INDIAN INSTITUTE OF TECHNOLOGY, MADRAS 2023

* Arin Gangal; Roll no - ME21B029. *E-mail address:* ME21B029@smail.iitm.ac.in
* Abesech Inbasekar; Roll no - ME21B003. *E-mail address:* ME21B003@smail.iitm.ac.in
* Aravind S; Roll no - ME21B027. *E-mail address:* ME21B027@smail.iitm.ac.in
* Edla Laxman; Roll no - ME21B056. *E-mail address:* ME21B056@smail.iitm.ac.in

Abstract

A Boeing CH-47 Chinook is a tandem-rotor heavy-lift helicopter. It has two rotors in the front and back which allow it to be one of the heaviest lifting helicopters. Tandem Rotor helicopters like Chinook have a wide range of center of gravity, making them ideal as vertical takeoff cargo carriers. And since the rotors rotate in opposite directions to cancel out the counter-rotation, no tail rotor is necessary to stabilize the yaw. The challenge, however, lies in stabilizing the pitch. The two tandem rotors need to balance the lift dynamically to achieve the desired pitch. In this report, we build a control algorithm using arduino to control the pitch based on external loading conditions and demonstrate it using a simplified physical setup..

Keywords: Pitch, Arduino, BLDC, Propellers

The Boeing CH 47 Chinook is a twin-engine, tandem-rotor heavy-lift helicopter used by the USA and other militaries around the world. It is capable of carrying 12000 pounds of cargo. This report aims at building a control algorithm to stabilize the pitch and demonstrate it using a simplified model, under external loading conditions.

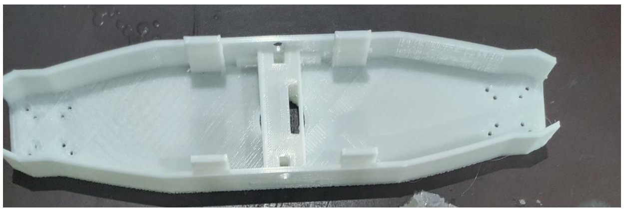
Nomenclature

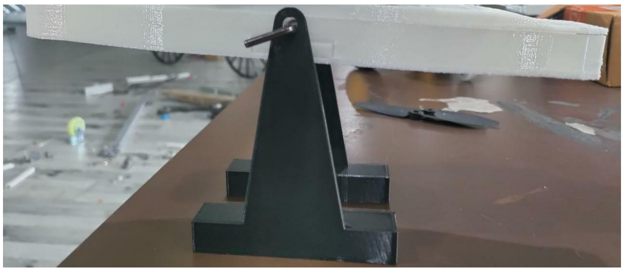
Pitch	The angle of tilt with respect to the origin.
Kp	Proportional gain
Kd	Derivative gain
Ki	Integral gain

1.1. Component List

We divide the model into two parts: The Chassis and the Electronics.
The Chassis:

- 3D Printed, made of PLA (PolyLactic Acid)
- Contains two components, The main chassis with the motors and the Base about which the Chassis is pivoted.
- A compact model, with apt spacing between the Base and the chassis.





Dimensions:

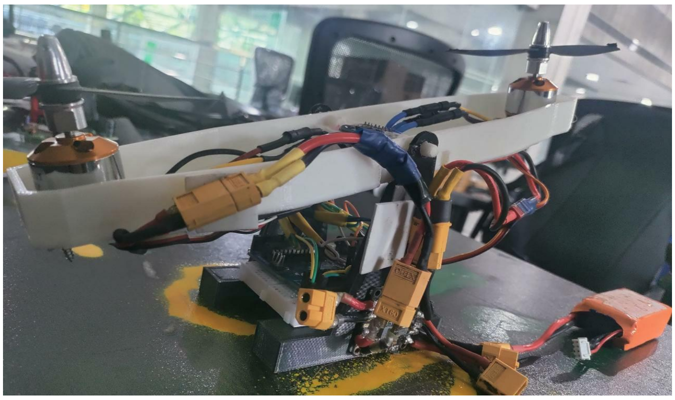
Length of the Chassis - 290 mm
Width - 65mm
Elevation from the base - 115 mm
Weight of the Chassis - 350gm
Weight of the Base - 150gm

Considering the given problem statement the chassis was required to achieve a pitch range of -30 degrees to +30 degrees. The Design is compact with ample space for the electronics. There is a small step in the center to place the IMU (MPU6050) sensor with holes in the sides for the wires to go underneath. The chassis has two extensions as pivots which make a revolute joint with the base.

Electronics:

- Two 1400KV BLDC motors
- Two 30A ESCs
- MPU6050 Sensor
- Arduino UNO
- Breadboard
- Jumper wires and Connectors
- Power Distributor
- 3 cell, Li-Po Battery

Considering the thrust requirements for controlling the Pitch, a 1400KV brushless DC motor was chosen. It was controlled by a 30 Ampere ESC, with a power distributor. To sense the movement, an MPU6050 sensor was chosen as it is one of the most commonly used IMU in the market with great data and applications on the web hence making it the most convenient choice. The algorithm was coded in Arduino IDE and was implemented through UNO. All of the connections were made using Jumper wires.



1.2. Mechanical Design of the System:

1.2.1 Free Body Diagram

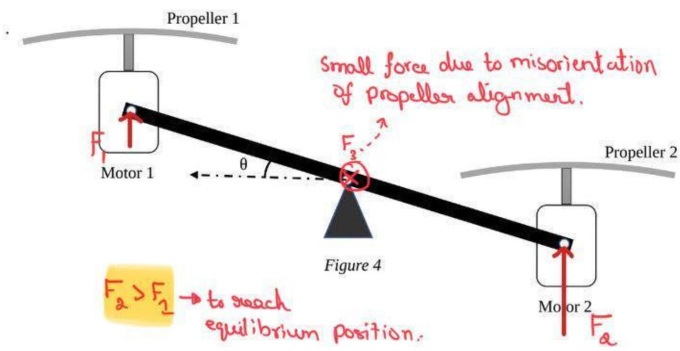
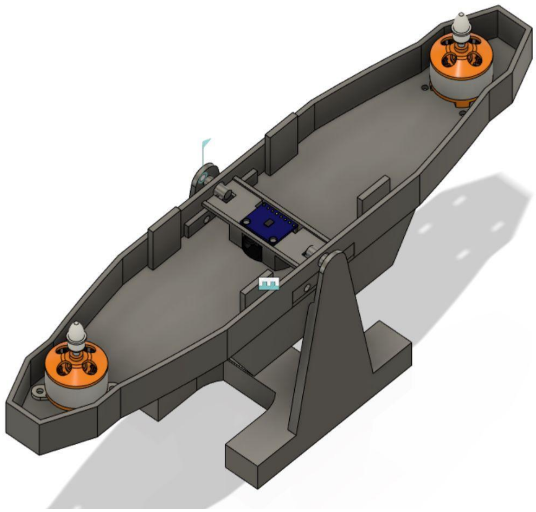
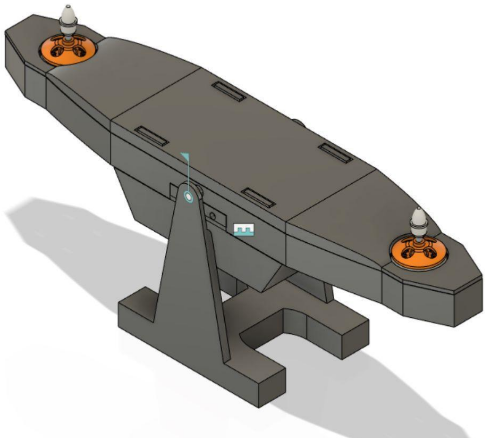
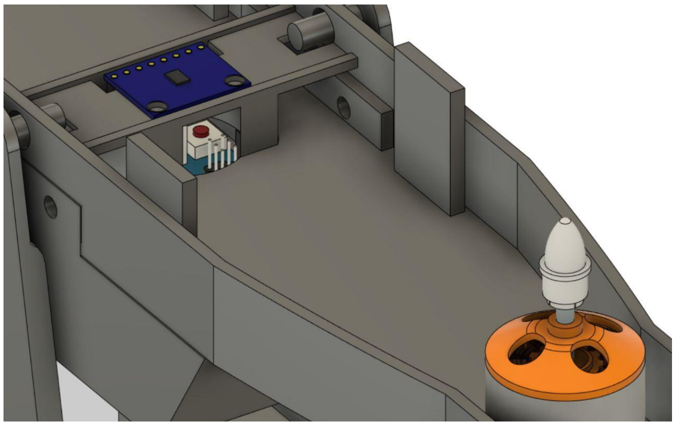


Figure 4

1.2.2. CAD model

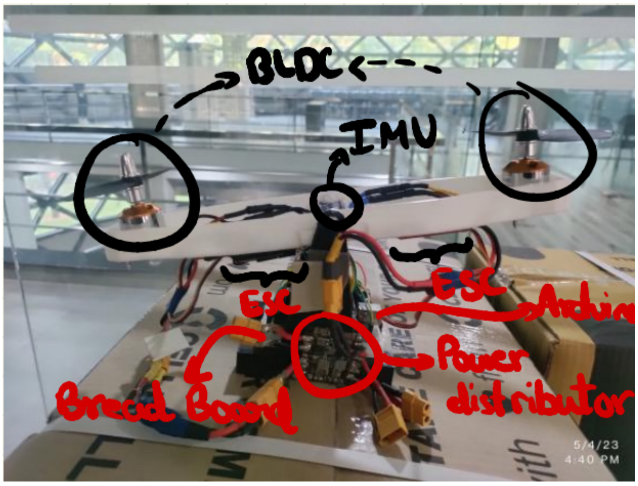




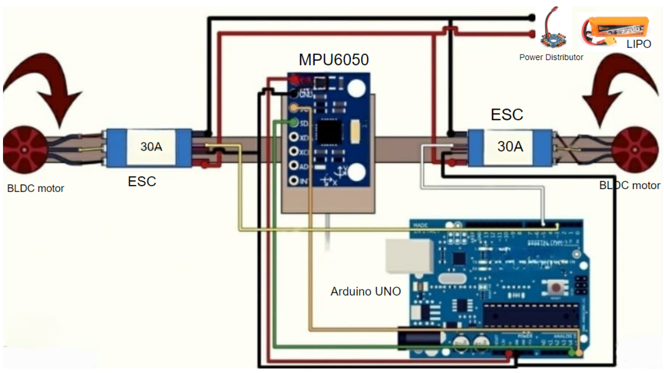


1.2.3. Placement of Components (From bottom to top)

- The breadboard is pasted onto the bottom of the base.
- The Arduino is attached to the breadboard using double-sided tape.
- The IMU is pasted onto the chassis using a double-sided tape.
- The two BLDC motors were screwed into either side of the chassis using M3 screws..
- The ESC's were pasted at the bottom of the chassis (using double-sided tapes) and are wired to the motors through the holes on the sides of the IMU.
- The power distributed is pasted to the side of the base and is connected to the ESC's and the battery.



The electronics are arranged as shown above in the Chassis.
1.3. Electronic circuit design
The circuit was designed mainly with Arduino UNO, a breadboard, and a set of Jumper wires and XT60 connectors. IMU6050 was connected.

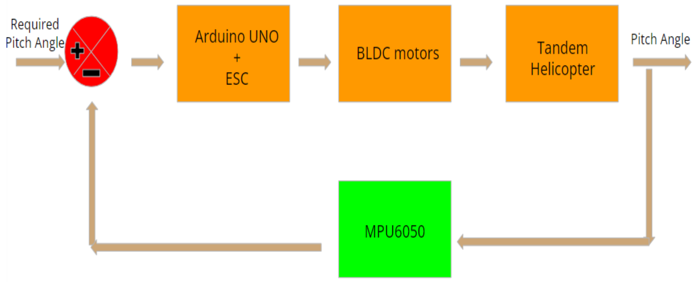


1.3.1. Controller design and Arduino program

Controller Design:

A PID controller is used to control the chassis orientation. Acceleration and gyroscopic values are read from the IMU. To remove noise, an average of three consecutive readings at an interval of 5ms is taken and used as the reading. This method, though being very simple, was found effective for the application.
Oscillations at zero pitch were reduced further using a High Pass filter which filtered all the noise for angular velocity less than 5 degrees/sec and pitch less than 3 degrees. These values were chosen based on hit and trial and were found to be effective.
Tuning of the controller was then done on hit and trial method first for a PD controller, setting first some values for the Kp and Kd. Keeping Kp constant, Kd was modified to get better results. For the achieved value of Kd, Kp was modified to further reduce oscillations.
The controller was then modified by adding Integral correction as well. For the set Kp and Kd, a very low value of Ki was chosen to observe the system response. Accordingly, Ki was modified to get better results.

1.4. Control System block diagram and transfer function model



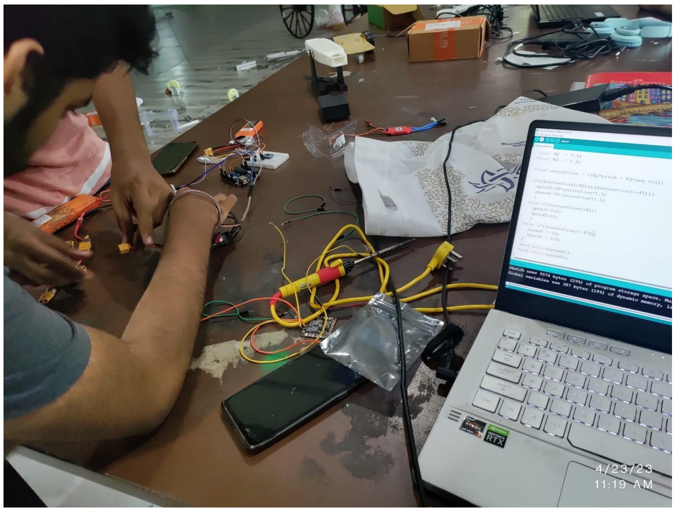
Let's denote the angular velocity of the front rotor as ω_1 and the angular velocity of the rear rotor as ω_2. The output variable, representing the system's balancing angle(pitch), can be denoted as θ.
Equations of Motion: The equations of motion for the tandem rotor can be expressed as:
$I_1 * d^2\theta/dt^2 = -c_1 * \omega_1 + c_2 * \omega_2$
$I_2 * d^2\theta/dt^2 = c_1 * \omega_1 - c_2 * \omega_2$
Here, I_1 and I_2 represent the moments of inertia of the front and rear rotors, respectively. c_1 and c_2 are the coupling coefficients that determine the interaction between the rotors.
Assuming small deviations from the operating point, we can linearize the equations by neglecting the cross-coupling terms:
$I_1 * d^2\theta/dt^2 \approx -c_1 * \omega_1$
$I_2 * d^2\theta/dt^2 \approx c_1 * \omega_1$
Applying the Laplace transform to the linearized equations, we have:
$s^2 * (I_1 + I_2) * \Theta(s) = -c_1 * \omega_1(s)$
Here, $\Theta(s)$ represents the Laplace transform of the balancing angle $\theta(s)$, and $\omega_1(s)$ is the Laplace transform of $\omega_1(t)$.
Rearranging the equation, we can express the transfer function G(s) as:
$G(s) = \Theta(s)/\omega_1(s) = -c_1 / (s^2 * (I_1 + I_2))$

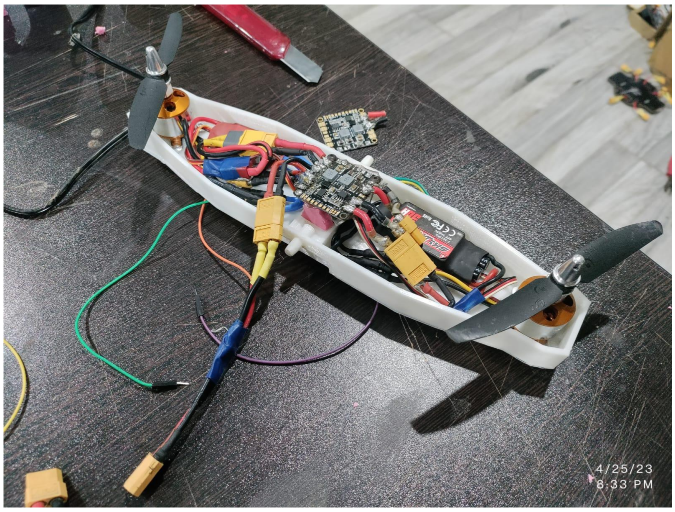
The transfer function relates the output (balancing angle) $\Theta(s)$ to the input (angular velocity of the front rotor) $\omega_1(s)$ in the frequency domain.

1.6. Conclusions and lessons learned:

Upon applying an external load, we could see the chassis stabilizing itself, taking input from the IMU sensor, and changing motor speed. The oscillations were less and the steady state error, though present, was minimal. The control parameters were working fine and pitch stability was achieved.

This project made us understand and use various concepts of Mechatronics taught in Class. From modeling the chassis in Fusion 360 to 3-D printing it, calculating the Force and thrust parameters, and mounting electronic components based on the requirements, it was an enriching experience. On the Software side, we used ArduinoIDE to code the algorithm and tested it thoroughly by trial and error methods to stabilize the model. The tuning parameters were determined by trial and error. The code was optimized after each test to arrive at better results. There were some accidents throughout the process with some of us getting our fingers cut to damage to our laptops but still, we enjoyed every second of it and laughed about the mistakes we made throughout the process. This project gave us an overall introduction to the world of mechatronics and made us apply the theory in practice.





Equation:

Most of the equations used for modeling were given as a part of Arduino code (with comments). However here are some of the main equations
(1) correction = -(Kp*pitch + Kd*ang_vel+corr_i)
where corr_i = Ki(∑pitch).

Individual Contributions:

Arin- Design of Chassis, Data collection from IMU, Electronic circuit design, and Assembly.

Abesech - Filtering of data collected, Calculation of Transfer Function, Filing of 3D printed components to achieve a snug fit and Assembly.

Aravind- Tuning by hit and trial , Placement of electronic components in a compact way, and Assembly.

Edla Laxman- Assembly.

Cost Analysis:

Item	Cost (in rupee)
Arduino UNO	350
MPU6050	125
BLDC Motors (2)	778
ESC (2)	760
Jumper Wires	58
Breadboard	42
Power Distributor	88
Total	2201

Acknowledgments:

We thank Dr. Manish Anand for giving us the opportunity to put our theoretical knowledge in use through this amazing and innovative project. Not only his class and his material, but this project work was also interesting and demanded our blood and sweat.

References:

[1] William Bolton. Mechatronics: Electronic control systems in mechanical and electrical Engineering, 6th edition.
[2] www.arduino.cc.
[3] Dr. Manish Anand. Lecture notes, 2023
[4] Video and Code for the project: Link