DESIGN LAB PROJECT

TILT ROTOR MECHANISM

SUBMITTED BY:

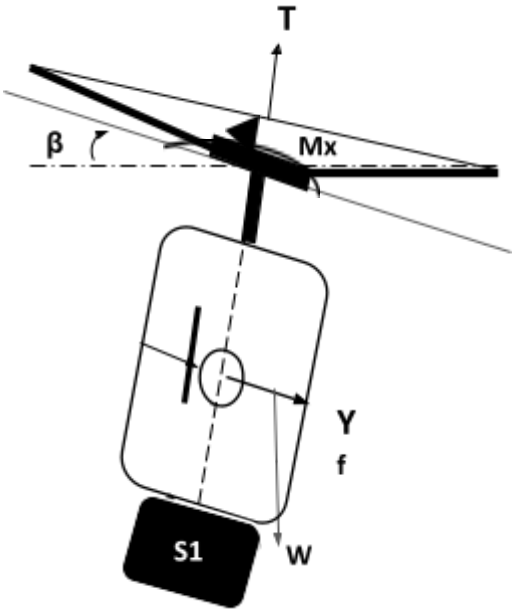
Gobind Puri (2021MEB1283)
Guguloth Krishna Prasad (2021MEB1284)
Kayitha Saran Yadav (2021MEB1290)
Korupolu Deekshitha (2021MEB1292)
Nishika Nakka (2021MEB1301)

1.1 Aim

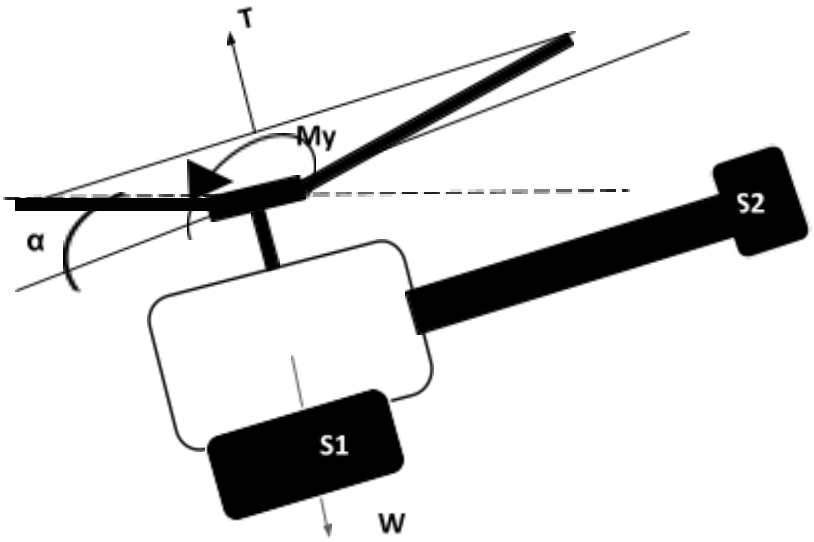
Creating a Tilt Rotor Mechanism and Measurement involving theoretical calculations and design aided by simulations.

1.2 Theory

Movement of the Tilt Rotor Mechanism is defined by the rotation of the axes:
Rotation of X-axis – Roll
Rotation of Y-axis – Pitch



Front View



Side View

- S1 and S2 represent the servo motors. One servo motor is used for Pitch and other servo motor is used for Roll.
- α represents the Roll Angle.
- β represents the Pitch Angle.
- Each servo motor is responsible for the moment in the respective directions.
- The Roll motion occurs in the X-Z Plane.
- The Pitch motion occurs in the X-Y Plane.
- The positioning of each component has been decided in-order to balance the model

1.3 Equipment Specification

The equipments used in our project are:

1. MG995 High Speed Servo Motors – 2
2. ESC -2 Speed Controller
3. MPU 6050 Gyroscopic Sensor
4. Arduino UNO
5. DC 3-phase Motor
6. Propeller blades
7. Nylon Rod
8. Wooden Blocks
9. Jumper Wires
10. Battery

1.4 Calculation

<u>VTOL in Equilibrium:</u>

Considering that W = Weight , T = Thrust , D = Drag, H = Horizontal Force

1. **In Vertical Direction : $W - T_M \cos\alpha_s \cos\beta_s + Dv - H \sin\alpha_s = 0$**
   But since we aim to perform the tilt at a stable horizontal equilibrium, The Horizontal Force component and Drag should be equal to each other.
   So, $Dv - H \sin\alpha_s = 0$
   - ⇨ **Vertical Direction equilibrium equation = $W - T \cos\alpha_s \cos\beta_s = 0$**

2. **In Longitudinal Direction: $Dp - T_M \cos\alpha_s \cos\beta_s - H \cos\alpha_s = 0$**
   But since we aim to perform the tilt at a stable horizontal equilibrium, The Horizontal Force component and Drag should be equal to each other.
   So, **$Dp - H \cos\alpha_s = 0$**
   - ⇨ **$T_M \cos\alpha_s \cos\beta_s = 0$**

3. **In Lateral Direction: $T_T \cos\beta_s + T_M \cos\alpha_s \sin\beta_s = 0$**

4. **Equilibrium in Pitch:**

   $Ms1+My_M - W(x_{cg} \cos\alpha_s - h \sin\alpha_s) - Dp(h \cos\alpha_s + x_{cg} \sin\alpha_s) = 0$

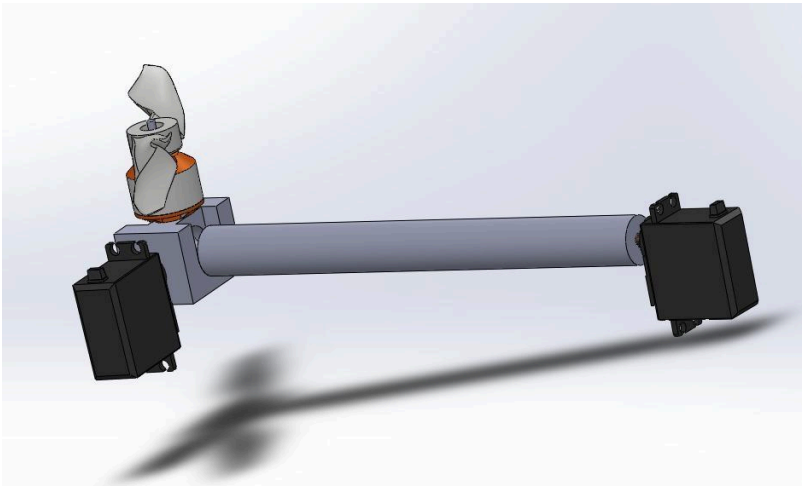
Since the horizontal force is zero, the Drag force which is supposed to balance the Horizontal force is zero. So, Dp = 0

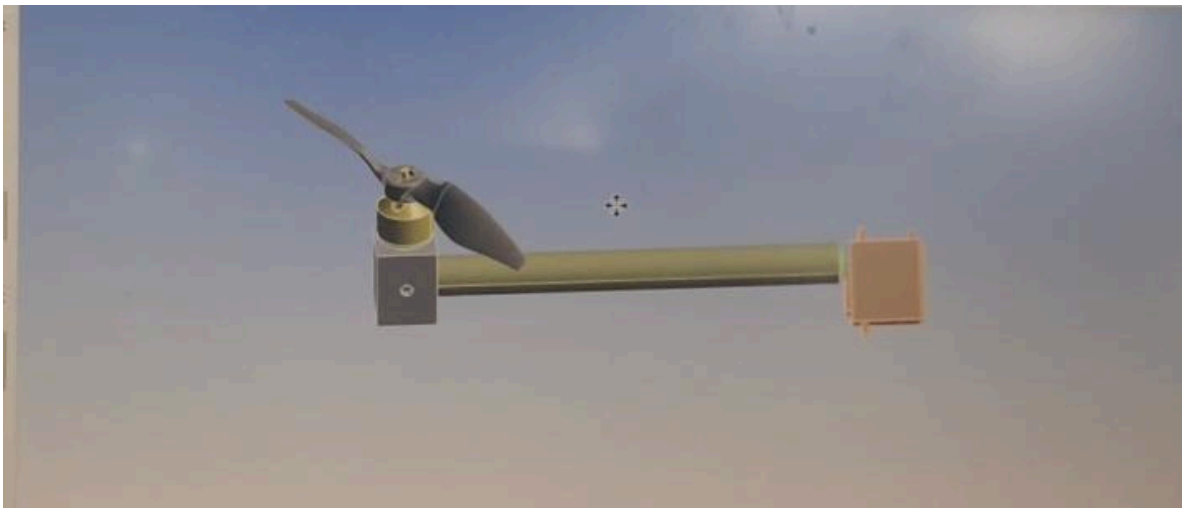
- ⇨ **M_{yM} - W(x_{cg} Cos α_s – h Sin α_s) = 0**

5. **Equilibrium in Roll: $M_{s2}+M_{xm} + T_T h_T + W (h\text{Sin } \beta_s – Y_{cg} \text{Cos } \beta_s) = 0$**
   Where M represents the Moment

To calculate Thrust we will use T = R x F

1.5 Simulation





1.6 Code

```
#include <Servo.h>
const int SERVO_PIN = A0;
const int SUPPLY_VOLTAGE = 12;
int servoPosition = 90;
Servo myServo;
Servo motorController;
void moveServoLeft(int targetPosition) {
  for (servoPosition; servoPosition >= targetPosition; servoPosition--) {
    myServo.write(servoPosition);
    delay(20);
  }
}
void returnServoToCenter() {
  if (servoPosition > 90) {
    for (servoPosition; servoPosition >= 90; servoPosition--) {
      myServo.write(servoPosition);
      delay(20);
    }
  } else if (servoPosition < 90) {
    for (servoPosition; servoPosition <= 90; servoPosition++) {
      myServo.write(servoPosition);
      delay(20);
    }
  }
}
void moveServoRight(int targetPosition) {
  for (servoPosition; servoPosition <= targetPosition; servoPosition++) {
    myServo.write(servoPosition);
    delay(20);
  }
}
void setup() {
  Serial.begin(19200);
  myServo.attach(SERVO_PIN);
}
void loop() {
  if (Serial.available() > 0) {
    String serialInput = Serial.readStringUntil('\n');
    Serial.println(serialInput);
    char command = serialInput[0];
    char valueChar[6];
    serialInput.substring(2).toCharArray(valueChar, sizeof(valueChar));
    long numericValue = String(valueChar).toInt();
    if (command == 'l') {
      if (numericValue < 90)
        moveServoLeft(numericValue);
    } else if (command == 'r') {
      if (numericValue > 90)
        moveServoRight(numericValue);
    } else if (command == 'o') {
      returnServoToCenter();
    }
  }
}
```

1.6 Prototype: