6. Kinematics of Serial Chain Manipulators

6.1 Assignment of reference frames

In a multi-degree-of-freedom mechanical system consisting of rigid bodies, it is useful to attach reference frames to each rigid body. In robotics, most researchers adopt a convention for attaching reference frames that is due to Denavit and Hartenberg\textsuperscript{1}. It assumes that the mechanical system is a serial chain and each joint is an axial joint. By axial joint, we mean that the joint has an axis that defines the relative motion of the connecting bodies. For example, revolute, prismatic and helical joints are axial joints, while spherical joints are not.

We deviate from the so-called \textit{D-H} approach in the interests of keeping things simple. As will become apparent, there is a minor disadvantage of our approach. Quite simply, the user has to make many choices and there is no unique assignment of reference frames. We will assume that we are dealing with serial chains consisting of \textit{binary} links (each link, except the first and last link, is connected to two other links) and \textit{simple} joints (each joint connects two links). Further, as explained above, each joint is an axial joint. Our approach has the following important steps.

1. Identify all the \textit{joint axes} (For a revolute, cylindrical, or a screw joint, the axis is obvious. For a prismatic joint we can pick any axis that is parallel to the direction of translation). Number them from 1 through \(n\).

2. Identify pairs of adjacent links for each axes so that we have two links on either side of each joint and two joints on either side of each link. The links are numbered from 0 through \(n\). 0 is the base link that is connected to the first joint (joint 1). Next comes link 1 which in turn is connected to joint 2 and so on. The last link is link \(n\).

3. We will assign reference frames to each link so that one of three coordinate axes (\(x, y, \) or \(z\)) is aligned along the axis of the distal joint (the joint further away from the base).
6.2 Transformations between reference frames

Once the reference frames are assigned, we must develop transformations between the reference frames. In order to do this most conveniently, it is beneficial to follow the following two additional steps.

4. Position or draw the manipulator in a home configuration. This is the configuration that will be the reference for all displacements. We will consider all joint displacements to be zero at the home configuration.

5. Identify constant offsets (translations) and rotations between reference frames. Since each adjacent pair of reference frames spans a joint, the transformation between the reference frames must include a joint displacement variable. This variable is zero at the home configuration but non zero at other configurations.

6.3 Direct Kinematics

We want to relate the position and orientation of the most distal reference frame (and therefore the most distal link) to the position and orientation of the base frame (attached to the most proximal link) in terms of the joint displacement variables. The basic idea is to construct transformation matrices, \( A_i \), for every pair of adjacent frames and then compose these transformations.

**Example 1**

The forward kinematics for the Stanford Arm like \( R-R-P-R-R-R \) structure shown in Figure 1 is given by the chain of transforms:

\[
\begin{align*}
&\text{Trans}(z, a_1) \text{ Rot}(z, \theta_1) \text{ Trans}(x, a_2) \text{ Rot}(x, \theta_2) \text{ Trans}(z, d_3) \text{ Rot}(x, \theta_4) \text{ Trans}(z, a_4) \\
&\text{Rot}(y, \theta_5) \text{ Rot}(z, \theta_6)
\end{align*}
\]

---

Figure 1  The Stanford Arm like $R-R-P-R-R-R$ structure.

In Table 1, $s_i$ is the vector of Plücker coordinates (normalized) in frame $i$ that describes the axis of rotation (or translation) which characterizes the motion of frame $i$ relative to frame $i-1$. 
Figure 2  The Stanford Arm

Table 1  The six unit line (Plücker) vectors for the axes of the Stanford Arm like R-R-P-R-R-R structure.

<table>
<thead>
<tr>
<th>Axis</th>
<th>Frame</th>
<th>Description</th>
<th>6×1 line vector, s_i</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>rotation about z axis</td>
<td>[0, 0, 1; 0, 0, 0]^T</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>rotation about x axis</td>
<td>[1, 0, 0; 0, 0, 0]^T</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>translation along z axis</td>
<td>[0, 0, 0; 0, 0, 1]^T</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>rotation along x axis</td>
<td>[1, 0, 0; 0, 0, 0]^T</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>rotation about y axis</td>
<td>[0, 1, 0; 0, 0, 0]^T</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>rotation about z axis</td>
<td>[0, 0, 1; 0, 0, 0]^T</td>
</tr>
</tbody>
</table>

We consider 7 reference frames, numbered \( \{F_0\} \) through \( \{F_6\} \) or simply 0 through 6. The zeroth frame is \( x\text{-}y\text{-}z \), shown in the figure. The transforms below show the intermediate frames:

\[ ^0A_1 = \text{Trans}(z, a_1) \text{ Rot}(z, \theta_1) \]
\[ ^1A_2 = \text{Trans}(x, a_2) \text{ Rot}(x, \theta_2) \]
\[ ^2A_3 = \text{Trans}(z, d_3) \]
\[ ^3A_4 = \text{Rot}(x, \theta_4) \]
\[ ^4A_5 = \text{Trans}(z, a_4) \text{ Rot}(y, \theta_5) \]
\[ ^5A_6 = \text{Rot}(z, \theta_6) \]
Note the key rules governing the assignment of intermediate frames are:

1. The homogeneous transformation matrix relating adjacent frames must have only one joint variable; and
2. The \( i \text{th} \) axis of rotation or translation must be easily identifiable in the \( i \text{th} \) frame. [If the \( i \text{th} \) joint is rotational joint, make sure its axis passes through the origin of the \( i \text{th} \) frame.]

Example 2

The forward kinematics for the PUMA manipulator shown in Figure 3 is given by the chain of transforms:

\[
\text{Trans}(z, a_1) \text{ Rot}(z, \theta_1) \text{ Trans}(x, a_2) \text{ Rot}(x, \theta_2) \text{ Trans}(z, a_3) \text{ Rot}(x, \theta_3) \text{ Trans}(z, a_4)
\]
Trans\((y, -a_5)\) Rot\((z, \theta_4)\) Rot\((x, \theta_5)\) Rot\((z, \theta_6)\)

We consider 7 reference frames, numbered \(\{F_0\}\) through \(\{F_6\}\) or simply 0 through 6. The zeroth frame is \(x-y-z\), shown in the figure. The transforms below show the intermediate frames:

\[
\begin{align*}
^0A_1 &= \text{Trans}(z, a_1) \text{ Rot}(z, \theta_1) \\
^1A_2 &= \text{Trans}(x, a_2) \text{ Rot}(x, \theta_2) \\
^2A_3 &= \text{Trans}(z, a_3) \text{ Rot}(x, \theta_3) \\
^3A_4 &= \text{Trans}(z, a_4) \text{ Trans}(y, -a_5) \text{ Rot}(z, \theta_4) \\
^4A_5 &= \text{Rot}(x, \theta_5) \\
^5A_6 &= \text{Rot}(z, \theta_6)
\end{align*}
\]

Table 2 The six unit line (Plucker) vectors for the axes of the Puma Manipulator.

<table>
<thead>
<tr>
<th>Axis</th>
<th>Frame</th>
<th>Description</th>
<th>6×1 unit line vector, (s_i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>rotation about (z) axis</td>
<td>([0, 0, 1; 0, 0, 0]^T)</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>rotation about (x) axis</td>
<td>([1, 0, 0; 0, 0, 0]^T)</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>rotation along (x) axis</td>
<td>([1, 0, 0; 0, 0, 0]^T)</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>rotation about (z) axis</td>
<td>([0, 0, 1; 0, 0, 0]^T)</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>rotation along (x) axis</td>
<td>([1, 0, 0; 0, 0, 0]^T)</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>rotation about (z) axis</td>
<td>([0, 0, 1; 0, 0, 0]^T)</td>
</tr>
</tbody>
</table>
Sample Maple code

```maple
> restart;
> with(linalg):

Library of procedures for direct kinematics.

Elemental translations and rotations

> TransX:=x-> vector([x, 0, 0]):
> TransY:=y-> vector([0, y, 0]):
> TransZ:=z-> vector([0, 0, z]):
> RotX:= t -> array(1..3,1..3,[[1, 0, 0], [0, cos(t), -sin(t)], [0, sin(t), cos(t)]]):
> RotY:=t -> array(1..3,1..3,[[cos(t), 0, sin(t)], [0, 1, 0], [-sin(t), 0, cos(t)]]):
> RotZ:=t -> array(1..3,1..3,[[cos(t), -sin(t), 0], [sin(t), cos(t), 0], [0, 0, 1]]):

Homogeneous transformation matrix from rotation matrix and translation vector

> HomTrans:=(R, d) -> array(1..4,1..4,[[R[1,1], R[1,2], R[1,3], d[1]], [R[2,1], R[2,2], R[2,3], d[2]], [R[3,1], R[3,2], R[3,3], d[3]], [0, 0, 0, 1]]):

Rotation matrix and translation vector from the homogeneous transformation matrix from

> TransHomTrans:= A-> vector([A[1, 4], A[2, 4], A[3, 4]]):

Skew symmetric matrix operator corresponding to a 3x1 vector.

> SkewMatrixOp := a -> array(1..3,1..3,[[0, -a[3], a[2]], [a[3], 0, -a[1]], [-a[2], a[1], 0]]):

6x6 twist transformation matrix (Gamma) corresponding to a 4x4 homogeneous transformation matrix.

> AdjOp:= proc(A) local X, R; R:=transpose(RotHomTrans(A)); evalm(SkewMatrixOp(TransHomTrans(A)) &* R); array(1..6, 1..6, [[R[1, 1], R[1, 2], R[1, 3], 0, 0, 0], [R[2, 1], R[2, 2], R[2, 3], 0, 0, 0], [R[3, 1], R[3, 2], R[3, 3], 0, 0, 0], [X[1, 1], X[1, 2], X[1, 3], R[1, 1], R[1, 2], R[1, 3]], [X[2, 1], X[2, 2], X[2, 3], R[2, 1], R[2, 2], R[2, 3]], [X[3, 1], X[3, 2], X[3, 3], R[3, 1], R[3, 2], R[3, 3]]) end:

3x1 vector corresponding to a skew symmetric matrix operator, 6x1 twist vector corresponding to a twist matrix.

> ExtractVector:= X -> vector([-X[2,3], X[1,3], -X[1,2]]): ExtractTwist:= X -> vector([-X[2,3], X[1,3], -X[1,2], X[1,4], X[2, 4], X[3,4]]):

Inverse of a homogeneous transformation matrix

> InvHomTrans:=proc(A) local R; R:=transpose(RotHomTrans(A)); HomTrans(R, scalarmult(multiply(R, TransHomTrans(A)), -1)) end:

Abbreviate cos(ti) by ci, sin(ti) by si.

> alias(seq(c.i=cos(t.i), i=1..6), seq(s.i=sin(t.i), i=1..6)):

Direct Kinematics

> A1:=HomTrans(RotZ(t1), TransZ(a1)):  
> A2:=HomTrans(RotX(t2), TransX(a2)):  
> A3:=HomTrans(RotZ(0), TransZ(d3)):  
> A4:=HomTrans(RotX(t4), TransX(0)):  
> A5:=HomTrans(RotY(t5), TransZ(a4)):  
> A6:=HomTrans(RotZ(t6), TransZ(0)):  
> Tool:=HomTrans(RotZ(0), TransZ(a5)):  
> A:=multiply(A1, A2, A3, A4, A5, A6):
```

-7-