1. Develop a open-loop model of the traction drive shown in Figure 1(a). And please derive its transfer function.

The open loop model of the traction drive motor and capstan roller is shown in below Figure 1. As per the given problem statement the drive uses a DC armature-controlled motor with a capstan roller attached to the shaft. The motor rotates the capstan roller which in-turn moves the Drive bar. The drive bar moves the linear slide table according to the desired position for the machine.

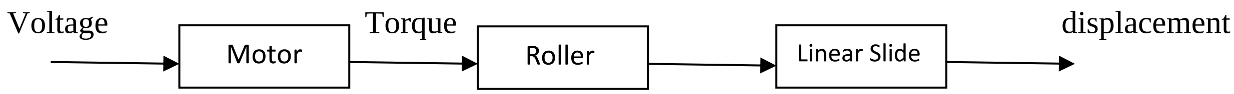

Figure 1

For obtaining the transfer function from the open-loop model we need to know the exact transfer function of the Motor, Roller and the Linear Slide.

a) DC armature-controlled Motor

In a armature controlled DC motor, the field current is held constant, and the armature current is controlled through the armature voltage. The transfer function from the input armature current to the resulting motor torque is

$$\frac{Tm(s)}{Ia(s)} = Km$$

Km is the Torque constant

For a typical motor the following voltage relationship holds true (on the armature side of the motor)

$$V_a = V_R + V_L + V_b = R_m + L_m\,(d\, i_a / dt) + V_b \quad \text{……….} \ (a)$$

Where,

V_b is the “back EMF”

i_a is the armature current

R_m is Motor Resistance

L_m is Motor Inductance

The back EMF V_b is proportional to the speed ω and is given by:

$$V_b\,(s) = K_b\,\omega(s)$$

K_b is the Back emf constant

Taking Laplace transform of Equation (a) gives

$$V_a(s) - K_b\,\omega(s) = (R_m + L_m s)\, I_a(s)$$

By summing moments for the rotational motion of the motor as well as the load i.e. capstan roller attached to the shaft gives

$$T_m = J_m\,\dot{w} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(2)}$$

T_m is the resulting motor torque

J_m is the Inertia of the roller, shaft, motor and tachometer

The above equation assumes that the motor viscous friction constant is zero. Using the above equations the transfer function for a DC armature-controlled motor can be calculated as:

$$\frac{\omega(s)}{Va\ (s)} = \frac{\frac{Km}{JLm}}{\left(s+\frac{Rm}{Lm}\right)s+\left(\frac{KbKm}{JLm}\right)} \quad \ldots\ldots\ldots\ldots\ldots\ldots \text{(3)}$$

The block diagram of the above calculated transfer function is shown in figure (2)

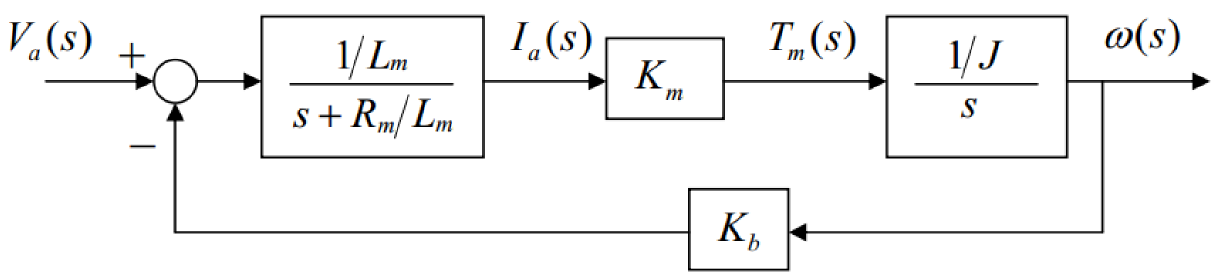

Figure 2

Using Laplace Transform on equation (2), we get

$$\frac{\omega(s)}{Tm(s)} = \frac{1}{sJ}$$

Using the above equation we can get the Transfer function relating the input armature voltage and the resulting motor torque

$$\frac{Tm(s)}{Va\ (s)} = \frac{\frac{Km}{Lm}s}{\left(s+\frac{Rm}{Lm}\right)s+\left(\frac{KbKm}{JLm}\right)} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(4)}$$

From the above equation we can find the resulting motor torque (Tm). Since the capstan roller (with the given radius r) is attached to the shaft of the motor, the force generated due to this torque can be calculated using:

$$\text{force} = \frac{Torque}{r} \text{ ………………………..……… (5)}$$

This force will be used by the drive bar to move the linear slide table resulting in the final motion. The following set of equations relate the displacement of the linear slide table to the applied armature voltage (Va):

$$\text{force} = M\ddot{x} \text{………..……………………………….. (6)}$$

Where,

M is the combined mass of the slide and the drive bar ($M_s + M_b$)
x is the displacement of the linear slide and $\ddot{x}$ is the double derivative of x

Applying Laplace transform on equation (5) and (6) we get

$$F(s) = \frac{Tm(s)}{r} \text{ ………………………………………... (7)}$$

$$F(s) = s^2 MX(s) \text{……………………………… (8)}$$

Using equations (4), (7) and (8) the transfer function of the traction drive comes out to be:

$$\frac{X(s)}{Va\ (s)} = \frac{\frac{1}{s}\frac{Km}{rMLm}}{\left(s+\frac{Rm}{Lm}\right)s+\left(\frac{KbKm}{JLm}\right)} \text{ ………………….. (9)}$$

2. Design a switch mode PWM DC-DC converter for the speed regulation of the DC motor.

Voltage source = 60 v

PWM saw tooth magnitude = 5v

Therefore we can use buck converter as shown in Figure 3 -

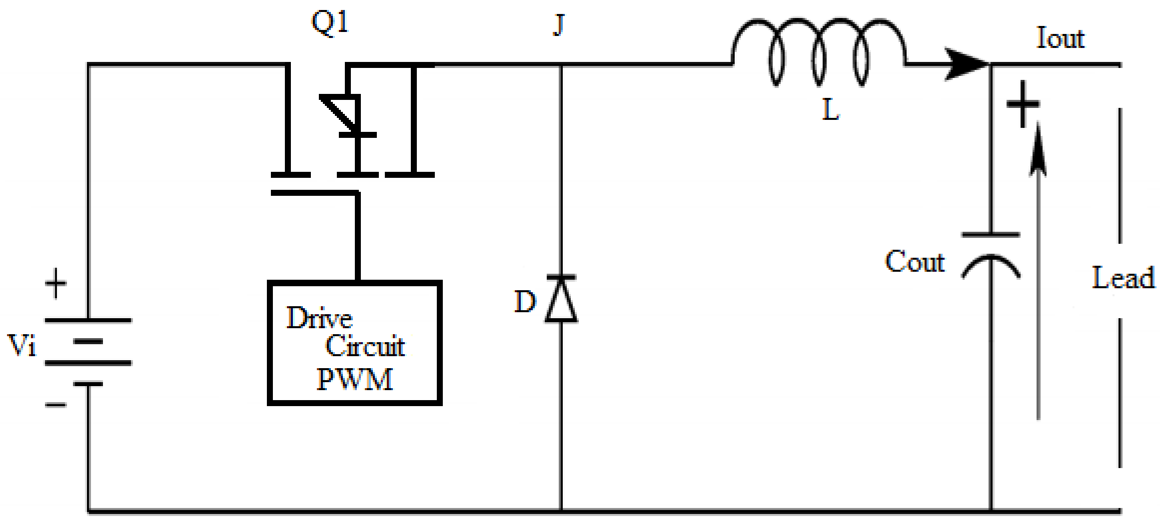

Figure 3

Duty cycle (D) of PWM

$$D = \frac{V_{out}}{V_{in}}$$

The inductance L of the converter can be calculated using the following equation:

$$L = \frac{V_{out}\,(V_{in} - V_{out}\,)}{\Delta I_L * f_s * V_{IN}}$$

Where,

V_{in} = Input Voltage

V_{out} = Output Voltage

f_s = Converter switching frequency

ΔI_L = inductor ripple current = 20% of I_{out}

The capacitance is found using the equation:

$$Cout = \frac{\Delta I_L}{8 * f_{s*}\, \Delta V_{out}}$$

c. If the switching frequency, f_s is high, L and C_{out} values will be low. Typical choice of the switching frequency can be,
f_s = 100 khz or 500 khz.

d. During normal operation the transistor Q1 is repeatedly switched on and off, with the on &off times governed by the PWM driver circuits. This switching action causes a train of pulses at junction J, which is filtered by the LC of output filter to produce a DC output voltage

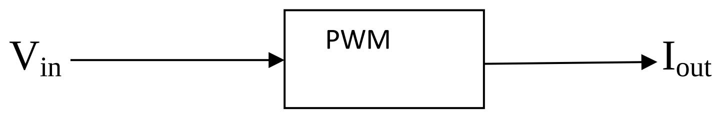

Figure 4

3. Design the current loop controller so that phase margin of the loop transfer function should be over 45^o, and the crossover frequency is no larger than 5 kHz.

The Electrical Drives are mainly used for the following tasks:

i) Starting
ii) Speed Control
iii) Braking

During the starting, there is a chance of huge current flow through the motor circuit. Figure 5 shows a current loop control strategy to limit the current flowing through the buck converter as well as the DC motor. The feedback loop works when the current limit exceeds the set threshold value and brings down the current below the safe limit. In case of normal operation the feedback loop remains open hence it doesn't affect the normal operation.

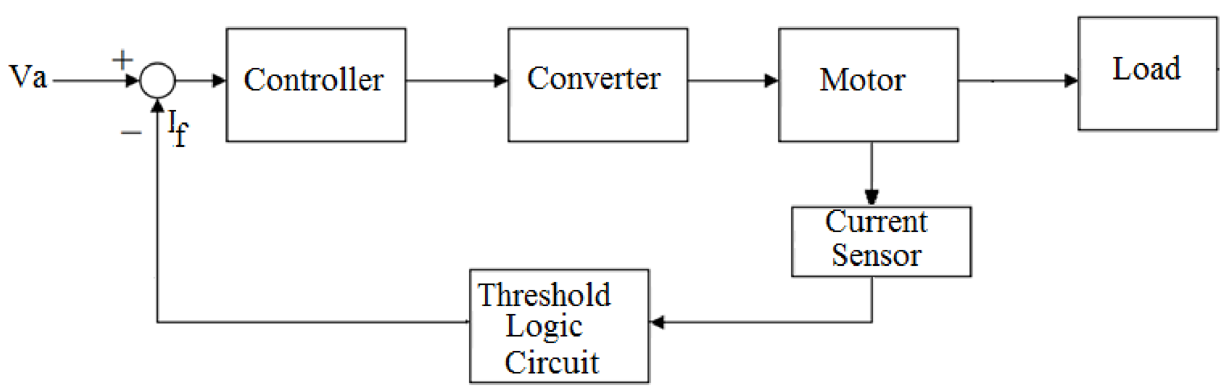

Figure 5

The main purpose of the current controller is to prevent overloading the motor. The current controller also plays a part in regulating the power supply (as it controls the output current) thus it is required that this controller should produce a quick stable control action. Further, since the design incorporates a power electronic converter (the DC-DC Buck Converter) as armature supply, a proportional-integral controller (PI) is adequate to meet the specifications required for the current loop control. As the response of the current controller is required to be fast, the choice of a PI controller is justified. The transfer function of a simple PI controller can be of the form:

$$Ci(s) = Kp + \frac{Ki}{s}$$

Where,
Gc is the gain
T_1 is the integrating time constant

Using equations (a) and (4) we can calculate the transfer function relating the armature current with the armature voltage. The transfer function comes out to be:

$$\frac{\mathrm{Ia(s)}}{Va\ (s)} = \frac{\frac{1}{Lm}s}{\left(s+\frac{Rm}{Lm}\right)s+\left(\frac{KbKm}{JLm}\right)} \text{………………………..(10)}$$

Using the values given in table 1, we get:

$$\frac{\mathrm{Ia(s)}}{Va\ (s)} = \frac{277.78s}{(s+377.78)s+(17875.456)}$$

Using the above transfer function and tuning the PI controller for phase margin greater than 45^o with the cross over frequency not exceeding 5 Khz (1000 π rad/sec) the values of Kp and Ki are found to be:

Kp = 0.3955
Ki = 702.7961

At these values the Phase margin (rad/s) comes out to be 120^o at frequency equal to 390 rad/s. Thus, the above designed controller meets the specifications required for the current control. The step response with and without controller are shown in the figures below:

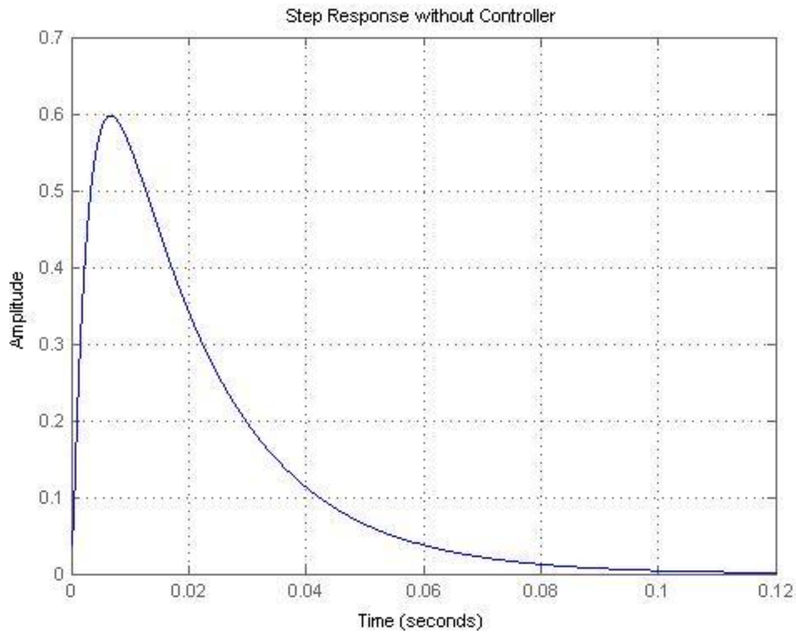

Figure 6

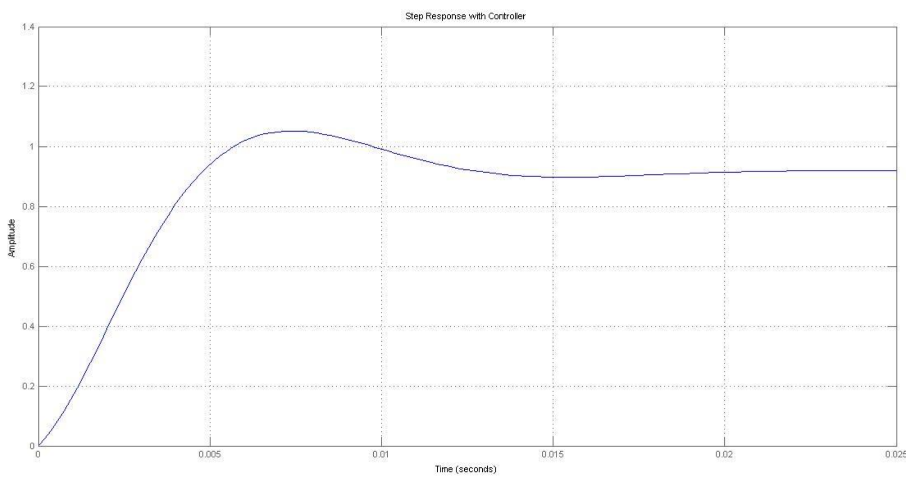

Figure 7

Clearly, the PI controller reduces the overshoot as well as the steady state error and tries to maintain a constant current through the converter and the motor thus, protecting both of them from current surges.

4. Design the speed loop controller so that phase margin of the loop transfer function should be over 45^o, and the crossover frequency is no larger than 5kHz.

Speed control loops are perhaps the most widely used feedback loops for drives. With the block diagram of this loop it will be a lot easier to understand this type of control.

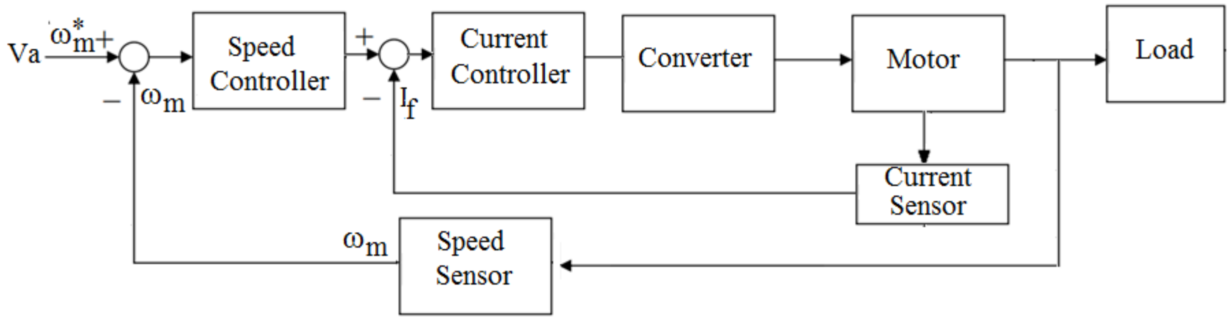

Figure 8

We can see from the diagram that there are two control loops, an inner loop and outer loop. The inner loop is for current control and limits the current flowing through the motor and the converter. The other loop primarily controls the speed of the motor and tries to maintain a constant speed throughout. If there is an increase in the set reference speed ωm* the error Δωm will be positive indicating that the speed at which the motor is rotating needs to be increased. In order to increase the speed of the motor, the current needs to be increased. This is taken care by the inner control loop as it increases the current keeping a check on the maximum allowable limit. As the current increases the motor accelerates and a constant feedback is send via the speed sensor. When the referenced or the desired value is reached the error term reduces close to zero (not exactly equal to zero) indicating that there is no further need to increase the speed of the motor.

Since the speed control of the motor requires both a quick response as well as a check on the desired value, a PID controller is best suited for such a control. Using the transfer function described in equation (3):

$$\frac{\omega(s)}{Va\ (s)} = \frac{\frac{Km}{JLm}}{\left(s+\frac{Rm}{Lm}\right)s+\left(\frac{KbKm}{JLm}\right)}$$

The transfer function mentioned above does not incorporate the Current controller (obtained transfer function of the PI controller). Therefore the transfer function needs to be multiplied by:

$$Ci(s) = 1.044 + \frac{246.9852}{s}$$

Substituting the values of all the variables from table 1 and multiplying the transfer function with the transfer function of the PI controller we get,

$$\frac{\omega(s)}{Va\ (s)} = \frac{22272.32s + 5269093.3421}{\{(s+377.78)s+(17875.456)\}s}$$

The PID controller can be tuned accordingly to match the required phase margin and crossover frequency criteria. The equation below describes a simple PID controller:

$$Cw(s) = Kp + \frac{Ki}{s} + sKd$$

The values of Kp, Ki and Kd are found to be:

Kp = 5.4721;
Ki = 70.2297;
Kd = 0.10659;

At these values the Phase margin (rad/s) comes out to be 116^{o} at frequency equal to 1.52 Krad/s. Thus, the above designed controller meets the required speed control. The step response with and without controller are shown in the figures below:

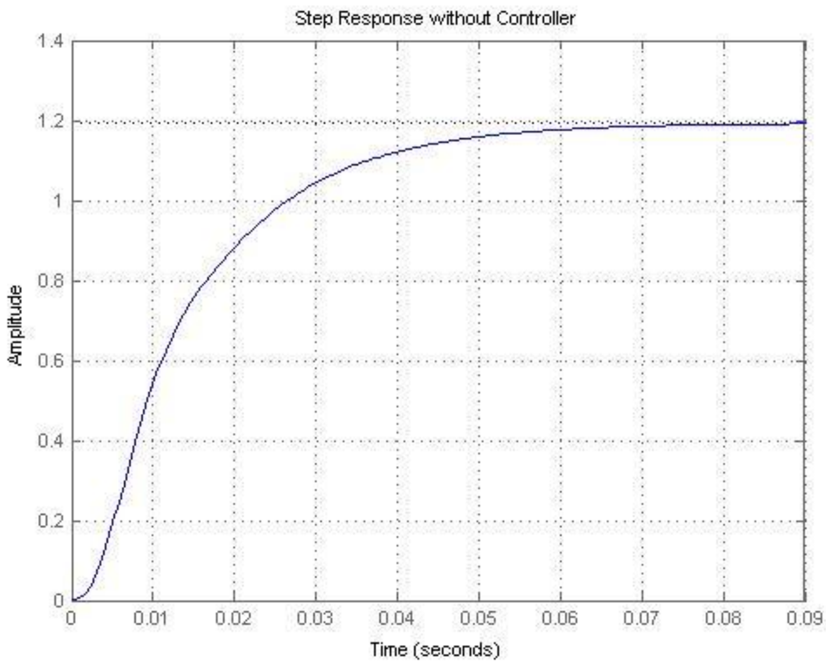

Figure 9

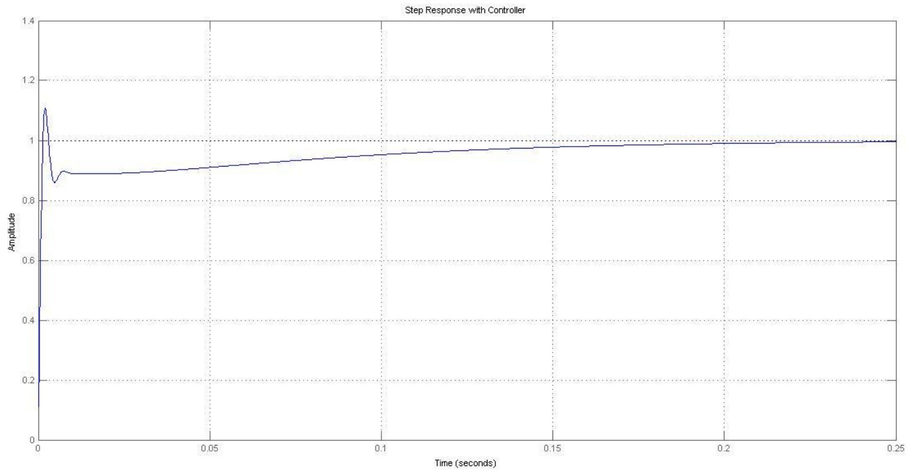

Figure 10

Clearly, the PID controller reduces the overshoot as well as the steady state error and tries to maintain a desired speed of the motor.

5. Design the position loop controller so that phase margin of the loop transfer function should be over 45^{o}, and the crossover frequency is no larger than 5kHz.

In the given system the linear slide is displaced via a capstan roller attached to the shaft of the motor. The position control can thus be a PID feedback controller relating the position axis and the resulting motor torque thereby controlling the rotatory motion of the motor. The transfer function described below clearly relates the resulting motor torque with the position of the linear slide.

$$\frac{\mathrm{X}(s)}{Tm\ (s)} = \frac{1}{rMs\text{^}2}$$

The block diagram describes the designed position control:

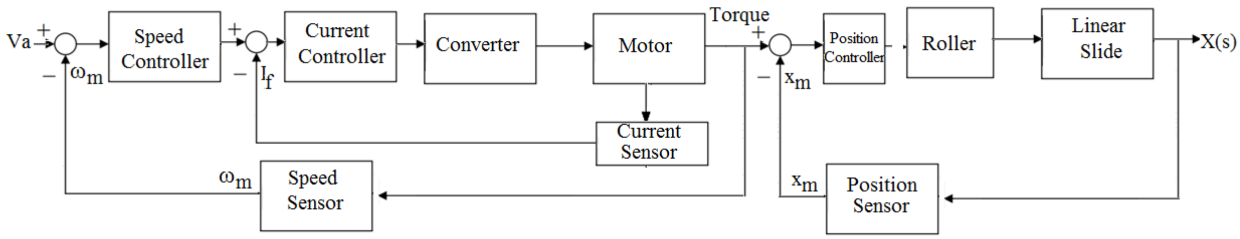

Figure 11

The figure below shows the step response of the above mentioned transfer function without any control:

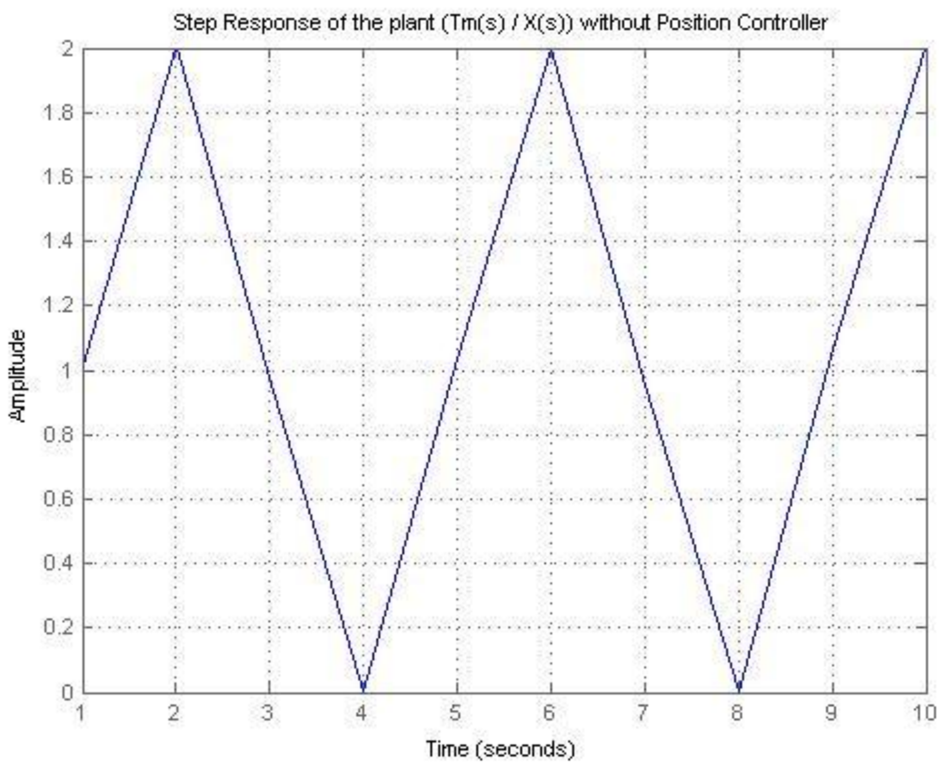

Figure 12

Clearly such a response can be controlled using a PID Controller. The equation below shows a typical position controller:

$$Cw(s) = Kp + \frac{Ki}{s} + sKd$$

The values come out to be:

Kp = 20.086
Ki = 3.4079
Kd = 3.5132

At this value the Phase margin (rad/s) comes out to be 121^{o} at crossover frequency equal to 10.1rad/s. Thus, the above designed controller meets the required position control. The step response with controller are shown in the figure below:

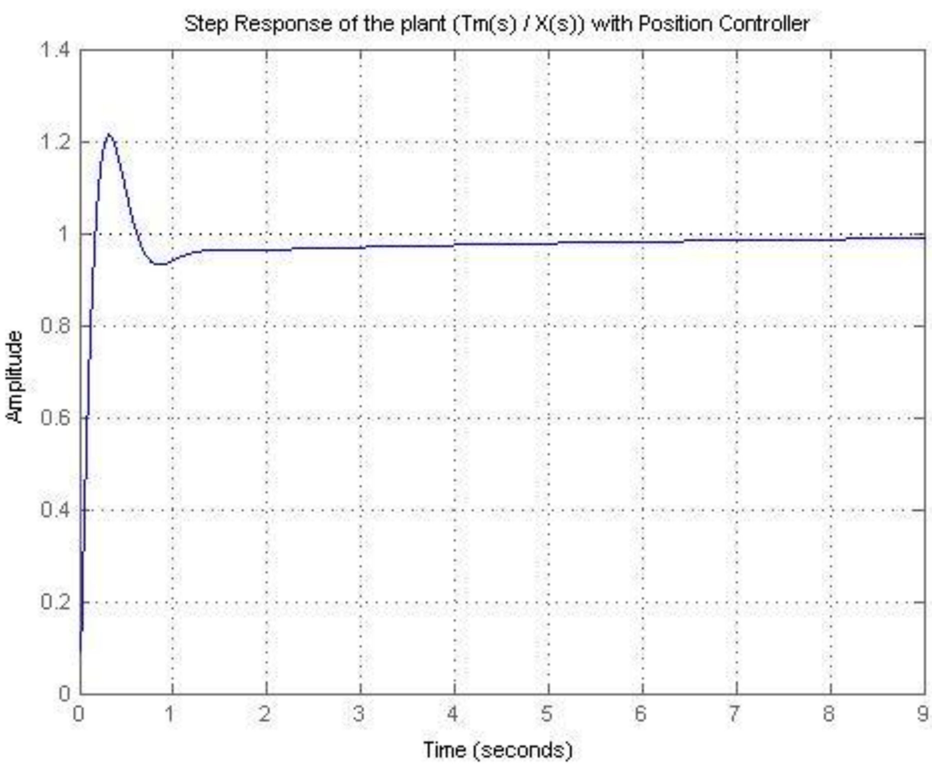

Figure 13

The PID controller reduces the overshoot as well as the steady state error and tries to maintain a desired position of the linear slide.

6. The Matlab code is attached at the end of the report.

The Matlab model is built step by step by first modeling the transfer function of the Motor continuing with the modeling od drive bar and linear slide. After this the transfer function of various controls are added. The current control is modelled as a unity gain feedback PI controller. For controlling the speed of the motor a unity gain feedback PID controller is employed. Together, the current control and the speed control form a parallel cascaded structure. Finally, the position control is made by controlling the resulting motor torque according to the referenced position of the linear slide. The controller employed is a unity gain feedback PID controller. The position controller along with the other two controllers (current loop controller and speed controller) form a series cascaded structure. The tuning of all the three controllers is done using the "pidtool" in Matlab. The "pidtool" just requires the transfer function of the plant and tunes the required (P, PI or PID) controller automatically, giving the values of Kp, Ki and Kd. If the automatically adjusted controller does not meet its specifications (phase margin > 45^o and crossover frequency < 5 KHz) then the response time of the controller can be adjusted to meet the specifications.

The figures below show the obtained Bode plots indicating the Phase Margin and the crossover frequencies for all the three controllers. The bode plots were plotted using the "margin" command on Matlab.

Please find below the documented Matlab code for the system:

```
% Electric Drives Project %
clear all;
close all;
% Declaring the different parameters for the Armature Comtrolled %
% DC Motor and the Capstan Slide %
J = 10.91*10^-3;         % Inertial of roller, shaft, motor and tachometer %
K = 0.8379;              % Torque constant / Back EMF constant %
R = 1.36;                % Motor Inductance %
L = 3.6*10^-3;           % Motor Inductance %
M = 12.6530;             % M is the sum of the mass of slide and drive bar %
R_rad = 31.75*10^-3;     % Roller Radius %
s = tf('s');

% Using different equations - transfer function relating input armature %
% voltage with the output current %
I_motor = (s/L)/((s+(R/L))*s + (K^2/(J*L)))
step(I_motor);
title('Step Response of the plant (Ia(s) / Va(s)) without the Controller');
grid on;

% Kp and Ki for current controller %
Kp = 0.3955;             % Value of Kp calculated using pidtool %
Ki = 702.7961;           % Value of Ki calculated using pidtool %
Ci = pid(Kp, Ki);
```

```
% Calculating the transfer function with the controller %
sys_cl = feedback(Ci*I_motor, 1);
figure;
bode(sys_cl, logspace(0,2))
margin(sys_cl)
grid on;
figure;
step(sys_cl);
title('Step Response of the plant (Ia(s) / Va(s)) with Controller');
grid on;

% Using different equations, transfer function relating input armature %
% voltage with the resulting motor speed %
Ia =  (1/L)/(s + R/L);
current_contr = feedback(Ci*Ia, 1);
Speed_contr = feedback(current_contr*K*(1/(s*J)), K);
figure;
step(Speed_contr);
title('Step Response of the plant (Wm(s) / Va(s)) without Speed Controller');
grid on;

% Kp, Ki and Kd for speed controller %
Kp = 5.4721             % Value of Kp calculated using pidtool %
Ki = 70.2297;           % Value of Ki calculated using pidtool %
Kd = 0.10659;           % Value of Kd calculated using pidtool %
Cs = pid(Kp, Ki, Kd);

% Calculating the transfer function with the controller %
sys_cl = feedback(Cs*Speed_contr, 1);
figure;
bode(sys_cl, logspace(0,2))
margin(sys_cl)
grid on;
figure;
step(sys_cl);
title('Step Response of the plant (Wm(s) / Va(s)) with Speed Controller');
grid on;

% Using different equations to model transfer function relating motor %
% torque with the position of the linear slide %

position_contr = feedback(1/(s^2*M*R_rad), 1);
figure;
step(position_contr, 1:10);
title('Step Response of the plant (Tm(s) / X(s)) without Position
Controller');
grid on;

% Kp, Ki and Kd for position controller %
Kp = 20.086;
Ki = 3.4079;
Kd = 3.5132;
C = pid(Kp, Ki, Kd);
```

```matlab
% Calculating the transfer function with the controller %
sys_cl = feedback(C*position_contr, 1);
figure;
bode(sys_cl, logspace(0,2))
margin(sys_cl)
grid on;
figure;
step(sys_cl);
title('Step Response of the plant (Tm(s) / X(s)) with Position Controller');
grid on;
```