End-to-End Full-State Dynamics of Generic Rotorcrafts

Author: Hyung Ju Suh

1 Introduction

Rotors (propellers and motors) provide various attractive characteristics as a main method of propulsion and control for autonomous Unmanned Aerial Vehicles (UAV). Compared to conventional aircrafts that utilize control surfaces (ailerons, rudders), such benefits include more maneuverability at low speeds, and ease of design / manufacturing. These unmanned rotorcrafts are popularized by the quadrotor, a simple vehicle with four rotors that show a good degree of maneuverability. Recent designs have also shown varieties of the quadrotors, with additional rotors attached to the body (hexacopter, octocopter). However, these designs are mostly limited by the fact that the rotors are all in-plane, and the vehicle's commanded force and attitude states are coupled - thus unable to achieve full controllability. For instance, quadrotors cannot move forward unless its pitch is tilted.

In order to overcome this limitation, several different design solutions exist. The first one is adding more actuation to the rotors, exemplified by the variable-pitch rotor [1,2]. These can succeed in decoupling the direction of thrust from the vehicle's attitude state by adding more actuation. Another solution is to use multiple rotors at different positions and orientations, then to use a simple in-plane solution [3]. In this work, we explore the second solution by generalizing the rotorcraft's dynamics with any different number of rotors, with different position and orientations. Understanding the full-state dynamics as a function of the geometrical variables allows optimizing the vehicle's geometry for controllability, more control authority, and etc.

In our work, the description of the dynamics is designed to be hardware-friendly, and aimed towards low-cost hardware platforms. Some characteristics of some platforms might include the following.

1. The rotors are actuated through brushless DC (BLDC) motors
2. The motors are controlled by pulse-position modulation (PPM), commonly known as "throttle" to individual ESC units that are available for purchase
3. The ESCs only support throttle in one direction (i.e. thrust are not reversible)

With these assumptions, we formulate a dynamic system with control input being the actual throttle signals. Many dynamic modeling of rotorcrafts assume rotor's angular velocity [4,5], or force and torque [7] as control inputs. However, these models fail to describe the first-order behavior of rotor velocity or force/torque commands, as in reality, it is not possible to command these quantities instantaneously without considering the transient change. By incorporating the first-order motor model, we not only describe the first-order change in angular velocity, but achieve an end-to-end dynamic system that is more hardware-friendly.

2 Full-State Dynamics

2.1 Simplified Abstraction: Force and Moment

The goal of describing full-state dynamics of a system is to observe how a system's state will evolve given its current state and a control action. In most 3-D formulations, we must consider a non-linear formulation of dynamics such that

$$\dot{\vec{x}} = f(\vec{x}, \vec{u}) \tag{1}$$

where $\vec{x}$ is the state, and $\vec{u}$ is the control input. Foremost, let us consider a formulation given in terms of an "abstracted" control action, by assuming that the body can generate force and moment in every direction. We will denote the combined force and moment as a screw wrench $\vec{F} = [\vec{f} \quad \vec{\tau}]^T \in \mathbb{R}^6$, where force is $\vec{f} \in \mathbb{R}^3$, and moment is $\vec{\tau} \in \mathbb{R}^3$. Throughout the notation, we will use w as the world frame, and b as the body frame. The superscript denotes the observer frame and the subscript denotes the frame being observed. Then, we have the following states $\vec{x} \in \mathbb{R}^{13}$, and $\vec{u} \in \mathbb{R}^6$.

$$\vec{x} = \begin{pmatrix} \vec{p}_b^w \\ \vec{v}^b \\ \vec{q}_b^w \\ \vec{\omega}^b \end{pmatrix} \qquad \vec{u} = \begin{pmatrix} \vec{f}^b \\ \vec{\tau}^b \end{pmatrix} \tag{2}$$

In our state, $\vec{p}_b^w \in \mathbb{R}^3$ is the position of the vehicle body frame observed from the world frame, $\vec{v}^b \in \mathbb{R}^3$ is the velocity of the vehicle body frame observed from the body frame, $\vec{q}_b^w \in \mathbb{R}^4$ is the orientation of the vehicle body frame observed from world frame, and $\vec{\omega}^b \in \mathbb{R}^3$ is the angular velocity (body rate) of the vehicle body frame observed from the body frame. If we define the body frame as the vehicle's center of mass (COM), the dynamics of this system can be described as

$$f(\vec{x}, \vec{u}) = \frac{d}{dt}\begin{pmatrix} \vec{p}_b^w \\ \vec{v}^b \\ \vec{q}_b^w \\ \vec{\omega}^b \end{pmatrix} = \begin{pmatrix} \mathbf{R}(\vec{q}_b^w)\vec{v}^b \\ \frac{1}{m}\vec{f}^b - [\vec{\omega}^b]_\times \vec{v}^b + \mathbf{R}^T(\vec{q}_b^w)\vec{g}^w \\ \frac{1}{2}\Omega(\vec{\omega}^b)\vec{q}_b^w \\ \mathbf{I}^{-1}\left[\vec{\tau}^b - [\vec{\omega}^b]_\times \mathbf{I}\omega^b\right] \end{pmatrix} \tag{3}$$

In Eq.(3), $\mathbf{R}(\vec{q}_b^w)$ is the quaternion-derived rotation matrix in $SO(3)$, $[\vec{\omega}^b]_\times$ is the skew-symmetric cross-product representation of body rate in $\mathfrak{so}(3)$, and $\Omega(\vec{\omega}^b)$ is the matrix representation of the Hamiltonian between the quaternion $[q_w, q_x, q_y, q_z]$ and the extended angular velocity vector $[0, \omega_x, \omega_y, \omega_z]$ that is given by

$$\Omega(\vec{\omega}^b) = \begin{pmatrix} -[\vec{\omega}^b]_\times & \vec{\omega}^b \\ -(\vec{\omega}^b)^T & 0 \end{pmatrix} \tag{4}$$

Finally, $\mathbf{I}$ is the inertial tensor of the body evaluated around its center of mass, which is a symmetric matrix

$$\mathbf{I} = \begin{pmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{xy} & I_y & -I_{yz} \\ -I_{xz} & -I_{yz} & I_z \end{pmatrix} \tag{5}$$

For many rotorcrafts, it is possible to impose additional simplification to $\mathbf{I}$ by leveraging the symmetry of the rotorcraft. For instance, a diagonal $\mathbf{I}$ matrix will often suffice for a quadcopter since it is symmetric in the xz and yz plane.

2.2 Propeller Forces to Body Wrench: Geometry Parameters

Since our aim is to achieve a generic representation of the rotor's end-to-end full-state dynamics, it is crucial to have good parametrization of the rotor geometries, and a true representation of hardware control inputs. First, we model how a single rotor's angular velocity contributes to thrust and torque from that rotor, and then observe how all of these thrusts and torques contribute to the body wrench. In a rotorcraft with N rotors, we can first think of the angular velocity values of all the rotors, represented as a vector

$$\vec{\phi} = \begin{pmatrix} \phi_1 & \phi_2 & \dots & \phi_N \end{pmatrix}^T \tag{6}$$

Given these values, it is possible to obtain the thrust and torque contributions of each rotor using a simple aerodynamic model.

$$\vec{f}_{prop} = C_L \vec{\phi} \circ \vec{\phi} = C_L \vec{\phi}^2 \qquad \vec{\tau}_{prop} = \kappa \vec{f}_{prop} + J\dot{\vec{\phi}} \tag{7}$$

In this notation, C_L is the lift coefficient of the propeller, $\circ$ is the Hadamard (element-wise) product operator, denoting a vector where each elements of $\vec{\phi}$ are squared. For notational convenience we will denote this vector as $\vec{\phi}^2$. In the torque term, there are two contributions - the first comes from propeller's drag, and it can be modeled using $\kappa = C_D/C_L$, the propeller-specific drag-to-lift coefficient. The second comes from motor's inertial torque, where J is the combined inertia of the rotor and the propeller. We assume that the rotorcraft has the same motor and propellers and leave J as a scalar parameter, but it may be extended to different motor combinations where C_L, κ, J are vectors.

2.2.1 Fixed Rotors

Now we can consider the geometry of each rotors, represented as three matrices:

$$\mathbf{K}_f = \begin{pmatrix} \vec{k}_1 & ... & \vec{k}_N \end{pmatrix} \qquad \mathbf{K}_\tau = \begin{pmatrix} \vec{k}_1^* & ... & \vec{k}_N^* \end{pmatrix} \qquad \mathbf{P} = \begin{pmatrix} \vec{p}_1 & ... & \vec{p}_N \end{pmatrix} \tag{8}$$

where $\vec{k}_1$ represents the rotor normal vector in the direction of thrust, and $\vec{k}_i^*$ represents the rotor normal vector in the direction of torque. If the produced torque pseudovector is in the opposite direction of the thrust, then $\vec{k}_i^* = -\vec{k}_i$, and vice versa. In easier hardware terms, CW blades have $\vec{k}_i = \vec{k}_i^*$, and CCW blades have $\vec{k}_i = -\vec{k}_i^*$. This geometrical configuration inherently requires two-fold parametrization because of the rotor's spinning direction, and how the rotor type (Fig. 1). Finally, $\vec{p}_i$ represents the position of each rotor relative to the center of mass. As a result, we have $\mathbf{K} \in \mathbb{R}^{3\times N}$ and $\mathbf{P} \in \mathbb{R}^{3\times N}$. Given these geometry parameters, the body wrench is connected to the propeller forces and torques by

$$\vec{F} = \begin{pmatrix} \vec{f}^b \\ \vec{\tau}^b \end{pmatrix} = \begin{pmatrix} \mathbf{K}_f \vec{f}_{prop} \\ (\mathbf{P} \times \mathbf{K}_f)\vec{f}_{prop} + \mathbf{K}_\tau \vec{\tau}_{prop} \end{pmatrix} = \begin{pmatrix} \mathbf{K}_f C_L \vec{\phi}^2 \\ (\mathbf{P} \times \mathbf{K}_f) C_L \vec{\phi}^2 + \mathbf{K}_\tau \kappa C_L \vec{\phi}^2 + \mathbf{K}_\tau J \dot{\vec{\phi}} \end{pmatrix} \tag{9}$$

In Eq.(9), the force term is a simple vector summation of all the thrust terms. The torque term is consisted of three components. The first term is created by thrust, the second one is gyroscopic torque created by propeller drag, and final term is gyroscopic torque created by inertia.

Figure 1: Clockwise (CW) and Counter-Clockwise (CCW) propellers

2.2.2 Variable Pitch

In some applications, the propellers are augmented by variable pitch. This means the geometry of the rotors can be controlled, restricted by some mechanical linkages. In most variable pitch linkages, a single rotary linkage is used to change the orientation of the rotor. Therefore, the $\mathbf{P}$ matrix stays the same, while the $\mathbf{K}$ matrices are now changing as a function of some commanded angle parameter. In order to represent rotation, it is most convenient to use the axis-angle representation since it allows for representing both the fixed axis of rotation (in this case, this is a fixed parameter), as well as the magnitude of rotation (which is a part of the control input).

To see how the $\mathbf{K}$ matrix can be described as a function of axis-angles, we first define another matrix that denotes the rotation axis of each rotary axis:

$$\mathbf{N} = \begin{pmatrix} \vec{n}_1 & ... & \vec{n}_N \end{pmatrix} \tag{10}$$

Now given a vector of rotated angles $\vec{\theta} = [\theta_1, ..., \theta_N]$, we have a matrix with every column representing the 3-dimensional axis-angle vector.

$$\mathbf{\Theta} = \mathbf{N}\text{diag}(\vec{\theta}) = \begin{pmatrix} \theta_1\vec{n}_1 & ... & \theta_N\vec{n}_N \end{pmatrix} \tag{11}$$

Now the $\mathbf{K}$ matrix can be obtained using

2.3 Motor Dynamics

In hardware, the motors are commanded by pulse-width modulation (PWM), where the applied voltage to the motors are controlled by a duty cycle $\alpha \in [0, 1]$ (This assumes that the ESC cannot reverse the direction of motors). Incorporating the motor model directly from duty cycle has several advantages. The first obvious advantage is that it achieves a true end-to-end dynamical model by using the actual control input. Mathematically, two other advantages exist. By indirectly controlling the rotor angular velocities, we have an affine dynamics model that can deal with the motor's inertial torque term. Also, it allows modeling the mechanical time-response of the motor, which is a more realistic assumption since motors cannot jump from one velocity to another. The motor dynamics are presented as follows: the torque applied from the motor is a function of the current, by the torque constant K_t. In turn, this current is connected to the net voltage across the motor by the motor's internal resistance.

$$J\frac{d\phi}{dt} = K_t(u)i = \frac{K_t(u)V_{applied}}{R} \tag{12}$$

The net voltage applied is commanded voltage minus the back EMF voltage. Assuming that the torque constant is the same with the back EMF constant, we have a first-order model.

$$\frac{d\vec{\phi}}{dt} = \frac{K_t(u)}{RJ}\left(V_{cc}\vec{u}_i - K_t\vec{\phi}_i\right) \tag{13}$$

We make several comments about this model. First, $K_t(u)$ is usually assumed constant in most models, but this assumption does not often hold. In order to achieve better modeling, $K_t(u)$ can be validated experimentally by varying the throttle signal and observing steady-state behavior. Another comment is that it is possible to make this model second-order by incorporating motor inductance L. However, since L is usually small for brushless motors often used in drones, we leave out this term for now.

2.4 End-to-End Full-State Dynamics

Now we consider an end-to-end dynamics model where the control inputs are formulated directly in terms of the ESC-commanded PWM signals, and the states are augmented with rotor velocities in order to make the system affine. In this case we have

$$\vec{x} = \begin{pmatrix} \vec{p}_b^w \\ \vec{v}^b \\ \vec{q}_b^w \\ \vec{\omega}^b \\ \vec{\phi} \end{pmatrix} \qquad \vec{u} = \begin{pmatrix} \alpha_1 \\ ... \\ \alpha_N \end{pmatrix} \tag{14}$$

Where $\vec{x} \in \mathbb{R}^{13+N}$ and $\vec{u} \in \mathbb{R}^N$, and $\vec{u}$ is consisted of PWM values commanded to the ESC. Given these states and control inputs, the full dynamics of the system is modeled as

$$\dot{\vec{x}} = f(\vec{x}, \vec{u}) = \frac{d}{dt}\begin{pmatrix} \vec{p}_b^w \\ \vec{v}^b \\ \vec{q}_b^w \\ \vec{\omega}^b \\ \vec{\phi} \end{pmatrix} = \begin{pmatrix} \mathbf{R}(\vec{q}_b^w)\vec{v}^b \\ \frac{1}{m}\mathbf{K}_f C_L\vec{\phi}^2 - [\vec{\omega}^b]_\times\vec{v}^b + \mathbf{R}^T(\vec{q}_b^w)\vec{g}^w \\ \frac{1}{2}\Omega(\vec{\omega}^b)\vec{q}_b^w \\ \mathbf{I}^{-1}\left[(\mathbf{P}\times\mathbf{K}_f)C_L\vec{\phi}^2 + \mathbf{K}_\tau\left[\frac{K_t}{R}(V_{cc}\vec{u} - K_t\vec{\phi}) - C_D\vec{\phi}^2\right] - [\omega^b]_\times\mathbf{I}\vec{\omega}^b\right] \\ \frac{1}{J}\left[\frac{K_t}{R}(V_{cc}\vec{u} - K_t\vec{\phi}) - C_D\vec{\phi}^2\right] \end{pmatrix} \tag{15}$$

Using affine notation where we can linearly separate the terms coupled with the control input $\vec{u}$, we can rewrite the dynamics as

$$\dot{\vec{x}} = \mathbf{f}(x) + \mathbf{g}(x)\vec{u} \tag{16}$$

where functions $\mathbf{f}(x)$ and $\mathbf{g}(x)$ are defined as

$$\mathbf{f}(x) = \begin{pmatrix} \mathbf{R}(\vec{q}_b^w)\vec{v}^b \\ \frac{1}{m}\mathbf{K}_f C_L \vec{\phi}^2 - [\vec{\omega}^b]_\times \vec{v}^b + \mathbf{R}^T(\vec{q}_b^W)\vec{g}^w \\ \frac{1}{2}\Omega(\vec{\omega}^b)\vec{q}_b^w \\ \mathbf{I}^{-1}\left[(\mathbf{P}\times\mathbf{K}_f)C_L\vec{\phi}^2 + \mathbf{K}_\tau[\kappa C_L \vec{\phi}^2 - \frac{K_t^2}{R}\vec{\phi}] - [\omega^b]_\times \mathbf{I}\vec{\omega}^b\right] \\ -\frac{K_t^2}{RJ}\vec{\phi} \end{pmatrix} \qquad \mathbf{g}(x) = \begin{pmatrix} \mathbf{0}^{10\times N} \\ \frac{K_t V_{cc}}{R}\mathbf{I}^{-1}\mathbf{K}_\tau \\ \frac{K_t V_{cc}}{RJ}\mathbf{I}^{N\times N} \end{pmatrix} \tag{17}$$

3 Obtaining Parameters

In practice, many different parameters are required for our dynamics. Some of them are purely geometrical, while others must be measured using different techniques. All the parameters required for the dynamics are written in Table.1

Table 1: Parameters required for Dynamics

Parameters	Description	Units	Quantity
Inertial Parameters			
m	Mass of the rotorcraft	kg	scalar
$\mathbf{I}$	Inertia of rotorcraft	$kg \cdot m^2$	matrix
Geometrical Parameters			
$\mathbf{K}_f$	Rotor normal vectors (positive force)	None	matrix
$\mathbf{K}_\tau$	Rotor normal vectors (positive torque)	None	matrix
$\mathbf{P}$	Rotor position vectors	m	matrix
Aerodynamic Parameters			
C_L	Propeller Lift Coefficient	None	scalar
κ	C_D/C_L, drag-to-lift ratio	None	scalar
Motor Parameters			
K_t	Motor torque constant & back-EMF constant	$V \cdot s/rad$	scalar
R	Motor internal resistance	Ω	scalar
J	Rotor and Propeller Inertia	$kg \cdot m^2$	scalar
V_{cc}	Rail (Battery) Voltage	V	scalar

3.1 Inertial Parameters

3.2 Motor & Aerodynamic Parameters

Since we assume that we use the same rotors in all quadcopters, it suffices to run a single test on a single motor-propeller configuration. In the motor equation, we impose a first-order dynamics model such that

$$J\dot{\phi} = \frac{K_t}{R}(V_{cc}\alpha - K_t\phi) \tag{18}$$

where we know the values of α, the commanded variable. It is also trivial to measure internal motor resistance R and rail voltage V_{cc} using a multimeter. (Additionally, R is usually a given parameter from the manufacturer as well). Finally, the thrust and torques from the propeller are connected to the angular velocity by

$$f = C_L\phi^2 \qquad \tau = \kappa C_L \phi^2 + J\dot{\phi} \tag{19}$$

The testbed for this system requires a force and torque loadcell to measure f and τ, a multimeter to measure V_{cc} and R, and finally, a tachometer / rotary encoder to measure ϕ.

3.2.1 Steady-State

In steady-state, $\dot{\omega} = 0$. Therefore, the equations simplify to

$$V_{cc}u - K_t\phi = 0 \tag{20}$$

since we know V_{cc}, u, and have measurement of ϕ, we can obtain K_t quite easily. It is also possible to obtain an experimental measurement of $K_t(u)$ Additionally, C_L and κ can also be easily obtained through observing the steady-state versions of Eq.(19).

$$C_L = \frac{f}{\phi^2} \qquad \kappa = \frac{\tau}{C_L\phi^2} \tag{21}$$

Thus through a simple steady-state analysis, we can obtain the values of K_t, C_L, κ.

3.2.2 Transient Fit

Finally, in order to measure J, we will need to numerically differentiate the obtained values of ϕ to obtain $\dot{\phi}$. Then we can analyze the transient curve to calculate

$$J = \frac{K_t}{R\dot{\phi}}\left(V_{cc}\alpha - K_t\phi\right) \tag{22}$$

4 Simulation and Examples: Quadcopter

4.1 Quadcopter

The most successful and widely used rotorcraft is the quadcopter, with $N = 4$ and the four rotors with the same force normals. In order to provide torque stability, the torque normals are alternating in different direction. Since we have a simple matrix representation of the rotor normals and positions, we only need to fit measure these parameters accordingly. In the quadcopter case, all the rotors point normal to the xy plane, with two alternating signs for gyroscopic torque. Thus the $\mathbf{K}_f, \mathbf{K}_\tau$, and $\mathbf{P}$ can be written as

$$\mathbf{K}_f = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix} \quad \mathbf{K}_\tau = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & -1 \end{pmatrix} \quad \mathbf{P} = \frac{1}{\sqrt{2}}\begin{pmatrix} r & r & -r & -r \\ r & -r & -r & r \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{23}$$

A simulation is designed with some parameters obtained through methods mentioned in Section 3, and some parameters that have just been guessed. Additionally, since this is not a control problem, we can only verify that certain control actions will lead to certain behavior of the body frame.

4.1.1 Equal Thrust

The simplest control command we can consider is that of equal magnitude, $\vec{u} = [0.48, 0.48, 0.48, 0.48]$. With this command, the quadcopter is expected to make a linear movement in the z direction. With this control action, simulated for $50s$ with $100Hz$, we can verify this behavior in Fig.2.

We can observe that indeed, x and y stays the same, while the z position and velocity increases. If we observe the motor velocity term, we can indeed see the first-order behavior until it reaches steady-state solution. The time constant is mainly decided by motor resistance R and inertia J. (With large propellers, it is common to observe that due to this first-order behavior, controllers based on angular velocity commands become unstable). Such first-order behavior can also be observed in the position and velocity term. Due to the effect of gravity, the position falls for a certain time before the lift value saturates, and the propeller goes upwards.

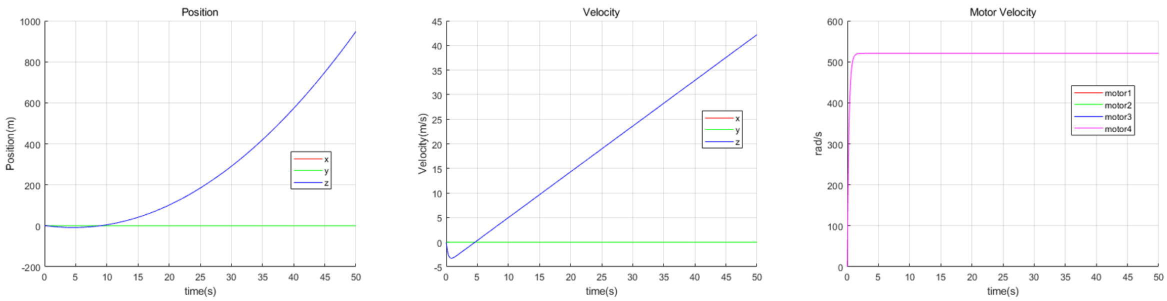

Figure 2: Given $u_1 = u_2 = u_3 = u_4 = 0.48$, A. Position vs. time. B. Velocity vs. time. C. Motor Velocity vs. time

In reality, there is another first-order behavior on the velocity value, since aerodynamic drag comes into play and is a main factor in limiting the top-velocity of the rotorcraft. However, this term is ignored for now because we assume the quadrotor moves in relatively slow-velocities, where aerodynamic drag can be ignored.

4.1.2 Pitch Command

In order to change the pitch, we give more thrust to the front two propellers, to cause the rotorcraft to rotate in pitch and 'flip' backwards. Since we still want relatively less z displacement, we can keep the magnitudes the same while having a small variation, $\vec{u} = [0.48, 0.48, 0.46, 0.46]$. The pitch is mainly dominated by torque created by the lift forces, and the resulting simulation is described in Fig.3.

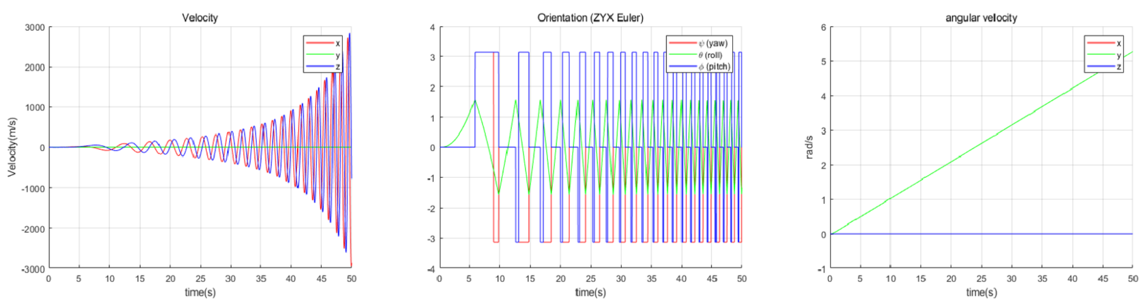

Figure 3: Given $u = [0.46, 0.46, 0.48, 0.48]$ A. Velocity vs. time. B. Orientation vs. time. C. Angular Velocity vs. time

We can see that the angular velocity is in the y-direction, and therefore the pitch is changing. The quaternions are converted and graphed into ZYX Euler angles for better interpretation: the Euler-angles are a bit more complicated since it is internally rotated, but we can confirm the angular velocity changes in the y direction. As a result, the velocity oscillates as it is defined in body-frame and is affected by centrifugal force.

4.1.3 Yaw Command

In this example, a command of $\vec{u} = [0.48, 0.46, 0.48, 0.46]$ is given. By having alternating torques, the purpose is to verify that the quadrotor changes yaw successfully by using gyroscopic torque. The resulting figure is illustrated in Fig.4.

We can see that since we are using ZYX convention, the gyroscopic torque indeed generates a linear increase in body angular velocity, and thus a quadratic increase in vehicle orientation.

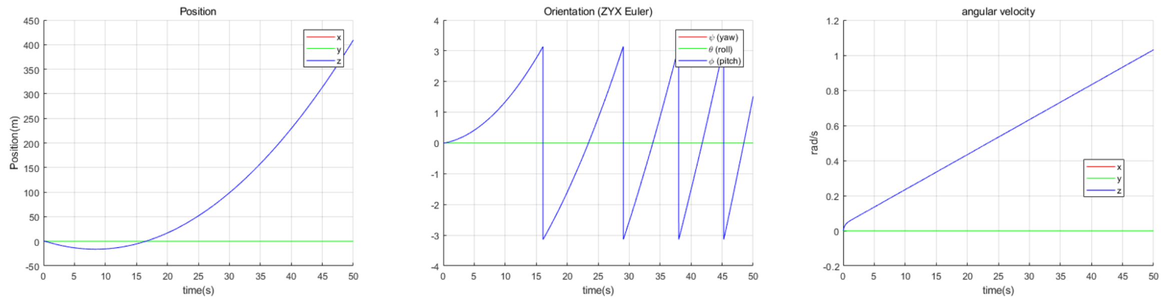

Figure 4: Given $u = [0.48, 0.46, 0.48, 0.46]$ A. Position vs. time. B. Orientation vs. time. C. Angular Velocity vs. time

5 Control Laws

5.1 Force-Torque Map-based PD Control

6 Conclusion and Future Works

In this work, the dynamics of a generic rotorcraft was formulated with a compact parametrization of the rotor's 3D placement. By including the motor dynamics, we successfully simulated the first-order behavior of force and torques, and their effects on vehicle's position and orientation. Figuring out this dynamics is useful for the following future works.

6.1 Design Optimization

Since we know how the vehicle will behave according to the dynamics, it is now possible to optimize the vehicle's rotor placements according to some criteria. This can be done by keeping $\mathbf{K}_f, \mathbf{K}_\tau$, and $\mathbf{P}$ as variables and employing optimization algorithms such as gradient descent, and quadratic programming. An example is given in [3].

6.2 Vehicle Control

Since we know the dynamics of the quadrotor, we can employ control algorithms to control the vehicle state. Many control strategies employ a simple PD control [3], where partial states are controlled. For instance, if we have external position measurements, we can use control actions to have the vehicle converge into that measurement. In this case, since we know how the full-state will evolve according to our control action, this allows us to study the system further, including controllability analysis based on $\mathbf{K}_f, \mathbf{K}_\tau, \mathbf{P}$, and reachability analysis on the state spaces.

6.3 Control Authority and Allocation

Finally, we aim to formulate control authority and allocation problems with these geometry parameters as well. Control authority is a measure of how well the vehicle can exert actions in some direction. This is especially important for underactuated systems such as quadcopters - for instance, a quadcopter is much more likely to be maneuverable in the z direction than it is in the x-direction. Additionally, Control allocation deals with overactuated systems with multiple solutions to a single control action. For instance, in an omnicopter [3], there are multiple solutions to how to get the same force and torque. Therefore, we need to decide a good solution for full-state trajectory generation.

References

[1] M. Cutler, N.K. Ure, B. Michini, J.P. How, "Comparison of Fixed and Variable Pitch Actuators for Agile Quadrotors", AIAA Guidance, Navigation, and Control Conference, Aug. 08, 2011

[2] M. Cutler, J.P. How, "Analysis and Control of a Variable-Pitch Quadrotor for Agile Flight", Journal of Dynamic Systems, Measurement, and Control, 137(10), Jul. 01, 2015

[3] D. Brescianini, R. D'Andrea, "Design, Modeling, and Control of an Omni-Directional Aerial Vehicle", IEEE International Conference on Robotics and Automation (ICRA), May. 16-21, 2016

[4] R. Beard, "Quadrotor Dynamics and Control", Brigham Young University Scholars Archive, 2008

[5] T. Luukkonen, "Modeling and Control of Quadcopter", Aalto University School of Science, Aug. 22, 2011

[6] P. Wang, Z. Man, Z. Cao, J. Zheng, Y. Zhao, "Dynamics Modeling and Linear Control of Quadcopter", IEEE International Conference of Advanced Mechatronic Systems, Nov. 30, 2016

[7] B.L.Stevens, F.L.Lewis, E.N.Johnson, "Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems", 3rd ed., John Wiley & Sons, 2016