1. INTRODUCTION:

Since the nineteenth century, DC motor drives are used for many applications in industries and households. DC motors are generally controlled by typical PID controllers as they are easy to design to meet the requirements. This report aims at designing PID controller for a DC motor.

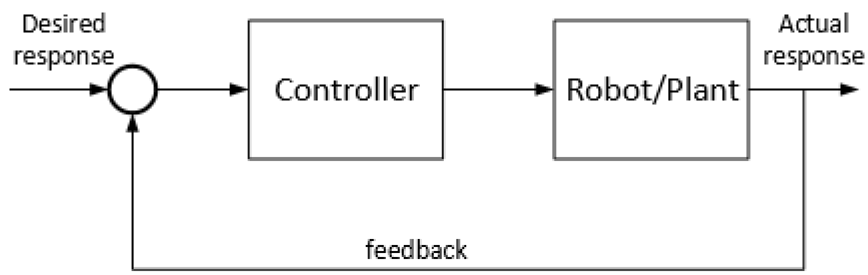


Fig 1: Simple Controller Design

We used Matlab/Simulink to run our simulations and final observations on current, angular velocity and angular position of the DC motor with respect to the controller for a step input is observed.

1. OBJECTIVES:

1. Build an open-loop model using Simulink.

a. Draw the waveforms of angular position, angular velocity, and current of the motor. Explain the function of each block;

b. Draw the waveforms of angular position, angular velocity, and current of the motor after changing the payload mass and the motor inertia;

2. Build a close-loop model using Simulink.

a. Draw the waveforms of angular position, angular velocity, and current of the motor;

b. Draw the waveforms of angular position, angular velocity, and current of the motor after changing the payload mass and the motor inertia;

c. When you set PID controller as kp = 10, ki = 0,kd = 0, you should obtain the output response as shown below. The desired u value was set to a constant 1 in the M file. The output oscillates for quite a long time. Adjust PID parameters to achieve an acceptable response (recall the control theory you have learned).

2. DYNAMIC MODEL OF THE SYSTEM:

This part covers the dynamic modelling of a closed loop control of a DC Motor. The simulations are made using matlab and Simulink. This system model consist of a armature controlled PM DC motor in which speed is being controlled by the applied armature voltage V and the field current If is held constant.

The given DC Motor specifications are as follows:

m_robot = 5;

Payload mass g = 9.81;

gravity rw = 0.02;

Wheel radius G = 1/30;

Gear ratio J_robot = m_robot*rw^2*1.1;

Robot inertia J_motor = 1.3e-4;

Motor inertia c = 1.0791e-5;

Viscous damping km = 0.415;

Motor torque constant ke = 50.68*60/(2*pi*1000);

Back emf constant L = 4.8e-3;

Armature winding Inductance R = 9.65;

Resistance V =12;   Input voltage

EQUIVALENT CIRCUIT OF A MOTOR:

A circuit of the motor drive system is constructed and evaluated using Kirchhoff's voltage law.

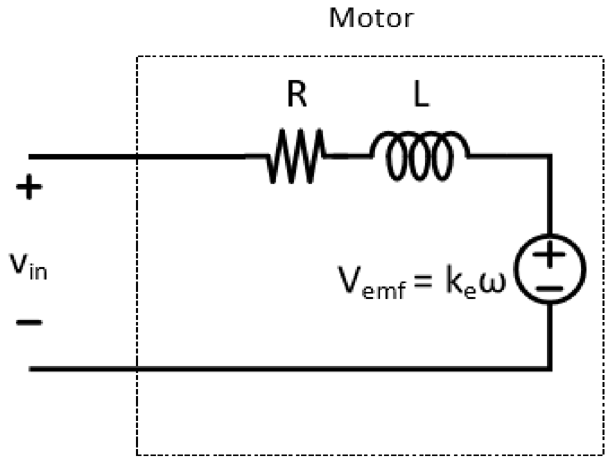


Fig 2: Eq circuit of motor

Summing voltages through the circuit gives

$$V_{in} = L\frac{di}{dt} + Ri + k_e\omega$$

where i is the current through the windings in the armature, L and R are the motor inductance and resistance, k_e is the electrical constant, $V_{emf} = k_e\omega$ the back emf and ω the rotational speed of the motor (rad/s).

Open loop transfer function of DC motor:

- $s(Js + b)Q(s) = KI(s)$
- $(Ls + R)I(s) = V(s) - KQ(s)$

$$\frac{Q(s)}{V(s)} = \frac{K}{(Js + \varepsilon)(Ls + R) + K^2}$$

Closed loop transfer function of DC motor:

$$\frac{Q(s)}{V(s)} = \frac{K}{s * ((Js + \varepsilon)(Ls + R) + K^2)}$$

$$sQ(s) = speed\ of\ the\ motor$$

The transfer function of the PID controller can be written as

$$C(s) = K_P + \frac{K_I}{s} + K_D s = \frac{K_D s^2 + K_P s + K_I}{s}$$

EQUATION OF MOTION:

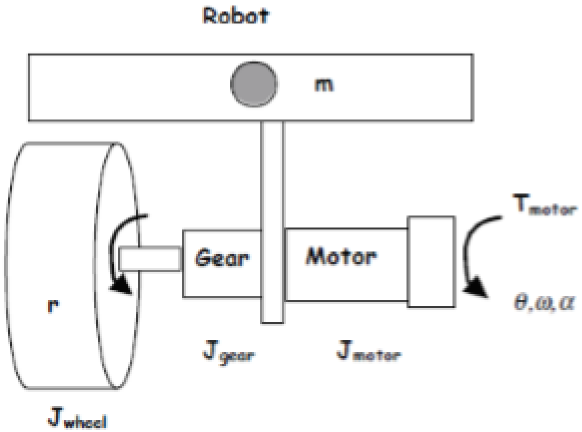


Fig 3: DC motor dynamic motion diagram

$$J\ddot{\theta} + c\dot{\theta} = T_m - T_{load}$$

$$J = J_{motor} + J_{gear} + (J_{wheel} + mr^2)\left(\frac{1}{G}\right)^2$$

J is found to be 1.9801

CONTROLLER METHOD:

A typical controlled DC motor system has the following

- An actuator (for example a motor) and its control board
- A sensor (speed, torque, position, etc.)
- A micro-controller board or Microchip IC components for appropriate gains

We should make sure that all of them must be supplied with appropriate electric energy in order to reach the desired value for the actuator which is called the set point. We can acquire this by using proper closed loop control system

"A control loop feedback system is a system that runs in a close loop, with a set-point to reach. The command given to the actuator to reach the set-point depends on the feedback. This feedback consists in the actuator's value measured by the sensor and compared to the set-point value. The resulting error is computed and re-injected into the initial order as a command that automatically corrects and adjusts the value of the actuator, in order to reach the set-point." [4]

A simple workflow of the control system:

- A desired value is sent to the motor controller (ex: spin at 12 rad/s)
- The sensor receives the status of the motor (ex: motor now spinning at 8rad/s)
- The error is calculated in the controller circuit (error is 4 rad/s)
- The error is injected into the motor (it happens in a form of compensation)

The control system will be efficiently able to inject the appropriate energy into the motor to get the desired rad/s even though if we add some load to the output.

Typically a control method can be classified into 2 types: open loop and closed loop controllers.

An open loop controller performs action independent of the current output. An example can be heating of water boiler which continuous to heat even though all water gets boiled. But a closed loop controller on this water boiler can switch it off when desired temperature of water is reached.

In our simulation we will be demonstrating the effect of closed loop control systems on DC motor energy and motion.

Proportional Controller:

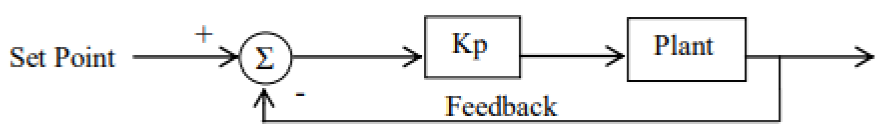


Fig 4: P controller Design

Proportional controller can decrease the steady state error of the system by increasing the proportional gain factor K. However, it cannot completely compensate the steady state error of the system. P controller generally leads to overshoot and causes oscillations before reaching the desired value. This can be prevented by adding an Integral controller along with P controller.

PI Controller:

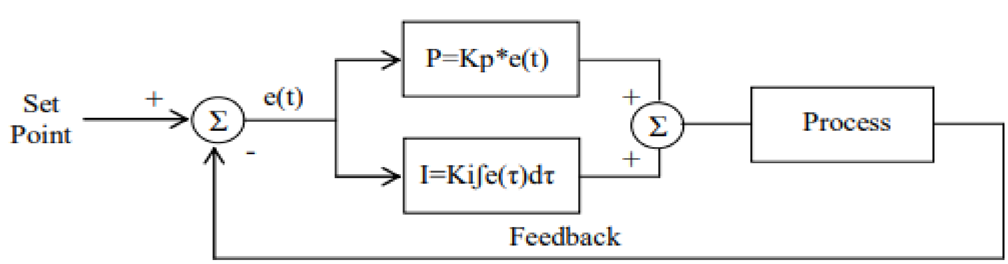


Fig 5: PI controller Design

It can effectively remove the steady state error from P controller. But, it is also used where speed is not a big issue. It has no ability to predict future errors and a sudden change can result in huge oscillations.

PD Controller:

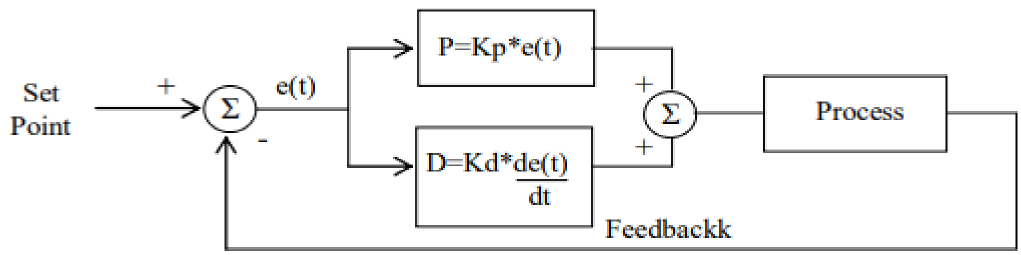


Fig 6: PD controller Design

PD Controller can improve the stabilty of the system by improving control since it has an ability to predict future error of the system response. The derivative is taken from output response which can avoid effects of sudden change in error signal. It application includes suspension car.

Proportional Integral Derivative -PID Controller:

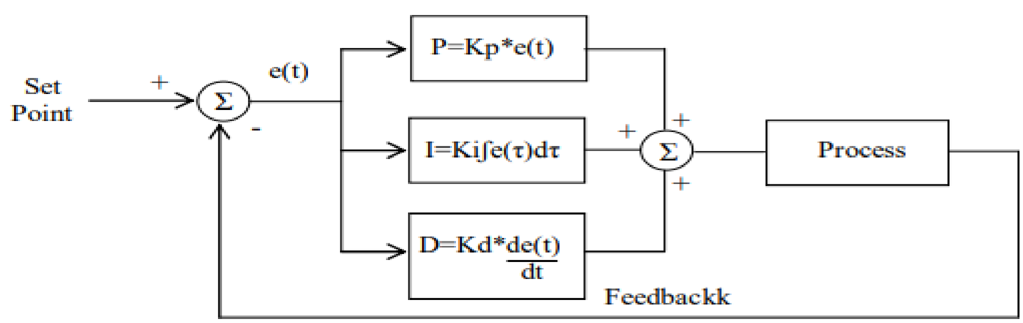


Fig 7: PID Controller Design

PID controllers are highly preferred for industrial applications due to the following reasons

- ✓ Zero Steady state error
- ✓ Fast response (short rise time)
- ✓ No Oscillation
- ✓ Higher Stabilty

The additional Derivative component to the PI controller eliminates the overshoot and oscillations. It improves the stability of the system and increases the gain Kp for fast control response. The output of PID controller consists of three terms the error signal, the error integral and the error derivative.

Effects of P, I, D components:

PARAMETERS	RISE TIME	OVERSHOOT	SETTLING TIME	S-S ERROR
K_p	**Decrease**	**Increase**	**Small Change**	**Decrease**
K_i	**Decrease**	**Increase**	**Increase**	**Eliminate**
K_d	**Small Change**	**Decrease**	**Decrease**	**Small Change**

This table can be used when tweaking the values of PID parameters. The above table may not function exactly the same as given, because changing a parameter may impact on other parameters as well.

Using these table we can tune the PID. There are various methods for tuning PID namely: Manual tuning method, Ziegler-nicols tuning method and PID tuning software.

Advantages and Disadvantages of different tuning methods:

METHODS	ADVANTAGES	DISADVANTAGES
Manual Tuning	No math required. – trial and error method	Requires experienced personnel.
Ziegler-Nichols	Proven method. – Trial and Error method	Mathematical calculations required
Software Tools	- Consistent tuning. - Online or offline method. - May include value and sensor analysis. - Allow simulation before downloading. - Can support non-steady state (NSS) tuning.	Some cost and training involved.

For our design, we used Simulink toolbox to tune the P,I and D parameters

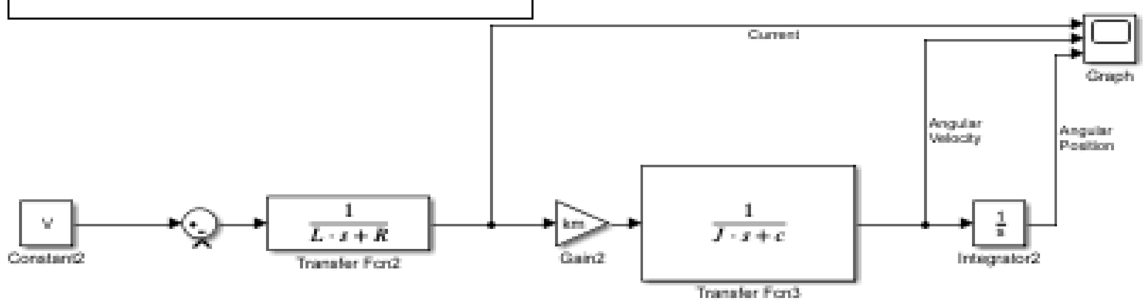
4. SIMULATIONS AND RESULTS:

The parameters loaded in matlab for Simulink simulations:

Size	Name	Value
1x1	c	1.0791e-05
1x1	g	9.8100
1x1	G	0.0333
1x1	J	1.9801
1x1	J_motor	1.3000e-04
1x1	J_robot	0.0022
1x1	ke	0.4840
1x1	km	0.4150
1x1	L	0.0048
1x1	m_robot	5
1x1	out	*1x1 SimulationOutpu*
1x1	R	9.6500
1x1	rw	0.0200
1x1	u	1
1x1	V	12

1. OPEN LOOP SIMULATIONS: *(QUESTION 1 : A and B)*

Fig 8: Open loop plant circuit



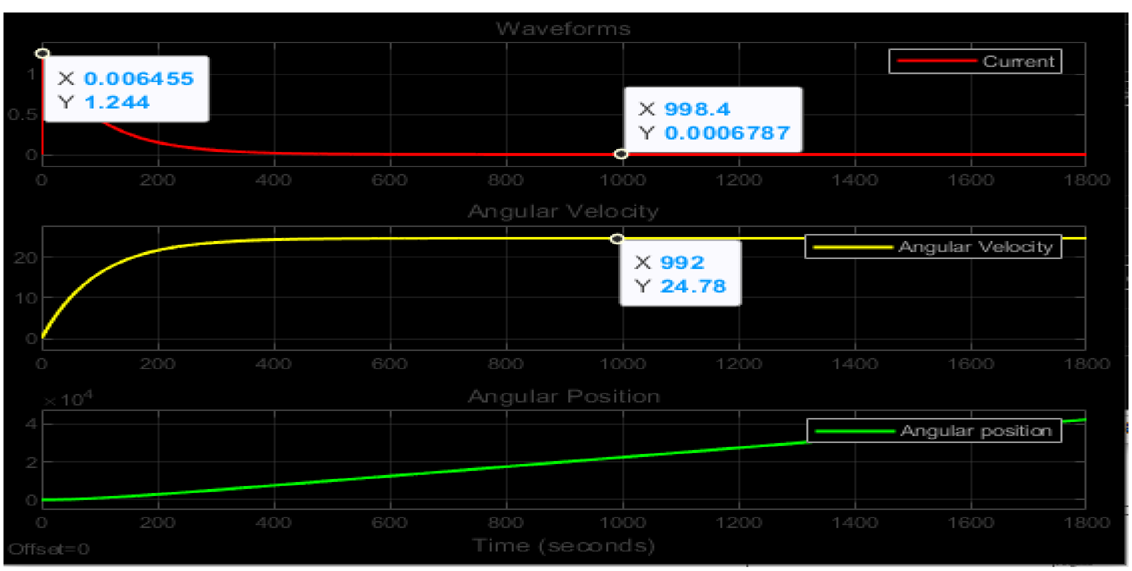


Fig 9: Open loop simulation results

The above circuit is the open loop of the plant model.

As you could see in the graph below, as the input is given, the velocity keeps increasing in linear pattern and angular position also encrease. The current goes to peak at 1.24 A and starts to drop because the motor reaches its maximum speed at no load conditions. At this speed, the back emf voltage in the motor is equal to the input voltage (minus a small amount to overcome inefficiencies in the motor)This equation represents a linear torque/speed relationship for a PM DC motor. The response (assuming a constant input applied voltage) looks like:

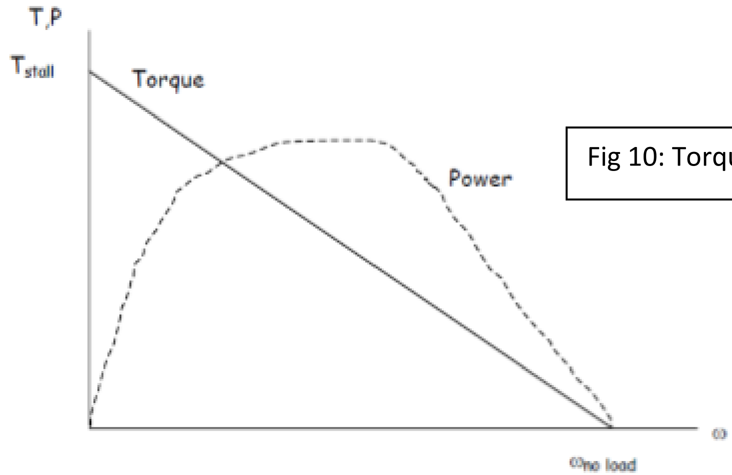


Fig 10: Torque vs Speed graph

And hence the current drops down to minimum at 24.78 rad/s of motor. It takes steptime = 1800 fr current to settle to 0.

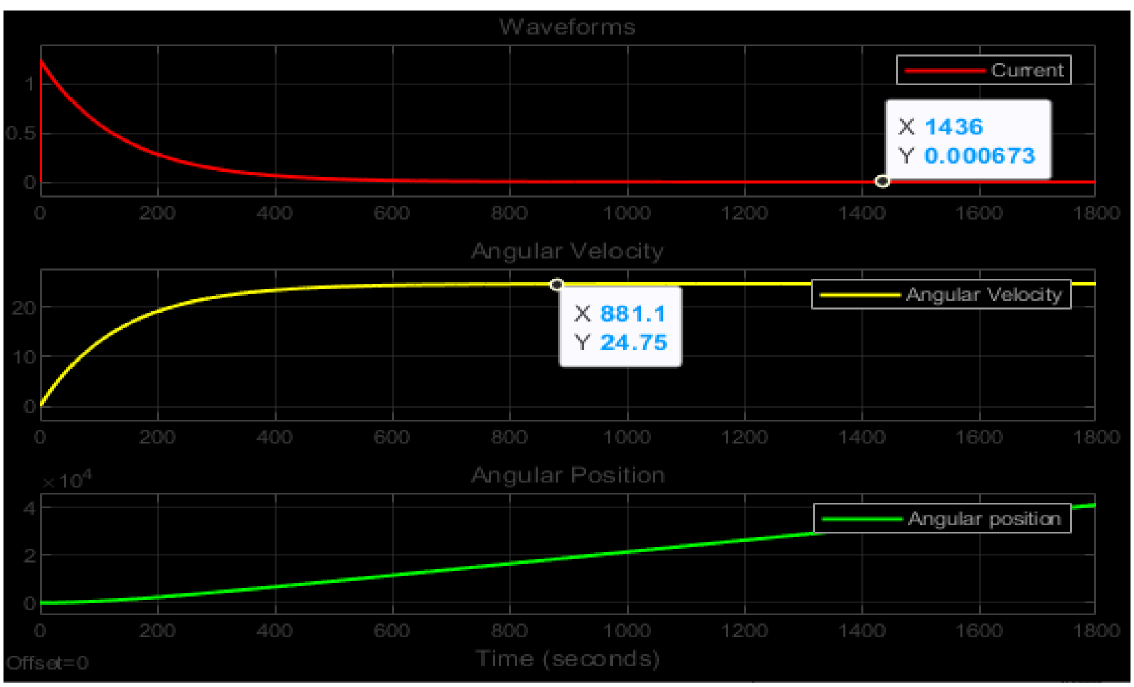


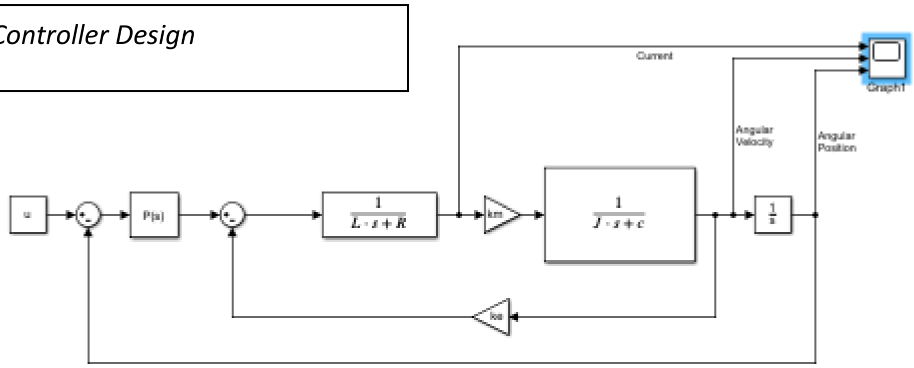
Fig 11: Open loop simulation results after change payload mass and J_motor. It appears to be getting the same result. This shows that our plant is consistent.

2. CLOSED LOOP SIMULATIONS:

We will start off by designing from P controller and see how our system reacts with various controllers. The desired u value was set to 1 in .mat file.

P CONTROLLER: *(Question 2 C)*

Fig 12: P Controller Design



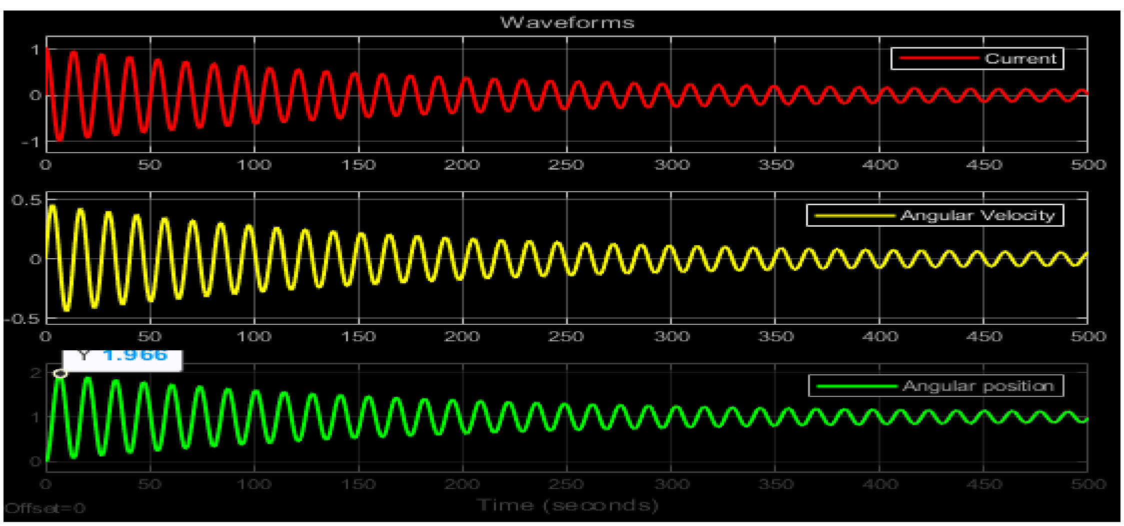


Fig 13: P Controller Simulation Results

As you could see in the graph that the angular position of the DC motor is not suitable. The stability of the controller is very poor (Too much oscillations).

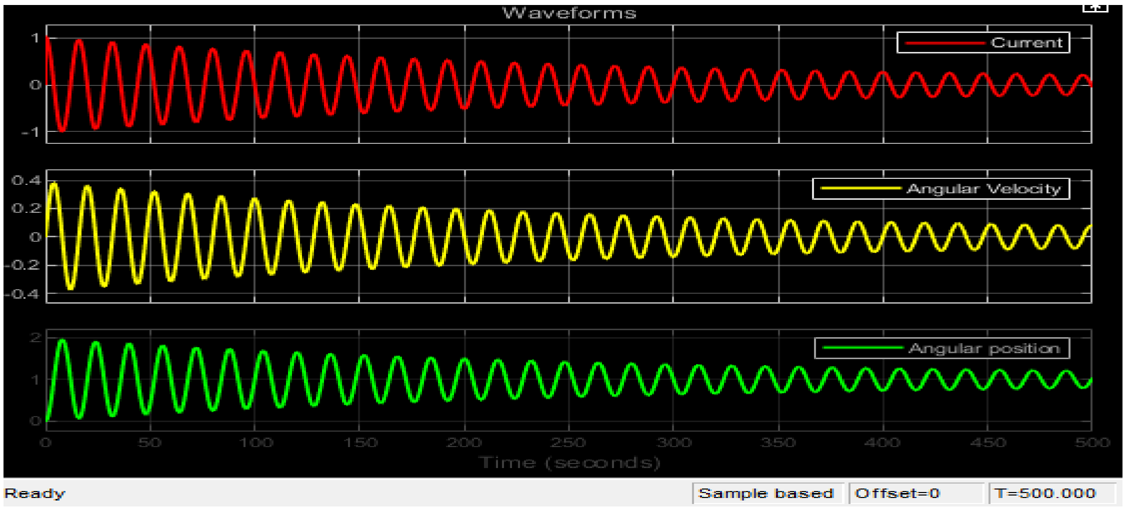


Fig 14: Simulation result of P controller after changing payload mass to 10kg and J_motor to 5.3e-4 kgm2. It appears to be getting the same result. This shows that our

PI CONTROLLER:

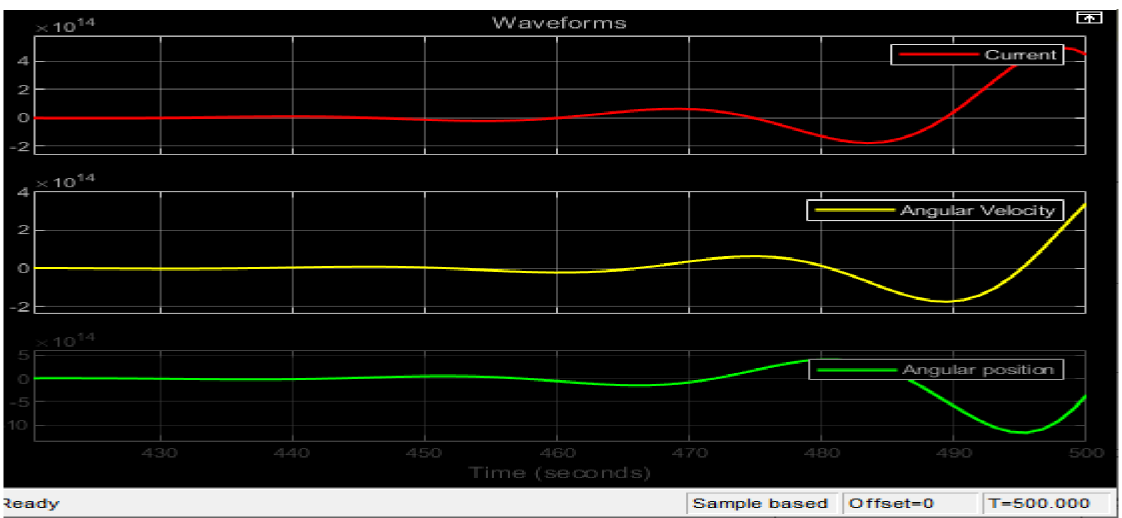


Fig 15: PI Controller Simulation Results

The PI controller appears to be very inefficient. It gives an undesirable response when the error signal is activated and making the system to overshoot and oscillate

PID CONTROLLER: *(Question 2: A and B)*

We can observe that our controller response has been significantly. We can achieve more stability, minimum overshoot and faster response

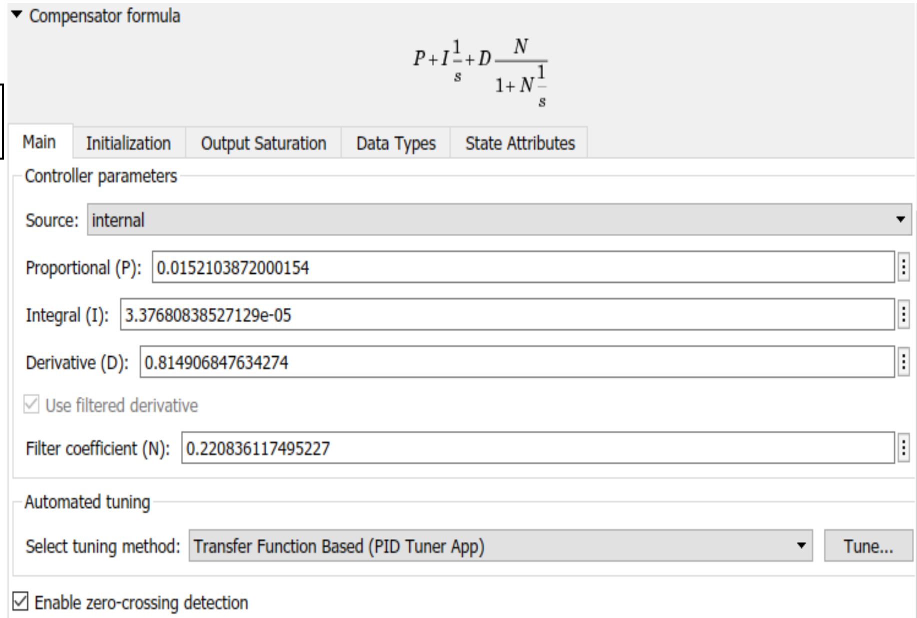


Fig 16: PID Values

For this PID values we get the following response.

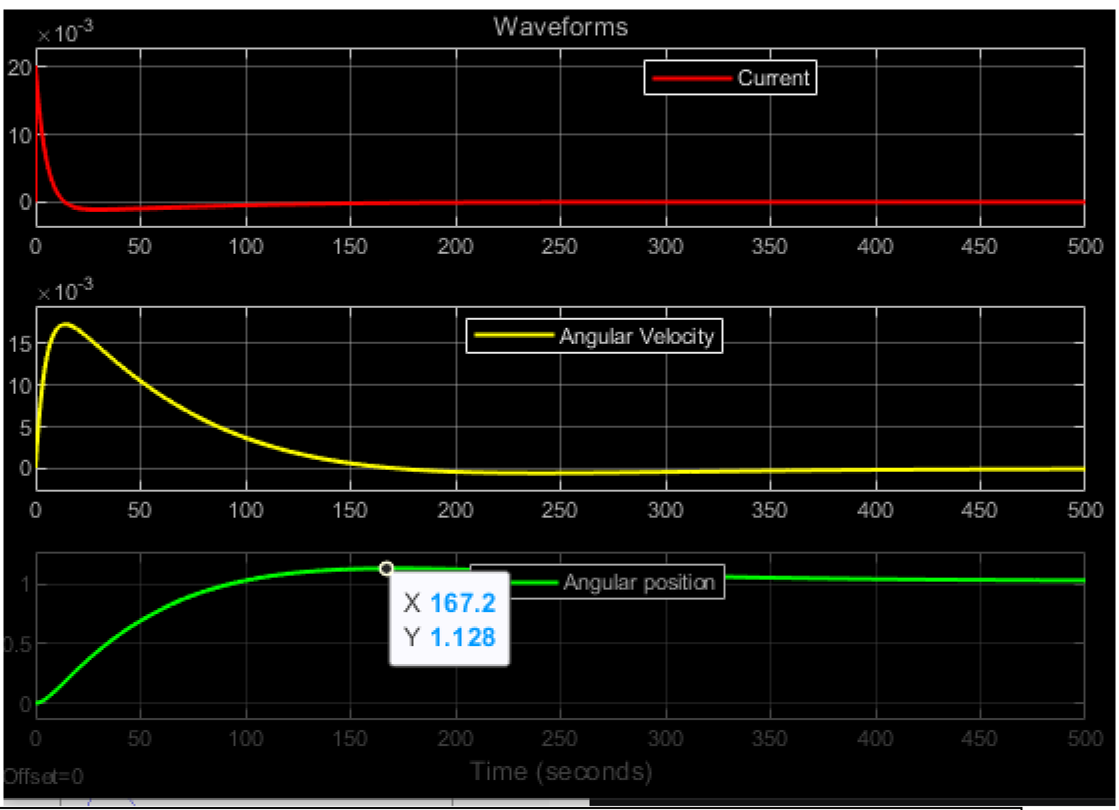


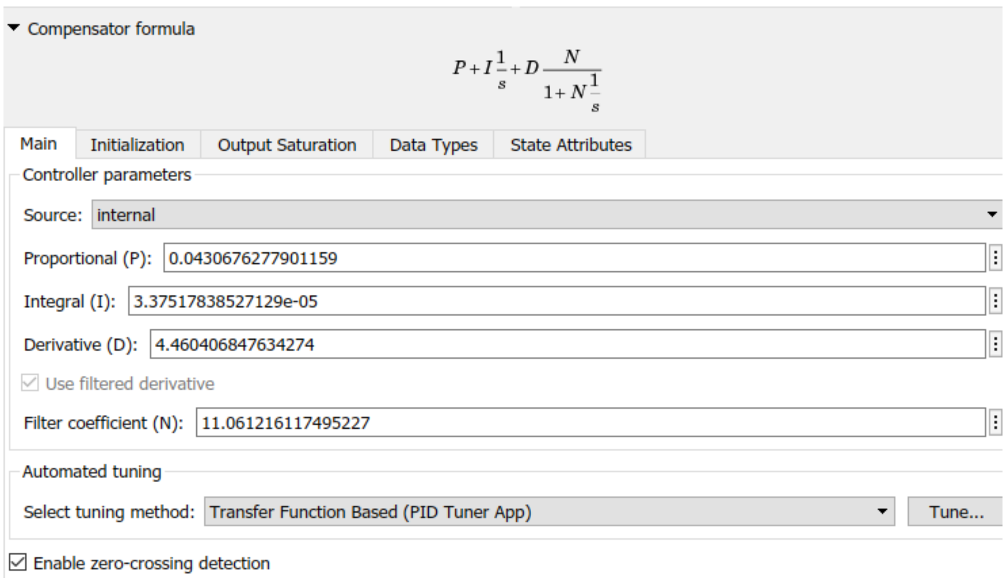
Fig 17: Simulation Result of PID Controller

The above graph has minute overshoot but this overshoot is less than 20% of steady state value and is in the tolerance band. The system takes 100 seconds to attain the desired stable response.

We can find that the current reaches its maximum when a step input is given and reduces to 0 in an exponential pattern. Unlike open loop, here the angular velocity of the motor is found to be decreasing after reaching its maximum. And the angular position remains stable.

A MORE BETTER PID RESPONSE:

Fig 18: Better PID Values



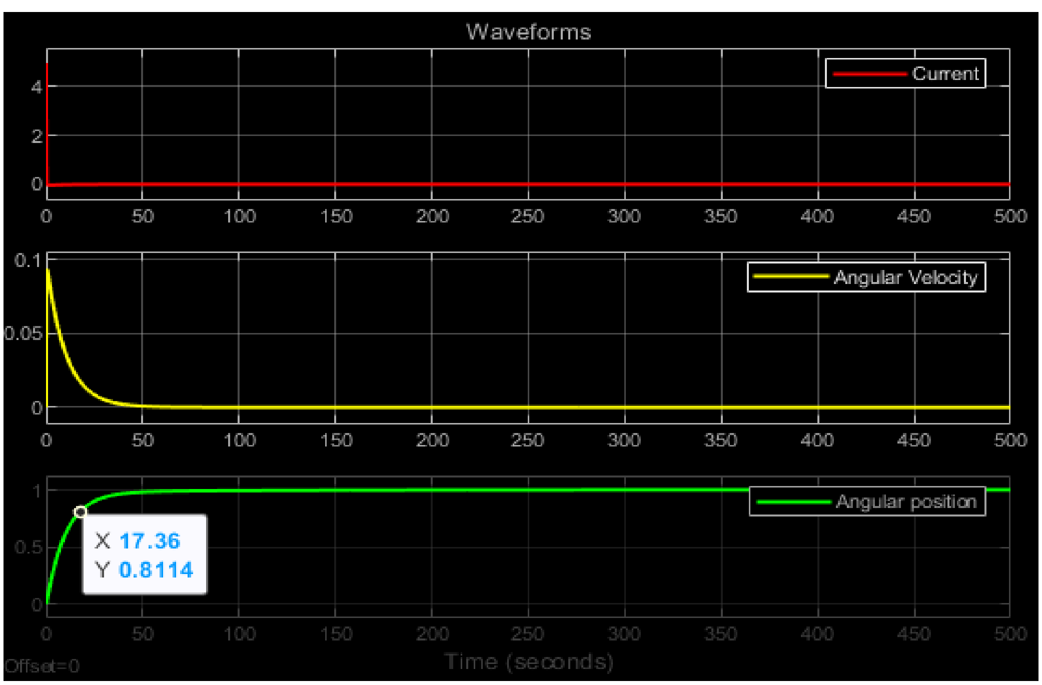


Fig 19: Better PID Results

We have tuned the PID for better result. We can achieve the following:

- ✓ Faster response (small rise time)
- ✓ Smaller settling time
- ✓ Critically damped system

Various simulations were made with different u-desired value. And all simulations appears to obtain desirable steady state response. The same can be observed by changing u value in the .mat file.

5. Conclusion:

We have successfully designed a simple PID Controller for the DC motor. The speed of a dc motor has been successfully controlled by using smart Controller. A generalized modeling of dc motor is done. After that a complete layout of DC drive system is obtained. The effect of Current, angular velocity and angular position has been observed. We can derive a stable response in our PID controller simulations. One should also verify that the controller is working in desired condition when different loads are used and also when the input is suddenly changed, the controller should be able to detect the undesirable value and either compensate it or shut down rather than overshooting and destroying the systemWe can improve our model by adding motor losses and developing a more realistic controller by using both continuous and discrete methods. A real world pid controller should consider the model parameters, required rise time, settling time and allowable overshoot, sampling time and windup losses. These simulations are very useful for hardware implementations. A typical PID controller can be easily implemented using Development kits like Arduino, Beagle board, FPGA and Analog IC components on a circuit board.

6. References:

[1]. Awang M.A.F., "DC Motor Speed Controller," no. November, 2010.

[2]. "PID Design Method for the DC Motor Speed Control", Internet: www.dii.unisi.it/~control/ctm/examples/motor/PID2.html, 2012 [Feb 20, 2015].

[3]. "Feedback and control systems" - JJ Di Steffano, AR Stubberud, IJ Williams. Schaums outline series, McGraw-Hill 1967

[4]. "An introduction to PID control with DC motor", Internet: https://medium.com/luosrobotics/an-introduction-to-pid-control-with-dc-motor-1fa3b26ec661, [Feb 27, 2018].