

Luigi Villani, Giuseppe Oriolo, Bruno Siciliano

Solution Manual for

Robotics

Modelling, Planning and Control

February 6, 2009

Springer

Preface

This manual presents the solutions to all the end-of-chapter problems contained in the textbook **Robotics: Modelling, Planning and Control** (ISBN 978-1-84628-641-4, e-ISBN 978-1-84628-642-1) by Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani and Giuseppe Oriolo, Springer-Verlag, London, 2009.

Solutions to analytical problems are developed by emphasizing the crucial steps towards the solution. Some problems may be solved in different ways; the solution reported in the manual is believed to be the most straightforward. The solutions to several problems contain useful analytical developments which are complementary to the theoretical derivation in the textbook.

Solutions to programming problems are accompanied by results of computer implementations in MATLAB[®] (version 7.4) with SIMULINK[®].¹ The code (downloadable from www.springer.com/978-1-84628-641-4) is available free of charge to those adopting this volume as a text for courses.

The software is not aimed at providing a complete toolbox, but only at solving the end-of-chapter problems. Nonetheless, the code has been developed in a modular fashion which should allow direct expansion to more complex problems as well as ease of changing the problems data.

For the problems solved in MATLAB, the solution is contained in a file with `.m` extension, where the first letter is an `s`, followed by the problem number, e.g., `s4_1.m` is the file to execute for solving Problem 4.1.

The problems requiring simulation of a dynamic system have been solved in SIMULINK and the solution is contained in a file with `.mdl` extension, e.g., `s3_21.mdl` is the file to execute for solving Problem 3.21. Each problem of this kind requires the initialization of certain variables before starting the simulation. This is performed in a file where the first letter is an `i`, followed by the problem number, e.g., `i3_21.m` is the initialization file for Problem 3.21.

¹ MATLAB and SIMULINK are registered trademarks of The MathWorks, Inc.

For both MATLAB- and SIMULINK-based problems, the output plots of relevant variables are obtained by executing a file where the first letter is a **p**, followed by the problem number, e.g., `p3_21.m` is the file for plotting the output variables of Problem 3.21.

Variable initialization and plot can be activated by double clicking respectively on the upper-left block and the lower-right block in the SIMULINK block diagrams.

For problems requiring the simulation of two different systems, two files have been created where letters **a** and **b** have been used to distinguish them, e.g., `s3_22a.mdl` and `s3_22b.mdl` are the files for solving Problem 3.22 with two different algorithms; accordingly, the files for plotting output variables have been named `p3_22a.m` and `p3_22b.m`.

The above files are supplemented by other function and script files which are needed to solve the programming problems.

All the files used to solve a given problem are collected into a folder with the same label of the problem, e.g. Folder `3_21` contains the files of Problem 3.21.

Helpful comments are added to each file to describe its contents and functions. A `readme.txt` file is also provided.

Finally, the authors wish to thank Luigi Freda for his contributions to the software developed for the solution of some problems of Chapter 12.

Naples and Rome
February 2009

Luigi Villani
Giuseppe Oriolo
Bruno Siciliano

Contents

2	Kinematics	1
3	Differential Kinematics and Statics	19
4	Trajectory Planning	39
5	Actuators and Sensors	51
6	Control Architecture	59
7	Dynamics	65
8	Motion Control	79
9	Force Control	101
10	Visual servoing	111
11	Mobile Robots	127
12	Motion Planning	143

Kinematics

Solution to Problem 2.1

Composition of rotation matrices with respect to the current frame gives

$$\mathbf{R}(\phi) = \mathbf{R}_z(\varphi)\mathbf{R}_{x'}(\vartheta)\mathbf{R}_{z''}(\psi).$$

Using the expressions of elementary rotation matrices in (2.6) and (2.8):

$$\begin{aligned}\mathbf{R}_z(\varphi) &= \begin{bmatrix} c_\varphi & -s_\varphi & 0 \\ s_\varphi & c_\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ \mathbf{R}_{x'}(\vartheta) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\vartheta & -s_\vartheta \\ 0 & s_\vartheta & c_\vartheta \end{bmatrix} \\ \mathbf{R}_{z''}(\varphi) &= \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}\end{aligned}$$

and taking the products gives

$$\mathbf{R}(\phi) = \begin{bmatrix} c_\varphi c_\psi - s_\varphi c_\vartheta s_\psi & -c_\varphi s_\psi - s_\varphi c_\vartheta c_\psi & s_\varphi s_\vartheta \\ s_\varphi c_\psi + c_\varphi c_\vartheta s_\psi & -s_\varphi s_\psi + c_\varphi c_\vartheta c_\psi & -c_\varphi s_\vartheta \\ s_\vartheta s_\psi & s_\vartheta c_\psi & c_\vartheta \end{bmatrix}.$$

As for the inverse problem, given a rotation matrix

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix},$$

the set of Euler angles ZXZ is given by

$$\begin{aligned}\varphi &= \text{Atan2}(r_{13}, -r_{23}) \\ \vartheta &= \text{Atan2}\left(\sqrt{r_{31}^2 + r_{32}^2}, r_{33}\right) \\ \psi &= \text{Atan2}(r_{31}, r_{32})\end{aligned}$$

when $\vartheta \in (0, \pi)$. Otherwise, if $\vartheta \in (-\pi, 0)$ then the solution is

$$\begin{aligned}\varphi &= \text{Atan2}(-r_{13}, r_{23}) \\ \vartheta &= \text{Atan2}\left(-\sqrt{r_{31}^2 + r_{32}^2}, r_{33}\right) \\ \psi &= \text{Atan2}(-r_{31}, -r_{32}).\end{aligned}$$

Solution to Problem 2.2

In the case $s_\vartheta = 0$, the rotation matrix in (2.18) becomes

$$\mathbf{R}(\phi) = \begin{bmatrix} c_{\varphi+\psi} & -s_{\varphi+\psi} & 0 \\ s_{\varphi+\psi} & c_{\varphi+\psi} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

when $\vartheta = 0$. Otherwise, if $\vartheta = \pi$, then the matrix is

$$\mathbf{R}(\phi) = \begin{bmatrix} -c_{\varphi-\psi} & -s_{\varphi-\psi} & 0 \\ -s_{\varphi-\psi} & c_{\varphi-\psi} & 0 \\ 0 & 0 & -1 \end{bmatrix}.$$

From the elements [1, 2] and [2, 2] it is possible to compute only the sum or difference of angles φ and ψ , i.e.,

$$\varphi \pm \psi = \text{Atan2}(-r_{12}, r_{22})$$

where the positive sign holds for $\vartheta = 0$ and the negative sign holds for $\vartheta = \pi$.

Solution to Problem 2.3

In the case $c_\vartheta = 0$, the rotation matrix in (2.21) becomes

$$\mathbf{R}(\phi) = \begin{bmatrix} 0 & s_{\psi-\varphi} & c_{\psi-\varphi} \\ 0 & c_{\psi-\varphi} & -s_{\psi-\varphi} \\ -1 & 0 & 0 \end{bmatrix}$$

when $\vartheta = \pi/2$. Otherwise, if $\vartheta = -\pi/2$, then the matrix is

$$\mathbf{R}(\phi) = \begin{bmatrix} 0 & -s_{\psi+\varphi} & -c_{\psi+\varphi} \\ 0 & c_{\psi+\varphi} & -s_{\psi+\varphi} \\ 1 & 0 & 0 \end{bmatrix}.$$

From the elements [2, 2] and [2, 3] it is possible to compute only the sum or difference of angles ψ and φ , i.e.,

$$\psi \pm \varphi = \text{Atan2}(-r_{23}, r_{22})$$

where the positive sign holds for $\vartheta = -\pi/2$ and the negative sign holds for $\vartheta = \pi/2$.

Solution to Problem 2.4

The rotation matrix can be obtained as in (2.24)

$$\mathbf{R}(\vartheta, \mathbf{r}) = \mathbf{R}_z(\alpha)\mathbf{R}_y(\beta)\mathbf{R}_z(\vartheta)\mathbf{R}_y(-\beta)\mathbf{R}_z(-\alpha),$$

where the elementary rotation matrices are given as in (2.6) and (2.7):

$$\begin{aligned} \mathbf{R}_z(\alpha) &= \begin{bmatrix} c_\alpha & -s_\alpha & 0 \\ s_\alpha & c_\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ \mathbf{R}_y(\beta) &= \begin{bmatrix} c_\beta & 0 & s_\beta \\ 0 & 1 & 0 \\ -s_\beta & 0 & c_\beta \end{bmatrix} \\ \mathbf{R}_z(\vartheta) &= \begin{bmatrix} c_\vartheta & -s_\vartheta & 0 \\ s_\vartheta & c_\vartheta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \end{aligned}$$

Taking the first product gives

$$\mathbf{R}_z(\alpha)\mathbf{R}_y(\beta) = \begin{bmatrix} c_\alpha c_\beta & -s_\alpha & c_\alpha s_\beta \\ s_\alpha c_\beta & c_\alpha & s_\alpha s_\beta \\ -s_\beta & 0 & c_\beta \end{bmatrix}.$$

The next product gives

$$\mathbf{R}_z(\alpha)\mathbf{R}_y(\beta)\mathbf{R}_z(\vartheta) = \begin{bmatrix} c_\alpha c_\beta c_\vartheta - s_\alpha s_\vartheta & -c_\alpha c_\beta s_\vartheta - s_\alpha c_\vartheta & c_\alpha s_\beta \\ s_\alpha c_\beta c_\vartheta + c_\alpha s_\vartheta & -s_\alpha c_\beta s_\vartheta + c_\alpha c_\vartheta & s_\alpha s_\beta \\ -s_\beta c_\vartheta & s_\beta s_\vartheta & c_\beta \end{bmatrix}.$$

Then, by observing that

$$\mathbf{R}_y(-\beta)\mathbf{R}_z(-\alpha) = (\mathbf{R}_z(\alpha)\mathbf{R}_y(\beta))^T,$$

the overall rotation matrix is

$$\begin{aligned} \mathbf{R}(\vartheta, \mathbf{r}) &= \begin{bmatrix} (s_\alpha^2 + c_\alpha^2 c_\beta^2)c_\vartheta + c_\alpha^2 s_\beta^2 & s_\alpha c_\alpha s_\beta^2(1 - c_\vartheta) - c_\beta s_\vartheta & \\ s_\alpha c_\alpha s_\beta^2(1 - c_\vartheta) + c_\beta s_\vartheta & (s_\alpha^2 c_\beta^2 + c_\alpha^2)c_\vartheta + s_\alpha^2 s_\beta^2 & \\ c_\alpha s_\beta c_\beta(1 - c_\vartheta) - s_\alpha s_\beta s_\vartheta & s_\alpha s_\beta c_\beta(1 - c_\vartheta) + c_\alpha s_\beta s_\vartheta & \\ & c_\alpha s_\beta c_\beta(1 - c_\vartheta) + s_\alpha s_\beta s_\vartheta & \\ & s_\alpha s_\beta c_\beta(1 - c_\vartheta) - c_\alpha s_\beta s_\vartheta & \\ & s_\beta^2 c_\vartheta + c_\beta^2 & \end{bmatrix}. \end{aligned}$$

Recalling the relations

$$s_\alpha = \frac{r_y}{\sqrt{r_x^2 + r_y^2}} \quad c_\alpha = \frac{r_x}{\sqrt{r_x^2 + r_y^2}}$$

$$s_\beta = \sqrt{r_x^2 + r_y^2} \quad c_\beta = r_z$$

$$r_x^2 + r_y^2 + r_z^2 = 1$$

and using the following identities:

$$s_\alpha^2 + c_\alpha^2 c_\beta^2 = 1 - r_x^2$$

$$s_\alpha^2 c_\beta^2 + c_\alpha^2 = 1 - r_y^2$$

leads to expression (2.25).

Solution to Problem 2.5

From the expression of $\mathbf{R}(\vartheta, \mathbf{r})$ in (2.25), summing the elements on the diagonal gives

$$r_{11} + r_{22} + r_{33} = 1 + 2 \cos \vartheta$$

from which the angle is

$$\vartheta = \cos^{-1} \left(\frac{r_{11} + r_{22} + r_{33} - 1}{2} \right).$$

Then, taking suitable differences between the off-diagonal elements gives

$$r_{32} - r_{23} = 2r_x \sin \vartheta$$

$$r_{13} - r_{31} = 2r_y \sin \vartheta$$

$$r_{21} - r_{12} = 2r_z \sin \vartheta$$

and thus

$$\mathbf{r} = \frac{1}{2 \sin \vartheta} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix}.$$

In the case $\sin \vartheta = 0$, if $r_{11} + r_{22} + r_{33} = 3$, then $\vartheta = 0$; this means that no rotation has occurred and \mathbf{r} is arbitrary. Instead, if $r_{11} + r_{22} + r_{33} = -1$, then $\vartheta = \pi$ and the expression of the rotation matrix becomes

$$\mathbf{R}(\pi, \mathbf{r}) = \begin{bmatrix} 2r_x^2 - 1 & 2r_x r_y & 2r_x r_z \\ 2r_x r_y & 2r_y^2 - 1 & 2r_y r_z \\ 2r_x r_z & 2r_y r_z & 2r_z^2 - 1 \end{bmatrix}.$$

The three components of the unit vector can be computed by taking any row or column; for instance, from the first column it is

$$\begin{aligned} r_x &= \pm \sqrt{\frac{r_{11} + 1}{2}} \\ r_y &= \frac{r_{12}}{2r_x} \\ r_z &= \frac{r_{13}}{2r_x}. \end{aligned}$$

However, if $r_x \approx 0$, then the computation of r_y and r_z is ill-conditioned. In that case, it is better to use another column to compute either r_y or r_z , and so forth.

Solution to Problem 2.6

With reference to (2.25), the quantities c_ϑ , $r_i s_\vartheta$ and $r_i r_j (1 - c_\vartheta)$ for $i, j = x, y, z$ can be respectively expressed as

$$\begin{aligned} c_\vartheta &= 2\cos^2(\vartheta/2) - 1 \\ &= 2\eta^2 - 1 \\ r_i s_\vartheta &= 2r_i \sin(\vartheta/2)\cos(\vartheta/2) \\ &= 2\eta\epsilon_i \\ r_i r_j (1 - c_\vartheta) &= 2r_i r_j \sin^2(\vartheta/2) \\ &= 2\epsilon_i \epsilon_j \end{aligned}$$

where (2.30) and (2.31) have been used. Hence, (2.33) follows.

Solution to Problem 2.7

Start observing that

$$\mathbf{R}(\vartheta, \mathbf{r})\mathbf{r} = \mathbf{R}^T(\vartheta, \mathbf{r})\mathbf{r} = \mathbf{r}$$

since \mathbf{r} is the axis of the rotation described by \mathbf{R} . Then, since $\boldsymbol{\epsilon}$ and \mathbf{r} are aligned, the result

$$\mathbf{R}(\eta, \boldsymbol{\epsilon})\boldsymbol{\epsilon} = \mathbf{R}^T(\eta, \boldsymbol{\epsilon})\boldsymbol{\epsilon} = \boldsymbol{\epsilon}$$

follows directly.

Solution to Problem 2.8

By taking the expressions of the diagonal elements of the matrix in (2.33), the following equality holds:

$$\frac{1}{2}\sqrt{r_{11} + r_{22} + r_{33} + 1} = 6\eta^2 + 2(\epsilon_x^2 + \epsilon_y^2 + \epsilon_z^2) - 2,$$

and thus using the constraint in (2.32) gives (2.34).

By taking the expressions of the elements $[2, 3]$ and $[3, 2]$ of the matrix in (2.33), together with the diagonal elements, the following equality holds:

$$\frac{1}{2}\text{sgn}(r_{32} - r_{23})\sqrt{r_{11} - r_{22} - r_{33} + 1} = \frac{1}{2}\text{sgn}(4\eta\epsilon_x)\sqrt{2(\epsilon_x^2 - \epsilon_y^2 - \epsilon_z^2 - \eta^2 + 1)},$$

and thus using again the constraint in (2.32) gives

$$\frac{1}{2}\text{sgn}(r_{32} - r_{23})\sqrt{r_{11} - r_{22} - r_{33} + 1} = \text{sgn}(4\eta\epsilon_x)|\epsilon_x| = \epsilon_x$$

where the assumption $\eta \geq 0$ has been exploited. A similar argument can be worked out to derive the expressions of ϵ_y and ϵ_z in (2.35).

Solution to Problem 2.9

Using the expression of the rotation matrix as a function of the unit quaternion in (2.33), the product $\mathbf{R}_1\mathbf{R}_2$ gives rise to a rotation matrix which is a function of $\{\eta_1, \epsilon_1\}$ and $\{\eta_2, \epsilon_2\}$. The diagonal elements of such matrix are

$$\begin{aligned} r_{11} &= (2(\eta_1^2 + \epsilon_{1x}^2) - 1)(2(\eta_2^2 + \epsilon_{2x}^2) - 1) \\ &\quad + 4(\epsilon_{1x}\epsilon_{1y} - \eta_1\epsilon_{1z})(\epsilon_{2x}\epsilon_{2y} + \eta_2\epsilon_{2z}) + 4(\epsilon_{1x}\epsilon_{1z} + \eta_1\epsilon_{1y})(\epsilon_{2x}\epsilon_{2z} - \eta_2\epsilon_{2y}) \\ r_{22} &= (2(\eta_1^2 + \epsilon_{1y}^2) - 1)(2(\eta_2^2 + \epsilon_{2y}^2) - 1) \\ &\quad + 4(\epsilon_{1y}\epsilon_{1z} - \eta_1\epsilon_{1x})(\epsilon_{2y}\epsilon_{2z} + \eta_2\epsilon_{2x}) + 4(\epsilon_{1x}\epsilon_{1y} + \eta_1\epsilon_{1z})(\epsilon_{2x}\epsilon_{2y} - \eta_2\epsilon_{2z}) \\ r_{33} &= (2(\eta_1^2 + \epsilon_{1z}^2) - 1)(2(\eta_2^2 + \epsilon_{2z}^2) - 1) \\ &\quad + 4(\epsilon_{1x}\epsilon_{1z} - \eta_1\epsilon_{1y})(\epsilon_{2x}\epsilon_{2z} + \eta_2\epsilon_{2y}) + 4(\epsilon_{1y}\epsilon_{1z} + \eta_1\epsilon_{1x})(\epsilon_{2y}\epsilon_{2z} - \eta_2\epsilon_{2x}). \end{aligned}$$

Then, compose these terms as follows:

$$\begin{aligned} r_{11} + r_{22} + r_{33} + 1 &= 4(\epsilon_{1x}^2\epsilon_{2x}^2 + \epsilon_{1y}^2\epsilon_{2y}^2 + \epsilon_{1z}^2\epsilon_{2z}^2 \\ &\quad + 2\epsilon_{1x}\epsilon_{2x}\epsilon_{1y}\epsilon_{2y} + 2\epsilon_{1x}\epsilon_{2x}\epsilon_{1z}\epsilon_{2z} + 2\epsilon_{1y}\epsilon_{2y}\epsilon_{1z}\epsilon_{2z}) \\ &\quad - 8\eta_1\eta_2(\epsilon_{1x}\epsilon_{2x} + \epsilon_{1y}\epsilon_{2y} + \epsilon_{1z}\epsilon_{2z}) \\ &\quad + (4\eta_1^2 - 2)(\epsilon_{2x}^2 + \epsilon_{2y}^2 + \epsilon_{2z}^2) + (4\eta_2^2 - 2)(\epsilon_{1x}^2 + \epsilon_{1y}^2 + \epsilon_{1z}^2) \\ &\quad + 12\eta_1^2\eta_2^2 - 6\eta_1^2 - 6\eta_2^2 + 4. \end{aligned}$$

Using the constraint in (2.32) yields

$$\frac{1}{4}(r_{11} + r_{22} + r_{33} + 1) = (\eta_1\eta_2 - \epsilon_1^T\epsilon_2)^2$$

which, in view of (2.34), coincides with the square of the scalar part of the quaternion product in (2.37).

A similar argument can be pursued to show the equivalence between the vector part of the quaternion that can be extracted from the product $\mathbf{R}_1\mathbf{R}_2$ using (2.33) and the vector part of the quaternion product in (2.37).

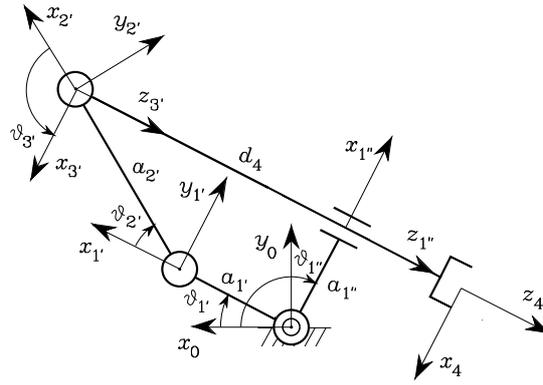


Fig. S2.1. Four-link closed-chain planar arm with frame assignment

Solution to Problem 2.10

The matrix

$$A_1^0 = \begin{bmatrix} R_1^0 & o_1^0 \\ \mathbf{0}^T & 1 \end{bmatrix}$$

can be inverted as a block-partitioned matrix. In fact, by recalling that

$$\begin{bmatrix} A & D \\ O & B \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} & -A^{-1}DB^{-1} \\ O & B^{-1} \end{bmatrix},$$

expression (2.45) follows by observing that $(R_1^0)^{-1} = R_0^1$.

Solution to Problem 2.11

Joint 4 was selected as the cut joint. The link frames can be assigned as in Fig. S2.1. With this choice, the Denavit-Hartenberg parameters are specified in Table S2.1.

Table S2.1. DH parameters for the four-link closed-chain planar arm

Link	a_i	α_i	d_i	ϑ_i
1'	$a_{1'}$	0	0	$\vartheta_{1'}$
2'	$a_{2'}$	0	0	$\vartheta_{2'}$
3'	0	$\pi/2$	0	$\vartheta_{3'}$
1''	$a_{1''}$	$-\pi/2$	0	$\vartheta_{1''}$
4	0	0	d_4	0

Notice that the parameters for Link 4 are all constant. For the first two revolute joints, the homogeneous transformation matrix defined in (2.52) has the

same structure as in (2.62), while for the third revolute joint, the homogeneous transformation matrix is

$$\mathbf{A}_{3'}^{2'}(\vartheta_{3'}) = \begin{bmatrix} c_{3'} & 0 & s_{3'} & 0 \\ s_{3'} & 0 & -c_{3'} & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Therefore, the coordinate transformations for the two branches of the tree are respectively:

$$\mathbf{A}_{3'}^0(\mathbf{q}') = \mathbf{A}_{1'}^0 \mathbf{A}_{2'}^{1'} \mathbf{A}_{3'}^{2'} = \begin{bmatrix} c_{1'2'3'} & 0 & s_{1'2'3'} & a_{1'}c_{1'} + a_{2'}c_{1'2'} \\ s_{1'2'3'} & 0 & -c_{1'2'3'} & a_{1'}s_{1'} + a_{2'}s_{1'2'} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where $\mathbf{q}' = [\vartheta_{1'} \quad \vartheta_{2'} \quad \vartheta_{3'}]^T$, and

$$\mathbf{A}_{1''}^0(q'') = \begin{bmatrix} c_{1''} & 0 & -s_{1''} & a_{1''}c_{1''} \\ s_{1''} & 0 & c_{1''} & a_{1''}s_{1''} \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where $q'' = \vartheta_{1''}$. To complete, the constant homogeneous transformation for the last link is

$$\mathbf{A}_4^{3'} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

With reference to (2.60), the orientation constraints are ($\vartheta_{3'1''} = \pi$)

$$\begin{aligned} z_{3'}^0(\mathbf{q}') &= z_{1''}^0(q'') \\ \mathbf{x}_{3'}^{0T}(\mathbf{q}') \mathbf{x}_{1''}^0(q'') &= -1 \end{aligned}$$

which give

$$\begin{aligned} s_{1'2'3'} &= -s_{1''} \\ c_{1'2'3'} &= -c_{1''} \end{aligned}$$

and thus

$$\vartheta_{2'} + \vartheta_{3'} = \pi - \vartheta_{1'} + \vartheta_{1''}. \quad (\text{S2.1})$$

On the other hand, the position constraints are

$$\begin{bmatrix} \mathbf{x}_{3'}^{0T}(\mathbf{q}') \\ \mathbf{y}_{3'}^{0T}(\mathbf{q}') \end{bmatrix} (\mathbf{p}_{3'}^0(\mathbf{q}') - \mathbf{p}_{1''}^0(q'')) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

which give

$$a_{1'} \cos(\vartheta_{2'} + \vartheta_{3'}) + a_{2'} \cos \vartheta_{3'} - a_{1''} \cos(\vartheta_{1'} + \vartheta_{2'} + \vartheta_{3'} - \vartheta_{1''}) = 0$$

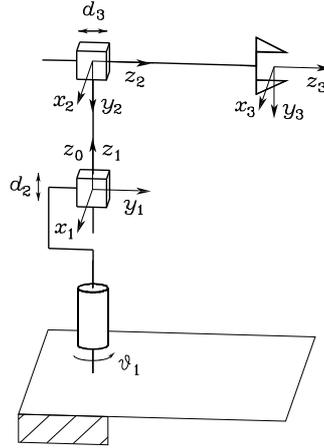


Fig. S2.2. Cylindrical arm with frame assignment

Table S2.2. DH parameters for the cylindrical arm

Link	a_i	α_i	d_i	ϑ_i
1	0	0	0	ϑ_1
2	0	$-\pi/2$	d_2	0
3	0	0	d_3	0

and, in view of (S2.1), it is

$$\cos \vartheta_{3'} = \frac{-a_{1''} + a_{1'} \cos(\vartheta_{1''} - \vartheta_{1'})}{a_{2'}}. \quad (\text{S2.2})$$

The solution to (S2.2) exists for any $\vartheta_{1'}$ and $\vartheta_{1''}$ provided that

$$a_{1'} + a_{1''} \leq a_{2'}.$$

Further, it is

$$s_{3'} = \pm \sqrt{1 - c_{3'}^2}$$

with $c_{3'}$ as in (S2.2). Therefore, it is

$$\begin{aligned} \vartheta_{3'} &= \text{Atan2}(s_{3'}, c_{3'}) \\ \vartheta_{2'} &= \pi - \vartheta_{1'} + \vartheta_{1''} - \vartheta_{3'}. \end{aligned}$$

It follows that the vector of joint variables is $\mathbf{q} = [\vartheta_{1'} \quad \vartheta_{1''}]^T$. These joints are natural candidates to be the actuated joints. The arm direct kinematics can be computed as $\mathbf{T}_4^0(\mathbf{q}) = \mathbf{A}_{3'}^0(\mathbf{q})\mathbf{A}_4^{3'}$ where the expressions of $\vartheta_{2'}$ and $\vartheta_{3'}$ have to be substituted into the homogeneous transformation $\mathbf{A}_{3'}^0$.

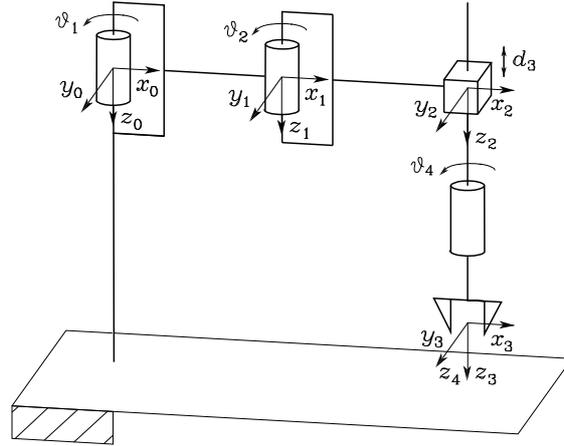


Fig. S2.3. SCARA manipulator with frame assignment

Solution to Problem 2.12

The link frames can be assigned as in Fig. S2.2. With this choice, the Denavit-Hartenberg parameters are specified in Table S2.2.

The homogeneous transformation matrices (2.52) for the three joints are:

$$\mathbf{A}_1^0(\vartheta_1) = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{A}_2^1(d_2) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{A}_3^2(d_3) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

and thus the arm direct kinematics is

$$\mathbf{T}_3^0(\mathbf{q}) = \mathbf{A}_1^0 \mathbf{A}_2^1 \mathbf{A}_3^2 = \begin{bmatrix} c_1 & 0 & -s_1 & -d_3 s_1 \\ s_1 & 0 & c_1 & d_3 c_1 \\ 0 & -1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where $\mathbf{q} = [\vartheta_1 \quad d_2 \quad d_3]^T$.

Solution to Problem 2.13

The link frames can be assigned as in Fig. S2.3. Since the typical approach to an object is from the top, it is reasonable to choose all the joint axes pointing downwards. With this choice, the Denavit-Hartenberg parameters are specified in Table S2.3.

Table S2.3. DH parameters for the SCARA manipulator

Link	a_i	α_i	d_i	ϑ_i
1	a_1	0	0	ϑ_1
2	a_2	0	0	ϑ_2
3	0	0	d_3	0
4	0	0	0	ϑ_4

The homogeneous transformation matrices (2.52) for the four joints are:

$$\mathbf{A}_i^{i-1}(\vartheta_i) = \begin{bmatrix} c_i & -s_i & 0 & a_i c_i \\ s_i & c_i & 0 & a_i s_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad i = 1, 2$$

$$\mathbf{A}_3^2(d_3) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{A}_4^3(\vartheta_4) = \begin{bmatrix} c_4 & -s_4 & 0 & 0 \\ s_4 & c_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

and thus the manipulator direct kinematics is

$$\mathbf{T}_4^0(\mathbf{q}) = \mathbf{A}_1^0 \mathbf{A}_2^1 \mathbf{A}_3^2 \mathbf{A}_4^3 = \begin{bmatrix} c_{124} & -s_{124} & 0 & a_1 c_1 + a_2 c_{12} \\ s_{124} & c_{124} & 0 & a_1 s_1 + a_2 s_{12} \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{S2.3})$$

where $\mathbf{q} = [\vartheta_1 \ \vartheta_2 \ d_3 \ \vartheta_4]^T$. It is worth noticing that the direct kinematics of this structure can be conceptually derived from that of a three-link planar arm in which $a_3 = 0$, ϑ_3 is replaced by ϑ_4 , and the z coordinate is d_3 .

Solution to Problem 2.14

The torso can be modelled as an anthropomorphic arm, corresponding to the first three DOFs. Therefore \mathbf{T}_3^0 in (2.78) and (2.79) has the expression (2.66).

The constant matrices \mathbf{T}_l^t and \mathbf{T}_r^t can be computed in terms of the angle β and of the lengths d_l and d_r of segments $O_t O_l$ and $O_t O_r$, being O_t , O_l and O_r the origins of frames t , l and r , respectively. In detail:

$$\mathbf{T}_l^t = \begin{bmatrix} \mathbf{R}_l^t & \mathbf{o}_{t,l}^t \\ \mathbf{0}^T & 1 \end{bmatrix} \quad \mathbf{T}_r^t = \begin{bmatrix} \mathbf{R}_r^t & \mathbf{o}_{t,r}^t \\ \mathbf{0}^T & 1 \end{bmatrix},$$

where

$$\mathbf{o}_{t,l}^t = \begin{bmatrix} d_l c_\beta \\ 0 \\ d_l s_\beta \end{bmatrix} \quad \mathbf{o}_{t,r}^t = \begin{bmatrix} d_r c_\beta \\ 0 \\ -d_r s_\beta \end{bmatrix}$$

are the positions of the origins of frames l and r with respect to frame t and

$$\mathbf{R}_l^t = \begin{bmatrix} 0 & -s_\beta & c_\beta \\ 1 & 0 & 0 \\ 0 & c_\beta & s_\beta \end{bmatrix} \quad \mathbf{R}_r^t = \begin{bmatrix} 0 & s_\beta & c_\beta \\ 1 & 0 & 0 \\ 0 & c_\beta & -s_\beta \end{bmatrix}$$

are the corresponding rotation matrices.

Finally, matrices $\mathbf{T}_{rh}^r = \mathbf{T}_{lh}^l$ can be computed using (2.72) with (2.77) in place of (2.73).

Solution to Problem 2.15

Consider two sets of Euler angles ZYZ:

$$\phi_1 = \begin{bmatrix} \pi/2 \\ \pi/2 \\ 0 \end{bmatrix} \quad \phi_2 = \begin{bmatrix} -\pi/2 \\ -\pi/2 \\ 0 \end{bmatrix}.$$

The corresponding rotation matrices are

$$\mathbf{R}(\phi_1) = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{bmatrix} \quad \mathbf{R}(\phi_2) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

Composition of rotations with respect to the current frame gives

$$\mathbf{R}(\phi_1)\mathbf{R}(\phi_2) = \begin{bmatrix} 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}$$

to which the following two sets of Euler angles correspond as for (2.19) and (2.20):

$$\phi_a = \begin{bmatrix} \pi \\ \pi/2 \\ -\pi/2 \end{bmatrix} \quad \phi_b = \begin{bmatrix} 0 \\ -\pi/2 \\ \pi/2 \end{bmatrix}.$$

On the other hand, composition of rotations with respect to the fixed frame gives

$$\mathbf{R}(\phi_2)\mathbf{R}(\phi_1) = \begin{bmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}$$

to which the following two sets of Euler angles correspond as for (2.19) and (2.20):

$$\phi_a = \begin{bmatrix} 0 \\ \pi/2 \\ -\pi/2 \end{bmatrix} \quad \phi_b = \begin{bmatrix} \pi \\ -\pi/2 \\ \pi/2 \end{bmatrix}.$$

It is evident that the two results differ, and then it is not possible to commute the order of rotations. Notice, also, that the rotation matrix resulting from direct sum of the two given sets of angles is

$$\mathbf{R}(\phi_1 + \phi_2) = \mathbf{I}$$

to which the set of Euler angles $\phi = [0 \ 0 \ 0]^T$ corresponds.

Solution to Problem 2.16

In view of the approximations $\cos(d\phi) \approx 1$ and $\sin(d\phi) \approx d\phi$, the elementary rotation matrices for infinitesimal angles can be written as:

$$\begin{aligned} \mathbf{R}_x(d\phi_x) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(d\phi_x) & -\sin(d\phi_x) \\ 0 & \sin(d\phi_x) & \cos(d\phi_x) \end{bmatrix} \approx \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -d\phi_x \\ 0 & d\phi_x & 1 \end{bmatrix} \\ \mathbf{R}_y(d\phi_y) &= \begin{bmatrix} \cos(d\phi_y) & 0 & \sin(d\phi_y) \\ 0 & 1 & 0 \\ -\sin(d\phi_y) & 0 & \cos(d\phi_y) \end{bmatrix} \approx \begin{bmatrix} 1 & 0 & d\phi_y \\ 0 & 1 & 0 \\ -d\phi_y & 0 & 1 \end{bmatrix} \\ \mathbf{R}_z(d\phi_z) &= \begin{bmatrix} \cos(d\phi_z) & -\sin(d\phi_z) & 0 \\ \sin(d\phi_z) & \cos(d\phi_z) & 0 \\ 0 & 0 & 1 \end{bmatrix} \approx \begin{bmatrix} 1 & -d\phi_z & 0 \\ d\phi_z & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \end{aligned}$$

Multiplying the first two matrices gives

$$\mathbf{R}_x(d\phi_x)\mathbf{R}_y(d\phi_y) \approx \begin{bmatrix} 1 & 0 & d\phi_y \\ 0 & 1 & -d\phi_x \\ -d\phi_y & d\phi_x & 1 \end{bmatrix}$$

where higher order terms have been neglected. Reversing the order of multiplication gives

$$\mathbf{R}_y(d\phi_y)\mathbf{R}_x(d\phi_x) \approx \begin{bmatrix} 1 & 0 & d\phi_y \\ 0 & 1 & -d\phi_x \\ -d\phi_y & d\phi_x & 1 \end{bmatrix},$$

which shows that the rotation resulting from any two elementary rotations is independent of the order of rotations when the angles of rotation are infinitesimal.

Constructing the matrix of the three elementary rotations about coordinate axes for infinitesimal angles gives

$$\mathbf{R}(d\phi_x, d\phi_y, d\phi_z) = \begin{bmatrix} 1 & -d\phi_z & d\phi_y \\ d\phi_z & 1 & -d\phi_x \\ -d\phi_y & d\phi_x & 1 \end{bmatrix}.$$

For another set of infinitesimal angles, the rotation matrix can be formally written as

$$\mathbf{R}(d\phi'_x, d\phi'_y, d\phi'_z) = \begin{bmatrix} 1 & -d\phi'_z & d\phi'_y \\ d\phi'_z & 1 & -d\phi'_x \\ -d\phi'_y & d\phi'_x & 1 \end{bmatrix}.$$

Multiplying these two rotation matrices and neglecting higher order terms gives

$$\begin{aligned} \mathbf{R}(d\phi_x, d\phi_y, d\phi_z)\mathbf{R}(d\phi'_x, d\phi'_y, d\phi'_z) &= \\ &= \begin{bmatrix} 1 & -(d\phi_z + d\phi'_z) & d\phi_y + d\phi'_y \\ d\phi_z + d\phi'_z & 1 & -(d\phi_x + d\phi'_x) \\ -(d\phi_y + d\phi'_y) & d\phi_x + d\phi'_x & 1 \end{bmatrix} \\ &= \mathbf{R}(d\phi_x + d\phi'_x, d\phi_y + d\phi'_y, d\phi_z + d\phi'_z). \end{aligned}$$

Solution to Problem 2.17

In order to find the arm reachable workspace, it is worth considering all the joint configurations giving a loss of mobility. These are the $2^3 = 8$ configurations for the joint limits:

$$\begin{aligned} \mathbf{q}_A &= \begin{bmatrix} -\pi/3 \\ -2\pi/3 \\ -\pi/2 \end{bmatrix} & \mathbf{q}_B &= \begin{bmatrix} -\pi/3 \\ -2\pi/3 \\ \pi/2 \end{bmatrix} & \mathbf{q}_C &= \begin{bmatrix} -\pi/3 \\ 2\pi/3 \\ -\pi/2 \end{bmatrix} & \mathbf{q}_D &= \begin{bmatrix} -\pi/3 \\ 2\pi/3 \\ \pi/2 \end{bmatrix} \\ \mathbf{q}_E &= \begin{bmatrix} \pi/3 \\ -2\pi/3 \\ -\pi/2 \end{bmatrix} & \mathbf{q}_F &= \begin{bmatrix} \pi/3 \\ -2\pi/3 \\ \pi/2 \end{bmatrix} & \mathbf{q}_G &= \begin{bmatrix} \pi/3 \\ 2\pi/3 \\ -\pi/2 \end{bmatrix} & \mathbf{q}_H &= \begin{bmatrix} \pi/3 \\ 2\pi/3 \\ \pi/2 \end{bmatrix}, \end{aligned}$$

the 4 configurations for $\vartheta_2 = 0$:

$$\mathbf{q}_I = \begin{bmatrix} -\pi/3 \\ 0 \\ -\pi/2 \end{bmatrix} \quad \mathbf{q}_J = \begin{bmatrix} -\pi/3 \\ 0 \\ \pi/2 \end{bmatrix} \quad \mathbf{q}_K = \begin{bmatrix} \pi/3 \\ 0 \\ -\pi/2 \end{bmatrix} \quad \mathbf{q}_L = \begin{bmatrix} \pi/3 \\ 0 \\ \pi/2 \end{bmatrix},$$

the 4 configurations for $\vartheta_3 = 0$:

$$\mathbf{q}_M = \begin{bmatrix} -\pi/3 \\ -2\pi/3 \\ 0 \end{bmatrix} \quad \mathbf{q}_N = \begin{bmatrix} -\pi/3 \\ 2\pi/3 \\ 0 \end{bmatrix} \quad \mathbf{q}_O = \begin{bmatrix} \pi/3 \\ -2\pi/3 \\ 0 \end{bmatrix} \quad \mathbf{q}_P = \begin{bmatrix} \pi/3 \\ 2\pi/3 \\ 0 \end{bmatrix},$$

and the 2 configurations for both $\vartheta_2 = 0$ and $\vartheta_3 = 0$:

$$\mathbf{q}_Q = \begin{bmatrix} -\pi/3 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{q}_R = \begin{bmatrix} \pi/3 \\ 0 \\ 0 \end{bmatrix}.$$

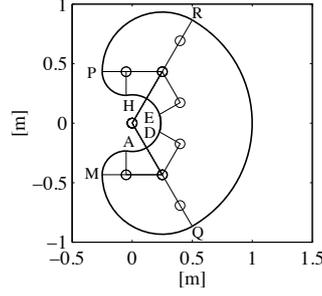


Fig. S2.4. Reachable workspace for a three-link planar arm

Starting from point A in Cartesian space corresponding to \mathbf{q}_A , it is necessary to draw all the arcs connecting such point to all other points corresponding to configurations that differ only for one component, i.e., points B, C, E, I, M . Repeating the procedure for each point leads to obtaining all the portions of the requested workspace. It can be recognized that the external contour of the area $AMQRPHEDA$ delimits the reachable workspace, as illustrated in Fig. S2.4.

Solution to Problem 2.18

When $s_3 = 0$, the two cases $c_3 = 1$ and $c_3 = -1$ have to be considered.

In the case $c_3 = 1$, the arm is outstretched with $\vartheta_3 = 0$. From (2.105)–(2.108), it is

$$\vartheta_{2,I} = \vartheta_{2,III} = \text{Atan2} \left(p_{Wz}, \sqrt{p_{Wx}^2 + p_{Wy}^2} \right) \quad (\text{S2.4})$$

$$\vartheta_{2,II} = \vartheta_{2,IV} = \text{Atan2} \left(p_{Wz}, -\sqrt{p_{Wx}^2 + p_{Wy}^2} \right). \quad (\text{S2.5})$$

Hence, if $\sqrt{p_{Wx}^2 + p_{Wy}^2} > 0$, two different solutions exist for ϑ_2 . The corresponding values of ϑ_1 are those in (2.109) or (2.110). On the other hand, if $\sqrt{p_{Wx}^2 + p_{Wy}^2} = 0$, i.e., $p_{Wx} = p_{Wy} = 0$, the admissibility of the solution requires that $p_{Wz} = \pm(a_2 + a_3)$. Hence, from (S2.4) and (S2.5) it is $\vartheta_2 = \pm\pi/2$. Moreover, in view of (2.109) and (2.110), an infinity of solutions exists for ϑ_1 .

In the case $c_3 = -1$, the arm is retracted with $\vartheta_3 = \pm\pi$. From (2.105)–(2.108), it is

$$\vartheta_{2,I} = \vartheta_{2,III} = \text{Atan2} \left((a_2 - a_3)p_{Wz}, (a_2 - a_3)\sqrt{p_{Wx}^2 + p_{Wy}^2} \right) \quad (\text{S2.6})$$

$$\vartheta_{2,II} = \vartheta_{2,IV} = \text{Atan2} \left((a_2 - a_3)p_{Wz}, -(a_2 - a_3)\sqrt{p_{Wx}^2 + p_{Wy}^2} \right). \quad (\text{S2.7})$$

Hence, assuming that $a_2 \neq a_3$, if $\sqrt{p_{Wx}^2 + p_{Wy}^2} > 0$, two different solutions exist for ϑ_2 . The corresponding values of ϑ_1 are those in (2.109) or (2.110). On

the other hand, if $\sqrt{p_{W_x}^2 + p_{W_y}^2} = 0$, i.e., $p_{W_x} = p_{W_y} = 0$, the admissibility of the solution requires that $p_{W_z} = \pm|a_2 - a_3|$. Hence, from (S2.6) and (S2.7), it is $\vartheta_2 = \pm\pi/2$. As in previous case, in view of (2.109), (2.110), an infinity of solutions exists for ϑ_1 . Notice that, if $a_2 = a_3$, it is $p_{W_x} = p_{W_y} = p_{W_z} = 0$ necessarily. Therefore, an infinity of solutions exists both for ϑ_1 and ϑ_2 .

Solution to Problem 2.19

From the direct kinematics as in Problem 2.12, the end-effector position is given by

$$\begin{aligned} p_x &= -d_3 s_1 \\ p_y &= d_3 c_1 \\ p_z &= d_2. \end{aligned}$$

The first joint variable can be computed from the first two equations as

$$\vartheta_1 = \text{Atan2}(-p_x, p_y).$$

Then, the third joint variable can be computed by squaring and summing those same two equations leading to

$$d_3 = \sqrt{p_x^2 + p_y^2}.$$

Finally, the second joint variables is

$$d_2 = p_z.$$

Solution to Problem 2.20

From the direct kinematics as in Problem 2.13, the end-effector position is given by

$$\begin{aligned} p_x &= a_1 c_1 + a_2 c_{12} \\ p_y &= a_1 s_1 + a_2 s_{12} \\ p_z &= d_3. \end{aligned}$$

The first two equations are the same as (2.91) and (2.92) for a three-link planar arm. Then it is

$$c_2 = \frac{p_x^2 + p_y^2 - a_1^2 - a_2^2}{2a_1 a_2}$$

with $-1 \leq c_2 \leq 1$ and

$$s_2 = \pm\sqrt{1 - c_2^2}$$

where the positive sign corresponds to $\vartheta_2 \in (0, \pi)$ and the negative sign corresponds to $\vartheta_2 \in (-\pi, 0)$. The second joint variable is

$$\vartheta_2 = \text{Atan2}(s_2, c_2).$$

The first joint variable can be computed from

$$s_1 = \frac{(a_1 + a_2 c_2)p_y - a_2 s_2 p_x}{p_x^2 + p_y^2}$$

$$c_1 = \frac{(a_1 + a_2 c_2)p_x + a_2 s_2 p_y}{p_x^2 + p_y^2}$$

and thus

$$\vartheta_1 = \text{Atan2}(s_1, c_1).$$

The end-effector orientation can be expressed as

$$\phi = \vartheta_1 + \vartheta_2 + \vartheta_4,$$

from which the fourth joint variable is

$$\vartheta_4 = \phi - \vartheta_1 - \vartheta_2.$$

Finally, the third joint variable is

$$d_3 = p_z.$$

Differential Kinematics and Statics

Solution to Problem 3.1

Let

$$\mathbf{R} = [\mathbf{x} \quad \mathbf{y} \quad \mathbf{z}]^T$$

where the unit vector triplet $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ forms a right-handed frame. Then, the product $\mathbf{RS}(\boldsymbol{\omega})\mathbf{R}^T$ can be written as

$$\begin{aligned} \mathbf{RS}(\boldsymbol{\omega})\mathbf{R}^T &= \begin{bmatrix} \mathbf{x}^T \\ \mathbf{y}^T \\ \mathbf{z}^T \end{bmatrix} \mathbf{S}(\boldsymbol{\omega}) [\mathbf{x} \quad \mathbf{y} \quad \mathbf{z}] \\ &= \begin{bmatrix} \mathbf{x}^T \mathbf{S}(\boldsymbol{\omega}) \mathbf{x} & \mathbf{x}^T \mathbf{S}(\boldsymbol{\omega}) \mathbf{y} & \mathbf{x}^T \mathbf{S}(\boldsymbol{\omega}) \mathbf{z} \\ \mathbf{y}^T \mathbf{S}(\boldsymbol{\omega}) \mathbf{x} & \mathbf{y}^T \mathbf{S}(\boldsymbol{\omega}) \mathbf{y} & \mathbf{y}^T \mathbf{S}(\boldsymbol{\omega}) \mathbf{z} \\ \mathbf{z}^T \mathbf{S}(\boldsymbol{\omega}) \mathbf{x} & \mathbf{z}^T \mathbf{S}(\boldsymbol{\omega}) \mathbf{y} & \mathbf{z}^T \mathbf{S}(\boldsymbol{\omega}) \mathbf{z} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{x}^T (\boldsymbol{\omega} \times \mathbf{x}) & \mathbf{x}^T (\boldsymbol{\omega} \times \mathbf{y}) & \mathbf{x}^T (\boldsymbol{\omega} \times \mathbf{z}) \\ \mathbf{y}^T (\boldsymbol{\omega} \times \mathbf{x}) & \mathbf{y}^T (\boldsymbol{\omega} \times \mathbf{y}) & \mathbf{y}^T (\boldsymbol{\omega} \times \mathbf{z}) \\ \mathbf{z}^T (\boldsymbol{\omega} \times \mathbf{x}) & \mathbf{z}^T (\boldsymbol{\omega} \times \mathbf{y}) & \mathbf{z}^T (\boldsymbol{\omega} \times \mathbf{z}) \end{bmatrix}. \end{aligned}$$

In view of the properties of scalar triple products, this matrix is skew-symmetric and then

$$\mathbf{RS}(\boldsymbol{\omega})\mathbf{R}^T = \mathbf{S}(\boldsymbol{\omega}') = \begin{bmatrix} 0 & -\omega'_z & \omega'_y \\ \omega'_z & 0 & -\omega'_x \\ -\omega'_y & \omega'_x & 0 \end{bmatrix} \quad (\text{S3.1})$$

where the elements of the previous matrix can be written as

$$\begin{aligned} \omega'_x &= \mathbf{z}^T (\boldsymbol{\omega} \times \mathbf{y}) = \boldsymbol{\omega}^T (\mathbf{y} \times \mathbf{z}) = \boldsymbol{\omega}^T \mathbf{x} \\ \omega'_y &= \mathbf{x}^T (\boldsymbol{\omega} \times \mathbf{z}) = \boldsymbol{\omega}^T (\mathbf{z} \times \mathbf{x}) = \boldsymbol{\omega}^T \mathbf{y} \\ \omega'_z &= \mathbf{y}^T (\boldsymbol{\omega} \times \mathbf{x}) = \boldsymbol{\omega}^T (\mathbf{x} \times \mathbf{y}) = \boldsymbol{\omega}^T \mathbf{z}. \end{aligned}$$

Hence, it is

$$\begin{aligned}\boldsymbol{\omega}' &= [\omega'_x \ \omega'_y \ \omega'_z]^T \\ &= [\boldsymbol{\omega}^T [\mathbf{x} \ \mathbf{y} \ \mathbf{z}]]^T \\ &= \mathbf{R}\boldsymbol{\omega}.\end{aligned}$$

Substituting $\boldsymbol{\omega}'$ in (S3.1) leads to conclude

$$\mathbf{R}\mathbf{S}(\boldsymbol{\omega})\mathbf{R}^T = \mathbf{S}(\mathbf{R}\boldsymbol{\omega}).$$

Solution to Problem 3.2

For the cylindrical arm in Fig. 2.35, the Jacobian is

$$\mathbf{J}(\mathbf{q}) = \begin{bmatrix} \mathbf{z}_0 \times (\mathbf{p} - \mathbf{p}_0) & \mathbf{z}_1 & \mathbf{z}_2 \\ \mathbf{z}_0 & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

where the various vectors can be computed from arm direct kinematics

$$\begin{aligned}\mathbf{p}_0 &= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} & \mathbf{p} &= \begin{bmatrix} -d_3 s_1 \\ d_3 c_1 \\ d_2 \end{bmatrix} \\ \mathbf{z}_0 = \mathbf{z}_1 &= \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} & \mathbf{z}_2 &= \begin{bmatrix} -s_1 \\ c_1 \\ 0 \end{bmatrix}.\end{aligned}$$

Then, the expression of the geometric Jacobian is

$$\mathbf{J} = \begin{bmatrix} -d_3 c_1 & 0 & -s_1 \\ -d_3 s_1 & 0 & c_1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad (\text{S3.2})$$

which reveals that it is inherently impossible to rotate about axes x and y .

Solution to Problem 3.3

For the SCARA manipulator in Fig. 2.36, the Jacobian is

$$\mathbf{J}(\mathbf{q}) = \begin{bmatrix} \mathbf{z}_0 \times (\mathbf{p} - \mathbf{p}_0) & \mathbf{z}_1 \times (\mathbf{p} - \mathbf{p}_1) & \mathbf{z}_2 & \mathbf{z}_3 \times (\mathbf{p} - \mathbf{p}_3) \\ \mathbf{z}_0 & \mathbf{z}_1 & \mathbf{0} & \mathbf{z}_3 \end{bmatrix}$$

where the various vectors can be computed from manipulator direct kinematics

$$\mathbf{p}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{p}_1 = \begin{bmatrix} a_1 c_1 \\ a_1 s_1 \\ 0 \end{bmatrix} \quad \mathbf{p}_3 = \begin{bmatrix} a_1 c_1 + a_2 c_{12} \\ a_1 s_1 + a_2 s_{12} \\ d_3 \end{bmatrix} \quad \mathbf{p} = \begin{bmatrix} a_1 c_1 + a_2 c_{12} \\ a_1 s_1 + a_2 s_{12} \\ d_3 \end{bmatrix}$$

$$\mathbf{z}_0 = \mathbf{z}_1 = \mathbf{z}_2 = \mathbf{z}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

Then, the expression of the geometric Jacobian is

$$\mathbf{J} = \begin{bmatrix} -a_1 s_1 - a_2 s_{12} & -a_2 s_{12} & 0 & 0 \\ a_1 c_1 + a_2 c_{12} & a_2 c_{12} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

which reveals that it is inherently impossible to rotate about axes x and y . In view of this, if a four-dimensional operational space ($r = 4$) is of concern, then the (4×4) analytical Jacobian can be extracted from \mathbf{J} by eliminating rows 4 and 5, i.e.,

$$\mathbf{J}_A = \begin{bmatrix} -a_1 s_1 - a_2 s_{12} & -a_2 s_{12} & 0 & 0 \\ a_1 c_1 + a_2 c_{12} & a_2 c_{12} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}. \quad (\text{S3.3})$$

Solution to Problem 3.4

From the expression of the geometric Jacobian in (3.35), it is possible to extract the analytical Jacobian by eliminating the three null rows, i.e.,

$$\mathbf{J}_A = \begin{bmatrix} -a_1 s_1 - a_2 s_{12} - a_3 s_{123} & -a_2 s_{12} - a_3 s_{123} & -a_3 s_{123} \\ a_1 c_1 + a_2 c_{12} + a_3 c_{123} & a_2 c_{12} + a_3 c_{123} & a_3 c_{123} \\ 1 & 1 & 1 \end{bmatrix}.$$

In order to compute its determinant, it is worth subtracting the second column from the first one and the third column from the second one, respectively, giving

$$\mathbf{J}'_A = \begin{bmatrix} -a_1 s_1 & -a_2 s_{12} & -a_3 s_{123} \\ a_1 c_1 & a_2 c_{12} & a_3 c_{123} \\ 0 & 0 & 1 \end{bmatrix}$$

whose determinant is

$$\det(\mathbf{J}'_A) = a_1 a_2 s_2.$$

It follows that, for $a_1, a_2 \neq 0$, the determinant vanishes whenever

$$\vartheta_2 = 0 \quad \vartheta_2 = \pi,$$

that are the same singularities of the two-link planar arm. It may be verified that the rank of the Jacobian does not further decrease, and thus the arm has no other singularities.

Solution to Problem 3.5

For the spherical arm in Fig. 2.22, the Jacobian is

$$\mathbf{J}(\mathbf{q}) = \begin{bmatrix} \mathbf{z}_0 \times (\mathbf{p} - \mathbf{p}_0) & \mathbf{z}_1 \times (\mathbf{p} - \mathbf{p}_1) & \mathbf{z}_2 \\ \mathbf{z}_0 & \mathbf{z}_1 & \mathbf{0} \end{bmatrix}$$

where the various vectors can be computed from arm direct kinematics

$$\mathbf{p}_0 = \mathbf{p}_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{p} = \begin{bmatrix} c_1 s_2 d_3 - s_1 d_2 \\ s_1 s_2 d_3 + c_1 d_2 \\ c_2 d_3 \end{bmatrix}$$

$$\mathbf{z}_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \mathbf{z}_1 = \begin{bmatrix} -s_1 \\ c_1 \\ 0 \end{bmatrix} \quad \mathbf{z}_2 = \begin{bmatrix} c_1 s_2 \\ s_1 s_2 \\ c_2 \end{bmatrix}.$$

Then, the expression of the geometric Jacobian is

$$\mathbf{J} = \begin{bmatrix} -s_1 s_2 d_3 - c_1 d_2 & c_1 c_2 d_3 & c_1 s_2 \\ c_1 s_2 d_3 - s_1 d_2 & s_1 c_2 d_3 & s_1 s_2 \\ 0 & -s_2 d_3 & c_2 \\ 0 & -s_1 & 0 \\ 0 & c_1 & 0 \\ 1 & 0 & 0 \end{bmatrix}.$$

With three DOFs, it is worth considering end-effector linear velocity only, corresponding to the first three rows of the Jacobian, i.e.,

$$\mathbf{J}_P = \begin{bmatrix} -s_1 s_2 d_3 - c_1 d_2 & c_1 c_2 d_3 & c_1 s_2 \\ c_1 s_2 d_3 - s_1 d_2 & s_1 c_2 d_3 & s_1 s_2 \\ 0 & -s_2 d_3 & c_2 \end{bmatrix}$$

which could have been also obtained by differentiating the vector \mathbf{p} above with respect to the joint variable vector \mathbf{q} (analytical Jacobian). In order to determine singularities of \mathbf{J}_P , its determinant has to be computed, giving

$$\det(\mathbf{J}_P) = -d_3^2 s_2.$$

This vanishes if $s_2 = 0$ and/or $d_3 = 0$. The first situation occurs whenever

$$\vartheta_2 = 0 \quad \vartheta_2 = \pi,$$

whereas the second situation occurs whenever

$$d_3 = 0,$$

i.e., when Frame 3 coincides with Frame 2 and thus Joint 2 velocity does not contribute to end-effector linear velocity. Finally, it is worth observing that both types of singularities are internal singularities.

Solution to Problem 3.6

From the geometric Jacobian in (S3.2), the Jacobian relative to end-effector linear velocity can be extracted by considering only the first three rows, i.e.,

$$\mathbf{J}_P = \begin{bmatrix} -d_3 c_1 & 0 & -s_1 \\ -d_3 s_1 & 0 & c_1 \\ 0 & 1 & 0 \end{bmatrix}.$$

Its determinant is

$$\det(\mathbf{J}_P) = d_3$$

which vanishes at the singularity

$$d_3 = 0.$$

This occurs when the end effector is located along Joint 1 axis, and thus this singularity is conceptually similar to the shoulder singularity of an anthropomorphic arm.

Solution to Problem 3.7

With reference to the analytical Jacobian in (S3.3), it is worth subtracting the second column from the first one. The modified Jacobian becomes

$$\mathbf{J}'_A = \begin{bmatrix} -a_1 s_1 & -a_2 s_{12} & 0 & 0 \\ a_1 c_1 & a_2 c_{12} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

whose determinant is

$$\det(\mathbf{J}'_A) = a_1 a_2 s_2.$$

For $a_1, a_2 \neq 0$, the determinant vanishes whenever

$$\vartheta_2 = 0 \quad \vartheta_2 = \pi$$

which are the same singularities of the two-link and three-link planar arms. In other words, the addition of a prismatic joint to a three-link planar arm—which makes it become a SCARA manipulator—does not introduce further singularities.

Solution to Problem 3.8

The singular values of \mathbf{J} are defined as the square roots of the eigenvalues of the matrix $\mathbf{J}\mathbf{J}^T$. On the other hand, the determinant of a matrix is given by the product of its eigenvalues, and thus the manipulability measure in (3.56) is given by the product of the singular values of the Jacobian matrix.

Solution to Problem 3.9

Let R be the radius of the circular object. The expression of the minimum distance of the arm from the obstacle can be written as

$$d = \min_{d_i}(d_i - R) \quad i = 1, 2, 3, \quad (\text{S3.4})$$

where d_i denotes the minimum distance of Link i from the centre \mathbf{o} of the obstacle. Let then $\mathbf{p}_i = [p_{ix} \ p_{iy}]^T$ denote the position vector of the point along Link i which is closest to \mathbf{o} . Link i lies along a line, whose equation can be written in the parametric form

$$p_{ix} = x_{i-1} + s_i(x_i - x_{i-1}) \quad (\text{S3.5})$$

$$p_{iy} = y_{i-1} + s_i(y_i - y_{i-1}) \quad (\text{S3.6})$$

where (x_{i-1}, y_{i-1}) and (x_i, y_i) are the coordinates of the origins of Link $i - 1$ and Link i frames, respectively; obviously, the parameter s_i varies in the range $(0, 1)$. Therefore, computation of the minimum distance is given by the solution to a minimization problem as a function of s_i , i.e.,

$$\min_{s_i} ((p_{ix} - x_o)^2 + (p_{iy} - y_o)^2) \quad 0 \leq s_i \leq 1$$

where (x_o, y_o) are the coordinates of \mathbf{o} . Using (S3.5) and (S3.6), and taking the derivative with respect to s_i of the above function leads to the solution

$$s_i^* = -\frac{(x_{i-1} - x_o)(x_i - x_{i-1}) + (y_{i-1} - y_o)(y_i - y_{i-1})}{a_i^2} \quad (\text{S3.7})$$

where $a_i^2 = (x_i - x_{i-1})^2 + (y_i - y_{i-1})^2$. For the point \mathbf{p}_i to belong to Link i , the parameter shall be subject to the constraint

$$s_i = \max\{0, \min\{1, s_i^*\}\}, \quad (\text{S3.8})$$

which then allows computing p_{ix} and p_{iy} as in (S3.5), (S3.6). In sum, the following operating procedure can be established.

Execute steps **a** and **b** for $i = 1, 2, 3$:

a. Compute s_i^* as in (S3.7) and s_i as in (S3.8).

b. Compute \mathbf{p}_i as in (S3.5), and

$$d_i = \sqrt{(p_{ix} - x_o)^2 + (p_{iy} - y_o)^2}.$$

To complete, compute d as in (S3.4).

Solution to Problem 3.10

The problem is that to invert differential kinematics $\mathbf{v}_e = \mathbf{J}\dot{\mathbf{q}}$ by tolerating a finite error $\boldsymbol{\epsilon}$, i.e.,

$$\mathbf{v}_e - \mathbf{J}\dot{\mathbf{q}} = \boldsymbol{\epsilon}.$$

The solution can be obtained by minimizing the cost functional

$$g''(\dot{\mathbf{q}}, \boldsymbol{\epsilon}) = \frac{1}{2}k^2\dot{\mathbf{q}}^T\dot{\mathbf{q}} + \frac{1}{2}\boldsymbol{\epsilon}^T\boldsymbol{\epsilon}.$$

To cast the problem as a classic constrained optimization problem, let

$$\mathbf{w} = \begin{bmatrix} \dot{\mathbf{q}} \\ \boldsymbol{\epsilon} \end{bmatrix} \quad \mathbf{A} = [\mathbf{J} \quad \mathbf{I}].$$

With these positions, the above differential kinematics constraint can be rewritten as

$$\mathbf{v}_e - \mathbf{A}\mathbf{w} = \mathbf{0},$$

while the cost functional becomes

$$g''(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{Q}\mathbf{w}$$

where

$$\mathbf{Q} = \begin{bmatrix} k^2\mathbf{I} & \mathbf{O} \\ \mathbf{O} & \mathbf{I} \end{bmatrix}$$

is a positive definite weighting matrix. By using the method of Lagrangian multipliers, the modified cost functional can be written as

$$g''(\mathbf{w}, \boldsymbol{\lambda}) = \frac{1}{2}\mathbf{w}^T\mathbf{Q}\mathbf{w} + \boldsymbol{\lambda}^T(\mathbf{v}_e - \mathbf{A}\mathbf{w})$$

where $\boldsymbol{\lambda}$ is an $(r \times 1)$ vector. The optimal solution is conceptually similar to (3.50), i.e.,

$$\mathbf{w} = \mathbf{Q}^{-1}\mathbf{A}^T(\mathbf{A}\mathbf{Q}^{-1}\mathbf{A}^T)^{-1}\mathbf{v}_e.$$

In view of the above expressions of \mathbf{w} , \mathbf{A} and \mathbf{Q} , the solution can be formally written after simple algebraic manipulation as

$$\begin{bmatrix} \dot{\mathbf{q}} \\ \boldsymbol{\epsilon} \end{bmatrix} = \begin{bmatrix} \mathbf{J}^T \\ k^2\mathbf{I} \end{bmatrix} (\mathbf{J}\mathbf{J}^T + k^2\mathbf{I})^{-1} \mathbf{v}_e.$$

It follows that the joint velocity is given by

$$\dot{\mathbf{q}} = \mathbf{J}^T (\mathbf{J}\mathbf{J}^T + k^2\mathbf{I})^{-1} \mathbf{v}_e$$

where

$$\mathbf{J}^* = \mathbf{J}^T (\mathbf{J}\mathbf{J}^T + k^2 \mathbf{I})^{-1}$$

is the damped least-squares inverse of \mathbf{J} , whereas the error is given by

$$\boldsymbol{\epsilon} = k^2 \left(\mathbf{J}\mathbf{J}^T + k^2 \mathbf{I} \right)^{-1} \mathbf{v}_e.$$

The damping factor k establishes the relative weight between the kinematic constraint $\mathbf{v}_e = \mathbf{J}\dot{\mathbf{q}}$ and the minimum norm joint velocity requirement. In the neighbourhood of a singularity, k is to be chosen large enough so as to render differential kinematics inversion well conditioned, whereas far from singularities, k can be chosen small (even $k = 0$) so as to guarantee accurate differential kinematics inversion.

Solution to Problem 3.11

From (3.6), it is

$$\mathbf{S}(\boldsymbol{\omega}) = \dot{\mathbf{R}}(\phi) \mathbf{R}^T(\phi)$$

with $\mathbf{R}(\phi)$ as in (2.18). Taking the required derivatives gives

$$\begin{aligned} \mathbf{S}(\boldsymbol{\omega}) &= \dot{\mathbf{R}}_z(\varphi) \mathbf{R}_{y'}(\vartheta) \mathbf{R}_{z''}(\psi) \mathbf{R}_{z''}^T(\psi) \mathbf{R}_{y'}^T(\vartheta) \mathbf{R}_z^T(\varphi) \\ &\quad + \mathbf{R}_z(\varphi) \dot{\mathbf{R}}_{y'}(\vartheta) \mathbf{R}_{z''}(\psi) \mathbf{R}_{z''}^T(\psi) \mathbf{R}_{y'}^T(\vartheta) \mathbf{R}_z^T(\varphi) \\ &\quad + \mathbf{R}_z(\varphi) \mathbf{R}_{y'}(\vartheta) \dot{\mathbf{R}}_{z''}(\psi) \mathbf{R}_{z''}^T(\psi) \mathbf{R}_{y'}^T(\vartheta) \mathbf{R}_z^T(\varphi). \end{aligned}$$

Then, in view of (2.4), (3.8), (3.11), the previous expression can be simplified into

$$\begin{aligned} \mathbf{S}(\boldsymbol{\omega}) &= \mathbf{S}(\dot{\varphi}z) \\ &\quad + \mathbf{R}_z(\varphi) \mathbf{S}(\dot{\vartheta} \mathbf{y}') \mathbf{R}_z^T(\varphi) \\ &\quad + \mathbf{R}_z(\varphi) \mathbf{R}_{y'}(\vartheta) \mathbf{S}(\dot{\psi} z'') \mathbf{R}_{y'}^T(\vartheta) \mathbf{R}_z^T(\varphi) \\ &= \mathbf{S}(\dot{\varphi}z) + \mathbf{S}(\dot{\vartheta} \mathbf{R}_z(\varphi) \mathbf{y}') + \mathbf{S}(\dot{\psi} \mathbf{R}_z(\varphi) \mathbf{R}_{y'}(\vartheta) z'') \end{aligned}$$

and then

$$\begin{aligned} \boldsymbol{\omega} &= [z \quad \mathbf{R}_z(\varphi) \mathbf{y}' \quad \mathbf{R}_z(\varphi) \mathbf{R}_{y'}(\vartheta) z''] \dot{\boldsymbol{\phi}} \\ &= \begin{bmatrix} 0 & -s_\varphi & c_\varphi s_\vartheta \\ 0 & c_\varphi & s_\varphi s_\vartheta \\ 1 & 0 & c_\vartheta \end{bmatrix} \dot{\boldsymbol{\phi}}. \end{aligned}$$

Solution to Problem 3.12

From (2.21), the contributions to the angular velocity of the derivatives of the three Roll–Pitch–Yaw angles can be written as:

$$\dot{\boldsymbol{\varphi}} = \dot{\varphi} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \dot{\boldsymbol{\vartheta}} = \dot{\vartheta} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \dot{\boldsymbol{\psi}} = \dot{\psi} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix},$$

where the superscripts indicate that rotations occur about axes of current frames. Then, expressing the last two vectors with respect to the reference frame gives

$$\begin{aligned}\dot{\boldsymbol{\vartheta}} &= \mathbf{R}_z(\varphi)\dot{\boldsymbol{\vartheta}}' = \dot{\vartheta} \begin{bmatrix} -s_\varphi \\ c_\varphi \\ 0 \end{bmatrix} \\ \dot{\boldsymbol{\psi}} &= \mathbf{R}_z(\varphi)\mathbf{R}_y(\vartheta)\dot{\boldsymbol{\psi}}'' = \dot{\psi} \begin{bmatrix} c_\varphi c_\vartheta \\ s_\varphi c_\vartheta \\ -s_\vartheta \end{bmatrix}.\end{aligned}$$

Summing the three contributions leads to

$$\begin{aligned}\boldsymbol{\omega} &= \dot{\boldsymbol{\varphi}} + \dot{\boldsymbol{\vartheta}} + \dot{\boldsymbol{\psi}} \\ &= \begin{bmatrix} 0 & -s_\varphi & c_\varphi c_\vartheta \\ 0 & c_\varphi & s_\varphi c_\vartheta \\ 1 & 0 & -s_\vartheta \end{bmatrix} \dot{\boldsymbol{\phi}},\end{aligned}$$

from which it is

$$\mathbf{T}(\boldsymbol{\phi}) = \begin{bmatrix} 0 & -s_\varphi & c_\varphi c_\vartheta \\ 0 & c_\varphi & s_\varphi c_\vartheta \\ 1 & 0 & -s_\vartheta \end{bmatrix}.$$

The determinant of matrix \mathbf{T} is $-c_\vartheta$, which implies that the relationship cannot be inverted for $\vartheta = -\pi/2, \pi/2$. These are representation singularities of $\boldsymbol{\phi}$; notice also that solutions (2.22) and (2.23) degenerate at such singularities.

Solution to Problem 3.13

The contributions to the angular velocity of the derivatives of the three Euler angles can be written as:

$$\dot{\boldsymbol{\varphi}} = \dot{\varphi}\mathbf{e}_\varphi \quad \dot{\boldsymbol{\vartheta}}' = \dot{\vartheta}\mathbf{e}'_\vartheta \quad \dot{\boldsymbol{\psi}}'' = \dot{\psi}\mathbf{e}''_\psi,$$

where \mathbf{e}_φ , \mathbf{e}'_ϑ and \mathbf{e}''_ψ are the constant unit vectors of the axes of the current frames, which depend on the particular triplet of Euler angles. The superscripts indicate that vectors are expressed in the current frames. In the case $\varphi = \vartheta = \psi = 0$ the current frames coincide and the three contributions can be added, i.e.,

$$\boldsymbol{\omega} = \dot{\boldsymbol{\varphi}} + \dot{\boldsymbol{\vartheta}}' + \dot{\boldsymbol{\psi}}'' = \mathbf{T}(\mathbf{0})\dot{\boldsymbol{\phi}}$$

with $\mathbf{T}(\mathbf{0}) = [\mathbf{e}_\varphi \quad \mathbf{e}'_\vartheta \quad \mathbf{e}''_\psi]$. Therefore, with the choice

$$\mathbf{e}_\varphi = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{e}'_\vartheta = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \mathbf{e}''_\psi = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},$$

i.e., for XYZ angles, it is $\mathbf{T}(\mathbf{0}) = \mathbf{I}$.

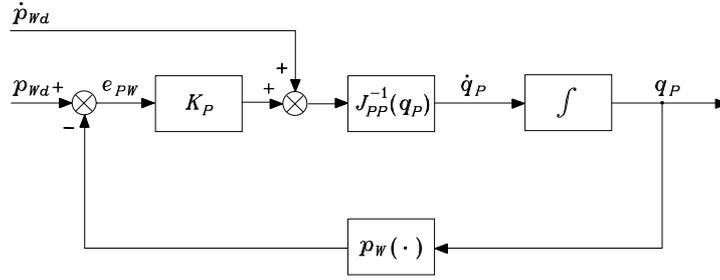


Fig. S3.1. Block scheme of the inverse kinematics algorithm for arm position

Solution to Problem 3.14

With reference to a 6-DOFs manipulator having a spherical wrist, let $\mathbf{q}_P = [q_1 \ q_2 \ q_3]^T$ denote the vector of joint variables determining arm position and $\mathbf{q}_O = [\vartheta_4 \ \vartheta_5 \ \vartheta_6]^T$ the vector of joint variables determining wrist orientation. For such manipulator, it is possible to solve inverse kinematics in two stages; namely, first solve for \mathbf{q}_P , and then solve for \mathbf{q}_O .

Let $\mathbf{p}_d(t)$ and $\mathbf{R}_d(t) = [\mathbf{n}_d(t) \ \mathbf{s}_d(t) \ \mathbf{a}_d(t)]$ be the desired time history of end-effector position and orientation. From (2.93), it is possible to compute the desired time behaviour of wrist position as

$$\mathbf{p}_{Wd} = \mathbf{p}_d - d_6 \mathbf{a}_d.$$

Taking time derivative of both sides gives

$$\dot{\mathbf{p}}_{Wd} = \dot{\mathbf{p}}_d - d_6 \boldsymbol{\omega}_d \times \mathbf{a}_d$$

where the time derivative of the unit vector \mathbf{a}_d (constant norm) has been expressed as $\dot{\mathbf{a}}_d = \boldsymbol{\omega}_d \times \mathbf{a}_d$.

By using the block-partitioned form of the Jacobian in (3.42) with $\mathbf{p} = \mathbf{p}_W$, the differential kinematics equation for arm linear velocity is

$$\dot{\mathbf{p}}_W = \mathbf{J}_{PP}(\mathbf{q}_P) \dot{\mathbf{q}}_P$$

where \mathbf{J}_{PP} formally coincides with \mathbf{J}_{11} evaluated for $\mathbf{p} = \mathbf{p}_W$. In view of this, the joint velocity solution of kind (3.70) can be written as

$$\dot{\mathbf{q}}_P = \mathbf{J}_{PP}^{-1}(\mathbf{q}_P) (\dot{\mathbf{p}}_{Wd} + \mathbf{K}_P e_{PW})$$

where $e_{PW} = \mathbf{p}_{Wd} - \mathbf{p}_W$ and \mathbf{K}_P is a positive definite matrix. The resulting block scheme is shown in Fig. S3.1.

Once the time behaviour of \mathbf{q}_P and $\dot{\mathbf{q}}_P$ has been computed in the first stage, the differential kinematics equation for wrist angular velocity is

$$\boldsymbol{\omega} = \mathbf{J}_{OP}(\mathbf{q}_P) \dot{\mathbf{q}}_P + \mathbf{J}_{OO}(\mathbf{q}_P, \mathbf{q}_O) \dot{\mathbf{q}}_O$$

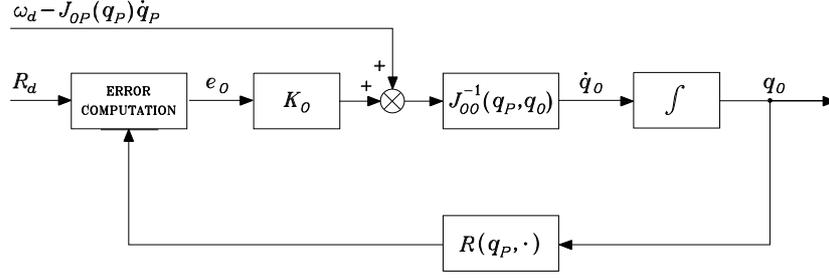


Fig. S3.2. Block scheme of the inverse kinematics algorithm for wrist orientation

where \mathbf{J}_{OP} and \mathbf{J}_{OO} respectively denote the Jacobians \mathbf{J}_{21} and \mathbf{J}_{22} in (3.42) evaluated for $\mathbf{p} = \mathbf{p}_W$. In view of this, the joint velocity solution of kind (3.70) can be written as

$$\dot{\mathbf{q}}_O = \mathbf{J}_{OO}^{-1}(\mathbf{q}_P, \mathbf{q}_O)(\boldsymbol{\omega}_d - \mathbf{J}_{OP}(\mathbf{q}_P)\dot{\mathbf{q}}_P + \mathbf{K}_O \mathbf{e}_O)$$

where \mathbf{e}_O is the orientation error in (3.85) and \mathbf{K}_O is a positive definite matrix. The resulting block scheme for the second stage is shown in Fig. S3.2.

It is anticipated that the two-stage algorithm leads to a computational savings with respect to the algorithm based on the inverse of the full Jacobian.

Solution to Problem 3.15

In the case $\dot{\mathbf{x}}_d \neq \mathbf{0}$, the time derivative of the Lyapunov function is given as in (3.77)

$$\dot{V}(\mathbf{e}) = \mathbf{e}^T \mathbf{K} \dot{\mathbf{x}}_d - \mathbf{e}^T \mathbf{K} \mathbf{J}_A(\mathbf{q}) \mathbf{J}_A^T(\mathbf{q}) \mathbf{K} \mathbf{e}$$

where the sign of the first term is indefinite, and the second term is less than or equal to zero. This function can be upper bounded as

$$\dot{V}(\mathbf{e}) \leq |\mathbf{e}^T \mathbf{K} \dot{\mathbf{x}}_d| - \|\mathbf{J}_A^T \mathbf{K} \mathbf{e}\|^2.$$

In the worst case, it is necessary to take the largest value of the first term and the smallest value of the second term, leading to

$$\dot{V}(\mathbf{e}) \leq \|\mathbf{e}\| \lambda_{\max}(\mathbf{K}) \|\dot{\mathbf{x}}_d\|_{\max} - \sigma_r^2(\mathbf{J}_A) \lambda_{\min}^2(\mathbf{K}) \|\mathbf{e}\|^2,$$

where $\lambda_{\max}(\mathbf{K})$ ($\lambda_{\min}(\mathbf{K})$) denotes the maximum (minimum) eigenvalue of \mathbf{K} , $\|\dot{\mathbf{x}}_d\|_{\max}$ denotes the maximum end-effector velocity, and $\sigma_r(\mathbf{J}_A)$ denotes the minimum singular value of \mathbf{J}_A . The quadratic term in the error prevails over the linear term as long as

$$\|\mathbf{e}\| \geq \frac{\|\dot{\mathbf{x}}_d\|_{\max} \lambda_{\max}(\mathbf{K})}{\sigma_r^2(\mathbf{J}_A) \lambda_{\min}^2(\mathbf{K})}$$

and $\dot{V}(\mathbf{e}) \leq 0$. It follows that an upper bound on the error is given by

$$\|\mathbf{e}\|_{\max} = \frac{\|\dot{\mathbf{x}}_d\|_{\max}}{k\sigma_r^2(\mathbf{J}_A)}$$

where \mathbf{K} has been conveniently chosen as a diagonal matrix $\mathbf{K} = k\mathbf{I}$.

Solution to Problem 3.16

The matrix product $\mathbf{R}_d\mathbf{R}^T$ gives

$$\mathbf{R}_d\mathbf{R}^T = \begin{bmatrix} n_{dx}n_x + s_{dx}s_x + a_{dx}a_x & n_{dx}n_y + s_{dx}s_y + a_{dx}a_y & n_{dx}n_z + s_{dx}s_z + a_{dx}a_z \\ n_{dy}n_x + s_{dy}s_x + a_{dy}a_x & n_{dy}n_y + s_{dy}s_y + a_{dy}a_y & n_{dy}n_z + s_{dy}s_z + a_{dy}a_z \\ n_{dz}n_x + s_{dz}s_x + a_{dz}a_x & n_{dz}n_y + s_{dz}s_y + a_{dz}a_y & n_{dz}n_z + s_{dz}s_z + a_{dz}a_z \end{bmatrix}.$$

In view of (3.84), subtracting the element [1, 2] from the element [2, 1] and equating the difference to $2r_zs_\vartheta$ as in (2.28) gives

$$2r_zs_\vartheta = (n_{dy}n_x - n_{dx}n_y) + (s_{dy}s_x - s_{dx}s_y) + (a_{dy}a_x - a_{dx}a_y),$$

and thus

$$r_zs_\vartheta = \frac{1}{2}((\mathbf{n} \times \mathbf{n}_d)_z + (\mathbf{s} \times \mathbf{s}_d)_z + (\mathbf{a} \times \mathbf{a}_d)_z)$$

where the subscripts z denote the z -components of the relevant vectors. In a similar way, subtracting the elements [3, 1] from the respective elements [1, 3] and equating the difference to $2r_y s_\vartheta$ gives

$$r_y s_\vartheta = \frac{1}{2}((\mathbf{n} \times \mathbf{n}_d)_y + (\mathbf{s} \times \mathbf{s}_d)_y + (\mathbf{a} \times \mathbf{a}_d)_y).$$

Yet, subtracting the elements [2, 3] from the respective elements [3, 2] and equating the difference to $2r_x s_\vartheta$ gives

$$r_x s_\vartheta = \frac{1}{2}((\mathbf{n} \times \mathbf{n}_d)_x + (\mathbf{s} \times \mathbf{s}_d)_x + (\mathbf{a} \times \mathbf{a}_d)_x).$$

In turn, grouping the previous three expressions leads to

$$\mathbf{r} \sin \vartheta = \frac{1}{2}(\mathbf{n} \times \mathbf{n}_d + \mathbf{s} \times \mathbf{s}_d + \mathbf{a} \times \mathbf{a}_d)$$

and, in view of the definition of orientation error as in (3.83), the result in (3.85) follows.

Solution to Problem 3.17

The expression of the orientation error from (3.85) is

$$\mathbf{e}_O = \frac{1}{2}(\mathbf{n} \times \mathbf{n}_d + \mathbf{s} \times \mathbf{s}_d + \mathbf{a} \times \mathbf{a}_d).$$

Taking the time derivative of the first term on the right-hand side gives

$$\begin{aligned}\frac{d}{dt}(\mathbf{n} \times \mathbf{n}_d) &= \dot{\mathbf{n}} \times \mathbf{n}_d + \mathbf{n} \times \dot{\mathbf{n}}_d \\ &= -\mathbf{S}(\mathbf{n}_d)\dot{\mathbf{n}} + \mathbf{S}(\mathbf{n})\dot{\mathbf{n}}_d.\end{aligned}$$

Expressing the time derivatives of the unit vectors as

$$\begin{aligned}\dot{\mathbf{n}} &= \boldsymbol{\omega} \times \mathbf{n} = -\mathbf{S}(\mathbf{n})\boldsymbol{\omega} \\ \dot{\mathbf{n}}_d &= \boldsymbol{\omega}_d \times \mathbf{n}_d = -\mathbf{S}(\mathbf{n}_d)\boldsymbol{\omega}_d\end{aligned}$$

leads to

$$\frac{d}{dt}(\mathbf{n} \times \mathbf{n}_d) = \mathbf{S}(\mathbf{n}_d)\mathbf{S}(\mathbf{n})\boldsymbol{\omega} - \mathbf{S}(\mathbf{n})\mathbf{S}(\mathbf{n}_d)\boldsymbol{\omega}_d.$$

Proceeding in a similar way for the other two terms on the right-hand side of (3.85) yields

$$\begin{aligned}\frac{d}{dt}(\mathbf{s} \times \mathbf{s}_d) &= \mathbf{S}(\mathbf{s}_d)\mathbf{S}(\mathbf{s})\boldsymbol{\omega} - \mathbf{S}(\mathbf{s})\mathbf{S}(\mathbf{s}_d)\boldsymbol{\omega}_d \\ \frac{d}{dt}(\mathbf{a} \times \mathbf{a}_d) &= \mathbf{S}(\mathbf{a}_d)\mathbf{S}(\mathbf{a})\boldsymbol{\omega} - \mathbf{S}(\mathbf{a})\mathbf{S}(\mathbf{a}_d)\boldsymbol{\omega}_d.\end{aligned}$$

In turn, grouping the previous three contributions to the time derivative of the orientation error gives

$$\begin{aligned}\dot{\epsilon}_O &= -\frac{1}{2}(\mathbf{S}(\mathbf{n})\mathbf{S}(\mathbf{n}_d) + \mathbf{S}(\mathbf{s})\mathbf{S}(\mathbf{s}_d) + \mathbf{S}(\mathbf{a})\mathbf{S}(\mathbf{a}_d))\boldsymbol{\omega}_d \\ &\quad + \frac{1}{2}(\mathbf{S}(\mathbf{n}_d)\mathbf{S}(\mathbf{n}) + \mathbf{S}(\mathbf{s}_d)\mathbf{S}(\mathbf{s}) + \mathbf{S}(\mathbf{a}_d)\mathbf{S}(\mathbf{a}))\boldsymbol{\omega}\end{aligned}$$

and, in view of the position (3.87), the result in (3.86) follows.

Solution to Problem 3.18

It is not difficult to see that, by using the skew-symmetric operator $\mathbf{S}(\cdot)$, (2.33) can be rewritten in compact form as

$$\mathbf{R}(\eta, \boldsymbol{\epsilon}) = (2\eta^2 - 1)\mathbf{I} + 2\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T + 2\eta\mathbf{S}(\boldsymbol{\epsilon}). \quad (\text{S3.9})$$

Computing the time derivative of (S3.9) gives

$$\dot{\mathbf{R}}(\eta, \boldsymbol{\epsilon}) = 4\dot{\eta}\eta\mathbf{I} + 2\dot{\boldsymbol{\epsilon}}\boldsymbol{\epsilon}^T + 2\boldsymbol{\epsilon}\dot{\boldsymbol{\epsilon}}^T + 2\dot{\eta}\mathbf{S}(\boldsymbol{\epsilon}) + 2\eta\mathbf{S}(\dot{\boldsymbol{\epsilon}}). \quad (\text{S3.10})$$

Substituting (S3.9) and (S3.10) into (3.6) gives:

$$\begin{aligned}\mathbf{S}(\boldsymbol{\omega}) &= \dot{\mathbf{R}}\mathbf{R}^T \\ &= 4\dot{\eta}\eta(2\eta^2 - 1)\mathbf{I} + 8\dot{\eta}\eta\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T - 8\dot{\eta}\eta^2\mathbf{S}(\boldsymbol{\epsilon}) \\ &\quad + 2(2\eta^2 - 1)\dot{\boldsymbol{\epsilon}}\boldsymbol{\epsilon}^T + 4\dot{\boldsymbol{\epsilon}}\boldsymbol{\epsilon}^T\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T - 4\eta\dot{\boldsymbol{\epsilon}}\boldsymbol{\epsilon}^T\mathbf{S}(\boldsymbol{\epsilon}) \\ &\quad + 2(2\eta^2 - 1)\boldsymbol{\epsilon}\dot{\boldsymbol{\epsilon}}^T + 4\boldsymbol{\epsilon}\dot{\boldsymbol{\epsilon}}^T\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T - 4\eta\boldsymbol{\epsilon}\dot{\boldsymbol{\epsilon}}^T\mathbf{S}(\boldsymbol{\epsilon}) \\ &\quad + 2\dot{\eta}(2\eta^2 - 1)\mathbf{S}(\boldsymbol{\epsilon}) + 4\dot{\eta}\mathbf{S}(\boldsymbol{\epsilon})\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T - 4\dot{\eta}\eta\mathbf{S}(\boldsymbol{\epsilon})\mathbf{S}(\boldsymbol{\epsilon}) \\ &\quad + 2\dot{\eta}(2\eta^2 - 1)\mathbf{S}(\dot{\boldsymbol{\epsilon}}) + 4\eta\mathbf{S}(\dot{\boldsymbol{\epsilon}})\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T - 4\eta^2\mathbf{S}(\dot{\boldsymbol{\epsilon}})\mathbf{S}(\boldsymbol{\epsilon}).\end{aligned} \quad (\text{S3.11})$$

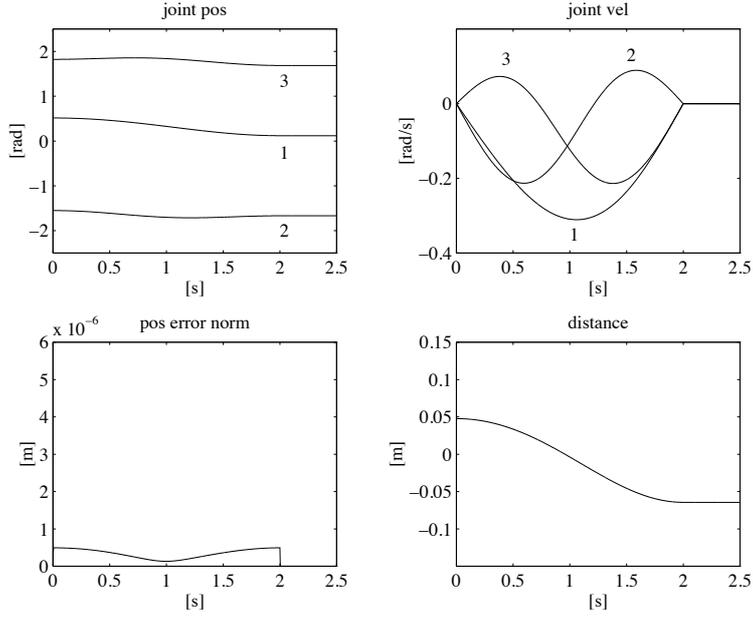


Fig. S3.4. Time history of the joint positions and velocities, of the norm of end-effector position error, and of the distance from the obstacle in the unconstrained case

Solution to Problem 3.19

Multiplying both sides of (S3.13) by ϵ^T and using (S3.12), (2.32) yields

$$\epsilon^T \omega = 2\eta \epsilon^T \dot{\epsilon} - 2\dot{\eta} \epsilon^T \epsilon = -2\dot{\eta}$$

and thus

$$\dot{\eta} = -\frac{1}{2} \epsilon^T \omega.$$

Multiplying both sides of (S3.13) by $\mathbf{S}(\epsilon)$ yields

$$\mathbf{S}(\epsilon) \omega = 2\mathbf{S}(\epsilon) \mathbf{S}(\epsilon) \dot{\epsilon} + 2\eta \mathbf{S}(\epsilon) \dot{\epsilon},$$

which, by virtue of (S3.12), (2.32), (S3.13) and the property $\mathbf{S}(\mathbf{x})\mathbf{S}(\mathbf{y}) = \mathbf{y}\mathbf{x}^T - \mathbf{x}^T\mathbf{y}\mathbf{I}$, can be rewritten in the form

$$\begin{aligned} \mathbf{S}(\epsilon) \omega &= -2\dot{\eta} \epsilon - 2(1 - \eta^2) \dot{\epsilon} + \eta(\omega - 2\eta \dot{\epsilon} + 2\dot{\eta} \epsilon) \\ &= -2\dot{\epsilon} + \eta \omega \end{aligned}$$

and thus

$$\dot{\epsilon} = \frac{1}{2} (\eta \mathbf{I} - \mathbf{S}(\epsilon)) \omega.$$

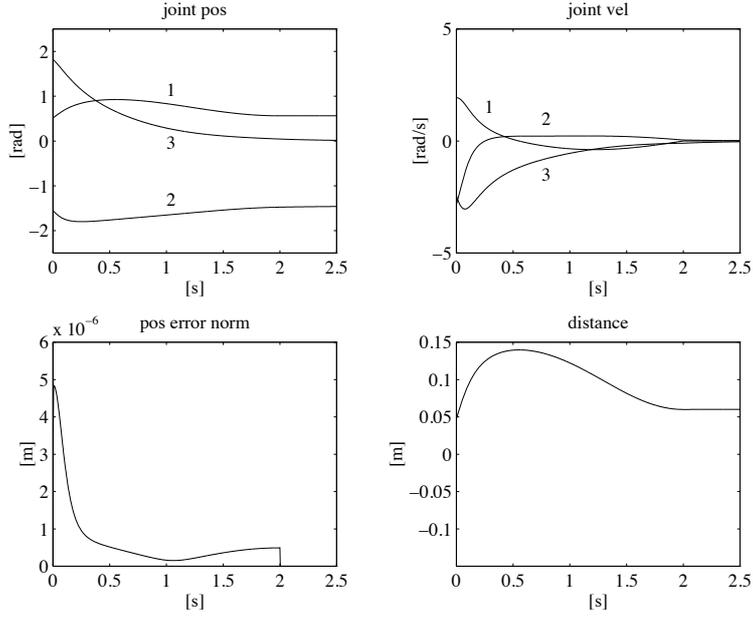


Fig. S3.5. Time history of the joint positions and velocities, of the norm of end-effector position error, and of the distance from the obstacle in the constrained case

Solution to Problem 3.20

Differentiating (3.96) with respect to time gives

$$\begin{aligned}\dot{V} &= 2(\eta_d - \eta)(\dot{\eta}_d - \dot{\eta}) + 2(\epsilon_d - \epsilon)^T(\dot{\epsilon}_d - \dot{\epsilon}) \\ &= (\eta\epsilon_d^T - \eta_d\epsilon^T - \mathbf{S}(\epsilon_d)\epsilon)(\omega_d - \omega)\end{aligned}$$

where the quaternion propagation in (3.94), (3.95), and the property $\mathbf{S}(\epsilon)\epsilon = \mathbf{0}$ have been exploited. The result in (3.97) follows by using (3.91) and (3.93).

Solution to Problem 3.21

From the given $\mathbf{p}_d(0)$, the initial posture of the arm $\mathbf{q}(0)$ is computed according to the formulæ in Sect. 2.12.1 (elbow-up) with an arbitrary value of ϕ , e.g., $\phi = \pi/4$. The desired trajectory regards the vertical component of end-effector position and is chosen as

$$p_{dy}(t) = p_{dy}(0) + 0.5(1 - \cos(0.5\pi t))(p_{dy}(2) - p_{dy}(0)) \quad 0 \leq t \leq 2$$

with $p_{dy}(0) = 0.2$ and $p_{dy}(2) = -0.2$. The desired velocity is found by differentiation of the above trajectory. As for the constraint, the minimum distance of the arm from the obstacle (3.58) is computed according to the solution to

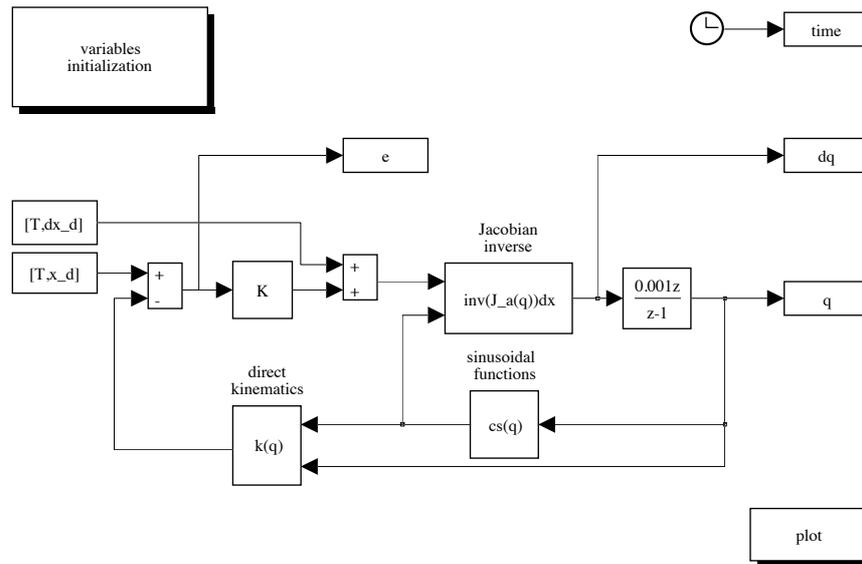


Fig. S3.6. SIMULINK block diagram of closed-loop Jacobian inverse algorithm

Problem 3.11. Its gradient is found by numerical differentiation with a step of 10^{-6} .

The inverse kinematics algorithm based on (3.72) is used, with the matrix gain $\mathbf{K} = \text{diag}\{500, 500\}$ and $\dot{\mathbf{q}}_0$ as in (3.55) with $k_0 = 35$. The resulting SIMULINK block diagram is shown in Fig. S3.3, where both the unconstrained case ($k_0 = 0$) and the constrained case can be run.

The files with the solution can be found in Folder 3_21.

The resulting joint positions and velocities, as well as the norm of end-effector position error and the distance from the obstacle, in the two cases, are shown in Figs. S3.4 and S3.5, respectively.

It can be recognized that the position error is small along the trajectory and tends to zero at steady state in both cases. In the unconstrained case, the arm is seen to collide with the obstacle (negative distance). On the other hand, in the constrained case, large initial values of joint velocities occur which are needed to move the arm away from the obstacle.

Solution to Problem 3.22

From the given $\mathbf{x}_d(0)$, the initial posture of the manipulator $\mathbf{q}(0)$ is computed according to the formulæ in the solution to Problem 2.20 (elbow-up). The desired trajectory for the end-effector position is chosen as

$$\mathbf{x}_d(t) = \mathbf{x}_d(0) + 0.5(1 - \cos(0.5\pi t))(\mathbf{x}_{dy}(2) - \mathbf{x}_{dy}(0)) \quad 0 \leq t \leq 2$$

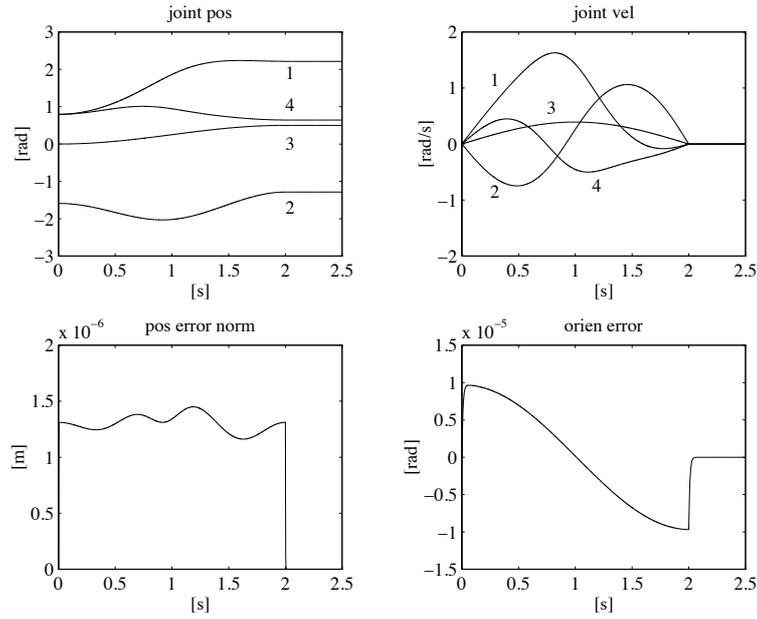


Fig. S3.7. Time history of the joint positions and velocities, of the norm of end-effector position error, and of the end-effector orientation error with closed-loop Jacobian inverse algorithm

with $\mathbf{x}_d(0) = [0.7 \ 0 \ 0 \ 0]^T$ and $\mathbf{x}_d(2) = [0 \ 0.8 \ 0.5 \ \pi/2]^T$. The desired velocity is found by differentiation of the above trajectory.

First, the inverse kinematics algorithm based on (3.70) is used where the analytical Jacobian is given in (S3.3). The matrix gain is chosen as $\mathbf{K} = \text{diag}\{500, 500, 500, 100\}$. The resulting SIMULINK block diagram is shown in Fig. S3.6.

The files with the solution can be found in Folder 3_22.

The resulting joint positions and velocities, as well as the norm of end-effector position error and the end-effector orientation error, are shown in Fig. S3.7. The tracking performance is satisfactory and the errors vanish at steady state.

Next, the inverse kinematics algorithm based on (3.76) is used with the same matrix gain as above. The resulting SIMULINK block diagram is shown in Fig. S3.8.

The resulting joint positions and velocities, as well as the norm of end-effector position error and the end-effector orientation error, are shown in Fig. S3.9. It can be recognized that tracking errors slightly increase, but steady-state performance remains satisfactory.

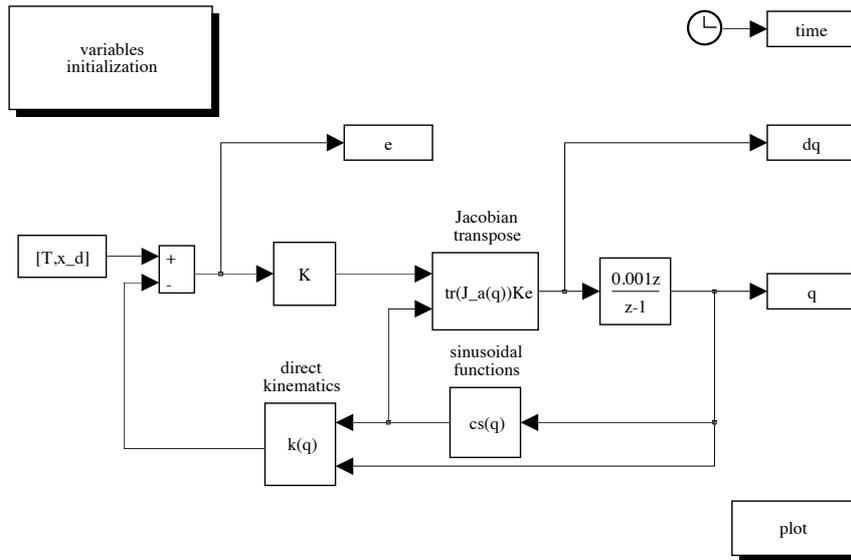


Fig. S3.8. SIMULINK block diagram of closed-loop Jacobian transpose algorithm

Solution to Problem 3.23

The core of the quadratic form defining the velocity manipulability ellipsoid in (3.123) is given by the inverse of the core of the quadratic form defining the force manipulability ellipsoid in (3.126). Let $\mathbf{A} = \mathbf{J}\mathbf{J}^T$ denote such core. The eigenvalues λ_i and the eigenvectors \mathbf{u}_i of \mathbf{A} satisfy the equation

$$\mathbf{A}\mathbf{u}_i = \lambda_i\mathbf{u}_i.$$

Premultiplying both sides by \mathbf{A}^{-1} and dividing by λ_i gives

$$\mathbf{A}^{-1}\mathbf{u}_i = \frac{1}{\lambda_i}\mathbf{u}_i,$$

which demonstrates the notable result that the directions of the principal axes (eigenvectors) of the force and velocity manipulability ellipsoids coincide while their dimensions (eigenvalues) are in inverse proportion.

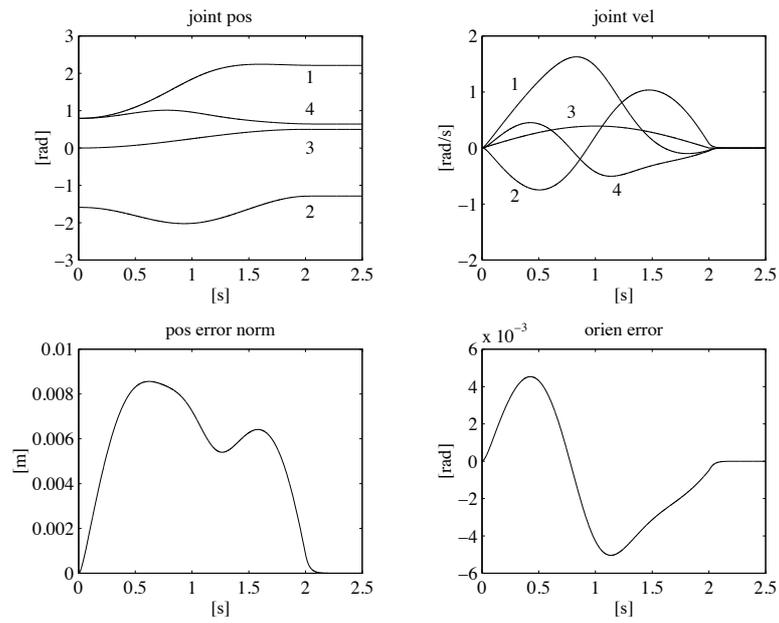


Fig. S3.9. Time history of the joint positions and velocities, of the norm of end-effector position error, and of the end-effector orientation error with closed-loop Jacobian transpose algorithm

Trajectory Planning

Solution to Problem 4.1

In order to satisfy initial and final constraints on position, velocity and acceleration, a fifth-order polynomial is needed

$$q(t) = a_5 t^5 + a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t + a_0.$$

The coefficients can be determined by solving the following system of linear equations:

$$\begin{aligned} a_0 &= q_i \\ a_1 &= \dot{q}_i \\ 2a_2 &= \ddot{q}_i \\ a_5 t_f^5 + a_4 t_f^4 + a_3 t_f^3 + a_2 t_f^2 + a_1 t_f + a_0 &= q_f \\ 5a_5 t_f^4 + 4a_4 t_f^3 + 3a_3 t_f^2 + 2a_2 t_f + a_1 &= \dot{q}_f \\ 20a_5 t_f^3 + 12a_4 t_f^2 + 6a_3 t_f + 2a_2 &= \ddot{q}_f. \end{aligned}$$

With the given data, the files used to compute the coefficients and to generate the time behaviour of position, velocity and acceleration can be found in Folder 4_1.

The coefficients are:

$$a_0 = 1 \quad a_1 = 0 \quad a_2 = 0 \quad a_3 = \frac{15}{4} \quad a_4 = -\frac{45}{16} \quad a_5 = \frac{9}{16}$$

and the resulting trajectory is illustrated in Fig. S4.1.

Solution to Problem 4.2

Imposing velocity constraints yields

$$\begin{aligned} \dot{q}_i &= 0 \\ \dot{q}_f &= k(1 - \cos(at_f)). \end{aligned}$$

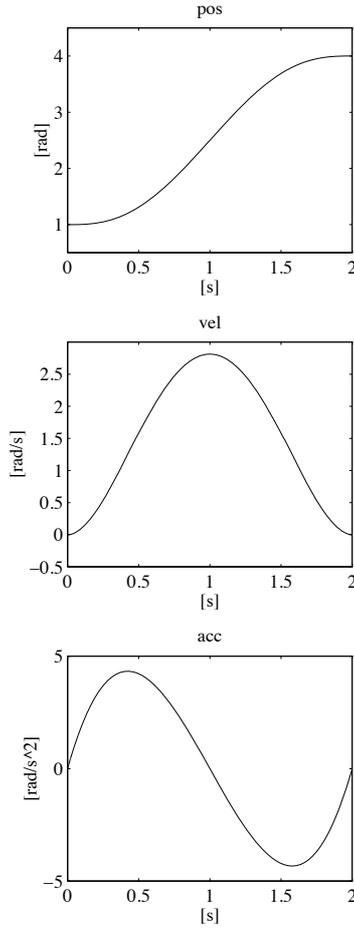


Fig. S4.1. Time history of position, velocity and acceleration with a fifth-order polynomial timing law

The second equation implies that

$$0 \leq \frac{\dot{q}_f}{k} \leq 2$$

which is satisfied for all k as long as $\dot{q}_f = 0$. In that case, it is $a = 2\pi/t_f$. Then, integrating the given velocity profile gives

$$q(t) = k_1 + k \left(t - \frac{t_f}{2\pi} \sin \left(\frac{2\pi t}{t_f} \right) \right),$$

and imposing position constraints yields the two coefficients

$$k_1 = q_i$$

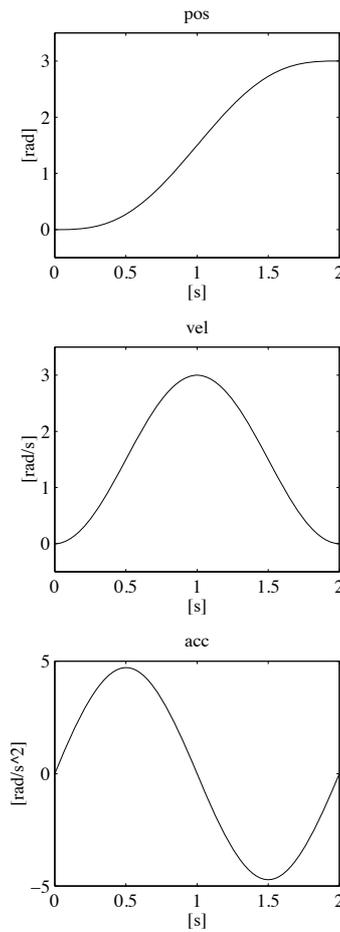


Fig. S4.2. Time history of position, velocity and acceleration with a raised cosine timing law

$$k = \frac{q_f - q_i}{t_f}.$$

With the given data, the files used to compute the two coefficients and to generate the time behaviour of position, velocity and acceleration can be found in Folder 4.2.

The coefficients are:

$$k = 1.5 \quad k_1 = 0$$

and the resulting trajectory is illustrated in Fig. S4.2.

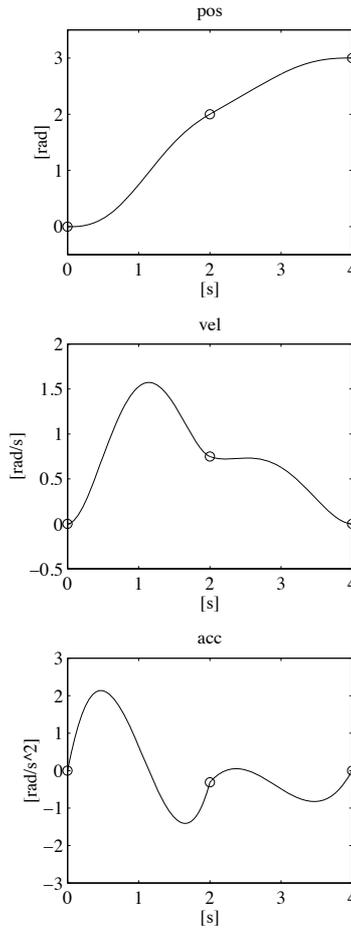


Fig. S4.3. Time history of position, velocity and acceleration with two fifth-order interpolating polynomials

Solution to Problem 4.3

It is required to satisfy initial and final constraints on position, velocity and acceleration (6 equations), intermediate position constraints (2 equations), and continuity of velocity and acceleration at the intermediate point (2 equations). Therefore, with two fifth-order interpolating polynomials it is possible to specify also velocity and acceleration at the intermediate point.

Let q_m denote the intermediate point at time t_m . A viable choice is to compute the mean velocities in the two intervals

$$\dot{q}_a = \frac{q_m - q_i}{t_m - t_i} \quad \dot{q}_b = \frac{q_f - q_m}{t_f - t_m},$$

and then to choose velocity at the intermediate point as

$$\dot{q}_m = \frac{\dot{q}_a + \dot{q}_b}{2}.$$

Likewise, for acceleration it is

$$\ddot{q}_a = \frac{\dot{q}_m - \dot{q}_i}{t_m - t_i} \quad \ddot{q}_b = \frac{\dot{q}_f - \dot{q}_m}{t_f - t_m}$$

and then

$$\ddot{q}_m = \frac{\ddot{q}_a + \ddot{q}_b}{2}.$$

With the given data and null initial and final velocities and accelerations, the files used to generate the time behaviour of position, velocity and acceleration can be found in the Folder 4_3.

The resulting trajectory is illustrated in Fig. S4.3.

Solution to Problem 4.4

By using (4.13), (4.14), (4.23), (4.16), (4.25), the time derivative of $\ddot{I}_k(t)$ in (4.27) can be written as

$$\begin{aligned} \dot{\ddot{I}}_k(t) = & -\frac{\ddot{I}_k(t_k)}{2\Delta t_k}(t_{k+1} - t)^2 + \frac{\ddot{I}_{k+1}(t_{k+1})}{2\Delta t_k}(t - t_k)^2 \\ & + \frac{q_{k+1} - q_k}{\Delta t_k} + \frac{\Delta t_k}{6} \left(\ddot{I}_k(t_k) - \ddot{I}_{k+1}(t_{k+1}) \right) \quad k = 1, \dots, N+1 \end{aligned} \quad (\text{S4.1})$$

where q_2 and q_{N+1} are unknown. Then, using (4.15), (4.24) leads to

$$\begin{aligned} \frac{\Delta t_k}{6} \ddot{I}_k(t_k) + \frac{\Delta t_k + \Delta t_{k+1}}{3} \ddot{I}_{k+1}(t_{k+1}) + \frac{\Delta t_{k+1}}{6} \ddot{I}_{k+2}(t_{k+2}) \\ = \frac{q_k}{\Delta t_k} - \left(\frac{1}{\Delta t_{k+1}} + \frac{1}{\Delta t_k} \right) q_{k+1} + \frac{q_{k+2}}{\Delta t_{k+1}} \quad k = 1, \dots, N \end{aligned} \quad (\text{S4.2})$$

where $\ddot{I}_1(t_1) = \ddot{q}_i$ as in (4.19) and $\ddot{I}_{N+2}(t_{N+2}) = \ddot{q}_f$ as in (4.22).

Evaluating (S4.1) for $k = 1$ at $t = t_1$ and applying (4.17), (4.18) yields

$$q_2 = q_i + \dot{q}_i \Delta t_1 + \ddot{q}_i \frac{\Delta t_1^2}{3} + \ddot{I}_2(t_2) \frac{\Delta t_1^2}{6}. \quad (\text{S4.3})$$

Likewise, evaluating (S4.1) for $k = N+1$ at $t = t_{N+2}$ and applying (4.20), (4.21) yields

$$q_{N+1} = q_f - \dot{q}_f \Delta t_{N+1} + \ddot{q}_f \frac{\Delta t_{N+1}^2}{3} + \ddot{I}_{N+1}(t_{N+1}) \frac{\Delta t_{N+1}^2}{6}. \quad (\text{S4.4})$$

In the case $N > 3$, substituting (S4.3) into (S4.2) for $k = 1, 2$ and (S4.4) into (S4.2) for $k = N-1, N$ leads to writing the linear system of N equations into N unknowns given in (4.28).

The coefficient matrix takes on the tridiagonal band structure

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & 0 & 0 \\ a_{21} & a_{22} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & a_{N-1,N-1} & a_{N-1,N} \\ 0 & 0 & \dots & a_{N,N-1} & a_{NN} \end{bmatrix}$$

where

$$\left\{ \begin{array}{l} a_{11} = \frac{\Delta t_1}{2} + \frac{\Delta t_2}{3} + \frac{\Delta t_1^2}{6\Delta t_2} \\ a_{kk} = \frac{\Delta t_k + \Delta t_{k+1}}{3} \quad k = 2, \dots, N-1 \\ a_{NN} = \frac{\Delta t_N}{3} + \frac{\Delta t_{N+1}}{2} + \frac{\Delta t_{N+1}^2}{6\Delta t_N} \\ a_{21} = \frac{\Delta t_2}{6} - \frac{\Delta t_1^2}{6\Delta t_2} \\ a_{k,k-1} = \frac{\Delta t_k}{6} \quad k = 3, \dots, N \\ a_{k,k+1} = \frac{\Delta t_{k+1}}{6} \quad k = 1, \dots, N-2 \\ a_{N-1,N} = \frac{\Delta t_N}{6} - \frac{\Delta t_{N+1}^2}{6\Delta t_N} \end{array} \right.$$

which depend only on the given time intervals Δt_k , $k = 1, \dots, N+1$.

Further, the components of the vector \mathbf{b} of known terms in (4.28) are:

$$\left\{ \begin{array}{l} b_1 = \frac{q_3 - q_i}{\Delta t_2} - \left(\frac{1}{\Delta t_2} + \frac{1}{\Delta t_1} \right) \left(\dot{q}_i \Delta t_1 + \ddot{q}_i \frac{\Delta t_1^2}{3} \right) - \ddot{q}_i \frac{\Delta t_1}{6} \\ b_2 = \frac{1}{\Delta t_2} \left(q_i + \dot{q}_i \Delta t_1 + \ddot{q}_i \frac{\Delta t_1^2}{3} \right) - \left(\frac{1}{\Delta t_3} + \frac{1}{\Delta t_2} \right) q_3 + \frac{q_4}{\Delta t_3} \\ b_k = \frac{q_k}{\Delta t_k} - \left(\frac{1}{\Delta t_{k+1}} + \frac{1}{\Delta t_k} \right) q_{k+1} + \frac{q_{k+2}}{\Delta t_{k+1}} \quad k = 3, \dots, N-2 \\ b_{N-1} = \frac{q_{N-1}}{\Delta t_{N-1}} - \left(\frac{1}{\Delta t_N} + \frac{1}{\Delta t_{N-1}} \right) q_N + \frac{1}{\Delta t_N} \left(q_f - \dot{q}_f \Delta t_{N+1} + \ddot{q}_f \frac{\Delta t_{N+1}^2}{3} \right) \\ b_N = \frac{q_N - q_f}{\Delta t_N} - \left(\frac{1}{\Delta t_{N+1}} + \frac{1}{\Delta t_N} \right) \left(-\dot{q}_f \Delta t_{N+1} + \ddot{q}_f \frac{\Delta t_{N+1}^2}{3} \right) - \ddot{q}_f \frac{\Delta t_{N+1}}{6} \end{array} \right.$$

where the values q_k are given for $k = 1, 3, \dots, N, N+2$.

On the other hand, in the case $N = 3$, the same expressions for the a 's can be used; for the b 's, instead, b_1 and b_3 can be computed as above and b_2 takes on the expression

$$b_2 = \frac{1}{\Delta t_2} \left(q_i + \dot{q}_i \Delta t_1 + \ddot{q}_i \frac{\Delta t_1^2}{3} \right) - \left(\frac{1}{\Delta t_3} + \frac{1}{\Delta t_2} \right) q_3 \\ + \frac{1}{\Delta t_3} \left(q_f - \dot{q}_f \Delta t_4 + \ddot{q}_f \frac{\Delta t_4^2}{3} \right).$$

Solution to Problem 4.5

The time instants of the two virtual points are chosen as $t_1 = 1$ and $t_4 = 3$. With the given data, the files used to compute the matrix \mathbf{A} and the vector \mathbf{b} in (4.28), and to generate the time behaviour of position, velocity and acceleration can be found in the Folder 4.5.

The matrix of coefficients is

$$\mathbf{A} = \begin{bmatrix} 1 & 1/6 & 0 \\ 0 & 2/3 & 0 \\ 0 & 1/6 & 1 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 2 \\ -1 \\ -1 \end{bmatrix},$$

and the resulting trajectory is illustrated in Fig. S4.4.

Solution to Problem 4.6

The interpolating trajectory given by a sequence of linear polynomials with parabolic blends can be analytically described as

$$q(t) = \begin{cases} a_{k1}(t - t_k) + a_{k0} & t_k + \frac{\Delta t'_k}{2} \leq t < t_{k+1} - \frac{\Delta t'_{k+1}}{2} \\ b_{k2}(t - t_k)^2 + b_{k1}(t - t_k) + b_{k0} & t_k - \frac{\Delta t'_k}{2} \leq t < t_k + \frac{\Delta t'_k}{2}. \end{cases}$$

The velocity in the linear segments is

$$\dot{q}_{k-1,k} = \begin{cases} \dot{q}_i & k = 1 \\ \frac{q_k - q_{k-1}}{\Delta t_{k-1}} & k = 2, \dots, N \\ \dot{q}_f & k = N + 1, \end{cases}$$

and then the coefficients of the linear polynomials are

$$a_{k0} = q_k \quad a_{k1} = \dot{q}_{k,k+1} \quad k = 1, \dots, N - 1.$$

Imposing continuity of velocity between the linear and parabolic polynomials at time instants $(t_k - \Delta t'_k/2)$ and $(t_k + \Delta t'_k/2)$ allows computing the coefficients

$$b_{k1} = \frac{\dot{q}_{k,k+1} + \dot{q}_{k-1,k}}{2} \quad b_{k2} = \frac{\dot{q}_{k,k+1} - \dot{q}_{k-1,k}}{2\Delta t'_k} \quad k = 1, \dots, N.$$

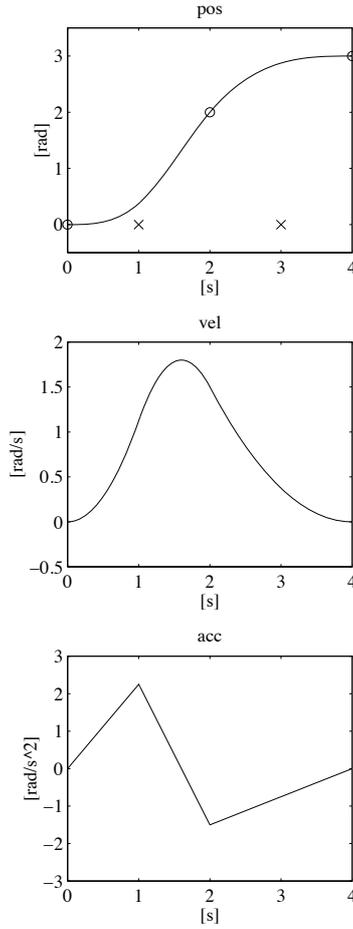


Fig. S4.4. Time history of position, velocity and acceleration with a cubic spline timing law

Then, imposing continuity of position at $(t_k + \Delta t'_k/2)$ yields

$$b_{k0} = q_k + (\dot{q}_{k,k+1} - \dot{q}_{k-1,k}) \frac{\Delta t'_k}{8} \quad k = 1, \dots, N;$$

it can be verified that this choice guarantees continuity of position also at $(t_k - \Delta t'_k/2)$.

The duration of the blend times is chosen as $\Delta t'_k = 0.2$ for $k = 1, 2, 3$. With the given data, the files used to compute the above coefficients, and to generate the time behaviour of position, velocity and acceleration can be found in Folder 4_6.

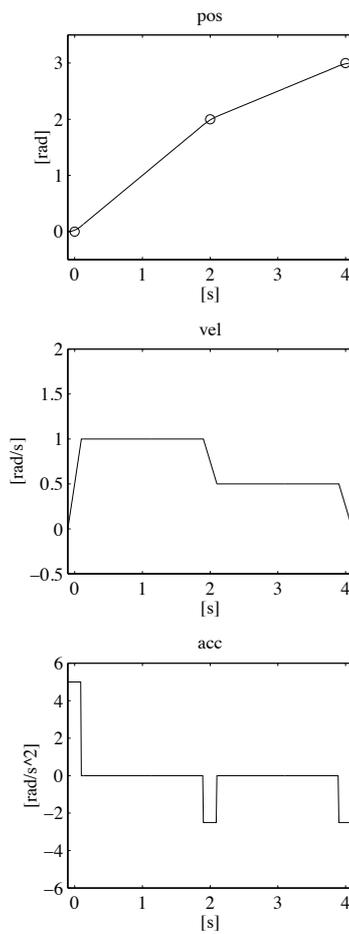


Fig. S4.5. Time history of position, velocity and acceleration with a timing law of interpolating linear polynomials with parabolic blends

The coefficients are:

$$a_{10} = 0 \quad a_{11} = 1 \quad a_{20} = 2 \quad a_{21} = 0.5$$

$$b_{10} = 0.025 \quad b_{11} = 0.5 \quad b_{12} = 2.5$$

$$b_{20} = 1.9875 \quad b_{21} = 0.75 \quad b_{22} = -1.25$$

$$b_{30} = 2.9875 \quad b_{31} = 0.25 \quad b_{32} = -1.25,$$

and the resulting trajectory is illustrated in Fig. S4.5.

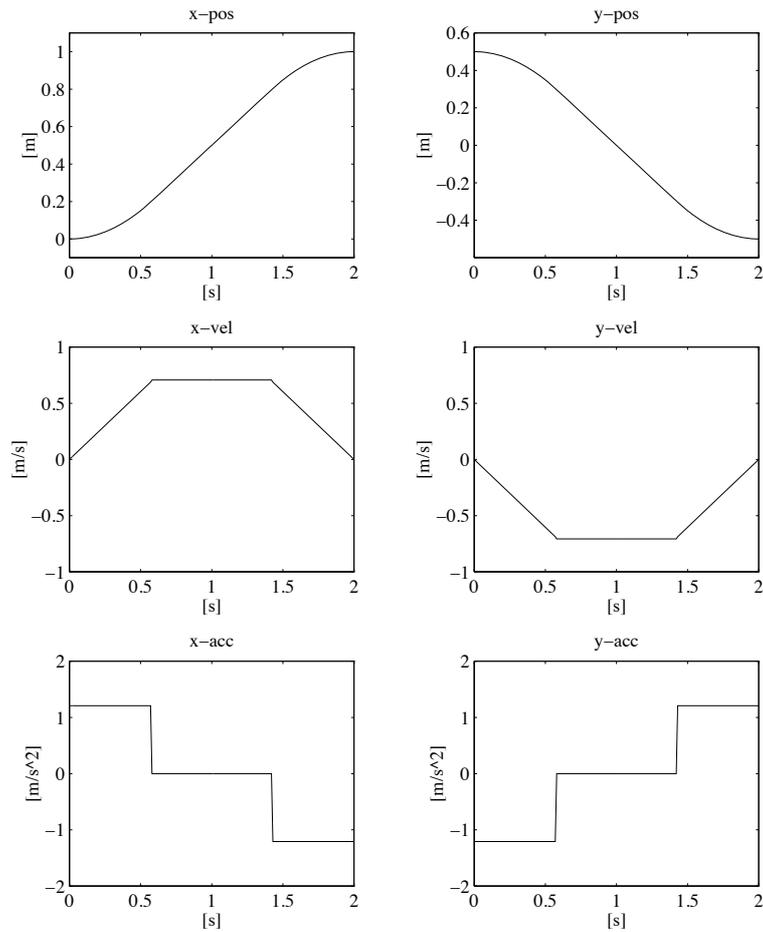


Fig. S4.6. Time history of position, velocity and acceleration of x - and y -components along a straight path with a trapezoidal velocity profile timing law

Solution to Problem 4.7

A trapezoidal velocity profile for the path coordinate s is assigned according to (4.8) with maximum velocity $\dot{s}_c = 1$. The timing law for $\mathbf{p}(t)$ and its derivatives is generated via (4.34) and (4.42).

With the given data, the files used to generate the time behaviour of position, velocity and acceleration can be found in Folder 4_7.

The resulting trajectories for the x - and y -components are illustrated in Fig. S4.6.

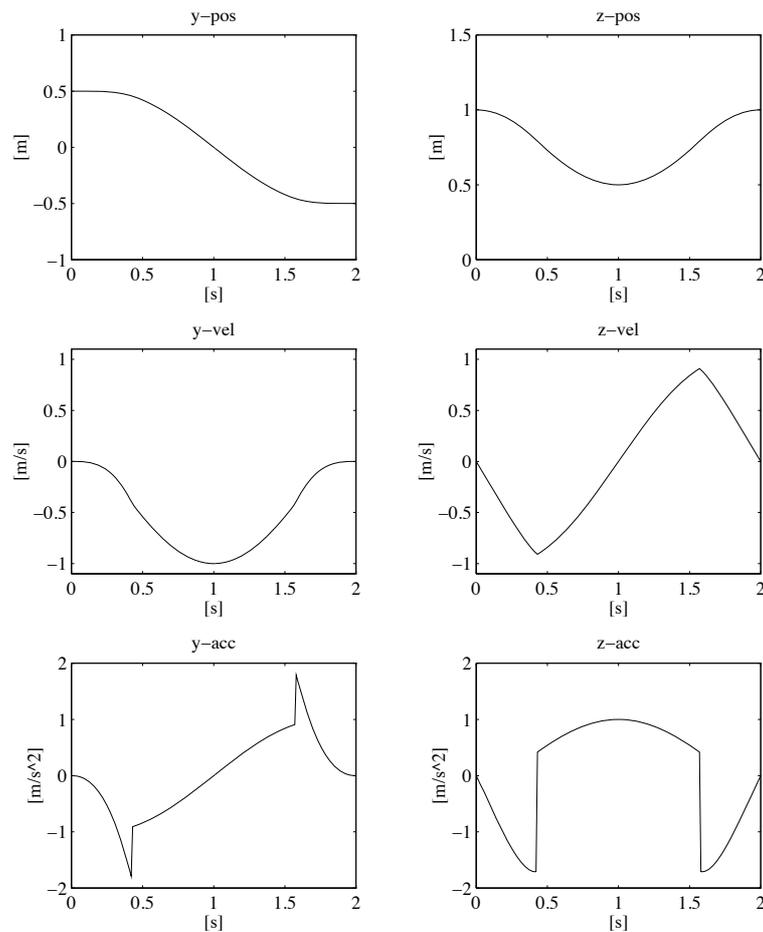


Fig. S4.7. Time history of position, velocity and acceleration of x - and y -components along a circular path with a trapezoidal velocity profile timing law

Solution to Problem 4.8

The unit vectors of the axes of the reference frame for the circle are chosen as

$$\mathbf{x}' = \mathbf{y} \quad \mathbf{y}' = -\mathbf{z} \quad \mathbf{z}' = -\mathbf{x};$$

notice that the choice $\mathbf{z}' = -\mathbf{x}$ ensures that the path is executed clockwise about axis \mathbf{x} . From (4.39), the path representation is

$$\mathbf{p}(s) = \begin{bmatrix} 0 \\ 0.5 \cos(2s) \\ 1 - 0.5 \sin(2s) \end{bmatrix}.$$

Since it is required to execute half a circle with $t_f = 2$, the final value of the path coordinate is $s_f = s(2) = \pi/2$. Then, a trapezoidal velocity profile for the path coordinate s is assigned according to (4.8) with maximum velocity $\dot{s}_c = 1$. The timing law for $\mathbf{p}(t)$ and its derivatives is generated via (4.39), (4.44).

With the given data, the files used to generate the time behaviour of position, velocity and acceleration can be found in Folder 4_8.

The resulting trajectories for the y - and z -components are illustrated in Fig. S4.7.

Actuators and Sensors

Solution to Problem 5.1

At steady state, (5.1)–(5.4) can be written in the time domain as

$$v_a = R_a i_a + v_g \quad (\text{S5.1})$$

$$v_g = k_v \omega_m \quad (\text{S5.2})$$

$$c_m = F_m \omega_m + c_r \quad (\text{S5.3})$$

$$c_m = k_t i_a \quad (\text{S5.4})$$

where

$$v_a = C_i(0)G_v(v'_c - k_i i_a). \quad (\text{S5.5})$$

Combining (S5.1), (S5.2), (S5.5) with $k_i = 0$ and $K = C_i(0)G_v$ gives

$$Kv'_c = R_a i_a + k_v \omega_m$$

and yet, using (S5.3) and (S5.4) with $c_r = 0$, it is

$$Kv'_c = \left(\frac{F_m R_a}{k_t} + k_v \right) \omega_m.$$

If $F_m \ll k_v k_t / R_a$, then

$$\omega_m \approx \frac{K}{k_v} v'_c.$$

If instead $k_i \neq 0$, on reduction of (S5.1)–(S5.5), it is

$$Kv'_c = \left(\frac{R_a + Kk_i}{k_t} \right) c_m + k_v \omega_m.$$

If $Kk_i \gg R_a$, then

$$c_m \approx \frac{k_t}{k_i} \left(v'_c - \frac{k_v}{K} \omega_m \right).$$

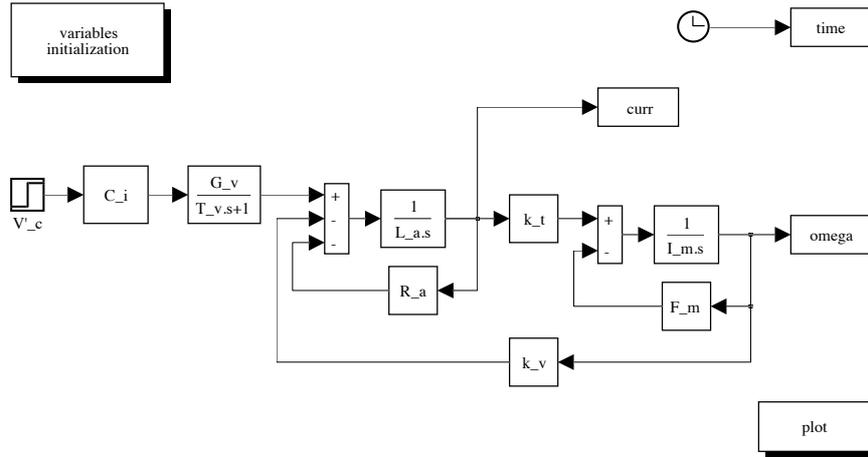


Fig. S5.1. SIMULINK block diagram of electric servomotor

From the block scheme in Fig. 5.3, the output can be computed via the closed-loop transfer functions; namely, the input/output and disturbance/output transfer functions, i.e.,

$$\Omega_m = \frac{\frac{Kk_t}{R_a(sI_m + F_m)}}{1 + \frac{k_vk_t}{R_a(sI_m + F_m)}} V'_c - \frac{\frac{1}{sI_m + F_m}}{1 + \frac{k_vk_t}{R_a(sI_m + F_m)}} C_r.$$

If $F_m \ll k_vk_t/R_a$, then

$$\Omega_m = \frac{\frac{K}{k_v}}{1 + s\frac{R_aI_m}{k_vk_t}} V'_c - \frac{\frac{R_a}{k_vk_t}}{1 + s\frac{R_aI_m}{k_vk_t}} C_r.$$

Finally, from the block scheme in Fig. 5.4, it is straightforward to compute the output as

$$\Omega_m = \frac{\frac{k_t}{k_iF_m}}{1 + s\frac{I_m}{F_m}} V'_c - \frac{\frac{1}{F_m}}{1 + s\frac{I_m}{F_m}} C_r.$$

Solution to Problem 5.2

With reference to the scheme in Fig. 5.2 ($k_i = 0$ and $C_r = 0$) and the given data, the resulting SIMULINK block diagram is shown in Fig. S5.1. The servo-

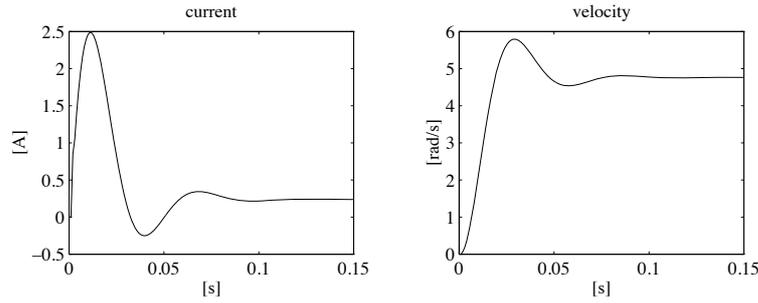


Fig. S5.2. Time history of current and velocity step response for an electric servomotor

motor is simulated as a continuous-time system using a variable-step integration method with a maximum step size of 1 ms. Notice that the amplifier time constant is too small to produce appreciable effects at this sampling time.

The files with the solution can be found in Folder 5_2.

The resulting current and velocity responses to a unit step voltage input V_c' are shown in Fig. S5.2. It can be seen that the steady-state velocity slightly deviates from the value of 5 rad/s as in (5.7), because of the presence of mechanical friction (F_m). This also implies that the steady-state current is different from zero.

Solution to Problem 5.3

By closing a current loop with large loop gain, the current response is expected to become much faster than the velocity response. Velocity plays the role of a disturbance for the voltage-to-current closed-loop system. Hence, it is worth choosing a PI control structure for $C_i(s)$ in order to obtain a null steady-state error, i.e.,

$$C_i(s) = K_I \frac{1 + sT_I}{s}.$$

With the given data, by setting $\Omega_m = 0$, the closed-loop input/output transfer function ($T_v \approx 0$) is

$$\frac{I_a}{V_c'} = \frac{1 + sT_I}{1 + s \frac{0.2 + K_I T_I}{K_I} + s^2 \frac{0.002}{K_I}}.$$

Since the settling time within 1% is given by

$$t_s = \frac{4.6}{\zeta \omega_n},$$

either the damping ratio or the natural frequency can be imposed. Choosing $\zeta = 0.7$ with $t_s = 2$ ms yields

$$K_I = 21592 \quad T_I = 0.000416.$$

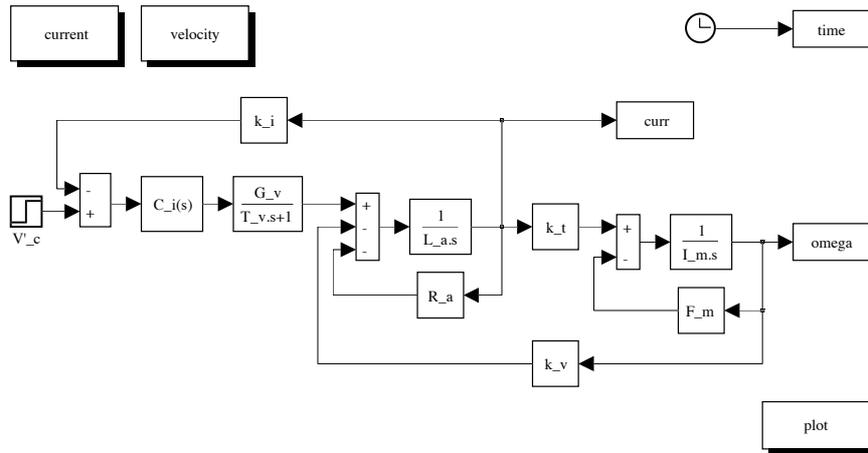


Fig. S5.3. SIMULINK block diagram of electric servomotor with current feedback loop and PI control

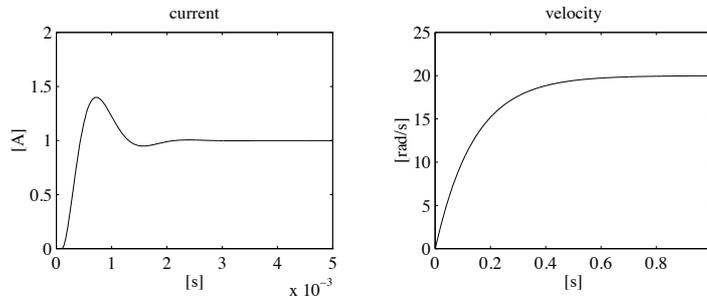


Fig. S5.4. Time history of current and velocity step response for an electric servomotor with current feedback loop and PI control

The resulting SIMULINK block diagram is shown in Fig. S5.3. The servomotor is simulated as a continuous-time system using a variable-step integration method. Since the current dynamics is much faster than the velocity dynamics, the maximum step size has been chosen much smaller than in Problem 5.2, i.e., 0.05 ms. Further, in order to make a comparison between the velocity response in this case and the case of Problem 5.2, the block diagram in Fig. S5.3 features a separate execution with a longer final time and a sampling time of 1 ms.

The files with the solution can be found in Folder 5_3.

The resulting current and velocity are shown in Fig. S5.4. The current response goes as expected, while the velocity response is slowed down with respect to the response in Fig. S5.2 because it is dominated by the time constant I_m/T_m .

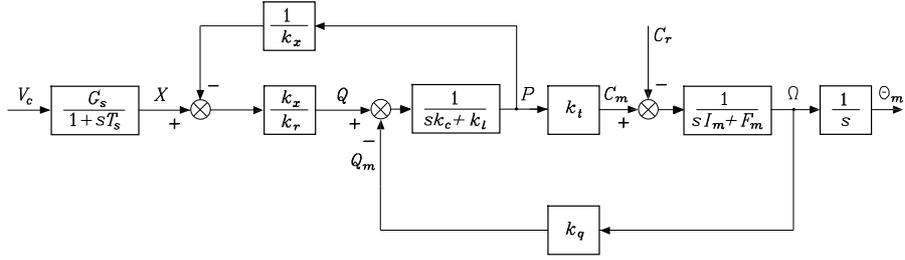


Fig. S5.5. Reduction on the block scheme of a hydraulic motor with servovalve and distributor

Solution to Problem 5.4

From this block scheme, the following relations can be found:

$$\begin{aligned}\Theta_m &= \frac{1}{s}\Omega_m \\ \Omega_m &= \frac{C_m - C_r}{sI_m + F_m} \\ C_m &= k_t P \\ P &= \frac{1}{sk_c + k_l} \left(\frac{k_x}{k_r} \left(X - \frac{P}{k_x} \right) - k_q \Omega_m \right) \\ &= \frac{k_x X - k_r k_q \Omega_m}{1 + k_r k_l + s k_r k_c} \\ X &= \frac{G_s}{1 + sT_s} V_c.\end{aligned}$$

Substituting (S5.6) into (S5.6), and then (S5.6) into (S5.6), yields

$$\left(1 + \frac{k_t k_r k_q}{(sI_m + F_m)(1 + k_r k_l + s k_r k_c)} \right) \Omega_m = \frac{k_t k_x}{(1 + k_r k_l + s k_r k_c)(sI_m + F_m)} X - \frac{1}{sI_m + F_m} C_r.$$

Solving for Ω_m , substituting it into (S5.6), and using (S5.6) leads to

$$\Theta_m = \frac{\frac{k_x G_s}{k_r k_q}}{s(1 + sT_s) \left(1 + \frac{F_m}{k_t k_r k_q} \left(1 + s \frac{I_m}{F_m} \right) (1 + k_r k_l + s k_r k_c) \right)} V_c$$

$$-\frac{\frac{1 + k_r k_l + s k_r k_c}{k_t k_r k_q}}{s \left(1 + \frac{F_m}{k_t k_r k_q} \left(1 + s \frac{I_m}{F_m} \right) (1 + k_r k_l + s k_r k_c) \right)} C_r.$$

Solution to Problem 5.5

Without loss of generality, the case of 4 bits is considered. The truth table of the interconversion logic circuit is reported in Table S5.1.

Table S5.1. Truth table of the interconversion logic circuit from Gray-code to binary code

#	Gray Code	Binary Code
	$x_1 x_2 x_3 x_4$	$y_1 y_2 y_3 y_4$
0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1
2	0 0 1 1	0 0 1 0
3	0 0 1 0	0 0 1 1
4	0 1 1 0	0 1 0 0
5	0 1 1 1	0 1 0 1
6	0 1 0 1	0 1 1 0
7	0 1 0 0	0 1 1 1
8	1 1 0 0	1 0 0 0
9	1 1 0 1	1 0 0 1
10	1 1 1 1	1 0 1 0
11	1 1 1 0	1 0 1 1
12	1 0 1 0	1 1 0 0
13	1 0 1 1	1 1 0 1
14	1 0 0 1	1 1 1 0
15	1 0 0 0	1 1 1 1

The first columns of the two codes are equal, and thus

$$y_1 = x_1.$$

From the second column at the output, it can be recognized that $y_2 = 1$ when $x_1 = 0$ and $x_2 = 1$, or else when $x_1 = 1$ and $x_2 = 0$, and thus

$$y_2 = \bar{x}_1 x_2 + x_1 \bar{x}_2 = x_1 \oplus x_2 = y_1 \oplus x_2$$

where “ \oplus ” is the symbol for the logical operator XOR. From the third column at the output, it can be recognized that $y_3 = 1$ when $y_2 = 0$ and $x_3 = 1$, or else when $y_2 = 1$ and $x_3 = 0$, and thus

$$y_3 = y_2 \oplus x_3.$$

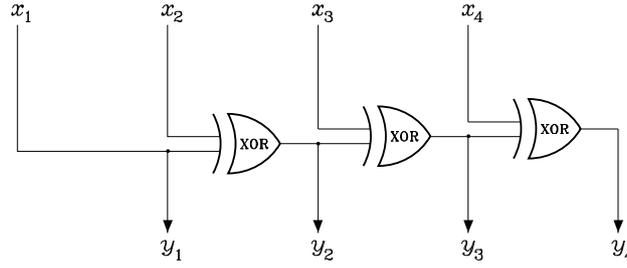


Fig. S5.6. Interconversion logic circuit from Gray code to binary code

Finally, from the fourth column it is

$$y_4 = y_3 \oplus x_4.$$

The resulting interconversion logic circuit is illustrated in Fig. S5.6 where the conventional symbol of XOR is used.

Solution to Problem 5.6

The skew-symmetric operator is

$$\mathbf{S}(\mathbf{r}_{cs}^c) = \begin{bmatrix} 0 & -0.2 & 0 \\ 0.2 & 0 & 0.3 \\ 0 & -0.3 & 0 \end{bmatrix},$$

and thus applying (5.32) gives

$$\begin{bmatrix} \mathbf{f}_c^c \\ \boldsymbol{\mu}_c^c \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0 & 0 & 1 \\ 0.3 & 0 & 0.2 & 0 & -1 & 0 \\ 0 & 0.3 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 20 \\ 0 \\ 0 \\ 0 \\ 6 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 20 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

that is,

$$\mathbf{f}_c^c = [0 \ 0 \ 20]^T \text{ N} \quad \boldsymbol{\mu}_c^c = [0 \ 0 \ 0]^T \text{ N}\cdot\text{m}.$$

As can be seen, there is no resulting moment in the contact frame.

Solution to Problem 5.7

Let the end-effector frame coincide with Frame 4 in Fig. S2.3. Then, with reference to (S2.3), the homogeneous transformation from the base frame to the end-effector frame at $\mathbf{q} = [0 \ \pi/4 \ 0.1 \ 0]$ is

$$\mathbf{T}_c^b = \begin{bmatrix} 0.707 & -0.707 & 0 & 0.854 \\ 0.707 & 0.707 & 0 & 0.354 \\ 0 & 0 & 1 & 0.1 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Using (5.35) gives

$$\tilde{\mathbf{p}}^c = \begin{bmatrix} 0.066 \\ 0.141 \\ 0.8 \\ 1 \end{bmatrix}$$

and, in view of (5.41), it is

$$\mathbf{\Omega} = \begin{bmatrix} 633.6 & 0 & 250 \\ 0 & 964 & 250 \\ 0 & 0 & 1 \end{bmatrix}$$

and thus from (5.40)

$$x_I = 241.82$$

$$y_I = 335.92$$

$$\lambda = 0.8.$$

Hence

$$X_I = 302.27 \quad Y_I = 419.90.$$

Control Architecture

Solution to Problem 6.1

With reference to the hierarchical control architecture in Fig. 6.1, the solution can be articulated at the task and action levels as follows.

At the task level, in order to decompose the task into a sequence of actions, the decision module shall consult the knowledge base available in the modelling module concerning the manipulator and the environment. If the environment is scarcely structured, the knowledge base shall be updated with the information to be presented by the sensor module via the use of heteroceptive sensors, e.g., vision. Once the environment is structured, the modelling module provides the relative location between the manipulator, the object and the obstacle. Feasibility of the task shall be verified, and a possible sequence of actions is generated: approach position A , pick object, move object while avoiding obstacle, approach position B , place object. The two approach actions are opportune to ensure smooth pick-and-place of the object.

At the action level, the symbolic commands coming from the task level are interpreted by the decision module which shall take the resolutions needed to execute the given sequence of actions, that is, choice of reference frame, motion in operational space, intermediate path or via points C , D , E , F (see Fig. S6.1), duration of time intervals, path primitives off the obstacle, and type of interpolating functions. By doing so, the decision module shall consult the knowledge base available in the modelling module to verify path feasibility. As for the approach actions, the decisions imparted to the primitive level shall regard low values of velocities and vertical orientation of the end effector, whereas the knowledge base shall be updated with the information available in the sensor module via the use of range or proximity sensors.

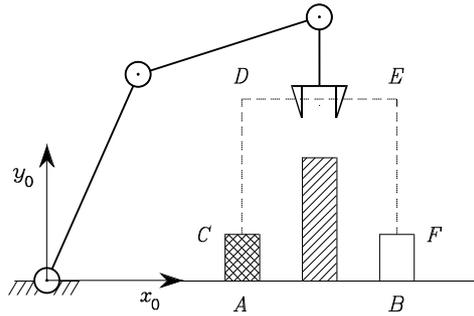


Fig. S6.1. Choice of reference frame and intermediate positions for an object pick-and-place task

Solution to Problem 6.2

With reference to the sequence of points illustrated in Fig. S6.1, the operational space motion primitives can be determined in the two cases of path points and via points, respectively, as follows.

In the case of path points, the various points can be connected by segments. To ensure safe grasp of the object at point C and safe release of the object at point F , it is advisable to choose null velocity and acceleration at each of those points. As for the intermediate points D and E , it is sufficient to impose null velocities. Hence, the timing law along the segments CD and EF shall be polynomials of at least fourth order, whereas the timing law along the segment DE shall be either a cubic polynomial or a trajectory with trapezoidal velocity profile.

In the case of via points, the actual path in the neighbourhood of D and E is of no concern, as long as it is kept at a safety distance from the obstacle. The passage in the proximity of the two via points can be obtained by using the same interpolating functions as for the previous case, on condition that the generation of the timing laws along the segments DE and EF are suitably anticipated in time. As above, null initial and final velocity and acceleration are to be imposed.

Finally, the presence of a redundant DOFs relative to a bidimensional position task can be exploited to keep end-effector orientation constant along $-y$.

Solution to Problem 6.3

Before executing the peg-in-hole task, the end effector shall approach the hole. It can be argued that it is difficult to guarantee exact positioning of the peg relative to the hole; this may render the insertion phase quite critical. It is assumed that the main cause of uncertainty regards only position, that is the axis of the peg can be considered to be aligned with the axis of the hole during

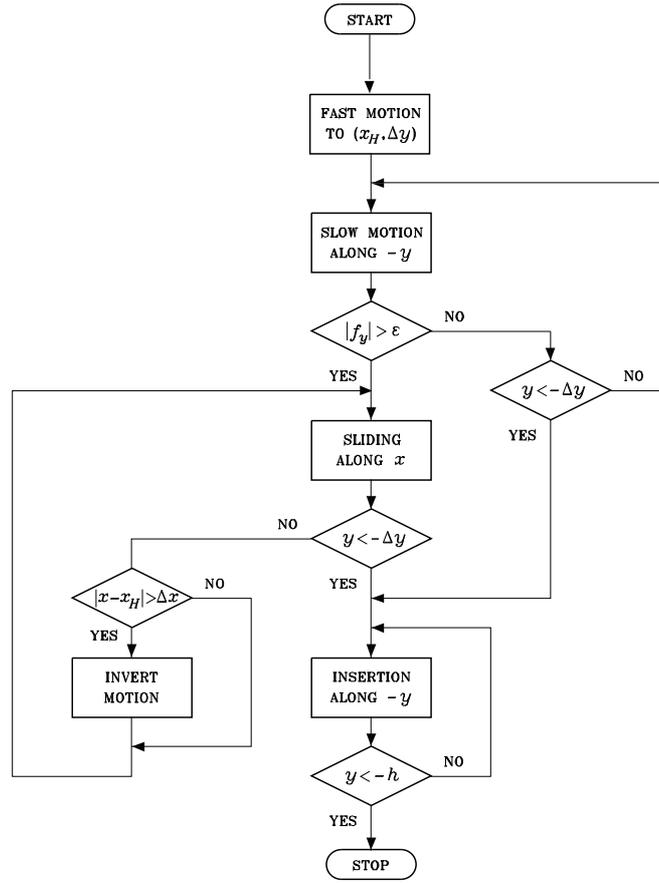


Fig. S6.2. Flow chart for the execution of a peg-in-hole task

the approach to the hole; also, it is assumed that no jamming occurs during insertion.

Choose a reference frame at the base of the arm, with axis x along the horizontal direction (positive from the base to the hole) and axis y along the positive vertical direction. A possible solution is to split the whole task into three subtasks as follows.

At first, a fast motion towards the position $(x_H, \Delta y)$ shall be executed, where x_H is the estimated x coordinate of the center of the hole and Δy is a small distance from the estimated height of the hole ($y = 0$). This is meant to avoid colliding the surface around the hole during the approach phase. A pure motion control strategy is needed to accomplish this subtask.

Next, a slow motion along $-y$ shall be executed, still with a pure motion control strategy. This goes on until y is below the given distance $-\Delta y$

(when the peg is considered to have entered the hole), unless a contact force f_y is sensed above a threshold ϵ , due to a collision with the surface. In that case, before proceeding with insertion, it is necessary to slide along the surface in either positive or negative horizontal direction until a force below the threshold is sensed; if the x coordinate of end-effector position goes too far from x_H (greater than a distance Δx), then the motion shall be inverted. A force/position control strategy is needed to accomplish this subtask where position is controlled along x and force is controlled along y to a desired pushing force. This goes on until the peg enters the hole.

Finally, the insertion along $-y$ is accomplished by using a force/position control strategy where position is controlled along y , force is controlled along x , and moment is controlled about z ; moment control is needed to keep the peg in axis with the hole during insertion.

The flow chart corresponding to the articulation of the whole task into the above three subtasks is illustrated in Fig. S6.2.

Solution to Problem 6.4

The task can be divided into the following sequence of actions:

- command new object on conveyor,
- fast motion of end effector to a location above object to pick,
- slow motion of end effector towards object,
- pick object from conveyor,
- fast motion of object to a loading location above pallet,
- slow motion of object downwards,
- release object.

At the beginning, the end effector is moved to a home location from which the task starts. The above sequence is iterated until the pallet is filled, and then the end effector is taken back to the home location.

Below a PASCAL program is listed to execute the given task.

```

program FILL_PALLET;
type  point: array[1..6] of real;
const home: point=(-----);      {end-effector home location}
      object: point=(-----);    {object location}
      pallet: point=(-----);    {lower-left pallet location}
      lift: -.-;                  {z-offset above object}
      length: -.-;                {object x-dimension}
      width: -.-;                 {object y-dimension}
      fast: -.-;                  {fast velocity}
      slow: -.-;                  {slow velocity}
var   i,j: integer;
      current: point;             {end-effector current location}
procedure MOVETO(var endpoint:point,velocity:real);
procedure GRASP;
procedure RELEASE;

```

```
procedure NEW_OBJECT;
begin
  MOVETO(home,fast);
  for i:=0 to 3 do
    for j:= 0 to 3 do
      begin
        NEW_OBJECT;
        current:=object;
        current[3]:=current[3]+lift;
        MOVETO(current,fast);
        MOVETO(object,slow);
        GRASP;
        MOVETO(current,slow);
        current:=pallet;
        current[1]:=current[1]+i*length;
        current[2]:=current[2]+j*width;
        current[3]:=current[3]+lift;
        MOVETO(current,fast);
        current[3]:=current[3]-lift;
        MOVETO(current,slow);
        RELEASE;
        current[3]:=current[3]+lift;
        MOVETO(current,slow);
      end
    end
  MOVETO(home,fast);
end
```


Dynamics

Solution to Problem 7.1

Consider the Cartesian arm in Fig. S7.1, for which the vector of generalized coordinates is $\mathbf{q} = [d_1 \ d_2]^T$.

Let m_{ℓ_1} , m_{ℓ_2} be the masses of the two links and m_{m_1} , m_{m_2} the masses of the rotors of the two joint motors. Let also I_{m_1} , I_{m_2} be the moments of inertia about the axes of the two rotors. It is assumed that $\mathbf{p}_{m_i} = \mathbf{p}_{i-1}$ and $\mathbf{z}_{m_i} = \mathbf{z}_{i-1}$, for $i = 1, 2$, i.e., the motors are located on the joint axes with centres of mass located at the origins of the respective frames.

With the choice of the frames in Fig. S7.1, it is $\mathbf{z}_1 = [1/\sqrt{2} \ 0 \ 1/\sqrt{2}]^T$. Hence, computation of the Jacobians in (7.16), (7.18) yields

$$\mathbf{J}_P^{(\ell_1)} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \quad \mathbf{J}_P^{(\ell_2)} = \begin{bmatrix} 0 & 1/\sqrt{2} \\ 0 & 0 \\ 1 & 1/\sqrt{2} \end{bmatrix}.$$

Obviously it is $\mathbf{J}_O^{(\ell_1)} = \mathbf{J}_O^{(\ell_2)} = \mathbf{O}$.

Computation of the Jacobians in (7.25), (7.26), (7.28), (7.29) respectively yields

$$\mathbf{J}_P^{(m_1)} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \mathbf{J}_P^{(m_2)} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}$$

$$\mathbf{J}_O^{(m_1)} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ k_{r1} & 0 \end{bmatrix} \quad \mathbf{J}_O^{(m_2)} = \begin{bmatrix} 0 & k_{r2}/\sqrt{2} \\ 0 & 0 \\ 0 & k_{r2}/\sqrt{2} \end{bmatrix}$$

where k_{r_i} is the gear reduction ratio of motor i .

From (7.32), the inertia matrix is

$$\mathbf{B} = \begin{bmatrix} m_{\ell_1} + m_{m_2} + k_{r1}^2 I_{m_1} + m_{\ell_2} & m_{\ell_2}/\sqrt{2} \\ m_{\ell_2}/\sqrt{2} & m_{\ell_2} + k_{r2}^2 I_{m_2} \end{bmatrix}.$$

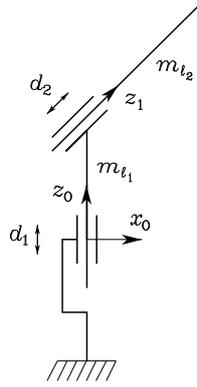


Fig. S7.1. Two-link Cartesian arm where Joint 2 axis forms an angle of $\pi/4$ with Joint 1 axis

Notice that \mathbf{B} is constant, which implies that $\mathbf{C} = \mathbf{O}$. As for the gravitational terms, with $\mathbf{g}_0 = [0 \ 0 \ -g]^T$, (7.39) gives:

$$g_1 = (m_{\ell_1} + m_{m_2} + m_{\ell_2})g \quad g_2 = m_{\ell_2}g/\sqrt{2}.$$

In the absence of friction and tip contact forces, the resulting equations of motion are

$$\begin{aligned} (m_{\ell_1} + m_{m_2} + k_{r1}^2 I_{m_1} + m_{\ell_2})\ddot{d}_1 + \frac{m_{\ell_2}}{\sqrt{2}}\ddot{d}_2 + (m_{\ell_1} + m_{m_2} + m_{\ell_2})g &= \tau_1 \\ \frac{m_{\ell_2}}{\sqrt{2}}\ddot{d}_1 + (m_{\ell_2} + k_{r2}^2 I_{m_2})\ddot{d}_2 + \frac{m_{\ell_2}}{\sqrt{2}}g &= \tau_2 \end{aligned}$$

where τ_1 and τ_2 denote the forces applied to the two joints. It is worth pointing out that, differently from the arm in Fig. 7.3, the dynamic model is coupled (nonnull off-diagonal terms in the inertia matrix) and gravity acts also on the second joint.

Solution to Problem 7.2

A natural choice for the elements of matrix \mathbf{C} satisfying (7.43) with h_{ijk} as in (7.41) is

$$c_{ij} = \sum_{k=1}^n \left(\frac{\partial b_{ij}}{\partial q_k} - \frac{1}{2} \frac{\partial b_{jk}}{\partial q_i} \right) \dot{q}_k.$$

For the two-link planar arm of Fig. 7.4, they become

$$c_{11} = \sum_{k=1}^2 \left(\frac{\partial b_{11}}{\partial q_k} - \frac{1}{2} \frac{\partial b_{1k}}{\partial q_1} \right) \dot{q}_k = -2m_{\ell_2}a_1\ell_2s_2\dot{\vartheta}_2$$

$$\begin{aligned}
c_{12} &= \sum_{k=1}^2 \left(\frac{\partial b_{12}}{\partial q_k} - \frac{1}{2} \frac{\partial b_{2k}}{\partial q_1} \right) \dot{q}_k = -m_{\ell_2} a_1 \ell_2 s_2 \dot{\vartheta}_2 \\
c_{21} &= \sum_{k=1}^2 \left(\frac{\partial b_{21}}{\partial q_k} - \frac{1}{2} \frac{\partial b_{1k}}{\partial q_2} \right) \dot{q}_k = m_{\ell_2} a_1 \ell_2 s_2 \dot{\vartheta}_1 - \frac{1}{2} m_{\ell_2} a_1 \ell_2 s_2 \dot{\vartheta}_2 \\
c_{22} &= \sum_{k=1}^2 \left(\frac{\partial b_{22}}{\partial q_k} - \frac{1}{2} \frac{\partial b_{2k}}{\partial q_2} \right) \dot{q}_k = \frac{1}{2} m_{\ell_2} a_1 \ell_2 s_2 \dot{\vartheta}_1.
\end{aligned}$$

Setting $h = -m_{\ell_2} a_1 \ell_2 s_2$ leads to

$$\mathbf{C} = \begin{bmatrix} 2h\dot{\vartheta}_2 & h\dot{\vartheta}_2 \\ -h\dot{\vartheta}_1 + \frac{h}{2}\dot{\vartheta}_2 & -\frac{h}{2}\dot{\vartheta}_1 \end{bmatrix}$$

while the time derivative of the inertia matrix can be written as

$$\dot{\mathbf{B}} = \begin{bmatrix} 2h\dot{\vartheta}_2 & h\dot{\vartheta}_2 \\ h\dot{\vartheta}_2 & 0 \end{bmatrix}.$$

Therefore, the matrix $\mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) = \dot{\mathbf{B}} - 2\mathbf{C}$ becomes

$$\mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} -2h\dot{\vartheta}_2 & -h\dot{\vartheta}_2 \\ 2h\dot{\vartheta}_1 & h\dot{\vartheta}_1 \end{bmatrix}$$

which clearly is not skew-symmetric. Nevertheless, it is easy to show that the product

$$\dot{\mathbf{q}}^T \mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} = 0,$$

confirming that (7.49) holds even for a choice of \mathbf{C} so that (7.48) does not hold.

Solution to Problem 7.3

Consider the SCARA arm in Fig. S7.1, for which the vector of generalized coordinates is $\mathbf{q} = [\vartheta_1 \ \vartheta_2 \ d_3 \ \vartheta_4]^T$ and link frames assigned as in Fig. S2.3. For the first two links, consider the same kinematics and dynamic parameters of the two-link planar arm of Example 7.2. Moreover, links 3 and 4 can be considered as a unique rigid body, referred as “last link”, which translates along axis z_2 of the prismatic Joint 3 and rotates about axis z_3 of the revolute Joint 4. Let m_{ℓ_3} denote the mass of the last link. For simplicity, it is assumed that the centre of mass \mathbf{p}_{ℓ_4} of the last link is located on axis z_3 ; let I_{ℓ_4} denote the element [3, 3] of the inertia tensor of the last link relative to the centre of mass. Also, assume that the motors of Joints 3 and 4 have negligible mass and inertia.

Due to the additivity property, the kinetic energy of the SCARA manipulator can be computed by adding the kinetic energy of the last link to the

kinetic energy of the two-link planar arm. Therefore, from (7.32), the inertia matrix can be computed as

$$\mathbf{B}(\mathbf{q}) = \mathbf{B}'(\mathbf{q}) + \mathbf{B}''(\mathbf{q})$$

where \mathbf{B}' is the (4×4) matrix obtained completing the (2×2) inertial matrix of the two-link planar arm of Sect. 7.3.2 with two null columns and two null rows, while \mathbf{B}'' is the contribution of the last link, which can be computed as

$$\mathbf{B}''(\mathbf{q}) = m_{\ell_3} \mathbf{J}_P^{(\ell_4)T} \mathbf{J}_P^{(\ell_4)} + \mathbf{J}_O^{(\ell_4)T} \mathbf{R}_4 \mathbf{I}_{\ell_4}^i \mathbf{R}_4^T \mathbf{J}_O^{(\ell_4)}. \quad (\text{S7.1})$$

The computation of the Jacobians in (7.16), (7.17) yields

$$\mathbf{J}_P^{(\ell_4)} = \begin{bmatrix} -a_1 s_1 - a_2 s_{12} & -a_2 s_{12} & 0 & 0 \\ a_1 c_1 + a_2 c_{12} & a_2 c_{12} & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \mathbf{J}_O^{(\ell_4)} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

Hence, from (S7.1), the nonnull elements of matrix $\mathbf{B}''(\mathbf{q})$ are

$$\begin{aligned} b''_{11} &= m_{\ell_3} (a_1^2 + a_2^2 + 2a_1 a_2 c_2) + I_{\ell_4} \\ b''_{12} &= b''_{21} = m_{\ell_3} (a_2^2 + a_1 a_2 c_2) + I_{\ell_4} \\ b''_{14} &= b''_{41} = I_{\ell_4} \\ b''_{22} &= m_{\ell_3} a_2^2 + I_{\ell_4} \\ b''_{24} &= b''_{42} = I_{\ell_4} \\ b''_{33} &= m_{\ell_3} \\ b''_{44} &= I_{\ell_4}. \end{aligned}$$

Therefore, the nonnull elements of the inertia matrix of the SCARA manipulator are:

$$\begin{aligned} b_{11} &= I_{\ell_1} + m_{\ell_1} \ell_1^2 + k_{r1}^2 I_{m_1} + I_{\ell_2} + m_{\ell_2} (a_1^2 + \ell_2^2 + 2a_1 \ell_2 c_2) \\ &\quad + I_{m_2} + m_{m_2} a_1^2 + m_{\ell_3} (a_1^2 + a_2^2 + 2a_1 a_2 c_2) + I_{\ell_4} \\ b_{12} &= b_{21} = I_{\ell_2} + m_{\ell_2} (\ell_2^2 + a_1 \ell_2 c_2) + k_{r2} I_{m_2} + m_{\ell_3} (a_2^2 + a_1 a_2 c_2) + I_{\ell_4} \\ b_{14} &= b_{41} = I_{\ell_4} \\ b_{22} &= I_{\ell_2} + m_{\ell_2} \ell_2^2 + k_{r2}^2 I_{m_2} + m_{\ell_3} a_2^2 + I_{\ell_4} \\ b_{24} &= b_{42} = I_{\ell_4} \\ b_{33} &= m_{\ell_3} \\ b_{44} &= I_{\ell_4}. \end{aligned}$$

The Christoffel symbols for the SCARA manipulator can be computed as

$$c_{ijk} = c'_{ijk} + c''_{ijk}$$

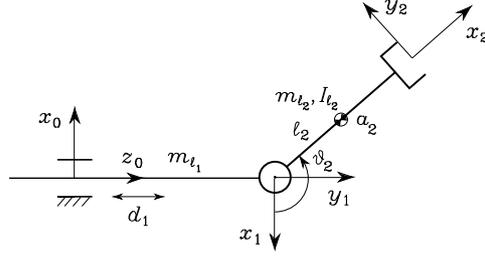


Fig. S7.2. Two-link planar arm with a prismatic joint and a revolute joint

where c'_{ijk} are the Christoffel symbols of the two-link planar arm of Sect. 7.3.2, while c''_{ijk} are those computed from the elements b''_{ij} of $\mathbf{B}''(\mathbf{q})$ as in (7.45). The nonnull elements are:

$$\begin{aligned} c_{112} &= -m_{\ell_2} a_1 \ell_2 s_2 - m_{\ell_3} a_1 a_2 s_2 = k \\ c_{122} &= k \\ c_{211} &= -k, \end{aligned}$$

leading to the matrix

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} k\dot{\vartheta}_2 & k(\dot{\vartheta}_1 + \dot{\vartheta}_2) & 0 & 0 \\ -k\dot{\vartheta}_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

As for the gravitational terms, it can be easily understood that gravity contributes only to Joint 3 force, which has the expression

$$g_3 = m_{\ell_3} g.$$

Solution to Problem 7.4

By using the expressions of the parameters in (7.83), the equations of motion (7.82) can be rewritten as

$$\begin{aligned} \tau_1 &= (a_1^2 \pi_1 + 2a_1 \pi_2 + \pi_3 + k_r^2 \pi_4 + (a_1^2 + 2a_1 a_2 c_2 + a_2^2) \pi_5 \\ &\quad + (2a_1 c_2 + 2a_2) \pi_6 + \pi_7 + \pi_8) \ddot{\vartheta}_1 \\ &\quad + ((a_1 a_2 c_2 + a_2^2) \pi_5 + (a_1 c_2 + 2a_2) \pi_6 + \pi_7 + k_r^2 \pi_8) \ddot{\vartheta}_2 \\ &\quad - (2a_1 a_2 s_2 \pi_5 + 2a_1 s_2 \pi_6) \dot{\vartheta}_1 \dot{\vartheta}_2 - (a_1 a_2 s_2 \pi_5 + a_1 s_2 \pi_6) \dot{\vartheta}_2^2 \\ &\quad + (a_1 \pi_1 + \pi_2 + a_1 \pi_5) g c_1 + (a_2 \pi_5 + \pi_6) g c_{12} \\ \tau_2 &= (a_2^2 \pi_5 + 2a_2 \pi_6 + \pi_7 + k_r^2 \pi_8) \ddot{\vartheta}_2 \\ &\quad + ((a_1 a_2 c_2 + a_2^2) \pi_5 + (a_1 c_2 + 2a_2) \pi_6 + \pi_7 + k_r^2 \pi_8) \ddot{\vartheta}_1 \\ &\quad + (a_1 a_2 s_2 \pi_5 + a_1 s_2 \pi_6) \dot{\vartheta}_1^2 \\ &\quad + (a_2 \pi_5 + \pi_6) g c_{12}. \end{aligned}$$

It is then possible to find the following linear combinations between the parameters which bring the number thereof down to a minimum of 5:

$$\begin{aligned}
\pi'_1 &= a_1\pi_1 + \pi_2 + a_1\pi_5 = a_1m_1 + m_1\ell_{C_1} + a_1m_2 \\
\pi'_2 &= a_1\pi_2 + \pi_3 + k_{r1}^2\pi_4 = a_1m_1\ell_{C_1} + \bar{I}_1 + k_{r1}^2I_{m_1} \\
\pi'_3 &= a_2\pi_5 + \pi_6 = a_2m_2 + m_2\ell_{C_2} \\
\pi'_4 &= a_2\pi_6 + \pi_7 = a_2m_2\ell_{C_2} + \bar{I}_2 \\
\pi'_5 &= \pi_8 = I_{m_2}.
\end{aligned} \tag{S7.2}$$

Accordingly, the elements of the regressor become

$$\begin{aligned}
y'_{11} &= a_1\ddot{\vartheta}_1 + gc_1 \\
y'_{12} &= \ddot{\vartheta}_1 \\
y'_{13} &= (2a_1c_2 + a_2)\ddot{\vartheta}_1 + (a_1c_2 + a_2)\ddot{\vartheta}_2 - 2a_1s_2\dot{\vartheta}_1\dot{\vartheta}_2 - a_1s_2\dot{\vartheta}_2^2 + gc_{12} \\
y'_{14} &= \ddot{\vartheta}_1 + \ddot{\vartheta}_2 \\
y'_{15} &= \ddot{\vartheta}_1 + k_{r2}\ddot{\vartheta}_2 \\
y'_{21} &= 0 \\
y'_{22} &= 0 \\
y'_{23} &= a_2\ddot{\vartheta}_2 + (a_1c_2 + a_2)\ddot{\vartheta}_1 + a_1s_2\dot{\vartheta}_1^2 + gc_{12} \\
y'_{24} &= \ddot{\vartheta}_2 + \ddot{\vartheta}_1 \\
y'_{25} &= k_{r2}^2\ddot{\vartheta}_2 + k_{r2}\ddot{\vartheta}_1.
\end{aligned} \tag{S7.3}$$

Solution to Problem 7.5

Consider the planar arm in Fig. S7.2, for which the vector of generalized coordinates is $\mathbf{q} = [d_1 \ \vartheta_2]^T$.

Let m_{ℓ_1} , m_{ℓ_2} be the masses of the two links, ℓ_1 , ℓ_2 the distances of the centres of mass from the respective joint axes, and m_{m_1} , m_{m_2} the masses of the rotors of the two joint motors. Let also \mathbf{I}_{ℓ_2} be the inertia tensor of Link 2 about Joint 2 axis, and I_{m_1} , I_{m_2} the moments of inertia about the axes of the two rotors. It is assumed that $\mathbf{p}_{m_i} = \mathbf{p}_{i-1}$ and $\mathbf{z}_{m_i} = \mathbf{z}_{i-1}$, for $i = 1, 2$, i.e., the motors are located on the joint axes with centres of mass located at the origins of the respective frames.

With the choice of the frames in Fig. S7.2, it is $\mathbf{z}_1 = [0 \ 1 \ 0]^T$, $\mathbf{p}_1 = \mathbf{p}_{m_2} = [0 \ 0 \ d_1]^T$, $\mathbf{p}_{\ell_2} = [-\ell_2c_2 \ 0 \ d_1 + \ell_2s_2]^T$. Computation of the Jacobians in (7.16), (7.18) yields

$$\mathbf{J}_P^{(\ell_1)} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \quad \mathbf{J}_P^{(\ell_2)} = \begin{bmatrix} 0 & \ell_2s_2 \\ 0 & 0 \\ 1 & \ell_2c_2 \end{bmatrix},$$

whereas computation of the Jacobians in (7.17), (7.19) yields

$$\mathbf{J}_O^{(\ell_1)} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \mathbf{J}_O^{(\ell_2)} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}.$$

Computation of the Jacobians in (7.25), (7.26) yields

$$\mathbf{J}_P^{(m_1)} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \mathbf{J}_P^{(m_2)} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix},$$

whereas computation of the Jacobians in (7.28), (7.29) yields

$$\mathbf{J}_O^{(m_1)} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ k_{r1} & 0 \end{bmatrix} \quad \mathbf{J}_O^{(m_2)} = \begin{bmatrix} 0 & 0 \\ 0 & k_{r2} \\ 0 & 0 \end{bmatrix}$$

where k_{r_i} is the gear reduction ratio of motor i .

From (7.32), the inertia matrix is

$$\mathbf{B}(\mathbf{q}) = \begin{bmatrix} b_{11} & b_{12}(\vartheta_2) \\ b_{12}(\vartheta_2) & b_{22} \end{bmatrix}$$

$$b_{11} = m_{\ell_1} + m_{m_2} + m_{\ell_2} + k_{r1}^2 I_{m_1}$$

$$b_{12} = m_{\ell_2} \ell_2 c_2$$

$$b_{22} = I_{\ell_2} + m_{\ell_2} \ell_2^2 + k_{r2}^2 I_{m_2}$$

where $I_{\ell_2} = I_{\ell_2 zz}$ is the constant moment of inertia about Joint 2 axis.

Computation of Christoffel symbols as in (7.45) gives:

$$c_{111} = c_{112} = c_{121} = c_{211} = c_{212} = c_{221} = c_{222} = 0$$

$$c_{122} = \frac{\partial b_{12}}{\partial q_2} - \frac{1}{2} \frac{\partial b_{22}}{\partial q_1} = -m_{\ell_2} \ell_2 s_2$$

leading to the matrix

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} 0 & -m_{\ell_2} \ell_2 s_2 \dot{\vartheta}_2 \\ 0 & 0 \end{bmatrix}.$$

Computing the matrix \mathbf{N} in (7.47) gives

$$\mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) = \dot{\mathbf{B}}(\mathbf{q}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$$

$$= \begin{bmatrix} 0 & m_{\ell_2} \ell_2 s_2 \dot{\vartheta}_2 \\ -m_{\ell_2} \ell_2 s_2 \dot{\vartheta}_2 & 0 \end{bmatrix}$$

which is skew-symmetric.

As for the gravitational terms, with $\mathbf{g}_0 = [-g \ 0 \ 0]^T$, (7.39) gives:

$$\begin{aligned} g_1 &= 0 \\ g_2 &= m_{\ell_2} \ell_2 g s_2. \end{aligned}$$

In the absence of friction and tip contact forces, the resulting equations of motion are

$$(m_{\ell_1} + m_{m_2} + m_{\ell_2} + k_{r1}^2 I_{m_1}) \ddot{d}_1 + m_{\ell_2} \ell_2 c_2 \ddot{\vartheta}_2 - m_{\ell_2} \ell_2 s_2 \dot{\vartheta}_2^2 = \tau_1 \quad (\text{S7.4})$$

$$m_{\ell_2} \ell_2 c_2 \ddot{d}_1 + (I_{\ell_2} + m_{\ell_2} \ell_2^2 + k_{r2}^2 I_{m_2}) \ddot{\vartheta}_2 + m_{\ell_2} \ell_2 g s_2 = \tau_2 \quad (\text{S7.5})$$

where τ_1 and τ_2 respectively denote the force applied to Joint 1 and the torque applied to Joint 2.

The presence of a concentrated tip payload of mass m_L alters Link 2 parameters as follows:

$$m'_{\ell_2} = m_{\ell_2} + m_L \quad \ell'_2 = \frac{m_{\ell_2} \ell_2 + m_L a_2}{m'_{\ell_2}}$$

$$I'_{\ell_2} = I_{\ell_2} + m_{\ell_2} (\ell'_2 - \ell_2)^2 + m_L (a_2 - \ell'_2)^2$$

and the same formal dynamic model can be written as above in terms of the new parameters.

A minimum parameterization of the dynamic model can be obtained in terms of the following parameter vector:

$$\begin{aligned} \boldsymbol{\pi}' &= [\pi'_1 \ \pi'_2 \ \pi'_3]^T \\ \pi'_1 &= m_{\ell_1} + m_{m_2} + m'_{\ell_2} + k_{r1}^2 I_{m_1} \\ \pi'_2 &= m'_{\ell_2} \ell'_2 \\ \pi'_3 &= I'_{\ell_2} + m'_{\ell_2} \ell'^2_2 + k_{r2}^2 I_{m_2}. \end{aligned}$$

Accordingly, the elements of the regressor are

$$\begin{aligned} y'_{11} &= \ddot{d}_1 \\ y'_{12} &= c_2 \ddot{\vartheta}_2 - s_2 \dot{\vartheta}_2^2 \\ y'_{13} &= 0 \\ y'_{21} &= 0 \\ y'_{22} &= c_2 \ddot{d}_1 + g s_2 \\ y'_{23} &= \ddot{\vartheta}_2. \end{aligned}$$

Solution to Problem 7.6

Let x_1 and x_2 denote the absolute angles of the two joints. The relationship between relative and absolute angles is

$$x_1 = \vartheta_1 \quad x_2 = \vartheta_1 + \vartheta_2.$$

This can be formally treated as a coordinate transformation from the joint space to the operational space of absolute angles. The analytical Jacobian characterizing such transformation at velocity level is then

$$\mathbf{J}_A = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix},$$

which is always nonsingular, implying that the absolute angles constitute a set of generalized coordinates for the system.

Therefore, the terms of the operational space dynamic model can be computed according to (7.134)–(7.136). From (7.134), the inertia matrix is

$$\mathbf{B}_A = \begin{bmatrix} b_{A11} & b_{A12} \\ b_{A12} & b_{A22} \end{bmatrix}$$

where, using the expressions of b_{ij} in Sect. 7.3.2, the elements in the new coordinates are

$$\begin{aligned} b_{A11} &= b_{11} + b_{22} - 2b_{12} \\ &= I_{\ell_1} + m_{\ell_1} \ell_1^2 + k_{r1}^2 I_{m_1} + m_{\ell_2} a_1^2 + (1 - k_{r2})^2 I_{m_2} + m_{m_2} a_1^2 \\ b_{A12} &= b_{12} - b_{22} \\ &= m_{\ell_2} a_1 \ell_2 \cos(x_2 - x_1) + k_{r2} (1 - k_{r2}) I_{m_2} \\ b_{A22} &= b_{22} \\ &= I_{\ell_2} + m_{\ell_2} \ell_2^2 + k_{r2}^2 I_{m_2}. \end{aligned}$$

Notice that the diagonal elements are configuration-invariant; also, if $k_{r2} = 1$ (direct drive), then the effect of motor 2 inertia on both b_{A12} and b_{A22} vanishes.

Further, from (7.135), the velocity-dependent term is

$$\mathbf{C}_A \dot{\mathbf{x}} = \begin{bmatrix} c_{11} - c_{12} & c_{21} - c_{22} \\ c_{21} & c_{22} \end{bmatrix}$$

which, using the expressions of c_{ij} in Sect. 4.3.2, in the new coordinates becomes

$$\mathbf{C}_A \dot{\mathbf{x}} = \begin{bmatrix} h \dot{x}_2^2 \\ -h \dot{x}_1^2 \end{bmatrix}$$

where $h = -m_{\ell_2} a_1 \ell_2 \sin(x_2 - x_1)$. Interestingly enough, there is no Coriolis effect; this can be explained by observing that rotation of Link 2 is represented with reference to Frame 0 and thus is independent of rotation of Link 1.

Finally, from (7.136), the gravity term is

$$\mathbf{g}_A = \begin{bmatrix} g_1 - g_2 \\ g_2 \end{bmatrix}$$

which, using the expressions of g_i in Sect. 4.3.2, in the new coordinates becomes

$$\mathbf{g}_A = \begin{bmatrix} (m_{\ell_1} \ell_1 + m_{m_2} a_1 + m_{\ell_2} a_1) g \cos x_1 \\ m_{\ell_2} \ell_2 g \cos x_2 \end{bmatrix}.$$

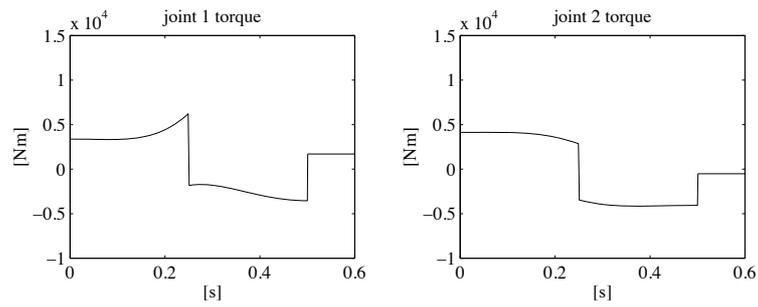


Fig. S7.3. Time history of joint torques for the first trajectory of Example 7.2

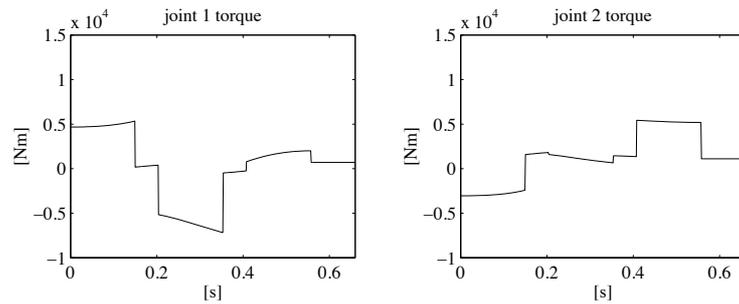


Fig. S7.4. Time history of joint torques for the second trajectory of Example 7.2

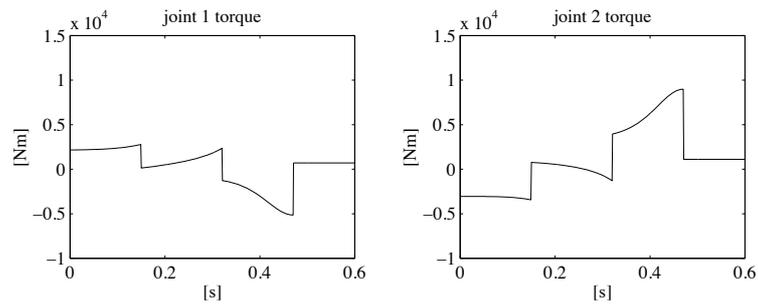


Fig. S7.5. Time history of joint torques for the third trajectory of Example 7.2

Solution to Problem 7.7

The various terms of the dynamic model are computed by using the minimum parameterization in the solution to Problem 7.4. The tip forces can be incorporated in the model as in (7.42), i.e., via the Jacobian transpose.

The files with the solution can be found in Folder 7_7.

The resulting torques for the three trajectories of Example 7.2 are illustrated in Figs. S7.3, S7.4, S7.5, respectively.

Solution to Problem 7.8

Start by imposing the initial conditions for the velocities and accelerations:

$$\ddot{\mathbf{p}}_0^0 - \mathbf{g}_0^0 = [g \ 0 \ 0]^T \quad \boldsymbol{\omega}_0^0 = \dot{\boldsymbol{\omega}}_0^0 = \mathbf{0}.$$

All quantities shall be referred to the current link frame. With the choice of the frames as in Fig. S7.2, the following vectors are obtained:

$$\mathbf{r}_{1,C_1}^1 = \begin{bmatrix} 0 \\ \ell_{C_1} \\ 0 \end{bmatrix} \quad \mathbf{r}_{0,1}^1 = \begin{bmatrix} 0 \\ d_1 \\ 0 \end{bmatrix} \quad \mathbf{r}_{2,C_2}^2 = \begin{bmatrix} \ell_{C_2} \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{r}_{1,2}^2 = \begin{bmatrix} a_2 \\ 0 \\ 0 \end{bmatrix}$$

where ℓ_{C_1} and ℓ_{C_2} are both negative quantities. The rotation matrices needed for vector transformation from one frame to another are:

$$\mathbf{R}_1^0 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \mathbf{R}_2^1 = \begin{bmatrix} c_2 & -s_2 & 0 \\ s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Further it is assumed that the axes of rotations of the rotors coincide with the respective joint axes, i.e., $\mathbf{z}_{m_i}^{i-1} = \mathbf{z}_0 = [0 \ 0 \ 1]^T$ for $i = 1, 2$.

According to (7.107)–(7.114), the Newton-Euler algorithm requires the execution of the following steps.

- Forward recursion: Link 1

$$\boldsymbol{\omega}_1^1 = \mathbf{0}$$

$$\dot{\boldsymbol{\omega}}_1^1 = \mathbf{0}$$

$$\ddot{\mathbf{p}}_1^1 = \begin{bmatrix} -g \\ \ddot{d}_1 \\ 0 \end{bmatrix}$$

$$\ddot{\mathbf{p}}_{C_1}^1 = \begin{bmatrix} -g \\ \ddot{d}_1 \\ 0 \end{bmatrix}$$

$$\dot{\boldsymbol{\omega}}_{m_1}^0 = \begin{bmatrix} 0 \\ 0 \\ k_{r1}\ddot{d}_1 \end{bmatrix}.$$

- Forward recursion: Link 2

$$\boldsymbol{\omega}_2^2 = \begin{bmatrix} 0 \\ 0 \\ \dot{\vartheta}_2 \end{bmatrix}$$

$$\dot{\omega}_2^2 = \begin{bmatrix} 0 \\ 0 \\ \ddot{\vartheta}_2 \end{bmatrix}$$

$$\ddot{\mathbf{p}}_2^2 = \begin{bmatrix} s_2 \ddot{d}_1 - a_2 \dot{\vartheta}_2^2 - g c_2 \\ c_2 \ddot{d}_1 + a_2 \ddot{\vartheta}_2 + g s_2 \\ 0 \end{bmatrix}$$

$$\ddot{\mathbf{p}}_{C_2}^2 = \begin{bmatrix} s_2 \ddot{d}_1 - (\ell_{C_2} + a_2) \dot{\vartheta}_2^2 - g c_2 \\ c_2 \ddot{d}_1 + (\ell_{C_2} + a_2) \ddot{\vartheta}_2 + g s_2 \\ 0 \end{bmatrix}$$

$$\dot{\omega}_{m_2}^1 = \begin{bmatrix} 0 \\ 0 \\ k_{r_2} \ddot{\vartheta}_2 \end{bmatrix}.$$

- Backward recursion: Link 2

$$\mathbf{f}_2^2 = \begin{bmatrix} m_2 s_2 \ddot{d}_1 - m_2 (\ell_{C_2} + a_2) \dot{\vartheta}_2^2 - m_2 g c_2 \\ m_2 c_2 \ddot{d}_1 + m_2 (\ell_{C_2} + a_2) \ddot{\vartheta}_2 + m_2 g s_2 \\ 0 \end{bmatrix}$$

$$\boldsymbol{\mu}_2^2 = \begin{bmatrix} * \\ * \\ m_2 (\ell_{C_2} + a_2) c_2 \ddot{d}_1 + (\bar{I}_{2zz} + m_2 (\ell_{C_2} + a_2)^2) \ddot{\vartheta}_2 + m_2 (\ell_{C_2} + a_2) g s_2 \end{bmatrix}$$

$$\tau_2 = m_2 (\ell_{C_2} + a_2) c_2 \ddot{d}_1 + (\bar{I}_{2zz} + m_2 (\ell_{C_2} + a_2)^2 + k_{r_2}^2 I_{m_2}) \ddot{\vartheta}_2 + m_2 (\ell_{C_2} + a_2) g s_2,$$

where the components marked by the symbol ‘*’ have not been computed, since they are not related to the joint torque τ_2 .

- Backward recursion: Link 1

$$\mathbf{f}_1^1 = \begin{bmatrix} -m_2 (\ell_{C_2} + a_2) s_2 \ddot{\vartheta}_2 - m_2 (\ell_{C_2} + a_2) c_2 \dot{\vartheta}_2^2 - (m_1 + m_2) g \\ (m_1 + m_2) \ddot{d}_1 + m_2 (\ell_{C_2} + a_2) c_2 \ddot{\vartheta}_2 - m_2 (\ell_{C_2} + a_2) s_2 \dot{\vartheta}_2^2 \\ 0 \end{bmatrix}$$

$$\tau_1 = (m_1 + m_2 + k_{r_1}^2 I_{m_1}) \ddot{d}_1 + m_2 (\ell_{C_2} + a_2) c_2 \ddot{\vartheta}_2 - m_2 (\ell_{C_2} + a_2) s_2 \dot{\vartheta}_2^2.$$

In view of the following equalities:

$$m_1 = m_{\ell_1} + m_{m_2} \quad m_2 = m_{\ell_2} \quad \ell_2 = \ell_{C_2} + a_2 \quad \bar{I}_{2zz} = I_{\ell_2},$$

the above dynamic model coincides with the model derived in (S7.4), (S7.5), with Lagrange formulation.

Solution to Problem 7.9

In order to differentiate matrix \mathbf{B}_A in (7.130) with respect to time, it is worth recalling the formula for the derivative of the inverse of a matrix \mathbf{A}

$$\frac{d}{dt}\mathbf{A}^{-1}(t) = -\mathbf{A}^{-1}(t)\dot{\mathbf{A}}(t)\mathbf{A}^{-1}(t).$$

Therefore, the time derivative of \mathbf{B}_A can be expressed as

$$\dot{\mathbf{B}}_A = -\mathbf{B}_A \left(\dot{\mathbf{J}}_A \mathbf{B}^{-1} \mathbf{J}_A^T - \mathbf{J}_A \mathbf{B}^{-1} \dot{\mathbf{B}} \mathbf{B}^{-1} \mathbf{J}_A^T + \mathbf{J}_A \mathbf{B}^{-1} \dot{\mathbf{J}}_A^T \right) \mathbf{B}_A.$$

As regards \mathbf{C}_A , in view of (3.62), the relation in (7.131) can be written as

$$\mathbf{C}_A \mathbf{J}_A \dot{\mathbf{q}} = \mathbf{B}_A \mathbf{J}_A \mathbf{B}^{-1} \mathbf{C} \dot{\mathbf{q}} - \mathbf{B}_A \dot{\mathbf{J}}_A \dot{\mathbf{q}},$$

and using a right pseudo-inverse of \mathbf{J}_A , e.g., the one in (7.138), leads to

$$\mathbf{C}_A = \mathbf{B}_A \mathbf{J}_A \mathbf{B}^{-1} \mathbf{C} \bar{\mathbf{J}}_A - \mathbf{B}_A \dot{\mathbf{J}}_A \bar{\mathbf{J}}_A.$$

Hence, the matrix $\dot{\mathbf{B}}_A - 2\mathbf{C}_A$ can be expressed as

$$\dot{\mathbf{B}}_A - 2\mathbf{C}_A = \bar{\mathbf{J}}_A^T (\dot{\mathbf{B}} - 2\mathbf{C}) \bar{\mathbf{J}}_A + \mathbf{B}_A \dot{\mathbf{J}}_A \bar{\mathbf{J}}_A - \bar{\mathbf{J}}_A^T \dot{\mathbf{J}}_A^T \mathbf{B}_A.$$

From the skew-symmetry of $\dot{\mathbf{B}} - 2\mathbf{C}$ it follows that the first term represents a skew-symmetric matrix. Also, by observing that the second term on the right-hand side is the transpose of the third term, the matrix resulting from the difference of those terms is skew-symmetric too. It can be concluded that the whole matrix $\dot{\mathbf{B}}_A - 2\mathbf{C}_A$ is skew-symmetric.

Solution to Problem 7.10

For a given $(r \times 1)$ vector of end-effector forces $\boldsymbol{\gamma}_A$, the problem can be formulated as that to find the $(n \times 1)$ vector $\boldsymbol{\tau}$ that satisfies the linear equation

$$\bar{\mathbf{J}}_A^T \boldsymbol{\tau} = \boldsymbol{\gamma}_A, \quad (\text{S7.6})$$

where $\bar{\mathbf{J}}_A$ is given in (7.138), and minimize the quadratic cost functional

$$g(\boldsymbol{\tau}) = \frac{1}{2}(\boldsymbol{\tau} - \boldsymbol{\tau}_a)^T \mathbf{B}^{-1}(\boldsymbol{\tau} - \boldsymbol{\tau}_a),$$

where $\boldsymbol{\tau}_a$ is an $(n \times 1)$ vector of arbitrary torques.

This problem can be solved with the method of Lagrangian multipliers. Consider the modified cost functional

$$g(\boldsymbol{\tau}, \boldsymbol{\lambda}) = \frac{1}{2}(\boldsymbol{\tau} - \boldsymbol{\tau}_a)^T \mathbf{B}^{-1}(\boldsymbol{\tau} - \boldsymbol{\tau}_a) + \boldsymbol{\lambda}^T (\boldsymbol{\gamma}_A - \bar{\mathbf{J}}_A^T \boldsymbol{\tau})$$

where $\boldsymbol{\lambda}$ is an $(r \times 1)$ vector of Lagrangian multipliers. The solution has to satisfy the necessary conditions:

$$\left(\frac{\partial g}{\partial \boldsymbol{\tau}} \right)^T = \mathbf{0} \quad \left(\frac{\partial g}{\partial \boldsymbol{\lambda}} \right)^T = \mathbf{0}.$$

From the first condition, it is

$$\boldsymbol{\tau} = \mathbf{B} \bar{\mathbf{J}}_A \boldsymbol{\lambda} + \boldsymbol{\tau}_a, \quad (\text{S7.7})$$

while the second condition gives the constraint (S7.6). Substituting (S7.7) in (S7.6) yields

$$\boldsymbol{\lambda} = (\bar{\mathbf{J}}_A^T \mathbf{B} \bar{\mathbf{J}}_A)^{-1} (\boldsymbol{\gamma}_A - \bar{\mathbf{J}}_A^T \boldsymbol{\tau}_a).$$

Then, substituting $\boldsymbol{\lambda}$ back in (S7.7) gives

$$\boldsymbol{\tau} = \mathbf{B} \bar{\mathbf{J}}_A (\bar{\mathbf{J}}_A^T \mathbf{B} \bar{\mathbf{J}}_A)^{-1} \boldsymbol{\gamma}_A + \left(\mathbf{I} - \mathbf{B} \bar{\mathbf{J}}_A (\bar{\mathbf{J}}_A^T \mathbf{B} \bar{\mathbf{J}}_A)^{-1} \bar{\mathbf{J}}_A^T \right) \boldsymbol{\tau}_a.$$

In view of (7.138) and (7.130), it is

$$\bar{\mathbf{J}}_A = \mathbf{B}^{-1} \mathbf{J}_A^T (\mathbf{J}_A \mathbf{B}^{-1} \mathbf{J}_A^T)^{-1},$$

and thus the above expression can be simplified into

$$\boldsymbol{\tau} = \mathbf{J}_A^T(\mathbf{q}) \boldsymbol{\gamma}_A + (\mathbf{I} - \mathbf{J}_A^T(\mathbf{q}) \bar{\mathbf{J}}_A^T(\mathbf{q})) \boldsymbol{\tau}_a$$

which is the sought solution. Notice that \mathbf{J}_A^T is a right pseudo-inverse of $\bar{\mathbf{J}}_A^T$ weighted by \mathbf{B}^{-1} , or else $\bar{\mathbf{J}}_A$ is a right pseudo-inverse of \mathbf{J}_A weighted by \mathbf{B} .

Solution to Problem 7.11

For a nonredundant manipulator, the core of the dynamic manipulability ellipsoid is given as in (7.149) which can be written as

$$\mathbf{J}^{-T}(\mathbf{q}) \mathbf{B}^T(\mathbf{q}) \mathbf{B}(\mathbf{q}) \mathbf{J}^{-1}(\mathbf{q}) = (\mathbf{J}(\mathbf{q}) \mathbf{B}^{-1}(\mathbf{q}) \mathbf{B}^{-T}(\mathbf{q}) \mathbf{J}^T(\mathbf{q}))^{-1}.$$

By analogy to the velocity manipulability ellipsoid (3.123), the dynamic manipulability measure can be defined as

$$w'(\mathbf{q}) = \sqrt{\det(\mathbf{J}(\mathbf{q}) \mathbf{B}^{-1}(\mathbf{q}) \mathbf{B}^{-T}(\mathbf{q}) \mathbf{J}^T(\mathbf{q}))}.$$

Since \mathbf{J} and \mathbf{B} are square matrices and $\det(\mathbf{B}) > 0$ for all \mathbf{q} , then

$$w'(\mathbf{q}) = \frac{|\det(\mathbf{J}(\mathbf{q}))|}{\det(\mathbf{B}(\mathbf{q}))} = \frac{w(\mathbf{q})}{\det(\mathbf{B}(\mathbf{q}))}$$

with $w(\mathbf{q})$ as in (3.56).

Motion Control

Solution to Problem 8.1

The transfer function of the forward path of the scheme in Fig. 5.10 is

$$P(s) = C_P(s)M(s) = \frac{k_m K_P (1 + sT_P)}{s^2(1 + sT_m)},$$

being $C_P(s)$ the transfer function of the PI position controller and

$$M(s) = \frac{k_m}{s(1 + sT_m)}.$$

The transfer function of the return path is

$$H(s) = k_{TP}.$$

It follows that the closed-loop input/output transfer function is

$$\frac{\Theta_m(s)}{\Theta_r(s)} = \frac{P(s)}{1 + P(s)H(s)} = \frac{\frac{1}{k_{TP}}}{1 + \frac{s^2(1 + sT_m)}{k_m K_P k_{TP}(1 + sT_P)}},$$

while the closed-loop disturbance/output transfer function is

$$\frac{\Theta_m(s)}{D(s)} = -\frac{M(s)}{1 + P(s)H(s)} = -\frac{\frac{sR_a}{k_t K_P k_{TP}(1 + sT_P)}}{1 + \frac{s^2(1 + sT_m)}{k_m K_P k_{TP}(1 + sT_P)}}.$$

Solution to Problem 8.2

Operating by the rules for block reduction on the scheme in Fig. 5.11, it is worth moving the input to the block k_{TV} onto Θ_m and subtracting the

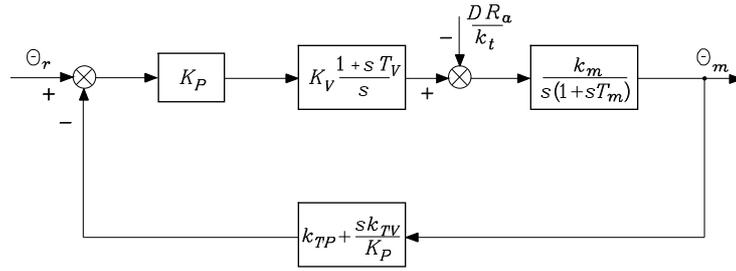


Fig. S8.1. Reduction on the block scheme of position and velocity feedback control

output to Θ_r . Then, the reduced block scheme of the system becomes that in Fig. S8.1.

The transfer function of the forward path is then

$$P(s) = K_P C_V(s) M(s) = \frac{k_m K_P K_V (1 + sT_V)}{s^2 (1 + sT_m)},$$

being $C_V(s)$ the transfer function of the PI velocity controller and $M(s)$ the transfer function of the actuator. The transfer function of the return path is

$$H(s) = k_{TP} \left(1 + s \frac{k_{TV}}{K_P k_{TP}} \right).$$

It follows that the closed-loop input/output transfer function with $T_V = T_m$ is

$$\frac{\Theta_m(s)}{\Theta_r(s)} = \frac{P(s)}{1 + P(s)H(s)} = \frac{\frac{1}{k_{TP}}}{1 + \frac{sk_{TV}}{K_P k_{TP}} + \frac{s^2}{k_m K_P k_{TP} K_V}}.$$

The closed-loop disturbance/output transfer function is

$$\frac{\Theta(s)}{D(s)} = -\frac{M(s)}{1 + P(s)H(s)} = -\frac{\frac{sR_a}{k_t K_P k_{TP} K_V (1 + sT_m)}}{1 + \frac{sk_{TV}}{K_P k_{TP}} + \frac{s^2}{k_m K_P k_{TP} K_V}}.$$

Solution to Problem 8.3

First, it is worth observing that the block scheme of Fig. 8.9 has been obtained from the scheme of Fig. 8.6 operating by the usual rules for block reduction. In particular, the input to the block k_{TA} has been moved onto $\dot{\Theta}_m$, while the output to the same block has been moved onto the output of $C_A(s)$. Likewise, the input to the block k_{TV} has been moved onto Θ_m , while the output to the

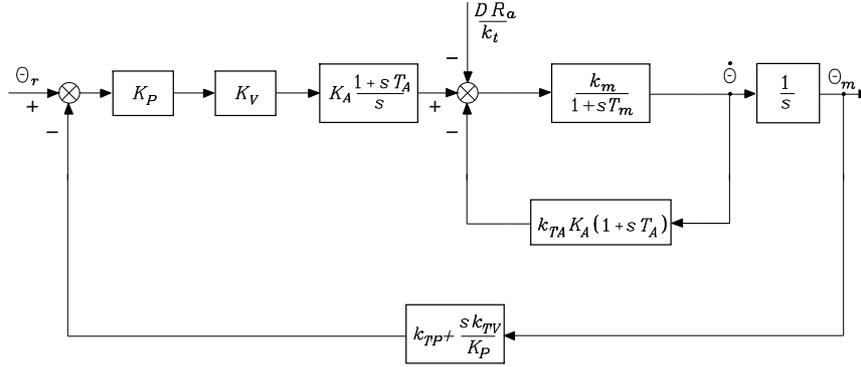


Fig. S8.2. Reduction on the block scheme of position, velocity and acceleration feedback control

same block has been moved onto the input to K_P ; then, the resulting two blocks in parallel have been summed giving $(k_{TP} + sk_{TV}/K_P)$. Solving the innermost feedback loop gives the transfer function

$$sM(s) = \frac{\frac{k_t}{sR_a I}}{1 + \frac{k_t k_v}{sR_a I}} = \frac{\frac{1}{k_v}}{1 + \frac{sR_a I}{k_v k_t}} = \frac{k_m}{1 + sT_m},$$

and thus the block scheme of the system becomes that in Fig. S8.2.

Solving the inner feedback loop gives the transfer function

$$\begin{aligned} G'(s) &= \frac{\frac{k_m}{1 + sT_m}}{1 + \frac{k_m k_{TA} K_A (1 + sT_A)}{(1 + sT_m)}} \\ &= \frac{k_m}{(1 + k_m K_A k_{TA}) \left(1 + \frac{sT_m \left(1 + k_m K_A k_{TA} \frac{T_A}{T_m} \right)}{(1 + k_m K_A k_{TA})} \right)}, \end{aligned}$$

and thus the block scheme of the system becomes that in Fig. S8.3.

The transfer function of the forward path is then

$$P(s) = \frac{K_P K_V C_A(s) G'(s)}{s} = \frac{K_P K_V K_A (1 + sT_A)}{s^2} G'(s).$$

The transfer function of the return path is

$$H(s) = k_{TP} \left(1 + \frac{sk_{TV}}{K_P k_{TP}} \right).$$

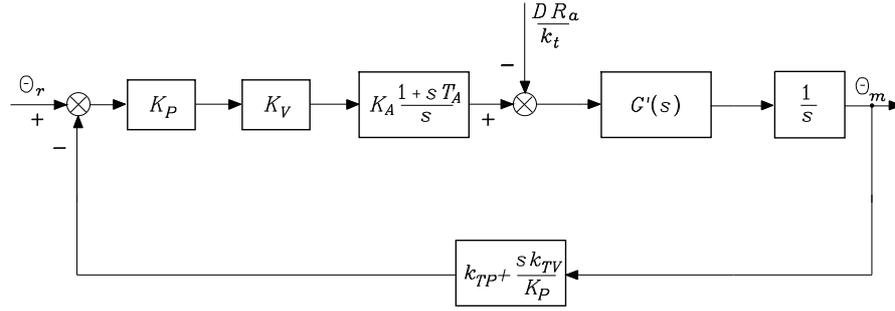


Fig. S8.3. Further reduction on the block scheme of position, velocity and acceleration feedback control

It follows that the closed-loop input/output transfer function with $T_A = T_m$ is

$$\frac{\Theta_m(s)}{\Theta_r(s)} = \frac{P(s)}{1 + P(s)H(s)} = \frac{\frac{1}{k_{TP}}}{1 + \frac{sk_{TV}}{K_P k_{TP}} + \frac{s^2(1 + k_m K_A k_{TA})}{k_m K_P k_{TP} K_V K_A}},$$

while the closed-loop disturbance/output transfer function is

$$\frac{\Theta_m(s)}{D(s)} = -\frac{\frac{G'(s)}{s}}{1 + P(s)H(s)} = -\frac{\frac{sR_a}{k_t K_P k_{TP} K_V K_A (1 + sT_A)}}{1 + \frac{sk_{TV}}{K_P k_{TP}} + \frac{s^2(1 + k_m K_A k_{TA})}{k_m K_P k_{TP} K_V K_A}}.$$

Solution to Problem 8.4

With the given data, the condition $F_m \ll k_v k_t / R_a$ is verified and $k_m = 2 \text{ rad}/(\text{V} \cdot \text{s})$, $T_m = 7.2 \text{ s}$, $k_{TP} = 1$. The closed-loop input/output transfer function is

$$W(s) = \frac{(1 + sT_P)}{1 + sT_P + \frac{s^2(1 + sT_m)}{k_m K_P k_{TP}}}.$$

Choosing $T_P = 10T_m = 72 \text{ s}$ leads to the root locus in Fig. S8.4 as a function of the gain $K' = 20K_P$.

If a damping ratio $\zeta = 0.4$ is desired, the resulting controller gain is $K_P = 0.0013$. For such value, two complex poles $(-0.062, \pm j0.142)$ and a real pole (-0.015) are obtained. Hence, the above transfer function can be written as

$$W(s) = \frac{(1 + 72s)}{(1 + 5.161s + 41.62s^2)(1 + 66.66s)}.$$

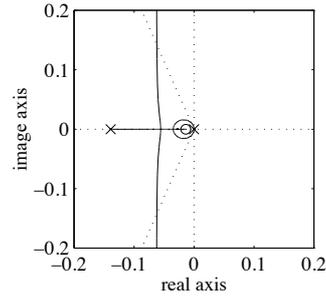


Fig. S8.4. Root locus for the position feedback control scheme

Correspondingly, the disturbance rejection factor is $X_R = 0.0013$, while an estimate of the output recovery time is $T_R = 72$ s. These values reveal the poor disturbance rejection performance of the closed-loop system under simple position feedback control.

Solution to Problem 8.5

With the given data, the condition $F_m \ll k_v k_t / R_a$ is verified and $k_m = 2$ rad/(V·s), $T_m = 7.2$ s, $k_{TP} = 1$, $k_{TV} = 1$. Choosing $T_V = T_m$ yields the closed-loop input/output transfer function

$$W(s) = \frac{1}{1 + \frac{s}{K_P} + \frac{s^2}{2K_P K_V}}.$$

With the given specifications for ζ and ω_n , the controller gains can be determined according to (8.30) and (8.31), giving $K_P = 25$ and $K_V = 8$, and thus the above transfer function becomes

$$W(s) = \frac{1}{1 + 0.04s + 0.0025s^2}$$

which has two complex poles $(-8, \pm j18.33)$. Correspondingly, the disturbance rejection factor is $X_R = 200$, while an estimate of the output recovery time is $T_R = 7.2$ s. These values reveal the good disturbance rejection performance of the closed-loop system under position and velocity feedback control; however, it is not possible to further decrease the output recovery time which is limited by the time constant T_m of the drive system.

Solution to Problem 8.6

With the given data, the condition $F_m \ll k_v k_t / R_a$ is verified and $k_m = 2 \text{ rad}/(\text{V}\cdot\text{s})$, $T_m = 7.2 \text{ s}$, $k_{TP} = 1$, $k_{TV} = 1$, $k_{TA} = 1$. Choosing $T_A = T_m$ yields the closed-loop input/output transfer function

$$W(s) = \frac{1}{1 + \frac{s}{K_P} + \frac{s^2(1+2K_A)}{2K_P K_V K_A}}.$$

With the given specifications for ζ , ω_n and X_R , the controller gains can be determined according to (8.39), (8.40), (8.41) giving $K_P = 25$, $K_V = 32$, $K_A = 0.5$, and thus the above transfer function becomes

$$W(s) = \frac{1}{1 + 0.04s + 0.0025s^2}$$

which is the same transfer function of the previous problem. Remarkably, in the case of position, velocity and acceleration feedback control, the disturbance rejection performance is improved in terms of X_R , although the output recovery time T_R is still the same as in the previous case. Finally, in order to reconstruct acceleration from velocity measurement, a first-order filter as in Fig. 8.11 can be used. With the given ζ and ω_n , the closed-loop system bandwidth is $\omega_3 = 26.84 \text{ rad/s}$, and then the filter bandwidth shall be chosen wide enough with respect to the drive system bandwidth, e.g., $\omega_{3f} = k_f = 400 \text{ rad/s}$.

Solution to Problem 8.7

As for the block scheme in Fig. 8.12, where the saturation block can be neglected, it is worth moving both the output of the block T_m/k_m and the output of the block $1/k_m$ onto the input to the block $k_P(1+sT_P)/s$; also the quantities $\dot{\Theta}_d$ and $\ddot{\Theta}_d$ can be generated from Θ_d as $s\Theta_d$ and $s^2\Theta_d$, respectively. Therefore, the reduced block scheme becomes that in Fig. S8.5.

The relationship between the desired input and the new reference input is then

$$\begin{aligned} \Theta'_r(s) &= \left(k_{TP} + \left(\frac{s^2 T_m}{k_m} + \frac{s}{k_m} \right) \frac{s}{K_P(1+sT_P)} \right) \Theta_d(s) \\ &= \left(k_{TP} + \frac{s^2(1+sT_m)}{k_m K_P(1+sT_P)} \right) \Theta_d(s). \end{aligned}$$

As for the block scheme in Fig. 8.13, where the saturation block can be neglected, it is worth moving both the output of the block $1/k_m K_V$ and the output of the block k_{TV} onto the input to the block K_P ; as above, the quantities $\dot{\Theta}_d$ and $\ddot{\Theta}_d$ can be generated from Θ_d as $s\Theta_d$ and $s^2\Theta_d$, respectively. Therefore, the reduced block scheme becomes that in Fig. S8.6.

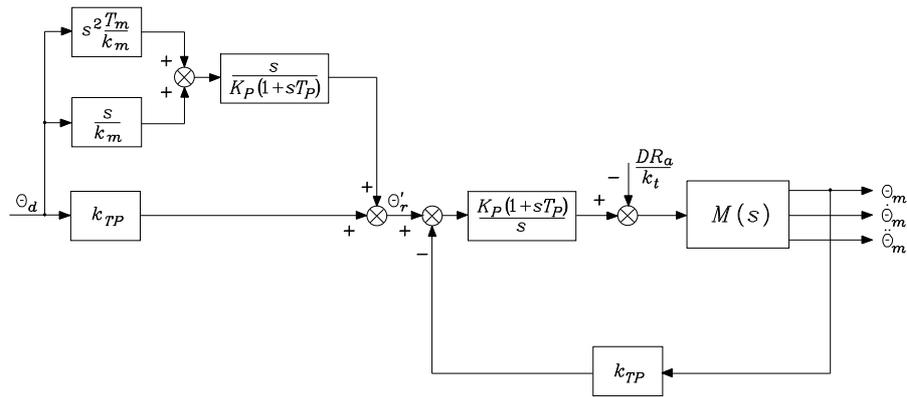


Fig. S8.5. Reduction on the block scheme of position feedback control with decentralized feedforward compensation

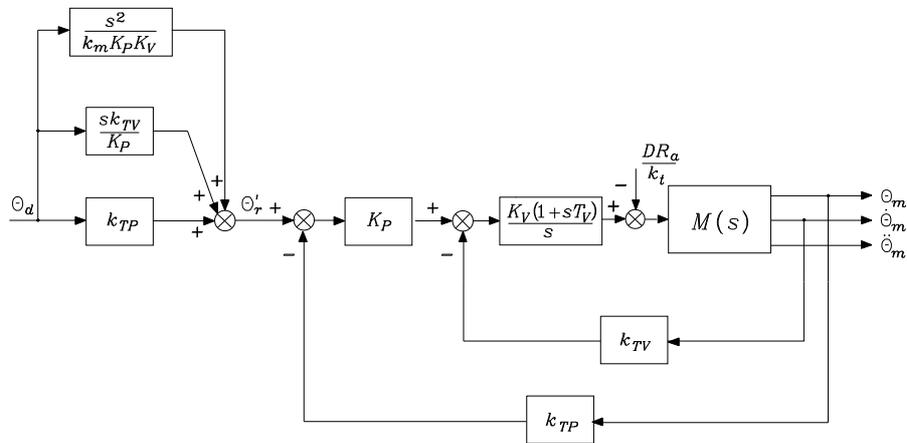


Fig. S8.6. Reduction on the block scheme of position and velocity feedback control with decentralized feedforward compensation

The relationship between the desired input and the new reference input is then

$$\Theta'_r(s) = \left(k_{TP} + \frac{s k_{TV}}{K_P} + \frac{s^2}{k_m K_P K_V} \right) \Theta_d(s).$$

As for the block scheme in Fig. 8.14, where the saturation block can be neglected, it is worth moving both the output of the block $(k_{TA} + 1/k_m K_A)$ and the output of the block k_{TV} onto the input to the block K_P ; again, the quantities $\dot{\Theta}_d$ and $\ddot{\Theta}_d$ can be generated from Θ_d as $s\Theta_d$ and $s^2\Theta_d$, respectively. Therefore, the reduced block scheme becomes that in Fig. S8.7.

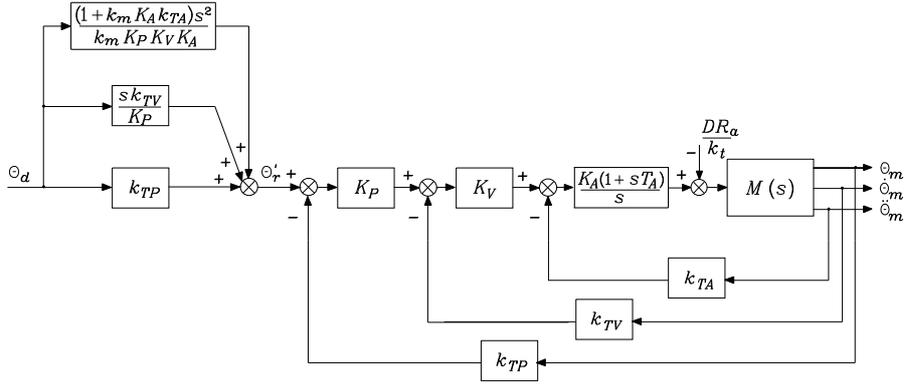


Fig. S8.7. Reduction on the block scheme of position, velocity and acceleration feedback control with decentralized feedforward compensation

The relationship between the desired input and the new reference input is then

$$\Theta'_r(s) = \left(k_{TP} + \frac{s k_{TV}}{K_P} + \frac{(1 + k_m K_A k_{TA}) s^2}{k_m K_P K_V K_A} \right) \Theta_d(s).$$

Solution to Problem 8.8

The block scheme in Fig. 8.15 is clearly equivalent to that in Fig. 8.12, since the block $K_P(1 + sT_P)/s$ can be split into two blocks in parallel, i.e., $K_P T_P$ and K_P/s .

Concerning the block scheme in Fig. 8.28, it is worth considering the input U and the output Y of the block $K_V(1 + sT_V)/s$. By neglecting the effect of the saturation block and setting $E = k_{TP}(\Theta_d - \Theta_m)$, the input U is given by

$$\begin{aligned} U &= K_P E + k_{TV}(\dot{\Theta}_d - \dot{\Theta}_m) + \frac{1}{k_m K_V} \ddot{\Theta}_d \\ &= K_P \left(1 + \frac{k_{TV}}{K_P k_{TP}} s \right) E + \frac{1}{k_m K_V} \ddot{\Theta}_d. \end{aligned}$$

Then the output can be expressed ($T_V = T_m$) as

$$\begin{aligned} Y &= \frac{K_V(1 + sT_V)}{s} \left(K_P \left(1 + \frac{k_{TV}}{K_P k_{TP}} s \right) E + \frac{1}{k_m K_V} \ddot{\Theta}_d \right) \\ &= K_P K_V \left(\frac{1}{s} + T_V \right) \left(1 + \frac{k_{TV}}{K_P k_{TP}} s \right) E + \frac{1}{k_m} \left(\frac{1}{s} + T_m \right) \ddot{\Theta}_d \\ &= \left(K_P K_V T_V + \frac{K_V k_{TV}}{k_{TP}} + \frac{K_P K_V}{s} + \frac{K_V T_V k_{TV}}{k_{TP}} s \right) E + \frac{1}{k_m} \dot{\Theta}_d + \frac{T_m}{k_m} \ddot{\Theta}_d \end{aligned}$$

corresponding to the block scheme in Fig. 8.16.

In a similar way, concerning the block scheme in Fig. 8.29, it is worth considering the input U and the output Y of the block $K_A(1 + sT_A)/s$. By neglecting the effect of the saturation blocks and setting $E = k_{TP}(\Theta_d - \Theta_m)$, the input U is given by

$$\begin{aligned} U &= K_V \left(K_P E + k_{TV}(\dot{\Theta}_d - \dot{\Theta}_m) \right) + \left(k_{TA} + \frac{1}{k_m K_A} \right) \ddot{\Theta}_d - k_{TA} \ddot{\Theta}_m \\ &= \left(K_P K_V + \frac{K_V k_{TV}}{k_{TP}} s + \frac{k_{TA}}{k_{TP}} s^2 \right) E + \frac{1}{k_m K_A} \ddot{\Theta}_d. \end{aligned}$$

Then the output can be expressed ($T_A = T_m$) as

$$\begin{aligned} Y &= K_A \left(\frac{1}{s} + T_A \right) \left(\left(K_P K_V + \frac{K_V k_{TV}}{k_{TP}} s + \frac{k_{TA}}{k_{TP}} s^2 \right) E + \frac{1}{k_m K_A} \ddot{\Theta}_d \right) \\ &= \left(K_P K_V K_A T_A + \frac{K_V k_{TV} K_A}{k_{TP}} + \frac{K_P K_V K_A}{s} \right. \\ &\quad \left. + \left(\frac{K_V k_{TV} K_A T_A + K_A k_{TA}}{k_{TP}} \right) s + \frac{K_A T_A k_{TA}}{k_{TP}} s^2 \right) E \\ &\quad + \frac{1}{k_m} \dot{\Theta}_d + \frac{T_m}{k_m} \ddot{\Theta}_d \end{aligned}$$

corresponding to the block scheme in Fig. 8.17.

Solution to Problem 8.9

From (8.75), the following matrix inequality can be written:

$$B_m \mathbf{I} \leq \mathbf{B}^{-1} \leq B_M \mathbf{I}$$

where all three matrices are positive definite. Setting

$$\hat{\mathbf{B}} = \frac{2}{B_M + B_m} \mathbf{I},$$

which is positive definite, allows writing the matrix inequality

$$\frac{2B_m}{B_M + B_m} \mathbf{I} \leq \mathbf{B}^{-1} \hat{\mathbf{B}} \leq \frac{2B_M}{B_M + B_m} \mathbf{I}.$$

Subtracting the identity matrix from each term gives

$$-\alpha \mathbf{I} \leq \mathbf{B}^{-1} \hat{\mathbf{B}} - \mathbf{I} \leq \alpha \mathbf{I}$$

where

$$\alpha = \frac{B_M - B_m}{B_M + B_m}$$

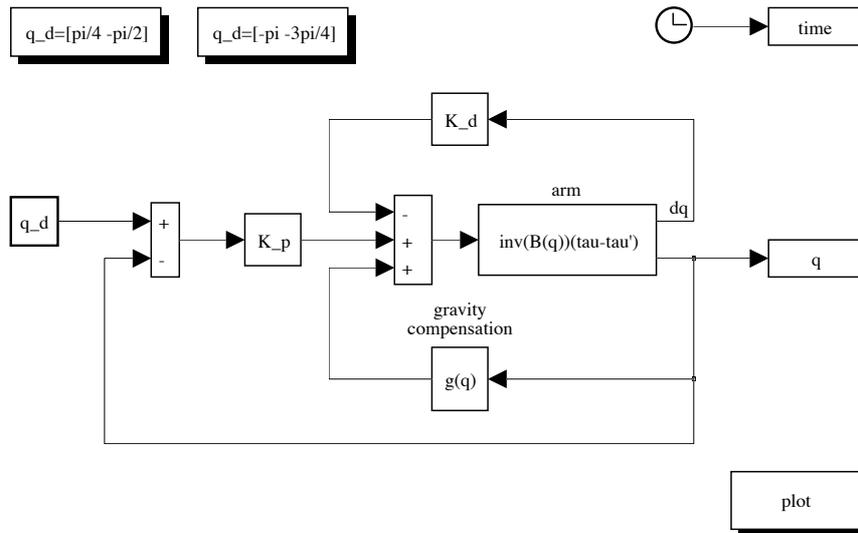


Fig. S8.8. SIMULINK block diagram of joint space PD control with gravity compensation

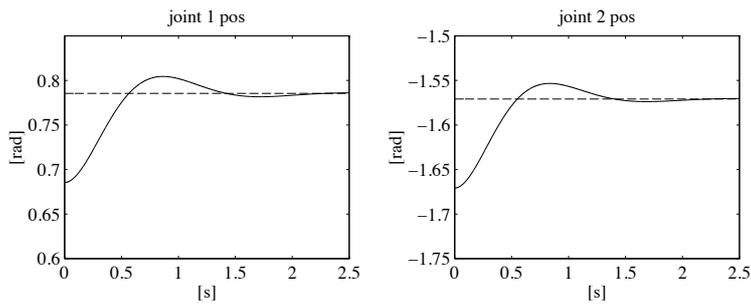


Fig. S8.9. Time history of the joint positions with joint space PD control with gravity compensation for the first posture

and $0 < \alpha < 1$. The matrix in the middle is lower bounded by a negative definite diagonal matrix and upper bounded by a positive definite diagonal matrix, where these two matrices have the same norm. Thus, it follows that

$$\|B^{-1}\widehat{B} - I\| \leq \alpha.$$

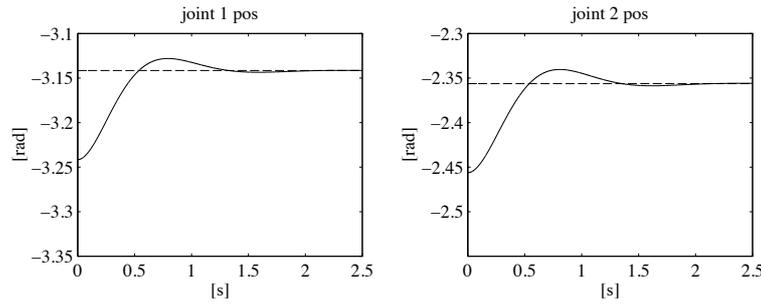


Fig. S8.10. Time history of the joint positions with joint space PD control with gravity compensation for the second posture

Solution to Problem 8.10

The control scheme in Fig. 8.20 is used with the same matrix gains as for case F in Sect. 8.7, i.e.,

$$\mathbf{K}_P = 3750\mathbf{I} \quad \mathbf{K}_D = 750\mathbf{I}.$$

The initial arm postures for the two cases are chosen as $\mathbf{q} = [0.1 \ 0.1]^T$. The various terms of the dynamic model are computed by using the minimum parameterization in the solution to Problem 7.4.

The resulting SIMULINK block diagram is shown in Fig. S8.8, where both desired postures can be assigned. The arm is simulated as a continuous-time system using a variable-step integration method with a maximum step size of 1 ms. All the blocks of the controller are simulated as discrete-time subsystems with the given sampling time of 1 ms.

The with the solution can be found in Folder 8_10.

The resulting joint positions for the two postures are shown in Figs. S8.9 and S8.10, respectively. The dashed line indicates the desired joint position, while the solid line indicates the actual joint position. It can be seen that both postures are reached at steady state.

Solution to Problem 8.11

With reference to the dynamic model in the solution to Problem 7.4, the presence of a payload alters the dynamic parameters of augmented Link 2. Since the payload is a mass m_L concentrated at the tip, the only parameter that varies is the mass of the augmented link which becomes

$$m'_2 = m_2 + m_L;$$

in fact, both the inertia first moment $m_2\ell_{C2}$ and inertia moment \bar{I}_2 remain the same because they are evaluated with respect to a frame located at the tip. Hence, only parameters π'_1 and π'_3 in (S7.2) have to be recomputed with the modified m'_2 .

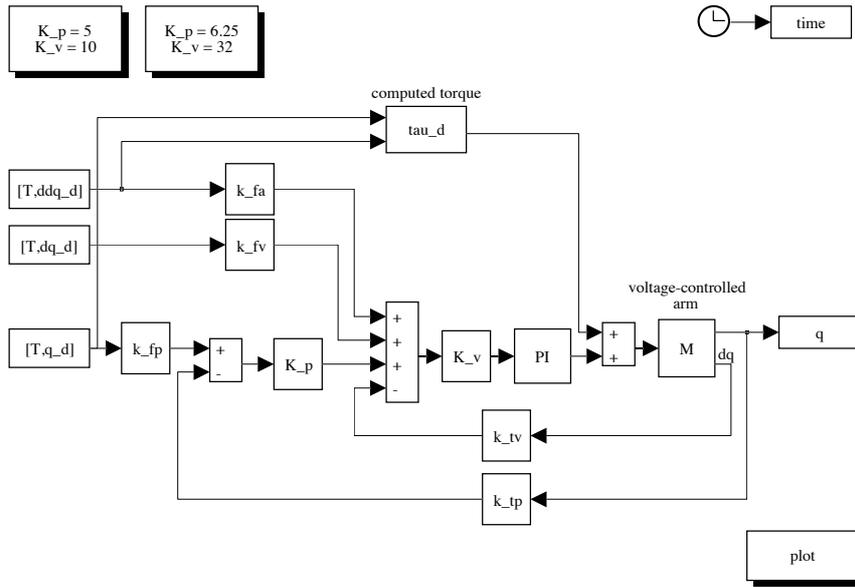


Fig. S8.11. SIMULINK block diagram of joint space computed torque control with feedforward compensation of the diagonal terms of the inertia matrix and of gravitational terms, and independent joint control with position and velocity feedback

The desired trajectories for the two joints are generated via trapezoidal velocity profiles with maximum velocities $\dot{q}_{c1} = 3\pi/4$ rad/s and $\dot{q}_{c2} = \pi/3$ rad/s, respectively.

The control scheme in Fig. 8.19 is used with feedforward compensation of the diagonal terms of the inertia matrix and of gravitational terms. As for the decentralized controller, the independent joint control with position and velocity feedback in Fig. 5.11 is adopted.

Initially, the same gains for each joint servo are chosen as for case A in Sect. 8.7, i.e., ($k_{TP} = k_{TV} = 1$)

$$K_P = 5 \quad K_V = 10,$$

corresponding to $\omega_n = 5$ rad/s and $\zeta = 0.5$, and thus to a closed-loop bandwidth $\omega_3 = 6.36$ rad/s. This gives a disturbance rejection factor $X_R = 50$.

Then, by observing that the required trajectory is faster than the corresponding trajectory in Sect. 8.7, a larger bandwidth is imposed. Setting $\omega_n = 10$ rad/s and $\zeta = 0.8$ gives $\omega_3 = 10.78$ rad/s. From (8.30) and (8.31), the gains become

$$K_P = 6.25 \quad K_V = 32$$

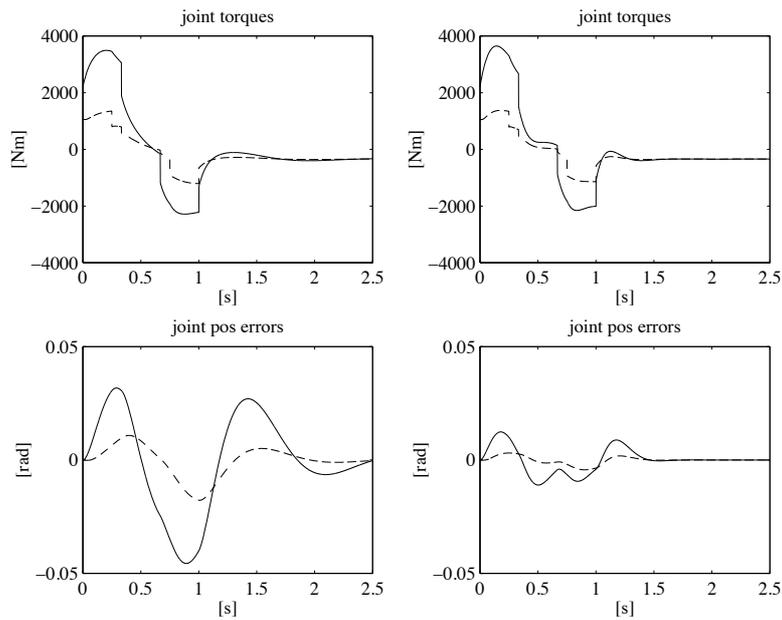


Fig. S8.12. Time history of the joint torques and position errors with joint space computed torque control with feedforward compensation of the diagonal terms of the inertia matrix and of gravitational terms, and independent joint control with position and velocity feedback; *left*—first choice of servo gains, *right*—second choice of servo gains

and, from (8.33), it is $X_R = 200$, i.e., a disturbance rejection factor which is four times as much as the previous one.

The resulting SIMULINK block diagram is shown in Fig. S8.11, where both choices of servo gains can be executed. The arm is voltage-controlled as in the scheme of Fig. 8.3 and is simulated as a continuous-time system using a variable-step integration method with a maximum step size of 1 ms. All the blocks of the controller are simulated as discrete-time subsystems with the given sampling time of 1 ms; in particular, the PI block is discretized assuming a zero-order hold on the inputs.

The files with the solution can be found in Folder 8_11.

The resulting joint torques and position errors are shown in Fig. S8.12. The solid line refers to Joint 1, while the dashed line refers to Joint 2. It can be seen that the tracking performance with the second choice of the servo gains is considerably improved, at the expense of a sharper behaviour of joint torques, though.

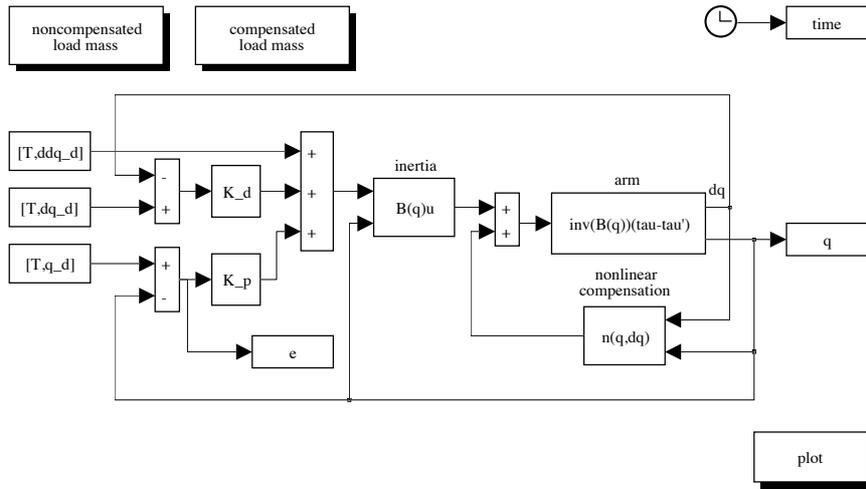


Fig. S8.13. SIMULINK block diagram of joint space inverse dynamics control

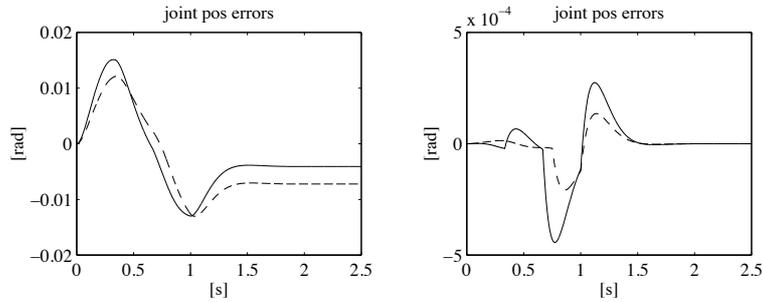


Fig. S8.14. Time history of the joint position errors with joint space inverse dynamics control; *left*—noncompensated load mass, *right*—compensated load mass

Solution to Problem 8.12

The modification of the dynamic model to account for the load mass and the generation of joint trajectories are accomplished as in the solution to Problem 8.11.

The control scheme in Fig. 8.22 is used where the matrix gains are chosen as

$$K_P = 100I \quad K_D = 16I$$

corresponding to an error dynamics for both joints characterized by $\omega_{ni} = 10 \text{ rad/s}$ and $\zeta_i = 0.8$ for $i = 1, 2$.

The resulting SIMULINK block diagram is shown in Fig. S8.13, where both the noncompensated load mass case and the compensated load mass case can

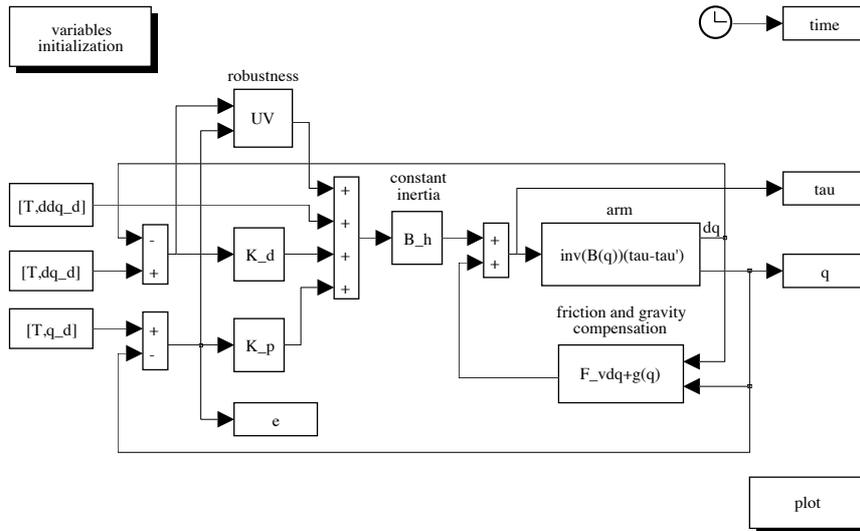


Fig. S8.15. SIMULINK block diagram of joint space robust control

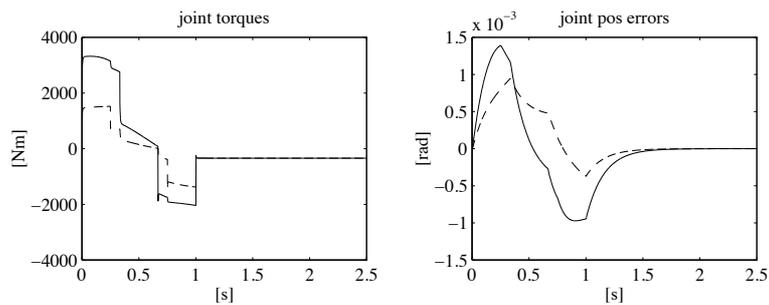


Fig. S8.16. Time history of the joint torques and position errors with joint space robust control

be executed. The arm is torque-controlled as in the scheme of Fig. 8.4 and is simulated as a continuous-time system using a variable-step integration method with a maximum step size of 0.1 ms. All the blocks of the controller are simulated as discrete-time subsystems with the given sampling time of 1 ms.

The files with the solution can be found in Folder 8_12.

The resulting joint position errors in the two cases are shown in Fig. S8.14. The solid line refers to Joint 1, while the dashed line refers to Joint 2. It can be seen that the tracking performance is excellent when load mass is compensated, while it is degraded both during transient and at steady state

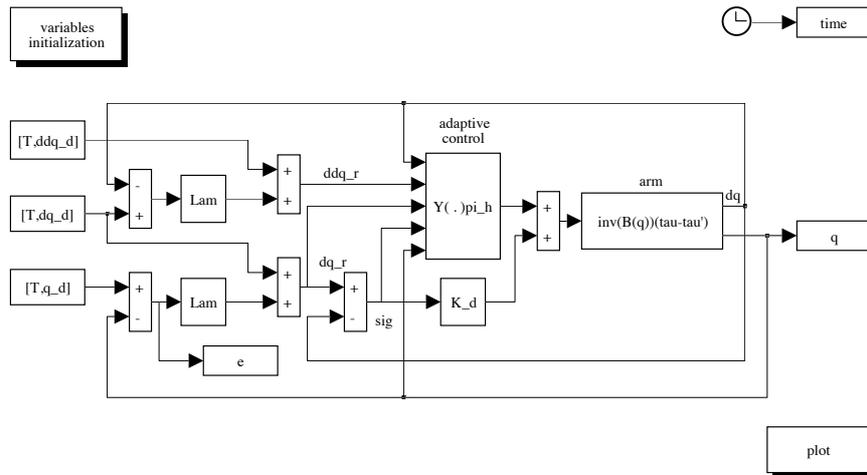


Fig. S8.17. SIMULINK block diagram of joint space adaptive control

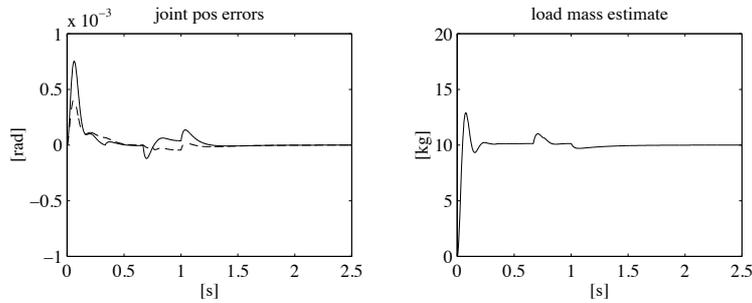


Fig. S8.18. Time history of the joint position errors and of the load mass estimate with joint space adaptive control

when the load mass is not compensated. Also, in the case of noncompensated load mass, steady-state errors occur.

Solution to Problem 8.13

The modification of the dynamic model to account for the load mass and the generation of joint trajectories are accomplished as in the solution to Problem 8.11.

The control scheme in Fig. 8.23 is used with constant inertia ($\hat{B} = \bar{B}$) and compensation of friction and gravity ($\hat{n} = F_v \dot{q} + g$). The gains defining the error dynamics are chosen as in the solution to Problem 8.12, i.e.,

$$K_P = 100I \quad K_D = 16I;$$

furthermore, the matrix \mathbf{P} in (8.81) is chosen as

$$\mathbf{P} = \mathbf{I},$$

and the gain ρ and the thickness of the boundary layer in (8.89) are respectively chosen as

$$\rho = 70 \quad \epsilon = 0.001.$$

The resulting SIMULINK block diagram is shown in Fig. S8.15. The arm is simulated as a continuous-time system using a variable step integration method with a maximum step size of 1 ms. All the blocks of the controller are simulated as discrete-time subsystems with the given sampling time of 1 ms.

The files with the solution can be found in Folder 8_13.

The resulting joint torques and position errors are shown in Fig. S8.16. The solid line refers to Joint 1, while the dashed line refers to Joint 2. It can be seen that, in spite of the imperfect dynamic model compensation, the tracking performance is satisfactory, thanks to the use of the robustness action.

Solution to Problem 8.14

A minimum parameterization of the dynamic model is obtained as in the solution to Problem 7.4, with suitable modification of augmented Link 2 parameters as in the solution to Problem 8.11. The generation of joint trajectories is also accomplished as in the solution to Problem 8.11.

The control scheme in Fig. 8.26 is used. In terms of the new reference velocity and acceleration vectors in (8.92), the regressor in (S7.3) becomes

$$\begin{aligned} y'_{11} &= a_1 \ddot{\vartheta}_{r1} + gc_1 \\ y'_{12} &= \ddot{\vartheta}_{r1} \\ y'_{13} &= (2a_1c_2 + a_2) \ddot{\vartheta}_{r1} + (a_1c_2 + a_2) \ddot{\vartheta}_{r2} - a_1s_2 \dot{\vartheta}_{r1} \dot{\vartheta}_{r2} - a_1s_2 \dot{\vartheta}_1 \dot{\vartheta}_{r2} - a_1s_2 \dot{\vartheta}_2 \dot{\vartheta}_{r2} \\ &\quad + gc_{12} \\ y'_{14} &= \ddot{\vartheta}_{r1} + \ddot{\vartheta}_{r2} \\ y'_{15} &= \ddot{\vartheta}_{r1} + k_{r2} \ddot{\vartheta}_{r2} \\ y'_{21} &= 0 \\ y'_{22} &= 0 \\ y'_{23} &= (a_1c_2 + a_2) \ddot{\vartheta}_{r1} + a_2 \ddot{\vartheta}_{r2} + a_1s_2 \dot{\vartheta}_1 \dot{\vartheta}_{r1} + gc_{12} \\ y'_{24} &= \ddot{\vartheta}_{r1} + \ddot{\vartheta}_{r2} \\ y'_{25} &= k_{r2} \ddot{\vartheta}_{r1} + k_{r2}^2 \ddot{\vartheta}_{r2}. \end{aligned}$$

In view of the presence of a concentrated tip load mass, the only variations occurring on the parameters in (S7.2) are:

$$\Delta\pi'_1 = a_1m_L \quad \Delta\pi'_3 = a_2m_L.$$

Hence, the control law in (8.98) can be formally written as

$$\mathbf{u} = \mathbf{Y}'(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}_r) \boldsymbol{\pi} + \mathbf{F}_v \dot{\mathbf{q}}_r + m_L \mathbf{y}_L + \mathbf{K}_D \boldsymbol{\sigma}$$

where \mathbf{Y}' is computed as above and

$$\mathbf{y}_L = \begin{bmatrix} a_1 y'_{11} + a_2 y'_{13} \\ a_1 y'_{21} + a_2 y'_{23} \end{bmatrix},$$

being $\pi = m_L$ the only unknown parameter.

The gains are chosen as

$$\boldsymbol{\Lambda} = 10\mathbf{I} \quad \mathbf{K}_D = 10000\mathbf{I} \quad k_\pi = 0.005;$$

the initial estimate of π is set to 0, and the true value of m_L is utilized only to update the simulated arm model.

The resulting SIMULINK block diagram is shown in Fig. S8.17. The arm is simulated as a continuous-time system using a variable-step integration method with a maximum step size of 1 ms. All the blocks of the controller are simulated as discrete-time subsystems with the given sampling time of 1 ms.

The files with the solution can be found in Folder 8_14.

The resulting joint position errors and load mass estimate are shown in Fig. S8.18. For the joint position errors, the solid line refers to Joint 1, while the dashed line refers to Joint 2. It can be seen that the tracking performance is satisfactory, thanks to the adaptive action on the load mass; in this case, the parameter estimate converges to the true value.

Solution to Problem 8.15

The modification of the dynamic model to account for the load mass is accomplished as in the solution to Problem 8.11.

The control scheme in Fig. 8.29 is used with the same matrix gains as for case K in Sect. 8.7, i.e.,

$$\mathbf{K}_P = 16250\mathbf{I} \quad \mathbf{K}_D = 3250\mathbf{I}.$$

The initial tip positions for the two postures are chosen as $\mathbf{p} = [0.1 \ 0.1]^T$.

The resulting SIMULINK block diagram is shown in Fig. S8.19, where both desired positions can be assigned. The arm is simulated as a continuous-time system using a variable-step integration method with a maximum step size of 1 ms. All the blocks of the controller are simulated as discrete-time subsystems with the given sampling time of 1 ms.

The files with the solution can be found in Folder 8_15.

The resulting components of tip position for the two postures are shown in Figs. S8.20 and S8.21, respectively. The dashed line indicates the desired tip coordinate, while the solid line indicates the actual tip coordinate. It can be seen that both postures are reached at steady state.

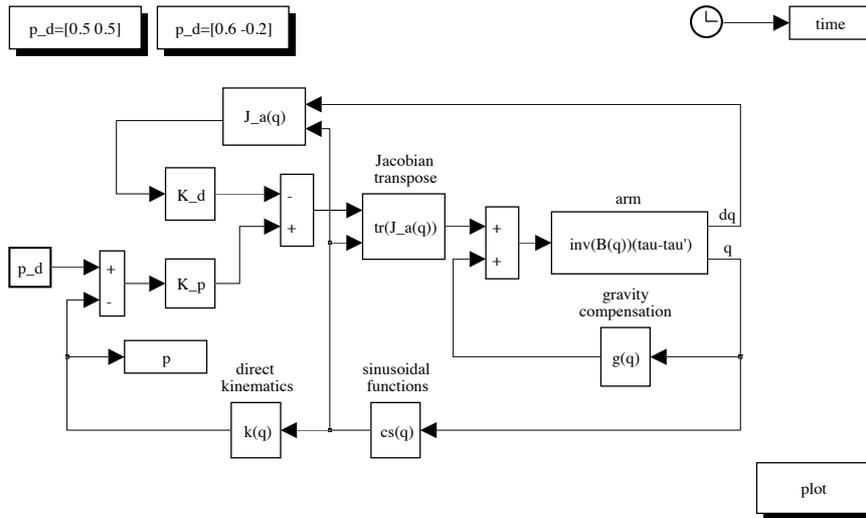


Fig. S8.19. SIMULINK block diagram of operational space PD control with gravity compensation

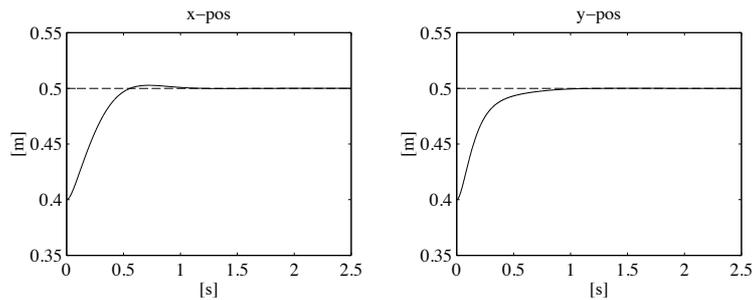


Fig. S8.20. Time history of the tip position components with operational space PD control with gravity compensation for the first posture

Solution to Problem 8.16

The modification of the dynamic model to account for the load mass is accomplished as in the solution to Problem 8.11. The desired tip trajectory is generated in terms of the path coordinate s via a trapezoidal velocity profile with maximum velocity $\dot{s}_c = 1.5$.

The control scheme in Fig. 8.30 is used with the matrix gains:

$$\mathbf{K}_P = 100\mathbf{I} \quad \mathbf{K}_D = 16\mathbf{I}.$$

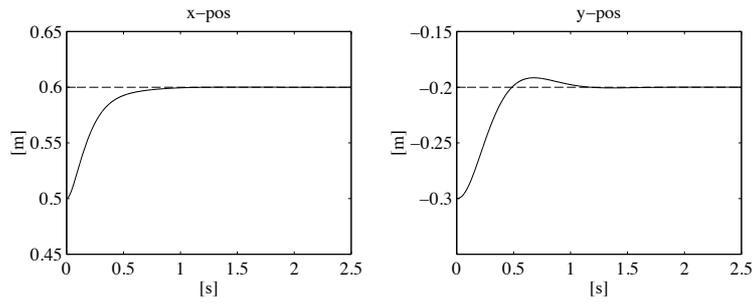


Fig. S8.21. Time history of the tip position components with operational space PD control with gravity compensation for the second posture

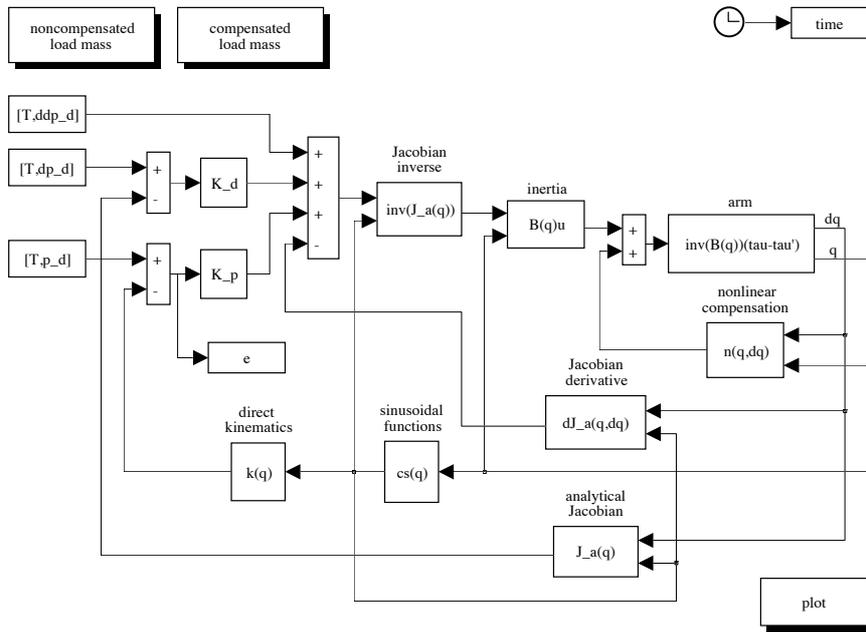


Fig. S8.22. SIMULINK block diagram of operational space inverse dynamics control

The resulting SIMULINK block diagram is shown in Fig. S8.22, where both the noncompensated load mass case and the compensated load mass case can be executed. The arm is simulated as a continuous-time system using a variable-step integration method with a maximum step size of 1 ms. All the blocks of the controller are simulated as discrete-time subsystems with the given sampling time of 1 ms.

The files with the solution can be found in Folder 8_16.

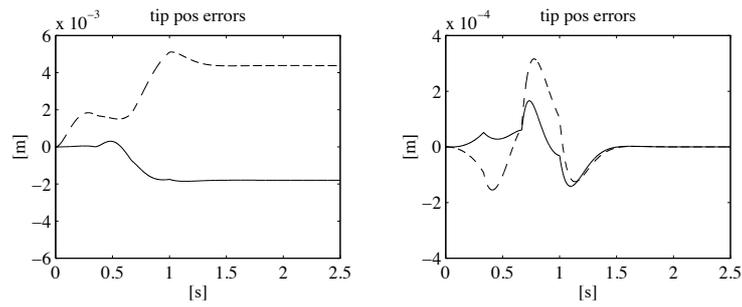


Fig. S8.23. Time history of the components of tip position error with operational space inverse dynamics control; *left*—noncompensated load mass, *right*—compensated load mass

The resulting components of tip position error in the two cases are shown in Fig. S8.23. The solid line refers to x component, while the dashed line refers to y component. It can be seen that the performance is excellent when the load mass is compensated, while it is degraded both during transient and at steady state when the load mass is not compensated.

Force Control

Solution to Problem 9.1

Computing the time derivative of $\mathbf{o}_{d,e}^d = \mathbf{R}_d^T(\mathbf{o}_e - \mathbf{o}_d)$ gives

$$\dot{\mathbf{o}}_{d,e}^d = \mathbf{R}_d^T(\dot{\mathbf{o}}_e - \dot{\mathbf{o}}_d) - \dot{\mathbf{R}}_d^T(\mathbf{o}_e - \mathbf{o}_d). \quad (\text{S9.1})$$

In view of (3.10), (3.7) and (3.11) the following equality holds

$$\dot{\mathbf{R}}_d^T = -\mathbf{R}_d^T \mathbf{S}(\boldsymbol{\omega}_d) = -\mathbf{S}(\mathbf{R}_d^T \boldsymbol{\omega}_d) \mathbf{R}_d^T = -\mathbf{S}(\boldsymbol{\omega}_d^d) \mathbf{R}_d^T \quad (\text{S9.2})$$

which, replaced in (S9.1), gives (9.10).

To derive (9.11), consider the equality

$$\dot{\mathbf{R}}_e^d = \dot{\mathbf{R}}_d^T \mathbf{R}_e + \mathbf{R}_d^T \dot{\mathbf{R}}_e = -\mathbf{S}(\boldsymbol{\omega}_d^d) \mathbf{R}_d^e + \mathbf{S}(\boldsymbol{\omega}_e^d) \mathbf{R}_d^e = \mathbf{S}(\boldsymbol{\omega}_{d,e}^d) \mathbf{R}_d^e,$$

where (S9.2), (3.10), and (3.11) have been used. It follows that (3.64) can be rewritten as

$$\boldsymbol{\omega}_{d,e}^d = \mathbf{T}(\boldsymbol{\phi}_{d,e}) \dot{\boldsymbol{\phi}}_{d,e}$$

being $\boldsymbol{\phi}_{d,e}$ the vector of Euler angles extracted from \mathbf{R}_e^d and $\boldsymbol{\omega}_{d,e}^d$ the angular velocity corresponding to $\dot{\mathbf{R}}_e^d$. Hence, expression (9.11) follows.

Solution to Problem 9.2

Pre-multiplying by \mathbf{K} both sides of the equality

$$d\mathbf{x}_{r,e} = d\mathbf{x}_{r,d} - d\mathbf{x}_{e,d},$$

and using (9.21) and (9.20) gives

$$\mathbf{h}_e = \mathbf{K} d\mathbf{x}_{r,d} - \mathbf{K} \mathbf{K}_P^{-1} \mathbf{h}_e$$

and thus

$$(\mathbf{I}_6 + \mathbf{K}\mathbf{K}_P^{-1})\mathbf{h}_e = \mathbf{K}d\mathbf{x}_{r,d},$$

from which expression (9.22) follows. Equality (9.23) is obtained replacing (9.22) into (9.20).

Solution to Problem 9.3

Similarly to Example 9.2, all the quantities can be referred to a base frame and control law with force measurement (9.30), (9.31) can be adopted. To compute the model of the contact force, in view of the geometry of the environment, it is useful to consider also a rotated base frame with axes parallel to the axes x_c and y_c of Fig. 9.16 and with the same origin of the base frame of axes x_0 and y_0 . The corresponding rotation matrix is

$$\mathbf{R}_c = \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}.$$

In the rotated base frame, the environment stiffness matrix has the simple expression

$$\mathbf{K}^c = \text{diag}\{0, k_y\},$$

corresponding to the absence of interaction forces along the direction of axis x_c ($\mathbf{f}_e^c = [0 \ f_y^c]^T$). The elastic force in the rotated base frame has the expression

$$\mathbf{f}_e^c = \mathbf{K}^c(\mathbf{o}_e^c - \mathbf{o}_r^c),$$

being \mathbf{o}_e^c the end-effector position and \mathbf{o}_r^c the equilibrium position of the plane in the rotated base frame. The above expression can be rewritten in the base frame in the form

$$\mathbf{f}_e = \mathbf{K}(\mathbf{o}_e - \mathbf{o}_r),$$

with

$$\mathbf{K} = \mathbf{R}_c \mathbf{K}^c \mathbf{R}_c^T = k_y \begin{bmatrix} 1/2 & -1/2 \\ -1/2 & 1/2 \end{bmatrix}$$

and $\mathbf{o}_r = [1 \ 0]^T$.

The impedance parameters can be set similarly to Example 9.2, on the basis of the equations

$$\begin{aligned} m_{dx}\ddot{x}_e^c + k_{Dx}\dot{x}_e^c + k_{Px}x_e^c &= k_{Px}x_d^c \\ m_{dy}\ddot{y}_e^c + k_{Dy}\dot{y}_e^c + (k_{Py} + k_y)y_e^c &= k_y y_r^c + k_{Py}y_d^c \end{aligned}$$

referred to the rotated base frame, where the diagonal matrices

$$\mathbf{M}_d^c = \text{diag}\{m_{dx}, m_{dy}\} \quad \mathbf{K}_D^c = \text{diag}\{k_{Dx}, k_{Dy}\} \quad \mathbf{K}_P^c = \text{diag}\{k_{Px}, k_{Py}\}$$

define an active impedance in the rotated base frame. The corresponding impedance matrices referred to the base frame are

$$\mathbf{M}_d = \mathbf{R}_c \mathbf{M}_d^c \mathbf{R}_c^T \quad \mathbf{K}_D = \mathbf{R}_c \mathbf{K}_D^c \mathbf{R}_c^T \quad \mathbf{K}_P = \mathbf{R}_c \mathbf{K}_P^c \mathbf{R}_c^T.$$

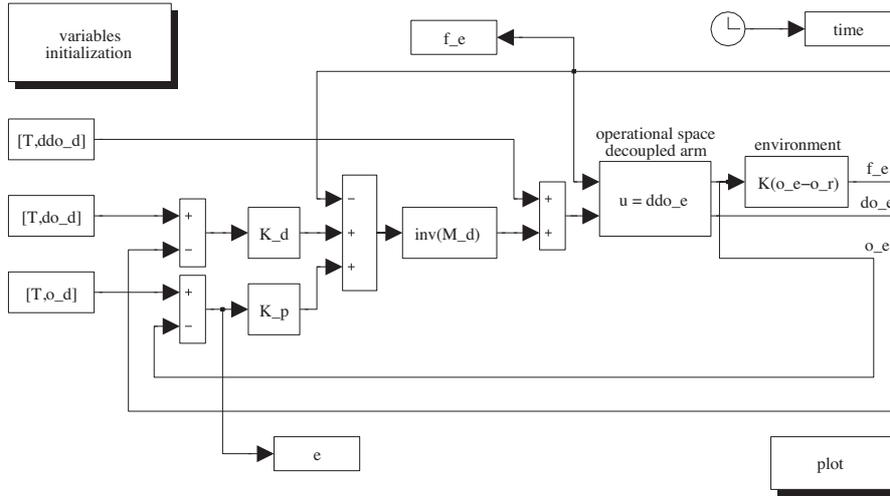


Fig. S9.1. SIMULINK block diagram of impedance control.

Notice that, differently from Example 9.2, the interaction occurs along y_c . Therefore, for the unconstrained motion along x_c , the behaviour is determined by

$$\omega_{nx} = \sqrt{\frac{k_{Px}}{m_{dx}}} \quad \zeta_x = \frac{k_{Dx}}{2\sqrt{m_{dx}k_{Px}}},$$

while, for the constrained motion along axis y_c , the behaviour is determined by

$$\omega_{ny} = \sqrt{\frac{k_{Py} + k_y}{m_{dy}}} \quad \zeta_y = \frac{k_{Dy}}{2\sqrt{m_{dy}(k_{Py} + k_y)}}.$$

With the choice

$$m_{dx} = m_{dy} = 100 \quad k_{Dx} = k_{Dy} = 1600 \quad k_{Px} = k_{Py} = 5000,$$

and the given value of environment stiffness k_y , the dynamics is characterized by

$$\omega_{nx} = 7.07 \text{ rad/s} \quad \zeta_x = 1.13$$

and

$$\omega_{ny} = 10 \text{ rad/s} \quad \zeta_y = 0.8.$$

The desired tip trajectory is generated in terms of the path coordinate s via a trapezoidal velocity profile with maximum velocity $\dot{s}_c = 0.5$.

The resulting SIMULINK block diagram is shown in Fig. S9.1. The arm is simulated as a continuous-time system using the a variable-step integration method with a minimum step size of 1 ms. All the blocks of the controller are simulated as discrete-time subsystems with the given sampling time of 1 ms.

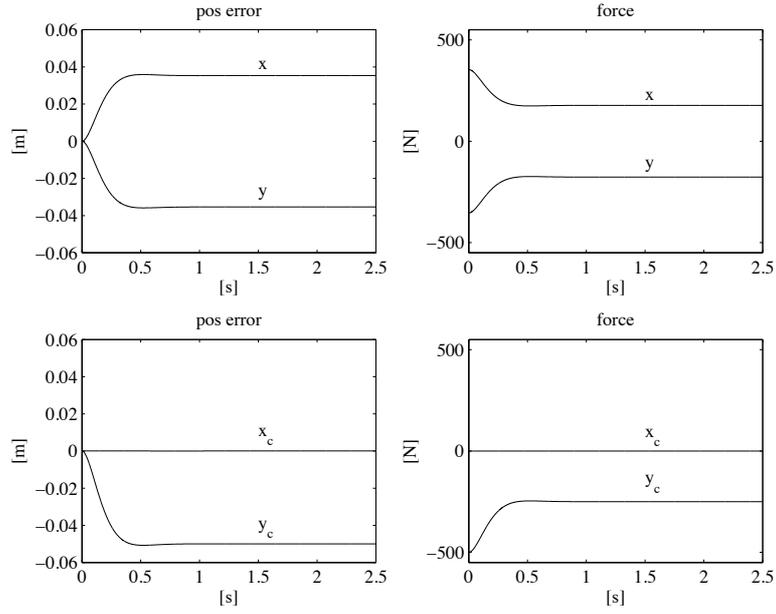


Fig. S9.2. Time history of the tip position error and of the contact force with impedance control. Top: components in the base frame. Bottom: components in the rotated base frame.

The files with the solution can be found in Folder 9_3.

The resulting tip position error and contact force are shown in Fig. S9.2, both referred to the base frame (top) and to the rotated base frame (bottom). It can be verified that the position error keeps small during the task execution, ensuring a bounded contact force. In particular, it can be observed that the position error is null along the unconstrained motion direction (axis x_c), where the force is null; on the other hand, the force response along the constrained motion direction (axis y_c) remains bounded and converges to a constant value which depends on the plane stiffness and undeformed position as well as on the active compliance.

Solution to Problem 9.4

From the block scheme in Fig. 9.8, at the equilibrium, it is $\dot{\mathbf{x}} \equiv \mathbf{0}$, $\ddot{\mathbf{x}} \equiv \mathbf{0}$. Then, the input to the block \mathbf{K}_P must be null, which implies

$$\mathbf{x}_d + \mathbf{x}_F - \mathbf{x}_e = \mathbf{0}.$$

In view of (9.21) and (9.41), it follows that

$$\mathbf{x}_d + \mathbf{C}_F(\mathbf{K}(\mathbf{x}_r - \mathbf{x}_e) + \mathbf{f}_d) - \mathbf{x}_e = \mathbf{0},$$

and thus the equilibrium location satisfies the equation

$$\mathbf{x}_e = \mathbf{x}_d + C_F(\mathbf{K}(\mathbf{x}_r - \mathbf{x}_e) + \mathbf{f}_d).$$

Solution to Problem 9.5

The norm of vector

$$\boldsymbol{\epsilon} = \mathbf{h}_e - S_f \boldsymbol{\lambda}$$

with weighting matrix \mathbf{W} is defined as

$$\|\boldsymbol{\epsilon}\|_W = \sqrt{\boldsymbol{\epsilon}^T \mathbf{W} \boldsymbol{\epsilon}}. \quad (\text{S9.3})$$

The vector $\boldsymbol{\lambda}$ which minimizes the norm (S9.3) can be computed also as the solution of a minimization problem for the quadratic cost functional

$$g(\boldsymbol{\epsilon}) = \frac{1}{2} \boldsymbol{\epsilon}^T \mathbf{W} \boldsymbol{\epsilon}.$$

The solution has to satisfy the necessary condition

$$\left(\frac{\partial g}{\partial \boldsymbol{\lambda}} \right)^T = \mathbf{0}$$

which gives

$$-S_f^T \mathbf{W} (\mathbf{h}_e - S_f \boldsymbol{\lambda}) = \mathbf{0},$$

and thus

$$\boldsymbol{\lambda} = (S_f^T \mathbf{W} S_f)^{-1} S_f^T \mathbf{W} \mathbf{h}_e.$$

Solution to Problem 9.6

Taking into account that $C\mathbf{K} = \mathbf{K}C = \mathbf{I}_6$, Equation (9.70) can be rewritten in the form

$$\mathbf{K}' = S_f (S_f^T C S_f)^{-1} S_f^T C \mathbf{K}$$

which, considering the definition (9.55) of weighted pseudo-inverse with weight C , can be rewritten as

$$\mathbf{K}' = S_f S_f^\dagger \mathbf{K} = P_f \mathbf{K}.$$

Solution to Problem 9.7

Assuming that the screwdriver can move along the slot of the screw, it is worth choosing the constraint frame attached to the screwdriver as in Fig. S9.3.

Natural constraints can be determined first. Motion constraints describe the impossibility to generate arbitrary linear velocities along axes y_c , z_c and

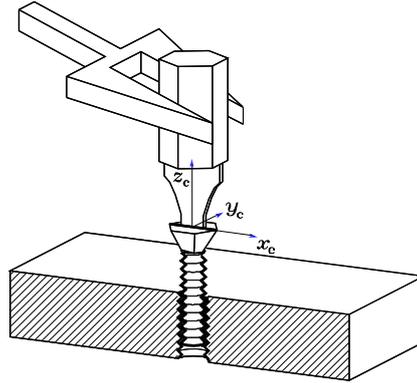


Fig. S9.3. Driving a screw in a hole.

Table S9.1. Natural and artificial constraints for the task of Fig. S9.3.

Natural Constraints	Artificial Constraints
\dot{p}_y^c	f_y^c
\dot{p}_z^c	f_z^c
ω_x^c	μ_x^c
ω_y^c	μ_y^c
f_x^c	\dot{p}_x^c
μ_z^c	ω_z^c

angular velocities about axes x_c , y_c . Force constraints describe the impossibility to exert arbitrary forces along axis x_c and moment about axis z_c .

As a consequence, artificial constraints regard the variables not subject to natural constraints. Hence, with reference to natural velocity constraints along axes y_c , z_c and about x_c , y_c , it is possible to specify artificial constraints for forces along y_c , z_c and moments about x_c , y_c . Also, with reference to natural generalized force constraints along axis y_c and about axis z_c , it is possible to specify artificial constraints for linear velocity along x_c and angular velocity about z_c . The set of constraints is summarized in Table S9.1. Notice that, in the case of a frictionless groove, once ω_z^c is specified, then \dot{p}_z^c is determined according to the pitch of the screw.

Solution to Problem 9.8

The control input (9.77) referred to the constraint frame, in view of the expressions of \mathbf{S}_f^c , \mathbf{S}_v^c and \mathbf{C}'^c in Example 9.4, can be computed as

$$\begin{aligned}\alpha_x^c &= \alpha_\nu \\ \alpha_y^c &= c_{2,2} f_\lambda.\end{aligned}$$

Taking into account (9.81), the control input α_y^c can be expressed as

$$\alpha_y^c = -c_{2,2}k_{D\lambda}\dot{\lambda} + k_{P\lambda}(\lambda_d - \lambda),$$

where a constant desired force λ_d was considered. In view of the equalities $\dot{\delta}_y^c = c_{2,2}\dot{\lambda}$ and $\lambda = f_y^c$, the equation can be rewritten in the form

$$\alpha_y^c = -k_{D\lambda}\dot{\delta}_y^c + k_{P\lambda}(\lambda_d - f_y^c),$$

showing that the the control action in the force controlled subspace consists of a proportional force control with inner velocity loop.

Solution to Problem 9.9

The computation of $\mathbf{S}_f^{c\dagger}$ using the compliance matrix

$$\mathbf{C}^c = \begin{bmatrix} c_{1,1} & c_{1,2} \\ c_{1,2} & c_{2,2} \end{bmatrix}$$

as a weighting matrix gives

$$\mathbf{S}_f^{c\dagger} = (\mathbf{S}_f^{cT} \mathbf{C}^c \mathbf{S}_f^c)^{-1} \mathbf{S}_f^{cT} \mathbf{C}^c = [c_{2,2}^{-1} c_{1,2} \quad 1].$$

On the other hand, computing \mathbf{S}_f^\dagger in the base frame gives

$$\mathbf{S}_f^\dagger = (\mathbf{S}_f^T \mathbf{C} \mathbf{S}_f)^{-1} \mathbf{S}_f^T \mathbf{C} = \frac{1}{\sqrt{2}} \begin{bmatrix} \frac{c_{1,2} - c_{2,2}}{c_{2,2}} & \frac{c_{1,2} + c_{2,2}}{c_{2,2}} \end{bmatrix},$$

where $\mathbf{C} = \mathbf{R}_c \mathbf{C}^c \mathbf{R}_c^T$. Notice that $\mathbf{S}_f^\dagger = \mathbf{S}_f^{c\dagger} \mathbf{R}_c^T$ and the same transformation rule holds for \mathbf{S}_v^\dagger . Also, if $\mathbf{f}_e^c = [f_x^c \quad f_y^c]^T$, then

$$\lambda = \mathbf{S}_f^{c\dagger} \mathbf{f}_e^c = \mathbf{S}_f^\dagger \mathbf{f}_e = \frac{c_{1,2}}{c_{2,2}} f_x^c + f_y^c,$$

independently from the frame to which \mathbf{S}_f^\dagger and \mathbf{f}_e are referred. On the other hand, if the contact force \mathbf{f}_e^c belongs to subspace $\mathcal{R}(\mathbf{S}_f^c)$, i.e., $\mathbf{f}_e^c = [0 \quad f_y^c]^T$, then it is $\lambda = f_y^c$ independently from the particular weighting matrix.

Analogously, the computation of $\mathbf{S}_v^{c\dagger}$ using the stiffness matrix

$$\mathbf{K}^c = \begin{bmatrix} k_{1,1} & k_{1,2} \\ k_{1,2} & k_{2,2} \end{bmatrix}$$

as a weighting matrix gives

$$\mathbf{S}_v^{c\dagger} = (\mathbf{S}_v^{cT} \mathbf{K}^c \mathbf{S}_v^c)^{-1} \mathbf{S}_v^{cT} \mathbf{K}^c = [1 \quad k_{1,1}^{-1} k_{1,2}]$$

in the constraint frame, and

$$\mathbf{S}_v^\dagger = \frac{1}{\sqrt{2}} \begin{bmatrix} \frac{k_{1,1} - k_{1,2}}{k_{1,1}} & \frac{k_{1,1} + k_{1,2}}{k_{1,1}} \end{bmatrix}$$

in the base frame.

Solution to Problem 9.10

In view of the results of Example 9.4, the hybrid control law can be designed by choosing the control inputs α_ν and f_λ according to (9.80) and (9.84), respectively, with suitable control gains.

In detail, the control gains in (9.80) can be set to

$$k_{P\nu} = 16 \quad k_{I\nu} = 100,$$

corresponding to $\omega_{n\nu} = 10$ rad/s and $\zeta_\nu = 0.8$ for the dynamics of the velocity error $\nu_d - \nu$.

The same dynamics can be imposed to the force λ with the choice

$$k_{D\lambda} = 16 \quad k_{P\lambda} = 100,$$

in the case that the stiffness of the environment is known.

In the case that only an estimate of the stiffness of the environment is available, the quantity $\mathbf{L}_f = l_f$ has the value

$$l_f = \frac{\hat{c}_{2,2}}{c_{2,2}}$$

and thus

$$\dot{\lambda} = \hat{c}_{2,2}^{-1} \delta y^c.$$

Assuming, for example, $\hat{c}_{2,2}^{-1} = 4 \cdot 10^3$ N/m, then $l_f = 1.25$, corresponding to $\omega_{n\lambda} = 11.18$ rad/s and $\zeta_\lambda = 0.71$ for the dynamics of the force.

The initial tip position is chosen on the plane as $\mathbf{o}_e(0) = [1 \ 0]^T$. The desired velocity along axis x_c is set as in Problem 9.3, while the desired force along axis y_c is set to -50 N.

The resulting SIMULINK block diagram is shown in Fig. S9.4. The arm is simulated as a continuous-time system using a variable-step integration method with a minimum step size of 1 ms. All the blocks of the controller are simulated as discrete-time subsystems with the given sampling time of 1 ms.

The files with the solution can be found in Folder 9_10.

The resulting tip velocity error along x_c and contact force along y_c are shown in Fig. S9.5. It can be verified that the velocity error keeps small during task execution; also, the contact force reaches the desired value at steady state. In Fig. S9.6, the continuous line represents the end-effector path in the base frame, while the dashed line corresponds to the plane at rest. It can be verified that the plane complies during the interaction.

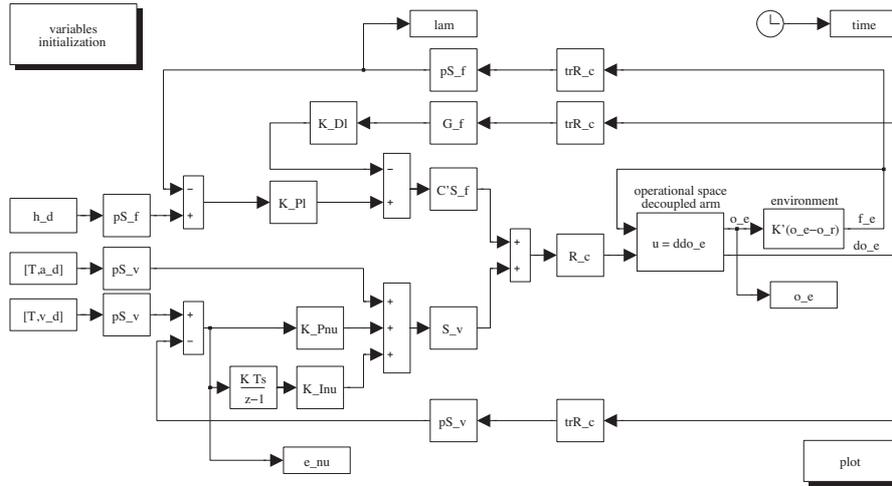


Fig. S9.4. SIMULINK block diagram of hybrid force/motion control.

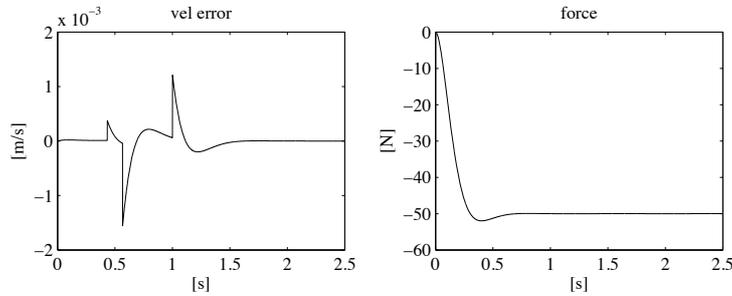


Fig. S9.5. Time history of the tip velocity error and of the contact force with hybrid force/motion control.

Solution to Problem 9.11

In view of equality $\dot{\mathbf{q}} = \mathbf{J}_\rho(\mathbf{r})\dot{\mathbf{r}}$ and (9.63) the following equality holds

$$\mathbf{v}_e = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} = \mathbf{J}(\mathbf{q})\mathbf{J}_\rho(\mathbf{r})\dot{\mathbf{r}} = \mathbf{S}_v(\mathbf{q})\dot{\mathbf{r}}.$$

Hence, pre-multiplying both sides of the above equation by \mathbf{S}_v^\dagger and using (9.59) gives

$$\boldsymbol{\nu} = \dot{\mathbf{r}}.$$

Also, it is $\dot{\boldsymbol{\nu}} = \ddot{\mathbf{r}}$. Therefore, control law (9.95) yields the closed-loop equation

$$\ddot{\mathbf{r}} - \ddot{\mathbf{r}}_d + \mathbf{K}_{Dr}(\dot{\mathbf{r}}_d - \dot{\mathbf{r}}) + \mathbf{K}_{Pr}(\mathbf{r}_d - \mathbf{r}) = \mathbf{0},$$

showing that tracking of a desired position $\mathbf{r}_d(t)$ is achieved.

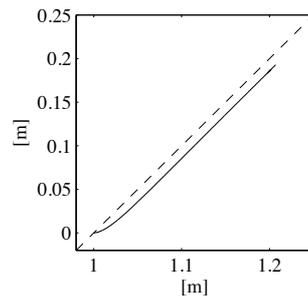


Fig. S9.6. Tip's path in the base frame (continuous line) and undeformed contact plane (dashed line).

Visual servoing

Solution to Problem 10.1

The principal moments of the region are the eigenvalues of matrix \mathcal{I} which are

$$\lambda = \frac{1}{2} \left(\mu_{2,0} + \mu_{0,2} + \sqrt{4\mu_{1,1}^2 + (\mu_{0,2} - \mu_{2,0})^2} \right)$$

$$\lambda' = \frac{1}{2} \left(\mu_{2,0} + \mu_{0,2} - \sqrt{4\mu_{1,1}^2 + (\mu_{0,2} - \mu_{2,0})^2} \right)$$

with $\lambda \geq \lambda' > 0$. The principal axis (i.e., the eigenvector of \mathcal{I}) corresponding to λ is aligned to vector

$$\mathbf{a}_\lambda = \begin{bmatrix} \mu_{1,1} \\ \lambda - \mu_{2,0} \end{bmatrix}.$$

Therefore

$$t_\alpha = \frac{1}{2\mu_{1,1}} \left(\mu_{0,2} - \mu_{2,0} + \sqrt{4\mu_{1,1}^2 + (\mu_{0,2} - \mu_{2,0})^2} \right),$$

where $t_\alpha = \tan \alpha$. Considering that

$$t_{2\alpha} = \frac{2t_\alpha}{1 - t_\alpha^2} = \frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}},$$

Equation (10.2) follows.

Solution to Problem 10.2

The singular value decomposition of the (12×9) matrix \mathbf{A} is given by

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \quad (\text{S10.1})$$

where \mathbf{U} is a (12×12) orthogonal matrix, \mathbf{V} is a (9×9) orthogonal matrix and

$$\boldsymbol{\Sigma} = \begin{bmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{O} & \mathbf{0} \end{bmatrix} \quad \mathbf{D} = \text{diag}\{\sigma_1, \sigma_2, \dots, \sigma_8\}$$

with $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_8 > 0$ being the non-null singular values of matrix \mathbf{A} , which is of rank 8. Post-multiplying by \mathbf{V} both sides of Eq. (S10.1) gives

$$\mathbf{AV} = \mathbf{U}\boldsymbol{\Sigma}$$

and thus $\mathbf{A}\mathbf{v}_i = \sigma_i\mathbf{u}_i$, for $i = 1, \dots, 8$ and $\mathbf{A}\mathbf{v}_9 = \mathbf{0}$, being \mathbf{v}_i the i -th column of \mathbf{V} , known as right eigenvector, and \mathbf{u}_i the i -th column of \mathbf{U} , known as left eigenvector, corresponding to σ_i . In sum, the right eigenvector \mathbf{v}_9 , corresponding to the null singular value of \mathbf{A} , is a non-null solution to (10.10).

Solution to Problem 10.3

Notice that equality $\text{Tr}(\mathbf{R}_o^{cT}\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T) = \text{Tr}(\mathbf{V}^T\mathbf{R}_o^{cT}\mathbf{U}\boldsymbol{\Sigma})$ derives from property $\text{Tr}(\mathbf{AB}) = \text{Tr}(\mathbf{BA})$, for square matrices \mathbf{A} and \mathbf{B} . Also, since $\mathbf{V}^T\mathbf{R}_o^{cT}\mathbf{U}$ is an orthogonal matrix, all its elements have absolute value lower than or equal to 1.

The problem of minimizing the Frobenius norm is equivalent to that of minimizing the quantity

$$\|\mathbf{R}_o^c - \mathbf{Q}\|_F^2 = \text{Tr}((\mathbf{R}_o^c - \mathbf{Q})^T(\mathbf{R}_o^c - \mathbf{Q})) = 3 + \text{Tr}(\mathbf{Q}^T\mathbf{Q}) - 2\text{Tr}(\mathbf{R}_o^{cT}\mathbf{Q}),$$

which is minimum when $\text{Tr}(\mathbf{R}_o^{cT}\mathbf{Q})$ is maximum. Consider the singular value decomposition

$$\mathbf{Q} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$$

where $\boldsymbol{\Sigma} = \text{diag}\{\sigma_1, \sigma_2, \sigma_3\}$, $\sigma_i > 0$. The following equality holds

$$\text{Tr}(\mathbf{R}_o^{cT}\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T) = \text{Tr}(\mathbf{V}^T\mathbf{R}_o^{cT}\mathbf{U}\boldsymbol{\Sigma}) = \sum_{i=1}^3 z_{i,i}\sigma_i$$

where $z_{i,i}$, $i = 1, 2, 3$ are the diagonal elements of the orthogonal matrix $\mathbf{V}^T\mathbf{R}_o^{cT}\mathbf{U}$, whose absolute values are lower than or equal to 1. Thus

$$\text{Tr}(\mathbf{R}_o^{cT}\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T) \leq \sum_{i=1}^3 \sigma_i$$

and the equality holds when $z_{i,i} = 1$. Hence the maximum of $\text{Tr}(\mathbf{R}_o^{cT}\mathbf{Q})$ is achieved by choosing \mathbf{R}_c as in (10.12), which ensures that $\mathbf{V}^T\mathbf{R}_o^{cT}\mathbf{U} = \mathbf{I}$.

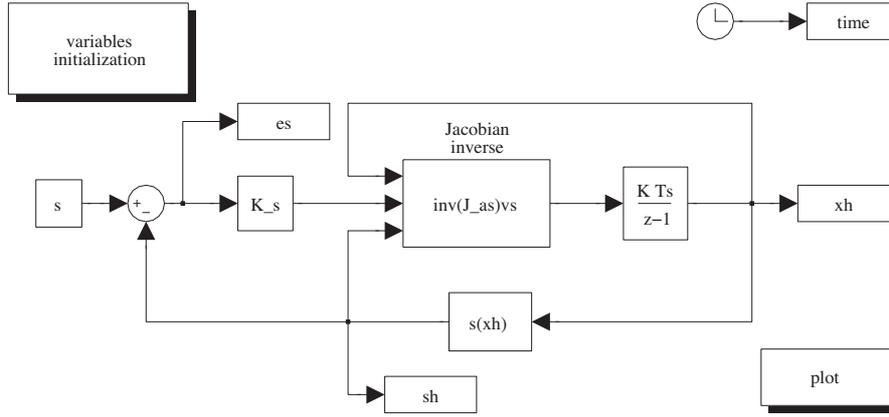


Fig. S10.1. SIMULINK block diagram of the pose estimation algorithm based on the inverse of image Jacobian. Case of two feature points.

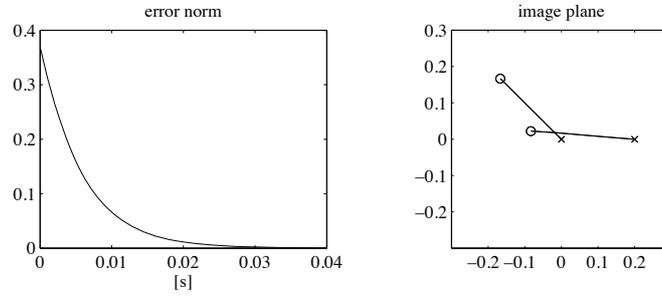


Fig. S10.2. Time history of the norm of the estimation error and corresponding paths of the feature points projections on image plane

Solution to Problem 10.4

In view of the results of Example 10.3, the Jacobian \mathbf{J}_{A_s} , based on the feature vector of two points, is a square (4×4) matrix. Hence, the pose estimation algorithm based on the inverse of image Jacobian can be adopted.

According to (10.40) and (10.41), the following equality holds

$$\hat{\mathbf{x}}_{c,o} = \hat{\mathbf{v}}_{c,o}^c = \mathbf{\Gamma}(\mathbf{o}_{c,o}^c) \mathbf{L}_s^{-1} \mathbf{K}_s \mathbf{e}_s,$$

where \mathbf{L}_s is the (4×4) interaction matrix of points P_1 and P_2 , obtained from the (4×6) interaction matrix of a set of two points by removing the fourth and fifth column. Also, $\mathbf{\Gamma}(\mathbf{o}_{c,o}^c)$ is the (4×4) matrix

$$\mathbf{\Gamma}(\mathbf{o}_{c,o}^c) = \begin{bmatrix} -\mathbf{I} & \mathbf{s}_3(\mathbf{o}_{c,o}^c) \\ \mathbf{0}^T & -1 \end{bmatrix},$$

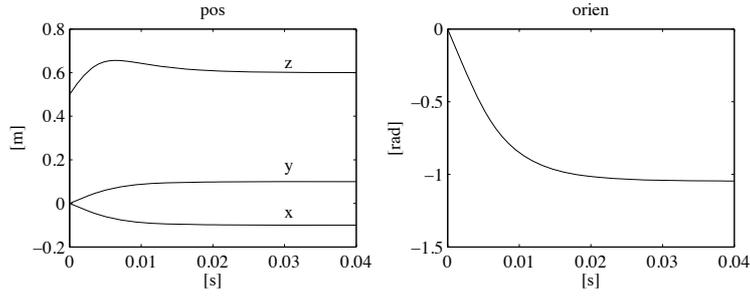


Fig. S10.3. Time history of camera pose estimate

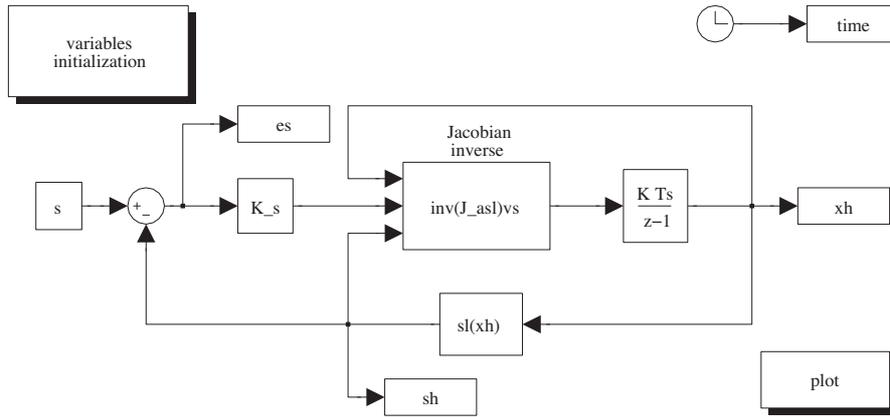


Fig. S10.4. SIMULINK block diagram of the pose estimation algorithm based on the inverse of image Jacobian. Case of two feature points.

being $s_3(\cdot)$ is the third column of matrix $S(\cdot)$.

Matrix gain K_s has been chosen as $K_s = 160I_4$. The initial estimate has been set to $\hat{x}_{c,o}(0) = [0 \ 0 \ 0.5 \ 0]$. The resulting SIMULINK block diagram is shown in Fig. S10.1. The Euler numerical integration method has been adopted, with sampling time of 1 ms.

The files with the solution can be found in Folder 10.4.

The results in Fig. S10.2 show that the norm of the estimation error of the feature parameters e_s tends to zero asymptotically with convergence of exponential type; moreover, due to the fact that matrix gain K_s was chosen diagonal with equal elements, the paths of the projections of the feature points on the image plane (between the initial positions marked with crosses and the final positions marked with circles) are line segments.

The corresponding time histories of the components of vector $\hat{x}_{c,o}$ for position and orientation are reported in Fig. S10.3. It can be verified that, with the chosen value of K_s , the algorithm converges in about 0.03 s, corresponding

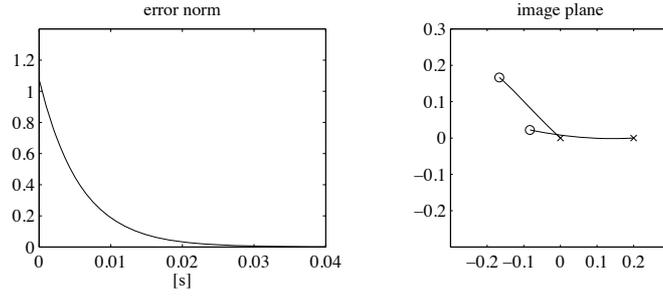


Fig. S10.5. Time history of the norm of the estimation error and corresponding paths of the feature points projections on image plane

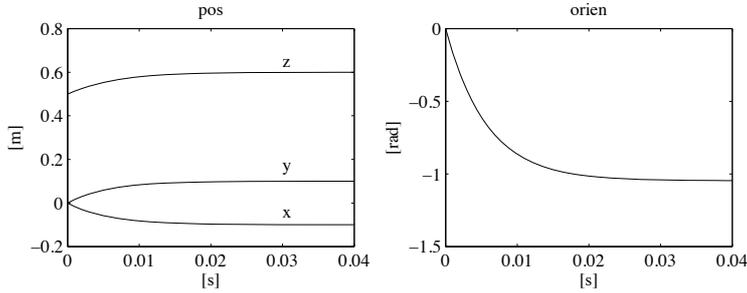


Fig. S10.6. Time history of camera pose estimate

to 30 iterations. The final value, namely, the estimated relative pose of the object with respect to the camera, is $\hat{\mathbf{x}}_{c,o} \simeq [-0.1 \ 0.1 \ 0.6 \ -\pi/3]^T$.

Solution to Problem 10.5

The solution is similar to that of Problem 10.4, but the feature parameters are defined according to (10.28), and the interaction matrix of a line segment is used.

Matrix gain \mathbf{K}_s has been chosen as $\mathbf{K}_s = 160\mathbf{I}_4$. The initial estimate has been set to $\hat{\mathbf{x}}_{c,o} = [0 \ 0 \ 0.5 \ 0]$. The feature vector of the line segment corresponding to that of Problem 10.4 is

$$\mathbf{s} = [-0.1250 \ 0.0945 \ 0.1668 \ -1.0470]^T.$$

The resulting SIMULINK block diagram is shown in Fig. S10.4. The Euler numerical integration method has been adopted, with sampling time of 1 ms.

The files with the solution can be found in Folder 10_5.

The results in Figs. S10.5 and S10.6 are quite similar to those of Problem 10.4. Notice that, however, the paths of the projections of the feature points on the image plane are not perfectly linear, due to the different choice

of feature vector. This is reflected also by a different time history of the z component of the camera position during the transient. Obviously, the same final value is achieved for the estimated relative pose of the object with respect to the camera, namely $\hat{\mathbf{x}}_{c,o} \simeq [-0.1 \ 0.1 \ 0.6 \ -\pi/3]^T$.

Solution to Problem 10.6

In view of the equalities $\mathbf{p}_i = \bar{\mathbf{p}}_i + \bar{\mathbf{p}}$ and $\mathbf{p}'_i = \bar{\mathbf{p}}'_i + \bar{\mathbf{p}}'$, function (10.55) can be rewritten as

$$\begin{aligned} g(\mathbf{o}) &= \sum_{i=1}^n \|\mathbf{p}_i - \mathbf{o} - \mathbf{R}\mathbf{p}'_i\|^2 = n\|\bar{\mathbf{p}} - \mathbf{o} - \mathbf{R}\bar{\mathbf{p}}'\|^2 + \sum_{i=1}^n \|\bar{\mathbf{p}}_i - \mathbf{R}\bar{\mathbf{p}}'_i\|^2 \\ &\quad + 2\sum_{i=1}^n (\bar{\mathbf{p}} - \mathbf{o} - \mathbf{R}\bar{\mathbf{p}}')^T (\bar{\mathbf{p}}_i - \mathbf{R}\bar{\mathbf{p}}'_i) \\ &= n\|\bar{\mathbf{p}} - \mathbf{o} - \mathbf{R}\bar{\mathbf{p}}'\|^2 + \sum_{i=1}^n \|\bar{\mathbf{p}}_i - \mathbf{R}\bar{\mathbf{p}}'_i\|^2, \end{aligned}$$

considering that $\sum_{i=1}^n \bar{\mathbf{p}}_i = \sum_{i=1}^n \bar{\mathbf{p}}'_i = \mathbf{0}$. Hence, with the choice of \mathbf{o} as in (10.56), function (10.55) is minimum. The same result can be obtained computing the value of \mathbf{o} which satisfies the necessary condition $(\partial g/\partial \mathbf{o})^T = \mathbf{0}$.

Solution to Problem 10.7

Function (10.57) can be rewritten as

$$\sum_{i=1}^n \|\bar{\mathbf{p}}_i - \mathbf{R}\bar{\mathbf{p}}'_i\|^2 = \sum_{i=1}^n \|\bar{\mathbf{p}}_i\|^2 - 2\sum_{i=1}^n \bar{\mathbf{p}}'^T_i \mathbf{R}^T \bar{\mathbf{p}}_i + \sum_{i=1}^n \|\mathbf{R}\bar{\mathbf{p}}'_i\|^2$$

which is minimum when the quantity $\sum_{i=1}^n \bar{\mathbf{p}}'^T_i \mathbf{R}^T \bar{\mathbf{p}}_i$ is maximum. This quantity, in view of the property $\mathbf{a}^T \mathbf{b} = \text{Tr}(\mathbf{b}\mathbf{a})$, can be rewritten as

$$\sum_{i=1}^n \bar{\mathbf{p}}'^T_i \mathbf{R}^T \bar{\mathbf{p}}_i = \sum_{i=1}^n \text{Tr}(\mathbf{R}^T \bar{\mathbf{p}}_i \bar{\mathbf{p}}'^T_i) = \text{Tr}(\mathbf{R}^T \mathbf{K}),$$

with $\mathbf{K} = \sum_{i=1}^n \bar{\mathbf{p}}_i \bar{\mathbf{p}}'^T_i$. Hence, the matrix \mathbf{R} which minimizes function (10.57) is the matrix which maximizes the trace of $\mathbf{R}^T \mathbf{K}$.

Solution to Problem 10.8

The control scheme in Fig. 10.14 is adopted, with the same matrix gains used in control scheme **A** of Sect. 10.9, namely,

$$\mathbf{K}_P = \text{diag}\{500, 500, 10, 10\} \quad \mathbf{K}_D = \text{diag}\{500, 500, 10, 10\}.$$

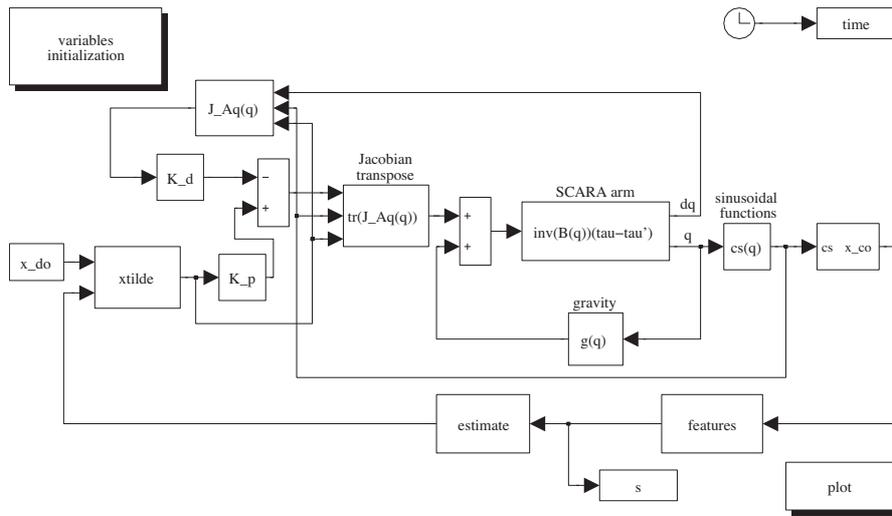


Fig. S10.7. SIMULINK block diagram of the position-based visual servoing scheme of PD type with gravity compensation

The initial pose of the camera frame is $\mathbf{x}_c(0) = [1 \ 1 \ 0.5 \ \pi/4]^T$ and the desired pose of the object frame with respect to the camera frame has been set to $\mathbf{x}_{d,o} = [-0.1 \ 0.1 \ 0.6 \ -\pi/3]^T$. The various terms of the dynamic model of the SCARA manipulator are computed as in the solution to Problem 7.3.

The pose estimation algorithm is based on the measurements of the features of the line segment P_1P_2 , as in the solution to Problem 10.5, using the same matrix gain, namely, $\mathbf{K}_s = 160\mathbf{I}_4$. The initial pose estimate is chosen as $\hat{\mathbf{x}}_{c,o}(0) = \mathbf{x}_{d,o}$.

The resulting SIMULINK block diagram is shown in Fig. S10.7. The arm is simulated as a continuous-time system using a variable-step integration method with a maximum step size of 1 ms. All the blocks of the controller are simulated as discrete-time subsystems with the given sampling time of 4 ms, while the sampling time of the estimation algorithm has been set to 1 ms.

The files with the solution can be found in Folder 10_8.

The results of Figs. S10.8 and S10.9 (right) are very similar to those of Figs. 10.18 and 10.19 (right), as expected. In fact, the two control schemes differ only for the estimation algorithms which, in both cases, have fast dynamics with negligible effects on the control loop dynamics.

Notice that, in Fig. S10.9 (left), the time history of the feature parameters of the line segment P_1P_2 used by the estimation algorithm in Fig. S10.7 is reported.

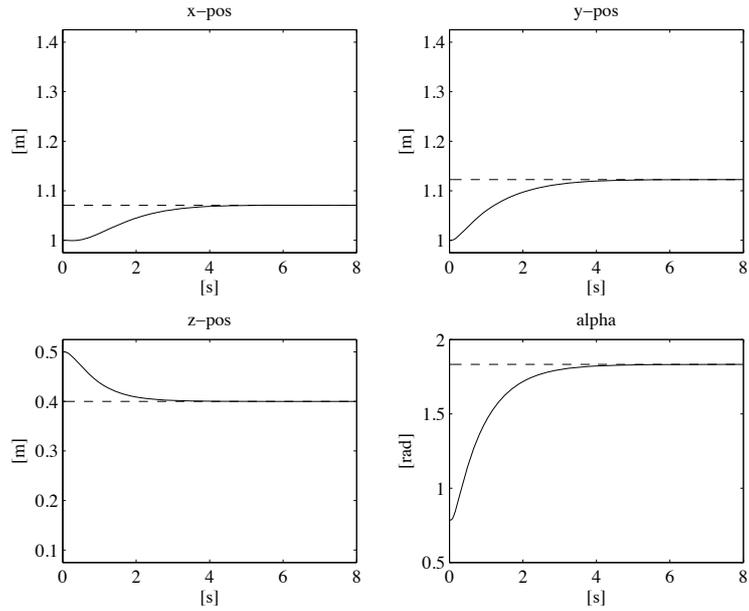


Fig. S10.8. Time history of camera frame position and orientation with position-based visual servoing scheme of PD type with gravity compensation

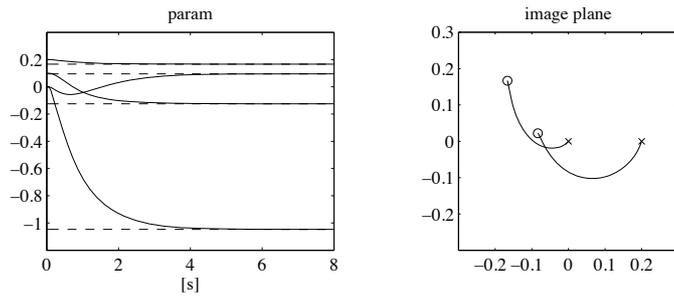


Fig. S10.9. Time history of feature parameters and corresponding path of feature points projections on image plane with position-based visual servoing scheme of PD type with gravity compensation

Solution to Problem 10.9

The control scheme in Fig. 10.17 is adopted, with the same matrix gain used in control scheme **B** of Sect. 10.9, namely, $\mathbf{K} = \mathbf{I}_4$.

The initial pose of the camera frame is $\mathbf{x}_c(0) = [1 \ 1 \ 0.5 \ \pi/4]^T$, and the desired pose of the object frame with respect to the camera frame has been set to $\mathbf{x}_{d,o} = [-0.1 \ 0.1 \ 0.6 \ -\pi/3]^T$.

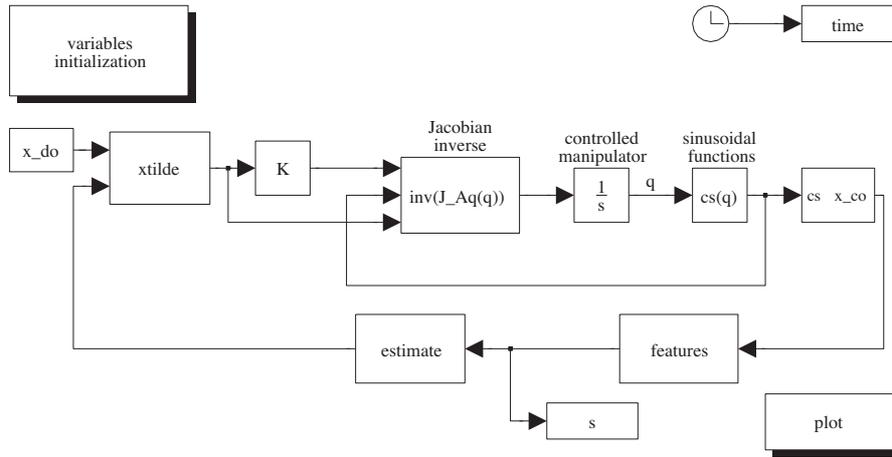


Fig. S10.10. SIMULINK block diagram of the resolved-velocity position-based visual servoing scheme

The pose estimation algorithm is based on the measurements of the features of the line segment P_1P_2 , as in the solution to Problem 10.5, using the same matrix gain, namely, $\mathbf{K}_s = 160\mathbf{I}_4$. The initial pose estimate is chosen as $\hat{\mathbf{x}}_{c,o}(0) = \mathbf{x}_{d,o}$.

The resulting SIMULINK block diagram is shown in Fig. S10.10. Notice that the dynamics of the controlled robot manipulator is approximated by a simple integrator. All the blocks of the controller are simulated as discrete-time subsystems with the given sampling time of 4 ms, while the sampling time of the estimation algorithm has been set to 1 ms.

The files with the solution can be found in Folder 10_9.

The results of Figs. S10.11 and S10.12 (right) are very similar to those of Figs 10.20 and 10.21 (right), as expected (see also the solution to Problem 10.8).

Solution to Problem 10.10

The control scheme in Fig. 10.16 is adopted with the matrix gains

$$\mathbf{K}_{Ps} = \text{diag}\{300, 300, 150, 50\} \quad \mathbf{K}_{Ds} = \text{diag}\{330, 330, 165, 40\},$$

which guarantee a closed-loop dynamics similar to that of control scheme **C** in Sect. 10.9. Notice that, in this case, the dynamics of the closed-loop depends on the type of feature parameters extracted from visual measurements.

The various terms of the dynamic model of the SCARA manipulator are computed as in the solution to Problem 7.3.

The initial pose of the camera frame is $\mathbf{x}_c(0) = [1 \ 1 \ 0.5 \ \pi/4]^T$, and the desired value of the feature parameters of the line segment P_1P_2 has

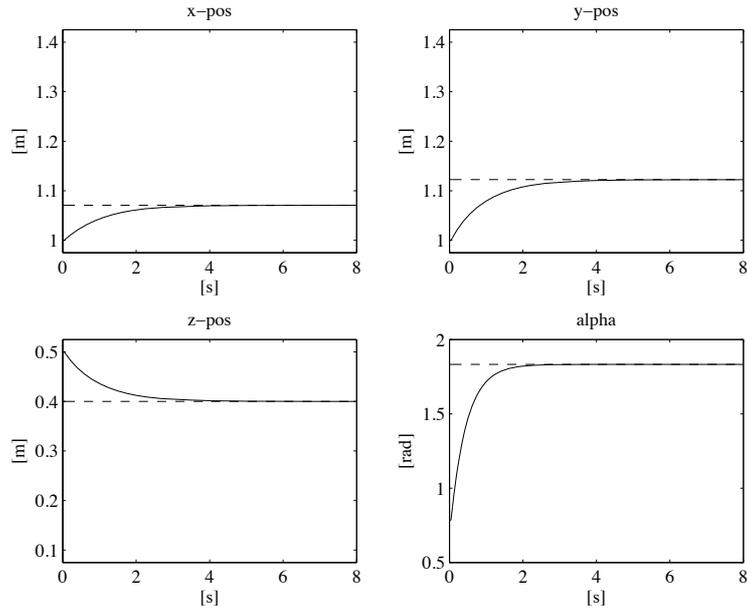


Fig. S10.11. Time history of camera frame position and orientation with resolved-velocity position-based visual servoing

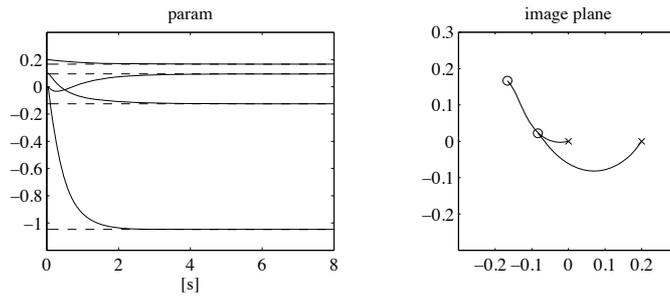


Fig. S10.12. Time history of feature parameters and corresponding path of feature points projections on image plane with resolved-velocity position-based visual servoing

been set to $\mathbf{s}_d = [-0.1250 \ 0.0945 \ 0.1667 \ -1.0472]^T$, corresponding to the desired pose $\mathbf{x}_{d,o}$.

Matrix $\mathbf{L}_s(\mathbf{s}, \mathbf{z}_c)$ has been approximated with matrix $\hat{\mathbf{L}}_s = \mathbf{L}_s(\mathbf{s}, \mathbf{z}_d)$, where \mathbf{z}_d is the third component of vector $\mathbf{x}_{d,o}$.

The resulting SIMULINK block diagram is shown in Fig. S10.13. The arm is simulated as a continuous-time system using a variable-step integration method with a maximum step size of 1 ms. All the blocks of the controller are

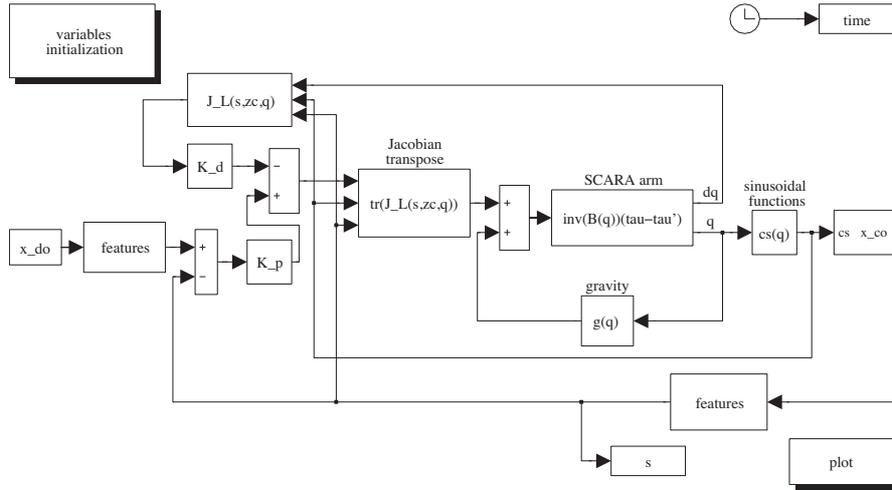


Fig. S10.13. SIMULINK block diagram of the image-based visual servoing scheme of PD type with gravity compensation

simulated as discrete-time subsystems with the given sampling time of 4 ms, while the sampling time of the estimation algorithm has been set to 1 ms.

The files with the solution can be found in Folder 10_10.

The results are shown in Figs. S10.14 and S10.15. With respect to the results of Figs 10.22 and 10.23, it can be observed that the phenomenon of camera retreat is mitigated.

Solution to Problem 10.11

The control scheme in Fig. 10.17 is adopted with the same matrix gain used in control scheme C of Sect. 10.9, namely, $K_s = I_4$.

The initial pose of the camera frame is $x_c(0) = [1 \ 1 \ 0.5 \ \pi/4]^T$, and the desired value of the feature parameters of the line segment P_1P_2 has been set to $s_d = [-0.1250 \ 0.0945 \ 0.1667 \ -1.0472]^T$, corresponding to the desired pose $x_{d,o}$.

Matrix $L_s(s, z_c)$ has been approximated with matrix $\hat{L}_s = L_s(s, z_d)$, where z_d is the third component of vector $x_{d,o}$.

The resulting SIMULINK block diagram is shown in Fig. S10.16. Notice that the dynamics of the controlled robot manipulator is approximated by a simple integrator. All the blocks of the controller are simulated as discrete-time subsystems with the given sampling time of 4 ms, while the sampling time of the estimation algorithm has been set to 1 ms.

The files with the solution can be found in Folder 10_11.

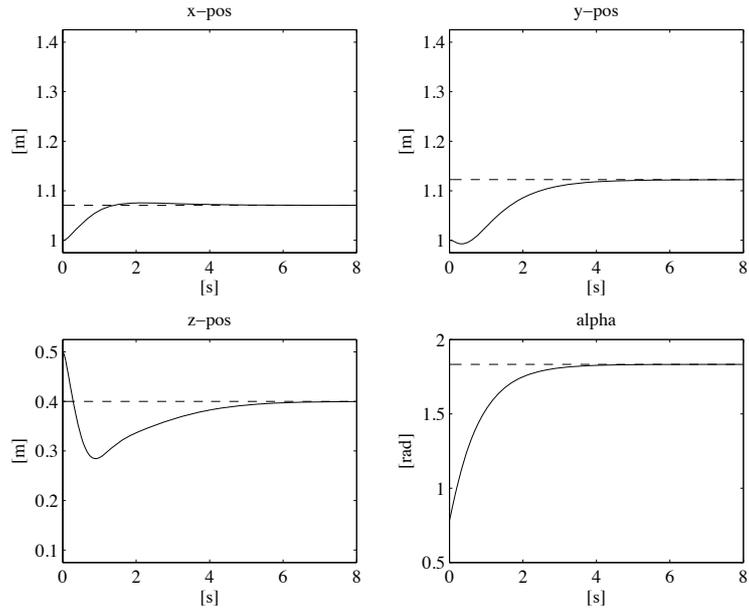


Fig. S10.14. Time history of camera frame position and orientation with image-based visual servoing scheme of PD type with gravity compensation

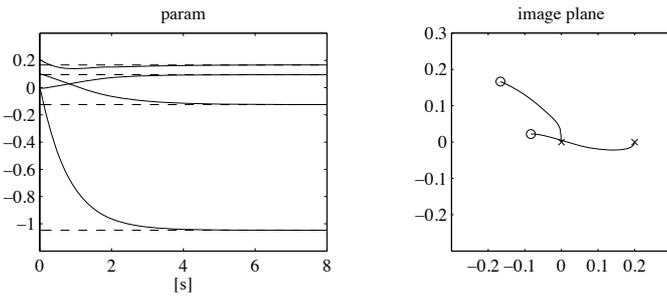


Fig. S10.15. Time history of feature parameters and corresponding path of feature points projections on image plane with image-based visual servoing scheme of PD type with gravity compensation

The results of Figs. S10.17 and S10.18 (right), with the given choice of the control gains, are quite similar to those of Figs 10.22 and 10.23 (right). However, it can be noticed the absence of the phenomenon of camera retreat.

Solution to Problem 10.12

The distance d_c can be computed as

$$d_c = \mathbf{n}^{cT} \mathbf{r}_c^c = z_c \mathbf{n}^{cT} \tilde{\mathbf{s}}_p,$$

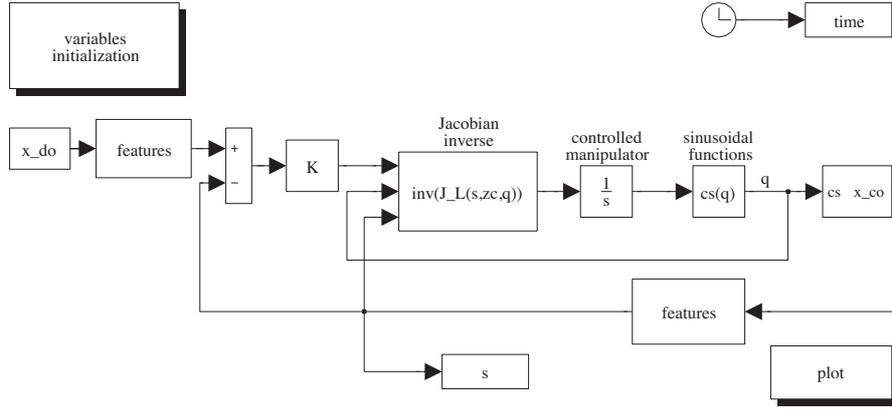


Fig. S10.16. SIMULINK block diagram of the resolved-velocity image-based visual servoing scheme

and thus

$$z_c = \frac{d_c}{\mathbf{n}^{cT} \tilde{\mathbf{s}}_p}.$$

Analogously, the following equality can be derived

$$z_d = \frac{d_d}{\mathbf{n}^{dT} \tilde{\mathbf{s}}_{p,d}}.$$

Hence, expression (10.94) follows directly.

In order to derive (10.95), consider that the distance d_c can be computed also as

$$d_c = d_d + \mathbf{n}^{cT} \mathbf{o}_{c,d}^c,$$

and thus

$$\frac{d_c}{d_d} = 1 + \mathbf{n}^{cT} \frac{\mathbf{o}_{c,d}^c}{d_d}.$$

Moreover, the following equality holds

$$\begin{aligned} \det(\mathbf{H}) &= \det\left(\mathbf{R}_d^c + \frac{1}{d_d} \mathbf{o}_{c,d}^c \mathbf{n}^{dT}\right) = \det\left(\left(\mathbf{I} + \frac{1}{d_d} \mathbf{o}_{c,d}^c \mathbf{n}^{cT}\right) \mathbf{R}_d^c\right) \\ &= \det\left(\mathbf{I} + \frac{1}{d_d} \mathbf{o}_{c,d}^c \mathbf{n}^{cT}\right). \end{aligned}$$

By direct computation, it is possible to verify the identity

$$\det\left(\mathbf{I} + \frac{1}{d_d} \mathbf{o}_{c,d}^c \mathbf{n}^{cT}\right) = 1 + \mathbf{n}^{cT} \frac{\mathbf{o}_{c,d}^c}{d_d},$$

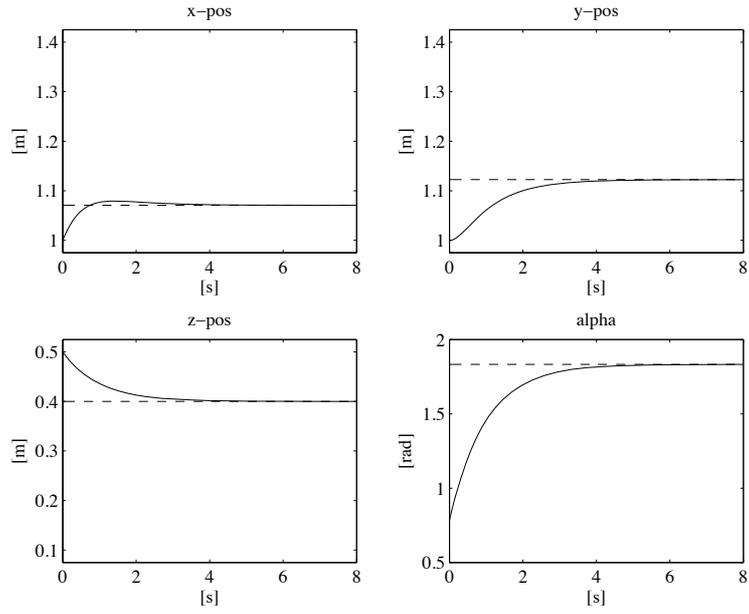


Fig. S10.17. Time history of camera frame position and orientation with resolved-velocity image-based visual servoing

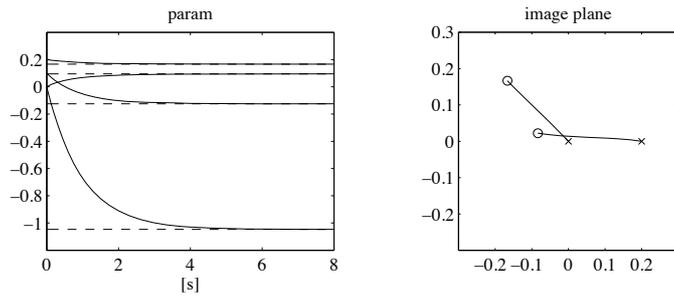


Fig. S10.18. Time history of feature parameters and corresponding path of feature points projections on image plane with resolved-velocity image-based visual servoing

and thus

$$\frac{d_c}{d_d} = \det(\mathbf{H}).$$

Solution to Problem 10.13

Notice that

$$\frac{\partial \mathbf{e}_p(\mathbf{r}_c^c)}{\partial \mathbf{r}_c^c} = \frac{z_c}{z_d \rho_z} \begin{bmatrix} -\frac{1}{z_c} & 0 & \frac{x_c}{z_c^2} \\ 0 & -\frac{1}{z_c} & \frac{y_c}{z_c^2} \\ 0 & 0 & -\frac{1}{z_c} \end{bmatrix} = \frac{1}{z_d \rho_z} \begin{bmatrix} -1 & 0 & X \\ 0 & -1 & Y \\ 0 & 0 & -1 \end{bmatrix}$$

$$\mathbf{S}(\mathbf{r}_c^c) = z_c \begin{bmatrix} 0 & -1 & Y \\ 1 & 0 & -X \\ -Y & X & 0 \end{bmatrix}.$$

Hence, in view of (10.26) and (10.96), the following equalities hold

$$\mathbf{J}_p = \frac{\partial \mathbf{e}_p(\mathbf{r}_c^c)}{\partial \mathbf{r}_c^c} = \frac{1}{z_d \rho_z} \begin{bmatrix} -1 & 0 & X \\ 0 & -1 & Y \\ 0 & 0 & -1 \end{bmatrix}$$

$$\mathbf{J}_o = -\frac{\partial \mathbf{e}_p(\mathbf{r}_c^c)}{\partial \mathbf{r}_c^c} \mathbf{S}(\mathbf{r}_c^c) = \begin{bmatrix} XY & -1 - X^2 & Y \\ 1 + Y^2 & -XY & -X \\ -Y & X & 0 \end{bmatrix}.$$

Mobile Robots

Solution to Problem 11.1

Denote by (x, y) the Cartesian coordinates of a representative point on the tricycle (e.g., the midpoint of the rear wheel axle), by ϕ the steering angle of the front wheel with respect to the vehicle, and by θ_0 the orientation of the vehicle with respect to the x axis. Once the configuration of the tricycle and of the first $i - 1$ trailers are given, the configuration of the i -th trailer is completely described by its orientation θ_i with respect to the x axis. The configuration vector for the complete vehicle is therefore obtained as $\mathbf{q} = [x \ y \ \phi \ \theta_0 \ \theta_1 \ \dots \ \theta_N]^T$, and takes values in $\mathbb{R}^2 \times SO(2) \times \dots \times SO(2)$ (with $SO(2)$ appearing $N + 2$ times). Another possibility is to replace the absolute angle θ_i with the relative angle $\theta_i - \theta_{i-1}$ to describe the orientation of each trailer.

Solution to Problem 11.2

Add the kinematic constraints side-by-side to obtain

$$3\ell \dot{q}_3 + r(\dot{q}_4 + \dot{q}_5 + \dot{q}_6) = 0$$

that can be integrated as

$$q_3 = -\frac{r}{3\ell}(q_4 + q_5 + q_6) + c,$$

where c is an integration constant. The kinematic constraints are then partially integrable. In particular, the orientation q_3 of the robot is a linear function of the wheel rotation angles q_4, q_5, q_6 .

Solution to Problem 11.3

Consider a single Pfaffian constraint

$$\mathbf{a}^T(\mathbf{q})\dot{\mathbf{q}} = \sum_{j=1}^n a_j(\mathbf{q})\dot{q}_j = 0.$$

The corresponding kinematic model is

$$\dot{\mathbf{q}} = \sum_{j=1}^{n-1} \mathbf{g}_j(\mathbf{q}) u_j,$$

where $\{\mathbf{g}_1(\mathbf{q}), \dots, \mathbf{g}_{n-1}(\mathbf{q})\}$ is a basis of $\mathcal{N}(\mathbf{a}^T(\mathbf{q}))$. Without loss of generality, assume that $a_1(\mathbf{q}) \neq 0$, and consider the following choice of input vector fields

$$\mathbf{g}_1 = \begin{bmatrix} -\gamma(\mathbf{q})a_2(\mathbf{q}) \\ \gamma(\mathbf{q})a_1(\mathbf{q}) \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{g}_2 = \begin{bmatrix} -\gamma(\mathbf{q})a_3(\mathbf{q}) \\ 0 \\ \gamma(\mathbf{q})a_1(\mathbf{q}) \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \quad \dots \quad \mathbf{g}_{n-1} = \begin{bmatrix} -\gamma(\mathbf{q})a_n(\mathbf{q}) \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ \gamma(\mathbf{q})a_1(\mathbf{q}) \end{bmatrix}$$

where $\gamma(\mathbf{q}) \neq 0$. The distribution $\Delta = \text{span}\{\mathbf{g}_1, \dots, \mathbf{g}_{n-1}\}$ is involutive if any Lie bracket $[\mathbf{g}_i, \mathbf{g}_j]$ can be written as a linear combination of $\mathbf{g}_1, \dots, \mathbf{g}_{n-1}$.

For example, consider $[\mathbf{g}_1, \mathbf{g}_2]$. A simple computation gives

$$[\mathbf{g}_1, \mathbf{g}_2] = \begin{bmatrix} \frac{\partial(\gamma a_2)}{\partial q_1} \gamma a_3 - \frac{\partial(\gamma a_3)}{\partial q_1} \gamma a_2 + \left(\frac{\partial(\gamma a_3)}{\partial q_2} - \frac{\partial(\gamma a_2)}{\partial q_3} \right) \gamma a_1 \\ \frac{\partial(\gamma a_1)}{\partial q_3} \gamma a_1 - \frac{\partial(\gamma a_1)}{\partial q_1} \gamma a_3 \\ \frac{\partial(\gamma a_1)}{\partial q_1} \gamma a_2 - \frac{\partial(\gamma a_1)}{\partial q_2} \gamma a_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Using the integrability condition expressed by (11.9), one easily verifies that

$$[\mathbf{g}_1, \mathbf{g}_2] = \frac{\partial(\gamma a_3)}{\partial q_1} \mathbf{g}_1 - \frac{\partial(\gamma a_2)}{\partial q_1} \mathbf{g}_2 + \frac{\partial(\gamma a_1)}{\partial q_1} \frac{a_3 \mathbf{g}_1 + a_2 \mathbf{g}_2}{a_1},$$

i.e., $[\mathbf{g}_1, \mathbf{g}_2]$ is a linear combination of $\mathbf{g}_1, \mathbf{g}_2$.

A similar construction can be repeated for any Lie bracket $[\mathbf{g}_i, \mathbf{g}_j]$; hence, the distribution Δ is involutive under the integrability condition (11.9).

Solution to Problem 11.4

Consider a set of k Pfaffian constraints

$$\mathbf{A}^T \dot{\mathbf{q}} = \mathbf{0}$$

with a constant constraint matrix \mathbf{A}^T . Then, the associated kinematic model

$$\dot{\mathbf{q}} = \sum_{j=1}^m \mathbf{g}_j u_j = \mathbf{G} \mathbf{u} \quad m = n - k \quad (\text{S11.1})$$

has constant input vector fields $\mathbf{g}_1, \dots, \mathbf{g}_m$. Its accessibility distribution $\Delta_{\mathcal{A}}$ is clearly involutive, because $[\mathbf{g}_i, \mathbf{g}_j] = 0$ for any i, j . The controllability condition (11.11) is hence violated, and in particular

$$\dim \Delta_{\mathcal{A}}(\mathbf{q}) = m.$$

Therefore, the above set of Pfaffian constraints is completely integrable, i.e., holonomic. This obvious conclusion could have also been reached by noting that (S11.1) is in fact a driftless linear system, and using directly the controllability rank condition (D.15).

Solution to Problem 11.5

Integration of the kinematic model (11.13) under the given input sequence starting from the initial configuration $\mathbf{q}_0 = [x_0 \ y_0 \ \theta_0]^T$ provides

$$\begin{aligned} x(4\varepsilon) &= x_0 + \varepsilon \cos \theta_0 - \varepsilon \cos(\theta_0 + \varepsilon) \\ y(4\varepsilon) &= y_0 + \varepsilon \sin \theta_0 - \varepsilon \sin(\theta_0 + \varepsilon) \\ \theta(4\varepsilon) &= \theta_0. \end{aligned}$$

The displacement is therefore

$$\mathbf{q}(4\varepsilon) - \mathbf{q}_0 = \begin{bmatrix} \varepsilon \cos \theta_0 - \varepsilon \cos(\theta_0 + \varepsilon) \\ \varepsilon \sin \theta_0 - \varepsilon \sin(\theta_0 + \varepsilon) \\ 0 \end{bmatrix} = \varepsilon^2 \mathbf{t},$$

with

$$\mathbf{t} = \begin{bmatrix} \frac{\cos \theta_0 - \cos(\theta_0 + \varepsilon)}{\varepsilon} \\ \frac{\sin \theta_0 - \sin(\theta_0 + \varepsilon)}{\varepsilon} \\ 0 \end{bmatrix}$$

— compare with the expression of the displacement for a generic two-input system as given in Appendix D.

Finally, one obtains

$$\lim_{\varepsilon \rightarrow 0} \mathbf{t} = \begin{bmatrix} -\left. \frac{d \cos \theta}{d \theta} \right|_{\theta=\theta_0} \\ -\left. \frac{d \sin \theta}{d \theta} \right|_{\theta=\theta_0} \\ 0 \end{bmatrix} = \begin{bmatrix} \sin \theta_0 \\ -\cos \theta_0 \\ 0 \end{bmatrix},$$

that coincides with the value at \mathbf{q}_0 of the Lie Bracket of the two input vector fields (see Sect. 11.2.1).

Solution to Problem 11.6

With reference to Figs. 1.13 and 11.3, denote by $\mathbf{p}_R = [x_R \ y_R]^T$, $\mathbf{p}_L = [x_L \ y_L]^T$ and $\mathbf{p}_M = [x_M \ y_M]^T$ respectively the position vector of the right wheel centre, of the left wheel centre and of the midpoint between the two. Since

$$\begin{aligned}\dot{x}_i &= r \omega_i \cos \theta \\ \dot{y}_i &= r \omega_i \sin \theta,\end{aligned}$$

where $i = R, L$, one obtains

$$\begin{aligned}\dot{x}_M &= \frac{\dot{x}_R + \dot{x}_L}{2} = r \cos \theta \frac{\omega_R + \omega_L}{2} \\ \dot{y}_M &= \frac{\dot{y}_R + \dot{y}_L}{2} = r \sin \theta \frac{\omega_R + \omega_L}{2}\end{aligned}$$

and therefore

$$v = \sqrt{\dot{x}_M^2 + \dot{y}_M^2} = \frac{r(\omega_R + \omega_L)}{2}.$$

The expression of the angular velocity of the robot can be obtained by applying the general velocity composition rule (B.4) and making use of the skew-symmetric operator $\mathbf{S}(\cdot)$:

$$\dot{\mathbf{p}}_R = \dot{\mathbf{p}}_L + \mathbf{S}(\omega)(\mathbf{p}_R - \mathbf{p}_L), \quad (\text{S11.2})$$

where

$$\mathbf{S}(\omega) = \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix}$$

because the motion of the robot is planar. Using the expressions of $\dot{\mathbf{p}}_R$, $\dot{\mathbf{p}}_L$ and the fact that

$$\mathbf{p}_R = \mathbf{p}_L + \begin{bmatrix} d \sin \theta \\ -d \cos \theta \end{bmatrix},$$

Equation (S11.2) becomes

$$\begin{bmatrix} r \omega_R \cos \theta \\ r \omega_R \sin \theta \end{bmatrix} = \begin{bmatrix} r \omega_L \cos \theta \\ r \omega_L \sin \theta \end{bmatrix} + \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix} \begin{bmatrix} d \sin \theta \\ -d \cos \theta \end{bmatrix}.$$

Squaring and adding side-by-side these two equations leads to

$$\omega^2 = \frac{r^2(\omega_R - \omega_L)^2}{d^2}.$$

Using again one of the two equations, it can be concluded that the only solution is

$$\omega = \frac{r(\omega_R - \omega_L)}{d}.$$

Solution to Problem 11.7

With reference to Fig. 11.4, denote by (x_C, y_C) the Cartesian coordinates of the instantaneous centre of rotation C in the world reference frame, and by (x'_C, y'_C) its coordinates in a moving frame having the origin at the centre of the rear wheel and the x' axis aligned with the bicycle body. Simple geometry gives

$$\begin{bmatrix} x'_C \\ y'_C \end{bmatrix} = \begin{bmatrix} 0 \\ \ell/\tan \phi \end{bmatrix}$$

and thus

$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \mathbf{R}(\theta) \begin{bmatrix} x'_c \\ y'_c \end{bmatrix} = \begin{bmatrix} x - \ell \sin \theta / \tan \phi \\ y + \ell \cos \theta / \tan \phi \end{bmatrix},$$

where

$$\mathbf{R}(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

is the (planar) rotation matrix of the moving frame with respect to the world frame.

Since the velocity vector of a generic point P on the robot body is tangent to the circle centred at C and passing through P , the modulus of the angular velocity of the bicycle is obtained as

$$\omega_{\text{body}} = v_P / R_P, \quad (\text{S11.3})$$

where v_P is the modulus of the velocity of P , and R_P is its instantaneous radius of rotation (i.e., its radius of curvature):

$$R_P = \sqrt{(x_P - x_C)^2 + (y_P - y_C)^2}.$$

The resulting ω_{body} is obviously the same for any choice of P . In particular, by choosing P as the centre of the rear wheel, one has $R_P = \ell / \tan \phi$, and thus

$$\omega_{\text{body}} = \dot{\theta} = v \tan \phi / \ell,$$

consistently with the evolution of θ predicted by the kinematic model (11.19). Finally, plugging this expression of ω in (S11.3) yields

$$v_P = R_P v \tan \phi / \ell.$$

Solution to Problem 11.8

With the configuration vector $\mathbf{q} = [x \ y \ \phi \ \theta_0 \ \theta_1 \ \dots \ \theta_N]^T$ defined in the solution to Problem 11.1, the $N + 2$ kinematic constraints are

$$\begin{aligned}\dot{x}_f \sin(\theta_0 + \phi) - \dot{y}_f \cos(\theta_0 + \phi) &= 0 \\ \dot{x} \sin \theta_0 - \dot{y} \cos \theta_0 &= 0 \\ \dot{x}_i \sin \theta_i - \dot{y}_i \cos \theta_i &= 0 \quad i = 1, \dots, N,\end{aligned}$$

where (x_f, y_f) and (x_i, y_i) are the Cartesian coordinates of the centre of the tricycle front wheel and of the i -th trailer wheel axle midpoint, respectively (compare with (11.15), (11.16)). Denote by ℓ the distance between the front wheel and the rear axle of the tricycle, and by ℓ_i the hinge-to-hinge length of the i -th trailer. It is

$$\begin{aligned}x_f &= x + \ell \cos \theta_0 \\ y_f &= y + \ell \sin \theta_0\end{aligned}$$

and

$$\begin{aligned}x_i &= x - \sum_{j=1}^i \ell_j \cos \theta_j \\ y_i &= y - \sum_{j=1}^i \ell_j \sin \theta_j,\end{aligned}$$

so that the kinematic constraints become

$$\begin{aligned}\dot{x} \sin(\theta_0 + \phi) - \dot{y} \cos(\theta_0 + \phi) - \dot{\theta}_0 \ell \cos \phi &= 0 \\ \dot{x} \sin \theta_0 - \dot{y} \cos \theta_0 &= 0 \\ \dot{x} \sin \theta_i - \dot{y} \cos \theta_i + \sum_{j=1}^i \dot{\theta}_j \ell_j \cos(\theta_i - \theta_j) &= 0 \quad i = 1, \dots, N.\end{aligned}$$

The null space of the constraint matrix is spanned by the two columns of

$$G(\mathbf{q}) = \begin{bmatrix} \cos \theta_0 & 0 \\ \sin \theta_0 & 0 \\ 0 & 1 \\ \frac{1}{\ell} \tan \phi & 0 \\ -\frac{1}{\ell_1} \sin(\theta_1 - \theta_0) & 0 \\ -\frac{1}{\ell_2} \cos(\theta_1 - \theta_0) \sin(\theta_2 - \theta_1) & 0 \\ \vdots & \vdots \\ -\frac{1}{\ell_i} \left(\prod_{j=1}^{i-1} \cos(\theta_j - \theta_{j-1}) \right) \sin(\theta_i - \theta_{i-1}) & 0 \\ \vdots & \vdots \\ -\frac{1}{\ell_N} \left(\prod_{j=1}^{N-1} \cos(\theta_j - \theta_{j-1}) \right) \sin(\theta_N - \theta_{N-1}) & 0 \end{bmatrix} = [\mathbf{g}_1(\mathbf{q}) \ \mathbf{g}_2(\mathbf{q})]$$

(compare with (11.19)). The kinematic control system is then

$$\dot{\mathbf{q}} = \mathbf{g}_1(\mathbf{q})v + \mathbf{g}_2(\mathbf{q})\omega,$$

where v and ω are respectively the driving velocity of the rear wheels and the steering velocity of the tricycle.

Solution to Problem 11.9

According to (11.14), the velocity inputs of the equivalent unicycle model are given by

$$v = \frac{r(\omega_R + \omega_L)}{2}$$

$$\omega = \frac{r(\omega_R - \omega_L)}{d},$$

where r is the wheel radius and d is the distance between the wheel centres. Solving for ω_R in the first and replacing the obtained expression in the second yields

$$\omega_L = \frac{2v - d\omega}{2r}$$

$$\omega_R = \frac{2v + d\omega}{2r}.$$

The resulting constraints on v , ω are then

$$\left| \frac{2v(t) - d\omega(t)}{2r} \right| \leq \omega_{RL}$$

$$\left| \frac{2v(t) + d\omega(t)}{2r} \right| \leq \omega_{RL},$$

that correspond to a rhombus-shaped admissible region in the v , ω plane. In particular, the maximum achievable values for these velocities are easily computed as the intercepts of the sides of the rhombus with the v , ω axes, i.e., $\pm r\omega_{RL}$ for the driving velocity and $\pm 2r\omega_{RL}/d$ for the steering velocity.

Solution to Problem 11.10

The geometric version of the (2,4) chained form is

$$z'_1 = \tilde{v}_1$$

$$z'_2 = \tilde{v}_2$$

$$z'_3 = z_2\tilde{v}_1$$

$$z'_4 = z_3\tilde{v}_1.$$

Choose the geometric inputs \tilde{v}_1, \tilde{v}_2 in the parameterized form (11.49), with $\Delta = z_{1,f} - z_{1,i}$ and $s \in [s_i, s_f] = [0, |\Delta|]$. Integrating in a recursive fashion the second, third and fourth equations and setting $z_2(s_f) = z_{2,f}$, $z_3(s_f) = z_{3,f}$ and $z_4(s_f) = z_{4,f}$ leads to a linear system in the form (11.50) where

$$\mathbf{D} = \begin{bmatrix} |\Delta| & \frac{\Delta^2}{2} & \frac{|\Delta|^3}{3} \\ \operatorname{sgn}(\Delta) \frac{\Delta^2}{2} & \frac{\Delta^3}{6} & \operatorname{sgn}(\Delta) \frac{\Delta^4}{12} \\ \frac{|\Delta|^3}{6} & \frac{\Delta^4}{24} & \frac{|\Delta|^5}{60} \end{bmatrix} \quad \mathbf{d} = \begin{bmatrix} z_{2,f} - z_{2,i} \\ z_{3,f} - z_{3,i} - z_{2,i} \Delta \\ z_{4,f} - z_{4,i} - z_{3,i} \Delta - z_{2,i} |\Delta| \end{bmatrix}.$$

Solution to Problem 11.11

It is sufficient to modify (11.48) as follows:

$$\tilde{v}_1 = \Delta = z_{1,f} - z_{1,i},$$

with the second input (11.49) unchanged. With this choice, z_1 will reach its final desired value exactly at $s_f = 1$. The coefficients c_0, \dots, c_{n-2} in (11.49) are computed by solving a linear system of equations that is still in the general form (11.50). For example, in the case of the (2, 3) chained form, one obtains

$$\mathbf{D} = \begin{bmatrix} 1 & 1/2 \\ \Delta/2 & \Delta/6 \end{bmatrix} \quad \mathbf{d} = \begin{bmatrix} z_{2,f} - z_{2,i} \\ z_{3,f} - z_{3,i} - z_{2,i} \Delta \end{bmatrix}.$$

Solution to Problem 11.12

As shown in the solution to Problem 11.11, the path planning scheme based on parameterized inputs (scheme A in the following) must be modified by letting

$$\tilde{v}_1 = \Delta = z_{1,f} - z_{1,i}$$

to obtain $s_f = 1$ and allow a comparison with the scheme that uses interpolating polynomials of different degree (scheme B in the following). With scheme A one has then

$$z_1(s) = z_{1,i} + (z_{1,f} - z_{1,i})s$$

that coincides with the expression of $z_1(s)$ generated by scheme B. Moreover, being

$$\tilde{v}_2 = c_0 + c_1 s$$

for a (2,3) chained form, it is

$$z_2(s) = z_{2,i} + c_0 s + c_1 s^2/2$$

and

$$z_3(s) = z_{3,i} + \Delta \int_0^s (c_0 \sigma + c_1 \sigma^2/2) d\sigma = z_{3,i} + \Delta (c_0 s^2/2 + c_1 s^3/6).$$

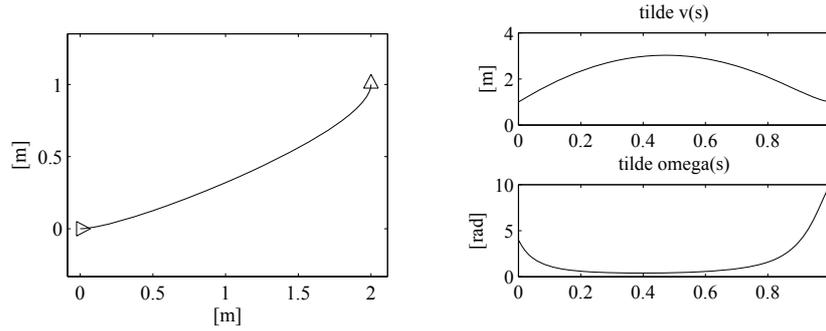


Fig. S11.1. Path planning via Cartesian polynomials; *left*: Cartesian path of the unicycle for the required parking manoeuvre, *right*: geometric inputs along the path

Hence, $z_3(s)$ with scheme A is a cubic polynomial in s , as is with scheme B. However, there is a unique cubic polynomial that meets the two boundary conditions on z_3 and the two boundary conditions on z_2

$$\frac{z_3'(0)}{z_1'(0)} = z_{2i} \quad \frac{z_3'(1)}{z_1'(1)} = z_{2f}.$$

Hence, the expressions of $z_3(s)$ generated by the two schemes coincide. Since z_1 and z_3 are flat outputs for a (2,3) chained form (see the end of Sect. 11.5.2), a unique $z_2(s)$ is associated to $z_1(s)$ and $z_3(s)$. One can then conclude that the paths in the \mathbf{z} space (and the associated geometric inputs) generated by the two schemes are exactly the same.

Solution to Problem 11.13

The files with the solution can be found in Folder 11_13. For the required manoeuvre, the coefficients of the interpolating polynomials

$$\begin{aligned} x(s) &= s^3 x_f - (s-1)^3 x_i + \alpha_x s^2 (s-1) + \beta_x s (s-1)^2 \\ y(s) &= s^3 y_f - (s-1)^3 y_i + \alpha_y s^2 (s-1) + \beta_y s (s-1)^2 \end{aligned}$$

are found to be

$$\begin{aligned} \alpha_x &= -6 & \beta_x &= 1 \\ \alpha_y &= -2 & \beta_y &= 0 \end{aligned}$$

where $k_i = k_f = 1$. The corresponding unicycle path is shown in Fig. S11.1.

Using the flatness property of the unicycle, expressed by (11.45), (11.46) and (11.47), File `s11_13.m` also generates the geometric inputs $\tilde{v}(s)$, $\tilde{\omega}(s)$ along the path, as shown in Fig. S11.1.

There are several ways to find a feasible timing law $s = s(t)$ over the planned path. For example, one may simply let $s = t$, so that $v(t)$ and $\omega(t)$

coincide with $\tilde{v}(s)$ and $\tilde{\omega}(s)$, respectively. However, both the inputs violate the given bounds; in particular, the maximum value is attained by ω at 10 rad/sec. A uniform scaling procedure with $T = 10$ can then be used to slow down the trajectory and recover the feasibility of the inputs (see (11.54), (11.55)).

Solution to Problem 11.14

The (2,3) chained form is

$$\begin{aligned}\dot{z}_1 &= v_1 \\ \dot{z}_2 &= v_2 \\ \dot{z}_3 &= z_2 v_1.\end{aligned}$$

Assume that a desired state trajectory $(z_{1d}(t), z_{2d}(t), z_{3d}(t))$ is given. For the desired trajectory to be feasible, it must satisfy the equations

$$\begin{aligned}\dot{z}_{1d} &= v_{1d} \\ \dot{z}_{2d} &= v_{2d} \\ \dot{z}_{3d} &= z_{2d} v_{1d}\end{aligned}$$

for some choice of the reference inputs $(v_{1d}(t), v_{2d}(t))$. Defining the state errors $e_i = z_{id} - z_i$, for $i = 1, 2, 3$, and the input errors $u_i = v_{id} - v_i$, for $i = 1, 2$, one has the following error dynamics

$$\begin{aligned}\dot{e}_1 &= u_1 \\ \dot{e}_2 &= u_2 \\ \dot{e}_3 &= z_{2d} v_{1d} - z_2 v_1 = v_{1d} e_2 + z_{2d} u_1 - e_2 u_1.\end{aligned}$$

Linearization around the reference trajectory yields the time-varying system

$$\dot{\mathbf{e}} = \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & v_{1d} & 0 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ z_{2d} & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \mathbf{A}(t)\mathbf{e} + \mathbf{B}(t)\mathbf{u}.$$

With the linear time-varying feedback

$$\begin{aligned}u_1 &= -k_1 e_1 \\ u_2 &= -k_2 e_2 - \frac{k_3}{v_{1d}} e_3\end{aligned}$$

the closed-loop linearized error dynamics becomes

$$\dot{\mathbf{e}} = \begin{bmatrix} -k_1 & 0 & 0 \\ 0 & -k_2 - \frac{k_3}{v_{1d}} \\ -k_1 z_{2d} & v_{1d} & 0 \end{bmatrix} \mathbf{e}.$$

The characteristic polynomial of this system is constant:

$$p(\lambda) = (\lambda + k_1)(\lambda^2 + k_2\lambda + k_3).$$

The eigenvalues are thus constant, and have negative real part provided that k_1 , k_2 and k_3 are positive. As discussed in Sect. 11.6.1, this does not guarantee stability in view of the time-varying nature of the linearized error dynamics, except for the case when v_{1d} and z_{2d} are constant¹. In this case, the linearized error system becomes time-invariant, and therefore it is asymptotically stable with the above choice of gains. Hence, the original error system is also asymptotically stable at the origin, although this result is not guaranteed to hold globally.

Solution to Problem 11.15

With reference to the kinematic model (11.19) and Fig. 11.4, consider the following outputs:

$$\begin{aligned} y_1 &= x + \ell \cos \theta + b \cos(\theta + \phi) \\ y_2 &= y + \ell \sin \theta + b \sin(\theta + \phi) \end{aligned}$$

with $b \neq 0$. They represent the Cartesian coordinates of a point P on the line passing through the centre of the front wheel and oriented as the wheel itself, located at a distance $|b|$ from the contact point of the wheel with the ground. The time derivatives of y_1 and y_2 are

$$\begin{aligned} \begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} &= \begin{bmatrix} \cos \theta - \tan \phi(\sin \theta + b \sin(\theta + \phi)/\ell) & -b \sin(\theta + \phi) \\ \sin \theta + \tan \phi(\cos \theta + b \cos(\theta + \phi)/\ell) & b \cos(\theta + \phi) \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \\ &= \mathbf{T}(\theta, \phi) \begin{bmatrix} v \\ \omega \end{bmatrix}. \end{aligned}$$

Matrix $\mathbf{T}(\theta, \phi)$ has determinant $b/\cos \phi$, and is therefore always invertible under the assumption that $b \neq 0$ and $|\phi(t)| \leq \pi/2$. It is then sufficient to use the following input transformation

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \mathbf{T}^{-1}(\theta, \phi) \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

to put the equations of the bicycle in the input-output linearized form:

$$\begin{aligned} \dot{y}_1 &= u_1 \\ \dot{y}_2 &= u_2 \\ \dot{\theta} &= \sin \phi(\cos(\theta + \phi)u_1 + \sin(\theta + \phi)u_2)/\ell \\ \dot{\phi} &= -(\cos(\theta + \phi)\sin \phi/\ell + \sin(\theta + \phi)/b)u_1 \\ &\quad -(\sin(\theta + \phi)\sin \phi/\ell - \cos(\theta + \phi)/b)u_2. \end{aligned}$$

¹ One may verify that, if the (2,3) chained form represents a unicycle under the transformation (11.23), (11.24), constant values of v_{1d} and z_{2d} correspond to reference trajectories over which ω_d is constant and v_d increases linearly over time (as in an Archimedean spiral).

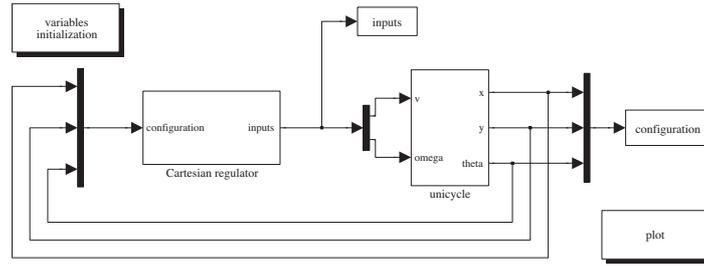


Fig. S11.2. SIMULINK block diagram of the unicycle with the modified cartesian regulator

At this point, a simple linear controller such as

$$\begin{aligned} u_1 &= \dot{y}_{1d} + k_1(y_{1d} - y_1) \\ u_2 &= \dot{y}_{2d} + k_2(y_{2d} - y_2), \end{aligned}$$

with $k_1 > 0$, $k_2 > 0$, guarantees exponential convergence to zero of the Cartesian tracking error, with decoupled dynamics on its two components. Note that the orientation and the steering angle, whose evolutions are governed by the third and fourth equation, are not controlled.

Solution to Problem 11.16

First of all, note that the Cartesian regulator (11.78), (11.79) may be equivalently written as

$$v = k_1 \mathbf{n}^T \mathbf{e}_p \quad (\text{S11.4})$$

$$\omega = k_2 \gamma, \quad (\text{S11.5})$$

where $\mathbf{n} = [\cos \theta \quad \sin \theta]^T$ is the unit vector aligned with the sagittal axis, $\mathbf{e}_p = [-x \quad -y]^T$ is the Cartesian error and $\gamma = \text{Atan2}(y, x) - \theta + \pi$ is the pointing error, i.e., the angle between \mathbf{n} and \mathbf{e}_p (see also Fig. 11.18). Expression (S11.5) shows that ω induces a rotation until \mathbf{n} and \mathbf{e}_p align, i.e., until the unicycle points to the origin; as a consequence, the latter will always be reached in forward motion.

Now, redefine the steering velocity (S11.5) as follows:

$$\omega = \begin{cases} k_2 \gamma & \text{if } \mathbf{n}^T \mathbf{e}_p > 0 \\ k_2(\gamma - \text{sign}(\gamma)\pi) & \text{if } \mathbf{n}^T \mathbf{e}_p \leq 0. \end{cases} \quad (\text{S11.6})$$

With this modification, the unicycle is forced to align with \mathbf{e}_p if the pointing error is acute, and with $-\mathbf{e}_p$ otherwise. In the second case, the origin will be reached in backward motion.

Unicycle control with the modified position regulator (S11.4), (S11.6) is implemented by the SIMULINK block diagram shown in Fig. S11.2. The files

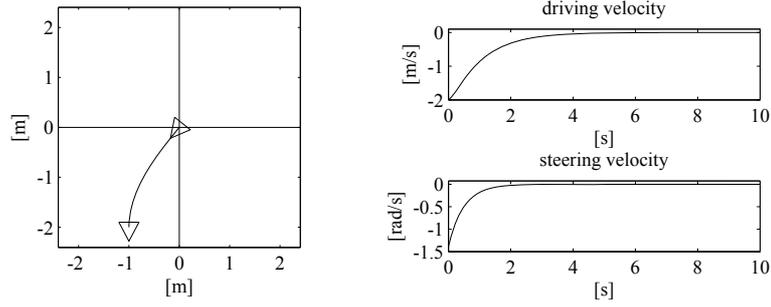


Fig. S11.3. Regulation to the origin of the Cartesian position of the unicycle with the modified controller (S11.4), (S11.6); *left*: Cartesian motion of the unicycle, *right*: time evolution of the velocity inputs v and ω

with the solution can be found in Folder 11_16. The unicycle is simulated as a continuous-time system using a variable-step integration method, with a maximum step size of 0.01 s. The controller gains can be chosen in the initialization file.

When the initial pointing error is an acute angle, the modified and the original position regulators produce exactly the same trajectories. For example, this is the case of the simulation in Fig. 11.19 (left). However, in the case of Fig. 11.19 (right), the initial pointing error is obtuse, and thus the modified position regulator leads the unicycle to the origin in backward motion, as shown in Fig. S11.3.

Solution to Problem 11.17

Assume that the velocity inputs are constant within each sampling interval:

$$v(t) = v_k \quad \omega(t) = \omega_k \quad t \in [t_k, t_{k+1}].$$

Integration of the kinematic equations (11.13) of the unicycle readily provides

$$\theta(t) = \theta_k + \omega_k (t - t_k)$$

and

$$\begin{aligned} x(t) &= x_k + v_k \int_0^{t-t_k} \cos(\theta_k + \omega_k \tau) d\tau = x_k + \frac{v_k}{\omega_k} (\sin(\theta_k + \omega_k(t - t_k)) - \sin \theta_k) \\ &= x_k + \frac{v_k}{\omega_k} (\sin \theta(t) - \sin \theta_k) \\ y(t) &= y_k + v_k \int_0^{t-t_k} \sin(\theta_k + \omega_k \tau) d\tau = y_k - \frac{v_k}{\omega_k} (\cos(\theta_k + \omega_k(t - t_k)) - \cos \theta_k) \\ &= y_k - \frac{v_k}{\omega_k} (\cos \theta(t) - \cos \theta_k). \end{aligned}$$

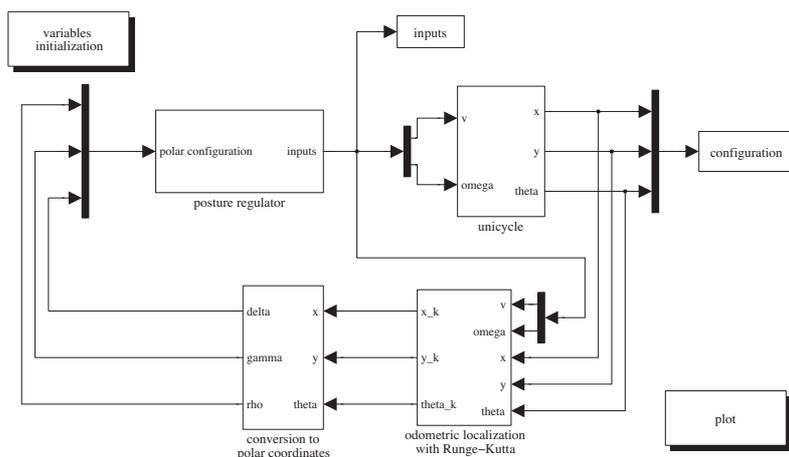


Fig. S11.4. Simulink block diagram of the unicycle with the posture regulator and Runge–Kutta odometric localization

Evaluating the above expressions at $t = t_{k+1} = t_k + T_s$ leads to the odometric prediction (11.85). Alternative proofs can be devised using geometric arguments (the unicycle travels along an arc of circle during each sampling interval) or resorting to a chained-form transformation.

Solution to Problem 11.18

Control of the unicycle with the posture regulator (11.81), (11.82) and Runge–Kutta odometric localization (11.84) is implemented by the SIMULINK block diagram shown in Fig. S11.4. The files with the solution can be found in Folder 11_18. The unicycle is simulated as a continuous-time system using a variable-step integration method, with a maximum step size of 0.01 s. The controller gains and the duration T_s of the sampling interval for odometric localization can be chosen in the initialization file.

The results obtained with $T_s = 0.01$ s and $T_s = 0.1$ s are shown in Fig. S11.5 and Fig. S11.6, respectively. The initial configuration of the unicycle is the same of Fig. 11.20 (right). Notice how for $T_s = 0.1$ s the low accuracy of the odometric localization does not hinder the convergence of the unicycle to the destination. However, a further increase of T_s will ultimately destabilize the controlled unicycle.

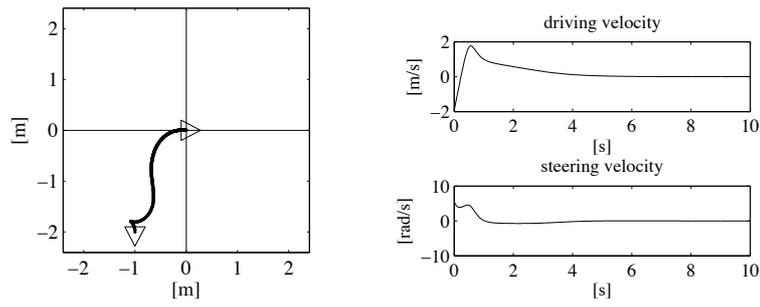


Fig. S11.5. Regulation to the origin of the posture of the unicycle with the controller (11.81), (11.82) and odometric localization (11.84) with $T_s = 0.01$ s; *left*: Cartesian motion of the unicycle (continuous) and odometric estimate (dots), *right*: time evolution of the velocity inputs v and ω

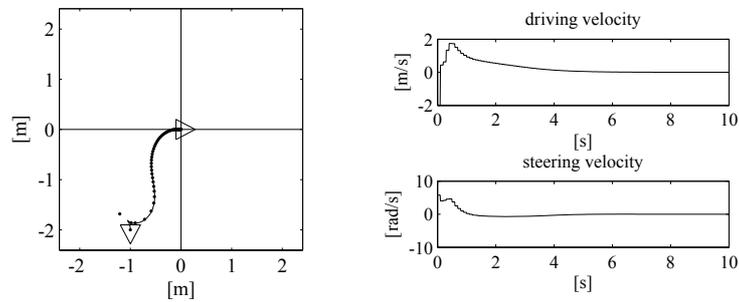


Fig. S11.6. Regulation to the origin of the posture of the unicycle with the controller (11.81), (11.82) and odometric localization (11.84) with $T_s = 0.1$ s; *left*: Cartesian motion of the unicycle (continuous) and odometric estimate (dots), *right*: time evolution of the velocity inputs v and ω

Motion Planning

Solution to Problem 12.1

The configuration of the mobile manipulator is $\mathbf{q} = [x \ y \ \theta_0 \ \theta_1 \ \dots \ \theta_6]^T$, where (x, y) are the Cartesian coordinates of the contact point of the wheel with the ground (equivalently, of the wheel centre), θ_0 is the orientation of the unicycle with respect to the x axis, and $\theta_1, \dots, \theta_6$ are the manipulator joint variables. The configuration space is therefore $\mathcal{C} = \mathbb{R}^2 \times SO(2) \times \dots \times SO(2)$ (with $SO(2)$ appearing 7 times), and has dimension 9.

Solution to Problem 12.2

Assume that the configuration \mathbf{q} takes values in the subset \mathcal{Q} of Fig. 12.1 — this can be obtained by computing the joint variables q_1 and q_2 modulo 2π). Given two configurations $\mathbf{q}_A = (q_{1,A}, q_{2,A})$, $\mathbf{q}_B = (q_{1,B}, q_{2,B})$ in \mathcal{Q} , define

$$\begin{aligned}\Delta_1 &= \min(|q_{1,A} - q_{1,B}|, 2\pi - |q_{1,A} - q_{1,B}|) \\ \Delta_2 &= \min(|q_{2,A} - q_{2,B}|, 2\pi - |q_{2,A} - q_{2,B}|)\end{aligned}$$

and let

$$d_3(\mathbf{q}_A, \mathbf{q}_B) = \sqrt{\Delta_1^2 + \Delta_2^2}.$$

This definition of configuration space distance clearly satisfies the requirement of the problem.

It can be easily shown that $d_3(\mathbf{q}_A, \mathbf{q}_B)$ coincides with $d_2(\mathbf{q}_A, \mathbf{q}_B)$ given by eq. (12.2) whenever the Euclidean distance between \mathbf{q}_A and \mathbf{q}_B is not larger than $\sqrt{2}\pi$.

Solution to Problem 12.3

A possible solution is shown in Fig. S12.1. In the first scene, the robot is triangular and there is a single square obstacle, while in the second the robot

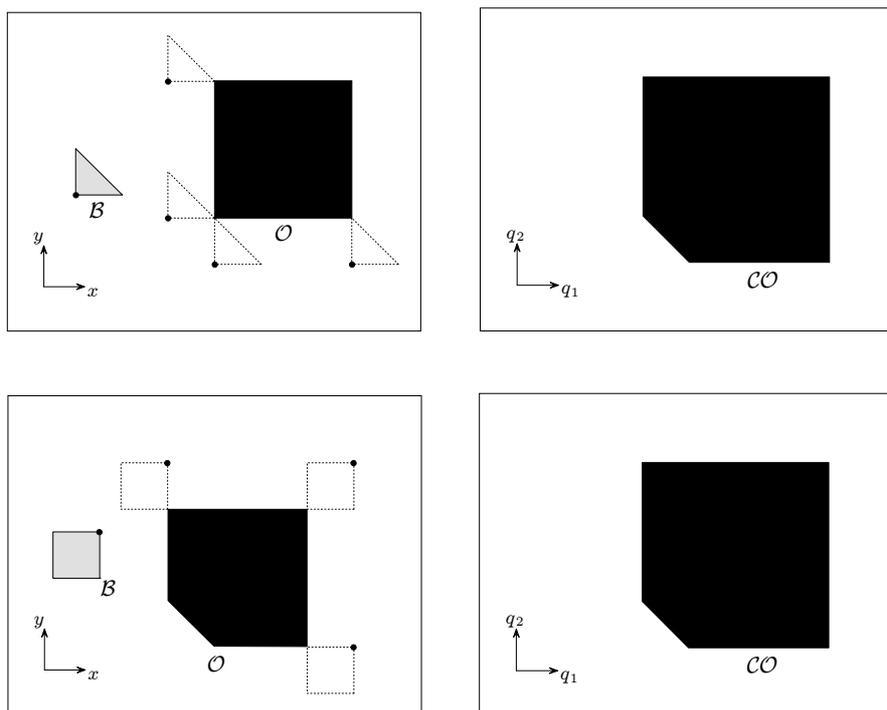


Fig. S12.1. Two different scenes (above/below) that result in the same \mathcal{C} -obstacle region. For each scene, *left*: the robot \mathcal{B} , the obstacle \mathcal{O} and the growing procedure for building \mathcal{C} -obstacles, *right*: the configuration space \mathcal{C} and the \mathcal{C} -obstacle \mathcal{CO}

is square and the obstacle is a pentagon. Under the assumption that the robots can freely translate without changing their orientation, the \mathcal{C} -obstacle region is exactly the same for the two scenes.

Solution to Problem 12.4

The problem can be solved by visual inspection of Fig. 12.4. For example, the following configurations of the 2R manipulator lie in the three disjoint components of $\mathcal{C}_{\text{free}}$:

$$\mathbf{q}_a = \begin{bmatrix} \pi/4 \\ \pi/2 \end{bmatrix} \quad \mathbf{q}_b = \begin{bmatrix} 4.12 \\ \pi/6 \end{bmatrix} \quad \mathbf{q}_c = \begin{bmatrix} 4.12 \\ 2.88 \end{bmatrix} \quad [\text{rad}].$$

The manipulator posture and the position in \mathcal{C} for each of these configurations is shown in Fig. S12.2.

Solution to Problem 12.5

Consider the lists $\mathbf{V} = \{V_1, \dots, V_v\}$ and $\mathbf{S} = \{S_1, \dots, S_s\}$ of all the vertices and sides of the given limited polygonal subset \mathcal{K} of \mathbb{R}^2 . A rough sketch of

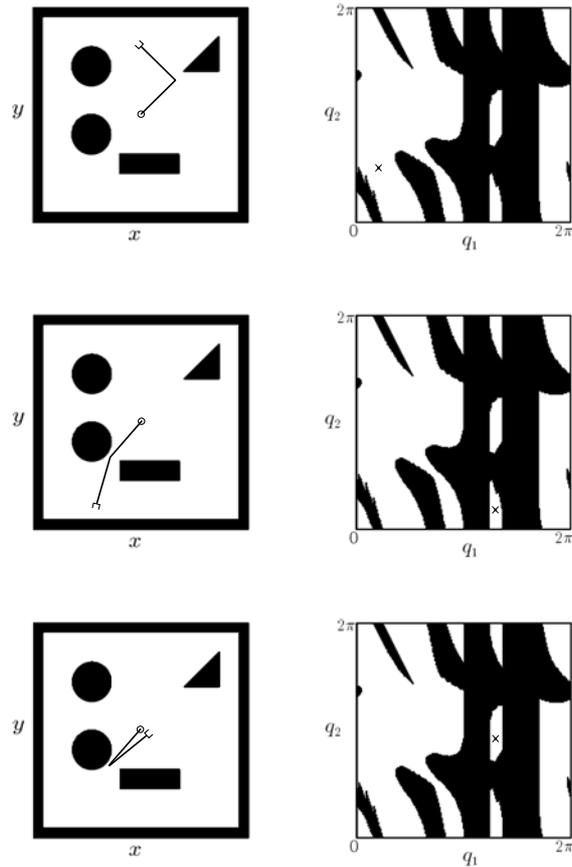


Fig. S12.2. Three configurations of the 2R manipulator that lie in disjoint components of C_{free} : q_a (top), q_b (centre) and q_c (bottom). For each of them, *left*: the corresponding manipulator posture, *right*: the position in \mathcal{C}

a naive algorithm for computing the generalized Voronoi diagram of \mathcal{K} is the following.

1. Build all the equidistance curves as follows:
 - 1a. For each vertex-vertex pair (V_i, V_j) , derive the equation of the line $L_{V_i V_j}$ passing through the midpoint of the segment $V_i V_j$ and orthogonal to the segment itself.
 - 1b. For each vertex-side pair (V_i, S_j) , denote by L_{S_j} the line containing S_j , and derive the equation of the parabola $P_{V_i S_j}$ having V_i as focus and L_{S_j} as directrix.
 - 1c. For each side-side pair (S_i, S_j) , denote by L_{S_i} and L_{S_j} the lines containing S_i and S_j , respectively, and derive the equation of the bisectrix $L_{S_i S_j}$ of the angle formed by L_{S_i} and L_{S_j} which contains S_i and S_j .

2. For each possible pair of equidistance curves, compute the intersection points — at most one in the case of a line/line pair, at most two for a line/parabola or parabola/parabola pair. Each of these points is characterized by the two distances from the features (vertex/vertex, line/line or line/vertex) that generate the two intersecting equidistance curves.
3. Discard all intersection points that do not belong to \mathcal{K} .
4. Discard all intersection points for which the two characteristic distances are not equal.
5. For each remaining intersection point, compute its clearance, and discard all intersection points for which the clearance is not equal to the characteristic distance.
6. All the remaining intersection points are nodes of the generalized Voronoi diagram. Its arcs are the portions of the equidistance curves that are enclosed by these nodes.

Solution to Problem 12.6

Refer to the example shown in Fig. 12.7, and assume that the goal configuration \mathbf{q}_g is moved somewhere in cell c_6 , so that $c_s = c_3$ and $c_g = c_6$. The shortest channel joining c_s to c_g is clearly $\{c_3, c_2, c_6\}$. The path extraction procedure described in Sect. 12.4.1 would produce a broken line with the following 4 vertices: \mathbf{q}_s , the midpoint of the common boundary between c_3 and c_2 , the midpoint of the common boundary between c_2 and c_6 , and finally \mathbf{q}_g . Hence, the second of the 3 edges forming the path would lie completely on the boundary of cell c_2 , and in particular would go through the leftmost vertex of the obstacle.

To solve this problem, the path extraction procedure can be modified by including additional vertices in the path. For example, in addition to \mathbf{q}_s , \mathbf{q}_g and the midpoints of the common boundaries in the channel, one may include the centroids (or a generic internal point) of all the cells crossed by the channel, with the exception of c_s and c_g . In the above case, the resulting free path would be a broken line with 5 vertices: \mathbf{q}_s , the midpoint of the common boundary between c_3 and c_2 , the centroid of cell c_2 , the midpoint of the common boundary between c_2 and c_6 , and finally \mathbf{q}_g .

Solution to Problem 12.7

The files with the solution can be found in Folder 12_7; run `s12_7.m` to execute the program. The robot is a planar manipulator whose first two joints are prismatic (so that the base can move arbitrarily) followed by an arbitrary number of revolute joints. An admissible range can be specified for each joint variable; in particular, a fixed-base manipulator is specified by setting to zero the minimum and maximum values of the prismatic joint variables. The obstacles may be line segments, circles or polygons. The user must also input the start and goal configurations, the maximum number of configurations in

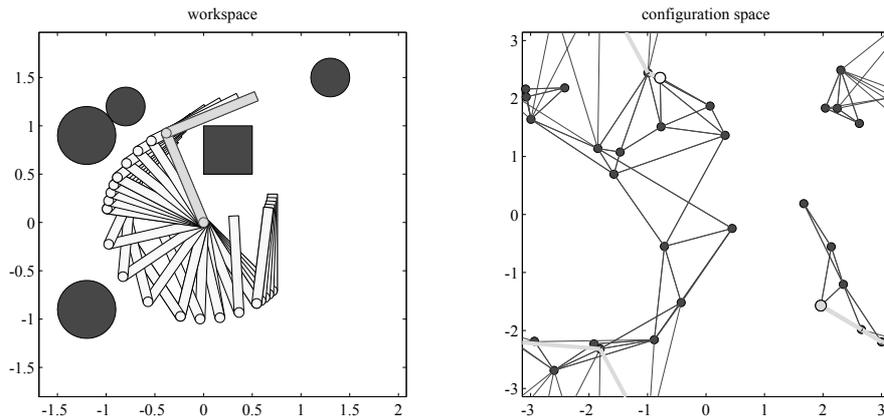


Fig. S12.3. Motion planning for a 2R planar manipulator via the PRM method; *left*: stroboscopic motion of the robot in the workspace, *right*: the PRM and the solution path (thick gray) in the configuration space

the roadmap, the maximum number of neighbours to which a new sample can be connected, and the maximum distance between neighbours.

The solution of a specific planning problem for a 2R fixed-base manipulator is reported in Fig. S12.3. The start and goal configurations are $\mathbf{q}_s = [-\pi/4 \quad 3\pi/4]^T$ and $\mathbf{q}_g = [5\pi/8 \quad -\pi/2]^T$ [rad,rad], respectively. Note that the configuration space, shown as a square in the figure, is correctly treated as a two-dimensional torus in the program. In particular, the configuration space distance is a generalization of the definition proposed in the solution to Problem 12.2.

Solution to Problem 12.8

The files with the solution can be found in Folder 12_8; run `s12_8.m` to execute the program. The obstacles may be line segments, circles or polygons. The user must also input the start and goal configurations, the motion primitives (the discrete set of admissible constant values of the driving and steering velocities within the time interval Δ), the value of Δ and the maximum number of configurations in the roadmap.

Since the unicycle cannot reach arbitrary configurations in $\mathcal{C}_{\text{free}}$ with this planner, the program terminates whenever a configuration is reached that is sufficiently close to the desired goal; to this end, a user-selectable distance threshold is used. Moreover, to bias the search towards the goal, \mathbf{q}_{rand} may be set to \mathbf{q}_{goal} — rather than to a random configuration — with a probability that can be chosen arbitrarily.

The solution of a specific problem is reported in Fig. S12.4. The start and goal configurations are $\mathbf{q}_s = [-3 \quad 0 \quad 0]^T$ and $\mathbf{q}_g = [3 \quad -3.5 \quad 0]^T$ [m,m,rad], respectively. Note that the unicycle does not exactly reach \mathbf{q}_g , shown in gray

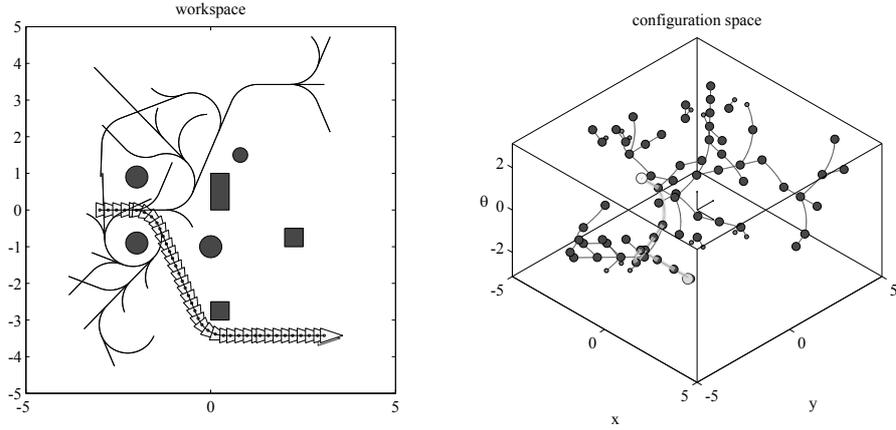


Fig. S12.4. Motion planning for a unicycle via the RRT method; *left*: stroboscopic motion of the robot and projection of the RRT on the workspace, *right*: the RRT and the solution path (thick gray) in the configuration space

in the stroboscopic plot. The configuration space, shown as a parallelepiped in the figure, is correctly treated as $\mathbb{R}^2 \times SO(2)$ in the program. In particular, the configuration space distance is an adaptation of the definition proposed in the solution to Problem 12.2.

Solution to Problem 12.9

Let

$$U_a(\mathbf{q}) = \begin{cases} \frac{1}{2} k_a \|\mathbf{e}(\mathbf{q})\|^2 & \text{if } \|\mathbf{e}(\mathbf{q})\| \leq \rho \\ k_b \|\mathbf{e}(\mathbf{q})\| & \text{if } \|\mathbf{e}(\mathbf{q})\| > \rho. \end{cases}$$

Continuity of the attractive force at the transition radius ρ is guaranteed by imposing

$$k_a \mathbf{e}(\mathbf{q}) = k_b \frac{\mathbf{e}(\mathbf{q})}{\|\mathbf{e}(\mathbf{q})\|} \quad \text{for } \|\mathbf{e}(\mathbf{q})\| = \rho,$$

i.e., $k_b = \rho k_a$.

Solution to Problem 12.10

Consider Fig. S12.5, and in particular the half-line that originates at \mathbf{q}_g and passes through the centre of the obstacle. Visualize a point that travels along this half-line, coming from the point at infinity, where the total potential is infinity — because of the attractive potential. The total potential goes back to infinity when the point reaches the boundary of the obstacle — because of the repulsive potential. Therefore, there is a point on the line where U has a local minimum; call this point P .

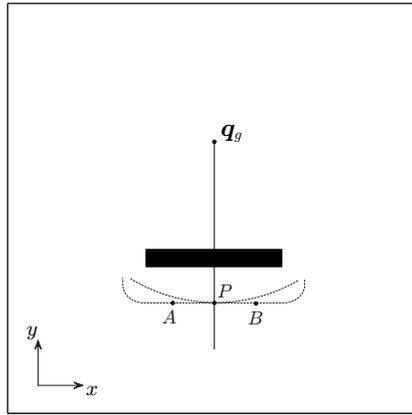


Fig. S12.5. The emergence of local minima due to the superposition of attractive and repulsive potentials

Now consider points A and B that are located along the same equipotential contour of the repulsive potential U_r as P . Since the attractive potential is higher at A and B than at C , P is a local minimum also in the direction of the line AB . In formulæ:

$$\left. \frac{\partial U}{\partial x} \right|_P = 0 \quad \left. \frac{\partial U}{\partial y} \right|_P = 0,$$

which imply that P is a stationary point for U , i.e., the gradient of U is zero at P . To show that P is indeed a local minimum, one should consider the Hessian matrix of U

$$\begin{bmatrix} \frac{\partial^2 U}{\partial x^2} & \frac{\partial^2 U}{\partial x \partial y} \\ \frac{\partial^2 U}{\partial y \partial x} & \frac{\partial^2 U}{\partial y^2} \end{bmatrix}$$

and prove that it is positive definite in P . This can be verified by using the analytic expression of U . Note that the elements of the diagonal are certainly positive in P in view of the above arguments.

Solution to Problem 12.11

Denote by (x, y) the Cartesian coordinates of the robot, by (x_i, y_i) the Cartesian coordinates of the centre C_i of the i -th circular obstacle, and by ρ_i its radius. The attractive potential is

$$U_a(x, y) = \frac{1}{2} k_a (x^2 + y^2),$$

while the repulsive potentials ($\gamma = 2$) are

$$U_{r,i}(x, y) = \begin{cases} \frac{k_{r,i}}{2} \left(\frac{1}{\eta_i(x, y)} - \frac{1}{\eta_{0,i}} \right)^2 & \text{if } \eta_i(x, y) \leq \eta_{0,i} \\ 0 & \text{if } \eta_i(x, y) > \eta_{0,i}, \end{cases}$$

for $i = 1, 2, 3$, with $\eta_i(x, y) = \sqrt{(x - x_i)^2 + (y - y_i)^2} - \rho_i$. The total potential is

$$U_t(x, y) = U_a(x, y) + \sum_{i=1}^3 U_{r,i}(x, y),$$

while the total force is

$$\mathbf{f}(x, y) = \mathbf{f}_a(x, y) + \sum_{i=1}^3 \mathbf{f}_{r,i}(x, y)$$

with

$$\mathbf{f}_a(x, y) = -k_a \begin{bmatrix} x \\ y \end{bmatrix}$$

and

$$\mathbf{f}_{r,i}(x, y) = \begin{cases} \frac{k_{r,i}}{\eta_i^2(x, y)} \left(\frac{1}{\eta_i(x, y)} - \frac{1}{\eta_{0,i}} \right) \nabla \eta_i(x, y) & \text{if } \eta_i(x, y) \leq \eta_{0,i} \\ 0 & \text{if } \eta_i(x, y) > \eta_{0,i}. \end{cases}$$

In view of the above expression for $\eta_i(x, y)$, one has

$$\nabla \eta_i(x, y) = \begin{bmatrix} \frac{x - x_i}{\sqrt{(x - x_i)^2 + (y - y_i)^2}} \\ \frac{y - y_i}{\sqrt{(x - x_i)^2 + (y - y_i)^2}} \end{bmatrix}.$$

The actual numerical expressions depend on the choice of the constants k_a and $k_{r,i}$, for $i = 1, 2, 3$, as well as of the obstacle range of influence $\eta_{0,i}$.

Assuming that the ranges of influence of the obstacles do not overlap, the saddle points of the total potential will be clearly located along the lines that join the origin of the reference frame (the goal) with the centres of the circular obstacles, in the ‘shadow zone’ of each obstacle. In particular, the saddle points are found by solving the equations

$$-k_a \begin{bmatrix} x \\ y \end{bmatrix} = \frac{k_{r,i}}{\eta_i^2(x, y)} \left(\frac{1}{\eta_i(x, y)} - \frac{1}{\eta_{0,i}} \right) \nabla \eta_i(x, y) \quad i = 1, 2, 3,$$

each representing the balance between the attractive force and the i -th repulsive force.

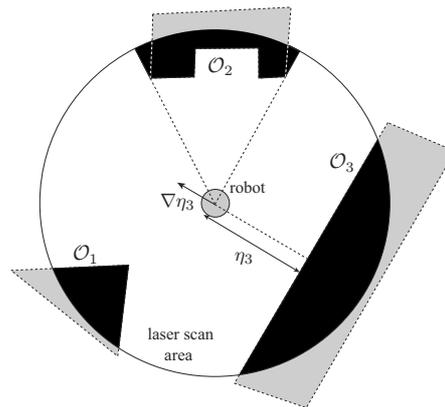


Fig. S12.6. On-line planning based on artificial potential fields for a circular mobile robot equipped with a rotating laser range finder.

Solution to Problem 12.12

In principle, there are two main approaches for on-line motion planning based on sensor data.

- The first approach is to store the data coming from the sensors in a map — i.e., a representation of the portion of the environment that the robot has visited so far — and use it to plan in an incremental fashion (deliberative planning).
- The second approach is to react only to the obstacles that are currently perceived of the robot, without any form of memory (reactive planning).

The artificial potential field technique naturally lends itself to a reactive implementation, as briefly discussed below.

Refer to the situation shown in Fig. S12.6. The workspace obstacles \mathcal{O}_1 , \mathcal{O}_2 and \mathcal{O}_3 are depicted in black for the portion falling inside the laser scan area, and in gray outside. Note how the perceived obstacle is larger than the actual obstacle for \mathcal{O}_2 , due to the fact that the laser range finder is a visibility-based sensor.

Consider first the case of a point robot. Since the goal is known, the attractive potential can be built as in the off-line case — this is true provided that the robot is equipped with a localization module. To build the repulsive potential, it is reasonable to define the range of influence $\eta_{0,i}$ of the obstacles to be smaller than the maximum measurable range R — if the robot does not detect the obstacle, it does not react to it. A typical choice is $\eta_{0,i} = R$, for all i . As seen in Sect. 12.6.2, the perceived obstacle region must be decomposed in convex components to guarantee continuity of the repulsive potential field; in the case of Fig. S12.6, only \mathcal{O}_2 needs to undergo this procedure. At this point, the repulsive force produced by each convex component is computed as

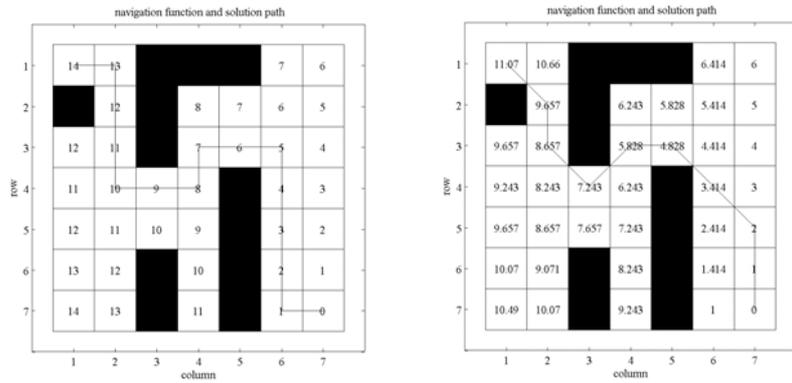


Fig. S12.7. The assigned occupancy grid, the navigation function and the solution path from cell (1, 1) to cell (7, 7); *left*: using 1-adjacency, *right*: using 2-adjacency

in (12.15), with $\eta_i, \nabla\eta_i$ directly obtained from the range scan profile — see the example in Fig. S12.6.

In the case of a robot with a finite nonzero radius ρ , one simply uses $\eta_i - \rho$ in place of η_i in (12.15), and obtains $\nabla\eta_i$ as in the point robot case.

It is easy to realize that this reactive version of the artificial potential method behaves exactly as the off-line version, in the sense that the resulting motion of the robot is the same. Clearly, this is true as long as the environment is continuously scanned during the motion and the above procedure for computing the potential field can be efficiently implemented, so as to be executed within the duration of the motion control cycle.

Solution to Problem 12.13

The files with the solution can be found in Folder 12.13. The user is required to input the binary gridmap representing the configuration space, the start and goal cells, and the adjacency type (1- or 2-adjacency). The cells are assumed to be squares with unit side. During the expansion of the wavefront that originates at \mathbf{q}_g , the increase of the navigation function between adjacent cells (cost-to-go) is set equal to the Euclidean distance between the cell centres. The solution path is then found by following the steepest descent from the goal cell.

For illustration, Fig. S12.7 shows the results obtained on a (7×7) gridmap, with (1, 1) as the start cell and (7, 7) as the goal cell. The path found using 2-adjacency is shorter as expected. It is easy to realize that even when the solution paths are in the same homotopy class — i.e., one can be continuously deformed into the other — the 1-adjacency path may be longer than the 2-adjacency path by a factor of as much as $\sqrt{2}$. On the other hand, 2-adjacency produces paths that may graze the \mathcal{C} -obstacles; as a consequence, it is recommended that these are slightly enlarged before planning.



<http://www.springer.com/978-1-84628-641-4>

Robotics

Modelling, Planning and Control

Siciliano, B.; Sciavicco, L.; Villani, L.; Oriolo, G.

2009, XXIV, 632 p. 298 illus. With online files/update.,

Hardcover

ISBN: 978-1-84628-641-4