INF3480/INF4380 Exam Solution (Spring 2018)

Task 1 [INF3480 - Bachelor Level] (15%):

a) (5%) In the context of Robot Operating System (ROS) and "MoveIt!", there were five types of motion planning kinematic constraints discussed. Please list all five types together with a brief explanation about each one.

Answer:
- **Position constraints** – restrict the position of a link to lie within a region of space
- **Orientation constraints** – restrict the orientation of a link to lie within specified roll, pitch or yaw limits
- **Visibility constraints** – restrict a point on a link to lie within the visibility cone for a particular sensor
- **Joint constraints** – restrict a joint to lie between two values
- **User-specified constraints** – you can also specify your own constraints with a user-defined callback.

b) (5%) Robot Operating System (ROS) messages are known to be language agnostic. What does it mean? If you were asked to code in a custom ROS message, which would include the robot name as well as starting and finishing points in a two-dimensional map, how would the .msg file look?

Answer:
Language agnostic means that ROS messages are not defined to any programming language. This means that C++ can talk to Python nodes through messages.

Custom ROS message (.msg):
string robotName
uint32 (or float, or int, or double - any numerical variable type) startX
uint32 (or float, or int, or double - any numerical variable type) startY
uint32 (or float, or int, or double - any numerical variable type) goalX
uint32 (or float, or int, or double - any numerical variable type) goalY

c) (5%) One of the main advantages of Robot Operating System (ROS) is modular design. What is it and why is it beneficial when designing a robotic system with many hardware components?

Answer guidelines:
Modular design allows to build a flexible system combined of many modules (nodes). Each node performs a specific task and communicates with other nodes through topics (using messages) to make the whole system run. Having modular design allows us to make changesto

some nodes without affecting the rest of the system, for example exchange the sensors or add new ones, even during runtime of the system.

Things to be mentioned in the answer:
- Flexible design
- Change or add new nodes (modules) without affecting the rest of the system
- Easily interchange hardware components if needed
- Only parts of algorithm can be modified
- Speeds up the development by reusing existing nodes that suit our system
- Makes system more scalable
- Computational heavy algorithms/systems can be split to run on multiple machines

Task 1 [INF4380 - Master Level] (15%):

We want to use the 3DOF CrustCrawler setup with a 45° angle pen offset at the tip to a draw a circle somewhere on a slightly curved surface (see **Figure 1**). The whole circle will be within the curved surface.

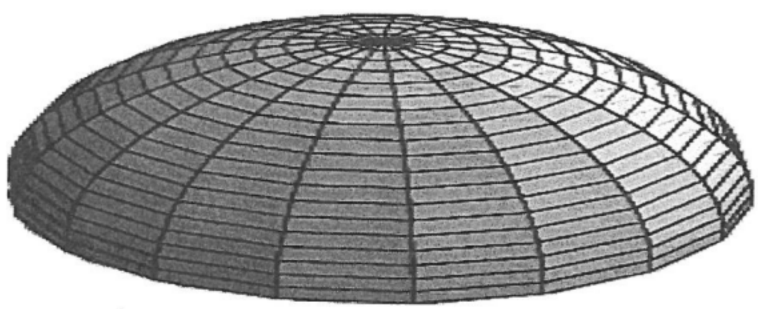


Figure 1. Curved surface

Assume that the whole hemisphere is stationary and within the workspace of the robot. You may also consider the friction between the pen and the surface to be zero, although the friction is high enough for the pen to leave a trace wherever it touches the hemisphere.

a) (5%) Briefly explain why we could use a force sensor to draw the circle on the hemisphere. Would you then need to model the hemisphere? What kind of controller would you use, and why?

Answer:
A force sensor can be used at the tip of the manipulator to project a two-dimensional trajectory onto a three-dimensional surface by measuring the contact force with the surface and keeping it constant while moving along the desired trajectory. The benefit of using the force sensor is that no model of three-dimensional surface is required, and thus the desired trajectory can be projected onto almost any unknown three-dimensional surface. To achieve

this kind of control an impedance controller could be implemented, because the impedance controller will adapt the trajectory with regards to the environment.

b) (5%) Assume we now have a 3DOF wrist with a gripper attached to the tip of the robot arm. The pen used to draw circles is now held in a penholder in front of the robot. The robot now needs to pick up the pen before it can begin to draw. We will use a camera mounted at the tip of the robot to help control the robot to perform this task.

What method of visual servo control would you use to help pick up the pen? Describe the concept of the method in question.

Answer:
An eye-in-hand image-based visual servo controller can be used. This controller uses the image data directly to control the robot motion.

Image features on the object of interest (easily detectable points and lines) are used, and an error function is defined as the vector difference between the desired and the measured locations of these features in the image. The image error is mapped to robot motion, and this will, in effect, move the image until the measured locations of the image features matches the desired locations.

Alternative answer:
An eye-in-hand position-based visual servo controller can be used. This controller uses a partial 3d representation of the world to determine the set points of the robot controller.

The position and orientation of the grasp point on the 3d-representation of the object-of-interest is determined relative to the camera coordinate frame and then provided as set points to the robot controller. **Optional:** *This method is usually less robust due to difficulties with building a 3d representation in real time, and due to the problem with having the object of interest stay in the camera filed of view while there is no direct control over the image itself.*

c) (5%) [Essentially the same question as Task 1b for INF3480 (Bachelor Level)]

Task 2 [INF3480/INF4380] (40%)

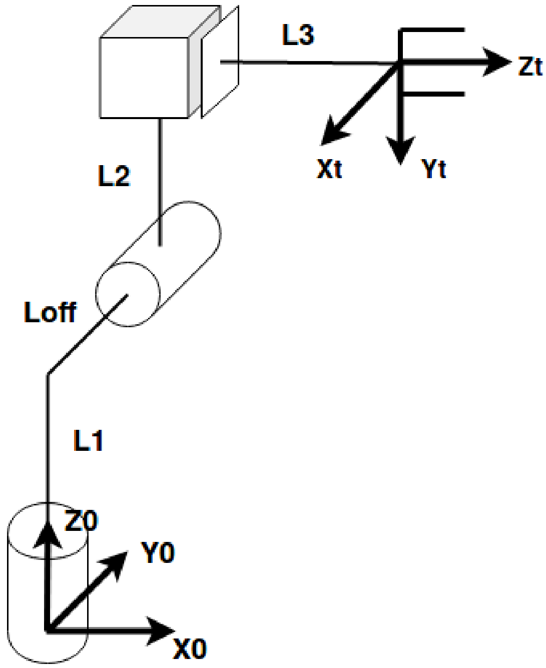
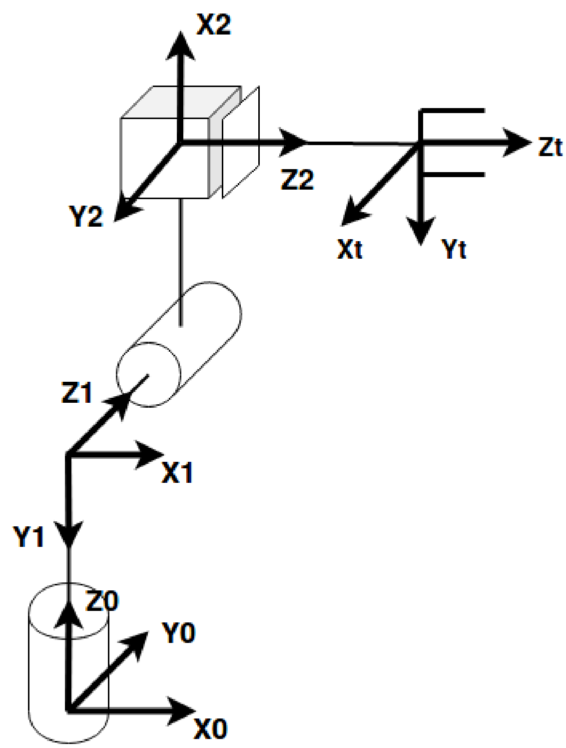


Figure 2: Robot

The given 3DOF robot in **Figure 2** has a revolute joint 1 with a vertical axis of rotation (about **Z0**); a revolute joint 2 with an axis of rotation that is initially parallel to **Y0** and an offset $\mathbf{L_{off}}$ along the same axis; and a prismatic joint 3 that extends along the axis that is perpendicular to the rotation axes of joint 1 and joint 2.

a) (5%) Assign the coordinate frames to the robot in Figure 2 using the Denavit-Hartenberg convention. Write the Denavit-Hartenberg parameters in a table.

Solution:



Parameter Table:

i	θ	**d**	**a**	α
1	$\theta_1^\star$	L_1	0	$-90°$
2	$\theta_2^\star - 90°$	L_{off}	L_2	$-90°$
3	$90°$	$L_3^\star$	0	0

b) (5%) Derive the forward kinematics for the robot from the base coordinate system to the tool coordinate system at the tip of the robot.

Solution:

Canonical transformation:

$$\begin{bmatrix} c_{\theta_i} & -s_{\theta_i}c_{\alpha_i} & s_{\theta_i}s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i}c_{\alpha_i} & -c_{\theta_i}s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Transformation matrices:

$$A_1 = \begin{bmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} s_2 & 0 & c_2 & L_2 s_2 \\ -c_2 & 0 & s_2 & -L_2 c_2 \\ 0 & -1 & 0 & L_{off} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & L_3^* \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_2^0 = A_1 A_2 = \begin{bmatrix} c_1 s_2 & s_1 & c_1 c_2 & L_2 c_1 s_2 - L_{off} s_1 \\ s_1 s_2 & -c_1 & s_1 c_2 & L_2 s_1 s_2 + L_{off} c_1 \\ c_2 & 0 & -s_2 & L_1 + L_2 c_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Full transformation from base to tip:

$$H_t^B = A_1 A_2 A_3 = \begin{bmatrix} s_1 & -c_1 s_2 & c_1 c_2 & L_2 c_1 s_2 + L_3^* c_1 c_2 - L_{off} s_1 \\ -c_1 & -s_1 s_2 & s_1 c_2 & L_2 s_1 s_2 + L_3^* s_1 c_2 + L_{off} c_1 \\ 0 & -c_2 & -s_2 & L_1 + L_2 c_2 - L_3^* s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

c) (10%) Derive the Jacobian Matrix.

Solution:

$$J = \begin{bmatrix} Jv \\ J_\omega \end{bmatrix}$$

Linear part:

$$J_v = \begin{bmatrix} z_0 \times o_3^0 & z_1 \times (o_3^0 - o_1^0) & z_2 \end{bmatrix}$$

First column (refer to forward kinematics - take o_3^0 from H_t^B):

$$J_{v_1} = z_0 \times o_3^0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \begin{bmatrix} L_2 c_1 s_2 + L_3^* c_1 c_2 - L_{off} s_1 \\ L_2 s_1 s_2 + L_3^* s_1 c_2 + L_{off} c_1 \\ L_1 + L_2 c_2 - L_3^* s_2 \end{bmatrix} = \begin{bmatrix} -L_2 s_1 s_2 - L_3^* s_1 c_2 - L_{off} c_1 \\ L_2 c_1 s_2 + L_3^* c_1 c_2 - L_{off} s_1 \\ 0 \end{bmatrix}$$

Second column (refer to forward kinematics - take z_1 from A_1, take o_3^0 from H_t^B and subtract L_1 on the z-element):

$$J_{v_2} = z_1 \times (o_3^0 - o_1^0) = \begin{bmatrix} -s_1 \\ c_1 \\ 0 \end{bmatrix} \times \begin{bmatrix} L_2 c_1 s_2 + L_3^* c_1 c_2 - L_{off} s_1 \\ L_2 s_1 s_2 + L_3^* s_1 c_2 + L_{off} c_1 \\ L_2 c_2 - L_3^* s_2 \end{bmatrix} = \begin{bmatrix} L_2 c_1 c_2 - L_3^* c_1 s_2 \\ L_2 s_1 c_2 - L_3^* s_1 s_2 \\ -L_2 s_2 - L_3^* c_2 \end{bmatrix} \leftarrow \text{simplified}$$

Third column (refer to forward kinematics - take z_2 from H_2^0):

$$J_{v_3} = z_2 = \begin{bmatrix} c_1 c_2 \\ s_1 c_2 \\ -s_2 \end{bmatrix}$$

Angular part:

$$J_\omega = \begin{bmatrix} z_0 & z_1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -s_1 & 0 \\ 0 & c_1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

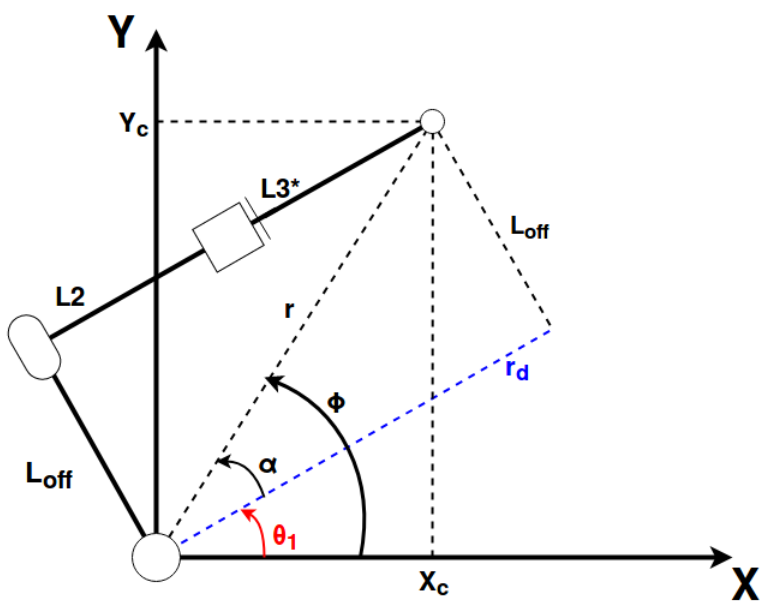
Assembling the full Jacobian matrix:

$$J = \begin{bmatrix} -L_2 s_1 s_2 - L_3^* s_1 c_2 - L_{off} c_1 & L_2 c_1 c_2 - L_3^* c_1 s_2 & c_1 c_2 \\ L_2 c_1 s_2 + L_3^* c_1 c_2 - L_{off} s_1 & L_2 s_1 c_2 - L_3^* s_1 s_2 & s_1 c_2 \\ 0 & -L_2 s_2 - L_3^* c_2 & -s_2 \\ 0 & -s_1 & 0 \\ 0 & c_1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

d) (10%) Derive the inverse kinematics for the robot.

Geometric solution:

Top view illustration:



Helper variables (top view):

$$r = \pm\sqrt{x_c^2 + y_c^2}$$

$$r_d = \pm\sqrt{r^2 - L_{off}^2}$$

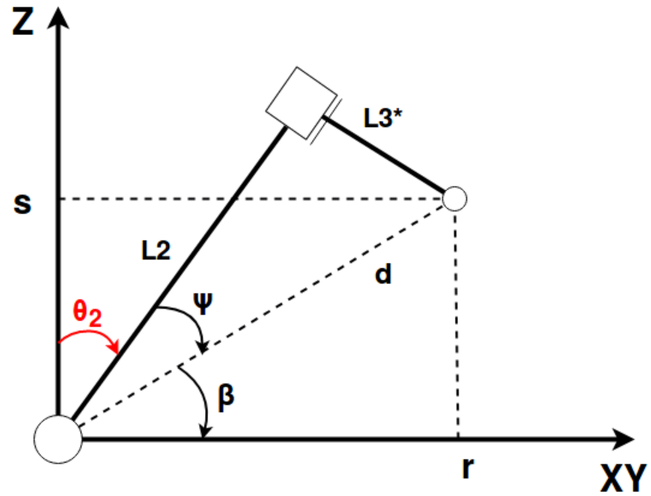
Geometric analysis (top view):

$$\phi = atan2(x_c, y_c)$$

$$\alpha = atan2(r_d, L_{off})$$

$$\theta_1 = \phi - \alpha$$

Side view illustration:



Helper variables (side view):

$$s = z - L_1$$

$$d^2 = r^2 + s^2$$

Geometric analysis (side view):

$$L_3^* = \sqrt{d^2 - L_2^2}$$

$$\psi = atan2(L_2, L_3^*)$$

$$\beta = atan2(r, s)$$

$$\theta_2 = 90^\circ - \psi - \beta$$

e) (10%) Compute the robot's singularities. Draw all the different singularities. Explain all of the different singularities [task invalidated due to extreme complexity].

Valid approach:

Solve $det(J_v) = 0$.

Task 3 [INF3480/INF4380] (20%):

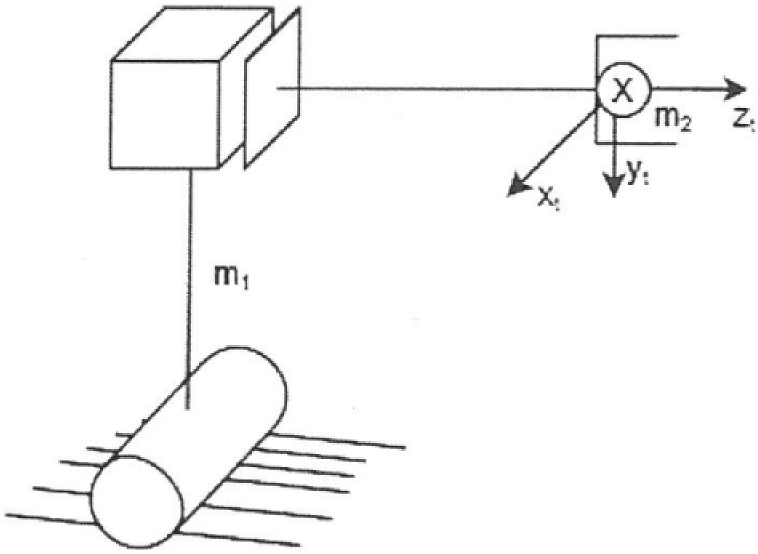


Figure 3: Simplified robot

Figure 3 shows a planar robot with two degrees of freedom. This is a simplification of the robot in Task 2, where joint 1, link 1 and the offset L_{off} are not in use. Assume that link 1 has a length L_2 and a uniformly distributed mass m_1 with the following inertia tensor:

$$I_1 = \begin{bmatrix} I_{1xx} & 0 & 0 \\ 0 & I_{1yy} & 0 \\ 0 & 0 & I_{1zz} \end{bmatrix}$$

Assume further that link 2 has a length L_3 and has a point mass m_2 located at the tip of the robot.

a) (10%) Find the Lagrangian $\mathcal{L}$ of the robotic system in Figure 3.

Solution:

Adjusting the Jacobian matrix:

• The first joint has been removed - turn to zero all the elements in the first column. Then, set to zero the following values: L_1, L_{off} and θ_1. Since θ_1 is now 0, the following holds:

$s_1 = 0$, $c_1 = 0$. Applying changes:

$$J* = \begin{bmatrix} 0 & L_2c_2 - L_3^*s_2 & c_2 \\ 0 & 0 & 0 \\ 0 & -L_2s_2 - L_3^*c_2 & -s_2 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

• Result must be adjusted further - base frame has changed. The mapping between the axes of the old base frame and the new base frame: $x \to x$, $y \to z$, $z \to -y$

Thus, the first row in J_v and J_ω will remain untouched, the second row will take the place of the third row, while the third row will will be mutliplied by -1 and take the place of the second row. The other way to perform all these row swtiching operations may be to separately multiply J_v and J_ω by $R_{z,-90^\circ}$ and then reassembling the full Jacobian. Applying changes:

$$\mathcal{J} = \begin{bmatrix} 0 & L_2c_2 - L_3^*s_2 & c_2 \\ 0 & L_2s_2 + L_3^*c_2 & s_2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}$$

This new Jacobian matrix will be used to find the velocity of m_2.

The system's potential energy:

$$P = P_1 + P_2$$

$$P_1 = m_1gh_1 = m_1g\frac{L_2}{2}c_2$$

$$P_2 = m_2gh_2 = m_2g(L_2c_2 - L_3^*s_2)$$

Kinetic energy of the first mass:

$$K_1 = \frac{m_1v_1^2}{2} + \frac{\omega_1^T RI_1R^t\omega_1}{2}$$

Translational component:

$$v_1 = \mathcal{J}_{v1}\dot{q}$$

To obtain $\mathcal{J}_{v_1}$, we need to set the third column of $\mathcal{J}_{\sqsubseteq}$ to zero (removing the influence of the prismatic joint) and halve L_2, since the center of m_1 is at half-length:

$$\mathcal{J}_{v_1} = \begin{bmatrix} 0 & \frac{L_2}{2}c_2 - L_3^*s_2 & 0 \\ 0 & \frac{L_2}{2}s_2 + L_3^*c_2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Then,

$$v_1 = \begin{bmatrix} \frac{L_2}{2}c_2\dot{\theta}_2 - L_3^* s_2\dot{\theta}_2 \\ \frac{L_2}{2}s_2\dot{\theta}_2 + L_3^* c_2\dot{\theta}_2 \\ 0 \end{bmatrix}$$

and

$$v_1^2 = \frac{L_2^2}{4}\dot{\theta}_2^{\,2} + L_3^{*^2}\dot{\theta}_2^{\,2}$$

Rotational component:
The rotation of the first mass seems to be in the base frame (inertial frame) - applying a rotation matrix **R** should not be necessary in this case:

$$\frac{\omega_1^T I_1 \omega_1}{2} = \frac{\begin{bmatrix} 0 & 0 & \dot{\theta}_2 \end{bmatrix} \begin{bmatrix} I_{1xx} & 0 & 0 \\ 0 & I_{1yy} & 0 \\ 0 & 0 & I_{1zz} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_2 \end{bmatrix}}{2} = \frac{I_{1zz}\dot{\theta}_2^{\,2}}{2}$$

The total kinetic energy of the first mass:

$$K_1 = \frac{m(\frac{L_2^2}{4}\dot{\theta}_2^{\,2} + L_3^{*^2}\dot{\theta}_2^{\,2})}{2} + \frac{I_{1zz}\dot{\theta}_2^{\,2}}{2}$$

The second mass is a point mass - it only has translational kinetic energy:

$$K_2 = \frac{m_2 v_2^2}{2}$$

where one of the possible ways to find v_2 and its square is as follows:

$$v_2 = \mathcal{J}_v \dot{q} = \begin{bmatrix} 0 & L_2c_2 - L_3^* s_2 & c_2 \\ 0 & L_2 s_2 + L_3^* c_2 & s_2 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ \dot{\theta}_2 \\ \dot{L}_3^* \end{bmatrix} = \begin{bmatrix} (L_2c_2 - L_3^* s_2)\dot{\theta}_2 + \dot{L}_3^* c_2 \\ (L_2 s_2 + L_3^* c_2)\dot{\theta}_2 + \dot{L}_3^* s_2 \\ 0 \end{bmatrix}$$

$$v_2^2 = ((L_2c_2 - L_3^* s_2)\dot{\theta}_2 + \dot{L}_3^* c_2)^2 + ((L_2 s_2 + L_3^* c_2)\dot{\theta}_2 + \dot{L}_3^* s_2)^2 + 0$$

Some terms can be paired to produce a shorter expression, while others negate each other:

$$v_2^2 = [(L_2^2 c_2^2 - 2L_2 L_3^* c_2 s_2 + L_3^{*^2} s_2^2)\dot{\theta}_2^{\,2} + 2(L_2c_2 - L_3^* s_2)\dot{\theta}_2 \dot{L}_3^* c_2 + \dot{L}_3^{*^2} c_2^2] +$$

$$[(L_2 s_2^2 + 2L_2 L_3^* s_2 c_2 + L_3^{*^2} c_2^2)\dot{\theta}^2 + 2(L_2 s_2 + L_3^* c_2)\dot{\theta}_2 \dot{L}_3^* s_2 + \dot{L}_3^{*^2} s_2^2] =$$

$$[(L_2^2 - 0 + L_3{*}^2)\dot{\theta}_2^{\,2} + 2(L_2 c_2^2 \dot{\theta}_2 \dot{L}_3^* - L_3^* s_2 c_2 \dot{\theta}_2 \dot{L}_3^*) + \dot{L}_3^{*^2}] + [2(L_2 s_2^2 \dot{\theta}_2 \dot{L}_3^* - L_3^* c_2 s_2 \dot{\theta}_2 \dot{L}_3^*)] =$$

$$[(L_2^2 + L_3^{*2})\dot{\theta}_2^{\,2} + 2L_2 \dot{L}_3^* \dot{\theta}_2 - 0 + \dot{L}_3^{*^2}]$$

Therefore,

$$K_2 = m_2 \frac{(L_2^2 + L_3^{*^2})\dot{\theta}_2^{\,2} + 2L_2 \dot{L}_3^* \dot{\theta}_2 + \dot{L}_3^{*^2}}{2}$$

Finally, it is possible to form the Lagrangian:

$$\mathcal{L} = K - P$$

$$\mathcal{L} = \frac{m_1(\frac{L_2^2}{4}\dot{\theta_2}^2 + L_3^{*^2}\dot{\theta_2}^2)}{2} + \frac{I_{1zz}\dot{\theta_2}^2}{2} + \frac{m_2(L_2^2 + L_3^{*^2})\dot{\theta_2}^2 + 2L_2\dot{L_3^*}\dot{\theta_2} + \dot{L_3^*}^2}{2} - m_1 g\frac{L_2}{2}c_2 - m_2 g(L_2c_2 - L_3^* s_2)$$

or, if unfactored,

$$\mathcal{L} = \frac{m_1L_2^2\dot{\theta_2}^2}{8} + \frac{m_1L_3^{*^2}\dot{\theta_2}^2}{2} + \frac{I_{1zz}\dot{\theta_2}^2}{2} + \frac{m_2L_2^2\dot{\theta_2}^2}{2} + \frac{m_2L_3^{*^2}\dot{\theta_2}^2}{2} + m_2L_2\dot{L_3^*}\dot{\theta_2} + \frac{m_2\dot{L_3^{*2}}}{2}$$
$$-\frac{m_1gL_2c_2}{2} - m_2gL_2c_2 + m_2gL_3^*s_2$$

b) (10%) Derive the dynamic equations for the robot using the Euler-Lagrange formulation. Formulate the Euler-Lagrange equations of the form $D(q)\ddot{q}+C(q,\dot{q})\dot{q}+G(q)=\tau$

Solution:

$$\tau = \begin{bmatrix} \tau_2 \\ f_3 \end{bmatrix}$$

The torque on the revolute joint:

$$\tau_2 = \frac{d}{dt}\frac{\partial \mathcal{L}}{\dot{\theta_2}} - \frac{\partial \mathcal{L}}{\theta_2}$$

$$\tau_2 = \frac{m_1L_2^2\ddot{\theta_2}}{4} + 2m_1L_3^*\dot{L_3^*}\dot{\theta_2} + m_1L_3^{*^2}\ddot{\theta_2} + I_{1zz}\ddot{\theta_2} + m_2L_2^2\ddot{\theta_2} + 2m_2L_3^*\dot{L_3^*}\dot{\theta_2} + m_2L_3^{*^2}\ddot{\theta_2} + m_2L_2\ddot{L_3^*}$$
$$-\frac{m_1gL_2s_2}{2} - m_2gL_2s_2 - m_2gL_3^*c_2$$

The force on the prismatic joint:

$$f_3 = \frac{d}{dt}\frac{\partial \mathcal{L}}{\dot{L_3^*}} - \frac{\partial \mathcal{L}}{L_3^*}$$

$$f_3 = m_2L_2\ddot{\theta_2} + m_2\ddot{L_3^*} - m_1L_3^*\dot{\theta_2}^2 - m_2L_3^*\dot{\theta_2}^2 - m_2gs_2$$

Rewriting τ:

$$\tau = D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q)$$

$$\tau = \begin{bmatrix} \frac{m_1L_2^2}{4} + m_1L_3^{*^2} + I_{1zz} + m_2L_2^2 + m_2L_3^{*^2} & m_2L_2 \\ m_2L_2 & m_2 \end{bmatrix} \begin{bmatrix} \ddot{\theta_2} \\ \ddot{L_3^*} \end{bmatrix} +$$
$$\begin{bmatrix} 2(m_1+m_2)L_3^*\dot{L_3} & 0 \\ -m_1L_3^*\dot{\theta_2} - m_2L_3^*\dot{\theta_2} & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta_2} \\ \dot{L_3^*} \end{bmatrix} +$$
$$\begin{bmatrix} -\frac{m_1gL_2s_2}{2} - m_2gL_2s_2 - m_2gL_3^*c_2 \\ -m_2gs_2 \end{bmatrix}$$

Note that the $C(q,\dot{q})$ matrix is not unique in this case - there many other ways to split the equation that are just as valid.

Task 4 [INF3480/INF4380] (20%):

We have the system $J\ddot{\theta}(t) + B\dot{\theta}(t) + K\theta(t) = \tau$. When we use the Laplace transform on the system, we get $Js^2\theta(s) + Bs\theta(s) + K\theta(s) = \tau$.

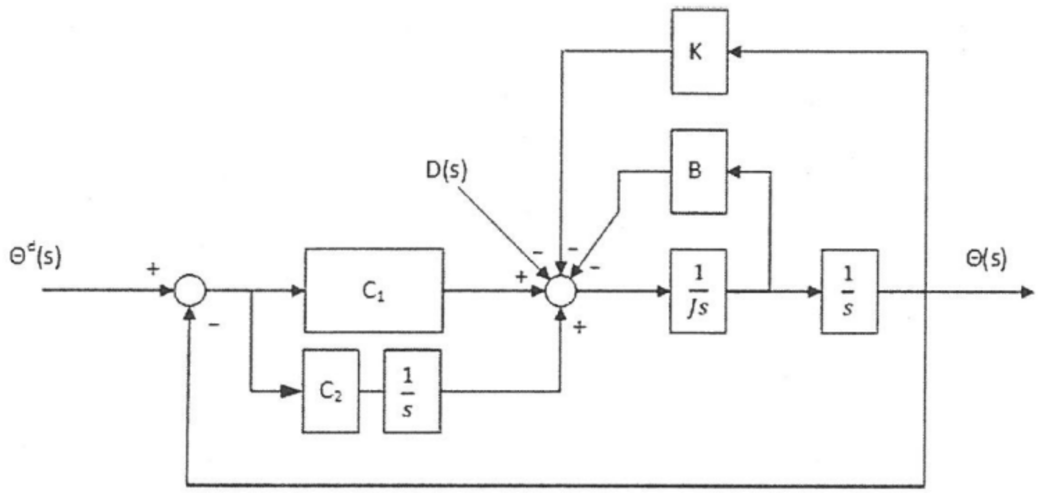


Figure 4: Control system

a) (2.5%) Figure 4 shows up a set-point tracking control system in the Laplace domain. What is the name of the controller used here? What properties does it provide to the system?

Answer:
Figure 4 demonstrates a **PI** (Proportional-Integral) controller.

Proportional component property - regulates towards the setpoint.

Integral component property - removes the steady-state error.

b) (2.5%) The system with the controller in Figure 4 have oscillations. How can we remove the oscillations while keeping the other system properties? What is the name of the new controller?

Answer:
The goal can be achieved by adding the **D**erivative component to the controller. The new controller is therefore a **PID** (Proportional-Integral-Derivative) controller.

c) (5%) Find the closed-loop transfer function between the input value ($\theta_d(s)$ - desired angle) and output value ($\theta(s)$ - actual/measured angle) for the system with this new improved controller.

Solution:

The system equation:

$$U(s) = Js^2\theta + Bs\theta + K\theta + D$$

The controller equation:

$$U(s) = C_1(\theta_d - \theta) + \frac{C_2}{s}(\theta_d - \theta) - C_3 s\theta$$

The complete equation (the system and the controller):

$$Js^2\theta + Bs\theta + K\theta + D = C_1(\theta_d - \theta) + \frac{C_2}{s}(\theta_d - \theta) - C_3 s\theta$$

Factorizing terms and solving for output θ:

$$\theta = \frac{\theta_d(C_1 + \frac{C_2}{s}) - D}{Js^2 + Bs + K + C_1 + \frac{C_2}{s} + C_3 s}$$

The transfer function we are looking for is the Signal Transfer Function (as opposed to the Disturbance Transfer Function). This allows us to disgregard distrubance D:

$$\theta = \frac{\theta_d(C_1 + \frac{C_2}{s})}{Js^2 + Bs + K + C_1 + \frac{C_2}{s} + C_3 s}$$

The transfer function is then:

$$H(s) = \frac{\theta}{\theta_d} = \frac{\theta_d C_1 + \frac{C_2}{s}}{Js^2 + Bs + K + C_1 + \frac{C_2}{s} + C_3 s}$$

d) (5%) Use the final value theorem to calculate the steady state error for the closed-loop control system with the new improved controller. Consider both the desired angle $\theta_d(s)$ and the disturbance D(s) to be "step inputs". Comment on the result.

Solution:

Step input:

$$\theta_d = \frac{\Omega_d}{s}$$

Constant disturbance:

$$D = \frac{d}{s}$$

The final value theorem:

$$\lim_{t\to\infty} \theta(t) = \lim_{s\to 0} s\theta(s)$$

Using the rearranged equation from **(c)**:

$$\lim_{s\to 0} s\theta(s) = \lim_{s\to 0} \frac{s(\frac{\Omega_d}{s}(C_1 + \frac{C_2}{s}) - \frac{d}{s})}{Js^2 + Bs + K + C_1 + \frac{C_2}{s} + C_3 s} =$$

$$\lim_{s\to 0} \frac{\Omega_d(C_1+\frac{C_2}{s}) - d| * s}{Js^2+Bs+K+C_1+\frac{C_2}{s}+C_3 s| * s} =$$

$$\lim_{s\to 0} \frac{\cancel{C_1\Omega_d s} + C_2\Omega_d - \cancel{ds}}{\cancel{Js^3}+\cancel{Bs^2}+\cancel{Ks}+\cancel{C_1 s}+C_2+\cancel{C_3 s^2}} =$$

$$\frac{C_2\Omega_d}{C_2} = \Omega_d$$

<u>Steady-state error:</u>

$$\lim_{t\to\infty}(\theta_d(t)-\theta(t)) = \lim_{s\to 0}(s\theta_d(s) - s\theta(s)) = \frac{\cancel{s}\Omega_d}{\cancel{s}} - \Omega_d = 0$$

The steady-state error evaluates to zero due to the presence of the Integral term/component in the controller.

e) (5%) Transform the following equation the Laplace domain:

$$m\ddot{x} + kx = u$$

Find the transfer function of this system, assuming that u is the input and x is the ouput. Draw a block diagram of the system.

Solution:

Laplace transform:

$$ms^2X(s) + kX(s) = U(s)$$

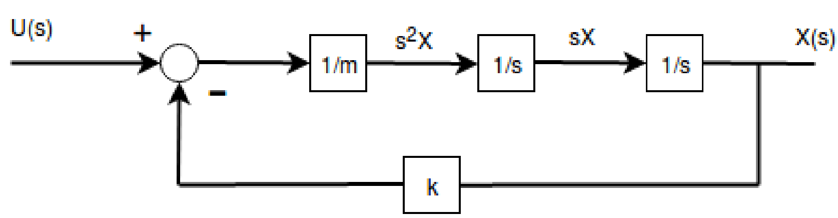
Rearranging:

$$X(s)(ms^2+k) = U(s)$$

The transfer function:

$$\frac{X(s)}{U(s)} = \frac{1}{ms^2+k}$$

The block diagram:



f) (2.5%) How would you examine the stability of the control system, and what is required to get a stable system? Find the poles and zeros of the system in Task 4e when the spring constant is $k = 8$ and the mass of the system is $m = 2$. Show the poles and zeros in a plot. What do we call a system like this?

Solution:

To examine the stability of the system, we need to find the poles of the transfer function. The system is asymptotically stable, if all the poles are located in the left half-plane (real component ¡ 0). The system is marginally stable if at least one of the poles has the real component equal to zero (positioned on the imaginary axis).
The transfer function:

$$H(s) = \frac{1}{ms^2 + k}$$

The function does not have zeros, since $1 \neq 0$. To examine the poles of the function, set the denominator to zero:

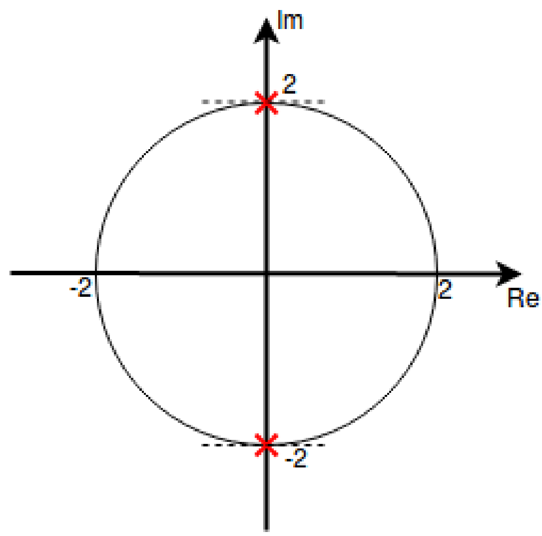
$$ms^2 + k = 0$$

$$s = \pm\sqrt{-\frac{k}{m}}$$

Substituting $k = 8$ and $m = 2$:

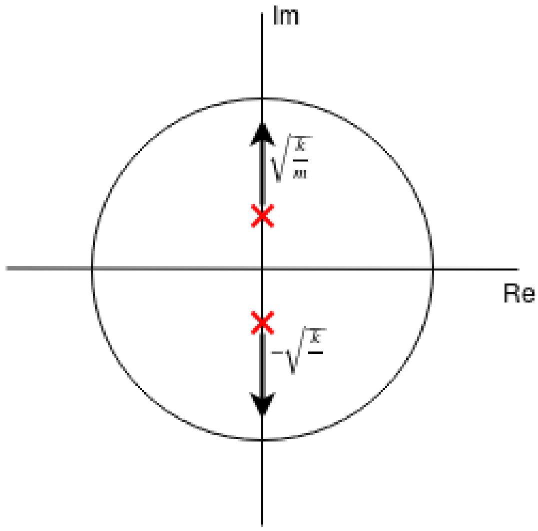
$$s = \pm\sqrt{-\frac{8}{2}} = \pm\sqrt{-4} = 0 \pm 2i$$

Since both poles are positioned exactly on the imaginary axis, the system is marginally stable. The pole-zero plot is as follows:



g) (2.5%) What is a root locus plot? Explain what it tells us and show it graphically.

Solution:
A root locus plot shows how the poles of the system move with the changes in the control parameters. Visual example:


Im
$\sqrt{\frac{k}{m}}$
Re