CS 4750 Foundations of Robotics

Cornell University

Kinematics*

September 6, 2016

*These notes are mainly based on [1], [2] and [3].

1 The Rigid Body Assumption

Let G represent a *global reference frame*, i.e., an inertial frame fixed in a 3-dimensional Cartesian space $\mathbb{R}^3$ at a point O with axes defined by the unit vectors $\{\hat{x}_G, \hat{y}_G, \hat{z}_G\}$.

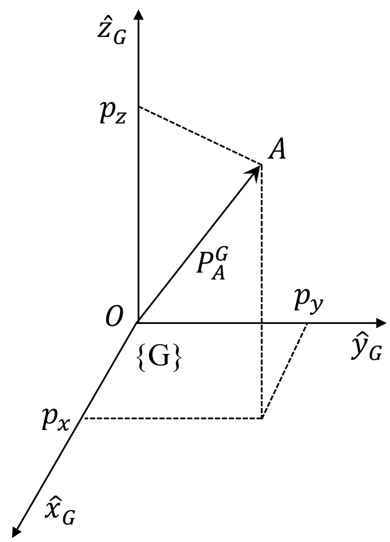


Figure 1: A position vector ${}^G P_A$, defined with respect to the reference frame $\{G\}$.

The kinematic state of a *particle* can be fully specified by its *position* with respect to G. The position of a particle A with respect to G can be given by a 3-vector P_A^G, which is called a *position vector*:

$$P_A^G = \begin{bmatrix} p_x & p_y & p_z \end{bmatrix}^T \tag{1}$$

where p_x, p_y and p_z denote the projections of P_A^G on the corresponding axes of G (see Fig. 1).

A *rigid body* can be defined as a collection of particles such that the distance between any two particles remains fixed, regardless of any motions of the body or forces exerted on the body. In other words, a rigid body is an idealized, completely undeformable solid body. In robotics, we use the rigid body assumption to model a robot's whole body or its components, e.g. the links of a manipulator.

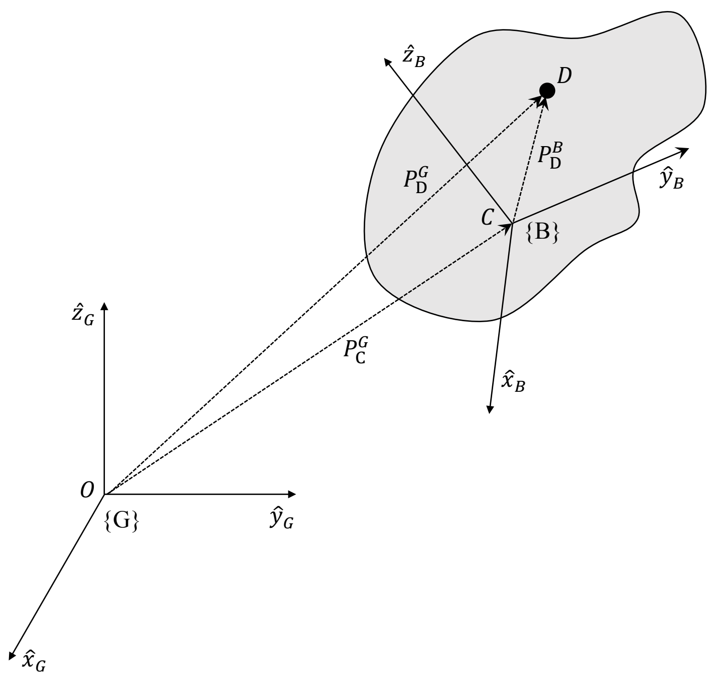


Figure 2: A rigid body, with a frame $\{B\}$ attached to it, defined with respect to the global frame $\{G\}$.

In order to fully specify the kinematic state of a rigid body, we need descriptions of both its *position* and its *orientation*. To do that, we attach a frame $\{B\}$ to the body, called the *body frame* (see Fig. 2). Let $\{B\}$ be attached at a point[1] C and comprise of the axes defined by the unit vectors $\{\hat{x}_B, \hat{y}_B, \hat{z}_B\}$ (these vectors are expressed with respect to $\{G\}$ but we omit the superscript notation for brevity). The position vector P_C^G represents the position of the body with respect to $\{G\}$, whereas the orientation of $\{B\}$ with respect to $\{G\}$ represents the orientation of the body.

The motion of a particle in Euclidean space can be fully described by its position with respect to an inertial Cartesian coordinate frame at every instant in time. A *rigid motion* of an object is a continuous movement of the particles in the object, such that the distance between any two of its particles remains invariant in time. The net movement of an object via rigid motion is called a *rigid displacement*. A rigid displacement may be the result of *translation*, *rotation*, or both at the same time.

[1]A body frame can be attached anywhere in the body, although for most applications the center of mass is usually selected.

2 Rotation Representations

2.1 The Rotation Matrix

Following the previous definitions, the most obvious way to represent the orientation of a body is through the unit vectors of the body frame, expressed with respect to the global frame. A common way to do that is by forming a 3×3 matrix, referred to as the *Rotation matrix*:

$$R_B^G = \begin{bmatrix} \hat{x}_B^G & \hat{y}_B^G & \hat{z}_B^G \end{bmatrix} \tag{2}$$

The columns of R_B^G are unit vectors, orthogonal to each other (since they form a coordinate frame); such a matrix is called *orthogonal*.

As noted earlier, a position vector is formed by its projections to a selected reference frame. Therefore, each column in the rotation matrix contains the projections of the corresponding unit vector (that is also an axis for $\{B\}$) to $\{G\}$. The projection of a vector onto another vector is simply the dot product of the two vectors; hence R_B^G can be written as follows:

$$R_B^G = \begin{bmatrix} \hat{x}_B \cdot \hat{x}_G & \hat{y}_B \cdot \hat{x}_G & \hat{z}_B \cdot \hat{x}_G \\ \hat{x}_B \cdot \hat{y}_G & \hat{y}_B \cdot \hat{y}_G & \hat{z}_B \cdot \hat{y}_G \\ \hat{x}_B \cdot \hat{z}_G & \hat{y}_B \cdot \hat{z}_G & \hat{z}_B \cdot \hat{z}_G \end{bmatrix} \tag{3}$$

The dot product of a pair of unit vectors yields the cosine of the angle between them; for this reason the elements of a rotation matrix are often referred to as *direction cosines* and R_B^G can be rewritten as:

$$R_B^G = \begin{bmatrix} \cos(\hat{x}_B, \hat{x}_G) & \cos(\hat{y}_B, \hat{x}_G) & \cos(\hat{z}_B, \hat{x}_G \\ \cos(\hat{x}_B, \hat{y}_G) & \cos(\hat{y}_B, \hat{y}_G) & \cos(\hat{z}_B, \hat{y}_G) \\ \cos(\hat{x}_B, \hat{z}_G) & \cos(\hat{y}_B, \hat{z}_G) & \cos(\hat{z}_B, \hat{z}_G) \end{bmatrix} \tag{4}$$

2.1.1 Properties of the Rotation Matrix

Since the dot product is commutative, we can see that:

$$R_G^B = \begin{bmatrix} \hat{x}_G^B & \hat{y}_G^B & \hat{z}_G^B \end{bmatrix} = \begin{bmatrix} \hat{x}_G \cdot \hat{x}_B & \hat{y}_G \cdot \hat{x}_B & \hat{z}_G \cdot \hat{x}_B \\ \hat{x}_G \cdot \hat{y}_B & \hat{y}_G \cdot \hat{y}_B & \hat{z}_G \cdot \hat{y}_B \\ \hat{x}_G \cdot \hat{z}_B & \hat{y}_G \cdot \hat{z}_B & \hat{z}_G \cdot \hat{z}_B \end{bmatrix} = \begin{bmatrix} (\hat{x}_B^G)^T \\ (\hat{y}_B^G)^T \\ (\hat{z}_B^G)^T \end{bmatrix} = (R_B^G)^T \tag{5}$$

This suggests that the inverse of a rotation matrix is equal to its transpose, which can be verified as follows:

$$R_G^B R_B^G = (R_B^G)^T R_B^G = \begin{bmatrix} (\hat{x}_B^G)^T \\ (\hat{y}_B^G)^T \\ (\hat{z}_B^G)^T \end{bmatrix} \begin{bmatrix} \hat{x}_B^G & \hat{y}_B^G & \hat{z}_B^G \end{bmatrix} = I_3 \tag{6}$$

Hence, indeed:

$$R_G^B = (R_B^G)^T = (R_B^G)^{-1} \tag{7}$$

Another notable property of a rotation matrix R_G^B is that

$$\det(R_G^B) = 1 \tag{8}$$

under the assumption of right-handed coordinate systems (in general it can be $\det(R_G^B) = \pm 1$).

Example 1. Planar Rotation

Consider the case of a planar rotation; assume that we rotate a frame around its z axis, for an angle θ (see Fig. 3), to transition from an initial orientation 0 to a final orientation 1. Using eq. (4), we derive the following expression:

$$R_z(\theta) = R_1^0 = \begin{bmatrix} \hat{x}_1^0 & \hat{y}_1^0 & \hat{z}_1^0 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{9}$$

where the notation $R_z(\theta)$ is used for convenience, to denote rotation about the z axis for an angle θ.

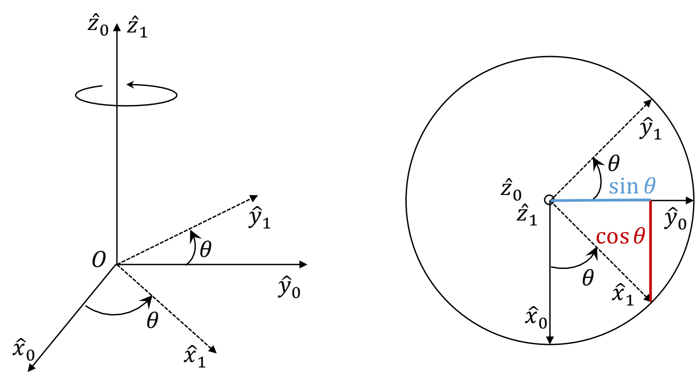


Figure 3: Example of a planar rotation about z axis from a side (left) and top view.

For convenience and for future reference, we also give the expressions for the rotation matrices $R_y(\theta)$:

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{10}$$

and $R_x(\theta)$:

$$R_y(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \tag{11}$$

$$\tag{12}$$

2.1.2 Interpretations of the Rotation Matrix

There are three main interpretations of the rotation matrix:

1. An *expression* of the orientation of a coordinate frame with respect to another frame.
2. A coordinate *transformation* associating the coordinates of a point P in two different frames.
3. An *operator* that rotates a vector to a new orientation in the same coordinate frame.

In the beginning of this section, we explained how the rotation matrix, following its definition, can be used to describe the orientation of a frame with respect to another (see Fig. 4). Here we will demonstrate how the rotation matrix can be used to represent a point in different cartesian coordinate frames and also to rotate a vector in the same frame, with two simple examples.

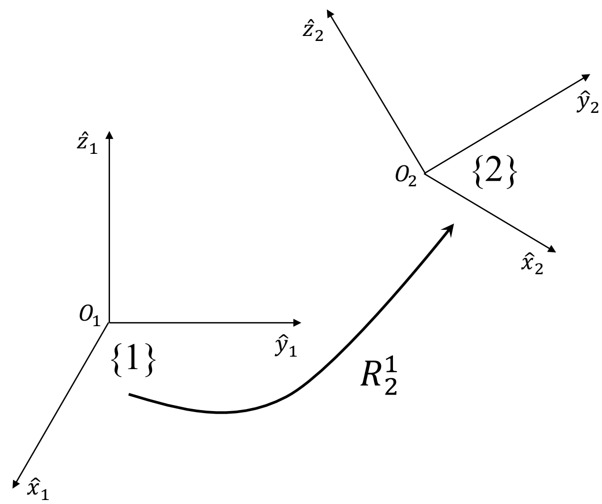


Figure 4: The orientation of frame $\{2\}$ with respect to frame $\{1\}$ can be expressed with the rotation matrix R_2^1.

Example 2. Describing a point in different frames

Assume we are given the coordinates of point A with respect to the frame $\{2\}$ (defined by the unit vectors $\hat{x}_2$, $\hat{y}_2$, $\hat{z}_2$ Fig. 5) in the form of a position vector $P_A^2 = \left[u, v, w\right]^T$. Our goal is to determine the coordinates of A with respect to frame $\{1\}$, i.e., the position vector P_A^1.

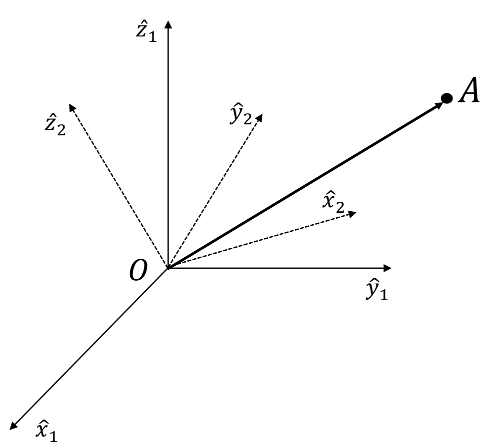


Figure 5: The rotation matrix can be used to express a point A with respect to different coordinate frames.

The elements of P_A^2 are its projections on the axes of frame $\{2\}$; therefore we can rewrite P_A^2 as:

$$P_A^2 = u\hat{x}_2 + v\hat{y}_2 + w\hat{z}_2 \tag{13}$$

For P_A^1, we can also write a similar expression:

$$P_A^1 = \begin{bmatrix} P_A^2 \cdot \hat{x}_1 \\ P_A^2 \cdot \hat{y}_1 \\ P_A^2 \cdot \hat{z}_1 \end{bmatrix} \tag{14}$$

From equations (14) and (13), we can write:

$$\begin{aligned} P_A^1 &= \begin{bmatrix} (u\hat{x}_2 + v\hat{y}_2 + w\hat{z}_2) \cdot \hat{x}_1 \\ (u\hat{x}_2 + v\hat{y}_2 + w\hat{z}_2) \cdot \hat{y}_1 \\ (u\hat{x}_2 + v\hat{y}_2 + w\hat{z}_2) \cdot \hat{z}_1 \end{bmatrix} \\ &= \begin{bmatrix} u\hat{x}_2 \cdot \hat{x}_1 + v\hat{y}_2 \cdot \hat{x}_1 + w\hat{z}_2 \cdot \hat{x}_1 \\ u\hat{x}_2 \cdot \hat{y}_1 + v\hat{y}_2 \cdot \hat{y}_1 + w\hat{z}_2 \cdot \hat{y}_1 \\ u\hat{x}_2 \cdot \hat{z}_1 + v\hat{y}_2 \cdot \hat{z}_1 + w\hat{z}_2 \cdot \hat{z}_1 \end{bmatrix} \\ &= \begin{bmatrix} \hat{x}_2 \cdot \hat{x}_1 & \hat{y}_2 \cdot \hat{x}_1 & \hat{z}_2 \cdot \hat{x}_1 \\ \hat{x}_2 \cdot \hat{y}_1 & \hat{y}_2 \cdot \hat{y}_1 & \hat{z}_2 \cdot \hat{y}_1 \\ \hat{x}_2 \cdot \hat{z}_1 & \hat{y}_2 \cdot \hat{z}_1 & \hat{z}_2 \cdot \hat{z}_1 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \end{aligned} \tag{15}$$

The matrix in the final step of eq. (15) is the rotation matrix R_2^1 and therefore:

$$P_A^1 = R_2^1 P_A^2 \tag{16}$$

Eq. (16) is a useful expression that allows us to express the coordinates of a point (or a position vector) from one frame to another.

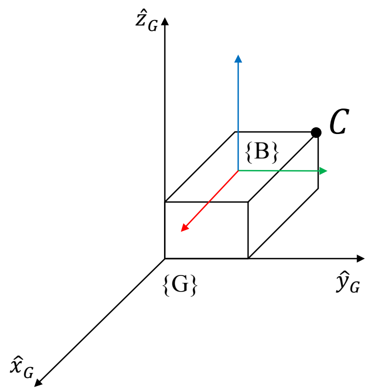
Example 3. Rotating a vector

Consider the block of Fig. 6. The block is rotated from the configuration of Fig. 6a to the configuration of Fig. 6b; this rotation corresponds to an angle of π radians about axis $\hat{z}_0$. A body frame $\{B\}$, attached at the block, is also rotated as shown in Fig. 6b. From the derivation of eq. (9), regarding rotation about the z axis, we can derive the rotation matrix representing the *final* orientation of the body frame (that also represents the final orientation of the block) with respect to the global frame $\{G\}$:

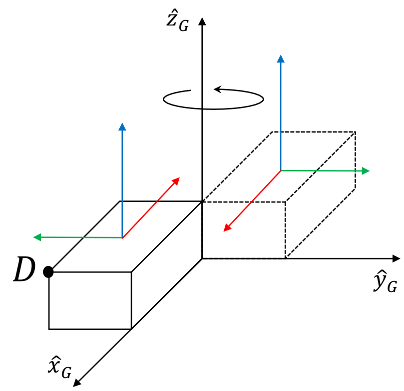
$$R_B^G = R_z(\pi) = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{17}$$

Using eq. (16), we can obtain an expression relating P_D^G with P_C^B:

$$P_D^G = R_B^G P_D^B \tag{18}$$



(a)



(b)

Figure 6: A rectangular block is rotated by π about $\hat{z}_0$, from the configuration of Fig. 6a to the configuration of Fig. 6b.

Note that the position vectors P_D^B and P_C^G are equal, since the body frame $\{B\}$ is rigidly attached at the block and the points C and D correspond to the same position on the block. Hence, we can rewrite eq. (18) as:

$$P_D^G = R_B^G P_C^G \tag{19}$$

Therefore, eq. (19) can relate the descriptions of a point that is rigidly attached on a body, before and after a rotation described by the rotation matrix. As a result, the rotation matrix can be used as an operator that performs *pure rotation* on a rigid body.

2.1.3 Expressing Rotations to different frames

A rotation matrix can be thought of as a change of basis from a frame to another. As noted earlier, the columns of the rotation matrix represent the unit vectors that define the new basis with respect to the old basis. It is often necessary to express a general linear transformation[2] with respect to a different frame. This can be performed with a *similarity transformation*:

$$A' = T^{-1}AT \tag{20}$$

where A is a linear transformation $A \in \mathbb{R}^{n\times n}$, expressed with respect to an initial basis and T is a matrix formed by a set of basic vectors defining another coordinate frame.

[2] A rotation matrix is an example of a linear transformation, when used as a rotation operator.

For example, assume that A is a linear transformation, expressed with respect to a frame $\{0\}$ and A' is the same transformation, expressed with respect to another frame $\{1\}$. Given the rotation matrix R_1^0, A can be expressed with respect to $\{1\}$ as:

$$A' = (R_1^0)^{-1} A R_1^0 \tag{21}$$

Therefore, this relationship allows us to express a rotation operation, described by a rotation matrix A with respect to a desired frame.

2.1.4 Composition of Rotations

2.1.4.1 Rotation with respect to the Current Frame

Most of the common robotic mechanisms are composed of several rigid bodies, attached to each other with joints of various kinds. As a result, to describe their kinematics, it is often necessary to superimpose the motions of their components. The typical approach to this problem is to attach a frame to each body, determine the relative displacements between successive bodies and compose those displacements into a net displacement. In this section we focus on rotations and specifically describe how to synthesize consecutive *rotational* displacements of frames. Each one of those rotational displacements is described as a rotation with respect to the *current frame*.

Consider three coordinate frames $\{0\}$, $\{1\}$ and $\{2\}$. Any point $p \in \mathbb{R}^3$ can be described with respect to each one of those frames through multiplication with an appropriate rotation matrix as we showed in the previous section. Consider for example the following relationships

$$p^0 = R_1^0 p^1 \tag{22}$$

$$p^1 = R_2^1 p^2 \tag{23}$$

$$p^0 = R_2^0 p^2 \tag{24}$$

Substituting eq. (23) into eq. (22), we derive:

$$p^0 = R_1^0 R_2^1 p^2 \tag{25}$$

Looking at eqs. (24) and (25), we notice that:

$$R_2^0 = R_1^0 R_2^1 \tag{26}$$

This simple example reflects the composition law for composing rotational transformations; consecutive rotational transformations can be synthesized by multiplying their corresponding rotation matrices.

2.1.4.2 Rotation with respect to the Fixed Frame

In some applications, we might need to perform a sequence of rotations, all expressed with respect to a given fixed frame instead of having a sequence of rotations about successive current frames. In those cases, in order to compose the individual rotations into a net rotation, we can use similarity transformations to reduce the problem to a sequence of rotations with respect to the current frame.

For example assume that the rotation matrix R_1^0, representing the rotation of a frame $\{1\}$ with respect to a fixed frame $\{0\}$ is available. Consider also another rotation R with respect to $\{0\}$. We can transform R into a rotation R_2^1 by taking the following similarity transformation:

$$R_2^1 = (R_1^0)^{-1} R R_1^0 \tag{27}$$

Then, determining the total rotation R_2^0 can be done as shown in the previous section:

$$R_2^0 = R_1^0 (R_1^0)^{-1} R R_1^0 = R R_1^0 \tag{28}$$

Therefore, when a rotation R is performed with respect to the global frame, it can be incorporated in a net rotation by *premultiplying* the current rotation matrix by R.

2.2 Rotation Parameterizations

From the definition of a rotation matrix R_1^0, it follows that:

$$\|\hat{x}_1^0\| = \|\hat{y}_1^0\| = \|\hat{z}_1^0\| \tag{29}$$

and

$$\hat{x}_1^0 \cdot y_1^0 = \hat{x}_1^0 \cdot z_1^0 = \hat{y}_1^0 \cdot z_1^0 \tag{30}$$

These are 6 equations, constraining the 9 elements of the rotation matrix. Mathematically, this indicates that only three of those elements are independent quantities, which represents the fact that a rigid body possesses at most 3 rotational degrees of freedom. Therefore, any rotation can be represented with 3 independent quantities. In this section, we present a few notable rotation parameterizations: the Euler angles, the axis-angle pair and the quaternions.

2.2.1 Euler Angles

The Euler angles parameterization is one of the most common ways to represent rotations in robotics. The main idea behind Euler angles is to decompose a rotation into a sequence of three successive rotations. Consider a fixed frame $\{0\}$ and a

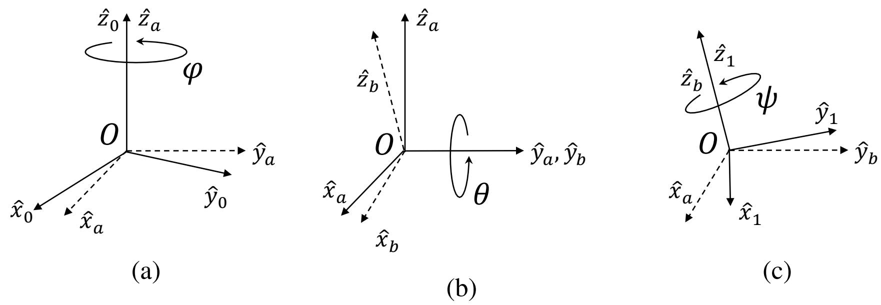


Figure 7: Euler Angles (ϕ, θ, ψ), representing the orientation of frame $\{1\}$ with respect to frame $\{0\}$.

rotated frame $\{1\}$. We can represent the orientation of $\{1\}$ with respect to $\{0\}$ with a set of Euler angles (ϕ, θ, ψ) as follows (the steps are depicted in Fig. 7):

1. Rotate $\{0\}$ about the z-axis by the angle ϕ to derive a new frame $\{a\}$.
2. Rotate $\{a\}$ about its y-axis by the angle θ to derive a new frame $\{b\}$.
3. Rotate $\{b\}$ about its z-axis by the angle ψ.

The aforementioned rotations can be represented as rotation matrices $R_z(\phi)$, $R_y(\theta)$, $R_z(\psi)$ that describe rotations *with respect to the current frame*. They can be composed into a rotation matrix[3] R_{ZYZ} as follows:

$$R_{ZYZ} = R_z(\phi)R_y(\theta)R_z(\psi) \tag{31}$$

$$= \begin{bmatrix} c_\phi & -s_\phi & 0 \\ s_\phi & c_\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} = \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{32}$$

$$= \begin{bmatrix} c_\phi c_\theta c_\psi - s_\phi c_\psi & -c_\phi c_\theta s_\psi - s_\phi c_\psi & c_\phi s_\theta \\ s_\phi c_\theta c_\psi + c_\phi s_\psi & -s_\phi c_\theta s_\psi + c_\phi c_\psi & s_\phi s_\theta \\ -s_\theta c_\psi & s_\theta s_\psi & c_\theta \end{bmatrix} \tag{33}$$

where we used the notation c_{angle}, s_{angle} to denote the $\cos(angle)$ and $\sin(angle)$ respectively, with $angle \in \{\phi, \theta, \psi\}$. The rotation matrix R_{ZYZ} is called the ZYZ-Euler angle transformation. Different conventions for Euler angles can be defined, depending on the sequence of rotations considered (how many?).

[3]The subscript denotes the angle convention. Different conventions for sequence of individual rotations can be defined.

Many times in robotics, we face the opposite problem, i.e., we need to determine a set of Euler angles, given an orientation, in the form of a rotation matrix. Consider a given rotation matrix in its general form:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{34}$$

We break the problem into two cases:

- Case 1: Not both r_{13} and r_{23} are zero.
  Then from eq. (33), $s_\theta \neq 0$, which implies that $c_\theta \neq \pm 1$. Consequently, (a) r_{31} and r_{32} cannot be both zero and (b) $r_{33} \neq \pm 1$. From the trigonometric *Pythagorean identity*, we can derive the expression $s_\theta = \pm\sqrt{1 - r_{33}^2}$. Therefore

  $$\theta = \text{Atan2}\left(r_{33}, \sqrt{1 - r_{33}^2}\right) \tag{35}$$

  or

  $$\theta = \text{Atan2}\left(r_{33}, -\sqrt{1 - r_{33}^2}\right) \tag{36}$$

  If we choose eq. (35), then it follows that $s_\theta > 0$ and therefore:

  $$\phi = \text{Atan2}\left(r_{13}, r_{23}\right) \tag{37}$$

  $$\psi = \text{Atan2}\left(-r_{31}, r_{32}\right) \tag{38}$$

  Otherwise, if we choose eq. (36), then $s_\theta < 0$ and therefore:

  $$\phi = \text{Atan2}\left(-r_{13}, -r_{23}\right) \tag{39}$$

  $$\psi = \text{Atan2}\left(r_{31}, -r_{32}\right) \tag{40}$$

  Hence, depending on the sign of θ we get a different solution.

- Case 2: $r_{13} = 0$ and $r_{23} = 0$.
  Since R is *orthogonal*, its columns are unit vectors; therefore $r_{33} = c_\theta = \pm 1$, which also implies that $s_\theta = 0$. Hence $r_{31} = r_{32} = 0$. Thus R has the following form:

  $$R = \begin{bmatrix} r_{11} & r_{12} & 0 \\ r_{21} & r_{22} & 0 \\ 0 & 0 & \pm 1 \end{bmatrix} \tag{41}$$

If $r_{33} = 1$, then $c_\theta = 1$ and $\theta = 0$. Substituting in eq. (33) and applying some trigonometric sum identities, we get:

$$R = \begin{bmatrix} c_\phi c_\psi - s_\phi c_\psi & -c_\phi s_\psi - s_\phi c_\psi & 0 \\ s_\phi c_\psi + c_\phi s_\psi & -s_\phi s_\psi + c_\phi c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_{\phi+\psi} & -s_{\phi+\psi} & 0 \\ s_{\phi+\psi} & c_{\phi+\psi} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (42)$$

Therefore, we can get the following expression for the sum $\phi + \psi$:

$$\phi + \psi = \text{Atan2}(r_{11}, r_{21}) = \text{Atan2}(r_{11}, -r_{12}) \quad (43)$$

which implies that there are infinitely many solutions for ϕ and ψ.

Finally, if $r_{33} = -1$, then in a similar way we derive:

$$R = \begin{bmatrix} -c_{\phi-\psi} & -s_{\phi-\psi} & 0 \\ s_{\phi-\psi} & c_{\phi-\psi} & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (44)$$

which leads to a similar expression for the difference of $\phi - \psi$:

$$\phi - \psi = \text{Atan2}(-r_{11}, -r_{12}) \quad (45)$$

This also implies that there are infinitely many solutions for ϕ and ψ.

The main weakness of the Euler angles representation arises from its main idea of representing the rotation with 3 numbers. This leads to singularities. For example, in the case of the case of the ZYZ parameterization, singularities occur at $R = I$. By singularities, we mean the lack of global, smooth solutions to the inverse problem of determining the Euler angles from a given rotation matrix. In particular, assuming euler angles (ϕ, θ, ψ) of the form $(\phi, 0, \psi)$ yields $R = I$. Therefore there are infinitely many representations of the identity rotation in the ZYZ Euler angle parameterization. Such singularities occur at all conventions, at different points.

2.2.2 Axis-Angle Representation

Another common way to represent rotations is by a *axis-angle* pair. This convention makes use of the fact that any rotation matrix can be represented by a single rotation about a suitable axis, by a suitable angle.

Consider a fixed frame $\{0\}$ and a unit vector $\hat{k} = \begin{bmatrix} k_x & k_y & k_z \end{bmatrix}^T$, expressed with respect to $\{0\}$, defining an axis (see Fig. 8). Assume that we want to perform a rotation about $\hat{k}$ by an angle θ. To represent this rotation we may derive a rotation

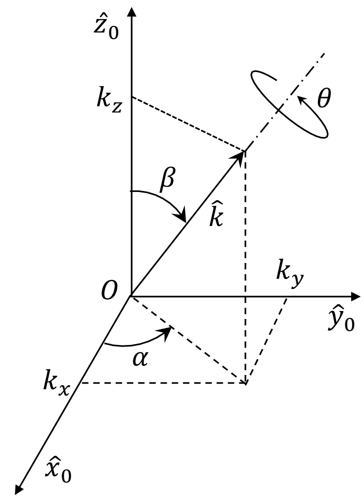


Figure 8: Axis-angle Representation.

matrix $R_k(\theta)$. There are many different ways to do so. For example, note that a rotational transformation of the form $R = R_z(\alpha)R_y(\beta)$ by appropriate angles α and β first about the z and then about the y axis may align the z-axis of $\{0\}$ with $\hat{k}$. Then, performing a rotation about $\hat{k}$ can be done with a similarity transformation[4]:

$$
\begin{aligned}
R_k(\theta) &= RR_z(\theta)R^{-1} && (46)\\
&= R_z(\alpha)R_y(\beta)R_z(\theta)(R_z(\alpha)R_y(\beta))^{-1} && (47)\\
&= R_z(\alpha)R_y(\beta)R_z(\theta)R_y^{-1}(\beta)R_z^{-1}(\alpha) && (48)\\
&= R_z(\alpha)R_y(\beta)R_z(\theta)R_y(-\beta)R_z(-\alpha) && (49)
\end{aligned}
$$

2.2.3 Quaternions

Quaternions can be thought of as a generalization of the complex numbers and can be used to represent rotations in a 3-dimensional cartesian space in the same way as complex can be used to represent planar rotations on the unit circle. Quaternions avoid the singularities of Euler angles, by using four parameters to represent a rotation.

A quaternion $Q \in \mathbb{Q} \subseteq R^4$, can be defined as the following vector:

$$Q = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k} \tag{50}$$

[4]This similarity transformation changes the basis of $R_z(\theta)$, from $\{0\}$ to $\hat{k}$.

with $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$ are called the fundamental quaternion units, $q_i \in \mathbb{R}, i = 0, 1, 2, 3$ where q_0 is the *scalar* component of Q and $\vec{q} = (q_1, q_2, q_3) \in \mathbb{R}^3$ is its *vector* component. We usually describe a quaternion as $Q = (q_0, \vec{q})$.

The set of quaternions $\mathbb{Q}$ forms a group with the operation of multiplication. The quaternion multiplication is *distributive* and *associative* but non-*commutative*. It also satisfies the following properties:

$$a\mathbf{i} = a\mathbf{i} \qquad a\mathbf{j} = \mathbf{j}a \qquad a\mathbf{k} = \mathbf{k}a \qquad a \in \mathbb{R} \tag{51}$$

$$\mathbf{i} \cdot \mathbf{i} = \mathbf{j} \cdot \mathbf{j} = \mathbf{k} \cdot \mathbf{k} = \mathbf{i} \cdot \mathbf{j} \cdot \mathbf{k} = -1 \tag{52}$$

$$\mathbf{i} \cdot \mathbf{j} = -\mathbf{j} \cdot \mathbf{i} = \mathbf{k} \qquad \mathbf{j} \cdot \mathbf{k} = -\mathbf{k} \cdot \mathbf{j} = \mathbf{i} \qquad \mathbf{k} \cdot \mathbf{i} = -\mathbf{i} \cdot \mathbf{k} = \mathbf{j} \tag{53}$$

The *conjugate* of a quaternion $Q = (q_0, \vec{q})$ is given by $Q^* = (q_0, -\vec{q})$ and the following equation is satisfied:

$$\|Q\|^2 = Q \cdot Q^* = q_0^2 + q_0^2 + q_2^1 + q_3^2. \tag{54}$$

The inverse of a quaternion is given by $Q^{-1} = Q^*/\|Q\|^2$ and it can be seen that $Q = (1, 0)$ is the identity element for quaternion multiplication.

The product of two quaternions $Q = (q_0, \vec{q})$ and $P = (p_0, \vec{p})$ can be shown to be equal to:

$$Q \cdot P = (q_0 p_0 - \vec{q} \cdot \vec{p},\ q_0 \vec{p} + p_0 \vec{q} + \vec{q} \times \vec{p}) \,. \tag{55}$$

In order to represent and compose rotations, we usually make use of the subset of *unit quaternions* $\{Q \in \mathbb{Q} : \|Q\| = 1\}$. There is a nice correspondence between the axis-angle representation, presented in the previous section and a unit quaternion. In particular, consider a rotation defined as the axis-angle pair $(\hat{k}, \theta)$. It can be shown that this rotation corresponds to the quaternion

$$Q = (\cos\theta/2, \hat{k}\sin\theta/2). \tag{56}$$

For the inverse operation, given a unit quaternion $Q = (q_0, \vec{q})$, we can extract the corresponding axis-angle pair as:

$$\theta = 2\cos^{-1} q_0 \qquad \hat{k} = \begin{cases} \frac{\vec{q}}{\sin(\theta/2)}, & \text{if } \theta \neq 0, \\ 0, & otherwise \end{cases}. \tag{57}$$

Given a rotation matrix R, we may derive its corresponding quaternion as:

$$q_0 = \frac{1}{2}\left(1 + r_{11} + r_2 + r_{33}\right)^{1/2} \qquad 0 \leq \theta \leq \pi \tag{58}$$

$$\vec{q} = \frac{1}{4q_0} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix}. \tag{59}$$

In order to compose consecutive rotations, we can multiply their corresponding quaternions in the same way as we did in the rotation matrix representations. Consider the quaternions Q_1^0 and Q_2^1, representing respectively the orientation of frame $\{1\}$ with respect to frame $\{0\}$ and the orientation of frame $\{2\}$ with respect to frame $\{1\}$. We may derive the quaternion expressing the orientation of frame $\{2\}$ with respect to frame $\{0\}$ by the multiplication

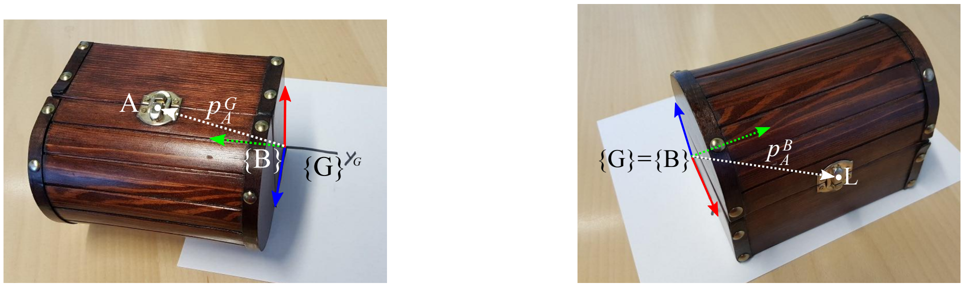
$$Q_2^0 = Q_1^0 \cdot Q_2^1. \tag{60}$$

2.3 Long Form Example

Example 4. Fixed Frame vs. Current Frame Euler Angles
Concepts reviewed: *express coordinates in a new frame*, *coordinate transformation*, *composing rotations*, *Euler angles*, *body frame vs. global frame*, *current frame vs. fixed frame*.
Problem: Suppose you want to open the treasure chest in Fig. 9a. which is lying on its back with latch at point A identified by coordinates p_A^B or p_A^G. You know the coordinates p_L^G of a point L, corresponding to the latch on the unrotated chest in Fig. 9b. You want to open the latch and reveal its treasure, but your robot is in the global frame $\{G\}$, and it cannot directly reach to the point A without transforming it into the $\{G\}$ frame first. You are given the orientation of the box with Euler angles $\phi = \frac{\pi}{2}, \theta = \frac{\pi}{2}, \psi = \frac{\pi}{2}$ using the Euler angle convention R_{XYZ}. However, we forgot to specify current or fixed frame. Determine which one is correct and find p_A^G.
Solution: We compute the Euler angle twice, as both fixed-frame and current-frame, and check which one fits the final orientation.

Note that this problem can be interpreted in two ways. First, we could interpret the problem as *expressing coordinates in a new frame*. We note that $p_A^B = p_L^B$ because the chest is a rigid body and points L and A both correspond to the same latch. Furthermore, $p_L^B = p_L^G$ because L is defined when the body's coordinate frame is aligned with the global one. Thus, we know p_A^B and merely need to express it in the global coordinate frame as $p_A^G = R_B^G p_A^B$.

Alternatively, we could interpret the problem as a *coordinate transformation* by recognizing that at some point in the past, the chest was rotated from its canonical configuration to the way we found it. At that time, the latch moved from the point L to the point A. Therefore, $p_A^G = R_B^G p_L^G$.



(a) Rotated body frame (b) Unrotated global frame

Figure 9: Problem statement: a treasure chest is configured as shown in (a), but the position of the latch at point A is only known in the global frame as in (b). Find p_A^B. The red, green, and blue axes represent the coordinate frame $\{B\}$ of the body of the treasure chest, and they are fixed to its three straight edges. The global frame $\{G\}$ is on the paper. In (b), the two frames are aligned.

Fixed Frame. To perform a rotation in the fixed frame (which in our case is the global frame), we premultiply. Thus,

$$R_B^G = R_{XYZ} = R_Z(\psi)R_Y(\theta)R_X(\phi) \tag{61}$$

$$= \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi & c_\phi \end{bmatrix}. \tag{62}$$

Let us consider these steps one at a time.

1. Premultiply

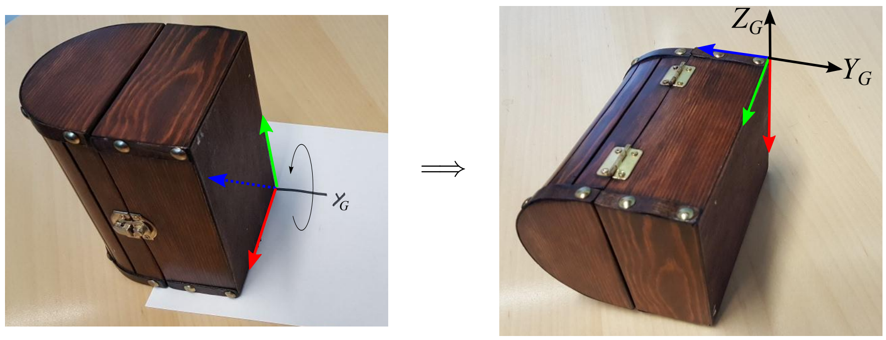
$$R_X(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi & c_\phi \end{bmatrix} \tag{63}$$

This rotates about the global $+X$ axis, which is also the body $+X$ axis.

2. Premultiply

$$R_Y(\theta) = \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \tag{64}$$

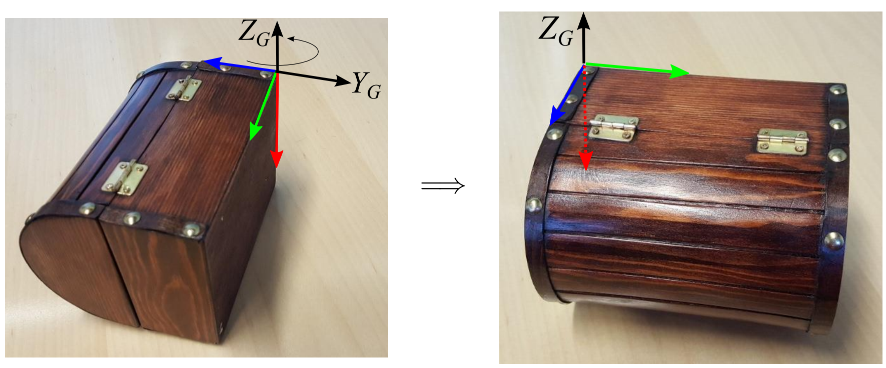
This rotates about the global $+Y$ axis, which is also the body $-Z$ axis.



3. Premultiply

$$R_Z(\psi) = \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{65}$$

This rotates about the global $+Z$ axis, which is also the body $-X$ axis.



$$
\begin{aligned}
p_A^G =& R_B^G p_A^B = R_{XYZ} p_A^B && (66)\\
=& \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi & c_\phi \end{bmatrix} p_A^B && (67)\\
=& \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} p_A^B && (68)\\
=& \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} p_A^B. && (69)
\end{aligned}
$$

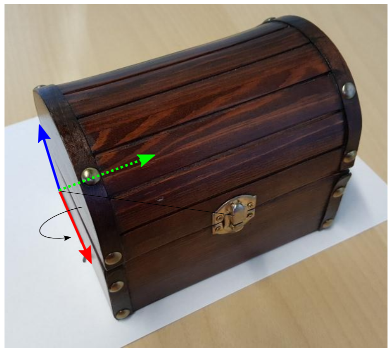
One can visually verify the computed rotation matrix by noting that its three column vectors match the three body-frame coordinate axes, when expressed in the global frame. This configuration does not match the desired final configuration. □

Current Frame. Rotations in the current frame are accomplished by postmultiplying. Thus,

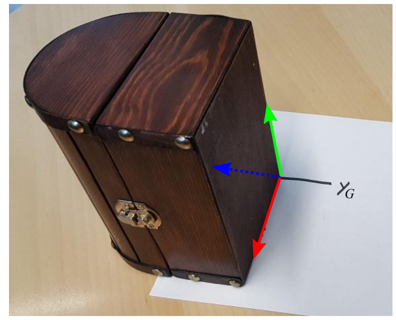
$$
\begin{aligned}
R_B^G =& R_{XYZ} = R_X(\phi) R_Y(\theta) R_Z(\psi) && (70)\\
=& \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi & c_\phi \end{bmatrix} \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}. && (71)
\end{aligned}
$$

Let us again consider these steps one at a time.

1. Postmultiply
$$
R_X(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi & c_\phi \end{bmatrix} \qquad (72)
$$
This rotates about the body $+X$ axis, which is also the global $+X$ axis.



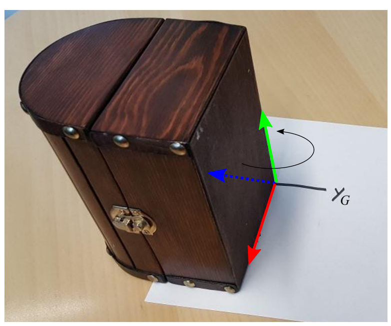
$\Longrightarrow$



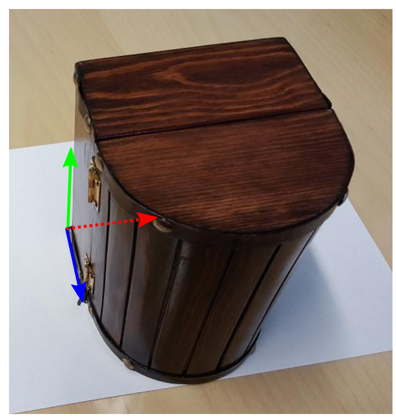
2. Postmultiply

$$R_Y(\theta) = \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \tag{73}$$

This rotates about the body $+Y$ axis, which is also the global $+Z$ axis.



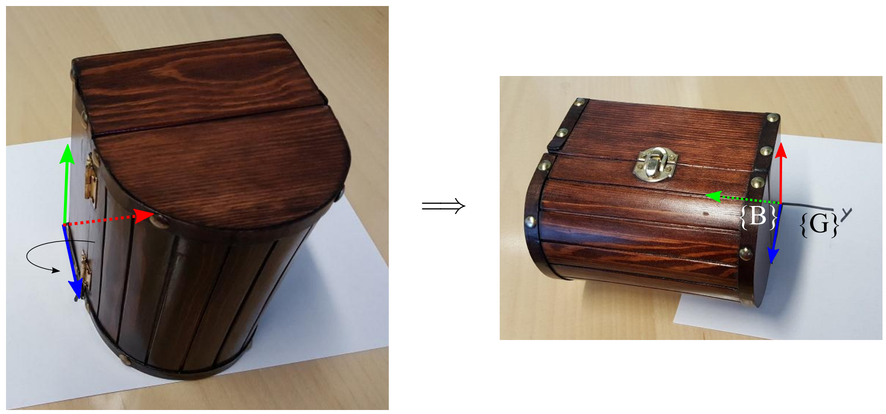
$\Longrightarrow$



3. Postmultiply

$$R_Z(\psi) = \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{74}$$

This rotates about the body $+Z$ axis, which is also the global $+X$ axis.



This time, we have arrived in the correct configuration, so we can conclude that this rotation is expressed in the current frame as

$$
\begin{aligned}
p_A^G =& R_B^G p_A^B = R_{XYZ} p_A^B & (75)\\
=& \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi & c_\phi \end{bmatrix} \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} p_A^B & (76)\\
=& \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} p_A^B & (77)\\
=& \begin{bmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{bmatrix} p_A^B. \quad \square & (78)
\end{aligned}
$$

3 Rigid Motion

As explained earlier, a *rigid motion* is in general the result of *translation* and *rotation*. So far we have focused on pure rotation. In this section we demonstrate how these concepts can be combined to fully describe the motion of a rigid body.

A rigid motion can be defined as a tuple (d, R), where $d \in \mathbb{R}^3$ and $R \in SO(3)$ (SO(3) is the group of all rotations about the origin of $\mathbb{R}^3$, under the operation of composition). The group of all rigid motions, also called the *Special Euclidean Group*, is defined as $SE(3) = \mathbb{R}^3 \times SO(3)$.

Consider the rotation matrix R_1^0 that describes the orientation of a frame $\{1\}$ with respect to a fixed frame $\{0\}$. Let p^1 denote the coordinates of a point $p \in \mathbb{R}^3$ with respect to the frame $\{1\}$ and $d^0 \in \mathbb{R}^3$ be a position vector from the origin of $\{0\}$ to the origin of $\{1\}$. Then the coordinates of p with respect to $\{0\}$ can be derived as follows:

$$p^0 = R_1^0 p^1 + d^0 \tag{79}$$

Consider now three frames $\{0\}$, $\{1\}$ and $\{2\}$. Let d_1^0 be the coordinates of the origin of frame $\{1\}$ with respect to frame $\{0\}$ and d_2^1 be the coordinates of the origin of frame $\{2\}$ with respect to frame $\{1\}$. Then for a point p we can write down the following equations:

$$p^1 = R_2^1 p^2 + d_2^1 \tag{80}$$

and

$$p^0 = R_1^0 p^1 + d_1^0 \tag{81}$$

Composing these equations we get the following expression for p^0:

$$\begin{align}
p^0 &= R_1^0 \left(R_2^1 p^2 + d_2^1\right) + d_1^0 \tag{82}\\
&= R_1^0 R_2^1 p^2 + R_1^0 d_2^1 + d_1^0 \tag{83}\\
&= R_2^0 p^2 + (d_2^0 + d_1^0) \tag{84}
\end{align}$$

Eq. (84) demonstrates how a rigid motion can be decomposed into a rotation and a translation and introduces the main idea behind the *homogeneous transformation*, a unified representation that describes a rigid motion, consisting of a translation and a rotation.

3.1 The Homogeneous Transformation

The homogeneous transformation is a matrix representation of a rigid motion that allows us to reduce the composition of rigid motions into matrix multiplications. Consider a rotation matrix R_1^0 representing the orientation of a frame $\{1\}$ with respect to a frame $\{0\}$ and a vector d_1^0 describing the coordinates of the origin of $\{1\}$ with respect to the origin of $\{0\}$. The homogeneous transform describing the the *rigid motion* (position and orientation) of the frame $\{1\}$ with respect to frame $\{0\}$ is defined as:

$$T_1^0 = \begin{bmatrix} R_1^0 & d_1^0 \\ \mathbf{0} & 1 \end{bmatrix} \tag{85}$$

where **0** is a zero row vector.

It can be shown that the inverse transformation, i.e., the transformation describing the *rigid motion* of frame $\{0\}$ with respect to frame $\{1\}$ is given by the following equation:

$$(T_1^0)^{-1} = T_0^1 = \begin{bmatrix} (R_1^0)^T & -(R_1^0)^T d_0^1 \\ \mathbf{0} & 1 \end{bmatrix} \tag{86}$$

Homogeneous transformations can be used to perform rigid operations to position vectors, in the same way that rotation matrices were used to reorient vectors. However, since the homogeneous transformation is a matrix 4×4, in order to define such an operation in the form of a multiplication, we need to augment a given position vector p as $P = \begin{bmatrix} p^T & 1 \end{bmatrix}^T$. This augmented version of p is called a homogeneous representation.

As an example, consider the homogeneous transform T_1^0 and a position vector p^1, expressing the position of point p with respect to the frame $\{1\}$. Upon augmenting p^1 into $P^1 = \begin{bmatrix} (p^1)^T & 1 \end{bmatrix}^T$, we can derive an expression for P^0 as:

$$P^0 = T_1^0 P^1 \tag{87}$$

The composition of homogeneous transforms follows the same rules as the composition of rotations. Consider a homogeneous transformation T_1^0 relating frame $\{1\}$ with frame $\{0\}$ and a second transformation T relative with the current frame. These transformations can be composed to form T_2^0 as:

$$T_2^0 = T_1^0 T. \tag{88}$$

On the other hand, if T is also expressed with respect to the fixed frame, then, as in the case of *rotation with respect to the fixed frame*, it can be shown that:

$$T_2^0 = T T_1^0. \tag{89}$$

4 Kinematics

4.1 Manipulator Kinematics and Forward Kinematic Chains

A robot manipulator is composed of a set of links connected together by joints. In this section, we only concern about simple joints such as a revolute joint or a prismatic joint. In this case , the joint has only a single degree-of-freedom (DOF)of motion: the angle of rotation in the case of a revolute joint, and the amount of linear displacement in the case of a prismatic joint.

With the assumption that each joint has 1 DOF, the action of each joint can be described by a single real number; the angle of rotation in the case of a revolute joint or the displacement in the case of a prismatic joint. The objective of forward kinematic analysis is to determine the cumulative effect of the entire set of joint variables, that is, to determine the position and orientation of the end effector given the values of these joint variables.

Manipulators consist of nearly rigid links, which are connected by joints that allow relative motion of neighboring links. The links are numbered starting from the immobile base of the arm, which might be called link 0. The first moving body is link 1, and so on, out to the free end of the arm, which is link n. In order to position an end-effector generally in 3D space, a minimum of six joints is required. For the purposes of obtaining the kinematic equations of the mechanism, a link is considered only as a rigid body that defines the relationship between two neighboring joint axes of a manipulator. Joint axes are defined by lines in space. Joint axis i is defined by a line in space, or a vector direction, about which link i rotates relative to link i-1. It turns out that, for kinematic purposes, a link can be specified with two numbers, which define the relative location of the two axes in space.

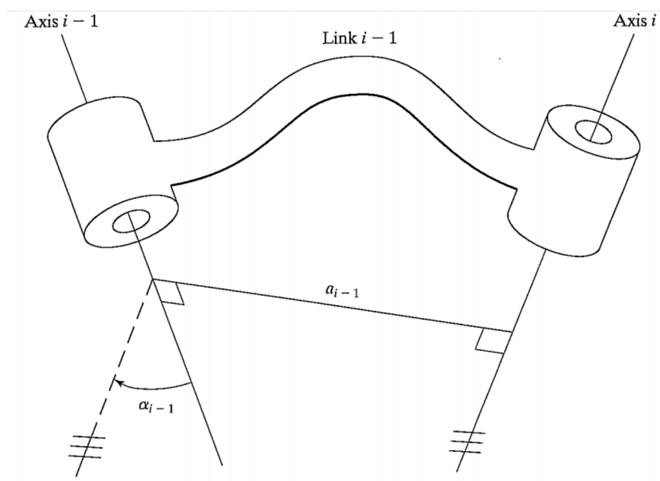


Figure 10: The kinematic function of a link is to maintain a relationship between the two joint axes it supports. This relationship can be described with two parameters: the link length, a, and the link twist, α. [1]

For any two axes in 3D space, there exists a well-defined measure of distance between them. This distance is measured along a line that is mutually perpendicular to both axes. This mutual perpendicular always exists; it is unique except when both axes are parallel, in which case there are many mutual perpendiculars of equal length. Figure 10 shows link i-1 and the mutually perpendicular line along which the link length, a_{i-1},is measured. Another way to visualize the link parameter a_{i-1} is to imagine an expanding cylinder whose axis is the joint i-1 axis —- when it just touches joint axis i, the radius of the cylinder is equal to a_{i-1}

The second parameter needed to define the relative location of the two axes is called the link twist. If we imagine a plane whose normal is the mutually perpendicular line just constructed, we can project the axes i-1 and i onto this plane and measure the angle between them. This angle is measured from axis i-1 to axis i in the right-hand sense about a_{i-1}. We will use this definition of the twist of link i-1, α_{i-1}. In Fig 10, α_{i-1} is indicated as the angle between axis i-1 and axis i. (The lines with the triple hash marks are parallel)

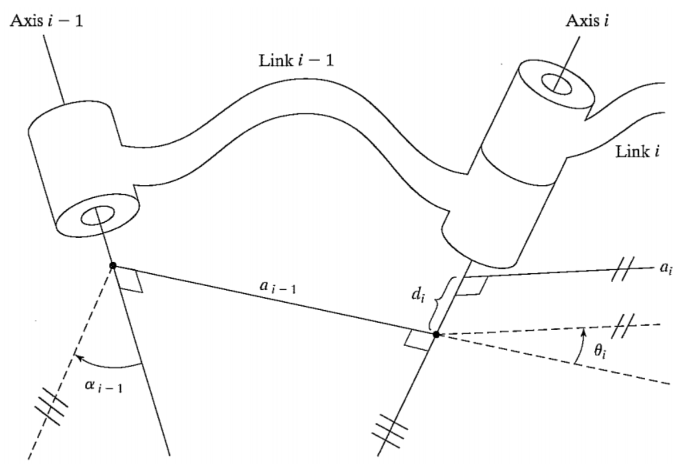


Figure 11: The link offset, d, and the joint angle, θ, are two parameters that are used to describe the connection between neighboring links. [1]

4.1.1 Link-Connection Description

Neighboring links have a common joint axis between them. One parameter of interconnection has to do with the distance along this common axis from one link to the next. This parameter is called the link offset. The offset at joint axis i is called d_i. The second parameter describes the amount of rotation about this common axis between one link and its neighbor. This is called the joint angle, θ_i.

Figure 11 shows the interconnection of link i-1 and link i. The first parameter of interconnection is the link offset, d_i, which is the signed distance measured along the axis of joint i from the point where a_{i-1} intersects the axis to the point where a_i intersects the axis. The offset d_i is indicated in Fig. 11. The link offset is variable if joint i is prismatic. The second parameter of interconnection is the angle made between an extension of a_{i-1} and a_i measured about the axis of joint i. This is indicated in Fig. 11, where the lines with the double hash marks are parallel. This parameter is named $\theta_i$and is variable for a revolute joint.

Link length, and link twist, depend on joint axes i and i + 1. Hence, a_1 through a_{n-1} and α_1 through α_{n-1} are defined as was discussed in this section. At the ends of the chain, it will be our convention to assign zero to these quantities. That is, a_0=a_n=0 and α_0=α_n=0, Link offset, d_i and joint angle, θ_i, are well defined for joints 2 through n-1 according to the conventions discussed in this section. If joint 1 is revolute, the zero position for may be chosen arbitrarily; d_1 = 0.0 will be our convention. Similarly, if joint 1 is prismatic, the zero position of d_1 may be chosen arbitrarily; θ_1= 0.0 will be our convention. Exactly the same statements apply to joint n.

These conventions have been chosen so that, in a case where a quantity could be assigned arbitrarily, a zero value is assigned so that later calculations will be as simple as possible.

Hence, any robot can be described kinematically by giving the values of four quantities for each link. Two describe the link itself, and two describe the link's connection to a neighboring link. In the usual case of a revolute joint, θ_i is called the joint variable, and the other three quantities would be fixed link parameters. For prismatic joints, d_i is the joint variable, and the other three quantities are fixed link parameters. The definition of mechanisms by means of these quantities is a usually called the Denavit—Hartenberg notation.

4.1.2 Convention For Affixing Frames To Links

In order to describe the location of each link relative to its neighbors, we define a frame attached to each link. The link frames are named by number according to the link to which they are attached. That is, frame $\{i\}$ is attached rigidly to link i.

The convention we will use to locate frames on the links is as follows: The $\hat{Z}$-axis of frame $\{i\}$, called $\hat{Z}_i$, is coincident with the joint axis i. The origin of frame i is located where the perpendicular intersects the joint i axis. $\hat{X}_i$ points along a_i in the direction from joint i to joint i + 1.

In the case of a_i = 0, $\hat{X}_i$ is normal to the plane of $\hat{Z}_i$ and $\hat{Z}_{i+1}$. We define α_i as being measured in the right-hand sense about $\hat{X}_i$ and so we see that the freedom of choosing the sign of α_i in this case corresponds to two choices for the direction of $\hat{X}_i$. $\hat{Y}_i$ is formed by the right-hand rule to complete the i^{th} frame. Figure 12 shows the location of frames i-1 and i for a general manipulator.

We attach a frame to the base of the robot, or link 0, called frame 0. This frame does not move; for the problem of arm kinematics, it can be considered the reference frame. We may describe the position of all other link frames in terms of this frame.

Frame 0 is arbitrary, so it always simplifies matters to choose $\hat{Z}_0$ along axis 1 and to locate frame 0 so that it coincides with frame 1 when joint variable 1 is zero. Using this convention, we will always have a_0 = 0, α_0 = 0. Additionally, this ensures that d_1=0 if joint 1 is revolute, or θ_1 = 0 if joint 1 is prismatic.

For joint n revolute, the direction of $\hat{X}_N$ is chosen so that it aligns with $\hat{X}_{N-1}$ when θ_n=0, and the origin of frame N is chosen so that d_n = 0. For joint n prismatic, the direction of $\hat{X}_N$ is chosen so that θ_n= 0, and the origin of frame N is chosen at the intersection of X_{N-1} and joint axis n when d_n=0

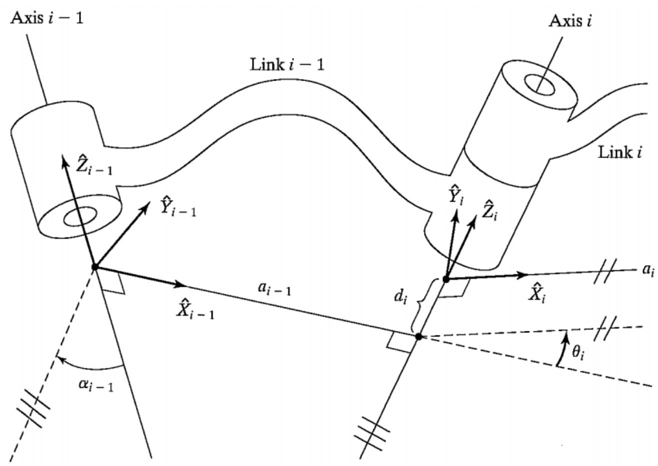


Figure 12: Link frames are attached so that frame i is attached rigidly to link i. [1]

4.1.3 Derivation of Link Transformations in Manipulator Kinematics

In this part, we derive the general form of the transformation that relates the frames attached to neighboring links. We then concatenate these individual transformations to solve for the position and orientation of link n relative to link 0.

We wish to construct the transform that defines frame {i} relative to the frame {i-1}. In general, this transformation will be a function of the four link parameters. For any given robot, this transformation will be a function of only one variable, the other three parameters being fixed by mechanical design. By defining a frame for each link, we have broken the kinematics problem into n subproblems. In order to solve each of these subproblems, namely T_i^{i-1}, we will further break each subproblem into four subsubproblems. Each of these four transformations will be a function of one link parameter oniy and will be simple enough that we can write down its form by inspection. We begin by defining three intermediate frames for each link — {P}, {Q}, and {R}.

Figure 13 shows the same pair of joints as before with frames {P}, {Q} and {R} defined. Note that only the $\hat{X}$ and $\hat{Z}$ axes are shown for each frame, to make the drawing clearer. Frame {R} differs from frame {i-1} only by a rotation of α_{i-1}. Frame {Q} differs from {R} by a translation Frame {P} differs from {Q} by a rotation θ_i and frame {i} differs from P by a translation d_i. If we wish to write the transformation that transforms vectors defined in i to their description in i-1, we may write

$$P^{i-1} = T_R^{i-1} T_Q^R T_P^Q T_i^P P^i \tag{90}$$

or

$$P^{i-1} = T_i^{i-1} P^i \tag{91}$$

where

$$T_i^{i-1} = T_R^{i-1} T_Q^R T_P^Q T_i^P \tag{92}$$

Considering each of these transformations, we see that (92) may be written

$$T_i^{i-1} = R_X(\alpha_{i-1}) D_X(a_{i-1}) R_Z(\theta_i) D_Z(d_i) \tag{93}$$

Multiplying out (93), we obtain the general form of T_i^{i-1}:

$$T_i^{i-1} = \begin{bmatrix} c(\theta_i) & -s(\theta_i) & 0 & a_{i-1} \\ s(\theta_i)c(\alpha i-1) & c(\theta_i)c(\alpha i-1) & -s(\alpha_{i-1}) & -s(\alpha_{i-1})d_i \\ s(\theta_i)s(\alpha i-1) & c(\theta_i)s(\alpha i-1 & c(\alpha_{i-1}) & c(\alpha_{i-1})d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{94}$$

where $c(\theta_i) = cos(\theta_i)$, $s(\theta_i) = sin(\theta_i)$

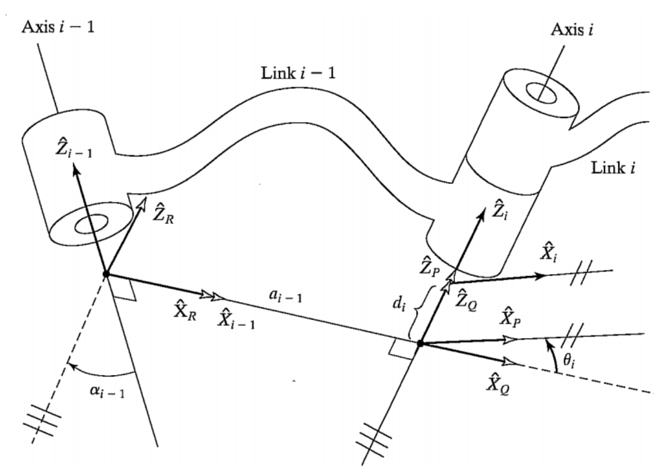


Figure 13: Location of intermediate frames {P}, {Q}, and {R}.[1]

4.1.4 Concatenating Link Transformations

Once the link frames have been defined and the corresponding link parameters found, developing the kinematic equations is straightforward. From the values of the link parameters, the individual link-transformation matrices can be computed.

Then, the link transformations can be multiplied together to find the single transformation that relates frame {N} to frame {0}:

$$T_N^0 = T_1^0 T_2^1 T_3^2 ... T_N^{N-1} \tag{95}$$

This transformation, T_N^0, will be a function of all n joint variables. If the robot's joint-position sensors are queried, the Cartesian position and orientation of the last link can be computed by T_N^0

4.2 Mobile Robot Kinematics

4.2.1 Assumptions of Kinematic Models of Mobile Robots

Here are assumptions we need to follow in the mobile robot kinematic models:

(1)The robot moves in a planar surface (2D Space).

(2)The guidance axis are perpendicular to floor.

(3)Wheels rotate without any slippery problems.

(4)During small amounts of time, which direction is maintained constant, the vehicle will move from one point to other following a circumference arc.

(5)The robot is considered as a solid rigid body, and any movable parts are the direction wheels, which are moved following a commanded control position

4.2.2 Configurations of Mobile Robots

The two sketches shown in figure 14 show the differential and the classical three-wheeled vehicles (e.g. Pioneer Robot). The differential configuration use independent velocities in both wheels left and right (v_L, and v_R, respectively) to move in the 2D plane to a specific point (x,y) and specific orientation ϕ . The three wheeled vehicle uses a single controlled angle and speed wheel to move to a desired position and orientation.

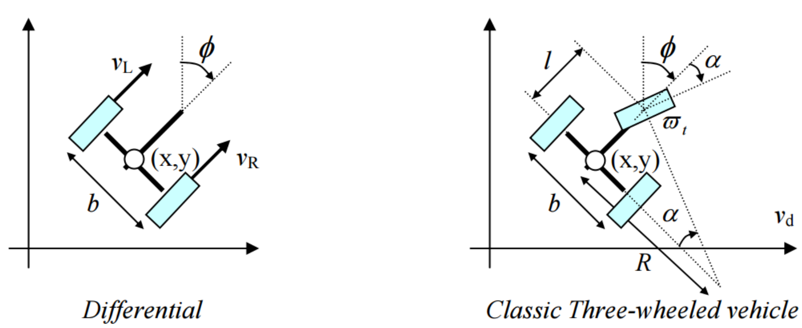


Figure 14: Typical mobile robot kinematic model.

4.2.3 Kinematic Equation of Mobile Robot

Wheel movement (speed) in direction x is calculated by radius r and angle speed rotation θ by (see figure 15):

$$\dot{x} = r\dot{\theta} \tag{96}$$

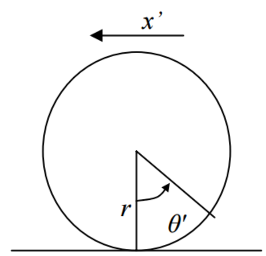


Figure 15: Speed of wheels.

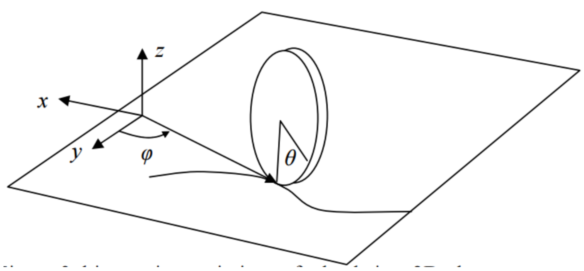


Figure 16: Speed of wheels.

This is, the speed in the x direction is directly proportional by the angular velocity of the wheel. However, other restrictions appear in wheels when the movement is restricted to a 2D plane (x,y). Assume the angular orientation of a wheel is defined by angle ϕ. Then, while the wheel is following a path and having no slippery conditions, the velocity of the wheel at a given time, which is set by $r\dot{\theta}$, has the following restrictive velocity components ($\dot{x}$,$\dot{y}$) with respect to coordinate axes X and Y.

$$r\dot{\theta} = -\dot{x}sin(\phi) + \dot{y}cos(\phi) \tag{97}$$

$$0 = \dot{x}cos(\phi) + \dot{y}sin(\phi) \tag{98}$$

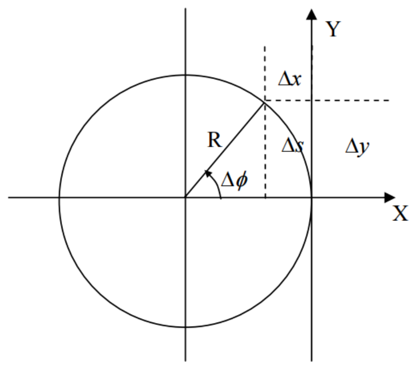


Figure 17: Circumference movement of the vehicle.

Consider now that the mobile robot (or vehicle) follows a circular trajectory as shown in figure 17. Notice that the lineal and angular velocities of the vehicle are given by

$$v = \frac{\Delta s}{\Delta t} \tag{99}$$

and

$$\omega = \frac{\Delta \phi}{\delta t} \tag{100}$$

where Δs and $\Delta \phi$ are the arc distance traveled by the wheel, and its respective orientation with respect to the global coordinates. The arc distance Δs traveled in Δt time is obtained by:

$$\Delta s = R\Delta \phi \tag{101}$$

where R is the circumference radius of the wheel. Moreover, the curvature is defined as the inverse of the radius R. Then the movement equations in the initial position are given by the following two expressions

$$\Delta x = R\left(cos(\Delta \phi) - 1\right) \tag{102}$$

$$\Delta y = Rsin(\Delta \phi) \tag{103}$$

An extension of the later equations is provided in the next expressions, considering an specific initial orientation of angle ϕ. This is accomplished by rotating the earlier initial coordinates (102) and (103)

$$\Delta x = R\left(cos(\Delta \phi) - 1\right) cos\phi - Rsin(\Delta \phi)sin\phi \tag{104}$$

$$\Delta y = R\left(cos(\Delta \phi) - 1\right) sin\phi + Rsin(\Delta \phi)cos\phi \tag{105}$$

Assuming now that the control interval is sufficiently small, then we can assume that the orientation change would be small enough, so that $cos(\Delta\phi) \simeq 1$ and $sin(\Delta\phi) \simeq \Delta\phi$. After substitution and considering equation 101, we get

$$\Delta x = -\Delta s sin\phi \tag{106}$$

$$\Delta y = \Delta s cos\phi \tag{107}$$

Dividing both sides of equations (106) and (107) by Δt , and considering (99), if Δt tends to zero, we finally got:

$$\dot{x} = -v sin\phi \tag{108}$$

$$\dot{y} = v cos\phi \tag{109}$$

$$\dot{\phi} = \omega \tag{110}$$

4.2.4 Differential Drive Robot Kinematics

Assume for differential configuration model, that ω_L and ω_R are the corresponding angular velocities of the left and right wheels. Given the radius of the wheels as r, the corresponding linear and angular velocities of the vehicle are given by

$$v = \frac{v_R + v_L}{2} = \frac{\omega_R + \omega_L}{2} \tag{111}$$

$$\omega = \frac{v_R - v_L}{b} = \frac{\omega_R - \omega_L}{b} r \tag{112}$$

where b is the bias of the vehicle (separation of the two central wheels). Also, if the linear and angular velocities are provided, then the angular velocities of the wheels can be obtained by

$$\omega_L = \frac{v - (b/2)\omega}{r} \tag{113}$$

$$\omega_R = \frac{v + (b/2)\omega}{r} \tag{114}$$

Substituting equations (111) and (112) into the model equation of mobile robots, we got

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} -(r sin\phi)/2 \\ (r cos(\phi))/2 \\ -r/b \end{bmatrix} \omega_L + \begin{bmatrix} -(r sin\phi)/2 \\ (r cos\phi)/2 \\ r/b \end{bmatrix} \omega_R \tag{115}$$

4.2.5 Three-wheeled Robot Kinematics

For this vehicle the control angle for direction is defined by angle α (or by its angular velocity ω_α), and the angula velocity of the wheel itself ω_t (or by its total velocity v_t = rω_t . Assume that the guidance point of the vehicle is in the back part of the control wheel (central back axis). For this configuration, the corresponding model is obtained by

$$v = v_t cos\alpha = r\omega_t cos\alpha \tag{116}$$

and

$$\dot{\alpha} = \omega_\alpha \tag{117}$$

Also, the angular velocity orientation is given by

$$\dot{\phi} = \frac{r\omega_t}{l} sin\alpha = \frac{v_t}{l} sin\alpha \tag{118}$$

After substituting into kinematic model of mobile robot (i.e. 108, 109 and 110), we found the model by

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} -sin\phi cos\alpha \\ cos\phi cos\alpha \\ (sin\alpha)/l \\ 0 \end{bmatrix} v_t + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \omega_\alpha = \begin{bmatrix} -sin\phi cos\alpha & 0 \\ cos\phi cos\alpha & 0 \\ (sin\alpha)/l & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_t \\ \omega_\alpha \end{bmatrix} \tag{119}$$

Notice that once known the desired lineal v and angular velocities ω of the vehicle, the control variables α and ω_t can be obtained by

$$\alpha = arctan(\frac{l}{R}) = arctan(\frac{l\omega}{v}) \tag{120}$$

$$\omega_t = \frac{v_t}{r} = \frac{\sqrt{v^2 + \omega^2 l^2}}{r} \tag{121}$$

4.3 Inverse Manipulator Kinematics

In the previous section we showed how to determine the end-effector position and orientation in terms of the joint variables. This section is concerned with the inverse problem of finding the joint variables in terms of the end-effector position and orientation. This is the problem of inverse kinematics, and it is, in general, more difficult than the forward kinematics problem.

4.3.1 Solvability of Inverse Kinematics Problems

The problem of solving the kinematic equations of a manipulator is a nonlinear one. Given the numerical value of T_N^0, we attempt to find values of θ_1, θ_2, ..., θ_n. For the case of an arm with six degrees of freedom, we have 12 equations and six unknowns. However, among the 9 equations arising from the rotation-matrix portion of T_6^0, only 3 are independent. These, added to the 3 equations from the position-vector portion of T_6^0, give 6 equations with six unknowns. These equations are nonlinear, transcendental equations, which can be quite difficult to solve. Therefore, as with any nonlinear set of equations, we must concern ourselves with the existence of solutions, with multiple solutions, and with the method of solution.

4.3.2 Existence of Solutions

The question of whether any solution exists at all raises the question of the manipulator's work space. Roughly speaking, work space is that volume of space that the end-effector of the manipulator can reach. For a solution to exist, the specified goal point must lie within the work space. Sometimes, it is useful to consider two definitions of workspace: Dexterous work space is that volume of space that the robot end-effector can reach with all orientations. That is, at each point in the dexterous workspace, the end-effector can be arbitrarily oriented. The reachable workspace is that volume of space that the robot can reach in at least one orientation. Clearly, the dexterous workspace is a subset of the reachable workspace.

Consider the workspace of the two-link manipulator in Fig. 18. If l_1 = l_2, then the reachable workspace consists of a disc of radius $2l_1$.The dexterous workspace consists of only a single point, the origin. If $l_1 \neq l_2$, then there is no dexterous workspace, and the reachable workspace becomes a ring of outer radius l_1+l_2 and inner radius $|l_1$-$l_2|$. Inside the reachable workspace there are two possible orientations of the end-effector. On the boundaries of the workspace there is only one possible orientation.

These considerations of workspace for the two-link manipulator have assumed that all the joints can rotate 360 degrees. This is rarely true for actual mechanisms. When joint limits are a subset of the full 360 degrees, then the workspace is obviously correspondingly reduced, either in extent, or in the number of possible orientations attainable. For example, if the arm in Fig. 18 has full 360-degree motion for θ_1, but only $0 \leq \theta_2 \leq 180$ degree, then the reachable workspace has the same extent, but only one orientation is attainable at each point.

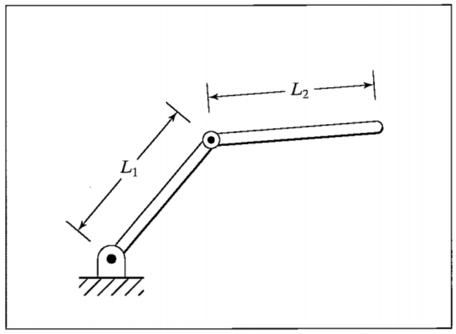


Figure 18: Two-link manipulator with link lengths l_1 and l_2.[1]

When a manipulator has fewer than six degrees of freedom, it cannot attain general goal positions and orientations in 3D space. Clearly, the planar manipulator in Fig. 18 cannot reach out of the plane, so any goal point with a nonzero Z-coordinate value can be quickly rejected as unreachable. In many realistic situations, manipulators with four or five degrees of freedom are employed that operate out of a plane, but that clearly cannot reach general goals. Each such manipulator must be studied to understand its workspace. In general, the workspace of such a robot is a subset of a subspace that can be associated with any particular robot.

4.3.3 Multiple Solutions

Another possible problem encountered in solving kinematic equations is that of multiple solutions. A planar arm with three revolute joints has a large dexterous workspace in the plane (given ”good” link lengths and large joint ranges), because any position in the interior of its workspace can be reached with any orientation. Figure 19 shows a three-link planar arm with its end-effector at a certain position and orientation. The dashed lines indicate a second possible configuration in which the same end-effector position and orientation are achieved.

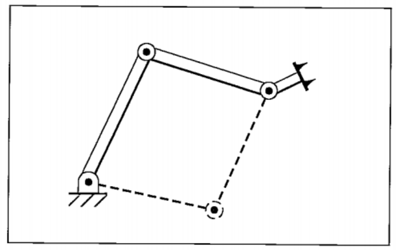


Figure 19: Three-link manipulator. Dashed lines indicate a second solution. [1]

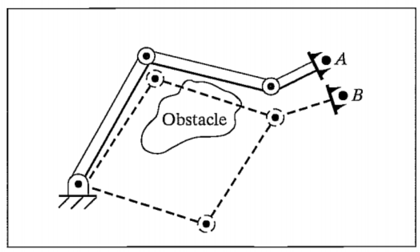


Figure 20: One of the two possible solutions to reach point B causes a collision. [1]

The fact that a manipulator has multiple solutions can cause problems, because the system has to be able to choose one. The criteria upon which to base a decision vary, but a very reasonable choice would be the closest solution. For example, if the manipulator is at point A, as in Fig. 20, and we wish to move it to point B, a good choice would be the solution that minimizes the amount that each joint is required to move. Hence, in the absence of the obstacle, the upper dashed configuration in Fig. 20 would be chosen. This suggests that one input argument to our kinematic inverse procedure might be the present position of the manipulator. In this way, if there is a choice, our algorithm can choose the solution closest in joint-space. However, the notion of ”close” might be defined in several ways. For example, typical robots could have three large links followed by three smaller, orienting links near the end-effector. In this case, weights might be applied in the calculation of which solution is ”closer” so that the selection favors moving smaller joints rather than moving the large joints, when a choice exists. The presence of obstacles might force a ”farther” solution to be chosen in cases where the ”closer” solution would cause a collision — in general, then, we need to be able to calculate all the possible solutions. Thus, in Fig. 20, the presence of the obstacle implies that the lower dashed configuration is to be used to reach point B. In addition, The number of solutions depends upon the number of joints in the manipulator but is also a function of the link parameters (α_i, a_i and d_i for a rotary joint manipulator) and the allowable ranges of motion of the joints.

In general, the more nonzero link parameters there are, the more ways there will be to reach a certain goal. For example, consider a manipulator with six rotational joints. Figure 21 shows how the maximum number of solutions is related to how many of the link length parameters (the a_i) are zero. The more that are nonzero, the bigger is the maximum number of solutions. For a completely general rotary-jointed manipulator with six degrees of freedom, there are up to sixteen solutions possible.

a_i	Number of solutions
$a_1 = a_3 = a_5 = 0$	≤ 4
$a_3 = a_5 = 0$	≤ 8
$a_3 = 0$	≤ 16
All $a_i \neq 0$	≤ 16

Figure 21: Number of solutions vs. nonzero a_1. [1]

4.3.4 Method of Solution

Unlike linear equations, there are no general algorithms that may be employed to solve a set of nonlinear equations. In considering methods of solution, it will be wise to define what constitutes the ”solution” of a given manipulator.

A manipulator will be considered solvable if the joint variables can be determined by an algorithm that allows one to determine all the sets of joint variables associated with a given position and orientation.

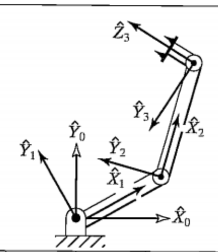
The main point of this definition is that we require, in the case of multiple solutions, that it be possible to calculate all solutions. Hence, we do not consider some numerical iterative procedures as solving the manipulator — namely, those methods not guaranteed to find all the solutions.

We will split all proposed manipulator solution strategies into two broad classes: closed-form solutions and numerical solutions. In this section, We will restrict our attention to closed-form solution methods. In this context, ”closed form” means a solution method based on analytic expressions or on the solution of a polynomial of degree 4 or less, such that noniterative calculations suffice to arrive at a solution. Within the class of closed-form solutions, we distinguish two methods of obtaining the solution: algebraic and geometric.

A major recent result in kinematics is that, according to our definition of solvability, all systems with revolute and prismatic joints having a total of six degrees of freedom in a single series chain are solvable. However, this general solution is a numerical one. Only in special cases can robots with six degrees of freedom be solved analytically. These robots for which an analytic (or closed-form) solution exists are characterized either by having several intersecting joint axes or by having many equal to 0 or ± 90 degrees. Calculating numerical solutions is generally time consuming relative to evaluating analytic expressions; hence, it is considered very important to design a manipulator so that a closed-form solution exists. Manipulator designers discovered this very soon, and now virtually all industrial manipulators are designed sufficiently simply that a closed-form solution can be developed. A sufficient condition that a manipulator with six revolute joints have a closed-form solution is that three neighboring joint axes intersect at a point.

As an introduction to solving kinematic equations, we will consider two different approaches to the solution of a simple planar three-link manipulator.

4.3.5 Algebraic Solution



i	$\alpha_i - 1$	$\alpha_i - 1$	d_i	θ_i
1	0	0	0	θ_1
2	0	L_1	0	θ_2
3	0	L_2	0	θ_3

Figure 22: Three-link planar manipulator and its link parameters. [1]

Consider a simple three-link planar manipulator. It is shown with its link parameters in Fig. 22.

Following the method of forward kinematics section, we can use the link parameters easily to find the kinematic equations of this arm:

$$T_W^B = T_3^0 = \begin{bmatrix} c_{123} & -s_{123} & 0 & l_1c_1 + l_2c_{12} \\ s_{123} & c_{123} & 0 & l_1s_1 + l_2s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{122}$$

where c_{123} = $cos(\theta_1 + \theta_2 + \theta_3)$, s_{123} = $sin(\theta_1 + \theta_2 + \theta_3)$, c_{12} = $cos(\theta_1 + \theta_2)$, s_{12} = $sin(\theta_1 + \theta_2)$, W is wrist frame, B is base frame

To focus our discussion on inverse kinematics, we will assume that the necessary transformations have been performed so that the goal point is a specification of the wrist frame relative to the base frame, that is, T_W^B. Because we are working with a planar manipulator, specification of these goal points can be accomplished most easily by specifying three numbers: x, y, and ϕ where ϕ is the orientation of link 3 in the plane (relative to the +$\hat{X}$ axis). Hence, rather than giving a general T_W^B as a goal specification, we will assume a transformation with the structure

$$T_W^B = \begin{bmatrix} c_\phi & -s_\phi & 0 & x \\ s_\phi & c_\phi & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{123}$$

All attainable goals must lie in the subspace implied by the structure of equation 123. By equating 122 and 123, we arrive at a set of four nonlinear equations that must be solved for θ_1, θ_2 and θ_3:

$$c_\phi = c_{123} \tag{124}$$

$$s_\phi = s_{123} \tag{125}$$

$$x = l_1c_1 + l_2c_{12} \tag{126}$$

$$y = l_1s_1 + l_2s_{12} \tag{127}$$

We now begin our algebraic solution of equations (124) through (127). If we square both (126) and (127) and add them, we obtain

$$x^2 + y^2 = l_1^2 + l_2^2 + 2l_1l_2c_2 \tag{128}$$

where we have made use of

$$c_{12} = c_1c_2 - s_1s_2 \tag{129}$$

$$s_{12} = c_1s_2 + s_1c_2 \tag{130}$$

Solving (128) for c_2, we obtain

$$c_2 = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1 l_2} \tag{131}$$

In order for a solution to exist, the right-hand side of (131) must have a value between —1 and 1. In the solution algorithm, this constraint would be checked at this time to find out whether a solution exists. Physically, if this constraint is not satisfied, then the goal point is too far away for the manipulator to reach.

Assuming the goal is in the workspace, we write an expression for s_2 as

$$s_2 = \pm\sqrt{1 - c_2^2} \tag{132}$$

Finally, we compute θ_2, using the two-argument arctangent routine:

$$\theta_2 = Atan2(s_2, c_2) \tag{133}$$

The choice of signs in (132) corresponds to the multiple solution in which we can choose the ”elbow-up” or the ”elbow-down” solution. In determining θ_2, we have used one of the recurring methods for solving the type of kinematic relationships that often arise, namely, to determine both the sine and cosine of the desired joint angle and then apply the two-argument arctangent. This ensures that we have found all solutions and that the solved angle is in the proper quadrant.

Having found we can solve (126) and (127) for θ_1. We write (126) and (127) in the form

$$x = k_1 c_1 - k_2 s_1 \tag{134}$$

$$y = k_1 s_1 + k_2 c_1 \tag{135}$$

where

$$k_1 = l_1 + l_2 c_2 \tag{136}$$

$$k_2 = l_2 s_2 \tag{137}$$

In order to solve an equation of this form, we perform a change of variables. Actually, we are changing the way in which we write the constants k_1 and k_2. If

$$r = \sqrt{k_1^2 + k_2^2} \tag{138}$$

and

$$\gamma = Atan2(k_2, k_1) \tag{139}$$

then

$$k_1 = rcos\gamma \tag{140}$$

$$k_2 = rsin\gamma \tag{141}$$

Equation (134) and (135) can now be written as

$$\frac{x}{r} = cos\gamma cos\theta_1 - sin\gamma sin\theta_1 \tag{142}$$

$$\frac{y}{r} = cos\gamma sin\theta_1 + sin\gamma cos\theta_1 \tag{143}$$

so

$$cos(\gamma + \theta_1) = \frac{x}{r} \tag{144}$$

$$sin(\gamma + \theta_1) = \frac{y}{r} \tag{145}$$

Using the two-argument arctangent, we get

$$\gamma + \theta_1 = Atan2(\frac{y}{r}, \frac{x}{r}) = Atan2(y, x) \tag{146}$$

and so

$$\theta_1 = Atan2(y, x) - Atan2(k_2, k_1) \tag{147}$$

Note that, when a choice of sign is made in the solution of θ_2 above, it will cause a sign change in k_2, thus affecting θ_1. The substitutions used, (138) and (141), constitute a method of solution of a form appearing frequently in kinematics—namely, that of (126) or (127). Note also that, if x = y = 0, then (147) becomes undefined—in this case, θ_1 is arbitrary. Finally, from (124) and (125), we can solve for the sum of θ_1 through θ_3:

$$\theta_1 + \theta_2 + \theta_3 = Atan2(s_\phi, c_\phi) = \phi \tag{148}$$

From this, we can solve for θ_3, because we know the first two angles. It is typical with manipulators that have two or more links moving in a plane that, in the course of solution, expressions for sums of joint angles arise.

In summary, an algebraic approach to solving kinematic equations is basically one of manipulating the given equations into a form for which a solution is known.

4.3.6 Geometric Solution

In a geometric approach to finding a manipulator's solution, we try to decompose the spatial geometry of the arm into several plane-geometry problems. For many manipulators (particularly when the = 0 or ± 90) this can be done quite easily. Joint angles can then be solved for by using the tools of plane geometry. For the arm with three degrees of freedom shown in Fig. 22, because the arm is planar, we can apply plane geometry directly to find a solution.

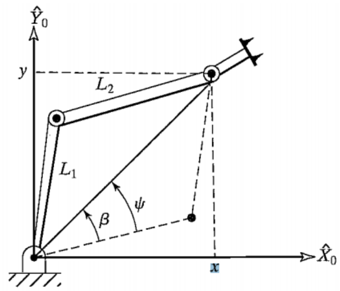


Figure 23: Plane geometry associated with a three-link planar robot. [1]

Figure 23 shows the triangle formed by l_1, l_2, and the line joining the origin of frame 0 with the origin of frame 3. The dashed lines represent the other possible configuration of the triangle, which would lead to the same position of the frame 3. Considering the solid triangle, we can apply the ”law of cosines” to solve for θ_2:

$$x^2 + y^2 = l_1^2 + l_2^2 - 2l_1 l_2 cos(180 + \theta_2) \tag{149}$$

Now; $cos(180 + \theta_2) = -cos(\theta_2)$, so we have

$$c_2 = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1 l_2} \tag{150}$$

In order for this triangle to exist, the distance to the goal point ,$\sqrt{x^2 + y^2}$ must be less than or equal to the sum of the link lengths, l_1 + l_2. This condition would be checked at this point in a computational algorithm to verify existence of solutions. This condition is not satisfied when the goal point is out of reach of the manipulator. Assuming a solution exists, this equation is solved for that value of θ_2 that lies between 0 and —180 degrees, because only for these values does the triangle in Fig. 4.8 exist. The other possible solution (the one indicated by the dashed-line triangle) is found by symmetry to be θ_2' = -θ_2.

To solve θ_1, we find expressions for angles ψ and β as indicated in Fig. 23. First, β may be in any quadrant, depending on the signs of x and y. So we must use a two-argument arctangent:

$$\beta = Atan2(y, x) \tag{151}$$

We again apply the law of cosines to find ψ:

$$cos\psi = \frac{x^2 + y^2 + l_1^2 - l_2^2}{2l_1\sqrt{x^2 + y^2}} \tag{152}$$

Here, the arccosine must be solved so that $0 \leq \psi \leq 180$ degree, in order that the geometry which leads to (152) will be preserved. These considerations are typical when using a geometric approach—we must apply the formulas we derive only over a range of variables such that the geometry is preserved. Then we have

$$\theta_1 = \beta \pm \psi \tag{153}$$

where the plus sign is used if $\theta_2 < 0$ and the minus sign if $\theta_2 > 0$ We know that angles in a plane add, so the sum of the three joint angles must be the orientation of the last link:

$$\theta_1 + \theta_2 + \theta_3 = \phi \tag{154}$$

This equation is solved for θ_3 to complete our solution.

4.3.7 Piepper's Solution When Three Axes Intersect

Although a completely general robot with six degrees of freedom does not have a closed-form solution, certain important special cases can be solved. Pieper studied manipulators with six degrees of freedom in which three consecutive axes intersect at a point. In this section, we outline the method he developed for the case of all six joints revolute, with the last three axes intersecting.Pieper's work applies to the majority of commercially available industrial robots.

When the last three axes intersect, the origins of link frames 4, 5, and 6 are all located at this point of intersection. This point is given in base coordinates as

$$P^0_{4ORG} = T^0_1 T^1_2 T^2_3 P^3_{4ORG} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{155}$$

(where ORG means the origin of the coordinate)or

$$P^0_{4ORG} = T^0_1 T^1_2 T^2_3 \begin{bmatrix} a_3 \\ -d_4 s\alpha_3 \\ d_4 c\alpha_3 \\ 1 \end{bmatrix} \tag{156}$$

or

$$P^0_{4ORG} = T^0_1 T^1_2 \begin{bmatrix} f_1(\theta_3) \\ f_2(\theta_3) \\ f_3(\theta_3) \\ 1 \end{bmatrix} \tag{157}$$

where

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ 1 \end{bmatrix} = T_3^2 \begin{bmatrix} a_3 \\ -d_4 s\alpha_3 \\ d_4 c\alpha_3 \\ 1 \end{bmatrix} \tag{158}$$

Using (94) for in (158) yields the following expressions for f_1:

$$f_1 = a_3 c_3 + d_4 s\alpha_3 s_3 + a_2 \tag{159}$$

$$f_2 = a_3 c\alpha_2 s_3 - d_4 s\alpha_3 c\alpha_2 c_3 - d_4 s\alpha_2 c\alpha_3 - d_3 s\alpha_2 \tag{160}$$

$$f_3 = a_3 s\alpha_2 s_3 - d_4 s\alpha_3 s\alpha_2 c_3 + d_4 c\alpha_2 c\alpha_3 + d_3 c\alpha_2 \tag{161}$$

Using (94) for T_1^0 and T_2^1 in (157), we obtain

$$P_{4ORG}^0 = \begin{bmatrix} c_1 g_1 - s_1 g_2 \\ s_1 g_1 + c_1 g_2 \\ g_3 \\ 1 \end{bmatrix} \tag{162}$$

where

$$\begin{aligned} g_1 &= c_2 f_1 = s_2 f_2 + a_1 \\ g_2 &= s_2 c\alpha_1 f_1 + c_2 c\alpha_1 f_2 - s\alpha_1 f_3 - d_2 s\alpha_1 \\ g_3 &= s_2 s\alpha_1 f_1 + c_2 s\alpha_1 f_2 + c\alpha_1 f_3 + d_2 c\alpha_1 \end{aligned} \tag{163}$$

We now write an expression for the squared magnitude of which we will denote as r = x^2 + y^2 + z^2, and which is seen from (162) to be

$$r = g_1^2 + g_2^2 + g_3^2 \tag{164}$$

so, using (163) for g_i the we have

$$r = f_1^2 + f_2^2 + f_3^2 + a_1^2 + d_2^2 + 2d_2 f_3 + 2a_1(c_2 f_1 - s_2 f_2) \tag{165}$$

We now write this equation, along with the Z-component equation from (162), as a system of two equations in the form

$$\begin{aligned} r &= (k_1 c_2 + k_2 s_2)2a_1 + k_3 \\ z &= (k_1 s_2 - k_2 c_2)s\alpha_1 + k_4 \end{aligned} \tag{166}$$

where

$$\begin{aligned} k_1 &= f_1 \\ k_2 &= -f_2 \\ k_3 &= f_1^2 + f_2^2 + f_3^2 + a_1^2 + d_2^2 + 2d_2 f_3 \\ k_4 &= f_3 c\alpha_1 + d_2 c\alpha_1 \end{aligned} \tag{167}$$

Equation (166) is useful because dependence on θ_1 has been eliminated and because dependence on θ_2 takes a simple form.

Now let us consider the solution of (166) for θ_3. We distinguish three cases:

1. If a_1 = 0, then we have r = k_3, where r is known. The right-hand side (k_3) is a function of θ_3 only. After the substitution, a quadratic equation in tan($\frac{\theta_3}{2}$) may be solved for θ_3.

2. If $s\alpha_1$ = 0, then we have z = k_4, where z is known. Again, after substitution, a quadratic equation arises that can be solved for θ_3.

3. Otherwise, eliminate s_2 and c_2 from (166) to obtain

$$\frac{(r-k_3)^2}{4a_1^2} + \frac{(z-k_4)^2}{s^2\alpha_1} = k_1^2 + k_2^2 \tag{168}$$

This equation, after substitution for θ_3, results in an equation of degree 4, which can be solved for θ_3.

Having solved for we can solve (166) for θ_2 and (162) for θ_1.

To complete our solution, we need to solve for θ_4, θ_5 and θ_6, and These axes intersect, so these joint angles affect the orientation of only the last link. We can compute them from nothing more than the rotation portion of the specified goal, R_6^0 Having obtained θ_1, θ_2 and θ_3, we can compute $R_4^0|_{\theta_4=0}$ by which notation we mean the orientation of link frame $\{4\}$ relative to the base frame when θ_4 = 0. The desired orientation of $\{6\}$ differs from this orientation only by the action of the last three joints. Because the problem was specified as given R_6^0,we can compute

$$R_6^4|_{\theta_4=0} = (R_4^0)^{-1}|_{\theta_4=0} R_6^0 \tag{169}$$

For many manipulators, these last three angles can be solved for by using exactly the Z—Y—Z Euler angle solution, applied to $R_6^4|_{\theta_4}$ = 0 For any manipulator (with intersecting axes 4, 5, and 6), the last three joint angles can be solved for as a set of appropriately defined Euler angles. There are always two solutions for these last three joints, so the total number of solutions for the manipulator will be twice the number found for the first three joints.

References

[1] J. J. Craig. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 1989. ISBN 0201095289.

[2] R. M. Murray, S. S. Sastry, and L. Zexiang. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 1994. ISBN 0849379814.

[3] M. W. Spong, S. Hutchinson, and M. Vidyasagar. *Robot modeling and control*. John Wiley & Sons, Hoboken (N.J.), 2006. ISBN 0-471-64990-2.