

Non-zero Setpoint Tracking for a Dual-Axis Tilting Quadrotor Helicopter

Nicholas Von Klemperer¹ and Edward Boje²

Abstract—This work applies non-zero attitude and position setpoint tracking to a quadrotor helicopter; achieved by mechanically solving the problem of a quadrotor’s inherent underactuation. Introduction of overactuation results in a multibody, non-linear system to be modelled and stabilized. High level setpoint tracking controllers design virtual control inputs which are then allotted out to the plant’s actuators by a lower level allocation rule. The non-linear state-space controller(s) are optimized iteratively using a swarm algorithm; producing a control suite that successfully tracks an orbital trajectory.

Index terms— control, non-linear, over actuated, optimization, quadrotor

I. INTRODUCTION

Quadrotor helicopters, first published in [1], are currently an extremely popular field of control research. As almost every quadrotor based papers will mention; this is due to a recent advent in cheap embedded systems for implementation of the UAV’s control loop. Quadrotor helicopters are inherently underactuated, making the control plant ill-posed. Their control problem is design of 4 controllable inputs (namely the four propeller’s rotational speed) to articulate a 6 degree of freedom (*DOF*) body. Standard control solutions linearize the plant around the attitude’s origin; reducing the degrees of controllable freedom the system has. What follows is a proposal that achieves non-zero setpoint tracking for a quadrotor by introducing overactuation to the vehicle, apply no such linearisation to the dynamic equations of motion. A high level control law designs net torque and force inputs to be applied to the multibody vehicle about it’s center of motion. A lower level allocation rule then allots out that virtual control input amongst the actuator set to be physically implemented.

Over actuated control is most common in the context of satellite attitude control. Typically a satellite’s extra actuators are included for the sake of fault tolerance, implemented only when higher priority acutators fail mechanically. Fault tolerant control for satellites is discussed in [2], where the additional actuators are included for redundancy rather than complementing a control input. Alternatively, [3] implemented quaternion based backstepping to track a satellite’s orbital trajectory whilst making use of all available actuators, using a pseudo-inversion allocation rule.

A series of allocation schemes is surveyed in [4], most allocation applies some sort of inversion to find explicit actuator positions.

Not much work has been done on reconfigurable aerospace vehicles. The only commerical example is the DJI Inspire1 which only can raise or lower it’s support arms. Reconfigurable aerospace frames are usually avoided due to their dynamic complexities. Dual-axis tilting capabilities are often incorporated into birotor aircraft; [5] presented a dual-axis tilting bi-rotor design but in the control constrained the actuation to an effective single axis of tilt. In the case of tilting quadrotor aircraft, [6] proposed a single co-axial rotation of a quadrotor’s propellers which was subsequently tested in [7]. Another paper, [8], similarly proposed a single axis tilting quadrotor aircraft. Neither single axis quadrotor papers consider the dynamic consequences of reconfiguring the quadcopter mid-flight. The only dual-axis tilting quadrotor work, [9], treats the control inputs as a regular fixed axis quadrotor, but with an introduced actuator plant that produces gyroscopic torque from pitching/rolling the rotating propeller module.

The remainder of this paper is arranged as follows; the prototype’s design and it’s reference conventions are briefly described in II. Thereafter, III presents the state differential dynamic model used for 6-DOF equations of motion, including the multibody non-linearities which stem from the prototype’s structure and mid-flight reconfiguration. Position and attitude controllers are derived in IV, together with an allocation rule. Each control law’s coefficients are designed using a swarm optimization discussed in V. Finally setpoint tracking is empirically tested in VI with conclusions drawn in VII.

II. DESIGN AND CONVENTIONS

Two Cartesian reference frames are considered here; first the fixed inertial frame \mathcal{F}^I which all motion is taken with respect to. Secondly the body frame \mathcal{F}^b which is attached and centred on the vehicle itself. Rotation (or *transformation*) of some vector \vec{v}_I from the inertial frame to the body frame is applied by a quaternion operation:

$$v_b = Q_b \otimes \vec{v}_I \otimes Q_b^* \in \mathcal{F}^b \quad (1)$$

Wherein $Q_b = [q_0 \ \vec{q}]^T$ is the attitude’s quaternion, Q_b^* is the quaternion complex conjugate and \otimes is the hamilton product between two quaternions. Quaternions are detailed thoroughly in [10], if such concepts are unfamiliar to the reader...

¹MSc (Eng) Candidate, Engineering & the Built Environment, Faculty of Electrical Engineering, University of Cape Town, Cape Town, South Africa
nicholas.vonklemperer@alumni.uct.ac.za

²Professor and Head of department, Faculty of Electrical Engineering, University of Cape Town, Cape Town, South Africa,
edward.boje@uct.ac.za

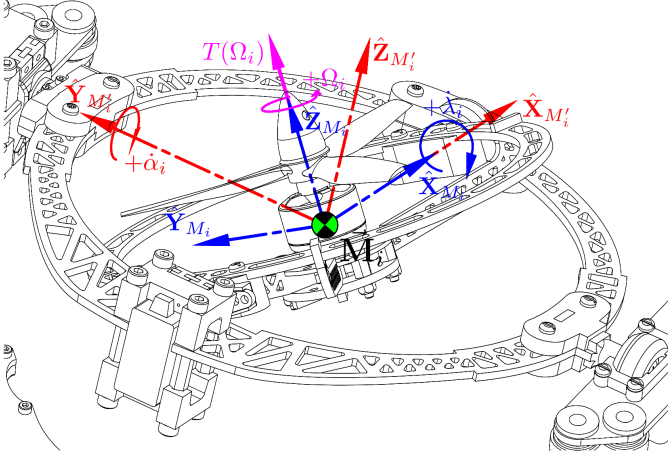


Fig. 1. Actuator motor module

The prototype extends the actuation suite of a regular quadcopter, such as [1], to an overactuated 12-*DOF* system by introducing 8 additional actuator plants. A standard quadrotor consists of only 4 controllable inputs; namely each of the propeller's rotational speeds $\Omega_{1 \rightarrow 4}$, under actuating the control plant. The design here adds 2 revolute actuators to respectively pitch and roll *each* of the 4 motors and propellers about their \hat{X}_{M_i} and \hat{Y}_{M_i} axes, aligned as per Fig:1.

The *motor module* frame \mathcal{F}^{M_i} is attached to the i^{th} motor ($i \in [1 : 4]$), whose propeller produces an input thrust in the \hat{Z}_{M_i} axis direction of it's frame. The two actautors rotate the motor/propeller in sequential gimbal-like fashion; first about it's \hat{X}_{M_i} axis by an angle of λ_i . Then about an intermediate frame's \hat{Y}_{M_i}' axis by an angle of α_i . The resultant thrust vector $\vec{T}(\Omega_i)$ in the \hat{Z}_{M_i} direction is calculated using blade-element-momentum (*BEM*) theory, [11]. Thrust and power coefficients, C_T and C_P respectively, relate the propeller's rotational speed in [RPS] to it's produced thrust and aerodynamic torque as per:

$$\vec{T}(\Omega_i) = C_T(J)\rho\Omega_i^2 D^4 \cdot \hat{Z}_{M_i} \quad [\text{N}] \quad (2a)$$

$$\vec{\tau}_Q(\Omega)_i = C_P(J)\rho\Omega_i^3 D^5 (1/R\Omega_i) \cdot \hat{Z}_{M_i} \quad [\text{Nm}] \quad (2b)$$

Where J is the propeller's advance ratio, ρ is the air density taken at *STP* and D is the propeller's diameter in [m]. Further details of *BEM* theory and a propeller's aerodynamics are presented in [11]. Each thrust and power coefficient used was linearly interpolated from data in [12] to match physical tests applied to the hardware under consideration.

Each of the four motor modules, in Fig:1, are aligned relative to the body frame's center \bar{O}_b as per Fig:2. The 1st and 3rd motor modules have their positive and negative $\pm\hat{X}_{M_1/M_3}$ axes aligned with the body frame's \hat{X}_b axis. While the 2nd and 4th modules have their positive and negative $\pm\hat{Y}_{M_2/M_4}$ axes aligned with the \hat{Y}_b body frame's axis. Motors 1 and 3 have clockwise propeller rotations, $+\Omega_{1/3}$, whereas motors 2 and 4 have anti-clockwise propeller rotations, $-\Omega_{2/4}$. An instantaneous configuration of the multibody system is fully described by the 4 propeller's rotational speed and the 8 additional actuator rotational positions.

That actuator configuration input matrix u , within the saturation limited actuator subset \mathbb{U} , is such that:

$$u \triangleq [\Omega_1 \quad \lambda_1 \quad \alpha_1 \quad \dots \quad -\Omega_4 \quad \lambda_4 \quad \alpha_4]^T \in \mathbb{R}^{12} \quad (3)$$

A propeller's generated thrust or torque, (2a) and (2b) respectively, has the body frame vector counterpart:

$$\vec{T}_{1 \rightarrow 4} = Q_{M_i}^* \otimes \vec{T}(\Omega_i) \otimes Q_{M_i} \quad [\text{N}], \in \mathcal{F}^b \quad (4a)$$

$$\vec{Q}_{1 \rightarrow 4} = Q_{M_i}^* \otimes \vec{Q}(\Omega_i) \otimes Q_{M_i} \quad [\text{Nm}], \in \mathcal{F}^b \quad (4b)$$

Where each Q_{M_i} is the relative quaternion from the motor module's frame \mathcal{F}^{M_i} to the body frame \mathcal{F}^b such that:

$$Q_{M_i} \triangleq Q_z(\sigma_i) \otimes Q_y(\alpha_i) \otimes Q_x(\lambda_i) \quad (4c)$$

With Q_z , Q_y and Q_x being quaternion rotations about each \hat{Z} , \hat{Y} and \hat{X} axes respectively. Note that σ_i is the \hat{Z}_b orthogonal angular rotation between each motor module; detailed in the layout of Fig:2. The net heave and control torque acting on the body due to actuation action is simply:

$$\vec{F}_\mu = \sum_{i=1}^4 \vec{T}(\Omega_i, \lambda_i, \alpha_i) \quad [\text{N}] \quad (5a)$$

$$\vec{\tau}_\mu = \sum_{i=1}^4 \vec{L}_i \times \vec{T}(\Omega_i, \lambda_i, \alpha_i) + \vec{\tau}_Q(\Omega_i, \lambda_i, \alpha_i) \quad [\text{Nm}] \quad (5b)$$

Each thrust vector $\vec{T}(\Omega_i, \lambda_i, \alpha_i) \in \mathcal{F}^b$ is a combined function of (2a) and (4a). The net torque $\vec{\tau}_\mu \in \mathcal{F}^b$ comprises of differential torque arm cross-products from input thrusts $\vec{T}_{1 \rightarrow 4}$ each acting at a length \vec{L} in Fig:2 and the propeller's aerodynamic torque, $\vec{\tau}_Q(\Omega_i, \lambda_i, \alpha_i)$, as a combined function of (2b) and (4b). The latter aerodynamic torque $\sum \vec{\tau}_Q(\Omega_i, \lambda_i, \alpha_i)$ is a term to be *compensated* for in feedback, not used as a controllable input.

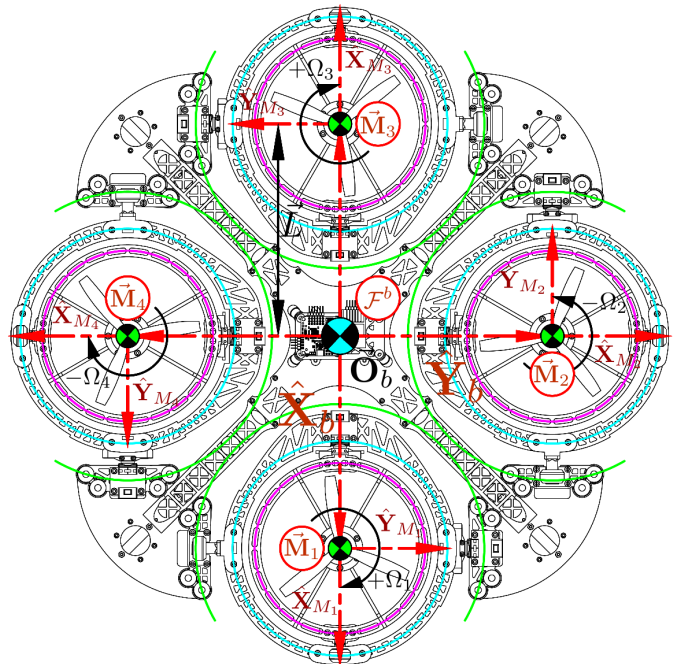


Fig. 2. Body frame axes layout

III. DYNAMICS

The typical Lagrangian dynamics for a *rigid* 6-DOF body are as follows; derived from first principles in [15]. For a rigid body with an angular velocity $\vec{\omega}_b$ in \mathcal{F}^b and an inertial matrix J_b aligned with axes in \mathcal{F}^b ; the quaternion dynamics for the attitude's differential equations of motion are:

$$\dot{Q}_b = 1/2(Q_b \otimes \vec{\omega}_b) \in \mathcal{F}^I \quad (6a)$$

$$\dot{\vec{\omega}}_b = J_b^{-1}(-\vec{\omega}_b \times J_b \vec{\omega}_b + \vec{\tau}_\mu) [\text{rad.s}^{-2}], \in \mathcal{F}^b \quad (6b)$$

For a net torque control input $\vec{\tau}_\mu \in \mathcal{F}^b$. Similarly the vehicle's translational dynamics, for an inertial position $\vec{\mathcal{E}} \in \mathcal{F}^I$ and a translational velocity $\vec{v}_b \in \mathcal{F}^b$ and a net input force $\vec{F}_\mu \in \mathcal{F}^b$:

$$\dot{\vec{\mathcal{E}}} = Q_b^* \otimes \vec{v}_b \otimes Q_b [\text{m.s}^{-1}], \in \mathcal{F}^I \quad (7a)$$

$$\dot{\vec{v}}_b = m_b^{-1}(-\vec{\omega}_b \times \vec{v}_b + m_b \vec{G}_b + \vec{F}_\mu) [\text{m.s}^{-2}], \in \mathcal{F}^b \quad (7b)$$

For a mass m_b and a gravitational acceleration \vec{G}_b transformed to the body frame. Particulars associated with the multibody dynamics resulting from the system's actuation were additively introduced to the above equations of motion (6) and (7). Furthermore, explicit calculation of the instantaneous net inertial matrix is not a constant J_b but a function the actuator matrix (3); $J_b(u)$. The complete multibody dynamics for the design are presented in [15], from which this paper is adapted, but are excluded here for the sake of brevity. In summary; each of the 13 independent bodies are revolutely connected with only one permissible degree of rotational freedom between each sequential structure. Fig:3 shows the exploded components of a single motor module; each with their own rotational velocity relative to \mathcal{F}^b , resulting from the actuation rates $\dot{\lambda}_i$ and $\dot{\alpha}_i$. The non-Newtonian frames for each rotating sub-body made quantifying the net multibody's response to rotational and translational motion highly complicated. A complete Lagrangian, [13], was constructed for each body's net kinetic and potential energies. As per Lagrangian dynamics, the partial derivatives with respect to the Lagrangian's path co-ordinates results in the *generalized forces*, in this case torques $\vec{\tau}_b$, acting on the system.

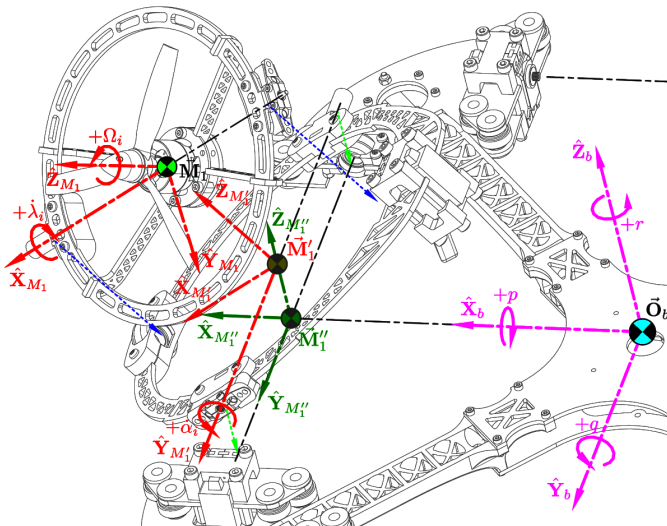


Fig. 3. Multi-body overview

Each motor module contributes some non-zero inertial derivative towards the dynamic response calculations, complicating their solutions. If a body has an inertia J in it's principle frame, but the body is rotated by θ about some unit axis \hat{u} . The effective transformed inertia is:

$$J' = R_{\hat{u}}(\theta)(J)R_{\hat{u}}^{-1}(\theta) \quad (8a)$$

Where $R_{\hat{u}}(\theta)$ is the rotation matrix about that unit axis, [16]. Quaternions are ill suited for inertial transformations; hence the use of rotation matrices here. If θ is a time varying angular rotation then (8a) has a time derivative

$$\dot{J}' = d/dt(R_{\hat{u}}(\theta)(J)R_{\hat{u}}^{-1}(\theta)) \quad (8b)$$

$$= [\dot{\theta}]_{\times} R_{\hat{u}}(\theta)(J)R_{\hat{u}}^{-1}(\theta) + R_{\hat{u}}(\theta)(\dot{J})R_{\hat{u}}^{-1}(\theta) - R_{\hat{u}}(\theta)(J)[\dot{\theta}]_{\times} R_{\hat{u}}^{-1}(\theta) \quad (8c)$$

Where $\vec{\theta}$ is the projected vector of θ in the direction of it's unit axis \hat{u} , or that $\vec{\theta} \triangleq \theta \cdot \hat{u}$. Then \dot{J} is the inertia's time derivative (if any) in it's principle frame. Finally $[\cdot]_{\times}$ is such that a two vector cross product $\vec{a} \times \vec{b}$, is applied as a matrix multiplication:

$$\vec{a} \times \vec{b} \equiv [\vec{a}]_{\times} \vec{b} \quad (8d)$$

A body's *generalized force* (torque) response to an applied angular velocity/acceleration consists of three components; an inertial rate component, an inertial response to the applied acceleration and a gyroscopic cross-product of the relative angular velocities. Those net responses are lumped into an additive torque term $\hat{\tau}_b(u)$ which, like the body's net rotational inertia, is a function of the actuator configuration $u \in \mathbb{U}$. The multibody response derivation is, however, far too cumbersome to include here and so is omitted for conciseness. The angular differential equation, (6), then treats the net aerodynamic torque $\vec{\tau}_Q$ from (5b) and the multibody non-linearity responses $\hat{\tau}_b(u)$ as terms to be compensated for:

$$\therefore \dot{\vec{\omega}}_b = J_b(u)^{-1}(-\vec{\omega}_b \times J_b(u)\vec{\omega}_b - \hat{\tau}_b(u) + \vec{\tau}_Q + \vec{\tau}_g + \vec{\tau}_\mu) \quad (9a)$$

Lastly, the torque $\vec{\tau}_g$ is from the eccentric center of gravity produced by rotated motor modules. The net center of gravity is $C.M_b(u)$ and so the gravity torque is a cross-product of it's deviation from the body frame's origin \vec{O}_b and the gravitational force on the body:

$$\vec{\tau}_g = (\vec{O}_b - C.M_b(u)) \times m_b \vec{G}_b \quad (9b)$$

IV. CONTROLLER DESIGN

The control block is divided into two components; a higher level setpoint tracking controller designs a virtual control inputs $\vec{v}_d = [\vec{F}_d \ \vec{\tau}_d]^T \in \mathcal{F}^b$ for (7b) and (6b). Thereafter an allocation rule solves for an explicit actuator positions $u \in \mathbb{U}$ which physically command that control input, \vec{v}_c as per the actuators *effectiveness* function. The effectiveness function or $B(\vec{x}, t, u) = [\vec{F}_c \ \vec{\tau}_c]^T = \vec{v}_c$, for collective attitude and position states $\vec{x} = [\vec{\mathcal{E}} \ Q_b]^T$, is a combination of thrust and torque inputs (5a) and (5b) and their aerodynamics (2a-2b).

Only first order setpoint tracking is applied here to follow a state setpoint $\vec{x}_d = [\vec{\mathcal{E}}_d \ Q_d]^T$. Higher order (velocity/acceleration) state derivative setpoints are then $\dot{\vec{x}}_d = [\dot{\vec{\mathcal{E}}}_d \ \dot{Q}_d]^T = \vec{0}$. The control problem of setpoint tracking for an error state \vec{x}_e is to find a control law $\vec{v}_d = H(\vec{x}_e, t)$ such that the error is asymptotically stabilized; $\lim_{t \rightarrow \infty} \vec{x}_e \rightarrow \vec{0}$.

A. Attitude Control Derivation

The attitude plant, (6) and extended to (9a), is stabilized first because of its independence from the position control loop. For a quaternion setpoint Q_d , the quaternion error is a *multiplicative* term defined as:

$$Q_e \triangleq Q_b^* \otimes Q_d \quad (10a)$$

An angular velocity setpoint is defined with respect to the attitude setpoint's frame, or that $\vec{\omega}_d \in \mathcal{F}^d$. The subtractive angular velocity error $\vec{\omega}_e$ transforms that setpoint $\vec{\omega}_d$ back to \mathcal{F}^b :

$$\vec{\omega}_e \triangleq Q_e \otimes \vec{\omega}_d \otimes Q_e^* - \vec{\omega}_b = -\vec{\omega}_b \Big|_{\vec{\omega}_d=\vec{0}} \in \mathcal{F}^b \quad (10b)$$

Proposing a plant dependent Proportional-Derivative (PD) attitude controller with compensation feedback terms. The PD controller designs an input torque $\vec{\tau}_d \in \mathcal{F}^b$ using the angular velocity error $\vec{\omega}_e$ and the *vector* quaternion error \vec{q}_e :

$$\vec{\tau}_d = \underbrace{K_p \vec{q}_e + K_d \vec{\omega}_e}_{\text{independent}} + \underbrace{\vec{\omega}_b \times J_b(u) + \hat{\tau}_b(u) - \vec{\tau}_Q - \vec{\tau}_g}_{\text{compensation}} \quad (11)$$

Where K_p and K_d are both positive diagonal 3×3 gain coefficient matrices. A controller's stability throughout its general trajectory is proven through Lyapunov stability theorem, [14]. A Lyapunov function candidate (LFC) for the attitude's trajectory *error* such that $V_1(\vec{0}) = 0$ is:

$$V_1(Q_e, \vec{\omega}_e) = \vec{q}_e^T \vec{q}_e + (1 - |q_0|)^2 + \frac{1}{2} \vec{\omega}_e^T J_b(u) K_p^{-1} \vec{\omega}_e > 0, \forall (Q_e, \vec{\omega}_e) \quad (12a)$$

From the unit quaternion identity $\|Q\| = \vec{q}^T \vec{q} + q_0^2 = 1$ and (10b); V_1 then has a simplified derivative:

$$\dot{V}_1(Q_e, \vec{\omega}_e) = -2\dot{q}_0 + \vec{\omega}_b^T J_b(u) K_p^{-1} \vec{\omega}_b \quad (12b)$$

Expanding the quaternion derivative (6a) into component form:

$$\dot{Q} = \begin{bmatrix} -1/2(\vec{q}^T \vec{\omega}) \\ 1/2([\vec{q}]_\times + q_0 \mathbb{I}_{3 \times 3}) \vec{\omega} \end{bmatrix} \quad (12c)$$

Substituting the quaternion's scalar derivative \dot{q}_0 from (12c), angular acceleration $\dot{\vec{\omega}}_b$ from (6b) and the PD control law $\vec{\tau}_d$ from (11); the LFC derivative \dot{V}_1 reduces:

$$\dot{V}_1 = -\vec{q}_e^T \vec{\omega}_b + \vec{\omega}_b^T K_p^{-1} (-K_d \vec{\omega}_b + K_p \vec{q}_e) \quad (12d)$$

$$= -\vec{\omega}_b K_p^{-1} K_d \vec{\omega}_b < 0, \exists (K_p^{-1}, K_d) > 0 \quad (12e)$$

The LFC derivative (12e) is negative definite for $\forall (Q_e, \vec{\omega}_e)$; which leads to the global asymptotic stability limits:

$$\lim_{t \rightarrow \infty} \vec{\omega}_e = \vec{0} \therefore \lim_{t \rightarrow \infty} \vec{\omega}_b = \vec{0}^- \quad (13a)$$

$$\lim_{t \rightarrow \infty} Q_e = [\pm 1 \quad \vec{0}]^T \quad (13b)$$

B. Position Control Derivation

Position setpoints $\vec{\mathcal{E}}_d$ are defined in \mathcal{F}^I , whereas body translational velocity \vec{v}_b is in \mathcal{F}^b . Translational position error is:

$$\vec{\mathcal{E}}_e \triangleq \vec{\mathcal{E}}_d - \vec{\mathcal{E}}_b \in \mathcal{F}^I \quad (14a)$$

Whilst, for a velocity setpoint $\vec{v}_d = \vec{0}$, the velocity error is:

$$\vec{v}_e \triangleq \vec{v}_d - \vec{v}_b = -\vec{v}_b \Big|_{\vec{v}_d=\vec{0}} \in \mathcal{F}^b \quad (14b)$$

A PD position controller designs a control force from the position and velocity errors but in the *same* frame. Transforming the inertial position error to \mathcal{F}^b :

$$\vec{X}_b \triangleq Q_b \otimes \vec{\mathcal{E}}_b \otimes Q_b^* \in \mathcal{F}^b \quad (15a)$$

$$\therefore \vec{X}_e = \vec{X}_d - \vec{X}_b = Q_b \otimes (\vec{\mathcal{E}}_d - \vec{\mathcal{E}}_b) \otimes Q_b^* \quad (15b)$$

$$\rightarrow \dot{\vec{X}}_e = Q_b \otimes (\dot{\vec{\mathcal{E}}}_d - \dot{\vec{\mathcal{E}}}_b) \otimes Q_b^* = -\vec{v}_b \Big|_{\dot{\vec{\mathcal{E}}}_d=\vec{0}} \quad (15c)$$

Note that (15b) holds true because of a quaternion multiplication's associative, but *not* commutative, properties. A PD position controller then designs an input force $\vec{F}_d \in \mathcal{F}^b$:

$$\vec{F}_d = \underbrace{A_p \vec{X}_e + A_d \vec{v}_e}_{\text{independent}} + \underbrace{\vec{\omega}_b \times m_b \vec{v}_b - m_b \vec{G}_b}_{\text{compensation}} \quad (16)$$

With positive diagonal 3×3 gain coefficients A_p and A_d . The Coriolis acceleration component of (16), $\vec{\omega}_b \times m_b \vec{v}_b$, and the transformation of $\vec{\mathcal{E}}_e \in \mathcal{F}^b$ to $\vec{X}_e \in \mathcal{F}^b$ are the reasons that attitude must be stabilized before position control can be applied. Again, using an LFC to prove the stability of (16):

$$V_2(\vec{X}_e, \vec{v}_e) = \frac{1}{2} \vec{X}_e^T A_p \vec{X}_e + \frac{1}{2} \dot{\vec{X}}_e^T m_b \dot{\vec{X}}_e \quad (17a)$$

$$= \frac{1}{2} \vec{X}_e^T A_p \vec{X}_e + \frac{1}{2} \vec{v}_b^T m_b \vec{v}_b > 0, \forall (\vec{X}_e, \dot{\vec{X}}_e) \quad (17b)$$

Where (15c) was used to simplify (17b). That LFC then has a derivative \dot{V}_2 , with (15c) substituted for $\dot{\vec{X}}_e$:

$$\dot{V}_2(\vec{X}_e, \dot{\vec{X}}_e) = -\vec{X}_e^T A_p \vec{v}_b + \vec{v}_b^T m_b \dot{\vec{v}}_b \quad (17c)$$

Then, introducing (7) for $\dot{\vec{v}}_b$ and substituting the PD control law \vec{F}_d from (16), (17c) reduces:

$$\dot{V}_2 = -\vec{X}_e A_p \vec{v}_b + \vec{v}_b^T (A_p \vec{X}_e - A_d \vec{v}_b) \quad (17d)$$

$$= -\vec{v}_b^T A_d \vec{v}_b < 0, \exists (A_p, A_d) > 0 \quad (17e)$$

The asymptotic stability asserted (17e) holds for $\forall (\vec{\mathcal{E}}_e, \dot{\vec{\mathcal{E}}}_e)$, irrespective of the transformation applied in (15). The stabilizing limits then follow:

$$\lim_{t \rightarrow \infty} \vec{X}_e = \vec{0} \therefore \lim_{t \rightarrow \infty} \vec{\mathcal{E}}_b = \vec{\mathcal{E}}_d \quad (18a)$$

$$\lim_{t \rightarrow \infty} \dot{\vec{X}}_e = -\vec{v}_b = \vec{0} \Big|_{\dot{\vec{\mathcal{E}}}_d=\vec{0}} \quad (18b)$$

C. Control Allocation

Lastly values for the 12 actuators in (3) need to be found from the virtual control input(s), (11) and (16). Control allocation is explained in [4]; so the details of the minimization problem are not considered here. If \vec{v}_c is the commanded input counterpart of the virtual control input \vec{v}_d , the two are related by the allocator's effectiveness matrix $B(\vec{x}, u, t)$:

$$\vec{v}_d = H(\vec{x}_e, t) \iff B(\vec{x}, t, u) = \vec{v}_c \quad (19)$$

Typical pseudo-inversion, as applied here, requires an *affine* relationship between the effectiveness function and the input matrix; $B(\vec{x}, t, u) \equiv B'(\vec{x}, t)u$. The combination of net inputs in (5) and aerodynamics in (2) are not easily reduced to a multiplicative relationship as needed. An abstraction is then used such that an allocation rule finds four 3-dimensional thrust vectors $\vec{T}_{1 \rightarrow 4} \in \mathcal{F}^b$ to then be actuated by each motor module (Fig:1). From (5), *without* the net aerodynamic torque, the commanded input \vec{v}_c is the a function of $\vec{T}_{1 \rightarrow 4}$:

$$\vec{v}_c = \begin{bmatrix} \vec{F}_\mu \\ \vec{\tau}_\mu \end{bmatrix}^T = \begin{bmatrix} \mathbb{I}_{3 \times 3} & \cdots & \mathbb{I}_{3 \times 3} \\ [\vec{L}_1]_\times & \cdots & [\vec{L}_4]_\times \end{bmatrix} \begin{bmatrix} \vec{T}_1 \\ \vdots \\ \vec{T}_4 \end{bmatrix} \quad (20a)$$

$$\rightarrow \vec{v}_c = B'(\vec{x}, t) [\vec{T}_{1 \rightarrow 4}] \quad (20b)$$

Then, using the pseudo-inversion of (20), the 4 thrust vectors are found from the controller designed inputs $\vec{v}_d = [\vec{F}_d \ \vec{\tau}_d]^T$:

$$\vec{T}_{1 \rightarrow 4} = B'^T(\vec{x}, t) (B'(\vec{x}, t) B'^T(\vec{x}, t))^{-1} \vec{v}_d \quad (21a)$$

$$\rightarrow \vec{T}_{1 \rightarrow 4} = B^\dagger(\vec{x}, t) \vec{v}_d \quad (21b)$$

Each of the four thrust vectors are used to find a corresponding motor module propeller's rotational speed Ω_i and both actuator rotational positions λ_i and α_i . A trigonometric inversion “unwinds” the rotations applied to the produced the thrust vector from (4a). Here *rotation matrices* are used in lieu of quaternions for (4a) to solve λ_i and α_i :

$$\vec{T}_i = R_{M_i}(\sigma_i, \alpha_i, \lambda_i) \vec{T}(\Omega_i) \quad (22a)$$

$$= R_z(\sigma_i) R_y(\alpha_i) R_x(\lambda_i) \vec{T}(\Omega_i) \in \mathcal{F}^b \quad (22b)$$

The propeller's thrust vector $\vec{T}(\vec{\Omega}_i)$ is entirely in the \hat{Z}_{M_i} direction. That (22b) simplifies for the i^{th} motor module with an orthogonal σ_i rotation about the \hat{Z}_b axis:

$$\rightarrow \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} = \begin{bmatrix} s\sigma_i s\lambda_i + c\sigma_i s\alpha_i c\lambda_i \\ s\sigma_i s\alpha_i c\lambda_i - c\sigma_i s\alpha_i \\ c\alpha_i c\lambda_i \end{bmatrix} T(\Omega_i) \quad (22c)$$

Then inverse trigonometry can be used to solve for each actuator angle in (22c). Because $\sigma_{1 \rightarrow 4} \in [0^\circ, 90^\circ, 180^\circ, 270^\circ]$ inversion equations for each motor module will be different. First, the thrust vector's magnitude $|\vec{T}_i|$ are used to solve for the propeller's speed Ω_i from (2a). Then for reference; the 1st motor module's thrust inversion equation (when $\sigma_1 = 0^\circ$) is as follows:

$$\begin{bmatrix} \Omega_1 \\ \lambda_1 \\ \alpha_1 \end{bmatrix} = \begin{bmatrix} \left(\sqrt{T_x^2 + T_y^2 + T_z^2} / C_T(J) \rho D^4 \right)^{1/2} \\ \text{atan2}\left(T_x, |\vec{T}_1| \sqrt{|\vec{T}_1|^2 - T_x^2}\right) \\ -\text{atan2}\left(T_x, T_z |\vec{T}_1|\right) \end{bmatrix} \quad (23)$$

Thrust inversions solving for the remaining motor module's actuator positions are simply analogues of (22c) but with a \hat{Z}_b rotation for the respective intervals of σ_i . The arctangent2 function is used in (23) to find full quadrature results for the actuator positions, not just their acute solutions. Similarly the identity $\arctan(x) \equiv \arcsin(x/\sqrt{1-x^2})$ was used to simplify (22c) to accommodate for the use of *atan2*.

In summary, controllers (11) and (16) design \vec{v}_d . Thrust vectors to be actuated by each module are found with pseudo-inversion in (21). Lastly propeller and servo values are solved for using the module's respective thrust inversion (23).

V. COEFFICIENT SELECTION

Stability for the attitude and position controllers are respectively shown in (12) and (17). However, proportional and derivative coefficients are still as yet undefined. Often a *monte carlo* approach is used at the controller coefficient selection stage; choosing coefficient values seemingly at random without clear design or forethought. Here a Particle Swarm Optimization (*PSO*) is used to apply a *gradient free* optimization of the controller coefficients, [17]. Each coefficient is modelled as a particle within the swarm \vec{x} , the swarm then progresses through a search space iteratively testing the n^{th} interval's performance and directing subsequent searches based on previous performances. The progression of the swarm is governed by the velocity function:

$$\vec{x}_{n+1} = \vec{v}_n + \vec{x}_n \quad (24a)$$

$$\vec{v}_{n+1} \triangleq \vec{v}_n + c_1 * r_1 (\vec{P}_{best} - \vec{x}_n) + c_2 * r_2 (\vec{G}_{best} - \vec{x}_n) \quad (24b)$$

Each $*$ operator in (24) applies an element-by-element matrix coefficient multiplication, \vec{P}_{best} and \vec{G}_{best} are previous swarm positions of local and global optima. Random seeded vectors r_1 and r_2 explore the search space, biased by 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. Each swarm's position is simulated over a number of set point attitude and position step tests; it's performance averaged from an ITAE cost function integrated throughout that simulation:

$$\zeta \triangleq \int_{t_0}^{t_\infty} t |\vec{e}(t)|.dt \quad (25)$$

With an error $\vec{e}(t)$ deviating from the plant's given setpoint. The ITAE integral is calculated over an 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 behavior. The optimization procedure implemented is thoroughly detailed in [15]. The optimized diagonal coefficient matrices for both PD controllers were respectively:

$$K_p = \text{diag}(3.568, 5.270, 6.070)$$

$$\text{and } K_d = \text{diag}(9.015, 11.485, 20.183) \quad (26a)$$

$$A_p = \text{diag}(2.417, 3.156, 2.590)$$

$$\text{and } A_d = \text{diag}(3.479, 3.385, 3.870) \quad (26b)$$

VI. SETPOINT AND TRAJECTORY TRACKING

The efficacy of the proposed attitude and position controllers were evaluated with a simulated state step from the origin. Starting from $Q_b(0) = [1 \ 0]^T$ and $\vec{E}_b(0) = [5 \ 5 \ 5]^T$, the attitude and position states were both stepped to:

$$Q_d = [-0.33 \ 0.22 \ -0.26 \ 0.88]^T$$

$$\text{and } \vec{E}_d = [7.5 \ 4 \ 3]^T \quad (27)$$

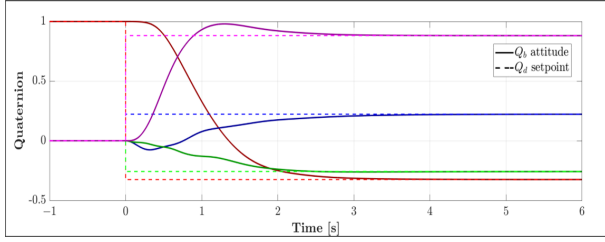


Fig. 4. Attitude step

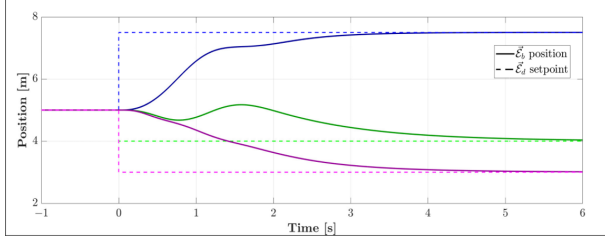


Fig. 5. Position step

Step responses for both attitude and position controllers are shown in Fig:4-5 respectively. Neither controllers induced saturation of any actuator for the step and both states settled to 95% in $t_{95} = 5.62$ [s]. The objective was dynamic setpoint tracking so; to test that, an orbital trajectory was generated and commanded as a state setpoint. The trajectory orbits a central point at 0.5 [Hz], completing one orbit in 120 [s], starting from $Q_b(0) = [1 \ 0]^T$ and $\mathcal{E}_b(0) = [5 \ 5 \ 5]^T$. Tracking of that trajectory with an oscillating quaternion attitude and inertial position setpoints is shown in Fig:6-7 respectively.

VII. CONCLUSION

The non-zero setpoint tracking goal for a quadcopter's state is achieved with the control structure(s) proposed. Only first order setpoints were defined in (10b) and (14b) which led to some phase lag in the resultant tracked trajectory. Commanding angular and translational velocity setpoints would diminish such an error. The combination of non-zero attitude and position setpoint tracking is an interesting result for a quadrotor helicopter; the proposed control solutions are expanded and improved upon in [15] wherein exponentially stabilizing control laws are investigated.

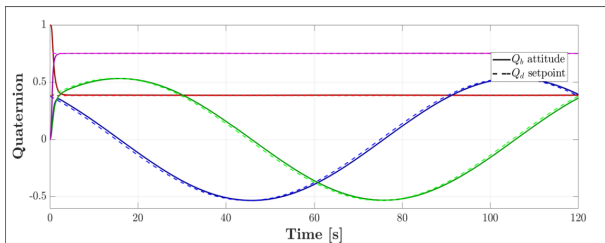


Fig. 6. Attitude trajectory

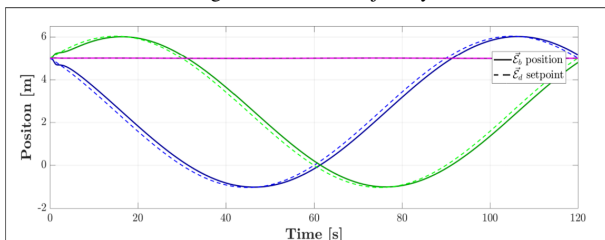


Fig. 7. Position trajectory

The final point of consideration is the application of a pseudo inversion control allocation in (21). Such an inversion, as described in [4], applies a least squares optimization to it's associated actuator matrix. The physical consequence of this is that the magnitude $|\vec{T}_{1 \rightarrow 4}|$ for thrust vectors to be commanded is minimized (*optimized*); meaning the control algorithm prioritizes pitching those thrust vectors with λ_i or α_i before changing their magnitudes with Ω_i propeller speed changes...

REFERENCES

- [1] P. Pounds, R. Mahony, P. Hynes and J. Roberts, "Design of a four-rotor aerial robot", In Friedrich, Werner & Lim, Patrick, Proceeding of Australasian conference on robotics and automation, pp. 145-150, Nov 2002.
- [2] P. Baldi, B. Mogens, P. Castaldi, N. Mimmo and S. Simani, "Adaptive FTC based on control allocation and fault accommodation for satellite reaction wheels", IEEE, Conference on control and fault-tolerant Systems, vol. 3, pp672-677, Sept 2016.
- [3] R. Kristiansen and P.J. Nicklasson, "Satellite attitude control by quaternion-based backstepping", IEEE, Trans. on control systems technology, vol. 17, pp227-232, Jan 2009.
- [4] T.A. Johansen and T.I. Fossen, "Control allocation - a Survey", Automatica, vol. 49, pp1087-1103, Jun 2013.
- [5] G.R. Gress, "Lift fans as gyroscopes for controlling compact VTOL air vehicles: overview and development status of oblique active tilting", American Helicopter Society Forum, vol 63, May 2007.
- [6] M. Ryll and H. Bulthoff and P. Robuffo Giordano, "Modelling and control of a quadrotor UAV with tilting propellers", IEEE, Conference on robotics and automation, pp4606-4613, May 2012. 2002.
- [7] M. Ryll and H. Bulthoff and P. Robuffo Giordano, "First flight tests of a quadrotor UAV with tilting propellers", IEEE, Conference on robotics and automation, pp295-302, Dec 2013.
- [8] A. Nematy and M.Kumar, "Modelling and control of a single Axis tilting quadcopter", American control conference, pp3077-3082, Jun 2014.
- [9] P.S. Gasco and Y. Al-Rihani, "Development of a dual-axis tilt rotorcraft UAV", Univ. Cranfield, department of Engineering, Nov 2012.
- [10] K. Shoemaker, "Quaternions", Univ. Pennsylvania, Department of computer and information science, 1987.
- [11] M. Bangura and R. Mahony, "Non-linear dynamic modeling for high performance control of a quadrotor", Australian robotics and automation association, Conference on robotics and automation, pp1-10, Dec 2012.
- [12] J. Brandt and M. Selig, "Propeller performance data at low Reynolds numbers", American institute of aeronautics and astronautics, Sciences meeting, vol. 49, pp 1-18, Jan 2011.
- [13] T. Thornton and J.B. Marion, "Classical dynamics of particles and systems", Thompson Brooks/Cole, 5th Edition, Ch 7, pp 228-289, 2003.
- [14] E. Boje, "Lyapunov stability analysis", Univ. Kwazulu natal, Department of electrical, electronic and computer engineering, Control Systems: ENEL4CN, Jan 2005, unpublished.
- [15] N. Von Klemperer, "Dual-axis tilting quadrotor aircraft", Univ. Cape Town, Department of Electrical Engineering, Oct 2017, in press.
- [16] D.D. Garanin, "Rotational motion of rigid bodies", Univ. New York, Department of engineering, Nov 2008.
- [17] F. van den Berg, A.P. Engelbrecht, "A study of particle swarm optimization particle trajectories", Univ. Pretoria, Department of computer science, Mar 2005.