**EE-361: Principles of Feedback Control – Fall 2019**

**Lab Project**

**[Landing Gear Mechanism]**

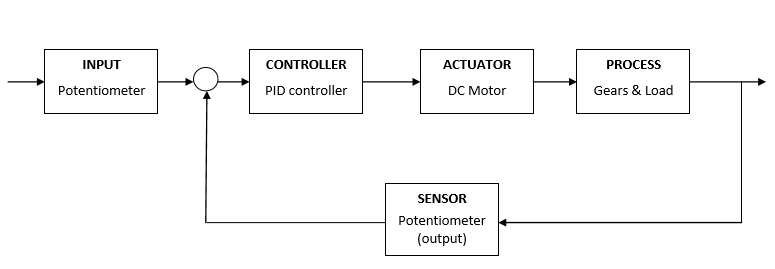
**[Yousuf Moiz (ya03482) & Syed Muhammad Affan (sa03571)]**

**Introduction**

A servo system is an electro mechanical or hydraulic system that can follow a desired position, speed or time in a closed loop fashion. A servo system consists of a motive apparatus, a feedback sensor and logic that command the motive apparatus to a desired set point.

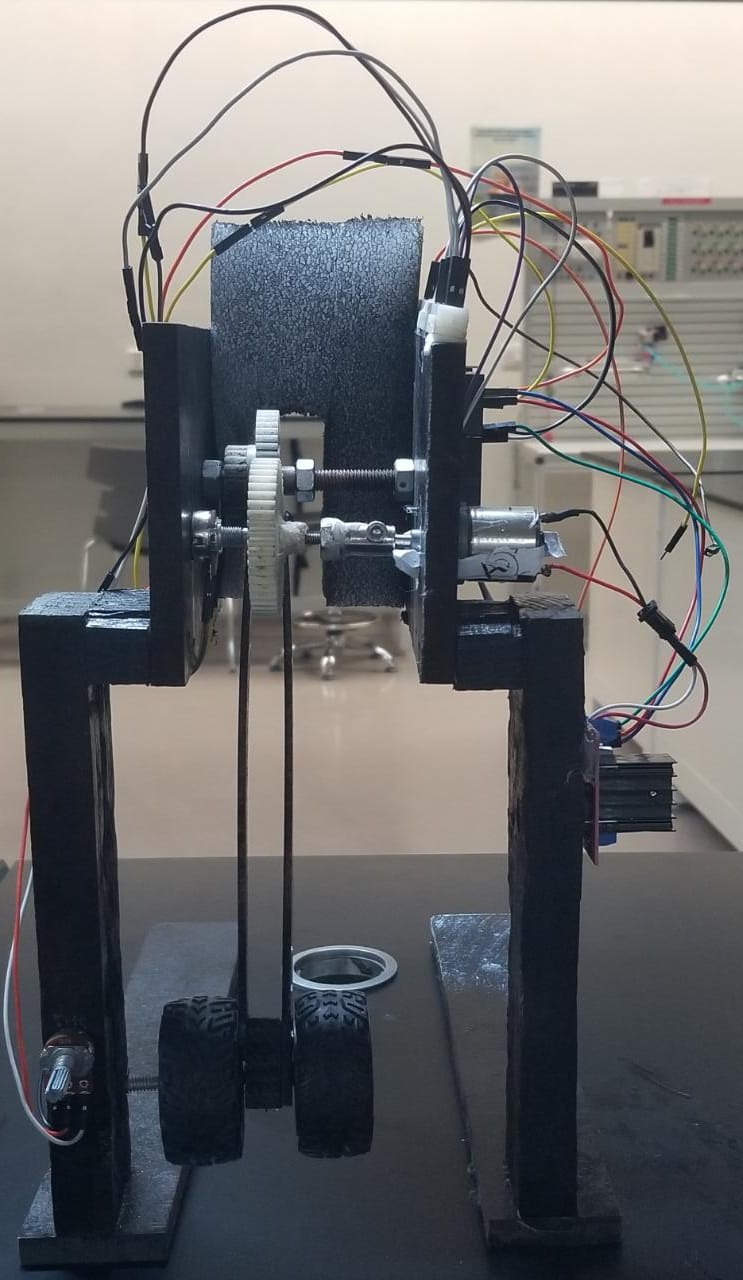
Using a servo system, we have design a retractable landing gear mechanism that is used in aircrafts to improve its efficiency, enabling them to fly faster. This system makes the landing of the aircraft more efficient. The main idea is that whatever input we give to the potentiometer as an angle it is converted to voltage at its output, which is sent to the P controller implemented using Arduino. The P controller sends the voltage signal to the motor and then the gears attached to the shaft of the motor rotates thus giving us our desired angle. Attached to the gears is our retractable mechanism. Rotation of the shaft rotates the gears causing our mechanism to fold the tire up or down when potentiometer is rotated clockwise or anti-clockwise. The output from the gears is fed back to through the potentiometer in the feedback path to the potentiometer at input which to correct for any error in rotation of the tire previously.

**Methods and Materials**

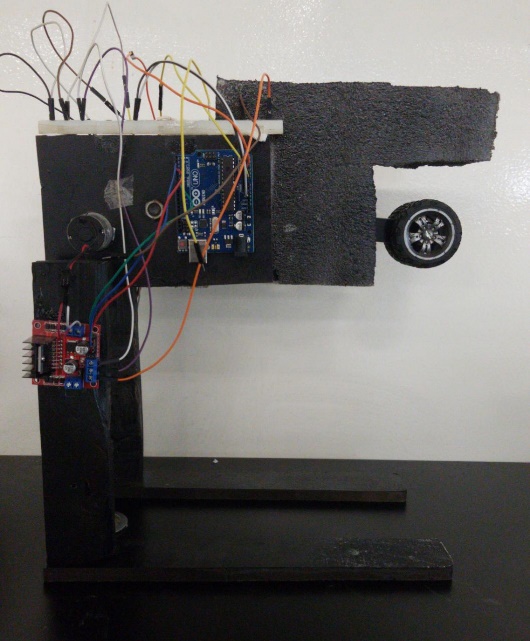
****

**Figure 01**: Block Diagram of System

The mechanical structure of our plant consists of a motor whose shaft is connected to a gear with a 48 teeths. This gear in turn is connected to another gear with 45 teeths. After coupling these two gears we have the main part of our mechanical structure. We have two tires connected to two acrylic rods which gives us a pendulum like arrangement. Basically, it can be visualized as a pendulum connected to the two gears with the tires replacing the bob at the end of the pendulum. This pendulum will be allowed to rotate 90 degrees, thus simulating the landing gear system of an airplane. Figure 01 shows the block diagram of our system. In our system we have used Proportional Controller. Actuator in our case is a DC motor. The Plant of our system comprises of Gears and Landing wheel. The block diagram of our system is given in the next page in which the process is the plant which consists of the pendulum coupled to the motor with the help of gears.

****

**Figure 02**: Front View



**Figure 03**: Side View

Figure 02 & Figure 03 shows us the picture of our Actual System.

**(1)**

Equation 1 gives us the open loop transfer function of our system.

**(2)**

Equation 2 gives us the closed loop transfer function of our system.

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Resistance of Motor (Ra) | 59.89 Ω |
| Inductance of Motor (La) | 30 mH |
| Inertia of Motor (Jm) | 7.621 × 10-5 kg.m2 |
| Back EMF Constant (Kb) | 0.0287 V-s/rad |
| Motor Constant (Km) | 0.0287 N.m/ |
| Damping Constant (Dm) | 1.577 × 10--6 N-s/m |
| Gear Ratio (N) | 0.9375 |
| Inertia of load (Jl) | 2.423 × 10-3 kg-m2 |
| Total inertia of system (Jt) | 7.665 × 10-5 kg-m2 |

**Table 01**: Summary of Motor parameters

Table 01 summarizes all the parameters that we obtained for our DC motor. Details on how we obtained these readings can be found in the appendix.

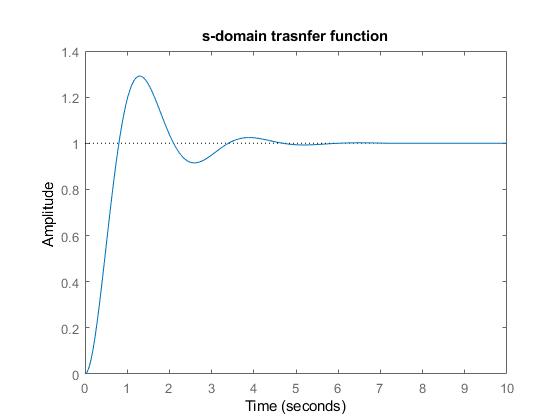
|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Mass of rod (mr) | 39.4 g |
| Mass of Tyres (Mr) | 40.6 g |
| Length of rod (L) | 23.5 cm |

**Table 02**: Summary of Plant Parameter

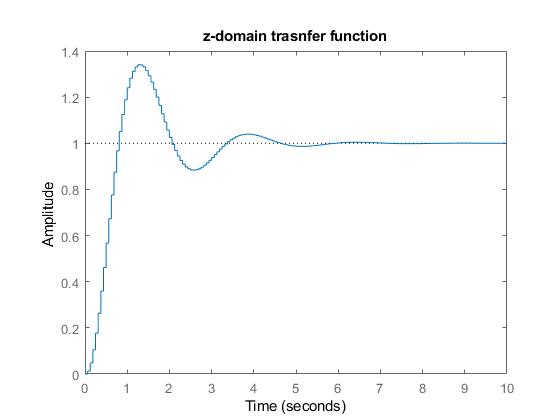
Table 02 summarizes all the parameters that we obtained for Plant i.e. the values of mass and length of landing gear mechanism.

**Results**

Given below is the motor position step response in different domains:



**Figure 04**: Step Response of Closed Loop System is S-domain



**Figure 05**: Step Response of Closed Loop System is Z-domain

Figure 04 & Figure05 represent the output from our derived model in S-domain and Z-domain respectively. From the actual system observations and our simulated results we can say that our derived model for the system is valid.

|  |  |
| --- | --- |
| **Controller** | **Observation** |
| Proportional | Best |
| Proportional + Differential | Satisfactory |
| Proportional + Integral | Worst |
| Proportional + Differential + Integral | Worst |

**Table 03**: Summary of Controllers

**Proportional Controller:**

The best output was obtained from this controller. We had the least value of overshoot, least number of oscillations and response time was nominal. We were able to get the best output at the value of Kp = 8.5

**Proportional and Integral Controller:**

Using this controller we were not able to achieve a desired fixed point. The system was unstable and there were constant oscillations.

**Proportional and Differential Controller:**

After proportional controller, the best output was obtained through this controller. Using this controller we were able to achieve the fastest response time.

**Proportional Integral and Differential Controller:**

Using this controller we were not able to achieve a desired fixed point. The system was unstable and there were constant oscillations.

**Conclusions**

We can say that we were able to achieve our purpose of controlling the position of the landing gear mechanism through the DC motor. There were a few challenges that we faced on our way:

* We used logarithmic potentiometer for our input and output sensor, which made our system vulnerable to even small changes in input.
* The calibration of the gears was also another issue. When the gears were not calibrated accordingly, then the friction between them increased which affected the efficiency of our system.
* The effect of the force of gravity also affected the overall performance of our controller as the landing gear sometimes reached undesirable positions.

The best output in our case was seen using a proportional controller whereas PID controller turned out to be the worst controller for our system. The system was very unstable when controlled with a PID controller. We also observed that our simulation and actual output match each other. From simulation, we saw that there is a bit of overshoot which can be seen in our actual model as well. Due to the time constraints, we were not able to improve the efficiency of our system. Overall, we can say that our results were satisfactory.

**References**

* Understanding small, permanent magnet brushed DC motors ( Gideon Gouws, March 2008).

**Appendices**

* **Derivation of Motor Parameters**

By using first principles, we came up with the below values:

* **Ra :** To calculate the value of Ra we simply measured the resistance across the positive and negative terminals of the motor through a multi-meter. The value that we were getting was 58.89 Ω.
* **La :** To calculate La, we applied a sinusoidal voltage to the motor and measured the rms values of voltage and current across the motor. We did this experiment for various frequencies for the sinusoidal voltage and calculated the impedance from the rms values of voltage and current. Then we used the following formulae to calculate L for each frequency,

We averaged all the values of L obtained through this formula which gave us a value of 0.996 H. This value was obviously very high and inaccurate, therefore, we measured the value of La through the LCR meter, which gave us a reading of 30 mH, which was an accurate value, therefore, we opted for this value of La.

* **Km /Kb :** To calculate the motor and back emf constants, we used the circuit diagram of a motor and applied KVL on the circuit to obtain the equation:

La = 0 because we are applying a DV voltage and inductors are shorted when a DC voltage is applied across them.

Omega ( was found by noting the motor shaft rpm which was 57 and then calculating the inner gear rpm by multiplying it by the gear ratio which was 70:1. Va was just the DC voltage applied to motor, which was 12 V, and Ia was measured through multi-meter, which came out to be 328 µA. Kb was then calculated by using the above formula and it was found to be 0.0287 V-s/rad. Since Km is equal to Kb, therefore, Km = 0.0287 N.m/ .

* **Dm =** To find the value of Dm, we used the following formula:

The value was found to be 1.577 × 10--6 N-s/m.

* **Jm :** To find Jm, we conducted an experiment in which the we used a low-rpm motor whose J can be assumed to be zero. We coupled both the motors together and applied an input sinusoidal voltage to the bigger motor. The voltage applied to the bigger motor induces a voltage on the smaller motor whose graph was recorded via an oscilloscope. Then we found the time constantof the graph, which was found to be 4.97s, and used the formula below to calculate Jm:

This equation was obtained by making the transfer function of w(s)/Va(s) a first order transfer function and then measuring the time constant of this function through an oscilloscope.

The value of Jm was found to be 7.621 × 10-5 kg.m2.

* **Derivation of Systems Transfer Function**

**General Transfer function of motor:**

**Transfer Function of our motor:**

**Transfer Function of Plant:**

Our system can be assumed to be a model of a rigid physical pendulum, whose inertia can be found by the following equation:

Where,

* Mass of rod.
* Mass of tires.
* Radius of tires.
* Length of rod from pivot position.
* **Arduino Code:**

int enA = 7;

int in1 = 2;

int in2 = 3;

int input = A1;

int output = A0;

float r = 0;

float y = 0;

float e = 0;

float MAPr = 0;

float MAPy = 0;

float MAPe =0;

int motorOutput = 0;

float Kp = 8.5;

float Ki = 0;

float Kd = 0;

double drerror = 0;

double elapsedTime;

double cumError;

double currentTime;

double previousTime;

void setup() {

  // put your setup code here, to run once:

  pinMode(enA, OUTPUT);

  pinMode(in1, OUTPUT);

  pinMode(in2, OUTPUT);

  pinMode(input, INPUT);

  pinMode(output, INPUT);

  Serial.begin(9600);

  cli();//stop interrupts

  //set timer1 interrupt at 1Hz

  TCCR1A = 0;// set entire TCCR1A register to 0

  TCCR1B = 0;// same for TCCR1B

  TCNT1  = 0;//initialize counter value to 0

  OCR1A = 1000;// = (16\*10^6) / (1\*1024) - 1 (must be <65536)

  TCCR1B |= (1 << WGM12);

  TCCR1B |= (1 << CS12) | (1 << CS10);

  TIMSK1 |= (1 << OCIE1A);

  sei();//allow interrupts

}

// Interrupt routine

void motorControl(float error)

{

  if(error == 0.0)

  {

    analogWrite(enA , motorOutput);

  }

  else if(error < 0.0)

  {

      digitalWrite(in1, HIGH);

      digitalWrite(in2, LOW);

      analogWrite(enA ,motorOutput);

  }

  else

  {

     digitalWrite(in1, LOW);

     digitalWrite(in2, HIGH);

     analogWrite(enA ,motorOutput);

  }

}

ISR(TIMER1\_COMPA\_vect)

{

  r = analogRead(input);//\*12/500;

  y = analogRead(output);

  MAPr = map(r , 0, 1023 , 0, 255);

  MAPy = map(y , 0, 1023 , 0, 255);

  e = r-y;

  MAPe = map(e , 0, 1023 , 0, 255);

  currentTime = millis();

  elapsedTime = (currentTime - previousTime);

  cumError += MAPe \* elapsedTime;

  drerror = e / elapsedTime;

  motorOutput = (Kp\*MAPe) + (Ki\*cumError)+ (Kd\*drerror);

    if (motorOutput < 0)

    {

      motorOutput = -motorOutput;

    }

    Serial.print(MAPr);

    Serial.print(",");

    Serial.print(MAPy);

}

void loop() {

  motorControl(MAPe);

}