NORTH SOUTH UNIVERSITY

Department of Electrical and Computer Engineering

Position Control of DC Servo Motor

Course : Control Engineering (EEE 342)
Presented to: Dr. Lamia Iftekhar

Presented by:

TEAM PINK PONIES

Tasnim Tamman Anika(17) 123 0197 643
Md. Hasib Zunair (32) 132 0262 643
Md. Jubayer Kabir(33) 132 0369 043

Table of Contents

1. Acknowledgement

We are very thankful to Allah, the most beneficent and merciful. We would also like to convey gratitude to our instructor, **Dr. Lamia Iftekhar** for giving us support and showing immense enthusiasm in our project. Lastly, we are thankful to our family and friends for their kindness and patience.

2. Abstract

Servo motors are different kind of machines. It moves to a position that is specific with respect to the given signal with the help of error sensing feedback. It is used in industrial applications for robot arms, prosthetic limbs, automatic door opener, antenna positioning and many more. This report is based on the position control of the motor, its step response, mathematical model as well as its transfer function and its characteristics. The final goal of this project would be to stabilize the system with minimum oscillation and negligible steady state error.

3. Introduction

Servo motor is a unique kind of dc motor which has a servomechanism. The specialty of this is that it provides angular precision they way we want. These motors are used in industrial equipments, control system of first starter of alternators. In the field of robotics it also has huge application from robotic arm manipulators to hobby airplane wing control and also for solar tracking. It consists or some major parts which together form the servo motor.

4. Physical Model

The servo motor consists of a dc motor, a potentiometer, a gear arrangement, and an intelligent circuitry. This circuitry is responsible for the positioning of the motor. Also this system is a Single input Single output system. For a given voltage signal we get a position in theta.

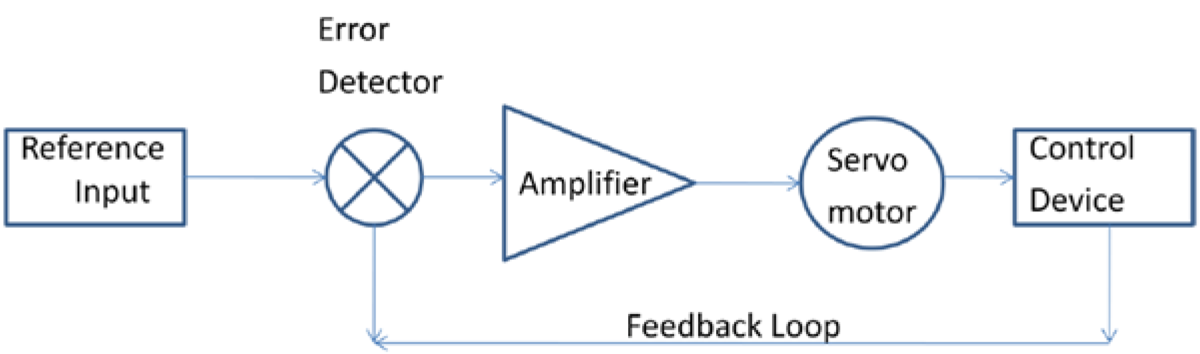

Fig 1: Typical Servo system block diagram

The above is a closed loop system. This is controlled by a feedback signal using the output of previous signal and input of current signal as reference. The given voltage is responsible for positioning of the servo motor

5. Notation And Parameters

r = Input angular position

c =Output angular position

e = Error signal

e_a = Armature voltage

e_v= Error voltage

$K_\circ$ = Proportionality constant

K_1 = Gain constant

K_2 = Motor torque constant

K_3 =Back-EMF constant

i_a = Armature current

R_a = Resistance

L_a = Inductor

T = Torque of the shaft

b = Friction co-efficient

θ = Angular Displacement

J = Rotational Momentum

The Physical parameters used here in this project are taken from:

http://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=SystemAnalysis

J=3.228E-6

b=3.5077E-6

K_1=0.0314

K_2 =0.0274

K_3=0.0356

R_a = 4

L_a = 2.75E-6

6. Mathematical Model and Transfer Function

The error signal in terms of voltage,

$$e_v = e_r - e_c \tag{1}$$

where e_r is the input signal and e_c is the output signal, and e_v is the error signal.

Relating the voltage and angular positions we get,

$$e_v = rK_0 \quad \text{and} \quad e_c = cK_0 \qquad (2) \,\&\, (3)$$

where K_0 is the proportionality constant.

Speed of rotation depends on armature voltage,

$$E_a = e_v\, K_1 \tag{4}$$

This is the output to the amplifier.

Next we know Torque is directly proportional to armature current, so:

$$T = K_2 i_a \tag{5}$$

K_2 is a constant as field flux is assumed to be constant, this is represented as motor torque constant.

When rotating, a voltage is proportional to the flux and angular velocity, keeping flux constant, we see voltage is proportional to angular velocity $\frac{d\theta}{dt}$

$$E_b = K_3 \frac{d\theta}{dt} \tag{6}$$

Using KVL we get,

$$L_a \frac{di_a}{dt} + R_a\, i_a + e_b = e_a \tag{7}$$

And from the torque equilibrium expression,

$$J \frac{d^2\theta}{dt^2} + b \frac{d\theta}{dt} = K_2\, i_a \tag{8}$$

Applying substitution techniques and using desired input and output ratio and applying Laplace transform we get our transfer function model:

$$H(s) = \frac{K_1\, K_2}{s(sL_a + R_a)(J_s + b + sK_2\, K_3)} \tag{9}$$

This is the transfer function model.

7. Open Loop System Analysis

For simulating the open loop system performance we use step input in MATLAB. And from poles and zeroes plot we get idea about stability.

Step Response

Our transfer function is name Q_V and when we apply 1V to the system we see that the position increases rapidly and it goes on. Hence our system is unstable.

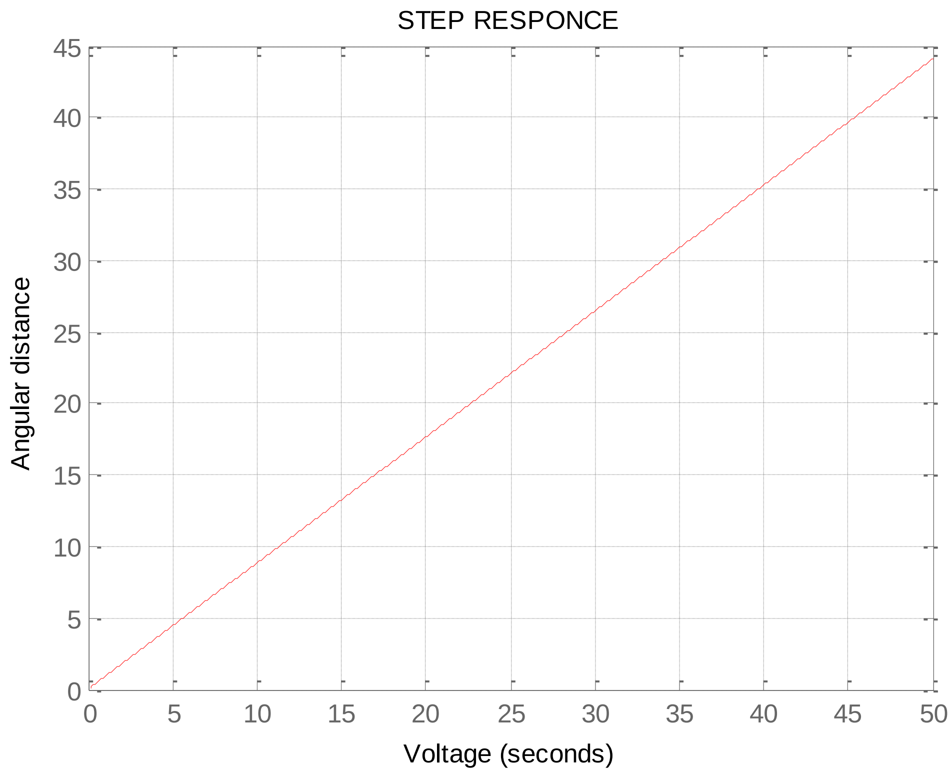

Figure 3: Step Response

Fig 2: Step response cure for open loop system

Pole-Zero Plots

From pole-zero plot we can identify whether the system is stable or not. Again using MATLAB we do pole zero plot. We have one pole on the left hand side of the s plane and one on the origin. Again concluding, the system is unstable.

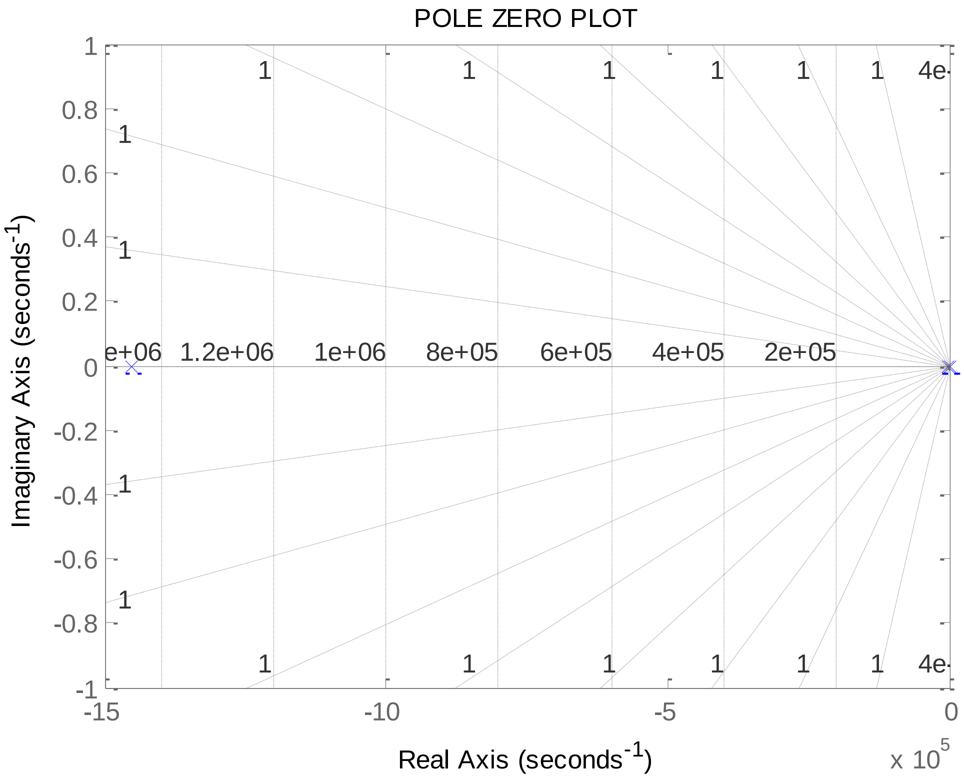

Figure 4 : Pole zero plot

8. Goals

Our goal is to stabilize the motor and its oscillation with minimum steady state error and overshoot. Therefore we want to use feedback system which would give good theta as output. And we also want our system to be efficient and cause no damage to surroundings.

9. Controller Design

Previously we saw that our system was highly unstable, for any bounded input we had an unbounded output – meaning the output would increase exponentially. To fix this, we are going to apply a controller to our system as shown in the diagram below. This will make our system a closed loop system.

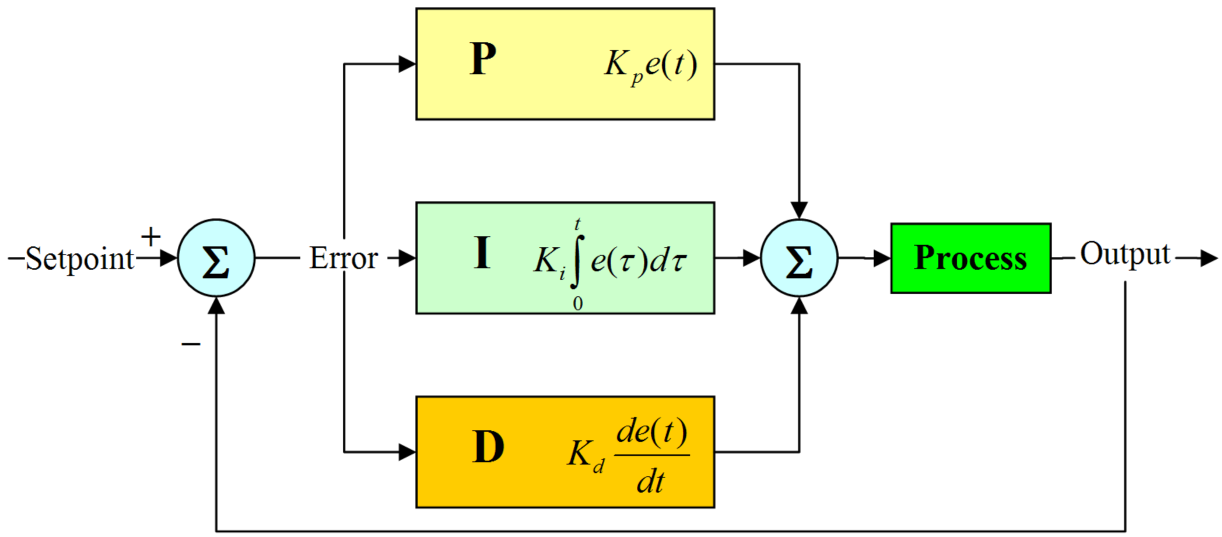

Figure 5 : Closed loop system

10. Transfer function for closed loop system

Now we will experiment will different values of the PID controller to see which combination of the constants suits our need.

i) Using P Controller

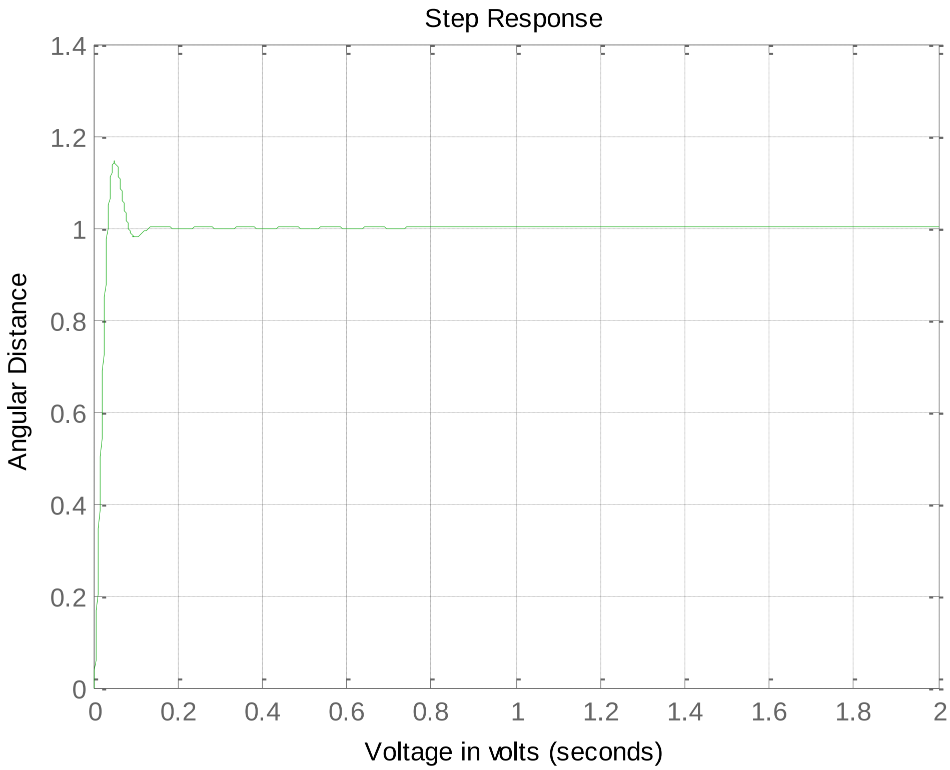

Figure 6: Using P controller

Here using a kp value of 80 , keeping others zero, we see that our sysem becomes stable but we have and overshoot.

ii) Using I controller

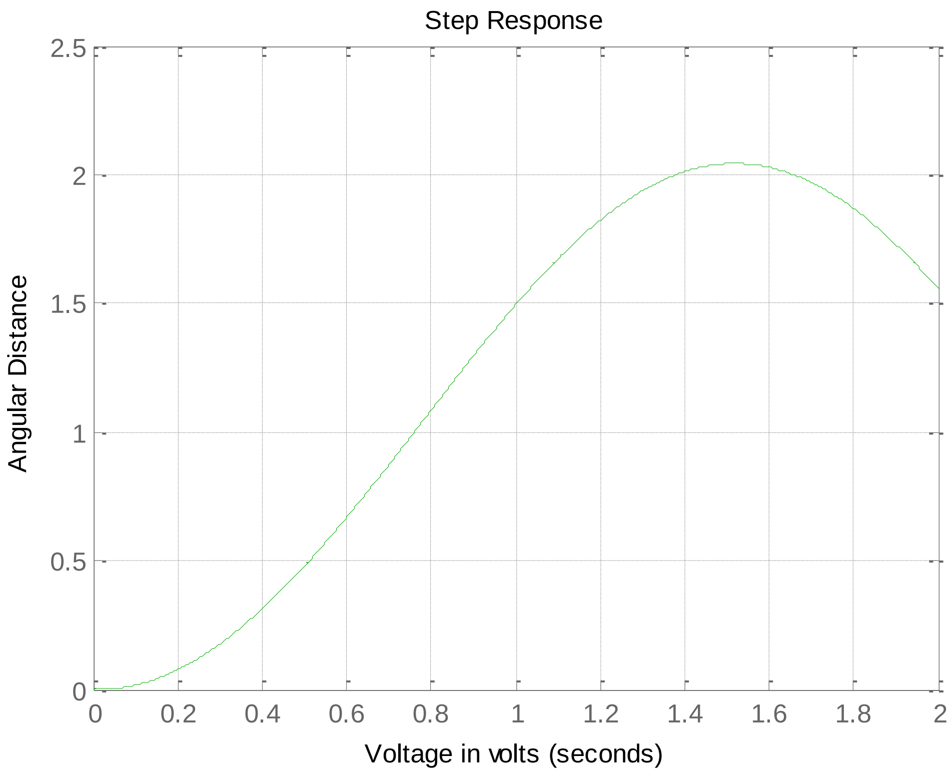

Figure 7: Using I controller

Using the I controller with ki of 5 we see that our system oscillates vigorously. We want zero oscillation in our system to prevent damage of the motor so we fully ignore the I controller.

iii) Using the D controller

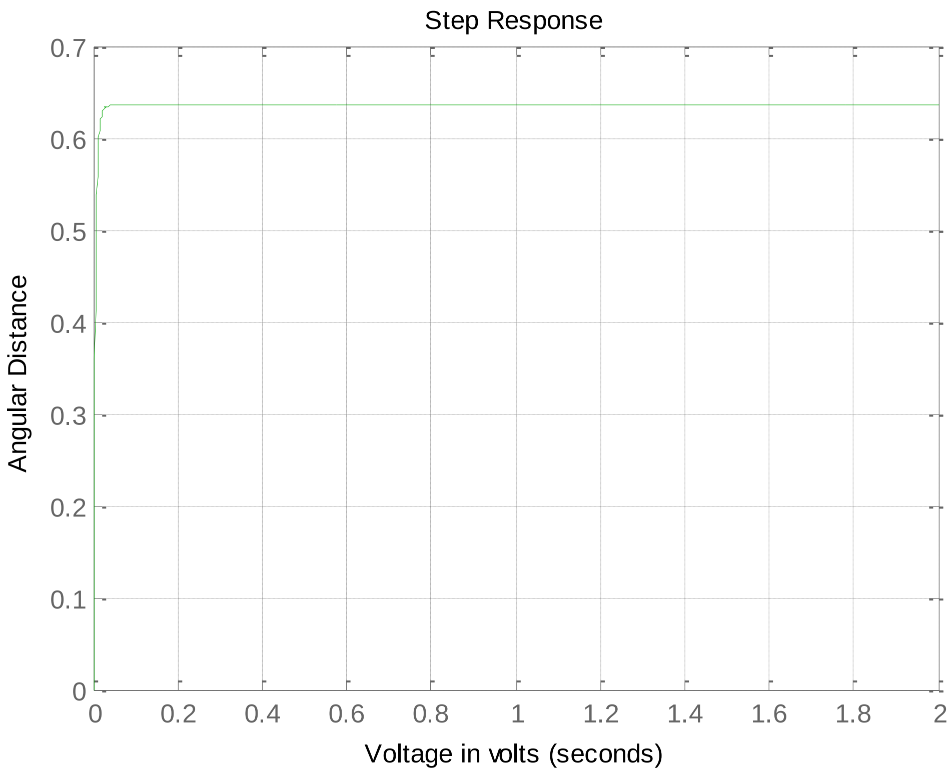

Figure 8: Using D controller

Using the D controller with kd of 2 we see that our system reaches stability at a much higher rate but it has a steady state error. We do want steady state error so we have to minimize it.

iv) Using the PID controller

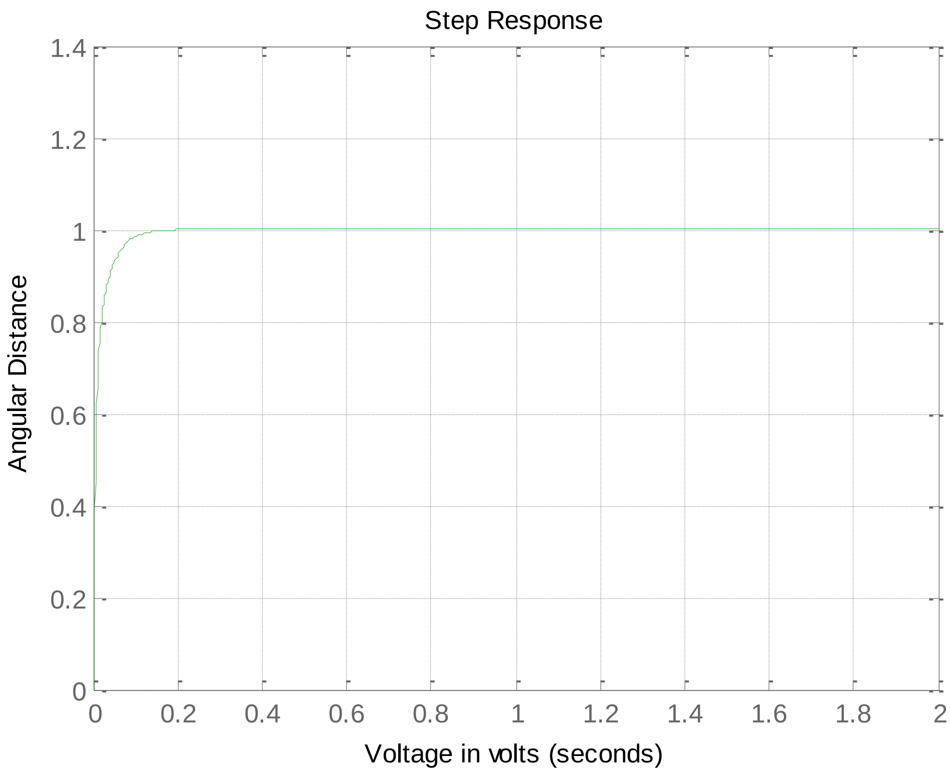

Figure 9: Using PID controller

Using kp 80 ki 5 kd 2 we see that this is the graph that we want to see in our output. We have zero ss-error, no oscillation, and system is fast. Doing some more experiments….

v) Using the PD controller

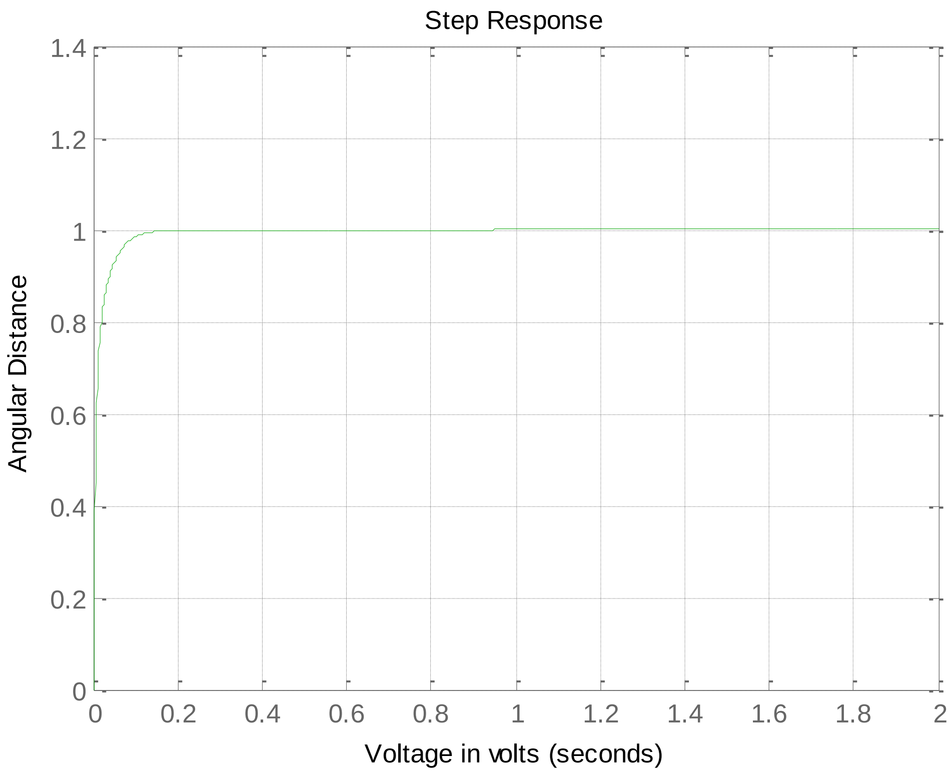

Figure 10: Using PD controller

Using the PD controller we can see that our system is much faster than with the PID controller and also it has no steady state error and absolutely no oscillation.

11. Limitation and Fault

Initially , when modeling the transfer function we assumed that the electric and magnetic fields in the motor are somewhat constant. This was done for easy manipulation otherwise the system is much more complex in general. Also, we could have used higher value of the constants kp kd to make much faster, but then there would have been a trade off between cost and efficiency.

12.Conclusion and Application

Here we conclude that our system is stable and fast after using the PD controller which has value of kp = 80 , ki = 0 and kd = 2. We had similar values using the PID controller but avoided it as PD is much more simple when applied practically.

Lastly, our final system will not just be limited to Matlab. It has practical applications. Here in Bangladesh our system can be applied to the wing control off aircrafts. It can also be applied in industrial level were there is use of timing belts and precise positioning. This will be relatively cheap without compromising stability.

12.Appendix

MATLAB Code for step response

```
%parameters
J=3.228E-6;
b=3.5077E-6;
K1=0.0314;
K2=0.0274;
K3=0.0356;
R=4;
L=2.75E-6;

%step resp
t=0:0.001:50;
s=tf('s');
Q_V=K1*K2/(s*((L*s+R)*(J*s+b)+K2*K3));
step(Q_V,t,'r');
grid on
title('STEP RESPONCE')

Q_V=K1*K2/(s*((L*s+R)*(J*s+b)+K2*K3));
pzmap(Q_V)
title('POLE ZERO PLOT')
grid on
```

MATLAB Code for final controller

```
%parameters for transfer function
J=3.228E-6;
b=3.5077E-6;
K1=0.0314;
K2=0.0274;
K3=0.0356;
R=4;
L=2.75E-6;

%time
t=0:0.001:2;

%controller values kp = 80, kd = 1
Kp = 80;
Ki = 0;
Kd = 2;


s=tf('s');
%pid transfer function
```

```
U_s = Kp + (Ki/s) + s*Kd ;

%initial transfer function
H_m=K1*K2/(s*((L*s+R)*(J*s+b)+K2*K3));

%step responce
figure(1)
step(H_m,t,'b');
grid on

% TF after applying controller
H_mP = (( U_s*H_m)/ (1 + U_s*H_m));

%new TF
figure(2)
step(H_mP,t,'g');
xlabel('Voltage in volts')
ylabel('Angular Distance')
grid on
```

14. References

1) www.ctms.engin.umich.edu/CTMS/index. php? example=MotorPosition&section=System Modeling

2) http://en.wikipedia.org/wiki/Dc motor

3) A Textbook of electrical technology (Vol.2). New Delhi: S.chandcompany.

4) http://www.electrical4u.com/working-or-operating-principle-of-dc-motor

5) http://stm32f4-discovery.net/2014/11/project-03-stm32f4xx-pid-controller/

6) http://www.globalspec.com/reference/10801/179909/chapter-3-ac-and-dc-motors-servomotors-general-principles-of-operation

7) http://users.encs.concordia.ca/~realtime/elec372/docs/MODELING_SERVO_MOTOR_SYSTEM2.pdf