Attitude Control of a 1D Dalancing Bot

A thesis submitted in fulfilment of the requirements

for the degree of B.Tech (Honors)

by

Kushal Agarwal

DEPARTMENT OF MECHANICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY BOMBAY

May 2025

Contents

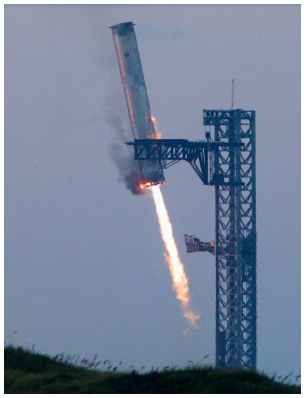
List of Figures

Chapter 1

Introduction

1.1 Background

Attitude control refers to the process of controlling the orientation of a rigid body in space relative to a reference frame, such as Earth's gravitational field or an inertial frame. In the context of mobile robotics and aerospace engineering, attitude control is critical for stabilizing and guiding vehicles, including drones, spacecraft, and autonomous robots. These systems rely on sensors like gyroscopes and accelerometers, alongside feedback control mechanisms, to maintain or change orientation. Classic approaches to attitude control include Proportional-Integral-Derivative (PID) controllers, Linear Quadratic Regulators (LQR), and model-based control strategies. Recent advancements in machine learning and reinforcement learning have also started to offer promising approaches to adaptive control in dynamic environments. By studying attitude control through robotic platforms, we can gain deeper insights into stability, real-time control, and error correction. While most attitude control problems address controlling the orientation of a body in three dimensions, the coupled model of these orientations is complex. However, concepts from a controlling orientation in one plane can build strong concepts for developing a full-fledged attitude controller.



((A)) SpaceX rocket booster

((B)) MQ-9 Reaper of US Air Force

FIGURE 1.1: Attitude control used in civil and defense applications

1.2 Motivation

The motivation behind this project is to explore the complexities of attitude control in a practical and accessible manner. Balancing robots, which require real-time adjustments to maintain stability, provide an ideal testbed for understanding how various control algorithms respond to dynamic forces and sensor feedback. With a focus on autonomous balancing, this thesis aims to advance knowledge in control theory, particularly in how systems can adapt to real-time feedback while maintaining stability in the presence of disturbances. To develop a robust controller that can control the orientation of an object in three-dimensional free space, we first built a robust controller for precise control of the orientation of our robot in its trivial plane. Even a simple system like this presents several complexities as we take into account constraints arising from motors, propellers, contact surface and physical limits. We want to build a controller that understands and is robust to this phenomenon and guarantees the robot is able to hold as well as follow a trajectory of orientation without colliding into the contact surface and being aware of the limitations posed by its hardware. Apart from this, the system also provides exposure to control of overactuated systems and building contingencies in our system by leveraging this advantage. This project will not only contribute to the body of knowledge in robotics

and control theory but also offer insights into the practical applications of autonomous systems, such as drones and self-balancing vehicles, where attitude control is crucial for performance and safety.

1.3 Organization of the Thesis

The first chapter introduces the problem and the aim of the project with an overview of what to expect in the report. The second chapter discusses the mechanics of the model, as well as the kinematics and dynamics of the system. The third chapter discusses the physical model of the robot along with the hardware used. The fourth chapter takes you through the controller design and the features built into it. The fifth chapter provides results with information on the simulation environment. The final chapter provides a conclusion and scope of future work.

Chapter 2

Mechanics of the Model

2.1 Link structure and definitions

A one dimensional balancing robot can be modelled as 2 links attacked perpendicular to each other with the corners of one of the links in contact with the ground. We want the robot to always stay in contact with the ground, so we denote the ground to be a plane passing through this point O.

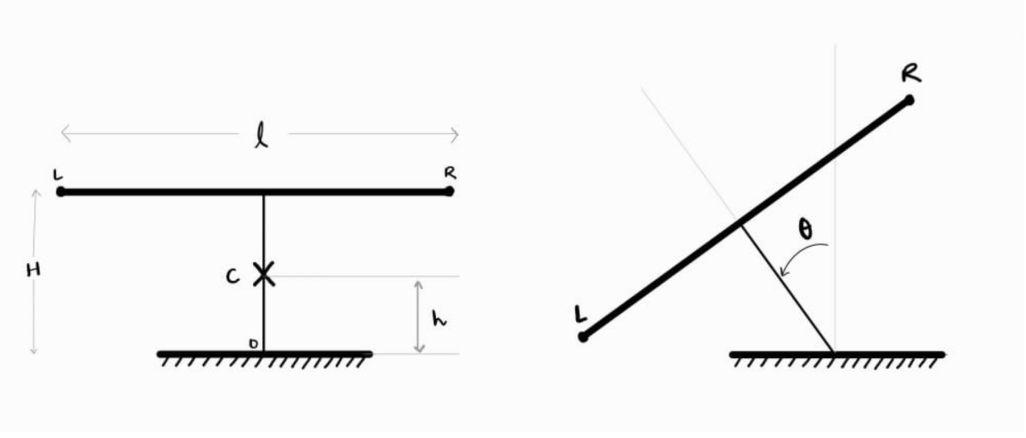


FIGURE 2.1: 1D balancer

Since the system is symmetric, we assume the center of mass of the system, including links, motors, propellers, battery and electronics to be at point C at a constant distance h from

point O. Let the line joining O and C make an angle θ in the counterclockwise direction from a line perpendicular to the ground plane and passing through point O. We define the ends of the other links as points where the effort is acted upon. This point should lie of the axis of the motor and the propellor. Further we define this end as the closest point on this line to point O such that it lies inside the physical robot. This will become the additional point of contact of the robot with the ground when the robot is resting on its side. Let the distance of this point frompoint O along line OC be H and the distance between two such points be l.

We define a coordinate system S to serve as our inertial coordinate system. This is defined with its origin at point O, z-axis coming out of the plane and y-axis opposite to the action of gravity. Correspondingly, x-axis is towards right.

2.2 Kinematics

The position vector of the center of mass in S is

$$\vec{r}_C = -h(\sin\theta\hat{i} - \cos\theta\hat{j}) \tag{2.1}$$

Orientation of the system is given by θ.

The velocity of the center of mass of the system is given by,

$$\dot{\vec{r}}_C = \vec{v}_C = -h\dot{\theta}(\cos\theta\hat{i} + h\sin\theta\hat{j}) \tag{2.2}$$

The angular rate of the system is given the rate of change of the orientation, denoted by ω.

$$\omega = \dot{\theta} \tag{2.3}$$

2.3 Dynamics

Let I be the rotation inertia of the system about center of mass along z-axis (I_{zz}) by I. The inertia of the system about point O is given by,

$$I_0 = I + Mh^2 \tag{2.4}$$

The kinetic energy of the system is

$$KE = \frac{I_0\omega^2}{2} = \frac{I_0\dot{\theta}^2}{2} \tag{2.5}$$

The potential energy of the system is

$$PE = Mgh\cos\theta \tag{2.6}$$

taking ground as a reference potential.

The dynamics of the system are calculated using Euler-Lagrange equation.

$$L = KE - PE \tag{2.7}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = \tau \tag{2.8}$$

Where τ is the torque that actuates the angle θ. Let u_L and u_R be the thrusts of the left and right propellers, respectively. These forces act of the balancer in the upward direction. Then the expression for τ is given by $\tau = (u_R - u_L)l/2$. Solving the Euler-Lagrange equation, we get

$$I_0\ddot{\theta} - Mgh\sin\theta = \frac{(u_R - u_L)l}{2} \tag{2.9}$$

To ensure that the balancer is in contact with the ground at exactly one point (O), the following conditions have to be satisfied.

$$(u_L + u_R)\cos\theta < \kappa M g \tag{2.10}$$

$$|\theta| < \frac{\pi}{2} - \tan^{-1}(l/2H) \tag{2.11}$$

where κ is parameter with the constraint $\kappa \leq 1$ Additionally, since motor thrust is required to only be in a single direction, we get some additional constraints.

$$u_L, u_R > 0 \tag{2.12}$$

In total, the feasible solution of the thrusts are restricted in a convex triangular domain as shown below.

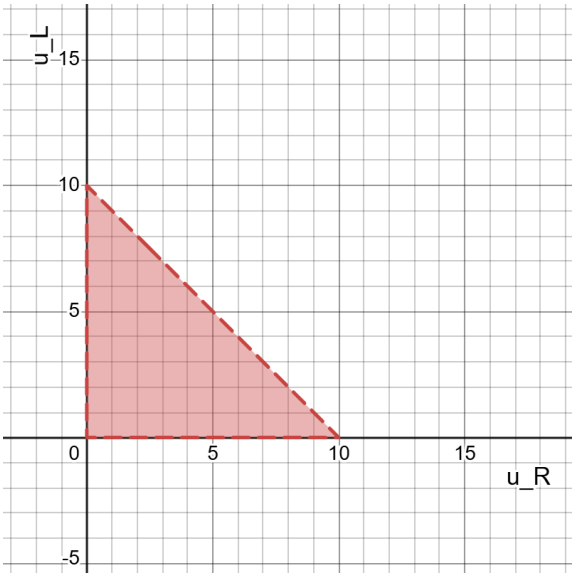


FIGURE 2.2: Domain of thrusts for $M = 1$, $\theta = 0$ and $\kappa = 1$

2.4 State Space Representation

The system can be represented with the following state space representation.

$$\dot{x} = Ax + Bu + G \tag{2.13}$$

$$y = Cx + Du \tag{2.14}$$

$$x = [\theta, \dot{\theta}]^T \qquad u = [u_R, u_L]^T \tag{2.15}$$

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \qquad B = \frac{l}{2I_0}[1, -1]^T \qquad G = \frac{Mgh\sin\theta}{I_0}[0, 1]^T \tag{2.16}$$

$$C = [1, 0] \qquad D = 0_{1X2} \tag{2.17}$$

Chapter 3

Physical Model

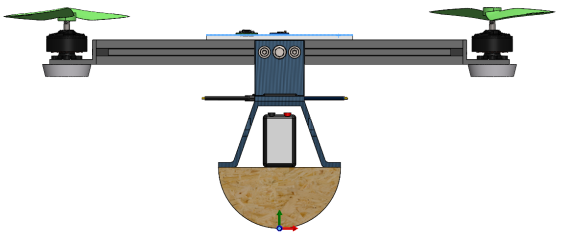


FIGURE 3.1: 1D balancer

This chapter discusses the design - various features and constraints - put of the physical model of the attitude control bot.

3.1 Design Iteration

To achieve the aim of controlling one orientation of the bot, the design had to be able to move freely along one particular axis. A 2D model for the same was made where we required a point contact of the bot with the ground and the bot (considering rigid) can rotate about the axis passing through the point of contact perpendicular to the surface. Further, the model was extruded to 2D, the bot now having a line contact with the ground.

3.2 CAD Model

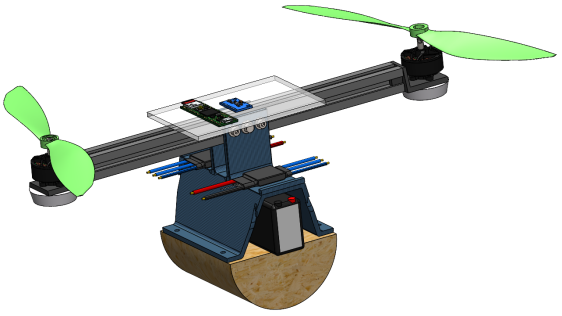


FIGURE 3.2: 1D balancer

3.2.1 Contact Surface

The contact surface required was such that it can be abstracted to a single point in a 2D abstraction of the bot. In the full model, we have line contact. Since the entire load of the bot will be passing through this one line, wear and damage were expected. Initial testing will a wooden material confirmed this. An obvious upgrade was thought to make the contact body out of metal but the metal contact line would not be able to provide enough friction to prevent any possible yaw in the bot. To tackle this, rubber has been used to serve an ideal material for contact. The contact surface was changed to cylindrical, and the deformation of the rubber would significantly increase available friction. It would also damp vibrations from the bot.

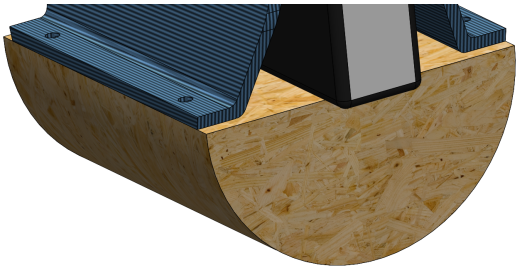


FIGURE 3.3: 1D balancer

3.2.2 Arms

The arm of the bot is made of 2020T aluminum. The profile and material provide a lightweight yet rigid base to mount the propellers. They are readily available, easy to modify in length and easy to mount. It is also very easy to mount and unmount things on the arms.

3.2.3 Motors and Propellors

BLDC motors are used to provide the range of speeds required by nominal propellers to produce thrust. The motor and propellors in the CAD are representative. A motor mount made of PLA, using FDM printing, mates with the profile of the 2020T and provide mount holes for the motor. On the other side of the mount a hole is used to mount stoppers preventing the propellers from ever thought the ground.

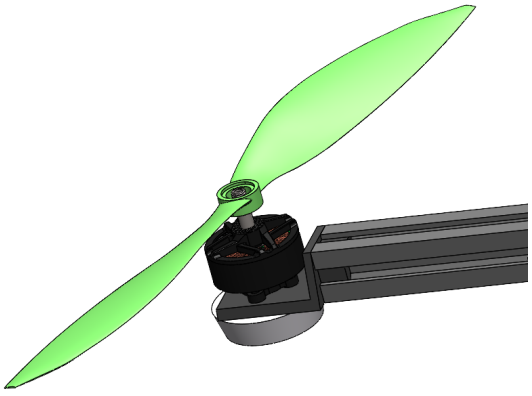


FIGURE 3.4: 1D balancer

3.2.4 Electronics

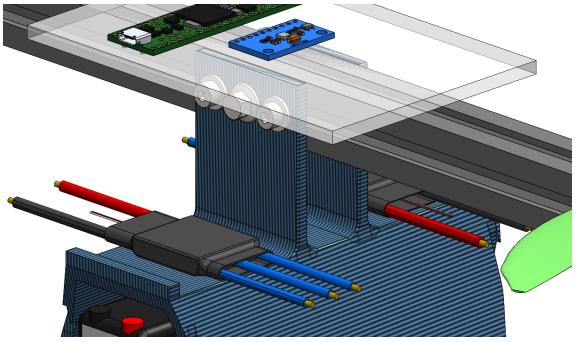


FIGURE 3.5: 1D balancer

The electronics consist of an Electronic Speed Controller, a power distribution board, an IMU sensor and a Teenzy. The Teenzy, IMU and placed on top of a plate. The ESC and power distribution board are placed on top of the battery hull.

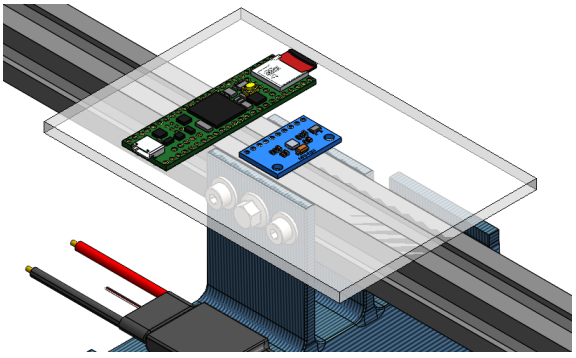


FIGURE 3.6: 1D balancer

3.2.5 Design Parameters

The various parameters of the design are mentioned in the table below.

Table 3.1: Design 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

3.3 Hardware

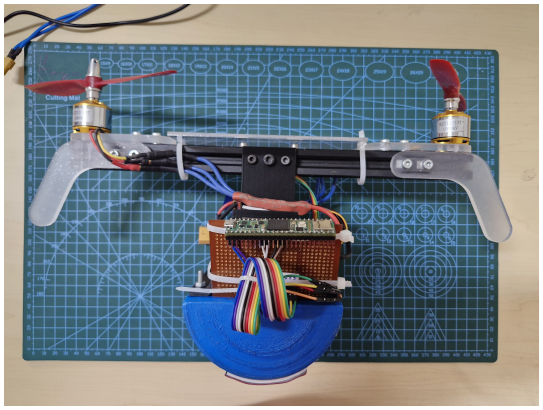
3.3.1 Mechanical

3.3.1.1 Base

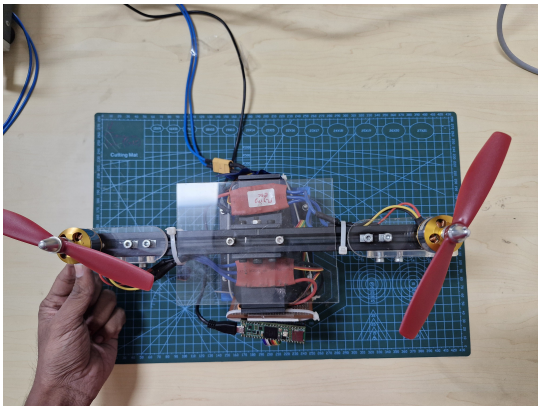
The bottom part of the balancer was made out of a nylon rod. The density of nylon used was much higher than the average density of the model. This is desirable as the location of the center of mass is closer to the contact surface and this naturally increases the stability of the system.

3.3.1.2 Hub

The middle part was fabricated using FDM 3D printing. The material used is PLA and printed on a Bamboo Labs machine at 40% infill.



((A)) Front View



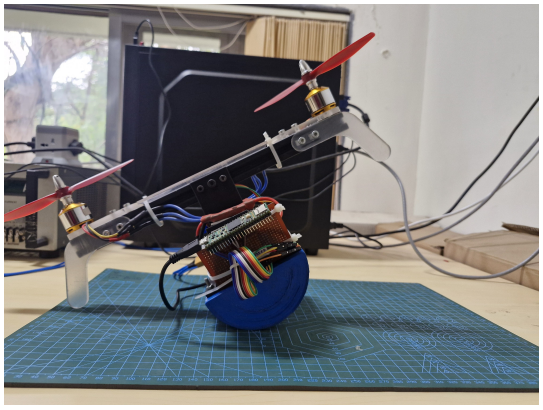
((B)) Top View

FIGURE 3.7: Hardware: Propeller Balancer

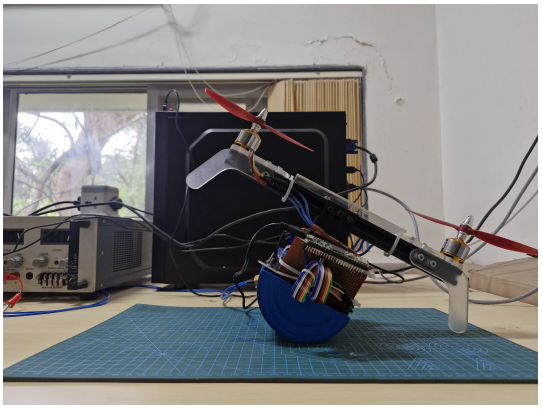
3.3.1.3 Mounts and Endstops

The mounts for the motors were modified and laser cut out of 5mm acrylic sheet instead. This allowed us to modify the inter-shaft distance depending on the size of the propeller. The drawback is that the model may not be symmetric.

In addition, endstops were also laser cut from a 5mm acrylic sheet. The current configuration of endstops allow us to modify the angle limits that the model permits. The drawback is that the angle limits on either sides may not be equal. Experimentally it was observed that the angle limits were -24°to 30°as measured from vertical.



((A)) Left-side limit



((B)) Right-side limit

FIGURE 3.8: Physical Angle limits by Endstops

Tests were carried out with 2 kinds of propellers: 1045 and 8045.

The position of the IMU and the microcontroller were changed so that the electronics stay clear of the fast moving blades and at the same time remains accessible for reset or cabling check.

3.3.2 Electronics

3.3.2.1 Power Supply

Keeping in the mind the application of such a setup in teaching control theory of overactuated system, the power supply was changed from a Li-Po battery to a dedicated external power supply. A male XT60 connector is available at the rear end of the balancer for connection to the external power supply. The hub for the battery is now used to house a power distribution board and wiring.

3.3.2.2 Power distribution board

The power distribution board used is the CC3D ZMR BEC. It is used to provide a stable 5V power supply to the microcontroller. It takes a 12V input. It also shorts to power supply output to the ESC's to its power input.

3.3.2.3 Electronic Speed Controllers (ESCs)

The ESCs used here are the SimonK 30A ESC. These are required to produced appropriate voltages at the three terminals of the BLDC motors. They contain the timing circuit. They take speed as a PWM signal, essentially abstracting the motor+ESC system as a velocity control servo. Throttle is provided as a PWM signal. Speed is proportional to the width of the PWM pulse. A 50Hz PWM is used and has a 20ms period.

Table 3.2: Serco and ESC/BLDC PWM effects

Duration of high signal per period	Effect in Servo	Effect in BLDC
1ms	0°	min RPM
2ms	180°	max RPM

Every ESC has to caallibrated. The input PWM frequency can vary from 50-60Hz and min max pulse width can vary about 1ms and 2ms respectively.

3.3.2.4 IMU

The IMU used is the MPU6050. Calibration is done via IMU_Zero.ino. In an example file from the MPU6050 library by Electronics Cats. The sensor communicates with I2C communication.

3.3.3 Microcontroller and connections

Teensy 4.1 microcontroller has been used.

The system has only two electrical connections from the outside. One is the DC power supply which is easily detachable with the XT60 connector and the other is a USB cable for communicating with the microcontroller.

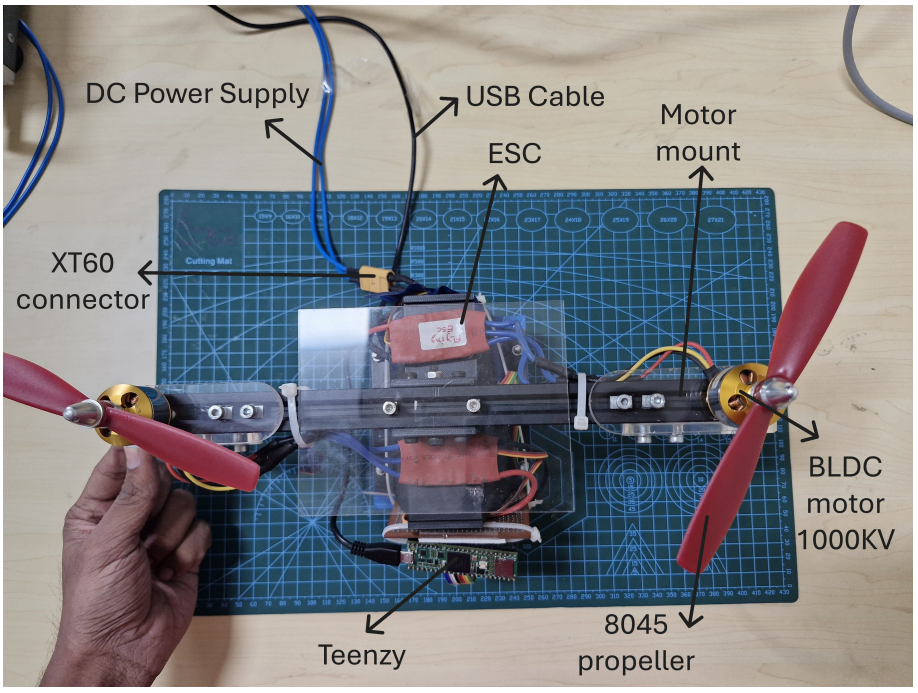


FIGURE 3.9: Labelled parts

3.3.4 Safety Cage and Basestation

Propellers run at extremely high speeds and pose a danger for the operator/learner. Hence, to provide a safe environment during testing, a safety cage was made. The pillars were made of packing thermocol and the net was made out of steel. Since these are the parts that the propellers would interact with the most likely, these are made of soft material which can either get easily cut (in case of thermocol) or easily bent (in case of steel mesh) to absord the energy. In addition, the power supply is placed close to the user and away from he setup for ready access to either supply or cut the power as and when needed.

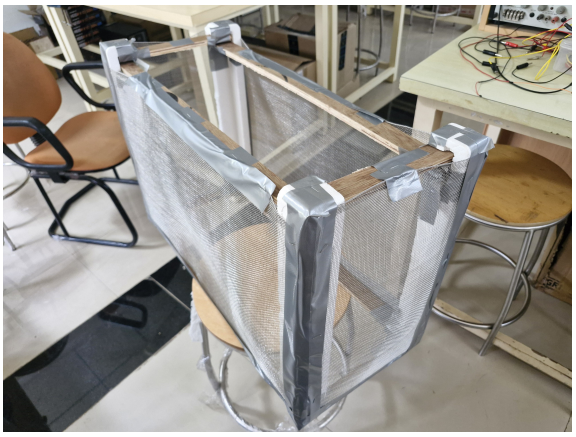


FIGURE 3.10: Safety Cage

Chapter 4

Controller

4.1 Simple PD controller

At first, a simple PD controller is implemented. Due to the nature of the system, the friction produced a point O cannot actuate the angle θ. Hence, we do not expect to see a steady state error and an PID controller was not used.

Suppose we want to tract a trajectory $\theta_t(t)$. We define error dynamics to converge exponentially.

$$\ddot{e} + K_d\dot{e} + K_p e = 0 \tag{4.1}$$

for $K_p, K_d > 0$. Here, the error is defined as $e = \theta - \theta_t$.

To converge the error according to this dynamics, the required acceleration is given by,

$$\ddot{\theta}_{des} = \ddot{\theta}_t - (K_p e + K_d\dot{e}) \tag{4.2}$$

Let,

$$u_d := 2I_0\ddot{\theta}_{des}/l \qquad u_m := \kappa Mg \qquad \alpha_1 := 2h/\kappa l \qquad u_0 := u_d - \alpha_1 u_m \sin\theta \tag{4.3}$$

To produce the desired acceleration from Eq 4.2, we get an equation for thrusts using dynamics from Eq 2.9.

$$u_R - u_L = u_0 \tag{4.4}$$

α_1 is a non-dimensional measure of severity according to which gravity affects balancing. $\alpha_1 = 0$ implies independence of gravity or negligible torque due to gravity. This can occur when $h << l$. Nominally, we get $\alpha_1 = 0.739$.

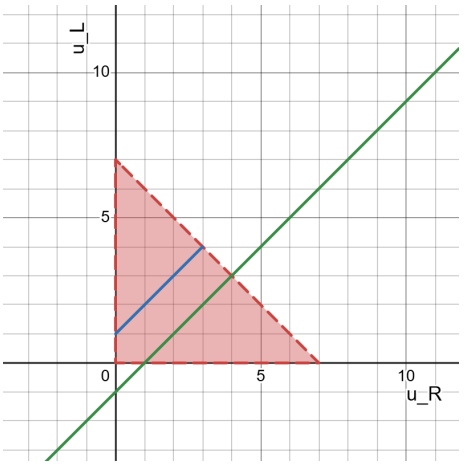
4.2 Ground Contact and Propellor Constraints

Upon applying constraint from Eq 2.10 (ensuring normal contact from ground $N > 0$) and 2.12 (propellers able to produce positive thrust), we get,

$$0 < u_R < \frac{u_m \sec\theta + u_0}{2} \tag{4.5}$$

$$0 < u_L < \frac{u_m \sec\theta - u_0}{2} \tag{4.6}$$

along with Eq 4.4. Note that now only getting one of u_R and u_L is enough.



Green Line is the solution from the simple PD controller in Eq 4.4 (For $u_0 = 1$).

Blue Line is the solution upon applying valid constraints to the PD controller output (For $u_0 = -1$)

FIGURE 4.1: Solution sets using PD controller ($\kappa = 0.7, \theta = 0$)

4.2.1 Existence of a Solution

For a solution to exist, the intersection of the PD controller line solution should intersect the thrust-weight limit line in the first quadrant.

$$-u_m < u_0 \cos\theta < u_m \tag{4.7}$$

Upon solving,

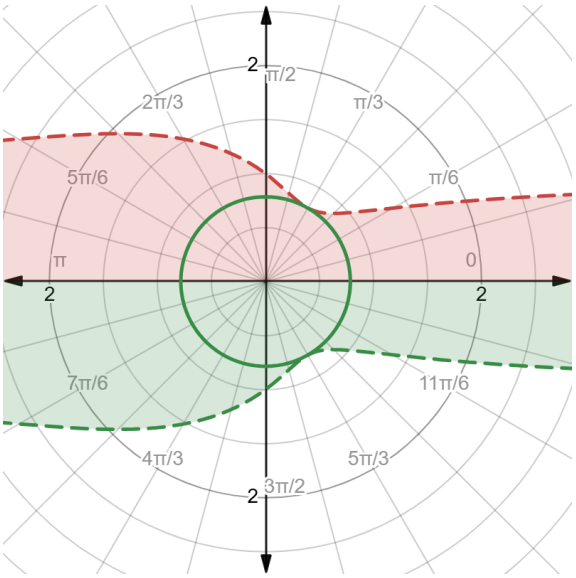
$$\alpha_1 \sin\theta - \sec\theta < (u_d/u_m) < \alpha_1 \sin\theta + \sec\theta \tag{4.8}$$

Since u_m and α_1 are constants, the above equation creates a space in which u_d (and by extension $\ddot{\theta}_{des}$) should lie. For different instantaneous orientations (θ) of the bot, there are different ranges. Note that u_d can be negative.

The reason for getting a condition for existence in terms of u_d is because u_d purely depends on the tunable parameters of the controller and the trajectory. The set of controller gains and trajectory accelerations that guarantee the existence of a solution is dependent on the orientation of the robot. However, since the inequality in Eq 4.8 couples trajectory accelerations, errors, and controller gains, for any set of gains, there can always be trajectories that produce a control requirement outside the constraints. So, we filter the calculated command itself and choose gains and accelerations accordingly. To do this, a limit of the command (u_d) is required.

The figure 4.2 shows a domain of command values. The figure can be understood in this way. Let us say $\theta = \pi/3$, the balancer is inclined towards left at $\pi/3$ from the vertical. In the figure, this is represented by the $5\pi/6$ line for the positive limit and the $11\pi/6$ line for the negative limit. It is clear that we the bot is inclined towards left, there is more space to choose a positive command and when it is inclined towards left, there is more space to choose a negative command.

From Fig. 4.2, it is clear that the limits of the existence of a solution vary with orientation. This is not ideal. So, we will first restrict the magnitude of u_d so that the solution always exists. This can be done by limiting the control space to the largest circle with center



The x-axis can be thought of as the ground with the upright position represented by $\pi/2$ in the above graph. The radial distance from the center is a measure of u_d/u_m for a given orientation, θ. The red region is the upper positive boundary, and the green region is the negative lower boundary. For example, when the bot is upright, then $-1 < u_d/u_m < 1$. At $\theta = \pm\pi/2$, $-\infty < u_d/u_m < \infty$.

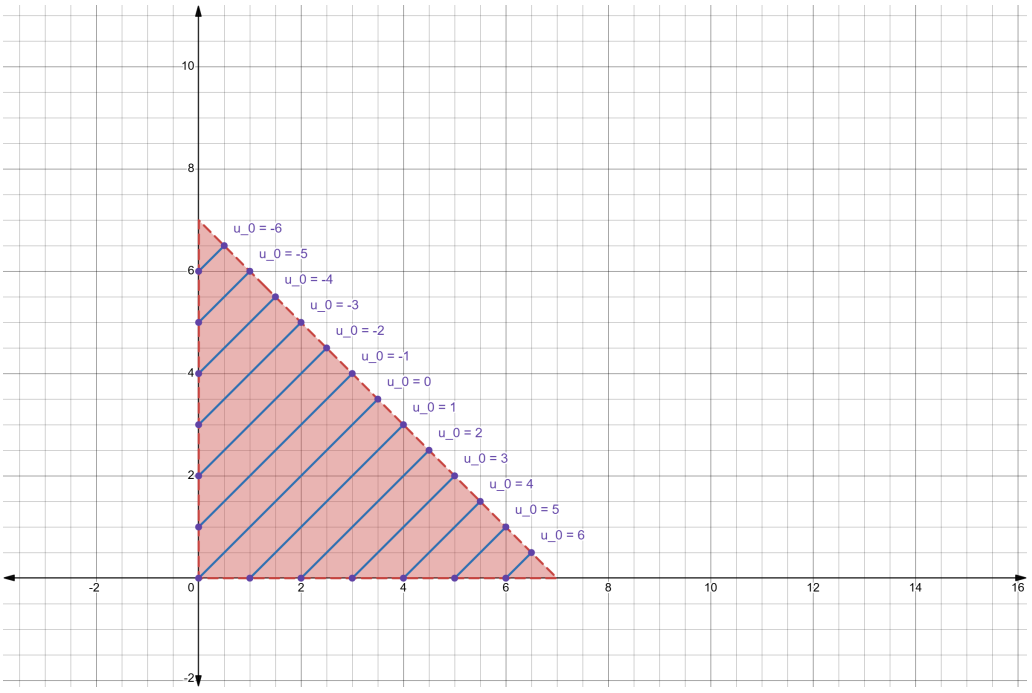
FIGURE 4.2: Command Space ($\alpha_1 = 0.739$)

origin that can fit in the command space. The radius of this circle cannot be calculated analytically. A representation is shown as a green circle in Fig 4.2. The radius of this circle is 0.785. The radius of this circle can be calculated as.

$$R = min(\alpha_1 \sin\theta - \sec\theta) \tag{4.9}$$

As long as the calculated value of $|u_d| < R$, a feasible solution for thrust exists. When the equalities are satisfied, the speed of one of motors has to be 0 (No Thrust). Since calculating R is not analytically possible, we will instead calculate the limit on u_0 and ensure that our solution always exists. This is straightforward as per Eq. 4.7.

For various values of u_0, various lines of solution exist. Refer to Figure 4.3. Each time the controller runs, u_0 should be modified such that the solution never reaches the corners of the feasible triangle range. Furthermore, the point solution on the line never reaches any of the sides of the triangle. This will be done in the following sections.



The x-axis is u_R and the y-axis is u_L. Blue lines are various solution of Eq 4.7

FIGURE 4.3: Distribution of solution of simple PD controller ($\kappa = 0.7, \theta = 0$)

4.3 Saturation and Limiting Condition

To guarantee the solutions never reach the point of 0 thrust, we will have to saturate the value of u_0. Since, $|u_0 \cos\theta| < u_m$, for large values of u_0, we want it to reach $u_m \sec\theta$. At the same time, we want to maintain a large portion of the control region as linear. To do this, α_2 fraction of the magnitude of u_0 is kept linear and the magnitudes above that are saturated; this saturation is performed exponentially. The mapping also ensures $C0$ and $C1$ continuity for smooth trajectories. Let $r = |u_0 \cos\theta / u_m|$ and r' denote the adjusted value for r. Let

$$r' = \begin{cases} r & 0 \leq r \leq \alpha_2 \\ 1 - (1 - \alpha_2) e^{\frac{(\alpha_2 - r)}{1 - \alpha_2}} & r > \alpha_2 \end{cases} \tag{4.10}$$

Using this saturation, we can ensure that lines of solution never reachs the corners of the feasible solution region.

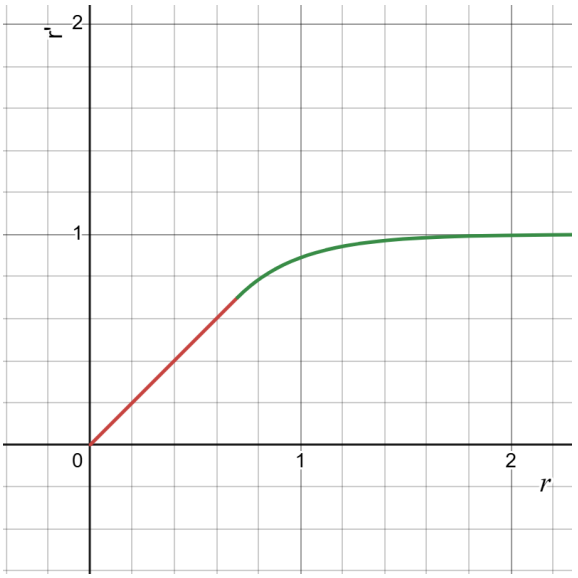
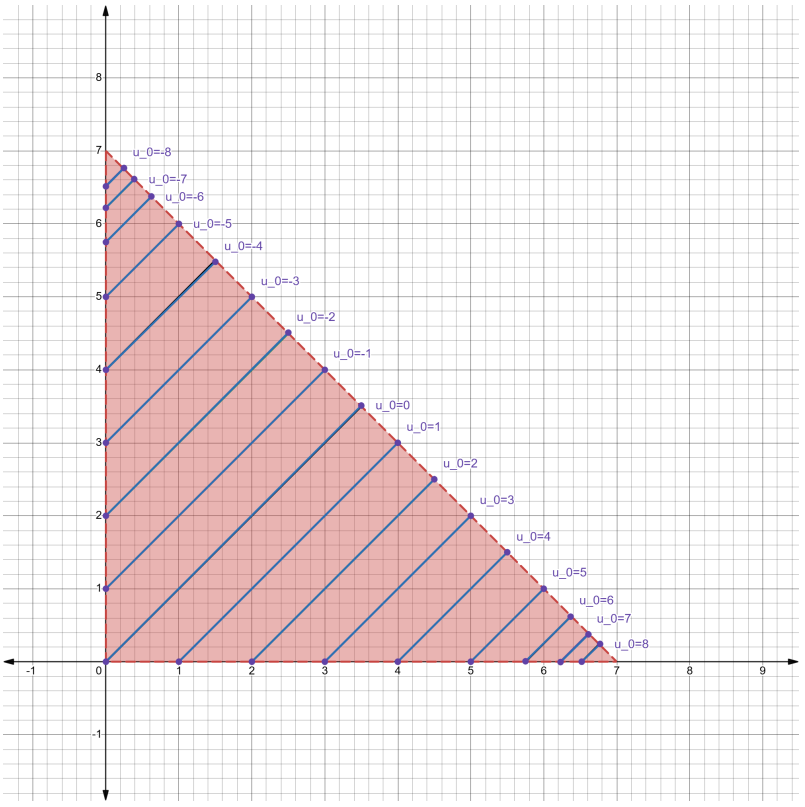


FIGURE 4.4: Saturation function ($\alpha_2 = 0.7$)



The x-axis is u_R and the y-axis is u_L. Blue lines are various solution of Eq 4.7 with saturation from Eq 4.10. Note how u_0 can now be greater than u_m.

FIGURE 4.5: Distribution of solution with saturation ($\kappa = 0.7, \theta = 0, \alpha_2 = 0.7$)

Now that we are sure that we can always find a feasible solution, we have to choose a solution from the line of feasible solutions.

4.4 Optimal Selection of Thrusts

For any given interaction of the controller, the controller computes u_0 from Eq 4.3, saturates it to $u_m \sec\theta$ by Eq 4.10. Now we have to choose one of the solutions of the equation $u_R - u_L = u_0'$. The most greedy approach is the minimize the power consumption by the motors. Let k_T and k_F be the torque and force constants of the propellers. Then if the propellers are rotating at a speed ω, the thrust u and torque τ produced by it are,

$$u = k_F\omega^2 \qquad \tau = k_T\omega^2 \tag{4.11}$$

Power consumed by this motor will be $P = \tau\omega$

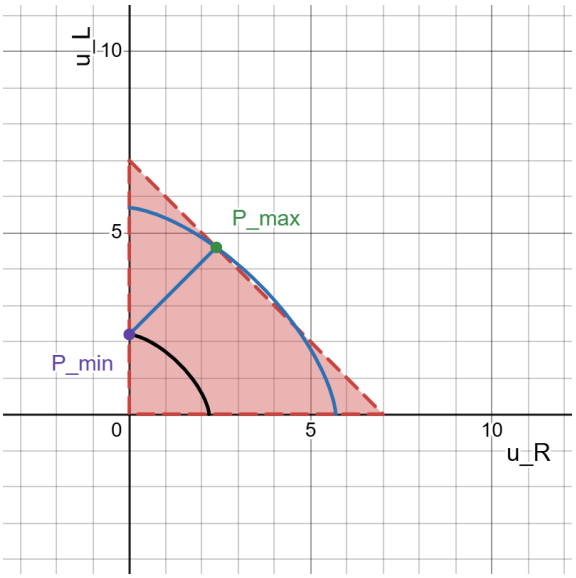
Power consumed by the motors of the bot for a given command pair (u_R, u_L) is,

$$P = \frac{k_T}{k_F^{3/2}}(u_R^{3/2} + u_L^{3/2}) \tag{4.12}$$

Since $u_R - u_L = u_0'$, the most optimum solution would be when one of the command inputs are zero. The maximum power would occur at the intersection point of thrust-weight constraint.

$$P_{min} = \frac{k_T}{k_F^{3/2}}u_0'^{3/2} \qquad P_{max} = \frac{k_T}{k_F^{3/2}}((\frac{u_m + u_0'}{2})^{3/2} + (\frac{u_m - u_0'}{2})^{3/2}) \tag{4.13}$$

Since both conditions are undesirable, we have to ensure that our solution is somewhere in between. At the same time, the amount of power that our solution consumes such that there is a certain amount of power available for it to increase or decrease in the next iteration. Let ϵ fraction of maximum minus minimum power be the minimum amount of power that the bot must consume to ensure the fans don't stop. Let β be the fraction of the maximum - minimum power that is available on either side of our operating point.



The x-axis is u_R and the y-axis is u_L. Blue lines are various solution of Eq 4.7 with saturation from Eq 4.10. Note how u_0 can now be greater than u_m.

FIGURE 4.6: Range of available power ($\kappa = 0.7, \theta = 0$)

Then, we require,

$$\epsilon + 2\beta < 1 \tag{4.14}$$

Then, our bot always consumes at least ϵP_{max} power and ideally operates at $(\epsilon + \beta)P_{max}$ power. To solve this condition and obtain values for (u_R, u_L) would be computationally intensive due to the non-linearity of Eq 4.12. Therefore, instead of lineraly optimizing in power domain, we will linearly choose the operating point in the thrust domains with the extremum corresponding to maximum and minimum power. We know that the solutions to those are,

$$\mathbf{u}_{min} = (u_{Rmin}, u_{Lmin}) \qquad \mathbf{u}_{max} = (u_{Rmax}, u_{Lmax}) \tag{4.15}$$

$$u_{Rmin} = max(u_m, 0) \qquad u_{Lmin} = max(-u_m, 0) \tag{4.16}$$

$$u_{Rmax} = \frac{u_m + u_0'}{2} \qquad u_{Lmax} = \frac{u_m - u_0'}{2} \tag{4.17}$$

Now our operating point becomes,

$$\mathbf{u} = \mathbf{u}_{min} + (\epsilon + \beta)(\mathbf{u}_{max} - \mathbf{u}_{min}) \tag{4.18}$$

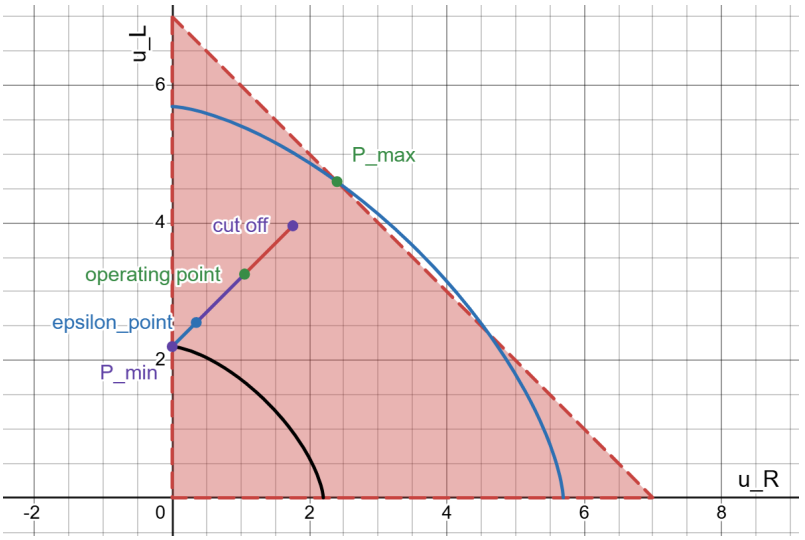


FIGURE 4.7: Optimization solution ($\kappa = 0.7, \theta = 0, \epsilon = 0.15, \beta = 0.3$)

4.5 Implementation

As an initial implementation, a simple PID controller was implemented with speed saturation. One of the motors was always Off, which is decided by the sign of the PID controller's output. The magnitude decides the speed of the motor.

$$PID = -(K_p * e + K_i \int edt + K_d \dot{e}) \tag{4.19}$$

$$\omega_L = max(0, -PID) \tag{4.20}$$

$$\omega_R = max(0, PID) \tag{4.21}$$

Where ω_L and ω_R are speeds of the left and right propellers respectively. The implementation caps these values at 100% and any higher value is automatically interpreted by the ESC as 100%.

Detailed implementation and code can be found at `github.com/kushal-a/propeller_balancer`

Chapter 5

Results

5.1 Simulation

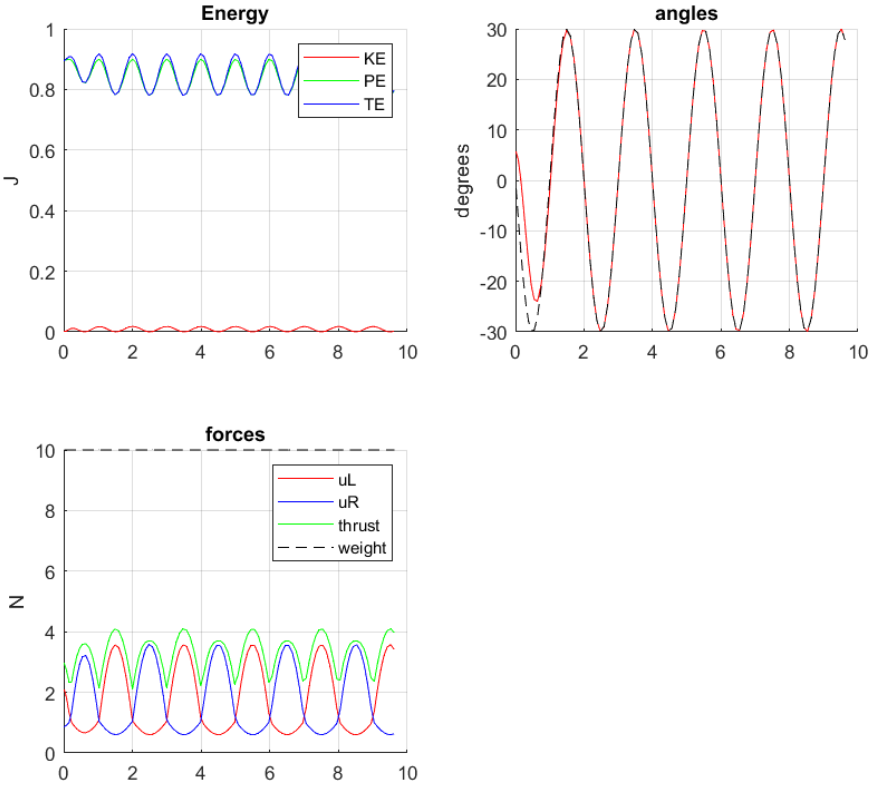


FIGURE 5.1: Controller performance in Simulation

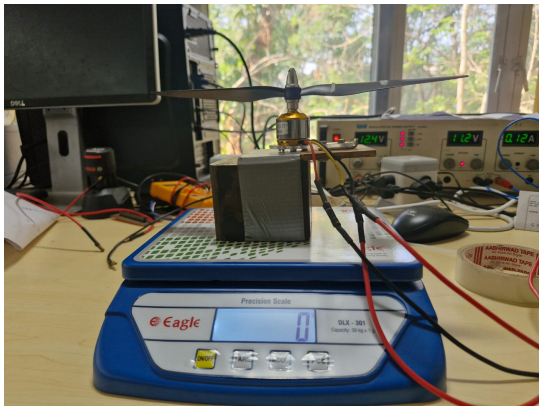
The balancer was asked to follow a sinusoidal trajectory of amplitude $\pi/6$ and frequency $0.5Hz$. The trajectory starts from $\theta = 0$. The initial conditions of the system were 0.1 radians and zero velocity.

In the first figure, we see the energy of the system becoming periodic within one time period. The amplitude of potential energy oscillations is also higher than that of kinetic energy. In the second figure, we see that the balancer is able to follow the trajectory almost identically, after one-fourth of the first oscillation. In the third figure we see how the thrust values are changing. This reveals important information. First, we see that the net upward force (green line) by the propellers is 40% of the weight of the bot. Hence, the normal contact is always greater than 60% of the weight of the bot. Next we see that the individual thrusts by the motors are exactly out of phase and have the same frequency and amplitude. In one time period, a significant portion is sinusoidal. This is the linear part in the saturation equation. The part with the lower amplitude is due to exponential saturation. We also see that thrusts from both the motors are always above certain value, ensuring non-zero speed.

5.2 Hardware

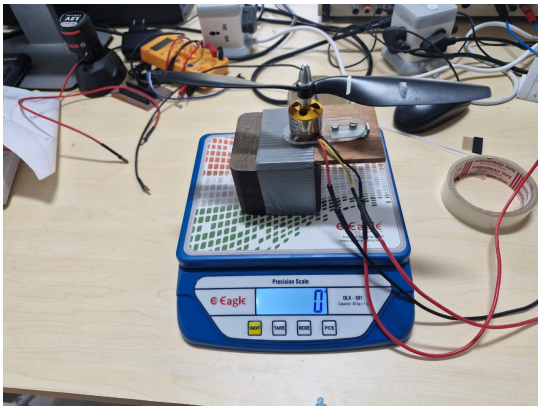
5.2.1 Static Thrust Test

To get an approximate understanding of the kind of thrusts the propellers are able to generate in this setup, a static thrust test was run. 1045 propellers were coupled with the motor and an additional weight of 2 KGs was attached the motor. This assembly was placed on a weighing scale and the scale was trimmed to zero. As thrust was generated, the scale should the negative weight it was measuring, which is equal to the thrust produced in grams.



((A)) Front View

((B)) Top View



((C)) Isometric View

FIGURE 5.2: Setup of static thrust test

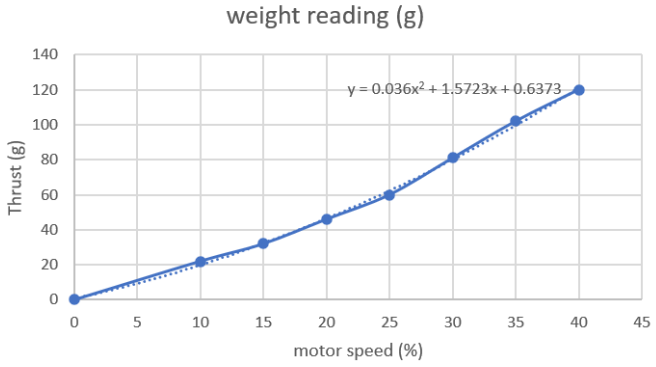


FIGURE 5.3: Thrust vs Motor Speed

The thrust characteristics seen have a significant contribution from both the square and linear part. With the polynomial approximation, it was expected that a 100% speed would

produce 500g of thrust. Using two such motor could potentially bring the net thrust of the system to be very close to weight of the setup. Hence, to ensure safety and get enough low speed operating range, a smaller propeller was used.

1045 propeller has a length of 10 inches. We require thrust to be about half of what we observed with the 1045 propeller, we wanted the new propeller length to be about 0.7 (THRUST α L^2) times the 1045 propeller length. This would result in using a 7 inch propeller. But due to the linear contribution of the system, an 8 inch propeller was used instead. Hence, we used an 8045 propeller for running PID.

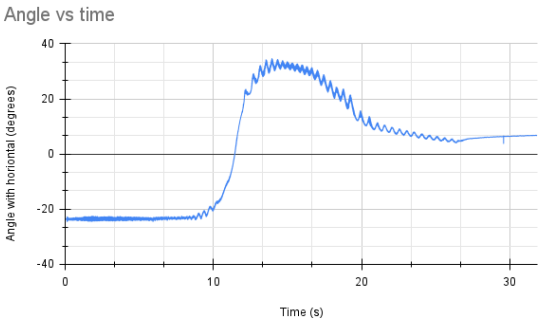
5.2.2 PID Test

The system was released from its left extreme and the control law was run until steady-state was observed, i.e., until about 30 seconds. Full video of the experiment is available at `youtu.be/ESJKh3KnrgE`.

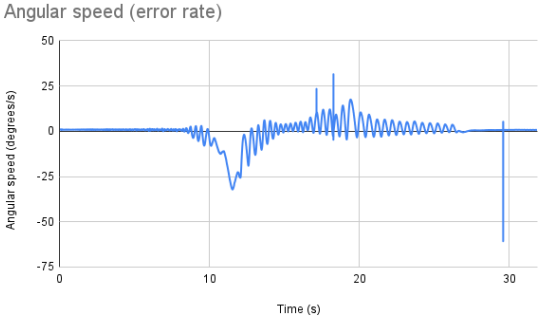
Due to the absence of gravity feedforward, the use of Integral gain in PID was essential. Additionally, we observe that the integral contribution is much smoother compared to the Proportional and Derivative contributions. The system settled with minor osscilations at less that 5°at about 27s. Post this, the power supply was cut and the resulting fall is captured in the 27s to 30s range.

There is clear requirement of using a low pass filter for the angle and angular rate to get a smoother response.

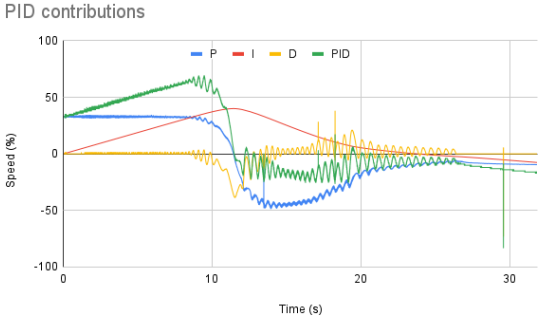
Another important observation is that it took almost 72% of motor velocity to overcome the initial gravity force. Due to this, not a large amount of power availability could be ensured and hence, the system was operated in the minimum power mode where one of the propellers is always at rest.



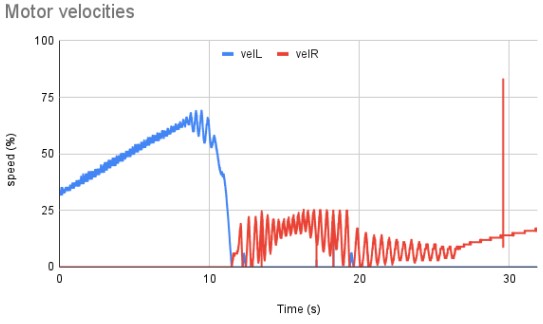
((A)) Angle with vertical



((B)) Angular rate of balancer



((C)) PID contributions



((D)) Motor Speeds

FIGURE 5.4: Hardware performance
$K_p = 1.4$ $K_i = 80$ $K_d = 1.2$

Chapter 6

Conclusion

6.1 Summary

Attitude control refers to orientation control of a body in space and has several applications in civil and defense domains in drones, aircraft and spacecraft. To understand the intricacies of attitude control, we have developed a robotic platform able to rotate in its trivial plane with propellers used to generate torque along that axis. Here we restrict the platform from taking off. A controller is designed which is aware of the physical limitations of the physical model. A PD controller with gravity feedforward is used to compute desired torques. To add constraints, a depth study is performed, restricting the domain of valid inputs and further ensuring that the solution lies well inside the domain. Power optimization is performed to get a good estimate of control values which prevents non-linear behaviors of the model as well as prevents necessarily high amounts of thrusts. We are able to see the controller perform well in setpoint and trajectory tracking, as well as the saturation and minimum power requirements being satisfied.

For hardware, a novice controller was used as a proof of concept for the system. PID was implemented without any feedforward. The best case PID tuned results were discussed. It is important to have a low pass filter for sensor data or use another sensor with a Kalman filter to get noise free data. This will significantly reduce the noise in the system.

6.2 Future work

In the future, we want to increase the size of the propeller to a 9 inch one and then varying the pitch to get the thrust range we require. Further, the control law discussed in simulation can be implemented for smoother results.