Robotics II

February 5, 2018

Exercise 1

Consider a robot manipulator with $\boldsymbol{q} \in \mathbb{R}^n$ joint variables that is redundant with respect to a task described by $\boldsymbol{r} \in \mathbb{R}^m$ variables, with $m < n$. The $m \times n$ task Jacobian matrix $\boldsymbol{J}(\boldsymbol{q})$ relates task and joint velocities, i.e., $\dot{\boldsymbol{r}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}$. For task regulation problems, kinematic control typically defines a law of the form

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{\#}(\boldsymbol{q})\boldsymbol{K}\boldsymbol{e} + \Big(\boldsymbol{I} - \boldsymbol{J}^{\#}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})\Big)\,\dot{\boldsymbol{q}}_0, \qquad \boldsymbol{K} > 0\,(\text{diagonal}), \quad \boldsymbol{e} = \boldsymbol{r}_d - \boldsymbol{r}, \tag{1}$$

where $\boldsymbol{r}_d$ is the desired value for the task variables. The first term in (1) leads to $\dot{\boldsymbol{e}} = -\boldsymbol{K}\boldsymbol{e}$, so that exponential convergence of the error $\boldsymbol{e}$ to zero is guaranteed (out of task singularities). The second term allows shaping the robot configuration during motion without affecting task execution, using a joint velocity $\dot{\boldsymbol{q}}_0 \in \mathbb{R}^n$ projected in the null space of the task Jacobian. When $m = n$, there is no null space to explore (the second term vanishes); when $m > n$, the pseudoinverse command $\dot{\boldsymbol{q}} = \boldsymbol{J}^{\#}(\boldsymbol{q})\boldsymbol{K}\boldsymbol{e}$ guarantees only that a minimum error (in norm) is achieved, but still with $\boldsymbol{e} \neq \boldsymbol{0}$.

With the above in mind, consider a visual servoing problem that uses as task variables M point features, namely the coordinates $\boldsymbol{f}_i = (u_i\ v_i)^T \in \mathbb{R}^2$, $i = 1, \ldots, M$, of M points on the 2D image plane of an eye-in-hand camera. Let the task vector be $\boldsymbol{r} = \boldsymbol{f} \in \mathbb{R}^m$, with $m = 2M$, while vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ (both in $\mathbb{R}^M$) collect the coordinates of the image points. The task Jacobian $\boldsymbol{J}$ is the product $\boldsymbol{J}_p(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{Z})\boldsymbol{J}_m(\boldsymbol{q})$, where $\boldsymbol{J}_p$ is the $m \times 6$ interaction matrix of the M point features (depending on the sensed image and on the depths $Z_i > 0$ of the 3D points), and $\boldsymbol{J}_m$ is the $6 \times n$ geometric Jacobian of the manipulator carrying the camera. The rank of $\boldsymbol{J}$ will be equal to $\rho \leq \min\{m, 6, n\}$, no matter how large n is, and the dimension of the null space $\mathcal{N}\{\boldsymbol{J}\}$ will be $n - \rho$.

To handle critical issues related to the lack of full row rank for the task Jacobian, it was recently proposed to modify the way the regulation task is accomplished. Instead of considering the original task as a m-dimensional vector, so as to achieve the goal $\boldsymbol{r} = \boldsymbol{r}_d$, or $\boldsymbol{e} = \boldsymbol{0}$, by driving each and every component to its desired value, we can define the task as the norm of the error $\boldsymbol{e} \in \mathbb{R}^m$,

$$\eta = \|\boldsymbol{e}\|, \tag{2}$$

and achieve the goal by specifying $\eta = 0$ as the desired task value. This is indeed equivalent to obtaining $\boldsymbol{e} = \boldsymbol{0}$. Thanks to the modified definition, the task is one-dimensional and the null space of the associated task Jacobian will always be of dimension equal to (at least) $n - 1$.

Formulate and address the problem of task-based kinematic control for $\eta \in \mathbb{R}$. In particular:

- determine the general form of the $1 \times n$ task Jacobian matrix $\boldsymbol{J}_\eta$ associated to (2);
- analyze the singularities and/or the problems of definition of $\boldsymbol{J}_\eta$;
- write the explicit expression of the pseudoinverse $\boldsymbol{J}_\eta^{\#}$ in the control law (1), assuming that $\boldsymbol{e} \neq 0$ and $\boldsymbol{J}_\eta$ is full (row) rank;
- specify the main terms needed in this type of task-based kinematic control for the case of a visual servoing task with $M = 2$ point features.

[Turn sheet for next exercises]

Exercise 2

For a robot of the cylindrical type with a sequence PRP of joints and mounted on a vertical wall, provide the dynamic model in the usual form

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{F}\dot{\boldsymbol{q}} = \boldsymbol{\tau}, \tag{3}$$

using the generalized coordinates $\boldsymbol{q} \in \mathbb{R}^3$ and the dynamic coefficients defined in Fig. 1. In (3), $\boldsymbol{F} > 0$ is a diagonal matrix of viscous coefficients. Make reasonable assumptions on the zero values of the variables q_i, $i = 1, 2, 3$, and neglect the small offset between joint axes 2 and 3 (i.e., assume that these two axes intersect). Moreover, assume that the center of mass of each link is placed on the joint axis having the same index.

Without any a priori knowledge of dynamic parameters, define all the terms needed in the design of an adaptive control law for this robot so as to achieve global asymptotic tracking of a desired joint trajectory $\boldsymbol{q}_d(t)$, with $q_{di}(t) = q_{0i} + A_i(1 - \cos(2\pi t/T))^\nu$ $(i = 1, 2, 3)$ of amplitude A_i and period T, for an unlimited time $t \geq 0$. Which is the minimum value of the integer $\nu \in \mathbb{N}$ that allows asymptotic exact tracking?

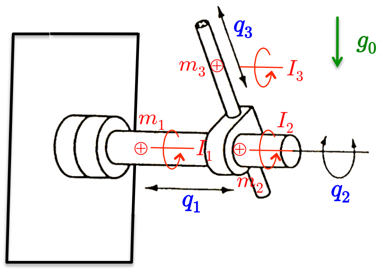

Figure 1: A cylindrical PRP-type robot mounted on a vertical wall, with its generalized coordinates $\boldsymbol{q} = (q_1, q_2, q_3)$ and relevant dynamic parameters.

Exercise 3

Consider an actuated pendulum that moves in the vertical plane under a joint torque u. The single link is a uniform thin rod of mass $m = 10$ kg and length $L = 2$ m. The downward equilibrium is at $\theta = 0$. Suppose that the link is initially at rest in $\theta_0 = 0$, and that we want to regulate its angular position to $\theta_d = \pi/3$ rad using the control scheme

$$u = k_p(\theta_d - \theta) - k_d\dot{\theta} + u_{i-1}, \qquad i = 1, 2, \ldots \tag{4}$$

with $k_p = 500$ and $k_d = 45$. The constant feedforward u_{i-1} in (4) is updated at every new reached equilibrium configuration θ_i, $i = 1, 2, \ldots$, as

$$u_i = u_{i-1} + k_p(\theta_d - \theta_i), \quad i = 1, 2, \ldots, \qquad \text{with } u_0 = 0. \tag{5}$$

Will the error $e_i = \theta_d - \theta_i$ converge to zero for $i \to \infty$ with this iterative control scheme? Why, or why not? If (or when) it does, can you predict which is the minimum number $i_{\min} > 0$ of iterations guaranteeing that the error will satisfy $|e_i| < \varepsilon = 0.01$ rad for all $i \geq i_{\min}$?

[150 minutes; open books]

Solution

February 5, 2018

Exercise 1

We first rewrite more explicitly eq. (2) as

$$\eta = \|\boldsymbol{e}\| = \sqrt{\boldsymbol{e}^T\boldsymbol{e}} = \|\boldsymbol{r}_d - \boldsymbol{r}\|. \tag{6}$$

Since $\boldsymbol{r}_d$ is constant, taking the time derivative of (6) yields

$$\dot{\eta} = \frac{1}{2}\,\frac{2\,\boldsymbol{e}^T\dot{\boldsymbol{e}}}{\sqrt{\boldsymbol{e}^T\boldsymbol{e}}} = -\frac{\boldsymbol{e}^T\dot{\boldsymbol{r}}}{\eta} = -\frac{\boldsymbol{e}^T\boldsymbol{J}(\boldsymbol{q})}{\|\boldsymbol{e}\|}\,\dot{\boldsymbol{q}} = \boldsymbol{J}_\eta(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{7}$$

where $\boldsymbol{J}$ is the Jacobian matrix associated to the original task variables $\boldsymbol{r}$. Thus, the new $1\times n$ task Jacobian is

$$\boldsymbol{J}_\eta(\boldsymbol{q}) = -\frac{\boldsymbol{e}^T\boldsymbol{J}(\boldsymbol{q})}{\|\boldsymbol{e}\|}. \tag{8}$$

This matrix has rank one whenever $\boldsymbol{e} \notin \mathcal{N}\{\boldsymbol{J}^T(\boldsymbol{q})\}$. On the other hand, the condition $\boldsymbol{e} = \boldsymbol{0}$ (i.e., exactly where the task is accomplished!) is critical because both the numerator and the denominator go to zero, so that a further analysis is needed (which is out of the scope of this exercise). Far from these situations, and similarly to (1), we define the control law as

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_\eta^{\#}(\boldsymbol{q})\,k\,\eta + \left(\boldsymbol{I} - \boldsymbol{J}_\eta^{\#}(\boldsymbol{q})\boldsymbol{J}_\eta(\boldsymbol{q})\right)\dot{\boldsymbol{q}}_0, \tag{9}$$

with the pseudoinverse $\boldsymbol{J}_\eta^{\#}$ of the row vector $\boldsymbol{J}_\eta$ being the column vector computed as

$$\boldsymbol{J}_\eta^{\#}(\boldsymbol{q}) = \boldsymbol{J}_\eta^T(\boldsymbol{q})\left(\boldsymbol{J}_\eta(\boldsymbol{q})\boldsymbol{J}_\eta^T(\boldsymbol{q})\right)^{-1} = -\frac{\|\boldsymbol{e}\|}{\boldsymbol{e}^T\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{e}}\,\boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{e}. \tag{10}$$

It is easy to see that (9) will work properly: in fact, plugging this $\dot{\boldsymbol{q}}$ in (7) leads to the exponentially stable (scalar) error system

$$\dot{\eta} = -k\,\eta. \tag{11}$$

Moreover, the projection matrix $\boldsymbol{P} = \boldsymbol{I} - \boldsymbol{J}_\eta^{\#}\boldsymbol{J}_\eta$ in (9) has rank one, and the null space in which we can now accommodate a desired extra motion $\dot{\boldsymbol{q}}_0$ is now $(n-1)$-dimensional.

For a visual servoing task with $M = 2$ point features, $\boldsymbol{r} = \boldsymbol{f} \in \mathbb{R}^4$, we need first to define the 4×6 interaction matrix $\boldsymbol{J}_p$, as well as the (generic) $6\times n$ geometric Jacobian $\boldsymbol{J}_m$, so that

$$\dot{\boldsymbol{f}} = \boldsymbol{J}_p(\boldsymbol{u},\boldsymbol{v},\boldsymbol{Z})\begin{pmatrix} V \\ \Omega \end{pmatrix}, \qquad \begin{pmatrix} V \\ \Omega \end{pmatrix} = \boldsymbol{J}_m(\boldsymbol{q})\dot{\boldsymbol{q}} \qquad \Rightarrow \qquad \boldsymbol{J} = \boldsymbol{J}_p\boldsymbol{J}_m, \tag{12}$$

being $V \in \mathbb{R}^3$ and $\Omega \in \mathbb{R}^3$, respectively the linear and angular velocity of the eye-in-hand camera. The interaction matrix takes the expression

$$\boldsymbol{J}_p(\boldsymbol{u},\boldsymbol{v},\boldsymbol{Z}) = \begin{pmatrix} \boldsymbol{J}_{p1}(u_1,v_1,Z_1) \\ \boldsymbol{J}_{p2}(u_2,v_2,Z_2) \end{pmatrix} = \begin{pmatrix} -\dfrac{\lambda}{Z_1} & 0 & \dfrac{u_1}{Z_1} & \dfrac{u_1v_1}{\lambda} & -\left(\lambda + \dfrac{u_1^2}{\lambda}\right) & v_1 \\ 0 & -\dfrac{\lambda}{Z_1} & \dfrac{v_1}{Z_1} & \lambda + \dfrac{v_1^2}{\lambda} & -\dfrac{u_1v_1}{\lambda} & -u_1 \\ -\dfrac{\lambda}{Z_2} & 0 & \dfrac{u_2}{Z_2} & \dfrac{u_2v_2}{\lambda} & -\left(\lambda + \dfrac{u_2^2}{\lambda}\right) & v_2 \\ 0 & -\dfrac{\lambda}{Z_2} & \dfrac{v_2}{Z_2} & \lambda + \dfrac{v_2^2}{\lambda} & -\dfrac{u_2v_2}{\lambda} & -u_2 \end{pmatrix}, \tag{13}$$

with the camera focal length $\lambda > 0$. Moreover, from

$$\boldsymbol{f} = \begin{pmatrix} u_1 & v_1 & u_2 & v_2 \end{pmatrix}^T, \qquad \boldsymbol{f}_d = \begin{pmatrix} u_{1d} & v_{1d} & u_{2d} & v_{2d} \end{pmatrix}^T, \qquad \boldsymbol{e} = \boldsymbol{f}_d - \boldsymbol{f}, \tag{14}$$

the scalar task takes the expression

$$\eta = \|\boldsymbol{e}\| = \sqrt{(u_{1d} - u_1)^2 + (v_{1d} - v_1)^2 + (u_{2d} - u_2)^2 + (v_{2d} - v_2)^2}. \tag{15}$$

The associated task Jacobian is compactly written as

$$\boldsymbol{J}_\eta = -\frac{1}{\eta}\,\boldsymbol{e}^T \begin{pmatrix} \boldsymbol{J}_{p1}(u_1, v_1, Z_1) \\ \boldsymbol{J}_{p2}(u_2, v_2, Z_2) \end{pmatrix} \boldsymbol{J}_m(\boldsymbol{q}). \tag{16}$$

Finally, the pseudoinverse (10) takes the form

$$\boldsymbol{J}_\eta^{\#} = -\eta\, \frac{\boldsymbol{J}_m^T(\boldsymbol{q}) \begin{pmatrix} \boldsymbol{J}_{p1}^T(u_1, v_1, Z_1) & \boldsymbol{J}_{p2}^T(u_2, v_2, Z_2) \end{pmatrix} \boldsymbol{e}}{\boldsymbol{e}^T \begin{pmatrix} \boldsymbol{J}_{p1}(u_1, v_1, Z_1) \\ \boldsymbol{J}_{p2}(u_2, v_2, Z_2) \end{pmatrix} \boldsymbol{J}_m(\boldsymbol{q})\, \boldsymbol{J}_m^T(\boldsymbol{q}) \begin{pmatrix} \boldsymbol{J}_{p1}^T(u_1, v_1, Z_1) & \boldsymbol{J}_{p2}^T(u_2, v_2, Z_2) \end{pmatrix} \boldsymbol{e}}. \tag{17}$$

Exercise 2

Following a Lagrangian approach, we compute first the kinetic energy $T = T_1 + T_2 + T_3$. Since the position of the center of mass of the third link will be an unknown dynamic parameter, we need to define q_3 in a purely kinematic way as the radial position of the distal end of the third link with respect to the joint axis 2. The radial position of the center of mass of link 3 will then be given by $q_3 - d_3$, being $d_3 > 0$ the distance of the center of mass from the link end. With this and under the other given assumptions, considering the sequence PRP of joint types we have

$$\begin{aligned} T_1 &= \tfrac{1}{2} m_1 \dot{q}_1^2, \qquad T_2 = \tfrac{1}{2} m_2 \dot{q}_1^2 + \tfrac{1}{2} I_2\, \dot{q}_2^2, \\ T_3 &= \tfrac{1}{2} m_3 \left(\dot{q}_1^2 + (q_3 - d_3)^2\, \dot{q}_2^2 + \dot{q}_3^2\right) + \tfrac{1}{2} I_3\, \dot{q}_2^2 \end{aligned} \qquad \Rightarrow \qquad T = \tfrac{1}{2} \dot{\boldsymbol{q}}^T \boldsymbol{M}(\boldsymbol{q}) \dot{\boldsymbol{q}},$$

with

$$\boldsymbol{M}(\boldsymbol{q}) = \begin{pmatrix} m_1 + m_2 + m_3 & 0 & 0 \\ 0 & I_2 + I_3 + m_3 (q_3 - d_3)^2 & 0 \\ 0 & 0 & m_3 \end{pmatrix}. \tag{18}$$

For the Coriolis and centrifugal terms, the requested adaptive control law will use the factorization $\boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \dot{\boldsymbol{q}}$ such that $\dot{\boldsymbol{M}} - 2\boldsymbol{C}$ is a skew-symmetric matrix. This is automatically guaranteed if the components of the Coriolis and centrifugal vector $\boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ are computed using the Christoffel's symbols, i.e.,

$$c_i(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \dot{\boldsymbol{q}}^T \boldsymbol{C}_i(\boldsymbol{q}) \dot{\boldsymbol{q}}, \quad \boldsymbol{C}_i(\boldsymbol{q}) = \frac{1}{2} \left(\frac{\partial \boldsymbol{m}_i(\boldsymbol{q})}{\partial \boldsymbol{q}} + \left(\frac{\partial \boldsymbol{m}_i(\boldsymbol{q})}{\partial \boldsymbol{q}} \right)^T - \frac{\partial \boldsymbol{M}(\boldsymbol{q})}{\partial \boldsymbol{q}_i} \right), \quad i = 1, 2, 3, \tag{19}$$

being $\boldsymbol{m}_i$ the ith column of the inertia matrix $\boldsymbol{M}$. Using (18) and (19), we obtain

$$\boldsymbol{C}_1(\boldsymbol{q}) = \boldsymbol{0} \qquad \Rightarrow \qquad c_1(\boldsymbol{q}, \dot{\boldsymbol{q}}) = 0,$$

$$\boldsymbol{C}_2(\boldsymbol{q}) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & m_3 (q_3 - d_3) \\ 0 & m_3 (q_3 - d_3) & 0 \end{pmatrix} \qquad \Rightarrow \qquad c_2(\boldsymbol{q}, \dot{\boldsymbol{q}}) = 2 m_3 (q_3 - d_3)\, \dot{q}_2 \dot{q}_3,$$

$$\boldsymbol{C}_3(\boldsymbol{q}) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -m_3 (q_3 - d_3) & 0 \\ 0 & 0 & 0 \end{pmatrix} \qquad \Rightarrow \qquad c_3(\boldsymbol{q}, \dot{\boldsymbol{q}}) = -m_3 (q_3 - d_3)\, \dot{q}_2^2.$$

A factorization that satisfies the skew-symmetric property is then given by

$$\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) = \begin{pmatrix} \dot{\boldsymbol{q}}^T\boldsymbol{C}_1(\boldsymbol{q}) \\ \dot{\boldsymbol{q}}^T\boldsymbol{C}_2(\boldsymbol{q}) \\ \dot{\boldsymbol{q}}^T\boldsymbol{C}_3(\boldsymbol{q}) \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & m_3(q_3-d_3)\,\dot{q}_3 & m_3(q_3-d_3)\,\dot{q}_2 \\ 0 & -m_3(q_3-d_3)\,\dot{q}_2 & 0 \end{pmatrix}. \tag{20}$$

For the potential energy due to gravity, $U_g = U_1 + U_2 + U_3$, we have (up to a constant)

$$U_1 = 0, \qquad U_2 = 0, \qquad U_3 = -m_3 g_0\,(q_3-d_3)\cos q_2,$$

where we assumed that the third link is vertical and points downward for $q_2 = 0$. Thus

$$\boldsymbol{g}(\boldsymbol{q}) = \left(\frac{\partial U_g(\boldsymbol{q})}{\partial \boldsymbol{q}}\right)^T = \begin{pmatrix} 0 \\ m_3 g_0\,(q_3-d_3)\sin q_2 \\ -m_3 g_0 \cos q_2 \end{pmatrix}. \tag{21}$$

The dynamic model of the robot, including the viscous friction term with $\boldsymbol{F} = \operatorname{diag}\{f_1, f_2, f_3\}$, can thus be written in the linear parametrized form,

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{F}\dot{\boldsymbol{q}} = \boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\,\boldsymbol{a} = \boldsymbol{\tau}, \tag{22}$$

with the 3×7 regressor

$$\boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}}) = \begin{pmatrix} \ddot{q}_1 & 0 & 0 & 0 & \dot{q}_1 & 0 & 0 \\ 0 & \ddot{q}_2 & q_3^2\,\ddot{q}_2 + 2q_3\,\dot{q}_2\dot{q}_3 + g_0\sin q_2 & -2q_3\,\ddot{q}_2 - 2\,\dot{q}_2\dot{q}_3 - g_0\sin q_2 & 0 & \dot{q}_2 & 0 \\ 0 & 0 & \ddot{q}_3 - q_3\,\dot{q}_2^2 - g_0\cos q_2 & \dot{q}_2^2 & 0 & 0 & \dot{q}_3 \end{pmatrix} \tag{23}$$

and the vector of dynamic coefficients

$$\boldsymbol{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{pmatrix} = \begin{pmatrix} m_1+m_2+m_3 \\ I_2+I_3+m_3 d_3^2 \\ m_3 \\ m_3 d_3 \\ f_1 \\ f_2 \\ f_3 \end{pmatrix} \in \mathbb{R}^7. \tag{24}$$

Defining $\dot{\boldsymbol{q}}_r = \dot{\boldsymbol{q}}_d + \boldsymbol{\Lambda}\boldsymbol{e} = \dot{\boldsymbol{q}}_d + \boldsymbol{\Lambda}(\boldsymbol{q}_d - \boldsymbol{q})$, with a diagonal matrix $\boldsymbol{\Lambda} > 0$, two diagonal gain matrices $\boldsymbol{K}_D > 0$ and $\boldsymbol{K}_P = \boldsymbol{K}_D\boldsymbol{\Lambda} > 0$, and a diagonal estimation gain matrix $\boldsymbol{\Gamma} > 0$, the adaptive controller will have dimension 7 and the expression

$$\begin{aligned} \boldsymbol{u} &= \hat{\boldsymbol{M}}(\boldsymbol{q})\ddot{\boldsymbol{q}}_r + \hat{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_r + \hat{\boldsymbol{g}}(\boldsymbol{q}) + \boldsymbol{K}_P\boldsymbol{e} + \boldsymbol{K}_D\dot{\boldsymbol{e}} = \boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\dot{\boldsymbol{q}}_r,\ddot{\boldsymbol{q}}_r)\hat{\boldsymbol{a}} + \boldsymbol{K}_P\boldsymbol{e} + \boldsymbol{K}_D\dot{\boldsymbol{e}} \\ \dot{\hat{\boldsymbol{a}}} &= \boldsymbol{\Gamma}\,\boldsymbol{Y}^T(\boldsymbol{q},\dot{\boldsymbol{q}},\dot{\boldsymbol{q}}_r,\ddot{\boldsymbol{q}}_r)\,(\dot{\boldsymbol{q}}_r - \dot{\boldsymbol{q}}), \qquad \hat{\boldsymbol{a}}(0) = \text{arbitrary}, \end{aligned} \tag{25}$$

where

$$\boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\dot{\boldsymbol{q}}_r,\ddot{\boldsymbol{q}}_r) =$$

$$\begin{pmatrix} \ddot{q}_{r1} & 0 & 0 & 0 & \dot{q}_{r1} & 0 & 0 \\ 0 & \ddot{q}_{r2} & q_3^2\ddot{q}_{r2} + q_3(\dot{q}_2\dot{q}_{r3} + \dot{q}_{r2}\dot{q}_3) + g_0\sin q_2 & -2q_3\ddot{q}_{r2} - (\dot{q}_2\dot{q}_{r3} + \dot{q}_{r2}\dot{q}_3) - g_0\sin q_2 & 0 & \dot{q}_{r2} & 0 \\ 0 & 0 & \ddot{q}_{r3} - q_3\,\dot{q}_2\dot{q}_{r2} - g_0\cos q_2 & \dot{q}_2\dot{q}_{r2} & 0 & 0 & \dot{q}_{r3} \end{pmatrix} \tag{26}$$

and

$$\hat{\boldsymbol{a}} = \begin{pmatrix} \hat{a}_1 & \hat{a}_2 & \hat{a}_3 & \hat{a}_4 & \hat{a}_5 & \hat{a}_6 & \hat{a}_7 \end{pmatrix}^T. \tag{27}$$

Finally, the desired trajectory is sufficiently smooth already with $\nu = 1$ and guarantees thus permanent exact tracking when the error $\boldsymbol{e}(t) = \boldsymbol{q}_d(t) - \boldsymbol{q}(t)$ asymptotically vanishes. In particular, we shall need the quantities

$$q_{di}(t) = q_{0i} + A_i \left(1 - \cos\frac{2\pi t}{T}\right), \quad \dot{q}_{di}(t) = \frac{2\pi A_i}{T}\sin\frac{2\pi t}{T}, \quad \ddot{q}_{di}(t) = \frac{4\pi^2 A_i}{T^2}\cos\frac{2\pi t}{T} \tag{28}$$

for $i = 1, 2, 3$.

Exercise 3

The dynamic equation of the pendulum is

$$\left(I_0 + md^2\right)\ddot{\theta} + mdg_0 \sin\theta = u, \tag{29}$$

with $d = L/2 = 1$ m and $I_0 = mL^2/12$, which is the inertia of a uniform thin rod of mass m and length L around an orthogonal axis passing through its center of mass. However, note that the value of I_0 (as well as that of the total inertia $I_0 + md^2$) will be irrelevant in the solution of our problem. Same for the gain k_d.

The gradient of the gravity term can be easily bounded as

$$\left\| \frac{\partial g(\theta)}{\partial \theta} \right\| = |mdg_0 \cos\theta| \leq mdg_0 = 98.1 = \alpha. \tag{30}$$

Being

$$k_p = 500 > 196.2 = 2\alpha, \tag{31}$$

the iterative scheme (4–5) will certainly converge to $e = \theta_d - \theta = 0$. Moreover, we can take out from k_p a factor $1/\beta$ in the following way

$$k_p = \frac{k'_p}{\beta} = 500 \qquad \Rightarrow \qquad k'_p = 100 > 98.1 = \alpha, \qquad 0 < \beta = \frac{1}{5} \leq \frac{1}{2}, \tag{32}$$

so that we recognize the sufficient conditions for contraction of the iterative learning control (4–5). From the proof of the related theorem and the value of β in (32), we have

$$\|e_i\| < \frac{\beta}{1-\beta}\|e_{i-1}\| \qquad \Rightarrow \qquad |\theta_d - \theta_i| = |e_i| < \frac{\frac{1}{5}}{1-\frac{1}{5}}\,|e_{i-1}| = 0.25\,|\theta_d - \theta_{i-1}|. \tag{33}$$

As a result, we know in advance that the error will be reduced *at least* by a factor 4 from one iteration to the other. Thus, starting with the known initial error

$$e_0 = \theta_d - \theta_0 = \frac{\pi}{3} = 1.0472, \tag{34}$$

we can iteratively estimate upper bounds $\hat{e}_i$ for the absolute errors $|e_i|$:

$$\begin{aligned}
|e_1| &= |\theta_d - \theta_1| < 0.25\,|e_0| = \frac{\pi}{12} = 0.2618 = \hat{e}_1, \\
|e_2| &= |\theta_d - \theta_2| < 0.25\,|e_1| = \frac{\pi}{48} = 0.0654 = \hat{e}_2, \\
|e_3| &= |\theta_d - \theta_3| < 0.25\,|e_2| = \frac{\pi}{208} = 0.0164 = \hat{e}_3, \\
|e_4| &= |\theta_d - \theta_4| < 0.25\,|e_3| = \frac{\pi}{832} = 0.0041 = \hat{e}_4 < 0.01 = \varepsilon.
\end{aligned} \tag{35}$$

We can conclude that the absolute error with respect to θ_d will be reduced and kept below the required tolerance ε starting with the iteration $i_{\min} = 4$. Note that the control scheme tolerates a large uncertainty for the bound α on the gravity term. In the present case, we could handle a link mass which is up to 250% larger than the nominal value and would still converge, though progressively slower, without modifying the chosen proportional control gain (nor anything else).

We can do an exact calculation of the solution sequence of angles θ_i and feedforward torques u_i, even without performing a dynamic simulation but just knowing in advance the value $M = mdg_0 = 98.1$. In fact, every new equilibrium configuration $\theta = \theta_i$ will have to satisfy the nonlinear equation

$$mg_0 d \sin\theta = k_p(\theta_d - \theta) + u_{i-1}, \qquad i = 1, 2, \ldots \tag{36}$$

We can solve numerically (36) using, e.g., the matlab function `fsolve`

```
theta(i) = fsolve(@(theta) M * sin(theta) - kp * (thetad - theta) - u(i - 1), theta(i - 1))
```

providing each time θ_{i-1} as initial guess. We update then the feedforward torque u_i using the recursion (5). Table 1 shows the actual convergence of the iterative control process, and also a comparison of the actual errors vs. their estimated bounds in (35). Indeed, the actual error converges to zero faster than its estimated bound.

i	θ_i	e_i	$\hat{e}_i$	u_i
0 (init)	0	1.0472	1.0472	0
1	0.8942	0.1530	0.2618	76.4904
2	1.0318	0.0154	0.0654	84.1916
3	1.0458	0.0014	0.0164	84.888
4 (stop)	1.0471	0.0001	0.0041	84.9510
true	1.0472	0	–	84.9571

Table 1: Iterative learning process for regulation at $\theta_d = \pi/3 = 1.0472$ rad with gravity torque estimation. Iterations are stopped when $\hat{e}_i \leq \varepsilon = 0.01$. Angles/errors in [rad], torques in [Nm].

* * * * *