Contents

Section 3 Kinematics

3.1 Introduction

The location of an object in a 3D space can be found by using coordinate frames and by applying transformations that are fully described by the transformation matrix. The original (world) reference frame can be attached on the robot base or on the end effector.

Kinematics is the relationship between the positions, velocities and accelerations of the links of a manipulator. The task here is to define the position and orientation of the end effector with respect to the robot base by the transformation matrices that were derived in the previous paragraphs. The manipulator will be considered to be a series of links that are connected with joints. As before the first link (link 0) is the immobile robot base and the last (link n) is the robot hand. The first joint (joint 1) is the waist and it connects link 0 with link 1. Hence the joint n connects link n-1 with link n. The link that is closer to the base, with regard to a joint, is called the proximal link and the link that is the most distant is called the distal link (here it is the next link since one degree of freedom is assumed). Finally every link has an axis that connects the two joints.

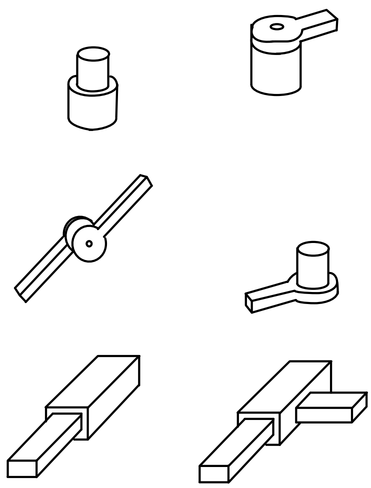

Figure 3.1 Prismatic and revolute joints

It has already been mentioned that there are revolute and prismatic joints, Figure 3.1. To simplify the mathematics these joints will be considered to be of one degree of freedom. Hence in a revolute joint only one rotation is allowed and one translation for the prismatic joint. An axis will describe the translation and the rotation, the joint axis, Figure 3.3. Revolute joints can rotate around their axis and prismatic axes can slide.

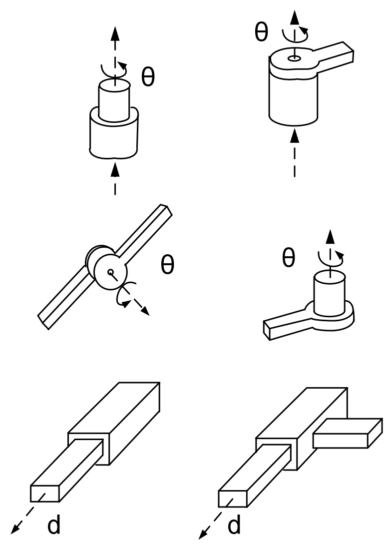

Figure 3.2 Prismatic and revolute joints with the axis of rotation and translation

If the joint axis and the link axis coincide then the joint is called collinear and if the joint axis is perpendicular to the proximal link axis then it is termed orthogonal, Figure 3.3. A new coordinate frame is now going to be attached to each link. The angles and displacements between the links are named as joint coordinates and will define the transformation from the one frame to the other, i.e. from the one link to the other. By combining all these translations and rotations it is possible to describe the last link with respect to the base, or to describe the orientation and the position of a frame that is attached to the last link with respect to the original frame that is attached in the base.

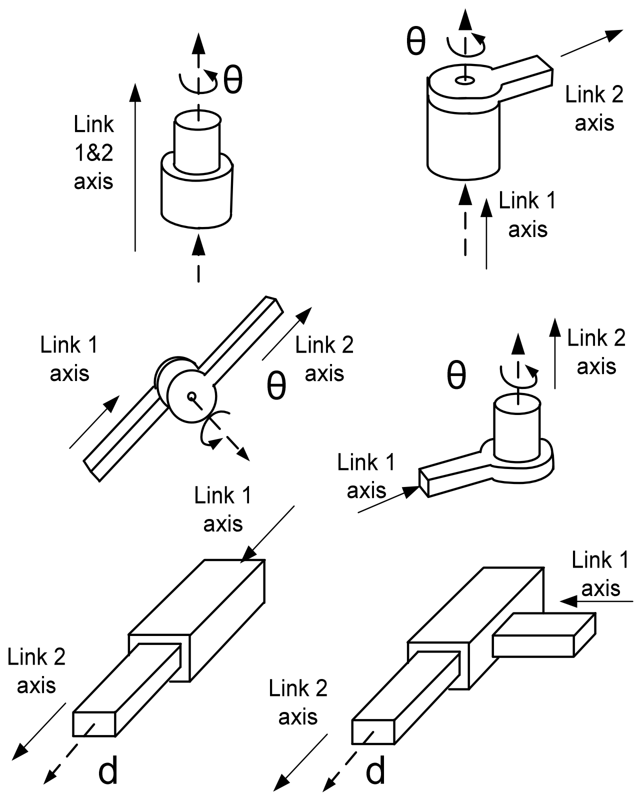

Figure 3.3 Joint and link axes

3.2 Denavit and Hartenberg method

The relationships between the frames can follow many configurations but by far the most popular is the one of "Denavit and Hartenberg". This method of describing the frames in each link is straightforward and it can greatly simplify the mathematics. The whole idea of this method is to **assign reference frames on each link (including the robot base) and to compute the relative transformation(s) between them.**

The first frame {0} will describe the robot base (link 0) and frame {1} will describe the torso (link 1). So we have to define a transformation matrix from {0} to {1}. Thus, we have to properly assign the reference frames of {0} and {1} or in the general case {n-1} and {1} which are connected through the joint n.

The first thing that must be established is the relation between two joints, n-1 and n:

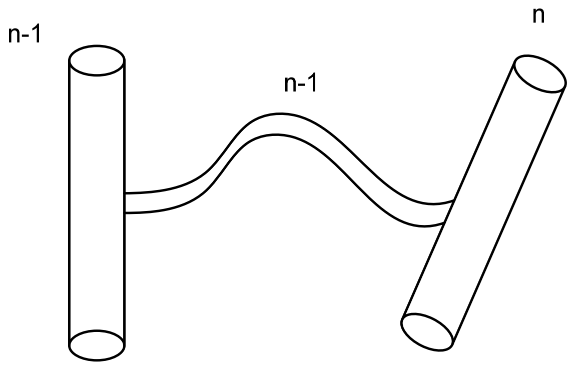

Figure 3.4 A single link with the two joints

As it has been said every joint has 1 degree of freedom and this is described by the joint axis:

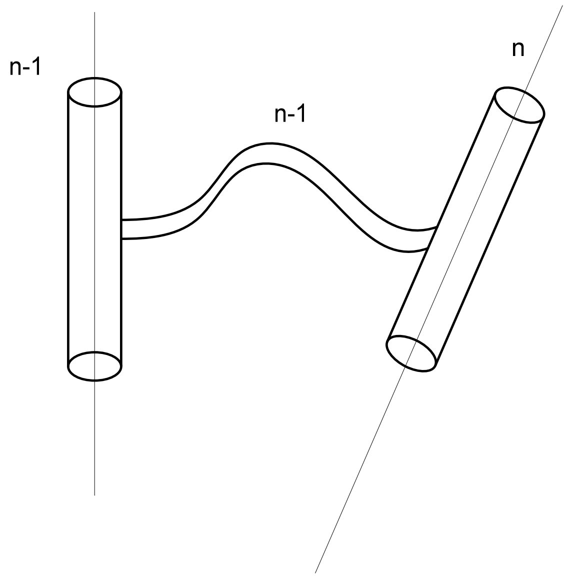

Figure 3.5 Joint axes

Two parameters now can fully define the location of the joint axis n with respect to the n-1. **First of all the link length, l, which is the distance between the two skew lines:**

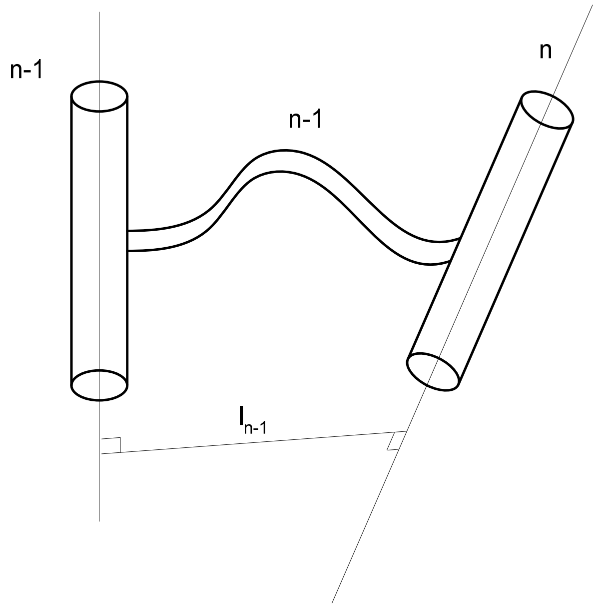

Figure 3.6 Link length

Secondly if a line parallel to the joint axis n is created at the intersection of the joint axis n-1 and the common normal then a new **angle, a, will define the link twist;**

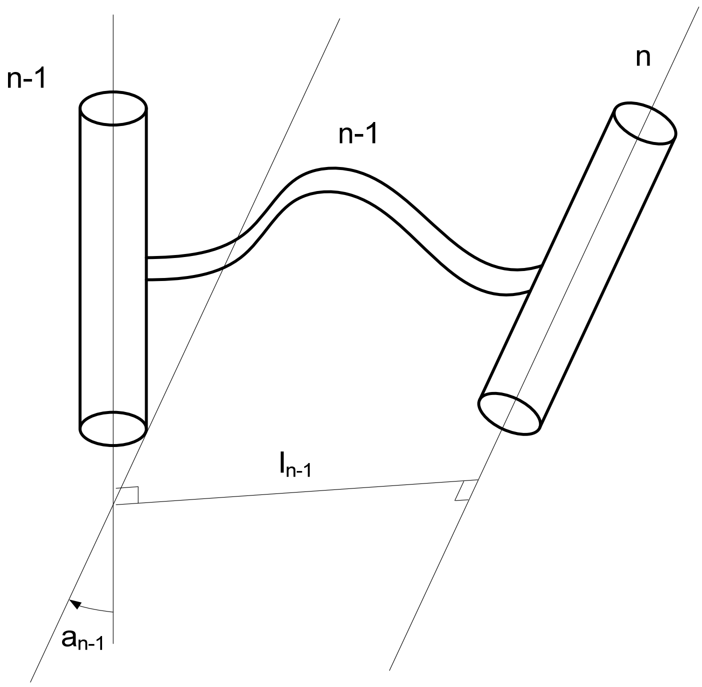

Figure 3.7 Link twist

If there are three joints then two link lengths and two link twists will be defined:

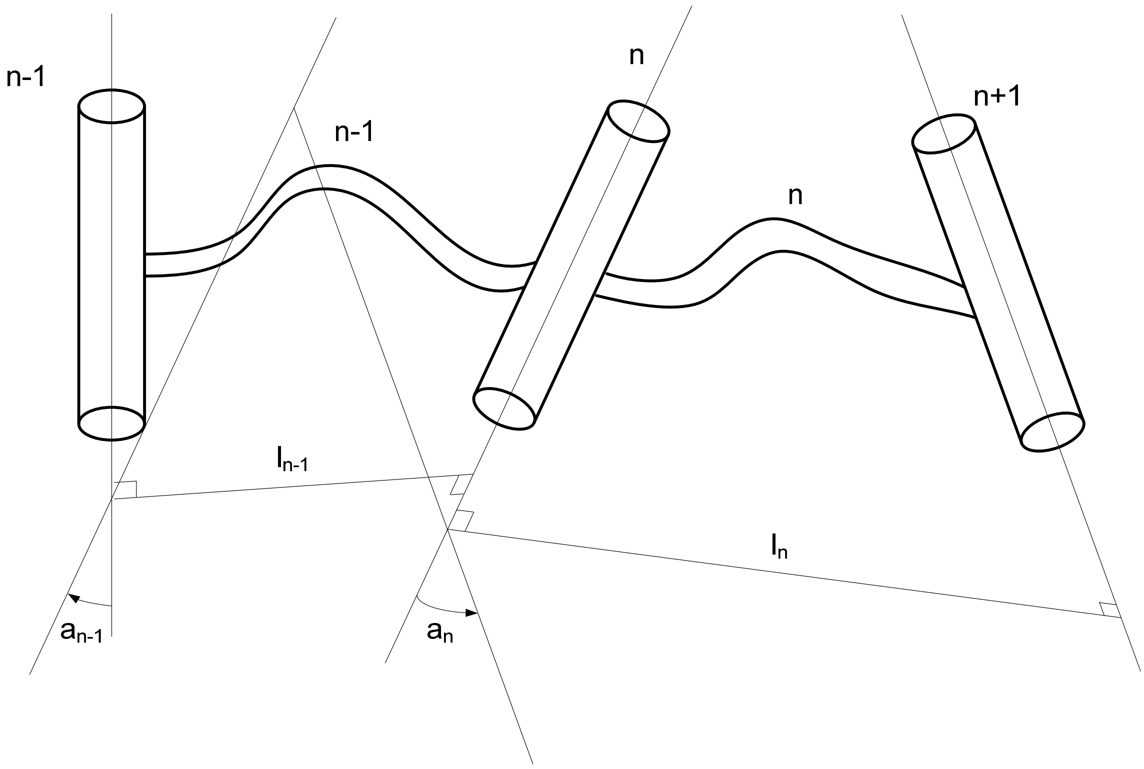

Figure 3.8 Link twists and lengths

The task now is to describe the link n with respect to n-1. To do this two more parameters have to be defined. The first one is **the link offset, d, which is the distance between the points A and B**:

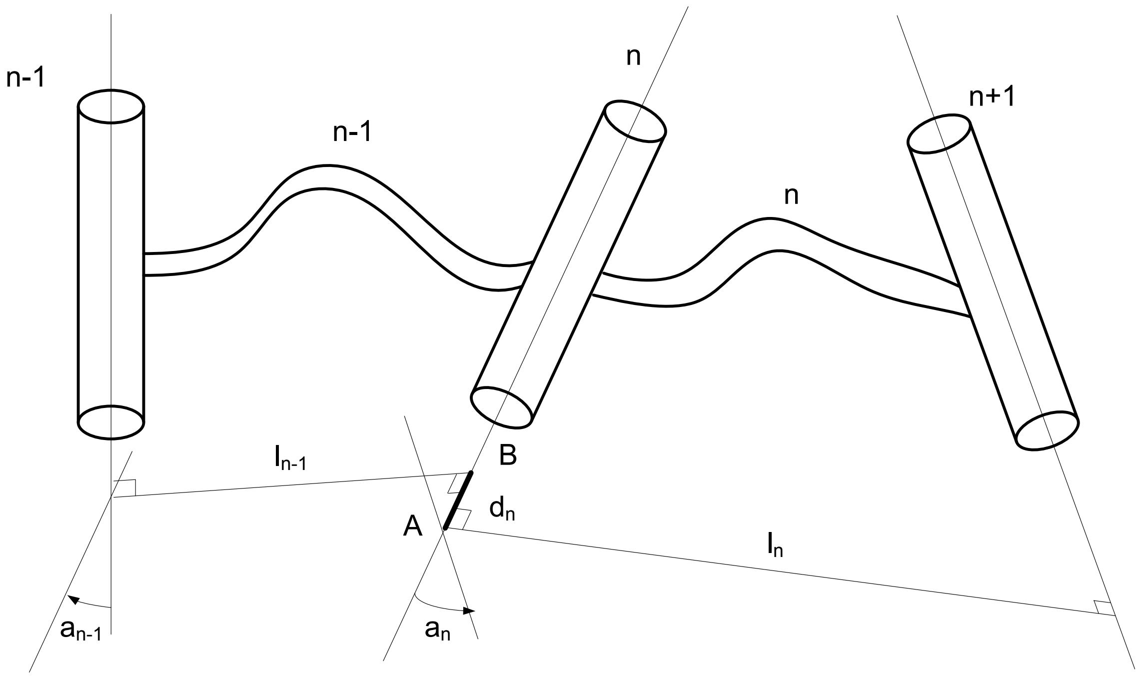

Figure 3.9 Link twists, lengths and offset

It is very important now to notice that if the joint n was prismatic then the link offset would have been variable:

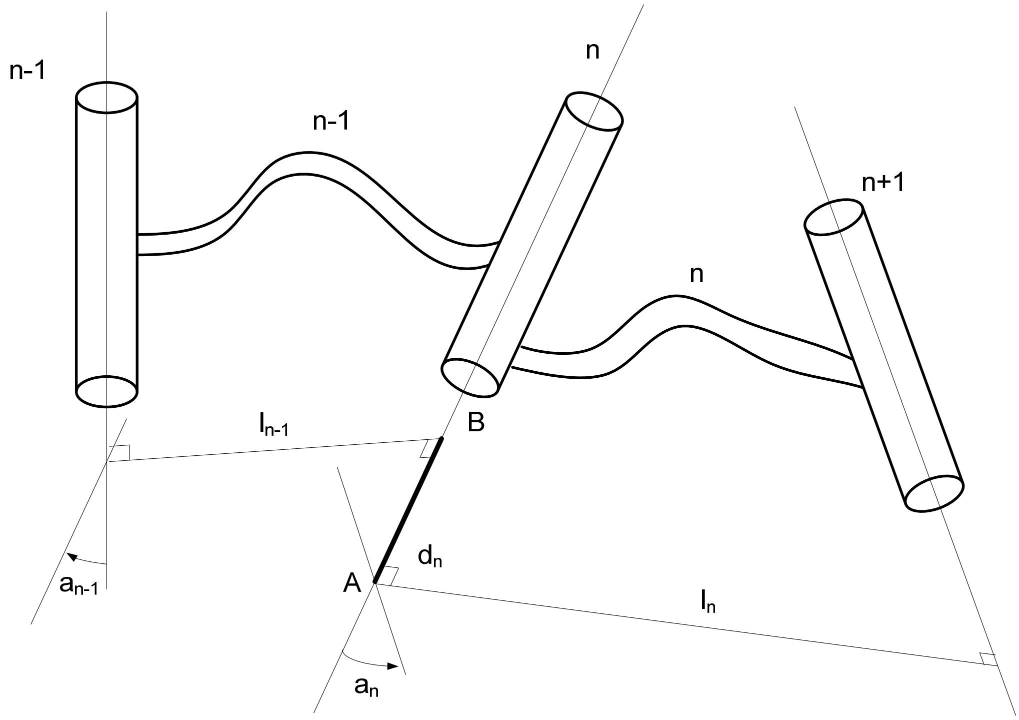

Figure 3.10 Variable link offset

The last parameter now that has to be defined is the **link angle, θ**. To do this, create a parallel to the link length l_n at the intersection of the common normal of joints n-1 and n and the link axis n:

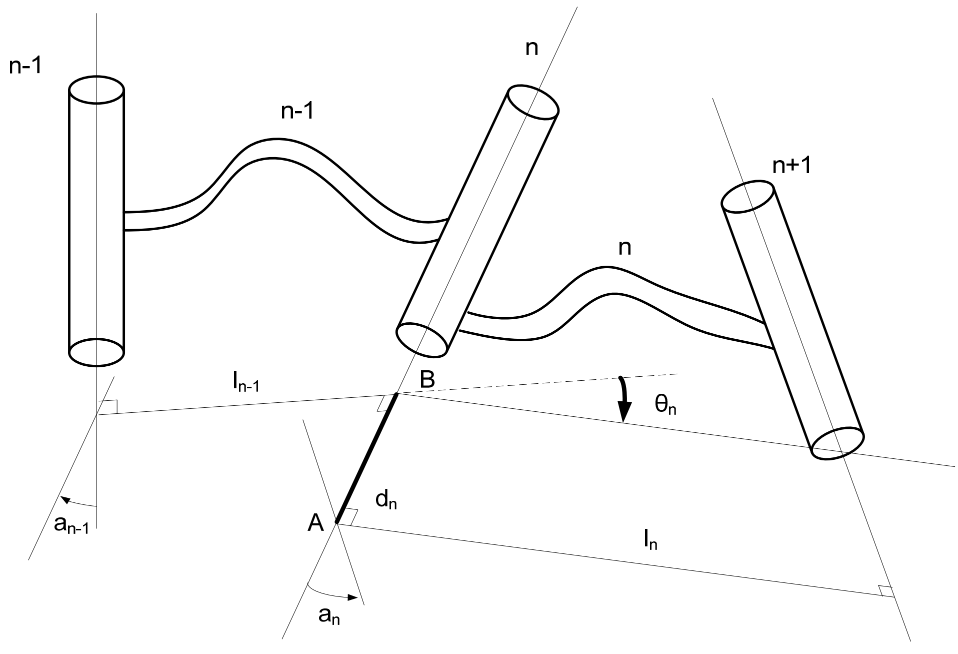

Figure 3.11 Link twists, lengths and joint angle

Notice that if joint n is revolute then the angle θ_n is variable. The link length, twist, offset and angle between the first link (robot base) and the second link (1) are assumed to be zero.
The axis that describes the translation or rotation of a joint is the z-axis. Hence, the z-axis of the frame {n} will coincide with the joint axis n:

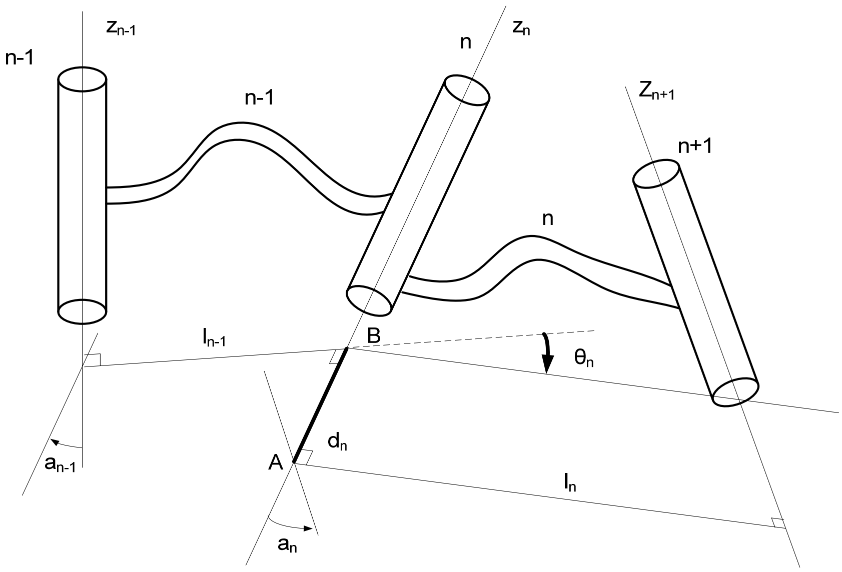

Figure 3.12 z-axes allocation

The origin now of the frame {n-1} is located at the intersection of the line l_{n-1} and the joint axis n-1. The direction of **the x-axis will coincide with the line l_{n-1} and it will point towards joint n**.

The y axis will be found from the right hand rule:

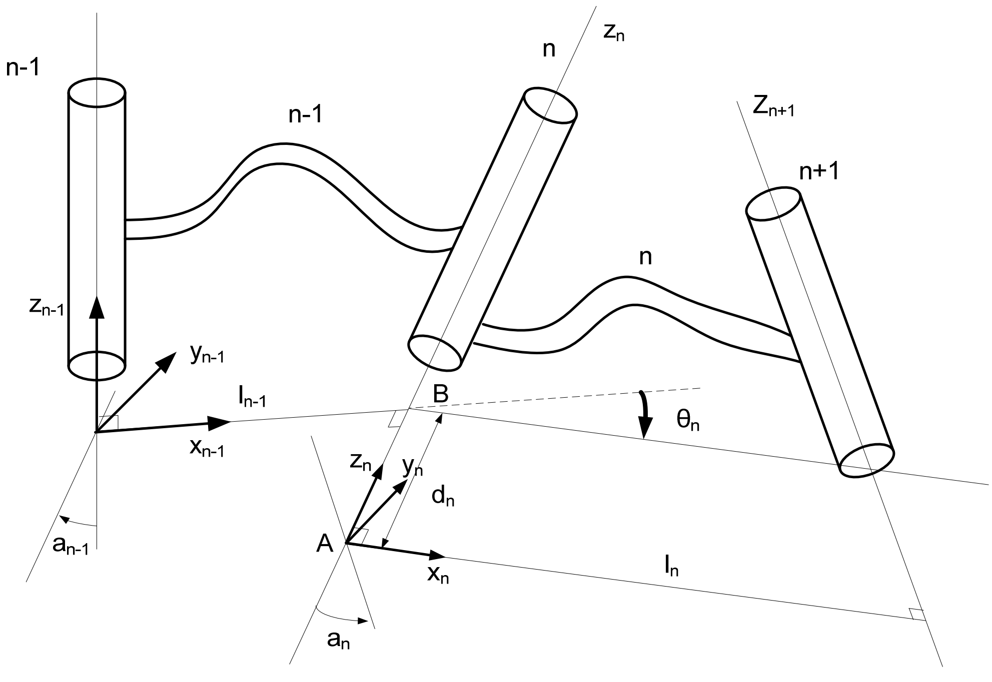

Figure 3.13 Frame allocation

Special Cases
In the case where there is an intersection between the two skew lines the common normal has zero length and the origin has to be placed at the intersection of the two skew lines.
If the two joint axes are parallel there is an infinite number of common normal(s) and hence there can be an arbitrary selection of origins. The most common choice is to select the origin in such location that will make the d_{n-1} zero or the x-axis will have the direction of the previous common normal.

First and last frame
The location of the zero frame can be located anywhere. But is usually chosen to be in such a way that it will coincide with frame {1} when the joint variable is zero.

For the purposes of this module usually all x-axes will remain parallel when the joint variable is zero.

Having assigned the DH link parameters, it is common to see the link parameters summarised in a table

Link Number	l_{n-1}	a_{n-1}	d_n	θ_n
1, ${}^0T_1$	l_0	a_0	d_1	θ_1
2, ${}^1T_2$	l_1	a_1	d_2	θ_2
…..	……	……	……	……
n-1, ${}^{n-2}T_{n-1}$	l_{n-2}	a_{n-2}	d_{n-1}	θ_{n-1}
n, ${}^{n-1}T_n$	l_{n-1}	a_{n-1}	d_n	θ_n

By using all the above parameters the transformation from the frame {n-1} to {n} is given by:

$$\boxed{{}^{n-1}T_n = Rot(x, a_{n-1})Trans(l_{n-1},0,0)Rot(z,\theta_n)Trans(0,0,d_n)}$$

Finally to find the total transformation matrix, multiply all the above matrices:

$$\boxed{{}^RT_H = {}^0T_1\,{}^1T_2\,{}^2T_3 \cdots {}^{n-1}T_H}$$

So for example to go from the robot base to the frame that describes the first link: $\boxed{{}^{n-1}T_n = Rot(x, a_0)Trans(l_0,0,0)Rot(z,\theta_1)Trans(0,0,d_1)}$

3.2.1 Example 1

Find the transformation matrix: ${}^{n-1}T_n$

$${}^{n-1}T_n = Rot(x, a_{n-1})Trans(l_{n-1},0,0)Rot(z,\theta_n)Trans(0,0,d_n)$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(a_{n-1}) & -\sin(a_{n-1}) & 0 \\ 0 & \sin(a_{n-1}) & \cos(a_{n-1}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & l_{n-1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \cos(\theta_n) & -\sin(\theta_n) & 0 & 0 \\ \sin(\theta_n) & \cos(\theta_n) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} 1 & 0 & 0 & l_{n-1} \\ 0 & \cos(a_{n-1}) & -\sin(a_{n-1}) & 0 \\ 0 & \sin(a_{n-1}) & \cos(a_{n-1}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \cos(\theta_n) & -\sin(\theta_n) & 0 & 0 \\ \sin(\theta_n) & \cos(\theta_n) & 0 & 0 \\ 0 & 0 & 1 & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} \cos(\theta_n) & -\sin(\theta_n) & 0 & l_{n-1} \\ \sin(\theta_n)\cos(a_{n-1}) & \cos(\theta_n)\cos(a_{n-1}) & -\sin(a_{n-1}) & -\sin(a_{n-1})d_n \\ \sin(\theta_n)\sin(a_{n-1}) & \cos(\theta_n)\sin(a_{n-1}) & \cos(a_{n-1}) & -\cos(a_{n-1})d_n \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3.2.2 Example 2

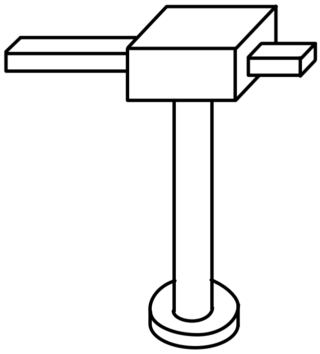

Figure 3.14 RP Robot

I have two joints and three links (0=Robot base, 1=Torso, 2=Upper arm):

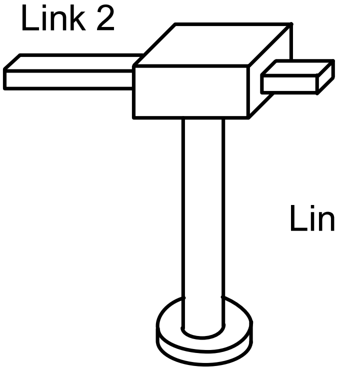

Figure 3.15 Link assignment

Since I have 3 links I have to define 3 reference frames {0}, {1}, {2}.

- So, joint 1 connects link 0 with link 1 or {0} with {1}. This implies that I have to create a matrix ${}^{0}\mathbf{T}_1$. Also since this joint is revolute I will have a joint variable θ.
- But, remember that ${}^{n-1}\mathbf{T}_n = \mathbf{Rot}(x, a_{n-1})\mathbf{Trans}(x, l_{n-1})\mathbf{Trans}(z, d_n)\mathbf{Rot}(z, \theta_n)$.
- In that case n-1=0 and n=1 the joint variable θ_n is θ_1.
- Similarly, joint 2 connects link 1 with link 2. So I have to create a matrix ${}^{1}\mathbf{T}_2$.
- ${}^{n-1}\mathbf{T}_n = \mathbf{Rot}(x, a_{n-1})\mathbf{Trans}(x, l_{n-1})\mathbf{Trans}(z, d_n)\mathbf{Rot}(z, \theta_n)$.
- In that case n-1=1 and n=2 the joint variable d_n is d_2.

Create the joint axes:
These two axes will give me the direction of z_1 and z_2 axes.

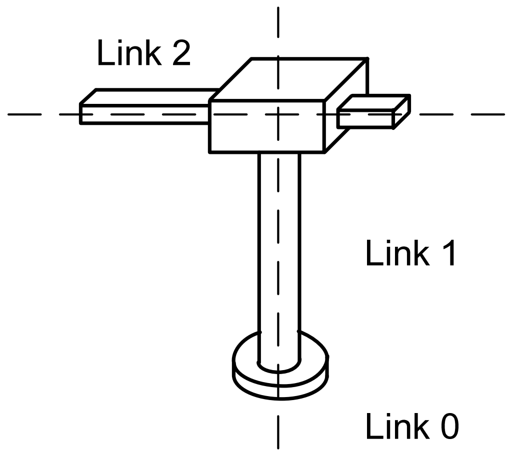

Figure 3.16 Link axes

Let's focus on {1} (since there is not link -1):

- The origin of {1} is located at the intersection of the two joint axes.
- Since z_1 intersects z_2 then the x_1 axis will be perpendicular to the plane created by z_1, z_2.

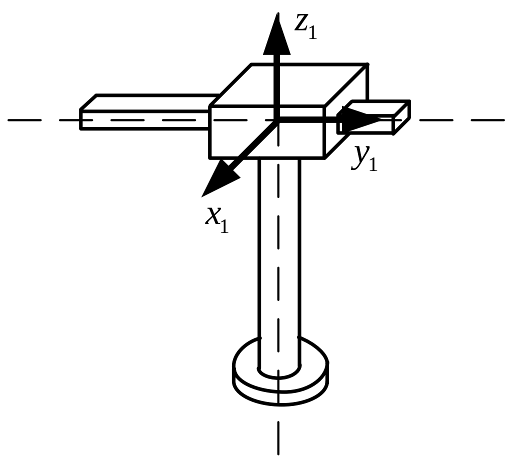

Figure 3.17 First ref frame

Since the joint 1 is revolute:

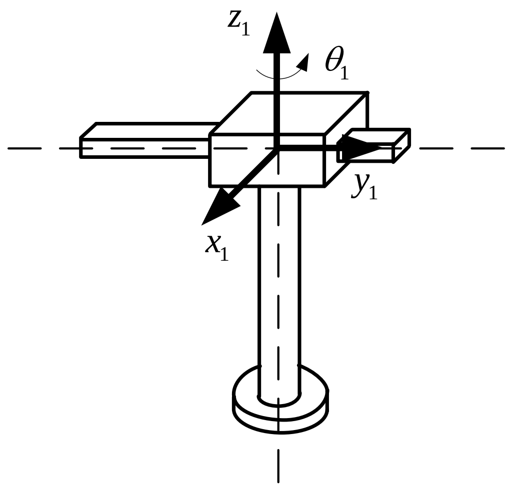

Figure 3.18 Joint variable

First reference frame, i.e. {0}: The frames {1} and {0} must coincide when the joint variable is zero, hence when $\theta_1 = 0$, so:

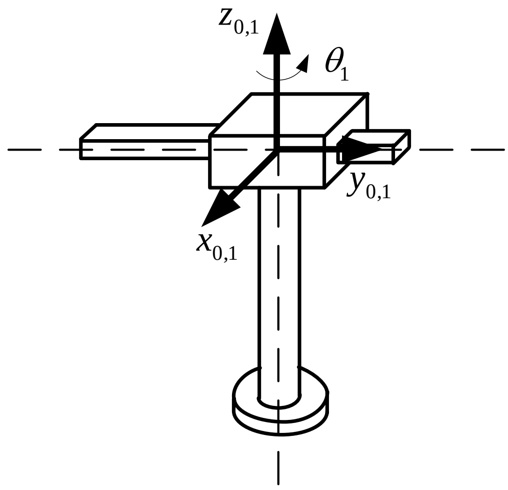

Figure 3.19 Robot base frame

Last reference frame:
This is the last reference frame {2} can be attached anywhere we want!!! So let's attach it at the end of link 2, so that it will also describe the end effector. Also According to the DH rules x_3 must be chosen so that θ_2 is zero, i.e. x_2 and x_3 are parallel. Hence I must rotate with respect to x_{n-1} for $+90^o$.

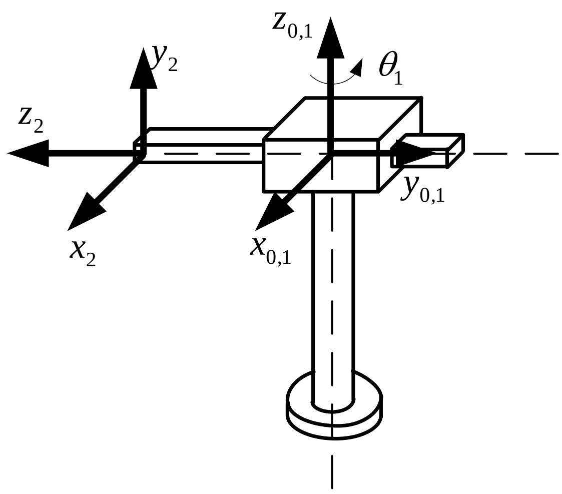

Figure 3.20 Final assignment

So the link table is:

Link, T	a_{n-1}	l_{n-1}	d_1	θ_n
1, ${}^0T_1$	$a_0 = 0$	$l_0 = 0$	$d_1 = 0$	θ_1
2, ${}^1T_2$	$a_1 = 90$	$l_1 = 0$	d_2	$\theta_2 = 0$

$$
\begin{aligned}
{}^0\mathbf{T}_1 &= \mathbf{Rot}(x,a_0)\mathbf{Trans}(x,l_0)\mathbf{Trans}(z,d_1)\mathbf{Rot}(z,\theta_1) = \mathbf{Rot}(z,\theta_1) \\
{}^1\mathbf{T}_2 &= \mathbf{Rot}(x,a_1)\mathbf{Trans}(x,l_1)\mathbf{Trans}(z,d_2)\mathbf{Rot}(z,\theta_2) = \mathbf{Rot}(x,90)\mathbf{Trans}(z,d_2) \\
{}^0\mathbf{T}_2 &= {}^0\mathbf{T}_1{}^1\mathbf{T}_2 = \mathbf{Rot}(z,\theta_1)\mathbf{Rot}(x,90)\mathbf{Trans}(z,d_2)
\end{aligned}
$$

3.2.3 Example 3

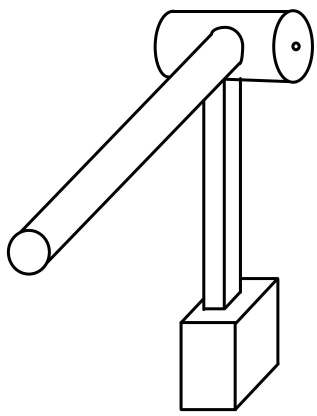

Figure 3.21

Joint 1 prismatic: So d_1
Joint 2 revolute: So θ_2

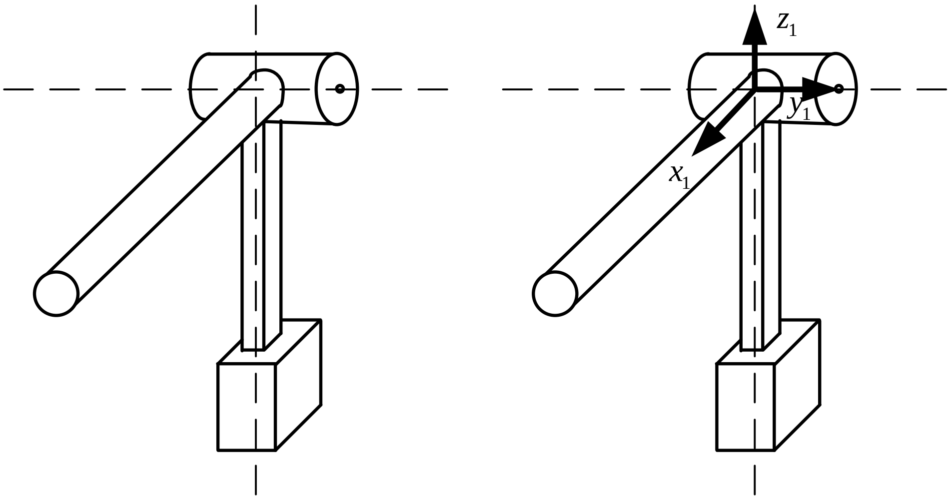

Figure 3.22

Error! Not a valid link.

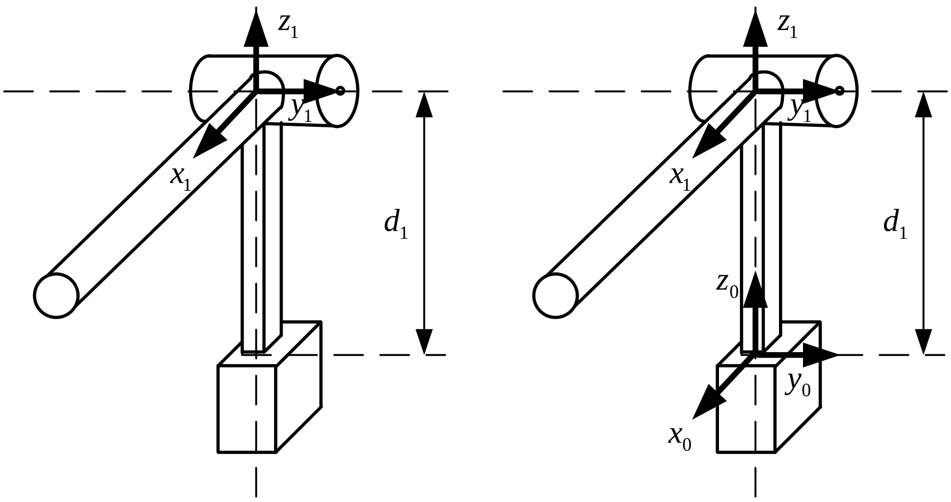

Figure 3.23

$x_2 // x_1$, z_2 must be on the joint axis 2 and $d_2 = 0$:

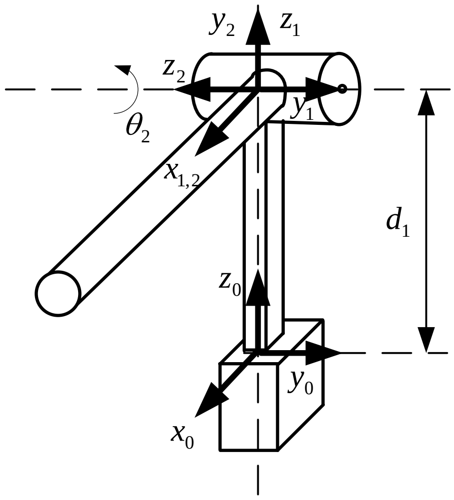

Figure 3.24

So the link table is:

Link, T	a_{n-1}	l_{n-1}	d_1	θ_n
1, ${}^0T_1$	$a_0 = 0$	$l_0 = 0$	d_1	$\theta_1 = 0$
2, ${}^1T_2$	$a_1 = 90$	$l_1 = 0$	$d_2 = 0$	θ_2

$$
\begin{aligned}
{}^0\mathbf{T}_1 &= \mathbf{Rot}(x, a_0)\mathbf{Trans}(x, l_0)\mathbf{Trans}(z, d_1)\mathbf{Rot}(z, \theta_1) = \mathbf{Trans}(z, d_1) \\
{}^1\mathbf{T}_2 &= \mathbf{Rot}(x, a_1)\mathbf{Trans}(x, l_1)\mathbf{Trans}(z, d_2)\mathbf{Rot}(z, \theta_2) = \mathbf{Rot}(x, 90)\mathbf{Rot}(z, \theta_2) \\
{}^0\mathbf{T}_2 &= {}^0\mathbf{T}_1 {}^1\mathbf{T}_2 = \mathbf{Trans}(z, d_1)\mathbf{Rot}(x, 90)\mathbf{Rot}(z, \theta_2)
\end{aligned}
$$

What should you do if you wanted to describe the end effector?

3.2.4 Example 4

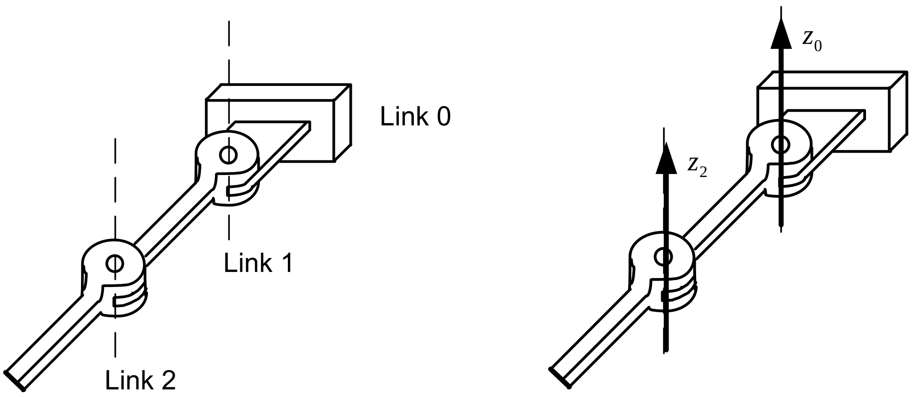

Figure 3.25

I arbitrary choose the origin of {1} and {0} as:

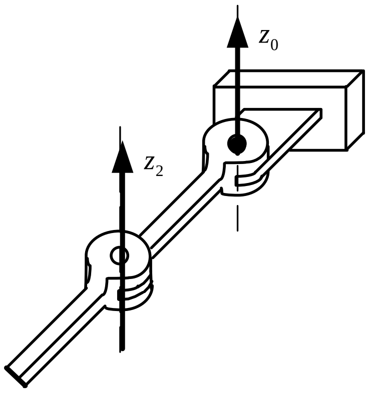

Figure 3.26

Hence the common normal is:

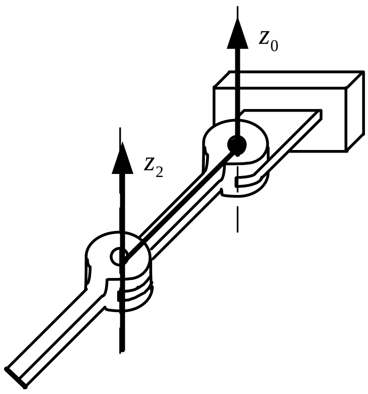

Figure 3.27

So:

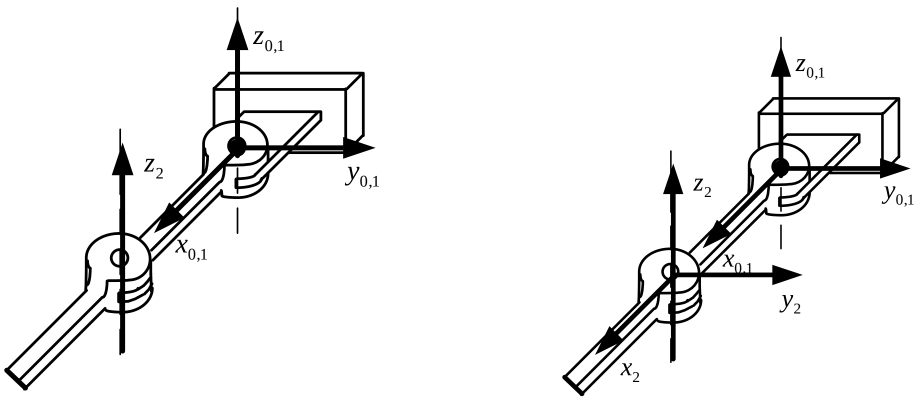

Figure 3.28

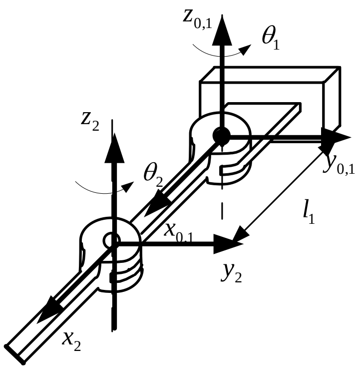

Figure 3.29

So the link table is:

Link, T	a_{n-1}	l_{n-1}	d_1	θ_n
1, ${}^0T_1$	$a_0 = 0$	$l_0 = 0$	$d_1 = 0$	θ_1
2, ${}^1T_2$	$a_1 = 0$	l_1	$d_2 = 0$	θ_2

$${}^0\mathbf{T}_1 = \mathbf{Rot}(x, a_0)\mathbf{Trans}(x, l_0)\mathbf{Trans}(z, d_1)\mathbf{Rot}(z, \theta_1) = \mathbf{Rot}(z, \theta_1)$$
$${}^1\mathbf{T}_2 = \mathbf{Rot}(x, a_1)\mathbf{Trans}(x, l_1)\mathbf{Trans}(z, d_2)\mathbf{Rot}(z, \theta_2) = \mathbf{Trans}(x, l_1)\mathbf{Rot}(z, \theta_2)$$
$${}^0\mathbf{T}_2 = {}^0\mathbf{T}_1{}^1\mathbf{T}_2 = \mathbf{Rot}(z, \theta_1)\mathbf{Trans}(x, l_1)\mathbf{Rot}(z, \theta_2)$$

3.3 Arm Orientation

The final step in the solution of the forward kinematics is to find the orientation of the end effector. The orientation matrix is given by:

$$^{R}\mathbf{T}_N = \begin{bmatrix} x_x & y_x & z_x & p_x \\ x_y & y_y & z_y & p_y \\ x_z & y_z & z_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_z \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} RPY(\phi,\theta,\psi) =$$

$$= \begin{bmatrix} C(\phi)C(\theta) & C(\phi)S(\theta)S(\psi) - S(\phi)C(\psi) & C(\phi)S(\theta)C(\psi) + S(\phi)S(\psi) & p_x \\ S(\phi)C(\theta) & S(\phi)S(\theta)S(\psi) + C(\phi)C(\psi) & S(\phi)S(\theta)c(\psi) - C(\phi)s(\psi) & p_y \\ -S(\theta) & C(\theta)S(\psi) & C(\theta)C(\psi) & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

By equating the two matrices it can be found that:

$$\theta = -\sin^{-1}(x_z)$$

$$\psi = \sin^{-1}\left(\frac{y_z}{\cos(\theta)}\right)$$

$$\phi = \sin^{-1}\left(\frac{x_y}{\cos(\theta)}\right)$$

These three equations describe the orientation of the end effector with respect to the base.

3.3.1 Example 1

For the robot manipulator of Fig.3.30 find the general transformation matrix. Assuming that $l_1=l_2=0.5$m and $\theta_1=60^o$, $\theta_2=-35^o$ find the orientation of the end effector:

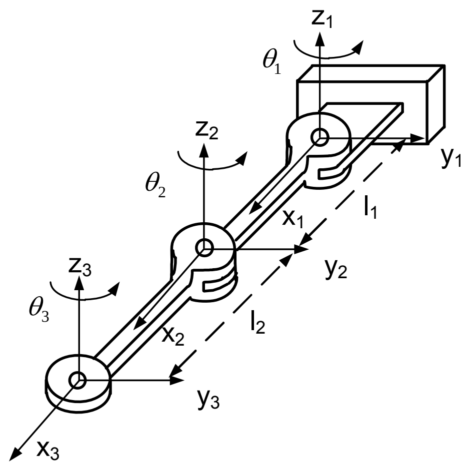

Figure 3.30

The transformation matrix is $^{0}T_3 = \begin{bmatrix} C_{12} & -S_{12} & 0 & l_2C_{12}+l_1C_1 \\ S_{12} & C_{12} & 0 & l_2S_{12}+l_1S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ where:

$C_{12} = \cos(\theta_1 + \theta_2)$ and $S_{12} = \sin(\theta_1 + \theta_2)$

For $\theta_1 = 60$, $\theta_2 = -35$, l_1=0.5, l_2=0.5 the transformation matrix is:

$^{0}T_3 = \begin{bmatrix} 0.9063 & -0.4226 & 0 & 0.7032 \\ 0.4226 & 0.9063 & 0 & 0.6443 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ and hence the rotation angles are:

$$\left.\begin{array}{l} \theta = -\sin^{-1}(0) \\ \psi = \sin^{-1}\left(\dfrac{0}{\cos(\theta)}\right) \\ \phi = \sin^{-1}\left(\dfrac{0.4226}{\cos(\theta)}\right) \end{array}\right\} \Leftrightarrow \left\{\begin{array}{l} \theta = 0 \\ \psi = 0 \\ \phi = \sin^{-1}(0.4226) = 0.4364\, rad = 25^{o} \end{array}\right\}$$

. This can be graphically proved:

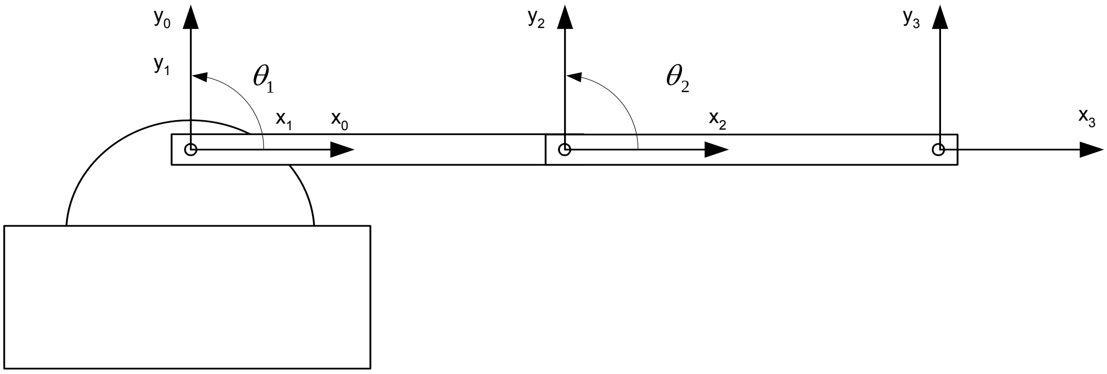

Figure 3.31

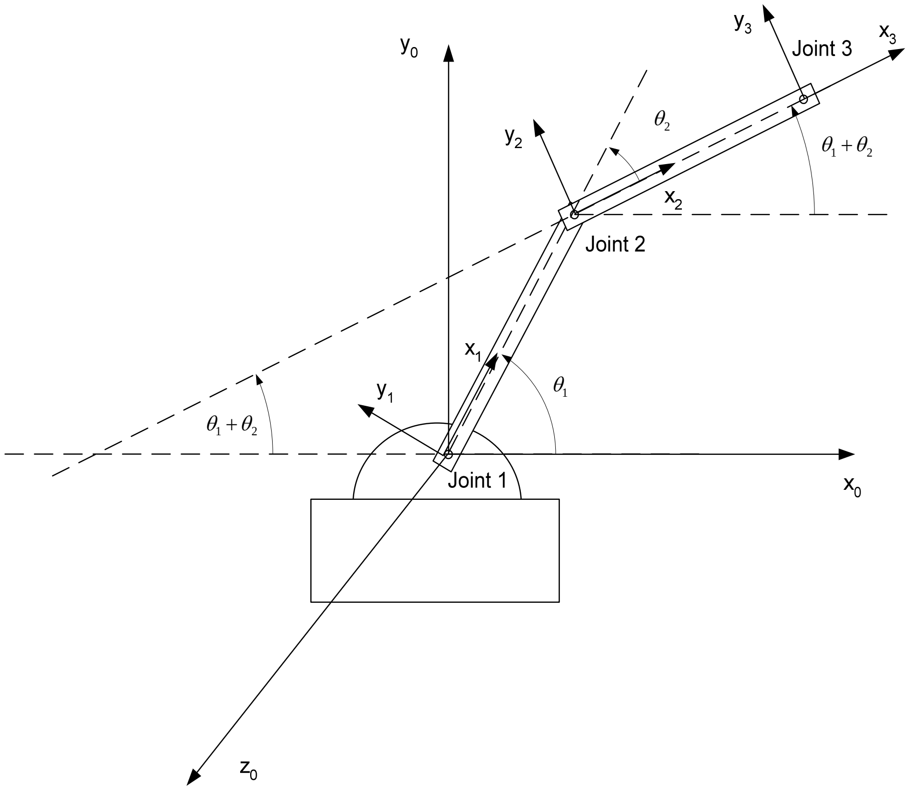

Figure 3.32

3.3.2 Example 2

For the robot manipulator of Fig.3.33 find the general transformation matrix. Assuming that $d_1=d_2=0.5$m and $\theta_1=60^o$, find the orientation of the end effector:

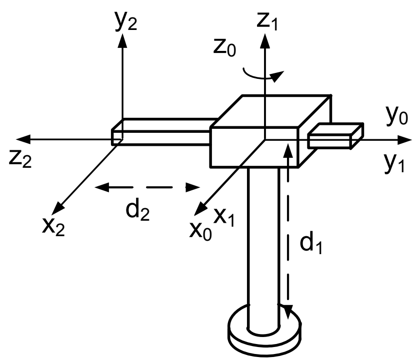

Figure 3.33

The transformation matrix is ${}^{0}T_2 = \begin{bmatrix} C_1 & 0 & S_1 & S_1 d_2 \\ S_1 & 0 & -C_1 & -C_1 d_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$, where: $C_1 = \cos(\theta_1)$ and $S_1 = \sin(\theta_1)$ or ${}^{0}T_2 = \begin{bmatrix} 0.5 & 0 & 0.866 & 0.433 \\ 0.86 & 0 & -0.5 & -0.25 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \Rightarrow \begin{Bmatrix} \theta = 0 \\ \psi = 0 \\ \phi = 60^o \end{Bmatrix}$.

3.3.3 Example 3

For the robot manipulator of Fig.1.96 find the general transformation matrix. Assuming that l_2=0.5m and θ_2=60^o, find the orientation of the end effector:

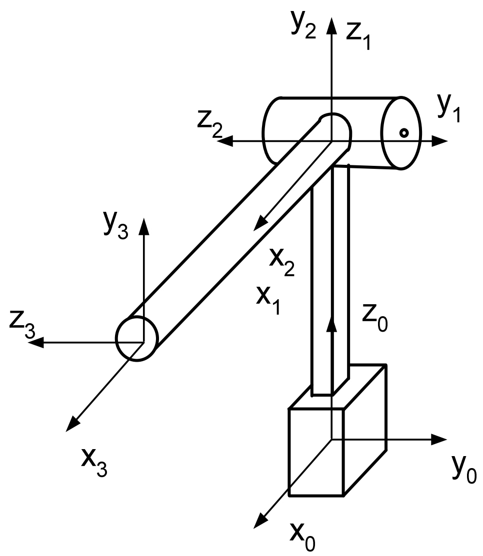

Figure 3.34

The transformation matrix is: ${}^{0}T_3 = \begin{bmatrix} C_2 & -S_2 & 0 & l_2 C_2 \\ 0 & 0 & -1 & 0 \\ S_2 & C_2 & 0 & d_1 + l_2 S_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ and hence:

$$ {}^{0}T_3 = \begin{bmatrix} 0.5 & -0.866 & 0 & 0.25 \\ 0 & 0 & -1 & 0 \\ 0.866 & 0.5 & 0 & 0.933 \\ 0 & 0 & 0 & 1 \end{bmatrix} \Rightarrow \begin{Bmatrix} \theta = -60^o \\ \psi = 90^o \\ \phi = 0 \end{Bmatrix}. $$

3.3.4 Robust Orientation

In the previous paragraph the end effector orientation was calculated by using Roll – Pitch – Yaw angles. The algorithm that was found is based on the use of $\cos(\theta)$ and $\sin(\theta)$. The angle ψ was calculated as $\psi = \sin^{-1}\left(\frac{y_z}{\cos(\theta)}\right)$. But this algorithm, even though it is very efficient, is not robust when the angle φ is +/-90°. To overcome this problem a new function has to be used to calculate the angles. This method uses the function atan2:

```
function phi=atan2(y,x)

% phi belongs to [-pi, pi]

if x==0 & y>0
    phi=pi/2;
elseif x==0 & y<0
    phi=-pi/2;
end

if y==0 & x>0
    phi=0;
elseif y==0 & x<0
    phi=-pi;
end

if x==0 & y==0
    phi=0;
end

if y~=0 & x~=0
    phi=atan(y/x);
end
```

To use the atan2 function; the transformation matrix must be further manipulated to:

$$ {}^{R}\mathbf{T}_N = \begin{bmatrix} x_x & y_x & z_x & p_x \\ x_y & y_y & z_y & p_y \\ x_z & y_z & z_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_z \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} RPY(\phi,\theta,\psi) \Leftrightarrow $$

$$ {}^{R}\mathbf{T}_N = Transl(p_x, p_y, p_z) Rot(z,\phi) Rot(y,\theta) Rot(x,\psi) \Leftrightarrow $$

$$ (Rot(z,\phi))^{-1} (Transl(p_x, p_y, p_z))^{-1}\, {}^{R}\mathbf{T}_N = Rot(y,\theta) Rot(x,\psi) \Leftrightarrow $$

$$ \begin{bmatrix} c\phi & s\phi & 0 & 0 \\ -s\phi & c\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -p_x \\ 0 & 1 & 0 & -p_y \\ 0 & 0 & 1 & -p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_x & y_x & z_x & p_x \\ x_y & y_y & z_y & p_y \\ x_z & y_z & z_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\theta & 0 & s\theta & 0 \\ 0 & 1 & 0 & 0 \\ -s\theta & 0 & c\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\psi & -s\psi & 0 \\ 0 & s\psi & c\psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \Leftrightarrow $$

$$ \begin{bmatrix} x_x c\phi + x_y s\phi & y_x c\phi + y_y s\phi & z_x c\phi + z_y s\phi & 0 \\ x_y c\phi - x_x s\phi & y_y c\phi - y_x s\phi & z_y c\phi + z_x s\phi & 0 \\ x_z & y_z & z_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\theta & s\theta\, s\psi & s\theta\, c\psi & 0 \\ 0 & c\psi & -s\psi & 0 \\ -s\theta & c\theta\, s\psi & c\theta\, c\psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

By equating the two matrices:

$$ x_y c\phi - x_x s\phi = 0 \Rightarrow \phi = \operatorname{atan2}(\mathrm{x}_y, \mathrm{x}_x) $$

$$ \left.\begin{aligned} x_x c\phi + x_y s\phi &= c\theta \\ x_z &= -s\theta \end{aligned}\right\} \Rightarrow \theta = \operatorname{atan2}(-x_z, x_x c\phi + x_y s\phi) $$

$$ \left.\begin{aligned} y_z &= c\theta\, s\psi \\ z_z &= c\theta\, c\psi \end{aligned}\right\} \Rightarrow \psi = \operatorname{atan2}(y_z, z_z) $$

3.4 Inverse kinematics

Until now the question that had to be answered was to find the location of the end effector by knowing the joint and link variables. Also in paragraphs 1.10.1 & 1.10.3 the overall orientation was calculated by knowing the joint and link variables by using Roll – Pitch –Yaw angles. The inverse problem is to know the general orientation and position of the end effector and to try to calculate the appropriate joint variables.

Generally speaking this problem is much more difficult and complicated than the forward one. Usually it requires the solution of 12 nonlinear equations. The solution of these nonlinear equations is called the **arm solution** and every robot has its own solution, which may not be unique. This means that the desired orientation and position may be achieved with more than one configuration. This phenomenon is called **redundancy**.

On the other hand, usually, there are some restrictions that may only allow one solution to be considered. For example the angles of revolute joints cannot be more than 360^o or maybe there is an obstacle that makes a specific solution impossible in practice. Hence a solution is not always possible and there are orientations and positions that a robot may not be able to achieve.

Solvability attempts to formally define if a solution is possible, and requires knowledge of the robot workspace which is roughly the space or locus that the robot end effector may reach.

The third important component of the inverse kinematics problem, going with solvability and redundancy, is the **method of solution**. There are mainly two methods to solve any equation, a closed form and a numerical (or iterative) method. The later is not preferred since it requires much processing power and, depending on the numerical method, may not find all the possible configurations. Finally the closed form solutions may be classified as geometric and algebraic solutions.

The biggest problem of finding the solution of a manipulator arm is the fact that every robot has its own configuration (link lengths and angles) and hence there is no a clear method of how to achieve an acceptable closed form solution.

The steps that have to be followed can be arbitrary and with no clear reasoning. McKerrow in “Introduction to Robotics”, chapter 4 tries to give a general guideline but this is not very clear.

The only thing that can be said when the solution is investigated is to use the 4th column of the transformation matrix, which is the translation. The elements that describe the orientation are usually coupled and they require complex solutions.