

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 state setpoint tracking to a quadrotor helicopter; equipped with additional motors to solve the problem of a quadrotor's inherent underactuation. Introduction of overactuation results in a multibody, nonlinear system to be modeled 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 nonlinear state-space controller(s) are optimized iteratively using a swarm algorithm; producing a control suite that successfully follows a simple orbital trajectory, demonstrating non-zero setpoint tracking.

Index terms— control, nonlinear, overactuated, optimization, quadrotor

I. INTRODUCTION

Quadrotor helicopters, ostensibly published first in [1], are currently a popular control research field. This is due to recent advances in cheap embedded systems which can implement the autonomous vehicle's control loop. Quadrotor helicopters are inherently underactuated with only four controllable inputs (namely the four propeller's rotational speeds) to articulate a 6 degree of freedom (*DOF*) body; making the control problem ill-posed. Standard control solutions linearize the plant around the attitude's origin; thus reducing the system's controllable degrees of freedom.

What follows is a controller design which achieves non-zero attitude and position setpoint tracking for a quadrotor by introducing overactuation to the vehicle, without any linearization of the dynamic equations of motion. A higher level controller designs net torque and force inputs to be applied to the multibody vehicle about its center of motion. A lower level allocation rule then distributes that virtual control input amongst the actuator set to be implemented.

Overactuated control is mostly presented 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 actuators 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] uses quaternion based backstepping to track a satellite's orbital trajectory whilst making use of all available actuators via a pseudo-inversion allocation rule. A series of allocation schemes is surveyed in [4], most of which apply some form of matrix inversion to find explicit actuator positions.

Some work has been done on reconfigurable aerospace vehicles; the most obvious commercial example being the Bell/Boeing V-22 Osprey. Reconfigurable aerospace frames are, however, usually avoided due to their dynamic complexity. Dual-axis tilting capabilities are often incorporated into birotor aircraft; [5] presented a dual-axis tilting birotor design but the control algorithm applied constrained the actuation to a 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]. Similarly, [8] proposed a single-axis tilting quadrotor aircraft. Neither of these single-axis quadrotor papers consider the dynamic implications of reconfiguring the quadcopter mid-flight. One 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 the rotating propeller module.

The remainder of this paper is arranged as follows; the prototype's design and its 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 nonlinearities which stem from the prototype's structure and in-flight reconfiguration. Position and attitude controllers are derived in IV, together with an allocation rule. Each control law's coefficients are selected using a swarm optimization discussed briefly in V. Finally setpoint tracking is verified with simulations in VI and conclusions are subsequently drawn in VII.

II. DESIGN AND CONVENTIONS

Two Cartesian reference frames are used; 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 achieved with a quaternion operation:

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

Where $Q_b = [q_0 \ \vec{q}]^T$ is the attitude quaternion, Q_b^* is the quaternion complex conjugate and \otimes is the Hamilton product between two quaternions. Quaternion operands with zero scalar components are implied for the operation in (1). Quaternions and their caveats are discussed in [10].

The prototype's design extends the actuation suite of a regular quadcopter, such as [1], to an overactuated 12-*DOF* system by introducing eight additional actuator inputs.

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

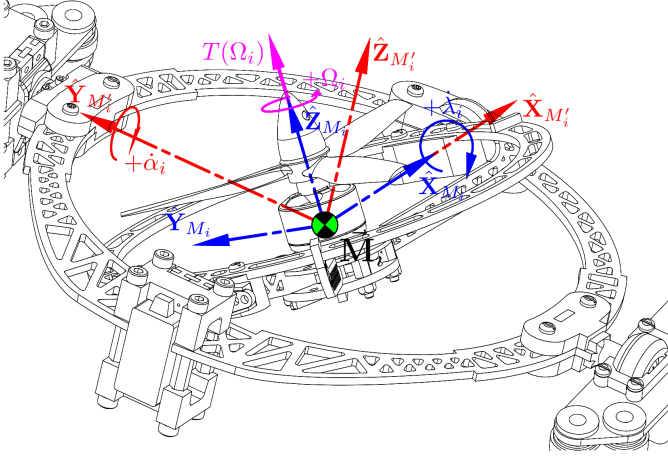


Fig. 1. Actuator motor module with vectored propeller thrust

A standard quadrotor comprises of four controllable inputs; those being each propeller's rotational speeds Ω_i for $i \in [1 : 4]$, which underactuates the control plant. The design here adds two revolute actuators to respectively pitch and roll *each* of the four motors and propellers about their \hat{X}_{M_i} and \hat{Y}_{M_i} axes, shown in Fig:1.

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

$$\vec{T}(\Omega_i) = C_T(J)\rho\Omega_i^2 D^4 \cdot \hat{Z} \in \mathcal{F}^{M_i} \quad (2a)$$

$$\vec{H}(\Omega_i) = C_H(J)\rho\Omega_i^3 D^5 (1/R\Omega_i) \cdot \hat{Z} \in \mathcal{F}^{M_i} \quad (2b)$$

Where J is the propeller's advance ratio, ρ is the air density taken at *standard temperature* and *pressure* and D is the propeller's diameter. Further details of BEM theory and a propeller's aerodynamics are presented in [11]. Thrust and drag coefficients used for the prototype were linearly interpolated from data in [12] to match physical tests applied.

Each of the four motor modules is aligned relative to the body frame's center \hat{O}_b as per Fig:2. The 1st and 3rd motor modules have their \hat{X}_{M_1/M_3} axes aligned with the body frame's positive and negative $\pm\hat{X}_b$ axis directions respectively. While the 2nd and 4th modules have their \hat{Y}_{M_2/M_4} axis directions aligned with the positive and negative $\pm\hat{Y}_b$ body frame's axes respectively. There is an orthogonal 90° rotation about the \hat{Z}_b axis between each motor module.

Motors 1 and 3 have clockwise propeller rotations $\Omega_1, \Omega_3 > 0$, whereas motors 2 and 4 have anti-clockwise propeller rotations $\Omega_2, \Omega_4 < 0$. An instantaneous configuration of the multibody system is fully described by the four propeller's rotational speed and the eight 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 vectored counterpart for:

$$\vec{T}_i = Q_{M_i}^* \otimes \vec{T}(\Omega_i) \otimes Q_{M_i} \in \mathcal{F}^b \quad (4a)$$

$$\vec{H}_i = Q_{M_i}^* \otimes \vec{H}(\Omega_i) \otimes Q_{M_i} \in \mathcal{F}^b \quad (4b)$$

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 are simply:

$$\vec{F}_\mu = \sum_{i=1}^4 \vec{T}_i(\Omega_i, \lambda_i, \alpha_i) \in \mathcal{F}^b \quad (5a)$$

$$\vec{\tau}_\mu = \sum_{i=1}^4 \vec{L}_i \times \vec{T}_i(\Omega_i, \lambda_i, \alpha_i) + \vec{H}_i(\Omega_i, \lambda_i, \alpha_i) \in \mathcal{F}^b \quad (5b)$$

Each thrust vector $\vec{T}_i(\Omega_i, \lambda_i, \alpha_i)$ is a combined function of produced thrust (2a), redirected by (4a). The net torque $\vec{\tau}_\mu$ comprises of differential torque arm cross-products of input thrusts \vec{T}_i , each acting at a length \vec{L}_i in Fig:2, and the propeller's aerodynamic torque $\vec{H}_i(\Omega_i, \lambda_i, \alpha_i)$ being a similar combined function of (2b) and (4b). Aerodynamic drag $\vec{\tau}_H \triangleq \sum \vec{H}_i(\Omega_i, \lambda_i, \alpha_i)$ is to be *compensated* for in feedback, but could otherwise be used as a controllable input.

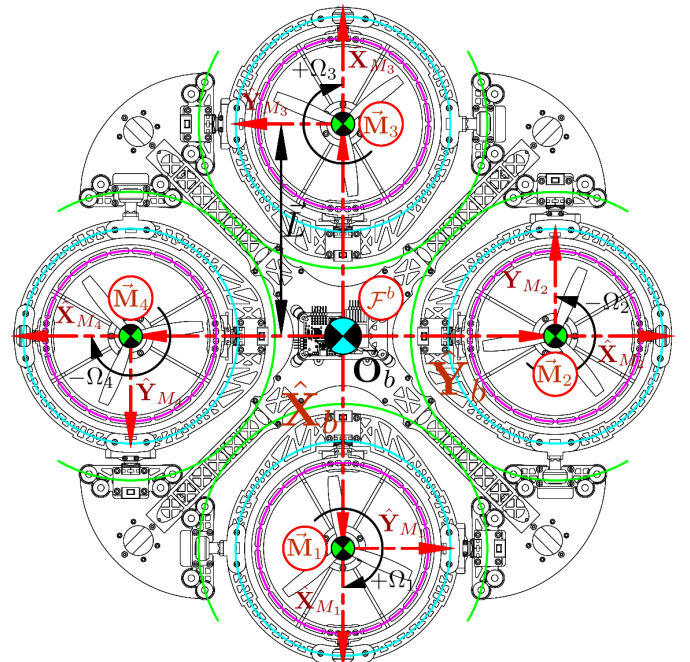


Fig. 2. Body frame axes layout

III. DYNAMICS

For a *rigid* 6-DOF body, the Lagrangian dynamics of which are derived from first principles in [14], having 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, [15], for the attitude's differential equations of motion are:

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

$$\dot{\vec{\omega}}_b = J_b^{-1}(-\vec{\omega}_b \times J_b \vec{\omega}_b + \vec{\tau}_\mu) \in \mathcal{F}^b \quad (6b)$$

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

$$\dot{\vec{\mathcal{E}}}_I = Q_b^* \otimes \vec{v}_b \otimes Q_b \in \mathcal{F}^I \quad (7a)$$

$$\dot{\vec{v}}_b = m_b^{-1}(-\vec{\omega}_b \times \vec{v}_b + m_b \vec{G}_b + \vec{F}_\mu) \in \mathcal{F}^b \quad (7b)$$

With 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 added 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 [14], from which this paper is adapted, but are excluded here for the sake of brevity.

In summary; each of the thirteen independent bodies are connected with only one permissible degree of rotational freedom between each sequential structure. Fig:3 shows the exploded view of a single motor module; each body with its own rotational velocity relative to \mathcal{F}^b , resulting from the actuation rates $\dot{\lambda}_i$ and $\dot{\alpha}_i$. Non-Newtonian frames of each rotating sub-body made quantifying the net multi-body's response to rotational and translational motion highly complicated. As per [13], partial derivatives of the multibody's Lagrangian with respect to each actuator's path co-ordinates produces the *generalized forces*, in this case torques $\vec{\tau}_b(u)$, acting on the system as a response to actuation input $u \in \mathbb{U}$.

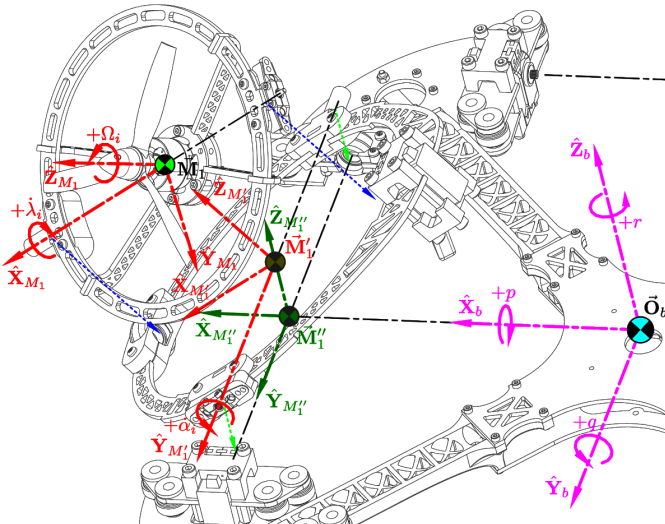


Fig. 3. Multi-body overview

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 its principle frame, but the body is rotated by θ about some unit axis \hat{n} . The effective transformed inertia is:

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

Where $R_{\hat{n}}(\theta)$ is the rotation matrix about that unit axis, [15]. 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

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

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

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

$$\vec{a} \times \vec{b} \equiv [\vec{a}]_{\times} \vec{b} \triangleq \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \vec{b} \quad (8d)$$

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 $\vec{\tau}_b(u)$ 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. Equation (6b) for angular acceleration treats the net aerodynamic torque $\vec{\tau}_H$ from (5b) and the multibody nonlinearity responses $\vec{\tau}_b(u)$ as terms to be compensated for in feedback:

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

State estimates \hat{u} for actuator positions are used in (9) to incorporate transfer functions of each individual actuator. Lastly, $\vec{\tau}_g$ is a gravity torque term caused by any disparity in the *rotated* motor modules' gravitational centroids.

IV. CONTROLLER DESIGN

The control block is divided into two components; a higher level setpoint tracking controller which 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 explicit actuator positions $u \in \mathbb{U}$ which physically command the control input \vec{v}_c as per the actuator's *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-5b) and their aerodynamics (2a-2b).

Only first order setpoints are commanded such that the state's setpoint is $\vec{x}_d = [\vec{\mathcal{E}}_d \ Q_d]^T$. 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; $\lim_{t \rightarrow \infty} \vec{x}_e \rightarrow \vec{0}$. Basic Proportional-Derivative (PD) controllers are used here, because of their simplicity and efficient optimization; [14].

A. Attitude Control Derivation

The attitude plant, (6) and extended to (9), is independent of the position loop and hence stabilized first. A quaternion setpoint Q_d leads to a *multiplicative* quaternion error, [15]:

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

An angular velocity setpoint is defined with respect to the attitude setpoint's frame, or that $\vec{\omega}_d \in \mathcal{F}^d$. The 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 PD attitude controller with compensation feedback terms. That 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 with actuator response compensation estimates for $J(\hat{u})$ and $\vec{\tau}_b(\hat{u})$. Those estimate errors from $\hat{u} \rightarrow u$ are not considered here but are incorporated into adaptive backstepping in [14].

$$\begin{aligned} \vec{\tau}_d = & \underbrace{K_p \vec{q}_e + K_d \vec{\omega}_e}_{\text{independent}} \\ & + \underbrace{\vec{\omega}_b \times J_b(\hat{u}) \vec{\omega}_b + \vec{\tau}_b(\hat{u}) - \vec{\tau}_H - \vec{\tau}_g}_{\text{compensation}} \in \mathcal{F}^b \end{aligned} \quad (11)$$

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, [16]. A Lyapunov function candidate (LFC) for the attitude's trajectory *error*, such that $V_1(\vec{0}) = 0$, is:

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

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 components:

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

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:

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

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

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

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

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

B. Position Control Derivation

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

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

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 both 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

Positions for the twelve actuators in (3) must be solved for using 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 $B(\vec{x}, \hat{u}, t)$:

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

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 $\vec{T}_i \in \mathcal{F}^b$ for $i \in [1 : 4]$ 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 four thrust vectors:

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

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

Then, using the pseudo-inversion of (20), the 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 $\vec{T}(\Omega_i)$ is entirely in the \hat{Z}_{M_i} direction. Then (22b) simplifies for the i^{th} motor module with an orthogonal σ_i rotation about the \hat{Z}_b axis:

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

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 $|\vec{T}_i|$ is 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:

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

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; its 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 [14]. The optimized diagonal coefficient matrices for both PD controllers were respectively:

$$K_p = \text{diag}(3.568, 5.270, 6.070) \quad \text{and} \quad K_d = \text{diag}(9.015, 11.485, 20.183) \quad (26a)$$

$$A_p = \text{diag}(2.417, 3.156, 2.590) \quad \text{and} \quad A_d = \text{diag}(3.479, 3.385, 3.870) \quad (26b)$$

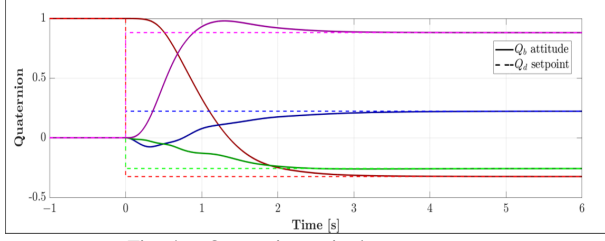


Fig. 4. Quaternion attitude step response

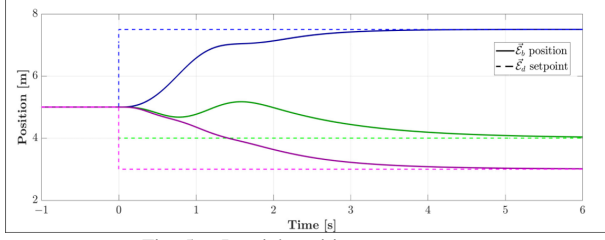


Fig. 5. Inertial position step response

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 $\vec{E}_b(0) = [5 \ 5 \ 5]^T$ m, the attitude and position states were both stepped to respective setpoints:

$$Q_d = [-0.33 \ 0.22 \ -0.26 \ 0.88]^T \quad \text{and} \quad \vec{E}_d = [7.5 \ 4 \ 3]^T \text{ m} \quad (27)$$

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 $\vec{E}_b(0) = [5 \ 5 \ 5]^T$ m. 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 [14] wherein exponentially stabilizing and adaptive control laws are investigated.

The final point of consideration is the application of a pseudo-inverse control allocation in (21). Such an inversion, as described in [4], applies a least squares optimization to its associated actuator matrix. The physical consequence of this is that the magnitude of $|\vec{T}_{1 \rightarrow 4}|$ for thrust vectors to be physically commanded is minimized (*optimized*); meaning the control algorithm prioritizes pitching thrust vectors with λ_i or α_i before varying their magnitudes with changes to propeller speeds Ω_i .

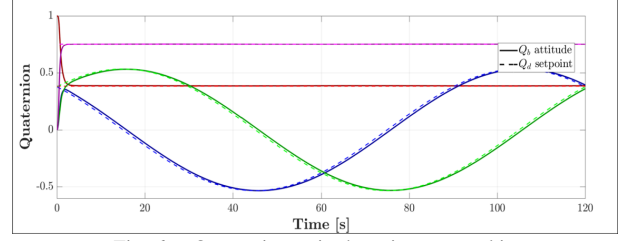


Fig. 6. Quaternion attitude trajectory tracking

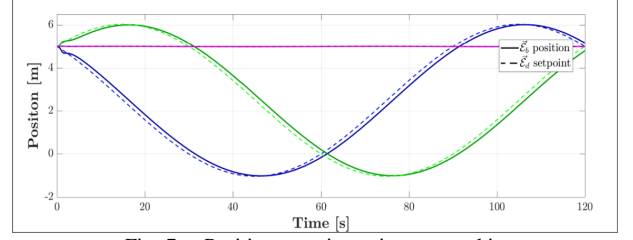


Fig. 7. Position timepoint trajectory tracking

REFERENCES

- [1] P. Pounds, R. Mahony, P. Hynes and J. Roberts, "Design of a four-rotor aerial robot", *proc. Australasian conference on robotics and automation (ARAA)*, vol. 3, 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 (SysTol)*, vol. 3, pp.672–677, Sept 2016.
- [3] R. Kristiansen and P.J. Nicklasson, "Satellite attitude control by quaternion-based backstepping", *IEEE, trans. Control systems technology*, vol. 17, pp.227–232, January 2009.
- [4] T.A. Johansen and T.I. Fossen, "Control allocation - a Survey", *Automatica*, vol. 49, pp.1087–1103, June 2013.
- [5] G.R. Gress, "Lift fans as gyroscopes for controlling compact VTOL air vehicles: overview and development status of oblique active tilting", *proc. 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, International conference on robotics and automation (ICRA)*, vol. 14, pp.4606–4613, May 2012.
- [7] M. Ryll and H. Bulthoff and P. Robuffo Giordano, "First flight tests of a quadrotor UAV with tilting propellers", *IEEE, International conference on robotics and automation (ICRA)*, vol. 15, pp.295–302, December 2013.
- [8] A. Nemati and M.Kumar, "Modelling and control of a single Axis tilting quadcopter", *IEEE, American control conference (ACC)*, pp.3077–3082, June 2014.
- [9] P.S. Gasco and Y. Al-Rihani, "A novel actuation concept for a multi rotor UAV", Springer Netherlands, *Journal of intelligent & robotic systems*, vol. 74, pp.173–191, April 2014.
- [10] K. Shoemaker, "Animating rotations with quaternion curves", *The Singer Company, SIGGRAPH*, vol. 3, no. 3, pp.245–254 July 1985.
- [11] M. Bangura and R. Mahony, "Non-linear dynamic modelling for high performance control of a quadrotor", *proc. Australasian robotics and automation association (ARAA)*, Conference on robotics and automation, pp.1–10, Dec 2012.
- [12] J. Brandt and M. Selig, "Propeller performance data at low Reynolds numbers", *trans. American institute of aeronautics and astronautics (AIAA)*, 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] N. Von Klemperer, "Dual-axis tilting quadrotor aircraft", *Univ. Cape Town, Department of Electrical Engineering*, Oct 2017, in press.
- [15] J.B. Kuipers, "Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality", *Univ. Press Princeton*, pp.127–143, Sept 1999.
- [16] M. Vidyasagar, "Nonlinear systems analysis", *Society for industrial and applied mathematics*, 2nd ed., pp.135–269, 2002.
- [17] F. van den Berg, A.P. Engelbrecht, "A study of particle swarm optimization particle trajectories", *Elsevier, Journal of Information Sciences*, vol. 176, pp.937–971, April 2006.