ETH zürich

Lecture «Robot Dynamics»: Summary

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

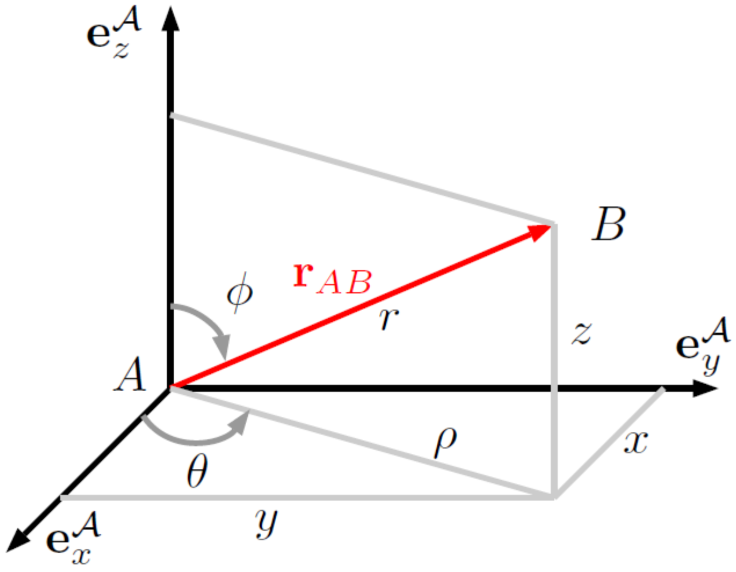
	Topic		Title
20.09.2016	Intro and Outline	L1	Course Introduction; Recapitulation Position, Linear Velocity, Transformation
27.09.2016	Kinematics 1	L2	Rotation Representation; Introduction to Multi-body Kinematics
28.09.2016	Exercise 1a	E1a	Kinematics Modeling the ABB arm
04.10.2016	Kinematics 2	L3	Kinematics of Systems of Bodies; Jacobians
05.10.2016	Exercise 1b	L3	Differential Kinematics and Jacobians of the ABB Arm
11.10.2016	Kinematics 3	L4	Kinematic Control Methods: Inverse Differential Kinematics, Inverse Kinematics; Rotation Error; Multi-task Control
12.10.2016	Exercise 1c	E1b	Kinematic Control of the ABB Arm
18.10.2016	Dynamics L1	L5	Multi-body Dynamics
19.10.2016	Exercise 2a	E2a	Dynamic Modeling of the ABB Arm
25.10.2016	Dynamics L2	L6	Dynamic Model Based Control Methods
26.10.2016	Exercise 2b	E2b	Dynamic Control Methods Applied to the ABB arm
01.11.2016	Legged Robots	L7	Case Study and Application of Control Methods
08.11.2016	Rotorcraft 1	L8	Dynamic Modeling of Rotorcraft I
15.11.2016	Rotorcraft 2	L9	Dynamic Modeling of Rotorcraft II & Control
16.11.2016	Exercise 3	E3	Modeling and Control of Multicopter
22.11.2016	Case Studies 2	L10	Rotor Craft Case Study
29.11.2016	Fixed-wing 1	L11	Flight Dynamics; Basics of Aerodynamics; Modeling of Fixed-wing Aircraft
30.11.2016	Exercise 4	E4	Aircraft Aerodynamics / Flight performance / Model derivation
06.12.2016	Fixed-wing 2	L12	Stability, Control and Derivation of a Dynamic Model
07.12.2016	Exercise 5	E5	Fixed-wing Control and Simulation
13.12.2016	Case Studies 3	L13	Fixed-wing Case Study
20.12.2016	Summery and Outlook	L14	Summery; Wrap-up; Exam

Position

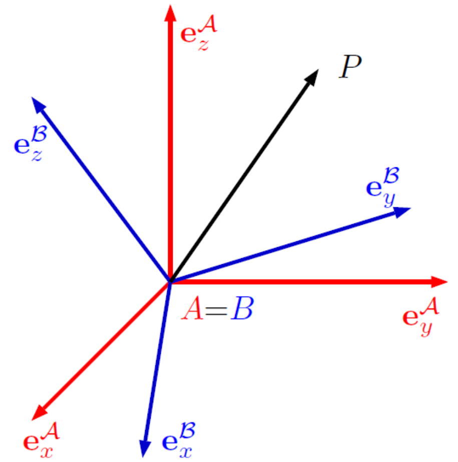
- Position ${}_{\mathcal{A}}\mathbf{r}_{AB} \in \mathbb{R}^3$, reference frames $\mathcal{A}$

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

- Different parameterizations, e.g.
  - Cartesian coordinates $\boldsymbol{\chi}_{Pc} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$
    - Position vector ${}_{\mathcal{A}}\mathbf{r} = x\mathbf{e}_x^{\mathcal{A}} + y\mathbf{e}_y^{\mathcal{A}} + z\mathbf{e}_z^{\mathcal{A}} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$
  - Cylindrical coordinates $\boldsymbol{\chi}_{Pz} = \begin{pmatrix} \rho \\ \theta \\ z \end{pmatrix}$
    - Position vector ${}_{\mathcal{A}}\mathbf{r} = \begin{pmatrix} \rho\cos\theta \\ \rho\sin\theta \\ z \end{pmatrix}$



Rotation



- Rotation $\phi_{\mathcal{AB}} \in SO(3)$
- Rotation matrix ${}_{\mathcal{A}}\mathbf{r}_{AP} = \left[{}_{\mathcal{A}}\mathbf{e}_x^{\mathcal{B}} \quad {}_{\mathcal{A}}\mathbf{e}_y^{\mathcal{B}} \quad {}_{\mathcal{A}}\mathbf{e}_z^{\mathcal{B}}\right] \cdot {}_{\mathcal{B}}\mathbf{r}_{AP} = \mathbf{C}_{\mathcal{AB}} \cdot {}_{\mathcal{B}}\mathbf{r}_{AP}$
- Different parameterizations, e.g.
  - Euler angles $\chi_{R,eulerZYZ} = \begin{pmatrix} z_1 \\ y \\ z_2 \end{pmatrix}$
  - Quaternions $\chi_{R,quat} = \boldsymbol{\xi} = \begin{pmatrix} \xi_0 \\ \check{\boldsymbol{\xi}} \end{pmatrix} \in \mathbb{H}$
  - …

Introduction to algebra with quaternions, e.g.

$$\boldsymbol{\xi}_{\mathcal{AB}} \otimes \boldsymbol{\xi}_{\mathcal{BC}} \longleftrightarrow \mathbf{C}_{\mathcal{AB}}\mathbf{C}_{\mathcal{BC}}$$

- Relation to rotation matrix:

$$\mathbf{C}_{\mathcal{AB}} = \mathbb{I}_{3\times3} + 2\xi_0\left[\check{\boldsymbol{\xi}}\right]_\times + 2\left[\check{\boldsymbol{\xi}}\right]_\times^2 = \left(2\xi_0^2 - 1\right)\mathbb{I}_{3\times3} + 2\xi_0\left[\check{\boldsymbol{\xi}}\right]_\times + 2\check{\boldsymbol{\xi}}\check{\boldsymbol{\xi}}^T$$
$$= \begin{bmatrix} \xi_0^2+\xi_1^2-\xi_2^2-\xi_3^2 & 2\xi_1\xi_2-2\xi_0\xi_3 & 2\xi_0\xi_2+2\xi_1\xi_3 \\ 2\xi_0\xi_3+2\xi_1\xi_2 & \xi_0^2-\xi_1^2+\xi_2^2-\xi_3^2 & 2\xi_2\xi_3-2\xi_0\xi_1 \\ 2\xi_1\xi_3-2\xi_0\xi_2 & 2\xi_0\xi_1+2\xi_2\xi_3 & \xi_0^2-\xi_1^2-\xi_2^2+\xi_3^2 \end{bmatrix}.$$

Velocity

- Linear velocity $\dot{\mathbf{r}}_{AB}$
  - Representation $\dot{\boldsymbol{\chi}}_P$

$$\dot{\mathbf{r}} = \mathbf{E}_P\left(\boldsymbol{\chi}_P\right)\dot{\boldsymbol{\chi}}_P$$
$$\dot{\boldsymbol{\chi}}_P = \mathbf{E}_P^{-1}\left(\boldsymbol{\chi}_P\right)\dot{\mathbf{r}}$$

e.g. cylindrical coordinates $\mathbf{E}_{P_z}\left(\boldsymbol{\chi}_{P_z}\right) = \begin{bmatrix} \cos\theta & -\rho\sin\theta & 0 \\ \sin\theta & \rho\cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$

- Angular velocity $\left[{}_{\mathcal{A}}\boldsymbol{\omega}_{\mathcal{AB}}\right]_\times = \dot{\mathbf{C}}_{\mathcal{AB}} \cdot \mathbf{C}_{\mathcal{AB}}^T$

$$\left[{}_{\mathcal{A}}\boldsymbol{\omega}_{\mathcal{AB}}\right]_\times = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}, \qquad {}_{\mathcal{A}}\boldsymbol{\omega}_{\mathcal{AB}} = \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix}$$

  - Representation ${}_{\mathcal{A}}\boldsymbol{\omega}_{\mathcal{AB}} = \mathbf{E}_R(\boldsymbol{\chi}_R) \cdot \dot{\boldsymbol{\chi}}_R$
  - e.g. quaternions

$$\mathbf{E}_{R,quat} = 2\mathbf{H}(\boldsymbol{\xi}),$$
$$\mathbf{E}_{R,quat}^{-1} = \frac{1}{2}\mathbf{H}(\boldsymbol{\xi})^T$$

$$\mathbf{H}(\boldsymbol{\xi}) = \begin{bmatrix} -\check{\boldsymbol{\xi}} & \left[\check{\boldsymbol{\xi}}\right]_\times + \xi_0\mathbb{I}_{3\times3} \end{bmatrix} \in \mathbb{R}^{3\times4}$$
$$= \begin{bmatrix} -\xi_1 & \xi_0 & -\xi_3 & \xi_2 \\ -\xi_2 & \xi_3 & \xi_0 & -\xi_1 \\ -\xi_3 & -\xi_2 & \xi_1 & \xi_0 \end{bmatrix}.$$

Kinematics of Systems of Bodies

- Generalized coordinates $\mathbf{q} = \begin{pmatrix} q_1 \\ \vdots \\ q_n \end{pmatrix}$
- End-effector position and orientation $\mathbf{x}_e = \begin{pmatrix} \mathbf{r}_e \\ \boldsymbol{\phi}_e \end{pmatrix} \in SE(3)$ paramterized by $\boldsymbol{\chi}_e = \begin{pmatrix} \boldsymbol{\chi}_{eP} \\ \boldsymbol{\chi}_{eR} \end{pmatrix} = \begin{pmatrix} \chi_1 \\ \vdots \\ \chi_m \end{pmatrix} \in \mathbb{R}^m$
- Forward kinematics $\boldsymbol{\chi}_e = \boldsymbol{\chi}_e(\mathbf{q})$
- Forward differential kinematics
  - Analytic

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

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

Depending on parameterization!!

  - Geometric

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

Independent of parameterization

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

$$\downarrow$$

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

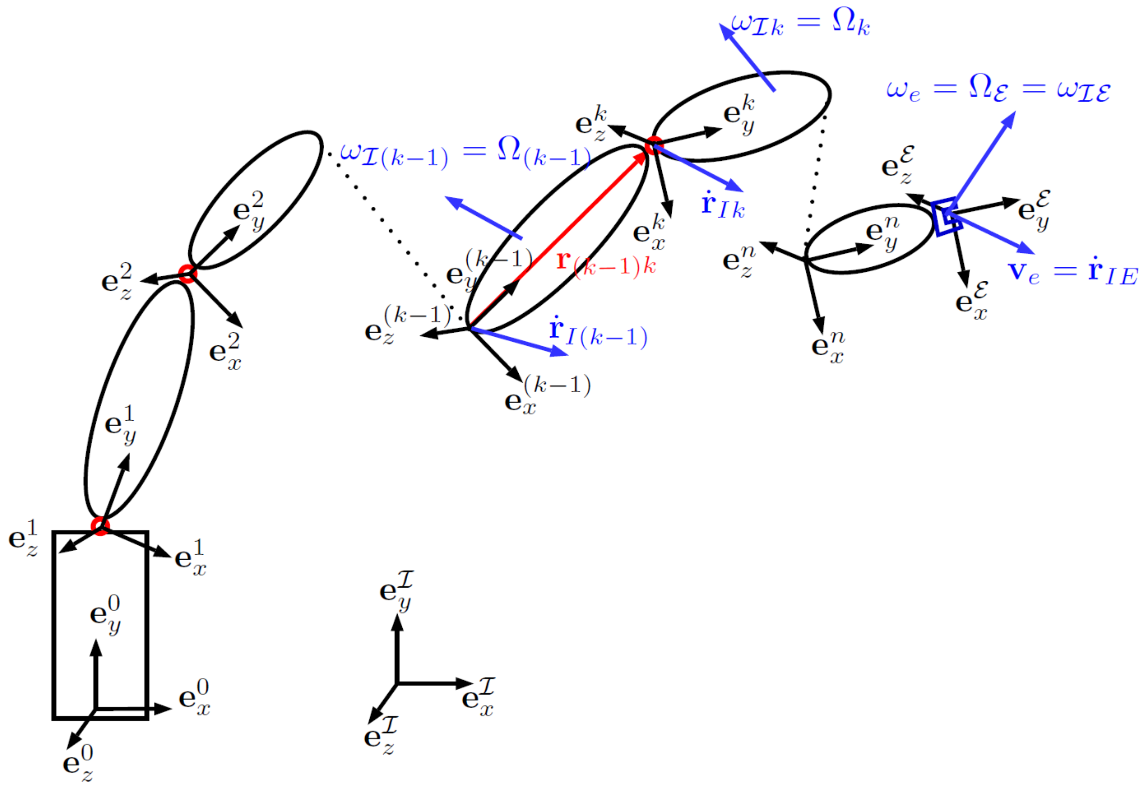
Geometric Jacobian Derivation

- Linear velocity

$$\dot{\mathbf{r}}_{IE} = \underbrace{\left[\mathbf{n}_1 \times \mathbf{r}_{1(n+1)} \quad \mathbf{n}_2 \times \mathbf{r}_{2(n+1)} \quad \dots \quad \mathbf{n}_n \times \mathbf{r}_{n(n+1)}\right]}_{\mathbf{J}_{e0_P}} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{pmatrix}$$

- Angular velocity

$$\omega_{\mathcal{IE}} = \sum_{i=1}^{n} \mathbf{n}_i \dot{q}_i = \underbrace{\left[\mathbf{n}_1 \quad \mathbf{n}_2 \quad \dots \quad \mathbf{n}_n\right]}_{\mathbf{J}_{e0_R}} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{pmatrix}$$



Analytical and Kinematic Jacobian

- Analytical Jacobian

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

- Geometric (or basic) Jacobian

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

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

Analytical Jacobian:

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

Geometric (or basic) Jacobian:

- Relates end-effector velocity to generalized velocities
- Unique for every robot
- Used in most cases

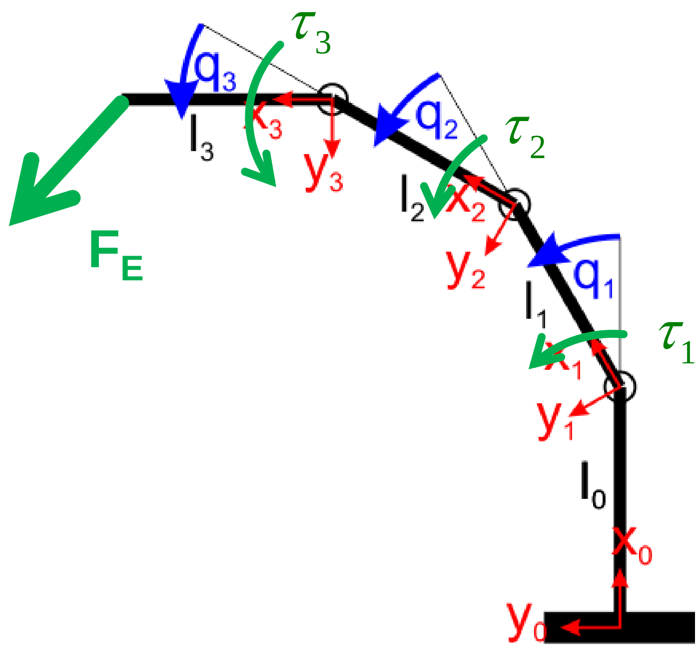
Importance of Jacobian

- 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

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}}$$
$$\boldsymbol{\tau} = \mathbf{J}^T \mathbf{F}$$



Inverse Differential Kinematics

- Differential kinematics $\mathbf{w}_e = \mathbf{J}_{e0}\dot{\mathbf{q}}$
- Inverse differential kinematics $\dot{\mathbf{q}} = \mathbf{J}_{e0}^{+}\mathbf{w}_e^*$
  - Singularity: minimizing $\left\|\mathbf{w}_e^* - \mathbf{J}_{e0}\dot{\mathbf{q}}\right\|^2$
  - Redundancy: $\dot{\mathbf{q}} = \mathbf{J}_{e0}^{+}\mathbf{w}_e^* + \mathbf{N}\dot{\mathbf{q}}_0$ null-space projection matrix $\mathbf{N} = \mathcal{N}\left(\mathbf{J}_{e0}\right)$
- Multi-task control: $task_i := \{\mathbf{J}_i, \mathbf{w}_i^*\}$
  - Equal priority

$$\dot{\mathbf{q}} = \underbrace{\begin{bmatrix} \mathbf{J}_1 \\ \vdots \\ \mathbf{J}_{n_t} \end{bmatrix}^{+}}_{\bar{\mathbf{J}}} \underbrace{\begin{pmatrix} \mathbf{w}_1^* \\ \vdots \\ \mathbf{w}_{n_t}^* \end{pmatrix}}_{\bar{\mathbf{w}}}$$

Multi-task with prioritization

$$\dot{\mathbf{q}} = \sum_{i=1}^{n_t} \mathbf{N}_i \dot{\mathbf{q}}_i, \quad \text{with} \quad \dot{\mathbf{q}}_i = (\mathbf{J}_i \mathbf{N}_i)^{+} \left(\mathbf{w}_i^* - \mathbf{J} \sum_{k=1}^{i-1} \mathbf{N}_k \dot{\mathbf{q}}_k \right)$$

Inverse Kinematics

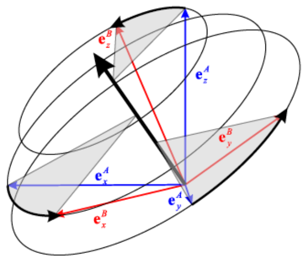
- Numerical approach $\Delta\chi_e = \mathbf{J}_{eA}\Delta\mathbf{q}$

Algorithm 1 Numerical Inverse Kinematics

$\mathbf{q} \leftarrow \mathbf{q}^0$ ▷ Start configuration
while $\|\chi_e^* - \chi_e(\mathbf{q})\| > tol$ **do** ▷ While the solution is not reached
    $\mathbf{J}_{eA} \leftarrow \mathbf{J}_{eA}(\mathbf{q}) = \frac{\partial \chi_e}{\partial \mathbf{q}}(\mathbf{q})$ ▷ Evaluate Jacobian for $\mathbf{q}$
    $\mathbf{J}_{eA}^+ \leftarrow (\mathbf{J}_{eA})^+$ ▷ Calculate the pseudo inverse
    $\Delta\chi_e \leftarrow \chi_e^* - \chi_e(\mathbf{q})$ ▷ Find the end-effector configuration error vector
    $\mathbf{q} \leftarrow \mathbf{q} + \mathbf{J}_{eA}^+\Delta\chi_e$ ▷ Update the generalized coordinates
end while

Position/Rotation Errors and Trajectory Control

- Position error $$\Delta \mathbf{r}_e^t = \mathbf{r}_e^*(t) - \mathbf{r}_e(\mathbf{q}^t)$$
  - Trajectory control with position-error feedback $$\dot{\mathbf{q}} = \mathbf{J}_{e0_P}^{+}(\dot{\mathbf{r}}^* + k_{PP}\Delta \mathbf{r}_e^t)$$



- Rotation error $\Delta\varphi$ is not $\varphi^* - \varphi^t$ ➡ $$\mathbf{C}_{\mathcal{GS}}(\Delta\varphi) = \mathbf{C}_{\mathcal{GI}}(\varphi^*)\mathbf{C}_{\mathcal{SI}}^T(\varphi^t)$$
  - Trajectory control with rotation-error feedback $$\dot{\mathbf{q}} = \mathbf{J}_{e0_R}^{+}(\omega(t)_e^* + k_{PR}\Delta\varphi)$$

Floating Base Kinematics

- Describe system by base and joint coordinates $\mathbf{q} = \begin{pmatrix} \mathbf{q}_b \\ \mathbf{q}_j \end{pmatrix}$

- Base coordinates: rotation and position of base $\mathbf{q}_b = \begin{pmatrix} \mathbf{q}_{b_P} \\ \mathbf{q}_{b_R} \end{pmatrix} \in \mathbb{R}^3 \times SO(3)$

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

Multi-body Dynamics

- We learned how to get the equation of motion in joint space
  - Newton-Euler
  - Projected Newton-Euler
  - Lagrange II

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{b}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) + \mathbf{J}_c^T\mathbf{F}_c = \boldsymbol{\tau}$$

$\ddot{\mathbf{q}}$	Generalized accelerations
$\mathbf{M}(\mathbf{q})$	Mass matrix
$\mathbf{b}(\mathbf{q},\dot{\mathbf{q}})$	Centrifugal and Coriolis forces
$\mathbf{g}(\mathbf{q})$	Gravity forces
$\boldsymbol{\tau}$	Generalized forces
$\mathbf{F}_c$	External forces
$\mathbf{J}_c$	Contact Jacobian

- Started from the principle for virtual work

$$\delta W = \int_{\mathcal{B}} \delta\mathbf{r}^T \cdot (\ddot{\mathbf{r}}\mathrm{d}m - \mathrm{d}\mathbf{F}_{ext}) = 0$$

$\mathrm{d}\mathbf{F}_{ext}$	external forces acting on element i
$\ddot{\mathbf{r}}$	acceleration of element i
dm	mass of element i
$\delta\mathbf{r}$	virtual displacement of element i

Impulse and angular momentum

- Use the following definitions

$\mathbf{p}_S = m\mathbf{v}_S$	linear momentum
$\mathbf{N}_S = \mathbf{\Theta}_S\mathbf{\Omega}_S$	angular momentum
$\dot{\mathbf{p}}_S = m\mathbf{a}_S$	change in linear momentum
$\dot{\mathbf{N}}_S = \mathbf{\Theta}_S\mathbf{\Psi} + \mathbf{\Omega} \times \mathbf{\Theta}_S\mathbf{\Omega}$	change in angular momentum

- Conservation of impulse and angular momentum

Newton

$$0 = \delta W = \begin{pmatrix} \delta\mathbf{r}_s \\ \delta\mathbf{\Phi} \end{pmatrix}^T \left(\begin{pmatrix} \dot{\mathbf{p}}_S \\ \dot{\mathbf{N}}_S \end{pmatrix} - \begin{pmatrix} \mathbf{F}_{ext} \\ \mathbf{T}_{ext} \end{pmatrix} \right) \quad \forall \begin{pmatrix} \delta\mathbf{r}_s \\ \delta\mathbf{\Phi} \end{pmatrix}$$

Euler

A free body can move
In all directions

$$\dot{\mathbf{p}}_S = \mathbf{F}_{ext}$$
$$\dot{\mathbf{N}}_S = \mathbf{T}_{ext}$$

External forces and moments

Change in impulse and angular momentum

Projected Newton Euler

- Consider only directions the system can move (c.f. generalized coordinates)

$$0 = \delta W = \delta \mathbf{q}^T \sum_{i=1}^{n_b} \underbrace{\begin{pmatrix} \mathbf{J}_{S_i} \\ \mathbf{J}_{R_i} \end{pmatrix}^T \begin{pmatrix} m\mathbf{J}_{S_i} \\ \mathbf{\Theta}_{S_i}\mathbf{J}_{R_i} \end{pmatrix}}_{\mathbf{M}(\mathbf{q})} \ddot{\mathbf{q}} + \begin{pmatrix} \mathbf{J}_{S_i} \\ \mathbf{J}_{R_i} \end{pmatrix}^T \underbrace{\begin{pmatrix} m\dot{\mathbf{J}}_{S_i}\dot{\mathbf{q}} \\ \mathbf{\Theta}_{S_i}\dot{\mathbf{J}}_R\dot{\mathbf{q}} + \mathbf{J}_{R_i}\dot{\mathbf{q}} \times \mathbf{\Theta}_{S_i}\mathbf{J}_{R_i}\dot{\mathbf{q}} \end{pmatrix}}_{\mathbf{b}(\mathbf{q},\dot{\mathbf{q}})} - \underbrace{\begin{pmatrix} \mathbf{J}_{P_i} \\ \mathbf{J}_{R_i} \end{pmatrix}^T \begin{pmatrix} \mathbf{F}_{ext,i} \\ \mathbf{T}_{ext,i} \end{pmatrix}}_{\mathbf{g}(\mathbf{q})} \quad \forall \delta \mathbf{q}$$

- Resulting in

$$\mathbf{M} = \sum_{i=1}^{n_b} \left({}_{\mathcal{A}}\mathbf{J}_{S_i}^T \cdot m \cdot {}_{\mathcal{A}}\mathbf{J}_{S_i} + {}_{\mathcal{B}}\mathbf{J}_{R_i}^T \cdot {}_{\mathcal{B}}\mathbf{\Theta}_{S_i} \cdot {}_{\mathcal{B}}\mathbf{J}_{R_i} \right)$$

$$\mathbf{b} = \sum_{i=1}^{n_b} \left({}_{\mathcal{A}}\mathbf{J}_{S_i}^T \cdot m \cdot {}_{\mathcal{A}}\dot{\mathbf{J}}_{S_i} \cdot \dot{\mathbf{q}} + {}_{\mathcal{B}}\mathbf{J}_{R_i}^T \cdot \left({}_{\mathcal{B}}\mathbf{\Theta}_{S_i} \cdot {}_{\mathcal{B}}\dot{\mathbf{J}}_{R_i} \cdot \dot{\mathbf{q}} + {}_{\mathcal{B}}\mathbf{\Omega}_{S_i} \times {}_{\mathcal{B}}\mathbf{\Theta}_{S_i} \cdot {}_{\mathcal{B}}\mathbf{\Omega}_{S_i} \right) \right)$$

$$\mathbf{g} = \sum_{i=1}^{n_b} \left(-{}_{\mathcal{A}}\mathbf{J}_{S_i}^T \, {}_{\mathcal{A}}\mathbf{F}_{g,i} \right)$$

Lagrange II

- Get equation of motion from $$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\mathbf{q}}}\right)-\frac{\partial T}{\partial \mathbf{q}}+\frac{\partial U}{\partial \dot{\mathbf{q}}}=\boldsymbol{\tau}$$

  - Kinetic energy $$\mathcal{T}=\sum_{i=1}^{n_b}\left(\frac{1}{2}m_i\,{}_{\mathcal{A}}\dot{\mathbf{r}}_{S_i}^T\,{}_{\mathcal{A}}\dot{\mathbf{r}}_{S_i}+\frac{1}{2}{}_{\mathcal{B}}\boldsymbol{\Omega}_{S_i}^T\cdot{}_{\mathcal{B}}\boldsymbol{\Theta}_{S_i}\cdot{}_{\mathcal{B}}\boldsymbol{\Omega}_{S_i}\right)$$

  - Potential energy $$\mathbf{F}_{g_i}=m_i\,g\,{}_I\mathbf{e}_g$$ $$\mathcal{U}_g=-\sum_{i=1}^{n_b}\mathbf{r}_{S_i}^T\,\mathbf{F}_{g_i}$$ $$\mathcal{U}_{E_j}=\frac{1}{2}k_j\left(d(\mathbf{q})-d_0\right)^2$$

Dynamic Control Methods

- Joint impedance control
- Inverse dynamics control
- Generalized motion and force control

Joint Impedance Control

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{b}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau}$$

- Torque as function of position and velocity error $\boldsymbol{\tau}^* = \mathbf{k}_p(\mathbf{q}^* - \mathbf{q}) + \mathbf{k}_d(\dot{\mathbf{q}}^* - \dot{\mathbf{q}})$

- Closed loop behavior

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{b}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} = \mathbf{k}_p(\mathbf{q}^* - \mathbf{q}) + \mathbf{k}_d(\dot{\mathbf{q}}^* - \dot{\mathbf{q}})$$

  - ➢ Static offset due to gravity

- Impedance control and gravity compensation

$$\boldsymbol{\tau}^* = \mathbf{k}_p(\mathbf{q}^* - \mathbf{q}) + \mathbf{k}_d(\dot{\mathbf{q}}^* - \dot{\mathbf{q}}) + \hat{\mathbf{g}}(\mathbf{q})$$

Estimated gravity term

Simple setup…
but configuration dependent load

Inverse Dynamics Control

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{b}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau}$$

- Compensate for system dynamics $\boldsymbol{\tau} = \hat{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{q}}^* + \hat{\mathbf{b}}(\mathbf{q},\dot{\mathbf{q}}) + \hat{\mathbf{g}}(\mathbf{q})$
- In case of no modeling errors,
  - the desired dynamics can be perfectly prescribed $\mathbb{I}\ddot{\mathbf{q}} = \ddot{\mathbf{q}}^*$
- PD-control law $\mathbb{I}\ddot{\mathbf{q}} = \ddot{\mathbf{q}}^* = \mathbf{k}_p(\mathbf{q}^* - \mathbf{q}) + \mathbf{k}_d(\dot{\mathbf{q}}^* - \dot{\mathbf{q}})$

Can achieve great performance…
but requires accurate modeling

Operational Space Control

Generalized framework to control motion and force

$$\boldsymbol{\tau} = \mathbf{J}_e^T \mathbf{F}_e$$

$$\mathbf{\Lambda} = \left(\mathbf{J}_e \mathbf{M}^{-1} \mathbf{J}_e^T\right)^{-1}$$
$$\boldsymbol{\mu} = \mathbf{\Lambda} \mathbf{J}_e \mathbf{M}^{-1} \mathbf{b} - \mathbf{\Lambda} \dot{\mathbf{J}}_e \dot{\mathbf{q}}$$
$$\mathbf{p} = \mathbf{\Lambda} \mathbf{J}_e \mathbf{M}^{-1} \mathbf{g}$$

- Joint-space dynamics

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau}$$

End-effector dynamics

$$\mathbf{\Lambda}\dot{\mathbf{w}}_e + \boldsymbol{\mu} + \mathbf{p} = \mathbf{F}_e$$

  - Determine the corresponding joint torque

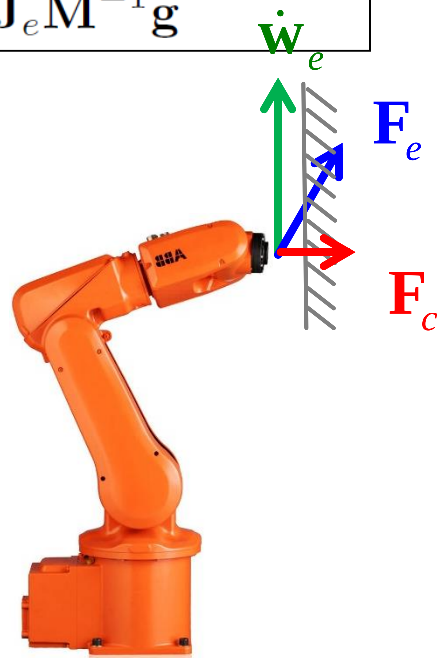
$$\boldsymbol{\tau}^* = \hat{\mathbf{J}}^T \left(\hat{\mathbf{\Lambda}}_e \dot{\mathbf{w}}_e^* + \hat{\boldsymbol{\mu}} + \hat{\mathbf{p}}\right)$$

- Extend end-effector dynamics in contact with contact force

$$\mathbf{F}_c + \mathbf{\Lambda}\dot{\mathbf{w}}_e + \boldsymbol{\mu} + \mathbf{p} = \mathbf{F}_e$$

- Introduce selection matrices to separate motion force directions

$$\boldsymbol{\tau}^* = \hat{\mathbf{J}}^T \left(\hat{\mathbf{\Lambda}}\mathbf{S}_M \dot{\mathbf{w}}_e + \mathbf{S}_F \mathbf{F}_c + \hat{\tilde{\boldsymbol{\mu}}} + \hat{\mathbf{p}}\right)$$



Inverse Dynamics of Floating Base Systems

- Equation of motion of floating base systems

$$\mathbf{M}(\mathbf{q})\dot{\mathbf{u}} + \mathbf{b}(\mathbf{q},\mathbf{u}) + \mathbf{g}(\mathbf{q}) + \mathbf{J}_c^T\mathbf{F}_c = \mathbf{S}^T\boldsymbol{\tau}$$

- Support-consistent $\mathbf{N}_c^T(\mathbf{M}\ddot{\mathbf{q}} + \mathbf{b} + \mathbf{g}) = \mathbf{N}_c^T\mathbf{S}^T\boldsymbol{\tau}$
- Inverse-dynamics $\boldsymbol{\tau}^* = (\mathbf{N}_c^T\mathbf{S}^T)^+\mathbf{N}_c^T(\mathbf{M}\ddot{\mathbf{q}} + \mathbf{b} + \mathbf{g})$
- Multiple solutions $\boldsymbol{\tau}^* = (\mathbf{N}_c^T\mathbf{S}^T)^+\mathbf{N}_c^T(\mathbf{M}\ddot{\mathbf{q}}^* + \mathbf{b} + \mathbf{g}) + \mathcal{N}(\mathbf{N}_c^T\mathbf{S}^T)\boldsymbol{\tau}_0^*$

Operational Space Control as Quadratic Program

A general problem

$$\min_{\mathbf{x}} \quad \|\mathbf{A}_i \mathbf{x} - \mathbf{b}_i\|_2 \qquad \mathbf{x} = \begin{pmatrix} \dot{\mathbf{u}} \\ \mathbf{F}_c \\ \boldsymbol{\tau} \end{pmatrix}$$

- We search for a solution that fulfills the equation of motion

$$\mathbf{M}(\mathbf{q})\dot{\mathbf{u}} + \mathbf{b}(\mathbf{q},\mathbf{u}) + \mathbf{g}(\mathbf{q}) + \mathbf{J}_c^T \mathbf{F}_c = \mathbf{S}^T \boldsymbol{\tau} \Rightarrow \mathbf{A} = \begin{bmatrix} \hat{\mathbf{M}} & \hat{\mathbf{J}}_c^T & -\mathbf{S}^T \end{bmatrix} \qquad \mathbf{b} = -\hat{\mathbf{b}} - \hat{\mathbf{g}}$$

- Motion tasks: $\mathbf{J}\dot{\mathbf{u}} + \dot{\mathbf{J}}\mathbf{u} = \dot{\mathbf{w}}^*$ $\Rightarrow \mathbf{A} = \begin{bmatrix} \hat{\mathbf{J}}_i & 0 & 0 \end{bmatrix} \qquad \mathbf{b} = \dot{\mathbf{w}}^* - \hat{\dot{\mathbf{J}}}_i \mathbf{u}$
- Force tasks: $\mathbf{F}_i = \mathbf{F}_i^*$ $\Rightarrow \mathbf{A} = \begin{bmatrix} 0 & \hat{\mathbf{J}}_i^T & 0 \end{bmatrix} \qquad \mathbf{b} = \mathbf{F}_i^*$
- Torque min: $\min \|\boldsymbol{\tau}\|_2$ $\Rightarrow \mathbf{A} = \begin{bmatrix} 0 & 0 & \mathbb{I} \end{bmatrix} \qquad \mathbf{b} = 0$