Chapter 4

EFFECTOR MODEL AND MOTION PRIMITIVES

Dr. Mayez Al-Mouhamed
Professor, Computer Engineeing Department
King Fahd University of petroleum and Minerals

In Chapter 2, the robot hand position and orientation has been geometrically represented by using the position vector $O_oO_{n,o}$ and the orientation matrix M_o^n . The matrix M_o^n is not always the most convenient representation for the robot hand orientation. In a robot programming environment, the use of the orientation matrix as a method to specify the robot hand orientation is faced to two problems. The orientation matrix includes nine redundant parameters which are difficult to manage when this matrix is the only way to manipulate the robot hand orientation.

This chapter discusses the problem of geometrically representing the robot hand effector. It is suggested to use the Euler angles when explicit specification of the hand orientation is required in robot programming. This allows representing the robot hand by means of three cartesian coordinates and three euler angles. A non redundant set of six parameters will then be available for representing the robot effector. Methods for establishing the correspondence between the orientation matrix and the set of euler angles will be discussed.

In most cases, the robot hand is to be attached to the tool whose motion becomes the objective of moving the robot arm. To achieve motion coordination with respect to the tool, frame of reference is defined using the knowledge on its position and orientation with respect to the robot hand frame. To move the tool in a coordinated way, one needs to find the implied motion of the effector frame and convert this information into the robot configuration space in order to achieve the requirements on the tool frame. Since, motion coordination is conditioned by the resulting motion of the tool. One important question is to find relation between the motion of the robot effector and that of the tool. A logical description of the task would deal with the tool description and motion rather than with that of the robot arm itself. Implication on the motion of the robot arm should be undertaken, by the system, from that of the tool. One important question is the study of geometric transformations that allow the tool frame be translated and rotated by specific amounts and direction. This leads to study the incremental translation and rotation of the tool frame of reference.

The incremental translation and rotation is very useful for programming tasks that use sensory feedback in a closed loop form. Generally, these tasks consists of adjusting the position and orientation of the tool frame by converting information issued from the robot environment into incremental corrections. For example, the operation of inserting a peg into a hole requires the use of a force sensor to sense the friction forces between the peg and the hole. To achieve the insertion, one needs to convert the force information into incremental corrections with respect to the peg position and orientation. To obtain good sensitivity, sensors are placed close to the robot hand effector. The knowledge of the tool position and orientation with respect to the effector frame allow converting these forces into force with respect to the tool frame. Therefore, corrections on position and orientation of the tool frame require the evaluation of their effect on the effector frame. Mathematical transformations will be found for each case. A set of motion primitives is presented as flexible tools for robot programming environment that belong to the effector and object description levels. These transformations will be presented in this Chapter.

4.1 Basic effector representation

The simplest way to represent the robot arm is to use the articular variables $(\theta_1, ..., \theta_n)$ as a non-redundant set of geometric parameters. To each position and orientation of the the robot hand, we can associate one configuration of the the articular variables. The transformation is then simple, fast, and straightforward. However, At least two drawbacks can be observed with this representation:

1. The trajectories or task specification that are described in this manner cannot support any modification because of the absolute representation.

2. The articular variables are very difficult to be manipulated in a user defined program. In Chapter 2, we have studied the geometric method which allows representing the robot hand by means of a position vector $O_0O_n, 0$ and an orientation matrix M_0^n :

$$
\begin{aligned}
O_0O_{n,0} &= (X\ Y\ Z)^t \\
M_0^n &= [X_{n,0}\ Y_{n,0}\ Z_{n,0} =] = \begin{bmatrix} X_x & X_y & X_z \\ X_y & Y_y & Z_y \\ X_z & Y_z & Z_z \end{bmatrix}
\end{aligned}
\tag{4.1}
$$

In this representation, the desired position and orientation of the effector is specified by means of twelve redundant parameters. The manipulation of the orientation matrix M_o^6 and its elements is difficult to handle in a robot programming context. Even when a non-redundant set of parameters is selected within matrix M_0^n, the interpretation and meaning of these parameters still difficult to understand and to use in an effector level programming environment. With an effector level programming environment, the task is specified by means of a set motions with respect to the robot hand frame of reference. Example of such effector level programming is the VAL programming environment. Since, the orientation matrix is not convenient to use as a method for user-defining orientation. In a higher level of task programming, the orientation matrix can be seen as a machine internal representation that can be manipulated together with other coherent specifications. For example, in object level programming environment the task is defined by means of a set of transformation on the manipulated object. Naturally, objects are associated frame of references including orientation matrices. In this sense, the robot hand orientation matrix can be recovered from two specifications:

1. The interface between the robot hand frame of reference and the manipulated object.

2. The transformations on the manipulated object can be easily converted, by the computer, into transformations on the robot hand frame. This problem will be discussed later in this Chapter in the context of motion coordination of the tool frame of reference.

Note that finding the coordinate of the origin and the orientation matrix is the prerequisite to finding the robot arm configuration which is the only way to control the robot arm.

In the next two Section we study the use of three independent angles to represent the effector orientation. This methods are useful for finding straightforward mapping between the user space and object space including the robot, the tool, and manipulated objects.

4.2 The euler angles

The orientation of a body in the three dimensional space can be represented by means of three independent angles φ_1 (phi), φ_2, and φ_3 . These angles allows defining the transfer matrix between the fixed frame R_0 and the robot hand frame of reference R_n . To deal with orientation only, we report the frame R_0 to the origin O_n of the hand frame R_n .

Initially we assume both frames R_0 and R_n coincide with respect to their origin and their orthonormal axes are parallel to each other as shown in Figure 1. The rotation angle φ_1 defines the rotation of frame R_n about axis Y_0.

Figure 3.2 shows the corresponding rotation matrix which is $ROTY(\varphi_1)$. Angle φ_2 defines the rotation of frame R_n about axis X_0 . Note here that axis X_n has been affected by the first rotation operator which is defined by $ROTY(\varphi_1)$ as shown in Figure 3.3. Therefore, the current orientation matrix between frames R_n ' and ' R_0 becomes the product of rotation matrices $ROTY(\varphi_1)ROTX(\varphi_2)$. Finally, angle φ_3 defines the rotation of frame R_n about axis Z_n, which has been affected by the previous two rotation operators $ROTY(\varphi_1)$ and $ROTX(\varphi_2)$. Figure 3.4 shows the effect of the three rotation angles on frame R_n .

The resulting transfer matrix between frame R_n and frame R_0 is then function of the three rotations previously defined:

$$ROTY(\varphi_1).ROTX(\varphi_2).ROTZ(\varphi_3) \tag{4.2}$$

Using the three rotation angles φ_1, φ_2, and φ_3, the transfer matrix between frames R_n and R_0 is then identical to the orientation matrix M_o^n, we have:

$$M_o^n = ROTY(\varphi_1).ROTX(\varphi_2).ROTZ(\varphi_3) \tag{4.3}$$

Using the definition of the three rotations, we obtain:

$$\begin{aligned} M_o^n &= \begin{bmatrix} C1 & 0 & -S1 \\ 0 & 1 & 0 \\ -S1 & 0 & C1 \end{bmatrix} . \begin{bmatrix} 1 & 0 & 0 \\ 0 & C2 & S2 \\ 0 & -S2 & C2 \end{bmatrix} . \begin{bmatrix} C3 & S3 & 0 \\ -S3 & C3 & 0 \\ 0 & -0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} C1C3 + S1S2S3 & -C1S3 + S1S2C3 & S1C2 \\ CS.S3 & C2C3 & -S2 \\ -S1C3 + C1S2S3 & S1S3 + C1S2C3 & C1C2 \end{bmatrix} \end{aligned} \tag{4.4}$$

Using the euler angles, the orientation matrix of the robot hand frame can be easily obtained by evaluating the product of three rotation matrices. This result in finding the matrix M_0^n. The position and orientation of the robot hand will then be fully represented by means of three cartesian coordinates, i.e. those of the origin O_n, and the three euler angles which give the orientation of frame R_n with respect to frame R_0.

4.2.1 Finding the euler angles

The inverse problem consists of finding the three euler angles that transform a frame, which is initially identical to R_0, to a frame whose orientation matrix is identical to a given matrix M_0^n . The problem is to find the angles φ_1, φ_2, and φ_3 that satisfy the following equation:

$$\begin{aligned} M_0^n &= \begin{bmatrix} X_x & X_y & X_z \\ X_y & Y_y & Z_y \\ X_z & Y_z & Z_z \end{bmatrix} \\ &= \begin{bmatrix} C1C3 + S1S2S3 & -C1S3 + S1S2C3 & S1C2 \\ C2.S3 & C2C3 & -S2 \\ -S1C3 + C1S2S3 & S1S3 + C1S2C3 & C1C2 \end{bmatrix} \end{aligned} \tag{4.5}$$

Angle φ_2 is characterized by the following equations:

$$\begin{aligned} S2 &= -Z_y \\ C2 &= \pm\sqrt{X_y^2 + Y_y^2} = \pm\sqrt{Z_x^2 + Z_z^2} \end{aligned} \tag{4.6}$$

Obviously, the cosine of angle φ_2 cannot be found in a unique fashion. Two solutions φ_2^+ and φ_2^- are expected. The solution φ_2^+ is defined as follows:

$$\varphi_2^+ = tan^{-1}\left(S2, C2^+\right) \tag{4.7}$$

where $C2^+ = +\sqrt{X_y^2 + Y_y^2}$. As φ_2^+ and φ_2^- have identical sine and opposite cosine, therefore we have:

$$\varphi_2^- = Sign\left(\varphi_2^+\right).\Pi - \varphi_2^+ \tag{4.8}$$

Now consider the following equations which reference the sine and cosine of angle φ_1:

$$Z_x = S1.C2 \text{ and } Z_z = C1.C2$$

Since, the terms $S1$ and $C1$ can only be evaluated when $C2 \neq 0$. In this case, we can evaluate the sine and cosine of φ_1 as functions of the previously determined value $C2^+$:

$$S1^+ = \frac{Z_x}{\sqrt{X_y^2 + Y_y^2}} \text{ and } C1^+ = \frac{Z_z}{\sqrt{X_y^2 + Y_y^2}} \tag{4.9}$$

This gives the solution:

$$\varphi_1^+ = tan^{-1}\left(S1^+, C1^+\right) \tag{4.10}$$

The other solution φ_1^- has opposite sine and opposite cosine when compared with φ_1^+, therefore φ_1^- always differs from φ_1^+ by an angle equal to Pi . In the interval $[-\Pi, +\Pi]$ angle φ_1^- can then be obtained as follows:

$$\varphi_1^- = \varphi_1^+ - Sign\left(\varphi_1^+\right).\Pi \tag{4.11}$$

In a similar manner we consider the following equations which reference the sine and cosine of angle φ_3:

$$X_y = C2.S3 \text{ and } Y_y = C2.C3$$

Since, the terms $S3$ and $C3$ can only be evaluated when $C2 \neq 0$. In this case, we can evaluate the sine and cosine of φ_3 as functions of $C2^+$:

$$S3^+ = \frac{Z_x}{\sqrt{X_y^2 + Y_y^2}} \text{ and } C3^+ = \frac{Z_z}{\sqrt{X_y^2 + Y_y^2}} \tag{4.12}$$

This gives the solution:

$$\varphi_3^+ = tan^{-1}\left(S3^+, C3^+\right) \tag{4.13}$$

The other solution φ_3^- has opposite sine and opposite cosine when compared with φ_3^+, therefore φ_3^- always differs from φ_3^+ by an angle equal to Π . In the interval $[-\Pi, +\Pi]$ angle φ_3^- can then be obtained as follows:

$$\varphi_3^- = \varphi_3^+ - Sign\left(\varphi_3^+\right).\Pi \tag{4.14}$$

As result of the previous discussion, two sets of solutions are then obtained:

$$\left\{\varphi_1^+, \varphi_2^+, \varphi_3^+\right\} \text{ and } \left\{\varphi_1^-, \varphi_2^-, \varphi_3^-\right\} \tag{4.15}$$

One may observe that the sum of φ_1^+ and φ_1^- is always equal to Π . Since when either φ_1^+ or φ_1^- is close to $\Pi/2$, the two solutions become closer to each another and comparison with the previous value of φ_2 cannot give a sharp decision.

On the other hand, the solutions $(\varphi_1^+, \varphi_1^-)$ and $(\varphi_3^+, \varphi_3^-)$ always differ by an angle Π, respectively. Therefore, we can select the solution φ_1^+ or φ_1^- that is closer to the previous value of φ_1 . This allows finding the current sign of term $C2$ which result in finding one set of solutions out of two. When $C2 = 0$, no solution can be found by using the previous system equations. In this case the orientation matrix M_0^n becomes:

$$M_0^n = \begin{bmatrix} C1C3 \pm S1S3 & -C1S3 \pm S1C3 & 0 \\ 0 & 0 & -+1 \\ -S1C3 \pm C1S3 & S1S3 \pm C1C3 & 0 \end{bmatrix} \quad (4.16)$$

Using trigonometric transformation we obtain:

$$M_0^n = \begin{bmatrix} C\,(\varphi_1 \pm \varphi_3) & S\,(\varphi_1 \pm \varphi_3) & 0 \\ 0 & 0 & \pm 1 \\ -S\,(\varphi_1 \pm \varphi_3) & -C\,(\varphi_1 \pm \varphi_3) & 0 \end{bmatrix} \quad (4.17)$$

This shows clearly that only the sum $\varphi_1 + \varphi_3$ or the difference $\varphi_1 - \varphi_3$ can be found depending on whether S2 is equal to $+1$ or -1 . This result can be explained using geometrical criteria: when $C2$ is nil, the angle $\varphi_2 = +-\Pi$ which means that the rotation axes, that defines angles φ_1 and φ_3 becomes co-linear. Therefore the resulting orientation of frame R_n is no more helpful in finding separate solution for φ_1 and φ_3 . The resulting orientation gives only either the sum or the difference of the angles φ_1 and φ_3 . To find separate solutions, one needs to find a solution for either φ_1 or φ_3 by using a heuristic approach and then use of the sum or the difference in order to find a solution for the other angle. Assume the solution φ_1^* is found by extrapolating a time polynomial function $\varphi_1(t)$. A solution φ_3^* can then be found by using either the difference $\varphi_1 - \varphi_3$ or the sum $\varphi_1 + \varphi_3$ depending on whether $S2$ is equal to $+1$ or -1, respectively. In either cases, these values can be found as follows:

$$(\varphi_1 - \varphi_3) = tan^{-1}\left(S(\varphi_1 - \varphi_3), C(\varphi_1 - \varphi_3)\right) \quad (4.18)$$

or

$$(\varphi_1 - \varphi_3) = tan^{-1}\left(S(\varphi_1 + \varphi_3), C(\varphi_1 + \varphi_3)\right) \quad (4.19)$$

The angle φ_3^* can then be found by using either formulas:

$$\varphi_3^* = \varphi_1^* - (\varphi_1 - \varphi_3) \text{ or } \varphi_3^* = (\varphi_1 + \varphi_3) - \varphi_1^* \quad (4.20)$$

The obtained solutions satisfy the trajectory continuity criteria which has been discussed in Chapter 2. This criteria assumes that the temporal evolution of these angles are maintained throughout the motion. Naturally, independent solution for the three angles can only be found when the rotation axes of both angles φ_1 and φ_2 are not parallel.

4.2.2 Final mapping of the euler angles

The use of the three euler angles offers a non-redundant set of angles for referencing the robot hand orientation. Referencing the robot effector can be achieved by using six independent parameters:
Position : $(X_n\ Y_n\ Z_n)$
Orientation : $(\varphi_1\ \varphi_2\ \varphi_3)$

To evaluate the orientation matrix M_o^n, a straightforward transformation is then required:

$$(\varphi_1\ \varphi_2\ \varphi_3) \rightarrow M_o^n$$

Given the matrix M_o^n, the evaluation of φ_1, φ_2 and φ_3 is subject to two solutions:

$$M_o^n \rightarrow (\varphi_1, \varphi_2, \varphi_3)_{1,2} \tag{4.21}$$

The selection of one solution can be achieved by taking the solution which is the closest to the previous values of φ_1, φ_2, and φ_3 .

4.3 Rotations with respect to absolute axes

The orientation of a body in the three dimensional space can also be represented by means of three independent angles φ_1, φ_2 and φ_3 which represent rotations with respect to the fixed axes X_0, Y_0, and Z_0 . The set of rotation angles φ_1, φ_2, and φ_3 are called roll, pitch, and yaw, respectively. These angles allows defining the transfer matrix between frame R_0 and the robot hand frame of reference R_n . In a similar manner, frame R_0 is reported to the origin O_n of the hand frame R_n .

Initially we assume both frames R_0 and R_n coincide and their orthonormal axes are parallel to each another. The rotation angle φ_1 defines the rotation of frame R_n about axis Y_0 . The corresponding rotation matrix is then $ROTY(\varphi_1)$. Angle φ_2 defines the rotation of frame R_n about axis X_0 . Note here that axis X_0 is not affected by the first rotation operator defined by $ROTY(\varphi_1)$. therefore, the current orientation matrix between frames R_n and R_0 becomes the product of rotation matrices $ROTX(\varphi_2)ROTY(\varphi_1)$. Finally, angle φ_3 defines the rotation of frame R_n about axis Z_0 . which is not affected by the previous two rotation operators $ROTY(\varphi_1)$ and $ROTX(\varphi_2)$. Figure shows the effect of the three rotation angles on frame R_n .

The resulting transfer matrix between frame R_n and frame R_0 is then function of the three rotations previously defined:

$$ROTZ(\varphi_3).ROTX(\varphi_2).ROTY(\varphi_1) \tag{4.22}$$

The order of these rotation operators is the opposite of the euler angle set of rotations. Using the three rotation angles φ_1, φ_2, and φ_3, the transfer matrix between frames R_n and R_0 is then identical to the orientation matrix M_o^n, we have:

$$M_o^n = ROTZ(\varphi_3).ROTX(\varphi_2).ROTY(\varphi_1) \tag{4.23}$$

By definition of the three rotations, we obtain:

$$\begin{aligned} M_o^n &= \begin{bmatrix} C3 & -S3 & 0 \\ S3 & C3 & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} 1 & 0 & 0 \\ 0 & C2 & S2 \\ 0 & -S2 & C2 \end{bmatrix} . \begin{bmatrix} C1 & 0 & S1 \\ 0 & 1 & C1 \\ -S1 & 0 & C1 \end{bmatrix} \\ &= \begin{bmatrix} C1C2 & C1S2S3 - S1C3 & C1S2C3 + S1S3 \\ S1C2 & S1S2S3 + C1C3 & S1S2C3 - C1S3 \\ -S2 & C2S3 & C2C3 \end{bmatrix} \end{aligned} \tag{4.24}$$

In a similar manner, this set of rotation matrices can be used for representing the orientation of the robot hand frame. One characteristic of these angles is that their rotation axes are fixed and are not affected by the rotation itself. This explain why these angles are sometimes preferred over the euler angles particularly in aeronautical applications. In robotics applications, the euler angles have a significant advantage the set $(\varphi_3, \varphi_2, \varphi_1)$ because the euler angles are successive transformation on the moving frame while the present angles are with respect to fixed axes.

Given the structure of the matrix product, the inverse problem is similar to that of the euler angles as discussed in Section 3.4. The same problems as those encountered in finding the euler angles are also present for the angles $(\varphi_3, \varphi_2, \varphi_1)$. Both systems present the same complexity with respect to their solutions

4.4 Motion coordination of the tool

Manipulating objects by means of a robot arm consists of attaching the robot hand to that object and moving the arm such that to obtain the desired motion of the manipulated object. Therefore, the operation objective is to move the object itself by means of the arm. The problem is find the relationships between the desired trajectory of the object and that of the robot arm. The robot arm is controlled with respect to its set of articular variables which are related to the robot hand by means of the geometric model. Therefore, assigning trajectory to the tool frame consists of finding the corresponding trajectory in the articular space bypassing the geometric model. This explain why we need to find the geometric relationships between the tool frame of reference and the robot hand frame. Note that the robot hand frame is also called the effector frame of reference.

In this Section we study the geometric relations between a tool frame of reference and that of the robot hand. For this we assume a rigid interface between the tool and the robot hand. In other words, the geometric relationship between robot hand and tool is maintained throughout the motion.

Initially, we assume the tool frame of reference R_t is parallel to the effector frame R_n . The transfer matrix between frames R_t and R_n is the identity matrix because both frame have identical orientation as shown in Figure 3.5.

Let us assume the tool frame of reference is defined by means of a translation vector $O_nO_{t,n}$ and a transfer matrix φ, both are defined with respect to the effector frame R_n . Vector $O_nO_{t,n}$ denotes the translation of the origins between the effector frame and the tool frame. Matrix φ denotes the orientation matrix of the tool frame with respect to the effector frame. Figure 3.6 shows the tool frame with respect to the effector frame.

A point A, which is observed in frame R_t, can be associated a vector $O_tA_{R_t}$. Point A may also be observed with respect to frame R_t . The following transformation should then be used:

$$O_nA_{R_n} = O_nO_{t,n} + \Phi.O_tA_{R_t} \tag{4.25}$$

where vector $O_nA_{R_n}$ is associated to point A which is now is now observed with respect to frame R_n . Note that both vector $O_nO_{t,n}$ and matrix Φ are defined with respect to the effector frame of reference R_n .

The interface between frames R_n and frame R_t is assumed to be rigid. This means that when the tool frame is moved, no change may occur in the position and orientation of frame R_t with respect to the effector frame R_n . The vector $O_0O_{t,0}$ defines the tool frame origin which is observed in the fixed frame R_0 . This vector can always be expressed as function of the tool frame origin and orientation:

$$O_0O_{t,0} = O_0O_n, 0 + M_0^n.\Phi.O_nO_t, n \tag{4.26}$$

1111

4.5 Incremental translation of tool frame

In the following we study the effect on frame R_n of moving the tool frame by either a translation or a rotation.

Assume we want to translate the tool frame by an increment vector L, which is referenced with respect to frame R_t . Initially, the orientation matrix of frame R_n is M_0^n and that of frame R_t is M_0^n Φ, respectively. The general position and orientation of the tool frame is shown in Figure 3.7 with respect to the effector frame. As the required motion is a pure translation, then the orientation of both frames R_n and R_t will not be affected after completion of the translation. As a result of the translation, the tool frame origin will be translated by the increment vector $\Delta O_0 O_{t,0}$, which is given by:

$$\Delta O_0 O_{t,0} = M_0^n.\Phi.L$$

Both frames R_n and R_t are rigidly attached, then frame R_n will be translated by the same increment vector as that of frame R_t . Therefore the increment vector of frame R_n is defined by:

$$\Delta O_0 O_{n,0} = M_0^n.\Phi.L$$

This means the following: to translate the tool frame by an increment vector L, which is defined with respect to frame R_t, the effector frame origin should be translated by the increment vector $M_0^n.\Phi.L$. We also note that the final orientation matrices, of the effector and tool frames, are not affected by the translation operator and are equal to M_0^n and $M_0^n.\Phi$, respectively.

The knowledge of the increment vector $\Delta O_0 O_{n,0}$ and the orientation matrix M_0^n allow finding the arm solution in order to set the effector position and orientation as previously discussed. following the translation of the tool frame only the effect on the effector frame is useful because the setting of the arm requires the knowledge of the position vector and the orientation matrix of the effector frame R_n .

In the following we discuss the design of a primitive motion function using the general relationship between the tool frame and the effector frame.

4.6 Example of motion primitive design

Consider an incremental translation L of frame R_t which is defined along the X_t axis of frame R_t:

$$L = (\Delta X \ 0 \ 0)^t$$

The desired translation allows moving frame R_t by an amount ΔX along axis X_t of frame R_t .

Initially we assume the tool frame of reference is defined by means of a vector of translation origin $O_n O_{t,n}$ and an orientation matrix Φ . Since the matrix Φ defines the orientation matrix of frame R_t with respect to frame R_n, therefore we have:

$$\Phi = [X_{t,n} \ Y_{t,n} \ Z_{t,n}] \tag{4.27}$$

where $X_{t,n}$ $Y_{t,n}$ and $Z_{t,n}$ are the orthonormal vectors of frame R_t expressed with respect to frame R_n . with respect to the effector frame origin, the incremental translation moves the origin O_n by a translation vector:

$$\Delta O_0 O_{n,0} = M_0^n.\Phi.L$$

Let us evaluate the orientation matrix of the tool frame:

$$\begin{aligned} M_0^t.\Phi &= \begin{bmatrix} X_x & Y_x & Z_x \\ X_y & Y_y & Z_y \\ X_z & Y_z & Z_z \end{bmatrix} . \begin{bmatrix} X_{x_{t,n}} & Y_{x_{t,n}} & Z_{x_{t,n}} \\ X_{y_{t,n}} & Y_{y_{t,n}} & Z_{y_{t,n}} \\ X_{z_{t,n}} & Y_{z_{t,n}} & Z_{z_{t,n}} \end{bmatrix} \\ &= [X_{t,0} \ Y_{t,0} Z_{t,0}] \end{aligned} \tag{4.28}$$

where $X_{t,0}$ $Y_{t,0}$ and $Z_{t,0}$ are the orthonormal vectors of frame R_t that are expressed in frame R_0. This is obtained because the product $M_0^t.\Phi$ denotes the orientation matrix of frame R_t with respect to frame R_0 .

To achieve an incremental translation L, we have:

$$\Delta O_0 O_{n,0} = M_0^n.\Phi.L = [X_{t,0}] .\Delta X \tag{4.29}$$

where the translation vector $L = [\Delta X \ 0 \ 0]$

4.7 Incremental rotation of the tool frame

In the following we study the effect on frame R_n of rotating the tool frame by an incremental rotation Ψ (psi) . The rotation operator Ψ is defined with reference to the tool frame R_t which means that Ψ can be represented by a set of at most three rotations about the axes of the tool frame.

Initially, the orientation matrix of frame R_n is M_0^n and that of frame R_t is $M_0^n.\Phi$, respectively. Also the vectors $O_0O_{n,0}$ and $O_0O_{t,0}$ denotes the position vectors of the origin for frames R_n and R_t, respectively.

After completion of the incremental rotation, the tool frame orientation becomes:

$$M_0^n.\Phi.\Psi$$

The product denotes the new orientation matrix of frame R_t with respect to the fixed frame R_0 . As the effector frame R_n is rigidly attached to R_t, then frame R_n will be affected by its orientation and also its position. Denote by M the new orientation matrix of R_n . As both frames R_n and R_t are rigidly attached, the orientation of the tool frame, which is $M_0^n.\Phi.\Psi$, is equal to the product of the orientation matrix M of R_n by the tool orientation matrix Φ, we have:

$$M.\Phi = M_0^n.\Phi.\Psi$$

The orientation matrix M of frame R_n can then be obtained by pre-multiplying the previous Equation by the matrix Φ^{-1}, we obtain:

$$M = M_0^n.\Phi.\Psi.\Phi^{-1}$$

As Φ is at most the product of three rotations matrices, then its inverse is equal to its transpose, we have:

$$M = M_0^n.\Phi.\Psi.\Phi^t$$

In the following we evaluate the incremental translation of frame R_n as a result of the previously defined incremental rotation of the tool frame. Before performing the rotation of the tool frame, the position of the origin of frame R_t can be evaluated as follows:

$$O_0O_{t,0} = O_0O_{n,0} + M_0^n.O_nO_{t,n} \tag{4.30}$$

After completion of the rotation, frame R_t is only rotated by the operator Ψ, then its origin is not affected by that rotation, we can write:

$$O_0O_{t,0} = O_0O'_{n,0} + M_0^n.\Phi.\Psi.\Phi^t.O_nO_{t,n} \tag{4.31}$$

where $O_0O'_{n,0}$ denotes the position vector of frame R_n after completion of the rotation. By differentiating the two previous Equations, we obtain the increment vector $\Delta O_0O_{n,0}$ of frame R_n:

$$\Delta O_0O_{n,0} = M_0^n.\left(I - \Phi.\Psi.\Phi^t\right) O_nO_{t,n} \tag{4.32}$$

The knowledge of the increment vector $\Delta O_0O_{n,0}$ and the orientation matrix $M_0^n.\Phi.\Psi.\Phi^t$ for the effector frame, allow finding the arm solution in order to set the effector position and orientation as required by the tool frame. Following the translation of the tool frame only the effect on the effector frame is useful because the setting of the arm requires the knowledge of the position vector and the orientation matrix of the effector frame R_n . In other words the origin O_n must be translated by the vector $\Delta O_0O_{n,0}$ when the tool frame is rotated by an angle Φ .

Assume we want the tool, which could be a welding gun, to describe a cylindrical trajectory as shown in the Figure. The robot arm is to move outside the cylindrical trajectory such that the tool frame origin moves on the lateral envelop of the cylinder. This can be accomplished by selecting the tool frame of reference as a frame identical to R_n, after translating it along the Z_n axis such that the frame origin becomes on the cylinder axis. In this case we have:

$$\Phi = I \text{ and } O_nO_{t,n} = (0\ 0\ L)^t$$

where I is the identity matrix and L is a scalar that denotes the distance from the effector frame origin to the tool frame origin. Since, the tool frame becomes identical to the cylinder frame, a rotation of the tool frame such that $\Psi = ROTY(\phi)$ would rotate the tool frame by an angle Ψ with respect to its previous orientation M_0^n . In this case, the increment vector will be:

$$\Delta O_0O_{n,0} = M_o^n.\left(I - ROTY(\phi)\right).O_nO_{t,n} \tag{4.33}$$

where the matrix $(I - ROTY(\phi))$ will be evaluated as follows:

$$(I - ROTY(\phi)) = \begin{bmatrix} 1 - C(\phi) & 0 & -S(\phi) \\ 0 & 1 & 0 \\ -S(\phi) & 0 & 1 - C(\phi) \end{bmatrix} \tag{4.34}$$

4.8 Incremental translation and rotation

In the following we study the effect on frame R_n of an incremental translation L and an incremental rotation Ψ of the tool frame. The incremental translation consists of translating the origin of the tool frame by a vector L which is defined with reference to R_t, i.e., vector L defines a relative motion of the tool frame. The incremental rotation Ψ defines the rotation matrix between the final and initial orientations of the tool frame. This means that Ψ is the transfer matrix between the new and old tool frame.

Before achieving the required operation, the tool frame origin is related to the effector frame origin by the following relation:

$$O_0O_{t,0} = O_0O_n, 0 + M_0^n.O_nO_{t,n} \tag{4.35}$$

After completion of the incremental translation and rotation, the tool frame orientation becomes:

$$M_0^n.\Phi.\Psi$$

The product denotes the new orientation matrix of frame R_t with respect to the fixed frame R_0 . As discussed in the previous Section, the new orientation matrix of the effector frame becomes:

$$M_0^n.\Phi.\Psi.\Phi^t$$

The product denotes the new orientation matrix of frame R_n with respect to the fixed frame R_0 . As the effector frame R_n is rigidly attached to R_t, then frame R_n will be affected by its orientation and also its position. The origin of the tool frame will be evaluated as follows:

$$O_0O_t, 0^{'} = O_0O_n, 0^{'} + M_0^n.\Phi.\Psi.\Phi^t.O_nO_{t,n} \tag{4.36}$$

where $O_0O_{t,0}^{'}$ and $O_0O_{n,0}^{'}$ denote the vectors of the origin of frames R_t and R_n after completion of the incremental translation and rotation, respectively. On the other hand, the incremental translation moves the origin of frame R_t by the vector $M_0^n.\Phi.L$, therefore we have:

$$O_0O_{t,0}^{'} = O_0O_{t,0}^{'} + M_0^n.\Phi.L \tag{4.37}$$

By combining these two Equations, we obtain:

$$O_0O_{t,0} = O_0O_n, 0^{'} + M_0^n.\Phi.\Psi.\Phi^t.O_nO_{t,n} + M_0^n.\Phi.L \tag{4.38}$$

By taking the difference between Equations 1 and 2, we obtain the incremental vector that affect the origin of the effector frame:

$$\Delta O_0 O_{n,0} = M_0^n . \left[\left(I - \Phi.\Psi.\Phi^t \right) O_n O_{t,n} + \Phi.L \right] \tag{4.39}$$

The knowledge of the increment vector $\Delta O_0 O_{n,0}$ and the orientation matrix $M_0^n.\Phi.\Psi.\Phi^t$ of the effector frame, allow finding the arm solution in order to set the effector position and orientation as required by the tool frame. Following the translation of the tool frame, only the effect on the effector frame is useful because the setting of the arm requires the knowledge of the position vector and the orientation matrix of the effector frame R_n .

The incremental translation and rotation is very useful for programming tasks that use sensory feedback in a closed loop form. Generally, these tasks consists of adjusting the position and orientation of the tool frame by converting information issued from the robot environment into incremental corrections. For example, the operation of inserting a peg into a hole requires the use of a force sensor to sense the friction forces between the peg and the hole. To achieve the insertion, one needs to convert the force information into incremental correction with respect to the peg position and orientation. To obtain good sensitivity, sensors are placed close to the robot hand effector. The knowledge of the tool position and orientation with respect to the effector frame allow converting these forces into force with respect to the tool frame. Therefore, corrections on position and orientation of the tool frame require the evaluation of their effect on the effector frame. These transformations have been presented in this Section.

Exercises

1. A motion primitive, that belong to the effector level, can be implemented using the form: MOVE TO $X,\ Y,\ Z,\ \Phi_1,\ \Phi_2,\ \Phi_3$ where $O_oO_{n,o} = (X, Y, Z)^t$ and Φ_1, Φ_2 and Φ_3 represent the robot hand position and orientation, respectively.

   (a) Find the required mathematical transformation as to generate the input parameters of the inverse geometrical model: $O_oO_{6,o}$ and M_o^6 .

   (b) Evaluate the number of fundamental operation required to evaluate M_o^6 .

2. Give the block diagram of a motion coordination system that uses the geometric transform and the 3 Euler angles as inputs and generates the geometric angles of the robot.

3. It is desired to store the trajectory of the robot hand into the computer memory. We assume that n references are required in order to play back the trajectory

   (a) Investigate the memory requirement when each reference consists of $\{O_oO_n, M_o^n\}$, or $O_oO_n, \Phi_1, \Phi_2, \Phi_3$, or $\{O_oO_6, X_6, Y_6\}$.

4. What are the advantages and drawbacks of using the Euler angles for referencing the robot hand in terms of ease of programming, memory requirements, rounding error, and accuracy.

5. What are the advantages and drawbacks of defining motion primitives with respect to the tool frame of reference. Study the implication of this question on the use of structured motion primitives, their computational complexity, and their applications.

6. Develop the following idea: A programming environment whose specifications are related to the object is more suitable for task programming in robotics.

7. Find the general relation between a tool frame R_t whose origin is defined by having $O_nO_{t,n}$ and its orientation is defined by matrix Φ with respect to the effector frame.

   (a) Study the case of a translation of R_t and find the corresponding motion of R_n .

   (b) Study the case of a rotation of R_t and find the corresponding motion of R_n .

8. Find the general relation between a tool frame R_t whose origin is defined by having $O_nO_{t,0}$ and its orientation is defined by matrix Φ with respect to the base frame of reference R_0 .

   (a) Study the case of a translation of R_t and find the corresponding motion of R_n .

   (b) Study the case of a rotation of R_t and find the corresponding motion of R_n .

9. We assume that an error of 10^{-5} occurs every time two floating-point numbers are multiplied. Study the effect of these errors on multiple matrix product of the form $M \leftarrow M\Psi$. What is the maximum number of matrix product that are allowed such that the norms $|X|, |Y|, |Z|$ are within the range $[1-\epsilon, 1+\epsilon]$ and $\epsilon = 10^{-3}$.

10. We assume a database contains information about the work space objects such that geometric model including frame of references for each object.

    (a) Suggest a method for determining the position and orientation of the tool frame every time an object is to be manipulated by the robot.

    (b) Define the syntax of motion primitives that allow grasping and positioning and orienting, and performing some tasks such as pick and place with objects.