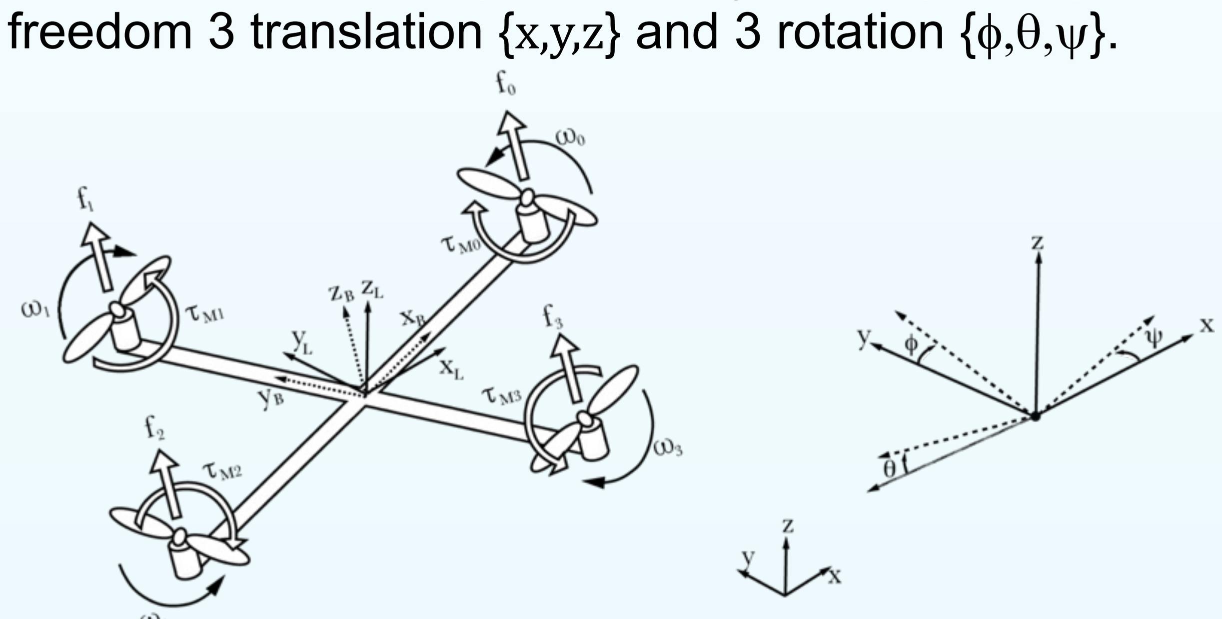
A NOVEL FAULT TOLERANT DUAL-LOOP CONTROL SYSTEM FOR AUTONOMOUS NAVIGATION

Statement of Problem

Autonomous unmanned vehicles (UAVs) are growing in usage for both commercial and military applications. Quadcopters have gained interest because of their high versatility and maneuverability, as well as their structural simplicity. These attributes lead to multiple applications in delivery, aerial surveillance, search and rescue, as well as communication base stations. However, the complexity of quadcopter dynamics makes navigation and stability challenging. Further, real world conditions like wind or humidity can cause fault in the actuators or sensors, creating a high potential for the quadcopter to crash. This project researches a UAV fault-tolerant control system (FTC), which would allow for greater stability, thereby increasing quadcopter adoption.

Introduction

Quadcopter dynamics require understanding how the angular velocity of the motors create thrust and torque resulting in the quadcopter having 6 degrees of freedom 3 translation $\{x,y,z\}$ and 3 rotation $\{\phi,\theta,\psi\}$.



The thrust, T_B, is the force to keep the quadcopter in the air against gravity. It is a function of motor angular velocity, ω_i, and observed lift constant, k. The torques, τ_B, rotate the quadcopter so the quadcopter can move in the correct direction. In each angle $\{\phi,\theta,\psi\}$, a torque is created as a function of motor angular velocity, ω_i, the quadcopter length, l, empirical drag, b, and lift constants k.

$$T_B = \begin{bmatrix} 0 \\ 0 \\ T = \sum_{i=1}^{4} f_i = k\sum_{i=1}^{4} \omega_i^2 \end{bmatrix} \qquad \tau_B = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} lk(-\omega_2^2 + \omega_4^2) \\ lk(-\omega_1^2 + \omega_3^2) \\ \sum_{i=1}^{4} \tau_{Mi} = -b\sum_{i=1}^{4}(-1)^i \omega_i^2 \end{bmatrix}$$

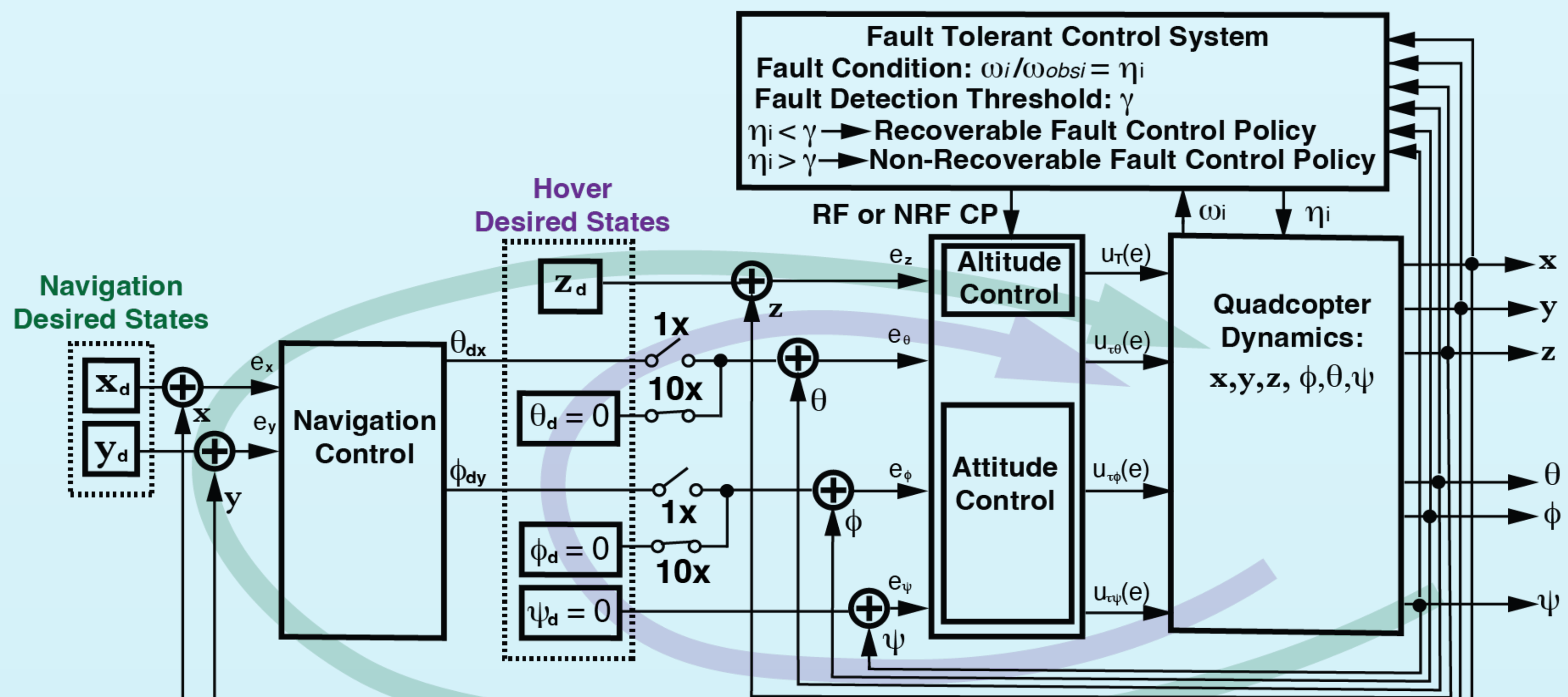
Using Newtonian translational and rotational physics with the mass, m, and Inertia, I, the thrust and torques can provide the x, y, z acceleration and ϕ, θ, ψ angular acceleration of the quadcopter.

$$\begin{bmatrix} \alpha_{B\phi} \\ \alpha_{B\theta} \\ \alpha_{B\psi} \end{bmatrix} = \begin{bmatrix} \tau_\phi/I_{xx} \\ \tau_\theta/I_{yy} \\ \tau_\psi/I_{zz} \end{bmatrix} + \begin{bmatrix} (I_{yy}-I_{zz})\omega_{B\theta}\omega_{B\psi}/I_{xx} \\ (I_{zz}-I_{xx})\omega_{B\phi}\omega_{B\psi}/I_{yy} \\ (I_{xx}-I_{yy})\omega_{B\theta}\omega_{B\phi}/I_{zz} \end{bmatrix} - I_r \begin{bmatrix} \omega_{B\theta}/I_{xx} \\ -\omega_{B\phi}/I_{xx} \\ 0 \end{bmatrix} \omega_\Gamma$$

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}$$

Methodology

A Novel Dual Loop Control Feedback System is developed for Quadcopter Stabilization and Navigation. The dual-loop control system utilizes the outer loop (green) to determine control inputs for navigation while the inner hover loop (purple) moves the quadcopter states closer to a hover state. Fault detection occurs during the hover loop. The error between the desired state and the measured output from the gyroscope and accelerometer sensors determines the control input (u_z, u_θ, u_ϕ, u_ψ) from the control algorithm.



The simulated control algorithms included the PID, Backstepping Method, and Sliding Method. Using the control inputs, the control units are then calculated via the equations listed below. The final motors speeds, as shown in the Simulation Section, are determined based on the control units and the control policy.

$$U_T = (g + u_z)\frac{m}{\cos\theta\cos\phi} \qquad U_{\tau\phi} = I_{xx}u_\phi$$

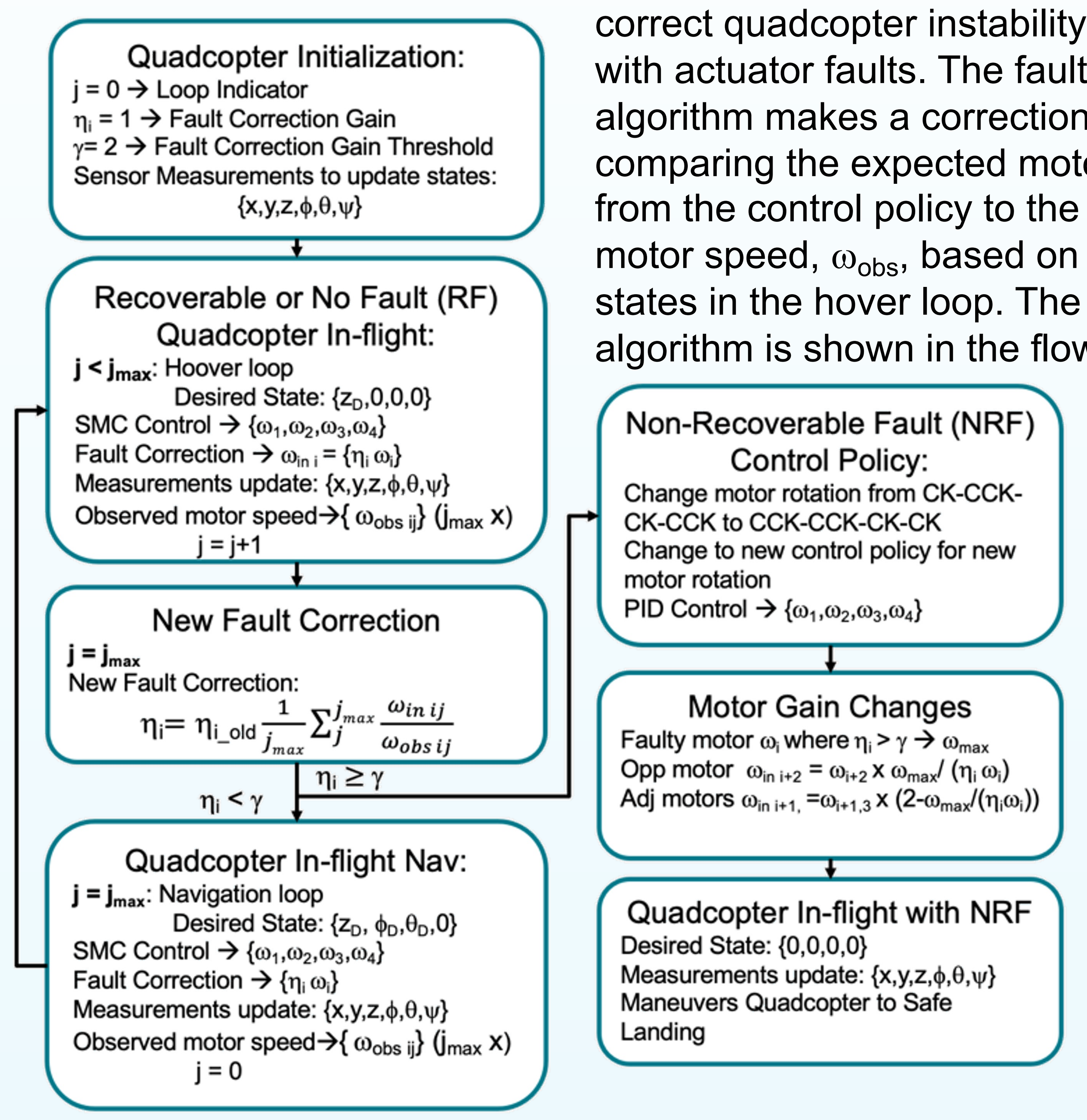
$$\theta_D = \frac{m}{U_T}u_x \qquad U_{\tau\theta} = I_{yy}u_\theta$$

$$\phi_D = \frac{m}{U_T}u_y \qquad U_{\tau\psi} = I_{zz}u_\psi$$

All diagrams/images were created by author

Simulations

The FTC system determines how to correct quadcopter instability when faced with actuator faults. The fault tolerant algorithm makes a correction based on comparing the expected motor speed, ω_i, from the control policy to the observed motor speed, ω_{obs}, based on the measured states in the hover loop. The complete algorithm is shown in the flow-chart.



In the event of a non-recoverable fault, a change in the motor rotations is further required to maintain stability. With the conventional control policy (on left), if the motor speeds in the pitch and roll directions are imbalanced, the yaw angle accelerates, quickly destabilizing the quadcopter and likely resulting in a crash, as seen in the MATLAB simulation below (left). To counteract this instability created when the four motor speeds are uneven, the FTC system creates a new hardware state for motor rotation, so there is no acceleration in the yaw angle. Further, a novel control system is utilized (right), so the quadcopter remains stable during extreme non-recoverable fault conditions (right).

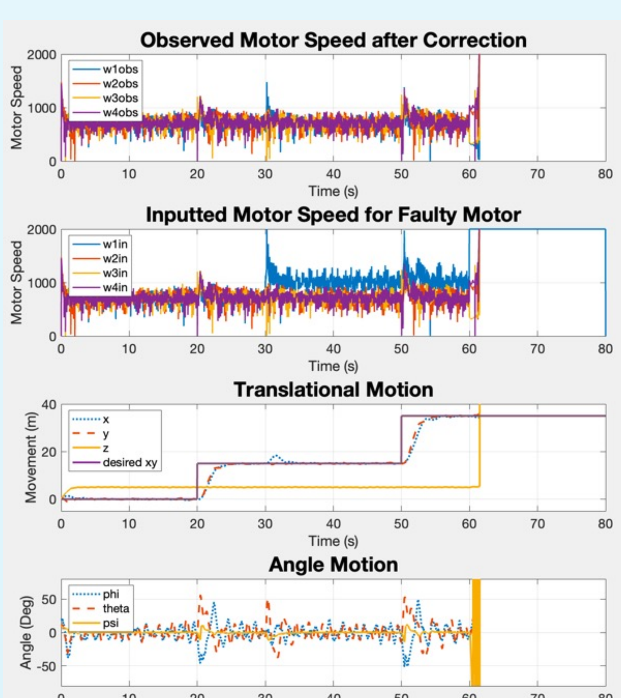
Classical Control Policy

$$\begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} = \begin{bmatrix} \frac{1}{4k} & 0 & -\frac{1}{2kl} & \frac{1}{4b} \\ \frac{1}{4k} & -\frac{1}{2kl} & 0 & -\frac{1}{4b} \\ \frac{1}{4k} & 0 & \frac{1}{2kl} & \frac{1}{4b} \\ \frac{1}{4k} & \frac{1}{2kl} & 0 & -\frac{1}{4b} \end{bmatrix} \begin{bmatrix} U_T \\ U_{\tau\phi} \\ U_{\tau\theta} \\ U_{\tau\psi} \end{bmatrix}$$

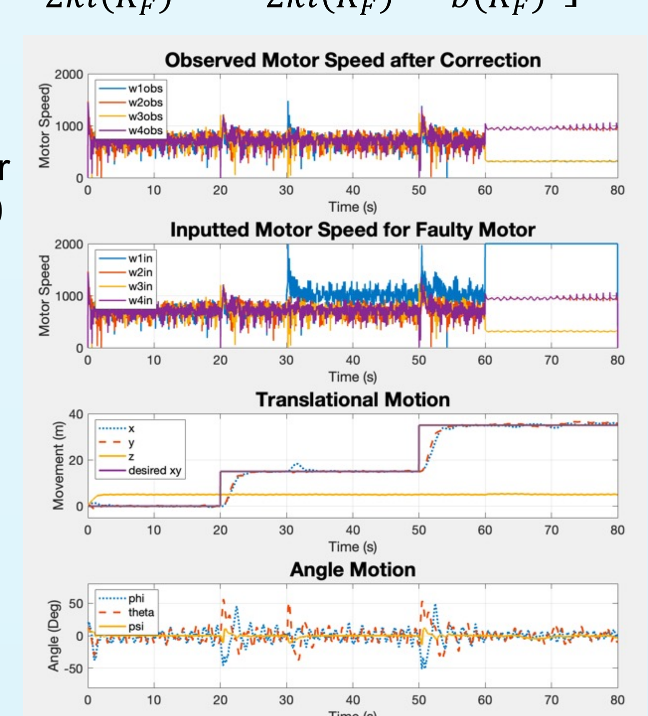
Fault Tolerant Control Policy

$$where\ K_F = 4b^2 + 2(kl)^2$$

$$\begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} = \begin{bmatrix} \frac{1}{4k} & \frac{b^2}{2kl(K_F)} & -\frac{K_F - b^2}{2kl(K_F)} & -\frac{1}{b(K_F)} \\ \frac{1}{4k} & -\frac{K_F - b^2}{2kl(K_F)} & \frac{b^2}{2kl(K_F)} & -\frac{1}{b(K_F)} \\ \frac{1}{4k} & -\frac{b^2}{2kl(K_F)} & \frac{K_F - b^2}{2kl(K_F)} & \frac{1}{b(K_F)} \\ \frac{1}{4k} & \frac{K_F - b^2}{2kl(K_F)} & -\frac{b^2}{2kl(K_F)} & \frac{1}{b(K_F)} \end{bmatrix} \begin{bmatrix} U_T \\ U_{\tau\phi} \\ U_{\tau\theta} \\ U_{\tau\psi} \end{bmatrix}$$

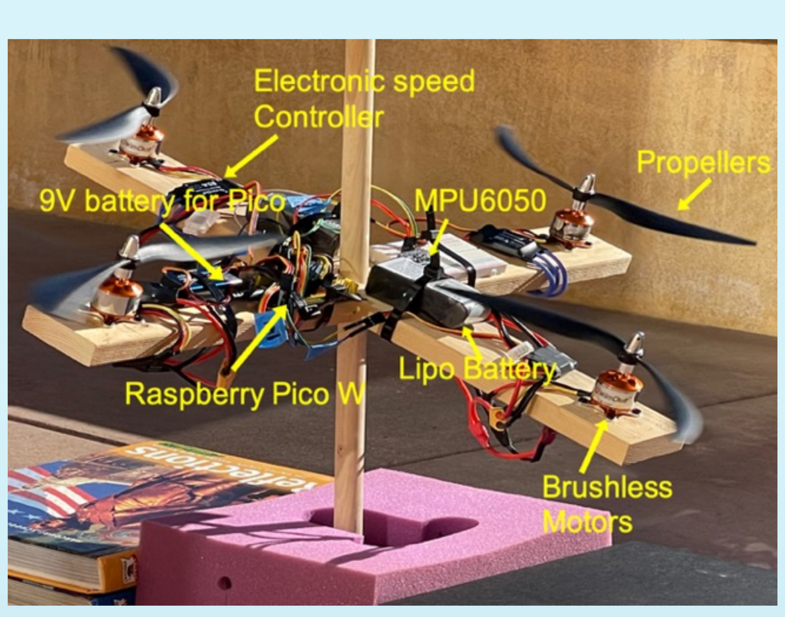


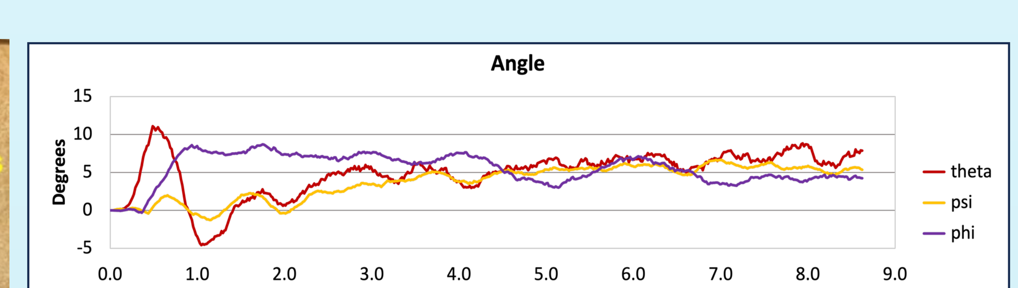
Simulations for Quadcopter with recoverable fault at 30 sec, Motor 1 speed is reduced to 70% at 30 sec, and non-recoverable fault at 60 sec, Motor 1 at max speed setting is effectively only 50% of required speed. Non-recoverable Fault using conventional control algorithm shows instability

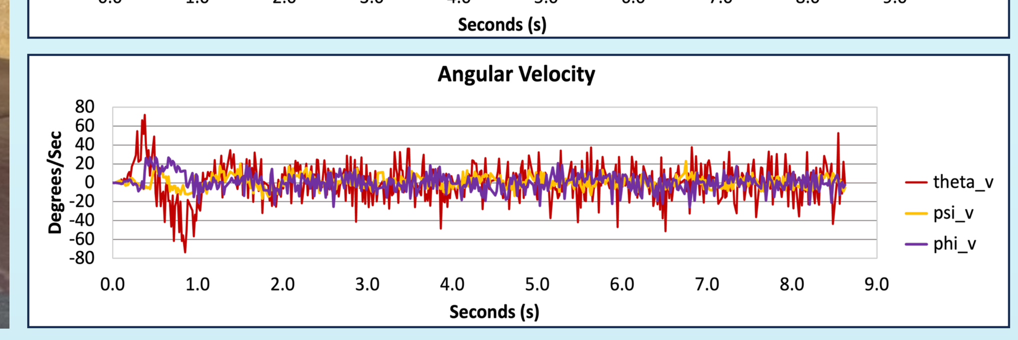


Experimental Data

A quadcopter prototype was designed to validate the new fault tolerant control policy. The dowel in the center allows the model to move freely in the ϕ, θ, ψ, and z direction, while restricting movement in the x and y direction. Flight data was collected using the novel fault tolerant control policy shows the quadcopter can be stable for a limited amount of time.







Conclusion

A Novel Fault Tolerant Dual Loop Control System is designed and validated with MATLAB simulations and prototype data. Current fault tolerant control algorithms do not minimize yaw acceleration and rate in many fault conditions, causing the quadcopter to become unstable and crash. This research showcases a control system that minimizes yaw in the event of motor actuator fault, resulting in stability so the quadcopter can make a safe landing.