ME/AER 676 Robot Modeling & Control
Spring 2023

Forward Kinematics & Jacobians

Hasan A. Poonawala

Department of Mechanical Engineering
University of Kentucky

Email: hasan.poonawala@uky.edu
Web: https://www.engr.uky.edu/~hap

Introduction

- We consider robots modeled as links joined in series.
- The degrees of freedom at the joints form the joint variables q.
- Task variables X capture quantities describing what the robot must do.
- Traditional robot control focuses on the conversion of joint variables to task variables (forward kinematics) and back (inverse kinematics)

$$X = f(q); \quad q = f^{-1}(X)$$

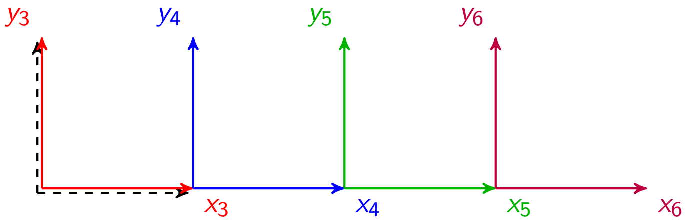
Forward Kinematics as Homogenous Transformations

- This problem involves composing a number of relative link (homogenous) transformations
- It may be solved numerically, with the specific details depending on how these link transformations are parametrized
- The transformation (d, R) may be represented by
  - origin and Euler angles (URDF)
  - D-H Parameters
  - Twist (Screw Theory)
  - etc. ...

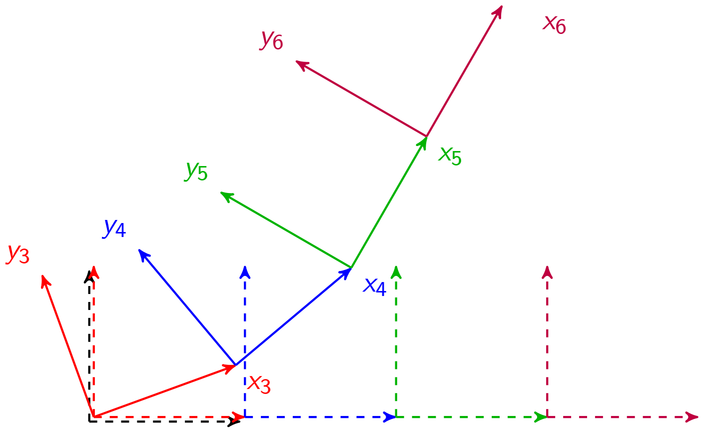
Serial Kinematic Chains

- We look at serial kinematic chains where all joints are simple.
- We number links as 0 for base to n in sequence.
- The assumption of single-parameter joints means we can use basic transformations to handle coordinate transformations.
- These basic transformation are denoted $A_i(q_i)$, where $q_i \in \mathbb{R}$ is the joint variable.
- q_i is either an angle θ_i (revolute joints) or a distance d_i (prismatic joints).

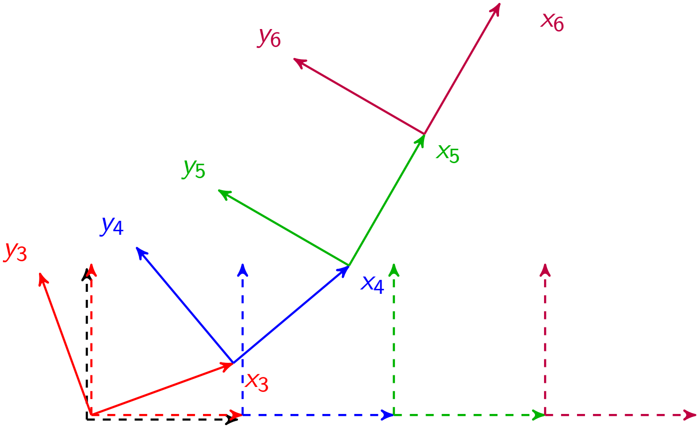
Example: Planar3R



Example: Planar3R



Example: Planar3R



$$o_6^2 = \begin{bmatrix} \mathrm{Rot}_{z,q_1} & \begin{bmatrix}0\\0\\0\end{bmatrix} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{Rot}_{z,q_2} & \begin{bmatrix}1\\0\\0\end{bmatrix} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{Rot}_{z,q_3} & \begin{bmatrix}1\\0\\0\end{bmatrix} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I_3 & \begin{bmatrix}1\\0\\0\end{bmatrix} \\ 0 & 1 \end{bmatrix} \begin{bmatrix}0\\0\\0\\1\end{bmatrix}$$

Forward Kinematics of Serial Chains

Given link i and $i-1$,

$$A_i = \begin{bmatrix} R_i^{i-1} & o_i^{i-1} \\ 0 & 1 \end{bmatrix} \tag{1}$$

Transformations between links i and j is T_j^i, where we are expressing frame j in frame i.

$$T_j^i = \begin{cases} A_{i+1}A_{i+2}\cdots A_{j-1}A_j & i < j \\ I & i = j \\ \left(T_j^i\right)^{-1} & i > j \end{cases} \tag{2}$$

Forward Kinematics of Serial Chains

- For an n-link serial chain manipulator, the task variables are a combination of
  - Origin of frame n (*end-effector* or *tool* frame)
  - Orientation of frame n
- $$T_n^0(q) = \begin{bmatrix} R_n^0(q) & d_n^0(q) \\ 0 & 1 \end{bmatrix}$$
- X is derived from $R_n^0(q)$ and/or $d_n^0(q)$
  i.e. $X = f(q)$

Modern Robotics

- The book "Modern Robotics" uses exponential coordinates (twists) to represent homogenous transformations.
- It does not follow the D-H convention (next slide).
- The main difference to D-H is that in MR frame i fixed to link i is at joint i, not joint $i+1$.
- Videos on FK in this course follow MR's convention of locating frame i at joint i.
- Universal Robot Description Formats (URDFs) also follow this approach

Denavit-Hartenberg Convention

In this convention

- All motion happens along the z axis
- Four numbers are enough to define relative link transformations (instead of 6 or 12).

The D-H convention is based on two restrictions:

(DH1) The x_1 axis intersects the z_0 axis.
(DH2) The x_1 axis is orthogonal to the z_0 axis.

This restriction makes the transformation matrix between link i and $i-1$ given in (1) reduce to

$$A_i = \mathrm{Rot}_{z,\theta_i}\mathrm{Trans}_{z,d_i}\mathrm{Trans}_{x,a_i}\mathrm{Rot}_{x,\alpha_i} \tag{3}$$

This convention is more common in earlier robotics texts, and is used in many systems.

Positions $\rightarrow$ Velocities

- We assign coordinates – aka rigid body pose – (d, R) to frame, relative to reference.
  $d \in \mathbb{R}^3$, $R \in \mathrm{SO}(3)$

Positions $\to$ Velocities

- We assign coordinates – aka rigid body pose – (d, R) to frame, relative to reference.
  $d \in \mathbb{R}^3$, $R \in \mathrm{SO}(3)$

- If the rigid body pose tells us where a frame is located, its position, what is the **rate-of-change of the position**?

Positions $\rightarrow$ Velocities

- We assign coordinates – aka rigid body pose – (d, R) to frame, relative to reference.
  $d \in \mathbb{R}^3$, $R \in \mathrm{SO}(3)$

- If the rigid body pose tells us where a frame is located, its position, what is the **rate-of-change of the position**?

- For a position vector in $\mathbb{R}^n$, we know that the rate of change of position is another vector in $\mathbb{R}^n$, called the **velocity**

Positions $\rightarrow$ Velocities

- We assign coordinates – aka rigid body pose – (d, R) to frame, relative to reference.
  $d \in \mathbb{R}^3$, $R \in \mathrm{SO}(3)$
- If the rigid body pose tells us where a frame is located, its position, what is the **rate-of-change of the position**?
- For a position vector in $\mathbb{R}^n$, we know that the rate of change of position is another vector in $\mathbb{R}^n$, called the **velocity**
- However, the orientation coordinate (d, R) is not a vector!
  What is $\frac{\mathrm{d}}{\mathrm{dt}}R(t)$?

Velocities in $\mathrm{SO}(3)$

- The angular velocity $\omega \in \mathbb{R}^3$ can be represented using two different sets of 3 numbers:
  - Analytic: As the three derivatives of the three numbers used to parametrize $\mathrm{SO}(3)$ (not a physical vector). Example parametrization: roll-pitch-yaw

Velocities in $\mathrm{SO}(3)$

- The angular velocity $\omega \in \mathbb{R}^3$ can be represented using two different sets of 3 numbers:
  - Analytic: As the three derivatives of the three numbers used to parametrize $\mathrm{SO}(3)$ (not a physical vector). Example parametrization: roll-pitch-yaw
  - Geometric: As a vector in 3D describing the instantaneous axis of rotation in a frame and speed of rotation.

Velocities in $\mathrm{SO}(3)$: Analytic

- Example parametrization: roll-pitch-yaw $\phi - \theta - \psi$ (RPY).

Velocities in $\mathrm{SO}(3)$: Analytic

- Example parametrization: roll-pitch-yaw $\phi - \theta - \psi$ (RPY).
- Consider an end-effector whose orientation R_n^0 in the base frame is parametrized by RPY Euler angles $\alpha = (\phi, \theta, \psi)$.

Velocities in $\mathrm{SO}(3)$: Analytic

- Example parametrization: roll-pitch-yaw $\phi - \theta - \psi$ (RPY).
- Consider an end-effector whose orientation R_n^0 in the base frame is parametrized by RPY Euler angles $\alpha = (\phi, \theta, \psi)$.
- There are two ways to derive R_n^0;

$$\text{by definition:} \quad R_n^0(\phi, \theta, \psi) = \mathrm{Rot}_{z,\psi}\ \mathrm{Rot}_{y,\theta}\mathrm{Rot}_{x,\phi} \tag{4}$$

$$\text{FK :} \quad R_n^0(q) = A_1(q_1)A_1(q_2)\cdots A_n(q_n) \tag{5}$$

Velocities in $\mathrm{SO}(3)$: Analytic

- Example parametrization: roll-pitch-yaw $\phi - \theta - \psi$ (RPY).
- Consider an end-effector whose orientation R_n^0 in the base frame is parametrized by RPY Euler angles $\alpha = (\phi, \theta, \psi)$.
- There are two ways to derive R_n^0;

$$\text{by definition:} \quad R_n^0(\phi, \theta, \psi) = \mathrm{Rot}_{z,\psi}\ \mathrm{Rot}_{y,\theta}\mathrm{Rot}_{x,\phi} \tag{4}$$

$$\text{FK}: \quad R_n^0(q) = A_1(q_1)A_1(q_2)\cdots A_n(q_n) \tag{5}$$

- With a little work, we can derive $X(t) = \alpha(t) = f(q(t))$

Velocities in $\mathrm{SO}(3)$: Analytic

- Example parametrization: roll-pitch-yaw $\phi - \theta - \psi$ (RPY).
- Consider an end-effector whose orientation R_n^0 in the base frame is parametrized by RPY Euler angles $\alpha = (\phi, \theta, \psi)$.
- There are two ways to derive R_n^0;

$$\text{by definition:} \quad R_n^0(\phi, \theta, \psi) = \mathrm{Rot}_{z,\psi}\ \mathrm{Rot}_{y,\theta}\mathrm{Rot}_{x,\phi} \quad (4)$$

$$\text{FK}: \quad R_n^0(q) = A_1(q_1)A_1(q_2)\cdots A_n(q_n) \quad (5)$$

- With a little work, we can derive $X(t) = \alpha(t) = f(q(t))$
- Therefore

$$\dot{X}(t) = \dot{\alpha}(t) \in \mathbb{R}^3$$

Velocities in $\mathrm{SO}(3)$: Analytic

- Example parametrization: roll-pitch-yaw $\phi - \theta - \psi$ (RPY).
- Consider an end-effector whose orientation R_n^0 in the base frame is parametrized by RPY Euler angles $\alpha = (\phi, \theta, \psi)$.
- There are two ways to derive R_n^0;

$$\text{by definition:} \quad R_n^0(\phi, \theta, \psi) = \mathrm{Rot}_{z,\psi}\ \mathrm{Rot}_{y,\theta}\mathrm{Rot}_{x,\phi} \tag{4}$$

$$\text{FK}: \quad R_n^0(q) = A_1(q_1)A_1(q_2)\cdots A_n(q_n) \tag{5}$$

- With a little work, we can derive $X(t) = \alpha(t) = f(q(t))$
- Therefore

$$\dot{X}(t) = \dot{\alpha}(t) \in \mathbb{R}^3$$

- Note that we can also derive $\frac{\mathrm{d}}{\mathrm{dt}} R_n^0(\phi, \theta, \psi)$ as a matrix function of α and $\dot{\alpha}$.

Velocities in $\mathrm{SO}(3)$: Geometric

- It turns out that
$$\dot{R}(t) = SR,$$
where S satisfies $S + S^T = 0$

Velocities in $\mathrm{SO}(3)$: Geometric

- It turns out that
$$\dot{R}(t) = SR,$$
where S satisfies $S + S^T = 0$
- S is a skew-symmetric matrix, and has the form
$$S = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix},$$
for any three numbers ω_1, ω_2, ω_3

Velocities in $\mathrm{SO}(3)$: Geometric

- It turns out that
$$\dot{R}(t) = SR,$$
where S satisfies $S + S^T = 0$
- S is a skew-symmetric matrix, and has the form
$$S = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix},$$
for any three numbers ω_1, ω_2, ω_3
- Physically, the vector $\omega = \begin{bmatrix} \omega_1 & \omega_2 & \omega_3 \end{bmatrix}^T$ defines the instantaneous angular velocity in base/space frame $\{0\}$

Velocities in $\mathrm{SO}(3)$: Geometric

- It turns out that
$$\dot{R}(t) = SR,$$
where S satisfies $S + S^T = 0$

- S is a skew-symmetric matrix, and has the form
$$S = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix},$$
for any three numbers ω_1, ω_2, ω_3

- Physically, the vector $\omega = \begin{bmatrix} \omega_1 & \omega_2 & \omega_3 \end{bmatrix}^T$ defines the instantaneous angular velocity in base/space frame $\{0\}$

- So, when a task is $x(t) = (d(t), R(t)) \in \mathbb{R}^3 \times \mathrm{SO}(3)$, its velocity is
$$\xi \in \mathbb{R}^6 = \underbrace{\mathbb{R}^3}_{\text{linear velocity}} \times \underbrace{\mathbb{R}^3}_{\text{angular velocity}}$$

Jacobians and Forward Velocity Kinematics

X is derived from $R_n^0(q)$ and/or $d_n^0(q)$, where
$$T_n^0(q) = \begin{bmatrix} R_n^0(q) & d_n^0(q) \\ 0 & 1 \end{bmatrix}$$

Jacobians and Forward Velocity Kinematics

X is derived from $R_n^0(q)$ and/or $d_n^0(q)$, where
$T_n^0(q) = \begin{bmatrix} R_n^0(q) & d_n^0(q) \\ 0 & 1 \end{bmatrix}$

$$\text{Forward Kinematics: } X = f(q) \tag{6}$$

$$\text{Forward Velocity Kinematics: } \dot{X} = ? \tag{7}$$

Jacobians and Forward Velocity Kinematics

X is derived from $R_n^0(q)$ and/or $d_n^0(q)$, where
$T_n^0(q) = \begin{bmatrix} R_n^0(q) & d_n^0(q) \\ 0 & 1 \end{bmatrix}$

$$\text{Forward Kinematics: } X = f(q) \tag{6}$$

$$\text{Forward Velocity Kinematics: } \dot{X} = J(q)\dot{q} \tag{7}$$

- $J(q)$: Jacobian matrix
- Size of $J(q)$ depends on joint and task space dimensions
- Derivation of $J(q)$ depends on type of coordinates for whether we use analytic or geometric representation of angular velocity
  - Analytic Jacobians
  - Geometric Jacobians

Jacobians

- Forward Kinematics provides X, whose change over time is the vector ξ

Jacobians

- Forward Kinematics provides X, whose change over time is the vector ξ
- The relationship between ξ and $\dot{q}$ is linear:

$$\xi = J(q)\dot{q}.$$

Jacobians

- Forward Kinematics provides X, whose change over time is the vector ξ
- The relationship between ξ and $\dot{q}$ is linear:

$$\xi = J(q)\dot{q}.$$

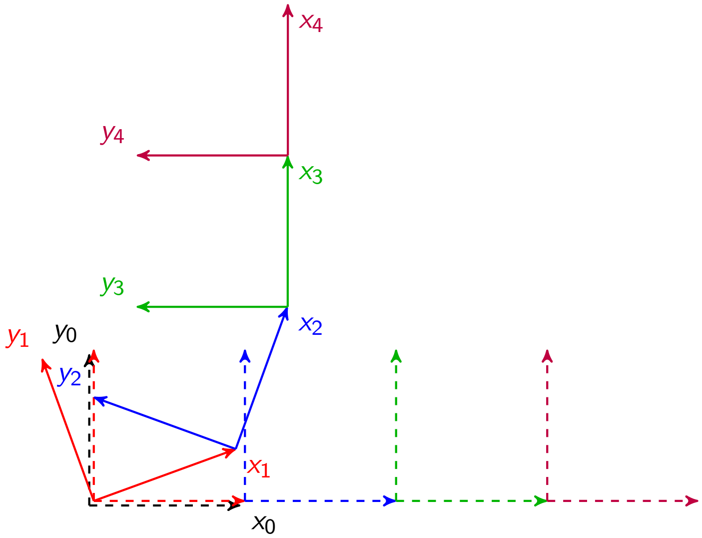
- When the orientation of X is given by a vector of three numbers $\alpha = f(q)$, then $\xi = \dot{X}(t)$, and the Jacobian is the **analytic Jacobian** given by

$$J_a(q) = \frac{\partial f}{\partial q}.$$

Jacobians

- Forward Kinematics provides X, whose change over time is the vector ξ
- The relationship between ξ and $\dot{q}$ is linear:
$$\xi = J(q)\dot{q}.$$
- When the orientation of X is given by a vector of three numbers $\alpha = f(q)$, then $\xi = \dot{X}(t)$, and the Jacobian is the **analytic Jacobian** given by
$$J_a(q) = \frac{\partial f}{\partial q}.$$
- When we represent change of orientation using angular velocity, $J(q)$ is the geometric Jacobian, derived using spatial geometry.

Example: Planar3R Geometric Jacobian



Example: Planar3R Geometric Jacobian

$$T_1^0 = \begin{bmatrix} \mathrm{Rot}_{z,q_1} & \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_1^0 & o_1^0 \\ 0 & 1 \end{bmatrix}$$

$$T_2^0 = \begin{bmatrix} \mathrm{Rot}_{z,q_1} & \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{Rot}_{z,q_2} & \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_2^0 & o_2^0 \\ 0 & 1 \end{bmatrix}$$

$$T_3^0 = \begin{bmatrix} \mathrm{Rot}_{z,q_1} & \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{Rot}_{z,q_2} & \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{Rot}_{z,q_3} & \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_3^0 & o_3^0 \\ 0 & 1 \end{bmatrix}$$

$$T_4^0 = \begin{bmatrix} \mathrm{Rot}_{z,q_1} & \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{Rot}_{z,q_2} & \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{Rot}_{z,q_3} & \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I_3 & \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\ 0 & 1 \end{bmatrix}$$

Building The Geometric Jacobian

If $\xi \in \mathbb{R}^6$ and $q \in \mathbb{R}^n$, the Jacobian $J(q)$ is of size $6 \times n$, where three rows form the velocity Jacobian J_v and three rows form the angular velocity Jacobian J_ω.

Assuming all joint axes are the z-direction of the link frame, the i^{th} column J_{v_i} of J_v is

$$J_{v_i} = \begin{cases} z_i^0 & , \quad \text{if joint } i \text{ is prismatic} \\ z_i^0 \times \left(o_n^0 - o_i^0\right) & , \quad \text{if joint } i \text{ is revolute} \end{cases} \tag{8}$$

We compute the i^{th} column J_{ω_i} of J_ω as

$$J_{\omega_i} = \begin{cases} 0_{3\times 1} & , \quad \text{if joint } i \text{ is prismatic} \\ z_i^0 & , \quad \text{if joint } i \text{ is revolute} \end{cases} \tag{9}$$

Uses of the Jacobian

- Forward Velocity Kinematics: Compute end-effector velocity ξ given joint angle derivatives $\dot{q}$
- Inverse Velocity Kinematics: Compute $\dot{q}$ given ξ
- Relates end-effector forces F to joint torques τ at equilibrium: $\tau = J(q)^T F$
- Defines the manipulability μ and the manipulability ellipsoid (next slide)

Manipulability

1. The manipulability μ is then given by

$$\mu = \Pi_{i=1}^{m} \sigma_i \tag{10}$$

where σ_i are the singular values of $J \in \mathbb{R}^{m \times n}$; $J = U \Sigma V$.

Manipulability

1. The manipulability μ is then given by

$$\mu = \Pi_{i=1}^{m} \sigma_i \tag{10}$$

   where σ_i are the singular values of $J \in \mathbb{R}^{m \times n}$; $J = U \Sigma V$.

2. Let $\mathrm{rank}(J) = m$, and $w = U^T \xi$. Then

$$\dot{q} = J^{+} \xi \implies \|\dot{q}\|^2 = \xi^T (J J^T)^{-1} \xi, \text{ where}$$

$$\xi^T (J J^T)^{-1} \xi = (U^T \xi)^T \Sigma_m^{-2} (U^T \xi) = w^T \Sigma_m^{-2} w = \sum_{i=1}^{m} \frac{w_i^2}{\sigma_{m_i}^2}$$

   and Σ_m is a square diagonal matrix formed from the m largest singular values of J

Manipulability

1. The manipulability μ is then given by

$$\mu = \Pi_{i=1}^{m} \sigma_i \tag{10}$$

where σ_i are the singular values of $J \in \mathbb{R}^{m \times n}$; $J = U \Sigma V$.

2. Let $\mathrm{rank}(J) = m$, and $w = U^T \xi$. Then

$$\dot{q} = J^{+} \xi \implies \|\dot{q}\|^2 = \xi^T (J J^T)^{-1} \xi, \text{ where}$$

$$\xi^T (J J^T)^{-1} \xi = (U^T \xi)^T \Sigma_m^{-2} (U^T \xi) = w^T \Sigma_m^{-2} w = \sum_{i=1}^{m} \frac{w_i^2}{\sigma_{m_i}^2}$$

and Σ_m is a square diagonal matrix formed from the m largest singular values of J

3. If $\|\dot{q}\|^2 = 1 = \xi^T (J J^T)^{-1} \xi$ then corresponding ξ form an ellipsoid in space of task velocities ξ.

Manipulability Ellipsoid

- The manipulability μ is related to the volume of the ellipsoid formed by unit norm q mapped under $J \in \mathbb{R}^{m \times n}$.

Manipulability Ellipsoid

- The manipulability μ is related to the volume of the ellipsoid formed by unit norm q mapped under $J \in \mathbb{R}^{m \times n}$.
- When J is close to losing full-rank, μ is close to zero, and *vice versa*.

Manipulability Ellipsoid

- The manipulability μ is related to the volume of the ellipsoid formed by unit norm q mapped under $J \in \mathbb{R}^{m \times n}$.
- When J is close to losing full-rank, μ is close to zero, and *vice versa*.
- When J is full-rank, the ellipsoid has non-zero volume

Manipulability Ellipsoid

- The manipulability μ is related to the volume of the ellipsoid formed by unit norm q mapped under $J \in \mathbb{R}^{m \times n}$.
- When J is close to losing full-rank, μ is close to zero, and *vice versa*.
- When J is full-rank, the ellipsoid has non-zero volume
- This ellipsoid has two physical interpretations:

Manipulability Ellipsoid

- The manipulability μ is related to the volume of the ellipsoid formed by unit norm q mapped under $J \in \mathbb{R}^{m \times n}$.
- When J is close to losing full-rank, μ is close to zero, and *vice versa*.
- When J is full-rank, the ellipsoid has non-zero volume
- This ellipsoid has two physical interpretations:
  - When there's no contact, this ellipsoid describes achievable task velocities given unit-size joint velocities.

Manipulability Ellipsoid

- The manipulability μ is related to the volume of the ellipsoid formed by unit norm q mapped under $J \in \mathbb{R}^{m \times n}$.
- When J is close to losing full-rank, μ is close to zero, and *vice versa*.
- When J is full-rank, the ellipsoid has non-zero volume
- This ellipsoid has two physical interpretations:
  - When there's no contact, this ellipsoid describes achievable task velocities given unit-size joint velocities.
  - During static contact, this ellipsoid describes achievable task forces given unit-size joint torques.