Robotics 2

Midterm Test – April 13, 2022

Exercise #1

We need to calibrate the link lengths of a planar 2R robot, whose nominal values are $\hat{l}_1 = \hat{l}_2 = 1$ [m]. All other kinematic parameters are assumed to be good enough. At four different Denavit-Hartenberg configurations $\boldsymbol{q}$, the following data (in [m]) for the position $\boldsymbol{p} \in \mathbb{R}^2$ of the robot end-effector are collected by an accurate external measurement system:

$$\begin{array}{lcl} \boldsymbol{q}_a = (0,0) & \Rightarrow & \boldsymbol{p}_a = (2,0) \\ \boldsymbol{q}_b = (\pi/2,0) & \Rightarrow & \boldsymbol{p}_b = (0,2) \\ \boldsymbol{q}_c = (\pi/4,-\pi/4) & \Rightarrow & \boldsymbol{p}_c = (1.6925, 0.7425) \\ \boldsymbol{q}_d = (0,\pi/4) & \Rightarrow & \boldsymbol{p}_d = (1.7218, 0.6718). \end{array}$$

Provide the best estimate of the actual lengths l_1 and l_2 of the two robot links, using the above information. Is this calibration problem linear or nonlinear?

Exercise #2

A robot is driven by joint acceleration commands $\ddot{\boldsymbol{q}} \in \mathbb{R}^n$ which are kept constant for a (sufficiently small) sampling time T_c, i.e., $\ddot{\boldsymbol{q}}(t) = \ddot{\boldsymbol{q}}_k$, for $t \in [t_k, t_{k+1}) = [t_k, t_k + T_c)$. Thus, the next velocity at time $t = t_{k+1}$ can be expressed as $\dot{\boldsymbol{q}}_{k+1} = \dot{\boldsymbol{q}}(t_{k+1}) = \dot{\boldsymbol{q}}_k + T_c\, \ddot{\boldsymbol{q}}_k$. At time $t = t_k$, the robot is in the state $(\boldsymbol{q}_k, \dot{\boldsymbol{q}}_k)$ and has to realize a desired task acceleration $\ddot{\boldsymbol{r}}_{d,k} \in \mathbb{R}^m$, with $m < n$, being the task function $\boldsymbol{r} = \boldsymbol{f}(\boldsymbol{q})$. What is the expression of the command $\ddot{\boldsymbol{q}}_k$ that executes the task while minimizing the squared norm of the joint velocity at the *next* sampled instant t_{k+1}?

Exercise #3

Consider the spatial 3R robot in Fig. 1. Using the D-H generalized coordinates defined therein, compute the robot inertia matrix $\boldsymbol{M}(\boldsymbol{q})$. Assume that the links have their center of mass on $\boldsymbol{x}_1$, $\boldsymbol{y}_2$, and $\boldsymbol{x}_3$, respectively, and that the barycentric link inertia matrices are diagonal, i.e., ${}^i\boldsymbol{I}_{ci} = \text{diag}\,\{I_{ci,xx}, I_{ci,yy}, I_{ci,zz}\}$, $i = 1,2,3$.

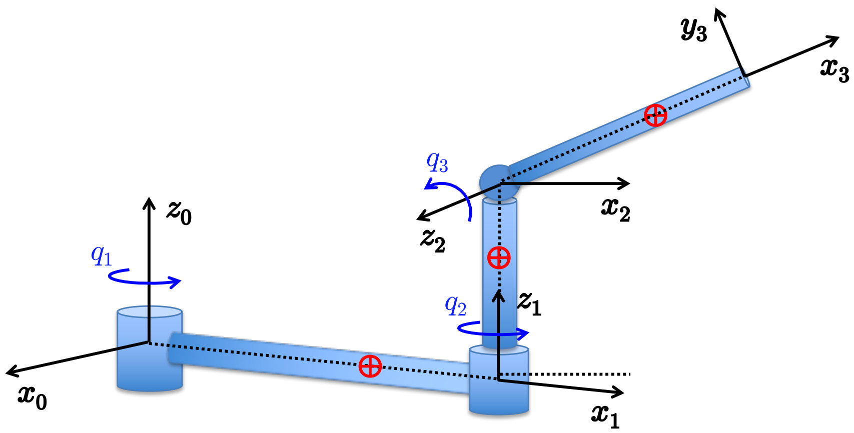


Figure 1: A spatial 3R robot, with D-H frames assigned to each link.

Exercise #4

A planar 3R robot with unitary link lengths is commanded by a joint velocity $\dot{\boldsymbol{q}} \in \mathbb{R}^3$ with components bounded as $|\dot{q}_i| \leq 2$ [rad/s], $i = 1, 2, 3$. The D-H joint variables have limited ranges specified by

$$q_1 \in [-\pi/2, \pi/2], \qquad q_2 \in [0, 2\pi/3], \qquad q_3 \in [-\pi/4, \pi/4].$$

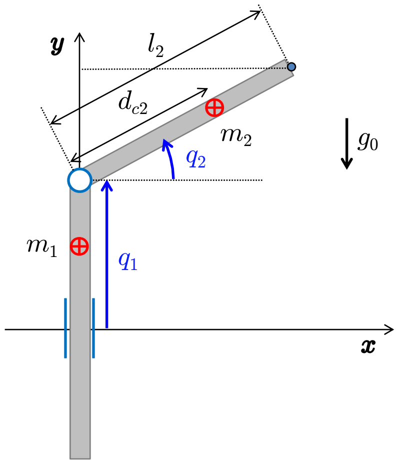
At the configuration $\widehat{\boldsymbol{q}} = (2\pi/5, \pi/2, -\pi/4)$, the robot should move its end-effector horizontally with a speed $v_x = -3$ [m/s], while trying to keep the joints close to their midranges. Compute the value of the instantaneous joint velocity $\dot{\boldsymbol{q}}$ that performs the Cartesian task while improving at best the criterion $H_{range}(\boldsymbol{q})$. Check if this joint velocity is feasible and, if not, perform the least end-effector task scaling to recover feasibility.

Exercise #5

Figure 2 shows a PR robot and its inertia matrix, already expressed in terms of three dynamic coefficients a, b and c. The robot moves in a vertical plane. A task trajectory $y_d(t) \in \mathbb{R}$ is assigned to the coordinate y of the end-effector position. With the robot being at rest in the configuration $\bar{\boldsymbol{q}} = \begin{pmatrix} 1 & \pi/2 \end{pmatrix}^T$, provide the joint force/torque inputs $\boldsymbol{\tau}_A \in \mathbb{R}^2$ and $\boldsymbol{\tau}_B \in \mathbb{R}^2$ executing the desired task that instantaneously minimize, respectively,

$$H_A = \frac{1}{2}\|\boldsymbol{\tau}\|^2 \qquad \text{or} \qquad H_B = \frac{1}{2}\|\boldsymbol{\tau}\|^2_{\boldsymbol{M}^{-2}(\bar{\boldsymbol{q}})}.$$

Which of the two solutions $\boldsymbol{\tau}_A$ and $\boldsymbol{\tau}_B$ has the largest first component in absolute value?



$$\boldsymbol{M}(\boldsymbol{q}) = \begin{pmatrix} a & b\cos q_2 \\ b\cos q_2 & c \end{pmatrix} > 0$$

Figure 2: A planar PR robot and its inertia matrix.

Exercise #6

For the same PR robot in Fig. 2, determine the gravity term $\boldsymbol{g}(\boldsymbol{q})$ in the dynamic model and define a tight upper bound $\alpha > 0$ on the norm of the square matrix $\partial \boldsymbol{g}(\boldsymbol{q})/\partial \boldsymbol{q}$, for any value of $\boldsymbol{q}$.

[180 minutes (3 hours); open books]

Solution

April 13, 2022

Exercise #1

This calibration task is formulated as a *linear* least squares problem. In fact, the relevant measurement equations for the planar 2R robot can be written as

$$\Delta\boldsymbol{p} = \boldsymbol{p} - \hat{\boldsymbol{p}} = \begin{pmatrix} l_1 c_1 + l_2 c_{12} \\ l_1 s_1 + l_2 s_{12} \end{pmatrix} - \begin{pmatrix} \hat{l}_1 c_1 + \hat{l}_2 c_{12} \\ \hat{l}_1 s_1 + \hat{l}_2 s_{12} \end{pmatrix} = \begin{pmatrix} \Delta l_1\, c_1 + \Delta l_2\, c_{12} \\ \Delta l_1\, s_1 + \Delta l_2\, s_{12} \end{pmatrix} = \begin{pmatrix} c_1 & c_{12} \\ s_1 & s_{12} \end{pmatrix} \begin{pmatrix} \Delta l_1 \\ \Delta l_2 \end{pmatrix},$$

or

$$\Delta\boldsymbol{p} = \boldsymbol{\Phi}(\boldsymbol{q})\,\Delta\boldsymbol{l}, \qquad \text{with } \boldsymbol{\Phi}(\boldsymbol{q}) = \begin{pmatrix} c_1 & c_{12} \\ s_1 & s_{12} \end{pmatrix},$$

without the need of any local approximation because the link lengths appear linearly in the direct kinematics of the robot. From the nominal model, we compute in the chosen configurations

$$\hat{\boldsymbol{p}}_a = \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \qquad \hat{\boldsymbol{p}}_b = \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \qquad \hat{\boldsymbol{p}}_c = \begin{pmatrix} 1.7071 \\ 0.7071 \end{pmatrix}, \qquad \hat{\boldsymbol{p}}_d = \begin{pmatrix} 1.7071 \\ 0.7071 \end{pmatrix}.$$

Note that the first two nominal positions of the end-effector correspond to the measured ones. Stacking the results of the four experiments, we obtain the overdetermined linear system of equations

$$\Delta\bar{\boldsymbol{p}} = \begin{pmatrix} \Delta\boldsymbol{p}_a \\ \Delta\boldsymbol{p}_b \\ \Delta\boldsymbol{p}_c \\ \Delta\boldsymbol{p}_c \end{pmatrix} = \begin{pmatrix} \boldsymbol{\Phi}(\boldsymbol{q}_a) \\ \boldsymbol{\Phi}(\boldsymbol{q}_b) \\ \boldsymbol{\Phi}(\boldsymbol{q}_c) \\ \boldsymbol{\Phi}(\boldsymbol{q}_d) \end{pmatrix} \Delta\boldsymbol{l} = \bar{\boldsymbol{\Phi}}\,\Delta\boldsymbol{l},$$

or

$$\Delta\bar{\boldsymbol{p}} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -0.0146 \\ 0.0354 \\ 0.0146 \\ -0.0354 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \\ 0.7071 & 1 \\ 0.7071 & 0 \\ 1 & 0.7071 \\ 0 & 0.7071 \end{pmatrix} \Delta\boldsymbol{l} = \bar{\boldsymbol{\Phi}}\,\Delta\boldsymbol{l}.$$

By pseudoinversion of the 8×2 matrix $\bar{\boldsymbol{\Phi}}$, we obtain the value that minimizes the estimation error in a least squares sense,

$$\Delta\boldsymbol{l} = \bar{\boldsymbol{\Phi}}^{\#}\Delta\bar{\boldsymbol{p}} = \begin{pmatrix} 0.05 \\ -0.05 \end{pmatrix} = \begin{pmatrix} \Delta l_1 \\ \Delta l_2 \end{pmatrix}. \tag{1}$$

Therefore, the resulting estimates of the lengths of the two links are

$$l_1 = \hat{l}_1 + \Delta l_1 = 1.05, \qquad l_2 = \hat{l}_2 + \Delta l_2 = 0.95 \qquad \text{[m]}.$$

We finally note that the second and third regressor equations provide no information (all zeros!), whereas the fourth equation is a repetition of the first one. These phenomena are related to the singularity of the $\boldsymbol{\Phi}(\boldsymbol{q})$ matrix when $\sin q_2 = 0$ (e.g., in the configurations $\boldsymbol{q}_a$ and $\boldsymbol{q}_b$ —not the best choices for calibration!). Therefore, these rows can be safely eliminated from the computation without any change in the final result.

Exercise #2

We are in the presence of redundancy ($m < n$). The objective function to be minimized at time $t = t_k$ is a complete quadratic function of the joint acceleration $\ddot{\boldsymbol{q}}_k$, the input to be chosen. We have

$$H(\ddot{\boldsymbol{q}}_k) = \frac{1}{2}\left\|\dot{\boldsymbol{q}}_{k+1}\right\|^2 = \frac{1}{2}\left\|\dot{\boldsymbol{q}}_k + T_c\,\ddot{\boldsymbol{q}}_k\right\|^2 = \frac{T_c^2}{2}\,\ddot{\boldsymbol{q}}_k^T\ddot{\boldsymbol{q}}_k + T_c\,\dot{\boldsymbol{q}}_k^T\ddot{\boldsymbol{q}}_k + c,$$

with the constant $c = \frac{1}{2}\,\dot{\boldsymbol{q}}_k^T\dot{\boldsymbol{q}}_k$. The unconstrained minimization of $H(\ddot{\boldsymbol{q}}_k)$ would yield the *preferred* acceleration $\ddot{\boldsymbol{q}}_k = -\dot{\boldsymbol{q}}_k/T_c$, which produces in fact a zero value for the non-negative objective function H. However, the required robot task is expressed by imposing the equality constraint

$$\boldsymbol{J}(\boldsymbol{q}_k)\,\ddot{\boldsymbol{q}}_k = \ddot{\boldsymbol{r}}_{d,k} - \dot{\boldsymbol{J}}(\boldsymbol{q}_k)\dot{\boldsymbol{q}}_k,$$

which is linear in the joint acceleration. Thus, the problem is in the standard form of LQ optimization and the solution is found by applying the general formula with $\boldsymbol{x} = \ddot{\boldsymbol{q}}_k$, $\boldsymbol{W} = T_c^2\,\boldsymbol{I}$, $\boldsymbol{x}_0 = -\dot{\boldsymbol{q}}_k/T_c$, and $\boldsymbol{y} = \ddot{\boldsymbol{r}}_{d,k} - \dot{\boldsymbol{J}}(\boldsymbol{q}_k)\dot{\boldsymbol{q}}_k$ (see the slides). Assuming a full rank Jacobian, we obtain

$$\begin{aligned}
\ddot{\boldsymbol{q}}_k &= -\frac{\dot{\boldsymbol{q}}_k}{T_c} + \frac{1}{T_c^2}\,\boldsymbol{J}^T(\boldsymbol{q}_k)\left(\frac{1}{T_c^2}\,\boldsymbol{J}(\boldsymbol{q}_k)\,\boldsymbol{J}^T(\boldsymbol{q}_k)\right)^{-1}\left(\ddot{\boldsymbol{r}}_{d,k} - \dot{\boldsymbol{J}}(\boldsymbol{q}_k)\dot{\boldsymbol{q}}_k - \boldsymbol{J}(\boldsymbol{q}_k)\left(-\frac{\dot{\boldsymbol{q}}_k}{T_c}\right)\right)\\
&= -\frac{\dot{\boldsymbol{q}}_k}{T_c} + \boldsymbol{J}^T(\boldsymbol{q}_k)\left(\boldsymbol{J}(\boldsymbol{q}_k)\,\boldsymbol{J}^T(\boldsymbol{q}_k)\right)^{-1}\left(\ddot{\boldsymbol{r}}_{d,k} - \dot{\boldsymbol{J}}(\boldsymbol{q}_k)\dot{\boldsymbol{q}}_k + \boldsymbol{J}(\boldsymbol{q}_k)\frac{\dot{\boldsymbol{q}}_k}{T_c}\right)\\
&= \boldsymbol{J}^{\#}(\boldsymbol{q}_k)\left(\ddot{\boldsymbol{r}}_{d,k} - \dot{\boldsymbol{J}}(\boldsymbol{q}_k)\dot{\boldsymbol{q}}_k\right) - \left(\boldsymbol{I} - \boldsymbol{J}^{\#}(\boldsymbol{q}_k)\,\boldsymbol{J}(\boldsymbol{q}_k)\right)\frac{\dot{\boldsymbol{q}}_k}{T_c}.
\end{aligned} \tag{2}$$

Exercise #3

We compute the kinetic energy of the three links. Denote by m_i the mass of link i, by l_i its length (i.e., the parameter d_i or a_i of the D-H convention), and by ${}^i\boldsymbol{I}_{ci} = \operatorname{diag}\{I_{ci,xx}, I_{ci,yy}, I_{ci,zz}\}$ its inertia matrix, for $i = 1, 2, 3$. Moreover, let $d_{ci} > 0$ be the distance of the center of mass (CoM) of link i from the axis of joint i; because of the assumption on the location of the CoM of each link, only one scalar is needed for each link[1].

Link 1

$$T_1 = \frac{1}{2}\left(I_{c1,zz} + m_1 d_{c1}^2\right)\dot{q}_1^2.$$

Link 2

$$T_2 = \frac{1}{2}\,m_2 l_1^2\,\dot{q}_1^2 + \frac{1}{2}\,I_{c2,yy}\left(\dot{q}_1 + \dot{q}_2\right)^2.$$

Link 3

$$\boldsymbol{p}_{c3} = \begin{pmatrix} l_1 c_1 + d_{c3} c_3 c_{12} \\ l_1 s_1 + d_{c3} c_3 s_{12} \\ l_2 + d_{c3} s_3 \end{pmatrix} \quad\Rightarrow\quad \boldsymbol{v}_{c3} = \dot{\boldsymbol{p}}_{c3} = \begin{pmatrix} -\left(l_1 s_1 \dot{q}_1 + d_{c3} c_3 s_{12}\left(\dot{q}_1 + \dot{q}_2\right) + d_{c3} s_3 c_{12} \dot{q}_3\right) \\ l_1 c_1 \dot{q}_1 + d_{c3} c_3 c_{12}\left(\dot{q}_1 + \dot{q}_2\right) - d_{c3} s_3 s_{12} \dot{q}_3 \\ d_{c3} c_3 \dot{q}_3 \end{pmatrix}$$

$${}^1\boldsymbol{\omega}_1 = \begin{pmatrix} 0 \\ 0 \\ \dot{q}_1 \end{pmatrix} \;\Rightarrow\; {}^2\boldsymbol{\omega}_2 = \begin{pmatrix} 0 \\ \dot{q}_1 + \dot{q}_2 \\ 0 \end{pmatrix} \;\Rightarrow\; {}^3\boldsymbol{\omega}_3 = {}^2\boldsymbol{R}_3^T(q_3)\left({}^2\boldsymbol{\omega}_2 + \begin{pmatrix} 0 \\ 0 \\ \dot{q}_3 \end{pmatrix}\right) = \begin{pmatrix} s_3\left(\dot{q}_1 + \dot{q}_2\right) \\ c_3\left(\dot{q}_1 + \dot{q}_2\right) \\ \dot{q}_3 \end{pmatrix}$$

[1] If using the moving frames algorithm for the computation of ${}^i\boldsymbol{v}_{ci}$ in the kinetic energy, it will be convenient to define the constant vectors of CoM positions in each of the local frame as follows: ${}^1\boldsymbol{r}_{c1} = (-l_1 + d_{c1}, 0, 0)$, ${}^2\boldsymbol{r}_{c2} = (0, -l_2 + d_{c2}, 0)$ —although this is not relevant in ${}^2\boldsymbol{v}_{c2}$, and ${}^3\boldsymbol{r}_{c3} = (-l_3 + d_{c3}, 0, 0)$. These symbolic choices in the recursive algorithm provide the same result as with the direct computations used in the text.

$$
\begin{aligned}
T_3 &= \frac{1}{2}\, m_3\, \boldsymbol{v}_{c3}^T \boldsymbol{v}_{c3} + \frac{1}{2}\, {}^3\boldsymbol{\omega}_3^T\, {}^3\boldsymbol{I}_{c3}\, {}^3\boldsymbol{\omega}_3 \\
&= \frac{1}{2} m_3 \left(l_1^2 \dot{q}_1^2 + d_{c3}^3 c_3^2 \left(\dot{q}_1 + \dot{q}_2\right)^2 + 2 l_1 d_{c3} \left(c_2 c_3\, \dot{q}_1 \left(\dot{q}_1 + \dot{q}_2\right) - s_2 s_3\, \dot{q}_1 \dot{q}_3 \right) \right) \\
&\quad + \frac{1}{2} \left(I_{c3,xx} s_3^2 + I_{c3,yy} c_3^2 \right) \left(\dot{q}_1 + \dot{q}_2\right)^2 + \frac{1}{2} I_{c3,zz}\, \dot{q}_3^2 .
\end{aligned}
$$

Inertia matrix

$$
\boldsymbol{M}(\boldsymbol{q}) = \begin{pmatrix} m_{11}(q_2, q_3) & m_{12}(q_2, q_3) & m_{13}(q_2, q_3) \\ m_{12}(q_2, q_3) & m_{22}(q_3) & 0 \\ m_{13}(q_2, q_3) & 0 & m_{33} \end{pmatrix} \tag{3}
$$

with

$$
\begin{aligned}
m_{11}(q_2, q_3) &= I_{c1,zz} + m_1 d_{c1}^2 + I_{c2,yy} + (m_2 + m_3) l_1^2 + m_3 d_{c3}^2 c_3^2 + 2 m_3 l_1 d_{c3}\, c_2 c_3 + \left(I_{c3,xx} s_3^2 + I_{c3,yy} c_3^2 \right) \\
m_{12}(q_2, q_3) &= I_{c2,yy} + m_3 d_{c3}^2 c_3^2 + m_3 l_1 d_{c3}\, c_2 c_3 + \left(I_{c3,xx} s_3^2 + I_{c3,yy} c_3^2 \right) \\
m_{13}(q_2, q_3) &= -m_3 l_1 d_{c3}\, s_2 s_3 \\
m_{22}(q_3) &= I_{c2,yy} + m_3 d_{c3}^2 c_3^2 + \left(I_{c3,xx} s_3^2 + I_{c3,yy} c_3^2 \right) \\
m_{33} &= I_{c3,zz} + m_3 d_{c3}^2 .
\end{aligned}
$$

Note finally that one can remove the presence of s_3^2 by replacing it everywhere with $(1 - c_3^2)$. This is also what MATLAB does when applying a `simplify` instruction to the symbolic expressions. The affected elements of $\boldsymbol{M}(\boldsymbol{q})$ become then

$$
\begin{aligned}
m_{11}(q_2, q_3) &= I_{c1,zz} + m_1 d_{c1}^2 + I_{c2,yy} + (m_2 + m_3) l_1^2 + I_{c3,xx} + 2 m_3 l_1 d_{c3}\, c_2 c_3 + \left(I_{c3,yy} + m_3 d_{c3}^2 - I_{c3,xx} \right) c_3^2 \\
m_{12}(q_2, q_3) &= I_{c2,yy} + I_{c3,xx} + m_3 l_1 d_{c3}\, c_2 c_3 + \left(I_{c3,yy} + m_3 d_{c3}^2 c_3^2 - I_{c3,xx} \right) c_3^2 \\
m_{22}(q_3) &= I_{c2,yy} + I_{c3,xx} + \left(I_{c3,yy} + m_3 d_{c3}^2 - I_{c3,xx} \right) c_3^2 .
\end{aligned}
$$

Exercise #4

The planar 3R robot ($n = 3$) is redundant for the Cartesian position task ($m = 2$). When the joint limits are not regarded as hard constraints, the solution to the stated problem is

$$
\dot{\boldsymbol{q}} = \boldsymbol{J}^{\#}(\boldsymbol{q}) \dot{\boldsymbol{r}} - \left(\boldsymbol{I} - \boldsymbol{J}^{\#}(\boldsymbol{q}) \boldsymbol{J}(\boldsymbol{q}) \right) \nabla_q H_{range}(\boldsymbol{q}),
$$

where the task velocity is

$$
\boldsymbol{r} = \begin{pmatrix} p_x \\ p_y \end{pmatrix} \qquad \Rightarrow \qquad \dot{\boldsymbol{r}} = \begin{pmatrix} v_x \\ v_y \end{pmatrix} = \begin{pmatrix} -3 \\ 0 \end{pmatrix},
$$

and the associated Jacobian, evaluated at $\widehat{\boldsymbol{q}} = (2\pi/5, \pi/2, -\pi/4)$, is given by

$$
\boldsymbol{J}(\boldsymbol{q}) = \begin{pmatrix} -\left(s_1 + s_{12} + s_{123}\right) & -\left(s_{12} + s_{123}\right) & -s_{123} \\ c_1 + c_{12} + c_{123} & c_{12} + c_{123} & c_{123} \end{pmatrix} \Rightarrow \boldsymbol{J} = \begin{pmatrix} -2.1511 & -1.2000 & -0.8910 \\ -1.0960 & -1.4050 & -0.4540 \end{pmatrix}.
$$

For each joint i, we have a range $[q_{m,i}, q_{M,i}]$ and a midrange $\bar{q}_i = \left(q_{M,i} + q_{m,i}\right)/2$. As a result, the objective function to be minimized is

$$
H_{range}(\boldsymbol{q}) = \frac{1}{2n} \sum_{i=1}^{n} \frac{\left(q_i - \bar{q}_i\right)^2}{\left(q_{M,i} - q_{m,i}\right)^2} = \frac{1}{6} \left(\frac{q_1^2}{\pi^2} + \frac{\left(q_2 - (\pi/3)\right)^2}{\left(2\pi/3\right)^2} + \frac{q_3^2}{\left(\pi/2\right)^2} \right).
$$

Its gradient evaluated at $\widehat{\boldsymbol{q}} = (2\pi/5, \pi/2, -\pi/4)$ is

$$\nabla_q H_{range}(\boldsymbol{q}) = \frac{1}{3}\begin{pmatrix} q_1/\pi^2 \\ (q_2 - \pi/3)/\left(2\pi/3\right)^2 \\ q_3/\left(\pi/2\right)^2 \end{pmatrix} \quad \Rightarrow \quad \nabla_q H_{range} = \begin{pmatrix} 0.0424 \\ 0.0398 \\ -0.1061 \end{pmatrix}.$$

As a result, the two terms of the solution are separately evaluated as

$$\dot{\boldsymbol{q}}_r = \boldsymbol{J}^{\#}\dot{\boldsymbol{r}} = \begin{pmatrix} 2.1076 \\ -1.9261 \\ 0.8730 \end{pmatrix}, \qquad \dot{\boldsymbol{q}}_n = -\left(\boldsymbol{I} - \boldsymbol{J}^{\#}\boldsymbol{J}\right)\nabla_q H_{range} = \begin{pmatrix} -0.0437 \\ 0 \\ 0.1056 \end{pmatrix},$$

yielding thus

$$\dot{\boldsymbol{q}} = \dot{\boldsymbol{q}}_r + \dot{\boldsymbol{q}}_n = \begin{pmatrix} 2.0638 \\ -1.9261 \\ 0.9786 \end{pmatrix}. \tag{4}$$

The first component of the solution exceeds the (positive) velocity bound. This is true as well for the minimum norm solution $\dot{\boldsymbol{q}}_r$; the first component of the null space term $\dot{\boldsymbol{q}}_n$, being negative, mildens the situation but is not sufficient to recover feasibility. Therefore, the largest scaling factor $k < 1$ of the task velocity $\dot{\boldsymbol{r}}$ that allows to obtain a feasible solution w.r.t. the joint velocity bounds (uniformly equal to $\dot{q}_{max} = 2$ [rad/s] for all joints) is computed as follows:

$$\dot{\boldsymbol{r}} \to k\,\dot{\boldsymbol{r}} \;\Rightarrow\; \dot{\boldsymbol{q}} \to k\,\dot{\boldsymbol{q}}_r + \dot{\boldsymbol{q}}_n \;\Rightarrow\; k\,\dot{q}_{r,1} + \dot{q}_{n,1} \overset{\downarrow}{=} \dot{q}_{max} \;\Rightarrow\; k^* = \frac{\dot{q}_{max} - \dot{q}_{n,1}}{\dot{q}_{r,1}} = \frac{2 + 0.0437}{2.1076} = 0.9697.$$

Therefore, the scaled task velocity and the scaled joint velocity that recovers feasibility are

$$\dot{\boldsymbol{r}}_s = k^*\dot{\boldsymbol{r}} = \begin{pmatrix} -2.9091 \\ 0 \end{pmatrix} \quad \Rightarrow \quad \dot{\boldsymbol{q}}_s = k^*\dot{\boldsymbol{q}}_r + \dot{\boldsymbol{q}}_n = \begin{pmatrix} 2 \\ -1.8678 \\ 0.9521 \end{pmatrix} \quad \Rightarrow \quad \boldsymbol{J}\dot{\boldsymbol{q}}_s = \begin{pmatrix} -2.9091 \\ 0 \end{pmatrix}. \tag{5}$$

It should be noted that, in this particular case, we could have chosen a larger step $\alpha > 1$ (rather than $\alpha = 1$) along the negative gradient direction of H_{range} within the term $\dot{\boldsymbol{q}}_n$, thus recovering feasibility of the solution without the need of task scaling. On the other hand, a direct application of the SNS method to recover feasibility would not be correct, since the solution $\dot{\boldsymbol{q}}$ in (4) contains also a null-space term that does not scale with the task velocity $\dot{\boldsymbol{r}}$.

Exercise #5

The planar PR robot ($n = 2$) is redundant with respect to a task of dimension $m = 1$. For the specified (scalar) task, we have

$$r = y = q_1 + l_2 s_2 \qquad \Rightarrow \qquad \dot{r} = \dot{y} = \begin{pmatrix} 1 & -l_2 c_2 \end{pmatrix}\begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \end{pmatrix} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}},$$

with the Jacobian being always full rank. The closed-form solutions to the two problems of dynamic redundancy optimization are obtained from the general LQ formulation as

$$\boldsymbol{\tau}_A = \left(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{M}^{-1}(\boldsymbol{q})\right)^{\#}\left(\ddot{\boldsymbol{r}} - \dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}} + \boldsymbol{J}(\boldsymbol{q})\boldsymbol{M}^{-1}(\boldsymbol{q})\left(\boldsymbol{c}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q})\right)\right)$$

and

$$\boldsymbol{\tau}_B = \boldsymbol{M}(\boldsymbol{q})\boldsymbol{J}^{\#}(\boldsymbol{q})\left(\ddot{\boldsymbol{r}} - \dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}} + \boldsymbol{J}(\boldsymbol{q})\boldsymbol{M}^{-1}(\boldsymbol{q})\left(\boldsymbol{c}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q})\right)\right).$$

Since the robot is at rest, the velocity terms $\boldsymbol{c}$ and $\dot{\boldsymbol{J}}\dot{\boldsymbol{q}}$ are zero. Evaluating the inertia matrix and the task Jacobian in the configuration $\bar{\boldsymbol{q}} = \begin{pmatrix} 1 & \pi/2 \end{pmatrix}^T$,

$$\boldsymbol{M}(\bar{\boldsymbol{q}}) = \begin{pmatrix} a & 0 \\ 0 & c \end{pmatrix}, \qquad \boldsymbol{J}(\bar{\boldsymbol{q}}) = \begin{pmatrix} 1 & 0 \end{pmatrix},$$

we compute

$$\begin{aligned} \boldsymbol{\tau}_A &= \left(\begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} 1/a & 0 \\ 0 & 1/c \end{pmatrix} \right)^{\#} \left(\ddot{y}_d + \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} 1/a & 0 \\ 0 & 1/c \end{pmatrix} \boldsymbol{g}(\boldsymbol{q}) \right) \\ &= \begin{pmatrix} 1/a & 0 \end{pmatrix}^{\#} \left(\ddot{y}_d + \begin{pmatrix} 1/a & 0 \end{pmatrix} \boldsymbol{g}(\boldsymbol{q}) \right) = \begin{pmatrix} a \\ 0 \end{pmatrix} \left(\ddot{y}_d + (1/a)\, g_1(\bar{\boldsymbol{q}}) \right) = \begin{pmatrix} a\, \ddot{y}_d + g_1(\bar{\boldsymbol{q}}) \\ 0 \end{pmatrix}. \end{aligned} \tag{6}$$

Similarly,

$$\begin{aligned} \boldsymbol{\tau}_B &= \begin{pmatrix} a & 0 \\ 0 & c \end{pmatrix} \begin{pmatrix} 1 & 0 \end{pmatrix}^{\#} \left(\ddot{y}_d + \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} 1/a & 0 \\ 0 & 1/c \end{pmatrix} \boldsymbol{g}(\boldsymbol{q}) \right) \\ &= \begin{pmatrix} a \\ 0 \end{pmatrix} \left(\ddot{y}_d + (1/a)\, g_1(\bar{\boldsymbol{q}}) \right) = \begin{pmatrix} a\, \ddot{y}_d + g_1(\bar{\boldsymbol{q}}) \\ 0 \end{pmatrix} = \boldsymbol{\tau}_A. \end{aligned} \tag{7}$$

As a result, the two solutions (6) and (7) are identical in this very particular case (in fact, it is here $\left(\boldsymbol{J}\boldsymbol{M}^{-1}\right)^{\#} = \boldsymbol{M}\boldsymbol{J}^{\#}$, an identity which is not true in general). Note that there is no need to derive the expression of the model term $\boldsymbol{g}(\boldsymbol{q})$ for this comparison.

A final remark is in order. The torque commands $\boldsymbol{\tau}_A$ and $\boldsymbol{\tau}_B$, which have been obtained above from the general solution of the associated constrained minimization problems, could have been found in this specific case by inspection. In the configuration $\bar{\boldsymbol{q}}$, the PR robot is fully stretched along the vertical y-axis. In addition, being the robot at rest, any torque applied at the second joint would give *no* contribution to the desired task acceleration $\ddot{y}_d$. Since we pursue in both cases a (weighted) minimum torque norm solution, the second joint torque τ_2 should simply be zero; the entire task (task acceleration $\ddot{y}_d$ in the vertical direction plus gravity compensation) is executed in a unique way by the first joint only.

Exercise #6

The gravity term of the PR robot in Fig. 2 is obtained as the gradient of the sum of the potential energies of each link

$$U_i(\boldsymbol{q}) = -m_i\, \boldsymbol{g}^T \boldsymbol{r}_{0,ci} = -m_i \begin{pmatrix} 0 & -g_0 & 0 \end{pmatrix} \boldsymbol{r}_{0,ci} = m_i g_0\, r_{0,ci_y}, \quad i = 1,2.$$

Thus (neglecting an arbitrary constant), we have

$$U(\boldsymbol{q}) = U_1(q_1) + U_2(q_1, q_2) = m_1 g_0\, q_1 + m_2 g_0\, (q_1 + d_{c2} s_2)$$

that gives

$$\boldsymbol{g}(\boldsymbol{q}) = \left(\frac{\partial U(\boldsymbol{q})}{\partial \boldsymbol{q}} \right)^T = \begin{pmatrix} (m_1 + m_2)\, g_0 \\ m_2 g_0 d_{c2} c_2 \end{pmatrix}.$$

The gradient of $\boldsymbol{g}(\boldsymbol{q})$ w.r.t. $\boldsymbol{q}$ is the symmetric (here, negative semi-definite) Hessian matrix

$$\frac{\partial \boldsymbol{g}(\boldsymbol{q})}{\partial \boldsymbol{q}} = \frac{\partial^2 U(\boldsymbol{q})}{\partial \boldsymbol{q}^2} = \begin{pmatrix} 0 & 0 \\ 0 & -m_2 g_0 d_{c2} s_2 \end{pmatrix}.$$

Its norm (associated to the standard Euclidean norm of vectors) is given by

$$\left\| \frac{\partial \boldsymbol{g}(\boldsymbol{q})}{\partial \boldsymbol{q}} \right\| = \sqrt{\lambda_{max} \left\{ \frac{\partial \boldsymbol{g}(\boldsymbol{q})}{\partial \boldsymbol{q}} \left(\frac{\partial \boldsymbol{g}(\boldsymbol{q})}{\partial \boldsymbol{q}} \right)^T \right\}} = \sqrt{\lambda_{max} \left\{ \begin{pmatrix} 0 & 0 \\ 0 & m_2^2 g_0^2 d_{c2}^2 s_2^2 \end{pmatrix} \right\}} = m_2 g_0 d_{c2} \left| s_2 \right| .$$

Thus, an upper bound for this norm is

$$\left\| \frac{\partial \boldsymbol{g}(\boldsymbol{q})}{\partial \boldsymbol{q}} \right\| \leq \alpha = m_2 g_0 d_{c2}, \qquad \forall \boldsymbol{q}. \tag{8}$$

This upper bound is tight, being attained at $q_2 = \pm\pi/2$.

* * * * *