Lecture «Robot Dynamics»: Kinematics 3

151-0851-00 V

lecture: CAB G11   Tuesday 10:15 – 12:00, every week
exercise: HG E1.2   Wednesday 8:15 – 10:00, according to schedule (about every 2nd week)

Marco Hutter, Roland Siegwart, and Thomas Stastny

|					
19.09.2017	Intro and Outline	Course Introduction; Recapitulation Position, Linear Velocity			
26.09.2017	Kinematics 1	Rotation and Angular Velocity; Rigid Body Formulation, Transformation	26.09.2017	Exercise 1a	Kinematics Modeling the ABB arm
03.10.2017	Kinematics 2	Kinematics of Systems of Bodies; Jacobians	03.10.2017	Exercise 1b	Differential Kinematics of the ABB arm
10.10.2017	Kinematics 3	Kinematic Control Methods: Inverse Differential Kinematics, Inverse Kinematics; Rotation Error; Multi-task Control	10.10.2017	Exercise 1c	Kinematic Control of the ABB Arm
17.10.2017	Dynamics L1	Multi-body Dynamics	17.10.2017	Exercise 2a	Dynamic Modeling of the ABB Arm
24.10.2017	Dynamics L2	Floating Base Dynamics	24.10.2017		
31.10.2017	Dynamics L3	Dynamic Model Based Control Methods	31.10.2017	Exercise 2b	Dynamic Control Methods Applied to the ABB arm
07.11.2017	Legged Robot	Dynamic Modeling of Legged Robots & Control	07.11.2017	Exercise 3	Legged robot
14.11.2017	Case Studies 1	Legged Robotics Case Study	14.11.2017		
21.11.2017	Rotorcraft	Dynamic Modeling of Rotorcraft & Control	21.11.2017	Exercise 4	Modeling and Control of Multicopter
28.11.2017	Case Studies 2	Rotor Craft Case Study	28.11.2017		
05.12.2017	Fixed-wing	Dynamic Modeling of Fixed-wing & Control	05.12.2017	Exercise 5	Fixed-wing Control and Simulation
12.12.2017	Case Studies 3	Fixed-wing Case Study (Solar-powered UAVs - AtlantikSolar, Vertical Take-off and Landing UAVs – Wingtra)			
19.12.2017	Summery and Outlook	Summery; Wrap-up; Exam			

Multi-body Kinematics
Intro

- Machines are built and controlled to
  - achieve extremely accurate positions,
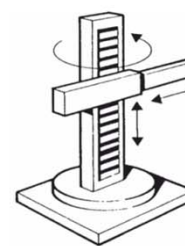 - independent of the load the robot carries

  - Very stiff structure
  - Play-free gears and transmissions
  - High-accurate joint sensors

  - End-effector accuracy +/- 0.02mm!
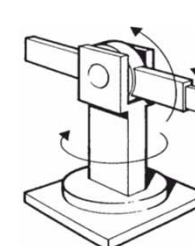
- Large variety of robot arms
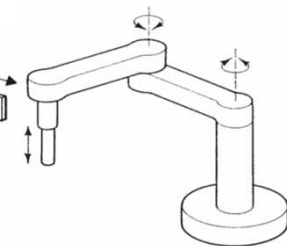

HumorOn.com
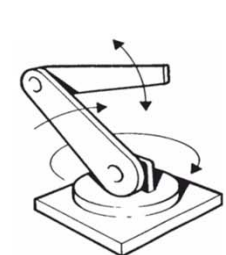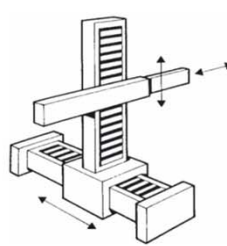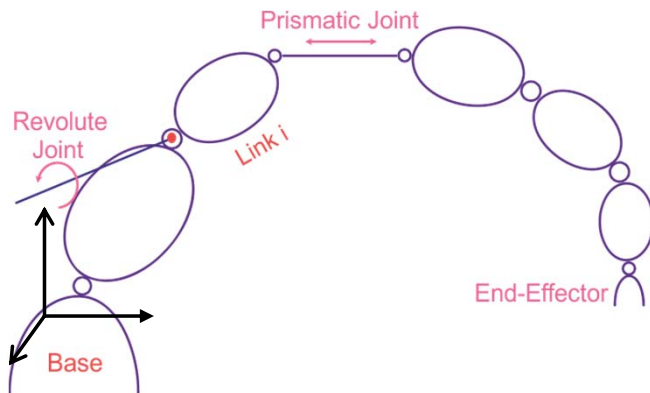

Antropomorphic   Cylindric   Polar   Scara   Cartesian

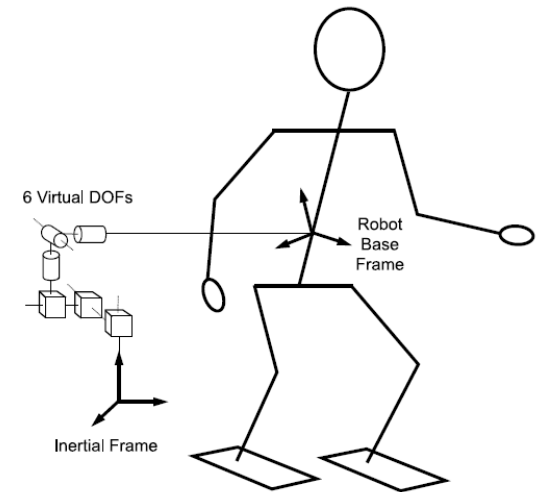# Fixed Base vs. Floating Base Robot

- Base frame is rigidly connected to ground
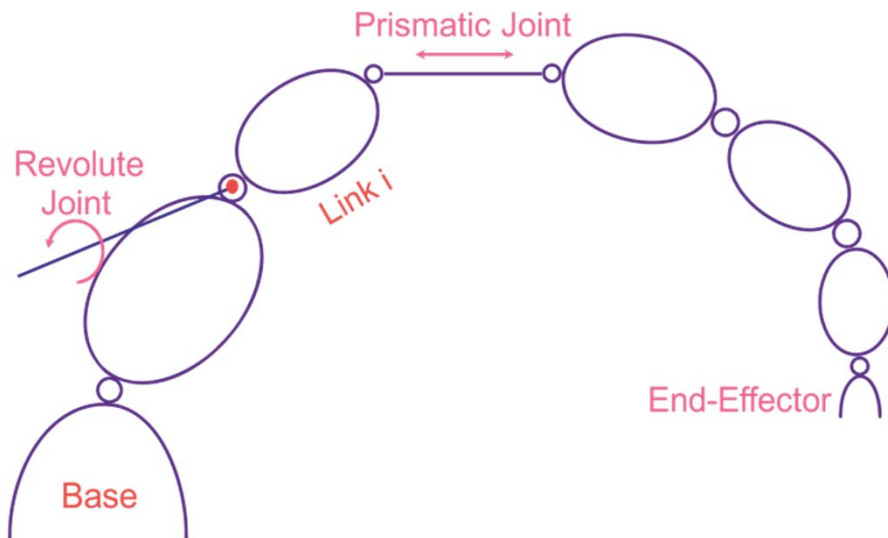 - Often indicated as CS 0

- Base frame is free floating
  - Often indicated as CS $\mathcal{B}$ (base)
  - 6 unactuated DOFs!
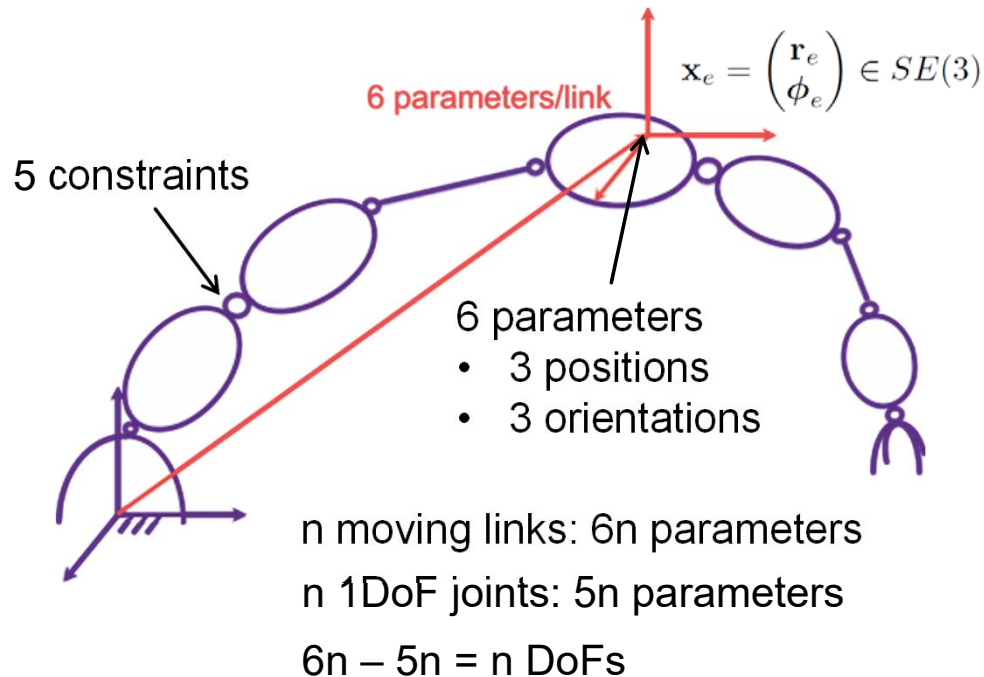
Classical Serial Kinematic Linkages
Generalized robot arm

- n_j joints
  - revolute (1DOF)
  - prismatic (1DOF)
- $n_l = n_j + 1$ links
  - n_j moving links
  - 1 fixed link

Configuration Parameters
Generalized coordinates

$$\mathbf{x}_e = \begin{pmatrix} \mathbf{r}_e \\ \boldsymbol{\phi}_e \end{pmatrix} \in SE(3)$$

6 parameters/link

5 constraints

6 parameters
- 3 positions
- 3 orientations

n moving links: 6n parameters

n 1DoF joints: 5n parameters

6n – 5n = n DoFs

Generalized coordinates

A set of scalar parameters **q** that describe the robot's configuration

- Must be **complete**
- (Must be **independent**)
  => minimal coordinates
- Is **not unique**

$$\mathbf{q} = \begin{pmatrix} q_1 \\ \vdots \\ q_{n_j} \end{pmatrix} \in \mathbb{R}^{n_j}$$

Degrees of Freedom
- Nr of minimal coordinates

End-effector Configuration Parameters

- End-effector configuration parameters
  - A set of m parameters that completely specify the end-effector position and orientation with respect to $\mathcal{I}$

$$\boldsymbol{\chi}_e = \begin{pmatrix} \boldsymbol{\chi}_{eP} \\ \boldsymbol{\chi}_{eR} \end{pmatrix} = \begin{pmatrix} \chi_1 \\ \vdots \\ \chi_m \end{pmatrix} \in \mathbb{R}^m$$

- Operational space coordinates
  - the m_0 configuration parameters are independent => m_0 number of degrees of freedom of end-effector

$$\boldsymbol{\chi}_o = \begin{pmatrix} \boldsymbol{\chi}_{oP} \\ \boldsymbol{\chi}_{oR} \end{pmatrix} = \begin{pmatrix} \chi_1 \\ \vdots \\ \chi_{m_0} \end{pmatrix}$$
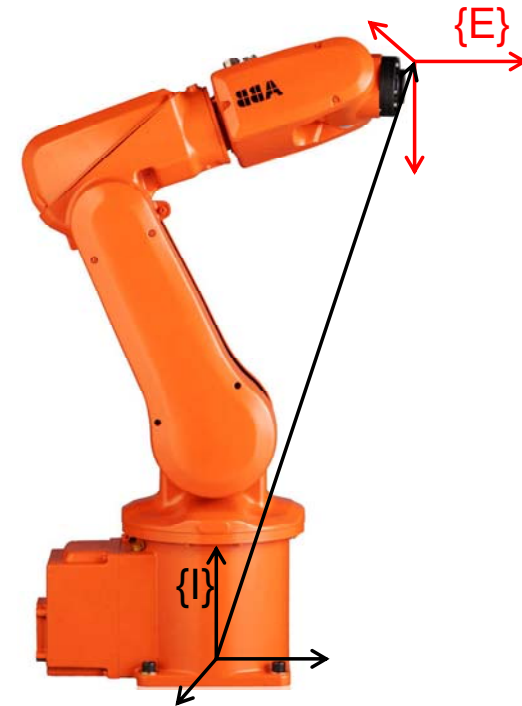
End-effector Configuration Parameters
Example

- Most general robot arm:
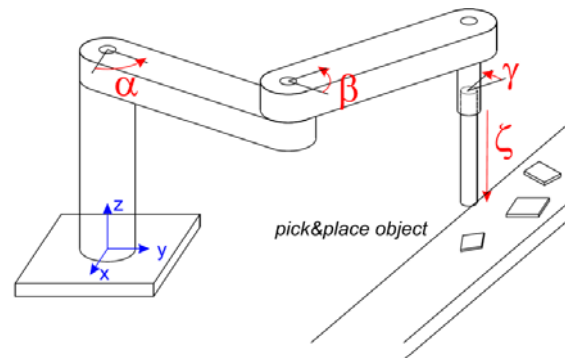 - $q =$
  - $m_e =$ $m_o =$
  - $\chi_e =$ $\chi_o =$
- SCARA robot arm
  - $q =$
  - $m_e =$ $m_o =$
  - $\chi_e =$ $\chi_o =$
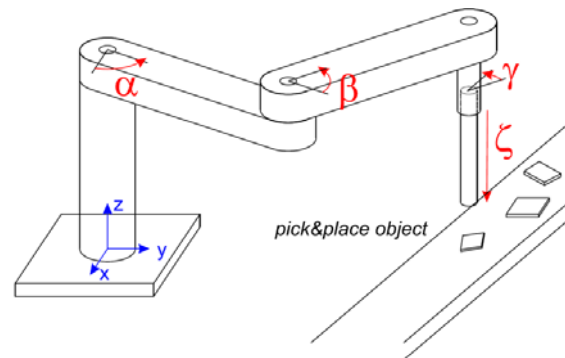- ANYpulator: robot arm with 4 rotational joints
  - $q =$
  - $m_e =$ $m_o =$
  - $\chi_e =$ $\chi_o =$

pick&place object
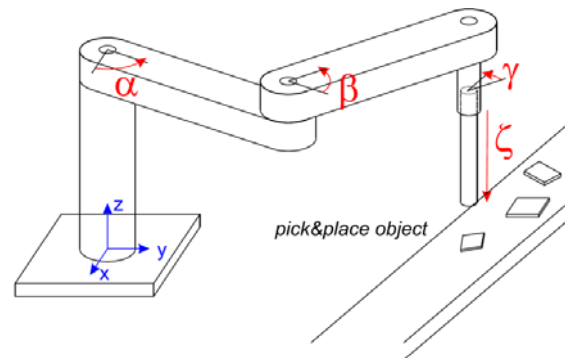
End-effector Configuration Parameters
Example

- Most general robot arm:
  - $\mathbf{q} = \left(q_1 ... q_{n_j} \right)$
  - $m_e = 6$ $m_o = 6$
  - $\boldsymbol{\chi}_e = \left(x, y, z, \alpha_x, \beta_y, \gamma_z \right)$ $\boldsymbol{\chi}_o = \left(x, y, z, \alpha_x, \beta_y, \gamma_z \right)$

- SCARA robot arm
  - $\mathbf{q} = \left(\alpha, \beta, \gamma, \zeta \right)$
  - $m_e = 6$ $m_o = 4$
  - $\boldsymbol{\chi}_e = \left(x, y, z, \alpha_x, \beta_y, \gamma_z \right)$ $\boldsymbol{\chi}_o = \left(x, y, z, \gamma_z \right)$

- ANYpulator: robot arm with 4 rotational joints
  - $\mathbf{q} = \left(q_1, q_2, q_3, q_4 \right)$
  - $m_e = 6$ $m_o = 4$
  - $\boldsymbol{\chi}_e = \left(x, y, z, \alpha_x, \beta_y, \gamma_z \right)$ $\boldsymbol{\chi}_o =$

pick&place object

End-effector Configuration Parameters
Example

- Most general robot arm:
  - $\mathbf{q} = \left(q_1 ... q_{n_j} \right)$
  - $m_e = 6$ $m_o = 6$
  - $\boldsymbol{\chi}_e = \left(x, y, z, \alpha_x, \beta_y, \gamma_z \right)$ $\boldsymbol{\chi}_o = \left(x, y, z, \alpha_x, \beta_y, \gamma_z \right)$

- SCARA robot arm
  - $\mathbf{q} = \left(\alpha, \beta, \gamma, \zeta \right)$
  - $m_e = 6$ $m_o = 4$
  - $\boldsymbol{\chi}_e = \left(x, y, z, \alpha_x, \beta_y, \gamma_z \right)$ $\boldsymbol{\chi}_o = \left(x, y, z, \gamma_z \right)$

- ANYpulator: robot arm with 4 rotational joints
  - $\mathbf{q} = \left(q_1, q_2, q_3, q_4 \right)$
  - $m_e = 6$ $m_o = 4$
  - $\boldsymbol{\chi}_e = \left(x, y, z, \alpha_x, \beta_y, \gamma_z \right)$ $\boldsymbol{\chi}_o =$

pick&place object
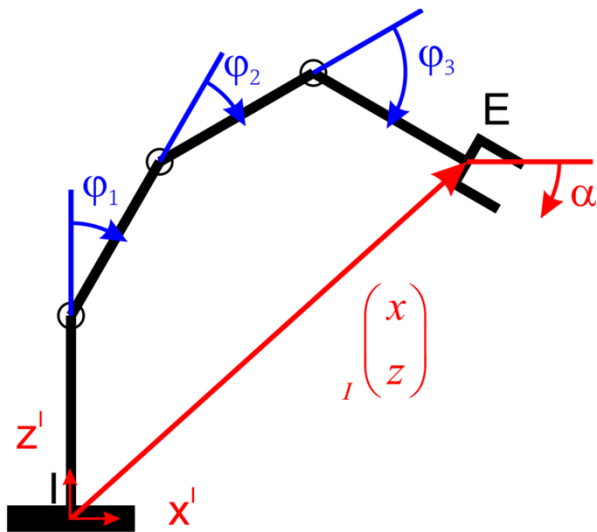
End-effector Configuration Parameters
Simple example

- Planar robot arm
  - 3 revolute joints
  - 1 end-effector (gripper) <= *don't consider this for the moment*

- What are the joint coordinates (generalized coordinates)?

- What are the end-effector parameters?
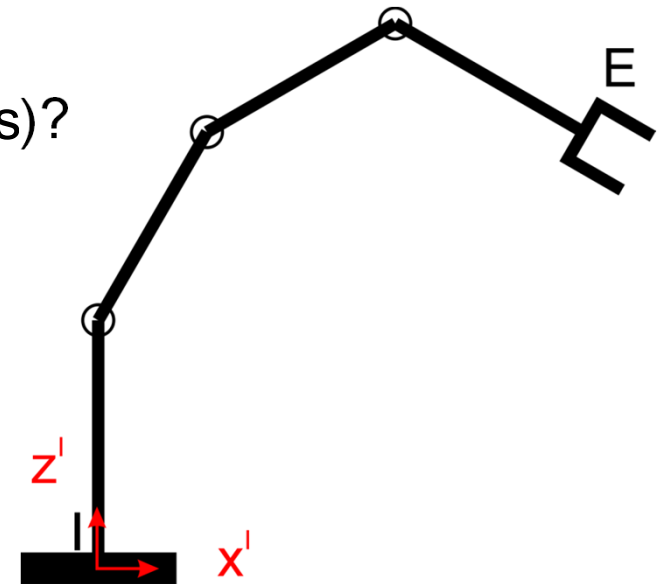
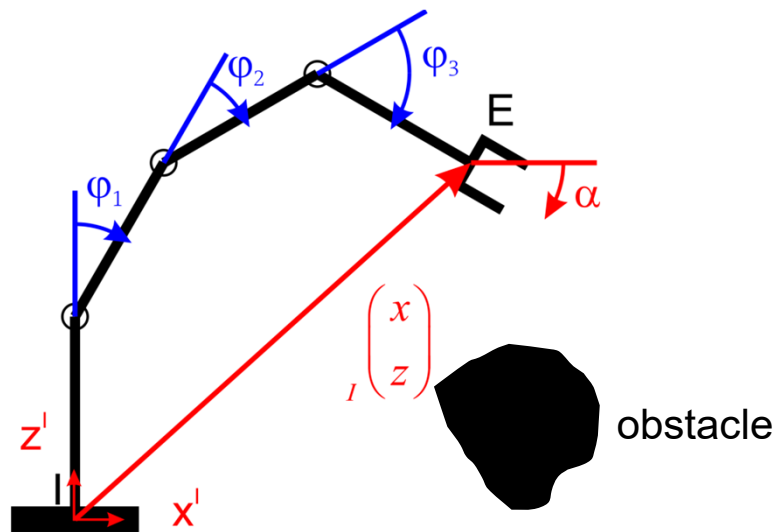# Configuration Space ⇔ Joint Space

- **Joint Coordinates**

- **Operational Coordinates**

Configuration Space ⇔ Joint Space

- **Joint Coordinates**   =>   **Joint Space**

obstacle
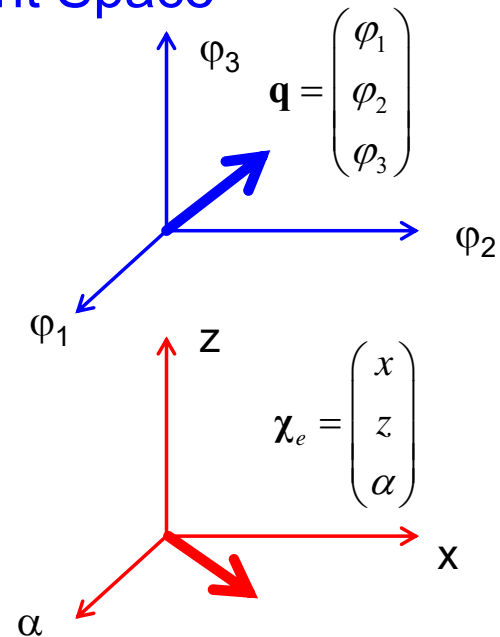
- **Operational Coordinates**   =>   **Operational Space**

Forward Kinematics

- End-effector configuration as a function of generalized coordinates

$$\chi_e = \chi_e(\mathbf{q}) \in \mathbb{R}^{n_e}$$

- For multi-body system, use transformation matrices

$$\mathbf{T}_{\mathcal{IE}}(\mathbf{q}) = \mathbf{T}_{\mathcal{I}0} \cdot \left(\prod_{k=1}^{n_j} \mathbf{T}_{k-1,k}(q_k) \right) \cdot \mathbf{T}_{n_j \mathcal{E}} = \begin{bmatrix} \mathbf{C}_{\mathcal{IE}}(\mathbf{q}) & {}_{\mathcal{I}}\mathbf{r}_{IE}(\mathbf{q}) \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix}$$

Forward Kinematics
Simple example

- What is the end-effector configuration as a function of generalized coordinates?

$$\mathbf{q} = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix} = \begin{pmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \end{pmatrix}$$
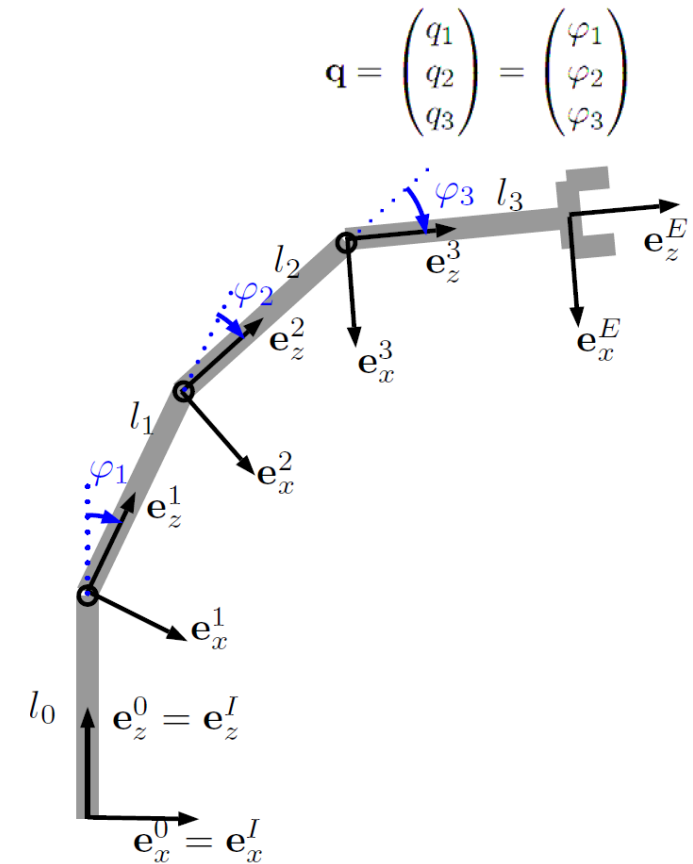
$$\mathbf{T}_{IE} = \mathbf{T}_{I0} \cdot \mathbf{T}_{01} \cdot \mathbf{T}_{12} \cdot \mathbf{T}_{23} \cdot \mathbf{T}_{3E}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_1 & 0 & s_1 & 0 \\ 0 & 1 & 0 & 0 \\ -s_1 & 0 & c_1 & l_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_2 & 0 & s_2 & 0 \\ 0 & 1 & 0 & 0 \\ -s_2 & 0 & c_2 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_3 & 0 & s_3 & 0 \\ 0 & 1 & 0 & 0 \\ -s_3 & 0 & c_3 & l_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & l_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \ldots = \begin{bmatrix} c_{123} & 0 & s_{123} & l_1 s_1 + l_2 s_{12} + l_3 s_{123} \\ 0 & 1 & 0 & 0 \\ -s_{123} & 0 & c_{123} & l_0 + l_1 c_1 + l_2 c_{12} + l_3 c_{123} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{\chi}_{eP}(\mathbf{q}) = \begin{pmatrix} x \\ z \end{pmatrix} = \begin{pmatrix} l_1 \sin(q_1) + l_2 \sin(q_1 + q_2) + l_3 \sin(q_1 + q_2 + q_3) \\ l_0 + l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) + l_3 \cos(q_1 + q_2 + q_3) \end{pmatrix}$$

$$\boldsymbol{\chi}_{eR}(\mathbf{q}) = \chi_{eR}(\mathbf{q}) = q_1 + q_2 + q_3$$

Forward Differential Kinematics
Analytical Jacobian

- Forward Kinematics $\quad \boldsymbol{\chi}_e = \begin{pmatrix} \boldsymbol{\chi}_{e_P} \\ \boldsymbol{\chi}_{e_R} \end{pmatrix} = \boldsymbol{\chi}_e(\mathbf{q})$

$$\mathbf{T}_{\mathcal{I}\mathcal{E}}(\mathbf{q}) = \mathbf{T}_{\mathcal{I}0} \cdot \left(\prod_{k=1}^{n_j} \mathbf{T}_{k-1,k}(q_k) \right) \cdot \mathbf{T}_{n_j\mathcal{E}} = \begin{bmatrix} \mathbf{C}_{\mathcal{I}\mathcal{E}}(\mathbf{q}) & {}_{\mathcal{I}}\mathbf{r}_{IE}(\mathbf{q}) \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix}$$

- Forward **Differential** Kinematics

Forward Differential Kinematics
Analytical Jacobian

- Forward Kinematics $\quad \boldsymbol{\chi}_e = \begin{pmatrix} \boldsymbol{\chi}_{e_P} \\ \boldsymbol{\chi}_{e_R} \end{pmatrix} = \boldsymbol{\chi}_e(\mathbf{q})$

$$\mathbf{T}_{\mathcal{IE}}(\mathbf{q}) = \mathbf{T}_{\mathcal{I}0} \cdot \left(\prod_{k=1}^{n_j} \mathbf{T}_{k-1,k}(q_k) \right) \cdot \mathbf{T}_{n_j \mathcal{E}} = \begin{bmatrix} \mathbf{C}_{\mathcal{IE}}(\mathbf{q}) & {}_{\mathcal{I}}\mathbf{r}_{IE}(\mathbf{q}) \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix}$$

- Forward **Differential** Kinematics
  - Analytic:

$$\boldsymbol{\chi}_e + \delta\boldsymbol{\chi}_e = \boldsymbol{\chi}_e(\mathbf{q} + \delta\mathbf{q}) = \boldsymbol{\chi}_e(\mathbf{q}) + \frac{\partial \boldsymbol{\chi}_e(\mathbf{q})}{\partial \mathbf{q}} \delta\mathbf{q} + O(\delta\mathbf{q}^2)$$

$$\delta\boldsymbol{\chi}_e \approx \frac{\partial \boldsymbol{\chi}_e(\mathbf{q})}{\partial \mathbf{q}} \delta\mathbf{q} = \mathbf{J}_{eA}(\mathbf{q})\,\delta\mathbf{q} \quad \text{with} \quad \mathbf{J}_{eA} = \frac{\partial \boldsymbol{\chi}_e}{\partial \mathbf{q}} = \begin{bmatrix} \frac{\partial \chi_1}{\partial q_1} & \cdots & \frac{\partial \chi_1}{\partial q_{n_j}} \\ \vdots & \ddots & \vdots \\ \frac{\partial \chi_m}{\partial q_1} & \cdots & \frac{\partial \chi_m}{\partial q_{n_j}} \end{bmatrix}$$

$$\boxed{\dot{\boldsymbol{\chi}}_e = \mathbf{J}_{eA}(\mathbf{q})\,\dot{\mathbf{q}} \quad \text{with } \mathbf{J}_{eA}(\mathbf{q}) \in \mathbb{R}^{m_e \times n_j}}$$
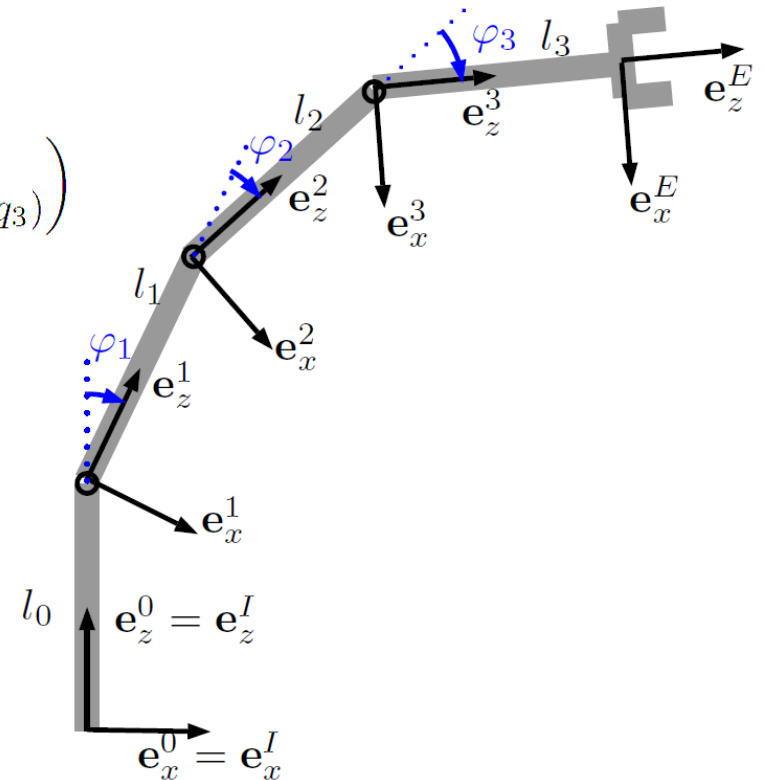
Analytical Jacobian
Planar robot arm

- Given (from last example)

$$\chi_{eP}(\mathbf{q}) = \begin{pmatrix} x \\ z \end{pmatrix} = \begin{pmatrix} l_1 \sin(q_1) + l_2 \sin(q_1 + q_2) + l_3 \sin(q_1 + q_2 + q_3) \\ l_0 + l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) + l_3 \cos(q_1 + q_2 + q_3) \end{pmatrix}$$

$$\chi_{eR}(\mathbf{q}) = \chi_{eR}(\mathbf{q}) = q_1 + q_2 + q_3$$

- Determine the analytical Jacobian

Analytical Jacobian
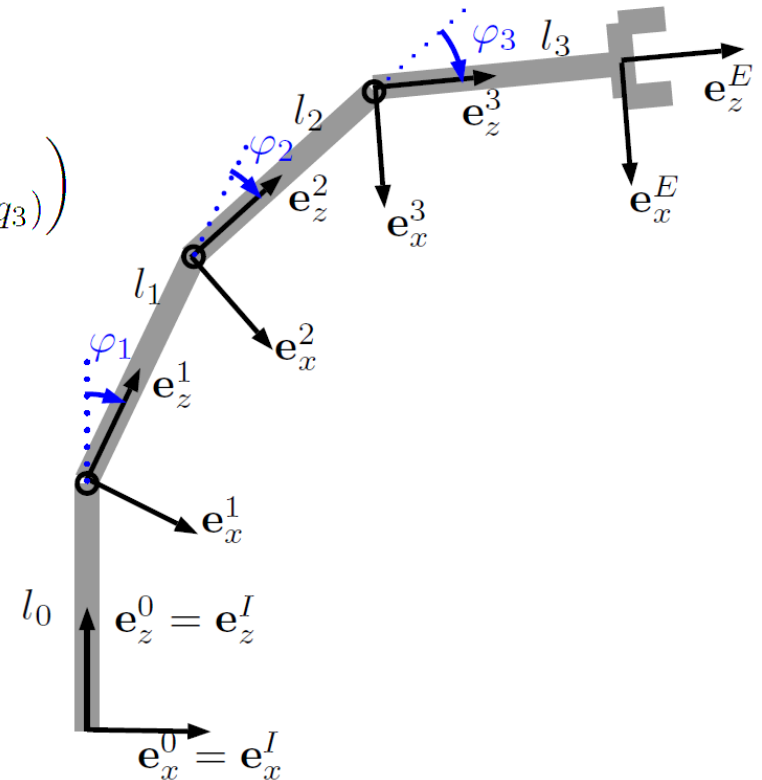## Planar robot arm

- Given (from last example)

$$\boldsymbol{\chi}_{eP}(\mathbf{q}) = \begin{pmatrix} x \\ z \end{pmatrix} = \begin{pmatrix} l_1 \sin(q_1) + l_2 \sin(q_1 + q_2) + l_3 \sin(q_1 + q_2 + q_3) \\ l_0 + l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) + l_3 \cos(q_1 + q_2 + q_3) \end{pmatrix}$$

$$\boldsymbol{\chi}_{eR}(\mathbf{q}) = \chi_{eR}(\mathbf{q}) = q_1 + q_2 + q_3$$

- Determine the analytical Jacobian

$$\mathbf{J}_{eAP}(\mathbf{q}) = \frac{\partial \boldsymbol{\chi}_{eP}}{\partial \mathbf{q}} = \begin{bmatrix} l_1 c_1 + l_2 c_{12} + l_3 c_{123} & l_2 c_{12} + l_3 c_{213} & l_3 c_{213} \\ -l_1 s_1 - l_2 s_{12} - l_3 s_{123} & -l_2 s_{12} - l_3 s_{213} & -l_3 s_{213} \end{bmatrix} \in \mathbb{R}^{2 \times 3}$$

$$\mathbf{J}_{eAR}(\mathbf{q}) = \frac{\partial \boldsymbol{\chi}_{eR}}{\partial \mathbf{q}} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \in \mathbb{R}^{1 \times 3}$$

Forward Differential Kinematics

Forward Differential Kinematics

- Analytic: $\delta \boldsymbol{\chi}_e \approx \dfrac{\partial \boldsymbol{\chi}_e(\mathbf{q})}{\partial \mathbf{q}} \delta \mathbf{q} = \mathbf{J}_{eA}(\mathbf{q})\, \delta \mathbf{q}$ with $\mathbf{J}_{eA} = \dfrac{\partial \boldsymbol{\chi}_e}{\partial \mathbf{q}} = \begin{bmatrix} \frac{\partial \chi_1}{\partial q_1} & \cdots & \frac{\partial \chi_1}{\partial q_{n_j}} \\ \vdots & \ddots & \vdots \\ \frac{\partial \chi_m}{\partial q_1} & \cdots & \frac{\partial \chi_m}{\partial q_{n_j}} \end{bmatrix}$

$$\dot{\boldsymbol{\chi}}_e = \mathbf{J}_{eA}(\mathbf{q})\, \dot{\mathbf{q}} \quad \text{with } \mathbf{J}_{eA}(\mathbf{q}) \in \mathbb{R}^{m_e \times n_j}$$

Depending on parameterization!!

$$\mathbf{w}_e = \mathbf{E}_e(\boldsymbol{\chi}_e)\, \dot{\boldsymbol{\chi}}_e$$

$$\downarrow$$

$$\mathbf{J}_{e0}(\mathbf{q}) = \mathbf{E}_e(\boldsymbol{\chi})\, \mathbf{J}_{eA}(\mathbf{q})$$

- Geometric: $\mathbf{w}_e = \begin{pmatrix} \mathbf{v}_e \\ \boldsymbol{\omega}_e \end{pmatrix} = \mathbf{J}_{e0}(\mathbf{q})\, \dot{\mathbf{q}}$ with $\mathbf{J}_{e0}(\mathbf{q}) \in \mathbb{R}^{6 \times n_j}$

Independent of parameterization

- Algebra: $\mathbf{w}_C = \begin{pmatrix} \mathbf{v}_C \\ \boldsymbol{\omega}_C \end{pmatrix} = \mathbf{w}_B + \mathbf{w}_{BC}$

$$\mathbf{J}_C \dot{\mathbf{q}} = \mathbf{J}_B \dot{\mathbf{q}} + \mathbf{J}_{BC} \dot{\mathbf{q}}$$

$$_{\mathcal{A}}\mathbf{J}_C = {}_{\mathcal{A}}\mathbf{J}_B + {}_{\mathcal{A}}\mathbf{J}_{BC}$$

Velocity in Moving Bodies

- Definitions

$\mathbf{v}_P$: the absolute velocity of P

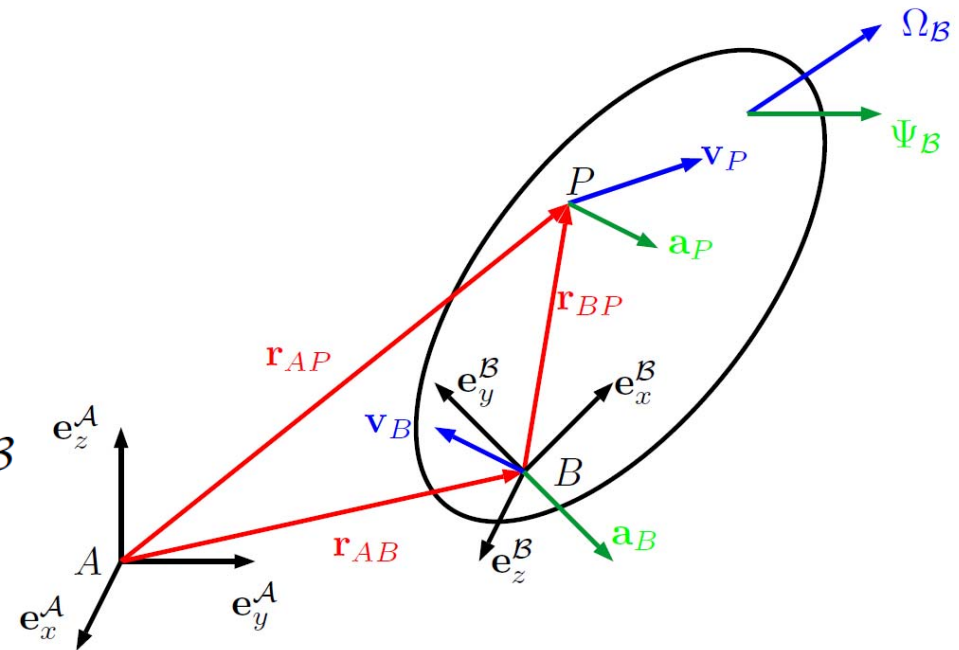$\mathbf{a}_P = \dot{\mathbf{v}}_P$: the absolute acceleration of P

$\Omega_{\mathcal{B}} = \omega_{\mathcal{AB}}$: (absolute) angular velocity of body $\mathcal{B}$

$\Psi_{\mathcal{B}} = \dot{\Omega}_{\mathcal{B}}$: (absolute) angular acceleration of body $\mathcal{B}$

- Remember the difference:

  - Velocity

  - Time derivative of coordinates:

Vector Differentiation in Moving Frame
Euler differentiation rule

Vector Differentiation in Moving Frame
Euler differentiation rule

- For non-moving reference frames:

$$_{\mathcal{A}}\mathbf{v}_P = {}_{\mathcal{A}}\dot{\mathbf{r}}_{AP}$$

- For moving reference frames:

$$\mathbf{v}_P \neq \dot{\mathbf{r}}_{AP}$$

- Vector differentiation in moving frames ($\mathcal{A}$ = inertial/reference frame):

$$
\begin{aligned}
_{\mathcal{B}}\mathbf{v}_P &= \mathbf{C}_{\mathcal{BA}} \cdot \frac{d}{dt}\left(\mathbf{C}_{\mathcal{AB}} \cdot {}_{\mathcal{B}}\mathbf{r}_{AP}\right) \\
&= \mathbf{C}_{\mathcal{BA}} \cdot \left(\mathbf{C}_{\mathcal{AB}} \cdot {}_{\mathcal{B}}\dot{\mathbf{r}}_{AP} + \dot{\mathbf{C}}_{\mathcal{AB}} \cdot {}_{\mathcal{B}}\mathbf{r}_{AP}\right) \\
&= \mathbf{C}_{\mathcal{BA}} \cdot \left(\mathbf{C}_{\mathcal{AB}} \cdot {}_{\mathcal{B}}\dot{\mathbf{r}}_{AP} + \left[{}_{\mathcal{A}}\omega_{\mathcal{AB}}\right]_\times \cdot \mathbf{C}_{\mathcal{AB}} \cdot {}_{\mathcal{B}}\mathbf{r}_{AP}\right) \\
&= {}_{\mathcal{B}}\dot{\mathbf{r}}_{AP} + \mathbf{C}_{\mathcal{BA}} \cdot \left[{}_{\mathcal{A}}\omega_{\mathcal{AB}}\right]_\times \cdot \mathbf{C}_{\mathcal{AB}} \cdot {}_{\mathcal{B}}\mathbf{r}_{AP} \\
&= {}_{\mathcal{B}}\dot{\mathbf{r}}_{AP} + {}_{\mathcal{B}}\omega_{\mathcal{AB}} \times {}_{\mathcal{B}}\mathbf{r}_{AP}
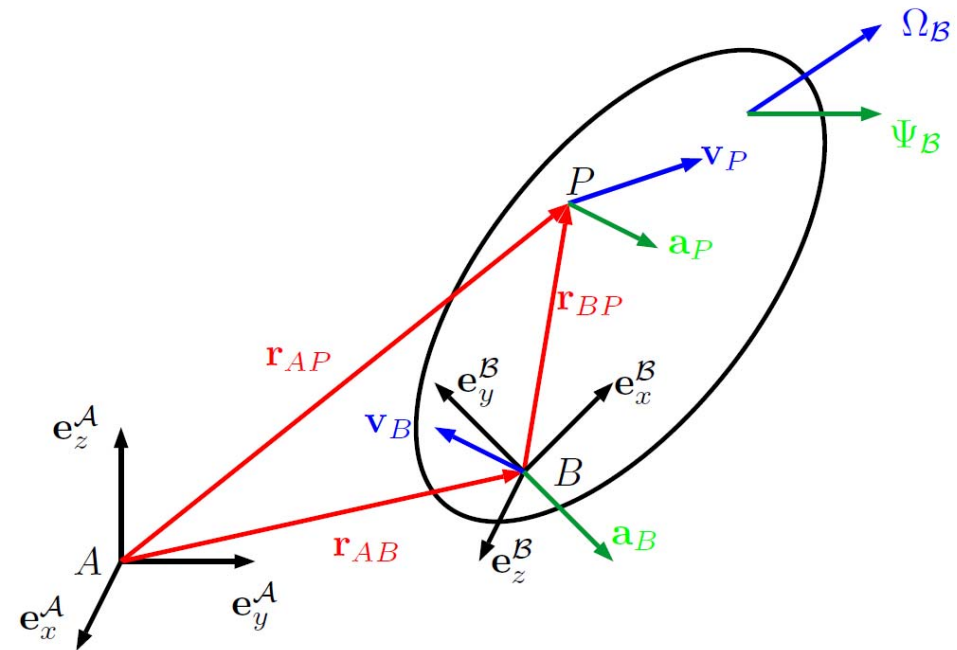\end{aligned}
$$

$$\left[{}_{\mathcal{A}}\omega_{\mathcal{AB}}\right]_\times = \dot{\mathbf{C}}_{\mathcal{AB}}\mathbf{C}_{\mathcal{AB}}^T$$

$$\left[{}_{\mathcal{B}}\omega_{\mathcal{AB}}\right]_\times = \mathbf{C}_{\mathcal{BA}} \cdot \left[{}_{\mathcal{A}}\omega_{\mathcal{AB}}\right]_\times \cdot \mathbf{C}_{\mathcal{AB}}$$

Velocity in Moving Bodies
Rigid body formulation

- Apply transformation rule as learned before

- Differentiate with respect to time

- Substitute

- Rigid body formulation

Velocity in Moving Bodies
Rigid body formulation

- Apply transformation rule as learned before

$$_\mathcal{A}\mathbf{r}_{AP} = {}_\mathcal{A}\mathbf{r}_{AB} + {}_\mathcal{A}\mathbf{r}_{BP} = {}_\mathcal{A}\mathbf{r}_{AB} + \mathbf{C}_{\mathcal{AB}} \cdot {}_\mathcal{B}\mathbf{r}_{BP}$$

- Differentiate with respect to time

$$_\mathcal{A}\dot{\mathbf{r}}_{AP} = {}_\mathcal{A}\dot{\mathbf{r}}_{AB} + \mathbf{C}_{\mathcal{AB}} \cdot {}_\mathcal{B}\dot{\mathbf{r}}_{BP} + \dot{\mathbf{C}}_{\mathcal{AB}} \cdot {}_\mathcal{B}\mathbf{r}_{BP}$$

- Substitute $\dot{\mathbf{C}}_{\mathcal{AB}} = [{}_\mathcal{A}\boldsymbol{\omega}_{\mathcal{AB}}]_\times \cdot \mathbf{C}_{\mathcal{AB}}$

- Rigid body formulation

$$\boxed{\begin{aligned} _\mathcal{A}\dot{\mathbf{r}}_{AP} &= {}_\mathcal{A}\dot{\mathbf{r}}_{AB} + [{}_\mathcal{A}\boldsymbol{\omega}_{\mathcal{AB}}]_\times \cdot \mathbf{C}_{\mathcal{AB}} \cdot {}_\mathcal{B}\mathbf{r}_{BP} \\ &= {}_\mathcal{A}\dot{\mathbf{r}}_{AB} + {}_\mathcal{A}\boldsymbol{\omega}_{\mathcal{AB}} \times {}_\mathcal{A}\mathbf{r}_{BP} \end{aligned}}$$

$$\mathbf{v}_P = \mathbf{v}_B + \boldsymbol{\Omega} \times \mathbf{r}_{BP}$$
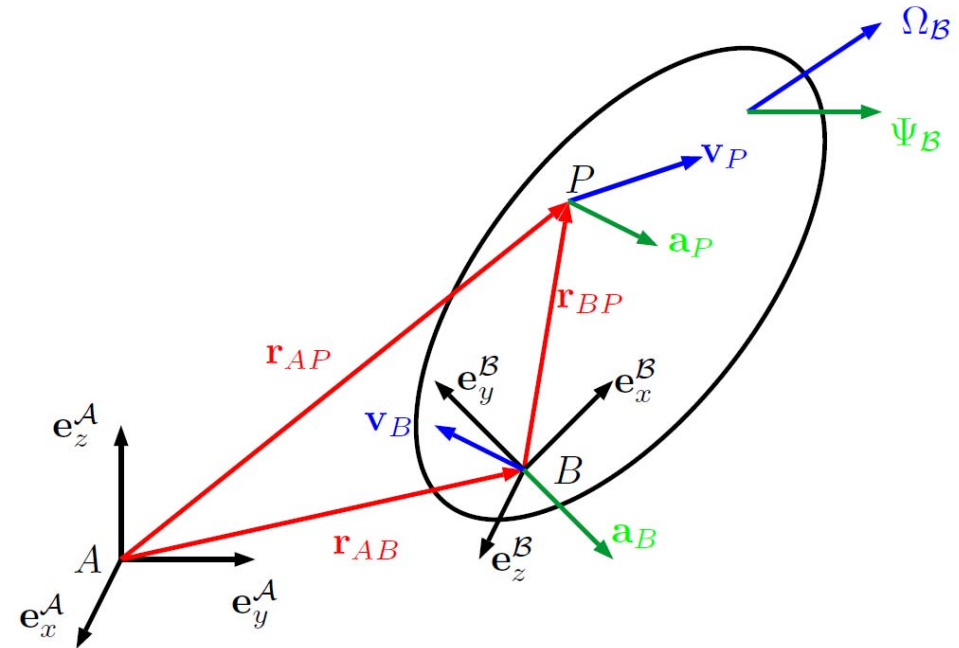
Geometric Jacobian Derivation

- Rigid body formulation at a single element

- Apply this to all bodies

- Angular velocity propagation

- …get the end-effector velocity
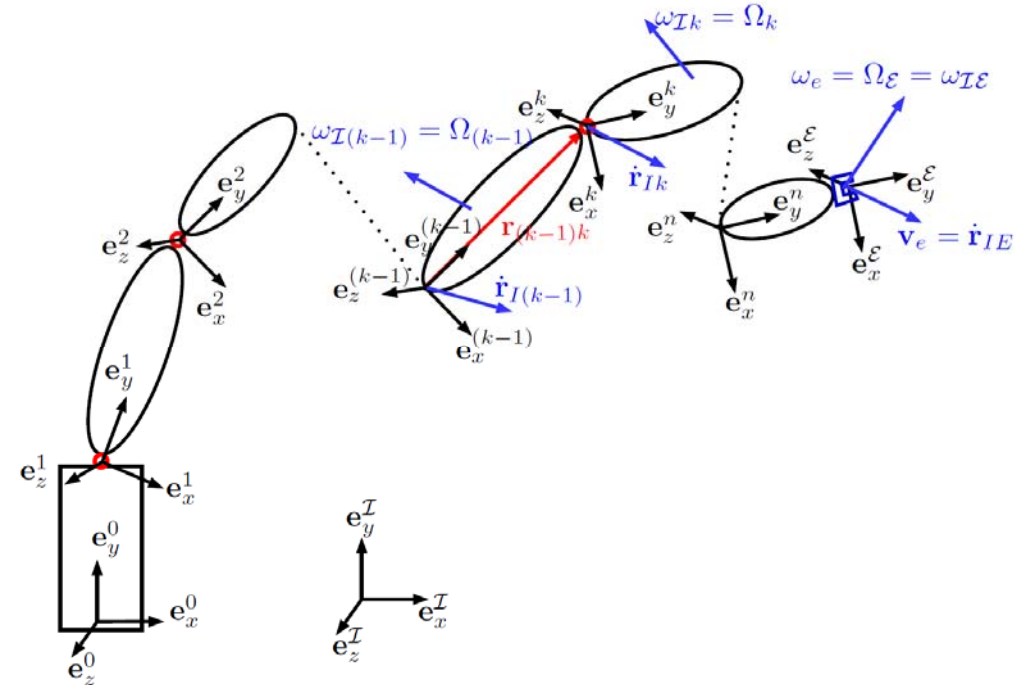
Geometric Jacobian Derivation

- Rigid body formulation at a single element

$$\dot{\mathbf{r}}_{Ik} = \dot{\mathbf{r}}_{I(k-1)} + \boldsymbol{\omega}_{\mathcal{I}(k-1)} \times \mathbf{r}_{(k-1)k}$$
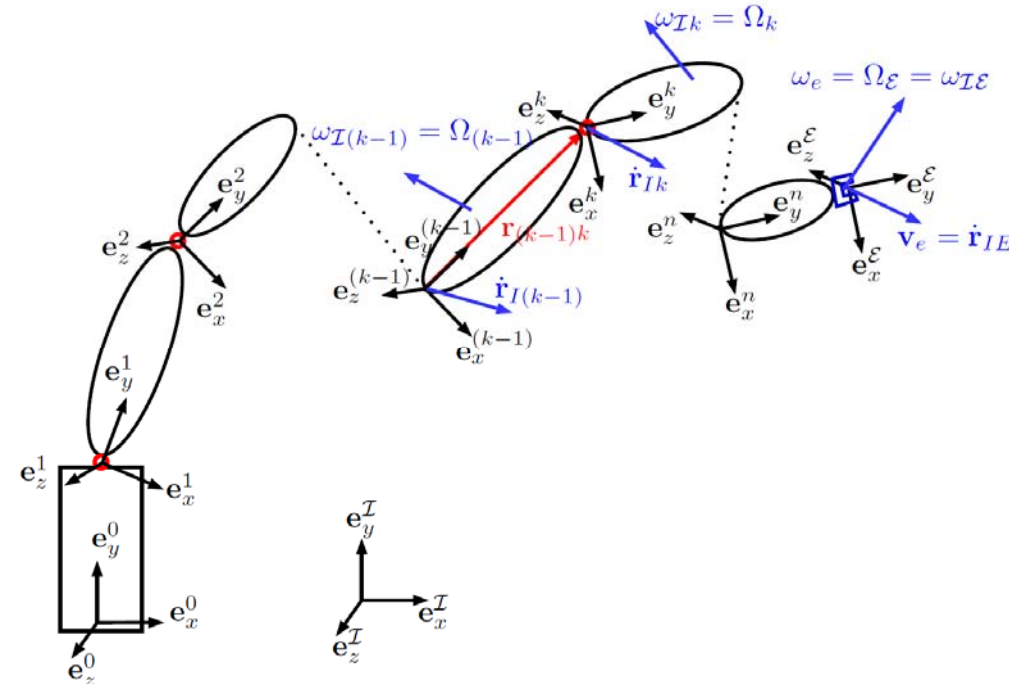
- Apply this to all bodies

$$\dot{\mathbf{r}}_{IE} = \sum_{k=1}^{n} \boldsymbol{\omega}_{\mathcal{I},k} \times \mathbf{r}_{k(k+1)}$$

- Angular velocity propagation

$$\left.\begin{array}{l} \boldsymbol{\omega}_{\mathcal{I}(k)} = \boldsymbol{\omega}_{\mathcal{I}(k-1)} + \boldsymbol{\omega}_{(k-1)k} \\[2mm] \text{with} \quad \boldsymbol{\omega}_{(k-1)k} = \mathbf{n}_k \dot{q}_k \end{array}\right\} \quad \boldsymbol{\omega}_{\mathcal{I}k} = \sum_{i=1}^{k} \mathbf{n}_i \dot{q}_i$$

- …get the end-effector velocity

$$\dot{\mathbf{r}}_{IE} = \sum_{k=1}^{n} \left\{ \sum_{i=1}^{k} (\mathbf{n}_i \dot{q}_i) \times \mathbf{r}_{k(k+1)} \right\} = \sum_{k=1}^{n} \mathbf{n}_k \dot{q}_k \times \mathbf{r}_{k(n+1)}$$
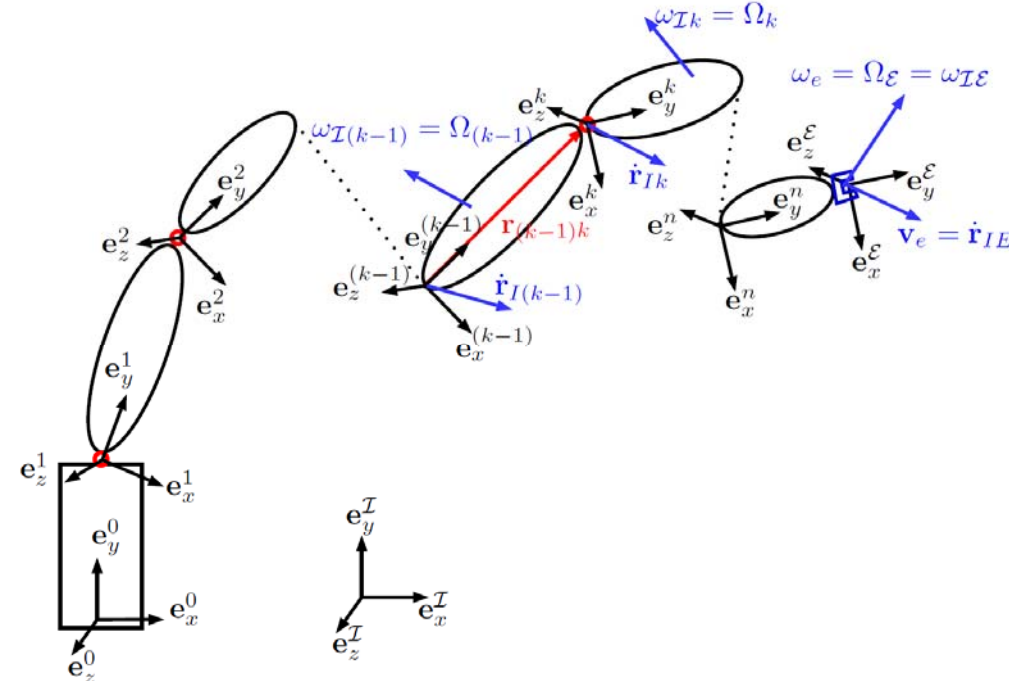
Geometric Jacobian Derivation

- Position Jacobian $\quad \dot{\mathbf{r}}_{IE} = \sum_{k=1}^{n} \mathbf{n}_k \dot{q}_k \times \mathbf{r}_{k(n+1)}$

$$\dot{\mathbf{r}}_{IE} = \underbrace{\begin{bmatrix} \mathbf{n}_1 \times \mathbf{r}_{1(n+1)} & \mathbf{n}_2 \times \mathbf{r}_{2(n+1)} & \cdots & \mathbf{n}_n \times \mathbf{r}_{n(n+1)} \end{bmatrix}}_{\mathbf{J}_{e0_P}} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{pmatrix}$$

- Rotation Jacobian from $\quad \omega_{\mathcal{I}k} = \sum_{i=1}^{k} \mathbf{n}_i \dot{q}_i$

$$\omega_{\mathcal{I}\mathcal{E}} = \sum_{i=1}^{n} \mathbf{n}_i \dot{q}_i = \underbrace{\begin{bmatrix} \mathbf{n}_1 & \mathbf{n}_2 & \cdots & \mathbf{n}_n \end{bmatrix}}_{\mathbf{J}_{e0_R}} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{pmatrix}$$

$$\Rightarrow \quad {}_{\mathcal{I}}\mathbf{J}_{e0} = \begin{bmatrix} {}_{\mathcal{I}}\mathbf{J}_{e0_P} \\ {}_{\mathcal{I}}\mathbf{J}_{e0_R} \end{bmatrix} = \begin{bmatrix} {}_{\mathcal{I}}\mathbf{n}_1 \times {}_{\mathcal{I}}\mathbf{r}_{1(n+1)} & {}_{\mathcal{I}}\mathbf{n}_2 \times {}_{\mathcal{I}}\mathbf{r}_{2(n+1)} & \cdots & {}_{\mathcal{I}}\mathbf{n}_n \times {}_{\mathcal{I}}\mathbf{r}_{n(n+1)} \\ {}_{\mathcal{I}}\mathbf{n}_1 & {}_{\mathcal{I}}\mathbf{n}_2 & \cdots & {}_{\mathcal{I}}\mathbf{n}_n \end{bmatrix}$$

$$\boxed{{}_{\mathcal{I}}\mathbf{n}_k = \mathbf{C}_{\mathcal{I}(k-1)} \cdot {}_{(k-1)}\mathbf{n}_k}$$

Geometric Jacobian
Planar robot arm

- Preparation: determine the rotation matrices

$$\mathbf{C}_{\mathcal{I}1} = \begin{bmatrix} c_1 & 0 & s_1 \\ 0 & 1 & 0 \\ -s_1 & 0 & c_1 \end{bmatrix} \qquad \mathbf{C}_{\mathcal{I}2} = \mathbf{C}_{\mathcal{I}1} \cdot \begin{bmatrix} c_2 & 0 & s_2 \\ 0 & 1 & 0 \\ -s_2 & 0 & c_2 \end{bmatrix} = \begin{bmatrix} c_{12} & 0 & s_{12} \\ 0 & 1 & 0 \\ -s_{12} & 0 & c_{12} \end{bmatrix} \qquad \mathbf{C}_{\mathcal{I}3} = \dots$$

- Determine the rotation axes
  - Locally $\quad _0\mathbf{n}_1 = {}_1\mathbf{n}_2 = {}_2\mathbf{n}_3 = \mathbf{e}_y$ $\qquad$ Inertial frame

$$\begin{aligned} _\mathcal{I}\mathbf{n}_1 &= {}_0\mathbf{n}_1 = \mathbf{e}_y \\ _\mathcal{I}\mathbf{n}_2 &= \mathbf{C}_{I1} \cdot {}_1\mathbf{n}_2 = \mathbf{e}_y \\ _\mathcal{I}\mathbf{n}_3 &= \mathbf{C}_{I2} \cdot {}_2\mathbf{n}_3 = \mathbf{e}_y \end{aligned}$$

- Determine the position vectors

$$_\mathcal{I}\mathbf{r}_{1E} = {}_\mathcal{I}\mathbf{r}_{12} + {}_\mathcal{I}\mathbf{r}_{23} + {}_\mathcal{I}\mathbf{r}_{3E} = \mathbf{C}_{I1} \cdot {}_1\mathbf{r}_{12} + \mathbf{C}_{I2} \cdot {}_2\mathbf{r}_{23} + \mathbf{C}_{I3} \cdot {}_3\mathbf{r}_{3E} = l_1 \begin{pmatrix} s_{q_1} \\ 0 \\ c_{q_1} \end{pmatrix} + l_2 \begin{pmatrix} s_{12} \\ 0 \\ c_{12} \end{pmatrix} + l_3 \begin{pmatrix} s_{123} \\ 0 \\ c_{123} \end{pmatrix}$$

$$_\mathcal{I}\mathbf{r}_{2E} = {}_\mathcal{I}\mathbf{r}_{23} + {}_\mathcal{I}\mathbf{r}_{3E} = \dots \qquad _\mathcal{I}\mathbf{r}_{3E} = \dots$$

- Get the Jacobian

$$_\mathcal{I}\mathbf{J}_{e0_P} = \begin{bmatrix} _\mathcal{I}\mathbf{n}_1 \times {}_\mathcal{I}\mathbf{r}_{1E} & _\mathcal{I}\mathbf{n}_2 \times {}_\mathcal{I}\mathbf{r}_{2E} & _\mathcal{I}\mathbf{n}_3 \times {}_\mathcal{I}\mathbf{r}_{3E} \end{bmatrix}$$

$$= \begin{bmatrix} l_1 c_1 + l_2 c_{12} + l_3 c_{123} & l_2 c_{12} + l_3 c_{123} & l_3 c_{123} \\ 0 & 0 & 0 \\ -l_1 s_1 - l_2 s_{12} - l_3 c_{123} & -l_2 s_{12} - l_3 s_{123} & -l_3 s_{123} \end{bmatrix}$$

$$_\mathcal{I}\mathbf{J}_{e0_R} = \begin{bmatrix} _\mathcal{I}\mathbf{n}_1 & _\mathcal{I}\mathbf{n}_2 & _\mathcal{I}\mathbf{n}_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$
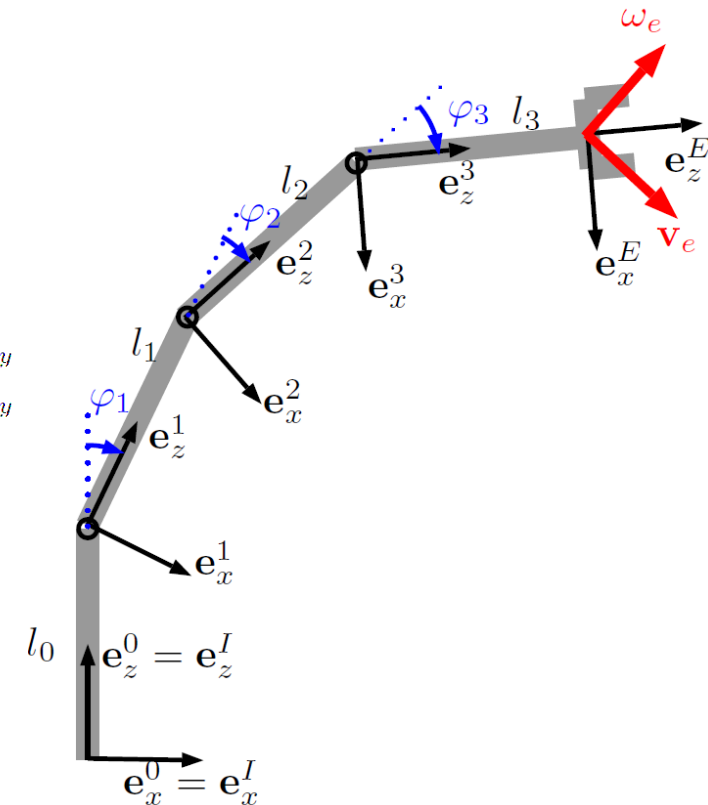
Recapitulation
Analytical and Kinematic Jacobian

- Analytical Jacobian

$$\dot{\chi}_e = \mathbf{J}_{eA}(\mathbf{q})\,\dot{\mathbf{q}}$$

- Geometric (or basic) Jacobian

$$\mathbf{w}_e = \begin{pmatrix} \mathbf{v}_e \\ \omega_e \end{pmatrix} = \mathbf{J}_{e0}(\mathbf{q})\,\dot{\mathbf{q}}$$

$$\mathbf{J}_{e0}(\mathbf{q}) = \mathbf{E}_e(\chi)\,\mathbf{J}_{eA}(\mathbf{q})$$

- Relates time-derivatives of config. parameters to generalized velocities
- Depending on selected parameterization (mainly rotation) in 3D $\quad \Delta\chi \Leftrightarrow \Delta\mathbf{q}$
  Note: there exist no "rotation angle"
- Mainly used for numeric algorithms

- Relates end-effector velocity to generalized velocities
- Unique for every robot

- Used in most cases

Importance of Jacobian

- Kinematics (mapping of changes from joint to task space)
    - Inverse kinematics control
    - Resolve redundancy problems
    - Express contact constraints
- Statics (and later also dynamics)
    - Principle of virtual work
        - Variations in work must cancel for all virtual displacement
        - Internal forces of ideal joint don't contribute

$$\delta W = \sum_i \mathbf{f}_i \mathbf{x}_i = \boldsymbol{\tau}^T \delta \mathbf{q} + \left(-\mathbf{F}_E\right)^T \delta \mathbf{x}_E$$

$$= \boldsymbol{\tau}^T \delta \mathbf{q} + \left(-\mathbf{F}_E\right)^T \mathbf{J} \delta \mathbf{q} = 0 \quad \forall \delta \mathbf{q}$$

- ➢ Dual problem from principle of virtual work

$$\boxed{\begin{array}{l} \dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}} \\ \boldsymbol{\tau} = \mathbf{J}^T \mathbf{F} \end{array}}$$
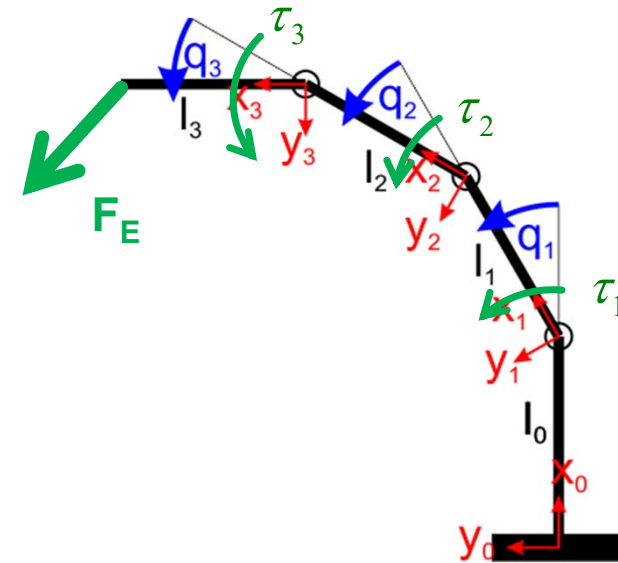
Floating Base Kinematics

151-0851-00 V

lecture:	CAB G11	Tuesday 10:15 – 12:00, every week
exercise:	HG E1.2	Wednesday 8:15 – 10:00, according to schedule (about every 2nd week)
office hour:	LEE H303	Friday 12.15 – 13.00

Marco Hutter, Roland Siegwart, and Thomas Stastny
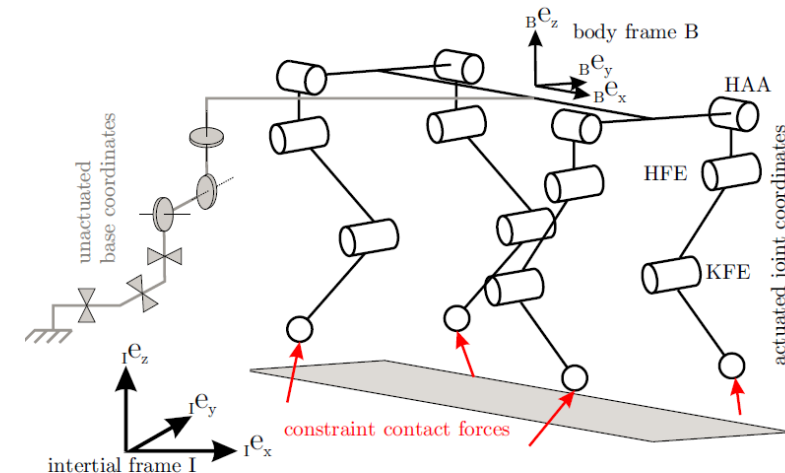
Floating Base Systems
Kinematics

- Generalized coordinates

$$\mathbf{q} = \begin{pmatrix} \mathbf{q}_b \\ \mathbf{q}_j \end{pmatrix} \quad \text{with} \quad \mathbf{q}_b = \begin{pmatrix} \mathbf{q}_{b_P} \\ \mathbf{q}_{b_R} \end{pmatrix} \in \mathbb{R}^3 \times SO(3)$$
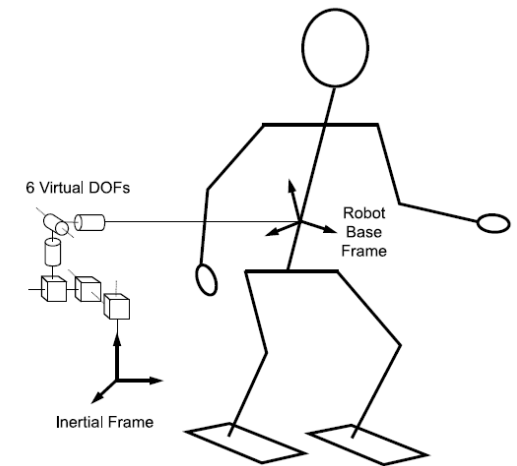
- Generalized velocities and accelerations?
  - Time derivatives $\dot{\mathbf{q}}, \ddot{\mathbf{q}}$ depend on parameterization

  - Often $\quad \mathbf{u} = \begin{pmatrix} _I\mathbf{v}_B \\ _B\boldsymbol{\omega}_{IB} \\ \dot{\varphi}_1 \\ \vdots \\ \dot{\varphi}_{n_j} \end{pmatrix} \in \mathbb{R}^{6+n_j} = \mathbb{R}^{n_u} \qquad \dot{\mathbf{u}} = \begin{pmatrix} _I\mathbf{a}_B \\ _B\boldsymbol{\psi}_{IB} \\ \ddot{\varphi}_1 \\ \vdots \\ \ddot{\varphi}_{n_j} \end{pmatrix} \in \mathbb{R}^{6+n_j}$

  - Linear mapping $\quad \mathbf{u} = \mathbf{E}_{fb} \cdot \dot{\mathbf{q}}, \text{with} \quad \mathbf{E}_{fb} = \begin{bmatrix} \mathbb{I}_{3\times3} & 0 & 0 \\ 0 & \mathbf{E}_{\boldsymbol{\chi}_R} & 0 \\ 0 & 0 & \mathbb{I}_{n_j \times n_j} \end{bmatrix}$

(a) Quadruped

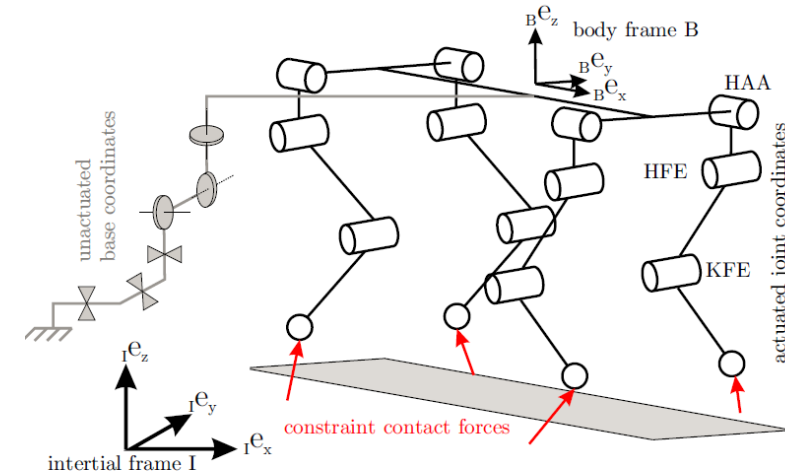(b) Humanoid

Floating Base Systems
Differential kinematics

- Position of an arbitrary point on the robot

$$\mathcal{I}\mathbf{r}_{IQ}(\mathbf{q}) = \boxed{\mathcal{I}\mathbf{r}_{IB}(\mathbf{q})} + \boxed{\mathbf{C}_{\mathcal{IB}}(\mathbf{q})} \cdot \boxed{\mathcal{B}\mathbf{r}_{BQ}(\mathbf{q})}$$

$$\mathcal{I}\mathbf{r}_{IB}(\mathbf{q}_b) \quad \mathbf{C}_{\mathcal{IB}}(\mathbf{q}_b) \quad \mathcal{B}\mathbf{r}_{BQ}(\mathbf{q}_j)$$
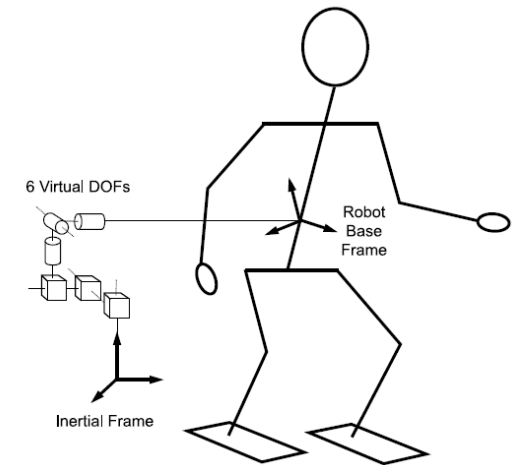

(a) Quadruped

- Velocity of this point

$$\mathcal{I}\mathbf{v}_Q = \mathcal{I}\mathbf{v}_B + \dot{\mathbf{C}}_{\mathcal{IB}} \cdot \mathcal{B}\mathbf{r}_{BQ} + \mathbf{C}_{\mathcal{IB}} \cdot \mathcal{B}\dot{\mathbf{r}}_{BQ}$$

$$= \mathcal{I}\mathbf{v}_B + \mathbf{C}_{\mathcal{IB}} \cdot [\mathcal{B}\boldsymbol{\omega}_{\mathcal{IB}}]_\times \cdot \mathcal{B}\mathbf{r}_{BQ} + \mathbf{C}_{\mathcal{IB}} \cdot \mathcal{B}\dot{\mathbf{r}}_{BQ}$$

$$= \mathcal{I}\mathbf{v}_B - \mathbf{C}_{\mathcal{IB}} \cdot [\mathcal{B}\mathbf{r}_{BQ}]_\times \cdot \mathcal{B}\boldsymbol{\omega}_{\mathcal{IB}} + \mathbf{C}_{\mathcal{IB}} \cdot \mathcal{B}\dot{\mathbf{r}}_{BQ}$$

$$= \mathcal{I}\mathbf{v}_B - \mathbf{C}_{\mathcal{IB}} \cdot [\mathcal{B}\mathbf{r}_{BQ}]_\times \cdot \mathcal{B}\boldsymbol{\omega}_{\mathcal{IB}} + \mathbf{C}_{\mathcal{IB}} \cdot \mathcal{B}\mathbf{J}_{P_{q_j}}(\mathbf{q}_j) \cdot \dot{\mathbf{q}}_j$$

$$= \boxed{\left[\mathbb{I}_{3\times3} \quad -\mathbf{C}_{\mathcal{IB}} \cdot [\mathcal{B}\mathbf{r}_{BQ}]_\times \quad \mathbf{C}_{\mathcal{IB}} \cdot \mathcal{B}\mathbf{J}_{P_{q_j}}(\mathbf{q}_j) \right]} \cdot \mathbf{u} \quad \text{with} \quad \mathbf{u} = \begin{pmatrix} \mathcal{I}\mathbf{v}_B \\ \mathcal{B}\boldsymbol{\omega}_{IB} \\ \dot{\varphi}_1 \\ \vdots \\ \dot{\varphi}_{n_j} \end{pmatrix}$$

$$\mathcal{I}\mathbf{J}_Q(\mathbf{q})$$

$$[\mathcal{B}\boldsymbol{\omega}_{\mathcal{IB}}]_\times = \mathbf{C}_{\mathcal{BI}}[\mathcal{I}\boldsymbol{\omega}_{\mathcal{IB}}]_\times \mathbf{C}_{\mathcal{BI}}^T$$

$$= \mathbf{C}_{\mathcal{IB}}^T \dot{\mathbf{C}}_{\mathcal{IB}} \mathbf{C}_{\mathcal{IB}}^T \mathbf{C}_{\mathcal{IB}} = \mathbf{C}_{\mathcal{IB}}^T \dot{\mathbf{C}}_{\mathcal{IB}}$$

(b) Humanoid

Contact Constraints

- A contact point C_i is not allowed to move:

- Constraint as a function of generalized coordinates:

- Stack of constraints

Contact Constraints

- A contact point C_i is not allowed to move:

$$_\mathcal{I}\mathbf{r}_{IC_i} = const, \quad _\mathcal{I}\dot{\mathbf{r}}_{IC_i} = _\mathcal{I}\ddot{\mathbf{r}}_{IC_i} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

- Constraint as a function of generalized coordinates:

$$_\mathcal{I}\mathbf{J}_{C_i}\mathbf{u} = \mathbf{0}, \qquad _\mathcal{I}\mathbf{J}_{C_i}\dot{\mathbf{u}} + _\mathcal{I}\dot{\mathbf{J}}_{C_i}\mathbf{u} = \mathbf{0}$$
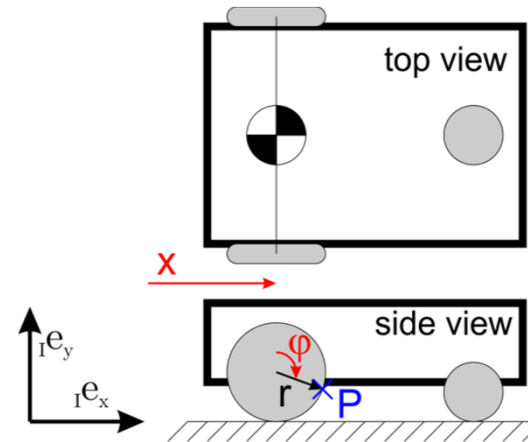
- Stack of constraints

$$\mathbf{J}_c = \begin{bmatrix} \mathbf{J}_{C_1} \\ \vdots \\ \mathbf{J}_{C_{n_c}} \end{bmatrix} \in \mathbb{R}^{3n_c \times n_n}$$

Contact Constraint
Wheeled vehicle simple example

- Contact constraints

  - Point on wheel

  - Jacobian

  - Contact constraints

Contact Constraint
Wheeled vehicle simple example

- Contact constraints

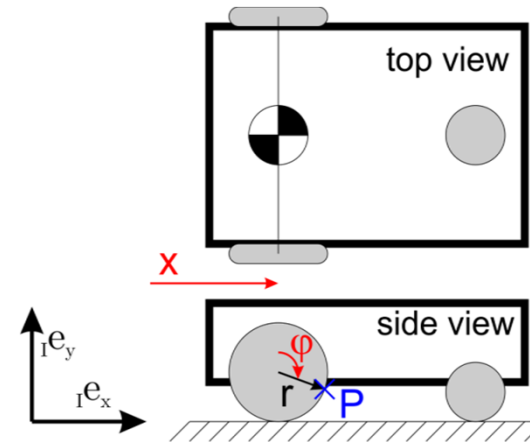 - Point on wheel
  $$_{\mathcal{I}}\mathbf{r}_{IP} = \begin{pmatrix} x + r\sin(\varphi) \\ r + r\cos(\varphi) \\ 0 \end{pmatrix}$$

  - Jacobian
  $$_{\mathcal{I}}\mathbf{J}_P = \begin{bmatrix} 1 & r\cos(\varphi) \\ 0 & -r\sin(\varphi) \\ 0 & 0 \end{bmatrix}$$

  - Contact constraints
  $$_{\mathcal{I}}\dot{\mathbf{r}}_{IP}\big|_{\varphi=\pi} = {}_{\mathcal{I}}\mathbf{J}_P\big|_{\varphi=\pi}\dot{\mathbf{q}} = \begin{bmatrix} 1 & -r \\ 0 & 0 \\ 0 & 0 \end{bmatrix}\begin{pmatrix} \dot{x} \\ \dot{\varphi} \end{pmatrix} = \mathbf{0}$$

=> Rolling condition $\quad \dot{x} - r\dot{\varphi} = 0$

top view

side view

$_{\mathcal{I}}e_y$

$_{\mathcal{I}}e_x$

$$\mathbf{q} = \begin{pmatrix} x \\ \varphi \end{pmatrix}$$

Un-actuated base

Actuated joints

Properties of Contact Jacobian

- Contact Jacobian tells us, how a system can move.
  - Separate stacked Jacobian $\mathbf{J}_c = \begin{bmatrix} \boxed{\mathbf{J}_{c,b}} & \mathbf{J}_{c,j} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{r}_c}{\partial \mathbf{q}_b} & \frac{\partial \mathbf{r}_c}{\partial \mathbf{q}_j} \end{bmatrix} \in \mathbb{R}^{n_c \times (n_b + n_j)}$

    relation between base motion and constraints

  - Base is fully controllable if $rank(\mathbf{J}_{c,b}) = 6$

  - Nr of kinematic constraints for joint actuators: $rank(\mathbf{J}_c) - rank(\mathbf{J}_{c,b})$

- Generalized coordinates DON'T correspond to the degrees of freedom
  - Contact constraints!
- Minimal coordinates (= correspond to degrees of freedom)
  - Require to switch the set of coordinates depending on contact state (=> never used)

Quadrupedal Robot with Point Feet

- Floating base system with 12 actuated joint and 6 base coordinates (18DoF)
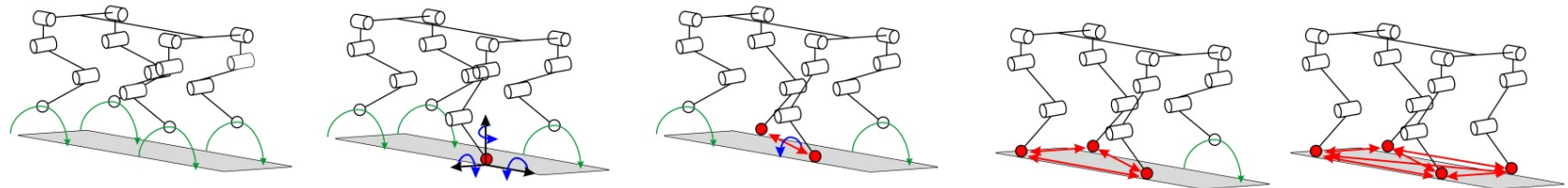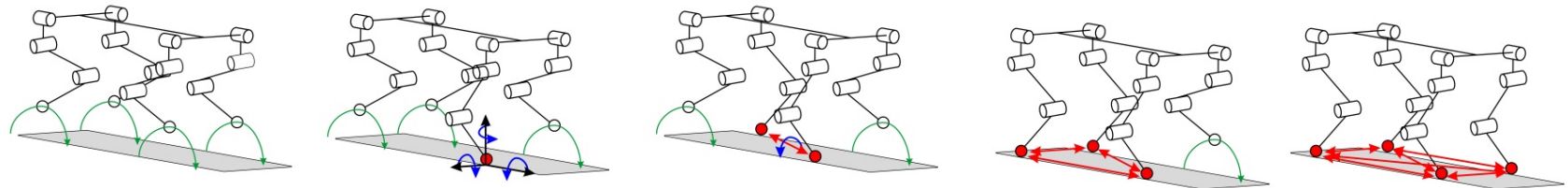

| Total constraints |
| Internal constraints |
| Uncontrollable DoFs |

Quadrupedal Robot with Point Feet

- Floating base system with 12 actuated joint and 6 base coordinates (18DoF)

Total constraints	0	3	6	9	12
Internal constraints	0	0	1	3	6
Uncontrollable DoFs	6	3	1	0	0

Outlook

- ## Exercise TOMORROW
  - Differential Kinematics
  - Use it as extended office hour!

- ## Next Lecture
  - Script Section 2.9 (Kinematic Control Methods)
  - Inverse Kinematics
  - Inverse Differential Kinematics

{E}

Configuration end-effector

Inverse Kinematics

Joint angles

{I}