MCT432: Hybrid Control Systems

Project Report

Submitted by:

Team 3

1. Hend Mohamed (17P8140)
2. Ahmed Sherif (17P8028)
3. Kerlos Rady (17P8178)
4. Verina Goerge (17P8106)
5. Reem Hesham (17P8047)

Submitted to:

1. Dr. Mohamed Ibrahim
2. Eng. Hesham Salam

Contents

List of Figures

1. Project description and features:

We are designing a gripper actuated using four-bar mechanism. It has four states depending on the power and pushbuttons. The states are:

- Stop.
- Start.
- Forward Motion.
- Backward Motion.
- Gripping.

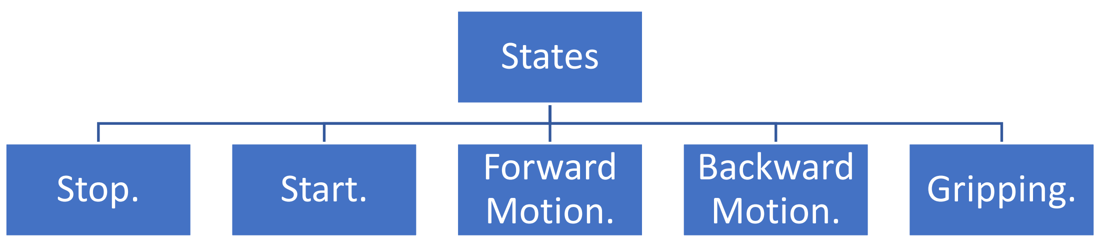


Figure 1: States

Forward Motion, Backward Motion, and Gripping states are indicated by three LEDs. Each motion is actuated using its pushbuttons or by command from the mobile application. The mobile application is communicating with the Arduino using Bluetooth module. Potentiometer is used to select slow or rapid motion of the four-bar mechanism (Stepper motor). The State is displayed live on the LCD display.

2. Design Methodology and Flowcharts of the Program's Main Flow:

2.1. Design Methodology:

The first step of the project is to receive all inputs from Arduino pins and preprocess any noise. Then these inputs enter the state flow. The stop state is the default system state with power switch not off. The system is idle and waiting for the switch to be on and for the communication between the mobile app and the Arduino to establish. This idle state has the highest priority from all other state and is only when power is off at any time. When the system enters the start states it is waiting for one of the pushbuttons to be pressed or waiting for a command to be sent from the mobile application. After the blue button is pushed, the FW state starts and the four-bar mechanism moves forward and a blue LED indicates the states is turned on, the system runs this state until it reaches the limit switch, or the button is released. Then it returns to the start state. If the green button is pushed, the BW state starts and the four-bar mechanism moves backward, until it reaches the limit switch. Then it returns to the start state again. The final state is the gripping state, it uses a PID controller to control a servo motor based on the pressure required to hold the object.

2.2. Flow Chart:

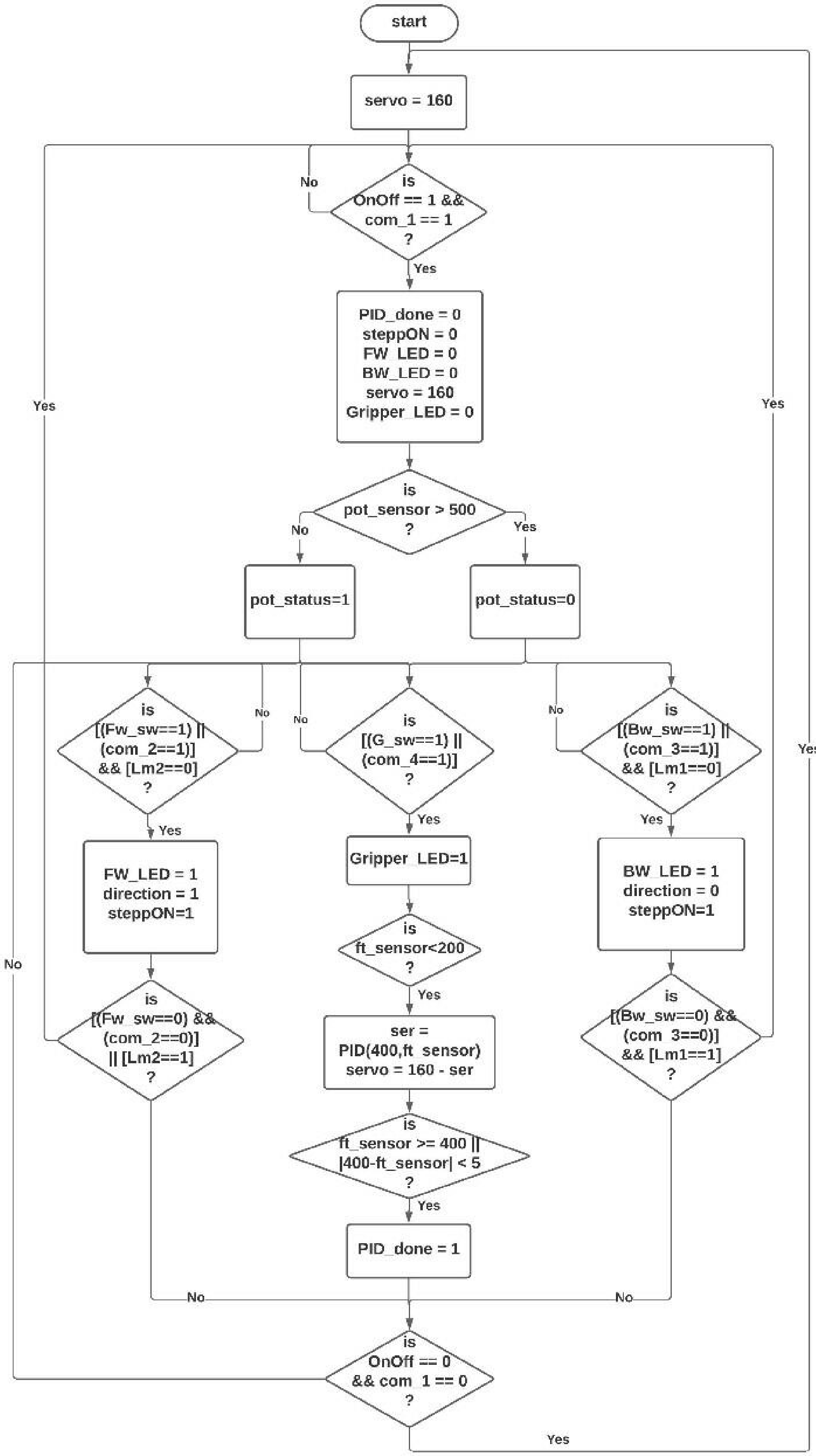


Figure 2: Flow Chart

3. The list of components used and the circuits topology:

3.1. List of Components:

The project is constructed from the following components:

- Three modes indicating LEDs (Blue, Red, Green).
- Three pushbuttons (Blue, Red, Green).
- Two limit switches.
- One servo motor for the gripper actuation.
- One stepper motor for the four-bar mechanism actuation.
- Potentiometer for stepper speed.
- LCD display for states.
- Bluetooth module to control the project from mobile application.

3.2. Circuit Topology:

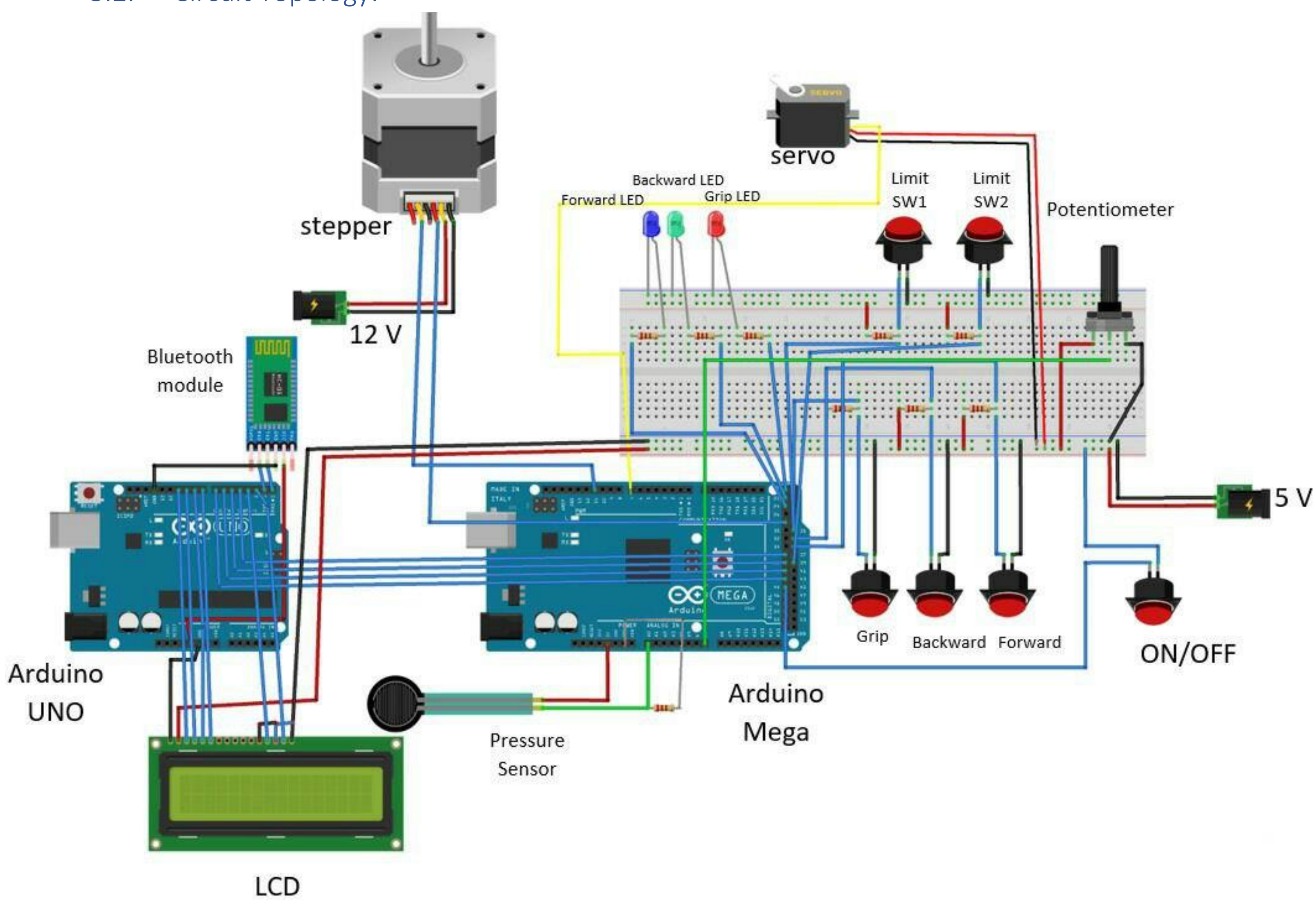


Figure 3: Circuit

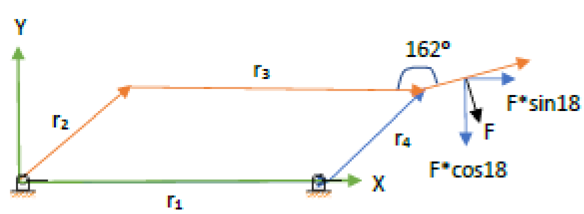
4. System mechanical construction:

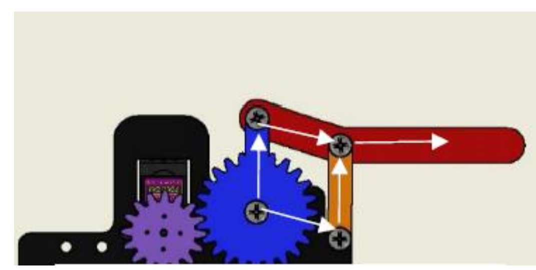
Calculations

we use all the friction forces to study the 4-bar mechanism of half of the gripper to get the mechanical advantage and to determine the motor sizing of the gripper motor as following:

Four Bar Mechanism “Gripper”

- $r1 = 36.81$ mm
- $r2 = 40$ mm
- $r3 = 36.81$ mm
- $r4 = 40$ mm
- $\theta 1 = 0°$





Vector Loop: $r_1 + r_4 = r_2 + r_3$

Position:

$$r_1\cos\theta_1 + r_4\cos\theta_4 = r_2\cos\theta_2 + r_3\cos\theta_3$$
$$r_1\sin\theta_1 + r_4\sin\theta_4 = r_2\sin\theta_2 + r_3\sin\theta_3$$

$\theta 2 = 68°$

$36.81 + 40\cos\theta 4 - 40\cos(68) = 36.81\cos\theta 3$
$40\sin\theta 4 - 40\sin(68) = 36.81\sin\theta 3$

➢ By squaring both sides and adding the equations:

$(36.81)^2 + 2(36.81)(40\cos\theta 4 - 40\cos(68)) + (40)^2 - 2(40)(40)(\cos(68))(\cos\theta 4) + (40)^2(\cos2(68)) + (40)^2(\sin^2(68)) - 2[(40)\sin(68)][(40)\sin\theta 4] = (36.81)^2$

➢ Solve to get θ4:

- $\theta 4 = 68°$

➢ Substitute to get θ3:

- $\theta 3 = 0°$

Therefore, the shape is of a parallelogram.
Differentiate Position with respect to θ2:
$- r4(\sin\theta 4)h4 = - r2\sin\theta 2 - r3(\sin\theta 3)h3$
$40\sin(68)h4 = 40\sin(68) + 0$

h4 = 1

r4(cosθ4)h4 = r2cosθ2 + r3(cosθ3)h3
40cos(68)(1) = 40cos(68) + 36.81h3
h3 = 0

rc(x) = r4cosθ4 + rc/cosθ3
Fc(x) = - r4(sinθ4)h4 – 0
Fc(x) = -40sin(68)(1) = -37.1
rc(y) = r4sinθ4 + rc/sinθ3
Fc(y) = r4(cosθ4)(h4) + 0
Fc(y) = 40cos(68)(1) = 15

$$V_c = \begin{bmatrix} F_{cx} \\ F_{cy} \\ 0 \end{bmatrix} \dot{\theta}_2 = \begin{bmatrix} -37.1 \\ 15 \\ 0 \end{bmatrix} \dot{\theta}_2$$

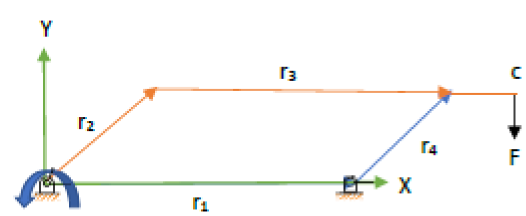
P = $\boldsymbol{T*\omega+F*V=0}$

$$P = \boldsymbol{T} * \dot{\theta}_2 + \begin{bmatrix} \boldsymbol{sin18} \\ \boldsymbol{-cos18} \\ \boldsymbol{0} \end{bmatrix} \mathbf{FG} \cdot \begin{bmatrix} -37.1 \\ 15 \\ 0 \end{bmatrix} \dot{\theta}_2 = 0$$

T1 = 31.65 * 15cos18 – 31.65 *-37.1sin18 = 814 N.mm
T1' (torque of the two gears of the gripper) = 8.14 x 2 = 16.28 kg.cm
T2 = $\boldsymbol{T1/Reduction\ Rate}$=$\mathbf{16.28/1.5}$=$\mathbf{10.85}$ $\boldsymbol{kg.cm}$

Four Bar Mechanism

➢ Assume:



- r1 = r3 = 20 mm
- r2 = r4 = 15 mm
- θ1 = 0°
- θ2 = 30°

Vector Loop:

r1 + r4 = r2 + r3

Position:

r1cosθ1 + r4cosθ4 = r2cosθ2 + r3cosθ3

r1sinθ1 + r4sinθ4 = r2sinθ2 + r3sinθ3

➢ Substitution:

20 + 15cosθ4 – 15cos(30) = 20cosθ3

15sinθ4 – 15sin(30) = 20sinθ3

➢ By squaring both sides and adding the equations:

Let U = $tan\frac{\theta_4}{2}$, $cos\theta_4 = \frac{1-U^2}{1+U^2}$, $sin\theta_4 = \frac{2U}{1+U^2}$

$$210.3(1-U^2) - 225(2U) = 69.75(1+U^2)$$

$$210.3 - 210.3U^2 - 450U = 69.75 + 69.75U^2$$

$$210U^2 + 450U - 140.55 = 0$$

➢ Solve to get θ4:

- U = -1.8748 , U = 0.2677
- θ4 = 236.15° , θ4 = 30° ✓

➢ Substitute to get θ3:

θ3 = 0° or 180°

from geometry

- θ3 = 0°
- r5 = 20 mm
- θ1 = θ5 = 0°
- **θ5=θ1=0**

Vector Loop: $rc = r4 + r5$

Position:

rc(x) = r4cosθ4 + r5cosθ5 = 15cos(30) + 20cos(0) = 33 cm

rc(y) = r4sinθ4 + r5sinθ5 = 15sin(30) + 20sin(0) = 7.5 cm

Fc(x) = - r4(sinθ4)h4 = - 15(sin(30))h4 = - 7.5h4

Fc(y) = r4(cosθ4)(h4) = 15(cos(30))h4 = 13h4

$$V_o = \begin{bmatrix} f_{cx} \\ f_{cy} \\ 0 \end{bmatrix} * \dot{\theta}_2 , \omega = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} * \dot{\theta}_2 , \overrightarrow{T_D} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} T_D , \overrightarrow{F_L} = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} F_L$$

➢ Differentiate position equations to get h3, h4 with respect to θ2:

$$-r4h4sin\theta4 = -r2sin\theta2 - r3h3sin\theta3 \quad r4h4cos\theta4 = r2cos\theta2 + r3h3cos\theta3$$

➢ Substitution:

$$-15h4sin30 = -15sin30 - 20h3sin0 \quad 15h4cos30 = 15cos30 + 20h3cos0$$

➢ Solve to get h3 & h4:

- h3 = 0
- h4 = 1

➢ Solve to get fcx & fcy:

- fcx = -7.5
- fcy = 13

$\boldsymbol{P=0=T\omega+Fv}$

$\boldsymbol{T\omega=-Fv}$

$\boldsymbol{T\,\theta 2=FL\,fcy\,\theta 2}$

Assume FL = 1 kg = $\boldsymbol{1\ x\ 9.81=9.81\ N}$

$T_{\text{Driven Gear}}$ = $\boldsymbol{9.81\ x\ 13=127.53\ N.cm=1275.3\ N.mm}$, $T_{\text{Driver Gear}}$ = $\boldsymbol{1275.3/3=}$

$\boldsymbol{425.1\ N.mm=4.251\ kg.cm}$, M_a = $\boldsymbol{F_L/T=1/f_{cy}=0.077\ cm^{-1}}$

4.1. CAD:

We used Black Acrylic for all the major components while using wood for the supports, base and the electric box.

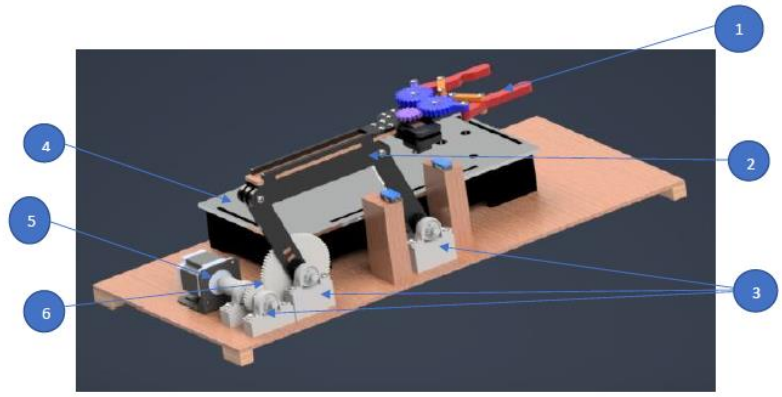


Figure 4: CAD

4.1.1. Gripper Assembly

The gripper components are the claws, four bar mechanism with gears meshed with servo motor gear and base assembly to the rest of the mechanism. All materials for the gripper are acrylic 6mm with spacers also used. The servo used has a 17 kg.cm rated torque. The gearing ratio is 1 to 1.5.

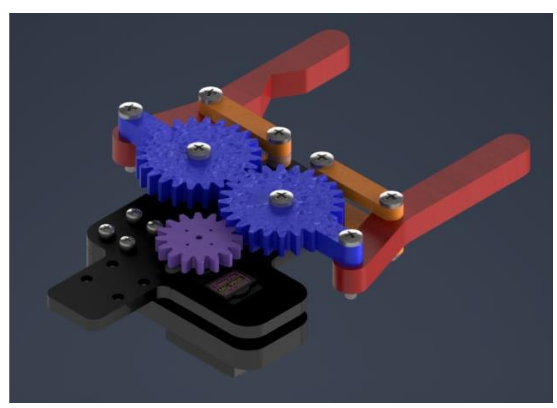


Figure 5: Gripping Mechanism

4.1.2. Four Bar Mechanism Assembly

Four bar components are crank, coupler, cross, rocker, shafts. All materials for the gripper are acrylic 6mm.

4.1.3. Shaft and Bearing Assembly

The shaft used was a threaded steel shaft of 10 mm diameter with vertical bearings of the same diameter.

4.1.4. Electric Box

It houses the cables, Arduino, and Buttons.

4.1.5. Motor & Fixation

The Motor used was a Nema 23 having 16 kg.cm of rated torque.

4.1.6. Gears

The gears are also made of 6mm acrylic having a ratio of 1 to 3.

4.2. Drawings:

4.2.1. Gripper Claws:

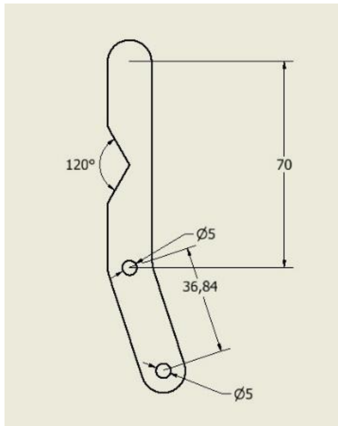


Figure 6: V-Shape Gripper Claw WD

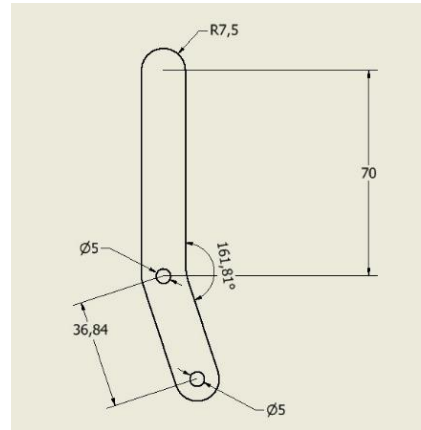


Figure 7: Flat Gripper Claw WD

4.2.2. Gripper Links & Bases:

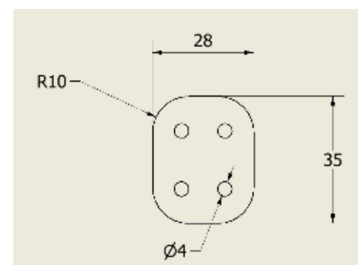


Figure 8: Spacer

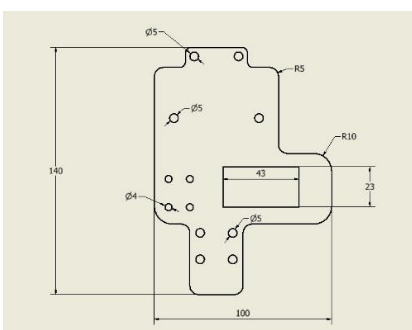


Figure 9: Gripper Base

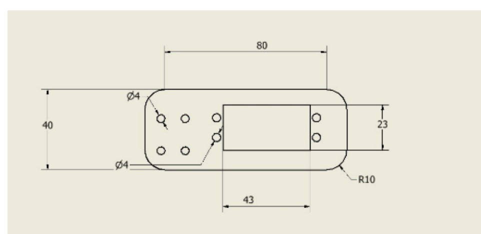


Figure 10: Servo Base

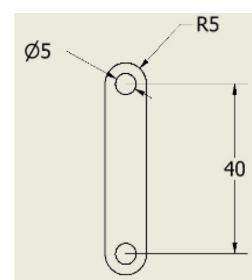


Figure 11: Gripper Link

4.2.3. Crank:

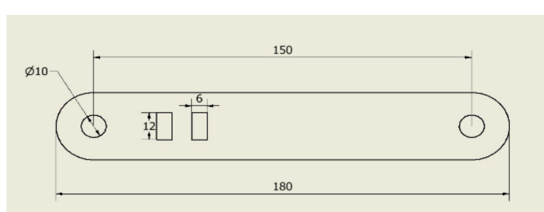


Figure 12: Crank WD

4.2.4. Coupler:

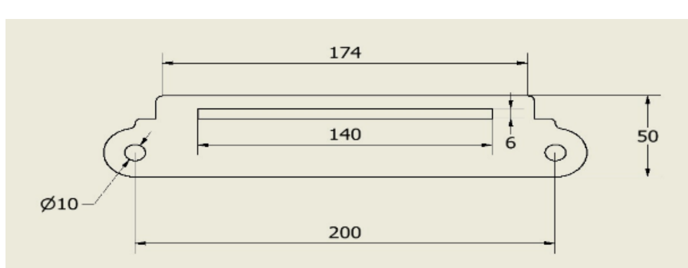


Figure 13: Coupler WD

4.2.5. Cross:

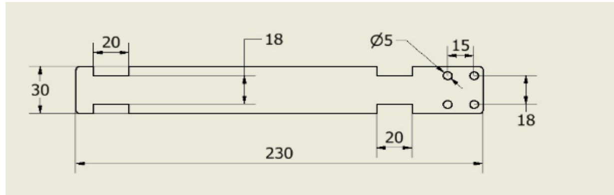


Figure 14: Cross WD

4.2.6. Rocker:

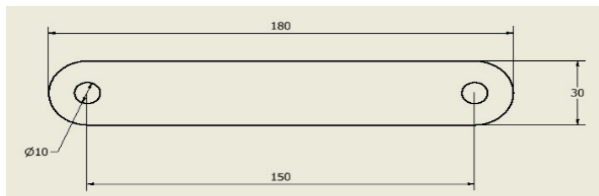


Figure 15: Rocker WD

4.2.7. Shafts:

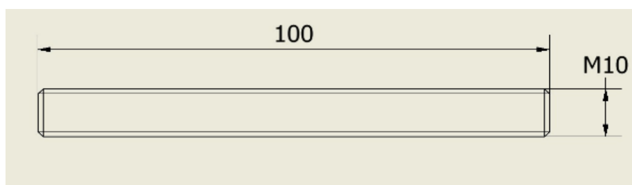
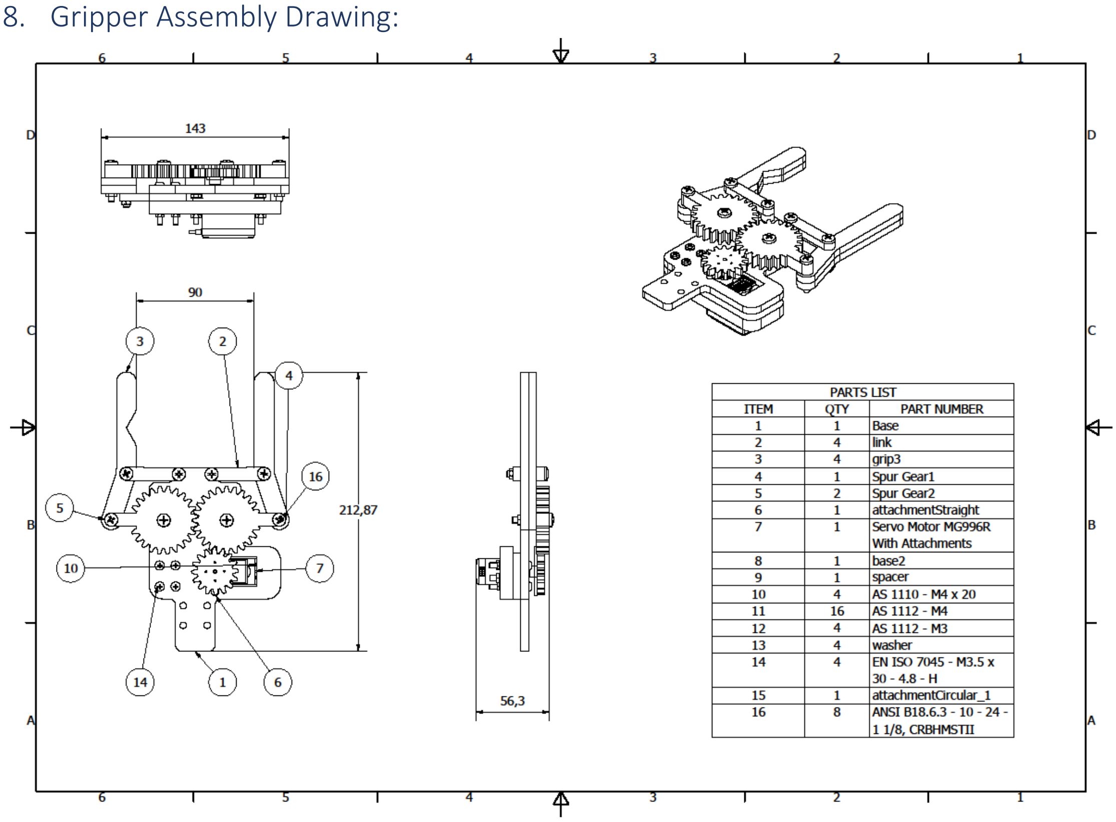


Figure 16: Shaft WD

4.2.8. Gripper Assembly Drawing:



PARTS LIST

ITEM	QTY	PART NUMBER
1	1	Base
2	4	link
3	4	grip3
4	1	Spur Gear1
5	2	Spur Gear2
6	1	attachmentStraight
7	1	Servo Motor MG996R With Attachments
8	1	base2
9	1	spacer
10	4	AS 1110 - M4 x 20
11	16	AS 1112 - M4
12	4	AS 1112 - M3
13	4	washer
14	4	EN ISO 7045 - M3.5 x 30 - 4.8 - H
15	1	attachmentCircular_1
16	8	ANSI B18.6.3 - 10 - 24 - 1 1/8, CRBHMSTII

Figure 17: Gripper Construction Drawing

5. PID Controller tuning:

The PID is used to control the gripping pressure using the servo angle,

The Gripping force is calculated as:

$$F_k = \frac{mgsin(\propto_1)}{\mu(\sin(\propto_1) + \sin(\propto_2) + \sin(\propto_3))}$$

Where $\propto_1, \propto_2, \propto_3 = 120\ ,30, 30$ and they represent gripper parameters and m is the mass of the object that is assumed to be 0.05 kg.

$$F_k = \ 7.4\ N$$

$$P = M\sqrt[3]{F_k * \frac{E^2}{r^2}}$$

$$r = \sin(theta) * l * 2$$

Where E = 0.5 GPa, M = 0.388, μ = 0.3, L= 0.35.

By modelling the system on MATLAB:

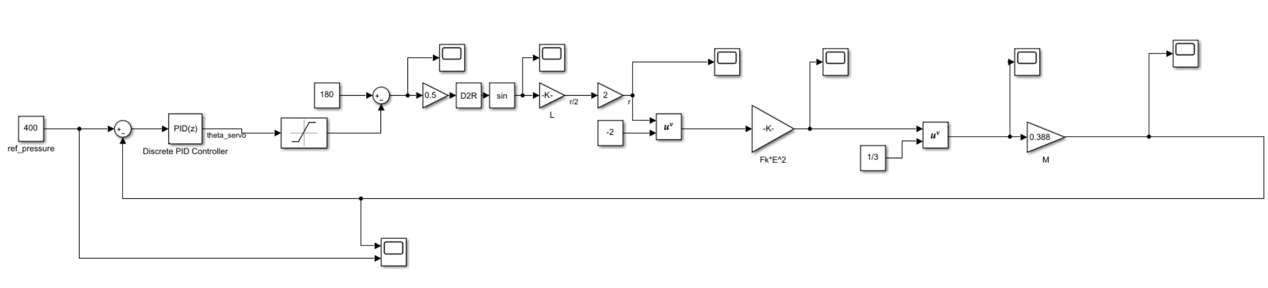


Figure 18: Gripper model with PID

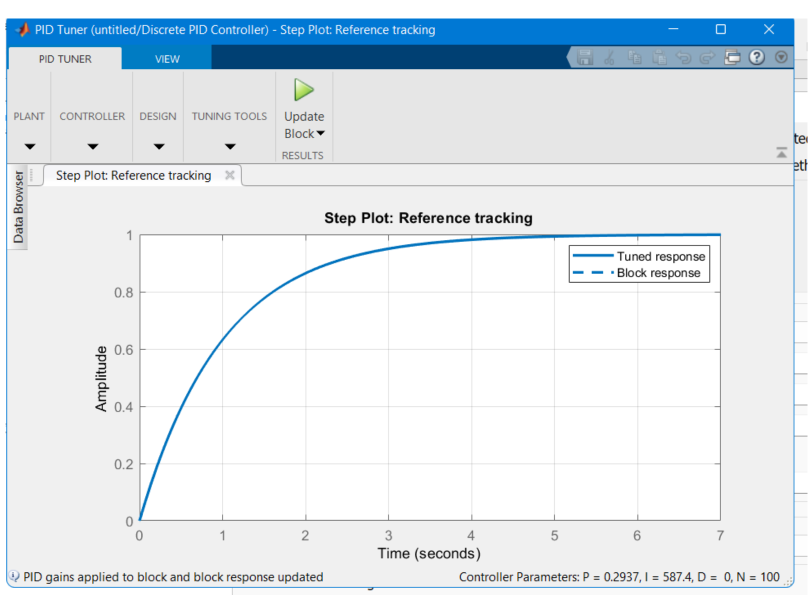


Figure 19: PID Tuner App

Figure 20: PID Tuner Response

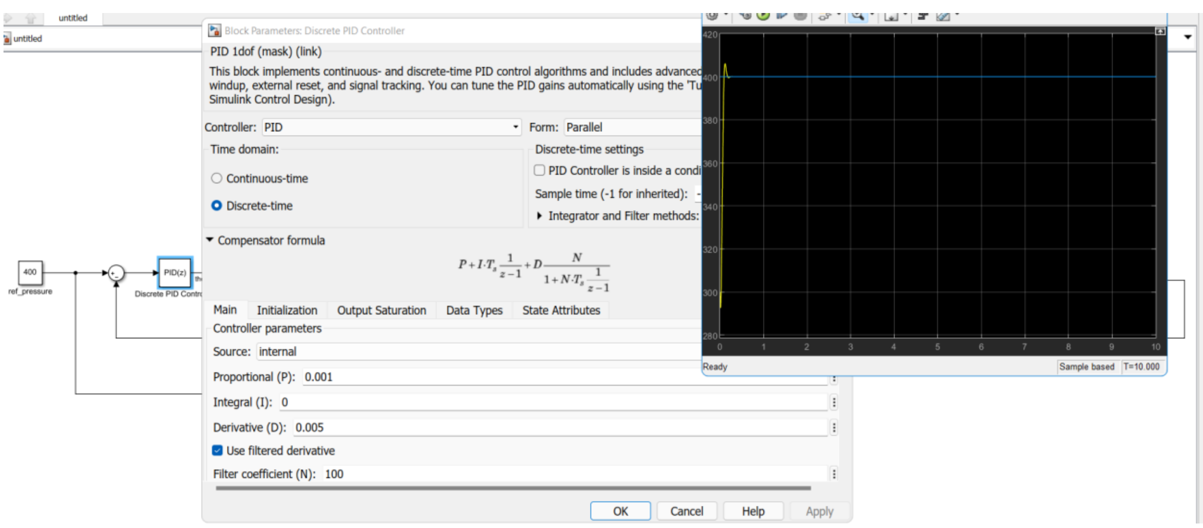


Figure 21: PID After Manual Tuning

6. Simulink model screenshot:

6.1. Full Simulink Model:

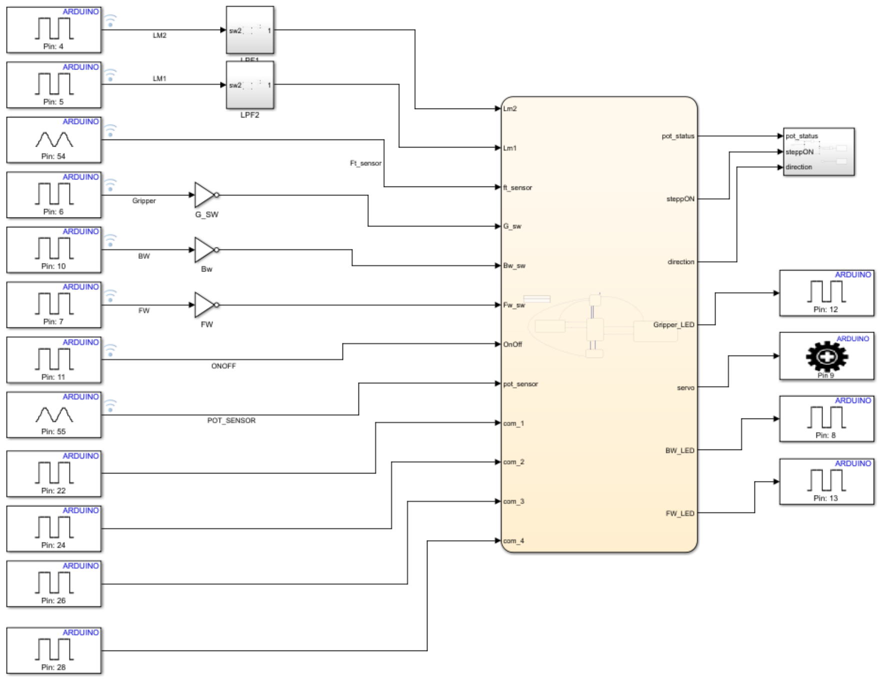


Figure 22: Simulink Model

6.2. Simulink Input:

Our inputs are:

- Limit Switches.
- Potentiometer.
- Three switches.
- Force Sensor
- Power Switch
- Four pins for communication with another Arduino (The one controlling LCD and Bluetooth module).

The inputs are read using the Arduino support inputs. Some inputs are denoised using the "Low pass filter" subblock and some is inverted. All inputs are then sent to the state block.

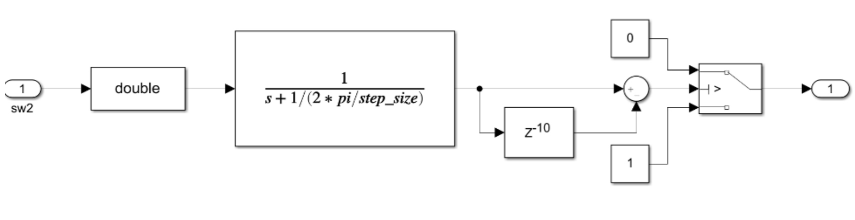


Figure 24: Our LPF Sub-Block

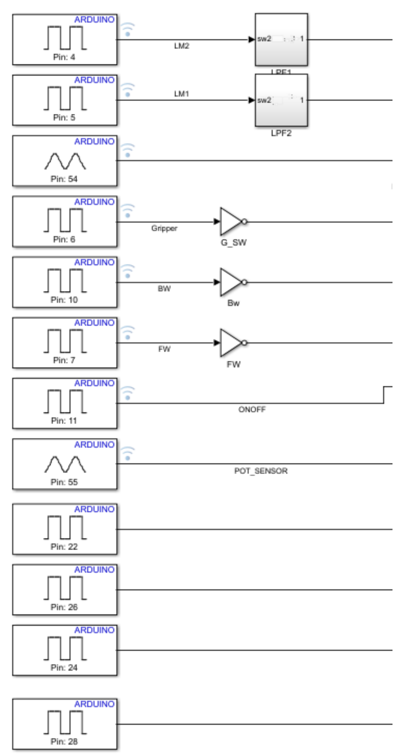


Figure 23: Arduino Inputs

6.2.1. Arduino Uno:

- Controlling the LCD and the receiving command from the mobile app using Bluetooth module.
- Sending actuation commands to pins 22, 24, 26, and 28 on Arduino Mega.

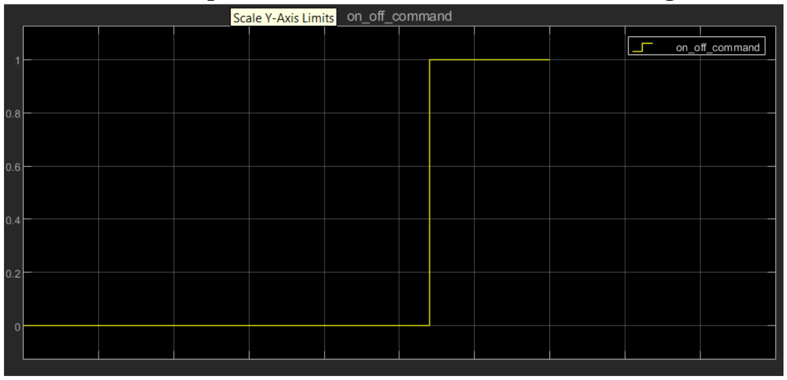


Figure 25: On- Off Bluetooth Command

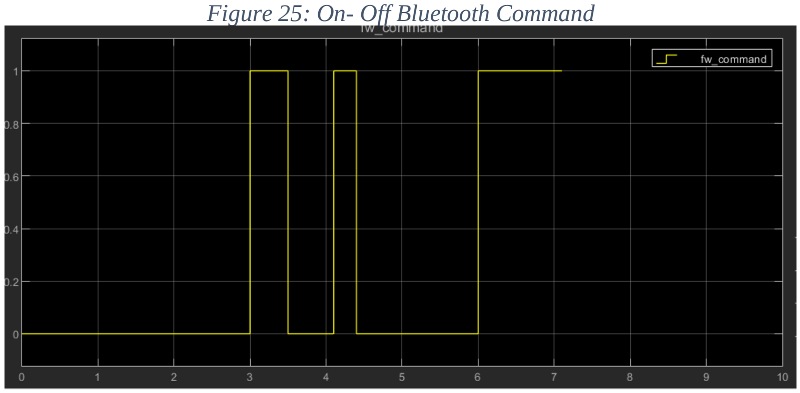


Figure 26: Forward Bluetooth Command

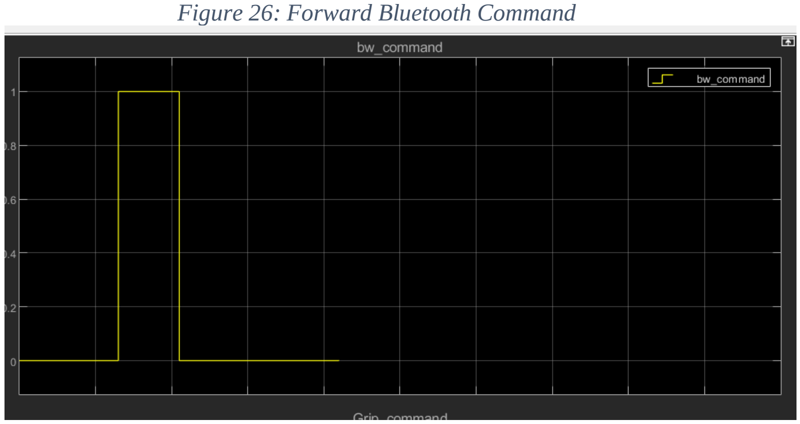


Figure 27: Backward Bluetooth Command

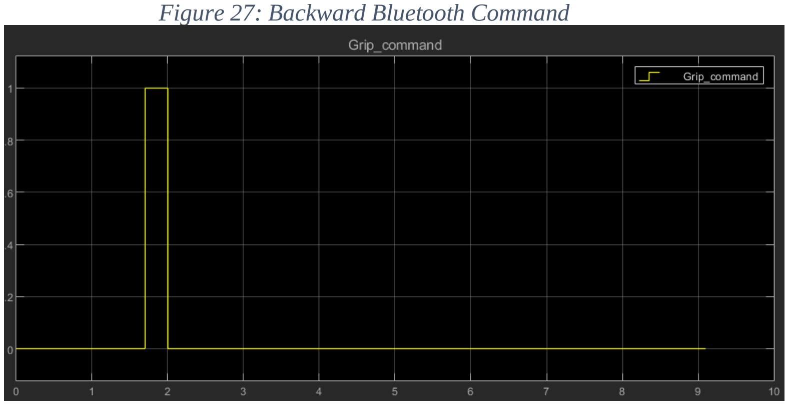


Figure 28: Gripping Bluetooth Command

6.2.2. On-Off Switch:

- Controls power.
- Connected to pin 11 on Arduino Mega

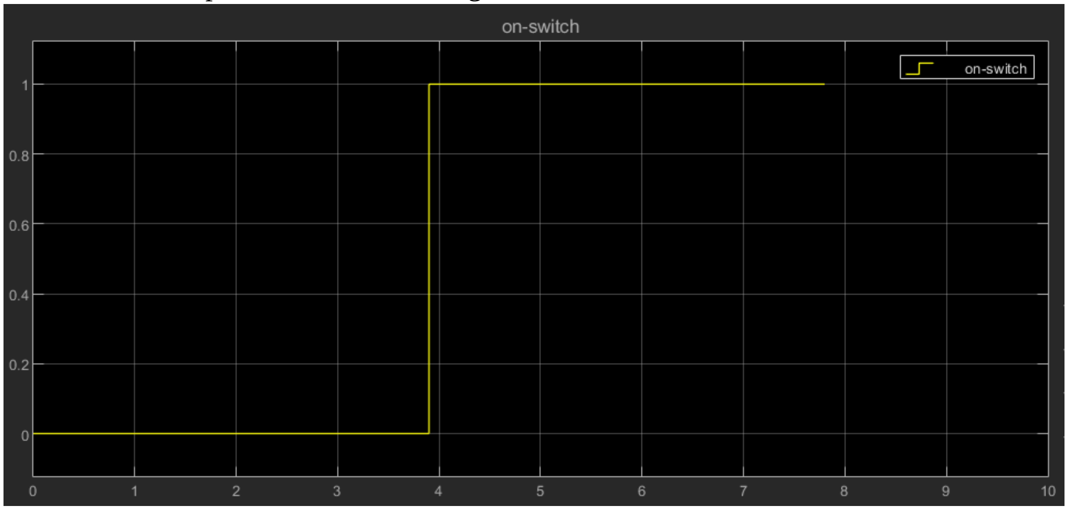


Figure 29: On- Off Switch

6.2.3. Gripping Switch:

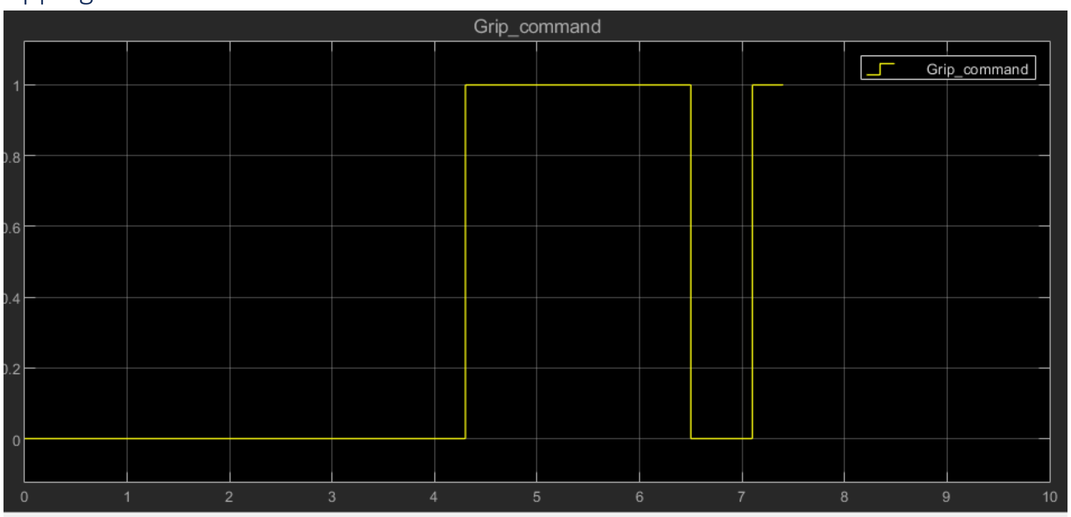


Figure 30: Gripper Switch

6.2.4. Backward Switch:

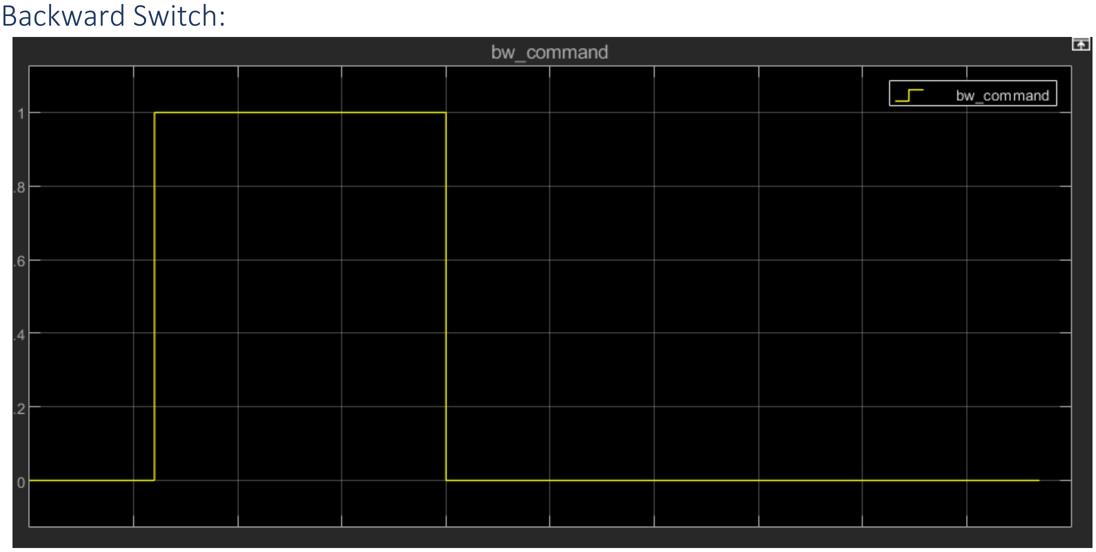


Figure 31: Backward Switch

6.2.5. Potentiometer Sensor:

- Connected to Arduino Mega pin 0.

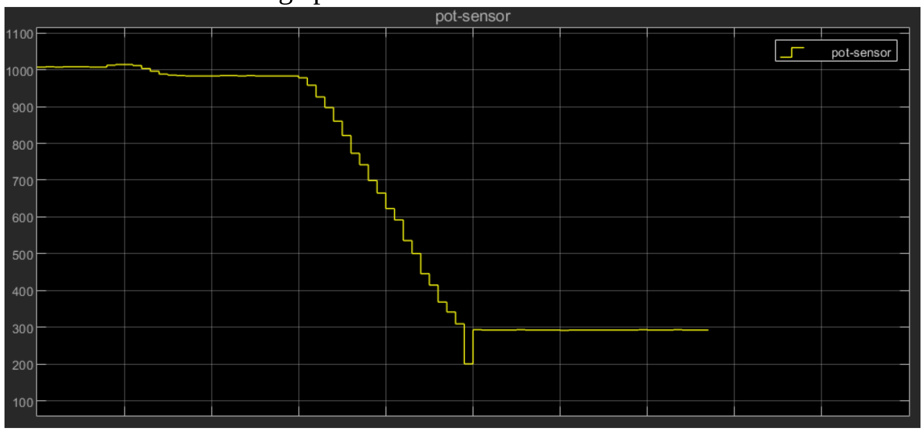


Figure 32: Potentiometer Sensor

6.2.6. Limit switches:

- Used to stop the forward and backward motion.
- Connected to pins 4 & 5 on Arduino Mega.

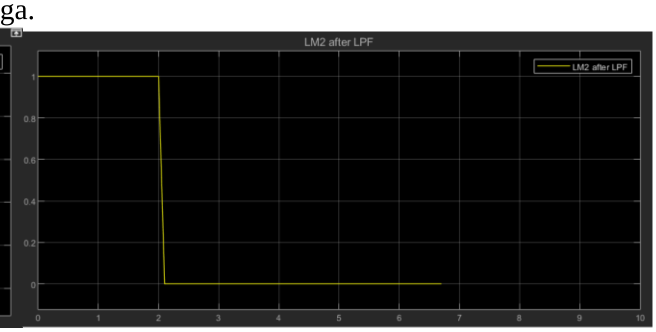


Figure 33: Limit Switch (1) after LPF

Figure 34: Limit Switch (2) after LPF

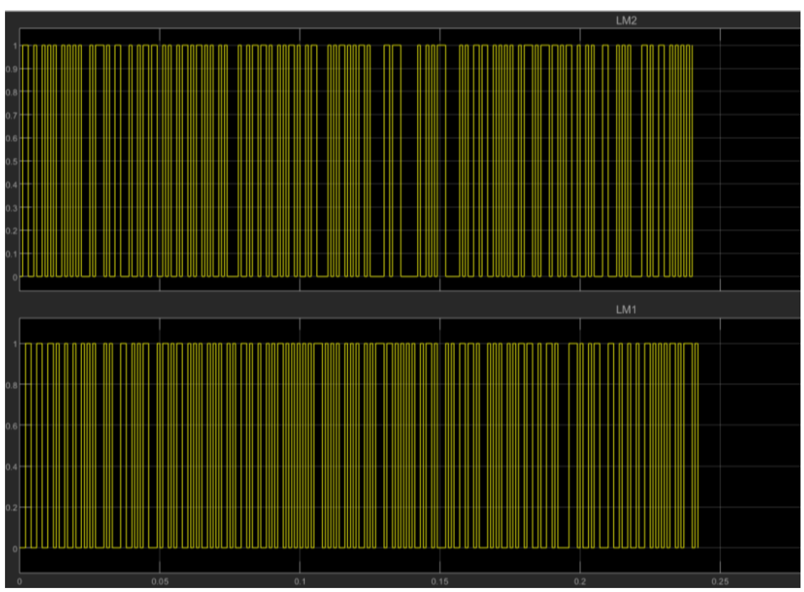


Figure 35: Limit Switched before LPF

6.2.7. Force Sensor:

- Connected to Arduino Mega pin 0.

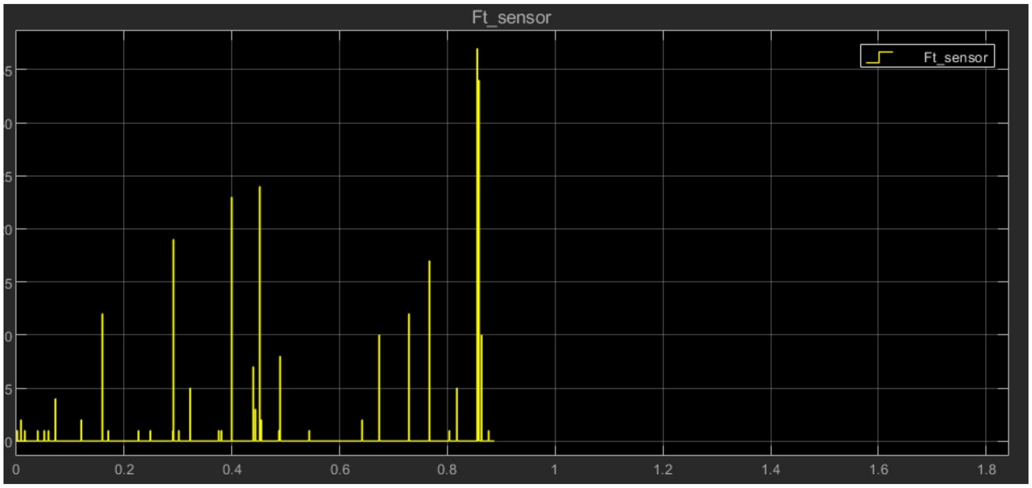


Figure 36: Force Sensor

6.3. Simulink Outputs:

Our Outputs are:

- Stepper Motor
- Servo Motor
- Three LEDs.

The output of the servo motor is the desired angle. And for three LEDs is a binary number. Commands are sent over to the Arduino using the supported output signal.

The third output is the stepper motor. The speed of the stepper depends on the value of the potentiometer. For high values of the potentiometer the stepper has high speed. The variable pot_status is a binary output of if speed is high or low.

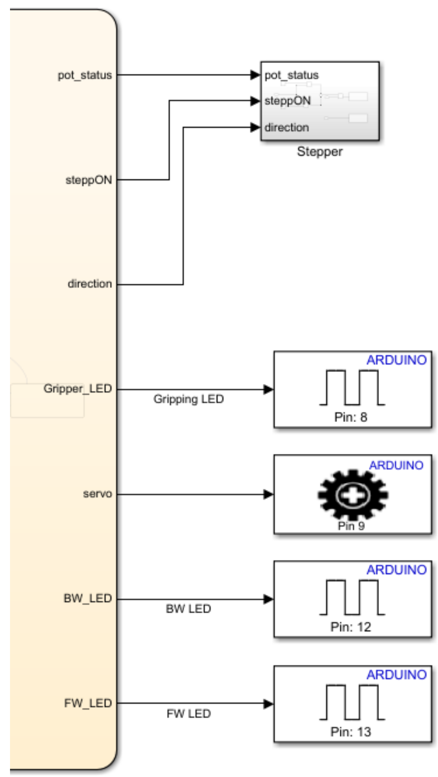


Figure 37: Model Outputs

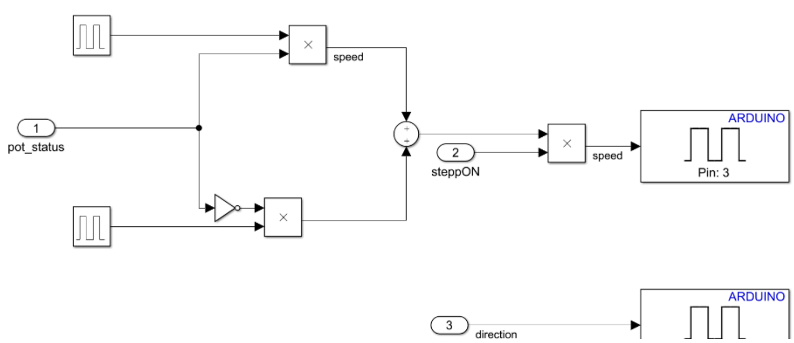


Figure 38: Stepper Motor Sub-block

6.4. Simulink State Blocks:

Our model consists of four states:

- Stop.
- Start.
- Forward Motion.
- Backward Motion.
- Gripping.

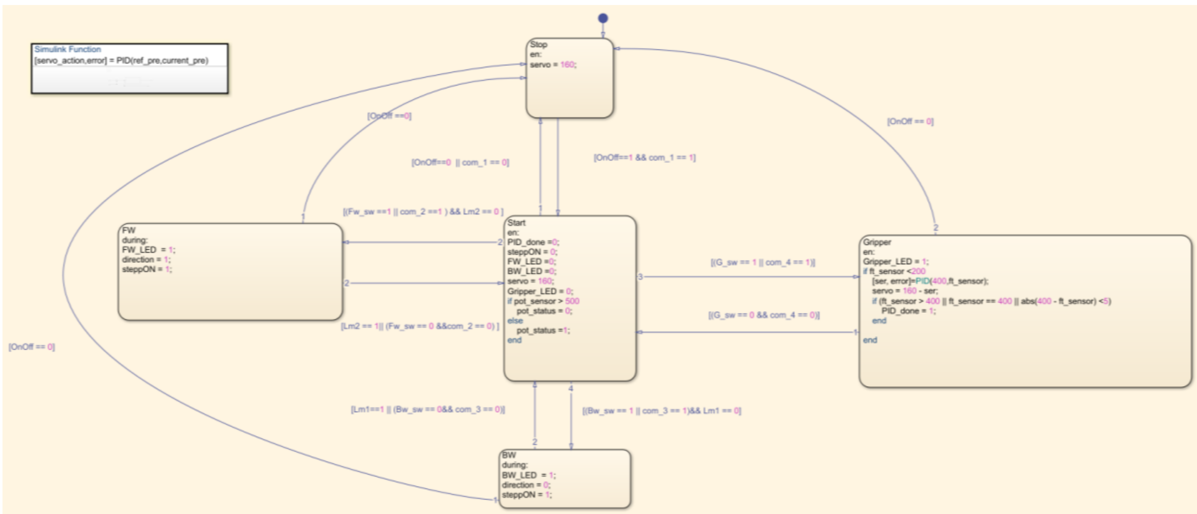


Figure 39: State Flow

6.4.1. Stop state:

This state is the idle state it only initializes the servo and waits for the on_switch.

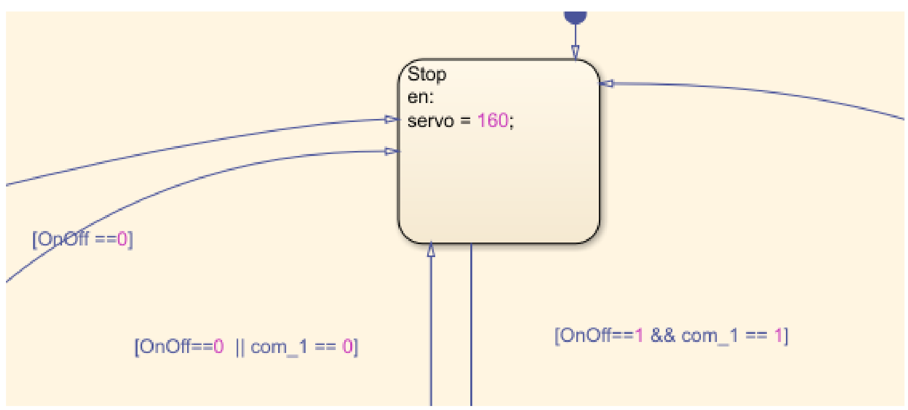


Figure 40: Stop State

The system moves to the next state when both on_switch and Bluetooth communication is established.

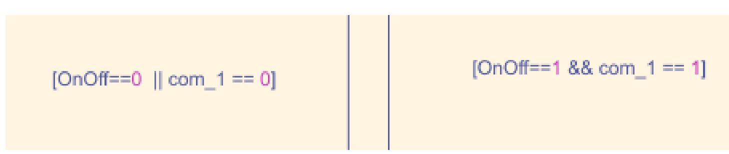


Figure 41: Transition from stop to start

6.4.2. Start state:

This state:

- Initialize all variables
- Reset LEDs
- Reads Potentiometer.

The system moves to the next state when one of the three switches is pressed and returns when it's released or when Limit Switches are reached.

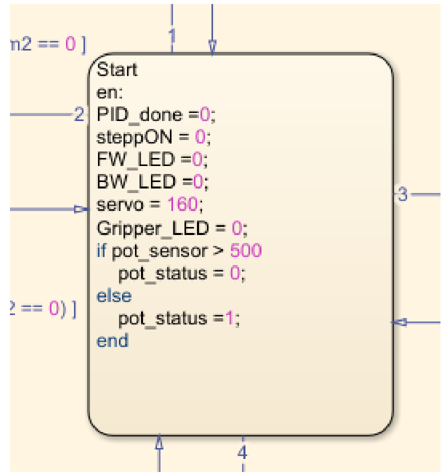


Figure 42: Start State

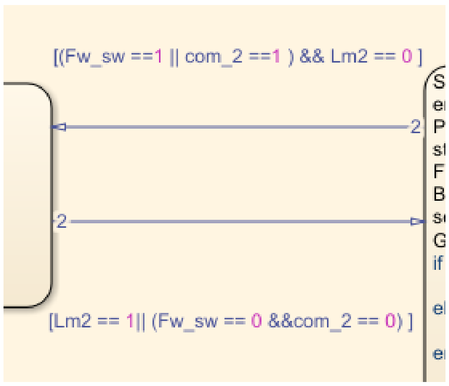


Figure 43: Transition from start to fw

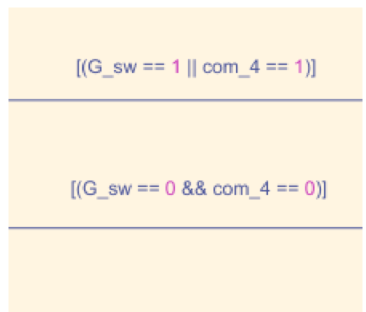


Figure 44: Transition from start to gripping

6.4.3. Forward State:

This state:

- Initialize FW LED
- Actuate Stepper motor in forward direction.

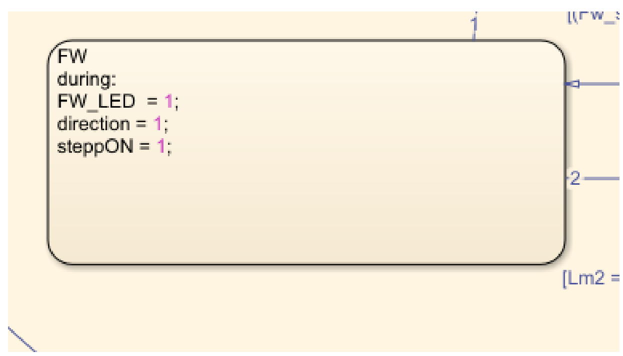


Figure 45: Forward State

6.4.4. Backward State

This state:

- Initialize BW LED
- Actuate Stepper motor in backward direction.

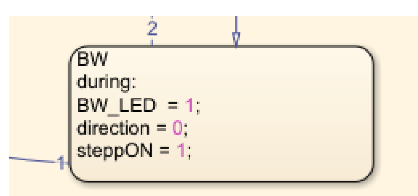


Figure 46: BW State

6.4.5. Gripping State:

This state:

- Initialize Gripping LED
- Actuate servo motor by calling PID Simulink function.

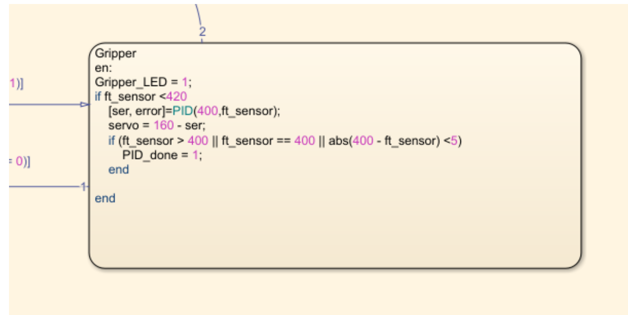


Figure 48: Gripping State

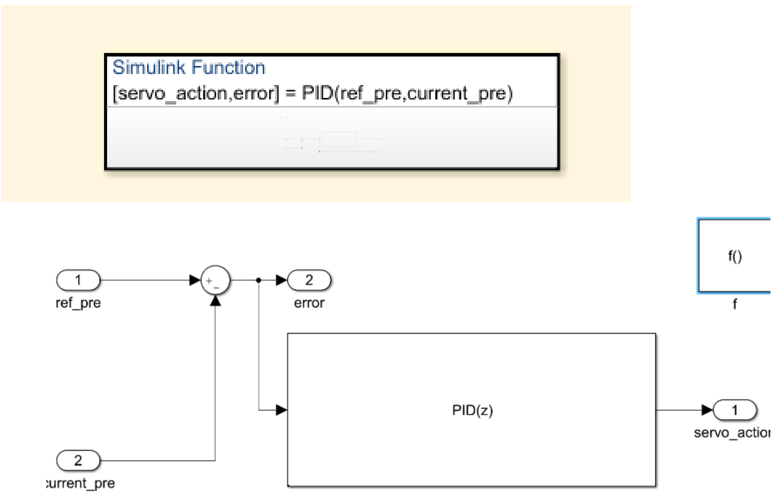


Figure 50: PID Function

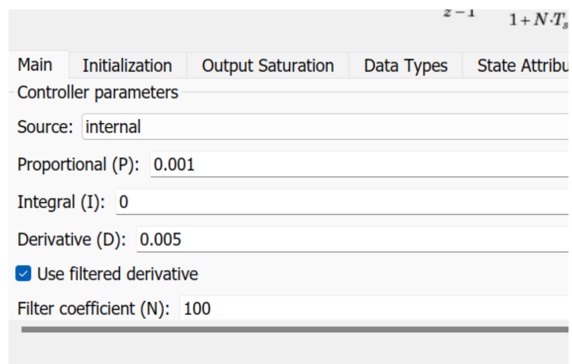


Figure 49: PID Gains

7. Problems Faced and How We Managed to Solve It:

7.1. Limit switches Noise:

During the input readings, the data was found to have high noise which was affecting the data processing. Our answer was using the LPF with some modifications

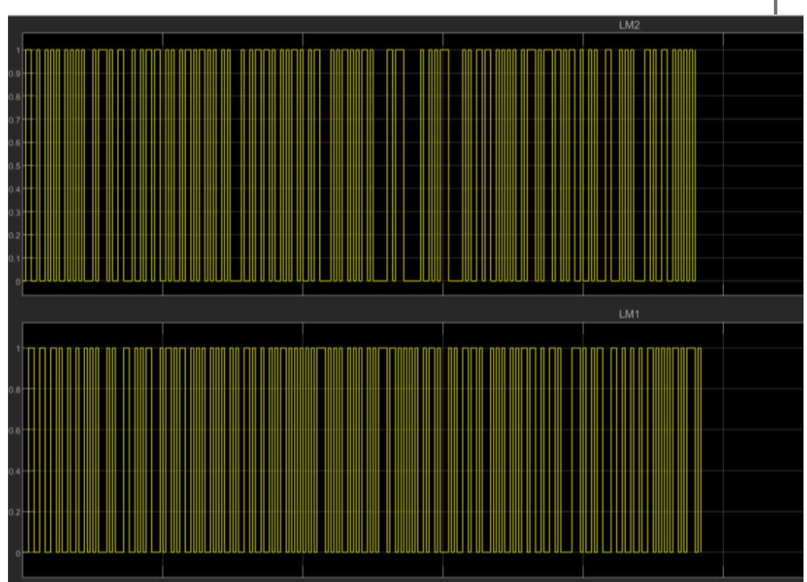


Figure 51: Limit Switch Noise

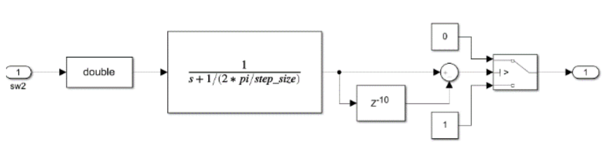


Figure 52: Designed LPF

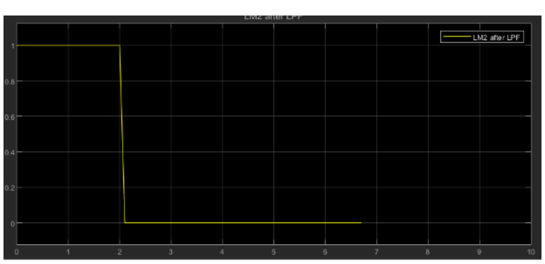


Figure 53: Output

7.2. Stepper Motor Actuation:

One pin of the stepper motor is actuated only using PWM, so the proposed solution was using switch between this Pulse Generator. However, switch inputs should be of same pulse width. This was solved as:

1. In case pot_status = 1: the upper branch with have value and lower one will be zero.
2. In case pot_status = 0: the lower branch with have value and upper one will be zero.

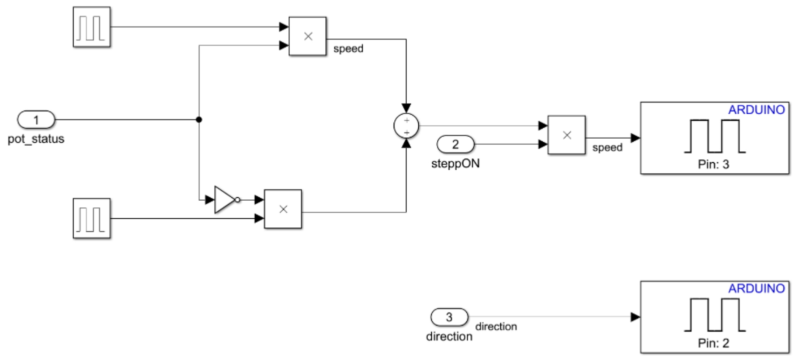


Figure 54: Switch Solution

8. Links:

8.1. Video:

https://drive.google.com/folderview?id=1L0iJrhx-nc_TFKU6miRPu-MTuPNgy04D

8.2. GitHub Repo:

https://github.com/hendaafia/State-flow-model-of-a-gripping-mechanism-using-MATLAB