Chapter 2
Math Fundamentals

Part 2

2.4 Kinematics of Mechanisms

2.5 Orientation and Angular Velocity

Outline

- 2.4 Kinematics of Mechanisms
- 2.5 Orientation and Angular Velocity

Outline

- 2.4 Kinematics of Mechanisms
  - 2.4.1 Forward Kinematics
  - 2.4.2 Inverse Kinematics
  - 2.4.3 Differential Kinematics
  - Summary
- 2.5 Orientation and Angular Velocity

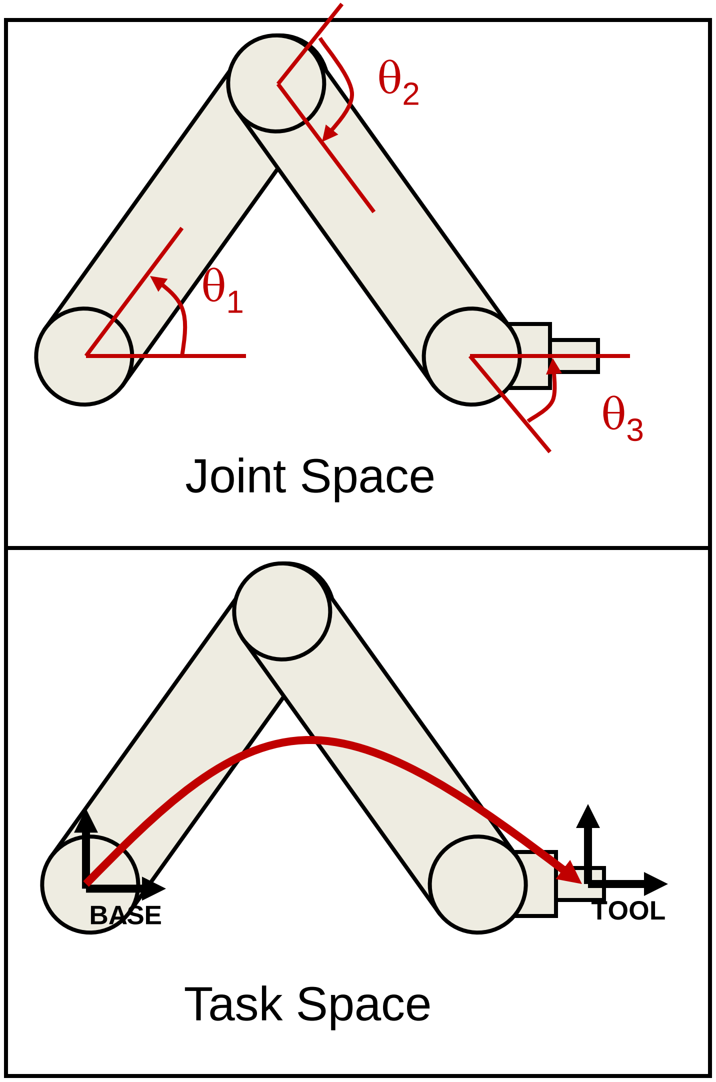
Definitions

- Motion = movement of the whole body thru space.
- Articulation = reconfigures mass without substantial motion
- Attitude = pitch and roll.
- Orientation = attitude & (heading or yaw).
- Pose = position & orientation

$$\begin{bmatrix} x & y & \psi \end{bmatrix}^T \qquad \begin{bmatrix} x & y & z & \theta & \phi & \psi \end{bmatrix}^T$$

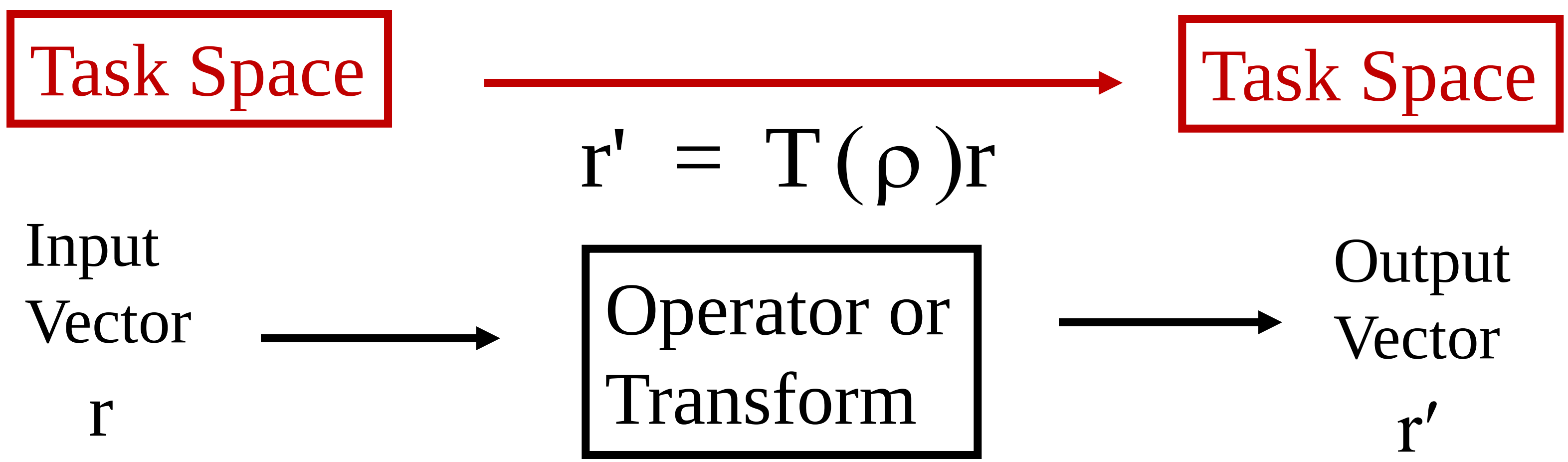
Task and Joint Space

- For manipulators, joint space is also C Space.
- Manipulators are difficult for operators to control in joint space.



Linear Mapping

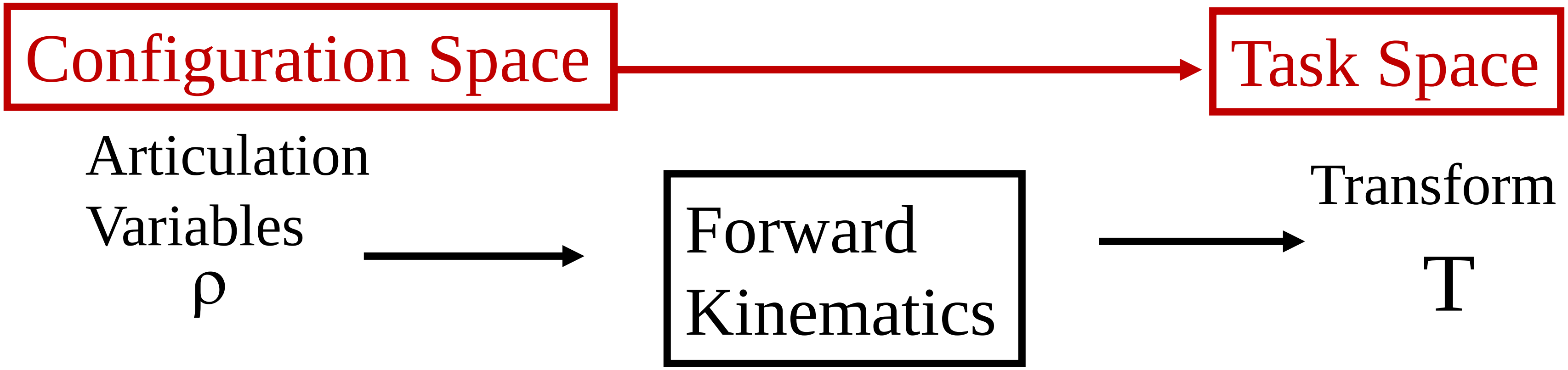
- So far, we have used:



- We have considered this to be a <u>linear mapping</u> in r
  - In part, because the matrix is considered a constant.

2.4.1.1 Nonlinear Mapping

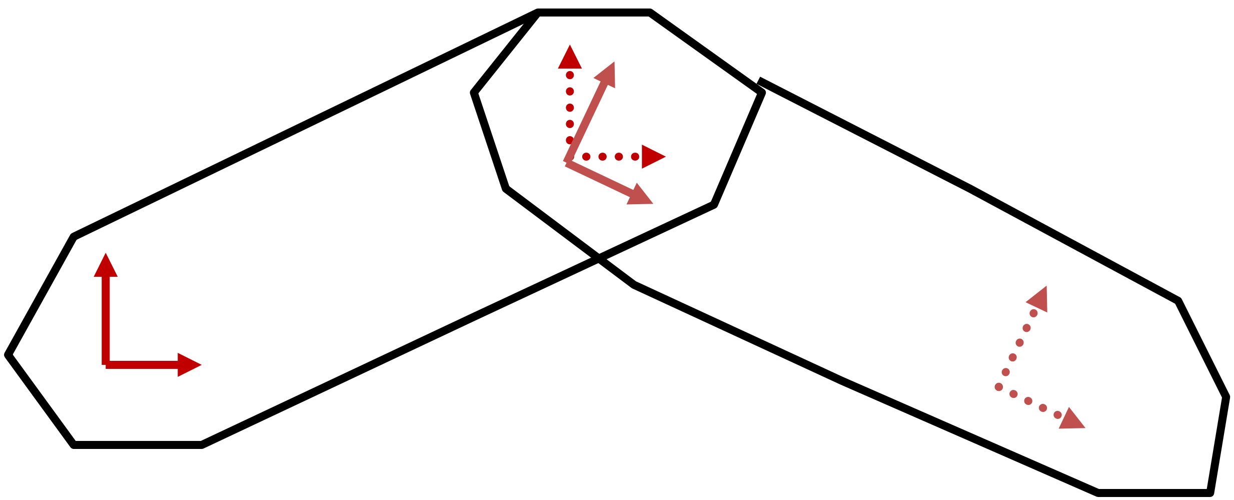
- There is another view of <u>just the T(ρ) part</u> when the "aligning operations" correspond to real joints:



- This is a <u>nonlinear mapping</u> in ρ.
- T is a matrix-valued function of the configuration vector ρ.
- Recall that T can represent the pose of a rigid body (we saw this in 2D with HTs).

2.4.1.2 Mechanism Models

- Let ρ represent the articulations of a mechanism.
- It is convenient to think about the moving axis operations which align a sequence of frames with each other.



Conventional Rules of Forward Kinematics

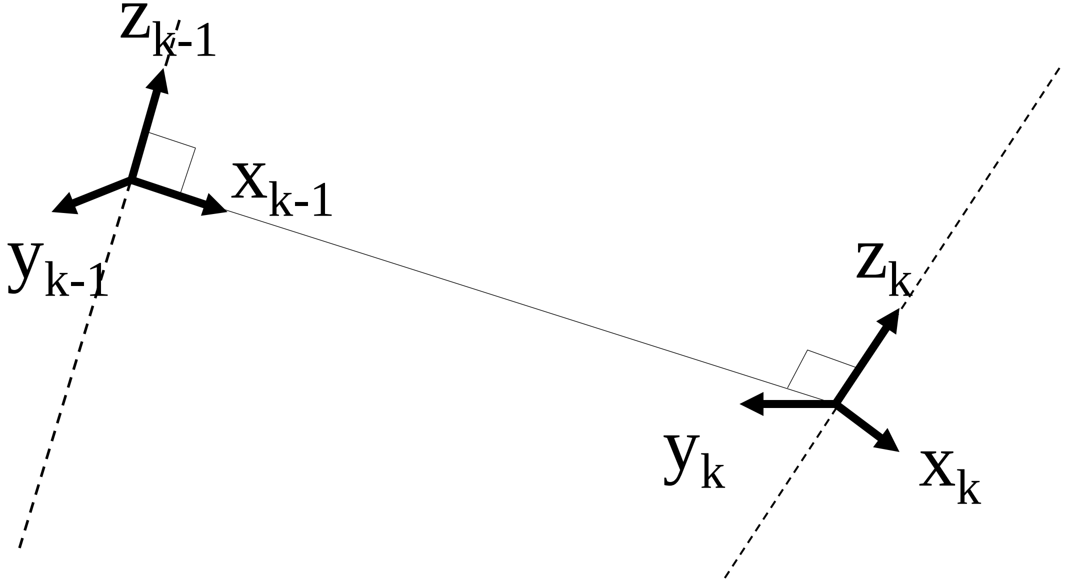
- 1: Assign embedded frames to the links in sequence such that the operations which move each frame into coincidence with the next are a function of the current joint variable.
- 2: Write the *orthogonal operator matrices* which correspond to these operations in *left to right order*.
- This process will generate the matrix that:
  - A: represents the position and orientation of the last embedded frame with respect to the first, or equivalently,
  - B: which converts the coordinates of a point from the last to the first.

2.4.1.3 Denavit-Hartenberg Convention

- A special product of 4 fundamental operators is used as a basic conceptual unit.
- Its still an orthogonal transform so it has the properties of the component transforms, namely:
  - Operates on points
  - Converts coordinates
  - Represents axes of one system wrt another

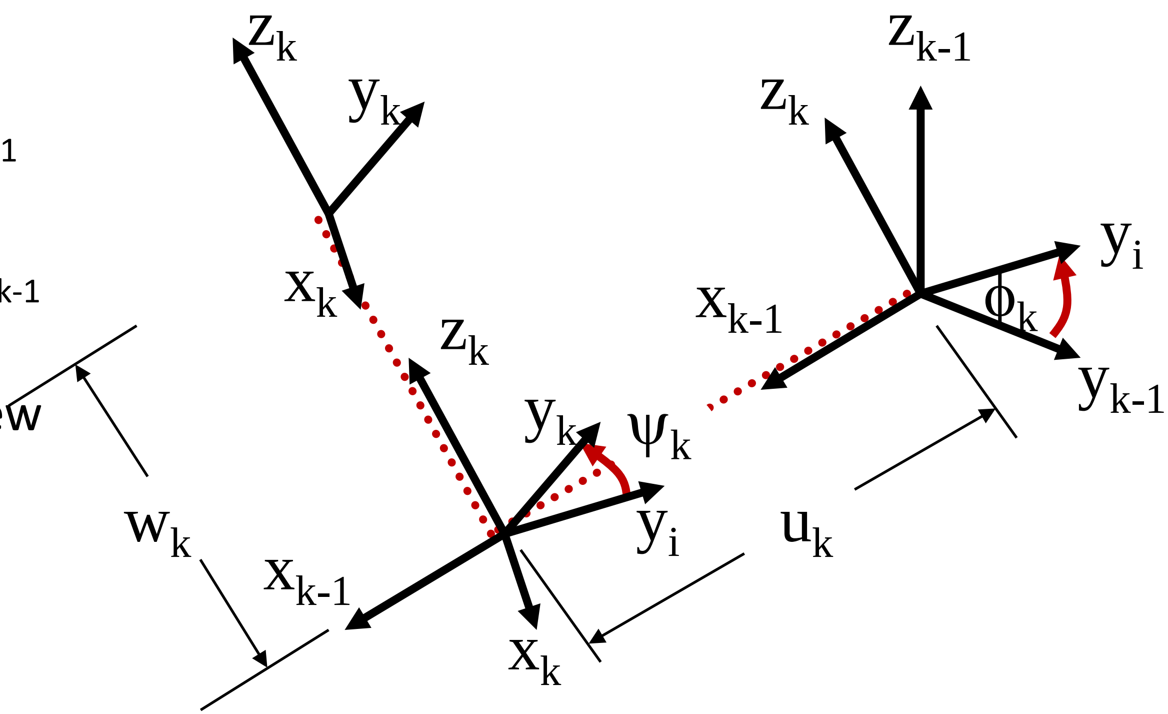
Denavit- Hartenberg Convention

- Rules:
  - Assign frames to links. Consider in order from base to end.
  - Place z axis of each frame on joint linear or rotary axis.
  - Point x axis along mutual perpendicular



Denavit- Hartenberg Convention

- Move first frame into coincidence with second thus:
  - rotate around the x_{k-1} axis by an angle ϕ_k
  - translate along the x_{k-1} axis by a distance u_k
  - rotate around the new z axis by an angle ψ_k
  - translate along the new z axis by a distance w_k



How can we get away with just four dof?

Denavit- Hartenberg Convention

- Moving axis <u>operations</u> in <u>left to right</u> order:

$$T_k^{k-1} = Rotx(\phi_k) Trans(u_k, 0, 0) Rotz(\psi_k) Trans(0, 0, w_k)$$

$$T_k^{k-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\phi_k & -s\phi_k & 0 \\ 0 & s\phi_k & c\phi_k & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & u_k \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\psi_k & -s\psi_k & 0 & 0 \\ s\psi_k & c\psi_k & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & w_k \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_k = T_k^{k-1} = \begin{bmatrix} c\psi_k & -s\psi_k & 0 & u_k \\ c\phi_k s\psi_k & c\phi_k c\psi_k & -s\phi_k & -s\phi_k w_k \\ s\phi_k s\psi_k & s\phi_k c\psi_k & c\phi_k & c\phi_k w_k \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

4 “parameters”

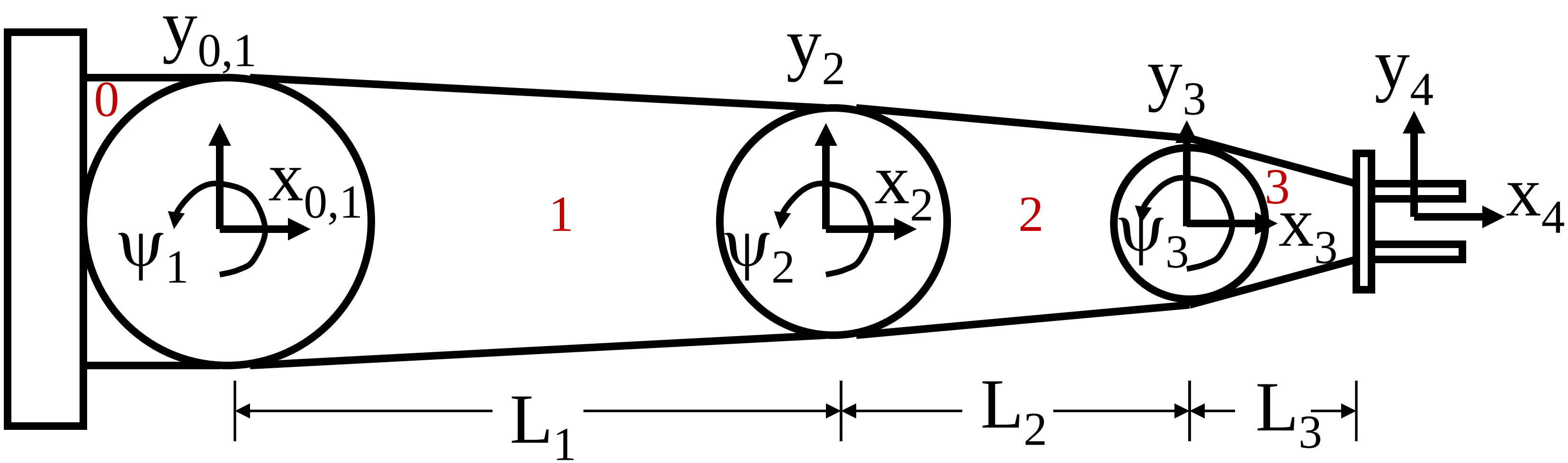
Denavit- Hartenberg Convention

$$A_k = T_k^{k-1} = \begin{bmatrix} c\psi_k & -s\psi_k & 0 & u_k \\ c\phi_k s\psi_k & c\phi_k c\psi_k & -s\phi_k & -s\phi_k w_k \\ s\phi_k s\psi_k & s\phi_k c\psi_k & c\phi_k & c\phi_k w_k \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Special Notation

- This matrix has the following interpretations:
  - It will move a point or rotate a direction by the operator which describes how frame k is related to frame k-1.
  - Its columns represent the axes and origin of frame k expressed in frame k-1 coordinates.
  - It converts coordinates from frame k to frame k-1.

2.4.1.4 Example: 3 Link Planar Manipulator



Link	ϕ	u	ψ	w
0	0	0	ψ_1	0
1	0	L_1	ψ_2	0
2	0	L_2	ψ_3	0
3	0	L_3	0	0

Example: 3 Link Planar Manipulator

$$A_k = T_k^{k-1} = \begin{bmatrix} c\psi_k & -s\psi_k & 0 & u_k \\ c\phi_k s\psi_k & c\phi_k c\psi_k & -s\phi_k & -s\phi_k w_k \\ s\phi_k s\psi_k & s\phi_k c\psi_k & c\phi_k & c\phi_k w_k \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

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

$$A_2 = \begin{bmatrix} c_2 & -s_2 & 0 & L_1 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_2^{-1} = \begin{bmatrix} c_2 & s_2 & 0 & -c_2 L_1 \\ -s_2 & c_2 & 0 & s_2 L_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} c_3 & -s_3 & 0 & L_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_3^{-1} = \begin{bmatrix} c_3 & s_3 & 0 & -c_3 L_2 \\ -s_3 & c_3 & 0 & s_3 L_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_4 = \begin{bmatrix} 1 & 0 & 0 & L_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_4^{-1} = \begin{bmatrix} 1 & 0 & 0 & -L_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Example: 3 Link Planar Manipulator

$$T_4^0 = T_1^0 T_2^1 T_3^2 T_4^3 = A_1 A_2 A_3 A_4$$

$$T_4^0 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_2 & -s_2 & 0 & L_1 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_3 & -s_3 & 0 & L_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & L_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_4^0 = \left[\begin{array}{cc|c|c} c_{123} & -s_{123} & 0 & (c_{123}L_3 + c_{12}L_2 + c_1L_1) \\ \hline s_{123} & c_{123} & 0 & (s_{123}L_3 + s_{12}L_2 + s_1L_1) \\ \hline 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

orientation position

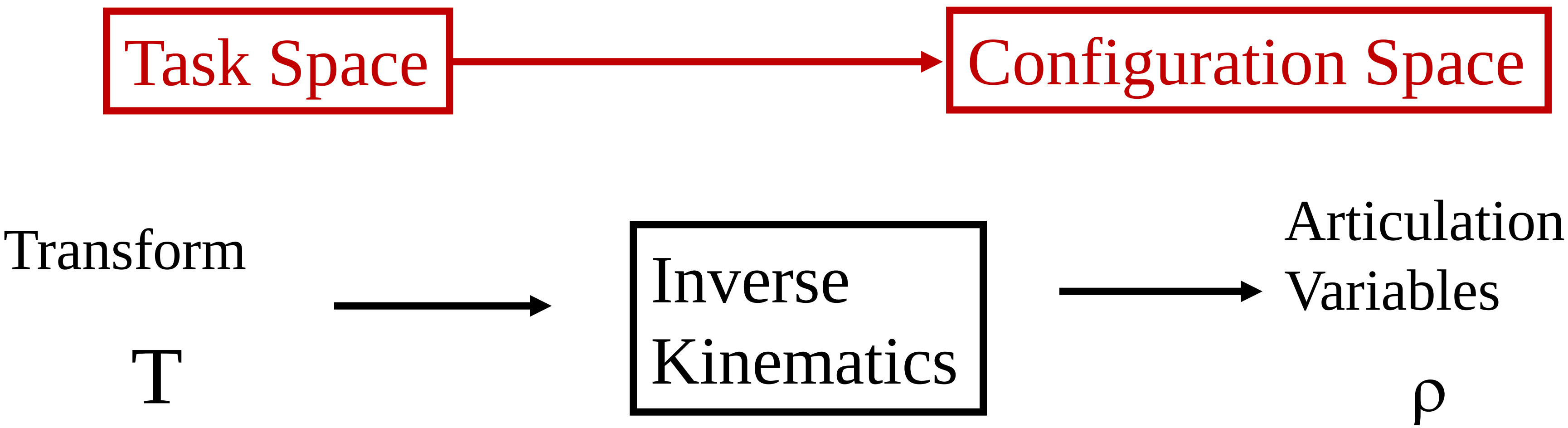
A Completely General Forward Kinematics Solution!

Outline

- 2.4 Kinematics of Mechanisms
  - 2.4.1 Forward Kinematics
  - <u>2.4.2 Inverse Kinematics</u>
  - 2.4.3 Differential Kinematics
  - Summary
- 2.5 Orientation and Angular Velocity

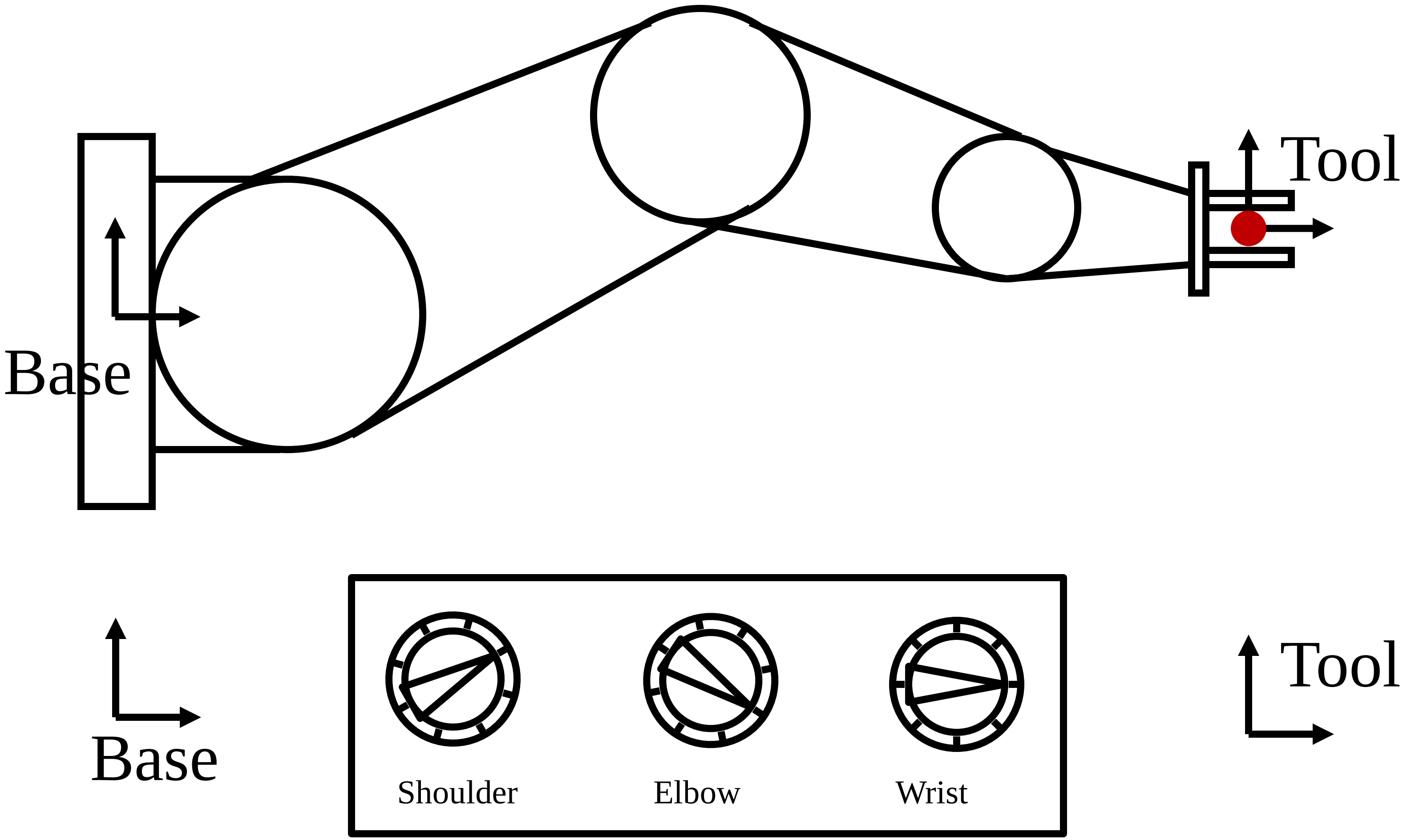
Nonlinear Mapping

- Recall our view of: T(ρ)



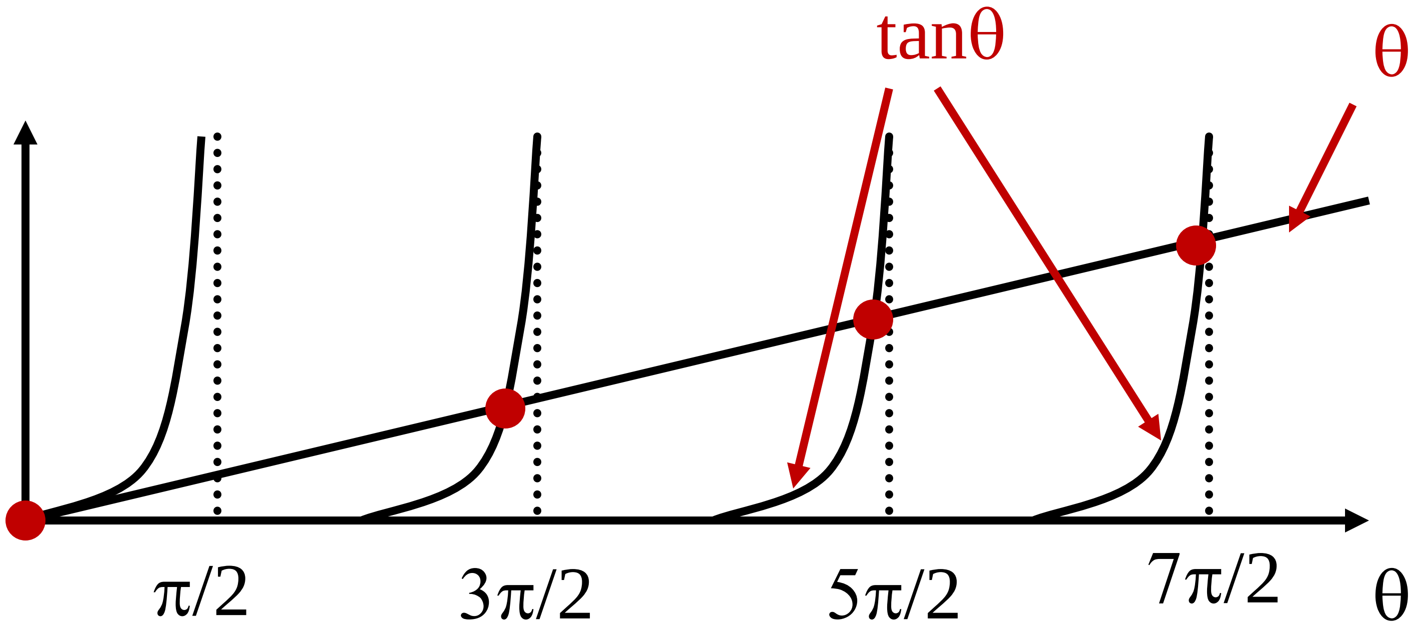
- The <u>harder</u> (by far) of the two directions.
- T is given as a block of numbers, not symbols.

Kinematics



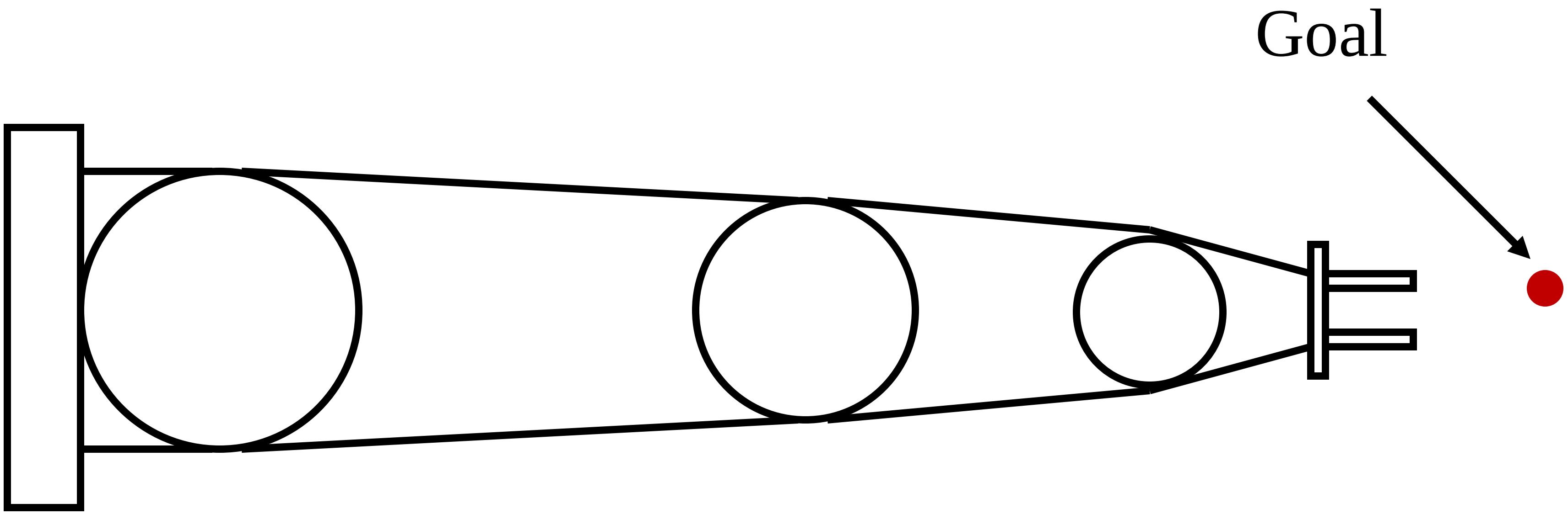
2.4.2.1 Existence and Uniqueness

- Existence
  - Nonlinear equations need not be solveable.
- Uniqueness
  - There is no rule requiring only one answer.
  - E.g. $\tan\theta = \theta$



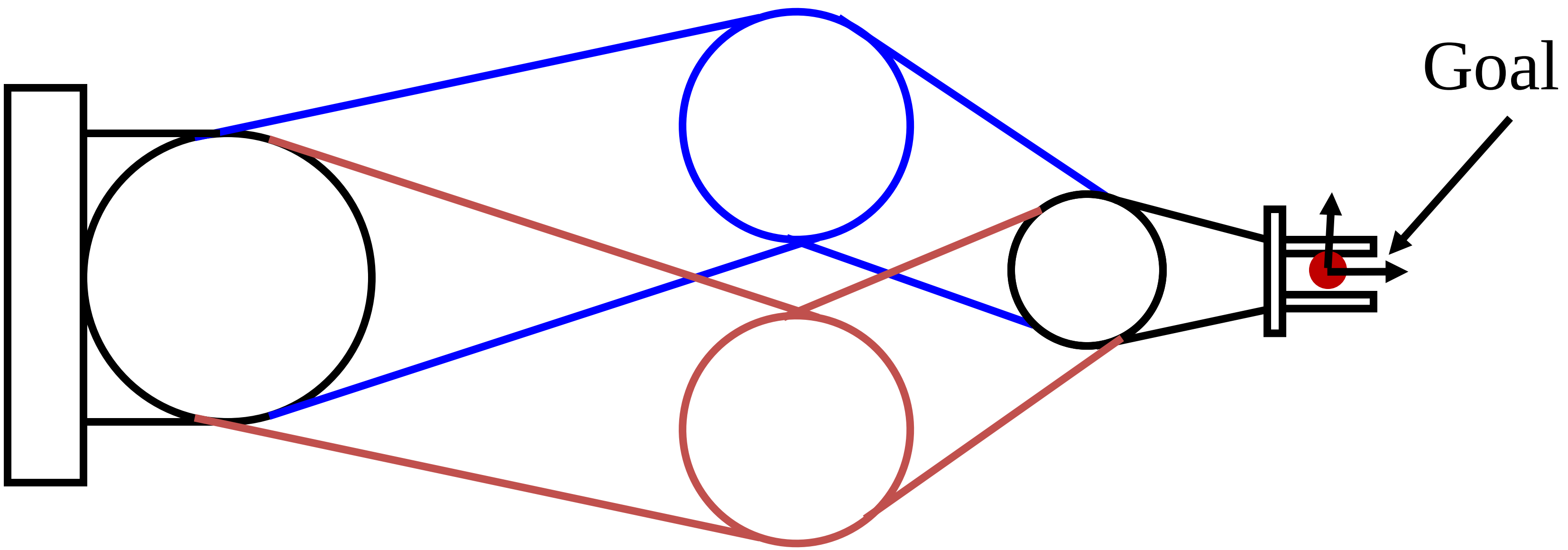
Existence

- Nonlinear equations need not be solveable.



Uniqueness

- There is no rule requiring only one answer.



2.4.2.2 Technique

- Rewrite equations in multiple ways in order to isolate unknowns.
  - No new info generated, its just algebra.

Premultiply

$$T_4^0 = A_1 A_2 A_3 A_4$$

$$A_1^{-1} T_4^0 = A_2 A_3 A_4$$

$$A_2^{-1} A_1^{-1} T_4^0 = A_3 A_4$$

$$A_3^{-1} A_2^{-1} A_1^{-1} T_4^0 = A_4$$

$$A_4^{-1} A_3^{-1} A_2^{-1} A_1^{-1} T_4^0 = I$$

Postmultiply

$$T_4^0 = A_1 A_2 A_3 A_4$$

$$T_4^0 A_4^{-1} = A_1 A_2 A_3$$

$$T_4^0 A_4^{-1} A_3^{-1} = A_1 A_2$$

$$T_4^0 A_4^{-1} A_3^{-1} A_2^{-1} = A_1$$

$$T_4^0 A_4^{-1} A_3^{-1} A_2^{-1} A_1^{-1} = I$$

2.4.2.3 Example: 3 Link Planar Manipulator

- Account for known quantities.
- This is a 2D problem, so…

$$T_4^0 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & 0 & p_x \\ r_{21} & r_{22} & 0 & p_y \\ 0 & 0 & 1 & \boxed{0} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Example: 3 Link Planar Manipulator

$$T_4^0 = A_1 A_2 A_3 A_4$$

$$\begin{bmatrix} r_{11} & r_{12} & 0 & p_x \\ r_{21} & r_{22} & 0 & p_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_{123} & -s_{123} & 0 & (c_{123}L_3 + c_{12}L_2 + c_1L_1) \\ s_{123} & c_{123} & 0 & (s_{123}L_3 + s_{12}L_2 + s_1L_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- From the (1,1) and (2,1) elements we have:

$$\psi_{123} = \text{atan2}(r_{21}, r_{11})$$

Example: 3 Link Planar Manipulator

$$T_4^0 A_4^{-1} = A_1 A_2 A_3$$

$$\begin{bmatrix} r_{11} & r_{12} & 0 & -r_{11}L_3 + p_x \\ r_{21} & r_{22} & 0 & -r_{21}L_3 + p_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_{123} & -s_{123} & 0 & c_1(c_2L_2 + L_1) - s_1(s_2L_2) \\ s_{123} & c_{123} & 0 & s_1(c_2L_2 + L_1) + c_1(s_2L_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} r_{11} & r_{12} & 0 & -r_{11}L_3 + p_x \\ r_{21} & r_{22} & 0 & -r_{21}L_3 + p_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_{123} & -s_{123} & 0 & c_{12}L_2 + c_1L_1 \\ s_{123} & c_{123} & 0 & s_{12}L_2 + s_1L_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- From the (1,4) and (2,4) elements we have:

$$k_1 = -r_{11}L_3 + p_x = c_{12}L_2 + c_1L_1$$

$$k_2 = -r_{21}L_3 + p_y = s_{12}L_2 + s_1L_1$$

Example: 3 Link Planar Manipulator

- Repeat from last slide: from the (1,4) and (2,4) elements we have:

$$k_1 = -r_{11}L_3 + p_x = c_{12}L_2 + c_1L_1$$

$$k_2 = -r_{21}L_3 + p_y = s_{12}L_2 + s_1L_1$$

- Square and add (eliminates θ_1) to yield:

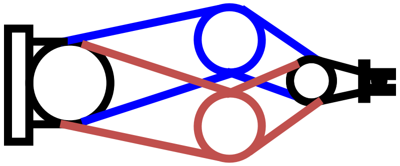
$$k_1^2 + k_2^2 = L_2^2 + L_1^2 + 2L_2L_1(c_1c_{12} + s_1s_{12})$$

- Or:

$$k_1^2 + k_2^2 = L_2^2 + L_1^2 + 2L_2L_1c_2 \quad \leftarrow \cos\psi_2$$

- Rearranging gives the answer:

$$\psi_2 = \operatorname{acos}\left[\frac{(k_1^2 + k_2^2) - (L_2^2 + L_1^2)}{2L_2L_1}\right]$$



- Each solution gives different values for the other two angles.

Example: 3 Link Planar Manipulator

- 2nd time, from the (1,4) and (2,4) elements we have:

$$k_1 = -r_{11}L_3 + p_x = c_{12}L_2 + c_1L_1$$

$$k_2 = -r_{21}L_3 + p_y = s_{12}L_2 + s_1L_1$$

- With ψ2 known, these are:

$$c_1k_3 - s_1k_4 = k_1$$

$$s_1k_3 + c_1k_4 = k_2$$

- That's a standard form. The solution is:

$$\psi_1 = \operatorname{atan2}[(k_2k_3 - k_1k_4), (k_1k_3 + k_2k_4)]$$

- The last angle is then:

$$\psi_3 = \psi_{123} - \psi_2 - \psi_1$$

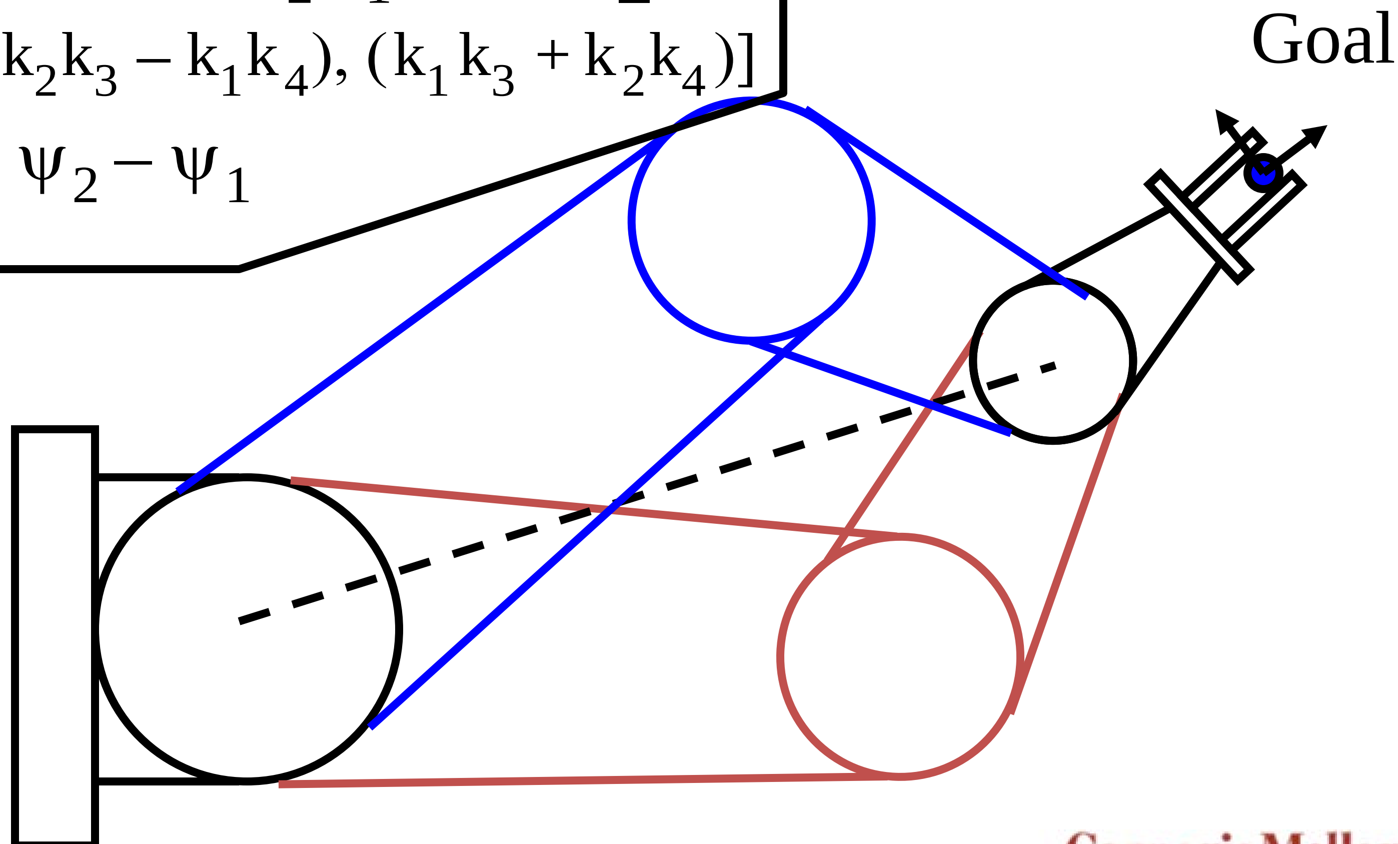
That's PRETTY AWESOME!

$$\psi_{123} = \operatorname{atan2}(r_{21}, r_{11})$$

$$\psi_2 = \operatorname{acos}\left[\frac{(k_1^2 + k_2^2) - (L_2^2 + L_1^2)}{2L_2L_1}\right]$$

$$\psi_1 = \operatorname{atan2}[(k_2k_3 - k_1k_4), (k_1k_3 + k_2k_4)]$$

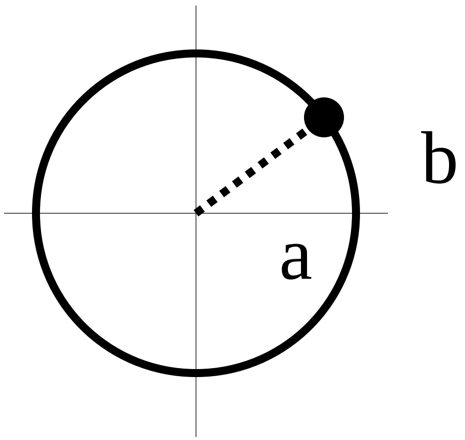
$$\psi_3 = \psi_{123} - \psi_2 - \psi_1$$



2.4.2.4 Standard Forms: Explicit Tangent

- Occurs very frequently:

$$a = c_n$$
$$b = s_n$$



- Isolate tangent:

- Clearly:

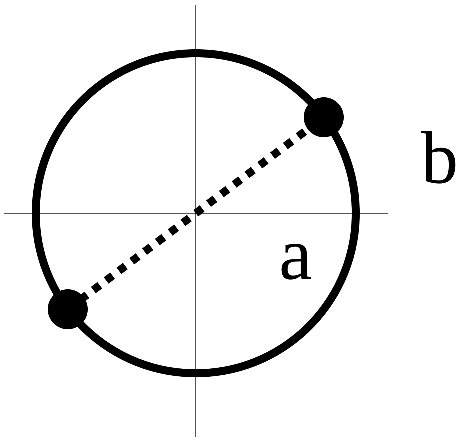
$$\psi_n = \text{atan2}(b, a)$$

Why do we need two equations for one unknown?

2.4.2.4 Standard Forms: Point Symmetric

- Occurs as:

$$s_n a - c_n b = 0$$



- Isolate tangent:

$$\frac{s_n}{c_n} = \frac{b}{a}$$

- Clearly:

$$\psi_n = \operatorname{atan2}(b, a)$$

$$\psi_n = \operatorname{atan2}(-b, -a)$$

Why two solutions?

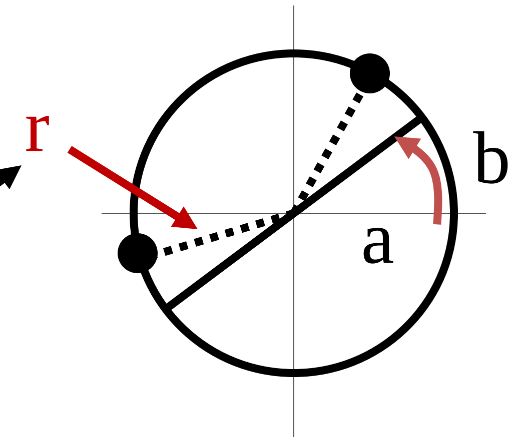
2.4.2.4 Standard Forms: Line Symmetric

- Occurs as:

$$s_n a - c_n b = c$$

- Trig substitution:

$$a = r\cos(\theta)$$
$$b = r\sin(\theta)$$



- Leads to:

$$s(\theta - \psi_n) = c / r$$

- So: $$c(\theta - \psi_n) = \pm \mathrm{sqrt}(1 - (c/r)^2)$$

- Solution

$$\psi_n = \mathrm{atan2}(b, a) - \mathrm{atan2}[c, \pm \mathrm{sqrt}(r^2 - c^2)]$$

Inverse Kinematics of a DH HT?

Transform T → Inverse Kinematics → Articulation Variables ρ

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\psi_i & -s\psi_i & 0 & u_i \\ c\phi_i s\psi_i & c\phi_i c\psi_i & -s\phi_i & -s\phi_i w_i \\ s\phi_i s\psi_i & s\phi_i c\psi_i & c\phi_i & c\phi_i w_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Inverse Kinematics of a DH HT?

$$T_b^a = Rotx(\phi_i)Trans(u_i, 0,0)Rotz(\psi_i)Trans(w_i, 0,0)$$

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\psi_i & -s\psi_i & 0 & u_i \\ c\phi_i s\psi_i & c\phi_i c\psi_i & -s\phi_i & -s\phi_i w_i \\ s\phi_i s\psi_i & s\phi_i c\psi_i & c\phi_i & c\phi_i w_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Translation part is easy:

$$u_i = p_x$$

$$w_i = \sqrt{p_y^2 + p_z^2}$$

Inverse Kinematics of a DH HT?

$$T_b^a = \mathrm{Rotx}(\phi_i)\mathrm{Trans}(u_i, 0,0)\mathrm{Rotz}(\psi_i)\mathrm{Trans}(w_i, 0,0)$$

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\psi_i & -s\psi_i & 0 & u_i \\ c\phi_i s\psi_i & c\phi_i c\psi_i & -s\phi_i & -s\phi_i w_i \\ s\phi_i s\psi_i & s\phi_i c\psi_i & c\phi_i & c\phi_i w_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Rotation not much harder:

$$\psi_i = atan2(-r_{12}, r_{11})$$
$$\phi_i = atan2(-r_{23}, r_{33})$$

38

Outline

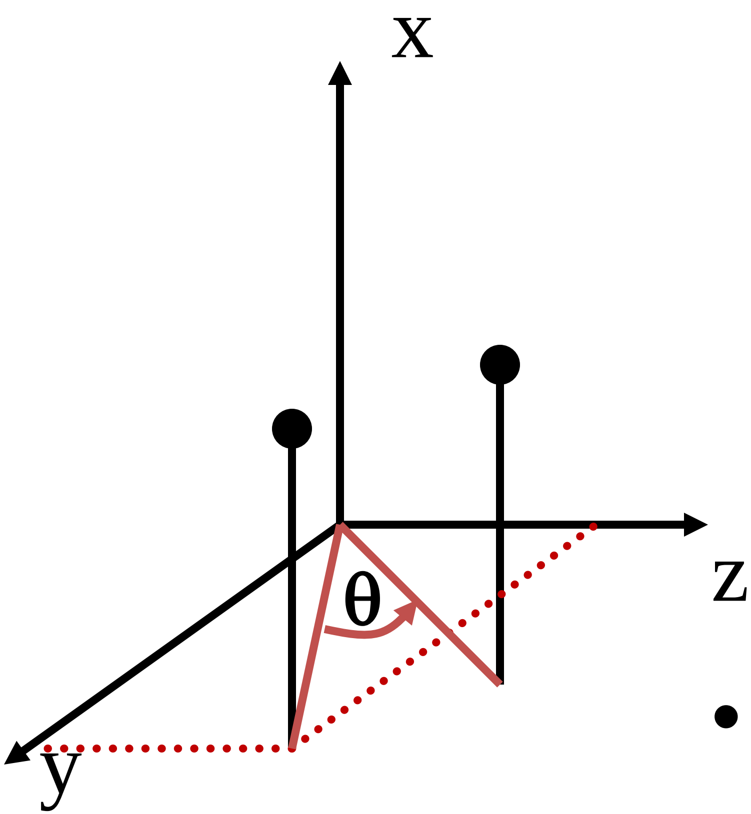
- 2.4 Kinematics of Mechanisms
  - 2.4.1 Forward Kinematics
  - 2.4.2 Inverse Kinematics
  - 2.4.3 Differential Kinematics
  - Summary
- 2.5 Orientation and Angular Velocity

Differential Kinematics

- Studies the derivatives (first order behavior) of kinematic models.
- Use Jacobians (multidimensional derivatives) to do this.
- Use them for:
  - Resolved rate control
  - Sensitivity analysis
  - Uncertainty propagation
  - Static force transformation

Derivatives of Fundamental Operators

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



$$\frac{\partial}{\partial \phi}\mathrm{Rotx}(\phi) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -s\phi & -c\phi & 0 \\ 0 & c\phi & -s\phi & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

- Note capital R in Rotx().

Derivatives of Fundamental Operators

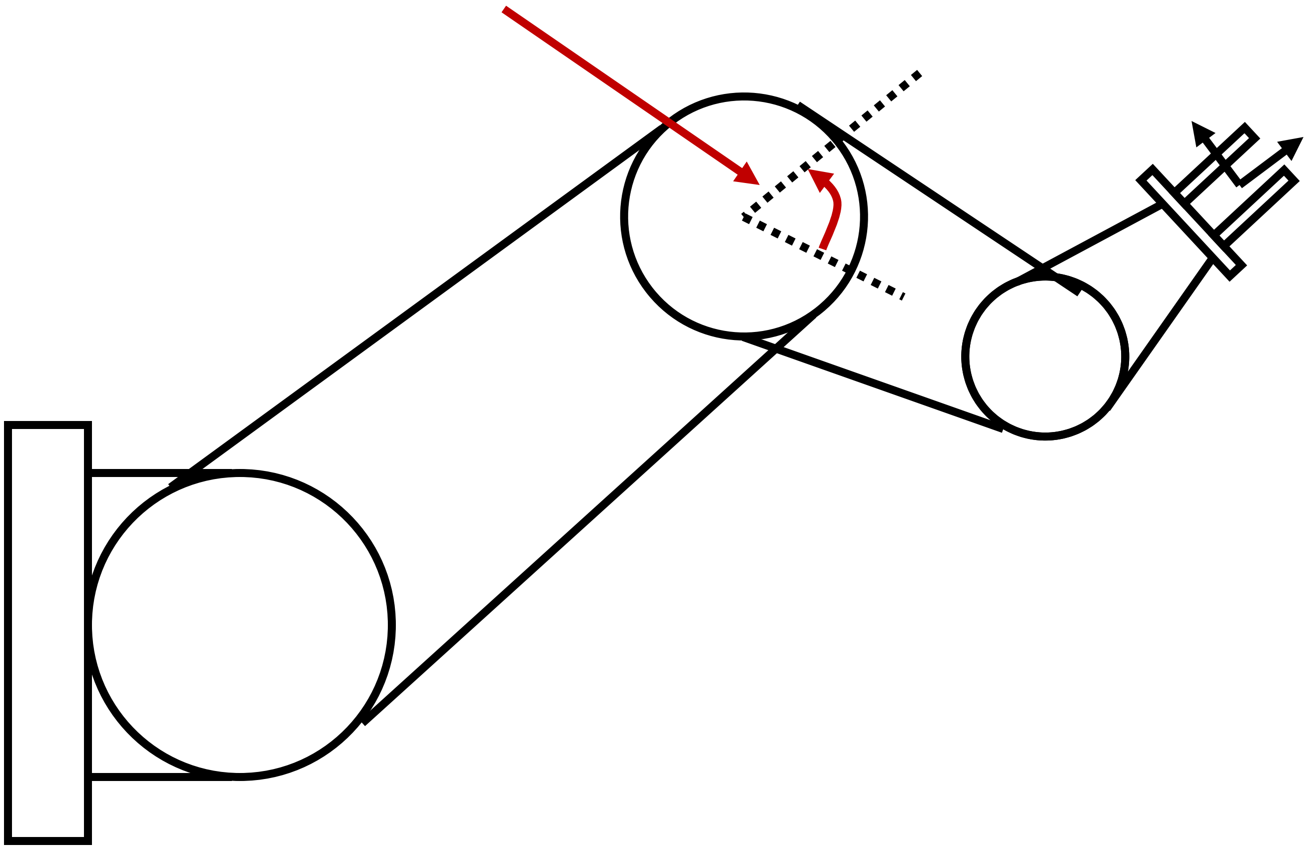
$$\frac{\partial}{\partial u}\mathrm{Trans}(u, v, w) = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad \frac{\partial}{\partial \phi}\mathrm{Rotx}(\phi) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -s\phi & -c\phi & 0 \\ 0 & c\phi & -s\phi & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\frac{\partial}{\partial v}\mathrm{Trans}(u, v, w) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad \frac{\partial}{\partial \theta}\mathrm{Roty}(\theta) = \begin{bmatrix} -s\theta & 0 & c\theta & 0 \\ 0 & 0 & 0 & 0 \\ -c\theta & 0 & -s\theta & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\frac{\partial}{\partial w}\mathrm{Trans}(u, v, w) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad \frac{\partial}{\partial \psi}\mathrm{Rotz}(\psi) = \begin{bmatrix} -s\psi & -c\psi & 0 & 0 \\ c\psi & -s\psi & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

2.4.3.2 Mechanism Jacobian

- How much does end effector move if I tweak this angle 1 degree?



2.4.3.2 Mechanism Jacobian

- Recall a Jacobian....
- If:

$$\underline{x} = \underline{F}(\underline{q})$$

x and q can be anything

- Then:

$$J = \frac{\partial \underline{x}}{\partial \underline{q}} = \frac{\partial}{\partial \underline{q}}(\underline{F}(\underline{q})) = \left[\frac{\partial x_i}{\partial q_j}\right] = \begin{bmatrix} \frac{\partial x_1}{\partial q_1} & \dots & \frac{\partial x_1}{\partial q_n} \\ \dots & \dots & \dots \\ \frac{\partial x_n}{\partial q_1} & \dots & \frac{\partial x_n}{\partial q_n} \end{bmatrix}$$

- Basic use:

$$d\underline{x} = Jd\underline{q}$$

Resolved Rate Control

- By the Chain Rule of Differentiation:

$$\frac{\mathrm{d}\underline{x}}{\mathrm{d}t} = \left(\frac{\partial \underline{x}}{\partial \underline{q}}\right)\left(\frac{\mathrm{d}\underline{q}}{\mathrm{d}t}\right)$$

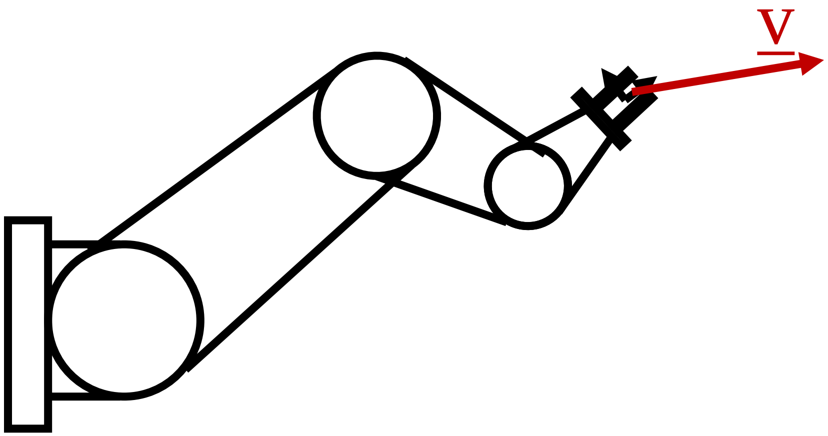
- This is just:

$$\dot{\underline{x}} = J\dot{\underline{q}}$$

- Suppose x is the position of the end effector. This is:
  - Nonlinear in joint variables
  - Linear in the joint rates!

What does that imply?

Resolved Rate Control



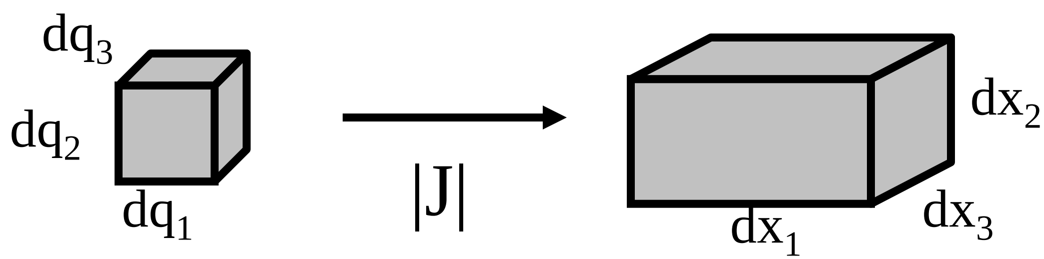
- Direct mapping from desired velocity onto joint rates!

$$\dot{\underline{q}} = J(\underline{q})^{-1}\dot{\underline{x}}$$

- Singularity (infinite joint rates) occurs:
  - at points where two different inverse kinematic solutions converge
  - when joint axes become aligned or parallel
  - when the boundaries of the workspace are reached

2.4.3.5 Jacobian Determinant

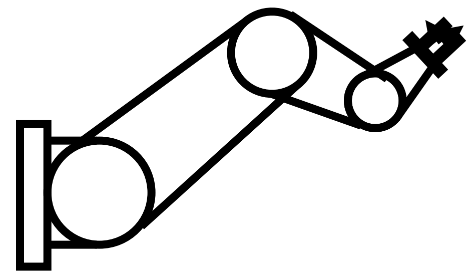
$$(dx_1 dx_2 \ldots dx_n) = |J|(dq_1 dq_2 \ldots dq_m)$$



- Relates differential volumes in task space to differential volumes in configuration space.
  - Used in calculus for double, triple, etc integrals.
- We will use this later to figure out when landmarks are in unfavorable configurations for navigation.

2.4.3.4 Example: 3 Link Planar Manipulator

- Forward kinematics:



$$x = (c_{123}L_3 + c_{12}L_2 + c_1L_1)$$
$$y = (s_{123}L_3 + s_{12}L_2 + s_1L_1)$$
$$\psi = \psi_1 + \psi_2 + \psi_3$$

- Differentiate wrt ψ_1, ψ_2, and ψ_3.

$$\dot{x} = -(s_{123}\dot{\psi}_{123}L_3 + s_{12}\dot{\psi}_{12}L_2 + s_1\dot{\psi}_1L_1)$$
$$\dot{y} = (c_{123}\dot{\psi}_{123}L_3 + c_{12}\dot{\psi}_{12}L_2 + c_1\dot{\psi}_1L_1)$$
$$\dot{\psi} = (\dot{\psi}_1 + \dot{\psi}_2 + \dot{\psi}_3)$$

- In matrix form:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} (-s_{123}L_3 - s_{12}L_2 - s_1L_1) & (-s_{123}L_3 - s_{12}L_2) & -s_{123}L_3 \\ (c_{123}L_3 + c_{12}L_2 + c_1L_1) & (c_{123}L_3 + c_{12}L_2) & c_{123}L_3 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \dot{\psi}_1 \\ \dot{\psi}_2 \\ \dot{\psi}_3 \end{bmatrix}$$

Outline

- 2.4 Kinematics of Mechanisms
  - 2.4.1 Forward Kinematics
  - 2.4.2 Inverse Kinematics
  - 2.4.3 Differential Kinematics
  - Summary
- 2.5 Orientation and Angular Velocity

Summary

- We can model the forward kinematics of mechanisms by
  - embedding frames in rigid bodies
  - employing the fundamental orthogonal operator matrices
  - employing a few rules for writing them in the right order to represent a mechanism.
- The DH convention consists of a special compound orthogonal transform and a few rules for orienting the link frames.
  - These can be used just like the fundamental orthogonal operator matrices to do forward kinematics.

Summary

- Inverse kinematics requires more skill and relies on rewriting the forward equations in an attempt to isolate unknowns.
- Various derivatives of kinematic transforms can be taken and each has some use.

Outline

- 2.4 Kinematics of Mechanisms
- 2.5 Orientation and Angular Velocity
  - 2.5.1 Orientation in Euler Angle Form
  - 2.5.2 Angular Rates and Small Angles
  - 2.5.3 Angular Velocity and Orientation Rates in Euler Angle Form
  - 2.5.4 Angular Velocity and Orientation Rates in Angle-Axis Form
  - 2.5.5 Summary

Definitions

- Yaw ψ = rotation about vertical axis
- Pitch θ = rotation about sideways axis
- Roll ϕ = rotation about forward axis.
- Attitude = roll & pitch $[\phi\ \theta]$
- Orientation = attitude + yaw $[\phi\ \theta\ \psi]$
- Pose = position + orientation $[x\ y\ z\ \phi\ \theta\ \psi]$
- Azimuth = yaw (for a pointing device)
- Elevation = pitch (for a pointing device)
- Heading = angle of path tangent. Sometimes same as yaw. Sometimes not.

Definitions

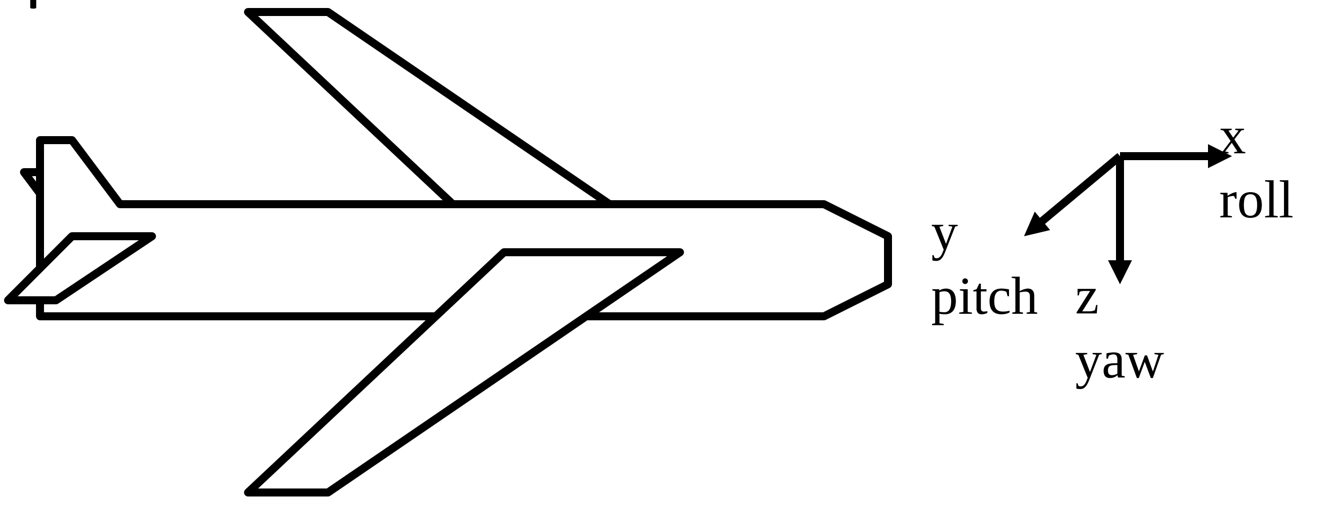
- Pose = position & orientation

- In 2D: $\begin{bmatrix} x & y & \psi \end{bmatrix}^T$

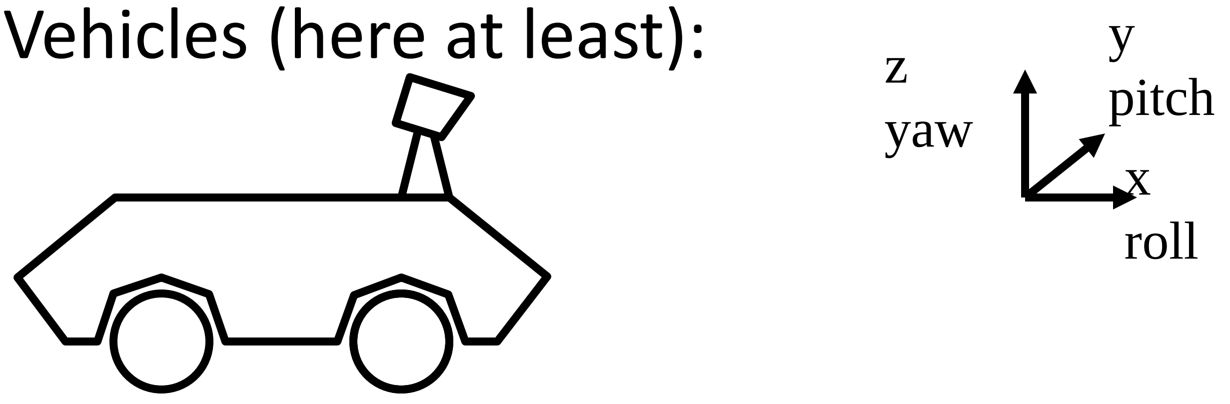
- In 3D: $\begin{bmatrix} x & y & z & \theta & \phi & \psi \end{bmatrix}^T$

2.5.1.1. Axis Conventions

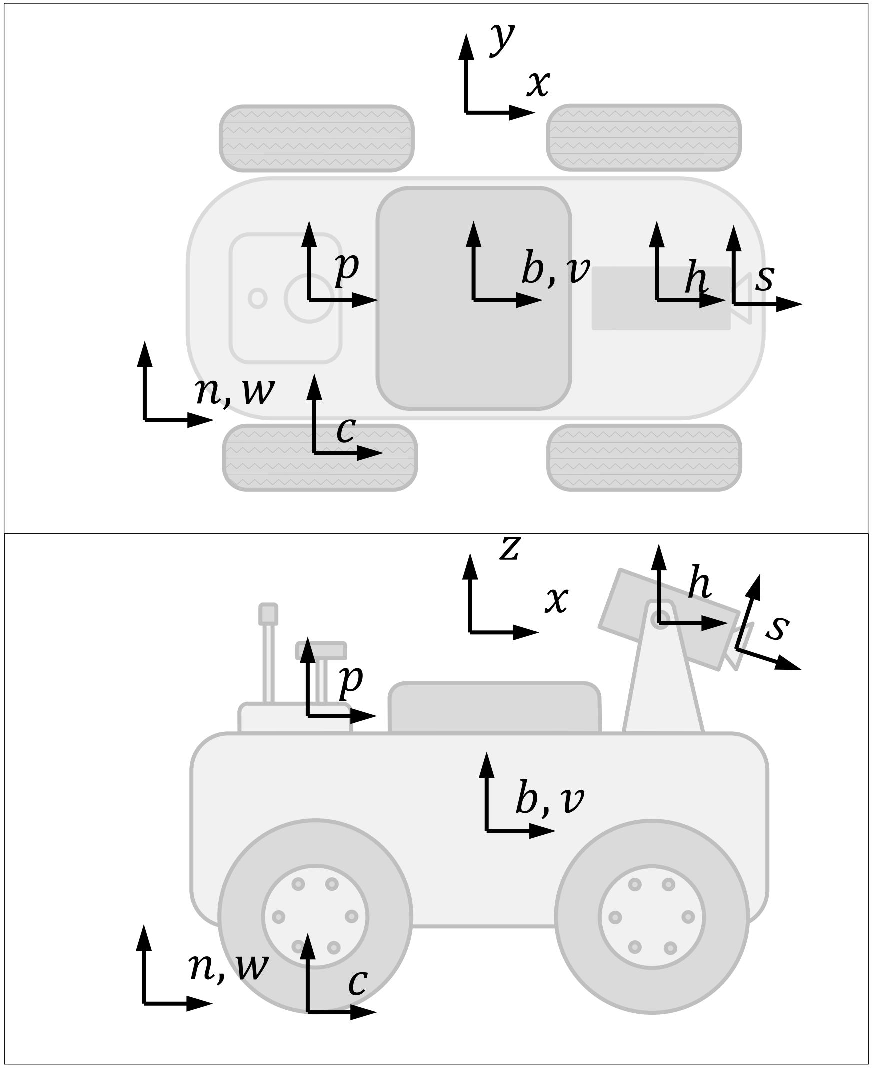
- Aerospace:



- Ground Vehicles (here at least):



2.5.1.2 Frame Assignment



Letter	Name
n,w	Navigation, world
w	world
p	positioner
b,v	body, vehicle
h	head
s	sensor
c	wheel contact

2.5.1.3 The RPY Transform

- Similar to DH Matrix but encodes 6 dof:
- Aligning operations that move ‘a’ into coincidence with ‘b’:
  - translate along the (x,y,z) axes of frame ‘a’ by (u,v,w) until its origin coincides with that of frame ‘b’
  - rotate about the new z axis by an angle ψ called yaw
  - rotate about the new y axis by an angle θ called pitch
  - rotate about the new x axis by an angle ϕ called roll

RPY Forward Kinematics

$$T_b^a = Trans(u, v, w)Rotz(\psi)Roty(\theta)Rotx(\phi)$$

$$T_b^a = \begin{bmatrix} 1 & 0 & 0 & u \\ 0 & 1 & 0 & v \\ 0 & 0 & 1 & w \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\psi & -s\psi & 0 & 0 \\ s\psi & c\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta & 0 & s\theta & 0 \\ 0 & 1 & 0 & 0 \\ -s\theta & 0 & c\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\phi & -s\phi & 0 \\ 0 & s\phi & c\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_b^a = \begin{bmatrix} c\psi c\theta & (c\psi s\theta s\phi - s\psi c\phi) & (c\psi s\theta c\phi + s\psi s\phi) & u \\ s\psi c\theta & (s\psi s\theta s\phi + c\psi c\phi) & (s\psi s\theta c\phi - c\psi s\phi) & v \\ -s\theta & c\theta s\phi & c\theta c\phi & w \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.5.1.4 RPY Transform Inverse Kinematics

$$T_b^a = Trans(u,v,w)Rotz(\psi)Rotx(\theta)Roty(\phi)$$

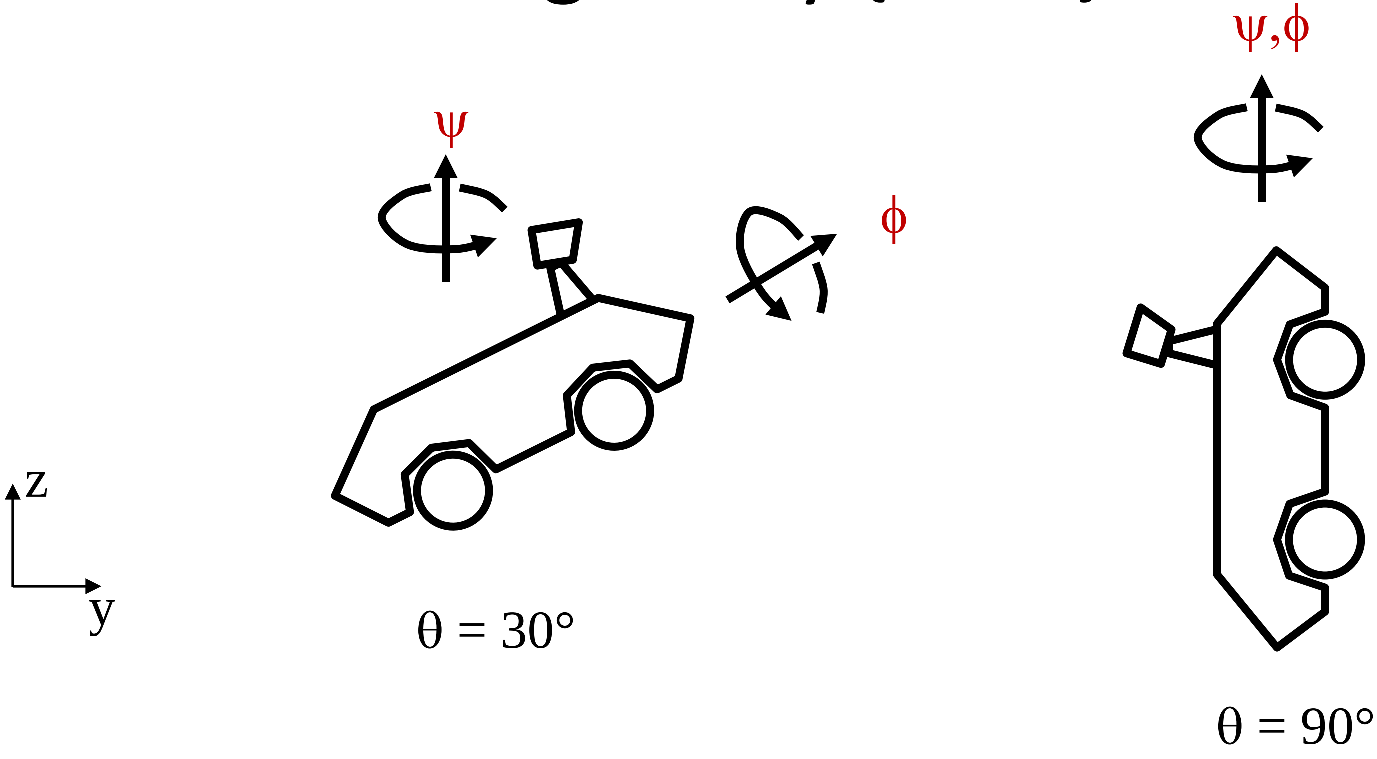
$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\psi c\theta & (c\psi s\theta s\phi - s\psi c\phi) & (c\psi s\theta c\phi + s\psi s\phi) & u \\ s\psi c\theta & (s\psi s\theta s\phi + c\psi c\phi) & (s\psi s\theta c\phi - c\psi s\phi) & v \\ -s\theta & c\theta s\phi & c\theta c\phi & w \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

body x axis in world

$$\psi = \text{atan2}(r_{21}, r_{11})$$

- Yaw can be determined from a vector aligned with the body x axis expressed in world coordinates .
- Is there another solution?
  - Only if there are 2 solutions for θ.
- What if cθ is zero?

Singularity {cθ=0}



- Yaw and roll are the same rotation when pitch is 90 °.
- Only their sum can be determined in inverse kinematics. Set the value of either.

2.5.1.4 RPY Transform Inverse Kinematics

$$[Rotz(\psi)]^{-1}[Trans(u, v, w)]^{-1}T_b^a = Roty(\theta)Rotx(\phi)$$

$$\begin{bmatrix} (r_{11}c\psi + r_{21}s\psi) & (r_{12}c\psi + r_{22}s\psi) & (r_{13}c\psi + r_{23}s\psi) & 0 \\ (-r_{11}s\psi + r_{21}c\psi) & (-r_{12}s\psi + r_{22}c\psi) & (-r_{13}s\psi + r_{23}c\psi) & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\theta & s\theta s\phi & s\theta c\phi & 0 \\ 0 & c\phi & -s\phi & 0 \\ -s\theta & c\theta s\phi & c\theta c\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

body x axis wrt yawed frame

$$\theta = \text{atan2}(-r_{31}, r_{11}c\psi + r_{21}s\psi)$$

- Pitch can also be determined from a vector aligned with the body x axis, expressed in "yawed" coordinates.
- Could get roll from this step too.

2.5.1.4 RPY Transform Inverse Kinematics

$$[\mathrm{Rotx}(\theta)]^{-1}[\mathrm{Rotz}(\psi)]^{-1}[\mathrm{Trans}(u,v,w)]^{-1}T_b^a = \mathrm{Roty}(\phi)$$

$$\begin{bmatrix} . & c\theta(r_{12}c\psi + r_{22}s\psi) - r_{32}s\theta & . & 0 \\ . & (-r_{12}s\psi + r_{22}c\psi) & . & 0 \\ . & s\theta(r_{12}c\psi + r_{22}s\psi) + r_{32}c\theta & . & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\phi & -s\phi & 0 \\ 0 & s\phi & c\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\phi = \mathrm{atan2}(s\theta(r_{12}c\psi + r_{22}s\psi) + r_{32}c\theta, -r_{12}s\psi + r_{22}c\psi)$$

- Roll can be determined from a vector aligned with the body y axis, expressed in yawed-pitched coordinates.

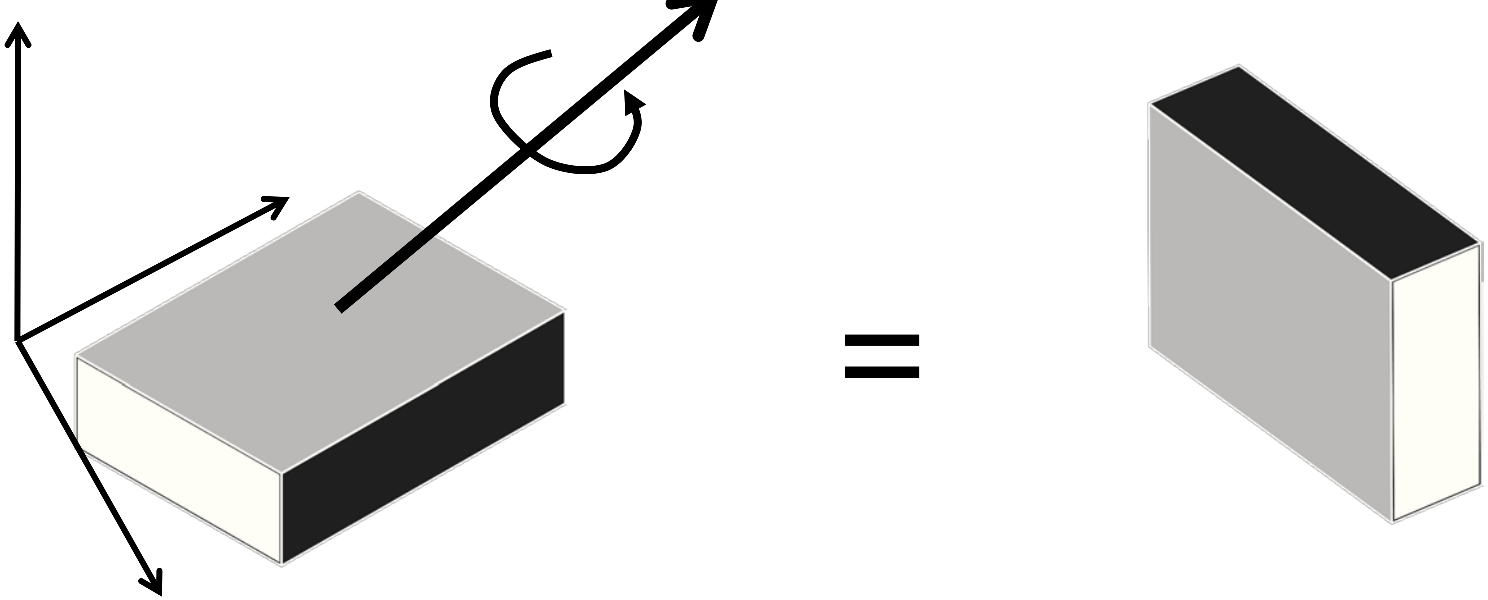
Outline

- 2.4 Kinematics of Mechanisms
- 2.5 Orientation and Angular Velocity
  - 2.5.1 Orientation in Euler Angle Form
  - <u>2.5.2 Angular Rates and Small Angles</u>
  - 2.5.3 Angular Velocity and Orientation Rates in Euler Angle Form
  - 2.5.4 Angular Velocity and Orientation Rates in Angle-Axis Form
  - 2.5.5 Summary

2.5.2.1 Euler's Theorem

- All rigid body rotations in 3D can be represented as a single rotation by some amount (angle) about a fixed axis:

$$\mathrm{Rot}(\phi, \theta, \psi) \leftrightarrow (\hat{\Theta}, \Theta)$$



2.5.2.2 Rotation Vector Representation

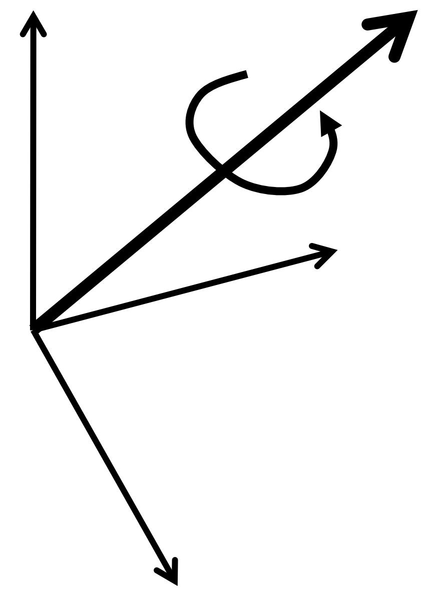
- Idea: Use a unit vector scaled by the magnitude of the rotation:

$$\underline{\Theta} = \begin{bmatrix} \theta_x & \theta_y & \theta_z \end{bmatrix}^T$$

- The axis of rotation is:

$$\hat{\underline{\Theta}} = \underline{\Theta} / |\underline{\Theta}|$$

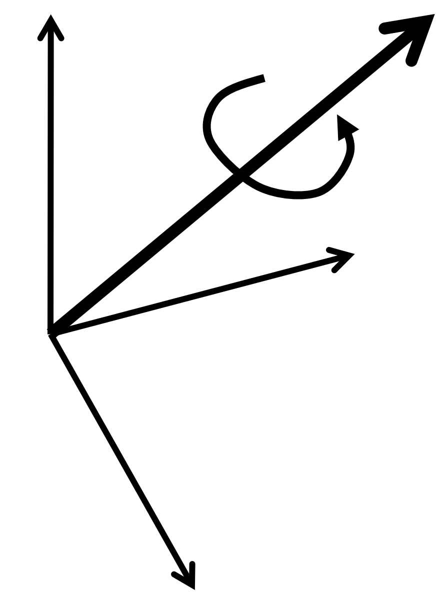
- This is not a true vector in the linear algebra sense.
  - Cannot be added vectorially to produce a meaningful result.



2.5.2.3 Relationship to Angular Velocity

- Suppose the rotation angle is differentially small.

$$d\underline{\Theta} = \begin{bmatrix} d\theta_x & d\theta_y & d\theta_z \end{bmatrix}^T$$



- The angular velocity is such that:

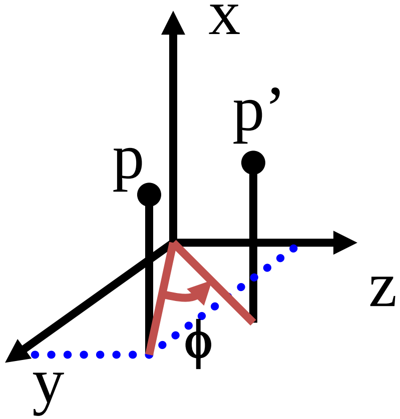
$$\underline{\omega} = \frac{|d\underline{\Theta}|}{dt}\left(\frac{d\underline{\Theta}}{|d\underline{\Theta}|}\right) = \omega d\hat{\underline{\Theta}} = \omega\hat{\omega}$$

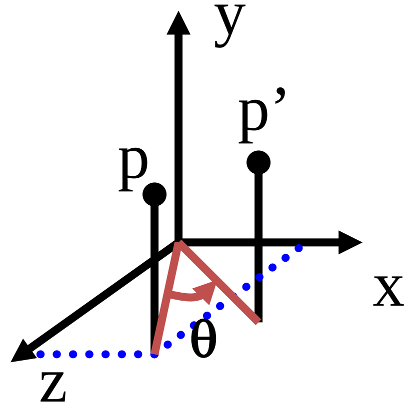
- This is a true vector in the linear algebra sense.
- Conversely, the rotation vector is:

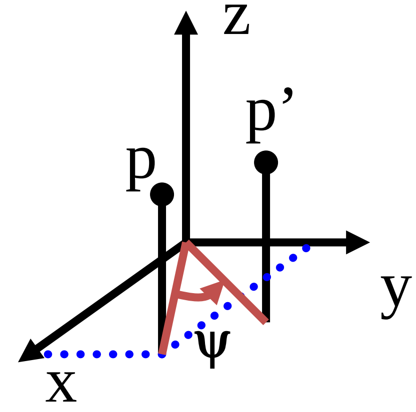
$$d\underline{\Theta} = \underline{\omega}\,dt = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T dt$$

Carnegie Mellon
THE ROBOTICS INSTITUTE

Recall the Rot() Operators







$$\mathrm{Rotx}(\phi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\phi & -s\phi & 0 \\ 0 & s\phi & c\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathrm{Roty}(\theta) = \begin{bmatrix} c\theta & 0 & s\theta & 0 \\ 0 & 1 & 0 & 0 \\ -s\theta & 0 & c\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathrm{Rotz}(\psi) = \begin{bmatrix} c\psi & -s\psi & 0 & 0 \\ s\psi & c\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.5.2.4 Rotations Through Small Angles

- For small angles:

$$\sin(\theta) \approx \theta$$

$$\cos(\theta) \approx 1$$

- Substituting into the Rot() operators:

$$\mathrm{Rotx}(\delta\phi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -\delta\phi & 0 \\ 0 & \delta\phi & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathrm{Roty}(\delta\theta) = \begin{bmatrix} 1 & 0 & \delta\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\delta\theta & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathrm{Rotz}(\delta\psi) = \begin{bmatrix} 1 & -\delta\psi & 0 & 0 \\ \delta\psi & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Useful for computing rapid differentials of complicated kinematic expressions.
- Turns out that while 3D rotations do not commute, <u>differential 3D rotations do commute</u>.

General Differential Rotations

- Consider a 3D composite differential rotation:

$$\delta\underline{\Theta} = \begin{bmatrix}\delta\phi & \delta\theta & \delta\psi\end{bmatrix}^T$$

- Substituting into the differential Rot() operators and cancelling H.O.T:

$$\text{Rotx}(\delta\phi)\text{Roty}(\delta\theta)\text{Rotz}(\delta\psi) = \text{Rot}(\delta\underline{\Theta}) = \begin{bmatrix} 1 & -\delta\psi & \delta\theta & 0 \\ \delta\psi & 1 & -\delta\phi & 0 \\ -\delta\theta & \delta\phi & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- To first order, the result <u>does not depend on the order</u> the rotations are applied.

2.5.2.4 Skew Matrices

- Last result can be written in terms of a skew matrix:

$$\mathrm{Rot}(\delta\underline{\Theta}) = I + [\delta\underline{\Theta}]^{X}$$

- Where:

$$Skew(\delta\underline{\Theta}) = [\delta\underline{\Theta}]^{\times} = \begin{bmatrix} 0 & -\delta\psi & \delta\theta & 0 \\ \delta\psi & 0 & -\delta\phi & 0 \\ -\delta\theta & \delta\phi & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Beware:
TYPO in book!
This is correct

Outline

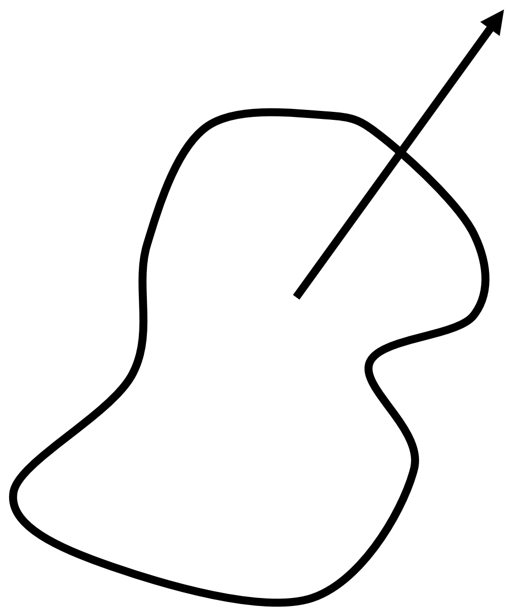
- 2.4 Kinematics of Mechanisms
- 2.5 Orientation and Angular Velocity
  - 2.5.1 Orientation in Euler Angle Form
  - 2.5.2 Angular Rates and Small Angles
  - <u>2.5.3 Angular Velocity and Orientation Rates in Euler Angle Form</u>
  - 2.5.4 Angular Velocity and Orientation Rates in Angle-Axis Form
  - 2.5.5 Summary

2.5.3.1 Relation to Euler Angle rates

- The pitch θ, yaw ψ, and roll ϕ angles are called <u>Euler</u> angles.
- Their rates are not exactly the angular velocity vector.

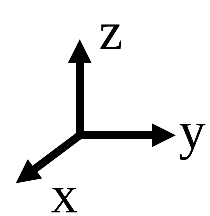
$$\frac{d}{dt}\begin{bmatrix}\theta \\ \phi \\ \psi\end{bmatrix}$$

measured about
moving axes



$$\underline{\omega} = \begin{bmatrix}\omega_x \\ \omega_y \\ \omega_z\end{bmatrix}$$

measured about
fixed axes



Relation to Euler Angle rates

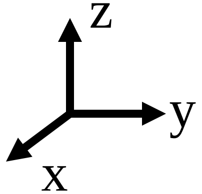
- Use Chain Rule. Convert coordinates for each component into the body frame. Note use of <u>transform</u> matrices (from world to body).

$$\underline{\omega}^b = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + rot(x,\phi)\begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + rot(x,\phi)rot(y,\theta)\begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix}$$

"unrolled"

"unrolled" "unpitched"

Note Lowercase "r" in rot

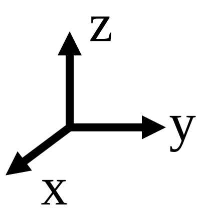


$$\underline{\omega}^b = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} \dot{\phi} - s\theta\dot{\psi} \\ c\phi\dot{\theta} + s\phi c\theta\dot{\psi} \\ -s\phi\dot{\theta} + c\phi c\theta\dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & s\phi c\theta \\ 0 & -s\phi & c\phi c\theta \end{bmatrix}\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

Why express ω in the body frame?

Inverted form

- Most useful in inverted form:



$$\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} = \begin{bmatrix}\omega_x + \omega_y s\phi t\theta + \omega_z c\phi t\theta \\ \omega_y c\phi - \omega_z s\phi \\ \omega_y \frac{s\phi}{c\theta} + \omega_z \frac{c\phi}{c\theta}\end{bmatrix} = \begin{bmatrix}1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & \frac{s\phi}{c\theta} & \frac{c\phi}{c\theta}\end{bmatrix}\begin{bmatrix}\omega_x\\ \omega_y\\ \omega_z\end{bmatrix}$$

- because: $\omega_z c\phi + \omega_y s\phi = c\theta\dot{\psi}$
- because: $\omega_y c\phi - \omega_z s\phi = \dot{\theta}$

Why convert ω to Euler Angle Rates?

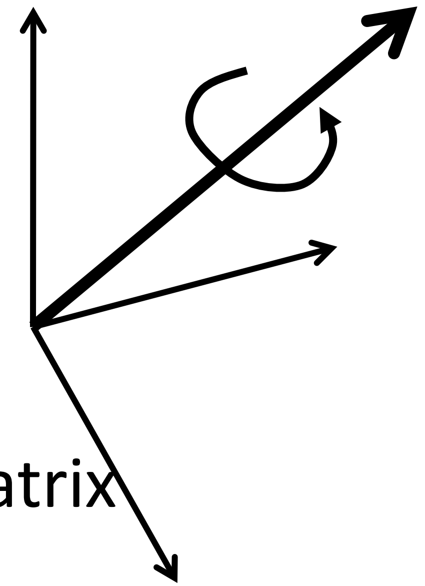
Outline

- 2.4 Kinematics of Mechanisms
- 2.5 Orientation and Angular Velocity
  - 2.5.1 Orientation in Euler Angle Form
  - 2.5.2 Angular Rates and Small Angles
  - 2.5.3 Angular Velocity and Orientation Rates in Euler Angle Form
  - <u>2.5.4 Angular Velocity and Orientation Rates in Angle-Axis Form</u>
  - 2.5.5 Summary

2.5.4.1 Angular Velocity as a Skew Matrix

- Recall the skew matrix formed from such a small 3D rotation.

$$\text{Skew}(d\underline{\Theta}) = [d\underline{\Theta}]^{X} = \begin{bmatrix} 0 & -d\theta_z & d\theta_y & 0 \\ d\theta_z & 0 & -d\theta_x & 0 \\ -d\theta_y & d\theta_x & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$



- Dividing by dt permits an equivalent matrix to be formed from the angular velocity vector:

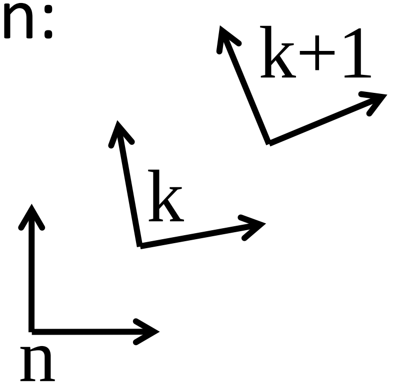
$$\Omega = \text{Skew}\left(\frac{d\underline{\Theta}}{dt}\right) = [\underline{\omega}]^{X} = \begin{bmatrix} 0 & -\omega_z & \omega_y & 0 \\ \omega_z & 0 & -\omega_x & 0 \\ -\omega_y & \omega_x & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

This is very closely related to the derivative of a rotation matrix.

2.5.4.2 Time Derivative of a Rotation Matrix

- Let the matrix R_k^n track the orientation of a moving frame k with respect to frame n.
- For small time steps, the update to time k+1 is a composition with a small perturbation:

$$R_{k+1}^n = R_k^n R_{k+1}^k = R_k^n [\{ I + [\delta \underline{\Theta}]^X \}]$$



- By definition of derivative:

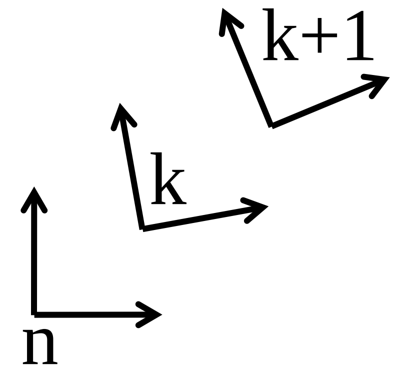
$$\dot{R}_k^n = \lim_{\delta t \to 0} \frac{\delta R_k^n}{\delta t} = \lim_{\delta t \to 0} \frac{[R_k^n(t + \delta t) - R_k^n(t)]}{\delta t}$$

2.5.4.2 Time Derivative of a Rotation Matrix

- Recall from last slide:

$$\dot{R}_k^n = \lim_{\delta t \to 0} \frac{\delta R_k^n}{\delta t} = \lim_{\delta t \to 0} \frac{[R_k^n(t+\delta t) - R_k^n(t)]}{\delta t}$$

- Write this in terms of a perturbation.



$$\dot{R}_k^n = \lim_{\delta t \to 0} \frac{[R_k^n[\{I + [\delta \underline{\Theta}]^X\}] - R_k^n]}{\delta t}$$

- Hence, since R_k^n is fixed:

$$\dot{R}_k^n = \lim_{\delta t \to 0} \frac{R_k^n[\delta \underline{\Theta}]^X}{\delta t} = R_k^n \lim_{\delta t \to 0} \frac{[\delta \underline{\Theta}]^X}{\delta t}$$

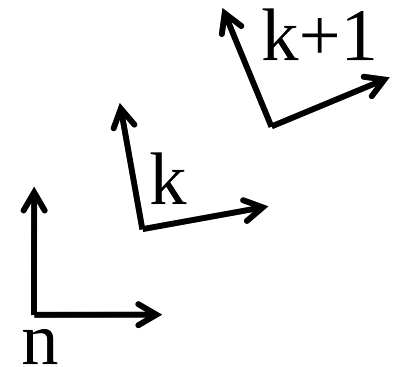
- Hence:

$$\dot{R}_k^n = R_k^n \left\{ \frac{d[\underline{\Theta}]^X}{dt} \right\} = R_k^n \Omega_k$$

Time Derivative of a Rotation Matrix

- In other words, we have the remarkable result that:

$$\boxed{\dot{R}^n_k = R^n_k \Omega_k}$$



- This holds so long as the components of the skew matrix are expressed in the moving frame k:

$$\Omega^k = \text{Skew}({}^k\underline{\omega}^n_k) = \begin{bmatrix} 0 & -\omega_z & \omega_y & 0 \\ \omega_z & 0 & -\omega_x & 0 \\ -\omega_y & \omega_x & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Why?

The perturbation was expressed as a rotation matrix composition.

2.5.4.3 Direction Cosines from Angular Velocity

- Once again, compound the angular perturbations thus:

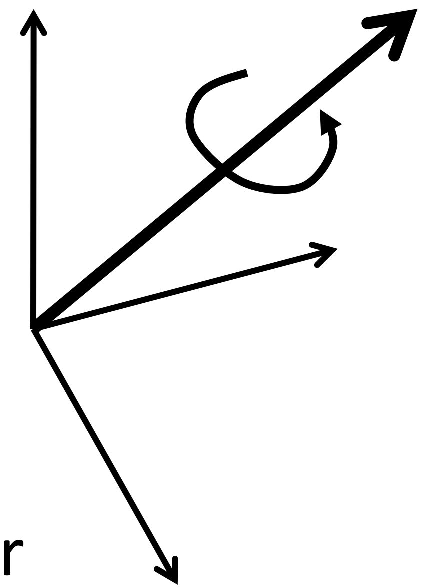
$$R^n_{k+1} = R^n_k R^k_{k+1}$$

- Based on previous derivative formula:

$$R^k_{k+1} = \int_{t_k}^{t_{k+1}} \Omega^k d\tau$$

- Assuming the integrand is constant over a small time step, we have:

$$R^n_{k+1} = R^n_k \exp\{[d\underline{\Theta}]^X\}$$



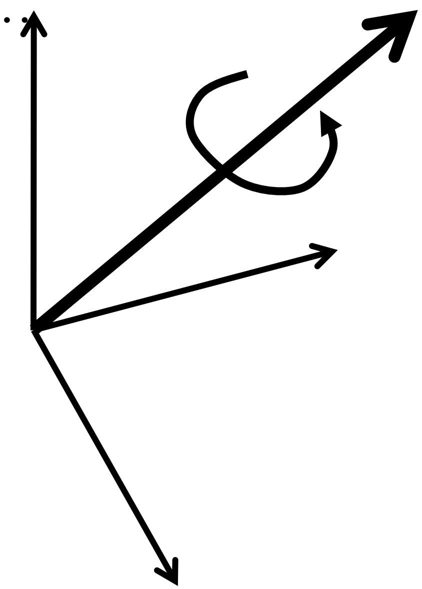
Recall:

$$\int_0^t A d\tau = \exp\{At\}$$

Direction Cosines from Angular Velocity

- This can be written in terms of two simple functions because for any $\underline{v}$:

$$\exp\{[\underline{v}]^X\} = I + [\underline{v}]^X + \frac{([\underline{v}]^X)^2}{2!} + \frac{([\underline{v}]^X)^3}{3!} + \frac{([\underline{v}]^X)^4}{4!} + \dots$$



- But its easy to show that:

$$[[\underline{v}]^X]^3 = -v^2\underline{v}^X \qquad [[\underline{v}]^X]^4 = -v^2[\underline{v}^X]^2$$

- Etc. So, this simplifies to:

$$\exp\{\underline{v}^X\} = I + f_1(v)[\underline{v}]^X + f_2(v)([\underline{v}]^X)^2$$

$$f_1(v) = \frac{\sin v}{v} \qquad f_2(v) = \frac{(1-\cos v)}{v^2}$$

Direction Cosines from Angular Velocity

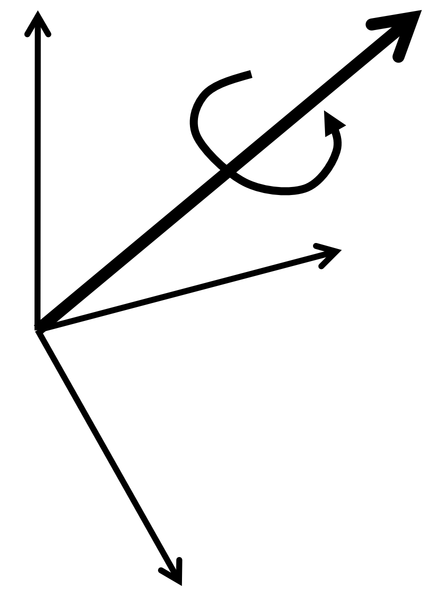
- Hence, the direct transformation from angular velocity to direction cosines is the recursion:

$$R^n_{k+1} = R^n_k R^k_{k+1}$$

$$R^k_{k+1} = I + f_1(\delta\Theta)[\delta\underline{\Theta}]^{\times} + f_2(\delta\Theta)([\delta\underline{\Theta}]^{\times})^2$$

$$f_1(\delta\Theta) = \frac{\sin\delta\Theta}{\delta\Theta} \qquad f_2(\delta\Theta) = \frac{(1-\cos\delta\Theta)}{\delta\Theta^2}$$

- Where dt is the time step, $d\Theta = |d\underline{\Theta}|$ and $d\underline{\Theta} = \underline{\omega}\, dt$.
- Advantage: You don't need to solve for the Euler angles.



Outline

- 2.4 Kinematics of Mechanisms
- 2.5 Orientation and Angular Velocity
  - 2.5.1 Orientation in Euler Angle Form
  - 2.5.2 Angular Rates and Small Angles
  - 2.5.3 Angular Velocity and Orientation Rates in Euler Angle Form
  - 2.5.4 Angular Velocity and Orientation Rates in Angle-Axis Form
  - <u>2.5.5 Summary</u>

Summary

- The RPY matrix is yet another compound orthogonal operator matrix. Unlike the DH matrix, it has 6 dof, so it is completely general.
- Angular velocity is the time derivative of the rotation vector.
- Compositions of small angle rotations behave commutatively.
- Angular velocity and small rotations can be expressed as skew matrices.
- Angular velocity is related in a complicated manner to the rates of roll, pitch, and yaw angles.
- The skew of angular velocity is related in a more elegant manner to the rate of the rotation matrix.