Robotics I

February 6, 2014

Exercise 1

A pan-tilt[1] camera sensor, such as the commercial webcams in Fig. 1, is mounted on the fixed base of a robot manipulator and is used for pointing at a (point-wise) target in the 3D Cartesian space. The tilt rotation is typically limited to maximum ±90° w.r.t. the vertical axis, or slightly more. The motion of the optical axis of the camera can be described with the Denavit-Hartenberg (DH) formalism, as that of a 2-dof simple robot manipulator.

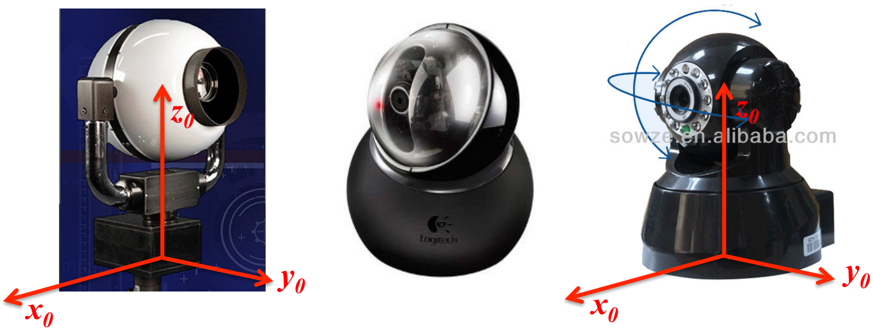

Figure 1: Pan-tilt cameras, with placement of the reference frame RF_0 (shown for two of them)

- Assign the frames according to the DH convention and provide the associated table of parameters. Use *mandatorily* the reference frame $RF_0 = \{\boldsymbol{x}_0, \boldsymbol{y}_0, \boldsymbol{z}_0\}$ as indicated in Fig. 1. One of the axes of the last frame should be coincident with the optical axis of the camera.

- Determine the unit vector of the pointing axis as a function of the joint angles $\boldsymbol{q} \in \mathbb{R}^2$.

- For a given target position ${}^0\boldsymbol{p}_T = \begin{pmatrix} {}^0x_T & {}^0y_T & {}^0z_T \end{pmatrix}^T$, with ${}^0z_T \geq 0$ (and sufficiently large), determine the value(s) of $\boldsymbol{q}$ that solve the pointing task. Is this inverse kinematics problem always solvable or well defined?

Exercise 2

In a pick-and-place task, beside the desired initial Cartesian pose $\boldsymbol{r}_0$ at time $t = t_0$ and the final pose $\boldsymbol{r}_f$ at $t = t_f$ of the robot, it is common to assign two intermediate robot poses, $\boldsymbol{r}_1$ at $t = t_1$ and $\boldsymbol{r}_2$ at $t = t_2$ (with $t_0 < t_1 < t_2 < t_f$), so as to shape the total robot motion in three phases: *Lift off* from $\boldsymbol{r}_0$ to $\boldsymbol{r}_1$; *Travel* from $\boldsymbol{r}_1$ to $\boldsymbol{r}_2$; and, *Set down* from $\boldsymbol{r}_2$ to $\boldsymbol{r}_f$. Smoothness of the trajectory is requested, with continuity up to the acceleration at any $t \in [t_0, t_f]$. The first and last phases are in 'guarded' move, and should be performed with caution because of the closeness to environmental surfaces. Therefore, we require also zero velocity and acceleration at the initial and final poses. The four Cartesian poses $\boldsymbol{r}_0$, $\boldsymbol{r}_1$, $\boldsymbol{r}_2$, and $\boldsymbol{r}_f$ have been transformed into four configurations $\boldsymbol{q}_0$, $\boldsymbol{q}_1$, $\boldsymbol{q}_2$, and $\boldsymbol{q}_f$ of a robot with revolute joints by means if its inverse kinematics.

[1] Panning refers to left/right rotations around the vertical axis, tilting to up/down rotations around an horizontal axis.

Define a 4-3-4 trajectory for a generic joint q addressing the given task. Give the expressions used for each polynomial tract, formulate the problem, and provide the value of all coefficients using:

$$t_0 = 0, \qquad t_1 = 2, \qquad t_2 = 4, \qquad t_1 = 6 \qquad [\text{s}]$$

$$q_0 = q(t_0) = 0, \qquad q_1 = q(t_1) = 10, \qquad q_2 = q(t_2) = 80, \qquad q_f = q(t_f) = 90 \qquad [\text{deg}].$$

Hint: You can solve the problem numerically (by a direct method) or analytically (by proper choosing the structure of the polynomials, so as to ease the analysis). Either way is fine. The former approach is rather straightforward, but only if you use a (simple) Matlab code (do not include this, just report the results as requested). The latter is more complex, though quite elegant (you can reduce the problem to the solution of a suitable linear system of two equations in two unknowns, which is indeed solvable in closed form).

Exercise 3

Consider a robot manipulator with n joints, its configuration vector $\boldsymbol{q} \in \mathbb{R}^n$, and a task described by $\boldsymbol{r} = \boldsymbol{r}(\boldsymbol{q})$, with $\boldsymbol{r} \in \mathbb{R}^m$. Assume that $m = n$ and a desired task trajectory $\boldsymbol{r}_d(t)$ is given. Prove that the kinematic control law

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{-1}(\boldsymbol{q})\,\dot{\boldsymbol{r}}_d + k\,\boldsymbol{J}^T(\boldsymbol{q})\,(\boldsymbol{r}_d - \boldsymbol{r}(\boldsymbol{q}))\,, \tag{1}$$

with the task Jacobian matrix $\boldsymbol{J}(\boldsymbol{q}) = \partial \boldsymbol{r}(\boldsymbol{q})/\partial \boldsymbol{q}$ and a scalar $k > 0$, will force any initial error $\boldsymbol{e}(0) = \boldsymbol{r}_d(0) - \boldsymbol{r}(\boldsymbol{q}(0))$ at time $t = 0$ to converge to zero, as long as no singularities of $\boldsymbol{J}(\boldsymbol{q})$ are encountered. Discuss the above controller in terms of off-line versus on-line computations needed.

Next, assume that $m < n$. How can the kinematic control law (1) be modified in order to guarantee the same previous features?

[210 minutes for all exercises; open books]

Solutions

February 6, 2014

Exercise 1

The DH frame assignment is shown in Fig. 2, with the associated parameters in Table 1. Note that a non-zero offset $d_1 > 0$ is present, since the reference frame cannot be assigned arbitrarily, as specified by the text of the problem (otherwise, we could move up the origin O_0 along $\boldsymbol{z}_0$ until intersecting the second joint axis).

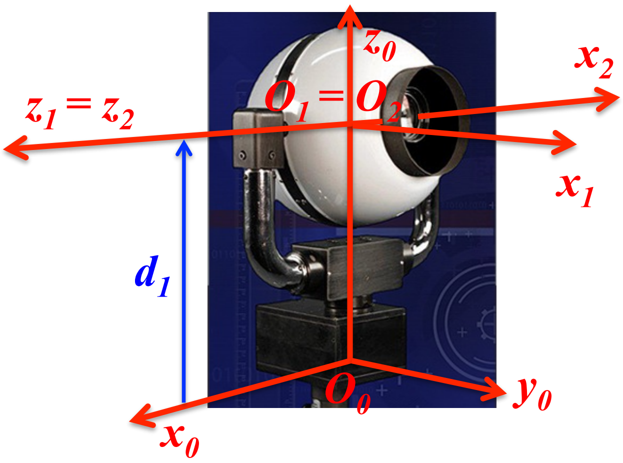

Figure 2: Assignment of Denavit-Hartenberg frames for a pan-tilt camera

i	α_i	a_1	d_i	θ_i
1	$\pi/2$	0	d_1	q_1
2	0	0	0	q_2

Table 1: Denavit-Hartenberg parameters for the pan-tilt camera

The optical axis of the camera is the unit axis ${}^0\boldsymbol{x}_2$ of last frame, whose expression is given by

$$
{}^0\boldsymbol{x}_2 = {}^0\boldsymbol{R}_1(q_1)\,{}^1\boldsymbol{R}_2(q_2)\begin{pmatrix}1\\0\\0\end{pmatrix} = \begin{pmatrix}\cos q_1 & 0 & \sin q_1\\ \sin q_1 & 0 & -\cos q_1\\ 0 & 1 & 0\end{pmatrix}\begin{pmatrix}\cos q_2 & -\sin q_2 & 0\\ \sin q_2 & \cos q_2 & 0\\ 0 & 0 & 1\end{pmatrix}\begin{pmatrix}1\\0\\0\end{pmatrix}
$$
$$
= \begin{pmatrix}\cos q_1 \cos q_2\\ \sin q_1 \cos q_2\\ \sin q_2\end{pmatrix}.
$$

For the pointing task, observe first that any 3D position ${}^0\boldsymbol{p}_T = \begin{pmatrix} {}^0x_T & {}^0y_T & {}^0z_T \end{pmatrix}^T$ of a target having ${}^0z_T \geq 0$ is mapped into a single value of two task variables, namely the pan angle α and the tilt angle β, as

$$\alpha = \text{ATAN2}\,\{{}^0y_T, {}^0x_T\}, \qquad \beta = \text{ATAN2}\,\{{}^0z_T, \sqrt{{}^0x_T^2 + {}^0y_T^2}\}. \tag{2}$$

When the target is on the (positive) $\boldsymbol{z}_0$ axis, α is not defined[2]. Otherwise, the mapping in (2) is surjective (but not injective, since all Cartesian points on a half-ray from the origin give the same values of (α, β)). If there were no base offset ($d_1 = 0$), these two angles would also coincide with the values (q_1, q_2), in the same order, solution of the inverse kinematics problem. Nevertheless, simple geometry shows that the inverse kinematics solution for the two-dimensional pointing task is given by

$$q_1 = \text{ATAN2}\,\{{}^0y_T, {}^0x_T\}, \qquad q_2 = \text{ATAN2}\,\{{}^0z_T - d_1, \sqrt{{}^0x_T^2 + {}^0y_T^2}\}, \tag{3}$$

except for the singular case ${}^0y_T = {}^0x_T = 0$, where the angle q_1 is not specified (any value of q_1 satisfies the problem). Note that there is no choice of signs for angle q_2, because of the range limitation of this joint (moreover, the target position is assumed to lie in the Cartesian region with positive values of ${}^0z_T - d_1$).

Exercise 2

A numerical procedure for obtaining the solution is detailed in steps **A**–**E** below. We may call this a *brute force* approach.

A. Define the interpolating polynomials for the three phases L = *Lift off* (degree 4), T = *Travel* (degree 3), and S = *Set down* (degree 4):

$$\begin{aligned}
q_L(t) &= a_{L0} + a_{L1}t + a_{L2}t^2 + a_{L3}t^3 + a_{L4}t^4 && t \in [t_0, t_1] \\
q_T(t) &= a_{T0} + a_{T1}t + a_{T2}t^2 + a_{T3}t^3 && t \in [t_1, t_2] \\
q_S(t) &= a_{S0} + a_{S1}t + a_{S2}t^2 + a_{S3}t^3 + a_{S4}t^4 && t \in [t_1, t_f].
\end{aligned} \tag{4}$$

The 4-3-4 trajectory will be the concatenation of these three polynomials for the motion interval $t \in [t_0, t_f]$:

$$q_{434}(t) = \begin{cases} q_L(t) & \text{for } t \in [t_0, t_1] \\ q_T(t) & \text{for } t \in [t_1, t_2] \\ q_S(t) & \text{for } t \in [t_2, t_f]. \end{cases} \tag{5}$$

B. The 14 unknown coefficients are organized in the vector

$$\boldsymbol{x}^T = \begin{pmatrix} a_{L0} & a_{L1} & a_{L2} & a_{L3} & a_{L4} & a_{T0} & a_{T1} & a_{T2} & a_{T3} & a_{S0} & a_{S1} & a_{S2} & a_{S3} & a_{S4} \end{pmatrix}. \tag{6}$$

C. Impose the requested 14 boundary conditions on the 14 coefficients:

$$\begin{aligned}
& q_L(t_0) = q_0 \qquad \dot{q}_L(t_0) = 0 \qquad \ddot{q}_L(t_0) = 0 \qquad \dot{q}_L(t_1) = q_1 \\
& \dot{q}_L(t_1) = \dot{q}_T(t_1) \qquad \ddot{q}_L(t_1) = \ddot{q}_T(t_1) \quad \text{(continuity of velocity and acceleration at } t_1) \\
& q_T(t_1) = q_1 \qquad q_T(t_2) = q_2 \\
& \dot{q}_T(t_2) = \dot{q}_S(t_2) \qquad \ddot{q}_T(t_2) = \ddot{q}_S(t_2) \quad \text{(continuity of velocity and acceleration at } t_2) \\
& q_S(t_2) = q_2 \qquad q_S(t_f) = q_f \qquad \dot{q}_S(t_f) = 0 \qquad \ddot{q}_S(t_f) = 0.
\end{aligned} \tag{7}$$

[2] In order to be interpreted as trigonometric values (sine and cosine of some angle), the arguments of each of the two ATAN2 functions in the eqs. (2) should be divided, respectively by $\sqrt{{}^0x_T^2 + {}^0y_T^2}$ and by $\sqrt{{}^0x_T^2 + {}^0y_T^2 + {}^0z_T^2}$. When ${}^0x_T^2 +^0 y_T^2 \neq 0$, this division by a (positive) value can be avoided. Instead, for ${}^0x_T^2 =^0 y_T^2 = 0$, the pan angle α would remain anyway undefined. So, such divisions are skipped altogether in eqs. (2), while introducing a preliminary warning in (or close to) a singularity. Similar arguments hold for eqs. (3).

D. Following the same order, the conditions (7) can be written in matrix form as

$$\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{b}, \tag{8}$$

with

$$\boldsymbol{A} = \begin{pmatrix}
1 & t_0 & t_0^2 & t_0^3 & t_0^4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 2t_0 & 3t_0^2 & 4t_0^3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2 & 6t_0 & 12t_0^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & t_1 & t_1^2 & t_1^3 & t_1^4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 2t_1 & 3t_1^2 & 4t_1^3 & 0 & -1 & -2t_1 & -3t_1^2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2 & 6t_1 & 12t_1^2 & 0 & 0 & -2 & -6t_1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & t_1 & t_1^2 & t_1^3 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & t_2 & t_2^2 & t_2^3 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 2t_2 & 3t_2^2 & 0 & -1 & -2t_2 & -3t_2^2 & -4t_2^3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 6t_2 & 0 & 0 & -2 & -6t_2 & -12t_2^2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & t_2 & t_2^2 & t_2^3 & t_2^4 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & t_f & t_f^2 & t_f^3 & t_f^4 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 6t_f & 12t_f^2
\end{pmatrix} \tag{9}$$

and

$$\boldsymbol{b} = \begin{pmatrix} q_0 & 0 & 0 & q_1 & 0 & 0 & q_1 & q_2 & 0 & 0 & q_2 & q_f & 0 & 0 \end{pmatrix}^T.$$

It can be shown that matrix $\boldsymbol{A}$ in (9) is always non-singular as long as $t_0 < t_1 < t_2 < t_f$, a condition which is satisfied by assumption. Using the problem data, we have

$$\boldsymbol{A} = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 2 & 4 & 8 & 16 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 4 & 12 & 32 & 0 & -1 & -4 & -12 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2 & 12 & 48 & 0 & 0 & -2 & -12 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 2 & 4 & 8 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 4 & 16 & 64 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 8 & 48 & 0 & -1 & -8 & -48 & -256 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 24 & 0 & 0 & -2 & -24 & -192 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 16 & 64 & 256 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 6 & 36 & 216 & 1296 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 12 & 108 & 864 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 36 & 432
\end{pmatrix}$$

and

$$\boldsymbol{b} = \begin{pmatrix} 0 & 0 & 0 & 10 & 0 & 0 & 10 & 80 & 0 & 0 & 80 & 90 & 0 & 0 \end{pmatrix}^T.$$

E. By using Matlab, the linear system of equations (8) is solved as

$$\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b},$$

providing the desired coefficients of the three polynomials:

$$\begin{aligned} \boldsymbol{x}^T &= \begin{pmatrix} a_{L0} & a_{L1} & a_{L2} & a_{L3} & a_{L4} & a_{T0} & a_{T1} & a_{T2} & a_{T3} & a_{S0} & a_{S1} & a_{S2} & a_{S3} & a_{S4} \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 & 0 & -.625 & 0.9375 & 90 & -125.5 & 56.25 & -6.25 & -990 & 742.5 & -191.25 & 21.875 & -0.9375 \end{pmatrix}. \end{aligned} \tag{10}$$

Figure 3 shows the position, velocity, and acceleration of the obtained 4-3-4 trajectory. It can be noted that the trajectory is fully symmetric because of the specific data distribution. As requested, the solution trajectory $q_{434}(t)$ is continuous up to the acceleration, and has zero initial and final values of the first two time derivatives.

Once the above numerical procedure has been coded (e.g., in Matlab), it can be used for other sets of data. For instance, we can change just the timing through the same knots, i.e.,

$$\begin{array}{lllll} t_0 = 0, & t_1 = 1, & t_2 = 4, & t_1 = 6 & [\text{s}] \\ q_0 = 0, & q_1 = 10, & q_2 = 80, & q_f = 90 & [\text{deg}]. \end{array} \tag{11}$$

Figure 4 shows the resulting 4-3-4 trajectory. Symmetry of motion is now lost.

Matrix $\boldsymbol{A}$ in (9) is clearly sparse. As a matter of fact, no special structure has been exploited in order to reduce the number of independent equations to be solved numerically. A more careful definition of the interpolating polynomials can lead instead to a complete analytical solution in closed form. For this, normalization of time is used in each motion phase, together with a symmetric definition of the first and last quartic polynomials (reversing time for the latter). Moreover, a similar idea is used as in spline interpolation in order to break the problem in three separate parts.

The analytical solution is found by the procedure **a–f** below. For the time being, assume that $q_0 \neq q_1$ and $q_2 \neq q_f$. Special cases will be treated at the end.

a. Define the quartic polynomial in the *Lift off* phase as

$$q_L(\tau_1) = q_0 + (q_1 - q_0)\left(a_{13}\,\tau_1^3 + a_{14}\,\tau_1^4\right), \qquad \tau_1 = \frac{t - t_0}{t_1 - t_0} \in [0, 1]. \tag{12}$$

It is easy to see that, by construction, $q_L(\tau_1)$ satisfies the three initial boundary conditions at $t = t_0$ (or, $\tau_1 = 0$) on position, velocity, and acceleration. Moreover,

$$q_L(1) = q_0 + (q_1 - q_0)\,(a_{13} + a_{14}) = q_1 \qquad \Rightarrow \qquad a_{13} + a_{14} = 1.$$

Therefore, for the derivatives and their values at $t = t_1$ (or, $\tau_1 = 1$) we have

$$\dot{q}_L(\tau_1) = \frac{q_1 - q_0}{t_1 - t_0}\left(3\,a_{13}\,\tau_1^2 + 4\,a_{14}\,\tau_1^3\right) \quad \Rightarrow \quad \dot{q}_L(1) = \frac{q_1 - q_0}{t_1 - t_0}\,(3 + a_{14}) \tag{13}$$

and

$$\ddot{q}_L(\tau_1) = \frac{q_1 - q_0}{(t_1 - t_0)^2}\left(6\,a_{13}\,\tau_1 + 12\,a_{14}\,\tau_1^2\right) \quad \Rightarrow \quad \ddot{q}_L(1) = \frac{6(q_1 - q_0)}{(t_1 - t_0)^2}\,(1 + a_{14})\,. \tag{14}$$

b. Define the quartic polynomial in the *Set down* phase as

$$q_S(\tau_3) = q_f + (q_2 - q_f)\left(a_{33}(1 - \tau_1)^3 + a_{34}(1 - \tau_1)^4\right), \qquad \tau_3 = \frac{t - t_2}{t_f - t_2} \in [0, 1]. \tag{15}$$

By construction, $q_S(\tau_3)$ satisfies the three final boundary conditions at $t = t_f$ (or, $\tau_3 = 1$) on position, velocity, and acceleration. Moreover,

$$q_S(0) = q_f + (q_2 - q_f)\,(a_{33} + a_{34}) = q_2 \qquad \Rightarrow \qquad a_{33} + a_{34} = 1.$$

Therefore, for the derivatives and their values at $t = t_2$ (or, $\tau_3 = 0$) we have

$$\dot{q}_S(\tau_3) = \frac{q_f - q_2}{t_f - t_2}\left(3\,a_{33}(1-\tau_3)^2 + 4\,a_{34}(1-\tau_3)^3\right) \quad \Rightarrow \quad \dot{q}_S(0) = \frac{q_f - q_2}{t_f - t_2}\,(3 + a_{34}) \tag{16}$$

and

$$\ddot{q}_S(\tau_3) = -\frac{q_f - q_2}{(t_f - t_2)^2}\left(6\,a_{33}(1-\tau_3) + 12\,a_{34}(1-\tau_3)^2\right) \quad \Rightarrow \quad \ddot{q}_S(0) = -\frac{6(q_f - q_2)}{(t_f - t_2)^2}\,(1 + a_{34})\,. \tag{17}$$

c. For the cubic polynomial in the *Travel* phase, we choose the symmetric form

$$q_T(\tau_2) = q_2\tau_2 + q_1(1-\tau_2) + a_{21}\,\tau_2(1-\tau_2)^2 + a_{22}\,\tau_2^2(1-\tau_2), \qquad \tau_2 = \frac{t - t_1}{t_2 - t_1} \in [0,1]. \tag{18}$$

By construction, $q_T(\tau_2)$ satisfies the interpolating conditions on position in $t = t_1$ ($\tau_2 = 0$) and $t = t_2$ ($\tau_2 = 1$). Denote at this stage the (yet unknown) velocities at the internal knots as v_1 and v_2, respectively. By imposing

$$\dot{q}_T(0) = v_1, \qquad \dot{q}_T(1) = v_2,$$

the cubic polynomial (18) becomes fully specified as

$$q_T(\tau_2) = q_1 + (q_2 - q_1)\tau_2 + \left[v_1(t_2 - t_1) + (q_2 - q_1)\right]\tau_2(1-\tau_2)^2 + \left[(q_2 - q_1) - v_2(t_2 - t_1)\right]\tau_2^2(1-\tau_2), \tag{19}$$

for $\tau_2 \in [0,1]$. For later use, we compute also the expression of its acceleration:

$$\ddot{q}_T(\tau_2) = \frac{6(q_2 - q_1)}{(t_2 - t_1)^2}\,(1 - 2\tau_2) + \frac{1}{t_2 - t_1}\,\left(6(v_1 + v_2)\,\tau_2 - 2(2v_1 + v_2)\right). \tag{20}$$

d. Using the expressions in (13) and (16), and imposing the equalities

$$\dot{q}_L(1) = v_1, \qquad \dot{q}_S(0) = v_2,$$

we solve for the coefficients of $q_L(\tau_1)$ and $q_S(\tau_3)$ as

$$a_{14} = v_1\,\frac{t_1 - t_0}{q_1 - q_0} - 3, \qquad a_{13} = 1 - a_{14} = 4 - v_1\,\frac{t_1 - t_0}{q_1 - q_0} \tag{21}$$

and

$$a_{34} = v_2\,\frac{t_f - t_2}{q_f - q_2} - 3, \qquad a_{33} = 1 - a_{34} = 4 - v_2\,\frac{t_f - t_2}{q_f - q_2}. \tag{22}$$

e. Having tailored the solution as above, we are left only with the problem of finding the correct values v_1 and v_2. For this, the conditions of continuity of the acceleration at the intermediate knots are used, namely

$$\ddot{q}_L(1) = \ddot{q}_T(0), \qquad \ddot{q}_T(1) = \ddot{q}_S(0).$$

Using eqs. (14) and (17), substituting therein the coefficients given by (21) and (22), and evaluating (20) at $\tau_2 = 0$ and $\tau_2 = 1$ yields

$$\begin{aligned}
\ddot{q}_L(1) = \frac{6v_1}{t_1 - t_0} - \frac{12(q_1 - q_0)}{(t_1 - t_0)^2} &= -\frac{2(v_2 + 2v_1)}{t_2 - t_1} + \frac{6(q_2 - q_1)}{(t_2 - t_1)^2} = \ddot{q}_T(0) \\
\ddot{q}_T(1) = \frac{2(v_1 + 2v_2)}{t_2 - t_1} - \frac{6(q_2 - q_1)}{(t_2 - t_1)^2} &= -\frac{6v_2}{t_f - t_2} + \frac{12(q_f - q_2)}{(t_f - t_2)^2} = \ddot{q}_S(0).
\end{aligned} \tag{23}$$

After some manipulation, the following linear system of *two* equations in *two* unknowns in matrix form is obtained:

$$\begin{aligned}
\boldsymbol{M}\begin{pmatrix} v_1 \\ v_2 \end{pmatrix} &= \begin{pmatrix} 6\,(t_2 - t_1) + 4\,(t_1 - t_0) & 2\,(t_1 - t_0) \\ 2\,(t_f - t_2) & 6\,(t_2 - t_1) + 4\,(t_f - t_2) \end{pmatrix}\begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \\
&= \begin{pmatrix} 6(q_2 - q_1)\,\dfrac{t_1 - t_0}{t_2 - t_1} + 12(q_1 - q_0)\,\dfrac{t_2 - t_1}{t_1 - t_0} \\ 6\,(q_2 - q_1)\,\dfrac{t_f - t_2}{t_2 - t_1} + 12\,(q_f - q_2)\,\dfrac{t_2 - t_1}{t_f - t_2} \end{pmatrix} = \begin{pmatrix} n_1 \\ n_2 \end{pmatrix} = \boldsymbol{n}.
\end{aligned} \tag{24}$$

Since

$$\det \boldsymbol{M} = 12\left\{3(t_2 - t_1)^3 + 2(t_2 - t_1)\,[(t_1 - t_0) + (t_f - t_2)] + (t_1 - t_0)(t_f - t_2)\right\} \neq 0, \tag{25}$$

its solution is

$$\begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \boldsymbol{M}^{-1}\boldsymbol{n} = \frac{1}{\det \boldsymbol{M}}\begin{pmatrix} 6\,(t_2 - t_1) + 4\,(t_f - t_2) & -2\,(t_1 - t_0) \\ -2\,(t_f - t_2) & 6\,(t_2 - t_1) + 4\,(t_1 - t_0) \end{pmatrix}\begin{pmatrix} n_1 \\ n_2 \end{pmatrix}, \tag{26}$$

where n_1 and n_2 are defined in (24), and eq. (25) is used for the determinant. The closed form solution of the problem is found by replacing the analytical expressions of v_1 and v_2 in eq. (19) for $q_T(\tau_2)$ and in eqs. (21–22) for $q_L(\tau_1)$ and $q_S(\tau_3)$.

f. Using the problem data, we obtain

$$v_1 = v_2 = 22.5 \quad \text{[deg/s]} \tag{27}$$

and so

$$a_{13} = -0.5, \quad a_{14} = 1.5, \quad a_{33} = -0.5, \quad a_{34} = 1.5.$$

Indeed, the resulting trajectory is the same as the one obtained with the numerical method. However, since the symbolic expressions of the used polynomials are different, also the associated numerical values of the coefficients will be different. In Fig. 3, the computed values (27) are shown as magenta dots in the velocity plots. Note also that identical values are obtained for the coefficients of the two quartic polynomials, because of the symmetric nature of the given data. For the alternative data in (11), we obtain instead (see the plots in Fig. 4)

$$v_1 = 21.4184, \qquad v_2 = 14.3972 \qquad \text{[deg/s]} \tag{28}$$

and

$$a_{13} = 1.8582, \quad a_{14} = -0.8582, \quad a_{33} = 1.1206, \quad a_{34} = -0.1206.$$

To complete the analysis, we have to consider the degenerate (or singular) cases, namely when $q_0 = q_1$ and/or $q_2 = q_f$. The numerical approach is totally unaffected by any repetition of values at the knots (i.e., matrix $\boldsymbol{A}$ does not lose rank). On the other hand, in the more tailored analytical approach some changes are needed. In fact, the expressions of the quartic polynomials (12) and (15) would be forced to become constant in such cases (while they should not, unless $q_0 = q_1 = q_2 = q_f$ and no motion is needed). In a singular case, it is sufficient to replace the previous definitions by

$$q_L(\tau_1) = q_0 + v_1(t_1 - t_0)\left(\tau_1^4 - \tau_1^3\right), \qquad \tau_1 = \frac{t - t_0}{t_1 - t_0} \in [0, 1]$$

when $q_1 = q_0$, and/or by

$$q_S(\tau_3) = q_f + v_2(t_f - t_2)\left((1 - \tau_3)^3 - (1 - \tau_3)^4\right), \qquad \tau_3 = \frac{t - t_2}{t_f - t_2} \in [0, 1]$$

when $q_2 = q_f$. In this way, we have in particular

$$\ddot{q}_L(1) = \frac{6v_1}{t_1 - t_0}, \qquad \ddot{q}_S(0) = -\frac{6v_2}{t_f - t_2}$$

that coincide with the expressions in (23) of the accelerations at the intermediate knots, respectively when $q_1 = q_0$ and when $q_2 = q_f$. Therefore, v_1 and v_2 can be found using (26) as before. For instance, consider the double degenerate case

$$\begin{gathered} t_0 = 0, \qquad t_1 = 2, \qquad t_2 = 4, \qquad t_1 = 6 \qquad \text{[s]} \\ q_0 = q_1 = 0, \qquad q_2 = q_f = 90 \qquad \text{[deg].} \end{gathered} \tag{29}$$

Figure 5 shows the result obtained via the analytical method, yielding again $v_1 = v_2 = 22.5$ [deg/s] (the same trajectory would be obtained also with the numerical method). From the position profile, we can see that there is some motion also in the first and last time intervals, which is needed to guarantee continuity up to the acceleration in the intermediate knots. However, an under- and over-shooting is present, which makes this particular situation no longer interesting for a collision-free guarded motion of the robot, when close to environmental surfaces.

Two different Matlab programs for the numerical solution and for the analytical solution (including treatment of degenerate cases) are available upon request.

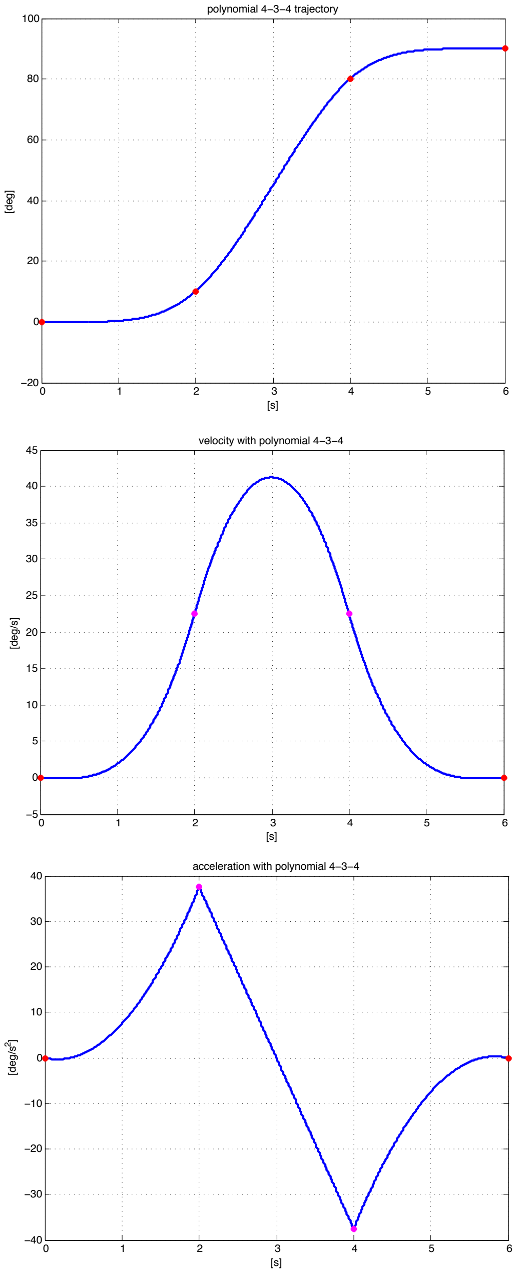

Figure 3: Position (top, the red dots are the interpolated knots), velocity (center, the magenta dots are the computed v_1 and v_2), and acceleration (bottom) of the 4-3-4 solution trajectory

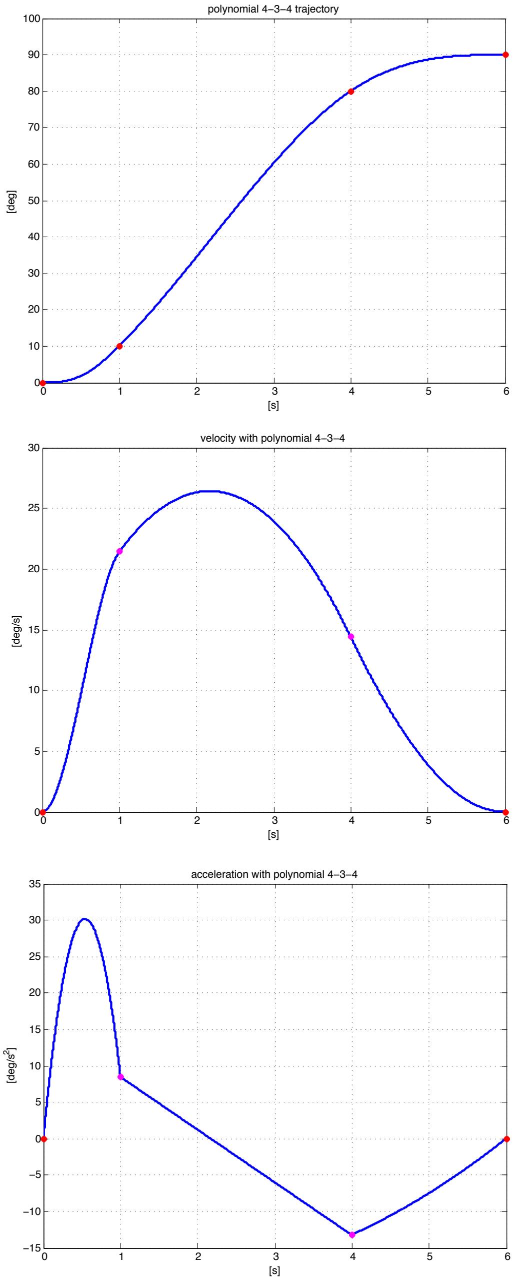

Figure 4: Position (top), velocity (center), and acceleration (bottom) of the 4-3-4 solution trajectory for the motion task in (11): Same knots, but different timing

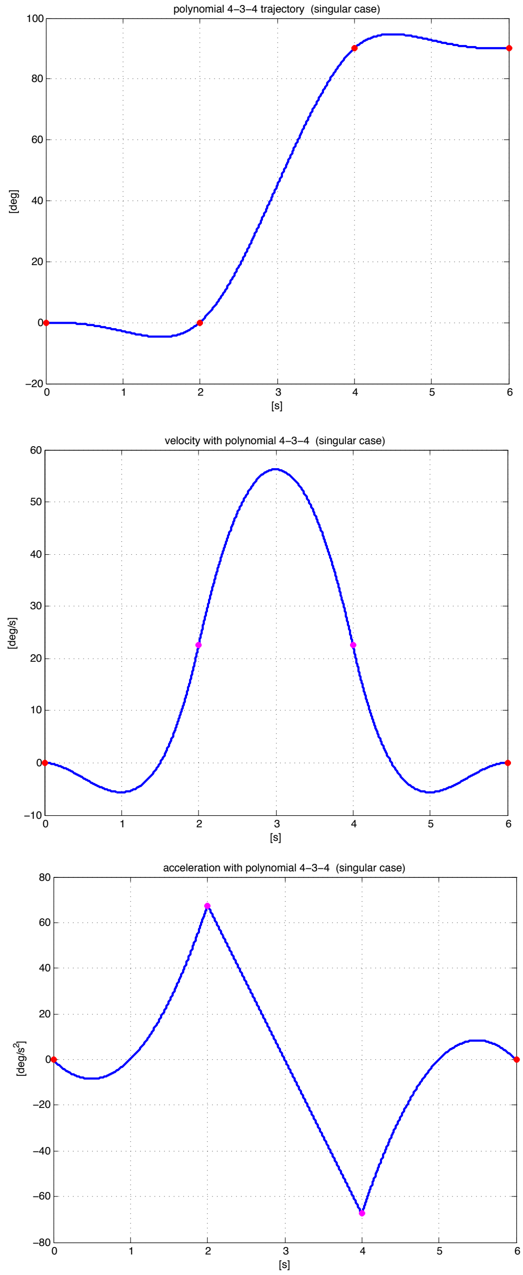

Figure 5: Position (top), velocity (center), and acceleration (bottom) of the 4-3-4 solution trajectory for the motion task in (29): Initial and first intermediate knot, as well as second intermediate and final knot have repeated position values

Exercise 3

The proof is based on Lyapunov stability, namely on the analysis of the time evolution of the error function $V = \frac{1}{2}\,\boldsymbol{e}^T\boldsymbol{e} = \frac{1}{2}\left(\boldsymbol{r}_d - \boldsymbol{r}\right)^T\left(\boldsymbol{r}_d - \boldsymbol{r}\right) \geq 0$. Using eq. (1) and dropping dependencies, we have

$$\begin{aligned} \dot{V} = \boldsymbol{e}^T\dot{\boldsymbol{e}} = \boldsymbol{e}^T\left(\dot{\boldsymbol{r}}_d - \dot{\boldsymbol{r}}\right) \;&=\; \boldsymbol{e}^T\dot{\boldsymbol{r}}_d - \boldsymbol{e}^T\boldsymbol{J}\dot{\boldsymbol{q}} \\ &=\; \boldsymbol{e}^T\dot{\boldsymbol{r}}_d - \boldsymbol{e}^T\boldsymbol{J}\left[\boldsymbol{J}^{-1}\dot{\boldsymbol{r}}_d + k\,\boldsymbol{J}^T\left(\boldsymbol{r}_d - \boldsymbol{r}\right)\right] = -k\,\boldsymbol{e}^T\boldsymbol{J}\boldsymbol{J}^T\boldsymbol{e} \leq 0. \end{aligned} \tag{30}$$

As long as $\boldsymbol{J}(\boldsymbol{q})$ is non-singular, we have $\boldsymbol{J}^T\boldsymbol{e} = \boldsymbol{0}$ (and so $\dot{V} = 0$) *if and only if* $\boldsymbol{e} = \boldsymbol{0}$. Therefore, the controlled robot will be an asymptotically stable system, and the error $\boldsymbol{e}(t)$ will converge to zero from any initial condition $\boldsymbol{e}(0)$. Note that, since the closed-loop system is still nonlinear, asymptotic stability (and convergence) will not be exponential in general.

The two terms in the control law (1), the first with the inverse Jacobian and the second with the Jacobian transpose, need both to be computed on line (i.e., all vectors and matrices are evaluated at the current configuration $\boldsymbol{q}$), even if the desired task trajectory $\boldsymbol{r}_d(t)$, $t \in [0, T]$, is completely known in advance for an arbitrary duration T.

When the robot is kinematically redundant for the given task ($m < n$), we can just replace in (1) the inverse of the Jacobian $\boldsymbol{J}$ by its pseudoinverse $\boldsymbol{J}^{\#}$. In fact, as long as the Jacobian is full (row) rank, it is $\boldsymbol{J}\boldsymbol{J}^{\#} = \boldsymbol{J}\boldsymbol{J}^T\left(\boldsymbol{J}\boldsymbol{J}^T\right)^{-1} = \boldsymbol{I}$. Therefore, we obtain the same cancellation of terms as in (30).

* * * * *