Robotics 1
January 13, 2025
[students with midterm]

Exercise 1

For the PRR planar robot in Fig. 1, consider first the task vector $\boldsymbol{r}$ made by the position $\boldsymbol{p} \in \mathbb{R}^2$ of the end-effector and by its angle $\phi \in \mathbb{R}$ with respect to the $\boldsymbol{x}_w$ axis. Compute the corresponding Jacobian $\boldsymbol{J_r}(\boldsymbol{q})$ and find all its singularities. With the robot in a generic singular configuration $\boldsymbol{q}_s$:

a) provide the expression of all joint velocities $\dot{\boldsymbol{q}}$ that produce no task velocity $\dot{\boldsymbol{r}}$;

b) determine all task velocities $\dot{\boldsymbol{r}}$ that cannot be instantaneously realized.

Next, consider only the two-dimensional task vector $\boldsymbol{r} = \boldsymbol{p}$ for the same robot and find all singularities of the corresponding Jacobian $\boldsymbol{J_p}(\boldsymbol{q})$. When the robot is in a configuration $\boldsymbol{q}_s$ with all strictly positive joint values and such that the matrix $\boldsymbol{J_p}(\boldsymbol{q}_s)$ loses rank:

c) provide the expression of all forces $\boldsymbol{f} \in \mathbb{R}^2$ applied to the end-effector that need no joint force/torque $\boldsymbol{\tau} \in \mathbb{R}^3$ to be balanced;

d) determine the $\boldsymbol{\tau}$ that statically balances a force $\boldsymbol{f} = \begin{pmatrix} 3 & 1 \end{pmatrix}^T$ [N] applied to the end-effector.

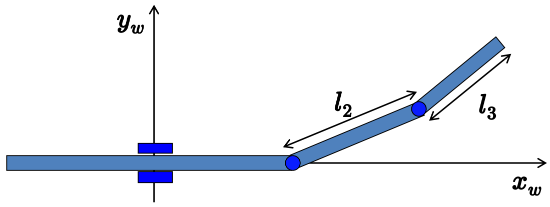

Figure 1: A PRR planar robot with the last two links of length l_2 and $l_3 \neq l_2$.

Exercise 2

A cylindrical robot has the direct kinematics of its end-effector position expressed by

$$\boldsymbol{p}(\boldsymbol{q}) = \begin{pmatrix} q_3 \cos q_2 \\ q_3 \sin q_2 \\ q_1 \end{pmatrix}.$$

When the desired position is $\boldsymbol{p}_d = \begin{pmatrix} 1 & -1 & 3 \end{pmatrix}^T$ [m], provide the first few iterations of a Newton algorithm for the numerical solution of the inverse kinematics problem in the following two cases:

a) starting from the initial guess $\boldsymbol{q}_a^{[0]} = \begin{pmatrix} -2 & 0.7\pi & \sqrt{2} \end{pmatrix}^T$ [m,rad,m];

b) starting from the initial guess $\boldsymbol{q}_b^{[0]} = \begin{pmatrix} 2 & \pi/4 & \sqrt{2} \end{pmatrix}^T$ [m,rad,m].

In case of convergence, the algorithm should stop as soon as $\left\|\boldsymbol{e}^{[k]}\right\| = \left\|\boldsymbol{p}_d - \boldsymbol{p}(\boldsymbol{q}^{[k]})\right\| \leq \epsilon = 0.1$ mm.

Exercise 3

A 2R planar robot with link lengths $l_1 = 1.2$, $l_2 = 0.8$ [m] is at rest at $t = 0$ in the configuration $\boldsymbol{q}_0 = \boldsymbol{0}$ (stretched along the $\boldsymbol{x}_0$ axis). A pointwise target moves at constant speed $v = 1.5$ m/s on a straight line with an angle $\delta = 15°$ from the $\boldsymbol{x}_0$ axis, being in $\boldsymbol{p}_0 = \begin{pmatrix} -2 & 1 \end{pmatrix}^T$ [m] at $t = 0$ and entering after in the robot workspace. Solve the following rendez-vous problem:

a) define a trajectory that will bring the robot end-effector on the target when the latter crosses the $\boldsymbol{y}_0$ axis; the end-effector should have then the same velocity $\boldsymbol{v}_t \in \mathbb{R}^2$ of the target;

b) provide the rendez-vous time $t_{rv} > 0$ and the expression of the command $\dot{\boldsymbol{q}}(t) \in \mathbb{R}^2$, $t \in [0, t_{rv}]$.

How would you modify the velocity command $\dot{\boldsymbol{q}}(t)$ as a function of $\boldsymbol{q}(t)$ so as to reach the target at the rendez-vous position if the robot starts from a configuration close but different from $\boldsymbol{q}_0$?

[180 minutes (3 hours); open books]

Solution

January 13, 2025
[students with midterm]

Exercise 1

The kinematics of the three-dimensional task vector $\boldsymbol{r} = (\boldsymbol{p}, \phi)$ for the PRR robot of Fig. 1 is

$$\boldsymbol{r} = \begin{pmatrix} q_1 + l_2 \cos q_2 + l_3 \cos(q_2 + q_3) \\ l_2 \sin q_2 + l_3 \sin(q_2 + q_3) \\ q_2 + q_3 \end{pmatrix} = \boldsymbol{f}(\boldsymbol{q}),$$

with the corresponding (analytic) Jacobian

$$\boldsymbol{J_r}(\boldsymbol{q}) = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{q}} = \begin{pmatrix} 1 & -l_2 \sin q_2 - l_3 \sin(q_2 + q_3) & -l_3 \sin(q_2 + q_3) \\ 0 & l_2 \cos q_2 + l_3 \cos(q_2 + q_3) & l_3 \cos(q_2 + q_3) \\ 0 & 1 & 1 \end{pmatrix}.$$

Since $\det \boldsymbol{J_r}(\boldsymbol{q}) = l_2 \cos q_2$, singular configurations are obtained when $q_2 = \pm \pi/2$.

For $q_2 = \pi/2$, we have

$$\bar{\boldsymbol{J}} = \boldsymbol{J_r}|_{q_2=\pi/2} = \begin{pmatrix} 1 & -l_2 - l_3 \cos q_3 & -l_3 \cos q_3 \\ 0 & -l_3 \sin q_3 & -l_3 \sin q_3 \\ 0 & 1 & 1 \end{pmatrix}$$

with the third column being the sum of the first column multiplied by l_2 and the second column. Therefore, all joint velocities that produce $\dot{\boldsymbol{r}} = \boldsymbol{0}$ are of the form

$$\dot{\boldsymbol{q}} = \nu \begin{pmatrix} l_2 \\ 1 \\ -1 \end{pmatrix} \in \mathcal{N}\{\bar{\boldsymbol{J}}\}, \qquad \text{for any } \nu \in \mathbb{R}.$$

On the other hand, task velocities $\dot{\boldsymbol{r}}$ that cannot be instantaneously realized lie outside the range of $\bar{\boldsymbol{J}}$. Taking then as basis for $\mathcal{R}\{\bar{\boldsymbol{J}}\}$, e.g., its first two columns, one should find vectors $\dot{\boldsymbol{r}} \in \mathbb{R}^3$ that are linear independent from this basis, i.e., such that

$$\det \begin{pmatrix} 1 & -l_2 - l_3 \cos q_3 & \dot{r}_1 \\ 0 & -l_3 \sin q_3 & \dot{r}_2 \\ 0 & 1 & \dot{r}_3 \end{pmatrix} \neq 0 \qquad \Rightarrow \qquad \dot{r}_3\, l_3 \sin q_3 + \dot{r}_2 \neq 0 \;\; \text{(and no condition on } \dot{r}_1\text{)}.$$

There are many instances of vectors that satisfy the above inequality, e.g., $\dot{\boldsymbol{r}} = (0, \lambda, 0)$ for any $\lambda \neq 0$, and thus such that $\dot{\boldsymbol{r}} \notin \mathcal{R}\{\bar{\boldsymbol{J}}\}$. However, the most elegant way to supplement the range space of $\bar{\boldsymbol{J}}$ is to build the null space of $\bar{\boldsymbol{J}}^T$, since $\mathcal{R}\{\bar{\boldsymbol{J}}\} \oplus \mathcal{N}\{\bar{\boldsymbol{J}}^T\} = \mathbb{R}^3$. Being

$$\dot{\boldsymbol{r}}_0 = \mu \begin{pmatrix} 0 \\ 1 \\ l_3 \sin q_3 \end{pmatrix} \in \mathcal{N}\{\bar{\boldsymbol{J}}^T\}, \qquad \text{for any } \mu \neq 0,$$

we can conclude that any task velocity $\dot{\boldsymbol{r}}$ that has a component along $\dot{\boldsymbol{r}}_0$ cannot be realized. Conversely, any $\dot{\boldsymbol{r}}$ that has no component along $\dot{\boldsymbol{r}}_0$ belongs to $\mathcal{R}\{\bar{\boldsymbol{J}}\}$ and thus can be realized.

For the other type of singular configurations having $q_2 = -\pi/2$, it is

$$\bar{\bar{\boldsymbol{J}}} = \boldsymbol{J_r}|_{q_2=-\pi/2} = \begin{pmatrix} 1 & l_2 + l_3\cos q_3 & l_3\cos q_3 \\ 0 & l_3\sin q_3 & l_3\sin q_3 \\ 0 & 1 & 1 \end{pmatrix}.$$

Following the same procedure as above, we obtain

$$\dot{\boldsymbol{q}} = \nu \begin{pmatrix} l_2 \\ -1 \\ 1 \end{pmatrix} \in \mathcal{N}\{\bar{\bar{\boldsymbol{J}}}\}, \qquad \text{for any } \nu \in \mathbb{R}$$

and

$$\dot{\boldsymbol{r}}_0 = \mu \begin{pmatrix} 0 \\ -1 \\ l_3\sin q_3 \end{pmatrix} \in \mathcal{N}\{\bar{\bar{\boldsymbol{J}}}^T\}, \qquad \text{for any } \mu \neq 0.$$

As for the two-dimensional task vector $\boldsymbol{r} = \boldsymbol{p}$, the corresponding Jacobian reduces to the 2×3 matrix

$$\boldsymbol{J_p}(\boldsymbol{q}) = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{q}} = \begin{pmatrix} 1 & -l_2\sin q_2 - l_3\sin(q_2+q_3) & -l_3\sin(q_2+q_3) \\ 0 & l_2\cos q_2 + l_3\cos(q_2+q_3) & l_3\cos(q_2+q_3) \end{pmatrix},$$

obtained by simply eliminating the third row from $\boldsymbol{J_r}(\boldsymbol{q})$. The singular configurations are found by zeroing the three minors obtained by deleting one column at a time from $\boldsymbol{J_p}(\boldsymbol{q})$ (resulting in three 2×2 matrices $\boldsymbol{J}_{-i}(\boldsymbol{q})$, $i = 1,2,3$). One has

$$\det \boldsymbol{J}_{-1} = l_2 l_3 s_3 = 0 \qquad \det \boldsymbol{J}_{-2} = l_3 c_{23} = 0 \qquad \det \boldsymbol{J}_{-3} = l_2 c_2 + l_3 c_{23} = 0,$$

with the shorthand notation used for trigonometric functions (e.g., $c_{23} = \cos(q_2+q_3)$). From these, the singular configurations are characterized by

$$c_2 = 0 \cap s_3 = 0 \qquad \Rightarrow \qquad (q_2, q_3) = \{(\pi/2, 0), (\pi/2, \pi), (-\pi/2, 0), (-\pi/2, \pi)\}.$$

As it is always the case, the first variable q_1 never matters in the definition of kinematic singularities.

Since it is asked to use the Jacobian only in a singular configuration $\boldsymbol{q_s}$ characterized by all strictly positive joint values, we choose $\boldsymbol{q_s} = (q_1, \pi/2, \pi)$, with a generic value $q_1 > 0$. The matrix

$$\boldsymbol{J}_s^T = \boldsymbol{J}_{\boldsymbol{p}}^T(\boldsymbol{q}_s) = \begin{pmatrix} 1 & 0 \\ l_3 - l_2 & 0 \\ l_3 & 0 \end{pmatrix}$$

has clearly rank 1. All Cartesian forces $\boldsymbol{f} \in \mathbb{R}^2$ applied to the end-effector that need no joint force/torque $\boldsymbol{\tau} \in \mathbb{R}^3$ to be balanced have the simple form

$$\boldsymbol{f} = \varphi \begin{pmatrix} 0 \\ 1 \end{pmatrix} \in \mathcal{N}\{\boldsymbol{J}_s^T\}, \qquad \text{for any } \varphi \in \mathbb{R}.$$

Finally, to balance the force $\boldsymbol{f} = (3, 1)$ [N] applied to the robot end-effector so that the robot remains in static conditions, the motors need to provide the vector of force (at the first joint) and torques (at the second and third joint)

$$\boldsymbol{\tau} = -\boldsymbol{J}_s^T \boldsymbol{f} = \begin{pmatrix} -3 \\ 3(l_2 - l_3) \\ -3l_3 \end{pmatrix} \text{ [N,Nm,Nm]},$$

where the minus sign in the general formula stands for the need of having zero balance of generalized forces at all joints. The resulting vector $\boldsymbol{\tau}$ would be the same for any applied Cartesian force of the form $\boldsymbol{f} = (3, \varphi)$ [N], with arbitrary φ.

Exercise 2

The Jacobian associated to the positional direct kinematics of the cylindrical robot is

$$\boldsymbol{J}(\boldsymbol{q}) = \frac{\partial \boldsymbol{p}}{\partial \boldsymbol{q}} = \begin{pmatrix} 0 & -q_3 \sin q_2 & \cos q_2 \\ 0 & q_3 \cos q_2 & \sin q_2 \\ 1 & 0 & 0 \end{pmatrix}$$

This matrix is singular when $\det \boldsymbol{J} = -q_3 = 0$. It is simple to write a code that performs a number of times the Newton iteration

$$\boldsymbol{q}^{[k]} = \boldsymbol{q}^{[k-1]} + \boldsymbol{J}^{-1}(\boldsymbol{q}^{[k-1]}) \left(\boldsymbol{p}_d - \boldsymbol{p}(\boldsymbol{q}^{[k-1]})\right)$$

for solving the inverse kinematics problem, starting from a given initial guess $\boldsymbol{q}^{[0]}$.

For the problem data $\boldsymbol{p}_d = (1, -1, 3)$ [m], Tab. 1 summarizes the obtained numerical results when starting from $\boldsymbol{q}_a^{[0]} = (-2, 0.7\pi, \sqrt{2})$ [m,rad,m]). The algorithm stops at the third iteration, having reached the requested accuracy of $\epsilon = 10^{-4}$ m on the norm of the Cartesian position error. The inverse kinematics solution found is $\boldsymbol{q}_{\text{sol},1} = (3, 2.3562, -1.4142)$ [m,rad,m].

iteration k	$\boldsymbol{q}^{[k]}$	$\boldsymbol{p}^{[k]}$	$\lVert \boldsymbol{e}^{[k]} \rVert$	$\det \boldsymbol{J}(\boldsymbol{q}^{[k]})$
0	$(-2, 2.1991, 1.4142)$	$(-0.8312, 1.1441, 2)$	5.74027	-1.4142
1	$(3, 2.0426, -1.3968)$	$(0.6349, 1.2441, 3)$	0.43918	1.3968
2	$(3, 2.3549, -1.3452)$	$(0.9500, -0.9524, 3)$	0.06895	1.3452
3	$(3, 2.3562, -1.4142)$	$(1.0000, -0.9999, 3)$	0.00009	1.4142

Table 1: Steps of the Newton method for the inverse kinematics problem when starting from $\boldsymbol{q}_a^{[0]}$.

When starting from $\boldsymbol{q}_b^{[0]} = \left(2, \pi/4, \sqrt{2}\right)$ [m,rad,m], where $\boldsymbol{p}^{[0]} = (1, 1, 2)$ [m] and $\lVert \boldsymbol{e}^{[0]} \rVert = 2.2361$, the next configuration computed by Newton's method is

$$\boldsymbol{q}^{[1]} = \begin{pmatrix} 3 \\ -0.2146 \\ 0 \end{pmatrix}, \qquad \text{with } \boldsymbol{p}^{[1]} = \begin{pmatrix} 0 \\ 0 \\ 3 \end{pmatrix}, \ \lVert \boldsymbol{e}^{[1]} \rVert = 1.4142, \ \text{ and } \det \boldsymbol{J}(\boldsymbol{q}^{[1]}) = 0.$$

Having reached a singular configuration, the algorithm should be stopped at this first iteration to prevent divergence (`NaN` in MATLAB).

Note that if one replaces the (numerical) pseudoinverse of the Jacobian to its inverse, i.e.,

$$\boldsymbol{q}^{[k]} = \boldsymbol{q}^{[k-1]} + \boldsymbol{J}^{\#}(\boldsymbol{q}^{[k-1]}) \left(\boldsymbol{p}_d - \boldsymbol{p}(\boldsymbol{q}^{[k-1]})\right),$$

the modified algorithm will tolerate the singularity after the first iteration, and eventually converge in six iterations to the other inverse kinematics solution $\boldsymbol{q}_{\text{sol},2} = (3, -0.7854, 1.4142)$ [m,rad,m], as shown in Tab. 2.

iteration k	$\boldsymbol{q}^{[k]}$	$\boldsymbol{p}^{[k]}$	$\left\lVert \boldsymbol{e}^{[k]} \right\rVert$	$\det \boldsymbol{J}(\boldsymbol{q}^{[k]})$
0	$(2, 0.7854, 1.4142)$	$(1, 1, 2)$	2.23607	-1.4142
1	$(3, -0.2146, 0)$	$(0, 0, 3)$	1.41421	0
2	$(3, -0.2146, 1.1900)$	$(1.1627, -0.2534, 3)$	0.76410	-1.1900
3	$(3, -0.8567, 1.1900)$	$(0.7794, -0.8993, 3)$	0.24252	-1.1900
4	$(3, -0.7720, 1.4106)$	$(1.0107, -0.9840, 3)$	0.01921	-1.4106
5	$(3, -0.7854, 1.4141)$	$(0.9999, -0.9999, 3)$	0.00013	-1.4141
6	$(3, -0.7854, 1.4142)$	$(1, -1, 3)$	0.00000	-1.4142

Table 2: Steps of the modified Newton method (with pseudoinverse) for the inverse kinematics problem when starting from $\boldsymbol{q}_b^{[0]}$.

Exercise 3

The trajectory of the target is

$$\boldsymbol{p}_T(t) = \boldsymbol{p}_0 + v\,t \begin{pmatrix} \cos\delta \\ \sin\delta \end{pmatrix}, \qquad t \in [0.t_{rv}],$$

and the time instant of rendez-vous (appointment), i.e., when the target crosses the $\boldsymbol{y}_0$ axis, is obtained from

$$p_{Tx}(t_{rv}) = p_{0x} + v\,t_{rv}\cos\delta = 0 \qquad \Rightarrow \qquad t_{rv} = \frac{-p_{0x}}{v\cos\delta} = 1.380\,\text{s}.$$

Accordingly, the y-component of the Cartesian rendez-vous position is given by

$$p_{Ty} = p_{0y} + v\,t_{rv}\sin\delta = 1.5359 \qquad \Rightarrow \qquad \boldsymbol{p}_{rv} = \begin{pmatrix} 0 \\ 1.5359 \end{pmatrix} \text{ [m]}.$$

For rendez-vous position $\boldsymbol{p}_{rv}$, the two (regular) inverse kinematics solutions for the 2R robot are computed from the usual formulas as

$$\boldsymbol{q}_a = \begin{pmatrix} 1.0294 \\ 1.4250 \end{pmatrix} \text{ [rad]} \quad \text{(right arm)} \qquad \boldsymbol{q}_b = \begin{pmatrix} 2.1122 \\ -1.4250 \end{pmatrix} \text{ [rad]} \quad \text{(left arm)}.$$

The velocity of the target is constant and equal to

$$\dot{\boldsymbol{p}}_T = v \begin{pmatrix} \cos\delta \\ \sin\delta \end{pmatrix} = \begin{pmatrix} 1.4489 \\ 0.3882 \end{pmatrix} \text{ [m/s]}.$$

Therefore, the robot joint velocity for matching the target velocity at the rendez-vous is given by

$$\dot{\boldsymbol{q}}_a = \boldsymbol{J}^{-1}(\boldsymbol{q}_a)\dot{\boldsymbol{p}}_T = \begin{pmatrix} -0.7359 \\ -0.6278 \end{pmatrix} \text{ [rad/s]} \quad \text{(right arm)}$$

$$\dot{\boldsymbol{q}}_b = \boldsymbol{J}^{-1}(\boldsymbol{q}_b)\dot{\boldsymbol{p}}_T = \begin{pmatrix} -1.1508 \\ 0.6278 \end{pmatrix} \text{ [rad/s[} \quad \text{(left arm)}.$$

A suitable trajectory for solving the problem is computed in the joint space,[1] as cubic polynomials interpolating the following alternative boundary conditions

$$\boldsymbol{q}(0)=\mathbf{0} \quad \dot{\boldsymbol{q}}(0)=\mathbf{0} \qquad \Rightarrow \qquad \begin{array}{lll} \boldsymbol{q}(t_{rv})=\boldsymbol{q}_a & \dot{\boldsymbol{q}}(t_{rv})=\dot{\boldsymbol{q}}_a & \text{(right arm)} \\ \boldsymbol{q}(t_{rv})=\boldsymbol{q}_b & \dot{\boldsymbol{q}}(t_{rv})=\dot{\boldsymbol{q}}_b & \text{(left arm).} \end{array}$$

The interpolating cubics are conveniently written in normalized time as

$$\boldsymbol{q}_d(\tau)=\left(3\boldsymbol{q}_{rv}-t_{rv}\dot{\boldsymbol{q}}_{rv}\right)\tau^2+\left(-2\boldsymbol{q}_{rv}+t_{rv}\dot{\boldsymbol{q}}_{rv}\right)\tau^3, \qquad \tau=\frac{t}{t_{rv}}\in[0,1],$$

which applies to both cases, just replacing a or b for rv in $\boldsymbol{q}_{rv}$ and $\dot{\boldsymbol{q}}_{rv}$. The corresponding joint velocity command is

$$\dot{\boldsymbol{q}}_d(\tau)=\frac{2}{t_{rv}}\left(3\boldsymbol{q}_{rv}-t_{rv}\dot{\boldsymbol{q}}_{rv}\right)\tau+\frac{3}{t_{rv}}\left(-2\boldsymbol{q}_{rv}+t_{rv}\dot{\boldsymbol{q}}_{rv}\right)\tau^2, \qquad \tau=\frac{t}{t_{rv}}\in[0,1].$$

Substituting the numerical values of the boundary conditions, we obtain the joint trajectories in normalized time

$$\boldsymbol{q}_{d.a}(\tau)=\begin{pmatrix}4.1040\\5.1415\end{pmatrix}\tau^2+\begin{pmatrix}-3.0746\\-3.7165\end{pmatrix}\tau^3, \qquad \tau=\frac{t}{t_{rv}}\in[0,1] \quad \text{(right arm)}$$

or

$$\boldsymbol{q}_{d,b}(\tau)=\begin{pmatrix}7.9251\\-5.1415\end{pmatrix}\tau^2+\begin{pmatrix}-5.8129\\3.7165\end{pmatrix}\tau^3, \qquad \tau=\frac{t}{t_{rv}}\in[0,1] \quad \text{(left arm).}$$

When using the original time t, the equivalent expressions are

$$\boldsymbol{q}_{d.a}(t)=\begin{pmatrix}2.1539\\2.6984\end{pmatrix}t^2+\begin{pmatrix}-1.1690\\-1.4130\end{pmatrix}t^3, \qquad t\in[0,t_{rv}] \quad \text{(right arm)}$$

or

$$\boldsymbol{q}_{d,b}(t)=\begin{pmatrix}4.1593\\-2.6984\end{pmatrix}t^2+\begin{pmatrix}-2.2101\\1.4130\end{pmatrix}t^3, \qquad t=\in[0,t_{rv}] \quad \text{(left arm).}$$

Accordingly, the joint velocity commands in the two cases are

$$\dot{\boldsymbol{q}}_{d.a}(\tau)=\begin{pmatrix}5.9462\\7.4495\end{pmatrix}\tau+\begin{pmatrix}-6.6821\\-8.0773\end{pmatrix}\tau^2, \qquad \tau=\frac{t}{t_{rv}}\in[0,1] \quad \text{(right arm)}$$

or

$$\dot{\boldsymbol{q}}_{d,b}(\tau)=\begin{pmatrix}11.4826\\-7.4495\end{pmatrix}\tau+\begin{pmatrix}-12.6334\\8.0773\end{pmatrix}\tau^2, \qquad \tau=\frac{t}{t_{rv}}\in[0,1] \quad \text{(left arm),}$$

and in terms of the original time

$$\dot{\boldsymbol{q}}_{d.a}(t)=\begin{pmatrix}4.3077\\5.3967\end{pmatrix}t+\begin{pmatrix}-3.5069\\-4.2391\end{pmatrix}t^2, \qquad t\in[0,t_{rv}] \quad \text{(right arm)}$$

or

$$\dot{\boldsymbol{q}}_{d,b}(t)=\begin{pmatrix}8.3185\\-5.3967\end{pmatrix}t+\begin{pmatrix}-6.6303\\4.2391\end{pmatrix}t^2, \qquad t\in[0,t_{rv}] \quad \text{(left arm).}$$

[1] In this way, we avoid any problem related to singularities in the inversion of a Cartesian trajectory. The initial position of the robot end-effector is singular, but the joint configuration is already specified.

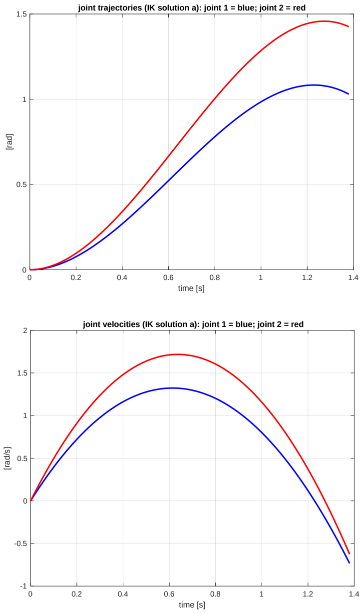

Figure 2: The planned trajectory for the 2R robot when choosing the **right** arm configuration at the rendez-vous: joint positions (top) and velocities (bottom).

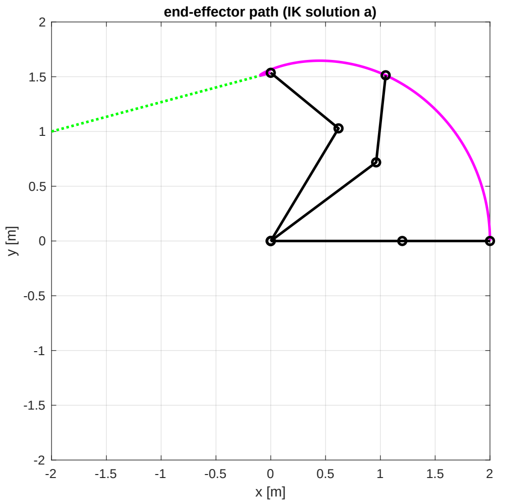

Figure 3: Cartesian path of the end-effector for the joint trajectory in Fig. 2.

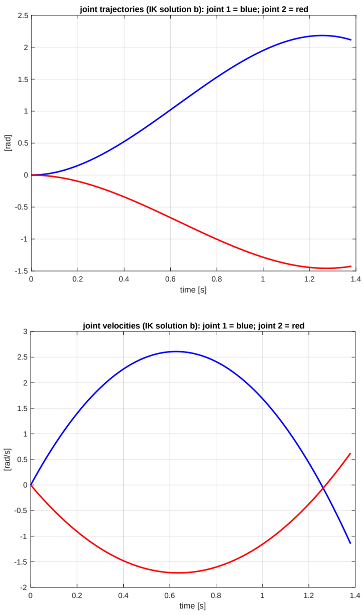

Figure 4: The planned trajectory for the 2R robot when choosing the **left** arm configuration at the rendez-vous: joint positions (top) and velocities (bottom).

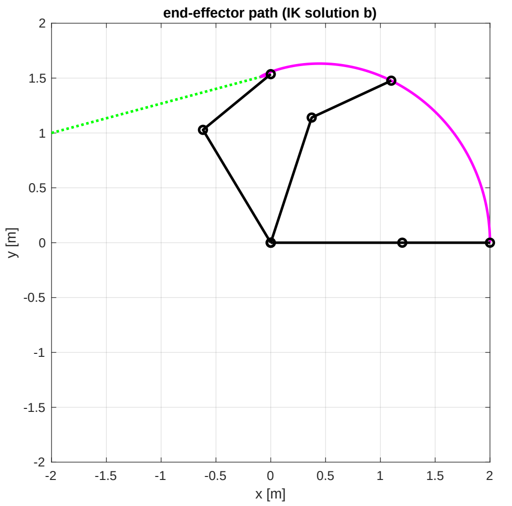

Figure 5: Cartesian path of the end-effector for the joint trajectory in Fig. 4.

Figure 2 shows the joint positions and velocity for case a (right arm configuration at rendez-vous), while the resulting path of the end-effector is given in Fig. 3. Therein, the path of the target is traced as a dotted green line. Similarly, Figs. 4–5 refer to case b (left arm configuration at rendez-vous). Note that in both cases the end-effector reaches approximately the rendez-vous position and then slightly moves back to the right to reach the exact desired position while matching also the velocity of the pointwise target.

If the robot starts from a perturbed initial configuration that is different from $\boldsymbol{q}_0$, the planned trajectory (and the corresponding joint velocity command $\dot{\boldsymbol{q}}_d(t)$, for $t \in [0, t_{rv}]$) will fail to solve the rendez-vous task. Rather than replanning a trajectory from the new initial configuration (a different one for every new initialization!), it is more convenient to include a control action with a feedback term proportional to the joint position error, i.e.,

$$\dot{\boldsymbol{q}}_c(t) = \dot{\boldsymbol{q}}_d(t) + \boldsymbol{K}_p \left(\boldsymbol{q}_d(t) - \boldsymbol{q}(t) \right),$$

with a positive definite, 2×2 (diagonal) gain matrix $\boldsymbol{K}_p$. This additional action will recover the error with respect to the originally planned joint trajectory in an exponentially fast way, depending on the choice of the control gains. For a diagonal $\boldsymbol{K}_p$, by choosing sufficiently large gains $k_{p,1}$ and $k_{p,2}$, the error will vanish in practice well before the planned rendez-vous time instant t_{rv}, thus allowing successful completion also for the perturbed task. Indeed, the larger is the initial perturbation from $\boldsymbol{q}_0$, the larger will have to be the control gains. If the required gains become too large (i.e., when the initial configuration is too far away from $\boldsymbol{q}_0$), risking a saturation of the applied command, then it would be appropriate to replan the rendez-vous trajectory $\boldsymbol{q}_d(t)$.

* * * * *