Robotics 2

Remote Exam – September 11, 2020

Exercise #1

Consider the planar 3R robot with links of equal length L in Fig. 1, driven by joint velocity commands $\dot{\boldsymbol{q}} \in \mathbb{R}^3$. The end effector should trace the nominal linear path from A to B with a constant speed $v > 0$, while the entire robot avoids any collision with a single static obstacle O. The obstacle is of known circular shape but uncertain radius $R \leq L/4$, and is placed at least at a distance $\rho_{min} = L$ from the robot base and not farther than $\rho_{max} = 3L$. Its actual location is unknown a priori, but the obstacle can be detected by an omnidirectional proximity sensor mounted on the robot end effector and having a sensing range $\sigma = 1.5L$.

Define a sensor-based control scheme that makes the robot perform *at best* the assigned task, and describe qualitatively its expected performance.

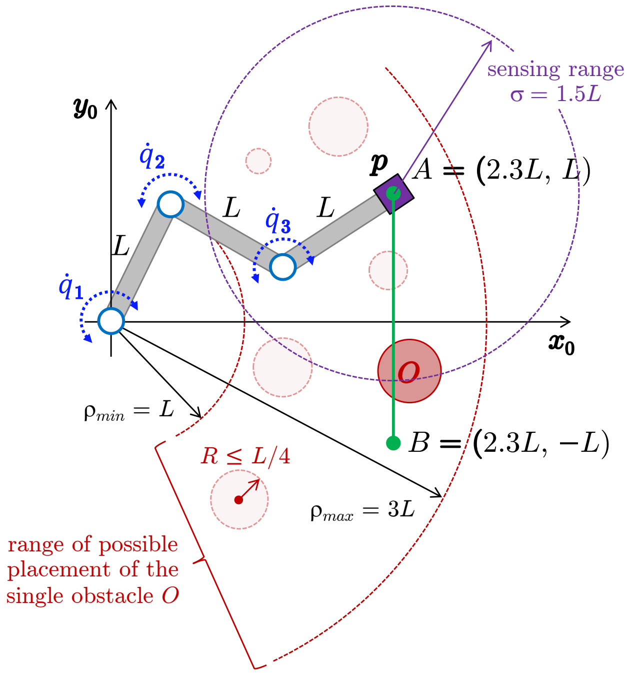

Figure 1: The 3R robot with the nominal Cartesian path AB to be traced and the static circular obstacle O to be avoided (several possible alternative locations of the single obstacle are shown).

Exercise #2

In an image-based visual servoing control scheme, the camera mounted on the robot end effector (eye-in-hand) looks for three specific points in the scene, characterized by their point features in the image plane with coordinates (u_i, v_i), for $i = 1, 2, 3$. Derive the expression of the 2×6 *interaction matrix* $\boldsymbol{J}_b$ associated to the geometric barycenter $\boldsymbol{b} \in \mathbb{R}^2$ of the triangle in the image having these point features as the three vertices, namely

$$\dot{\boldsymbol{b}} = \boldsymbol{J}_b \begin{pmatrix} \boldsymbol{V} \\ \boldsymbol{\Omega} \end{pmatrix},$$

where $\boldsymbol{V} \in \mathbb{R}^3$ and $\boldsymbol{\Omega} \in \mathbb{R}^3$ are respectively the linear and angular velocity of the camera.

Provide then an instantaneous motion of the camera such that $\dot{\boldsymbol{b}} = \boldsymbol{0}$.

Exercise #3

With reference to the planar RP robot in Fig. 2, using the symbolic parameters specified therein, determine the complete expression of the *Cartesian inertia matrix* $\boldsymbol{M}_{\boldsymbol{p}}(\boldsymbol{q})$ of the robot at the tip $\boldsymbol{p} \in \mathbb{R}^2$ as a function of the configuration $\boldsymbol{q}$.

Provide then the numerical value of $\boldsymbol{M}_{\boldsymbol{p}}(\boldsymbol{q}^*)$ at $\boldsymbol{q}^* = \begin{pmatrix} 0 & 3 \end{pmatrix}^T$ [rad, m], using the parameters:

$$l_1 = 1, \quad d_1 = 0.5, \quad m_1 = 3, \quad I_1 = 0.25; \qquad d_2 = 0.5, \quad m_2 = 0.5, \quad I_2 = 0.875.$$

Assuming that this robot is at rest in $\boldsymbol{q}^*$ on a horizontal plane, check that the tip acceleration $\ddot{\boldsymbol{p}} \in \mathbb{R}^2$ has the same direction of any force $\boldsymbol{F} \in \mathbb{R}^2$ in the plane applied to the tip of the robot.

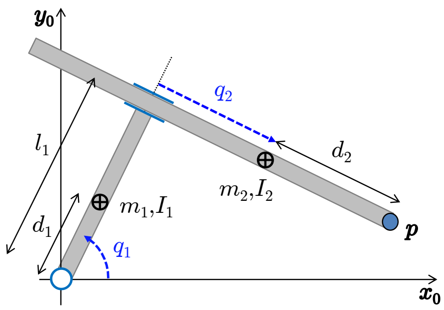

Figure 2: A 2-dof RP robot with its relevant kinematic and dynamic parameters.

Exercise #4

A 3R robot moves on a horizontal plane in the presence of geometric constraints in the Cartesian space, as illustrated in Fig. 3. Assume that the distance k satisfies the inequalities $l_1 < k < l_1 + l_2$.

- Determine the dimension M of the constraints and their possible expression $\boldsymbol{h}(\boldsymbol{q}) = \boldsymbol{0}$, with the associated Jacobian matrix $\boldsymbol{A}(\boldsymbol{q}) = \partial \boldsymbol{h}(\boldsymbol{q})/\partial \boldsymbol{q}$.
- Define the $(3-M) \times 3$ matrix $\boldsymbol{D}(\boldsymbol{q})$ to complete $\boldsymbol{A}(\boldsymbol{q})$ in a nonsingular way, as well as the blocks $\boldsymbol{E}(\boldsymbol{q})$ and $\boldsymbol{F}(\boldsymbol{q})$ of the inverse that appear in the *reduced dynamics* of this constrained robot. The reduced model should hold for any $\boldsymbol{q}$ such that the contact situation remains as in Fig. 3.
- Let m_{ij} be the elements of the (symmetric) inertia matrix $\boldsymbol{M}(\boldsymbol{q})$ of this robot in free space. Give the full expression of the $(3-M)\times(3-M)$ *reduced inertia matrix* in the constrained case.

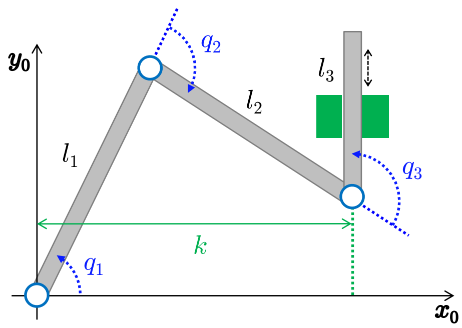

Figure 3: A 3R robot with geometric constraints in the Cartesian space limiting its motion.

Exercise #5

Consider a 2R robot arm of human-like size and weight moving in a vertical plane with negligible friction. Two model-based controllers are being compared in a trajectory tracking problem in the joint space. The first is based on feedback linearization, yielding a control torque $\boldsymbol{u}_{FBL}(t)$, the second is a Lyapunov-based control law with global asymptotic convergence property, yielding a torque $\boldsymbol{u}_{GLB}(t)$.

Assuming that the PD gain matrices are the same in both control laws, provide the explicit expression of the torque difference $\boldsymbol{\Delta u}(t) = \boldsymbol{u}_{FBL}(t) - \boldsymbol{u}_{GLB}(t)$.

Next, when the desired joint trajectory and the PD gains are respectively

$$\boldsymbol{q}_d(t) = \begin{pmatrix} \dfrac{\pi}{2} + 3\sin\dfrac{\pi t}{2} \\ 1 - \cos 2\pi t \end{pmatrix}, \qquad \boldsymbol{K}_P = 100 \cdot I_{2\times 2}, \quad \boldsymbol{K}_D = 20 \cdot I_{2\times 2},$$

assume that at time $t = 2$ s, the current robot configuration is $\boldsymbol{q}(2) = \begin{pmatrix} \pi/2 & -\pi/2 \end{pmatrix}^T$ [rad] and the velocity tracking error is zero. For each robot joint, determine which is the controller that uses the larger instantaneous torque in absolute value.

[240 minutes (4 hours); open books]

Solution

September 11, 2020

Exercise #1

This is an open-ended exercise and many possible schemes could be devised. Here, we present one where the 3R robot uses its redundancy with respect to the two-dimensional motion task assigned to its end effector as a mean to avoid the single obstacle detected online. The end-effector motion task has a higher priority than the collision avoidance task, as long as the minimum distance between the robot and the obstacle O stays above some threshold $\varepsilon > 0$. Otherwise, the controller switches the priority order and moves the robot primarily to avoid collision with the obstacle, trying to keep also the tracking error as small as possible. Indeed, when the obstacle is placed on the end-effector path, this switching will certainly happen. To implement such a control strategy, we use the following items.

- A clearance function defined by

$$H(\boldsymbol{q}) = \min_{\substack{\boldsymbol{a}(\boldsymbol{q}) \in \text{robot} \\ \boldsymbol{b} \in \text{obstacle}}} \|\boldsymbol{a}(\boldsymbol{q}) - \boldsymbol{b}\|. \tag{1}$$

  In this expression, the detection of points $\boldsymbol{b}$ belonging to the circular obstacle O is made by the proximity sensor mounted on the end-effector. The sensor is able to reconstruct the entire visible surface of the obstacle and thus determine the closest point to the robot. The range of the proximity sensor covers the entire area of interest. The location of every point $\boldsymbol{a}(\boldsymbol{q})$ on the robot body is known from the encoder measures of $\boldsymbol{q}$ and via the direct kinematics of the robot.

  The clearance function in (1) can also be approximated by choosing a number of control points on the robot body for robot-obstacle distance computation, rather than the entire robot skeleton. For instance, one can use the three points $\boldsymbol{P}_{j2}$ = location of joint 2, $\boldsymbol{P}_{j3}$ = location of joint 3, and $\boldsymbol{P}_{ee}$ = end-effector location. Then, (1) would be replaced by the clearance function

$$H(\boldsymbol{q}) = \min_{\boldsymbol{b} \in \text{obstacle}} \left\{\|\boldsymbol{P}_{j2}(q_1) - \boldsymbol{b}\|, \|\boldsymbol{P}_{j3}(q_1, q_2) - \boldsymbol{b}\|, \|\boldsymbol{P}_{ee}(q_1, q_2, q_3) - \boldsymbol{b}\|\right\}. \tag{2}$$

  In both cases, we can use the gradient $\nabla_{\boldsymbol{q}} H(\boldsymbol{q}) = \left(\partial H(\boldsymbol{q})/\partial \boldsymbol{q}\right)^T$ in the control scheme, together with a stepsize factor $\alpha > 0$. While both clearance functions (1) and (2) are continuous (in space and time), their gradient may have discontinuities.

- The primary task Jacobian associated to the end-effector position $\boldsymbol{p}$, as given by

$$\boldsymbol{J}(\boldsymbol{q}) = \frac{\partial \boldsymbol{p}(\boldsymbol{q})}{\partial \boldsymbol{q}} = \begin{pmatrix} -\left(l_1 s_1 + l_2 s_{12} + l_3 s_{123}\right) & -\left(l_2 s_{12} + l_3 s_{123}\right) & -l_3 s_{123} \\ l_1 c_1 + l_2 c_{12} + l_3 c_{123} & l_2 c_{12} + l_3 c_{123} & l_3 c_{123} \end{pmatrix},$$

  with the usual shorthand notation for trigonometric quantities (e.g., $s_{123} = \sin(q_1 + q_2 + q_3)$). We will use the pseudoinverse $\boldsymbol{J}^{\#}(\boldsymbol{q})$ of this matrix. Moreover, the desired task velocity will be $\dot{\boldsymbol{p}}_d = v\,(B - A)\,/\|B - A\|$.

- A switching control law defined as

$$\dot{\boldsymbol{q}} = \begin{cases} \boldsymbol{J}^{\#}(\boldsymbol{q})\left(\dot{\boldsymbol{p}}_d + \boldsymbol{K}_P\left(\boldsymbol{p}_d - \boldsymbol{p}(\boldsymbol{q})\right)\right) + \Big(\boldsymbol{I} - \boldsymbol{J}^{\#}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})\Big)\,\alpha \nabla_{\boldsymbol{q}} H(\boldsymbol{q}) & \\ \quad = \alpha \nabla_{\boldsymbol{q}} H(\boldsymbol{q}) + \boldsymbol{J}^{\#}(\boldsymbol{q})\Big(\dot{\boldsymbol{p}}_d + \boldsymbol{K}_P\left(\boldsymbol{p}_d - \boldsymbol{p}(\boldsymbol{q})\right) - \alpha \boldsymbol{J}(\boldsymbol{q}) \nabla_{\boldsymbol{q}} H(\boldsymbol{q})\Big), & \text{if } H(\boldsymbol{q}) > \epsilon, \\ \alpha \nabla_{\boldsymbol{q}} H(\boldsymbol{q}), & \text{if } H(\boldsymbol{q}) \leq \epsilon. \end{cases} \tag{3}$$

When the obstacle is sufficiently far, with the control law (3) the robot executes exactly the desired end-effector trajectory $\boldsymbol{p}_d(t)$, while the gradient of the clearance function is projected in the one-dimension null space of the primary task. Instead, when the robot is getting too close to the obstacle, the control law will move away the nearest point of the robot. In the given form of the gradient of $H(\boldsymbol{q})$ in the configuration space, obstacle repulsion is a three-dimensional task which leaves no space for a secondary consideration of the original tracking task. Therefore, during this phase, the end-effector path is typically abandoned[1]. When the minimum clearance ε is recovered, the control will switch back to the previous law, trying to recover the accumulated error with respect to the desired Cartesian trajectory. In order to do so, a position error feedback term with (diagonal) matrix gain $\boldsymbol{K}_P > 0$ has been added in (3).

We note finally that the proposed control scheme is purely reactive (there is no planning for obstacle avoidance) and only local in scope (we cannot exclude that the robot gets stuck before reaching the point B). Further, there is no motion stop explicitly involved with a control switch. As a consequence, switches are typically associated with discontinuities of the joint velocity commands.

Exercise #2

Associated to a point $\boldsymbol{P}_i = (X_i, Y_i, Z_i)$ in the Cartesian space, with its coordinates expressed in the camera frame, there is a point feature $\boldsymbol{f}_{\boldsymbol{p}_i} = (u_i, v_i)$ in the image plane whose interaction matrix $\boldsymbol{J}_{p_i}$ takes the form (see the lecture slides)

$$\boldsymbol{J}_{p_i}(u_i, v_i, Z_i) = \begin{pmatrix} -\dfrac{\lambda}{Z_i} & 0 & \dfrac{u_i}{Z_i} & \dfrac{u_i v_i}{\lambda} & -\dfrac{u_i^2}{\lambda} - \lambda & v_i \\ 0 & -\dfrac{\lambda}{Z_i} & \dfrac{v_i}{Z_i} & \dfrac{v_i^2}{\lambda} + \lambda & -\dfrac{u_i v_i}{\lambda} & -u_i \end{pmatrix},$$

where $\lambda > 0$ is the focal length of the camera and the depth $Z_i > 0$ is limited (by the visual range of the camera). The geometric barycenter $\boldsymbol{b} \in \mathbb{R}^2$ of a triangle in the image plane is simply obtained from its three vertices, namely the coordinates of the point features $\boldsymbol{f}_{\boldsymbol{p}_i}$, $i = 1, 2, 3$, as

$$\boldsymbol{b} = \begin{pmatrix} b_u \\ b_v \end{pmatrix} = \frac{1}{3} \begin{pmatrix} u_1 + u_2 + u_3 \\ v_1 + v_2 + v_3 \end{pmatrix}.$$

Therefore, the interaction matrix $\boldsymbol{J}_{\boldsymbol{b}}$ takes the form

$$\begin{aligned} \boldsymbol{J}_{\boldsymbol{b}} &= \frac{1}{3}\Big(\boldsymbol{J}_{p_1}(u_1, v_1, Z_1) + \boldsymbol{J}_{p_2}(u_2, v_2, Z_2) + \boldsymbol{J}_{p_3}(u_3, v_3, Z_3)\Big) \\ &= \frac{1}{3} \begin{pmatrix} -\lambda \displaystyle\sum_{i=1}^{3} \frac{1}{Z_i} & 0 & \displaystyle\sum_{i=1}^{3} \frac{u_i}{Z_i} & \dfrac{1}{\lambda} \displaystyle\sum_{i=1}^{3} u_i v_i & -\dfrac{1}{\lambda} \displaystyle\sum_{i=1}^{3} u_i^2 - 3\lambda & \displaystyle\sum_{i=1}^{3} v_i \\ 0 & -\lambda \displaystyle\sum_{i=1}^{3} \frac{1}{Z_i} & \displaystyle\sum_{i=1}^{3} \frac{v_i}{Z_i} & \dfrac{1}{\lambda} \displaystyle\sum_{i=1}^{3} v_i^2 + 3\lambda & -\dfrac{1}{\lambda} \displaystyle\sum_{i=1}^{3} u_i v_i & -\displaystyle\sum_{i=1}^{3} u_i \end{pmatrix} \\ &= \boldsymbol{J}_{\boldsymbol{b}}(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{Z}), \end{aligned}$$

[1] Other collision avoidance schemes can be defined, still using an artificial potential to keep the robot away from the obstacle. Obstacle avoidance can be formulated as a two-dimensional task, when the repulsive action is defined along the gradient of the Cartesian clearance, or even as a one-dimensional task, when only the projection of the repulsive Cartesian alction along the clearance direction is specified. In both cases, it is possible to accommodate the tracking task in the null space of the avoidance task.

with $\boldsymbol{u} = (u_1, u_2, u_3)$, $\boldsymbol{v} = (v_1, v_2, v_3)$, and depths $\boldsymbol{Z} = (Z_1, Z_2, Z_3)$. Note that the obtained interaction matrix $\boldsymbol{J_b}$ is in general *different* from the interaction matrix associated to the single point feature of the Cartesian barycenter $\boldsymbol{P_b} = \frac{1}{3}\left(\boldsymbol{P}_1 + \boldsymbol{P}_2 + \boldsymbol{P}_3\right)$ of the three points in 3D space. Possible camera motions that belong to the null space of $\boldsymbol{J_b}$ are

$$\boldsymbol{V} = \begin{pmatrix} \displaystyle\sum_{i=1}^{3} \frac{u_i}{Z_i} \\ \displaystyle -\sum_{i=1}^{3} \frac{v_i}{Z_i} \\ \displaystyle \lambda \sum_{i=1}^{3} \frac{1}{Z_i} \end{pmatrix}, \quad \boldsymbol{\Omega} = \boldsymbol{0} \qquad \text{(a pure linear motion)},$$

or

$$\boldsymbol{V} = \begin{pmatrix} \displaystyle\sum_{i=1}^{3} v_i \\ \displaystyle\sum_{i=1}^{3} u_i \\ 0 \end{pmatrix}, \quad \boldsymbol{\Omega} = \begin{pmatrix} 0 \\ 0 \\ \displaystyle \lambda \sum_{i=1}^{3} \frac{1}{Z_i} \end{pmatrix} \qquad \begin{array}{c}\text{(linear motion parallel to the image plane,}\\ \text{with angular motion around the optical axis).}\end{array}$$

More independent camera motions with $\dot{\boldsymbol{b}} = \boldsymbol{0}$ exist, since the dimension of $\mathcal{N}\left\{\boldsymbol{J_b}\right\}$ is at least 4.

Exercise #3

We compute first the kinetic energy of the RP robot. For the two links, we have

$$T_1 = \frac{1}{2}\left(I_1 + m_1 d_1^2\right)\dot{q}_1^2, \qquad T_2 = \frac{1}{2}\, I_2\, \dot{q}_1^2 + \frac{1}{2}\, m_2\, \boldsymbol{v}_{c2}^T \boldsymbol{v}_{c2},$$

with

$$\boldsymbol{v}_{c2} = \dot{\boldsymbol{p}}_{c2} = \frac{d}{dt}\begin{pmatrix} l_1 \cos q_1 + q_2 \sin q_1 \\ l_1 \sin q_1 - q_2 \cos q_1 \end{pmatrix} = \begin{pmatrix} \left(q_2 \cos q_1 - l_1 \sin q_1\right)\dot{q}_1 + \sin q_1\, \dot{q}_2 \\ \left(l_1 \cos q_1 + q_2 \sin q_1\right)\dot{q}_1 - \cos q_1\, \dot{q}_2 \end{pmatrix}.$$

Therefore, from $T = T_1 + T_2 = \frac{1}{2}\dot{\boldsymbol{q}}^T \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}}$, we obtain the inertia matrix

$$\boldsymbol{M}(\boldsymbol{q}) = \begin{pmatrix} I_1 + m_1 d_1^2 + I_2 + m_2 l_1^2 + m_2 q_2^2 & -m_2 l_1 \\ -m_2 l_1 & m_2 \end{pmatrix}.$$

The Jacobian associated to the linear velocity $\boldsymbol{v} = \dot{\boldsymbol{p}} \in \mathbb{R}^2$ of the robot tip is computed as

$$\dot{\boldsymbol{p}} = \frac{d}{dt}\begin{pmatrix} l_1 \cos q_1 + (q_2 + d_2)\sin q_1 \\ l_1 \sin q_1 - (q_2 + d_2)\cos q_1 \end{pmatrix} = \frac{\partial \boldsymbol{p}(\boldsymbol{q})}{\partial \boldsymbol{q}}\,\dot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}$$

$$\Rightarrow \quad \boldsymbol{J}(\boldsymbol{q}) = \begin{pmatrix} (q_2 + d_2)\cos q_1 - l_1 \sin q_1 & \sin q_1 \\ l_1 \cos q_1 + (q_2 + d_2)\sin q_1 & -\cos q_1 \end{pmatrix}.$$

A singularity occurs when $\det \boldsymbol{J}(\boldsymbol{q}) = -(q_2+d_2) = 0$. Out of this singularity, and using also a shorthand notation for the trigonometric terms, the Cartesian inertia matrix at the robot tip is

$$
\begin{aligned}
\boldsymbol{M}_{\boldsymbol{p}}(\boldsymbol{q}) &= \boldsymbol{J}^{-T}(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})\boldsymbol{J}^{-1}(\boldsymbol{q}) \\
&= \frac{1}{(q_2+d_2)^2}\begin{pmatrix} -c_1 & -l_1c_1-(q_2+d_2)s_1 \\ -s_1 & (q_2+d_2)c_1 - l_1 s_1 \end{pmatrix}\begin{pmatrix} I_1+m_1d_1^2+I_2+m_2l_1^2+m_2q_2^2 & -m_2l_1 \\ -m_2l_1 & m_2 \end{pmatrix} \\
&\qquad \cdot \begin{pmatrix} -c_1 & -s_1 \\ -l_1c_1-(q_2+d_2)s_1 & (q_2+d_2)c_1-l_1s_1 \end{pmatrix},
\end{aligned}
$$

with elements

$$
\begin{aligned}
\boldsymbol{M}_{\boldsymbol{p},11} &= \frac{1}{(q_2+d_2)^2}\Big(I_1+I_2+m_1d_1^2+m_2q_2^2+\left(2m_2d_2q_2+m_2d_2^2-I_1-I_2-m_1d_1^2\right)s_1^2\Big), \\
\boldsymbol{M}_{\boldsymbol{p},12} = \boldsymbol{M}_{\boldsymbol{p},21} &= \frac{1}{(q_2+d_2)^2}\Big(I_1+I_2+m_1d_1^2-m_2d_2^2-2m_2d_2q_2\Big)s_1c_1, \\
\boldsymbol{M}_{\boldsymbol{p},22} &= \frac{1}{(q_2+d_2)^2}\Big(m_2(q_2+d_2)^2+\left(I_1+I_2+m_1d_1^2-m_2d_2^2-2m_2d_2q_2\right)s_1^2\Big).
\end{aligned}
$$

The Cartesian inertia matrix is not diagonal in general. However, when evaluating $\boldsymbol{M}_{\boldsymbol{p}}(\boldsymbol{q}^*)$ for $\boldsymbol{q}^* = \begin{pmatrix} 0 & 3 \end{pmatrix}^T$ [rad, m] and using the numerical parameters given in the text, we obtain

$$
\begin{aligned}
\boldsymbol{M}_{\boldsymbol{p}}(\boldsymbol{q}^*) &= \boldsymbol{J}^{-T}(\boldsymbol{q}^*)\boldsymbol{M}(\boldsymbol{q}^*)\boldsymbol{J}^{-1}(\boldsymbol{q}^*) \\
&= \begin{pmatrix} 0.2857 & 0.2857 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} 6.625 & -0.5 \\ -0.5 & 0.5 \end{pmatrix}\begin{pmatrix} 0.2857 & 0 \\ 0.2857 & -1 \end{pmatrix} = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix} = 0.5\cdot \boldsymbol{I}_{2\times 2}
\end{aligned}
$$

As a result, we can immediately verify that the Cartesian acceleration $\ddot{\boldsymbol{p}} \in \mathbb{R}^2$ of the robot end effector in response to an arbitrary force $\boldsymbol{F} \in \mathbb{R}^2$ applied at the tip when the robot is at rest in $\boldsymbol{q}^*$ and in the absence of gravity is

$$\ddot{\boldsymbol{p}} = \boldsymbol{M}_{\boldsymbol{p}}^{-1}(\boldsymbol{q}^*)\boldsymbol{F} = 2\boldsymbol{F},$$

namely, $\ddot{\boldsymbol{p}}$ has the same direction of the applied force $\boldsymbol{F}$.

Exercise #4

The 3R robot (having $N = 3$ degrees of freedom) is geometrically constrained in its Cartesian motion in two ways:

- it cannot change the absolute orientation of the last link, which remains always parallel to $\boldsymbol{y}_0$;
- it cannot move the tip of the second link away from the axis $x = k$.

Therefore, this situation can be modeled by $M = 2$ scalar constraints, written in the joint space as

$$\boldsymbol{h}(\boldsymbol{q}) = \begin{pmatrix} q_1+q_2+q_3-\dfrac{\pi}{2} \\ l_1\cos q_1 + l_2\cos(q_1+q_2) - k \end{pmatrix} = \boldsymbol{0}.$$

The Jacobian of these constraints is

$$\boldsymbol{A}(\boldsymbol{q}) = \frac{\partial \boldsymbol{h}(\boldsymbol{q})}{\partial \boldsymbol{q}} = \begin{pmatrix} 1 & 1 & 1 \\ -\left(l_1\sin q_1 + l_2\sin(q_1+q_2)\right) & -l_2\sin(q_1+q_2) & 0 \end{pmatrix}.$$

The rank of matrix $\boldsymbol{A}(\boldsymbol{q})$ is 2, except when $\sin(q_1+q_2) = \sin q_1 = 0$, which occur if and only if $q_1 = \{0, \pi\}$ and $q_2 = \{0, \pi\}$, i.e., when the first two links are stretched or folded along the $\boldsymbol{x}_0$-axis. However, these singular configurations are not allowed by the geometric constraints (thanks to the two inequalities imposed on the parameter k, otherwise arbitrary).

The first step for deriving the reduced dynamics of this constrained robot is to find a matrix $\boldsymbol{D}(\boldsymbol{q})$ of size $(N-M)\times M = 1\times 3$ such that it completes $\boldsymbol{A}(\boldsymbol{q})$, building a nonsingular square matrix in the operating region. A suitable choice that satisfies this requirement is

$$\boldsymbol{D}(\boldsymbol{q}) = \begin{pmatrix} l_1\cos q_1 + l_2\cos(q_1+q_2) & l_2\cos(q_1+q_2) & 0 \end{pmatrix},$$

which leads to

$$\begin{pmatrix} \boldsymbol{A}(\boldsymbol{q}) \\ \boldsymbol{D}(\boldsymbol{q}) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ -\left(l_1\sin q_1 + l_2\sin(q_1+q_2)\right) & -l_2\sin(q_1+q_2) & 0 \\ l_1\cos q_1 + l_2\cos(q_1+q_2) & l_2\cos(q_1+q_2) & 0 \end{pmatrix}, \quad \det \begin{pmatrix} \boldsymbol{A}(\boldsymbol{q}) \\ \boldsymbol{D}(\boldsymbol{q}) \end{pmatrix} = l_1 l_2 \sin q_2. \tag{4}$$

The determinant is never zero for all $\boldsymbol{q}$ such that the contact situation remains the same as in Fig. 3. As a matter of fact, this choice of $\boldsymbol{D}(\boldsymbol{q})$ reconstructs the 2×2 Jacobian of the 2R substructure made by the first two links of the 3R robot. The constrained robot has only one degree of freedom left, which is described by the scalar term

$$v = \boldsymbol{D}(\boldsymbol{q})\dot{\boldsymbol{q}} = \left(l_1\cos q_1 + l_2\cos(q_1+q_2)\right)\dot{q}_1 + l_2\cos(q_1+q_2)\,\dot{q}_2.$$

This pseudovelocity represents the motion of the tip of the second link along the direction $\boldsymbol{y}_0$.

The second step of the procedure is to invert the matrix in (4) so as to define the blocks $\boldsymbol{E}(\boldsymbol{q})$ and $\boldsymbol{F}(\boldsymbol{q})$ in the inverse. We obtain

$$\begin{pmatrix} \boldsymbol{A}(\boldsymbol{q}) \\ \boldsymbol{D}(\boldsymbol{q}) \end{pmatrix}^{-1} = \frac{1}{l_1 l_2 \sin q_2} \begin{pmatrix} 0 & l_2\cos(q_1+q_2) & l_2\sin(q_1+q_2) \\ 0 & -\left(l_1\cos q_1 + l_2\cos(q_1+q_2)\right) & -\left(l_1\sin q_1 + l_2\sin(q_1+q_2)\right) \\ l_1 l_2\sin q_2 & l_1\cos q_1 & l_1\sin q_1 \end{pmatrix}.$$

Thus, we have the partition into the first $M = 2$ columns

$$\boldsymbol{E}(\boldsymbol{q}) = \frac{1}{l_1 l_2 \sin q_2} \begin{pmatrix} 0 & l_2\cos(q_1+q_2) \\ 0 & -\left(l_1\cos q_1 + l_2\cos(q_1+q_2)\right) \\ l_1 l_2\sin q_2 & l_1\cos q_1 \end{pmatrix}$$

and the last $N-M = 1$ column

$$\boldsymbol{F}(\boldsymbol{q}) = \frac{1}{l_1 l_2 \sin q_2} \begin{pmatrix} l_2\sin(q_1+q_2) \\ -\left(l_1\sin q_1 + l_2\sin(q_1+q_2)\right) \\ l_1\sin q_1 \end{pmatrix}.$$

Finally, introducing in symbolic form the elements of the robot inertia of the 3R robot in free space

$$\boldsymbol{M}(\boldsymbol{q}) = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{12} & m_{22} & m_{23} \\ m_{13} & m_{23} & m_{33} \end{pmatrix} > 0,$$

we have that the reduced inertia matrix is a scalar given by[2]

$$
\begin{aligned}
\boldsymbol{F}^T(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})\boldsymbol{F}(\boldsymbol{q}) &= \boldsymbol{F}^T(\boldsymbol{q}) \cdot \frac{1}{l_1 l_2 \sin q_2} \begin{pmatrix} (m_{11} - m_{12})\, l_2 \sin(q_1+q_2) + (m_{13} - m_{12})\, l_1 \sin q_1 \\ (m_{12} - m_{22})\, l_2 \sin(q_1+q_2) + (m_{23} - m_{22})\, l_1 \sin q_1 \\ (m_{13} - m_{23})\, l_2 \sin(q_1+q_2) + (m_{33} - m_{23})\, l_1 \sin q_1 \end{pmatrix}. \\
&= \frac{1}{(l_1 l_2 \sin q_2)^2} \Big((m_{11} + m_{22} - 2m_{12})\, l_2^2 \sin^2(q_1+q_2) \\
&\qquad + (m_{22} + m_{33} - 2m_{23})\, l_1^2 \sin^2 q_1 \\
&\qquad + 2\,(m_{22} - m_{12} + m_{13} - m_{23})\, l_1 l_2 \sin q_1 \sin(q_1+q_2) \Big).
\end{aligned}
$$

Exercise #5

The trajectory tracking control law based on feedback linearization is

$$\boldsymbol{u}_{FBL} = \boldsymbol{M}(\boldsymbol{q})\Big(\ddot{\boldsymbol{q}}_d + \boldsymbol{K}_D(\dot{\boldsymbol{q}}_d - \dot{\boldsymbol{q}}) + \boldsymbol{K}_P(\boldsymbol{q}_d - \boldsymbol{q})\Big) + \boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}). \tag{5}$$

The Lyapunov-based control law with global asymptotic convergence property is

$$\boldsymbol{u}_{GLB} = \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}_d + \boldsymbol{S}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_d + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{K}_D(\dot{\boldsymbol{q}}_d - \dot{\boldsymbol{q}}) + \boldsymbol{K}_P(\boldsymbol{q}_d - \boldsymbol{q}), \tag{6}$$

where $\dot{\boldsymbol{M}} - 2\boldsymbol{S}$ is a skew symmetric matrix and the PD gains are by hypothesis the same as in (5). Since the Coriolis and centrifugal terms in (5) can always be rewritten as $\boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{S}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}$ using the same factorization used in (6), the difference between the two control torques can be written in general as

$$\boldsymbol{\Delta u} = \boldsymbol{u}_{FBL} - \boldsymbol{u}_{GLB} = (\boldsymbol{M}(\boldsymbol{q}) - \boldsymbol{I})\,(\boldsymbol{K}_D(\dot{\boldsymbol{q}}_d - \dot{\boldsymbol{q}}) + \boldsymbol{K}_P(\boldsymbol{q}_d - \boldsymbol{q})) + \boldsymbol{S}(\boldsymbol{q}, \dot{\boldsymbol{q}})\,(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_d)\,. \tag{7}$$

From the desired joint trajectory, we obtain

$$\dot{\boldsymbol{q}}_d(t) = \begin{pmatrix} \dfrac{3\pi}{2}\cos\dfrac{\pi t}{2} \\ 2\pi \sin 2\pi t \end{pmatrix}, \qquad \ddot{\boldsymbol{q}}_d(t) = \begin{pmatrix} -\dfrac{3\pi^2}{4}\sin\dfrac{\pi t}{2} \\ 4\pi^2 \cos 2\pi t \end{pmatrix}.$$

At time $t = 2$ s, we have thus

$$\boldsymbol{q}_d(2) = \begin{pmatrix} \dfrac{\pi}{2} \\ 0 \end{pmatrix}, \qquad \dot{\boldsymbol{q}}_d(2) = \begin{pmatrix} -\dfrac{3\pi}{2} \\ 0 \end{pmatrix}, \qquad \ddot{\boldsymbol{q}}_d(2) = \begin{pmatrix} 0 \\ 4\pi^2 \end{pmatrix},$$

while the robot state and the position and velocity errors are

$$\boldsymbol{q}(2) = \begin{pmatrix} \dfrac{\pi}{2} \\ -\dfrac{\pi}{2} \end{pmatrix} \quad \Rightarrow \quad \boldsymbol{e}(2) = \begin{pmatrix} 0 \\ \dfrac{\pi}{2} \end{pmatrix}, \qquad \dot{\boldsymbol{q}}(2) = \dot{\boldsymbol{q}}_d(2) \quad \Rightarrow \quad \dot{\boldsymbol{e}}(2) = \boldsymbol{0}.$$

In this case, the only information needed in eq. (7) is the inertia matrix of the 2R robot. From the lecture slides, this matrix has the form

$$\boldsymbol{M}(\boldsymbol{q}) = \begin{pmatrix} a_1 + 2a_2\cos q_2 & a_3 + a_2 \cos q_2 \\ a_3 + a_2 \cos q_2 & a_3 \end{pmatrix},$$

[2]Performing computations by hand in the given sequence is surprisingly faster than setting up a similarly efficient code using symbolic programming!

with dynamic coefficients $a_i > 0$, $i = 1, 2, 3$. Using the PD gains given in the text, we finally obtain

$$\boldsymbol{\Delta u}(2) = (\boldsymbol{M}(\boldsymbol{q}(2)) - \boldsymbol{I}) \cdot \boldsymbol{K}_P\, \boldsymbol{e}(2) = \begin{pmatrix} a_1 - 1 & a_3 \\ a_3 & a_3 - 1 \end{pmatrix} \cdot 100 \begin{pmatrix} 0 \\ \dfrac{\pi}{2} \end{pmatrix} = 50\pi \cdot \begin{pmatrix} a_3 \\ a_3 - 1 \end{pmatrix}.$$

It is quite reasonably to assume that $a_3 = I_2 + m_2 d_{c2}^2 > 1$, being the robot arm of human-like size and weight. Thus, both components of $\boldsymbol{\Delta u}(2)$ are positive. However, to determine which controller is using the larger torques in absolute value at $t = 2$ s, we need to assess also the signs of the components of at least one of the two torque commands[3]. We evaluate then the Lyapunov-based tracking controller under the assumed conditions, obtaining

$$\boldsymbol{u}_{GLB}(2) = \boldsymbol{M}(\boldsymbol{q}(2))\, \ddot{\boldsymbol{q}}_d(2) + \boldsymbol{S}(\boldsymbol{q}(2), \dot{\boldsymbol{q}}_d(2))\, \dot{\boldsymbol{q}}_d(2) + \boldsymbol{g}(\boldsymbol{q}(2)) + \boldsymbol{K}_P\, \boldsymbol{e}(2).$$

For each term in the expression of $\boldsymbol{u}_{GLB}(2)$, the following can be easily observed:

$$\boldsymbol{M}(\boldsymbol{q}(2))\, \ddot{\boldsymbol{q}}_d(2) = \begin{pmatrix} a_1 & a_3 \\ a_3 & a_3 \end{pmatrix} \begin{pmatrix} 0 \\ 4\pi^2 \end{pmatrix} = 4\pi^2 a_3 \begin{pmatrix} 1 \\ 1 \end{pmatrix} > \boldsymbol{0},$$

$$\boldsymbol{S}(\boldsymbol{q}(2), \dot{\boldsymbol{q}}_d(2))\, \dot{\boldsymbol{q}}_d(2) = \boldsymbol{c}(\boldsymbol{q}(2), \dot{\boldsymbol{q}}_d(2)) = \begin{pmatrix} -a_2 \sin q_2(2) \left(\dot{q}_{d2}^2(2) - 2\, \dot{q}_{d1}(2) \dot{q}_{d2}(2) \right) \\ a_2 \sin q_2(2)\, \dot{q}_{d1}^2(2) \end{pmatrix}$$

$$= -\frac{9\pi^2}{4}\, a_2 \begin{pmatrix} 0 \\ 1 \end{pmatrix} \leq \boldsymbol{0},$$

$$\boldsymbol{g}(\boldsymbol{q}(2)) = \begin{pmatrix} a_4 \cos q_1(2) + a_5 \cos(q_1(2) + q_2(2)) \\ a_5 \cos(q_1(2) + q_2(2)) \end{pmatrix} = a_5 \begin{pmatrix} 1 \\ 1 \end{pmatrix} > \boldsymbol{0},$$

$$\boldsymbol{K}_P\, \boldsymbol{e}(2) = 50 \begin{pmatrix} 0 \\ \pi \end{pmatrix} \geq \boldsymbol{0}.$$

Despite of the negative addend in the second component of the velocity term, it can be safely concluded that this single term is compensated by the multiple other positive ones, so that $\boldsymbol{u}_{GLB}(2) > \boldsymbol{0}$ holds componentwise. Thus, it is also $\boldsymbol{u}_{FBL}(2) = \boldsymbol{u}_{GLB}(2) + \boldsymbol{\Delta u}(2) > \boldsymbol{0}$ componentwise. For both two components, the feedback linearization law requires at $t = 2$ s a larger torque (in absolute value, but in fact positive) than the Lyapunov-based control law.

* * * * *

[3] Suppose that $\Delta u = a - b > 0$. If $b > 0$, both a and b will be positive and a is certainly larger than b. If instead $b < 0$, we could have both $|b| > |a|$ or viceversa in absolute value, and thus also the sign of a should be checked.