Indian Institute of Technology Gandhinagar

Self-balancing Stick by Reaction Wheel

ME 352 : Mechanical Engineering Lab 2

Lab Report - Experiment 5

Group - 8

28 February 2023 (Week 6)

Authors

Pavidhar Jain - 20110136

Priya Gupta - 20110147

Rahul R Pai - 20110155

Raj Krish Dipakkumar - 20110160

Under The Guidance Of

Prof. Jayaprakash

OBJECTIVE

To balance a stick by reaction wheel.

ABSTRACT

A reaction wheel is a flywheel that is designed to give a reaction torque when torque is applied to it. Spacecraft primarily use reaction wheels for attitude control, which replaces the need for thrusters. Reaction wheels have the benefit of being able to apply torque to a system in the absence of an external structure or medium to do so. Motors used to apply torque to the reaction wheels will create reaction forces that could keep the pendulum standing.

EXPERIMENTAL DESIGN

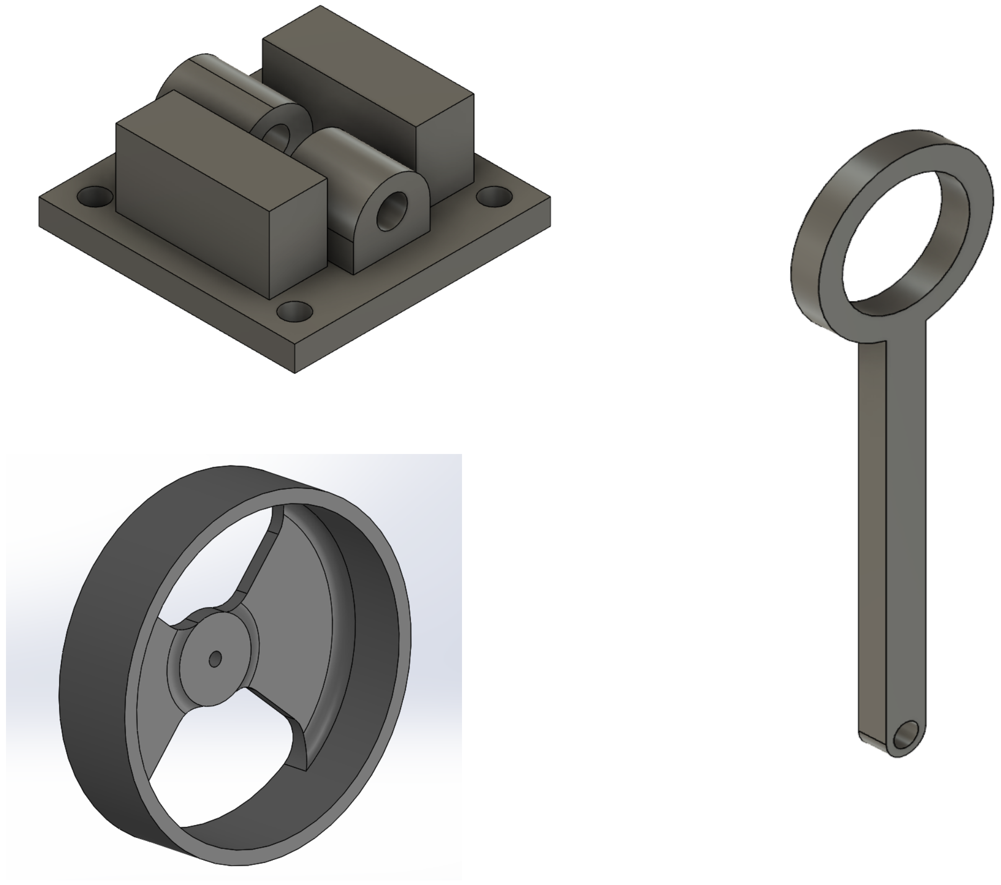

Figure 1:CAD model of setup

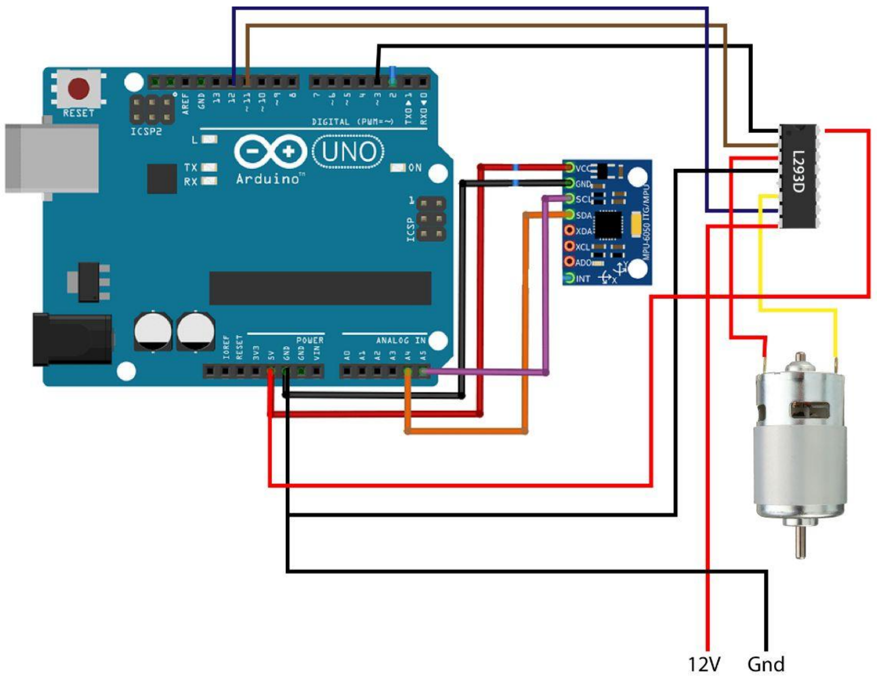

Figure 2 : Schematic of electrical connections

FABRICATION DETAILS

The platform that holds the components, reaction wheel, and stick was built using a 3D printer with certain dimensions. The reaction wheels were designed to have low mass while simultaneously keeping a high moment of inertia.

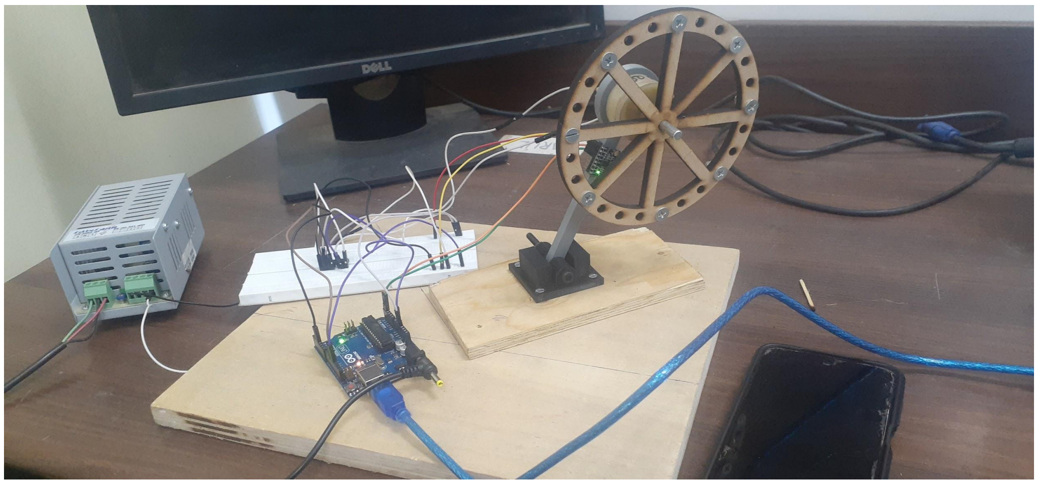

Figure 3 : Trial experimental setup

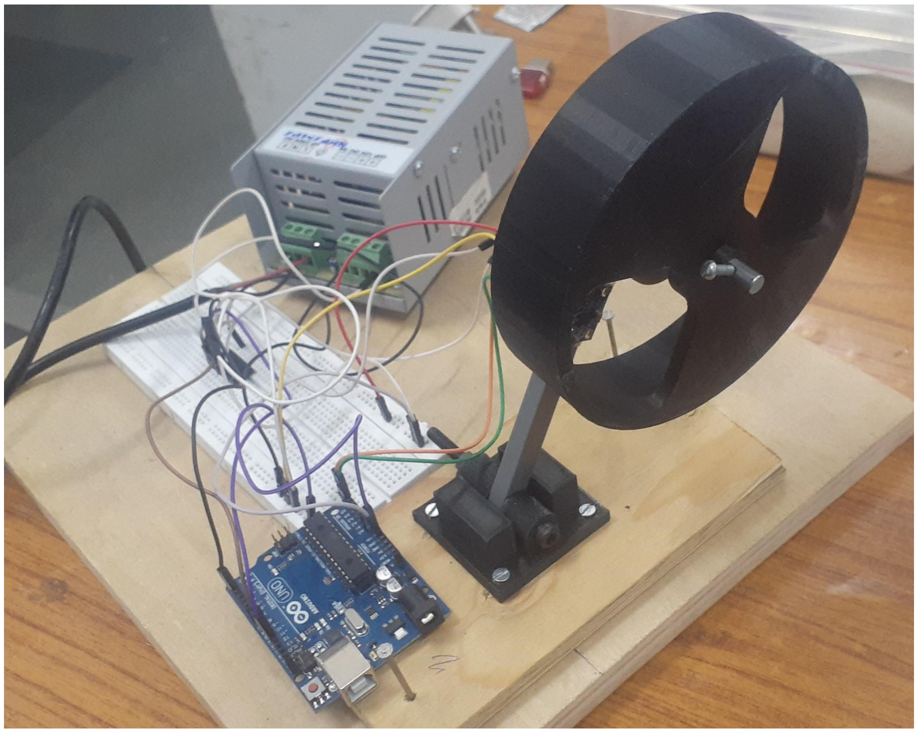

Figure 4 : Final experimental setup

MATHEMATICAL MODELING

Involved equations

$$m_p r_{cog}^2 \omega_p + J_\omega \omega_\omega = m_p r_{cog} g \sin\theta_p$$

PID controller

A Proportional-Integral-Derivative(PID) controller is a feedback control mechanism widely used in industrial, engineering, and robotic applications to regulate a system's behavior. It continuously monitors the system's output, compares it to a desired reference or setpoint value, and generates an error signal that adjusts the system's input. A PID controller can be an effective tool for balancing a stick using a reaction wheel by continuously measuring the angle of the stick and using that information to adjust the stick's position in real-time.

A self-balancing stick using a reaction wheel is a system that uses a motorized wheel to control the orientation of the stick and keep it balanced. PID controller is a feedback control system that is commonly used to control such systems as it takes in the current angle of the stick and calculates the error between the desired angle and the current angle.

The PID controller uses three terms to determine the output to the motor: proportional, integral, and derivative

Proportional term: Computes an output that is proportional to the error signal. The larger the error, the larger the correction signal. It would adjust the stick's position in proportion to the in proportion to the angle of the stick. This term provides a quick initial response to disturbances but may result in overshooting or instability.

$$P_{out} = K_p e(t)$$

$where, \quad K_p = proportional\ constant\ = 0.01$

$and\ e(t)\ = distance\ between\ the\ stick's\ position\ and\ vertical$

Integral term: Integrates the error over time, which can help eliminate steady-state errors by gradually adjusting the stick’s position over time. It reduces the system's error over time but may also introduce overshoots or oscillations.

$$I_{out} = K_i \int_0^t e(t)dt$$

$where, \quad K_i = integral\ constant$

$and, \int_0^t e(t)dt\ = Integration\ of\ e(t)\ over\ time$

Derivative term: Computes the rate of change of the error signal, which can help predict the future trend and dampen oscillations. It would predict the future trend of the stick’s angle and help prevent oscillations. It provides a correction signal proportional to the error rate of change, helping to reduce overshoot and improve stability.

$$D_{out} = K_d \frac{de(t)}{dt}$$

where,

$K_d = derivative\ constant$

$and\ \frac{de(t)}{dt} = derivative\ of\ e(t)\ at\ a\ particular\ time$

The three control terms are combined to generate the final output of the controller, which is used to adjust the system's input. The relative weights of the terms are set by tuning the controller's parameters, which depend on the specific application and system.

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt}$$

An optical rotary encoder was used to measure the stick's angle with vertical and provide feedback to the controller to implement the PID controller in the self-balancing-sticksystem. The controller compared the stick's position to a desired setpoint(vertical) and generated an error signal, which adjusted the stick's angle. The controller's parameters were tuned to optimize the system's performance and stability to keep the stick vertically upright.

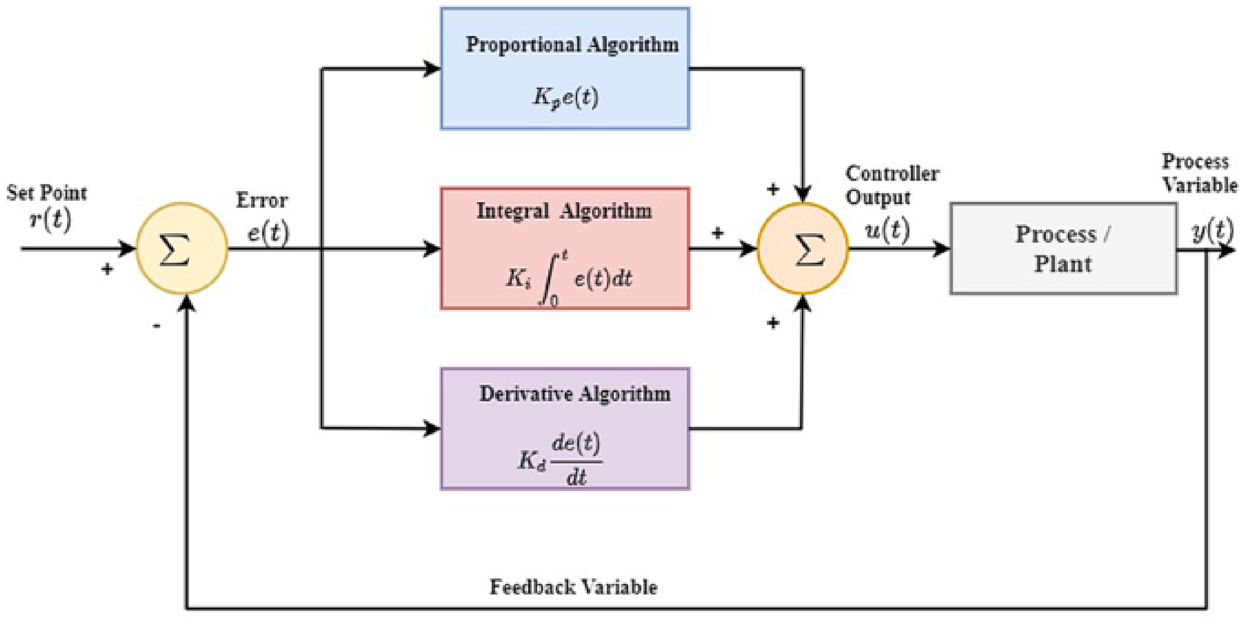

Figure 5 : Schematic of PID controller

CODE

```
#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
#include <Wire.h>

Adafruit_MPU6050 mpu;

//const int PWMA = 5;   // attachment of motor_driver to arduino
//const int AIN1 = 10;
//const int AIN2 = 8;
float rpm;

const int pwm = 3;
const int dir1 = 12;
const int dir2 = 13;

int Read = 0;
float angle = 0.0;
float elapsedTime, time, timePrev;        //Variables for time control
float angle_previous_error, angle_error;
int period = 50;  //Refresh rate period of the loop is 50ms

float kp=50;
float ki=0;
float kd=0;
float angle_setpoint = 0;           //Should be the distance from sensor to the middle of the
bar in mm
float PID_p, PID_i, PID_d, PID_total;
float duration;
```

```
void setup(void) {
  Serial.begin(115200);

  // Try to initialize!
  if (!mpu.begin()) {
    Serial.println("Failed to find MPU6050 chip");
    while (1) {
      delay(10);
    }
  }
  Serial.println("MPU6050 Found!");

  pinMode(dir1, OUTPUT);
  pinMode(dir2, OUTPUT);
  pinMode(pwm, OUTPUT);

  // set accelerometer range to +-8G
  mpu.setAccelerometerRange(MPU6050_RANGE_8_G);

  // set gyro range to +- 500 deg/s
  mpu.setGyroRange(MPU6050_RANGE_500_DEG);

  // set filter bandwidth to 21 Hz
  mpu.setFilterBandwidth(MPU6050_BAND_21_HZ);

  time = millis();
}

void loop() {
```

```
/* Get new sensor events with the readings */

if (millis() > time+period)
{
  time = millis();
  angle = get_angle(1);

  if(angle<25)
  {
  angle_error = angle_setpoint - angle;
  PID_p = kp * angle_error;
  float angle_diference = angle_error - angle_previous_error;
  PID_d = kd*((angle_error - angle_previous_error)/period);

  if(-3 < angle_error && angle_error < 3)
  {
    PID_i = PID_i + (ki * angle_error);
  }
  else
  {
    PID_i = 0;
  }

  PID_total = PID_p + PID_i + PID_d;




  if (angle_error<=0){
  PID_total = map(angle, 0, -25, 0, -255);
  Serial.print(PID_total);
```

```
    //if(PID_total < 50){PID_total = 10;}
    if(PID_total > 255) {PID_total = 255; }

    digitalWrite(dir1, LOW);
    digitalWrite(dir2, HIGH);
    analogWrite(pwm, PID_total);
    }
    else{
    PID_total = map(angle, 0, 25, 0, -255);


    //if(PID_total < 100){PID_total = 100;}
    if(PID_total > 255) {PID_total = 255; }
    digitalWrite(dir1, HIGH);
    digitalWrite(dir2, LOW);
    analogWrite(pwm, PID_total);
    Serial.print(PID_total);

      }


    angle_previous_error = angle_error;
  }
}
}


  float get_angle(int n)
```

```
 {
 long sum=0;
 for(int i=0;i<n;i++)
 {
 sensors_event_t a, g, temp;
 mpu.getEvent(&a, &g, &temp);
 sum = sum + a.acceleration.x;

 }
 float a_x=sum/n;

 float angle =180*(asin(a_x/9.81))/3.14;

 Serial.print("_____");
 Serial.println(angle);
 return(angle);
//  delay(10);
}
```

RESULTS

- The system was not able to balance the stick.

REASONS FOR MISMATCHING

- Sensor noise: The sensors used to measure the angle and position of the stick may have noise, which can lead to inaccurate readings.
- Model inaccuracies: The model used to represent the self-balancing stick system may not be entirely accurate, leading to errors in the control algorithm.

- Delay in control: There may be a delay between when the control algorithm calculates the necessary control input and when it is applied to the system, leading to errors.
- Motor limitations: The motor used to move the stick may have limitations, such as limited torque or speed, which can make it difficult to achieve the desired control.
- Environmental disturbances: External factors such as vibrations, air currents, or uneven surfaces can also affect the balance of the stick, leading to errors in the control algorithm.
- Mechanical friction: Friction in the mechanical components of the self-balancing stick such as bearings, gears, or drive belts can create disturbances in the system, leading to errors in the control.

Overall, these sources of error can lead to inaccuracies in the control algorithm, which can result in instability or oscillations in the self-balancing stick system.

SCOPE OF IMPROVEMENT

- Model-based control: Using a more accurate model of the self-balancing stick system can lead to better control performance.
- Advanced control techniques: There are several advanced control techniques such as Model Predictive Control (MPC) or Nonlinear Model Predictive Control (NMPC) that can be used to improve the control performance of the self-balancing stick.
- Adaptive control: Adaptive control algorithms can adapt to changes in the system dynamics and parameter uncertainties, leading to better control performance.
- Sensor fusion: Combining data from multiple sensors such as accelerometers, gyroscopes, and encoders can improve the accuracy of the measurements and reduce the impact of sensor noise.

- Feedforward control: Incorporating a feedforward control component that takes into account the expected disturbances can improve the overall control performance of the system.
- Mechanical improvements: Improvements in the mechanical components of the self-balancing stick such as reducing friction, increasing motor torque or speed, or optimizing the weight distribution can improve the overall stability of the system.

Overall, incorporating these advanced techniques can help improve the control performance of the self-balancing stick system and make it more robust to uncertainties and disturbances.

REFERENCES

- https://www.youtube.com/watch?v=Ih-izQyXJCI&ab_channel=simplefoc
- https://ieeexplore.ieee.org/document/8646093

ACKNOWLEDGEMENT

In completing our project on **'Self-balancing stick by reaction wheel,'** we would like to convey our special gratitude to Mr. Jayaprakash KR. Your valuable guidance and suggestions helped us with project completion. We will always be thankful to you in this regard. We would also like to thank our TAs for their continuous motivation and support. We would also like to thank the Tinkerer's Lab and Machine Shop team for letting us build our project in it and providing us with the tools necessary for the project's successful completion.