Robotics: Mechanics & Control

Chapter 4: Differential Kinematics

In this chapter we review the Jacobian analysis for serial robots. First the definition to angular and linear velocities are given, then the Jacobian matrix is defined in conventional and screw-based representation, while their general and iterative derivation methods are given. Next the static wrench and its relation to Jacobian transpose is introduced, and Jacobian characteristics such as singularity, isotropy, dexterity and manipulability are elaborated. Inverse Jacobian solution for fully-, under- and redundantly-actuator robots are formulated, and redundancy resolution schemes are detailed. Finally, Stiffness analysis of robotic manipulators is reviewed in detail.

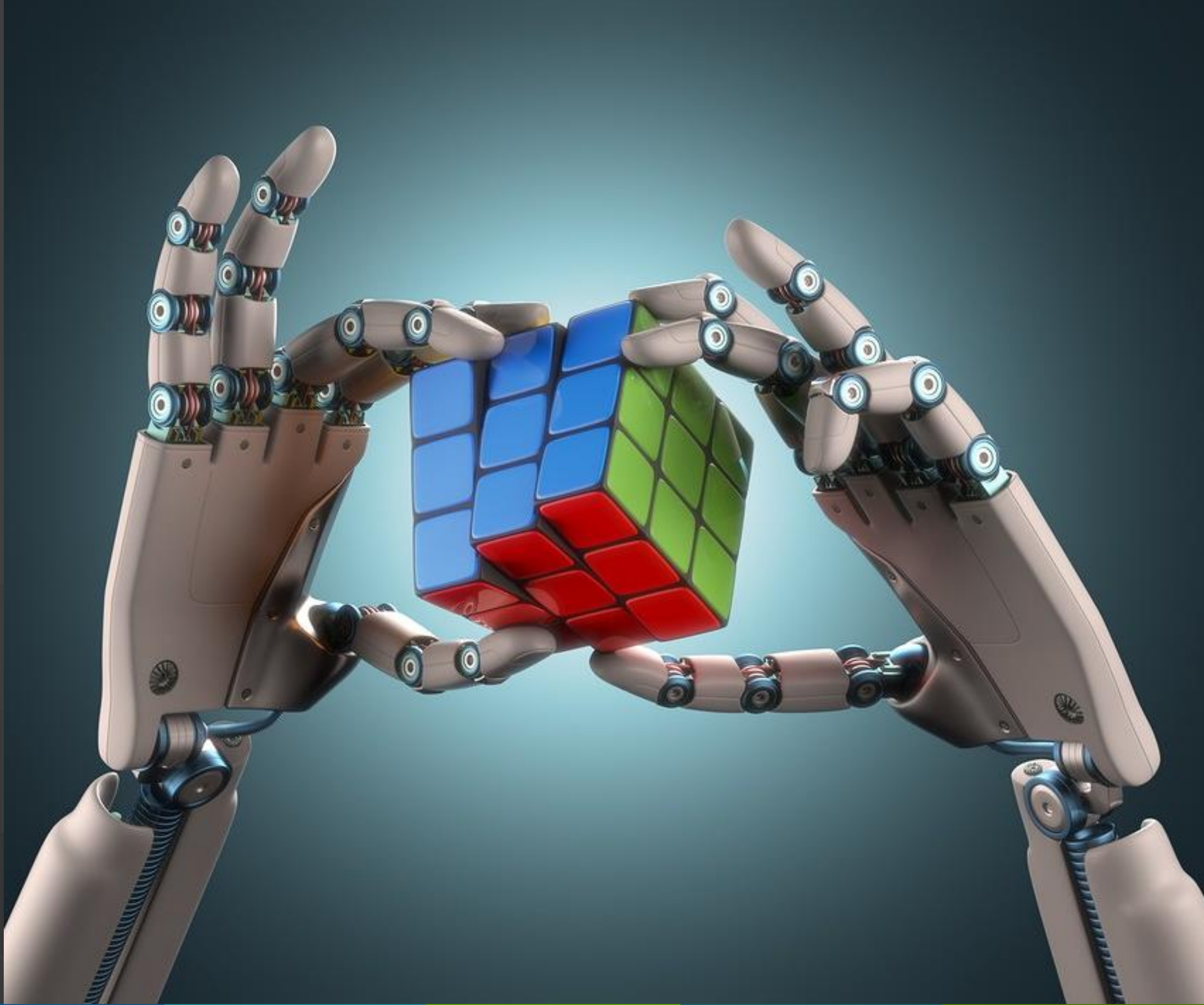
K. N. Toosi University of Technology, Faculty of Electrical Engineering,
Department of Systems and Control, Advanced Robotics and Automated Systems

Welcome

To Your Prospect Skills

On Robotics :

Mechanics and Control



01

About ARAS

ARAS Research group originated in 1997 and is proud of its 22+ years of brilliant background, and its contributions to the advancement of academic education and research in the field of Dynamical System Analysis and Control in the robotics application. ***ARAS*** are well represented by the industrial engineers, researchers, and scientific figures graduated from this group, and numerous industrial and R&D projects being conducted in this group. The main asset of our research group is its human resources devoted all their time and effort to the advancement of science and technology. One of our main objectives is to use these potentials to extend our educational and industrial collaborations at both national and international levels. In order to accomplish that, our mission is to enhance the breadth and enrich the quality of our education and research in a dynamic environment.

Contents

In this chapter we review the Jacobian analysis for serial robots. First the definition to angular and linear velocities are given, then the Jacobian matrix is defined in conventional and screw-based representation, while their general and iterative derivation methods are given. Next the static wrench and its relation to Jacobian transpose is introduced, and Jacobian characteristics such as singularity, isotropy, dexterity and manipulability are elaborated. Inverse Jacobian solution for fully-, under- and redundantly-actuator robots are formulated, and redundancy resolution schemes are detailed. Finally, Stiffness analysis of robotic manipulators is reviewed in detail.

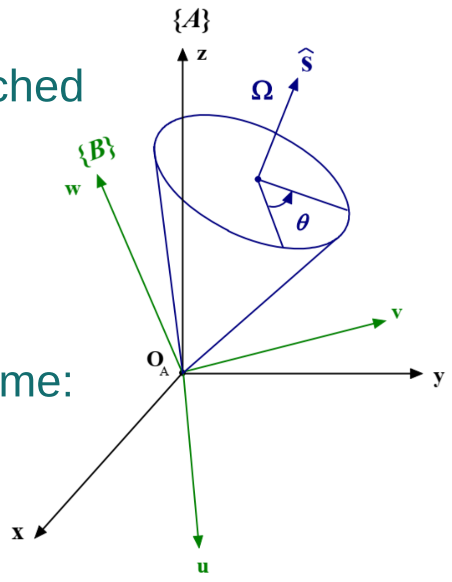
Introduction

- Preliminaries
  - ✓ Angular Velocity of a Rigid Body
    - Attribute of the whole rigid body
    - The rate of instantaneous rotation of frame $\{B\}$ attached to the rigid body with respect to a fixed frame $\{B\}$.

      A vector denoted by $\mathbf{\Omega}$ along the screw axis

      With the value equal to the rate of rotation $\dot{\theta}$.

$$\mathbf{\Omega} \doteq \dot{\theta}\,\hat{\mathbf{s}}.$$



- Angular velocity vector can be expressed in any frame:

$$\begin{aligned} {}^{A}\mathbf{\Omega} &= \Omega_x\hat{\mathbf{x}} + \Omega_y\hat{\mathbf{y}} + \Omega_z\hat{\mathbf{z}} \\ &= \dot{\theta}\,(s_x\hat{\mathbf{x}} + s_y\hat{\mathbf{y}} + s_z\hat{\mathbf{z}}) \end{aligned}$$

In which, Ω_x, Ω_y, Ω_z are the components of this vector.

Preliminaries

✓ Angular Velocity & Rotation Matrix Rate

- Angular velocity is defined based-on screw representation
- What is its relation to the rotation matrix representation?

Note that

$$ {}^{A}\boldsymbol{R}_B \, {}^{A}\boldsymbol{R}_B^T = \boldsymbol{I}, $$

Differentiate both side with respect to time

$$ {}^{A}\dot{\boldsymbol{R}}_B \, {}^{A}\boldsymbol{R}_B^T + {}^{A}\boldsymbol{R}_B \, {}^{A}\dot{\boldsymbol{R}}_B^T = \boldsymbol{0}. $$

Substitute: ${}^{A}\boldsymbol{R}_B^T = {}^{A}\boldsymbol{R}_B^{-1}$ and ${}^{A}\boldsymbol{R}_B = \left({}^{A}\boldsymbol{R}_B^{-1}\right)^T$

$$ \left({}^{A}\dot{\boldsymbol{R}}_B \, {}^{A}\boldsymbol{R}_B^{-1}\right) + \left({}^{A}\dot{\boldsymbol{R}}_B \, {}^{A}\boldsymbol{R}_B^{-1}\right)^T = \boldsymbol{0}. $$

This means that ${}^{A}\dot{\boldsymbol{R}}_B \, {}^{A}\boldsymbol{R}_B^{-1}$ is a 3×3 skew symmetric matrix $\boldsymbol{\Omega}^{\times}$:

$$ \boldsymbol{\Omega}^{\times} \equiv {}^{A}\dot{\boldsymbol{R}}_B \, {}^{A}\boldsymbol{R}_B^{-1} = \begin{bmatrix} 0 & -\Omega_z & \Omega_y \\ \Omega_z & 0 & -\Omega_x \\ -\Omega_y & \Omega_x & 0 \end{bmatrix}. $$

It can be shown that the three parameters $\Omega_x, \Omega_y, \Omega_z$ are the components of angular velocity vector.

Preliminaries

✓ Angular Velocity & Euler Angles Rate

- Angular velocity is a vector but Euler angels are not.
- Angular velocity is not equal to the rate of Euler Angels

$$\boldsymbol{\Omega} \neq \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} \qquad \text{But} \qquad \boldsymbol{\Omega} = \boldsymbol{E}(\alpha,\beta,\gamma) \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix}$$

Use

$$\boldsymbol{\Omega}^{\times} \equiv {}^{A}\dot{\boldsymbol{R}}_{B}\, {}^{A}\boldsymbol{R}_{B}^{-1}$$

Or equivalently

$$\Omega_x = \dot{r}_{31}r_{21} + \dot{r}_{32}r_{22} + \dot{r}_{33}r_{23},$$
$$\Omega_y = \dot{r}_{11}r_{31} + \dot{r}_{12}r_{32} + \dot{r}_{13}r_{33},$$
$$\Omega_z = \dot{r}_{21}r_{11} + \dot{r}_{22}r_{12} + \dot{r}_{23}r_{13},$$

To derive $E(\alpha,\beta,\gamma)$. For example for $w-v-w$ Euler angles we have:

$$\boldsymbol{R}_{wvw}(\alpha,\beta,\gamma) = \boldsymbol{R}_{w}(\alpha)\boldsymbol{R}_{v}(\beta)\boldsymbol{R}_{w}(\gamma)$$

$$= \begin{bmatrix} c\alpha c\beta c\gamma - s\alpha s\gamma & -c\alpha c\beta s\gamma - s\alpha c\gamma & c\alpha s\beta \\ s\alpha c\beta c\gamma + c\alpha s\gamma & -s\alpha c\beta s\gamma + c\alpha c\gamma & s\alpha s\beta \\ -s\beta c\gamma & s\beta s\gamma & c\beta \end{bmatrix} \quad \text{and} \quad \boldsymbol{E}_{wvw} = \begin{bmatrix} 0 & -s\alpha & c\alpha s\beta \\ 0 & c\alpha & s\alpha s\beta \\ 1 & 0 & c\beta \end{bmatrix}.$$

Preliminaries

✓ Linear Velocity of a Point

- Linear velocity of a point $\boldsymbol{P}$ is the time derivative of the position vector $\boldsymbol{p}$ with respect to a fixed frame.

$$v_p = \dot{\boldsymbol{p}} = \left(\frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t}\right)_{fix}$$

- Relative velocity with respect to a moving frame is denoted by

$$v_{rel} = \left(\frac{\partial \boldsymbol{p}}{\partial t}\right)_{mov}$$

In which the partial derivative notation is used to denote relativeness

- Golden Rule

$$\left(\frac{\mathrm{d}(\cdot)}{\mathrm{d}t}\right)_{fix} = \left(\frac{\partial(\cdot)}{\partial t}\right)_{mov} + \boldsymbol{\Omega} \times (\cdot), \quad \text{OR} \quad \left(\frac{\mathrm{d}(\cdot)}{\mathrm{d}t}\right)_{fix} = \left(\frac{\partial(\cdot)}{\partial t}\right)_{mov} + \boldsymbol{\Omega}^{\times}(\cdot)$$

In which $\boldsymbol{\Omega}$ denotes the angular velocity of the moving frame with respect to the fixed frame, and $\boldsymbol{\Omega}^{\times}$ denotes its skew-symmetric matrix representation

Preliminaries

✓ Linear Velocity of a Point

- Verify the derivative of the rotation matrix

$$\left(\frac{\mathrm{d}({}^{A}\boldsymbol{R}_{B})}{\mathrm{d}t}\right)_{fix} = \left(\frac{\partial({}^{A}\boldsymbol{R}_{B})}{\partial t}\right)_{mov} + \boldsymbol{\Omega}^{\times}({}^{A}\boldsymbol{R}_{B}),$$

While

$$\left(\frac{\partial({}^{A}\boldsymbol{R}_{B})}{\partial t}\right)_{mov} = 0.$$

Hence,

$${}^{A}\dot{\boldsymbol{R}}_{B} = \boldsymbol{\Omega}^{\times}\ {}^{A}\boldsymbol{R}_{B}.$$

This verifies the relation of angular velocity vector with the rate of rotation matrix

$$\boldsymbol{\Omega}^{\times} \equiv {}^{A}\dot{\boldsymbol{R}}_{B}\ {}^{A}\boldsymbol{R}_{B}^{-1}$$

Preliminaries

✓ Linear Velocity of a Point

- Consider the position vector $\boldsymbol{P}$

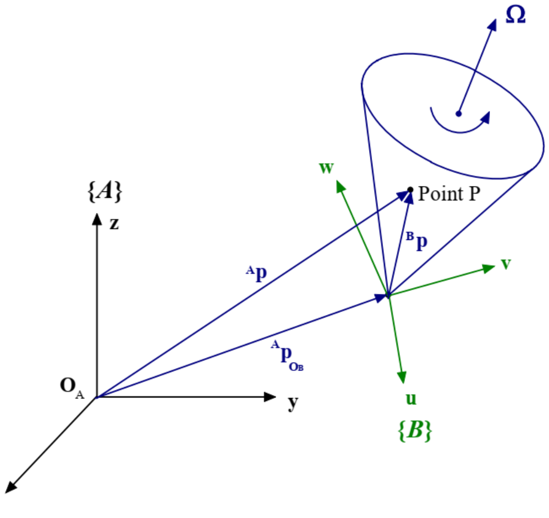
$$^{A}\boldsymbol{P} = {}^{A}\boldsymbol{P}_{O_B} + {}^{A}\boldsymbol{R}_B\,{}^{B}\boldsymbol{P},$$

Differentiate with respect to time

$$^{A}\dot{\boldsymbol{P}} = {}^{A}\dot{\boldsymbol{P}}_{O_B} + {}^{A}\dot{\boldsymbol{R}}_B\,{}^{B}\boldsymbol{P} + {}^{A}\boldsymbol{R}_B\,{}^{B}\dot{\boldsymbol{P}}$$

$$^{A}\boldsymbol{v}_p = {}^{A}\boldsymbol{v}_{O_B} + {}^{A}\dot{\boldsymbol{R}}_B\,{}^{B}\boldsymbol{P} + {}^{A}\boldsymbol{R}_B\,{}^{B}\boldsymbol{v}_p,$$

where ${}^{B}\boldsymbol{v}_p = v_{rel}$



The time derivative of rotation matrix is given

$$^{A}\dot{\boldsymbol{R}}_B = {}^{A}\boldsymbol{\Omega}^{\times}\,{}^{A}\boldsymbol{R}_B.$$

Hence,

$$^{A}\boldsymbol{v}_p = {}^{A}\boldsymbol{v}_{O_B} + {}^{A}\boldsymbol{R}_B\,{}^{B}\boldsymbol{v}_p + {}^{A}\boldsymbol{\Omega}^{\times}\,{}^{A}\boldsymbol{R}_B\,{}^{B}\boldsymbol{P}.$$

If $\boldsymbol{P}$ is embedded in the rigid body, the relative velocity is ${}^{B}\boldsymbol{v}_p$ zero. Then

$$^{A}\boldsymbol{v}_p = {}^{A}\boldsymbol{v}_{O_B} + {}^{A}\boldsymbol{\Omega}^{\times}\,{}^{A}\boldsymbol{R}_B\,{}^{B}\boldsymbol{P}.$$

Preliminaries

- Twist: Screw Coordinates
  - ✓ General Motion: Screw Representation
    - General Motion =

      Rotation about $\hat{\boldsymbol{s}}$ + Translation along $\hat{\boldsymbol{s}}$

      $$\{\hat{\boldsymbol{s}}, \theta\} \quad + \quad \{\boldsymbol{s_0}, d\}$$

      Assume the ratio of d to θ is denoted by pitch λ

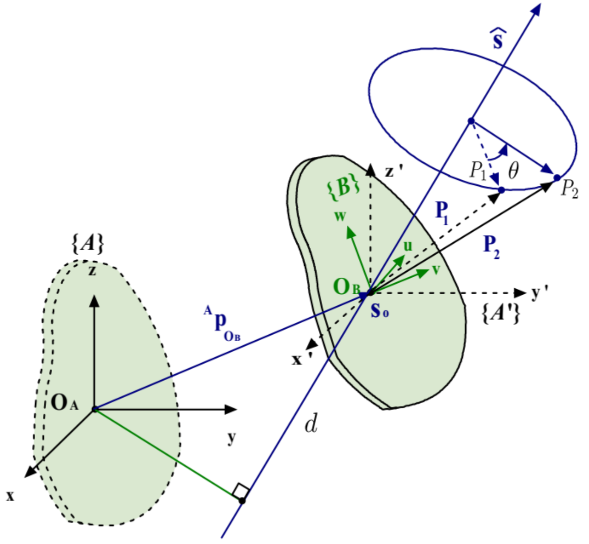
      $$\lambda = \frac{d}{\theta} \text{ or } \lambda = \frac{\dot{d}}{\dot{\theta}} \text{ in } (m/rad) \text{ unit}$$

    - Define Screw Coordinate (6×1)

      Unit Screw coordinate $\hat{\$}$ by pair of two vectors:

      $$\hat{\$} = \begin{bmatrix} \hat{s} \\ s_o \times \hat{s} + \lambda \hat{s} \end{bmatrix} = \begin{bmatrix} \$_1 \\ \$_2 \\ \$_3 \\ \$_4 \\ \$_5 \\ \$_6 \end{bmatrix}$$

      In which s_0 could be selected on any arbitrary point on the axis $\hat{\boldsymbol{s}}$.



Preliminaries

- Twist: Screw Coordinates
  - ✓ General motion of a point $\boldsymbol{P}$ on the rigid body
    - Twist: A (6×1) Tuple

$$\text{Twist} = \begin{bmatrix} \text{Angular velocity of the rigid body} \\ \text{Linear velocity of the point } \boldsymbol{P} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Omega} \\ \dot{\boldsymbol{P}} \end{bmatrix}$$

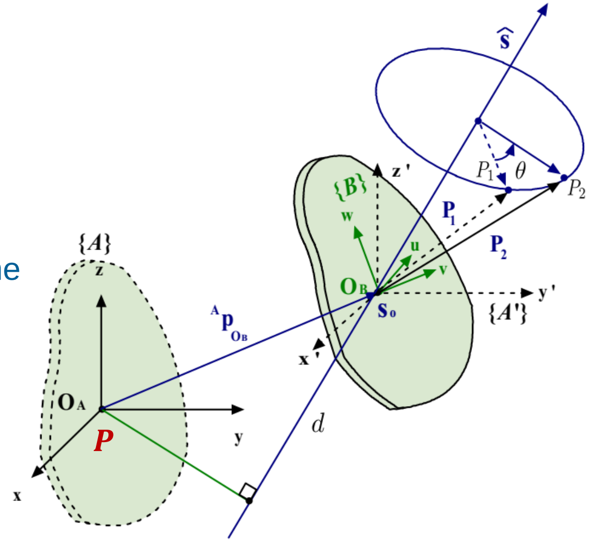
To find the screw for point $\boldsymbol{P}$, attach an instantaneous fixed frame On point P aligned with the reference frame {0} then

Twist: $\$ = \dot{q}\,\hat{\$}$

In which, the first vector reads: $\hat{\mathbf{s}}\,\dot{\theta} = {}^{A}\boldsymbol{\Omega}.$

and the second vector is:

$$\begin{aligned}(\boldsymbol{s}_o \times \hat{\mathbf{s}} + \lambda\hat{\mathbf{s}})\dot{\theta} &= \boldsymbol{s}_o \times \dot{\theta}\hat{\mathbf{s}} + \lambda\dot{\theta}\hat{\mathbf{s}} \\ &= \boldsymbol{s}_o \times \boldsymbol{\Omega} + \lambda\dot{\theta}\hat{\mathbf{s}} \\ &= \boldsymbol{\Omega} \times (-\boldsymbol{s}_o) + \dot{d}\hat{\mathbf{s}} \\ &= \boldsymbol{\Omega} \times {}^{B}\boldsymbol{P}_{O_A} + \dot{d}\hat{\mathbf{s}}.\end{aligned}$$



This gives the linear velocity of the interested embedded point $\boldsymbol{P}$ on the rigid body

Preliminaries

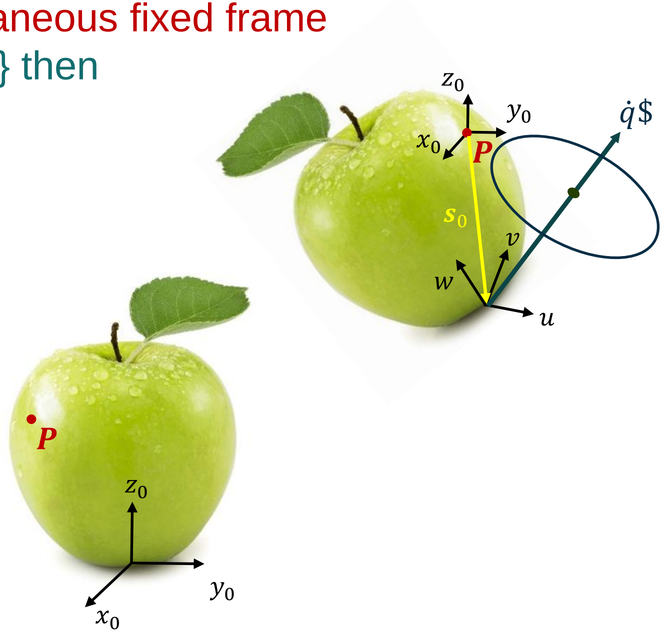
- Twist: Screw Coordinates
  - ✓ General motion of a point ***P*** on the rigid body
    - To find the screw for point ***P***, attach an instantaneous fixed frame On point ***P*** aligned with the reference frame {0} then

$$\text{Twist} = \begin{bmatrix} \text{Angular velocity of the rigid body} \\ \text{Linear velocity of the point } \boldsymbol{P} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Omega} \\ \dot{\boldsymbol{P}} \end{bmatrix}$$

Both vectors with respect to the fixed frame {0}

- Screw coordinate

Twist: $$\$ = \dot{q}\,\hat{\$} = \dot{q} \begin{bmatrix} \hat{\boldsymbol{s}} \\ \boldsymbol{s}_o \times \hat{\boldsymbol{s}} + \lambda \hat{\boldsymbol{s}} \end{bmatrix}$$



Preliminaries

- Twist: Screw Representation
  - ✓ Twist for Revolute joint (R)
    - For pure rotational joint $\lambda = 0$ and $\dot{q} = \dot{\theta}$

      The twist is represented by

$$\$ = \begin{bmatrix} \hat{\mathbf{s}} \\ \mathbf{s}_o \times \hat{\mathbf{s}} \end{bmatrix} \dot{\theta}.$$

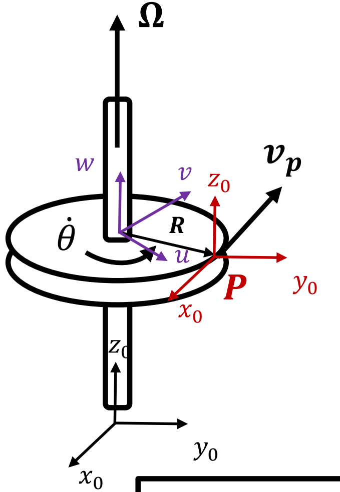
      where, instantaneous frame {0} is attached on point ***P***
  - ✓ Twist for Prismatic joint (P)
    - For pure translational joint $\lambda = \infty$ and $\dot{q} = \dot{d}$

      The twist is represented by

$$\$ = \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{s}} \end{bmatrix} \dot{d}.$$

    - Since we use the primary joint in serial manipulators these two screw representations are used in the differential kinematics.

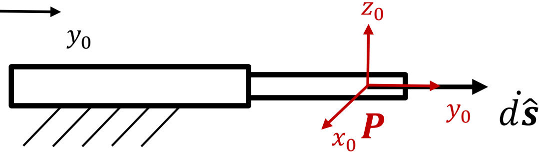
Consider the angular and linear velocity of a point ***P*** in a circular disk with rotary joint



$$\$ = \begin{bmatrix} \hat{\mathbf{s}} \\ \mathbf{s}_0 \times \hat{\mathbf{s}} \end{bmatrix} \dot{\theta} = \begin{bmatrix} \boldsymbol{\Omega} \\ \boldsymbol{\Omega} \times (-\mathbf{s}_0) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Omega} \\ \boldsymbol{\Omega} \times \boldsymbol{R} \end{bmatrix}$$

$$\boldsymbol{\omega_{disk}} = \boldsymbol{\Omega}$$

$$\boldsymbol{v_p} = \boldsymbol{\Omega} \times \boldsymbol{R}$$



Consider a point ***P*** on a moving piston:

$$\$ = \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{s}} \end{bmatrix} \dot{d} \rightarrow \begin{matrix} \boldsymbol{\omega_p} = \mathbf{0} \\ \boldsymbol{v_p} = \dot{d}\hat{\mathbf{s}} \end{matrix}$$

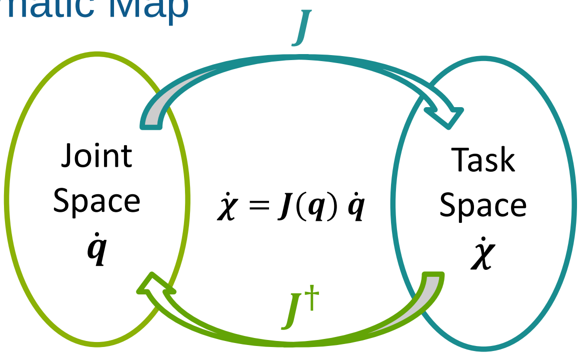
Contents

In this chapter we review the Jacobian analysis for serial robots. First the definition to angular and linear velocities are given, then the Jacobian matrix is defined in conventional and screw-based representation, while their general and iterative derivation methods are given. Next the static wrench and its relation to Jacobian transpose is introduced, and Jacobian characteristics such as singularity, isotropy, dexterity and manipulability are elaborated. Inverse Jacobian solution for fully-, under- and redundantly-actuator robots are formulated, and redundancy resolution schemes are detailed. Finally, Stiffness analysis of robotic manipulators is reviewed in detail.

Jacobian

- Definition
  - ✓ Differential Kinematic Map

    - Forward Map
      Given $\dot{\boldsymbol{q}}$ find $\dot{\boldsymbol{\chi}}$



    - Inverse Map
      Given $\dot{\boldsymbol{\chi}}$ find $\dot{\boldsymbol{q}}$

    - Forward kinematics is a nonlinear map

$$\chi_i = f_i(q_1, q_2, \dots, q_n) \quad \text{for} \quad i = 1,2,\dots,n$$

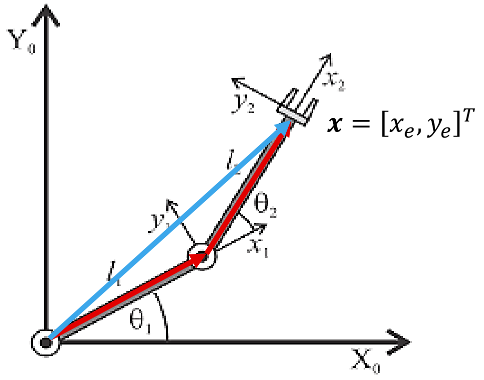
    - Take time derivative:

$$\dot{\boldsymbol{\chi}} = \boldsymbol{J}(\boldsymbol{q})\,\dot{\boldsymbol{q}}\ , \qquad \text{in which,} \qquad \boldsymbol{J}(\boldsymbol{q}) = \begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \frac{\partial f_1}{\partial q_2} & \cdots & \frac{\partial f_1}{\partial q_n} \\ \vdots & & \ddots & \vdots \\ \frac{\partial f_n}{\partial q_1} & \frac{\partial f_n}{\partial q_2} & \cdots & \frac{\partial f_n}{\partial q_n} \end{bmatrix}$$

is called the Jacobian matrix

Jacobian

- Motivating Example
  - ✓ Direct approach
    - Consider 2R manipulator



Denote $\dot{\boldsymbol{q}} = [\dot{\theta}_1, \dot{\theta}_2]^T$ and $\dot{\boldsymbol{\chi}} = [\dot{x}_E, \dot{y}_E]^T$

Forward Kinematics:

$$x_e = l_1 c_1 + l_2 c_{12}$$
$$y_e = l_1 s_1 + l_2 s_{12}$$

Take time derivative:

$$\dot{x}_e = -l_1 s_1 \dot{\theta}_1 - l_2 s_{12}(\dot{\theta}_1 + \dot{\theta}_2)$$
$$\dot{y}_e = l_1 c_1 \dot{\theta}_1 + l_2 c_{12}(\dot{\theta}_1 + \dot{\theta}_2)$$

Determine Jacobian:

$$\dot{\boldsymbol{\chi}} = \boldsymbol{J}\,\dot{\boldsymbol{q}}\ , \qquad \text{in which,} \qquad \boldsymbol{J} = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{bmatrix}$$

Jacobian

- Definition
  - ✓ In General

$$\dot{\boldsymbol{q}} = [\dot{q}_1, \dot{q}_2, \dots, \dot{q}_n]^T \text{ in which } \dot{q}_i = \begin{cases} \dot{\theta}_i & \text{for a revolute joint} \\ \dot{d}_i & \text{for a prismatic joint} \end{cases}$$

- While for the task space variable

$$\dot{\boldsymbol{\chi}} = \dot{\boldsymbol{v}} = \begin{bmatrix} \boldsymbol{v}_E \\ \boldsymbol{\omega}_E \end{bmatrix} \quad \text{For Conventional Jacobian and}$$

$$\dot{\boldsymbol{\chi}} = \dot{\boldsymbol{v}} = \begin{bmatrix} \boldsymbol{\omega}_E \\ \boldsymbol{v}_E \end{bmatrix} \quad \text{For Screw-based Jacobian}$$

In which $\boldsymbol{v}_E$ is the velocity of the end effector, $\boldsymbol{\omega}_E$ denotes the angular velocity of the end effector link.

- Linear velocity and angular velocity sub-Jacobians

$$\dot{\boldsymbol{\chi}} = \begin{bmatrix} \boldsymbol{v}_E \\ \boldsymbol{\omega}_E \end{bmatrix} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} = \begin{bmatrix} \boldsymbol{J}_v \\ \boldsymbol{J}_\omega \end{bmatrix} \dot{\boldsymbol{q}}$$

In which $\boldsymbol{J}_v$ corresponds to the linear velocity Jacobian,
While$\boldsymbol{J}_\omega$ corresponds to the angular velocity Jacobian.

Jacobian

- Definition

  ✓ In General

$$\dot{\boldsymbol{q}} = [\dot{q}_1, \dot{q}_2, \dots, \dot{q}_n]^T \text{ and } \dot{\boldsymbol{\chi}} = \dot{\boldsymbol{\nu}} = \begin{bmatrix} \boldsymbol{v}_E \\ \boldsymbol{\omega}_E \end{bmatrix}$$

The joint and task variable can be given with reference to any frame

Hence,

$$^0\dot{\boldsymbol{q}} = {}^0\boldsymbol{J}\,{}^0\dot{\boldsymbol{\chi}} \quad \text{or} \quad {}^n\dot{\boldsymbol{q}} = {}^n\boldsymbol{J}\,{}^n\dot{\boldsymbol{\chi}}$$

In which

From

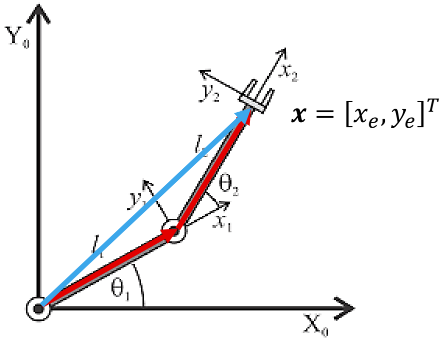
$$\begin{bmatrix} {}^A v \\ {}^A\omega \end{bmatrix} = \left[\begin{array}{c|c} {}^A_B R & 0 \\ \hline 0 & {}^A_B R \end{array}\right] \begin{bmatrix} {}^B v \\ {}^B\omega \end{bmatrix}.$$

We may conclude :

$$^A\boldsymbol{J}(\boldsymbol{q}) = \begin{bmatrix} {}^A_B\boldsymbol{R} & \boldsymbol{0} \\ \boldsymbol{0} & {}^A_B\boldsymbol{R} \end{bmatrix} {}^B\boldsymbol{J}(\boldsymbol{q}).$$

Jacobian

- Motivating Example
  - ✓ Different Frames
    - Consider task variables in end – effector frame {2}



Denote $\dot{\boldsymbol{q}} = [\dot{\theta}_1, \dot{\theta}_2]^T$ and $\dot{\boldsymbol{\chi}} = [\dot{x}_E, \dot{y}_E]^T$

While in base frame: ${}^0\boldsymbol{J} = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{bmatrix}$

In frame {2}: ${}^2\boldsymbol{J} = {}^0_2\boldsymbol{R}\, {}^0\boldsymbol{J} = \begin{bmatrix} c_{12} & -s_{12} \\ s_{12} & c_{12} \end{bmatrix} \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{bmatrix}$

$${}^2\boldsymbol{J} = \cdots = \begin{bmatrix} l_1 s_2 & 0 \\ l_1 c_2 + l_2 & l_2 \end{bmatrix}$$

- Note: Although the appearance is different, the invariant properties of the Jacobians are the same, i.e.

$$det({}^0\boldsymbol{J}) = -l_1 l_2 s_1 c_{12} - l_2^2 s_{12} c_{12} + l_1 l_2 c_1 s_{12} + l_2^2 c_{12} s_{12} = l_1 l_2 s_2$$
$$det({}^2\boldsymbol{J}) = l_1 l_2 s_2 = det({}^0\boldsymbol{J})$$

Jacobian

- Conventional Jacobian:
  - ✓ General Derivation Method

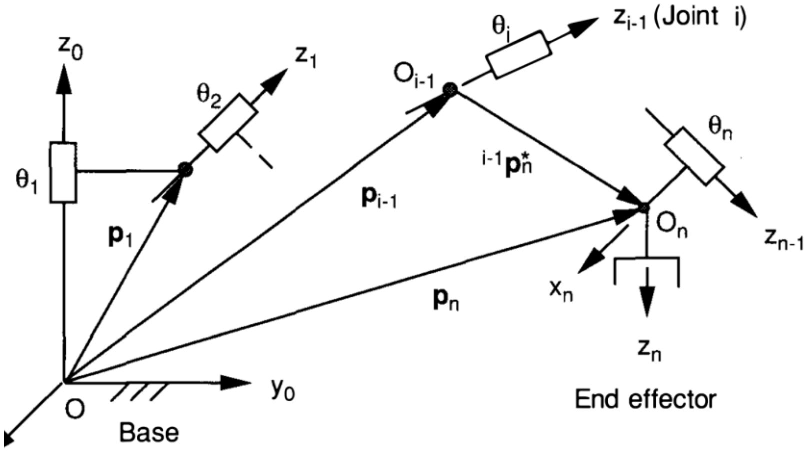
$$\dot{\chi} = J(q)\dot{q} = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix} \dot{q}$$

In which

$$J = [J_1,\ J_2,\ \ldots,\ J_n],$$

$$J_i = \begin{bmatrix} \mathbf{z}_{i-1} \times {}^{i-1}\mathbf{p}_n^* \\ \mathbf{z}_{i-1} \end{bmatrix} \quad \text{for a revolute joint,}$$

$$J_i = \begin{bmatrix} \mathbf{z}_{i-1} \\ \mathbf{0} \end{bmatrix} \quad \text{for a prismatic joint.}$$



Where as shown in the figure ${}^{i-1}\boldsymbol{p}_n^*$ is defined as a vector from origin of the $(i-1)$link frame to the origin of the end effector frame (n)

All the vectors shall be expressed in the frame of interest.

Jacobian

- Conventional Jacobian:
  - ✓ General Derivation Method
    - To derive the Jacobian

      The direction and location of each joint shall be determined.

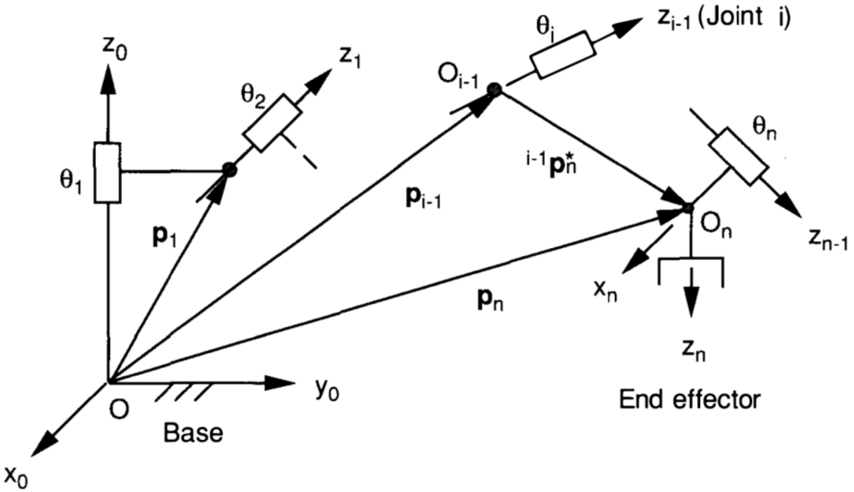
$$\mathbf{z}_{i-1} = {}^{0}R_{i-1}\begin{bmatrix}0\\0\\1\end{bmatrix},$$

$${}^{i-1}\mathbf{p}_n^* = {}^{0}R_{i-1}\,{}^{i-1}\mathbf{r}_i + {}^{i}\mathbf{p}_n^*,$$

Where,

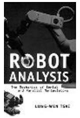
$${}^{i-1}\mathbf{r}_i = \begin{bmatrix}a_i c\theta_i\\a_i s\theta_i\\d_i\end{bmatrix}$$

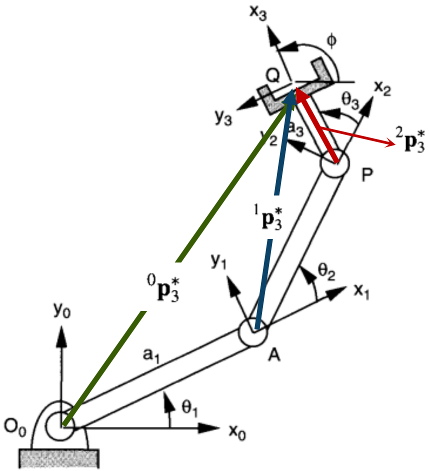
Denotes the vector $\overrightarrow{O_{i-1}O_i}$ expressed in frame $\{i-1\}$.



Conventional Jacobians

- Examples:
  - ✓ Example 1: Planar RRR Manipulator





- Denote $\dot{\boldsymbol{q}} = [\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3]^T$ and $\dot{\boldsymbol{\chi}} = [\dot{x}_E, \dot{y}_E, \dot{\phi}]^T$

  First compute the vectors $\boldsymbol{z}_{i-1}$and ${}^{i-1}\mathbf{p}_3^*$, for $i = 1,2,3$

$$\mathbf{z}_0 = \mathbf{z}_1 = \mathbf{z}_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \qquad {}^1\mathbf{p}_3^* = \begin{bmatrix} a_2 c\theta_{12} + a_3 c\theta_{123} \\ a_2 s\theta_{12} + a_3 s\theta_{123} \\ 0 \end{bmatrix},$$

$${}^2\mathbf{p}_3^* = \begin{bmatrix} a_3 c\theta_{123} \\ a_3 s\theta_{123} \\ 0 \end{bmatrix}, \qquad {}^0\mathbf{p}_3^* = \begin{bmatrix} a_1 c\theta_1 + a_2 c\theta_{12} + a_3 c\theta_{123} \\ a_1 s\theta_1 + a_2 s\theta_{12} + a_3 s\theta_{123} \\ 0 \end{bmatrix},$$

Hence $\dot{\boldsymbol{\chi}} = \boldsymbol{J}\,\dot{\boldsymbol{q}}$ where,

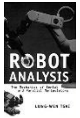
$$J = \begin{bmatrix} -(a_1 s\theta_1 + a_2 s\theta_{12} + a_3 s\theta_{123}) & -(a_2 s\theta_{12} + a_3 s\theta_{123}) & -a_3 s\theta_{123} \\ (a_1 c\theta_1 + a_2 c\theta_{12} + a_3 c\theta_{123}) & (a_2 c\theta_{12} + a_3 c\theta_{123}) & a_3 c\theta_{123} \\ 1 & 1 & 1 \end{bmatrix}.$$

Note Jacobian of the wrist position $\boldsymbol{P}$ will be:

$$J = \begin{bmatrix} -(a_1 s\theta_1 + a_2 s\theta_{12}) & -(a_2 s\theta_{12}) & 0 \\ (a_1 c\theta_1 + a_2 c\theta_{12}) & (a_2 c\theta_{12}) & 0 \\ 1 & 1 & 1 \end{bmatrix}.$$

Conventional Jacobians

- Examples:
  - ✓ Example 2: SCARA Manipulator



- Denote $\dot{\boldsymbol{q}} = [\dot{\theta}_1, \dot{\theta}_2, \dot{d}_3, \dot{\theta}_4]^T$ and $\dot{\boldsymbol{\chi}} = [\dot{x}_E, \dot{y}_E, \dot{z}_E, \omega_E]^T$

Recall DH-parameters and homogeneous transformations:

$$ {}^0_1T = \begin{bmatrix} c_1 & -s_1 & 0 & a_1c_1 \\ s_1 & c_1 & 0 & a_1s_1 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^1_2T = \begin{bmatrix} c_2 & s_2 & 0 & a_2c_2 \\ s_2 & -c_2 & 0 & a_2s_2 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, $$

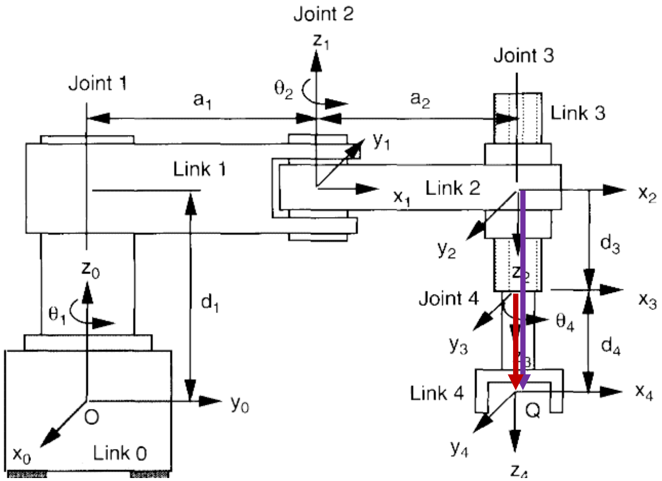
$$ {}^2_3T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^3_4T = \begin{bmatrix} c_4 & -s_4 & 0 & 0 \\ s_4 & c_4 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}. $$

First compute the vectors $\boldsymbol{z}_{i-1}$

$$ \boldsymbol{z}_0 = \boldsymbol{z}_1 = [0, 0, 1]^T, \qquad \boldsymbol{z}_2 = \boldsymbol{z}_3 = [0,0,-1]^T $$

Now compute: ${}^{i-1}\boldsymbol{p}_4^*$, for $i = 3{,}4$ by inspection (red/purple vectors):

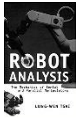
$$ {}^3\boldsymbol{p}_4^* = \begin{bmatrix} 0 \\ 0 \\ -d_4 \end{bmatrix}, \quad {}^2\boldsymbol{p}_4^* = \begin{bmatrix} 0 \\ 0 \\ -d_3 - d_4 \end{bmatrix}. $$

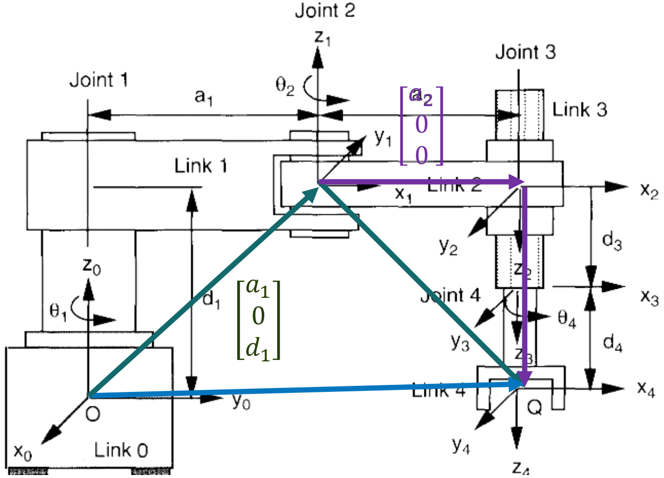


i	α_i	a_i	d_i	θ_i
1	0	a_1	d_1	θ_1
2	π	a_2	0	θ_2
3	0	0	d_3	0
4	0	0	d_4	θ_4

Conventional Jacobians

- Examples:
  - ✓ Example 2: SCARA Manipulator





- Denote $\dot{\boldsymbol{q}} = \left[\dot{\theta}_1, \dot{\theta}_2, \dot{d}_3, \dot{\theta}_4\right]^T$ and $\dot{\boldsymbol{\chi}} = [\dot{x}_E, \dot{y}_E, \dot{z}_E, \omega_E]^T$

  Furthermore, calculate ${}^{i-1}\boldsymbol{p}_4^*$, for $i = 1{,}2$ iteratively:

$$
{}^{1}\boldsymbol{p}_4^* = {}^{0}_{2}\boldsymbol{R}\begin{bmatrix} a_2 \\ 0 \\ 0 \end{bmatrix} + {}^{2}\boldsymbol{p}_4^* = \begin{bmatrix} a_2 c_{12} \\ a_2 s_{12} \\ -d_3 - d_4 \end{bmatrix},
$$

$$
{}^{0}\boldsymbol{p}_4^* = {}^{0}_{1}\boldsymbol{R}\begin{bmatrix} a_1 \\ 0 \\ d_1 \end{bmatrix} + {}^{1}\boldsymbol{p}_4^* = \begin{bmatrix} a_1 c_1 + a_2 c_{12} \\ a_1 s_1 + a_2 s_{12} \\ d_1 - d_3 - d_4 \end{bmatrix}.
$$

  Hence $\dot{\boldsymbol{\chi}} = \boldsymbol{J}\,\dot{\boldsymbol{q}}$ where, $\boldsymbol{J}$ is a 6×4 matrix as:

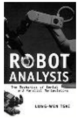
$$
\boldsymbol{J} = \begin{bmatrix} \boldsymbol{z}_0 \times {}^{0}\boldsymbol{p}_4^* & \boldsymbol{z}_1 \times {}^{1}\boldsymbol{p}_4^* & \boldsymbol{z}_3 & \boldsymbol{z}_4 \times {}^{3}\boldsymbol{p}_4^* \\ \boldsymbol{z}_0 & \boldsymbol{z}_1 & \boldsymbol{0} & \boldsymbol{z}_4 \end{bmatrix} = \begin{bmatrix} -a_1 s_1 - a_2 s_{12} & a_2 s_{12} & 0 & 0 \\ a_1 c_1 + a_2 c_{12} & a_2 c_{12} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & -1 \end{bmatrix}.
$$

  Note: The angular velocity is found as $\omega_E = \dot{\theta}_1 + \dot{\theta}_2 - \dot{\theta}_4$ in z direction.

Matlab Program: Jacobian_scara.m

Conventional Jacobians

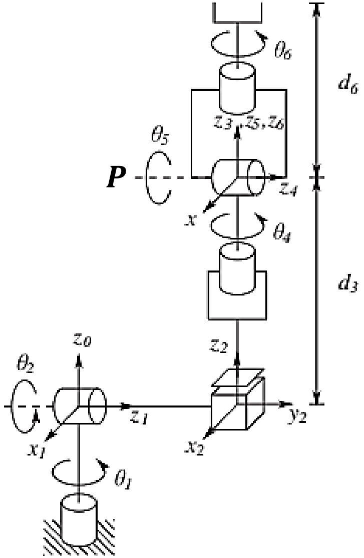
✓ Example 3: Stanford Manipulator



- For wrist $\boldsymbol{P}$ position $\dot{\boldsymbol{q}} = \left[\dot\theta_1, \dot\theta_2, \dot d_3, \dot\theta_4, \dot\theta_5, \dot\theta_6\right]^T$ and $\dot{\boldsymbol{\chi}} = \left[\dot{\boldsymbol{x}}_p, \boldsymbol{\omega}_p\right]^T$

  Recall DH parameters, and homogeneous transformations

i	α_i	a_i	d_i	θ_i
1	$-\pi/2$	0	0	θ_1
2	$\pi/2$	0	d_2	θ_2
3	0	0	d_3	0
4	$-\pi/2$	0	0	θ_4
5	$\pi/2$	0	0	θ_5
6	0	0	0	θ_6



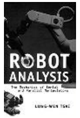
$$
{}^0A_1 = \begin{bmatrix} c\theta_1 & 0 & -s\theta_1 & 0 \\ s\theta_1 & 0 & c\theta_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^1A_2 = \begin{bmatrix} c\theta_2 & 0 & s\theta_2 & 0 \\ s\theta_2 & 0 & -c\theta_2 & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix},
$$

$$
{}^2A_3 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^3A_4 = \begin{bmatrix} c\theta_4 & 0 & -s\theta_4 & 0 \\ s\theta_4 & 0 & c\theta_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},
$$

$$
{}^4A_5 = \begin{bmatrix} c\theta_5 & 0 & s\theta_5 & 0 \\ s\theta_5 & 0 & -c\theta_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^5A_6 = \begin{bmatrix} c\theta_6 & -s\theta_6 & 0 & 0 \\ s\theta_6 & c\theta_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
$$

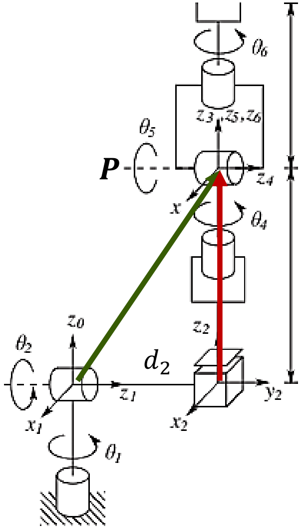
Conventional Jacobians

✓ Example 3: Stanford Manipulator



- For wrist $\boldsymbol{P}$ position $\dot{\boldsymbol{q}} = [\dot{\theta}_1, \dot{\theta}_2, \dot{d}_3, \dot{\theta}_4, \dot{\theta}_5, \dot{\theta}_6]^T$ and $\dot{\boldsymbol{\chi}} = [\dot{\boldsymbol{x}}_p, \boldsymbol{\omega}_p]^T$

  First compute the vectors $\boldsymbol{z}_{i-1}$



$$\mathbf{z}_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{z}_1 = {}^0R_1 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -s\theta_1 \\ c\theta_1 \\ 0 \end{bmatrix}, \quad \mathbf{z}_2 = \mathbf{z}_3 = {}^0R_2 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} c\theta_1 s\theta_2 \\ s\theta_1 s\theta_2 \\ c\theta_2 \end{bmatrix},$$

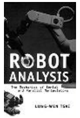
$$\mathbf{z}_4 = {}^0R_4 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -s\theta_1 s\theta_4 + c\theta_1 c\theta_2 c\theta_4 \\ c\theta_1 s\theta_4 + s\theta_1 c\theta_2 c\theta_4 \\ -s\theta_2 c\theta_4 \end{bmatrix}, \quad \mathbf{z}_5 = {}^0R_5 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} s\theta_1 c\theta_4 s\theta_5 + c\theta_1 c\theta_2 s\theta_4 s\theta_5 + c\theta_1 s\theta_2 c\theta_5 \\ -c\theta_1 c\theta_4 s\theta_5 + s\theta_1 c\theta_2 s\theta_4 s\theta_5 + s\theta_1 s\theta_2 c\theta_5 \\ -s\theta_2 s\theta_4 s\theta_5 + c\theta_2 c\theta_5 \end{bmatrix}.$$

and ${}^{i-1}\mathbf{p}_6^*$, for $i = 1,2, \dots, 6$: ${}^3\mathbf{p}_6^* = {}^4\mathbf{p}_6^* = {}^5\mathbf{p}_6^* = \mathbf{0}$

$${}^2\mathbf{p}_6^* = \begin{bmatrix} d_3 c\theta_1 s\theta_2 \\ d_3 s\theta_1 s\theta_2 \\ d_3 c\theta_2 \end{bmatrix}, \quad {}^1\mathbf{p}_6^* = \begin{bmatrix} d_3 c\theta_1 s\theta_2 - d_2 s\theta_1 \\ d_3 s\theta_1 s\theta_2 + d_2 c\theta_1 \\ d_3 c\theta_2 \end{bmatrix}, \quad {}^0\mathbf{p}_6^* = \begin{bmatrix} d_3 c\theta_1 s\theta_2 - d_2 s\theta_1 \\ d_3 s\theta_1 s\theta_2 + d_2 c\theta_1 \\ d_3 c\theta_2 \end{bmatrix}.$$

Conventional Jacobians

✓ Example 3: Stanford Manipulator



- For wrist $\boldsymbol{P}$ position $\dot{\boldsymbol{q}} = [\dot{\theta}_1, \dot{\theta}_2, \dot{d}_3, \dot{\theta}_4, \dot{\theta}_5, \dot{\theta}_6]^T$ and $\dot{\boldsymbol{\chi}} = [\dot{\boldsymbol{x}}_p, \boldsymbol{\omega}_p]^T$

  Hence $\dot{\boldsymbol{\chi}} = \boldsymbol{J}\,\dot{\boldsymbol{q}}$ in reference frame $\{0\}$ is given by:

$$J = \begin{bmatrix} -d_3 s\theta_1 s\theta_2 - d_2 c\theta_1 & d_3 c\theta_1 c\theta_2 & c\theta_1 s\theta_2 & 0 & 0 & 0 \\ d_3 c\theta_1 s\theta_2 - d_2 s\theta_1 & d_3 s\theta_1 c\theta_2 & s\theta_1 s\theta_2 & 0 & 0 & 0 \\ 0 & -d_3 s\theta_2 & c\theta_2 & 0 & 0 & 0 \\ 0 & -s\theta_1 & 0 & c\theta_1 s\theta_2 & & \\ 0 & c\theta_1 & 0 & s\theta_1 s\theta_2 & \mathbf{z}_4 & \mathbf{z}_5 \\ 1 & 0 & 0 & c\theta_2 & & \end{bmatrix}$$

  Where $\boldsymbol{z}_4, \boldsymbol{z}_5$ are joint axis unit vectors given before.

  Note 1: Since the wrist position is considered for the manipulations, the Jacobian matrix is upper triangular.

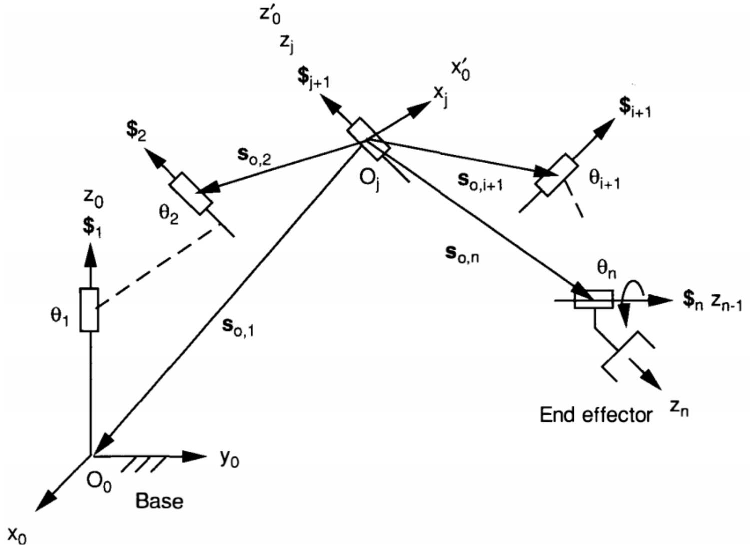
  Note 2: The Jacobian matrix will be much simplified if it is given w.r.t frame $\{2\}$.

Jacobian

- Screw-based Jacobian:
  - ✓ General Derivation Method

$$\dot{\chi} = \begin{bmatrix} \boldsymbol{\omega}_E \\ \boldsymbol{v}_P \end{bmatrix} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} = \begin{bmatrix} \boldsymbol{J}_\omega \\ \boldsymbol{J}_v \end{bmatrix} \dot{\boldsymbol{q}}$$

$$\dot{\chi} = \sum_{i=1}^{n} \hat{\$}_i \dot{q}_i$$



Where the unit twist is defined in slide 15 as:

- For rotary joint (R)

$$\$ = \begin{bmatrix} \hat{s} \\ s_O \times \hat{s} \end{bmatrix} \dot{\theta}.$$

- For prismatic joint (P)

$$\$ = \begin{bmatrix} 0 \\ \hat{s} \end{bmatrix} \dot{d}.$$

Therefore, the Jacobian matrix consists of the unit screws:

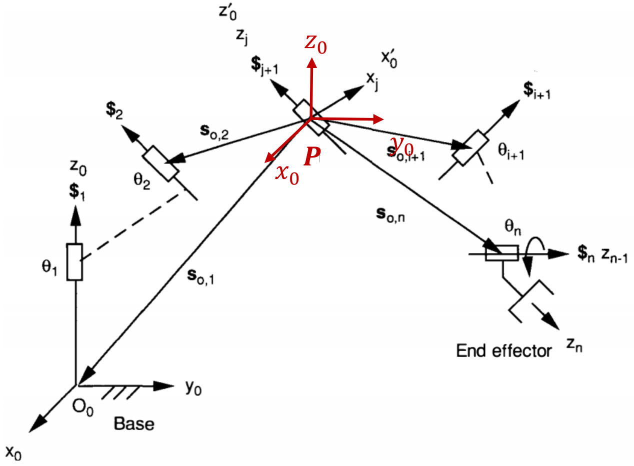
$$\boldsymbol{J} = [\hat{\$}_1, \hat{\$}_2, \dots, \hat{\$}_n].$$

Jacobian

- Screw-based Jacobian:
  - ✓ General Derivation Method

$$\dot{\chi} = \begin{bmatrix} \boldsymbol{\omega}_E \\ \boldsymbol{v}_P \end{bmatrix} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} = \begin{bmatrix} \boldsymbol{J}_\omega \\ \boldsymbol{J}_v \end{bmatrix} \dot{\boldsymbol{q}}$$

$$\boldsymbol{J} = [\hat{\$}_1, \hat{\$}_2, \dots, \hat{\$}_n]$$



  - ▪ Note that the task space variable

$$\dot{\chi} = \begin{bmatrix} \boldsymbol{\omega}_E \\ \boldsymbol{v}_P \end{bmatrix}$$

    Consist of the angular velocity of the end effector

    But linear velocity of any point $\boldsymbol{P}$ (including the end effector $\boldsymbol{E}$)
  - ▪ To assign the screw parameters

    Consider an instantaneous fixed frame on the point of interest $\boldsymbol{P}$.
  - ▪ The direction of the joint axes can be determined by inspection or by the third column of ${}_{i-1}^{\;\;0}\boldsymbol{A}$.
  - ▪ The distance of the screw axes from this instantaneous frame is denoted by $\boldsymbol{s}_{o,i}$ represented in this instantaneous frame
  - ▪ If the origin of intermediate frames (3 or 4) is used as the point of interest, the Jacobian is much simpler. Notice the notation of $\boldsymbol{s}_{o,i}$ denotes the origin of frame i w.r.t the instantaneous frame on point $\boldsymbol{P}$.

Jacobian

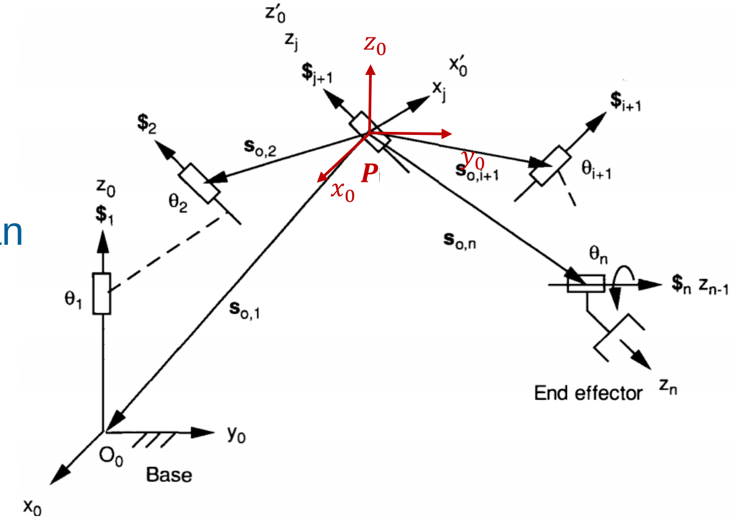
- Screw-based Jacobian:
  - ✓ Iterative Recipe:
    - Initial Conditions

      Consider frame $\{j\}$ to represent the Jacobian

      Begin with $s_{j+1} = [0,0,1]^T, s_{o,j+1} = [0,0,0]^T$

    - Forward Computation

      For $i = j+1, \ldots, n-1$ compute



$$\mathbf{s}_{i+1} = ({}^jR_i)({}^i\mathbf{z}_i),$$

$$\mathbf{s}_{o,i+1} = \mathbf{s}_{o,i} + ({}^jR_i)({}^i\mathbf{r}_i),$$

$${}^jR_{i+1} = ({}^jR_i)({}^iR_{i+1}).$$

Jacobian

- Screw-based Jacobian:
  - ✓ Iterative Recipe:
    - ▪ Backward Computation

For $i = j-1, \ldots, 0$ compute

$$\mathbf{s}_{i+1} = ({}^{j}R_i)({}^{i}\mathbf{z}_i),$$

$$\mathbf{s}_{o,i+1} = \mathbf{s}_{o,i+2} - ({}^{j}R_{i+1})({}^{i+1}\mathbf{r}_{i+1}),$$

$${}^{j}R_{i-1} = ({}^{j}R_i)({}^{i}R_{i-1}),$$

Where ${}^{i}R_{i-1} = ({}^{i-1}R_i)^{\mathrm{T}}$

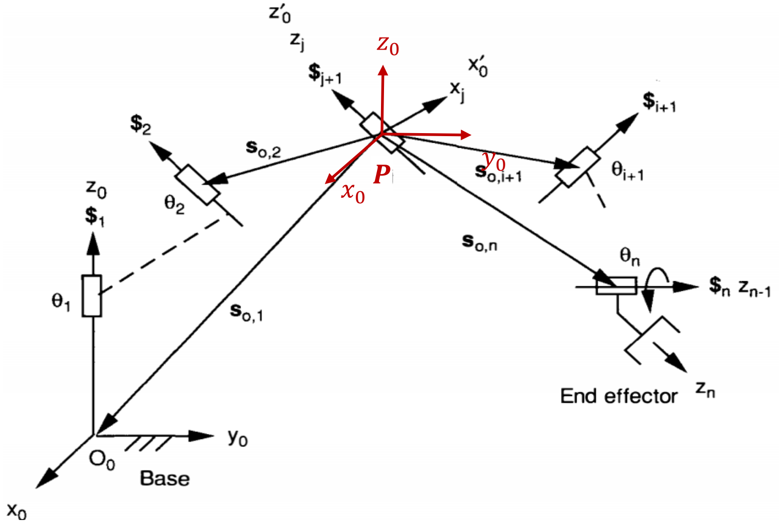
Furthermore,

$${}^{i}\mathbf{z}_i = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \text{and} \quad {}^{i}\mathbf{r}_i = \begin{bmatrix} a_i \\ d_i s\alpha_i \\ d_i c\alpha_i \end{bmatrix}$$

Is the position vector from O_{i-1} to O_i expressed in i^{th} frame.

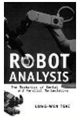
Assembling the unit screws derived above, yields to the Jacobian of the point ***P*** as:

$$\begin{bmatrix} {}^{0}\boldsymbol{\omega}_E \\ {}^{0}\boldsymbol{v}_p \end{bmatrix} = \boldsymbol{J}\dot{\boldsymbol{q}} \rightarrow \boldsymbol{J} = [J_n, J_2, \ldots, J_n] \quad \text{and } J_i = \begin{bmatrix} \boldsymbol{s}_i \\ \boldsymbol{s}_{o,i} \times \boldsymbol{s}_i \end{bmatrix} \text{ for (R) joint or } J_i = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{s}_i \end{bmatrix} \text{ for (P) joints.}$$



Screw-Based Jacobians

- Examples:
  - ✓ Example 1: Planar RRR Manipulator



- Denote $\dot{\boldsymbol{q}} = [\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3]^T$ and $\dot{\boldsymbol{\chi}} = [\boldsymbol{\omega}_E, \boldsymbol{v}_Q]^T$
- Put an instantaneous frame $\{0\}$ on point $\boldsymbol{Q}$.
- Find the screw details by inspection:

For $\$_3$:

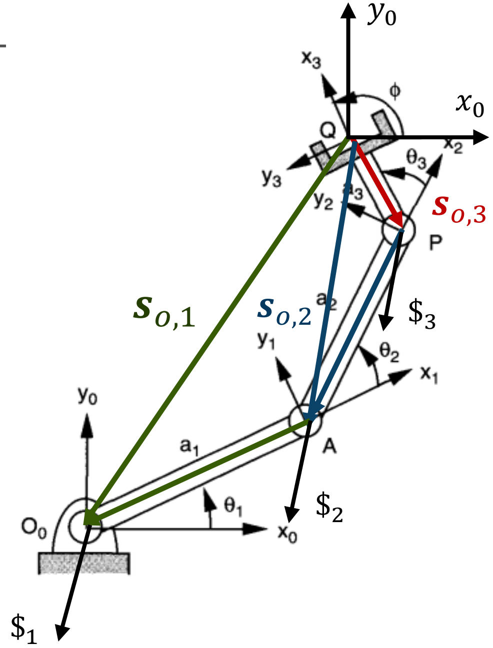
$$\boldsymbol{s}_3 = [0,0,1]^T, \ \ \boldsymbol{s}_{o,3} = {}^0_3\boldsymbol{R} \begin{bmatrix} -a_3 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -a_3 c_{123} \\ -a_3 s_{123} \\ 0 \end{bmatrix}.$$

For $\$_2$:

$$\boldsymbol{s}_2 = [0,0,1]^T, \ \ \boldsymbol{s}_{o,2} = \boldsymbol{s}_{o,3} + {}^0_2\boldsymbol{R} \begin{bmatrix} -a_2 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -a_2 c_{12} - a_3 c_{123} \\ -a_2 s_{12} - a_3 s_{123} \\ 0 \end{bmatrix}.$$

For $\$_1$:

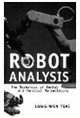
$$\boldsymbol{s}_1 = [0,0,1]^T, \ \ \boldsymbol{s}_{o,1} = \boldsymbol{s}_{o,2} + {}^0_1\boldsymbol{R} \begin{bmatrix} -a_1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -a_1 c_1 - a_2 c_{12} - a_3 c_{123} \\ -a_1 s_1 - a_2 s_{12} - a_3 s_{123} \\ 0 \end{bmatrix}$$



Matlab Program: Jacobian_screw_RRR_inspection.m

Screw-Based Jacobians

✓ Example 1: Planar RRR Manipulator



- Denote $\dot{\boldsymbol{q}} = [\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3]^T$ and $\dot{\boldsymbol{\chi}} = [\boldsymbol{\omega}_E, \boldsymbol{v}_Q\,]^T$
- The Jacobian $\dot{\boldsymbol{\chi}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}$ is found by $\boldsymbol{J} = [\$_1, \$_2, \$_3]$,
- In which, $\$_i = \begin{bmatrix} \boldsymbol{s}_i \\ \boldsymbol{s}_{o,i} \times \boldsymbol{s}_i \end{bmatrix}$, hence:

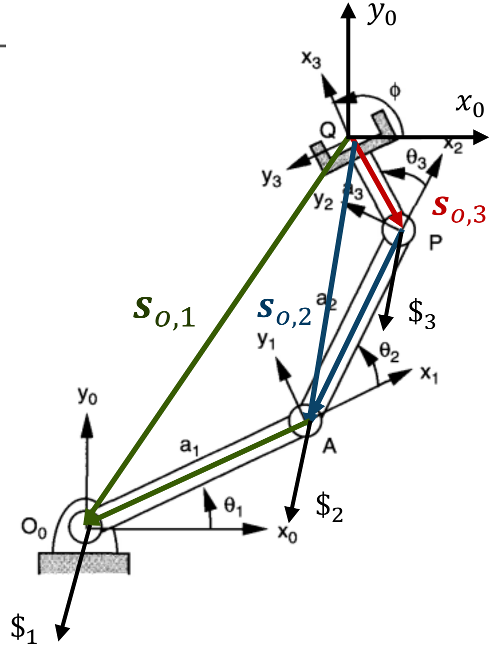
$$\boldsymbol{J} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \\ -a_1 s_1 - a_2 s_{12} - a_3 s_{123} & -a_2 s_{12} - a_3 s_{123} & -a_3 s_{123} \\ a_1 c_1 + a_2 c_{12} + a_3 c_{123} & a_2 c_{12} + a_3 c_{123} & a_3 c_{123} \\ 0 & 0 & 0 \end{bmatrix}$$

- In planar coordinates, this means:

$$\omega_{\boldsymbol{z}} = \dot{\phi} = \dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3$$

$$\dot{x}_{\boldsymbol{Q}} = -(a_1 s_1 + a_2 s_{12} + a_3 s_{123})\dot{\theta}_1 - (a_2 s_{12} + a_3 s_{123})\dot{\theta}_2 - (a_3 s_{123})\dot{\theta}_3$$

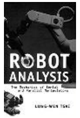
$$\dot{y}_{\boldsymbol{Q}} = (\ a_1 c_1 + a_2 c_{12} + a_3 c_{123})\dot{\theta}_1 - (a_2 c_{12} + a_3 c_{123})\dot{\theta}_2 - (a_3 c_{123})\dot{\theta}_3$$



Which is exactly as found before (see slide 23).

Screw-Based Jacobians

✓ Example 1: Planar RRR Manipulator



- Jacobian for wrist Point $\boldsymbol{P}$: $\dot{\boldsymbol{q}} = [\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3]^T$ and $\dot{\boldsymbol{\chi}} = [\boldsymbol{\omega}_E, \boldsymbol{v}_p\,]^T$
- Put an instantaneous frame $\{0\}$ on point $\boldsymbol{P}$.
- Find the screw details by inspection:

$$\boldsymbol{s}_3 = \boldsymbol{s}_2 = \boldsymbol{s}_1 = [0,0,1]^T, \quad \boldsymbol{s}_{o,3} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{s}_{o,2} = {}^0_2\boldsymbol{R} \begin{bmatrix} -a_2 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -a_2 c_{12} \\ -a_2 s_{12} \\ 0 \end{bmatrix}.$$

$$\boldsymbol{s}_{o,1} = \boldsymbol{s}_{o,2} + {}^0_1\boldsymbol{R} \begin{bmatrix} -a_1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -a_1 c_1 - a_2 c_{12} \\ -a_1 s_1 - a_2 s_{12} \\ 0 \end{bmatrix}$$

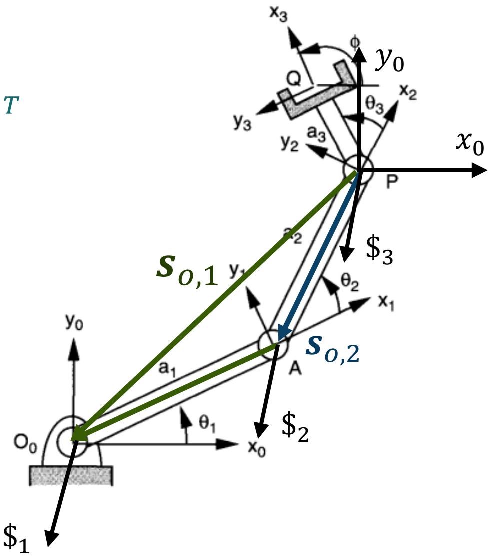
- Hence,

$$\boldsymbol{J} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \\ -a_1 s_1 - a_2 s_{12} & -a_2 s_{12} & 0 \\ a_1 c_1 + a_2 c_{12} & a_2 c_{12} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

or component-wise:

$$\omega_z = \dot{\phi} = \dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3$$
$$\dot{x}_p = -(a_1 s_1 + a_2 s_{12})\dot{\theta}_1 - (a_2 s_{12})\dot{\theta}_2$$
$$\dot{y}_p = (a_1 c_1 + a_2 c_{12})\dot{\theta}_1 + (a_2 c_{12})\dot{\theta}_2$$

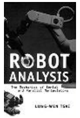
Which is exactly as found before (see slide 23).



Matlab Program: Jacobian_screw_RRR_inspection.m

Screw-Based Jacobians

- Examples:
  - ✓ Example 1: Planar RRR Manipulator



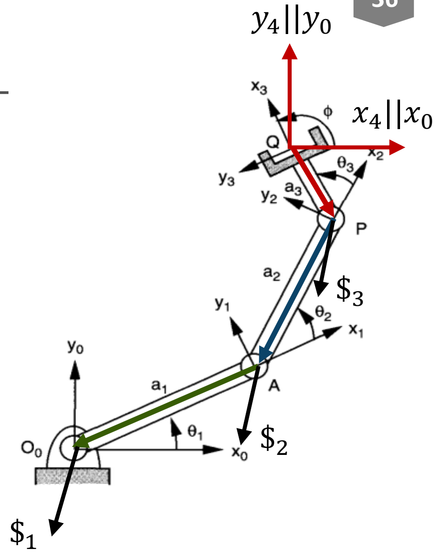
- Denote $\dot{\boldsymbol{q}} = [\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3]^T$ and $\dot{\boldsymbol{\chi}} = [\boldsymbol{\omega}_E, \boldsymbol{v}_Q]^T$
- Put an instantaneous frame $\{0\}$ on point $\boldsymbol{Q}$.
- Find the screw details by iteration:
- Initial Conditions:

  For $j = 3, \boldsymbol{s}_4 = [0,0,1]^T, \boldsymbol{s}_{o,4} = [0,0,0]^T$.
- (BI) Now look backward, for $i = 2$:

$$\boldsymbol{s}_3 = {}^3_2\boldsymbol{R}\,\boldsymbol{z}_2 = \begin{bmatrix}0\\0\\1\end{bmatrix},\ {}^0\boldsymbol{s}_{o,3} = {}^0\boldsymbol{s}_{o,4} - {}^0_3\boldsymbol{R}\,{}^3_3\boldsymbol{R}\begin{bmatrix}a_3\\0\\0\end{bmatrix} = \begin{bmatrix}-a_3c_{123}\\-a_3s_{123}\\0\end{bmatrix}$$

$$\boldsymbol{s}_2 = {}^3_1\boldsymbol{R}\,\boldsymbol{z}_1 = \begin{bmatrix}0\\0\\1\end{bmatrix},\ \boldsymbol{s}_{o,2} = \boldsymbol{s}_{o3} - {}^0_2\boldsymbol{R}\begin{bmatrix}a_2\\0\\0\end{bmatrix} = \begin{bmatrix}-a_2c_{12} - a_3c_{123}\\-a_2s_{12} - a_3s_{123}\\0\end{bmatrix}$$

$$\boldsymbol{s}_1 = {}^3_0\boldsymbol{R}\,\boldsymbol{z}_0 = \begin{bmatrix}0\\0\\1\end{bmatrix},\ \boldsymbol{s}_{o,1} = \boldsymbol{s}_{o,2} - {}^0_1\boldsymbol{R}\begin{bmatrix}a_1\\0\\0\end{bmatrix} = \begin{bmatrix}-a_1c_1 - a_2c_{12} - a_3c_{123}\\-a_1s_1 - a_2s_{12} - a_3s_{123}\\0\end{bmatrix}$$

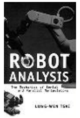


Matlab Program: Jacobian_screw_RRR.m

$$\begin{bmatrix}{}^0\boldsymbol{\omega}_E\\{}^0\boldsymbol{v}_Q\end{bmatrix} = \boldsymbol{J}\dot{\boldsymbol{q}} \ \rightarrow \boldsymbol{J} = [J_1, J_2, \dots, J_3], \text{ and } J_i = \begin{bmatrix}\boldsymbol{s}_i\\ \boldsymbol{s}_{o,i}\times\boldsymbol{s}_i\end{bmatrix}$$

$$\boldsymbol{J} = \begin{bmatrix}0 & 0 & 0\\0 & 0 & 0\\1 & 1 & 1\\-a_1s_1 - a_2s_{12} - a_3s_{123} & -a_2s_{12} - a_3s_{123} & -a_3s_{123}\\a_1c_1 + a_2c_{12} + a_3c_{123} & a_2c_{12} + a_3c_{123} & a_3c_{123}\\0 & 0 & 0\end{bmatrix}$$

Screw-Based Jacobians



✓ Example 2: Elbow Manipulator

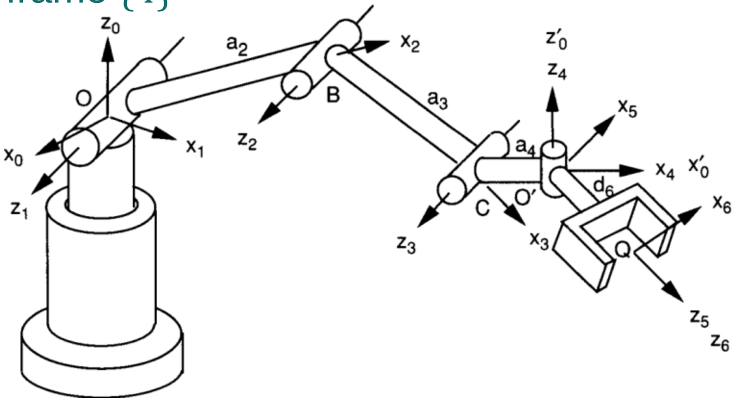
- Consider the point of Interest $\boldsymbol{O'}$ the origin of frame $\{4\}$
- Denote $\dot{\boldsymbol{q}} = \left[\dot{\theta}_1, \dots, \dot{\theta}_6\right]^T$ and $\dot{\boldsymbol{\chi}} = [\boldsymbol{\omega}_E, \boldsymbol{v}_{O'}]^T$
- Initial Conditions:

  For $j = 4, \boldsymbol{s}_5 = [0,0,1]^T, s_{o,5} = [0,0,0]^T$.
- (FI) Find the 6th axes details. For $i = 5$:

$$\mathbf{s}_6 = {}^4R_5\,{}^5\mathbf{z}_5 = \begin{bmatrix} s\theta_5 \\ -c\theta_5 \\ 0 \end{bmatrix}, \quad \mathbf{s}_{o,6} = \mathbf{s}_{o,5} + {}^4R_5\,{}^5r_5 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

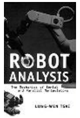
- (BI) Now look backward, For $i = 3$:

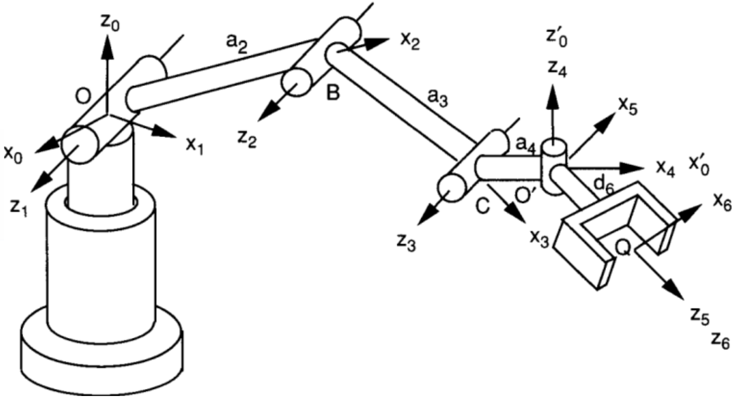
$$\mathbf{s}_4 = {}^4R_3\,{}^3\mathbf{z}_3 = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}, \quad \mathbf{s}_{o,4} = \mathbf{s}_{o,5} - {}^4R_4\,{}^4r_4 = \begin{bmatrix} -a_4 \\ 0 \\ 0 \end{bmatrix}, \quad {}^4R_2 = {}^4R_3\,{}^3R_2 = \begin{bmatrix} c\theta_{34} & s\theta_{34} & 0 \\ 0 & 0 & -1 \\ -s\theta_{34} & c\theta_{34} & 0 \end{bmatrix}.$$



Screw-Based Jacobians

✓ Example 2: Elbow Manipulator





- Denote $\dot{\boldsymbol{q}} = [\dot{\theta}_1, \dots, \dot{\theta}_6]^T$ and $\dot{\boldsymbol{\chi}} = [\boldsymbol{\omega}_E, \boldsymbol{v}_{O'}]^T$
- For $i = 2$

$$\mathbf{s}_3 = {}^4R_2\, {}^2\mathbf{z}_2 = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}, \quad \mathbf{s}_{o,3} = \mathbf{s}_{o,4} - {}^4R_3\, {}^3r_3 = \begin{bmatrix} -a_3 c\theta_4 - a_4 \\ 0 \\ a_3 s\theta_4 \end{bmatrix},$$

$${}^4R_1 = {}^4R_2\, {}^2R_1 = \begin{bmatrix} c\theta_{234} & s\theta_{234} & 0 \\ 0 & 0 & -1 \\ -s\theta_{234} & c\theta_{234} & 0 \end{bmatrix}.$$

- For $i = 1$:

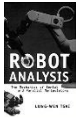
$$\mathbf{s}_2 = {}^4R_1\, {}^1\mathbf{z}_1 = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}, \quad \mathbf{s}_{o,2} = \mathbf{s}_{o,3} - {}^4R_2\, {}^2r_2 = \begin{bmatrix} -a_2 c\theta_{34} - a_3 c\theta_4 - a_4 \\ 0 \\ a_2 s\theta_{34} + a_3 s\theta_4 \end{bmatrix}, \quad {}^4R_0 = {}^4R_1\, {}^1R_0 = \begin{bmatrix} c\theta_1 c\theta_{234} & s\theta_1 c\theta_{234} & s\theta_{234} \\ -s\theta_1 & c\theta_1 & 0 \\ -c\theta_1 s\theta_{234} & -s\theta_1 s\theta_{234} & c\theta_{234} \end{bmatrix}.$$

- For $i = 0$:

$$\mathbf{s}_1 = {}^4R_0\, {}^0\mathbf{z}_0 = \begin{bmatrix} s\theta_{234} \\ 0 \\ c\theta_{234} \end{bmatrix}, \quad \mathbf{s}_{o,1} = \mathbf{s}_{o,2} - {}^4R_1\, {}^1r_1 = \begin{bmatrix} -a_2 c\theta_{34} - a_3 c\theta_4 - a_4 \\ 0 \\ a_2 s\theta_{34} + a_3 s\theta_4 \end{bmatrix}.$$

Screw-Based Jacobians

✓ Example 2: Elbow Manipulator



- Denote $\dot{\boldsymbol{q}} = \left[\dot{\theta}_1, \ldots, \dot{\theta}_6\right]^T$ and $\dot{\boldsymbol{\chi}} = [\boldsymbol{\omega}_E, \boldsymbol{v}_{O'}]^T$

  Hence $\dot{\boldsymbol{\chi}} = \boldsymbol{J}\,\dot{\boldsymbol{q}}$ in reference frame $\{4\}$ is given by:

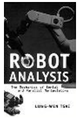
$$
{}^4\boldsymbol{J} = \begin{bmatrix}
s\theta_{234} & 0 & 0 & 0 & 0 & s\theta_5 \\
0 & -1 & -1 & -1 & 0 & -c\theta_5 \\
c\theta_{234} & 0 & 0 & 0 & 1 & 0 \\
0 & a_2 s\theta_{34} + a_3 s\theta_4 & a_3 s\theta_4 & 0 & 0 & 0 \\
x_{51} & 0 & 0 & 0 & 0 & 0 \\
0 & a_2 c\theta_{34} + a_3 c\theta_4 + a_4 & a_3 c\theta_4 + a_4 & a_4 & 0 & 0
\end{bmatrix}
$$

In which,

$$
x_{51} = a_2 c\theta_2 + a_3 c\theta_{23} + a_4 c\theta_{234}.
$$

Screw-Based Jacobians

✓ Example 3: Stanford Arm (2RP3R)



- Consider the wrist point $\boldsymbol{P}$ the origin of frame $\{3\}$
- Denote $\dot{\boldsymbol{q}} = [\dot{\theta}_1, \dots, \dot{\theta}_6]^T$ and $\dot{\boldsymbol{\chi}} = [\boldsymbol{\omega}_E, \boldsymbol{v}_P]^T$
- Initial Conditions:

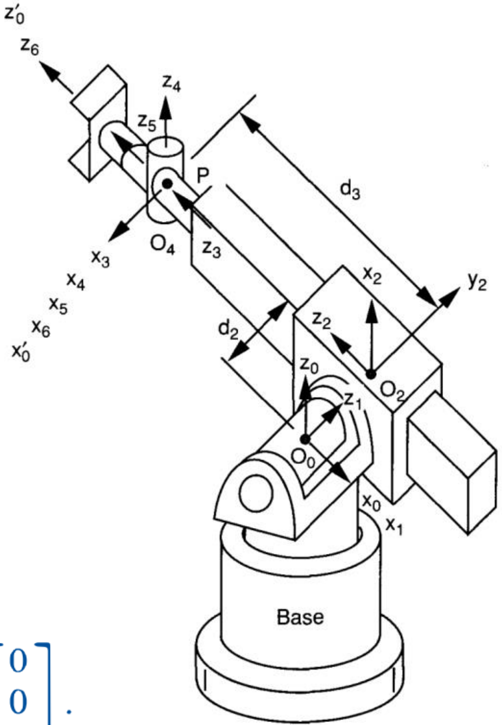
  For $j = 3, \boldsymbol{s}_4 = [0,0,1]^T, s_{o,4} = [0,0,0]^T$.
- Find the 5th and 6th axes details. For $i = 4$:

$$\mathbf{s}_5 = {}^3R_4\,{}^4\mathbf{z}_4 = \begin{bmatrix} -s\theta_4 \\ c\theta_4 \\ 0 \end{bmatrix}, \quad \mathbf{s}_{o,5} = \mathbf{s}_{o,4} + {}^3R_4\,{}^4r_4 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},$$

$${}^3R_5 = {}^3R_4\,{}^4R_5 = \begin{bmatrix} c\theta_4 c\theta_5 & -s\theta_4 & c\theta_4 s\theta_5 \\ s\theta_4 c\theta_5 & c\theta_4 & s\theta_4 s\theta_5 \\ -s\theta_5 & 0 & c\theta_5 \end{bmatrix}.$$

For $i = 5$:

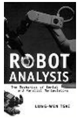
$$\mathbf{s}_6 = {}^3R_5\,{}^5\mathbf{z}_5 = \begin{bmatrix} c\theta_4 s\theta_5 \\ s\theta_4 s\theta_5 \\ c\theta_5 \end{bmatrix}, \quad \mathbf{s}_{o,6} = \mathbf{s}_{o,5} + {}^3R_5\,{}^5r_5 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$



Matlab Program: Jacobian_screw_stanford.m

Screw-Based Jacobians

✓ Example 3: Stanford Arm (2RP3R)



- Denote $\dot{\boldsymbol{q}} = [\dot{\theta}_1, \dots, \dot{\theta}_6]^T$ and $\dot{\boldsymbol{\chi}} = [\boldsymbol{\omega}_E, \boldsymbol{v}_P]^T$

  Find the 3rd, 2nd and 1st axes details. For $i = 2$:

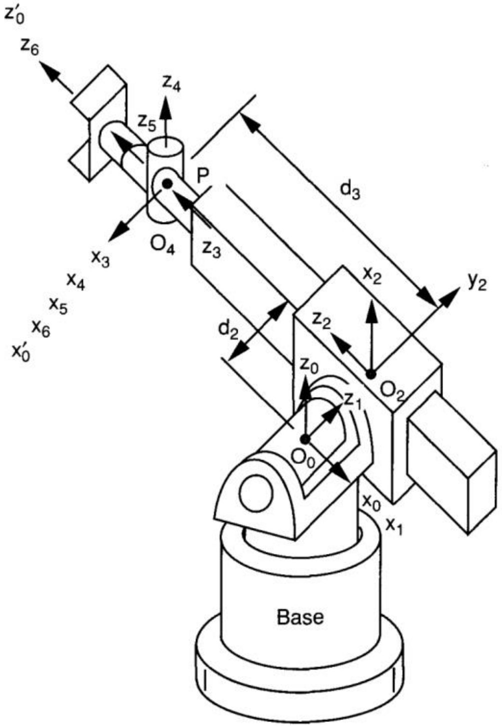
$$\mathbf{s}_3 = {}^3R_2\,{}^2\mathbf{z}_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{s}_{o,3} = \mathbf{s}_{o,4} - {}^3R_3\,{}^3r_3 = \begin{bmatrix} 0 \\ 0 \\ -d_3 \end{bmatrix},$$

$${}^3R_1 = {}^3R_2\,{}^2R_1 = \begin{bmatrix} 0 & 0 & -1 \\ c\theta_2 & s\theta_2 & 0 \\ s\theta_2 & -c\theta_2 & 0 \end{bmatrix}.$$

For $i = 1$:

$$\mathbf{s}_2 = {}^3R_1\,{}^1\mathbf{z}_1 = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{s}_{o,2} = \mathbf{s}_{o,3} - {}^3R_2\,{}^2r_2 = \begin{bmatrix} d_2 \\ 0 \\ -d_3 \end{bmatrix},$$

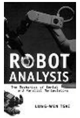
$${}^3R_0 = {}^3R_1\,{}^1R_0 = \begin{bmatrix} s\theta_1 & -c\theta_1 & 0 \\ c\theta_1 c\theta_2 & s\theta_1 c\theta_2 & -s\theta_2 \\ c\theta_1 s\theta_2 & s\theta_1 s\theta_2 & c\theta_2 \end{bmatrix}.$$



Matlab Program: Jacobian_screw_stanford.m

Screw-Based Jacobians

✓ Example 3: Stanford Arm (2RP3R)



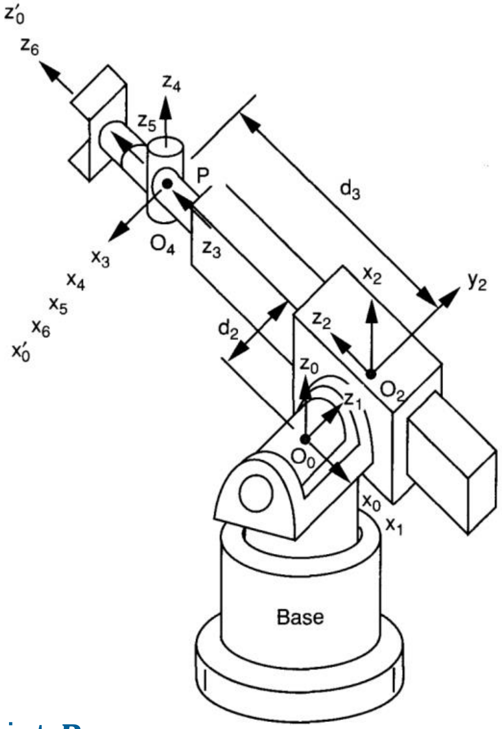
- Denote $\dot{\boldsymbol{q}} = [\dot{\theta}_1, \dots, \dot{\theta}_6]^T$ and $\dot{\boldsymbol{\chi}} = [\boldsymbol{\omega}_E, \boldsymbol{v}_P]^T$

For $i = 0$:

$$\mathbf{s}_1 = {}^3R_0\,{}^0\mathbf{z}_0 = \begin{bmatrix} 0 \\ -s\theta_2 \\ c\theta_2 \end{bmatrix}, \quad \mathbf{s}_{o,1} = \mathbf{s}_{o,2} - {}^3R_1\,{}^1r_1 = \begin{bmatrix} d_2 \\ 0 \\ -d_3 \end{bmatrix}.$$

Hence $\dot{\boldsymbol{\chi}} = \boldsymbol{J}\,\dot{\boldsymbol{q}}$ in reference frame $\{3\}$ is given by:

$${}^3J = \begin{bmatrix} 0 & -1 & 0 & 0 & -s\theta_4 & c\theta_4 s\theta_5 \\ -s\theta_2 & 0 & 0 & 0 & c\theta_4 & s\theta_4 s\theta_5 \\ c\theta_2 & 0 & 0 & 1 & 0 & c\theta_5 \\ -d_3 s\theta_2 & 0 & 0 & 0 & 0 & 0 \\ -d_2 c\theta_2 & d_3 & 0 & 0 & 0 & 0 \\ -d_2 s\theta_2 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$



Observe that the Jacobian is greatly simplified for the wrist point $\boldsymbol{P}$.

Matlab Program: Jacobian_screw_stanford.m

Contents

In this chapter we review the Jacobian analysis for serial robots. First the definition to angular and linear velocities are given, then the Jacobian matrix is defined in conventional and screw-based representation, while their general and iterative derivation methods are given. Next the static wrench and its relation to Jacobian transpose is introduced, and Jacobian characteristics such as singularity, isotropy, dexterity and manipulability are elaborated. Inverse Jacobian solution for fully-, under- and redundantly-actuator robots are formulated, and redundancy resolution schemes are detailed. Finally, Stiffness analysis of robotic manipulators is reviewed in detail.

Static Wrench

- Applied Wrench to the Environment
  - ✓ How much actuator effort is needed to apply such forces/Moments
    - Define the actuator torque/force

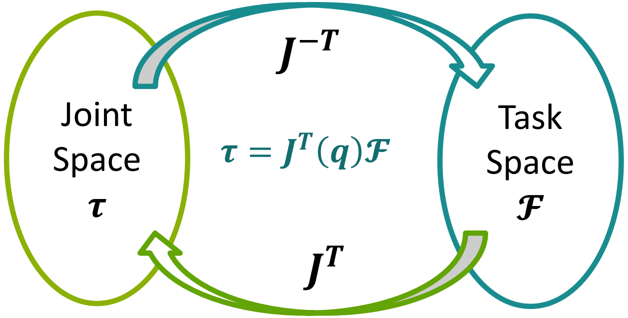
$$\boldsymbol{\tau} = [\tau_1, \tau_2, \dots, \tau_n]^T \text{ in which } \tau_i = \begin{cases} \tau_i & \text{for a revolute joint} \\ f_i & \text{for a prismatic joint} \end{cases}$$

    - Define the applied wrench to the environment: (6×1) tuple

$$\boldsymbol{\mathcal{F}} = [\boldsymbol{F}_E, \boldsymbol{n}_E]^T \text{ in which } \begin{cases} \boldsymbol{F}_E = [F_x \quad F_y \quad F_z]^T & \text{The applied force} \\ \boldsymbol{n}_E = [n_x \quad n_y \quad n_z]^T & \text{The applied torque} \end{cases}$$

Wrench is a screw-based coordinate as twist

    - Jacobian transpose maps the joint space variables to task space by:



Static Wrench

- Principle of Virtual Work
  - ✓ Virtual Displacement
    - Infinitesimal change in the position and orientation $\delta\boldsymbol{q}$ or $\delta\boldsymbol{\chi}$

      Which does not really change the posture and force distribution in the robot.

      $\delta\boldsymbol{q} = [\delta q_1, \delta q_2, \dots \delta q_n]^T$: The virtual displacement of the joint variables

      $\delta\boldsymbol{\chi} = \left[\delta x, \delta y, \delta z, \delta\theta_x, \delta\theta_y, \delta\theta_z\right]^T$: The virtual displacement of the end effector, where $\left[\delta\theta_x, \delta\theta_y, \delta\theta_z\right]^T = \delta\theta\hat{\boldsymbol{s}}$ is the orientation variable in screw representation.
  - ✓ System Under Static Balance
    - The total virtual work, δW, done by all the actuators and external forces is equal to zero.

      $$\delta W = \boldsymbol{\tau}^T\delta\boldsymbol{q} - \boldsymbol{\mathcal{F}}^T\delta\boldsymbol{\chi} = 0.$$

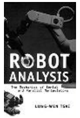
      where, $-\boldsymbol{\mathcal{F}}^T$ is used in here, to include the wrench applied to the robot by environment.
    - Jacobian maps: $\dot{\boldsymbol{\chi}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}$ therefore, $\delta\boldsymbol{\chi} = \boldsymbol{J}(\boldsymbol{q})\,\delta\boldsymbol{q}$.
    - Substitute $\quad \delta W = \left(\boldsymbol{\tau}^T - \boldsymbol{\mathcal{F}}^T\boldsymbol{J}(\boldsymbol{q})\right)\delta\boldsymbol{q} = 0$
    - This holds for any arbitrary virtual displacement $\delta\boldsymbol{q}$; Hence

      $$\boldsymbol{\tau}^T - \boldsymbol{\mathcal{F}}^T\boldsymbol{J}(\boldsymbol{q}) = 0 \quad \text{or} \quad \boxed{\boldsymbol{\tau} = \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{\mathcal{F}}}$$

      This means $\boldsymbol{J}^T(\boldsymbol{q})$ maps the wrenches $\boldsymbol{\mathcal{F}}$ applied to the environment into the actuator torques $\boldsymbol{\tau}$

Static Wrench

✓ Example 1: Planar RRR Manipulator



- Denote $\boldsymbol{\tau} = [\tau_1, \tau_2, \tau_3]^T$: The actuator forces in the joints, and
- and $\boldsymbol{\mathcal{F}} = \left[f_x, f_y, n_z\right]^T$ the planar force $\boldsymbol{F_E} = \left[f_x, f_y\right]^T$ and n_z the torque exerted to the environment
- The Jacobian map $\boldsymbol{\tau} = \boldsymbol{J^T}(\boldsymbol{q})\boldsymbol{\mathcal{F}}$ may be used

  In which,

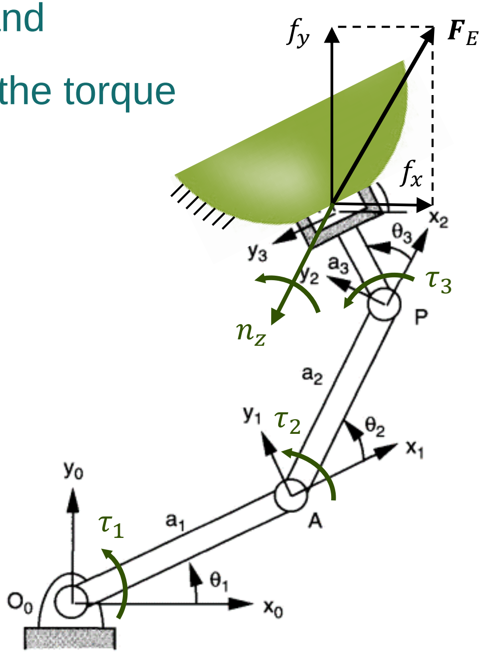
$$\boldsymbol{J}(\boldsymbol{q}) = \begin{bmatrix} -a_1 s_1 - a_2 s_{12} - a_3 s_{123} & -a_2 s_{12} - a_3 s_{123} & -a_3 s_{123} \\ a_1 c_1 + a_2 c_{12} + a_3 c_{123} & a_2 c_{12} + a_3 c_{123} & a_3 c_{123} \\ 1 & 1 & 1 \end{bmatrix}$$

- This means:

$$\tau_1 = -(a_1 s_1 + a_2 s_{12} + a_3 s_{123}) f_x + (a_1 c_1 + a_2 c_{12} + a_3 c_{123}) f_y +$$

$$\tau_2 = -(a_2 s_{12} + a_3 s_{123}) f_x + (a_2 c_{12} + a_3 c_{123}) \dot{\theta}_2 + n_z$$

$$\tau_3 = -(a_3 s_{123}) f_x + (a_3 c_{123}) f_y + n_z$$



- This may be verified by Newton-Euler free body diagram method.

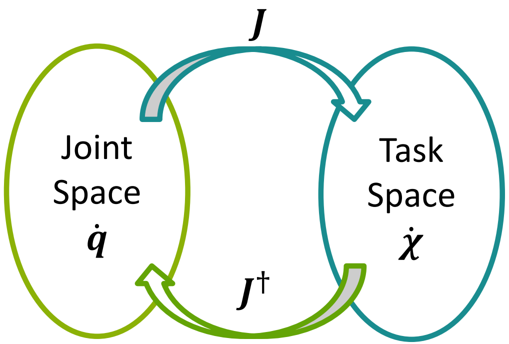
Contents

In this chapter we review the Jacobian analysis for serial robots. First the definition to angular and linear velocities are given, then the Jacobian matrix is defined in conventional and screw-based representation, while their general and iterative derivation methods are given. Next the static wrench and its relation to Jacobian transpose is introduced, and Jacobian characteristics such as singularity, isotropy, dexterity and manipulability are elaborated. Inverse Jacobian solution for fully-, under- and redundantly-actuator robots are formulated, and redundancy resolution schemes are detailed. Finally, Stiffness analysis of robotic manipulators is reviewed in detail.

Jacobian Characteristics

- Singularity
  - ✓ Jacobian reveals the forward differential kinematic map $\dot{\boldsymbol{\chi}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}$

▪ Forward Map
Given $\dot{\boldsymbol{q}}$ find $\dot{\boldsymbol{\chi}}$



Inverse Map
Given $\dot{\boldsymbol{\chi}}$ find $\dot{\boldsymbol{q}}$

▪ Consider the inverse map
For square Jacobians if $\boldsymbol{J}^{-1}(\boldsymbol{q})$ exists then $\dot{\boldsymbol{q}} = \boldsymbol{J}^{-1}(\boldsymbol{q})\dot{\boldsymbol{\chi}}$
This is used to find the required joint speeds to achieve a desired velocity in task space.

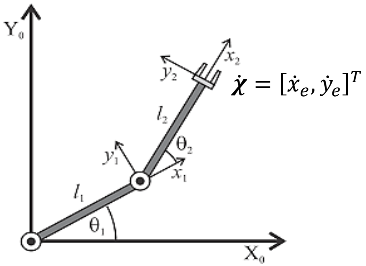
▪ At singular configurations of $\boldsymbol{J}(\boldsymbol{q})$, this matrix is not invertible ($\det(\boldsymbol{J}) = 0$).
@ singular configuration, with finite joint speeds all arbitrary task velocities are not achievable!

▪ This will happen at the boundary of the workspace, and …

Singularity

- Motivating Example
  - ✓ Consider the planar 2R manipulator
    - Denote $\dot{\boldsymbol{q}} = [\dot{\theta}_1, \dot{\theta}_2]^T$ and $\dot{\boldsymbol{\chi}} = [\dot{x}_E, \dot{y}_E]^T$

      Jacobian in frame$\{0\}$: ${}^0\boldsymbol{J} = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{bmatrix}$, and in frame $\{2\}$: ${}^2\boldsymbol{J} = \begin{bmatrix} l_1 s_2 & 0 \\ l_1 c_2 + l_2 & l_2 \end{bmatrix}$.

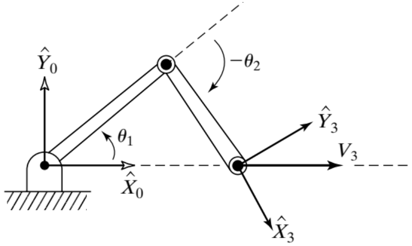


    - Singular configurations

      $$det({}^0\boldsymbol{J}) = det({}^2\boldsymbol{J}) = l_1 l_2 s_2 = 0 \text{ if } s_2 = 0 \text{ or } \theta_2 = 0 \text{ or } \pi$$

    - Physically: Fully extended or retracted arms (We saw this when one double solution for IK occurs, on the boundaries of the workspace)
    - Let us find the inverse solution:

      $$\dot{\boldsymbol{q}} = \boldsymbol{J}^{-1}(\boldsymbol{q})\dot{\boldsymbol{\chi}} \rightarrow \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = \frac{1}{l_1 l_2 s_2} \begin{bmatrix} l_2 c_{12} & l_2 s_{12} \\ -l_1 c_1 - l_2 c_{12} & -l_1 s_1 - l_2 s_{12} \end{bmatrix} \begin{bmatrix} \dot{x}_E \\ \dot{y}_E \end{bmatrix}$$



To visualize, consider $\dot{x}_E = 1$ while $\dot{y}_E = 0$ (move in x direction), then

$$\dot{\theta}_1 = \frac{1}{l_1 l_2 s_2}, \quad \dot{\theta}_2 = \frac{-1}{l_1 l_2 s_2}(l_1 c_1 + l_2 c_{12})$$

As the arms are fully extended $s_2 \rightarrow 0$, and $\dot{\theta}_1, \dot{\theta}_2 \rightarrow \infty$

At the boundary of the workspace $s_2 = 0$, no further out movement in x direction is possible.

Singularity

- General Description
  - ✓ Consider a $6 \times n$ Jacobian $\boldsymbol{J}(q)$ of a nDoF robot denoted by

$$\dot{\chi} = J_1 \dot{q}_1 + J_2 \dot{q}_2 + \cdots + J_n \dot{q}_n \text{ or } \dot{\chi} = \$_1 \dot{q}_1 + \$_2 \dot{q}_2 + \cdots + \$_n \dot{q}_n$$

Where J_i or $\$_i$ are the columns of the Jacobian matrix

Reaching high:
Fully extended neck

- Robot EE can reach any arbitrary twist if $\mathrm{rank}(\boldsymbol{J}(q)) = 6$.

  $\mathrm{rank}(\boldsymbol{J}(q))$ = No. of independent J_i or $\$_1$ (Configuration Dependent)

  For a 6DoF robot $\mathrm{rank}(\boldsymbol{J}(q)) \leq 6$ and @ q that < 6 singularity occurs

  For a 2DoF robot $\mathrm{rank}(\boldsymbol{J}(q)) \leq 2$ and @ q that < 2 singularity occurs

- At singular configurations:

  Certain direction of motion is unattainable (undesired)

  Bounded end-effector velocities → unbounded joint speeds

  Bounded joint torques → unbounded end-effector forces (desired)

  Often occurs @ boundary of workspace (where one double solution for IK occurs)

Fully extended arms to lift heavy loads

Singularity

- Singular Configurations
  - ✓ For Square Jacobians: Find q such that $\det(\boldsymbol{J}(q)) = 0$.
  - ✓ Decoupling of Singularities
    - For the case of $n = 6$: $\boldsymbol{J}(q) = [\boldsymbol{J}_{arm} \;\; | \;\; \boldsymbol{J}_{wrist}] = \begin{bmatrix} \boldsymbol{J}_{11} & \boldsymbol{J}_{12} \\ \boldsymbol{J}_{21} & \boldsymbol{J}_{22} \end{bmatrix}$
    - If wrist axes are revolute and intersect at a point then $\boldsymbol{J}_{wrist} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{J}_{22} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ \hat{\boldsymbol{s}}_4 & \hat{\boldsymbol{s}}_5 & \hat{\boldsymbol{s}}_5 \end{bmatrix}$
    - The Jacobian is upper triangular $\boldsymbol{J}(q) = \begin{bmatrix} \boldsymbol{J}_{11} & \boldsymbol{0} \\ \boldsymbol{J}_{21} & \boldsymbol{J}_{22} \end{bmatrix}$
    - Singularity occurs @ q, in which:

$$\det(\boldsymbol{J}(q)) = \det(\boldsymbol{J}_{11}(q)) \cdot \det(\boldsymbol{J}_{22}(q)) = 0$$

    - Determine Singular configuration of arm and wrist separately.

Wrist singularity occurs @ q, in which: $\det(\boldsymbol{J}_{22}(q)) = 0$

Arm singularity occurs @ q, in which: $\det(\boldsymbol{J}_{11}(q)) = 0$

Singularity

- Decoupling of Singularities
  - ✓ Wrist Singularities
    - Consider 3R intersecting wrist:

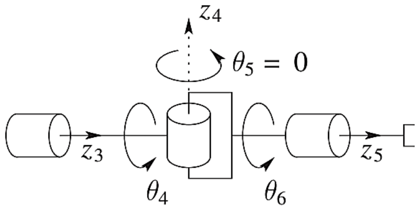
A typical industrial design is like $w-u-w$ Euler configuration.

$$\det\left(\boldsymbol{J}_{22}(q)\right)=0$$

This happens when the z_i axes are linearly dependent.

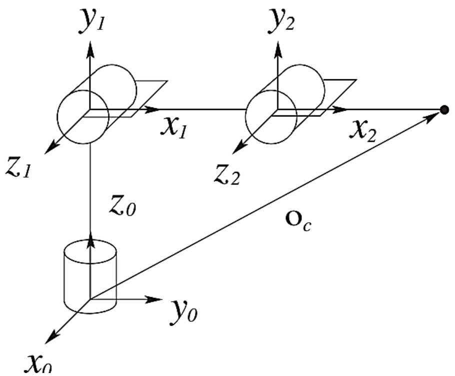
Singular configuration: when z_3 and z_5 are collinear.

Then: $\theta_5=0$ or π



  - ✓ Wrist Singularities
    - Consider 3R Elbow manipulator like design

$$J_{11}=\begin{bmatrix} -a_2s_1c_2-a_3s_1c_{23} & -a_2s_2c_1-a_3s_{23}c_1 & -a_3c_1s_{23} \\ a_2c_1c_2+a_3c_1c_{23} & -a_2s_1s_2-a_3s_1s_{23} & -a_3s_1s_{23} \\ 0 & a_2c_2+a_3c_{23} & a_3c_{23} \end{bmatrix}$$

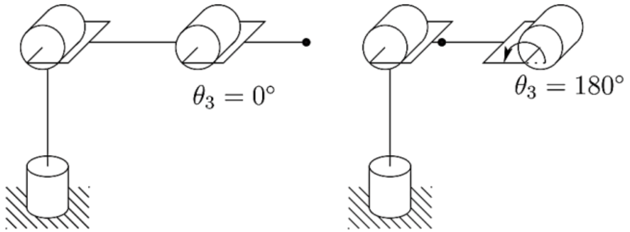


    - The determinant is:

$$\det J_{11} = -a_2a_3s_3(a_2c_2+a_3c_{23})$$

Singularity

- Decoupling of Singularities
  - ✓ Wrist Singularities
    - Consider 3R Elbow manipulator like design
    - Singular configurations: If $s_3 = 0$ or $\theta_3 = 0$ or π



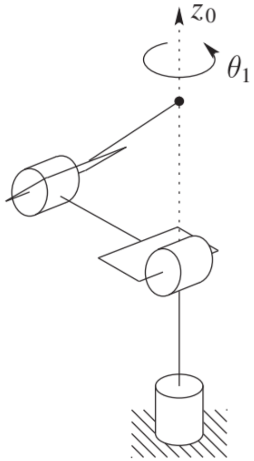
Fully extended or retracted.

- Or when $a_2 c_2 + a_3 c_{23} = 0$

The wrist point intersect the base axis

This case occurs @ infinitely many configurations

Where infinitely many solution exist for IK.



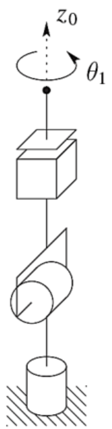
If the elbow manipulator has an offset this singular configuration vanishes.

Singularity

- Decoupling of Singularities
  - ✓ Wrist Singularities
    - Consider 2RP Spherical Manipulator with no off set

      By inspection singular configuration exists if:

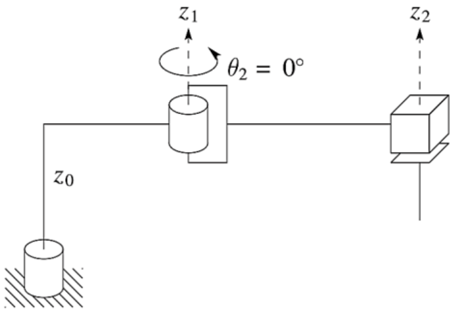
      The wrist point intersect the base axis:



    - Consider SCARA Manipulator

      The Jacobian is derived before, in which

$$J_{11} = \begin{bmatrix} \alpha_1 & \alpha_3 & 0 \\ \alpha_2 & \alpha_4 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad \text{where} \quad \begin{aligned} \alpha_1 &= -a_1 s_1 - a_2 s_{12} \\ \alpha_2 &= a_1 c_1 + a_2 c_{12} \\ \alpha_3 &= -a_1 s_{12} \\ \alpha_4 &= a_1 c_{12} \end{aligned}$$



$$\det(J_{11}) = 0 \quad \text{if} \quad \alpha_1\alpha_4 - \alpha_2\alpha_3 = 0.$$

This occurs if $s_2 = 0$, which implies $\theta_2 = 0$ or π.

This is similar to Elbow manipulator for fully extended or retracted arm.

Dexterity

- Motivation Example:
  - ✓ Cobra Attack: Optimal Posture

Dexterity

- Definition:
  - ✓ Skill in performing tasks, especially with the hands. “quickness”

Dexterity

- "Quickness" in Multi Dimensional Space?
  - ✓ Consider norm bound joint velocities (unit sphere)

$$\|\dot{\boldsymbol{q}}\|_2^2 = \dot{q}_1^2 + \dot{q}_2^2 + \cdots + \dot{q}_n^2 \leq 1$$

  - ✓ What happens to the task space velocities?

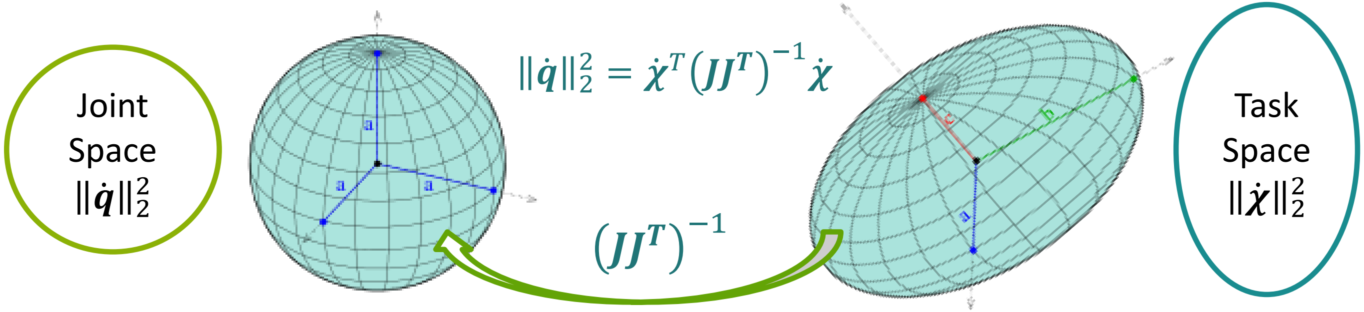
$$\|\dot{\boldsymbol{q}}\|_2^2 = \dot{\boldsymbol{q}}^T\dot{\boldsymbol{q}} = \left[\boldsymbol{J}^\dagger\dot{\boldsymbol{\chi}}\right]^T\boldsymbol{J}^\dagger\dot{\boldsymbol{\chi}} = \dot{\boldsymbol{\chi}}^T\boldsymbol{J}^{\dagger^T}\boldsymbol{J}^\dagger\dot{\boldsymbol{\chi}} = \dot{\boldsymbol{\chi}}^T\left(\boldsymbol{J}\boldsymbol{J}^T\right)^\dagger\dot{\boldsymbol{\chi}}$$

For all fat, square or tall Jacobians: $\boldsymbol{J}\boldsymbol{J}^T$ is a 6×6 matrix, hence

$$\|\dot{\boldsymbol{q}}\|_2^2 = \dot{\boldsymbol{\chi}}^T\left(\boldsymbol{J}\boldsymbol{J}^T\right)^{-1}\dot{\boldsymbol{\chi}}$$

    - This result into Dexterity or Manipulability Ellipsoid

for a uniform input $\|\dot{\boldsymbol{q}}\|_2^2 = 1$, the output task velocities shall have a weighted norm along this ellipsoid



Isotropy

- Eigenvalues and Eigenvectors
  - ✓ The ellipsoid is characterized by its eigen parameters

$$\det(\boldsymbol{J}\boldsymbol{J}^T) = \lambda_1 \cdot \lambda_2 \cdots \lambda_n$$

  - ✓ Two extreme cases:
    - Singularity:

      $\exists\ \lambda_i = 0 \ \rightarrow\ \det(\boldsymbol{J}\boldsymbol{J}^T) = \lambda_1 \cdot \lambda_2 \cdots \lambda_n = 0$

      The ellipsoid is changed to a cylinder in v_i (eigenvector) direction.

      There exist no finite joint velocities to reach to task velocities in v_i direction

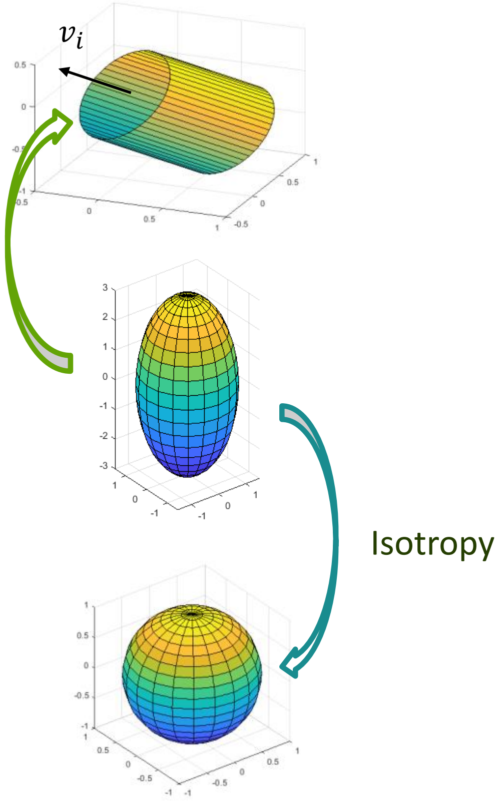
    - Isotropy:

      $\forall\ \lambda_i = 1 \ \rightarrow \boldsymbol{J}\boldsymbol{J}^T = \boldsymbol{I}$ unit matrix $\ \rightarrow \det(\boldsymbol{J}\boldsymbol{J}^T) = 1$

      The ellipsoid is changed to a sphere

      Dexterity in all task space direction with finite joint velocities

      Isotropy in applying equal velocities in all directions



Manipulability

- Gain of the Velocity Map
  - ✓ Applying a uniform and unit norm input $\|\dot{\boldsymbol{q}}\|_2^2 = 1$
    - Gain of the output is given by

$$\det(\boldsymbol{J}\boldsymbol{J}^T) = \lambda_1 \cdot \lambda_2 \cdots \lambda_n$$

Where λ_i denote the eigenvalue of $\boldsymbol{J}\boldsymbol{J}^T$

    - Singular value of $\boldsymbol{J}$?

For a general even non-square matrix $\boldsymbol{J}$

$$\sigma_i(\boldsymbol{J}) = \sqrt{\lambda_i(\boldsymbol{J}\boldsymbol{J}^T)}$$

Where σ_i denotes the singular value of matrix $\boldsymbol{J}$

    - Manipulability Measure μ of $(\boldsymbol{J}(q))$

$$\mu = \sqrt{\det(\boldsymbol{J}\boldsymbol{J}^T)} = \sqrt{\lambda_1 \cdot \lambda_2 \cdots \lambda_n} = \sigma_1 \cdot \sigma_2 \cdots \sigma_n \text{ singular values of } (\boldsymbol{J})$$

The measure is configuration dependent.

If $\mu \to 0$ the configuration of the robot tends to singularity.

If $\mu \to 1$ the configuration of the robot tends to isotropy.

$\mu = 0$, if and only if $\mathrm{rank}(\boldsymbol{J}) < n$, the DoF's of the robot.

Manipulability

- Gain of the Velocity Map
  - ✓ Manipulability Measure μ of $(\boldsymbol{J}(q))$
    - If the robot is under actuated it is deficient and $\mu = 0$.
    - If the robot is redundantly actuated, there are extra σ_i's than Dof's

      This could be used to increase μ (Biological designs!)

      Reconsider Cobra attack, the snake uses redundant posture for the attack.
    - In general

$$\|\dot{\boldsymbol{q}}\|_2^2 = \dot{\boldsymbol{\chi}}^T\left(\boldsymbol{J}\boldsymbol{J}^T\right)^{-1}\dot{\boldsymbol{\chi}} \leq \frac{1}{\|\boldsymbol{J}\boldsymbol{J}^T\|}\|\dot{\boldsymbol{\chi}}\|_2^2 = \frac{1}{\mu^2}\|\dot{\boldsymbol{\chi}}\|_2^2 \rightarrow \|\dot{\boldsymbol{\chi}}\| \geq \mu\|\dot{\boldsymbol{q}}\|$$

$$\max\|\dot{\boldsymbol{\chi}}\| = \sigma_{max}\|\dot{\boldsymbol{q}}\| \ \text{ in direction of } v_{max}$$
$$\min\|\dot{\boldsymbol{\chi}}\| = \sigma_{min}\|\dot{\boldsymbol{q}}\| \ \text{ in direction of } v_{min}$$

μ denotes the gain required to generate a specific task space velocity

If $\mu = 0$ some velocity directions are not attainable.

If $\mu = 1$ the uniform input is projected uniformly in all directions of the outputs.

The shape of the ellipsoid is also very informative on the attainable directions.

Manipulability

- Other Measures
  - ✓ Reciprocal of Condition number of $(\boldsymbol{J}(q))$
    - Definition: Condition number of a matrix $\boldsymbol{J}$ is $\kappa(\boldsymbol{J}) = \frac{\sigma_{max}(\boldsymbol{J})}{\sigma_{min}(\boldsymbol{J})}$

      In which $\sigma_{max}(\boldsymbol{J})$, and $\sigma_{min}(\boldsymbol{J})$ denotes the largest and smallest singular value of $\boldsymbol{J}$, respectively.
    - The measure: $\mathrm{rcond}(\mathbf{J}) = 1/\kappa(\boldsymbol{J}) = \frac{\sigma_{min}(\boldsymbol{J})}{\sigma_{max}(\boldsymbol{J})}$
    - If $\mathrm{rcond} = 0$: at least one of the singular values are zero: singular configuration
    - If $\mathrm{rcond} = 1$: All singular values are one: isotropic configuration

      μ considers all the singular values by rcond only extreme values

      The analysis are similar but the ellipsoids are not analyzed in rcond.
  - ✓ Global Measures
    - All measures are configuration dependent

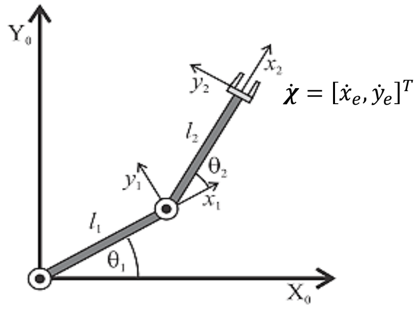
      There could be good at a pose and bad at another.

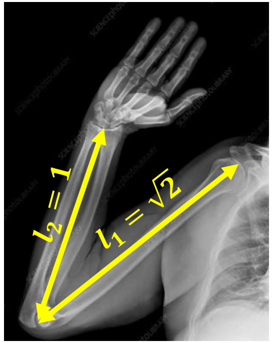
      Integrate the measure in the whole space to get an averaged global measure

Manipulability

- Examples:
  - ✓ Example 1: Isotropy Analysis of 2R Robot
    - In 2R manipulator:

$$\dot{\boldsymbol{\chi}} = \boldsymbol{J}\,\dot{\boldsymbol{q}}\ \text{, in which } \boldsymbol{J} \text{ in frame } \{2\} \text{ is } {}^{2}\boldsymbol{J} = \begin{bmatrix} l_1 s_2 & 0 \\ l_1 c_2 + l_2 & l_2 \end{bmatrix}$$



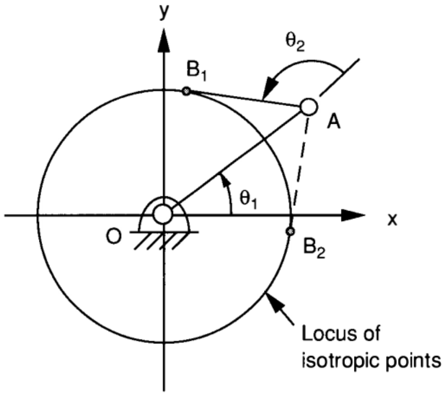


    - Inspired by human arm, consider $l_1 = \sqrt{2}$, and $l_2 = 1$.

Use symbolic manipulator to find $\boldsymbol{J}\boldsymbol{J}^T$ and its eigenvalues:

$$\boldsymbol{J}\boldsymbol{J}^T = \begin{bmatrix} 2s_2^2 & 2c_2 s_2 + \sqrt{2}s_2 \\ 2c_2 s_2 + \sqrt{2}s_2 & \left(2\sqrt{2}c_2 + 1\right)^2 + 1 \end{bmatrix}; \quad \det(\boldsymbol{J}\boldsymbol{J}^T) = 2s_2^2,$$

$$\lambda_{1,2} = \pm\left(4c_2^2 + 4\sqrt{2}\,c_2 + 2\right)^{1/2} + \sqrt{2}c_2 + 2$$



In Isotropic configurations $\lambda_1 = \lambda_2 = 1$.

Only for this bio-inspired design isotropy happens @ $\theta_2 = \pm\frac{3\pi}{4}\ \forall\theta_1$.

The locus of isotropic configurations are shown in figure.

While singularity happens at fully-extended or retracted arm!

Manipulability

✓ Example 1: (Cont.)

- Now consider $\theta_2 = \frac{\pi}{2}$; for this configuration:

$$\boldsymbol{J}\boldsymbol{J}^{\boldsymbol{T}}(\theta_2 = \tfrac{\pi}{2}) = \begin{bmatrix} 2 & \sqrt{2} \\ \sqrt{2} & 2 \end{bmatrix}; \quad \left(\boldsymbol{J}\boldsymbol{J}^{\boldsymbol{T}}\right)^{-\mathbf{1}} = \begin{bmatrix} 1 & -1/\sqrt{2} \\ -1/\sqrt{2} & 1 \end{bmatrix}$$

Calculate the velocity map gains

For $\boldsymbol{J}\boldsymbol{J}^{\boldsymbol{T}} \rightarrow \begin{matrix} \lambda_1 = 3.414 \\ \lambda_2 = 0.586 \end{matrix}$; while $v_1 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix}, v_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} -1 \\ 1 \end{bmatrix},$

then for $\left(\boldsymbol{J}\boldsymbol{J}^{\boldsymbol{T}}\right)^{-\mathbf{1}} \rightarrow \begin{matrix} \lambda_1 = 0.292 \\ \lambda_2 = 1.707 \end{matrix}$

For $\boldsymbol{J}$; $\begin{matrix} G_{max} = \sigma_{max} = 1.847 \\ G_{min} = \sigma_{min} = 0.734 \end{matrix}$; while $\vec{v}_{max} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \vec{v}_{min} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$

The gains are singular values of $\boldsymbol{J}$.

Directions are found by eigenvalues of $\boldsymbol{J}\boldsymbol{J}^{\boldsymbol{T}}$

Manipulability

✓ Example 1: (Cont.)

- Proof of the mapping gains for $\theta_2 = \frac{\pi}{2}$ configuration:

From $\|\dot{\boldsymbol{q}}\|_2^2 = \dot{\boldsymbol{\chi}}^T \left(\boldsymbol{J}\boldsymbol{J}^T\right)^{-1} \dot{\boldsymbol{\chi}} \rightarrow \dot{\theta}_1^2 + \dot{\theta}_2^2 = v_x^2 - \sqrt{2}\, v_x v_y + v_y^2$

$$\dot{\theta}_1^2 + \dot{\theta}_2^2 = 0.292 \left(\frac{v_x}{\sqrt{2}} + \frac{v_y}{\sqrt{2}}\right)^2 + 1.707 \left(\frac{v_x}{\sqrt{2}} - \frac{v_y}{\sqrt{2}}\right)^2$$

For $v_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, v_x = v_y \rightarrow \dot{\theta}_1^2 + \dot{\theta}_2^2 = 0.292\left(\sqrt{2}\, v_x\right)^2 \rightarrow \left(\sqrt{2}\, v_x\right)^2 = 3.414\left(\dot{\theta}_1^2 + \dot{\theta}_2^2\right)$

$\Rightarrow \dot{\chi}_{max} = 1.848\|\dot{\boldsymbol{q}}\|$ in $\vec{v}_{max} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ direction

For $v_1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, v_x = -v_y \rightarrow \dot{\theta}_1^2 + \dot{\theta}_2^2 = 1.707\left(\sqrt{2}\, v_x\right)^2 \rightarrow \left(\sqrt{2}\, v_x\right)^2 = 0.586 \left(\dot{\theta}_1^2 + \dot{\theta}_2^2\right)$

$\Rightarrow \dot{\chi}_{min} = 0.734 \|\dot{\boldsymbol{q}}\|$ in $\vec{v}_{min} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$direction

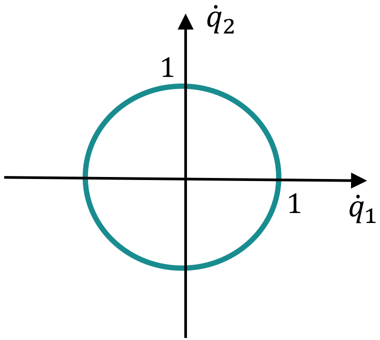
Manipulability

✓ Example 1: (Cont.)

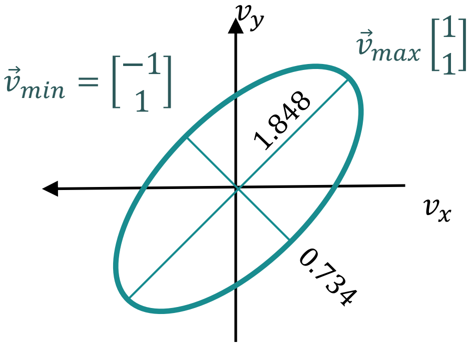
- For $\theta_2 = \frac{\pi}{2}$ configuration:

$$\Rightarrow \dot{\chi}_{max} = 1.848\,\|\dot{\boldsymbol{q}}\| = \sigma_{max}\,\|\dot{\boldsymbol{q}}\| \text{ in } \vec{v}_{max} \text{ direction}$$

$$\Rightarrow \dot{\chi}_{min} = 0.734\|\dot{\boldsymbol{q}}\| = \sigma_{min}\,\|\dot{\boldsymbol{q}}\| \text{ in } \vec{v}_{min} \text{ direction}$$



$$\|\dot{\boldsymbol{q}}\|_2^2 = \dot{\boldsymbol{\chi}}^T\left(\boldsymbol{J}\boldsymbol{J}^T\right)^{-1}\dot{\boldsymbol{\chi}}$$



- Dexterity Measures: $\mu = \sqrt{\det(\boldsymbol{J}\boldsymbol{J}^T)} = \sqrt{\lambda_1 \cdot \lambda_2} = \sqrt{2}$, $\text{rcond} = \frac{\sigma_{min}}{\sigma_{max}} = \frac{0.764}{1.848} = 0.414$

Contents

In this chapter we review the Jacobian analysis for serial robots. First the definition to angular and linear velocities are given, then the Jacobian matrix is defined in conventional and screw-based representation, while their general and iterative derivation methods are given. Next the static wrench and its relation to Jacobian transpose is introduced, and Jacobian characteristics such as singularity, isotropy, dexterity and manipulability are elaborated. Inverse Jacobian solution for fully-, under- and redundantly-actuator robots are formulated, and redundancy resolution schemes are detailed. Finally, Stiffness analysis of robotic manipulators is reviewed in detail.

Inverse Solutions

- Definition:
  - ✓ Jacobian Forward Map $\dot{\boldsymbol{\chi}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}$ or $\boldsymbol{\tau} = \boldsymbol{J}^{\boldsymbol{T}}(\boldsymbol{q})\boldsymbol{\mathcal{F}}$
  - ✓ Inverse Solution for Fully Actuated Robot $(m = n = 6)$
    - Jacobian matrix is square:

      In non-singular configurations $\boldsymbol{q}$, where $\boldsymbol{J}^{-\boldsymbol{1}}(\boldsymbol{q})$ exists:

      $$\dot{\boldsymbol{q}} = \boldsymbol{J}^{-\boldsymbol{1}}(\boldsymbol{q})\dot{\boldsymbol{\chi}} \quad \text{or} \quad \boldsymbol{\mathcal{F}} = \boldsymbol{J}^{-\boldsymbol{T}}(\boldsymbol{q})\,\boldsymbol{\tau}$$
    - Near singular configurations:

      To achieve a finite velocity $\dot{\boldsymbol{\chi}}$ very large joint velocities is required $\dot{\boldsymbol{q}} \rightarrow \infty$.

      Very large forces could be applied to the environment with low actuator torques
  - ✓ Inverse Solution for Under Actuated Robot $(m < 6)$
    - Jacobian matrix is tall rectangular $(6 \times m)$:

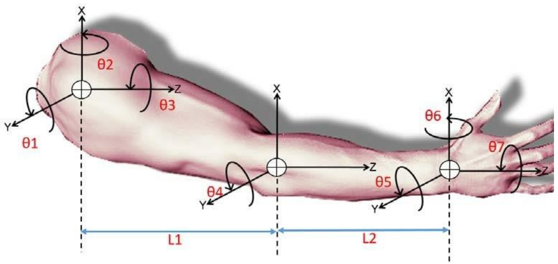
      Solution exist only if $\dot{\boldsymbol{\chi}}$ lies in the range space of $\boldsymbol{J}(\boldsymbol{q})$ or $\boldsymbol{\tau}$ lies in the range space of $\boldsymbol{J}^{\boldsymbol{T}}(\boldsymbol{q})$

      This is satisfied if $\operatorname{rank} \boldsymbol{J}(\boldsymbol{q}) = \operatorname{rank}\,[\boldsymbol{J}(\boldsymbol{q}) \mid \dot{\boldsymbol{\chi}}]$
    - The solution is found by left pseudo inverse of $\boldsymbol{J}(\boldsymbol{q})$

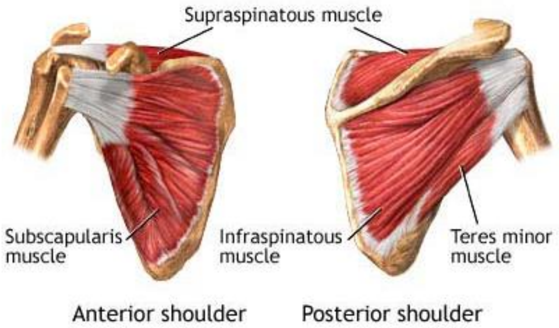
      $$\dot{\boldsymbol{q}} = \boldsymbol{J}^{\dagger}(\boldsymbol{q})\dot{\boldsymbol{\chi}} \quad \text{where} \quad \boldsymbol{J}^{\dagger} = \left(\boldsymbol{J}^{\boldsymbol{T}}\boldsymbol{J}\right)^{-\boldsymbol{1}}\boldsymbol{J}^{\boldsymbol{T}} \quad \text{Note: } (\boldsymbol{J}^{\boldsymbol{T}}\boldsymbol{J}) \text{ is } m \times m:$$

Redundancy in Nature

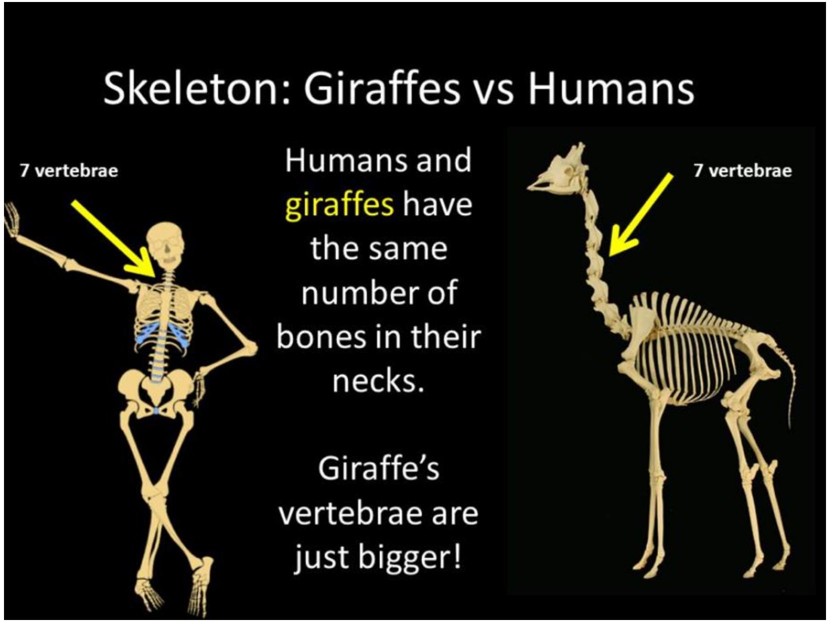
Human Arm: 7 Joints



Human Shoulder: 4 Muscles

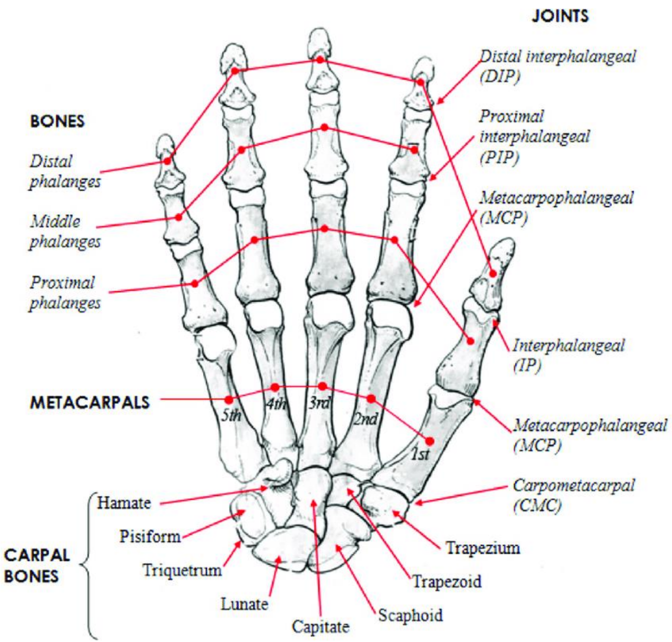


Mammal's Neck: 7 Vertebra

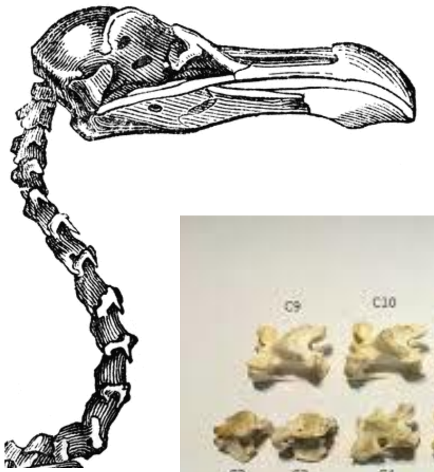


Redundancy in Nature

Human wrist: 8 Bones



Bird's Neck: 14 Vertebra!



Inverse Solution

- ✓ Inverse Solution for Redundantly Actuated Robot $(m > 6)$
  - Jacobian matrix is fat rectangular $(6 \times m)$:

    Infinitely many solution exists for the inverse problem

    Basic solution is found by min-norm or least-squares solution:

    Find $\dot{\boldsymbol{q}}$ such that $\dot{\boldsymbol{\chi}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}$ while $\|\dot{\boldsymbol{q}}\|_2$ is minimized
  - The solution is found by right pseudo inverse of $\boldsymbol{J}(\boldsymbol{q})$

    $\dot{\boldsymbol{q}}_{\boldsymbol{LS}} = \boldsymbol{J}^{\dagger}(\boldsymbol{q})\dot{\boldsymbol{\chi}}$ where $\boldsymbol{J}^{\dagger} = \boldsymbol{J}^{\boldsymbol{T}}\left(\boldsymbol{J}\boldsymbol{J}^{\boldsymbol{T}}\right)^{-\boldsymbol{1}}$ Note: $(\boldsymbol{J}^{\boldsymbol{T}}\boldsymbol{J})$ is 6×6
  - Right pseudo inverse properties:

    $$\boldsymbol{J}\boldsymbol{J}^{\dagger} = \boldsymbol{J}\boldsymbol{J}^{\boldsymbol{T}}\left(\boldsymbol{J}\boldsymbol{J}^{\boldsymbol{T}}\right)^{-\boldsymbol{1}} = \boldsymbol{I}$$
  - Set of all solutions:

    $$\dot{\boldsymbol{q}} = \boldsymbol{J}^{\dagger}(\boldsymbol{q})\dot{\boldsymbol{\chi}} + \left(\boldsymbol{I} - \boldsymbol{J}^{\dagger}\boldsymbol{J}\right)\boldsymbol{b}$$

    In which, $\boldsymbol{b} \in \mathbb{R}^n$ is any arbitrary vector, and $\left(\boldsymbol{I} - \boldsymbol{J}^{\dagger}\boldsymbol{J}\right) \neq \boldsymbol{0}$.

    All vectors in the form of $\dot{\boldsymbol{q}}_n = \left(\boldsymbol{I} - \boldsymbol{J}^{\dagger}\boldsymbol{J}\right)\boldsymbol{b}$ lie in the null space of $\boldsymbol{J}$: $\mathcal{N}(\boldsymbol{J})$

    $\dot{\boldsymbol{q}}_n \neq \boldsymbol{0}$ but the corresponding task space velocity $\dot{\boldsymbol{\chi}}_n = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}_n = \boldsymbol{0}$ (self –motion)

Inverse Solution

✓ Inverse Solution for Redundantly Actuated Robot $(m > 6)$

- LS solution is always a suitable alternative;
- Redundancy Resolution

Finding suitable $\dot{\boldsymbol{q}}_n$ to accomplish some other objectives

1. Obstacle Avoidance
2. Singularity Circumvention
3. Increasing Dexterity
4. Maximizing Manipulability
5. Limited Actuator Torques

A combined objective

Inverse Solution

✓ Inverse Solution for Redundantly Actuated Robot $(m > 6)$

- Optimization Problem

  Define a cost function to be minimized by redundancy resolution: $V(\boldsymbol{q}, \dot{\boldsymbol{q}})$ or $V(\chi, \dot{\chi})$ (e.g. $\|\dot{\boldsymbol{q}}\|_2$)

  Consider Jacobian mapping as an equality constraint: $\dot{\boldsymbol{\chi}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}$

  Consider Forward kinematics as a nonlinear equality constraint: $\boldsymbol{\chi} = \boldsymbol{f}_{FK}(\boldsymbol{q})$

  Consider joint limits as inequality constraints: $\boldsymbol{q}_{min} \leq \boldsymbol{q} \leq \boldsymbol{q}_{max}$ and/or $\dot{\boldsymbol{q}}_{min} \leq \dot{\boldsymbol{q}} \leq \dot{\boldsymbol{q}}_{max}$

$$\min_{\boldsymbol{q}, \dot{\boldsymbol{q}}} V(\boldsymbol{q}, \dot{\boldsymbol{q}})$$

$$\text{Subject to} \begin{cases} \dot{\boldsymbol{\chi}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \\ \boldsymbol{\chi} = \boldsymbol{f}_{FK}(\boldsymbol{q}) \\ \boldsymbol{q}_{min} \leq \boldsymbol{q} \leq \boldsymbol{q}_{max} \\ \dot{\boldsymbol{q}}_{min} \leq \dot{\boldsymbol{q}} \leq \dot{\boldsymbol{q}}_{max} \\ \vdots \end{cases}$$

- Analytical Solutions: Lagrange and KKT Multipliers
- Numerical Solutions : Interior Point Method “fmincon” in Matlab

  Genetic Algorithms , …

Inverse Solution

- ✓ Inverse Solution for Redundantly Actuated Robot $(m > 6)$
  - Numerical Solution: `fmincon` function in Matlab

    Nonlinear Programming Solver

$$\min_{x} f(x) \text{ such that } \begin{cases} c(x) \le 0 \\ ceq(x) = 0 \\ A \cdot x \le b \\ Aeq \cdot x = beq \\ lb \le x \le ub, \end{cases}$$

Syntax

```
x = fmincon(fun,x0,A,b)
x = fmincon(fun,x0,A,b,Aeq,beq)
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub)
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon)
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,options)
```

All Algorithms	
`Algorithm`	Choose the optimization algorithm: • `'interior-point'` (default) • `'trust-region-reflective'` • `'sqp'` • `'sqp-legacy'` (`optimoptions` only) • `'active-set'`

Inverse Acceleration

✓ Differentiate Jacobian Forward Map $\dot{\boldsymbol{\chi}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}$

$$\ddot{\boldsymbol{\chi}} = \boldsymbol{J}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}}$$

- Inverse Solution for Fully Actuated Robot $(m = n = 6)$

$$\boldsymbol{J}(\boldsymbol{q})\ddot{\boldsymbol{q}} = \ddot{\boldsymbol{\chi}} - \dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}}$$

- Jacobian matrix is square:

In non-singular configurations $\boldsymbol{q}$, where $\boldsymbol{J}^{-1}(\boldsymbol{q})$ exists:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{-1}(\boldsymbol{q})\dot{\boldsymbol{\chi}}$$

This results in:

$$\ddot{\boldsymbol{q}} = \boldsymbol{J}^{-1}(\boldsymbol{q})\left[\ddot{\boldsymbol{\chi}} - \dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}}\right] \quad \text{OR}$$

$$\ddot{\boldsymbol{q}} = \boldsymbol{J}^{-1}(\boldsymbol{q})\left[\ddot{\boldsymbol{\chi}} - \dot{\boldsymbol{J}}(\boldsymbol{q})\boldsymbol{J}^{-1}(\boldsymbol{q})\dot{\boldsymbol{\chi}}\right]$$

For non-square Jacobians such manipulation is not possible.

Obstacle Avoidance

- Example 2:
  - ✓ Redundancy Resolution for 3R Robot:
    - Consider a desired vertical motion (2D)

      From $\boldsymbol{q}_0 = [20^o, 30^o, 20^o]^T \rightarrow Q_0 \cong [x_0, y_0]^T$ To $Q_f \cong [x_0, 0]^T$

      Avoiding obstacle shown in the figure:
    - In task space:

      Move $\boldsymbol{\chi}_0 = [x_0, y_0]^T \cong [1.7, 1.4]^T$ To $\boldsymbol{\chi}_f = [x_0, 0]^T$ in one seconds.

      Use cubic trajectory planning:

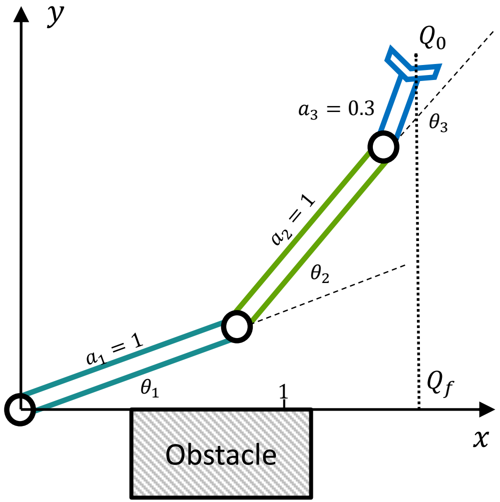
      $$x_d(t) = x_0, \qquad y_d(t) = y_0(1 - (3 - 2t)t^2)$$
    - In velocity space given:

      $$\dot{x}_d(t) = 0, \qquad \dot{y}_d(t) = y_0(6t^2 - 6t)$$

      Find $\boldsymbol{q}(t) = [q_1, q_2, q_3]^T$ to traverse this path while avoiding obstacle.
    - Robot has 3Dof, and for this task has one degree of redundancy

      To move along $\boldsymbol{\chi}_{2\times1}(t)$ there exist infinite number of joint space solutions $\boldsymbol{q}_{3\times1}(t)$

      Find the one to avoid interfering with the obstacle.



Obstacle Avoidance

- Example 2: (Cont.)
  - ✓ Redundancy Resolution for 3R Robot:
    - The Jacobin is (2×3), the LS solution is not good

      Robot shall go to elbow-up posture to avoid the obstacle
    - Formulate an optimization problem

$$\min_{\boldsymbol{q},\dot{\boldsymbol{q}}} V(\boldsymbol{q},\dot{\boldsymbol{q}})$$

$$\text{Subject to} \begin{cases} \dot{\boldsymbol{\chi}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \\ \boldsymbol{\chi} = \boldsymbol{f}_{\boldsymbol{FK}}(\boldsymbol{q}) \end{cases}$$

Robot desired posture (elbow up) $\boldsymbol{q}_d = [45^o, -70^o, -10^o]^T$

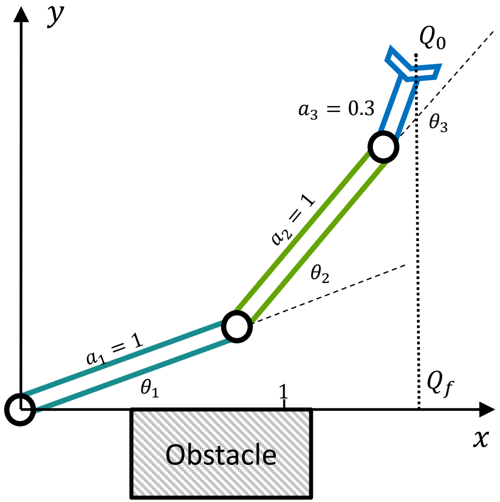
Performance index: minimize $V(\boldsymbol{q}) = \|\boldsymbol{q} - \boldsymbol{q}_d\|_2$

Robot Jacobian $\dot{\boldsymbol{\chi}}(t) = \boldsymbol{J}\dot{\boldsymbol{q}}(t)$, where

$$\boldsymbol{J} = \begin{bmatrix} -a_1 s_1 - a_2 s_{12} - a_3 s_{123} & -a_2 s_{12} - a_3 s_{123} & -a_3 s_{123} \\ a_1 c_1 + a_2 c_{12} + a_3 c_{123} & a_2 c_{12} + a_3 c_{123} & a_3 c_{123} \end{bmatrix}$$

Robot Forward Kinematics $\boldsymbol{\chi} = \boldsymbol{f}_{\boldsymbol{FK}}(\boldsymbol{q})$:

$$\chi_1 = a_1 c_1 + a_2 c_{12} + a_3 c_{123}$$
$$\chi_2 = a_1 s_1 + a_2 s_{12} + a_3 s_{123}$$



Obstacle Avoidance

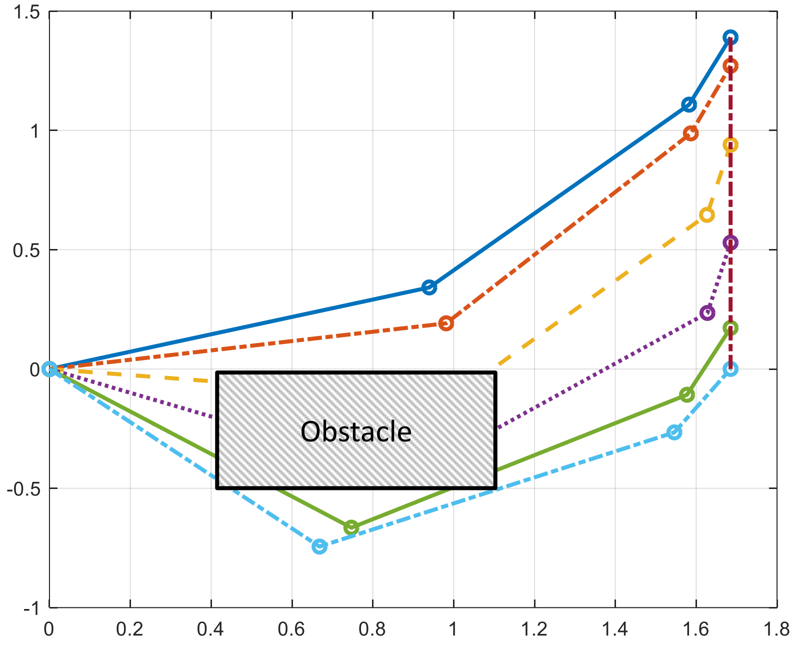
- Example 2: (Cont.)
  - ✓ Redundancy Resolution for 3R Robot:
    - Consider the base Solution:

$$\dot{q}_{LS} = J^{\dagger}(q)\dot{\chi} \quad \text{where} \quad J^{\dagger} = J^T\left(JJ^T\right)^{-1}$$

    - This solution minimizes

$$\left\| \begin{matrix} q \\ \dot{q} \end{matrix} \right\|_2$$

    - But It is not good for obstacle avoidance:



Matlab Program: Obstacle_3R.m

Obstacle Avoidance

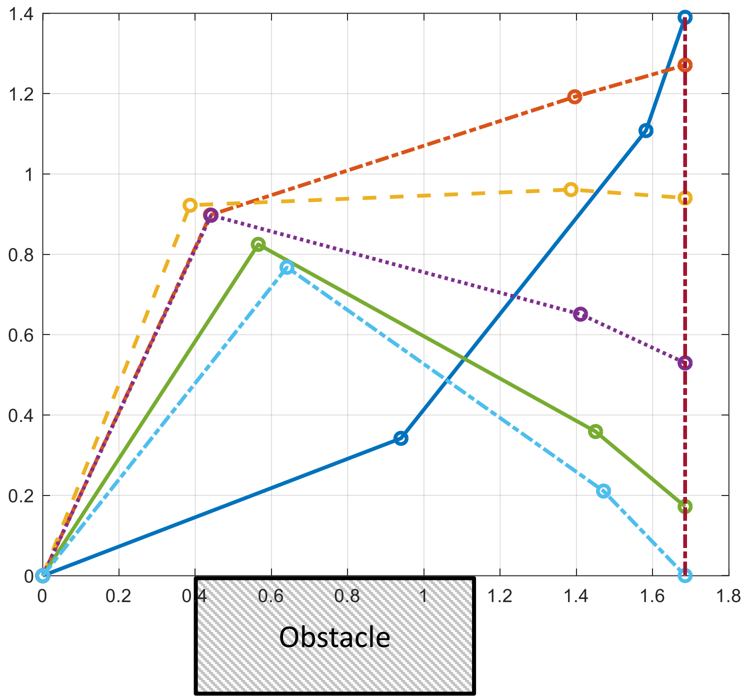
- Example 2: (Cont.)
  - ✓ Redundancy Resolution for 3R Robot:
    - Solve optimization problem

$$\min_{\boldsymbol{q},\dot{\boldsymbol{q}}} \|\boldsymbol{q}-\boldsymbol{q}_d\|_2$$

$$\text{Subject to} \begin{cases} \dot{\boldsymbol{\chi}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \\ \boldsymbol{\chi} = \boldsymbol{f}_{FK}(\boldsymbol{q}) \end{cases}$$

For Robot desired posture (elbow up)
$\boldsymbol{q}_d = [45^o, -70^o, -10^o]^T$

The trajectory is traversed without interfering with the obstacle.



Obstacle Avoidance

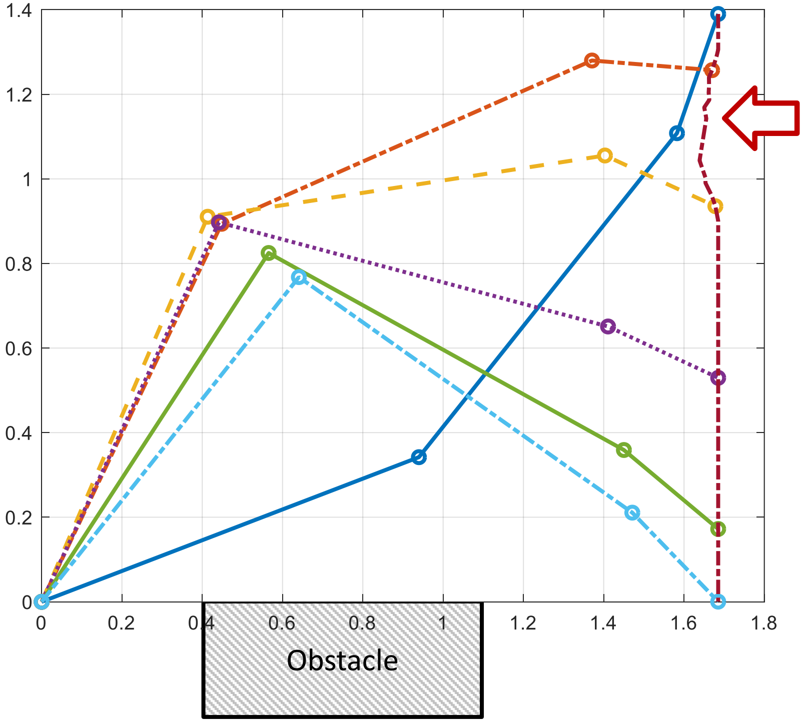
- Example 2: (Cont.)
  - ✓ Redundancy Resolution for 3R Robot:
    - Solve optimization problem

$$\min_{\boldsymbol{q},\dot{\boldsymbol{q}}} \|\boldsymbol{q} - \boldsymbol{q}_d\|_2$$

$$\text{Subject to } \begin{cases} \dot{\boldsymbol{\chi}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \\ \boldsymbol{\chi} = \boldsymbol{f}_{\boldsymbol{FK}}(\boldsymbol{q}) \\ \dot{\boldsymbol{q}}_{\boldsymbol{min}} \le \dot{\boldsymbol{q}} \le \dot{\boldsymbol{q}}_{\boldsymbol{max}} \end{cases}$$

With $-1.2 \le \dot{\boldsymbol{q}}_{\boldsymbol{i}} \le 1.2$

At some instances, the robot needs to exceed the velocity bound, and therefore, the trajectory is not traversed perfectly.



Singularity Circumvention

- Example 3:
  - ✓ Redundancy Resolution for 3R Robot:
    - Consider a desired vertical motion (2D) near singular configuration

      From $\boldsymbol{q}_0 = [-180^o, -179^o, 10^o]^T \rightarrow \ Q_0 = [x_0, y_0]^T$

      To $\ Q_f = [x_0, -0.1]^T$
    - In task space:

      Move $\boldsymbol{\chi}_0 = [x_0, y_0]^T \cong [1.7, 0.07]^T$ To $\boldsymbol{\chi}_f = [x_0, 0]^T$ in one seconds.

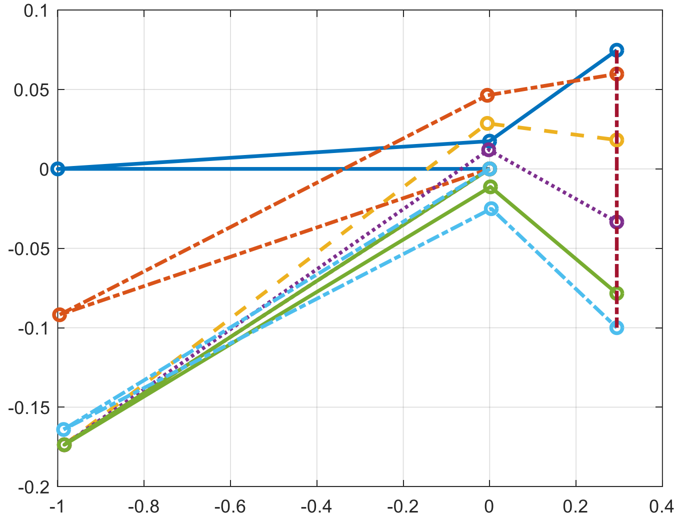
      Use cubic trajectory planning:

      $x_d(t) = x_0, \qquad y_d(t) = y_0 - (3 - 2t)t^2\,(y_0 + 0.1)$
    - In velocity space given:

      $\dot{x}_d(t) = 0, \qquad \dot{y}_d(t) = (y_0 + 0.1)(18t^2 - 6t)$

      Find $\boldsymbol{q}(t) = [q_1, q_2, q_3]^T$ to traverse this path while circumventing singularities.
    - Consider the base Solution:

      $\dot{\boldsymbol{q}}_{LS} = \boldsymbol{J}^{\dagger}(\boldsymbol{q})\dot{\boldsymbol{\chi}}$ where $\boldsymbol{J}^{\dagger} = \boldsymbol{J}^{\boldsymbol{T}}\left(\boldsymbol{J}\boldsymbol{J}^{\boldsymbol{T}}\right)^{-1}$
    - But this solution is close to singular configurations.



Matlab Program: Singularity_3R.m

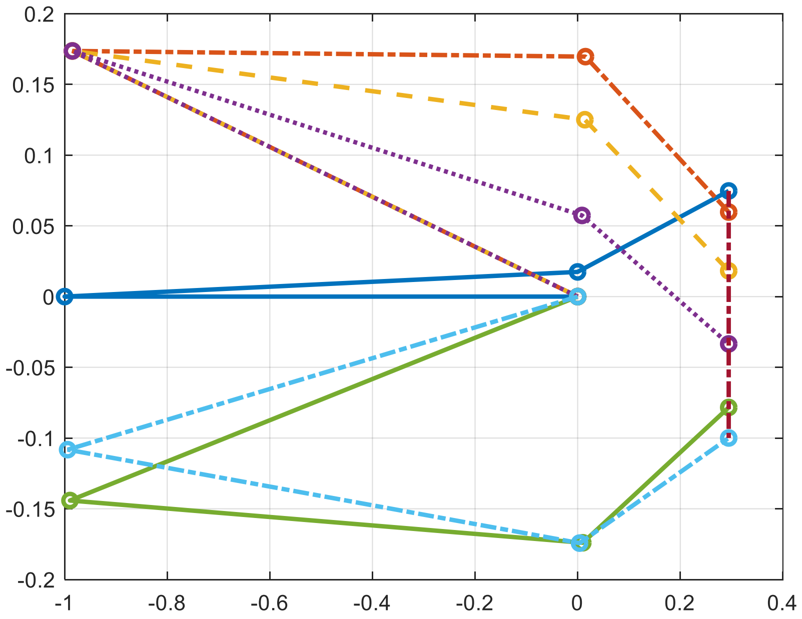
Singularity Circumvention

- Example 2: (Cont.)
  - ✓ Redundancy Resolution for 3R Robot:
    - Solve optimization problem

$$\min_{\boldsymbol{q},\dot{q}} -\mu(q) = -\sqrt{\det(JJ^T)}$$

Subject to $\begin{cases} \dot{\chi} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \\ \chi = \boldsymbol{f}_{\boldsymbol{FK}}(\boldsymbol{q}) \end{cases}$

In which, J in μ is considered to be the Jacobian of the first two links, in order to traverse he trajectory while getting away from fully retracted arms.



Matlab Program: Singularity_3R.m

Contents

In this chapter we review the Jacobian analysis for serial robots. First the definition to angular and linear velocities are given, then the Jacobian matrix is defined in conventional and screw-based representation, while their general and iterative derivation methods are given. Next the static wrench and its relation to Jacobian transpose is introduced, and Jacobian characteristics such as singularity, isotropy, dexterity and manipulability are elaborated. Inverse Jacobian solution for fully-, under- and redundantly-actuator robots are formulated, and redundancy resolution schemes are detailed. Finally, Stiffness analysis of robotic manipulators is reviewed in detail.

Stiffness Analysis

Consider a robot in contact to the environment

Applying wrench to the environment

Focus on the deflections resulting by the applied wrench

Stiffness

01 Actuators and Transmissions
The main source of compliance in the robots, the flexible elements in the transmission and force transducers

02 Links
The size and material of the links contribute greatly in the stiffness

03 Joints
Flexible versus rigid joints contribute in the compliance of the robot

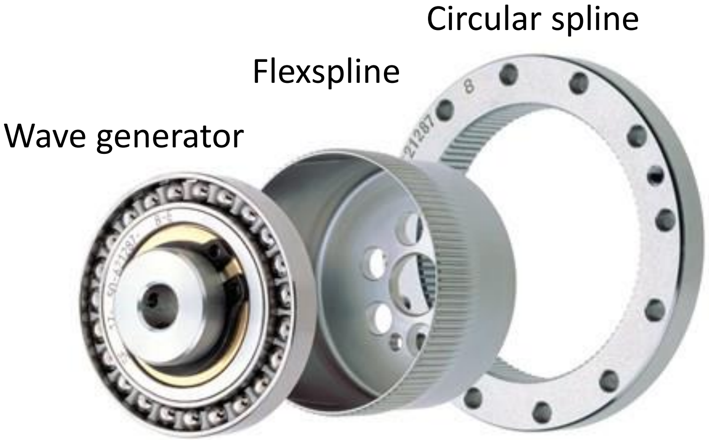
04 Robot Structure
Closed Kinematic chains cause higher stiffness to the robot

05 Control Structure
Through suitable control schemes, the stiffness of the robot might be shaped to the desired value

Stiffness Analysis

- Sources of Stiffness
  - ✓ Consider just the actuators and transmissions
  - ✓ As the main source of compliance
    - ▪ Harmonic Drive Systems

      Transmission is performed through a flexible element (Flexspline)



Stiffness Analysis

- Compliance and Stiffness Matrix
  - ✓ Note the stiffness relation in the i^{th} robot joint

$$\tau_i = k_i \Delta q_i$$

  - ▪ Joint stiffness constant: k_i

    τ_i denotes the transmitted torque through the actuator

    Δq_i denotes the resulting deflection at the joint

  - ▪ Use vector notation

$$\boldsymbol{\tau} = \mathcal{K}\, \Delta \boldsymbol{q}$$

$\boldsymbol{\tau} = [\tau_1, \tau_2, \dots, \tau_m]^T$ denotes the vector of transmitted torques

$\boldsymbol{\Delta q} = [q_1, q_2, \dots, q_m]^T$ denotes the vector of resulting deflections at the joints and

$$\mathcal{K} = diag[k_1, k_2, \dots, k_m]$$

  - ▪ Use Jacobian relation

$$\boldsymbol{\Delta\chi} = \boldsymbol{J}\, \boldsymbol{\Delta q} \quad \text{and} \quad \boldsymbol{\tau} = \boldsymbol{J}^T \boldsymbol{\mathcal{F}}$$

This results in $\boldsymbol{\Delta\chi} = \boldsymbol{J}\, \boldsymbol{\Delta q} = \boldsymbol{J}\, \mathcal{K}^{-1} \boldsymbol{\tau} = \boldsymbol{J}\, \mathcal{K}^{-1} \boldsymbol{J}^T \boldsymbol{\mathcal{F}} = \boldsymbol{C}\boldsymbol{\mathcal{F}}$ for a squared Jacobian.

Stiffness Analysis

- Compliance and Stiffness Matrix
  - ✓ Manipulator Compliance Matrix $\boldsymbol{C}$:

$$\Delta\boldsymbol{\chi} = \boldsymbol{C}\,\boldsymbol{\mathcal{F}} \quad \text{where} \quad \boldsymbol{C}_{m\times m} = \boldsymbol{J}\,\mathcal{K}^{-1}\boldsymbol{J}^{\boldsymbol{T}}$$

  - ✓ Manipulator Stiffness Matrix $\boldsymbol{K} = \boldsymbol{C}^{-1}$; the inverse map:

$$\boldsymbol{\mathcal{F}} = \boldsymbol{K}\,\Delta\boldsymbol{\chi} \quad \text{where} \quad \boldsymbol{K}_{m\times m} = \boldsymbol{J}^{-T}\mathcal{K}\,\boldsymbol{J}^{-1}$$

    - Both $\boldsymbol{C}$ and $\boldsymbol{K}$ are configuration dependent.
    - For uniform joint stiffness $k_1 = k_2 = \cdots = k_m = k$:

$$\boldsymbol{C} = k^{-1}\left(\boldsymbol{J}\boldsymbol{J}^{\boldsymbol{T}}\right) \quad \text{while} \quad \boldsymbol{K} = k\left(\boldsymbol{J}\boldsymbol{J}^{\boldsymbol{T}}\right)^{-\mathbf{1}}$$

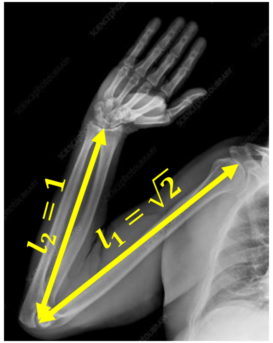
    - Interesting to reach to the same manipulability matrix $\boldsymbol{J}\boldsymbol{J}^{\boldsymbol{T}}$
    - Scaled Ellipsoid for Force – Deflection relation
    - For a uniform and unit end effector deflection $\|\Delta\boldsymbol{\chi}\|_2 = 1$.

$$\Delta\boldsymbol{\chi}^{\boldsymbol{T}}\Delta\boldsymbol{\chi} = 1 \;\rightarrow\; \boldsymbol{\mathcal{F}}^{\boldsymbol{T}}\boldsymbol{C}^{\boldsymbol{T}}\,\boldsymbol{C}\,\boldsymbol{\mathcal{F}} = 1.$$

Stiffness Analysis

✓ Example: Stiffness Analysis of 2R Manipulator

- Inspired by human arm, consider $l_1 = \sqrt{2}$, and $l_2 = 1$.
- Consider $\theta_2 = \frac{\pi}{2}$, and $k_1 = k_2 = 1N \cdot m$, then for this configuration:



$$\boldsymbol{C} = \boldsymbol{J}\boldsymbol{J}^T\left(\theta_2 = \frac{\pi}{2}\right) = \begin{bmatrix} 2 & \sqrt{2} \\ \sqrt{2} & 2 \end{bmatrix}; \quad \boldsymbol{C}^T\boldsymbol{C} = \begin{bmatrix} 6 & 4\sqrt{2} \\ 4\sqrt{2} & 6 \end{bmatrix}$$

Calculate the compliance map gains

For $\boldsymbol{C}^T\boldsymbol{C} \rightarrow \begin{matrix} \lambda_1 = 11.657 \\ \lambda_2 = 0.3431 \end{matrix}$; while $v_1 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix}, v_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} -1 \\ 1 \end{bmatrix},$

For $\boldsymbol{C}$; $\begin{matrix} G_{max} = \sigma_{max} = 3.414 \\ G_{min} = \sigma_{min} = 0.586 \end{matrix}$; while $\vec{v}_{max} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \vec{v}_{min} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$

Stiffness Analysis

✓ Example 1:

- Proof of the obtained gains for $\theta_2 = \frac{\pi}{2}$ configuration:

From $\|\boldsymbol{\Delta\chi}\|_2^2 = \boldsymbol{\mathcal{F}}^T\boldsymbol{C}^T\boldsymbol{C}\boldsymbol{\mathcal{F}} \rightarrow \Delta x^2 + \Delta y^2 = 6f_x^2 + 8\sqrt{2}f_x f_y + f_y^2$

$$\Delta x^2 + \Delta y^2 = 11.657\left(\frac{f_x}{\sqrt{2}} + \frac{f_y}{\sqrt{2}}\right)^2 + 0.343\left(\frac{f_x}{\sqrt{2}} - \frac{f_y}{\sqrt{2}}\right)^2$$

For $v_1 = \begin{bmatrix}1\\1\end{bmatrix}, f_x = f_y \rightarrow \Delta x^2 + \Delta y^2 = 11.657\left(\sqrt{2}f_x\right)^2 \rightarrow \left(\sqrt{2}f_x\right)^2 = 0.085\,(\Delta x^2 + \Delta y^2)$

$$\Rightarrow f_{min} = 0.292\|\boldsymbol{\Delta\chi}\| = \frac{1}{\sigma_{max}}\|\boldsymbol{\Delta\chi}\| \text{ in } \vec{v}_{min} = \begin{bmatrix}1\\1\end{bmatrix} \text{ direction}$$

For $v_1 = \begin{bmatrix}-1\\1\end{bmatrix}, f_x = -f_y \rightarrow \Delta x^2 + \Delta y^2 = 0.343\left(\sqrt{2}f_x\right)^2 \rightarrow \left(\sqrt{2}f_x\right)^2 = 2.914\,(\Delta x^2 + \Delta y^2)$

$$\Rightarrow f_{max} = 1.707\|\boldsymbol{\Delta\chi}\| = \frac{1}{\sigma_{min}}\|\boldsymbol{\Delta\chi}\| \text{ in } \vec{v}_{max} = \begin{bmatrix}-1\\1\end{bmatrix} \text{ direction}$$

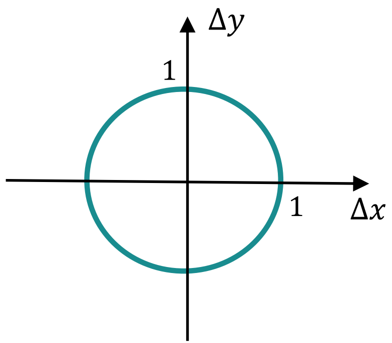
Stiffness Analysis

✓ Example: (Cont.)

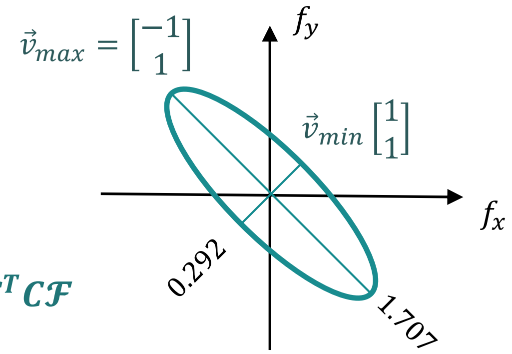
- For $\theta_2 = \frac{\pi}{2}$ configuration:

$$\Rightarrow \boldsymbol{\mathcal{F}}_{max} = 1.707\|\boldsymbol{\Delta\chi}\| = \frac{1}{\sigma_{min}}\|\boldsymbol{\Delta\chi}\| \text{ in } \vec{v}_{max} \text{ direction}$$

$$\Rightarrow \boldsymbol{\mathcal{F}}_{min} = 0.292\|\boldsymbol{\Delta\chi}\| = \frac{1}{\sigma_{max}}\|\boldsymbol{\Delta\chi}\| \text{ in } \vec{v}_{min} \text{ direction}$$



$$\|\boldsymbol{\Delta\chi}\|_2^2 = \boldsymbol{\mathcal{F}}^T\boldsymbol{C}^T\boldsymbol{C}\boldsymbol{\mathcal{F}}$$



Hamid D. Taghirad
Professor

About Hamid D. Taghirad

Hamid D. Taghirad has received his B.Sc. degree in mechanical engineering from Sharif University of Technology, Tehran, Iran, in 1989, his M.Sc. in mechanical engineering in 1993, and his Ph.D. in electrical engineering in 1997, both from McGill University, Montreal, Canada. He is currently the University Vice-Chancellor for Global strategies and International Affairs, Professor and the Director of the Advanced Robotics and Automated System (ARAS), Department of Systems and Control, Faculty of Electrical Engineering, K. N. Toosi University of Technology, Tehran, Iran. He is a senior member of IEEE, and Editorial board of International Journal of Robotics: Theory and Application, and International Journal of Advanced Robotic Systems. His research interest is *robust* and *nonlinear control* applied to *robotic systems*. His publications include five books, and more than 250 papers in international Journals and conference proceedings.

Robotics: Mechanics & Control

Chapter 4: Differential Kinematics

To read more and see the course videos
visit our course website:
http://aras.kntu.ac.ir/arascourses/robotics/

Thank You