Representing Position & Orientation & State

METR 4202: **Robotics** & Automation

Dr Surya Singh -- Lecture # 3 (Ekka Day)

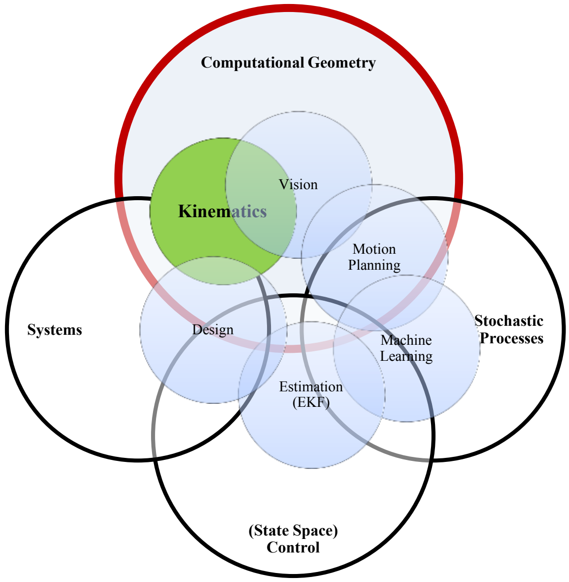
August 10, 2016

metr4202@itee.uq.edu.au
http://robotics.itee.uq.edu.au/~metr4202/ [http://**metr4202.com**]

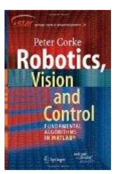
Schedule of Events

Week	Date	Lecture (W: 12:05-1:50, 50-N202)
1	27-Jul	Introduction
2	3-Aug	Representing Position & Orientation & State (Frames, Transformation Matrices & Affine Transformations)
3	**10-Aug**	**Robot Kinematics Review (&** *Ekka Day*)
4	17-Aug	Robot Dynamics
5	24-Aug	Robot Sensing: Perception
6	31-Aug	Robot Sensing: Multiple View Geometry
7	7-Sep	Robot Sensing: Feature Detection (as Linear Observers)
8	14-Sep	Probabilistic Robotics: Localization
9	21-Sep	Probabilistic Robotics: SLAM
	28-Sep	*Study break*
10	5-Oct	Motion Planning
11	12-Oct	State-Space Modelling
12	19-Oct	Shaping the Dynamic Response
13	26-Oct	LQR + Course Review

Course Organization



Follow Along Reading:



[Robotics, Vision & Control](#)
by [Peter Corke](#)

Also online:[*SpringerLink*](#)

[UQ Library eBook: 364220144X](#)

Today

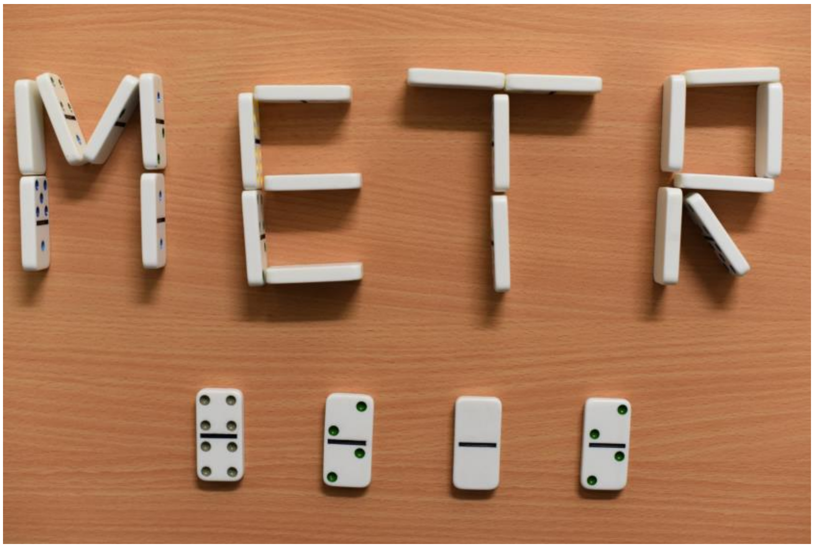
➔ Representing Space 🡐

- RVC
  - Chapter 7: Robot Arm Kinematics
  -

- Inverse Kinematics
  - RVC
    §7.3: Robot Arm Kinematics

Next Time

The Project!
"Robotics: Domino Effect"



Generalizing

Special Orthogonal & Special Euclidean Lie Algebras

- SO(n): R**o**tations

$$SO(n) = \{R \in \mathbb{R}^{n\times n} : RR^T = I, \det R = +1\}.$$

$$\exp(\widehat{\omega}\theta) = e^{\widehat{\omega}\theta} = I + \theta\widehat{\omega} + \frac{\theta^2}{2!}\widehat{\omega}^2 + \frac{\theta^3}{3!}\widehat{\omega}^3 + \ldots$$

- SE(n): Transformations of EUCLIDEAN space

$$SE(n) := \mathbb{R}^n \times SO(n).$$

$$SE(3) = \{(p, R) : p \in \mathbb{R}^3, R \in SO(3)\} = \mathbb{R}^3 \times SO(3).$$

Projective Transformations …

Group	Matrix	Distortion	Invariant properties
Projective 8 dof	$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$		Concurrency, collinearity, **order of contact**: intersection (1 pt contact); tangency (2 pt contact); inflections (3 pt contact with line); tangent discontinuities and cusps. cross ratio (ratio of ratio of lengths).
Affine 6 dof	$\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Parallelism, ratio of areas, ratio of lengths on collinear or parallel lines (e.g. midpoints), linear combinations of vectors (e.g. centroids). The line at infinity, $\mathbf{l}_\infty$.
Similarity 4 dof	$\begin{bmatrix} sr_{11} & sr_{12} & t_x \\ sr_{21} & sr_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Ratio of lengths, angle. The circular points, **I**, **J** (see section 2.7.3).
Euclidean 3 dof	$\begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Length, area

p.44, R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*

Homogenous Coordinates

$$\widehat{p} = \begin{bmatrix} \rho p_x & \rho p_y & \rho p_z & \rho \end{bmatrix}^T$$

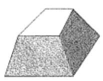
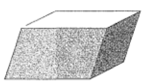
- ρ is a scaling value

Homogenous Transformation

$$\begin{bmatrix} {}^{A}R_{B} & {}^{A}p \\ \gamma & \rho \end{bmatrix}$$

- γ is a projective transformation
- The Homogenous Transformation is a **<u>linear operation</u>** (even if projection is not)

Projective Transformations & Other Transformations of 3D Space

Group	Matrix	Distortion	Invariant properties
Projective 15 dof	$\begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{v}^{\mathsf{T}} & v \end{bmatrix}$		Intersection and tangency of surfaces in contact. Sign of Gaussian curvature.
Affine 12 dof	$\begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix}$		Parallelism of planes, volume ratios, centroids. The plane at infinity, π_∞, (see section 3.5).
Similarity 7 dof	$\begin{bmatrix} s\mathtt{R} & \mathbf{t} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix}$		The absolute conic, Ω_∞, (see section 3.6).
Euclidean 6 dof	$\begin{bmatrix} \mathtt{R} & \mathbf{t} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix}$		Volume.

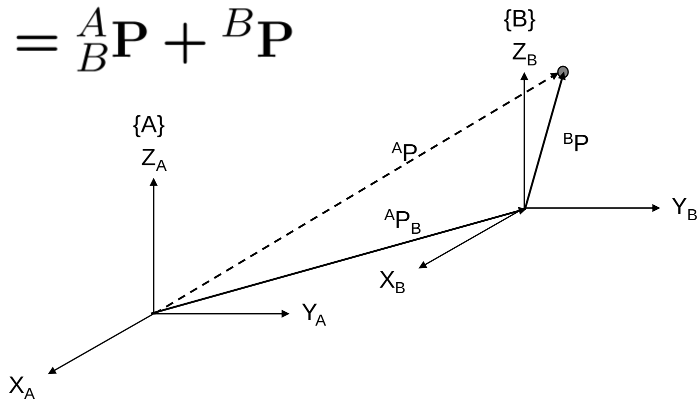
p.78, R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*

Coordinate Transformations [1]

- Translation Again:

  If {B} is translated with respect to {A} **without rotation**, then it is a vector sum

$$ {}^{A}\mathbf{P} = {}^{A}_{B}\mathbf{P} + {}^{B}\mathbf{P} $$



Coordinate Transformations [2]

- Rotation Again:

  {B} is rotated with respect to {A} then
  use rotation matrix to determine new components

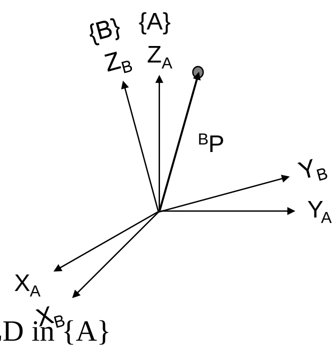
- NOTE:

$$ {}^{A}\mathbf{P} = {}^{A}_{B}\mathbf{R}\,{}^{B}\mathbf{P} $$

  – The Rotation matrix's ***subscript*** matches the position vector's **superscript**

$$ {}^{A}\mathbf{P} = {}^{A}_{[\![B]\!]}\mathbf{R}\,{}^{[\![B]\!]}\mathbf{P} $$

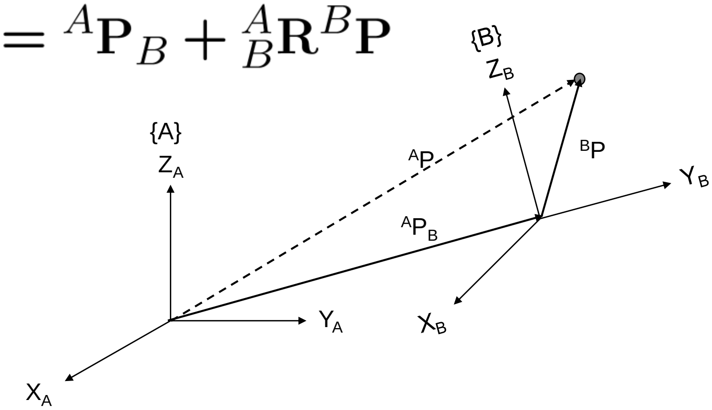
  – This gives Point Positions of {B} ORIENTED in {A}



Coordinate Transformations [3]

- Composite transformation:
  {B} is moved with respect to {A}:

$$^{A}\mathbf{P} = {}^{A}\mathbf{P}_{B} + {}^{A}_{B}\mathbf{R}\,{}^{B}\mathbf{P}$$



General Coordinate Transformations [1]

- A compact representation of the translation and rotation is known as the **Homogeneous Transformation**

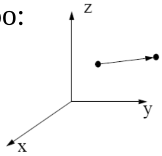
$$^{A}_{B}\mathbf{T} = \begin{bmatrix} & {}^{A}_{B}\mathbf{R} & & {}^{A}\mathbf{P}_{B} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

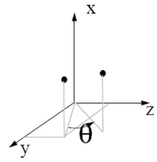
- This allows us to cast the rotation and translation of the general transform in a single matrix form

$$\begin{bmatrix} {}^{A}\mathbf{P} \\ 1 \end{bmatrix} = {}^{A}_{B}\mathbf{T} \begin{bmatrix} {}^{B}\mathbf{P} \\ 1 \end{bmatrix}$$

General Coordinate Transformations [2]

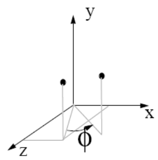
- Similarly, fundamental orthonormal transformations can be represented in this form too:

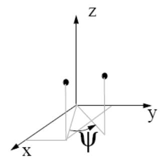




$$Trans(u, v, w) = \begin{bmatrix} 1 & 0 & 0 & u \\ 0 & 1 & 0 & v \\ 0 & 0 & 1 & w \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Rotx(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\theta & -s\theta & 0 \\ 0 & s\theta & c\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$





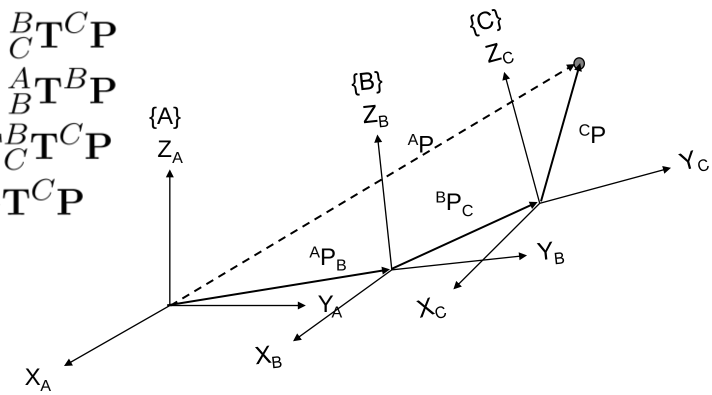
$$Roty(\phi) = \begin{bmatrix} c\phi & 0 & s\phi & 0 \\ 0 & 1 & 0 & 0 \\ -s\phi & 0 & c\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Rotz(\psi) = \begin{bmatrix} c\psi & -s\psi & 0 & 0 \\ s\psi & c\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

General Coordinate Transformations [3]

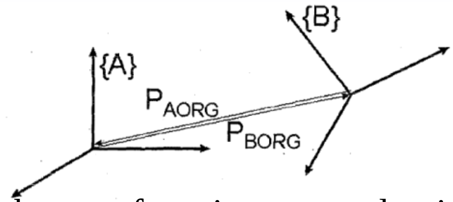
- Multiple transformations compounded as a chain

$$\begin{aligned} {}^{B}\mathbf{P} &= {}^{B}_{C}\mathbf{T}\,{}^{C}\mathbf{P} \\ {}^{A}\mathbf{P} &= {}^{A}_{B}\mathbf{T}\,{}^{B}\mathbf{P} \\ &= {}^{A}_{B}\mathbf{T}\,{}^{B}_{C}\mathbf{T}\,{}^{C}\mathbf{P} \\ &= {}^{A}_{C}\mathbf{T}\,{}^{C}\mathbf{P} \end{aligned}$$



$${}^{A}_{C}\mathbf{T} = \begin{bmatrix} {}^{A}_{B}\mathbf{R}\,{}^{B}_{C}\mathbf{R} & {}^{A}\mathbf{P}_B + {}^{A}_{B}\mathbf{R}\,{}^{B}\mathbf{P}_C \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix}$$

Inverse of a Homogeneous Transformation Matrix



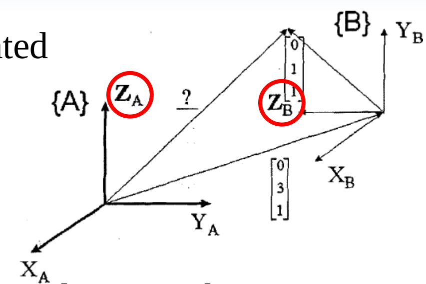
- The inverse of the transform is **not** equal to its transpose because this 4×4 matrix is not orthonormal ($T^{-1} \neq T^T$)
- Invert by parts to give:

$$ {}^{A}_{B}T = \begin{bmatrix} {}^{A}_{B}R & {}^{A}\mathbf{p}_{Borg/O_A} \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix} $$

$$ {}^{A}_{B}T^{-1} = {}^{B}_{A}T = \begin{bmatrix} {}^{A}_{B}R^T & -{}^{A}_{B}R^T \cdot {}^{A}\mathbf{p}_{Borg/O_A} \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^{B}_{A}R & {}^{B}\mathbf{p}_{Aorg/O_B} \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix} $$

Tutorial Problem

The origin of frame *{B}* is translated to a position [0 3 1] with respect to frame *{A}*.



We would like to find:

1. The homogeneous transformation between the two frames in the figure.
2. For a point *P* defined as as [0 1 1] in frame *{B}*, we would like to find the vector describing this point with respect to frame *{A}*.

Tutorial Solution

- The matrix $_BT^A$ is formed as defined earlier:
- Since P in the frame is:
- We find vector **p** in frame {A} using the relationship

➔

- The matrix $_BT^A$ is formed as defined earlier:

$$ {}^A_BT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 3 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

- Since P in the frame is: $ {}^B\mathbf{p} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} $

- We find vector **p** in frame *{A}* using the relationship

$$ {}^Ap = {}^A_BT\,{}^Bp $$

➔ $ {}^A\mathbf{p} = \begin{bmatrix} 0 \\ 2 \\ 2 \\ 1 \end{bmatrix} $

Cool Robotics Share

http://www.kinemasystems.com/

Looking in Detail: Forward & Inverse Kinematics

1. Forward Kinematics ($\theta \rightarrow x$)
2. Inverse Kinematics ($x \rightarrow \theta$)
3. Denavit Hartenberg [DH] Notation
4. Affine Transformations &
5. Theoretical (General) Kinematics

Forward Kinematics

Forward Kinematics [1]

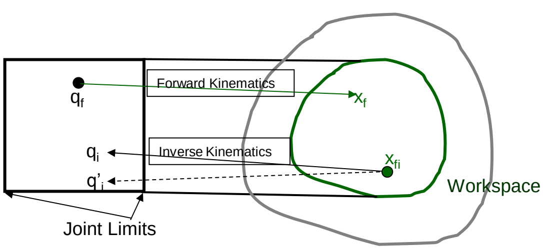
- Forward kinematics is the process of chaining homogeneous transforms together. For example to:
  - Find the articulations of a mechanism, or
  - the fixed transformation between two frames which is known in terms of linear and rotary parameters.
- Calculates the final position from the **machine** (**joint variables**)
- Unique for an open kinematic chain (**serial arm**)
- "Complicated" (multiple solutions, etc.) for a closed kinematic chain (**parallel arm**)

Forward Kinematics [2]

- Can think of this as "spaces":
  - Workspace (x,y,z,α,β,γ): The robot's position & orientation
  - Joint space ($\theta_1 \ldots \theta_n$): A state-space vector of joint variables

$$\vec{\mathbf{x}} = \begin{bmatrix} \vec{\mathbf{p}} \\ \vec{\Theta} \end{bmatrix}$$

$$\vec{\mathbf{q}} = \begin{bmatrix} q_1 \\ \vdots \\ q_n \end{bmatrix}$$

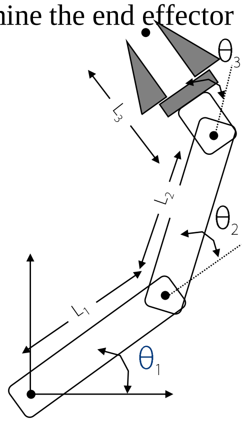


Forward Kinematics [3]

- Consider a planar RRR manipular
- Given the joint angles and link lengths, we can determine the end effector pose:

$$x = L_1 \cos\theta_1 + L_2 \cos(\theta_1 + \theta_2) + \dots$$
$$L_3 \cos(\theta_1 + \theta_2 + \theta_3)$$

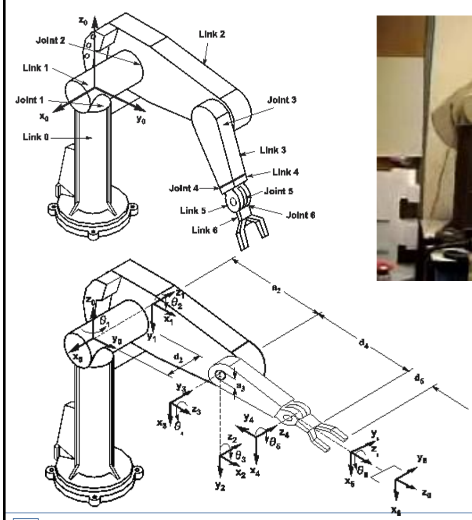
$$y = L_1 \sin\theta_1 + L_2 \sin(\theta_1 + \theta_2) + \dots$$
$$L_3 \sin(\theta_1 + \theta_2 + \theta_3)$$



- This isn't too difficult to determine for a simple, planar manipulator. BUT …

Forward Kinematics [4]: The PUMA 560!

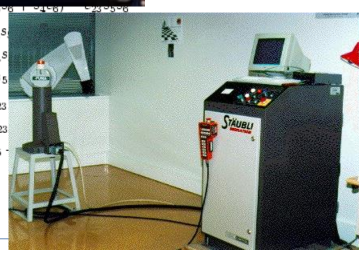
- What about a more complicated mechanism?



$$\begin{pmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$) - s_1(s_4c_5c_6 + c_4s_6)$
$) + c_1(s_4c_5c_6 + c_4s_6)$
$s_6) - s_1(-s_4c_5s_6 + c_4c_6)$
$s_6) + c_1(-s_4c_5s_6 + c_4c_6)$

$a_x = c_1(c_{23}c_4s$
$a_y = s_1(c_{23}c_4s$
$a_z = -s_{23}c_4s_5$
$p_x = c_1(d_6(c_{23}$
$p_y = s_1(d_6(c_{23}$
$p_z = d_6(c_{23}c_5$



Denavit Hartenberg [DH] Notation

Denavit Hartenberg [DH] Notation

- J. Denavit and R. S. Hartenberg first proposed the use of homogeneous transforms for articulated mechanisms

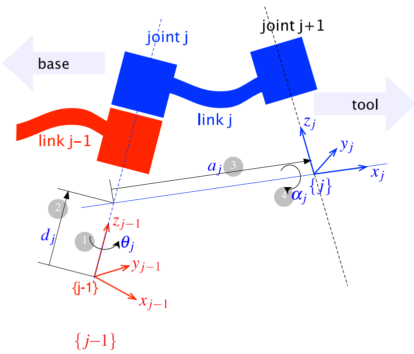
  (But B. Roth, introduced it to robotics)

- A kinematics "short-cut" that reduced the number of parameters by adding a structure to frame selection

- For two frames positioned in space, the first can be moved into coincidence with the second by a sequence of 4 operations:
  - rotate around the x_{i-1} axis by an angle α_i
  - translate along the x_{i-1} axis by a distance a_i
  - translate along the new z axis by a distance d_i
  - rotate around the new z axis by an angle θ_i

Denavit-Hartenberg Convention

- link length a_i the offset distance between the z_{i-1} and z_i axes along the x_i axis;
- link twist α_i the angle from the z_{i-1} axis to the z_i axis about the x_i axis;



Art c/o P. Corke

- link offset d_i the distance from the origin of frame i-1 to the x_i axis along the z_{i-1} axis;
- joint angle θ_i the angle between the x_{i-1} and x_i axes about the z_{i-1} axis.

DH: Where to place frame?

1. Align an axis along principal motion
   1. Rotary (**R**): align rotation axis along the z axis
   2. Prismatic (**P**): align slider travel along x axis
2. Orient so as to position x axis towards next frame
3. $\theta_{(\text{rot } z)} \rightarrow d_{(\text{trans } z)} \rightarrow a_{(\text{trans } x)} \rightarrow \alpha_{(\text{rot } x)}$

Denavit-Hartenberg → Rotation Matrix

- Each transformation is a product of 4 “basic” transformations (instead of 6)

$$
{}^{i-1}A_i = Rot_{z,\theta_i} Trans_{z,d_i} Trans_{x,a_i} Rot_{x,\alpha_i}
$$

$$
= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdots
$$

$$
\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} c_{\alpha_i} & s_{\theta_i} s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i} c_{\alpha_i} & -c_{\theta_i} s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

DH Example [1]: RRR Link Manipulator

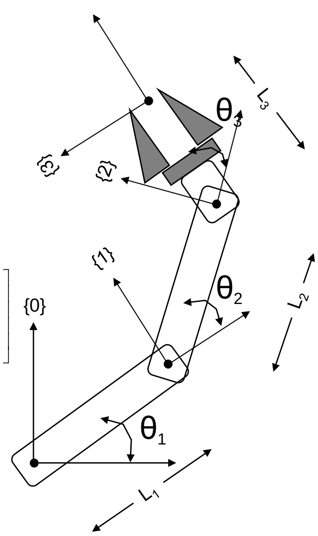
1. Assign the frames at the joints …
2. Fill DH Table …

Link	a_i	α_i	d_i	θ_i
1	L_1	0	0	$\boldsymbol{\theta_1}$
2	L_2	0	0	$\boldsymbol{\theta_2}$
3	L_3	0	0	$\boldsymbol{\theta_3}$

$$
{}^0A_1 = \begin{bmatrix} c_{\theta_1} & -s_{\theta_1} & 0 & L_1 c_{\theta_1} \\ s_{\theta_1} & c_{\theta_1} & 0 & L_1 s_{\theta_1} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^1A_2 = \begin{bmatrix} c_{\theta_2} & -s_{\theta_2} & 0 & L_2 c_{\theta_2} \\ s_{\theta_2} & c_{\theta_2} & 0 & L_2 s_{\theta_2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^2A_3 = \begin{bmatrix} c_{\theta_3} & -s_{\theta_3} & 0 & L_3 c_{\theta_3} \\ s_{\theta_3} & c_{\theta_3} & 0 & L_3 s_{\theta_3} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^0T_3 = {}^0A_1 \, {}^1A_2 \, {}^2A_3
$$

$$
= \begin{bmatrix} c_{\theta_{123}} & -s_{\theta_{123}} & 0 & L_1 c_{\theta_1} + L_2 c_{\theta_{12}} + L_3 c_{\theta_{123}} \\ s_{\theta_{123}} & c_{\theta_{123}} & 0 & L_1 s_{\theta_1} + L_2 s_{\theta_{12}} + L_3 s_{\theta_{123}} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$



DH Example [2]: RRP Link Manipulator

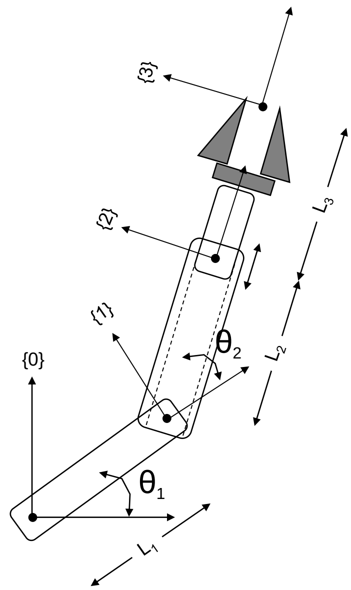
1. Assign the frames at the joints …
2. Fill DH Table …

Link	a_i	α_i	d_i	θ_i
1	L_1	0	0	**θ_1**
2	L_2	0	0	**θ_2**
3	**L_3**	0	0	0

$$
{}^0A_1 = \begin{bmatrix} c_{\theta_1} & -s_{\theta_1} & 0 & L_1 c_{\theta_1} \\ s_{\theta_1} & c_{\theta_1} & 0 & L_1 s_{\theta_1} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\quad {}^1A_2 = \begin{bmatrix} c_{\theta_2} & -s_{\theta_2} & 0 & L_2 c_{\theta_2} \\ s_{\theta_2} & c_{\theta_2} & 0 & L_2 s_{\theta_2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\quad {}^2A_3 = \begin{bmatrix} 1 & 0 & 0 & L_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^0T_3 = {}^0A_1\,{}^1A_2\,{}^2A_3
$$

$$
= \begin{bmatrix} c_{\theta_{12}} & -s_{\theta_{12}} & 0 & L_1 c_{\theta_1} + (L_2 + L_3) c_{\theta_{12}} \\ s_{\theta_{12}} & c_{\theta_{12}} & 0 & L_1 s_{\theta_1} + (L_2 + L_3) s_{\theta_{12}} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$



DH Example [3]: Puma 560

- “Simple” 6R robot exercise for the reader …

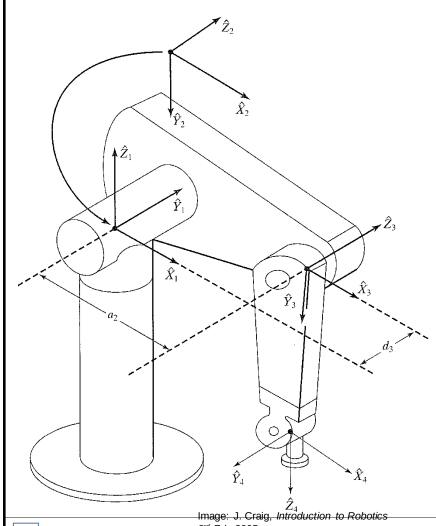
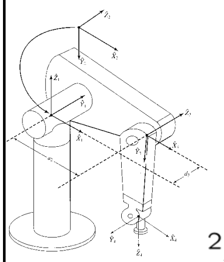


Image: J. Craig, *Introduction to Robotics* 3rd Ed., 2005

Link	a_i	α_i	d_i	θ_i
1	0	0	0	**θ_1**
2	0	$-\pi/2$	0	**θ_2**
3	L_2	0	D_3	**θ_3**
4	L_3	$-\pi/2$	D_4	**θ_4**
5	0	$\pi/2$	0	**θ_5**
6	0	$-\pi/2$	0	**θ_6**

DH Example [3]: Puma 560 [2]



$$^0A_1 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad ^1A_2 = \begin{bmatrix} c_2 & -s_2 & 0 & 0 \\ 0 & 0 & 1 & d_2 \\ -s_2 & -c_2 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

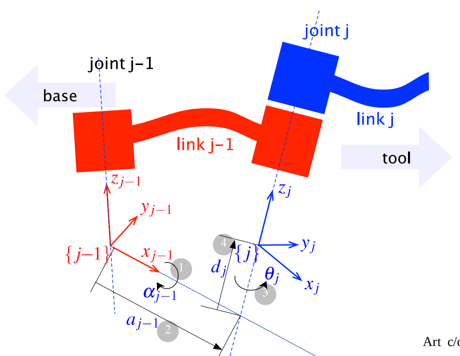
$$^2A_3 = \begin{bmatrix} c_3 & -s_3 & 0 & L_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad ^3A_4 = \begin{bmatrix} c_4 & -s_4 & 0 & L_3 \\ 0 & 0 & 1 & d_4 \\ -s_4 & -c_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^4A_5 = \begin{bmatrix} c_4 & -s_5 & 0 & L_3 \\ 0 & 0 & 1 & d_4 \\ -s_5 & -c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad ^5A_6 = \begin{bmatrix} c_6 & -s_6 & 0 & L_3 \\ 0 & 0 & -1 & 0 \\ -s_6 & -c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^0T_6 = {}^0A_1 {}^1A_2 {}^2A_3 {}^3A_4 {}^4A_5 {}^5A_6$$

Modified DH

- Made “popular” by Craig’s *Intro. to Robotics* book
- Link coordinates attached to the near by joint

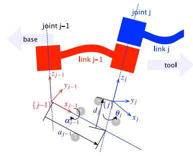


Art c/o P. Corke

- a (trans *x-1*) → α (rot *x-1*) → θ (rot *z*) → d (trans *z*)

Modified DH [2]

- Gives a similar result
  (but it's not commutative)



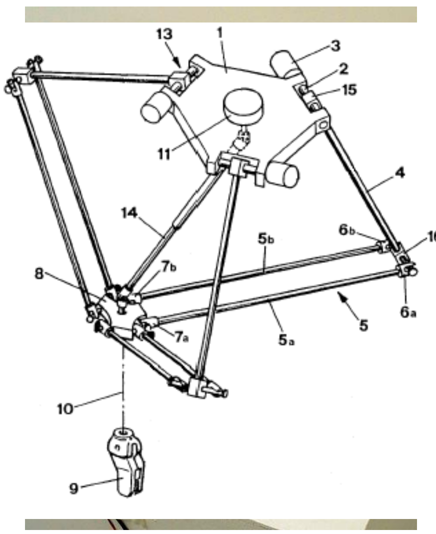
$$\Rightarrow {}^{i-1}A_i = R_x(\alpha_{i-1})\, T_x(a_{i-1})\, R_z(\theta_i)\, T_x(d_i)$$

- Refactoring Standard → to Modified

$$\underbrace{\{R_z(\theta_1)\, T_z(d_1)\, T_x(a_1)\, R_x(\alpha_1)\}}_{\text{DH}_1} \cdot \underbrace{\{R_z(\theta_2)\, T_z(d_2)\, T_x(a_2)\, R_x(\alpha_2)\}}_{\text{DH}_2} \cdot \underbrace{\{R_z(\theta_3)\, T_z(d_3)\}}_{\text{End Effector}}$$

$$= \underbrace{\{R_z(\theta_1)\, T_z(d_1)\}}_{\text{Base}} \cdot \underbrace{\{T_x(a_1)\, R_x(\alpha_1)\, R_z(\theta_2)\, T_z(d_2)\}}_{\text{MDH}_1} \cdot \underbrace{\{T_x(a_2)\, R_x(\alpha_2)\, R_z(\theta_3)\, T_z(d_3)\}}_{\text{MDH}_2}$$

Parallel Manipulators

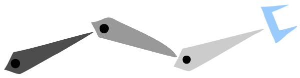


Sources: Wikipedia, "Delta Robot", ParallelMic.Org, "Delta Parallel Robot", and US Patent 4,976,582

- The "central" Kinematic structure is made up of closed-loop chain(s)
- Compared to Serial Mechanisms:
  - \+ Higher Stiffness
  - \+ Higher Payload
  - \+ Less Inertia
  - – Smaller Workspace
  - – Coordinated Drive System
  - – More Complex & $$$

Inverse Kinematics

- Forward: angles → position
  $\mathbf{x} = f(\boldsymbol{\theta})$
- Inverse: position → angles
  $\boldsymbol{\theta} = f^{-1}(\mathbf{x})$
- Analytic Approach



- Numerical Approaches:
  - Jacobian: $J = \frac{\delta x}{\delta q} \rightarrow \delta q \approx J^{-1}\delta x$
  - J^T Approximation: $\tau = J^T \cdot \mathbf{F} \rightarrow \Delta q \approx J^T \Delta x$
    - Slotine & Sheridan method
  - Cyclical Coordinate Descent

Inverse Kinematics

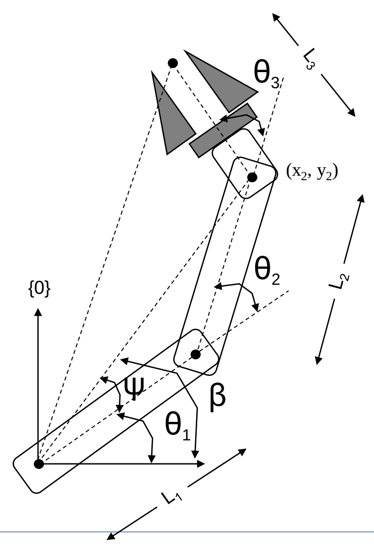
- Inverse Kinematics is the problem of finding the joint parameters given only the values of the homogeneous transforms which model the mechanism
  (i.e., the pose of the end effector)
- Solves the problem of where to drive the joints in order to get the hand of an arm or the foot of a leg in the right place
- In general, this involves the solution of a set of simultaneous, non-linear equations
- Hard for serial mechanisms, easy for parallel

Solution Methods

- Unlike with systems of linear equations, there are no general algorithms that may be employed to solve a set of nonlinear equation
- **Closed-form** and **numerical** methods exist
- Many exist: Most general solution to a 6R mechanism is Raghavan and Roth (1990)
- Three methods of obtaining a solution are popular:
  (1) **geometric** | (2) **algebraic** | (3) **DH**

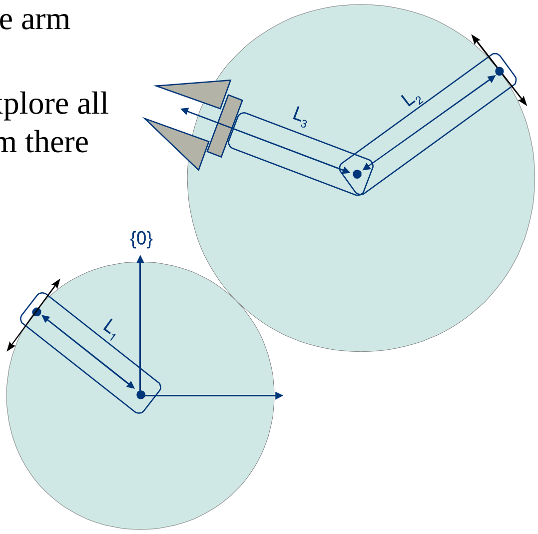
Inverse Kinematics: Geometrical Approach

- We can also consider the geometric relationships defined by the arm



Inverse Kinematics: Geometrical Approach [2]

- We can also consider the geometric relationships defined by the arm
- Start with what is fixed, explore all geometric possibilities from there



Inverse Kinematics: Algebraic Approach

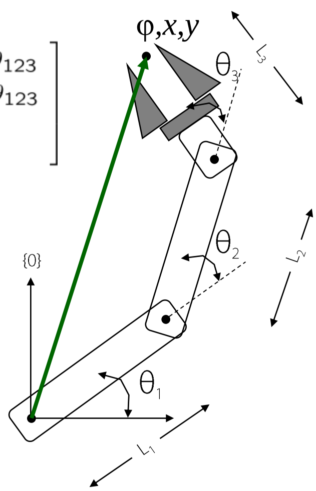
- We have a series of equations which define this system
- Recall, from Forward Kinematics:

$$ {}^0T_3 = \begin{bmatrix} c_{\theta_{123}} & -s_{\theta_{123}} & 0 & L_1 c_{\theta_1} + L_2 c_{\theta_{12}} + L_3 c_{\theta_{123}} \\ s_{\theta_{123}} & c_{\theta_{123}} & 0 & L_1 s_{\theta_1} + L_2 s_{\theta_{12}} + L_3 s_{\theta_{123}} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

- The end-effector pose is given by

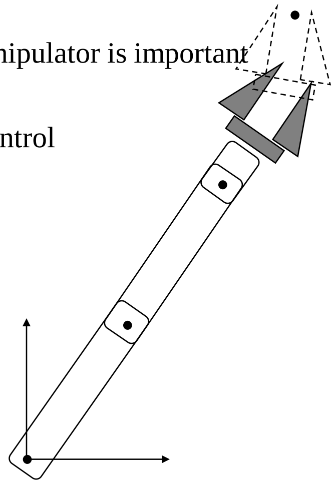
$$ {}^0T_3 = \begin{bmatrix} c_\phi & -s_\phi & 0 & x \\ s_\phi & c_\phi & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

- Equating terms gives us a set of algebraic relationships



No Solution - Singularity

- Singular positions:



- An understanding of the workspace of the manipulator is important
- There will be poses that are not achievable
- There will be poses where there is a loss of control

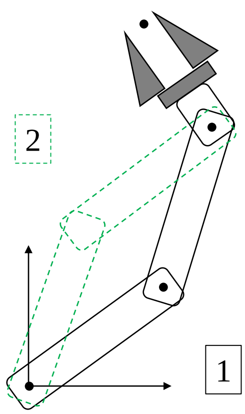
- Singularities also occur when the manipulator loses a DOF
  - This typically happens when joints are aligned
  - **det[Jacobian]=0**

Multiple Solutions

- There will often be multiple solutions for a particular inverse kinematic analysis

- Consider the three link manipulator shown. Given a particular end effector pose, two solutions are possible

- The choice of solution is a function of proximity to the current pose, limits on the joint angles and possible obstructions in the workspace

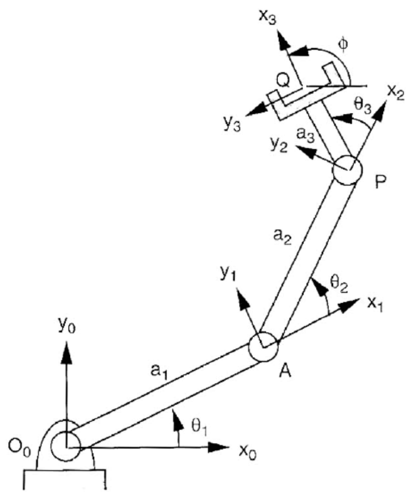


Inverse Kinematics

Inverse Kinematics [More Generally]

- Freudenstein (1973) referred to the inverse kinematics problem of the most general **6R** manipulator as the "Mount Everest" of kinematic problems.
- Tsai and Morgan (1985) and Primrose (1986) proved that this has at most 16 real solutions.
- Duffy and Crane (1980) derived a closed-form solution for the general **7R** single-loop spatial mechanism.
  - The solution was obtained in the form of a 16 x 16 delerminant in which every element is a second-degree polynomial in one joint variable. The determinant, when expended, should yield a 32nd-degree polynomial equation and hence confirms the upper limit predicted by Roth *et al*. (1973).
- Tsai and Morgan (1985) used the homotopy continuation method to solve the inverse kinematics of the general 6R manipulator and found only 16 solutions
- Raghavan and Roth (1989, 1990) used the dyalitic elimination method to derive a 16th-degree polynomial for the general 6R inverse kinematics problem.

Example: FK/IK of a 3R Planar Arm



- Derived from Tsai (p. 63)

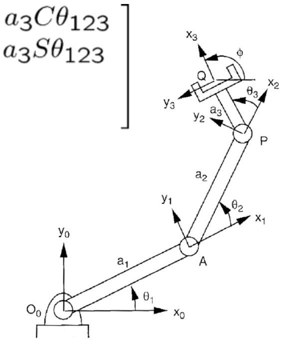
Example: 3R Planar Arm [2]

Position Analysis: 3·Planar 1-R Arm rotating about **Z [Ⓩ]**

$$^{0}A_3 = \; ^{0}A_1 \cdot ^{1}A_2 \cdot ^{2}A_3$$

Substituting gives:

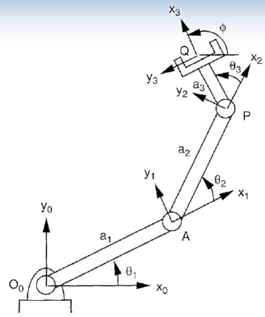
$$^{0}A_3 = \begin{bmatrix} C\theta_{123} & -S\theta_{123} & 0 & a_1C\theta_1 + a_2C\theta_{12} + a_3C\theta_{123} \\ S\theta_{123} & C\theta_{123} & 0 & a_1S\theta_1 + a_2S\theta_{12} + a_3S\theta_{123} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Example: 3R Planar Arm [2]

Forward Kinematics

(solve for **x given θ** → **x** = *f* (**θ**))



Fairly straight forward:

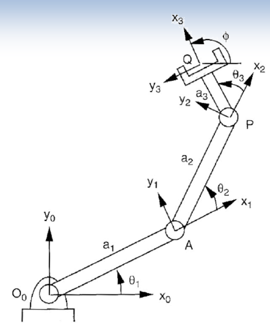
$$ {}^0R_3 = \begin{bmatrix} C\theta_{123} & -S\theta_{123} & 0 \\ S\theta_{123} & C\theta_{123} & 0 \\ 0 & 0 & 1 \end{bmatrix} $$

$$ {}^0P_3 = \begin{bmatrix} a_1C\theta_1 + a_2C\theta_{12} + a_3C\theta_{123} \\ a_1S\theta_1 + a_2S\theta_{12} + a_3S\theta_{123} \\ 0 \end{bmatrix} $$

Example: 3R Planar Arm [3]

Inverse Kinematics

(solve for **θ given x** → **x** = *f* (**θ**))



- Start with orientation φ:

$C\theta_{123} = C\phi,\ S\theta_{123} = S\phi$

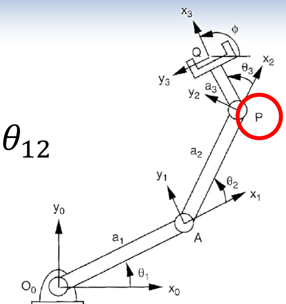
$\Rightarrow \theta_{123} = \theta_1 + \theta_2 + \theta_3 = \phi$

- Get overall position $\boldsymbol{q} = [q_x \quad q_y]$:

$q_x - a_3C\phi = a_1C\theta_1 + a_2C\theta_{12}$

$q_y - a_3S\phi = a_1S\theta_1 + a_2S\theta_{12}\ \ldots$

Example: 3R Planar Arm [4]



- Introduce $\boldsymbol{p} = [p_x \quad p_y]$ before "wrist"

$$p_x = a_1 C\theta_1 + a_2 C\theta_{12}, p_y = a_1 S\theta_1 + a_2 S\theta_{12}$$
$$\Rightarrow p_x^2 + p_y^2 = a_1^2 + a_2^2 + 2a_1 a_2 C\theta_2$$

- Solve for θ_2:

$$\theta_2 = \cos^{-1}\kappa, \kappa = \frac{p_x^2+p_y^2-a_1^2-a_2^2}{2a_1 a_2}$$ (2 ℝ roots if |κ|<1)

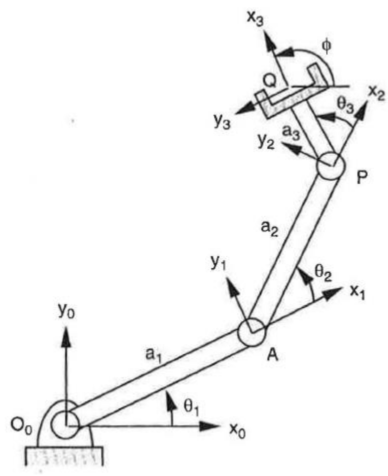
- Solve for θ_1:

$$C\theta_1 = \frac{p_x(a_1+a_2C\theta_2)+p_y a_2 S\theta_2}{a_1^2+a_2^2+2a_1a_2C\theta_2}, S\theta_1 = \frac{-p_x a_2 S\theta_2+p_y(a_1+a_2C\theta_2)}{a_1^2+a_2^2+2a_1a_2C\theta_2}$$

$$\theta_1 = atan2(S\theta_1, C\theta_1)$$

Inverse Kinematics: Example I

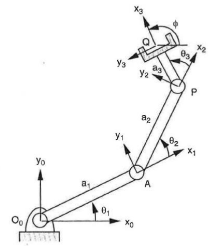
Planar Manipulator:



Inverse Kinematics: Example I

- Forward Kinematics:

[For the Frame {Q} at the end effector]:



$$\begin{bmatrix} q_x \\ q_y \\ q_z \\ 1 \end{bmatrix} = {}^0A_3 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 c\theta_1 + a_2 c\theta_{12} + a_3 c\theta_{123} \\ a_1 s\theta_1 + a_2 s\theta_{12} + a_3 s\theta_{123} \\ 0 \\ 1 \end{bmatrix}$$

$\because$

$${}^0A_3 = \begin{bmatrix} c\theta_{123} & -s\theta_{123} & 0 & a_1 c\theta_1 + a_2 c\theta_{12} + a_3 c\theta_{123} \\ s\theta_{123} & c\theta_{123} & 0 & a_1 s\theta_1 + a_2 s\theta_{12} + a_3 s\theta_{123} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

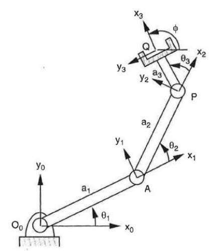
- For an arbitrary point **G** in the end effector: ${}^3\mathbf{g} = [g_u, g_v, 0, 1]^T$

$$\begin{bmatrix} g_x \\ g_y \\ g_z \\ 1 \end{bmatrix} = {}^0A_3 \begin{bmatrix} g_u \\ g_v \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} g_u c\theta_{123} - g_v s\theta_{123} + a_1 c\theta_1 + a_2 c\theta_{12} + a_3 c\theta_{123} \\ g_u s\theta_{123} + g_v c\theta_{123} + a_1 s\theta_1 + a_2 s\theta_{12} + a_3 s\theta_{123} \\ 0 \\ 1 \end{bmatrix}$$

Inverse Kinematics: Example I

- Forward Kinematics:

[For the Frame {Q} at the end effector]:



$$\begin{bmatrix} q_x \\ q_y \\ q_z \\ 1 \end{bmatrix} = {}^0A_3 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 c\theta_1 + a_2 c\theta_{12} + a_3 c\theta_{123} \\ a_1 s\theta_1 + a_2 s\theta_{12} + a_3 s\theta_{123} \\ 0 \\ 1 \end{bmatrix}$$

$\because$

$${}^0A_3 = \begin{bmatrix} c\theta_{123} & -s\theta_{123} & 0 & a_1 c\theta_1 + a_2 c\theta_{12} + a_3 c\theta_{123} \\ s\theta_{123} & c\theta_{123} & 0 & a_1 s\theta_1 + a_2 s\theta_{12} + a_3 s\theta_{123} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

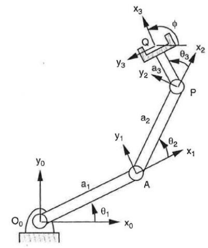
- For an arbitrary point **G** in the end effector: ${}^3\mathbf{g} = [g_u, g_v, 0, 1]^T$

$$\begin{bmatrix} g_x \\ g_y \\ g_z \\ 1 \end{bmatrix} = {}^0A_3 \begin{bmatrix} g_u \\ g_v \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} g_u c\theta_{123} - g_v s\theta_{123} + a_1 c\theta_1 + a_2 c\theta_{12} + a_3 c\theta_{123} \\ g_u s\theta_{123} + g_v c\theta_{123} + a_1 s\theta_1 + a_2 s\theta_{12} + a_3 s\theta_{123} \\ 0 \\ 1 \end{bmatrix}$$

Inverse Kinematics: Example I

- Inverse Kinematics:
  - Set the final position equal to the Forward Transformation Matrix $^{0}\mathbf{A}_3$:

$$^{0}A_3 = \begin{bmatrix} c\phi & -s\phi & 0 & q_x \\ s\phi & c\phi & 0 & q_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



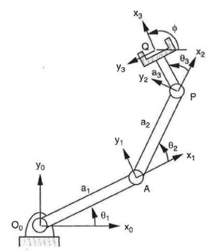
- The solution strategy is to equate the elements of $^{0}\mathbf{A}_3$ to that of the given position (q_x, q_y) and orientation ϕ

Inverse Kinematics: Example I

- Orientation (ϕ):

$$c\theta_{123} = c\phi,$$
$$s\theta_{123} = s\phi.$$

$$\theta_{123} = \theta_1 + \theta_2 + \theta_3 = \phi.$$



- Now Position of the 2DOF point **P**:

$$p_x = a_1 c\theta_1 + a_2 c\theta_{12},$$
$$p_y = a_1 s\theta_1 + a_2 s\theta_{12},$$

$\therefore \quad p_x = q_x - a_3 c\phi \qquad p_y = q_y - a_3 s\phi$

- Substitute: θ_3 disappears and now we can eliminate θ_1:

$$p_x^2 + p_y^2 = a_1^2 + a_2^2 + 2a_1a_2c\theta_2.$$

Inverse Kinematics: Example I

- we can eliminate θ_1…

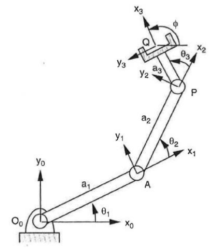
  $p_x^2 + p_y^2 = a_1^2 + a_2^2 + 2a_1a_2c\theta_2.$

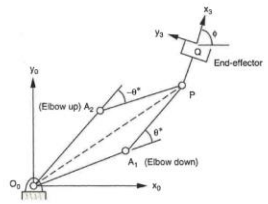
- Then solve for θ_{12}:

  $$\theta_2 = \cos^{-1}\kappa, \qquad \kappa = \frac{p_x^2 + p_y^2 - a_1^2 - a_2^2}{2a_1a_2}$$

  – This gives 2 real ($\mathbb{R}$) roots if $|\kappa| < 1$
  – One double root if $|\kappa| = 1$
  – No real roots if $|\kappa| > 1$

- Elbow up/down:
  – In general, **if** θ_2 is a solution **then** $-\theta_2$ is a solution





Inverse Kinematics: Example I

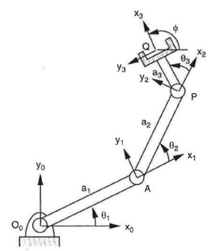
- Solving for θ_1…
  – Corresponding to each θ_2, we can solve θ_1

$$(a_1 + a_2c\theta_2)c\theta_1 - (a_2s\theta_2)s\theta_1 = p_x$$
$$(a_2s\theta_2)c\theta_1 + (a_1 + a_2c\theta_2)s\theta_1 = p_y$$

$$c\theta_1 = \frac{p_x(a_1 + a_2c\theta_2) + p_ya_2s\theta_2}{\Delta},$$
$$s\theta_1 = \frac{-p_xa_2s\theta_2 + p_y(a_1 + a_2c\theta_2)}{\Delta}$$

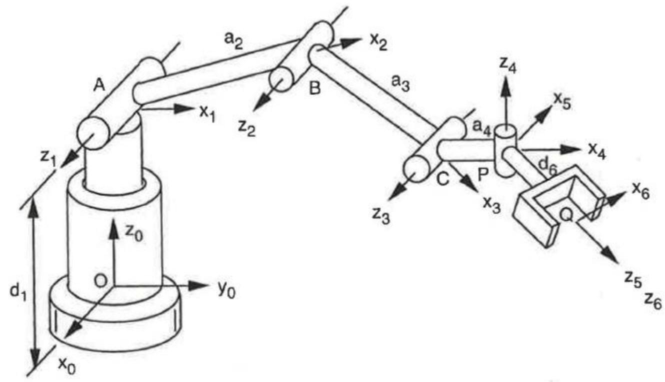
$$\Delta = a_1^2 + a_2^2 + 2a_1a_2c\theta_2$$

$$\theta_1 = \text{Atan2}(s\theta_1, c\theta_1)$$



Inverse Kinematics: Example II

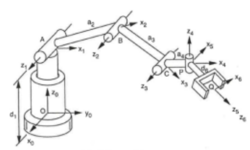
Elbow Manipulator:



Inverse Kinematics: Example II

- Target Position:

$\mathbf{u} = [u_x, u_y, u_z]^T$, $\mathbf{v} = [v_x, v_y, v_z]^T$, $\mathbf{w} = [w_x, w_y, w_z]^T$, and
$\mathbf{p} = [p_x, p_y, p_z]^T$.



- Transformation Matrices:

$$A_1 = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad ({}^0A_1)^{-1} = \begin{bmatrix} c\theta_1 & s\theta_1 & 0 & 0 \\ -s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

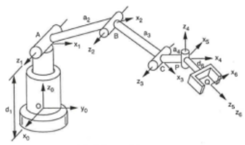
$$A_2 = \begin{bmatrix} c\theta_2 & 0 & -s\theta_2 & 0 \\ 0 & 1 & 0 & 0 \\ s\theta_2 & 0 & c\theta_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$A_3 = \begin{bmatrix} c\theta_3 & 0 & -s\theta_3 & a_2(1 - c\theta_3) \\ 0 & 1 & 0 & 0 \\ s\theta_3 & 0 & c\theta_3 & -a_2 s\theta_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_4 = \begin{bmatrix} c\theta_4 & 0 & -s\theta_4 & (a_2 + a_3)(1 - c\theta_4) \\ 0 & 1 & 0 & 0 \\ s\theta_4 & 0 & c\theta_4 & -(a_2 + a_3)s\theta_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_5 = \begin{bmatrix} c\theta_5 & -s\theta_5 & 0 & (a_2 + a_3 + a_4)(1 - c\theta_5) \\ s\theta_5 & c\theta_5 & 0 & -(a_2 + a_3 + a_4)s\theta_5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Inverse Kinematics: Example II

- Key Matrix Products:

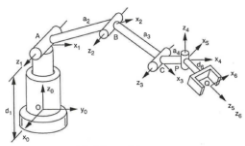


$$A_2A_3A_4 = \begin{bmatrix} c\theta_{234} & 0 & -s\theta_{234} & a_2c\theta_2 + a_3c\theta_{23} - (a_2+a_3)c\theta_{234} \\ 0 & 1 & 0 & 0 \\ s\theta_{234} & 0 & c\theta_{234} & a_2s\theta_2 + a_3s\theta_{23} - (a_2+a_3)s\theta_{234} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_1A_2A_3A_4 = \begin{bmatrix} c\theta_1c\theta_{234} & -s\theta_1 & -c\theta_1s\theta_{234} & c\theta_1[a_2c\theta_2 + a_3c\theta_{23} - (a_2+a_3)c\theta_{234}] \\ s\theta_1c\theta_{234} & c\theta_1 & -s\theta_1s\theta_{234} & s\theta_1[a_2c\theta_2 + a_3c\theta_{23} - (a_2+a_3)c\theta_{234}] \\ s\theta_{234} & 0 & c\theta_{234} & [a_2s\theta_2 + a_3s\theta_{23} - (a_2+a_3)s\theta_{234}] \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Inverse Kinematics: Example II

- Inverse Kinematics:



$$\mathbf{p} = A_1A_2A_3A_4\mathbf{p}_0.$$

$$A_1^{-1}\begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = A_2A_3A_4\begin{bmatrix} a_2+a_3+a_4 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

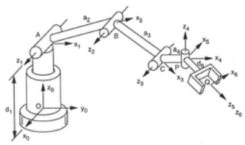
$$p_xc\theta_1 + p_ys\theta_1 = a_2c\theta_2 + a_3c\theta_{23} + a_4c\theta_{234},$$

$$-p_xs\theta_1 + p_yc\theta_1 = 0,$$

$$p_z = a_2s\theta_2 + a_3s\theta_{23} + a_4s\theta_{234}.$$

Inverse Kinematics: Example II

- Solving the System:



$$\theta_1 = \tan^{-1}\frac{p_y}{p_x}.$$

$$\theta_5 = \sin^{-1}(-w_x s\theta_1 + w_y c\theta_1).$$

$$\theta_{234} = \text{Atan2}\left[w_z/c\theta_5, (w_x c\theta_1 + w_y s\theta_1)/c\theta_5\right].$$

$$a_2 c\theta_2 + a_3 c\theta_{23} = k_1, \qquad k_1 = p_x c\theta_1 + p_y s\theta_1 - a_4 c\theta_{234}$$

$$a_2 s\theta_2 + a_3 s\theta_{23} = k_2, \qquad k_2 = p_z - a_4 s\theta_{234}$$

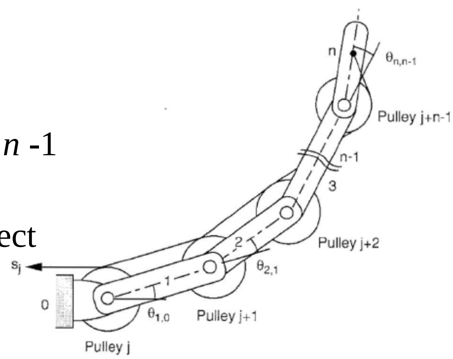
$$a_2^2 + a_3^2 + 2a_2a_3c\theta_3 = k_1^2 + k_2^2.$$

$$\theta_3 = \cos^{-1}\frac{k_1^2 + k_2^2 - a_2^2 - a_3^2}{2a_2a_3}.$$

$$\theta_6 = \text{Atan2}(s\theta_6, c\theta_6).$$

Advanced Concept: Tendon-Driven Manipulators

- Tendons may be modelled as a transmission line
- in which the links are labeled sequentially from 0 to n and the pulleys are labeled from *j* to *j* + *n* -1
- Let θ_{ji} denote the angular displacement of link *j* with respect to link *i*.
- We can write a circuit equation once for each pulley pair as follows:



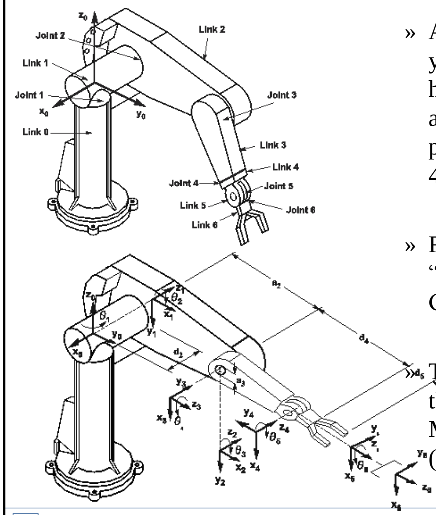
$$r_{j+i-1}\theta_{j+i-1,i} = \pm r_{j+i}\theta_{j+i,i} \qquad \text{for } i = 1, 2, \ldots, n-1.$$

$$\theta_{j+i-1,i} = \theta_{j+i-1,i-1} - \theta_{i,i-1} \qquad \text{for } i = 1, 2, \ldots, n.$$

$$\theta_{j,0} = \theta_{1,0} \pm (r_{j+1}/r_j)\theta_{2,1} \pm \cdots \pm (r_{j+n-1}/r_j)\theta_{n,n-1}.$$

Inverse Kinematics

- What about a more complicated mechanism?



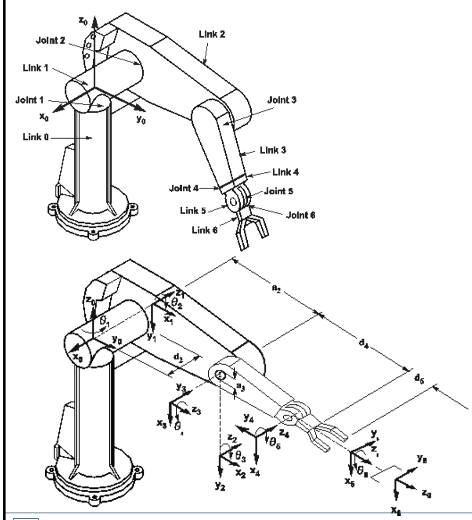
» A sufficient condition for a serial manipulator to yield a closed-form inverse kinematics solution is to have any three consecutive joint axes intersecting at a common point or any three consecutive joint axes parallel to each other. (Pieper and Roth (1969) via 4×4 matrix method)

» Raghavan and Roth 1990
"Kinematic Analysis of the 6R Manipulator of General Geometry"

» Tsai and Morgan 1985, "Solving the Kinematics of the Most General Six and Five-Dcgree-of-Freedom Manipulators by Continuation Methods" (posted online)

Inverse Kinematics

- What about a more complicated mechanism?



$$^{0}\boldsymbol{T}_6 = {}^{0}\boldsymbol{T}_1\, {}^{1}\boldsymbol{T}_2\, {}^{2}\boldsymbol{T}_3\, {}^{3}\boldsymbol{T}_4\, {}^{4}\boldsymbol{T}_5\, {}^{5}\boldsymbol{T}_6 = \begin{pmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\begin{aligned}
n_x &= c_1(c_{23}(c_4c_5c_6 - s_4s_6) - s_{23}s_5c_6) - s_1(s_4c_5c_6 + c_4s_6) \\
n_y &= s_1(c_{23}(c_4c_5c_6 - s_4s_6) - s_{23}s_5c_6) + c_1(s_4c_5c_6 + c_4s_6) \\
n_z &= -s_{23}(c_4c_5c_6 - s_4s_6) - c_{23}s_5c_6 \\
s_x &= c_1(-c_{23}(c_4c_5s_6 + s_4c_6) + s_{23}s_5s_6) - s_1(-s_4c_5s_6 + c_4c_6) \\
s_y &= s_1(-c_{23}(c_4c_5s_6 + s_4c_6) + s_{23}s_5s_6) + c_1(-s_4c_5s_6 + c_4c_6) \\
s_z &= s_{23}(c_4c_5s_6 + s_4c_6) - c_{23}s_5s_6 \\
a_x &= c_1(c_{23}c_4s_5 + s_{23}c_5) - s_1s_4s_5 \\
a_y &= s_1(c_{23}c_4s_5 + s_{23}c_5) + c_1s_4s_5 \\
a_z &= -s_{23}c_4s_5 + c_{23}c_5) \\
p_x &= c_1(d_6(c_{23}c_4s_5 + s_{23}c_5) + s_{23}d_4 + a_3c_{23} + a_2c_2) - s_1(d_6s_4s_5 + d_2) \\
p_y &= s_1(d_6(c_{23}c_4s_5 + s_{23}c_5) + s_{23}d_4 + a_3c_{23} + a_2c_2) + c_1(d_6s_4s_5 + d_2) \\
p_z &= d_6(c_{23}c_5 - s_{23}c_4s_5) + c_{23}d_4 - a_3s_{23} - a_2s_2
\end{aligned}$$

Symmetrical Parallel Manipulator

A sub-class of Parallel Manipulator:

- # Limbs (***m***) = # DOF (***F***)
- The joints are arranged in an identical pattern
- The # and location of actuated joints are the same

Thus:

- Number of Loops (L): One less than # of limbs

$$L = m - 1 = F - 1$$

- Connectivity (C_k)

$$\sum_{k=1}^{m} C_k = (\lambda + 1)\, F - \lambda$$

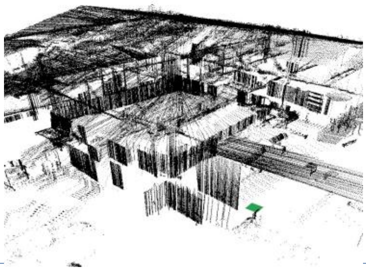
Where: λ: The DOF of the space that the system is in (e.g., λ=6 for 3D space).

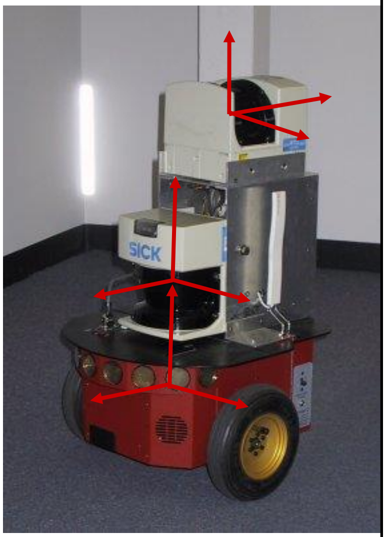
Mobile Platforms

- The preceding kinematic relationships are also important in mobile applications
- When we have sensors mounted on a platform, we need the ability to translate from the sensor frame into some world frame in which the vehicle is operating
- Should we just treat this as a P(*) mechanism?

Mobile Platforms [2]

- We typically assign a frame to the base of the vehicle
- Additional frames are assigned to the sensors
- We will develop these techniques in coming lectures





Summary

- Many ways to view a rotation
  - Rotation matrix
  - Euler angles
  - Quaternions
  - Direction Cosines
  - Screw Vectors

- Homogenous transformations
  - Based on homogeneous coordinates

Cool Robotics Share