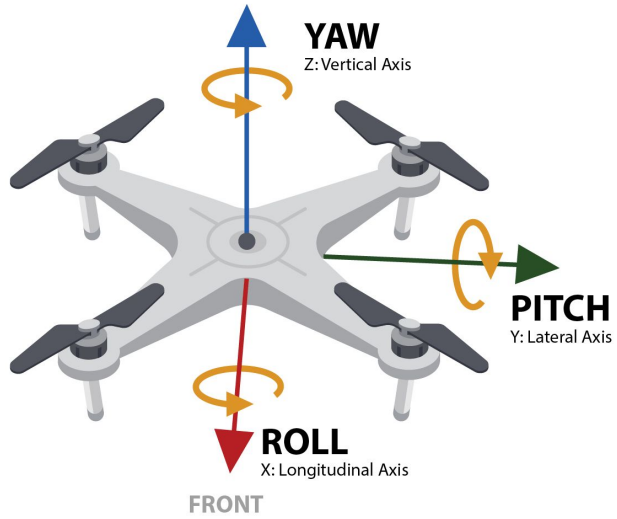
Bachelor Thesis Project - 2
Attitude Control of One DOF Drone

Kushal Agarwal
210100087

Guide: Prof. Vivek Sangwan
Department of Mechanical Engineering,
Indian Institute of Technology Bombay

Introduction and Objective

Attitude control refers to the control of the orientation and position of an object in 3D space



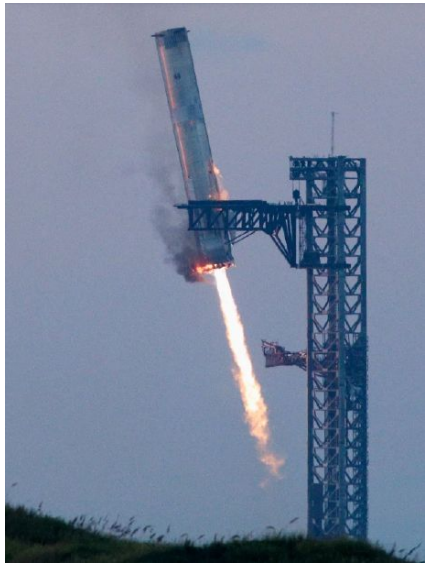
Attitude nomenclature

Applications:

- Coupled dynamics of aerial vehicles/submarines
- Balancing boards for rehabilitation
- Robots for maintaining orientation
- In defense for guided missiles

Learning Outcomes:

- Non-linear coupled dynamics control
- Nested loop controller
- Dynamics simulation



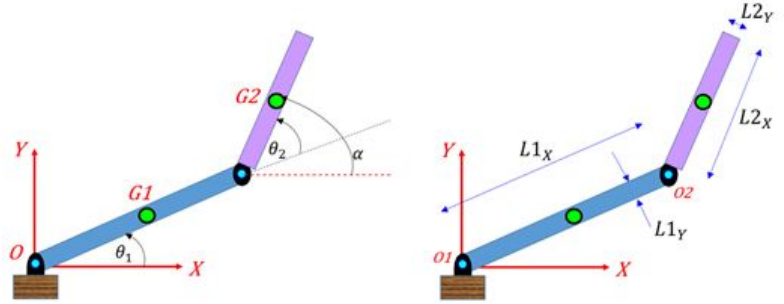
SpaceX rocket booster

MQ-9 Reaper of US Air Force

Understanding Simulation - 2R Robot Arm

Objective:

- Understanding dynamics derivation using Lagrange Equation
- Simulation of a real system using ode45
- Designing, implementing and tuning controllers for a given system

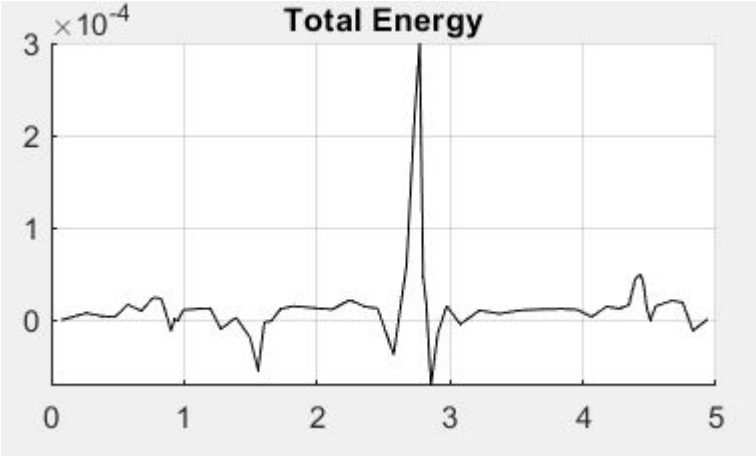


2R robot arm

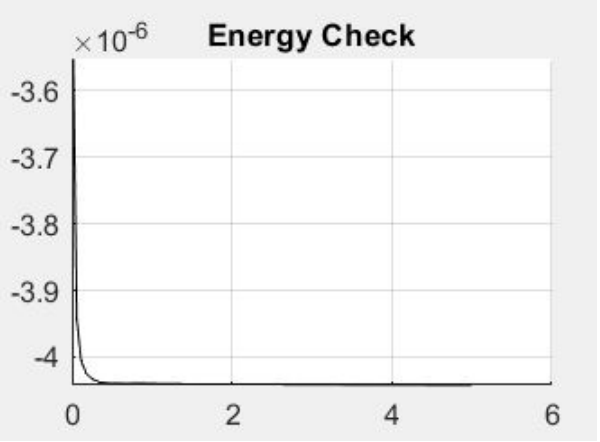
Kinematics	Energies	Dynamics
$V_{C_1} = \dot{\theta}_1 a_1(-\sin\theta_1 \mathbf{e_x} + \cos\theta_1 \mathbf{e_y})$ $V_{C_2} = (l_1/a_1)V_{C_1} + (\dot{\theta}_1 + \dot{\theta}_2)a_2(\cdot$ $-\sin(\theta_1+\theta_2)\mathbf{e_x} + \cos(\theta_1+\theta_2)\mathbf{e_y})$ $\omega_1 = \dot{\theta}_1 \qquad \omega_2 = \dot{\theta}_1 + \dot{\theta}_2$	$KE_1 = (1/2)(m_1 a_1^2 {\omega_1}^2 + I_1 {\omega_1}^2)$ $KE_2 = (1/2)(m_2(l_1^2\omega_1^2 + a_2^2\omega_2^2 + 2l_1a_2\omega_1\omega_2\cos\theta_2) + I_2\omega_2^2)$ $PE_1 = m_1 g a_1 \sin\theta_1$ $PE_2 = m_1 g(l_1\sin\theta_1 + a_2\sin(\theta_1+\theta_2))$	$\mathbf{M}(\theta_1,\theta_2)\begin{pmatrix}\ddot{\theta}_1\\ \ddot{\theta}_2\end{pmatrix} + \mathbf{C}(\theta_1,\theta_2,\dot{\theta}_1,\dot{\theta}_2)\begin{pmatrix}\dot{\theta}_1\\ \dot{\theta}_2\end{pmatrix} + \mathbf{G}(\theta_1,\theta_2) = \begin{pmatrix}\tau_1\\ \tau_2\end{pmatrix}$
State Space		Controller
$\begin{pmatrix}\ddot{\theta}_1\\ \ddot{\theta}_2\\ \dot{\theta}_1\\ \dot{\theta}_2\end{pmatrix} = \begin{pmatrix}-M^{-1}C & 0\\ I_{2X2} & 0\end{pmatrix}\begin{pmatrix}\dot{\theta}_1\\ \dot{\theta}_2\\ \theta_1\\ \theta_2\end{pmatrix} + M^{-1}\begin{pmatrix}\tau_1\\ \tau_2\\ 0\\ 0\end{pmatrix} - M^{-1}\begin{pmatrix}G\\ 0\end{pmatrix}$		Proportional Integral Derivative Controller

Understanding Simulation - 2R Robot Arm

- ode45 was used to simulate the system and a PID controller was used for setpoint and trajectory tracking
- Momentum was calculated both, from torque integration and velocities for verification



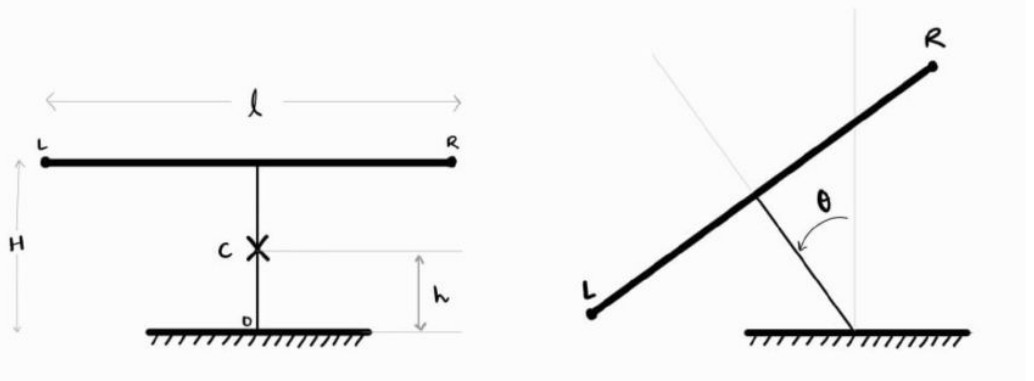
Total energy of a freely falling system



Difference in total energy calculated from momentum integration and kinematics

1D balancing - Mechanics

A one dimensional balancing robot free to rotate about ground contact point



Kinematics:

$$\vec{r}_C = -h(\sin\theta\hat{i} - \cos\theta\hat{j})$$
$$\dot{\vec{r}}_C = \vec{v}_C = -h\dot{\theta}(\cos\theta\hat{i} + h\sin\theta\hat{j})$$
$$\omega = \dot{\theta}$$

Dynamics:

Kinetic Energy: $KE = \frac{I_0\omega^2}{2} = \frac{I_0\dot{\theta}^2}{2}$

Potential Energy: $PE = Mgh\cos\theta$

Lagrange: $L = KE - PE$

Euler Lagrange Equation: $\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = \tau$

Upon solving,

$$I_0\ddot{\theta} - Mgh\sin\theta = \frac{(u_R - u_L)l}{2}$$

1D balancing - Mechanics

Constraints:

Robot is always in touch with the ground

$$(u_L + u_R)\cos\theta < \kappa M g \qquad \kappa \leq 1$$

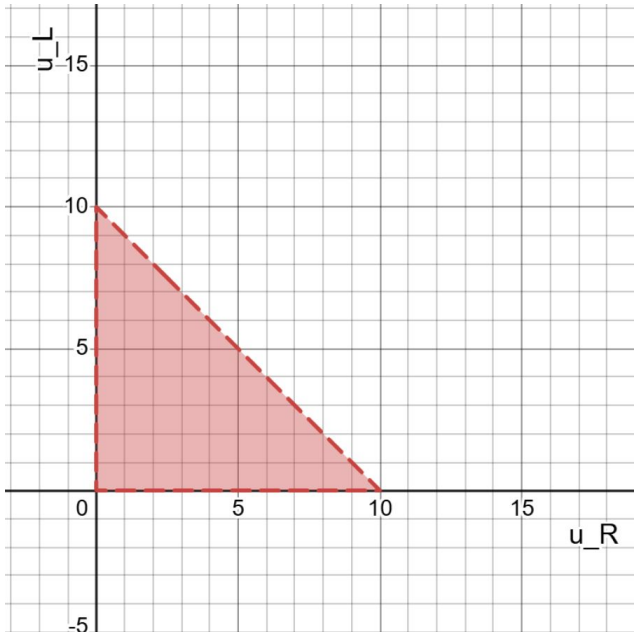
Only one point of the robot is in contact with the ground

$$|\theta| < \frac{\pi}{2} - \tan^{-1}(l/2H)$$

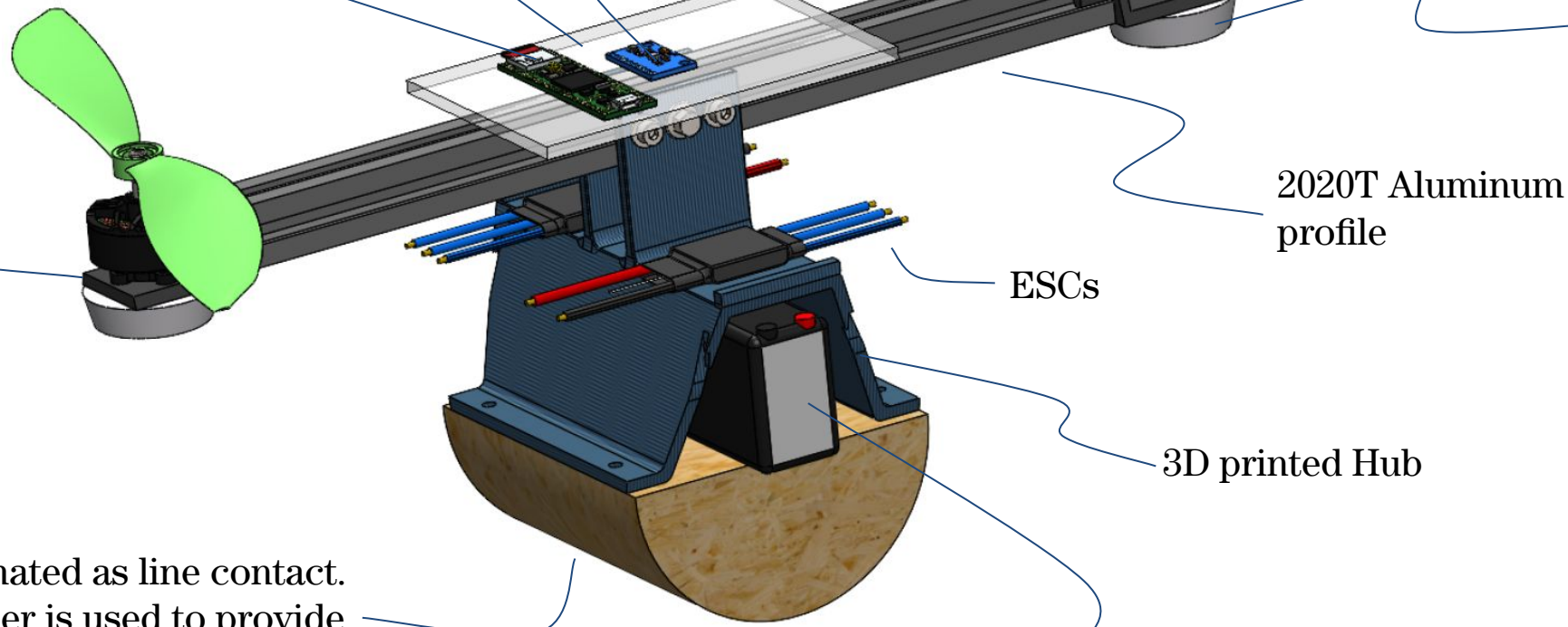
Propellers can produce thrust in one direction

$$u_L, u_R > 0$$

Solution Space:

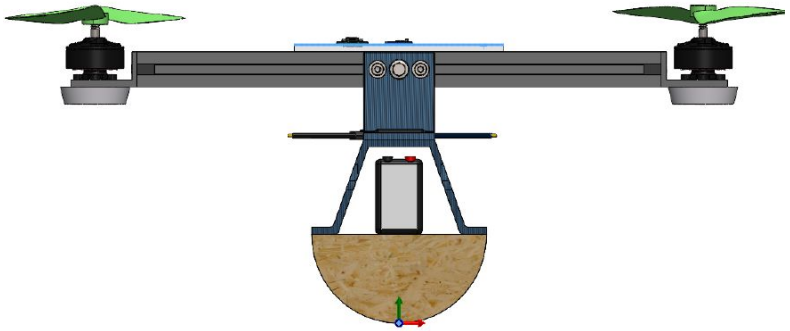


Physical Model

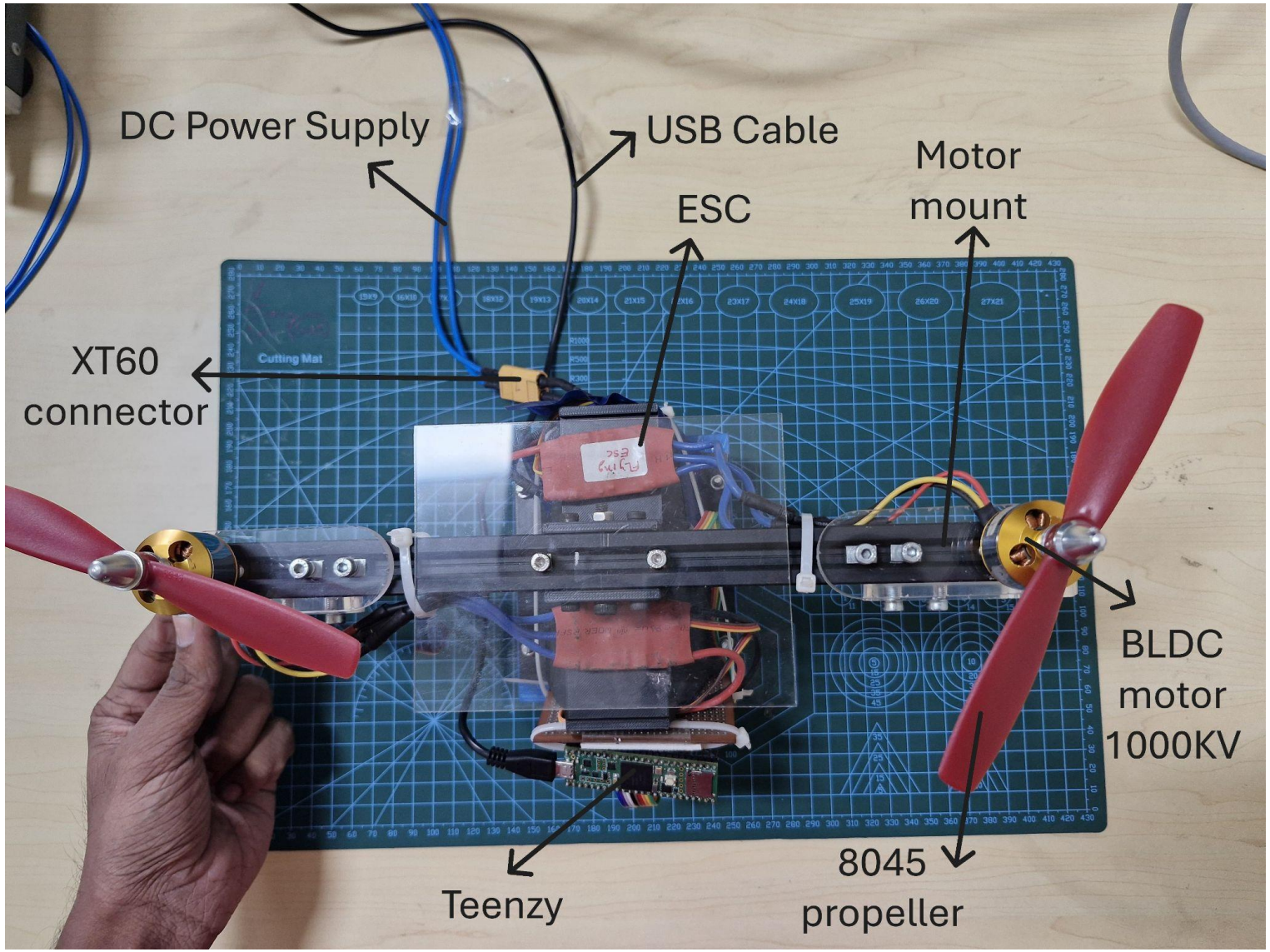


Physical parameters

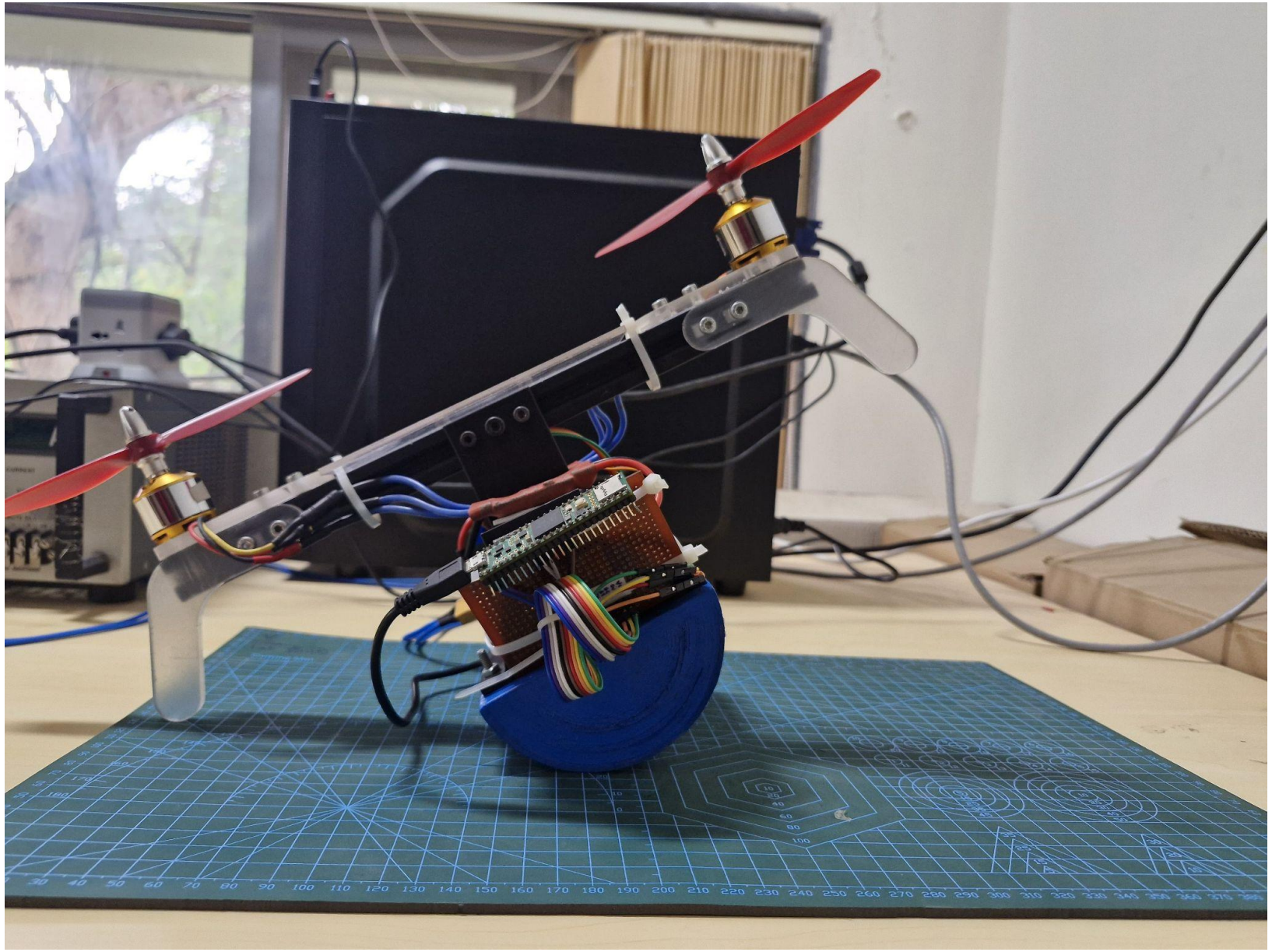
Description	Notation	Units	Current Value
Distance between propellors	l	mm	348
Height of COM above the ground	h	mm	90
Height of the lowest point of the motor-motor mount assembly	H	mm	124
Moment of inertia about COM through nominal axis	I	Kg/m^2	0.005
Mass	M	Kg	1.05



Hardware


DC Power Supply
USB Cable
ESC
Motor mount
XT60 connector
BLDC motor 1000KV
Teenzy
8045 propeller

Hardware



Controller

PD controller:

Error dynamics $$\ddot{e} + K_d\dot{e} + K_p e = 0$$

Commanded acceleration $$\ddot{\theta}_{des} = \ddot{\theta}_t - (K_p e + K_d \dot{e})$$

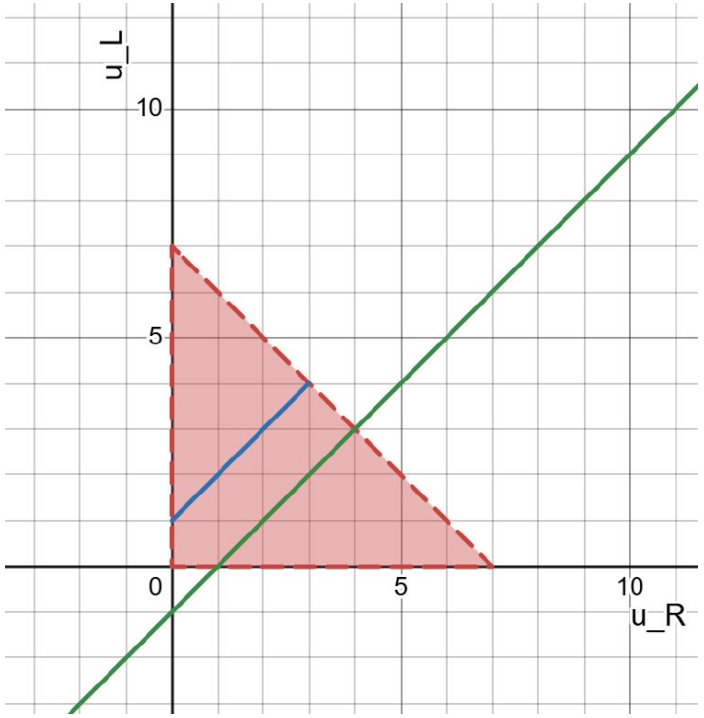
Commanded thrust $$u_R - u_L = u_0 \qquad u_0 := u_d - \alpha_1 u_m \sin\theta$$

$$u_d := 2I_0\ddot{\theta}_{des}/l$$
$$u_m := \kappa M g$$
$$\alpha_1 := 2h/\kappa l$$

With Constraints:

Taking into account:

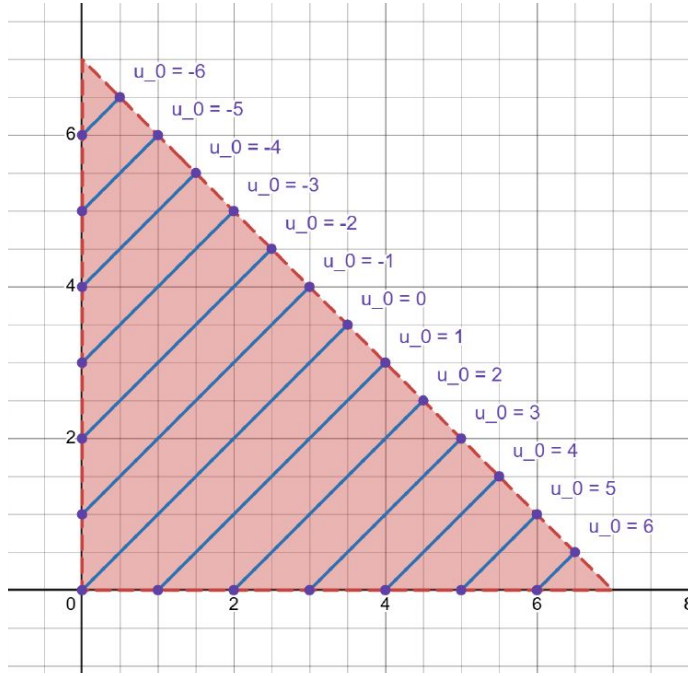
1. Normal Contact > 0
2. Thrust > 0



Green line is a solution of the PD controller without constraints. Blue line is a solution of the PD controller with constraints

$$0 < u_R < \frac{u_m \sec\theta + u_0}{2}$$

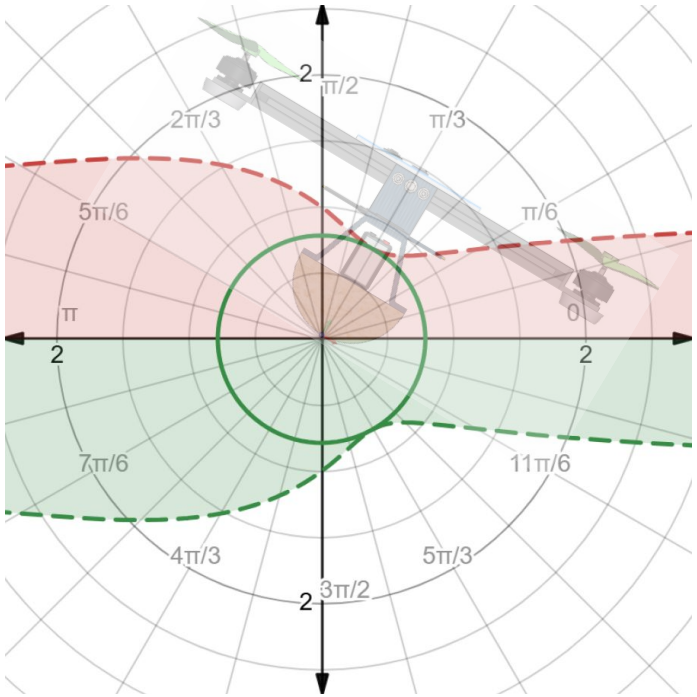
$$0 < u_L < \frac{u_m \sec\theta - u_0}{2}$$



Solution space for varying u_0

Controller: Existence of a solution

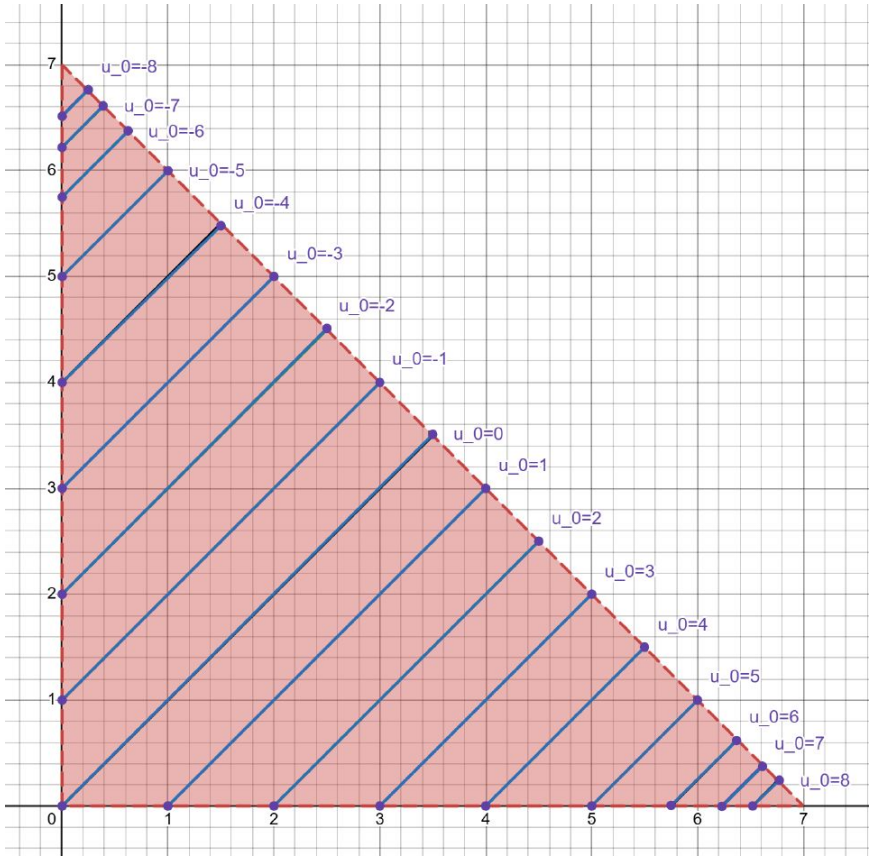
Since u_d depends only on controller gains and trajectory inputs, we try to constrain them so that the controller is always able to find a solution



Solution space of u_d as a function of time

- For a given orientation, the red and green lines represent the maximum and minimum possible values of u_d permissible
- To guarantee a solution, we will restrict u_d to inside the green circle
- Analytical solution for radius of the green circle does not exist
- With numerical estimates and trajectory acceleration bounds, estimate Kp and Kd

Saturation of thrust is required



Required saturation of u_0

Controller: Saturation

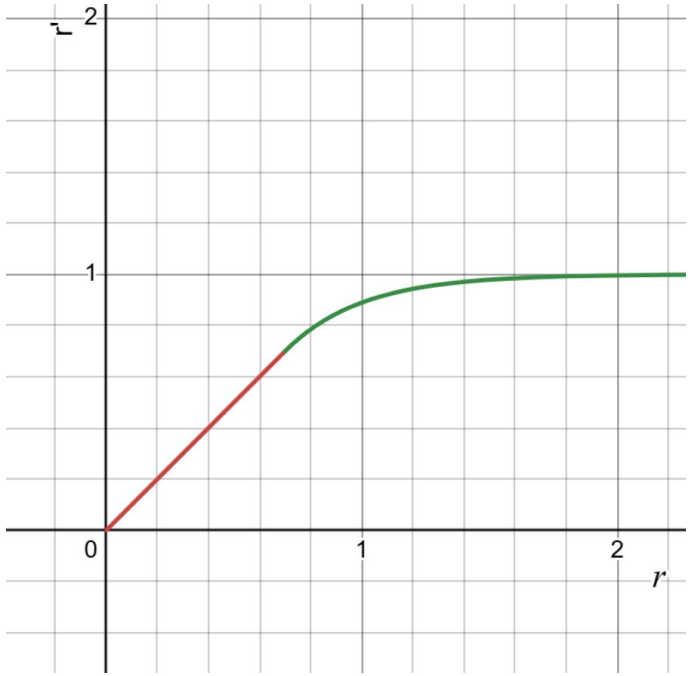
u_0 is saturated the prevent individual thrusts to cross the constraints. $r = |u_0 \cos\theta / u_m|$

Saturation function requirements:

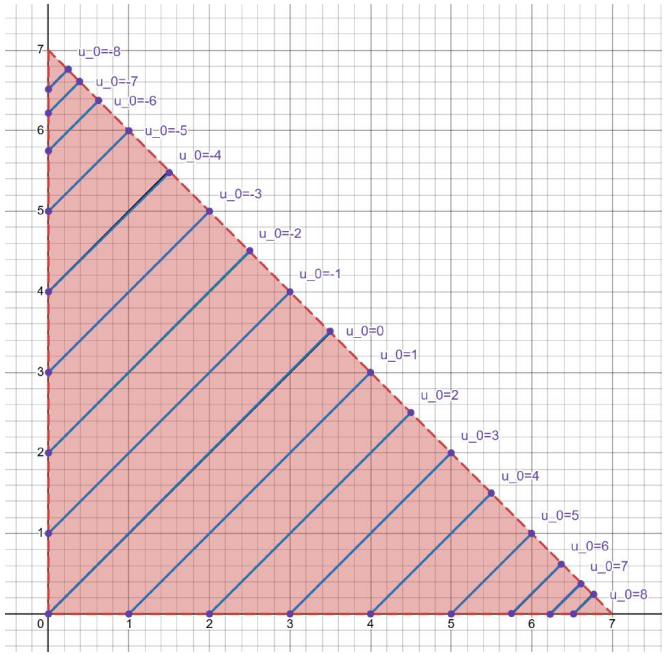
- Linear to a large extent
- C0 and C1 continuity for smooth trajectories
- Asymptotically approaching saturation limits

Exponential function is chosen with linearity in central region

$$r' = \begin{cases} r & 0 \le r \le \alpha_2 \\ 1 - (1-\alpha_2)e^{\frac{(\alpha_2 - r)}{1-\alpha_2}} & r > \alpha_2 \end{cases}$$



Exponential saturation function

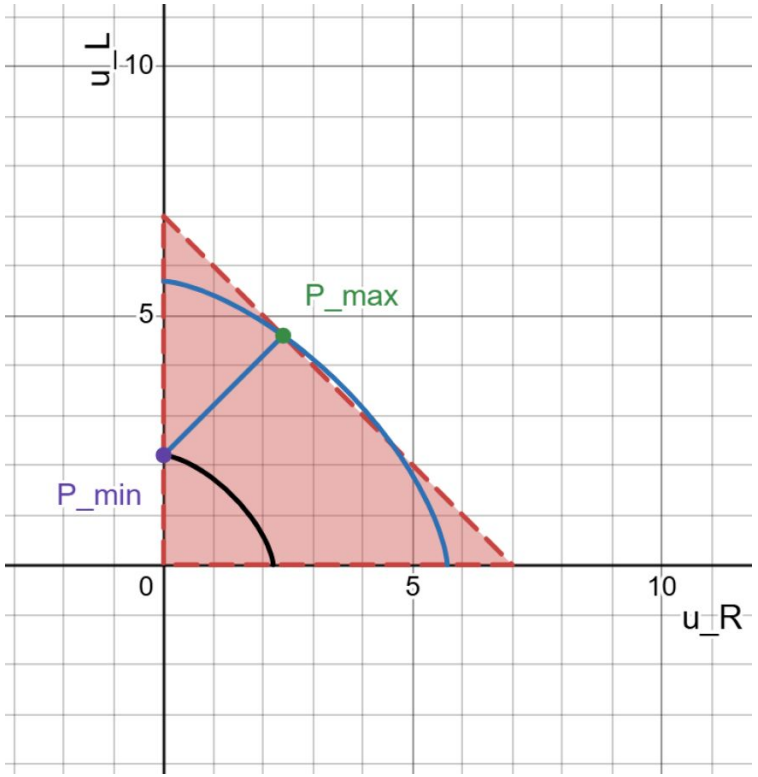


Exponential saturation of u_0

Controller: Selecting Optimal Gains

Optimizing power requirements:

$$P = \frac{k_T}{k_F^{3/2}}(u_R^{3/2} + u_L^{3/2})$$

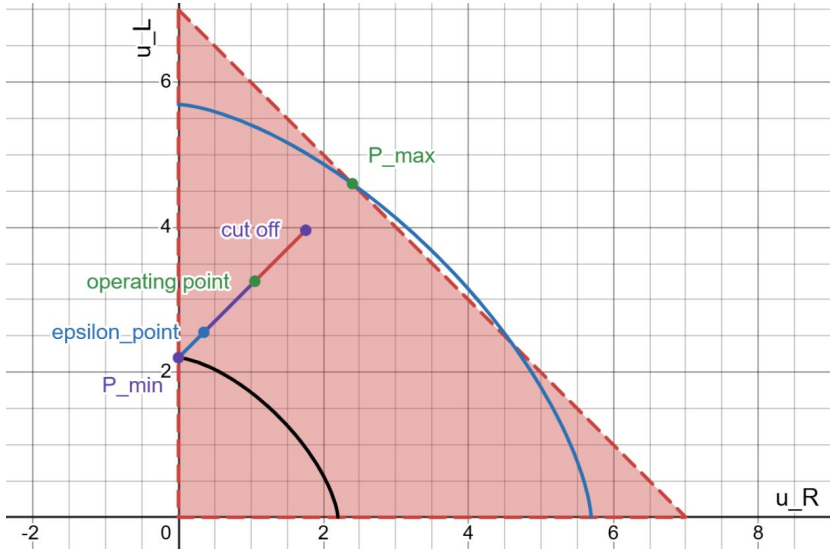


Minimum and maximum power points

- We want atleast ε fraction of power in each motor for propellers to run
- We want atleast β fraction of power to be available on each side of the operating point

$$\epsilon + 2\beta < 1$$

- Since power is non-linear in thrusts and interpolating power to get thrust is computationally heavy, we will approximate our power requirements to vary linearly with thrusts



Identification of operating point

$\epsilon = 0.15, \beta = 0.3$

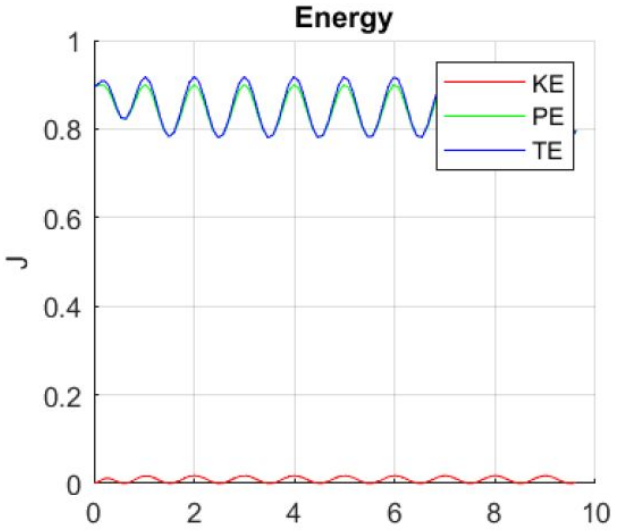
Controller implementation

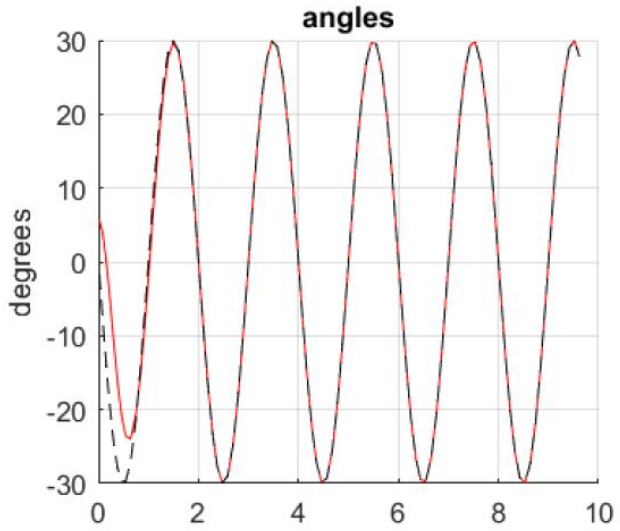
$$PID = -(K_p * e + K_i \int edt + K_d \dot{e})$$

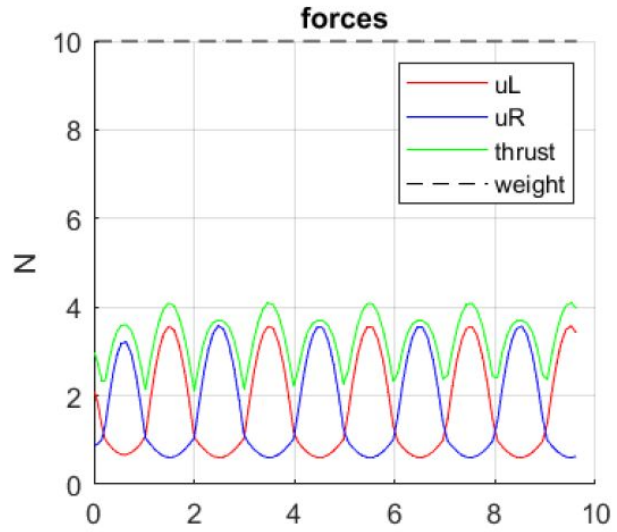
$$\omega_L = max(0, -PID)$$

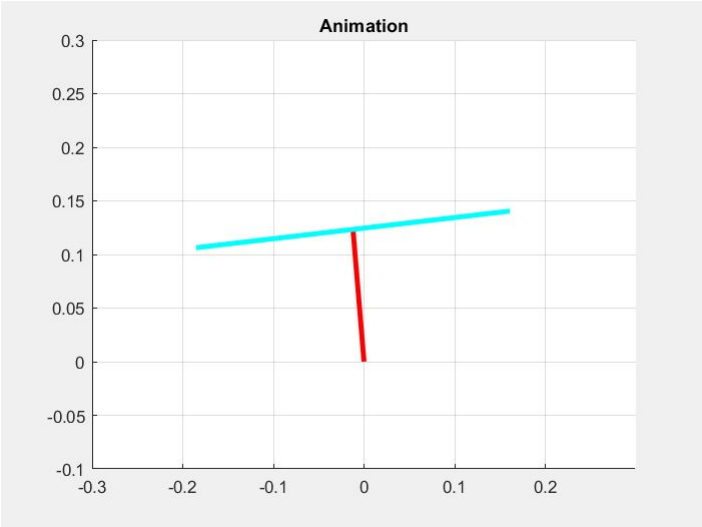
$$\omega_R = max(0, PID)$$

Results: Simulation







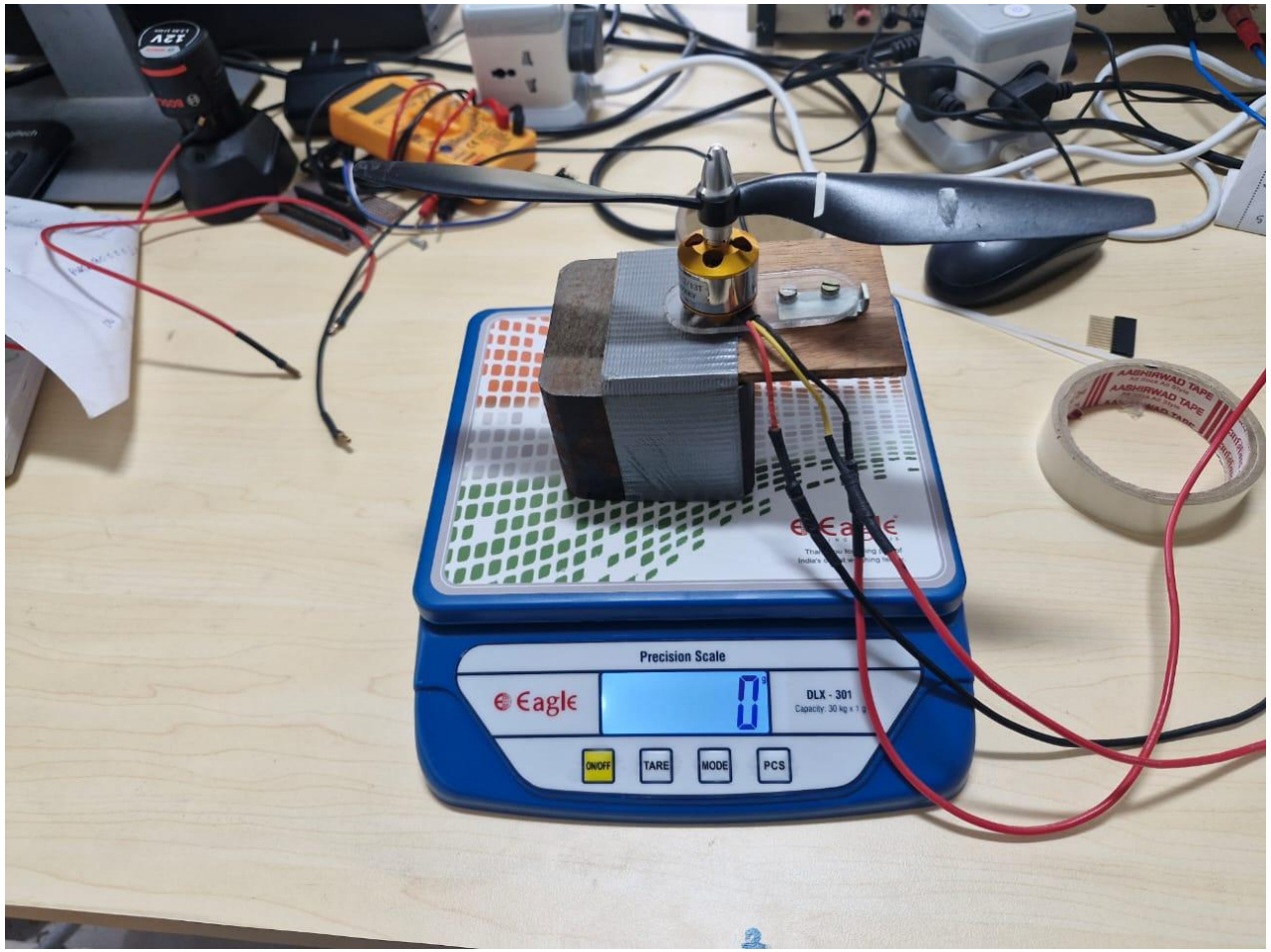


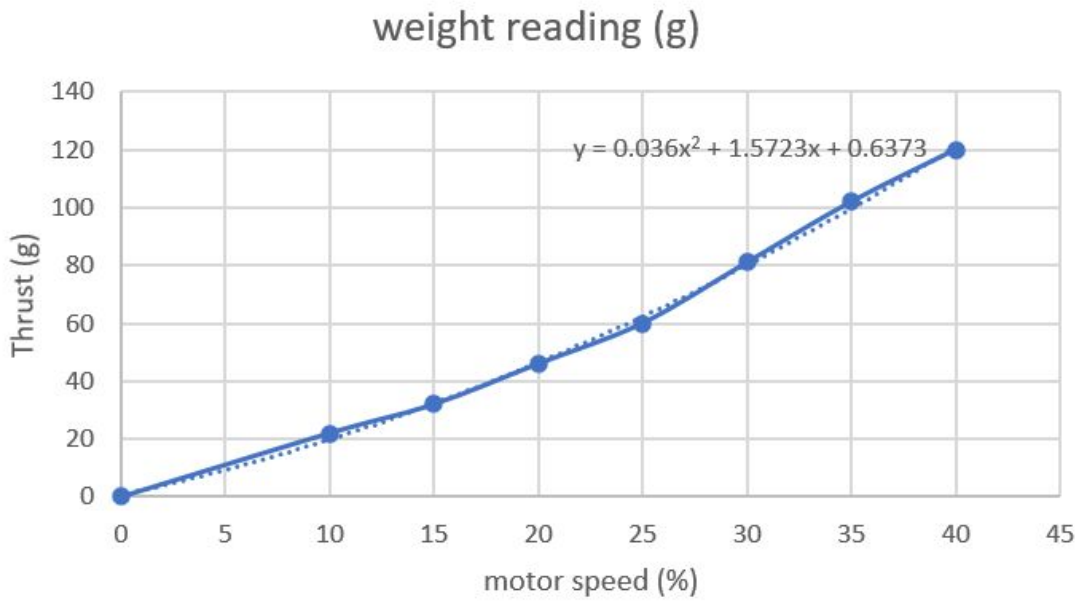
Results from trajectory tracking simulation in MATLAB:

- ode45 used for system simulation
- 0.5Hz sinusoidal trajectory requirement
- Non zero initial conditions

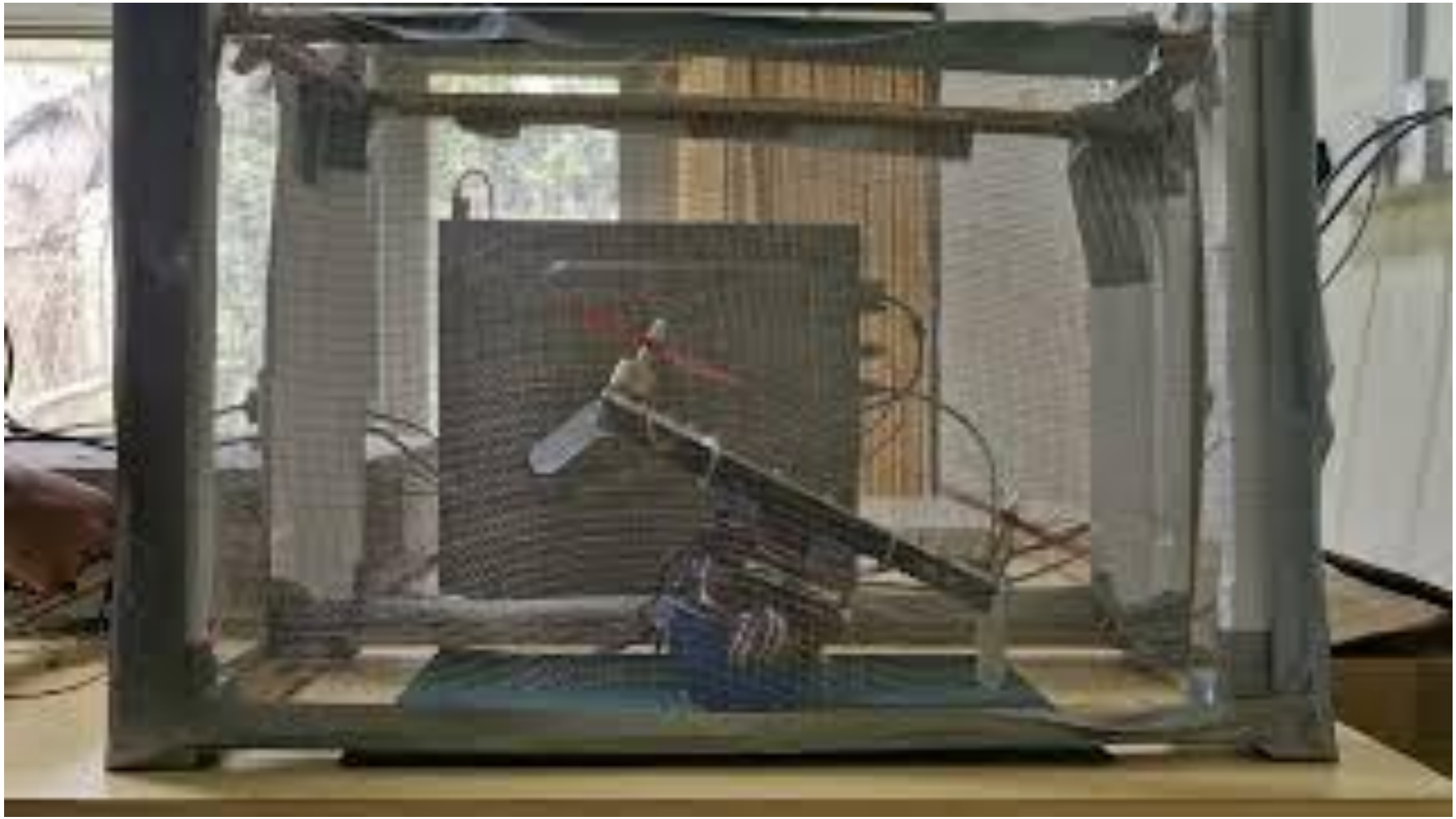
- We observe thrusts are always positive and much lesser than weight
- Thrusts get saturated for low requirements
- Continuous variation of thrusts but with abrupt change in jerk
- Kinetic energy has lower magnitude oscillations compared to potential energy
- Tracking catches up in ~1s

Results: Static Thrust Test

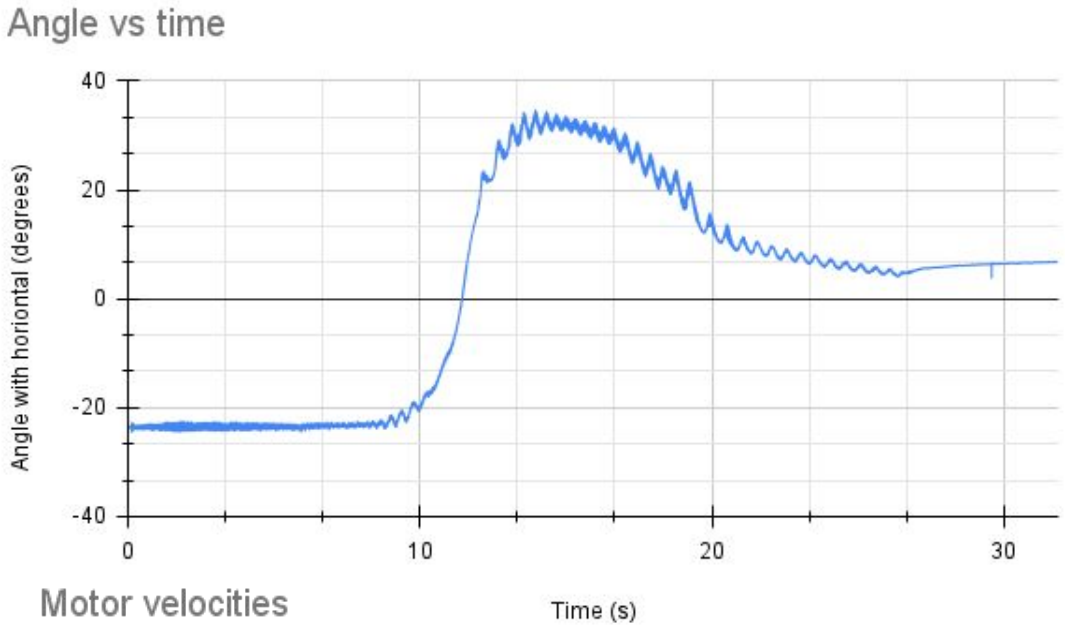


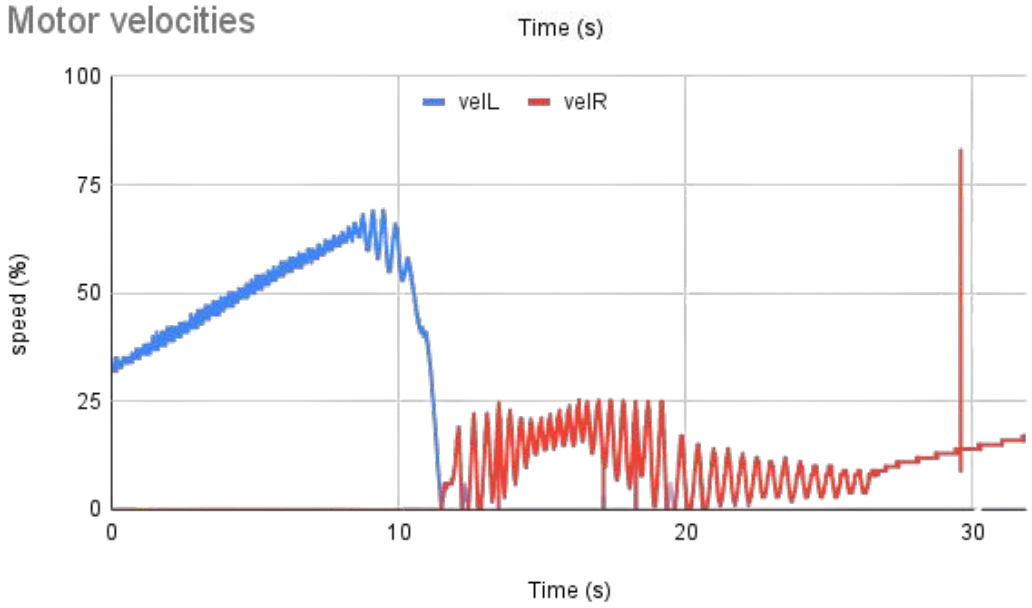


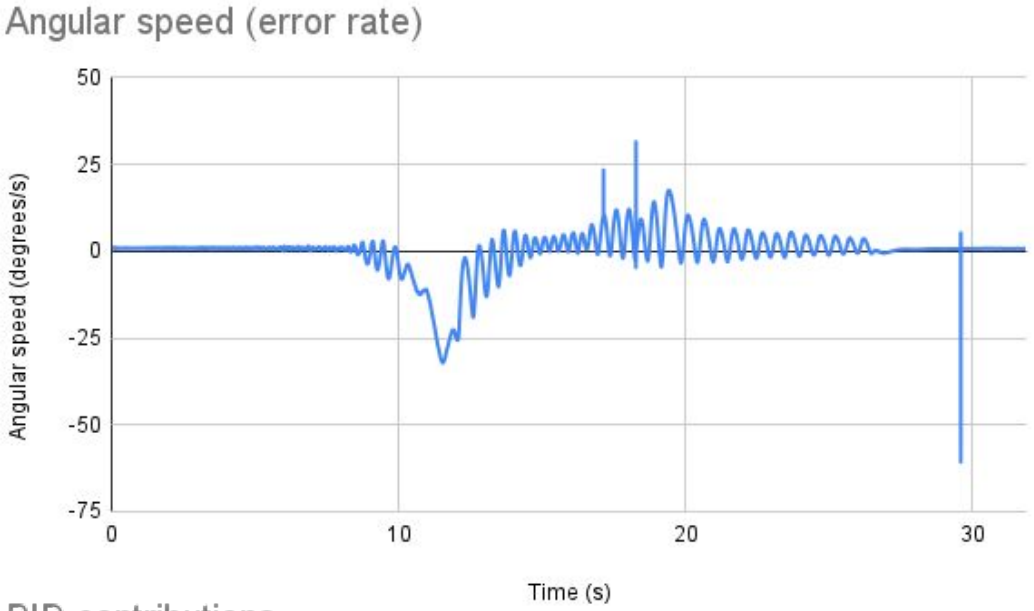
Results: Hardware

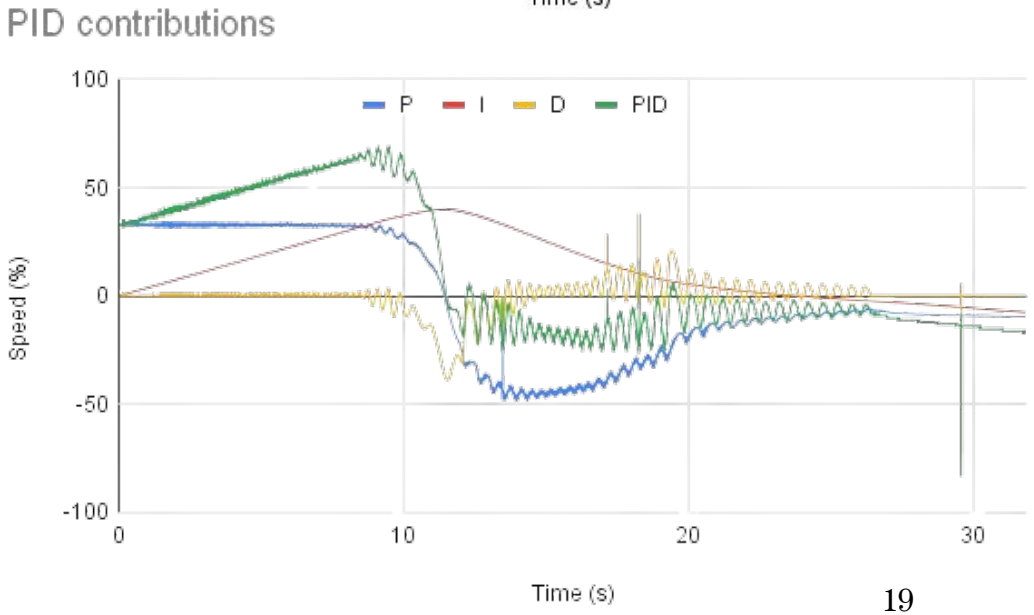


Results: Hardware









Future Work

- Increase Propeller size
- Use Simulated Control Law
- Implement Yaw Control

Understanding Simulation - 2R Robot Arm

Dynamics:

$$\mathbf{M}(\theta_1,\theta_2)\begin{pmatrix}\ddot{\theta}_1\\ \ddot{\theta}_2\end{pmatrix}+\mathbf{C}(\theta_1,\theta_2,\dot{\theta}_1,\dot{\theta}_2)\begin{pmatrix}\dot{\theta}_1\\ \dot{\theta}_2\end{pmatrix}+\mathbf{G}(\theta_1,\theta_2)=\begin{pmatrix}\tau_1\\ \tau_2\end{pmatrix}$$

$$\mathbf{M}(\theta_1,\theta_2)=(I_2+m_2a_2^2)I_{2X2}+\begin{pmatrix}I_1+m_1a_1^2+m_2(l_1^2+2l_1a_2\cos(\theta_2)) & m_2l_1a_2\cos(\theta_2)\\ m_2l_1a_2\cos(\theta_2) & 0\end{pmatrix}$$

$$\mathbf{C}(\theta_1,\theta_2,\dot{\theta}_1,\dot{\theta}_2)=m_2l_1a_2\sin\theta_2\begin{pmatrix}-\dot{\theta}_2 & -(\dot{\theta}_1+\dot{\theta}_2)\\ \dot{\theta}_1 & 0\end{pmatrix}\quad \mathbf{G}(\theta_1,\theta_2)=m_2ga_2\cos(\theta_1+\theta_2)(\mathbf{e_x}+\mathbf{e_y}+g\cos\theta_1\begin{pmatrix}m_1a_1+m_2l_1\\ 0\end{pmatrix}$$

State Space:

$$\begin{pmatrix}\ddot{\theta}_1\\ \ddot{\theta}_2\\ \dot{\theta}_1\\ \dot{\theta}_2\end{pmatrix}=\begin{pmatrix}-M^{-1}C & 0\\ I_{2X2} & 0\end{pmatrix}\begin{pmatrix}\dot{\theta}_1\\ \dot{\theta}_2\\ \theta_1\\ \theta_2\end{pmatrix}+M^{-1}\begin{pmatrix}\tau_1\\ \tau_2\\ 0\\ 0\end{pmatrix}-M^{-1}\begin{pmatrix}G\\ 0\end{pmatrix}$$

ode45 was used to simulate the system and a PID controller was used for setpoint and trajectory tracking.

Momentum was calculated both, from torque integration and velocities for verification