2.3. Homogeneous Coordinates

We should be happy by now. We have a way to represent rigid motion that is fairly simple and intuitive:

$$P' = RP + T \tag{2.11}$$

So what is this homogeneous coordinates thing? Well, it turns out some people were not happy and looked for ways to make the simple Eq. (2.11) even simpler. The problem was the following. The rigid transformation involves two data structures and these are treated differently: the one is multiplicative the other is additive. And if one wants to apply two rigid transformations then one would get three terms instead of one. Furthermore, there are operations that cannot be done by just applying Eq. (2.11), the most useful of which is projection. But most important, some mathematicians had scribbled down some theory, called *Projective Geometry*, and were looking desperately something to try it out, pretty much like a hammer looking for a nail. Luckily, very little of this thing survives today, since every transformation, projection, etc can be done more conveniently with a few lowly matrices and vectors. All that survives and is useful from this theory can be explained in about half a lecture hour. So assume as before that

$$P = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$$

and

$$P' = \begin{bmatrix} p'_x \\ p'_y \\ p'_z \end{bmatrix}$$

are the coordinates of a 3-D point before and after the motion, and

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

where R and T are the rotation and translation as before. It is easy then to verify that

$$\mathbf{P'} = \mathbf{TP} \tag{2.12}$$

where

$$\mathbf{P} = \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}$$

and similarly for $\mathbf{P}'$. Symbols $\mathbf{P}$, $\mathbf{P}'$ and $\mathbf{T}$ are called homogeneous vectors and homogeneous transformation respectively.

With this simple change of notation we managed to simplify a simple formula. The gains in simplicity increase dramatically if we change coordinate systems several times. For instance, if we change rigid motion twice we end up with three terms if we use Eq. (2.11) but a single term if we use Eq. (2.12).

The interpretation of a homogeneous vector is obvious. Just throw away the last element which is unity and what is left is a familiar vector. But what happens if it is not unity? Although it is unlikely to encounter such a situation with rigid transformations, we will see that if we use homogeneous coordinates for projection, we will have to deal with this. But we should not panic. Two homogeneous vectors represent the same thing if the one is a scaled version of the other (provided that the scale factor is not zero, of course). So we divide the homogeneous vector by the forth element and bring it in the desired form.

3. Robot Arm Kinematics

Robotic manipulators can be of many kinds, but we will concentrate on the open chain variety, that is the arm can be thought of as a series of links joined at joints and every joint having one degree of freedom, that is one variable parameter controlled by a motor. Joints come in two flavours. Revolute (a.k.a. rotational), and prismatic (a.k.a. telescopic).

To represent the chain of links in a mathematically tractable fashion we have to eliminate all details that are not immediately useful. So a joint is just an axis around which the to links can rotate (or slide) relative to each other and a link just keeps the axes of its two joints at a fixed relative position (Fig. 3.1).

So we need two numbers to describe link $i-1$: the length a_{i-1} of the common normal between the two axes and the angle α_{i-1} (twist) of the two axes relative to each other. And we need two more to describe joint i: the distance d_i between the feet of the common normals with the previous axis $i-1$ and the next axis $i+1$, and the angle θ_i of these two common normals. Either θ_i or d_i are variable and controlled by a motor.

Now to do any geometry on the links and joints we have to define coordinate systems. Every link will have a coordinate system rigidly attached to it. We can put it anywhere we like and of course we like it where it is easier to do the math. After some heavy thinking, the originators of this idea, decided to put the origin of the coordinate system of the $i-1$ link on the $i-1$ axis, at the foot of the common normal with the next axis. The Z_{i-1} axis is in the direction of the $i-1$ axis and the X_{i-1} axis is in the direction of the common normal with the next axis. And the coordinate system of the i link is on the i axis, at the foot of the common normal with axis $i+1$, the Z_i axis is on the axis i and the X_i axis is on the common normal with the $i+1$ axis. So the transformation from the coordinate system $i-1$ to system i involves a translation along axis X_{i-1} axis by a_{i-1}, rotation around the same axis by α_{i-1}, translation along Z_i by d_i and rotation around the same axis by θ_i which can be written as

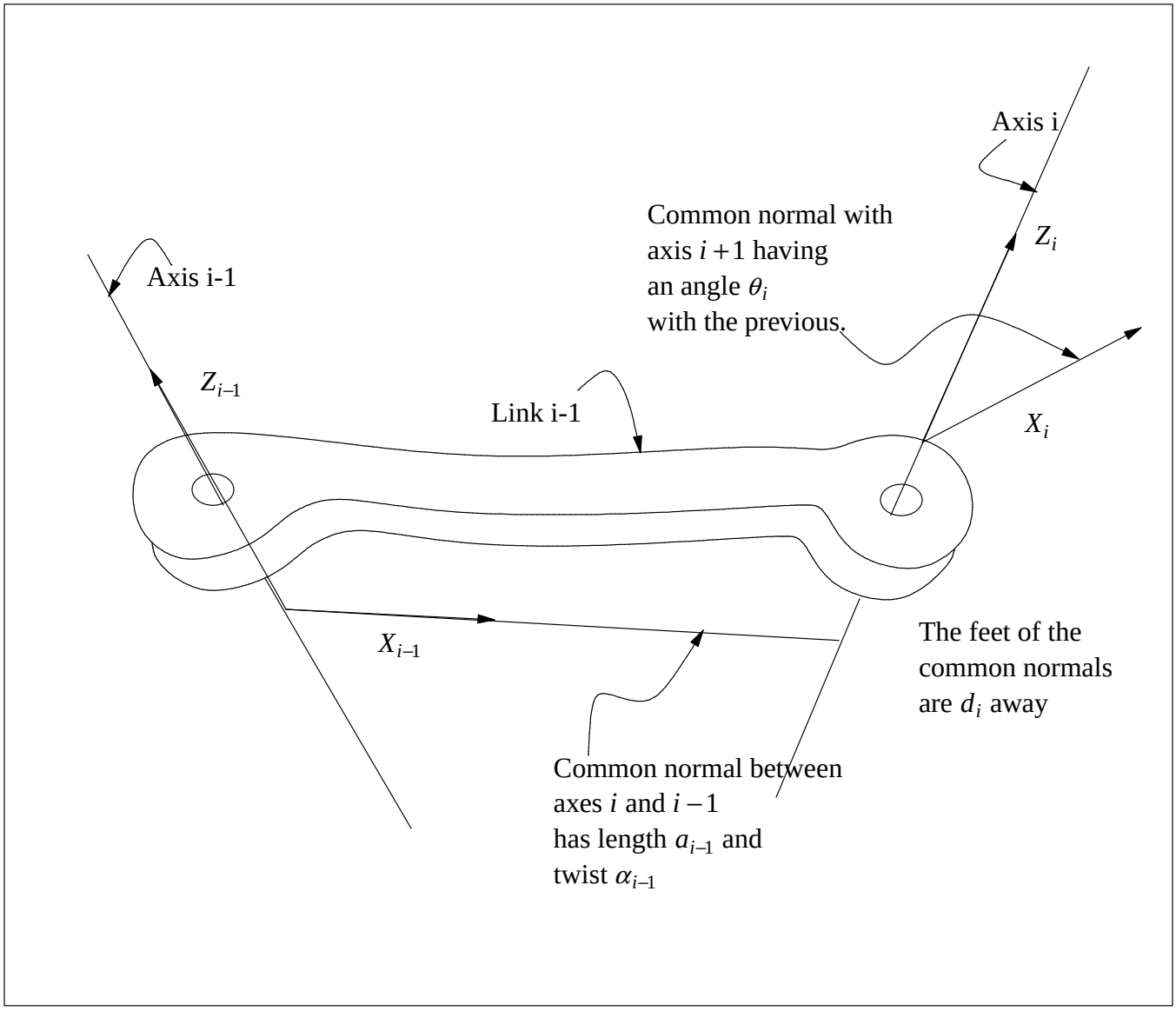

Figure 3.1: The link $i-1$ holds axes $i-1$ and i at a fixed distance a_{i-1} and angle α_{i-1} relative to each other. Axis i holds the two neighboring common normals at a distance d_i and angle θ_i.

$$ {}^{i-1}_{i}\mathbf{T} = \mathbf{T}_X(a_{i-1})\mathbf{R}_X(\alpha_{i-1})\mathbf{T}_Z(d_i)\mathbf{R}_Z(\theta_i) $$

which can be used to transform point ${}^{i}P$ expressed in coordinate system i to point ${}^{i-1}P$ expressed in coordinate system $i-1$

$$ {}^{i-1}P = {}^{i-1}_{i}\mathbf{T}\,{}^{i}P. $$

The above notation that uses left superscripts and subscripts to indicate the coordinate systems involved is very useful in this business. A 7 degree of freedom robot can easily

have 10 coordinate systems involved, so one would need up to 90 different transformations and as many symbols for them. Few alphabets can provide enough single literal symbols.

The last thing we have to know to fully describe an arm with matrices is a couple more conventions about what to do with the zeroth and last frames, and what to do when the axes are parallel and there is no unique common normal. The zeroth frame (a.k.a base frame) for a revolute (rotary) first joint shares origin with the first frame, Its Z_0 axis is the same as the first frame's axis Z_1 and its X_0 axis is the same as the first frame's axis X_1 *only* if $\theta_1 = 0$. If the joint is prismatic (telescopic) then X_0 is parallel to X_1 and the origins of frames 0 and 1 coincide only if $d_1 = 0$. The last frame T (a.k.a. tool frame), if it is revolute, has its origin at the foot of the common normal with the previous frame (since there is no next frame) and the orientation of the X_T axis is the same as X_{T-1} when the last θ is zero. If it is prismatic then axis X_T is parallel to the previous X_{T-1} axis and the origin coincides with the origin of the previous frames *only* if $\theta_T = 0$.

As it must have become obvious, the convention shows a marked preference for simplicity since roboticists are paid to get the job done, not to brag about their smarts. The same principle of simplicity can be applied to decide what to do if two axes are parallel. In this case we select the origin to be the foot of the common normal with the previous axis. If this fails because all axes right down to the first are parallel and none of them prismatic, we look at the next axis, the one after the next etc until we find one that is not parallel to the rest or one that is prismatic and choose all the origins in such a way that as many as possible d_i's are zero.