Introduction to Robotics

Lecture 4

Shuang Li, Jianwei Zhang
[sli, zhang]@informatik.uni-hamburg.de

University of Hamburg
Faculty of Mathematics, Informatics and Natural Sciences
Department of Informatics
Technical Aspects of Multimodal Systems

May 15, 2020

IK Review

- Workspace
  - reachable workspace
  - dexterous workspace
- closed solutions:
  - algebraic solution
  - geometrical solution

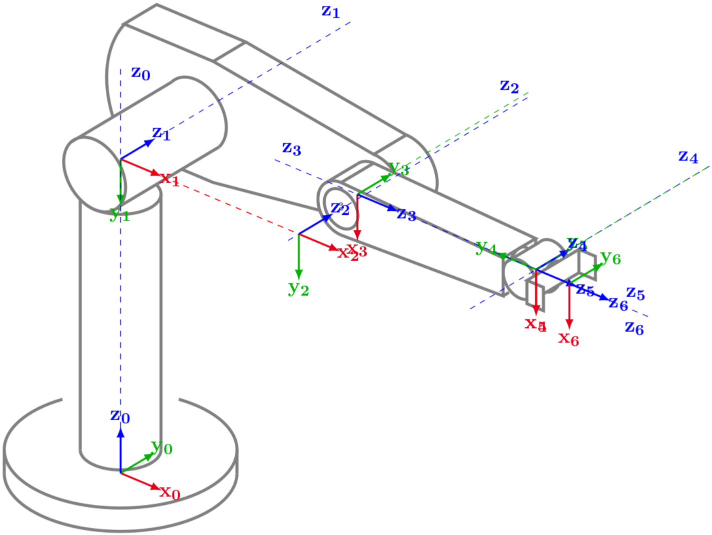
The closed solution exists if specific constraints (sufficient constraints) for the arm geometry are satisfied:

If 3 sequent axes intersect in a given point

or if 3 sequent axes are parallel to each other

- numerical solutions

Example featuring PUMA 560



Exercise

-

Assume we have derived the forward kinematics as:

$$^0T_3 = \begin{bmatrix} C_1 C_{23} & -C_1 S_{23} & S_1 & C_1(C_2 l_2 + l_1) \\ S_1 C_{23} & -S_1 S_{23} & -C_1 & S_1(C_2 l_2 + l_1) \\ S_{23} & C_{23} & 0 & S_2 l_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

And we know:

$$^0T_3 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Question: How to solve the inverse kinematics?

$$
{}^{0}T_3 = \begin{bmatrix} C_1 C_{23} & -C_1 S_{23} & S_1 & C_1(C_2 l_2 + l_1) \\ S_1 C_{23} & -S_1 S_{23} & -C_1 & S_1(C_2 l_2 + l_1) \\ S_{23} & C_{23} & 0 & S_2 l_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
S_1 = r_{13} \qquad (18)
$$

$$
C_1 = -r_{23} \qquad (19)
$$

Using the two-argument arctangent to solve for θ_1,

$$
\theta_1 =
$$

Exercise

$$C_1(C_2 l_2 + l_1) = p_x \tag{20}$$
$$S_1(C_2 l_2 + l_1) = p_y \tag{21}$$
$$S_2 l_2 = p_z \tag{22}$$

solve θ_2 from (20 - 22),

Exercise

$$S_{23} = r_{31} \quad (23)$$

$$C_{23} = r_{32} \quad (24)$$

solve θ_3 from (20 - 22),

Outline

Outline (cont.)

Differential motion

- Forward kinematics: $\theta \longrightarrow x$
- Inverse kinematics: $x \longrightarrow \theta$
- instantaneous kinematics: $\theta + \delta\theta \longrightarrow x + \delta x$
- Relationship $\delta\theta \leftrightarrow \delta x$

$$\dot{\theta} \leftrightarrow \dot{x}$$

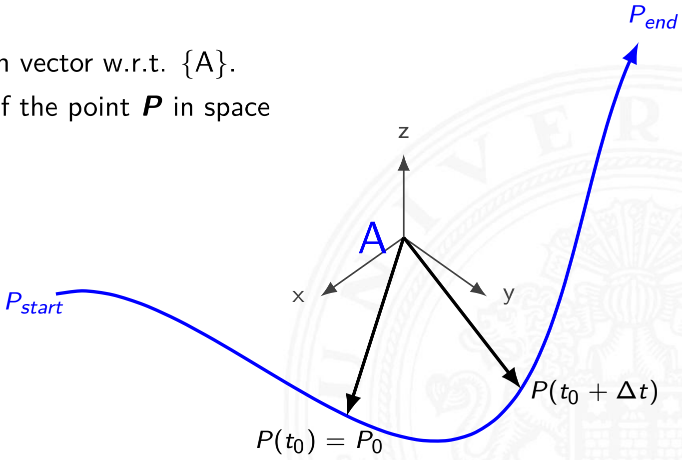
Joint velocities $\leftrightarrow$ end-effector velocities

- Linear velocity
- Angular velocity

Linear velocity

$$^{A}V_{P} = \frac{d}{dt}(^{A}P) = \lim_{\Delta t \to 0} \frac{\Delta \boldsymbol{P}(t)}{\Delta t} = \lim_{\Delta t \to 0} \frac{\boldsymbol{P}(t+\Delta t) - \boldsymbol{P}(t)}{\Delta t} \quad (25)$$

- $\boldsymbol{P}$ is a time-varying position vector w.r.t. $\{A\}$.
- $^{A}V_{P}$ is the linear velocity of the point $\boldsymbol{P}$ in space



Linear velocity (cont.)

Representing ${}^{A}V_P$ in another frame $\{B\}$, then we get

$$ {}^{B}({}^{A}V_P) = {}^{B}(\frac{d}{dt}({}^{A}P)) = \frac{d}{dt}({}^{B}R_A({}^{A}P)) = {}^{B}R_A \frac{d}{dt}({}^{A}P) = {}^{B}R_A \cdot {}^{A}V_P $$

Note, as ${}^{A}R_B$ remains invariant during the motion.

Notation

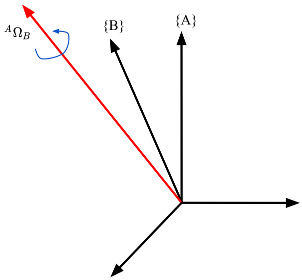
- if $\boldsymbol{P}$ is the origin of a frame $\{C\}$, which is moving, we typically use $v_c = {}^{U}V_C$ to denote the linear velocity of the origin of $\{c\}$ w.r.t. the reference frame $\{U\}$
- ${}^{A}v_c$ means the linear velocity of the origin of $\{C\}$ w.r.t. $\{U\}$ expressed in $\{A\}$

Angular velocity

Angular velocity describes rotational motion of a frame.

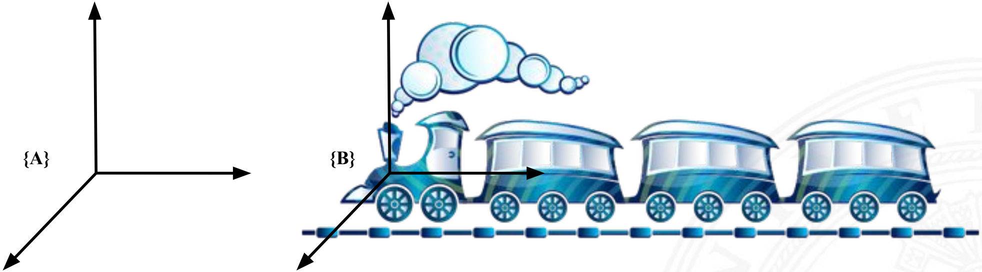
Notation

- ${}^{A}\Omega_{B}$ denotes the angular velocity of $\{B\}$ w.r.t. $\{A\}$
- $\omega_c = {}^{U}\Omega_C$ denotes the angular velocity of $\{c\}$ w.r.t. $\{U\}$



- the direction of ${}^{A}\Omega_{B}$ indicates the instantaneous axis of rotation
- the magnitude of ${}^{A}\Omega_{B}$ indicates the speed of rotation

Linear velocity of rigid body

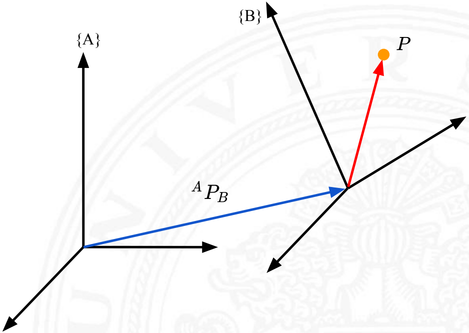


Assume that there is only a linear motion of {B} w.r.t. {A}

$$^{A}P = {}^{A}P_{B} + {}^{A}R_{B} \cdot {}^{B}P$$

Differentiating the above equation

$$\begin{aligned} ^{A}V_{P} &= {}^{A}V_{B} + \frac{d}{dt}({}^{A}R_{B} \cdot {}^{B}P) \\ &= {}^{A}V_{B} + {}^{A}R_{B}\frac{d}{dt}({}^{B}P) \\ &= {}^{A}V_{B} + {}^{A}R_{B} \cdot {}^{B}V_{P} \end{aligned}$$

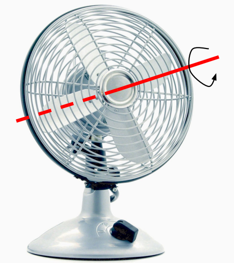


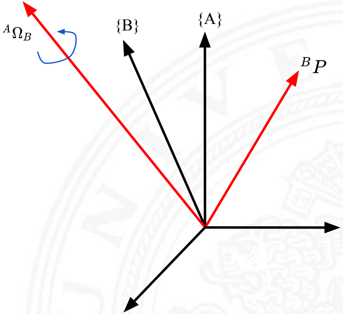
Note, as $^{A}R_{B}$ remains invariant during the motion.

Angular velocity of rigid body

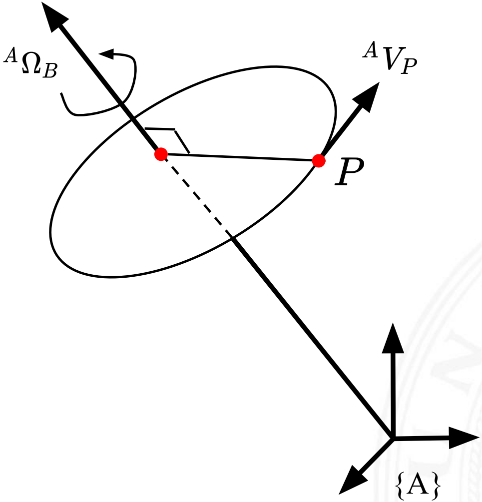
Assume that:

1. No linear velocity of {B} w.r.t. {A}
2. There is a rotational velocity of {B} w.r.t. {A}, ${}^{A}R_{B}$ is time-varying.
3. Point P is fixed in {B}

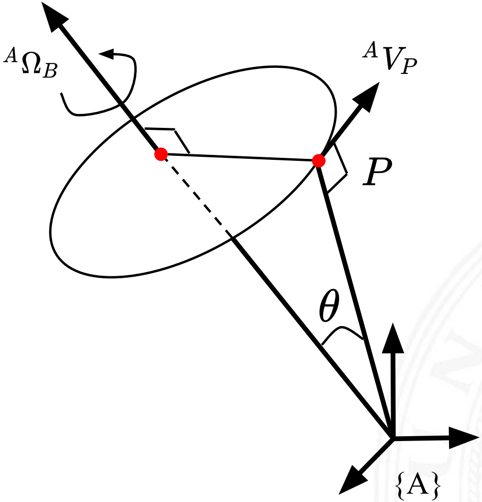




Angular velocity of rigid body (cont.)



Angular velocity of rigid body (cont.)



Angular velocity of rigid body

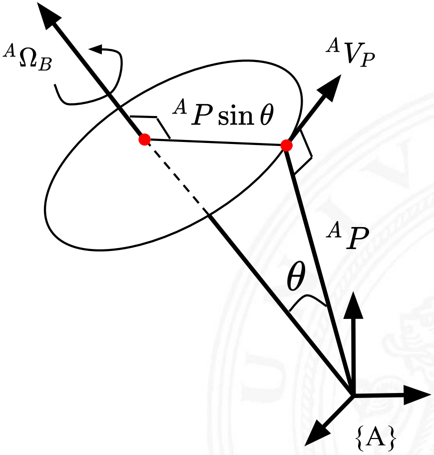
${}^A V_P$ is proportional to:

- $\|{}^A\Omega_B\|$
- $\|{}^A P \sin\theta\|$

and

- ${}^A V_P \perp {}^A\Omega_B$
- ${}^A V_P \perp {}^A P$

$$ {}^A V_P = {}^A\Omega_B \times {}^A P $$



Cross Product Operator

$$a = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}, b = \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} \longrightarrow c = a \times b \Longrightarrow c = \hat{a} b$$

$$a\times \Longrightarrow \hat{a} : \text{ a skew-symmetric matrix}$$

vectors $\Longrightarrow$ matrices

$$c = \hat{a} b = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix}$$

Cross Product Operator

$$ {}^{A}V_{P} = {}^{A}\Omega_{B} \times {}^{A}P = {}^{A}\hat{\Omega}_{B}{}^{A}P $$

$$ {}^{A}\Omega_{B} = \begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix}, {}^{A}P = \begin{bmatrix} {}^{A}P_x \\ {}^{A}P_y \\ {}^{A}P_z \end{bmatrix} $$

$$ {}^{A}V_{P} = {}^{A}\hat{\Omega}_{B}{}^{A}P = \begin{bmatrix} 0 & -\Omega_z & \Omega_y \\ \Omega_z & 0 & -\Omega_x \\ -\Omega_y & \Omega_x & 0 \end{bmatrix} \begin{bmatrix} {}^{A}P_x \\ {}^{A}P_y \\ {}^{A}P_z \end{bmatrix} $$

Simultaneous linear and angular motion

Assume that:

1. No linear velocity of {B} w.r.t. {A}
2. There is a rotational velocity of {B} w.r.t. {A}, ${}^{B}R_{A}$ is time-varying.
3. ~~Point P is fixed in {B}~~

$$ {}^{A}V_{P} = {}^{A}\Omega_{B} \times {}^{A}P $$
$$ \Downarrow {}^{B}V_{P} $$

$$ \begin{aligned} {}^{A}V_{P} &= {}^{A}R_{B}{}^{B}V_{P} + {}^{A}\Omega_{B} \times {}^{A}P \\ &= {}^{A}R_{B}{}^{B}V_{P} + {}^{A}\Omega_{B} \times {}^{A}R_{B}{}^{B}P \end{aligned} $$

Simultaneous linear and angular motion

Assume that:

1. ~~No linear velocity of {B} w.r.t. {A}~~
2. There is a rotational velocity of {B} w.r.t. {A}, ${}^{B}R_{A}$ is time-varying.
3. ~~Point Q is fixed in {B}~~

$$
{}^{A}V_{P} = {}^{A}R_{B}\,{}^{B}V_{P} + {}^{A}\Omega_{B} \times {}^{A}R_{B}\,{}^{B}P
$$

$$
\Downarrow \; {}^{A}V_{B}
$$

$$
{}^{A}V_{P} = {}^{A}V_{B} + {}^{A}R_{B}\,{}^{B}V_{P} + {}^{A}\Omega_{B} \times {}^{A}R_{B}\,{}^{B}P
$$

Velocity of rigid body

- Linear motion

$$ {}^{A}V_{P} = {}^{A}V_{B} + {}^{A}R_{B}\,{}^{B}V_{P} $$

- Rotational motion

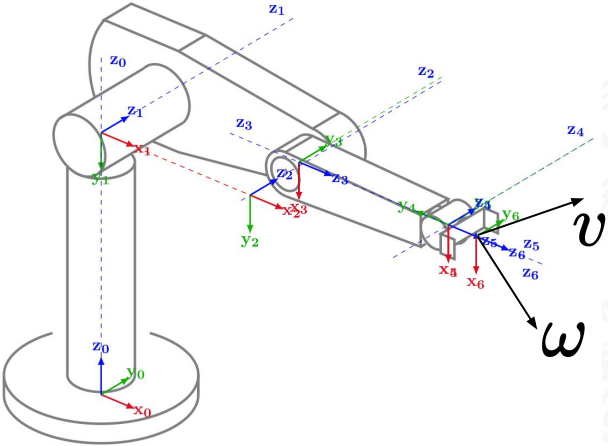
$$ {}^{A}V_{P} = {}^{A}R_{B}\,{}^{B}V_{P} + {}^{A}\Omega_{B} \times {}^{A}R_{B}\,{}^{B}P $$

- General

$$ {}^{A}V_{P} = {}^{A}V_{B} + {}^{A}R_{B}\,{}^{B}V_{P} + {}^{A}\Omega_{B} \times {}^{A}R_{B}\,{}^{B}P $$

Motion of the links of a manipulator.

- v: linear velocity
- ω : angular velocity

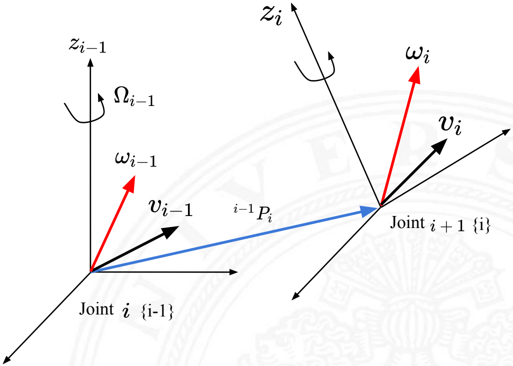


Angular velocity propagation

For a revolute joint i, the angular velocity ${}^{i-1}\omega_{i-1}$ of the link i is:

$\dot{\theta}_i {}^{i}Z_{i-1}$

- $\dot{\theta}_i$ is a scalar, the velocity of the joint i
- ${}^{i}Z_{i-1} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$
- scalar multiplication



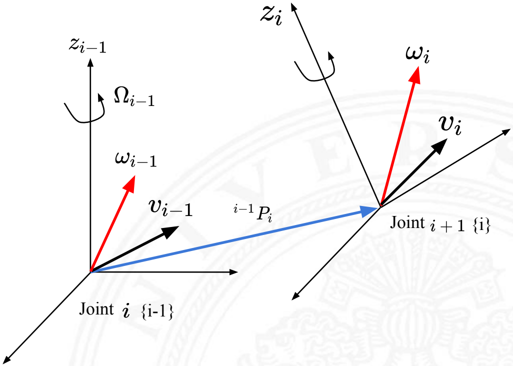
Angular velocity propagation

Angular velocity ${}^{i-1}\omega_i$ of the link $i+1$ is influenced by:

- the angular velocity ${}^{i-1}\omega_{i-1}$ of the link i
- if joint $i+1$ is a revolute joint, the joint velocity along the z-axis Z_i of the link

$${}^{i-1}\omega_i = {}^{i-1}\omega_{i-1} + {}^{i-1}R_i \dot{\theta}_{i+1} {}^{i}Z_i$$

$${}^{i}\omega_i = {}^{i}R_{i-1} {}^{i-1}\omega_{i-1} + \dot{\theta}_{i+1} {}^{i}Z_i$$

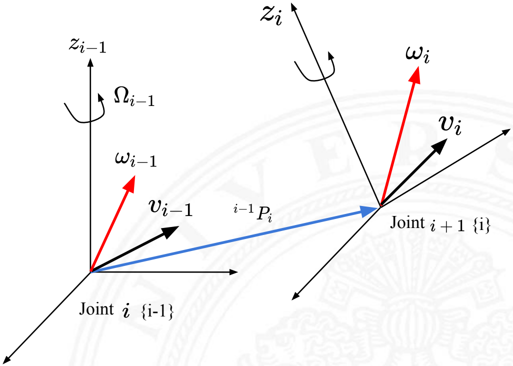


Linear velocity propagation

For a prismatic joint i, the linear velocity ${}^{i-1}v_{i-1}$ of the link i is:

$\dot{d}_i {}^{i}Z_{i-1}$

- $\dot{d}_i$ is a scalar, the velocity of the link i
- ${}^{i}Z_{i-1} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$
- scalar multiplication



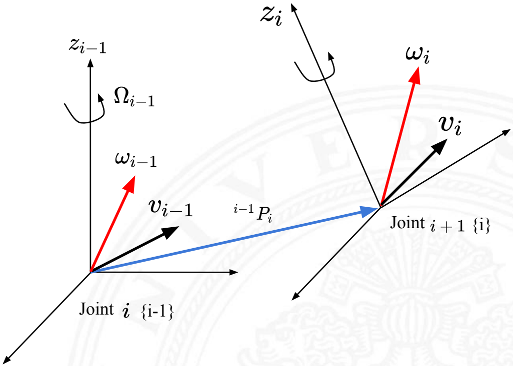
Linear velocity propagation

Linear velocity ${}^{i-1}v_i$ of the link $i+1$ is influenced by:

- the linear velocity ${}^{i-1}v_{i-1}$ of the joint i
- if joint i is a revolute joint, the linear velocity of the origin of frame $\{i+1\}$
- if joint $i+1$ is a prismatic joint, the joint velocity along the z-axis Z_i of the joint

$${}^{i-1}v_i = {}^{i-1}v_{i-1} + {}^{i-1}\omega_{i-1} \times {}^{i-1}P_i + \dot{d}_{i+1}{}^{i}Z_i$$

$${}^{i}v_i = {}^{i}R_{i-1}({}^{i-1}v_{i-1} + {}^{i-1}\omega_{i-1} \times {}^{i-1}P_i) + \dot{d}_{i+1}{}^{i}Z_i$$



Velocity propagation summary

- Prismatic joint

$$ {}^{i}v_i = {}^{i}R_{i-1}({}^{i-1}v_{i-1} + {}^{i-1}\omega_{i-1} \times {}^{i-1}P_i) + \dot{d}_{i+1}{}^{i}Z_i $$

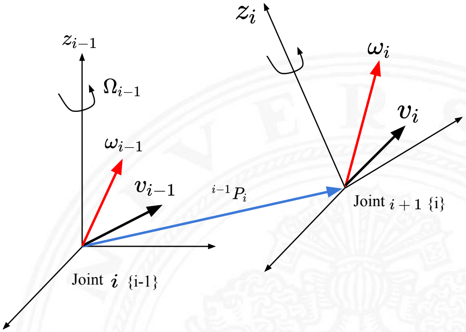
$$ {}^{i}\omega_i = {}^{i}R_{i-1}{}^{i-1}\omega_{i-1} $$

- Revolute joint

$$ {}^{i}v_i = {}^{i}R_{i-1}({}^{i-1}v_{i-1} + {}^{i-1}\omega_{i-1} \times {}^{i-1}P_i) $$

$$ {}^{i}\omega_i = {}^{i}R_{i-1}{}^{i-1}\omega_{i-1} + \dot{\theta}_{i+1}{}^{i}Z_i $$

$$ \begin{bmatrix} {}^{0}v_n \\ {}^{0}\omega_n \end{bmatrix} = \begin{bmatrix} {}^{0}R_n & 0 \\ 0 & {}^{0}R_n \end{bmatrix} \begin{bmatrix} {}^{n}v_n \\ {}^{n}\omega_n \end{bmatrix} $$



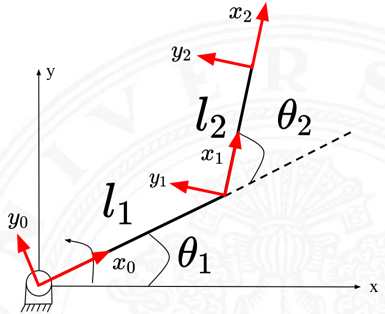
Example

Given the 2dof planar robot, find the velocity of the origin of {2} w.r.t. {2} and {0}.

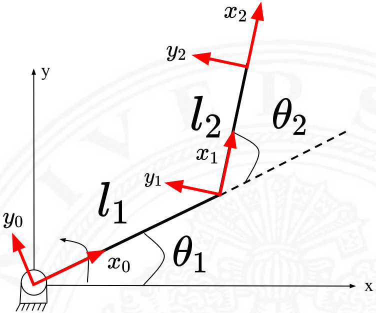
${}^{0}\omega_0 =$, ${}^{0}v_0 =$

${}^{1}\omega_1 =$

${}^{1}v_1 =$



Example



Velocity propagation

How to simplify the calculation of end-effector velocity?

Joint velocities $\Leftrightarrow$ End-effector velocities

$\Downarrow$

Jacobian

Jacobian of a manipulator

Definition

In the field of robotics, we generally use Jacobians to relate joint velocities to Cartesian velocities of the end-effecter.

$$x = f(q), \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_m \end{bmatrix} = \begin{bmatrix} f_1(q) \\ f_2(q) \\ \dots \\ f_n(q) \end{bmatrix} \tag{26}$$

- x is the Cartesian location of the end-effector
- m is number of degree of freedom in the Cartesian space
- Define $q = [q_1, q_2, ..q_n]^T$, $q_1, q_2, ..q_n$ are joint variables of an n-link manipulator

Jacobian of a manipulator (cont.)

By the chain rule of differentiation:

$$\delta x_1 = \frac{\partial f_1}{\partial q_1}\delta q_1 + \ldots + \frac{\partial f_1}{\partial q_n}\delta q_n$$

$$\vdots$$

$$\delta x_m = \frac{\partial f_m}{\partial q_1}\delta q_1 + \ldots + \frac{\partial f_m}{\partial q_n}\delta q_n$$

$$\delta x = \begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \cdots & \frac{\partial f_1}{\partial q_n} \\ \vdots & \ldots & \vdots \\ \frac{\partial f_m}{\partial q_1} & \cdots & \frac{\partial f_m}{\partial q_n} \end{bmatrix} \cdot \delta q \quad (27)$$

$$\delta x_{(m\times 1)} = J_{(m\times n)}\delta q_{(n\times 1)} \quad \textit{where} \quad J_{ij}(q) = \frac{\partial}{\partial q_j} f_i(q) \quad (28)$$

Jacobian of a manipulator (cont.)

$$\partial x_{(m \times 1)} = J_{(m \times n)} \partial q_{(n \times 1)}$$

$$\dot{x}_{(m \times 1)} = J_{(m \times n)} \dot{q}_{(n \times 1)}$$

- A Jacobian-matrix is a multidimensional representation of partial derivatives.
- If we divide both sides with the differential time element, we can think of the Jacobian as mapping velocities in q to those in x.
- Jacobians are time-varying linear transformations.

Jacobian of a manipulator (cont.)

- ${}^0\omega_n$ to be the angular velocity of the end effector
- ${}^0v_n$ is the linear velocity of the end effector
- The **Jacobian** matrix consists of two components, that solve the following equations:

$$ {}^0v_n = {}^0J_v\dot{q} \qquad \text{and} \qquad {}^0\omega_n = {}^0J_w\dot{q} $$

The manipulator Jacobian

$$ J = \begin{bmatrix} J_v \\ J_w \end{bmatrix}, \quad \begin{bmatrix} {}^0v_n \\ {}^0\omega_n \end{bmatrix} = \begin{bmatrix} J_v \\ J_w \end{bmatrix} \dot{q} \tag{29} $$

Angular velocity Jacobian

Angular velocity ${}^{i-1}\omega_i$ is:

$$ {}^{i-1}\omega_i = {}^{i-1}\omega_{i-1} + {}^{i-1}R_i \dot{\theta}_{i+1} {}^{i}Z_i $$

We get:

$$ \begin{aligned} {}^{0}\omega_n &= p_1 \dot{q}_1 {}^{0}Z_0 + p_2 \dot{q}_2 {}^{0}R_1 {}^{1}Z_1 + \ldots + p_n \dot{q}_n {}^{0}R_{n-1} {}^{n-1}Z_{n-1} \\ &= p_1 \dot{q}_1 {}^{0}Z_0 + p_2 \dot{q}_2 {}^{0}Z_1 + \ldots + p_n \dot{q}_n {}^{0}Z_{n-1} \end{aligned} $$

where:

$$ p_i = \begin{cases} 0 & \text{if joint i is prismatic} \\ 1 & \text{if joint i is revolute} \end{cases} \tag{30} $$

Angular velocity Jacobian (cont.)

The Angular Velocity Jacobian

$$J_w = [p_1{}^0Z_0 \quad p_2{}^0Z_1 \quad \dots \quad p_n{}^0Z_{n-1}] \tag{31}$$

(Hint: J_w is a $3xn$ matrix.)

Linear velocity Jacobian

The linear velocity of the end effector is: $^0v_n = {}^0\dot{x}_n = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}$

By the chain rule of differentiation:

$$^0\dot{x}_n = \frac{\partial^0 x_n}{\partial q_1}\dot{q}_1 + \frac{\partial^0 x_n}{\partial q_2}\dot{q}_2 + \ldots + \frac{\partial^0 x_n}{\partial q_n}\dot{q}_n$$

therefore the linear part of the Jacobian is:

$$J_v = \begin{bmatrix} \frac{\partial^0 x_n}{\partial q_1} & \frac{\partial^0 x_n}{\partial q_2} & \ldots & \frac{\partial^0 x_n}{\partial q_n} \end{bmatrix} \tag{32}$$

Computing the final Jacobian

Two approaches:

1. derive v, ω for each link until the end-effector
2. use the explicit form

Computing the final Jacobian

- get ${}^0J_v$

$$ {}^0T_6 = \begin{bmatrix} {}^0R_N & {}^0P_N \\ 0 & 1 \end{bmatrix} \rightarrow {}^0x \longrightarrow {}^0v_n \longrightarrow {}^0J_v $$

- get ${}^0J_\omega$

$$ J_w = [p_1 {}^0Z_0 \quad p_2 {}^0Z_1 \quad \dots \quad p_n {}^0Z_{n-1}] $$

- ${}^0x_i$ is equal to the first three elements of the 4th column of matrix ${}^0T_i$
- ${}^0Z_i$ is equal to the first three elements of the 3rd column of matrix ${}^0T_i$

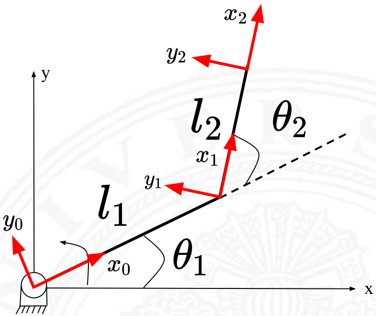
${}^0T_i$ has to be computed for every joint.

Example1

$$^{0}\omega_2 = {}^{0}R_2 {}^{2}\omega_2 = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 + \dot{\theta}_2 \end{bmatrix}$$

$$^{0}v_2 = {}^{0}R_2 {}^{2}v_2 = \begin{bmatrix} -l_1 s_1 \dot{\theta}_1 - l_2 s_{12}(\dot{\theta}_1 + \dot{\theta}_2) \\ l_1 c_1 \dot{\theta}_1 + l_2 c_{12}(\dot{\theta}_1 + \dot{\theta}_2) \\ 0 \end{bmatrix}$$

Give the ${}^{0}J$ Jacobian matrix.



Example2

For a 3-DOF robot, given the following transformation matrices, find the Jacobian ${}^0J$.

$$
{}^0T_1 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & h \\ 0 & 0 & 0 & 1 \end{bmatrix}, {}^1T_2 = \begin{bmatrix} c_2 & -s_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, {}^2T_3 = \begin{bmatrix} c_3 & -s_3 & 0 & e \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, {}^3T_4 = \begin{bmatrix} 1 & 0 & 0 & f \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},
$$

where h, e, f are the length of the $1^{st}, 2^{nd} and 3^{rd}$ link, respectively.

$$
{}^0T_4 = \begin{bmatrix} c_1c_{23} & -c_1s_{23} & s_1 & ec_1c_2 + fc_1c_{23} \\ s_1c_{23} & -s_1c_{23} & -c_1 & es_1c_2 + fs_1c_{23} \\ s_{23} & c_{23} & 0 & h + es_2 + fs_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

Example2

Calculate ${}^0T_1, {}^0T_2, {}^0T_3, {}^0T_4$:

$$
{}^0T_1 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & h \\ 0 & 0 & 0 & 1 \end{bmatrix}, {}^0T_2 = {}^0T_1 {}^1T_2 = \begin{bmatrix} c_1c_2 & -s_2c_1 & s_1 & 0 \\ s_1c_2 & -s_1s_2 & -c_1 & 0 \\ s_2 & c_2 & 0 & h \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^0T_3 = {}^0T_2 {}^2T_3 = \begin{bmatrix} -s_2s_3c_1 + c_1c_2c_3 & -s_2c_1c_3 - s_3c_1c_2 & s_1 & ec_1c_2 \\ -s_1s_2s_3 + s_1c_2c_3 & -s_1s_2c_3 - s_1s_3c_2 & -c_1 & es_1c_2 \\ s_2c_3 + s_3c_2 & -s_2s_3 + c_2c_3 & 0 & es_2 + h \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^0T_4 = \begin{bmatrix} c_1c_{23} & -c_1s_{23} & s_1 & ec_1c_2 + fc_1c_{23} \\ s_1c_{23} & -s_1c_{23} & -c_1 & es_1c_2 + fs_1c_{23} \\ s_{23} & c_{23} & 0 & h + es_2 + fs_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

Example2 (cont.)

$$
{}^{0}J = \begin{bmatrix} J_v \\ J_w \end{bmatrix} = \begin{bmatrix} -es_1c_2 - fs_1c_{23} & -ec_1s_2 - fc_1s_{23} & -fc_1s_{23} \\ ec_1c_2 + fc_1c_{23} & -es_1s_2 - fs_1s_{23} & -fs_1s_{23} \\ 0 & ec_2 + fc_{23} & fc_{23} \\ 0 & s_1 & s_1 \\ 0 & -c_1 & -c_1 \\ 1 & 0 & 0 \end{bmatrix}
$$

Changing a Jacobian's frame of reference

Given a Jacobian written in frame {B},

$$\begin{bmatrix} {}^B v_n \\ {}^B \omega_n \end{bmatrix} = \begin{bmatrix} {}^B J_v \\ {}^B J_w \end{bmatrix} \dot{q}$$

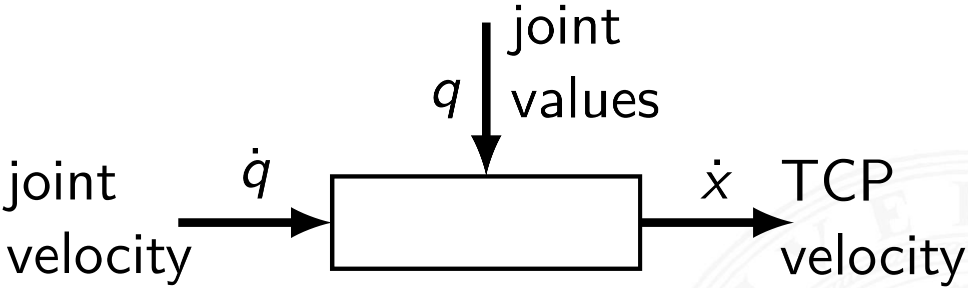
A 6 x 1 Cartesian velocity vector given in {B} is described relative to {A} by the transformation

$$\begin{bmatrix} {}^A v_n \\ {}^A \omega_n \end{bmatrix} = \begin{bmatrix} {}^A R_B & 0 \\ 0 & {}^A R_B \end{bmatrix} \begin{bmatrix} {}^B v_n \\ {}^B \omega_n \end{bmatrix}$$

Hence, we can get

$$\begin{bmatrix} {}^A v_n \\ {}^A \omega_n \end{bmatrix} = \begin{bmatrix} {}^A R_B & 0 \\ 0 & {}^A R_B \end{bmatrix} \begin{bmatrix} {}^B J_v \\ {}^B J_w \end{bmatrix} \dot{q} \tag{33}$$

Inverse Jacobian



Question

Is the Jacobian invertible?

If it is, then: $$\dot{\mathbf{q}} = J^{-1}(\mathbf{q})\dot{\mathbf{x}}$$

$\Longrightarrow$ to move the the end effector of the robot in Cartesian Space with a certain velocity.

Singular Configurations

For most manipulators there exist values of $\mathbf{q}$ where the Jacobian gets singular.

Singularity

$$\det J = 0 \Longrightarrow J \text{ is not invertible}$$

Such configurations are called singularities of the manipulator.

Singularities

From the Task Space perspective:

- reduce the degree of freedom in velocity domain in task space

From the Joint Space perspective:

- Infinite solutions to the inverse kinematics problem may exist
- Near the singularity, small velocities in operational space may cause large velocities in the joint space.

Singularities (cont.)

Two Main types of Singularities:

- **Workspace boundary singularities** occur when the manipulator is fully stretched out or folded back on itself in such a way that the end-effector is at or very near the boundary of the workspace.
- **Workspace internal singularities** occur away from the workspace boundary; they generally are caused by a lining up of two or more joint axes

Singular Configurations – Workarounds

$N = 6$ For fully actuated robots, the Jacobian (6×6) is invertible

$$\delta x_{(m\times 1)} = J_{(m\times n)} \delta q_{(n\times 1)} \quad where \quad J_{ij}(q) = \frac{\partial}{\partial q_j} f_i(q)$$

- m is number of degree of freedom of the manipulator in the Cartesian space
- n is the number of joint variables of the manipulator

$N = 6$ For fully actuated robots, the Jacobian (6×6) is invertible

$N < 6$ Under actuated robots $(6 \times N)$

$\Longrightarrow$ remove some spatial degrees of freedom, get a square Jacobian matrix.

Example:

$$\begin{bmatrix} {}^{T_6}d_x \\ {}^{T_6}d_y \end{bmatrix} = J_{2\times 2} \begin{bmatrix} dq_1 \\ dq_2 \end{bmatrix}$$

for a 2-joint planar manipulator

Singular Configurations – Workarounds

$N = 6$ For fully actuated robots, the Jacobians (6×6) are invertible

$N < 6$ Under actuated robots $(6 \times N)$

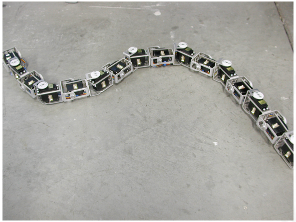
$\Longrightarrow$ remove some spatial degrees of freedom

$N > 6$ Over actuated robots $(6 \times N)$

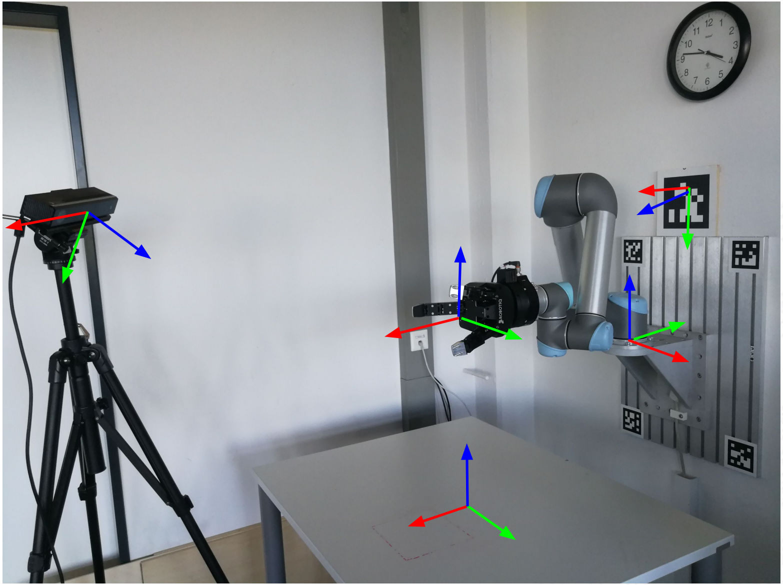
- have spare joints
- use the `pseudoinverse of J`

$$\dot{q} = J(q)^{+} v \tag{34}$$

$$J^{+} = (J^T J)^{-1} J^T \tag{35}$$



UR5 example



23

[23] https://www.youtube.com/watch?v=6Wmw4lUHIX8

Bibliography

[1] G.-Z. Yang, R. J. Full, N. Jacobstein, P. Fischer, J. Bellingham, H. Choset, H. Christensen, P. Dario, B. J. Nelson, and R. Taylor, "Ten robotics technologies of the year," 2019.

[2] J. K. Yim, E. K. Wang, and R. S. Fearing, "Drift-free roll and pitch estimation for high-acceleration hopping," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8986–8992, IEEE, 2019.

[3] J. F. Engelberger, *Robotics in service*.
MIT Press, 1989.

[4] K. Fu, R. González, and C. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*.
McGraw-Hill series in CAD/CAM robotics and computer vision, McGraw-Hill, 1987.

[5] R. Paul, *Robot Manipulators: Mathematics, Programming, and Control: the Computer Control of Robot Manipulators*.
Artificial Intelligence Series, MIT Press, 1981.

[6] J. Craig, *Introduction to Robotics: Pearson New International Edition: Mechanics and Control*.
Always learning, Pearson Education, Limited, 2013.

[7] W. Böhm, G. Farin, and J. Kahmann, "A Survey of Curve and Surface Methods in CAGD," *Comput. Aided Geom. Des.*, vol. 1, pp. 1–60, July 1984.

[8] J. Zhang and A. Knoll, "Constructing Fuzzy Controllers with B-spline Models - Principles and Applications," *International Journal of Intelligent Systems*, vol. 13, no. 2-3, pp. 257–285, 1998.

[9] M. Eck and H. Hoppe, "Automatic Reconstruction of B-spline Surfaces of Arbitrary Topological Type," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, (New York, NY, USA), pp. 325–334, ACM, 1996.

[10] M. C. Ferch, *Lernen von Montagestrategien in einer verteilten Multiroboterumgebung*. PhD thesis, Bielefeld University, 2001.

[11] J. H. Reif, "Complexity of the Mover's Problem and Generalizations - Extended Abstract," *Proceedings of the 20th Annual IEEE Conference on Foundations of Computer Science*, pp. 421–427, 1979.

Bibliography (cont.)

[12] J. T. Schwartz and M. Sharir, "A Survey of Motion Planning and Related Geometric Algorithms," *Artificial Intelligence*, vol. 37, no. 1, pp. 157–169, 1988.

[13] J. Canny, *The Complexity of Robot Motion Planning*.
MIT press, 1988.

[14] T. Lozano-Pérez, J. L. Jones, P. A. O'Donnell, and E. Mazer, *Handey: A Robot Task Planner*.
Cambridge, MA, USA: MIT Press, 1992.

[15] O. Khatib, "The Potential Field Approach and Operational Space Formulation in Robot Control," in *Adaptive and Learning Systems*, pp. 367–377, Springer, 1986.

[16] J. Barraquand, L. Kavraki, R. Motwani, J.-C. Latombe, T.-Y. Li, and P. Raghavan, "A Random Sampling Scheme for Path Planning," in *Robotics Research* (G. Giralt and G. Hirzinger, eds.), pp. 249–264, Springer London, 1996.

[17] R. Geraerts and M. H. Overmars, "A Comparative Study of Probabilistic Roadmap Planners," in *Algorithmic Foundations of Robotics V*, pp. 43–57, Springer, 2004.

Bibliography (cont.)

[18] K. Nishiwaki, J. Kuffner, S. Kagami, M. Inaba, and H. Inoue, "The Experimental Humanoid Robot H7: A Research Platform for Autonomous Behaviour," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 365, no. 1850, pp. 79–107, 2007.

[19] R. Brooks, "A robust layered control system for a mobile robot," *Robotics and Automation, IEEE Journal of*, vol. 2, pp. 14–23, Mar 1986.

[20] M. J. Mataric, "Interaction and intelligent behavior.," tech. rep., DTIC Document, 1994.

[21] M. P. Georgeff and A. L. Lansky, "Reactive reasoning and planning.," in *AAAI*, vol. 87, pp. 677–682, 1987.

[22] J. Zhang and A. Knoll, *Integrating Deliberative and Reactive Strategies via Fuzzy Modular Control*, pp. 367–385.
Heidelberg: Physica-Verlag HD, 2001.

[23] J. S. Albus, "The nist real-time control system (rcs): an approach to intelligent systems research," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 9, no. 2-3, pp. 157–174, 1997.

Bibliography (cont.)

[24] A. Meystel, "Nested hierarchical control," 1993.

[25] G. Saridis, "Machine-intelligent robots: A hierarchical control approach," in *Machine Intelligence and Knowledge Engineering for Robotic Applications* (A. Wong and A. Pugh, eds.), vol. 33 of *NATO ASI Series*, pp. 221–234, Springer Berlin Heidelberg, 1987.

[26] T. Fukuda and T. Shibata, "Hierarchical intelligent control for robotic motion by using fuzzy, artificial intelligence, and neural network," in *Neural Networks, 1992. IJCNN., International Joint Conference on*, vol. 1, pp. 269–274 vol.1, Jun 1992.

[27] R. C. Arkin and T. Balch, "Aura: principles and practice in review," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 9, no. 2-3, pp. 175–189, 1997.

[28] E. Gat, "Integrating reaction and planning in a heterogeneous asynchronous architecture for mobile robot navigation," *ACM SIGART Bulletin*, vol. 2, no. 4, pp. 70–74, 1991.

[29] L. Einig, *Hierarchical Plan Generation and Selection for Shortest Plans based on Experienced Execution Duration*.
Master thesis, Universität Hamburg, 2015.

Bibliography (cont.)

[30] J. Craig, *Introduction to Robotics: Mechanics & Control. Solutions Manual.* Addison-Wesley Pub. Co., 1986.

[31] H. Siegert and S. Bocionek, *Robotik: Programmierung intelligenter Roboter: Programmierung intelligenter Roboter.* Springer-Lehrbuch, Springer Berlin Heidelberg, 2013.

[32] R. Schilling, *Fundamentals of robotics: analysis and control.* Prentice Hall, 1990.

[33] T. Yoshikawa, *Foundations of Robotics: Analysis and Control.* Cambridge, MA, USA: MIT Press, 1990.

[34] M. Spong, *Robot Dynamics And Control.* Wiley India Pvt. Limited, 2008.