SOLUTIONS MANUAL

INTRODUCTION TO ROBOTICS

MECHANICS AND CONTROL

THIRD EDITION

JOHN J. CRAIG

PEARSON Prentice Hall

Upper Saddle River, New Jersey 07458

Associate Editor: *Alice Dworkin*
Executive Managing Editor: *Vince O'Brien*
Managing Editor: *David A. George*
Production Editor: *Craig Little*
Supplement Cover Manager: *Daniel Sandin*
Manufacturing Buyer: *Ilene Kahn*

Pearson Prentice Hall
Pearson Education, Inc.
Upper Saddle River, NJ 07458

Printed in the United States of America
10 9 8 7 6 5 4 3 2 1

ISBN 0201-54362-1

Pearson Education Ltd., *London*
Pearson Education Australia Pty. Ltd., *Sydney*
Pearson Education Singapore, Pte. Ltd.
Pearson Education North Asia Ltd., *Hong Kong*
Pearson Education Canada, Inc., *Toronto*
Pearson Educación de Mexico, S.A. de C.V.
Pearson Education—Japan, *Tokyo*
Pearson Education Malaysia, Pte. Ltd.
Pearson Education, Inc., *Upper Saddle River, New Jersey*

Contents

Solutions Manual

Chapter 1
Introduction

Exercises

1.1) Here's just an example of a reasonable response:
(ref. [8] in Chap. 1)

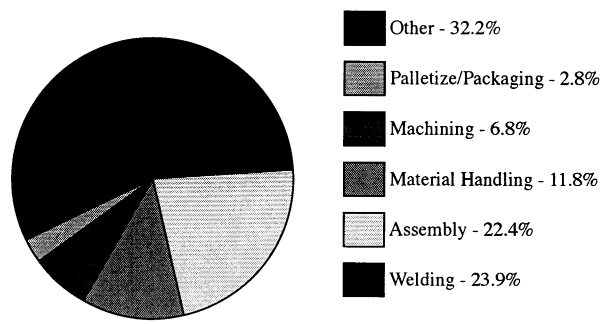
1955	Denavit & Hartenberg developed methodology for describing linkages.
1961	George Devol patents design of rst robot.
1961	First unimate robot installed.
1968	Shakey Robot developed at S.R.I.
1975	Robot institute of America formed.
1975	Unimation becomes rst Robot Co. to be protable.
1978	First Puma Robot shipped to GM.
1985	Total U.S. market exceeds 500 million dollars (annual revenue).

Developments might be split into a technical list and a business list.

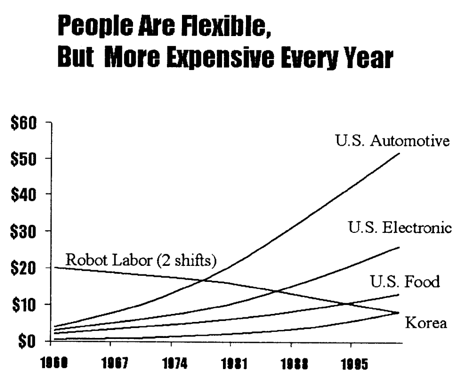
1.2) (Based on 1981 numbers)

Source:

L. Conigliaro, "robotics presentation, institutional investors conf.", May 28, 1981, Bache Newsletter 81–249.:



1.3)



Chapter 2

Spatial Descriptions and Transformations

Exercises

2.1) $R = \mathrm{rot}(\hat{x}, \phi)\,\mathrm{rot}(\hat{z}, \theta)$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\phi & -S\phi \\ 0 & S\phi & C\phi \end{bmatrix} \begin{bmatrix} C\theta & -S\theta & 0 \\ S\theta & C\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} C\theta & -S\theta & 0 \\ C\phi S\theta & C\phi C\theta & -S\phi \\ S\phi S\theta & S\phi C\theta & C\phi \end{bmatrix}$$

2.2) $R = \mathrm{rot}(\hat{x}, 45^\circ)\,\mathrm{rot}(\hat{y}, 30^\circ)$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & .707 & -.707 \\ 0 & .707 & .707 \end{bmatrix} \begin{bmatrix} .866 & 0 & .5 \\ 0 & 1 & 0 \\ -.5 & 0 & .866 \end{bmatrix}$$

$$= \begin{bmatrix} .866 & 0 & .5 \\ .353 & .707 & -.612 \\ -.353 & .707 & .612 \end{bmatrix}$$

2.3) Since rotations are performed about axes of the frame being rotated, these are Euler-Angle style rotations:

$R = \mathrm{rot}(\hat{z}, \theta)\,\mathrm{rot}(\hat{x}, \phi)$

We might also use the following reasoning:

$${}^{A}_{B}R(\theta, \phi) = {}^{B}_{A}R^{-1}(\theta, \phi)$$

$$= [\mathrm{rot}(\hat{x}, -\phi)\,\mathrm{rot}(\hat{z}, -\theta)]^{-1}$$

$$= \mathrm{rot}^{-1}(\hat{z}, -\theta)\,\mathrm{rot}^{-1}(\hat{x}, -\phi)$$

$$= \mathrm{rot}(\hat{z}, \theta)\,\mathrm{rot}(\hat{x}, \phi)$$

Yet another way of viewing the same operation:

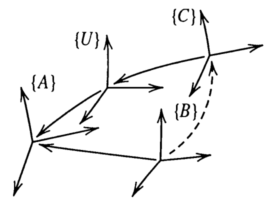
1st rotate by $\mathrm{rot}(\hat{z}, \theta)$

2nd rotate by $\mathrm{rot}(\hat{z}, \theta)\,\mathrm{rot}(\hat{x}, \phi)\,\mathrm{rot}^{-1}(z, \theta)$

2.13) By just following arrows, and reversing (by inversion) where needed, we have:

$${}^{B}_{C}T = {}^{B}_{A}T \; {}^{U}_{A}T^{-1} \; {}^{C}_{U}T^{-1}$$

Inverting a transform is done using eq. (2.40) in book. Rest is boring.



2.14) This rotation can be written as:

$${}^{A}_{B}T = \text{trans}({}^{A}\hat{P}, |{}^{A}P|)\,\text{rot}(\hat{K}, \theta)\text{trans}(-{}^{A}\hat{P}, |{}^{A}P|)$$

Where $\text{rot}(\hat{K}, \theta)$ is written as in eq. (2.77),

$$\text{And} \quad \text{trans}({}^{A}\hat{P}, |{}^{A}P|) = \begin{bmatrix} 1 & 0 & 0 & P_x \\ 0 & 1 & 0 & P_y \\ 0 & 0 & 1 & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{And} \quad \text{trans}(-{}^{A}\hat{P}, |{}^{A}P|) = \begin{bmatrix} 1 & 0 & 0 & -P_x \\ 0 & 1 & 0 & -P_y \\ 0 & 0 & 1 & -P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Multiplying out we get:

$${}^{A}_{B}T = \left[\begin{array}{ccc|c} R_{11} & R_{12} & R_{13} & Q_x \\ R_{21} & R_{22} & R_{23} & Q_y \\ R_{31} & R_{32} & R_{33} & Q_2 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

where the R_{ij} are given be eq. (2.77). And:

$$Q_x = P_x - P_x(K_x^2 V\theta + C\theta) - P_y(K_x K_y V\theta - K_z S\theta)$$
$$- P_z(K_x K_z V\theta + K_y S\theta)$$

$$Q_y = P_y - P_x(K_x K_y V\theta + K_z S\theta) - P_y(K_y^2 V\theta + C\theta)$$
$$- P_z(K_y K_z V\theta + K_x S\theta)$$

$$Q_z = P_z - P_x(K_x K_z V\theta - K_y S\theta) - P_y(K_y K_z V\theta + K_x S\theta)$$
$$- P_z(K_z^2 V\theta + C\theta)$$

5.15) The kinematics can be done easily to obtain:

$$^0P_{\text{YORG}} = \begin{bmatrix} (d_2 + L_2 + L_3)S_1 \\ -(d_2 + L_2 + L_3)C_1 \\ 0 \end{bmatrix}$$

$$^0V = {}^0J\dot{\theta}$$

$$^0J = \begin{bmatrix} \dfrac{\partial^0 P_{\text{YORGX}}}{\partial\theta_1} & \dfrac{\partial^0 P_{\text{YORGX}}}{\partial\theta_2} & \dfrac{\partial^0 P_{\text{YORGX}}}{\partial\theta_3} \\ \dfrac{\partial^0 P_{\text{YORGY}}}{\partial\theta_1} & \dfrac{\partial^0 P_{\text{YORGY}}}{\partial\theta_2} & \dfrac{\partial^0 P_{\text{YORGY}}}{\partial\theta_3} \\ \dfrac{\partial^0 P_{\text{YORGZ}}}{\partial\theta_1} & \dfrac{\partial^0 P_{\text{YORGZ}}}{\partial\theta_2} & \dfrac{\partial^0 P_{\text{YORGZ}}}{\partial\theta_3} \end{bmatrix}$$

So,

$$^0J = \begin{bmatrix} (d_2 + L_2 + L_3)C_1 & S_1 & 0 \\ (d_2 + L_2 + L_3)S_1 & -C_1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

5.16) You're in luck! The answer is given by Equation (5.42):

$$W = \begin{bmatrix} 0 & -S_1 & C_1S_2 \\ 0 & C_1 & S_1S_2 \\ 1 & 0 & C_2 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}$$

5.17) For a prismatic joint i, the motion of the tool due to the joint is:

$$^0V_i = \dot{d}_i \, {}^0\hat{z}_i$$

$$^0W_i = O$$

Now,

$$^0V_{\text{TOOL}} = {}^0V_1 + {}^0V_2 + \cdots + {}^0V_6$$

$$^0W_{\text{TOOL}} = {}^0W_1 + {}^0W_2 + \cdots + {}^0W_6$$

so,

$$J(\theta) = \left[\begin{array}{c:c:c:c:c:c} A_1 & A_2 & A_3 & A_4 & A_5 & A_6 \\ \hdashline B_1 & B_2 & B_3 & B_4 & B_5 & B_6 \end{array}\right]$$

Where each A_i or B_i is a 3×1 vector given by:

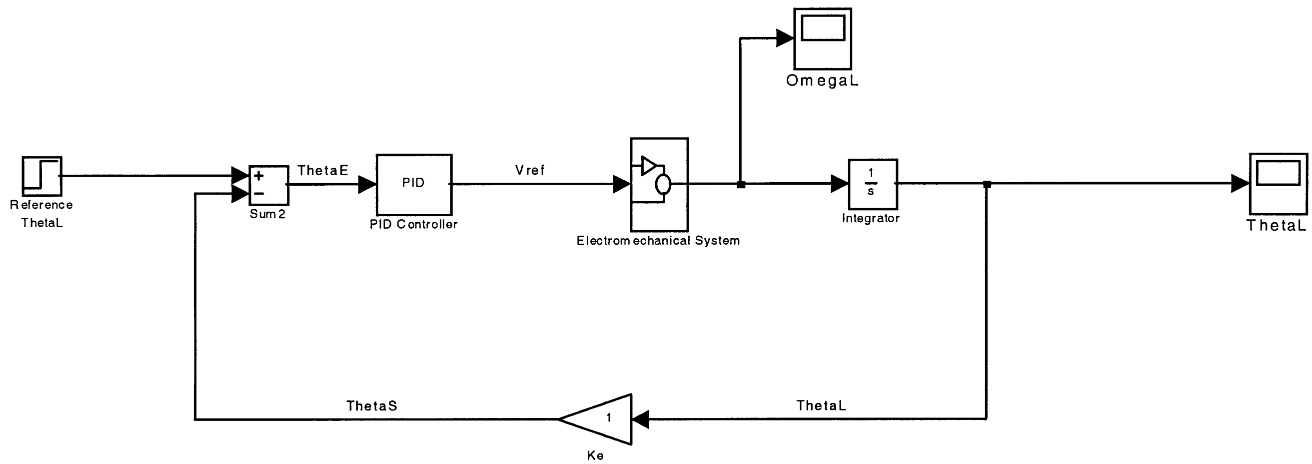
$$A_i = \begin{cases} {}^0\hat{Z}_i \otimes ({}^0P_{\text{tool}} - {}^0P_{\text{iorg}}) & \text{if } i \text{ revolute} \\ {}^0\hat{Z}_i & \text{if } i \text{ linear} \end{cases}$$

$$B_i = \begin{cases} {}^0\hat{Z}_i & \text{if } i \text{ revolute} \\ O & \text{if } i \text{ linear} \end{cases}$$

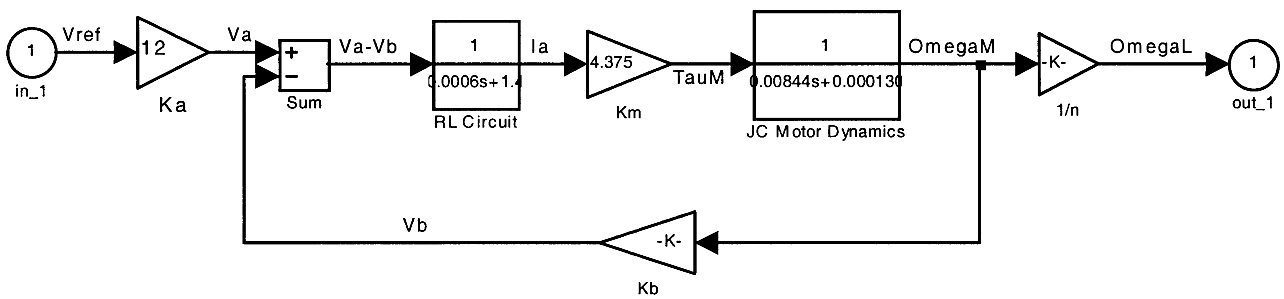
```
    ch: array[0..3] of char;
    grid: array[1..60,1..80] of char;

begin
xscale:=30.0;  yscale:=22.5;
ch[0]:='<';  ch [1]:='V';  ch[2]:='>';  ch[3]:='↑';
for i:=1 to 60 do
   for j:=1 to 80 do grid[i,j]:=' ';
for j:=1 to 80 do grid [31,j]:='-';
for i:=1 to 60 do grid [i,40]:='1';
for k:=1 to nticks do
  begin
  i:=31-round (trajectory[k][2] * yscale);
  j:=round(trajectory[k][1] * xscale)+40;
  grid[i,j]:=ch[(round(trajectory[k][3]+225) div 90) mod 4];
  end;
for k:=1 to npnt do
  begin
  where (viapnt[k],trels);
  itou (trels,place);
  i:=31-round (place[2] * yscale);
  j:=round (place[1] * xscale)+40;
  grid[i,j]:=' *'
  end;
for i:=1 to 57 do
  begin
  for j:=1 to 78 do write(grid[i,j]);
  writeln;
  end;
end;

BEGIN  { Main Test Program }
  INITDATA;
  WRITE ('Care to plan some swell robot paths? ');
  READLN (ans);
  WHILE  ans='y' DO
  BEGIN
    npnt:=1;
    for i:=1 to 3 do current[i]:=0.0;
{ enter the path via points and convert to joint angles }
     WRITELN ('Enter the initial position (x,y,phi): ');
     READLN (place[1],place[2],place[3]);
     UTOI (place,trels);
     SOLVE (trels,current,viapnt [npnt],far,sol);
     while sol do
       begin
       npnt:=npnt+1;
       writeln ('Enter the next via point (x,y,phi): ');
       writeln ('(enter a point with no sol to terminate) ');
       readln (place [1],place[2],place[3]);
       UTOI (place,trels);
       SOLVE (trels,viapnt[npnt-1],viapnt[npnt],far,sol);
       end;
     npnt:=npnt-1;
     JOINTVEL(viapnt,npnt,viavel);
     for i:=1 to  npnt-1 do
      for j:=1 to 3 do
       CUBCOEFF (viapnt[i][j],viapnt[i+1][j],viavel[i][j],
                                viavel[i+1][j],path[i,j]);
    RUNPATH(path,npnt);
    PLOTPATH;
    WRITE ('Would you like to do some more? ');
    READLN (ans);
  END;
END.
```

Matlab Exercise 9

Here is the Simulink implementation of the closed-loop feedback diagram:



where the Electromechanical system is the open-loop system:



Now, for trial-and-error PID controller design with the step input, there are infinite solutions for "good" performance. One possible solution is $K_P = 2$, $K_I = 4$, $K_D = 1$; the resulting load angle output from the closed-loop feedback control simulation is shown below: