Chapter 1
Introduction

Exercises

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

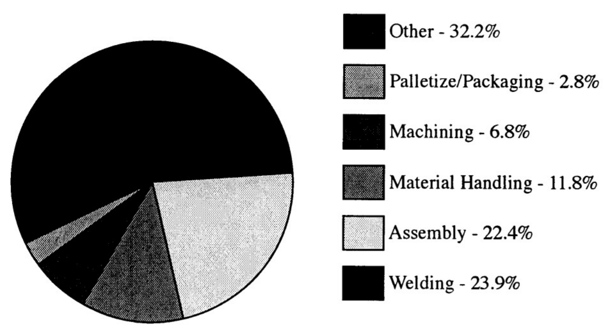
1955	Denavit & Hartenberg developed methodology for describing linkages.
1961	George Devol patents design of rst robot.
1961	First unimate robot installed.
1968	Shakey Robot developed at S.R.I.
1975	Robot institute of America formed.
1975	Unimation becomes rst Robot Co. to be protable.
1978	First Puma 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.

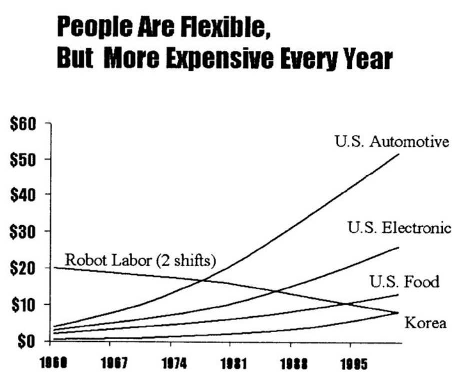
1.2) (Based on 1981 numbers)

Source:

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



1.3)



1.4) *Kinematics* is the study of position and derivatives of position without regard to forces which cause the motion. *Workspace* is the locus of positions and orientations achievable by the end-effector of a manipulator. *Trajectory* is a time based function which specifies the position (and higher derivatives) of the robot mechanism for any value of time.

1.5) *Frame* is a coordinate system, usually specified in position and orientation relative to some imbedding frame. *Degrees of freedom* is the number of independent variables which must be specified in order to completely locate all members of a (rigid-body) mechanism. *Position control* implies the use of a control system, usually in a closed-loop manner, to control the position of one or more moving bodies.

1.6) *Force control* is the use of (usually closed-loop) algorithms to control the forces of contact generated when a robot touches its work environment. A *robot programming language* is a programming language intended for use in specifying manipulator actions.

1.7) *Structural stiffness* is the "K" in $F = K\Delta X$ (A.K.A "Hooke's law") which describes the rigidity of some structure. *Nonlinear control* refers to a closed loop control system in which either the system to be controlled, or the control algorithm itself is nonlinear in nature. *Off line programming* is the process of creating a program for a device without access to that device.

1.8) See references. For example, in 1985 average labor costs of $15 to $20 per hour are reasonable (depending how fringe benefits are calculated).

1.9) Obviously it has increased dramatically. Recently (1988–1990) the ratio doubles or even triples each year.

1.10) See Figure 1.3, but use latest data you can find.

Chapter 2
Spatial Descriptions and Transformations

Exercises

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

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

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

2.2) $R = \text{rot}(\hat{x}, 45°)\,\text{rot}(\hat{y}, 30°)$

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

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

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

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

We might also use the following reasoning:

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

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

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

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

Yet another way of viewing the same operation:

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

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

2.3) *(continued)*

(This is a similarity transform)

Composing these two rotations:

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

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

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

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

2.4) This is the same as 2.3 only with numbers.

$R = \text{rot}(\hat{z}, 30°)\,\text{rot}(\hat{x}, 45)$

$$= \begin{bmatrix} .866 & -.353 & .353 \\ .50 & .612 & -.612 \\ 0 & .707 & .707 \end{bmatrix}$$

2.5) If V_i is an eigenvector of R, then

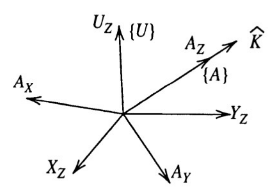
$RV_i = 7V_i$

If the eigenvalue associated with V_i is 1, then

$RV_i = V_i$

Hence the vector is not changed by the rotation R.
So V_i is the axis of rotation.

2.6) Imagine a frame $\{A\}$ whose $\hat{z}$ axis is aligned with the direction $\hat{k}$:



Then, the rotation with rotates vectors about $\hat{k}$ by θ degrees could be written:

$R = {}^{U}_{A}R\,\text{rot}({}^{A}\hat{z}, \theta)\,{}^{A}_{U}R \quad [1]$

We write the description of $\{A\}$ in $\{U\}$ as:

$${}^{U}_{A}R = \begin{bmatrix} A & D & K_x \\ B & E & K_y \\ C & F & K_z \end{bmatrix}$$

If we multiply out Eq. [1] above, and then simplify using $A^2 + B^2 + C^2 = 1$, $D^2 + E^2 + F^2 = 1$, $[ABC] \cdot [DEF] = 0$, $[ABC] \otimes [DEF] = [K_x K_y K_z]$ we arrive at Eq. (2.80) in the book. Also, see [R. Paul]* page 25.