Robotics I

October 28, 2016

Exercise 1

Consider the following matrix

$$^{A}\boldsymbol{R}_B(\rho,\sigma) = \begin{pmatrix} \cos\rho & -\sin\rho & 0 \\ \sin\rho\cos\sigma & \cos\rho\cos\sigma & -\sin\sigma \\ \sin\rho\sin\sigma & \cos\rho\sin\sigma & \cos\sigma \end{pmatrix}.$$

- Prove that this is a rotation matrix (representing thus the orientation of a frame B with respect to a fixed frame A) for any value of the pair of angles (ρ, σ).
- Which is the sequence of two elementary rotations around *fixed* coordinate axes providing $^{A}\boldsymbol{R}_B(\rho,\sigma)$?
- Which is the sequence of two elementary rotations around *moving* coordinate axes providing $^{A}\boldsymbol{R}_B(\rho,\sigma)$?
- Verify your statements for $\rho = 90°$ and $\sigma = -90°$.

Exercise 2

Consider the planar 2R robot in Fig. 1, having link lengths $\ell_1 = 0.8$ and $\ell_2 = 0.6$ [m], and let the direct kinematic mapping that characterizes the position of its end-effector be defined as $\boldsymbol{p} = \boldsymbol{f}(\boldsymbol{q})$. The motion of this robot is controlled by specifying the joint accelerations $\ddot{\boldsymbol{q}}$.

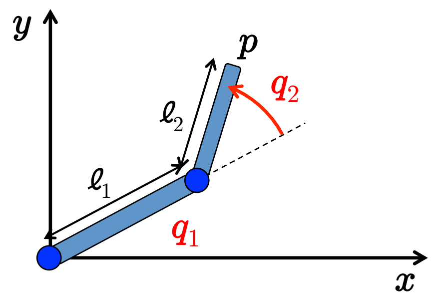

Figure 1: A planar 2R robot.

- What is the expression of the nominal joint acceleration command $\ddot{\boldsymbol{q}} = \ddot{\boldsymbol{q}}_d$ when the robot is in a state $(\boldsymbol{q}, \dot{\boldsymbol{q}})$ and its end-effector needs to move instantaneously with a desired acceleration $\ddot{\boldsymbol{p}}_d$? Try out your expression by determining the numerical value of $\ddot{\boldsymbol{q}}_d$ at the time instant $t = \bar{t} = 0.8$ [s], when

$$\boldsymbol{q}(\bar{t}) = \begin{pmatrix} 0 \\ \pi/2 \end{pmatrix} \text{ [rad]}, \qquad \dot{\boldsymbol{q}}(\bar{t}) = \begin{pmatrix} -\pi \\ \pi \end{pmatrix} \text{ [rad/s]}, \qquad \boldsymbol{p}_d(t) = \begin{pmatrix} 0 \\ 0.6\,(t^3 - 1) \end{pmatrix} \text{ [m]}.$$

- For the same end-effector trajectory specified above, assume now that, at time $t = 0$, the robot is in an initial state $(\boldsymbol{q}(0), \dot{\boldsymbol{q}}(0))$ such that $\boldsymbol{p}(0) = \boldsymbol{f}(\boldsymbol{q}(0)) = \boldsymbol{p}_d(0)$, but $\dot{\boldsymbol{p}}(0) \neq \dot{\boldsymbol{p}}_d(0)$. What should be the expression of the feedback control law for the joint acceleration $\ddot{\boldsymbol{q}}$ in order to recover the initial Cartesian trajectory error over time, achieving thus asymptotic trajectory tracking? Define all the needed terms and parameters in this second-order kinematic control law, and determine accordingly the initial numerical value $\ddot{\boldsymbol{q}}(0)$ of the control law.

[150 minutes; open books]

Solution

October 28, 2016

Exercise 1

It is easy to verify that the given matrix ${}^A\boldsymbol{R}_B(\rho,\sigma)$ is a rotation matrix: for any pair (ρ,σ), its three columns are of unitary norm and orthogonal each to other, while $\det {}^A\boldsymbol{R}_B(\rho,\sigma) = +1$. Moreover, matrix ${}^A\boldsymbol{R}_B(\rho,\sigma)$ is obtained as the product of two elementary rotation matrices in the form

$$\boldsymbol{R}_{\boldsymbol{x}}(\sigma)\boldsymbol{R}_{\boldsymbol{z}}(\rho) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\sigma & -\sin\sigma \\ 0 & \sin\sigma & \cos\sigma \end{pmatrix} \begin{pmatrix} \cos\rho & -\sin\rho & 0 \\ \sin\rho & \cos\rho & 0 \\ 0 & 0 & 1 \end{pmatrix} = {}^A\boldsymbol{R}_B(\rho,\sigma).$$

Therefore, it represents

- either a sequence of two rotations around *fixed* axes: first a rotation by ρ around the $\boldsymbol{z}$-axis, and then a rotation by σ around the original $\boldsymbol{x}$-axis;
- or, a sequence of two rotations around *moving* axes: first a rotation by σ around the $\boldsymbol{x}$-axis, and then a rotation by ρ around the already rotated $\boldsymbol{z}$-axis (i.e., $\boldsymbol{z}'$).

By substituting $\rho = \pi/2$ and $\sigma = -\pi/2$, we obtain

$${}^A\boldsymbol{R}_B(\pi/2,-\pi/2) = \begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \boldsymbol{R}_{\boldsymbol{x}}(-\pi/2)\boldsymbol{R}_{\boldsymbol{z}}(\pi/2).$$

Considering for example the case of moving axes, the first (clockwise) rotation by $\sigma = -\pi/2$ keeps the $\boldsymbol{x}$-axis unchanged, $\boldsymbol{x}' \equiv \boldsymbol{x}$, while $\boldsymbol{y}' \equiv -\boldsymbol{z}$ and $\boldsymbol{z}' \equiv \boldsymbol{y}$; the second (counterclockwise) rotation by $\rho = \pi/2$ keeps the current $\boldsymbol{z}'$-axis unchanged, $\boldsymbol{z}'' \equiv \boldsymbol{z}'$, while $\boldsymbol{x}'' \equiv \boldsymbol{y}'$ and $\boldsymbol{y}'' \equiv -\boldsymbol{x}'$; concatenating the two rotations, we obtain $\boldsymbol{x}'' \equiv -\boldsymbol{z}$, $\boldsymbol{y}'' \equiv -\boldsymbol{x}$, and $\boldsymbol{z}'' \equiv \boldsymbol{y}$, which is ${}^A\boldsymbol{R}_B(\pi/2,-\pi/2)$ as expected.

Exercise 2

The following piece of Matlab code summarizes the computations needed to answer to the first question:

```
% first question
tbar=0.8;
ddp1=0;ddp2=3.6*tbar;
ddpd=[ddp1;ddp2]  % outputs the desired Cartesian acceleration at time t=0.8 s
% current state
q1=0;q2=pi/2;
dq1=-pi;dq2=pi;
dq=[dq1; dq2];
% direct kinematics
p=[l1*cos(q1)+l2*cos(q1+q2);
     l1*sin(q1)+l2*sin(q1+q2)];
% Jacobian matrix
J=[-l1*sin(q1)-l2*sin(q1+q2) -l2*sin(q1+q2);
    l1*cos(q1)+l2*cos(q1+q2)  l2*cos(q1+q2)];
% time derivative of the Jacobian
dJ=[-l1*cos(q1)*dq1-l2*cos(q1+q2)*(dq1+dq2) -l2*cos(q1+q2)*(dq1+dq2);
    -l1*sin(q1)*dq1-l2*sin(q1+q2)*(dq1+dq2) -l2*sin(q1+q2)*(dq1+dq2)];
ddqd=inv(J)*(ddpd - dJ*dq) % outputs the requested joint acceleration command
% end
```

The two resulting outputs of this code are

$$\ddot{\boldsymbol{p}}_d(0.8) = \begin{pmatrix} 0 \\ 2.88 \end{pmatrix} \text{ [m/s}^2\text{]}, \qquad \ddot{\boldsymbol{q}}_d(0.8) = \begin{pmatrix} 3.6 \\ -16.7595 \end{pmatrix} \text{ [rad/s}^2\text{]}.$$

Similarly, at time $t = 0$ we request

$$\boldsymbol{p}_d(0) = \left. \begin{pmatrix} 0 \\ 0.6\,(t^3 - 1) \end{pmatrix} \right|_{t=0} = \begin{pmatrix} 0 \\ -0.6 \end{pmatrix} \text{ [m]}$$

and

$$\dot{\boldsymbol{p}}_d(0) = \left. \begin{pmatrix} 0 \\ 1.8\,t^2 \end{pmatrix} \right|_{t=0} = \mathbf{0} \text{ [m/s]}, \qquad \ddot{\boldsymbol{p}}_d(0) = \left. \begin{pmatrix} 0 \\ 3.6\,t \end{pmatrix} \right|_{t=0} = \mathbf{0} \text{ [m/s}^2\text{]}.$$

The robot should be in an initial state $(\boldsymbol{q}(0), \dot{\boldsymbol{q}}(0))$ such that $\boldsymbol{p}(0) = \boldsymbol{f}(\boldsymbol{q}(0)) = \boldsymbol{p}_d(0)$, but $\dot{\boldsymbol{p}}(0) \neq \dot{\boldsymbol{p}}_d(0)$. To determine $\boldsymbol{q}(0)$, we solve the inverse kinematics for $\boldsymbol{p}_d(0)$, picking just one of the two solutions (in an arbitrary way):

```
pd0=[0; -0.6];
% second joint computations
c2=(pd0(1)^2+pd0(2)^2-l1^2-l2^2)/(2*l1*l2);
s2=sqrt(1-c2^2);  %other solution: -sqrt(1-c2^2)
% first joint computations
det=l1^2+l2^2+2*l1*l2*c2;
s1=(pd0(2)*(l1+l2*c2)-pd0(1)*l2*s2)/det;
c1=(pd0(1)*(l1+l2*c2)+pd0(2)*l2*s2)/det;
% output
q01=atan2(s1,c1);
q02=atan2(s2,c2);
q0=[q01; q02]
```

We note that the desired Cartesian position is strictly inside the workspace of the 2R robot, so that we are away from kinematic singularities. The output of the above code gives

$$\boldsymbol{q}(0) = \begin{pmatrix} -2.4119 \\ 2.3005 \end{pmatrix} \text{ [rad]} = \begin{pmatrix} -138.19 \\ 131.81 \end{pmatrix} \text{ [deg]},$$

yielding no initial Cartesian position error at $t = 0$, $\boldsymbol{e}(0) = \boldsymbol{p}_d(0) - \boldsymbol{p}(0) = \boldsymbol{p}_d(0) - \boldsymbol{f}(\boldsymbol{q}(0)) = \mathbf{0}$, as desired. In order to be sure that $\dot{\boldsymbol{p}}(0) = \boldsymbol{J}(\boldsymbol{q}(0))\dot{\boldsymbol{q}}(0) \neq \dot{\boldsymbol{p}}_d(0) = \mathbf{0}$, we just need to avoid the specific choice $\dot{\boldsymbol{q}}(0) = \mathbf{0}$. For example, by choosing

$$\dot{\boldsymbol{q}}(0) = \begin{pmatrix} 0.5 \\ 0.1 \end{pmatrix} \text{ [rad/s]} \qquad \Rightarrow \qquad \dot{\boldsymbol{e}}(0) = \dot{\boldsymbol{p}}_d(0) - \dot{\boldsymbol{p}}(0) = -\boldsymbol{J}(\boldsymbol{q}(0))\,\dot{\boldsymbol{q}}(0) = \begin{pmatrix} -0.3067 \\ -0.0596 \end{pmatrix} \text{ [m/s]}.$$

To recover any initial Cartesian trajectory error (in velocity and/or position) over time and achieve thus asymptotic trajectory tracking, the control law for the joint acceleration input should be chosen as

$$\ddot{\boldsymbol{q}} = \boldsymbol{J}^{-1}(\boldsymbol{q}) \left(\ddot{\boldsymbol{p}}_d + \boldsymbol{K}_d \,(\dot{\boldsymbol{p}}_d - \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}) + \boldsymbol{K}_p \,(\boldsymbol{p}_d - \boldsymbol{f}(\boldsymbol{q})) - \dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}} \right),$$

with (symmetric) gain matrices $\boldsymbol{K}_p > 0$, $\boldsymbol{K}_d > 0$ (a PD feedback action). By choosing for instance

$$\boldsymbol{K}_p = 100 \cdot \boldsymbol{I}_{2\times 2}, \qquad \boldsymbol{K}_d = 20 \cdot \boldsymbol{I}_{2\times 2},$$

we finally obtain at time $t = 0$

$$\ddot{\boldsymbol{q}}(0) = \begin{pmatrix} -9.8614 \\ -2.2639 \end{pmatrix} \text{ [rad/s}^2\text{]}.$$

* * * * *