CHAPTER 5. JACOBIANS: VELOCITIES AND STATIC FORCES

Part I: Velocities – linear and angular (Sections 5.1 ~ 5.6)
Part II: Jacobians – differential kinematics (Sections 5.7 ~ 5.8)
Part III: Robot statics (Sections 5.9 ~ 5.11)

- Attach a coordinate system (frame) to a body
  → Motion of rigid bodies: motion of **frames** relative to one another

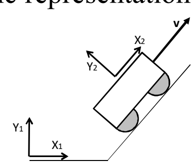
Differentiation of Position Vector (of a Point)

- Derivative of a vector **Q** relative to frame $\{B\}$: ${}^{B}\mathbf{V}_{Q} = \frac{d}{dt}\,{}^{B}\mathbf{Q} = \lim_{\Delta t \to 0} \frac{{}^{B}\mathbf{Q}(t+\Delta t) - {}^{B}\mathbf{Q}(t)}{\Delta t} = {}^{B}({}^{B}\mathbf{V}_{Q})$
  (Indicate the frame in which the vector is differentiated.)

- A velocity vector is described in terms of a reference frame which is noted with a leading superscript.
  → When expressed in terms of frame $\{A\}$: ${}^{A}({}^{B}\mathbf{V}_{Q}) = \frac{{}^{A}d}{dt}\,{}^{B}\mathbf{Q}$

- Note: Numerical values describing a (linear or translational) velocity vector depend on **two** frames
  Frame (of observer) with respect to which the differentiation is done ($\{B\}$) → vector construction
  Frame (of writer) in which the resulting velocity vector is expressed ($\{A\}$) → vector components

- Dual-superscript notation: **Two** reference frames for description of kinematic vectors (linear position/velocity/acceleration of a point and angular velocity/acceleration of a frame)
  - **Defined** as viewed by an observer fixed in a reference frame: "relative to" or "with respect to" *observer*'s frame → Geometric vector
  - **Resolved** into components with respect to a reference frame: "referred to," "expressed in," or "written in" *writer*'s frame → Algebraic representation of the geometric vector



- ${}^{B}({}^{B}\mathbf{V}_{Q}) = {}^{B}\mathbf{V}_{Q}$
- $\boxed{{}^{A}({}^{B}\mathbf{V}_{Q}) = {}^{A}R_{B}\,{}^{B}({}^{B}\mathbf{V}_{Q}) = {}^{A}R_{B}\,{}^{B}\mathbf{V}_{Q}}$ (use rotation matrix to change the reference frame)
  (Note: ${}^{A}R_{B}\,{}^{B}\mathbf{V}_{Q} \neq {}^{A}\mathbf{V}_{Q}$)
- Velocity of origin of a frame $\{C\}$ relative to a universe reference frame $\{U\}$: $\boldsymbol{\upsilon}_{C} = {}^{U}\mathbf{V}_{CORG}$

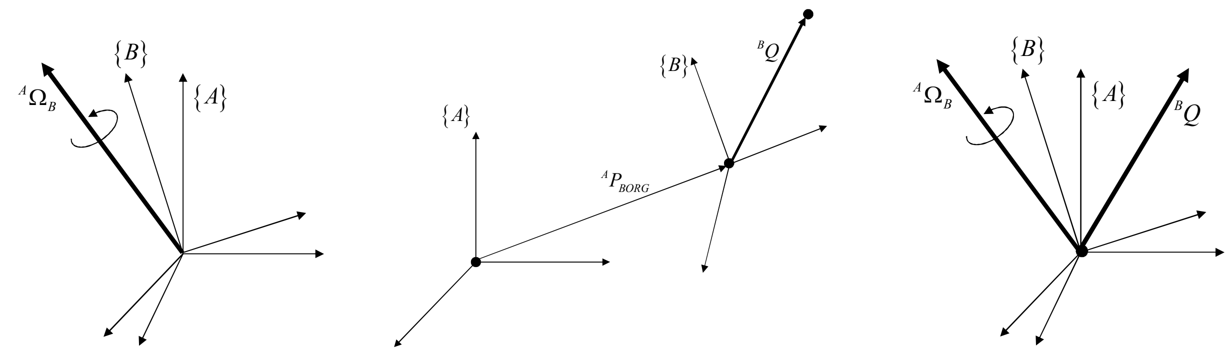
- Example 5.1 (Craig's 4th Ed.): (Do it yourself)

Angular Velocity Vector (of a Body)

- Always attach a frame to each rigid body → angular velocity describes rotational motion of a frame

- ${}^{A}\mathbf{\Omega}_{B}$: rotation of frame $\{B\}$ relative to $\{A\}$
  Direction: instantaneous axis of rotation — Magnitude: rotation speed

- ${}^{C}({}^{A}\mathbf{\Omega}_B)$: angular velocity of frame $\{B\}$ relative to $\{A\}$ expressed in terms of frame $\{C\}$
- Angular velocity of a frame $\{C\}$ relative to a universe reference frame $\{U\}$: $\boldsymbol{\omega}_C = {}^{U}\mathbf{\Omega}_C$



Linear Velocity of Rigid Bodies
- Frame $\{B\}$ attached to a rigid body, and $\{A\}$ is fixed.
- Motion of point Q relative to $\{A\}$: due to ${}^{A}\mathbf{P}_{BORG}$ and ${}^{B}\mathbf{Q}$
- Assume relative orientation of $\{B\}$ and $\{A\}$ is constant.
- Linear velocity (assume constant ${}^{A}R_B$) of point Q in terms of $\{A\}$:
  ${}^{A}({}^{A}\mathbf{V}_Q) = {}^{A}({}^{A}\mathbf{V}_{BORG}) + {}^{A}R_B{}^{B}({}^{B}\mathbf{V}_Q)$ or equivalently, ${}^{A}\mathbf{V}_Q = {}^{A}\mathbf{V}_{BORG} + {}^{A}R_B{}^{B}\mathbf{V}_Q$

Rotational Velocity of Rigid Bodies
- Frames $\{B\}$ and $\{A\}$ with coincident origins (${}^{A}\mathbf{P}_{BORG} = \mathbf{0}$)
- Generally, vector $\mathbf{Q}$ also changes with respect to frame $\{B\}$.
- ${}^{A}\mathbf{V}_Q = \underbrace{{}^{A}({}^{B}\mathbf{V}_Q)}_{wrt\ \{B\}} + \underbrace{{}^{A}\mathbf{\Omega}_B \times {}^{A}\mathbf{Q}}_{rotation}$ (from undergraduate dynamics)

  => ${}^{A}\mathbf{V}_Q = {}^{A}R_B{}^{B}\mathbf{V}_Q + {}^{A}\mathbf{\Omega}_B \times {}^{A}R_B{}^{B}\mathbf{Q}$

  (Note: here and in the textbook, ${}^{A}\mathbf{Q}$ indicates ${}^{A}({}^{B}\mathbf{Q})$, and ${}^{B}\mathbf{Q}$ indicates ${}^{B}({}^{B}\mathbf{Q})$.)

General Linear and Rotational Velocity of Rigid Bodies
- Origins are not coincident
- General velocity of a vector in frame $\{B\}$ as seen from $\{A\}$:
  $\boxed{{}^{A}\mathbf{V}_Q = {}^{A}\mathbf{V}_{BORG} + {}^{A}R_B{}^{B}\mathbf{V}_Q + {}^{A}\mathbf{\Omega}_B \times {}^{A}R_B{}^{B}\mathbf{Q}}$

Rotation Matrix (= proper orthonormal matrix)
- $RR^T = I_3$ → $\dot{R}R^T + R\dot{R}^T = 0_3$ => $\dot{R}R^T + (\dot{R}R^T)^T = 0_3$
- **Angular velocity matrix**: $\boxed{S = \dot{R}R^T = \dot{R}R^{-1}}$ → $S + S^T = 0_3$ (_________________ matrix)

Rotating Reference Frame
- Fixed vector with respect to frame $\{B\}$: ${}^{B}\mathbf{P}$ → With respect to $\{A\}$: ${}^{A}\mathbf{P} = {}^{A}R_B{}^{B}\mathbf{P}$
- If frame $\{B\}$ rotates → ${}^{A}\mathbf{V}_P = {}^{A}\dot{\mathbf{P}} = {}^{A}\dot{R}_B{}^{B}\mathbf{P} = \underbrace{{}^{A}\dot{R}_B{}^{A}R_B^{-1}}_{={}^{A}S_B}{}^{A}\mathbf{P}$ => ${}^{A}\mathbf{V}_P = {}^{A}S_B{}^{A}\mathbf{P}$

- Let $S = \begin{bmatrix} 0 & -\Omega_z & \Omega_y \\ \Omega_z & 0 & -\Omega_x \\ -\Omega_y & \Omega_x & 0 \end{bmatrix}$

- **Angular velocity vector**: $\mathbf{\Omega} = \begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix}$ → describes motion of frame $\{B\}$ with respect to $\{A\}$

  => $S\mathbf{P} = \mathbf{\Omega} \times \mathbf{P}$ for any vector $\mathbf{P}$ => $\therefore\ {}^A\mathbf{V}_P = {}^A\mathbf{\Omega}_B \times {}^A\mathbf{P}$

- $\dot{R} = \lim\limits_{\Delta t \to 0} \dfrac{R(t+\Delta t) - R(t)}{\Delta t}$ and let $R(t+\Delta t) = R_K(\Delta\theta)R(t)$ (why?) => $\dot{R} = \left(\lim\limits_{\Delta t \to 0} \dfrac{R_K(\Delta\theta) - I_3}{\Delta t} \right) R(t)$

- Recall: for ${}^A\hat{\mathbf{K}} = \begin{bmatrix} k_x \\ k_y \\ k_z \end{bmatrix}$ → $R_K(\theta) = \begin{bmatrix} k_x k_x v\theta + c\theta & k_x k_y v\theta - k_z s\theta & k_x k_z v\theta + k_y s\theta \\ k_y k_x v\theta + k_z s\theta & k_y k_y v\theta + c\theta & k_y k_z v\theta - k_x s\theta \\ k_z k_x v\theta - k_y s\theta & k_z k_y v\theta + k_x s\theta & k_z k_z v\theta + c\theta \end{bmatrix}$

  For $\Delta\theta << 1$ → $R_K(\Delta\theta) = \begin{bmatrix} 1 & -k_z\Delta\theta & k_y\Delta\theta \\ k_z\Delta\theta & 1 & -k_x\Delta\theta \\ -k_y\Delta\theta & k_x\Delta\theta & 1 \end{bmatrix}$

  => $\dot{R} = (\lim\limits_{\Delta t \to 0} \dfrac{\begin{bmatrix} 0 & -k_z\Delta\theta & k_y\Delta\theta \\ k_z\Delta\theta & 0 & -k_x\Delta\theta \\ -k_y\Delta\theta & k_x\Delta\theta & 0 \end{bmatrix}}{\Delta t}) \cdot R(t) = \begin{bmatrix} 0 & -k_z\dot{\theta} & k_y\dot{\theta} \\ k_z\dot{\theta} & 0 & -k_x\dot{\theta} \\ -k_y\dot{\theta} & k_x\dot{\theta} & 0 \end{bmatrix} R(t)$

  $\therefore\ \dot{R}R^{-1} = \begin{bmatrix} 0 & -\Omega_z & \Omega_y \\ \Omega_z & 0 & -\Omega_x \\ -\Omega_y & \Omega_x & 0 \end{bmatrix}$

- $\boxed{\mathbf{\Omega} = \begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix} = \begin{bmatrix} k_x\dot{\theta} \\ k_y\dot{\theta} \\ k_z\dot{\theta} \end{bmatrix} = \dot{\theta}\hat{\mathbf{K}}}$ (← Definition of angular velocity vector)

  : At any instant the change in orientation of rotating frame is a rotation about **instantaneous axis of rotation** $\hat{\mathbf{K}}$ (unit vector). Speed of rotation $(\dot{\theta})$ is the angular velocity vector's magnitude.

Euler Angle Rates

- Rates of *Z-Y-Z* Euler angles: $\dot{\mathbf{\Theta}}_{Z'Y'Z'} = \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix}$

- Recall $S = \dot{R}R^T = \begin{bmatrix} 0 & -\Omega_z & \Omega_y \\ \Omega_z & 0 & -\Omega_x \\ -\Omega_y & \Omega_x & 0 \end{bmatrix} \rightarrow \begin{cases} \Omega_x = \dot{r}_{31}r_{21} + \dot{r}_{32}r_{22} + \dot{r}_{33}r_{23} \\ \Omega_y = \dot{r}_{11}r_{31} + \dot{r}_{12}r_{32} + \dot{r}_{13}r_{33} \\ \Omega_z = \dot{r}_{21}r_{11} + \dot{r}_{22}r_{12} + \dot{r}_{23}r_{13} \end{cases}$

  where entries r_{ij} $(i, j = 1, 2, 3)$ are functions of Euler angles, i.e., $r_{ij} = r_{ij}(\alpha, \beta, \gamma)$

  $\rightarrow \dot{r}_{ij} = \dfrac{d}{dt} r_{ij}(\alpha, \beta, \gamma) = \dot{\alpha}\dfrac{\partial r_{ij}}{\partial \alpha} + \dot{\beta}\dfrac{\partial r_{ij}}{\partial \beta} + \dot{\gamma}\dfrac{\partial r_{ij}}{\partial \gamma}$ $\quad \therefore\ \Omega_x, \Omega_y, \Omega_z$ are .... of $\dot{\alpha}, \dot{\beta}, \dot{\gamma}$

- $\mathbf{\Omega} = E_{Z'Y'Z'}(\mathbf{\Theta}_{Z'Y'Z'})\dot{\mathbf{\Theta}}_{Z'Y'Z'}$

  $E_{Z'Y'Z'}(\mathbf{\Theta}_{Z'Y'Z'})$: Jacobian matrix relating Euler angle rate vector and angular velocity vector

- Example 5.2 (Craig's 4th Ed.): $E_{Z'Y'Z'} = \begin{bmatrix} 0 & -s\alpha & c\alpha s\beta \\ 0 & c\alpha & s\alpha s\beta \\ 1 & 0 & c\beta \end{bmatrix}$ (use $R_{Z'Y'Z'}$ for derivation)

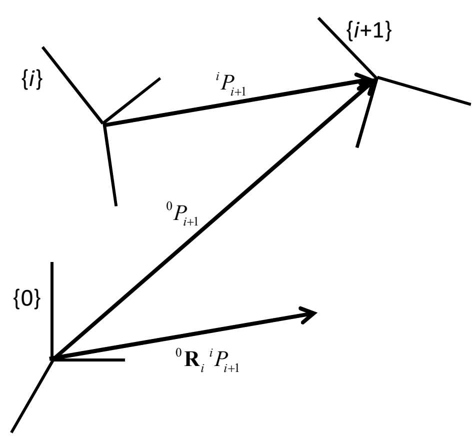
<u>Notation Convention Review</u>

- Generally, three scripts, including two reference frames for dual-superscript notation, are required to describe a kinematic vector **A**: linear position/velocity/acceleration (the subscript indicates a point) or angular velocity/acceleration (the subscript indicates a frame).

  $$\boxed{{}^{\text{[expressed in writer's frame]}}\underbrace{\left({}^{\text{[with respect to observer's frame]}}\mathbf{A}_{\text{[describe point or frame of interest]}}\right)}_{\text{a vector is defined}}}$$

  (Note: The frame of expression for rotation matrix ${}^{\text{[with respect to which frame]}}R_{\text{[describe frame of interest]}}$ is identical to that of the observer.)

- Note: ${}^0R_i\,{}^i\mathbf{P}_{i+1} \neq {}^0\mathbf{P}_{i+1}$ $\quad \therefore$ Even for position vector, all three scripts are required.

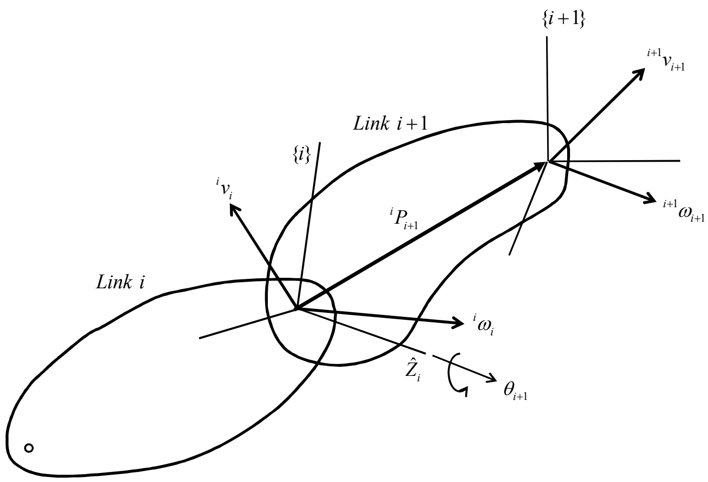


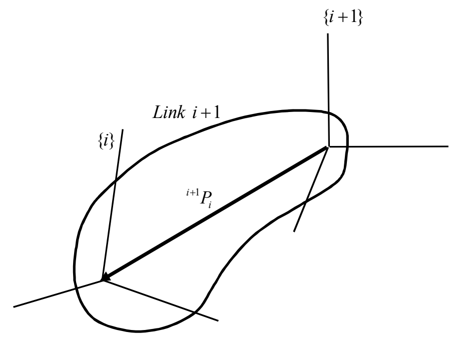
<u>Velocities (absolute) of Links</u>

Reference (or global) frame - link Frame {0}

v_i: linear velocity of origin of link Frame $\{i\}$ with respect to (or observer is at) the global frame

ω_i : angular velocity of link Frame $\{i\}$ with respect to (or observer is at) the global frame





- Each link as a rigid body → linear and angular velocity vectors written in its own link frame (rather than the global frame); note the notations here that ${}^{i}\upsilon_i = {}^{i}({}^{0}\upsilon_i)$ and ${}^{i}\omega_i = {}^{i}({}^{0}\omega_i)$
- [Link *i*+1 velocity] = [Link *i* velocity] + [relative velocity added by Joint *i*+1]
- Compute the velocities of each link starting from the base (**outward**) → Apply successively from link 0 to link *n*. → ${}^{n}\omega_n$ and ${}^{n}\upsilon_n$.
- Multiply by ${}^{0}R_n$ → expressed in global frame

Joint \ Velocity	Linear	Angular
Revolute	✓	✓
Prismatic	✓	✓

$$
\begin{array}{c}
{}^{0}\omega_0 = \mathbf{0} \\
\downarrow \\
\dot{\theta}_1 \longrightarrow {}^{1}\omega_1 = ? \\
\downarrow \\
\dot{\theta}_2 \longrightarrow {}^{2}\omega_2 = ? \\
\downarrow \\
\vdots \\
\downarrow \\
\dot{\theta}_n \longrightarrow {}^{n}\omega_n = ?
\end{array}
$$

<u>Link Velocities for Revolute Joint *i*+1</u>

- $\dot{\theta}_{i+1}\,{}^{i}\hat{Z}_i = {}^{i}\begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_{i+1} \end{bmatrix}$
- Angular velocity of Link *i*+1
  - with respect to Frame $\{i\}$: ${}^{i}\omega_{i+1} = {}^{i}\omega_i + \dot{\theta}_{i+1}\,{}^{i}\hat{Z}_i$
  - with respect to Frame $\{i+1\}$: $\boxed{{}^{i+1}\omega_{i+1} = {}^{i+1}R_i({}^{i}\omega_i + \dot{\theta}_{i+1}\,{}^{i}\hat{Z}_i)}$
- Proofs

  1) ${}^{k}({}^{k}\omega_{i+1}) = {}^{k}({}^{k}\omega_i) + \dot{\theta}_{i+1}\,{}^{k}\hat{Z}_i$ => ${}^{i}R_k \times [{}^{k}({}^{k}\omega_{i+1}) = {}^{k}({}^{k}\omega_i) + \dot{\theta}_{i+1}\,{}^{k}\hat{Z}_i]$

     => ${}^{i}({}^{k}\omega_{i+1}) = {}^{i}({}^{k}\omega_i) + \dot{\theta}_{i+1}\,{}^{i}\hat{Z}_i \quad \therefore {}^{i}\omega_{i+1} = {}^{i}\omega_i + \dot{\theta}_{i+1}\,{}^{i}\hat{Z}_i$

  2) ${}^{i+1}R_i \times [{}^{i}({}^{k}\omega_{i+1}) = {}^{i}({}^{k}\omega_i) + \dot{\theta}_{i+1}\,{}^{i}\hat{Z}_i]$ => ${}^{i+1}({}^{k}\omega_{i+1}) = {}^{i+1}R_i[{}^{i}({}^{k}\omega_i) + \dot{\theta}_{i+1}\,{}^{i}\hat{Z}_i]$

     $\therefore\ {}^{i+1}\omega_{i+1} = {}^{i+1}R_i({}^{i}\omega_i + \dot{\theta}_{i+1}\,{}^{i}\hat{Z}_i)$

- Linear velocity of origin of Frame $\{i+1\}$
  - with respect to Frame $\{i\}$: ${}^{i}\upsilon_{i+1} = {}^{i}\upsilon_i + {}^{i}\omega_{i+1} \times {}^{i}P_{i+1}$
  - with respect to Frame $\{i+1\}$: $\boxed{{}^{i+1}\upsilon_{i+1} = {}^{i+1}R_i({}^{i}\upsilon_i + {}^{i}\omega_{i+1} \times {}^{i}P_{i+1})}$

- Proofs

1) In ${}^{A}\mathbf{V}_{Q} = {}^{A}\mathbf{V}_{BORG} + {}^{A}R_{B}\,{}^{B}\mathbf{V}_{Q} + {}^{A}\mathbf{\Omega}_{B} \times {}^{A}R_{B}\,{}^{B}\mathbf{Q}$ (textbook equation (5.13)), let $\{A\} = \{K\}$, $\{B\} = \{i+1\}$, $\mathbf{Q}$ = origin of $\{i\}$, ${}^{A}P_{BORG} = {}^{K}P_{i+1}$, and ${}^{B}Q = {}^{i+1}P_{i}$.

$${}^{K}({}^{K}v_{i}) = {}^{K}({}^{K}v_{i+1}) + {}^{K}R_{i+1}\,\cancel{{}^{i+1}({}^{i+1}v_{i})} + {}^{K}({}^{K}\omega_{i+1}) \times {}^{K}R_{i+1}\,{}^{i+1}P_{i}$$

$${}^{i}R_{K} \times [{}^{K}({}^{K}v_{i}) = {}^{K}({}^{K}v_{i+1}) + {}^{K}({}^{K}\omega_{i+1}) \times {}^{K}R_{i+1}\,{}^{i+1}P_{i}] \Rightarrow {}^{i}({}^{K}v_{i+1}) = {}^{i}({}^{K}v_{i}) + {}^{i}({}^{K}\omega_{i+1}) \times \underbrace{\left(-{}^{i}R_{i+1}\,{}^{i+1}P_{i}\right)}_{= {}^{i}P_{i+1} \text{ from (2.44)}}$$

$$\therefore\ {}^{i}v_{i+1} = {}^{i}v_{i} + {}^{i}\omega_{i+1} \times {}^{i}P_{i+1}$$

2) ${}^{i+1}R_{i} \times [{}^{i}({}^{K}v_{i+1}) = {}^{i}({}^{K}v_{i}) + {}^{i}({}^{K}\omega_{i+1}) \times {}^{i}P_{i+1}] \Rightarrow \therefore\ {}^{i+1}v_{i+1} = {}^{i+1}R_{i}\,{}^{i}v_{i} + {}^{i+1}\omega_{i+1} \times {}^{i+1}R_{i}\,{}^{i}P_{i+1}$

<u>Link Velocities for Prismatic Joint *i*+1</u>

- Angular velocity of Link *i*+1 with respect to Frame $\{i+1\}$: $\boxed{{}^{i+1}\omega_{i+1} = {}^{i+1}R_{i}\,{}^{i}\omega_{i}}$
- Linear velocity of Frame $\{i+1\}$ origin with respect to Frame $\{i+1\}$:
$\boxed{{}^{i+1}\upsilon_{i+1} = {}^{i+1}R_{i}({}^{i}\upsilon_{i} + {}^{i}\omega_{i+1} \times {}^{i}P_{i+1} + \dot{d}_{i+1}\,{}^{i}\hat{Z}_{i})}$

<u>Link Velocities for Joint *i* (Unified Form)</u>

- In general, if Joint *i* is:

revolute $\theta_{i} = \tilde{\theta}_{i} + q_{i} \rightarrow \dot{\theta}_{i} = \dot{q}_{i}$ and $\dot{d}_{i} = 0$

prismatic $d_{i} = \tilde{d}_{i} + q_{i} \rightarrow \dot{d}_{i} = \dot{q}_{i}$ and $\dot{\theta}_{i} = 0$

Therefore, regardless of the joint type (revolute or prismatic),

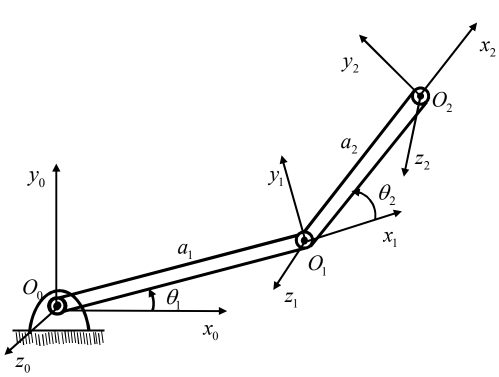
angular velocity of Link *i*: $\omega_{i} = \omega_{i-1} + \dot{\theta}_{i}\hat{Z}_{i-1}$

linear velocity of origin of Frame $\{i\}$: $\upsilon_{i} = \upsilon_{i-1} + \omega_{i} \times {}^{i-1}P_{i} + \dot{d}_{i}\hat{Z}_{i-1}$

(For simplicity, the frames of expression are omitted in the notations.)

<u>Example 5.3</u> (with standard DH convention)

A two-link manipulator with rotational joints is shown in the figure below. Calculate the (absolute linear) velocity of the tip (i.e., the origin of Frame {2}) of the arm as a function of joint rates (i.e., joint velocities). Give the answer in two forms—in terms of (i.e., written in) Frame {2} and Frame {0}.



Solution) Two different methods—with and without using the iterative formulas—are available.

Method 1 (using the iterative formulas):

$$ {}^0T_1 = \begin{bmatrix} c_1 & -s_1 & 0 & a_1c_1 \\ s_1 & c_1 & 0 & a_1s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad {}^1T_2 = \begin{bmatrix} c_2 & -s_2 & 0 & a_2c_2 \\ s_2 & c_2 & 0 & a_2s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad {}^0T_2 = \begin{bmatrix} c_{12} & -s_{12} & 0 & a_1c_1 + a_2c_{12} \\ s_{12} & c_{12} & 0 & a_1s_1 + a_2s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Use ${}^{i+1}\omega_{i+1} = {}^{i+1}R_i({}^i\omega_i + \dot{\theta}_{i+1}\,{}^i\hat{Z}_i)$ and ${}^{i+1}\upsilon_{i+1} = {}^{i+1}R_i({}^i\upsilon_i + {}^i\omega_{i+1} \times {}^iP_{i+1})$ sequentially from link to link to compute the velocity of the origin of each frame, starting from the base frame {0}, which has zero velocity:

$$ {}^0\omega_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \; {}^1\omega_1 = \begin{bmatrix} c_1 & s_1 & 0 \\ -s_1 & c_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix}, \; {}^2\omega_2 = \begin{bmatrix} c_2 & s_2 & 0 \\ -s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}\left(\begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_2 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 + \dot{\theta}_2 \end{bmatrix} $$

$$ {}^0v_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \text{ and since } {}^0\omega_1 = {}^0R_1\,{}^1\omega_1 = \begin{bmatrix} c_1 & -s_1 & 0 \\ s_1 & c_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} \text{ and } {}^0P_1 = \begin{bmatrix} a_1c_1 \\ a_1s_1 \\ 0 \end{bmatrix}, $$

$$ {}^1v_1 = \begin{bmatrix} c_1 & s_1 & 0 \\ -s_1 & c_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\left(\begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} \times \begin{bmatrix} a_1c_1 \\ a_1s_1 \\ 0 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ a_1\dot{\theta}_1 \\ 0 \end{bmatrix}. $$

$$ \text{Likewise, since } {}^1\omega_2 = {}^1R_2\,{}^2\omega_2 = \begin{bmatrix} c_2 & -s_2 & 0 \\ s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 + \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 + \dot{\theta}_2 \end{bmatrix} \text{ and } {}^1P_2 = \begin{bmatrix} a_2c_2 \\ a_2s_2 \\ 0 \end{bmatrix}, $$

$$ {}^2v_2 = \begin{bmatrix} c_2 & s_2 & 0 \\ -s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}\left(\begin{bmatrix} 0 \\ a_1\dot{\theta}_1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 + \dot{\theta}_2 \end{bmatrix} \times \begin{bmatrix} a_2c_2 \\ a_2s_2 \\ 0 \end{bmatrix}\right) = \begin{bmatrix} a_1\dot{\theta}_1 s_2 \\ a_1\dot{\theta}_1 c_2 + a_2(\dot{\theta}_1 + \dot{\theta}_2) \\ 0 \end{bmatrix}. \text{ (Ans.)} $$

To find these velocities with respect to the nonmoving base frame, we rotate them with the rotation matrix as follows:

$$ {}^0v_2 = {}^0R_2\,{}^2v_2 = \begin{bmatrix} c_{12} & -s_{12} & 0 \\ s_{12} & c_{12} & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} a_1\dot{\theta}_1 s_2 \\ a_1\dot{\theta}_1 c_2 + a_2(\dot{\theta}_1 + \dot{\theta}_2) \\ 0 \end{bmatrix} = \begin{bmatrix} -a_1\dot{\theta}_1 s_1 - a_2(\dot{\theta}_1 + \dot{\theta}_2)s_{12} \\ a_1\dot{\theta}_1 c_1 + a_2(\dot{\theta}_1 + \dot{\theta}_2)c_{12} \\ 0 \end{bmatrix} \text{ (Ans.)} $$

Method 2 (without using the iterative formulas):

$$ {}^0P_2 = \begin{bmatrix} a_1c_1 + a_2c_{12} \\ a_1s_1 + a_2s_{12} \\ 0 \end{bmatrix} \rightarrow {}^0v_2 = {}^0\dot{P}_2 = \begin{bmatrix} -a_1\dot{\theta}_1 s_1 - a_2(\dot{\theta}_1 + \dot{\theta}_2)s_{12} \\ a_1\dot{\theta}_1 c_1 + a_2(\dot{\theta}_1 + \dot{\theta}_2)c_{12} \\ 0 \end{bmatrix} \text{ (Ans.)} $$

$$ {}^2v_2 = {}^2R_0\,{}^0v_2 = \begin{bmatrix} c_{12} & s_{12} & 0 \\ -s_{12} & c_{12} & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} -a_1\dot{\theta}_1 s_1 - a_2(\dot{\theta}_1 + \dot{\theta}_2)s_{12} \\ a_1\dot{\theta}_1 c_1 + a_2(\dot{\theta}_1 + \dot{\theta}_2)c_{12} \\ 0 \end{bmatrix} = \begin{bmatrix} a_1\dot{\theta}_1 s_2 \\ a_1\dot{\theta}_1 c_2 + a_2(\dot{\theta}_1 + \dot{\theta}_2) \\ 0 \end{bmatrix} \text{ (Ans.)} $$

End-Effector Velocities

Angular velocity of end-effector: $\boxed{\omega_n = \sum_{i=1}^{n} \dot{\theta}_i \hat{Z}_{i-1}}$

Linear velocity of end-effector frame's origin: $\boxed{\upsilon_n = \sum_{i=1}^{n} \dot{\theta}_i \hat{Z}_{i-1} \times (P_n - P_{i-1}) + \dot{d}_i \hat{Z}_{i-1}}$

- Proofs

Angular velocity. Sum up $\omega_i - \omega_{i-1} = \dot{\theta}_i \hat{Z}_{i-1}$ where $\omega_0 = 0$:

$$\cancel{\omega_1} - \omega_0 = \dot{\theta}_1 \hat{Z}_0$$
$$\cancel{\omega_2} - \cancel{\omega_1} = \dot{\theta}_2 \hat{Z}_1$$
$$\vdots$$
$$\cancel{\omega_{n-1}} - \cancel{\omega_{n-2}} = \dot{\theta}_{n-1} \hat{Z}_{n-2}$$
$$+)\ \omega_n - \cancel{\omega_{n-1}} = \dot{\theta}_n \hat{Z}_{n-1}$$
$$\Downarrow$$
$$\therefore\ \omega_n = \sum_{i=1}^{n} \dot{\theta}_i \hat{Z}_{i-1}$$

Linear velocity. Sum up $\upsilon_i - \upsilon_{i-1} = \omega_i \times {}^{i-1}P_i + \dot{d}_i \hat{Z}_{i-1}$ where $v_0 = 0$:

$$\cancel{\upsilon_1} - \upsilon_0 = \omega_1 \times {}^{0}P_1 + \dot{d}_1 \hat{Z}_0$$
$$\cancel{\upsilon_2} - \cancel{\upsilon_1} = \omega_2 \times {}^{1}P_2 + \dot{d}_2 \hat{Z}_1$$
$$\vdots$$
$$\cancel{\upsilon_{n-1}} - \cancel{\upsilon_{i-2}} = \omega_{n-1} \times {}^{n-2}P_{n-1} + \dot{d}_{n-1} \hat{Z}_{n-2}$$
$$+)\ \upsilon_n - \cancel{\upsilon_{n-1}} = \omega_n \times {}^{n-1}P_n + \dot{d}_n \hat{Z}_{n-1}$$
$$\Downarrow$$
$$\upsilon_n = \sum_{i=1}^{n} (\omega_i \times {}^{i-1}P_i + \dot{d}_i \hat{Z}_{i-1})$$

From $\omega_i = \sum_{j=1}^{i} \dot{\theta}_j \hat{Z}_{j-1}$, and a double summation identity $\sum_{i=1}^{n}\sum_{j=1}^{i} a_{i,j} = \sum_{j=1}^{n}\sum_{i=j}^{n} a_{i,j}$, the first term is:

$$\sum_{i=1}^{n} \omega_i \times {}^{i-1}P_i = \sum_{i=1}^{n} [\sum_{j=1}^{i} (\dot{\theta}_j \hat{Z}_{j-1}) \times {}^{i-1}P_i] = \sum_{i=1}^{n}\sum_{j=1}^{i} [\dot{\theta}_j \hat{Z}_{j-1} \times {}^{i-1}P_i]$$
$$= \sum_{j=1}^{n}\sum_{i=j}^{n} [\dot{\theta}_j \hat{Z}_{j-1} \times {}^{i-1}P_i] = \sum_{j=1}^{n} [\dot{\theta}_j \hat{Z}_{j-1} \times \underbrace{\sum_{i=j}^{n} {}^{i-1}P_i}_{= {}^{j-1}P_n = P_n - P_{j-1}}] = \sum_{j=1}^{n} \dot{\theta}_j \hat{Z}_{j-1} \times (P_n - P_{j-1})$$

$$\therefore\ \upsilon_n = \sum_{i=1}^{n} \dot{\theta}_i \hat{Z}_{i-1} \times (P_n - P_{i-1}) + \dot{d}_i \hat{Z}_{i-1}$$

Jacobian (in general; analytical method)

Derivative in multidimensional (vector) space (vs. derivative with respect to scalar variable(s))
Mapping (linear) in tangential (velocity) space

- Given m functions with n independent variables

$$y_1 = f_1(x_1, \ldots, x_n),$$
$$y_2 = f_2(x_1, \ldots, x_n),$$
$$\vdots \qquad\qquad \text{or, } \mathbf{Y} = \mathbf{F}(\mathbf{X})$$
$$y_m = f_m(x_1, \ldots, x_n).$$

- Differentials of y_i with respect to x_j (linear combinations)

$$\delta y_1 = \frac{\partial f_1}{\partial x_1}\delta x_1 + \frac{\partial f_1}{\partial x_2}\delta x_2 + \ldots + \frac{\partial f_1}{\partial x_n}\delta x_n ,$$
$$\delta y_2 = \frac{\partial f_2}{\partial x_1}\delta x_1 + \frac{\partial f_2}{\partial x_2}\delta x_2 + \ldots + \frac{\partial f_2}{\partial x_n}\delta x_n ,$$
$$\vdots \qquad\qquad \text{or, } \delta\mathbf{Y} = \frac{\partial \mathbf{F}}{\partial \mathbf{X}}\delta\mathbf{X} = J(\mathbf{X})\delta\mathbf{X}$$
$$\delta y_m = \frac{\partial f_m}{\partial x_1}\delta x_1 + \frac{\partial f_m}{\partial x_2}\delta x_2 + \ldots + \frac{\partial f_m}{\partial x_n}\delta x_n .$$

- $J(\mathbf{X}) = \dfrac{\partial \mathbf{F}}{\partial \mathbf{X}}$: m x n **Jacobian matrix**; time-varying linear transformation

$$\boxed{J(\mathbf{X}) = \frac{\partial \mathbf{F}_{(m\times 1)}}{\partial \mathbf{X}_{(n\times 1)}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}_{(m\times n)}}$$
(derivative of a vector with respect to another vector)

- Det(J): **Jacobian**
- In kinematics, $\delta\mathbf{Y} = J(\mathbf{X})\delta\mathbf{X}$: infinitesimal (or differential) motion
- In kinematics, $\boxed{\dot{\mathbf{Y}} = J(\mathbf{X})\dot{\mathbf{X}}}$: mapping velocities in **X** to **Y**
- Note: Jacobian for angular velocity cannot be derived directly from analytical method.

Jacobian (in robotics; geometric method)

Directly mapping joint velocities to Cartesian (angular, as well as linear) velocities of end-effector

- Let $\mathbf{q}$: n-DOF joint variables vector; ${}^0\mathbf{V} = \begin{bmatrix} {}^0\mathbf{v}_{(3\times 1)} \\ {}^0\boldsymbol{\omega}_{(3\times 1)} \end{bmatrix}_{(6\times 1)}$: Cartesian linear and angular velocity vector

$\boxed{{}^0\mathbf{V}_{(6\times 1)} = {}^0J(\mathbf{q})_{(6\times n)}\,\dot{\mathbf{q}}_{(n\times 1)}}$ (differential kinematics)

- Changing Jacobian's frame of reference from $\{B\}$ to $\{A\}$

Given $\begin{bmatrix} {}^B\mathbf{v} \\ {}^B\boldsymbol{\omega} \end{bmatrix} = {}^B\mathbf{V} = {}^BJ(\mathbf{q})\dot{\mathbf{q}}$; use $\begin{bmatrix} {}^A\mathbf{v} \\ {}^A\boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} {}^AR_B & 0 \\ 0 & {}^AR_B \end{bmatrix} \begin{bmatrix} {}^B\mathbf{v} \\ {}^B\boldsymbol{\omega} \end{bmatrix}$

$\therefore \; {}^AJ(\mathbf{q}) = \left[\begin{array}{c:c} {}^AR_B & 0 \\ \hdashline 0 & {}^AR_B \end{array}\right] {}^BJ(\mathbf{q})$

- Example: 2-link arm linear Jacobian ${}^0J(\mathbf{q}) = \begin{bmatrix} -l_1s_1 - l_2s_{12} & -l_2s_{12} \\ l_1c_1 + l_2c_{12} & l_2c_{12} \end{bmatrix}$, ${}^2J(\mathbf{q}) = \begin{bmatrix} l_1s_2 & 0 \\ l_1c_2 + l_2 & l_2 \end{bmatrix}$

<u>Jacobian Matrix Computation using Geometric Method</u>

- Partition into 3x1 column vectors: $J_{P,i}(\mathbf{q})$ for position and $J_{O,i}(\mathbf{q})$ for orientation

$$J(\mathbf{q})_{(6\times n)} = [J_1(\mathbf{q})_{(6\times1)} \mid \ldots \mid J_i(\mathbf{q})_{(6\times1)} \mid \ldots \mid J_n(\mathbf{q})_{(6\times1)}] = \left[\begin{array}{c|c|c|c|c} J_{P,1}(\mathbf{q})_{(3\times1)} & \ldots & J_{P,i}(\mathbf{q})_{(3\times1)} & \ldots & J_{P,n}(\mathbf{q})_{(3\times1)} \\ J_{O,1}(\mathbf{q})_{(3\times1)} & & J_{O,i}(\mathbf{q})_{(3\times1)} & & J_{O,n}(\mathbf{q})_{(3\times1)} \end{array}\right]$$

$$\mathbf{V}_{(6\times1)} = J(\mathbf{q})_{(6\times n)}\dot{\mathbf{q}}_{(n\times1)} \rightarrow \upsilon_n = \sum_{i=1}^{n} \dot{q}_i J_{P,i}(\mathbf{q}) \;\&\; \omega_n = \sum_{i=1}^{n} \dot{q}_i J_{O,i}(\mathbf{q})$$

$\dot{q}_i J_{P,i}(\mathbf{q})$: contribution of single Joint i velocity to the end-effector frame origin's linear velocity

$\dot{q}_i J_{O,i}(\mathbf{q})$: contribution of single Joint i velocity to the end-effector frame's angular velocity

$$\boxed{J_i(\mathbf{q})_{(6\times1)} = \begin{bmatrix} J_{P,i}(\mathbf{q})_{(3\times1)} \\ \hdashline J_{O,i}(\mathbf{q})_{(3\times1)} \end{bmatrix} = \begin{cases} \begin{bmatrix} \hat{Z}_{i-1} \\ \hdashline \mathbf{0} \end{bmatrix} & \leftarrow \text{Prismatic joint } i \\ \begin{bmatrix} \hat{Z}_{i-1} \times (P_n - P_{i-1}) \\ \hdashline \hat{Z}_{i-1} \end{bmatrix} & \leftarrow \text{Revolute joint } i \end{cases}}$$

The vectors $\hat{Z}_{i-1}$, P_n, and P_{i-1} are all functions of the joint variables. If written in Frame {0}:

$\hat{Z}_{i-1}$: obtained from the third column of ${}^0R_{i-1}(q_1,\ldots,q_{i-1})$ or ${}^0T_{i-1}(q_1,\ldots,q_{i-1})$.

$\rightarrow \hat{Z}_{i-1} = {}^0R_{i-1}[0 \quad 0 \quad 1]^T$ OR $\begin{bmatrix} \hat{Z}_{i-1} \\ \hdashline 0 \end{bmatrix} = {}^0T_{i-1}[0 \quad 0 \quad 1 \quad 0]^T$

P_n: obtained from the fourth column of ${}^0T_n(q_1,\ldots,q_n)$. $\rightarrow \begin{bmatrix} P_n \\ \hdashline 1 \end{bmatrix} = {}^0T_n[0 \quad 0 \quad 0 \quad 1]^T$

P_{i-1}: obtained from the fourth column of ${}^0T_{i-1}(q_1,\ldots,q_{i-1})$. $\rightarrow \begin{bmatrix} P_{i-1} \\ \hdashline 1 \end{bmatrix} = {}^0T_{i-1}[0 \quad 0 \quad 0 \quad 1]^T$

- Proofs

(a) $J_{O,i}(\mathbf{q})$: Since $\omega_n = \sum_{i=1}^{n} \dot{\theta}_i \hat{Z}_{i-1}$ and $\omega_n = \sum_{i=1}^{n} \dot{q}_i J_{O,i}(\mathbf{q})$, $\sum_{i=1}^{n} \dot{q}_i J_{O,i}(\mathbf{q}) = \sum_{i=1}^{n} \dot{\theta}_i \hat{Z}_{i-1}$.

$\therefore \; J_{O,i}(\mathbf{q}) = \hat{Z}_{i-1}$ for revolute joint and $J_{O,i}(\mathbf{q}) = \mathbf{0}$ for prismatic joint.

(b) $J_{P,i}(\mathbf{q})$: Since $\upsilon_n = \sum_{i=1}^{n} \dot{\theta}_i \hat{Z}_{i-1} \times (P_n - P_{i-1}) + \sum_{i=1}^{n} \dot{d}_i \hat{Z}_{i-1}$ and $\upsilon_n = \sum_{i=1}^{n} \dot{q}_i J_{P,i}(\mathbf{q})$,

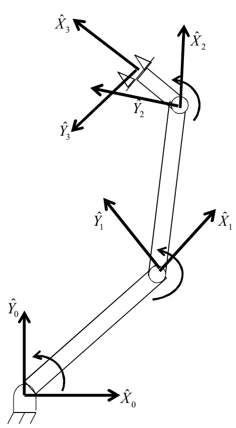
$$\sum_{i=1}^{n} \dot{q}_i J_{P,i}(\mathbf{q}) = \sum_{i=1}^{n} \dot{\theta}_i \hat{Z}_{i-1} \times (P_n - P_{i-1}) + \sum_{i=1}^{n} \dot{d}_i \hat{Z}_{i-1} .$$

$\therefore$ $J_{P,i}(\mathbf{q}) = \hat{Z}_{i-1} \times (P_n - P_{i-1})$ for revolute joint and $J_{P,i}(\mathbf{q}) = \hat{Z}_{i-1}$ for prismatic joint.

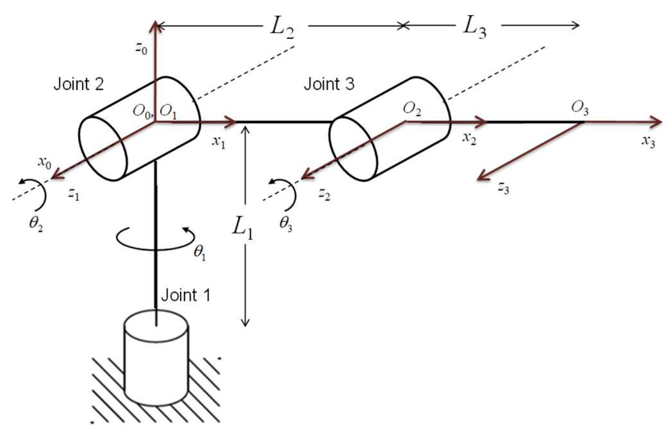
- Kinematic interpretations
  - Assume that all joints, other than Joint *i*, are instantaneously fixed, and thus all the links from Link *i* to the end-effector can be regarded as a single rigid body.

The contribution of prismatic (allows pure translation) joint velocity to the end-effector frame's
angular velocity: None. $\because$ A rigid body in translation has zero angular velocity.
linear velocity (of origin): Vector addition of the prismatic joint velocity. $\because$ All points in the rigid body have same linear velocities (translation).

The contribution of revolute (allows pure rotation) joint velocity to the end-effector frame's
angular velocity: Vector addition of the revolute joint velocity. $\because$ An angular velocity vector (free vector) due to the revolute joint's rotation can be transported to the end-effector frame.
linear velocity (of origin): Rotation of the position vector of the end-effector frame's origin relative to the origin of Joint *i* axis frame. Note that, unlike the other three cases, this is the only quantity that depends on the end-effector's (relative) position.



Three-link planar arm



Anthropomorphic arm

Example: Three-link Planar Arm

- The position vectors and the joint axes' unit vectors, all written in Frame {0}, are:

$$P_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},\ P_1 = \begin{bmatrix} a_1c_1 \\ a_1s_1 \\ 0 \end{bmatrix},\ P_2 = \begin{bmatrix} a_1c_1 + a_2c_{12} \\ a_1s_1 + a_2s_{12} \\ 0 \end{bmatrix},\ P_3 = \begin{bmatrix} a_1c_1 + a_2c_{12} + a_3c_{123} \\ a_1s_1 + a_2s_{12} + a_3s_{123} \\ 0 \end{bmatrix}, \text{and } \hat{Z}_0 = \hat{Z}_1 = \hat{Z}_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

where $c_{12} = \cos(\theta_1 + \theta_2)$, $c_{123} = \cos(\theta_1 + \theta_2 + \theta_3)$, $s_{12} = \sin(\theta_1 + \theta_2)$, $s_{123} = \sin(\theta_1 + \theta_2 + \theta_3)$, etc.

$$\therefore\; J=\begin{bmatrix}\hat{Z}_0\times(P_3-P_0) & \hat{Z}_1\times(P_3-P_1) & \hat{Z}_2\times(P_3-P_2)\\ \hat{Z}_0 & \hat{Z}_1 & \hat{Z}_2\end{bmatrix}=\begin{bmatrix}-a_1s_1-a_2s_{12}-a_3s_{123} & -a_2s_{12}-a_3s_{123} & -a_3s_{123}\\ a_1c_1+a_2c_{12}+a_3c_{123} & a_2c_{12}+a_3c_{123} & a_3c_{123}\\ 0&0&0\\0&0&0\\0&0&0\\1&1&1\end{bmatrix}.$$

<u>Example: Anthropomorphic Arm</u> (shown in its home configuration)

i	θ_i	d_i	a_i	α_i	Variable
1	$\theta_1=90^o+q_1$	0	0	90°	q_1
2	$\theta_2=0+q_2$	0	L_2	0	q_2
3	$\theta_3=0+q_3$	0	L_3	0	q_3

$${}^0T_1=\begin{bmatrix}\cos\theta_1 & 0 & \sin\theta_1 & 0\\ \sin\theta_1 & 0 & -\cos\theta_1 & 0\\ 0&1&0&0\\0&0&0&1\end{bmatrix},\; {}^1T_2=\begin{bmatrix}\cos\theta_2 & -\sin\theta_2 & 0 & L_2\cos\theta_2\\ \sin\theta_2 & \cos\theta_2 & 0 & L_2\sin\theta_2\\ 0&0&1&0\\0&0&0&1\end{bmatrix},\text{ and}$$

$${}^2T_3=\begin{bmatrix}\cos\theta_3 & -\sin\theta_3 & 0 & L_3\cos\theta_3\\ \sin\theta_3 & \cos\theta_3 & 0 & L_3\sin\theta_3\\ 0&0&1&0\\0&0&0&1\end{bmatrix}$$

$$\therefore\; {}^0T_2={}^0T_1\,{}^1T_2=\begin{bmatrix}c_1c_2 & -c_1s_2 & s_1 & L_2c_1c_2\\ s_1c_2 & -s_1s_2 & -c_1 & L_2s_1c_2\\ s_2 & c_2 & 0 & L_2s_2\\ 0&0&0&1\end{bmatrix},\; {}^0T_3={}^0T_2\,{}^2T_3=\begin{bmatrix}c_1c_{23} & -c_1s_{23} & s_1 & c_1(L_2c_2+L_3c_{23})\\ s_1c_{23} & -s_1s_{23} & -c_1 & s_1(L_2c_2+L_3c_{23})\\ s_{23} & c_{23} & 0 & L_2s_2+L_3s_{23}\\ 0&0&0&1\end{bmatrix}.$$

- The position vectors and the joint axes' unit vectors, all written in Frame $\{0\}$, are:

$$P_0=P_1=\begin{bmatrix}0\\0\\0\end{bmatrix},\; P_2=\begin{bmatrix}L_2c_1c_2\\ L_2s_1c_2\\ L_2s_2\end{bmatrix},\text{ and } P_3=\begin{bmatrix}c_1(L_2c_2+L_3c_{23})\\ s_1(L_2c_2+L_3c_{23})\\ L_2s_2+L_3s_{23}\end{bmatrix};\; \hat{Z}_0=\begin{bmatrix}0\\0\\1\end{bmatrix}\text{ and } \hat{Z}_1=\hat{Z}_2=\begin{bmatrix}s_1\\-c_1\\0\end{bmatrix}.$$

$$\therefore\; J=\begin{bmatrix}\hat{Z}_0\times(P_3-P_0) & \hat{Z}_1\times(P_3-P_1) & \hat{Z}_2\times(P_3-P_2)\\ \hat{Z}_0 & \hat{Z}_1 & \hat{Z}_2\end{bmatrix}=\begin{bmatrix}-s_1(L_2c_2+L_3c_{23}) & -c_1(L_2s_2+L_3s_{23}) & -L_3c_1s_{23}\\ c_1(L_2c_2+L_3c_{23}) & -s_1(L_2s_2+L_3s_{23}) & -L_3s_1s_{23}\\ 0 & L_2c_2+L_3c_{23} & L_3c_{23}\\ 0 & s_1 & s_1\\ 0 & -c_1 & -c_1\\ 1 & 0 & 0\end{bmatrix}.$$

(Note: If the global origin is on the ground, the solution will be different and will include L_1.)

Singularities

Determinant = 0
In robotics: $Det(J) = 0$ → J loses full rank

- If J is nonsingular, i.e., $Det(J) \neq 0$ → $\dot{\mathbf{q}} = J^{-1}(\mathbf{q})\mathbf{V}$ (differential kinematics → inverse kinematics)
- If J is singular → manipulator loses one or more DOFs in Cartesian space (from implicit function theorem and/or differential geometry theory); it cannot move along some direction(s).
  - Joint rates approach infinity (why?)

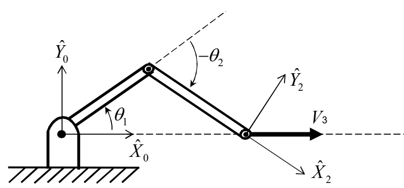
- Workspace singularities
  Workspace boundary singularities: links are fully stretched out or folded back
  Workspace interior singularities: when two or more joint axes are aligned
  → Use to construct workspaces

- Example: 2-link arm

$$Det[{}^0J(\mathbf{q})] = \begin{vmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{vmatrix} = l_1 l_2 s_2 = 0$$

  → singular when θ_2 = 0, 180° (stretched out or folded back) → workspace boundary singularities

- Example 5.5 (Craig's 4th Ed.): Consider a two-link robot moving its end-effector along the $\hat{X}$ axis at 1.0 m/s. Show that as a singularity is approached at $\theta_2 = 0$, joint rates tend to infinity.



Sol) The inverse of the Jacobian written in Frame {0} is ${}^0J^{-1}(\mathbf{q}) = \dfrac{1}{l_1 l_2 s_2}\begin{bmatrix} l_2 c_{12} & l_2 s_{12} \\ -l_1 c_1 - l_2 c_{12} & -l_1 s_1 - l_2 s_{12} \end{bmatrix}$.

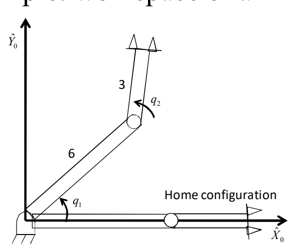
Then using $\dot{\mathbf{q}} = J^{-1}(\mathbf{q})\mathbf{V}$ with $\mathbf{V} = [1, 0]^T$, the joint rates as a function of manipulator configuration is:

$\dot{\theta}_1 = \dfrac{c_{12}}{l_1 s_2}$ and $\dot{\theta}_2 = -\dfrac{c_1}{l_2 s_2} - \dfrac{c_{12}}{l_1 s_2}$ $\therefore$ As $\theta_2 \to 0$ (arm stretches out), $\dot{\theta}_1 \to \infty$ and $\dot{\theta}_2 \to \infty$.

Robot Workspace

The (continuum) set of points in space that can be reached by a point on end-effector.

- Example: workspace of a 2-link arm



Joint	θ	d	a	α
1	$0 + q_1$	0	6	0
2	$0 + q_2$	0	3	0

${}^0T_n$ → $x = 6\cos q_1 + 3\cos(q_1+q_2)$, $y = 6\sin q_1 + 3\sin(q_1+q_2)$

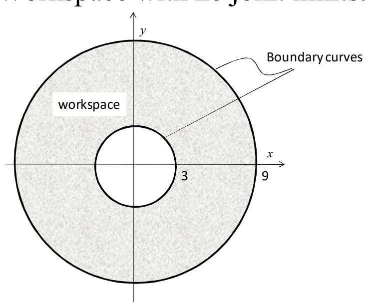
Jacobian $J = \begin{bmatrix} \dfrac{\partial x}{\partial q_1} & \dfrac{\partial x}{\partial q_2} \\ \dfrac{\partial y}{\partial q_1} & \dfrac{\partial y}{\partial q_2} \end{bmatrix}$; $Det(J) = 18\sin q_2 = 0$ → $q_2 = 0, \pi$

Find set(s) of $\mathbf{q}$ that make J singular:

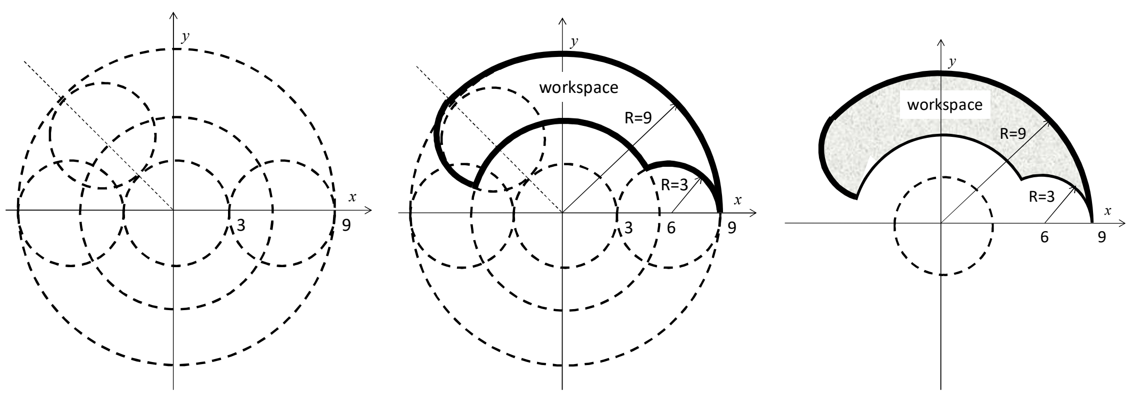
$P_{singular}(q_1, q_2 = 0) = \begin{bmatrix} 6\cos q_1 + 3\cos(q_1+0) \\ 6\sin q_1 + 3\sin(q_1+0) \end{bmatrix} = \begin{bmatrix} 9\cos q_1 \\ 9\sin q_1 \end{bmatrix}$ → circle of radius 9 and center at the origin

$P_{singular}(q_1, q_2 = \pi) = \begin{bmatrix} 6\cos q_1 + 3\cos(q_1+\pi) \\ 6\sin q_1 + 3\sin(q_1+\pi) \end{bmatrix} = \begin{bmatrix} 3\cos q_1 \\ 3\sin q_1 \end{bmatrix}$ → circle of radius 3 and center at the origin

(a) Workspace with no joint limits:



(b) Workspace with joint limits $0 < q_1 < 135$ and $0 < q_2 < 120$:

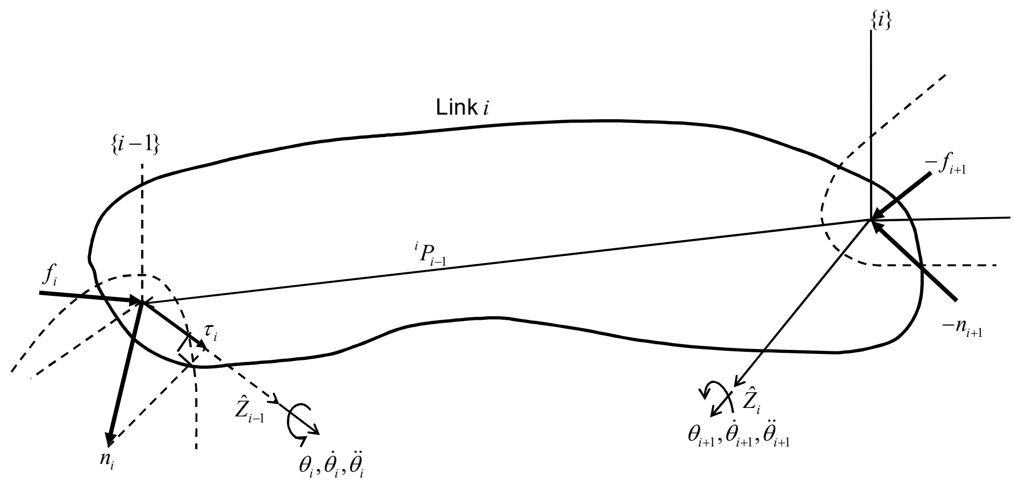


Static Forces

f_i = force exerted......Link i......Link i-1

n_i = moment exerted......Link i......Link i-1

- Note: In general, a FBD should include all forces/moments exerted "on" the system of interest "by" the environment.



- Static equilibrium
  - Force: $\sum f = 0$ => ${}^if_i - {}^if_{i+1} = 0$ => ${}^if_i = {}^if_{i+1}$
  - Moment **about** origin of Frame $\{i\}$: $\sum n = 0$ => ${}^in_i - {}^in_{i+1} + {}^iP_{i-1} \times {}^if_i = 0$ => ${}^in_i = {}^in_{i+1} - {}^iP_{i-1} \times {}^if_i$

- Start with a description of the forces and moments applied at the end-effector (Link n)
  → calculate from Link n to Link 0 (inward)

- Static force/moment propagation from link to link expressed in each link frame:
  $\boxed{{}^if_i = {}^iR_{i+1}\,{}^{i+1}f_{i+1},\ {}^in_i = {}^iR_{i+1}\,{}^{i+1}n_{i+1} - {}^iP_{i-1} \times {}^if_i}$

- All components of the force and moment vectors are resisted by the reaction from the structure of the mechanism itself, except for the force/moment component (actuation) along the joint axis.
- Actuation required to maintain static equilibrium
  - Joint actuator torque (revolute joint i): $\tau_i = {}^in_i^T\,{}^i\hat{Z}_{i-1}$
  - Joint actuator force (prismatic joint i): $\tau_i = {}^if_i^T\,{}^i\hat{Z}_{i-1}$

Jacobians in the Force Domain

- Let
  - $\mathbf{F}$: 6x1 Cartesian force-moment vector applied on the end-effector
  - $\delta\mathbf{X}$: 6x1 infinitesimal Cartesian displacement of the end-effector
  - $\boldsymbol{\tau}$: nx1 joint actuator torque vector
  - $\delta\mathbf{q}$: nx1 infinitesimal joint variables vector

- **Principle of virtual work** (static equilibrium)
  $\mathbf{F}\bullet\delta\mathbf{X} - \boldsymbol{\tau}\bullet\delta\mathbf{q} = 0$ => $\mathbf{F}^T\delta\mathbf{X} = \boldsymbol{\tau}^T\delta\mathbf{q}$
  → [work done in Cartesian terms] = [work done in joint space terms]
  : Work is the same measured in any set of generalized coordinates

- Recall: Jacobian $\delta\mathbf{X} = J\delta\mathbf{q}$ → $\mathbf{F}^T J\delta\mathbf{q} = \boldsymbol{\tau}^T\delta\mathbf{q}$ ($\forall\ \delta\mathbf{q}$)
  $\therefore \boxed{\boldsymbol{\tau} = J^T\mathbf{F}}$
  : Jacobian transpose maps Cartesian forces/moments into equivalent joint torques

(Note: In the above equation, $\boldsymbol{\tau}$ are the joint torques producing effects that are "equivalent" to those of **F**; on the other hand, the joint torques that are in static "equilibrium" with **F** is $\boldsymbol{\tau} = -J^T\mathbf{F}$.)

- Kineto-statics duality: $\delta\mathbf{X} = J\delta\mathbf{q}$ vs. $\boldsymbol{\tau} = J^T\mathbf{F}$

- If J is singular (i.e., loses full rank) or near singular: **F** can be increased or decreased in null-space basis directions without changes in $\boldsymbol{\tau}$.
  → mechanical advantage goes infinity; small $\boldsymbol{\tau}$ required to generate large forces at the end-effector

- Singular configuration → singularity in both position and force domains

<u>Cartesian Transformation of Velocities and Static Forces</u>

- 6x1 general velocity of a body: $\mathbf{V} = \begin{bmatrix} \mathbf{v}_{(3\times1)} \\ \boldsymbol{\omega}_{(3\times1)} \end{bmatrix}$
- 6x1 general force vector: $\mathbf{F} = \begin{bmatrix} f_{(3\times1)} \\ n_{(3\times1)} \end{bmatrix}$ (f: 3x1 force vector; n: 3x1 moment vector)
- 6x6 transformations to map from Frame $\{A\}$ to $\{B\}$ at each time instant

- Velocity transformation
  Recall: ${}^{i+1}\omega_{i+1} = {}^{i+1}R_i({}^i\omega_i + \dot{\theta}_{i+1}\,{}^i\hat{Z}_i)$ and ${}^{i+1}\upsilon_{i+1} = {}^{i+1}R_i({}^i\upsilon_i + {}^i\omega_{i+1} \times {}^iP_{i+1})$
  with $\dot{\theta}_{i+1} = 0$ ($\because$ the two frames are rigidly connected) and $\{i\} = \{A\}$, $\{i+1\} = \{B\}$
  → matrix form: $\begin{bmatrix} {}^B\mathbf{v}_B \\ {}^B\boldsymbol{\omega}_B \end{bmatrix} = \left[\begin{array}{c|c} {}^BR_A & -{}^BR_A\,{}^AP_{BORG}\times \\ \hline 0 & {}^BR_A \end{array}\right] \begin{bmatrix} {}^A\mathbf{v}_A \\ {}^A\boldsymbol{\omega}_A \end{bmatrix}$ or ${}^B\mathbf{V}_B = {}^BT_{vA}{}^A\mathbf{V}_A$ (6x6 operator)

  where $P\times = \begin{bmatrix} 0 & -p_z & p_y \\ p_z & 0 & -p_x \\ -p_y & p_x & 0 \end{bmatrix}$ (Note: recall similar formula for angular velocity matrix!)

- Inversion: $\begin{bmatrix} {}^A\mathbf{v}_A \\ {}^A\boldsymbol{\omega}_A \end{bmatrix} = \left[\begin{array}{c|c} {}^AR_B & ({}^AP_{BORG}\times)\cdot{}^AR_B \\ \hline 0 & {}^AR_B \end{array}\right] \begin{bmatrix} {}^B\mathbf{v}_B \\ {}^B\boldsymbol{\omega}_B \end{bmatrix}$ or ${}^A\mathbf{V}_A = {}^AT_{vB}{}^B\mathbf{V}_B$

- Force-moment transformation
  Recall: ${}^if_i = {}^iR_{i+1}{}^{i+1}f_{i+1}$ and ${}^in_i = {}^iR_{i+1}{}^{i+1}n_{i+1} - {}^iP_{i-1} \times {}^if_i$
  → matrix form: $\begin{bmatrix} {}^Af_A \\ {}^An_A \end{bmatrix} = \left[\begin{array}{c|c} {}^AR_B & 0 \\ \hline -({}^AP_{\text{joint}A}\times)\cdot{}^AR_B & {}^AR_B \end{array}\right] \begin{bmatrix} {}^Bf_B \\ {}^Bn_B \end{bmatrix}$ or ${}^A\mathbf{F}_A = {}^A\mathbf{T}_{fB}{}^B\mathbf{F}_B$
- ${}^AT_{fB} = {}^AT_{vB}^T$

- Example 5.8 (Craig's 4th Ed.): (Do it yourself)

<u>Redundancy Resolution</u>

- Given m function equations with n-DOF joint variables → J: mxn Jacobian matrix
- If $m < n$ (i.e., redundant), infinite solutions of $\dot{\mathbf{q}}$ exist for $\mathbf{V}_{(m\times1)} = J(\mathbf{q})_{(m\times n)}\dot{\mathbf{q}}_{(n\times1)}$.

- Solution methods: Formulate as a constrained optimization problem.
  - Jacobian pseudo-inverse
  - Numerical trajectory optimization (e.g., collocation method, single/multiple shooting methods, etc.)

Jacobian Pseudo-Inverse

- Let the end-effector velocity is $\mathbf{V}$, Jacobian J (for given $\mathbf{q}$) has full rank, and W is a suitable (nxn) symmetric positive definite weight matrix. Then the optimal solution $\dot{\mathbf{q}}^*$ that satisfies $\mathbf{V} = J\dot{\mathbf{q}}$ and minimizes the quadratic cost functional $g(\dot{\mathbf{q}}) = \frac{1}{2}\dot{\mathbf{q}}^T W \dot{\mathbf{q}}$ is $\dot{\mathbf{q}}^* = J^+\mathbf{V}$, where $J^+ = W^{-1}J^T(JW^{-1}J^T)^{-1}$ is the weighted right pseudo-inverse of J, i.e., $JJ^+ = I_n$.

- Proof (Use Method of Lagrange multipliers)
  Minimize $g(\dot{\mathbf{q}}, \boldsymbol{\lambda}) = \frac{1}{2}\dot{\mathbf{q}}^T W \dot{\mathbf{q}} + \boldsymbol{\lambda}^T(\mathbf{V} - J\dot{\mathbf{q}})$, where $\boldsymbol{\lambda}$ is a (mx1) vector of unknown Lagrange multipliers. Since $\partial^2 g / \partial \dot{\mathbf{q}}^2 = W$ is positive definite, the necessary conditions for minimum are:
  $\dfrac{\partial g}{\partial \dot{\mathbf{q}}} = \mathbf{0}^T$ → $\dot{\mathbf{q}} = W^{-1}J^T\boldsymbol{\lambda}$ (where W^{-1} exists); and $\dfrac{\partial g}{\partial \boldsymbol{\lambda}} = \mathbf{0}^T$ → $\mathbf{V} = J\dot{\mathbf{q}}$
  => $\mathbf{V} = JW^{-1}J^T\boldsymbol{\lambda}$ → $\boldsymbol{\lambda} = (JW^{-1}J^T)^{-1}\mathbf{V}$ ($\because$ $JW^{-1}J^T$: (mxm) square matrix of rank m and invertible)
  → $\dot{\mathbf{q}}^* = W^{-1}J^T(JW^{-1}J^T)^{-1}\mathbf{V}$

- If $W = I_n$ → $\boxed{J^+ = J^T(JJ^T)^{-1}}$: right pseudo-inverse of J → minimizes $\|\dot{\mathbf{q}}\|$

- If the cost functional is $g'(\dot{\mathbf{q}}) = \frac{1}{2}(\dot{\mathbf{q}} - \dot{\mathbf{q}}_0)^T(\dot{\mathbf{q}} - \dot{\mathbf{q}}_0)$, where $\dot{\mathbf{q}}_0$ is a vector of arbitrary joint velocities
  → $\dot{\mathbf{q}}^* = J^+\mathbf{V} + (I_n - J^+J)\dot{\mathbf{q}}_0$ (from the Method of Lagrange multipliers)
  - $J^+\mathbf{V}$: minimizes $\|\dot{\mathbf{q}}\|$
  - $(I_n - J^+J)\dot{\mathbf{q}}_0$: homogeneous solution; attempts to satisfy additional constraints to specify via $\dot{\mathbf{q}}_0$.

  Remark: $J(I_n - J^+J)\dot{\mathbf{q}}_0 = \mathbf{0}$, i.e., $I_n - J^+J$ projects $\dot{\mathbf{q}}_0$ in the null space of J, and $\dot{\mathbf{q}}_0$ generates internal motions of $(I_n - J^+J)\dot{\mathbf{q}}_0$ without violating the end-effector's $\mathbf{V} = J\dot{\mathbf{q}}$.

- Remark: If $m > n$ (i.e., over-constrained), no solution of $\dot{\mathbf{q}}$ exists for $\mathbf{V}_{(m\times 1)} = J(\mathbf{q})_{(m\times n)}\dot{\mathbf{q}}_{(n\times 1)}$.
  → (weighted) left pseudo-inverse of J ($J^+J = I_n$) → approximate solution to minimize $\|\mathbf{V} - J\dot{\mathbf{q}}\|$

CHAPTER 1. INTRODUCTION

Background

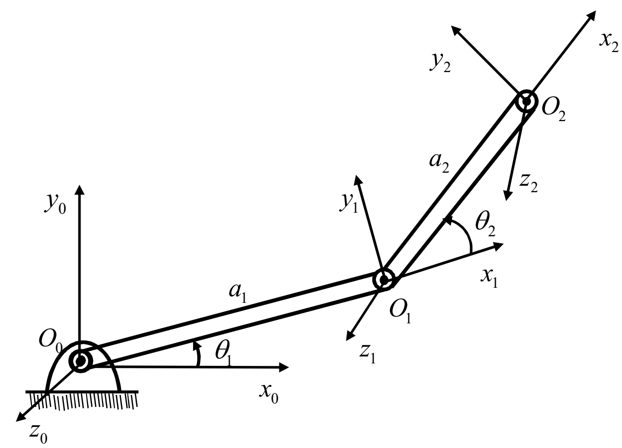
- Definition from Robot Institute of America (RIA)
  "A *Robot* is a reprogrammable, multifunctional manipulator designed to move material, parts, tools, or specialized devices through variable programmed motions for the performance of a variety of tasks."

    - Advantage: reduces human labor, increases accuracy, productivity, and flexibility

- The term "robota (= labor)" is originated from Czech play "Rossum's Universal Robots."

- History of robots
  1954: one degree-of-freedom robot patented in U.S. by G.C. Devol.
  1961: first practical industrial robot developed by Unimation.
  1968: first Japanese industrial robot from Kawasaki.
  1970s: specialized industrial robots.
  1980s and 1990s: ↓
  21C and future: various applications – service, medical, military, etc.

- Robotics as a multidisciplinary field
  Statics
  Kinematics and dynamics
  Machine/mechanism design
  Control
  Sensing
  Vision
  Artificial intelligence
  Mechatronics
  Computer algorithm and programming



Terminologies and Overview

- Position and orientation
    - Step 1: Attach a coordinate system ("local frame") rigidly to each single rigid body.
    - Step 2: Describe the position and orientation of each local frame with respect to a reference coordinate system ("global frame" or "base frame").

- Mechanical manipulator (or manipulator): rigid links connected with joints → allows relative motion of neighboring links
  → Joint displacement or joint variable – relative (position sensor)
  Revolute joint (model) – joint angle
  Prismatic joint (model) – translation or joint offset
  (vs. physical joint)

- Degrees of freedom (DOFs): number of independent coordinates required to describe the configuration of a system
    - Particle unconstrained in 3D → 3 DOF (3 translations)
    - Rigid body unconstrained in 3D → 6 DOF (3 translations + 3 rotations)

- End-effector: free end of the chain of links which make up the manipulator

- Forward kinematics: given a set of joint variables, compute the position and orientation of the end-effector's local frame (i.e., tool frame) relative to the global frame
  → Description mapping: joint space → Cartesian space (= operational space, task space)

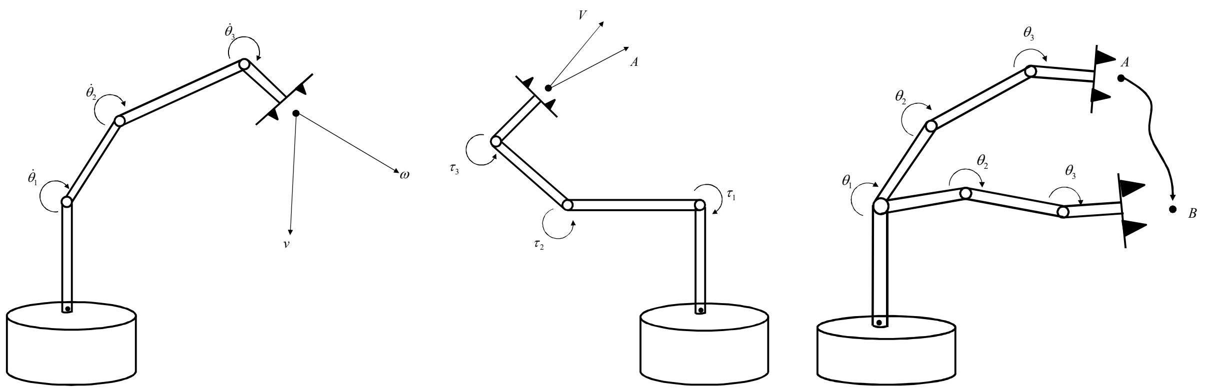
- Inverse kinematics: given the position and orientation of the end-effector, calculate the joint variables

- Workspace → existence/nonexistence of a kinematic solution

- Jacobian matrix → mapping from joint space velocities to Cartesian space velocities
  - Singularity point → mapping is not invertible

- Joint actuator; actuator torques → manipulator statics (equilibrium) and dynamics (equations of motion)

- Trajectory generation
  Trajectory: spatial and temporal (function of time)
  Path: spatial, but not temporal



- Position control system: automatically compensate for errors in knowledge of the parameters of a system, and suppress disturbances which tend to perturb the system from the desired trajectory
  - Position/velocity sensors → control algorithm → actuator torque computation

- Nonlinear position control: nonlinear dynamics of the manipulator

- Force control: addresses the interaction (e.g., contact force) with the environment (e.g., parts, tools, surfaces, etc.)
  - Complementary to position control → hybrid position/force control

<u>Steps of Solving Mechanics Problems</u>

Step 1: Identify and isolate system of interest.
Step 2: Draw free-body diagram (FBD) of the system of interest, its interactions (i.e., external forces and moments) with the environment, and coordinate frame(s).
Step 3: Formulate governing equations.

- Note: In general, a FBD should include all forces/moments exerted "on" the <u>system of interest</u> "by" the <u>environment</u>.

CHAPTER 2. SPATIAL DESCRIPTIONS AND TRANSFORMATIONS

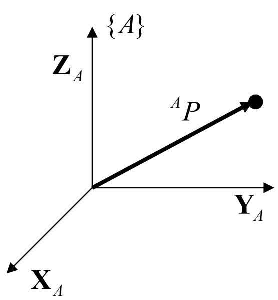
- Global (= universe = inertial = Newtonian = world) coordinate system

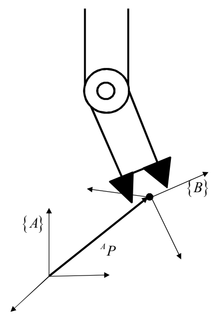
Position (of a point)

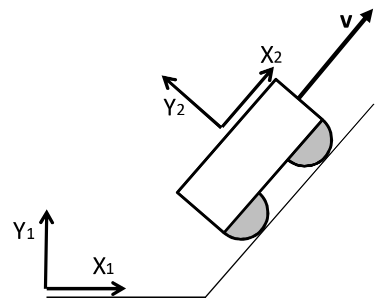
- 3x1 position vector (e.g., ${}^{A}\mathbf{P}$) → identify the coordinate system $\{A\}$ of description

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

  - Components of ${}^{A}\mathbf{P}$: distances along axes of $\{A\}$







Orientation (of a rigid body)

Attach a coordinate system to a body → describe this frame relative to the reference frame
$\{B\}$ relative to $\{A\}$ → orientation of the body
Write unit vectors of principal axes of $\{B\}$ in terms of $\{A\}$.

- Dual-superscript notation: **Two** reference frames for description of kinematic vectors (linear position/velocity/acceleration of a point and angular velocity/acceleration of a frame)
  - **Defined** as viewed by an observer fixed in a reference frame: "relative to" or "with respect to" *observer*'s frame → Geometric vector
  - **Resolved** into components with respect to a reference frame: "referred to," "expressed in," or "written in" *writer*'s frame → Algebraic representation of the geometric vector

- $\begin{bmatrix} \hat{\mathbf{X}}_B \\ \hat{\mathbf{Y}}_B \\ \hat{\mathbf{Z}}_B \end{bmatrix}$: Principal unit vectors of $\{B\}$ written in terms of $\{B\}$

- $\begin{bmatrix} {}^{A}\hat{\mathbf{X}}_B \\ {}^{A}\hat{\mathbf{Y}}_B \\ {}^{A}\hat{\mathbf{Z}}_B \end{bmatrix}$: Principal unit vectors of $\{B\}$ written in terms of $\{A\}$

  → Columns of 3x3 **rotation matrix** (= direction cosine matrix) of $\{B\}$ relative to $\{A\}$: ${}^{A}_{B}R$ or ${}^{A}R_B$

- $\boxed{{}^{A}R_B = [\,{}^{A}\hat{\mathbf{X}}_B \;\vdots\; {}^{A}\hat{\mathbf{Y}}_B \;\vdots\; {}^{A}\hat{\mathbf{Z}}_B\,] = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}}$ (orientation)

- Note

  Position of a point→ vector (position vector)
  Orientation of a body→ matrix (rotation matrix)

- Note

	Configuration	Motion
Linear	....	....
Angular	....	....

- $$\boxed{{}^{A}R_{B} = [\,{}^{A}\hat{\mathbf{X}}_{B} \mid {}^{A}\hat{\mathbf{Y}}_{B} \mid {}^{A}\hat{\mathbf{Z}}_{B}\,] = \begin{bmatrix} \hat{\mathbf{X}}_{B}\bullet\hat{\mathbf{X}}_{A} & \hat{\mathbf{Y}}_{B}\bullet\hat{\mathbf{X}}_{A} & \hat{\mathbf{Z}}_{B}\bullet\hat{\mathbf{X}}_{A} \\ \hat{\mathbf{X}}_{B}\bullet\hat{\mathbf{Y}}_{A} & \hat{\mathbf{Y}}_{B}\bullet\hat{\mathbf{Y}}_{A} & \hat{\mathbf{Z}}_{B}\bullet\hat{\mathbf{Y}}_{A} \\ \hat{\mathbf{X}}_{B}\bullet\hat{\mathbf{Z}}_{A} & \hat{\mathbf{Y}}_{B}\bullet\hat{\mathbf{Z}}_{A} & \hat{\mathbf{Z}}_{B}\bullet\hat{\mathbf{Z}}_{A} \end{bmatrix}}$$ (arbitrary choice of frame for description)

  Elements are the **direction cosines**.
  ${}^{A}\hat{\mathbf{X}}_{B}$, ${}^{A}\hat{\mathbf{Y}}_{B}$, ${}^{A}\hat{\mathbf{Z}}_{B}$: unit orthogonal vectors

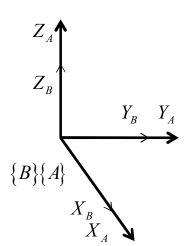
- Note: rotation matrix ${}^{[\text{with respect to which frame}]}R_{[\text{describe frame of interest}]}$ does not require the frame of expression

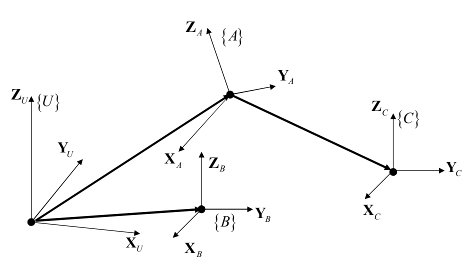
- Rows are unit vectors of $\{A\}$ expressed in $\{B\}$: $${}^{A}R_{B} = [\,{}^{A}\hat{\mathbf{X}}_{B} \mid {}^{A}\hat{\mathbf{Y}}_{B} \mid {}^{A}\hat{\mathbf{Z}}_{B}\,] = \begin{bmatrix} {}^{B}\hat{\mathbf{X}}_{A}^{T} \\ \hdashline {}^{B}\hat{\mathbf{Y}}_{A}^{T} \\ \hdashline {}^{B}\hat{\mathbf{Z}}_{A}^{T} \end{bmatrix}$$

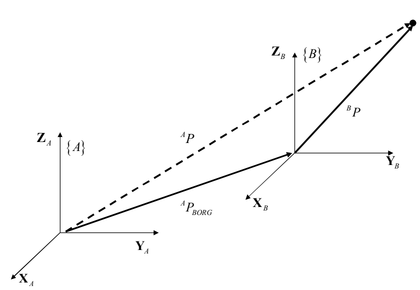
  → ${}^{A}R_{B} = {}^{B}R_{A}^{T}$ and ${}^{A}R_{B} = {}^{B}R_{A}^{-1}$
  $\therefore\ {}^{A}R_{B} = {}^{B}R_{A}^{-1} = {}^{B}R_{A}^{T}$ → Rotation matrix is .... matrix (i.e., $RR^{T} = I_3$).

- Example

$${}^{A}R_{B} = \begin{bmatrix} \hat{\mathbf{X}}_{B}\bullet\hat{\mathbf{X}}_{A} & \hat{\mathbf{Y}}_{B}\bullet\hat{\mathbf{X}}_{A} & \hat{\mathbf{Z}}_{B}\bullet\hat{\mathbf{X}}_{A} \\ \hat{\mathbf{X}}_{B}\bullet\hat{\mathbf{Y}}_{A} & \hat{\mathbf{Y}}_{B}\bullet\hat{\mathbf{Y}}_{A} & \hat{\mathbf{Z}}_{B}\bullet\hat{\mathbf{Y}}_{A} \\ \hat{\mathbf{X}}_{B}\bullet\hat{\mathbf{Z}}_{A} & \hat{\mathbf{Y}}_{B}\bullet\hat{\mathbf{Z}}_{A} & \hat{\mathbf{Z}}_{B}\bullet\hat{\mathbf{Z}}_{A} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$







<u>Frame</u>

Describes one coordinate system with respect to another.
Represents both position and orientation.
A set of four vectors – position vector and rotation matrix

- Position description – in general, choose the origin of the body-attached (= local) frame

- $\{B\} = \{{}^{A}R_{B},\ {}^{A}\mathbf{P}_{BORG}\}$

Mapping

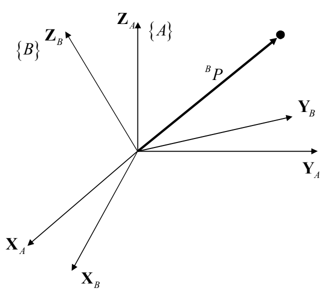
Changing **descriptions** (only!) from frame to frame
Original vector is not changed in space
Computes new description of the vector relative to another frame

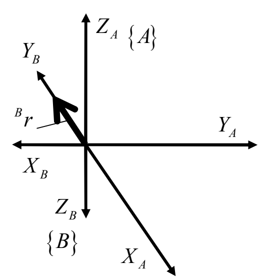
- Mapping of translation (**same orientations**): ${}^{A}\mathbf{P} = {}^{B}\mathbf{P} + {}^{A}\mathbf{P}_{BORG}$
  (Note: vector additions in terms of different frames can be calculated only when their orientations are equivalent!)

- Mapping of rotation (**same origins**)

$$\rightarrow \text{Components of } {}^{A}\mathbf{P}: \begin{cases} {}^{A}p_x = {}^{B}\hat{\mathbf{X}}_A \bullet {}^{B}\mathbf{P} = {}^{B}\hat{\mathbf{X}}_A^T\,{}^{B}\mathbf{P} \\ {}^{A}p_y = {}^{B}\hat{\mathbf{Y}}_A \bullet {}^{B}\mathbf{P} = {}^{B}\hat{\mathbf{Y}}_A^T\,{}^{B}\mathbf{P} \\ {}^{A}p_z = {}^{B}\hat{\mathbf{Z}}_A \bullet {}^{B}\mathbf{P} = {}^{B}\hat{\mathbf{Z}}_A^T\,{}^{B}\mathbf{P} \end{cases} \Rightarrow {}^{A}\mathbf{P} = \begin{bmatrix} {}^{B}\hat{\mathbf{X}}_A^T \\ \hdashline {}^{B}\hat{\mathbf{Y}}_A^T \\ \hdashline {}^{B}\hat{\mathbf{Z}}_A^T \end{bmatrix} {}^{B}\mathbf{P}$$

$\therefore\ {}^{A}\mathbf{P} = {}^{A}R_{B}\,{}^{B}\mathbf{P}$: mapping of a same vector's description from $\{B\}$ to $\{A\}$.

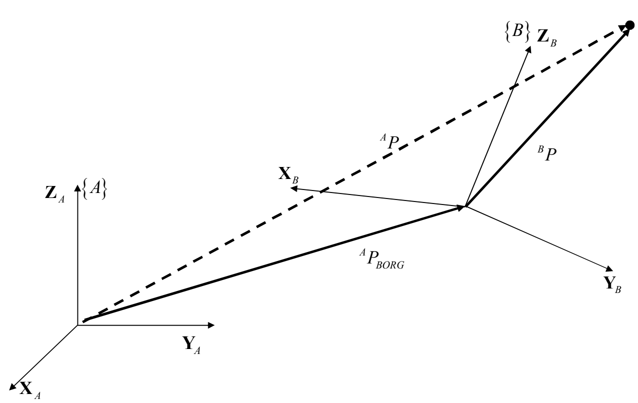


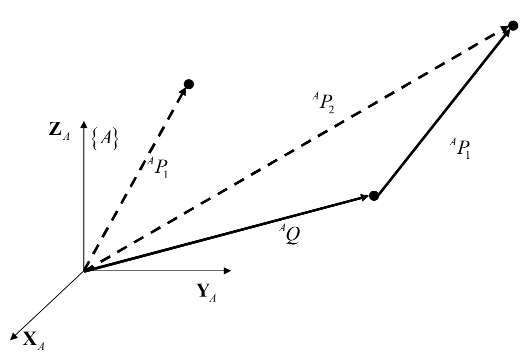


- Example

$${}^{A}R_{B} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix};\ {}^{B}\mathbf{r} = [0\ \ 2\ \ 0]^T;\ {}^{A}\mathbf{r} = {}^{A}R_{B}\,{}^{B}\mathbf{r} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}\begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix} = \begin{bmatrix} -2 \\ 0 \\ 0 \end{bmatrix}$$

- Mapping of general frames' transformation: translation + rotation (affine transformation)
  $\boxed{{}^{A}\mathbf{P} = {}^{A}R_{B}\,{}^{B}\mathbf{P} + {}^{A}\mathbf{P}_{BORG}}$ (Note 1: nonhomogeneous; Note 2: ${}^{A}R_{\cancel{B}}\,{}^{\cancel{B}}\mathbf{P} \neq {}^{A}\mathbf{P}$)





- Construct a 4x4 "augmented" matrix operator T using 4x1 "augmented" position vectors

$$\begin{bmatrix} {}^{A}\mathbf{P} \\ \hline 1 \end{bmatrix} = \underbrace{\left[\begin{array}{ccc|c} \multicolumn{3}{c|}{\underbrace{{}^{A}R_{B}}_{orientation}} & \underbrace{{}^{A}\mathbf{P}_{BORG}}_{position} \\ \hline 0 & 0 & 0 & 1 \end{array}\right]}_{={}^{A}T_{B}} \begin{bmatrix} {}^{B}\mathbf{P} \\ \hline 1 \end{bmatrix} \Rightarrow {}^{A}\mathbf{P} = {}^{A}T_{B}\, {}^{B}\mathbf{P}$$

${}^{A}T_{B} = \left[\begin{array}{c|c} {}^{A}R_{B} & {}^{A}\mathbf{P}_{BORG} \\ \hline \mathbf{0}^{T} & 1 \end{array}\right]$: **Homogeneous transform** – describes $\{B\}$ relative to $\{A\}$; mapping ${}^{B}\mathbf{P} \mapsto {}^{A}\mathbf{P}$

- Example

Operators

→ Transform points and/or vectors in a given frame (**only one coordinate system** is involved)
- Use the mapping transform

- Translational operators: moves a point in space by a vector

→ ${}^{A}\mathbf{P}_1$ translated by ${}^{A}\mathbf{Q} = \begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix}$: ${}^{A}\mathbf{P}_2 = {}^{A}\mathbf{P}_1 + {}^{A}\mathbf{Q}$

→ Matrix operator: ${}^{A}\mathbf{P}_2 = D_Q(q)\, {}^{A}\mathbf{P}_1$ ($q = \left\|\hat{Q}\right\| = \sqrt{q_x^2 + q_y^2 + q_z^2}$)

$$D_Q(q) = \begin{bmatrix} I_3 & & & \hat{Q} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & q_x \\ 0 & 1 & 0 & q_y \\ 0 & 0 & 1 & q_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Rotational operators: rotates ${}^{A}\mathbf{P}_1$ to become ${}^{A}\mathbf{P}_2$ by means of R

→ ${}^{A}\mathbf{P}_2 = R\, {}^{A}\mathbf{P}_1$ or ${}^{A}\mathbf{P}_2 = R_K(\theta)\, {}^{A}\mathbf{P}_1$ ($\hat{K}$: axis direction, θ : angle)

Example: $R_z(\theta) = \left[\begin{array}{ccc|c} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$ (3x3 or 4x4)

- General transformation operator: Frame

→ ${}^{A}\mathbf{P}_2 = T\, {}^{A}\mathbf{P}_1$: T operates on (i.e., rotates and translates) ${}^{A}\mathbf{P}_1$ to compute ${}^{A}\mathbf{P}_2$

Transformation Arithmetic

- Compound: ${}^{A}T_C = {}^{A}T_B\, {}^{B}T_C \Rightarrow {}^{A}T_C = \left[\begin{array}{c|c} {}^{A}R_B\, {}^{B}R_C & {}^{A}R_B\, {}^{B}\mathbf{P}_{CORG} + {}^{A}\mathbf{P}_{BORG} \\ \hline 0\ 0\ 0 & 1 \end{array}\right]$

- Inversion: ${}^{B}\mathbf{P}_{BORG} = {}^{B}R_{A}\,{}^{A}\mathbf{P}_{BORG} + {}^{B}\mathbf{P}_{AORG} = \mathbf{0}$
  (Note: The point of interest is $\mathbf{P}_{BORG}$. Thus the notation ${}^{B}({}^{A}\mathbf{P}_{BORG})$ in the textbook is not proper.)

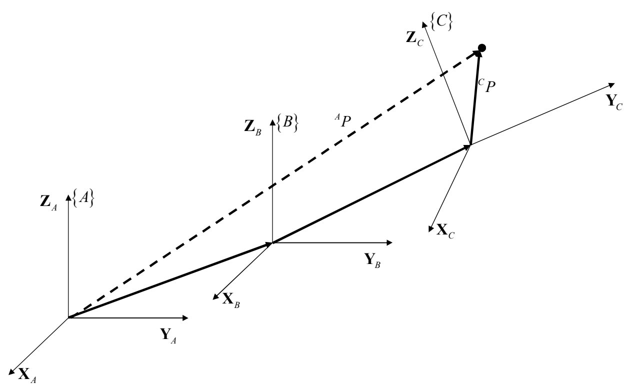
  $$\Rightarrow\ {}^{B}\mathbf{P}_{AORG} = -{}^{B}R_{A}\,{}^{A}\mathbf{P}_{BORG} = -{}^{A}R_{B}^{T}\,{}^{A}\mathbf{P}_{BORG} \ \Rightarrow\ {}^{A}T_{B}^{-1} = {}^{B}T_{A} = \left[\begin{array}{c|c} {}^{A}R_{B}^{T} & -{}^{A}R_{B}^{T}\,{}^{A}\mathbf{P}_{BORG} \\ \hline 0\ 0\ 0 & 1 \end{array}\right]$$

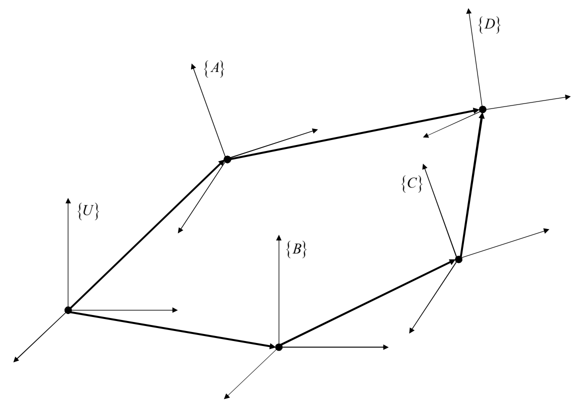
  - Alternative derivation: ${}^{A}\mathbf{P} = {}^{A}R_{B}\,{}^{B}\mathbf{P} + {}^{A}\mathbf{P}_{BORG}$

    $$\Rightarrow\ {}^{A}R_{B}^{T}\,{}^{A}\mathbf{P} = {}^{A}R_{B}^{T}\,{}^{A}R_{B}\,{}^{B}\mathbf{P} + {}^{A}R_{B}^{T}\,{}^{A}\mathbf{P}_{BORG} = {}^{B}\mathbf{P} + {}^{A}R_{B}^{T}\,{}^{A}\mathbf{P}_{BORG} \ \Rightarrow\ {}^{B}\mathbf{P} = {}^{A}R_{B}^{T}\,{}^{A}\mathbf{P} - {}^{A}R_{B}^{T}\,{}^{A}\mathbf{P}_{BORG}$$

    $$\Rightarrow\ \left[\begin{array}{c} {}^{B}\mathbf{P} \\ \hline 1 \end{array}\right] = \left[\begin{array}{c|c} {}^{A}R_{B}^{T} & -{}^{A}R_{B}^{T}\,{}^{A}\mathbf{P}_{BORG} \\ \hline \mathbf{0}^{T} & 1 \end{array}\right]\left[\begin{array}{c} {}^{A}\mathbf{P} \\ \hline 1 \end{array}\right]$$

- Transform equation: ${}^{U}T_{A}\,{}^{A}T_{D} = {}^{U}T_{B}\,{}^{B}T_{C}\,{}^{C}T_{D}$





Orientation

- Rotation matrix $R = [\qquad\qquad]$ → $Det(R) = 1$ (i.e., Proper orthonormal matrix)

  Recall: ${}^{A}R_{B}\,{}^{B}R_{C} \neq {}^{B}R_{C}\,{}^{A}R_{B}$ (not commutative)

- Cayley's formula: $R = (I_3 - S)^{-1}\,(I_3 + S)$ (where S is a skew-symmetric matrix; ______)

- $S = \begin{bmatrix} 0 & -s_z & s_y \\ s_z & 0 & -s_x \\ -s_y & s_x & 0 \end{bmatrix}$ → $\therefore$ R: 3 independent parameters

- $\|\hat{\mathbf{X}}\| = \|\hat{\mathbf{Y}}\| = \|\hat{\mathbf{Z}}\| = 1$ and $\hat{\mathbf{X}}\bullet\hat{\mathbf{Y}} = \hat{\mathbf{X}}\bullet\hat{\mathbf{Z}} = \hat{\mathbf{Y}}\bullet\hat{\mathbf{Z}} = 0$ → 9 elements and 6 equations → $\therefore$ ___ unknowns

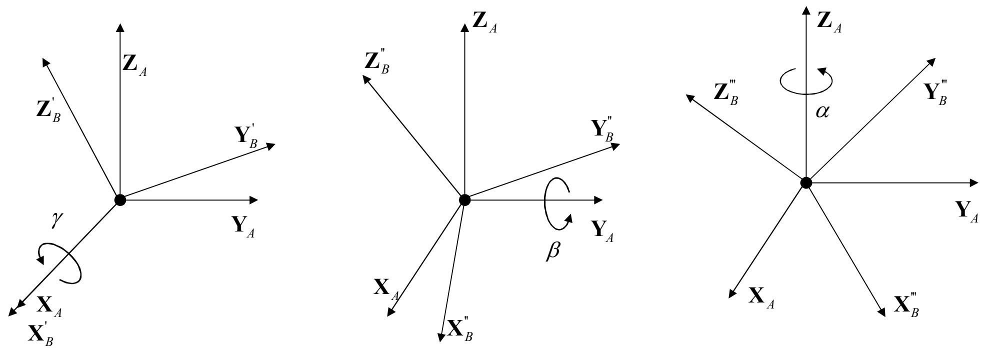
Rotation of Frames

Fixed angle rotation (absolute transform)
Moving (i.e., current) frame rotation (relative transform)

Fixed Angle Rotation

Rotations are specified about the fixed frame.
Each of three rotations takes place about an axis in the fixed frame (e.g., $\{A\}$).

- *X*-*Y*-*Z* fixed angles (roll-pitch-yaw): initially $\{B\}$ coincides with $\{A\}$
  → (1) rotate $\{B\}$ about $\hat{\mathbf{X}}_A$ by γ → (2) rotate $\{B\}$ about $\hat{Y}_A$ by β → (3) rotate $\{B\}$ about $\hat{Z}_A$ by α



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

$$= \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\gamma & -s\gamma \\ 0 & s\gamma & c\gamma \end{bmatrix} = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix}$$

- "Multiply rotation matrices **from right to left**; premultiplying" (rotations as operators)

- Let ${}^{A}R_{BXYZ}(\gamma,\beta,\alpha) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$. If $c\beta \neq 0$, then $\begin{cases} \beta = \text{Atan2}(-r_{31}, \sqrt{r^2{}_{11} + r^2{}_{21}}) \\ \alpha = \text{Atan2}(r_{21}/c\beta, r_{11}/c\beta) \\ \gamma = \text{Atan2}(r_{32}/c\beta, r_{33}/c\beta) \end{cases}$.

  (Atan2(y, x): two-argument arc tangent function or four-quadrant arc tangent)

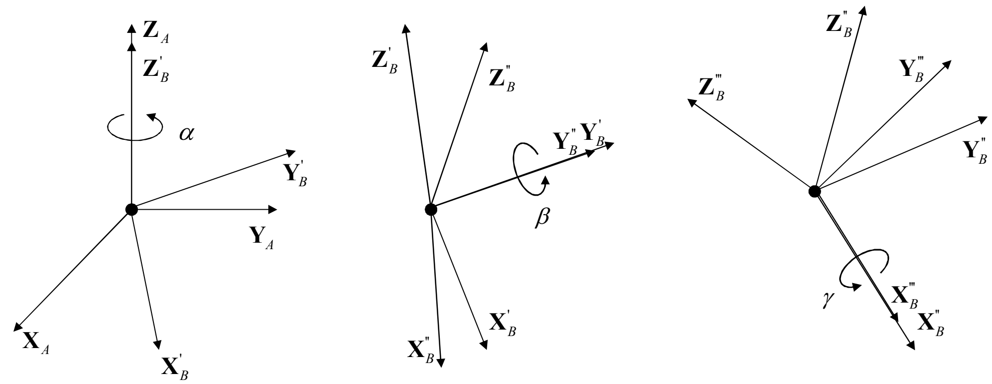
$$\theta = \text{Atan2}(y,x) = \begin{cases} 0 \le \theta \le 90 & +x \quad +y \\ 90 \le \theta \le 180 & -x \quad +y \\ -180 \le \theta \le -90 & -x \quad -y \\ -90 \le \theta \le 0 & +x \quad -y \end{cases}$$

- For one-to-one function, assume $-90.0^o \le \beta \le 90.0^o$.
- If $\beta = \pm 90.0^o$ (i.e., $\cos\beta = 0$): singular → only sum or difference of α and γ available. Choose arbitrary α or γ (e.g., $\alpha = 0.0$). (Read textbook for further development.)

<u>Moving Frame Rotation</u>

Each rotation is performed about an axis of the moving system (e.g., $\{B\}$).
Euler angles

- *Z*-*Y*-*X* Euler angles: initially $\{B\}$ coincides with $\{A\}$
  → (1) rotate $\{B\}$ about $\hat{Z}_B$ by α → (2) rotate $\{B\}$ about $\hat{Y}_B$ by β → (3) rotate $\{B\}$ about $\hat{X}_B$ by γ



- ${}^{A}R_{B} = {}^{A}R_{B'}\,{}^{B'}R_{B''}\,{}^{B''}R_{B}$ ($\because$ for a given vector $\mathbf{P}$, ${}^{A}\mathbf{P} = {}^{A}R_{B'}\,{}^{B'}\mathbf{P}$, ${}^{B'}\mathbf{P} = {}^{B'}R_{B''}\,{}^{B''}\mathbf{P}$, and ${}^{B''}\mathbf{P} = {}^{B''}R_{B}\,{}^{B}\mathbf{P}$)

$$\boxed{{}^{A}R_{BZ'Y'X'}(\alpha,\beta,\gamma) = R_Z(\alpha)R_Y(\beta)R_X(\gamma)}$$
$$= \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\gamma & -s\gamma \\ 0 & s\gamma & c\gamma \end{bmatrix} = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix}$$

- “Multiply rotation matrices **from left to right**; postmultiplying” (rotations as mapping)
- Note: Same final orientation as the fixed axes rotation in **opposite** order.

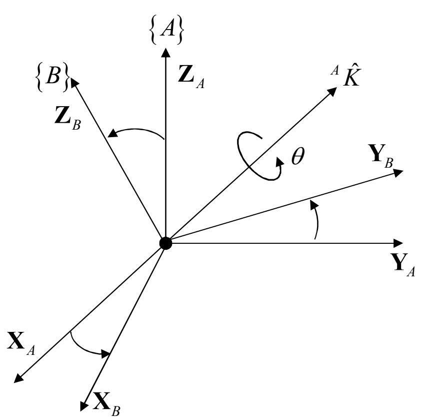
- *Z-Y-Z* Euler angles: (Read textbook)

- 24 Angle set conventions (12 fixed angles + 12 Euler angles)

Equivalent Angle-Axis

> If the axis is a general direction, any orientation may be obtained through proper axis and angle selection.

- Euler’s theorem on rotation: initially $\{B\}$ coincides with $\{A\}$
  → rotate $\{B\}$ about ${}^{A}\hat{K}$ by θ (according to right hand rule)
    ${}^{A}\hat{K}$: Equivalent axis of finite rotation; unit vector
    $K = \theta \cdot {}^{A}\hat{K}$: 3x1 orientation vector



- Equivalent rotation matrix for ${}^{A}\hat{K} = [k_x \ k_y \ k_z]^T$

$$R_K(\theta) = {}^{A}R_B(\hat{K},\theta) = \begin{bmatrix} k_x k_x v\theta + c\theta & k_x k_y v\theta - k_z s\theta & k_x k_z v\theta + k_y s\theta \\ k_y k_x v\theta + k_z s\theta & k_y k_y v\theta + c\theta & k_y k_z v\theta - k_x s\theta \\ k_z k_x v\theta - k_y s\theta & k_z k_y v\theta + k_x s\theta & k_z k_z v\theta + c\theta \end{bmatrix}$$

(versed sine: versine(θ) = vers(θ) = $v\theta = 1 - c\theta$)

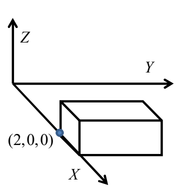
Examples: $R_X(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$, $R_Y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$, $R_Z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$

- Rotate a vector Q about a vector $\hat{K}$ by θ → a new vector Q'
  Rodriques' formula: $Q' = R_K(\theta)Q = Q\cos\theta + \sin\theta\left(\hat{K} \times Q\right) + (1-\cos\theta)\left(\hat{K} \cdot Q\right)\hat{K}$

- Let ${}^{A}R_{BK}(\theta) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$ => $\theta = \text{Acos}\left(\dfrac{r_{11} + r_{22} + r_{33} - 1}{2}\right)$; $\hat{K} = \dfrac{1}{2\sin\theta}\begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix}$ ($0^\circ < \theta < 180^\circ$)
- $({}^{A}\hat{K},\theta) \equiv (-{}^{A}\hat{K},-\theta)$
- Small angular rotation: $\theta \to 0$ => ill-defined rotation axis ($\theta = 0$ or $\theta = \pi$)
- Two special cases
  i) $\theta = 0$: No rotation; R is identity; any nonzero $\hat{K}$ is suitable
  ii) $\theta = \pi$: Half turn; sense of axis vector is arbitrary; $R(\hat{K},\pi) = R(-\hat{K},\pi)$
  To find $\hat{K}$, set $\sin\theta = 0$, $\cos\theta = -1$, and $v\theta = 1 - \cos\theta = 2$, and use the first row of R
  => $2K_x^2 - 1 = r_{11}$, $2K_xK_y = r_{12}$, $2K_yK_z = r_{13}$
  => $\therefore\ K_x = \sqrt{(1 + r_{11})/2}$, $K_y = \dfrac{r_{12}}{2K_x} = \dfrac{r_{12} + r_{21}}{4K_x}$, $K_z = \dfrac{r_{13} + r_{31}}{4K_x}$

- Rotation about $\hat{K}$ which does not pass through the origin
  : [position change] + [same final orientation as if $\hat{K}$ had passed through the origin]

- Example: Rotate about Z-axis
  $K = [0 \ 0 \ 1]^T$; $\phi = 90^o$; $\mathbf{r} = [2 \ 0 \ 0]^T$
  => $\mathbf{r}' = [0 \ 2 \ 0]^T$



- Example 2.9 (Craig's 4th Ed.): A frame $\{B\}$ is described as initially coincident with $\{A\}$. We then rotate $\{B\}$ about the vector ${}^{A}\hat{K} = [0.707\ \ 0.707\ \ 0.0]^{T}$ (passing through point ${}^{A}P$ = [1.0 2.0 3.0]) by an amount $\theta = 30$ degrees. Give the frame description of $\{B\}$.
  (Do it yourself)

- Exercise 2.14 (Craig's 4th Ed.): (Do it yourself)

Euler Parameters (= Unit Quaternion)
(Skip)

Transformation of Free Vectors
Equal vectors: same magnitude and direction
Equivalent vectors: produce same effect in a certain capacity

- Vector quantities
  Free vector: may be positioned anywhere in space (e.g., couple vector on a rigid body, translational velocity of a nonrotating body)
  Sliding (or line) vector: effects depend on specified line of action (e.g., force applied on a rigid body)
  Bound (or fixed) vector: effects depend on point of application (e.g., force applied on a deformable body, force applied on a particle)

CHAPTER 3. MANIPULATOR KINEMATICS

- Kinematics: Science of motion without regard to the forces and moments that cause it

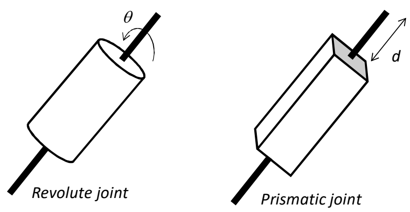
Link Description

- Definitions

  Manipulator: A set of bodies (links) connected in a **chain** by joints
  Links: Bodies of a manipulator or a chain. Mathematical concept relating two neighboring joint axes
  Joints: Connection between a neighboring pair of links

- In robotics, for **modeling**, each joint has one DOF. → One link – one joint – one DOF

  revolute joint vs. prismatic (or sliding) joint



- Note: A (physical) joint with n DOF can be modeled as n joints (revolute and prismatic combined) of one DOF connected with n-1 links of zero and/or non-zero lengths.

- Numbering of links

  Link 0: immobile base of manipulator (e.g., inertial frame, reference frame, ground, etc.)
  Link 1: first moving body
  Link i: ith moving body
  Link n: free end of manipulator

- Joint axis i: vector direction about which link i rotates relative to link i-1

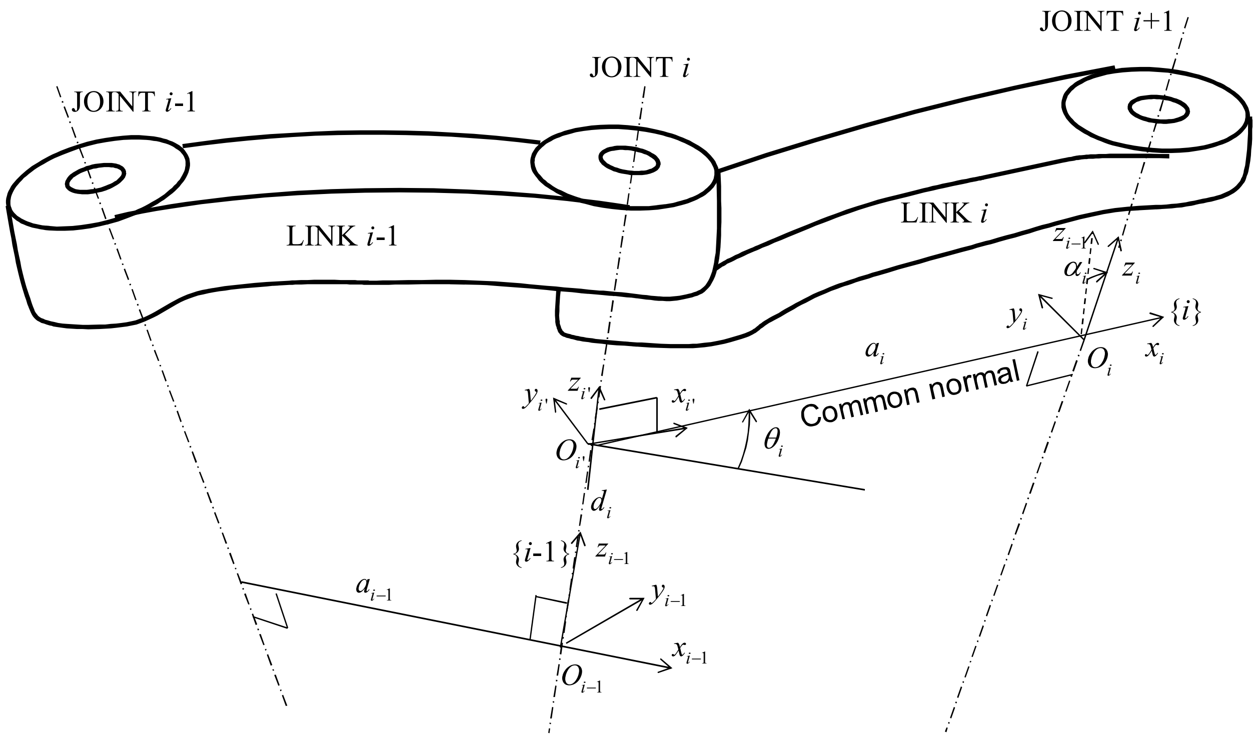
- Recall: Distance between any two axes in 3D is that of the common normal which is perpendicular to both axes.
  (Existence and uniqueness except for parallel axes; parallel axes have infinite number of mutual perpendiculars of equal length.)

Denavit-Hartenberg (DH) Convention

(References: [1] Sciavicco and Siciliano, *Modeling and Control of Robot Manipulators*, McGraw Hill, 1996; [2] Spong, M.W., Hutchinson, S., and Vidyasagar, M., *Robot Modeling and Control*, Wiley, 2006)

- Overall steps for standard DH Convention

  [DH STEP I] Attach a local frame to each link. Frame $\{i\}$ is attached rigidly to link i.
  [DH STEP II] Assign DH parameters and construct DH table.
  [DH STEP III] Compute homogeneous transformation matrices and forward kinematics.



[DH STEP I] Attach a local frame to each link

- Define and attach link Frame $\{i\}$:
  - **Step I-1)** Let Joint Axis i denote the axis of the joint connecting Link i-1 to Link i.
  - **Step I-2)** Choose axis z_i along the axis of Joint i+1.
  - **Step I-3)** Locate the origin O_i at the intersection of axis z_i with the common normal to axes z_{i-1} and z_i. Also, locate $O_{i'}$ at the intersection of the common normal with axis z_{i-1}.
  - **Step I-4)** Choose axis x_i along the common normal to axes z_{i-1} and z_i with direction from Joint i to Joint i+1. ($x_i \perp z_{i-1}$ and points away from z_{i-1})
  - **Step I-5)** Choose axis y_i so as to complete the right-handed frame.

- The DH convention gives a nonunique definition of link frames in the following cases:
  - **Case 1)** For Frame $\{0\}$, only direction of axis z_0 is specified; then O_0 and x_0 can be arbitrarily chosen.
  - **Case 2)** For Frame $\{n\}$, since there is no Joint n+1, z_n is not uniquely defined while x_n has to be normal to z_{n-1}. Typically, Joint n is revolute, and thus z_n is to be aligned with z_{n-1}.
  - **Case 3)** When two consecutive axes are parallel, the common normal between them is not uniquely defined.
  - **Case 4)** When two consecutive axes z_{i-1} and z_i intersect, x_i is chosen normal to the plane formed by z_{i-1} and z_i. The positive direction of x_i is arbitrary. The most natural choice for the origin O_i in this case is at the point of intersection of z_{i-1} and z_i. Note that, in this case, $a_i = 0$. (In general, the line that is normal to the plane formed by two intersecting axes can be viewed as a converging case of the common normal of two non-intersecting axes as they approach to each other and eventually intersect.)
  - **Case 5)** When Joint i is prismatic, the direction sense of z_{i-1} is arbitrary.

- In general, 6 parameters are required for the transformation between two frames. However, the DH convention imposes the following 2 conditions, reducing the required number of parameters to 4:

  $\left.\begin{array}{l}\textbf{DH1)}\ x_i \perp z_{i-1}. \\ \textbf{DH2)}\ x_i \text{ and } z_{i-1} \text{ axes intersect.}\end{array}\right\}$ → x_i along the common normal to axes z_{i-1} and z_i

[DH STEP II] Assign DH parameters and construct DH table

- Once the link frames have been established, the position and orientation of Frame $\{i\}$ with respect to Frame $\{i\text{-}1\}$ are completely specified by the following DH parameters.
  - θ_i : Angle between axes x_{i-1} and x_i about axis z_{i-1} to be taken positive with counter-clockwise
  - d_i: Coordinate (+/–) of $O_{i'}$ along z_{i-1}
  - a_i: Distance between O_i and $O_{i'}$ (Note: x_i "with direction from Joint i to Joint i+1" or "points away from z_{i-1}" ensures that a_i is positive, thus distance, not coordinate.)
  - α_i : Angle between axes z_{i-1} and z_i about axis x_i to be taken positive with counter-clockwise

- Reference configuration (= home configuration = zero configuration) of a robot manipulator
  : configuration with respect to which the joint displacements of the manipulator are measured
  - The configuration of a manipulator when all joint variables are equal to zero
  - The location of the end-effector and the locations of the joint axes are known.
  - Can be chosen arbitrarily; usually chosen at the location where the coordinates of all joint axes can be easily identified
  - DH parameters do not represent the angle of rotation or the distance of translation about a joint axis.

- Target configuration (= desired configuration)
  - Manipulator displaced from the reference configuration to the target configuration by a series of joint displacements about all joint axes.
  - To obtain actual joint displacements, subtract joint variables associated with the reference configuration from that of a target configuration.

- Link parameters (design) and joint variables (control) (where $\tilde{\theta}_i$ and $\tilde{d}_i$ are reference configurations)
  - If joint is revolute $\theta_i = \tilde{\theta}_i + q_i$ → joint variable: q_i link parameters: d_i, a_i, α_i
  - If joint is prismatic $d_i = \tilde{d}_i + q_i$ → joint variable: q_i link parameters: θ_i, a_i, α_i

  → Joint variable vector: $\mathbf{q} = \begin{bmatrix} q_1 \\ \vdots \\ q_n \end{bmatrix}$ (for n-DOF manipulator)

- DH parameter table

  For two types Joint i: revolute and prismatic

Joint i	θ_i	d_i	a_i	α_i	Joint variable $\mathbf{q}$
Revolute	$\theta_i = \tilde{\theta}_i + q_i$	d_i	a_i	α_i	q_i
Prismatic	θ_i	$d_i = \tilde{d}_i + q_i$	a_i	α_i	q_i

For an n-DOF manipulator

Joint #	θ_i	d_i	a_i	α_i	Joint variable $\mathbf{q}$
1 (if revolute)	$\theta_1 = \tilde{\theta}_1 + q_1$	d_1	a_1	α_1	q_1
2 (if prismatic)	θ_2	$d_2 = \tilde{d}_2 + q_2$	a_2	α_2	q_2
⋮	⋮	⋮	⋮	⋮	⋮
n (if revolute)	$\theta_n = \tilde{\theta}_n + q_n$	d_n	a_n	α_n	q_n

- Note: "DH parameter = home configuration + joint variable" in DH parameters table
  - Clarifies which DH parameter corresponds to joint degree of freedom (revolute or prismatic)
  - Identifies the home configuration parameter value of the corresponding joint

[DH STEP III] Compute homogeneous transformation matrices and forward kinematics

- Derivation of link transformation ${}^{i-1}T_i$: define Frame $\{i\}$ relative to Frame $\{i-1\}$

  → Four transformations (sub-problems) – each of four transformations will be a function of one DH parameter only

  $Rot(z,\theta_i)$: $T_{z,\theta}$ => $Trans(0,0,d_i)$: $T_{z,d}$ => $Trans(a_i,0,0)$: $T_{x,a}$ => $Rot(x,\alpha_i)$: $T_{x,\alpha}$

  (Note: the rotations in this case are moving frame rotations; Euler angles; .... product)

$$
{}^{i-1}T_i = T_{z,\theta} T_{z,d} T_{x,a} T_{x,\alpha}
$$

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

$$
\therefore \boxed{ {}^{i-1}T_i = \left[\begin{array}{ccc|c} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ \hline 0 & 0 & 0 & 1 \end{array}\right] }
$$

- ${}^{i-1}T_i = {}^{i-1}T_i(q_i)$: function of only **one** variable q_i where $\begin{cases} \theta_i = \tilde{\theta}_i + q_i & \text{for revolute joint} \\ d_i = \tilde{d}_i + q_i & \text{for prismatic joint} \end{cases}$

- $Screw_Q(r,\phi)$: translation by distance r along, and rotation by angle ϕ about axis $\hat{Q}$

  Examples: $Screw_z(d,\theta) = T_{z,\theta} T_{z,d}$ and $Screw_x(a,\alpha) = T_{x,a} T_{x,\alpha}$

- **Forward kinematics** – concatenating link transformations

  ${}^0T_n = {}^0T_1(q_1)\,{}^1T_2(q_2)\ldots{}^{i-1}T_i(q_i)\ldots{}^{n-1}T_n(q_n)$ (for n-DOF manipulator)

  ${}^0T_i = {}^0T_i(q_1,q_2,\ldots,q_i)$: function of the first i joint variables

  → computes Cartesian position and orientation of the ith link

  ${}^0T_n = {}^0T_n(q_1,q_2,\ldots,q_n)$: function of all n joint variables

  → computes Cartesian position and orientation of the last (nth) link

DH Convention Procedure Summary

1) Find and number consecutively the joint axes; set the directions of axes $z_0, \ldots, z_{n-1}$.
2) Choose Frame {0} by locating the origin on axis z_0; axes x_0 and y_0 are chosen according to right-hand rule. If feasible, it is worth choosing Frame {0} to coincide with the base frame.

Execute steps **3** to **5** for $i = 1, \ldots, n-1$:

3) Locate the origin O_i at the intersection of z_i with the common normal to axes z_{i-1} and z_i. If axes z_{i-1} and z_i are parallel and Joint i is revolute, then locate O_i so that $d_i = 0$; if Joint i is prismatic, locate O_i at a reference position for the joint range, e.g., a mechanical limit.
4) Choose axis x_i along the common normal to axes z_{i-1} and z_i with direction from Joint i to Joint $i+1$.
5) Choose axis y_i according to right-hand rule.

To complete:

6) Choose Frame $\{n\}$; if Joint n is revolute, then align z_n and z_{n-1}; otherwise, if Joint n is prismatic, then choose z_n arbitrarily. Axis x_n is set according to step **4**.
7) For $i = 1, \ldots, n$, construct the table of DH parameters θ_i, d_i, a_i, α_i.
8) On the basis of the DH parameters in **7**, compute the homogenous transformation matrices ${}^{i-1}T_i(q_i)$ for $i = 1, \ldots, n$.
9) Compute the homogenous transformation ${}^0T_n(\mathbf{q}) = {}^0T_1 \ldots {}^{n-1}T_n$ that yields the position and orientation of Frame $\{n\}$ with respect to Frame {0}.
10) Given ${}^bT_0$ (from base to Frame {0}) and ${}^nT_e$ (from Frame $\{n\}$ to end-effector), compute the direct kinematic function as ${}^bT_e(\mathbf{q}) = {}^bT_0 {}^0T_n {}^nT_e$ that yields the position and orientation of the end-effector frame with respect to the base frame.

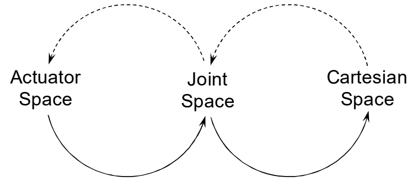
DH Parameters (Quick Summary)

θ_i (joint angle): joint angle from x_{i-1} to x_i about z_{i-1} (revolute joint variable)
d_i (link offset): shortest distance between x_{i-1} to x_i axis (prismatic joint variable)
a_i (link length): shortest distance between z_{i-1} and z_i axis
α_i (link twist): angle from z_{i-1} to z_i about x_i axis

(Note: shortest distance between axes = length of the common normal)

Actuator Space, Joint Space, and Cartesian Space

- nx1 joint vector **q**: set of n joint variables (generalized coordinates) that specifies the position and orientation of all the links of an n-DOF manipulator
- Joint space: vector space of all joint vectors
- Cartesian space = task-oriented space = operational space
- Actuator vector: actuator positions (determine joint vector) → actuator space
  - Examples: two actuators for a single joint, four-bar linkage (linear → revolute), muscles, etc.
- Mappings between 3 different space representations of manipulator's position and orientation



Example: 2-Link 2R Planar Manipulator

- DH parameters table

i	θ	d	a	α	Joint variable q_i
1	$\theta_1 = 0° + q_1$	0	a_1	0	q_1
2	$\theta_2 = 0° + q_2$	0	a_2	0	q_2

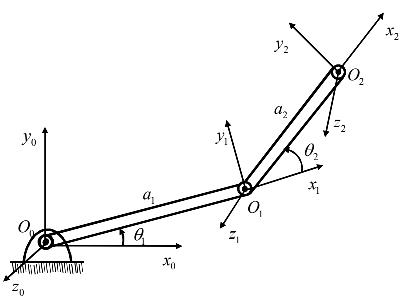
- Homogeneous transformation matrix in joint space

$$
{}^0T_2 = {}^0T_1\,{}^1T_2 = \begin{bmatrix} c_1 & -s_1 & 0 & a_1c_1 \\ s_1 & c_1 & 0 & a_1s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} c_2 & -s_2 & 0 & a_2c_2 \\ s_2 & c_2 & 0 & a_2s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_1c_2 - s_1s_2 & -c_1s_2 - s_1c_2 & 0 & a_1c_1 + a_2c_1c_2 - a_2s_1s_2 \\ c_1s_2 + s_1c_2 & c_1c_2 - s_1s_2 & 0 & a_1s_1 + a_2s_1c_2 + a_2c_1s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

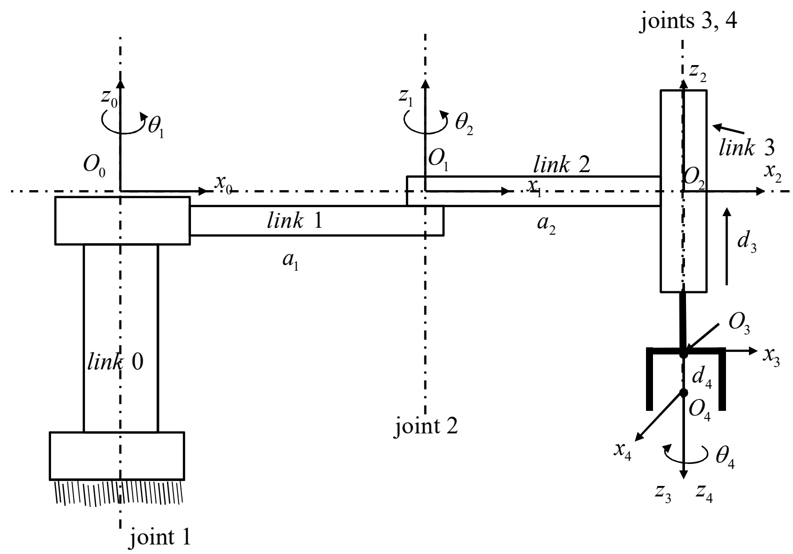
$$\cos(\theta_1 + \theta_2) = \cos\theta_1 \cos\theta_2 - \sin\theta_1 \sin\theta_2 \Rightarrow c_{12} = c_1c_2 - s_1s_2$$

$$\sin(\theta_1 + \theta_2) = \sin\theta_1 \cos\theta_2 + \cos\theta_1 \sin\theta_2 \Rightarrow s_{12} = s_1c_2 + c_1s_2$$

$$
\therefore\ {}^0T_2 = \begin{bmatrix} c_{12} & -s_{12} & 0 & a_1c_1 + a_2c_{12} \\ s_{12} & c_{12} & 0 & a_1s_1 + a_2s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$



2R Planar Manipulator



SCARA Robot

Example: SCARA Robot

- Local frame: since all joint axes are parallel, the locations of the origins are not unique. In this case, the origins are located at each joint.
- DH parameters table

i	θ_i	d_i	a_i	α_i	Joint variable q_i
1	$\theta_1 = 0 + q_1$	0	a_1	0	q_1

2	$\theta_2 = 0 + q_2$	0	a_2	0	q_2
3	0	$d_3 = 0 + q_3$	0	π	q_3
4	$\theta_4 = 0 + q_4$	d_4	0	0	q_4

- Homogeneous transformation matrices

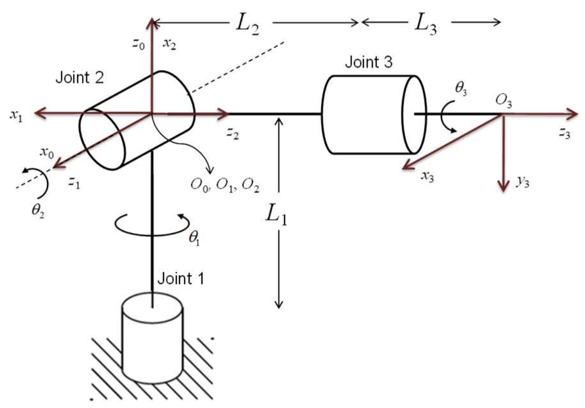
$$ {}^0T_1 = \begin{bmatrix} c_1 & -s_1 & 0 & a_1c_1 \\ s_1 & c_1 & 0 & a_1s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \ {}^1T_2 = \begin{bmatrix} c_2 & -s_2 & 0 & a_2c_2 \\ s_2 & c_2 & 0 & a_2s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \ {}^2T_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \ {}^3T_4 = \begin{bmatrix} c_4 & -s_4 & 0 & 0 \\ s_4 & c_4 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

- Forward kinematics equation

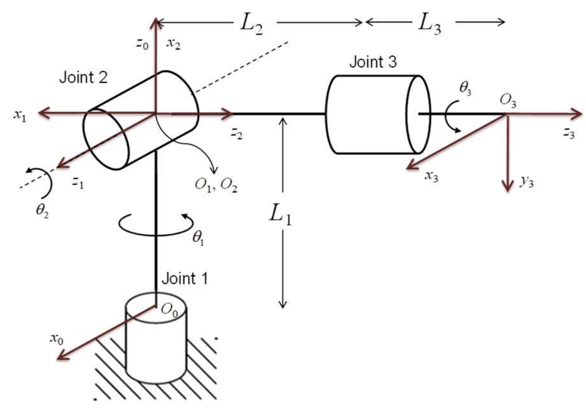
$$ {}^0T_4 = {}^0T_1\,{}^1T_2\,{}^2T_3\,{}^3T_4 = \begin{bmatrix} c_{12}c_4 + s_{12}s_4 & -c_{12}s_4 + s_{12}c_4 & 0 & a_1c_1 + a_2c_{12} \\ s_{12}c_4 - c_{12}s_4 & -s_{12}s_4 - c_{12}c_4 & 0 & a_1s_1 + a_2s_{12} \\ 0 & 0 & -1 & d_3 - d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Example: Spherical Manipulator

- Three orthogonal revolute joints, shown in its home configuration in the figures.
- The positive directions for the rotations of Joints 1, 2, and 3 are given as upward, out of the plane, and to the right, respectively, in the figures.
- Indeterminacies on: the global frame's origin O_0 and its x_0 axis; the local frame {3}; and the positive directions of x_1 and x_2 along their lines of axes (since z_0, z_1, and z_2 intersect). Here, they are given as in the figures, with two options for O_0 and x_0.



Option 1



Option 2

Option 1

- DH parameters table

i	θ_i	d_i	a_i	α_i	Joint variable q_i
1	$\theta_1 = -\pi/2 + q_1$	0	0	$-\pi/2$	q_1
2	$\theta_2 = -\pi/2 + q_2$	0	0	$\pi/2$	q_2
3	$\theta_3 = \pi/2 + q_3$	L_2+L_3	0	0	q_3

- Homogeneous transformation matrices ($c_1 = \cos\theta_1$, $s_1 = \sin\theta_1$, etc.)

$$ {}^0T_1 = \begin{bmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},\ {}^1T_2 = \begin{bmatrix} c_2 & 0 & s_2 & 0 \\ s_2 & 0 & -c_2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},\ {}^2T_3 = \begin{bmatrix} c_3 & -s_3 & 0 & 0 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & L_2+L_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

- Forward kinematics equation

$$ {}^0T_3 = \begin{bmatrix} c_1c_2c_3 - s_1s_3 & -c_1c_2s_3 - s_1c_3 & c_1s_2 & (L_2+L_3)c_1s_2 \\ s_1c_2c_3 + c_1s_3 & -s_1c_2s_3 + c_1c_3 & s_1s_2 & (L_2+L_3)s_1s_2 \\ -s_2c_3 & s_2s_3 & c_2 & (L_2+L_3)c_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

<u>Option 2</u>

- DH parameters table

i	θ_i	d_i	a_i	α_i	Joint variable q_i
1	$\theta_1 = -\pi/2 + q_1$	L_1	0	$-\pi/2$	q_1
2	$\theta_2 = -\pi/2 + q_2$	0	0	$\pi/2$	q_2
3	$\theta_3 = \pi/2 + q_3$	L_2+L_3	0	0	q_3

- Homogeneous transformation matrices

$$ {}^0T_1 = \begin{bmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix},\ {}^1T_2 = \begin{bmatrix} c_2 & 0 & s_2 & 0 \\ s_2 & 0 & -c_2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},\ {}^2T_3 = \begin{bmatrix} c_3 & -s_3 & 0 & 0 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & L_2+L_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

- Forward kinematics equation

$$ {}^0T_3 = \begin{bmatrix} c_1c_2c_3 - s_1s_3 & -c_1c_2s_3 - s_1c_3 & c_1s_2 & (L_2+L_3)c_1s_2 \\ s_1c_2c_3 + c_1s_3 & -s_1c_2s_3 + c_1c_3 & s_1s_2 & (L_2+L_3)s_1s_2 \\ -s_2c_3 & s_2s_3 & c_2 & (L_2+L_3)c_2 + L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

<u>Remarks</u>

- If $L_1 = L_2 = 0$, the first three frame origins intersect at a single point for both options in the figures, representing a spherical manipulator.
- As a consequence of the choice made for the coordinate frames, the block matrix ${}^0R_3$ that can be extracted from ${}^0T_3$ coincides with the rotation matrix of ZYZ Euler angles for $\theta_1, \theta_2, \theta_3$ with respect to the reference frame O_0-$x_0y_0z_0$.

CHAPTER 4. INVERSE MANIPULATOR KINEMATICS

Inverse kinematics vs. direct (or forward) kinematics

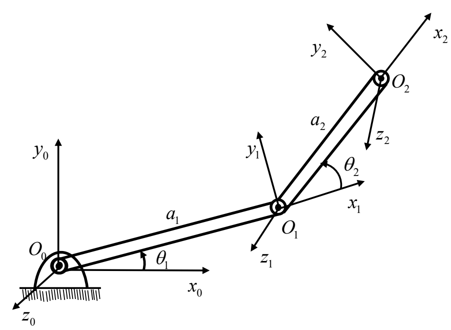
Problem \ Space	Joint Space		Cartesian Space
Forward Kinematics	$\mathbf{q}$ (known)	→	${}^{0}T_{n}(\mathbf{q})$ (unknown)
Inverse Kinematics	$\mathbf{q}$ (unknown)	←	${}^{0}T_{n}(\mathbf{q})$ (known)

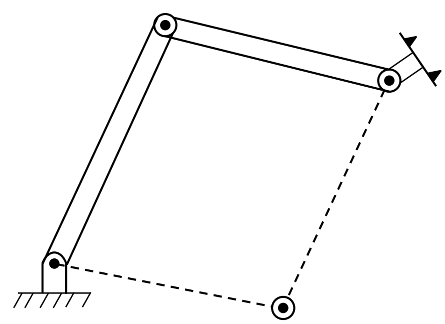
Solvability

- Given ${}^{0}T_{n}$ → find $q_1, q_2, \ldots, q_n$ (Cartesian space → joint space)
- $${}^{0}T_{n} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^{0}T_{1}(q_1)\,{}^{1}T_{2}(q_2)\ldots{}^{n-1}T_{n}(q_n) \quad \Rightarrow \quad q_i = f_i(r_{11}, \ldots, r_{33}, p_x, p_y, p_z),\ \ i = 1, \ldots, n$$
- Nonlinear transcendental equations

Existence and Uniqueness

- Workspace: volume of space which the manipulator's end-effector can reach
  - Dexterous workspace: end-effector can reach with all orientations. At each point, end-effector can be arbitrarily oriented.
  - Reachable workspace: end-effector can reach in at least one orientation.
  - {dexterous workspace} ⊂ {reachable workspace}





- Number of solutions depends on link parameters, joint limits, and number of joints.
  - If [DOF or number of unknowns] = [number of equations] → unique or <u>finite</u> number of solutions
  - If [DOF or number of unknowns] < [number of equations] → solution may not exist; manipulator cannot attain general goal positions and orientations in 3D space.
  - If [DOF or number of unknowns] > [number of equations] → infinite solutions may exist; kinematically redundant (flexible, dexterous, controllable); optimization is required.

  (Note: For complete position and orientation of the end-effector, the number of equations is 6. However, in general, the number of equations depends on a given task as well as the manipulator.)

Methods of Solution

- Solvable: all the sets of joint variables can be determined for a given position and orientation.
  - Closed form solutions – analytic expressions or polynomial of degree 4 or less
  - Numerical solutions (e.g., Bisection method, Newton-Raphson method, Secant method, Muller's method, Brent's algorithm, etc.)

- Closed form solution methods of kinematic equations
  1) Algebraic solution: specify end-effector frame relative to base frame $\rightarrow$ manipulate given equations
  2) Geometric solution: decompose spatial geometry of the manipulator into several plane geometry $\rightarrow$ use plane geometry to solve for joint angles

  (Note: Frequently, the mix of algebraic and geometric approaches is used.)

- A sufficient condition that a manipulator with 6 revolute joints will have a closed form solution is that three neighboring joint axes intersect at a point.

- Recall: Two-argument arctangent function $\phi = \text{atan2}(y, x)$

  Defined on all four quadrants ($-\pi \le \phi < \pi$)

Case	Quadrants	$\phi = \text{atan2}(y,x)$
$x > 0$	1, 4	$\phi = \arctan(y/x)$
$x = 0$	1, 4	$\phi = \underbrace{\text{sgn}(y)}_{=\pm 1}(\pi/2)$
$x < 0$	2, 3	$\phi = \arctan(y/x) + \text{sgn}(y)\cdot\pi$

<u>Algebraic Solution by Reduction to Polynomial</u>

- Let $\boxed{u = \tan\frac{\theta}{2}}$ and substitute $\cos\theta = \frac{1-u^2}{1+u^2}$, $\sin\theta = \frac{2u}{1+u^2}$ (Weierstrass Substitution)

  => Transcendental (e.g., trigonometric) equations in θ $\rightarrow$ polynomial equations in u

  (Note: polynomials up to degree 4 have closed form solutions.)

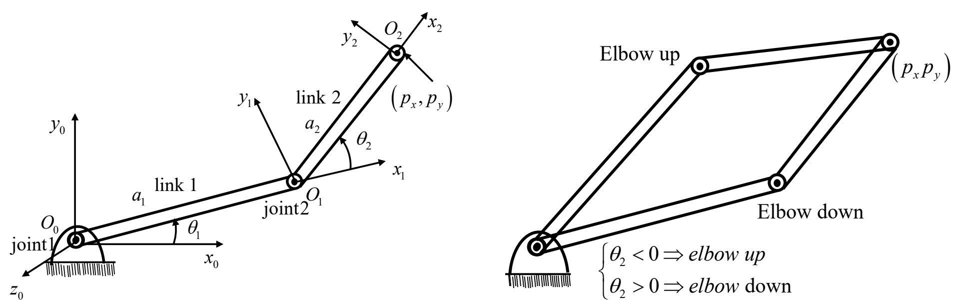
- Closed form solvable manipulators: Manipulators which are sufficiently simple to be solved by algebraic equations of up to degree 4.

<u>Repeatability and Accuracy</u>

- Taught point: point that the manipulator is moved to physically, and then the joint position sensors are read, and the joint angles are stored; teach and playback
- Repeatability of manipulator: specification of how precisely a manipulator can return to a taught point
- Computed point: point in a manipulator's workspace which was never taught; if a goal position and orientation are specified in Cartesian space, required joint variables must be solved for by computing inverse kinematics.
- Accuracy: Precision with which a computed point can be attained. Accuracy of a manipulator is bounded by the repeatability.

<u>Example: 2R Planar Manipulator</u>

Determine θ_1 and θ_2 in terms of p_x and p_y.



Kinematic equations

Homogeneous transformation between links: $^0T_2 = {}^0T_1\,{}^1T_2 = \begin{bmatrix} c_{12} & -s_{12} & 0 & a_1c_1 + a_2c_{12} \\ s_{12} & c_{12} & 0 & a_1s_1 + a_2s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

End-effector's position and orientation: $^RT_H = \begin{bmatrix} \cos\phi & -\sin\phi & 0 & p_x \\ \sin\phi & \cos\phi & 0 & p_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

1) Calculation of θ_2

$p_x = a_1\cos\theta_1 + a_2\cos(\theta_1+\theta_2) = a_1c_1 + a_2c_{12}$ and $p_y = a_1\sin\theta_1 + a_2\sin(\theta_1+\theta_2) = a_1s_1 + a_2s_{12}$

$\Rightarrow\ p_x^2 + p_y^2 = a_1^2 + a_2^2 + 2a_1a_2(c_1c_{12} + s_1s_{12}) = a_1^2 + a_2^2 + 2a_1a_2c_2$ (also from law of cosines)

$\Rightarrow\ c_2 = \cos\theta_2 = \dfrac{p_x^2 + p_y^2 - a_1^2 - a_2^2}{2a_1a_2}$ → The solution exists only if $-1 \le \dfrac{p_x^2 + p_y^2 - a_1^2 - a_2^2}{2a_1a_2} \le 1$. If not, the target point is outside of the reachable workspace.

$s_2 = \sin\theta_2 = \pm\sqrt{1-c_2^2}$; $\theta_2 = \mathrm{atan2}(s_2, c_2)$ → redundancy – elbow-up vs. elbow-down

2) Calculation of θ_1

$p_x = a_1\cos\theta_1 + a_2\cos(\theta_1+\theta_2) = a_1c_1 + a_2c_{12} = a_1c_1 + a_2(c_1c_2 - s_1s_2) = (a_1 + a_2c_2)c_1 - a_2s_2s_1$

$p_y = a_1\sin\theta_1 + a_2\sin(\theta_1+\theta_2) = a_1s_1 + a_2s_{12} = a_1c_1 + a_2(c_1s_2 + s_1c_2) = (a_1 + a_2c_2)s_1 + a_2s_2c_1$

Method 1 for θ_1:

$(a_1 + a_2c_2)c_1 - (a_2s_2)s_1 = p_x$ and $(a_2s_2)c_1 + (a_1 + a_2c_2)s_1 = p_y$

Cramer's formula $\Rightarrow\ c_1 = \dfrac{(a_1 + a_2c_2)p_x + a_2s_2p_y}{(a_1 + a_2c_2)^2 + (a_2s_2)^2}$ and $s_1 = \dfrac{-a_2s_2p_x + (a_1 + a_2c_2)p_y}{(a_1 + a_2c_2)^2 + (a_2s_2)^2}$

$\theta_1 = \mathrm{atan2}(s_1, c_1) = \mathrm{atan2}(-a_2s_2p_x + (a_1 + a_2c_2)p_y, (a_1 + a_2c_2)p_x + a_2s_2p_y)$

Method 2 for θ_1:

$$\frac{p_y}{p_x} = \frac{(a_1 + a_2 c_2) s_1 + a_2 s_2 c_1}{(a_1 + a_2 c_2) c_1 - a_2 s_2 s_1} = \frac{\dfrac{s_1}{c_1} + \dfrac{a_2 s_2}{a_1 + a_2 c_2}}{1 - \dfrac{s_1}{c_1} \dfrac{a_2 s_2}{a_1 + a_2 c_2}}$$

Let $\tan\gamma = \dfrac{a_2 s_2}{a_1 + a_2 c_2}$ or $\gamma = \text{atan2}(a_2 s_2, a_1 + a_2 c_2)$

$$\frac{p_y}{p_x} = \frac{\tan\theta_1 + \tan\gamma}{1 - \tan\theta_1 \tan\gamma} = \tan(\theta_1 + \gamma) \;\Rightarrow\; \theta_1 = \text{atan2}(p_y, p_x) - \gamma$$

$$\Rightarrow\; \theta_1 = \text{atan2}(p_y, p_x) - \text{atan2}(a_2 s_2, a_1 + a_2 c_2)$$