2103-530
Industrial Robots 1

Phongsaen Pitakwatchara

November 16, 2014

To those who work hard in robotics

Contents

Preface

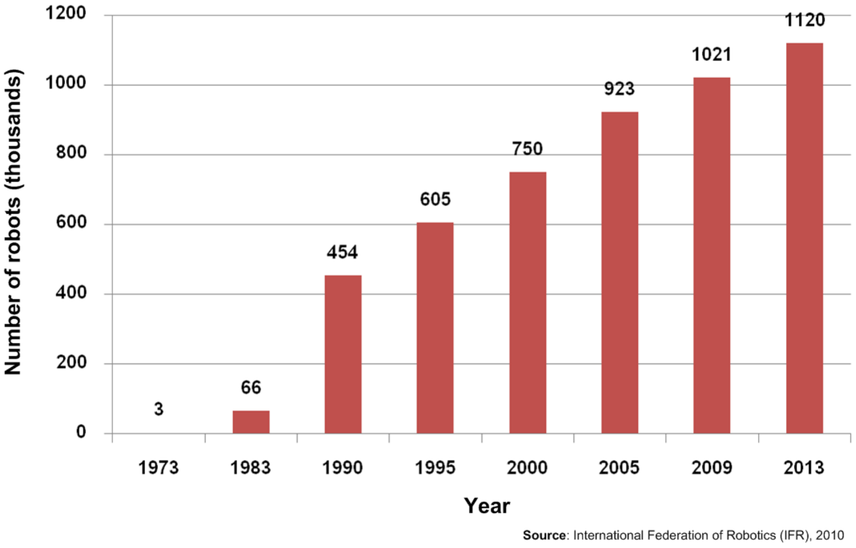
This is the lecture note for the course 2103-530 “Industrial Robots 1” taught at Chulalongkorn University. Audiences of this course are the fourth year standing undergraduate students as well as graduate students who are interested in robotics and never have taken any course in robotics before. This one-semester course introduces fundamental topics in robotics science emphasizing in its mechanical dynamics. In particular, the course takes the serial manipulator as an examplar and studies the mathematical modeling of it. For this introductory course, topics to be covered are the kinematics, trajectory generation, mechanism design, and introductory to control of the manipulator. The continuing course 2103-630 “Industrial Robots 2” will use these basic knowledge in studying dynamics and control of it.

Although I have worked my best to prepare and revise the lecture note, there might be some uncaught errors or some poorly explained issues. Therefore, I will be very grateful for any notice or comment which will help me improving the material. Please send them to `phongsaen@gmail.com`. Lastly, I hope this lecture note will be useful to the students and readers for their studies and careers.

Chulalongkorn University — Phongsaen Pitakwatchara
November 2014

Chapter 1

Introduction



Source: International Federation of Robotics (IFR), 2010

Figure 1.1: Bar chart showing the increasing number of the industrial robots used worldwide.

Robotics is an applied science that emerges from the utilization of knowledge in many disciplines together for analyzing and designing the robots. Such fundamentals sciences are, for example, mechanical engineering which will be useful for the design and analysis of the mechanisms that can produce the desired motion; electrical engineering concerns with the electronics and circuits in controlling the robots; computer engineering that addresses the efficient algorithms and program development. In this introductory course, the topics to be pursued shall be limited to the mechanical aspects of the serial type manipulators, however.

1.1 Background

According to the Robot Institute of America (RIA), the robot is defined as

> A robot is a reprogrammable, multifunctional manipulator designed to move materials, parts, tools, or specialized devices through variable programmed motion for the performance of a variety of tasks.

This definition reflects the characteristics of the industrial robots in 1980's that have main usage in the assembly lines. Nevertheless, the robots has undergone substantial development such that the above definition cannot cover all today's existed robots. Few examples are humanoid robots or the insect robots. However, industrial robots still possess the largest share in the market. The International Federation of Robotics (IFR) has estimated that, by the end of 2013, the number of the industrial robots will be raised to 1,120,000 as shown in Fig. 1.1.

There are essentially four main components comprising the robotic system, which are

1. *Robot* is the physical structure which performs the task corresponding to the peripheral information and the control processing. The structure may be divided into two main parts. The first one is the locomotion, which is the mechanism that enables the motion of the robot from one place to another place. The simplest mechanism, also being the most widely employed, is the wheel mechanism. Although the legged mechanism is much more complicated, it gives the robot more freedom to move. Another part of the structure is the manipulation part which is mainly responsible for the task execution. Common structures of this part yield the appearance of the hand and the arm.

2. *Actuator* is the part which causes the robot motion and makes it perform the desired task. By the current technology limitation, most actuators employed in the robotic system today are motors. Often they are equipped with the reduction and transmission mechanisms. Other common kinds of the actuators are fluid-power driven actuators such as hydraulics and pneumatics. Novel actuators are made of synthesized polymer-form materials and have their usage in bio-inspired robotic systems, for example.

3. *Sensor* is the components that acquires many information about the robot system itself and the environment in real time. These information will be further processed and transferred to the control unit. Information from different kinds of sensors may possess different levels of complexity. Position and velocity information from the encoder and the tachometer are relatively simple compare to the image data retrieving from the camera.

4. *Processing and control unit* has the role which is comparable to the human brain. It receives the raw information from various sensors. They are fused and processed to appropriate forms as the commands for the control unit in calculating the corresponding control signals. These are then sent to the actuators, driving the robot to perform the desired tasks.

Relationship between these four components are depicted in Fig. 1.2. Boundaries for these unit might be physically vague in some systems due to the integration of them to produce more efficient robotic systems.

1.2 The Mechanics and Control of Manipulators

Before studying in details the fundametals of the mechanics of the robot, this section offers a short overview about regulating the robot at large. The problem is to control the robot end effector such that its motion tracks the given desired

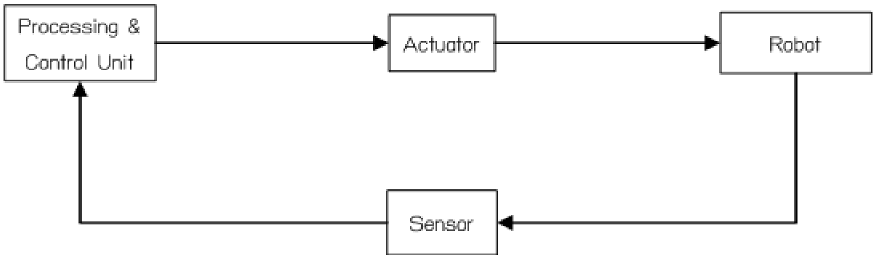


Figure 1.2: Robot components.

trajectory. This trajectory may be generated such that it causes the grasped object to be moved to a new desired position.

Generally, the motion of the robot end effector happens in the three dimensional space, involving both the position and the orientation. Especially, the three dimensional orientation is far more complicated than the degenerated in-plane orientation. Therefore, as a starting it is necessary to first describe the desired three dimensional motion of the end effector by means of some suitable representations.

To achieve the goal of controlling the motion of the end effector, the appropriate force and torque must be provided to the robot typically at the joints. These required force and torque may be calculated via solving the robot's equation of motion. However, the given desired trajectory is usually described in the robot task space, which is not appropriate for such calculation directly. This can be fixed by transforming the desired end effector trajectory to the equivalent desired robot joint motion, known as the inverse kinematics calculation.

The dynamic equations of the robot are the set of nonlinear differential equations governing the relationship between the applied force and torque and the resulting motion of the robot. This dynamics may be derived from the kinematics of the robot and its inertial parameters. When the desired trajectory has been transformed into the joint space already, dynamic equations of the robot in joint space may be used to calculate the necessary joint torque and force, called the computed torque control. These joint torques may further be used as the reference signals of the controller through the actuators.

1.3 Course Overview

"2103-530: Industrial Robots 1" is considered an introductory course in robotics where the study is confined to the mechanical aspects of the serial type manipulators. It starts with chapter 2 where various methods in describing the position and orientation in three dimensional space are provided. Related operations such as the rigid body translation and rotation are considered as well. This chapter serves as the important basis for the mechanics study of the robot.

Chapter 3 and 4 are closely related. The process of calculating the corresponding robot posture (position and orientation) from the specified robot's joint variables, called the forward kinematics, is explained in Chapter 3 based on the modified Denavit-Hartenberg convention. Chapter 4 is the reverse problem. It is related to calculating the corresponding robot's joint variables from the given robot posture, commonly by specifying its end effector posture.

Chapter 5 extends the kinematical analysis of the robot to the velocity level. Basically, the relationship between the end effector velocity and the joint velocity is linear, which is controlled by the Jacobian matrix. By the duality, Jacobian matrix also regulates the static relationship between the force acting at the end effector and the torque applied at the joint. Rank deficient of this matrix signals the robot is in singular condition.

Once the study of robot kinematics is completed, generation of the desired end effector trajectory may be possible. Chapter 6 treats such topic on the account of the kinematical aspect solely. The path may be planned either in the joint space, or more naturally in the Cartesian space.

The next two chapters 7 and 8 provides a glimpse on the design of the robot and the introductory to the hybrid force/position control. The material may instantly found to be useful in the applications. Finally, the appendix provides a concise explanation of the Robotics Toolbox based on MatLAB® [6]; an excellent tool in analysis and design of the robot system.

Chapter 2

Spatial Descriptions and Transformations

In this chapter, methods of describing the position and the orientation of the objects in three dimensional space will be discussed. This can be achieved through the notion of the frame, which leads naturally to the rotation matrix and, more generally, the homogeneous transformation matrix introduced in subsection 2.2.3. The description can be viewed from different angles, which leads to the other two interpretations of this result: the mapping and the operator, explained in section 2.2 and 2.3.

Several transformations may be combined to produce a new, usually more complicated, compound transformation described in section 2.4. It is proven to be a useful tool in the analysis of the robot kinematics. In the last section, some other different ways of describing the orientation will be investigated.

2.1 Descriptions: Positions, Orientations, and Frames

2.1.1 Description of a position

The fundamentals of kinematical analysis is the means to describe or express the position and orientation. It may be rather intuitive in planar motion for which the position is described by the position vector and the orientation by the angle. Working with the general robot inevitably needs to analyze the motion in the three dimensional space. Therefore, here the kinematical problems are treated in a formal way for the three dimensional case.

First of all, although vector quantities are inherently independent of the coordinates, to describe or express them, coordinate systems have to be defined. The coordinate system consists of the *frame*, the origin of the frame, and the coordinates. The most commonly used one is the Cartesian coordinate system consisting of the rectangular frame, the origin, and the coordinates $x - y - z$ depicting the distances of the point with respect to the origin along the coordinate axes of the frame. Figure 2.1 shows a point P represented in two different coordinate frames $\{1\}$ and $\{2\}$.

In Cartesian coordinate system, the position vector $\bar{p}$ may be expressed by a column vector of the dimension 3×1;

$$^{A}\bar{p} = \begin{bmatrix} p_x & p_y & p_z \end{bmatrix}^T,$$

where the letter A on the left superscript denotes the frame in which this vector is described. If the frame has not been named, its coordinates $\{x\,y\,z\}$ may be used explicitly. Often, a point is commonly presented in several different frames. It is obvious from Fig. 2.1 that

$$^{A}\bar{p} \neq {}^{B}\bar{p}.$$

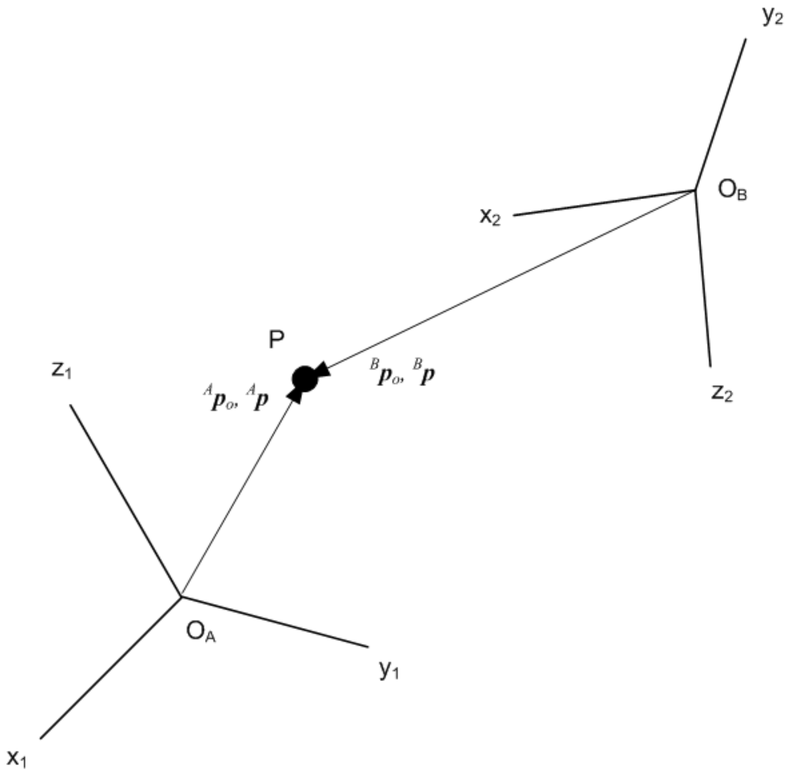

y_2
x_2
O_B
P
z_1
${}^{B}\boldsymbol{p}_o$, ${}^{B}\boldsymbol{p}$
${}^{A}\boldsymbol{p}_o$, ${}^{A}\boldsymbol{p}$
z_2
O_A
y_1
x_1

Figure 2.1: A position vector and two Cartesian coordinate systems. ([1], pp. 19)

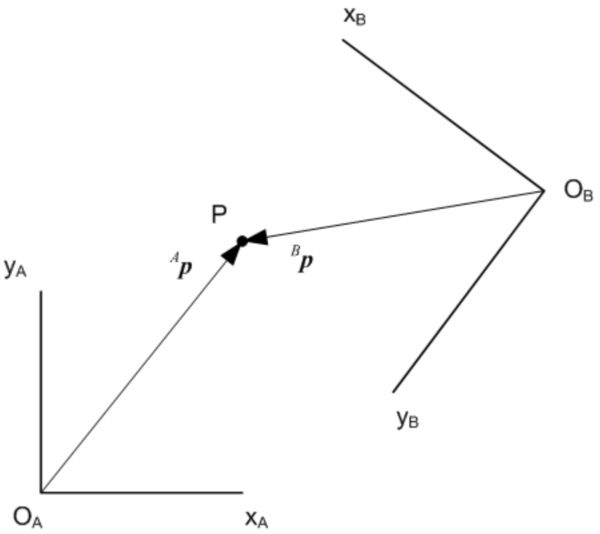


Figure 2.2: Example 2.1. ([1], pp. 20)

Example 2.1 Refer to Fig. 2.2. Write the column vector for the following position vector: ${}^{A}\bar{p}$, ${}^{B}\bar{p}$, ${}^{B}\bar{o}_A$, ${}^{A}\bar{o}_B$. ([1], Prob. 2.1)

Solution By the direct measurement of the distance of point P with respect to the origin of $\{A\}$ and $\{B\}$ along their coordinate axes, position vector of P may be expressed as

$$
\begin{aligned}
{}^{A}\bar{p} &= \begin{bmatrix} 4 & 5 \end{bmatrix}^T, \\
{}^{B}\bar{p} &= \begin{bmatrix} 4.2 & 4.4 \end{bmatrix}^T.
\end{aligned}
$$

${}^{B}\bar{o}_A$ and ${}^{A}\bar{o}_B$ are the position vector of the origin of $\{A\}$ described in $\{B\}$ and the position vector of the origin of $\{B\}$ described in $\{A\}$, which can be determined in a similar manner.

$$
\begin{aligned}
{}^{B}\bar{o}_A &= \begin{bmatrix} 4.4 & 10.8 \end{bmatrix}^T, \\
{}^{A}\bar{o}_B &= \begin{bmatrix} 10 & 6 \end{bmatrix}^T.
\end{aligned}
$$

From the example, it can be said that in general, ${}^{B}\bar{o}_A \neq {}^{A}\bar{o}_B$ unless the coordinate axes of the frames are parallel and opposite to each other.

2.1.2 Description of an orientation

There are many ways to describe an orientation in three dimensional space. In this section, the clearest and widely used method of such will be explained. In order to describe the three dimensional orientation of an object, a frame called the *body-fixed frame* is attached to the object first. The orientation of the frame is then the same as the orientation of the body. Orientation of the body-fixed frame may be described with respect to another frame called the *reference frame*, which needs not be fixed. The origins of both frames may be placed at the same location for the sake of clarification. Figure 2.3 illustrates the body-fixed frame $\{x'\,y'\,z'\}$ and the reference frame $\{x\,y\,z\}$.

Orientation of the Cartesian body-fixed frame may be represented by the 3×3 matrix. Consider Fig. 2.4 showing two frames. $\{A\} = \{x\,y\,z\}$ is the reference frame and $\{B\} = \{x'\,y'\,z'\}$ is the body-fixed frame. A unit vector along the coordinate axes of $\{B\}$ may be written in $\{A\}$ by projecting the vector onto the coordinate axes of $\{A\}$. The projections are its components along the coordinate axes of $\{A\}$, for which they add up to provide the original unit vector. Mathematically,

$$
\begin{aligned}
\hat{i}' &= \cos\theta_{x'x}\hat{i} + \cos\theta_{x'y}\hat{j} + \cos\theta_{x'z}\hat{k} \\
\hat{j}' &= \cos\theta_{y'x}\hat{i} + \cos\theta_{y'y}\hat{j} + \cos\theta_{y'z}\hat{k} \\
\hat{k}' &= \cos\theta_{z'x}\hat{i} + \cos\theta_{z'y}\hat{j} + \cos\theta_{z'z}\hat{k},
\end{aligned}
\tag{2.1}
$$

where $\theta_{x'x}$ is the angle between the axes x' and x, etc. Also $\cos\theta_{x'x}$ is the directional cosine of the vector $\hat{i}'$ along the direction of the x-axis. By this

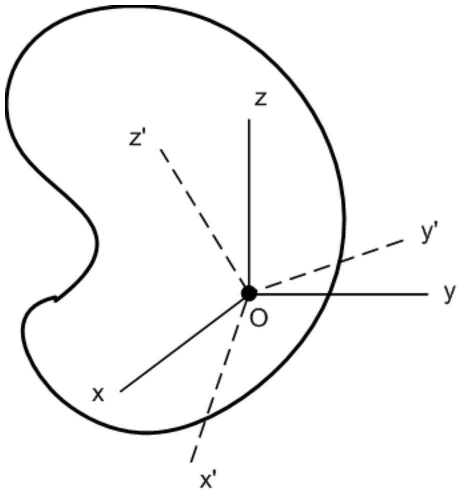


Figure 2.3: A body-fixed frame $\{x'\, y'\, z'\}$ affixed to the object. ([1], pp. 22)

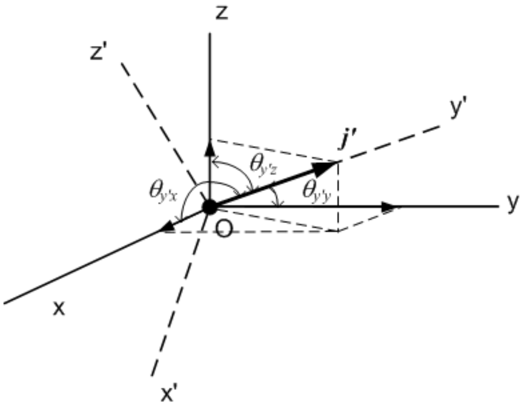


Figure 2.4: Projection of a unit vector along the body-fixed coordinate axis onto the reference frame. ([1], pp. 19)

means, Eq. 2.1 describes the orientation of the moving frame $\{B\}$ and therefore the orientation of the object.

It is convenient to express the above equations in a single compact unit with the help of the matrix. Commonly adopted notation in robotics for this matrix is to write the directional cosine vectors of the coordinate axes of the moving frame, represented in the fixed frame, in a column-wise manner. This matrix is called a *rotation matrix.* According to Eq. 2.1, the rotation matrix of $\{B\}$ with respect to $\{A\}$ is

$$ {}^{A}_{B}R = \left[\begin{array}{ccc} {}^{A}\hat{\imath}_B & {}^{A}\hat{\jmath}_B & {}^{A}\hat{k}_B \end{array} \right] = \left[\begin{array}{ccc} c\theta_{x'x} & c\theta_{y'x} & c\theta_{z'x} \\ c\theta_{x'y} & c\theta_{y'y} & c\theta_{z'y} \\ c\theta_{x'z} & c\theta_{y'z} & c\theta_{z'z} \end{array} \right]. \tag{2.2} $$

The name of the frame to describe the orientation is subscripted and the name of the frame which this orientation is measured with respect to is superscripted. In addition, the equation also shows the shorthand notation of $\cos\theta$ as $c\theta$, which is very useful for typically long expressions in robotics. Other relevant notations are $s\theta$ and $t\theta$ for $\sin\theta$ and $\tan\theta$, respectively.

Using the fact that the directional cosine of arbitrary two vectors may be obtained by computing the scalar product of their unit vectors, elements of the rotation matrix may be expressed in a different way. Namely, with the expression of

$$ \cos\theta_{MN} = \frac{\bar{M}}{\|\bar{M}\|} \cdot \frac{\bar{N}}{\|\bar{N}\|}, \tag{2.3} $$

the rotation matrix in Eq. 2.2 becomes

$$ {}^{A}_{B}R = \left[\begin{array}{ccc} \hat{\imath}' \cdot \hat{\imath} & \hat{\jmath}' \cdot \hat{\imath} & \hat{k}' \cdot \hat{\imath} \\ \hat{\imath}' \cdot \hat{\jmath} & \hat{\jmath}' \cdot \hat{\jmath} & \hat{k}' \cdot \hat{\jmath} \\ \hat{\imath}' \cdot \hat{k} & \hat{\jmath}' \cdot \hat{k} & \hat{k}' \cdot \hat{k} \end{array} \right]. \tag{2.4} $$

If the rotation matrix is read row-wise instead, each row is seen as the directional cosine vectors of the coordinate axes of the fixed frame described in the moving frame. Hence, it may be written as

$$ {}^{A}_{B}R = \left[\begin{array}{c} \left({}^{B}\hat{\imath}_A\right)^T \\ \left({}^{B}\hat{\jmath}_A\right)^T \\ \left({}^{B}\hat{k}_A\right)^T \end{array} \right]. $$

With the symmetry of $\{A\}$ and $\{B\}$, the rotation matrix of $\{A\}$ with respect to

$\{B\}$ may be expressed as

$$ {}^{B}_{A}R = \begin{bmatrix} {}^{B}\hat{i}_A & {}^{B}\hat{j}_A & {}^{B}\hat{k}_A \end{bmatrix} = \begin{bmatrix} \left({}^{B}\hat{i}_A\right)^T \\ \left({}^{B}\hat{j}_A\right)^T \\ \left({}^{B}\hat{k}_A\right)^T \end{bmatrix}^T , $$

which is obviously the transpose of ${}^{A}_{B}R$. Particularly,

$$ {}^{B}_{A}R = \left({}^{A}_{B}R\right)^T . \tag{2.5} $$

Furthermore, since

$$ \left({}^{A}_{B}R\right)^T {}^{A}_{B}R = \begin{bmatrix} \left({}^{A}\hat{i}_B\right)^T \\ \left({}^{A}\hat{j}_B\right)^T \\ \left({}^{A}\hat{k}_B\right)^T \end{bmatrix} \begin{bmatrix} {}^{A}\hat{i}_B & {}^{A}\hat{j}_B & {}^{A}\hat{k}_B \end{bmatrix} = I_3, $$

where I_3 is the 3×3 identity matrix, it may be concluded that

$$ R^{-1} = R^T . \tag{2.6} $$

In summary, calculation of the inverse of the rotation matrix is simply its transpose. This nice property holds because the rotation matrix is an orthonormal matrix. Moreover, it can be shown that if the physical frame is the right-handed coordinate frame, determinant of the rotation matrix is equal to $+1$.

Example 2.2 Fig. 2.5 shows $\{B\}$ which is rotated relative to $\{A\}$ about the common y-axis by $-150°$. The y-axis is pointing out of the page. Determine the rotation matrix ${}^{A}_{B}R$ and ${}^{B}_{A}R$.

Solution Equation 2.4 reveals the structure of the rotation matrix as the collection of the dot product of the unit vectors along the coordinate axes. From Fig. 2.5, these dot products may be formed visually. Then, applying Eq. 2.4 for this problem,

$$ {}^{A}_{B}R = \begin{bmatrix} -\frac{\sqrt{3}}{2} & 0 & -\frac{1}{2} \\ 0 & 1 & 0 \\ \frac{1}{2} & 0 & -\frac{\sqrt{3}}{2} \end{bmatrix} $$

and

$$ {}^{B}_{A}R = \begin{bmatrix} -\frac{\sqrt{3}}{2} & 0 & \frac{1}{2} \\ 0 & 1 & 0 \\ -\frac{1}{2} & 0 & -\frac{\sqrt{3}}{2} \end{bmatrix} . $$

Cross-checking with the property in Eq. 2.5, the result is verified.

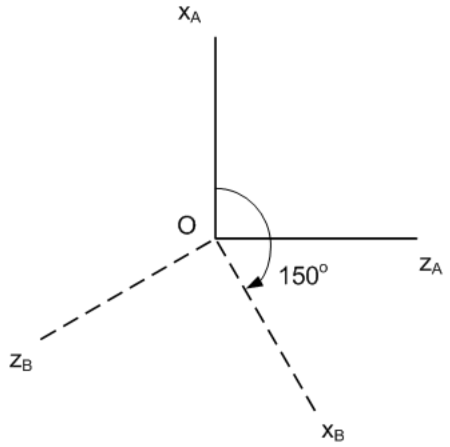


Figure 2.5: Example 2.2.

2.1.3 Description of a frame

Position and orientation are needed to completely specify the *posture.* For an object, its posture may be described by attaching a body-fixed frame onto it. Then the object's posture may be determined by specifying the position of the frame's origin and the orientation of the frame itself, which is the description of a frame. From section 2.1.1 and 2.1.2, a position may be described by the 3×1 position vector and an orientation may be described by three 3×1 directional vectors, or the 3×3 rotation matrix.

In Fig. 2.6, frame $\{B\}$ may be described with respect to $\{A\}$ by ${}^A_BR$ and ${}^A\bar{o}_B$, where the latter denotes the position vector of the origin of $\{B\}$ written in $\{A\}$ coordinates. Conceptually,

$$\{B\} = \left\{\ {}^A_BR,\quad {}^A\bar{o}_B\ \right\}.$$

In the next section, the homogeneous transform is introduced where a frame can be described by the 4×4 homogeneous transformation matrix.

2.2 Mappings: Changing the Description

Often, there is a need in expressing the same quantity in terms of several reference coordinate systems. As depicted in Fig 2.1, a position vector $\bar{p}$ may be expressed in either $\{A\}$ or $\{B\}$. In this section, a formal method in *mapping* or changing the description from one frame to another frame is explained. Note that the mapping does not change the quantity *per se*; only the representation is changed.

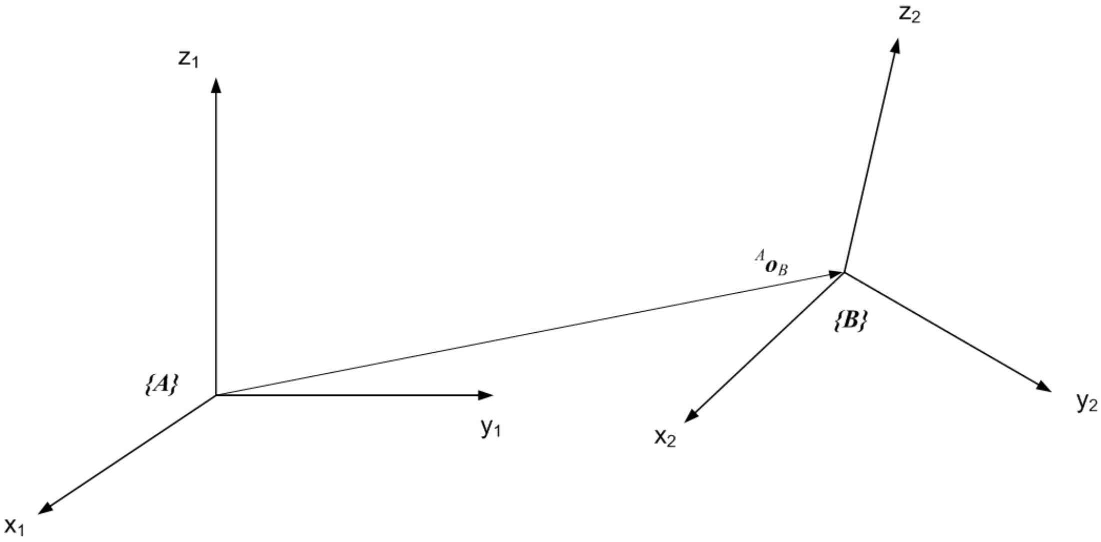


Figure 2.6: Two frames positioned and oriented differently by $^A\bar{o}_B$ and $^A_B R$.

2.2.1 Mapping involving translated frame

Consider Fig. 2.7 which illustrates a point P and two parallel frames $\{A\}$ and $\{B\}$. A position of the point P may be described in either frame, for which their relationships are governed by the following equation;

$$^A\bar{p} =^B \bar{p} + {}^A\bar{o}_B. \tag{2.7}$$

This equation may be viewed as the (translational) mapping of the vector from $\{B\}$ to $\{A\}$, where $^A\bar{o}_B$ defines the mapping.

2.2.2 Mapping involving rotated frame

Now consider Fig. 2.8 which illustrates a free vector $\bar{p}$ and two rotated frames $\{A\}$ and $\{B\}$. The vector may be expressed in $\{A\}$ as

$$^A\bar{p} = \left[\begin{array}{ccc} p_x & p_y & p_z \end{array}\right]^T,$$

or in $\{B\}$ as

$$^B\bar{p} = \left[\begin{array}{ccc} p_{x'} & p_{y'} & p_{z'} \end{array}\right]^T.$$

To determine the mapping between the two rotated frames, note that the components of any vector are just the projections of that vector onto the unit vector along the coordinate axes. As a consequence, the components of $^A\bar{p}$ may

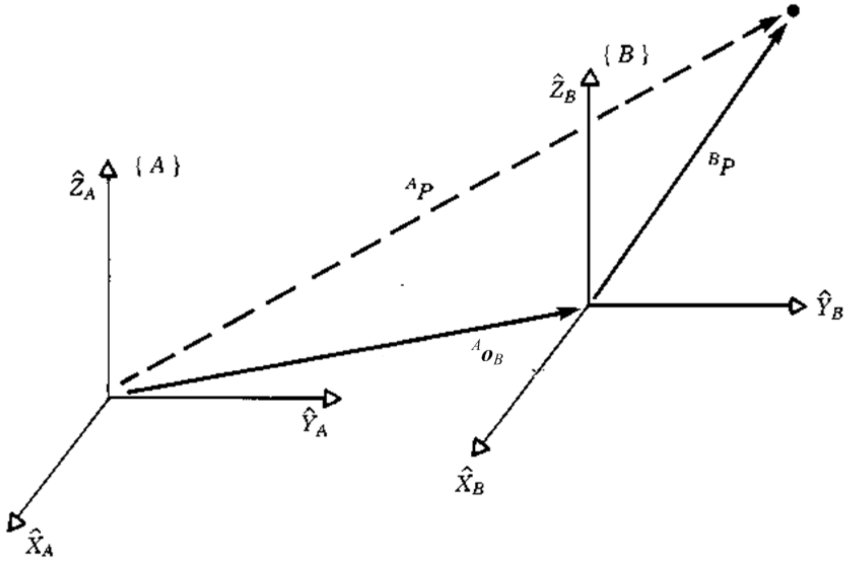


Figure 2.7: Position of a point P described in two parallel frames $\{A\}$ and $\{B\}$. ${}^A\bar{o}_B$ defines the mapping. ([4], pp. 24)

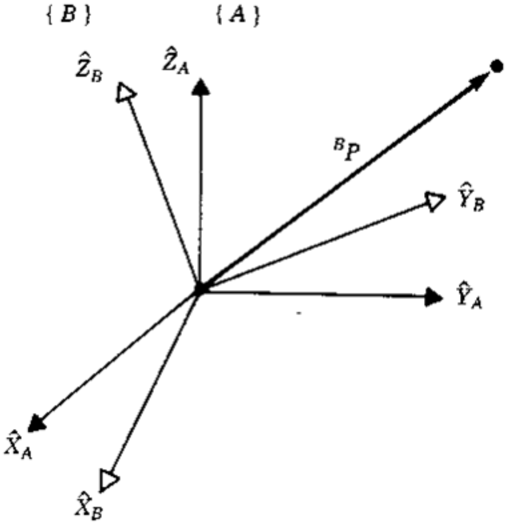


Figure 2.8: A free vector $\bar{p}$ may be expressed in two rotated frames $\{A\}$ or $\{B\}$. ${}^A_BR$ defines the mapping. ([4], pp. 25)

be calculated as

$$\begin{aligned} p_x &= {}^B\hat{\imath}_A \cdot {}^B\bar{p} \\ p_y &= {}^B\hat{\jmath}_A \cdot {}^B\bar{p} \\ p_z &= {}^B\hat{k}_A \cdot {}^B\bar{p}. \end{aligned}$$

The equations may then be arranged in the matrix-vector form as

$$ {}^A\bar{p} = {}^A_BR\,{}^B\bar{p}. \tag{2.8}$$

Hence this equation may be viewed as the (rotational) mapping of the vector from $\{B\}$ to $\{A\}$, where ${}^A_BR$ defines the mapping.

2.2.3 Mapping involving general frames

Lastly, let consider Fig. 2.9 which involves a vector $\bar{p}$ and two frames $\{A\}$ and $\{B\}$ arranged in an arbitrary manner. Relative posture of these frames are specified by ${}^A_BR$ and ${}^A\bar{o}_B$. The description of the vector $\bar{p}$ may be changed from $\{B\}$ to $\{A\}$ by the following steps. First introduce an intermediate frame $\{I\}$ which has the same orientation as $\{A\}$ but its origin is coincident with the origin of $\{B\}$. From section 2.2.2 of the rotational mapping, the vectors $\bar{p}$ represented in $\{B\}$ and $\{I\}$ obey the following relationship;

$$ {}^I\bar{p} = {}^A_BR\,{}^B\bar{p}.$$

Next, the change of description may proceeds to $\{A\}$ readily by the translational mapping in section 2.2.1;

$$ {}^A\bar{p} = {}^I\bar{p} + {}^A\bar{o}_B.$$

Collectively the general mapping of the vector from $\{B\}$ to $\{A\}$ may be formulated as

$$ {}^A\bar{p} = {}^A_BR\,{}^B\bar{p} + {}^A\bar{o}_B. \tag{2.9}$$

Rotation and translation may be treated as an integral unit of general mapping ${}^A_BT$, which packs the above equation as the *homogeneous mapping*

$$ {}^A\bar{P} = {}^A_BT\,{}^B\bar{P}, \tag{2.10}$$

where the quantities are expressed in the homogeneous four dimensional space. In particular,

$$\begin{bmatrix} {}^A\bar{p} \\ 1 \end{bmatrix} = \begin{bmatrix} {}^A_BR & {}^A\bar{o}_B \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^B\bar{p} \\ 1 \end{bmatrix}.$$

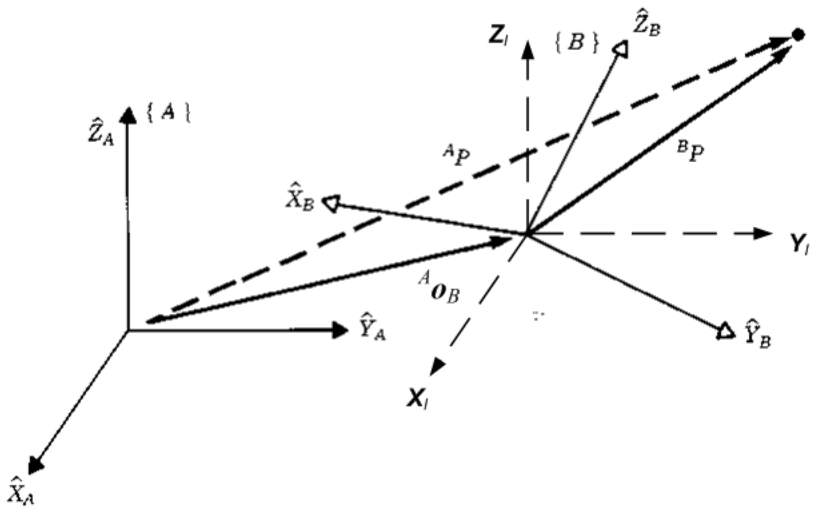


Figure 2.9: A vector $\bar{p}$ may be expressed in arbitrary two frames $\{A\}$ and $\{B\}$, of which its relative posture is described by ${}^A_B T$.

Note that the vector in the homogeneous space is the vector in the usual three dimensional space appended with the fourth element of 1. The matrix

$$ {}^A_B T = \begin{bmatrix} {}^A_B R & {}^A\bar{o}_B \\ 0 & 1 \end{bmatrix} \tag{2.11} $$

is called the *homogeneous transformation matrix*, which collects the information of both the relative position and orientation in the same place. Referring back to section 2.1.3, the homogeneous transformation matrix can then be used as an description of the frame.

Example 2.3 The origin of $\{B\}$ is located relative to the origin of $\{A\}$ by 3 units along y_A-axis and 1 unit along z_A-axis. The orientation of $\{B\}$ is generated from the rotation of $\{A\}$ by 90° about the x-axis. If the position vector of the point P, described in $\{A\}$ is $\begin{bmatrix} 0 & 2 & 2 \end{bmatrix}^T$, determine the position vector for the same point in $\{B\}$.

SOLUTION Figure 2.10 shows the point P, the frames $\{A\}$ and $\{B\}$, and the position vector ${}^A\bar{p}$ and ${}^B\bar{p}$ as the problem stated. To determine the unknown ${}^B\bar{p}$, the homogeneous mapping Eq. 2.10 may be applied as

$$ {}^B\bar{P} = {}^B_A T {}^A\bar{P}, $$

of which

$$ {}^A\bar{P} = \begin{bmatrix} 0 \\ 2 \\ 2 \\ 1 \end{bmatrix} \qquad {}^B_A T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & -1 & 0 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

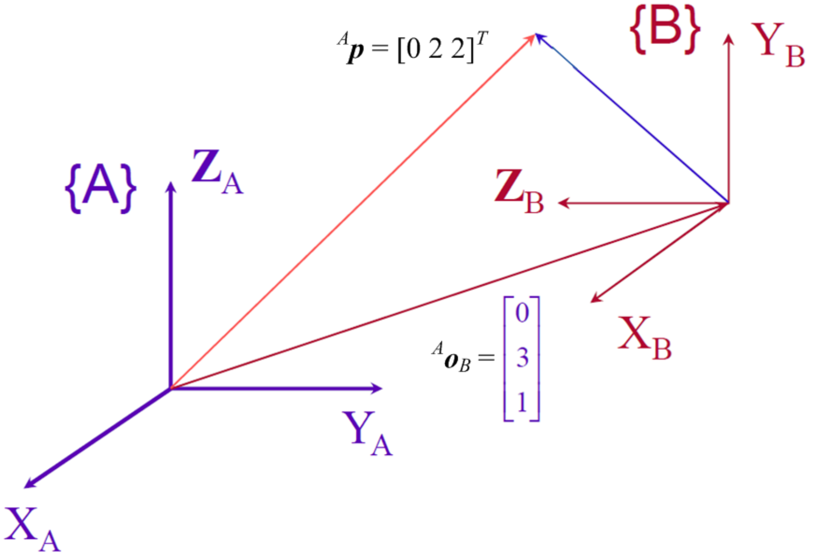

${}^A\boldsymbol{p} = [0\ 2\ 2]^T$
{B}
Y_B
{A}
Z_A
Z_B
X_B
${}^A\boldsymbol{o}_B = \begin{bmatrix} 0 \\ 3 \\ 1 \end{bmatrix}$
Y_A
X_A

Figure 2.10: Example 2.3.

for this problem. By simple matrix multiplication,

$$ {}^B\bar{P} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}, $$

as can be verified directly from the figure.

2.3 Operators

Robot motion in general may be imagined as the simultaneous translation and rotation of the connecting rigid links altogether. For the purpose of describing these actions mathematically, basic tools acting as the operators to move the object, which is just a collection of infinitely many points, are needed. In fact, the description of frame, on the other view, serves for this objective.

2.3.1 Translational Operators

Consider Fig. 2.11 which shows a frame and two points P and Q. Point Q is constructed from translating the original point P along the direction and distance dictated by the vector $\bar{o}$. In other words, a point P is operated by the translational operator, of which its result is the newly translated point Q. This operator in action may be written vectorially as

$$ \bar{q} = Trans\left(\bar{o}\right)\bar{p} = \bar{p} + \bar{o}. \tag{2.12} $$

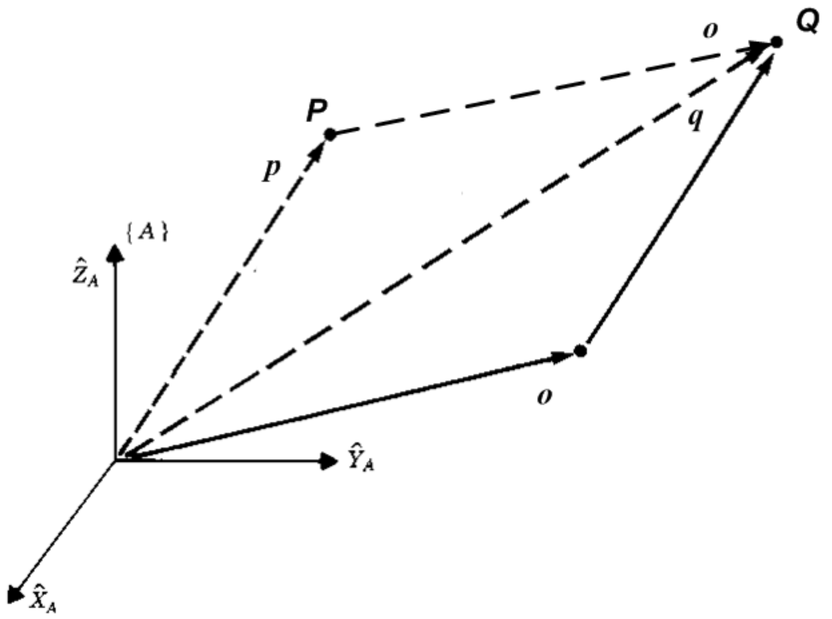


Figure 2.11: A point P is operated by the translational operator, $Trans\,(\bar{o})$, of which its result is the newly translated point Q.

Comparing Eq. 2.12 to Eq. 2.7, the intimate roles of the mapping and the operator may be understood. For the mapping, the resulting vector represents the same point in the new frame $\{A\}$ that is translated by $-\bar{o}$. For the operator, the resulting vector corresponds to the new point which occurs by the translating action of $\bar{o}$. It may be said that the relative position may be obtained by the motion of the point, or by the opposite motion of the representing frame.

2.3.2 Rotational Operators

Now consider Fig. 2.12 which illustrates two frames $\{A\} = \{x\,y\,z\}$ and $\{B\} = \{x'\,y'\,z'\}$, and two points P and Q. Point Q is in this case constructed from rotating the original point P around the origin. Frame $\{B\}$ is the body-fixed frame rotated from $\{A\}$ along with the point Q.

According to this setup, components of the vector of the rotated point Q described in the rotating frame $\{B\}$ must be the same as those of the original point P represented in the initial frame $\{A\}$. That is

$$^{B}\bar{q} = {}^{A}\bar{p}.$$

From the mapping point of view, the vector $\bar{q}$ may be represented in $^{A}\bar{q}$ as well by

$$^{A}\bar{q} = {}^{A}_{B}R\,{}^{B}\bar{q}.$$

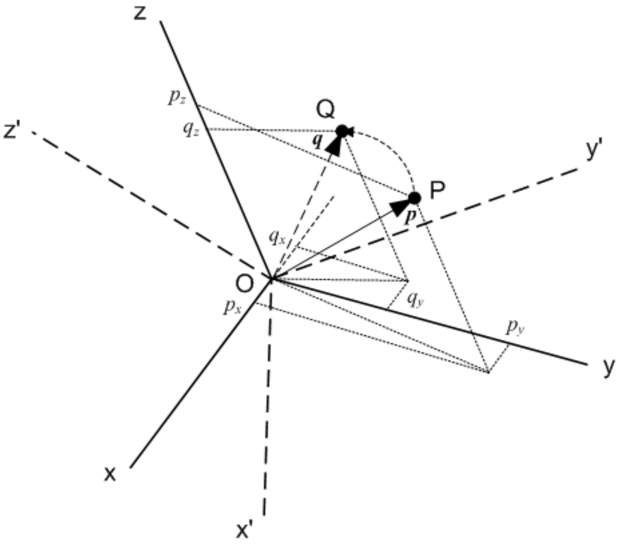


Figure 2.12: A point P is operated by the rotational operator, $Rot\,(R)$, of which its result is the newly rotated point Q around the origin. ([1], pp. 29)

Combining both relations, the relationship between the rotated vectors may be written as

$$\bar{q} = Rot\,(R)\,\bar{p} = R\bar{p}, \tag{2.13}$$

where R is the rotation matrix that rotates point P to point Q in the same direction as the initial frame $\{A\}$ is rotated to the current frame $\{B\}$. Since every quantity is described in the same frame, notations used for the frame in the operator equation may be dropped.

Similar to the translational case, comparing Eq. 2.13 to Eq. 2.8, the intimate roles of the mapping and the operator may again be understood. For the mapping, the resulting vector represents the same point in the new frame $\{A\}$ that is rotated by R^{-1}. For the operator, the resulting vector corresponds to the new point which occurs by the rotating action of R. Recapitulating, it may be said that the relative position may be obtained by the motion of the point, or by the opposite motion of the representing frame.

2.3.3 Transformation Operators

Lastly, let consider Fig. 2.13 which involves a frame and two points P and Q. Point Q is constructed from rotating the original point P about the origin by the rotational operator $Rot\,(R)$, then translating the intermediate point along the direction and distance dictated by the translational operator $Trans\,(\bar{o})$. Therefore, the two vectors describing these points are related by

$$\bar{q} = R\bar{p} + \bar{o}. \tag{2.14}$$

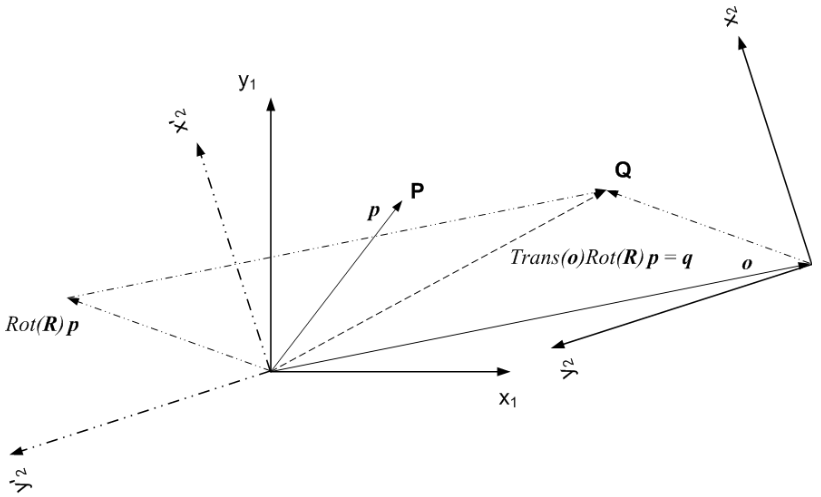


Figure 2.13: A point P is operated by the general transformation operator, T, of which its result is the newly rotated and translated point Q. Note that ${}^B\bar{q} = {}^A\bar{p}$.

Effectively, these successive operations may be combined using the homogeneous transformation matrix T as

$$\bar{Q} = T\bar{P}, \tag{2.15}$$

where the vectors are expressed in the homogeneous four dimensional space. Specifically,

$$\begin{bmatrix} \bar{q} \\ 1 \end{bmatrix} = \begin{bmatrix} R & \bar{o} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \bar{p} \\ 1 \end{bmatrix}.$$

Here, the transformation matrix T is viewed as the general operator which rotates and translates point P to point Q in the same manner as the initial frame $\{A\}$ is transformed to the current frame $\{B\}$. Note again that all quantities are described in the same frame.

Equation 2.10 and Eq. 2.15 are in fact the same equations interpreted differently. Namely for the mapping in Eq. 2.10, the resulting vector represents the same point in the new frame $\{A\}$ that is transformed by T^{-1}. For the operator, the resulting vector corresponds to the new point which is generated from the general motion operator T. In other words, the relative posture may be obtained by the motion of the point, or by the opposite motion of the representing frame.

Example 2.4 A point P is located by the position vector $\bar{p} = \begin{bmatrix} 3 & 7 & 0 \end{bmatrix}^T$. The point is rotated about the origin by 30° and then translated along the x- and y-axis by 10 and 5 units, in turn. Locate the position of the resulting point. Interpret this operation as the equivalent vector mapping between two frames.

Solution According to the problem statement, Fig. 2.14 shows the point P and the resulting point Q. Position of this point may be determined by the

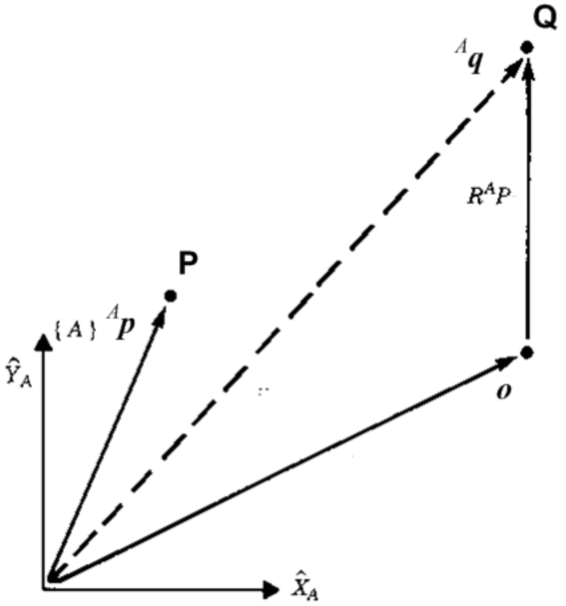


Figure 2.14: Example 2.4 (viewed as transformation operator).

direct measurement for this simple example or it may be carried out formally through the following homogeneous transformation matrix,

$$T = \begin{bmatrix} c30^\circ & -s30^\circ & 0 & 10 \\ s30^\circ & c30^\circ & 0 & 5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

which combines the explained rotational and translational operator. Hence, position of the transformed point Q may be calculated as

$$\bar{Q} = T\bar{P} = \begin{bmatrix} c30^\circ & -s30^\circ & 0 & 10 \\ s30^\circ & c30^\circ & 0 & 5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 7 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 9.098 \\ 12.562 \\ 0 \\ 1 \end{bmatrix}.$$

Figure 2.15 presents the above equation from the mapping point of view, i.e.

$$^{A}\bar{P} =^{A}_{B} T^{B}\bar{P} = \begin{bmatrix} c30^\circ & -s30^\circ & 0 & 10 \\ s30^\circ & c30^\circ & 0 & 5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 7 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 9.098 \\ 12.562 \\ 0 \\ 1 \end{bmatrix}.$$

Position vector of the point P expressed in the moving frame $\{B\}$ is mapped to the one represented in the fixed frame $\{A\}$ by the homogeneous mapping

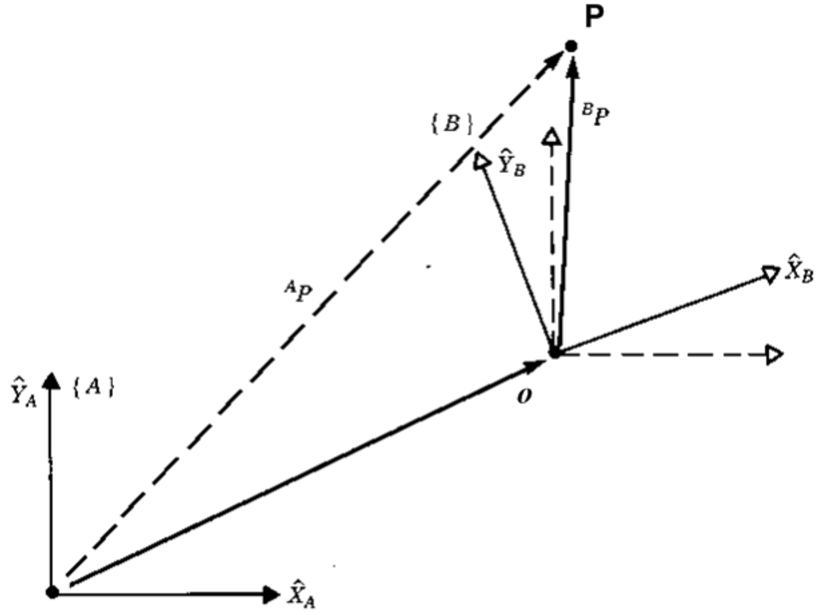


Figure 2.15: Example 2.4 (viewed as general mapping).

${}^A_BT$. This matrix, which is the same as the homogeneous transformation operator matrix moving point P to Q, is the inverse of the matrix used to transform $\{B\}$ to $\{A\}$.

2.4 Transformation Arithmetics and Equations

For the analysis of a manipulator, as will be seen in chapter 3, many frames are involved. As a consequence, there is a need to determine the result of mapping or transforming through several frames. Accordingly, there are two operations for the set of homogeneous transformation matrices: the multiplication and the inversion. In this section, it is shown how these operations are related to the physical mapping or transformation of frames.

2.4.1 Compound Transformation: Multiplication Operator

Mapping between two frames $\{A\}$ and $\{B\}$ can be generalized to the mapping through several frames naturally. Consider Fig. 2.16 where there are three frames: $\{A\}$, $\{B\}$, and $\{C\}$. Succesive transformations ${}^A_BT$ and ${}^B_CT$ are known. It is desired to map the vector $\bar{p}$ represented in $\{C\}$ to the one described in $\{A\}$. From section 2.2.3, ${}^C\bar{p}$ may be mapped to the intermediate frame as ${}^B\bar{p}$ by

$${}^B\bar{P} = {}^B_CT^C\bar{P}.$$

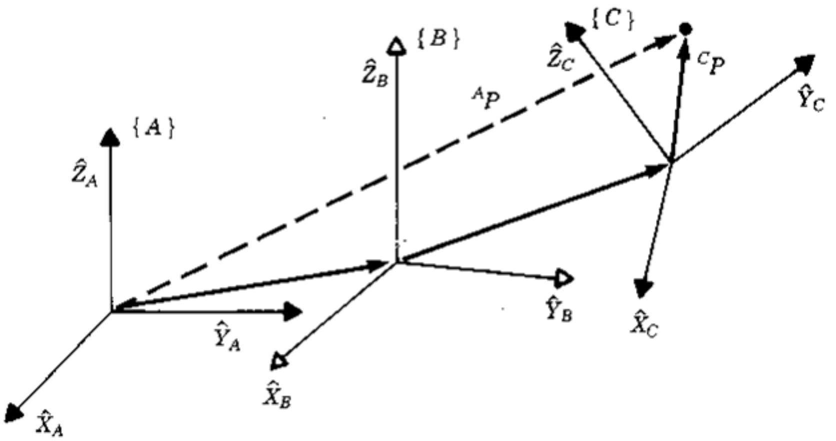


Figure 2.16: Several transformation matrix may be combined to yield the compound transformation. ([4], pp. 35)

Similarly, the mapped $^B\bar{p}$ may be furthered mapped to the fixed frame as $^A\bar{p}$ by

$$^A\bar{P} = {}^A_BT^B\bar{P}.$$

Combining the above mappings, the compound mapping may be formulated;

$$^A\bar{P} = {}^A_BT^B_CT^C\bar{P},$$

from which the following compound mapping,

$$^A_CT = {}^A_BT^B_CT, \tag{2.16}$$

is obvious. Cancellation of the subscript and the superscript of the intermediate frame $\{B\}$ acts as a good mnemonic of straightforwardly writing the compound mapping equation. Explicit multiplication of the detailed transformation matrices leads to the formula of ^A_CT;

$$^A_CT = \begin{bmatrix} ^A_BR^B_CR & ^A_BR^B\bar{o}_C + {}^A\bar{o}_B \\ 0 & 1 \end{bmatrix}. \tag{2.17}$$

Compound transformation or operator may be formulated readily by acknowledging it as just another interpretation of the mapping, as explained earlier.

2.4.2 Inversion Operator

As shown in Eq. 2.6, the inverse of the rotation matrix can be calculated simply by taking the transpose. This does not extend to the homogeneous transformation matrix. Nevertheless, one need not perform the general inverse matrix calculation due to the special structure of the matrix.

Physical meaning of the inverse operation of the transformation matrix is to exchange the role of two frames. In other words,

$$\left({}^{A}_{B}T\right)^{-1} = {}^{B}_{A}T.$$

From the definition of the homogeneous transformation matrix, ${}^{B}_{A}T$ may be written as

$${}^{B}_{A}T = \begin{bmatrix} {}^{B}_{A}R & {}^{B}\bar{o}_{A} \\ 0 & 1 \end{bmatrix}.$$

Since ${}^{B}\bar{o}_{A}$ is the negative of ${}^{A}\bar{o}_{B}$ but expressed with respect to the different frames, to establish this relationship, the mapping must be applied. Mathematically,

$${}^{B}\bar{o}_{A} = -{}^{B}_{A}R\,{}^{A}\bar{o}_{B} = -{}^{A}_{B}R^{T\,A}\bar{o}_{B}.$$

Substituting this relation into the transformation matrix, the inverse of ${}^{A}_{B}T$ may be computed as

$$\left({}^{A}_{B}T\right)^{-1} = \begin{bmatrix} {}^{A}_{B}R^{T} & -{}^{A}_{B}R^{T\,A}\bar{o}_{B} \\ 0 & 1 \end{bmatrix}. \qquad (2.18)$$

2.4.3 Transform Equations

A typical situation in the manipulator analysis is to determine the motion, or, more primitive, the transformation of the end effector with respect to some reference frame. This is achieved through the compound transformation along the intermediate frames conventionally attached to the robot linkages. More details are given in the next chapter. Other scenario might entail the transformation between the end effector and the manipulated object via the camera frame, the robot base frame, and the working table base frame. See Fig. 2.17.

Generally, one may formulate the transform equation from the successive frame transformation matrices by walking through the loop path of the frames. As an example, Fig. 2.18 displays the set of frames of which the transformation relative to their consecutive frames are assumed available first. The arrow joining the origin of two frames indicates their relative representation.

It should be noted that there are several variations to derive the transform equations. In Fig. 2.18, for example, one may choose $\{U\}$ and $\{D\}$ as the reference and the target frames respectively first. Then, the compound transformation between these two frames may be calculated either by

$${}^{U}_{D}T = {}^{U}_{A}T\,{}^{D}_{A}T^{-1},$$

or by

$${}^{U}_{D}T = {}^{U}_{B}T\,{}^{B}_{C}T\,{}^{D}_{C}T^{-1}.$$

Consequently, they may be equate to obtain the transform equation as

$${}^{U}_{A}T\,{}^{D}_{A}T^{-1} = {}^{U}_{B}T\,{}^{B}_{C}T\,{}^{D}_{C}T^{-1}. \qquad (2.19)$$

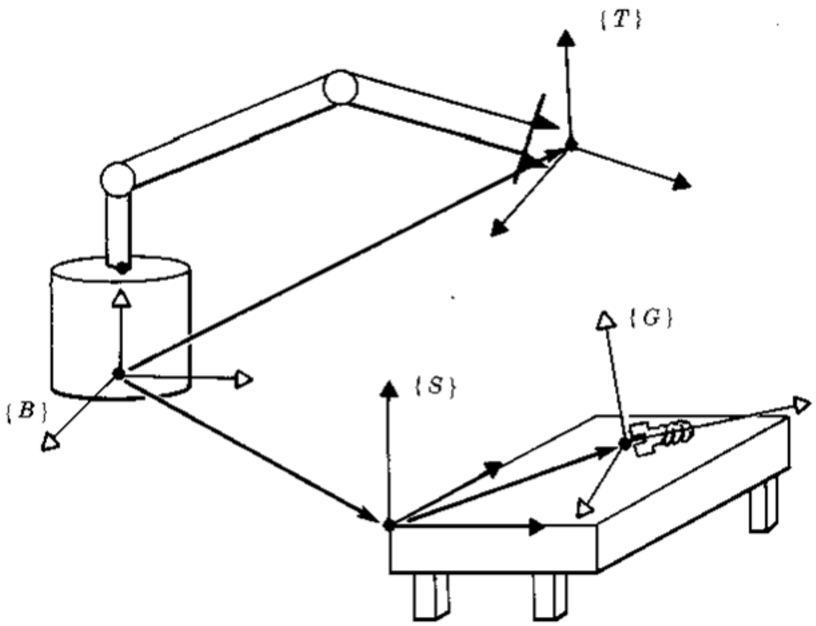


Figure 2.17: Transformation between the end effector and the manipulated object. ([4], pp. 39)

If the actual requirement is to determine the transformation ${}^D_CT$, corresponding to the relative posture of the object relative to the end effector frame, matrix manipulation may be performed on the above equation to yield

$$ {}^D_CT = {}^D_AT\,{}^U_AT^{-1}\,{}^U_BT\,{}^B_CT. $$

Example 2.5 Figure 2.19 depicts the robot grasping an object. Relative position and orientation of the base frame $\{B\}$, the working table frame $\{S\}$, the object frame $\{O\}$, the end effector frame $\{E\}$, and the camera frame $\{C\}$ are as shown in the figure. Determine the homogeneous transformation matrix ${}^B_ET$, ${}^S_OT$, ${}^C_OT$, ${}^C_ST$, and ${}^C_ET$. From these results, calculate ${}^B_OT$, ${}^E_ST$, and ${}^E_OT$. Compare the results with the direct observation.

Solution From Fig. 2.19, ${}^B_ET$ may be determined by first observing that $\{B\}$ is oriented relative to $\{S\}$ with the rotation of $-30°$ about ${}^{\{S\}}z$-axis. That is

$$ {}^S_BR = R_{z,-30°} = \begin{bmatrix} c\,(-30°) & -s\,(-30°) & 0 \\ s\,(-30°) & c\,(-30°) & 0 \\ 0 & 0 & 1 \end{bmatrix}. $$

Similarly, $\{E\}$ is oriented relative to $\{S\}$ with first the rotation of $-20°$ about ${}^{\{S\}}z$-axis, and then the rotation of 40° about the moving y-axis. The resulting

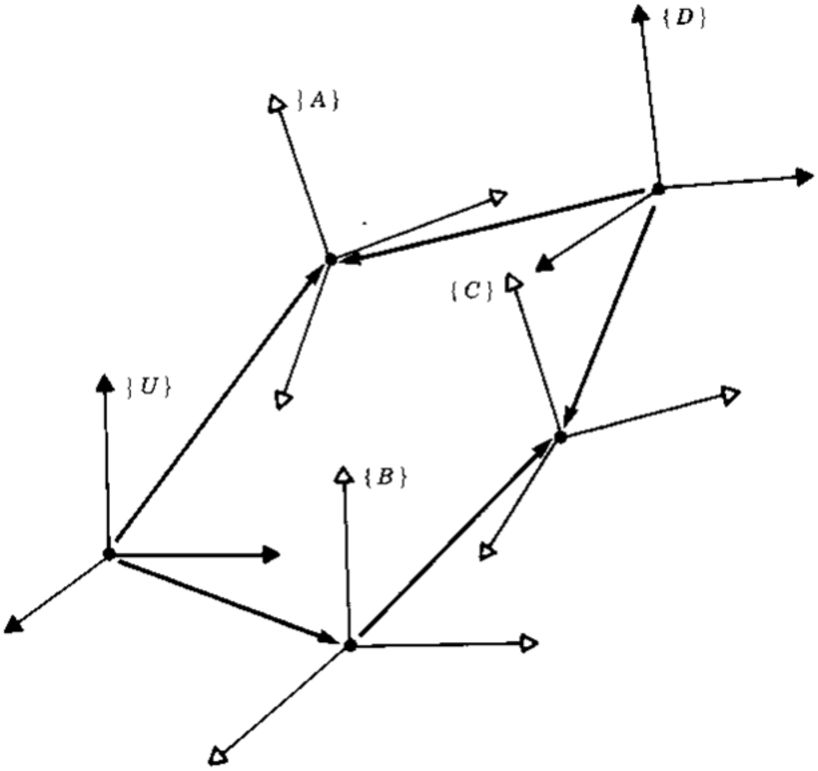


Figure 2.18: Frame diagram to formulate the frame transform equation. ([4], pp. 38)

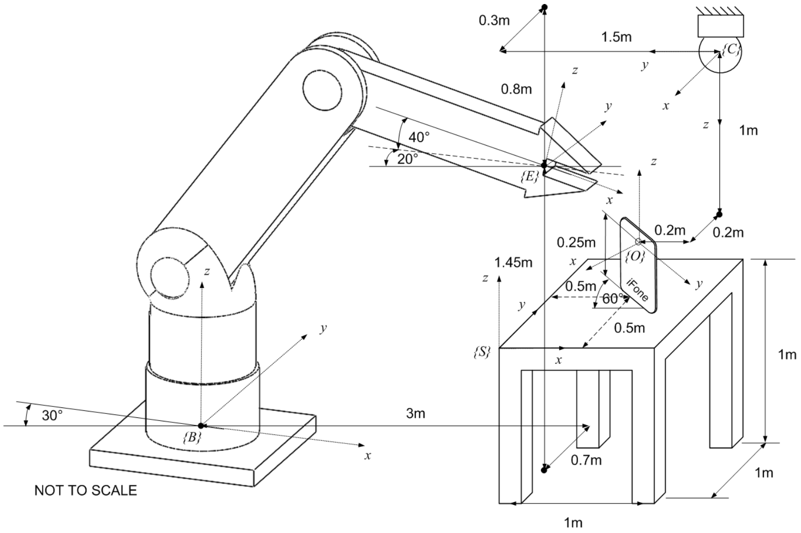


Figure 2.19: Example 2.5. ([1], pp. 64)

transformation would be

$$ {}^{S}_{E}R = R_{z,-20^\circ} R_{y,40^\circ} = \begin{bmatrix} c\,(-20^\circ) & -s\,(-20^\circ) & 0 \\ s\,(-20^\circ) & c\,(-20^\circ) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\,(40^\circ) & 0 & s\,(40^\circ) \\ 0 & 1 & 0 \\ -s\,(40^\circ) & 0 & c\,(40^\circ) \end{bmatrix}. $$

Consequently, the orientation of $\{E\}$ relative to $\{B\}$ may be obtained by the following compound transformation:

$$ {}^{B}_{E}R = {}^{S}_{B}R^{-1}\,{}^{S}_{E}R = R_{z,30^\circ} R_{z,-20^\circ} R_{y,40^\circ} = \begin{bmatrix} 0.7544 & -0.1736 & 0.6330 \\ 0.1330 & 0.9848 & 0.1116 \\ -0.6428 & 0 & 0.7660 \end{bmatrix}. $$

Moreover, the relative position of the origin of $\{E\}$ to $\{B\}$ is seen from Fig. 2.19 to be

$$ {}^{S}\bar{o}_{E/B} = \begin{bmatrix} 3 & -0.7 & 1.45 \end{bmatrix}^T, $$

which must further be represented in $\{B\}$ by

$$ {}^{B}\bar{o}_{E} = {}^{S}_{B}R^{-1}\,{}^{S}\bar{o}_{E/B} = \begin{bmatrix} 2.9481 & 0.8938 & 1.45 \end{bmatrix}^T. $$

Putting the orientation and position information together, the transformation matrix

$$ {}^{B}_{E}T = \begin{bmatrix} 0.7544 & -0.1736 & 0.6330 & 2.9481 \\ 0.1330 & 0.9848 & 0.1116 & 0.8938 \\ -0.6428 & 0 & 0.7660 & 1.45 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

is determined.

Next, posture of the object will now be determined relative to the table corner. Relative position of the origins is readily observed to be

$$ {}^{S}\bar{o}_{O} = \begin{bmatrix} 0.5 & 0.5 & 0.25 \end{bmatrix}^T. $$

Rotation about ${}^{\{S\}}z$-axis by -150° brings the orientation of $\{O\}$. As a result,

$$ {}^{S}_{O}T = \begin{bmatrix} R_{z,-150^\circ} & {}^{S}\bar{o}_{O} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} -0.8660 & 0.5 & 0 & 0.5 \\ -0.5 & -0.8660 & 0 & 0.5 \\ 0 & 0 & 1 & 0.25 \\ 0 & 0 & 0 & 1 \end{bmatrix}. $$

To determine ${}^{C}_{O}T$, first note that

$$ {}^{C}\bar{o}_{O} = \begin{bmatrix} 0.2 & 0.2 & 1 \end{bmatrix}^T. $$

Relative orientation may be determined by representing the unit vectors of the x-y-z-axes of $\{O\}$ in $\{C\}$. As a result,

$$ {}^{C}_{O}R = \begin{bmatrix} 0.5 & 0.8660 & 0 \\ 0.8660 & -0.5 & 0 \\ 0 & 0 & -1 \end{bmatrix}. $$

Therefore,

$$ {}^{C}_{O}T = \begin{bmatrix} 0.5 & 0.8660 & 0 & 0.2 \\ 0.8660 & -0.5 & 0 & 0.2 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}. $$

Similarly, ${}^{C}_{S}T$ may be observed directly from Fig. 2.19 as

$$ {}^{C}_{S}T = \begin{bmatrix} 0 & -1 & 0 & 0.7 \\ -1 & 0 & 0 & 0.7 \\ 0 & 0 & -1 & 1.25 \\ 0 & 0 & 0 & 1 \end{bmatrix}. $$

Position of the origin of $\{E\}$ with respect to $\{C\}$ is obvious from Fig. 2.19;

$$ {}^{C}\bar{o}_E = \begin{bmatrix} -0.3 & 1.5 & 0.8 \end{bmatrix}^T. $$

However, the relative orientation should be calculated indirectly via

$$ \begin{aligned} {}^{C}_{E}R &= {}^{C}_{S}R\,{}^{S}_{E}R \\ &= \begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} c\,(-20^\circ) & -s\,(-20^\circ) & 0 \\ s\,(-20^\circ) & c\,(-20^\circ) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\,(40^\circ) & 0 & s\,(40^\circ) \\ 0 & 1 & 0 \\ -s\,(40^\circ) & 0 & c\,(40^\circ) \end{bmatrix} \\ &= \begin{bmatrix} 0.2620 & -0.9397 & 0.2198 \\ -0.7198 & -0.3420 & -0.6040 \\ 0.6428 & 0 & -0.7660 \end{bmatrix}. \end{aligned} $$

Hence,

$$ {}^{C}_{E}T = \begin{bmatrix} 0.2620 & -0.9397 & 0.2198 & -0.3 \\ -0.7198 & -0.3420 & -0.6040 & 1.5 \\ 0.6428 & 0 & -0.7660 & 0.8 \\ 0 & 0 & 0 & 1 \end{bmatrix}. $$

The remaining transformation matrices may be obtained by formulating the transform equations and substituting the above results.

$$ \begin{aligned} {}^{B}_{O}T &= {}^{B}_{E}T\,{}^{E}_{C}T\,{}^{C}_{O}T = {}^{B}_{E}T\,{}^{C}_{E}T^{-1}\,{}^{C}_{O}T \\ &= \begin{bmatrix} -0.5 & 0.8660 & 0 & 4.3239 \\ -0.8660 & -0.5 & 0 & 1.1108 \\ 0 & 0 & 1 & 1.25 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \end{aligned} $$

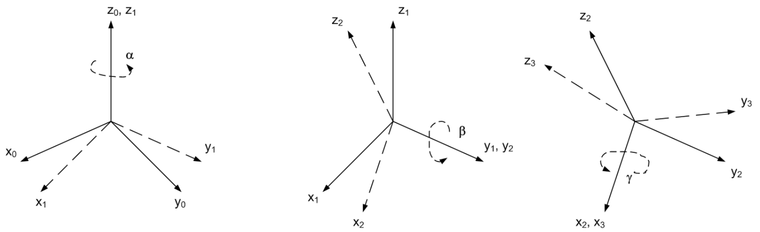


Figure 2.20: Orientation description by the ZYX Euler angles. ([1], pp. 34)

$$
\begin{aligned}
{}^{E}_{S}T &= {}^{E}_{C}T\,{}^{C}_{O}T\,{}^{O}_{S}T = {}^{C}_{E}T^{-1}\,{}^{C}_{O}T\,{}^{S}_{O}T^{-1} \\
&= \begin{bmatrix} 0.7198 & -0.2620 & -0.6428 & 1.1271 \\ 0.3420 & 0.9397 & 0 & -0.6661 \\ 0.6040 & -0.2198 & 0.7660 & 0.3583 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
\end{aligned}
$$

$$
\begin{aligned}
{}^{E}_{O}T &= {}^{E}_{C}T\,{}^{C}_{O}T = {}^{C}_{E}T^{-1}\,{}^{C}_{O}T \\
&= \begin{bmatrix} -0.4924 & 0.5868 & -0.6428 & 1.1954 \\ -0.7660 & -0.6428 & 0 & -0.0252 \\ -0.4132 & 0.4924 & 0.7660 & 0.7419 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
\end{aligned}
$$

2.5 Other Representation of the Orientation

Section 2.1.2 presented a rotation matrix to describe the three dimensional orientation. Due to the constraints of three mutually perpendicular unit vectors along the rectangular coordinate axes, there are totally six inherent constraints among nine elements of the rotation matrix. Consequently, only three parameters can describe arbitrary three dimensional orientation completely. There are several choices, in fact! Here, additional three different ways of describing the orientation shall be studied.

2.5.1 Euler Angles Representation

This method chooses three independent parameters for describing the orientation to be three consecutive angles of the basic rotations around the axes of the current, or moving, frames. Successive rotations must not occur about the same axis, however. Therefore, this representation must also be specified with the sequence of three axes about which the rotations occur: totally of $3\times2\times2 = 12$ possibilities. These three angles of the basic rotations about the axes of the moving frames are called *Euler angles*. Commonly used axes are the ZYX, the ZYZ, and ZXZ.

Let (α, β, γ) be the Euler angles around the ZYX axes as shown in Fig. 2.20. For this case, the orientation is constructed from three successive rotations as follow. Initially, the rotated frame coincides with $\{x_0\,y_0\,z_0\}$. First rotation occurs from the rotation about z_0-axis with the angle α, making the original frame rotated to $\{x_1\,y_1\,z_1\}$. Second rotation occurs from the rotation about y_1-axis of the current frame with the angle β that makes the frame rotated to $\{x_2\,y_2\,z_2\}$. The third, and the last, rotation is constructed from the rotation about x_2-axis of the current frame with the angle γ. This turns the frame to be coincident with $\{x_3\,y_3\,z_3\}$, eventually. See Fig. 2.20.

The equivalent rotation matrix of the ZYX Euler angles (α, β, γ) may be calculated by recognizing each sub-rotation as the description of the resulting frame relative to the previous one. Hence the equivalent rotation matrix is the description of $\{x_3\,y_3\,z_3\}$ relative to $\{x_0\,y_0\,z_0\}$ of which its detailed mappings are

$$\begin{aligned} {}^{\{x_0\,y_0\,z_0\}}_{\{x_3\,y_3\,z_3\}}R &= {}^{\{x_0\,y_0\,z_0\}}_{\{x_1\,y_1\,z_1\}}R^{\{x_1\,y_1\,z_1\}}_{\{x_2\,y_2\,z_2\}}R^{\{x_2\,y_2\,z_2\}}_{\{x_3\,y_3\,z_3\}}R \\ &= R_{z,\alpha}R_{y,\beta}R_{x,\gamma} \\ &= \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix}. \end{aligned} \tag{2.20}$$

Equation 2.20 determines the corresponding rotation matrix of the ZYX Euler angles (α, β, γ). The opposite problem is to determine the Euler angles out of the given rotation matrix. In particular, if a rotation matrix

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

describing an arbitrary rotation is provided, in the following the equivalent ZYX Euler angles (α, β, γ) will be calculated. First, assume that r_{11} and r_{21} are not zero at the same time. By comparing the given matrix elements with the expressions in Eq. 2.20, one can conclude that the common factor $c\beta \neq 0$ and the member r_{32} and r_{33} must not be zero simultaneously too. Moreover, since any row or column of the rotation matrix is a directional unit vector, $-s\beta = r_{31} \neq \pm 1$ and $c\beta = \pm\sqrt{1 - r_{31}^2} \neq 0$. Hence, the angle β has two possible values depending on whether $c\beta > 0$ or $c\beta < 0$. The remaining angles will also be different for each case.

If $c\beta > 0$, the Euler angles are determined as

$$\begin{aligned} \alpha &= \operatorname{atan2}(r_{21}, r_{11}) \\ \beta &= \operatorname{atan2}\left(-r_{31}, \sqrt{1 - r_{31}^2}\right) \\ \gamma &= \operatorname{atan2}(r_{32}, r_{33}). \end{aligned} \tag{2.21}$$

While if $c\beta < 0$, the Euler angles will be different:

$$\begin{aligned} \alpha &= \operatorname{atan2}\left(-r_{21}, -r_{11}\right) \\ \beta &= \operatorname{atan2}\left(-r_{31}, -\sqrt{1 - r_{31}^2}\right) \\ \gamma &= \operatorname{atan2}\left(-r_{32}, -r_{33}\right). \end{aligned} \tag{2.22}$$

Therefore, there are two ways in achieving the specified orientation around the particular sequence of rotation axes.

For the case when the values of both r_{11} and r_{21} be zero simultaneously, r_{32} and r_{33} will be zero as well. Imposing the constraint of unit vector for each row or column, it can be said that $r_{31} = -s\beta = \pm 1$. In this case, Eq. 2.21 and 2.22 cannot be used to calculate the Euler angles because the function atan2 $(\cdot)$ is not defined when both arguments are zero.

If $r_{31} = 1$, it implies $s\beta = -1$ and $c\beta = 0$. Therefore, Eq. 2.20 reduces to

$$R = \begin{bmatrix} 0 & -s_{\alpha+\gamma} & -c_{\alpha+\gamma} \\ 0 & c_{\alpha+\gamma} & -s_{\alpha+\gamma} \\ 1 & 0 & 0 \end{bmatrix}.$$

Consequently, the values of the Euler angles will become

$$\begin{aligned} \beta &= -\frac{\pi}{2} \\ \alpha + \gamma &= \operatorname{atan2}\left(-r_{12}, -r_{13}\right) = \operatorname{atan2}\left(-r_{12}, r_{22}\right). \end{aligned} \tag{2.23}$$

However, if $r_{31} = -1$, it implies $s\beta = 1$ and $c\beta = 0$. In this case, Eq. 2.20 reduces to

$$R = \begin{bmatrix} 0 & -s_{\alpha-\gamma} & c_{\alpha-\gamma} \\ 0 & c_{\alpha-\gamma} & s_{\alpha-\gamma} \\ 1 & 0 & 0 \end{bmatrix}.$$

Consequently, the values of the Euler angles will become

$$\begin{aligned} \beta &= \frac{\pi}{2} \\ \alpha - \gamma &= \operatorname{atan2}\left(-r_{12}, r_{13}\right) = \operatorname{atan2}\left(-r_{12}, r_{22}\right). \end{aligned} \tag{2.24}$$

For these latter cases, the values of α and γ cannot be deduced. In other words, there are infinitely many values of Euler angles that lead to this same orientation. This is called the *representational singularity of Euler angles.* It happens when the first and the third sub-rotation occur about the same physical axis of rotation. For the ZYX Euler angles, the representational singularity will happen when $\beta = \pm\frac{\pi}{2}$, for which the axis z_0 and x_2 will line up.

Example 2.6 An orientation is constructed from the following sub-rotations. Firstly, rotate about the z-axis by 90°. Then follow by the rotation about the y-axis of the current frame by $-180°$. Finally, rotate about the current x-axis by $-90°$. Describe this orientation with the ZYX Euler angles.

Solution The specified orientation occurs from the sub-rotations about the body-fixed frame. Hence the resultant rotation matrix may be calculated by post-multiplying them in order;

$$\begin{aligned} R &= R_{z,90^\circ} R_{y,-180^\circ} R_{x,-90^\circ} \\ &= \begin{bmatrix} c\,(90^\circ) & -s\,(90^\circ) & 0 \\ s\,(90^\circ) & c\,(90^\circ) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\,(-180^\circ) & 0 & s\,(-180^\circ) \\ 0 & 1 & 0 \\ -s\,(-180^\circ) & 0 & c\,(-180^\circ) \end{bmatrix} \\ &\quad \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\,(-90^\circ) & -s\,(-90^\circ) \\ 0 & s\,(-90^\circ) & c\,(-90^\circ) \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \end{aligned}$$

This rotation matrix can also be represented by the ZYX Euler angles where the values are determined by Eq. 2.21 and Eq. 2.22 as

$$\begin{aligned} \alpha &= \operatorname{atan2}(-1, 0) = -90^\circ \\ \beta &= \operatorname{atan2}(0, 1) = 0^\circ \\ \gamma &= \operatorname{atan2}(1, 0) = 90^\circ, \end{aligned}$$

or

$$\begin{aligned} \alpha &= \operatorname{atan2}(1, 0) = 90^\circ \\ \beta &= \operatorname{atan2}(0, -1) = 180^\circ \\ \gamma &= \operatorname{atan2}(-1, 0) = -90^\circ. \end{aligned}$$

This means that the specified orientation may also be constructed from the sub-rotations in the same order, but with different angles of $-90°$, 0°, and 90°.

2.5.2 Fixed Angles Representation

Three independent parameters for describing the orientation may be the three consecutive angles of the basic rotations about the axes of the fixed, or reference, frames. Again, successive rotations must not occur about the same axis, however. Similarly, this representation must also be specified with the sequence of three axes about which the rotations occur. Hence it has totally 12 possibilities. These three angles of the basic rotations about the axes of the fixed frame are called

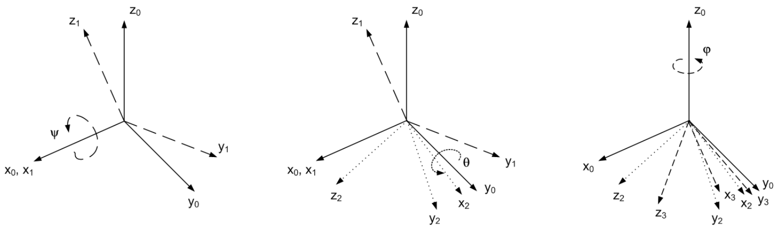


Figure 2.21: Orientation description by the XYZ fixed angles. ([1], pp. 38)

fixed angles. Commonly used axes are the XYZ and ZYX. As a historical note, fixed angles description of the orientation has its root from the roll-pitch-yaw angles used to represent the orientation of the vehicle in nautical and aeronautical science.

Let (ψ, θ, ϕ) be the fixed angles around the XYZ axes as shown in Fig. 2.21. For this case, the orientation is constructed from three successive rotations as follow. Initially, the rotated frame coincides with $\{x_0\, y_0\, z_0\}$. First rotation occurs from the rotation about x_0-axis with the angle ψ, making the original frame rotated to $\{x_1\, y_1\, z_1\}$. Second rotation occurs from the rotation about y_0-axis of the fixed frame by the angle θ that makes the frame rotated to $\{x_2\, y_2\, z_2\}$. The third, and the last, rotation is constructed from the rotation about z_0-axis of the fixed frame with the angle ϕ. This turns the frame to be coincident with $\{x_3\, y_3\, z_3\}$, eventually. See Fig. 2.21.

The equivalent rotation matrix of the XYZ fixed angles (ψ, θ, ϕ) may be calculated by realizing each sub-rotation as the operator which further rotates the resulting frame from the previous rotation. Hence the equivalent rotation matrix will be the operator that rotates $\{x_0\, y_0\, z_0\}$ to $\{x_3\, y_3\, z_3\}$ of which its detailed sub-rotations are

$$\begin{aligned} R &= R_{z,\phi} R_{y,\theta} R_{x,\psi} \\ &= \begin{bmatrix} c\phi c\theta & c\phi s\theta s\psi - s\phi c\psi & c\phi s\theta c\psi + s\phi s\psi \\ s\phi c\theta & s\phi s\theta s\psi + c\phi c\psi & s\phi s\theta c\psi - c\phi s\psi \\ -s\theta & c\theta s\psi & c\theta c\psi \end{bmatrix}. \end{aligned} \tag{2.25}$$

Equation 2.25 determines the corresponding rotation matrix of the XYZ fixed angles (ψ, θ, ϕ). The opposite problem is to determine the fixed angles out of the given rotation matrix. This can be done in a similar manner to the previous Euler angles description. Hence only the results shall be mentioned. The XYZ fixed angles description for the specified rotation matrix may be calculated as follow. For the case when the values of r_{11} and r_{21}, or r_{32} and r_{33}, are not zero simultaneously, there are two possible sets of the angles, i.e.

$$\begin{aligned}\psi &= \operatorname{atan2}(r_{32}, r_{33}) \\ \theta &= \operatorname{atan2}\left(-r_{31}, \sqrt{1-r_{31}^2}\right) \\ \phi &= \operatorname{atan2}(r_{21}, r_{11}),\end{aligned} \tag{2.26}$$

when $c\theta > 0$, and

$$\begin{aligned}\psi &= \operatorname{atan2}(-r_{32}, -r_{33}) \\ \theta &= \operatorname{atan2}\left(-r_{31}, -\sqrt{1-r_{31}^2}\right) \\ \phi &= \operatorname{atan2}(-r_{21}, -r_{11}),\end{aligned} \tag{2.27}$$

when $c\theta < 0$.

The following cases cause the *representational singularity of fixed angles*, which happen when the resulting rotation can be constructed by merely two, or one, sub-rotations. For the XYZ fixed angles, the singularity will happen when $\theta = \pm\frac{\pi}{2}$. The x-axis of the resultant frame will always direct along the z-axis of the reference frame. This indicates that the resulting rotation can be constructed from just two consecutive rotations, i.e. the first rotation about y_0-axis by $\pm\frac{\pi}{2}$ and follow by the appropriate rotation about z_0-axis.

When $\theta = \pm\frac{\pi}{2}$, $c\theta = 0$ and $s\theta = \pm 1$. For the case $r_{31} = 1$, there are infinitely many fixed angles which can describe this same orientation. Their values are as follow;

$$\begin{aligned}\theta &= -\frac{\pi}{2} \\ \phi + \psi &= \operatorname{atan2}(-r_{12}, -r_{13}) = \operatorname{atan2}(-r_{12}, r_{22}).\end{aligned} \tag{2.28}$$

If $r_{31} = -1$ instead, the angles are changed to

$$\begin{aligned}\theta &= \frac{\pi}{2} \\ \phi - \psi &= \operatorname{atan2}(-r_{12}, r_{13}) = \operatorname{atan2}(-r_{12}, r_{22}).\end{aligned} \tag{2.29}$$

Example 2.7 Describe the rotation in Ex. 2.6 with the XYZ fixed angles.

SOLUTION Referring to the rotation matrix

$$R = \begin{bmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

This rotation matrix may be represented by the XYZ fixed angles where the values are determined by Eq. 2.26 and Eq. 2.27 as

$$\begin{aligned}\psi &= \operatorname{atan2}(1, 0) = 90^\circ \\ \theta &= \operatorname{atan2}(0, 1) = 0^\circ \\ \phi &= \operatorname{atan2}(-1, 0) = -90^\circ,\end{aligned}$$

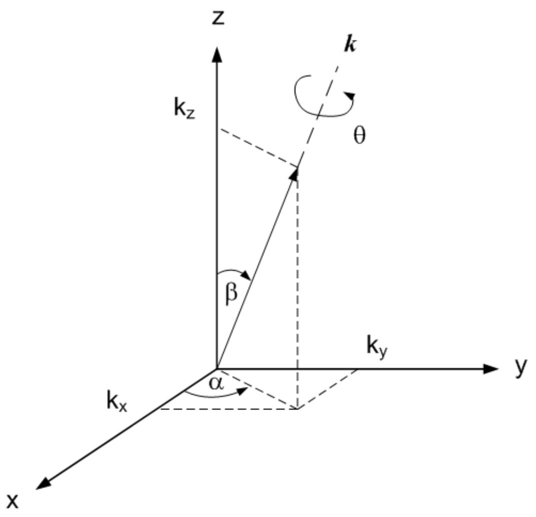


Figure 2.22: Orientation description by the angle-axis representation $\left(\theta,\hat{k}\right)$. ([1], pp. 41)

or

$$\begin{aligned}\psi &= \operatorname{atan2}(-1,0) = -90^\circ\\ \theta &= \operatorname{atan2}(0,-1) = 180^\circ\\ \phi &= \operatorname{atan2}(1,0) = 90^\circ.\end{aligned}$$

2.5.3 Angle-Axis Representation

Arbitrary orientation may be described in the most natural way by specifying the angle and the axis of rotation. This realization is supported by the Euler's theorm of rotation; a subordinate of the Chasles' theorem. The description is called the *angle-axis representation.*

Let $\hat{k} = [k_x\, k_y\, k_z]^T$ be a unit vector along the axis of rotation written in the reference frame $\{x\, y\, z\}$. Positive direction of $\hat{k}$ corresponds to the direction of positive rotation which obeys the right-hand rule. Let θ be the angle of the rotation about $\hat{k}$. Hence the representation is commonly written as the pair of the angle and the directional unit vector of the rotation: $\left(\theta,\hat{k}\right)$ or $\theta\hat{k}$.

Of course, the rotation by θ about $\hat{k}$ can be represented by the rotation matrix. Consider Fig. 2.22. Such rotation may be realized indirectly by first redirecting the rotation axis $\hat{k}$ to align with the z-axis. Then the rotation of θ about $\hat{k}$, which is now coincident with the z-axis, is executed. However, actually this must happen around the untouched $\hat{k}$. Therefore, after the rotation, the current $\hat{k}$ must be redirected back to the original direction. Combination of these three sub-rotations is equivalent to the rotation of θ about $\hat{k}$ directly. Mathematically,

$$R_{\hat{k},\theta} = {}^{\{xyz\}}_{\{x'y'\hat{k}\}}R \cdot R_{z,\theta} \cdot {}^{\{x'y'\hat{k}\}}_{\{xyz\}}R, \tag{2.30}$$

where the sub-rotations are referred to the reference frame and so the post-multiplication implies. Further, ${}^{\{xyz\}}_{\{x'y'\hat{k}\}}R$ may be computed from the post-multiplication of the sub-rotations about the y- and z-axis of the reference frame with the angle β and α;

$$ {}^{\{xyz\}}_{\{x'y'\hat{k}\}}R = R_{z,\alpha} \cdot R_{y,\beta}. \tag{2.31}$$

From Fig. 2.22, the trigonometric function of α and β may be written explicitly as

$$\begin{aligned} \sin\alpha &= \frac{k_y}{\sqrt{k_x^2 + k_y^2}} \\ \cos\alpha &= \frac{k_x}{\sqrt{k_x^2 + k_y^2}} \\ \sin\beta &= \sqrt{k_x^2 + k_y^2} \\ \cos\beta &= k_z. \end{aligned}$$

These expressions are substituted into Eq. 2.30 and 2.31 to obtain

$$R_{\hat{k},\theta} = \begin{bmatrix} k_x^2 v\theta + c\theta & k_x k_y v\theta - k_z s\theta & k_x k_z v\theta + k_y s\theta \\ k_x k_y v\theta + k_z s\theta & k_y^2 v\theta + c\theta & k_y k_z v\theta - k_x s\theta \\ k_x k_z v\theta - k_y s\theta & k_y k_z v\theta + k_x s\theta & k_z^2 v\theta + c\theta \end{bmatrix}. \tag{2.32}$$

The function $v\theta = \text{vers}\theta = 1 - c\theta$ is used to make the matrix more compact. Similar to other rotation descriptions, the inverse problem shall be analyzed. If the rotation matrix,

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

is specified, it is desirable to determine the angle and axis for which such rotation will yield the same orientation. By observing the matrix elements in Eq. 2.32, summing the diagonal elements will give the following equality;

$$v\theta + 3c\theta = r_{11} + r_{22} + r_{33},$$

where $\hat{k}$ is recognized as a unit vector. The angle of rotation θ hence may be calculated as

$$\theta = \cos^{-1}\left(\frac{r_{11} + r_{22} + r_{33} - 1}{2}\right). \tag{2.33}$$

There are two possible answers which are the negative of each other. The axis of rotation $\hat{k}$ can be retrieved by subtracting the appropriate off-diagonal elements. As a result,

$$\hat{k} = \frac{1}{2s\theta}\left[\begin{array}{c} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{array}\right]. \tag{2.34}$$

It can be verified that the axis of rotation associated to each angle is the negative of each other. Therefore the angle-axis representation for a particular rotation is not unique. Indeed, in general, there are two solutions: $\left(\theta, \hat{k}\right)$ and $\left(-\theta, -\hat{k}\right)$.

There are two possible rotations for this method of orientation description to fail. If $\theta = 0$ or 2π, where the rotation matrix becomes the identity matrix, determination of the axis of rotation from Eq. 2.34 will fail. This corresponds to the physics that the axis of rotation can be chosen to point in any direction since there is really no rotation!

Example 2.8 A rotation is generated by a rotation of 90° about z-axis, followed by a rotation of 30° about the current y-axis, and by the final rotation of −60° about the current x-axis. How such rotation be created by a single rotation.

Solution From the given orientation description, the rotation matrix may be calculated as

$$\begin{aligned} R &= R_{z,90^\circ} R_{y,30^\circ} R_{x,-60^\circ} \\ &= \left[\begin{array}{ccc} 0 & -\frac{1}{2} & -\frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{4} & \frac{1}{4} \\ -\frac{1}{2} & -\frac{3}{4} & \frac{\sqrt{3}}{4} \end{array}\right]. \end{aligned}$$

There are two possible direct rotations which yield the desired orientation. Using Eq. 2.33, the angle of rotation is

$$\theta = \cos^{-1}\left(-\frac{1}{2}\right) = \pm\frac{2\pi}{3}.$$

These angles have the corresponding axes of

$$\hat{k} = \frac{1}{2\sin\left(\pm\frac{2\pi}{3}\right)}\left[\begin{array}{c} -1 \\ \frac{1-\sqrt{3}}{2} \\ \frac{1+\sqrt{3}}{2} \end{array}\right] = \pm\frac{1}{\sqrt{3}}\left[\begin{array}{c} -1 \\ \frac{1-\sqrt{3}}{2} \\ \frac{1+\sqrt{3}}{2} \end{array}\right].$$

Therefore there are two possible rotations. Either the rotation about the axis $\frac{1}{\sqrt{3}}\left[-1\ \frac{1-\sqrt{3}}{2}\ \frac{1+\sqrt{3}}{2}\right]^T$ by the angle of 120°, or the rotation about the axis $-\frac{1}{\sqrt{3}}\left[-1\ \frac{1-\sqrt{3}}{2}\ \frac{1+\sqrt{3}}{2}\right]^T$ by the angle of −120° would work.

Problems

1. Vector $[3\ 4\ -2]^T$ is rotated about the x-axis of the reference frame, followed by the rotation about the z-axis of the current frame, and then the rotation about the y-axis of the reference frame. The angles of these successive rotations are 30°, $-50°$, and 100° respectively. Determine the resulting final vector.

2. ${}^{A}\bar{p} = [3\ 4]^T$ and ${}^{B}\bar{p} = [0.2730\ -4.9925]^T$ are the description of the free vector $\bar{p}$ in $\{A\}$ and $\{B\}$. Determine the relationship between these two frames.

3. Determine the equivalent ZYZ-fixed angles (ψ, θ, ϕ) of the rotation matrix
$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}.$$
Discuss the representational singularity problem in this case.

4. Transform the orientational representation of the ZYX-Euler angles $(30°, -50°, -80°)$ into the equivalent XYX-fixed angles representation.

5. A rotation is constructed from the following sub-rotations; initially the rotation occurs about the x-axis of the reference frame, then follows by the rotation about the z-axis of the current frame, the rotation about the x-axis of the reference frame, and ends with the rotation about y-axis of the reference frame. The corresponding angles executed are 20°, $-50°$, 200°, and $-80°$ respectively. Find the equivalent single rotation about the proper axis.

6. A letter 'A' is shown in Fig. 2.23 along with the coordinates of its edges. If the letter is rotated about the axis $\hat{k} = [-2\ 3\ -7]^T$ by the angle of 190°, write the letter at the new location. Does the letter has the same size and shape?

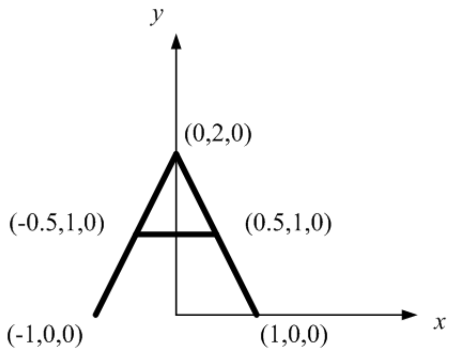


Figure 2.23: A letter ‘A’.

7. Consider Fig. 2.24 showing a robot set up 1 m from the table. The table dimension is 1 m square of its top and 1 m in height. It is affixed with $\{x_1\, y_1\, z_1\}$. A small cube, with which is assigned the frame $\{x_2\, y_2\, z_2\}$, is placed at the center of the table. A camera, located by $\{x_3\, y_3\, z_3\}$, is installed above the table corner opposite to $\{x_1\, y_1\, z_1\}$ by 2 m. Determine the transformations relating each of these frames to the base frame $\{x_0\, y_0\, z_0\}$. Determine the position of the box seen from the camera, by the transformation calculation and by direct observation.

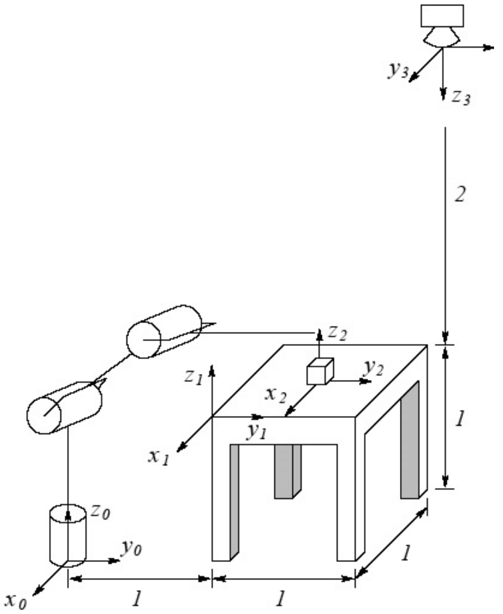


Figure 2.24: A robot grasping an object.

Chapter 3

Manipulator Kinematics

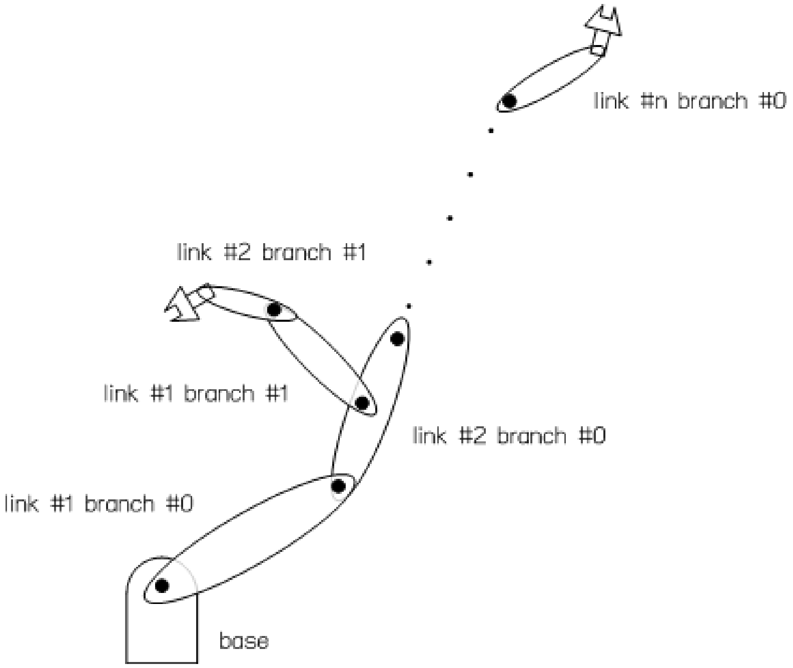


Figure 3.1: Conceptual schematic diagram of the open chain structured serial manipulator. ([1], pp. 8)

Method of describing the position and the orientation by the homogeneous transformation matrix in chapter 2 will be applied to analyze the kinematics of the serial manipulator. Specifically, the frames are attached to the linkages of the robot according to the Denavit-Hartenberg convention introduced in section 3.2. Successive transformation of them can then be performed so that the posture of the end-effector frame may be determined in terms of the robot joint parameters through the evaluation of the robot transformation equation in section 3.3. The last section explains a practical problem of mapping between the joint and the actuator variables, leading to the relationship between the actuator parameters and the robot posture.

3.1 Link Description

Serial robot or manipulator may be think of as the mechanism constructed from the links connecting together with the joints sequentially from the base to the end effector. As depicted in Fig. 3.1, serial robot employs the open chain(s) structure. Normally, the joints used in the robot are the simple joints such as the revolute or the prismatic joint. Also, the robot can be moved by the actuators which are often installed at the robot joints. Hence the robot posture is determined by the joint parameters. Consequently, in the following, the relationship of the robot posture, especially its end effector, in terms of the joint variables shall be studied:

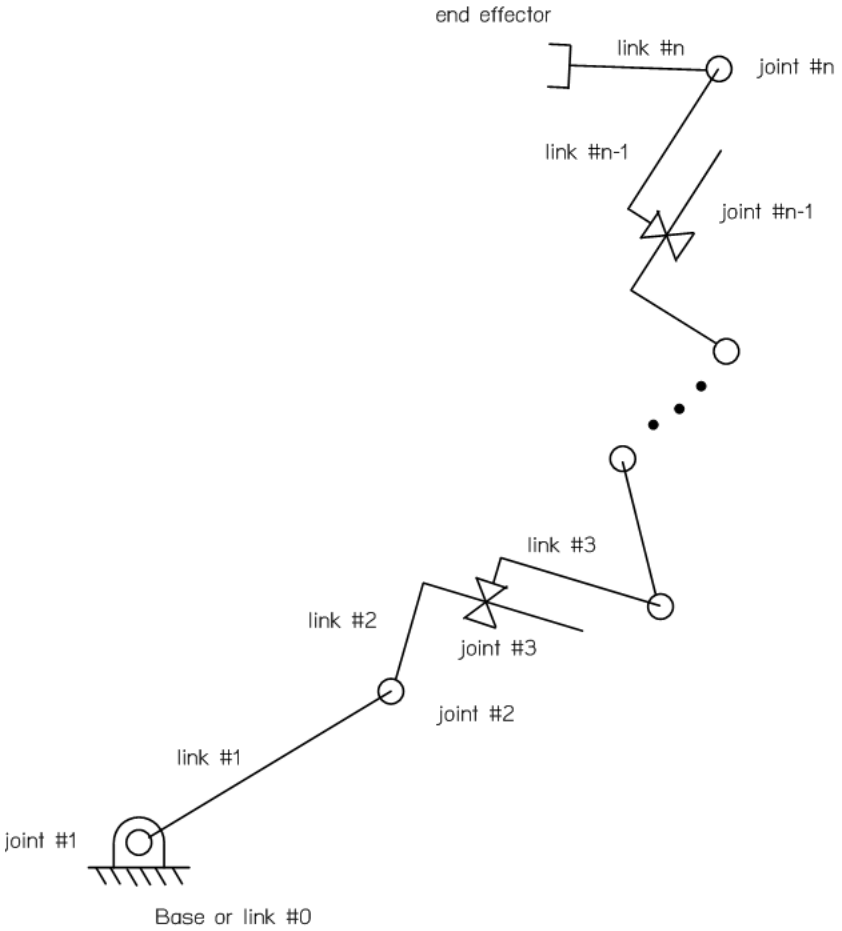


Figure 3.2: Open chain structure of the serial manipulator. ([1], pp. 68)

a basis to the robot motion. This is known as the *forward kinematics* analysis.

From the bird's eye view, if $\{E\}$ and $\{B\}$ are affixed to the end effector and the base of the robot, the forward kinematics problem is to determine the homogeneous transformation matrix ${}^{B}_{E}T$ as a function of the joint variables $(q_1, q_2, \ldots, q_n)$. The analysis may be carried out straightforwardly by observing the geometry of the robot at hand. A set of equations then shall be set up and some manipulation should be performed to deduce the elements of ${}^{B}_{E}T$ in terms of the joint variables.

Unfortunately, the geometric approach works well to the robots with simple structure only. Denavit and Hartenberg [7] recognized this problem in 1950's and proposed the formal method of the forward kinematics analysis by introducing the frames attached to the robot linkages. The procedure in setting up these frames follows what is known as the Denavit-Hartenberg (DH) convention. In this lecture note, a variation of the DH convention as used in [4] will be adopted. As a preparation to the forward kinematics analysis, notations for describing the robot linkages and joints will be explained first.

Figure 3.2 depicts a schematic diagram of the open chain structured serial robot which mechanically consists of

- LINKAGES of $n+1$ links counting from the robot base, as link #0, sequentially to the link #n where there exists the end effector.

- JOINTS of n joints connecting the linkages in the serial topology. The first joint, called joint #1, connects link #1 to the base. Counting outward to the end effector, totally there are n joints.

According to the above enumeration scheme, joint #i will join link #$(i-1)$ to link #i. Also, by the serial structure of the robot, actuation of joint #i causes the motion of link #i, #$(i+1)$, ..., #n. With these notations, it is ready to start analyzing the kinematics of the serial robot. By the repeating structure of each joint connecting two linkages, kinematic analysis of the robot can be performed recursively from the base to the end effector. This analysis is hence called the *forward kinematics* analysis, reflecting the progressive direction of the analysis procedure from the base to the end effector.

The analysis starts by attaching the frame $\{x_i\, y_i\, z_i\}$ onto the link #i. Note that no matter how the robot moves, any point on the linkage will always be described by the fixed coordinates in its moving frame. In addition to the frame $\{x_0\, y_0\, z_0\}$ fixated to the base, there might be a specific base frame or the fixed reference frame, usually denoted by $\{B\}$. Similarly, in addition to the frame $\{x_n\, y_n\, z_n\}$ attached to the last link #n where the end effector is installed, it is assigned with the the end effector frame $\{E\} = \{x_e\, y_e\, z_e\}$ next to $\{x_n\, y_n\, z_n\}$ to indicate its posture. Figure 3.3 depicts the robot in Fig. 3.2 equipped with these auxiliary frames.

According to the serial robot structure with the simple joints and these enumeration schemes, at any instant, the posture of $\{x_i\, y_i\, z_i\}$ relative to $\{x_{i-1}\, y_{i-1}\, z_{i-1}\}$ depends solely on the joint variable q_i. If the joint is of revolute type, the joint variable will be the rotated angle of the linkage θ_i. In case of the prismatic joint, it will be the displacement d_i. This is captured by the transformation ${}^{i-1}_{i}T\,(q_i)$.

As in subsection 2.4.3, the matrix ${}^{B}_{E}T$ which describes the posture of the end effector relative to the base may be determined by forming the compound transformation from the multiplication of the sub-transformations of the intermediate frames between $\{B\}$ and $\{E\}$;

$$ {}^{B}_{E}T = {}^{B}_{0}T\,{}^{0}_{1}T\,(q_1)\,{}^{1}_{2}T\,(q_2)\cdots{}^{n-1}_{n}T\,(q_n)\,{}^{n}_{E}T. \tag{3.1}$$

The equation presumes no relative motion between $\{n\}$ and $\{E\}$, i.e. both frames are affixed to the last link #n.

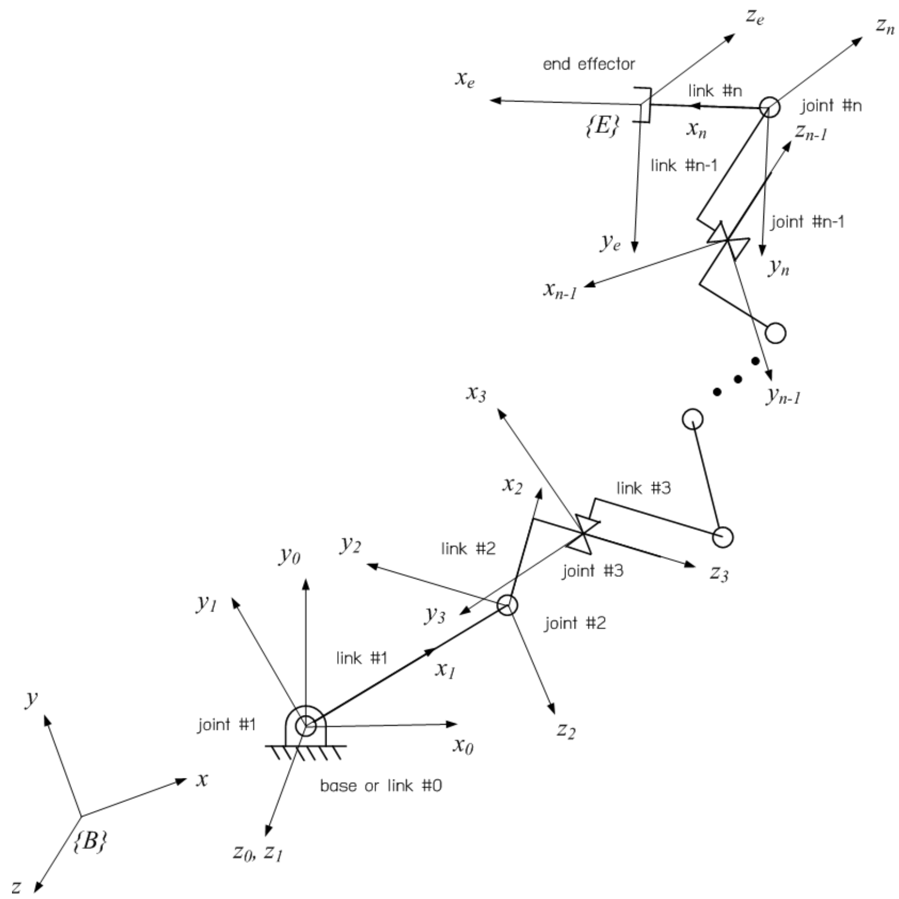


Figure 3.3: Serial manipulator and the associated frames for kinematical analysis. ([1], pp. 69)

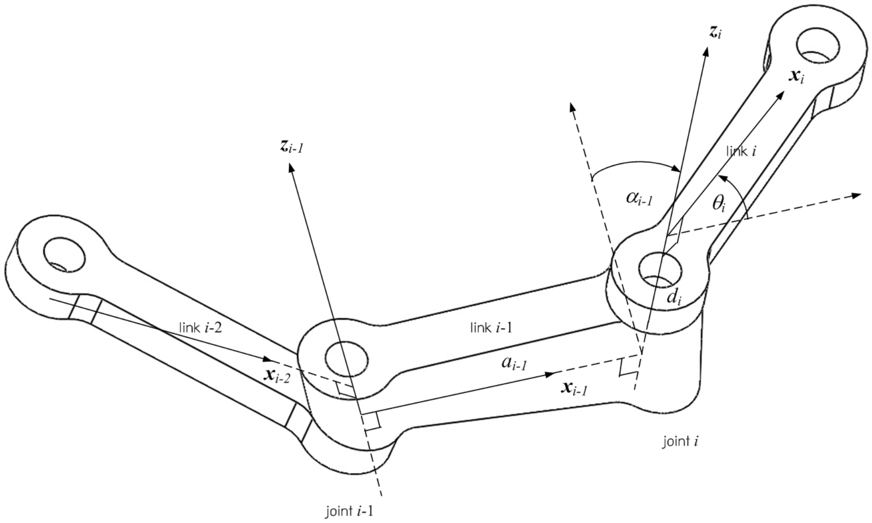


Figure 3.4: Frames and their parameters according to the DH convention. ([1], pp. 70)

3.2 Denavit-Hartenberg Convention

In general, link frames may be affixed to the robot linkages arbitrarily. Nevertheless, it is quite common to adopt the Denavit-Hartenberg (DH) convention to construct and attach the frames in a systematic manner. In this course, notations for the DH convention used in [4] are adopted.

The DH convention has two constraints in formulating the frames. They are

- The x-axis of a frame must be intersecting with the z-axis of the next frame.
- The x-axis of a frame must be perpendicular to the z-axis of the next frame.

As a result, the number of indepedent variables used to describe the transformation between two successive frames is reduced from six to four only.

In the following, the steps of attaching the frames to the serial robot with the simple joints only according to the DH convention will be explained. In case the robot possesses the compound joint, it will be decomposed as the serial connection of the simple joint first.

1. Enumerate the $(n+1)$-linkages starting from the link #0 for the base, successively counting to the link #n where the end effector is situated.
2. Enumerate the n-joints starting from the joint #1 which joins the link #1 to the base, successively counting toward the end effector joint. Joint #i will connect link #i to link #$(i-1)$.

3. Define the base frame $\{B\}$ at the base for the reference frame at any convenient location. However, $\{B\}$ should be the one that makes ${}^{B}_{0}T$ as simplest as possible.

4. Define the end effector frame $\{E\}$ at any convenient location on the end effector. However, $\{E\}$ should be the one that makes ${}^{n}_{E}T$ as simplest as possible.

5. Define the frame $\{i\}$ which moves along with link $\#i$. The frame is constructed in a way that z_i-axis coincides the joint axis $\#i$. The axis is oriented in the direction where the rotation or the translation is defined positive.

6. The x_i-axis is chosen to coincide with the common normal line between the z_i and z_{i+1}-axes. The x_i-axis is oriented in the direction from the z_i to z_{i+1}-axis.

7. In case when the z_i and z_{i+1}-axis intersect, the x_i-axis will be perpendicular to the plane formed by these two axes. Positive direction is selected such that it points toward the end effector joint.

8. In case when z_i and z_{i+1}-axis are parallel, the x_i-axis should be chosen such that it intersects with the x_{i-1}-axis. This would simplify ${}^{i-1}_{i}T$.

9. Direction of the x_n-axis can be chosen arbitrarily. However, if possible, it should be such that ${}^{n-1}_{n}T$ and ${}^{n}_{E}T$ be less complicated. For example, $\{n\}$ is parallel to $\{E\}$.

10. The origin of $\{i\}$ is the intersecting point of the x_i and z_i-axes.

11. Lastly, install $\{0\}$ affixed to the base. It should be chosen so that it coincides with $\{1\}$ at the robot home position

After finishing the frame installation to the robot, the DH parameters shall then be determined. They are used to describe the relative posture between the successive frames. To transform $\{i-1\}$ to $\{i\}$, the DH parameters involved are

- *Link length* a_{i-1} is the mathematical length of the link $\#(i-1)$. It is the distance from z_{i-1} to z_i along x_{i-1}-axis.

- *Link twist* α_{i-1} is the mathematical twist angle between the joint axes of the link $\#(i-1)$. It is the angle measured from z_{i-1} to z_i around x_{i-1}-axis.

- *Joint displacement* d_i is the distance measured from x_{i-1} to x_i along z_i-axis.

- *Joint angle* θ_i is the angle measured from x_{i-1} to x_i around z_i-axis.

Figure 3.4 displays the frames attached to the linkages and their relevant DH parameters. With the notations adopted in this course, ${}^{i-1}_{i}T$ will be a function of a_{i-1}, α_{i-1}, d_i, and θ_i. Accordingly, $\{i\}$ may be generated from $\{i-1\}$ in the following manner.

1. Translate $\{i-1\}$ along the x_{i-1}-axis by a_{i-1}. This corresponds to the operator

$$Tr_{x_{i-1},a_{i-1}} = \begin{bmatrix} 1 & 0 & 0 & a_{i-1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

2. Rotate the resulting frame around the (unchanged) x_{i-1}-axis by α_{i-1}. This corresponds to the operator

$$Rot_{x_{i-1},\alpha_{i-1}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha_{i-1} & -s\alpha_{i-1} & 0 \\ 0 & s\alpha_{i-1} & c\alpha_{i-1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

3. Translate the resulting frame along the (resulting) z_i-axis by d_i. This corresponds to the operator

$$Tr_{z_i,d_i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

4. Rotate the resulting frame around the (unchanged) z_i-axis by θ_i. This corresponds to the operator

$$Rot_{z_i,\theta_i} = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Because all the rotations and translations happen with respect to the current frame, the equivalent operator, which is the homogeneous transformation matrix describing the posture of $\{i\}$ relative to $\{i-1\}$, may be calculated by

$$\begin{aligned} {}^{i-1}_{i}T &= Tr_{x_{i-1},a_{i-1}} Rot_{x_{i-1},\alpha_{i-1}} Tr_{z_i,d_i} Rot_{z_i,\theta_i} \\ &= \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ c\alpha_{i-1}s\theta_i & c\alpha_{i-1}c\theta_i & -s\alpha_{i-1} & -d_i s\alpha_{i-1} \\ s\alpha_{i-1}s\theta_i & s\alpha_{i-1}c\theta_i & c\alpha_{i-1} & d_i c\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \end{aligned} \tag{3.2}$$

Table 3.1: Format of the table of DH parameters.

i	a_{i-1}	α_{i-1}	d_i	θ_i
0	a_b	α_b	d_b	θ_b
1	a_0	α_0	d_1	θ_1^*
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
i	a_{i-1}	α_{i-1}	d_i^*	θ_i
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
n	a_{n-1}	α_{n-1}	d_n	θ_n^*
E	a_e	α_e	d_e	θ_e

This transformation matrix can then be used to determine ${}^B_ET$ in Eq. 3.1.

For general robot analysis, it is advisable to create the table of DH parameters to assist the systematic calculation of the transformation matrices. As depicted in Table 3.1, the table has 5 columns indicating the order of the sub-transformation $\#i$ and the DH parameters a_{i-1}, α_{i-1}, d_i, θ_i. Row of the order $\#i$ contains the parameters used to transform $\{i-1\}$ to $\{i\}$. The first row will be the transformational parameters from $\{B\}$ to $\{0\}$, while the last row will be the ones from $\{n\}$ to $\{E\}$. Joint variables will be donoted by the star-mark, i.e. θ_i^* for the revolute and d_i^* for the prismatic joint. Totally, the table would have $(n+2)$ rows.

3.3 Manipulator Kinematics

In this section, the forward kinematics analysis of two robots are performed. The first example is simple while the second one is from the well known PUMA 560 industrial robot.

Example 3.1 *Articulated Arm*: Perform the forward kinematics analysis of the articulated robot depicted in Fig. 3.5.

Solution Following the guideline explained in section 3.2, the associated frames according to DH convention are attached to the robot as shown in Fig. 3.5. For the cases where there are options to define the frame, it will be chosen such that the frames' origin be coincident and the joint offset distance be zero. Specifically, table 3.2 is the table of DH parameters for the robot.

Note that most of the parameter's values are zero. Therefore the homogeneous transformation matrices will be simplified. Applying Eq. 3.2 to determine each sub-transformation, the results are as follow.

$$ {}^B_0T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & h \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad {}^0_1T = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

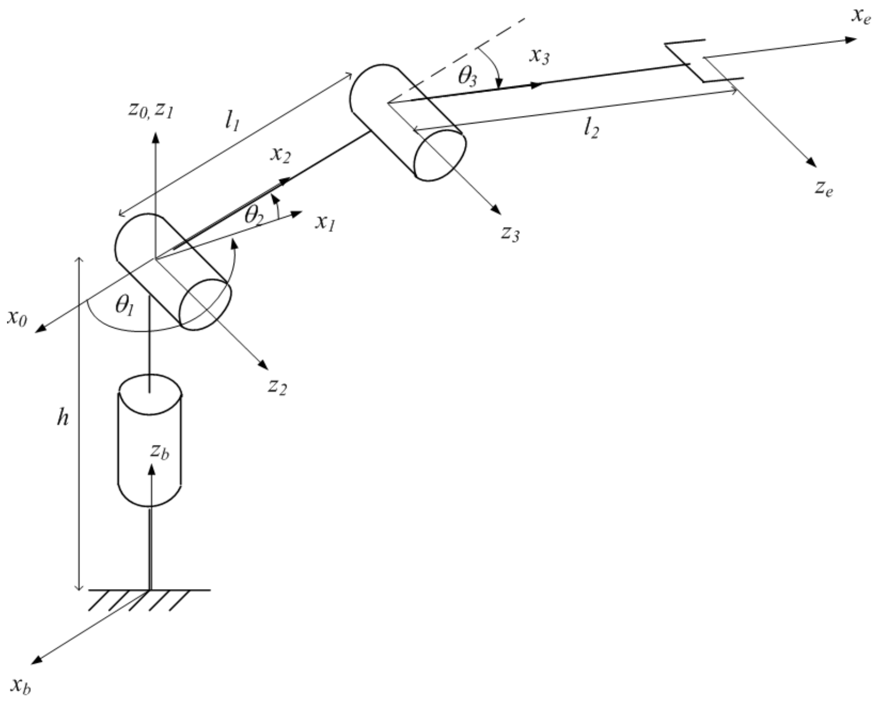


Figure 3.5: Example 3.1.

Table 3.2: Table of DH parameters for the articulated robot in example 3.1.

i	a_{i-1}	α_{i-1}	d_i	θ_i
0	0	0	h	0
1	0	0	0	θ_1^*
2	0	$\pi/2$	0	θ_2^*
3	l_1	0	0	θ_3^*
E	l_2	0	0	0

Table 3.3: Table of DH parameters for the PUMA 560 robot in example 3.2.

i	a_{i-1}	α_{i-1}	d_i	θ_i
0	0	0	H	0
1	0	0	0	θ_1^*
2	0	$-\pi/2$	0	θ_2^*
3	a_2	0	d_3	θ_3^*
4	a_3	$-\pi/2$	d_4	θ_4^*
5	0	$\pi/2$	0	θ_5^*
6	0	$-\pi/2$	0	θ_6^*
E	0	0	e	0

$$
{}^{1}_{2}T = \begin{bmatrix} c_2 & -s_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad {}^{2}_{3}T = \begin{bmatrix} c_3 & -s_3 & 0 & l_1 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{3}_{E}T = \begin{bmatrix} 1 & 0 & 0 & l_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
$$

Consequently, the homogeneous transformation matrix of $\{E\}$ with respect to $\{B\}$, representing the forward kinematics of the robot, may be determined by Eq. 3.1;

$$
\begin{aligned}
{}^{B}_{E}T &= {}^{B}_{0}T{}^{0}_{1}T{}^{1}_{2}T{}^{2}_{3}T{}^{3}_{E}T \\
&= \begin{bmatrix} c_1c_{23} & -c_1s_{23} & s_1 & c_1\left(l_1c_2 + l_2c_{23}\right) \\ s_1c_{23} & -s_1s_{23} & -c_1 & s_1\left(l_1c_2 + l_2c_{23}\right) \\ s_{23} & c_{23} & 0 & h + l_1s_2 + l_2s_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix}.
\end{aligned}
$$

Example 3.2 *PUMA 560 Robot*: Perform the forward kinematics analysis of the six degrees of freedom PUMA 560 robot depicted in Fig. 3.6. ([4], pp. 77)

SOLUTION Schematic diagram of the robot geometry is illustrated in Fig. 3.7 where the frames of the upper arm portion are attached according to the procedure outlined in section 3.2. In this figure, the robot is in the posture that makes all joint angles equal to zero. The frames of the forearm portion are depicted in Fig. 3.8. The base and the end effector frames are introduced for generalizing the result. According to the selected frames, table 3.3 contains the DH parameters of the robot.

Now, it is straightforward to apply Eq. 3.2 to determine each of the link

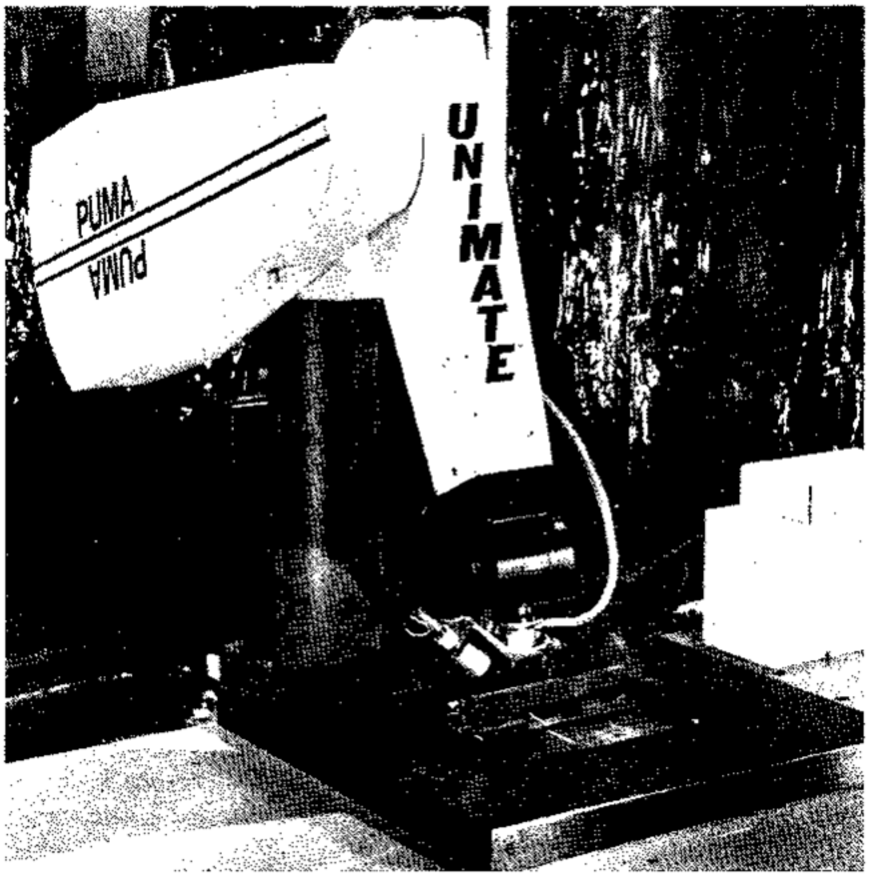


Figure 3.6: PUMA 560 by Unimation Incorporate. ([4], pp. 78)

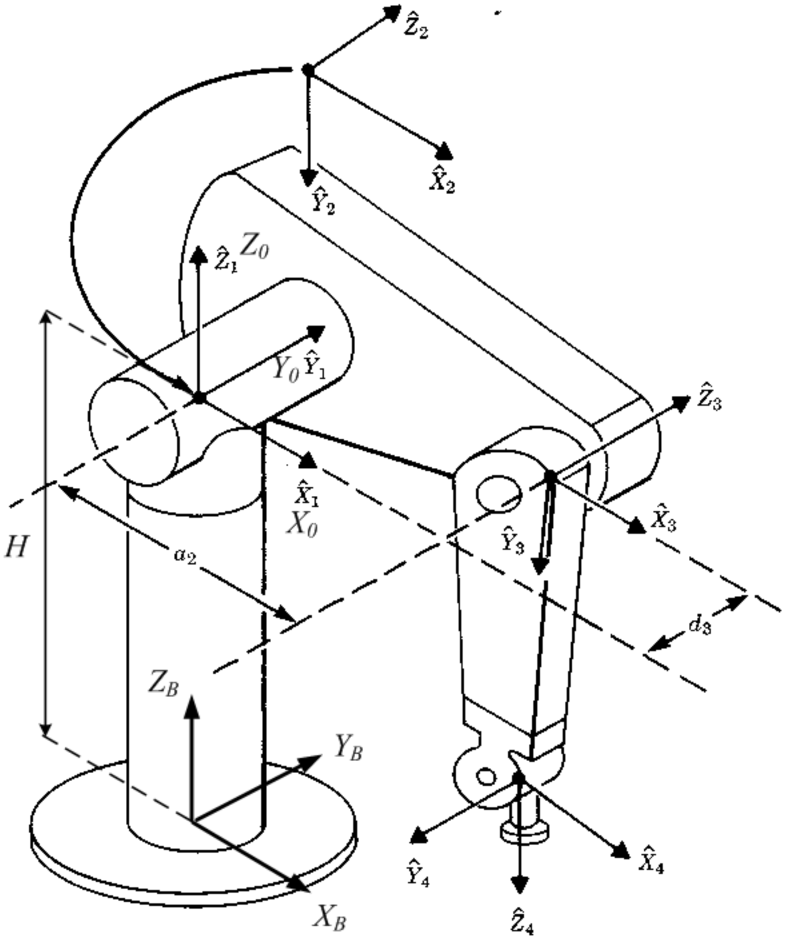


Figure 3.7: Frame assignments for the upper arm of the PUMA 560.

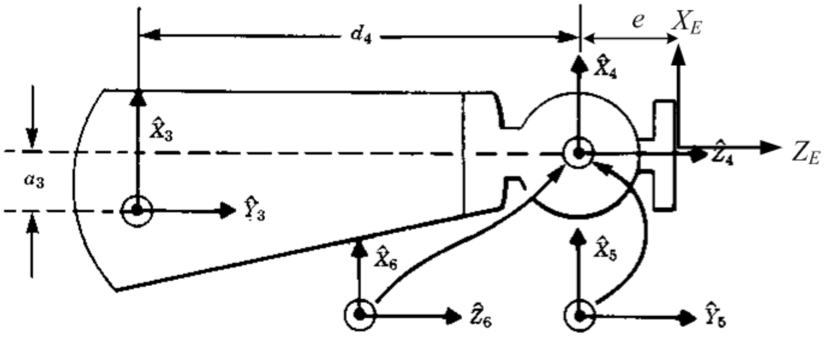


Figure 3.8: Frame assignments for the forearm of the PUMA 560.

transformation:

$$
{}^{B}_{0}T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & H \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad
{}^{0}_{1}T = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{1}_{2}T = \begin{bmatrix} c_2 & -s_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad
{}^{2}_{3}T = \begin{bmatrix} c_3 & -s_3 & 0 & a_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{3}_{4}T = \begin{bmatrix} c_4 & -s_4 & 0 & a_3 \\ 0 & 0 & 1 & d_4 \\ -s_4 & -c_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad
{}^{4}_{5}T = \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{5}_{6}T = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_6 & -c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad
{}^{6}_{E}T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & e \\ 0 & 0 & 0 & 1 \end{bmatrix}.
$$

Consequently, the homogeneous transformation matrix of $\{E\}$ with respect to $\{B\}$, representing the forward kinematics of the robot, may be determined by Eq. 3.1;

$$
\begin{aligned}
{}^{B}_{E}T &= {}^{B}_{0}T\,{}^{0}_{1}T\,{}^{1}_{2}T\,{}^{2}_{3}T\,{}^{3}_{4}T\,{}^{4}_{5}T\,{}^{5}_{6}T\,{}^{6}_{E}T \\
&= {}^{B}_{3}T\,{}^{3}_{4}T\,{}^{4}_{E}T \\
&= \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix},
\end{aligned}
$$

where

$$
{}^{B}_{3}T = \begin{bmatrix} c_1c_{23} & -c_1s_{23} & -s_1 & a_2c_1c_2 - d_3s_1 \\ s_1c_{23} & -s_1s_{23} & c_1 & a_2s_1c_2 + d_3c_1 \\ -s_{23} & -c_{23} & 0 & H - a_2s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{3}_{4}T = \begin{bmatrix} c_4 & -s_4 & 0 & a_3 \\ 0 & 0 & 1 & d_4 \\ -s_4 & -c_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad
{}^{4}_{E}T = \begin{bmatrix} c_5c_6 & -c_5s_6 & -s_5 & -es_5 \\ s_6 & c_6 & 0 & 0 \\ s_5c_6 & -s_5s_6 & c_5 & ec_5 \\ 0 & 0 & 0 & 1 \end{bmatrix},
$$

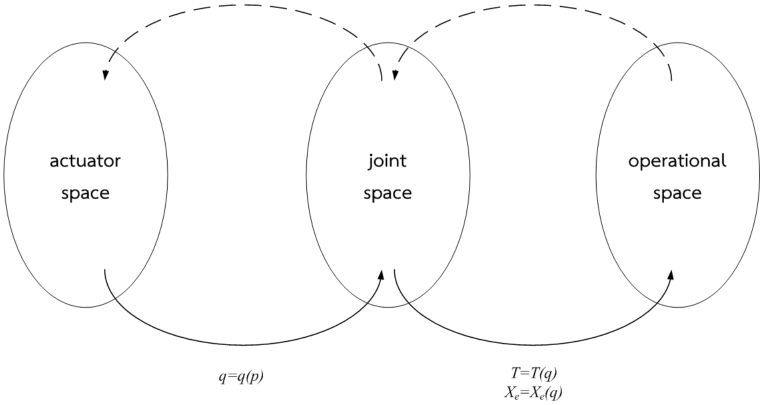


Figure 3.9: Mapping between three spaces. ([1], pp. 94)

and

$$\begin{aligned}
r_{11} &= c_1\left[c_{23}\left(c_4c_5c_6 - s_4s_6\right) - s_{23}s_5c_6\right] + s_1\left(s_4c_5c_6 + c_4s_6\right)\\
r_{21} &= s_1\left[c_{23}\left(c_4c_5c_6 - s_4s_6\right) - s_{23}s_5c_6\right] - c_1\left(s_4c_5c_6 + c_4s_6\right)\\
r_{31} &= -s_{23}\left(c_4c_5c_6 - s_4s_6\right) - c_{23}s_5c_6\\
r_{12} &= c_1\left[-c_{23}\left(c_4c_5s_6 + s_4c_6\right) + s_{23}s_5s_6\right] + s_1\left(c_4c_6 - s_4c_5s_6\right)\\
r_{22} &= s_1\left[-c_{23}\left(c_4c_5s_6 + s_4c_6\right) + s_{23}s_5s_6\right] - c_1\left(c_4c_6 - s_4c_5s_6\right)\\
r_{32} &= s_{23}\left(c_4c_5s_6 + s_4c_6\right) + c_{23}s_5s_6\\
r_{13} &= -c_1\left(c_{23}c_4s_5 + s_{23}c_5\right) - s_1s_4s_5\\
r_{23} &= -s_1\left(c_{23}c_4s_5 + s_{23}c_5\right) + c_1s_4s_5\\
r_{33} &= s_{23}c_4s_5 - c_{23}c_5\\
p_x &= c_1\left(a_2c_2 + a_3c_{23} - d_4s_{23}\right) - d_3s_1 - e\left[c_1\left(c_{23}c_4s_5 + s_{23}c_5\right) + s_1s_4s_5\right]\\
p_y &= s_1\left(a_2c_2 + a_3c_{23} - d_4s_{23}\right) + d_3c_1 - e\left[s_1\left(c_{23}c_4s_5 + s_{23}c_5\right) - c_1s_4s_5\right]\\
p_z &= H - a_2s_2 - a_3s_{23} - d_4c_{23} - e\left(-s_{23}c_4s_5 + c_{23}c_5\right).
\end{aligned}$$

3.4 Actuator Space, Joint Space, and Cartesian Space

Results from the forward kinematics analysis shows that the robot posture will be completely determined through the joint variables. For convenience, they are grouped as the *joint vector*:

$$\bar{q} = \left[\theta_1 \ \cdots \ \theta_i \ d_j \ \cdots \ \theta_n\right]^T. \tag{3.3}$$

The space of all possible joint vectors is called the *joint space*.

In practice, the joint varaibles are not regulated by the actuator variables, except for the direct drive robot. Usually, there must have intermediate mechanisms which drive the robot joints from the actuator motion. Likewise, the actuator variables may be written as the *actuator vector*

$$\bar{p} = \left[p_1 \ \cdots \ p_i \ \cdots \ p_m\right]^T, \tag{3.4}$$

and the space of all possible actuator vectors is called the *actuator space.*

Generally, tasks for the robot will determine the robot end effector motion: position and orientation. Position can be described readily via, e.g., the Cartesian coordinate system and the position vector. Rotation in three dimensional space, nevertheless, is more complicated since it is not the vector quantity. Methods in chapter 2 may be used to describe the end effector posture, such as the homogeneous transformation matrix

$$T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

or the 6-tuples of the Cartesian position vector and the Euler angles or the fixed angles for its orientation

$$\bar{X}_e = \left[p_x \, p_y \, p_z \, \psi \, \theta \, \phi\right]^T.$$

The space of all possible three dimensional positions and orientations is called the *operational space*, *task space*, or sometimes *Cartesian space.*

In the previous and this chapter, several methods of describing the posture of the robot end effector, which depends on the joint variables, are studied. Mathematically, they are viewed as the mapping from the joint space to the operational space;

$$T = T\left(\bar{q}\right) \tag{3.5}$$

or

$$\bar{X}_e = \bar{X}_e\left(\bar{q}\right). \tag{3.6}$$

This mapping falls into the nonlinear conformal mapping and normally is so complex that it may not be written as a usual function explicitly.

Additional mapping of the actuator space to the joint space

$$\bar{q} = \bar{q}\left(\bar{p}\right) \tag{3.7}$$

is necessary to calculate the joint motion from the actuators where the encoders are assembled to and the low level feedback control happens. Conceptual picture of the mapping between these three spaces, both forward and backward, is depicted in Fig. 3.9.

Problems

1. Analyze the forward kinematics of the planar robot arm illustrated in Fig. 3.10. The first and second joint type are of revolute and prismatic, respectively. Do the problem using the geometric approach and the formal frame setup approach under the Denavit-Hartenberg notation. Analyze for its workspace as well.

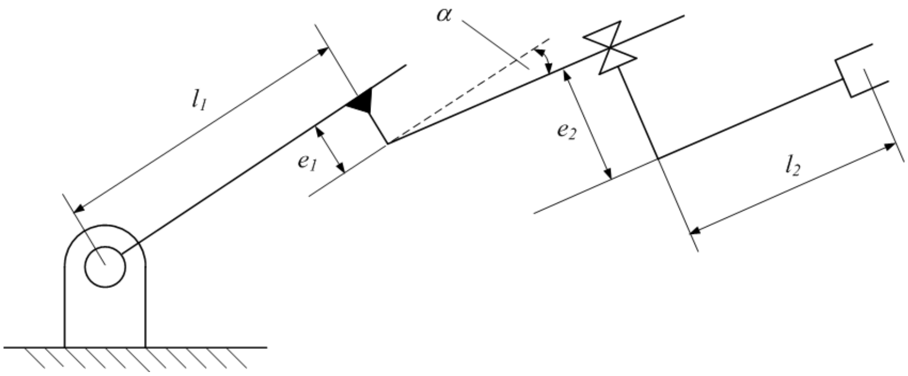


Figure 3.10: Schematic diagram of a planar revolute-prismatic (RP) joint robot arm.

2. Consider a 3R non-planar arm in Fig. 3.11. Set up the necessary frames and derive its forward kinematics and workspace.

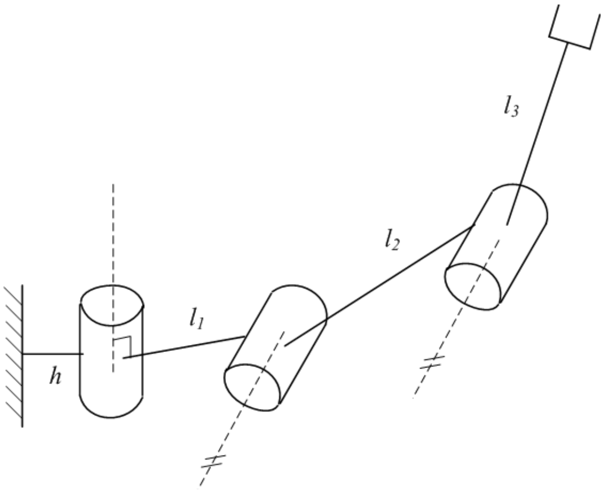


Figure 3.11: Schematic diagram of a 3R non-planar arm.

3. Figure 3.12 depicts the kinematic diagram of a 3-DOF non-orthogonal wrist. Assign the necessary frames to the mechanism and determine its forward kinematics and workspace.

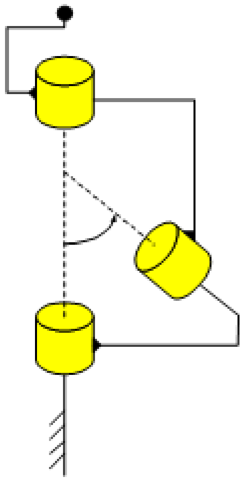


Figure 3.12: Schematic diagram of a 3-DOF non-orthogonal wrist.

4. Determine the forward kinematics of the SCARA robot as shown in Fig. 3.13. Analyze its workspace as well.

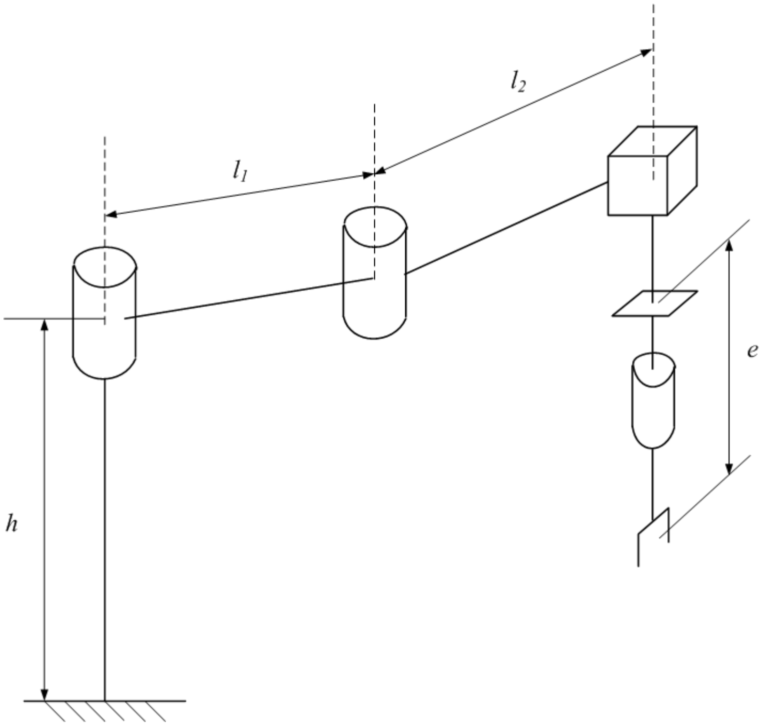


Figure 3.13: Schematic diagram of a SCARA robot arm.

Chapter 4

Inverse Manipulator Kinematics

Manipulator inverse kinematics problem is the converse problem of the forward kinematics in the previous chapter. Particularly, the inverse kinematics analyzes for the function of joint variables in terms of the given robot posture. For the serial manipulator, the inverse problem is more difficult, unfortunately due to the nonlinearity of the trigonometric functions which are not bijective. Hence, there is no book-keeping procedure for the inverse kinematics analysis in general.

In section 4.1, the issue of problem solvability will be addressed first. Then, two methods of the algebraic and the geometric approaches in solving the problem will be discussed in section 4.2. Inverse kinematics of the manipulators in chapter 3 will finally be considered.

4.1 Solvability

The essence of the inverse kinematics problem is to solve for the joint angles

$$\bar{q} = [\theta_1 \; \cdots \; \theta_i \; d_j \; \cdots \; \theta_n]^T$$

provided the robot posture is given. If this is specified by the homogeneous transformation matrix ${}^B_E T$, the inverse kinematics analysis may be viewed as the problem of solving a set of the following nonlinear equations:

$$\begin{array}{llll} r_{11} = r_{11}(\bar{q}) & r_{12} = r_{12}(\bar{q}) & r_{13} = r_{13}(\bar{q}) & p_x = p_x(\bar{q}) \\ r_{21} = r_{21}(\bar{q}) & r_{22} = r_{22}(\bar{q}) & r_{23} = r_{23}(\bar{q}) & p_y = p_y(\bar{q}) \\ r_{31} = r_{31}(\bar{q}) & r_{32} = r_{32}(\bar{q}) & r_{33} = r_{33}(\bar{q}) & p_z = p_z(\bar{q}) . \end{array} \tag{4.1}$$

Physically, twelve equations can be formulated. However, it is known that those nine elements for the rotation matrix are dependent. Indeed only three independent equations may be formulated for a particular orientation. Combined with the position vector, there will be totally merely six equations that may be used to solve for the joint angles. These equations heavily involve with the trigonometric functions which are nonlinear and not bijective, unfortunately. Therefore solving the inverse kinematics is a much more difficult problem and it typically does not have a book-keeping procedure to follow.

4.1.1 Existence of Solution

The first question before starting to look for the corresponding joint angles is that does such solution really exist. This is called the existence of the solution. The necessary and sufficient condition of the existence of the solution is the specified posture must physically be a posture in the robot workspace. For the solution $\bar{q}$ to always exist, the number of kinematically independent robot joints must not be less than the number of degrees of freedom (DOF) for specifying an

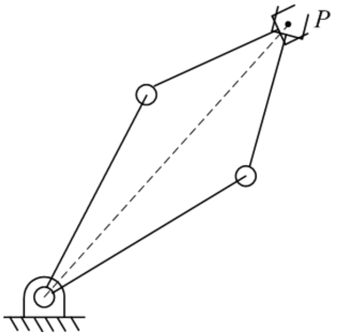


Figure 4.1: The desired end effector position $\bar{P}$ can be reached in a multiple ways. ([1], pp. 113)

arbitrary position and orientation in the robot workspace. If task is confined to be in a particular plane, the robot should possess at any instant at least three independent joints. For the robot to perform general task in three dimensional space, it should be designed in a manner that there are minimally six independent joints available in the workspace.

Moreover, the existence of solution depends on the joint limits and the obstacles as well. Solution of the joint angles from the inverse problem of $T(\bar{q})$ may not be realizable since it is out of the joint range or it causes the collision of the robot arm and the obstacle. In pracetice, it is thus necessary to check the inverse kinematics solution against these issues before the execution.

4.1.2 Multiple Solutions

If there exists joint angles $\bar{q}$ for a particular robot posture, a chance that there are multiple solutions is possible because the trigonometric functions are not the injective function. As an example, Fig. 4.1 depicts a two DOF planar robot. Its workspace is defined to be the set of all reachable end-effector positions. The figure illustrates two robot postures which correspond to the specified end effector position. This is in agreement with two solutions from the algebraic approach that will be explained in the next section.

Consider another example of a planar three DOF robot in Fig. 4.2. The workspace is defined as the set of all end effector positions and orientations. If the robot end effector position and orientation are specified, there will be two robot postures which satisfies the requirement. See the left figure of Fig. 4.2. However, if the end effector orientation causes no effect to the robot operation as if the workspace is the set of merely the end effector positions, then for a given robot end effector position, there will be infinitely many robot postures according to a bunch of possible orientation of the end effector by the third joint angle rotation.

The number of solutions for a specific posture depends on the number of

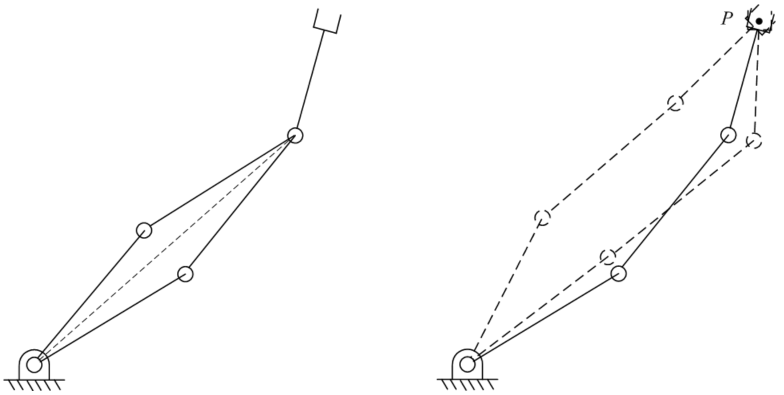


Figure 4.2: Inverse kinematics solution of a three DOF planar robot when the position and orientation (left) or only the position (right) of the end effector are specified. ([1], pp. 114)

independent robot joints. If the latter is greater than the number of required DOF for the task space, there will be infinitely many inverse kinematics solutions. On the other hand, if it is less than the number of required DOF, there might have no solution for a certain postures. Moreover, the number of solutions depends on the kinematical structure of the robot as well. Generally, the more the number of nonzero link length a_i or joint displacement d_i, the more the finite number of solutions will be. Similarly, the more the number of the joint twist α_i or the joint angle θ_i which are not 0° or 90°, the more the finite number of solutions will be. For a six DOF revolute joint robot, there are as many as 16 solutions of $\bar{q}$. This will be shown in analyzing the inverse kinematics of the PUMA 560.

4.2 Algebraic vs. Geometric Approaches

There is no general way to solve the arbitrary nonlinear algebraic equations except the numerical method. This method, nevertheless, has a drawback of providing a single solution which might not be the required one. Moreover, the method basically relies on recursive evaluation. Hence the time spent cannot be predicted. Worse yet, there is no guarantee whether the algorithm will converge to a solution. Consequently, the numerical method for solving the inverse kinematics is rarely used in real time control problem. However, it is often used in the simulation of the complex robot systems that are not possible to determine the closed form inverse kinematics solution.

Practically, the robot is often designed with a simple structure that possesses the closed form inverse kinematics solution. Solution of the joint vector $\bar{q}$ will be called the closed form solution if it can be written as the explicit function of the given parameters describing the robot posture. Unfortunately in many cases, the expression of $\bar{q}$ must be written using the implicit functions or with the composite functions to reduce the complexity of the expression. There are two main approaches in solving the inverse kinematics: the algebraic and the geometric approaches.

4.2.1 Algebraic Approach

This approach compares the specified position and orientation with the robot forward kinematics expressions and solves for $\bar{q}$. If the homogeneous transformation representation is employed, twelve equations may be formulated. However, as mentioned in section 4.1, there are only six independent equations which implies no more than six unknowns (possibly may not be the joint variables) are constrained. Nevertheless, after the appropriate set of equations are formed, the next step is to solve this system of nonlinear equations.

Any technique or methodology, especially the trigonometric identities, may be employed to solve the equations for the joint variables. Commonly, the equations are manipulated to eliminate several scalar variables so that only single variable is left in the equation. It may further be managed to write such variable as the explicit function of the specified posture. Issues pertaining to the existence of and mutiple solutions need to be addressed. They will bring about the conditions on the parameters.

Example 4.1 *Three DOF Planar Robot*: Perform the inverse kinematics analysis of a three DOF planar robot as shown in Fig. 4.3. Use the algebraic approach. ([1], Prob. 4.1)

Solution As a prerequisite, the forward kinematics must be analyzed first. Following the DH convention studied in chapter 3, the frames are set up and the homogeneous transformation matrix between the end effector and the base frame, $\{E\}$ and $\{0\}$, may be determined as

$$^{0}_{E}T = \begin{bmatrix} c_{123} & -s_{123} & 0 & l_1c_1 + l_2c_{12} + l_3c_{123} \\ s_{123} & c_{123} & 0 & l_1s_1 + l_2s_{12} + l_3s_{123} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Also, arbitrary posture of the robot end effector may be expressed by the general homogeneous transformation matrix:

$$H = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Corresponding joint angles θ_1, θ_2, and θ_3 may be determined from the fact that they equate $^{0}_{E}T$ to the specified H. Twelve algebraic equations can be formed.

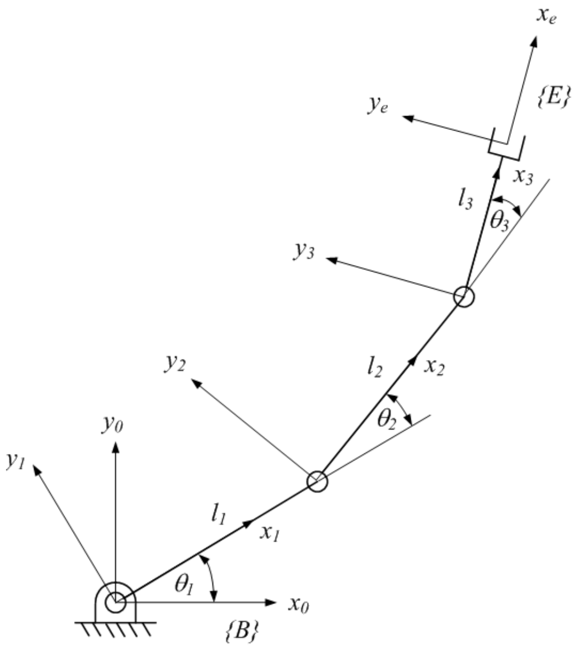


Figure 4.3: Example 4.1. ([1], pp. 75)

Among them, the explicit conditions are

$$r_{13} = 0 \qquad r_{23} = 0 \qquad r_{31} = 0 \qquad r_{32} = 0 \qquad r_{33} = 1 \qquad p_z = 0.$$

Therefore, if the given H does not obey to these requirements, the specified posture is out of the robot workspace and hence there is simply no solution.

The remaining equations are related to the joint variables. They are

$$\begin{aligned} r_{11} &= c_{123} \\ r_{12} &= -s_{123} \\ r_{21} &= s_{123} \\ r_{22} &= c_{123} \\ p_x &= l_1c_1 + l_2c_{12} + l_3c_{123} \\ p_y &= l_1s_1 + l_2s_{12} + l_3s_{123}. \end{aligned}$$

It may then be further concluded that the element $(1,1)$ and $(2,2)$ of H must be equal. Also the element $(2,1)$ must be the negative of the element $(1,2)$.

Replace c_{123} and s_{123} by r_{11} and r_{21}, in turn, in the expression of p_x and p_y, they may be rewritten as

$$\begin{aligned} p_x - l_3r_{11} &= l_1c_1 + l_2c_{12} \\ p_y - l_3r_{21} &= l_1s_1 + l_2s_{12}. \end{aligned}$$

Squaring each equation and summing them together lead to

$$(p_x - l_3 r_{11})^2 + (p_y - l_3 r_{21})^2 = l_1^2 + l_2^2 + 2l_1 l_2 c_2.$$

This equation has only one unknown of θ_2 which is inside the cosine function. Let us introduce

$$\begin{aligned} x &= p_x - l_3 r_{11} \\ y &= p_y - l_3 r_{21}, \end{aligned}$$

for conciseness of the development. Immediately, the cosine of θ_2 is simply

$$c_2 = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1 l_2}.$$

Since range of the cosine function is the closed interval $[-1, 1]$, the condition for the existence of the solution is

$$(l_1 - l_2)^2 \leq x^2 + y^2 \leq (l_1 + l_2)^2 .$$

The equation implies the specified posture of the end effector must induce the distance from the first to the third joint that is no less than the difference of the length of the first two links. Furthermore, such distance must not be greater than the sum of the length of the first two links as well. If the specified distance is out of the range, it simply cannot be reached.

If this condition is satisfied, the sine of θ_2 will be

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

Hence there are two possible solutions of θ_2 and they are

$$\theta_2 = \operatorname{atan2}(s_2, c_2) .$$

θ_1 may thus be determined by substituting the already known θ_2 back into the equations of p_x and p_y. That is, for each value of θ_2,

$$\begin{aligned} x &= (l_1 + l_2 c_2)\, c_1 - l_2 s_2 s_1 \\ y &= l_2 s_2 c_1 + (l_1 + l_2 c_2)\, s_1. \end{aligned}$$

After some manipulation, θ_1 may be solved;

$$\theta_1 = \operatorname{atan2}(y, x) - \operatorname{atan2}(l_2 s_2, l_1 + l_2 c_2) .$$

Lastly, for each set of θ_1 and θ_2, the corresponding θ_3 may be determined simply from

$$\theta_3 = \operatorname{atan2}(r_{21}, r_{11}) - \theta_1 - \theta_2.$$

In conclusion, there are two sets of the joint angles of the three DOF planar robot corresponding to a specified reachable posture.

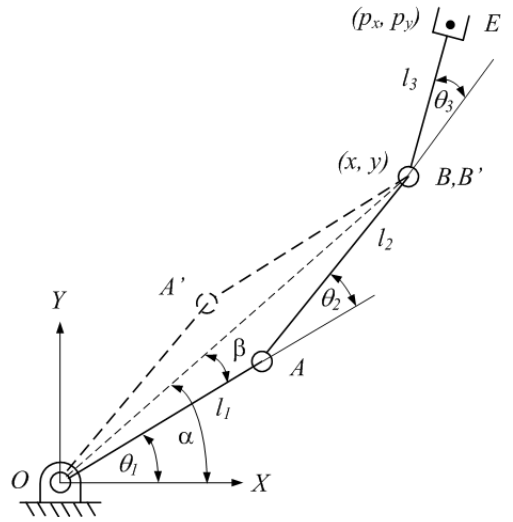


Figure 4.4: Example 4.2. ([1], pp. 119)

4.2.2 Geometric Approach

The geometric approach, on the contrary, will develop the kinematical equations from the structural geometric relationship of the robot. Usually, the relationship is derived by noticing the sum of the vectors alongside of a polygon must yield a null vector. The polygon may not be confined to lie on a plane, however. One may also set up the equations by projecting the polygon onto a plane which is orthogonal to the joint axis that defines the unknown joint variable. Particularly, if the joint variable θ_i is to be solved for, it might be useful to project the robot links onto the x_i-y_i plane first. Geometrical analysis of the projected polygon in the plane will then be performed.

Example 4.2 *Three DOF Planar Robot*: Perform the inverse kinematics analysis of a three DOF planar robot as shown in Fig. 4.4. Now use the geometric approach instead. ([1], Prob. 4.2)

SOLUTION Figure 4.4 displays the geometry of the robot pertinent to the following analysis. From the figure, if a straight line connecting the first and the third joint is drawn, a geometric relationship can be determined from the triangle OAB. Note that only the joint variable θ_2 is involved and thus the equation may be used to determine its value directly.

$\overrightarrow{OB}$ which represents a side of the triangle OAB may be determined by subtracting the vector associated with the third link from the position vector of the end effector. That is

$$\overrightarrow{OB} = \overrightarrow{OE} - \overrightarrow{BE}.$$

From the geometry in Fig. 4.4, the vector equation may be represented in $\{XY\}$

as

$$\overrightarrow{OB} = \begin{bmatrix} p_x \\ p_y \end{bmatrix} - \begin{bmatrix} l_3c_{123} \\ l_3s_{123} \end{bmatrix}.$$

Since the rotation matrix of the end effector frame must represent the rotation in the XY plane in accordance with the robot geometry, the elements of the specified homogeneous transformation matrix must conform to the conditions:

$$r_{13} = 0 \qquad r_{23} = 0 \qquad r_{31} = 0 \qquad r_{32} = 0 \qquad r_{33} = 1 \qquad p_z = 0.$$

Moreover, elements $(1,1)$, $(2,1)$, $(1,2)$, and $(2,2)$ must be equal to the cosine, sine, minus sine, and cosine of the rotated angle of the end effector. Because the rotated angle is obviously $\theta_1+\theta_2+\theta_3$, $\overrightarrow{OB}$ may be expressed in terms of the given position and orientation as

$$\overrightarrow{OB} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} p_x - l_3r_{11} \\ p_y - l_3r_{21} \end{bmatrix},$$

where, physically, x and y represent the coordinates of the third joint in the base frame.

Law of cosine may be applied to the triangle OAB for setting up the relation in determining θ_2 as

$$x^2 + y^2 = l_1^2 + l_2^2 - 2l_1l_2\cos(\pi - \theta_2) = l_1^2 + l_2^2 + 2l_1l_2c_2.$$

Hence

$$c_2 = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1l_2},$$

which corresponds to the result using the algebraic approach. In the same manner, numerical value of the right hand side term must lie in the closed interval $[-1,\ 1]$, leading to the condition for the existence of the solution;

$$(l_1 - l_2)^2 \leq x^2 + y^2 \leq (l_1 + l_2)^2.$$

Physical interpretation of this inequality may be understood from the underlying geometry in Fig. 4.4. θ_2 is solvable if and only if the distance from the first to the third joint determined from the specified end effector posture must not be longer than $(l_1 + l_2)$, which would be equal when the robot fully stretches out its arm so the first and the second links line up. Additionally, the distance must not be shorter than $|l_1 - l_2|$, which would be equal when the robot fully folds its arm back.

If the above condition is satisfied, θ_2 may be determined from the inverse of the cosine function

$$\theta_2 = \text{acos}\left(\frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1l_2}\right),$$

for which the value of θ_2 is limited to the interval $[0,\ \pi]$ corresponding to the geometry of the triangle. Another solution of θ_2 is for the case when the first and the second link are in the posture shown by the dotted lines in Fig. 4.4. Therefore the second solution may be readily determined since the triangle OAB and $OA'B'$ are the mirror images of each other about the line $\overline{OB}$:

$$\theta_{2'} = -\theta_2.$$

Referring to Fig. 4.4, if the first link is thought of as a human upper arm, the second link the lower arm, and the second joint as the elbow joint, two solutions of the second joint angles will make the arm be in the *elbow down* pose for θ_2 and the *elbow up* pose for $\theta_{2'}$.

θ_1 will be the difference of the angles α and β associated with θ_2 for the triangle OAB. However it will be the sum of α and β associated with $\theta_{2'}$ for the mirrored triangle $OA'B'$. From Fig. 4.4, angle α is readily determined as

$$\alpha = \mathrm{atan2}\left(y, x\right).$$

Angle β is an angle of the triangle OAB and hence may be determined from the cosine law. That is,

$$\beta = \mathrm{acos}\left(\frac{x^2 + y^2 + l_1^2 - l_2^2}{2l_1\sqrt{x^2 + y^2}}\right)$$

for which the value is limited to $[0,\ \pi]$ by the physical geometry of the triangle. Hence,

$$\theta_1 = \begin{cases} \alpha - \beta, & \theta_2 \geq 0 \\ \alpha + \beta, & \theta_2 < 0 \end{cases}$$

Lastly, from Fig. 4.4, since the angle the end effector oriented in plane ϕ is equal to $\theta_1 + \theta_2 + \theta_3$, and from the constraint of the workspace onto the valid homogeneous transformation matrix as explained above, θ_3 may thus be determined as

$$\theta_3 = \phi - \theta_1 - \theta_2 = \mathrm{atan2}\left(r_{21}, r_{11}\right) - \theta_1 - \theta_2.$$

Note that although the expression of θ_1 obtained from the geometric and the algebraic approach are different, they yield the same angle as can be viewed from different triangles.

4.3 Examples

This section provides more examples of analyzing the inverse kinematics of two robots in the previous chapter. This is the continuation of their forward kinematics analysis.

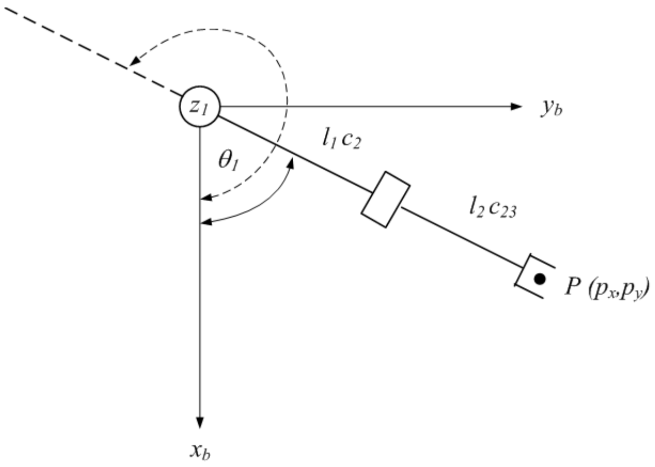


Figure 4.5: Projection of the articulated arm onto x_1-y_1 plane and the related parameters. ([1], pp. 122)

Example 4.3 *Articulated Arm*: Perform the inverse kinematics analysis of the articulated robot illustrated in Fig. 3.5. ([1], Prob. 4.3)

Solution If the first joint does not exist, the robot arm is degenerated to merely a two DOF planar robot. Thus one may apply the analysis result of the three DOF planar robot by setting the length of the third link to zero. Since the robot has just three DOF, it cannot achieve arbitrary specified posture in three dimensions. Therefore, one may specify only the desired end effector position

$$\bar{P} = [p_x \; p_y \; p_z]^T$$

represented in the base frame. If the robot is projected onto the x_1-y_1 plane which is parallel to x_b-y_b, the second and the third link images will be a single line connecting the origin $\begin{bmatrix} 0 & 0 \end{bmatrix}^T$ to the projected end effector point $\begin{bmatrix} p_x & p_y \end{bmatrix}^T$ as depicted in Fig. 4.5. Hence, it can be concluded that

$$\theta_1 = \text{atan2}\,(p_y, p_x)\,.$$

Moreover, when the arm faces its back to P,

$$\theta_1 = \text{atan2}\,(p_y, p_x) + \pi,$$

as the arm can then flip back to reach P as well.

To apply the inverse kinematics analysis of the planar arm in this problem, firstly the end effector position must be expressed in $\{x_1 y_1 z_1\}$. Since the upper and lower arm lie in x_1-z_1 plane as illustrated in Fig. 4.6 for the case when $\theta_1 = \text{atan2}\,(p_y, p_x)$ is chosen,

$${}^{\{x_1 z_1\}}\bar{P} = \begin{bmatrix} \sqrt{p_x^2 + p_y^2} & p_z - h \end{bmatrix}^T.$$

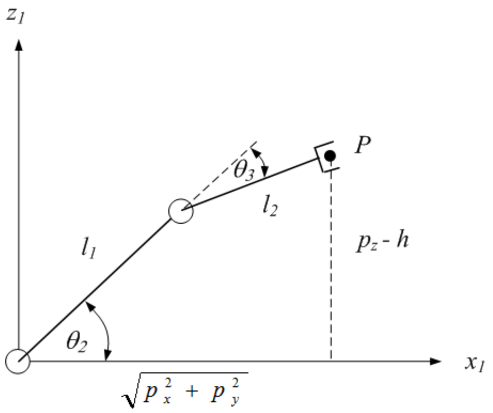


Figure 4.6: Projection of the articulated arm onto x_1-z_1 plane, which degenerates the three dimensional robot to a planar robot. ([1], pp. 123)

Geometrically, this must be equal to

$$\begin{bmatrix} l_1 c_2 + l_2 c_{23} & l_1 s_2 + l_2 s_{23} \end{bmatrix}^T,$$

as observed in Fig. 4.6.

Consequently, inverse kinematics result of the previous example may now be applied as follow.

$$\begin{aligned} c_3 &= \frac{p_x^2 + p_y^2 + p_z^2 + h^2 - 2hp_z - l_1^2 - l_2^2}{2l_1 l_2} \\ s_3 &= \pm\sqrt{1 - c_3^2} \\ \theta_3 &= \text{atan2}\,(s_3, c_3) \\ \theta_2 &= \text{atan2}\left(p_z - h, \sqrt{p_x^2 + p_y^2}\right) - \text{atan2}\,(l_2 s_3, l_1 + l_2 c_3) \end{aligned}$$

There are two solutions for θ_2 and θ_3 according to the elbow up or elbow down configuration.

If $\theta_1 = \text{atan2}\,(p_y, p_x) + \pi$ is chosen instead, the end effector position expressed in $\{x_1 z_1\}$ will be

$${}^{\{x_1 z_1\}}\bar{P} = \begin{bmatrix} -\sqrt{p_x^2 + p_y^2} & p_z - h \end{bmatrix}^T.$$

Hence the corresponding θ_2 and θ_3 will be

$$\begin{aligned} c_3 &= \frac{p_x^2 + p_y^2 + p_z^2 + h^2 - 2hp_z - l_1^2 - l_2^2}{2l_1 l_2} \\ s_3 &= \pm\sqrt{1 - c_3^2} \\ \theta_3 &= \text{atan2}\,(s_3, c_3) \\ \theta_2 &= \text{atan2}\left(p_z - h, -\sqrt{p_x^2 + p_y^2}\right) - \text{atan2}\,(l_2 s_3, l_1 + l_2 c_3) \end{aligned}$$

In summary, for a specified end effector position, mathematically the articulated arm yields four solutions. The arm posture will be in a manner that, for the first two solutions, the robot will face towards the position and the arm be either in the elbow up or elbow down. For the other two solutions, the robot will turn its back to the position and the second joint angle will be rotated greater than 90° so the arm still can reach the position which is located at its back. Looking at the final posture, however, the arm postures are not changed.

Example 4.4 *PUMA 560 Robot*: Perform the inverse kinematics analysis of the six DOF PUMA 560 robot illustrated in Fig. 3.6.

SOLUTION Since the robot has full six DOF, both the desired position and orientation of the end effector may be specified arbitrarily. If

$$ {}^{B}_{E}T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

is given, the inverse kinematics problem is to determine the solution of the robot six joint angles, θ_1 to θ_6. From example 3.2,

$$ {}^{B}_{E}T =^{B}_{0} T^{0}_{6}T^{6}_{E}T. $$

Thus,

$$ \begin{aligned} {}^{0}_{6}T &= \left({}^{B}_{0}T\right)^{-1}\left({}^{B}_{E}T\right)\left({}^{6}_{E}T\right)^{-1} \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -H \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -e \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x - er_{13} \\ r_{21} & r_{22} & r_{23} & p_y - er_{23} \\ r_{31} & r_{32} & r_{33} & p_z - H - er_{33} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \end{aligned} $$

One may need to look for simple relations that contain few unknowns at at time so it may be capable of solving for the analytical solutions. θ_1 may be solved first by rewriting the above statement as

$$\begin{aligned}
\left({}^0_1T\right)^{-1}\left({}^0_6T\right) &= {}^1_6T, \\
\begin{bmatrix} c_1 & s_1 & 0 & 0 \\ -s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x - er_{13} \\ r_{21} & r_{22} & r_{23} & p_y - er_{23} \\ r_{31} & r_{32} & r_{33} & p_z - H - er_{33} \\ 0 & 0 & 0 & 1 \end{bmatrix} & \\
= \begin{bmatrix} * & * & * & a_3c_{23} - d_4s_{23} + a_2c_2 \\ * & * & * & d_3 \\ * & * & * & -d_4c_{23} - a_3s_{23} - a_2s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}. &
\end{aligned}$$

Multiplying the subtransformation matrices in example 3.2, the $(2,4)$ element on the right hand side is the least complex one, which fortunately is independent of other unknowns;

$$(p_y - er_{23})\,c_1 - (p_x - er_{13})\,s_1 = d_3.$$

There are two possible solutions of θ_1, which may be expressed as

$$\begin{aligned}
\theta_1 &= \operatorname{atan2}\left(-\left(p_x - er_{13}\right), p_y - er_{23}\right) \\
&\pm\operatorname{atan2}\left(\sqrt{p_x^2 + p_y^2 - d_3^2 + e^2\left(r_{13}^2 + r_{23}^2\right) - 2e\left(p_xr_{13} + p_yr_{23}\right)}, d_3\right).
\end{aligned}$$

Further, the equality of $(1,4)$ and $(3,4)$ elements on both sides require that

$$\begin{aligned}
a_3c_{23} - d_4s_{23} + a_2c_2 &= (p_x - er_{13})\,c_1 + (p_y - er_{23})\,s_1 \\
-d_4c_{23} - a_3s_{23} - a_2s_2 &= p_z - H - er_{33},
\end{aligned}$$

where now θ_1 is treated as the known variable. Square the above two equations and the first one as well. Sum those three together and rearrange the equation into the following form

$$\begin{aligned}
& a_3c_3 - d_4s_3 \\
&= \frac{(p_x - er_{13})^2 + (p_y - er_{23})^2 + (p_z - H - er_{33})^2 - a_2^2 - a_3^2 - d_3^2 - d_4^2}{2a_2} = K.
\end{aligned}$$

Hence two solutions of θ_3 may be solved in a similar manner as of θ_1, i.e.

$$\theta_3 = \operatorname{atan2}\left(-d_4, d_3\right) \pm \operatorname{atan2}\left(\sqrt{a_3^2 + d_4^2 - K^2}, K\right).$$

The next unknown that should be looked for is θ_2. Consider the governing equation of ${}^0_6T$ again. The equation may be rewritten so the left hand side appears as a function of θ_2. In particular,

$$\left({}^0_3T\right)^{-1}\left({}^0_6T\right) = {}^3_6T,$$

$$\begin{bmatrix} c_1c_{23} & s_1c_{23} & -s_{23} & -a_2c_3 \\ -c_1s_{23} & -s_1s_{23} & -c_{23} & a_2s_3 \\ -s_1 & c_1 & 0 & -d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x - er_{13} \\ r_{21} & r_{22} & r_{23} & p_y - er_{23} \\ r_{31} & r_{32} & r_{33} & p_z - H - er_{33} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} c_4c_5c_6 - s_4s_6 & -c_4c_5s_6 - s_4c_6 & -c_4s_5 & a_3 \\ s_5c_6 & -s_5s_6 & c_5 & d_4 \\ -s_4c_5c_6 - c_4s_6 & s_4c_5s_6 - c_4c_6 & s_4s_5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Setting up the equations from the $(1,4)$ and $(2,4)$ elements since they contain only θ_2 as unknown;

$$\begin{aligned} (p_x - er_{13})\,c_1c_{23} + (p_y - er_{23})\,s_1c_{23} - (p_z - H - er_{33})\,s_{23} - a_2c_3 &= a_3 \\ -(p_x - er_{13})\,c_1s_{23} - (p_y - er_{23})\,s_1s_{23} - (p_z - H - er_{33})\,c_{23} + a_2s_3 &= d_4. \end{aligned}$$

The equations may be solved simultaneously for s_{23} and c_{23} as

$$\begin{aligned} s_{23} &= \frac{-(a_2c_3 + a_3)(p_z - H - er_{33}) + (a_2s_3 - d_4)\left[(p_x - er_{13})\,c_1 + (p_y - er_{23})\,s_1\right]}{\left[(p_x - er_{13})\,c_1 + (p_y - er_{23})\,s_1\right]^2 + (p_z - H - er_{33})^2} \\ c_{23} &= \frac{(a_2c_3 + a_3)\left[(p_x - er_{13})\,c_1 + (p_y - er_{23})\,s_1\right] + (a_2s_3 - d_4)(p_z - H - er_{33})}{\left[(p_x - er_{13})\,c_1 + (p_y - er_{23})\,s_1\right]^2 + (p_z - H - er_{33})^2}. \end{aligned}$$

Thus,

$$\begin{aligned} \theta_{23} = {} & \operatorname{atan2}\left(-(a_2c_3 + a_3)(p_z - H - er_{33}) + (a_2s_3 - d_4)\left[(p_x - er_{13})\,c_1 + (p_y - er_{23})\,s_1\right],\right. \\ & \left.(a_2c_3 + a_3)\left[(p_x - er_{13})\,c_1 + (p_y - er_{23})\,s_1\right] + (a_2s_3 - d_4)(p_z - H - er_{33})\right). \end{aligned}$$

It is seen that the value of θ_{23} depends on θ_1 and θ_3, each of which has two solutions. Hence there are four possible combinations and so four corresponding solutions of

$$\theta_2 = \theta_{23} - \theta_3.$$

The remaining unknowns of θ_4, θ_5, and θ_6 may now be solved straightforwardly if one recognize that the rotation matrix in ${}^3_6T$ is of the same form as the corresponding rotation matrices of the Euler angles or the fixed angles descriptions. Inverse kinematics problem of this kind has already been addressed in subsection 2.5.1 and 2.5.2. The same technique may be applied here as follow.

Assume the value of the $(2,3)$ element of ${}^3_6T$ is neither one or minus one. That is $c_5 \neq \pm 1$ or $\theta_5 \neq 0, \pi$. One then can set up the following relations;

$$\begin{aligned} c_5 &= -(r_{13}c_1s_{23} + r_{23}s_1s_{23} + r_{33}c_{23}) \\ s_5 &= \pm\sqrt{1 - (r_{13}c_1s_{23} + r_{23}s_1s_{23} + r_{33}c_{23})^2}. \end{aligned}$$

According to the selected values of θ_1, θ_2, and θ_3, there are two possible solutions of θ_5:

$$\theta_5 = \operatorname{atan2}\left(\pm\sqrt{1-\left(r_{13}c_1s_{23}+r_{23}s_1s_{23}+r_{33}c_{23}\right)^2}, -\left(r_{13}c_1s_{23}+r_{23}s_1s_{23}+r_{33}c_{23}\right)\right).$$

The remaining variables of θ_4 and θ_6 depends on the selected value of θ_5. Specifically, if $s_5 = +\sqrt{1-\left(r_{13}c_1s_{23}+r_{23}s_1s_{23}+r_{33}c_{23}\right)^2}$ is chosen, then

$$\begin{aligned}\theta_4 &= \operatorname{atan2}\left(-r_{13}s_1+r_{23}c_1, -\left(r_{13}c_1c_{23}+r_{23}s_1c_{23}-r_{33}s_{23}\right)\right)\\ \theta_6 &= \operatorname{atan2}\left(r_{12}c_1s_{23}+r_{22}s_1s_{23}+r_{32}c_{23}, -\left(r_{11}c_1s_{23}+r_{21}s_1s_{23}+r_{31}c_{23}\right)\right).\end{aligned}$$

Instead, if $s_5 = -\sqrt{1-\left(r_{13}c_1s_{23}+r_{23}s_1s_{23}+r_{33}c_{23}\right)^2}$ is chosen,

$$\begin{aligned}\theta_4 &= \operatorname{atan2}\left(r_{13}s_1-r_{23}c_1, r_{13}c_1c_{23}+r_{23}s_1c_{23}-r_{33}s_{23}\right)\\ \theta_6 &= \operatorname{atan2}\left(-\left(r_{12}c_1s_{23}+r_{22}s_1s_{23}+r_{32}c_{23}\right), r_{11}c_1s_{23}+r_{21}s_1s_{23}+r_{31}c_{23}\right).\end{aligned}$$

When $-\left(r_{13}c_1s_{23}+r_{23}s_1s_{23}+r_{33}c_{23}\right) = 1$, $\theta_5 = 0$. However, it will not be possible to solve for θ_4 and θ_6. This is the 'physical' singular configuration (c.f. the representational configuration in subsection 2.5.1 and 2.5.2) of the roll-pitch-roll wrist, which will happen when the wrist outstretches so the joint axes of θ_4 and θ_6 line up. In this configuration, one would never know what are exactly the contributing angles from the fourth and the sixth joints to the resultant rolling angle of the end effector. Only their sum would know. Mathematically,

$$\begin{aligned}\theta_5 &= 0\\ \theta_4+\theta_6 &= \operatorname{atan2}\left(r_{11}s_1-r_{21}c_1, r_{12}s_1-r_{22}c_1\right).\end{aligned}$$

Similarly, when $-\left(r_{13}c_1s_{23}+r_{23}s_1s_{23}+r_{33}c_{23}\right) = -1$, $\theta_5 = \pi$. This is another physical singular configuration of the wrist corresponding to the time when the wrist fully folds back. For this configuration, the difference of the fourth and the sixth joint angles yields the end effector rolling angle. In other words,

$$\begin{aligned}\theta_5 &= 0\\ \theta_4-\theta_6 &= \operatorname{atan2}\left(-r_{11}s_1+r_{21}c_1, r_{12}s_1-r_{22}c_1\right).\end{aligned}$$

In summary, if the robot is not in one of its singular configuration, there will be eight solutions of the joint angles which can achieve the specified end effector posture. The first three joints have four choices corresponding to the left/right arm and elbow up/down solutions as illustrated in Fig. 4.7. There are two solutions for the latter three joints for the wrist to flip up/down. However, some of them may not be possible because they will make the arm jammed with the environment. The solution to choose for will, in general, be the one of which the joint angles are closest to the present values. In other words, the solution will stay in the same branch. This choice results in smooth motion of the robot since it will not run into the singular configuration unnecessarily.

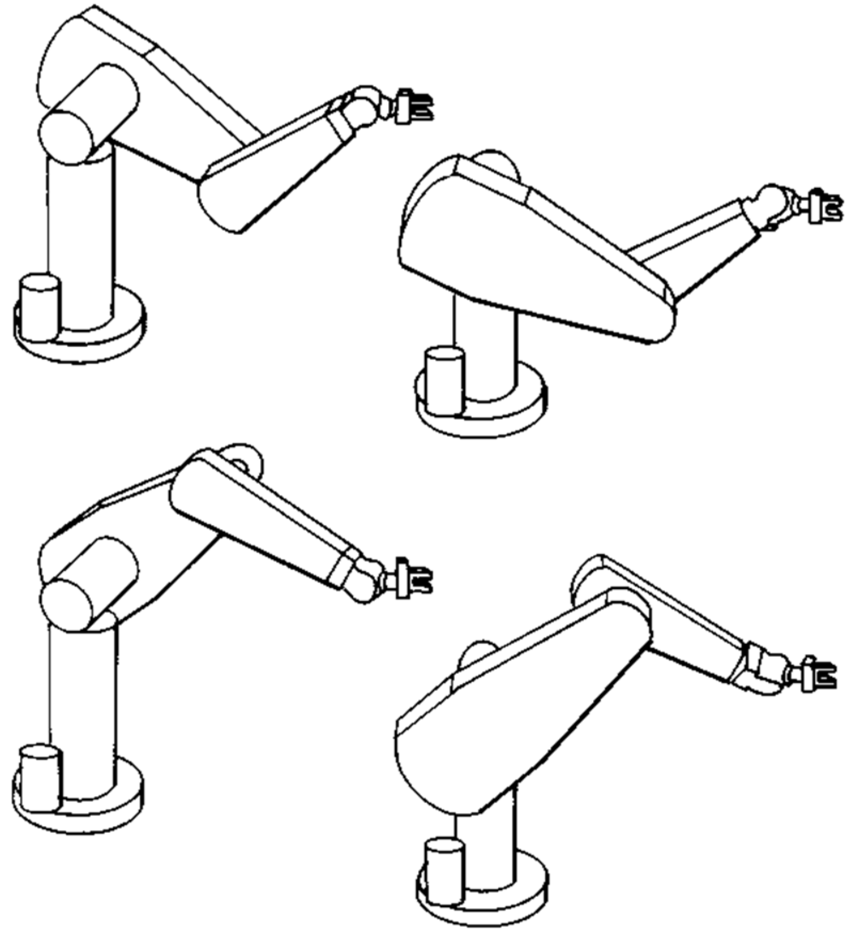


Figure 4.7: Four solutions for the first three joints of PUMA 560. ([4], pp. 105)

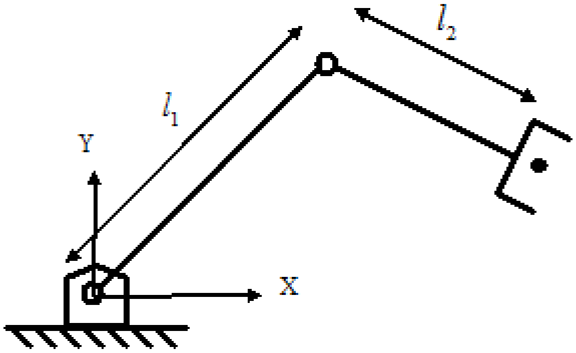


Figure 4.8: Schematic diagram of a 2-DOF revolute joint planar arm.

Problems

1. Perform the inverse kinematics analysis of a simple 2-DOF planar robot in Fig. 4.8 by both the algebraic and geometric approach.

2. Determine the inverse kinematics of the robot arm in Prob. 2 of Chapter 3.

3. Determine the inverse kinematics of the 3-DOF non-orthogonal wrist in Prob. 3 of Chapter 3 from its forward kinematics result.

4. Determine the inverse kinematics of the SCARA arm depicted in Prob. 4 of Chapter 3 by your convenient method.

Chapter 5

Jacobians

This chapter analyzes the robot kinematics at the velocity level to understand how the robot will be moved by the motion of its joints. The velocity propagation approach will be studied to derive the velocity kinematics in section 5.1. The result is a linear mapping from the joint velocity space to the end effector spatial velocity space. It may be compactly expressed by the geometric Jacobian matrix, as discussed in section 5.2, for which it can be constructed on-the-fly.

Since Jacobian matrix is an important matrix in robot analysis, for this introductory course, two usages of it for the singularity analysis and the static force analysis will be studied in section 5.3 and 5.4.

5.1 Velocity Propagation

Velocity kinematics analysis of the robot is the determination of the linear and angular velocity at various points of the end effector or the linkages. Several methods are available to develop these quantities. In this course, the method of velocity propagation taught in the introductory to dynamics course will be adopted.

5.1.1 Relative Linear Velocity of Any Two Points

Consider any two points A and B moving independently in three dimensional space, depicted in Fig. 5.1. Motion of A may be observed from the observer who travels with B. From the figure, $\{xyz\}$, which has its origin at B and is rotating with the angular velocity $\bar{\omega}$, is used to describe the position vector of A relative to B:

$$\bar{p}_{A/B} = x\hat{\imath} + y\hat{\jmath} + z\hat{k}. \tag{5.1}$$

Since $\bar{p}_{A/B}$ describes the motion of A relative to B, its time differentiation is the relative velocity of A with respect to B.

$$\bar{v}_A - \bar{v}_B = \frac{d}{dt}\bar{p}_{A/B} = \left(x\frac{d}{dt}\hat{\imath} + y\frac{d}{dt}\hat{\jmath} + z\frac{d}{dt}\hat{k}\right) + \left(\dot{x}\hat{\imath} + \dot{y}\hat{\jmath} + \dot{z}\hat{k}\right).$$

Because $\hat{\imath}$, $\hat{\jmath}$, and $\hat{k}$ are unit vectors, the time rate of change of these vectors are due solely to their changes in direction. Indeed,

$$\frac{d}{dt}\hat{\imath} = \bar{\omega} \times \hat{\imath} \qquad \frac{d}{dt}\hat{\jmath} = \bar{\omega} \times \hat{\jmath} \qquad \frac{d}{dt}\hat{k} = \bar{\omega} \times \hat{k}.$$

Substitute these relations into the above relative velocity equation. After some manipulation, the velocity of A may be written as

$$\bar{v}_A = \bar{v}_B + \bar{\omega} \times \bar{p}_{A/B} + \bar{v}_{\text{rel}}, \tag{5.2}$$

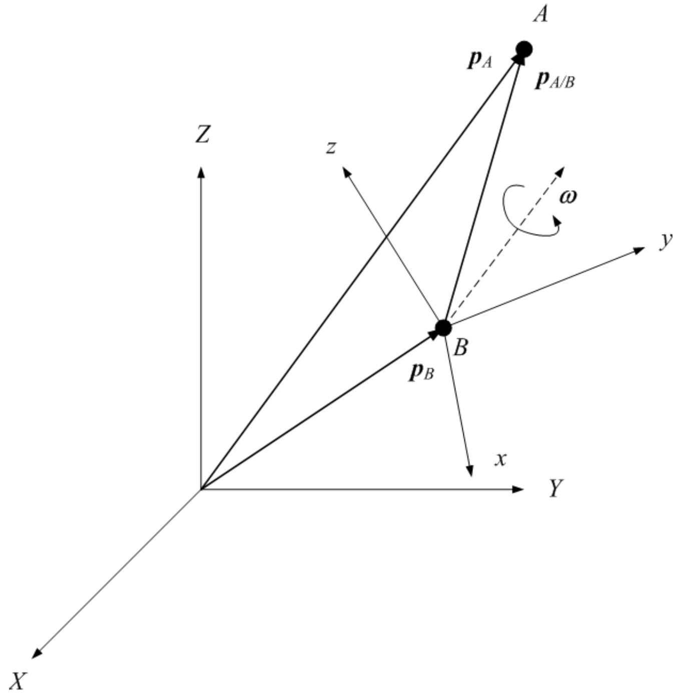


Figure 5.1: Relevant frames and position vectors for the relative motion analysis of arbitrary two points A and B. ([1], pp. 171)

where $\bar{v}_{\text{rel}} = \dot{x}\hat{i} + \dot{y}\hat{j} + \dot{z}\hat{k}$ is the velocity of A seen by the observer. This equation says that the velocity of A may be determined from the velocity of B once their relative velocity is known.

5.1.2 Relative Angular Velocity of the Objects

Angular velocity of an object describes the time rate of change of its rotation. Let $\bar{\omega}_C$ and $\bar{\omega}_D$ be the angular velocity of two objects C and D. Since the angular velocity is a free vector, the difference of the angular velocity of C and D, called the relative angular velocity of C with respect to D, may be calculated simply by

$$\bar{\omega}_{C/D} = \bar{\omega}_C - \bar{\omega}_D.$$

Thus, it may be said that the angular velocity of C may be determined from the angular velocity of D once their relative velocity is known:

$$\bar{\omega}_C = \bar{\omega}_D + \bar{\omega}_{C/D}. \tag{5.3}$$

5.1.3 Velocity of the Robot

In this subsection, the relative velocity formula developed earlier will be applied to determine the velocity of the robot. Consider link#$(i-1)$ and link#i for

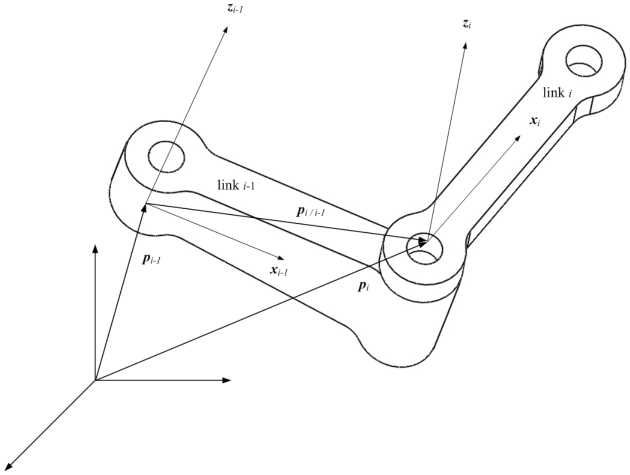


Figure 5.2: Link frames and the relevant position vectors. ([1], pp. 172)

$i \in [1, 2, \ldots, n]$ of a serial link manipulator. Frames $\{i-1\}$ and $\{i\}$ established after DH convention and the pertaining position vectors are depicted in Fig. 5.2. Recalling Eq. 5.2, the linear velocity of $\{i\}$ may be determined from the linear and angular velocity of $\{i-1\}$;

$$\bar{v}_i = \bar{v}_{i-1} + \bar{\omega}_{i-1} \times \bar{p}_{i/i-1} + \bar{v}_{\text{rel}}. \tag{5.4}$$

In dual, the angular velocity of $\{i\}$ may be determined from the angular velocity of $\{i-1\}$ as

$$\bar{\omega}_i = \bar{\omega}_{i-1} + \bar{\omega}_{i/i-1}. \tag{5.5}$$

If joint#i is the prismatic joint, there will be no relative rotation of $\{i\}$ and $\{i-1\}$. Hence, $\bar{\omega}_{i/i-1} = \bar{0}$. Substituting this value into Eq. 5.5,

$$\bar{\omega}_i = \bar{\omega}_{i-1}. \tag{5.6}$$

Instead, the prismatic actuation causes the motion of the origin of $\{i\}$ with respect to the origin of $\{i-1\}$ by the relative linear velocity $\bar{v}_{\text{rel}} = \dot{d}_i\hat{k}_i$. Applying this value to Eq. 5.4,

$$\bar{v}_i = \bar{v}_{i-1} + \bar{\omega}_{i-1} \times \bar{p}_{i/i-1} + \dot{d}_i\hat{k}_i. \tag{5.7}$$

If joint#i is the revolute joint, it will cause the rotation of $\{i\}$ with respect to $\{i-1\}$ by the relative angular velocity $\bar{\omega}_{i/i-1} = \dot{\theta}_i\hat{k}_i$. Substituting such value into Eq. 5.5,

$$\bar{\omega}_i = \bar{\omega}_{i-1} + \dot{\theta}_i\hat{k}_i. \tag{5.8}$$

However, the rotating actuation causes no motion of the origin of $\{i\}$ relative to the origin of $\{i-1\}$. As a result, $\bar{v}_{\text{rel}} = \bar{0}$. Consequently, from Eq. 5.4,

$$\bar{v}_i = \bar{v}_{i-1} + \bar{\omega}_{i-1} \times \bar{p}_{i/i-1}. \tag{5.9}$$

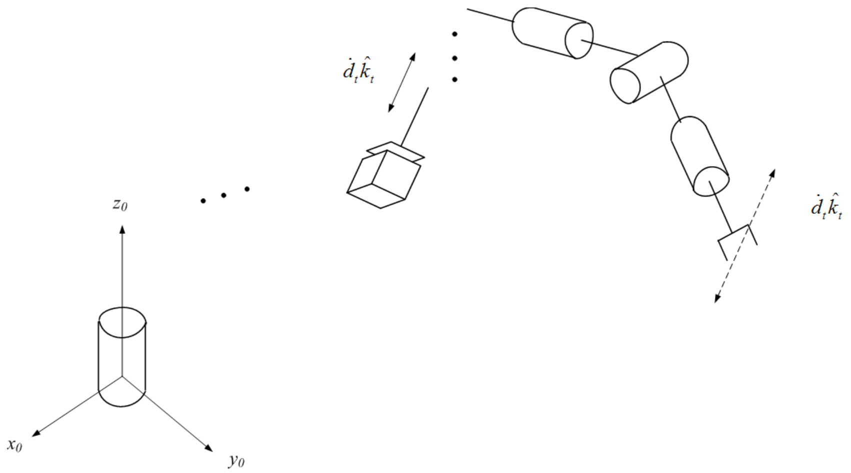


Figure 5.3: Influence of the prismatic joint actutation onto the end effector linear velocity. ([1], pp. 174)

5.2 Jacobian Matrix

From the previous section, it is seen that the velocity of the end effector when the serial robot possesses only the simple joints may be computed from the developed relative velocity equations 5.6, 5.7, 5.8, and 5.9 successively. In addition, the following relative velocity equations are used to determine the velocity of $\{E\}$ from the velocity of $\{n\}$:

$$\begin{aligned}\bar{\omega}_E &= \bar{\omega}_n \\ \bar{v}_E &= \bar{v}_n + \bar{\omega}_n \times \bar{p}_{E/n}.\end{aligned} \tag{5.10}$$

Let n_r and n_t be the set of the revolute and prismatic joint enumerations, respectively. Thus the end effector angular velocity may be expressed compactly as

$$\bar{\omega}_E = \sum_{r\in n_r} \dot{\theta}_r \hat{k}_r. \tag{5.11}$$

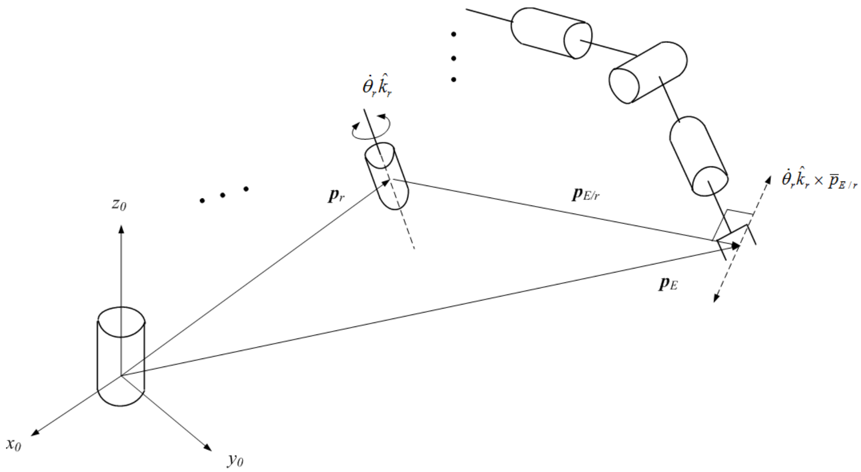
The equation is developed from Eq. 5.10 using the recursive substitution of Eqs. 5.6 and 5.8. Note the base is assumed to be immobile.

Linear velocity of the end effector may be determined in the same vein. With the recursive substitution of Eqs. 5.7, 5.9, 5.6, and 5.8 into Eq. 5.10, linear velocity of the end effector may be computed by

$$\bar{v}_E = \sum_{t\in n_t} \dot{d}_t \hat{k}_t + \sum_{r\in n_r} \dot{\theta}_r \hat{k}_r \times \bar{p}_{E/r}, \tag{5.12}$$

on the condition $\bar{v}_0 = \bar{0}$ and $\bar{\omega}_0 = \bar{0}$.

From Eq. 5.11, the angular velocity of the end effector is the sum of the angular velocity of all revolute joints. In other words, motion from the prismatic



$\dot{\theta}_r \hat{k}_r$
z_0
p_r
$p_{E/r}$
$\dot{\theta}_r \hat{k}_r \times \bar{p}_{E/r}$
p_E
x_0
y_0

Figure 5.4: Influence of the revolute joint actutation onto the end effector linear velocity. ([1], pp. 175)

joints do not contribute to the end effector angular velocity. On the other hand, as shown in Eq. 5.12, motion from both types of the joints have an affect on the linear velocity of the end effector. Specifically, motion of the prismatic joints contribute to the end effector linear velocity directly as the sum of the linear velocity of all prismatic joints. However, the end effector motion caused by the revolute joints depends on the position vector $\bar{p}_{E/r}$ from the joint to the end effector as well. Their contributions to the linear velocity are expressed by the term $\sum_{r \in n_r} \dot{\theta}_r \hat{k}_r \times \bar{p}_{E/r}$ in Eq. 5.12.

Furthermore, it can be observed from Eqs. 5.11 and 5.12 that each joint velocity influences the end effector velocity indepedently. This can be imagined that at a time there is only one joint being actuated while the others are freezed. Figure 5.3 illustrates the linear motion of the end effector caused by the prismatic joint actuation while Fig. 5.4 shows the linear motion caused by the revolute joint, corresponding to the terms in Eq. 5.12.

The terms $\hat{k}_i$ and $\bar{p}_{E/i}$ in Eqs. 5.11 and 5.12 are the functions of the joint variables $\bar{q}$ solely corresponding to the explicit separation of $\dot{\bar{q}}$, consisting of $\dot{d}_t$ and $\dot{\theta}_r$, in the expressions. Hence both equations may be written as the multiplication of the matrix and the column vector of the joint velocity as

$$\begin{bmatrix} \bar{v}_E \\ \bar{\omega}_E \end{bmatrix} = J(\bar{q})\,\dot{\bar{q}} = \begin{bmatrix} J_v(\bar{q}) \\ J_\omega(\bar{q}) \end{bmatrix} \dot{\bar{q}}. \tag{5.13}$$

This equation indicates the linear mapping by the matrix $J(\bar{q})$ from the space of the joint velocity $\dot{\bar{q}}$ to the space of the end effector velocity $\begin{bmatrix} \bar{v}_E^T & \bar{\omega}_E^T \end{bmatrix}^T$ when the current robot pose corresponds to the joint variables $\bar{q}$. The matrix $J(\bar{q})$ is the (geometric) *Jacobian matrix* describing the contribution of the each joint motion to the robot end effector motion. The matrix is related to the partial derivatives of the multivariable functions, which is originated from the idea of a great German mathematician named Carl Gustav Jacob Jacobi on the topic of Jacobian determinant.

From Eqs. 5.11 and 5.12, the Jacobian matrix may be decomposed into two submatrices as shown in Eq. 5.13. The submatrix $J_v(\bar{q})$ dictates the relationship between the joint velocity vector and the end effector linear velocity. Similarly, the submatrix $J_\omega(\bar{q})$ governs the relationship between the joint velocity vector and the end effector angular velocity. $J_v(\bar{q})$ may be determined from Eq. 5.12 and $J_\omega(\bar{q})$ from Eq. 5.11. The i^{th}-column of $J(\bar{q})$, denoted J_i, when $i \in [1, 2, \ldots, n]$ may be expressed as

$$J_i = \begin{bmatrix} J_{vi} \\ J_{\omega i} \end{bmatrix} = \begin{cases} \begin{bmatrix} \hat{k}_i \\ \bar{0} \end{bmatrix}, & i^{\text{th}} - \text{prismatic joint} \\ \begin{bmatrix} \hat{k}_i \times \bar{p}_{E/i} \\ \hat{k}_i \end{bmatrix}, & i^{\text{th}} - \text{revolute joint} \end{cases}. \tag{5.14}$$

The term $\hat{k}_i$ is the unit vector along the i^{th}-joint axis. If it is described in the reference frame, its value is determined by the third column of the rotation matrix ${}^{B}_{i}R$. The other term $\bar{p}_{E/i}$ denotes the displacement vector pointing the origin of $\{E\}$ with respect to the origin of $\{i\}$. It is determined by subtracting the embedded position vectors in ${}^{B}_{E}T$ and ${}^{B}_{i}T$ when $\bar{p}_{E/i}$ is thus expressed in the base frame. A suggestion on the way is therefore to complete the manipulator kinematics analysis before deriving the robot Jacobian matrix. Also, the analysis may be applied to determine the linear velocity of arbitrary point on the robot and the angular velocity of any link.

Example 5.1 *Articulated Arm*: Determine the Jacobian matrix of the articulated robot in Fig. 3.5.

Solution The Jacobian matrix linking the end effector velocity to the joint velocity may be determined straightforwardly by Eq. 5.14. Since all three joints of the robot are the revolute type,

$$J(\bar{q}) = \begin{bmatrix} \hat{k}_1 \times (\bar{p}_E - \bar{p}_1) & \hat{k}_2 \times (\bar{p}_E - \bar{p}_2) & \hat{k}_3 \times (\bar{p}_E - \bar{p}_3) \\ \hat{k}_1 & \hat{k}_2 & \hat{k}_3 \end{bmatrix}.$$

All relevant parameters may be computed from the manipulator kinematics analysis in Ex. 3.1. If the Jacobian matrix is chosen to be expressed in the base frame, the unit vectors $\hat{k}_1$, $\hat{k}_2$, and $\hat{k}_3$ is the third column vector of ${}^{0}_{1}R$, ${}^{0}_{2}R$, and ${}^{0}_{3}R$ respectively. In particular,

$$\hat{k}_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad \hat{k}_2 = \begin{bmatrix} s_1 \\ -c_1 \\ 0 \end{bmatrix} \qquad \hat{k}_3 = \begin{bmatrix} s_1 \\ -c_1 \\ 0 \end{bmatrix}.$$

Note that $\hat{k}_2 = \hat{k}_3$ reflecting the second and the third joint axis pointing in the same direction.

Position vector in ${}^0_1T$, ${}^0_2T$, ${}^0_3T$, and ${}^0_ET$ is $\bar{p}_1$, $\bar{p}_2$, $\bar{p}_3$, and $\bar{p}_E$ expressed in $\{0\}$. Their values are

$$\bar{p}_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad \bar{p}_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad \bar{p}_3 = \begin{bmatrix} l_1c_1c_2 \\ l_1s_1c_2 \\ l_1s_2 \end{bmatrix} \qquad \bar{p}_E = \begin{bmatrix} c_1\left(l_1c_2 + l_2c_{23}\right) \\ s_1\left(l_1c_2 + l_2c_{23}\right) \\ l_1s_2 + l_2s_{23} \end{bmatrix}.$$

Substitute these values into the Jacobian matrix and perform the evaluation, one gets

$$J\left(\theta_1, \theta_2, \theta_3\right) = \begin{bmatrix} -s_1\left(l_1c_2 + l_2c_{23}\right) & -c_1\left(l_1s_2 + l_2s_{23}\right) & -l_2c_1s_{23} \\ c_1\left(l_1c_2 + l_2c_{23}\right) & -s_1\left(l_1s_2 + l_2s_{23}\right) & -l_2s_1s_{23} \\ 0 & l_1c_2 + l_2c_{23} & l_2c_{23} \\ 0 & s_1 & s_1 \\ 0 & -c_1 & -c_1 \\ 1 & 0 & 0 \end{bmatrix}.$$

Example 5.2 *PUMA 560 Robot*: Determine the Jacobian matrix of the PUMA 560 robot in Fig. 3.6.

Solution For this six DOF robot where the last three joints constitute the spherical wrist, it is more convenient to formulate the Jacobian matrix of the wrist center rather than the Jacobian matrix of the end effector. In addition, this choice will simplify the singularity analysis as done in Ex. 5.4.

The Jacobian matrix which calculates the velocity $\begin{bmatrix} \bar{v}_{O_6}^T & \bar{\omega}_{O_6}^T \end{bmatrix}^T$ of the wrist center on the last link will be expressed in $\{3\}$ to reduce the complexity. For this purpose, the forward kinematics performed in Ex. 3.2 must be represented in $\{3\}$. Using the compound transformation, the homogeneous transformation of every frame described with respect to $\{3\}$ may be computed.

$${}^3_1T = {}^3_0T\,{}^0_1T = \begin{bmatrix} c_{23} & 0 & -s_{23} & -a_2c_3 \\ -s_{23} & 0 & -c_{23} & a_2s_3 \\ 0 & 1 & 0 & -d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^3_2T = {}^3_1T\,{}^1_2T = \begin{bmatrix} c_3 & s_3 & 0 & -a_2c_3 \\ -s_3 & c_3 & 0 & a_2s_3 \\ 0 & 0 & 1 & -d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^3_3T = I_{4\times 4} \qquad {}^3_4T = \begin{bmatrix} c_4 & -s_4 & 0 & a_3 \\ 0 & 0 & 1 & d_4 \\ -s_4 & -c_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$
{}^{3}_{5}T = {}^{3}_{4}T\,{}^{4}_{5}T = \begin{bmatrix} c_4c_5 & -c_4s_5 & s_4 & a_3 \\ s_5 & c_5 & 0 & d_4 \\ -s_4c_5 & s_4s_5 & c_4 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{3}_{6}T = {}^{3}_{5}T\,{}^{5}_{6}T = \begin{bmatrix} -s_4s_6 + c_4c_5c_6 & -s_4c_6 - c_4c_5s_6 & -c_4s_5 & a_3 \\ s_5c_6 & -s_5s_6 & c_5 & d_4 \\ -c_4s_6 - s_4c_5c_6 & -c_4c_6 + s_4c_5s_6 & s_4s_5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
$$

According to Eq. 5.14 and since all joints are the revolute type, the Jacobian matrix of the PUMA 560 will have the form of

$$
\begin{aligned}
J(\bar{q}) \;=\; & \left[\begin{matrix} \hat{k}_1 \times (\bar{p}_{O_6} - \bar{p}_1) & \hat{k}_2 \times (\bar{p}_{O_6} - \bar{p}_2) & \hat{k}_3 \times (\bar{p}_{O_6} - \bar{p}_3) \\ \hat{k}_1 & \hat{k}_2 & \hat{k}_3 \end{matrix} \right. \\
& \left. \begin{matrix} \hat{k}_4 \times (\bar{p}_{O_6} - \bar{p}_4) & \hat{k}_5 \times (\bar{p}_{O_6} - \bar{p}_5) & \hat{k}_6 \times (\bar{p}_{O_6} - \bar{p}_6) \\ \hat{k}_4 & \hat{k}_5 & \hat{k}_6 \end{matrix} \right].
\end{aligned}
$$

Since the Jacobian matrix is chosen to be expressed in $\{3\}$, the unit vectors $\hat{k}_1$, $\hat{k}_2$, $\ldots$, $\hat{k}_6$ is the third column of ${}^{3}_{1}R$, ${}^{3}_{2}R$, $\ldots$, ${}^{3}_{6}R$ in order. Specifically from the above homogeneous transformation matrices,

$$
\hat{k}_1 = \begin{bmatrix} -s_{23} \\ -c_{23} \\ 0 \end{bmatrix} \qquad \hat{k}_2 = \hat{k}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}
$$

$$
\hat{k}_4 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \qquad \hat{k}_5 = \begin{bmatrix} s_4 \\ 0 \\ c_4 \end{bmatrix} \qquad \hat{k}_6 = \begin{bmatrix} -c_4s_5 \\ c_5 \\ s_4s_5 \end{bmatrix}.
$$

Similarly, position vectors $\bar{p}_1$, $\bar{p}_2$, $\ldots$, $\bar{p}_6$ is the position vector in ${}^{3}_{1}T$, ${}^{3}_{2}T$, $\ldots$, ${}^{3}_{6}T$ in order. Lastly, $\bar{p}_{O_6} = \bar{p}_6$ since the wrist center always coincides with the origin of $\{6\}$. In particular,

$$
\bar{p}_1 = \bar{p}_2 = \begin{bmatrix} -a_2c_3 \\ a_2s_3 \\ -d_3 \end{bmatrix} \qquad \bar{p}_3 = \bar{0}_3 \qquad \bar{p}_4 = \bar{p}_5 = \bar{p}_6 = \bar{p}_{O_6} = \begin{bmatrix} a_3 \\ d_4 \\ 0 \end{bmatrix}.
$$

Substituting these parameters into the form of the Jacobian matrix above, one obtain

$$
J(\bar{q}) = \begin{bmatrix} -d_3c_{23} & a_2s_3 - d_4 & -d_4 & 0 & 0 & 0 \\ d_3s_{23} & a_2c_3 + a_3 & a_3 & 0 & 0 & 0 \\ a_2c_2 + a_3c_{23} - d_4s_{23} & 0 & 0 & 0 & 0 & 0 \\ -s_{23} & 0 & 0 & 0 & s_4 & -c_4s_5 \\ -c_{23} & 0 & 0 & 1 & 0 & c_5 \\ 0 & 1 & 1 & 0 & c_4 & s_4s_5 \end{bmatrix}.
$$

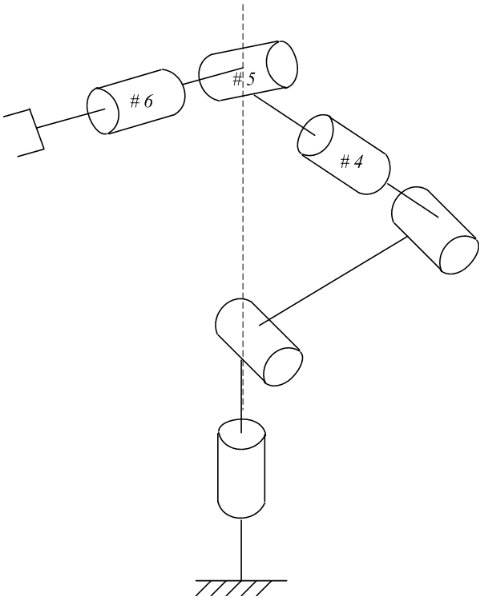


Figure 5.5: Workspace-interior singularity of the articulated arm when the wrist center lies on the first joint axis. This is a case where the singularity locations are known a-priori. ([1], pp. 194)

5.3 Singularities

As explained in the previous section, the Jacobian matrix $J\left(\bar{q}\right)$ is the linear mapping from the space of the joint velocity to the space of the end effector velocity as depicted in Eq. 5.13. The equation may be rewritten as the linear combination of the columns J_i of the Jacobian matrix;

$$\begin{bmatrix} \bar{v}_E \\ \bar{\omega}_E \end{bmatrix} = J_1\dot{q}_1 + J_2\dot{q}_2 + \cdots + J_n\dot{q}_n. \tag{5.15}$$

Because the end effector velocity is the six dimensional real vector space, or $\begin{bmatrix} \bar{v}_E^T & \bar{\omega}_E^T \end{bmatrix}^T \in \mathbb{R}^6$, for the end effector to be able to move arbitrarily, the number of linearly indepedent columns of Jacobian matrix must be equal to six. From the linear algebra, the rank of a matrix will be equal to the number of maximally linearly independent rows or columns of that matrix. Therefore, it may be stated that for the end effctor to be capable of executing arbitrary motion, the rank of its Jacobian matrix must be equal to six. For $J \in \mathbb{R}^{6\times n}$, rank $J \leq \min\left(6, n\right)$.

Since the Jacobian matrix is the function of the joint variables $\bar{q}$, its value and hence its rank will be changed accordingly. The robot is said to be at the *singular configuration* when rank J is less than its maximally possible value. Awareness of the singularity of the robot is important for the following reasons.

1. At the singularity, the robot cannot achieve arbitrary motion. In other

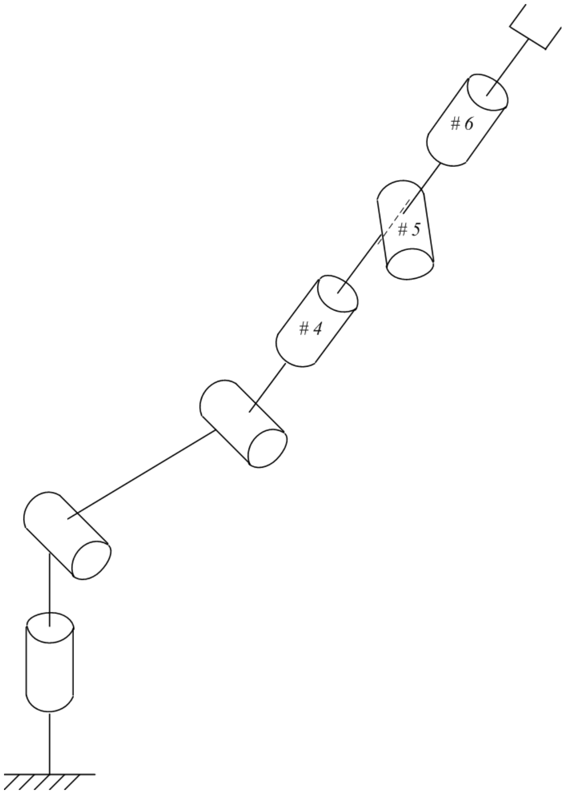


Figure 5.6: Workspace-interior singularity of the articulated arm when the fourth and sixth joint axes of the wrist aligns. It may happen at any location in the workspace. ([1], pp. 195)

words, there are some directions in which the end effector cannot translate or rotate.

2. There may be infinitely many solutions of the robot inverse kinematics at the singularity.

3. At the singularity, the bounded operational space velocity of the end effector may require the unbounded joint space velocity. Also, in the neighborhood of the singularity, even the robot end effector moves with low speed, its joint velocity might be very large.

4. At the singularity, the bounded joint torque vector may be associated with the unbounded force acting at the end effector. Additionally, in the neighborhood of the singularity, even the robot is supplied with small joint torque, the applied end effector force might be very large.

Such behaviors cause the problem in controlling the robot at or near the singularities. Thus it is necessary to analyze the robot singularity throughout the workspace first. Otherwise, the computed control signals will be impractically large, making the response not perform as expected. The system becomes unstable eventually. Besides, singularities may be classified according to its location into two types as

1. Workspace-boundary Singularity is the singularity which happens when the robot fully outstretches or fully folds back their linkages. With these configurations, the robot is unable to move further in such direction. Typically, this type of singularity is not a problem since their locations are predetermined. In practice, the robot trajectory will be planned to stay far enough from the workspace boundary and this singularity can be avoided.

2. Workspace-interior Singularity is the singularity which happens when the end effector is not at the workspace boundary. This type of singularity is often caused by the parallelism of two or more robot joint axes. Since the end effector is still be inside the workspace, it may cause a problem of non-smooth trajectory when the robot passes through the singular point.
   Corresponding posture of some interior singularity configurations brings the information of where they happen. For example, it is known that the robot will run into the singularity when the wrist center of the articulated arm lies on the first joint axis. See Fig. 5.5. Thus, the singularity may be prevented by designing the end effector trajectory not to get close to the first joint axis.
   However, it might not be possible to identify where the interior singularity configuration will happen. Figure 5.6 displays the articulated robot singularity when the fourth and sixth joint axes of the spherical wrist aligns.

> Precise location of the end effector when this singularity occurs is not known until the inverse kinematics analysis is performed.

Determination of the robot singularity for the special case where the dimension of the Jacobian matrix is $n \times n$, happening when the number of the operational DOF and the number of the joints are equal, may be performed simply by checking its determinant. Specifically, the robot will be at the singular configuration if and only if

$$\det\ J(\bar{q}) = 0. \tag{5.16}$$

Example 5.3 *Articulated Arm*: Analyze the singularity of the articulated robot in Fig. 3.5.

Solution In Ex. 5.1, the Jacobian matrix of the robot is derived;

$$J(\theta_1, \theta_2, \theta_3) = \begin{bmatrix} -s_1(l_1c_2 + l_2c_{23}) & -c_1(l_1s_2 + l_2s_{23}) & -l_2c_1s_{23} \\ c_1(l_1c_2 + l_2c_{23}) & -s_1(l_1s_2 + l_2s_{23}) & -l_2s_1s_{23} \\ 0 & l_1c_2 + l_2c_{23} & l_2c_{23} \\ 0 & s_1 & s_1 \\ 0 & -c_1 & -c_1 \\ 1 & 0 & 0 \end{bmatrix}.$$

Dimension of the Jacobian matrix is 6×3, in which the determinant is not defined. However, if the specified task involves with the translational motion of the end effector only, the Jacobian matrix may be degenerated into the sub-Jacobian matrix of the linear velocity portion. That is

$$J_v(\theta_1, \theta_2, \theta_3) = \begin{bmatrix} -s_1(l_1c_2 + l_2c_{23}) & -c_1(l_1s_2 + l_2s_{23}) & -l_2c_1s_{23} \\ c_1(l_1c_2 + l_2c_{23}) & -s_1(l_1s_2 + l_2s_{23}) & -l_2s_1s_{23} \\ 0 & l_1c_2 + l_2c_{23} & l_2c_{23} \end{bmatrix}.$$

Consequently, the robot singularity may be determined from the following condition of

$$\det\ J_v(\bar{q}) = -l_1l_2s_3(l_1c_2 + l_2c_{23}) = 0.$$

This equation will be satisfied when $s_3 = 0$ and/or $l_1c_2 + l_2c_{23} = 0$.

$s_3 = 0$ will be true when $\theta_3 = 0$ or π. This makes the links l_1 and l_2 lie in the same straight line. The arm will be fully outstretch or fully fold back accordingly. These configurations are the boundary singularity type, as shown in Fig. 5.7

The other case of $l_1c_2 + l_2c_{23} = 0$ will occur when the end effector lies on the first joint axis for which this singularity is of the interior singularity type. See Fig. 5.8. The end effector will not be able to move in the direction perpendicular to the plane formed by the links l_1 and l_2. Moreover, there are infinitely many solutions of the robot inverse kinematics at this configuration. Value of the first joint can be chosen arbitrarily as it does not cause any translation of the end effector.

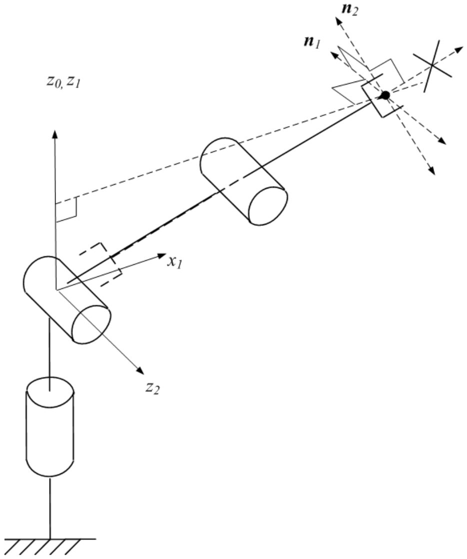


Figure 5.7: Singularity of the articulated arm when the arm fully outstretches or folds back.

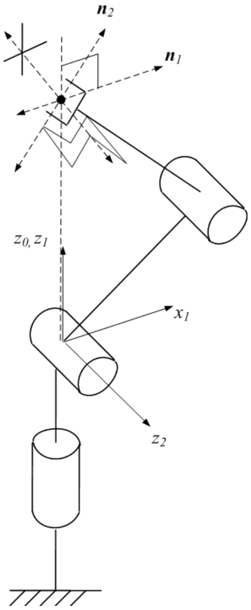


Figure 5.8: Singularity of the articulated arm when the end effector lies on the first joint axis.

Example 5.4 *PUMA 560 Robot*: Determine the singularity of the PUMA 560 robot in Fig. 3.6.

Solution The Jacobian matrix of PUMA 560 is derived in Ex. 5.2. It is shown again here

$$J\left(\bar{q}\right)=\begin{bmatrix} -d_3c_{23} & a_2s_3-d_4 & -d_4 & 0 & 0 & 0 \\ d_3s_{23} & a_2c_3+a_3 & a_3 & 0 & 0 & 0 \\ a_2c_2+a_3c_{23}-d_4s_{23} & 0 & 0 & 0 & 0 & 0 \\ -s_{23} & 0 & 0 & 0 & s_4 & -c_4s_5 \\ -c_{23} & 0 & 0 & 1 & 0 & c_5 \\ 0 & 1 & 1 & 0 & c_4 & s_4s_5 \end{bmatrix}=\begin{bmatrix} J_{11} & \bar{0}_{3\times 3} \\ J_{21} & J_{22} \end{bmatrix}$$

for convenience. The matrix is a lower-block triangular matrix. Hence, the singularity analysis may be decomposed into two subproblems of the arm and the wrist singularity, coincident with the situation when they are controlled separately.

Condition when the singularity of the arm, driven by the first three joints, will happen is that

$$\det\ J_{11}=a_2\left(a_3s_3+d_4c_3\right)\left(a_2c_2+a_3c_{23}-d_4s_{23}\right)=0.$$

This will occur when $a_3s_3+d_4c_3=0$ or $a_2c_2+a_3c_{23}-d_4s_{23}=0$. Considering the robot kinematical structure in Figs. 3.7 and 3.8, the first case will hold if the robot arm stretches out or folds back the link a_2 and d_4 in a way that one can draw a line which passes joint axis#2, joint axis#3, and the wrist center point. If this happens, the point will not be able to move along that line. The second case will hold if the wrist center lies on the first joint axis, making it unable to move perpendicular to the plane formed by the link a_2 and d_4. The robot posture is similar to the one illustrated in Fig. 5.5.

On the other hand, condition when the singularity of the wrist, driven by the last three joints, will happen is that

$$\det\ J_{22}=-s_5=0.$$

When $\theta_5=0$ or π, this condition will holds. Physically, it makes joint axis#4 and joint axis#6 aligned. Thus the roll-pitch-roll wrist will be unable to rotate in the yaw direction, which is perpendicular to the plane formed by joint axis#5 and joint axis#6. Arrangement of the linkages resemble the one shown in Fig. 5.6.

In summary, the conditions which brings the PUMA 560 to one of its singular point are

- The link a_2 and d_4 stretch out or fold back in a way that there is a line passing through joint axis#2, joint axis#3, and the wrist center.
- The wrist center lies on the first joint axis.
- The wrist fully stretches out or folds back.

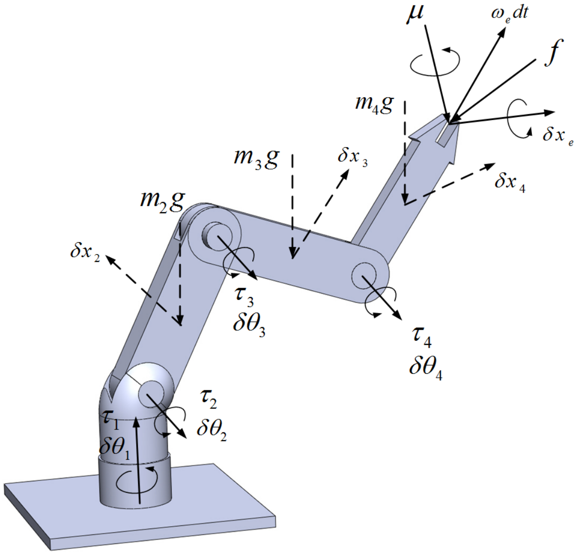


Figure 5.9: Relationship of the force acting onto the environment at the end effector and the actuator torques applied at the robot joints. ([1], pp. 230)

Locations of the wrist center for the first two cases are known priori and thus can be prevented. However, the last case of singularity can happen everywhere inside the workspace. Therefore, it must be checked whether $s_5 \approx 0$ at every sampling before proceeding. Moreover, the last two cases of singularities bring about infinitely many inverse kinematic solutions.

5.4 Static Force

In section 5.2, it is shown that the geometric Jacobian matrix governs the relationship between the joint velocity and the end effector velocity as summarized in Eq. 5.13. Nevertheless, recall the statement of the *principle of the virtual work*:

> The virtual work acting by the external active forces to the ideal mechanism under the equilibrium condition will be equal to zero.

From the principle, it can be shown that when the robot is in equilibrium, the Jacobian matrix also dictates the relationship between the force acting at the end effector and the supplied torque at the joints.

Consider Fig. 5.9 showing the ideal serial manipulator of which the friction may be omitted. In addition, if it is legitimate to neglect the gravity force or it has been compensated separately, the remaining external active forces acting on the robot will be merely the actuator torques at the joints and the force at the end effector.

Actuator torques may be grouped and written as the joint torque vector in $\mathbb{R}^n$;

$$\bar{\tau} = \left[\begin{array}{cccc} \tau_1 & \tau_2 & \cdots & \tau_n \end{array} \right]^T. \tag{5.17}$$

For the force acting onto the environment at the end effector, generally it is composed of the force $\bar{f}$ and the couple $\bar{\mu}$ which may be written as a vector in $\mathbb{R}^6$ as

$$\bar{F} = \left[\begin{array}{cccccc} f_x & f_y & f_z & \mu_x & \mu_y & \mu_z \end{array} \right]^T = \left[\begin{array}{cc} \bar{f}^T & \bar{\mu}^T \end{array} \right]^T. \tag{5.18}$$

When the external active forces $\bar{\tau}$ and $\bar{F}$ acting on the robot that is in equilibrium, assume the joints are displaced by an arbitrary deformation $\delta\bar{q}$ from the equilibria. Correspondingly, the displacement $\delta\bar{X} = \left[\begin{array}{cc} \delta\bar{x}^T & (\bar{\omega}dt)^T \end{array} \right]^T$ at the end effector admissible to the kinematic constraints is related to the joint displacement by

$$\delta\bar{X} = J\left(\bar{q}\right)\delta\bar{q}. \tag{5.19}$$

$\delta\bar{q}$ and $\delta\bar{X}$ are the virtual displacements because these displacement are guaranteed to satisfy only the kinematic constraints.

Virtual work is the work done by the force on the system according to the virtual displacement. If it is decided that the work done by the environment on the system is positive and the work done by the system on the environment is negative, the application of the principle of the virtual work when the robot is in equilibrium yields

$$\bar{\tau}^T\delta\bar{q} - \bar{F}^T\delta\bar{X} = 0.$$

With the robot kinematic constraints of Eq. 5.19, the above equation may be written as

$$\left(\bar{\tau}^T - \bar{F}^T J\right)\delta\bar{q} = 0.$$

Since the principle of the virtual work must hold for arbitrary virtual displacement, one may conclude that

$$\bar{\tau} = J^T\bar{F}. \tag{5.20}$$

It can be stated that the mapping from the end effector reaction force to the robot joint torque is determined by the transpose of the Jacobian matrix.

Of course, the relationship between the end effector force and the joint torque at the equilibrium may be derived by first drawing the free-body-diagram of all participating linkages and elements. Then the general equilibrium equations of

$$\begin{array}{rcl} \sum\bar{f} & = & \bar{0} \\ \sum\bar{\mu} & = & \bar{0} \end{array}$$

may be applied to develop the relationship between the forces acting on various parts of the robot. After eliminating the internal forces and grouping the equations altogether, the relationship between the external forces acting at the end effector and the joints may be attained.

Problems

1. Derive the Jacobian of the manipulator in Prob. 1 of Chapter 3 expressed in the base frame. Identify, if any, the singularity of this robot.

2. Derive the Jacobian of the robot arm in Prob. 2 of Chapter 3 expressed in the base and in the end effector frame. Determine a set of joint angles which bring the robot at its workspace boundary and workspace interior singularities.

3. Determine the Jacobian of the robot wrist in Prob. 3 of Chapter 3 expressed in the base frame and in the end effector frame. Which description is simpler? Analyze for the singularities.

4. Determine the Jacobian of the robot wrist in Prob. 4 of Chapter 3 expressed in the base frame and in the end effector frame. Which description is simpler? Analyze for the singularities.

5. A schematic diagram of the 2-DOF long-reach robot arm is shown in Fig. 5.10. The first link weighs 50 kg and the second link 10 kg. Their total lengths are 1.5 m and 4 m, respectively. At the depicted posture, where the first link is rotated by 110° and the second link is traveled by 3.5 m forward, determine the required torque and force of both actuators to maintain the configuration. For simplicity, let the C.G. of the links be at their mid-point.

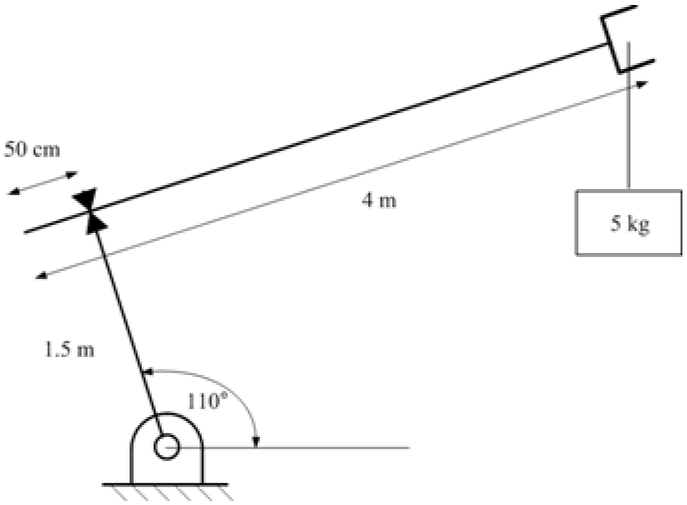


Figure 5.10: Schematic diagram of a 2-DOF long-reach arm.

Chapter 6

Trajectory Generation

In order to command the robot to move, one must have the desired motion in mind. In this chapter, several methods to generate the trajectory for the end effector to follow will be studied. The trajectory or path will have to satisfy a set of requirements. Typically they must pass or go through certain points with some particular velocity.

Section 6.1 explains some concepts and related terms. There are two main approaches in generating the trajectory. They are the joint space scheme and the Cartesian space scheme, named after in which space the trajectory are planned. Consequently, the next two sections describe each method in details.

6.1 Path Description

The term *trajectory* generally refers to a time history of position, velocity, and acceleration of the object motion. For the robot control point of view, it could be the trajectory of the joint or the end effector motion. Nevertheless, common users will provide a set of points or postures through which the end effector should pass. Then, from this specification it must be the computer program that create the trajectory. Usually, points along the trajectory will be computed at a certain rate depending on the control loop update rate.

The simplest specification would be to give the initial and final posture the end effector must be. Homogeneous transformation description might be used. Hence T_i and T_f are specified. During the way, how the end effector move depends on the generated trajectory.

Often, it requires more than merely where the end effector should be finally. It may be necessary that the end effector be in a certain posture with some velocity and/or at a specified time during the motion to accomplish some tasks, e.g. the assembly task. Therefore the user might need to specify a sequence of desired postures, called *via points*, between the initial and final ones. Collection of these points may be called *path points* to include both the initial and final points as well.

6.2 Joint Space Schemes

With the given path points, the joint space schemes generate the robot trajectory at the joint level. Hence, firstly the joint angles which cause the end effector to be at the specified set of postures must be computed. This can be achieved by the inverse kinematics calculation. Accordingly, for each joint, a smooth function is determined that passes through the stream of discrete points of the joint angles. Beware the time spent during the same segment of each joint must be the same for the end effector to actually meet the specified via points.

By this implementation, the joint space schemes yield the desired end effctor pose at the via points. Shape of the motion during the via points at joint level is described by the functions used to fit the via points. However the shape of the corresponding motion of the end effector is not put in the design requirement and hence might turn out to be arbitrarily complex. The drawback may be resolved naturally by simply introducing more constrained via points during the desired path. This will shorten the length of each segment of the joint trajectory. As a result, segments of the end effector trajectory will be shorten as well so that each of them joining the adjacent via points may be described by a linear interpolating function.

The joint space schemes have some advantages. First and foremost, it is easier to generate the trajectory in joint space. Computational cost is much less compared to the Cartesian space schemes mainly because the inverse kinematics will be performed at the via points only. Another important issue is there will be no singularity problem if the trajectory is constructed in the joint space. Singularity is the problem in which the inverse kinematics function fails at certain points in the workspace. Of course, the user must not give the via point coincident with the singularity point.

In the following, typical methods used to generate the trajectory given the via points under some constraints are considered. The simplest case of two path points will be considered first.

6.2.1 Trajectory Generation for Two Path Points

The problem here is to move the end effector from its initial to final pose by the specified time interval. Usually, inverse kinematics calculation yielding the corresponding joint angles will be performed first. Then, a smooth function $\theta(t)$ for each joint which passes through its initial and final joint angles, θ_i and θ_f, at the time instant t_i and t_f, will be determined.

6.2.1.1 Cubic Polynomial Function

However, there are many such functions and so the trajectories which satisfy the given conditions as depicted in Fig. 6.1. Typically, at the two end points, both position and velocity are specified;

$$\begin{array}{ll} \theta(t_i) = \theta_i & \theta(t_f) = \theta_f \\ \dot{\theta}(t_i) = \omega_i & \dot{\theta}(t_f) = \omega_f. \end{array} \tag{6.1}$$

From these requirements, one may use the cubic polynomial function

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \tag{6.2}$$

as the trajectory. Substituting the position constraints into the equation,

$$\begin{aligned} \theta_i &= a_0 + a_1 t_i + a_2 t_i^2 + a_3 t_i^3 \\ \theta_f &= a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 \end{aligned}$$

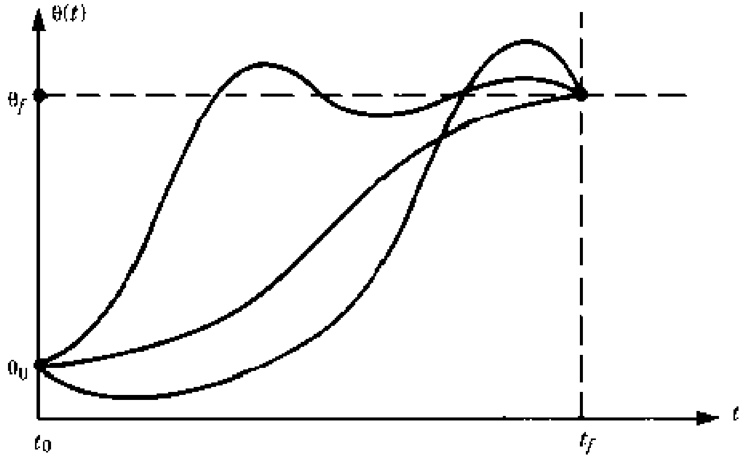


Figure 6.1: Several possible trajectories for a single joint which satisfy the ending joint values. ([4], pp. 204)

must hold. Analogously, the velocity conditions call for

$$\begin{aligned}\omega_i &= a_1 + 2a_2t_i + 3a_3t_i^2 \\ \omega_f &= a_1 + 2a_2t_f + 3a_3t_f^2,\end{aligned}$$

where the velocity equation is obtained by taking the time derivative of Eq. 6.2:

$$\omega\left(t\right) = a_1 + 2a_2t + 3a_3t^2. \tag{6.3}$$

The above four equations may then be used to solve for the coefficients of the cubic polynomial function. For the special case of zero initial and final velocity and setting $t_i = 0$, simplified analytical expressions may be determined as

$$\begin{aligned}a_0 &= \theta_i \\ a_1 &= 0 \\ a_2 &= \frac{3}{t_f^2}\left(\theta_f - \theta_i\right) \\ a_3 &= -\frac{2}{t_f^3}\left(\theta_f - \theta_i\right).\end{aligned} \tag{6.4}$$

It should be mentioned that since the trajectory function is cubic, the velocity function according to the coefficients in Eq. 6.4 will be a parabola maximally/minimally at the midpoint and the acceleration will be a linearly decreasing/increasing function having zero value at the midpoint.

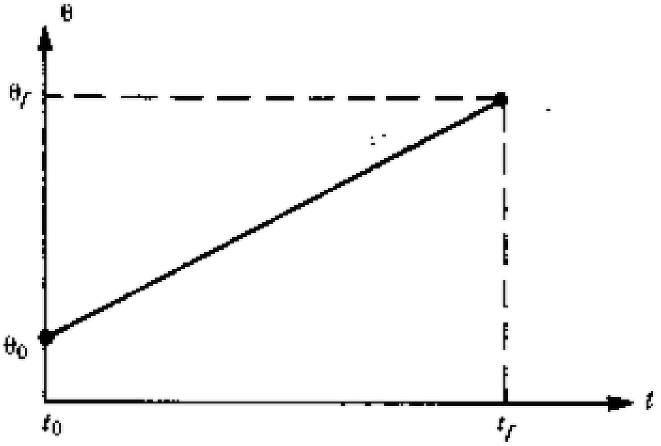


Figure 6.2: Linear trajectory connecting the intial and final joint angles. ([4], pp. 210)

6.2.1.2 Quintic Polynomial Function

If additional constraints on acceleration of

$$\ddot{\theta}\left(t_i\right) = \alpha_i \qquad \ddot{\theta}\left(t_f\right) = \alpha_f \tag{6.5}$$

are specified, a fifth-order or quintic polynomial function

$$\theta\left(t\right) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \tag{6.6}$$

is necessary. Coefficients of the equations may be solved in the same manner with the velocity and acceleration profile of

$$\omega\left(t\right) = a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4 \tag{6.7}$$

$$\alpha\left(t\right) = 2a_2 + 6a_3 t + 12a_4 t^2 + 20a_5 t^3. \tag{6.8}$$

However the analytical expressions will be more complicated.

6.2.1.3 Linear Function with Parabolic Blends

Yet another way to generate a function for the trajectory that passes through the given end points is to use the linear segment function as shown in Fig. 6.2. However, it will make the velocity to be discontinuous at the end points, which leads to an impulsive acceleration that can deteriorate the tracking result. To fix this problem, a parabolic function is introduced at the ends to smoothen the path with continuous position and velocity. In other words, the linear segment trajectory is blended with the parabolic function.

A typical linear function with parabolic blends trajectory and its velocity and acceleration profile are illustrated in Fig. 6.3. During the blending period, the

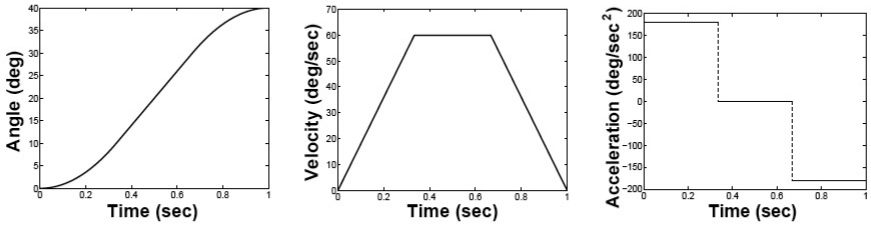


Figure 6.3: Linear function with parabolic blends trajectory and its velocity and acceleration profile. ([3], pp. 195)

associated velocity will be changed linearly with a constant acceleration. In the linear segment portion, the motion will be executed with a constant velocity.

To construct such trajectory, typically one first needs to specify the initial and final joint angles, θ_i and θ_f, the according velocities, $\dot{\theta}_i$ and $\dot{\theta}_f$, and the time duration, $t_f - t_i$, of the motion. Without loss of generalization, a special case of zero initial and final velocity and setting $t_i = 0$ will be considered.

Let t_b be the symmetric blending time. Hence, during the first blend, the motion is described by the parabolic function of

$$\theta\left(t\right) = \theta_i + \frac{1}{2}\alpha t^2, \quad 0 \leq t \leq t_b \tag{6.9}$$

where α is the constant acceleration. At time $t = t_b$, equality of the velocity at the end of the first blend and at the beginning of the constant velocity, V, segment requires the relationship of the following parameters;

$$\alpha t_b = V. \tag{6.10}$$

During the linear segment portion, the motion is thus described by the linear function of

$$\theta\left(t\right) = \theta\left(t_b\right) + V\left(t - t_b\right), \quad t_b < t \leq t_f - t_b.$$

To determine $\theta\left(t_b\right)$, the following equation

$$\theta\left(\frac{t_f}{2}\right) = \theta\left(t_b\right) + V\left(\frac{t_f}{2} - t_b\right) = \frac{\theta_i + \theta_f}{2}$$

holds due to the symmetry of the trajectory. Thus,

$$\theta\left(t_b\right) = \frac{\theta_i + \theta_f - Vt_f}{2} + Vt_b, \tag{6.11}$$

resulting in the linear segement trajectory of

$$\theta\left(t\right) = \frac{\theta_i + \theta_f - Vt_f}{2} + Vt, \quad t_b < t \leq t_f - t_b. \tag{6.12}$$

Moreover, the trajectory of Eq. 6.9 and 6.12 must blend at t_b. Thus the blending time may be written as a function of the specified end conditions and the constant velocity:

$$t_b = \frac{\theta_i - \theta_f + V t_f}{V}. \tag{6.13}$$

Furthermore, since $0 < t_b \leq \frac{t_f}{2}$, the desired constant velocity value must be in between

$$\frac{\theta_f - \theta_i}{t_f} < V \leq 2\frac{\theta_f - \theta_i}{t_f} \tag{6.14}$$

or the motion is not possible.

The blending time may also be expressed as a function of the acceleration. Rewrite Eq. 6.13 by making use of Eq. 6.10, the relationship between the blending time and the acceleration is governed by the quadratic

$$\alpha t_b^2 - \alpha t_f t_b + \theta_f - \theta_i = 0.$$

Consequently, the blend time may be determined by

$$t_b = \frac{t_f}{2} - \frac{\sqrt{\alpha^2 t_f^2 - 4\alpha\left(\theta_f - \theta_i\right)}}{2\alpha}. \tag{6.15}$$

Analogously, it can be shown that since $0 < t_b \leq \frac{t_f}{2}$, the following condition

$$0 \leq \frac{\sqrt{\alpha^2 t_f^2 - 4\alpha\left(\theta_f - \theta_i\right)}}{\alpha} < t_f$$

must hold. This implies the desired constant acceleration value

$$\alpha \geq \frac{4\left(\theta_f - \theta_i\right)}{t_f^2} \tag{6.16}$$

must be satisfied.

Finally, during the last blend, the motion is again described by the parabolic function of

$$\theta\left(t\right) = \left(\theta_f - \theta\left(t_b\right)\right) + V\left(t - t_f + t_b\right) - \frac{1}{2}\alpha\left(t - t_f + t_b\right)^2, \quad t_f - t_b < t \leq t_f.$$

Substituting the expression of α and $\theta\left(t_b\right)$ from Eq. 6.10 and 6.11 into the above equation and performing some reduction, $\theta\left(t\right)$ may be expressed as

$$\theta\left(t\right) = \theta_f - \frac{1}{2}\alpha t_f^2 + \alpha t_f t - \frac{1}{2}\alpha t^2 - \frac{\theta_i + \theta_f - V\left(t_f - t_b\right)}{2}.$$

Applying the boundary angle condition θ_f at time t_f,

$$\frac{\theta_i + \theta_f - V\left(t_f - t_b\right)}{2} = 0$$

may be concluded. Thus, the trajectory in the last segment may be described by

$$\theta(t) = \theta_f - \frac{1}{2}\alpha t_f^2 + \alpha t_f t - \frac{1}{2}\alpha t^2, \quad t_f - t_b < t \leq t_f. \tag{6.17}$$

Example 6.1 A single-link robot with a rotary joint is motionless at $\theta = 15°$. It is desired to move the joint in a smooth manner to $\theta = 75°$ in 3 seconds. Determine the linear function with parabolic blends trajectory if the velocity of the linear segment is chosen to be 30°/sec. What will be the new trajectory if the acceleration of the parbolic blends is specified to be 40°/sec^2 instead.

Solution The blending time according to the given velocity may be calculated from Eq. 6.13 as

$$t_b = \frac{15 - 75 + 30 \times 3}{30} = 1\,\text{sec.}$$

Acceleration during the parabolic blends must obey Eq. 6.10. Thus,

$$\alpha = \frac{30}{1} = 30°/\text{sec}^2.$$

Therefore, the linear function with parabolic blends for the first case may be determined from Eqs. 6.9, 6.12, and 6.17;

$$\theta(t) = \begin{cases} 15 + 15t^2, & 0 \leq t \leq 1 \\ 30t, & 1 < t \leq 2 \\ -60 + 90t - 15t^2, & 2 < t \leq 3. \end{cases}$$

For the case where the acceleration is specified, the blending time may now be computed from Eq 6.15 as

$$t_b = \frac{3}{2} - \frac{\sqrt{40^2 \times 3^2 - 4 \times 40 \times (75 - 15)}}{2 \times 40} = 0.6340\,\text{sec.}$$

Velocity during the linear segment must obey Eq. 6.10. Thus,

$$V = 40 \times 0.6340 = 25.3590°/\text{sec.}$$

As in the first case, the linear function with parabolic blends for the second case may be determined;

$$\theta(t) = \begin{cases} 15 + 20t^2, & 0 \leq t \leq 0.6340 \\ 6.9615 + 25.3590t, & 0.6340 < t \leq 2.3660 \\ -105 + 120t - 20t^2, & 2.3660 < t \leq 3. \end{cases}$$

Plots of the angle, velocity, and acceleration for both cases are displayed in Fig. 6.4 and 6.5 respectively.

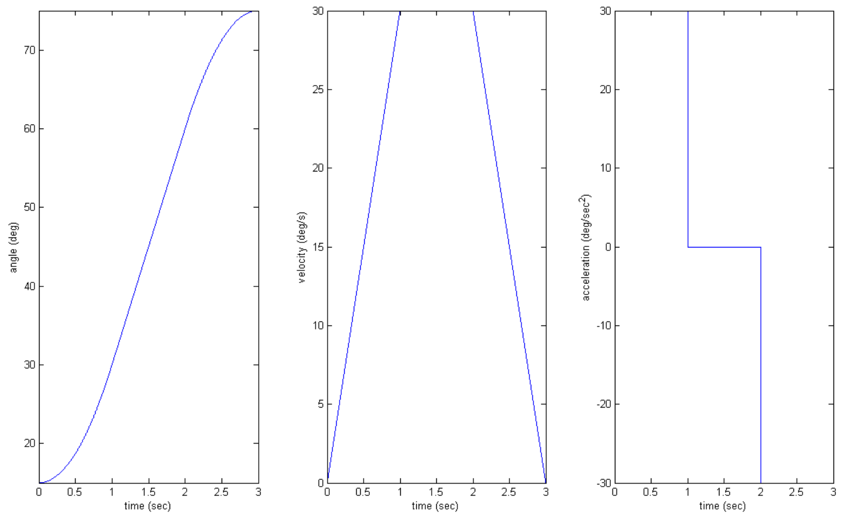


Figure 6.4: Linear function with parabolic blends trajectory and its velocity and acceleration profile for the first case.

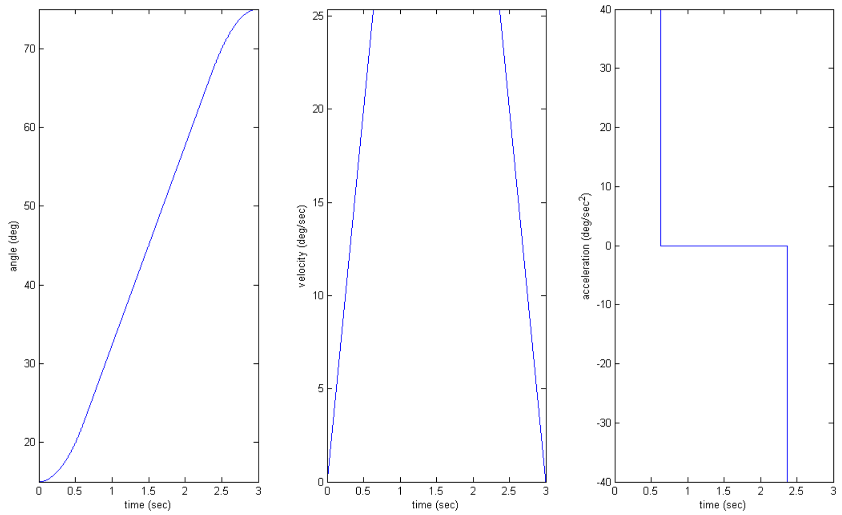


Figure 6.5: Linear function with parabolic blends trajectory and its velocity and acceleration profile for the second case.

6.2.2 Trajectory Generation for a Path with Via Points

As explained in section 6.1, the user may need to specify a sequence of path points rather than merely two end points. Therefore the generated trajectory is expected to pass through these via points as well. If the end effector is to come to rest during each visited via point, the solutions in subsection 6.2.1 may be applied immediately. In general, it is required that the end effector passes through the via points without stopping. Therefore, extensions of the previous derivations are necessary.

6.2.2.1 Cubic Polynomial Function

If the desired velocities of the joints at the via points are specified, it is simple to construct segments of cubic polynomial trajectory which connect two neighboring via points. At a particular segment, the following constraints

$$\begin{aligned} \theta\left(t_i\right) &= \theta_i & \theta\left(t_f\right) &= \theta_f \\ \dot{\theta}\left(t_i\right) &= \omega_i & \dot{\theta}\left(t_f\right) &= \omega_f \end{aligned} \tag{6.18}$$

must be satisfied. Using the cubic polynomial function of Eq. 6.2, the above constraints may be substituted to obtain

$$\begin{aligned} \theta_i &= a_0 \\ \theta_f &= a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3, \end{aligned}$$

where for each via point, the initial time is reset to zero, i.e. $t_i = 0$. This implies the final time t_f is the time duration spent in that segment. Similarly, applying the velocity constraints to the velocity equation of the trajectory in Eq. 6.3, the following equations

$$\begin{aligned} \omega_i &= a_1 \\ \omega_f &= a_1 + 2a_2 t_f + 3a_3 t_f^2 \end{aligned}$$

may be formed. Solving the above four equations for the coefficients, one then have

$$\begin{aligned} a_0 &= \theta_i \\ a_1 &= \omega_i \\ a_2 &= \frac{3}{t_f^2}\left(\theta_f - \theta_i\right) - \frac{2}{t_f}\omega_i - \frac{1}{t_f}\omega_f \\ a_3 &= -\frac{2}{t_f^3}\left(\theta_f - \theta_i\right) + \frac{1}{t_f^2}\left(\omega_f + \omega_i\right). \end{aligned} \tag{6.19}$$

In practice, specifying the velocity at each via point may not be intuitive to the user. Therefore, the path generator system should have the capability to

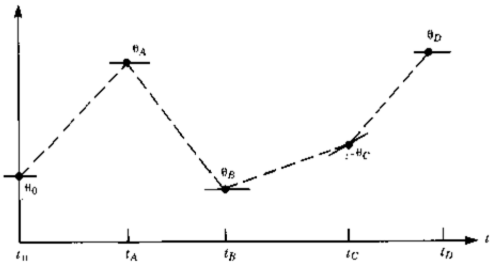


Figure 6.6: Heuristic decision of assigning the velocity at the via points for the cubic polynomial function trajectory. ([4], pp. 208)

compute the appropriate via point velocity autonomously. The following simple heuristic algorithm may be used in selecting the velocity. Consider the path points of a robot joint in Fig. 6.6. Imagine the via points connected with straight line segments. If the slope of the line changes sign at the via point, the zero velocity is chosen. If the sign does not change, the velocity at the via point is selected to be the average of the two slopes. These velocities are then to be used as the velocity constraints at the via point as before.

Another approach for this issue is not to pay attention to the specific velocity values at the via points. Rather, the new constraints that both the velocity and acceleration be continuous are employed. As a result, the trajectory for a segment will depend on the adjacent ones as well. Fortunately, the equations arising from applying these via point constraints may be represented in the matrix-vector equation. Tri-diagonal form of the matrix expedites the computation of the coefficients of the cubic polynomial trajectories.

6.2.2.2 Higher-Order Polynomial Function

If additional constraints on acceleration at the via points are specified, a higher-order polynomial function is needed since it has more room to satisfy all of them. In a similar vein to the cubic polynomial function, work of using the quintic polynomial in subsection 6.2.1.2 for generating a path joining two end points may be extended for the general case of given path points.

A drastically different approach in trajectory generation for the path points is to solve for a single high-order polynomial function which passes through all points with possibly additional velocity and acceleration constraints. Order of the polynomial depends on the total number of the constraints to be satisfied.

For example, motion of a joint is required to pass through θ_0, θ_1, and θ_2 at time t_0, t_1, and t_2. In addtion, the initial and final velocity and acceleration of ω_0, ω_2, α_0, and α_2 are specified. Hence the constrained equations to be formulated

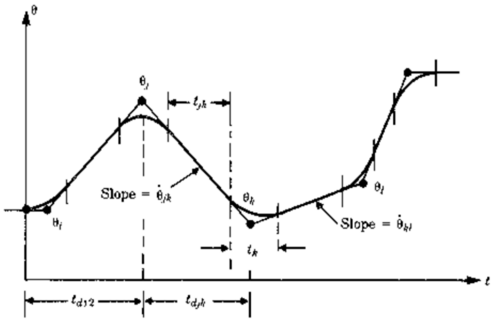


Figure 6.7: Multisegment linear function with parabolic blends trajectory for the specified path points and time durations. ([4], pp. 214)

are

$$\begin{aligned} &\theta_0\theta\left(t_0\right) \qquad &&\theta_1=\theta\left(t_1\right) \qquad &&\theta_2=\theta\left(t_2\right) \\ &\omega_0=\dot{\theta}\left(t_0\right) && &&\omega_2=\dot{\theta}\left(t_2\right) \\ &\alpha_0=\ddot{\theta}\left(t_0\right) && &&\alpha_2=\ddot{\theta}\left(t_2\right). \end{aligned}$$

The determinate polynomial function which satisfies these constraints is a sixth order polynomial

$$\theta\left(t\right)=a_0+a_1t+a_2t^2+a_3t^3+a_4t^4+a_5t^5+a_6t^6,$$

of which its coefficients may be solved uniquely.

An advantage of this approach is that the trajectory, and its velocity and acceleration, is intrinsically continuous at the via points. A big disadvantage will be seen when many via points and/or constraints are specified. This introduces a large system of linear equations to be solved.

6.2.2.3 Linear Function with Parabolic Blends

Linear function with parabolic blends for two end points may be modified to serve for the case when the path points are specified. Figure 6.7 depicts the smooth trajectory which fits the via points. Qualitatively, straight lines are constructed to join the via points first. Then parabolic blends are employed at each via point to round the path. Effectively, the resulting blended path will not pass through the via points. It only comes close to the point. Resolution to this issue will be discussed later.

According to Fig. 6.7, consider three neighboring via points θ_j, θ_k, and θ_l. Time duration of the segment connecting points j and k, t_{djk}, is specified. However, time duration during the blend region at point k, t_k, and time duration of the linear segment between points j and k, t_{jk}, are not specified. Corresponding constant acceleration and velocity are denoted as $\ddot{\theta}_k$ and $\dot{\theta}_{jk}$.

If the magnitude of the acceleration to use at each via point $\left|\ddot{\theta}_k\right|$ is specified, the following computation may be used to determine the blend times, the linear times, and the linear velocities at each segment of the connecting trajectory. These parameters will be used to generate the trajectory and its derivatives. Their streams of discrete values will be computed at the path update rate and fed as a reference input to the robot controller.

Refer to Fig. 6.7. For the first segment, the blend time t_1 may be solved by recognizing the velocity at the end of the blending portion must be equal to that during the linear segment;

$$\begin{aligned} \ddot{\theta}_1 &= \operatorname{sgn}(\theta_2 - \theta_1)\left|\ddot{\theta}_1\right| && (6.20) \\ \ddot{\theta}_1 t_1 &= \frac{\theta_2 - \theta_1}{t_{d12} - \frac{1}{2}t_1} \\ t_1 &= t_{d12} - \sqrt{t_{d12}^2 - \frac{2(\theta_2 - \theta_1)}{\ddot{\theta}_1}}. && (6.21) \end{aligned}$$

Hence, the linear velocity of the first segment follows immediately;

$$\dot{\theta}_{12} = \frac{\theta_2 - \theta_1}{t_{d12} - \frac{1}{2}t_1}. \qquad (6.22)$$

For the interior via points, the computation of trajectory parameters is performed with the following equations;

$$\begin{aligned} \dot{\theta}_{jk} &= \frac{\theta_k - \theta_j}{t_{djk}} && (6.23) \\ \dot{\theta}_{kl} &= \frac{\theta_l - \theta_k}{t_{dkl}} \\ \ddot{\theta}_k &= \operatorname{sgn}\left(\dot{\theta}_{kl} - \dot{\theta}_{jk}\right)\left|\ddot{\theta}_k\right| && (6.24) \\ t_k &= \frac{\dot{\theta}_{kl} - \dot{\theta}_{jk}}{\ddot{\theta}_k} && (6.25) \\ t_{jk} &= t_{djk} - \frac{1}{2}t_j - \frac{1}{2}t_k. && (6.26) \end{aligned}$$

Note from Fig. 6.7 that only half of the blend region of each via point enter the segment. Two end points of the entire path are exceptional where the full blend region must be counted. Another point is that the linear time of the first segment, t_{12}, can be computed after the blend time of the second point t_2 is known. That is

$$\begin{aligned}
\dot{\theta}_{23} &= \frac{\theta_3 - \theta_2}{t_{d23}} \\
\ddot{\theta}_2 &= \operatorname{sgn}\left(\dot{\theta}_{23} - \dot{\theta}_{12}\right)\left|\ddot{\theta}_2\right| \\
t_2 &= \frac{\dot{\theta}_{23} - \dot{\theta}_{12}}{\ddot{\theta}_2} \\
t_{12} &= t_{d12} - t_1 - \frac{1}{2}t_2. \qquad (6.27)
\end{aligned}$$

Intermediate segments are thus successively calculated using Eqs. 6.23 to 6.26. Finally, the last segment connecting θ_{n-1} and θ_n is reached. Its related parameters may be computed from matching the velocity of the linear segment with the starting velocity of the blending portion, assuming the zero ending velocity;

$$\begin{aligned}
\ddot{\theta}_n &= \operatorname{sgn}\left(\theta_{n-1} - \theta_n\right)\left|\ddot{\theta}_n\right| \qquad (6.28) \\
\frac{\theta_n - \theta_{n-1}}{t_{d(n-1)n} - \frac{1}{2}t_n} &= 0 - \ddot{\theta}_n t_n \\
t_n &= t_{d(n-1)n} - \sqrt{t^2_{d\{n-1\}n} + \frac{2\left(\theta_n - \theta_{n-1}\right)}{\ddot{\theta}_n}} \qquad (6.29) \\
\dot{\theta}_{(n-1)n} &= \frac{\theta_n - \theta_{n-1}}{t_{d(n-1)n} - \frac{1}{2}t_n} \qquad (6.30) \\
t_{(n-1)n} &= t_{d(n-1)n} - \frac{1}{2}t_{n-1} - t_n, \qquad (6.31)
\end{aligned}$$

where t_{n-1} is determined from Eqs. 6.25 and 6.24.

Example 6.2 The trajectory of a particular joint is specified as follow. Path points in degrees are 10, 60, 90, 75. Time spent during the segments are 2, 1, 3 seconds. For simplicity, the default acceleration of $50°/\text{sec}^2$ is to be used for all blendings. Calculate all segment velocities, blend times, and linear times. Generate the trajectory and its velocity and acceleration at a rate of 1 kHz.

SOLUTION The computation may be performed straightforwardly following a series of the above formulas. Particularly,

$$\begin{aligned}
\ddot{\theta}_1 &= \operatorname{sgn}(60-10)\times 50 = 50^\circ/\mathrm{sec}^2 \\
t_1 &= 2-\sqrt{2^2-\frac{2\,(60-10)}{50}} = 0.5858\,\mathrm{sec} \\
\dot{\theta}_{12} &= \frac{60-10}{2-0.5\times 0.5858} = 29.2893^\circ/\mathrm{sec} \\
\dot{\theta}_{23} &= \frac{90-60}{1} = 30^\circ/\mathrm{sec} \\
\ddot{\theta}_2 &= \operatorname{sgn}(30-29.2893)\times 50 = 50^\circ/\mathrm{sec}^2 \\
t_2 &= \frac{30-29.2893}{50} = 0.0142\,\mathrm{sec} \\
t_{12} &= 2-0.5858-0.5\times 0.0142 = 1.4071\,\mathrm{sec} \\
\ddot{\theta}_4 &= \operatorname{sgn}(90-75)\times 50 = 50^\circ/\mathrm{sec}^2 \\
t_4 &= 3-\sqrt{3^2+\frac{2\,(75-90)}{50}} = 0.1017\,\mathrm{sec} \\
\dot{\theta}_{34} &= \frac{75-90}{3-0.5\times 0.1017} = -5.0862^\circ/\mathrm{sec} \\
\ddot{\theta}_3 &= \operatorname{sgn}(-5.0862-30)\times 50 = -50^\circ/\mathrm{sec}^2 \\
t_3 &= \frac{-5.0862-30}{-50} = 0.7017\,\mathrm{sec} \\
t_{23} &= 1-0.5\times 0.0142-0.5\times 0.7017 = 0.6420\,\mathrm{sec} \\
t_{34} &= 3-0.5\times 0.7017-0.1017 = 2.5474\,\mathrm{sec.}
\end{aligned}$$

From these parameters, the trajectory generated by segments of the linear function with parabolic blends may be written explicitly as follow;

$$\theta(t) = \begin{cases}
10+25t^2 & 0\le t\le 0.585 \\
18.5556+29.2893\,(t-0.585) & 0.585<t\le 1.993 \\
59.7950+29.2893\,(t-1.993)+25\,(t-1.993)^2 & 1.993<t\le 2.007 \\
60.2099+30\,(t-2.007) & 2.007<t\le 2.649 \\
79.4699+30\,(t-2.649)-25\,(t-2.649)^2 & 2.649<t\le 3.351 \\
88.2098-5.0862\,(t-3.351) & 3.351<t\le 5.898 \\
75.2553-5.0862\,(t-5.898)+25\,(t-5.898)^2 & 5.898<t\le 6.
\end{cases}$$

Plots of the trajectory, velocity, and acceleration generated with the time step of 0.001 sec are shown in Fig. 6.8.

As mentioned earlier, the blending process causes the trajectory not passing through the via points unless the robot come to a stop. If a pause is acceptable, one merely insert the repeated via point in the path points specification. Nevertheless, the path can come pretty close to the points if high blending acceleration is employed.

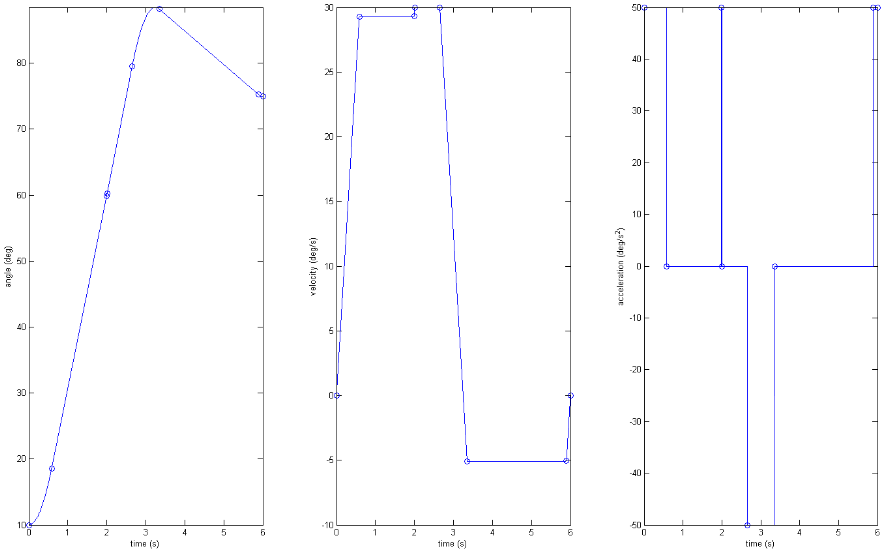


Figure 6.8: Trajectory generated by segments of the linear function with parabolic blend and its velocity and acceleration profile for the specified path points, time duration, and the blending acceleration. The 'o' marks indicate the segment transition point.

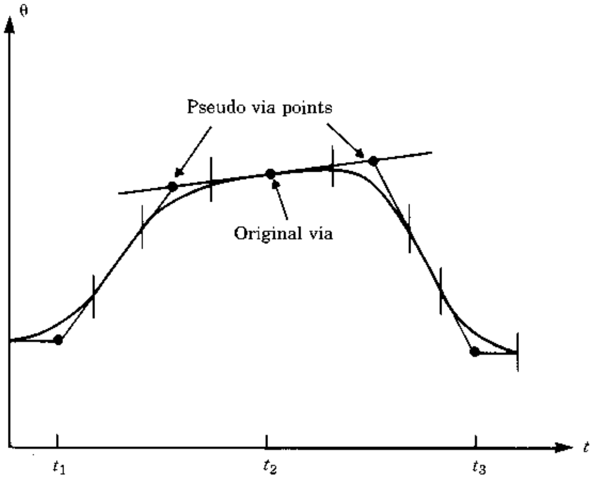


Figure 6.9: Use of pseudo via points to create a trajectory which passes through the original via with specified velocity. ([4], pp. 217)

Sometimes it is required that the motion passes through the via point without stopping. This can be achieved by replacing that via point with two pseudo via points, one on each side of the original. See Fig. 6.9. Placement of the pseudo via points are such that the original via point lies in the linear segment of the path connecting them. By this method, the designer may also impose the velocity used to pass through that point. The term *through point* might be named for a path point through which the manipulator is forced to pass exactly.

6.3 Cartesian Space Schemes

Cartesian space schemes generate the trajectory of the robot end effector directly in the Cartesian space or the task space. From the path planner viewpoint, it is natural and makes more sense to generate the trajectory by this scheme since the path points are visually specified in the task space. As a result, any desired path, such as the straight line, circular, or sinusoidal motion during the via points may be established.

Opposite to the joint space schemes, the Cartesian space one has a drawback of heavy inverse kinematics computation and its relates at every update time to bring the generated Cartesian coordinates and their derivatives to the joint space angles and their associates. Furthermore, the generated trajectory in task space may pass through some robot singular points incidentally, resulting in the failure of the mappings to the joint space.

Several Cartesian space schemes for the trajectory generation have been proposed. In the following, a scheme to generate a straight line end effector translational motion connecting two points in the workspace and simultaneously provide the smooth rotational transition between the specified orientations will be studied.

The first step of this scheme is to describe the specified task space path points using the 6-tuples consisting of the position vector, $\bar{p}$, for the position and the angle/axis representation, $\left(\theta, \hat{k}\right)$, for the orientation. Namely for each path point,

$$^{B}\bar{X}_E = \left[\begin{array}{cccccc} p_x & p_y & p_z & \theta k_x & \theta k_y & \theta k_z \end{array}\right]^T \tag{6.32}$$

dictates the position and the orientation of the specified $\{E\}$ with respect to $\{B\}$.

Concerning of the translational motion, to construct a straight line trajectory connecting two consecutive via points, it is readily seen that each coordinate must change linearly between two end values. Consequently, linear functions with parabolic blends may be applied to each coordinate of the position vector of the path points to generate segments of the end effector trajectory.

However, this notion does not work for the rotational motion since $\left[\begin{array}{ccc} \theta k_x & \theta k_y & \theta k_z \end{array}\right]^T$-tuple is not a vector. Nonethelesss, for simplicity in the calculation, one might want to apply the same interpolation scheme to all coordinates of the tuple. As a result, the rotational motion is not obtained by the rotation about the unique axis. This might cause the motion be unnatural. Note an issue of non-unique angle/axis representation, i.e.

$$\left(\theta, \hat{k}\right) = \left(\theta + 2\pi n, \hat{k}\right),$$

may be troublesome to the trajectory generation between two orientations ${}^{S}_{1}R$ and ${}^{S}_{2}R$. Typically, the angle/axis representation should be chosen such that

$$\left\| \begin{bmatrix} \theta_2 \left(k_x\right)_2 - \theta_1 \left(k_x\right)_1 \\ \theta_2 \left(k_y\right)_2 - \theta_1 \left(k_y\right)_1 \\ \theta_2 \left(k_z\right)_2 - \theta_1 \left(k_z\right)_1 \end{bmatrix} \right\|$$

is minimized. With this selection, the amount of rotational motion will be minimized.

Similar to the path generation in the joint space, the blending and the linear time spent during the same segment of each coordinate must be the same to ensure that the robot motion will follow a straight line in space (during the linear segments). Consequently, to meet this requirement, the calculation steps of the linear function with parabolic blends trajectory should be modified. Rather than specifying the blending acceleration, the blending time should be assigned and the acceleration be computed from the relationship. Mathematically, at the interior via points with the assigned t_k, a set of the following equations

$$\begin{aligned} \dot{\theta}_{jk} &= \frac{\theta_k - \theta_j}{t_{djk}} && (6.33) \\ \dot{\theta}_{kl} &= \frac{\theta_l - \theta_k}{t_{dkl}} \\ \ddot{\theta}_k &= \frac{\dot{\theta}_{kl} - \dot{\theta}_{jk}}{t_k} && (6.34) \\ t_{jk} &= t_{djk} - \frac{1}{2} t_j - \frac{1}{2} t_k && (6.35) \end{aligned}$$

should be evaluated instead. The required acceleration should be checked that it does not exceed the maximum value or the blending period might be prolonged.

Example 6.3 As a part of the particular task execution of PUMA 560 manipulator, the end effector is required to pass through the sequence of path points (in mm) with respect to the origin of $\{B\}$:

$$\bar{p}_1 = \begin{bmatrix} 500 \\ 500 \\ 500 \end{bmatrix}, \qquad \bar{p}_2 = \begin{bmatrix} 500 \\ 500 \\ 600 \end{bmatrix}, \qquad \bar{p}_3 = \begin{bmatrix} 400 \\ 500 \\ 600 \end{bmatrix}$$

$$\bar{p}_4 = \begin{bmatrix} 400 \\ 500 \\ 500 \end{bmatrix}, \qquad \bar{p}_5 = \begin{bmatrix} 400 \\ 600 \\ 400 \end{bmatrix}, \qquad \bar{p}_6 = \begin{bmatrix} 500 \\ 500 \\ 500 \end{bmatrix}.$$

Corresponding orientations are specified by the rotation matrix of $\{E\}$ described in $\{B\}$:

$$R_1 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \qquad R_2 = \begin{bmatrix} 1/\sqrt{2} & 0 & 1/\sqrt{2} \\ 0 & -1 & 0 \\ 1/\sqrt{2} & 0 & -1/\sqrt{2} \end{bmatrix},$$

$$R_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \qquad R_4 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1/\sqrt{2} & 1/\sqrt{2} \\ 0 & -1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix},$$

$$R_5 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}, \qquad R_6 = \begin{bmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}.$$

Planned trajectory should bring the end effector from the first to the second via point by 5 seconds. Then it is required to stop at the second via for 3 seconds. Time spent during each next three segments should be 4 seconds. However, the robot must pass through the fourth point precisely with the speed of 30 mm/s. Finally, the motion to the goal point during the last segment should be completed in 5 seconds. The time used for each blending is 1 second.

Dimensions of the PUMA arm are such that $a_2 = 431.8$, $a_3 = 20.3$, $d_3 = 150.1$, $d_4 = 433.1$, $H = 700$, $e = 100$ mm. Generate the trajectory using the simple Cartesian straight line motion scheme at a rate of 1 kHz. Investigate the velocity and acceleration as well.

Solution The given path points must be modified to fulfill the requirements. Since the robot must stay immobile for 3 seconds at the second via point, a repeated via point of the second one, point $2'$, is introduced. Another modification is at the fourth point where the robot is required to pass through exactly. A strategy is to replace this via point with two pseudo via points, points $4'$ and $4''$, of which their values are determined from the constraint on through speed. Figure 6.10 display the plot of the conceptual path points and the generated trajectory versus time of this problem.

Specifically, the time used for each segment may be chosen as

$$\begin{aligned} &t_1 = 1, \quad t_{12} = 3.5, \quad t_2 = 1, \quad t_{22'} = 2, \quad t_{2'} = 1 \\ &t_{2'3} = 3, \quad t_3 = 1, \quad t_{34'} = 1, \quad t_{4'} = 1, \quad t_{4'4''} = 3, \\ &t_{4''} = 1, \quad t_{4''5} = 1, \quad t_5 = 1, \quad t_{56} = 3.5, \quad t_6 = 1, \end{aligned}$$

for which the new points $4'$ and $4''$ are assigned at the half time between point 3-4 and 4-5, respectively. Accordingly, the modified path points might be designed

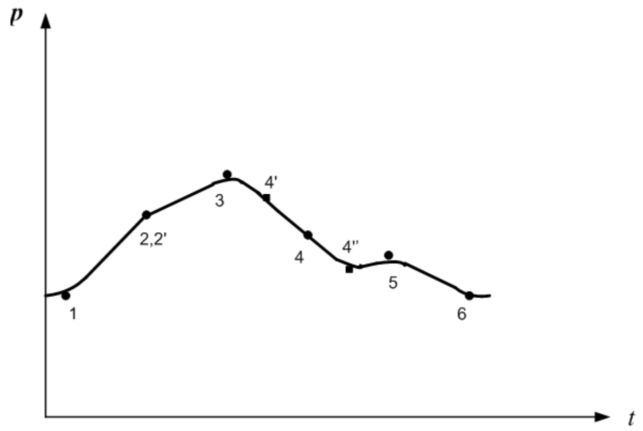


Figure 6.10: Conceptual path points and the trajectory vs. time of Ex. 6.3.

as follow;

$$\bar{p}_1 = \begin{bmatrix} 500 \\ 500 \\ 500 \end{bmatrix}, \qquad \bar{p}_2 = \begin{bmatrix} 500 \\ 500 \\ 600 \end{bmatrix}, \qquad \bar{p}_{2'} = \begin{bmatrix} 500 \\ 500 \\ 600 \end{bmatrix}, \qquad \bar{p}_3 = \begin{bmatrix} 400 \\ 500 \\ 600 \end{bmatrix}$$
$$\bar{p}_{4'} = \begin{bmatrix} 400 \\ 459.7 \\ 520 \end{bmatrix}, \qquad \bar{p}_{4''} = \begin{bmatrix} 400 \\ 540.3 \\ 480 \end{bmatrix}, \qquad \bar{p}_5 = \begin{bmatrix} 400 \\ 600 \\ 400 \end{bmatrix}, \qquad \bar{p}_6 = \begin{bmatrix} 500 \\ 500 \\ 500 \end{bmatrix}$$

for the position. Other choices of points $4'$ and $4''$ may be used. For the orientation, the equivalent angle/axis representation is used to generate the trajectory. They are selected to yield the minimum amount of rotational motion.

$$\left(\theta\hat{k}\right)_1 = \pi \begin{bmatrix} 1/\sqrt{2} \\ 0 \\ 1/\sqrt{2} \end{bmatrix}, \quad \left(\theta\hat{k}\right)_2 = \pi \begin{bmatrix} 0.9239 \\ 0 \\ 0.3827 \end{bmatrix}, \quad \left(\theta\hat{k}\right)_{2'} = \pi \begin{bmatrix} 0.9239 \\ 0 \\ 0.3827 \end{bmatrix}$$
$$\left(\theta\hat{k}\right)_3 = \pi \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \left(\theta\hat{k}\right)_{4'} = -\tfrac{5\pi}{4} \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}, \quad \left(\theta\hat{k}\right)_{4''} = -\tfrac{5\pi}{4} \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}$$
$$\left(\theta\hat{k}\right)_5 = -\tfrac{3\pi}{2} \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}, \quad \left(\theta\hat{k}\right)_6 = -\tfrac{4\pi}{3} \begin{bmatrix} -1/\sqrt{3} \\ 1/\sqrt{3} \\ -1/\sqrt{3} \end{bmatrix}.$$

With these specifications, computation of the parameters of the linear segment parabolic blends trajectory may be performed with the same formulas developed in section 6.2.2.3. In particular, for the first segment 1-2, $\ddot{\theta}_1 = \frac{\theta_2 - \theta_1}{t_1\left(t_{d12} - \frac{1}{2}t_1\right)}$. Applying this to the 6-tuple representation, one get the blending acceleration of the first via point

$$\ddot{\bar{p}}_1 = \begin{bmatrix} 0 \\ 0 \\ \frac{600-500}{1(5-0.5\times 1)} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 22.222 \end{bmatrix} \text{mm/s}^2,$$

and

$$\left.\frac{d^2\left(\theta\hat{k}\right)}{dt^2}\right|_1 = \begin{bmatrix} \frac{\left(0.9239-1/\sqrt{2}\right)\pi}{1(5-0.5\times 1)} \\ 0 \\ \frac{\left(0.3827-1/\sqrt{2}\right)\pi}{1(5-0.5\times 1)} \end{bmatrix} = \begin{bmatrix} 0.1513 \\ 0 \\ -0.2265 \end{bmatrix} \text{rad/s}^2.$$

Linear velocity of the segment 1-2 is then calculated from $\dot{\theta}_{12} = \frac{\theta_2-\theta_1}{t_{d12}-\frac{1}{2}t_1}$;

$$\dot{\bar{p}}_{12} = \begin{bmatrix} 0 \\ 0 \\ \frac{600-500}{5-0.5\times 1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 22.222 \end{bmatrix} \text{mm/s},$$

and

$$\left.\frac{d\left(\theta\hat{k}\right)}{dt}\right|_{12} = \begin{bmatrix} \frac{\left(0.9239-1/\sqrt{2}\right)\pi}{5-0.5\times 1} \\ 0 \\ \frac{\left(0.3827-1/\sqrt{2}\right)\pi}{5-0.5\times 1} \end{bmatrix} = \begin{bmatrix} 0.1513 \\ 0 \\ -0.2265 \end{bmatrix} \text{rad/s}.$$

To calculate the acceleration at point 2, the linear velocity of the segment 2-2′, $\dot{\theta}_{22'} = \frac{\theta_{2'}-\theta_2}{t_{d22'}}$, is needed. Hence,

$$\dot{\bar{p}}_{22'} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \text{mm/s}, \quad \left.\frac{d\left(\theta\hat{k}\right)}{dt}\right|_{22'} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \text{rad/s}$$

as would be expected since the original point 2 is repeated and this pauses the robot. Consequently, $\ddot{\theta}_2 = \frac{\dot{\theta}_{22'}-\dot{\theta}_{12}}{t_2}$ may be evaluated;

$$\ddot{\bar{p}}_2 = \begin{bmatrix} 0 \\ 0 \\ \frac{0-22.222}{1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -22.222 \end{bmatrix} \text{mm/s}^2,$$

and

$$\left.\frac{d^2\left(\theta\hat{k}\right)}{dt^2}\right|_2 = \begin{bmatrix} \frac{0-0.1513}{1} \\ 0 \\ \frac{0-(-0.2265)}{1} \end{bmatrix} = \begin{bmatrix} -0.1513 \\ 0 \\ 0.2265 \end{bmatrix} \text{rad/s}^2.$$

Similarly, to calculate the acceleration at point 2′, the linear velocity of the segment 2′-3, $\dot{\theta}_{2'3} = \frac{\theta_3-\theta_{2'}}{t_{d2'3}}$, is needed. Hence,

$$\dot{\bar{p}}_{2'3} = \begin{bmatrix} \frac{400-500}{4} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -25 \\ 0 \\ 0 \end{bmatrix} \text{mm/s},$$

and

$$\left.\frac{d\left(\theta\hat{k}\right)}{dt}\right|_{2'3} = \begin{bmatrix}\frac{(1-0.9239)\pi}{4}\\ 0\\ \frac{-0.3827\pi}{4}\end{bmatrix} = \begin{bmatrix}0.0598\\ 0\\ -0.3006\end{bmatrix} \text{rad/s.}$$

Consequently, $\ddot{\theta}_{2'} = \frac{\dot{\theta}_{2'3}-\dot{\theta}_{22'}}{t_{2'}}$ may be evaluated;

$$\ddot{p}_{2'} = \begin{bmatrix}\frac{-25}{1}\\ 0\\ 0\end{bmatrix} = \begin{bmatrix}-25\\ 0\\ 0\end{bmatrix} \text{mm/s}^2,$$

and

$$\left.\frac{d^2\left(\theta\hat{k}\right)}{dt^2}\right|_{2'} = \begin{bmatrix}\frac{0.0598}{1}\\ 0\\ \frac{-0.3006}{1}\end{bmatrix} = \begin{bmatrix}0.0598\\ 0\\ -0.3006\end{bmatrix} \text{rad/s}^2.$$

Similarly, to calculate the acceleration at point 3, the linear velocity of the segment 3-4′, $\dot{\theta}_{34'} = \frac{\theta_{4'}-\theta_3}{t_{d34'}}$, is needed. Hence,

$$\dot{p}_{34'} = \begin{bmatrix}0\\ \frac{459.7-500}{2}\\ \frac{520-600}{2}\end{bmatrix} = \begin{bmatrix}0\\ -20.15\\ -40\end{bmatrix} \text{mm/s,}$$

and

$$\left.\frac{d\left(\theta\hat{k}\right)}{dt}\right|_{34'} = \begin{bmatrix}\frac{(5/4-1)\pi}{2}\\ 0\\ 0\end{bmatrix} = \begin{bmatrix}0.3927\\ 0\\ 0\end{bmatrix} \text{rad/s.}$$

Consequently, $\ddot{\theta}_{3} = \frac{\dot{\theta}_{34'}-\dot{\theta}_{2'3}}{t_{3}}$ may be evaluated;

$$\ddot{p}_{3} = \begin{bmatrix}\frac{0-(-25)}{1}\\ \frac{-20.15}{1}\\ \frac{-40}{1}\end{bmatrix} = \begin{bmatrix}25\\ -20.15\\ -40\end{bmatrix} \text{mm/s}^2,$$

and

$$\left.\frac{d^2\left(\theta\hat{k}\right)}{dt^2}\right|_{3} = \begin{bmatrix}\frac{0.3927-0.0598}{1}\\ 0\\ \frac{-(-0.3006)}{1}\end{bmatrix} = \begin{bmatrix}0.3329\\ 0\\ 0.3006\end{bmatrix} \text{rad/s}^2.$$

Similarly, to calculate the acceleration at point 4′, the linear velocity of the segment 4′-4″, $\dot{\theta}_{4'4''} = \frac{\theta_{4''}-\theta_{4'}}{t_{d4'4''}}$, is needed. Hence,

$$\dot{p}_{4'4''} = \begin{bmatrix}0\\ \frac{540.3-459.7}{4}\\ \frac{480-520}{4}\end{bmatrix} = \begin{bmatrix}0\\ 20.15\\ -10\end{bmatrix} \text{mm/s,}$$

and

$$\left.\frac{d\left(\theta\hat{k}\right)}{dt}\right|_{4'4''} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \text{ rad/s.}$$

Consequently, $\ddot{\theta}_{4'} = \frac{\dot{\theta}_{4'4''} - \dot{\theta}_{34'}}{t_{4'}}$ may be evaluated;

$$\ddot{p}_{4'} = \begin{bmatrix} 0 \\ \frac{20.15-(-20.15)}{1} \\ \frac{-10-(-40)}{1} \end{bmatrix} = \begin{bmatrix} 0 \\ 40.3 \\ 30 \end{bmatrix} \text{ mm/s}^2,$$

and

$$\left.\frac{d^2\left(\theta\hat{k}\right)}{dt^2}\right|_{4'} = \begin{bmatrix} \frac{0-0.3927}{1} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -0.3927 \\ 0 \\ 0 \end{bmatrix} \text{ rad/s}^2.$$

Similarly, to calculate the acceleration at point $4''$, the linear velocity of the segment $4''$-5, $\dot{\theta}_{4''5} = \frac{\theta_5 - \theta_{4''}}{t_{d4''5}}$, is needed. Hence,

$$\dot{p}_{4''5} = \begin{bmatrix} 0 \\ \frac{600-540.3}{2} \\ \frac{400-480}{2} \end{bmatrix} = \begin{bmatrix} 0 \\ 29.85 \\ -40 \end{bmatrix} \text{ mm/s,}$$

and

$$\left.\frac{d\left(\theta\hat{k}\right)}{dt}\right|_{4''5} = \begin{bmatrix} \frac{(3/2-5/4)\pi}{2} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.3927 \\ 0 \\ 0 \end{bmatrix} \text{ rad/s.}$$

Consequently, $\ddot{\theta}_{4''} = \frac{\dot{\theta}_{4''5} - \dot{\theta}_{4'4''}}{t_{4''}}$ may be evaluated;

$$\ddot{p}_{4''} = \begin{bmatrix} 0 \\ \frac{29.85-20.15}{1} \\ \frac{-40-(-10)}{1} \end{bmatrix} = \begin{bmatrix} 0 \\ 9.7 \\ -30 \end{bmatrix} \text{ mm/s}^2,$$

and

$$\left.\frac{d^2\left(\theta\hat{k}\right)}{dt^2}\right|_{4''} = \begin{bmatrix} \frac{0.3927}{1} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.3927 \\ 0 \\ 0 \end{bmatrix} \text{ rad/s}^2.$$

Similarly, to calculate the acceleration at point 5, the linear velocity of the segment 5-6, $\dot{\theta}_{56} = \frac{\theta_6 - \theta_5}{t_{d56} - \frac{1}{2}t_6}$, is needed. Hence,

$$\dot{p}_{56} = \begin{bmatrix} \frac{500-400}{5-0.5\times1} \\ \frac{500-600}{5-0.5\times1} \\ \frac{500-400}{5-0.5\times1} \end{bmatrix} = \begin{bmatrix} 22.222 \\ -22.222 \\ 22.222 \end{bmatrix} \text{ mm/s,}$$

and

$$\left.\frac{d\left(\theta\hat{k}\right)}{dt}\right|_{56} = \begin{bmatrix} \frac{\left(4/3\sqrt{3}-3/2\right)\pi}{5-0.5\times 1} \\ \frac{-4\pi/3\sqrt{3}}{5-0.5\times 1} \\ \frac{4\pi/3\sqrt{3}}{5-0.5\times 1} \end{bmatrix} = \begin{bmatrix} -0.5098 \\ -0.5374 \\ 0.5374 \end{bmatrix} \text{ rad/s.}$$

Consequently, $\ddot{\theta}_5 = \frac{\dot{\theta}_{56}-\dot{\theta}_{4''5}}{t_5}$ may be evaluated;

$$\ddot{p}_5 = \begin{bmatrix} \frac{22.222}{1} \\ \frac{-22.222-29.85}{1} \\ \frac{22.222-(-40)}{1} \end{bmatrix} = \begin{bmatrix} 22.222 \\ -52.072 \\ 62.222 \end{bmatrix} \text{ mm/s}^2,$$

and

$$\left.\frac{d^2\left(\theta\hat{k}\right)}{dt^2}\right|_{5} = \begin{bmatrix} \frac{-0.5098-0.3927}{1} \\ \frac{-0.5374}{1} \\ \frac{0.5374}{1} \end{bmatrix} = \begin{bmatrix} -0.9025 \\ -0.5374 \\ 0.5374 \end{bmatrix} \text{ rad/s}^2.$$

Finally, the acceleration at point 6 is determined from matching the velocity of the linear segment and the initial velocity entering the last blend. That is, with $\ddot{\theta}_6 = -\frac{\theta_6-\theta_5}{t_6\left(t_{d56}-\frac{1}{2}t_6\right)}$,

$$\ddot{p}_6 = \begin{bmatrix} -\frac{500-400}{1(5-0.5\times 1)} \\ -\frac{500-600}{1(5-0.5\times 1)} \\ -\frac{500-400}{1(5-0.5\times 1)} \end{bmatrix} = \begin{bmatrix} -22.222 \\ 22.222 \\ -22.222 \end{bmatrix} \text{ mm/s}^2,$$

and

$$\left.\frac{d^2\left(\theta\hat{k}\right)}{dt^2}\right|_{6} = \begin{bmatrix} -\frac{\left(4/3\sqrt{3}-3/2\right)\pi}{1(5-0.5\times 1)} \\ -\frac{-4\pi/3\sqrt{3}}{1(5-0.5\times 1)} \\ -\frac{4\pi/3\sqrt{3}}{1(5-0.5\times 1)} \end{bmatrix} = \begin{bmatrix} 0.5098 \\ 0.5374 \\ -0.5374 \end{bmatrix} \text{ rad/s}^2.$$

With these parameters, the trajectory generated by segments of the linear function with parabolic blends for each element of the end effector 6-tuples may be written explicitly as follow;

$$p_x(t) = \begin{cases} 500 & 0 \le t \le 1 \\ 500 & 1 < t \le 4.5 \\ 500 & 4.5 < t \le 5.5 \\ 500 & 5.5 < t \le 7.5 \\ 500 - 12.5\,(t-7.5)^2 & 7.5 < t \le 8.5 \\ 500 - 25\,(t-8) & 8.5 < t \le 11.5 \\ 500 - 25\,(t-8) + 12.5\,(t-11.5)^2 & 11.5 < t \le 12.5 \\ 400 & 12.5 < t \le 13.5 \\ 400 & 13.5 < t \le 14.5 \\ 400 & 14.5 < t \le 17.5 \\ 400 & 17.5 < t \le 18.5 \\ 400 & 18.5 < t \le 19.5 \\ 400 + 11.111\,(t-19.5)^2 & 19.5 < t \le 20.5 \\ 400 + 22.222\,(t-20) & 20.5 < t \le 24 \\ 400 + 22.222\,(t-20) - 11.111\,(t-24)^2 & 24 < t \le 25. \end{cases}$$

$$p_y(t) = \begin{cases} 500 & 0 \le t \le 1 \\ 500 & 1 < t \le 4.5 \\ 500 & 4.5 < t \le 5.5 \\ 500 & 5.5 < t \le 7.5 \\ 500 & 7.5 < t \le 8.5 \\ 500 & 8.5 < t \le 11.5 \\ 500 - 10.075\,(t-11.5)^2 & 11.5 < t \le 12.5 \\ 500 - 20.15\,(t-12) & 12.5 < t \le 13.5 \\ 500 - 20.15\,(t-12) + 20.15\,(t-13.5)^2 & 13.5 < t \le 14.5 \\ 459.7 + 20.15\,(t-14) & 14.5 < t \le 17.5 \\ 459.7 + 20.15\,(t-14) + 4.85\,(t-17.5)^2 & 17.5 < t \le 18.5 \\ 540.3 + 29.85\,(t-18) & 18.5 < t \le 19.5 \\ 540.3 + 29.85\,(t-18) - 26.036\,(t-19.5)^2 & 19.5 < t \le 20.5 \\ 600 - 22.222\,(t-20) & 20.5 < t \le 24 \\ 600 - 22.222\,(t-20) + 11.111\,(t-24)^2 & 24 < t \le 25. \end{cases}$$

$$p_z(t) = \begin{cases} 500 + 11.111t^2 & 0 \le t \le 1 \\ 500 + 22.222\,(t - 0.5) & 1 < t \le 4.5 \\ 500 + 22.222\,(t - 0.5) - 11.111\,(t - 4.5)^2 & 4.5 < t \le 5.5 \\ 600 & 5.5 < t \le 7.5 \\ 600 & 7.5 < t \le 8.5 \\ 600 & 8.5 < t \le 11.5 \\ 600 - 20\,(t - 11.5)^2 & 11.5 < t \le 12.5 \\ 600 - 40\,(t - 12) & 12.5 < t \le 13.5 \\ 600 - 40\,(t - 12) + 15\,(t - 13.5)^2 & 13.5 < t \le 14.5 \\ 520 - 10\,(t - 14) & 14.5 < t \le 17.5 \\ 520 - 10\,(t - 14) - 15\,(t - 17.5)^2 & 17.5 < t \le 18.5 \\ 480 - 40\,(t - 18) & 18.5 < t \le 19.5 \\ 480 - 40\,(t - 18) + 31.111\,(t - 19.5)^2 & 19.5 < t \le 20.5 \\ 400 + 22.222\,(t - 20) & 20.5 < t \le 24 \\ 400 + 22.222\,(t - 20) - 11.111\,(t - 24)^2 & 24 < t \le 25. \end{cases}$$

$$\theta k_x(t) = \begin{cases} \frac{\pi}{\sqrt{2}} + 0.07565t^2 & 0 \le t \le 1 \\ \frac{\pi}{\sqrt{2}} + 0.1513\,(t - 0.5) & 1 < t \le 4.5 \\ \frac{\pi}{\sqrt{2}} + 0.1513\,(t - 0.5) - 0.07565\,(t - 4.5)^2 & 4.5 < t \le 5.5 \\ 0.9239\pi & 5.5 < t \le 7.5 \\ 0.9239\pi + 0.0299\,(t - 7.5)^2 & 7.5 < t \le 8.5 \\ 0.9239\pi + 0.0598\,(t - 8) & 8.5 < t \le 11.5 \\ 0.9239\pi + 0.0598\,(t - 8) + 0.16645\,(t - 11.5)^2 & 11.5 < t \le 12.5 \\ \pi + 0.3927\,(t - 12) & 12.5 < t \le 13.5 \\ \pi + 0.3927\,(t - 12) - 0.19635\,(t - 13.5)^2 & 13.5 < t \le 14.5 \\ \frac{5\pi}{4} & 14.5 < t \le 17.5 \\ \frac{5\pi}{4} + 0.19635\,(t - 17.5)^2 & 17.5 < t \le 18.5 \\ \frac{5\pi}{4} + 0.3927\,(t - 18) & 18.5 < t \le 19.5 \\ \frac{5\pi}{4} + 0.3927\,(t - 18) - 0.45125\,(t - 19.5)^2 & 19.5 < t \le 20.5 \\ \frac{3\pi}{2} - 0.5098\,(t - 20) & 20.5 < t \le 24 \\ \frac{3\pi}{2} - 0.5098\,(t - 20) + 0.2549\,(t - 24)^2 & 24 < t \le 25. \end{cases}$$

$$\theta k_y\left(t\right) = \begin{cases} 0 & 0 \le t \le 1 \\ 0 & 1 < t \le 4.5 \\ 0 & 4.5 < t \le 5.5 \\ 0 & 5.5 < t \le 7.5 \\ 0 & 7.5 < t \le 8.5 \\ 0 & 8.5 < t \le 11.5 \\ 0 & 11.5 < t \le 12.5 \\ 0 & 12.5 < t \le 13.5 \\ 0 & 13.5 < t \le 14.5 \\ 0 & 14.5 < t \le 17.5 \\ 0 & 17.5 < t \le 18.5 \\ 0 & 18.5 < t \le 19.5 \\ -0.2687\left(t-19.5\right)^2 & 19.5 < t \le 20.5 \\ -0.5374\left(t-20\right) & 20.5 < t \le 24 \\ -0.5374\left(t-20\right)+0.2687\left(t-24\right)^2 & 24 < t \le 25. \end{cases}$$

$$\theta k_z\left(t\right) = \begin{cases} \frac{\pi}{\sqrt{2}} - 0.11325t^2 & 0 \le t \le 1 \\ \frac{\pi}{\sqrt{2}} - 0.2265\left(t-0.5\right) & 1 < t \le 4.5 \\ \frac{\pi}{\sqrt{2}} - 0.2265\left(t-0.5\right)+0.11325\left(t-4.5\right)^2 & 4.5 < t \le 5.5 \\ 0.3827\pi & 5.5 < t \le 7.5 \\ 0.3827\pi - 0.1503\left(t-7.5\right)^2 & 7.5 < t \le 8.5 \\ 0.3827\pi - 0.3006\left(t-8\right) & 8.5 < t \le 11.5 \\ 0.3827\pi - 0.3006\left(t-8\right)+0.1503\left(t-11.5\right)^2 & 11.5 < t \le 12.5 \\ 0 & 12.5 < t \le 13.5 \\ 0 & 13.5 < t \le 14.5 \\ 0 & 14.5 < t \le 17.5 \\ 0 & 17.5 < t \le 18.5 \\ 0 & 18.5 < t \le 19.5 \\ 0.2687\left(t-19.5\right)^2 & 19.5 < t \le 20.5 \\ 0.5374\left(t-20\right) & 20.5 < t \le 24 \\ 0.5374\left(t-20\right)-0.2687\left(t-24\right)^2 & 24 < t \le 25. \end{cases}$$

These results of the end effector trajectory functions are evaluated at a rate of 1 kHz. At a specific time t, the equivalent homogeneous transformation matrix representation, ${}^{B}_{E}T$, of ${}^{B}\bar{X}_{E}$ is computed. Then the corresponding joint angles may be determined by substituting the matrix element values into the PUMA 560 inverse kinematics closed form solution of Ex. 4.4. Specifically, the branch of the solution where

$$\begin{aligned}
\theta_1 &= \operatorname{atan2}\left(-\left(p_x - er_{13}\right), p_y - er_{23}\right) \\
&\quad + \operatorname{atan2}\left(\sqrt{p_x^2 + p_y^2 - d_3^2 + e^2\left(r_{13}^2 + r_{23}^2\right) - 2e\left(p_x r_{13} + p_y r_{23}\right)}, d_3\right), \\
K &= \frac{\left(p_x - er_{13}\right)^2 + \left(p_y - er_{23}\right)^2 + \left(p_z - H - er_{33}\right)^2 - a_2^2 - a_3^2 - d_3^2 - d_4^2}{2a_2}, \\
\theta_3 &= \operatorname{atan2}\left(-d_4, d_3\right) - \operatorname{atan2}\left(\sqrt{a_3^2 + d_4^2 - K^2}, K\right), \\
\theta_{23} &= \operatorname{atan2}\left(-\left(a_2 c_3 + a_3\right)\left(p_z - H - er_{33}\right) + \left(a_2 s_3 - d_4\right)\left[\left(p_x - er_{13}\right)c_1 + \left(p_y - er_{23}\right)s_1\right],\right. \\
&\quad \left.\left(a_2 c_3 + a_3\right)\left[\left(p_x - er_{13}\right)c_1 + \left(p_y - er_{23}\right)s_1\right] + \left(a_2 s_3 - d_4\right)\left(p_z - H - er_{33}\right)\right), \\
\theta_2 &= \theta_{23} - \theta_3, \\
\theta_5 &= \operatorname{atan2}\left(\sqrt{1 - \left(r_{13}c_1 s_{23} + r_{23}s_1 s_{23} + r_{33}c_{23}\right)^2}, -\left(r_{13}c_1 s_{23} + r_{23}s_1 s_{23} + r_{33}c_{23}\right)\right), \\
\theta_4 &= \operatorname{atan2}\left(-r_{13}s_1 + r_{23}c_1, -\left(r_{13}c_1 c_{23} + r_{23}s_1 c_{23} - r_{33}s_{23}\right)\right), \\
\theta_6 &= \operatorname{atan2}\left(r_{12}c_1 s_{23} + r_{22}s_1 s_{23} + r_{32}c_{23}, -\left(r_{11}c_1 s_{23} + r_{21}s_1 s_{23} + r_{31}c_{23}\right)\right),
\end{aligned}$$

is selected.

Plots of the trajectory, velocity, and acceleration of the joint angles are shown in Figs. 6.11-6.16. The velocity and acceleration are computed numerically by the backward Euler differentiation formula. It should be mentioned that the computed joint space trajectory does not have the shape of the linear function with parabolic blends as planned in the task space through the elements of ${}^B\bar{X}_E$. This is because the nonlinear inverse kinematic mapping has distorted the path shape. Another point is there are large impulsive accelerations. It is caused by the significant digit rounding off of the adjacent trajectory functions at the task space level. The nonlinear kinematic mapping is a smooth mapping and so should not be blamed for.

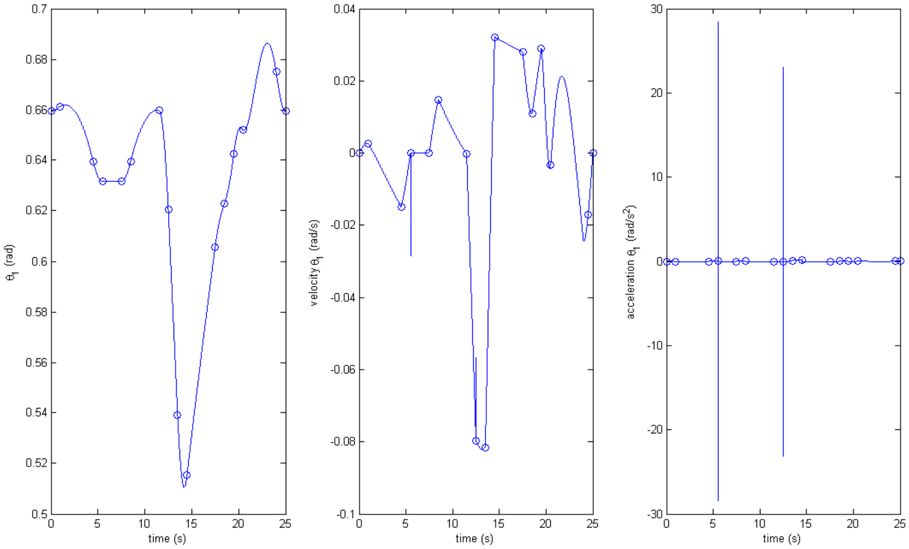


Figure 6.11: Joint θ_1 trajectory and its velocity and acceleration profile generated by segments of the linear function with parabolic blend in the task space for the specified path points, time spent during the segments, and the blending time. The ‘o’ marks indicate the segment transition point.

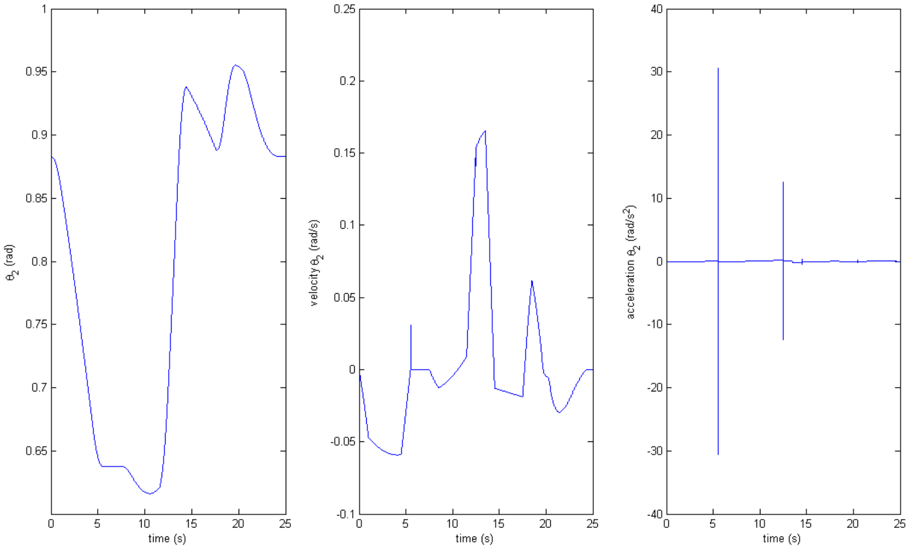


Figure 6.12: Joint θ_2 trajectory and its velocity and acceleration profile generated by segments of the linear function with parabolic blend in the task space for the specified path points, time spent during the segments, and the blending time.

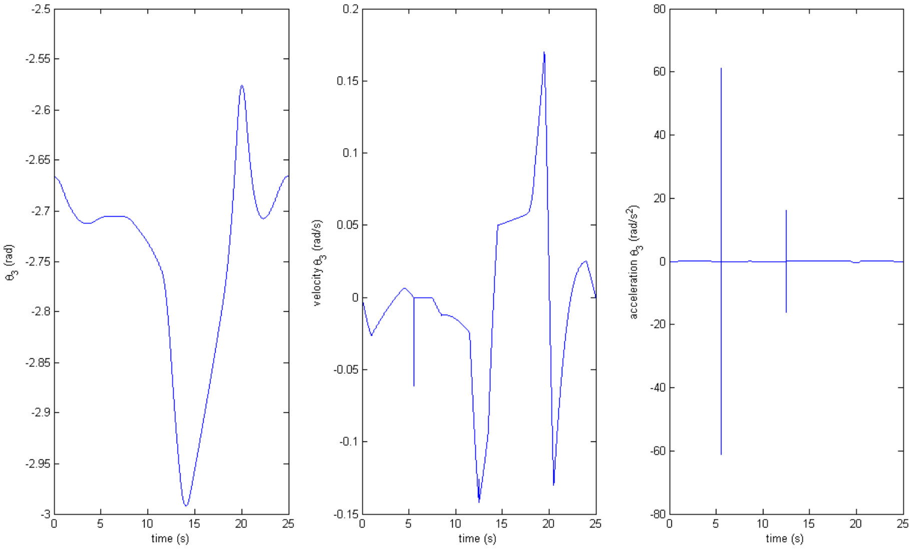


Figure 6.13: Joint θ_3 trajectory and its velocity and acceleration profile generated by segments of the linear function with parabolic blend in the task space for the specified path points, time spent during the segments, and the blending time.

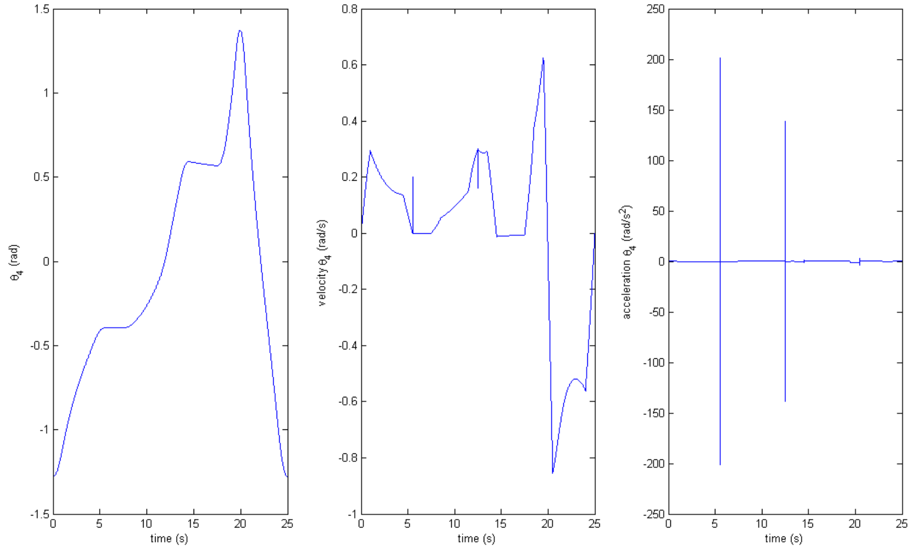


Figure 6.14: Joint θ_4 trajectory and its velocity and acceleration profile generated by segments of the linear function with parabolic blend in the task space for the specified path points, time spent during the segments, and the blending time.

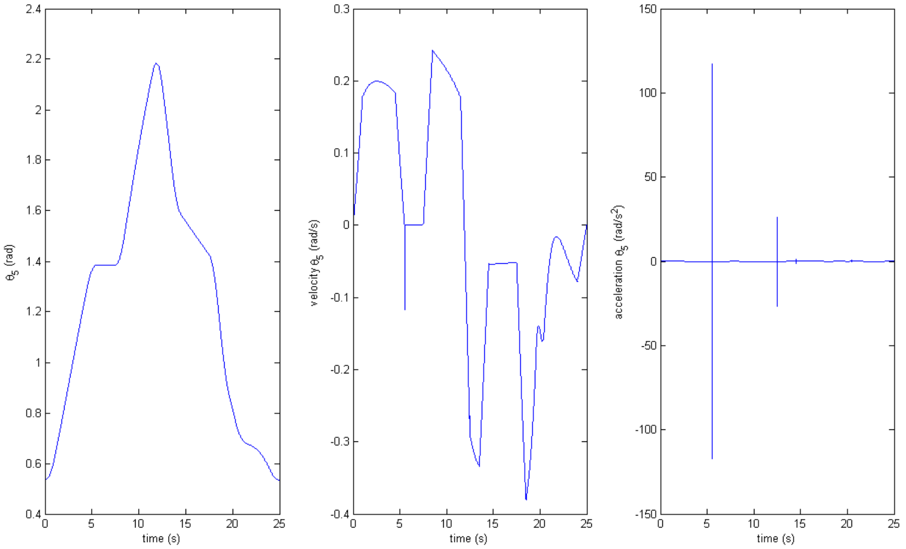


Figure 6.15: Joint θ_5 trajectory and its velocity and acceleration profile generated by segments of the linear function with parabolic blend in the task space for the specified path points, time spent during the segments, and the blending time.

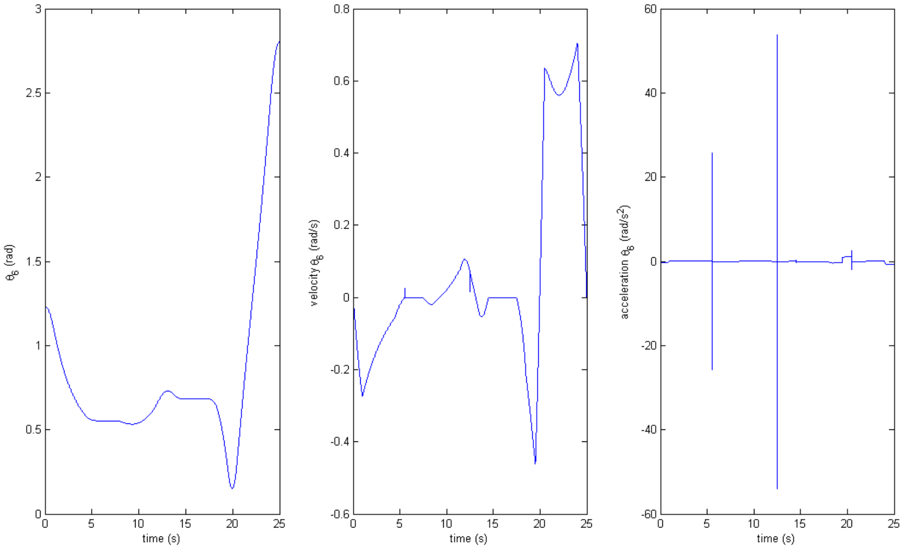


Figure 6.16: Joint θ_6 trajectory and its velocity and acceleration profile generated by segments of the linear function with parabolic blend in the task space for the specified path points, time spent during the segments, and the blending time.

Problems

1. A single revolute joint robot is to be rotated from 40° to 130°, rest to rest, in 6 seconds. Design the trajectory using the LSPB scheme. Time spent for each blend is chosen to be 1 second. Determine the equations of the joint angle for every interval. Sketch the profiles of the joint angle, velocity, and acceleration by stacking them vertically. Indicate significant points in each graph.

2. Calculate $\dot{\theta}_{12}$, $\dot{\theta}_{23}$, t_1, t_2, and t_3 for a two-segment LSPB. For this joint, $\theta_1 = 5°$, $\theta_2 = 15°$, $\theta_3 = 40°$. Assume that $t_{d12} = t_{d23} = 1$ second and that the default acceleration to use during blends is $80°/\text{second}^2$. Sketch plots of position, velocity, and acceleration of θ.

Chapter 7

Manipulator-Mechanism Design

Design of a manipulator is a complex task involving knowledge from many disciplinary areas. It is almost impossible to work on any aspect of the problem in isolation from the others since they all affect one another. For this reason, the design process becomes highly iterative with later decisions requiring revision of earlier ones.

The tasks in which a manipulator can perform depend greatly with the particular design. Measures for the capability of the manipulator are, for example, the load capacity, speed, size and shape of the workspace, resolution, and repeatability. Although the issues in designing the manipulator are comprehension of *arts and sciences*, this chapter attempts to discuss some important aspects of them. The material starts by examining the topics that have the greatest effect on the design. Then more detailed yet less important issues are considered in order.

7.1 Design Based on Task Requirements

Though the concept of the manipulator or the robot is to act as a programmable machine that can perform any general task, in practice each robot is designed specifically to serve a limited set of tasks. It is impractical to have the car assembly robot perform the task of inserting the electronic components on the circuit board. In most cases, task requirements may be specified by the following common attributes.

Number of degrees of freedom Generally the number of DOF for a manipulator should match or exceed the number required by the tasks to be performed. In the latter case, the manipulator will be redundant. The robot will then have extra freedom(s) to position itself to the pose most suitable to perform the task. Not all tasks require a full six DOF, however.

Workspace or sometimes called as the working volume or the working envelope is one of the first requirements in designing the robot since it must be capable of reaching and interacting with the objects or the environment within the confined vicinity to perfom the task. The overall scale of the task and the restricted environment set the required workspace of the manipulator. In most cases, the shape of the workspace and the location of the robot singularity will also need to be considered.

Load capacity The manipulator must be designed to bear the specified load capacity, usually applied at the end effector. This affects the sizing of the robot structure, the materials used, the sizing of the actuator, and the transmission system. Naturally, load capacity is configuration dependent and dynamically dependent on its motion.

Speed Typically the manipulator with the potential of moving with high speed is desirable. However, this implies the need for large actuators and lightweight robot structure among others. In addition, to increase the motion speed to the maximum value in the short amount of time, the actuator is required to possess good acceleration capability as well.

Repeatability and accuracy Accuracy is the ability to be positioned at the desired posture within the error bound. Repeatability is the ability to be re-positioned in the same vicinity, no matter how large the error value will be. These properties depend largely on the manufacturing and assembling quality of the robot parts. It also depends intrinsically on the sizing of the robot, the actuator/sensor resolution, and the control methodology adopted.

7.2 Kinematic Configuration

As the designed requirements have been settled already, the robot design process may begin. The first step should be the design of the kinematic structure of the manipulator. For the serial robot structure, the number of joints equals the number of the robot DOFs.

If no other special reasons, the placement of the robot joints should be so that the last $n-3$ joints orient the end effector and their axes mutually intersect at the *wrist point*. The rest, i.e. the first three joints, are used to position this wrist point. By this design, the robot structure is composed of the positioning structure serially connected to the orienting structure or the wrist. Indeed, it is widely employed by many industrial manipulators. Referring to the Pieper's solution [8], this particular structure will always have the closed form kinematic solutions. Furthermore, the positioning structure should be designed to be kinematically simple, having the link twists equal to 0° or ±90° and having many of the link lengths and link offsets equal to zero. The most commonly used configurations of the positioning structure are described below.

Cartesian is the most straightforward configuration since the actuated motion direction corresponds to the intuitive X, Y, and Z orthogonal Cartesian coordinate direction. Thus the inverse kinematics is trivial. Figure 7.1 shows a Cartesian manipulator and its cubical workspace. It is seen that the first three joints of the robot are prismatic type aligned mutally orthogonal to each other. The construction may be achieved using the parallel structure yielding very stiff robot. Workspace of the Cartesian robot that lies inside itself, while those of the others are outside, is probably the main disadvantange of deploying this robot structure to the existing workcell.

Articulated or anthropomorphic robot typically consists of two joints forming a *shoulder* and one more joint for the *elbow*. Figure 7.2 displays the articulated robot with its workspace being shaded. The first joint axis is vertically

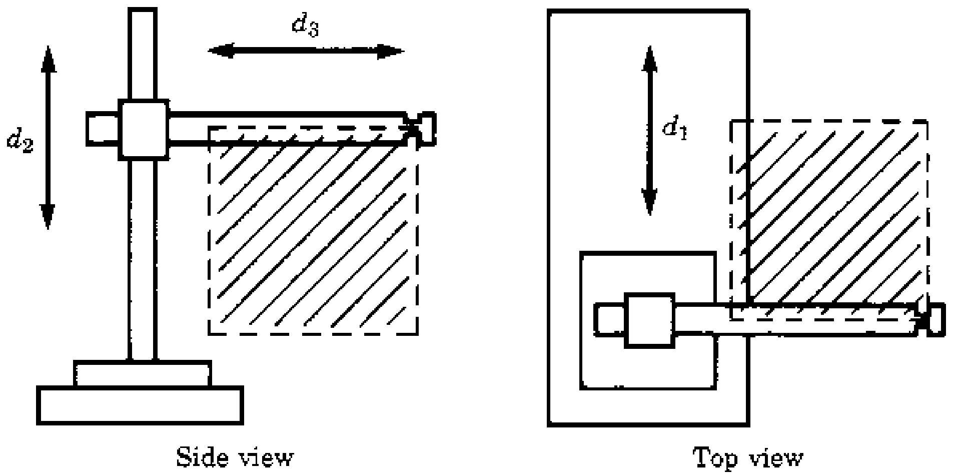


Figure 7.1: Cartesian manipulator and its workspace. ([4], pp. 234)

aligned to rotated the whole robot about its trunk. The second joint axis is horizontally aligned to provide the elevation out of the horizontal plane. The third joint, or the elbow joint, axis is usually parallel to the second joint axis. It provides the freedom for the wrist point to be positioned arbitrarily in the vertical plane. The structure looks like our human arms that allow us to reach the target with minimal invasion, making it be capable of reaching into confined space. Therefore it has been employed by many industrial robots.

SCARA stands for "Selective Compliant Assembly Robot Arm" is another popular structure which employs three parallel revolute joints to move and orient the object in the plane. The fourth prismatic joint is for moving the object normal to the plane. Hence this structure is best suit to the planar tasks such as pick and place. Figure 7.3 depicts the SCARA manipulator and its workspace. By the virtue of this structure, only the carrying load will be seen by the fourth actuator. The first three joint do not have to support the load or the robot weight.

Spherical configuration is made up by replacing the elbow joint of the articulated type with the prismatic joint. This manipulator may be more suitable to the task that requires the telescoping motion normal to the spherical volume. Natural choice of the robot generalized coordinates would be the spherical coordinate system (θ, ϕ, r). Figure 7.4 shows the spherical manipulator with its workspace.

Cylindrical manipulator is inspired by the cylindrical coordinates (h, θ, r) used to describe the position of a point in 3D. Figure 7.5 illustrates the robot

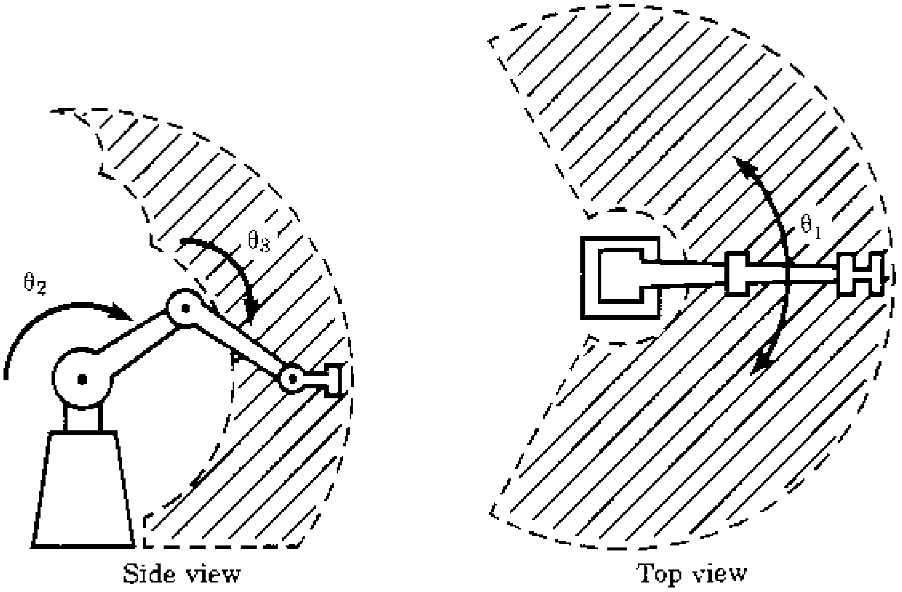


Figure 7.2: Articulated manipulator and its workspace. ([4], pp. 235)

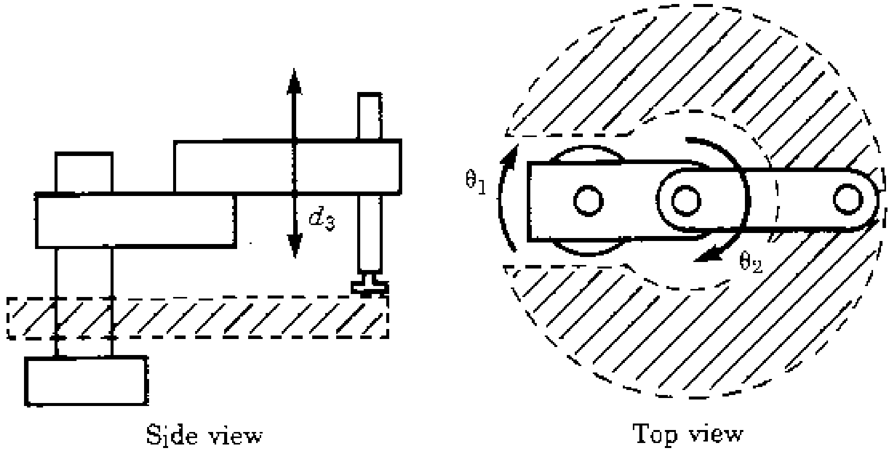


Figure 7.3: SCARA manipulator and its workspace. ([4], pp. 236)

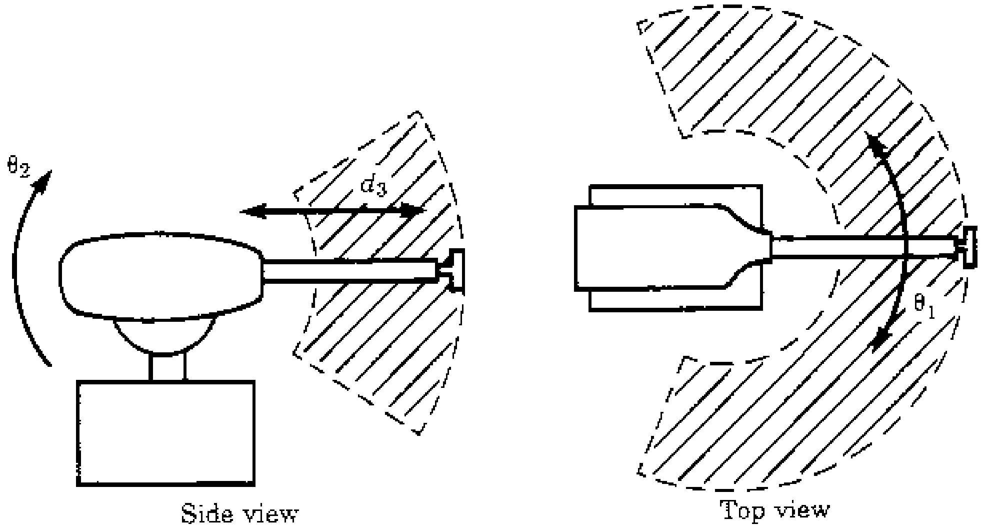


Figure 7.4: Spherical manipulator and its workspace. ([4], pp. 236)

and its workspace. The first prismatic joint translates the arm vertically up by h. The second joint rotates the arm by θ about the trunk. And the third prismatic joint extends/retracts the wrist point horizontally by r. Consequently, the shape of its workspace will be the hollow cylinder.

The design for the orienting structure or the wrist is another subject in its own. The most common configuration employs two or three revolute joints with the mutually orthogonal axes intersecting at the wrist point. Assuming no joint angle limit, the wrist using three consecutive intersecting joints is capable of succeeding arbitrary orientation in 3D. Furthermore, it possesses the closed form inverse kinematic solution. As an example, Fig. 7.6 depict a compact design of such a wrist using the set of bevel gears to drive the wrist remotely from the other end through three concentric shafts. However, this design has a drawback of angle limitations that prevent the continuous rotation of each joint, otherwise the end effector will be jammed into the link structure. The resolution may be to employ an additional joint which will also increase the wrist dexterity. Another way might be to design the wrist using the nonorthogonal joint axes. In this design, all three joints can rotate continuously without limit. However, the workspace will be reduced to the subspace of $SO3$.

7.3 Quantitative Measures of Workspace Attributes

Some interesting quantitative measures of various workspace attributes are listed below. They may be used in designing phase of the manipulator to satisfy certain

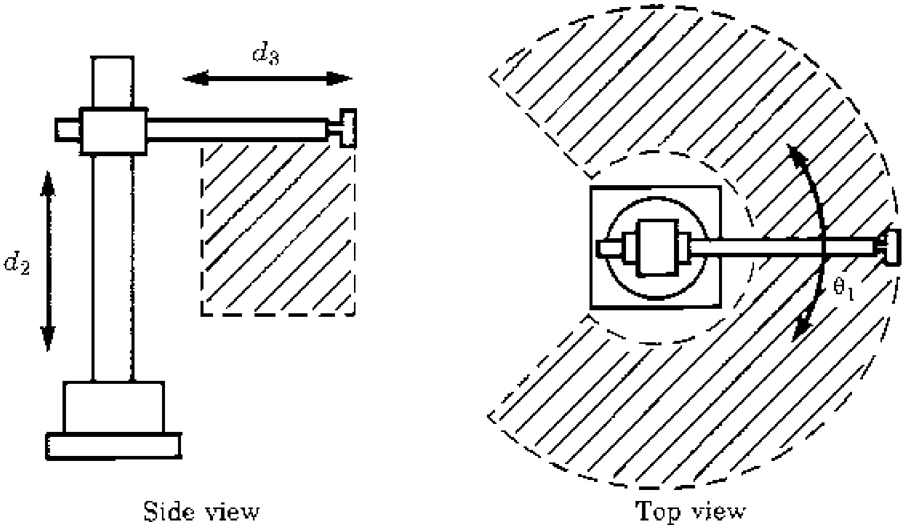


Figure 7.5: Cylindrical manipulator and its workspace. ([4], pp. 237)

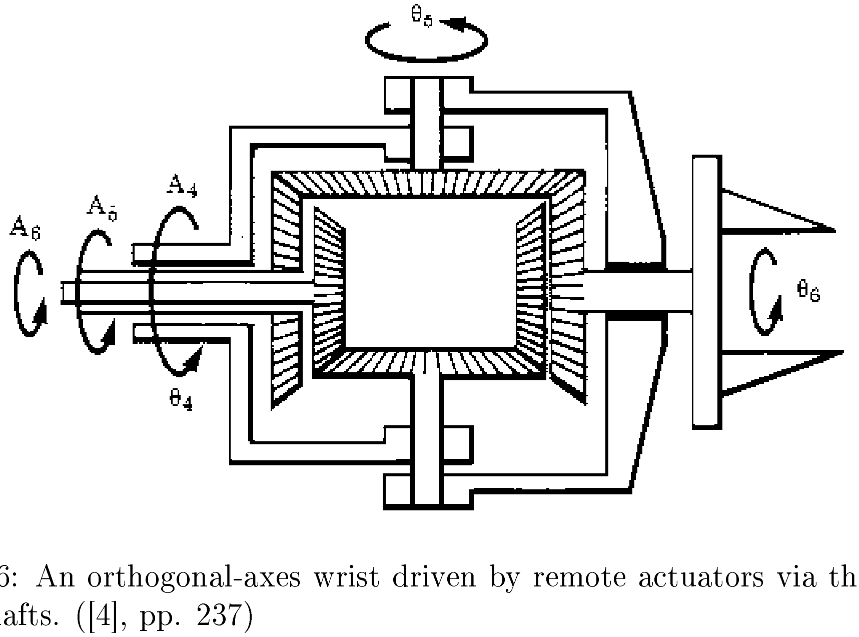


Figure 7.6: An orthogonal-axes wrist driven by remote actuators via three concentric shafts. ([4], pp. 237)

criteria. Generally, one would like the manipulator to possess large workspace. The following indices are related to the ability to generate the workspace.

Length sum gives an approximate measure of the total length of all linkages forming the robot;

$$L = \sum_{i=1}^{N} \left(a_{i-1} + d_i \right), \tag{7.1}$$

where a_{i-1} and d_i are the link length and joint offset of link $\#(i-1)$. Note that d_i must be interpreted as a constant equal to the travel stroke for the prismatic joint. As a matter of fact, the robots made from the prismatic joints, e.g. the Cartesian robot, tend to have a large value of length sum than those of same workspace volume made from the revolute joints, e.g. the articulated robot. The index indicates how much material have been spent to construct the robot.

Structural length index is defined as the ratio of the robot length sum to the cube root of the workspace volume W, i.e.

$$Q_L = L/\sqrt[3]{W}. \tag{7.2}$$

Hence lower Q_L is desirable to achieve a compact manipulator that can reach more spacious workspace. The optimality of Q_L may be used as a guide to the kinematic design of robot.

Not merely the large workspace, but also the ability of the robot to maneuver smoothly is desirable. Recalling chapter 5, the robot will effectively lose one or more DOFs at the singular points. This behavior is not a sudden. In the neighbor of the singular points, the robot could fail to be *well-conditioned*. In other words, the ability to move and apply force uniformly in all direction will be gradually deteriorated. There are several measures which may be used to quantify such effect.

Kinematic Manipulability is the end effector velocity output subject to the constrained unit hyper-sphere joint velocity input. A simpler index called the manipulability measure w, defined [9] as

$$w = \sqrt{\det \left(J \left(\bar{q} \right) J^T \left(\bar{q} \right) \right)}, \tag{7.3}$$

may be used rather. Thus, the robot should be designed to have large area of its workspace characterized by high value of w.

Dynamic Manipulability is the end effector acceleration output subject to the constrained unit applied task space force input [10]. This may be indicated by the eigenvalues and the eigenvectors of the Cartesian mass matrix,

$$M_x \left(\bar{q} \right) = J^{-T} \left(\bar{q} \right) M \left(\bar{q} \right) J^{-1} \left(\bar{q} \right), \tag{7.4}$$

that forms the inertia ellipsoid given by

$$\bar{X}^T M_x \left(\bar{q}\right) X = 1. \tag{7.5}$$

Typically, the well-conditioned robot pose will have large and spherical-shape inertia ellipsoid. Near the singular configuration, the ellipsoid flattens along the directions which are difficult to accelerate or move.

7.4 Actuation

Once the kinematic structure of the manipulator has been decided, the actuation needs to be considered. Generally, the actuator, the reduction, and the transmission are the coupled issues that must be designed altogether.

Actuators There are three common types of the actutators employed in the manipulators. They are electrical, hydraulic, and pneumatic actuators. An important factor is the ratio of the force to the weight of the actuator, for which the higher the value, the more compact the actuator will be.

In general, electrical actuator has quite low force/weight ratio. Nevertheless, it can run at high speed and hence has moderate power/weight ratio value. As a result, usage of the speed reducer to match the required joint velocity is typical. This also improves the actuator system force/weight ratio. Unfortunately, the dynamic response is deteriorated. Anyway, the electric actuator is clean and does not require unwieldy peripheral instruments. The maintainance is not a burden and it is flexible in adapting to various applications. Consequently, electrical actuation holds the biggest share in driving the manipulators.

In the current commercialized technology, hydraulic actuator has the highest force/weight ratio. Therefore, it can be applied to drive the robot joints directly without the speed reducer. Unfortunately, the range of its output force is too high for typical human tasks. Additionally, it requires the augmented circulating and compressing systems. Thus the hydraulic force is mainly employed in the heavy-duty manipulating systems.

Similarly, pneumatic actuator must be supplied with the piping and compressed air, making it more suitable in the automation factory where such things are already available. The hindrance is the compliant behavior of the air which makes it having low force/weight ratio as well. From the control point of view, it is then difficult to make the robot move with high precision using this pnematic force. Therefore, the light-duty on-off tasks may be more appropriate for this kind of actuator.

Reduction and Transmission The best place to install the actuator is at the joint it drives, which is known as the *direct-drive* configuration. It has the advantage of simplicity in the design and control dynamics. The joint motion can be controlled by merely controlling the actuator motion itself. However, in many cases this choice is not possible. Usually the actuator alone cannot produce enough joint torque and hence call for the *speed reduction unit* which will boost up the torque on the price of slowing down motion. Furthermore, the actuator tends to be bulky and heavy. It is thus preferable to place it at or near the base to reduce the effective inertia of the robot. For this purpose, the *transmission unit* is necessary to transfer the motion from the actutor to the remotely located joint.

Aside from added complexity, the major disadvantage of the reduction and transmission is that they introduce additional friction and flexibility into the mechanism. According to their properties and characteristics, a suggested design is to place the light-weight reduction unit at or near the joint after the transmission, in which its flexibility will be less of a problem. The optimal distribution of the reduction stages throughout the transmission depends on the flexibility of the transmission, the weight and friction of the reduction, and the ease of incorporating these components into the overall manipulator design. In the following, some common mechanical elements used in the transmission/reduction unit are described.

Gear is the most popular component used in the reduction unit. It can provide large reduction within compact space. It may also be simultaneously used as the transmission unit to change or transform the motion depending on different gear types. The major drawback of using gear is the backlash and friction.

Flexible Band, Cable, Belt wrapping around the pulleys fulfill both the reduction and transmission functions in transferring the power from the actuator to the robot joint. These elements come with the inherent flexibility. Therefore the tensioning mechanism is required to preload the loop; ensuring that the belt or cable wrapped to the pulley securely. It should be aware that too much tension will induce large friction force and deformation.

Lead Screw, Ball-Bearing Screw yields large reduction and the rotational to translational motion conversion in a compact package. Ball-bearing screw is an improvement of the lead screw in that it has a circulation of ball bearings rolling between the threads of the screw and nut. Hence the friction has been substantially reduced, while for the lead screw the friction is amplified to withstand large load; thus making the mechanism self-locking.

Table 7.1: Stiffness of some commonly used mechanical elements.

Element	Stiffness
uniform rounded shaft	$\frac{G\pi d^4}{32l}$
$1:1$ spur gear	$C_g b r^2$
$\eta:1$ rigid speed reduction	$k_o = \eta^2 k_i$
simple belt drive	$\frac{EA}{l}$
hollow linkage	$\frac{3\pi E\left(d_o^4-d_i^4\right)}{64l^3}$ for round hollow beam $\frac{E\left(w_o^4-w_i^4\right)}{4l^3}$ for square hollow beam

7.5 Stiffness and Deflection

Typically, the manipulator and the drive system should be designed to be stiff as much as possible. This is to achieve the rigid body accuracy in performing the tasks and to avoid the resonance generated in the range of working frequency bandwidth. In the preliminary design, it is often enough to perform rough calculation of the static deflection based on the stiffness of the robot alone. The effective stiffness is influenced by the structural stiffness and the controller stiffness, where only the first one will be considered in this section.

Mechanical parts that make up the structure may be connected in series or in parallel. The net stiffness of the structure forming by the series connection of two flexible parts is

$$k_{\text{series}} = \frac{1}{k_1} + \frac{1}{k_2}, \tag{7.6}$$

while for the parallel connection, the value will be

$$k_{\text{parallel}} = k_1 + k_2. \tag{7.7}$$

The stiffness of some commonly used mechanical elements are listed in Table 7.1. More on this may be studied further in the subject of theory of elasticity and the finite element methods.

7.6 Sensing

Many types of sensors have been employed in the robots to measure various kinds of physical quantities. To mention a few, most sensors are used to measure the primitive quantities such as the encoder to measure the position, the tachometer to measure the velocity, or the gas sensor to measure the amount of CO_2 at the site of the mobile robot. Recently, more sophisticated sensors have also been used to acquire a bulk of data set that will form the meaningful information. Some common ones are, for example, the video camera to obtain a sequence of images for further vision processing, or the laser range finder to measure the distance of the nearby obstacles to the robot which is useful in the navigation.

In this introductory course, only the simple position and force sensors used in almost every robotic system are discussed.

Position Sensors

Position feedback controller is necessary for the robot to be able to perform the desired motion precisely. One may think of sensing the position of the robot end effector directly, using the laser, the magnetic wave, or the vision, for example. However, those kinds of sensor systems are still expensive and yield rather low accuracy (0.01 mm resolution) compared to the position obtained indirectly from the readings of the basic motor encoders, among which the rotary optical encoder is the most popular one.

Basic principle of the sensing is that as the encoder shaft turns, a grating disk having many radial lines embedded interrupts a light beam. The interruption is detected by an optosensor. It will then generate a train of ON/OFF pulses in synchronous with the opening/blocking phases. The shaft angle can then be determined by counting the number of pulses where the direction of the rotation is determined by relative phase of the two square pulses. Most encoders used in robotic system are incremental type because of the lower price and simple installation. However, the robot needs to be calibrated for the home position before using.

Force Sensors

If the robot needs to manipulate the objects or interact with the environment, it should be equipped with force sensors to acquire the actual interaction force. They are invaluable in successfully executing the robot tasks. Most force sensors, or load cells, are to be installed between the robot end effector flange and the tool. Commonly, the sensor has the strain gauges bonded to a specially designed structure. Once the external force/torque is applied to the sensor, the structure is deformed by a small amount which is detected from the voltage unbalance in the bridge circuit. Calibrating the sensor yields the conversion from the voltage readouts to the applied force/torque.

Chapter 8

Introduction to Control of Manipulators

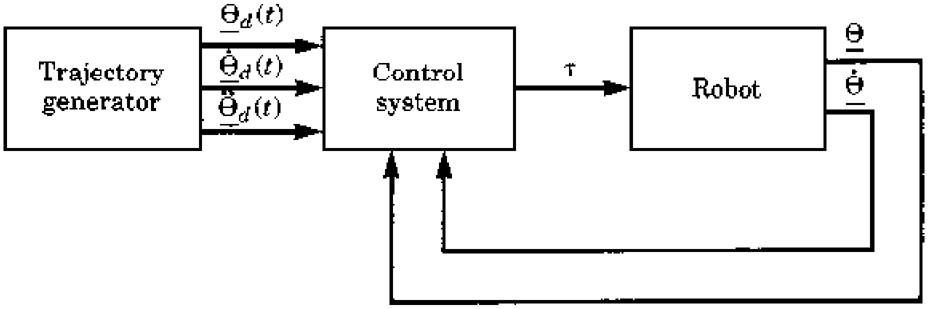

Trajectory generator
$\underline{\Theta}_d(t)$
$\dot{\underline{\Theta}}_d(t)$
$\ddot{\underline{\Theta}}_d(t)$
Control system
τ
Robot
$\underline{\Theta}$
$\dot{\underline{\Theta}}$

Figure 8.1: Overall block diagram of the robot control system. ([4], pp. 263)

Control system is a big subject in itself. Therefore, many control techniques and methods have been applied to the control of manipulators. A selected control method as well as the manner in which it is implemented can have a significant impact on the performance of the robot and on the ranges of its possible applications. For example, the trajectory tracking task requires a different control architecture than does the regulation task.

In addition, mechanical design of the robot pays a great influence on the types of control scheme needed. The control problem of a Cartesian manipulator are fundamentally different from the ones of articulated structure mainly because the system dynamics are different. This is the hardware/software trade-off encountered in controlling the physical plant.

In this chapter, a general control block diagram of the robot is considered. Then the simplest robot control scheme namely the independent joint control will be discussed. Finally, the hybrid position/force control scheme, widely adopted in the manipulation tasks, is introduced.

8.1 Feedback and Closed Loop Control

Most of the robots have been equipped with the joint encoders which allow one to determine the robot posture and use it as the feedback signal to generate appropriate actuator torque commands that control the robot to follow the desired motion. Figure 8.1 captures the basic manipulator control system displaying the interconnection between the trajectory generator, the controller, and the robot. It is quite simple to see that if the controller computed torque according to the commanded motion matches the driving torque of the exact robot dynamics, the robot would be driven along the commanded trajectory. Essentially, the controller relies on the model solely. This is called the open-loop controller.

Certainly, in the actual system the scheme will not be successful due to imperfect knowledge of the robot model and unavoidable presence of disturbances

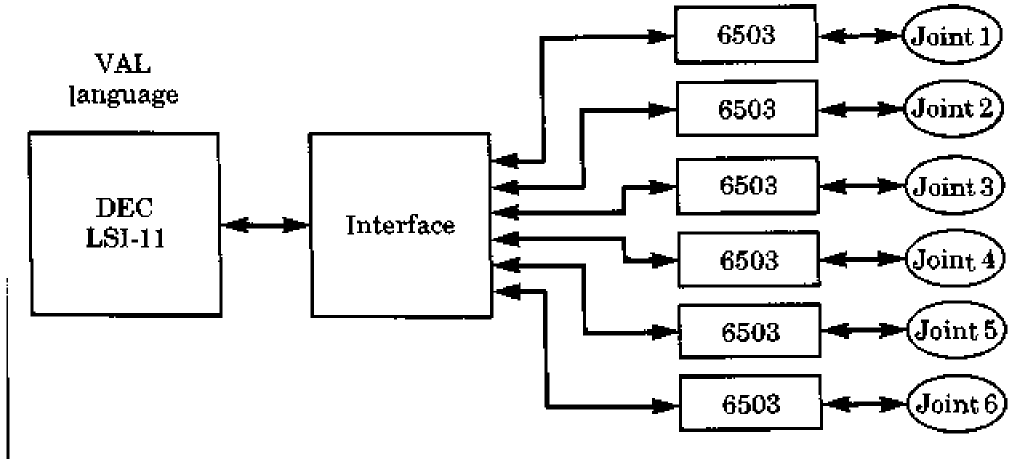


Figure 8.2: PUMA 560 control system architecture. ([4], pp. 284)

and noises to the system. To cope with these unmodeled dynamics, a feedback control scheme must be used. This is indicated by the feedback signal looping back to the controller block as shown in Fig. 8.1. Specifically, the motion feedback is used to compute the servo error as the difference between the desired and the actual motion, i.e.

$$\begin{aligned} e &= \theta_d - \theta \\ \dot{e} &= \dot{\theta}_d - \dot{\theta}. \end{aligned}$$

The control law will then determine the actuator torque required to reduce such error while maintaining the stability of the control system.

From Fig. 8.1, it is seen that the robot control is inherently a multi-input multi-output (MIMO) control problem due to its joint coupling dynamics. Nevertheless with some reasonable justifications, each joint of the robot may be controlled separately. This simplifies the problem to N independent single-input single-output (SISO) control system. It is the design approach adopted by most industrial robots.

8.2 Independent Joint Control Scheme

Figure 8.2 depicts the architecture of the PUMA 560 robot control system. It consists of the high level controller hosted by a DEC LSI-11 computer generating and passing motion commands down to the low level six Rockwell 6503 microprocessors. Each of them controls an individual joint angle through a PID torque control law as depicted in the functional block diagram of Fig. 8.3.

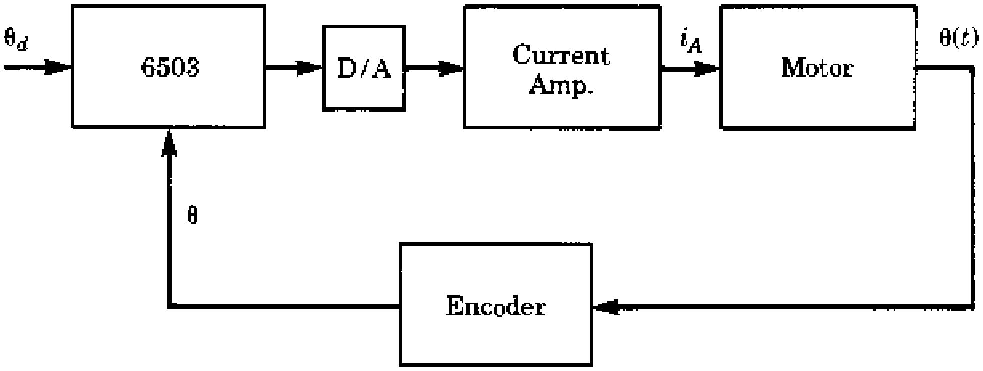


Figure 8.3: Functional block diagram of the low level PUMA 560 joint controller. ([4], pp. 285)

8.2.1 Simplified Robot Model

A simplified model of a single rotary joint of the robot will now be developed. The actuator used in the analysis is the DC torque controlled motor. It has the motor torque constant k_m that relates the armature current i_a to the output torque τ_m:

$$\tau_m = k_m i_a. \tag{8.1}$$

Assume that

- the inductance of the armature circuit is small enough; making the time constant of the armature current very short compared to that of the robot motion.
- the back emf voltage is small compared to the armature voltage.
- the motor torque ripple due to the winding commutation can be neglected.

Hence, the electrical dynamics of the motor may be omitted and therefore the motor torque may be commanded directly via the armature current through Eq. 8.1.

Now consider the mechanical model of a DC motor and load as shown in Fig. 8.4. The motor is connected to the load through the gear unit having the ratio $\eta > 1$. With the supplied motor torque, τ_m, and the associated motor speed, $\dot{\theta}_m$, it causes an increase in the output torque to the load, τ, and a reduction in the speed of the load, $\dot{\theta}$, governed by

$$\tau = \eta \tau_m, \tag{8.2}$$

$$\dot{\theta} = \frac{1}{\eta}\dot{\theta}_m. \tag{8.3}$$

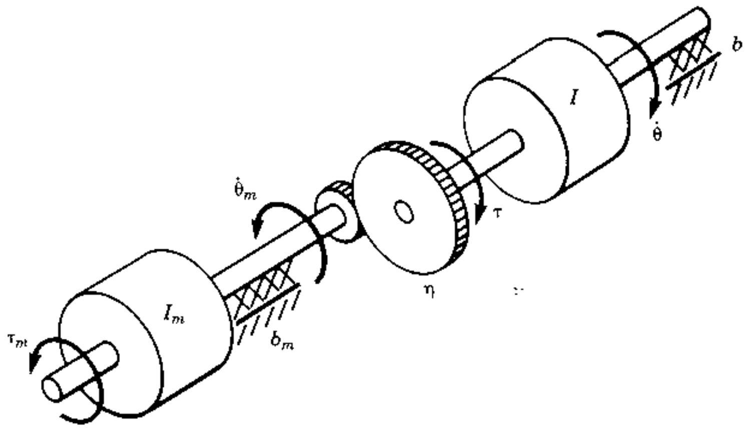


Figure 8.4: Diagram of a motor connected to the load through the gear unit. ([4], pp. 280)

Formulating a torque balance of the system at the motor side and applying Eq. 8.2 yield

$$\tau_m = I_m\ddot{\theta}_m + b_m\dot{\theta}_m + \frac{1}{\eta}\left(I\ddot{\theta} + b\dot{\theta}\right), \tag{8.4}$$

where I_m, I, b_m, and b are the inertia and the support damping coefficient of the motor and the load respectively. Then, recalling Eq. 8.3, one may write Eq. 8.4 in terms of the motor variables solely as

$$\tau_m = \left(I_m + \frac{I}{\eta^2}\right)\ddot{\theta}_m + \left(b_m + \frac{b}{\eta^2}\right)\dot{\theta}_m. \tag{8.5}$$

Moreover, the equation might also be expressed in terms of the load variables as

$$\tau = \left(I + \eta^2 I_m\right)\ddot{\theta} + \left(b + \eta^2 b_m\right)\dot{\theta}. \tag{8.6}$$

The terms $I + \eta^2 I_m$ and $b + \eta^2 b_m$ are the effective inertia/damping seen by the load, while $I_m + \frac{I}{\eta^2}$ and $b_m + \frac{b}{\eta^2}$ are the effective inertia/damping seen by the motor. Hence if the robot is highly geared, i.e. $\eta \gg 1$, the contribution of the inertia from the motor, I_m, will far dominate that from the load, I, which is usually configuration dependent. Therefore, one may estimate the total inertia to be constant of combining the maximum load I_{max} and the motor inertia together. Consequently, the design of the robot controller will become greatly simplified because the robot dynamics may be approximated by the n-decoupled second order ODEs. A variety of well established tools in the linear system can now be used to design the linear controller readily.

Practically in designing the controller, one cannot choose the closed loop poles to be arbitrarily fast because this would inevitably excite the unmodeled dynamics of the system. During deriving the equations of motion of Eqs. 8.5

or 8.6, the flexibility of highly stiff parts such as the gear unit, the shaft, and the robot link have not been considered. They mainly create unmodeled dynamics of the system and it will be observed if the system is operated at high frequency range, i.e. when the robot moves quickly. Fast robot motion might be from the fast controller responding to the error promptly. Thus as a rule of thumb, if the lowest structural resonance of the robot is ω_r, then the bandwidth of the closed loop system, ω_n, should be limited such that

$$\omega_n \leq \frac{1}{2}\omega_r \tag{8.7}$$

to avoid exciting the unmodeled dynamics caused by the structural compliance. Typical industrial robots have the structural resonance in the range of 5 to 25 Hz. On the contrary, a high performance direct-drive robotics research platform may possess the good bandwidth up to 70 Hz.

8.2.2 A Simple Robot Controller

According to the developed robot model of Eq. 8.6, a simple control law can then be designed as follow. For each robot joint, the effective load inertia reflected to the motor axis must be determined. In order to design a simple linear controller, the maximum load inertia will be employed. Therefore the equation of motion may be written as

$$\tau = \left(I_{max} + \eta^2 I_m\right)\ddot{\theta} + \left(b + \eta^2 b_m\right)\dot{\theta}. \tag{8.8}$$

The following control law

$$\tau = \left(I_{max} + \eta^2 I_m\right)\tau' + \left(b + \eta^2 b_m\right)\dot{\theta} \tag{8.9}$$

is proposed where τ' is the new control input to be designed. This technique is called the partitioned control method. Combining the system and the controller, the closed loop system becomes

$$\tau' = \ddot{\theta} \tag{8.10}$$

as if it appears to be a unit mass.

If $\theta_d\left(t\right)$ is the desired joint angle determined from the trajectory generator along with its derivatives $\dot{\theta}_d$ and $\ddot{\theta}_d$, the new control input may be proposed as

$$\tau' = \ddot{\theta}_d + k_v\left(\dot{\theta}_d - \dot{\theta}\right) + k_p\left(\theta_d - \theta\right). \tag{8.11}$$

Substituting this control law into Eq. 8.10, the closed loop control system becomes

$$\ddot{e} + k_v\dot{e} + k_p e = 0 \tag{8.12}$$

where $e = \theta_d - \theta$ is the tracking error between the desired and actual angles. Hence, the error dynamics is governed by a second order ODE. Usually the velocity and the position gains, k_v and k_p, will be selected such that the error response is critically damped. Particularly,

$$k_p = \omega_n^2 = \tfrac{1}{4}\omega_r^2 \tag{8.13}$$

$$k_v = 2\sqrt{k_p} = \omega_r. \tag{8.14}$$

However, in the real system the unmodeled system dynamics and the disturbance distort the response from this simple behavior.

8.3 Hybrid Position/Force Control Scheme

Position control is appropriate for the task that the robot is moving in free space such as spot welding or spray painting. In many tasks, the robot needs to make a contact with the environment; for example, in the deburring task in manufacturing. Inevitably, there will be some uncertainties in the surface geometry of the workpiece or a small error in the tracking of the robot. They will cause a large interaction force which is likely to damage the object or destroy the precision of the robot.

Such tasks may be achieved naturally by controlling the normal force to the surface in contact rather than the position in that direction. In general, the control framework in this section is applicable to the task in which motion of the robot is partially constrained by contact with the surfaces. The idea is that every manipulation task may be decomposed into subtasks which are defined by a particular contact scenario between the robot end effector, or the tool, and the environment. For each subtask, there is a set of *natural constraints* that govern the interaction according to the geometry of the matings. A robot in contact with the rigid surface is not free to move in the normal direction and hence the natural velocity constraint occurs. On the other hand, a robot free to slide along the frictionless surface implies null force. Thus a natural force constraint exists so the robot is unable to apply the force in that direction.

For the task at hand, a *generalized surface* may be defined so that there exist natural velocity constraints along the normal direction and natural force constraints along the tangent. Therefore, a *constraint frame* $\{C\}$ should be set up at the port of interaction to clarify the directions. Figure 8.5 depicts a robot turning a crank with $\{C\}$ attached. Natural constraints can then be analyzed. Since the robot is gripping the handle, the translation and rotation of the end effector along the x and z-direction is not possible; $v_x = 0$, $v_z = 0$, $\omega_x = 0$, $\omega_y = 0$. However, it is allowed to freely rotate about the handle. Also, the pulling along the y-axix does not required any effort for the ideal frictionless crank. Therefore, the following natural force constraints, $n_z = 0$ and $f_y = 0$, are applied to the

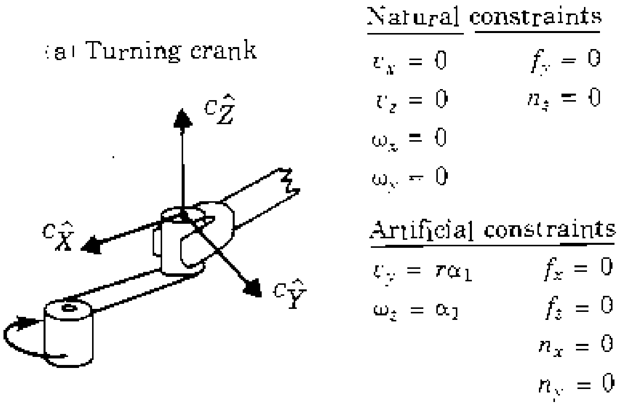


Figure 8.5: Natural and artificial constraints of the turning crank task. ([4], pp. 320)

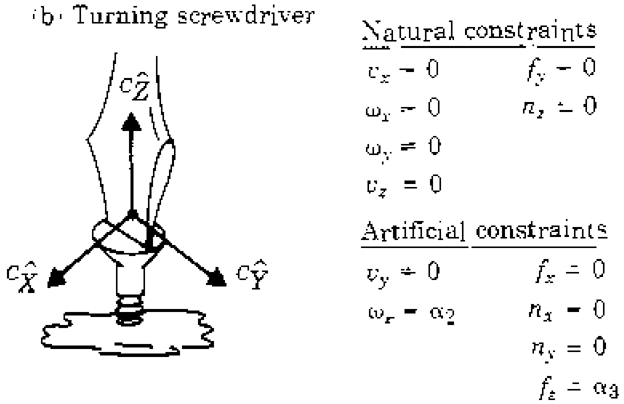


Figure 8.6: Natural and artificial constraints of the turning screwdriver task. ([4], pp. 320)

system. As another example, Fig. 8.6 displays a screwdriver with $\{C\}$ attached. If the friction torque can be neglected, the screwdriver can turn along its axis freely; $n_z = 0$. Additionally, the screwdriver can slip out of the slot of the screw head; $f_y = 0$. However, it is not allowed in the normal function to move along the x- and z-axes as well as to rotate about the x- and y-axes. In other words, $v_x = 0$, $v_z = 0$, $\omega_x = 0$, and $\omega_y = 0$.

Accordingly, these two types of constraints partition the problem into position and force control. In the hybrid position/force control scheme, *artificial constraints* will be imposed according with their associated natural constraints in order to specify the desired motion or force for accomplishing the task. To be consistent with the natural constraints, artificial force constraints will be determined along the surface normal while artificial velocity constraints along the tangent. Consequently, for the turning crank task in Fig. 8.5, one may specify

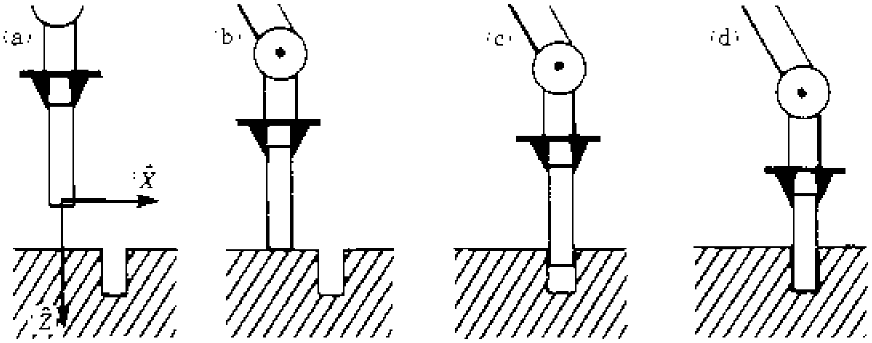


Figure 8.7: Subtasks of the peg-in-hole problem. ([4], pp. 321)

the velocity in rotating the crank to be $v_z = r\alpha_1$ corresponding to $\omega_z = \alpha_1$. The applied force and moment along the immovable direction may be all set to zero to avoid any possible damage; $f_x = 0$, $f_z = 0$, $n_x = 0$, and $n_y = 0$. For the other problem in Fig. 8.6, one may specify the turning motion of the screwdriver by $\omega_z = \alpha_2$ while maintaining the contact of the screwdriver with the head by some contact force $f_z = \alpha_3$. In addition, $v_y = 0$ is set to prevent the slipping of the screwdriver out of the slot. As usual, the applied force along the constrained motion direction will be set to zero, i.e. $f_x = 0$, $n_x = 0$, and $n_y = 0$.

Some tasks are far complicated to be described by merely a single natural constraints. Consider a typical assembly task known as 'peg-in-hole' problem. This task may be divided into four different subtasks as shown in Fig. 8.7. In order to transit from a subtask to the next one, the controller must detect a threshold change in the force or velocity signal of the natural constraint. Thus, the artificial constraint must be changed and enforced by the control system.

Frame $\{C\}$ is attached to the peg for the description of the constraints. For the first subtask (a), the peg is free to move which implies the zero natural force constraints. Therefore, the artificial velocity constraints may be enforced which moves the peg toward the ground surface, i.e. $v_z = v_{\text{approach}}$, while the other velocity components are all zero.

At the moment the peg touch the ground, $f_z \geq f_{\text{threshold}}$ will be detected. This signifies the change to a new subtask (b) with the new natural constraints; $\omega_x = 0$, $\omega_y = 0$, $v_z = 0$, $f_x = 0$, $f_y = 0$, $n_z = 0$. Therefore, the following artificial constraints of $v_x = v_{\text{slide}}$, $v_y = 0$, $f_z = f_{\text{contact}}$, $n_x = 0$, $n_y = 0$, $\omega_z = 0$, may be applied so the peg will be slided along the ground firmly.

As the peg starts falling into the hole, the velocity $v_z \geq v_{\text{threshold}}$ will be detected. Thus the assembly enters the next subtask (c) with the new constraints; $v_x = 0$, $v_y = 0$, $\omega_x = 0$, $\omega_y = 0$, $f_z = 0$, $n_z = 0$. To insert the peg into the perfectly straight hole, the robot merely imparts the velocity in the downward direction and regulating the rotation about the z-axis, i.e. $v_z = v_{\text{insert}}$ and $\omega_z = 0$. Along the immovable directions, zero force/torque reaction are monitored to prevent

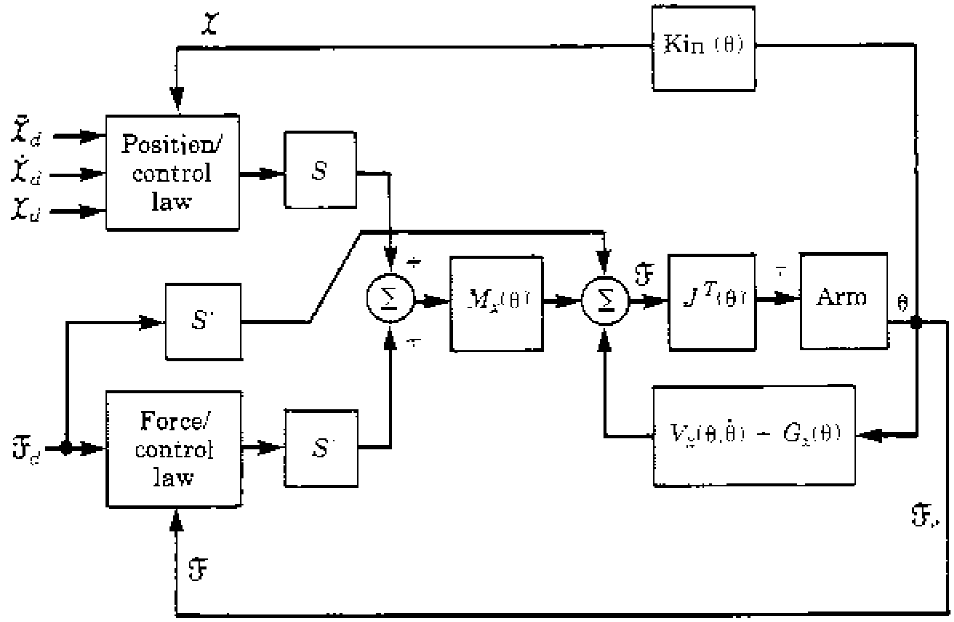


Figure 8.8: Hybrid position/force controller block diagram. ([4], pp. 331)

jamming into the hole; $f_x = 0$, $f_y = 0$, $n_x = 0$, $n_y = 0$. Finally, when there is a change in force along the z-axis, or $f_z \geq f_{\text{threshold}}$, the peg has just reached the bottom of the hole and the task is finished.

Block diagram of the hybrid force/position control scheme is illustrated in Fig. 8.8. The inner loop controller attempts to linearize and decouple the robot so the end effector closed loop dynamics appears as a set of independent uncoupled unit masses. Note that the robot dynamics is expressed in the task natural constraint frame $\{C\}$. The Jacobian transpose is required to transform the end effector control force to the equivalent joint torque.

The outer loop controller actually implements the hybrid force/position control. Details of the position and force controllers are abstracted into the black blocks, which imply possible variations in their implementations. Desired position and force trajectory of the robot end effector may be specified and expressed in the base reference frame $\{B\}$ for convenience. Thus, the transformation to $\{C\}$ must be performed in the respective controllers. Errors are determined and utilized in the controller to generate appropriate control efforts. They are then fed through the switching matrices S and S' to select whether the position or force control will actually be used in each degree of freedom of $\{C\}$. These matrices may be changed according to the task the robot is currently performed.

Appendix: Robotics Toolbox

This appendix provides an introduction to the usage of the Robotics Toolbox [6] in MatLAB® program. The toolbox is developed by Prof. Peter Corke, currently at the school of Electrical Engineering, University of Melbourne. It is assumed that the readers are familiar with MatLAB® to a certain extent. The contents will therefore be directed to the commonly useful commands in the toolbox which complements the study in this course.

Setting Up the System

Generally, the reader needs to install the MatLAB® program first, which should be of the release R2011a or higher for compatibility reason. Then one can obtain a copy of the Robotics Toolbox with no charge at `http://petercorke.com/Robotics_Toolbox.html` by filling some visitor information. The current toolbox version is 9.8 (as of February 2013). After downloading the zipped file of the toolbox, unpack the archive which will create the directory `rvctools` and its subdirectories.

The next step is to modify the `MATLABPATH` environment in MatLAB® so that `rvctools` directory is included in the list. Then, every time you would like to use the Robotics Toolbox, the script in `rvctools/startup_rvc.m` file needs to be executed first. This may be done by issuing `startup_rvc` at the MatLAB® command prompt >>. The script will setup appropriate shell environments required to execute the codes in the toolbox. To verify that the setup is successful, issue `rtbdemo` to run the demo of this toolbox.

Useful Commands

Following are some useful commands in Robotics Toolbox which complements the study in this course. Commands will be grouped according to the relevant topics. Only basic usage of these commands are described here. Interested readers might need to consult the toolbox manual `robot.pdf` which is available at `rvctools/robot` directory for complete list of all commands and their usages.

Generating the Rotation and Transformation Matrices

rotx

generate the rotation matrix corresponding to the rotation about the x-axis.

Syntax#1: `R = rotx(theta)` yields the rotation matrix `R` that represents the rotation of `theta` radian about the x-axis.

SYNTAX#2: `R = rotx(theta, ‘deg’)` with the option `‘deg’`, one can specify the angle of rotation `theta` in degree.

roty

generate the rotation matrix corresponding to the rotation about the y-axis.

rotz

generate the rotation matrix corresponding to the rotation about the z-axis.

trotx

generate the homogeneous transformation matrix corresponding to the rotation about the x-axis. Note that the translational components are all zero.

SYNTAX#1: `T = trotx(theta)` yields the homogeneous transformation matrix `T` that represents the rotation of `theta` radian about the x-axis.

SYNTAX#2: `T = trotx(theta, ‘deg’)` with the option `‘deg’`, one can specify the angle of rotation `theta` in degree.

troty

generate the homogeneous transformation matrix corresponding to the rotation about the y-axis.

trotz

generate the homogeneous transformation matrix corresponding to the rotation about the z-axis.

transl

generate the homogeneous transformation matrix corresponding to the pure translation.

SYNTAX#1: `T = transl(x, y, z)` yields the homogeneous transformation matrix representing the pure translation along the vector, of which its components are `x`, `y`, and `z`.

SYNTAX#2: `T = transl(p)` specifies the pure translation by the 1 × 3 vector `p`.

Changing the Orientation Representation

tr2eul

calculate the ZYZ Euler angles, in radians or degrees, corresponding to the specified orientation by either the homogeneous transformation or the rotation matrix. Singularity is handled by setting the first Euler angle to zero.

Syntax#1: `eul = tr2eul(T)` yields the row vector `eul` which describes the rotational part of the specified homogeneous transformation matrix `T` in the ZYZ Euler angles representation. Option `'deg'` may be used to retrieve the angles in degree instead of the default values in radian.

Syntax#2: `eul = tr2eul(R)` uses the rotation matrix `R` to specify the orientation.

eul2tr

compute the homogeneous transformation matrix equivalent to the specified ZYZ Euler angles.

Syntax#1: `T = eul2tr(eul)` yields the homogeneous transformation matrix `T` where its rotational part corresponds to the specified row vector `eul` containing the rotated angles about the moving axes Z, Y, and Z respectively. Option `'deg'` may be used if the angles are measured in degree instead of the default values in radian.

Syntax#2: `T = eul2tr(phi, theta, psi)` may be called if the three Euler angles `phi`, `theta`, and `psi` about the moving axes Z, Y, and Z are given explicitly.

tr2rpy

calculate the XYZ fixed angles representation of the specified orientation by either the homogeneous transformation or the rotation matrix. Singularity is handled by setting the first fixed angle to zero.

Syntax#1: `rpy = tr2rpy(T)` yields the row vector `rpy` which describes the rotational part of the specified homogeneous transformation matrix `T` in the XYZ fixed angles representation. Option `'deg'` may be used to retrieve the angles in degree instead of the default values in radian.

Syntax#2: `rpy = tr2rpy(R)` uses the rotation matrix `R` to specify the orientation.

rpy2tr

compute the homogeneous transformation matrix equivalent to the specified roll, pitch, yaw angles (or the XYZ fixed angles).

SYNTAX#1: `T = rpy2tr(rpy)` yields the homogeneous transformation matrix `T` where its rotational part corresponds to the specified row vector `rpy` containing the rotated angles about the fixed axes X, Y, and Z (or the roll, pitch, and yaw motion) respectively. Option `'deg'` may be used if the angles are measured in degree instead of the default values in radian.

SYNTAX#2: `T = rpy2tr(roll, pitch, yaw)` may be called if the three angles are given explicitly.

tr2angvec

calculate the angle and axis of which the rotation by such angle about this axis is described by the specified rotation matrix.

SYNTAX#1: `[theta, v] = tr2angvec(R)` yields the rotated angle `theta` in radian and the axis `v` which represent the rotation in angle-axis format of the specified rotation matrix `R`.

SYNTAX#2: `[theta, v] = tr2angvec(T)` is the same as the above call except the rotation matrix is specified through the homogeneous transformation matrix.

angvec2tr

compute the homogeneous transformation matrix which describes the rotation about the given axis by the specified angle.

SYNTAX: `T = angvec2tr(theta, v)` yields the homogeneous transformation matrix `T` where its rotational part corresponds to the rotation about the axis `v` by the angle `theta`. The translational part of `T` is zero.

Trajectory Generation

tpoly

generate a set of points along the quintic polynomial curve joining the designated initial and final angles of a single DOF joint. The starting and ending velocity and acceleration are set to zero.

SYNTAX#1: `[s, sd, sdd] = tpoly(s0, sf, m)` returns a sequence of equally time-spaced `m` discrete points along the quintic polynomial curve joining the specified initial and final angles `s0` and `sf`. Corresponding velocity and acceleration values may be obtained through the optional vector output arguments `sd` and `sdd`.

SYNTAX#2: `[s, sd, sdd] = tpoly(s0, sf, T)` specifies the sequence of time through the vector `T` at which the discrete points will be computed accordingly. The starting (0) and ending time are associated with `s0` and `sf`.

lspb

generate a set of points along the linear segment with parabolic blends (LSPB) curve joining the designated initial and final angles of a single DOF joint.

SYNTAX#1: `[s, sd, sdd] = lspb(s0, sf, m, v)` returns a sequence of equally time-spaced `m` discrete points along the LSPB curve joining the specified initial and final angles `s0` and `sf`. Corresponding velocity and acceleration values may be obtained through the optional vector output arguments `sd` and `sdd`. The constant velocity `v` of the linear segment may be optionally specified. Otherwise the command automatically computes the suitable velocity.

SYNTAX#2: `[s, sd, sdd] = lspb(s0, sf, T, v)` specifies the sequence of time through the vector `T` at which the discrete points will be computed accordingly. The starting (0) and ending time are associated with `s0` and `sf`.

mtraj

generate a sequence of samples along the customized trajectory joining the designated initial and final angles of multi DOFs joints.

SYNTAX#1: `[q, qd, qdd] = mtraj(tfunc, q0, qf, m)` returns a sequence of equally time-spaced `m` discrete samples of the joint vector `q` along the scalar trajectory generating function `tfunc` joining the specified initial and final angles. These values of possibly multi DOFs joints are assigned by the row vectors `q0` and `qf`. The heading of `tfunc` function is

```
    [s, sd, sdd] = tfunc(s0, sf, m);
```

Corresponding velocity and acceleration values may be obtained through the optional output arguments `qd` and `qdd`.

SYNTAX#2: `[q, qd, qdd] = mtraj(tfunc, q0, qf, T)` specifies the sequence of time through the vector `T` at which the discrete samples will be computed accordingly. The starting (0) and ending time are associated with `q0` and `qf`.

mstraj

generate a sequence of samples along the multi-segment LSPB curves joining a set of given via points and the starting point. The function is applicable to multi DOFs joints.

Syntax: `traj = mstraj(p, qdmax, q0, dt, tacc, qd0, qdf)` returns a sequence of samples `traj` along the multi-segment LSPB curves joining the initial joint values by the row vector `q0` and its via points by the matrix `p`. The argument `qdmax` in row vector is used to assign the velocity limit of each joint. If given in column vector format, it will be interpreted as the time duration of the segments. User may also specify the blending time for the transition between the each segment through the vector `tacc`. If a scalar is given for this argument, the value will be applied to all segments. `dt` is the time step used to proceed from the current sample to the next. Additionally, the initial and final joint velocity vector may be specified through `qd0` and `qdf`.

jtraj

generate a sequence of samples along the quintic polynomial curves joining the designated initial and final angles of multi DOFs joints.

Syntax#1: `[q, qd, qdd] = jtraj(q0, qf, m, qd0, qdf)` returns a sequence of equally time-spaced `m` samples `q` along the quintic polynomial curves joining the specified initial and final joint vectors `q0` and `qf`. Optionally, the initial and final joint velocity vectors `qd0` and `qdf` may be specified.

Syntax#2: `[q, qd, qdd] = jtraj(q0, qf, T, qd0, qdf)` specifies the sequence of time through the vector `T` at which the discrete samples will be computed accordingly. The starting (0) and ending time are associated with `q0` and `qf`.

ctraj

generate a sequence of samples in $SE3$ along the straight line joining the designated initial and final pose.

Syntax#1: `tc = ctraj(T0, T1, n)` returns a sequence of equally time-spaced `n` samples `tc` in $SE3$ joining the specified initial and final pose `T0` and `T1`. The motion created follows the trapezoidal velocity profile along the straight line.

Syntax#2: `tc = ctraj(T0, T1, s)` specifies the sequence of fractional distances in the range $[0, 1]$ for the samples to be computed.

tranimate

produce the animation of a coordinate frame moving from the initial pose to the final pose.

Syntax#1: `tranimate(p1, p2)` animates the motion of a Cartesian coordinate frame starting from the initial pose `p1` to the final pose `p2`. The pose may be specified using the homogeneous transformation matrix, the rotation matrix, or the Quaternion.

Syntax#2: `tranimate(pseq)` animates the motion of a Cartesian coordinate frame moving along the sequence of specified poses.

Robot Creation

Link

create a link object representing the physical robot linkage with the associated properties such as the kinematic and dynamic parameters. Linkages will comprise the robot. For the matter of this introductory course, only kinematic properties of the link are of concern. To create a link object `L` with the corresponding DH parameters, the following syntax may be useful.

```
    L = Link([THETA D A ALPHA SIGMA], ‘modified’)
```

The arguments `THETA`, `D`, `A`, and `ALPHA` are the four DH parameters of joint angle θ, joint displacement d, link length a, and link twist α. `SIGMA` is the flag telling that the joint is revolute (0) or prismatic (1). If not specified, the revolute joint is assumed. Variation of the DH parameters used in the course is the `‘modified’` form and needed to be specified, otherwise the `‘standard’` form is assumed. Accordingly, an extra constant transformation is needed to bring the frame at the robot last joint to its end effector ($\{n\}$ to $\{E\}$).

SerialLink

create a serial robot object representing the physical serial robot. The robot is created by specifying the array of linkages sorting in an ascending order from the base to the end effector. Taking the 3 DOFs articulated arm of Fig. 3.1 as an example to create the virtual robot in MatLAB®, first its linkages are created. According to the DH parameters in Table 3.2 and the numerical values of the link length $h = 1.5$, $l_1 = 1$ and $l_2 = 0.8$,

```
    L(1) = Link([0 1.5 0 0], ‘modified’)
    L(2) = Link([0 0 0 0], ‘modified’)
    L(3) = Link([0 0 0 pi/2], ‘modified’)
    L(4) = Link([0 0 1 0], ‘modified’)
    L(5) = Link([0 0 0.8 0], ‘modified’)
```

Note the exact values of `THETA` are not important here since the joints are revolute type and so the values will have to be input as the arguments for each command.

The robot may now be created and kept in `ArticulatedBot` variable with the following construct specifying the array of linkages `L` as the input;

```
    ArticulatedBot = SerialLink(L)
```

Details of the robot will be printed out as well. `SerialLink` object holds several methods. Some useful methods are listed.

SerialLink.plot

visually draw or plot the robot.

Syntax: `Robot.plot(q)` receives the vector of joint variables, being θ or d for the revolute or prismatic joint, and displays the graphical pose of `Robot` rendering the simple cylinders for the robot linkages. Furthermore, the function can animate the robot motion along the given matrix `q` composing of the robot joint trajectories. For the articulated arm,

```
    ArticulatedBot.plot([0 pi/6 pi/4 -pi/9 0])
```

draws the robot having the posture corresponding to $\theta_1 = 30°$, $\theta_2 = 45°$, and $\theta_3 = -20°$ as shown in Fig. 8.9. Note that the first and the last joint angles have been assigned the value of $0°$ since they belong to constant transformations from $\{B\}$ to $\{0\}$ and from $\{3\}$ to $\{E\}$, respectively.

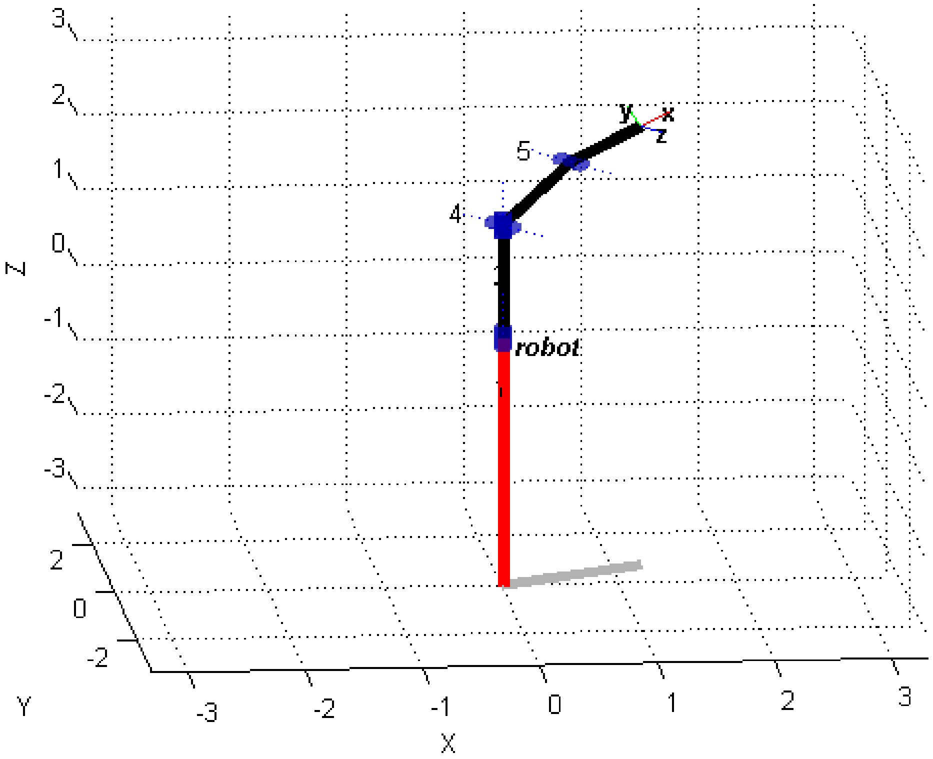


Figure 8.9: A posture of the articulated robot rendered by MatLAB® with Robotics Toolbox.

SerialLink.fkine

calculate the forward kinematics of the robot.

SYNTAX: `T = Robot.fkine(q)` determines the homogeneous transformation matrix `T` of the robot end effector frame according to the specified joint variable vector `q`. Rather, if the matrix `q` composing of the robot joint trajectories are given, the resulting matrix `T` will be a 3-dimensional matrix arraying the respective homogeneous transformation matrices.

SerialLink.ikine

calculate the inverse kinematics of the robot.

SYNTAX: `q = Robot.ikine(T, Q0, M)` determines the joint variable vector `q` according to the specified homogeneous transformation matrix `T` of the robot end effector frame, using the Jacobian matrix iterative method. An optional estimation of the joint vector `Q0` may be specified to obtain other possible solutions. For the case when `Robot` possesses fewer DOFs than six, the 6-element mask vector `M` corresponding to the translation in and rotation about the x, y, z-axis of the end effector frame must be specified to inform the function that which DOF(s) will be ignored. The ignored DOF must be set to 0 and the number of non-zero elements should be equal to the number of the robot DOF. Note the function might not be able to determine the solution due to the divergence of the answer during the iteration cycles.

SerialLink.jacob0

calculate the geometric Jacobian matrix of the robot, expressed in the world frame.

SYNTAX: `j0 = Robot.jacob0(q)` returns the geometric Jacobian matrix `j0` associated with `Robot` pose specified by the joint variable vector `q`. The matrix is expressed in the robot world frame.

SerialLink.jacobn

calculate the geometric Jacobian matrix of the robot, expressed in the end effector frame.

SYNTAX: `jn = Robot.jacobn(q)` returns the geometric Jacobian matrix `jn` associated with `Robot` pose specified by the joint variable vector `q`. The matrix is expressed in the robot end effector frame.

Bibliography

[1] P. Pitakwatchara. *Fundamentals of Robotics: Mechanics of the Serial Manipulators*. Chulalongkorn University Press, Bangkok, 2014. (in Thai)

[2] R. P. Paul. *Robot Manipulators: Mathematics, Programming and Control*. MIT Press, Cambridge, 1982.

[3] M. W. Spong, S. Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. John Wiley & Sons, New York, 2006.

[4] J. J. Craig. *Introduction to Robotics: Mechanics and Control*. Third edition, Pearson Prentice Hall, New Jersy, 2005.

[5] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics: Modelling, Planning, and Control*. Springer-Verlag, London, 2009.

[6] P. I. Corke, "A Robotic Toolbox for MatLAB®," *IEEE Robotics and Automation Magazine*, Vol. 3, No. 1, pp. 24-32, 1996.

[7] J. Denavit and R. S. Hartenberg, "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices," *ASME Journal of Applied Mechanics*, Vol. 22, pp. 215-221, 1955.

[8] D. L. Pieper and B. Roth, "The Kinematics of Manipulators Under Computer Control," *Proceedings of the Second International Congress on Theory of Machines and Mechanisms*, Vol. 2, pp. 159-169, 1969.

[9] T. Yoshikawa, "Manipulability of Robotic Mechanisms," *International Journal of Robotics Research*, Vol. 4, No. 2, pp. 3-9, 1985.

[10] H. Asada, "A Geometrical Representation of Manipulator Dynamics and Its Application to Arm Design," *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 105, pp. 131-142, 1983.