ARC Centre of Excellence for Autonomous Systems

MTRX 4700 : Experimental Robotics

Kinematics & Dynamics

Lecture 2

Stefan B. Williams

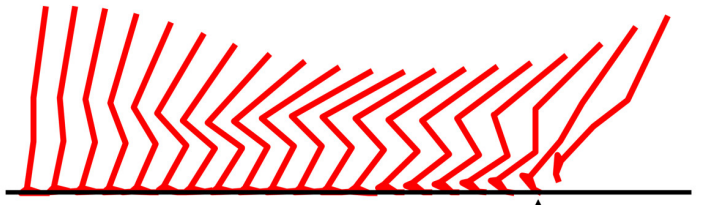
Course Outline

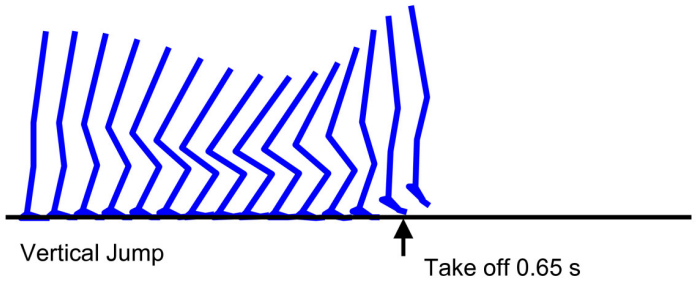
Week	Date	Content	Labs	Due Dates
1	5 Mar	Introduction, history & philosophy of robotics		
2	12 Mar	Robot kinematics & dynamics	Kinematics/Dynamics Lab	
3	19 Mar	Sensors, measurements and perception	"	
4	26 Mar	Robot vision and vision processing.	*No Tute (Good Friday)*	**Kinematics Lab**
	2 Apr	BREAK		
5	9 Apr	Localization and navigation	Sensing with lasers	
6	16 Apr	Estimation and Data Fusion	Sensing with vision	
7	23 Apr	Extra tutorial session (sensing)	Robot Navigation	**Sensing Lab**
8	30 Apr	Obstacle avoidance and path planning	Robot Navigation	
9	7 May	Extra tutorial session (nav demo)	Major project	**Navigation Lab**
10	14 May	Robotic architectures, multiple robot systems	"	
11	21 May	Robot learning	"	
12	28 May	Case Study	"	
13	4 June	Extra tutorial session (Major Project)	"	**Major Project**
14		Spare		

Kinematics

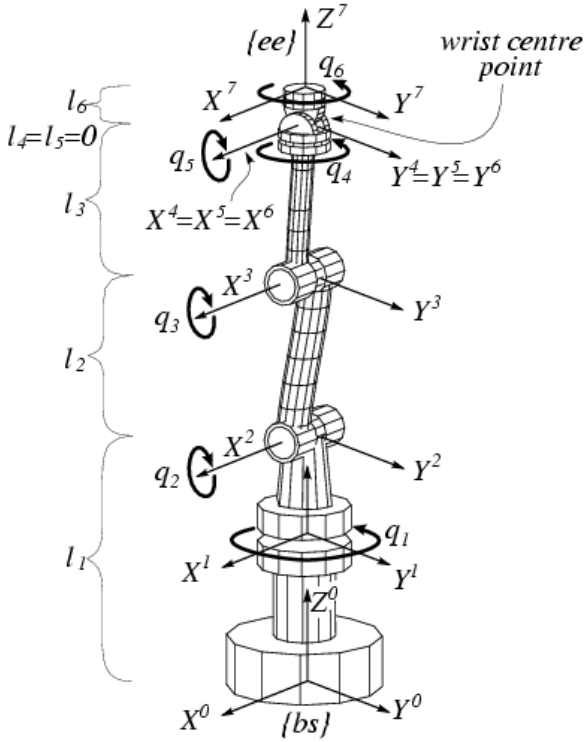
Ampère: “cinématique”, study of motion 1800's

- **Kinematics** describes the *motion* of objects *without* consideration of the *forces* that cause it
- aka “The Geometry of Motion”
- *How* objects move



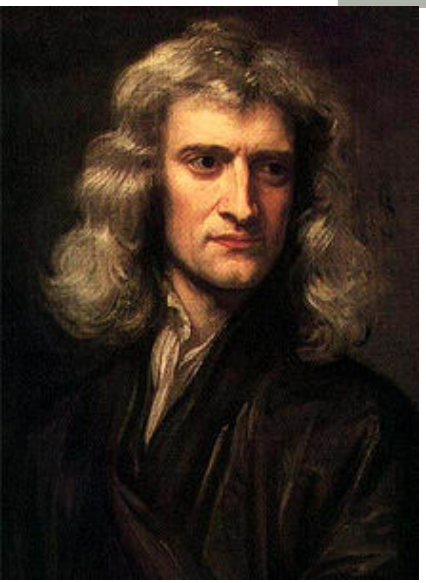


Kinematics of a human jumping



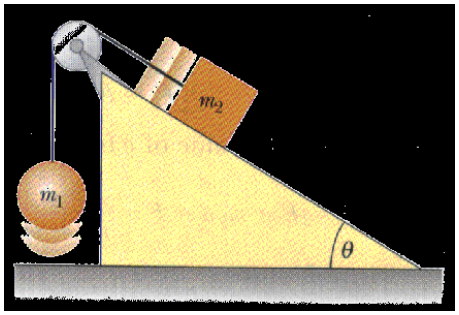
Kinematics of a serial robot arm

Dynamics



Newton:
F = ma
1600's

- **Dynamics** studies the *causes* of motion, *changes* in motion, *forces* and *torques*
- *Why* objects are in motion



Rigid body dynamics

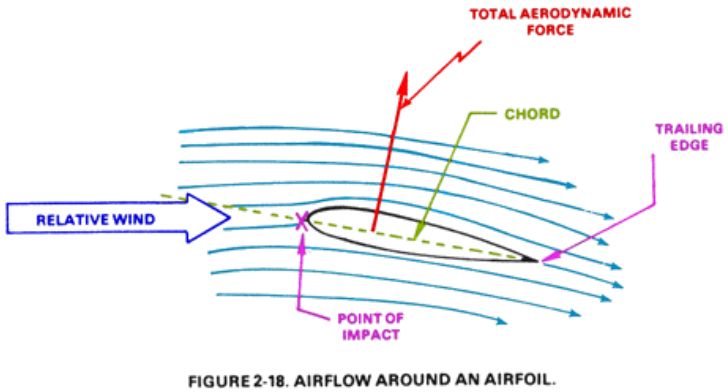
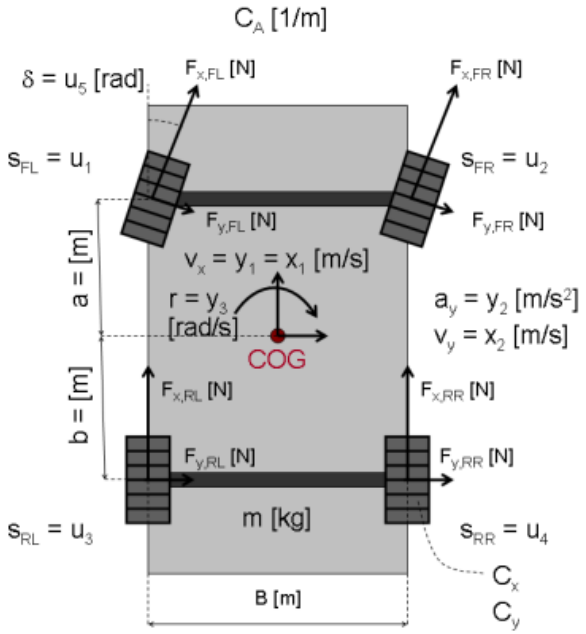


FIGURE 2-18. AIRFLOW AROUND AN AIRFOIL.

Aerodynamics



Vehicle Dynamics

Kinematics + Dynamics

Application video

Key Concepts

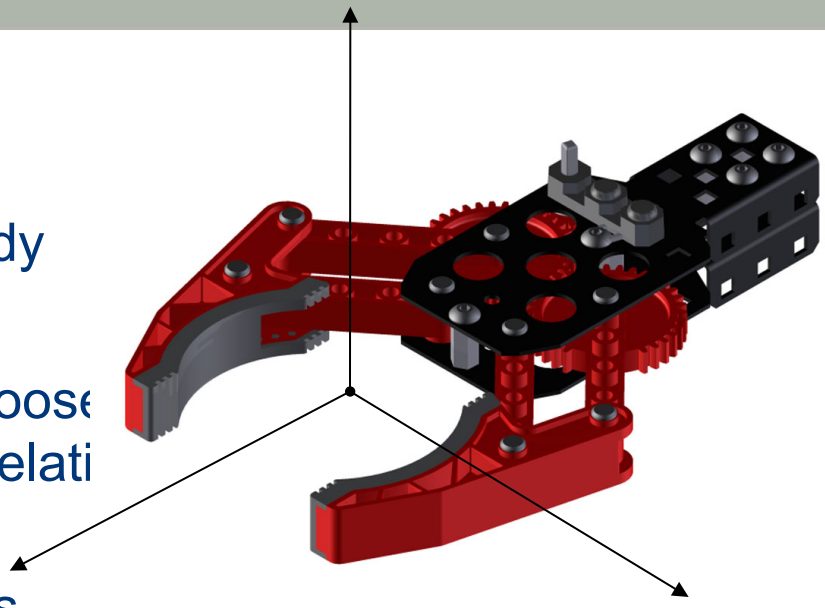
Coordinates and Transformations
- Frames, Coordinate Systems, Transformations
- Rotations: Matrices, Euler Angles, Fixed Angles
- Homogeneous Transformations

Kinematics
- Serial Chain Mechanisms (Arms)
- Denavit-Hartenberg Notation
- Forward, Inverse Kinematics
  - Workspace, Configuration Space
  - Singularities
- The Jacobian Matrix: Arm Velocities, Static Forces

Dynamics (time-permitting)
- Newton-Euler
- Lagrangian

Frames

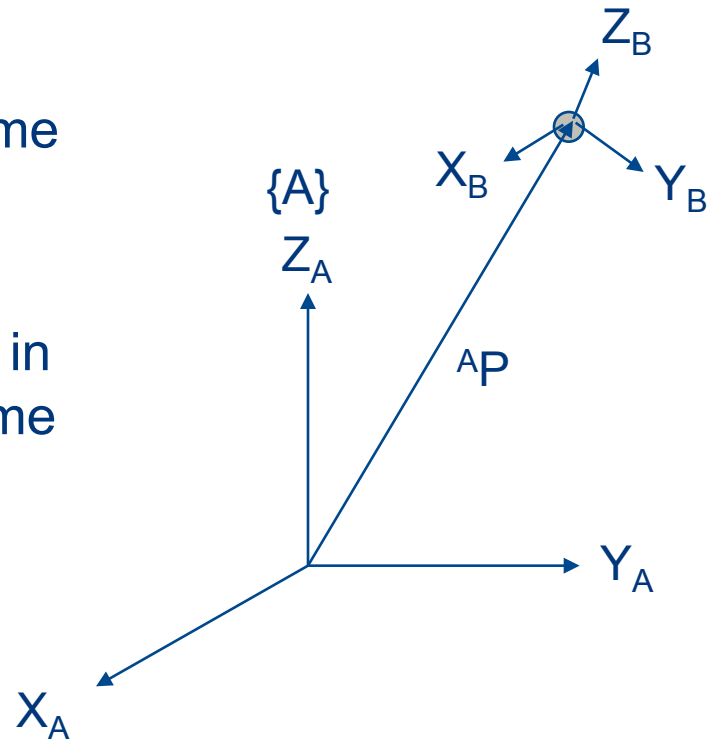
- We want to describe positions and orientations of bodies in space
- We attach a “frame” to each rigid body
- The frame follows the body
- There's some freedom in how we choose the frame's position and orientation relati to the body
  - DH notation partially standardises
- The combination of position and orientation of a frame is its “pose”



Coordinate Systems

Descartes: Cartesian Coordinates 1600's

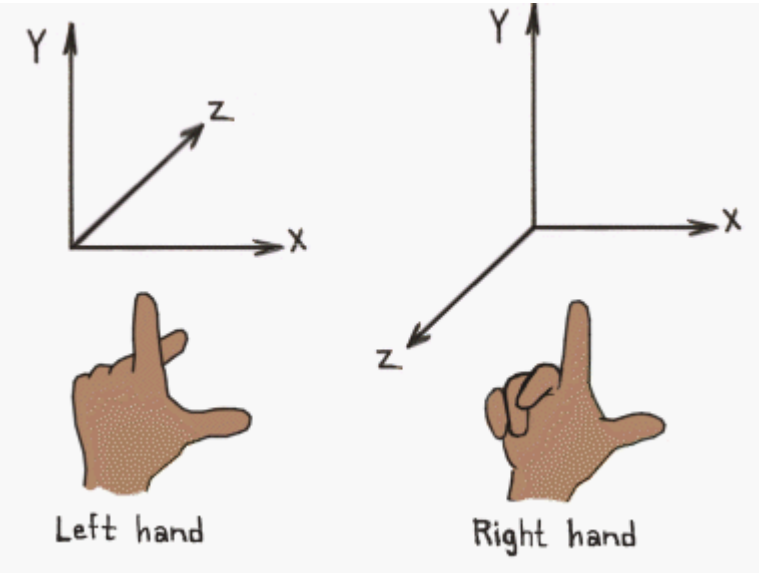
- A frame's pose only makes sense relative to another frame, e.g. hand pose relative to forearm pose...
- Everything is relative to some "global" frame
  - aka the "inertial frame"
  - There is some freedom in choosing the global frame

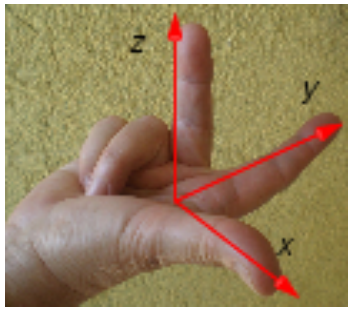


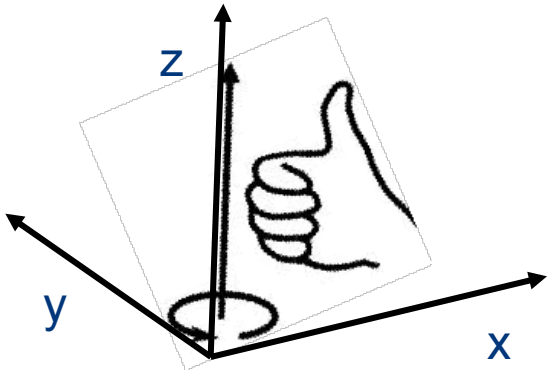
$$^A\mathbf{P} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$$

Right Handed Coordinates

- We are free to choose the handedness of our coordinate system
- By convention we always choose right-handed coordinates
- Be sure to use the correct hand!



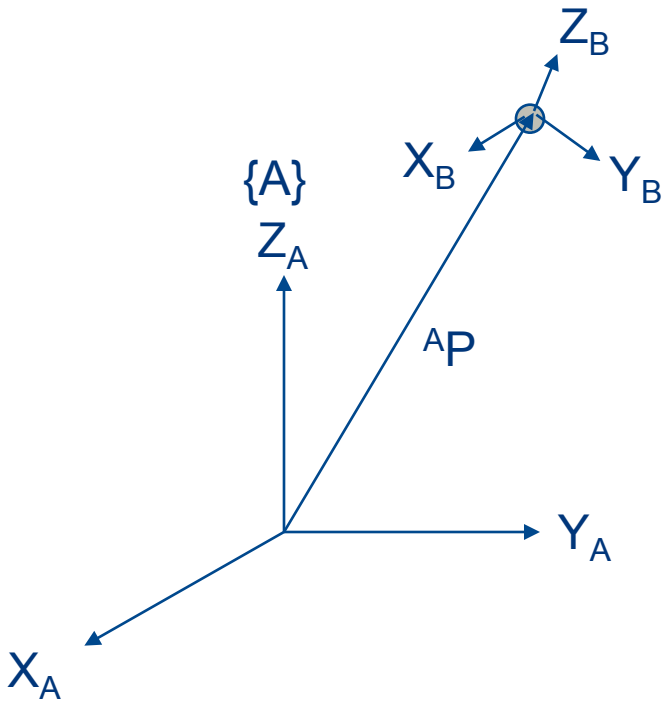




Position and Orientation

- Position of frame {B} relative to frame {A}

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



- Orientation of frame {B} relative to frame {A}
  - Many ways to express orientation
  - One way: rotation matrices
  - Project unit vectors B into reference frame {A} (dot products!)

$$ {}^{A}_{B}\boldsymbol{R} = \begin{bmatrix} {}^{A}\hat{X}_B & {}^{A}\hat{Y}_B & {}^{A}\hat{Z}_B \end{bmatrix} = \begin{bmatrix} \hat{X}_B \cdot \hat{X}_A & \hat{Y}_B \cdot \hat{X}_A & \hat{Z}_B \cdot \hat{X}_A \\ \hat{X}_B \cdot \hat{Y}_A & \hat{Y}_B \cdot \hat{Y}_A & \hat{Z}_B \cdot \hat{Y}_A \\ \hat{X}_B \cdot \hat{Z}_A & \hat{Y}_B \cdot \hat{Z}_A & \hat{Z}_B \cdot \hat{Z}_A \end{bmatrix} $$

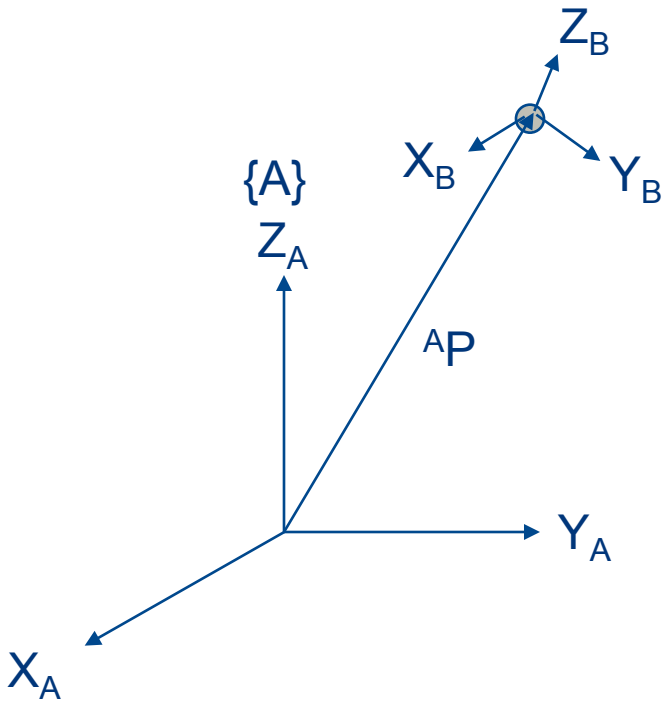
Rotation Matrices

- We can also find the orientation of frame {A} relative to frame {B}

$$ {}^{B}_{A}\boldsymbol{R} = \begin{bmatrix} \hat{X}_A \cdot \hat{X}_B & \hat{Y}_A \cdot \hat{X}_B & \hat{Z}_A \cdot \hat{X}_B \\ \hat{X}_A \cdot \hat{Y}_B & \hat{Y}_A \cdot \hat{Y}_B & \hat{Z}_A \cdot \hat{Y}_B \\ \hat{X}_A \cdot \hat{Z}_B & \hat{Y}_A \cdot \hat{Z}_B & \hat{Z}_A \cdot \hat{Z}_B \end{bmatrix} $$

- Compare

$$ {}^{A}_{B}\boldsymbol{R} = \begin{bmatrix} {}^{A}\hat{X}_B & {}^{A}\hat{Y}_B & {}^{A}\hat{Z}_B \end{bmatrix} = \begin{bmatrix} \hat{X}_B \cdot \hat{X}_A & \hat{Y}_B \cdot \hat{X}_A & \hat{Z}_B \cdot \hat{X}_A \\ \hat{X}_B \cdot \hat{Y}_A & \hat{Y}_B \cdot \hat{Y}_A & \hat{Z}_B \cdot \hat{Y}_A \\ \hat{X}_B \cdot \hat{Z}_A & \hat{Y}_B \cdot \hat{Z}_A & \hat{Z}_B \cdot \hat{Z}_A \end{bmatrix} $$



- We see that the inverse transformation equals the transpose – orthonormality (more later)

$$ {}^{B}_{A}\boldsymbol{R} = {}^{A}_{B}\boldsymbol{R}^{T} = {}^{A}_{B}\boldsymbol{R}^{-1} $$

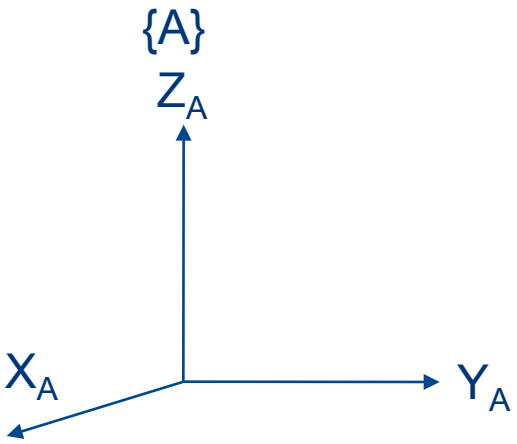
Rotation Matrices

- A rotation can also be constructed from cosines
- There are multiple ways to do this
- **Fixed Angles** (aka "Roll, Pitch, Yaw", or "R-P-Y")
  - X-Y-Z: First rotate about X_A, then Y_A, then Z_A
  - Order Matters!

$$
{}^{A}R_{BXYZ}(\gamma,\beta,\alpha) = R_Z(\alpha)R_Y(\beta)R_X(\gamma)
$$

$$
= \begin{bmatrix} c_\alpha & -s_\alpha & 0 \\ s_\alpha & c_\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_\beta & 0 & s_\beta \\ 0 & 1 & 0 \\ -s_\beta & 0 & c_\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\gamma & -s_\gamma \\ 0 & s_\gamma & c_\gamma \end{bmatrix}
$$

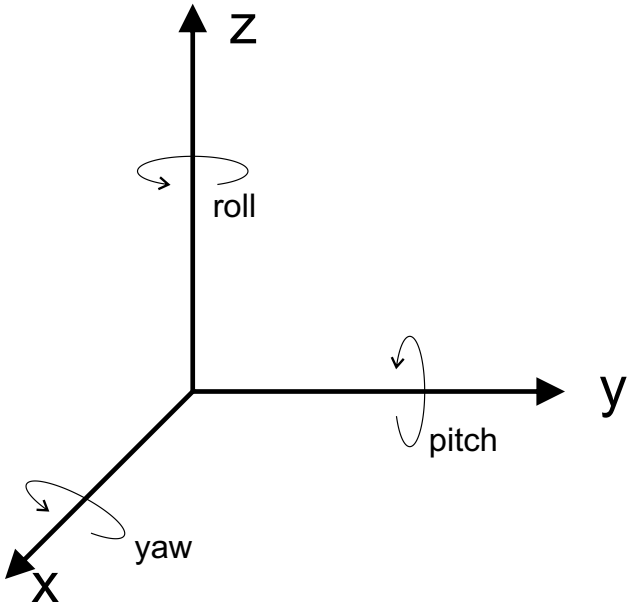
$$
= \begin{bmatrix} c_\alpha c_\beta & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma & c_\alpha s_\beta c_\gamma + s_\alpha s_\gamma \\ s_\alpha c_\beta & s_\alpha s_\beta s_\gamma + c_\alpha c_\gamma & s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma \\ -s_\beta & c_\beta s_\gamma & c_\beta c_\gamma \end{bmatrix}
$$



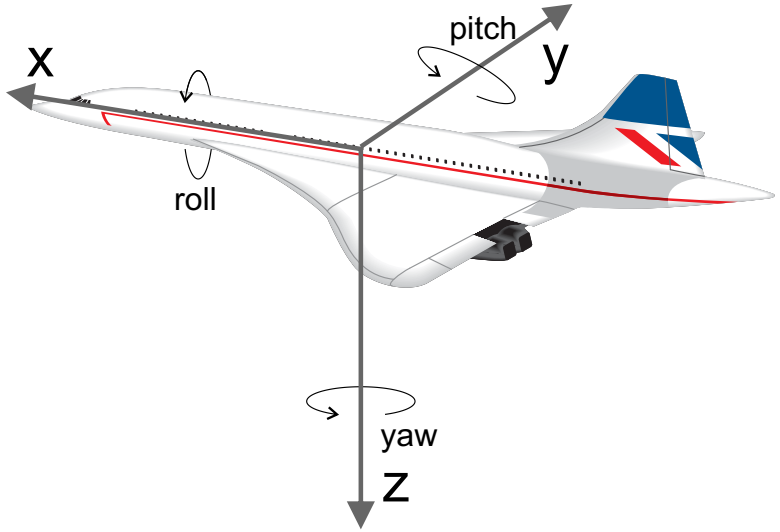
Rotation Matrices

- Roll, Pitch, Yaw (R-P-Y) Fixed Angles:
  - There are multiple standards, beware!

- In many Kinematics References:



In many Engineering Applications:



Rotation Matrices

- **Euler Angles**: rotations are about moving system {B}, not fixed system {A}
- Z-Y-X Euler:
  - Rotate first about Z_B, then
  - Rotate about (the new) Y_B, then
  - Rotate about (the new) X_B
- Each new rotation is about an axis determined by the previous rotation
- As an exercise: show that rotations about 3 axes of a fixed frame define the same final orientation as the same 3 rotations taken in the opposite order in the moving frame... a remarkable result!

Rotation Matrices

- Rotations are **orthonormal**
- Orthogonal
  - All columns are orthogonal (dot product = 0)
  - All rows are orthogonal
  - Preserves right angles
- Normal
  - All columns and rows have unit length
  - det(R) = 1
  - Preserves scale

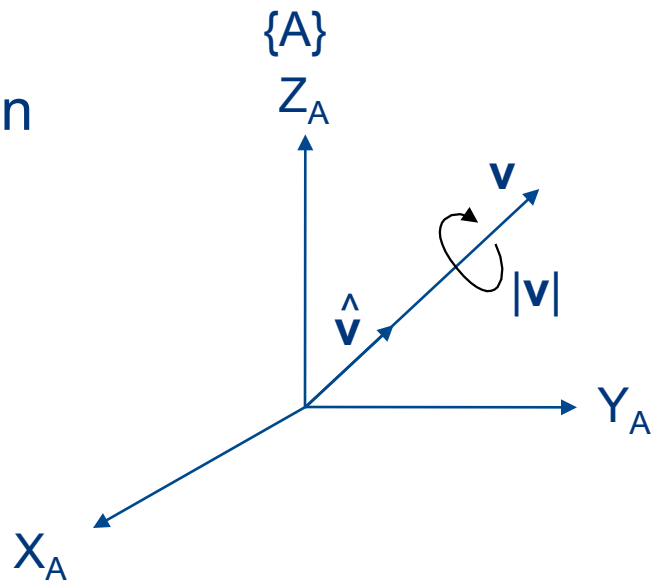
$$\boldsymbol{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \qquad \det(\boldsymbol{R}) = 1$$

$${}^{B}_{A}\boldsymbol{R} = {}^{A}_{B}\boldsymbol{R}^{T} = {}^{A}_{B}\boldsymbol{R}^{-1}$$

$$\begin{bmatrix} r_{11} \\ r_{21} \\ r_{31} \end{bmatrix} \cdot \begin{bmatrix} r_{12} \\ r_{22} \\ r_{32} \end{bmatrix} = 0 \qquad \begin{bmatrix} r_{11} \\ r_{21} \\ r_{31} \end{bmatrix} \cdot \begin{bmatrix} r_{11} \\ r_{21} \\ r_{31} \end{bmatrix} = 1$$

Still More Rotations

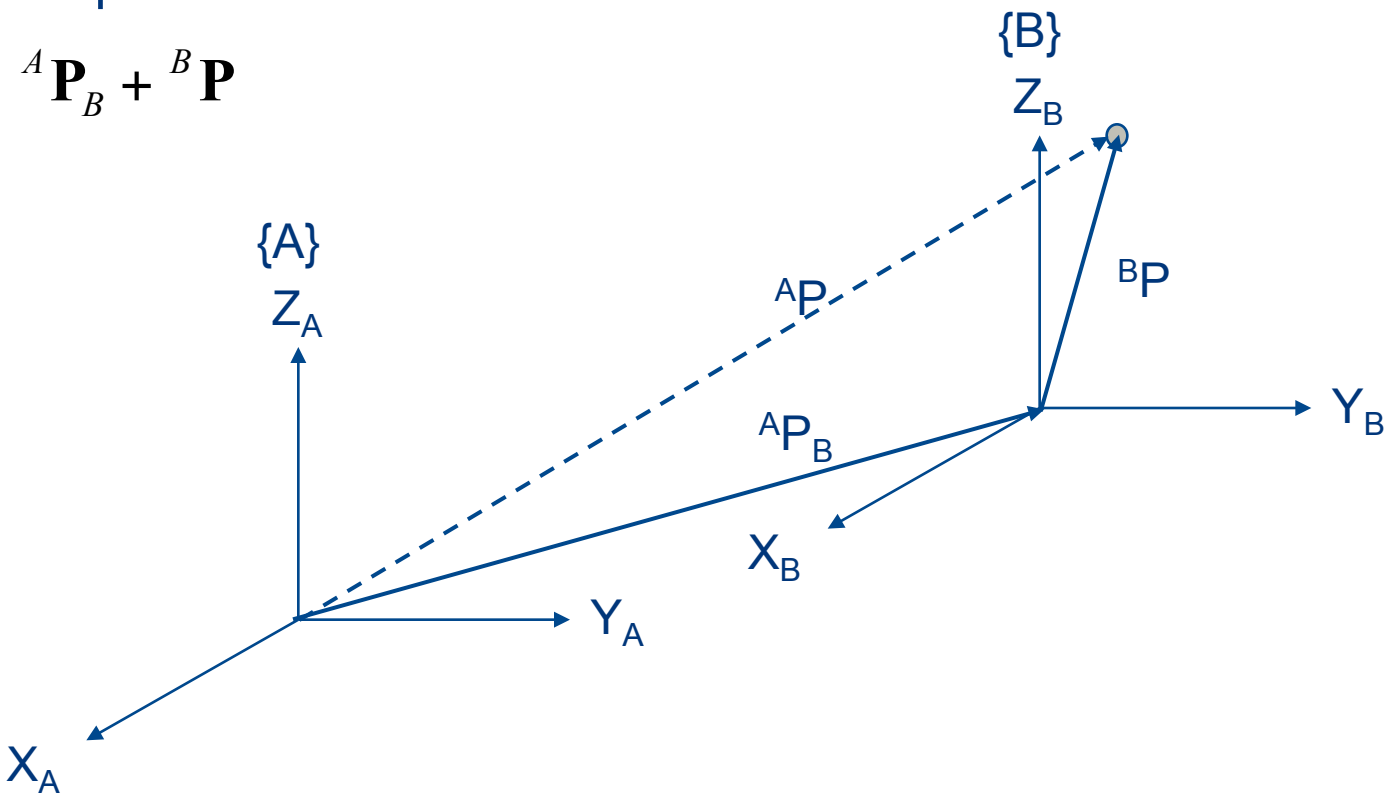
- A rotation matrix contains 9 elements...
- But a rotation is a 3-DOF operation... ?
  - → Rotation matrices are redundant
  - Non-orthonormal matrices are “wasted space”
- More compact form: the rotation vector
  - Vector of 3 numbers
  - Magnitude = magnitude of rotation
  - Direction = axis of rotation
- Also of note: unit quaternions
  - Vector of 4 numbers
  - No singularities (“gimbal lock”)
  - Elegant properties



Coordinate Transformations

- Express the vector or point P, fixed in frame {B}, in terms of frame {A}
- If {B} is translated wrt to {A} without rotation, we can sum position vectors

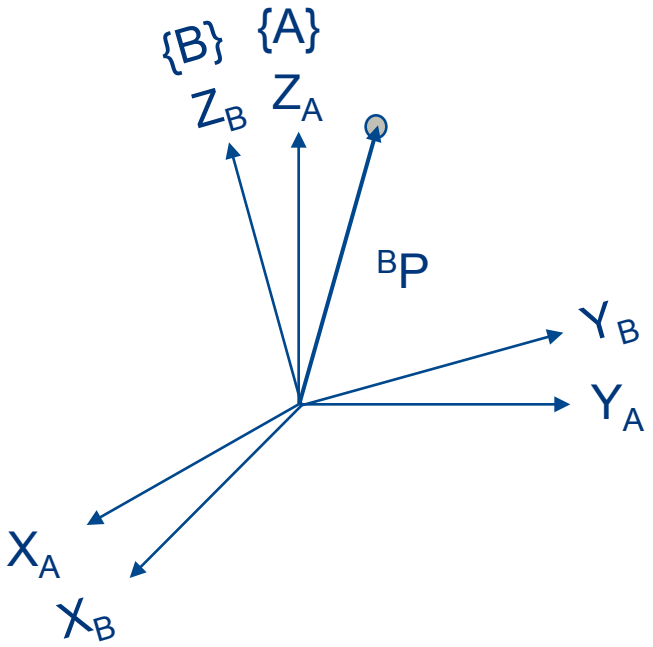
$$ {}^{A}\mathbf{P} = {}^{A}\mathbf{P}_{B} + {}^{B}\mathbf{P} $$



Coordinate Transformations

- If {B} is rotated wrt to {A} without translation, we can apply a rotation

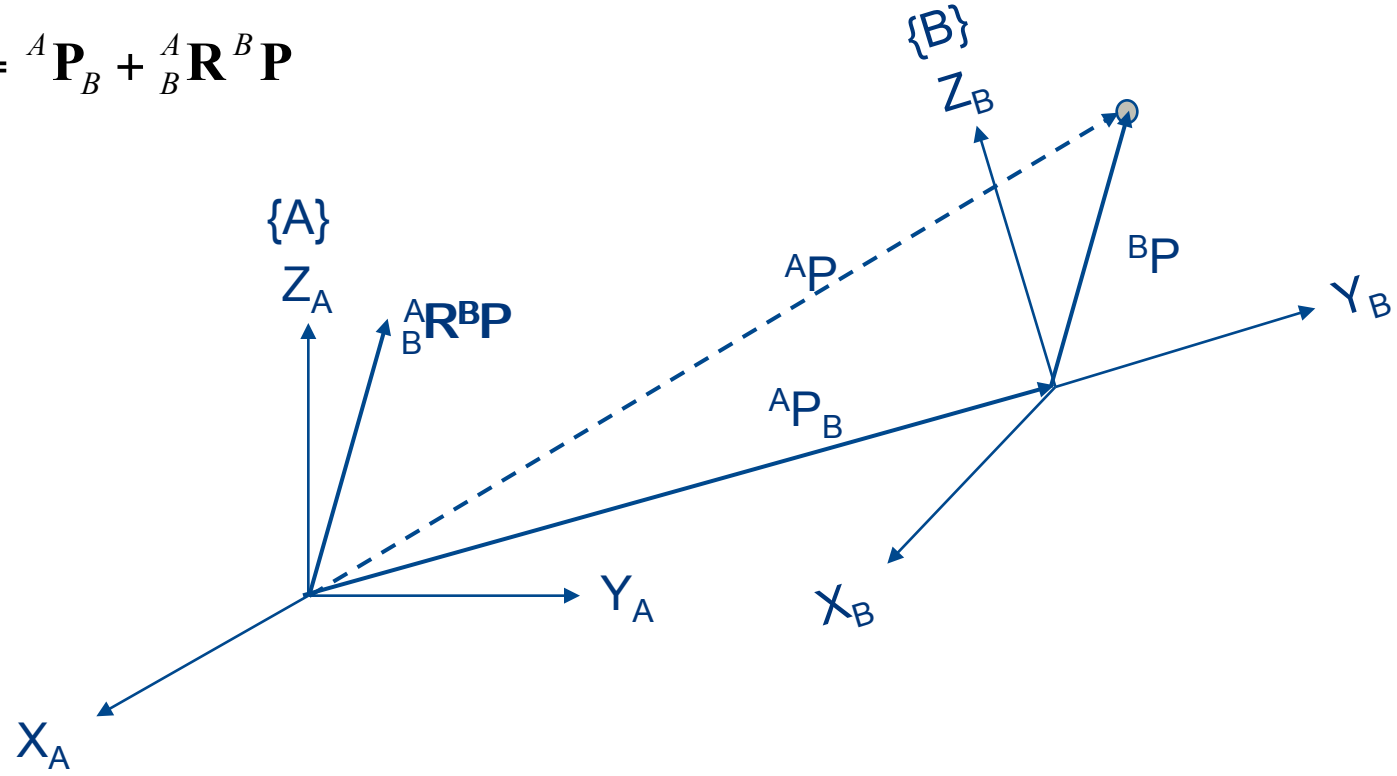
$$ {}^{A}\mathbf{P} = {}^{A}_{B}\mathbf{R}\,{}^{B}\mathbf{P} $$



Coordinate Transformations

- If {B} is rotated and translated wrt to {A}, we must account for the composite transformation
  - Watch out for order!

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



Homogeneous Transformations

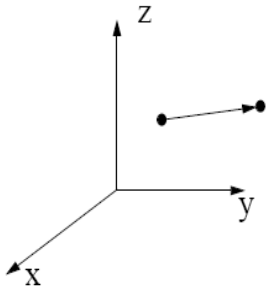
- A compact representation of the translation and rotation is known as the **Homogeneous Transformation**
- This allows us to cast the rotation and translation of the general transform in a single matrix form

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

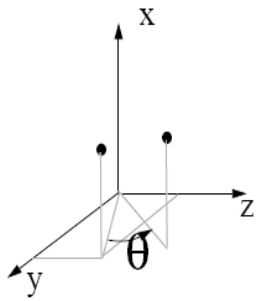
$$ \begin{bmatrix} {}^{A}\mathbf{P} \\ 1 \end{bmatrix} = {}^{A}_{B}\mathbf{T} \begin{bmatrix} {}^{B}\mathbf{P} \\ 1 \end{bmatrix} $$

Homogeneous Transformations

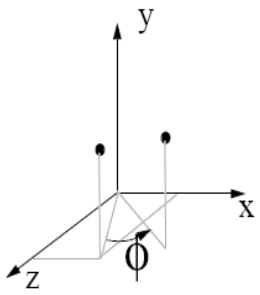
- Fundamental orthonormal transformations that can be represented in this form



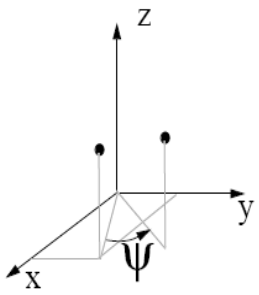
$$Trans(u, v, w) = \begin{bmatrix} 1 & 0 & 0 & u \\ 0 & 1 & 0 & v \\ 0 & 0 & 1 & w \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



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



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



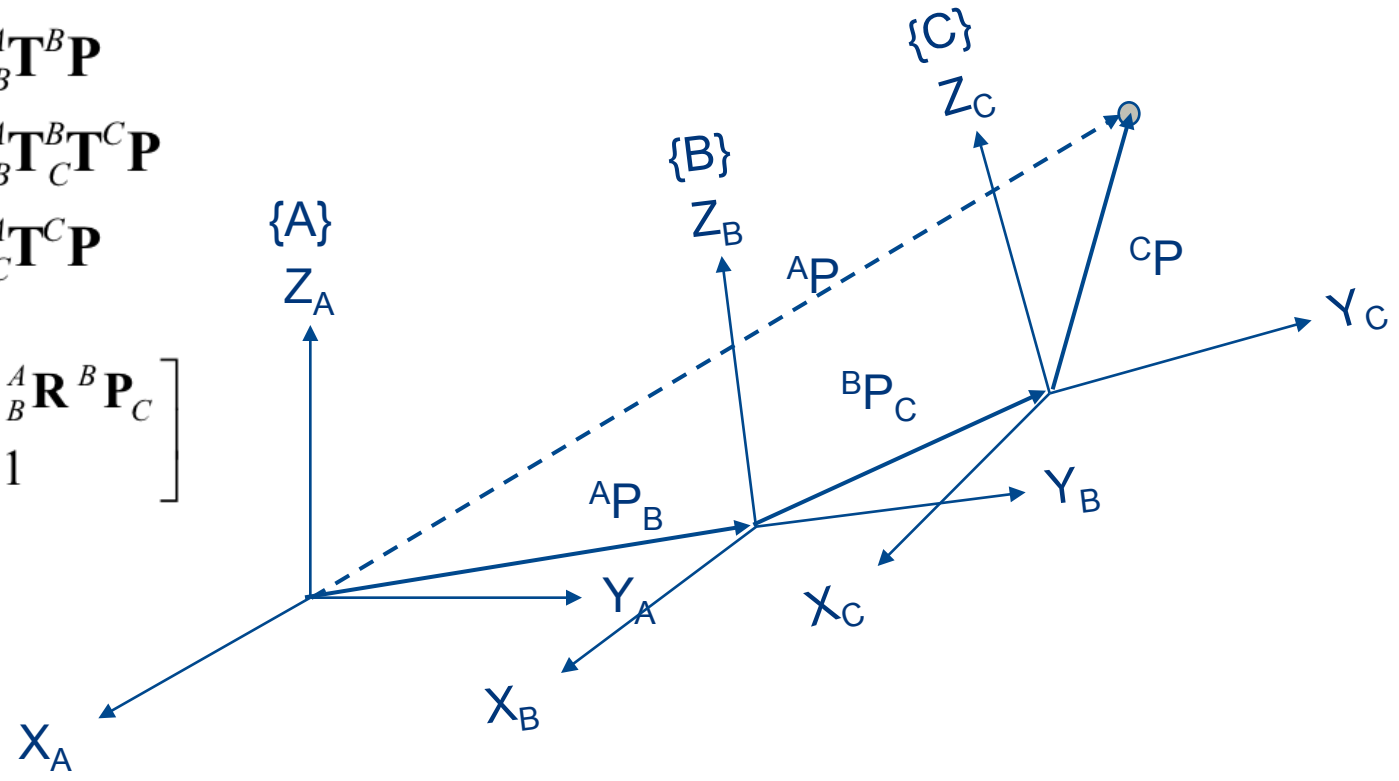
$$Rotz(\psi) = \begin{bmatrix} c\psi & -s\psi & 0 & 0 \\ s\psi & c\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Homogeneous Transformations

- Coordinate transforms can be compounded
  - → Homogeneous transform as "pose"

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

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



Kinematics

- Kinematics is the study of motion without regard to the forces which cause it
- The kinematics of manipulators involves the study of the geometric and time based properties of the motion, and in particular how the various links move with respect to one another and with time

Kinematics

- Kinematic modelling is one of the most important analytical tools of robotics.
- Used for modelling mechanisms, actuators and sensors
- Used for on-line control and off-line programming and simulation
- In mobile robots kinematic models are used for:
  - steering (control, simulation)
  - perception (image formation)
  - sensor head and communication antenna pointing
  - world modelling (maps, object models)
  - terrain following (control feedforward)
  - gait control of legged vehicles

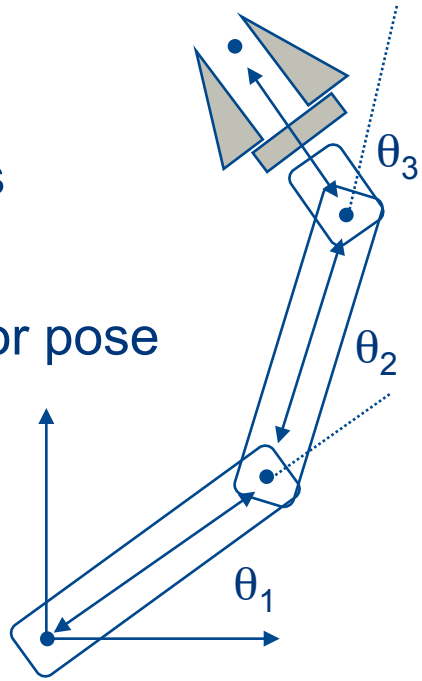
Kinematics

The three kinematics problems we'll discuss:

1. Describe a serial linkage (robot arm)
   - DH notation

For a particular arm:

1. Find end effector pose, given joint variables
   - “Forward Kinematics”
2. Find joint variables for a desired end effector pose
   - “Inverse Kinematics”



Forward Kinematics

- Forward kinematics is the process of chaining homogeneous transforms together in order to represent:
  - the articulations of a mechanism, or
  - the fixed transformation between two frames which is known in terms of linear and rotary parameters.
- In this process, the joint variables are given, and the problem is to find the poses of the end effector (and links in between).

$$\theta_1, \theta_2, \theta_3, \ldots \quad \rightarrow \quad {}^{A}_{B}\mathbf{T}$$

Joint Variables: "Configuration Space"

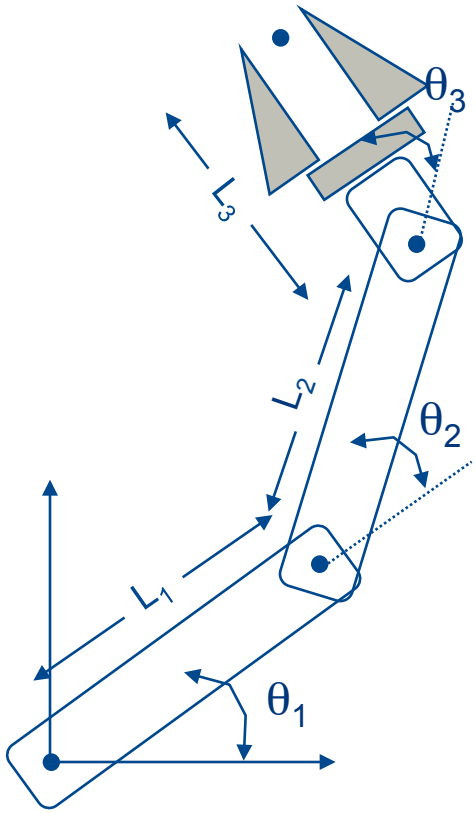
End Effector Poses: "Workspace"

Forward Kinematics

- Consider the schematic of the planar manipulator shown
- Given the joint angles and link lengths, we can determine the end effector pose

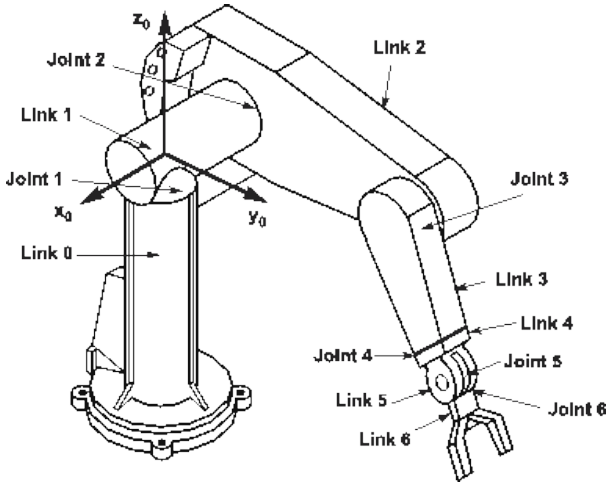
$$x = L_1 \cos\theta_1 + L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3)$$
$$y = L_1 \sin\theta_1 + L_2 \sin(\theta_1 + \theta_2) + L_3 \sin(\theta_1 + \theta_2 + \theta_3)$$

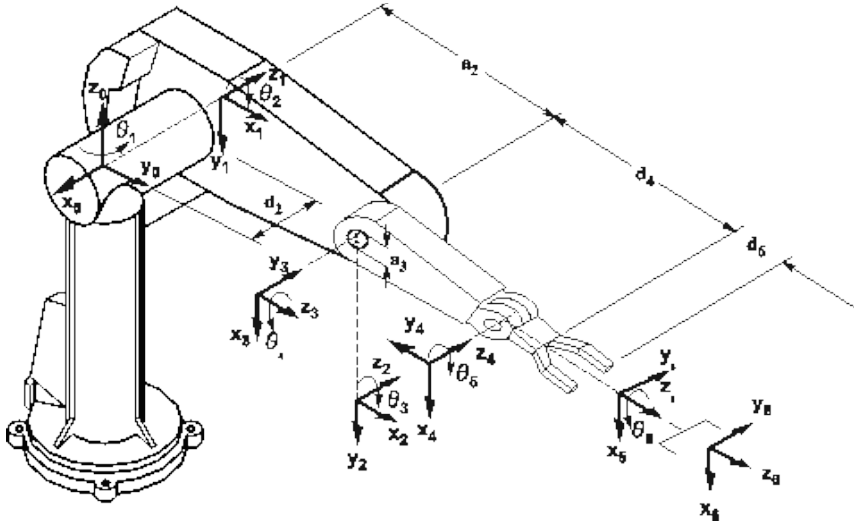
- This isn't too difficult to determine for a simple, planar manipulator



Forward Kinematics

- What about a more complicated mechanism?





$$ {}^0T_6 = {}^0T_1\,{}^1T_2\,{}^2T_3\,{}^3T_4\,{}^4T_5\,{}^5T_6 = \begin{pmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

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

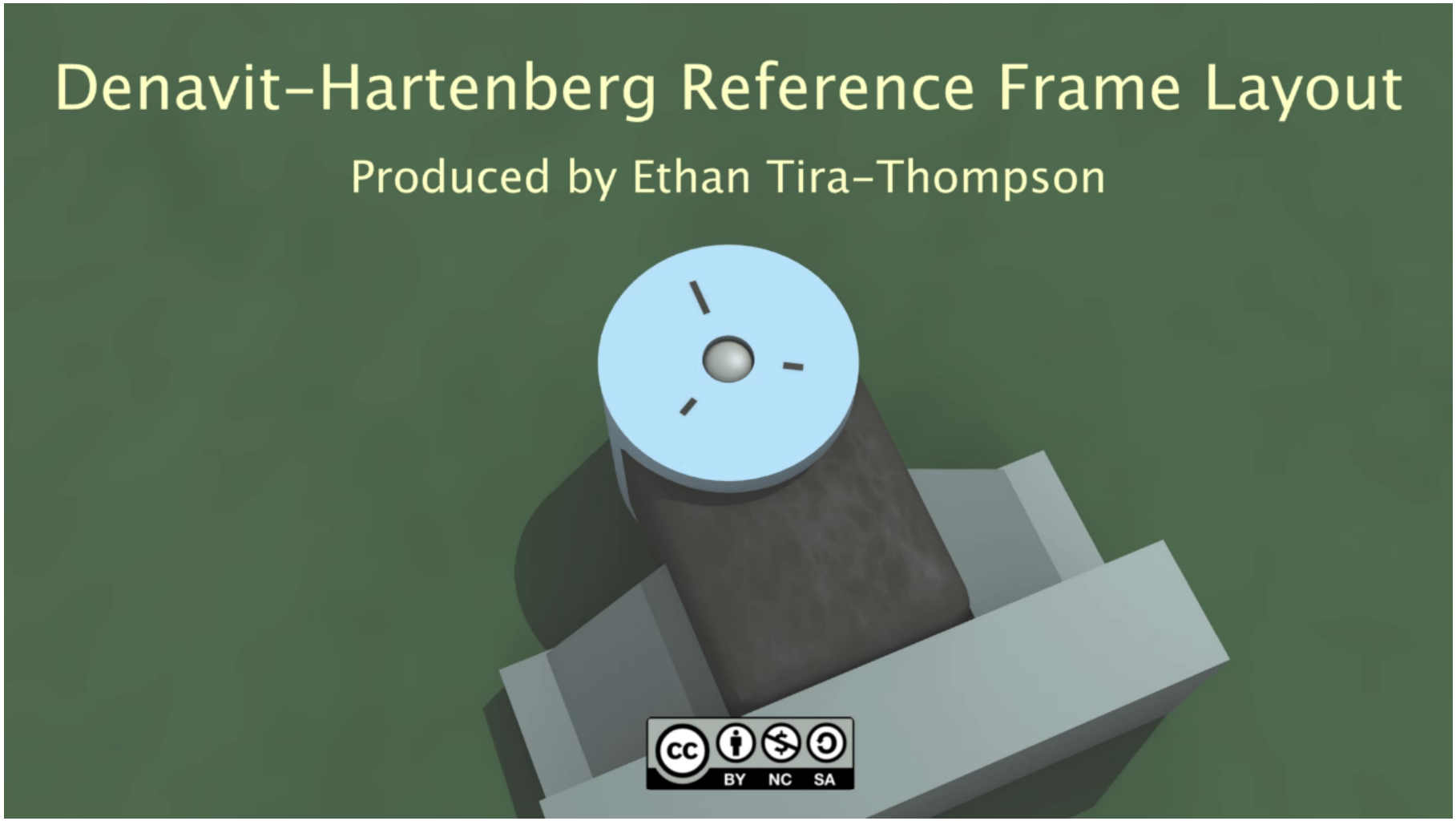
Forward Kinematics

- We require a systematic manner for modelling complex mechanisms
- The conventional rules for modelling a sequence of connected joints are as follows:
  - a link may be considered as a rigid body defining the relationship between two neighbouring joint axes
  - assign embedded frames to the links in sequence such that the transformations which move each frame into coincidence with the next are a function of the appropriate joint variable
  - write the frame transformations in left-to-right order, moving along the arm
- This process will generate a matrix that represents the position of the last embedded frame with respect to the first, or equivalently, which converts the coordinates of a point from the last to the first. This matrix will be called the mechanism model.

Forward Kinematics

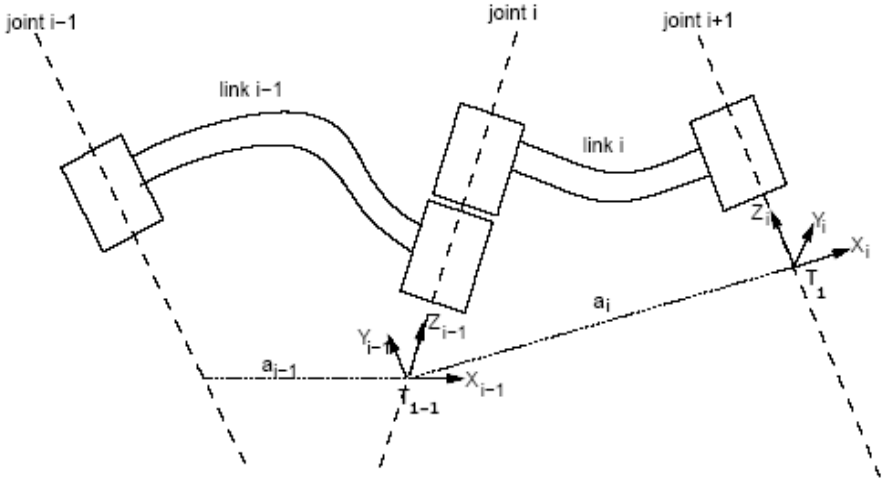
- In 1955, J. Denavit and R. S. Hartenberg first proposed the use of homogeneous transforms to represent the articulations of a mechanism
- “Denavit-Hartenberg (DH) notation”
- Assign a frame to each link
  - Z aligns with axis of rotation or along a prismatic joint
  - X points from one Z axis to the next
    - “Common normal”
- For two frames positioned in space, the first can be moved into coincidence with the second by a sequence of 4 operations...

Denavit-Hartenberg Convention

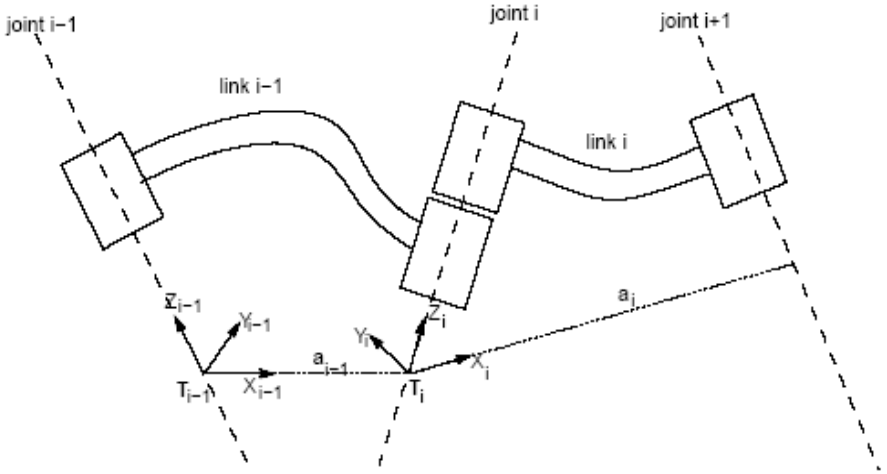


http://tekkotsu.no-ip.org/movie/dh-hd.mp4

Denavit-Hartenberg Convention



(a) Standard form



(b) Modified form

- 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.
- 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;

Denavit-Hartenberg Convention

- DH:
  - translate along z_{i-1} axis by a distance d_i
  - rotate around the z_{i-1} axis by an angle θ_i
  - translate along the new x axis by a distance a_i
  - rotate around the new x axis by an angle α_i
- Modified DH:
  - translate along the x_{i-1} axis by a distance a_i
  - rotate around the x_{i-1} axis by an angle α_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

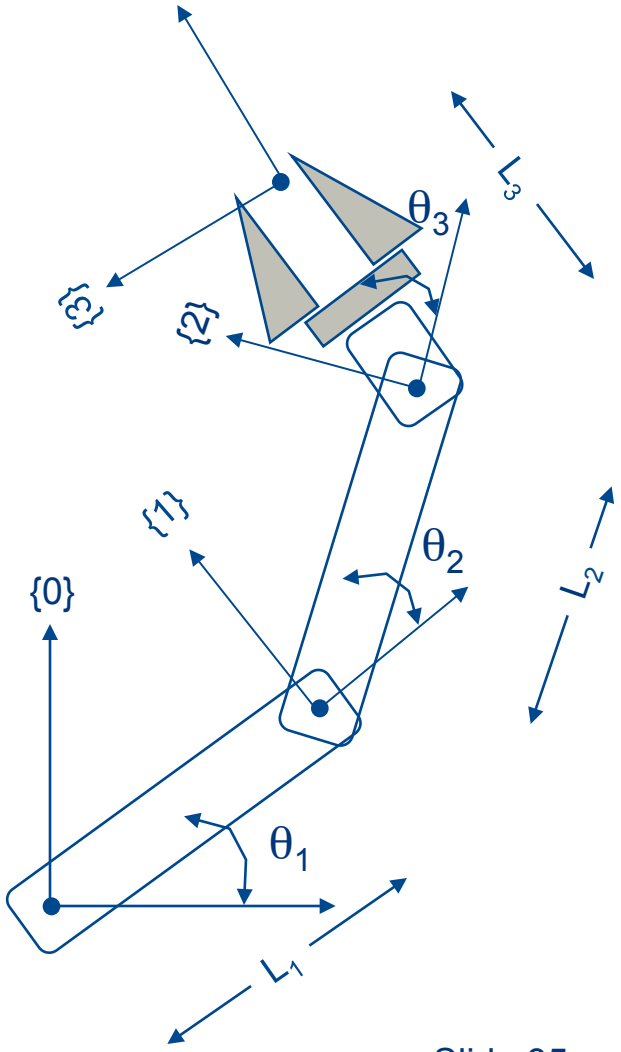
- Using this technique, each homogeneous transformation is represented as a product of four “basic” transformations

$$
\begin{aligned}
{}^{i-1}A_i &= Rot_{z,\theta_i} Trans_{z,d_i} Trans_{x,a_i} Rot_{x,\alpha_i} \\
&= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} c_{\alpha_i} & s_{\theta_i} s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i} c_{\alpha_i} & -c_{\theta_i} s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

Example: 3 Link Manipulator

- We assign frames at the joints, and write a table of standard DH parameters

Link	d_i	θ_i	a_i	α_i
1	0	θ_1	L_1	0
2	0	θ_2	L_2	0
3	0	θ_3	L_3	0



Standard DH in Mathematical order (right-to-left):

$\text{Rot}[x_i,\alpha_i]\ \text{Trans}[x_i,a_i]\ \text{Rot}[z_{i-1},\theta_i]\ \text{Trans}[z_{i-1},d_i]$

Example: 3 Link Manipulator

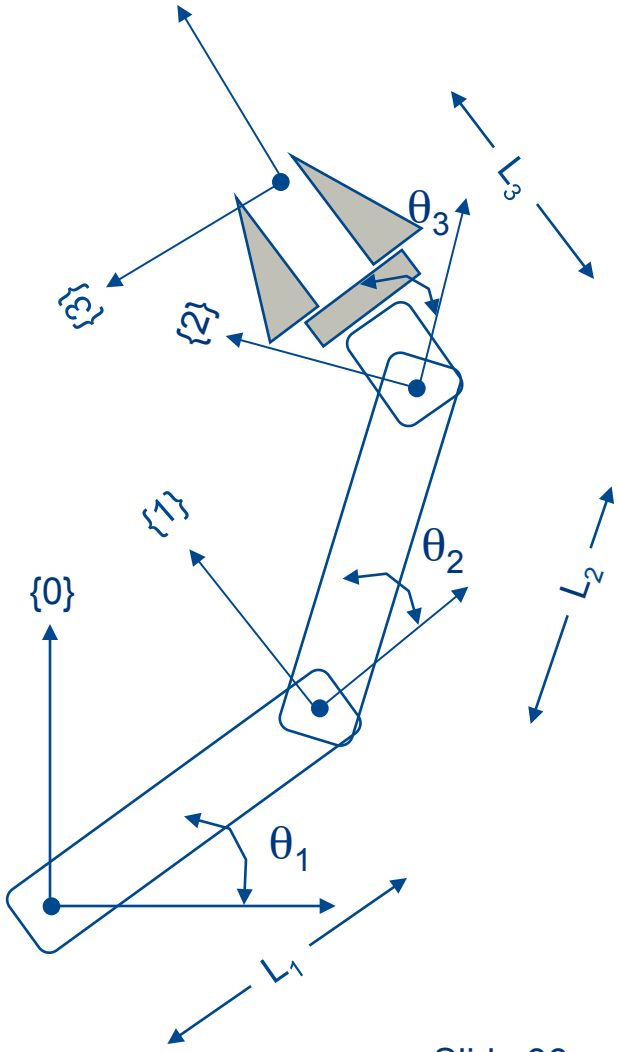
- The table translates easily into a forward-kinematics solution

Link	d_i	θ_i	a_i	α_i
1	0	θ_1	L_1	0
2	0	θ_2	L_2	0
3	0	θ_3	L_3	0

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

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

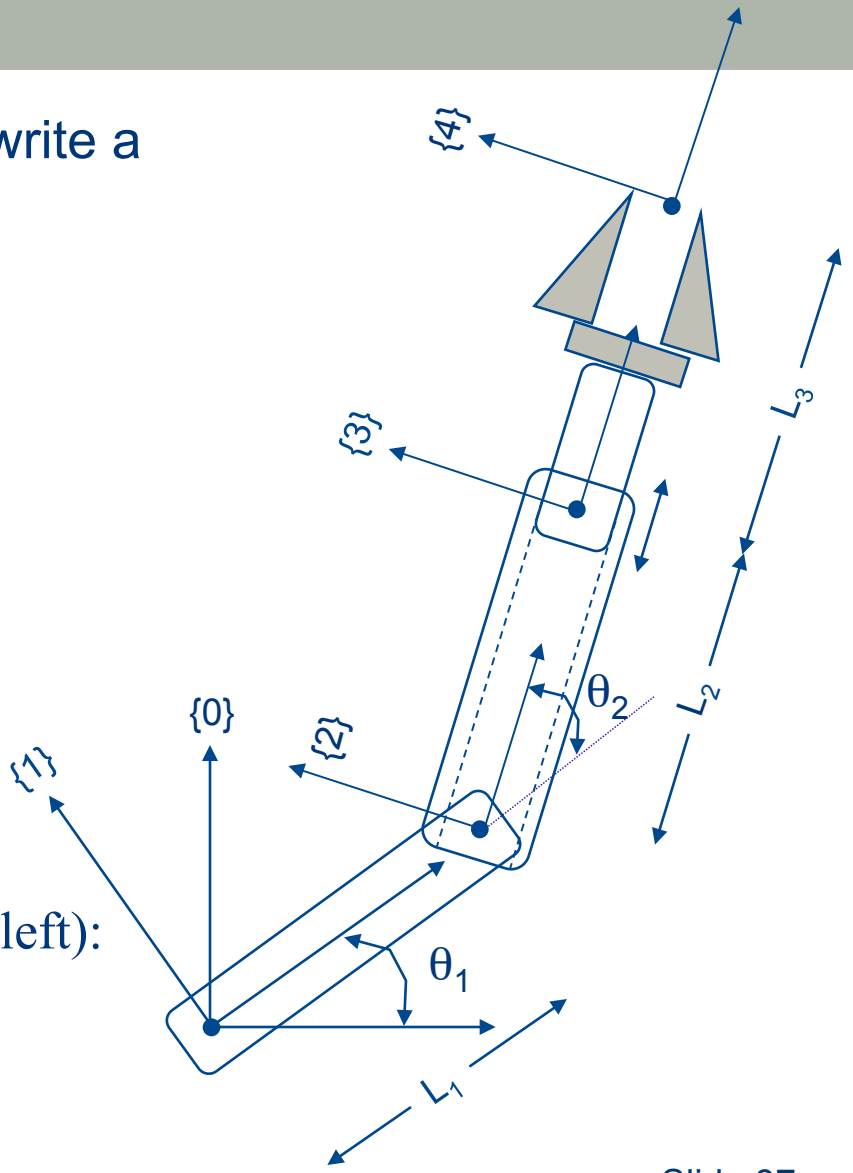
$$ = \begin{bmatrix} c_{\theta_{123}} & -s_{\theta_{123}} & 0 & L_1 c_{\theta_1} + L_2 c_{\theta_{12}} + L_3 c_{\theta_{123}} \\ s_{\theta_{123}} & c_{\theta_{123}} & 0 & L_1 s_{\theta_1} + L_2 s_{\theta_{12}} + L_3 s_{\theta_{123}} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$



Example: 2 Rot 1 Linear Link Manipulator

- We assign frames at the joints, and write a table of modified DH parameters

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



Modified DH in Mathematical order (right-to-left):

$\text{Rot}[z_i,\theta_i]\ \text{Trans}[z_i,d_i]\ \text{Rot}[x_{i-1},\alpha_i]\ \text{Trans}[x_{i-1},a_i]$

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

$$\theta_1, \theta_2, \theta_3, \ldots \quad \leftarrow \quad {}^{A}_{B}\mathbf{T}$$

Joint Variables: "Configuration Space"

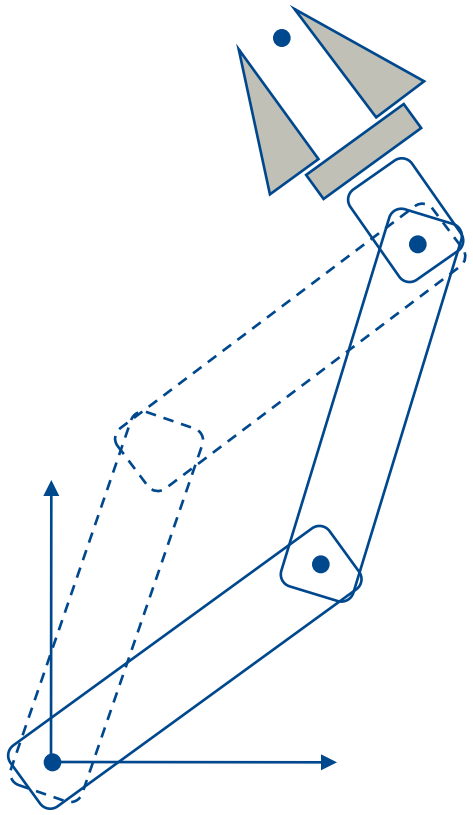
End Effector Pose: "Workspace"

Inverse Kinematics

- Unlike with systems of linear equations, there are no general algorithms that may be employed to solve a set of nonlinear equations
- **Closed-form** and **numerical** methods exist
- We will concentrate on analytical, closed-form methods
- These can be characterised by two methods of obtaining a solution: **algebraic** and **geometric**

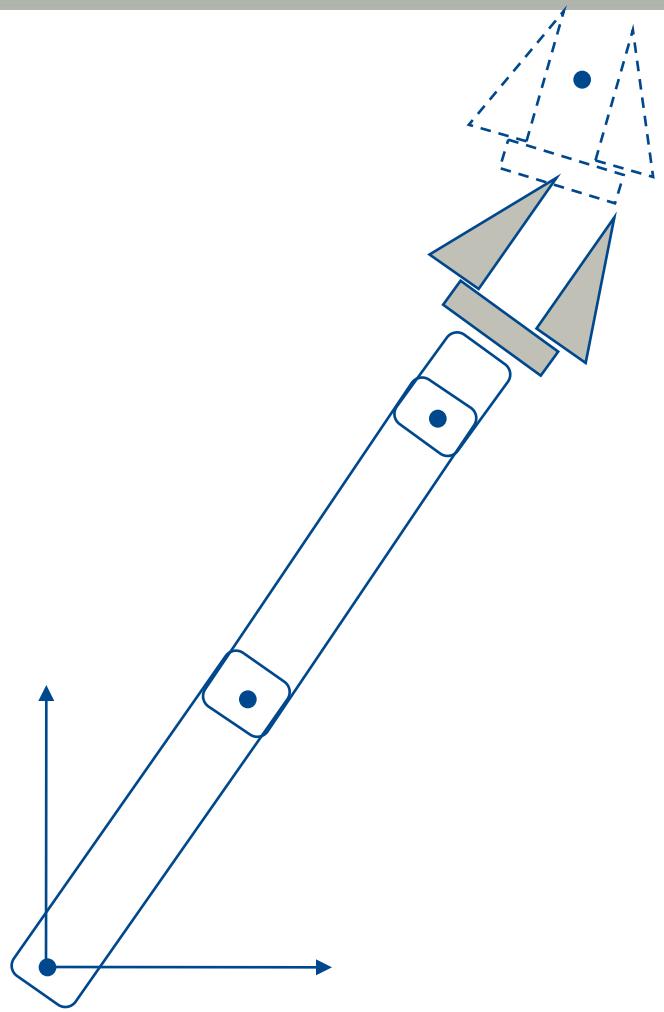
Inverse Kinematics: Multiple Solutions

- 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 can be a function of proximity to the current pose, limits on the joint angles and possible obstructions in the workspace



Inverse Kinematics: Singularities

- An understanding of the workspace of the manipulator is important: some poses are not achievable (no IK solution)
- These are called singular positions of the manipulator
- The workspace boundary is a singularity
- Workspace-interior singularities also occur whenever the manipulator loses a degree of freedom in Cartesian space.
- This typically happens when joints are aligned: there is a direction in which the manipulator cannot move regardless of joint rates.



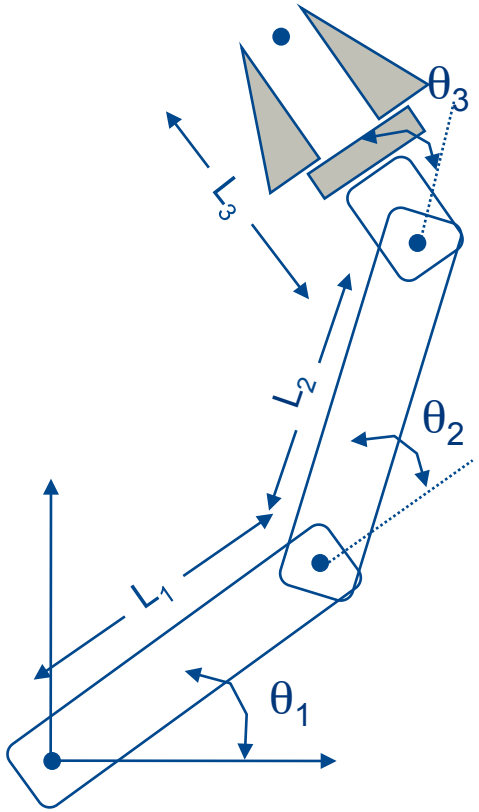
Inverse Kinematics: Algebraic Approach

- From Forward Kinematics we have a series of equations which define this system

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

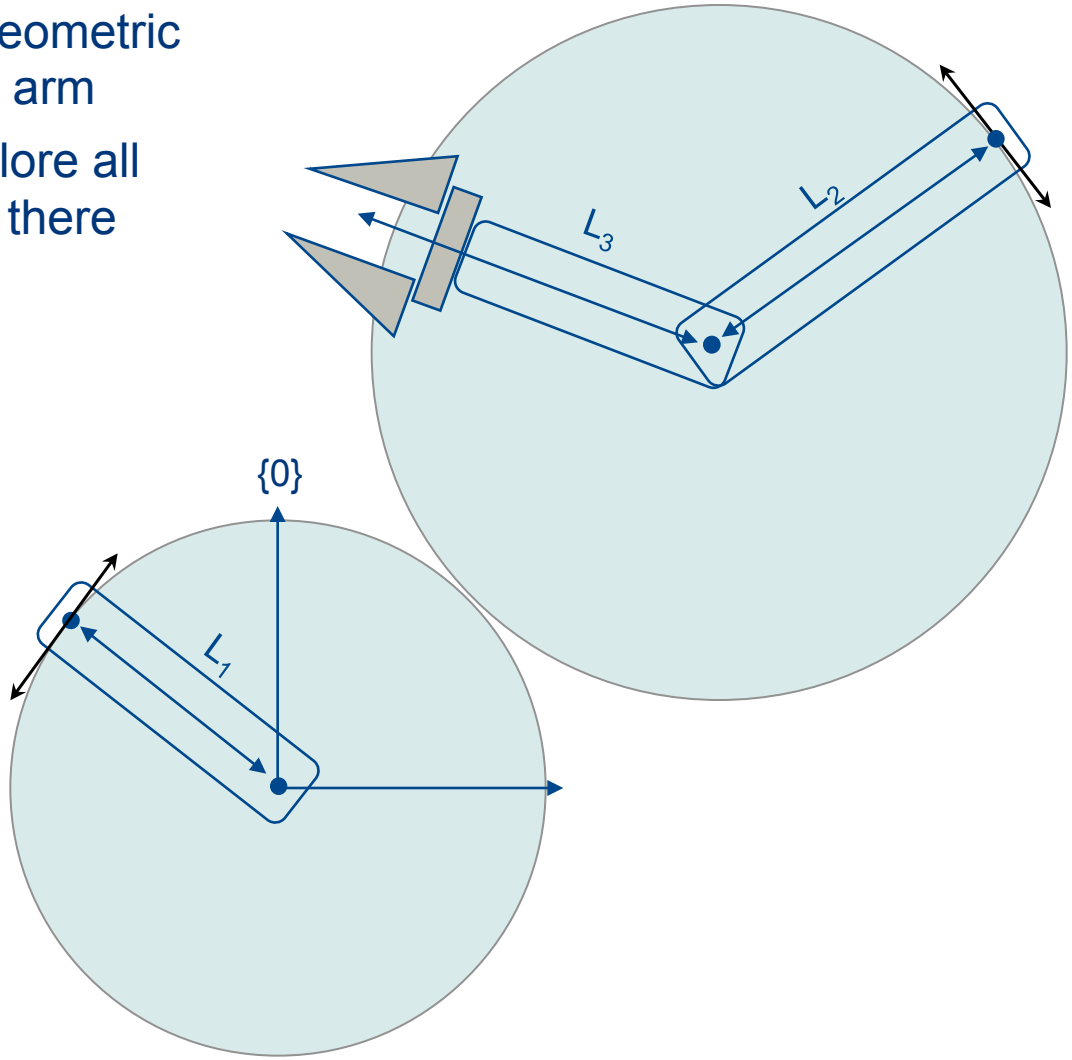
$$ {}^0T_3 = \begin{bmatrix} c_\phi & -s_\phi & 0 & x \\ s_\phi & c_\phi & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

- Eq … set of alg … ich we solve for the joint angles



Inverse Kinematics: Geometric Approach

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

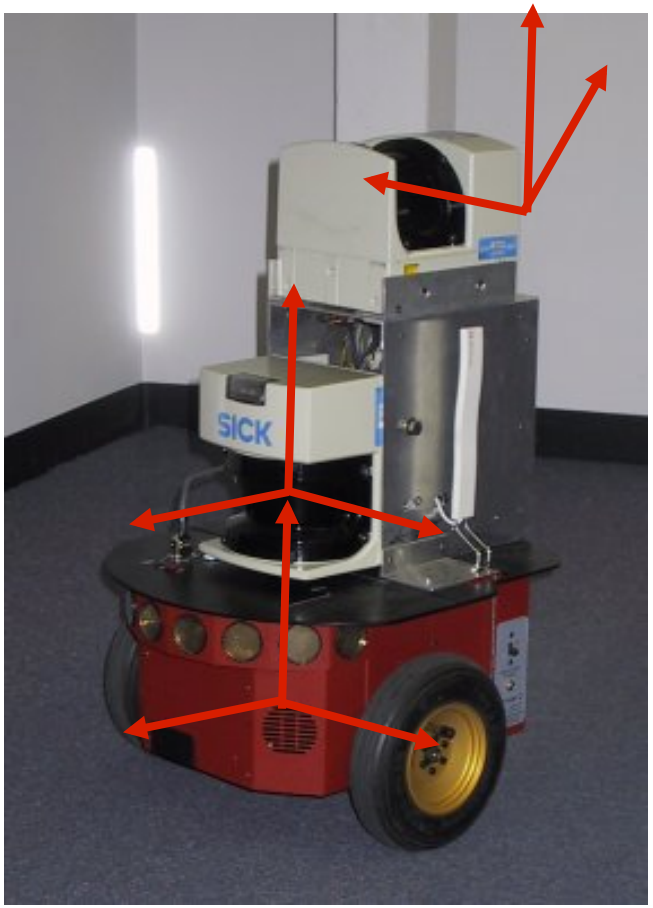


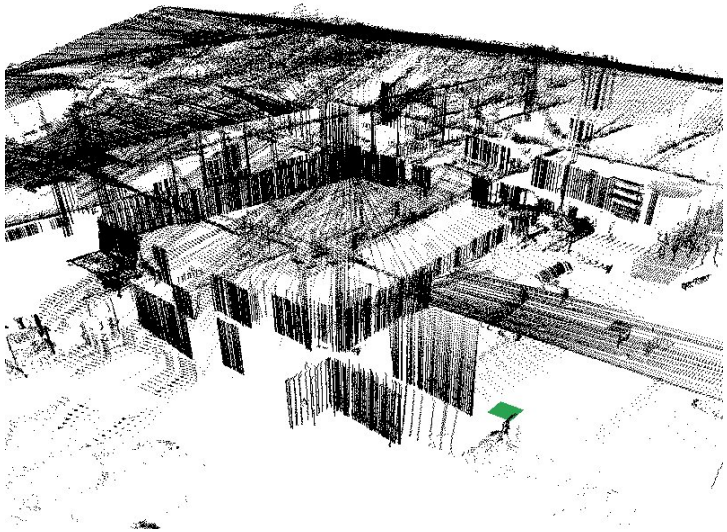
Kinematics for Mobile Platforms

- 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

Kinematics for Mobile Platforms

- 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





Kinematics: Velocity

- Now what if things start to move? We'd like to describe the motion of moving frames.
- In general, velocity is dependent on the frame from which it is observed. It will depend on the relative velocity of the frames, the velocity within the frame transformed into the global frame and the relative rotational velocities of the frames.

Kinematics: Velocity

- Recall a vector or point fixed in {B} can be expressed relative to {A} as

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

- Differentiating wrt to time we find the velocity of P expressed in {A}

$$ {}^{A}\mathbf{V}_{P} = \frac{d}{dt}\,{}^{A}\mathbf{P} = \lim_{\Delta t \to 0} \frac{{}^{A}\mathbf{P}(t+\Delta t) - {}^{A}\mathbf{P}(t)}{\Delta t} $$

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

Which can expressed as:

$$ {}^{A}\mathbf{V}_{P} = {}^{A}\mathbf{V}_{BORG} + {}^{A}\mathbf{R}_{B}\,{}^{B}\mathbf{V}_{P} + {}^{A}\Omega_{B} \times {}^{A}\mathbf{R}_{B}\,{}^{B}\mathbf{P} $$

Velocity of the origin of frame {B}, expressed in terms of frame {A}

Angular velocity vector: describes rotation of {B} relative to {A}; magnitude = *rate*

Kinematics: Velocity

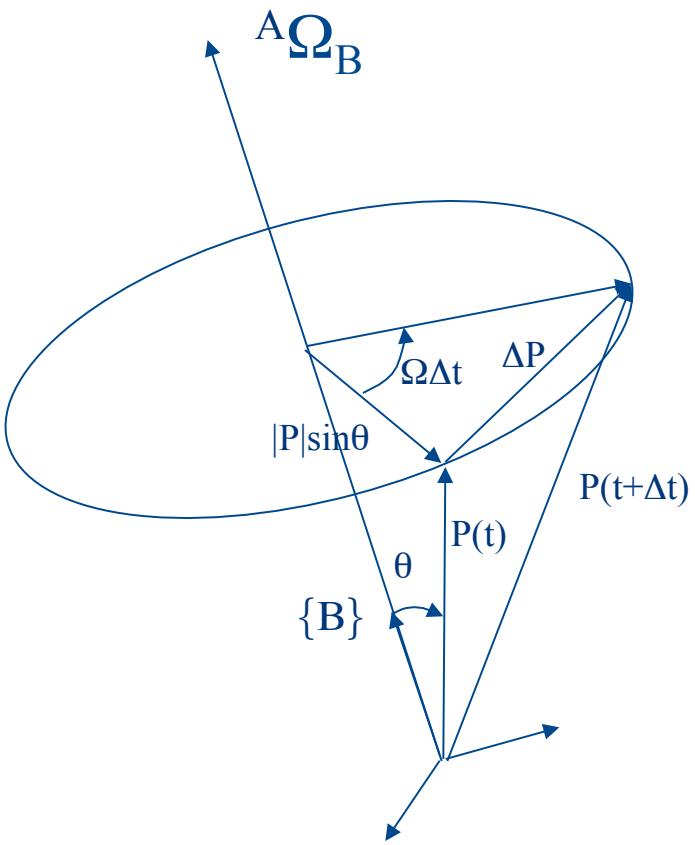
- If we look at a small timeslice as a frame rotates with a moving point, we find

$$|\Delta \mathbf{P}| = \left(\left|{}^{A}\mathbf{P}\right| \sin\theta\right)\left(\left|{}^{A}\Omega_{B}\right| \Delta t\right)$$

$$\frac{|\Delta \mathbf{P}|}{\Delta t} = \left(\left|{}^{A}\mathbf{P}\right| \sin\theta\right)\left(\left|{}^{A}\Omega_{B}\right|\right)$$

$$= {}^{A}\Omega_{B} \times {}^{A}\mathbf{P}$$

$${}^{A}\mathbf{V}_{P} = {}^{A}\Omega_{B} \times {}^{A}R_{B}\,{}^{B}\mathbf{P}$$



Kinematics: Manipulator Velocities

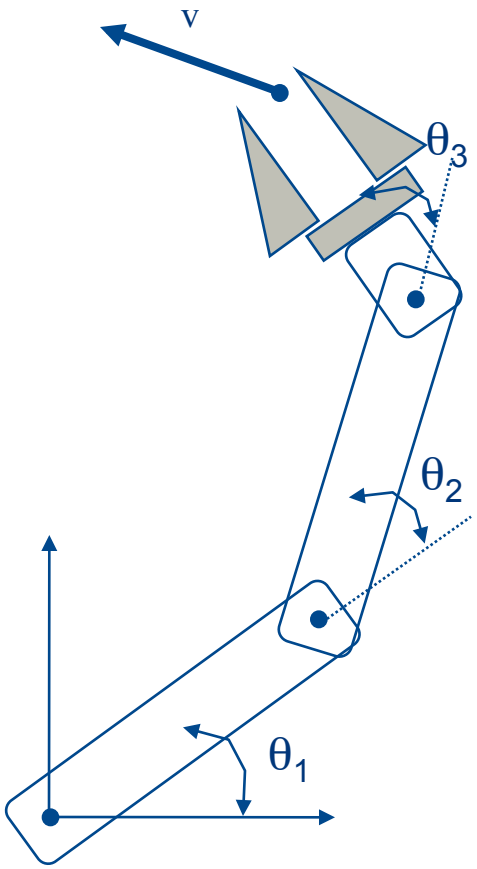
- Consider again the schematic of the planar manipulator shown. We found that the end effector position is given by

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

- Differentiating wrt to t

$$\dot{x} = -L_1 s_1 \dot{\theta}_1 - L_2 s_{12}(\dot{\theta}_1 + \dot{\theta}_2) - L_3 s_{123}(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)$$
$$\dot{y} = L_1 c_1 \dot{\theta}_1 + L_2 c_{12}(\dot{\theta}_1 + \dot{\theta}_2) + L_3 c_{123}(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)$$

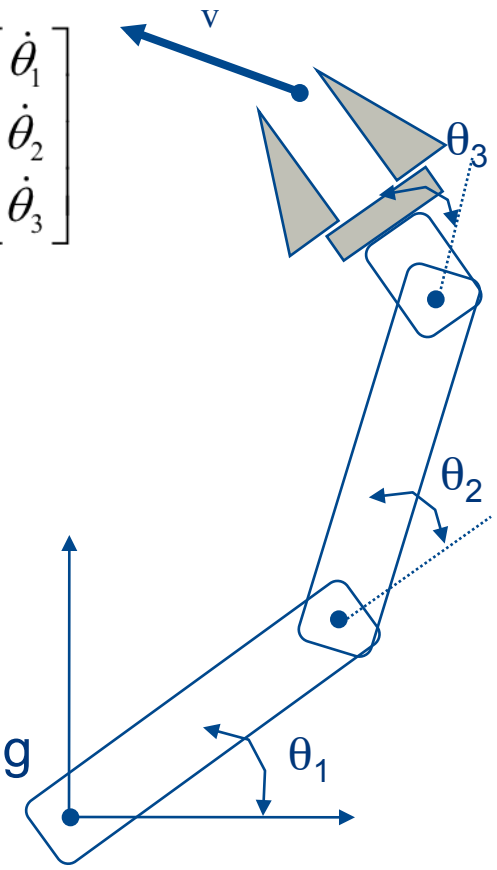
- This gives us the velocity of the end effector as a function of pose and joint velocities



Kinematics: Manipulator Velocities

- Rearranging, we can recast this relation in matrix form

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -L_1 s_1 - L_2 s_{12} - L_3 s_{123} & -L_2 s_{12} - L_3 s_{123} & -L_3 s_{123} \\ L_1 c_1 + L_2 c_{12} + L_3 c_{123} & L_2 c_{12} + L_3 c_{123} & L_3 c_{123} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}$$



- Or

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} & \frac{\partial x}{\partial \theta_3} \\ \frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} & \frac{\partial y}{\partial \theta_3} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}$$

- The resulting matrix is called the **Jacobian** and provides us with a mapping from Joint Space to Cartesian Space.

Kinematics: The Jacobian

- In general, the Jacobian takes the form

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_i \end{bmatrix} = \begin{bmatrix} \frac{\partial x_1}{\partial \theta_1} & \frac{\partial x_1}{\partial \theta_2} & \cdots & \frac{\partial x_1}{\partial \theta_j} \\ \frac{\partial x_2}{\partial \theta_1} & \frac{\partial x_2}{\partial \theta_2} & \cdots & \frac{\partial x_2}{\partial \theta_j} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial x_i}{\partial \theta_1} & \frac{\partial x_i}{\partial \theta_2} & \cdots & \frac{\partial x_i}{\partial \theta_j} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \vdots \\ \dot{\theta}_j \end{bmatrix}$$

- Or more succinctly

$$\dot{\mathbf{X}} = \mathbf{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}$$

Inverse Kinematics: Inverse Jacobian

- In many instances, we are also interested in computing the set of joint velocities that will yield a particular velocity at the end effector:

$$\dot{\boldsymbol{\theta}} = \mathbf{J}(\boldsymbol{\theta})^{-1}\dot{\mathbf{X}}$$

- But, the inverse of the Jacobian may be undefined at some points in the workspace.
- The points where the Jacobian is noninvertible ("singular") are the *singularities* of the mechanism.
- This is the same concept as we explored earlier, but now we have a mathematical definition.
- A good test for singularity is the determinant: $\det(\mathbf{J})=0$ for singular $\mathbf{J}$

Inverse Jacobian Example

- For a simple two link manipulator, we have

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

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

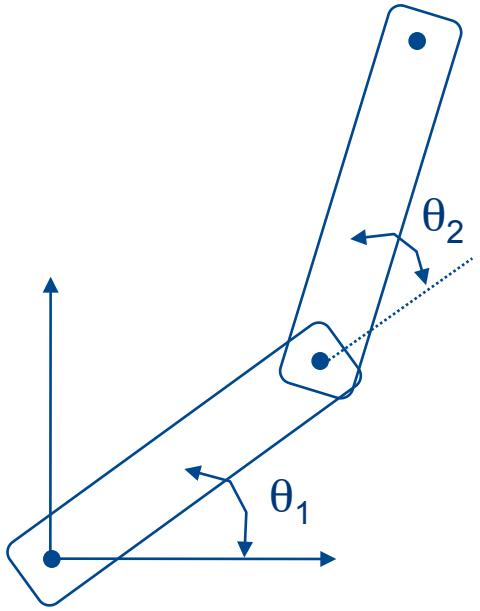
- The Jacobian for this is

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -L_1 s_1 - L_2 s_{12} & -L_2 s_{12} \\ L_1 c_1 + L_2 c_{12} & L_2 c_{12} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

- Taking the inverse of the Jacobian:

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = \frac{1}{L_1 L_2 s_2} \begin{bmatrix} L_2 c_{12} & L_2 s_{12} \\ -L_1 c_1 - L_2 c_{12} & -L_1 s_1 - L_2 s_{12} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}$$

- Clearly, as θ_2 approaches 0 or π this manipulator becomes singular



Static Forces

- We can also use the Jacobian to compute the joint torques required to maintain a particular force at the end effector
- Consider the concept of virtual work

$$F \cdot \delta \mathbf{X} = \tau \cdot \delta \boldsymbol{\theta}$$

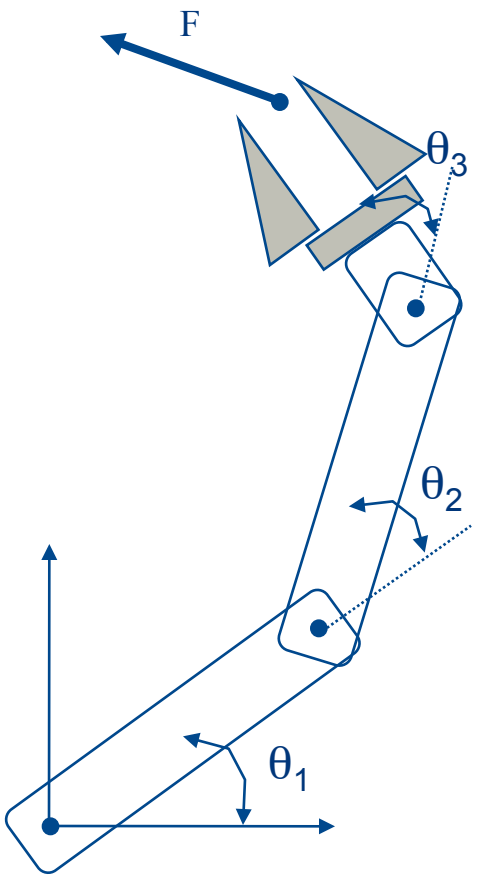
- Or

$$F^T \delta \mathbf{X} = \tau^T \delta \boldsymbol{\theta}$$

- Earlier we saw that
- So that $\delta \mathbf{X} = \mathbf{J} \delta \boldsymbol{\theta}$
- Or

$$F^T \mathbf{J} = \tau^T$$

$$\tau = \mathbf{J}^T F$$



Dynamics

- We can also consider the forces that are required to achieve a particular motion of a manipulator or other body
- Understanding the way in which motion arises from torques applied by the actuators or from external forces allows us to control these motions
- There are a number of methods for formulating these equations, including
  - Newton-Euler Dynamics
  - Langrangian Mechanics

Dynamics

- For Manipulators, the general form is

$$\tau = M(\Theta)\ddot{\Theta} + V(\Theta, \dot{\Theta}) + G(\Theta)$$

where

  - τ is a vector of joint torques
  - Θ is the nx1 vector of joint angles
  - $M(\Theta)$ is the nxn mass matrix
  - $V(\Theta,\dot{\Theta})$ is the nx1 vector of centrifugal and Coriolis terms
  - $G(\Theta)$ is an nx1 vector of gravity terms
- Notice that all of these terms depend on Θ so the dynamics varies as the manipulator moves

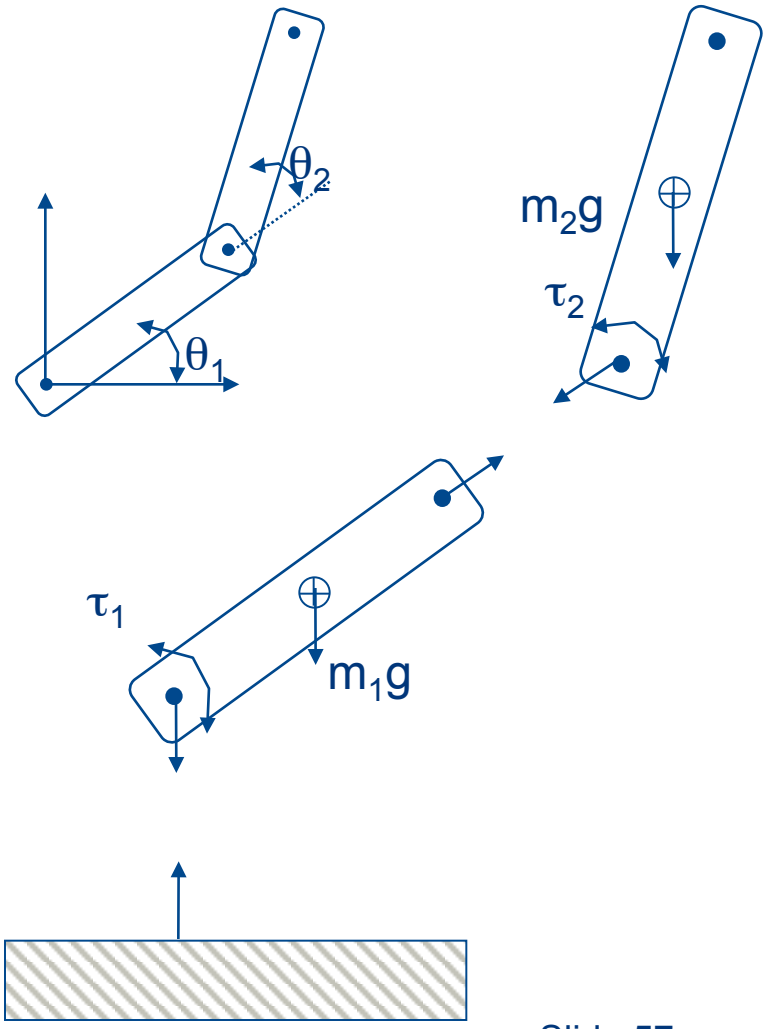
Dynamics – Newton-Euler

- In general, we could analyse the dynamics of robotic systems using classical Newtonian mechanics

$$\sum F = m\ddot{x}$$

$$\sum T = J\ddot{\theta}$$

- This can entail iteratively calculating velocities and accelerations for each link and then computing force and moment balances in the system
- Alternatively, closed form solutions may exist for simple configurations



Dynamics – Newton-Euler

The iterative Newton-Euler dynamics algorithm

The complete algorithm for computing joint torques from the motion of the joints is composed of two parts. First, link velocities and accelerations are iteratively computed from link 1 out to link n and the Newton-Euler equations are applied to each link. Second, forces and torques of interaction and joint actuator torques are computed recursively from link n back to link 1. The equations are summarized below for the case of all joints rotational.

Outward iterations: $i : 0 \rightarrow 5$

$$ {}^{i+1}\omega_{i+1} = {}^{i+1}_{i}R\,{}^{i}\omega_i + \dot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1}, \tag{6.45} $$

$$ {}^{i+1}\dot{\omega}_{i+1} = {}^{i+1}_{i}R\,{}^{i}\dot{\omega}_i + {}^{i+1}_{i}R\,{}^{i}\omega_i \times \dot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1} + \ddot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1}, \tag{6.46} $$

$$ {}^{i+1}\dot{v}_{i+1} = {}^{i+1}_{i}R\left({}^{i}\dot{\omega}_i \times {}^{i}P_{i+1} + {}^{i}\omega_i \times \left({}^{i}\omega_i \times {}^{i}P_{i+1}\right) + {}^{i}\dot{v}_i\right), \tag{6.47} $$

$$ {}^{i+1}\dot{v}_{C_{i+1}} = {}^{i+1}\dot{\omega}_{i+1} \times {}^{i+1}P_{C_{i+1}} + {}^{i+1}\omega_{i+1} \times \left({}^{i+1}\omega_{i+1} \times {}^{i+1}P_{C_{i+1}}\right) + {}^{i+1}\dot{v}_{i+1}, \tag{6.48} $$

$$ {}^{i+1}F_{i+1} = m_{i+1}\,{}^{i+1}\dot{v}_{C_{i+1}}, \tag{6.49} $$

$$ {}^{i+1}N_{i+1} = {}^{C_{i+1}}I_{i+1}\,{}^{i+1}\dot{\omega}_{i+1} + {}^{i+1}\omega_{i+1} \times {}^{C_{i+1}}I_{i+1}\,{}^{i+1}\omega_{i+1}. \tag{6.50} $$

Inward iterations: $i : 6 \rightarrow 1$

$$ {}^{i}f_i = {}^{i}_{i+1}R\,{}^{i+1}f_{i+1} + {}^{i}F_i, \tag{6.51} $$

$$ {}^{i}n_i = {}^{i}N_i + {}^{i}_{i+1}R\,{}^{i+1}n_{i+1} + {}^{i}P_{C_i} \times {}^{i}F_i + {}^{i}P_{i+1} \times {}^{i}_{i+1}R\,{}^{i+1}f_{i+1}, \tag{6.52} $$

$$ \tau_i = {}^{i}n_i^T\,{}^{i}\hat{Z}_i. \tag{6.53} $$

From: J.J. Craig
"Introduction to Robotics: Mechanics and Control", 2nd Edition, Addison-Wesley ISBN 0-201-09528-9

Dynamics – Langrangian Mechanics

- Alternatively, we can use Langrangian Mechanics to compute the dynamics of a manipulator (or other robotic system)
- The Langrangian is defined as the difference between the Kinetic and Potential energy in the system
- Using this formulation and the concept of virtual work we can find the forces and torques acting on the system
- This may seem more involved but is often easier to formulate for complex systems

$$L = K - P$$

$$F_i = \frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{x}_i}\right) - \frac{\partial L}{\partial x}$$

$$T_i = \frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{\theta}_i}\right) - \frac{\partial L}{\partial \theta}$$

Conclusions

- Kinematics is the study of motion without regard to the forces that create it
- Kinematics is important in many instances in robotics
- The study of dynamics allows us to understand the forces and torques which act on a system and result in motion
- Understanding these motions, and the required forces, is essential for designing these systems

Further Reading

- Niku
  - Chapter 2-4
- Craig
  - Chapters 3-6
- Spong & Vidayasagar
  - Chapters 3-6