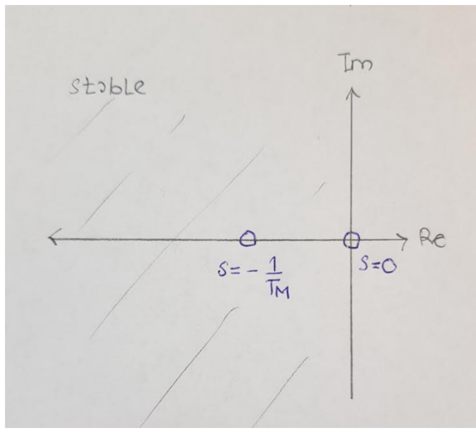
Exercise 1 - Control Theory - Linear systems

We have a motor transfer function as given below

$$\frac{X(s)}{U(s)} = \frac{1}{s(1 + T_M s)}$$

`a) Analyze the stability of this system by finding the poles of the system`

The poles of the system has two values where $s = 0$ and $s = -\frac{1}{T_M}$.



Since poles is taking place in the left hand side, the system is stable. But $s = 0$ make the system marginally stable.

`b) Draw a block diagram of the system by only using basic blocks (i.e. constants and integral blocks)`

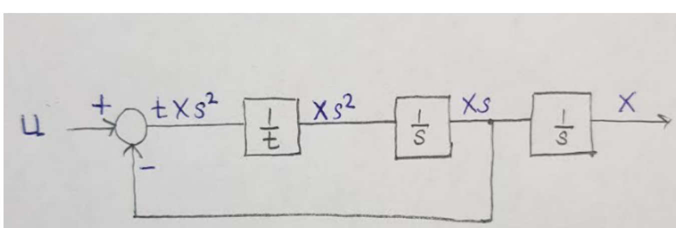
$$\frac{X}{U} = \frac{1}{s(1 + T_M s)}$$

$$Xs(1 + ts) = U$$

$$Xs + tXs^2 = U$$

$$tXs^2 = (U - Xs)$$

$$Xs^2 = \frac{1}{t}(U - Xs)$$



`c) The order of the system with and without the controller`

Without the controller, we have the transfer function as

$$\frac{X}{U} = \frac{1}{s(1 + T_M s)}$$

And the set-point would be,

$$X = \frac{U}{s(1 + T_M s)}$$

$$X = \frac{U}{T_M s^2 + s}$$

The poles at the denominators show that the system have second-order differential equation.

But, we are now having to include the PD controller, where

$$U(s) = K(1 + T_D s)E(s)$$

Where $E(s) = R(s) - X(s)$. Then we get,

$$U(s) = (K + KT_D s)(R - X)$$

We had the set-point equation,

$$X = \frac{U}{T_M s^2 + s}$$

Then we substitute the equation with U values that we get as we include PD controller,

$$X = \frac{(K + KT_D s)(R - X)}{T_M s^2 + s}$$

$$XT_M s^2 + Xs + KX + KT_D Xs = (K + KT_D s)R$$

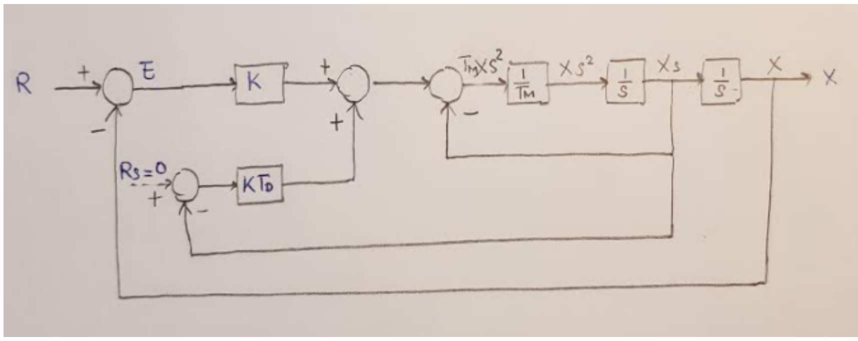
$$X(T_M s^2 + (1 + KT_D)s + K) = (K + KT_D s)R$$

$$X = \frac{(K + KT_D s)R}{T_M s^2 + (1 + KT_D)s + K}$$

The poles at the denominators show that the system with the PD-controller also have second-order differential equation.

d) Extend the block diagram with the controller and assume that R is constant, i.e. Rs = 0.
We have the set-point equation from (b) and make som extenstion as below

$$X = \frac{(K + KT_D s)R}{T_M s^2 + (1 + KT_D)s + K}$$



e) Find the poles and zeros of the system as an expression. Plot the poles and zeros in the complex plane for different values of K (using for instance MATLAB), assuming that TM = 2 and TD = 1. The resulting plot is an approximation of the root locus plot for the system. Comment the plot with regards to stability and oscillations.

Exercise 2 - Control Theory - Lyapunov direct method

Verify that the system

$$\dot{x} = -y - x^3$$

$$\dot{y} = x - y^3$$

is stable using the Lyapunov direct method. Use the following Lyapunov candidate function

$$V(x, y) = x^2 + y^2$$

In order to check that the system is stable using Lyapunov method, we have to differentiate $V(x, y) = x^2 + y^2$

$$\frac{\partial V}{\partial t} = 2x\dot{x} + 2y\dot{y}$$

$$= 2x(-y - x^3) + 2y(x - y^3)$$

$$= -2xy - 2x^4 + 2xy - 2y^4$$

$$= -2x^4 - 2y^4$$

The system is stable $\forall (x, y \neq (0, 0)$

Exercise 3 - Centralized Joint Space Motion Control - Simulink

`a) The full block diagram of the model in (6) and the controller`

$$B\ddot{q} + C\dot{q} + F_v\dot{q} + g = \tau \tag{1}$$

$$B\ddot{q} = \tau - C\dot{q} - F_v\dot{q} - g$$

$$\ddot{q} = B^{-1}(\tau - C\dot{q} - F_v\dot{q} - g)$$

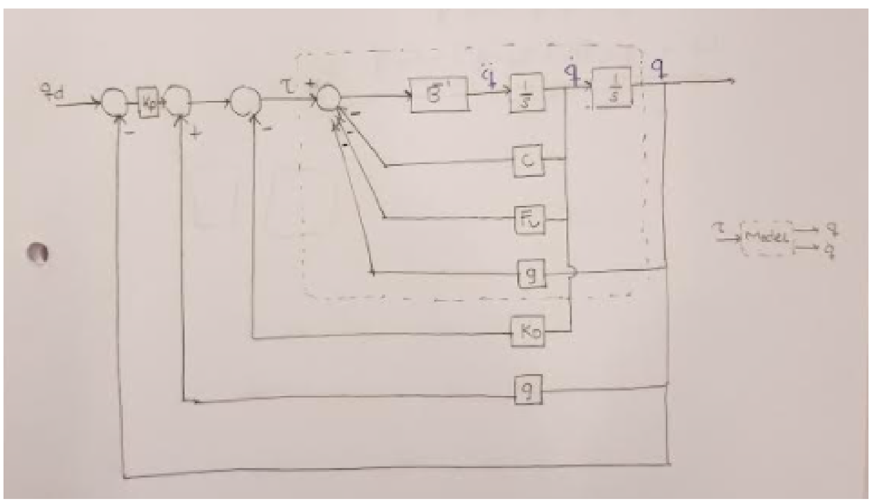
where the controller (8.51) in the text book is given,

$$u = g + K_P\tilde{q} - K_D\dot{q}$$

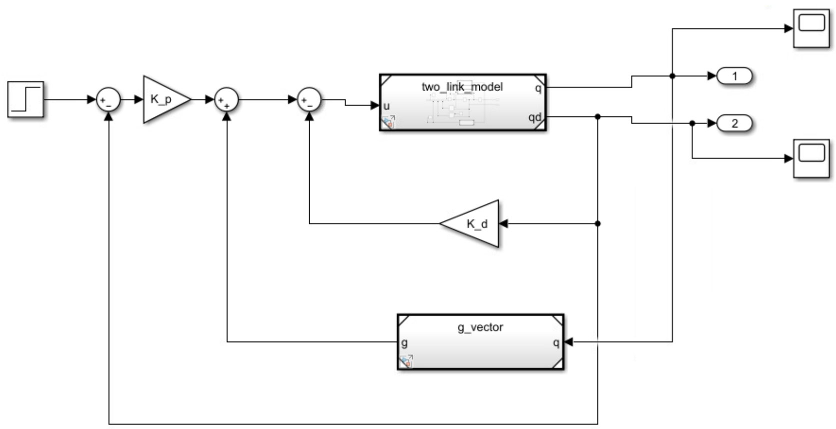
$$u = g + K_P(q_d - q) - K_D\dot{q}$$

where τ = u.

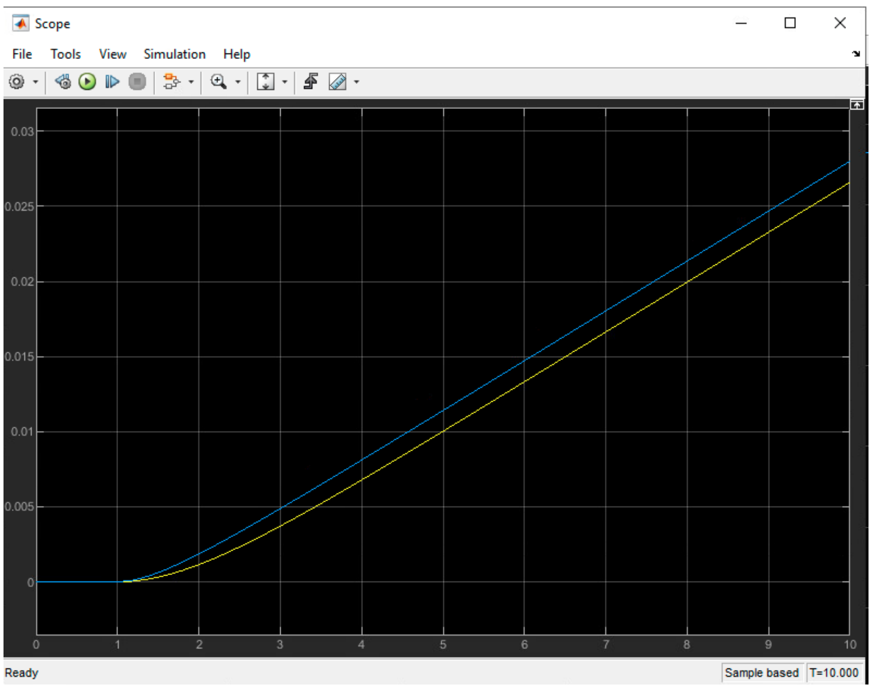
The full block diagram of the dynamic model of the manipulator and the controller is shown below

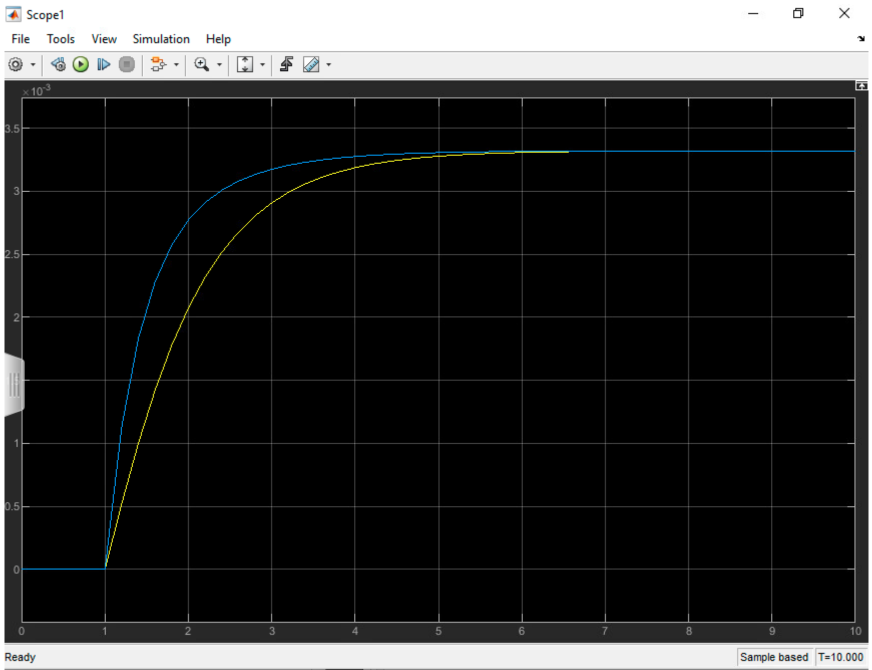


`b) Simulate the control system in simulink where the step response` q_d `from` $[0,0]^T$ `to` $[\frac{1}{2}\pi, \frac{1}{2}\pi]^T$



As the value of $K_P = 1$ and $K_D = 300$ was given by trial, a responsive system without oscillation is shonw as follow.





Scope1
File Tools View Simulation Help
×10⁻³
3.5
3
2.5
2
1
0.5
0
0 1 2 3 4 5 6 7 8 9 10
Ready
Sample based
T=10.000

`c) As we now turn our attention to the inverse dynamics control, we can write the matrix expression of the inverse dynamics control law (u) for the system as a function of` $q, \dot{q}, \tilde{q}, \dot{\tilde{q}}$ `and` $\ddot{q}$. `using the matrix` B, C, F_v `and` g in the expression.

As we turn our attention to the inverse dynamics control, we are considering the problem of tracking a joint space trajectory. The dynamics model of manipulator in equation (1) can be rewritten as

$$B\ddot{q} + n = u \tag{2}$$

where $n = C\dot{q} + F\dot{q} + g$
We will do an exact linerization based on non-linear feedback

$$By + n = u \tag{3}$$

Inserting the equation(3) into 2, we will get

$$\ddot{q} = y \tag{4}$$

We can now find the controller for y by choosing

$$y = -K_P q - K_D \dot{q} + r \tag{5}$$

This yields,

$$\ddot{q} = -K_P q - K_D \dot{q} + r$$

$$\ddot{q}_d + K_P q_d + K_D \dot{q}_d = r \tag{6}$$

Inserting the equation(6) into equation (5), we get,

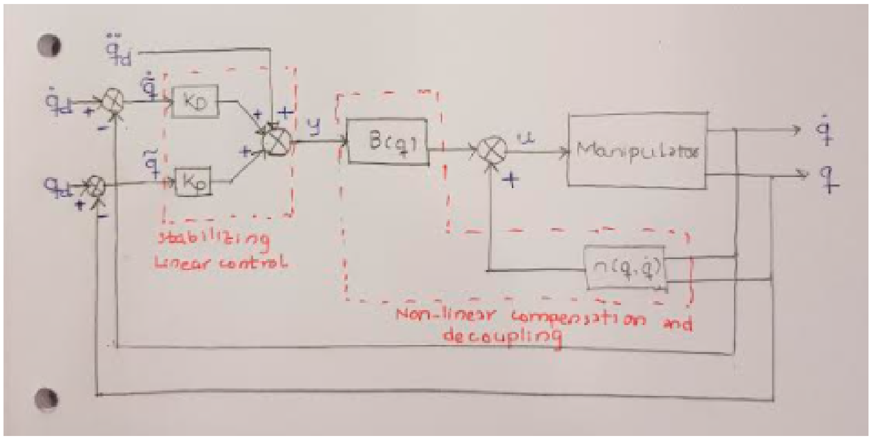
$$y = -K_P q - K_D \dot{q} + \ddot{q}_d + K_P q_d + K_D \dot{q}_d$$

$$y = \ddot{q}_d + K_D \dot{\tilde{q}} + K_P \tilde{q} \tag{7}$$

where $\tilde{q} = q_d - q$ Therefore we can express the equation(3) as below,

$$B(\ddot{q}_d + K_D \dot{\tilde{q}} + K_P \tilde{q}) + C\dot{q} + F\dot{q} + g = u \tag{8}$$

d) The full block diagram of the inverse dynamics control of the system



e) Simulate the control system in Simulink using a trajectory $q_d = [sin2\pi ft, sin2\pi ft]^T$, and gain $K_p = \text{diag}(25, 25)$, $K_D = \text{diag}(5, 5)$

$$q_d = sin(2\pi ft)$$

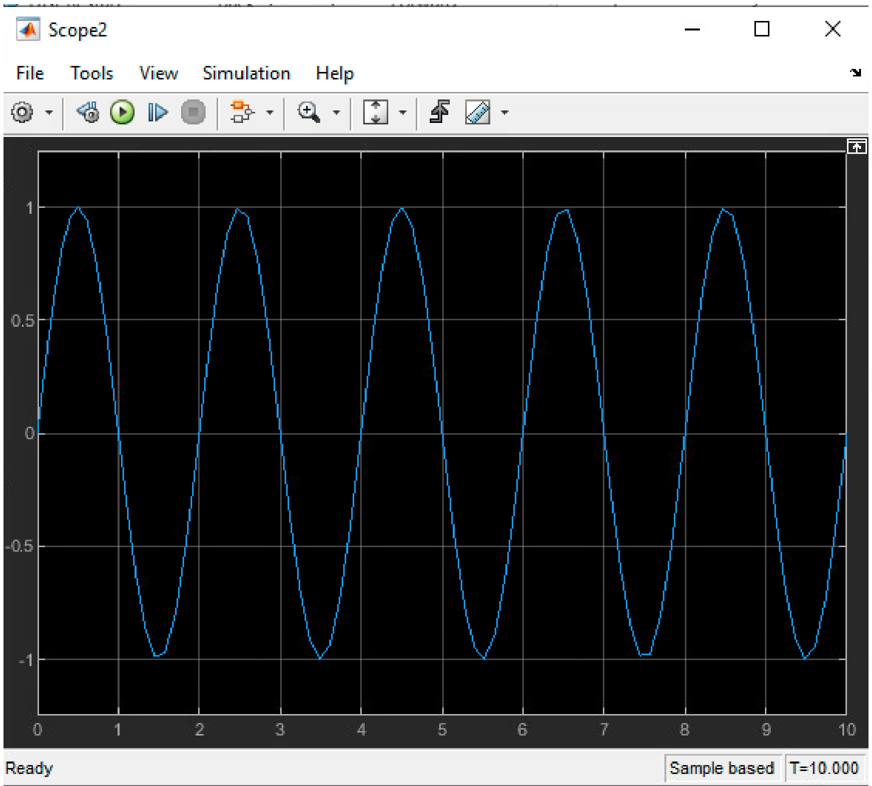
The first derivative of q_d gives the velocity of the desired position

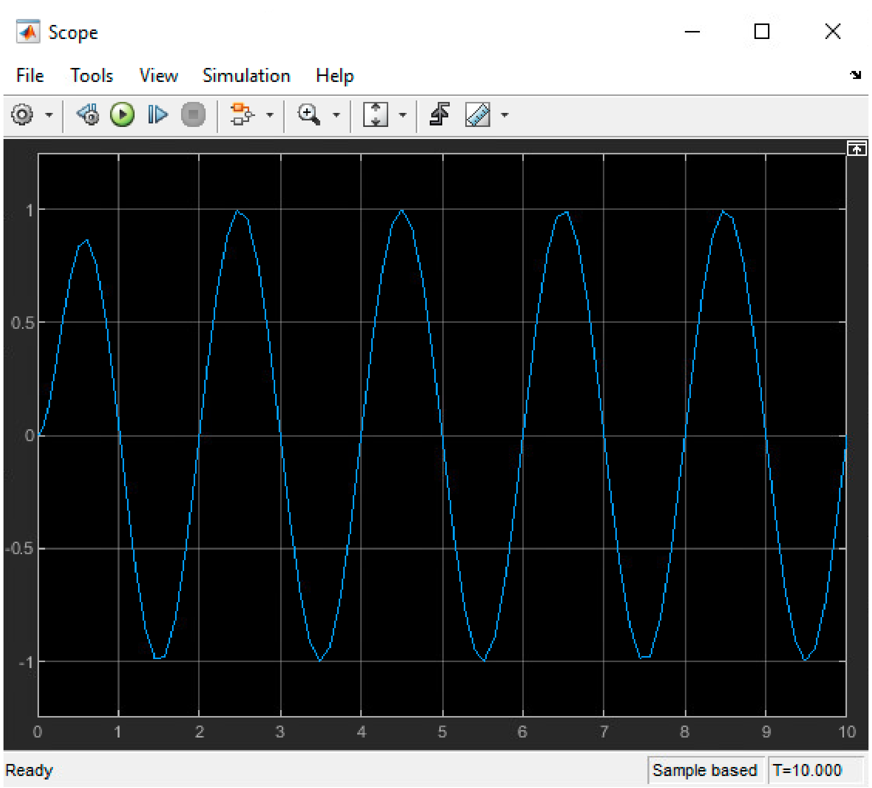
$$\dot{q_d} = 2\pi fcos(2\pi ft)$$

The second derivative of q_d gives the acceleration of the desired position

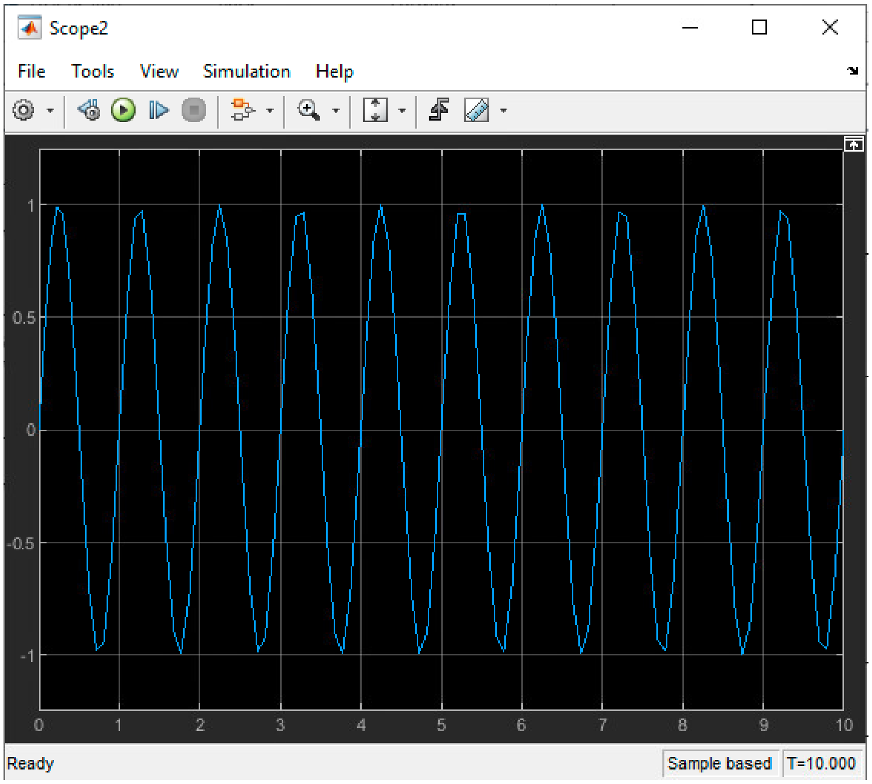
$$\ddot{q_d} = -4\pi^2 f^2 sin(2\pi ft)$$

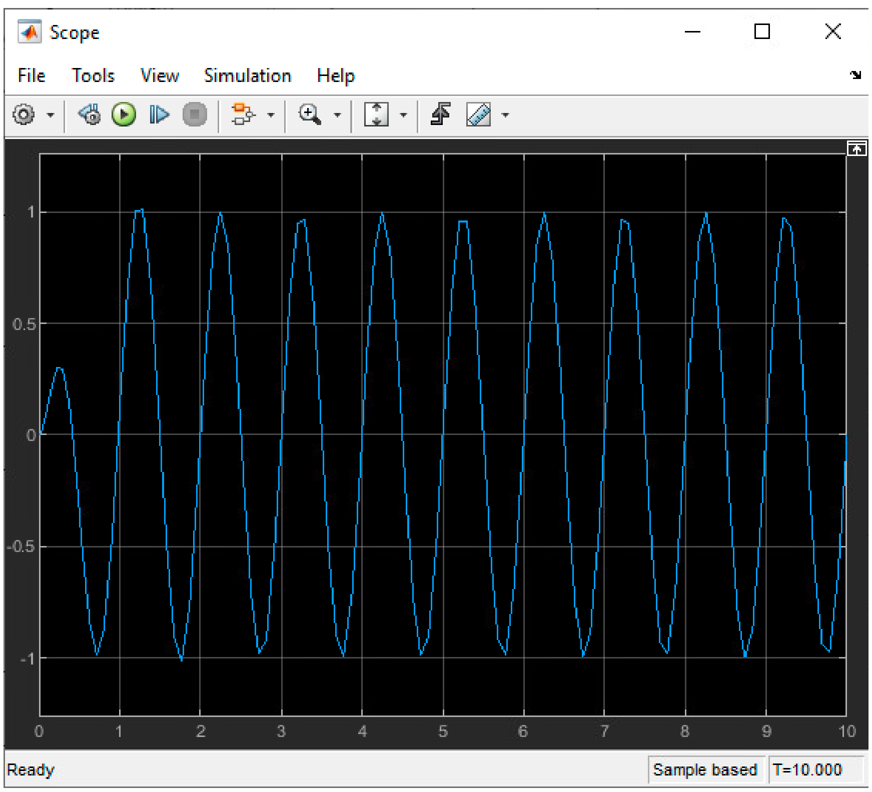
The plot of q and q_d for f = 0.5



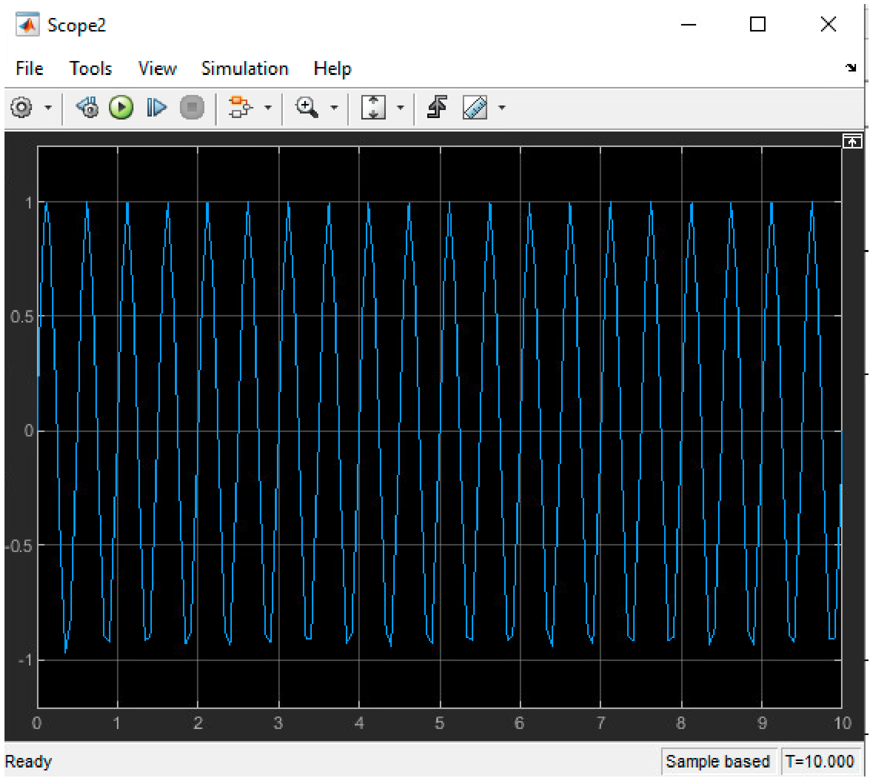


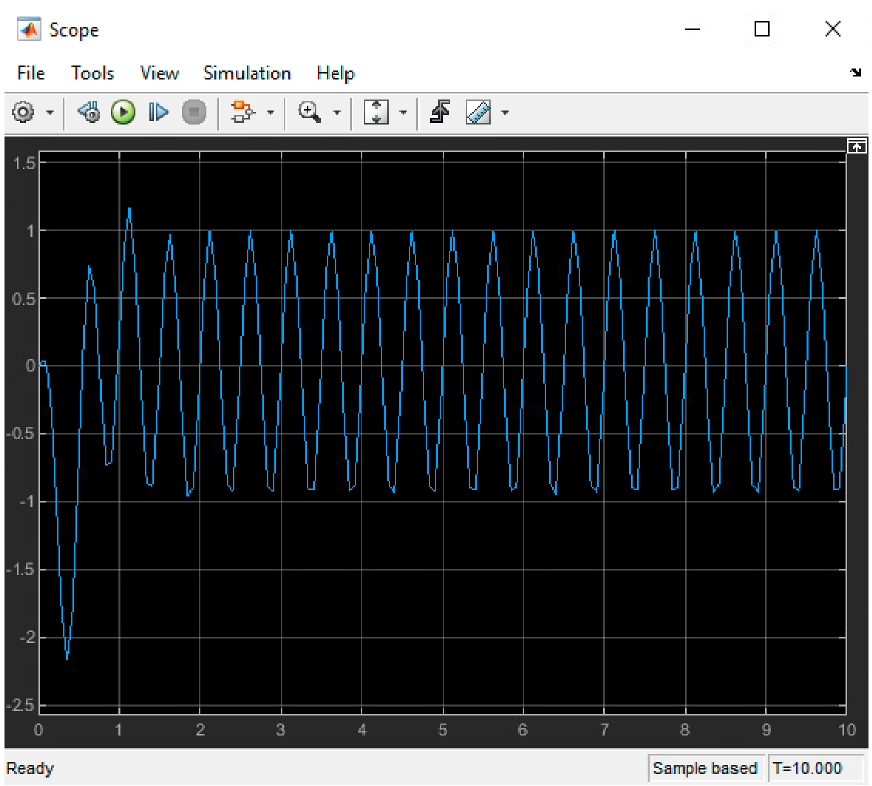
The plot of q and q_d for f = 1





The plot of q and q_d for f = 2





e)`Designing the control gains so that the system is critically damped with` $\omega_n = 5$ `for both degree of freedom.`

We had

$$K_p = 25 = \omega_n^2$$

$$\omega_n = 5$$

and we also had

$$K_d = 5$$

$$K_p = 5 = 2\zeta\omega_n$$

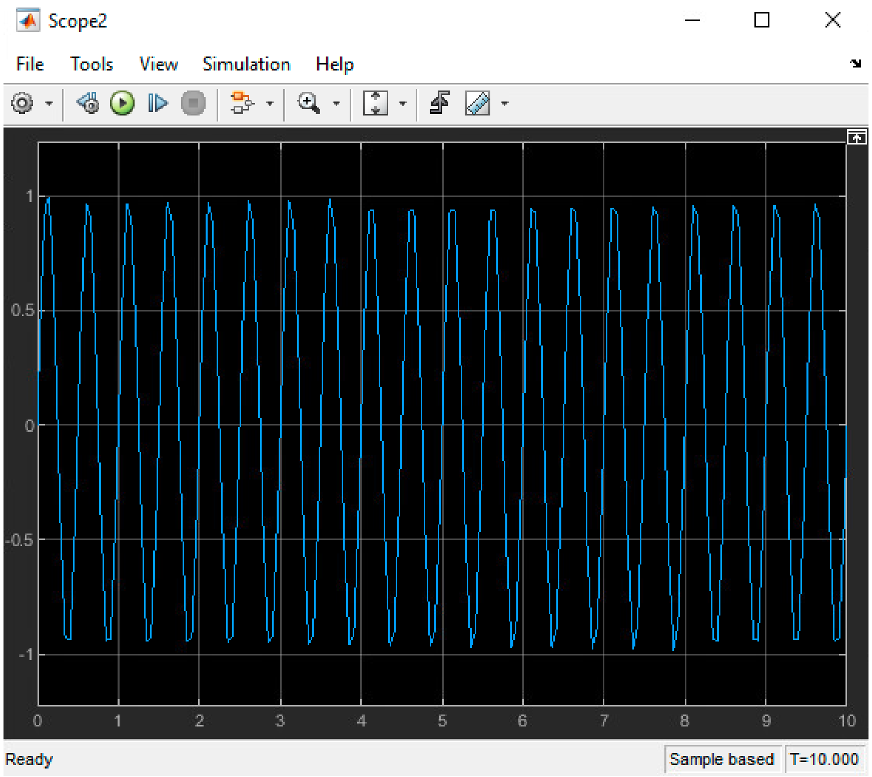
$$\zeta = \frac{1}{2}$$

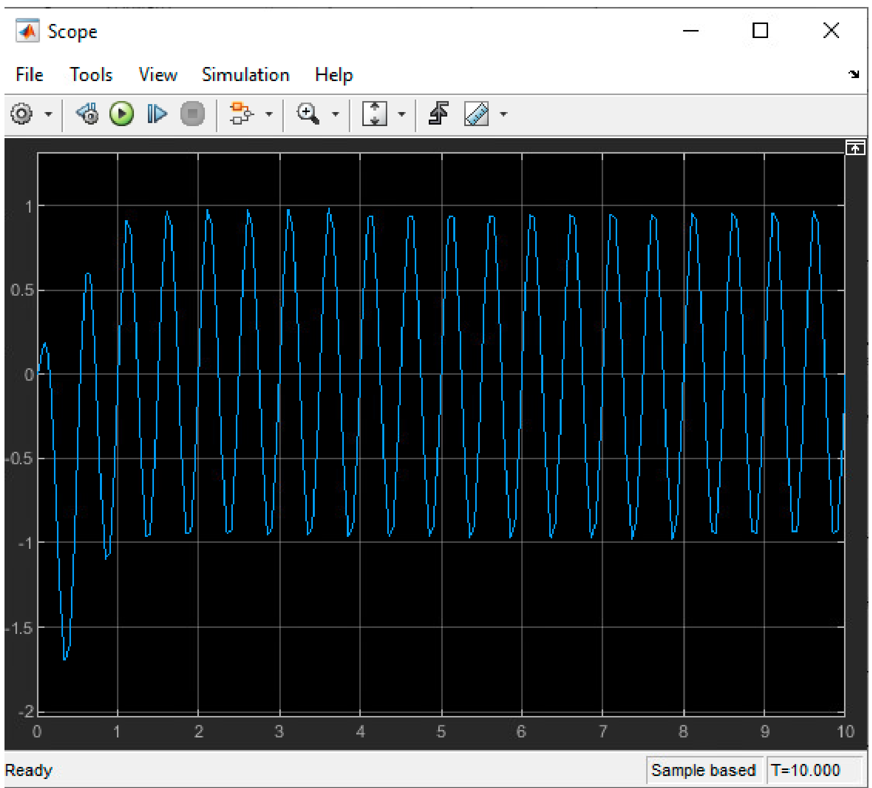
We got $\zeta = \frac{1}{2}$. Since $\zeta < 1$ means that the system is underdamped. The natural choice is $\zeta = 1$ for criticall dampted.

$$K_p = \omega_n^2 = 25$$

$$K_d = 2\varsigma\omega_n = 10$$

Now we are simulating for the new system and showing the plots of q_d and q for f = 2.

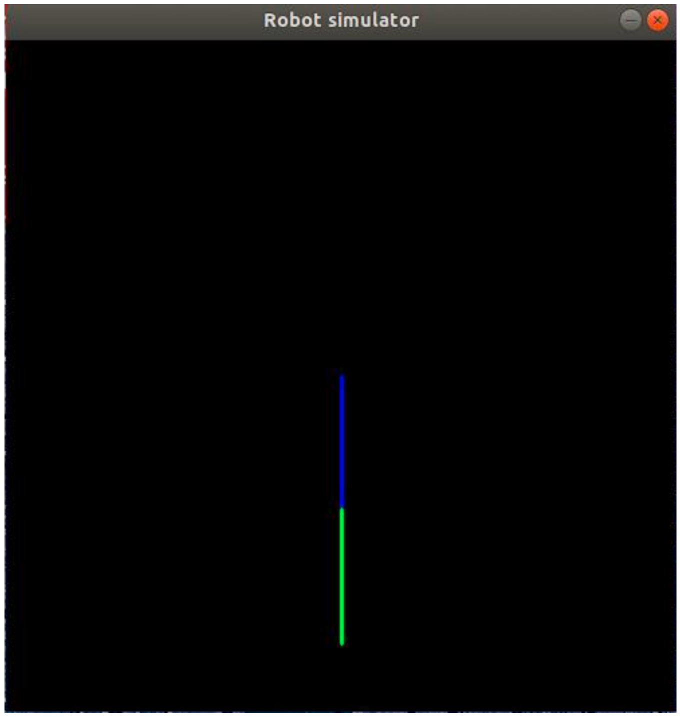




Exercise 4 - Independent Joint Control - ROS

`a)Expand the ROS nodes you made in the ROS intro tutorial to use the 2 DoF robot model`

The code is submitting in the code file.



`b)The complete dynamic model for the joint motors by combining (8.18), (8.11) from the textbook and the control input u given below.`

$$u = K_t R_a^{-1} G_v v_c \tag{9}$$

We also have the dynamic model of the manipulator as given below:

$$B(q)\ddot{q} + C(q,\dot{q})\dot{q} + F_v\dot{q} + g = \tau \tag{10}$$

As we combine the equation (8.5) and (8.6) from the textbook, we get,

$$G_v v_c = R_a i_a + K_v \dot{q_m} \tag{11}$$

Then we could insert equation (8.2) from the textbook in equation (11), we get,

$$G_v v_c = R_a i_a + K_v K_r \dot{q_m} \tag{12}$$

Then we solves for i_a,

$$i_a = R_a^{-1}(G_v v_c - K_v K_r \dot{q_m}) \tag{13}$$

Now we insert equation (13) into equation (8.4) in the text book, we get

$$\tau = K_r K_t R_a^{-1}(G_v v_c - K_v K_r \dot{q_m}) \tag{14}$$

So now we can insert this into the dynamic equation of the manipulator in equation (10) to get the total equation as shown below,

$$B(q)\ddot{q} + C(q,\dot{q})\dot{q} + F_v\dot{q} + g = K_r K_t R_a^{-1}(G_v v_c - K_v K_r \dot{q_m}) \tag{15}$$

As we are moving all joint position variable to the left side and control voltage on the right side, we get

$$B(q)\ddot{q} + C(q,\dot{q})\dot{q} + F_v\dot{q} + K_r K_t R_a^{-1} K_v K_r \dot{q_m} + g = K_r K_t R_a^{-1} G_v v_c \tag{16}$$

So now we can simplify the equation

$$B\ddot{q} + C\dot{q} + F\dot{q} + g = u \tag{17}$$

where $F\dot{q} = F_v\dot{q} + K_r K_t R_a^{-1} K_v K_r \dot{q_m}$ and $u = K_r K_t R_a^{-1} G_v v_c$

`b)Divide the model into one linear and decoupled part and one nonlinear coupled part`

Now we are going to derive linear and decoupled model of the robot and then use that model to control the robot

Based on the original dynamic model

$$B\ddot{q} + C\dot{q} + F\dot{q} + g = \tau \tag{18}$$

We have

$$K_r\dot{q} = \dot{q_m}$$
$$\dot{q} = K_r^{-1}\dot{q_m}$$
$$\ddot{q} = K_r^{-1}\ddot{q_m} \tag{19}$$

We also have

$$\tau_m = K_r^{-1}\tau$$
$$\tau = K_r\tau_m \tag{20}$$

Inserting equation (19) and (20) into equation (18), we get

$$K_r^{-1}BK_r^{-1}\ddot{q_m} + K_r^{-1}CK_r^{-1}\dot{q_m} + K_r^{-1}FK_r^{-1}\dot{q_m} + K_r^{-1}g = \tau_m \tag{21}$$

Divided B into two parts, where

$$B(q) = \bar{B} + \Delta B(q) \tag{22}$$

Now we can rewrite the dynamic equation to become a linear and decoupled system,

$$K_r^{-1}BK_r^{-1}\ddot{q_m} + F_m\dot{q_m} + d = \tau_m \tag{23}$$

where $F_m = K_r^{-1}F_vK_r^{-1}$

and $d = K_r^{-1}\Delta BK_r^{-1}\ddot{q_m} + K_r^{-1}CK_r^{-1}\dot{q_m} + K_r^{-1}g$ as nonlinear and coupled where we treats all this terms as disturbance and it depends on all the joint variable

Now we are going to look at the independent joint control using position feedback, as described on page 312 in the textbook.

d)Drawing the full block diagram of the linear decoupled part and the controller. The coupled part should come in as a disturbance.

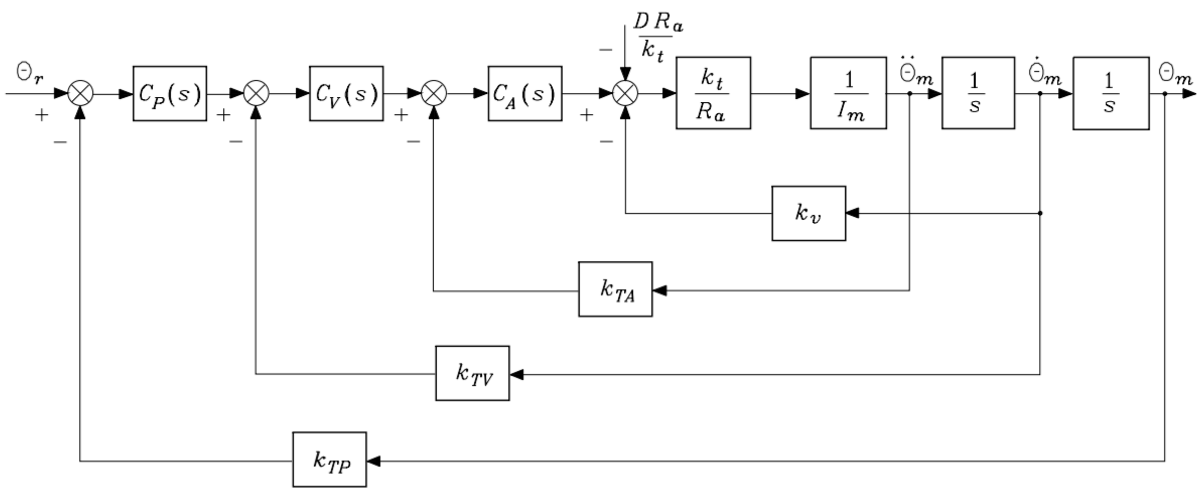


Fig. 8.6. Block scheme of general independent joint control

e)Transforming the control law to the time domain, in order to be able to implement the controller in C++

From equation (8.6) from textbook, position feedback controller is given,

$$C_p(s) = K_p \frac{1 + sT_p}{s} \tag{24}$$

where $C_v(s) = 1$, $C_A = 1$ and $K_{TV} = K_{TA} = 0$ In our case, the error is define as,

$$E(s) = \Theta_r - \Theta K_{TP} \Theta_m$$

and

$$U(s) = C(s)E(s)$$
$$U(s) = (K_p \frac{1 + sT_p}{s})E(s)$$
$$U(s) = (\frac{K_p}{s} + K_p T_p)E(s)$$
$$U(s) = K_p \frac{E(s)}{s} + K_p T_p E(s)$$

and making the equation Laplace transform, we get

$$U(t) = K_p \int e(t)dt + K_p T_p e(t)$$

f) Implement the controller in ROS by overwriting the controller already there. Have a look at the webpage below for tips on how to implement integration into your code

Implementing integration

Example on how to implement integration. The variable `e_i_m` is the integral of the error `e_m`. The integral is found by multiplying with the time period `dt`.

```
ros::Rate loop_rate(100);
double dt = 1.0/100.0;

Eigen::Vector2d e_i_m(0.0, 0.0);

while (ros::ok())
{
  Eigen::Vector2d q_r(0.25*M_PI*k_r_1, -0.25*M_PI*k_r_2);
  Eigen::Vector2d q_m = sim.getMotorPosition();

  Eigen::Vector2d e_m = q_r - q_m;
  e_i_m += dt*e_m;
}
```

g) Simulating the system and try to find good values for K_P and T_P. Include a plot of joint angles and the values you found in your report. Set the joint set point to $(\frac{1}{4}\pi, -\frac{1}{4}\pi)$. Remember that these values must be converted to motor positions before using them. This can be done by multiplying with K_r.