

This Report is Written in L^AT_EX

Syed Affan Ahmed

*Department of Avionics Engineering
College of Aeronautical Engineering, NUST
Islamabad, Pakistan*

Abstract—3d printers are widely being used in many applications. One of the simplest mechanisms is a simply supported beam with moving bed. In this mechanism independent joint control is implemented where each motor is controlled independently by its own control mechanism. The aim of this report is to mathematically model the transfer function of motors along with its gear and load. Afterwards, applying an control strategy to control each x, y and z motors by individual control feedback loop. Finally reading reference data from pseudo .stl file to perform 3d printing in MATLAB.

Index Terms—3d printer, independent joint control, feedback loop, mathematical modeling, MATLAB, simulink

I. INTRODUCTION

Now a days 3d printers are commonly being used in different applications. There are different types of 3d printers based on their mechanism. Simply supported beam with moving bed is one of such configurations which works on independent joint control mechanism. In independent joint control the control of each axis of the manipulator is controlled as a single input/single-output (SISO) system. Any coupling effects due to the motion of the other links is treated as a disturbance. In the above described 3d printer, it consists of the following mechanism.

Printer bed: A flat surface on which the objects get printed.

Extruder: The part that heats up and dispenses material used for printing the 3D object.

Motor X : Moves the printer bed in the x direction. The motor drives the printer bed through a gearing and a ball-screw/belt mechanism. The mechanism and printer bed acts as a load for this motor.

Motor Y: Moves the extruder in the y direction. The motor drives the extruder through a gearing and a ball-screw/belt mechanism. The mechanism and extruder acts as a load for this motor.

Motor Z: Moves the printer structures in the z direction. The motor moves the structure through a gearing and a ball-screw/belt mechanism. The structure and mechanism act as a load for this motor.

II. MATHEMATICAL MODELING

In order to derive the mathematical model of the motor one must know the dynamics of motor. Block diagram of a single axis DC Motor drive system is shown in Fig. 1. As shown, motor consists of three parts, electrical part, motor side mechanical part and load side mechanical part.

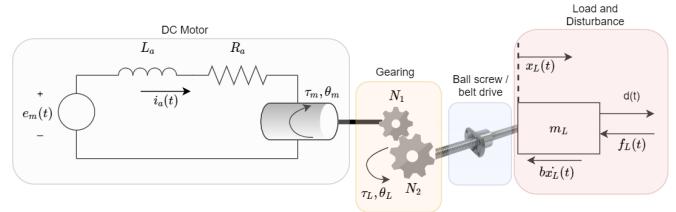


Fig. 1: Block diagram DC Motor drive system

A. Electrical part of motor

The differential equation for armature current in DC motor circuit schematic is then given by

$$L \frac{di_a}{dt} + Ri_a = e_m(t) - V_b \quad (1)$$

where V_b is given as

$$V_b = K_b \frac{d\theta_m}{dt} \quad (2)$$

Put eq(2) in eq(1)

$$L \frac{di_a}{dt} + Ri_a = e_m(t) - K_b \frac{d\theta_m}{dt} \quad (3)$$

$$\frac{di_a}{dt} + \frac{R}{L} i_a = \frac{e_m(t)}{L} - \frac{K_b}{L} \frac{d\theta_m}{dt} \quad (4)$$

As,

$$\theta_m = \frac{1}{K_R} x_m$$

and

$$x_m = \frac{N_2}{N_1} x_L$$

So,

$$\theta_m = \frac{1}{K_R} \frac{N_2}{N_1} x_L \quad (5)$$

Thus, eq(4) can be written as follows after replacing the value of θ_m

$$L \frac{di_a}{dt} + Ri_a = e_m(t) - \frac{K_b}{K_R} \frac{N_2}{N_1} \frac{dx_L}{dt} \quad (6)$$

Taking Laplace Transform of eq(6)

$$(Ls + R)I_a(s) = E_m(s) - \frac{K_b}{K_R} \frac{N_2}{N_1} s X_L(s) \quad (7)$$

B. Motor side mechanical part of motor

The equation of motion for motor side mechanical part is given by

$$J_{mot} \frac{d^2\theta_m}{dt^2} + D_1 \frac{d\theta_m}{dt} = \tau_m \quad (8)$$

where τ_m is given by

$$\tau_m = K_t i_a$$

$$J_{mot} \frac{d^2\theta_m}{dt^2} + D_1 \frac{d\theta_m}{dt} = K_t i_a \quad (9)$$

Substituting eq(5) in eq(9)

$$J_{mot} \frac{1}{K_R} \frac{N_2}{N_1} \frac{d^2x_L}{dt^2} + D_1 \frac{1}{K_R} \frac{N_2}{N_1} \frac{dx_L}{dt} = K_t i_a \quad (10)$$

Taking Laplace Transform of eq(10)

$$\left(J_{mot} \frac{1}{K_R} \frac{N_2}{N_1} s^2 + D_1 \frac{1}{K_R} \frac{N_2}{N_1} s \right) X_L(s) = K_t I_a \quad (11)$$

C. Load side mechanical part of motor

The equation of motion for load side mechanical part is given by

$$J_{load} \frac{d^2\theta_L}{dt^2} + D_2 \frac{d\theta_L}{dt} = \tau_L \quad (12)$$

where

$$\tau_L = K_s f_L(t) \quad (13)$$

from load and disturbance model of Fig. 1.

$$f_L(t) = m_L \frac{d^2x_L}{dt^2} + b \frac{dx_L}{dt} + d(t) \quad (14)$$

Thus, eq(12) will become

$$J_{load} \frac{d^2\theta_L}{dt^2} + D_2 \frac{d\theta_L}{dt} = K_s \left(m_L \frac{d^2x_L}{dt^2} + b \frac{dx_L}{dt} + d(t) \right) \quad (15)$$

Now, substituting $\theta_L = \frac{1}{K_R} x_L$

$$\frac{J_{load}}{K_R} \frac{d^2x_L}{dt^2} + \frac{D_2}{K_R} \frac{dx_L}{dt} = K_s \left(m_L \frac{d^2x_L}{dt^2} + b \frac{dx_L}{dt} + d(t) \right) \quad (16)$$

Taking Laplace transform of eq(16)

$$\left[\left(\frac{J_{load}}{K_R} - K_s m_L \right) s^2 + \left(\frac{D_2}{K_R} - K_s b \right) s \right] X_L(s) = K_s D(s) \quad (17)$$

D. Solving equations simultaneously using MATLAB

Solving eq(7), eq(11) and eq(17) simultaneously using MATLAB symbolic tool box. Finding the transfer function

$$\frac{X_L(s)}{E_m(s)}$$

by keeping $D(s) = 0$

```
clc
clear all
%--Solving for X_L/Em by setting D=0
syms s Em b D1 D2 Jmot Jlod Kb Kt Kr Ks La
    ↪ Ra mL N1 N2 Ia XL r
D=0;
eq2= (((Jmot/Kr)*(r)*s^2+(D1/Kr)*(r)*s))/
    ↪ XL == Kt*Ia
eq3= (((Jlod/Kr)-Ks*mL)*s^2+((D2/Kr)*(Ks*b
    ↪ )*s))/XL == Ks*D
temp=eq2-eq3
Ia=solve(temp,Ia)
eq1= (La*s+Ra)*Ia == Em - (Kb/Kr)*(r)*s/XL
vars=[XL Em]
[E_m, X_L]=solve(eq1,vars)
out=X_L/E_m
out=simplify(out,1000)
pretty(out)
```

Now solving for

$$\frac{X_L(s)}{D(s)}$$

by keeping $E_m(s) = 0$

```
%%
%--Solving for X_L/Em by setting Em=0
syms s D b D1 D2 Jmot Jlod Kb Kt Kr Ks La
    ↪ Ra mL N1 N2 Ia XL r
Em=0;
eq1= (La*s+Ra)*Ia == Em - (Kb/Kr)*(r)*s/XL
Ia=solve(eq1,Ia)
eq2= (((Jmot/Kr)*(r)*s^2+(D1/Kr)*(r)*s))/
    ↪ XL == Kt*Ia
eq3= (((Jlod/Kr)-Ks*mL)*s^2+((D2/Kr)*(Ks*b
    ↪ )*s))/XL == Ks*D
temp=eq2-eq3
vars=[XL D]
[Dis, X_L]=solve(temp,vars)
out2=X_L/Dis
out2=simplify(out2,1000);
pretty(out2)
```

Thus the resulting transfer function $G_1(s) = \frac{X_L(s)}{E_m(s)}$ after further manual simplification is given in Fig. I.
And the resulting transfer function $G_2(s) = \frac{X_L(s)}{D(s)}$ after further manual simplification is given in Fig. II.

$$\frac{X_L(s)}{E_m(s)} = \frac{K_R K_t}{s \left[\left(L_a s + R_a \right) \left(\left(\frac{N_2}{N_1} J_{mot} - J_{lod} + K_s K_R m_L \right) s + \left(\frac{N_2}{N_1} D_1 - D_2 + K_s K_R b \right) \right) + \frac{N_2}{N_1} K_t K_b \right]}$$

Fig. I: Simplified Transfer function $G_1(s)$

$$\frac{X_L(s)}{E_m(s)} = \frac{-K_R K_s (L_a s + R_a)}{s \left[\left(L_a s + R_a \right) \left(\left(\frac{N_2}{N_1} J_{mot} - J_{lod} + K_s K_R m_L \right) s + \left(\frac{N_2}{N_1} D_1 - D_2 + K_s K_R b \right) \right) + \frac{N_2}{N_1} K_t K_b \right]}$$

Fig. II: Simplified Transfer function $G_2(s)$

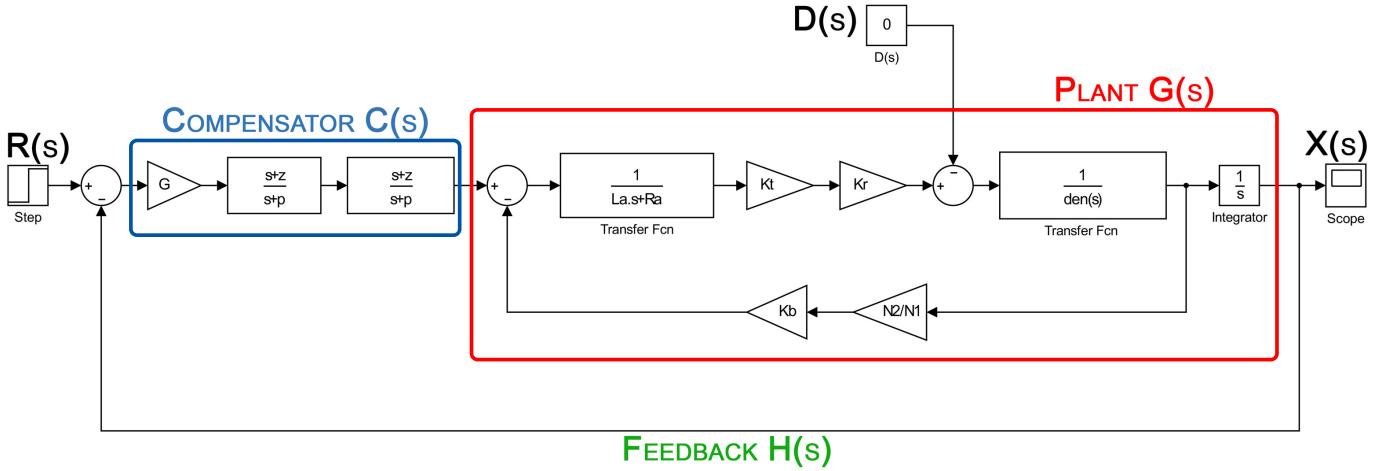


Fig. 2: Block diagram of DC Motor drive system

III. FEEDBACK CONTROL SYSTEMS

Designing of individual feedback back control system is done using root locus design tool. The same configuration shown in Fig. 2 is used for X, Y and Z motors. All the controllers are tuned to make sure all the motors have similar transient response which has a settling time of less than 1 second and an overshoot of less than 5%. The design process of each motor controller is discussed below.

A. Y motor controller design

The open loop poles of the system are:

- 0
- -2.4472
- -0.0530

The first pole at 0 making the system marginally stable and is the most dominant pole. The root locus of the system is shown in Fig. 3. Thus, when the value of gain increases, it move the poles to the unstable region. The aim is to reduce the settling time and increase the damping. Technically shifting the poles to left is also desirable i.e. moving the marginally stable pole to the stable region. Lead compensator via root locus design has been used because it increases

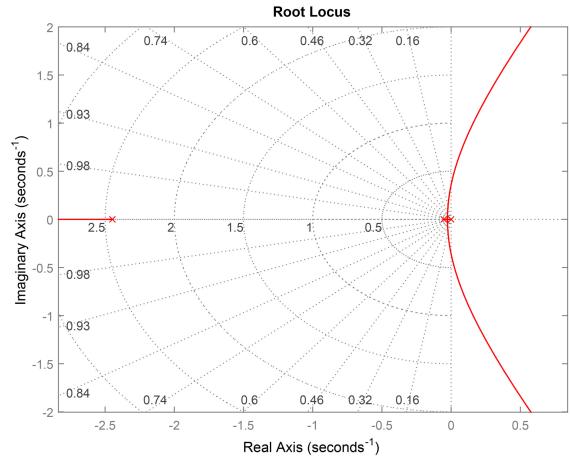


Fig. 3: Zoomed root locus of Y Motor without controller

relative stability by increasing phase margin. For a given system gain, lead compensator increase steady state error. To decrease steady state error, large compensator gain must be used. The additional effect of this it decreases the step response

settling time, i.e. increasing the system damping. Moreover, lead compensator also shifts the poles towards left half of the Real-Imaginary axis. First order Lead compensator doesn't meet the required specifications because the system has very less gain. Therefore, 2nd order Lead compensator is used i.e cascading two lead compensator. The tuned parameters through root locus design of the controller is shown in Fig. 4. The resulting root locus after tuning the compensator is

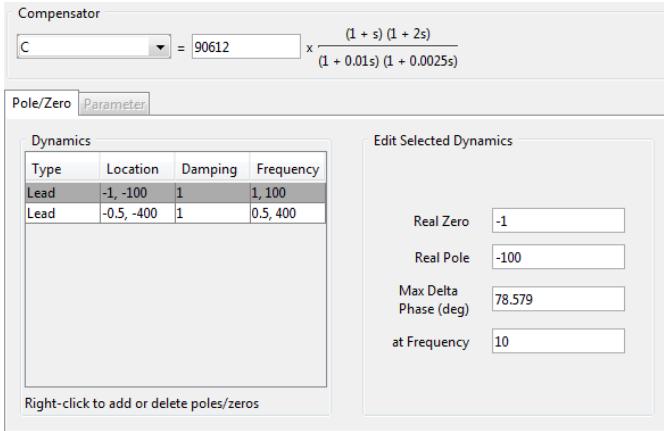


Fig. 4: Lead compensator tuned parameters for Y motor

shown in Fig. 5. In which pole zero cancellation removed the effect of dominant poles near to the origin resulting in large gain margin that helped in achieving the desired response. The

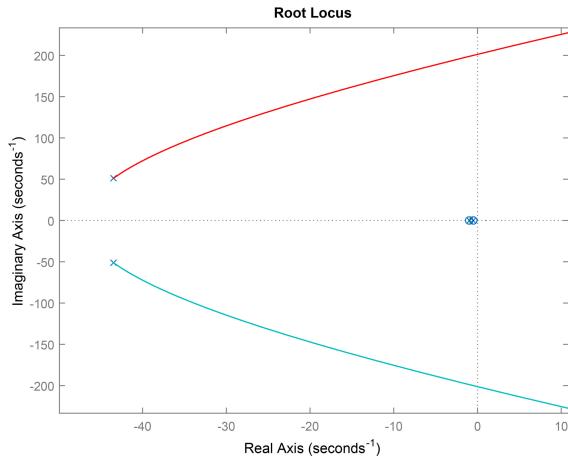


Fig. 5: Zoomed root locus of Y Motor with controller

step response of closed loop Y motor system is shown as Fig. 6.

B. Y motor controller design

The open loop poles of the X motor plant are:

- 0
- -2.3791
- -0.1210

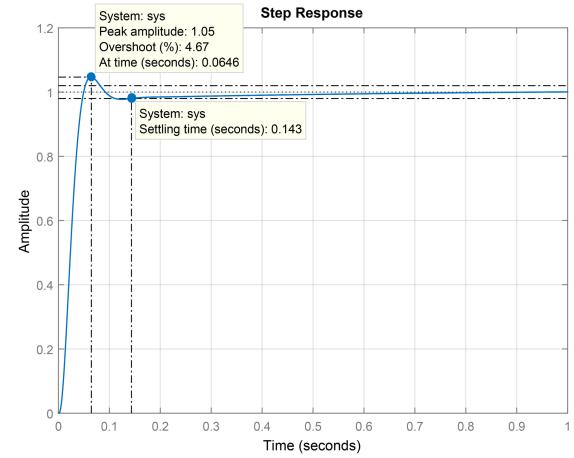


Fig. 6: Step response of Y Motor after feedback control

The first pole at 0 making the system marginally stable and is the most dominant pole. The root locus of the system is shown in Fig. 7. This case is also similar to Y motor. Thus, applying

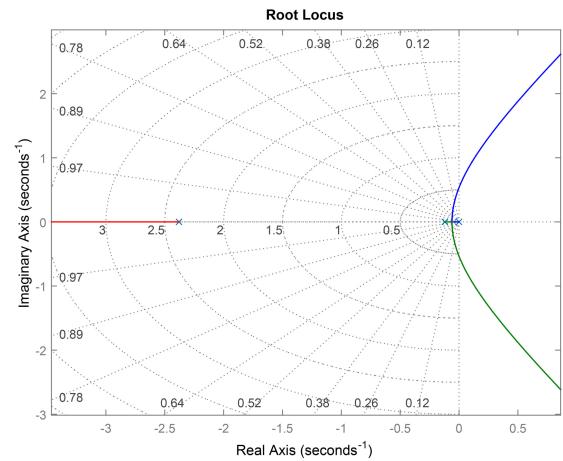


Fig. 7: Zoomed root locus of X Motor without controller

2nd order Lead compensator is used i.e cascading two lead compensator. The tuned parameters through root locus design of the controller is shown in Fig. 8. The resulting root locus after tuning the compensator is shown in Fig. 9. In which pole zero cancellation removed the effect of dominant poles near to the origin resulting in large gain margin that helped in achieving the desired response. The step response of closed loop X motor system is shown as Fig. 10.

C. Z motor controller design

The open loop poles of the Z motor plant are:

- 0
- -2.0170
- -0.4830

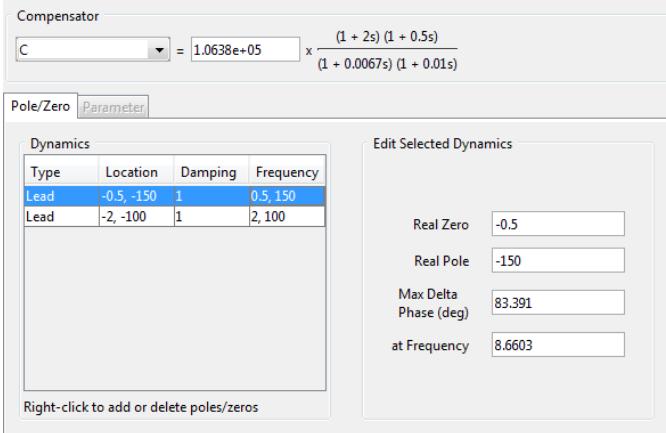


Fig. 8: Lead compensator tuned parameters for X motor

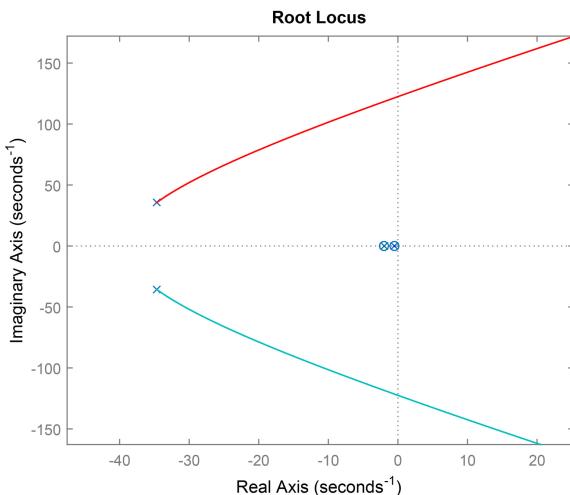


Fig. 9: Zoomed root locus of X Motor with controller

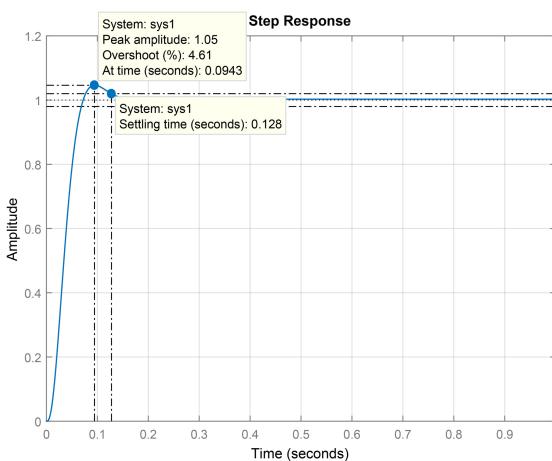


Fig. 10: Step response of X Motor after feedback control

The first pole at 0 making the system marginally stable and is the most dominant pole. The root locus of the system is shown in Fig. 11. This case is also similar to X and Y motor. Thus,

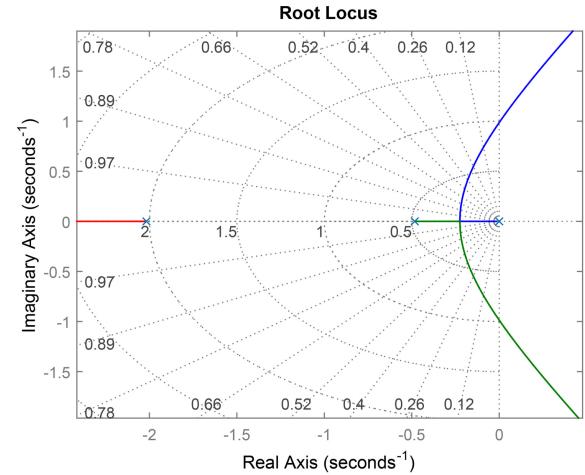


Fig. 11: Zoomed root locus of Z Motor without controller

applying 2nd order Lead compensator is used i.e cascading two lead compensator. The tuned parameters through root locus design of the controller is shown in Fig. 12. The resulting

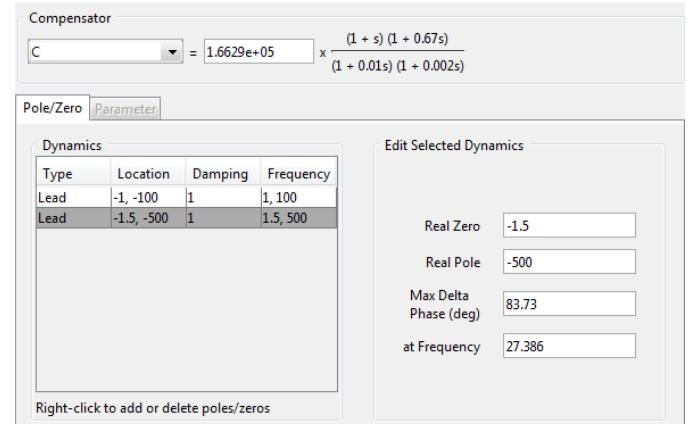


Fig. 12: Lead compensator tuned parameters for Z motor

root locus after tuning the compensator is shown in Fig. 13. In which pole zero cancellation removed the effect of dominant poles near to the origin resulting in large gain margin that helped in achieving the desired response. The step response of closed loop Z motor system is shown as Fig. 14.

D. Square wave response on controlled system in Simulink

Applying continuous step input i.e. square waves using pulse generator having period of 3 seconds and pulse width of 50%. The output response to the square wave is shown in Fig. 15.

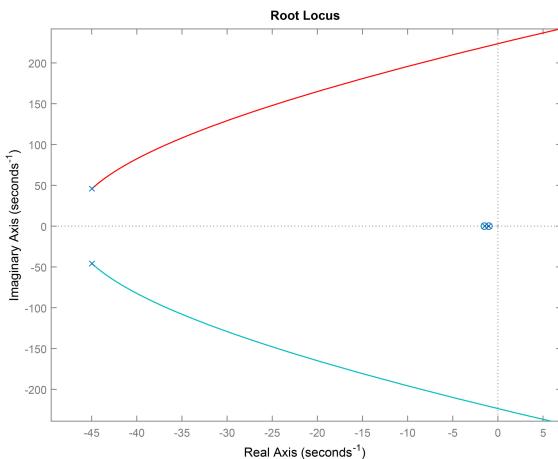


Fig. 13: Zoomed root locus of Z Motor with controller

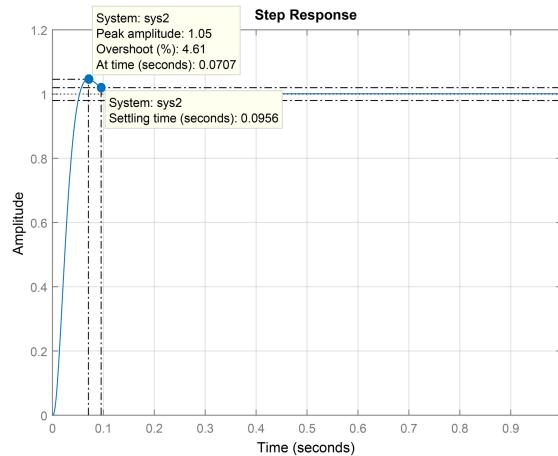


Fig. 14: Step response of Z Motor after feedback control

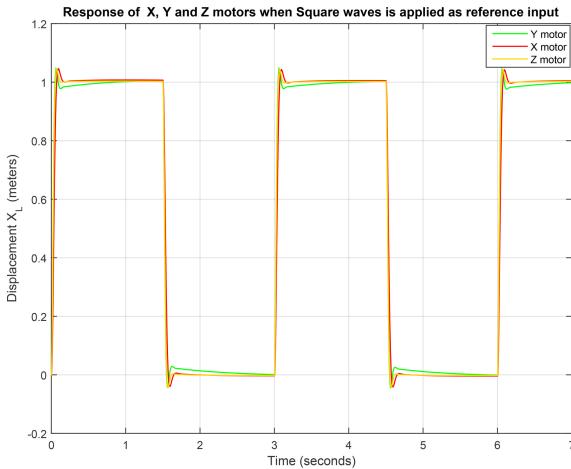


Fig. 15: Response of X, Y and Z motors when Square waves is applied as reference input

E. Effect of constant disturbance on output

Applying constant disturbance of 5N does not effect the output response of the system as shown in Fig. 16, which means the feedback controller of every motor can compensate the effect of the applied disturbance.

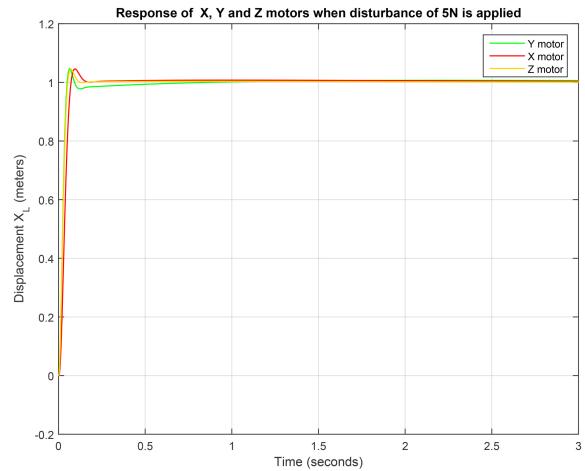


Fig. 16: Response of X, Y and Z motors when disturbance of 5N is applied

Moreover, a comparative plot of step response with and without disturbance is shown in Fig. 17. But a constant force of more than 100N results in slight deviation from the reference input as shown in Fig. 18 which is zoomed to 300%.

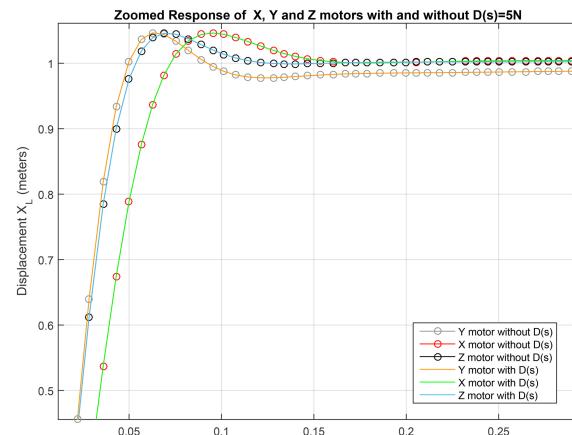


Fig. 17: Zoomed Response of X, Y and Z motors with and without disturbance

IV. 3D PRINTING WITHOUT DISTURBANCE

The following MATLAB code has been used to read data from pseudo .stl file and saving as .mat file for further use.

```
dataset=xlsread('pseudo_stl_file_data.xlsx'
    ↵,'CakeRaw'); % read excel file
```

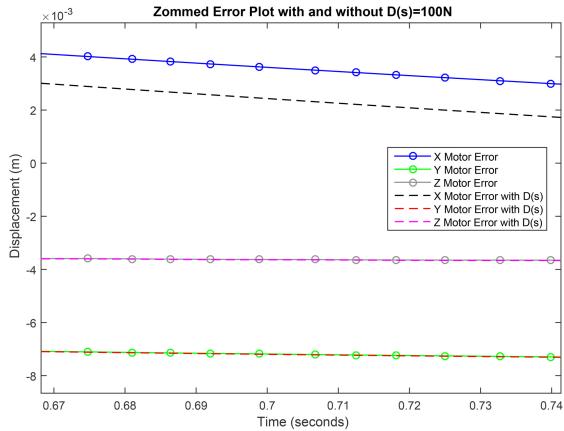


Fig. 18: 300% zoomed response of X, Y and Z motors with and without $D(s)=100N$

```
x=dataset(1:155,2)%specify rows and
    ↪ columns to read from the file
y=dataset(1:155,3)
z=dataset(1:155,4)
dispense=dataset(:,5)
stop=dataset(:,6)
incr=dataset(1:155,7)

save ('read_cakeRaw.mat', 'x','y','z',
    ↪ incr') %saving variables in .mat
    ↪ file for further use
```

Similar code as above has been used to read all the spread sheets from excel file and save it as .mat file.

Following code has been used to simulate the model in Simulink by giving input from MATLAB script and taking output back into MATLAB script and plot it. Error tolerance hasn't been catered because the error was already below 0.05 and unsynchronization of Simulink time with MATLAB function code loops. Moreover, the input to the simulink model is exactly after 1 second as incr in .stl is treated as time axis, which is greater than the settling time, therefore, the output printed object is doesn't show any large variation.

```
load read_cylinder.mat %load .mat file

simin_x=[incr,x]; %passing data to
    ↪ Workspace as Timeseries, where incr
    ↪ is time and x are coordinates for X
    ↪ Motor
simin_y=[incr,y];
simin_z=[incr,z];

sim('temp2',length(incr)); %running
    ↪ simulation temp2.mdl with simulation
    ↪ time equal to the time scale of .
    ↪ stl file
```

```
plot3(simout_x,simout_y,simout_z) %plot 3D
    ↪ on data obtained from ToWorkSpace
    ↪ block of simulink

title('Cylinder_with_constant_D(s)')
xlabel('x-axis_(meters)')
ylabel('y-axis_(meters)')
zlabel('z-axis_(meters)')
grid on

figure
plot(err_x) %plot of error from E(s) in
    ↪ feedback loop of Motor X
title('Motor_X_Error_Plot_with_D(s)')
ylabel('Displacement_(m)')
grid on

figure
plot (err_y) %plot of error from E(s) in
    ↪ feedback loop of Motor Y
title('Motor_Y_Error_Plot_with_D(s)')
ylabel('Displacement_(m)')
grid on

figure
plot (err_z) %plot of error from E(s) in
    ↪ feedback loop of Motor Z
ylabel('Displacement_(m)')
title('Motor_Z_Error_Plot_with_D(s)')
grid on
```

A. 3d printing of Rectangle

The 3d plot of Rectangle shown in Fig. 19 without any disturbance. The edges are slightly circular when the figure is

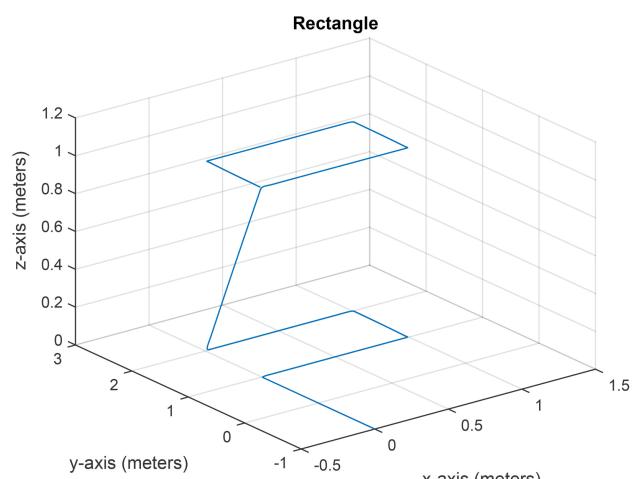


Fig. 19: Rectangle 3d plot without disturbance

zoomed because of error tolerance. The error plots of X, Y

and Z motors in case of rectangle in shown Fig. 20, Fig. 21 and Fig. 22 respectively.

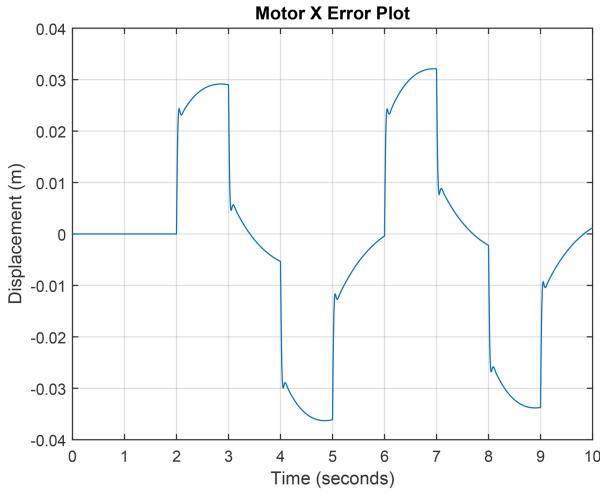


Fig. 20: X motor error plot while printing Rectangle

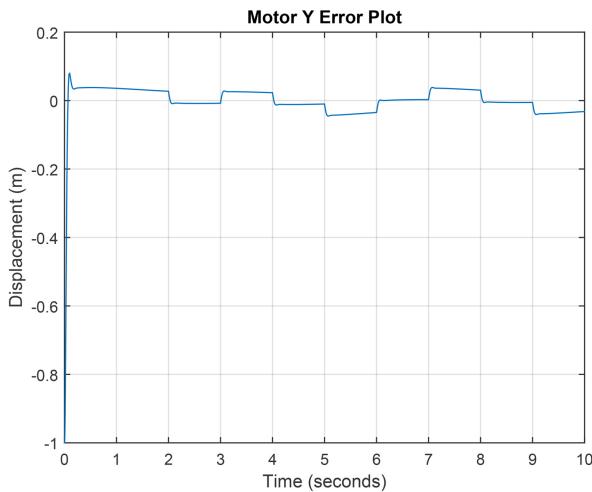


Fig. 21: Y motor error plot while printing Rectangle

B. 3d printing of Cylinder

The 3d plot of Cylinder shown in Fig. 23 without any disturbance. No error can be visualized in cylinder as it has no sharp edges for further elaboration the error plots of X, Y and Z motors are shown Fig. 24, Fig. 25 and Fig. 26 respectively.

C. 3d printing of Cube

The 3d plot of Cube shown in Fig. 27 without any disturbance. The edges of Cube is not sharp enough when the figure is zoomed because of error tolerance. The error plots of X, Y and Z motors in case of Cube in shown Fig. 28, Fig. 29 and Fig. 30 respectively.

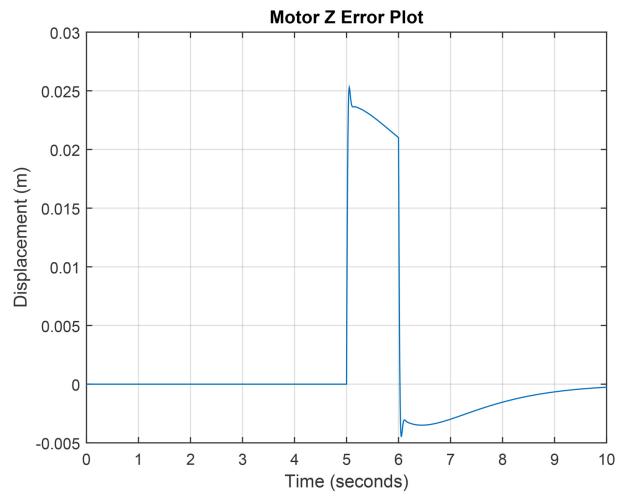


Fig. 22: Z motor error plot while printing Rectangle

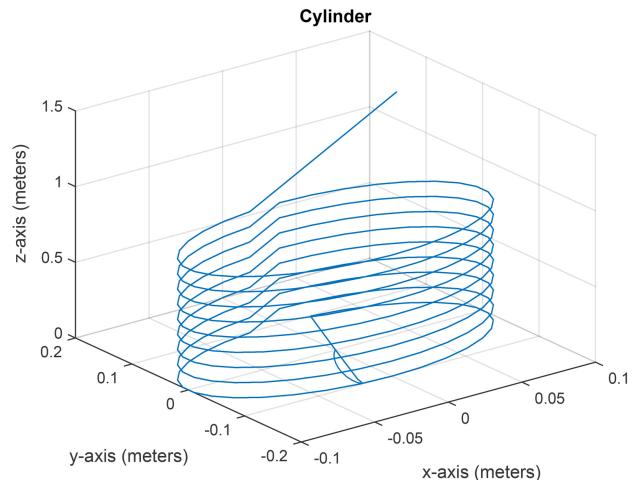


Fig. 23: Cylinder 3d plot without disturbance

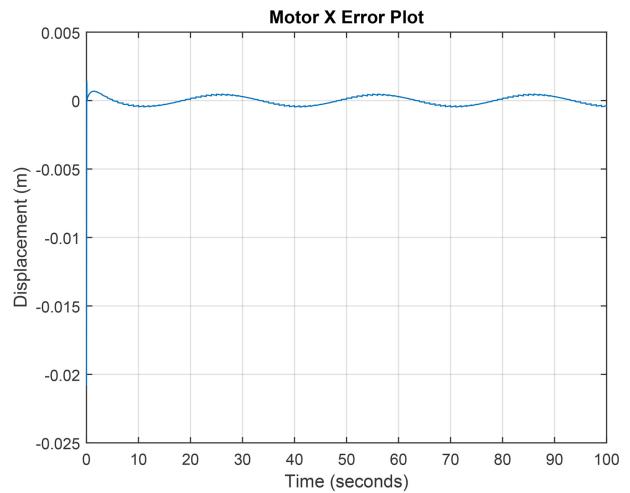


Fig. 24: X motor error plot while printing Cylinder

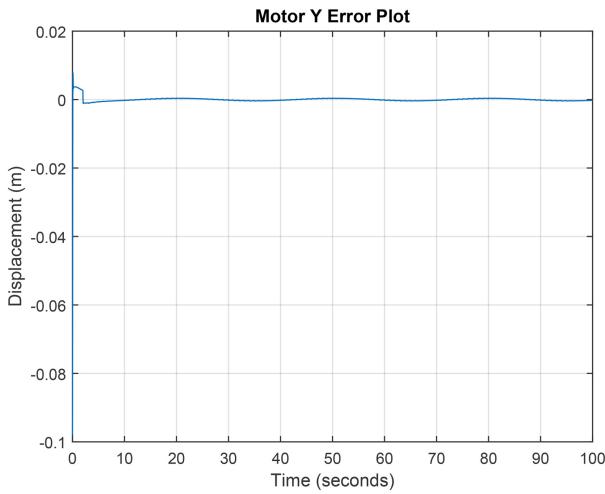


Fig. 25: Y motor error plot while printing Cylinder

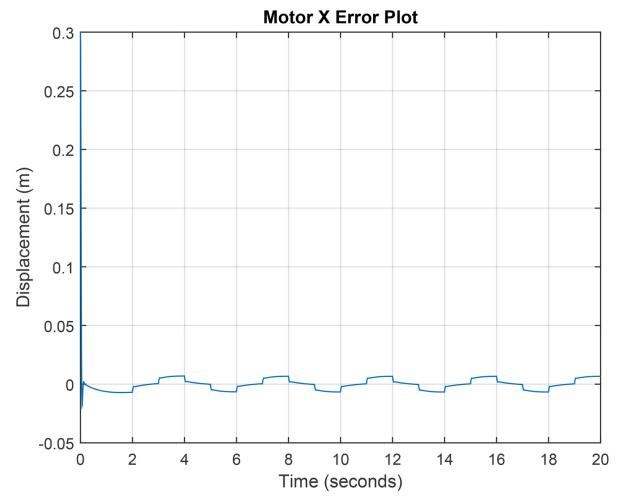


Fig. 28: X motor error plot while printing Cube

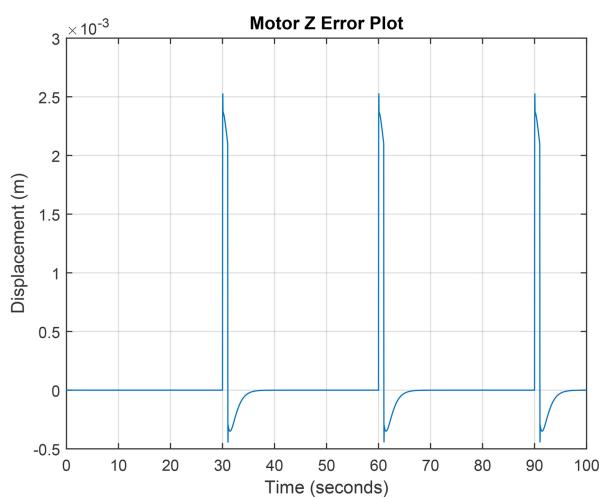


Fig. 26: Z motor error plot while printing Cylinder

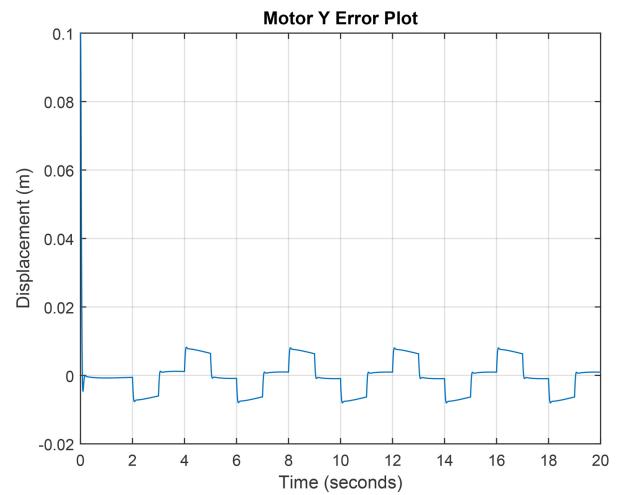


Fig. 29: Y motor error plot while printing Cube

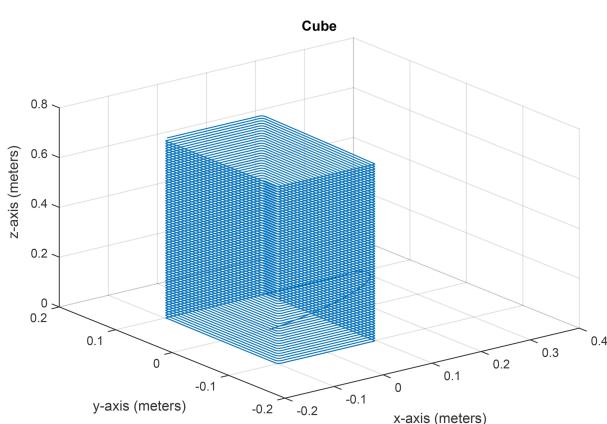


Fig. 27: Cube 3d plot without disturbance

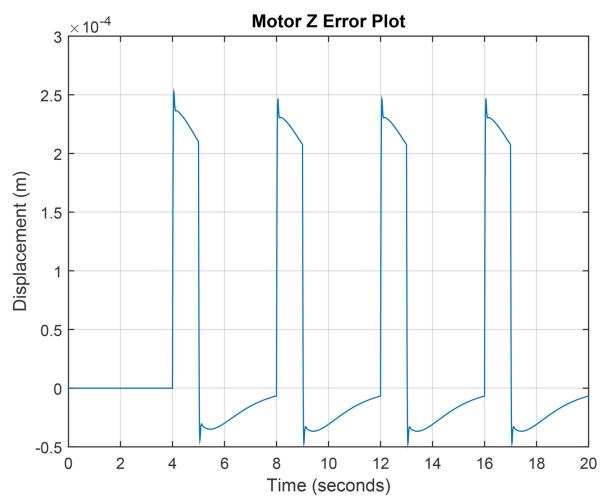


Fig. 30: Z motor error plot while printing Cube

D. 3d printing of Pyramid

The 3d plot of Pyramid shown in Fig. 31 without any disturbance. The edges of Pyramid is not sharp enough when

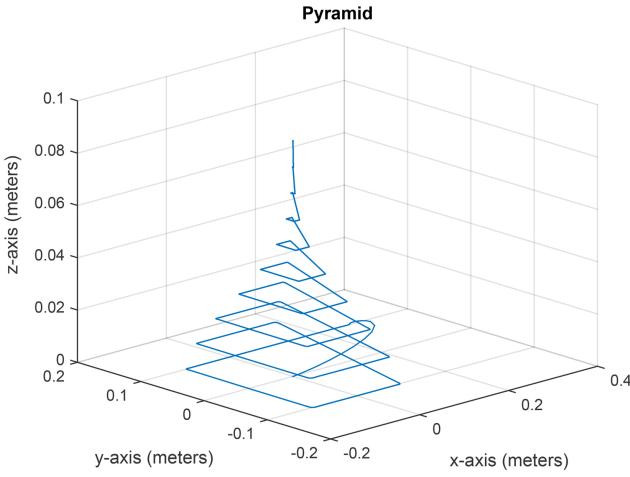


Fig. 31: Pyramid 3d plot without disturbance

the figure is zoomed because of error tolerance. The error plots of X, Y and Z motors in case of Pyramid in shown Fig. 32, Fig. 33 and Fig. 34 respectively.

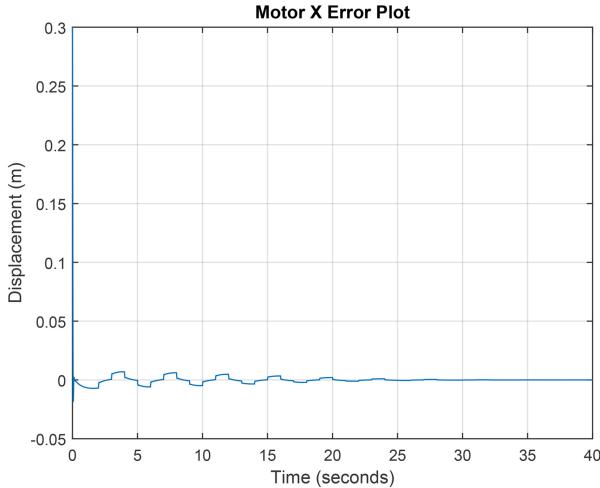


Fig. 32: X motor error plot while printing Pyramid

V. 3D PRINTING WITH DISTURBANCE OF 500N

A. 3d printing of Rectangle with $D(s)$

The 3d plot of Rectangle shown in Fig. 35 with a constant disturbance of 500N applied on each X, Y and Z motor. The edges are slightly circular when the figure is zoomed because of error tolerance. The error plots of X, Y and Z motors in case of rectangle in shown Fig. 36, Fig. 37 and Fig. 38 respectively. Although error increases with disturbance but was quiet small to produce any visual change in the printed object.

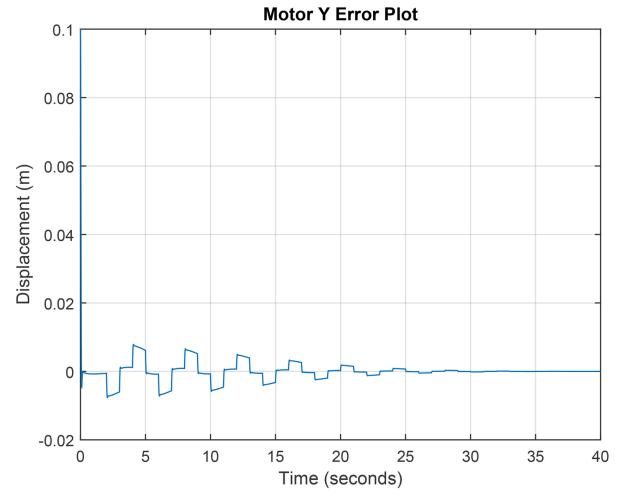


Fig. 33: Y motor error plot while printing Pyramid

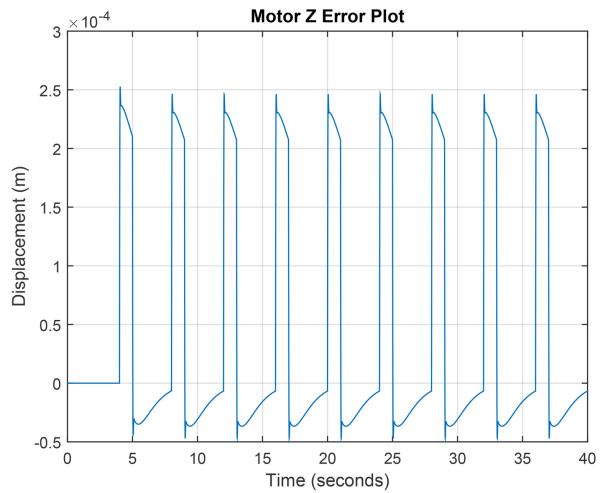


Fig. 34: Z motor error plot while printing Pyramid

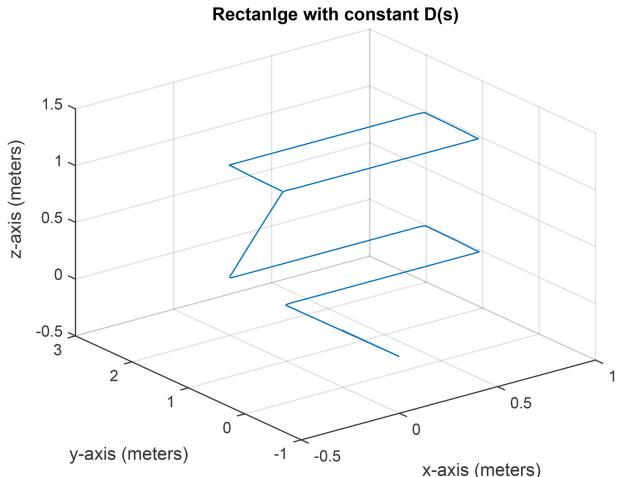


Fig. 35: Rectangle 3d plot with $D(s)=500N$

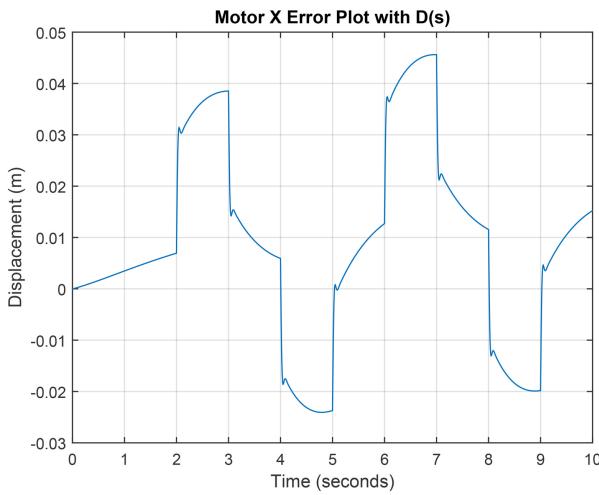


Fig. 36: X motor error plot in Rectangle printing with $D(s)$

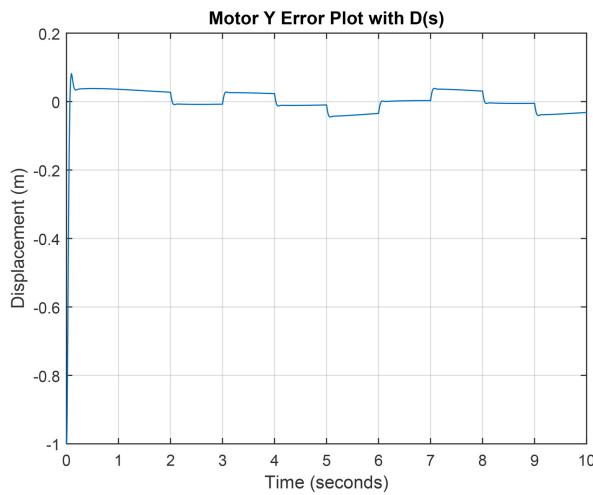


Fig. 37: Y motor error plot while printing Rectangle with $D(s)$

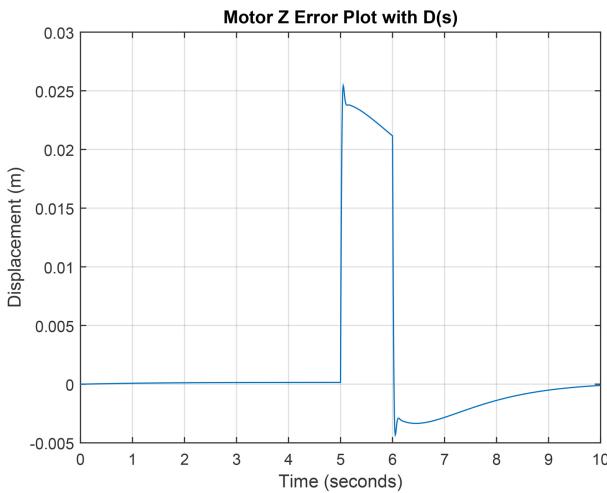


Fig. 38: Z motor error plot while printing Rectangle with $D(s)$

B. 3d printing of Cylinder with $D(s)$

The 3d plot of Cylinder shown in Fig. 39 without any disturbance. The change in layers of cylinder is not sharp

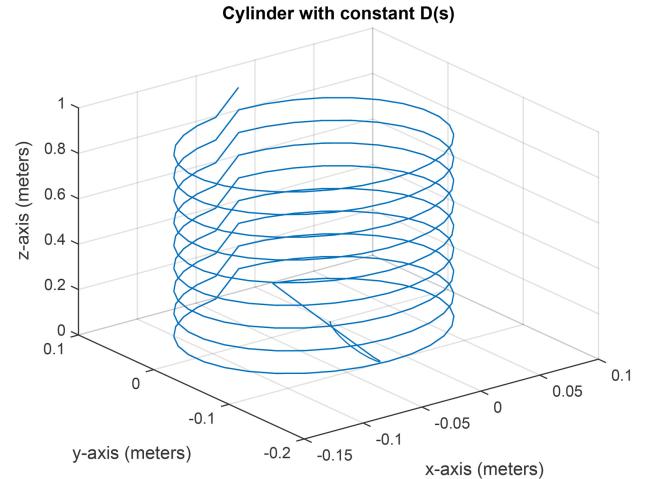


Fig. 39: Cylinder 3d plot with $D(s)=500N$

enough when the figure is zoomed because of error tolerance. The error plots of X, Y and Z motors in case of cylinder in shown Fig. 40, Fig.41 and Fig. 42 respectively. Although error increases with disturbance but was quiet small to produce any visual change in the printed object.

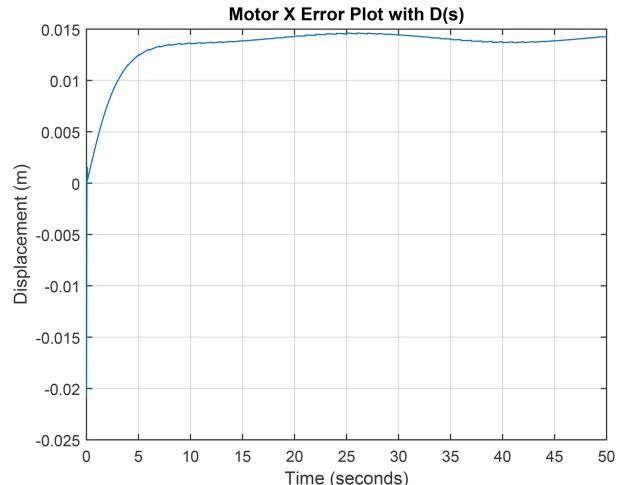


Fig. 40: X motor error plot while printing Cylinder with $D(s)$

C. 3d printing of Cube with $D(s)$

The 3d plot of Cube shown in Fig. 43 without any disturbance. The edges of Cube is not sharp enough when the figure is zoomed because of error tolerance. The error plots of X, Y and Z motors in case of Cube in shown Fig. 44, Fig. 45 and Fig. 46 respectively. Although error increases with disturbance but was quiet small to produce any visual change in the printed object.

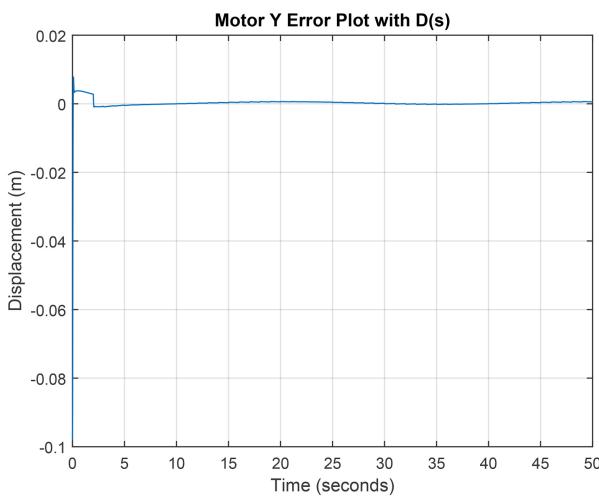


Fig. 41: Y motor error plot while printing Cylinder with D(s)

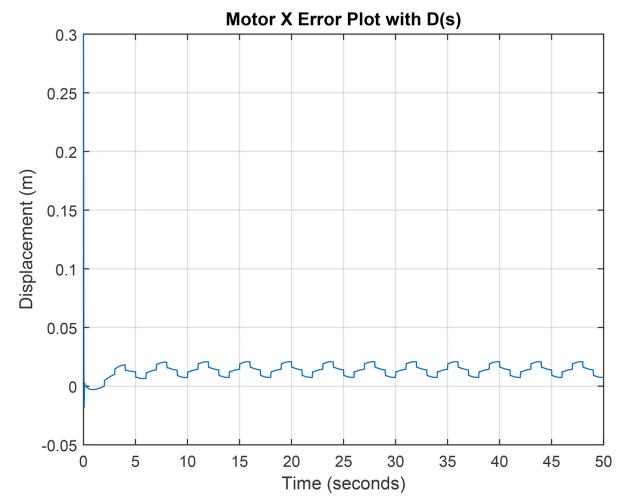


Fig. 44: X motor error plot while printing Cube with D(s)

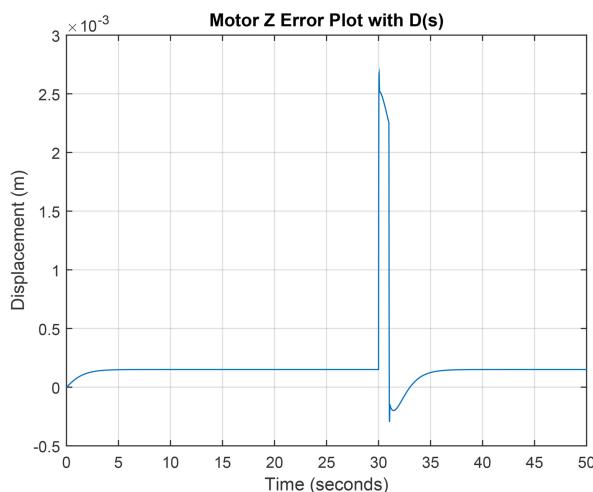


Fig. 42: Z motor error plot while printing Cylinder with D(s)

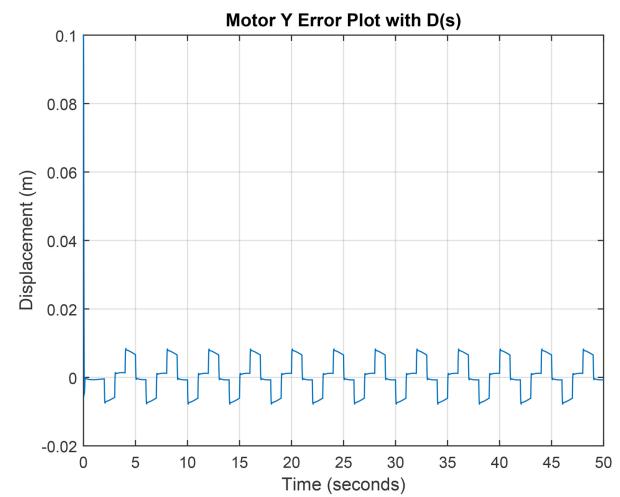


Fig. 45: Y motor error plot while printing Cube with D(s)

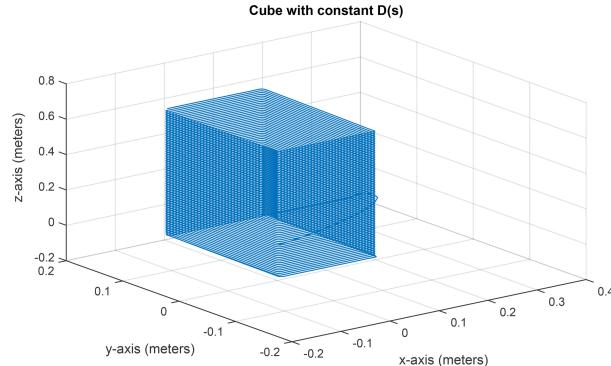


Fig. 43: Cube 3d plot without disturbance with D(s)=500N

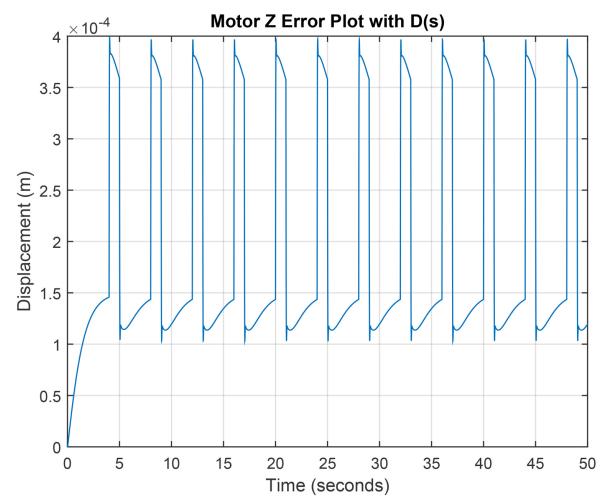


Fig. 46: Z motor error plot while printing Cube with D(s)

D. 3d printing of Pyramid with $D(s)$

The 3d plot of Pyramid shown in Fig. 47 without any disturbance. The edges of Pyramid is not sharp enough when

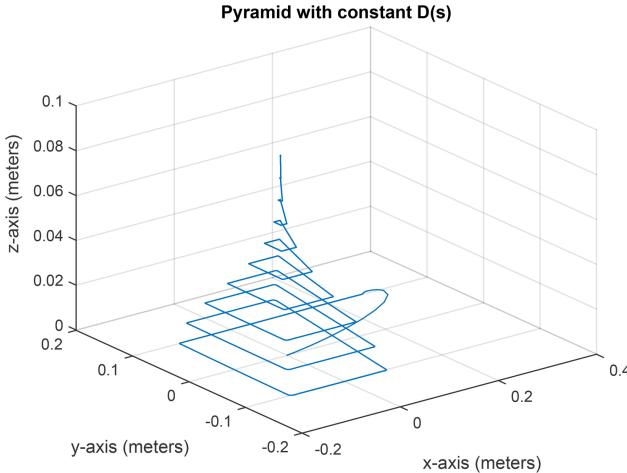


Fig. 47: Pyramid 3d plot without disturbance with $D(s)=500\text{N}$

the figure is zoomed because of error tolerance. The error plots of X, Y and Z motors in case of Pyramid in shown Fig. 48, Fig. 49 and Fig. 50 respectively. Although error increases with disturbance but was quiet small to produce any visual change in the printed object.

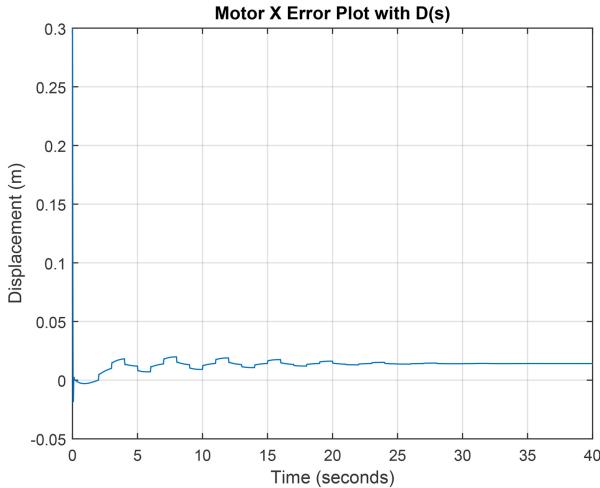


Fig. 48: X motor error plot while printing Pyramid with $D(s)$

E. Comparative error plots with and without $D(s)$

Comparative plots of error of motor X, Y and Z while printing **Cube** are shown in Fig. 51, Fig 52, Fig. 53. When the disturbance of 500N is applied , the error increases but unable to vary the settling time because of the designed compensator, therefore, slight variation doesn't effect the final printed object.

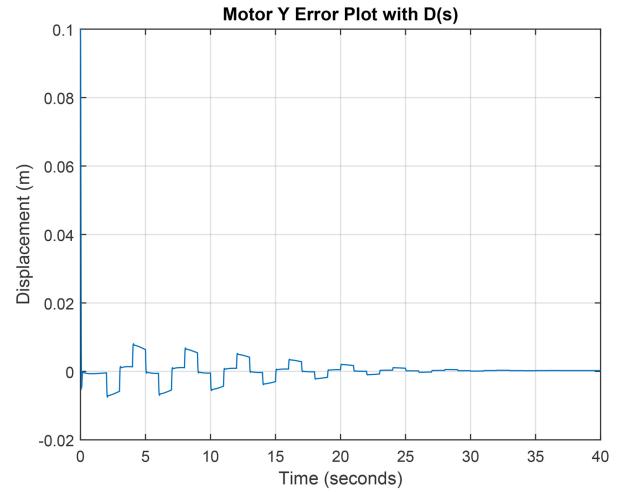


Fig. 49: Y motor error plot while printing Pyramid with $D(s)$

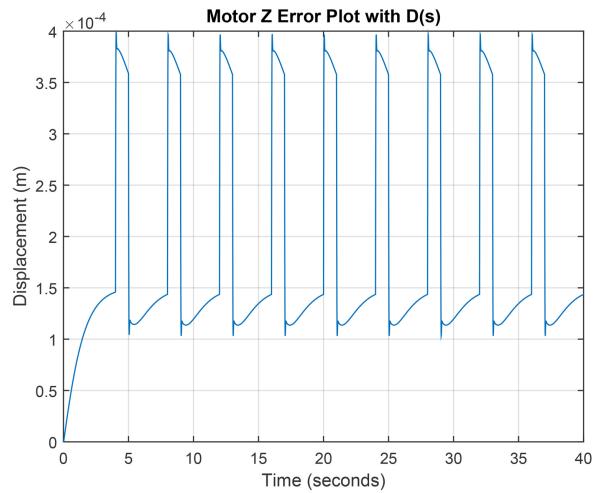


Fig. 50: Z motor error plot while printing Pyramid with $D(s)$

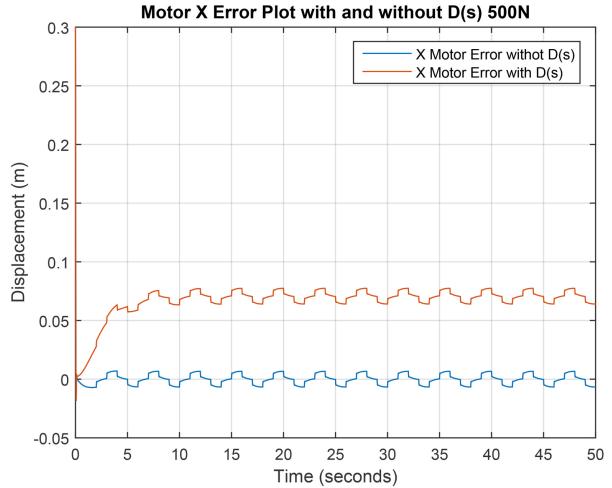


Fig. 51: X motor error plot with and without $D(s)$

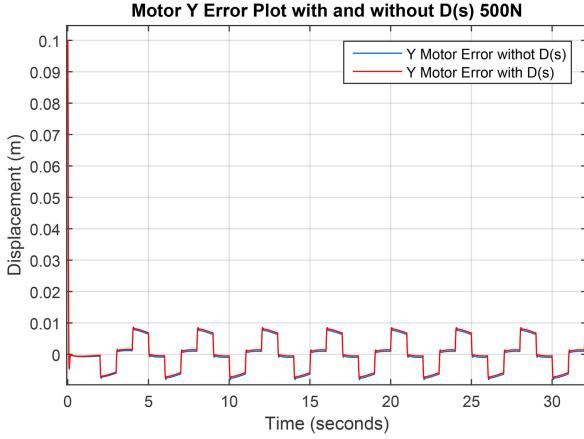


Fig. 52: Y motor error plot with and without D(s))

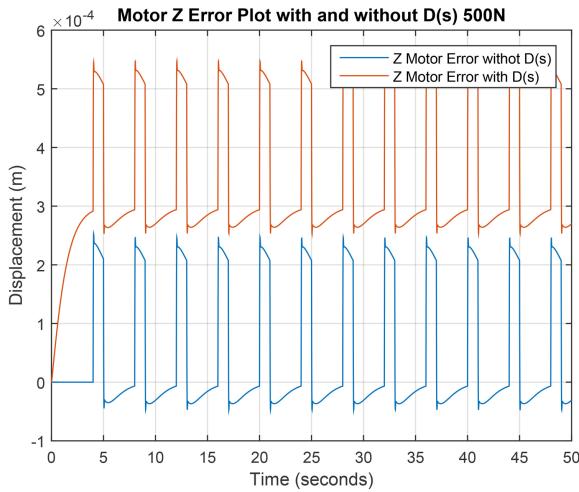


Fig. 53: Z motor motor error plot with and without D(s)

VI. CONCLUSION

The mathematical modeling of transfer function of X, Y and Z motors along with its gear and load has been performed. Then a Lead controller has been designed to control each X, Y and Z feedback loop individually. Simulation results proved that the controller can compensate a disturbance of upto 500N which is a quiet large value in terms of 3d printer. Moreover, 3d objects has been printed using designed 3d printer model in simulink by giving input coordinates from pseudo .stl file.

LIST OF SYMBOLS

- $e_m(t)$: motor input voltage
- $i_a(t)$: motor armature current
- L_a, R_a : motor armature inductance and resistance
- τ_m, θ_m : motor side torque and angular position
- $\frac{N_2}{N_1} = r$: gear ratio
- τ_L, θ_L : load side torque and angular position
- m_L : load mass
- $x_L(t)$: load displacement
- b : load friction

- $f_L(t)$: lead screw force
- J_m : moment of inertia (motor)
- J_p : moment of inertia (pinion connected to the motor)
- J_{mot} : $J_m + J_p$
- J_g : moment of inertia (big gear)
- J_s : moment of inertial (gear output shaft)
- J_{lod} : $J_g + J_s$
- K_s : lead screw power factor
- K_R : rotation to linear displacement conversion constant
- D_1 : motor friction
- D_2 : gear output shaft friction
- K_t : motor torque constant
- K_b : motor back emf constant

REFERENCES

- [1] Spong, Mark W., and Mathukumalli Vidyasagar. Robot dynamics and control. John Wiley and Sons, 2008.