Jacobian, Singularities, and Velocity Kinematics

Chapter 4

Jacobian

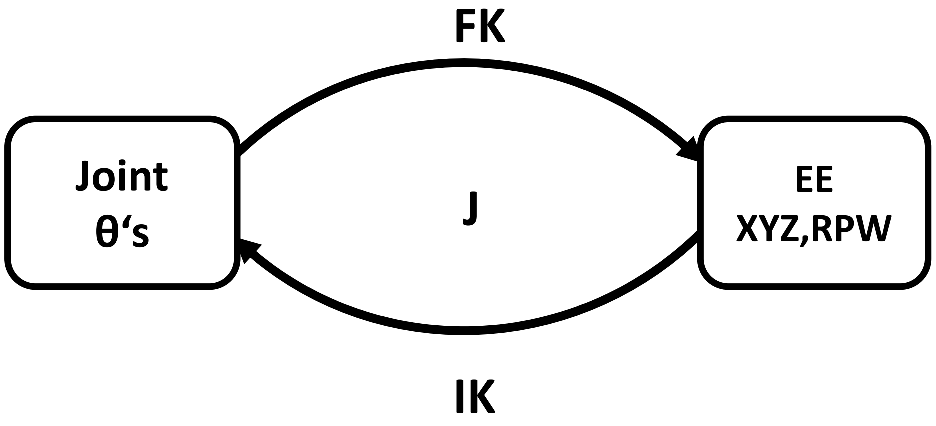
- Derivative of position with respect to joint variables

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \cdots & \frac{\partial f_1}{\partial q_{\mathrm{n}}} \\ \vdots & & \vdots \\ \frac{\partial f_m}{\partial q_1} & \cdots & \frac{\partial f_m}{\partial q_n} \end{bmatrix}$$

f: x, y or z equation
q: joint variable (d or θ)
∂: partial derivative
n: number of joints
m: number of equations

Jacobian

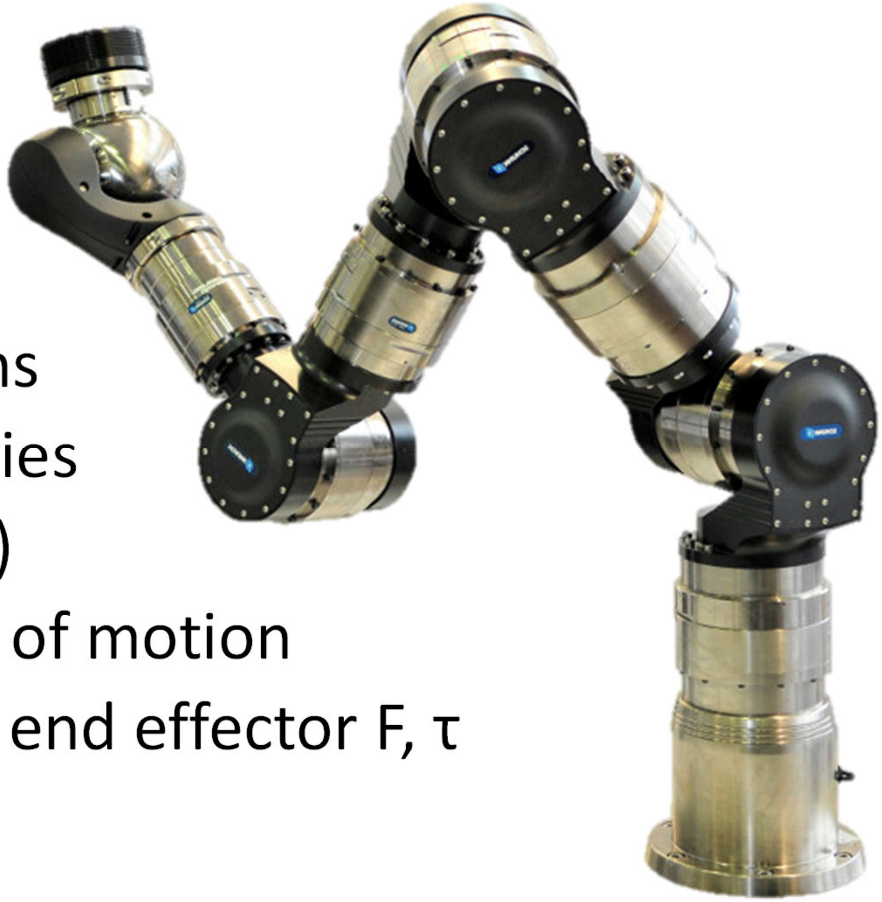
- For FK, IK: relate tip pose to joint positions



- For ∂K: relate task space to joint space velocities

$$J\dot{q} = \xi \;\; or \;\; J\dot{\theta} = \dot{X}$$

Jacobian



- Uses in robotics
  - Find singular configurations
  - Plan and execute trajectories
  - Coordinate motion (p, v, a)
  - Derive dynamic equations of motion
  - Find required joint τ given end effector F, τ
  - Resolve redundancies

Differential Kinematics

$$\xi = J\dot{q} \;\; or \;\; \dot{X} = J\dot{\theta}$$

J: 6xn matrix of derivatives

$\dot{\theta}, \dot{q}$: nx1 vector of joint velocities

$\dot{X}, \xi$: 6x1 vector of v, ω for tip

n: number of joints

$$\underbrace{\begin{bmatrix} v_n^0 \\ \omega_n^0 \end{bmatrix}}_{\xi} = \underbrace{\begin{bmatrix} J_v \\ J_\omega \end{bmatrix}}_{J} \underbrace{\begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{bmatrix}}_{\dot{\theta}, \dot{d}}$$

Differential Kinematics

$$J\dot{q} = \xi \;\; or \;\; J\dot{\theta} = \dot{X}$$

J: 6xn matrix of derivatives

$\dot{\theta}, \dot{q}$: nx1 vector of joint velocities

$\dot{X}, \xi$: 6x1 vector of v, ω for tip

n: number of joints

$$\underbrace{\begin{bmatrix} v_{nx}^{0} \\ v_{ny}^{0} \\ v_{nz}^{0} \\ \omega_{nx}^{0} \\ \omega_{ny}^{0} \\ \omega_{nz}^{0} \end{bmatrix}}_{\xi} = \underbrace{\begin{bmatrix} \frac{\partial f_x}{\partial \theta_1} & \cdots & \frac{\partial f_x}{\partial \theta_{\mathrm{n}}} \\ \frac{\partial f_y}{\partial \theta_1} & & \frac{\partial f_y}{\partial \theta_{\mathrm{n}}} \\ \frac{\partial f_z}{\partial \theta_1} & & \frac{\partial f_z}{\partial \theta_{\mathrm{n}}} \\ \frac{\partial f_\phi}{\partial \theta_1} & & \frac{\partial f_\phi}{\partial \theta_{\mathrm{n}}} \\ \frac{\partial f_\varphi}{\partial \theta_1} & & \frac{\partial f_\varphi}{\partial \theta_{\mathrm{n}}} \\ \frac{\partial f_\psi}{\partial \theta_1} & \cdots & \frac{\partial f_\psi}{\partial \theta_1} \end{bmatrix}}_{J} \underbrace{\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_{n-1} \\ \dot{q}_n \end{bmatrix}}_{\dot{\theta}, \dot{d}}$$

Ways to find Jacobian

- For an n-link robot with joint variables $q_1 \ldots q_n$, With forward kinematics:

$$T_n^0 = \begin{bmatrix} R_n^0 & o_n^0 \\ 0 & 1 \end{bmatrix}$$

Find the Jacobian using either:

1) Basic Method (simple robots *n* $\leq$ 3)
2) Formula Method (complex robots *n* > 3)

Types of Jacobians

- Geometric/standard: typical (one just learned), gives body linear and angular velocity in world frame J
- Body: linear and angular velocity of tip relative to world frame, expressed in tip frame J^B
- Spatial: linear velocity of a point passing through the world frame origin J^S
- Analytical: used to find end effector velocity and position using Euler angles J_A

Analytical Jacobian

- Used to find end effector velocity and position using ZYZ Euler angles

If $X = \begin{bmatrix} x(q) \\ y(q) \\ z(q) \\ \phi(q) \\ \theta(q) \\ \psi(q) \end{bmatrix}$ Then $B = \begin{bmatrix} c_\psi s_\theta & -s_\psi & 0 \\ s_\psi s_\theta & c_\psi & 0 \\ c_\theta & 0 & 1 \end{bmatrix}$

And $J_A(q) = \begin{bmatrix} I & 0 \\ 0 & B^{-1} \end{bmatrix} J(q)$

Adjoint Transformation

- Represent twist in new frame
- Comes from similarity transform

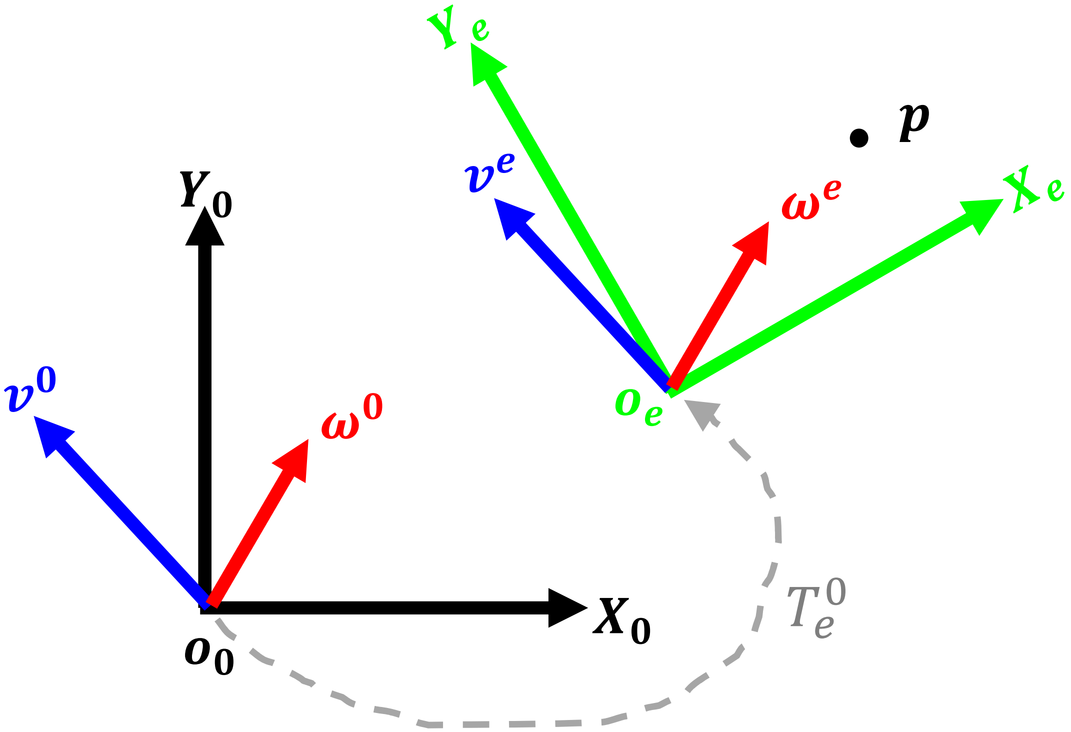
$$Ad_T = \begin{bmatrix} R & \hat{p}R \\ 0 & R \end{bmatrix}$$

$$(Ad_T)^{-1} = Ad_{T^{-1}}$$

$$\xi^0 = Ad_{T_e^0}\xi^e$$

$$\begin{bmatrix} v^0 \\ \omega^0 \end{bmatrix} = Ad_{T_e^0}\begin{bmatrix} v^e \\ \omega^e \end{bmatrix}$$

$$p^0 = T_e^0 p^e$$

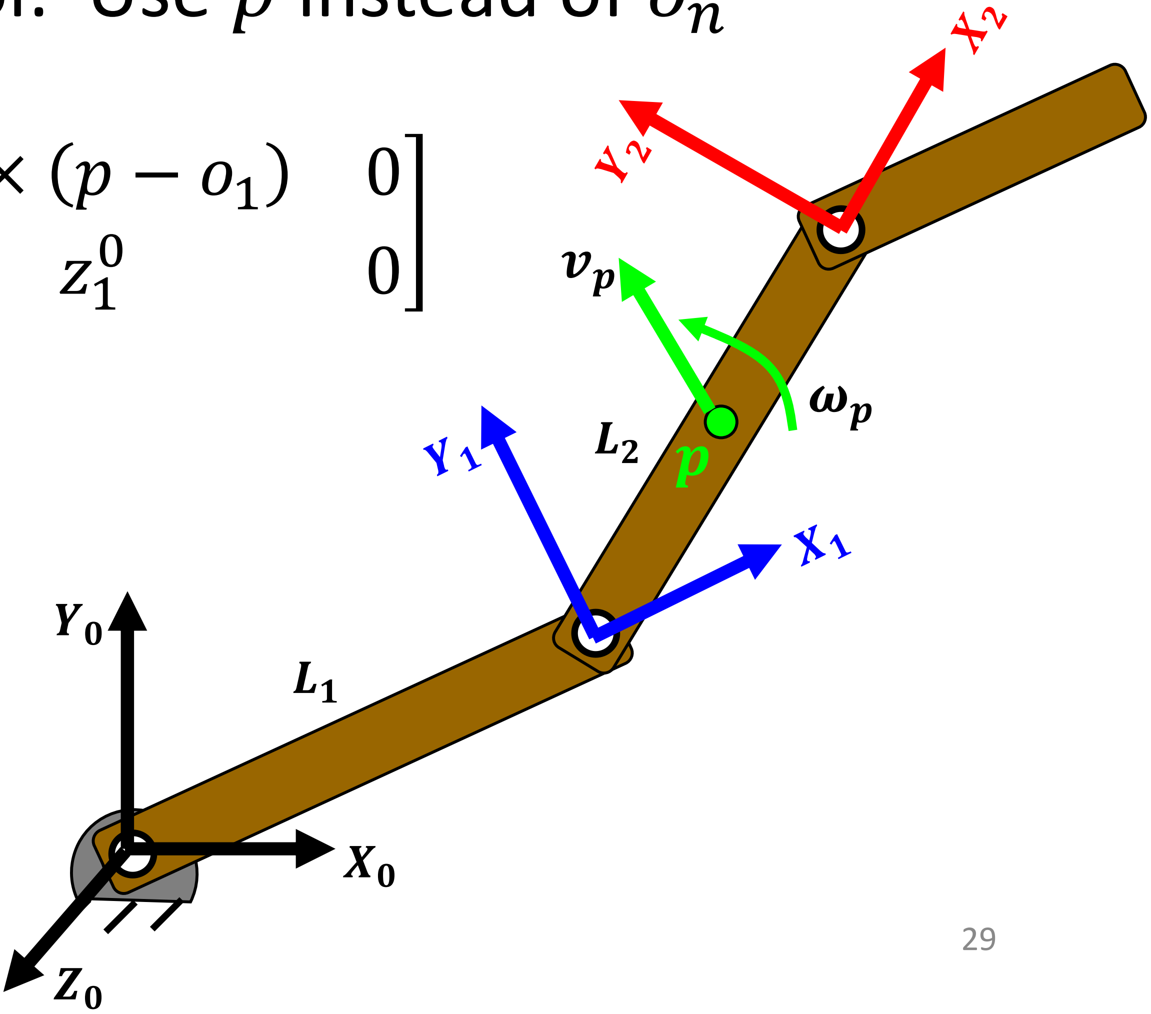


Point Velocity

- Use the Jacobian to calculate the velocity of any point on a manipulator. Use p instead of o_n

$$J(q) = \begin{bmatrix} z_0^0 \times (p - o_0) & z_1^0 \times (p - o_1) & 0 \\ z_0^0 & z_1^0 & 0 \end{bmatrix}$$

- This is like placing the tool frame at point *p*



Velocity and Acceleration Analysis

- To get joint velocities from end effector velocity:

$$\dot{X} = J\dot{q}$$

$$\rightarrow \ \dot{q} = \mathrm{J}^{-1}\dot{X}$$

```
In Matlab:
Jinv = inv(J) or J\
```

Velocity and Acceleration Analysis

- To get joint velocities from end effector velocity:

$$\dot{X} = J\dot{q}$$
$$\rightarrow \ \dot{q} = \mathrm{J}^{-1}\dot{X}$$

```
In Matlab:
Jinv = inv(J) or J\
```

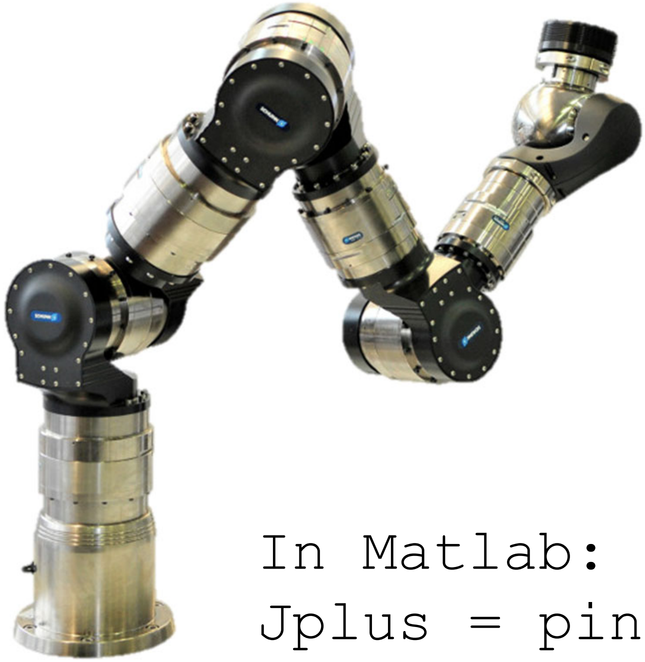
- To get acceleration:

$$\ddot{X} = \dot{J}\dot{q} + J\ddot{q}$$
$$\rightarrow \ddot{q} = J^{-1}(\ddot{X} - \dot{J}\dot{q})$$

```
In Matlab:
Jdot =(J(q(i))-J(q(i-1)))/dt
```

Redundant Robots

- Redundant robots have more DoF than needed
  - 2D: n > 3, 3D: n > 6
- Jacobian is not square
  - J is 6xn, n > 6
- Use pseudoinverse



$$(JJ^T)(JJ^T)^{-1} = I_{6x6} = J[J^+]$$

where J^+ is nx6

In Matlab:

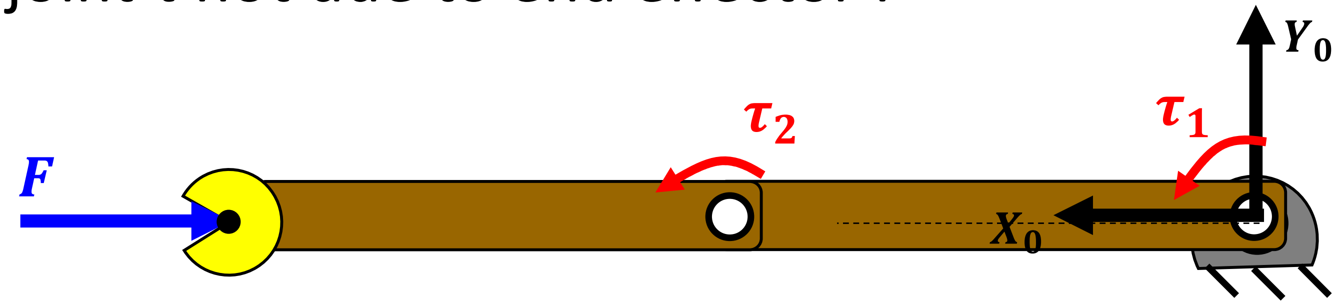
`Jplus = pinv(J)`

$$\dot{q} = J^+\dot{X} + (I - J^+J)b$$

Basis for null space of J (internal DoF)
b helps guide $\dot{q}$, if special joint behavior desired
For min $\sum \dot{q}^2$, set b = 0

Fundamental Theorem of Lin Alg

- For an mxn matrix J,
  - $Range(J) + Null(J^T) \in \mathbb{R}^m$
  - $Range(J^T) + Null(J) \in \mathbb{R}^n$
- Which implies:
  - $Range(J)$: achievable velocities
  - $Null(J^T)$: non-achievable velocities
  - $Range(J^T)$: joint τ due to end effector F
  - $Null(J)$: joint τ not due to end effector F



Manipulability

- A measure of closeness to singularity
  - Generally: $\mu = \det(J)$
  - Redundants: $\mu = \sqrt{\det(JJ^T)}$

- Manipulability ellipsoid: for J of full rank m, the m-dim ellipsoid of EE velocities.

$$1 \leq \dot{X}^T (JJ^T) \dot{X}$$