Exercise 1 - Force control simulink

`a) Expand the manipulator model in Simulink by including the model of the environment`

we can expand the manipulator model in Simulink by including the model of the environment by using the equation (9.1) in textbook as below,

$$B(q)\ddot{q} + C(q, \dot{q} + F_v\dot{q} + g(q) = u - J^T(q)h_e$$

In order to calculate the force from the environment, we needs to find the end-effector postion.

We can get enf-effector by finding forward kinematics of the two-link manipulator. In chapter 2.9.1 in the textbook, we could get the forward kinematics of the two-link manipulator by setting $\theta_3 = 0$ and $a_3 = 0$

$$T_2^0 = \begin{bmatrix} c_{12} & -s_{12} & 0 & a_1c_1 + a_2c_{12} \\ s_{12} & c_{12} & 0 & a_1s_1 + a_2s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Therefore, the forward kinematic be like this,

$$ForwardKinematic = \begin{bmatrix} a_1c_1 + a_2c_{12} \\ a_1s_1 + a_2s_{12} \\ 0 \end{bmatrix}$$

`b) Before implementing any controllers we will need the analytic Jacobian for the two-link planar arm`

In Section 3.2.1 we find the geometric Jacobian of a three-link planar arm robot. By setting $a_3 = 0$, we will get the geometric Jacobian of a two-link planar arm robot.

$$J = \begin{bmatrix} -a_1s_1 - a_2s_{12} & -a_2s_{12} & 0 \\ -a_1c_1 - a_2c_{12} & a_2c_{12} & 0 \end{bmatrix}$$

In order to implement any controllers, we need to find the analytic Jacobian for the two-link planar arm. But for simplicity in this case, disregarding the rotation around the z-axis. Therefore the manipulator in this exercise it can be shown that the geometric and analytic Jacobian are equal.

c) In order to simulate the active compliance control law, we can use the equation (9.15) in the textbook in Simulink.

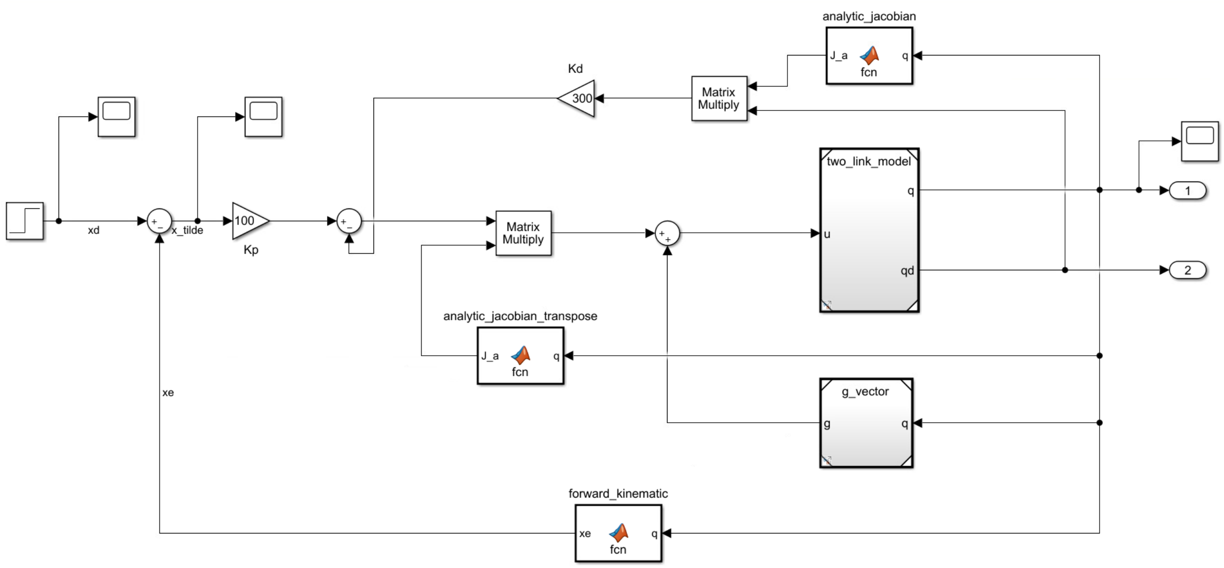
$$u = g(q) + J_{Ad}^T(q, \tilde{x})(K_p\tilde{x} - K_D J_{Ad}(q, \tilde{x})\dot{q})$$

By using the values which is giving in the task as

$x_r = 0.5$,

$q(t = 0) = [0.75\pi, -0.75\pi]^T x_d = [0.75, 0.25]^T$

$k_x = 100$, where we can simulate by varying the value of K_p and K_d.

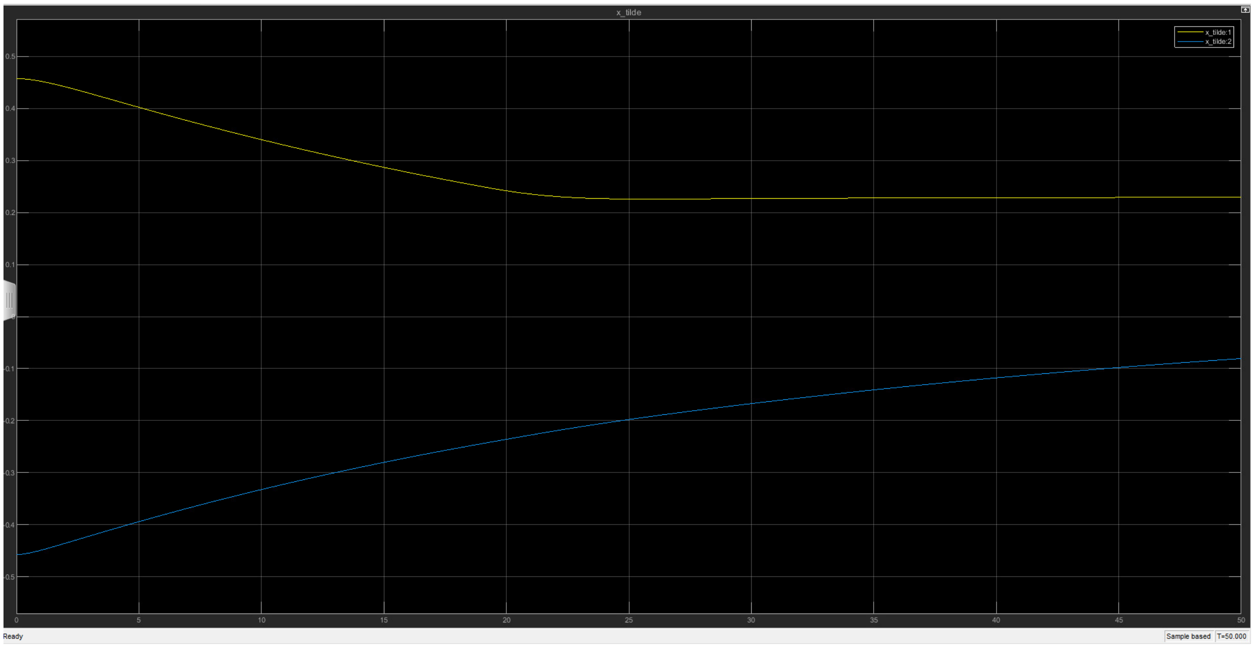


As it is written in example (9.1), the arm stiffness and environment stiffness influence the resulting equilibrium configuration. In the case, where

$$\frac{K_p}{k_x} << 1$$

$x_e \approx x_r$, the end effector position along the x-axis is almost equal to the environment rest position.

At this point, we can simulate to show that the above statement, we set the value of as $K_p = 10$ and the figure shows as below

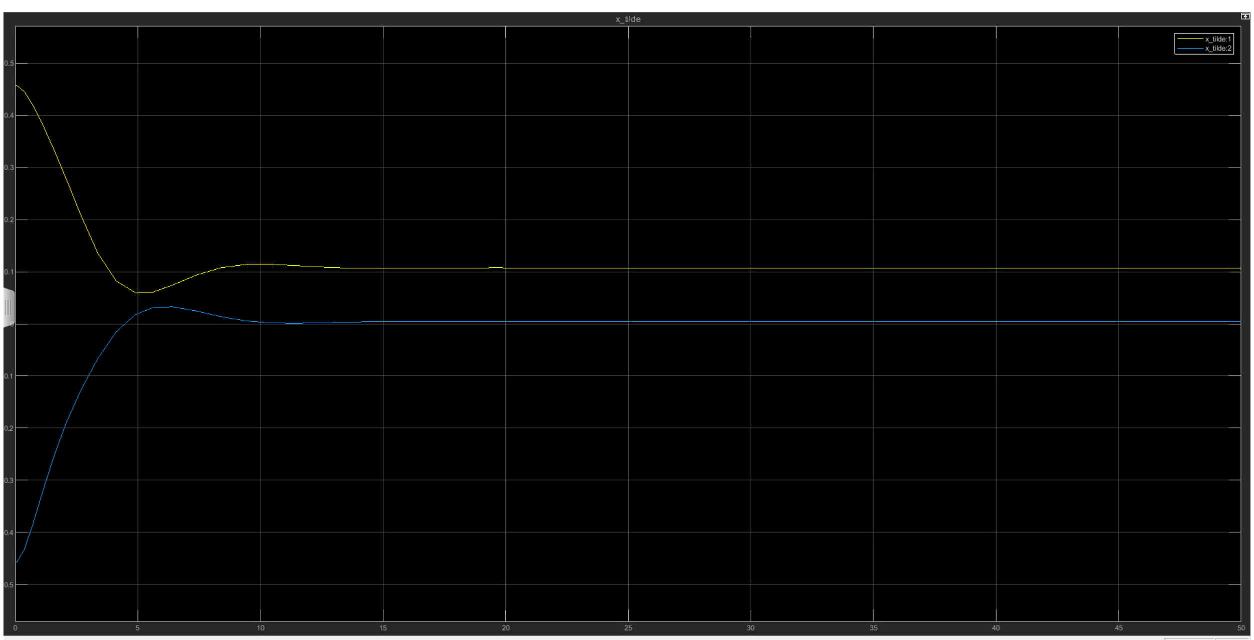


In the case, where

$$\frac{K_p}{k_x} >> 1$$

$x_e \approx x_r$, the end effector position along the x-axis is almost equal to the desired position and stabilizing at near zero stage.

At this point, we can simulate to show that the above statement, we set the value of as $K_p = 150$ and the figure shows as below



`d) In order to implement impedance control we will need the time derivative of the analytic Jacobian`

We can find the the time derivative of the analytic Jacobian ($\dot{J}_A$), based on the 2x2 analytic Jacobian used in c).

$$J_A = \begin{bmatrix} sin(\Theta_1) - sin(\Theta 1 + \Theta 2) & -sin(\Theta 1 + \Theta 2) \\ cos(\Theta_2) + cos(\Theta 1 + \Theta 2) & cos(\Theta 1 + \Theta 2) \end{bmatrix}$$

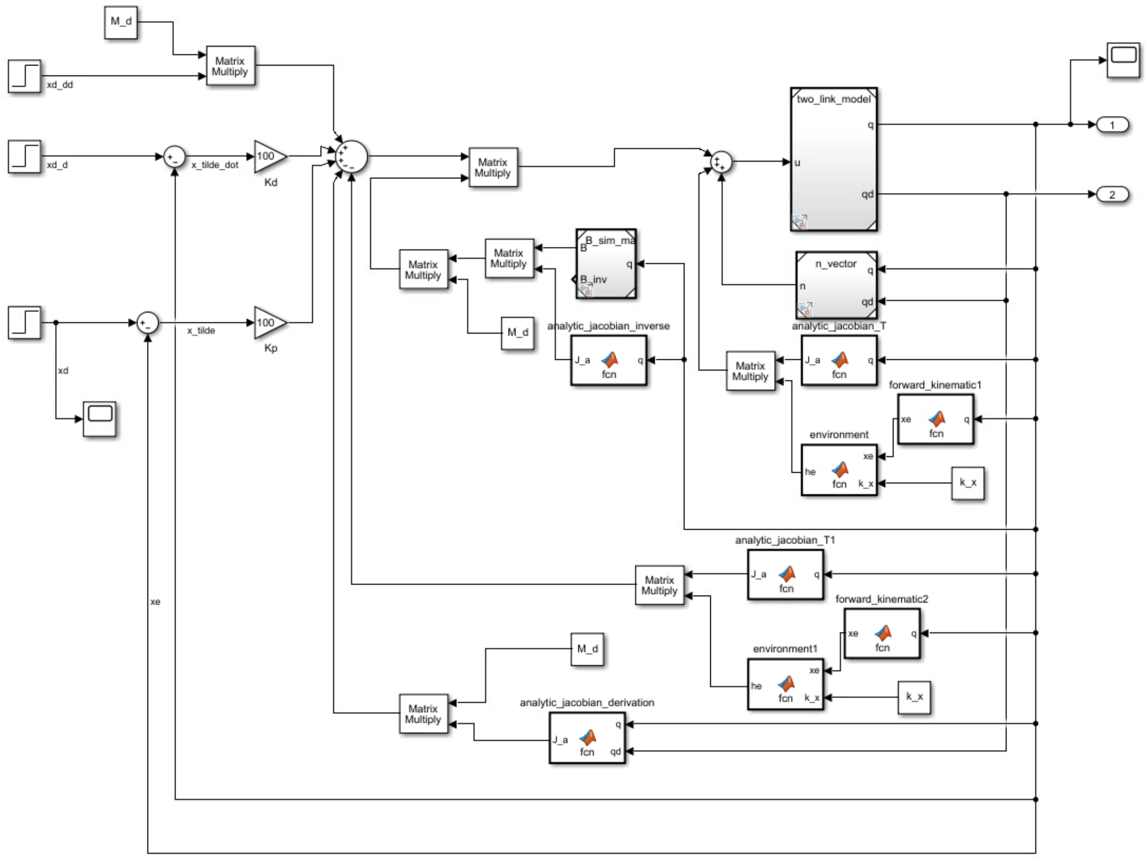
$$\dot{J}_A = \begin{bmatrix} \dot{\Theta}_1 cos(\Theta_1) - (\dot{\Theta}_1 + \dot{\Theta}_2) cos(\Theta 1 + \Theta 2) & -(\dot{\Theta}_1 + \dot{\Theta}_2) cos(\Theta 1 + \Theta 2) \\ -\dot{\Theta}_2 sin(\Theta_2) - (\dot{\Theta}_1 + \dot{\Theta}_2) sin(\Theta 1 + \Theta 2) & -(\dot{\Theta}_1 + \dot{\Theta}_2) sin(\Theta 1 + \Theta 2) \end{bmatrix}$$

e) `Simulate the impedance control law in (9.30) and (9.31) in Simulink`

The impedance control in (9.30) and (9.31) in textbook, we have

$$u = B(q)y + n(q, \dot{q}) + J^T(q)h_e$$

$$y = J_A^{-1}(q)M_d^{-1}(M_d\ddot{x}_d + K_D\dot{\tilde{x}} + K_p\tilde{x} - M_d\dot{J}_A(q, \dot{q})\dot{q} - h_A)$$

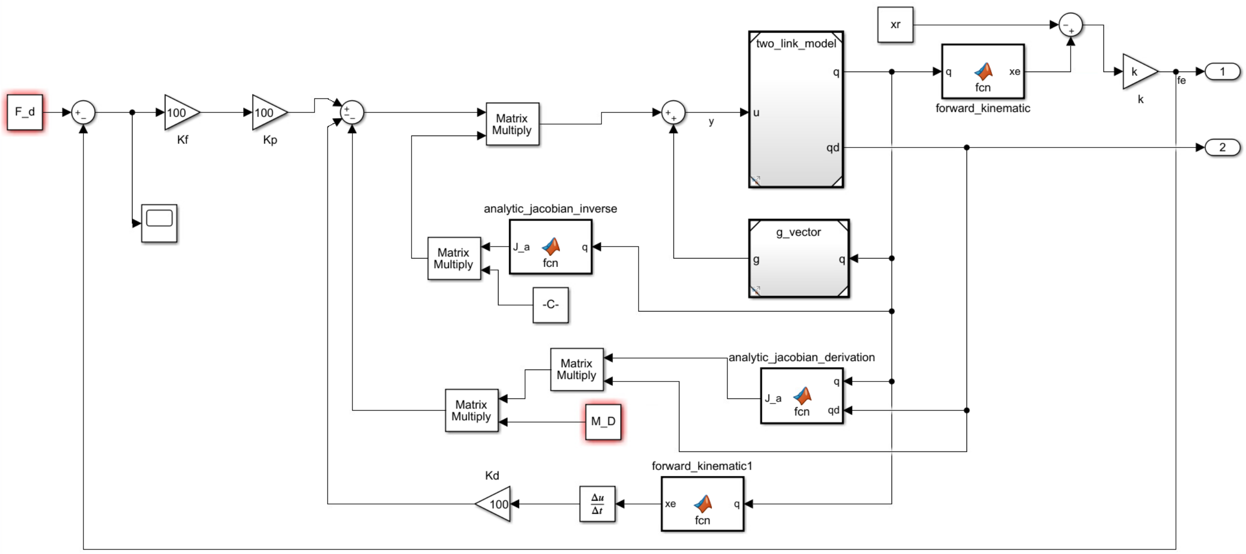


f) Try varying Md, KD and KP individually `g) Simulate the force control with inner velocity loop control law (9.45)`

The force control can be found in equation (9.45) in textbook.

$$y = J^{-1}(q)M_d^{-1}(-K_D\dot{x}_e + K_P x_F - M_d\dot{J}(q,\dot{q})\dot{q})$$

where $x_F = K_F(f_d - f_e)$ and the dynamics,



Exercise 3 - Visual servoing

`a) The principal difference between image-based and position-based visual servoing`

Position-based visual servoing uses operational space and reconstruct the target relative to pose, meaning that we can use the visual sensor or camera to estimate the position of the object that we want to interact with and uses that information to control the position of the robot operational space where we have this estimation step using camera in the control loop.

Image-based visual servoing on the other hand is that we extract a certain key features in the image and uses those key features directly into the control loop so that we have the error that we want to control or minimize the error.

`b) The benifit of using resolved velocity control when doing visual servoing`

There is some kind of problems regarding block scheme in Fig.10.14 in the text book where it has all the parameters in the same control equations, meaning the estimate has a lower update frequency based on the camera and as we know the control of the robot usually operate at a much higher frequency.
By using resolved velocity control in which we separate the control at a low level control of the robot and the visual servoing since they operates at different frequency. As we operates at a lower frequency of the controller, we also lower the performance of the controller. In that case, we have to have a low gain since we have a less update frequency.

`c) What does the image Jacobian and the interaction matrix represents?`

Image Jacobian is an expressiion that relates the absolute velocity of the camera v_c^c to the image plane velocity $\dot{s}$ ($\dot{s} = J_s V_c^c$) whereas the interaction matrix describes also the relationship between the velocity of the camera (held by the manipulator) and the image features. But the interaction matrix can be expressed by multiplying by the image Jacobian with $\Gamma(O_{c,o}^c)$, meaning $L_s(\text{the interactionmatrix}) = J_s \Gamma(O_{c,o}^c)$, where $\Gamma(O_{c,o}^c)$ represents

$$\Gamma(O_{c,o}^c) = \begin{bmatrix} -I & S(O_{c,o}^c) \\ 0 & -I \end{bmatrix}$$

`d) Prove the stability of the image-based visual servoing control law (10.84). This means showing the steps between (10.80), (10.81) and (10.85).`

Image-based visual servoing can be implemented using a PD control with gravity compensation defined on the basis of the image space error.
In Lynpunov direct method, we need to find the function, V and used that to prove the system stable,:

$$V(\dot{q}, e_s) = \frac{1}{2}\dot{q}^T B(q) q + \frac{1}{2} e_s^T K_p e_s$$

Now we have to make sure that our system is stable by using Lyapunov method, so first we needs to differentiate our system.

$$\dot{V}(\dot{q}, e_s) = \frac{1}{2}\dot{q}^T B(q)\dot{q} + \frac{1}{2}e_s^T K_p e_s$$

$$\dot{V}(\dot{q}, e_s) = \frac{1}{2}\ddot{q}^T B(q)\dot{q} + \frac{1}{2}\dot{q}^T \dot{B}(q)\dot{q} + \frac{1}{2}\dot{q}^T B(q)\ddot{q} + \frac{1}{2}\dot{e}_s^T K_p e_s + \frac{1}{2}e_s^T K_p \dot{e}_s$$

$$\dot{V}(\dot{q}, e_s) = \frac{1}{2}\dot{q}^T B^T(q)\ddot{q} + \frac{1}{2}\dot{q}^T \dot{B}(q)\dot{q} + \frac{1}{2}\dot{q}^T B(q)\ddot{q} + \frac{1}{2}\dot{e}_s^T K_p e_s + \frac{1}{2}\dot{e}_s^T K_p e_s$$

$$\dot{V}(\dot{q}, e_s) = \dot{q}^T B^T(q)\ddot{q} + \frac{1}{2}\dot{q}^T \dot{B}(q)\dot{q} + \dot{e}_s^T K_p e_s$$

$$\dot{V}(\dot{q}, e_s) = \dot{q}^T B(q)\ddot{q} + \frac{1}{2}\dot{q}^T \dot{B}(q)\dot{q} + \dot{e}_s^T K_p e_s \tag{1}$$

where $B^T(q) = B(q)$ and $\dot{e}_s^T = \dot{e}_s$ since they are symmatric matrixs.
Recall from (8.7) from the textbook, we had

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

and solve it for $B(q)\ddot{q}$ and insert it into equation(1), we get

$$B(q)\ddot{q} = -C(q, \dot{q} - F_v\dot{q} - g(q) + u$$

Then we can insert it into equation(1),

$$\dot{V}(\dot{q}, e_s) = \dot{q}^T(-C(q, \dot{q} - F_v\dot{q} - g(q) + u) + \frac{1}{2}\dot{q}^T \dot{B}(q)\dot{q} + \dot{e}_s^T K_p e_s$$

$$\dot{V}(\dot{q}, e_s) = \frac{1}{2}\dot{q}^T(\dot{B}(q) - 2C(q, \dot{q}))\dot{q} - \dot{q}^T F_v\dot{q} + \dot{q}^T(u - g(q)) + \dot{e}_s^T K_p e_s$$

From the dynamic of the manipulator, we have

$$\dot{q}^T(\dot{B}(q) - 2C(q, \dot{q}))\dot{q} = 0$$

Then we can simplify the equation as below.

$$\dot{V}(\dot{q}, e_s) = -\dot{q}^T F_v\dot{q} + \dot{q}^T(u - g(q)) + \dot{e}_s^T K_p e_s$$

Since $\dot{s}_d = 0$ and the object is fixed with respect to the base frame, we have

$$\dot{e}_s = -s = -J_l(s, z_c, q)\dot{q}$$

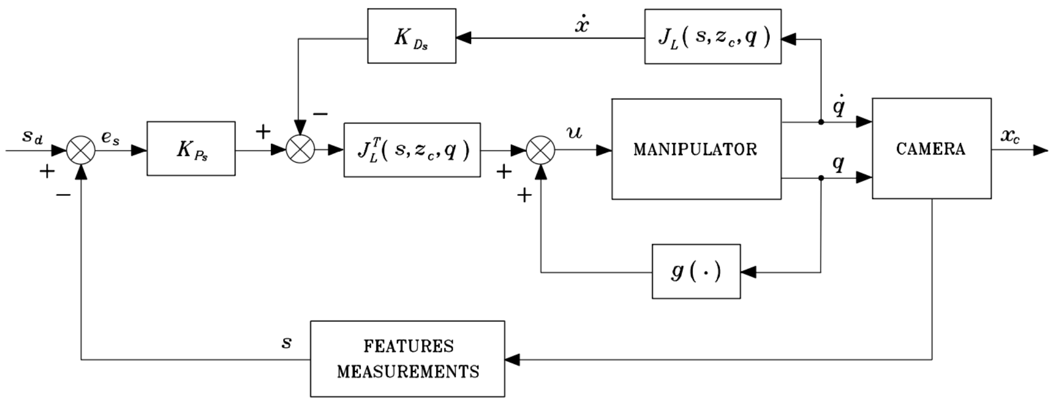


Fig. 10.16. Block scheme of image-based visual servoing of PD type with gravity compensation

where $J_L(s, z_c, q) = L_s(s, z_c) \begin{bmatrix} R_c^T & 0 \\ 0 & R_c^T \end{bmatrix} J(q)$ the camera frame and the end effector frame being coincident, therefore

$$u = g(q)J_L^T(s, z_c, q)(K_{Ps}e_s - K_{Ds}J_L(s, z_c, q)\dot{q})$$

Where K_Ds is a symmetric and positive definite (k x k) matrix, becomes

$$\dot{V}(q, \dot{e}_s) = -\dot{q}^T F\dot{q} - \dot{q}^T J_L^T K_{Ds} J_L \dot{q}$$

Exercise 4 - Tele-operations

`a) Explain the difference between unilateral and bilateral tele-operation`

Unilateral tele-operation a one way communication where the operator controls a robot 'master' at a distance and the robot 'slave' operates at a remote environment accordingly.

Bilateral tele-operation is two ways communication where the operator controls a robot 'master' at a distance and the robot 'slave' operates at a remote environment. But bilateral tele-operation is a two way communication while exchanging action and reaction of information between a master and slave in a real time through the communication medium.

`b) What does the hybrid matrix represent, and what is a transparent tele-operation system like?`

To achieve ideally transparent bilateral teleoperation system in the presence of time delay, the hybrid matrix is used to represent the linearized behavior around contact operating points.

A transparent tele-operation system be like for the same force $f_m = f_e$, where the force applies at the master is equal to the force applies at the remote environment, and we want the same velocity at the master and slave conditions at the same time.

`c) Briefly explain the forward-flow tele-operation controller. Is it unilateral or bilateral?`