{\eta}) = -\Psi(\eta)J_b\dot{\Psi}(\eta) + \Psi(\eta)^T\big[\Psi(\eta)\dot{\vec{\eta}}\big]_\times J_b\Psi(\eta) \tag{3.13c}$$

The relationship $\dot{\Psi} \equiv \Psi\dot{\Phi}\Psi$ was used to simplify Eq:3.13, the singularity present in Φ remains. The equation in Eq:3.13a completely describes the state derivative $\ddot{\vec{\eta}}$ in its own reference frame(s) $\in \mathcal{F}^{v2,v1,I}$. The two differential equations which fully describe the entire body's 6-DOF motion are:

$$\vec{F}_\mu = m_b\dot{\vec{\mathcal{E}}}_I + R_b^I(-\eta)\vec{\omega}_b \times m_b\dot{\vec{\mathcal{E}}}_I - m_b\vec{G}_I \qquad \in \mathcal{F}^I \tag{3.14a}$$

$$\vec{\tau}_\mu = \Psi(\eta)^{-1}M(\eta)\ddot{\vec{\eta}} + \Psi(\eta)^{-1}C(\eta,\dot{\eta}) \qquad \in \mathcal{F}^{v2,v1,I} \tag{3.14b}$$

In most cases the body frame counterparts in Eq:3.10 are used rather than Eq:3.14 when describing states. Eq:3.14 is superfluous when considering that inputs $\vec{F}_\mu$ and $\vec{\tau}_\mu$ both act in the body frame $\mathcal{F}^b$. Irrespective of the differential equation used; some singular transformation will still be performed by either $\Psi(\eta)$ from Eq:2.12g or $\Phi(\eta)$ from Eq:2.12i.

The generalized input forces and torques $\vec{F}_\mu$ and $\vec{\tau}_\mu$ respectively are produced by the system's controllable inputs but could include any external disturbances acting on the body. Those control inputs are directly affected by the vehicle's actuators. How actuator action produces the control inputs depends on the actuator's associated *effectiveness* function. In the general case, which is expanded in Sec:3.2, the control inputs for a regular quadrotor (Fig:3.1) are as follows.

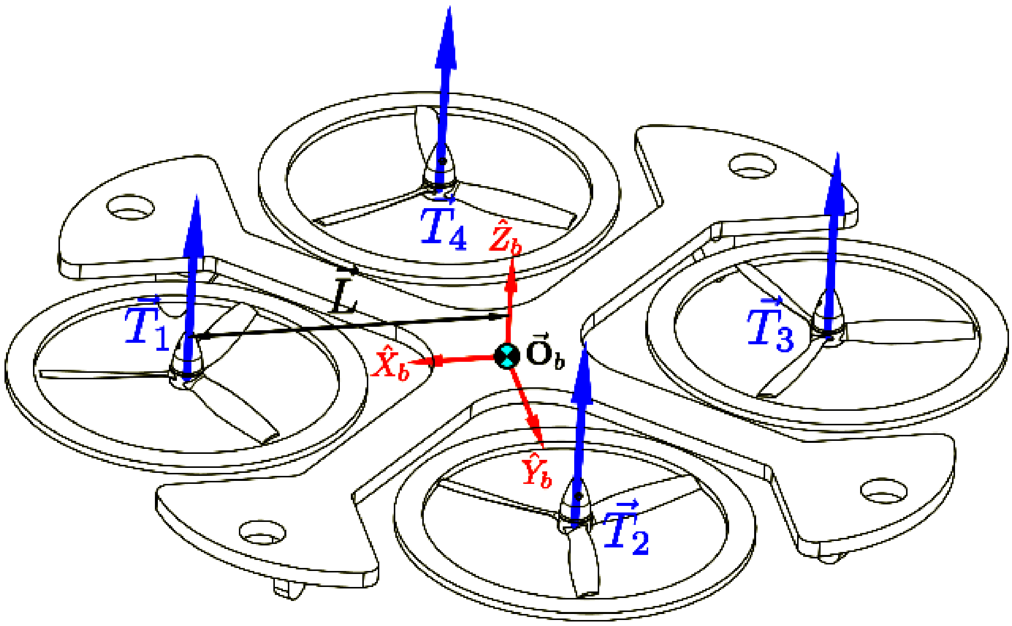


Figure 3.1: Generalized quadrotor net forces and torques

Typically $\vec{F}_\mu$ is the net heave force acting on the center of motion $\vec{\mathbf{O}}_b$. The net heave is the sum of all thrust forces produced by rotating propellers, as some function of those rotational speeds; $\vec{T}(\Omega_i)$.

$$\vec{F}_\mu = \sum_{i=1}^{4} \vec{T}(\Omega_i) \cdot \hat{Z}_b \qquad \in \mathcal{F}^b \tag{3.15a}$$

Similarly net torque $\vec{\tau}_\mu$ is the sum of all *differential* torques produced from opposing propeller thrust vectors. Each torque arm $\vec{L}_{1\to4}$ is the thrust's orthogonal displacement relative to the origin of *motion*.

$$\vec{\tau}_\mu = \sum_{i=1}^{4} \vec{L}_i \times \vec{T}(\Omega_i) \cdot \hat{Z}_b \qquad \in \mathcal{F}^b \tag{3.15b}$$

In Eq:3.15, the thrust vector $\vec{T}(\Omega_i)$ is a function of the i^{th} motor's rotational velocity Ω_i, fixed in the $\hat{Z}_b$ direction. Each thrust vector could potentially be $\in \mathbb{R}^3$ such as the redirected vector from Eq:2.17. All of the above equations are still applicable to any 6-DOF body; common simplifications applied to the system for quadrotor control are explored in App:A.1. Aerodynamic components pertinent for thrust and torque generation relative to Eq:3.15 are now introduced; obviously the contextual focus is on quadrotor with dual-tilting axis actuators...

3.2 Aerodynamics

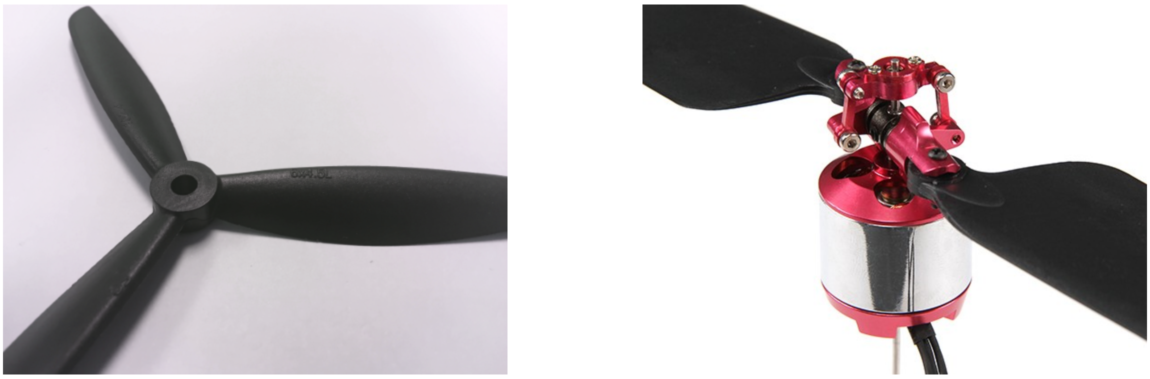
Aerodynamic effects detailed here and subsequent nonlinear multibody responses in Sec:3.4 both affect the generalized forces and torques acting on the body. The relationship between a propeller's rotational speed Ω_i, in revolutions per second or *RPS*, and its perpendicular thrust vector $\vec{T}(\Omega_i)$ is more complicated than the quadratic simplification taken at static conditions which most papers assume (e.g [110] etc. . .). Produced thrust is mostly dependent on the incident air stream flowing through the propeller's rotational plane; typically being the body velocity's component 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 [102,117] and fluid-dynamics momentum or *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 with low translational velocities.

3.2.1 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 [132] 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:3.2a) and not variable pitch swash-plate actuated propellers (Fig:3.2b).



(a) Twisted, fixed pitch **(b)** Swash-plate variable pitch; [58]

Figure 3.2: 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:3.3) has an incident upstream velocity, v_∞, and a resultant slip velocity, v_s, downstream relative to the rotational plane. The change of fluid flow as a result of the propeller's rotation can be given as:

$$v_s = \Delta v + v_\infty \tag{3.16}$$

Where Δv 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; $v_i \neq \Delta v$.

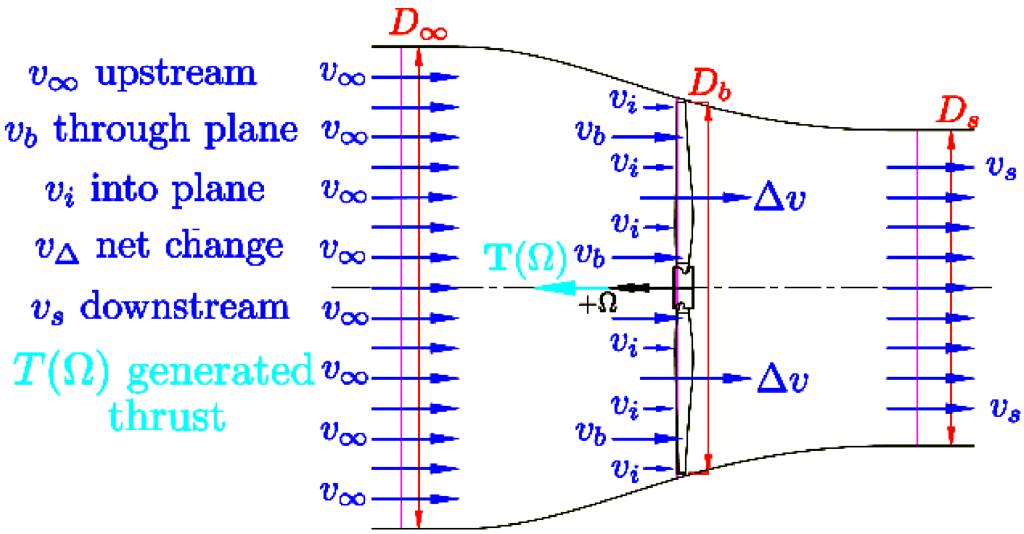


Figure 3.3: Disc Actuator Propeller Planar Flow

It is shown in [8] that at static conditions, using Bernoulli's pressure theorem, the net fluid flow through the propeller's plane is:

$$v_b = \frac{1}{2}(v_s - v_\infty) = \frac{1}{2}\Delta v = \frac{1}{2}v_s\big|_{v_\infty=0} \tag{3.17}$$

Stemming from classical disc actuator, or fluid *momentum* theory [113], the scalar force $T(\Omega_i)$ acting on the fluid is calculated as a function of mass flow rate with respect to the change in fluid velocity or *pressure differential*:

$$T = (A_b v_b)\Delta v = \rho\pi R_b^2 v_b \Delta v = \rho\pi R_b^2 (v_i + v_\infty)\Delta v = \frac{1}{2}\rho\pi R_b^2 \Delta v^2 \tag{3.18}$$

Where R_b is the disc (propeller) radius in m for the fluid stream under consideration, A_b is the swept area of that propeller disc. The fluid density of that stream, ρ, is typically 1.225 kg.m^{-3} at standard temperature and pressure (*stp*). However, the desired form of thrust generated is as a function of propeller rotational velocity, $T(\Omega_i)$ in RPS, so Eq:3.18 is not yet satisfactory.

Eq:3.18 could be solved from the aerodynamic propulsive power expended; using $\Delta P = T\Delta v$. The relationship between rotational kinetic energy of a propeller and its transferred propulsive power is difficult to quantify, compound parasitic losses deteriorate the efficiency of the propeller and motor. Furthermore, the fluid velocity through the propeller's plane is not purely normal but is in fact a vector.

Fluid flow induced by the propeller's rotation v_i directly in front of its plane of rotation has both axial and tangential induced components, termed a and a' respectively. Induced fluid velocity components are abstracted to induction factors which are dependent on the incident fluid velocity entering the propeller's plane of rotation:

$$v_i = a v_\infty \quad \text{in the axial direction} \tag{3.19a}$$

$$v_\theta = a'\Omega_i R_b \quad \text{in the tangential direction} \tag{3.19b}$$

Using the induction factors to rewrite the fluid's through velocity v_b and its slip stream velocity v_∞:

$$v_b = (1 + a)v_\infty \tag{3.20a}$$

$$v_s = (1 + 2a)v_\infty \tag{3.20b}$$

A consequence of the tangential fluid flow is that an angular momentum flow rate exists across the propeller plane. This produces a fluid-momentum torque opposing the rotational motion about the propeller's axis, analogous but perpendicular to Eq:3.18:

$$H = \rho\pi R_b^3 (v_\theta - v_\infty) v_b \tag{3.21}$$

Together, Eq:3.18 and Eq:3.21 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 at some radius r, the sectional view of which is illustrated in Fig:3.4. Each aerofoil element has a net local fluid velocity $\vec{U}$ across its profile, calculated as:

$$\vec{U} = \sqrt{(v_\infty + v_i)^2 + (v_\Omega + v_\theta)^2} \tag{3.22}$$

Where each profile has a chord length c and an inclination (or *pitch*) θ of the aerofoil *zero-lift line* relative to the horizontal. Local fluid velocities incident to the propeller profile (Fig:3.4) make their own angle of attack ϕ such that a true effective angle of attack α_{eff} is encountered:

$$\phi = \theta - \alpha_{eff} \tag{3.23}$$

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

$$\phi = tan^{-1}\left(\frac{v_\infty + v_i}{v_\Omega + v_\theta}\right) = tan^{-1}\left(\frac{v_\infty(1+a)}{\Omega r(1+a')}\right) \tag{3.24}$$

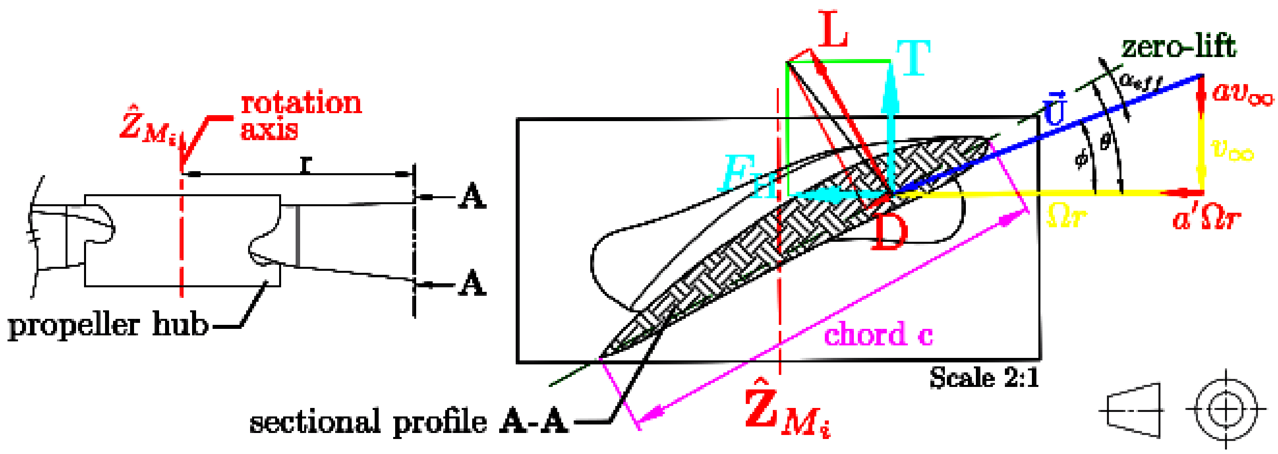


Figure 3.4: Blade element profile at radius r

In-plane fluid flow $\vec{U}(r,\phi)$, for an element at radius r with a local angle of attack ϕ, 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 flat propeller (Fig:3.2b).

$$\Delta L = \frac{1}{2}\rho\vec{U}(r,\phi)^2 cC_L \tag{3.25a}$$

$$\Delta D = \frac{1}{2}\rho\vec{U}(r,\phi)^2 cC_D \tag{3.25b}$$

With air density ρ again at *stp*. Lift and drag forces, when taken parallel and perpendicular to the plane of rotation, are thrust T and torque F_H forces (Fig:3.4). The in-plane force applies an aerodynamic torque H at the propellers hub because the force F_H acts at a radius r, [60].

$$dT = \frac{1}{2}\rho\vec{U}(r,\phi)^2 c\big(C_L cos(\theta) + C_D sin(\theta)\big).dr \tag{3.26a}$$

$$dF_H = \frac{1}{2}\rho\vec{U}(r,\phi)^2 c\big(C_L sin(\theta) + C_D cos(\theta)\big).dr \tag{3.26b}$$

$$\rightarrow dH = \frac{1}{2}\rho\vec{U}(r,\phi)^2 c\big(C_L sin(\theta) + C_D cos(\theta)\big)r.dr \tag{3.26c}$$

$$\rightarrow dP = \Omega r dF_H.dr \tag{3.26d}$$

Rotational power expended is a product of angular velocity and the opposing in-plane torque; Eq:3.26d. Power is mostly used instead of torque or drag terms in Eq:3.26c or Eq:3.26b respectively. Calculating forces and power terms as per momentum theory for each element, in terms of axial and tangential induction factors:

$$dT = \rho 4\pi r^2 v_\infty (1+a)a.dr \tag{3.27a}$$

$$dP = \rho 4\pi r^2 v_\infty (1+a)\Omega r(1+a').dr \tag{3.27b}$$

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 α_{eff}, typically characterized as:

$$C_L = a_L(\theta - \phi) \tag{3.28}$$

Firstly the lift coefficient curve gradient a_L is shown in [62] for an ideally twisted blade, like the fixed pitch propellers under consideration, to be 2π. An ideal lift coefficient is then a function:

$$C_L = 2\pi(\theta - \phi) \tag{3.29}$$

Secondly assuming tangentially induced velocities, v_θ, are small when compared to the propeller's translational speed at radius r, $v(\Omega_i) = \Omega_i r$. The tangential induction factor a' is then the ratio:

$$a' = \frac{v_\theta}{\Omega_i r} << 1 \tag{3.30}$$

Small angle approximations then apply to Eq:3.26a-3.26c; $cos(\phi + \alpha_{eff}) \approx 1$ and $sin(\phi + \alpha_{eff}) \approx \phi + \alpha_{eff}$. Similarly net inflow and axial velocities are $(v_\infty + v_i) << \Omega_i r$, the following integrals are then found:

$$T(\Omega_i) = \int_{r=0}^{R} \frac{1}{2} a_L bc\rho(\Omega_i r)^2 \left[\theta - \frac{v_\infty + v_i}{\Omega_i r}\right].dr \tag{3.31a}$$

$$P(\Omega_i) = \int_{r=0}^{R} \frac{1}{2} a_L bc\rho(\Omega_i r)^3 \left[(\theta - \frac{v_\infty + v_i}{\Omega_i r})(\frac{v_\infty + v_i}{\Omega_i r}) + C_d\right].dr \tag{3.31b}$$

Where b is the number of blades the propeller has. In practice 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:3.31a and Eq:3.31b can be solved by equating element and momentum terms, a full solution of which is given in App:A.2. Often dimensionless thrust and power coefficients are defined across the entire blade's length:

$$C_T(J) \triangleq \frac{T}{\rho \Omega_i^2 D^4} \tag{3.32a}$$

$$C_P(J) \triangleq \frac{P}{\rho \Omega_i^3 D^5} \tag{3.32b}$$

Where the propeller's diameter is D in m, then Ω_i is the propeller's rotational speed in *revolutions per second* (*RPS*) and different from other inertial equations like Eq:3.63, with units rad.s^{-1}. For fixed pitch propellers the thrust and power coefficients are easily determined and remain consistent. Both Eq:3.32a and Eq:3.32b vary as a function of the dimensionless *advance ratio* J.

$$J \triangleq \frac{v_\infty}{\Omega_i R} \tag{3.33}$$

Typically the net upstream velocity v_∞ in Eq:3.33 is simply the perpendicular component (projected onto the plane's normal vector $\hat{n}$, shown later in Eq:3.35) 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 [26] 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 [27].

The introduction of those coefficients drastically reduces thrust estimation complexity. For a typical 6×4.5 inch propeller the following coefficients were linearly interpolated from similar pitched database results in [26] to match subsequent physical test values. Static thrust and power coefficients determined from tests subsequently in Fig:3.6b and Fig:3.7b are respectively:

$$C_{T0} = 0.191 \tag{3.34a}$$

$$C_{P0} = 0.0877 \tag{3.34b}$$

Fig:3.5 plots interpolated coefficients for thrust C_T and power C_P as a function of the advance ratio J. As the incident upstream fluid velocity v_∞ 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.

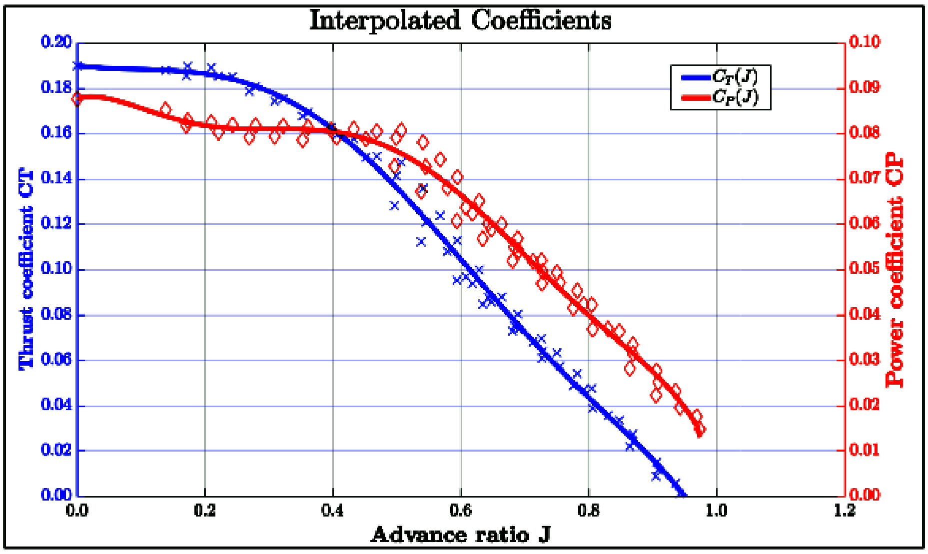
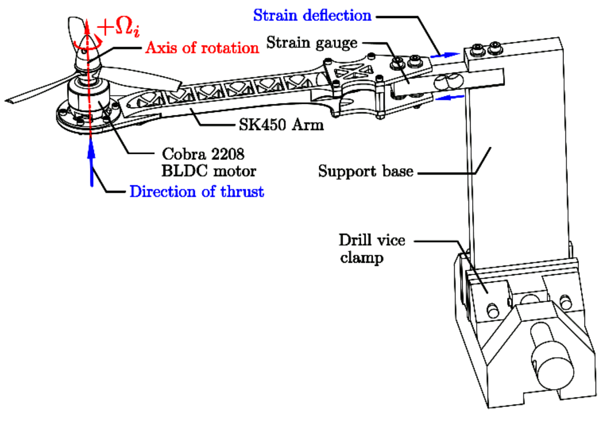
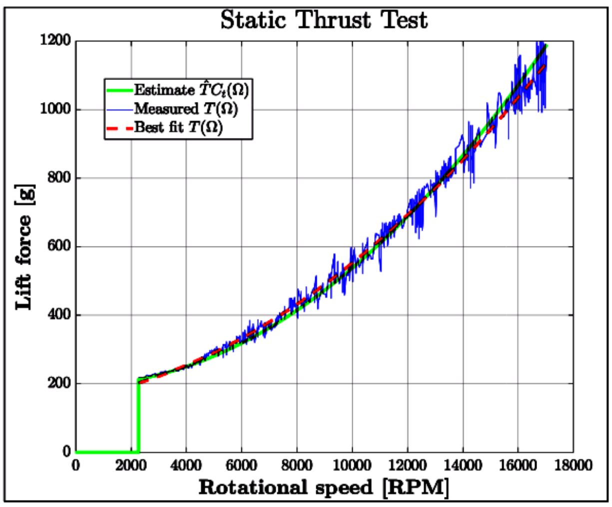


Figure 3.5: Thrust and power coefficients

Static thrust and torque tests were respectively performed on test rigs in Fig:3.6a and Fig:3.7a. Measured values for each test are plotted; $T(\Omega)$ in Fig:3.6b for thrust and $H(\Omega)$ in Fig:3.7b for torque. The physically tested values are fitted with quadratic trend-lines and plotted against static coefficient estimates using Eq:3.32a for thrust $\hat{T}C_t(\Omega)$ and Eq:3.32b for calculated torque $\hat{H}C_p(\Omega)$. Results from Fig:3.5 are used as a lookup table and values from Eq:3.32 are calculated, induced propeller thrust and torques can be accurately modelled quadratically, power is cubic with respect to rotational velocity.

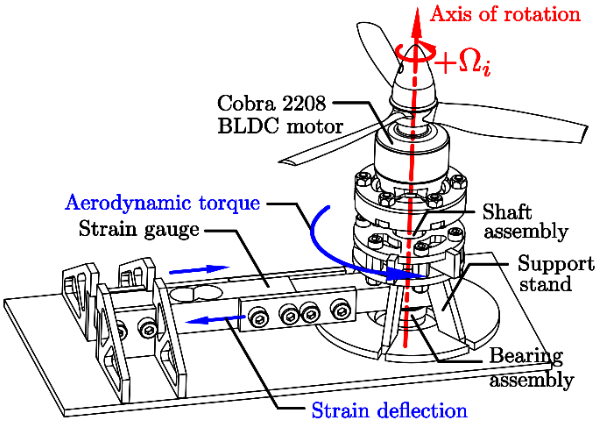


(a) Propeller thrust test rig

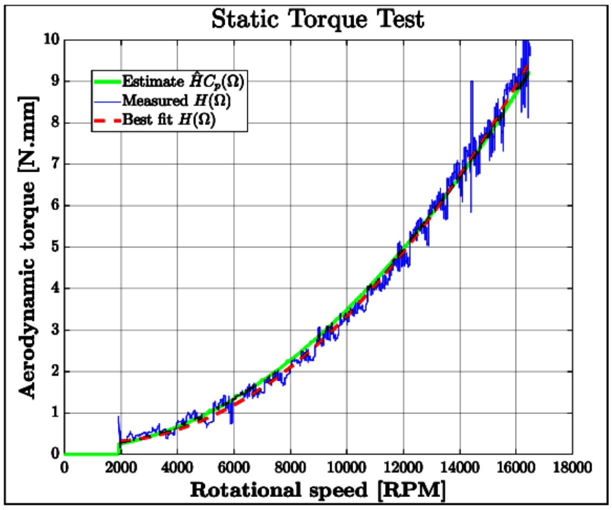


(b) Static lift force results

Figure 3.6: Propeller thrust tests



(a) Aerodynamic drag torque test rig



(b) Torque plot

Figure 3.7: Static induced torque results

Advance ratios, Eq:3.33, or rather the propeller incident fluid flows are dependent on the vehicle's net translational and angular velocity. Such that the fluid velocity's normal component to the propeller plane is given by:

$$v_\infty = (\vec{v}_b{}' + \vec{L}_{arm} \times \vec{\omega}_b{}') \cdot \hat{n}(\lambda_i, \alpha_i) \qquad \in \mathcal{F}^{M_i} \tag{3.35}$$

Where $\vec{v}_b{}'$ in m.s^{-1} is the body's translational velocity and $\vec{\omega}_b{}'$ in 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:3.33.

It is 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 [26].

Clockwise and anti-clockwise propellers and rotations were used for both thrust and torque tests. Despite both test rigs (Fig:3.6a and Fig:3.7a respectively) having been designed to specifically isolate each response, results from opposing directional tests were averaged in the hopes that stray opposing effects were cancelled. Both clockwise and anti-clockwise rotational testing results for thrust and torque measurements are included in App:C.1

Discrepancies which exist between the model or coefficient values derived can be accounted for with lumped uncertainty disturbance terms. Model uncertainty compensation can easily be incorporated into adaptive backstepping or H_∞ control algorithms. The deviation of the modelled thrust or torques from their true values would be simple to incorporate into a plant dependent Lyapunov candidate function; Sec:4.6.3.

3.2.2 Hinged Propeller Conning & Flapping

Aerodynamics which adversly affect a propeller's performance have all been well documented in their own right; mostly in the context of helicopter aerodynamic and propeller fields [25,123]. Typically such effects are more pronounced when observing hinged variable pitch propellers (Fig:3.2b), fixed pitch propellers with small radii have a diminished effect. Moreover, low translational velocities suppress such responses but they're worth mentioning.

Conning and flapping are the two most significant aerodynamic effects encountered by a propeller. Other phenomenon like cyclic vortex ring states are deemed to be inapplicable here and fall outside the scope of the investigation.

In translational flight, for a propeller without shrouding or a ducting, each blade encounters varying incident fluid flow throughout its cycle. The advancing blade relative to the body's translational direction encounters a greater fluid flow than the retreating blade, constructive and destructive interference from the body's translational velocity adds to local fluid flows. The effective local angle of attack, sectional Fig:3.4, for advancing and retreating propeller blades are then asymmetrical. Unbalanced angles of attack produce a dissymmetry of lift across the propeller blade's surface.

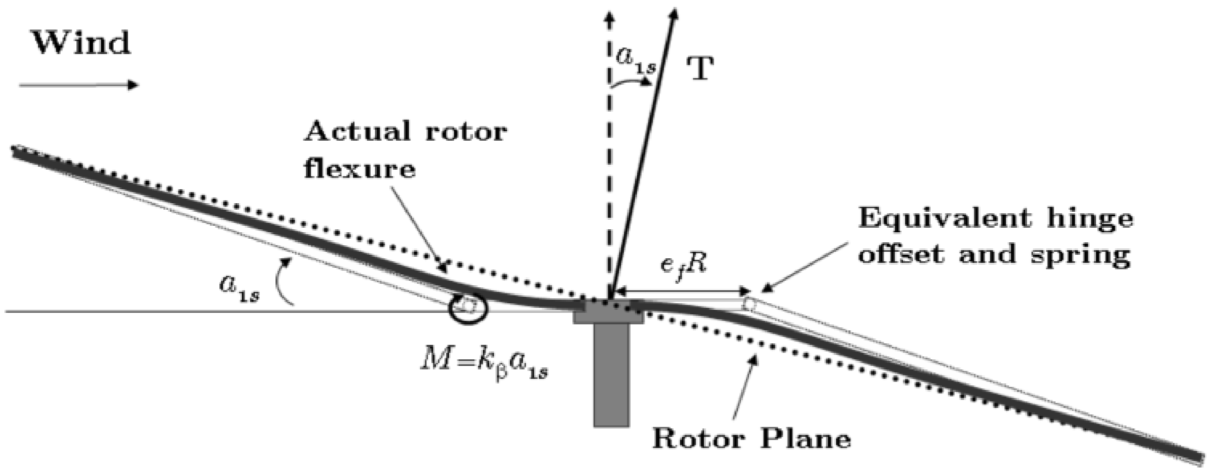


Figure 3.8: Propeller blade flapping; from [60]

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 and hence its susceptibility to deflection. The flapping pitches the effective propeller plane or *tip-path plane*, and hence the thrust vector line, away from its principle axis; shown in Fig:3.8.

The propeller's resultant thrust vector is pitched away from its perpendicular normal by some deflection angle, α_{1s} in Fig:3.8, 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.

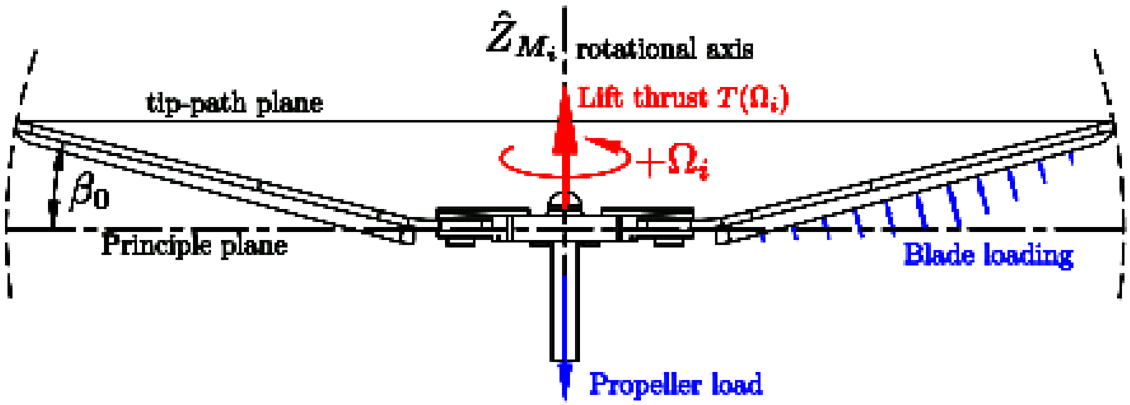


Figure 3.9: Propeller coning

Coning is another form of propeller deflection, illustrated in Fig:3.9, which again is dependent on the blade material's 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:3.31a. Increased loading accentuates the coning angle experienced by the propellers and as such reduces the tip-path plane.

Both aerodynamic propeller deflections can be quantified numerically. Their derivation and resultant equations are cumbersome however. In practice both effects on the produced prototype are not significant enough to affect the derived plant model. 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...

3.2.3 Drag

For any solid body with some non-zero relative translational velocity through a fluid, that fluid has a second order damping response opposing the body's movement. Net drag $\vec{D}_{net}$ is locally dependent on individual component cross-sections, for a vehicle's velocity $\vec{v}_b = [u\ v\ w]^T$ in $\mathcal{F}^b$, the drag force is:

$$\vec{D}_{net}(\vec{v}_b) = \begin{bmatrix} D_{ii} & D_{ij} & D_{ik} \\ D_{ji} & D_{jj} & D_{jk} \\ D_{ki} & D_{kj} & C_{kk} \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}^2 \qquad \in \mathcal{F}^b \tag{3.36}$$

Each drag coefficient's subscript; $\hat{\imath}, \hat{\jmath}$ and $\hat{k}$ is dependent on the body's directional cross-section area 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 are of little or no consequence and as such the drag equation can be simplified to the diagonal:

$$\vec{D}_{net}(\vec{v}_b) \approx diag\big(D_{ii},\ D_{jj},\ D_{kk}\big)\vec{v}_b^2 \qquad \in \mathcal{F}^b \tag{3.37}$$

Due to the second order degree of translational velocity on the drag force; such terms can be relegated to a lumped disturbance terms to be compensated for in the control loop, Sec:4.6.3. The time scale separation between velocity and wind drag effects within the control loop accommodates such an assumption. Analogous rotational drag-like effects opposing angular rates exist but, for the intents and purposes of most practical flight envelopes, can be disregarded.

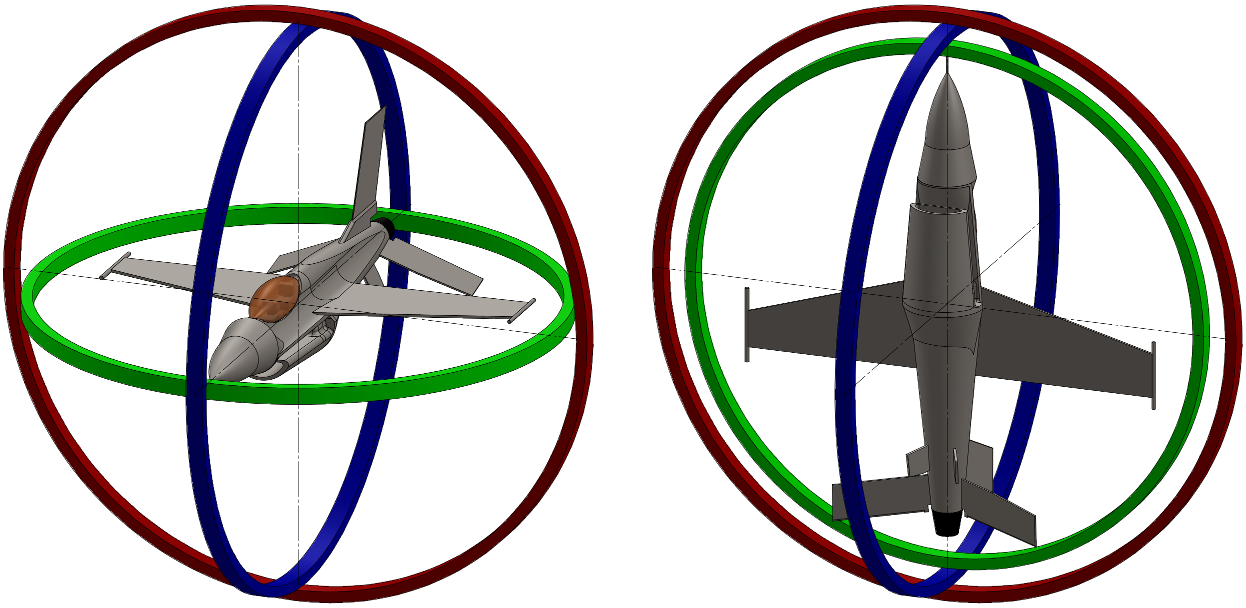
In simulation; if the plant has sufficient disturbance rejection then the drag term in Eq:3.36 would be easily accounted for in an adaptive backstepping algorithm. Drag, much like wind turbulence, is shown later in Sec:6.6 to be not consequential enough to be destabilizing. Furthermore it is possible to physically test for the drag coefficients to attain a higher certainty model but, given the flight conditions proposed for this research, such effects will be small if not negligible. As such those tests are outside the scope of investigation here...

3.3 Quaternion Attitude

3.3.1 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. In general, a singularity occurs for some matrix A in $\vec{y} = A\vec{x}$ when the matrix has a zero determinate; losing rank and hence differentiability of $\vec{y}$ in terms of $\vec{x}$. The combined rotation matrix from the inertial frame $\mathcal{F}^I$ to the body frame $\mathcal{F}^b$ is the singular component of an Euler parametrized sequence.

Considering the case of a rotational 3-axis gimbal system, illustrated in Fig:3.10a, which mimics the sequential nature of the Euler set. When the intermediary sequenced rotational angle is at $\pi/2$ rad, the remaining two axes become co-linear, Fig:3.10b. In a Z-Y-X rotation sequence, as adopted in this work, the singularity occurs from the rolling angle θ about the $\hat{Y}$ axis. Both the pitch ϕ and yaw ψ rotations will subsequently have the same rotational effect. Such a situation results in as a loss of a degree of freedom.



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

Figure 3.10: 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:2.12h 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:

$$\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} = \begin{bmatrix}1 & sin(\phi)tan(\theta) & cos(\phi)tan(\theta)\\ 0 & cos(\phi) & -sin(\phi)\\ 0 & sin(\phi)sec(\theta) & cos(\phi)sec(\theta)\end{bmatrix}\begin{bmatrix}p\\ q\\ r\end{bmatrix} = \Phi(\eta)\omega_b \quad \in \mathcal{F}^{v_1,v_2,I} \tag{3.38}$$

$$\rightarrow det\big(\Phi(\eta)\big) = cos(\phi)\big(cos(\phi)sec(\theta)\big) + sin(\phi)\big(\sin(\phi)sec(\theta)\big) = sec(\theta) \tag{3.39}$$

$$\therefore \lim_{\theta\rightarrow\pi/2} |\Phi(\eta)| = sec(\theta) \rightarrow \infty \tag{3.40}$$

The Euler matrix $\Phi(\eta)$ looses rank as $\theta \rightarrow \pi/2$ rad, 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:2.6.

$$R_I^b(\eta) \triangleq R_z(\psi)R_y(\theta)R_x(\phi) = \begin{bmatrix}c_\psi & -s_\psi & 0\\ s_\psi & c_\psi & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}c_\theta & 0 & s_\theta\\ 0 & 1 & 0\\ -s_\theta & 0 & c_\theta\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ 0 & c_\phi & -s_\phi\\ 0 & s_\phi & c_\phi\end{bmatrix} \tag{3.41a}$$

$$\therefore R_I^b(\eta) = \begin{bmatrix}c_\psi c_\theta & c_\psi s_\theta s_\phi - s_\psi c_\phi & c_\psi s_\theta c_\phi + s_\psi s_\phi\\ s_\psi c_\theta & s_\psi s_\theta s_\phi + c_\psi c_\phi & s_\psi s_\theta c_\phi - c_\psi s_\phi\\ -s_\theta & c_\theta s_\phi & c_\phi c_\theta\end{bmatrix} \tag{3.41b}$$

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

$$R_I^b(\eta) = \begin{bmatrix}0 & c_\psi s_\phi - s_\psi c_\phi & c_\psi c_\phi + s_\psi s_\phi\\ 0 & s_\psi s_\phi + c_\psi c_\phi & s_\psi c_\phi - c_\psi s_\phi\\ -1 & 0 & 0\end{bmatrix}\Bigg|_{\theta=\pi/2} \tag{3.41c}$$

$$= \begin{bmatrix}0 & s(\phi-\psi) & c(\phi-\psi)\\ 0 & c(\phi-\psi) & s(\phi-\psi)\\ -1 & 0 & 0\end{bmatrix} \tag{3.41d}$$

$$= R_{x'}(\phi-\psi) \tag{3.41e}$$

Where the resultant in Eq:3.41e 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 both ϕ and ψ effect the same angle.

3.3.2 Quaternion Dynamics

An algorithm proposed in [125] 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 compulationally exhaustive. Far more elegant is the use of *quaternion* attitude representations in $\mathbb{R}^4$, used in [49, 76] amongst others but most notably made popular by [124] for use in animation.

A quaternion is analogous to a rotation matrix in that it represents an attitude difference between two reference frames. An $\mathbb{R}^3$ attitude is paramterized as one rotation θ about a single unit *Euler* axis $\hat{u}$, demonstrated using the Rodriguez Formula in [92]. In brief a quaternion consists of a scalar component q_0 and complex vector component $\vec{q} \in \mathbb{C}^3$ such that:

$$Q \triangleq \begin{bmatrix} q_0 \\ \vec{q} \end{bmatrix} \quad \in \mathbb{R}^4 \tag{3.42}$$

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

$$R_I^b(\eta) \equiv R(Q_b) \triangleq \mathbb{I}_{3\times3} + 2q_0[\vec{q}]_\times + 2[\vec{q}]_\times{}^2 \tag{3.43}$$

Where $[.]_\times$ is the cross-product matrix defined previously in Eq:2.8c. All quaternions, unless otherwise specified, are unit quaternions $Q \in \mathbb{Q}_u$. Quaternions with a unity magnitude ensure that rotational operations maintain the vector operand's magnitude. A unit quaternion is defined:

$$\|Q\| = \sqrt{q_0{}^2 + \vec{q}^{\,2}} = 1 \tag{3.44}$$

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

$$Q \otimes P = \begin{bmatrix} q_0 \\ \vec{q} \end{bmatrix} \otimes \begin{bmatrix} p_0 \\ \vec{p} \end{bmatrix} \tag{3.45a}$$

$$\triangleq \begin{bmatrix} q_0 p_0 - \vec{q}\cdot\vec{p} \\ q_0\vec{p} + p_0\vec{q} + \vec{q}\times\vec{p} \end{bmatrix} \tag{3.45b}$$

$$= \underbrace{q_0 p_0 - \vec{q}\cdot\vec{p}}_{scalar} + \underbrace{p_0\vec{q} + q_0\vec{p} + \vec{q}\times\vec{p}}\, vector \tag{3.45c}$$

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

$$Q^* \triangleq \begin{bmatrix} q_0 \\ -\vec{q} \end{bmatrix} \tag{3.46}$$

It follows that the fundamental quaternion identity is:

$$Q \otimes Q^* = \mathbb{I}_{4\times4} \tag{3.47}$$

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

$$\begin{bmatrix} 0 \\ \vec{\nu}\,' \end{bmatrix} = Q \otimes \begin{bmatrix} 0 \\ \vec{\nu} \end{bmatrix} \otimes Q^* \tag{3.48}$$

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

$$\vec{\nu}\,' = Q \otimes (\vec{\nu}\,) \otimes Q^* \tag{3.49}$$

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

$$\vec{\nu}_b = R_I^b(\eta)\vec{\nu}_I \underset{Q}{\iff} Q_b \otimes (\vec{\nu}_I) \otimes Q_b^* \qquad \in \mathcal{F}^b \tag{3.50}$$

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 simpler for describing a body's orientation. A Z-Y-X sequenced body quaternion Q_b relative to the inertial frame can be constructed from its Euler angle counterparts using:

$$Q_b = Q_z \otimes Q_y \otimes Qx = \begin{bmatrix} cos(\psi/2) \\ 0 \\ 0 \\ sin(\psi/2) \end{bmatrix} \otimes \begin{bmatrix} cos(\theta/2) \\ 0 \\ sin(\theta/2) \\ 0 \end{bmatrix} \otimes \begin{bmatrix} cos(\phi/2) \\ sin(\phi/2) \\ 0 \\ 0 \end{bmatrix} \tag{3.51}$$

A quaternion's time derivative, defined in [41], with Q_ω being a quaternion with a vector component equal to angular velocity $\vec{\omega}_{b/I}$ and a zero scalar component, is:

$$\frac{d}{dt}Q_b = \frac{1}{2}Q_b \otimes Q_\omega = \frac{1}{2}Q_b \otimes \vec{\omega}_b \tag{3.52a}$$

$$= \begin{bmatrix} -\frac{1}{2}\vec{q}^T\vec{\omega}_b \\ \frac{1}{2}\left([\vec{q}]_\times + q_0\mathbb{I}\right)\vec{\omega}_b \end{bmatrix} \tag{3.52b}$$

Using quaternions to represent attitudes negates the need for an Euler Matrix, $\Phi(\eta)$ from Eq:2.12i, 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 in Eq:3.10a and Eq:3.10c respectively:

$$\dot{\mathcal{E}} = R_b^I(-\eta)\vec{v}_b \underset{Q}{\iff} Q_b(-\eta) \otimes \vec{v}_b \otimes Q_b^*(-\eta) = Q_b^* \otimes \vec{v}_b \otimes Q_b \qquad \in \mathcal{F}^I \tag{3.53a}$$

$$\dot{\eta} = \Phi(\eta)\vec{\omega}_b \quad \in \mathcal{F}^{v2,v1,I} \underset{Q}{\iff} \dot{Q}_b = \frac{1}{2}Q_b \otimes \vec{\omega}_b \tag{3.53b}$$

Second order time derivatives for quaternion acceleration are not as useful as their higher order, velocity counterparts. The second order derivative is provided here for the sake of completeness. If at all possible, quaternion accelerations are avoided due to their complexity. The quaternion analogue for angular acceleration Eq:3.14b, dependent on net torque acting on a body $\vec{\tau}_\mu$, is given by:

$$\ddot{Q}(\dot{Q}, Q, t) = \dot{Q} \otimes Q^* \otimes \dot{Q} + \frac{1}{2}Q \otimes \left[J_b^{-1}(\vec{\tau}_\mu - 4(Q^* \otimes \dot{Q}) \times (J_b(Q^* \otimes \dot{Q}))\right] \tag{3.54}$$

An Euler angle attitude error state, used for control input, is defined as the subtracted error between a desired and an existing attitude orientation; $\vec{\eta}_d$ and $\vec{\eta}_b$ respectively. Where $\vec{\eta}_d$ is some attitude setpoint produced from a trajectory generator.

$$\vec{\eta}_e \triangleq \vec{\eta}_d - \vec{\eta}_b \tag{3.55}$$

Quaternion attitude control and its stability goals are expanded upon subsequently in Sec:4.6.1. In contrast with Eq:3.55, a quaternion attitude error is a multiplicative term defined as the difference between two quaternions Q_d and Q_b;

$$Q_e \triangleq Q_b^* \otimes Q_d \tag{3.56}$$

3.3.3 Quaternion Unwinding

Although quaternions are indeed better than their Euler angle attitde counterparts and lack the associated singularity they do contain one caveat. Becuase 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, [92].

Each unit quaternion, stemming from Euler-Rodriguez theorem, represents a single Euler-axis rotation of θ about a unit axis $\hat{u}$ such that:

$$Q = \begin{bmatrix} q_0 \\ \vec{q} \end{bmatrix} \triangleq \begin{bmatrix} cos(\theta/2) \\ sin(\theta/2)\hat{u} \end{bmatrix} \tag{3.57}$$

That rotation is applied with a quaternion operator, Eq:3.49. For every attitude state in 3-D there exist two unique quaternions which correspond to the same orientation, differing by their rotational direction about the Euler-axis. The rotation angle θ 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:

$$Q_b = \begin{bmatrix} cos(\theta/2) \\ sin(\theta/2)\hat{u} \end{bmatrix} \tag{3.58a}$$

$$Q_b = \begin{bmatrix} cos(\pi - \theta/2) \\ sin(\pi - \theta/2)\hat{u} \end{bmatrix} = \begin{bmatrix} -cos(\theta/2) \\ sin(\theta/2)\hat{u} \end{bmatrix} \tag{3.58b}$$

$$\vec{\eta} \in \mathbb{R}^3 \underset{Q}{\Longleftrightarrow} \begin{bmatrix} \pm q_0 \\ \vec{q} \end{bmatrix} \in \mathbb{R}^4 \tag{3.58c}$$

Eq:3.58c 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 that two possible error state trajectories exist for every attitude difference. Both a clockwise, $+\theta$, and an anticlockwise, $2\pi - \theta$, rotation points to the same quaternion attitude error state. This could lead to an erroneous and unnecessary "unwinding" of a complete counter revolution. So for attitude controllers the requirement is that for positive and negative quaternion scalars the control input is consistent:

$$\vec{\tau}_d = h([q_0\ \vec{q}]^T, t) = h([-q_0\ \vec{q}]^T, t) \tag{3.59}$$

Or more simply that $Q_e \triangleq [|q_0|\ \vec{q}]^T$. The simplest 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 coordinates generated is sufficiently fine the control plant will not encounter a problem.

One proposal presented in [28] suggested using a *signum* operator to design the controller coefficient sign for the desired virtual angular velocity, $\vec{\omega}_d$ control plant input.

$$\vec{\omega}_d = \frac{2}{\Gamma_1} sgn(q_0)\vec{q} \tag{3.60a}$$

With Γ_1 being a proportional error coefficient and signum defining the operator's sign:

$$sgn(q_0) = \begin{cases} 1 & q_0 \geq 0 \\ -1 & q_0 < 0 \end{cases} \tag{3.60b}$$

Eq:3.60 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 using Eq:3.10d.

In [11], 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:4.6.3 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.

3.4 Multibody Nonlinearities

The unique component of the prototype's design which facilitates redirection of a propeller's thrust vector (Eq:2.17 and Sec:2.1.1) is also what makes finding the complete equations of motion drastically more complex. The relative (rotary) motion within the multibody system results in torque responses opposing those angular accelerations. Such induced responses, if left unmodelled, would almost definitely destabilize the attitude plant. Unmodelled inertia rate responses are shown to be destabilizing in [77]. Typically multibody dynamics are solved and simulated as a series of interacting torque and force constraints. There are different schools of thought on the subject, each proposing methodologies for stepping through the systems dynamics; *e.g* Implicit Euler integration [71, 143]. . .

The prototype investigated here is a multibody system connected with revolute joints, which permit a single degree of relative rotation between each connected rigid body. There are no translational degrees of freedom between each body. Opposed to the angular accelerating actuator action on a body are *gyroscopic* and *inertial* Newtonian torque responses. The responses from each body are solved independently and those excitation induced torque constraints are introduced as additive external torques to the dynamic model derived in Sec:3.1.1.

A distinction must be made between torque responses here and those previously in Eq:3.10d. Recalling the classical differential equation of angular motion already derived:

$$\dot{\vec{\omega}}_b = J_b^{-1}\big(- \vec{\omega}_b \times J_b \vec{\omega}_b + \vec{\tau}_\mu \big) \qquad \in \mathcal{F}^b \tag{3.61}$$

Eq:3.61 treats the entire body as rigid; included terms are as a result of the entire multibody's collective motion. What follows is an extension of that attitude state to incorporate relative movements between each connected body. The objective here is to model the multibody dynamic system with clear responses induced from servo rotations of inner and middle ring bodies, $\Delta\lambda_i$ and $\Delta\alpha_i$ respectively. The subsequent derivations are Lagrangian analytical dynamics applied to the multibody system under consideration. For the purposes of this derivation it is assumed that no potential energy can be stored within the structure from material flexure. The only potential energy contribution is as a result of gravitational potential energy.

Alternatively the net dynamics could indeed be derived from a Lagrangian for the *entire* 13 body dynamic system. Where those connected bodies are; four rotor/propeller bodies (Fig:2.11), four inner ring bodies (Fig:2.12), four middle ring bodies (Fig:2.13) and finally the frame structure (Fig:2.17) each with six degrees of freedom. Constraints on the assembly's joints would eventually reduce the degrees of freedom and simplify solving for net responses. The purpose here is to model the body's response to changes in the actuation servos' positions $\Delta\lambda_i$ and $\Delta\alpha_i$ so independent bodies are analyzed first. The final result is, in fact, a Lagrangian for those collective thirteen bodies, whose partial derivative with respect to the net angular velocity relative to the inertial frame $\partial\vec{\omega}_b$ produces the net torque acting on the system.

3.4.1 Relative Rotational Gyroscopic & Inertial Torques

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.

Each of the four motor modules are symmetrical and so the induced torque response characteristics from one module can be extrapolated simply through a $\hat{Z}_b$ reference frame rotation. Each motor module is positioned relative to the body frame's 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 energy component for the inner ring assembly attached to frame $\mathcal{F}^{M_i}$. There is no relative translational motion between each connected body and thus no translational kinetic energy contributions exist. The translational kinetic energy for each module is an extension of body's net kinetic energy in Eq:3.7 and independent of any actuator's position. The motor module's translational motion is incorporated in Eq:3.14a and not considered here. Regarding the i^{th} motor module...

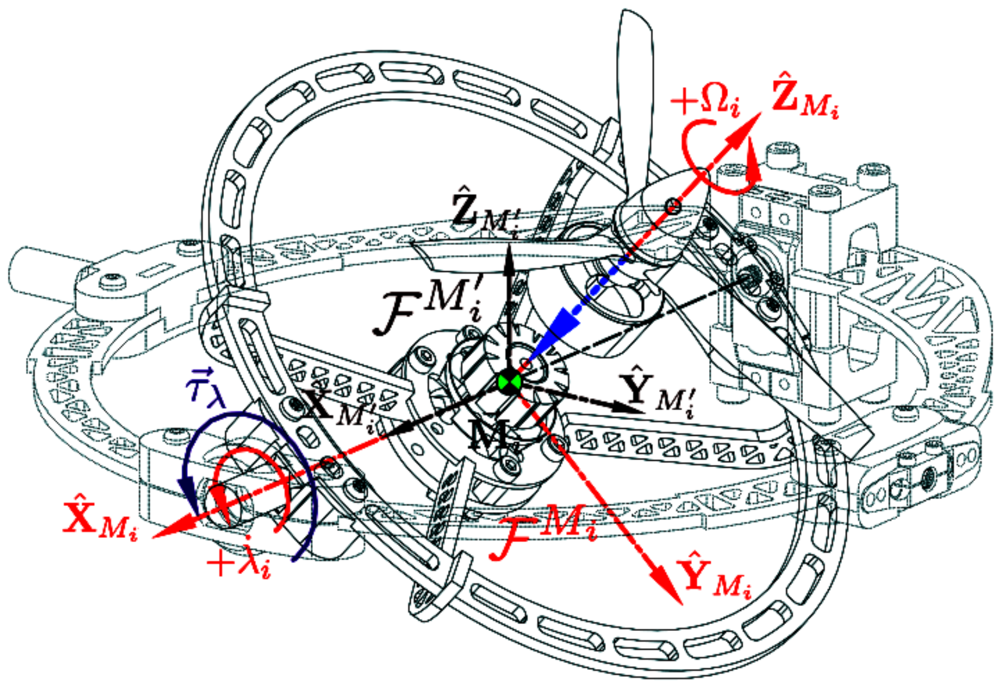


Figure 3.11: Exploded inner ring inertial bodies for $\vec{\tau}_\lambda(\lambda_i)$

Deriving dynamic responses for changes in the λ_i servo, acting on the inner ring frame $\mathcal{F}^{M_i}$ relative to the middle ring frame $\mathcal{F}^{M_i'}$, requires a relative path coordinate to be defined. Seeing that the only path variable between the two frames is that servo's rotational position λ_i about the $\hat{X}_{M_i}$ axis; the path coordinates $\vec{\mathbf{u}}(t) = \begin{bmatrix}\lambda_i & 0 & 0\end{bmatrix}^T$ are used to produce the Lagrangian for the inner ring's energy relative the middle ring frame, $\mathcal{L}_{n/m} \in \mathcal{F}^{M_i'}$.

The inner ring assembly consists of two separate bodies, exploded in Fig:3.11. Each with relative rotational motion and independent kinetic energies. Those bodies are; the rotor assembly with an inertia J_r defined earlier in Eq:2.19 and the inner ring which has an inertia J_{ir} *without including* the rotor assembly.

$$J_{ir} \triangleq J_n - J_r \tag{3.62}$$

Where J_n is the net inertia for the inner ring assembly, explicitly defined in Eq:2.21. The rotor assembly has an angular velocity $\vec{\omega}_{r/m}$ relative to the middle ring frame $\mathcal{F}^{M_i'}$ due to the BLDC motor's rotation Ω_i and the inner ring's servo rate $\dot{\lambda}_i$:

$$\vec{\omega}_{r/m} \triangleq R_x(\lambda)\vec{\Omega}_i + \frac{d\lambda}{dt}(\vec{\lambda}_i) \qquad \in \mathcal{F}^{M_i'} \tag{3.63a}$$

$$= R_x(\lambda)\vec{\Omega}_i + \dot{\vec{\lambda}}_i \tag{3.63b}$$

With the propeller's angular velocity vector; $\vec{\Omega}_i = \begin{bmatrix}0 & 0 & \Omega_i\end{bmatrix}^T \in \mathcal{F}^{M_i}$ and measured in rad.s^{-1} not in *revolutions per second*. The servo position is defined as a vector in the $\hat{X}_{M_i'}$ axis projected as $\vec{\lambda}_i \triangleq \lambda_i \cdot \hat{X}_{M_i'} = \begin{bmatrix}\lambda_i & 0 & 0\end{bmatrix}^T$ measured in rad. Next, the inner ring's angular velocity $\vec{\omega}_{n/m}$ relative to the middle ring $\mathcal{F}^{M_i'}$ is only as a result of $\dot{\lambda}_i$:

$$\vec{\omega}_{n/m} \triangleq \frac{d\lambda}{dt}(\vec{\lambda}_i) = \dot{\vec{\lambda}}_i \qquad \in \mathcal{F}^{M_i'} \tag{3.64}$$

The Lagrangian for the inner ring assembly's energy $\mathcal{L}_{n/m}$, relative to the middle ring frame $\mathcal{F}^{M_i'}$, consists purely of rotational kinetic energy from angular velocities described in Eq:3.63 and Eq:3.64. The relative gravitation potential energy as a result of the rotated center of mass for the inner ring is neglected here as it is already included in Eq:2.35d and is shown to simplify out subsequently in Eq:3.108 when considering the entire system as a whole. The inner ring Lagrangian is:

$$\mathcal{L}_{n/m} = \frac{1}{2}\vec{\omega}_{r/m}{}^{T}\big(J_r'\big)\vec{\omega}_{r/m} + \frac{1}{2}\vec{\omega}_{n/m}{}^{T}\big(J_{ir}'\big)\vec{\omega}_{n/m} \tag{3.65a}$$

Both inertias for the rotor and inner ring bodies, J_r and J_{ir} respectively, are transformed to align with the middle ring frame $\mathcal{F}^{M_i'}$ using an $R_x(\lambda)$ rotation to align with the middle ring's frame $\mathcal{F}^{M_i'}$.

$$J_r' = R_x(\lambda)\big(J_r\big)R_x^{-1}(\lambda) \quad \text{and} \quad J_{ir}' = R_x(\lambda)\big(J_{ir}\big)R_x^{-1}(\lambda) \tag{3.65b}$$

Then expanding the Lagrangian $\mathcal{L}_{n/m}$ in Eq:3.65a with the above definitions for transformed inertias and relative angular velocities $\vec{\omega}_{r/m}$ and $\vec{\omega}_{n/m}$ yields:

$$\begin{aligned} \rightarrow \mathcal{L}_{n/m} = \frac{1}{2}\Big(R_x(\lambda)\vec{\Omega}_i + \dot{\vec{\lambda}}_i\Big)^{T}\big(R_x(\lambda)\big(J_r\big)R_x^{-1}(\lambda)\big)\Big(R_x(\lambda)\vec{\Omega}_i + \dot{\vec{\lambda}}_i\Big) \\ + \frac{1}{2}\dot{\vec{\lambda}}_i^{T}\big(R_x(\lambda)\big(J_{ir}\big)R_x^{-1}(\lambda)\big)\dot{\vec{\lambda}}_i \end{aligned} \tag{3.65c}$$

Again reiterating J_{ir} is the inner ring's inertia independent from the rotor assembly J_r. Recalling the Euler-Lagrange formulation from Eq:3.3 using path coordinates $\vec{\mathbf{u}}(t)$ for the inner ring frame $\mathcal{F}^{M_i}$ relative to the middle ring frame $\mathcal{F}^{M_i'}$. The generalized (torque) forces $\vec{\mathbf{U}}$ acting on the middle ring are then:

$$\vec{\mathbf{U}}(\lambda_i) = \frac{d}{dt}\left(\frac{\partial \mathcal{L}_{n/m}}{\partial \dot{\vec{\mathbf{u}}}}\right) - \frac{\partial \mathcal{L}_{n/m}}{\partial \vec{\mathbf{u}}} \qquad \in \mathcal{F}^{M_i'} \tag{3.66}$$

From [10] the partial derivative of a rotation matrix $R_x(\lambda)$, and by extension the *transformation matrix* $R_x(-\lambda)$, is linearized using a Taylor series expansion. It follows that for some small perturbation $\partial\theta$ away from the nominal angle $\bar{\theta}$, a generalized rotation matrix about an axis $\hat{u}$ by that angle θ becomes a first order approximation:

$$R_u(\bar{\theta} + \partial\theta) \approx \underbrace{\big(1 - [\Phi_u(\bar{\theta})\partial\theta]_\times\big)}_{\text{infinitesimal rot}} R_u(\bar{\theta}) \tag{3.67}$$

Where $\Phi_u(\bar{\theta})$ is a generalized Euler matrix derivative from Eq:2.12i. The consequence of Eq:3.67 is that transformed rotational inertias in Eq:3.65, both $R_x(\lambda)\big(J_r\big)R_x^{-1}(\lambda)$ and $R_x(\lambda)\big(J_{ir}\big)R_x^{-1}(\lambda)$ can be approximated using their instantaneous transformation with no partial derivatives with respect to the relative path coordinate $\vec{\mathbf{u}}$.

$$R_x(\lambda)\big(J_r\big)R_x^{-1}(\lambda) = J_r' \rightarrow \frac{\partial}{\partial \vec{\mathbf{u}}}J_r' = \frac{\partial}{\partial \vec{\lambda}_i}J_r' \approx 0 \tag{3.68a}$$

$$R_x(\lambda)\big(J_{ir}\big)R_x^{-1}(\lambda) = J_{ir}' \rightarrow \frac{\partial}{\partial \vec{\mathbf{u}}}J_{ir}' = \frac{\partial}{\partial \vec{\lambda}_i}J_{ir}' \approx 0 \tag{3.68b}$$

Simplifications in Eq:3.68 are expanded upon and shown to be reasonable assumptions next in Sec:3.4.2. It follows that partial derivatives of the Lagrangian in Eq:3.65 with respect to $\vec{\mathbf{u}}$ are negligible; or that $\partial\mathcal{L}_{n/m}/\partial\vec{\mathbf{u}} \approx 0$. Only the partial derivatives with respect to the path rate $\dot{\vec{\mathbf{u}}}$ remain:

$$\therefore \vec{\mathbf{U}}(\lambda_i) \approx \frac{d}{dt}\left(\frac{\partial \mathcal{L}_{n/m}}{\partial \dot{\vec{\mathbf{u}}}}\right) = \frac{d}{dt}\Big(\big(J_r'\big)\Big(R_x(\lambda)\vec{\Omega}_i + \dot{\vec{\lambda}}_i\Big) + \big(J_{ir}'\big)\dot{\vec{\lambda}}_i\Big) \tag{3.69}$$

Transformed inertial rates $\dot{J}_r'$ and $\dot{J}_{ir}'$ must first be defined before evaluating the reduced Lagrangian derivative in Eq:3.69. Those inertial derivatives cannot be separated by time scale from the remainder of Eq:3.69 given that $\dot{\lambda}_i$ determines both inertial rates $\dot{J}_r'$ and $\dot{J}_{ir}'$ but is also a component of the kinetic energy in Eq:3.65c.

Starting with the general case; for some transformed inertia J to be aligned relative to a frame $\mathcal{F}^b$ where the inertia is originally defined with respect to a frame $\mathcal{F}^a$. If the two frames differ by some rotation angle θ about an Euler axis $\hat{u}$, the generalized rotation matrix from frame $\mathcal{F}^a$ to $\mathcal{F}^b$ is given by $R_{\hat{u}}(\theta)$ from Eq:2.7. The transformed inertia is then calculated as:

$$J' = R_{\hat{u}}(\theta)\big(J\big)R_{\hat{u}}^{-1}(\theta) \tag{3.70a}$$

Which, from the product rule and the rotation matrix time derivative definition previously in Eq:2.8, has its inertial rate as a result of the changing angle $\dot{\theta}$:

$$\dot{J}' = \frac{d}{dt}\Big(R_{\hat{u}}(\theta)\big(J\big)R_{\hat{u}}^{-1}(\theta)\Big) \tag{3.70b}$$

$$= \frac{d}{dt}\Big(R_{\hat{u}}(\theta)\Big)\big(J\big)R_{\hat{u}}^{-1}(\theta) + R_{\hat{u}}(\theta)\Big(\frac{d}{dt}\big(J\big)\Big)R_{\hat{u}}^{-1}(\theta) + R_{\hat{u}}(\theta)\big(J\big)\frac{d}{dt}\Big(R_{\hat{u}}^{-1}(\theta)\Big) \tag{3.70c}$$

$$= [\dot{\vec{\theta}}]_\times R_{\hat{u}}(\theta)\big(J\big)R_{\hat{u}}^{-1}(\theta) + R_{\hat{u}}(\theta)\big(\dot{J}\big)R_{\hat{u}}^{-1}(\theta) - R_{\hat{u}}(\theta)\big(J\big)[\dot{\vec{\theta}}]_\times R_{\hat{u}}^{-1}(\theta) \tag{3.70d}$$

Where $\dot{\vec{\theta}} \triangleq \dot{\theta} \cdot \hat{u}$ is the projcted angular velocity vector between the two frames. In most cases, the inertia will not be changing in its principle frame, or rather that $\dot{J} = 0$. Both the rotor assembly and inner ring inertias are constant in their principle frames. The transformed inertias then have the following derivatives; first for the rotor assembly:

$$\therefore \dot{J}_r' = \frac{d}{dt}\Big(R_x(\lambda)\big(J_r\big)R_x^{-1}(\lambda)\Big) \tag{3.71a}$$

$$= [\dot{\vec{\lambda}}_i]_\times R_x(\lambda)\big(J_r\big)R_x^{-1}(\lambda) - R_x(\lambda)\big(J_r\big)[\dot{\vec{\lambda}}_i]_\times R_x^{-1}(\lambda) \tag{3.71b}$$

Similarly for the inner ring's transformed inertial rate $\dot{J}_{ir}'$, again without the rotor's contribution:

$$\therefore \dot{J}_{ir}' = \frac{d}{dt}\Big(R_x(\lambda)\big(J_{ir}\big)R_x^{-1}(\lambda)\Big) \tag{3.72a}$$

$$= [\dot{\vec{\lambda}}_i]_\times R_x(\lambda)\big(J_{ir}\big)R_x^{-1}(\lambda) - R_x(\lambda)\big(J_{ir}\big)[\dot{\vec{\lambda}}_i]_\times R_x^{-1}(\lambda) \tag{3.72b}$$

Substituting those transformed inertial rates into Eq:3.69 and using Reynolds transportation theorem, Eq:3.5 for a vector's derivative in a rotating reference frame, the product rule then yields:

$$\rightarrow \frac{d}{dt}\bigg(\frac{\partial \mathcal{L}_{n/m}}{\partial \dot{\mathbf{\vec{u}}}}\bigg) = \Big[\big(\dot{J}_r'\big)\big(R_x(\lambda)\vec{\Omega}_i + \dot{\vec{\lambda}}_i\big) + \big(J_r'\big)R_x(\lambda)\dot{\vec{\Omega}}_i + \vec{\omega}_{r/m} \times \big(J_r'\big)R_x(\lambda)\vec{\Omega}_i + \big(J_r'\big)\ddot{\vec{\lambda}}_i + \vec{\omega}_{r/m} \times \big(J_r'\big)\dot{\vec{\lambda}}_i\Big] + \Big[\big(\dot{J}_{ir}'\big)\dot{\vec{\lambda}}_i + \big(J_{ir}'\big)\ddot{\vec{\lambda}}_i + \vec{\omega}_{n/m} \times \big(J_{ir}'\big)\dot{\vec{\lambda}}_i\Big] = \vec{\mathbf{U}}(\lambda_i) \tag{3.73}$$

Recombining inertial bodies with the same angular velocity ($J_r' + J_{ir}' = J_n'$) and recognizing that, from Eq:3.64, $\vec{\omega}_{n/m} = \dot{\vec{\lambda}}_i$ the generalized net torque encountered by a $\Delta\lambda_i$ rotation is:

$$\therefore \vec{\mathbf{U}}(\lambda_i) = \big(\dot{J}_r'\big)\vec{\Omega}_i' + \big(J_r'\big)\dot{\vec{\Omega}}_i' + \dot{\vec{\lambda}}_i \times \big(J_r'\big)\vec{\Omega}_i' + \big(\dot{J}_n'\big)\dot{\vec{\lambda}}_i + \big(J_n'\big)\ddot{\vec{\lambda}}_i + \dot{\vec{\lambda}}_i \times \big(J_n'\big)\dot{\vec{\lambda}}_i \qquad \in \mathcal{F}^{M_i'} \tag{3.74a}$$

Where both $\vec{\Omega}_i'$ and $\dot{\vec{\Omega}}_i'$ are the respective transformed rotational velocity and acceleration of the propeller in the middle ring frame:

$$\vec{\Omega}_i' \triangleq R_x(\lambda)\vec{\Omega}_i \qquad \in \mathcal{F}^{M_i'} \tag{3.74b}$$

$$\dot{\vec{\Omega}}_i' \triangleq \frac{d\vec{\Omega}}{dt}\big(R_x(\lambda)\vec{\Omega}_i\big) = R_x(\lambda)\dot{\vec{\Omega}}_i \qquad \in \mathcal{F}^{M_i'} \tag{3.74c}$$

The net torque response, $\vec{\tau}_\lambda(\lambda_i)$ from a $\Delta\lambda_i$ rotation, induced in the middle ring frame $\mathcal{F}^{M_i'}$, can be grouped into *inertial rates*, second order *inertial* and first order *gyroscopic* components;

$$\vec{\tau}_\lambda(\lambda_i) = \underbrace{\big(\dot{J}_r'\big)\vec{\Omega}_i' + \big(\dot{J}_n'\big)\dot{\vec{\lambda}}_i}_{\textit{Inertial rates}} + \underbrace{\big(J_r'\big)\dot{\vec{\Omega}}_i' + \big(J_n'\big)\ddot{\vec{\lambda}}_i}_{\textit{Inertial}} + \underbrace{\dot{\vec{\lambda}}_i \times \big(J_r'\big)\vec{\Omega}_i' + \dot{\vec{\lambda}}_i \times \big(J_n'\big)\dot{\vec{\lambda}}_i}_{\textit{Gyroscopic}} \qquad \in \mathcal{F}^{M_i'} \tag{3.75}$$

That equation represents the true torque response $\vec{\tau}_\lambda(\lambda_i)$, later in control design $\hat{\tau}_\lambda(\lambda_i)$ is used for feedback compensation. That torque $\hat{\tau}_\lambda(\lambda_i)$ is a *modelled estimate* derived from state dynamics and could potentially contain modelling or estimation errors. Moreover Eq:3.75 assumes instantaneous arguments for the actuator positions λ_i when in practice state estimates for $\hat{\lambda}_i$ are used.

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 $\vec{\mathbf{v}}(t) = \begin{bmatrix} 0 & \alpha_i & 0 \end{bmatrix}^T$. The entire motor module's structure consists of three separate rotating bodies each with their own relative angular velocities; the *rotor* assembly, *inner* and *middle* ring structures; exploded in Fig:3.12.

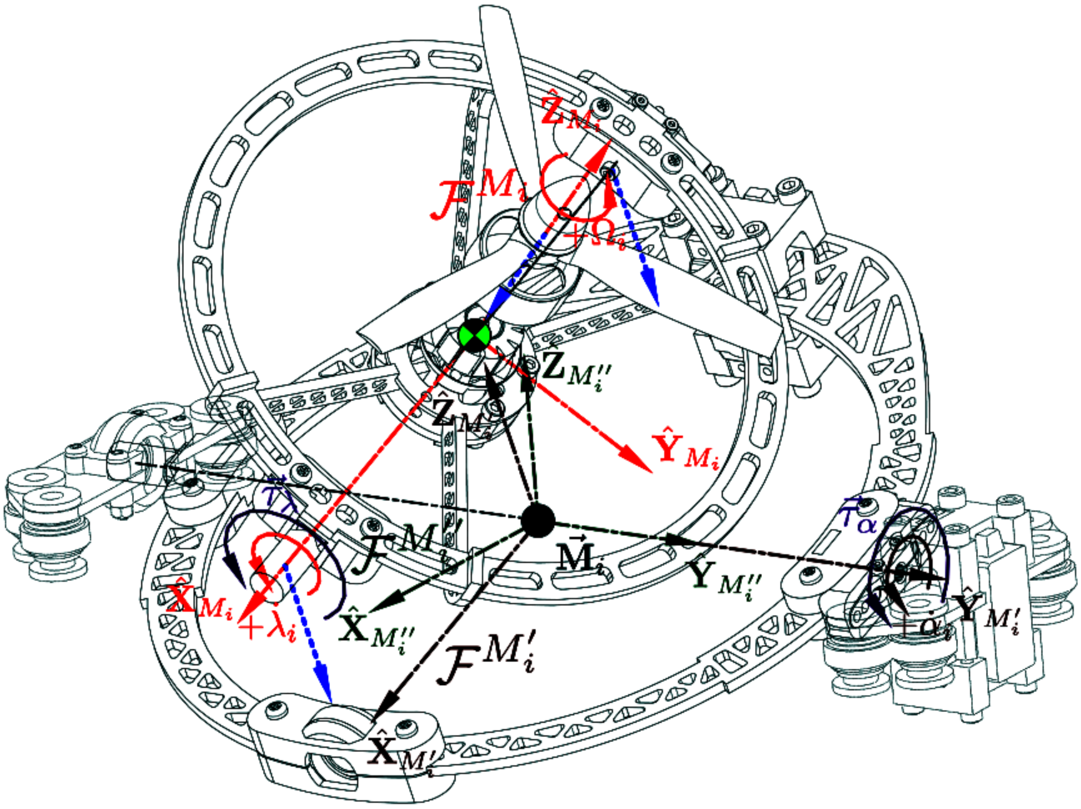


Figure 3.12: Exploded middle ring inertial bodies for $\vec{\tau}_\alpha(\lambda_i, \alpha_i)$

Applying the same process to evaluate the α_i servo's response, the middle ring assembly Lagrangian $\mathcal{L}_{m/p}$ is constructed but with respect to the intermediary frame $\mathcal{F}^{M_i''}$. First transforming the inertias; the rotor assembly, further rotated by α_i about its $\hat{Y}_{M_i'}$ axis, has an inertia aligned with axes in $\mathcal{F}^{M_i''}$:

$$J_r'' = R_y(\alpha)\big(J_r'\big)R_y^{-1}(\alpha) = R_y(\alpha)R_x(\lambda)\big(J_r\big)R_x^{-1}(\lambda)R_y^{-1}(\alpha) \tag{3.76a}$$

Which has a derivative $\dot{J}_r''$:

$$\dot{J}_r'' = R_y(\alpha)\big(\dot{J}_r'\big)R_y^{-1}(\alpha) + [\dot{\vec{\alpha}}_i]_\times R_y(\alpha)\big(J_r'\big)R_y^{-1}(\alpha) - R_y(\alpha)\big(J_r'\big)[\dot{\vec{\alpha}}_i]_\times R_y^{-1}(\alpha) \tag{3.76b}$$

The inner ring structure has an inertia, still *without* including the rotor assembly, aligned with $\mathcal{F}^{M_i''}$:

$$J_{ir}'' = R_y(\alpha)\big(J_{ir}'\big)R_y^{-1}(\alpha) = R_y(\alpha)R_x(\lambda)\big(J_{ir}\big)R_x^{-1}(\lambda)R_y^{-1}(\alpha) \tag{3.77a}$$

Similarly with a derivative $\dot{J}_{ir}''$:

$$\dot{J}_{ir}'' = R_y(\alpha)\big(\dot{J}_{ir}'\big)R_y^{-1}(\alpha) + [\dot{\vec{\alpha}}]_\times R_y(\alpha)\big(J_{ir}'\big)R_y^{-1}(\alpha) - R_y(\alpha)\big(J_{ir}'\big)[\dot{\vec{\alpha}}]_\times R_y^{-1}(\alpha) \tag{3.77b}$$

Finally the middle ring structure's inertia from Eq:2.24a, with neither the rotor's nor the inner ring's contributions:

$$J_m' = R_y(\alpha)\big(J_\mathrm{m}\big)R_y^{-1}(\alpha) \tag{3.78a}$$

Which, when using the collective motor module inertia J_p from Eq:2.24b, expands to:

$$= R_y(\alpha)\big(J_{\mathrm{p}}\big)R_y^{-1}(\alpha) - R_y(\alpha)R_x(\lambda)\big(J_{\mathrm{n}}\big)R_x^{-1}(\lambda)R_y^{-1}(\alpha) \tag{3.78b}$$

$$= J'_{\mathrm{p}} - J''_n = J'_{\mathrm{p}} - \big(J''_{ir} + J''_r\big) \tag{3.78c}$$

Which has a derivative purely as a result of $\dot{\alpha}$:

$$\dot{J}'_m = [\dot{\vec{\alpha}}_i]_\times R_y(\alpha)\big(J_m\big)R_y^{-1}(\alpha) - R_y(\alpha)\big(J_m\big)[\dot{\vec{\alpha}}_i]_\times R_y^{-1}(\alpha) \tag{3.78d}$$

However; that derivative $\dot{J}'_m$, using $\dot{J}''_r$ and $\dot{J}''_{ir}$ from Eq:3.76b and Eq:3.77b respectively, expands to:

$$\dot{J}'_m = [\dot{\vec{\alpha}}_i]_\times R_y(\alpha)\big(J_p\big)R_y^{-1}(\alpha) - R_y(\alpha)\big(J_p\big)[\dot{\vec{\alpha}}_i]_\times R_y^{-1}(\alpha) - (\dot{J}''_{ir} + \dot{J}''_r) \tag{3.78e}$$

Note that introducing the relations of Eq:3.78c and Eq:3.78e to the collective body inertia J_p is to simplify the subsequent equations. Each body then has its own relative angular velocity with respect to the intermediate frame $\mathcal{F}^{M''_i}$. For the rotor; $\vec{\omega}_{r/p}$ is the relative angular velocity of that assembly from the motor Ω_i and both inner and middle servo rates $\dot{\lambda}_i$ and $\dot{\alpha}_i$:

$$\vec{\omega}_{r/p} \triangleq R_y(\alpha)R_x(\lambda)\vec{\Omega}_i + \frac{d\lambda}{dt}\Big(R_y(\alpha)\vec{\lambda}_i\Big) + \frac{d\alpha}{dt}(\vec{\alpha}_i) \qquad \in \mathcal{F}^{M''_i} \tag{3.79a}$$

$$= R_y(\alpha)R_x(\lambda)\vec{\Omega}_i + R_y(\alpha)\dot{\vec{\lambda}}_i + \dot{\vec{\alpha}}_i \tag{3.79b}$$

$$\therefore \vec{\omega}_{r/p} = \vec{\Omega}''_i + \dot{\vec{\lambda}}'_i + \dot{\vec{\alpha}}_i \tag{3.79c}$$

Where $\vec{\Omega}''_i$ and $\dot{\vec{\lambda}}'_i$ are respectively propeller and inner servo velocities transformed to the frame $\mathcal{F}^{M''_i}$. Next, the inner ring has an angular velocity $\vec{\omega}_{n/p}$ relative to the intermediate frame $\mathcal{F}^{M''_i}$ from the two servo rates $\dot{\lambda}_i$ and $\dot{\alpha}_i$:

$$\vec{\omega}_{n/p} \triangleq \frac{d\lambda}{dt}\big(R_y(\alpha)\vec{\lambda}_i\big) + \frac{d\alpha}{dt}(\vec{\alpha}_i) \qquad \in \mathcal{F}^{M''_i} \tag{3.80a}$$

$$= R_y(\alpha)\dot{\vec{\lambda}}_i + \dot{\vec{\alpha}}_i = \dot{\vec{\lambda}}'_i + \dot{\vec{\alpha}}_i \tag{3.80b}$$

Lastly the middle ring body has an angular velocity $\vec{\omega}_{m/p}$ relative to the intermediary frame only as a result of the middle ring's servo velocity $\dot{\alpha}_i$:

$$\vec{\omega}_{m/p} \triangleq \frac{d\alpha}{dt}(\vec{\alpha}_i) = \dot{\vec{\alpha}}_i \qquad \in \mathcal{F}^{M''_i} \tag{3.81}$$

Using the relative path coordinate $\vec{\mathbf{v}}(t)$, the Lagrangian $\mathcal{L}_{m/p}$ can be constructed for the complete motor module relative to the intermediate frame $\mathcal{F}^{M''_i}$ with kinetic energies of the rotor assembly, inner and middle ring structures respectively:

$$\mathcal{L}_{m/p} = \frac{1}{2}\vec{\omega}^T_{r/p}\big(J''_r\big)\vec{\omega}_{r/p} + \frac{1}{2}\vec{\omega}^T_{n/p}\big(J''_{ir}\big)\vec{\omega}_{n/p} + \frac{1}{2}\vec{\omega}^T_{m/p}\big(J'_m\big)\vec{\omega}_{m/p} \tag{3.82}$$

Where Eq:3.82 again does not include any potential energy gravitational contributions because such quantities are incorporated in Eq:2.35d. The middle ring's relative Lagrangian $\mathcal{L}_{m/p}$ from Eq:3.82 therefore expands to:

$$\begin{aligned}\therefore \mathcal{L}_{m/p} = \frac{1}{2}\Big[R_y(\alpha)R_x(\lambda)\vec{\Omega}_i + R_y(\alpha)\dot{\vec{\lambda}}_i + \dot{\vec{\alpha}}\Big]^T\big(J''_r\big)\Big[R_y(\alpha)R_x(\lambda)\vec{\Omega}_i + Ry(\alpha)\dot{\vec{\lambda}}_i + \dot{\vec{\alpha}}_i\Big]\\ + \frac{1}{2}\Big[R_y(\alpha)\dot{\vec{\lambda}}_i + \dot{\vec{\alpha}}_i\Big]^T\big(J''_{ir}\big)\Big[R_y(\alpha)\dot{\vec{\lambda}}_i + \dot{\vec{\alpha}}_i\Big] + \frac{1}{2}\dot{\vec{\alpha}}_i^T\big(J'_m\big)\dot{\vec{\alpha}}_i\end{aligned} \tag{3.83}$$

Again, justifying the rotation matrix linearization using Eq:3.67; matrices J''_r, J''_{ir} and J'_m are all instantaneous transformed inertias. The Euler-Lagrange formulation then simplifies with the partial derivative $\partial\mathcal{L}_{m/p}/\partial\vec{\mathbf{v}} \approx 0$. So the generalized forces $\vec{\mathbf{V}}(\lambda_i, \alpha_i)$ are:

$$\vec{\mathbf{V}}(\lambda_i, \alpha_i) = \frac{d}{dt}\left(\frac{\partial\mathcal{L}_{m/p}}{\partial\dot{\vec{\mathbf{v}}}}\right) - \frac{\partial\mathcal{L}_{m/p}}{\partial\vec{\mathbf{v}}} \approx \frac{d}{dt}\left(\frac{\partial\mathcal{L}_{m/p}}{\partial\dot{\vec{\mathbf{v}}}}\right) \qquad \in \mathcal{F}^{M''_i} \tag{3.84}$$

Finding the partial derivative of $\mathcal{L}_{m/p}$ in Eq:3.83 with respect to the middle ring's servo relative path coordinate rate $\dot{\vec{\mathbf{v}}}$ yields:

$$\frac{\partial \mathcal{L}_{m/p}}{\partial \dot{\vec{\mathbf{v}}}} = \big(J_r''\big)\Big[\vec{\Omega}_i'' + \dot{\vec{\lambda}}_i' + \dot{\vec{\alpha}}_i\Big] + \big(J_{ir}''\big)\Big[\dot{\vec{\lambda}}_i' + \dot{\vec{\alpha}}_i\Big] + \big(J_m'\big)\dot{\vec{\alpha}}_i \tag{3.85a}$$

Which with relative rotor, inner and middle ring angular velocity definitions from Eq:3.79,3.80 and 3.81 respectively; expands to:

$$= \big(J_r''\big)\Big[R_y(\alpha)R_x(\lambda)\vec{\Omega}_i + R_y(\alpha)\dot{\vec{\lambda}}_i + \dot{\vec{\alpha}}_i\Big] + \big(J_{ir}''\big)\Big[R_y(\alpha)\dot{\vec{\lambda}}_i + \dot{\vec{\alpha}}_i\Big] + \big(J_m'\big)\dot{\vec{\alpha}}_i \tag{3.85b}$$

Then taking the time derivative of that partial derivative and using inertial rates for each body defined in Eq:3.76b,3.77b and 3.78e; split into product ruled derivative components:

$$\begin{aligned} \rightarrow \vec{\mathbf{V}}(\lambda_i, \alpha_i) = \frac{d}{dt}\left(\frac{\partial \mathcal{L}_{m/p}}{\partial \dot{\vec{\mathbf{v}}}}\right) &= \Big[\big(\dot{J}_r''\big)\big(\vec{\Omega}_i'' + \dot{\vec{\lambda}}_i' + \dot{\vec{\alpha}}_i\big)\Big] \\ + \Big[\big(J_r''\big)\dot{\vec{\Omega}}_i'' + \vec{\omega}_{n/p} \times \big(J_r''\big)\vec{\Omega}_i'' &+ \big(J_r''\big)\ddot{\vec{\lambda}}_i' + \vec{\omega}_{n/p} \times \big(J_r''\big)\dot{\vec{\lambda}}_i' + \big(J_r''\big)\ddot{\vec{\alpha}}_i + \vec{\omega}_{m/p} \times \big(J_r''\big)\dot{\vec{\alpha}}_i\Big] \\ + \Big[\big(\dot{J}_{ir}''\big)\big(\dot{\vec{\lambda}}_i' + \dot{\vec{\alpha}}_i\big)\Big] &+ \Big[\big(J_{ir}''\big)\ddot{\vec{\lambda}}_i' + \vec{\omega}_{n/p} \times \big(J_{ir}''\big)\dot{\vec{\lambda}}_i' + \big(J_{ir}''\big)\ddot{\vec{\alpha}}_i + \vec{\omega}_{m/p} \times \big(J_{ir}''\big)\dot{\vec{\alpha}}_i\Big] \\ &+ \Big[\big(\dot{J}_m'\big)\dot{\vec{\alpha}}_i\Big] + \Big[\big(J_m'\big)\ddot{\vec{\alpha}}_i + \vec{\omega}_{m/p} \times \big(J_m'\big)\dot{\vec{\alpha}}_i\Big] \end{aligned} \tag{3.85c}$$

With relative frame angular velocities; $\vec{\omega}_{n/p}$ of the inner ring relative to the intermediate frame, and $\vec{\omega}_{m/p}$ of the middle ring relative to the intermediate frame. Both are defined respectively:

$$\vec{\omega}_{n/p} \triangleq R_y(\alpha)\dot{\vec{\lambda}}_i + \dot{\vec{\alpha}}_i = \dot{\vec{\lambda}}_i' + \dot{\vec{\alpha}}_i \qquad \in \mathcal{F}^{M_i''} \tag{3.85d}$$

$$\vec{\omega}_{m/p} \triangleq \dot{\vec{\alpha}}_i \qquad \in \mathcal{F}^{M_i''} \tag{3.85e}$$

Eq:3.85c is an ominous and decidedly complicated result to try expand and make sense of. However it can be simplified; recognizing that generalized torques in Eq:3.85c contain kinetic energies already introduced in Eq:3.75, but transformed to the frame $\mathcal{F}^{M_i''}$. After some mathematics, Eq:3.85c can be simplified with responses pertinent to $\Delta\alpha_i$ and then the transformed generalized force response $R_y(\alpha)\vec{\tau}_\lambda(\lambda_i)$:

$$\begin{aligned} \vec{\mathbf{V}}(\lambda_i, \alpha_i) = R_y(\alpha)\frac{d}{dt}\left(\frac{\partial \mathcal{L}_{n/m}}{\partial \dot{\vec{\mathbf{u}}}}\right) + \Big(R_y(\alpha)\big(\dot{J}_r'\big)R_y^{-1}(\alpha)\Big)\dot{\vec{\alpha}} + \Big(\dot{J}_r'' - R_y(\alpha)\big(\dot{J}_r'\big)R_y^{-1}(\alpha)\Big)\big(\vec{\Omega}_i'' + \dot{\vec{\lambda}}_i' + \dot{\vec{\alpha}}_i\big) \\ + \big(J_r''\big)\ddot{\vec{\alpha}}_i + \dot{\vec{\alpha}}_i \times \big(J_r''\big)\Big(\vec{\Omega}_i'' + \dot{\vec{\lambda}}_i' + \dot{\vec{\alpha}}_i\Big) + \Big(R_y(\alpha)\big(\dot{J}_{ir}'\big)R_y^{-1}(\alpha)\Big)\dot{\vec{\alpha}} + \Big(\dot{J}_{ir}'' - R_y(\alpha)\big(\dot{J}_{ir}'\big)R_y^{-1}(\alpha)\Big)\big(\dot{\vec{\lambda}}_i' + \dot{\vec{\alpha}}_i\big) \\ + \big(J_{ir}''\big)\ddot{\vec{\alpha}}_i + \dot{\vec{\alpha}}_i \times \big(J_{ir}''\big)\Big(\dot{\vec{\lambda}}_i' + \dot{\vec{\alpha}}_i\Big) + \big(\dot{J}_m'\big)\dot{\vec{\alpha}}_i + \big(J_m'\big)\ddot{\vec{\alpha}}_i + \dot{\vec{\alpha}}_i \times \big(J_m'\big)\dot{\vec{\alpha}}_i \end{aligned} \tag{3.85f}$$

Paying special attention to differentiate $\dot{J}_r''$ and $\dot{J}_{ir}''$ from Eq:3.76b and Eq:3.77b respectively with $R_y(\alpha)\big(\dot{J}_r'\big)R_y^{-1}(\alpha)$ and $R_y(\alpha)\big(\dot{J}_{ir}'\big)R_y^{-1}(\alpha)$. Where the latter two terms are inertial rates from Eq:3.71 and Eq:3.72, but transformed to the frame $\mathcal{F}^{M_i''}$.

Generalized torques in Eq:3.85f can be further simplified by introducing combined inertial bodies $J_n = J_r + J_{ir}$ for the *entire* inner ring from Eq:2.21 and $J_p = J_m + R_x(\lambda)\big(J_n\big)R_x^{-1}(\lambda)$ for the *entire* motor module's inertia from Eq:2.24b. Using $J_p' = R_y(\alpha)\big(J_p\big)R_y^{-1}(\alpha)$ and $J_n' = R_y(\alpha)\big(J_n\big)R_y^{-1}(\alpha)$ for the net modules inertia and the entire inner ring inertia both respectively aligned with the frame $\mathcal{F}^{M_i''}$:

$$\begin{aligned} \rightarrow \vec{\mathbf{V}}(\lambda_i, \alpha_i) = R_y(\alpha)\vec{\mathbf{U}}(\lambda_i) + \Big(R_y(\alpha)\big(\dot{J}_n'\big)R_y(\alpha)\Big)\dot{\vec{\alpha}}_i + \Big(\dot{J}_p' - R_y(\alpha)\big(\dot{J}_p\big)R_y^{-1}(\alpha)\Big)\dot{\vec{\alpha}}_i \\ + \Big(\dot{J}_n'' - R_y(\alpha)\big(\dot{J}_n'\big)R_y^{-1}(\alpha)\Big)\dot{\vec{\lambda}}' + \Big(\dot{J}_r'' - R_y(\alpha)\big(\dot{J}_r'\big)R_y^{-1}(\alpha)\Big)\vec{\Omega}_i'' \\ + J_p'\ddot{\vec{\alpha}}_i + \dot{\vec{\alpha}}_i \times \Big(\big(J_p'\big)\dot{\vec{\alpha}} + \big(J_n''\big)\dot{\vec{\lambda}}_i' + \big(J_r''\big)\vec{\Omega}_i''\Big) \end{aligned} \tag{3.85g}$$

Noting that $\dot{J}_p = \dot{J}'_r + \dot{J}'_{ir} + \dot{J}_m$ and that $\dot{J}_m = 0$, it follows that $\dot{J}_p = \dot{J}'_n$. Isolating the servo's torque response from $\Delta\alpha_i$, and again grouping inertial bodies with shared angular velocities together. The *inertial rates*, second order *inertial* and first order *gyroscopic* responses are then:

$$\vec{\tau}_\alpha(\lambda_i, \alpha_i) = \underbrace{\left(\dot{J}'_p\right)\dot{\vec{\alpha}}_i + \left(\dot{J}''_n - R_y(\alpha)\left(\dot{J}'_n\right)R_y^{-1}(\alpha)\right)\dot{\vec{\lambda}}'_i + \left(\dot{J}''_r - R_y(\alpha)\left(\dot{J}'_r\right)R_y^{-1}(\alpha)\right)\vec{\Omega}''_i}_{Inertial\ rates}$$
$$+ \underbrace{\left(J'_{\mathrm{p}}\right)\ddot{\vec{\alpha}}_i}_{Inertial} + \underbrace{\dot{\vec{\alpha}}_i \times \left(\left(J'_p\right)\dot{\vec{\alpha}}_i + \left(J''_n\right)\dot{\vec{\lambda}}'_i + \left(J''_r\right)\vec{\Omega}''_i\right)}_{Gyroscopic} \quad \in \mathcal{F}^{M''_i} \tag{3.86}$$

It is important to repeat that the servo's response $\vec{\tau}_\alpha(\lambda_i, \alpha_i)$ is ***not*** the same as the generalized torque $\vec{\mathbf{V}}(\lambda_i, \alpha_i)$ described in Eq:3.85g. The latter contains terms for the inner ring's servo response. Careful inspection could have yielded the inertial and gyroscopic components of both Eq:3.75 and Eq:3.86, however the effect of inertial rates on the torque system is a far less obvious result. The assumption in Eq:3.67 that rotated inertias can be linearized is shown to hold true next in Sec:3.4.2 where simulations and physical tests corroborate the above models.

Both servo's respective induced torques, $\vec{\tau}_\lambda(\lambda_i)$ and $\vec{\tau}_\alpha(\lambda_i, \alpha_i)$, occur in sequential gimbal-like frames. The opposing negative responses to induced relative rotations effect the angular state dynamics in Eq:3.10d, and must be transformed to the common body frame:

$$\vec{\tau}_Q(u_i) = -\sum_{i=1}^{4}\Big(R_z(\sigma_i)R_y(\alpha_i)\vec{\tau}_\lambda(\lambda_i) + R_z(\sigma_i)\vec{\tau}_\alpha(\alpha_i, \lambda_i)\Big) \quad \in \mathcal{F}^b \tag{3.87a}$$
$$= -\sum_{i=1}^{4} R_z(\sigma_i)\vec{\mathbf{V}}(\lambda_i, \alpha_i) \tag{3.87b}$$

The last non-trivial torque term associated with the multibody motion which must be accounted for is the entire system's response to motion relative to the inertial frame $\mathcal{F}^I$. Specifically considering the responses relative rotations $\Delta\lambda$ and $\Delta\alpha$ have to the net angular velocity of the entire multibody system $\vec{\omega}_b$. Such responses are an extension of the fundamental rigid 6-DOF differential equation for angular motion, reiterated from Eq:3.61:

$$\dot{\vec{\omega}}_b = \left(J_b^{-1}\right)\Big(-\vec{\omega}_b \times (J_b)\vec{\omega}_b + \vec{\tau}_\mu\Big) \quad \in \mathcal{F}^b \tag{3.88}$$

Before continuing with a Lagrangian formulation applied to the entire multibody vehicle; it is worth first establishing a Lemma to add some clarity to the steps which follow. Consider the hypothetical rotating, non-Newtonian 2-D system illustrated in Fig:3.13.

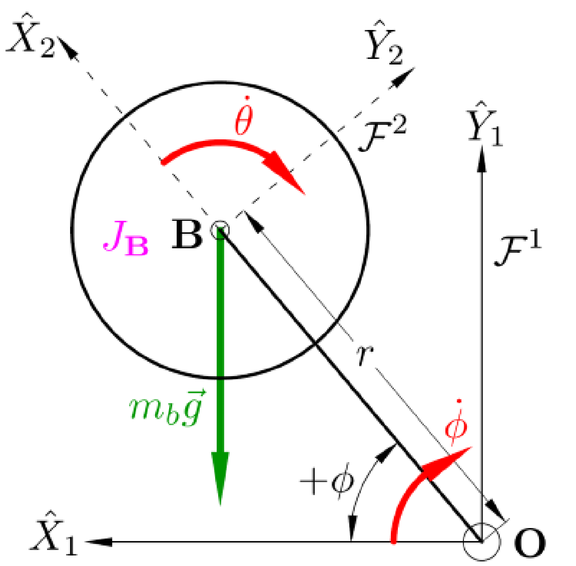
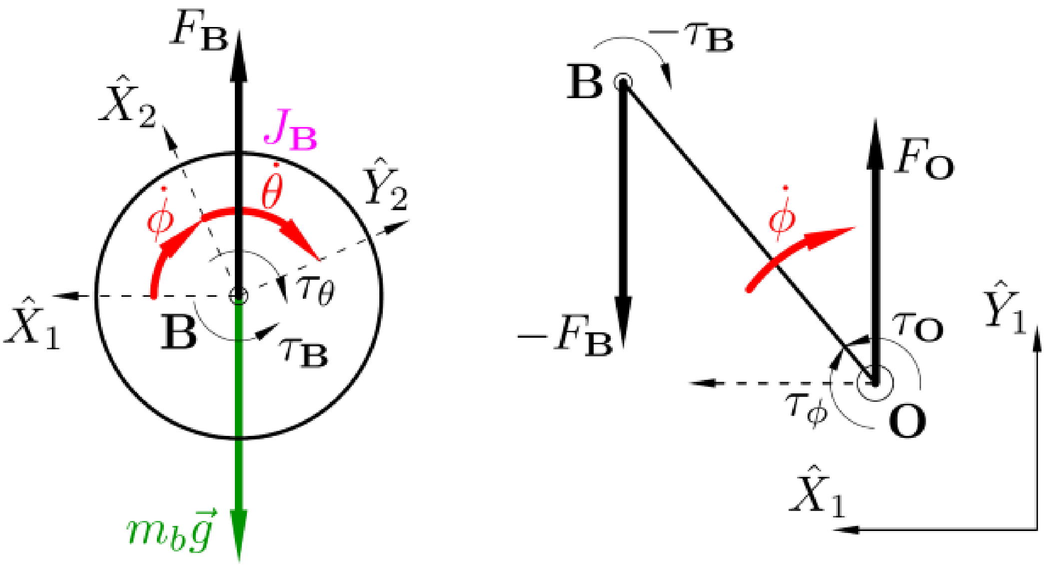


Figure 3.13: Rotating system

A massless rod of length r connects some rotational body, with a mass m_b, at point **B** to a center pivot point **O**. The principle frame $\mathcal{F}^1$ has axes $\hat{X}_1$ and $\hat{Y}_1$ as illustrated. The arm has a rotational velocity $\dot{\phi}$ relative to $\hat{X}_1$ in $\mathcal{F}^1$, applied by some "motor". Attached to the end of the rod is a secondary frame $\mathcal{F}^2$ with an $\hat{X}_2$ axis co-linear to the rod and a perpendicular $\hat{Y}_2$. The rotational body, centered at point **B**, has a rotational inertia $J_{\mathbf{B}}$ about the point (or axis) at **B**. That rotating body has a rotational velocity $\dot{\theta}$ from another "motor" relative to $\mathcal{F}^2$. The question is then how to find the net torque applied to the system about point **O** in terms of angular velocities $\dot{\phi}$ and $\dot{\theta}$ and their derivatives (or accelerations)?



(a) Rotational body **(b)** Massless rod

Figure 3.14: Free-body diagram for rotational system

Isolated free body diagrams for each body under consideration are illustrated in Fig:3.14. Considering the rotational body only, Fig:3.14a, the torque acting about point **B** is simply an inertial response to combined angular accelerations of $\ddot{\theta}$ and $\ddot{\alpha}$:

$$\tau_{\mathbf{B}} = -\tau_\theta = -J_{\mathbf{B}}(\ddot{\theta} + \ddot{\phi}) \qquad \in \mathcal{F}^2 \tag{3.89}$$

The net force acting on the rotational body is purely the gravitational force acting through point **B** as a result of the mass m_b and some gravitational force vector $\vec{g} \in \mathcal{F}^2$:

$$F_{\mathbf{B}} = -G = -m_b\vec{g} \qquad \in \mathcal{F}^2 \tag{3.90}$$

That torque and force pair, $F_{\mathbf{B}}$ and $\tau_{\mathbf{B}}$, are transferred to frame $\mathcal{F}^1$ through the massless rod connecting point **B** to **O**, Fig:3.14b. The net torque acting around point **O** is then comprised of three components; inferred torque from $\tau_{\mathbf{B}}$, a torque arm from force $F_{\mathbf{B}}$ and an inertial torque response to the effective "point-mass" at point **B** relative to **O**:

$$\tau_{\mathbf{O}} = -\tau_\phi = -\tau_{\mathbf{B}} - F_{\mathbf{B}} r\cos\phi + m_b r^2(\ddot{\phi}) \qquad \in \mathcal{F}^1 \tag{3.91}$$

The net response force acting at point **O**, $F_{\mathbf{O}}$, is of no consequence to the calculation of net torques. The "motor" applies a torque τ_ϕ to the rod to induce some angular acceleration $\ddot{\phi}$ on the whole system. Opposed to that angular acceleration is the torque $\tau_{\mathbf{O}}$ which acts against that rotation. The torque τ_ϕ acting on the system can then be simplified:

$$\tau_\phi = J_{\mathbf{B}}(\ddot{\theta} + \ddot{\phi}) + m_b r^2(\ddot{\phi}) - m_b\vec{g} r\cos\phi \qquad \in \mathcal{F}^1 \tag{3.92}$$

That result would not be as obvious when inferred from an energy equation. The equivalent Lagrangian for net kinetic and potential energy of the system, T and U respectively relative to $\mathcal{F}^1$, would be:

$$\mathcal{L} = T(\theta, \phi) - U(\theta, \phi) \tag{3.93a}$$

$$\mathcal{L} = \frac{1}{2}\vec{\omega}_{\mathbf{B}}^{T}(J_{\mathbf{B}})\vec{\omega}_{\mathbf{B}} + \frac{1}{2}\vec{\omega}_{\mathbf{O}}^{T}(J_{\mathbf{O}})\vec{\omega}_{\mathbf{O}} - m_b\vec{g}r\sin\phi \tag{3.93b}$$

Where $\vec{\omega}_{\mathbf{B}}$ and $\vec{\omega}_{\mathbf{O}}$ are net angular velocities of the rotational body and massless connection rod respectively. The important thing to consider is that $J_{\mathbf{O}}$, the net rotational inertia about the point $\mathbf{O}$, is simply the point mass inertia $m_b r^2$ and <u>*NOT*</u> the expected parallel axis theorem $J_{\mathbf{O}} \neq J'_{\mathbf{B}} = J_{\mathbf{B}} + m_b r^2$. Expanding Eq:3.93b and applying the Euler-Lagrange formulation, using a partial derivative with respect to the path coordinate ϕ to produce the generalized torque τ_ϕ acting on the system:

$$\rightarrow \mathcal{L} = \frac{1}{2}\left(\dot{\theta}+\dot{\phi}\right)^{T}(J_{\mathbf{B}})\left(\dot{\theta}+\dot{\phi}\right) + \left(\dot{\phi}\right)(m_b r^2)\left(\dot{\phi}\right) - m_b(-g)r\sin\phi \tag{3.93c}$$

$$\therefore \text{Genealized forces} = \frac{d}{dt}\left(\frac{\partial\mathcal{L}}{\partial\dot{\phi}}\right) - \frac{\partial\mathcal{L}}{\partial\phi} = \vec{\tau}_\phi \tag{3.93d}$$

$$= \frac{d}{dt}\left((J_{\mathbf{B}})\left(\dot{\theta}+\dot{\phi}\right) + (m_b r^2)\left(\dot{\phi}\right)\right) - m_b g r\cos\phi \tag{3.93e}$$

$$\therefore \tau_\phi = J_{\mathbf{B}}(\ddot{\theta}+\ddot{\phi}) + m_b r^2(\ddot{\phi}) - m_b g r\cos\phi \tag{3.93f}$$

$$= J_{\mathbf{B}}\ddot{\theta} + J'_{\mathbf{B}}\ddot{\phi} + \tau_g \tag{3.93g}$$

Where J'_b <u>is the parallel axis</u> inertia and τ_g is the gravitational torque arm contribution. The above then leads to the corollary asserted by the system in Fig:3.13:

Lemma 3.4.1. *A torque response opposed to angular acceleration of a doubly rotating body can be found as the contribution of the principle rotational inertia about the first axis of rotation with <u>only</u> the first rotational acceleration and a parallel axis inertia about the second rotational axis with the second, independent rotational acceleration.*
Or the same torque can be found as the inertial opposition to net angular acceleration (sum of both rotations) about the first axis and a point mass inertia opposed to the second rotation about its respective axis.

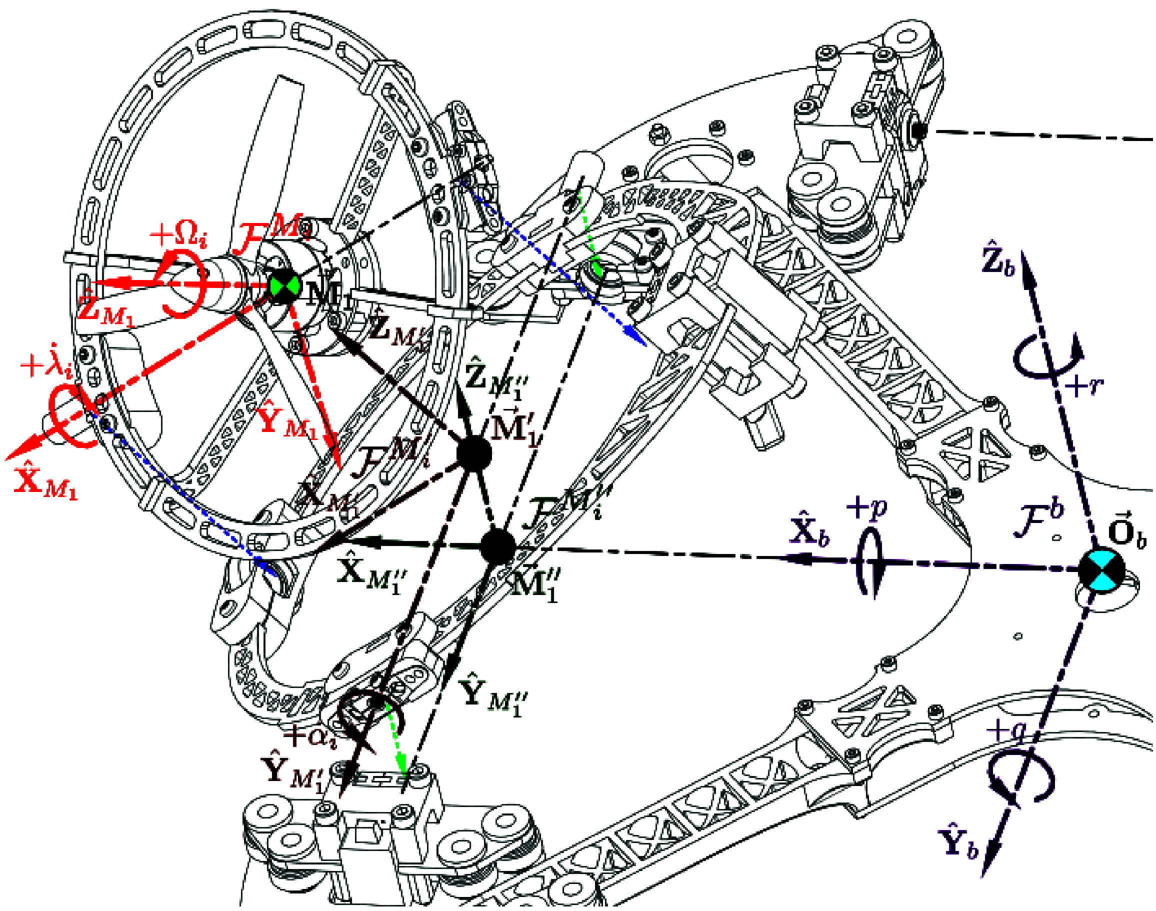


Figure 3.15: Exploded motor module inertial bodies for $\vec{\omega}_b$ response

Returning to the net multibody system and separating the motor module from the entire body structure first (exploded bodies for *motor module 1* in Fig:3.15). Considering only the additional contribution the angular velocity $\vec{\omega}_b$ has on a single motor module and later introducing the entire combined system; the Lagrangian derivation for motion relative to the inertial frame then follows...

The relative $\hat{Z}_{M_i''}$ rotation by σ_i is what differentiates the intermediate frame $\mathcal{F}^{M_i''}$ (used for calculations pertinent to Fig:3.12) and the body frame $\mathcal{F}^b$. The familiar rotor assembly, inner and middle ring structure's inertias from Eq:3.76a,3.77a and 3.78a have respective counterparts aligned with $\mathcal{F}^b$:

$$J_r''' = R_z(\sigma)\big(J_r''\big)R_z^{-1}(\sigma) = R_z(\sigma)R_y(\alpha)R_x(\lambda)\big(J_r\big)R_x^{-1}(\lambda)R_y^{-1}(\alpha)R_z^{-1}(\sigma) \tag{3.94a}$$

$$J_{ir}''' = R_z(\sigma)\big(J_{ir}''\big)R_z^{-1}(\sigma) = R_z(\sigma)R_y(\alpha)R_x(\lambda)\big(J_{ir}\big)R_x^{-1}(\lambda)R_y^{-1}(\alpha)R_z^{-1}(\sigma) \tag{3.94b}$$

$$J_m'' = R_z(\sigma)\big(J_m'\big)R_z^{-1}(\sigma) = R_z(\sigma)R_y(\alpha)\big(J_m\big)R_y^{-1}(\alpha)R_z^{-1}(\sigma) \tag{3.94c}$$

Where σ_i in Eq:3.94 is the relative orthogonal $\hat{Z}_{M_i''}$ difference between frames $\mathcal{F}^{M_i''}$ and $\mathcal{F}^b$ defined before in Eq:2.16 and illustrated previously in Fig:2.9. Because σ_i is constant for each $i \in [1:4]$, inertial rates for each component of the motor module is simply transformations of $\dot{J}_r''$,$\dot{J}_{ir}''$ and $\dot{J}_m'$ previously in Eq:3.76b,3.77b and 3.78e. Or more generally, for some inertia J constant in $\mathcal{F}^{M_i''}$:

$$\frac{d}{dt}\Big(R_z(\sigma)\big(J\big)R_z^{-1}(\sigma)\Big) = 0 \tag{3.95a}$$

So, dropping the σ argument to indicate $R_z(\sigma)$ is a constant, $R_z(\sigma) \rightarrow R_z$ is implied. The rotor, inner and middle inertial rates relative to the body frame $\mathcal{F}^b$ then follow respectively:

$$\dot{J}_r''' = R_z\big(\dot{J}_r''\big)R_z^{-1} \tag{3.95b}$$

$$\dot{J}_{ir}''' = R_z\big(\dot{J}_{ir}''\big)R_z^{-1} \tag{3.95c}$$

$$\dot{J}_m'' = R_z\big(\dot{J}_m'\big)R_z^{-1} \tag{3.95d}$$

Similarly, angular velocities for each separate body (rotor, inner and middle rings) in $\mathcal{F}^b$ but relative to the inertial frame $\mathcal{F}^I$ are then, first for the rotor:

$$\vec{\omega}_{r/I} = \vec{\Omega}_i''' + \dot{\vec{\lambda}}_i'' + \dot{\vec{\alpha}}_i' + \vec{\omega}_{b/I} \tag{3.96a}$$

$$= R_zR_y(\alpha)R_x(\lambda)\vec{\Omega}_i + R_zR_y(\alpha)\dot{\vec{\lambda}}_i + R_z\dot{\vec{\alpha}}_i + \vec{\omega}_b \qquad \in \mathcal{F}^b \tag{3.96b}$$

Extending that to the inner ring's rotational velocity:

$$\vec{\omega}_{n/I} = \dot{\vec{\lambda}}_i'' + \dot{\vec{\alpha}}_i' + \vec{\omega}_{b/I} \tag{3.97a}$$

$$= R_zR_y(\alpha)\dot{\vec{\lambda}}_i + R_z\dot{\vec{\alpha}}_i + \vec{\omega}_{b/I} \qquad \in \mathcal{F}^b \tag{3.97b}$$

And lastly the middle ring structure has a relative angular rate:

$$\vec{\omega}_{m/I} = \dot{\vec{\alpha}}_i' + \vec{\omega}_{b/I} \tag{3.98a}$$

$$= R_z\dot{\vec{\alpha}}_i + \vec{\omega}_b \qquad \in \mathcal{F}^b \tag{3.98b}$$

Noting that Lemma:3.4.1 and the parallel axis term in Eq:3.93g refer to the parallel axis difference between the *center of mass* and the resultant rotational axis. The vector difference between the rotated center of mass for a motor module $C.M_{\mathrm{p}}''(\lambda_i, \alpha_i)$ and the body frame origin $\vec{\mathbf{O}}_b$ is defined:

$$C.M_{\mathrm{p}}''(\lambda_i, \alpha_i) = \frac{m_{\mathrm{n}}C.M_{\mathrm{n}}'''(\lambda_i, \alpha_i) + m_{\mathrm{m}}C.M_{\mathrm{m}}''(\alpha_i)}{m_{\mathrm{p}}} \tag{3.99a}$$

With $C.M_{\mathrm{n}}'''(\lambda_i, \alpha_i)$ and $C.M_{\mathrm{m}}''(\alpha_i)$ being rotated inner and middle ring centers of mass respectively from Eq:2.29d and Eq:2.30d:

$$\therefore C.M_{\mathrm{p}}''(\lambda_i, \alpha_i) = \frac{m_{\mathrm{n}}R_zR_y(\alpha)R_x(\lambda)C.M_{\mathrm{n}} + m_{\mathrm{m}}R_zR_y(\alpha)C.M_{\mathrm{m}}}{m_{\mathrm{n}} + m_{\mathrm{n}}} \tag{3.99b}$$

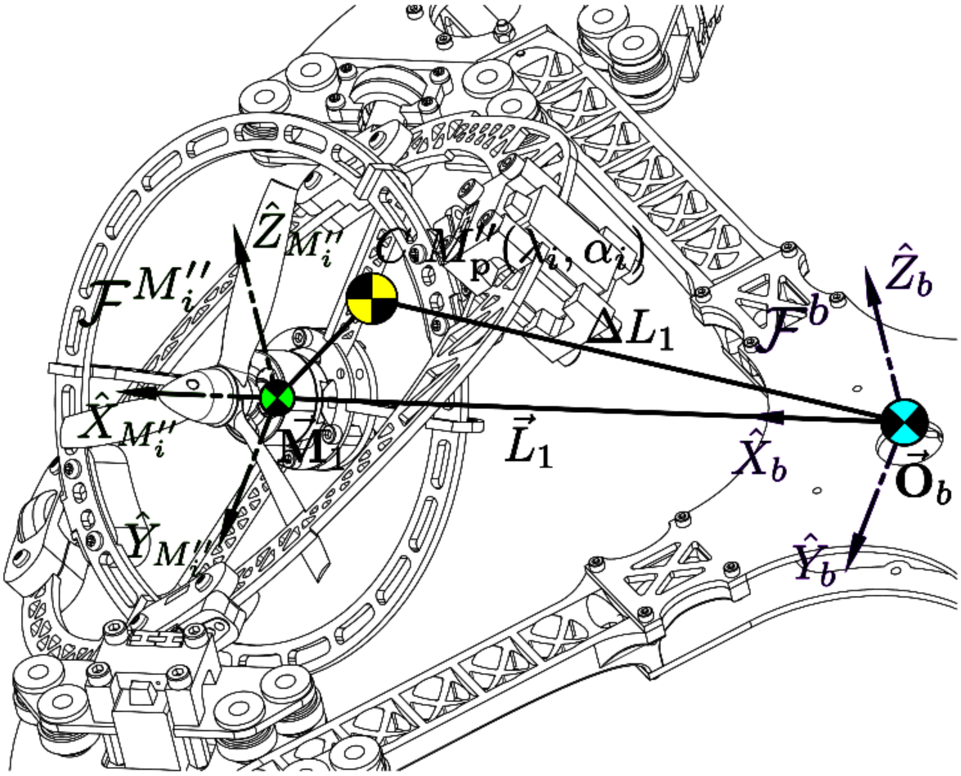


Figure 3.16: Illustration of rotated center of gravity $C.M_p''(\lambda_i, \alpha_i)$

Which leads to the vector difference ΔL_i, with $L = 196.15$ mm illustrated for module 1 in Fig:3.16.

$$\Delta L_i = \vec{L}_i + C.M_p''(\lambda_i, \alpha_i) \tag{3.99c}$$

The time derivative of that module's moving center of gravity, $d/dt\big(C.M_p''(\lambda_i, \alpha_i)\big)$ relative to the origin $\vec{\mathbf{O}}_b$, is:

$$\therefore \Delta \dot{L}_i = \frac{d}{dt}\Big(C.M_p''(\lambda_i, \alpha_i)\Big) \tag{3.99d}$$

$$= \frac{1}{m_p}\Big(m_n\big(R_z\big([\dot{\vec{\alpha}}_i]_\times R_y(\alpha)R_x(\lambda)C.M_n + R_y(\alpha)[\dot{\vec{\lambda}}_i]_\times R_x(\lambda)C.M_n\big) + m_m R_z[\dot{\vec{\alpha}}_i]_\times R_y(\alpha)C.M_m\Big) \tag{3.99e}$$

Then, extended from Lemma:3.4.1, the motor module's *point-mass* inertia J_H about the origin $\vec{\mathbf{O}}_b$ is defined, with net motor module mass $m_p = m_n + m_m$, using masses m_n and m_m from Eq:2.29a and Eq:2.30a:

$$J_H \triangleq m_p\Big((\Delta L_i \cdot \Delta L_i)\mathbb{I}_{3\times 3} - \Delta L_i \otimes \Delta L_i\Big) \tag{3.100a}$$

Or using the inner and outer products matrix definitions:

$$= m_p\Big(\big[\Delta L_i\big]^T\big[\Delta L_i\big] - \big[\Delta L_i\big]\big[\Delta L_i\big]^T\Big) \tag{3.100b}$$

Which leads to the that point mass' inertial rate $d/dt\big(J_H\big)$:

$$\dot{J}_H = m_p\Big(\big[\Delta \dot{L}_i\big]^T\big[\Delta L_i\big] + \big[\Delta L_i\big]^T\big[\Delta \dot{L}_i\big] - \big[\Delta \dot{L}_i\big]\big[\Delta L_i\big]^T - \big[\Delta L_i\big]\big[\Delta \dot{L}_i\big]^T\Big) \tag{3.100c}$$

Unfortunately that inertial rate, $\dot{J}_H$ in Eq:3.100c, cannot be simplified further to a more concise form. The Lagrangian $\mathcal{L}_{p/I}$ for the energy of a single motor module about the origin $\vec{\mathbf{O}}_b$ can then be constructed. This time, *including* the gravitational potential energy component:

$$\mathcal{L}_{p/I} = \frac{1}{2}\vec{\omega}_{r/I}^T(J_r''')\vec{\omega}_{r/I} + \frac{1}{2}\vec{\omega}_{n/I}^T(J_{ir}''')\vec{\omega}_{n/I} + \frac{1}{2}\vec{\omega}_{m/I}^T(J_m'')\vec{\omega}_{m/I} + \vec{\omega}_{b/I}^T(J_H)\vec{\omega}_{b/I} + m_p\vec{G}_b \cdot \big(R_I^b(\eta)\vec{\mathcal{E}}_I + \Delta L_i\big) \tag{3.101}$$

Where the term $m_b\vec{G}_b \cdot \big(R_I^b(\eta)\vec{\mathcal{E}}_I + \Delta L_i\big)$ is the vector analogue of gravitational potential energy $m\vec{g}h$ with $R_I^b(\eta)\vec{\mathcal{E}}_I$ being the relative X-Y-Z inertial frame position in the body frame $\mathcal{F}^b$ relative to the body origin $\vec{\mathbf{O}}_b$. Expanding $\mathcal{L}_{p/I}$ with terms defined previously:

$$\begin{aligned}\rightarrow \mathcal{L}_{p/I} = {} & \Big[\vec{\Omega}_i''' + \dot{\vec{\lambda}}_i'' + \dot{\vec{\alpha}}_i' + \vec{\omega}_b\Big]^T (J_r''')\Big[\vec{\Omega}_i''' + \dot{\vec{\lambda}}_i'' + \dot{\vec{\alpha}}_i + \vec{\omega}_b\Big] + \Big[\dot{\vec{\lambda}}_i'' + \dot{\vec{\alpha}}_i' + \vec{\omega}_b\Big]^T (J_{ir}''')\Big[\dot{\vec{\lambda}}_i'' + \dot{\vec{\alpha}}_i' + \vec{\omega}_b\Big] \\ & \Big[\dot{\vec{\alpha}}_i' + \vec{\omega}_b\Big]^T (J_m'')\Big[\dot{\vec{\alpha}}_i' + \vec{\omega}_b\Big] + \vec{\omega}_b^T\Big(m_p\Big(\big[\Delta L_i\big]^T\big[\Delta L_i\big] - \big[\Delta L_i\big]\big[\Delta L_i\big]^T\Big)\Big)\vec{\omega}_b \\ & + m_p\vec{G}_b \cdot \big(R_I^b(\eta)\vec{\mathcal{E}}_I + \Delta L_i\big) \end{aligned} \tag{3.102}$$

Applying partial derivatives of the Lagrangian formulation to $\mathcal{L}_{p/I}$ relative to the angular path coordinates $\vec{\eta}_b$ and $\vec{\omega}_b$ to find generalized forced $\vec{\mathbf{W}}(u \cdot i)$. Recalling $\vec{\eta}_b$ is the angular orientation from Eq:2.12e, defined entirely in the body frame $\mathcal{F}^b$, and similarly assuming that $\partial/\partial\vec{\eta}_b\big(\Delta L_i\big) \approx 0$:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}_{p/I}}{\partial \dot{\vec{\eta}}_b}\right) - \frac{\partial \mathcal{L}_{p/I}}{\partial \vec{\eta}_b} = \frac{d}{dt}\left(\frac{\partial \mathcal{L}_{p/I}}{\partial \vec{\omega}_b}\right) - \frac{\partial \mathcal{L}_{p/I}}{\partial \vec{\eta}_b} = \vec{\mathbf{W}}(u_i) = \vec{\tau}_M(u_i) \tag{3.103a}$$

$$\begin{aligned} = {} & \frac{d}{dt}\Big((J_r''')\Big[\vec{\Omega}_i''' + \dot{\vec{\lambda}}_i'' + \dot{\vec{\alpha}}_i' + \vec{\omega}_b\Big] + (J_{ir}''')\Big[\dot{\vec{\lambda}}_i'' + \dot{\vec{\alpha}}_i' + \vec{\omega}_b\Big] + (J_m'')\Big[\dot{\vec{\alpha}}_i' + \vec{\omega}_b\Big] + (J_H)\Big[\vec{\omega}_b\Big]\Big) \\ & - m_p\vec{G}_b \times \Delta L_i \end{aligned} \tag{3.103b}$$

Then using inertial rate derivatives from Eq:3.95b-3.95d and $\dot{J}_H$ from Eq:3.100c, and inserting relative angular velocities from Eq:3.96-3.98:

$$\begin{aligned} = {} & \Big[(\dot{J}_r''')(\vec{\Omega}_i''' + \dot{\vec{\lambda}}_i'' + \dot{\vec{\alpha}}_i' + \vec{\omega}_b)\Big] + \Big[(J_r''')\dot{\vec{\Omega}}_i''' + \vec{\omega}_{n/I} \times (J_r''')\vec{\Omega}_i''' + (J_r''')\ddot{\vec{\lambda}}_i'' + \vec{\omega}_{n/I} \times (J_r''')\dot{\vec{\lambda}}_i'' \\ & + (J_r''')\ddot{\vec{\alpha}}_i' + \vec{\omega}_{m/I} \times (J_r''')\dot{\vec{\alpha}}_i' + (J_r''')\dot{\vec{\omega}}_b + \vec{\omega}_{b/I} \times (J_r''')\vec{\omega}_b\Big] + \Big[(\dot{J}_{ir}''')(\dot{\vec{\lambda}}_i'' + \dot{\vec{\alpha}}_i' + \vec{\omega}_b)\Big] + \Big[(J_{ir}''')\ddot{\vec{\lambda}}_i'' \\ & + \vec{\omega}_{n/I} \times (J_{ir}''')\dot{\vec{\lambda}}_i'' + (J_{ir}''')\ddot{\vec{\alpha}}_i' + \vec{\omega}_{m/I} \times (J_{ir}''')\dot{\vec{\alpha}}_i' + (J_{ir}''')\dot{\vec{\omega}}_b + \vec{\omega}_{b/I} \times (J_{ir}''')\vec{\omega}_b\Big] + \Big[(\dot{J}_m'')(\dot{\vec{\alpha}}_i' + \vec{\omega}_b)\Big] \\ & \Big[(J_m'')\ddot{\vec{\alpha}}_i' + \vec{\omega}_{m/I} \times (J_m'')\dot{\vec{\alpha}}_i' + (J_m'')\dot{\vec{\omega}}_b + \vec{\omega}_{b/I} \times (J_m'')\vec{\omega}_b\Big] + \Big[(\dot{J}_h)\vec{\omega}_b\Big] + \Big[(J_H)\dot{\vec{\omega}}_b + \vec{\omega}_{b/I} \times (J_h)\vec{\omega}_b\Big] \\ & - \Big[m_p\vec{G}_b \times \Delta L_i\Big] \end{aligned} \tag{3.103c}$$

After expanding relative angular velocity terms; $\vec{\omega}_{n/I}$, $\vec{\omega}_{m/I}$ and $\vec{\omega}_{b/I}$ and applying some mathematics, Eq:3.103c is shown to include a transformed component of Eq:3.85c.

$$\begin{aligned}\rightarrow \frac{d}{dt}\left(\frac{\partial \mathcal{L}_{p/I}}{\partial \vec{\omega}_b}\right) - \frac{\partial \mathcal{L}_{p/I}}{\partial \vec{\eta}_b} = {} & R_z\frac{d}{dt}\left(\frac{\partial \mathcal{L}_{m/p}}{\partial \dot{\vec{\mathbf{v}}}}\right) + (\dot{J}_r''')\vec{\omega}_b + \vec{\omega}_b \times (J_r''')\vec{\Omega}_i''' + \vec{\omega}_b \times (J_r''')\dot{\vec{\lambda}}_i'' \\ & + \vec{\omega}_b \times (J_r''')\dot{\vec{\alpha}}_i' + \vec{\omega}_b \times (J_r''')\vec{\omega}_b + J_r'''\dot{\vec{\omega}}_b + (\dot{J}_{ir}''')\vec{\omega}_b + \vec{\omega}_b \times (J_{ir}''')\dot{\vec{\lambda}}_i'' + \vec{\omega}_b \times (J_{ir}''')\dot{\vec{\alpha}}_i' \\ & + \vec{\omega}_b \times (J_{ir}''')\vec{\omega}_b + (J_{ir}''')\dot{\vec{\omega}}_b + (\dot{J}_m'')\vec{\omega}_b + \vec{\omega}_b \times (J_m'')\dot{\vec{\alpha}}_i' + \vec{\omega}_b \times (J_m'')\vec{\omega}_b + (J_m'')\dot{\vec{\omega}}_b + (\dot{J}_H)\vec{\omega}_b \\ & + (J_H)\dot{\vec{\omega}}_b + \vec{\omega}_b \times (J_H)\vec{\omega}_b - m_p\vec{G}_b \times \Delta L_i \end{aligned} \tag{3.103d}$$

Combining inertial bodies with the same angular velocities and introducing terms $\vec{\tau}_\lambda$ and $\vec{\tau}_\alpha$ from Eq:3.75 and Eq:3.86 respectively:

$$\begin{aligned}\therefore \vec{\mathbf{W}}(u_i) = \vec{\tau}_M(u_i) = {} & R_z\vec{\tau}_\alpha(\lambda_i, \alpha_i) + R_zR_y(\alpha)\vec{\tau}_\lambda(\lambda_i) + \big(\dot{J}_r''' + \dot{J}_{ir}''' + \dot{J}_m'' + \dot{J}_H\big)\vec{\omega}_b \\ & + \big(J_r''' + J_{ir}''' + J_m'' + J_H\big)\dot{\vec{\omega}}_b + \vec{\omega}_b \times \big(J_r''' + J_{ir}''' + J_m'' + J_H\big)\vec{\omega}_b + \vec{\omega}_b \times \Big((J_r''')(\vec{\Omega}_i''' + \dot{\vec{\lambda}}_i'' + \dot{\vec{\alpha}}_i') \\ & + (J_{ir}''')(\dot{\vec{\lambda}}_i'' + \dot{\vec{\alpha}}_i') + (J_m'')(\dot{\vec{\alpha}}_i')\Big) - m_p\vec{G}_b \times \Delta L_i \end{aligned} \tag{3.103e}$$

And recognizing that $(J_r''' + J_{ir}''' + J_m'' + J_H)$ can be simplified to a parallel axis translation of the transformed net motor module inertia J_p' from Eq:2.24b; analogous to the net motor module inertia defined in Eq:2.27b:

$$\begin{aligned}(J_r''' + J_{ir}''' + J_m'' + J_H) \triangleq {} & R_zR_y(\alpha)R_x(\lambda)\big(J_r\big)R_x^{-1}(\lambda)R_y^{-1}(\alpha)R_z^{-1} \\ & + R_zR_y(\alpha)R_x(\lambda)\big(J_{ir}\big)R_x^{-1}(\lambda)R_y^{-1}(\alpha)R_z^{-1} + R_zR_y(\alpha)\big(J_m\big)R_y^{-1}(\alpha)R_z^{-1} + J_H \end{aligned} \tag{3.104a}$$

$$= R_z\big(J_p\big)R_z^{-1} + m_p\Big(\big[\Delta L_i\big]^T\big[\Delta L_i\big] - \big[\Delta L_i\big]\big[\Delta L_i\big]^T\Big) = J'_{\vec{\mathbf{M}}_i} \tag{3.104b}$$

Moreover, the above can be applied to the associated inertia rates; $\dot{J}'''_r$, $\dot{J}'''_{ir}$, $\dot{J}''_m$ and $\dot{J}_H$. Using Eq:3.95b,3.95c,3.95d and 3.100c it can be shown that:

$$\big(\dot{J}'''_r + \dot{J}'''_{ir} + \dot{J}''_m + \dot{J}_H\big) = \dot{J}'_{\vec{\mathbf{M}}_i} \tag{3.104c}$$

The generalized torque acting on a single motor module, $\vec{\tau}_M(u_i)$ from Eq:3.103e, is then found as combinations of responses to servos λ_i and α_i, the changing inertial rates $\dot{J}'_{\vec{\mathbf{M}}_i}$ as a result of those rotations and finally the net response to the entire frames angular velocity $\vec{\omega}_b$.

$$\begin{aligned}\vec{\tau}_M(u_i) = R_z\vec{\tau}_\alpha(\lambda_i,\alpha_i)) + R_zR_y(\alpha)\vec{\tau}_\lambda(\lambda_i) + \big(\dot{J}'_{\vec{\mathbf{M}}_i}\big)\vec{\omega}_b + \big(J'_{\vec{\mathbf{M}}_i}\big)\dot{\vec{\omega}}_b + \vec{\omega}_b \times \big(J'_{\vec{\mathbf{M}}_i}\big)\vec{\omega}_b \\ + \vec{\omega}_b \times \Big(\big(J''_p\big)\dot{\vec{\alpha}}'_i + \big(J'''_n\big)\dot{\vec{\lambda}}''_i + \big(J'''_r\big)\vec{\Omega}'''_i\Big) - m_p\vec{G}_b \times \Delta L_i \triangleq \vec{\mathbf{W}}(u_i) \qquad \in \mathcal{F}^b\end{aligned} \tag{3.105}$$

Considering the rigid body torque response $\vec{\tau}_y$ for the body structure's motion, J_y. That structure's inertia J_y is a constant and independent of actuator positions in $u \in \mathbb{U}$; explicitly defined in Eq:2.25d.

$$\vec{\tau}_y = \big(J_y\big)\dot{\vec{\omega}}_b + \vec{\omega}_b \times \big(J_y\big)\vec{\omega}_b - C.M_{\mathrm{y}} \times m_{\mathrm{y}}\vec{G}_b \qquad \in \mathcal{F}^b \tag{3.106}$$

The net response for the *entire* multibody system is then a sum of Eq:3.105 for modules $i \in [1:4]$ and $\vec{\tau}_y$ in Eq:3.106. By inspection, without constructing a complete Lagrangian for the entire system, the effective net torque $\vec{\tau}_\mu$, acting on the body frame $\mathcal{F}^b$ is shown to be:

$$\vec{\tau}_\mu = \big(J_y\big)\dot{\vec{\omega}}_b + \vec{\omega}_b \times \big(J_y\big)\vec{\omega}_b - C.M_y \times m_y\vec{G}_b + \sum_{i=1}^{4}\vec{\tau}_M(u_i) \qquad \in \mathcal{F}^b \tag{3.107}$$

Recalling the net vehicles rotational inertia $J_b(u)$, calculated as a function of the actuation matrix u, which was defined previously in 2.31a. It follows that Eq:3.107 reduces and expands to:

$$\begin{aligned}\vec{\tau}_\mu = \big(J_b(u)\big)\dot{\vec{\omega}}_b + \vec{\omega}_b \times \big(J_b(u)\big)\vec{\omega}_b \\ + \sum_{i=1}^{4}\Big[R_z\vec{\tau}_\alpha(\lambda_i,\alpha_i) + R_zR_y(\alpha)\vec{\tau}_\lambda(\lambda_i) + \big(\dot{J}_{\vec{\mathbf{M}}_i}\big)\vec{\omega}_b + \vec{\omega}_b \times \Big(\big(J''_p\big)\dot{\vec{\alpha}}'_i + \big(J'''_n\big)\dot{\vec{\lambda}}''_i + \big(J'''_r\big)\vec{\Omega}'''_i\Big)\Big] \\ - m_p\vec{G}_b \times \sum_{i=1}^{4}\Delta L_i\end{aligned} \tag{3.108}$$

The external torque $\vec{\tau}_\mu$ acting on the vehicle is as a response to the commanded control action, detailed next in Ch:4. The final sum of gravitational torque contributions can be simplified to $\vec{\tau}_g$, from Eq:2.35d, which considers the *net* resultant center of gravity. Then, extending the angular differential equation Eq:3.10d to incorporate the multibody responses derived above:

$$\vec{\tau}_\mu = \big(J_b\big)\dot{\vec{\omega}}_b + \vec{\omega}_b \times \big(J_b\big)\vec{\omega}_b + \vec{\tau}_b(u) - \vec{\tau}_g \tag{3.109a}$$

Defining a new response torque $\vec{\tau}_b(u)$ which represents collective responses from internal rotations relative each body. It can be considered a nonlinear extension of the gyroscopic component of the torque $\vec{\omega}_b \times (J_b)\vec{\omega}_b$ acting on the system. That nonlinear multibody torque is defined then as follows:

$$\vec{\tau}_b(u) \triangleq \dot{J}_b(u)\vec{\omega}_b + \sum_{i=1}^{4}\Big[R_z\vec{\tau}_\alpha(\lambda_i,\alpha_i) + R_zR_y(\alpha_i)\vec{\tau}_\lambda(\lambda_i) + \vec{\omega}_b \times \Big(\big(J''_p\big)\dot{\vec{\alpha}}'_i + \big(J'''_n\big)\dot{\vec{\lambda}}''_i + \big(J'''_r\big)\vec{\Omega}'''_i\Big)\Big] \tag{3.109b}$$

And using the net gravitational torque arm $\vec{\tau}_g$ defined earlier in Eq:2.35d:

$$\vec{\tau}_g \triangleq \Delta C.G \times m_b\vec{G}_b \tag{3.109c}$$

Noting that $\dot{J}_b(u)$ is another introduced term which is the sum of all motor module inertia rates from Eq:3.104c, given that body structures inertia J_y is constant:

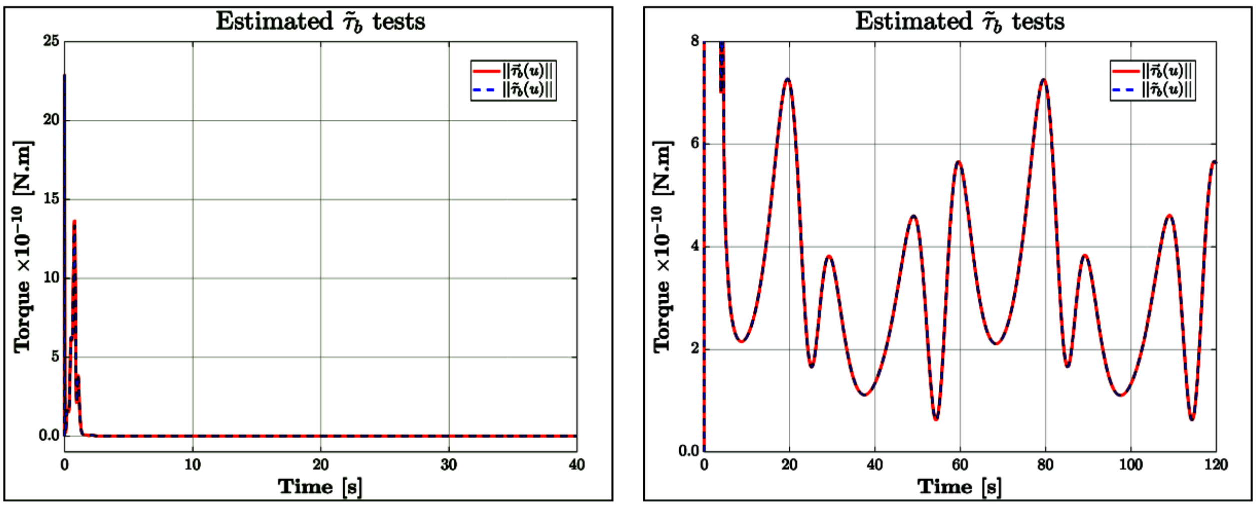
$$\dot{J}_b(u) \triangleq \sum_{i=1}^{4} \left(\dot{J}'_{\vec{\mathbf{M}}_i}\right) + \dot{J}_y = \sum_{i=1}^{4} \left(\dot{J}'_{\vec{\mathbf{M}}_i}\right) \tag{3.110}$$

The torque $\vec{\tau}_b(u)$ from Eq:3.109b is the most important result here however; definitions of $\vec{\tau}_\alpha(\lambda_i, \alpha_i)$ and $\vec{\tau}_\lambda(\lambda_i)$, in Eq:3.75 and Eq:3.86 respectively, were necessary to simplify and isolate different components of Eq:3.109b. The most complicated process in evaluating Eq:3.109b is calculating inertial rate derivatives at each sampling interval. Finding solutions to Eq:3.70d for each reconfigured inertial body is cumbersome; an alternative is to instead substitute continuous time inertia rates, calculated from angular velocities, with a discrete time approximations.

The difference of instantaneous calculations for $J_b(u)$, $J_p(u_i)$, $J_m(u_i)$ and $J_n(u_i)$, for the net body structure, each net motor module and each body within a motor module respectively, between sample times n and $n-1$ can be found simply as follows:

$$\Delta\widetilde{J}_b(u) \triangleq \big(J_b(u_n) - J_b(u_{n-1})\big)/\Delta t \approx \dot{J}_b(u) \tag{3.111}$$

Each inertia must be calculated at every control loop interval regardless, exploiting that fact will reduce computational overhead. Plots for magnitudes of both *true* torque response $\|\vec{\tau}_b(u)\|$, and the discrete approximation $\|\widetilde{\tau}_b(u)\|$ which uses exclusively discrete inertia rates are shown in Fig:3.17. Both responses were calculated over a typical flight envelope, tracking an orbital trajectory, but the focus is on the proposed approximation's quality. The error between the approximated torque $\|\widetilde{\tau}_b(u)\|$ and the full complexity body torque $\|\vec{\tau}_b(u)\|$ is similarly shown in Fig:3.18.

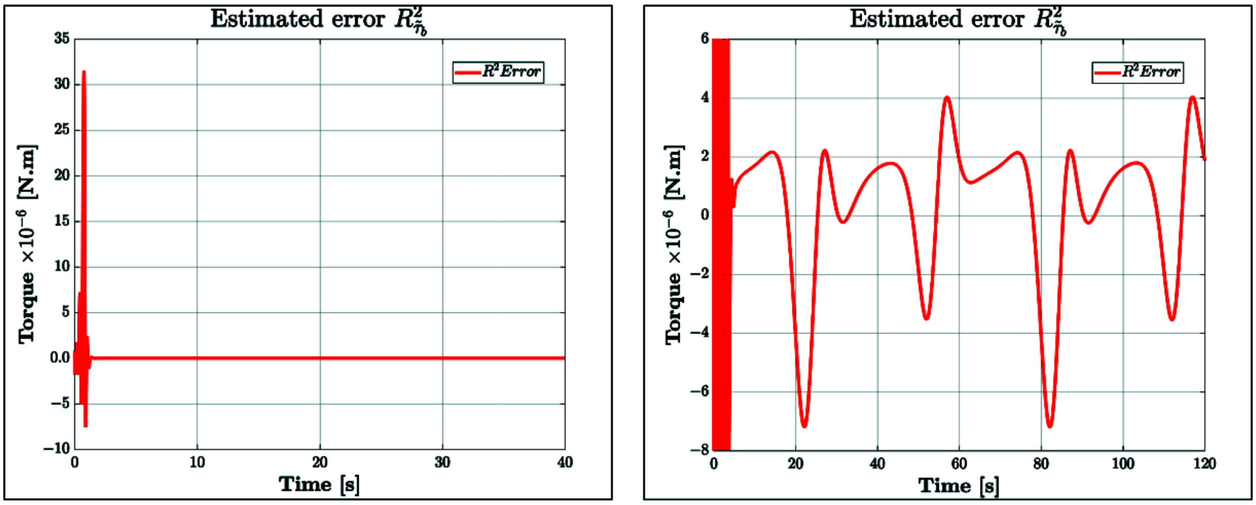


(a) Initial torque response spike **(b)** Steady state torque response

Figure 3.17: Approximated and true body torque responses

Sample rates of 100 Hz were used to calculate the instantaneous inertia values throughout the trajectory. Initial torques spike from time $t_0 \to 2$ s occurs as the vehicle first steps to the attitude setpoint from rest, with an attitude at the origin, shown in Fig:3.17a. Once the vehicle reaches steady state trajectory tracking the induced torque reduces dramatically, plotted for a longer time Fig:3.17b. The plot for $\|\tilde{\tau}_b(u)\|$ uses discrete time derivatives as per Eq:3.111.

Fig:3.18 shows the error between $\|\vec{\tau}_b(u)\|$ and approximated $\|\tilde{\tau}_b(u)\|$, again the initial spike is from starting configuration changes; thereafter the error reduces dramatically. On average the approximation asserted in Eq:3.111 is a reasonable one with an error typically three orders of magnitude smaller than the signal it represents, or $\times 10^{-3}$ Nm, less than the torque $\vec{\tau}_b(u)$. The transfer block for the dynamics in simulation still make use of a continuous time derivative model whilst plant dependent controller compensation uses a reduced complexity approximation…



(a) Initial step torque error

(b) Steady state torque error

Figure 3.18: Approximated and true body torque responses

Any measurement or modelling errors associated with the above inertial response models and their explicit values presented in Sec:2.3 can easily be compensated for as plant disturbances. More specifically errors in the above system are modelled as plant dependent state uncertainties; and could be adaptively compensated for accordingly.

3.4.2 Simulation and verification of induced model

Linearization simulation and comparison

Previously in Sec:3.4.1 the proposed Lagrangian energy functions for both $\Delta\lambda_i$ and $\Delta\alpha_i$ servo rotations and their response to body angular velocities $\vec{\omega}_b$ were derived. Recalling those equations; the inner ring net (kinetic) energy Lagrangian from Eq:3.65 is:

$$\mathcal{L}_{n/m} = \frac{1}{2}\vec{\omega}_{r/m}{}^T(J'_r)\vec{\omega}_{r/m} + \frac{1}{2}\vec{\omega}_{n/m}{}^T(J'_{ir})\vec{\omega}_{n/m} \tag{3.112a}$$

$$= \frac{1}{2}\Big(R_x(\lambda)\vec{\Omega}_i + \dot{\vec{\lambda}}_i\Big)^T\Big(R_x(\lambda)(J_r)R_x^{-1}(\lambda)\Big)\Big(R_x(\lambda)\vec{\Omega}_i + \dot{\vec{\lambda}}_i\Big) + \frac{1}{2}\big(\dot{\vec{\lambda}}_i\big)^T\Big(R_x(\lambda)(J_{ir})R_x^{-1}(\lambda)\Big)\big(\dot{\vec{\lambda}}_i\big) \tag{3.112b}$$

And similarly for the middle ring's net (kinetic) energy from Eq:3.82:

$$\mathcal{L}_{m/p} = \frac{1}{2}\vec{\omega}_{r/p}^{\,T}(J''_r)\vec{\omega}_{r/p} + \frac{1}{2}\vec{\omega}_{n/p}^{\,T}(J''_{ir})\vec{\omega}_{n/p} + \frac{1}{2}\vec{\omega}_{m/p}^{\,T}(J'_m)\vec{\omega}_{m/p} \tag{3.112c}$$

$$\begin{aligned} = \frac{1}{2}\Big(R_y(\alpha)R_x(\lambda)\vec{\Omega}_i + R_y(\alpha)\dot{\vec{\lambda}}_i + \dot{\vec{\lambda}}_i\Big)^T\Big(R_y(\alpha)(J'_r)R_y^{-1}(\alpha)\Big)\Big(R_y(\alpha)R_x(\lambda)\vec{\Omega}_i + R_y(\alpha)\dot{\vec{\lambda}}_i + \dot{\vec{\alpha}}_i\Big) \\ \frac{1}{2}\Big(R_y(\alpha)\dot{\vec{\lambda}}_i + \dot{\vec{\alpha}}_i\Big)^T\Big(R_y(\alpha)(J'_{ir})R_y^{-1}(\alpha)\Big)\Big(R_y(\alpha)\dot{\vec{\lambda}}_i + \dot{\vec{\alpha}}_i\Big) \\ + \frac{1}{2}\big(\dot{\vec{\alpha}}_i\big)^T\Big(R_y(\alpha)(J_m)R_y^{-1}(\alpha)\Big)\big(\dot{\vec{\alpha}}_i\big) \end{aligned} \tag{3.112d}$$

Solving for the generalized forces acting on each system requires application of Euler-Lagrange formulation, using partial derivatives relative to generalized path coordinates. Both the inner and middle ring systems were defined with relative coordinate paths for the angular servo position $\vec{\mathbf{u}} = \begin{bmatrix}\lambda_i & 0 & 0\end{bmatrix}^T$ and $\vec{\mathbf{v}} = \begin{bmatrix}0 & \alpha_i & 0\end{bmatrix}^T$ respectively. The generalized forces for both systems are then, using the Euler-Lagrange formulation:

$$\underbrace{\frac{d}{dt}\left(\frac{\partial\mathcal{L}_{n/m}}{\partial\dot{\vec{\mathbf{u}}}}\right) - \frac{\partial\mathcal{L}_{n/m}}{\partial\vec{\mathbf{u}}} = \vec{\mathbf{U}}(\lambda_i)}_{\text{Inner ring}} \quad \text{and} \quad \underbrace{\frac{d}{dt}\left(\frac{\partial\mathcal{L}_{m/p}}{\partial\dot{\vec{\mathbf{v}}}}\right) - \frac{\partial\mathcal{L}_{m/p}}{\partial\vec{\mathbf{v}}} = \vec{\mathbf{V}}(\lambda_i, \alpha_i)}_{\text{Middle ring}} \tag{3.113}$$

The assumption proposed in Eq:3.67, presented in [10], is used to linearize and reduce partial derivatives of the respective inner and middle ring Lagrangians in Eq:3.113. Those simplifications are such that both terms in Eq:3.113 respectively simplify to:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}_{n/m}}{\partial \dot{\vec{\mathbf{u}}}}\right) = \vec{\mathbf{U}}(\lambda_i)\ \Big|_{\left(\partial \mathcal{L}_{n/m}/\partial \vec{\mathbf{u}}\right)\approx 0} \quad \text{and} \quad \frac{d}{dt}\left(\frac{\partial \mathcal{L}_{m/p}}{\partial \dot{\vec{\mathbf{v}}}}\right) = \vec{\mathbf{V}}(\alpha_i, \lambda_i)\Big|_{\left(\partial \mathcal{L}_{m/p}/\partial \vec{\mathbf{v}}\right)\approx 0} \tag{3.114}$$

Considering first the case of the inner ring Lagrangian $\mathcal{L}_{n/m}$ from Eq:3.112b relative to the middle ring frame $\mathcal{F}^{M_i'}$; the partial derivative taken with respect to path variable $\vec{\mathbf{u}} = \vec{\lambda}_i$ was simplified:

$$\frac{\partial \mathcal{L}_{n/m}}{\partial \vec{\mathbf{u}}} = \frac{\partial}{\partial \vec{\lambda}_i}\left(\mathcal{L}_{n/m}\right) \approx 0 \tag{3.115}$$

Expanding Eq:3.115 and finding the partial derivative of that Lagrangian, $\mathcal{L}_{n/m}$ in Eq:3.112b, with respect to $\vec{\mathbf{u}} = \vec{\lambda}_i$ yields:

$$\begin{aligned}\frac{\partial \mathcal{L}_{n/m}}{\partial \vec{\mathbf{u}}} = \frac{\partial \mathcal{L}_{n/m}}{\partial \vec{\lambda}_i} &= \frac{1}{2}\left(\frac{\partial}{\partial \vec{\lambda}_i}\Big(R_x(\lambda)\vec{\Omega}_i\Big)\right)^T\Big(R_x(\lambda)\big(J_r\big)R_x^{-1}(\lambda)\Big)\Big(R_x(\lambda)\vec{\Omega}_i + \dot{\vec{\lambda}}_i\Big)\\ &+ \frac{1}{2}\Big(R_x(\lambda)\vec{\Omega}_i + \dot{\vec{\lambda}}_i\Big)^T\left(\frac{\partial}{\partial \vec{\lambda}_i}\Big(R_x(\lambda)\big(J_r\big)R_x^{-1}(\lambda)\Big)\Big(R_x(\lambda)\vec{\Omega}_i + \dot{\vec{\lambda}}_i\Big)\right.\\ &+ \frac{1}{2}\Big(R_x(\lambda)\vec{\Omega}_i + \dot{\vec{\lambda}}_i\Big)^T\Big(R_x(\lambda)\big(J_r\big)R_x^{-1}(\lambda)\Big)\left(\frac{\partial}{\partial \vec{\lambda}_i}\Big(R_x(\lambda)\vec{\Omega}_i\Big)\right)\\ &+ \frac{1}{2}\Big(\dot{\vec{\lambda}}_i\Big)^T\left(\frac{\partial}{\partial \vec{\lambda}_i}\Big(R_x(\lambda)\big(J_{ir}\big)R_x^{-1}(\lambda)\Big)\right)\Big(\dot{\vec{\lambda}}_i\Big)\end{aligned} \tag{3.116}$$

Testing that assumption, presented in Eq:3.67; the partial derivative components of Eq:3.116 are explicitly calculated. First, considering the rotor's transformed inertia $J_r' = R_x(\lambda)\big(J_r\big)R_x^{-1}(\lambda)$ with a partial derivative as per the differential product rule:

$$\frac{\partial}{\partial \vec{\mathbf{u}}}\left(J_r'\right) = \frac{\partial}{\partial \vec{\lambda}_i}\Big(R_x(\lambda)\big(J_r\big)R_x^{-1}(\lambda)\Big) \tag{3.117a}$$

$$= \frac{\partial}{\partial \vec{\lambda}_i}\Big(R_x(\lambda)\Big)\big(J_r\big)R_x^{-1}(\lambda) + R_x(\lambda)\big(J_r\big)\frac{\partial}{\partial \vec{\lambda}_i}\Big(R_x^{-1}(\lambda)\Big) \tag{3.117b}$$

Similarly for the inner ring transformed inertia's contribution, $J_{ir}' = R_x(\lambda)\big(J_{ir}\big)R_x^{-1}(\lambda)$ from Eq:3.112b, however, *without* including the rotor body's inertia:

$$\frac{\partial}{\partial \vec{\mathbf{u}}}\left(J_{ir}'\right) = \frac{\partial}{\partial \vec{\lambda}_i}\Big(R_x(\lambda)\Big)\big(J_{ir}\big)R_x^{-1}(\lambda) + R_x(\lambda)\big(J_{ir}\big)\frac{\partial}{\partial \vec{\lambda}_i}\Big(R_x^{-1}(\lambda)\Big) \tag{3.117c}$$

The final partial derivative to take note of is that for the rotor's angular velocity $\vec{\Omega}_i$, in rad.s^{-1}, but transformed to the inner ring frame $\mathcal{F}^{M_i'}$ which the Lagrangian in Eq:3.112a is taken with respect to:

$$\frac{\partial}{\partial \vec{\lambda}_i}\Big(R_x(\lambda)\vec{\Omega}_i\Big) \tag{3.117d}$$

The application of the rotation matrix linearization, from Eq:3.67, to the partial derivative rotations in Eq:3.117b-3.117d is:

$$R_x(\bar{\lambda}_i + \partial\lambda_i) \approx \Big(1 - [\Phi_x(\bar{\lambda}_i)\partial\lambda_i]_\times\Big)R_x(\bar{\lambda}_i) \tag{3.118a}$$

Where $\Phi_x(\bar{\lambda}_i)$ is simply the partial derivative of the Euler matrix, $\Phi(\eta)$ from Eq:2.12i, with respect to an $\hat{X}$ axis rotation. That being:

$$\Phi_x(\bar{\lambda}_i) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \sin\bar{\lambda}_i & \cos\bar{\lambda}_i \\ 0 & -\cos\bar{\lambda}_i & \sin\bar{\lambda}_i \end{bmatrix} \tag{3.118b}$$

So then, for some nominal angle $\bar{\lambda}_i$ perturbed by some small deviation $\partial\lambda_i$, Eq:3.118a expands to:

$$\therefore R_x(\bar{\lambda}_i + \partial\lambda_i) \approx R_x(\bar{\lambda}_i) - \begin{bmatrix} 0 & 0 & 0 \\ 0 & \big(c^2\bar{\lambda}_i - s^2\bar{\lambda}_i\big) & \big(-c\bar{\lambda}_i s\bar{\lambda}_i - s\bar{\lambda}_i c\bar{\lambda}_i\big) \\ 0 & \big(s\bar{\lambda}_i c\bar{\lambda}_i + c\bar{\lambda}_i s\bar{\lambda}_i\big) & \big(c^2\bar{\lambda}_i - s^2\bar{\lambda}_i\big) \end{bmatrix} \partial\lambda_i \tag{3.118c}$$

Which, obviously, when the perturbation $\partial\lambda_i$ away from the nominal $\bar{\lambda}_i$ is small, it follows that:

$$\rightarrow R_x(\bar{\lambda}_i + \partial\lambda_i) \approx R_x(\bar{\lambda}_i)\Big|_{\partial\lambda_i << 1} \tag{3.118d}$$

That simplification then applies to Eq:3.117b,3.117c and 3.117d, for a small $\partial\lambda_i$:

$$\frac{\partial}{\partial\vec{\lambda}_i} J'_r \approx 0 \tag{3.119a}$$

$$\frac{\partial}{\partial\vec{\lambda}_i} J'_{ir} \approx 0 \tag{3.119b}$$

$$\frac{\partial}{\partial\vec{\lambda}_i} R_x(\lambda)\vec{\Omega}_i \approx 0 \tag{3.119c}$$

It can therefore be said that the assumption in Eq:3.115 is not without merit, or that:

$$\frac{\partial\mathcal{L}_{n/m}}{\partial\vec{\mathbf{u}}} \approx 0 \rightarrow \frac{d}{dt}\bigg(\frac{\partial\mathcal{L}_{n/m}}{\partial\dot{\vec{\mathbf{u}}}}\bigg) - \frac{\partial\mathcal{L}_{n/m}}{\partial\vec{\mathbf{u}}} \approx \frac{d}{dt}\bigg(\frac{\partial\mathcal{L}_{n/m}}{\partial\dot{\vec{\lambda}}_i}\bigg) = \vec{\mathbf{U}}(\lambda_i) = \vec{\tau}_\lambda(\lambda_i) \qquad \in \mathcal{F}^{M'_i} \tag{3.120}$$

A final verification of the proposed rotation matrix approximation is done with a **Simulink** model whose structure is shown in Fig:3.19. The model applies a direct comparison of $\vec{\tau}_\lambda(\lambda_i)$, originally from Eq:3.75, both *with* and *without* the rotation matrix linearizations in Eq:3.119. Convention has it that estimated or approximated values are denoted with a hat accent. Therefore a calculated inner ring torque response, *with linearized* partial derivatives as per Eq:3.119, is termed as $\hat{\tau}_\lambda(\lambda_i)$. The simulated representation of the induced torque response *without* applied linearizations, $\vec{\tau}_\lambda$, is calculated from the block model detailed in Fig:3.19. The complete inner ring Euler-Lagrange formulation is calculated from Eq:3.113.

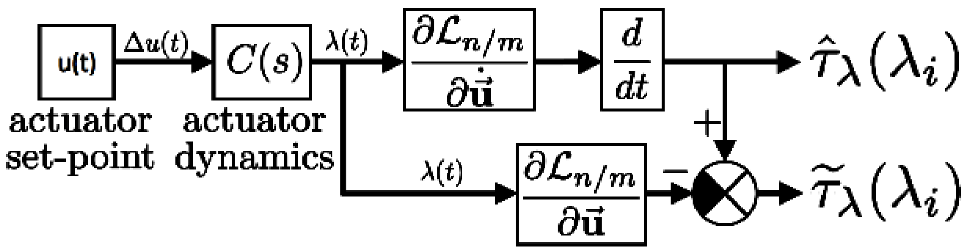
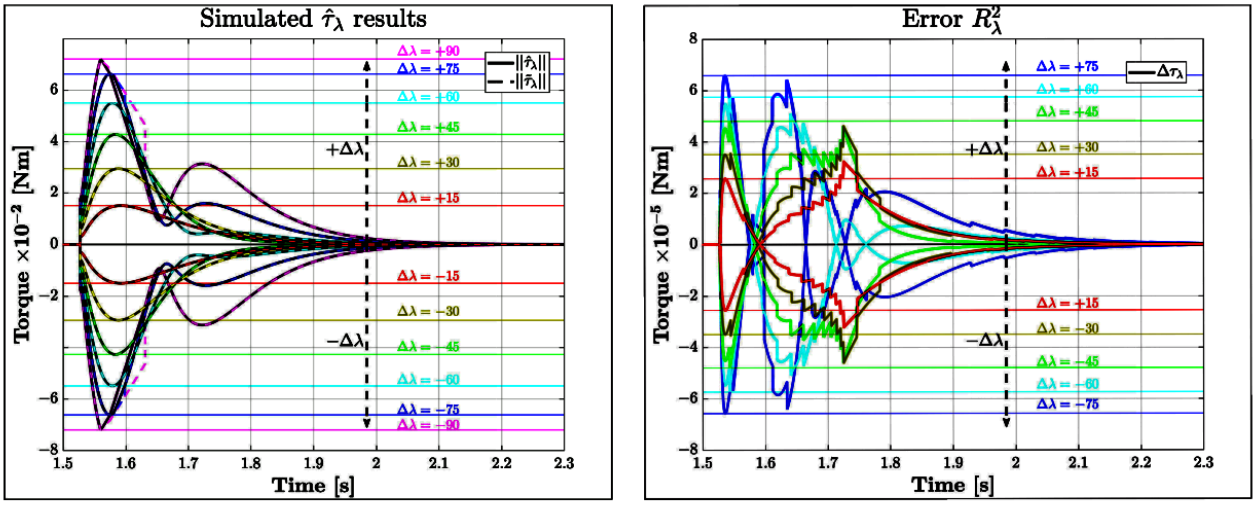


Figure 3.19: Simulink Lagrangian block

The block $u(t)$ in Fig:3.19 represents some commanded change in actuator position, within the actuator space $u \in \mathbb{U}$. That space consists of propeller speeds Ω_i and servo positions λ_i and α_i for $i \in [1:4]$; as detailed in Eq:2.18. Each actuator has its own transfer function, driven by the collective dynamic block $C(s)$, with transfer characteristics determined in Sec:2.4.1. That actuator argument $u(t)$ then leads to some time varying inner ring servo position $\lambda_i(t)$, with a rate $\dot{\lambda}_i(t)$. Both of which are used to calculate the complete Euler-Lagrangian equation:

$$\frac{d}{dt}\bigg(\frac{\partial\mathcal{L}_{n/m}}{\partial\dot{\vec{\mathbf{u}}}}\bigg) - \frac{\partial\mathcal{L}_{n/m}}{\partial\vec{\mathbf{u}}} = \vec{\tau}_\lambda(\lambda_i) \tag{3.121}$$

Seeing that Eq:3.121 produces a 3-D vector result and not a scalar; vector magnitudes $\|\vec{\tau}_\lambda(\lambda_i)\|$ and $\|\hat{\tau}_\lambda(\lambda_i)\|$ are considered. The objective here is to quantify the effect a rotation matrix linearization has on the estimated generalized torque calculations detailed above.



(a) $\|\vec{\tau}_\lambda(\lambda_i)\|$ and linearized $\|\hat{\tau}_\lambda(\lambda_i)\|$ step responses **(b)** R^2_λ error for linearized response

Figure 3.20: Inner ring induced torque responses for $\Delta\lambda_i$

Plotted in Fig:3.20a are both estimated $\|\hat{\tau}_\lambda(\lambda_i)\|$ and simulated $\|\vec{\tau}_\lambda(\lambda_i)\|$ torques, calculated with a nominal constant angular propeller speed $\Omega_i = 6000$ RPM. Increasing positive and negative step sizes for changes in $\Delta\lambda_i$ are shown, resulting in greater torque responses. The R^2_λ error between $\|\vec{\tau}_\lambda(\lambda_i)\|$ and $\|\hat{\tau}_\lambda(\lambda_i)\|$ is plotted in Fig:3.20b. That difference between $\|\vec{\tau}_\lambda(\lambda_i)\|$ and $\|\hat{\tau}_\lambda(\lambda_i)\|$ is precisely the partial derivative contribution $\partial\mathcal{L}_{n/m}/\partial\vec{\mathbf{u}}$. Or rather mathematically:

$$R^2_\lambda \triangleq \|\vec{\tau}_\lambda(\lambda_i)\| - \|\hat{\tau}_\lambda(\lambda_i)\| \tag{3.122a}$$

$$= \left\| \left(\frac{d}{dt} \left(\frac{\partial \mathcal{L}_{n/m}}{\partial \dot{\vec{\mathbf{u}}}} \right) \right) \right\| - \left\| \left(\frac{d}{dt} \left(\frac{\partial \mathcal{L}_{n/m}}{\partial \dot{\vec{\mathbf{u}}}} \right) - \frac{\partial \mathcal{L}_{n/m}}{\partial \vec{\mathbf{u}}} \right) \right\| \tag{3.122b}$$

$$\therefore R^2_\lambda = \left\| \frac{\partial \mathcal{L}_{n/m}}{\partial \vec{\mathbf{u}}} \right\| \tag{3.122c}$$

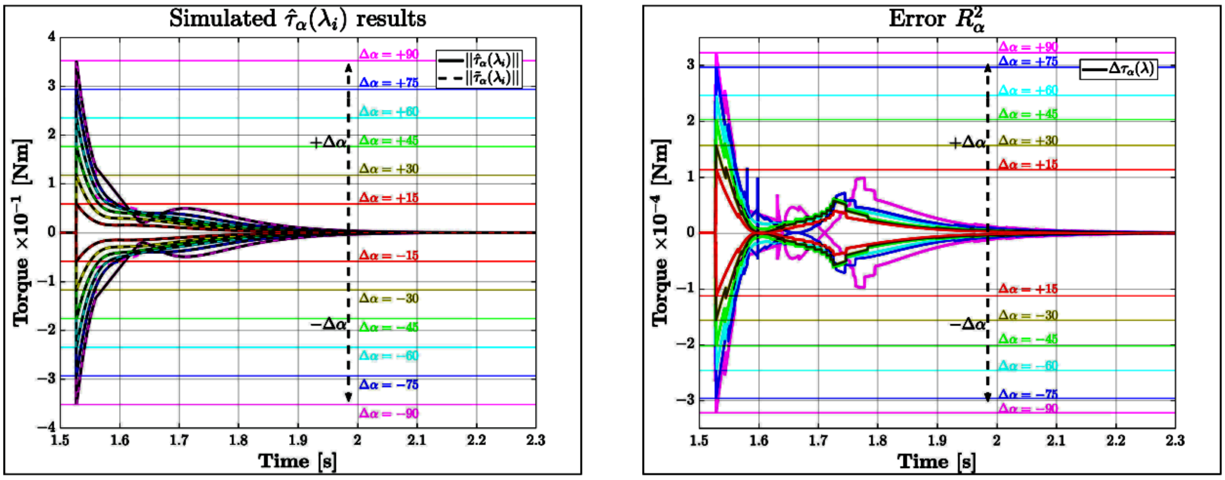
The simulation for R^2_λ suffered from tolerance errors within the MatLab integral approximator, this was as a result of the small deviations which were being calculated. Despite that; the differences, using the inertial matrices and dimensions defined for the prototype in Sec:2.3, were typically in the order of $\times 10^{-5}$ Nm for steps in $\Delta\lambda_i$.

Only for large angular changes in $\Delta\lambda_i$ does the approximation begin to deteriorate. Mostly both $\hat{\tau}_\lambda(\lambda_i)$ and $\vec{\tau}_\lambda(\lambda_i)$ were three orders of magnitude greater than their errors; torques were in the range of $\times 10^{-2}$ Nm. The error for when $\lambda_i = \pi/2$ was not included in Fig:3.21b because it was the only error which did not fit on the $\times 10^{-5}$ Nm scale, being an order of magnitude greater.

The same process was then applied to the middle ring Lagrangian $\mathcal{L}_{m/p}$ relative to the intermediate frame $\mathcal{F}^{M_i''}$, from Eq:3.112d to evaluate $\vec{\tau}_\alpha(\lambda_i, \alpha_i)$. Those results are plotted collectively in Fig:3.21. Note that both $\|\vec{\tau}_\alpha(\lambda_i, \alpha_i)\|$ and $\|\hat{\tau}_\alpha(\lambda_i, \alpha_i)\|$ are plotted, <u>not the generalized torques</u> $\vec{\mathbf{V}}(\lambda_i, \alpha_i)$ acting on the system. The generalized torque response $\vec{\mathbf{V}}(\lambda_i, \alpha_i)$ from Eq:3.113 and expanded in Eq:3.85g <u>includes the inner ring's energy contribution</u> $\vec{\mathbf{U}}(\lambda_i)$ or $\vec{\tau}_\lambda(\lambda_i)$; whilst $\vec{\tau}_\alpha(\lambda_i, \alpha_i)$ and $\hat{\tau}_\alpha(\lambda_i, \alpha_i)$ do not. From Eq:3.86, $\vec{\tau}_\alpha(\lambda_i, \alpha_i)$ is defined as a function of $\vec{\mathbf{V}}(\lambda_i, \alpha_i)$ and $\vec{\mathbf{U}}(\lambda_i)$:

$$\vec{\tau}_\alpha(\lambda_i) \triangleq \vec{\mathbf{V}}(\lambda_i, \alpha_i) - R_y(\lambda)\vec{\mathbf{U}}(\lambda_i) \tag{3.123}$$

Plots for the net generalized torque response $\vec{\mathbf{V}}(\lambda_i, \alpha_i)$ acting on the system with combined changes for $\Delta\alpha_i$ and $\Delta\lambda_i$ are included in App:C.3. Moreover only a constant value for the λ_i servo position was used for the tests in Fig:3.21. The same constant propeller speed $\Omega_i = 6000$ RPM was used together with a constant $\lambda_i = 0$ rad for the inner ring's servo, maintaining a constant inner ring inertia contribution.



(a) $\|\vec{\tau}_\alpha(\lambda_i, \alpha_i)\|$ and linearized $\|\hat{\tau}_\alpha(\lambda_i, \alpha_i)\|$ **(b)** R^2_α error for linearized responses

Figure 3.21: Middle ring induced torque responses for $\Delta\alpha_i$

The initial torque spike in $\vec{\tau}_\alpha(\lambda_i, \alpha_i)$ shown in Fig:3.21a is as a result of the significantly larger rotational inertia, $J_p(\lambda_i)$ for the entire motor module in Eq:2.24b, encountered by a second order angular acceleration $\ddot{\alpha}_i$. That response is depreciated in the case of the inner ring for $\vec{\tau}_\lambda(\lambda_i)$ because of how much smaller that rotational inertia, J_n for the inner ring only in Eq:2.21, physically is. As per power calculations for the servo using parameters from Sec:2.4.1; the servo's angular acceleration is rate limited to 472 rad.s^{-2}, but neither tests come close to reaching saturation.

The second torque peak which begins to manifest in the inner ring response for $\vec{\tau}_\lambda(\lambda_i)$ shown in Fig:3.20a is as a result of the angular velocity rate limit $\dot{\lambda}_{max} = 7.4799$ rad.s^{-1} being encountered. The same limit is encountered for the middle ring response $\vec{\tau}_\alpha(\lambda_i, \alpha_i)$ in Fig:3.21a but is far less significant in relation to the initial second order acceleration torque peak. The deviation between $\|\vec{\tau}_\alpha(\lambda_i, \alpha_i)\|$ and $\|\hat{\tau}_\alpha(\lambda_i, \alpha_i)\|$ is only of the order of 10^{-4} Nm whilst typical induced torque values are again three orders of magnitude greater, or of the order 10^{-1} Nm. Only at large angular changes for either $\Delta\lambda_i$ or $\Delta\alpha_i$ do the simplifications proposed begin to deteriorate.

Note that gravitational torque contributions are not included in either of the above tests for $\vec{\tau}_\lambda(\lambda_i)$ in Eq:3.75 or $\vec{\tau}_\alpha(\lambda_i, \alpha_i)$ in Eq:3.86. Plots in both Fig:3.20a and Fig:3.21a show the linearizations applied in Eq:3.114 hold true for most $\Delta\lambda_i$ or $\Delta\alpha_i$ steps; typically having an error three orders of magnitude smaller than the induced response considered. The control loop will only ever be dealing with minor step size changes for servo positions and so, the linearization is an appropriate one that will reduce interval computational complexity.

Dynamic model verification

In spite of the rigorous mathematical approached applied to the multibody system above, physical corroboration of the proposed model(s) is still required. The systems described in Eq:3.75 for $\vec{\tau}_\lambda(\lambda_i)$, Eq:3.86 for $\vec{\tau}_\alpha(\lambda_i, \alpha_i)$ and Eq:3.109b for $\vec{\tau}_b(u)$ require further verification before an accurate and reliable simulation can be constructed based upon them. Two test rigs were designed and constructed (Fig:3.22 and Fig3.24) to physically measure the induced torques in question. The first test rig recreates the relative motion of the inner ring actuated by the λ_i servo. Similarly the second test platform mimics the middle ring's response when driven by the outer α_i servo.

The net body response, $\vec{\tau}_b(u)$ relating to net angular body velocity $\vec{\omega}_b$ in Eq:3.109b, is harder to recreate on an isolated test rig. Such results are only discussed in the context of simulation. Considering first the inner most ring assembly; Fig:3.22 shows the test rig used to isolate and measure $\vec{\tau}_\lambda(\lambda_i)$ responses to $\Delta\lambda_i$ rotations. The inner ring is supported by two bearing assemblies; an extended shaft in the $-\hat{X}_{M_i}$ direction connects the inner ring to the driving servo block.

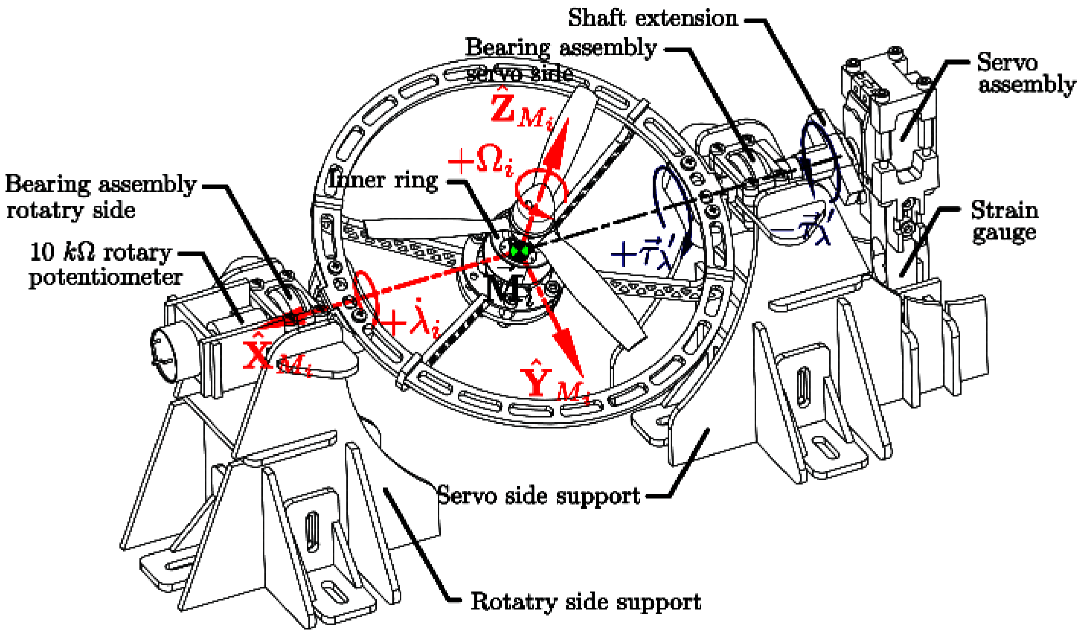


Figure 3.22: Inner ring torque test rig

Physical rotational torque $\vec{\tau}_\lambda(\lambda_i)$ is transferred through the shaft extension from the servo to the inner ring. The servo block is secured only by a vertically aligned and calibrated strain gauge (App:B.3). Deflection of the strain gauge is then proportional to the torque applied by the servo to rotate the inner ring structure. It is important to mention that, whilst the bearing assembly facilitates the transfer of the servo's rotational torque, the assembly isolates only the $\hat{X}_{M_i'}$ component of the induced torque. If $\vec{\tau}_\lambda'$ is the deflection torque physically measured; its relationship with the induced torque vector $\vec{\tau}_\lambda(\lambda_i)$ is given by:

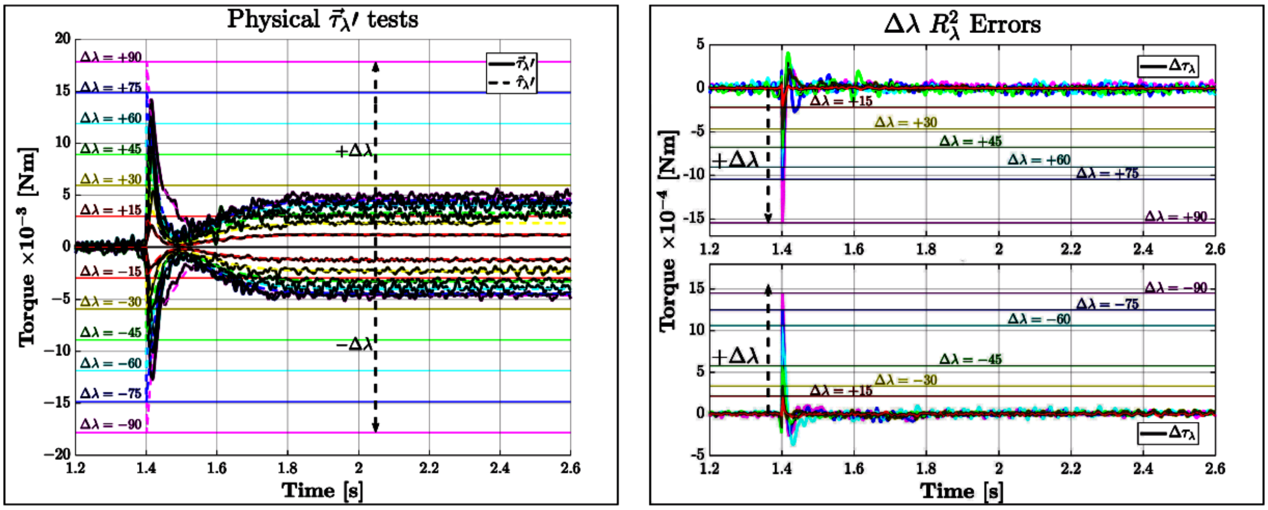
$$\vec{\tau}_\lambda' = \vec{\tau}_\lambda(\lambda_i) \cdot \hat{X}_{M_i'} \qquad \in \mathcal{F}^{M_i'} \tag{3.124}$$

One final thing to consider is that the modelled equation for $\vec{\tau}_\lambda(\lambda_i)$, previously in Eq:3.75, *does not* account for the gravitational torque from an eccentric center of gravity (Fig:2.12) or induced aerodynamic torque about the propellers hub (Fig:3.7 and Eq:3.31b). The derivations earlier in Sec:3.4.1 introduce net gravitational torque for an effective center of gravity $\vec{r}_g$ into Eq:3.109a. Moreover aerodynamic torque $\vec{H}(\Omega_i)$ about the propeller's rotational axis is to be included as an additive term. The torque response $\vec{\tau}_\lambda(\lambda_i)$ opposed to changes of $\Delta\lambda_i$, and hence the servo's acceleration $\ddot{\lambda}_i$ is then, from Eq:3.75 with introduced gravitational and aerodynamic torque components relative to the middle ring frame $\mathcal{F}^{M_i'}$:

$$\begin{aligned}\vec{\tau}_\lambda(\lambda_i) = \big(\dot{J}_r'\big)\vec{\Omega}_i' + \big(\dot{J}_n'\big)\dot{\vec{\lambda}}_i + \big(J_r'\big)\dot{\vec{\Omega}}_i' + \big(J_n'\big)\ddot{\vec{\lambda}}_i + \dot{\vec{\lambda}}_i \times \big(J_r'\big)\vec{\Omega}_i' + \dot{\vec{\lambda}}_i \times \big(J_n'\big)\dot{\vec{\lambda}}_i \\ R_x(\lambda)\big(H(\Omega_i)\cdot \hat{Z}_{M_i}\big) + m_{\mathrm{n}}\big(R_x(\lambda)C.M_{\mathrm{n}}\big) \times \vec{G}_{M_i'} \qquad \in \mathcal{F}^{M_i'}\end{aligned} \tag{3.125}$$

The term $m_{\mathrm{n}}\big(R_x(\lambda)C.M_{\mathrm{n}}\big) \times \vec{G}_{M_i'}$ is the gravitational torque from the rotated center of mass, $C.M_{\mathrm{n}}$; first defined in Eq:2.29d. The torque $H(\Omega_i)\cdot\hat{Z}_{M_i}$ is the scalar projection of aerodynamic torque from Fig:3.7b onto the propeller's $\hat{Z}_{M_i}$ axis, rotated onto the middle ring $\mathcal{F}^{M_i'}$ frame. Note the strain gauge's measured response encountered will be the negative torque response $-\vec{\tau}_\lambda(\lambda_i)$.

The plot illustrated in Fig:3.23a shows tests for the inner ring torque response at increments of relative servo step sizes: $\Delta\lambda_i = \pm[1/12\pi,\ 2/12\pi\ \ldots\ 5/12\pi,\ 6/12\pi]$. A constant propeller rotational speed $\Omega_i = +6000$ RPM was used. Step changes in the propeller's speed that manifest as a gyroscopic cross products in a perpendicular axis but will not affect the projected $\hat{X}_{M_i'}$ torque $\vec{\tau}_\lambda'$ from Eq:3.124. As per convention, in the plot Fig:3.23a, $\vec{\tau}_\lambda'$ represents the ***physically measured*** torque on the test rig illustrated in Fig:3.22 and $\hat{\tau}_\lambda'$ is the expected ***torque estimate*** calculated from Eq:3.125. Both torques are the projected $\hat{X}_{M_i'}$ components of the induced torque vector.



(a) Physical induced $\vec{\tau}_\lambda{}'$ torque

(b) R^2_λ errors for $\vec{\tau}_\lambda{}'$

Figure 3.23: Inner ring test rig response

The error between the physically measured $\vec{\tau}_\lambda{}'(\lambda_i)$ and modelled $\hat{\tau}_\lambda{}'(\lambda_i)$ torques are shown in Fig:3.23b. Peak induced torque as a result of a commanded rotation $\Delta\lambda_i$ increases proportionally with that step size. The co-axial support bearings on the test rig, despite being de-greased and cleaned ultrasonically, still dampened the faster elements of the transient torque response; moreover the overall small magnitude of measured signals meant samples were susceptible to vibration noise transformed through the mechanical structure. There is, however, a clear correlation between the simulated and physically measured signal. Within a margin of error, and considering the tolerances of the test rig, such step changes corroborate the proposed inner ring model in Eq:3.125.

Verification of the dynamics for the middle ring response requires more in-depth discussion. Unlike the inner ring's response, described in Eq:3.125; the middle ring's torque $\vec{\tau}_\alpha(\lambda_i, \alpha_i)$ from Eq:3.86 is ***not equivalent*** to the generalized torque response acting on the middle ring system $\vec{\mathbf{V}}(\lambda_i, \alpha_i)$, Eq:3.85g. As mentioned previously $\vec{\mathbf{V}}(\lambda_i, \alpha_i)$ includes a transformed component of the inner ring's generalized response $R_y(\alpha)\vec{\mathbf{U}}(\lambda_i)$ from Eq:3.74a, whilst the *servo response torque* $\vec{\tau}_\alpha(\lambda_i, \alpha_i)$ ***does not***...

To differentiate the servo's response torque $\vec{\tau}_\alpha(\lambda_i, \alpha_i)$ and the physical (generalized) torque being considered and tested here, $\vec{\Gamma}_\alpha(\lambda_i, \alpha_i)$ is used to refer to the induced torque response from the middle ring assembly's net rotation. That torque is the measured component of the middle ring response and equivalent to the generalized torque response. Reiterating the equation for the expected generalized torque $\vec{\mathbf{V}}(\lambda_i, \alpha_i)$ from Eq:3.85g, now with an included gravitational and aerodynamic torque components and induced torques as a result of the inner ring's rotation:

$$\begin{aligned}\vec{\Gamma}_\alpha(\lambda_i, \alpha_i) = R_y(\alpha)\vec{\mathbf{U}}(\lambda_i) + (\dot{J}'_p)\dot{\vec{\alpha}}_i + \Big(\dot{J}''_n - R_y(\alpha)(\dot{J}'_n)R_y^{-1}(\alpha)\Big)\dot{\vec{\lambda}}'_i + \Big(\dot{J}''_r - R_y(\alpha)(\dot{J}'_r)R_y^{-1}(\alpha)\Big)\vec{\Omega}''_i \\ + (J'_{\mathrm{p}})\ddot{\vec{\alpha}}_i + \dot{\vec{\alpha}}_i \times \Big((J'_p)\dot{\vec{\alpha}}_i + (J''_n)\dot{\vec{\lambda}}'_i + (J''_r)\vec{\Omega}''_i\Big) + R_y(\alpha)R_x(\lambda)\big(Q(\Omega_i)\cdot\hat{Z}_{M_i}\big) + m_{\mathrm{p}}C.M''_{\mathrm{p}}(\alpha_i, \lambda_i) \times \vec{G}_{M''_i} \\ = \vec{\mathbf{V}}(\lambda_i, \alpha_i) \qquad \in \mathcal{F}^{M''_i} \qquad (3.126)\end{aligned}$$

Where the term $C.M''_{\mathrm{p}}(\alpha_i, \lambda_i)$ is the net rotated center of gravity for the entire motor module as a function of both servo positions:

$$C.M''_{\mathrm{p}}(\alpha_i, \lambda_i) = \frac{m_{\mathrm{n}}R_y(\alpha)R_x(\lambda)C.M_{\mathrm{n}} + m_{\mathrm{m}}R_y(\alpha)C.M_{\mathrm{m}}}{m_{\mathrm{m}} + m_{\mathrm{n}}} \qquad (3.127)$$

With m_{m} and m_{n} being inner and middle ring structure's respective masses, $m_{\mathrm{m}} = 98$ g and $m_{\mathrm{n}} = 92$ g from Sec:2.3. Fig:3.24 shows the test rig used to measure torque responses for the motor module assembly which containing both inner and middle ring assemblies. The inner ring servo λ_i was tested both at a constant $\lambda_i = 0$ and at intervals for steps of equivalent in inner and middle ring servo angles.

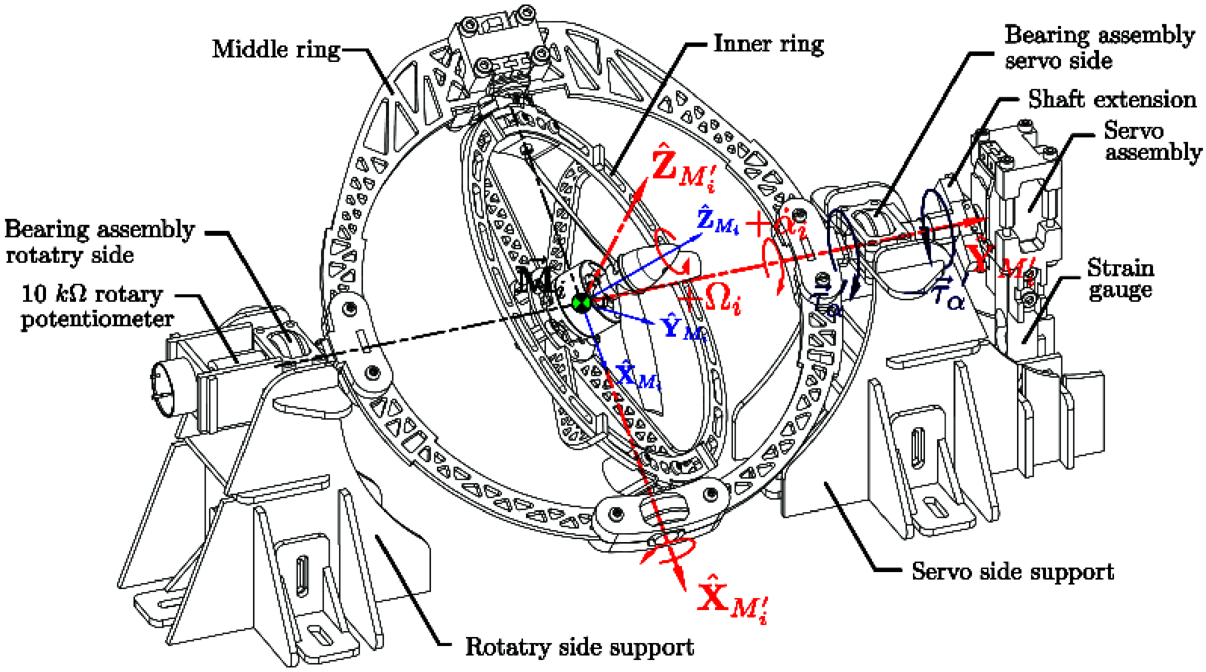
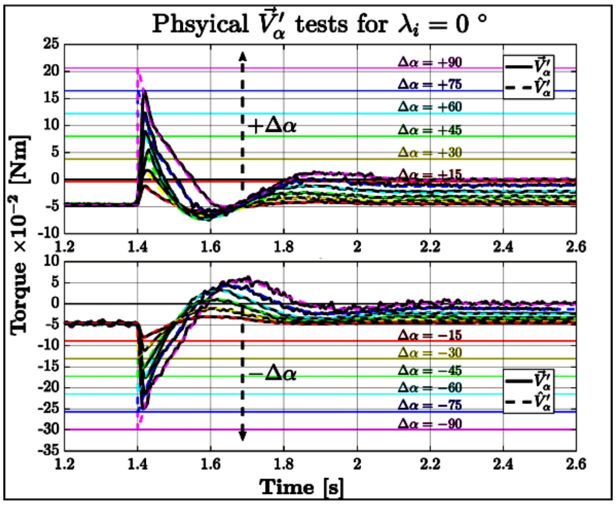


Figure 3.24: Middle ring torque test rig

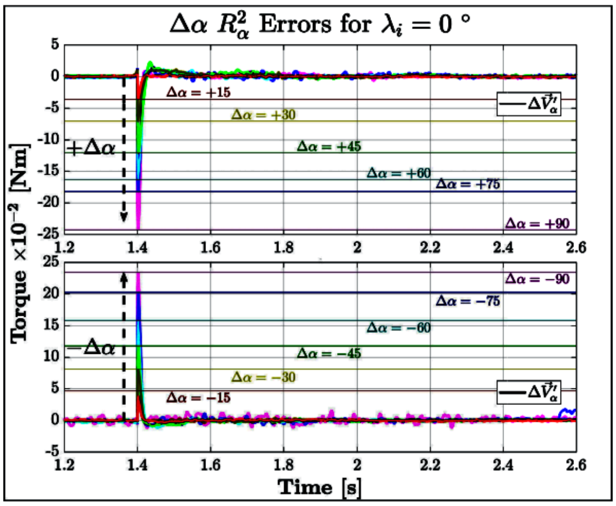
The middle ring servo α_i applies an accelerating torque $\vec{\Gamma}_\alpha(\lambda_i, \alpha_i)$ to the assembly; the test rig isolates only the $\hat{Y}_{M_i''}$ component of that torque. Because the strain gauge encounters only that axial deflection, it then deflects proportionally to the physical torque :

$$\vec{\Gamma}_\alpha{}'(\lambda_i, \alpha_i) = \vec{\Gamma}_\alpha(\lambda_i \alpha_i) \cdot \hat{Y}_{M_i''} \qquad \in \mathcal{F}^{M_i''} \tag{3.128}$$

Furthermore, the inner servo's torque contribution to Eq:3.126, or $R_y(\alpha)\vec{\mathbf{U}}(\lambda_i)$, is small for any case where the propeller's rotational speed and the inner ring's servo speed are both roughly constant; $\dot{\Omega}_i \approx 0$ and $\dot{\lambda}_i \approx 0$. Fig:3.25a plots results for measured torque $\vec{\Gamma}_\alpha{}'(\lambda_i, \alpha_i)$ and expected torque estimate $\hat{\Gamma}_\alpha{}'(\lambda_i, \alpha_i)$ for a constant inner ring servo position $\lambda_i = 0$. Again the propeller's rotational speed was kept constant at $\Omega_i = +6000$ RPM. The error deviation between the two measured and estimated torques is shown in Fig:3.25b, with larger fast torque spikes leading to damped errors of greater magnitude. Steps performed in Fig:3.25 at intervals $\Delta\alpha_i = \pm[1/12\pi,\ 2/12\pi\ \ldots\ 5/12\pi,\ 6/12\pi]$ simply verify the middle ring's inertial contribution to the model. With no inner ring servo velocity or $\dot{\lambda} \neq 0$, the complex dynamics are not completely present. It is however worth noting the dis-symmetry in the shape of the torque's positive and negative responses resulting from unsymmetrical inertias in J_p from Eq:2.24b...



(a) Test rig results for $\hat{\Gamma}_\alpha$

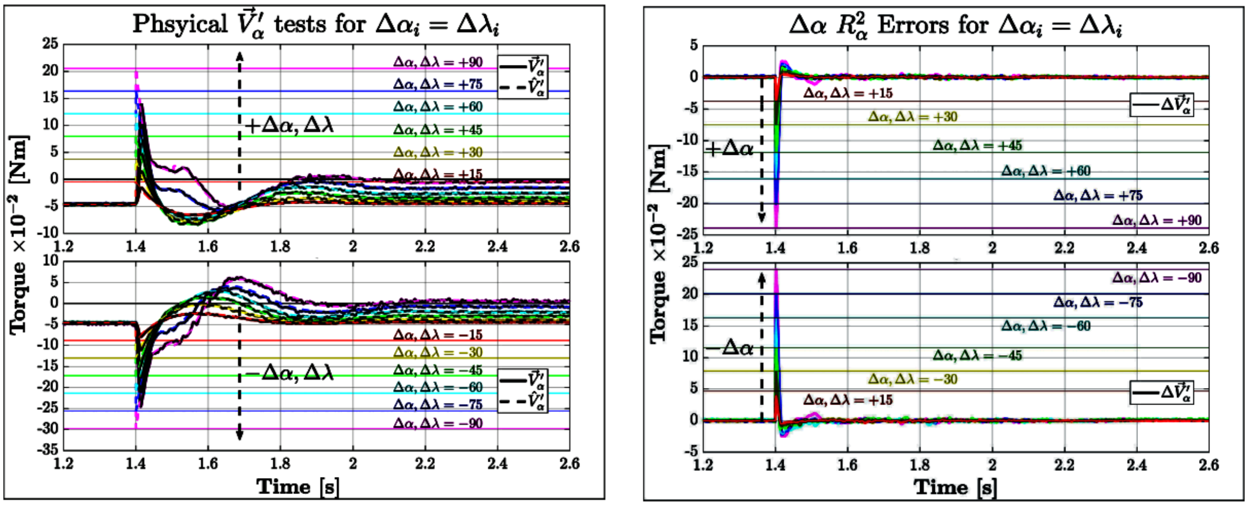


(b) Errors for R_α^2

Figure 3.25: Middle ring response

The larger inertia being rotated by the α_i middle ring servo contributes towards a greater initial torque spike as a result of the angular acceleration from $\ddot{\alpha}_i$, in the order of $\times 10^{-1}$ Nm. The damping effect applied by the support bearing is more pronounced in the middle ring case, producing errors with sizable magnitudes in Fig:3.25b. Without introducing a step for the inner ring λ_i, the response in Fig:3.25 is mostly a scaled version of the inner ring response in Fig:3.23…

Finally, testing combined rotations of λ_i and α_i stepped together, Fig:3.26 shows the manifestation of the complex dynamics involved in a single motor module's combined actuator action. Each interval step is performed with equal servo step sizes; $\Delta\lambda_i = \Delta\alpha_i$ for $\lambda_i, \alpha_i \in \pm[1/12\pi,\ 2/12\pi\ \ldots\ 5/12\pi,\ 6/12\pi]$. Still using a constant propeller speed $\Omega_i = 6000$ rpm, the introduction of gyroscopic torque begins to affect the step response shown in Fig:3.26a. As λ_i and α_i approach $\pi/2$ rad, the propeller's rotational aerodynamic torque begins to make a contribution towards Eq:3.126 as its rotational axis aligns with the measurement axis $\hat{Y}_{M_i''}$.



(a) Test rig results for $\hat{\Gamma}_\alpha$ with $\Delta\alpha = \Delta\lambda$ **(b)** Errors for R_α^2

Figure 3.26: Combined middle ring response

The unsymmetrical inertia of a static motor module, due to the unbalanced servo weight in Fig:2.13, skews the response torque; shown in previous tests in Fig:3.25a. Combining both servos to be actuated at the same time further skews the torque response curve. The induced gyroscopic product of both rotations both constructively and destructively affects Eq:3.126. Only positive $\Delta\alpha_i = \Delta\lambda_i$ tests were performed, conversely $\Delta\alpha_i = -\Delta\lambda_i$ would have a reciprocal effect. Again the fast initial torque spike is damped by the test set-up bearings; resulting in an initial error plotted in Fig:3.26b, which subsequently reduces very quickly. Because the inner ring has a center of gravity, $C.M_\mathrm{n}$ in Eq:2.29b, very close to the module's center of rotation, steady state torque offsets from gravitational torque contributions to $\vec{\Gamma}_\alpha{}'(\lambda_i, \alpha_i)$ are almost independent of the inner ring λ_i servo position.

Each of the above step tests in Fig:3.23,Fig:3.25 and Fig:3.26 were each performed three times and the resultant measured torques were averaged over those three tests. What plotted are the ten sample moving averages of those combined data sets from the three independent tests for each angular step.

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 [71, 143] coordinate system in lieu of the cartesian response equations developed above. However this was not implemented in Ch:6 and remains open to further testing and simulation…

3.5 Consolidated Model

Reiterating the different responses detailed above and consolidating the state equations from Eq:3.10a-3.10d. Then lifting the attitude states to $\mathbb{Q} \in \mathbb{R}^4$ using quaternions. Also introducing the nonlinear inertial and gyroscopic responses to induced perturbations, $\vec{\tau}_\lambda(\lambda_i)$ and $\vec{\tau}_\alpha(\lambda_i, \alpha_i)$ from Eq:3.75 and Eq:3.86 respectively, with nonlinear inertial matrix terms $J_b(u)$ from Section:2.3. Net forces and torques $\vec{F}_\mu(u)$ and $\vec{\tau}_\mu(u)$ are controllable inputs to be designed by a higher level setpoint tracking controller discussed next in Ch:4. The exact actuator effectiveness and allocation schemes are explored thereafter in Ch:5. The vehicle's inertial position and *body frame* velocity differential equations are:

$$\dot{\vec{\mathcal{E}}}_I = Q_b^* \otimes \vec{v}_b \otimes Q_b \qquad \in \mathcal{F}^I \tag{3.129a}$$

$$\dot{\vec{v}}_b = m_b^{-1}\big(- \vec{\omega}_b \times m_b \vec{v}_b + Q_b \otimes m_b \vec{G}_I \otimes Q_b^* + \vec{F}_\mu(u)\big) \qquad \in \mathcal{F}^b \tag{3.129b}$$

Similarly the vehicle's attitude quaternion rate and angular acceleration are respectively:

$$\dot{Q}_b = \frac{1}{2} Q_b \otimes \vec{\omega}_b \qquad \in \mathcal{F}^I \tag{3.129c}$$

$$\dot{\vec{\omega}}_b = J_b(u)^{-1}\big(- \vec{\omega}_b \times J_b(u)\vec{\omega}_b - \vec{\tau}_b(u) + \vec{\tau}_g + \vec{\tau}_H + \vec{\tau}_\mu(u)\big) \qquad \in \mathcal{F}^b \tag{3.129d}$$

The actuator space u is defined as per Eq:2.18, where each actuator has its own transfer function $C(s)$ described in Sec:2.4.1, leading to an actuator state estimate $\hat{u}$ used for control inputs $\vec{F}_\mu(\hat{u})$ and $\vec{\tau}_\mu(\hat{u})$, later compensation feedback terms.

$$u \triangleq \big[\Omega_1^+,\ \lambda_1,\ \alpha_1,\ \dots\ \Omega_4^-,\ \lambda_4,\ \alpha_4\big] \qquad \in \mathbb{U} \in \mathbb{R}^{12} \tag{3.129e}$$

Control force and torque plant inputs, $\vec{F}_\mu(\hat{u})$ and $\vec{\tau}_\mu(\hat{u})$ respectively, are a combination of Eq:3.15 with three dimensional thrust vectors $\vec{T}(\Omega_i)$ as per the quaternion analogue of Eq:2.17. Both are later abstracted to virtual control inputs later in the control allocation design Ch:5.

$$\vec{F}_\mu(\hat{u}) = \sum_{i=1}^{4} \vec{T}(\Omega_i, \lambda_i, \alpha_i) = \sum_{i=1}^{4} Q_{M_i}^* \otimes T(\Omega_i) \otimes Q_{M_i} \qquad \in \mathcal{F}^b \tag{3.130a}$$

$$\vec{\tau}_\mu(\hat{u}) = \sum_{i=1}^{4} \vec{L}_i \times \vec{T}(\Omega_i, \lambda_i, \alpha_i) = \sum_{i=1}^{4} \vec{L}_i \times \big(Q_{M_i}^* \otimes T(\Omega_i) \otimes Q_{M_i}\big) \qquad \in \mathcal{F}^b \tag{3.130b}$$

The torque term $\vec{\tau}_H$ is the net aerodynamic torque produced by the propeller's rotational velocity, to be compensated for in ***feedback***, and as such is separated from the controllable inputs in Eq:3.130:

$$\vec{\tau}_H = \sum_{i=1}^{4} \vec{H}(\Omega_i, \lambda_i, \alpha_i) = \sum_{i=1}^{4} Q_{M_i}^* \otimes \vec{H}(\Omega_i) \otimes Q_{M_i} \qquad \in \mathcal{F}^b \tag{3.131}$$

Scalar thrust $T(\Omega_i)$ is a function of the propeller's rotational velocity whereas $\vec{T}(\Omega_i, \lambda_i, \alpha_i)$ is that thrust's three dimensional counterpart in $\mathcal{F}^b$. Equivalently $H(\Omega_i)$ is the scalar aerodynamic torque in $\mathcal{F}^{M_i}$ about each motor's rotor $\hat{Z}_{M_i}$-axis, whereas $\vec{H}(\Omega_i, \lambda_i, \alpha_i)$ is the torque vector counterpart in $\mathcal{F}^b$. Both thrust and aerodynamic torque terms are calculated from their respective coefficients (plotted in Fig:3.5):

$$\vec{T}(\Omega_i) = C_T(J)\rho\Omega_i^2 D^4 \cdot \hat{Z}_{M_i} \qquad \in \mathcal{F}^{M_i} \tag{3.132a}$$

$$\vec{H}(\Omega_i) = C_P(J)\rho\Omega_i^3 D^5 \big(1/R\Omega_i\big) \cdot \hat{Z}_{M_i} \qquad \in \mathcal{F}^{M_i} \tag{3.132b}$$

Reiterating that Ω_i for aerodynamic calculations in Eq:3.132a and Eq:3.132b has units RPS. The nonlinear torque responses from multibody configuration changes in Eq:3.109b are introduced as terms for feedback compensation, calculated from instantaneous actuator states:

$$\vec{\tau}_b(u) \triangleq \dot{J}_b(u)\vec{\omega}_b + \sum_{i=1}^{4} \Big[\vec{\tau}_\alpha{}'(\lambda_i, \alpha_i) + \vec{\tau}_\lambda{}''(\lambda_i) + \vec{\omega}_b \times \Big(\big(J_p''\big)\dot{\vec{\alpha}}_i' + \big(J_n'''\big)\dot{\vec{\lambda}}_i'' + \big(J_r'''\big)\vec{\Omega}_i'''\Big)\Big] \qquad \in \mathcal{F}^b \tag{3.133}$$

With $\vec{\tau}_\alpha'(\lambda_i, \alpha_i)$ and $\vec{\tau}_\lambda''(\lambda_i)$ both transformed to the body frame $\mathcal{F}^b$. Then including variable gravitational torque as a result of an eccentric center of gravity from Eq:2.34b; also dependent on the vehicles configuration:

$$\vec{\tau}_g = \Delta C.G \times \vec{G}_b = \big(\vec{\mathbf{O}}_b - C.M_{\mathrm{b}}(u)\big) \times \vec{G}_b \qquad \in \mathcal{F}^b \tag{3.134}$$

And finally the vehicles net rotational inertia, aligned and centered with the body frame. That inertia is calculated as a function of all actuator positions; taken from Eq:2.31a and given as:

$$\underset{\vec{\mathbf{O}}_b}{J_b(u)} = \underset{\vec{\mathbf{O}}_b}{J_{\mathrm{y}}'} + \sum_{i=1}^{4} \underset{\vec{\mathbf{O}}_b}{J_{\mathrm{n}}(u_i)} + \sum_{i=1}^{4} \underset{\vec{\mathbf{O}}_b}{J_{\mathrm{m}}(u_i)} \qquad u \in \mathbb{U} \tag{3.135}$$

Both attitude (either euler angles $\vec{\eta}$ or quaternions Q_b) and translational position states $\vec{\mathcal{E}}_I$ could indeed be combined into a single state $\vec{\mathbf{x}}_b$. That could then be used for a complete state feedback control law which could potentially exploit or linearize the cross-coupling between the angular and translational plants. Such an approach would, however, dramatically increase the complexity in tuning actual control parameters (see Sec:6.2). Controllers for attitude and position loops are designed and optimized independently…

Chapter 4

Controller Development

4.1 Control Loop

The control problem is, as outlined in Ch:1, to achieve non-zero setpoint tracking (for both *attitude* and *position* states) on a quadrotor by solving the problem of its inherent underactuation. For the intents and purposes of the subsequent controller development, the plant for some state $\vec{\mathbf{x}}$ is described in the following typical nonlinear state-space form in the time domain:

$$\frac{d}{dt}\vec{\mathbf{x}} = f(\vec{\mathbf{x}}, t) + g(\vec{\mathbf{x}}, \vec{\nu}, t) \tag{4.1a}$$

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

Where the plant's dynamics are governed by state progression $f(\vec{\mathbf{x}}, t)$ and the plant's input response $g(\vec{\mathbf{x}}, \vec{\nu}, t)$ for a given control input $\vec{\nu}$. The latter could take the affine form; $g(\vec{\mathbf{x}}, t)\vec{\nu}$. Setpoint tracking aims for the output to track the plant's state; namely $\vec{y} = c(\vec{\mathbf{x}}, t) \equiv \vec{\mathbf{x}}$. The control problem is then to design a stabilizing control law $\mathcal{H}$ for some error state $\vec{\mathbf{x}}_e$:

$$\vec{\nu}_d = \mathcal{H}(\vec{\mathbf{x}}_e, \dot{\vec{\mathbf{x}}}_e, t) = \mathcal{H}(\vec{\mathbf{x}}_b, \dot{\vec{\mathbf{x}}}_b, \vec{\mathbf{x}}_d, \dot{\vec{\mathbf{x}}}_d, t) = \begin{bmatrix} \vec{F}_d \\ \vec{\tau}_d \end{bmatrix} \tag{4.2}$$

Such that the controlled plant is asymptotically stabilizing or that $\lim_{t\to\infty} \vec{\mathbf{x}}_e = 0$. Trajectory stability conditions are defined next in Sec:4.3. Note that it is possible to combine attitude and position states into a single common trajectory state such that:

$$\vec{\mathbf{x}}_b = \begin{bmatrix} \vec{\mathcal{E}}_I & Q_b \end{bmatrix}^T \tag{4.3}$$

The body's trajectory is then fully described by $\vec{\mathbf{x}}_b(t)$. Separate control laws are developed for attitude and position tracking and hence those states are not combined in the context of this control project. However for the purposes of describing the control plant, a single major loop control is structure considered.

Because of the plant's overactuatedness the control loop is split into two blocks; first a higher level *setpoint tracking* controller designs a virtual control input $\vec{\nu}_d$. That being net forces $\vec{F}_d$ and torques $\vec{\tau}_d$ to act on the body. A lower level *allocator* then solves for explicit actuator positions from $\vec{\nu}_d$ to physically actuate that *virtual* control input. The actuator set then implements a commanded control input $\vec{\nu}_c$ through its effectiveness function (Eq:3.132) with an actuator estimate $\hat{u}$ subject to transfer functions $C(s)$:

$$\vec{\nu}_c = B(\vec{\mathbf{x}}, \hat{u}, t) = \begin{bmatrix} \vec{F}_c(u) & \vec{\tau}_c(u) \end{bmatrix}^T \tag{4.4}$$

The allocator solves for commanded actuator values u_c such that $\vec{\nu}_c \to \vec{\nu}_d$. That allocation function, $B^\dagger$, can be *roughly* referred to as the effectiveness inverse:

$$u_c = B^\dagger(\vec{\mathbf{x}}, \vec{\nu}_d, t) \qquad \in \mathbb{U} \tag{4.5}$$

This chapter derives higher level controllers for $\vec{\nu}_d = \mathcal{H}(\vec{\mathbf{x}}_e, \dot{\vec{\mathbf{x}}}_e, t)$; allocation rules are discussed next in Ch:5. A collection of attitude and position controllers are presented here whose stability is proven with Lyapunov theorem [19, 61, 114]. Each controller is compared in the context of an overactuated quadrotor plant, similarly a series of allocation schemes are presented. For the most part, it is assumed errors between commanded actuator setpoints u and their state estimates $\hat{u}$ affected by transfer functions $C(s)$ are negligible. That plant dependent error is small and diminishing over time but is later incorporated into adaptive control in Sec:4.6.3. Resultant controller comparisons, their details and efficacy are evaluated subsequently in Ch:6.

A generalized overactuated control loop consists of a series of cascaded control blocks (Fig:4.1). From the trajectory's error state $\vec{\mathbf{x}}_e$, a control law designs a virtual control input $\vec{\nu}_d$ which is applied to the allocation block. The allocation law $B^\dagger(\vec{\mathbf{x}}, \vec{\nu}_d, t)$ solves for physical actuator positions $u_c \in \mathbb{U}$. Commanded actuator (*estimate*) positions affect a physical input $\vec{\nu}_c = B(\vec{\mathbf{x}}, \hat{u}, t)$ which is an input applied to the state's dynamics, Eq:4.1. Finally the output tracking state is estimated with some filter paradigm $\hat{\mathbf{x}} = A(\vec{\mathbf{x}}, t)$ which is fed back for error state calculation (Sec:6.9).

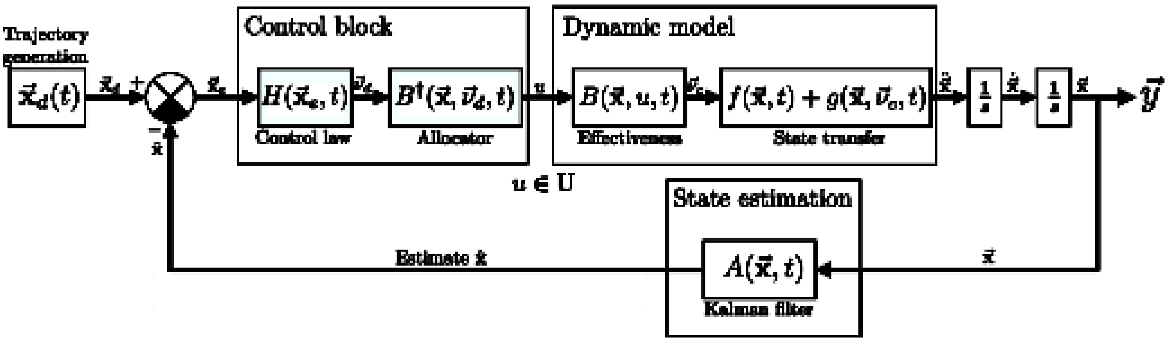


Figure 4.1: Generalized control loop with allocation

Fig:4.1 shows a generalized overactuated control loop's structure; the plant's dynamics from Eq:3.129 include state derivative feedback. Moreover aspects of the state transfer function includes multibody nonlinearities dependent on actuator positions and rates as detailed in Sec:3.4. That generalized case is now refined in the context of an overactuated quadcopter.

4.2 Control Plant Inputs

Control inputs for the differential state equations, from Eq:3.129, have mostly been described with net forces and torques; $\vec{F}_\mu(\hat{u})$ and $\vec{\tau}_\mu(\hat{u})$. The relationship of the *effectiveness function* between each propeller's rotational speed and servo positions with the produced thrust vector is calculated from Eq:3.130.

$$\vec{\nu}_c \triangleq \begin{bmatrix} \vec{F}_\mu(\hat{u}) & \vec{\tau}_\mu(\hat{u}) \end{bmatrix}^T = B(\vec{\mathbf{x}}, \hat{u}, t) \qquad \in \mathbb{R}^6,\ u \in \mathbb{U} \tag{4.6a}$$

$$\vec{F}_\mu(\hat{u}) = \sum_{i=1}^{4} Q^*_{M_i}(\lambda_i, \alpha_i) \otimes \vec{T}(\Omega_i) \otimes Q_{M_i}(\lambda_i, \alpha_i) \qquad \in \mathcal{F}^b \tag{4.6b}$$

$$\vec{\tau}_\mu(\hat{u}) = \sum_{i=1}^{4} \vec{L}_i \times \left(Q^*_{M_i}(\lambda_i, \alpha_i) \otimes \vec{T}(\Omega_i) \otimes Q_{M_i}(\lambda_i, \alpha_i) \right) \qquad \in \mathcal{F}^b \tag{4.6c}$$

As mentioned previously, a higher level controller $\mathcal{H}(\vec{\mathbf{x}}_e, \dot{\vec{\mathbf{x}}}_e, t)$ designs desired net plant inputs $\vec{\nu}_d = \begin{bmatrix} \vec{F}_d & \vec{\tau}_d \end{bmatrix}^T$ whilst a lower level allocator commands actuator positions $u_c = B^\dagger(\vec{\mathbf{x}}, \vec{\nu}_d, t)$. This separation allows for independent comparison of proposed control and allocation laws. However, typical allocation rules like pseudo-inversion require an affine relationship between plant and control inputs, detailed in Sec:1.2.2).

The relationship in Eq:4.6 is not reducible to a single multiplicative relationship with the actuator matrix $u \in \mathbb{U}$. So the effectiveness function needs an extra layer of abstraction to incorporate a multiplicative relationship. Rather than calculating explicit actuator positions directly from $\vec{\nu}_d$; a set of four 3-D thrust vectors $\vec{T}_{1\to 4}$ for each motor module are first calculated.

$$\vec{\nu}_c = \begin{bmatrix} \vec{F}_c(u) \\ \vec{\tau}_c(u) \end{bmatrix} = \begin{bmatrix} \mathbb{I}_{3\times 3} & \mathbb{I}_{3\times 3} & \mathbb{I}_{3\times 3} & \mathbb{I}_{3\times 3} \\ [\vec{L}_1]_\times & [\vec{L}_2]_\times & [\vec{L}_3]_\times & [\vec{L}_4]_\times \end{bmatrix} \begin{bmatrix} \vec{T}_1 & \vec{T}_2 & \vec{T}_3 & \vec{T}_4 \end{bmatrix}^T \tag{4.7a}$$

$$\to \vec{\nu}_c = B'(\vec{\mathbf{x}}, t) \begin{bmatrix} \vec{T}_1 & \vec{T}_2 & \vec{T}_3 & \vec{T}_4 \end{bmatrix}^T \tag{4.7b}$$

Where $[\vec{L}_i]_\times$ is the cross product vector of the i^{th} torque arm from Eq:2.8c. Explicit actuator positions for each module, $[\Omega_i, \lambda_i, \alpha_i]^T$, can then be solved from those thrust vectors $\vec{T}_i$ for $i \in [1:4]$ with some trigonometry, "undoing" the transformation applied in Eq:4.6. That trigonometric inversion is detailed later in Sec:5.2 but is described as the function $R^\dagger$:

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

The generalized control loop illustrated in Fig:4.1 is extended to include the abstracted allocation blocks of Eq:4.7 and Eq:4.8, shown in Fig:4.2. The net control block still solves for the same actuator matrix $u \in \mathbb{U}$. The entire loop accommodates for comparison of various $B^\dagger(\vec{\mathbf{x}}, \vec{\nu}_d, t)$ allocation rules without having to redesign the remainder of the loop's structure.

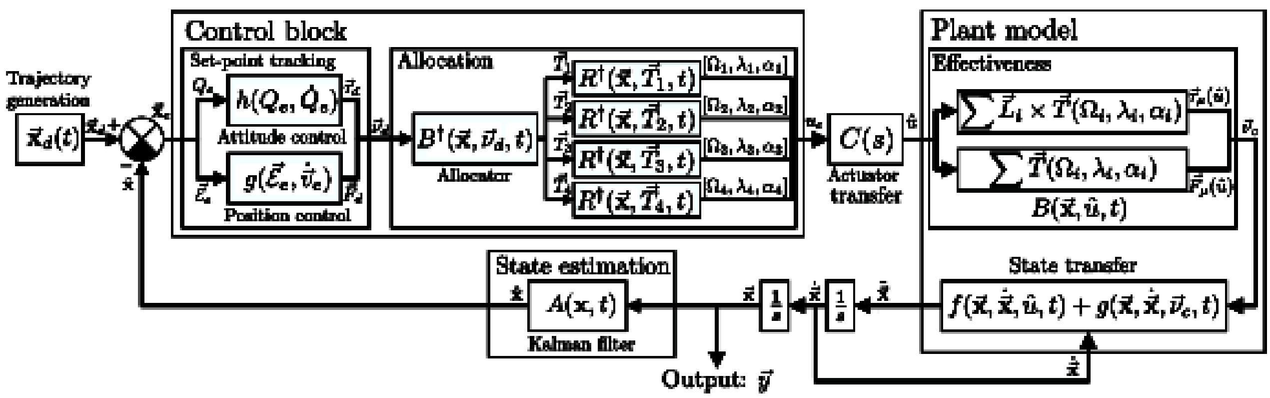


Figure 4.2: Extended control loop with overactuation

Certain blocks in Fig:4.2 use actuator estimates $\hat{u}$ to calculate responses for feedback compensation, other blocks are affected by instantaneous actuator positions u. The two are, for the most part, interchangeable as minor loop transfer dynamics of $C(s)$ in Sec:2.4.1. In summary; each controller designs a net force and torque to act on the body. Allocation rules decompose that virtual input into four separate 3-D thrust vectors, or twelve directional components. The force components are an abstracted allocation layer in place of explicit actuator positions, which are subsequently solved for...

$$B^\dagger(\mathbf{x}, \vec{\nu}_d, t) = \begin{bmatrix} T_{1x},\ T_{1y},\ T_{1z},\ \dots\ T_{4x},\ T_{4y},\ T_{4z} \end{bmatrix}^T \tag{4.9}$$

Each control law is co-dependent on an accompanying allocation algorithm. Traditional control loops (underactuated 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 in Section:4.6 and 4.7 respectively. Thereafter a series of allocation rules are proposed in Ch:5. Although presented independently, the controller and allocation laws are mutually inclusive. The stability of each controller is proven objectively but explicit controller coefficients are optimized in the subsequent Ch:6, in Sec:6.2.

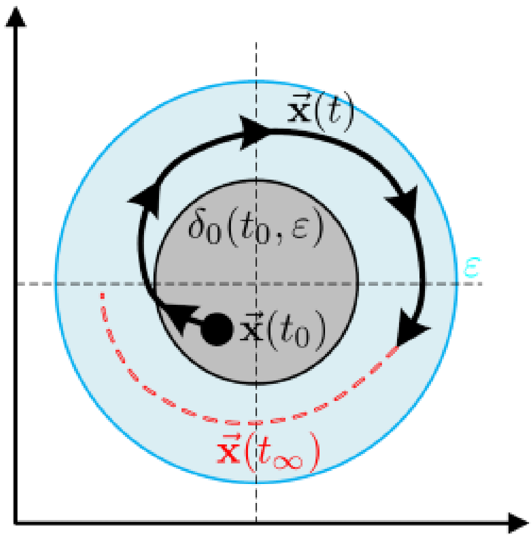
4.3 Stability

Before undertaking the control plant derivations, it is worth outlining definitions of the control objective's stability. The research question aims to achieve non-zero setpoint tracking of state's trajectory. A control loop then aims to *stabilize* the dynamics described previously in Sec:3.5 whilst tracking particular trajectories for attitude and position setpoints, $\vec{\mathbf{x}}_d(t) = [\vec{\mathcal{E}}_d(t)\ Q_d(t)]^T$.

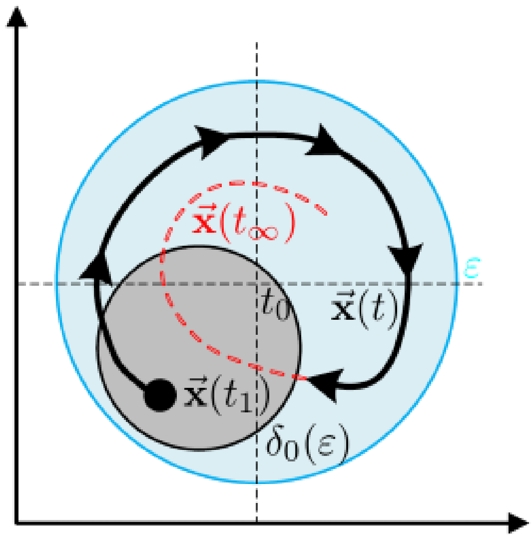
The entire system's control loop was previously detailed in Sec:4.1. Stability in the context of trajectory tracking must first be defined. Generalized trajectory stability definitions are not uncommon in the context of energy based control design, or Lyapunov theorem (Sec:4.4). Stability definitions pertinent to Lyapunov's stability theorem are briefly presented here; the following is adapted from [19, 61]. In general for some autonomous trajectory $\vec{\mathbf{x}}(t)$, an equilibrium point $\vec{\mathbf{x}}(t_0)$ is said to be stable (**S**) if and only if (*iff*) the following is true:

$$\forall \varepsilon > 0,\ \exists\ \delta_0(t_0, \varepsilon) :\ \ \|\vec{\mathbf{x}}(t_0)\| < \delta_0(t_0, \varepsilon) \tag{4.10a}$$
$$\Rightarrow \|\vec{\mathbf{x}}(t)\| < \varepsilon, \quad \forall t \geq t_0 \tag{4.10b}$$

The implication of which is that if, for some initial condition $\vec{\mathbf{x}}(t_0)$ whose magnitude is bound by the manifold $\delta_0(t_0, \varepsilon)$, the entire subsequent trajectory of $\vec{\mathbf{x}}(t)$ is bound from above by some other manifold ε. Basic stability is illustrated in Fig:4.3a for a 2-D trajectory.



(a) Basic stability



(b) Uniform stability

Figure 4.3: Trajectory illustrations for **S** and **US**

An equilibrium point is further said to be uniformly stable (**US**) *iff* for the time $t \in [t_0, \infty)$ the following criteria, being an extension of basic stability, is met:

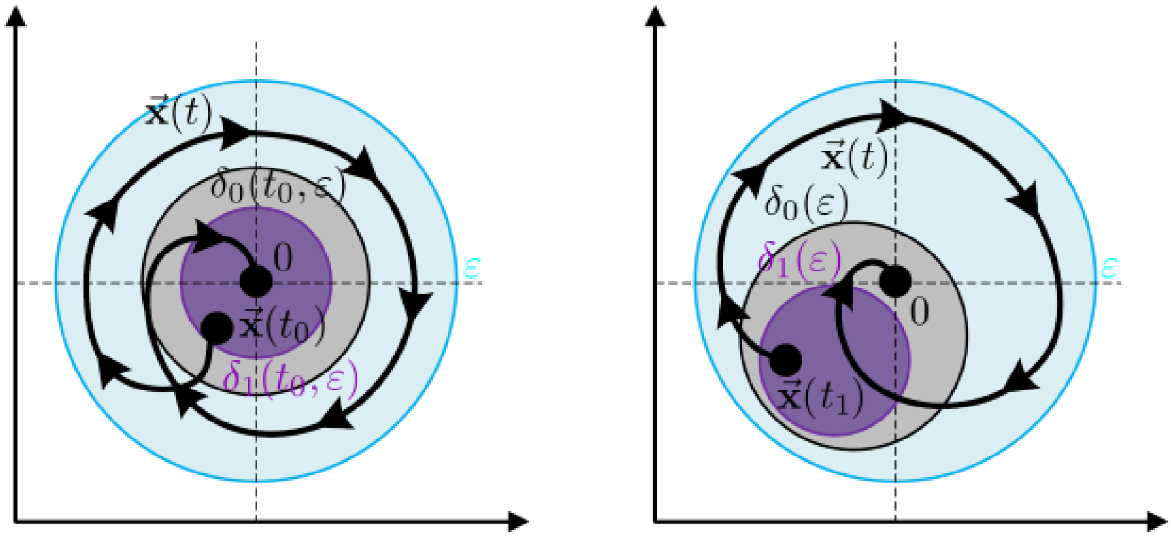
$$\forall \varepsilon > 0,\ \exists\ \delta_0(\varepsilon) > 0 :\ \ \|\vec{\mathbf{x}}(t_1)\| < \delta_0(\varepsilon), \quad t_1 > t_0 \tag{4.11a}$$
$$\Rightarrow \|\vec{\mathbf{x}}(t)\| < \varepsilon, \quad \forall t \geq t_1 \tag{4.11b}$$

US similarly bounds a trajectory from above by ε if the trajectory originates from within $\delta_0(\varepsilon)$. The difference is that the principle trajectory region $\delta_0(\varepsilon)$ is independent of t_0 in the case of **US**. The two surfaces are non-concentric; a **US** trajectory is illustrated in Fig:4.3b. Uniform stability is a subset of general stability, **US** $\subset$ **S**, however the converse is not true. Furthermore **US** is a stronger qualification of stability, each subsequent stability presented represents a stronger assertion of stability.

Extending stability definitions to include settling; an equilibrium point is said to be asymptotically stable (**AS**) *iff* conditions for **S** are met (Eq:4.10) and that the following holds true:

$$\exists\ \delta_1(t_0, \varepsilon) > 0 :\ \ \|\vec{\mathbf{x}}(t_0)\| < \delta_1(t_0, \varepsilon) \tag{4.12a}$$
$$\Rightarrow \lim_{t \to \infty} \|\vec{\mathbf{x}}(t)\| \to 0 \tag{4.12b}$$

This asserts that trajectories originating within some finer region $\delta_1(t_0, \varepsilon)$, being a subset of $\delta_0(t_0, \varepsilon)$, tends to and *asymptotically* settles at the origin. In the case of **AS** the origin is both *stable* and *attractive* (shown in Fig:4.4a). Asymptotic stability is typically the first requirement for any control law, being a stronger stability than both **US** and **S**, typically stabilizing a control setpoint's error…



(a) Asymptotic stability **(b)** Uniform asymptotic stability

Figure 4.4: Trajectory illustrations for **AS** and **UAS**

Uniform asymptotic stability (**UAS**), an extension of asymptotic stability **UAS** $\subset$ **AS**, occurs when the asymptotically stable bound region $\delta_1(\epsilon)$ is independent of the principle starting t_0. An equilibrium point is **UAS** *iff* conditions for **S** are met and that:

$$\exists\ \delta_1(\varepsilon) > 0 : \ \ \|\vec{\mathbf{x}}(t_1)\| < \delta_1(\varepsilon), \quad t_1 \geq t_0 \tag{4.13a}$$

$$\Rightarrow \lim_{t\to\infty} \|\vec{\mathbf{x}}(t)\| \to 0 \tag{4.13b}$$

A uniformly asymptotic equilibrium point implies stability from a non-concentric ball of attraction; settling to the origin (illustrated in Fig:4.4b).

An equilibrium point is regarded as exponentially stable (**UES**) if conditions for **UAS** are met and that there exist $\exists\, a, b, r$ that bound the settling of the trajectory such that:

$$\|\vec{\mathbf{x}}(t, t_0, \vec{\mathbf{x}}_0)\| \leq a\, \|\vec{\mathbf{x}}_0\|\, e^{-bt}, \quad \forall\, \|\vec{\mathbf{x}}_0\| \leq r \tag{4.14}$$

The term $a\,\|\vec{\mathbf{x}}_0\|\,e^{-bt}$ bounds the rate at which the trajectory settles to the origin, illustrated in Fig:4.5a. The initial point of the trajectory, $\vec{\mathbf{x}}_0$ is bound from above by some $r \triangleq \delta_1(\varepsilon)$. Moreover uniform stability is *implied* with exponential stability.

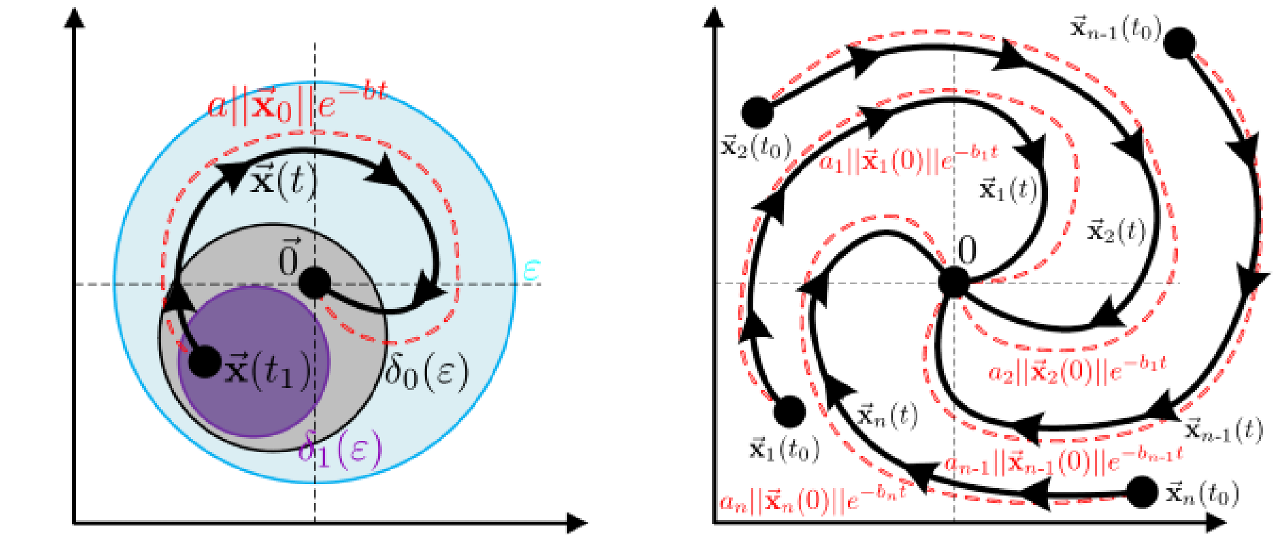
The above definitions of stabilities are only locally defined, and so the stabilities hold true only for local trajectories, only in the case of $\vec{\mathbf{x}}(t_0) \leq \varepsilon$. Extending **UAS** to global uniform asymptotic stability (**GUAS**); the origin's equilibrium point is **GUAS** *iff* conditions for **UAS** are first met, the origin is only the equilibrium point and the asymptotic approach can be extended such that:

$$\exists\ \delta_1(\varepsilon) > 0 : \ \ \|\vec{\mathbf{x}}(t_1)\| < \delta_1(\varepsilon), \quad t_1 \geq t_0 \tag{4.15a}$$

$$\Rightarrow \lim_{t\to\infty} \|\vec{\mathbf{x}}(t)\| \to 0, \quad \forall \vec{\mathbf{x}}(t_0) \tag{4.15b}$$

Similarly exponential stability can extend to the global case, shown in Fig:4.5b, but only *iff* **UES** conditions are first met. In the global case, the origin can be the *only equilibrium point*. Stability from Eq:4.14 is then globally:

$$\|\vec{\mathbf{x}}(t, t_0)\| \leq a\, \|\vec{\mathbf{x}}_0\|\, e^{-bt}, \quad \forall\, \|\vec{\mathbf{x}}_0\| \tag{4.16}$$



(a) Exponential stability **(b)** Global exponential stability

Figure 4.5: Trajectory illustrations for **UES** and **GUES**

Initial trajectory conditions are dropped in Eq:4.16, for any number of trajectories until $\vec{\mathbf{x}}_n(t)$; each trajectory is bound by an exponential $a_n \|\vec{\mathbf{x}}_t(0)\| e^{-b_n t}$. It follows that, irrespective of the starting point $\vec{\mathbf{x}}_n(t_0)$ for the trajectory, the system *always* settles to the origin. **GUES** is the strongest sense of stability and provides insight into the trajectory stabilizing rate. The most desirable control design outcome is a controller which applies globally uniform exponential stability to a plant.

4.4 Lyapunov Stability Theorem

Lyapunov's stability theory is an important aspect of nonlinear controller design. An abundance of literature exists on the subject, included in almost every meritable textbook or control paper. If the reader is unfamiliar with Lyapunov's theorem, [19, 93, 114] each explains the concept in detail. The following is adapted from [19] and [61] and briefly outlines how Lyapunov's stability theorem is used to prove (*global*) asymptotic stability for continuous time invariant systems, linear or otherwise. The theory analyzes a generalized energy function of a system's autonomous trajectory. If the trajectory has a negative energy derivative that implies the system's energy will always dissipate towards a state of zero energy or stable equilibrium point. Lyapunov analysis is a powerful tool for stability verification because the system's trajectory itself need not be explicitly defined for stability to be determined. Proof of Lyapunov's theorem is done with a contradiction disproof and, as such, the theoretical underpinning is somewhat cumbersome.

It is worth reiterating its fundamentals given that backstepping controllers are proposed later in Sec:4.6.3 for attitude control. A backstepping controllers enforce Lyapunov stability criteria onto the system through interative control structure design, [11, 74, 139]. In general, given a nonlinear time invariant system that follows some continually differentiable trajectory $\vec{\mathbf{x}}(t)$, typically the trajectory is going to progress subject to some rule:

$$\dot{\vec{\mathbf{x}}}(t) = f\big(\vec{\mathbf{x}}(t), u\big) \tag{4.17}$$

Then, constructing a generalized positive-definite function generalized energy or *Lyapunov function candidate* (*LFC*) $V(\vec{\mathbf{x}})$ for a trajectory $\vec{\mathbf{x}}(t)$. A positive definite matrix M is defined such that:

$$\mathbf{z}^T M \mathbf{z} \geq 0 \quad \forall \mathbf{z} \tag{4.18}$$

As such an LFC typically, but not exclusively, has the quadratic and positive-definite form with some positive square matrix $P \in \mathbb{R}^{n \times n} > 0$:

$$V(\vec{\mathbf{x}}) = \vec{\mathbf{x}}^T P \vec{\mathbf{x}}, \quad \vec{\mathbf{x}} \in \mathbb{R}^n \tag{4.19}$$

An LFC could simply be positive semi-definite over the trajectory's path, the quadratic form is just convenient for the use of backstepping. From its definition the trajectory Eq:4.17 is continually differentiable; there is then a gradient matrix for each element of $V(\vec{\mathbf{x}})$ in the form:

$$\nabla V(\vec{\mathbf{x}}) \triangleq \left[\frac{\partial V(\vec{\mathbf{x}})}{\partial x_1} \; \frac{\partial V(\vec{\mathbf{x}})}{\partial x_2} \; \ldots \; \frac{\partial V(\vec{\mathbf{x}})}{\partial x_n} \right] \quad \vec{\mathbf{x}} \in \mathbb{R}^n \tag{4.20}$$

The energy function's derivative, otherwise referred to as the *Lie derivative* in some texts; [93,114], is calculated from partial derivatives in Eq:4.20 as follows:

$$\dot{V}(\vec{\mathbf{x}}) \triangleq \nabla V(\vec{\mathbf{x}})^T \frac{d}{dt} f(\vec{\mathbf{x}}) = \frac{\delta V(\vec{\mathbf{x}})}{\delta x_1} \frac{df_1(x_1)}{dt} + \frac{\delta V(\vec{\mathbf{x}})}{\delta x_2} \frac{df_2(x_2)}{dt} + \; \ldots \; + \frac{\delta V(\vec{\mathbf{x}})}{\delta x_n} \frac{df_n(x_b)}{dt} \tag{4.21}$$

Lyapunov's theorem states that *iff* the candidate function $V(\vec{\mathbf{x}})$ is positive definite with $V(\vec{0}) = 0$ and its derivative is strictly negative; $\dot{V}(\vec{\mathbf{x}}) < 0 \;\; \forall \vec{\mathbf{x}}(t) \neq 0$, the system is then asymptotically stable (**AS** from Eq:4.12). Mathematically that means, for any $\vec{\mathbf{x}}(t)$ with $t \geq t_0$:

$$V\big(\vec{\mathbf{x}}(t)\big) = V\big(\vec{\mathbf{x}}(t_0)\big) + \int_{t_0}^{t} \dot{V}\big(\vec{\mathbf{x}}(t)\big).dt \leq V\big(\vec{\mathbf{x}}(t_0)\big) \tag{4.22}$$

Which can be physically interpreted as the system's generalized energy function dissipating, irrespective of the trajectory path taken. With a strictly decreasing energy function, the system will stabilize to a state of zero energy which, naturally, is a stable equilibrium point.

$$\lim_{t\to\infty} \big\| V\big(\vec{\mathbf{x}}(t)\big) \big\| \to 0 \tag{4.23}$$

The trajectory's asymptotic stability can be extended to exponential stability boundedness, such that *iff* the same conditions are met for asymptotic stability in Eq:4.22 and there exists some positive coefficient $\alpha > 0$ such that $\dot{V}(\vec{\mathbf{x}}) \leq -\alpha V(\vec{\mathbf{x}})$. That implies the system is globally exponentially stable as is bound in such a way that:

$$\big\| V\big(\vec{\mathbf{x}}(t)\big) \big\| \leq M e^{-\alpha t/2} \big\| V\big(\vec{\mathbf{x}}(t_0)\big) \big\| \tag{4.24}$$

4.5 Model Dependent & Independent Controllers

Two classes of controllers are included for a full state trajectory tracking control loop; both attitude and position control laws. Attitude setpoint tracking is the primary focus of this research project (Sec:4.6.1) and incorporates a more detailed schedule of controller design and evaluation.

The allocation law combines both virtual control inputs from attitude and position controllers, $\vec{\nu}_d = [\vec{F}_d \; \vec{\tau}_d]^T$, to solve for explicit actuator positions. Controller dependency on the plant's state is as a consequence of the actuator responses and complex inertial dynamics, as derived previously in Sec:3.4.1. Whilst not a prerequisite for stability, plant dependent compensation obviously improves controller performances. Independent and dependent cases are only considered for one type of controller; the most basic case proportional-derivative controller in Section:4.6.2 and tested in Sec:6.3.1. All other control laws compensate for unwanted plant dynamics in a feedback configuration.

The plant dependency makes backstepping controllers an effective controller choice for this dissertation's context. The proposed plant dependent control laws compensate for undesirable dynamics their design, basic PD and 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 (Pseudo or Moore-Penrose inversion) allocation.

4.6 Attitude Control

4.6.1 The Attitude Control Problem

The setpoint tracking control problem ([134]) for the attitude plant is to design a stabilizing control torque $\vec{\tau}_d = h(\vec{\mathbf{x}}_e, \dot{\vec{\mathbf{x}}}_e, t)$ such that for any desired attitude quaternion $\forall\ Q_d \in \mathbb{Q}$ and an instantaneous attitude body quaternion $Q_b \in \mathbb{Q}$ the error state asymptotically stabilizes to the origin; $Q_e \to [1\ \vec{0}\ ]^T$. Or that:

$$\vec{\tau}_d = h(Q_d,\ \dot{Q}_d,\ Q_b,\ \dot{Q}_b) \text{ such that } \lim_{t\to\infty} Q_b \to Q_d \tag{4.25}$$

Quaternion attitude error states are defined as the Hamilton product or *difference* between the desired and instantaneous quaternion attitude states, previously in Eq:3.56. Quaternion error states are multiplicative, in contrast with the subtractive relationship for Euler angle error states. The attitude error state is defined as:

$$Q_e \triangleq Q_b^* \otimes Q_d \tag{4.26}$$

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:3.129d. As such there is an angular rate error:

$$\vec{\omega}_e \triangleq \vec{\omega}_d - \vec{\omega}_b' \qquad \in \mathcal{F}^d \tag{4.27a}$$

The desired angular velocity $\vec{\omega}_d$ is taken with respect to the desired angular attitude frame, and so it must be transformed back onto the existing body frame.

$$\vec{\omega}_e = Q_e^* \otimes \vec{\omega}_d \otimes Q_e - \vec{\omega}_b \qquad \in \mathcal{F}^b \tag{4.27b}$$

Typically for the trajectories generated here only first order setpoints are commanded, hence 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 setpoints to accommodate for higher order state derivative tracking trajectories.

$$\vec{\omega}_e = -\vec{\omega}_b\Big|_{\vec{\omega}_d=\vec{0}} \tag{4.27c}$$

The time derivative for the quaternion error state is calculated from the quaternion derivative definition Eq:3.52. The derivative $\dot{Q}_e$ is then dependent on the angular velocity error and calculated as follows:

$$\dot{Q}_e = \frac{1}{2} Q_e \otimes \vec{\omega}_e = -\frac{1}{2} Q_e \otimes \vec{\omega}_b\Big|_{\vec{\omega}_d=\vec{0}} \tag{4.28}$$

4.6.2 Linear Controllers

PD Controller

The following control law is used as a reference case for comparing the remaining controllers derived. It is a simple proportional-derivative (*PD*) attitude controller, adapted from [41] and following a stability proof similar to the one derived in [134]. An attitude PD controller is proportional only to the *vector quaternion error*. Such that the error is the same dimension as the angular velocity error; $\vec{q}_e \in \mathbb{R}^3$. A PD controller designs the commanded torque input:

$$\vec{\tau}_{PD} = K_d \vec{\omega}_e + K_p \vec{q}_e \qquad \in \mathcal{F}^b \tag{4.29}$$

Where both K_d and K_p are *positive symmetrical* 3×3 gain coefficient matrices to be determined at a later stage. Eq:4.29 neglects the quaternion scalar error and is therefore susceptible to unwinding. Using a positive-definite Lyapunov function candidate V_{PD} for the attitude trajectory:

$$V_{PD}(Q_e, \vec{\omega}_e) = \vec{q}_e^{\,T} \vec{q}_e + (1 - q_0)^2 + \frac{1}{2} \vec{\omega}_e^{\,T} J_b(u) K_p^{-1} \vec{\omega}_e \quad > 0,\ \forall (Q_e, \vec{\omega}_e) \tag{4.30}$$

Note also that $V_{PD}([\pm 1\ \vec{0}]^T, \vec{0}) = 0$, making it a suitable LFC. Exploiting the unit quaternion's inherent magnitude property, it follows that:

$$\|Q\| = \vec{q}^{\,T}\vec{q} + {q_0}^2 = \vec{q}^{\,2} + {q_0}^2 = 1 \tag{4.31}$$

Substituting that and the angular velocity's error state, $\vec{\omega}_e = -\vec{\omega}_b$; the proportional derivative LFC in Eq:4.30 reduces to:

$$V_{PD} = \vec{q_e}^{\,2} + \big({q_0}^2 - 2q_0 + 1\big) + \frac{1}{2}\vec{\omega_e}^T J_b(u) K_p^{-1}\vec{\omega}_e \tag{4.32a}$$

$$= 2(1 - q_0) + \frac{1}{2}\vec{\omega_b}^T J_b(u) K_p^{-1}\vec{\omega}_b\Big|_{\vec{\omega}_e = -\vec{\omega}_b} \tag{4.32b}$$

Taking the derivative of that Lyapunov Function candidate then yields:

$$\dot{V}_{PD}(\vec{q}_e, \vec{\omega}_e) = -2\dot{q}_0 + \vec{\omega_b}^T J_b(u) K_p^{-1}\dot{\vec{\omega}}_b \tag{4.33}$$

Recalling the angular velocity differential equation from Eq:3.129d, $\dot{\vec{\omega}}_b$ with a control torque input $\vec{\tau}_{PD}$ from Eq:4.29:

$$\dot{\vec{\omega}}_b = {J_b}^{-1}(u)\big(-\vec{\omega}_b \times J_b(u)\vec{\omega}_b - \vec{\tau}_b(u) + \vec{\tau}_g + \vec{\tau}_H + \vec{\tau}_{PD}\big) \qquad \in \mathcal{F}^b \tag{4.34}$$

Where $\vec{\tau}_H$ is a simplified representation of the net aerodynamic torque experienced by the body from the rotating propellers, from Eq:3.131. Then, using the fact that a quaternion's derivative by definition is:

$$\dot{Q} = \begin{bmatrix} -\frac{1}{2}\vec{q}^{\,T}\vec{\omega} \\ \frac{1}{2}\big([\vec{q}\,]_\times + q_0\mathbb{I}_{3\times3}\big)\vec{\omega} \end{bmatrix} \tag{4.35}$$

Substituting the above into the LFC derivative $\dot{V}_{PD}$ in Eq:4.33 with an expanded $\vec{\tau}_{PD}$ yields:

$$\rightarrow \dot{V}_{PD} = \vec{q_e}^T\vec{\omega}_e + \vec{\omega_b}^T J_b(u) K_p^{-1}\Big(J_b(u)^{-1}\big(-\vec{\omega}_b \times J_b(u)\vec{\omega}_b - \vec{\tau}_b + \vec{\tau}_g + \vec{\tau}_H + K_d\vec{\omega}_e + K_p\vec{q}_e\big)\Big) \tag{4.36a}$$

$$= -\vec{q_e}^T\vec{\omega}_b + \vec{\omega_b}^T\vec{q}_e - \vec{\omega_b}^T K_p^{-1}K_d\vec{\omega}_b + \vec{\omega_b}^T K_p^{-1}\big(-\vec{\omega}_b \times J_b(u)\vec{\omega}_b - \vec{\tau}_b(u) + \vec{\tau}_g + \vec{\tau}_H\big) \tag{4.36b}$$

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:

$$\therefore \dot{V}_{PD} = -\vec{\omega_b}^T K_p^{-1}K_d\vec{\omega}_b + \vec{\omega_b}^T K_p^{-1}\big(-\vec{\omega}_b \times J_b(u)\vec{\omega}_b - \vec{\tau}_b(u) + \vec{\tau}_g + \vec{\tau}_Q\big) \tag{4.36c}$$

Then, as long as $\big(-\vec{\omega}_b \times J_b(u)\vec{\omega}_b - \vec{\tau}_b(u) + \vec{\tau}_g + \vec{\tau}_Q\big) \leq \vec{0}$, some basic stability is guaranteed. 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, $J_b(u)$, is approximately diagonal which, given the inertia ranges from Eq:2.32 and Eq:2.33, is reasonable. Similarly that the angular rate can be made small with appropriately slow trajectory updates such that the torque gyroscopic cross-product is negligible:
$$\big(\vec{\omega}_b \times J_b(u)\vec{\omega}_b\big) \approx \vec{0}$$
2. The actuator rate torque response, $\vec{\tau}_b(u)$, is a second order effect dependent on $d/dt(u)$. Typically the actuator rates are going to be kept small, Fig:3.17 shows torque magnitudes $|\vec{\tau}_b(u)|$ for a typical trajectory. For slow attitude steps those position changes are small enough to be considered negligible. The approximation is made:
$$\vec{\tau}_b(u) \approx \vec{0}$$
3. Finally, for the sake of the stability proof, the eccentric gravitational torque arm is neglected. Such a situation only holds true if $u \approx \vec{0}$ or that servo actuator positions are close to their zero positions.
$$\vec{\tau}_g \approx \vec{0}$$

All of the above assumptions are made under extraneous circumstances and can not be assumed for almost all of the prototype's flight envelope. The plant independent case is considered and simulated in Sec6.3.1 purely for contrition; mainly to demonstrate the need for plant dependent compensation.

If each of the assumptions made hold true, then the Lyapunov function's derivative is approximately negative definite. The stability proof for a very local trajectory is then:

$$\dot{V}_{PD} \approx -\vec{q}_e^{\,T}\vec{\omega}_b + \vec{\omega}_b^{\,T} K_p^{-1}\big(- K_d\vec{\omega}_b + K_p\vec{q}_e\big) \tag{4.37a}$$

$$\rightarrow \dot{V}_{PD} = -\vec{\omega}_b^{\,T} K_p^{-1} K_d \vec{\omega}_b = -K_p^{-1} K_d \, \|\vec{\omega}_b\|^{\,2} \quad < 0, \; \exists \; (K_p^{-1}, K_d) > 0 \tag{4.37b}$$

From Lyapunov stability theorem there then exists the limits for *local* asymptotic stability:

$$\lim_{t\to\infty} \vec{\omega}_e \rightarrow \vec{0} \quad \therefore \quad \lim_{t\to\infty} \vec{\omega}_b \rightarrow \vec{0}^{-} \tag{4.38a}$$

$$\lim_{t\to\infty} \vec{q}_e \rightarrow \vec{0} \;\text{ and }\; \lim_{t\to\infty} (1 - q_0) \rightarrow 0 \tag{4.38b}$$

Hence the quaternion error stabilizes $Q_e \rightarrow [1 \; \vec{0}\,]^T$ as $t \rightarrow \infty$. The stability shown in Eq:4.37b is only local; introducing plant dependent compensation to the PD control law in Eq:4.29 alleviates the stringent requirements on assumptions 1 through 3.

$$\vec{\tau}_{PD} = \underbrace{K_p\vec{q}_e + K_d\vec{\omega}_e}_{Independent} + \underbrace{\hat{\omega}_b \times J_b(\hat{u})\hat{\omega}_b + \vec{\tau}_b(\hat{u}) - \vec{\tau}_g - \vec{\tau}_Q}_{Compensation} \tag{4.39}$$

Obviously controller errors and compensation terms are *state estimates*, where inertias and torque responses are calculated using sampled $\hat{u}$ actuator states. Moreover the quaternion attitude and angular velocity states $\hat{Q}_b$ and $\hat{\omega}_b$ are both estimates and so a small degree of uncertainty exists; robust stability in the case of plant dependent uncertainty is investigated in Sec:6.6 but for now the estimates are *assumed to be error free.*

The resultant stability proof for the plant dependent case Eq:4.39 is much the same as that for the independent controller, Eq:4.29. The same LFC from Eq:4.30 shows that Eq:4.37b holds globally:

$$\rightarrow \dot{V}_{PD} = -\vec{\omega}_b^{\,T} K_p^{-1} K_d \vec{\omega}_b = -K_p^{-1} K_d \, \|\vec{\omega}_b\|^{\,2} < \vec{0}, \quad \forall (Q_e, \vec{\omega}_b), \quad \exists \; (K_p^{-1}, K_d) > 0 \tag{4.40}$$

Note that the inverse qualifier of K_p^{-1} in the above is redundant given that K_p is a symmetrical coefficient matrix. The plant dependent rule is not reliant on the same limiting assumptions needed for independent asymptotic stability to be achieved. Dynamic compensation in Eq:4.39 is simple to implement; especially considering the unwanted dynamics have already been quantified and corroborated in Sec:3.4.2 together with state estimate terms in Sec:2.4.1.

Auxiliary Plant Controller

Expanding on what has, in practice (Table:1.1 from Sec:1.2.1), proven to be a popular and effective controller for attitude stabilization, [131] proposed an auxiliary plant term to a PD attitude controller. Most significantly, the altered PD controller adds auxiliary terms proportional to the quaternion rate error (Eq:3.52). Moreover part of the auxiliary plant is proportional to the *quaternion scalar* q_0, a term that is otherwise neglected in the previous PD control law (Sec:4.6.2) and ensures unwinding is avoided when incorporated. The *auxilliarly* PD control torque is a function of errors states:

$$\vec{\tau}_{XPD} = \underbrace{\Gamma_2\widetilde{\Omega} + \Gamma_3\vec{q}_e - J_b(u)\dot{\widetilde{\Omega}}}_{Independent} + \underbrace{\hat{\omega}_b \times J_b(\hat{u})\hat{\omega}_b + \vec{\tau}_b(\hat{u}) - \vec{\tau}_g - \vec{\tau}_H}_{Compensation} \tag{4.41}$$

Wherein the coefficients Γ_2 and Γ_3 are both diagonal positive coefficient matrices whilst Γ_1, used next in Eq:4.42, is a symmetrical matrix. Each coefficient matrix is explicitly determined later. Auxiliary plants $\widetilde{\Omega}$ and $\dot{\widetilde{\Omega}}$ are defined as follows.

The first auxiliary plant $\bar{\Omega}$ is proportional to the quaternion error and hence its derivative $\dot{\bar{\Omega}}$ is a quaternion rate:

$$\bar{\Omega} \triangleq -\Gamma_1 \vec{q}_e \quad \therefore \quad \dot{\bar{\Omega}} = -\Gamma_1 \dot{\vec{q}}_e \tag{4.42a}$$

$$\rightarrow \dot{\bar{\Omega}} = -\frac{1}{2}\Gamma_1\big([\vec{q}_e]_\times + q_0 \mathbb{I}_{3X3}\big)\vec{\omega}_e \tag{4.42b}$$

$$= \frac{1}{2}\Gamma_1\big([\vec{q}_e]_\times + q_0 \mathbb{I}_{3X3}\big)\vec{\omega}_b\Big|_{\vec{\omega}_e=-\vec{\omega}_b} \tag{4.42c}$$

The second auxiliary plant, $\widetilde{\Omega}$, is proportional to both quaternion vector and angular velocity errors.

$$\widetilde{\Omega} \triangleq \vec{\omega}_e - \bar{\Omega} = \vec{\omega}_e + \Gamma_1 \vec{q}_e \tag{4.43a}$$

$$= -\vec{\omega}_b + \Gamma_1 \vec{q}_e\Big|_{\vec{\omega}_e=-\vec{\omega}_b} \tag{4.43b}$$

Using an LFC similar to the basic V_{PD} function candidate from Eq:4.30, but substituting an auxiliary term $\widetilde{\Omega}$ for the body's angular velocity $\vec{\omega}_b$ into the LFC V_{XPD}.

$$V_{XPD}\big(Q_e,\ \widetilde{\Omega}\big) = \vec{q}_e^{\,T}\vec{q}_e + \big(1-q_0\big)^2 + \frac{1}{2}\widetilde{\Omega}^T\Big(\Gamma_3^{-1}J_b(u)\Big)\widetilde{\Omega} \quad > 0,\ \forall(Q_e, \widetilde{\Omega}) \tag{4.44}$$

At the trajectory's origin the energy function Eq:4.44 is zero, or $V_{XPD}([\pm 1\ \vec{0}]^T, \vec{0}) = 0$. Again using the simplification from a quaternion's inherent properties in Eq:4.17, the LFC from Eq:4.44 then simplifies with the following derivative:

$$V_{XPD} = 2(1-q_0) + \frac{1}{2}\widetilde{\Omega}^T\Big(\Gamma_3^{-1}J_b(u)\Big)\widetilde{\Omega} \tag{4.45a}$$

$$\dot{V}_{XPD} = 2\frac{1}{2}\vec{q}_e^{\,T}\vec{\omega}_e + \frac{1}{2}\dot{\widetilde{\Omega}}^T\Big(\Gamma_3^{-1}J_b(u)\Big)\widetilde{\Omega} + \frac{1}{2}\widetilde{\Omega}^T\Big(\Gamma_3^{-1}J_b(u)\Big)\dot{\widetilde{\Omega}} \tag{4.45b}$$

$$\dot{V}_{XPD} = -\vec{q}_e^{\,T}\vec{\omega}_b + \frac{1}{2}\dot{\widetilde{\Omega}}^T\Big(\Gamma_3^{-1}J_b(u)\Big)\widetilde{\Omega} + \frac{1}{2}\widetilde{\Omega}^T\Big(\Gamma_3^{-1}J_b(u)\Big)\dot{\widetilde{\Omega}}\Big|_{\vec{\omega}_e=-\vec{\omega}_b} \tag{4.45c}$$

It then follows, substituting $\dot{\vec{\omega}}_b$ from Eq:4.43, the auxiliary plant's derivative $\dot{\widetilde{\Omega}}$ is:

$$\dot{\widetilde{\Omega}} = -\dot{\vec{\omega}}_b + \Gamma_1 \dot{q}_e = -\dot{\vec{\omega}}_b - \dot{\bar{\Omega}} \tag{4.46a}$$

$$\rightarrow \dot{\vec{\omega}}_b = J_b^{-1}(u)\big(-\vec{\omega}_b \times J_b(u)\vec{\omega}_b - \vec{\tau}_b(u) + \vec{\tau}_g + \vec{\tau}_H + \vec{\tau}_{XPD}\big) \tag{4.46b}$$

$$\therefore \dot{\widetilde{\Omega}} = -J_b^{-1}(u)\big(-\vec{\omega}_b \times J_b(u)\vec{\omega}_b - \vec{\tau}_b(u) + \vec{\tau}_g + \vec{\tau}_H + \vec{\tau}_{XPD}\big) - \dot{\bar{\Omega}} \tag{4.46c}$$

Substituting the auxiliary PD control law, $\vec{\tau}_{XPD}$ from Eq:4.41, into the auxiliary derivative $\dot{\widetilde{\Omega}}$:

$$\rightarrow \dot{\widetilde{\Omega}} = -J_b^{-1}(u)\Big(\Gamma_2\widetilde{\Omega} + \Gamma_3\vec{q}_e - J_b(u)\dot{\bar{\Omega}}\Big) - \dot{\bar{\Omega}} \tag{4.46d}$$

$$= J_b^{-1}(u)\Big(-\Gamma_2\widetilde{\Omega} - \Gamma_3\vec{q}_e\Big) \tag{4.46e}$$

From the *approximately* diagonal inertial matrix $J_b(u)$, combined Eq:2.32 and Eq:2.33, and the positive symmetric or *diagonal* properties of the coefficient matrices Γ_1,Γ_2 and Γ_3; the auxiliary plant $\dot{\widetilde{\Omega}}$ has a transpose:

$$\dot{\widetilde{\Omega}}^T = J_b^{-1}\Big(-\Gamma_2\widetilde{\Omega}^T - \Gamma_3\vec{q}_e^{\,T}\Big) \tag{4.47}$$

The PD auxiliary plant component(s) in the LFC derivative $\dot{V}_{XPD}$ in Eq:4.44 simplifies:

$$\frac{1}{2}\dot{\widetilde{\Omega}}^T\Big(\Gamma_3^{-1}J_b(u)\Big)\widetilde{\Omega} = \frac{1}{2}\Big(-\Gamma_2\widetilde{\Omega}^T - \Gamma_3\vec{q}_e^{\,T}\Big)\Gamma_3^{-1}\widetilde{\Omega} \tag{4.48a}$$

$$= \frac{1}{2}\Big(-\widetilde{\Omega}^T\Gamma_2\Gamma_3^{-1}\widetilde{\Omega} - \vec{q}_e^{\,T}\widetilde{\Omega}\Big) \tag{4.48b}$$

Substituting Eq:4.43 for $\vec{q}_e^{\,T}\widetilde{\Omega}$ into Eq:4.48b:

$$\rightarrow \frac{1}{2}\dot{\widetilde{\Omega}}^T\Big(\Gamma_3^{-}J_b(u)\Big)\widetilde{\Omega} = \frac{1}{2}\Big(-\widetilde{\Omega}^T\Gamma_2\Gamma_3^{-1}\widetilde{\Omega} + \vec{q}_e^{\,T}\vec{\omega}_b - \vec{q}_e^{\,T}\Gamma_1\vec{q}_e\Big)\Big|_{\vec{q}_e^{\,T}\widetilde{\Omega}=-\vec{q}_e^{\,T}\vec{\omega}_b+\Gamma_1\vec{q}_e^{\,T}} \tag{4.48c}$$

Similarly, for the transposed counterpart of Eq:4.48c in Eq:4.45c:

$$\frac{1}{2}\widetilde{\Omega}^T\Big(\Gamma_3^{-1}J_b(u)\Big)\dot{\widetilde{\Omega}} = \frac{1}{2}\Big(- \widetilde{\Omega}\Gamma_2\Gamma_3^{-1}\widetilde{\Omega}^T + \vec{q}_e\vec{\omega}_b{}^T - \vec{q}_e\Gamma_1\vec{q}_e{}^T\Big) \tag{4.48d}$$

Which, when substituted back into Eq:4.45c, then simplifies the LFC derivative to negative definite:

$$\Rightarrow \dot{V}_{XPD} = -\vec{q}_e{}^T\Gamma_1\vec{q}_e - \widetilde{\Omega}\Gamma_2\Gamma_3^{-1}\widetilde{\Omega}^T \quad < 0,\ \forall(\vec{q}_e,\widetilde{\Omega}),\ \exists(\Gamma_1,\Gamma_2,\Gamma_3) > 0 \tag{4.49}$$

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

$$\lim_{t\to\infty}\vec{q}_e = 0 \quad\text{and}\quad \lim_{t\to\infty}\widetilde{\Omega} = 0 \tag{4.50a}$$

Then, from the auxiliary plant definition(s) in Eq:4.43, the extended limits present themselves;

$$\lim_{t\to\infty}\vec{\omega}_b = \vec{0}\Big|_{\vec{\omega}_d=\vec{0}} \qquad\text{and}\qquad \lim_{t\to\infty}\bar{\Omega} = \vec{0} \tag{4.50b}$$

Whilst global asymptotic stability is indeed satisfactory, faster exponential stability is obviously more desirable. The stability proof for V_{XPD} can be extended to a stabilizing exponentially bounded trajectory. From a quaternion's inherent definition it follows that $0 \leq |q_0| \leq 1$. It can then be stated that:

$$1 - |q_0| \leq 1 - q_0^2 = \|\vec{q}_e\|^2 \tag{4.51}$$

Exponential stability is a maximum boundedness proof; the relationship Eq:4.51 can then replace the quaternion scalar term $2(1 - q_0)$ in V_{XPD} as an upper bound. The LFC is then rewritten in terms of its component's norm(s) to produce a bounding inequality:

$$V_{XPD} = \vec{q}_e{}^T\vec{q}_e + (q_0 - 1)^2 + \frac{1}{2}\widetilde{\Omega}^T\big(\Gamma_3^{-1}J_b(u)\big)\widetilde{\Omega} \tag{4.52a}$$

$$\to V_{XPD} \leq 2\,\|\vec{q}_e\|^2 + \frac{1}{2}\Gamma_3^{-1}J_b(u)||\widetilde{\Omega}||^2 \tag{4.52b}$$

Similarly the LFC's derivative can be written in terms of its norms as:

$$\dot{V}_{XPD} \leq -\Gamma_2\Gamma_3^{-1}||\widetilde{\Omega}||^2 - \Gamma_1\,\|\vec{q}_e\|^2 \tag{4.52c}$$

The LFC, V_{XPD}, has a maximum such that:

$$V_{XPD} \leq max\bigg\{2, \frac{\lambda_{max}(\Gamma_3^{-1}J_b(u))}{2}\bigg\}\big(\,\|\vec{q}_e\|^2 + ||\widetilde{\Omega}||^2\big) \tag{4.53}$$

Where the function λ_{max} represents the maximum eigenvalue of its argument; in this case $\Gamma_3^{-1}J_b(u)$. Similarly the *negative definite* LCF derivative is bound by the minimum:

$$\dot{V}_{XPD} \leq -min\big\{\lambda_{min}(\Gamma_1), \lambda_{min}(\Gamma_2\Gamma_3{}^{-1})\big\}\big(\,\|\vec{q}_e\|^2 + ||\widetilde{\Omega}||^2\big) \tag{4.54}$$

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 α is defined as the ratio:

$$\alpha = \frac{min\big\{\lambda_{min}(\Gamma_1), \lambda_{min}(\Gamma_2\Gamma_3{}^{-1})\big\}}{max\big\{2, \frac{\lambda_{max}(\Gamma_3{}^{-1}J_b(u))}{2}\big\}} \tag{4.55}$$

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

$$\big(\,\|\vec{q}_e(t)\|\,, ||\widetilde{\Omega}(t)||\big) \leq Me^{-\alpha t/2}\big(\,\|\vec{q}_e(0)\|\,, ||\widetilde{\Omega}(0)||\big) \tag{4.56}$$

The bounding exponential coefficient α can be found using maximum Eigen values for the maximum inertia $J_b(u_\Lambda)$ from Eq:2.32. Using the relationship in Eq:4.56 and testing proposed controller coefficients for $\Gamma_1,\ \Gamma_2,\ \Gamma_3$ the settling rate can be optimized.

The above stability proof for the auxiliary attitude controller was expanded upon and derived from [131], adapted to fit attitude setpoint tracking. Introduction of the quaternion error, which is dependent on the quaternion scalar, dramatically improves controller performance. The exponential stability notably improves settling times and overshoot errors, demonstrated in Sec:6.3.2.

Interestingly, a previous [70] was the precursor for PD based attitude plants with asymptotic exponential stability. That first proposed control law did not make use of any defined *auxiliary plants*, unlike Eq:4.41; however equivalent terms were effectively incorporated. The control law was developed for spacecraft attitude tracking and proposed a very similar exponentially stabilizing control scheme to that of $\vec{\tau}_{XPD}$. That controller, when changed to the notational convention used here, designs body torque as:

$$\vec{\tau}_{XPD}' = -\frac{1}{2}\Big[\big([\vec{q}_e]_\times + q_0\mathbb{I}_{3\times3}\big)\Gamma_1 + \alpha\big(1 - q_0\mathbb{I}_{3\times3}\big)\Big]\vec{q}_e - \Gamma_2\vec{\omega}_b \in \mathcal{F}^b \tag{4.57}$$

Eq:4.57 could easily incorporate plant dependent compensation to accommodate for unwanted nonlinear dynamics. Both exponentially stabilizing PD controllers, from Eq:4.41 and above Eq:4.57, bear a striking similarity to the ideal backstepping controllers derived in the sequel, Eq:4.66.

4.6.3 Nonlinear Controllers

Backstepping controllers([11,73,75],etc...) are a popular choice for nonlinear attitude control plants. The process, through iterative design, enforces Lyapunov stability criteria to ensure asymptotic stability. A report [139] surveys the fundamentals of backstepping procedure. Ideal backstepping control (*IBC*) is a precise control solution which requires exact plant matching, something that is difficult to achieve in practice. Considering that most compensating feedback terms use state estimates $\hat{\mathbf{x}}(t)$ or actuator state estimates $\hat{u}$.

The caveat of IBC control is theoretically poor robust stability performance; being especially susceptible to plant dependent uncertainty. Unmodelled disturbances and uncertainties could easily drive the energy function away from stability conditions. The ideal backstepping algorithm can then be extended to incorporate such uncertainties. Adaptively including disturbance and *estimate* uncertainty into the LFC energy function improves the stability's robustness (Adaptive backstepping control, *ABC*). By Lyapunov's theorem the respective estimation error terms are stabilized.

Ideal Backstepping Controller

Starting with the ideal case for the first proposed backstepping controller, similar to [75]; it is assumed the attitude plant described in Eq:3.129d from the consolidated model in Sec:3.5 exactly matches the dynamics of the physical prototype. The ideal backstepping controller aims to compensate for the plant's dynamic response to trajectory inputs perfectly. Neglecting uncertainties associated with the dynamic model, the aim here is to apply a stabilizing torque design. Recalling the quaternion tracking error from Eq:3.56; $Q_e = Q_b^* \otimes Q_e$, consider the first LFC proposal for a quaternion error Q_e:

$$V_1(Q_e) = \vec{q}_e^{\,T}\vec{q}_e + (1 - q_0)^2 \quad > 0,\ \forall(Q_e) \tag{4.58}$$

After substituting in the quaternion rates but *without* using the quaternion reduction proposed in Eq:4.32, $V_1(Q_e)$ has a Lie derivative:

$$\dot{V}_1 = 2\vec{q}_e^{\,T}\frac{1}{2}\big([\vec{q}_e]_\times + q_0\mathbb{I}_{3X3}\big)\vec{\omega}_e - 2\big(1 - q_0\big)\dot{q}_0 \tag{4.59a}$$

$$= \vec{q}_e^{\,T}\Big([\vec{q}_e]_\times + q_0\mathbb{I}_{3\times3}\Big)\vec{\omega}_e + \big(1 - q_0\big)\vec{q}_e^{\,T}\vec{\omega}_e \tag{4.59b}$$

Simplifying further and then substituting the angular velocity set point; $\vec{\omega}_e = \vec{\omega}_d - \vec{\omega}_b = -\vec{\omega}_b$:

$$= \vec{q}_e^{\,T}[\vec{q}_e]_\times\vec{\omega}_e + \vec{q}_e^{\,T}\vec{\omega}_e \tag{4.59c}$$

$$= -\vec{q}_e^{\,T}[\vec{q}_e]_\times\vec{\omega}_b - \vec{q}_e^{\,T}\vec{\omega}_b\Big|_{\vec{\omega}_e=-\vec{\omega}_b} \tag{4.59d}$$

Then choosing the first virtual backstepping control input γ_d. Note that γ_d is used here to *differentiate the backstepping design* variable from the trajectory commanded $\vec{\omega}_d$, Eq:4.27a. Choosing γ_d such that the first LFC Eq:4.58 is negative definite, $\dot{V}_1 < 0$:

$$\vec{\omega}_b \Rightarrow \gamma_d = \Gamma_1 \vec{q}_e \tag{4.60}$$

Where Γ_1 is the first symmetric positive definite gain matrix, a fact that is important to stress due to positive definite matrix's invertability. That backstepping input simplifies the LFC derivative $\dot{V}_1$ to the negative definite term:

$$\dot{V}_1 = -\vec{q}_e^{\,T}[\vec{q}_e]_\times \gamma_d - \vec{q}_e^{\,T}\gamma_d \tag{4.61a}$$

$$= -\vec{q}_e^{\,T}[\vec{q}_e]_\times \Gamma_1 \vec{q}_e - \vec{q}_e^{\,T}\Gamma_1 \vec{q}_e \tag{4.61b}$$

And considering a vector cross product with itself has a zero resultant, $\vec{q}_e^{\,T}[\vec{q}_e]_\times = \vec{0}$, $\dot{V}_1$ then reduces:

$$= -\vec{q}_e^{\,T}\Gamma_1 \vec{q}_e < 0 \tag{4.61c}$$

However, that backstepping input γ_d has its own associated error. A stabilizing law z_1 needs to control that error:

$$z_1 \triangleq \gamma_d - \vec{\omega}_b = \Gamma_1 \vec{q}_e - \vec{\omega}_b \Big|_{\gamma_d = \Gamma_1 \vec{q}_e} \tag{4.62a}$$

$$\rightarrow \vec{\omega}_b = \Gamma_1 \vec{q}_e - z_1 \tag{4.62b}$$

$$\therefore \dot{V}_1 = -\vec{q}_e^{\,T}\vec{\omega}_b = -\vec{q}_e^{\,T}\big(\Gamma_1 \vec{q}_e - z_1\big)\Big|_{\vec{\omega}_b \Rightarrow \gamma_d} \tag{4.62c}$$

$$= -\vec{q}_e^{\,T}\Gamma_1 \vec{q}_e + \vec{q}_e^{\,T} z_1 \tag{4.62d}$$

Introducing that error z_1 into a second LCF, which expands the first proposed V_1. And exploiting the fact that Γ_1 is symmetrical:

$$V_2(Q_e, z_1) = V_1(\vec{q}_e) + \frac{1}{2} z_1^{\,T} z_1 \tag{4.63a}$$

$$= \vec{q}_e^{\,T}\vec{q}_e + (1 - q_0)^2 + \frac{1}{2} z_1^{\,T} z_1 \quad > 0, \ \forall (Q_e, z_1) \tag{4.63b}$$

That first error z_1 has its own time derivative, and recalling the body's angular acceleration $\dot{\vec{\omega}}_b$ from earlier with an undefined input $\vec{\tau}_{IBC}$, which still has plant dependency compensation.

$$\dot{z}_1 = \Gamma_1 \dot{\vec{q}}_e - \dot{\vec{\omega}}_b \tag{4.64a}$$

$$= \frac{\Gamma_1}{2}\big([\vec{q}_e]_\times + q_0 \mathbb{I}_{3X3}\big)\vec{\omega}_e - \dot{\vec{\omega}}_b \tag{4.64b}$$

$$= -\frac{\Gamma_1}{2}\big([\vec{q}_e]_\times + q_0 \mathbb{I}_{3\times 3}\big)\vec{\omega}_b - \dot{\vec{\omega}}_b \Big|_{\vec{\omega}_e = -\vec{\omega}_b} \tag{4.64c}$$

$$= -\frac{\Gamma_1}{2}\big([\vec{q}_e]_\times + q_0 \mathbb{I}_{3\times 3}\big)\vec{\omega}_b - J_b(u)^{-1}\big(-\vec{\omega}_b \times J_b(u)\vec{\omega}_b - \vec{\tau}_b(u) + \vec{\tau}_g + \vec{\tau}_H + \vec{\tau}_{IBC}\big) \tag{4.64d}$$

So then, following from Eq:4.64d, finding the derivative of $\dot{V}_2$, with $\dot{V}_1 = -\vec{q}_e^{\,T}(\Gamma_1 \vec{q}_e - z_1)$:

$$\begin{aligned}\dot{V}_2 = -\vec{q}_e^{\,T}\big(\Gamma_1 \vec{q}_e - z_1\big) + z_1^{\,T}\Bigg(&-\frac{\Gamma_1}{2}\big([\vec{q}_e]_\times + q_0 \mathbb{I}_{3X3}\big)\vec{\omega}_b \\ &- J_b^{-1}(u)\big(-\vec{\omega}_b \times J_b(u)\vec{\omega}_b - \vec{\tau}_b(u) + \vec{\tau}_g + \vec{\tau}_H + \vec{\tau}_{IBC}\big)\Bigg)\end{aligned} \tag{4.65a}$$

$$\begin{aligned}= -\vec{q}_e^{\,T}\Gamma_1 \vec{q}_e + z_1^{\,T}\Bigg(&\vec{q}_e - \frac{\Gamma_1}{2}\big([\vec{q}_e]_\times + q_0 \mathbb{I}_{3X3}\big)\vec{\omega}_b \\ &- J_b^{-1}(u)\big(-\vec{\omega}_b \times J_b(u)\vec{\omega}_b - \vec{\tau}_b(u) + \vec{\tau}_g + \vec{\tau}_H + \vec{\tau}_{IBC}\big)\Bigg)\end{aligned} \tag{4.65b}$$

So then proposing the exactly matched stabilizing backstepping control law using state estimes:

$$\vec{\tau}_{IBC} = J_b(\hat{u})\vec{q}_e - \frac{J_b(\hat{u})\Gamma_1}{2}\big([\vec{q}_e]_\times + q_0\mathbb{I}_{3X3}\big)\vec{\omega}_b + J_b(\hat{u})\Gamma_2 z_1 + \hat{\omega}_b \times J_b(\hat{u})\vec{\omega}_b + \vec{\tau}_b(\hat{u}) - \vec{\tau}_g - \vec{\tau}_H \tag{4.66a}$$

Noting that $z_1 = \Gamma_1\vec{q}_e - \vec{\omega}_b$ and using the quaternion rate's vector definition, Eq:4.35, the IBC torque law reduces:

$$= \underbrace{J_b(\hat{u})\Big((\Gamma_1\Gamma_2 + 1)\vec{q}_e - \Gamma_2\hat{\omega}_b + \Gamma_1\dot{\vec{q}_e}\Big)}_{\text{Ideal backstepping}} + \underbrace{\hat{\omega}_b \times J_b(\hat{u})\hat{\omega}_b + \vec{\tau}_b(\hat{u}) - \vec{\tau}_g - \vec{\tau}_H}_{\text{Compenstation}} \quad \in \mathcal{F}^b \tag{4.66b}$$

With Γ_2 being another positive-definite symmetric coefficient matrix. Then with the control law $\vec{\tau}_{IBC}$ introduced into the LCF derivative and assuming state estimates have negligible errors, $\dot{V}_2$ simplifies to negative definite:

$$\dot{V}_2 = -\vec{q}_e{}^T\Gamma_1\vec{q}_e + z_1{}^T\Bigg(\vec{q}_e - \frac{\Gamma_1}{2}\big([\vec{q}_e]_\times + q_0\mathbb{I}_{3\times3}\big)\vec{\omega}_b - J_b^{-1}(u)\big(J_b(\hat{u})(\Gamma_1\Gamma_2 + 1)\vec{q}_e - J_b(\hat{u})\Gamma_2\hat{\omega}_b + J_b(\hat{u})\Gamma_1\dot{\vec{q}_e}\big)\Bigg) \tag{4.67a}$$

$$\therefore \dot{V}_2 = -\vec{q}_e{}^T\Gamma_1\vec{q}_e + z_1{}^T\big(\Gamma_1\Gamma_2\vec{q}_e - \Gamma_2\hat{\omega}_b\big) \tag{4.67b}$$

$$= -\vec{q}_e{}^T\Gamma_1\vec{q}_e - z_1{}^T\Gamma_2 z_1 \quad < 0,\ \forall\ (Q_e, z_1),\ \exists(\Gamma_1, \Gamma_2) > 0 \tag{4.67c}$$

As such $\vec{q}_e \to 0$ and $q_0 \to 1$ as $t \to \infty$. Similarly $z_1 \to 0$, which leads to the limit:

$$\lim_{t\to\infty}(\Gamma_1\vec{q}_e - \vec{\omega}_b) = \vec{0} \tag{4.68}$$

Because the quaternion error vector already tends to 0; $\vec{q}_e \to \vec{0}$, it follows that $\vec{\omega}_b \to \vec{0}$ as well. 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 $\vec{\tau}_{IBC}$ from Eq:4.66 and that of the auxiliary PD controller presented in Eq:4.41. Expanding $\vec{\tau}_{XPD}$ into state terms using the definitions of each auxiliary plant, $\widetilde{\Omega}$ and $\dot{\tilde{\Omega}}$:

$$\vec{\tau}_{XPD} = \big(\Gamma_1\Gamma_2 + \Gamma_3\big)\vec{q}_e - \Gamma_2\hat{\omega}_b - \frac{\Gamma_1 J_b(\hat{u})}{2}([\vec{q}_e]_\times + q_0\mathbb{I}_{3\times3})\hat{\omega}_b \tag{4.69}$$

Furthermore, using the same reasoning from Eq:4.52, the exponential stability proof is proposed in the sequel. Recalling Eq:4.51:

$$q_0 - 1 \le 1 - q_0^2 = \|\vec{q}_e\|^2 \tag{4.70a}$$

$$V_{IBC} \le V_{IBC}{}' = 2\,\|\vec{q}_e\|^2 + \frac{1}{2}\,\|z_1\|^2 \tag{4.70b}$$

$$\dot{V}_{IBC} \le \dot{V}_{IBC}{}' = -\Gamma_1\,\|\vec{q}_e\|^2 - \Gamma_2\,\|z_1\|^2 \tag{4.70c}$$

Then both the energy function and its derivative are bounded respectively by the following Eigen value limits:

$$V_{IBC} \le \left\{2,\ \frac{1}{2}\right\}(\|\vec{q}_e\|^2 + \|z_1\|^2) \tag{4.71a}$$

$$\dot{V}_{IBC} \le -min\big\{\lambda_{min}(\Gamma_1),\ \lambda_{min}(\Gamma_2)\big\}(\|\vec{q}_e\|^2 + \|z_1\|^2) \tag{4.71b}$$

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

$$\dot{V}_{IBC} \le -\alpha V_{IBC} \tag{4.72a}$$

$$\therefore V\big(\,\|\vec{q}_e(t)\|\,,\|z_1(t)\|\,\big) \le Me^{-\alpha t/2}V\big(\,\|\vec{q}_e(0)\|\,,\|z_1(0)\|\,\big) \tag{4.72b}$$

Whilst stabilizing, the IBC controller is not globaly stable. The plant conditions require exact plant matching and account for no unmodelled disturbances or measurement uncertainty. In practice the introduction of some disturbance torque $\vec{L}$ could potentially drive Eq:4.65 away from negative-definite such that stability is lost.

Adaptive Backstepping Controller

A lot of work has been done on the statistical nature of disturbance approximation and how best to adapt a nonlinear control system to the influence of unwanted disturbances; [12, 38, 52]. Considering only a lumped uncertainty/disturbance term for the adaptive case and assuming plant dependent uncertainties and estimate errors can all be included in such a term. A lumped $\vec{L}_b$ in the body frame is then added into the angular acceleration dynamics:

$$\dot{\vec{\omega}}_b = {J_b}^{-1}(u)\big(- \vec{\omega}_b \times J_b(u)\vec{\omega}_b - \vec{\tau}_b(u) + \vec{\tau}_g + \vec{\tau}_H + \vec{L}_b + \vec{\tau}_{ABC}\big) \qquad \in \mathcal{F}^b \tag{4.73}$$

Unmodelled disturbances act as external torques on the Lagrangian in Eq:3.9c. Plant modelling errors and disturbances could simply be compensated for in the control law; $-\vec{L}_b$. It is, however, practically difficult to estimate disturbances without any *apriori* knowledge about of its properties. Noise compensation in sensors can be done easily due to the known frequency bandwidth within which that noise occurs; the same cannot be said for wind disturbances or payload variations...

An approximate disturbance observer, $\hat{L}$, is used for that compensation in the designed control torque $\vec{\tau}_{ABC}$. Each estimate will have its own error deviating from the physical $\vec{L}_b$ acting on the vehicle:

$$\vec{L}_\Delta = \vec{L}_b - \hat{L} \tag{4.74}$$

Adaptive backstepping control introduces that observer's estimate error into an LFC to develop a derivative term for $\dot{\hat{L}}$, or a *disturbance update law*, to asymptotically stabilize the estimate error. Typically, disturbance update rules are the primary contribution for satellite and generalized attitude control research papers, the statistical nature of disturbance approximation is a subject for another project. That estimate error, $\vec{L}_\Delta$ is then introduced to an LFC derived from the IBC case (previously in Eq:4.63a):

$$V_{ABC}(Q_e, z_1, \vec{L}_\Delta) = V_{IBC}(Q_e, z_1) + \frac{1}{2}\vec{L}_\Delta^T\Gamma_L^{-1}\vec{L}_\Delta \tag{4.75a}$$

$$= {\vec{q}_e}^T\vec{q}_e + (1-q_0)^2 + \frac{1}{2}{z_1}^T z_1 + \frac{1}{2}\vec{L}_\Delta^T\Gamma_L^{-1}\vec{L}_\Delta \quad > 0,\ \forall(Q_e, z_1, \vec{L}_\Delta) \tag{4.75b}$$

Where Γ_L is the positive 3×3 positive adaptation gain matrix. That gain determines the rate at which the system *adapts* to disturbances. The stability proof starts with the LFC rate $\dot{V}_{ABC}$:

$$\dot{V}_{ABC}(Q_e, z_1, \vec{L}_\Delta) = \dot{V}_{IBC}(Q_e,\ z_1) + \frac{1}{2}\dot{\vec{L}}_\Delta^T\Gamma_L^{-1}\vec{L}_\Delta + \frac{1}{2}\vec{L}_\Delta^T\Gamma_L^{-1}\dot{\vec{L}}_\Delta \tag{4.76}$$

Recalling the definition of $\vec{L}_\Delta$ from Eq:4.74; for its derivative $\dot{\vec{L}}_\Delta$ it is reasonable to assume the rate at which the physical disturbance $\vec{L}$ changes is significantly slower than that of the control system, or that $\dot{\vec{L}}_b << \dot{\hat{L}}$. It then follows that:

$$\therefore \dot{\vec{L}}_\Delta = \dot{\vec{L}}_b - \dot{\hat{L}} \approx \vec{0} - \dot{\hat{L}} = -\dot{\hat{L}}\Big|_{\dot{\vec{L}}\approx\vec{0}} \tag{4.77}$$

Substituting that estimation error rate back into the LFC derivative $\dot{V}_{ABC}$ yields:

$$\begin{aligned}\dot{V}_{ABC} = -{\vec{q}_e}^T(\Gamma_1\vec{q}_e - z_1) + {z_1}^T\Bigg(&- \frac{\Gamma_1}{2}\big([\vec{q}_e]_\times + q_0\mathbb{I}_{3\times3}\big)\vec{\omega}_b \\ &- J_b^{-1}(u)\big(- \vec{\omega}_b \times J_b(u)\vec{\omega}_b - \vec{\tau}_b(u) + \vec{\tau}_g + \vec{\tau}_H + \vec{L}_b + \vec{\tau}_{ABC}\big)\Bigg) - \vec{L}_\Delta^T\Gamma_L^{-1}\dot{\hat{L}}\end{aligned} \tag{4.78a}$$

Note that the physical disturbance term $\vec{L}_b$ is included in Eq:4.78a. Extending the ideal back stepping control law, $\vec{\tau}_{IBC}$ from Eq:4.66, to include a disturbance *estimate* $\hat{L}$ compensation term:

$$\vec{\tau}_{ABC} = J_b(\hat{u})\Big(\big(\Gamma_1\Gamma_2 + 1\big)\vec{q}_e - \Gamma_2\hat{\omega}_b + \Gamma_1\dot{\vec{q}_e}\Big) + \hat{\omega}_b \times J_b(\hat{u})\hat{\omega}_b + \vec{\tau}_b(\hat{u}) - \vec{\tau}_g - \vec{\tau}_H - \hat{L} \qquad \in \mathcal{F}^b \tag{4.78b}$$

Turbulence torques $\vec{L}_b$ act in the body frame and so require no references frame transformation. Reiterating that the control law still compensates for disturbances and dynamics with plant estimates whose error is included in $\vec{L}_b$. The energy function's derivative $\dot{V}_{ABC}$ then reduces to:

$$\dot{V}_{ABC} = \dot{V}_{IBC} - {z_1}^T J_b^{-1}(u)\left(\vec{L}_b - \hat{L}\right) - {\vec{L}_\Delta}^T \Gamma_L^{-1} \dot{\hat{L}} \tag{4.78c}$$

$$= -{\vec{q}_e}^T \Gamma_1 \vec{q}_e - {z_1}^T \Gamma_2 z_1 - {z_1}^T J_b^{-1}(u) \vec{L}_\Delta - {\vec{L}_\Delta}^T \Gamma_L^{-1} \dot{\hat{L}} \tag{4.78d}$$

$$= -{\vec{q}_e}^T \Gamma_1 \vec{q}_e - {z_1}^T \Gamma_2 z_1 - {\vec{L}_\Delta}^T \Gamma_L^{-1}\left(\dot{\hat{L}} + \Gamma_L J_b^{-1}(u) z_1\right) \tag{4.78e}$$

The decision then needs to be made as to how the disturbance estimate is updated such that its error $\vec{L}_\Delta$ asymptotically stabilizes, or that $\dot{V}_{ABC} < 0$. The obvious choice for $\dot{\hat{L}}$ would be to exactly compensate for $\Gamma_L J_b^{-1}(u) z_1$ in the LFC:

$$\dot{\hat{L}} \triangleq \Gamma_L J_b^{-1}(\hat{u}) z_1 = -\Gamma_L J_b^{-1}(\hat{u})\left(\Gamma_1 \vec{q}_e - \hat{\omega}_b\right)\Big|_{z_1 = \Gamma_1 \vec{q}_e - \hat{\omega}_b} \tag{4.79}$$

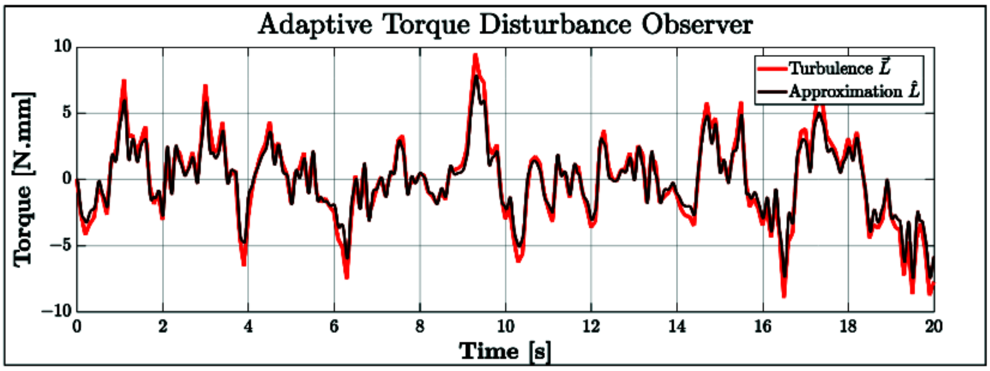
The disturbance is therefore compensated for and the estimate error is ensured to have asymptotic stability seeing that V_{ABC} is positive definite and $V_{ABC}(\vec{0}) = 0$.

$$\dot{V}_{ABC} = -{\vec{q}_e}^T \Gamma_1 \vec{q}_e - {z_1}^T \Gamma_2 z_1 \quad < 0,\ \forall\ (Q_e, z_1, \vec{L}_\Delta),\ \exists(\Gamma_1, \Gamma_2, \Gamma_L) > 0 \tag{4.80}$$

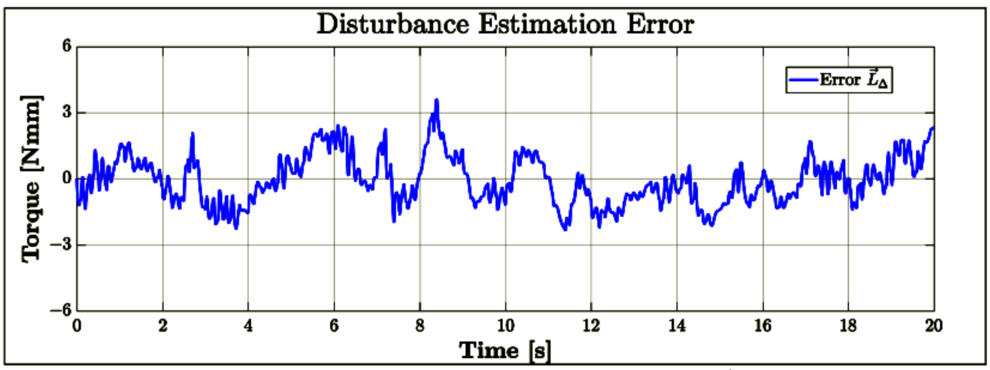
The same attitude stabilizing limits exists from Eq:4.80 can be drawn but, most importantly, the disturbance observer estimation error is stabilized:

$$\lim_{t\to\infty} \vec{L}_\Delta \to \vec{0} \quad \text{and} \quad \therefore \lim_{t\to\infty} \hat{L} \to \vec{L}_b \tag{4.81a}$$

Fig:4.6a shows how the disturbance observer $\hat{L}$ approximates a (single axis) torque turbulence acting the vehicle in *steady state* hovering. A moderate damping manifests on the estimate in relationto the physical disturbance; resulting in an error shown Fig:4.6b. The example shown in 4.6 contains no attitude steps or trajectory changes. The torque turbulence, the observer and the adaptive controller's performance is detailed later in Sec:6.6.



(a) Torque disturbance observer



(b) Torque disturbance error deviation $\vec{L}_\Delta$

Figure 4.6: Adaptive disturbance observer example

4.7 Position Control

Only two plant dependent position control laws are derived here, attitude control is the primary focus. The attitude control loop is stabilized independently from the position loop (Eq:3.129d and Eq:3.129b) but the Coriolis cross-coupling, from Eq:3.10b, means the position loop first needs a stable attitude before being stabilized itself. A simple Proportional-Derivative structure is presented first as the reference case. Thereafter an ideal backstepping control which is extended to an adaptive control law is derived. Recalling the dynamics for translational acceleration from Eq:3.129b:

$$\dot{\vec{v}}_b = m_b^{-1}\big(- \vec{\omega}_b \times m_b\vec{v}_b + m_b\vec{G}_b + \vec{F}_\mu(\hat{u})\big) \qquad \in \mathcal{F}^b \tag{4.82}$$

Reiterating that the Coriolis acceleration term $-\vec{\omega}_b \times m_b\vec{v}_b$ is what couples the position loop to the attitude plant. Noting that $\vec{G}_b$ is the gravitational acceleration transformed to the body frame. Furthermore most texts assume that under standard operating conditions (App:A.1) angular velocity is small if not negligible; $\vec{\omega}_b \approx \vec{0}$. Such an approximation makes the coupled Coriolis term assumed to be insignificant; $\vec{\omega}_b \times m\vec{v}_b \approx \vec{0}$. If the plant's state is known, or atleast estimated with a relative degree of certainty, it is easy to compensate for those dynamics rather than making assumptions about their influence on the system. Such an introduced plant dependency can be compensated for in the designed control force $\vec{F}_\mu(\hat{u})$. The translational velocity, $\vec{v}_b$, defined in the body frame is related to the inertial position rates through a quaternion transformation:

$$\dot{\vec{\mathcal{E}}}_b = Q_b \otimes \vec{v}_b \otimes Q_b^* \qquad \in \mathcal{F}^I \tag{4.83}$$

The difference in reference frames is an important distinction between the position and attitude state equations. Position error is calculated purely as a subtractive term from a particular setpoint $\vec{\mathcal{E}}_d$:

$$\vec{\mathcal{E}}_e = \vec{\mathcal{E}}_d - \vec{\mathcal{E}}_b \qquad \in \mathcal{F}^I \tag{4.84}$$

The translational position rate error $\dot{\vec{\mathcal{E}}}_b(t)$, *not velocity error* $\vec{v}_e$, can be similarly calculated but, in the same way. In this case both position rate and velocity setpoints are zero, $\dot{\vec{\mathcal{E}}}_d = \vec{v}_d = \vec{0}$.

$$\dot{\vec{\mathcal{E}}}_e = \dot{\vec{\mathcal{E}}}_d - \dot{\vec{\mathcal{E}}}_b = -\dot{\vec{\mathcal{E}}}_b\Big|_{\dot{\vec{\mathcal{E}}}_d=\vec{0}} \qquad \in \mathcal{F}^I \tag{4.85a}$$

$$\therefore \vec{v}_e = Q_b^* \otimes \big(\dot{\vec{\mathcal{E}}}_d - \dot{\vec{\mathcal{E}}}_b\big) \otimes Q_b = -\vec{v}_b \qquad \in \mathcal{F}^b \tag{4.85b}$$

Position setpoint tracking aims is to produce a stabilizing control law $g(\vec{\mathbf{x}}_e, \dot{\vec{\mathbf{x}}}_e, t)$ that ensures the position tracking error asymptotically tends to $\vec{0}$. Or more formally that:

$$\vec{F}_\mu(\hat{u}) = g\Big(\vec{\mathcal{E}}_d, \dot{\vec{\mathcal{E}}}_d, \vec{\mathcal{E}}_b, \dot{\vec{\mathcal{E}}}_b, t\Big) \equiv g(\vec{\mathcal{E}}_e, \dot{\vec{\mathcal{E}}}_e, t) \qquad \in \mathcal{F}^b \tag{4.86a}$$

$$\text{Such that: } \lim_{t\to\infty} \vec{\mathcal{E}}_e \to \vec{0} \tag{4.86b}$$

4.7.1 PD Controller

Starting with a simple PD controller to be used for the reference case. Plant dependent control designs the net force proportional to both the position error and the first derivative velocity error:

$$\vec{F}_{PD} = K_p\vec{\mathcal{E}}_e + K_d\dot{\vec{\mathcal{E}}}_e + \hat{\omega}_b \times m_b\hat{v}_b - m_b\vec{G}_b \qquad \in \mathcal{F}^b \tag{4.87a}$$

$$= K_p\big(\vec{\mathcal{E}}_d - \hat{\mathcal{E}}_b\big) - K_d\big(\dot{\hat{\mathcal{E}}}_b\big) + \hat{\omega}_b \times m_b\hat{v}_b - m_b\vec{G}_b\Big|_{\dot{\vec{\mathcal{E}}}_e=-\dot{\hat{\mathcal{E}}}_b} \tag{4.87b}$$

Note that position and attitude state estimates are used for the controller in Eq:4.87. As with attitude state estimates, it is assumed those estimates are error free and any plant errors are incorporated into the subsequent adaptive control law presented next in Sec:4.7.2.

The stability proof requires that error states are transformed to the body frame $\mathcal{F}^b$, such that the control input and error states all act in a common frame. Defining a position error state $\vec{X}_e$ transformed to the body frame:

$$\vec{X}_e \triangleq Q_b \otimes (\vec{\mathcal{E}}_d - \vec{\mathcal{E}}_b) \otimes Q_b^* = \vec{X}_d - \vec{X}_b \qquad \in \mathcal{F}^b \tag{4.88a}$$

Recalling the difference between position rates and translational velocity in Eq:3.129a, position rates are then:

$$\dot{\vec{X}}_e \triangleq Q_b \otimes (\dot{\vec{\mathcal{E}}}_d - \dot{\vec{\mathcal{E}}}_b) \otimes Q_b^* = -Q_b \otimes \dot{\vec{\mathcal{E}}}_b \otimes Q_b^* = -\vec{v}_b\Big|_{\dot{\vec{\mathcal{E}}}_d=\vec{0}} \tag{4.88b}$$

The control law from Eq:4.87, despite being $\in \mathcal{F}^b$ has arguments $\vec{\mathcal{E}}_e, \dot{\vec{\mathcal{E}}}_e \in \mathcal{F}^I$, which are substituted with the transformed position error $\vec{X}_e$ and its rate $\dot{\vec{X}}_e$:

$$\vec{F}_{PD} = K_p\vec{X}_e + K_d\dot{\vec{X}}_e + \hat{\omega}_b \times m_b\hat{v}_b - m_b\vec{G}_b \qquad \in \mathcal{F}^b \tag{4.89a}$$

$$= K_p\vec{X}_e - K_d\hat{v}_b + \hat{\omega}_b \times m_b\hat{v}_b - m_b\vec{G}_b \tag{4.89b}$$

Then proposing a positive definite Lyapunov function candidate:

$$V_{PD}(\vec{X}_e, \dot{\vec{X}}_e) = \frac{1}{2}\vec{X}_e{}^T K_p\vec{X}_e + \frac{1}{2}\dot{\vec{X}}_e{}^T m_b\dot{\vec{X}}_e > 0 \quad \forall(\vec{X}_e, \dot{\vec{X}}_e) \tag{4.90a}$$

$$= \frac{1}{2}\vec{X}_e{}^T K_p\vec{X}_e + \frac{1}{2}\vec{v}_b{}^T m_b\vec{v}_b\Big|_{\dot{\vec{X}}_e=-\vec{v}_b} \tag{4.90b}$$

Calculating that LFC's derivative $\dot{V}_{PD}$ with the PD control law substituted:

$$\dot{V}_{PD}(\vec{X}_e, \dot{\vec{X}}_e) = \vec{X}_e{}^T K_p\dot{\vec{X}}_e + \vec{v}_b{}^T m_b\dot{\vec{v}}_b \tag{4.91a}$$

$$= -\vec{X}_e{}^T K_p\vec{v}_b + \vec{v}_b{}^T m_b\dot{\vec{v}}_b \tag{4.91b}$$

$$= -\vec{X}_e{}^T K_p\vec{v}_b + \vec{v}_b{}^T\big(- \vec{\omega}_b \times m_b\vec{v}_b + m_b\vec{G}_b + \vec{F}_{PD}\big) \tag{4.91c}$$

$$= -\vec{X}_e{}^T K_p\vec{v}_b + \vec{v}_b{}^T\big(K_p\vec{X}_e - K_d\hat{v}_b\big) \tag{4.91d}$$

$$\therefore \dot{V}_{PD} = -\vec{v}_b{}^T K_d\vec{v}_b \quad < 0,\ \forall(\vec{X}_e, \dot{\vec{X}}_e),\ \exists(K_d, K_p) > 0 \tag{4.91e}$$

The global stability asserted in Eq:4.91e holds for $\forall(\vec{\mathcal{E}}_e, \dot{\vec{\mathcal{E}}}_e)$, irrespective of the transformation applied in Eq:4.88a and Eq:4.88b. Global asymptotically stabilizing limits then follow:

$$\lim_{t\to\infty} \vec{X}_e = Q_b \otimes (\vec{\mathcal{E}}_d - \vec{\mathcal{E}}_b) \otimes Q_b^* \to \vec{0} \tag{4.92a}$$

$$\therefore \lim_{t\to\infty} \vec{\mathcal{E}}_b \to \vec{\mathcal{E}}_d \tag{4.92b}$$

$$\lim_{t\to\infty} \dot{X}_e = Q_b^* \otimes (\dot{\vec{\mathcal{E}}}_d - \dot{\vec{\mathcal{E}}}_b) \otimes Q_b = -\vec{v}_b \to \vec{0}\Big|_{\dot{\vec{\mathcal{E}}}_e=0} \tag{4.92c}$$

4.7.2 Adaptive Backstepping Controller

An adaptive backstepping algorithm, analogue to the adaptive controller previously in Sec:4.6.3, is now applied to position control. The disturbance term, $\vec{D}_b \in \mathcal{F}^b$, introduced to the position state differential Eq:4.82, represents estimate errors together with any unmodelled lumped drag *and* wind forces encountered by the vehicle in flight. Backstepping iterations for the position control loop first need to stabilize the position error and only thereafter compensate those disturbances (solving for *IBC* then adding adaptivity).

$$\dot{\vec{v}}_b = m_b^{-1}\big(- \vec{\omega}_b \times m_b\vec{v}_b + m_b\vec{G}_b + \vec{D}_b + \vec{F}_{ABC}\big) \qquad \in \mathcal{F}^b \tag{4.93}$$

The compensation for $\vec{D}_b$ is obviously an approximation for that physical disturbance term; $\hat{D}_b$. Beginning the backstepping process for position with a position state tracking error:

$$z_1 \triangleq \vec{\mathcal{E}}_d - \vec{\mathcal{E}}_b = \vec{\mathcal{E}}_e \qquad \in \mathcal{F}^I \tag{4.94}$$

Which then has its own derivative:

$$\dot{z}_1 = \dot{\vec{\mathcal{E}}}_e = \dot{\vec{\mathcal{E}}}_d - \dot{\vec{\mathcal{E}}}_b \tag{4.95a}$$

$$= Q_b^* \otimes \big(\vec{v}_d - \vec{v}_b\big) \otimes Q_b = -Q_b^* \otimes \vec{v}_b \otimes Q_b\Big|_{\vec{v}_d=\vec{0}} \tag{4.95b}$$

Transforming that error z_1 to the body frame $\mathcal{F}^b$ in a similar fashion to Eq:4.88a makes the stability proof more concise. The reference frame transformation does not affect the Lie derivative as the energy function's gradient depends on its partial derivative w.r.t its positional trajectory only; namely $\mathcal{E}_e(t)$.

$$\hat{z}_1 \triangleq Q_b \otimes z_1 \otimes Q_b^* = Q_b \otimes \big(\vec{\mathcal{E}}_d - \vec{\mathcal{E}}_b\big) \otimes Q_b^* = \vec{X}_e \qquad \in \mathcal{F}^b \tag{4.96a}$$

$$\therefore \dot{\hat{z}}_1 = Q_b \otimes \dot{z}_1 \otimes Q_b^* = Q_b \otimes \big(\dot{\vec{\mathcal{E}}}_d - \dot{\vec{\mathcal{E}}}_b\big) \otimes Q_b^* = -\vec{v}_b \tag{4.96b}$$

Proposing the first LFC, $V_1(\hat{z}_1)$, in terms of that tracking error with a derivative $\dot{V}_1$:

$$V_1(\hat{z}_1) = \frac{1}{2}\hat{z}_1{}^T\hat{z}_1 \quad > 0,\ \forall(\hat{z}_1) \tag{4.97a}$$

$$\Rightarrow \dot{V}_1(\hat{z}_1) = \hat{z}_1{}^T\dot{\hat{z}}_1 = -\hat{z}_1{}^T\vec{v}_b \tag{4.97b}$$

The first stabilizing velocity function, γ_d, and its associated error, $\hat{z}_2$, are defined as:

$$\vec{v}_b \Rightarrow \gamma_d = \Gamma_1\hat{z}_1 \tag{4.98a}$$

$$\hat{z}_2 \triangleq \gamma_d - \vec{v}_b = \Gamma_1\hat{z}_1 - \vec{v}_b \tag{4.98b}$$

$$\therefore \vec{v}_b = \Gamma_1\hat{z}_1 - \hat{z}_2 \tag{4.98c}$$

Changing that first LFC with variable substitution such that:

$$V_1 = -\hat{z}_1^T\vec{v}_b = -\hat{z}_1^T\Gamma_1\hat{z}_1 + \hat{z}_1^T\hat{z}_2 \tag{4.99}$$

So that second error state $\hat{z}_2$ has a derivative:

$$\dot{\hat{z}}_2 = \dot{\gamma}_d - \dot{\vec{v}}_b = \Gamma_1\dot{\hat{z}}_1 - m_b^{-1}\big(-\vec{\omega}_b \times m_b\vec{v}_b + m_b\vec{G}_b + \vec{D}_b + \vec{F}_{ABC}\big) \tag{4.100a}$$

$$= -\Gamma_1\vec{v}_b - m_b^{-1}\big(-\vec{\omega}_b \times m_b\vec{v}_b + m_b\vec{G}_b + \vec{D}_b + \vec{F}_{ABC}\big) \tag{4.100b}$$

Introducing that second error $\hat{z}_2$ into a new LFC V_2:

$$V_2(\hat{z}_1, \hat{z}_2) = V_1(\hat{z}_1) + \frac{1}{2}\hat{z}_2{}^T\hat{z}_2 \tag{4.101a}$$

$$= \frac{1}{2}\hat{z}_1{}^T\hat{z}_1 + \frac{1}{2}\hat{z}_2{}^T\hat{z}_2 \quad > 0,\ \forall(\hat{z}_1, \hat{z}_2) \tag{4.101b}$$

Which has a derivative, with $\dot{\hat{z}}_2$ substituted from Eq:4.100b:

$$\dot{V}_2(\hat{z}_1, \hat{z}_2) = \dot{V}_1(\hat{z}_1) + \hat{z}_2{}^T\dot{\hat{z}}_2 = \hat{z}_1{}^T\dot{\hat{z}}_1 + \hat{z}_2{}^T\dot{\hat{z}}_2 \tag{4.102a}$$

$$= -\hat{z}_1^T\Gamma_1\hat{z}_1 + \hat{z}_1{}^T\hat{z}_2 + \hat{z}_2{}^T\dot{\hat{z}}_2 \tag{4.102b}$$

$$= -\hat{z}_1{}^T\Gamma_1\hat{z}_1 + \hat{z}_2{}^T\Bigg(\hat{z}_1 - \Gamma_1\vec{v}_b - m_b^{-1}\big(-\vec{\omega}_b \times m_b\vec{v}_b + m\vec{G}_b + \vec{D}_b + \vec{F}_{ABC}\big)\Bigg) \tag{4.102c}$$

An ideal backstepping control law, with the assumption that $\vec{D}_b$ is precisely known, is then:

$$\vec{F}_{IBC} = m_b\big(\hat{z}_1 - \Gamma_1\hat{v}_b + \Gamma_2\hat{z}_2\big) + \hat{\omega}_b \times m_b\hat{v}_b - m_b\vec{G}_b - \vec{D}_b \qquad \in \mathcal{F}^b \tag{4.103a}$$

$$= m_b\Big(\big(1+\Gamma_1\Gamma_2\big)\hat{z}_1 - \big(\Gamma_1+\Gamma_2\big)\hat{v}_b\big)\Big) + \hat{\omega}_b \times m_b\hat{v}_b - m_b\vec{G}_b - \vec{D}_b \tag{4.103b}$$

Making $\dot{V}_2$ negative definite:

$$\rightarrow \dot{V}_{IBC} = \dot{V}_2 = -\hat{z}_1{}^T\Gamma_1\hat{z}_1 - \hat{z}_2{}^T\Gamma_2\hat{z}_2 \quad < 0,\ \forall(\hat{z}_1,\hat{z}_2),\ \exists(\Gamma_1,\Gamma_2) > 0 \tag{4.103c}$$

Which leads to global asymptotic stability, assuming that the disturbance term $\vec{D}_b$ is known and can be compensated for. In the controller both Γ_1 and Γ_2 are positive symmetric control coefficient matrices to be optimized. Extending the backstepping rule and proposed LFC to incorporate an adaptive disturbance approximator $\hat{D}_b$, similar to the attitude controller in Sec:4.6.3. The approximation leads to an estimate error $\vec{D}_\Delta$, assuming that physical disturbances $\dot{\vec{D}}_b$ are far slower than the control dynamics; $\dot{\vec{D}}_b << \dot{\hat{D}}_b$.

$$\vec{D}_\Delta = \vec{D}_b - \hat{D}_b \qquad \in \mathcal{F}^b \tag{4.104a}$$

$$\therefore \dot{\vec{D}}_\Delta = \dot{\vec{D}}_b - \dot{\hat{D}}_b \approx \vec{0} - \dot{\hat{D}}_b = -\dot{\hat{D}}_b\Big|_{\dot{\vec{D}}_b \approx \vec{0}} \tag{4.104b}$$

The control law then designs a force, using that disturbance observer $\hat{D}_b$:

$$\vec{F}_{ABC} = m_b\big(\hat{z}_1 - \Gamma_1\hat{v}_b + \Gamma_2\hat{z}_2\big) + \hat{\omega}_b \times m_b\hat{v}_b - m_b\vec{G}_b - \hat{D}_b \qquad \in \mathcal{F}^b \tag{4.104c}$$

Proposing an LFC extended from the IBC case which includes that disturbance estimate error $\vec{D}_\Delta$ and finding it's derivative:

$$V_{ABC}(\hat{z}_1,\hat{z}_2,\vec{D}_\Delta) = V_{IBC}(\hat{z}_1,\hat{z}_2) + \frac{1}{2}\vec{D}_\Delta^T\Gamma_D^{-1}\vec{D}_\Delta \tag{4.105a}$$

$$= \frac{1}{2}\hat{z}_1{}^T\hat{z}_1 + \frac{1}{2}\hat{z}_2{}^T\hat{z}_2 + \frac{1}{2}\vec{D}_\Delta^T\Gamma_D^{-1}\vec{D}_\Delta \quad > 0,\ \forall(\hat{z}_1,\hat{z}_2,\vec{D}_\Delta) \tag{4.105b}$$

$$\Rightarrow \dot{V}_{ABC} = \hat{z}_1{}^T\dot{\hat{z}}_1 + \hat{z}_2{}^T\dot{\hat{z}}_2 + \vec{D}_\Delta^T\Gamma_D^{-1}\dot{\vec{D}}_\Delta \tag{4.105c}$$

Then substituting derivatives for $\dot{\hat{z}}_2$ and $\dot{\vec{D}}_\Delta$:

$$= -\hat{z}_1^T\Gamma_1\hat{z}_1 + \hat{z}_2^T\bigg(\hat{z}_1 - \Gamma_1\vec{v}_b - m_b^{-1}\big(-\vec{\omega}_b \times m_b\vec{v}_b + m_b\vec{G}_b + \vec{D}_b + \vec{F}_{ABC}\big)\bigg) - \vec{D}_\Delta^T\Gamma_D^{-1}\dot{\hat{D}} \tag{4.105d}$$

$$= -\hat{z}_1{}^T\Gamma_1\hat{z}_1 + \hat{z}_2{}^T\bigg(-\Gamma_2\hat{z}_2 - m_b^{-1}\big(\vec{D}_b - \hat{D}\big)\bigg) - \vec{D}_\Delta^T\Gamma_D^{-1}\dot{\hat{D}} \tag{4.105e}$$

$$= -\hat{z}_1{}^T\Gamma_1\hat{z}_1 - \hat{z}_2{}^T\Gamma_2\hat{z}_2 - m_b^{-1}\hat{z}_2^T\vec{D}_\Delta - \vec{D}_\Delta^T\Gamma_D^{-1}\dot{\hat{D}} \tag{4.105f}$$

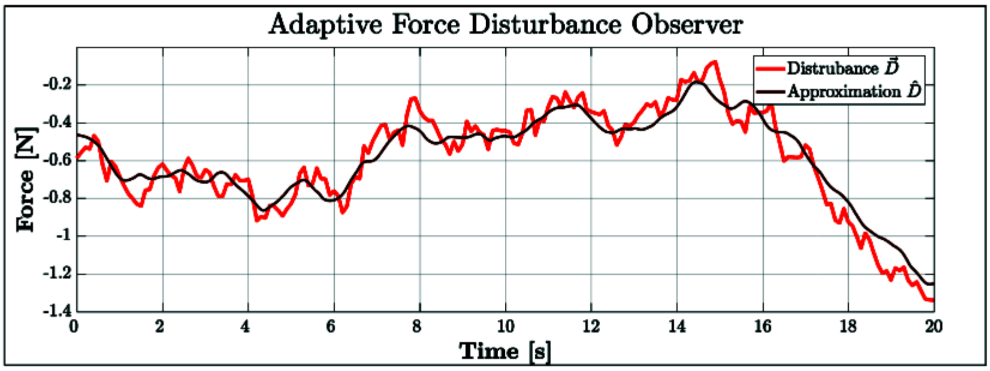
$$= -\hat{z}_1{}^T\Gamma_1\hat{z}_1 - \hat{z}_2{}^T\Gamma_2\hat{z}_2 - m_b^{-1}\vec{D}_\Delta^T\Gamma_D^{-1}\big(\Gamma_D\hat{z}_2 + \dot{\hat{D}}\big) \tag{4.105g}$$

Then, a self-evident choice for the disturbance update law would be; $\dot{\hat{D}} = -m^{-1}\Gamma_D\hat{z}_2$, which ensures asymptotic stability. Substituting that into the LFC derivative Eq:4.105g produces:

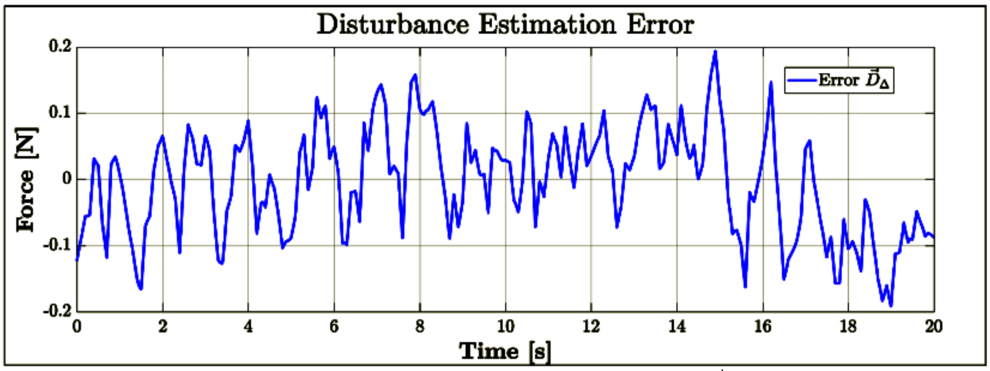
$$\dot{\hat{D}} = -m_b^{-1}\Gamma_D\hat{z}_2 = -m_b^{-1}\Gamma_D\Big(\Gamma_1\hat{z}_1 - \vec{v}_b\Big) \tag{4.106a}$$

$$\therefore \dot{V}_{ABC} = -\hat{z}_1{}^T\Gamma_1\hat{z} - \hat{z}_2{}^T\Gamma_2\hat{z}_2 \quad < 0,\ \forall(\hat{z}_1,\hat{z}_2,\vec{D}_\Delta),\ \exists(\Gamma_1,\Gamma_2,\Gamma_\Delta) \tag{4.106b}$$

The disturbance observer tracks a general single axis directional force disturbance as illustrated in Fig:4.7a. The disturbance is a combined fluctuating wind force and vector field; the model of which is later described in Sec:6.6.2. Note that Fig:4.7 tracks an *open loop* disturbance on a vehicle stabilized steady state. An estimation error for the deviation from the physical disturbance is plotted in Fig:4.7b. Again there is a damping between the physical and approximated forces; no new state information is used to estimate signals in both Fig:4.6a and Fig:4.7a for attitude and position disturbances respectively. Adaptive observers in Eq:4.80 and Eq:4.106a simply introduce additional free parameters to the control loop…



(a) Force disturbance observer



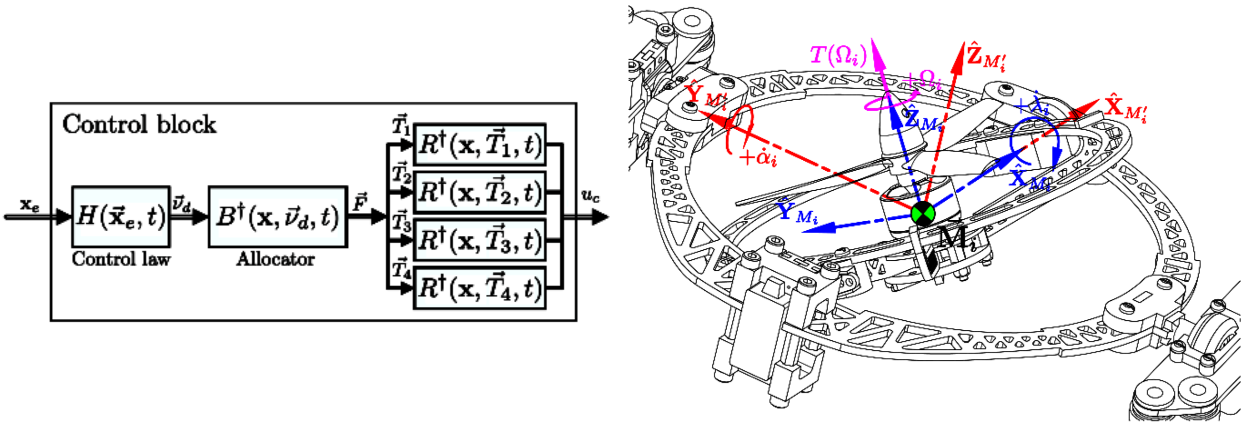
(b) Force disturbance error deviation $\vec{D}_\Delta$

Figure 4.7: Adaptive disturbance observer example

Chapter 5

Controller Allocation

Higher level attitude and position controllers (from Sec:4.6 and Sec:4.7 respectively) design a virtual control input $H(\vec{\mathbf{x}}_e, t) = \vec{\nu}_d = [\vec{F}_d \ \vec{\tau}_d]^T$ to be applied by the vehicles actuators. The system's overactuation was previously described in Sec:4.2; this chapter aims to solve for explicit actuator positions from that virtual input. A simplified allocation block, reduced from Fig:4.2,is shown in Fig:5.1a.



(a) Allocation block **(b)** Single thrust vector construction

Figure 5.1: Actuator allocation

A distribution rule is needed to *allocate* out physical actuator positions $u_c \in \mathbb{U}$ to command that input $\vec{\nu}_c$, from Eq:4.6. As mentioned previously (pseudo) inversion based allocation requires an affine actuator effectiveness function; the allocator is abstracted to first solve for four thrust vectors which are applied by each motor module, Eq:4.7.

$$B^\dagger(\vec{\mathbf{x}}, t)\vec{\nu}_d = \begin{bmatrix} \vec{T}_1 & \vec{T}_2 & \vec{T}_3 & \vec{T}_4 \end{bmatrix}^T \tag{5.1}$$

Each 3-D thrust vector is then used to solve for each module's propeller speed in RPM and both servo rotational positions in rad; undoing rotation applied by the motor module's structure in Fig:5.1b.

$$u \cdot i = \begin{bmatrix} \Omega_i & \lambda_i & \alpha_i \end{bmatrix}^T = R^\dagger(\vec{\mathbf{x}}, \vec{T}_i, t) \quad \text{for } i \in [1:4] \tag{5.2}$$

5.1 Generalized allocation

Regular, unconstrained control allocation is solved as an optimization problem; shown in [67, 100]. The aim is to minimize deviation (*slack*) between the virtual and commanded control inputs, $\vec{\nu}_d$ and $\vec{\nu}_c$ respectively. For the control law's virtual input $\vec{\nu}_d = \mathcal{H}(\vec{\mathbf{x}}_e, t)$ the optimization is then posed as:

$$\underset{u \in \mathbb{U}^m,\ s \in \mathbb{R}^n}{min} \big(\|Q_s\| \big) \ \text{ such that } \ \vec{\nu}_d - \vec{\nu}_c = \mathcal{H}(\vec{\mathbf{x}}_e, t) - B(\vec{\mathbf{x}}, t, u) \triangleq s \tag{5.3}$$

Where $u \in \mathbb{U}^m$ is the dimension of the actuator set and $\vec{\mathbf{x}}$, $\vec{\nu}_d$, $\vec{\nu}_c$, s are each the same dimension of virtual plant's input; $\in \mathbb{R}^n$ where n is the degrees of freedom the state has. In this case $u \in \mathbb{U}^{12}$ for twelve actuators and $\vec{\mathbf{x}} \in \mathbb{R}^6$ for the 6-DOF rigid body.

In Eq:5.3, $\|Q_s\|$ is the cost function prioritizing the slack variable; s. Typically that cost is just the L_2 norm of the slack. Overactuation asserts that there exists an entire set of suitable actuator values which are all solutions to Eq:5.3. Solving for explicit actuator positions requires an introduction of a secondary cost function or control objective $J(\vec{\mathbf{x}}, t, u)$ to refine the solution to of Eq:5.3.

$$\underset{u\in\mathbb{U}^{12},\ s\in\mathbb{R}^6}{min} \big(||Q_s|| + J(\vec{\mathbf{x}}, u, t)\big) \ \text{ such that } \ \mathcal{H}(\vec{\mathbf{x}}_e, t) - B(\vec{\mathbf{x}}, u, t) = s \tag{5.4}$$

That secondary control objective $J(\vec{\mathbf{x}}, t, u)$ and its associated *explicit* solution to Eq:5.4 is the subject of control allocation. Not much work has been done on overallocation for aerospace vehicles outside the field of satellite attitude control (Sec:1.2.2 for examples). Often satellites are over actuated for the sake of fault tolerance and redundancy [6, 84]. Actuator rate constraints can be further introduced such that u is limited by Δu, constraining sequential actuator position changes.

$$\therefore \underset{u\in\mathbb{U}^{12},\ s\in\mathbb{R}^6}{min} \big(||Q_s|| + J(\vec{\mathbf{x}}, u, t)\big) \ \text{ s.t } \ \mathcal{H}(\vec{\mathbf{x}}_e, t) - B(\vec{\mathbf{x}}, u, t) = s$$
$$\text{and subject to } \ u = u_{n-1} + \Delta u,\ \Delta u \in \mathbb{C} \tag{5.5}$$

Allocation *inverts* the effectiveness of the actuator set, $B(\vec{\mathbf{x}}, u, t)$ in Eq:4.4, to find actuator positions which satisfy the virtual control input. Inverting the allocation requires 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 an affine matrix, reduces to:

$$\begin{bmatrix} \vec{F}_d \\ \vec{\tau}_d \end{bmatrix} = \nu_d = \mathcal{H}(\vec{\mathbf{x}}_e, t) \Longleftrightarrow B(\vec{\mathbf{x}}, u, t) = B'(\vec{\mathbf{x}}, t) u_c = \vec{\nu}_c = \begin{bmatrix} \vec{F}_c(\hat{u}) \\ \vec{\tau}_c(\hat{u}) \end{bmatrix} \tag{5.6}$$

The allocator commands an actuator setpoint u_c which leads to actuator position estimates $\hat{u}_c = C(s)u_c$ from the actuator block transfer function defined in Eq:2.43. In Eq:5.6 state dimensions are such that $(\vec{\nu}_d, \vec{\nu}_c) \in \mathbb{R}^n, \mathbb{U} \in \mathbb{R}^m$, and $B \in \mathbb{R}^{m\times n}$. That abstraction to a multiplicative $B'(\vec{\mathbf{x}}, t)u_c$ in Eq:4.7 makes addressing the allocation conceptually simpler, accommodating the use of inversion based allocation laws (Sec:5.3.1-5.3.3).

5.2 Thrust vector inversion

The rotation *inversion* function $R^{\dagger}(\vec{\mathbf{x}}, \vec{F}_i, t)$ to solve for physical actuator positions to be commanded $u_c(i) = [\Omega_i \ \lambda_i \ \alpha_i]^T$ is as yet undefined. Assuming for now there is some allocation rule that, from $\vec{\nu}_d$, designs well four decomposed stabilizing 3-D thrust vectors $\vec{T}_{1\to 4}$ to be actuated by each motor module. It then follows that each of those four thrust vectors relate to their individual associated actuator positions through a quaternion *rotation*, not transformation:

$$\vec{T}_i = Q_{M_i} \otimes \vec{T}(\Omega_i) \otimes Q^*_{M_i} \qquad \in \mathcal{F}^b \tag{5.7a}$$

$$= Q_z(\sigma_i)Q_y(\alpha_i)Q_x(\lambda_i) \otimes \vec{T}(\Omega_i) \otimes Q^*_x(\lambda_i)Q^*_y(\alpha_i)Q^*_z(\sigma_i) \tag{5.7b}$$

Where each motor thrust vector, $\vec{T}(\Omega_i)$, is calculated using BEM thrust coefficients, Eq:3.32a with coefficients from Fig:3.5. The thrust produced is exclusively in the $\hat{Z}_{M_i}$ direction of the motor module's frame $\mathcal{F}^{M_i}$:

$$\vec{T}(\Omega_i) = \begin{bmatrix} 0 & 0 & T(\Omega_i) \end{bmatrix}^T = \begin{bmatrix} 0 \\ 0 \\ C_T(J)\rho\Omega_i^2 D^4 \end{bmatrix} \qquad \in \mathcal{F}^{M_i} \tag{5.7c}$$

Seeing that quaternion rotation (or *transformation*) operators change the reference frame but retains the vector operand's magnitude, it follows that $T(\Omega_i)$, and by extension the propeller speed Ω_i, can be found:

$$|\vec{T_i}| = \sqrt{\left\| \begin{bmatrix} T_x & T_y & T_z \end{bmatrix} \right\|} = \sqrt{T_x^2 + T_y^2 + T_z^2} = |T(\Omega_i)| = |C_T(J)\rho\Omega_i^2 D^4| \tag{5.8a}$$

$$\rightarrow \Omega_i = \sqrt{\frac{|\vec{T_i}|}{C_T(J)\rho D^4}} = \sqrt{\frac{\sqrt{T_x^2 + T_y^2 + T_z^2}}{C_T(J)\rho D^4}} \tag{5.8b}$$

Reversing (or *undoing*) that transformation from the motor module's frame to body frame in Eq:5.7a:

$$\vec{T}(\Omega_i) = Q_z^*(\sigma_i)Q_y^*(\alpha_i)Q_x^*(\lambda_i) \otimes \vec{T_i} \otimes Q_x(\lambda_i)Q_y(\alpha_i)Q_z(\sigma_i) \qquad \in \mathcal{F}^{M_i} \tag{5.9a}$$

$$\rightarrow \vec{T}(\Omega_i) = Q_{M_i}^* \otimes \vec{T_i} \otimes Q_{M_i} \qquad \in \mathcal{F}^{M_i} \tag{5.9b}$$

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, $\breve{T}_i$ and $\breve{T}(\Omega_i)$; then the following relationship can be used to construct a relative quaternion:

$$\breve{T}_i \triangleq \frac{\vec{T_i}}{|\vec{T_i}|} = \frac{\vec{T_i}}{\sqrt{T_x^2 + T_y^2 + T_z^2}} \qquad \in \mathcal{F}^b \tag{5.10a}$$

$$\breve{T}(\Omega_i) \triangleq \frac{\vec{T}(\Omega_i)}{|\vec{T}(\Omega_i)|} = \frac{\vec{T}(\Omega_i)}{|C_T(J)\rho\Omega^2 D^4|} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \qquad \in \mathcal{F}^{M_i} \tag{5.10b}$$

$$\therefore Q_{M_i} = \begin{bmatrix} q_0 \\ \vec{q} \end{bmatrix} = \begin{bmatrix} 1 + \breve{T}_i \cdot \breve{T}(\Omega_i) \\ -\breve{T}_i \times \breve{T}(\Omega_i) \end{bmatrix} \tag{5.10c}$$

Where Eq:5.10c is a quaternion operator's definition, rotating a vector around a single Euler axis, Eq:3.57, when applied to two unit vectors. That quaternion can indeed be used to solve for relative pitch, roll and yaw Euler angles (see App:A.3). However, Eq:5.10c solves for the **shortest rotational path** between the two vectors, a sequenced Z-Y-X rotation is by no means the shortest possible rotation. Associated $[\phi,\ \theta,\ \psi]^T$ solutions to Eq:A.16 are of no consequence when solving for the sequentially applied rotation angles $[\lambda_i,\ \alpha_i,\ \sigma_i]^T$, where σ_i is a known orthogonal multiplicate. Furthermore, when considering a sequenced Z-Y-X quaternion, angular operands cannot be extracted without applying significantly complex trigonometric inversions:

$$Q_b \triangleq \begin{bmatrix} cos\frac{\psi}{2} \\ 0 \\ 0 \\ sin\frac{\psi}{2} \end{bmatrix} \otimes \begin{bmatrix} cos\frac{\theta}{2} \\ 0 \\ sin\frac{\theta}{2} \\ 0 \end{bmatrix} \otimes \begin{bmatrix} cos\frac{\phi}{2} \\ sin\frac{\phi}{2} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} c\frac{\psi}{2}c\frac{\theta}{2}c\frac{\phi}{2} + s\frac{\psi}{2}s\frac{\theta}{2}s\frac{\phi}{2} \\ c\frac{\psi}{2}c\frac{\theta}{2}s\frac{\phi}{2} - s\frac{\psi}{2}s\frac{\theta}{2}c\frac{\phi}{2} \\ c\frac{\psi}{2}s\frac{\theta}{2}c\frac{\phi}{2} + s\frac{\psi}{2}c\frac{\theta}{2}s\frac{\phi}{2} \\ s\frac{\psi}{2}c\frac{\theta}{2}c\frac{\phi}{2} - c\frac{\psi}{2}s\frac{\theta}{2}s\frac{\phi}{2} \end{bmatrix} = \begin{bmatrix} q_0 \\ q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} q_0 \\ \vec{q} \end{bmatrix} \tag{5.11a}$$

$$\therefore \vec{T_i} = \begin{bmatrix} c\frac{\psi}{2}c\frac{\theta}{2}c\frac{\phi}{2} + s\frac{\psi}{2}s\frac{\theta}{2}s\frac{\phi}{2} \\ c\frac{\psi}{2}c\frac{\theta}{2}s\frac{\phi}{2} - s\frac{\psi}{2}s\frac{\theta}{2}c\frac{\phi}{2} \\ c\frac{\psi}{2}s\frac{\theta}{2}c\frac{\phi}{2} + s\frac{\psi}{2}c\frac{\theta}{2}s\frac{\phi}{2} \\ s\frac{\psi}{2}c\frac{\theta}{2}c\frac{\phi}{2} - c\frac{\psi}{2}s\frac{\theta}{2}s\frac{\phi}{2} \end{bmatrix} \otimes \vec{T}(\Omega_i) \otimes \begin{bmatrix} s\frac{\psi}{2}s\frac{\theta}{2}s\frac{\phi}{2} + c\frac{\psi}{2}c\frac{\theta}{2}c\frac{\phi}{2} \\ s\frac{\psi}{2}s\frac{\theta}{2}c\frac{\phi}{2} - c\frac{\psi}{2}c\frac{\theta}{2}s\frac{\phi}{2} \\ -c\frac{\psi}{2}s\frac{\theta}{2}c\frac{\phi}{2} - s\frac{\psi}{2}c\frac{\theta}{2}s\frac{\phi}{2} \\ c\frac{\psi}{2}s\frac{\theta}{2}s\frac{\phi}{2} - s\frac{\psi}{2}c\frac{\theta}{2}c\frac{\phi}{2} \end{bmatrix} \tag{5.11b}$$

Instead; returning to rotation matrices to resolve the inverse transformation and reiterating that Euler angle equivalents for the servos are; $[\phi,\ \theta,\ \psi]^T \iff [\lambda_i,\ \alpha_i,\ \sigma_i]^T$. The rotation matrix transformation from $\mathcal{F}^{M_i}$ to $\mathcal{F}^b$, analogous to Eq:5.9b, is:

$$\vec{T_i} = R_z(\sigma_i)R_y(\alpha_i)R_x(\lambda_i)\vec{T}(\Omega_i) \qquad \in \mathcal{F}^b \tag{5.12a}$$

$$= \begin{bmatrix} c\sigma_i & -s\sigma_i & 0 \\ s\sigma_i & c\sigma_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\alpha_i & 0 & s\alpha_i \\ 0 & 1 & 0 \\ -s\alpha_i & 0 & c\alpha_i \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\lambda_i & -s\lambda_i \\ 0 & s\lambda_i & c\lambda_i \end{bmatrix} \vec{T}(\Omega_i) \tag{5.12b}$$

$$\therefore \vec{T_i} = \begin{bmatrix} c\sigma_i c\alpha_i & c\sigma_i s\alpha_i s\lambda_i - s\sigma_i c\lambda_i & c\sigma_i s\alpha_i c\lambda_i + s\sigma_i s\lambda_i \\ s\sigma_i c\alpha_i & s\sigma_i s\alpha_i s\lambda_i + c\sigma_i c\lambda_i & s\sigma_i s\alpha_i c\lambda_i - c\sigma_i s\lambda_i \\ -s\alpha_i & c\alpha_i s\lambda_i & c\alpha_i c\lambda_i \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ T(\Omega_i) \end{bmatrix} \tag{5.12c}$$

Where σ_i is an orthogonal multiple applying a rotation about the $\hat{Z}_b$ axis (Fig:2.9). Because the thrust vector $\vec{T}(\Omega_i)$ is only in the motor frame's $\hat{Z}_{M_i}$ direction, solving for servo angles is simplified.

$$\rightarrow \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} = \begin{bmatrix} s\sigma_i s\lambda_i + c\sigma_i s\alpha_i c\lambda_i \\ s\sigma_i s\alpha_i c\lambda_i - c\sigma_i s\alpha_i \\ c\alpha_i c\lambda_i \end{bmatrix} T(\Omega_i) \qquad \in \mathcal{F}^b \tag{5.12d}$$

Eq:5.12d then reduces further with $R_z(\sigma_i)$ rotation matrices already defined in Eq:2.16b. The following four trigonometric relationships exist for each motor module respectively:

$$\begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} s\alpha_1 c\lambda_1 \\ -s\lambda_1 \\ c\alpha_1 c\lambda_1 \end{bmatrix}, \begin{bmatrix} s\lambda_2 \\ s\alpha_2 c\lambda_2 \\ c\alpha_2 c\lambda_2 \end{bmatrix}, \begin{bmatrix} -s\alpha_3 c\lambda_3 \\ s\lambda_3 \\ c\alpha_3 c\lambda_3 \end{bmatrix}, \begin{bmatrix} -s\lambda_4 \\ -s\alpha_4 c\lambda_4 \\ c\alpha_4 c\lambda_4 \end{bmatrix} \end{bmatrix} T(\Omega_{1\rightarrow 4}) \tag{5.13}$$

It is then a simple trigonometric inversion to solve for both λ_i and α_i. For the example case of $i = 1$, the sequel holds true and can similarly be extended to the remaining modules. Firstly using $T(\Omega_i) = ||\vec{T}_i||$ and implementing a four quadrant secondary arctangent2 function. Wherein $arctan2(x, y)$ is the four-quadrant tangent inverse [40], producing the principle argument of the complex operands:

$$arctan2(x,\ y) = PR\ arg(x + y\hat{\imath}) = Arg(x + y\hat{\imath}) \tag{5.14}$$

The use of a full quadrature arctangent function is to find solutions for Euler angles that are not only acute. Each inverse would otherwise need generalized reciprocal solutions with parity checks to establish which quadrant the angle occurs in. Furthermore exploiting the fact that $arctan(x) \equiv arcsin(x/\sqrt{1-x^2})$:

$$\lambda_i = arctan2\Big(-T_y,\ \sqrt{||\vec{T}_i||^2 - T_y^2}\Big) \tag{5.15a}$$

$$\alpha_i = arctan2\big(T_x,\ T_z\big) \tag{5.15b}$$

Therefore, the secondary component of the control allocation block, $R^\dagger(\vec{\mathbf{x}}, \vec{T}_i, t)$ from Fig:4.2 is then summarized as a single rotation inversion function (in this case for $i = 1$):

$$\begin{bmatrix} \Omega_1 \\ \lambda_1 \\ \alpha_1 \end{bmatrix} = R^\dagger(\mathbf{x}, \vec{T}_1, t) \triangleq \begin{bmatrix} \Big(\sqrt{{T_x}^2 + {T_y}^2 + {T_z}^2}/C_T(J)\rho D^4\Big)^{\frac{1}{2}} \\ atan2(-{T_y}^2,\ ||\vec{T}_1||\sqrt{||\vec{T}_1||^2 - {T_y}^2}) \\ atan2(T_x,\ T_z||\vec{T}_1||) \end{bmatrix} \tag{5.16}$$

Rotation inversion equations for the remaining motor modules are included in App:A.4. The complete control block only requires one final abstracted allocation algorithm, $B^\dagger(\mathbf{x}, \vec{\nu}_d, t)$; which is now addressed...

5.3 Allocators

5.3.1 Pseudo Inverse Allocator

The simplest control allocation solution to Eq:5.4 stems from what is categorized as *inversion*, based on controller effort optimization [67]. The requirements for inversion based allocation is that the effectiveness function $B(\vec{\mathbf{x}}, u, t)$ is a linear relationship which can be abstracted to $B'(\vec{\mathbf{x}}, t)u$. The objective is for some commanded control input $\vec{\nu}_c$ to find an inverted matrix $B^\dagger(\vec{\mathbf{x}}, t)$ such that for a virtual control input $\vec{\nu}_d$:

$$\vec{\nu}_d = H(\vec{\mathbf{x}}_e, t) \iff B'(\vec{\mathbf{x}}, t)u_c = \vec{\nu}_c \tag{5.17a}$$

$$\rightarrow u_c = B^\dagger(\vec{\mathbf{x}}, t)\vec{\nu}_d \tag{5.17b}$$

$$\therefore \vec{\nu}_c = B'(\vec{\mathbf{x}}, t)B^\dagger(\vec{\mathbf{x}}, t)\vec{\nu}_d \tag{5.17c}$$

Where the inverse namesake is as a result of the identity condition for both the effectiveness matrix $B'(\vec{\mathbf{x}},t)$ and it's inverse $B^{\dagger}(\vec{\mathbf{x}},t)$:

$$B'(\vec{\mathbf{x}},t)B^{\dagger}(\vec{\mathbf{x}},t) = \mathbb{I}_{m\times m} \tag{5.17d}$$

Or more generally, and without the dependency of the affine linearity:

$$u_c = B^{\dagger}(\vec{\mathbf{x}},\vec{\nu}_d,t) \tag{5.17e}$$

In Eq:5.17, the multiplicative effectiveness matrix $B'(\vec{\mathbf{x}},t)$ has the dimension $\in \mathbb{R}^{m\times n}$. In the case of overallocation there are more actuators than degrees of freedom, or that $m > n$ for $u \in \mathbb{R}^m$ and $\vec{\mathbf{x}} \in \mathbb{R}^n$. That implies that B' has full rank, therefore finding the inversion of $B^{\dagger}(\mathbf{x},t)$ is not trivial. Choosing the secondary control objective, $J(\vec{\mathbf{x}},u,t)$ in Eq:5.4, to be a quadratic cost function; actuator positions u_c to be commanded can be solved as a linear least squares problem. The quadratic least squares optimization aims to minimize controller effort (*magnitude*):

$$J(\vec{\mathbf{x}},u_c,t) = \min_{u_c\in\mathbb{U}}\frac{1}{2}\big(u_c-u_p\big)^T W\big(u_c-u_p\big) \ \text{ such that } \ \vec{\nu}_c = B'(\vec{\mathbf{x}},t)u_c \tag{5.18}$$

The least squares solution, [47], to Eq:5.18 then minimizes the commanded actuator actuator effort, $\|u\|_c$. This means that thrusts $\left\|\vec{T}_{1\to4}\right\|$ from Eq:5.1 are minimized. The magnitude of each thrust vector commanded to a motor module is affected by the propeller's rotational speed Ω_i. Effectively this results in an allocator that prioritizes pitching or rolling both servos λ_i and α_i over adjusting the propeller's velocity.

The positive symmetrical weighting matrix W in Eq:5.18 biases certain actuators (thrust components in this case) and creates its own class of inversion allocator, presented in Sec:5.3.3. Similarly u_p is the preferred actuator value to which the system naturally tends; discussed in Sec:5.3.2. For an inversion matrix $B^{\dagger}(\vec{\mathbf{x}},t)$ actuator thrust components are found:

$$\underset{u\in\mathbb{U}}{\left[\vec{T}_{1\to4}\right]} = \Big(\mathbb{I}_{m\times m} - CB(\vec{\mathbf{x}},t)\Big)\vec{T}_p + C\vec{\nu}_d \tag{5.19a}$$

$$C = W^{-1}B^T(\vec{\mathbf{x}},t)\big(B(\vec{\mathbf{x}},t)W^{-1}B^T(\vec{\mathbf{x}},t)\big)^{-1} \tag{5.19b}$$

Where $\vec{T}_p \in \mathbb{R}^{1\times12}$ are preferred thrust component values. The solution in Eq:5.19 is a *generalized inverse* with weighted actuators components and preferred values. In the case where no weightings nor preferred actuator values are specified, $W = \mathbb{I}_{n\times n}$ and $u_p = \vec{T}_p = \vec{0}$, the solution reduces:

$$\vec{T}_{1\to4} = B^T(\vec{\mathbf{x}},t)\big(B(\vec{\mathbf{x}},t).B^T(\vec{\mathbf{x}},t)\big)^{-1}\vec{\nu}_d \tag{5.20a}$$

$$= B^{\ddagger}(\vec{\mathbf{x}},t)\vec{\nu}_d \ \ , B^{\ddagger} \in \mathbb{R}^{6\times12} \tag{5.20b}$$

The simplified case in Eq:5.20 is termed a Moore-Penrose or pseudo-inversion of the actuator effectiveness matrix $B'(\vec{\mathbf{x}},t)$, [78]. Pseudo-inversion is the simplest allocation rule to implement, in most cases controller effort optimization is a satisfactory constraint without any additional weights or preferred values. For an effectiveness $B'(\vec{\mathbf{x}},t)$ matrix defined in Eq:4.7, the pseudo-inversion is:

$$B'(\vec{\mathbf{x}},t) = \begin{bmatrix} \mathbb{I}_{3\times3} & \mathbb{I}_{3\times3} & \mathbb{I}_{3\times3} & \mathbb{I}_{3\times3} \\ [\vec{L}_1]_\times & [\vec{L}_2]_\times & [\vec{L}_3]_\times & [\vec{L}_4]_\times \end{bmatrix} \quad \in \mathbb{R}^{12\times6} \tag{5.21a}$$

$$\therefore \vec{T}_{1\to4} = B^T\big(B.B^T\big)^{-1}\vec{\nu}_d = B^{\ddagger}(\vec{\mathbf{x}},t)\vec{\nu}_d \tag{5.21b}$$

Recalling that each motor modules displacement is at a distance $L_{arm} = 195.16$ mm from Fig:2.17. Then each module has a vector $\vec{L}_{1,3} = \begin{bmatrix}\pm195.16 & 0 & 0\end{bmatrix}^T$ and $\vec{L}_{2,4} = \begin{bmatrix}0 & \pm195.16 & 0\end{bmatrix}$. Each cross product vector in Eq:5.21a is defined from Eq:2.8c as:

$$\big[\vec{L}_i\big]_\times \triangleq \begin{bmatrix} 0 & -L_z & L_y \\ L_z & 0 & -L_x \\ -L_y & L_x & 0 \end{bmatrix} \tag{5.21c}$$

The numeric and constant pseudo-inverse matrix is then:

$$\therefore B^{\ddagger}(\vec{\mathbf{x}},t) = \begin{bmatrix} \frac{1}{4} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{4} & 0 & 0 & 0 & \frac{1}{4L} \\ 0 & 0 & \frac{1}{4} & 0 & \frac{-1}{2L} & 0 \\ \frac{1}{4} & 0 & 0 & 0 & 0 & \frac{-1}{4L} \\ 0 & \frac{1}{4} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{4} & \frac{1}{2L} & 0 & 0 \\ \frac{1}{4} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{4} & 0 & 0 & 0 & \frac{-1}{4L} \\ 0 & 0 & \frac{1}{4} & 0 & \frac{1}{2L} & 0 \\ \frac{1}{4} & 0 & 0 & 0 & 0 & \frac{1}{4L} \\ 0 & \frac{1}{4} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{4} & \frac{-1}{2L} & 0 & 0 \end{bmatrix} \tag{5.21d}$$

$$= \begin{bmatrix} 0.250 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.250 & 0.000 & 0.000 & 0.000 & 0.250 \\ 0.000 & 0.000 & 0.250 & 0.000 & -2.562 & 0.000 \\ 0.250 & 0.000 & 0.000 & 0.000 & 0.000 & -1.281 \\ 0.000 & 0.250 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.250 & 2.562 & 0.000 & 0.000 \\ 0.250 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.250 & 0.000 & 0.000 & 0.000 & -1.281 \\ 0.000 & 0.000 & 0.250 & 0.000 & 2.562 & 0.000 \\ 0.250 & 0.000 & 0.000 & 0.000 & 0.000 & 1.281 \\ 0.000 & 0.250 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.250 & -2.562 & 0.000 & 0.000 \end{bmatrix} \tag{5.21e}$$

Pseudo-inversion allocation guarantees that $\vec{T}_{1\to 4} = B^{\dagger}(\vec{\mathbf{x}},t)\vec{\nu}_d$ produces a feasible set of control thrust vectors $\vec{T}_{1\to 4}$ for some virtual control input $\vec{\nu}_d = \mathcal{H}(\vec{\mathbf{x}}_e,t)$. Those thrust vectors, each $\vec{T}_i$, is then solved for as explicit actuator positions $[\Omega_i, \lambda_i, \alpha_i]^T = R^{\dagger}(\mathbf{x}, \vec{T}_i, t)$ using Eq:5.16. That constructs an actuator matrix $u_c \in \mathbb{U} \in \mathbb{R}^{12}$ which will physically command $\vec{\nu}_c = B(\vec{\mathbf{x}},t)u_c$.

The actuator's effectiveness matrix $B(\vec{\mathbf{x}},t,u)$ does not necessarily have to be static (or affine) with respect to either the state vector $\vec{\mathbf{x}}$ or time t. However it was abstracted to such a static relationship to simplify the actuation process. Allocation in Eq:5.21 is the most simplified case of the least squares quadratically optimized equation for Eq:5.4 and is used as the base reference allocation law.

The direct (*pseudo*) inversion solution ensures the commanded virtual control input is met and that acutators are not 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* allocation to be performed numerically online, enforcing saturation for atleast some actuators and achievement of control objectives, [67]. Such an approach is avoided here as completely saturating any actuator is not desirable; moreover online allocation is outside the scope of applied allocation rules here, static explicit allocation rules are only investigated…

5.3.2 Priority Norm Inverse Allocator

Choosing a preferred actuator position from Eq:5.4 produces what is termed as a *priority norm* allocator. Specifically when $u_p = \vec{T}_p \neq \vec{0} \in \mathbb{U}$. An obvious choice for that value are the conditions required for stable hovering, those which simply keep the quadcopter airborne. The desired effect is that the controller constantly tends towards stability with small first and second order rates. There are, however, some intricacies which must be discussed with respect to what hovering conditions are.

For a vehicle with a weight m_b a net gravitational force acts on the vehicle through its center of gravity in the inertial frame; $-m_b\vec{G}_I \in \mathcal{F}^I$. Assuming torques produced by the eccentric gravitational center (Eq:3.134) at steady state are going to be constant, hovering conditions are then simply:

$$\vec{\nu}_I = \begin{bmatrix} \vec{F}_p \\ \vec{\tau}_p \end{bmatrix} = \begin{bmatrix} m_b\vec{G}_I \\ \Delta C.G \times m_b\vec{G}_I \end{bmatrix} \qquad \in \mathcal{F}^I \tag{5.22}$$

If preferred the hover conditions are taken with respect to the inertial frame as in Eq:5.22, then the resultant preferred actuator positions are independent from the body's current or desired attitude setpoint. The control loop then naturally tends towards a rest state attitude at $Q_d = [1\ \vec{0}\,]^T$ with $\vec{\nu}_I \equiv \vec{\nu}_b$. The commanded body frame control input is equivalent to the inertial frame hovering conditions. The free body diagram in Fig:5.2 illustrates a preferred hovering condition in the inertial frame and its tendency toward a natural state at the attitude's origin.

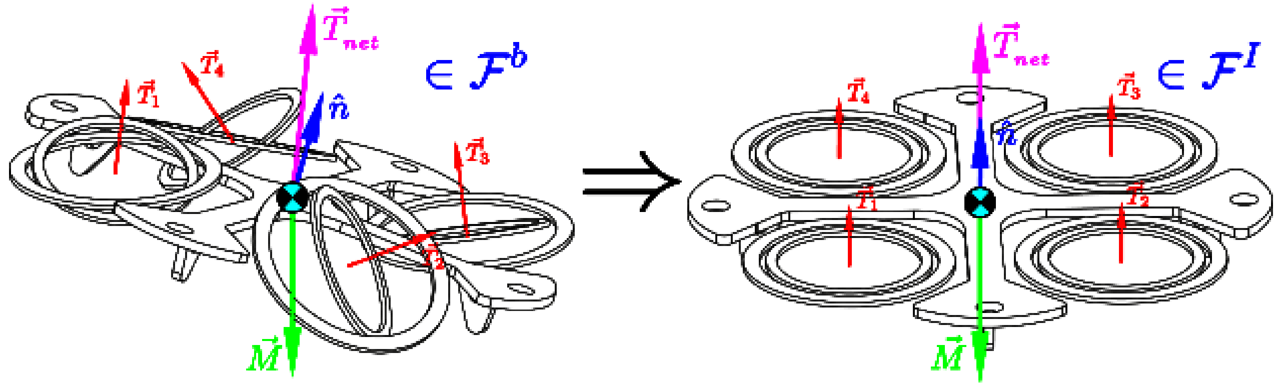


Figure 5.2: Hover conditions W.R.T the inertial frame $\mathcal{F}^I$

Alternatively, the hover conditions could be defined with respect to the body frame; being a function of the body's attitude illustrated in Fig:5.3. 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.

$$m_b\vec{G}_b \triangleq m_b Q_b^* \otimes \vec{G}_I \otimes Q_b \qquad \in \mathcal{F}^b \tag{5.23a}$$

$$\vec{\nu}_b = \begin{bmatrix} \vec{F}_p \\ \vec{\tau}_p \end{bmatrix} = \begin{bmatrix} m_b\vec{G}_b \\ \Delta C.G \times m_b\vec{G}_b \end{bmatrix} \qquad \in \mathcal{F}^b \tag{5.23b}$$

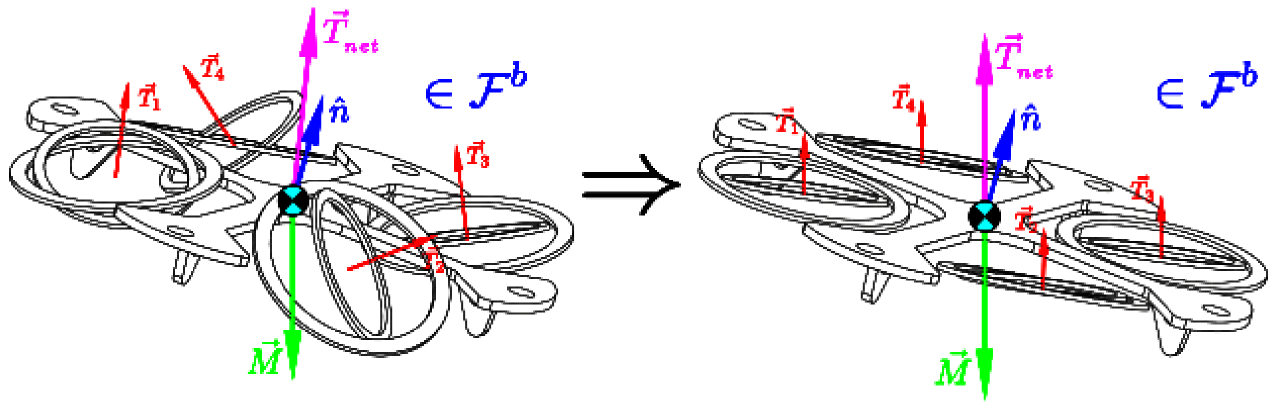


Figure 5.3: Hover conditions W.R.T the body frame $\mathcal{F}^b$

Specific module thrust values to be commanded are then solved for using Eq:5.22 and Eq:5.23 with pseudo inversion from Eq:5.20. The two solutions are then as follows:

$$\vec{T}_p^I = B^\dagger(\mathbf{x}, \vec{\nu}_I, t) \ \ \text{for hover in Fig5.2} \tag{5.24a}$$

$$\vec{T}_p^b = B^\dagger(\mathbf{x}, \vec{\nu}_b, t) \ \ \text{for hover in Fig:5.3} \tag{5.24b}$$

Both actuator matrices are then applied to Eq:5.19 and could be combined with a non-diagonal weighting matrix.

$$\left[\vec{T}_{1\to4}\right]_{u\in\mathbb{U}} = \left(\mathbb{I}_{m\times m} - CB(\vec{\mathbf{x}},t)\right)\vec{T}_p + C\vec{\nu}_d \tag{5.25a}$$

$$C = W^{-1}B^T(\vec{\mathbf{x}},t)\big(B(\vec{\mathbf{x}},t)W^{-1}B^T(\vec{\mathbf{x}},t)\big)^{-1} \tag{5.25b}$$

Applying the the inverse rotation operator $R^\dagger$ from Eq:5.16 to the above solves for propeller speeds and servo rotational positions in both respective cases. The physical consequences of either preferred hovering condition and its associated actuator positions are demonstrated in simulation in Sec:6.7. Priority actuator positions are not tested together with weighting matrices, the two are compared independently…

5.3.3 Weighted Pseudo Inverse Allocator

Adding weights to the inversion in Eq:5.19 but regarding preferred actuator positions as negligible, or that $\vec{T}_p = \vec{0}$, produces a *weighted pseudo inverse* allocator. Each weight in W biases a particular actuator's action in u, the positive symmetrical weighting matrix is square with respect to the actuator dimension here $W \in \mathbb{R}^{12\times12}$ but more generally $W \in \mathbb{R}^{m\times m}$. The Moore-Penrose inversion (Eq:5.20) assumes that each actuator is weighted equally. Such a case makes the weighting matrix W purely diagonal; $W_\ddagger \triangleq \mathbb{I}_{m\times m}$.

A weighting matrix could change adaptively over time or state dependency; 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. Adaptive weighting is not considered or discussed as that is out of the scope for this work and pertains more to FTC [6].

$$\begin{array}{cc} & \begin{array}{cccc} \vec{T}_1 \Downarrow & \vec{T}_2 \Downarrow & \vec{T}_3 \Downarrow & \vec{T}_4 \Downarrow \end{array} \\ \begin{array}{c} \vec{T}_1 \Rightarrow \\ \vec{T}_2 \Rightarrow \\ \vec{T}_3 \Rightarrow \\ \vec{T}_4 \Rightarrow \end{array} & \begin{bmatrix} \begin{bmatrix} W_{1:1} & W_{1:2} & W_{1:3} \\ W_{1:4} & W_{1:5} & W_{1:6} \\ W_{1:7} & W_{1:8} & W_{1:9} \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} W_{5:1} & W_{5:2} & W_{5:3} \\ W_{5:4} & W_{5:5} & W_{5:6} \\ W_{5:7} & W_{5:8} & W_{5:9} \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} W_{2:1} & W_{2:2} & W_{2:3} \\ W_{2:4} & W_{2:5} & W_{2:6} \\ W_{2:7} & W_{2:8} & W_{2:9} \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} W_{6:1} & W_{6:2} & W_{6:3} \\ W_{6:4} & W_{6:5} & W_{6:6} \\ W_{6:7} & W_{6:8} & W_{6:9} \end{bmatrix} \\ \begin{bmatrix} W_{5:1} & W_{5:2} & W_{5:3} \\ W_{5:4} & W_{5:5} & W_{5:6} \\ W_{5:7} & W_{5:8} & W_{5:9} \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} W_{3:1} & W_{3:2} & W_{3:3} \\ W_{3:4} & W_{3:5} & W_{3:6} \\ W_{3:7} & W_{3:8} & W_{3:9} \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} W_{6:1} & W_{6:2} & W_{6:3} \\ W_{6:4} & W_{6:5} & W_{6:6} \\ W_{6:7} & W_{6:8} & W_{6:9} \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} W_{4:1} & W_{4:2} & W_{4:3} \\ W_{4:4} & W_{4:5} & W_{4:6} \\ W_{4:7} & W_{4:8} & W_{4:9} \end{bmatrix} \end{bmatrix} \end{array}$$

Figure 5.4: Weighting matrix biasing

Each coefficient in W determines how the least squares solution to Eq:5.4 preferentially biases a particular actuator; here the weighting matrix's divisions correlate to mixed module thrust vectors. Each 3×3 diagonal group for $W_{1\to4}$ relate to individual thrust component biasing (T_{ix}, T_{iy}, T_{iz}) whilst off-centre 3×3 groupings mix separate thrust terms $\vec{T}_{1\to4}$.

Pseudo-inversion exactly matches the virtual control input $\vec{\nu}_d = B(\mathbf{x}, u_c, t) = \vec{\nu}_c$ so long as the actuators are not saturated. Biasing actuators could result in a slack within that control requirement. Such a case could potentially destabilize the trajectory tracking. Short of iteratively processing variable weights until a viable solution is found, a constraint on the nature of the weighting matrix needs to be introduced to avoid purposefully imposed control slack.

So long as each coefficient group's row and column vectors each sum to unity, the designed control inputs will be met. Namely $\sum(W_{row}) = \sum(W_{col}) = 1$. Physically the resultant thrusts and torque (thrust differentials) would be balanced amongst similarly directed components. One final constraint on the weighting matrix is that only opposing module's thrust vectors can be mixed; $\vec{T}_1 \& \vec{T}_3$ and $\vec{T}_2 \& \vec{T}_4$.

Priority biases for thrust vector components in the $\hat{X}_b$ and $\hat{Y}_b$ would, in theory, prioritizes using pitch or roll servos, λ_i and α_i, in lieu of changing the propeller's speed Ω_i. However, given that a quadratic optimization on the actuator effort in Eq:5.18, the weighting matrix's effect is in practice going to be diminished. Selection of weighting coefficients needs to be designed as per some heuristic. A suitable objective for the allocation block is to minimize each actuator's transfer rate, attempting to improve the net actuator block's bandwidth. A proposed set of weighting coefficients could then be simulated and penalized from actuator slew rate times together with a slack variable norm to ensure that a control objective is still met:

$$\int_{t_0}^{\infty} \big(a \left\| t^{\nu_d - \nu_c} - 1 \right\| + b \left\| s \right\| \big).dt \tag{5.26}$$

Where the integral in Eq:5.26 is run over the time $t_0 \rightarrow \infty$ for the length of a single simulation cycle. As such, the weighting matrix coefficients aim to reduce the transient time for the actuator block to settle whilst ensuring stability is not compromised. However, actuator rates are more dependent on the rotation inverse $R^{\dagger}(\vec{\mathbf{x}}, t)$, so the effect of a weighting matrix introduced to the allocation rule is not expected to be significant…

Chapter 6

Simulations and Results

6.1 Simulator description

The proposed attitude and position control laws, together with the system's equations of motion including each actuator's transfer function, were all tested in simulation to determine a particular controller's efficacy. The rigid-body equations of motion from Sec:3.1.1, with nonlinearities from Sec:3.2 and multibody responses from Sec:3.4, were incorporated into a high fidelity simulation environment. Closely matching the dynamics of the physical quadrotor prototype proposed in Sec:2.1; where measurement data produced by tests in Sec:3.4.2 provide a degree of confidence in the simulation's accuracy. The consolidated quaternion dynamics in Sec:3.5 formed the basis of the simulation; building a loop extended from the control structure in Fig:4.2. Each control law is optimized first without the effect of the servo's 180° saturation limit. Limiting the servos was a conscientious design decision and as such its effects are investigated in Sec:6.8; but for now the servos are treated as continuous...

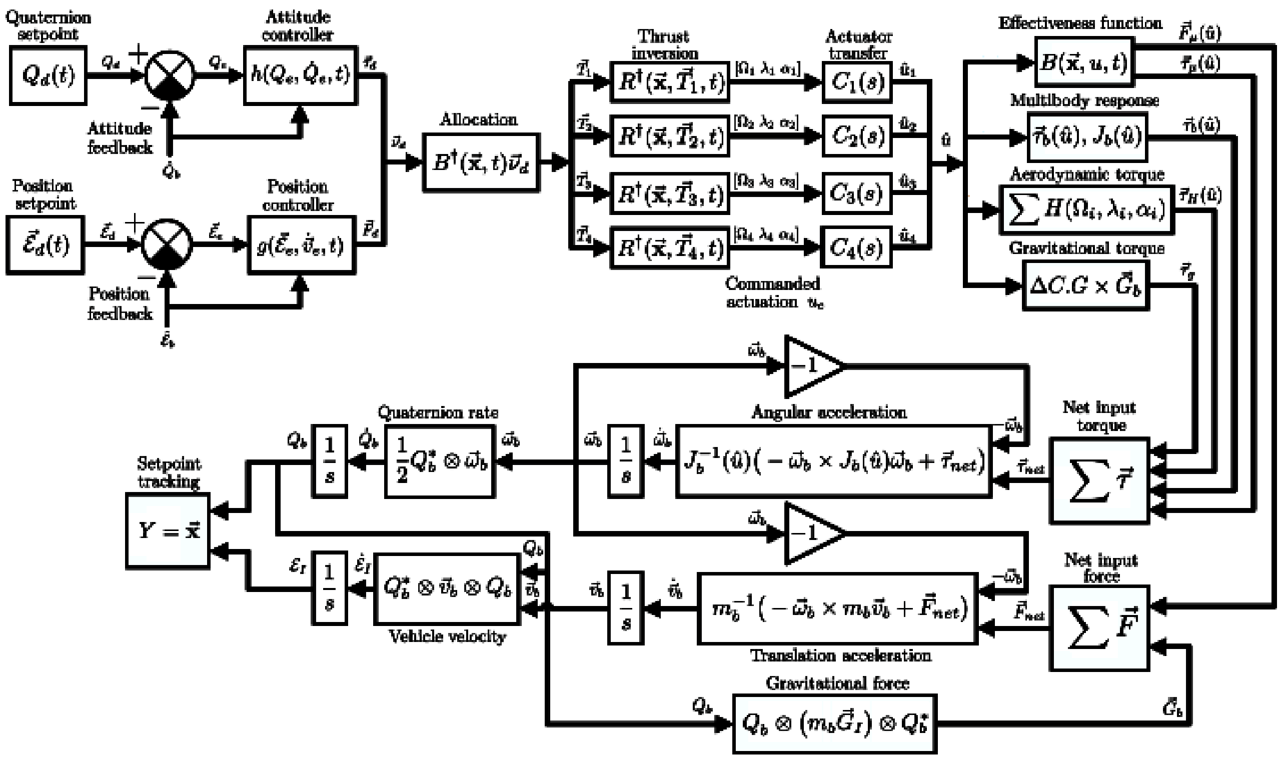


Figure 6.1: Simulation loop

An abstracted simulation loop is illustrated in Fig:6.1; incorporating both attitude and position control loops together with the additive nonlinearities. Certain feedback elements were omitted to retain clarity in the diagram, both Coriolis and Gyroscopic nonlinear cross-products were included to indicate the inherent coupling between attitude and position. Not shown is some form of state-estimation or discretization between the state-tracking output $y = \vec{\mathbf{x}} = [\vec{\mathcal{E}}_I \; Q_b]^T$ and the feedback state $\hat{\mathbf{x}}$ used for setpoint tracking. Discretized effects of state-estimators are discussed later in Sec:6.9.

Initial conditions for each state integrator, both position $\vec{\mathcal{E}}_I(0)$ and attitude $Q_b(0)$ origins, for their velocities $\dot{\mathcal{E}}_b$ and $\dot{Q}_b$ and accelerations $\dot{\vec{v}}_b$ and $\dot{\vec{\omega}}_b$ are not illustrated but implied. Obviously starting conditions are important for each trajectory's simulation but are specifically defined for each simulation in question. Actuator transfer functions from Sec:2.4.1 are combined into a bundled $C_i(S)$ block, accounting for transfer functions and the saturation limits of each motor module. Each bundled input $u_{1\to4}$ is similarly the projected actuator matrix:

$$u_i \triangle u \cdot i = \begin{bmatrix}\Omega_i & \lambda_i & \alpha_i\end{bmatrix} \quad \text{for } i \in [1:4] \tag{6.1}$$

The resultant thrust vector $\vec{T}_i$ produced by each motor module has a combined MIMO transfer function. Lastly, setpoints for both attitude and position states are either stepped or produced from a simple orbital trajectory. The former is used for controller optimization whilst the latter is used controller setpoint tracking performance evaluation. To discuss the question of non-zero setpoint tracking an orbital trajectory is generated; with attitude and position setpoints illustrated in Fig:6.2.

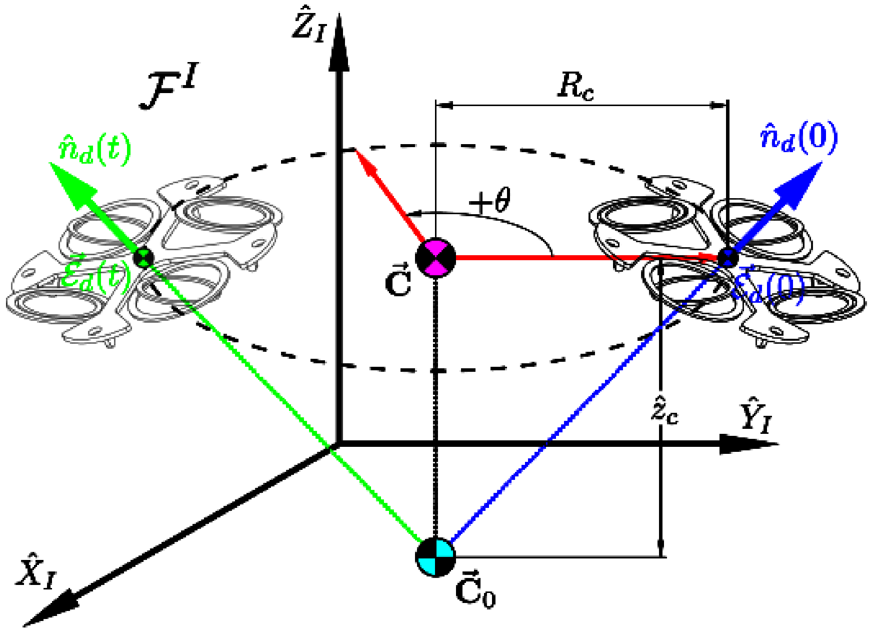


Figure 6.2: Orbital trajectory

The trajectory generates only first order attitude and position setpoints; a trajectory is entirely independent from actuator values for the aircraft's configuration. For a central point in the inertial frame $\vec{C}_0 \in \mathcal{F}^I$, the trajectory orbits at an angular rate of $\dot{\theta}$ Hz around that center. The orbit is at a height of $\hat{z}_c$ m and at a radius R_c m from the center $\vec{C}$. The position setpoint then follows:

$$\vec{\mathcal{E}}_d(t) = \begin{bmatrix} C_{0x} + R_c \cos\theta(t) \\ C_{0y} + R_c \sin\theta(t) \\ \hat{z}_c \end{bmatrix} \qquad \in \mathcal{F}^I \tag{6.2a}$$

The time varying trajectory's attitude setpoint is aligned with a normal vector $\hat{n}_d(t)$, pointing away from the center point $\vec{C}_0$:

$$\hat{n}_d(t) \triangleq \frac{\vec{\mathcal{E}}_d(t) - \vec{C}_0}{\sqrt{\hat{z}_c^2 + R_c^2}} \tag{6.2b}$$

The quaternion setpoint is then constructed from $\hat{n}_d$ such that:

$$Q_d(t) = \begin{bmatrix}\sin\frac{\theta(t)}{2} & \cos\frac{\theta(t)}{2}\hat{n}_d(t)\end{bmatrix}^T \tag{6.2c}$$

Neither first order nor higher derivative setpoints are applied for the trajectory in Fig:6.2. Both position and attitude rates are respectively $\dot{\vec{\mathcal{E}}}_d(t) = \vec{0}$ and $\dot{Q}_d(t) = \vec{0}$ throughout the entire trajectory, as per Eq:4.85a and Eq:4.27c.

6.2 Controller Tuning

Each controller's derivation and stability, shown previously in Ch:4, demonstrated only a control law's setpoint tracking ability; providing no further insight into the controller coefficient design. Lyapunov stability theorem, in the context of Sec:4.6-4.7, evaluates a particular trajectory's stability over $t \to \infty$ but nothing more. Often at the coefficient selection stage a *monte carlo* approach is applied; in most cases choosing coefficients seemingly at random and haphazardly, without any obvious forethought or design...

6.2.1 Partical Swarm Based Optimization

Particle swarm based optimization (*PSO*) has been shown in both [144] and [80], amongst others, to be an effective controller coefficient design tool. The algorithm treats each variable to be optimized as a *particle* which exists within some defined search space. The collection or *swarm* of particles explores the search space directed by both the swarm's previous performance as well as the relative performance between each particle. In [138] the statistical nature of the swarm's trajectory is discussed, however such investigations are well beyond the scope of this work. In general the PSO algorithm applies a *gradient free* based search of solutions for a given optimization problem. The lack of a specified gradient is an important distinction which differentiates PSO from other algorithms. Often a predefined gradient function is required to direct the optimization search; MatLab's own Fmincon [91] or Interior-Point optimizer [64] algorithms for example. Interval gradient calculations can be computationally exhaustive and reduce the rate of execution for the entire process. An optimizer's performance is directly proportional to the number of complete iterations it executes, if an iteration has a high degree of complexity (read *stiffness*) its solution time is then adversely affected. The PSO algorithm is defined as follows; if there exists a set $\vec{x}$ of k variables, $\vec{x} \in \mathbb{R}^{k\times 1}$ to be optimized. The swarm of particles, starting at position $\vec{x}_0$, has an n^{th} interval $\vec{x}_n$ which progresses through the search space as per a velocity function:

$$\vec{x}_{n+1} = \vec{v}_n + x_n \tag{6.3a}$$

$$\vec{v}_{n+1} \triangleq w * \vec{v}_n + c_1 * r_1\big(\vec{P}_{best} - \vec{x}_n\big) + c_2 * r_2\big(\vec{G}_{best} - \vec{x}_n\big) \tag{6.3b}$$

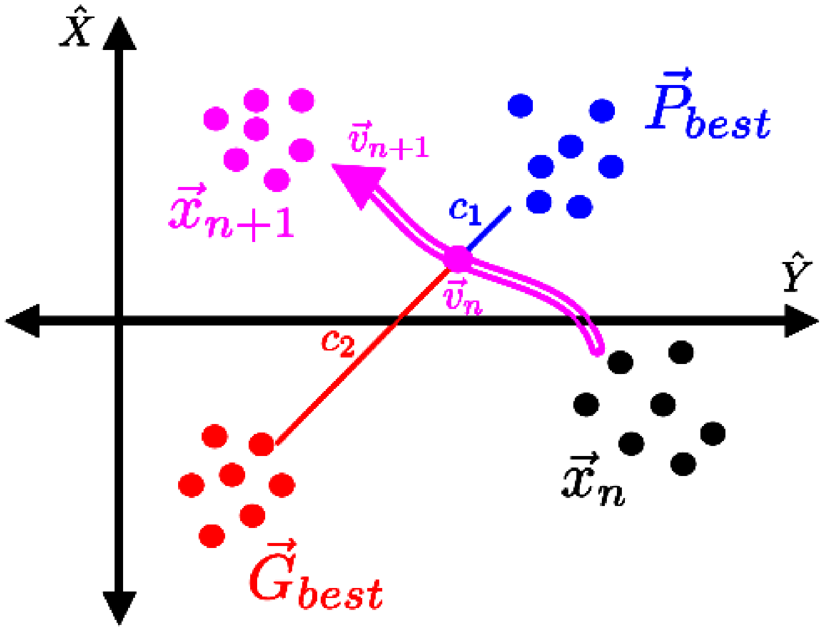


Figure 6.3: Swarm trajectory's velocity direction

Each $*$ operator in Eq:6.3b applies an element-by-element matrix coefficient multiplication. Both $\vec{P}_{best}$ and $\vec{G}_{best}$ are previous swarm positions where local and global optima were respectively achieved. Performance of the swarm's current interval is evaluated as per some cost function, responding to a system's dynamics; expanded on next. Finally r_1 and r_2 are random seeded $\mathbb{R}^{1\times k}$ exploratory matrices which progress the search direction, biased by the two weighting coefficients c_1 and c_2. The search is prejudiced toward local optima by c_1 whilst c_2 directs the swarm toward global optima. Fig:6.3 illustrates how positions of both local and global optima influence subsequent velocities.

The swarm's interval performance is evaluated by the response of a dynamic system to the swarm's position; typically an error deviation away from some desired state. Here, the simulation described in Fig:6.1, is parsed a swarm of controller coefficients as an argument and the plant's setpoint response is simulated over a series of step tests. Particulars with regards to attitude controller optimization is discussed in Sec:6.3, thereafter position controller optimization is detailed in Sec:6.4. The objective is for zero-error setpoint tracking so each swarm's coefficient performance metric calculates an integral-time-absolute-error ($ITAE$) cost function, [88].

$$\vec{\zeta} \triangleq \int_{t_0}^{t_\infty} t|\vec{e}(t)|.dt \tag{6.4}$$

With an error $\vec{e}(t)$ deviating from the plant's given setpoint. The ITAE integral $\vec{\zeta}$ is calculated over the entire simulation time, or an effective t_∞. The time multiplier ensures setpoint error *and* settling time optimality; punishing overshoot and under-damped or oscillatory like behaviour. In general a PSO algorithm progressing following the flow diagram in Fig:6.4. Seeing that each controller was empirically proven to be stable, independent of it's trajectory; the controller will settle irrespective of the proposed interval coefficient values. A consequence of this is that starting conditions $\vec{x}_0$ for the swarm have no bearing on the progression of the optimization. A rounded set of unity coefficients were selected as a starting point for each controller's optimization...

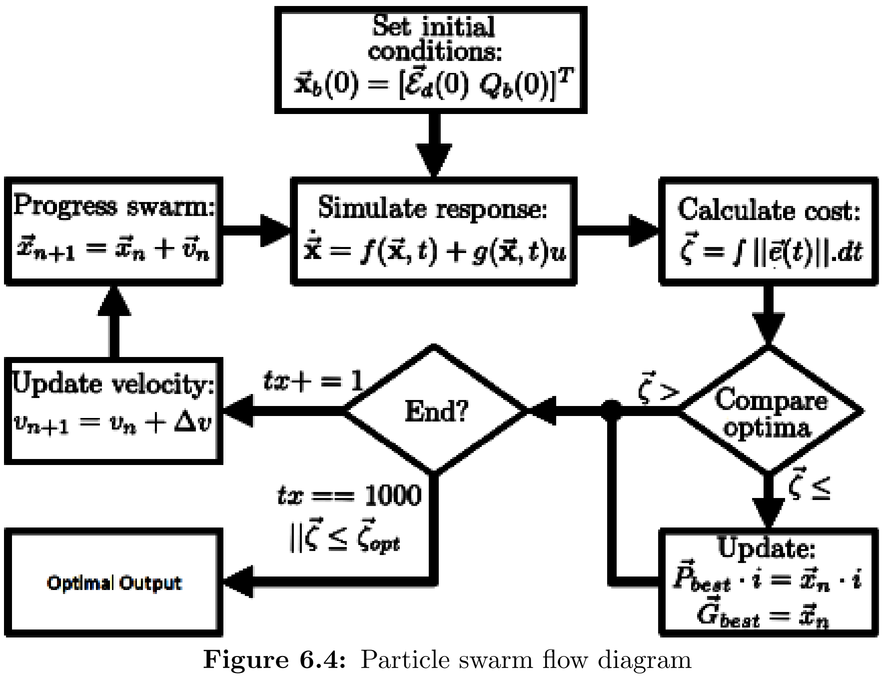


Figure 6.4: Particle swarm flow diagram

Termination conditions for the iterative optimization loop either limit the number of iteration cycles performed or break from the process once a result is regarded as sufficiently close to optimal. Each optimization cycle was iterated for $tx = 1000$ times, testing and evaluating one thousand different swarm values for a series of stepped setpoints. As the optimizer progressed through iterations it adapted its bias from a global to local optima, refining the way in which it searched for potential controller coefficients.

$$\vec{v}_{n+1} = \vec{v}_n + \frac{tx}{1000} * r_1(\vec{P}_{best} - \vec{x}_n) + \frac{1000 - tx}{1000} * r_2(\vec{G}_{best} - \vec{x}_n) \quad \text{for} \quad tx \in [1:1000] \tag{6.5}$$

This gave each controller an equal likelihood of reaching optimality. Obviously controllers which improve their performance between each iteration at a faster rate performed better. Moreover each swarm's progression was constrained such that it never violated the Lyapunov stability conditions of its respective control law...

6.3 Attitude Controllers

Attitude controllers derived in Sec:4.6 were optimized first, owing to their independence from the position loop. A constant hovering force condition was simply applied to the virtual plant input $\vec{F}_d$ for each test. Pseudo inverse allocation, Sec:5.3.1, was applied to the control loop when testing each attitude controller. To evaluate an individual swarm's performance a number of step tests were performed. Each attitude setpoint was first defined in the Euler angle parametrization, being conceptually easier to visualize. Thereafter the attitude setpoints were converted to a desired quaternion attitude and applied to the simulation.

$$\vec{\eta}_d(t) \triangleq \begin{bmatrix}\phi_d(t) & \theta_d(t) & \psi_d(t)\end{bmatrix}^T \underset{Q}{\Longleftrightarrow} Q_d(t) \tag{6.6}$$

Each of the three Euler angles were stepped in the range $[-90° : +90°]$ at intervals of 30°. This resulted in a test of three hundred and forty three possible attitude setpoints; making a workspace sphere as illustrated in Fig:6.5. Each attitude step was granted $t = 15$ s to reach its settling point, with an initial attitude position always set to the origin $Q_b(t_0) = [1\ \vec{0}\,]$, with a *positive* quaternion scalar. The performance for each attitude step test was evaluated by an ITAE integral for the quaternion error vector and angular velocity error:

$$\vec{\zeta}_Q = C_Q \int_{t=0}^{15} t|\vec{q}_e(t)|.dt + C_\omega \int_{t=0}^{15} t|\vec{\omega}_e(t)|.dt \qquad \in \mathbb{R}^3 \tag{6.7}$$

Weighting coefficients C_Q and Q_ω balance priority of either quaternion or angular velocity tracking, however, tracking both were equally important and so those weights were $C_Q = Q_\omega = 1$. The cost integral in Eq:6.7 was averaged over all three hundred and forty three possible attitude steps to determine the overall performance of a proposed swarm of controller coefficients.

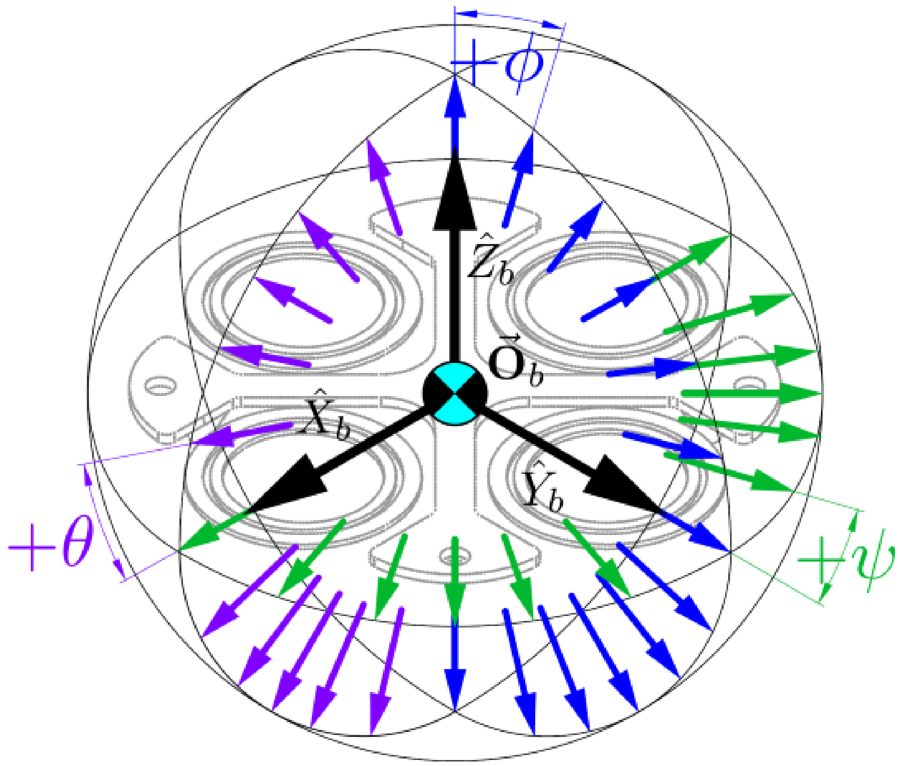


Figure 6.5: Attitude setpoint working space

The integral in Eq:6.7 produces a vector $\in \mathbb{R}^3$ result. Each coefficient in a particular controller contributes towards a local error in one of the $\hat{X}, \hat{Y}, \hat{Z}$ components, or in certain cases a pair of axial components. Each controller's local errors and the coefficients which affect them are discussed subsequently. A global error for the performance of each controller is simply the magnitude $\left\|\vec{\zeta}_Q\right\|$. The same global error is applicable to all controllers...

To compare the relative performance and effectiveness of each optimized control structure a single attitude step was investigated. That attitude change was chosen to be a sizeable step in all three Euler angles:

$$\vec{\eta}_d = \begin{bmatrix}\phi_d\\ \theta_d\\ \psi_d\end{bmatrix} = \begin{bmatrix}-142°\\ 167°\\ -45°\end{bmatrix} \underset{Q}{\Longleftrightarrow} \begin{bmatrix}-0.3254 & 0.2226 & -0.2579 & 0.8821\end{bmatrix}^T \tag{6.8}$$

Then each controller's settling time to 95% of its final value, t_{95}, and it's relative angular velocity (the setpoint for which is $\vec{\omega}_d = \vec{0}$) for such a step is evaluated. Settling time, overshoot and setpoint error are all factors to consider when discussing a controller's efficacy. Lastly, commanded (virtual) and applied input torque to the actuator set are discussed too. A feasible controller should not induce torque saturation or unachievable input rate changes.

6.3.1 PD

The first controller evaluated, the Proportional-Derivative structure, is investigated under three different circumstances. Before discussing each of those different scenarious, it is worth recalling that control structure from Sec:4.6.2. Control torque is designed by two coefficient matrices K_p and K_d:

$$\vec{\tau}_{PD} = \underbrace{K_p\vec{q}_e + K_d\vec{\omega}_e}_{Independent} + \underbrace{\hat{\omega}_b \times J_b(\hat{u})\hat{\omega}_b + \vec{\tau}_b(\hat{u}) - \vec{\tau}_g - \vec{\tau}_H}_{Compensation} \qquad \in \mathcal{F}^b \tag{6.9}$$

The first two tests regard both coefficient matrices as purely diagonal, with no skew elements; evaluating the effect inclusion of plant dependent compensation has on the controller's performance. Finally a plant dependent compensating PD controller is tested *with* symmetrical coefficient matrices. The diagonal coefficient matrices are defined and numbered as follows:

$$K_p \triangleq \begin{bmatrix}K_p(1) & 0 & 0\\ 0 & K_p(2) & 0\\ 0 & 0 & K_p(3)\end{bmatrix} \quad \text{and} \quad K_d \triangleq \begin{bmatrix}K_d(1) & 0 & 0\\ 0 & K_d(2) & 0\\ 0 & 0 & K_d(3)\end{bmatrix} \tag{6.10}$$

Then selection of each local and global position for the PSO algorithm needs to be defined. The proportional coefficient K_p acts on $\vec{q}_e$ whilst the derivative coefficient K_d acts on $\vec{\omega}_e$. Naturally the local and global optimal position are then selected and updated:

$$\vec{P}_{Best} \equiv \begin{bmatrix}K_p(1) \Rightarrow \min \vec{q}_e(1)\\ K_p(2) \Rightarrow \min \vec{q}_e(2)\\ K_p(3) \Rightarrow \min \vec{q}_e(3)\\ K_d(1) \Rightarrow \min \vec{\omega}_e(1)\\ K_d(2) \Rightarrow \min \vec{\omega}_e(2)\\ K_d(3) \Rightarrow \min \vec{\omega}_e(3)\end{bmatrix} \quad \text{and} \quad \vec{G}_{Best} \equiv \begin{bmatrix}K_p(1) \Rightarrow \min \vec{\zeta}_{PD}(1)\\ K_p(2) \Rightarrow \min \vec{\zeta}_{PD}(2)\\ K_p(3) \Rightarrow \min \vec{\zeta}_{PD}(3)\\ K_d(1) \Rightarrow \min \vec{\zeta}_{PD}(1)\\ K_d(2) \Rightarrow \min \vec{\zeta}_{PD}(2)\\ K_d(3) \Rightarrow \min \vec{\zeta}_{PD}(3)\end{bmatrix} \tag{6.11}$$

Where each local or global best coefficient position is updated if the minimum (best) result is improved on. For the symmetrical coefficient case, each off-diagonal element acts on two components of the error states. Those controller coefficients are then numbered:

$$K_p \triangleq \begin{bmatrix}K_p(1) & K_p(4) & K_p(5)\\ K_p(4) & K_p(2) & K_p(6)\\ K_p(5) & K_p(6) & K_p(3)\end{bmatrix} \quad \text{and} \quad K_d \triangleq \begin{bmatrix}K_d(1) & K_d(4) & K_d(5)\\ K_d(4) & K_d(2) & K_d(6)\\ K_d(5) & K_d(6) & K_d(3)\end{bmatrix} \tag{6.12}$$

The local and global coefficient positions are then found, with skew elements requiring combined improvement on *two* error components such that:

$$\vec{P}_{Best} \equiv \begin{bmatrix} K_p(1) \Rightarrow \min \vec{q}_e(1) & K_p(4) \Rightarrow \min \vec{q}_e(1) \&\& \vec{q}_e(2) \\ K_p(2) \Rightarrow \min \vec{q}_e(2) & K_p(5) \Rightarrow \min \vec{q}_e(1) \&\& \vec{q}_e(3) \\ K_p(3) \Rightarrow \min \vec{q}_e(3) & K_p(6) \Rightarrow \min \vec{q}_e(2) \&\& \vec{q}_e(3) \\ K_d(1) \Rightarrow \min \vec{\omega}_e(1) & K_d(4) \Rightarrow \min \vec{\omega}_e(1) \&\& \vec{\omega}_e(2) \\ K_d(2) \Rightarrow \min \vec{\omega}_e(2) & K_d(5) \Rightarrow \min \vec{\omega}_e(1) \&\& \vec{\omega}_e(3) \\ K_d(3) \Rightarrow \min \vec{\omega}_e(3) & K_d(6) \Rightarrow \min \vec{\omega}_e(2) \&\& \vec{\omega}_e(3) \end{bmatrix} \tag{6.13a}$$

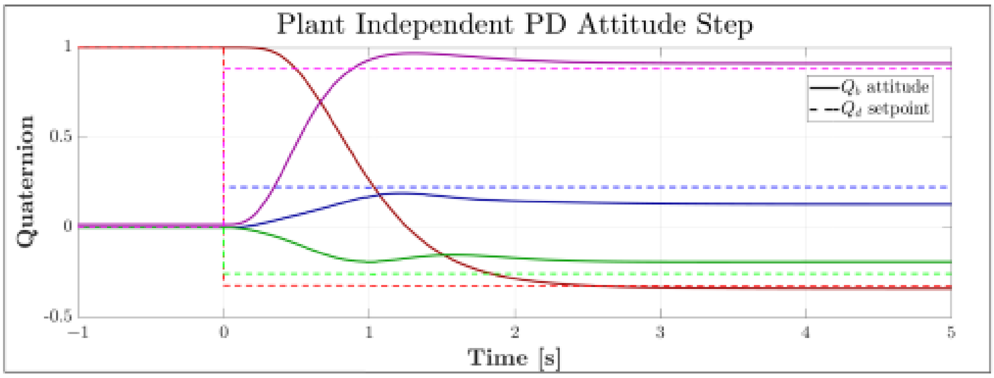
$$\vec{G}_{Best} \equiv \begin{bmatrix} K_p(1) \Rightarrow \min \vec{\zeta}_{PD}(1) & K_p(4) \Rightarrow \min \vec{\zeta}_{PD}(1) \&\& \vec{\zeta}_{PD}(2) \\ K_p(2) \Rightarrow \min \vec{\zeta}_{PD}(2) & K_p(5) \Rightarrow \min \vec{\zeta}_{PD}(1) \&\& \vec{\zeta}_{PD}(3) \\ K_p(3) \Rightarrow \min \vec{\zeta}_{PD}(3) & K_p(6) \Rightarrow \min \vec{\zeta}_{PD}(2) \&\& \vec{\zeta}_{PD}(3) \\ K_d(1) \Rightarrow \min \vec{\zeta}_{PD}(1) & K_d(4) \Rightarrow \min \vec{\zeta}_{PD}(1) \&\& \vec{\zeta}_{PD}(2) \\ K_d(2) \Rightarrow \min \vec{\zeta}_{PD}(2) & K_d(5) \Rightarrow \min \vec{\zeta}_{PD}(1) \&\& \vec{\zeta}_{PD}(3) \\ K_d(3) \Rightarrow \min \vec{\zeta}_{PD}(3) & K_d(6) \Rightarrow \min \vec{\zeta}_{PD}(2) \&\& \vec{\zeta}_{PD}(3) \end{bmatrix} \tag{6.13b}$$

Independent Performance

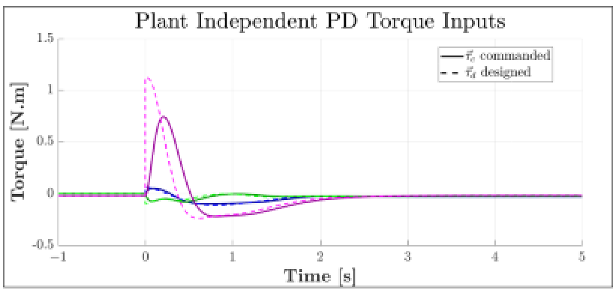
For the independent controller case, the same diagonal coefficients are used as those for the plant dependent case. The *attitude* compensation terms in Eq:6.9 are neglected to produce a plant independent controller. Optimizing the diagonal only PD controller produced the following coefficients:

$$K_p = \begin{bmatrix} 3.5679 & 0 & 0 \\ 0 & 5.2698 & 0 \\ 0 & 0 & 6.0695 \end{bmatrix} \quad \text{and} \quad K_d = \begin{bmatrix} 9.0150 & 0 & 0 \\ 0 & 11.4848 & 0 \\ 0 & 0 & 20.1827 \end{bmatrix} \tag{6.14}$$

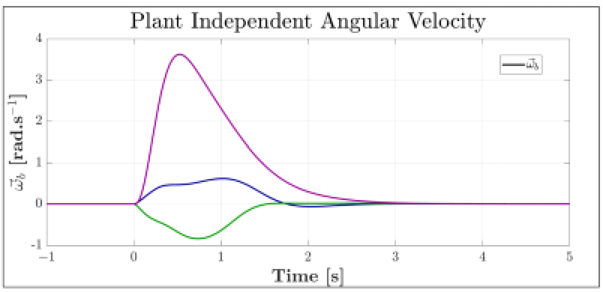
Fig:6.6a plots the quaternion response to an attitude step, described in Eq:6.8. The uncompensated plant never settles to its setpoint; constant errors manifests due to the uncompensated gravitational and aerodynamic torques. The plant does, however, stabilize to steady state in $t = 3.35$ s. Fig:6.6b compares the controller designed and physically actuated input torques, $\vec{\tau}_d$ and $\vec{\tau}_c$ respectively. Actuator transfer functions produce a lagging response to those input changes. Finally Fig:6.6c plots the body's angular velocity $\vec{\omega}_b \in \mathcal{F}^b$ which changes as an attitude step is applied.



(a) Quaternion attitude step



(b) Plant input torques

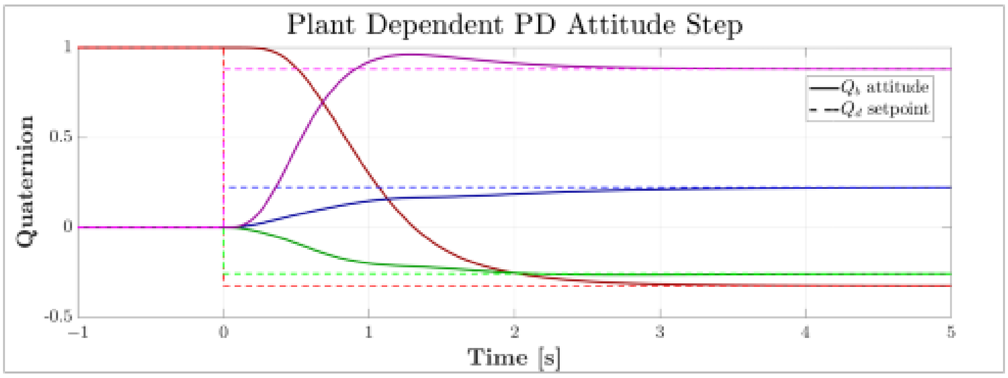


(c) Angular velocity

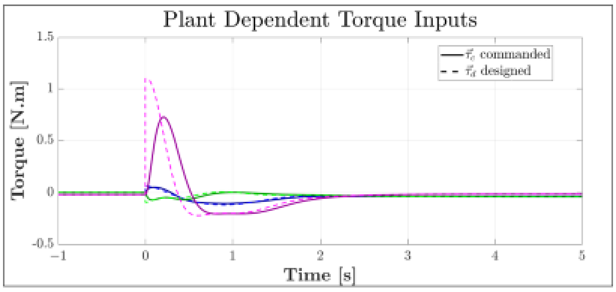
Figure 6.6: Independent diagonal PD

Dependent Performance

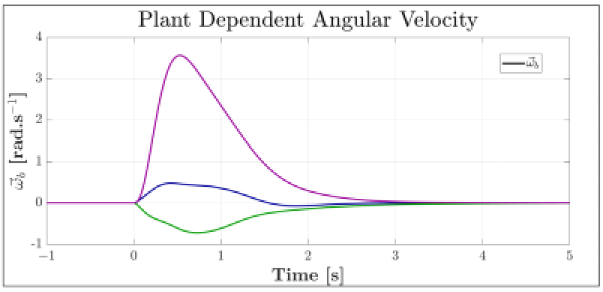
The inclusion of a plant independent PD controller is purely for the sake of contrition; showing that plant dependency is needed to account for steady state trajectory errors (Fig:6.14a). Those same controller coefficients from Eq:6.14 are now used to test the controller dependent case; wherein the controller compensates for plant dynamics in Eq:6.9 with a feedback terms. The standard quaternion attitude step is shown in Fig:6.7a for the plant dependent case. The attitude settles in $t_{95} = 3.0764$ s with a dynamic response much the same as that of the independent case, Fig:6.6a. However the dependent controller removes steady state tracking errors. The torque input is well still within the feasible range; not saturating any actuators. The difference is that, at steady state, the plant's torque input is marginally offset from zero and compensates for the previous steady state error.



(a) Quaternion attitude step



(b) Plant input torques



(c) Angular velocity

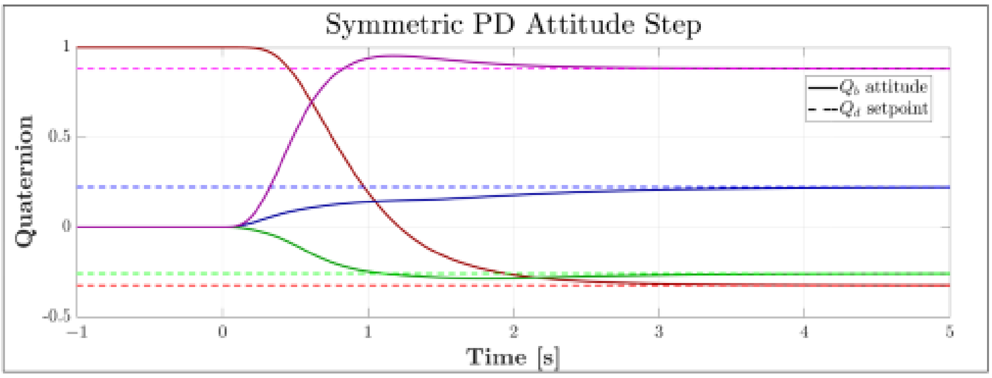
Figure 6.7: Dependent diagonal PD

Symmetric Controller Performance

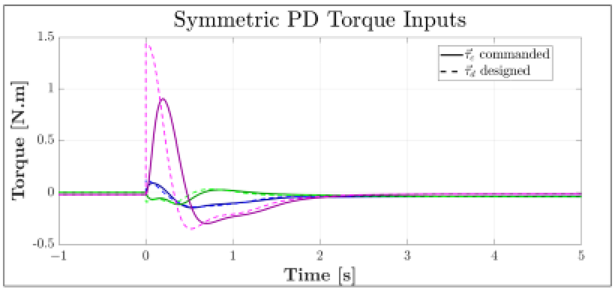
The last PD structured attitude controller considers both coefficient matrices with non-zero off-diagonal skew elements. Eq:6.12 shows the structure of both symmetric matrices and how their optimization positions are chosen. The optimized coefficients are then found to be:

$$K_p = \begin{bmatrix} 5.9157 & 0.4165 & 0.4714 \\ 0.4165 & 7.3141 & 0.4945 \\ 0.4714 & 0.4945 & 7.3135 \end{bmatrix} \quad \text{and} \quad K_d = \begin{bmatrix} 17.4318 & 0.45311 & 0.15258 \\ 0.45311 & 15.3569 & 0.57719 \\ 0.15258 & 0.57719 & 26.3436 \end{bmatrix} \tag{6.15}$$

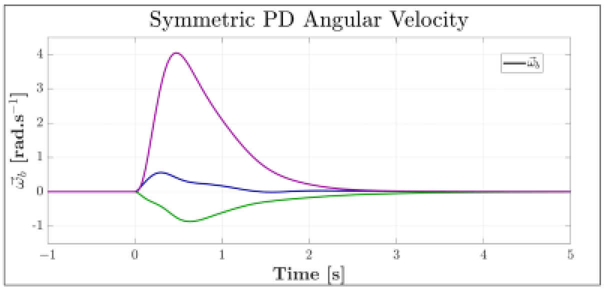
The first notable difference of the symmetric controller is that its effective gain, applied by Eq:6.15 to the error in Eq:6.9, is significantly larger than the gain in Eq:6.14. The off-diagonal elements are tending towards linearizing the induced gyroscopic cross-coupled torque terms in feedback, Eq:3.129d. The step response of the optimized symmetric PD controller is shown in Fig:6.8a. The increased effective gain in Eq:6.15 results in larger overshoot and a slower settling time, $t_{95} = 3.2993$ s. Neither greater commanded torque, Fig:6.8b, nor an increased angular velocity spike, Fig:6.8c are altogether unexpected consequences of a more aggressive control law. The increased number of coefficients to be tuned simply meant that optimization to produce Eq:6.15 was perhaps not as effective at interval reduction of step errors than the diagonal Eq:6.14.



(a) Quaternion attitude step



(b) Plant input torques



(c) Angular velocity

Figure 6.8: Dependent symmetric PD

6.3.2 Auxilliary Plant Controller

The first of two the exponentially stabilizing controllers is the Auxiliary Plant controller from Sec:4.6.2. Recalling that controller structure from Eq:4.41, with Auxiliary plants defined in Eq:4.42 and Eq:4.43:

$$\vec{\tau}_{XPD} = \underbrace{\Gamma_2\widetilde{\Omega} + \Gamma_3\vec{q}_e - J_b(\hat{u})\dot{\widetilde{\Omega}}}_{Independent} + \underbrace{\hat{\omega}_b \times J_b(\hat{u})\hat{\omega}_b + \vec{\tau}_b(\hat{u}) - \vec{\tau}_g - \vec{\tau}_H}_{Compensation} \quad \in \mathcal{F}^b \tag{6.16}$$

In Eq:6.16 both coefficients Γ_2 and Γ_3 are 3×3 diagonal matrices; whereas Γ_1 is a symmetrical 3×3 gain matrix. Those coefficients are then structured as follows:

$$\Gamma_1 \triangleq \begin{bmatrix} \Gamma_1(1) & \Gamma_1(4) & \Gamma_1(5) \\ \Gamma_1(4) & \Gamma_1(2) & \Gamma_1(6) \\ \Gamma_1(5) & \Gamma_1(6) & \Gamma_1(3) \end{bmatrix}, \quad \Gamma_2 \triangleq \begin{bmatrix} \Gamma_2(1) & 0 & 0 \\ 0 & \Gamma_2(2) & 0 \\ 0 & 0 & \Gamma_2(3) \end{bmatrix}$$

$$\text{and} \quad \Gamma_3 \triangleq \begin{bmatrix} \Gamma_3(1) & 0 & 0 \\ 0 & \Gamma_3(2) & 0 \\ 0 & 0 & \Gamma_3(3) \end{bmatrix} \tag{6.17}$$

Global and local optimum coefficient swarm positions are found from the error state components on which the particular coefficients act. The first gain matrix Γ_1 acts on both $\vec{q}_e$ and $\vec{\omega}_e$ so its local errors are best performance positions for the global error. The remaining two gain matrices Γ_2 and Γ_3 act on $\vec{q}_e$ and $\vec{\omega}_e$ respectively. The local best swarm positions are then found:

$$\vec{P}_{Best} \equiv \begin{bmatrix} \Gamma_1(1) \Rightarrow \min \vec{\zeta}_{XPD}(1) & \Gamma_1(4) \Rightarrow \min \vec{\zeta}_{XPD}(1)\&\&\vec{\zeta}_{XPD}(2) \\ \Gamma_1(2) \Rightarrow \min \vec{\zeta}_{PXD}(2) & \Gamma_1(5) \Rightarrow \min \vec{\zeta}_{XPD}(1)\&\&\vec{\zeta}_{XPD}(3) \\ \Gamma_1(3) \Rightarrow \min \vec{\zeta}_{XPD}(3) & \Gamma_1(6) \Rightarrow \min \vec{\zeta}_{XPD}(2)\&\&\vec{\zeta}_{XPD}(3) \\ \Gamma_2(1) \Rightarrow \min \vec{q}_e(1) & \Gamma_3(1) \Rightarrow \min \vec{\omega}_e(1) \\ \Gamma_2(2) \Rightarrow \min \vec{q}_e(2) & \Gamma_3(2) \Rightarrow \min \vec{\omega}_e(2) \\ \Gamma_2(3) \Rightarrow \min \vec{q}_e(3) & \Gamma_3(3) \Rightarrow \min \vec{\omega}_e(3) \end{bmatrix} \tag{6.18a}$$

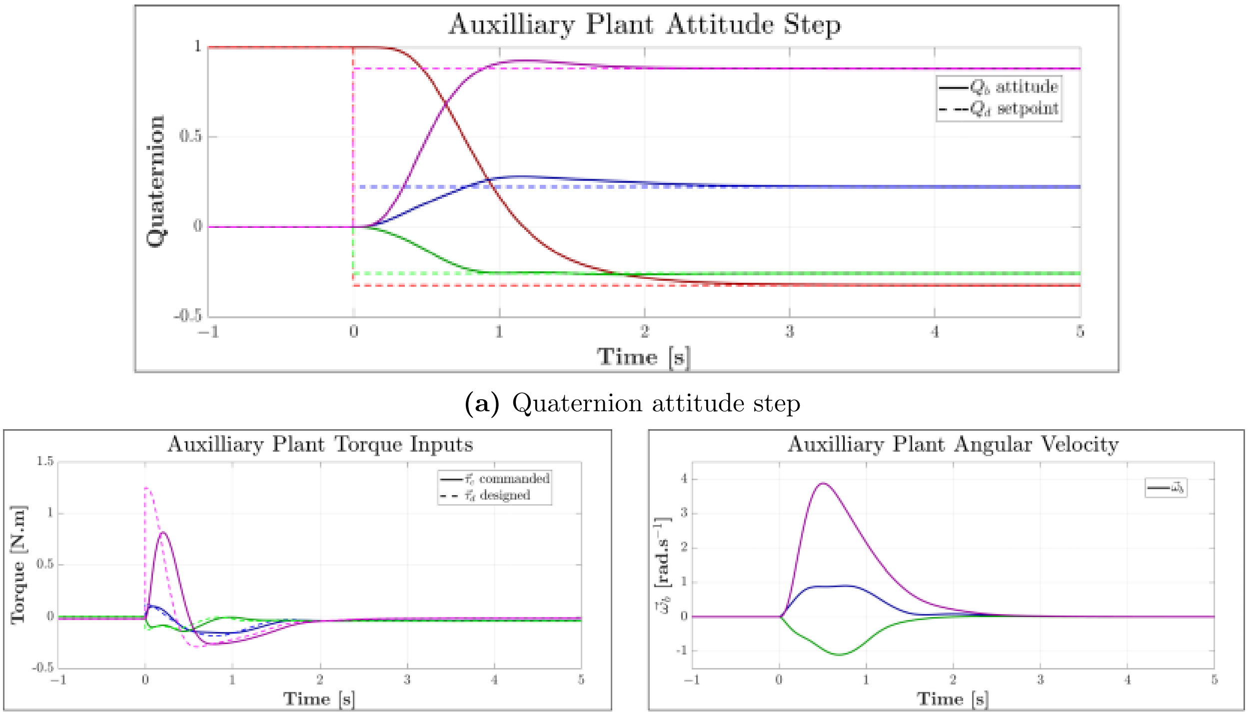
The position of the globally best performing swarm is found similarly:

$$\vec{G}_{Best} \equiv \begin{bmatrix} \Gamma_1(1) \Rightarrow \min \vec{\zeta}_{XPD}(1) & \Gamma_1(4) \Rightarrow \min \vec{\zeta}_{XPD}(1) \& \& \vec{\zeta}_{XPD}(2) \\ \Gamma_1(2) \Rightarrow \min \vec{\zeta}_{PXD}(2) & \Gamma_1(5) \Rightarrow \min \vec{\zeta}_{XPD}(1) \& \& \vec{\zeta}_{XPD}(3) \\ \Gamma_1(3) \Rightarrow \min \vec{\zeta}_{XPD}(3) & \Gamma_1(6) \Rightarrow \min \vec{\zeta}_{XPD}(2) \& \& \vec{\zeta}_{XPD}(3) \\ \Gamma_2(1) \Rightarrow \min \vec{\zeta}_{XPD}(1) & \Gamma_3(1) \Rightarrow \min \vec{\zeta}_{XPD}(1) \\ \Gamma_2(2) \Rightarrow \min \vec{\zeta}_{XPD}(2) & \Gamma_3(2) \Rightarrow \min \vec{\zeta}_{XPD}(2) \\ \Gamma_2(3) \Rightarrow \min \vec{\zeta}_{XPD}(3) & \Gamma_3(3) \Rightarrow \min \vec{\zeta}_{XPD}(3) \end{bmatrix} \tag{6.18b}$$

The optimized control coefficients, after $tx = 1000$ iterations, are as follows:

$$\Gamma_1 = \begin{bmatrix} 3.5924 & -0.2457 & -0.0277 \\ -0.2457 & 3.0666 & -0.0602 \\ -0.0277 & -0.0602 & 3.3809 \end{bmatrix}, \quad \Gamma_2 = \begin{bmatrix} 4.6943 & 0 & 0 \\ 0 & 4.1642 & 0 \\ 0 & 0 & 6.4109 \end{bmatrix}$$
$$\text{and} \quad \Gamma_3 = \begin{bmatrix} 1.1007 & 0 & 0 \\ 0 & 1.3369 & 0 \\ 0 & 0 & 1.1331 \end{bmatrix} \tag{6.19}$$

Besides from the stronger exponential stability, another distinctive feature of Auxilliary Controller (Eq:6.17) is the significant added complexity or stiffness in the control structure. Every simulation iteration in the optimizer took notably longer to complete than the simpler PD controller; typically in the order of 70-80% longer simulation times per step test. The quaternion attitude step response is shown in Fig:6.9a, settling in $t_{95} = 2.3688$ s which is significantly faster than previous tested controllers. That improved response time does, however, come at the cost of larger input torques, shown in Fig:6.9b. Moreover the angular velocity step $\vec{\omega}_b$ in Fig:6.9c is significantly larger than that of previous controllers, but with a smoother derivative applied to it...



(a) Quaternion attitude step

(b) Plant input torques

(c) Angular velocity

Figure 6.9: Auxilliary plant controller

In spite of the significantly improved settling time the Auxilliary plant controller achieves (23% faster), the commanded and applied input torques are not as aggressive as those applied by the higher gain symmetrical PD controller (Fig:6.8b). This concisely demonstrates the superiority of an exponentially stabilizing control law over a typical asymptotically stable alternative...

6.3.3 Ideal and Adaptive Backstepping Controllers

The second exponentially stabilizing controller and final attitude controller tested is the Ideal Backstepping Controller. Both Ideal and Adaptive backstepping controllers use the same error coefficients; the difference in structure between the two is the addition of an adaptive disturbance observer. That observer estimates compensation terms in the Adaptive Backstepping controller, improving robust stability. Reiterating the IBC structure from Eq:4.66:

$$= \underbrace{J_b(\hat{u})\Big((\Gamma_1\Gamma_2 + 1)\vec{q}_e - \Gamma_2\hat{\omega}_b + \Gamma_1\dot{\vec{q}}_e\Big)}_{\text{Ideal backstepping}} + \underbrace{\hat{\omega}_b \times J_b(\hat{u})\hat{\omega}_b + \vec{\tau}_b(\hat{u}) - \vec{\tau}_g - \vec{\tau}_H}_{\text{Compenstation}} \quad \in \mathcal{F}^b \tag{6.20}$$

Wherein the gain matrices Γ_1 and Γ_2 are both positive symmetrical 3×3 coefficient matrices:

$$\Gamma_1 \triangleq \begin{bmatrix} \Gamma_1(1) & \Gamma_1(4) & \Gamma_1(5) \\ \Gamma_1(4) & \Gamma_1(2) & \Gamma_1(6) \\ \Gamma_1(5) & \Gamma_1(6) & \Gamma_1(3) \end{bmatrix} \quad \text{and} \quad \Gamma_2 \triangleq \begin{bmatrix} \Gamma_2(1) & \Gamma_2(4) & \Gamma_2(5) \\ \Gamma_2(4) & \Gamma_2(2) & \Gamma_2(6) \\ \Gamma_2(5) & \Gamma_2(6) & \Gamma_2(3) \end{bmatrix} \tag{6.21}$$

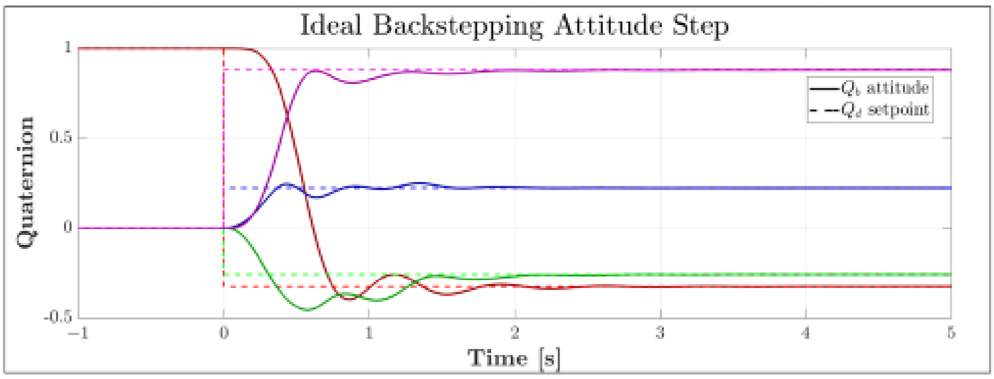
However, both coefficient matrices act on the two error vectors $\vec{q}_e$ and $\vec{\omega}_e$. Trying to differentiate the local and global coefficient optimum results then becomes difficult. Making both the local and global positions equivalent reduces the directed swarm search to a randomized *monte carlo* trial and error method. To avoid this Γ_1 is prioritized to control the quaternion vector error $\vec{q}_e$, similarly Γ_2 was dedicated to controlling the angular velocity error $\vec{\omega}_e$. It then follows that local and global best positions, $\vec{P}_{best}$ and $\vec{G}_{best}$ respectively, are found in the same way as the symmetrical PD controller, Eq:6.13. This means local and global positions can still be used to direct the search space:

$$\vec{P}_{Best} \equiv \begin{bmatrix} \Gamma_1(1) \Rightarrow \min \vec{q}_e(1) & \Gamma_1(4) \Rightarrow \min \vec{q}_e(1) \& \& \vec{q}_e(2) \\ \Gamma_1(2) \Rightarrow \min \vec{q}_e(2) & \Gamma_1(5) \Rightarrow \min \vec{q}_e(1) \& \& \vec{q}_e(3) \\ \Gamma_1(3) \Rightarrow \min \vec{q}_e(3) & \Gamma_1(6) \Rightarrow \min \vec{q}_e(2) \& \& \vec{q}_e(3) \\ \Gamma_2(1) \Rightarrow \min \vec{\omega}_e(1) & \Gamma_2(4) \Rightarrow \min \vec{\omega}_e(1) \& \& \vec{\omega}_e(2) \\ \Gamma_2(2) \Rightarrow \min \vec{\omega}_e(2) & \Gamma_2(5) \Rightarrow \min \vec{\omega}_e(1) \& \& \vec{\omega}_e(3) \\ \Gamma_2(3) \Rightarrow \min \vec{\omega}_e(3) & \Gamma_2(6) \Rightarrow \min \vec{\omega}_e(2) \& \& \vec{\omega}_e(3) \end{bmatrix} \tag{6.22a}$$

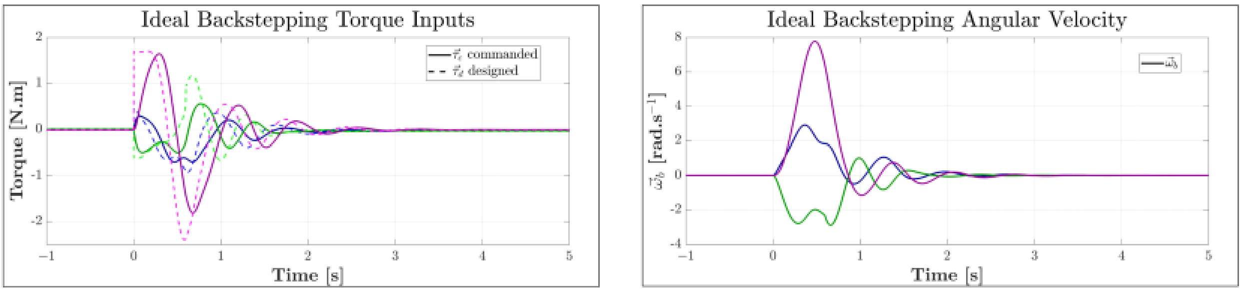
$$\vec{G}_{Best} \equiv \begin{bmatrix} \Gamma_1(1) \Rightarrow \min \vec{\zeta}_{IBC}(1) & \Gamma_1(4) \Rightarrow \min \vec{\zeta}_{IBC}(1) \& \& \vec{\zeta}_{IBC}(2) \\ \Gamma_1(2) \Rightarrow \min \vec{\zeta}_{IBC}(2) & \Gamma_1(5) \Rightarrow \min \vec{\zeta}_{IBC}(1) \& \& \vec{\zeta}_{IBC}(3) \\ \Gamma_1(3) \Rightarrow \min \vec{\zeta}_{IBC}(3) & \Gamma_1(6) \Rightarrow \min \vec{\zeta}_{IBC}(2) \& \& \vec{\zeta}_{IBC}(3) \\ \Gamma_2(1) \Rightarrow \min \vec{\zeta}_{IBC}(1) & \Gamma_2(4) \Rightarrow \min \vec{\zeta}_{IBC}(1) \& \& \vec{\zeta}_{IBC}(2) \\ \Gamma_2(2) \Rightarrow \min \vec{\zeta}_{IBC}(2) & \Gamma_2(5) \Rightarrow \min \vec{\zeta}_{IBC}(1) \& \& \vec{\zeta}_{IBC}(3) \\ \Gamma_2(3) \Rightarrow \min \vec{\zeta}_{IBC}(3) & \Gamma_2(6) \Rightarrow \min \vec{\zeta}_{IBC}(2) \& \& \vec{\zeta}_{IBC}(3) \end{bmatrix} \tag{6.22b}$$

When optimized two sets of gain coefficient matrices are:

$$\Gamma_1 = \begin{bmatrix} 5.8631 & 0.0515 & 1.0221 \\ 0.0515 & 13.8375 & 0.8533 \\ 1.0221 & 0.8533 & 11.9644 \end{bmatrix} \quad \text{and} \quad \Gamma_2 = \begin{bmatrix} 9.1127 & 0.2887 & 0.1353 \\ 0.2887 & 6.8389 & 0.1971 \\ 0.1353 & 0.1871 & 2.5294 \end{bmatrix} \tag{6.23}$$



(a) Quaternion attitude step



(b) Plant input torques

(c) Angular velocity

Figure 6.10: Independent diagonal PD

The regular attitude step's response in Fig:6.10a shows a dramatically faster response, with notable oscillations at the settling point. The step settles in $t_{95} = 1.6403$ s; almost twice as fast as a basic PD controller. However, the commanded control inputs in Fig:6.10b are the largest of all the attitude controllers considered. The Ideal Backstepping controller is by far the most aggressive which leads to sizeable and perhaps unsatisfactory overshoot. Moreover the commanded angular velocity changes for the IBC controller are, on average, twice that of the previous control laws. The Adaptive backstepping controller is tested and discussed later in Sec:6.6.1 in the context of robust trajectory stability; rather than stepped controller performances here...

6.4 Position Controllers

Following the attitude controller optimization, a similar approach was applied to the two proposed position control laws (in Sec:4.7). It is important to specify that, for position controller optimization, a plant dependent diagonal PD attitude controller (Sec:6.3.1) was used to stabilize the coupled attitude dynamics. To test each swarm's position controller coefficient performance the attitude setpoint was kept at a constant $Q_d = [1\ \vec{0}\,]^T$, while various position setpoints were applied. The same basic pseudo inversion allocator (Sec:5.3.1) was used for position control to allocate out the virtual control input $\vec{\nu}_d$. Each position setpoint is defined in the inertial frame:

$$\vec{\mathcal{E}}_d(t) \triangleq \begin{bmatrix} X_d(t) & Y_d(t) & Z_d(t) \end{bmatrix}^T \qquad \in \mathcal{F}^I \tag{6.24}$$

A collection of position setpoints were tested where each setpoint was positioned on the surface of a sphere at a radius of $C = 5$ m away from a central starting point. That starting position was consistently tested at $\vec{\mathcal{E}}_0 = [5\ 5\ 5]^T$ m, relative to the inertial frame's origin. Each setpoint was then distanced away from $\vec{\mathcal{E}}_0$ as per a rotated radial arm:

$$\vec{\mathcal{E}}_d(t) = \vec{\mathcal{E}}_0 + R_y(\theta_y)R_x(\phi_x)\begin{bmatrix} 0 & 0 & 5 \end{bmatrix}^T \tag{6.25}$$

Both test angles ϕ_x and θ_y rotate the radial arm C for a range $\phi_x \in [-180^\circ : 180^\circ]$ and $\theta_y \in [-90^\circ : 90^\circ]$; both at 30 ° increments. That results in test space position surface illustrated in Fig:6.11, with a total of 91 position setpoints to test. Performance of each position step was evaluated with another ITAE integral for the position and translational velocity errors, both transformed into the *body frame*, $\mathcal{F}^b$.

$$\zeta_{\mathcal{E}} = C_X \int_{t=0}^{15} t|\vec{X}_e(t)|.dt + C_v \int_{t=0}^{15} t|\vec{v}_e(t)|.dt \tag{6.26}$$

Again the simulation was given $t = 15$ s to reach its settling point when stepped from the starting point. Weighting coefficients C_X and C_v prioritize position and velocity errors respectively, both were weighted equally such that $C_X = C_v = 1$. Each swarm was then tested 91 times and the resultant cost of Eq:6.26 was averaged for an overall performance metric. Only plant dependent compensating position controllers were considered and optimized for the position control loop.

Not compensating for the graviational force acceleration applied to the vehicle in its differential equation of motion, Eq:3.129b, *would* result in instability. To compare the relative performance of position controllers a constant step test was applied in both cases:

$$\vec{\mathcal{E}}_d = \begin{bmatrix} X_d & Y_d & Z_d \end{bmatrix}^T = \begin{bmatrix} 7.5 & 4 & 3 \end{bmatrix}^T \quad \text{m}, \quad \in \mathcal{F}^I \tag{6.27}$$

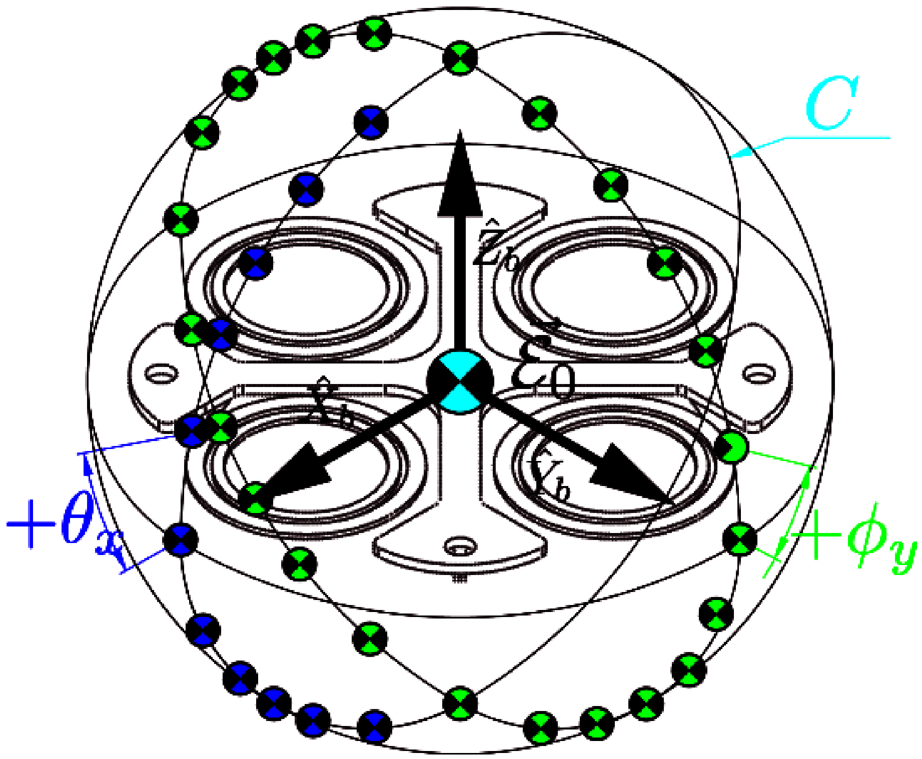


Figure 6.11: Position setpoint workspace

6.4.1 PD

The reference case for position control is the Proportional-Derivative controller, presented in Sec:4.7.1. The PD position controller designs a control force input, from Eq:4.87:

$$\vec{F}_{PD} = K_p\vec{X}_e + K_d\dot{\vec{X}}_e + \hat{\omega}_b \times m_b\hat{v}_b - m_b\vec{G}_b \qquad \in \mathcal{F}^b \tag{6.28}$$

Where both K_p and K_d are diagonal gain coefficient matrices. The introduction of symmetric coefficients did not yield any improvements for the attitude plant in Sec:6.3.1 so it was not investigated in the context of position control. The two gain coefficients for the PD controller are structured as follows:

$$K_p \triangleq \begin{bmatrix} K_p(1) & 0 & 0 \\ 0 & K_p(2) & 0 \\ 0 & 0 & K_p(3) \end{bmatrix} \quad \text{and} \quad K_d \triangleq \begin{bmatrix} K_d(1) & 0 & 0 \\ 0 & K_d(2) & 0 \\ 0 & 0 & K_d(3) \end{bmatrix} \tag{6.29}$$

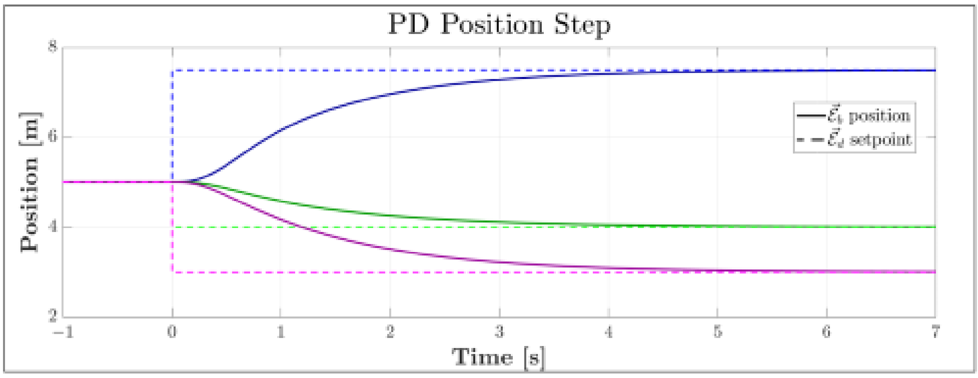
Each coefficient matrix acts on the position error vector, $\vec{X}_e$, and the velocity error vector, $\vec{v}_e$, independently. As a result the local and global coefficient optimum positions are selected simply as:

$$\vec{P}_{Best} \equiv \begin{bmatrix} K_p(1) \Rightarrow \min \vec{X}_e(1) \\ K_p(2) \Rightarrow \min \vec{X}_e(2) \\ K_p(3) \Rightarrow \min \vec{X}_e(3) \\ K_d(1) \Rightarrow \min \vec{v}_e(1) \\ K_d(2) \Rightarrow \min \vec{v}_e(2) \\ K_d(3) \Rightarrow \min \vec{v}_e(3) \end{bmatrix} \quad \text{and} \quad \vec{G}_{Best} \equiv \begin{bmatrix} K_p(1) \Rightarrow \min \vec{\zeta}_{PD}(1) \\ K_p(2) \Rightarrow \min \vec{\zeta}_{PD}(2) \\ K_p(3) \Rightarrow \min \vec{\zeta}_{PD}(3) \\ K_d(1) \Rightarrow \min \vec{\zeta}_{PD}(1) \\ K_d(2) \Rightarrow \min \vec{\zeta}_{PD}(2) \\ K_d(3) \Rightarrow \min \vec{\zeta}_{PD}(3) \end{bmatrix} \tag{6.30}$$

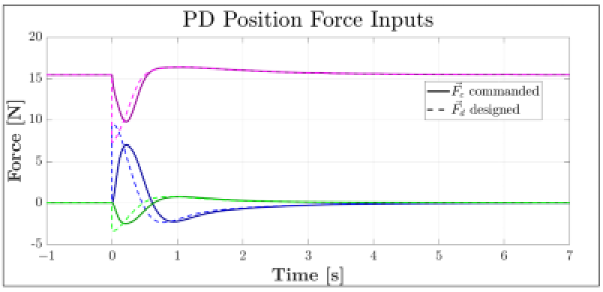
The following optimized coefficients were then produced:

$$K_p = \begin{bmatrix} 2.4167 & 0 & 0 \\ 0 & 2.1557 & 0 \\ 0 & 0 & 2.5904 \end{bmatrix} \quad \text{and} \quad K_d = \begin{bmatrix} 3.4794 & 0 & 0 \\ 0 & 3.3846 & 0 \\ 0 & 0 & 3.8698 \end{bmatrix} \tag{6.31}$$

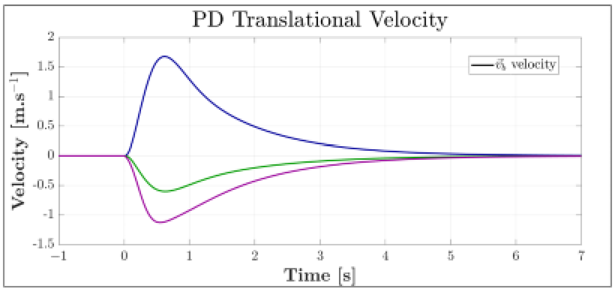
The inertial position's step had a response shown in Fig:6.12a; stepping from the initial position to the setpoint(s) described in Eq:6.8. The position step settled in $t_{95} = 4.007$ s without any overshoot. Not shown but still considered is the effect a position step has on the attitude plant's stability, which still remained stable at the origin with no deviations. Because the attitude setpoint is $Q_d = [1\ \vec{0}\,]^T$; almost all the force requirement in steady state is to oppose the gravitational downward force acting on the body, Fig:6.12b.



(a) Position step



(b) Plant input forces



(c) Position translational velocity

Figure 6.12: Position PD

6.4.2 Ideal and Adaptive Position Backstepping

The second and final position controller to be tested is the Ideal Backstabbing controller, the only exponentially stable position control law proposed. As is the case with attitude IBC, the coefficients selected for the Ideal Backstepping case are used again for the Adaptive cases; the latter is evaluated subsequently in Sec:6.6.2. Recalling the position IBC structure from Sec:4.7.2:

$$\vec{F}_{IBC} = m_b\Big(\big(1 + \Gamma_1\Gamma_2\big)\hat{z}_1 - \big(\Gamma_1 + \Gamma_2\big)\hat{v}_b\Big) + \hat{\omega}_b \times m_b\hat{v}_b - m_b\vec{G}_b \qquad \in \mathcal{F}^b \tag{6.32}$$

The two positive symmetric coefficient gain matrices in Eq:6.32 are structured as:

$$\Gamma_1 \triangleq \begin{bmatrix} \Gamma_1(1) & \Gamma_1(4) & \Gamma_1(5) \\ \Gamma_1(4) & \Gamma_1(2) & \Gamma_1(6) \\ \Gamma_1(5) & \Gamma_1(6) & \Gamma_1(3) \end{bmatrix} \quad \text{and} \quad \Gamma_2 \triangleq \begin{bmatrix} \Gamma_2(1) & \Gamma_2(4) & \Gamma_2(5) \\ \Gamma_2(4) & \Gamma_2(2) & \Gamma_2(6) \\ \Gamma_2(5) & \Gamma_2(6) & \Gamma_2(3) \end{bmatrix} \tag{6.33}$$

Both attitude and position ideal backstepping controllers have coefficients which act on both plant's error and error rates. This makes local and global coefficient position selection difficult without adversely affecting the swarm's optimization trajectory process.

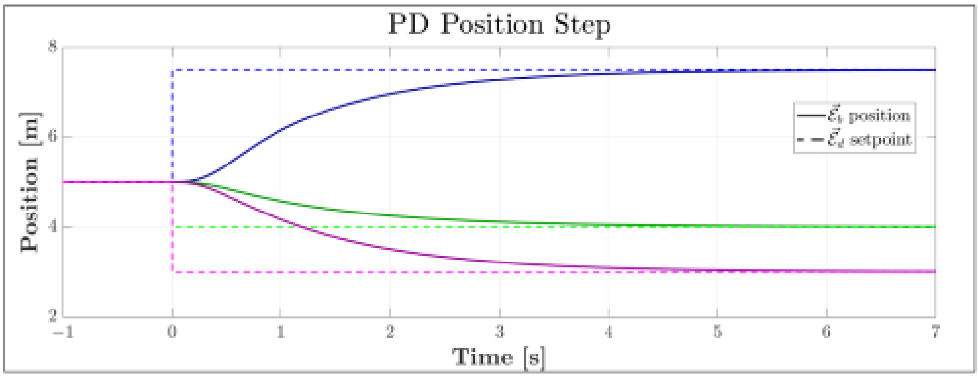
Using the first coefficient matrix Γ_1 to prioritize position tracking errors $\vec{X}_e$ and relegating Γ_2 to settle velocity errors $\vec{v}_e$; the local and global best positions are chosen as follows:

$$\vec{P}_{Best} \equiv \begin{bmatrix} \Gamma_1(1) \Rightarrow \min \vec{\mathcal{X}}_e(1) & \Gamma_1(4) \Rightarrow \min \vec{\mathcal{X}}_e(1)\&\&\vec{\mathcal{X}}_e(2) \\ \Gamma_1(2) \Rightarrow \min \vec{\mathcal{X}}_e(2) & \Gamma_1(5) \Rightarrow \min \vec{\mathcal{X}}_e(1)\&\&\vec{\mathcal{X}}_e(3) \\ \Gamma_1(3) \Rightarrow \min \vec{\mathcal{X}}_e(3) & \Gamma_1(6) \Rightarrow \min \vec{\mathcal{X}}_e(2)\&\&\vec{\mathcal{X}}_e(3) \\ \Gamma_2(1) \Rightarrow \min \vec{v}_e(1) & \Gamma_2(4) \Rightarrow \min \vec{v}_e(1)\&\&\vec{v}_e(2) \\ \Gamma_2(2) \Rightarrow \min \vec{v}_e(2) & \Gamma_2(5) \Rightarrow \min \vec{v}_e(1)\&\&\vec{v}_e(3) \\ \Gamma_2(3) \Rightarrow \min \vec{v}_e(3) & \Gamma_2(6) \Rightarrow \min \vec{v}_e(2)\&\&\vec{v}_e(3) \end{bmatrix} \tag{6.34a}$$

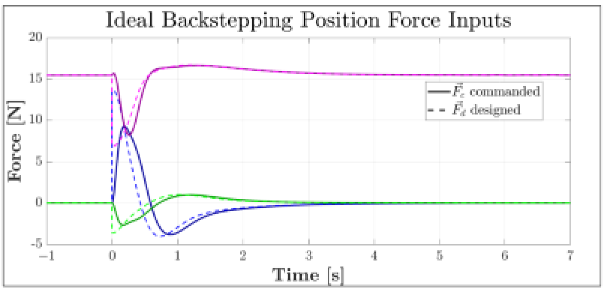
$$\vec{G}_{Best} \equiv \begin{bmatrix} \Gamma_1(1) \Rightarrow \min \vec{\zeta}_{IBC}(1) & \Gamma_1(4) \Rightarrow \min \vec{\zeta}_{IBC}(1)\&\&\vec{\zeta}_{IBC}(2) \\ \Gamma_1(2) \Rightarrow \min \vec{\zeta}_{IBC}(2) & \Gamma_1(5) \Rightarrow \min \vec{\zeta}_{IBC}(1)\&\&\vec{\zeta}_{IBC}(3) \\ \Gamma_1(3) \Rightarrow \min \vec{\zeta}_{IBC}(3) & \Gamma_1(6) \Rightarrow \min \vec{\zeta}_{IBC}(2)\&\&\vec{\zeta}_{IBC}(3) \\ \Gamma_2(1) \Rightarrow \min \vec{\zeta}_{IBC}(1) & \Gamma_2(4) \Rightarrow \min \vec{\zeta}_{IBC}(1)\&\&\vec{\zeta}_{IBC}(2) \\ \Gamma_2(2) \Rightarrow \min \vec{\zeta}_{IBC}(2) & \Gamma_2(5) \Rightarrow \min \vec{\zeta}_{IBC}(1)\&\&\vec{\zeta}_{IBC}(3) \\ \Gamma_2(3) \Rightarrow \min \vec{\zeta}_{IBC}(3) & \Gamma_2(6) \Rightarrow \min \vec{\zeta}_{IBC}(2)\&\&\vec{\zeta}_{IBC}(3) \end{bmatrix} \tag{6.34b}$$

The optimized gain coefficients for Γ_1 and Γ_2 were then produced by the PSO algorithm:

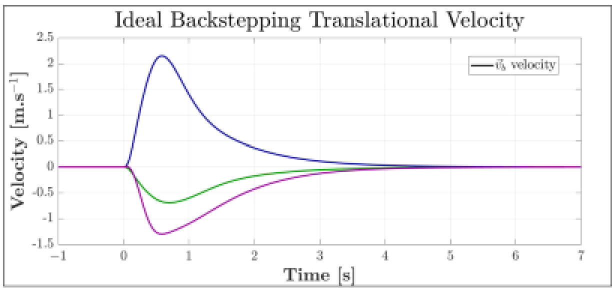
$$\Gamma_1 = \begin{bmatrix} r2.3409 & 0.1707 & -0.1644 \\ 0.1707 & 2.0493 & 0.1060 \\ -0.1644 & 0.1060 & 1.7322 \end{bmatrix} \quad \text{and} \quad \Gamma_2 = \begin{bmatrix} r1.5287 & 0.02928 & 0.0816 \\ 0.0292 & 1.4214 & -0.0410 \\ 0.0816 & -0.0410 & 1.4753 \end{bmatrix} \tag{6.35}$$



(a) Position step



(b) Plant input forces



(c) Position translational velocity

Figure 6.13: Position Backstepping Controller

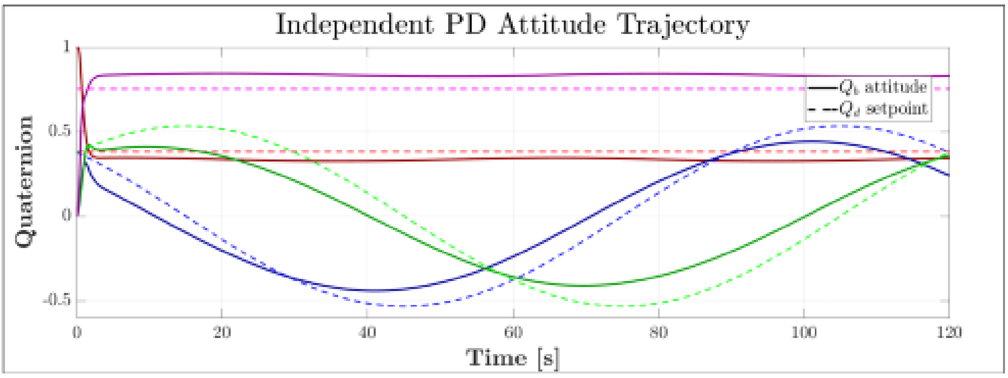
Fig:6.13a shows the step response to a change in translational position setpoint. Note that the position plotted in Fig:6.13a is the relative position in the inertial frame $\mathcal{F}^I$, not the backstepping input $\vec{X}_e$. The Ideal Backstepping controller settles in $t_{95} = 2.987$ s; demonstrating the improvement exponential stability has over asymptotic stability achieved by a PD controller previously...

It is not unexpected that with faster settling times greater input forces are commanded, the virtual and commanded input forces in Fig:6.13b spike to a larger degree than those previously in Fig:6.12b. The velocity step, in Fig:6.13c, follows a smooth rate change.

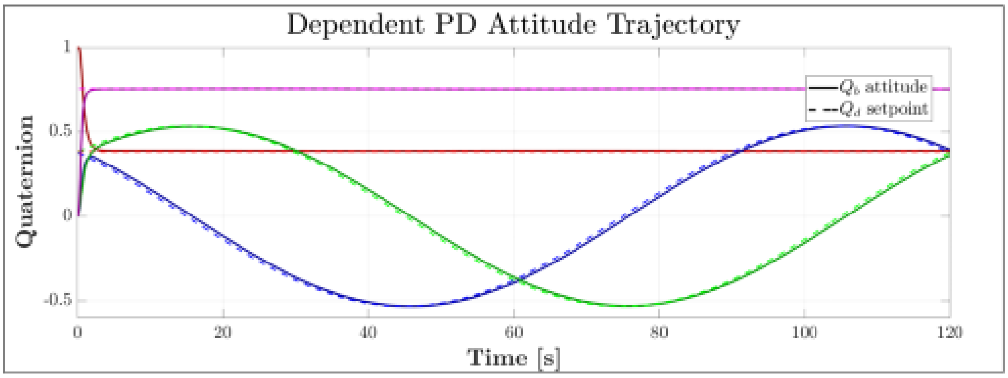
6.5 Setpoint Control Results

Each proposed attitude or position controllers was respectively stable in its own right. The trajectory used to corroborate dynamic setpoint tracking (Fig:6.2) is not a complicated one, using a slow orbital velocity of $\dot{\theta} = 0.5$ Hz such that a single orbit around $\vec{C}_0$ was completed every 120 s. Two Proportional Derivative *attitude* controllers are tested, both with diagonal gain matrices (from Eq:6.14) to compare the effects of plant dependent compensation. It was shown previously that symmetric gain coefficients yield no performance improvements for the PD case, so only diagonal coefficient matrices were used. Furthermore, Sec:6.3.1 demonstrated plant independent controllers result in steady state errors, the same is shown to be true for trajectory tracking. Adaptive backstepping controllers and their disturbance rejection properties are only discussed next in Sec:6.6.

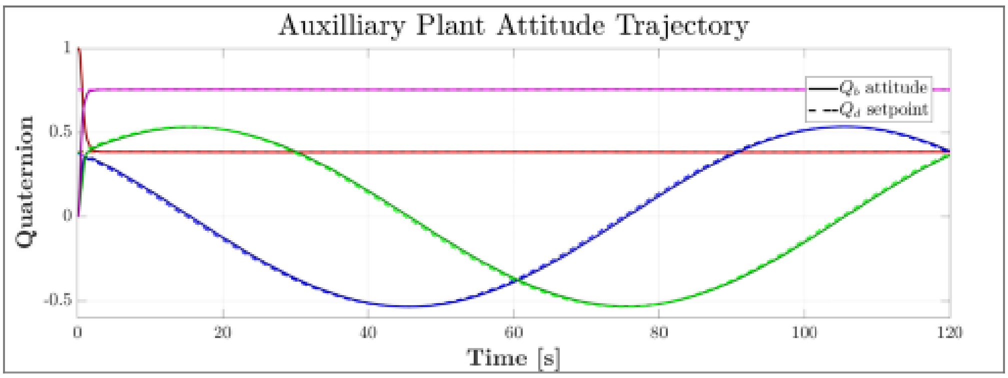
Each attitude controller was tested together with a common PD position controller, tracking the orbital XYZ position. Similarly each position controller is tested using a simple diagonal PD controller to track the attitude. The attitude controllers have an initial step to reach the orbital attitude from their starting attitude; $Q_0 = [1\ \vec{0}\,]^T$. Each control law was successful in tracking a given trajectory each with little to no error, except of course for the plant independent attitude controller, Fig:6.14a. Trajectory errors could be improved further with application of higher order state derivative in each controller (wherein $\vec{\omega}_d \neq \vec{0}$ and $\vec{v}_d \neq \vec{0}$).



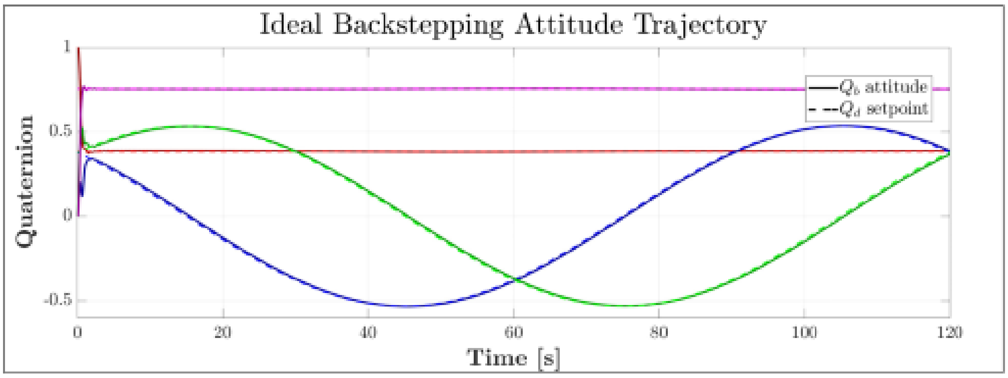
(a) Independent Proportional Derivative Attitude Controller



(b) Dependent Proportional Derivative Attitude Controller



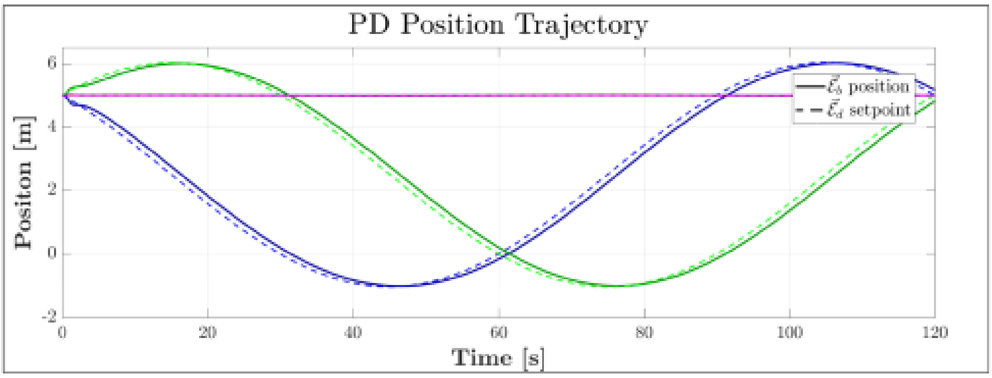
(c) Auxiliary Plant Attitude Controller



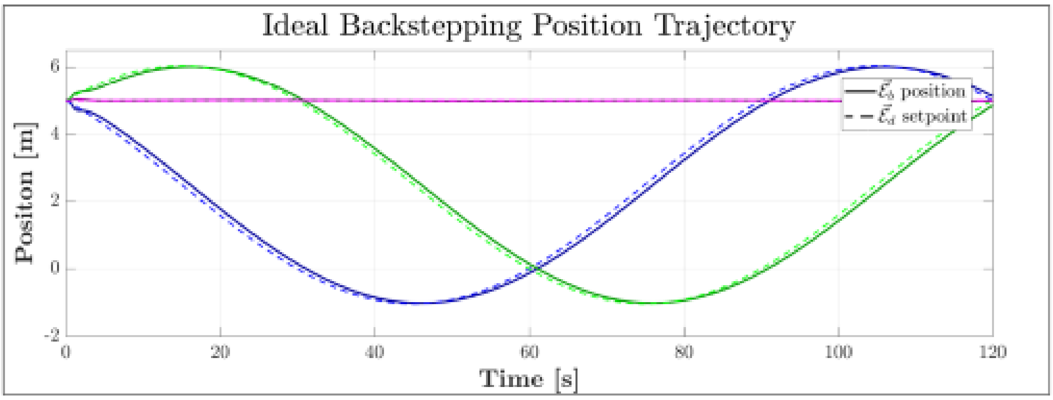
(d) Ideal Backstepping Controller

Figure 6.14: Attitude Trajectory Tracking

The only notable difference between each attitude controller in Fig:6.14 is that, at the initial attitude step, the Ideal Backstepping attitude controller (Fig:6.14d) oscillates whilst settling due to it's more aggressive response. The independent PD controller actually tracks some trajectory, but with a significant steady state error.



(a) Diagonal Proportional Derivative Controller



(b) Ideal Backstepping Controller

Figure 6.15: Position Trajectory Tracking

There is a small lag behind the position controller's trajectories, Fig:6.15. Once again this is a consequence of tracking only first order setpoints; if a velocity setpoint was applied as well that tracking error would be diminished...

6.6 Robust Stability and Disturbance Rejection

Despite deriving adaptive control laws in Sec:4.6.3 and Sec:4.7.2 for attitude and position controllers respectively; each of the proposed controls demonstrated acceptable stability under sizeable disturbances. App:C.4 shows each controller's trajectory response to uncompensated disturbances acting on the vehicle. The torque and force disturbances are now detailed...

6.6.1 Torque Disturbance Rejection

Torque turbulences are difficult to define without in-depth accompanying statistical and mathematical analysis. To expedite the stability/disturbance evaluation process, torque turbulences were approximated using a Dryden Gust model, [21, 79]. Alternatively the Von Karman aerospace disturbance model(s) could be implemented but that model is computationally more exhaustive.

Without going into too much detail, the Dryden Wind model produces turbulence signals from white noise filtered through a specified Dryden power spectrum. That power spectrum varies as per an aircraft's orientation, altitude and translational velocity. For the aircraft and trajectory under consideration here such a disturbance model is sufficient for simulating small interference patterns. Recalling then the torque disturbance observer derived for the attitude backstepping plant, from Eq:4.79:

$$\dot{\hat{L}} = -\Gamma_L J_b^{-1}(u)\big(\Gamma_1 \vec{q}_e - \vec{\omega_b}\big) \tag{6.36}$$

The gain adaptivity matrix Γ_L was tuned on steady state such that the observer's error deviation from an applied torque $\vec{L}$ was minimized. That resulted in the diagonal adaptivity matrix of $\Gamma_L = diag(29.58,\ 28.43,\ 4.60)$. The approximator tracks an applied disturbance as shown in Fig:6.16 over a disturbance range of ± 0.2 N.m, 20% of a stepped attitude's control input magnitude, for a short steady state test. Both pitch and roll torque approximator channels (ϕ and θ) track the torque with a relatively small error. Greater deviation from the applied torque does, however, occur in the ψ channel about the $\hat{Z}_b$ axis...

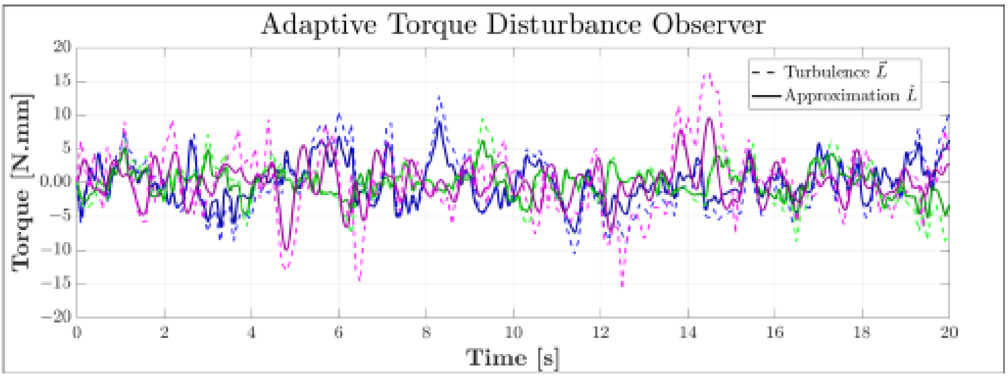


Figure 6.16: Attitude torque disturbance observer

Fig:6.17 then shows the Adaptive Backstepping controller's attitude response over the entire orbital trajectory whilst experiencing a fluctuating torque turbulence. The addition of a torque observer for compensation produces a slight improvement over an uncompensated IBC controller; shown in Fig:C.5c from App:C.4.

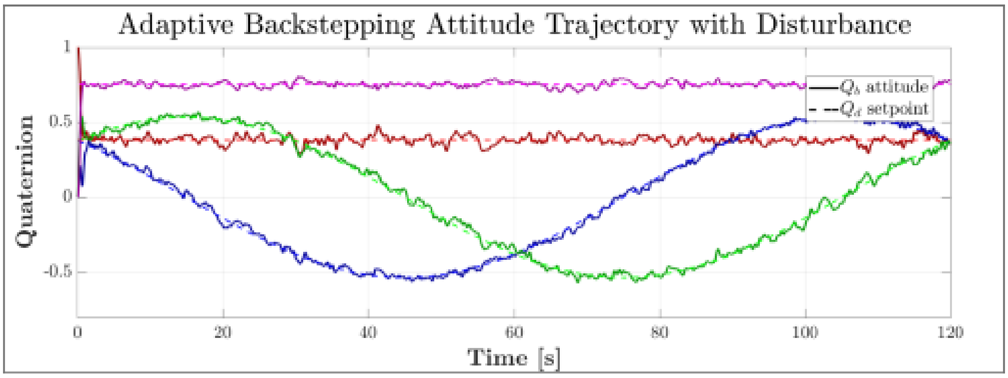


Figure 6.17: Adaptive backstepping attitude trajectory tracking

The cost of the disturbance approximator in the attitude plant is obviously more aggressive and greater bandwidth fluctuating input torques applied to the control plant. That could potentially reach actuator rate saturation, but for the trajectories tested here, were not saturation inducing.

6.6.2 Disturbance Force Rejection

Force disturbances are similarly emulated in simulation using a Dryden Gust model for wind turbulent velocity generation. Additionally, a wind vector field across the inertial frame test space was also used to introduce a constant force offset throughout the trajectory simulation. The force disturbance observer, from Eq:4.106a, has an estimate update rule such that:

$$\dot{\hat{D}} = -m_b^{-1}\Gamma_D\Big(\Gamma_1\vec{X}_e - \vec{v}_b\Big) \tag{6.37}$$

Where $\vec{X}_e$ is the inertial position error transformed to the body frame, $\vec{X}_e = Q_b \otimes \vec{\mathcal{E}}_e \otimes Q_b^*$. Then Γ_D is the force disturbance observer's adaptivity gain matrix. Using the coefficients $\Gamma_D = diag(4.20,\ 3.84,\ 3.97)$ the observer tracks a force disturbance acting on the vehicle over a range of $[-4 : 8]$ N. Fig:6.18 shows how the force observer adapts to the variable force turbulence applied, the plot is taken over an entire simulation (until $t = 120$ s) to illustrate the vector field effects.

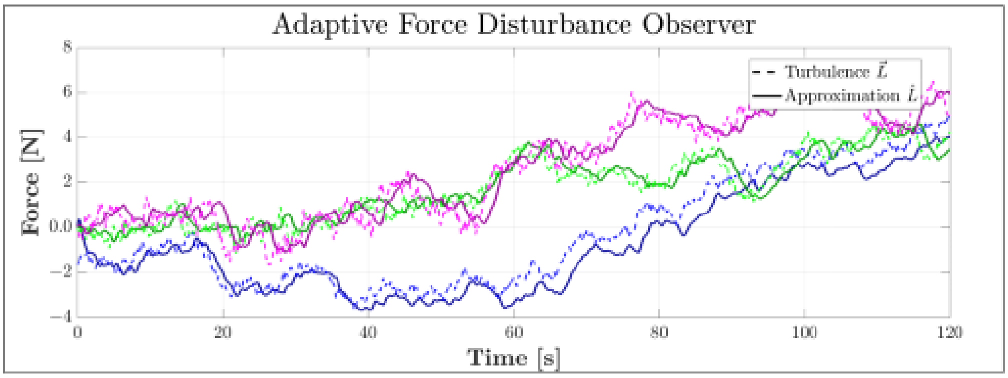


Figure 6.18: Position force disturbance observer

The position adaptive backstepping controller then tracks the inertial frame trajectory as shown in Fig:6.19. Again improving the trajectory tracking performance slightly when compared to the Ideal backstepping case from Fig:C.6b; but even without adaptive disturbance compensation, the plant is stable throughout the trajectory albeit somewhat noisy. The addition of a vector force field results in a fluctuating offset error from the trajectory, despite the adaptive compensation applied to the control loop.

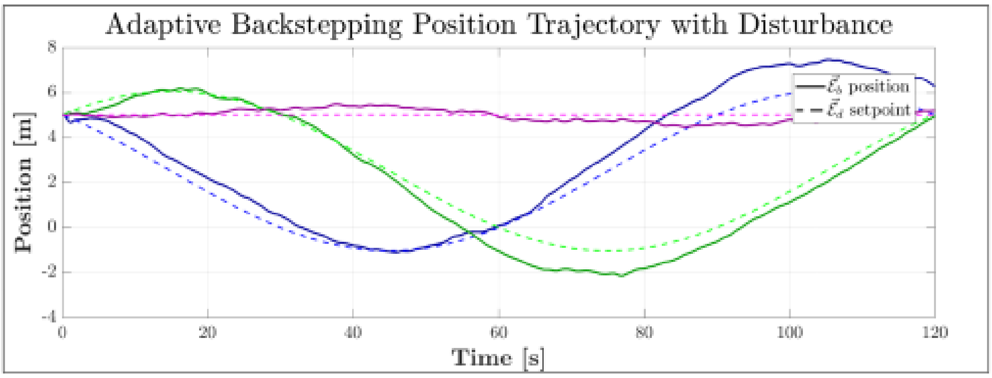


Figure 6.19: Adaptive backstepping position trajectory tracking

6.7 Allocation Tests

The various allocation rules, as derived in Ch:5, implement virtual control inputs to solve for explicit actuator positions. Each of the allocators tested here were compared with basic position and attitude Proportional Derivative controllers commanding a virtual input.

The abstraction applied to achieve an affine relationship required for inversion allocation (in Eq:6.38) meant that actuator transfer rates were independent from the allocation rule applied.

$$\vec{\nu}_c = B'(\vec{\mathbf{x}}, t)u = \begin{bmatrix} \mathbb{I}_{3\times3} & \mathbb{I}_{3\times3} & \mathbb{I}_{3\times3} & \mathbb{I}_{3\times3} \\ [\vec{L}_1]_\times & [\vec{L}_2]_\times & [\vec{L}_3]_\times & [\vec{L}_4]_\times \end{bmatrix} \begin{bmatrix} \vec{T}_1 & \dots & \vec{T}_4 \end{bmatrix}^T \tag{6.38a}$$

$$u_i = \begin{bmatrix} \Omega_i & \lambda_i & \alpha_i \end{bmatrix}^T = R^\dagger(\vec{\mathbf{x}}, \vec{T}_i, t) \ \text{ for } i \in [1:4] \tag{6.38b}$$

The transfer rate at which physically commanded inputs implement virtually designed control inputs, $\vec{\nu}_c \to \vec{\nu}_d$, is affected by the thrust version equation $R^\dagger(\vec{\mathbf{x}}, t)$, not allocation rules $B'(\vec{\mathbf{x}}, t)$. The consequence of this is that, in the context of actuator transfer rates, each allocation rule performed almost identically. Inverse solutions to Eq:5.18 solve for the quadratic least squares minimized actuator positions. That means that each $|\vec{T}_i|$ within the $\mathbb{R}^{1\times12}$ matrix $|\vec{T}_{1\to4}|$ is minimized. The solution is an actuator cost efficient one. In general psuedo inversion, wieghted and priority normalized inverse allocators each stem from Eq:5.19:

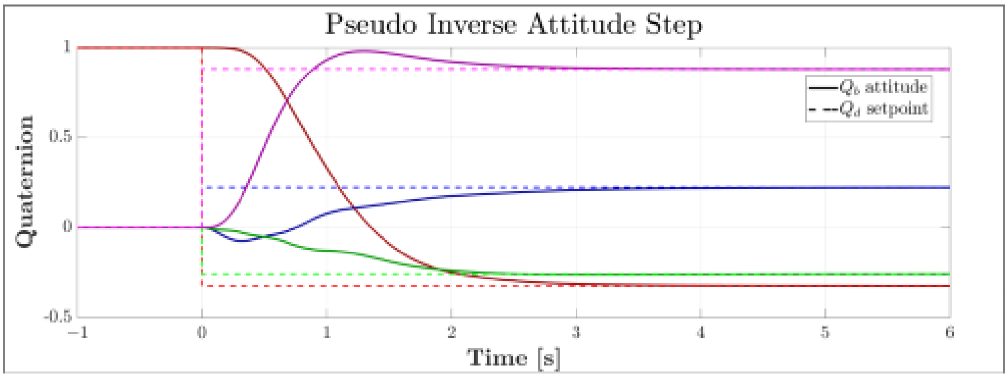
$$\begin{bmatrix} \vec{T}_{1\to4} \end{bmatrix} = \big(\mathbb{I}_{m\times m} - CB(\vec{\mathbf{x}}, t)\big)\vec{T}_p + C\vec{\nu}_d \tag{6.39a}$$

$$C = W^{-1}B^T(\vec{\mathbf{x}}, t)\big(B(\vec{\mathbf{x}}, t)W^{-1}B^T(\vec{\mathbf{x}}, t)\big)^{-1} \tag{6.39b}$$

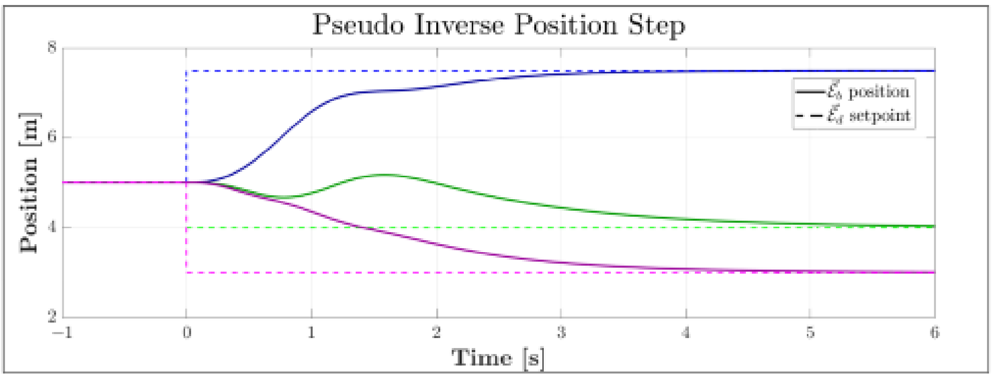
A combined step of setpoints for attitude *and* position states is used to compare each allocation rule. A pseudo inversion allocator, Eq:5.20, is used as the reference case to which subsequent allocator algorithms are evaluated against. The typical setpoint used for both position and attitude steps, with attitude in Euler angles $\vec{\eta}_d$ not quaternions Q_d, is:

$$\vec{\mathbf{x}}_d = \begin{bmatrix} \vec{\mathcal{E}}_d \\ \vec{\eta}_d \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 7.5 & 4 & 3 \end{bmatrix}^T \\ \begin{bmatrix} -142 & 167 & -45 \end{bmatrix}^T \end{bmatrix} \quad \begin{bmatrix} [\text{m}] \\ [^\circ] \end{bmatrix} \tag{6.40}$$

A pseudo inverse allocation solves for $\vec{T}_{1\to4}$ using $B^\ddagger(\vec{\mathbf{x}}, t)\vec{\nu}_d$, from Eq:5.20. Fig:6.20 shows the combined position and attitude step responses. The combined attitude and position step response with a pseudo inverse allocator $B^\ddagger(\vec{\mathbf{x}}, t)$ settles for *both* states in $t_{95} = 5.6233$ s from the state step.



(a) Attitude Step



(b) Position Step

Figure 6.20: Pseudo-Inverse step response

The preferred allocator positions, described in Sec:5.3.2, are hovering conditions defined with respect to either the inertial or body frames, $\mathcal{F}^I$ and $\mathcal{F}^b$. At steady state with an attitude at the origin, $Q_d = [1\ \vec{0}]^T$; the controller commands the virtual control input:

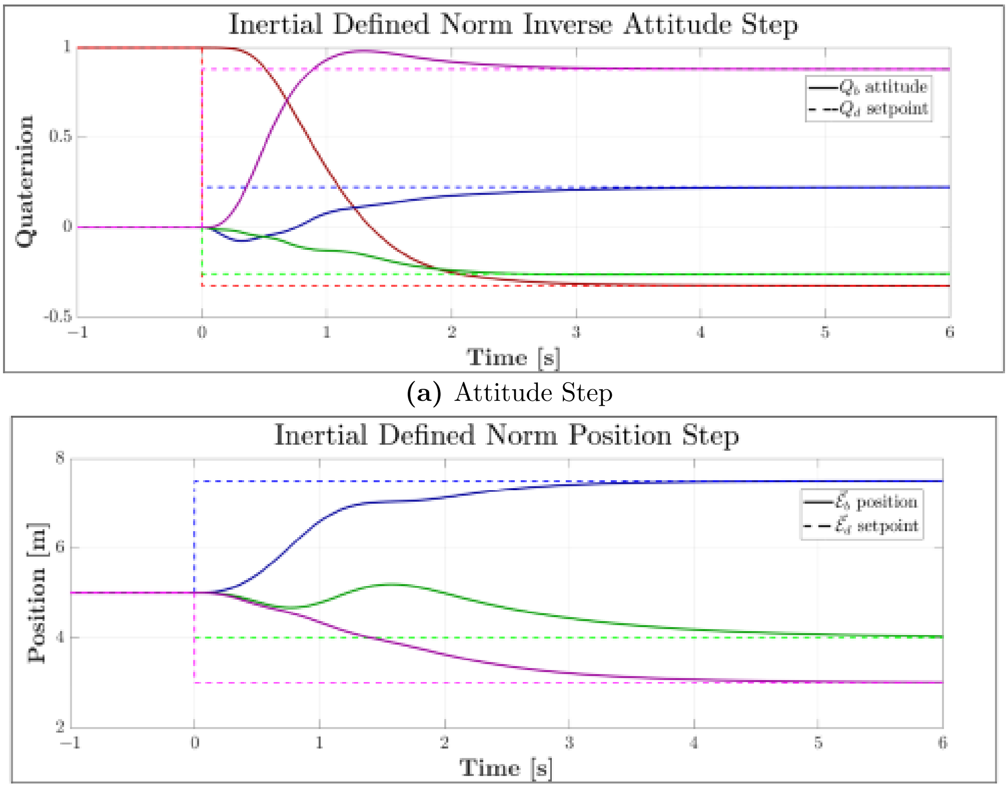
$$\vec{\nu}_p = \begin{bmatrix} \vec{F}_p \\ \vec{\tau}_p \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 0 & 0 & 15.45 \end{bmatrix}^T \\ \begin{bmatrix} 0.25 & 0.50 & -1.89 \end{bmatrix}^T \end{bmatrix} \quad \begin{bmatrix} [\text{N}] \\ [\text{N.mm}] \end{bmatrix} \quad \in \mathcal{F}^{b,I} \tag{6.41}$$

The small amount of control torque applied in Eq:6.41, about the $\hat{Z}_{I/b}$ axis, is to compensate for net gravitational torque due to the eccentric center of gravity and resultant aerodynamic torque $\vec{\tau}_H$ from the propellers rotational velocity. Applying the pseudo inverse allocation rule to the preferred input $\vec{\nu}_p$ in Eq:6.41 produces the following actuator positions which command hovering conditions:

$$\vec{T}_p^I = B^{\ddagger}(\mathbf{x}, t)\vec{\nu}_p = \begin{bmatrix} T_{1x} & T_{1Y} & T_{1Z} & \dots & \dots & T_{4x} & T_{4y} & T_{4z} \end{bmatrix} \tag{6.42a}$$

$$= \Big[\begin{bmatrix} 0.00 & -0.02 & 3.86 \end{bmatrix} \ \begin{bmatrix} 0.02 & 0 & 3.86 \end{bmatrix} \ \begin{bmatrix} 0 & 0.02 & 3.86 \end{bmatrix} \ \begin{bmatrix} -0.02 & 0 & 3.86 \end{bmatrix}\Big]^T \quad \text{N} \tag{6.42b}$$

Testing the same attitude and position setpoint steps, but with preferred actuator hovering conditions relative to the inertial frame (illustrated in Fig:5.2) produces a response shown in Fig:6.21. The plant settles in $t_{95} = 5.617$ s with a practically identical response to the pseudo-inverse case presented before in Fig:6.20.



(a) Attitude Step

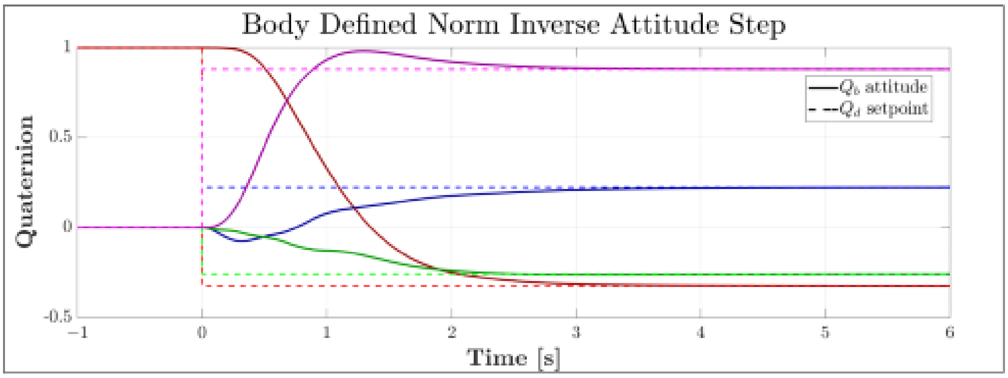
(b) Position Step

Figure 6.21: Inertial hover preferred actuator step response

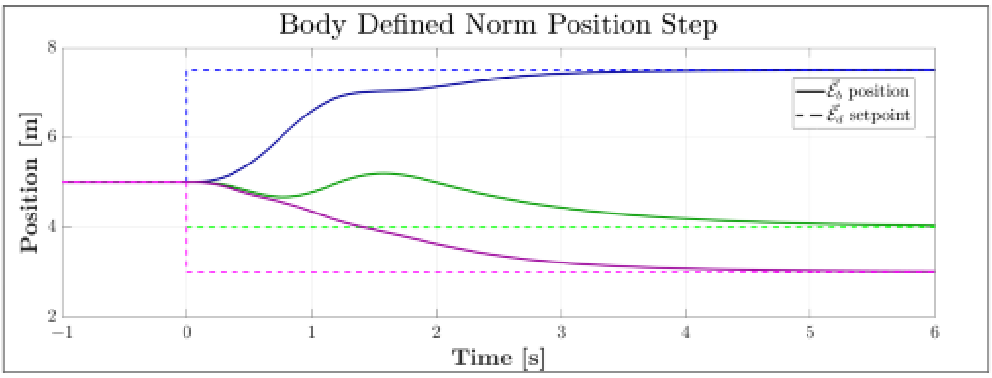
Transformation of those hovering conditions in Eq:6.41 from the inertial frame to the body frame is applied through an instantaneous quaternion transformation:

$$\vec{\nu}_p' = \begin{bmatrix} Q_b \otimes \vec{F}_p \otimes Q_b^* \\ Q_b \otimes \vec{\tau}_p \otimes Q_b^* \end{bmatrix} \quad \in \mathcal{F}^b \tag{6.43}$$

The hovering conditions are then always a function of the body's instantaneous attitude. However, because the control plant only tracks first order setpoints, the controller naturally tends towards stability at each controller interval, irrespective of the allocator rule applied. Again the plant settles in roughly the same time, $t_{95} = 5.614$ s, with a response shown in Fig:6.22. The difference between the two preferred allocator positions has no consequence on the performance of the control loop.



(a) Attitude Step



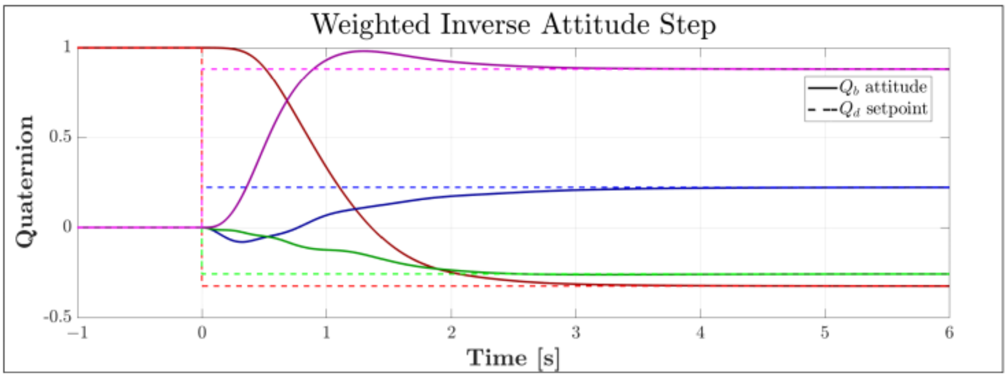
(b) Position Step

Figure 6.22: Body frame hover preferred actuator step response

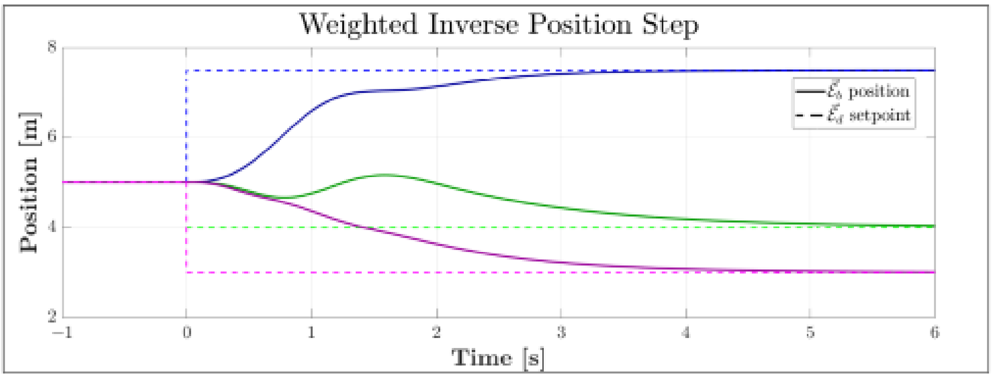
The weighted actuator allocation rule, proposed in Sec:5.3.3, prioritizes the use of certain input thrust components in Eq:6.38a. The weighting matrix is a 12×12 set of coefficients which bias various allocators as illustrated in Fig:**??**. In order to reduce slack and ensure setpoint tracking , being the primary control objective, each weighting coefficient row and column was constrained to a normalized summation. Furthermore it was proposed that coefficients were selected based on an optimization as per the penalty function Eq:5.26. In practice the weighting coefficients have no bearing on the input plant settling time for each actuator module; to demonstrate a weighted actuator case the following weighting matrix was used for $C = W^{-1}B^T(B.W^{-1}.B^T)^{-1}$ from Eq:6.39b:

$$W \triangleq \begin{bmatrix} \begin{bmatrix}72&9&9\\9&72&9\\9&9&72\end{bmatrix} & \begin{bmatrix}0&0&0\\0&0&0\\0&0&0\end{bmatrix} & \begin{bmatrix}8&1&1\\1&8&1\\1&1&8\end{bmatrix} & \begin{bmatrix}0&0&0\\0&0&0\\0&0&0\end{bmatrix} \\ \begin{bmatrix}0&0&0\\0&0&0\\0&0&0\end{bmatrix} & \begin{bmatrix}72&9&9\\9&72&9\\9&9&72\end{bmatrix} & \begin{bmatrix}0&0&0\\0&0&0\\0&0&0\end{bmatrix} & \begin{bmatrix}8&1&1\\1&8&1\\1&1&8\end{bmatrix} \\ \begin{bmatrix}8&1&1\\1&8&1\\1&1&8\end{bmatrix} & \begin{bmatrix}0&0&0\\0&0&0\\0&0&0\end{bmatrix} & \begin{bmatrix}72&9&9\\9&72&9\\9&9&72\end{bmatrix} & \begin{bmatrix}0&0&0\\0&0&0\\0&0&0\end{bmatrix} \\ \begin{bmatrix}0&0&0\\0&0&0\\0&0&0\end{bmatrix} & \begin{bmatrix}8&1&1\\1&8&1\\1&1&8\end{bmatrix} & \begin{bmatrix}0&0&0\\0&0&0\\0&0&0\end{bmatrix} & \begin{bmatrix}72&9&9\\9&72&9\\9&9&72\end{bmatrix} \end{bmatrix} \times 10^{-3} \tag{6.44}$$

The weighted allocator's response in Fig:6.23 shows the minor differences between each of the above allocation rules. The final weighted inversion allocator applied with Eq:6.44 settled in $t_{95} = 5.618$. The inversion based allocator's requirement for an affine effectiveness relationship meant that actuator transfer functions were separated from the allocation block, this made the actuator transfer rates independent from the allocation rule applied. Because of the constraint that each allocator still meets the slack variable requirements of setpoint tracking, each allocator's performance is much of the same. The twelve produced thrust component inputs are, within a margin of error, effectively the same across each allocation rule. . .



(a) Attitude Step



(b) Position Step

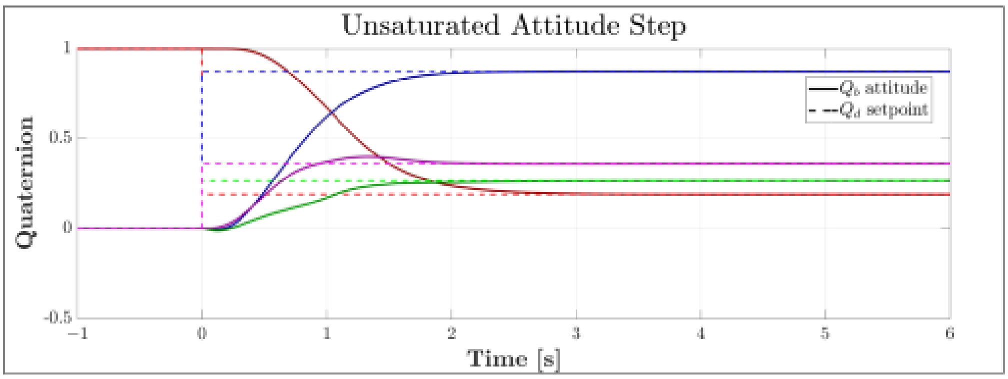
Figure 6.23: Weighted actuator allocation step response

6.8 Input Saturation

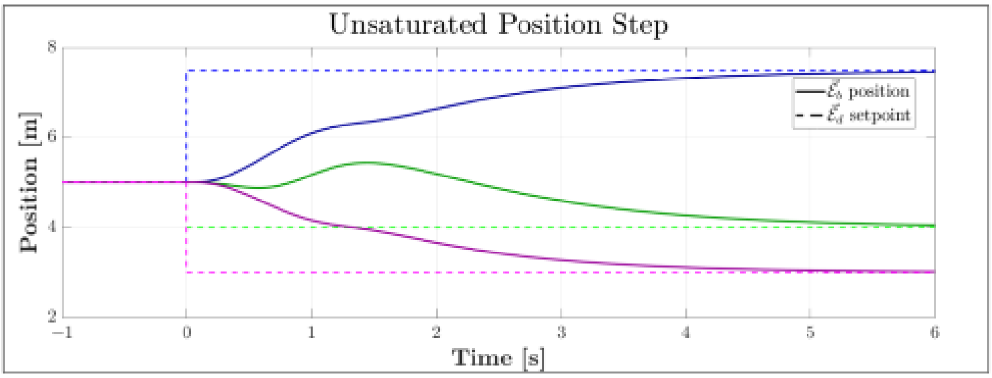
The introduction of a rotational limit to the actuating servos was a design decision made previously in Sec:2.4.1. The ±90° limit on the servos is something which can easily be solved by changing the mechanical design to incorporate continuous rotation actuators. The standard state setpoint step used thus far *does not* command any of the actuators beyond their rotational limits, neither does the applied trajectory tracking loop. Alternatively, the following state setpoint is used:

$$\vec{\mathbf{x}}_d' = \begin{bmatrix} \vec{\mathcal{E}}_d' \\ \vec{\eta}_d' \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 7.5 & 4 & 3 \end{bmatrix}^T \\ \begin{bmatrix} -142 & 35 & -45 \end{bmatrix}^T \end{bmatrix} \quad \begin{bmatrix} [\text{m}] \\ [^\circ] \end{bmatrix} \tag{6.45}$$

The attitude setpoint, $\vec{\eta}_d'$, was chosen because it commands each servo beyond their rotational limit. When using Proprotional Derivative controllers for both position and attitude control loops, Fig:6.25 shows that step response for attitude and positions collectively.



(a) Attitude step



(a) Position step

Figure 6.25: Step response without servo limits

Neither responses in Fig:6.25 are anything unexpected. The rotational positions for all eight servos are shown in Fig:6.26. Noting that each middle ring servo α_i for $i \in [1:4]$ settles to both above or below the $\pm\pi/2$ rotational limit.

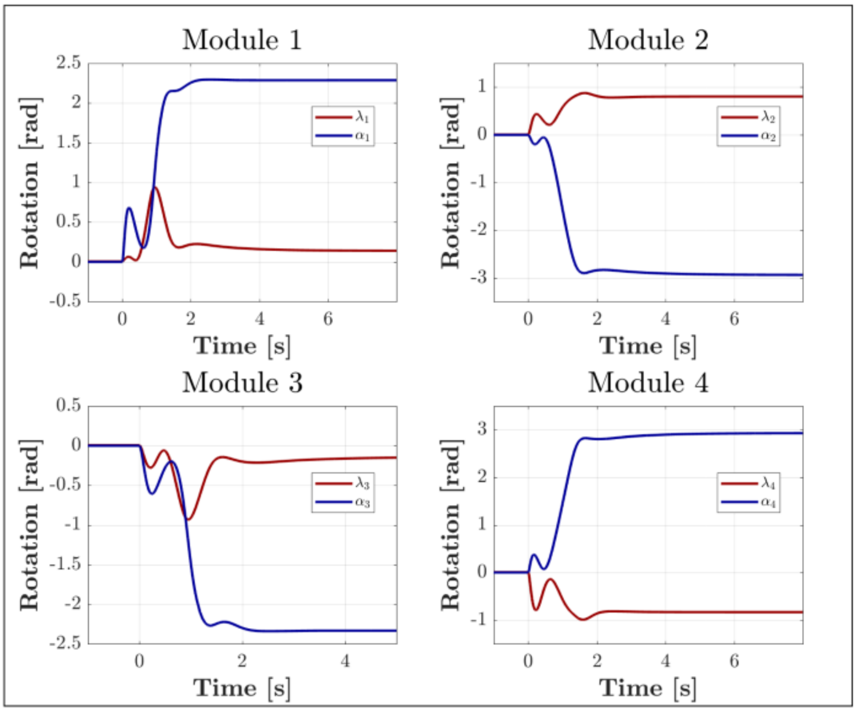
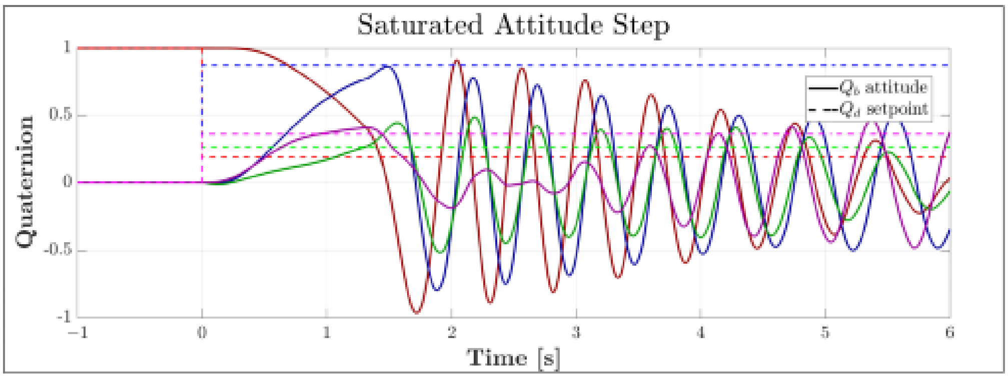
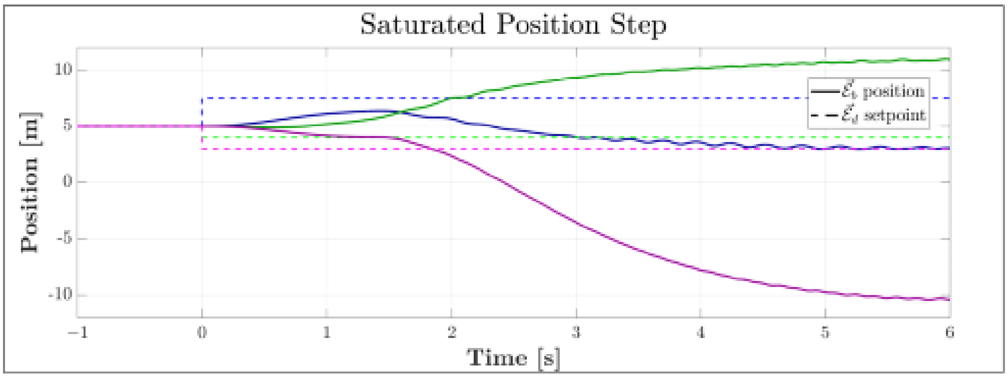


Figure 6.26: Servo inputs without limits

Then introducing that hard limit to the state step in Eq:6.45 produces a step response in Fig:6.27. The response obviously never reaches a settling point and destabilizes when the actuator saturation limits are applied.



(a) Attitude step



(b) Position step

Figure 6.27: Step response with servo limits

Fig:6.28 shows the limit cycles which the servos get stuck in when an attitude step which exceeds their hard limit is commanded.

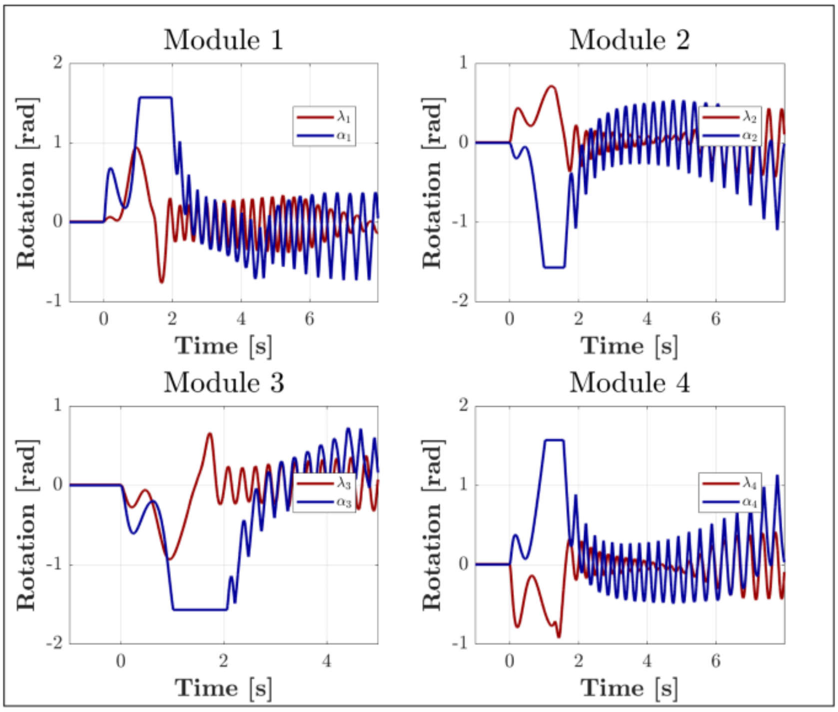


Figure 6.28: Servo inputs without limits

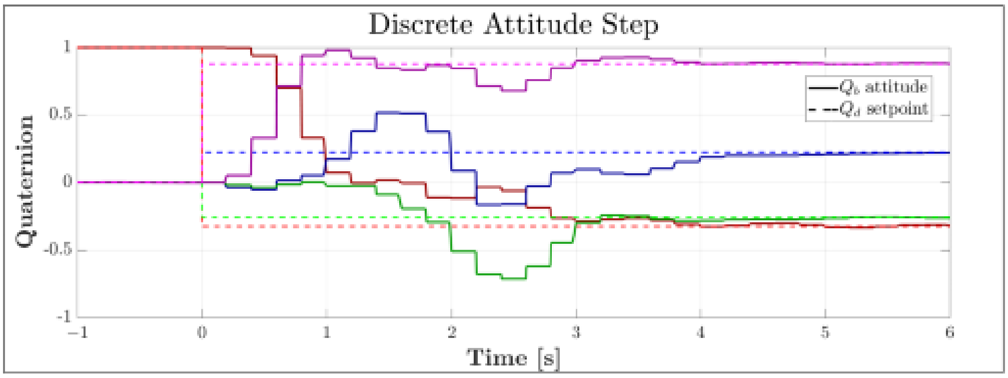
6.9 State Estimation

The final aspect of the control simulation to consider is the effect state estimation has on the controller's ability to track setpoints. It was proposed, in Ch:2, that a 9-axis inertial measurement unit would produce angular rates and inertial accelerations in the body frame. Then some form of filtration would fuse sensor measurements together to provide state-estimation only for the vehicle's attitude. Position displacement in the inertial frame cannot be approximated only using an IMU due to massive ammounts of integral drift. For that reason a motion capture VICON-like system, [111], was proposed to track the vehicles position under testing conditions.

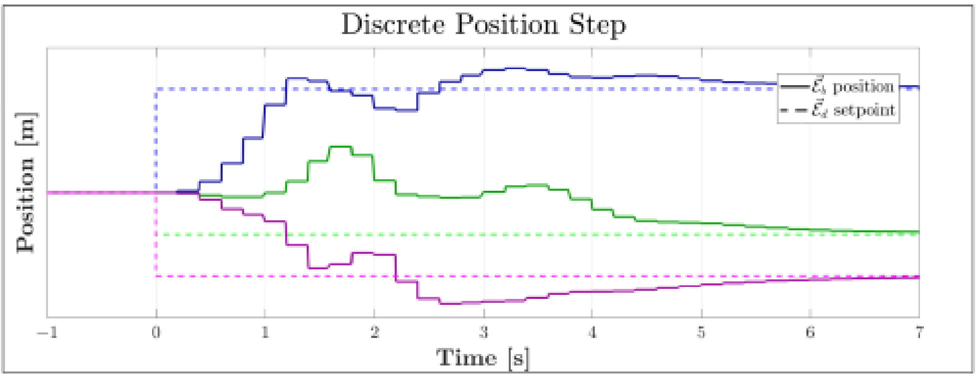
Such components of the embedded system loop could indeed be tested in simulation but, owing to the large amounts of noise a physical flight test would induce on those components, would not prove useful in evaluating the efficacy of the net proposed control system. Given the amount of vibrations and disturbances the prototype will undergo, simulations won't be able to accurately approximate such affects. Instead it was deemed pertinent to only apply discretization effects onto the state feedback elements within the control loop.

Quaternion position, angular velocity and translational velocity feedback terms were discretized and sampled at a rate of 70 Hz to emulate an IMU system based on the hardware proposed (Sec:2.4). In reality those signals would be processed by a Kalman filter and be subject to a relative degree of noise and integral drift. Then inertial position feedback was sampled at 50 Hz to emulate the proposed camera based state estimation from [111].

Both position and attitude control loops, testing with the basic Proportional Derivative controllers in both cases, were stable for the above proposed sampling rates. In fact the entire system was stable for sample rates as slow as 5 Hz, whose response for a typical attitude and position step (Eq:6.40) is plotted in Fig:6.29.



(a) Attitude step



(b) Position step

Figure 6.29: Discretized state steps

Chapter 7

Conclusions and Recommendations

The initial objective for the project was to design, simulate and physically test the prototype outlined in Sec:2.1. Modelling the responses which the multibody prototype had to the dynamic equations of motion, derived in Sec:3.4, proved to be dramatically more complex than initially anticipated. Non-zero inertial time rates (generalized in Eq:3.70d) introduced to the Lagrangian kinematics resulted in an entirely unique problem formulation. Each actuated motor module's response to the net vehicle's dynamics (Eq:3.75,3.86 and 3.105) required multiple revisions. Each step in the derivation was tested in simulation to ensure the mathematics applied were sound. The difficulties accompanying those derivations pushed back the project's time line significantly. As a result the final flight tests, which were initially envisioned were, unfortunately not completed. Physical implementation of the proposed control laws then remains open to further research.

With the above being considered; the produced dynamic model for the system would appear to be a significant contribution for this work. The uniqueness of the multibody structure made solving for the differential equations of motion a sizeable task. A consequence of the complex dynamics was an extremely high degree of stiffness in the system which adversely affected the simulation times. Alternatively, relative coordinates could have been implemented in lieu of the used Cartesian coordinates which describe the vehicle and its configuration. Moreover implicit Euler integration could have been applied to the simulation, both changes could potentially yield simulation improvements. The cost of such changes would be to construct a dedicated simulation program. Matlab's Simulink does not support implicit Euler integration for simulations.

The physical tests which corroporated aspects of the dynamic model, in Sec:3.4.2, would ideally be extended to physical flight tests. However considering the peculiarities of the system under consideration, the dynamics' verification (to a relative degree) was a significant result. The non-symmetrical inertias of each body within the entire multibody system was a consequence of the design process and the cost constraint applied to the prototype. In practice, if the rigid component of the frame (J_y from Eq:2.25d) was sufficiently greater than the inertial components of the actuated bodies, the complexities of the multibody interaction responses ($\vec{\tau}_b(\hat{u})$ from Eq:3.109b) would be diminished. The magnitude of those responses are actually inconsequential at slow or steady state operation (Fig:3.17), only spiking to significant values with large steps, Sec:3.4.2.

One of the original justifications for the project and its increased platform complexity was the improved actuator bandwidth accompanying the thrust vectoring. Asserting that pitching or rolling a thrust vector has a faster response than changing the propeller's rotational speed (see actuator transfer functions in Sec:2.4.1. The firmware changes made to the ESCs improved the brushless DC motor's transfer functions so much that the initial conjecture was rendered moot. Unfortunately the basic ESC's had exponential speed curves which were unsatisfactory for the control, so step tests comparing the before and after cases could not be performed.

The control solution(s) presented here each stabilize the trajectory in their own right. Results for controller attitude steps in Sec:6.3 and Sec:6.4 demonstrate the dramatic improvement exponential stability yields on a controller plant. None of the control laws proposed (attitude or position) were unable to track the applied trajectory, albeit a simple one. Each controller's optimization was a time error optimization which was shown in their respective performance. The controllers prioritized settling times and overshoot errors over aggression or input magnitude. Alternatively, the optimization apply penalty based on energy expenditure or induced torque response, prioritizing stability and smooth transitions over settling times. A particle swarm optimization in Sec:6.2.1 was chosen due to its simplicity and lack of an explicitly defined gradient function. Perhaps more complicated optimizers could produce stronger control responses in less time and the cost of computational complexity.

Certain constraints or assumptions were applied to the model in simulation. It was shown in Sec:6.8 that applying rotational limits to the actuation servo's dramatically hampered the over-all setpoint tracking ability of the control loop. Extending the actuators to accommodate for continuous rotation requires only a simple alteration to the mechanical design. The only significant assumption made on the plant's aerodynamics was neglecting to account for propeller's down wash being incident flow into other propeller modules. This would definitely have a sizeable impact on the thrust plant model, requiring a complicated fluid dynamics solution to approximate for such effects. Lastly, the decision to apply nonlinear state space control to the plant prevented the use of Model Predictive control. An MPC control law could potentially better compensate for or exploit the vehicle's non-linearities which were otherwise relegated to feedback compensation.

In conclusion, the non-zero state setpoint tracking goal was achieved by each of the control laws proposed. The control allocation rules applied did not have a notable effect on the plant's performance because of the structure applied in Sec:5.2. Finally there is a good reason why little to no work exists on reconfigurable aerospace vehicles, the dynamic complexity always out weighs the perceivable control improvements. Ironically those same dynamic complexities led catastrophic failures with earlier versions of Osprey [14], the inspiration for this project. Those complexities led to subsequent redesign of the Osprey's successor, the V-280 Valor, which has significantly smaller actuator inertias due to less moving parts (at the cost of higher maintenance fees)...

Appendix A

Expanded Equations

A.1 Standard Quadrotor Dynamics

Following the fundamental 6-DOF equations of motion for a rigid body derived in Sec:3.1.1, the common linearizations typically applied for generic "+" configured quadrotors are now presented. Reiterating those four differential equations, Eq:3.10, which describe a rigid body's motion (using rotation matrices and not quaternions):

$$\dot{\vec{\mathcal{E}}} = \mathbb{R}_b^I(-\eta)\vec{v}_b \qquad \in \mathcal{F}^I \tag{A.1a}$$

$$\dot{\vec{v}}_b = m_b^{-1}\big[-\vec{\omega}_b \times m_b\vec{v}_b + m_b\mathbb{R}_I^b(-\eta)\vec{G}_I + \vec{F}_{net}\big] \qquad \in \mathcal{F}^b \tag{A.1b}$$

$$\dot{\vec{\eta}} = \Phi(\eta)\vec{\omega}_b \qquad \in \mathcal{F}^{v2}, \mathcal{F}^{v1}, \mathcal{F}^I \tag{A.1c}$$

$$\dot{\vec{\omega}}_b = J_b^{-1}\big[-\vec{\omega}_b \times J_b\vec{\omega}_b + \vec{\tau}_{net}\big] \qquad \in \mathcal{F}^b \tag{A.1d}$$

With the Euler matrix, $\Phi(\eta)$, defined in Eq:2.12i. The net heave thrust produced by motors $i \in [1:4]$, bound perpendicularly to the $\hat{Z}_b$ axis, is given by:

$$\vec{T} = \sum_{i=1}^{4} F(\Omega_i) \cdot \hat{Z}_b \qquad \in \mathcal{F}^b \tag{A.2a}$$

The simplified relationship between the thrust scalar $T(\Omega_i)$ and the propeller's rotational speed Ω_i in [RPS] is approximately quadratic:

$$F(\Omega_i) \approx k_1\Omega_i^2 \tag{A.2b}$$

Similarly the aerodynamic torque opposing each rotating propeller, about the propellers $\hat{Z}_b$ axis, is:

$$Q(\Omega_i) \approx k_2\Omega_i^2 \tag{A.3}$$

Coefficients k_1 & k_2 are typically determined from physical thrust tests. The controllable pitch and roll torques, τ_ϕ & τ_θ about the $\hat{X}_b$ and $\hat{Y}_b$ axes respectively, are generated by opposing differential lift forces. Lastly the yaw torque, τ_ψ about the $\hat{Z}_b$ axis, is generated only a net response to the rotational aerodynamic propeller torques. The control torque inputs are then defined as:

$$\tau_\phi = L_{arm}\big(F(\Omega_1) - F(\Omega_3)\big) \cdot \hat{X}_b \tag{A.4a}$$

$$\tau_\theta = L_{arm}\big(F(\Omega_2) - F(\Omega_4)\big) \cdot \hat{Y}_b \tag{A.4b}$$

$$\tau_\psi = \sum_{i=1}^{4}(-1)^i Q(\Omega_i) \cdot \hat{Z}_b \tag{A.4c}$$

Then expanding the translational position and attitude state differentials, Eq:A.1b & Eq:A.1d, to their component forms (assuming the vehicle's inertial matrix J_b is diagonal):

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} = \begin{pmatrix} rv - qw \\ pw - ru \\ qu - pv \end{pmatrix} + \begin{pmatrix} -gsin(\theta) \\ gcos(\theta)sin(\phi) \\ gcos(\theta)cos(\phi) \end{pmatrix} + \frac{1}{m}\begin{pmatrix} 0 \\ 0 \\ T \end{pmatrix} \quad \in \mathcal{F}^b \tag{A.5a}$$

$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} \frac{J_{yy}-J_{zz}}{J_{xx}} qr \\ \frac{J_{zz}-J_{xx}}{J_{yy}} pr \\ \frac{J_{xx}-J_{yy}}{J_{zz}} pq \end{pmatrix} + J_b^{-1} \begin{pmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{pmatrix} \quad \in \mathcal{F}^b \tag{A.5b}$$

Considering the size of a typical angular rate, $\vec{\omega}_b \approx \vec{0}$; the gyroscopic and Coriolis effects on the body (namely both cross product terms) are sufficiently small enough to be regarded as negligible. Assuming too that the body has a (*roughly*) diagonal inertial matrix. Then the following holds true around the origin when $\vec{\omega}_b \approx \vec{0}$:

$$\begin{pmatrix} rv - qw \\ pw - ru \\ qu - pv \end{pmatrix} \approx \vec{0} \quad \text{and} \quad = \begin{pmatrix} \frac{J_{yy}-J_{zz}}{J_{xx}} qr \\ \frac{J_{zz}-J_{xx}}{J_{yy}} pr \\ \frac{J_{xx}-J_{yy}}{J_{zz}} pq \end{pmatrix} \approx \vec{0} \tag{A.6}$$

As a result, state differentials in Eq:A.5 can then reduce to the following:

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} = \begin{pmatrix} -gsin(\theta) \\ gcos(\theta)sin(\phi) \\ gcos(\theta)cos(\phi) \end{pmatrix} + \frac{1}{m}\begin{pmatrix} 0 \\ 0 \\ T \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} \frac{1}{\mathbb{I}_x}\tau_\phi \\ \frac{1}{\mathbb{I}_y}\tau_\theta \\ \frac{1}{\mathbb{I}_z}\tau_\psi \end{pmatrix} \tag{A.7}$$

Similarly, at an attitude near to the origin and at hovering conditions the following simplification applies to the Euler matrix $\Phi(\eta)$:

$$\Phi(\eta) \approx \vec{1} \quad \text{for} \quad \eta \approx \vec{0} \tag{A.8}$$

And so from Eq:A.1c the body's *Euler rates* are approximately equivalent to its angular velocity:

$$\begin{pmatrix} \dot{p} & \dot{q} & \dot{r} \end{pmatrix}^T \approx \begin{pmatrix} \ddot{\phi} & \ddot{\theta} & \ddot{\psi} \end{pmatrix}^T \Rightarrow \dot{\eta} \approx \omega_b \tag{A.9}$$

The above Eq:A.9 is not an insignificant result. The difficulty with Euler angle parameterization for body attitude is that each Euler angle is defined with respect to a sequential reference frame, Eq:2.12. As such, the state equations for Eq:A.5 then reduce to the following six SISO controllable plants when the vehicle's angular velocity is small:

$$\ddot{x} = (-cos(\phi)sin(\theta)cos(\psi) - sin(\phi)sin(\psi)\frac{1}{m}T \tag{A.10a}$$

$$\ddot{y} = (-cos(\phi)sin(\theta)sin(\psi) + sin(\phi)cos(\psi))\frac{1}{m}T \tag{A.10b}$$

$$\ddot{z} = g - (cos(\phi)cos(\theta)\frac{1}{m}T \tag{A.10c}$$

$$\ddot{\phi} = \frac{1}{J_{xx}}\tau_\phi \tag{A.10d}$$

$$\ddot{\theta} = \frac{1}{J_{yy}}\tau_\theta \tag{A.10e}$$

$$\ddot{\psi} = \frac{1}{J_{zz}}\tau_\psi \tag{A.10f}$$

Typically, the simplified Eq:A.10 is abstracted to an "augmented pilot control system". In such a case the controllable inputs are abstracted to T, $\ddot{\phi}$, $\ddot{\theta}$, $\ddot{\psi}$. Wherein the pilot dictates the attitude torques and net heave thrust for the quadrotor, mostly with various flavours of PD control for each channel.

A.2 Blade-Element Momentum Expansion

Expanding on the Blade-Element Momentum equations from Eq:3.27 and Eq:3.31a. Reiterating the integral equations, they are:

$$dT = \rho 4\pi r v_\infty (1 + a)a.dr \tag{A.11a}$$

$$dT = \frac{1}{2} a_L b c \rho (\Omega r)^2 \big(\theta - \frac{v_\infty + v_i}{\Omega r}\big).dr \tag{A.11b}$$

Both Eq:A.11a-A.11b are integrals taken accross the length of the propeller blade. 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:

$$\lambda^2 + \left(\frac{\sigma a_L}{8} + \lambda_c\right)\lambda - \frac{\sigma a_L}{8}\theta\frac{r}{R} = 0 \tag{A.12}$$

Where λ_c is the nominal free-stream inflow ratio when $v_i = 0$. Another term, σ, is defined as the propeller solidity and is given by:

$$\sigma = \frac{bc}{\pi R} \tag{A.13}$$

Then, solving Eq:A.12 for λ:

$$\lambda = \sqrt{\left(\frac{\sigma a_L}{16} - \frac{\lambda_c}{2}\right)^2 + \frac{\sigma a_L}{8}\theta\frac{r}{R}} - \left(\frac{\sigma a_L}{16} - \frac{\lambda_c}{2}\right) \tag{A.14}$$

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:

$$\lambda = \frac{v_i}{\Omega r} = \sqrt{\frac{C_{T0}}{2}} \tag{A.15}$$

Then substituting λ back into Eq:3.31a 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 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 [126], where the transformation to quaternions is based on Shoemake's [124] definition. Each quaternion can be constructed from sequenced Euler angles, as in Eq:3.51. Then, solving for each euler angle using simultaneous solutions and inverse trigonometry:

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} arctan2\big(2(q_0 q_x + q_y q_z),\ 1 - 2({q_x}^2 + {q_y}^2)\big) \\ arcsin\big(2(q_0 q_y - q_x q_z)\big) \\ arctan2\big(2(q_0 q_z + q_x q_y),\ 1 - 2({q_y}^2 + {q_z}^2)\big) \end{bmatrix} \tag{A.16}$$

A.4 Solving for Actuator Positions from Thrust Vectors

Expanding on Eq:5.16 from Sec:5.2 to solve for discrete propeller and servo positions $[\Omega_i\ \lambda_i\ \alpha_i]^T$ from allocated thrust vectors $\vec{T}_{1\to4}$, the following equations solve for actuator positions for *each* motor module $i \in [1:4]$. For motor module 1 when $\sigma_1 = 0°$:

$$\begin{bmatrix} \Omega_1 \\ \lambda_1 \\ \alpha_1 \end{bmatrix} = R^\dagger(\mathbf{x}, \vec{T}_1, t) \triangleq \begin{bmatrix} \left(\sqrt{{T_x}^2 + {T_y}^2 + {T_z}^2}/C_T(J)\rho D^4\right)^{\frac{1}{2}} \\ atan2(-{T_y}^2,\ ||\vec{T}_1||\sqrt{||\vec{T}_1||^2 - {T_y}^2}) \\ atan2(T_x,\ T_z||\vec{T}_1||) \end{bmatrix} \tag{A.17a}$$

$$\begin{bmatrix}\Omega_2\\ \lambda_2\\ \alpha_2\end{bmatrix} = R^{\dagger}(\mathbf{x}, \vec{T}_2, t) \triangleq \begin{bmatrix}\left(\sqrt{{T_x}^2 + {T_y}^2 + {T_z}^2}/C_T(J)\rho D^4\right)^{\frac{1}{2}}\\ atan2({T_x}^2,\ ||\vec{T}_2||\sqrt{||\vec{T}_2||^2 - {T_x}^2})\\ atan2(T_y,\ T_z||\vec{T}_2||)\end{bmatrix} \tag{A.17b}$$

$$\begin{bmatrix}\Omega_3\\ \lambda_3\\ \alpha_3\end{bmatrix} = R^{\dagger}(\mathbf{x}, \vec{T}_3, t) \triangleq \begin{bmatrix}\left(\sqrt{{T_x}^2 + {T_y}^2 + {T_z}^2}/C_T(J)\rho D^4\right)^{\frac{1}{2}}\\ atan2({T_y}^2,\ ||\vec{T}_2||\sqrt{||\vec{T}_2||^2 - {T_y}^2})\\ atan2(-T_x,\ T_z||\vec{T}_2||)\end{bmatrix} \tag{A.17c}$$

$$\begin{bmatrix}\Omega_4\\ \lambda_4\\ \alpha_4\end{bmatrix} = R^{\dagger}(\mathbf{x}, \vec{T}_4, t) \triangleq \begin{bmatrix}\left(\sqrt{{T_x}^2 + {T_y}^2 + {T_z}^2}/C_T(J)\rho D^4\right)^{\frac{1}{2}}\\ atan2(-{T_x}^2,\ ||\vec{T}_4||\sqrt{||\vec{T}_4||^2 - {T_x}^2})\\ atan2(-T_y,\ T_z||\vec{T}_4||)\end{bmatrix} \tag{A.17d}$$

Appendix B

Design Bill of Materials

B.1 Parts List

Part Name	No. Used	Unit Weight[g]
Electronics		
SPRacing F3 Deluxe Flight Controller	1	8
OrangeRx 615X 2.4 GHz 6CH Receiver	1	9.8
Signal Converter SBUS-PPM-PWM	1	5.0
STLink-V2 Debugger	1	3
RotorStar Super Mini S-BEC 10A	1	30
128x96" OLED Display	1	7
XBee-Pro S1	2	4
HobbyWing XRotor 20A Opto ESC	4	15
OrangeRX RPM Sensor	4	2
HobbyKing Multi-Rotor Power Distribution Board	1	49
Motors		
Corona DS-339MG	8	32
Cobra 2208 2000KV Brushlesss DC	4	44.2
Frame Components		
APM Flight Controller Damping Platform	1	7
HobbyKing SK450 Replacement Arm (2 pcs)	2	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

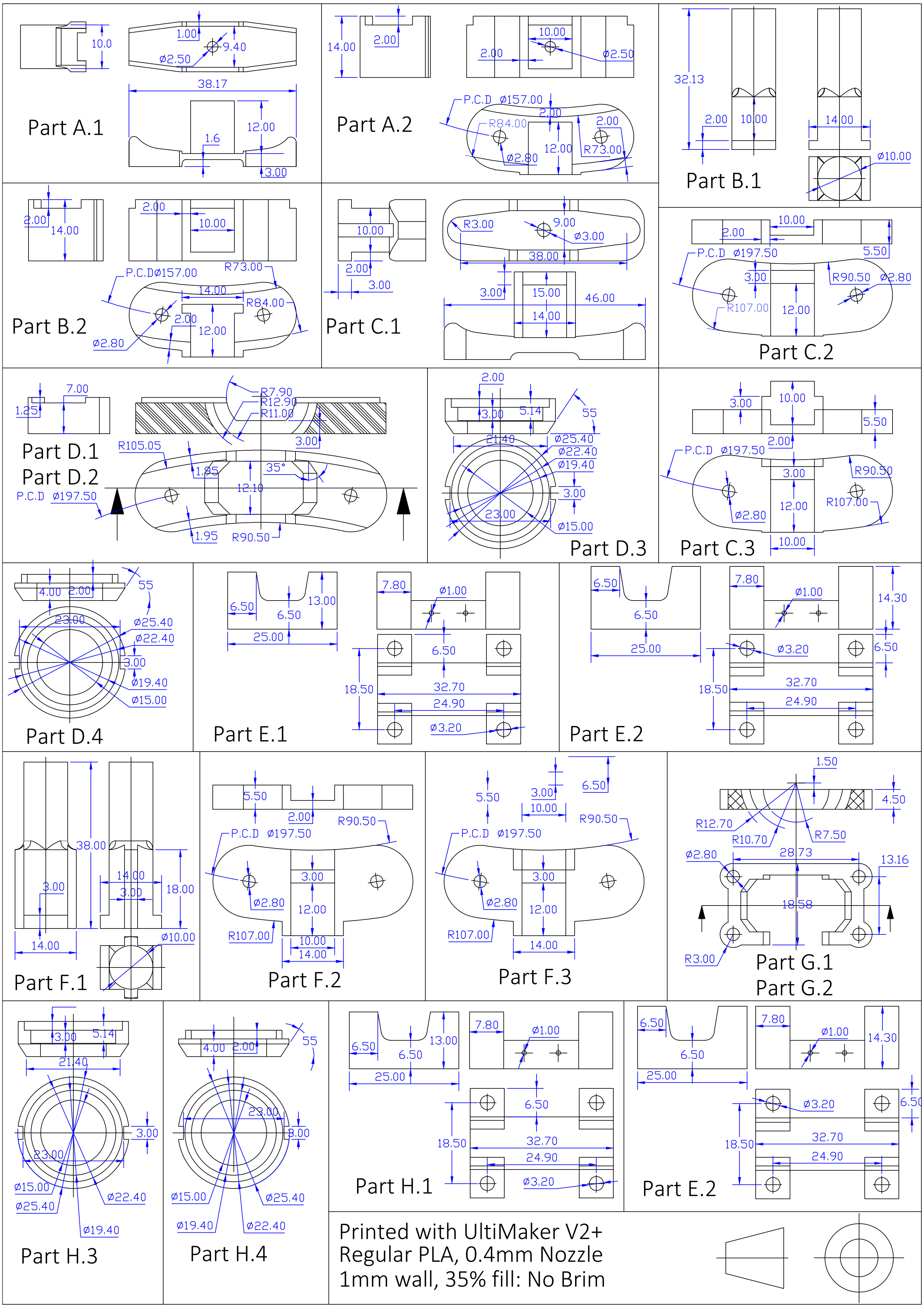


Table B.2: 3D Printed Parts

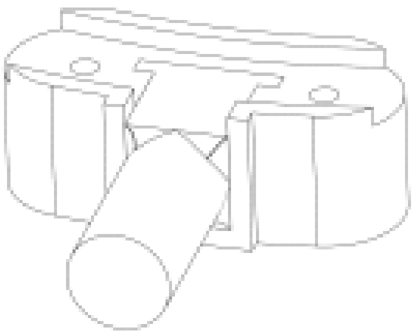
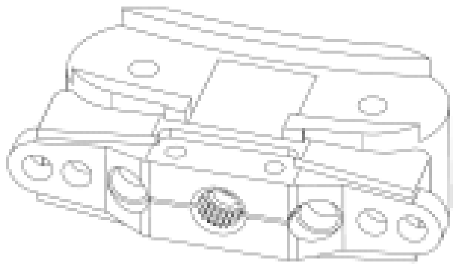
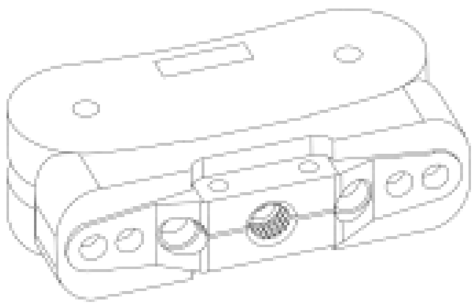
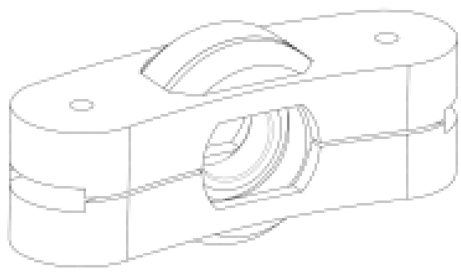
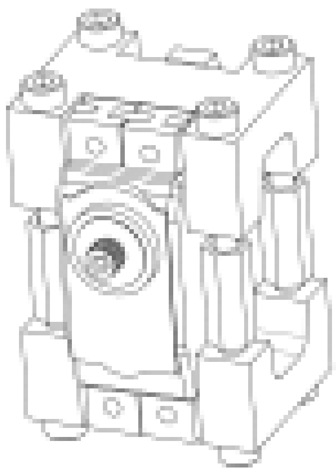
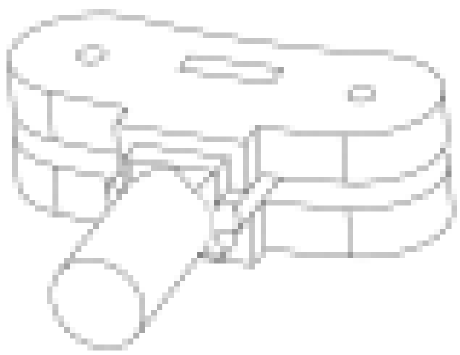
Bracket Assemblies 2	
 **Figure B.1:** Bearing Bracket Inner Ring Assembly Parts: A.1, A.2	 **Figure B.2:** Servo Bracket Inner Ring Assembly Parts: B.1, B.2, M3 Servo Horn
 **Figure B.3:** Servo Bracket Middle Ring Assembly Parts: C.1, C.2, C.3, M3 Servo Horn	 **Figure B.4:** Bearing Holder Middle Ring 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

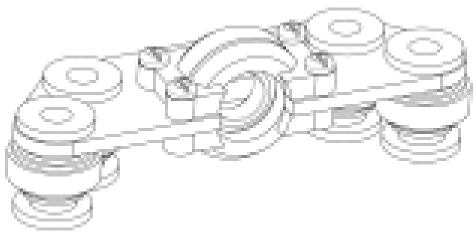
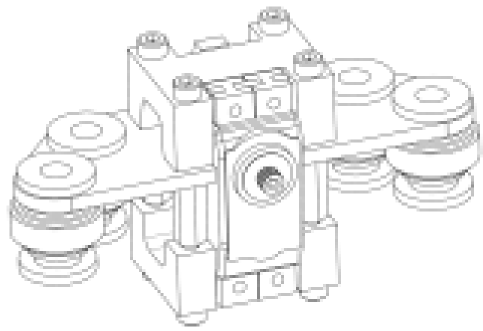
Bracket Assemblies 2	
 **Figure B.7:** Bearing Holder Damping Assembly Parts: G.1, G.2, G.3, G.4, 18-10 Bearing, 80g Damping Balls, Bearing Holder Damping Bracket	 **Figure B.8:** Servo Mount Damping Assembly Parts: H.1, H.2, Corona Servo & Fasteners, 80g Damping Balls, Servo Mount Damping Bracket

Table B.4: Damping Assemblies

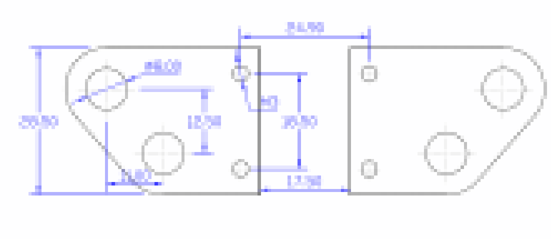
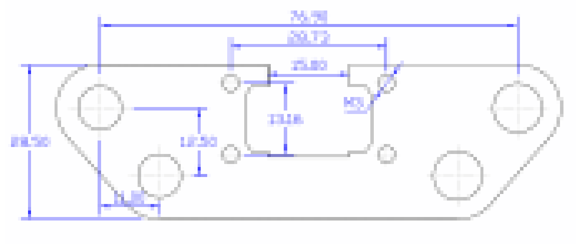
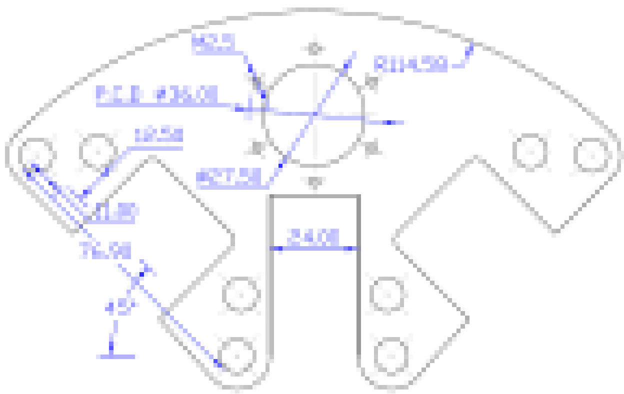
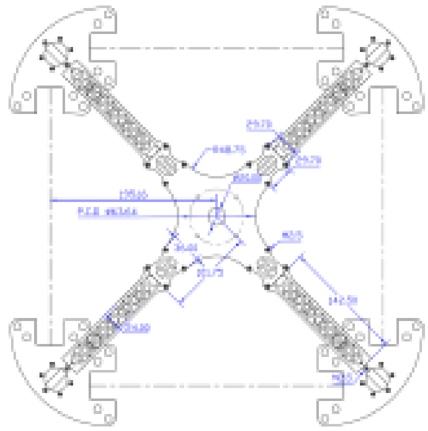
Laser Cut Brackets	
 **Figure B.9:** Servo Mount Damping Bracket	 **Figure B.10:** Bearing Holder Damping Bracket
 **Figure B.11:** Arm Mount Damping Bracket	 **Figure B.12:** Frame Brackets

Table B.5: Laser Cut Damping Brackets

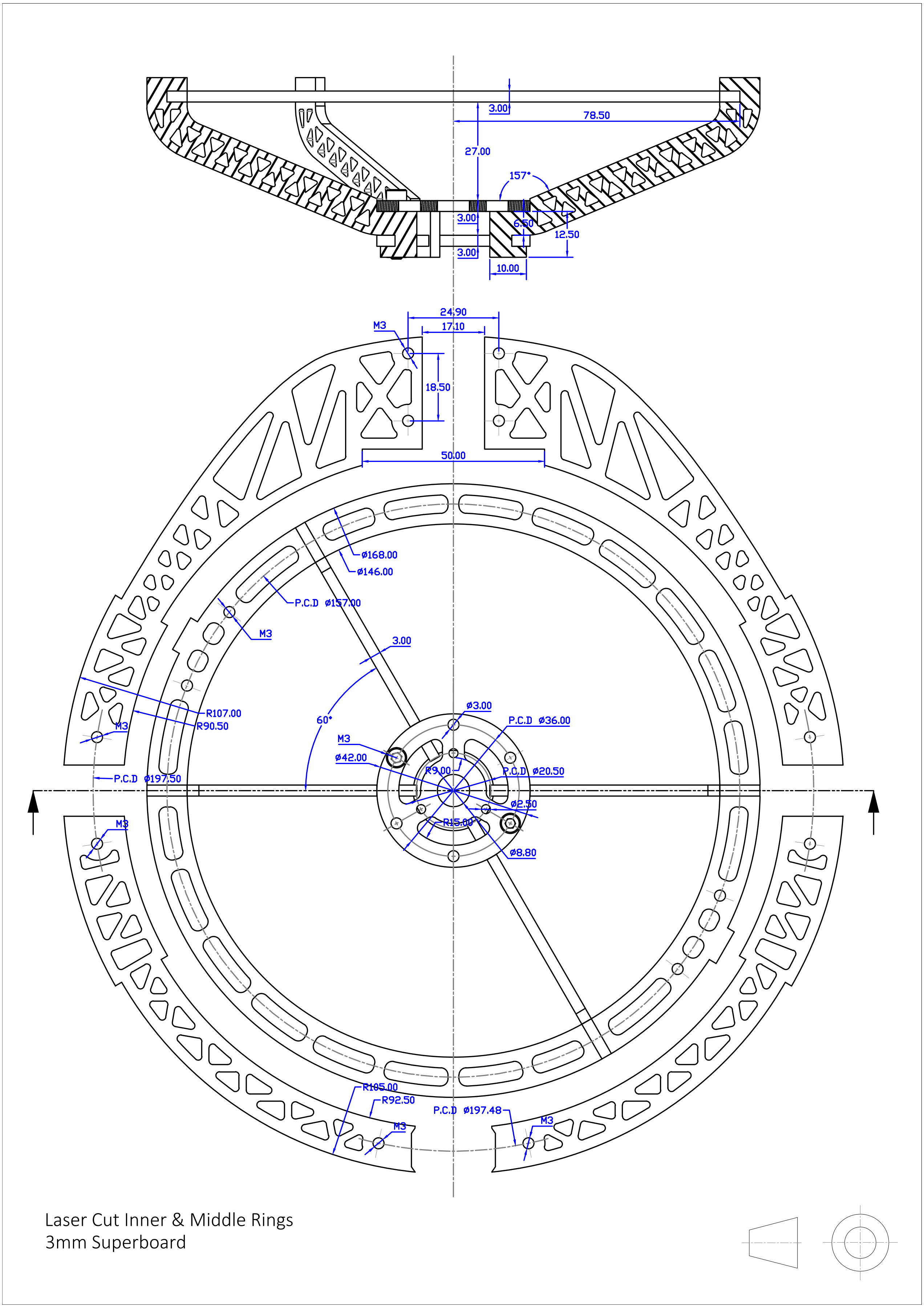


Table B.6: Laser Cut Parts

B.2 F3 Deluxe Schematic Diagram

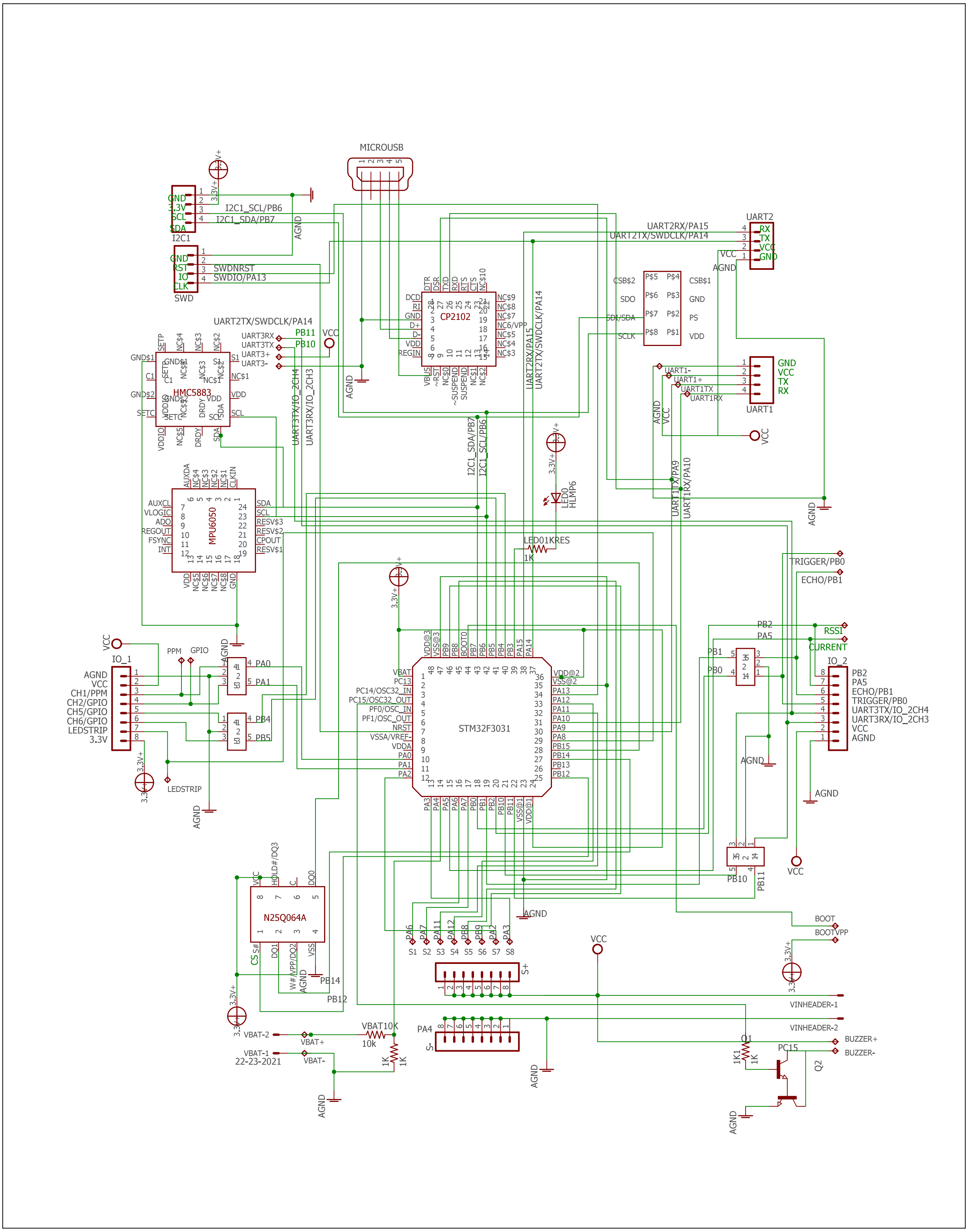


Figure B.13: F3 Deluxe Flight Controller Hardware Schematic

B.3 Strain Gauge Amplification

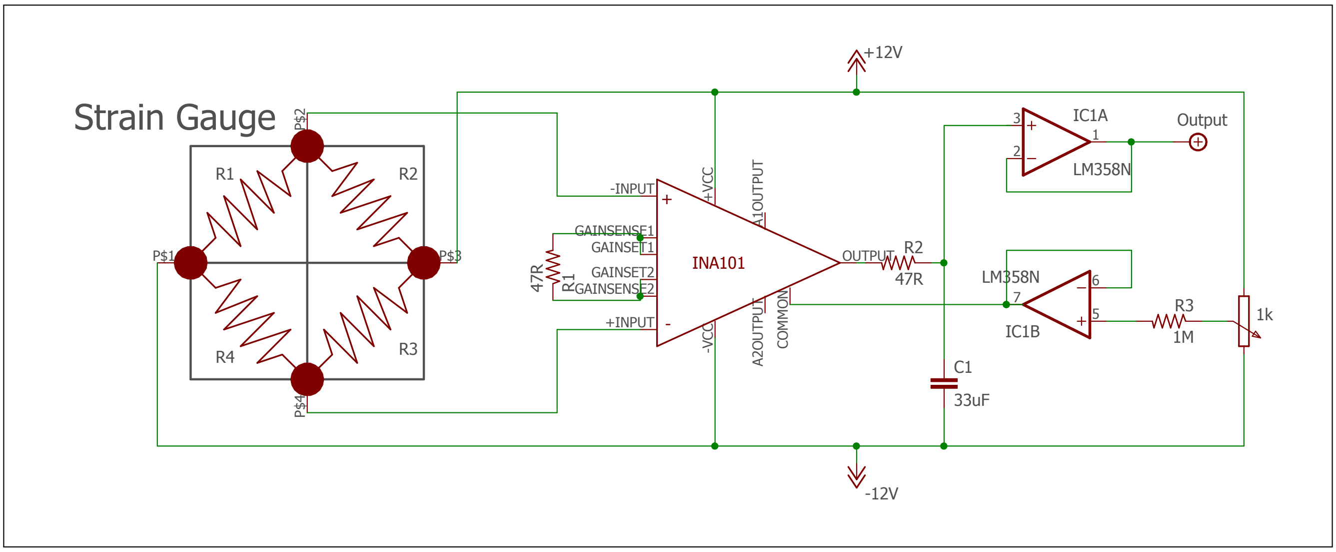
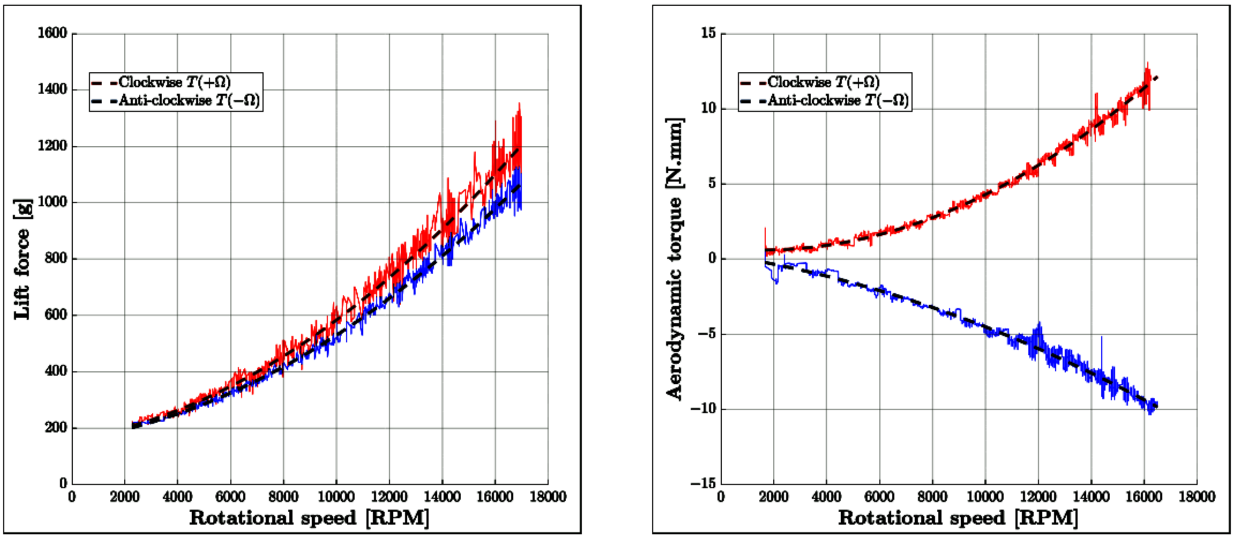


Figure B.14: Strain gauge full bridge amplifier

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

Thrust tests in Fig:C.1a causes lateral deflection of the strain gauge thrust test rig illustrated in Fig:3.6a. The deflection is in the direction of the propeller's rotational sense, as a result of the torque applied to the propeller. Clockwise and counter-clockwise tests where summed together and averaged to produce the thrust tests plotted in Fig:3.6.

Torque tests in Fig:C.1b shows thrust deflection in the rotational torque test rig in Fig:3.7a. Upward thrust still resulted in some small deflection in the resultant measurements so opposing clockwise and counter-clockwise results were subtracted and averaged out to produce the torque tests plotted in Fig:3.7.

C.2 Cobra CM2208-200KV Thrust Data

Cobra CM-2208/20 Motor Propeller Data

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

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

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

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

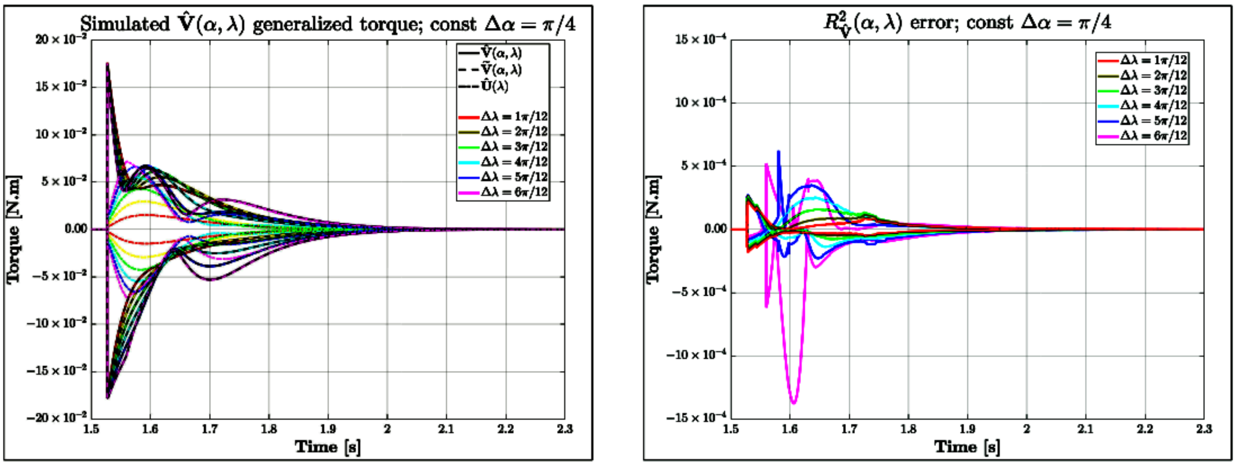
Prop Manf.	Prop Size	Li-Po Cells	Input Voltage	Motor Amps	Input Watts	Prop RPM	Pitch Speed in MPH	Thrust Grams	Thrust Ounces	Thrust Eff. Grams/W
APC	5.25x4.75-E	4	14.8	17.29	255.9	20,560	92.5	603	21.27	2.36
APC	5.5x4.5-E	4	14.8	17.87	264.5	20,436	87.1	635	22.40	2.40
APC	6x4-E	4	14.8	20.15	298.2	19,829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

Figure C.2: Official Test Results for Cobra Motors

C.3 Combined Simulated Torque Responses

The process in Sec:3.4.2 applied simulation tests to the generalized torque responses derived in Sec:3.4.1. Previous simulations only considered separate singular perturbations in either λ_i or α_i rotational positions alone. The generalized torque response $\vec{\mathbf{V}}(\alpha_i, \lambda_i)$, from Eq:3.85g, acts as a response to *net motor module* rotations of both inner and middle ring servos λ_i and α_i respectively.

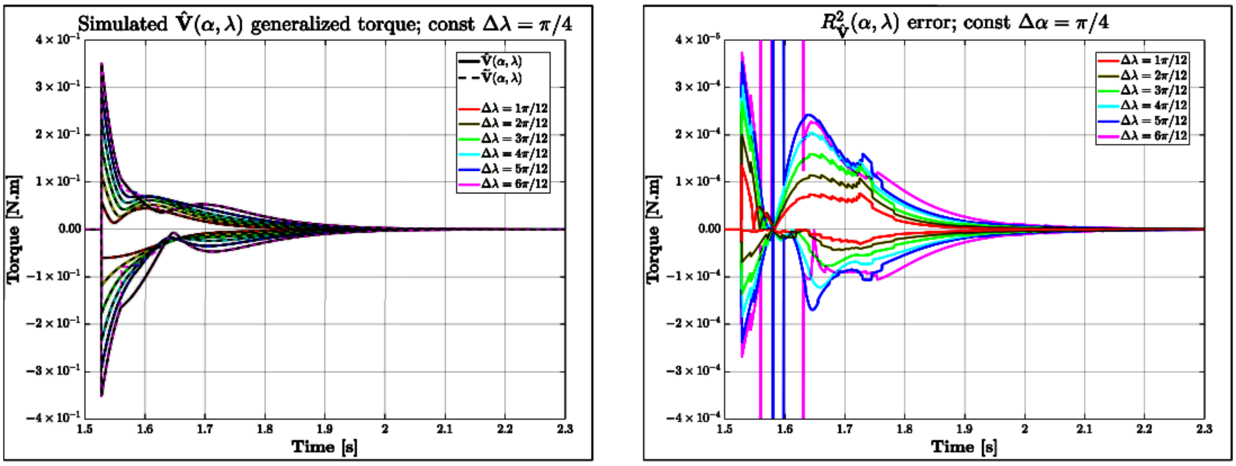
The plot in Fig:C.3 shows varying steps for $\Delta\lambda_i$ with a constant $\Delta\alpha_i = \pi/4$ step. The error between an estimated value $\hat{V}(\alpha_i, \lambda)$, with a linearized rotation partial derivative, and the true $\widetilde{V}(\alpha, \lambda)$ is shown in Fig:C.3b. That error is mostly of the order $\times 10^{-4}$ [N.m]; whilst both $\hat{V}(\alpha_i, \lambda_i)$ and $\widetilde{V}(\alpha_i, \lambda_i)$ are in the order of $\times 10^{-1}$ [N.m].



(a) Torque response $\vec{\mathbf{V}}(\alpha_i, \lambda_i)$ to steps in $\Delta\lambda_i$ only **(b)** Error for $\vec{\mathbf{V}}(\alpha_i, \lambda_i)$ for steps in $\Delta\lambda_i$ only

Figure C.3: Step changes in $\Delta\lambda_i$ with constant $\Delta\alpha_i = \pi/4$

Similarly, Fig:C.4a shows the same tests run for varying step sizes of $\Delta\alpha_i$ with a constant step size for $\Delta\lambda_i = \pi/4$. Again, the plot Fig:C.4b shows the error which, on average, is in the order of $\times 10^{-4}$ [N.m]. The error between a simplified $\hat{V}(\alpha_i, \lambda_i)$ and the true $\widetilde{V}(\alpha_i, \lambda_i)$ only becomes significant as the step size $\Delta\alpha_i$ tends to $\pi/2$.



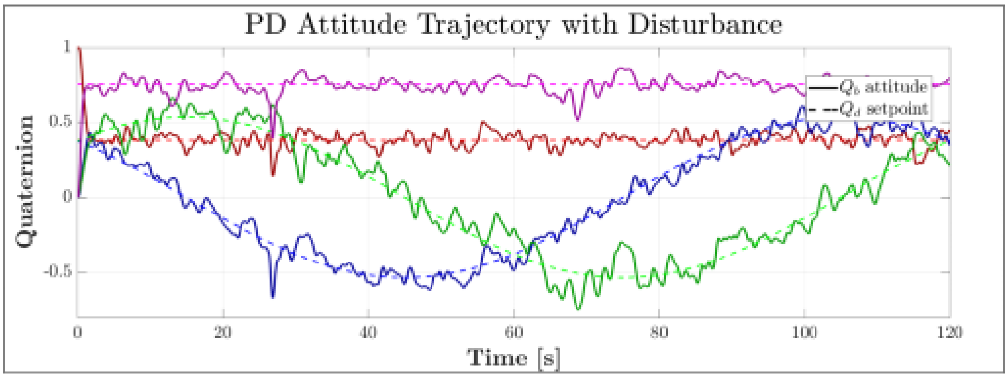
(a) Torque response $\vec{\mathbf{V}}(\alpha_i, \lambda_i)$ to steps in $\Delta\alpha_i$ only **(b)** Error for $\vec{\mathbf{V}}(\alpha_i, \lambda_i)$ for steps in $\Delta\alpha_i$ only

Figure C.4: Step changes in $\Delta\alpha_i$ with constant $\Delta\lambda_i = \pi/4$

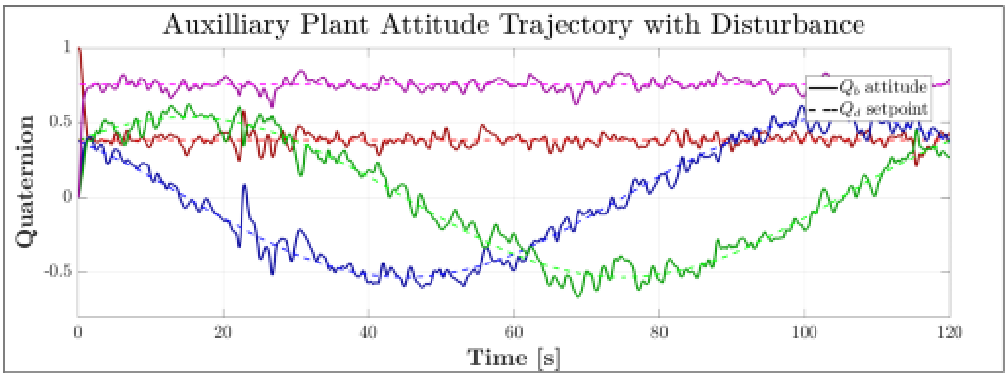
It is interesting to note that positive and negative step directions are not symmetrical in their responses for Fig:C.3a and Fig:C.4a. This is as a result of the gyroscopic cross product in the calculations for $\hat{V}(\alpha_i, \lambda_i)$. Both tests shown in Fig:C.3 and Fig:C.4 further corroborate the model proposed previously in Sec:3.4.1.

C.4 Controller Disturbance Rejection

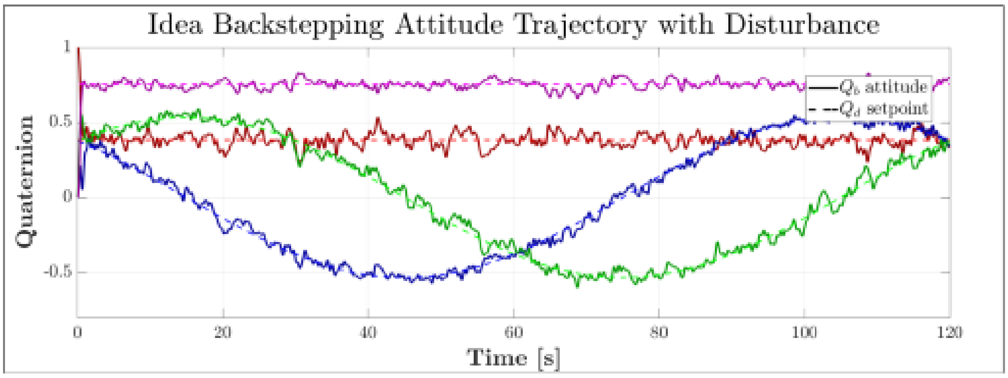
C.4.1 Attitude Controllers



(a) Diagonal Proportional Derivative Controller



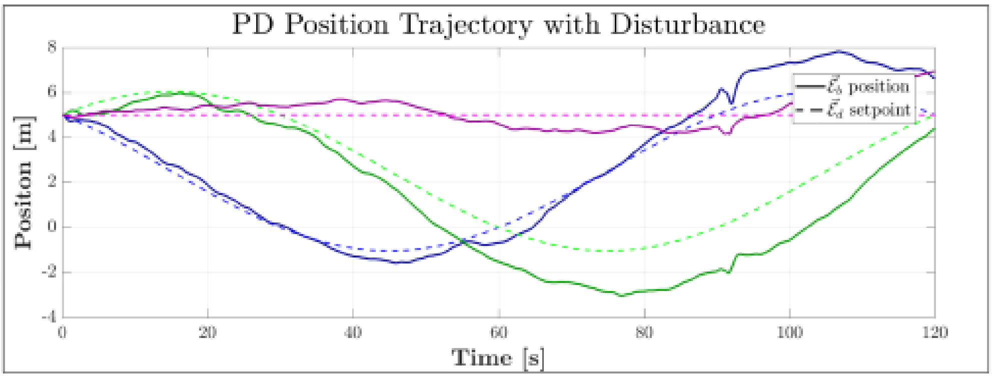
(b) Auxilliary Plant Controller



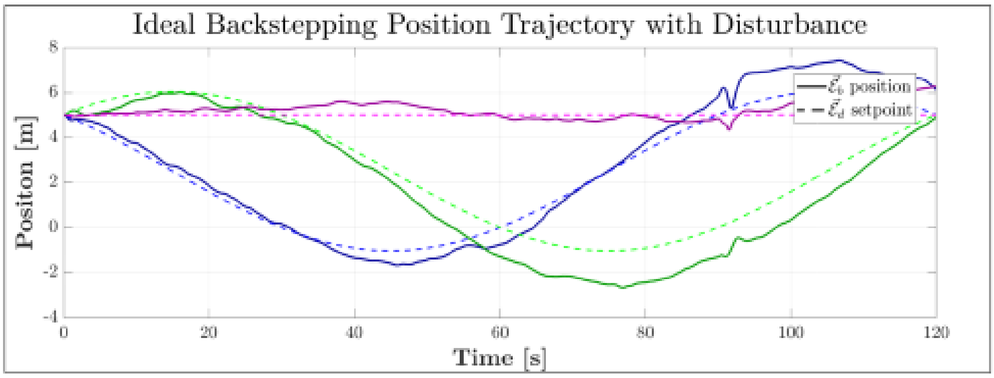
(c) Ideal Backstepping Controller

Figure C.5: Disturbances on Attitude Controllers

C.4.2 Position Controllers



(a) Proportional Derivative Controller



(b) Ideal Backstepping Controller

Figure C.6: Disturbances on Position Controllers

Bibliography

[1] Parrot AG. Products: Ar drone 2.0. `https://www.parrot.com/fr/drones/parrot-ardrone-20-elite-edition#technicals`, 2016. [Accessed:2017-01-16].

[2] Y. Al-Rihani. *Development of a Dual Axis Tilt Rotorcraft Uav: Design, Prototyping and Control*. Cranfield University: School of Engineering, 2012. Masters dissertation.

[3] N. Amiri, A. Ramirez-Serrano, and Davies R. Modelling of opposed lateral and longitudinal tilting dual-fan unmanned aerial vehicle. In *International Federation of Automatic Control*, volume N/A, pages 2054–2059, Milano, Italy, 8 2011. IEEE.

[4] APMCopter. Arducopter official site. `http://www.arducopter.co.uk/`, 2016. [Accessed:2016-5-01].

[5] E. Balasubramanian and R. Vasantharaj. Dynamic modelling and control of quadrotor. *International Journal of Engineering and Technology*, 5(1):63–39, 2 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*, volume N/A, pages 1–10, Wellington, New Zealand, 12 2012. ACRA.

[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] I.Y Bar-itzhack and Y. Oshman. Attitude determination from vector observations: Quaternion estimation. In *Transactions on Aerospace and Electronic Systems*, volume 1, pages 128–137, N/A, 1 1985. IEEE.

[10] 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.

[11] M. Basri, A. Husain, and K. Danapalasingam. Intelligent adaptive backstepping control for mimo uncertain non-linear quadrotor helicopter systems. *Transactions of the Institute of Measurement Control*, 4(37):345–361, 9 2017.

[12] Y.E Bayiz and R. Babuska. Nonlinear disturbance compensation and reference tracking via reinforcement learning with fuzzy approximators. *The International Federation of Automatic Control*, pages 5393–5398, 2014.

[13] R. Beard. Quadrotor dynamics and control. Report, Brigham Young University, 10 2008.

[14] R. Berler. Saving the pentagon's killer chopper-plane. Technical Report N/A, Wired Magazine, N/A, 7 2005.

[15] BetaFlight. Betaflight fc4 repo. Forked from the CleanFlight repo,`https://github.com/betaflight/betaflight`, 2016. [Accessed:2016-9-17].

[16] BLHeli. Blheli master branch (silabs). `https://github.com/bitdump/BLHeli/tree/master/SiLabs`, 2016. [Accessed:2016-11-05].

[17] Charles Blouin and Eric Lanteigne. Pitch control on an oblique active tilting bi-rotor. In *International Conference on Unmanned Aircraft Systems*, volume N/A, pages 791–799, Florida, USA, 5 2014. IEEE.

[18] R. Bodrany, W. Steyn, and M. Crawford. In-orbit estimation of the inertia matrix and thruster parameters of uosat-12. *Control Engineering Practice*, 12(N/A):87–98, 12 2004.

[19] Edward Boje. Lyapunov stability analysis. Report, University of Kwazulu-Nata: School of Electrical, Electronic and Computer Engineering, 1 2005. Course notes for: Control Systems ENEL4CN.

[20] H. Bolandi, M. Rezaei, R. Mohsenipour, H. Nemati, and S. Smailzadeh. Attitude control of a quadcopter with optimized pid. *Intelligent Control and Automation*, N/A(4):335–342, 8 2013.

[21] R.M. Botez, I. Boustani, and N. Vayani. Optimal control laws for gust load alleviation. In *Proceedings of the CASI Annual Conference*, volume 46, pages 649–655, 5 199.

[22] S. Bouabdallah, A. Noth, and R. Siegward. Pid vs lq control technqiues applied to an indoor micro quadrotor. In *International Conference on Intelligent Robots and Systems*, volume N/A, pages 2451–2456, Sendal, Japan, 9 2004. IEEE.

[23] S. Bouabdallah and R. Siegward. Backstepping and sliding mode technqiues applied to an indoor micro quadrotor. *IEEE International Conference on Intelligent Robots and Systems*, pages 2247–2252, 4 2005.

[24] S. Bouabdallah and R. Siegwart. Full control of a quadrotor. In *International Conference on Intelligent Robots and Systems*, volume N/A, pages 153–158, California, USA, 11 2007. AIAA.

[25] A.R.S Bramwell, D. Balmford, and G. Done. *Bramwell's Helicopter Dynamics*, chapter 1-3, pages 1–144. Elsevier Ltd, 2 edition, 1999.

[26] 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].

[27] J. Brandt and M. Selig. Propeller performance data at low reynolds numbers. *Aerospaec Sciences Meeting*, 49:1–18, 1 2011.

[28] D. Brescianini, M. Hehn, and R D'Andrea. Nonlinear quadrocopter attitude control. Technical report, Institute for Dynamic Systems and Control, ETH Zurich, 10 2013.

[29] Z. Cai and M. S. de Queiroz. Asymptotic adaptive regulation of parametric strict-feedback systems with additive distrubance. In *American Control Conference*, volume 5, pages 3707–3712, Oregon, USA, 6 2005. IEEE.

[30] J. Chen, A. Behal, and D. Dawson. Adaptive output feedback control for a class of mimo nonlinear systems. In *American Control Conference*, volume N/A, pages 5300–5306, Minnesota, USA, 6 2006. IEEE.

[31] A.B. Chowdhury, A. Kulhare, and G. Raina. A generalized control method for tilt-rotor uav stabilization. In *International Conference on Cyber Technology in Automation, Control and Intelligent Systems*, volume N/A, pages 309–314, Bangkok, Thailand, 5 2012. IEEE.

[32] CleanFlight. Cleanflight repo. `https://github.com/cleanflight/cleanflight`, 2016. [Accessed:2016-11-13].

[33] Dominic Clifton. Spracing f3 deluxe flight controller. `http://seriouslypro.com/spracingf3`, 2015. [Accessed:2016-10-04].

[34] Innov8tive Designs. Cobra cm2208/2000 motors. `http://innov8tivedesigns.com/cobra-cm-2208-20-multirotor-motor-kv-2000`, 2016. [Accessed:2016-06-10].

[35] C. Diao, B. Xian, Q. Yin, W. Zeng, H. Li, and Y. Yang. A nonlinear adaptive control approach for quadrotor uavs. In *Asian Control Conference Proceedings*, volume 8, pages 223–228, Kaohsiung, Taiwan, 8 2011. ACC.

[36] DJI Drones. Dji inspire one. `http://www.dji.com/product/inspire-1`, 2016. [Accessed:2016-07-10].

[37] DJI Drones. Products: Dji phantom. `http://www.dji.com/products/phantom`, 2016. [Accessed:2016-06-12].

[38] H. Du, G. Fan, and G. Yi. Disturbance compensated adaptive backstepping control of an unmanned seaplant. In *Proceeding of IEEE, International Conference on Roboics and Biomimetics*, pages 1725–1730, Bali, Indonesia, 12 2014. IEEE.

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

[40] 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.

[41] E. Fresk and G. Nikolakopoulos. Full quaternion based attitude control for a quadrotor. In *European Control Conference*, volume N/A, pages 3864–3869, Zurich, Switzerland, 7 2013. IEEE.

[42] D. Garanin. Rotational motion of rigid bodys. Report, City University of New York, Department of Engineering, 11 2008.

[43] P. Gasco. *Development of a Dual Axis Tilt Rotorcraft UAV: Modelling, Simulation and Control.* Cranfield University: School of Engineering, 2012. Masters dissertation.

[44] J. Gertler. V-22 osprey tilt-rotor aircraft: Background and issues for congress. Report, Congressional Research Service, 3 2011.

[45] GetFPV.com. Cobra 2208 2000kv bldc motor product page. `http://www.getfpv.com`, 2016. [Accessed:2017-02-19].

[46] HiSystems GmbH. Mikrokopter quadroxl. `http://www.mikrokopter.de/en/products/kits`, 2016. [Accessed:2016-06-13].

[47] G. Golub and C. Van Loan. *Matrix Computations*, volume 3. Johns Hopkins University Press., 1996.

[48] 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, page N/A, Virginia, USA, 5 2007. American Helicopter Society.

[49] Karsten Groekatthfer and Zizung Yoon. Introudction to quaternions for spacecraft attitude representation. Report, Technical University of Berlin: Department of Astronautics and Aeronatuics, 5 2012.

[50] N. Guenard, T. Hamel, and V. Moreau. Dynamic modelling and control strategy for an x4-flyer. In *International Conference on Control and Automation*, volume N/A, pages 141–146, Budapest, Hungary, 6 2005. IEEE.

[51] R. He, S. Prentice, and N. Roy. Planning in information space for a quadrotor helicopter in a gps-denied environment. In *International Conference on Robotics and Automation*, volume N/A, pages 1814–1820, California, USA, 5 2008. IEEE.

[52] Y. He, H. Pei, and T. Sun. Robust tracking control of helicopters using backstepping with disturbance observers. *Asian Journal of Control*, pages 1387–1402, 10 2013.

[53] M. Hehn and R. D'Andrea. Quadrocopter trajectory generation and control. In *Internation Federation of Automatic Control*, volume 44, pages 1484–1491, Milano, Italy, 9 2011. IEEE.

[54] Drone HiTech. Xrotor 20a esc. `http://dronehitech.com/wp-content/uploads/2016/04/IMG_0524.jpg`, 2016. [Accessed:2016-11-05].

[55] HobbyKing. Orangerx rpm sensor. `http://www.hobbyking.com/hobbyking/store/__61511__Orange_RPM_Sensor.html`, 2016. [Accessed:2016-10-09].

[56] HobbyKing. Rotorstar super mini s-bec. `http://www.hobbyking.com/hobbyking/store/__33987__RotorStar_Super_Mini_S_BEC_6S_10A.html`, 2016. [Accessed:2016-10-08].

[57] HobbyKing. Signal converter module sbus-ppm-pwm. `http://www.hobbyking.com/hobbyking/store/__88384__Signal_Converter_Module_SBUS_PPM_PWM_S2PW_.html`, 2016. [Accessed:2016-10-09].

[58] HobbyKing.com. A28l 920kv brushless outrunn with variable pitch prop assembly. `https://hobbyking.com/en_us/a28l-920kv-brushless-outrunner-w-variable-pitch-prop-assembly.html/?___store=en_us`, 2016. [Accessed:2017-19-0].

[59] HobbyKing.com. Hobby king: The ultimate hobby experience. `http://www.hobbyking.com/hobbyking/store/index.asp`, 2016. [Accessed:2016-06-12].

[60] 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*, volume N/A, pages 1–19, South Carolina, USA, 8 2010. AIAA.

[61] P.A Hokayem and E. Gallestey. Lyapunov stability theorem. Course notes: Nonlinear systems and control, Swiss Federal Institute of Technology: Department of Information Technology and Electrical Engineering, 4 2015.

[62] E.L Houghton and E.W Carpenter. *Aerodynamics for Engineering Students*, volume 5. Butterworth-Heinemann, 2003.

[63] 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`.

[64] COIN-OR Initiative. Ipopt project home. https://projects.coin-or.org/Ipopt, 2017. [Accessed:2017-10-15].

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

[66] 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, Jinan, China, 7 2010. IEEE.

[67] T. Johansen and T. Fossen. Control allocation - a survery. *Automatica*, 49(5):1087–1103, 5 2013.

[68] T. Johansen, T. Fossen, and P. Tondel. Efficient optimal constraint control allocation via multi-parametric programming. *Journal of Guidance, Control and Dynamics*, 28:506–515, 2005.

[69] T. Johansen and J. Tjnns. Adaptive control allocation. *Automatica*, 44(11):2754–2765, 10 2008.

[70] 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.

[71] 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.

[72] F. Kendoul, I. Fantoni, and R. Lozano. Modeling and control of a small autonomous aircraft having two tilting rotors. In *European Control Conference on Decision and Control*, volume N/A, pages 8144–8149, Seville, Span, 12 2005. IEEE.

[73] 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.

[74] P. Krishnamurthy and F. Khorrami. Adaptive backstepping and theta-d based controllers for a tilt-rotor aircraft. In *Mediterranean Conference on Control and Automation*, volume N/A, pages 540–545, Corfu, Greece, 6 2011. IEEE.

[75] R. Kristiansen and P. Nicklasson. Satellite attitude control by quaternion-based backstepping. In *American Control Coference*, volume N/A, pages 907–912, Oregon, USA, 6 2005. AACC.

[76] Jack B. Kuipers. *Quaternions and Rotation Sequences: A Prier with Aplication to Orbital Aerospace and Virtual Reality*, chapter 1, pages 127–143. Princeton University Press, 1 edition, 2002.

[77] W.E. Lang and J.P Young. Effect of inertia properties on attitutde stability of npnrigid spin-stabilized spacecraft. *National Aeronautics and Space Administaction*, pages 1–20, 1974.

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

[79] F.B Leahy. Discrete gust model for launch vehicle assesments. Report, National Aeronautics and Space Administration, 1 2008. https://ams.confex.com/ams/presview.cgi?username=129833&password=129833&uploadid=7813.

[80] J. Lee, B. Min, and E. Kim. Autopilot design of tilt-rotor uav using particle swarm optimization method. In *International Conference on Control, Automation and Systems*, volume N/A, pages 1629–1633, Seoul, Korea, 10 2007. IEEE.

[81] E.J Lefferts, F.L Markley, and M.D Shuster. Kalman filtering for spacecraft attitude estimation. In *AIAAS Aersoapce Sciences Megretski*, volume 20, pages 1–17, Orlando, Florida, 1 1982. AIAA.

[82] LibrePilot. Openpilot/librepilot wiki. `http://opwiki.readthedocs.io/en/latest/index.html`, 2016. [Accessed:2016-5-04].

[83] H. Lim, J. Park, D. Lee, and H. Kim. Build your own quadrotor. *Robotics & Automation*, N/A(N/A):33–45, 9 2012.

[84] 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 Astronauics.

[85] Kilowatt Classroom LLC. Vfd fundamentals. Report, Kilowatt Classroom LLC, 2 2003.

[86] SteadiDrone PTY LTD. Steadidrone home. `http://www.steadidrone.com/`, 2016. [Accessed:2016-06-08].

[87] T. Madani and A. Benallegue. Backstepping control for a quadrotor helicopter. In *International Conference on Intelligent Robots and Systems*, volume N/A, pages 3255–3260, Beijing, China, 10 2006. IEEE.

[88] D. Maiti, A. Acharya, M Chakraborty, and A. Konar. Tuning pid and pi$^{\gamma}$d$^{\delta}$ controllers using the integral time absolute error criterion. *International Conference on Information and Automation for Sustainability*, pages 457–462, June 2008.

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

[90] C. Mantas and J. Puche. Artificial neural networks are zero-order tsk fuzzy systems. *Transactions on Fuzzy Systems*, 16(3):630–644, 6 2010.

[91] MathWorks. Optimization tool with the fmincon solver. `https://www.mathworks.com/help/optim/ug/optimization-tool-with-the-fmincon-solver.html`, 2016. [Accessed:2017-10-15].

[92] C. Mayhew, R. Sanfelice, and A. Teel. On quaternion based attitude control and the unwinding phenomenon. In *American Control Conference*, volume N/A, pages 299–304, California, USA, 7 2011. AACC.

[93] A. Megretski. Dynamics on nonlinear systems: Finding lyapunov functions. Technical report, Department of Electrical Engineering and Computer Sciences: Massachusetts Institute of Technology, 7 2003.

[94] A. Mian and W. Daoboo. Modelling and backstepping-based nonlinear control of a 6dof quadrotor helicopter. *Chinese Journal of Aeronautics*, 8(21):261–268, 5 2008.

[95] S. Napsholm. *Prototype of a Tiltrotor Helicopter*. Norwegian University of Science and Technology: Department of Engineering Cybernetics, 2013. Masters dissertation.

[96] A. Nemati and M. Kumar. Modeling and control of a single axis tilting quadcopter. In *American Control Conference*, volume N/A, pages 3077–3082, Oregon, USA, 5 2014. AACC.

[97] 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.

[98] 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.

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

[100] 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.

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

[102] 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.

[103] C. Papachristos, K. Alexis, and A. Tzes. Design and experimental attitude control of an unmanned tilt-rotor aerial vehicle. In *International Conference on Advanced Robotics*, volume N/A, pages 465–470, Tallinn, Estonia, 6 2011. IEEE.

[104] P.N. Patel, M.A. Patel, R.M. Faldu, and Y.R. Dave. Quadcopter for agricultural surveilance. In *Advance in Electronic and Electrical Engineering*, volume 3, pages 427–432, India, 4 2013. Research India Publications.

[105] Google Patents. V22 osprey patent. `https://www.google.com/patents/US20110177748`, 2010. [Accessed:2017-03-03].

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

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

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

[109] J. Pomet and L. Praly. Adaptive nonlinear regulation: Estimation from the lyupanov equation. In *Transactions on Automatic Control*, volume 37, pages 729–740, Oregon, USA, 6 1992. IEEE.

[110] P. Pounds, R. Mahony, P. Hynes, and J. Roberts. Design of a four-rotor aerial robot. In *Australasian Conference on Robotics and Automation*, volume N/A, pages 145–150, Auckland, USA, 6 2002. IEEE.

[111] A. Pretorius and E. Boje. Shutter delay estimation for a low-ost stand-alone visual tracking system. In *International Federation of Automatic Control*, volume N/A, pages 1–6, Telouse, France, 8 2017. IEEE.

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

[113] 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`.

[114] 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`.

[115] O. Rawashdeh, H.C. Yang, R. AbouSleiman, and B. Sababha. Microraptor: A low cost autonomous quadrotor system. In *International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, volume N/A, pages 1–8, California, USA, 8 2009. American Society of Mechanical Engineers.

[116] Anastasia Razinkove, Igor Gaponov, and Hyun-Chan Cho. Adaptive control over quadcopter uav under disturbances. In *International Conference on Control, Automation and Systems*, volume N/A, pages 386–390, Gyeonggi-do, Korea, 10 2014. IEEE.

[117] M.K. Rwigema. Propeller blade element momentum theory with vortex wake deflection. In *International Congress of the Aeronautical Sciences*, volume N/A, pages 1–9, Nice, France, 9 2010. ICAS.

[118] M. Ryll, H. Bulthoff, and P. Robuffo Giordano. Modelling and control of a quadrotor uav with tilting propellers. In *International Conference on Robotics and Automation*, volume N/A, pages 4606–4613, Minnesota, USA, 5 2012. IEEE.

[119] M. Ryll, H. Bulthoff, and P. Robuffo Giordano. Modelling and control of a quadrotor uav with tilting propellers. In *International Conference on Robotics and Automation*, volume N/A, pages 295–302, Karlsruhe, Germany, 5 2013. IEEE.

[120] S. Salazar-Cruz, A. Palomino, and R. Lozano. Trajectory tracking for a four rotor mini-aircraft. In *ECC Conference on Decision and control*, volume N/A, pages 2505–2510, Seville, Sapin, 12 2005. IEEE.

[121] Inc. Saleae. Logic8. `https://www.saleae.com/`, 2017. [Accessed:2017-03-12].

[122] A. Sanchez, J. Escareo, O. Garcia, and R. Lozano. Autonomous hovering of a noncyclic tiltrotor uav: Modeling, control and implementation. In *International Federation of Automatic Control*, volume N/A, pages 803–808, Seoul, Korea, 6 2008. IEEE.

[123] J. Seddon. *Basic Helicopter Aerodynamics*, chapter 2-5, pages 4–66. BSP Professional Books, 1 edition, 1990.

[124] K. Shoemake. Quaternions. Siggraph course lecture notes, Department of Computer and Information Science; University of Pennsylvania, N/A 1987.

[125] P. Singla, D. Mortari, and J. Junkins. How to avoid singularity when using euler angles? *Advances in the Astronautical Sciences*, N/A(4):1409–1426, 1 2005.

[126] G. Slabaugh. Computing euler angles from a rotation matrix. Lecture notes, City University, London, 5 1999.

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

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

[129] 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`.

[130] 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`.

[131] A. Tayebi and S. McGilvray. Attitude stabilization of a vtol quadrotor aircraft. *Transactions on Control Systems Technology*, 14(3):562–571, 5 2006.

[132] 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.

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

[134] John Ting-Yung Wen and Kenneth Kreutz-Delgado. The attitude control problem. *Transactions on Automatic Control*, 36(10):1148–1162, 10 1991.

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

[136] P. Tsiotras, M. Corless, and J. Longuski. A novel approach to the attitude control of axisymmetric spacecraft. *Automatica*, 31(8):2754–2765, 8 1995.

[137] Ultimaker. Ultimaker v2+ product page. `https://ultimaker.com/en/products/ultimaker-2-plus#volume`, 2016. [Accessed:2016-9-11].

[138] F. van den Berg and A.P Engelbrecht. *A study of particle swarm optimization particle trajectories*, volume 1. Department of Computer Science, University of Pretoria,, 2005.

[139] 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: `https://www.studocu.com/en/course/technische-universiteit-delft/automatic-flight-control-system-design/13152`.

[140] R. Votel and D. Sinclair. Comparison of control moment gyros and reaction wheels for small earth-observing satellites. In *Conference on Small Satellites*, volume 26, pages 1–7, N/A, 8 2012. AIAA.

[141] Tao Wang, Tao Zhao, Du Hao, and Mingxi Wang. Transformable aerial vehicle. `http://www.google.com/patents/US20140263823`, 2014. [Accessed:2017-01-16].

[142] A Weiss, I. Kolmanovsky, D.S Bernstein, and A. Sanyal. Inertia-free spacecraft attitude control using reaction wheels. In *Journal of Guidance, Control and Dynamics*, volume 36, pages 1425–1439. AIAA, 8 2013.

[143] 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/`.

[144] X. Xiaozhu, L. Zaozhen, and C. Weining. Intelligent adaptive backstepping controller design based on the adaptive particle swarm optimization. In *Chinese Control and Decision Conference*, volume N/A, pages 13–17, Guilin, China, 6 2009. IEEE.

[145] S. Xin and Z. Zaojian. A fuzzy sliding mode controller with adaptive disturance approximation for an underwater robot. In *Asian Conference on Informatics in Control, Automation and Robotics*, volume 2, pages 50–53, Wuhan, China, 10 2010. IEEE.

[146] Z. Zaheer, A. Usmani, E. Khan, and M.A Qadeer. Aerial surveillance system using uav. In *Wireless and Optical Communications Netweorks*, volume 13, pages 1–7, Hyderabad, India, 12 2016. IEEE.