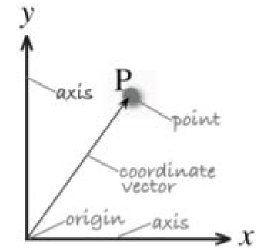
KINEMATIC MODELLING WORKSHOP

The next two sections refer to Peter Corke's Textbook [1]. His content is also available as videos in the Robot Academy website: https://robotacademy.net.au/

Understanding 3D space using Homogeneous Transforms

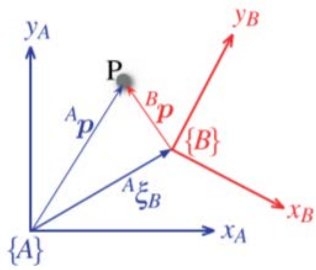
In robotics, we tend represent the world as a 3D Cartesian space where points are coordinate vectors with components on each axis:



$$ {}^{A}P = [T_x \quad T_y \quad T_z] $$

Each point in space is measured from a Cartesian coordinate frame and has a letter on top that tells you what frame it's measured from.

A similar set-up is done when you have multiple coordinate frames. A transform from one frame to another is represented as a pose (position and orientation). The letter at the top left of it tells you the frame it's measuring from and the bottom right letter tells you where the transform goes to:



$$ {}^{A}\xi_{B} = transform\ from\ frame\ A\ to\ frame\ B $$

This is useful when you want to find where one measured point is to another frame e.g.:

$$ {}^{A}P = {}^{A}\xi_{B}\,{}^{B}P $$

This means that the transform from frame A to B of the point measured in frame B becomes the same point measured in frame A. This transform can be cascaded to transform from many more coordinate frames:

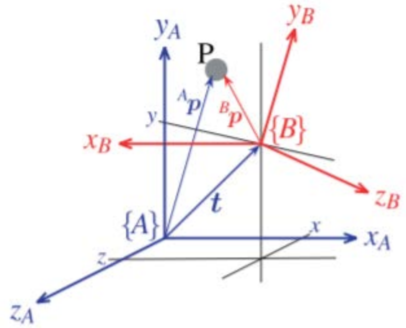
$$ {}^{A}P = {}^{A}\xi_{B}\,{}^{B}\xi_{C}\,{}^{C}P $$

For the robot arm project we'll use this theory to find where the gripper is from the base:

$$ {}^{Base}P_{Gripper} = {}^{Base}\xi_{Joint1}\,{}^{Joint1}\xi_{Joint2}\,{}^{Joint2}\xi_{Joint3}\,{}^{Joint3}P_{Gripper} $$

We can represent the transform as a matrix, like so:

$$ {}^{A}\xi_{B} \rightarrow \quad {}^{A}T_{B} = \begin{bmatrix} \cdots & \cdots & \cdots & T_x \\ \cdots & {}^{A}R_{B} & \cdots & T_y \\ \cdots & \cdots & \cdots & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} $$



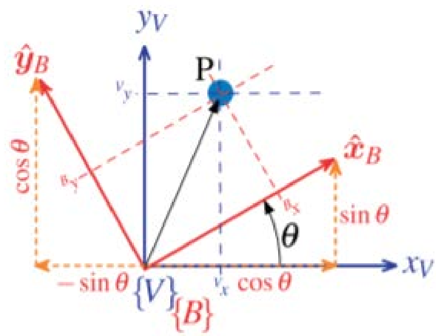
The matrix is 4 by 4 with the top left 3 by 3 being a 3D rotation Matrix and the forth column having the translational components

$t = [T_x \quad T_y \quad T_z]$ of the transform in each axis. The bottom row is what makes it possible to use the transforms in this way:

$$^{Base}T_{Gripper} = {}^{Base}T_{Joint1} \times {}^{Joint1}T_{Joint2} \times {}^{Joint2}T_{Joint3} \times {}^{Joint3}T_{Gripper}$$

Where you can use matrix multiplication (dot product) to cascade the transforms.

To understand the rotation matrix think of it as an array of components of each axis. Note: every axis is a unit vector. Based on the figure to the right, you'll see that new axis is split into components measured on the old axis:



$$^{V}R_B \rightarrow \begin{array}{r c} * & \begin{array}{ccc} x_B & y_B & z_B \end{array} \\ \begin{array}{r} along\ x_V \\ along\ y_V \\ along\ z_V \end{array} & \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array}$$

The matrix means that for some x distance along x_B is measured as that times $\cos\theta$ along x_V

And that for some y distance along y_B is measured as that times $-\sin\theta$ along x_V

And so on for each 3 components on each of the 3 axes … note that the z-axis in the figure is unchanged as theta rotates the frame about the z-axis, so therefore the z-axes components 'resemble' the identity matrix. A rotation matrix with no rotation is an identity matrix. All rotation matrix components must range between -1 and 1 and the determinant is always 1 to be a valid rotation matrix. A point can be rotated using matrix multiplication like this:

$$R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix}$$

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

$$R_z(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$^{V}P = {}^{V}R_B {}^{B}P$$

$$^{V}P = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} ^{B}T_X \\ ^{B}T_Y \\ ^{B}T_Z \end{bmatrix}$$

Rotation in 3D space means that a transform can not only translate in the x, y and z directions but rotate about the x, y and z axes. 3D rotation is commonly described by Euler angles roll pitch yaw. Rotation about the x-axis is roll, rotation about the y-axis is pitch and rotation about the z-axis is yaw.

$$R = R_X(\theta_{roll}) \times R_Y(\theta_{pitch}) \times R_Z(\theta_{yaw})$$

Putting this together a transformation matrix is used like this:

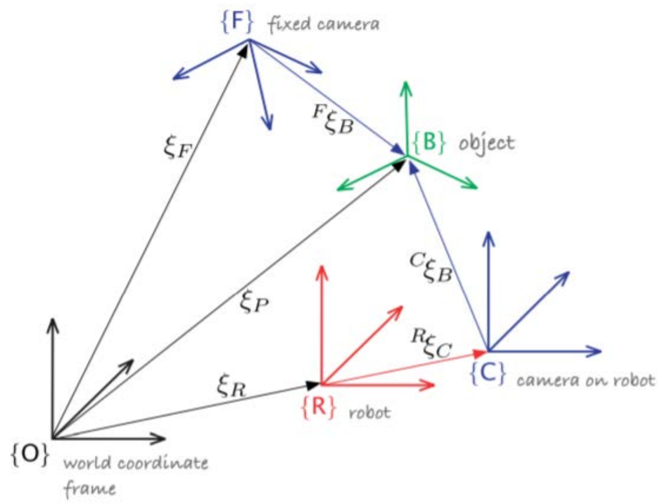
Let's say that $^{A}R_B$ is z-axis rotation: $R_Z(\theta)$ and translation vector is $t = [5, 10, -5]$

$$^{A}P = {}^{A}T_{B}\,{}^{B}P$$

$$^{A}P = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 5 \\ \sin\theta & \cos\theta & 0 & 10 \\ 0 & 0 & 1 & -5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{B}T_X \\ {}^{B}T_Y \\ {}^{B}T_Z \\ 1 \end{bmatrix}$$

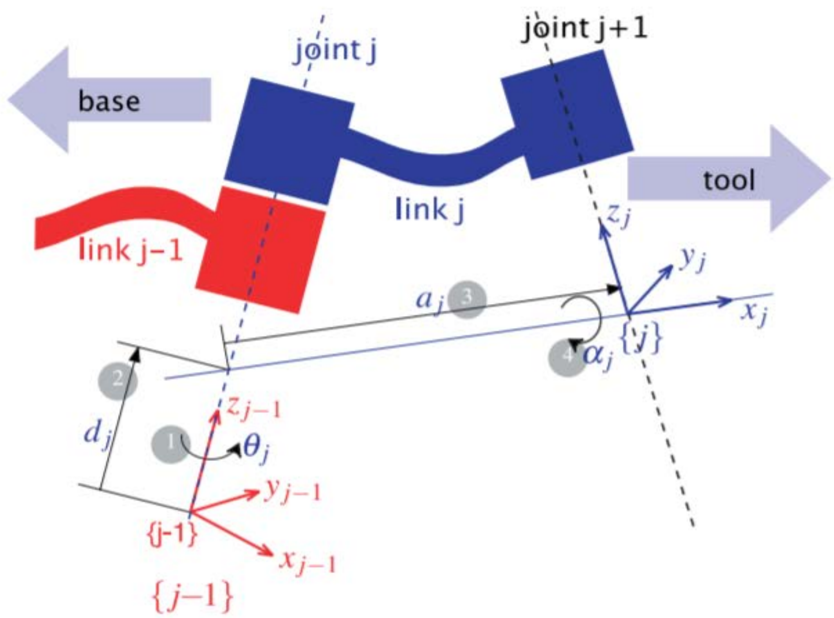
Which transforms the point ^{B}P to the same point but measured from the A frame

So overall, in 3D space, any object with a coordinate frame can be represented as a transform matrix from the world coordinate frame to it. The matrix consists of a 3 by 3 rotation matrix consisting of 3 parameters and a 3 by 1 translation vector consisting of 3 parameters. A transform is translation and rotation of an object in space based on 6 parameters: x, y, z, roll, pitch and yaw



DH Parameters in the Transform Matrix

In the previous section, the transform from one coordinate frame to another in 3D space is described by 6 parameters. Denavit and Hartenburg said that for 3D transforms between revolute joints (ones that rotate only in one axis) can be simplified to only need 4 parameters.



If you place the coordinate frames in a certain way (mainly placing the z axis as the axis of rotation and the x-axis as an axis to measure between axes), you can have the transform represented by only rotations and translations about the x and z axes:

$$^{j-1}\mathbf{A}_j(\theta_j, d_j, a_j, \alpha_j) = \mathbf{T}_{Rz}(\theta_j)\mathbf{T}_z(d_j)\mathbf{T}_x(a_j)\mathbf{T}_{Rx}(\alpha_j)$$

So from frame j-1 to frame j, the transform is described by parameters θ, d, a, α

This yields a standard transformation matrix that we can plug and use for every joint:

$$^{j-1}\mathbf{A}_j = \begin{pmatrix} \cos\theta_j & -\sin\theta_j\cos\alpha_j & \sin\theta_j\sin\alpha_j & a_j\cos\theta_j \\ \sin\theta_j & \cos\theta_j\cos\alpha_j & -\cos\theta_j\sin\alpha_j & a_j\sin\theta_i \\ 0 & \sin\alpha_j & \cos\alpha_j & d_j \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

So every time you want to compute the transformation matrix among the joints, you need to find 4 parameters:

Joint angle	θ_j	the angle between the x_{j-1} and x_j axes about the z_{j-1} axis	revolute joint variable
Link offset	d_j	the distance from the origin of frame $j-1$ to the x_j axis along the z_{j-1} axis	prismatic joint variable
Link length	a_j	the distance between the z_{j-1} and z_j axes along the x_j axis; for intersecting axes is parallel to $\hat{z}_{j-1} \times \hat{z}_j$	constant
Link twist	α_j	the angle from the z_{j-1} axis to the z_j axis about the x_j axis	constant

Solving the Forward Kinematics

Forward kinematics is the problem of finding out where the gripper is from the base of the robot. It can be represented as a function K of joint values to gripper pose:

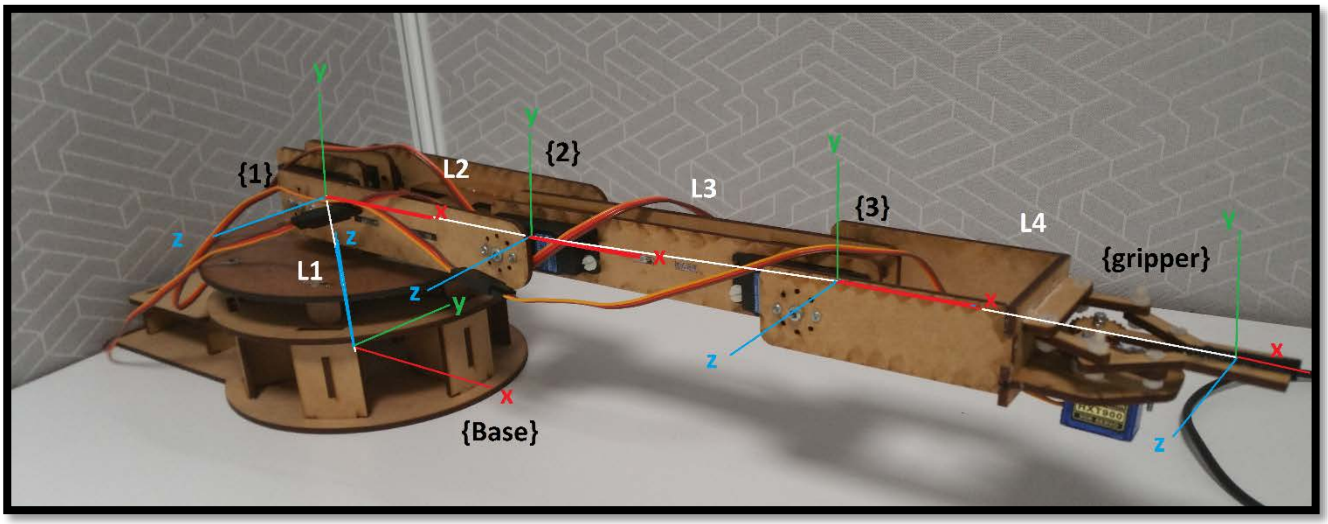
$$\boldsymbol{x} = K(\boldsymbol{q})$$

Where **x** is the tool point pose vector and **q** is the joint values vector of the robot arm

Finding the pose of the gripper from the base can be solved using the transformation matrices learnt before:

$$^{Base}T_{Gripper} = {}^{Base}T_{Joint1} \times {}^{Joint1}T_{Joint2} \times {}^{Joint2}T_{Joint3} \times {}^{Joint3}T_{Gripper}$$

Using the transform matrix ${}^{Base}T_{Gripper}$ we can extract the overall translation. The next figure shows how the coordinate frames have been set to find the 4 DH parameters for each joint:



Note we will need the arm lengths:

They are: $L1 = 100mm, L2 = 120mm, L3 = 140mm, L4 = 140mm$

All translation is measured in mm

Each joint does a rotation in radians.

The variable angle produced by each joint is

$$q_1, q_2, q_3, q_4$$

Computing the DH parameters yields:

DH Parameters	θ	d	a	α
${}^{Base}T_{Joint1}$				
${}^{Joint1}T_{Joint2}$				
${}^{Joint2}T_{Joint3}$				
${}^{Joint3}T_{Gripper}$				

We find the x, y, z coordinates by extracting the translation vector out of ${}^{Base}T_{Gripper}$

The other value that is of interest to know is the pitch angle of the gripper. That value in intuitively going to be the sum of the joints:

$$grasp\ angle = \theta_P = q_2 + q_3 + q_4$$

As we want to try controlling that angle and the position of the gripper