Modeling and Control of Coaxial UAV with Swashplate Controlled Lower Propeller

Richard Lee, Koushil Sreenath, Sebastian Scherer

CMU-RI-TR-16-33

June 2016

Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Abstract

There is a growing interest in the design and control of coaxial vehicles for the purposes of autonomous flight. These vehicles utilize two, contra-rotating propellers for generating thrust and swashplates for generating pitch and roll. In this report, we present a novel coaxial design in which both upper and lower rotors are contained within a ducted fan, the speeds of both rotors are independently controlled, and the lower rotor's cyclic pitch is controlled through a swashplate.

Based on this design, a simple dynamic model was developed with unique force and moment generation equations. Given this model, we are able to map desired force and moment values to the control inputs capable of producing them. Afterwards, position and attitude control were implemented over this nonlinear dynamic model in simulation, such that the vehicle was able to recover from poor initial conditions and follow desired trajectories. As demonstrated by the examples presented in this report, position control results in simulations with low max percent overshoot and reasonable settling times. These results prove promising for the implementation of position and attitude control on our physical system.

1 Introduction

Coaxial vehicles present potential key advantages over conventional quadrotors for the implementation of autonomous flight. The use of two larger motors over four smaller ones allows the vehicle to more efficiently draw energy, leading to longer flight times. Additionally, the placement of propellers directly above one another rather than on the corners of the vehicle allow for a smaller overall width. The design and control of many small, indoor coaxial vehicles such as the muFly2 and CoaX, both developed at ETH Zurich [2][6], and the PetiteLion developed by researchers at the National University of Singapore [8], have demonstrated the ability for various control algorithms to allow for successful hover control in indoor environments. However, vehicles of this scale also pose the additional problems of faster dynamics [4] and issues with low Reynolds number[2]. Larger coaxial vehicles include the Rotary Wing MAV at the University of Maryland [12] and, most notably, the CRDFH at the Nanjing University of Aeronautics and Astronautics [11], which are both working towards greater autonomy and improved flight performance.

Because our vehicle design differs from the aforementioned coax's, there exists a need to model our vehicle dynamics and implement useful control algorithms. While the use of Newton-Euler methods are widely adopted to model the non-linear differential equations of rigid body dynamics, the computation of force and moment generations have been found to differ among designs and vary in levels of complexity. For example, factors such as downwash caused by the upper rotor, gyroscopic forces, and cross-coupling between yaw and altitude or pitch and roll controls are highly dependent on the physical design of the vehicle itself. Many vehicles may also include stabilizer bars to offer greater stability to the system at the expense of faster dynamics. The addition of components such as stabilizer bars can be thought of as modules that offer their own individual contributions to the total force and moment inputs experienced [1].

Our vehicle weighs 2.76 kg and utilizes a 16 inch diameter ducted fan in which both the upper and lower rotor are contained. Doing so eliminates issues associated with downwash and wakes that would have otherwise been generated at the tips of the propeller blades. The swashplate has been specifically designed to avoid issues associated with cross-coupling, such that a single servo input can be expected to independently control roll while the other independently controls pitch; although, some cross-coupling between the inputs is still present. Lastly, unlike most smaller coax's, our design foregoes the use of a stabilizer bar for more responsive dynamics, which has been considered in other designs [4] [6].

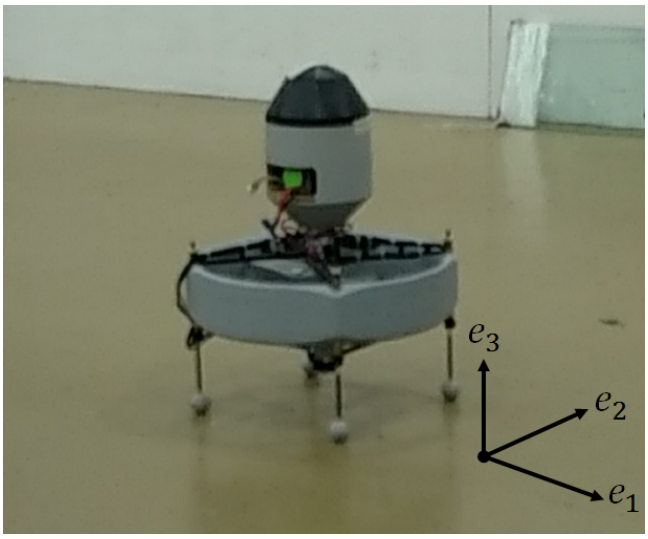

Figure 1: Coaxial vehicle in world coordinate frame

By removing issues associated with cross-coupling and the addition of a stabilizer bar module, we are able to produce simpler force and moment equations than those presented in other dynamic models. As a result, we arrive at model that allows us to directly map desired force and moment values generated by a position or attitude controller to the control inputs of our vehicle.

As varied as individual coaxial designs may be for the purposes of achieving autonomous flight, the implementation of control methods is just as diverse. These control methods include robust H_∞ control implemented on both the muFly and CRDFH [4] [11], traditional PID control [8], model reference sliding mode control [9], and fuzzy logic control [7]. Although many control methods were available to us, we chose to use traditional PID control for the purpose of simplicity.

2 State Space Model of the System in SE(3)

Our state space model describes the rigid body motion of any vehicle in SE(3) space. It is based on Newton-Euler methods and defines our governing equations of motion. The state space matrix is defined as:

$$X = \begin{bmatrix} x \\ v \\ R \\ \Omega \end{bmatrix}, \tag{1}$$

where x is a 3x1 vector representing rectilinear position and v is a 3x1 vector representing rectilinear velocity. R represents a 3x3 rotational matrix from the body-fixed frame to the world frame and Ω represents a 3x1 rotational velocity vector.

Given this matrix, its associated derivative is defined as:

$$\dot{X} = \begin{bmatrix} v \\ -ge_3 + \frac{1}{m} RF_b \\ R\hat{\Omega} \\ J^{-1}(M - \Omega \times J\Omega) \end{bmatrix}, \tag{2}$$

where m is the mass of the body, J is the moment of inertia tensor, f is the force vector imparted to body by the actuators, and M is the moment vector imparted to the body. Lastly the terms e_1, e_2, and e_3 represent the x, y, and z vectors of the world frame, respectively. The vector e_3 points upwards, and the value $\hat{\Omega}$ is the hat map of Ω, where the hat map function is defined in the Appendix.

3 Force and Moment Generation

Our particular system has four control inputs that contribute to generation of the forces and moments imparted on the body. Two of these inputs control the angular speeds of the upper and lower rotor through electronic speed controllers (ESCs) and the other two control servos that orient the swashplate of the lower propeller, allowing for pitch and roll. As mentioned in the Introduction, our physical system has been designed in such a way that it minimizes cross coupling effects and consists of less modules, and these effects are assumed to be negligible for the derivation of force and moment equations.

For the following equations, a_{1lo} and b_{1lo} represent the longitudinal and lateral flapping angles of the lower propeller with respect to the body frame and are directly the result of the servo inputs given to the swashplate. The force exerted on the body by the propellers alone can be defined as:

$$F_b = \begin{bmatrix} T_{lo}[-cos(b_{1lo})sin(a_{1lo})] \\ T_{lo}sin(b_{1lo}) \\ T_{up} + T_{lo}[-cos(b_{1lo})cos(a_{1lo})] \end{bmatrix}. \tag{3}$$

T_{lo} and T_{up} are the magnitudes of thrust exerted by the lower and upper propellers, respectively, and these two terms are dependent on the rotational speeds of the propellers. Although there are different ways to model the relationship between thrust and propeller speed such as was done in [4], the following aerodynamic model was adopted for the sake of simplicity, as in [1],

$$T_{up,lo} = k_{up,lo}^{lift} \Omega_{up,lo}^2. \tag{4}$$

$\Omega_{up,lo}$ refers to the rotational speed of either the upper or lower propeller, and $k_{up,lo}^{lift}$ is assumed to be a constant proportionality factor that is mainly governed by the shape of the propellers and determined experimentally. Our particular experimental results and the apparatus for determining these values are presented in the Appendix.

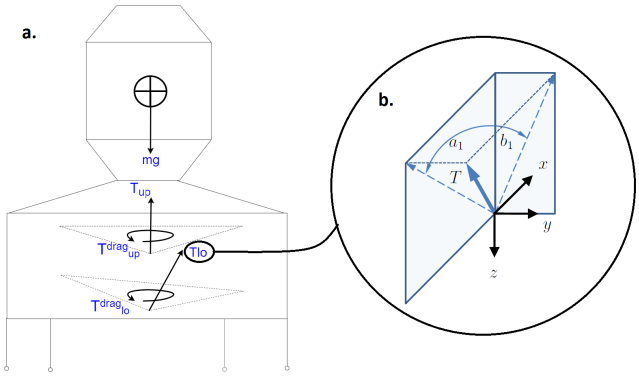

Figure 2: (a) Free body diagram of vehicle and (b) projection of flapping angles onto the body frame [1]

The moment forces exerted on the body are assumed to be mainly governed by the blade flapping stiffness of the propellers and the orientation of the thrust vector of the lower propeller, which is displaced from the vehicle's COM:

$$M = \begin{bmatrix} b_{1lo}[\cancel{\frac{e_{com}}{r} m_{bld}(\Omega_{lo}^2 r)^2}] + F_y d \\ a_{1lo}[\cancel{\frac{e_{com}}{r} m_{bld}(\Omega_{lo}^2 r)^2}] - F_x d \\ T_{up}^{drag} + T_{lo}^{drag} \end{bmatrix}. \tag{5}$$

Although, conventionally, the values for the x and y components, M_x and M_y, are dependent on factors such as the rotor blade radius r, the combined hinge offset e_{com}, and the mass of the blade m_{bld}, these terms can be ignored by assuming that the blades are rigid ($k_{eqv} = e_{eqv} = 0$) and rigidly connected to the rotating axle ($e = 0$), which is believed to be a fair assumption for our particular vehicle design. This assumption makes the hinge offset term $e_{com} = 0$ by ignoring a flapping moment, such that M_x and M_y are only dependent on the distance offset d between the lower propeller and COM:

$$e_{com} = e + e_{eqv}. \tag{6}$$

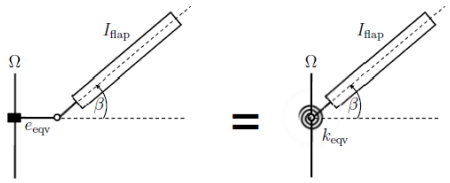

Figure 3: Hinge offset equivalent to blades flapping spring stiffness is zero [1]

The yaw moment M_y is defined by the difference of rotor speeds between the top and bottom propellers which leads to a difference of drag torques according to the equation below:

$$T^{drag}_{up,lo} = k^{drag}_{up,lo}\Omega^2_{up,lo}, \tag{7}$$

where $k^{drag}_{up,lo}$ is assumed to be a constant proportionality factor relating rotational speed to drag torque and can be determined experimentally. Although, for the purposes of our simulations, we had not determined these coefficient values, they can be related to $k^{lift}_{up,lo}$ values given that the shape of the propeller's airfoil is well known and documented.

4 Position and Attitude Control

Since the COM of the coaxial UAV is below the lower rotor, the swashplate of the lower rotor controls both the x and y components of the input force vector, F_x and F_y, but also generates values for M_y and M_x. Both orientation and position are linked to the swashplate control inputs, and they cannot be controlled simultaneously. In order to control rectilinear position, a desired force vector is provided through PD control and the vehicle's orientation is ignored. The resulting moment vector was left unrestricted, meaning the vehicle could become unstable if moment values became too large. The following equation defines the desired force value in the world frame:

$$F_d = k_x e_x + k_v e_v + mge_3 - m\ddot{x}_d. \tag{8}$$

The variable k_x is a proportional gain on the rectilinear position of the body, and the variable k_v is a derivative gain on the velocity of the body. The rectilinear position and velocity errors are defined below, where x_d and $\dot{x}_d$ represent the desired or reference position and velocity vectors, respectively:

$$e_x = x - x_d, \tag{9}$$

$$e_v = v - \dot{x}_d. \tag{10}$$

In a similar manner to force control, attitude control is implemented by generating a desired moment value. Generating values for M_x and M_y as defined in (5) necessitates specific values for F_y and F_x, respectively. The value for M_z is generated by a difference in motor speeds between the lower and upper propellers, and the sum of the thrusts exerted by the two propellers is set to the weight of the coax, mg. The moment control equation is then:

$$M_d = -k_R e_R - k_\Omega e_\Omega + \Omega \times J\Omega - J(\hat{\Omega} R^T R_d \Omega_d - R^T R_d \dot{\Omega}_d). \tag{11}$$

The terms R_d, Ω_d, and $\dot{\Omega}_d$ represent the desired or reference rotation matrix, angular velocity, and angular acceleration of the vehicle, respectively. k_R is a gain on the error in the rotational matrix and k_Ω is a gain on the error in rotation rate. The error values themselves are defined by the equations below:

$$e_R = \frac{1}{2}(R_d^T R - \check{} R^T R_d), \tag{12}$$

$$e_\Omega = \Omega - R^T R_d \Omega_d. \tag{13}$$

Where the $\vee$ accent in (12) represents the vee map function as defined in the Appendix.

5 Simulation of the System

Implementing both position and attitude control in a MATLAB simulation required first finding all relevant system properties of our real-world UAV, and then generating desired trajectories for which our vehicle would follow. A full list of these system properties can be found in the Appendix. Desired trajectories were generated as individual, time-dependent polynomials in the x, y, and z direction. These trajectories provided a desired position, velocity, and acceleration for each time step.

The most novel aspect of the resulting MATLAB simulations is the ability to directly map desired force and moment values, as dictated by equations (8) and (11), to desirable control inputs. This is possible because of the simplicity of our force and moment generation equations compared to those developed for other coaxial UAVs. By taking the three individual equations in (3), relating the force vector to our four control inputs, and the equation for M_z in (5), we are provided with four equations for four unknowns. Then using MATLAB's symbolic toolbox and system solver, approximate solutions were obtained for each of these unknowns, the accuracy of which will be demonstrated in the following example simulations. Note that although there are two additional equations in (5) for calculating M_x and M_y, the system is not overconstrained because these equations are not independent from those of F_y and F_x.

Although many simulation experiments were performed, only two are presented in this report as examples.

The first simulation demonstrates position control to a fixed location with a desired rectilinear velocity and acceleration of all zeroes. The coaxial UAV was provided a poor initial position, the value of which is listed in the Appendix. Position error is taken as the norm of errors in the x, y, and z directions. The vehicle experiences a max percent overshoot (taken as the norm of overshoots in x,y and z position) of 0.165 m with a settling time of 3.49 seconds (taken as 5% of the final value) when given an initial total distance offset of 3.91 m.

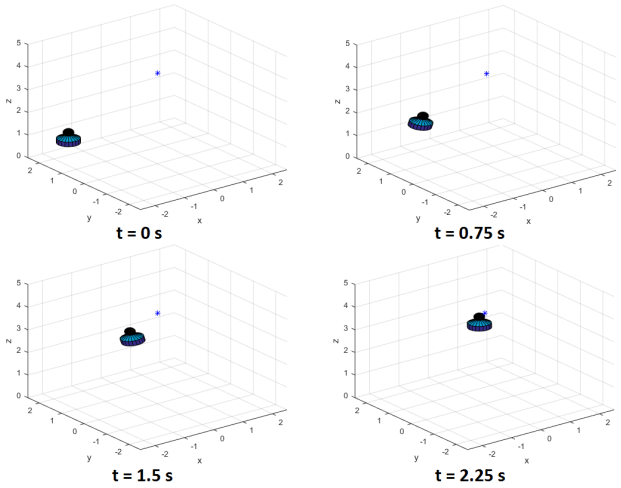

Figure 4: Animation of vehicle over time

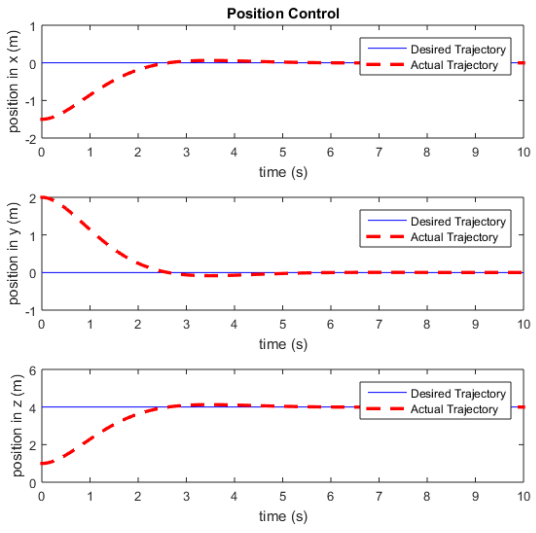

Figure 5: Trajectory following in x, y, and z

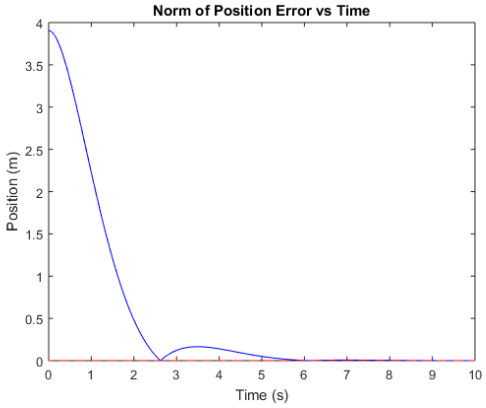

Figure 6: Norm of Position Error over time

As mentioned above, given a series of desired forces and moments (in this case, M_z is zero at all times), the desired angular velocity values for the propellers and swashplate angles a_{1lo} and b_{1lo} were calculated.The values for these control inputs were then used to recalculate the force vector and this vector is compared to the original desired force in Fig. 9. The inverse mapping force is the result of the recalculated force vector, which almost perfectly matches the original desired force vector, demonstrating the validity of our mapping between force and moment and control inputs.

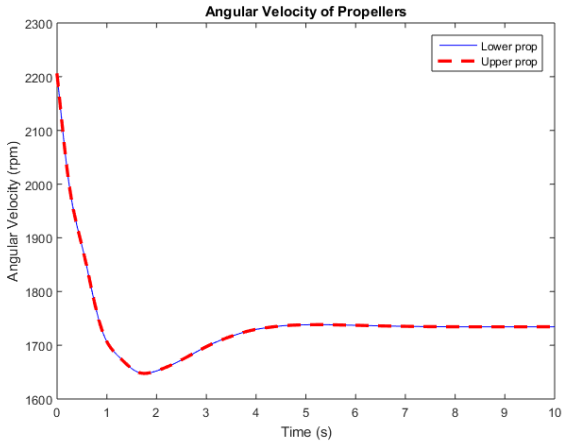

Figure 7: Desired angular velocity of propellers

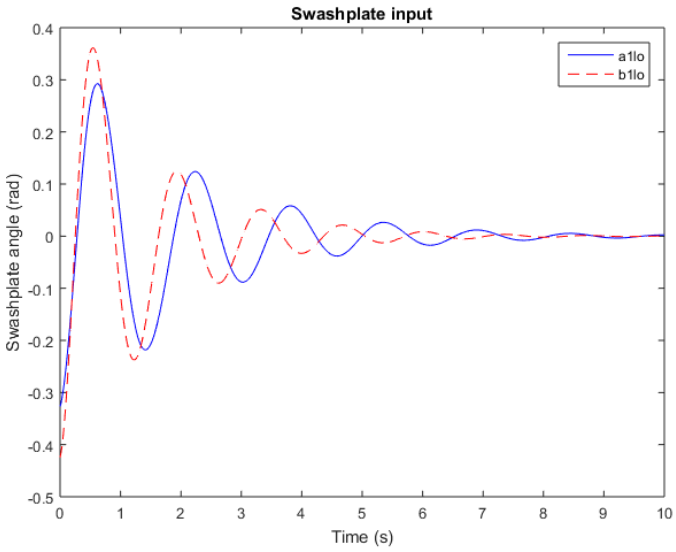

Figure 8: Desired swashplate angles for pitch and yaw

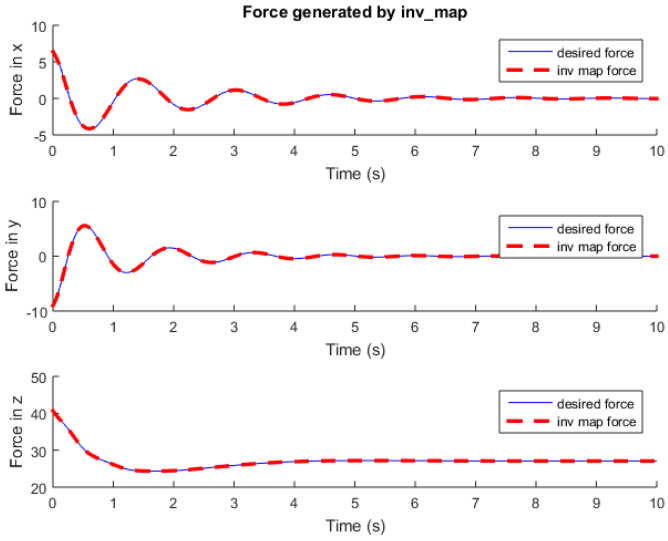

Figure 9: Desired force values for control of UAV

The second simulation demonstrates position control to a time-varying, desired trajectory with non-zero velocity and acceleration values, dictated by time-dependent polynomials in the x, y and z direction of at least third order. The system was also provided poor initial conditions with regards to initial position and orientation, as specified in the Appendix. The position error retains the same general shape as the previous simulation, but is generally of lower magnitude. The desired positions converge quickly towards the specified trajectory.

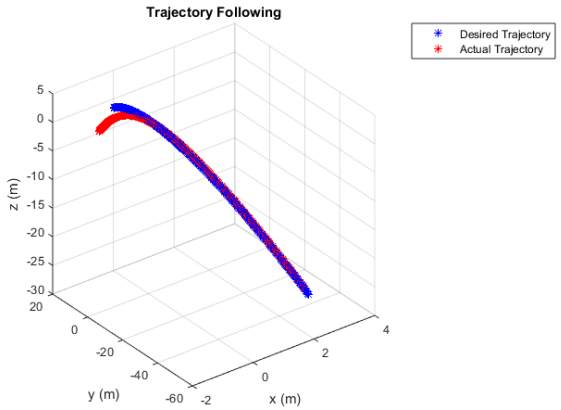

Figure 10: Desired trajectory vs Actual trajectory

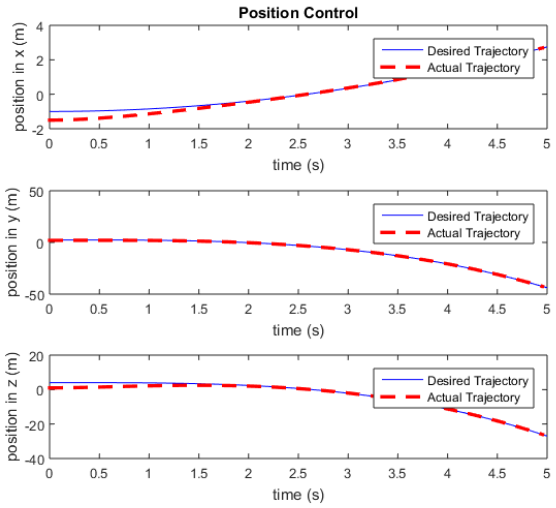

Figure 11: Trajectory following in x, y, and z

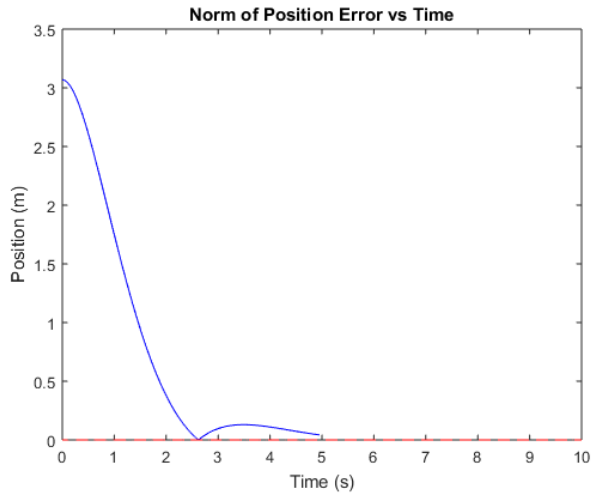

Figure 12: Norm of Position Error over time

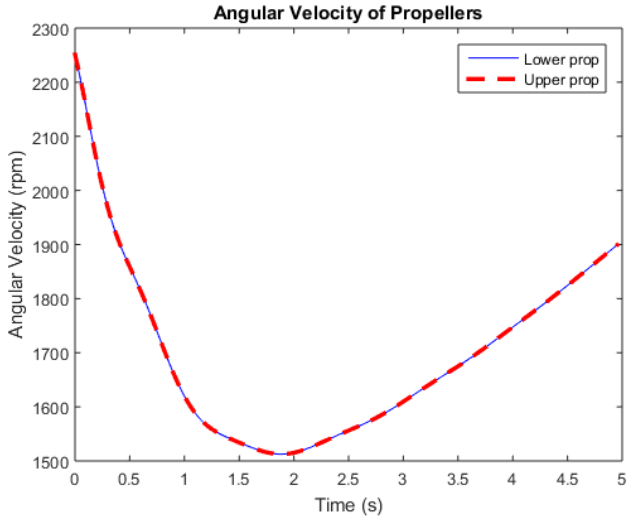

Figure 13: Desired angular velocity of propellers

It is clear from Fig. 14 that the swashplate angles differ greatly from the previous simulation but never reach extreme values. Also unlike the first simulation, there is a more prominent discrepancy between the desired force and the force produced by the inverse mapping to control inputs, which is more obvious initially in Fig. 15. Fortunately, this discrepancy is still small, and for the majority of the experiment, negligible.

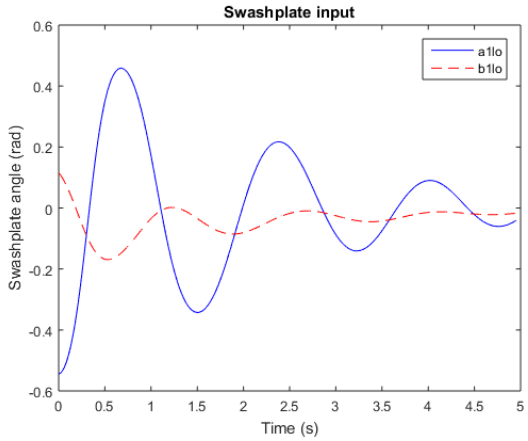

Figure 14: Desired swashplate angles for pitch and yaw

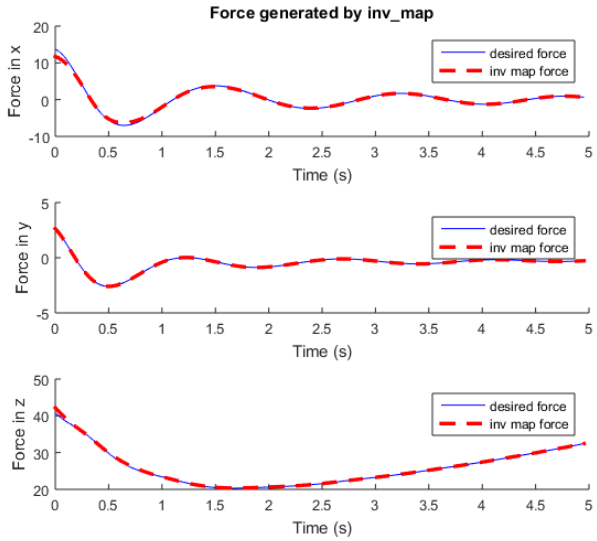

Figure 15: Desired force values for control of UAV

When implementing attitude control, there are often larger discrepancies and discontinuities between the actual desired force and moment governed by our controllers and those produced by the values assigned to our control inputs through inverse mapping. This may be because solutions to the moment equations are not directly calculated, but are related to forces necessary to cause those moments before calculating control inputs.

6 Conclusions and Perspectives

Based on these results, it can be seen that the dynamic model is successful in computing desired forces and moments based on a specific controller and then determining the rotor speeds and swashplate angles required to achieve those forces and moments. For the position controller in Simulation 1 in particular, the controller was able to apply forces to bring the vehicle to a desired position with little overshoot. In arriving at this position, the swashplate angle was initially large in magnitude in order to orient the vehicle towards its desired position, but oscillated and decreased in magnitude when nearing that position. Additionally, the vehicle also required higher rotor speeds when further from its desired position. Rotor speeds and swashplate angles were within desirable ranges for both simulations, but large gains could result in unacheivable speeds or irrecoverable orientations. Without being able to account for those conditions, this simulation's functionality is somewhat limited because it is unable to directly determine what controller gain values are physically possible. This issue could be addressed in future work.

Overall, however, these simulations are useful in understanding what physical changes are required of our system for position and attitude control. They are able to directly determine what previous simulations in literature have not, the swashplate angles and rotor speeds required for specific force and moment values. Moreover, the controllers themselves were able to follow desired trajectories given useful gain values. Based on these results, given a properly implemented controller on our physical system, we expect our vehicle to behave in a similar manner to the motion simulated in MATLAB when performing some form of trajectory following. However, further physical testing will be required in order to consolidate differences between the real-world system and the dynamic model, as flight behavior remains to be experimentally validated.

Appendix

1. The hat map function $\hat{}$ is defined as:

$$\hat{x} = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}, \tag{14}$$

where $x = [x_1; x_2; x_3]$ is a 3x1 vector in $\mathbb{R}^3$. The inverse of the hat map is referred to as the *vee* map $\vee$ and maps a 3x3 matrix to its corresponding 3x1 vector.

2. The following apparatus was used to estimate coefficient of lift values for our chosen propellers. Each propeller was attached to the motor, and the thrust generated caused the sliding bar to be displaced from its original position by a fixed distance. A spring of known spring constant k connected the sliding bar to its base, and was used to relate the distance displaced to a thrust value created by each propeller. Finally, a tachometer measured the rpm speed of each propeller when a piece of reflective tape was attached, and the rpm values were correlated to the thrust produced in order to approximate coefficients of lift.

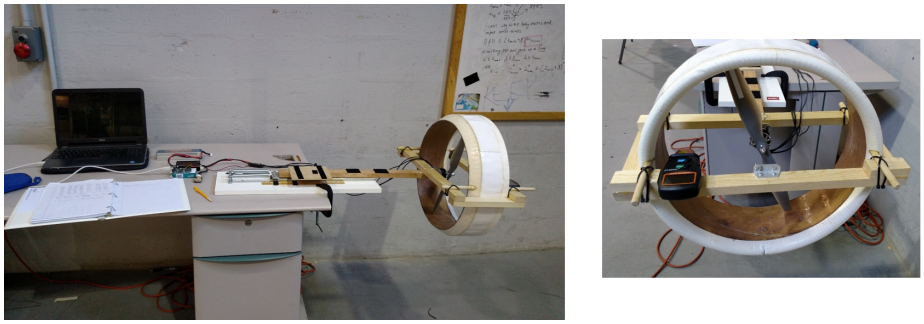

Figure 16: Apparatus for coefficient of lift

The resulting values for the coefficient of lift for propellers 1 and 2 were 5.036×10^{-6} and 3.83×10^{-6}, respectively, even though they were both the same shape. For the sake of simplicity, the coefficient of lifts for both propellers in our simulations were set to 4.5×10^{-6}, a number close to the average of these two values.

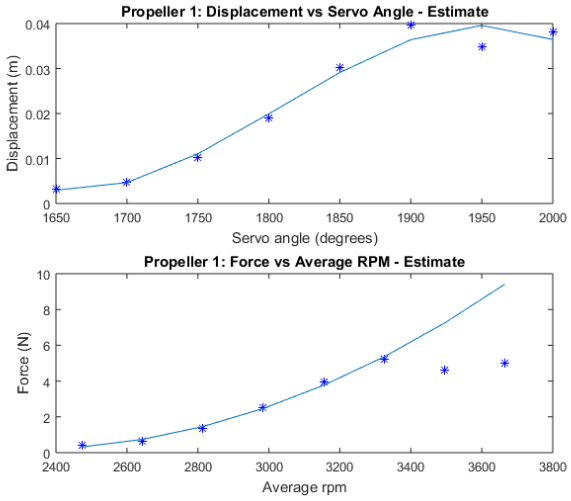

Figure 17: Test results for determining k_{up}^{lift}

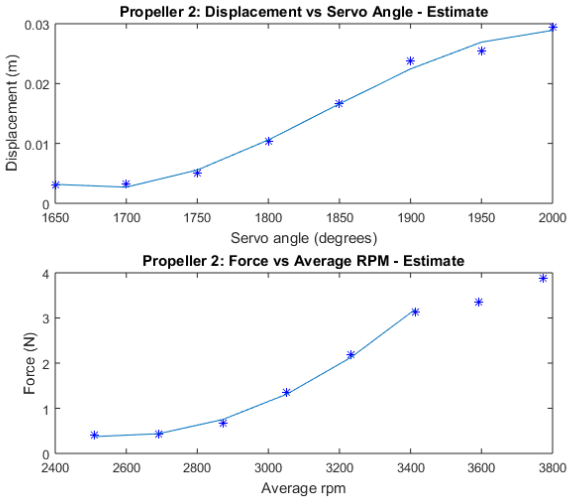

Figure 18: Test results for determining k_{lo}^{lift}

3. The following system properties are associated with the vehicle as a whole and constant among simulations:

Name	Symbol	Value	Unit
mass	m	2.76	kg
moment of inertia tensor	J	$\begin{bmatrix} .0736 & 0 & 0 \\ 0 & .097355 & 0 \\ 0 & 0 & .0732 \end{bmatrix}$	$kg \cdot m^2$
gravitational acceleration	g	9.81	$kg\frac{m}{s^2}$
coef. of lift upper rot.	k_{up}^{lift}	4.5×10^{-6}	
coef. of lift lower rot.	k_{lo}^{lift}	4.5×10^{-6}	
coef. of drag upper rot.	k_{up}^{drag}	2.5×10^{-7}	
coef. of drag lower rot.	k_{lo}^{drag}	2.5×10^{-7}	
COM to lower rotor dist.	d	0.0605	m
rotor radius	rad	0.371	m

4. The following initial conditions and gains apply to Simulation 1, as described in the Simulation of the System section:

Name	Symbol	Value	Unit
position	x_0	$\begin{bmatrix} -1.5 & 2 & 1 \end{bmatrix}^T$	m
rotation matrix	R_0	$\begin{bmatrix} .0736 & 0 & 0 \\ 0 & .097355 & 0 \\ 0 & 0 & .0732 \end{bmatrix}$	
rectilinear velocity	$\dot{x_0}$	$\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$	m/s
angular velocity	Ω_0	$\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$	rad/s
position error gain	k_x	4.5	
velocity error gain	k_v	5.0	

Given these initial conditions, the vehicle was asked to arrive at a state specified by the values in the table below. These values were held constant throughout the course of the simulation.

Name	Symbol	Value	Unit
position	x_d	$\begin{bmatrix}0 & 0 & 4\end{bmatrix}^T$	m
velocity	$\dot{x_d}$	$\begin{bmatrix}0 & 0 & 0\end{bmatrix}^T$	m/s
acceleration	$\ddot{x_d}$	$\begin{bmatrix}0 & 0 & 0\end{bmatrix}^T$	$\frac{m}{s^2}$

5. The following initial conditions and gains apply to Simulation 2, as described in the Simulation of the System section:

Name	Symbol	Value	Unit
position	x_0	$\begin{bmatrix}-1.5 & 2 & 1\end{bmatrix}^T$	m
rotation matrix	R_0	$\begin{bmatrix}cos(\pi/12) & 0 & sin(\pi/12) \\ 0 & 1 & 0 \\ sin(\pi/12) & 0 & cos(\pi/12)\end{bmatrix}$	
rectilinear velocity	$\dot{x_0}$	$\begin{bmatrix}0 & 0 & 0\end{bmatrix}^T$	m/s
angular velocity	Ω_0	$\begin{bmatrix}0 & 0 & 0\end{bmatrix}^T$	rad/s
position error gain	k_x	4.5	
velocity error gain	k_v	5.0	

For Simulation 2, the desired state values at each time step were constantly changing. For a full list of these values, please refer to the matrix `traj_1.mat` in the folder `Sim_011316` in the bitbucket repository listed below. This repository contains all MATLAB simulations run for the purposes of this report:
https://bitbucket.org/castacks/coax_uav_honors_research

References

[1] C. Bermes, "Design and dynamic modeling of autonomous coaxial micro helicopters," ETH (2010). [Online].

[2] C. Bermes, S. Bouabdallah, D. Schafroth, R. Siegwart, "Design of the autonomous micro helicopter muFly", *Mechatronics* 21.5 (2011) 765-775. [Online].

[3] T. Lee, M. Leok, and N. McClamroch, "Control of complex manuevers for a quadrotor UAV using geometric methods on SE(3)," arXiv. [Online].

[4] D. Schafroth, C. Bermes, S. Bouabdallah, R. Siegwart, "Modeling, system identification and robust control of a coaxial micro helicopter" *Control Engineering Practice* 18.7 (2010) 70-711. [Online].

[5] S. Bouabdallah, R. Siegwart, "Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor", *2005 IEEE Int. Conf. on Robotics and Automation* 2247 - 2252. [Online].

[6] P. Fankhauser, S. Bouabdallah, and R. Siegwart, "Modeling and decoupling control of the coax micro helicopter", *2011 IEEE/RSJ Int. Conf. Intell. Robot. Syst* 2223-2228. [Online].

[7] G. Limnaios, N. Tsourveloudis, "Fuzzy Logic Controller for a Mini Coaxial Indoor Helicopter" *Journal of Intelligent Robotic Systems*. 65.1-4 (2012): 187-201. [Online].

[8] F. Wang, T. Wang, B. Chen, T. Lee, "An Indoor Unmanned Coaxial Rotorcraft System with Vision Positioning" *2010 8th IEEE Int. Conf. on Control and Automation* 291-296. [Online].

[9] W. Wang, K. Nonami, Y. Ohira: "Model Reference Sliding Mode Control of Small Helicopter X.R.B based on Vision" *International Journal of Advanced Robotic Systems*. 5.3 (2008). [Online].

[10] L. Zhao and V. R. Murthy. "Optimal Controller for an Autonomous Helicopter in Hovering and Forward Flight", *2008 47th AIAA Guidance, Navigation and Control Conf. and Exhibit* [Online].

[11] H. Wang, D. Wang, X. Niu, and H. Duan, "Modeling and Hover Control of a Novel Unmanned Coaxial Rotor/Ducted-Fan Helicopter", *2007 IEEE International Conference on Automation and Logistics* 768–1773. [Online].

[12] F. Bohorquez, D. Pines "Hover performance and Swashplate design of a coaxial rotary wing micro air vehicle", *2004 American helicopter society 60th annual* [Online].