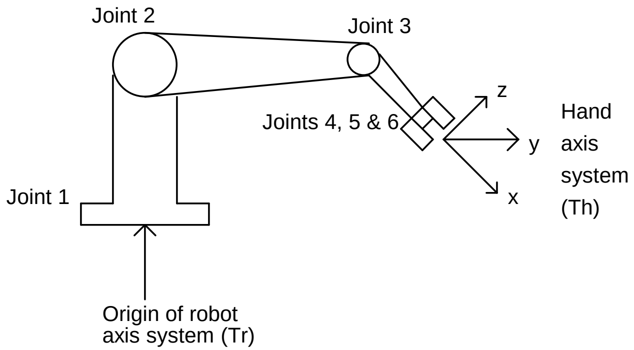
Robot Kinematics



- We know that a set of “joint angles” can be used to locate and orientate the hand in 3-D space
- We know that the joint angles can be combined into a 4x4 homogeneous transform which relates the location and orientation of the hand to the robot’s origin

The Basic Kinematic Equation

If *Tr* represents the axis system of the robot with respect to some global co-ordinate system then *Th*, the axis system of the hand with respect to that global system, will be given by -

$$Th = Tr.\ A1\ .\ A2\ .\ A3\ .\ A4\ .\ A5\ .\ A6$$

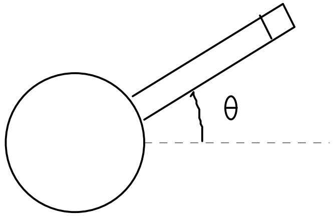
where

A1 is the transform representing a rotation about joint 1

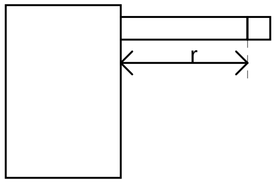
Etc.

NB The order of the transformations

A Simple 2 Joint Manipulator



Plan View



Front View

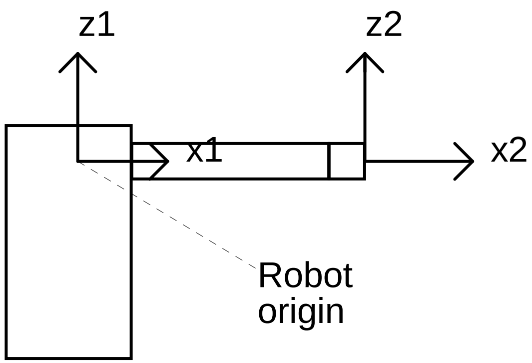
- We shall consider a very simple manipulator with just 2 joints
  - One joint rotates about the robot's trunk (theta)
  - One joint slides the arm radially in and out (r)
- This manipulator has just 2 degrees of freedom
  - It can only move in the horizontal plane and has no wrist articulation

Relating Co-ordinates to Joint Angles

- We shall derive equations for the x and y co-ordinates of the hand (z cannot be varied of course) in terms of the joint angles, theta and r
- We could use a geometric approach but this would let us down with more complicated manipulators
- We shall use a matrix algebra approach which should extrapolate nicely to manipulators with more joints

Co-ordinate Frames

- We start by associating a “co-ordinate frame” (axis system) with each joint
  - Frame 1 is placed on Joint 1 with its z-axis pointing up and its x-axis pointing out along the arm
  - Frame 2 is placed with its origin on the end of the arm and parallel to Frame 1



Joint Angle Transforms

- The matrix, *A1*, which describes the angle of the arm about the trunk is a z rotation
- The matrix, *A2*, which describes the distance along the x-axis from Joint 1 to the end of the arm is an x translation

$$A1 = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad A2 = \begin{bmatrix} 1 & 0 & 0 & r \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A1.A2 = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & r\cos\theta \\ \sin\theta & \cos\theta & 0 & r\sin\theta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The 6 Joint Arm

- We limit ourselves to arms whose joints are all revolute from now on
- We define co-ordinate frames for each joint
- We ensure that the A matrices all have the same general form
  - A rotation about z
  - A shift along x
  - A rotation about x
- Matrix Ai and its predecessors,

  A(i-1), A(i-2), etc. will correctly locate and orientate joint i+1 ready for its matrix, A(i+1), to carry out the appropriate rotation and displacement for the link which follows

Constructing the A Matrices

$$A_n = \begin{bmatrix} Cos\theta_n & -Sin\theta_n & 0 & 0 \\ Sin\theta_n & Cos\theta_n & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} 1 & 0 & 0 & d_n \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & Cos\alpha_n & -Sin\alpha_n & 0 \\ 0 & Sin\alpha_n & Cos\alpha_n & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\therefore A_n = \begin{bmatrix} Cos\theta_n & -Sin\theta_n Cos\alpha_n & Sin\theta_n Sin\alpha_n & d_n Cos\theta_n \\ Sin\theta_n & Cos\theta_n Cos\alpha_n & -Cos\theta_n Sin\alpha_n & d_n Sin\theta_n \\ 0 & Sin\alpha_n & Cos\alpha_n & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

For the 6 Joint Arm we have:

Joint 1: $\alpha_1 = 90$, $d_1 = 0$
Joint 2: $\alpha_2 = 0$, $d_2 \neq 0$
Joint 3: $\alpha_3 = 0$, $d_3 \neq 0$
Joint 4: $\alpha_4 = -90$, $d_4 \neq 0$
Joint 5: $\alpha_5 = 90$, $d_5 = 0$
Joint 6: $\alpha_6 = 0$, $d_6 = 0$

Complete 6 Joint Transform

$$T_h = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where

normal vectors

$$n_x = c_1(c_{234}c_5c_6 - s_{234}s_6) - s_1s_5c_6$$
$$n_y = s_1(c_{234}c_5c_6 - s_{234}s_6) - c_1s_5c_6$$
$$n_z = s_{234}c_5c_6 + c_{234}s_6$$

orientatio n vectors

$$o_x = -c_1(c_{234}c_5s_6 + s_{234}c_6) + s_1s_5s_6$$
$$o_y = -s_1(c_{234}c_5s_6 + s_{234}c_6) - c_1s_5s_6$$
$$o_z = -s_{234}c_5s_6 + c_{234}c_6$$

approach vectors

$$a_x = c_1c_{234}s_5 + s_1c_5$$
$$a_y = s_1c_{234}s_5 - c_1c_5$$
$$a_z = s_{234}s_5$$

position vectors

$$p_x = c_1(c_{234}d_4 + c_{23}d_3 + c_2d_2)$$
$$p_y = s_1(c_{234}d_4 + c_{23}d_3 + c_2d_2)$$
$$p_z = s_{234}d_4 + s_{23}d_3 + s_2d_2$$

and

$$c_i = \cos\theta_i \qquad c_{ij} = \cos(\theta_i + \theta_j)$$
$$s_i = \sin\theta_i \qquad s_{ij} = \sin(\theta_i + \theta_j)$$

Example I
The first 3 Joints

- Matrices *A1*, *A2* and *A3* will position the start of the wrist
- We shall relate the (x, y, z) co-ordinates of the wrist to the first 3 joint angles - using all 6 would be very tedious!
- The transform for the end of the arm (I.e. wrist location) is

  Ta = Tr . A1 . A2 . A3
- We shall assume *Tr* is the unit matrix for further simplicity
- The equations for x, y and z are derived as follows ...

Example I Continued

$$T_a = \begin{bmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c_2 & -s_2 & 0 & d_2c_2 \\ s_2 & c_2 & 0 & d_2s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c_3 & -s_3 & 0 & d_3c_3 \\ s_3 & c_3 & 0 & d_3s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} c_1c_2c_3 - c_1s_2s_3 & -c_1c_2s_3 - c_1s_2c_3 & s_1 & d_3c_1c_2c_3 - d_3c_1s_2s_3 + d_2c_1c_2 \\ s_1c_2c_3 - s_1s_2s_3 & -s_1c_2s_3 - s_1s_2c_3 & -c_1 & d_3s_1c_2c_3 - d_3s_1s_2s_3 + d_2s_1c_2 \\ s_2c_3 + c_2s_3 & -s_2s_3 + c_2c_3 & 0 & d_3s_2c_3 + d_3c_2s_3 + d_2s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Equating terms between the above and T_h:

$$p_x = d_3c_1c_2c_3 - d_3c_1s_2s_3 + d_2c_1c_2$$

$$p_y = d_3s_1c_2c_3 - d_3s_1s_2s_3 + d_2s_1c_2$$

$$p_z = d_3s_2c_3 + d_3c_2s_3 + d_2s_2$$

Example II
Roll-Pitch-Yaw

- Consider the problem of orientating the wrist now (in terms of roll, pitch and yaw angles)
- Given the T_h matrix for the 6 joints we are only interested in the 3x3 part in the top left since this gives the rotations
- We can forget about *Px*, *Py*, *Pz* for the time being
- By convention an R-P-Y set is
  - a z rotation (*R*) followed by
  - a y rotation (*P*) followed by
  - an x rotation (*Y*)

Example II Continued

- So we first determine T_{rpy} by composing it from the 3 rotation matrices –

$$T_{rpy} \ = \ R \, . \, P \, . \, Y$$

$$T_{rpy} = Roll(R).Pitch(P).Yaw(Y)$$

$$= Rot(z,\phi_z).Rot(y,\phi_y).Rot(x,\phi_x)$$

$$= \begin{bmatrix} c\phi_z & -s\phi_z & 0 & 0 \\ s\phi_z & c\phi_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} c\phi_y & 0 & s\phi_y & 0 \\ 0 & 1 & 0 & 0 \\ -s\phi_y & 0 & c\phi_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\phi_x & -s\phi_x & 0 \\ 0 & s\phi_x & c\phi_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} c\phi_z c\phi_y & c\phi_z s\phi_y s\phi_x - s\phi_z c\phi_x & c\phi_z s\phi_y c\phi_x + s\phi_z s\phi_x & 0 \\ s\phi_z c\phi_y & s\phi_z s\phi_y s\phi_x + c\phi_z c\phi_x & s\phi_z s\phi_y c\phi_x - c\phi_z s\phi_x & 0 \\ -s\phi_y & c\phi_y s\phi_x & c\phi_y c\phi_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Example II Continued

- We can now equate terms between T_h and T_{rpy} -

$$\begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} c\phi_z c\phi_y & c\phi_z s\phi_y s\phi_x - s\phi_z c\phi_x & c\phi_z s\phi_y c\phi_x + s\phi_z s\phi_x & 0 \\ s\phi_z c\phi_y & s\phi_z s\phi_y s\phi_x + c\phi_z c\phi_x & s\phi_z s\phi_y c\phi_x - c\phi_z s\phi_x & 0 \\ -s\phi_y & c\phi_y s\phi_x & c\phi_y c\phi_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$-\sin\phi_y = n_z \qquad \therefore \quad \phi_y = \sin^{-1}(-n_z)$$

$$\cos\phi_y \sin\phi_x = o_z \qquad \therefore \quad \phi_x = \sin^{-1}\left(\frac{o_z}{\cos\phi_y}\right)$$

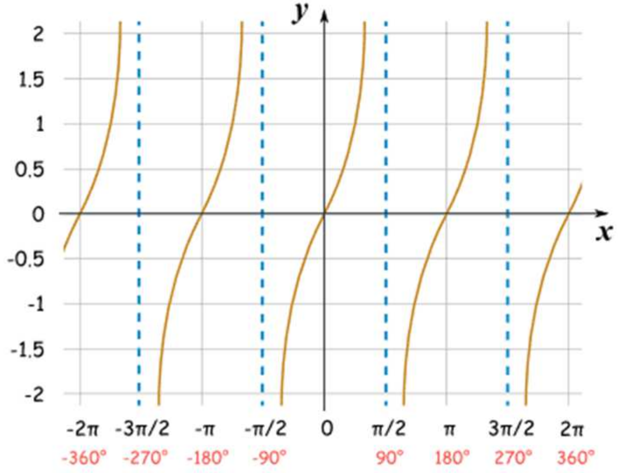
$$\sin\phi_z \cos\phi_y = n_y \qquad \therefore \quad \phi_z = \sin^{-1}\left(\frac{n_y}{\cos\phi_y}\right)$$

Example II Continued

- This is nice and easy but it isn’t good enough
  - Inverse sine (and cosine) are notoriously inaccurate
    - As we approach their turning points a small error in the input can create a large error in the output
  - They are also ambiguous
    - E.g. arcsin(1) might mean 90 or 270
  - So we need to avoid arcsin and arccos at all costs
  - Furthermore as the pitch angle tends to 90 our equations break down
    - Cos(90) = 0 so we cannot divide by it

arctan

- Inverse tangents are much more accurate and well-behaved than arcsin and arccos



ATAN2

- ATAN2 is a special arctan function available in all maths libraries which takes 2 arguments

ATAN2(x,y) = ARCTAN(x/y)

- ATAN2 keeps the numerator and denominator separate so their signs can be inspected
- The signs of x and y are used to determine the quadrant of the function so no ambiguity arises

R-P-Y in terms of ATAN2

$$T_h = T_{rpy} = Rot(z,\phi_z).Rot(y,\phi_y).Rot(x,\phi_x)$$

$$\therefore \quad Rot^{-1}(z,\phi_z).T_h = Rot^{-1}(z,\phi_z).T_{rpy} = Rot(y,\phi_y).Rot(x,\phi_x)$$

$$\therefore \begin{bmatrix} n_x c\phi_z + n_y s\phi_z & o_x c\phi_z + o_y s\phi_z & a_x c\phi_z + a_y s\phi_z & 0 \\ n_y c\phi_z - n_x s\phi_z & o_y c\phi_z - o_x s\phi_z & a_y c\phi_z - a_x s\phi_z & 0 \\ n_z & o_z & a_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} c\phi_y & s\phi_y s\phi_x & s\phi_y c\phi_x & 0 \\ 0 & c\phi_x & -s\phi_x & 0 \\ -s\phi_y & c\phi_y s\phi_x & c\phi_y c\phi_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Equating terms once again yields :

$$n_y c\phi_z - n_x s\phi_z = 0 \quad \therefore \quad 1 = \frac{n_x s\phi_z}{n_y c\phi_z} \quad \therefore \quad \frac{n_y}{n_x} = \tan(\phi_z)$$

$$\therefore \quad \phi_z = ATAN2(n_y, n_x)$$

Similarly

$$\phi_y = ATAN2(-n_z, n_x c\phi_z + n_y s\phi_z)$$

$$\phi_x = ATAN2(a_x s\phi_z - a_y c\phi_z, o_y c\phi_z - o_x s\phi_z)$$

Fixed Angles vs Euler Angles

- Fixed angle rotations are rotations of an object within a fixed unchanging reference frame
- Euler angle rotations are rotations of the reference frame itself. I.e. subsequent rotations are about a new axis system
  - This is the way we have been applying rotations so far but not with Euler angles

Euler Angles and Wrist Orientations

- If the wrist joints are not co-incident then it is natural to think of each joint rotating the reference frame in which the subsequent joint will do its work
- When they are coincident it can still be helpful to think about them in this ordered way
- Euler angles are therefore very popular for describing wrist orientations

Unique Angles

- For a given re-orientation in space there are an infinite number of combinations of rotations which can achieve it
- In particular the R-P-Y fixed set we determined previously was only unique when principal angles were chosen by ATAN2
- Euler angles suffer from the same problem unless we can constrain them in such a way that they have to be unique
- When we consider inverse kinematics this will be essential

Constraints

There are 2 types of constraint we can use

- Axes of rotation
  - 3 rotations are needed to achieve all possible orientations
  - Only 2 axes are needed
  - Typical Euler sets are
    - Z-X-Z (USA, Europe)
    - Z-Y-Z (UK)
- Magnitudes of rotations
  - Further constraints are needed on the angles

$$-\pi < \phi \leq \pi$$
$$0 \leq \theta \leq \pi$$
$$-\pi \leq \psi < \pi$$

General Z-Y-Z Euler Matrix

$$Rot(z,\phi_z).Rot(y,\theta_y).Rot(z,\psi_z)$$

$$\begin{bmatrix} \cos\phi_z & -\sin\phi_z & 0 & 0 \\ \sin\phi_z & \cos\phi_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} \cos\psi & -\sin\psi & 0 & 0 \\ \sin\psi & \cos\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} c\phi_z c\theta_y c\psi_z - s\phi_z s\psi_z & -c\phi_z c\theta_y s\psi_z - s\phi_z c\psi_z & c\phi_z s\theta_y & 0 \\ s\phi_z c\theta_y c\psi_z + c\phi_z s\psi_z & -s\phi_z c\theta_y s\psi_z + c\phi_z c\psi_z & s\phi_z s\theta_y & 0 \\ -s\theta_y c\psi_z & s\theta_y s\psi_z & c\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Constraints :} \quad \begin{aligned} -\pi < \ & \phi_z \le \pi \\ 0 \le \ & \theta_y \le \pi \\ -\pi \le \ & \psi_z < \pi \end{aligned}$$

Inverse Kinematics

$$\text{Joint Space} \xrightarrow{Th} \text{Cartesian Space}$$

$$\text{Cartesian Space} \xrightarrow{Th^{-1}} \text{Joint Space}$$

- So far we have only looked at obtaining Cartesian co-ordinates (x,y,z and orientation) given the joint angles
- As we have seen the most useful kind of control (e.g. for tool direction in lead-through programming and off-line programming) comes from specifying the Cartesian co-ordinates and then determining the joint angles
- This latter is *Inverse Kinematics*

Inverse Kinematics for the θ-r Manipulator (θ)

We equate the transform relating the TCP position and orientation to the robot's origin, T_h , to the 2 joint matrices, A_1 and A_2 :

$$T_h = A_1.A_2$$

$$where \quad A_1 = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad and \quad A_2 = \begin{bmatrix} 1 & 0 & 0 & r \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

So we equate,

$$\begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & r\cos\theta \\ \sin\theta & \cos\theta & 0 & r\sin\theta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

We note that here,

$$n_z = o_z = a_x = a_y = p_z = 0$$
$$a_z = 1$$

Equating terms yields,

$$r\cos\theta = p_x$$
$$r\sin\theta = p_y$$
$$\therefore \quad \theta = ATAN\ 2(p_y, p_x)$$

Inverse Kinematics for the θ-r Manipulator (r)

We cannot say,

$$r = \frac{p_x}{\cos\theta}$$

because this is not robust - it breaks down when $\theta = 90$.

To get a robust equation we pre - multiply T_h by A_1^{-1} ,

$$A_1^{-1}T_h = A_2$$

$$A_1^{-1}T_h = \begin{bmatrix} n_x c\theta + n_y s\theta & o_x c\theta + o_y s\theta & 0 & p_x c\theta + p_y s\theta \\ -n_x s\theta + n_y c\theta & -o_x s\theta + o_y c\theta & 0 & -p_x s\theta + p_y c\theta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and since,

$$A_2 = \begin{bmatrix} 1 & 0 & 0 & r \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\therefore \quad r = p_x\cos\theta + p_y\sin\theta$$

Inverse Kinematics for the 6 Joint Arm

$$\theta_1 = \tan^{-1}\left(\frac{p_y}{p_x}\right)$$

$$\left[\theta_{234} = \theta_2 + \theta_3 + \theta_4 = \tan^{-1}\left(\frac{a_z}{a_x c_1 + a_y s_1}\right)\right]$$

$$\theta_3 = \tan^{-1}\left(\frac{\sqrt{1-c_3^{\,2}}}{c_3}\right) \qquad where \quad c_3 = \frac{p_1^{\,2} + p_2^{\,2} - d_2^{\,2} - d_3^{\,2}}{2d_2 d_3}$$

$$and \quad \begin{aligned} p_1 &= p_x c_1 + p_y s_1 - d_4 c_{234} \\ p_2 &= p_z - d_4 s_{234} \end{aligned}$$

$$\theta_2 = \tan^{-1}\left(\frac{(d_3 c_3 + d_2) p_2 - d_3 s_3 p_1}{(d_3 c_3 + d_2) p_1 + d_3 s_3 p_2}\right)$$

$$\theta_4 = \theta_{234} - \theta_2 - \theta_3 = \tan^{-1}\left(\frac{a_z}{a_x c_1 + a_y s_1}\right) - \theta_2 - \theta_3$$

$$\theta_5 = \tan^{-1}\left(\frac{c_{234}(a_x c_1 + a_y s_1) + a_z s_{234}}{a_x s_1 - a_y c_1}\right)$$

$$\theta_6 = \tan^{-1}\left(\frac{s_5(o_x s_1 - o_y c_1) - c_5(c_{234}(o_x c_1 + o_y s_1) + o_z s_{234})}{o_z c_{234} - s_{234}(o_x c_1 + o_y s_1)}\right)$$

Problems with Inverse Kinematics

- There are 2 important problems to be aware of when attempting to derive the inverse kinematics -
- Degeneracies
  - There might be more than one set of joint angles which satisfies the TCP transform sought
  - More than one arrangement of the arm is possible
- Singularities
  - The equations might break down (through denominators of zero for instance) at certain TCP positions
  - These are also known as *dead points*

TCP Velocity Control

- How can we determine appropriate joint velocities so that we can ensure a specific TCP velocity (as needed in paint spraying, seam welding, etc.)?
- This is not trivial
- Consider the forward kinetics problem of determining a TCP velocity from a collection of joint velocities …

  The TCP velocity is dependent on a non-linear function of ALL of the joint velocities
- The inverse kinetics problem requires working back from a chosen TCP velocity to the joints

Differential Motions

- Velocity determination can be reduced to a problem of calculating the small *differential motions* which occur in a very small time interval
- In the forward kinetics problem we approximate the joint velocities by the differential displacements of the joints over a small time interval, dt
- We can do this if we choose a unit of time such that our small time interval is 1 -

dt = 1 => vel. = dx/dt = dx