TiltCopter

Filip Bystricky, Yuu Jinnai, Freddie Rice

May 14, 2018

1 Abstract

Today, quadcopters are versatile and have a variety of useful applications in many fields, but they are limited. There exist alternate designs, but these either add too little value or are overly complicated. We propose a relatively simple modification which allows the quadcopter to tilt two of its diagonally opposed propellers. This modification adds a servo motor to tilt these propellers, as well as a fifth propeller to help counteract the torques that this introduces. Whereas normal quadcopters must tilt their entire frames to fly anywhere and can only hover in one orientation, our design will be able to fly sideways without tilting its frame, and hover in place while rotating its frame. This should enhance its ability to explore environments and make it better at many of its existing use cases.

2 Introduction

Quadcopters are used for all kinds of tasks, including photography, fun, SLAM, aerial mapping, crop inspection, and delivery, to name a few. Most if not all of these require a camera, and often some additional equipment to stabilize this camera. For example, someone using a drone for aerial photography probably wants the camera to remain steady as the drone flies around. All similar use cases would benefit from a drone which can keep its camera stable by design.

Most quadcopters today use a traditional 4/6/8 non-tilting rotor design. While these simple designs can achieve remarkably athletic feats, they have significant limitations. It is difficult for them to flip upside down, get near walls, maneuver in tight spaces, and impossible to float in any position other than right-side up (or perhaps upside down). Furthermore, all movements tilt the whole frame, including any cameras and sensors. Many consumer quadcopter designs are forced to make up for this by putting cameras/sensors on stabilized and/or tilting platforms.

We propose to solve this problem by making two propellers tiltable, so that the quadcopter can move around by tilting only those two propellers and not its entire frame. Specifically, our design has an axle running between two diagonally opposed propellers, which is attached to a servo motor that can rotate the axle

180^o. The two propellers are attached to the axle rather than to the base, so when the servo motor rotates the axle, the propellers tilt along the diagonal axis that runs between them. When this happens, these two propellers exert some thrust downwards, and some sideways, so while the frame stays level the drone will fly sideways. Alternatively, the quadcopter can tilt its frame while simultaneously tilting these two propellers, in order to float in place.

We show the use of our model by simulation and actual experiments. Currently, we have built and assembled a full prototype of our new design, and are working to get it flying like a normal quadcopter, without tilting its propellers. We will complete at least two more prototypes, with a few minor improvements based on what we have learned so far. We have also implemented a simulation of our quadcopter's dynamics model.

3 Related Work

There exist racing quadcopters which are similar to our idea. However, these tilt all four rotors in the same direction, in order to increase forward speed and keep the camera level, and don't have significantly better agility or maneuverability than normal quadcopters.

[3] proposed a quadcopter model which tilts four of the propellers independently. They showed in a simulation that their model (1) gains full controllability over the quadcopter pose, and (2) use the actuation redundancy (of order 2) to optimize energy efficiency. [4] also proposed a model with four tilting propellers. They showed that they can move the quadcopter while fixing the direction of the camera to the ground. [2] showed that the quadcopter can hover while tilting two of the propellers on an axis with a same angle.

It has been shown that with four additional control parameters, we are able to control the quadcopter, more efficiently and with more maneuverability. However, such a design is much more mechanically complex than that of a normal quadcopter. We believe our idea can achieve significant improvements without sacrificing too much of the original design's simplicity.

In terms of SLAM, the state of the art algorithms assume a non-tilting quadcopter design, and as a result have to deal with significant changes in the camera orientation as the quadcopter moves around its environment. Thus there seems to be much room for improvement in the results.

4 Technical Approach

This project involves solving two distinct problems. We solve the mechanical problem of physically designing the drone, and the mathematical problem of controlling it.

4.1 Design

Our design gives the quadcopter the ability to tilt two of its diagonally opposed rotors. This enables the quadcopter to spin in place and move around without tilting its base, keeping its cameras and sensors facing in the same direction. It also makes it possible for it to hover with its base facing any direction. This means that it could have one set of sensors, and the ability to point those sensors in any direction.

When a propeller spins, it creates a torque on the quadcopter in the opposite direction. And when any spinning object is tilted, it exerts a torque on an axis which is rotated 90° from the axis of tilting. When the quadcopter tilts its tiltable propellers, both of these phenomena act to produce a roll torque on the quadcopter. To counteract this, we add a fifth propeller inside the quadcopter, facing so that it exerts a roll torque when it spins.

We used OpenJSCAD to design the parts we needed, as well as to design an arrangement of all the electronics. Figures 1-4 show this model and what it looks like when the propellers tilt. The green box is a raspberry pi, the blue one is a battery, the yellow one is a flight controller, and the black one is the servo motor. If you look closely you'll see the axle which runs horizontally between the left and right propellers.

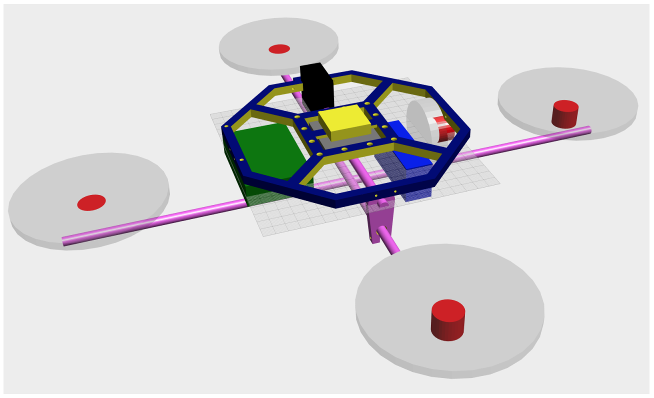


Figure 1: Bird's eye view of the model, in normal configuration (without tilted propellers)

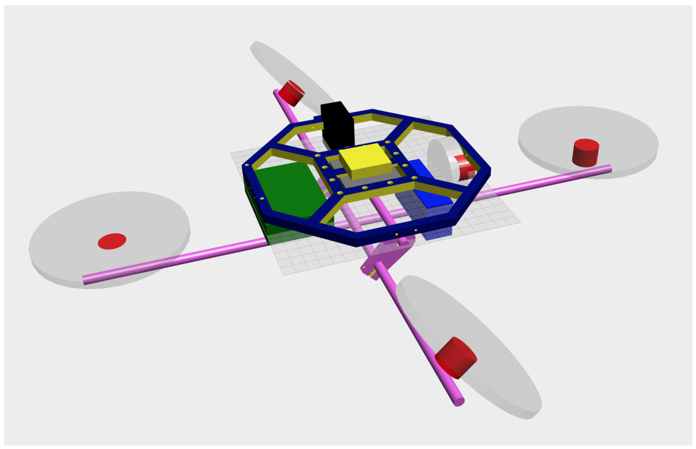


Figure 2: Bird's eye view of the model, with propellers tilted by 45^o

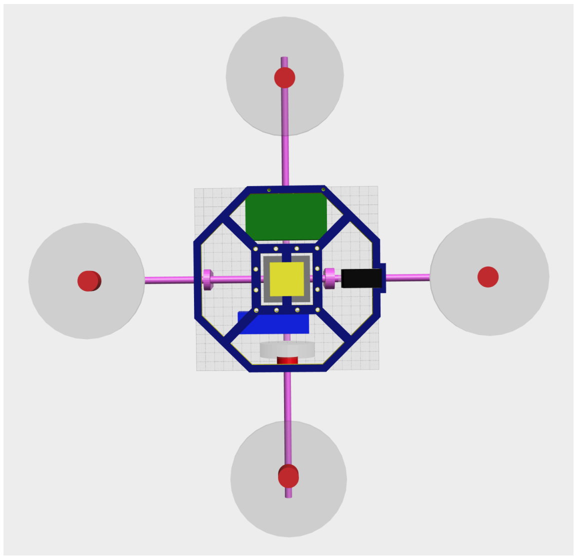


Figure 3: Bird's eye view of the model, in normal configuration (without tilted propellers)

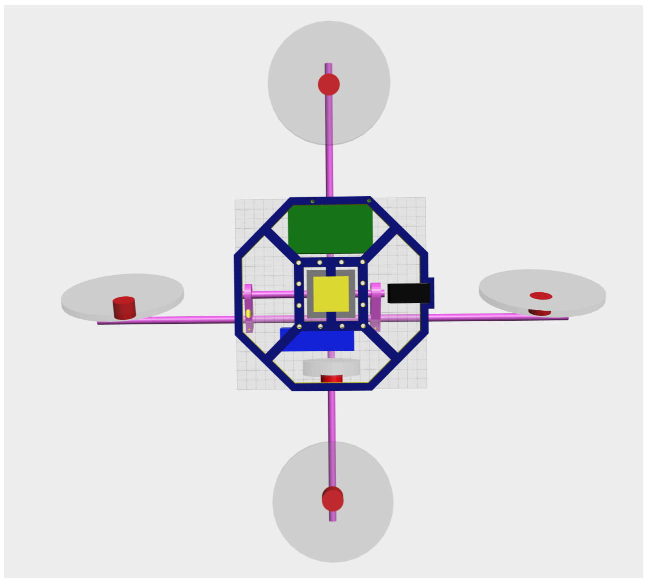


Figure 4: Bird's eye view of the model, with propellers tilted by 70°

4.2 Dynamics Model

Our simulation is based on the following model. First, we define a dynamics model for a standard quadcopter and our quadcopter. Then, we derive a control for hovering while the axis is tilted. Based on the hovering controller, we show a simple PD controller to follow a given trajectory for our model.

4.2.1 Dynamics of a Standard Quadcopter

Our model dymanics follows the assumptions on [2]. Figure 10 shows the body frame. The rod on the Y axis rotates along with propeller 1 and 3.

Let q be the linear and angular position of the quadcopter body in world axis. Euler angle transformations are defined by $\eta = [\phi\ \theta\ \psi]$: roll, pitch, yaw angle.

$$\xi = [x\ y\ z]^T \tag{1}$$
$$\eta = [\phi\ \theta\ \psi]^T \tag{2}$$
$$q = [\xi\ \eta]^T \tag{3}$$

Vertical forces along the X, Y, Z axes in the world frame are:

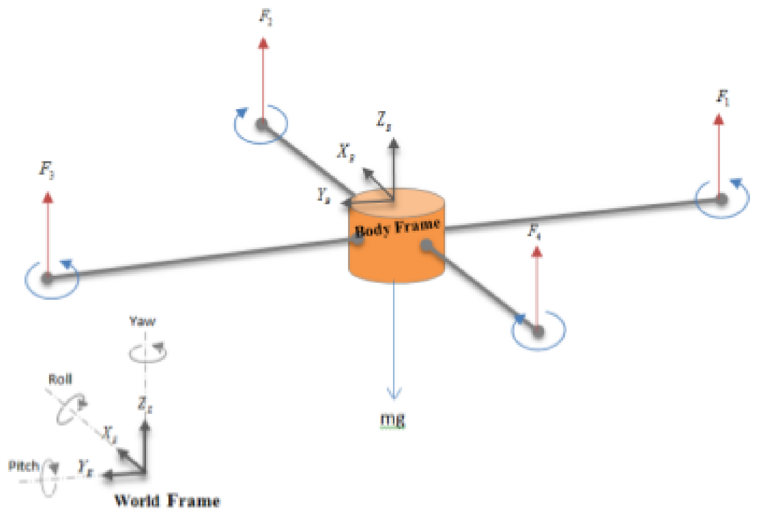


Figure 5: Body frame of the model. The rod on the Y axis tilts. Figure taken from [2]

$$m\ddot{x} = \sum F_i(s\phi s\psi + c\phi c\theta c\psi) - C_1\dot{x} \tag{4}$$
$$m\ddot{y} = \sum F_i(s\phi s\theta c\psi + c\phi s\psi) - C_2\dot{y} \tag{5}$$
$$m\ddot{z} = \sum F_i(c\theta c\psi) - mg - C_3\dot{z} \tag{6}$$
$$\tag{7}$$

where sx denotes $sinx$, cx denotes $cosx$, m is the total mass of quadcopter, g is the acceleration by the gravity. C_1, C_2, C_3 are drag coefficients.

F_i are forces produced by the rotors:

$$F_i = K_f\omega_i^2 \tag{8}$$

where ω_i is the angular velocity of i-th rotor, and K_f is a constant.

Angular accelerations of the quadroter are given by Euler equation:

$$I_x\ddot{\psi} = l(F_3 - F_1 - C_1'\dot{\psi}) \tag{9}$$
$$I_y\ddot{\theta} = l(F_4 - F_2 - C_2'\dot{\theta}) \tag{10}$$
$$I_y\ddot{\phi} = M_1 - M_2 + M_3 - M_4 - C_3'\dot{\phi} \tag{11}$$
$$\tag{12}$$

where l is a length of the arm, and C_1', C_2', C_3' are rotational drag coefficients. M_i are rotational drag produced by angular velocity of rotors:

$$M_i = K_m \omega_i^2 \tag{13}$$

where, K_m is a constant. Thus

$$M_i = kF_i \tag{14}$$

where $k = \frac{K_m}{K_f}$.

4.2.2 Dynamics of a Tilting Axis Quadcopter

For a tilting axis quadcopter we have two additional parameter μ and M_5. μ is the tilt angle of the axis with propeller 1 and 3 on. We assume that the propellers are right on top of the axis so that the forces generated by the propellers are perpendicular to these respective planes of rotations. M_5 is a rotational torque produced by the fifth rotor to the roll. We assume that the fifth rotor does not produce a thrust.

$$\begin{aligned}
m\ddot{x} &= F_1 s\mu c\psi c\theta + F_3 s\mu c\psi c\theta && (15)\\
&+ F_1 c\mu c\psi s\theta c\phi + F_2 c\psi s\theta c\phi && (16)\\
&+ F_3 c\mu c\psi s\theta c\phi + F_4 c\psi s\theta c\phi && (17)\\
&+ F_1 c\mu s\psi s\phi + F_2 s\psi s\phi && (18)\\
&+ F_3 c\mu s\psi s\phi + F_3 s\psi s\phi - C_1\dot{x} && (19)\\
m\ddot{y} &= F_1 s\mu c\psi c\phi + F_3 s\mu s\psi c\theta && (20)\\
&+ F_1 c\mu s\psi s\theta c\phi + F_2 s\psi s\theta s\phi && (21)\\
&+ F_3 c\mu s\psi s\theta c\phi + F_4 s\psi s\theta s\phi && (22)\\
&+ F_1 c\mu c\psi s\phi - F_2 c\psi s\phi && (23)\\
&- F_3 c\mu c\psi \phi - F_4 c\psi s\phi - C_2\dot{y} && (24)\\
m\ddot{z} &= -F_1 s\mu s\theta - F_3 s\mu s\theta && (25)\\
&+ F_1 c\mu c\theta c\phi + F_2 c\theta c\phi && (26)\\
&+ F_3 c\mu c\theta c\phi + F_4 c\theta c\phi && (27)\\
&- mg - C_3\dot{z} && (28)
\end{aligned}$$

The angular accelerations are determined by Euler equation:

$$\begin{aligned}
I_x\ddot{\phi} &= l(F_3 c\mu - F_1 c\mu - C_1'\dot{\phi}) && (29)\\
&+ (M_1 s\mu + M_3 s\mu) && (30)\\
I_y\ddot{\theta} &= l(F_4 - F_2 - C_2'\dot{\theta}) && (31)\\
&+ M' + M_5 && (32)\\
I_z\ddot{\psi} &= l(F_1 s\mu - F_3 s\mu - C_3'\dot{\psi}) && (33)\\
&+ (M_1 c\mu - M_2 + M_3 c\mu - M_4) && (34)
\end{aligned}$$

where M' is a moment produced by the servo motor, and M_5 is a rotational drag produced by the fifth rotor.

4.2.3 Rotor Model

Our rotors are driven by DC-motors which follows the model by [1].

$$L\frac{di}{dt} = u - Ri - k_e\omega \tag{35}$$

$$J\frac{d\omega}{dt} = \tau_m - \tau_d \tag{36}$$

where u is a motor input, k_e is a back EMF constant, τ_m is a motor torque, τ_d is a motor load, R is a motor internal resistance. We assume that an inductance of the motor is negligible, thus:

$$J\frac{d\omega}{dt} = -\frac{k_m^2}{R}\omega - \tau_d + \frac{k_m}{R}u \tag{37}$$

where I is an input current, I_0 is the current with no load on the motor, and K_t is a constant.

4.2.4 Hovering for a Tilting Model

First we consider the control for hovering. Our model can hover at a place while tilting the axis. Assume that all four rotors having equal rotational speed (thus $F_i = F$). Let μ be the fixed tilting axis. Then the vertical forces are as follows:

$$m\ddot{x} = F(s(\mu - \phi) + s(\mu - \phi) - s\phi - s\phi) \tag{38}$$

$$m\ddot{y} = 0 \tag{39}$$

$$m\ddot{z} = F(c(\mu - \phi) + c\phi - c(\mu + \phi) - c\phi) - mg \tag{40}$$

To hover we need all $\ddot{x}, \ddot{y}, \ddot{z}$ to be zero. Thus,

$$\phi = \frac{\mu}{2} \tag{41}$$

The motor speed is given by:

$$\omega_i = \omega_h = \sqrt{\frac{mg}{4k_f c\frac{\mu}{2}}} \tag{42}$$

4.2.5 Control Model

For a simulation we consider a simple PD controller based on the hovering attitude following the model by [2]. Given the derivations of the angle velocities of the body, the desired angular velocities of the rotors ω_i^{ref} are given as follows:

$$\begin{bmatrix} \omega_1^{ref} \\ \omega_2^{ref} \\ \omega_3^{ref} \\ \omega_4^{ref} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 & 1 \\ 1 & 1 & 0 & -1 \\ 1 & 0 & 1 & 1 \\ 1 & -1 & 0 & -1 \end{bmatrix} \begin{bmatrix} \omega_h + \Delta\omega_f \\ \Delta\omega_\phi \\ \Delta\omega_\theta \\ \Delta\omega_\psi \end{bmatrix} \tag{43}$$

We use the PID controller defined as follows to calculate the desired derivations of the angle velocities of the body:

$$\Delta\omega_\phi = k_{p,\phi}(\phi^{ref} - \phi) + k_{d,\phi}(p^{ref} - p) \tag{44}$$

$$\Delta\omega_\theta = k_{p,\theta}(\theta^{ref} - \theta) + k_{d,\theta}(q^{ref} - q) \tag{45}$$

$$\Delta\omega_\psi = k_{p,\psi}(\psi^{ref} - \psi) + k_{d,\psi}(r^{ref} - r) \tag{46}$$

$$\tag{47}$$

The relationship between the tilt angle μ^{ref} and the reference roll angle ϕ^{ref} is given as:

$$\mu = 2\phi_h^{ref} + \Delta\phi_h \tag{48}$$

where ϕ_h is reference roll angle and $\Delta\phi_h$ is a deviation.

$$\Delta\phi_h = k_{p,\phi_h}(\phi_h^{ref} - \phi) - k_{d,\phi_h}(p) \tag{49}$$

Note that the controller does not make use of the tilt axis for efficiency: for an efficient flight we should tilt the axis toward the desired direction.

5 Evaluation

5.1 Design

Design-wise, our goal was to design and build a prototype that can fly, tilt two of its propellers, and has a fifth “roll” propeller. In this, we were successful. We managed to design several prototypes, all of which were able to fly, tilt two of their propellers, and had a fifth “roll” propeller. However, we ran into a number of issues with 3D printed parts breaking, and ended up re-designing parts of our prototype each time.

Our first prototype, pictured in figure 6, had parts which were too rigid and did not absorb the shock of crashing well at all. As a result, multiple parts of it, including the frame, shattered on the first minor crash.

In building this prototype, we also realized that a number of its parts were unnecessarily heavy, so we bought smaller axles and re-designed the frame and other 3D printed parts to be thinner and lighter. We also made our first attempt at 3D printing some shock-absorbing feet to help protect the drone from crashes. This resulted in our second prototype, pictured in figure 7. Unfortunately, we

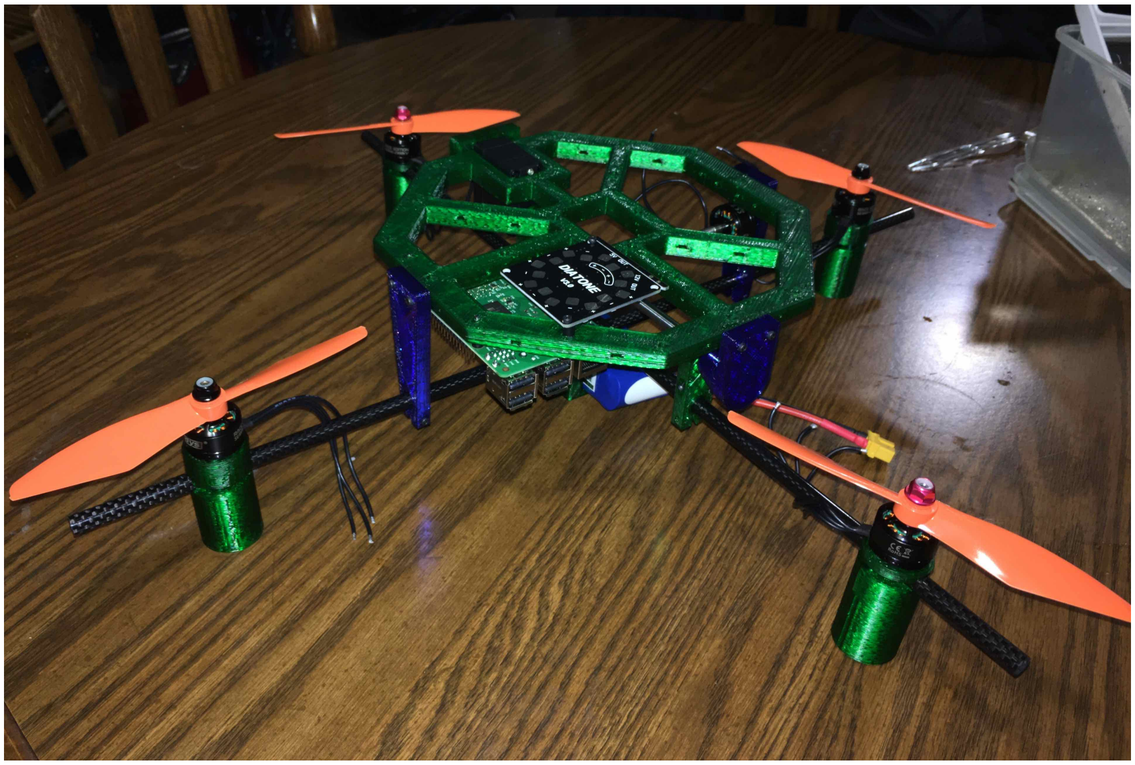


Figure 6: First prototype (missing some electronics). The far left and near right propellers tilt along the axis that connects them.

then spent a long while solving software bugs, and when we finally got it to fly it crashed and some 3D printed parts broke right away.

In the meantime, we managed to get three copies of our frame laser-cut out of acrylic at the BDW, and assembled our third prototype (pictured in figure 8 with two of its propellers tilted). For this prototype, we redesigned all of the small 3D-printed parts which broke in previous prototypes to be circular in an effort to make them capable of absorbing larger shocks. Unfortunately, all of these parts kept breaking when the drone fell during testing.

Finally, we abandoned hope that 3D printed parts could survive the impact of the drone crashing, and made our fourth and fifth prototypes (pictured in figures 9 and 10). These are identical, except that the one with orange propellers uses packing foam to absorb shock, while the other one uses bundles of zip-ties to do the same. These ended up working really well, and both prototypes have survived numerous crashes.

Overall, we completely achieved our hardware design goals with our fourth and fifth prototypes. Both can fly and tilt two propellers, and both have a fifth propeller which help them influence their roll.

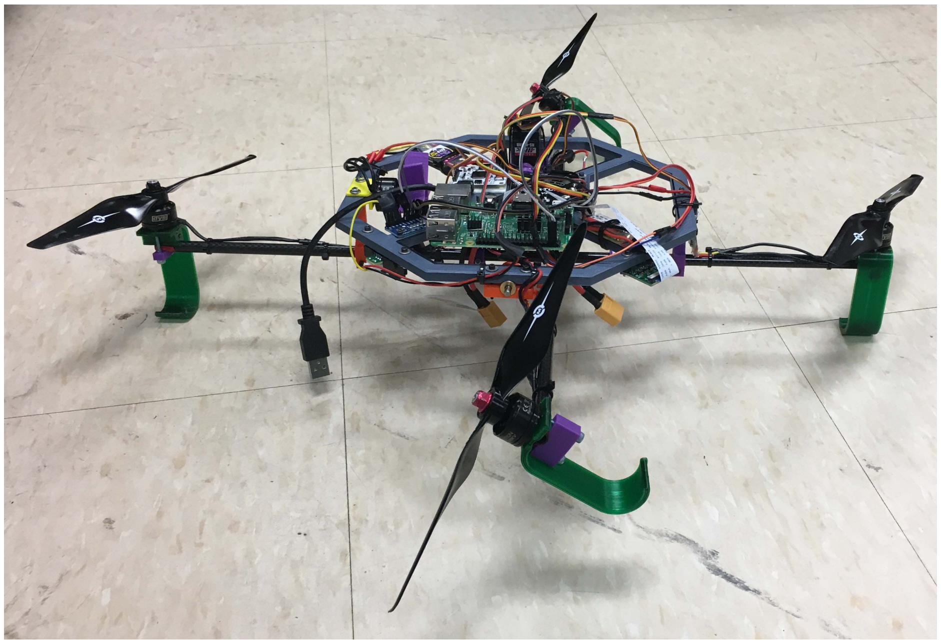


Figure 7: Second prototype

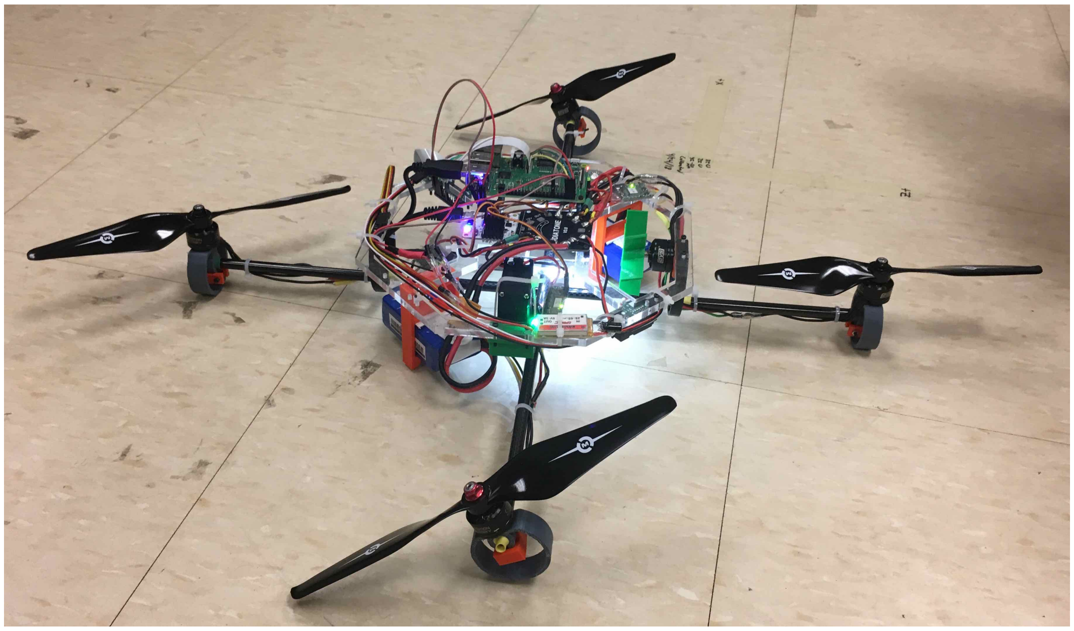


Figure 8: Third prototype

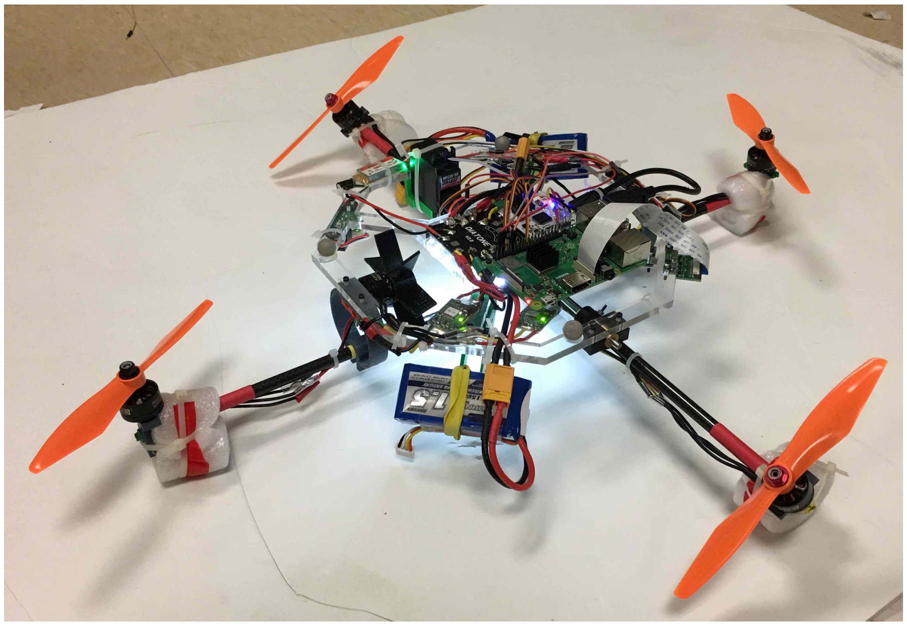


Figure 9: Fourth prototype

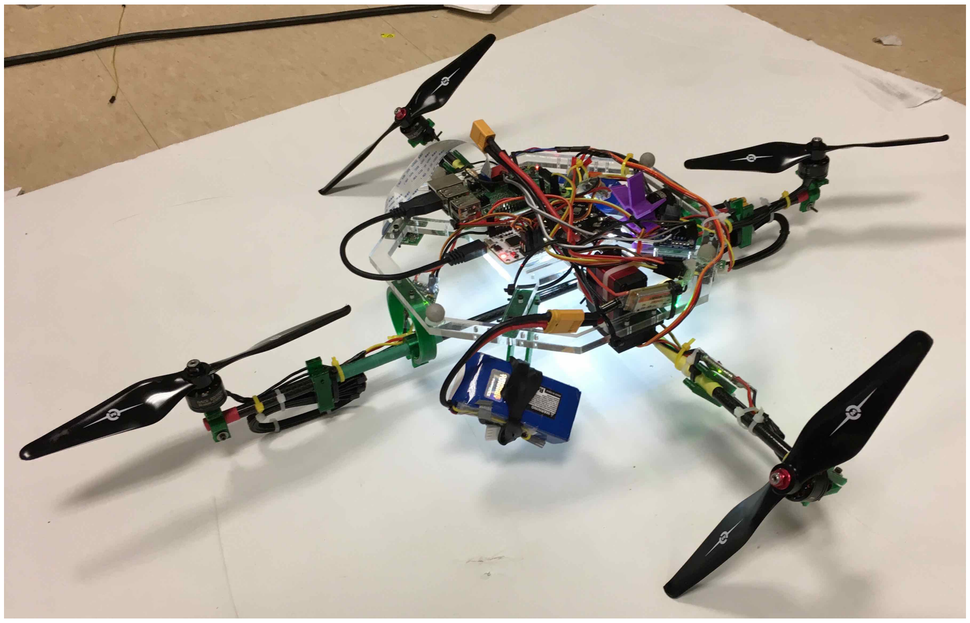


Figure 10: Fifth prototype

5.2 Simulation

5.2.1 Setup

We evaluated our model in a simulation for three tasks. First, we tested if we can take-off while the rod is tilted. Second, we evaluated our model on following a given trajectory, comparing against a model without tilting. Third, we ran a simulation to tilt the rod while hovering at a place.

5.2.2 Taking-off

The first task we consider is to take-off and keep the altitude to a fixed position. Note that we only keep the altitude but do not fix X and Y positions. For this task we use a same controller for models.

The videos are available at the following link.

1. a standard model `https://youtu.be/vz8QKnq6cmI`
2. a model with tilt angle fixed to 15 with a controller for a standard model `https://youtu.be/wdr1r41h1lE`
3. a servo model with a controller for a model `https://youtu.be/UjDNYp-GJJw`

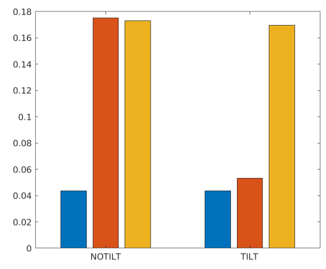
5.2.3 Following a Trajectory

We compare the performance of our model to follow a trajectory. The video comparing the trajectories of the two models is available at `https://youtu.be/kUVdCoH2x_w`. Figure 11 shows the comparison of the RMS errors with and without tilting. Note that the RMS error for X axis is smaller for the tilt model than the non-tilt model. This is because the initial position of the quadcopter is set in a way that it can use tilting to control X position, reducing the errors on X axis. Figure 12 and 13 show the trajectories and angles of the two models. The figures suggest that our model is faster to reach the desired position while it is as stable as the base model. Figure 14 is a simulated camera angle of the two models. The camera angle is defined as an angle of the Z axis of the body frame and the Z axis of the world frame. Our hypothesis is that our model can keep the camera angle more stable while moving. So far we did not see an evidence to support the claim: the camera angles of the two models are roughly the same stability.

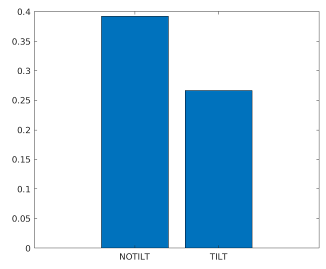
To understand the strength of the tilting, we additionally implemented and evaluated a model which can tilt both of the rods. The video is available at `https://youtu.be/VQUDTOCBM_0`.

5.2.4 Tilt and Hover

As explained in Section 4.2.5 our tilt model enables to tilt the body angle while stabilizing at a point. We ran a simulation to tilt the rod angle to 10 degree while hovering at a point. The video is available at `https://youtu.be/-aFCNNM8tF0`.

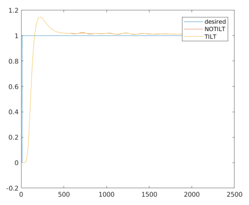


(a) X, Y, Z

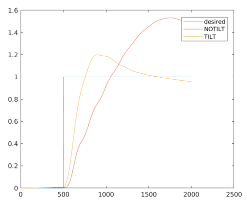


(b) Total

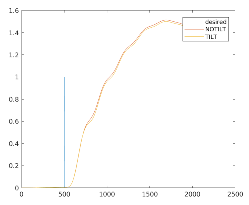
Figure 11: RMS errors with and without tilting. Figure on the left shows the errors for altitude, x, and y axis. Figure on the right shows the sum over three axis. A model with a tilt axis is achieving lower errors. The tilt model has lower error on X axis.



(a) Altitude

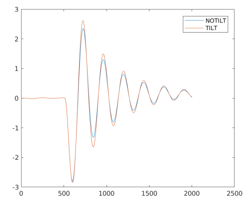


(b) X

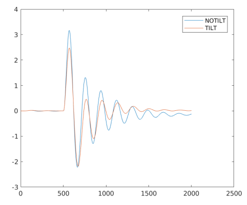


(c) Y

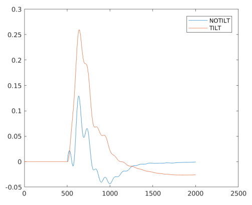
Figure 12: Comparison of trajectories by the model with and without tilting. The tilt model converges to the desired X position faster than the non-tilting model.



(a) Roll



(b) Pitch



(c) Yaw

Figure 13: Comparison of angles by the model with and without tilting. We observed that our model is as stable as the standard model without tilting.

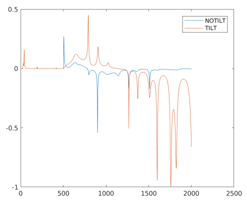


Figure 14: Comparison of camera angle by the model with and without tilting. We did not observe a significant difference in camera stability between the two models.

5.3 Flight

We have managed to get our prototype off the ground, but unfortunately broke a couple of propellers and 3D printed parts during PID tuning. We made some design improvements to prevent further breaking, such as foam feet instead of plastic feet to prevent the drone components from breaking. We were able to get our prototype to fly like a normal quadcopter, then we were able to get it to fly for around 20 seconds with a 10 degree tilt. We started to face issues with the camera not understanding velocity at higher tilt angles, so we put it into the motion capture system. Unfortunately, we failed to find the right change of basis between the coordinate system of the drone and the coordinate system of the motion capture system due to our limited understanding of quaternions, Eigen and VRPN. We did have one configuration that almost worked, but with a flipped-z axis, which meant that the drone could not determine its height. That being said, from what we could infer from our test flights, the standard PID control works well enough for the tilt drone at low angles. It was, however, important to only tilt the drone's axis once the drone was in the air, otherwise the drone takes off at high speed toward one direction. This effect is most notable at higher angles, such at 20 and above. We also encountered a large number of issues with the ADC chips. We had to replace 3 of them throughout the course of our design.

https://youtu.be/MLazhQg-RlY

6 Conclusion

Our final tilt drone design is a great platform for future research in the area. In the end, the code had too many bugs to give us everything we wanted, but the project as a whole gave some insights into new directions for quadcopter flight. Even though we missed some of our ambitious goals, this is not a lost project. We are convinced that the platform can be tuned to provide as dynamic flight as that of the simulation.

6.1 Future Work

Future work should start with finishing our original goals. It could use our project as a starting point to develop the code and perfect the design for a drone which can controllably fly around without tilting its base, or hover in place while tilting its base. We have shown that these work in simulation, but there is still work left to get them to work on a real prototype. Additionally, with some effort the design could be made much smaller and lighter.

In the slightly longer term, future work should develop a more standard platform for developing highly configurable drones. It should follow the SpaceX model of developing a prototype that can be reused after multiple failed experiments. We found that our first drone could only sustain one flight. Around 15-20 man hours of work had led to a broken bunch of wires and plastic. We had learned our lesson and decided to throw components at the ground to make sure that they wouldn't break during flight. This saved somewhere in the range of 2 drones worth of time. This resulted in our final drones being much more tolerant to large falls and led to a large number of iterations per drone. However, we think that even more time could be saved if an engineer could build a dynamic drone with multiple tilting axis that was tolerant to crashes at relatively high speeds. From there, future drones could

Even longer term work could attempt to make further design modifications, perhaps adding even more degrees of freedom and exploring the maneuverability benefits thereof. It could explore which modified designs are possible to build, program, fly, and which designs are better at which applications.

References

[1] Design and control of an indoor micro quadrotor. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 5, pages 4393–4398. IEEE, 2004.

[2] A. Nemati and M. Kumar. Modeling and Control of a Single Axis Tilting Quadcopter. pages 3077–3082, 2014.

[3] M. Ryll, H. H. Bulthoff, and P. R. Giordano. Modeling and control of a quadrotor UAV with tilting propellers. *2012 IEEE International Conference on Robotics and Automation*, pages 4606–4613, 2012.

[4] F. Senkul and E. Altug. Modeling and control of a novel tilt - Roll rotor quadrotor UAV. *2013 International Conference on Unmanned Aircraft Systems, ICUAS 2013 - Conference Proceedings*, pages 1071–1076, 2013.