



Lecture «Robot Dynamics»: Summary

151-0851-00 V

lecture:	HG F3	Tuesday 10:15 – 12:00, every week
exercise:	HG F3	Wednesday 8:15 – 10:00, according to schedule

Marco Hutter, Roland Siegwart, and Thomas Stastny

17.09.2019	Intro and Outline	Course Introduction; Recapitulation Position, Linear Velocity			
24.09.2019	Kinematics 1	Rotation and Angular Velocity; Rigid Body Formulation, Transformation	25.09.2019	Exercise 1a	Kinematics Modeling the ABB arm
01.10.2019	Kinematics 2	Kinematics of Systems of Bodies; Jacobians	02.10.2019	Exercise 1a	Differential Kinematics of the ABB arm
08.10.2019	Kinematics 3	Kinematic Control Methods: Inverse Differential Kinematics, Inverse Kinematics; Rotation Error; Multi-task Control	09.10.2019	Exercise 1b	Kinematic Control of the ABB Arm
15.10.2019	Dynamics L1	Multi-body Dynamics	16.10.2019	Midterm 1	Programming kinematics with matlab
22.10.2019	Dynamics L2	Floating Base Dynamics	23.10.2019	Exercise 2a	Dynamic Modeling of the ABB Arm
29.10.2019	Dynamics L3	Dynamic Model Based Control Methods	30.10.2019	Exercise 2b	Dynamic Control Methods Applied to the ABB arm
05.11.2019	Legged Robot	Dynamic Modeling of Legged Robots & Control	06.11.2019	Midterm 2	Programming dynamics with matlab
12.11.2019	Case Studies 1	Legged Robotics Case Study	13.11.2019	Exercise 3	Legged robot
19.11.2019	Rotorcraft	Dynamic Modeling of Rotorcraft & Control	20.11.2019		
26.11.2019	Case Studies 2	Rotor Craft Case Study	27.11.2019	Exercise 4	Modeling and Control of Multicopter
03.12.2019	Fixed-wing	Dynamic Modeling of Fixed-wing & Control	04.12.2019		
10.12.2019	Case Studies 3	Fixed-wing Case Study (Solar-powered UAVs - AtlantikSolar, Vertical Take-off and Landing UAVs – Wingtra)	11.12.2019	Exercise 5	Fixed-wing Control and Simulation
17.12.2019	Summery and Outlook	Summery; Wrap-up; Exam			Robot Dynamics - Summary 17.12.2019 2

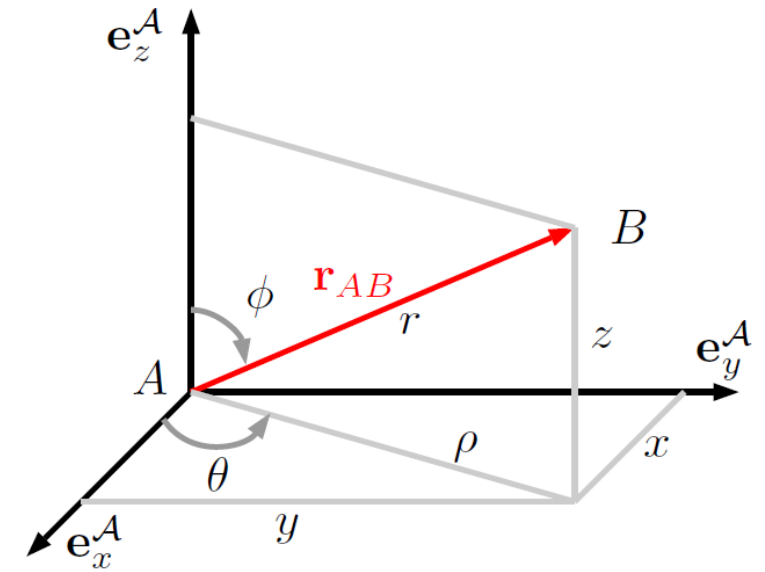
Position

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

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

- Different parameterizations, e.g.

- Cartesian coordinates $\chi_{Pc} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$
 - Position vector ${}^A\mathbf{r} = x\mathbf{e}_x^A + y\mathbf{e}_y^A + z\mathbf{e}_z^A = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$
- Cylindrical coordinates $\chi_{Pz} = \begin{pmatrix} \rho \\ \theta \\ z \end{pmatrix}$
 - Position vector ${}^A\mathbf{r} = \begin{pmatrix} \rho \cos \theta \\ \rho \sin \theta \\ z \end{pmatrix}$



Rotation

- Rotation $\phi_{\mathcal{AB}} \in SO(3)$
- Rotation matrix ${}_{\mathcal{A}}\mathbf{r}_{AP} = [{}_{\mathcal{A}}\mathbf{e}_x^{\mathcal{B}} \quad {}_{\mathcal{A}}\mathbf{e}_y^{\mathcal{B}} \quad {}_{\mathcal{A}}\mathbf{e}_z^{\mathcal{B}}] \cdot {}_{\mathcal{B}}\mathbf{r}_{AP} = \mathbf{C}_{\mathcal{AB}} \cdot {}_{\mathcal{B}}\mathbf{r}_{AP}$
- Different parameterizations, e.g.

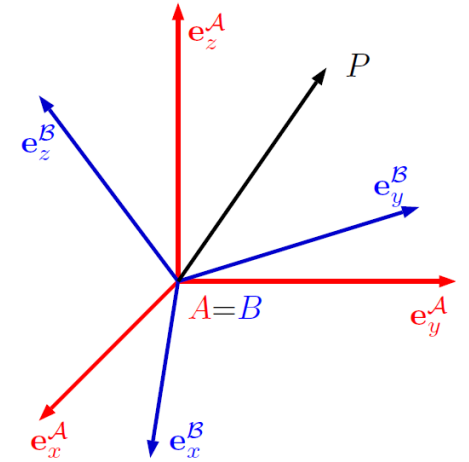
- Euler angles

$$\chi_{R,eulerZYZ} = \begin{pmatrix} z_1 \\ y \\ z_2 \end{pmatrix}$$

- Quaternions

$$\chi_{R,quat} = \xi = \begin{pmatrix} \xi_0 \\ \check{\xi} \end{pmatrix} \in \mathbb{H} \quad \xi = \begin{pmatrix} \cos(\theta/2) \\ \sin(\theta/2)\mathbf{n} \end{pmatrix}$$

- ...



Introduction to algebra with quaternions, e.g.

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

- Relation to rotation matrix:

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

Velocity

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

$$\begin{aligned}\dot{\mathbf{r}} &= \mathbf{E}_P(\boldsymbol{\chi}_P) \dot{\boldsymbol{\chi}}_P \\ \dot{\boldsymbol{\chi}}_P &= \mathbf{E}_P^{-1}(\boldsymbol{\chi}_P) \dot{\mathbf{r}}\end{aligned}$$

e.g. cylindrical coordinates

$$\mathbf{E}_{Pz}(\boldsymbol{\chi}_{Pz}) = \begin{bmatrix} \cos \theta & -\rho \sin \theta & 0 \\ \sin \theta & \rho \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Angular velocity $[_{\mathcal{A}}\boldsymbol{\omega}_{\mathcal{AB}}]_{\times} = \dot{\mathbf{C}}_{\mathcal{AB}} \cdot \mathbf{C}_{\mathcal{AB}}^T$
 - Representation $[_{\mathcal{A}}\boldsymbol{\omega}_{\mathcal{AB}}]_{\times} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}, \quad {}_{\mathcal{A}}\boldsymbol{\omega}_{\mathcal{AB}} = \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix}$
- Representation ${}_{\mathcal{A}}\boldsymbol{\omega}_{\mathcal{AB}} = \mathbf{E}_R(\boldsymbol{\chi}_R) \cdot \dot{\boldsymbol{\chi}}_R$

- e.g. quaternions

$$\mathbf{E}_{R,quat} = 2\mathbf{H}(\boldsymbol{\xi}),$$

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

$$\begin{aligned}\mathbf{H}(\boldsymbol{\xi}) &= \begin{bmatrix} -\check{\boldsymbol{\xi}} & [\check{\boldsymbol{\xi}}]_{\times} + \xi_0 \mathbb{I}_{3 \times 3} \end{bmatrix} \in \mathbb{R}^{3 \times 4} \\ &= \begin{bmatrix} -\xi_1 & \xi_0 & -\xi_3 & \xi_2 \\ -\xi_2 & \xi_3 & \xi_0 & -\xi_1 \\ -\xi_3 & -\xi_2 & \xi_1 & \xi_0 \end{bmatrix}.\end{aligned}$$

Kinematics of Systems of Bodies

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

- Analytic

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

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

Depending on
parameterization!!

- Geometric

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

Independent of
parameterization

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



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

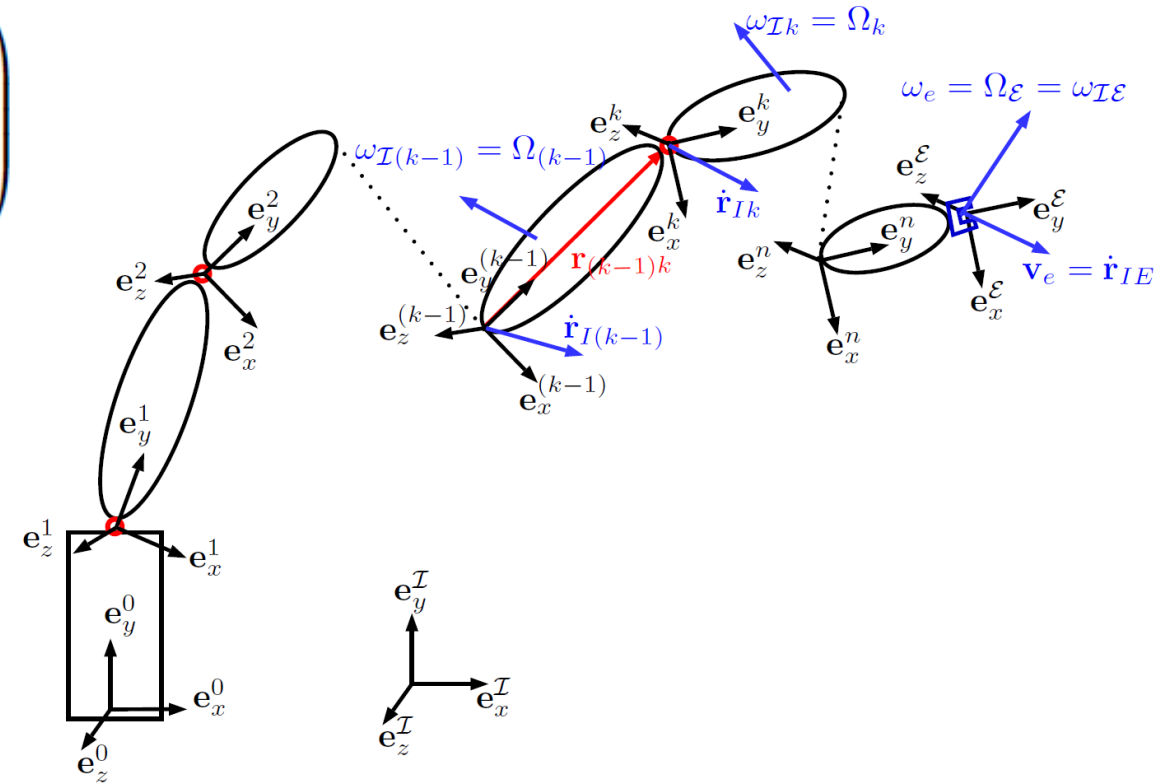
Geometric Jacobian Derivation

- Linear velocity

$$\dot{\mathbf{r}}_{IE} = \underbrace{\begin{bmatrix} \mathbf{n}_1 \times \mathbf{r}_{1(n+1)} & \mathbf{n}_2 \times \mathbf{r}_{2(n+1)} & \dots & \mathbf{n}_n \times \mathbf{r}_{n(n+1)} \end{bmatrix}}_{\mathbf{J}_{e0P}} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{pmatrix}$$

- Angular velocity

$$\omega_{IE} = \sum_{i=1}^n \mathbf{n}_i \dot{q}_i = \underbrace{\begin{bmatrix} \mathbf{n}_1 & \mathbf{n}_2 & \dots & \mathbf{n}_n \end{bmatrix}}_{\mathbf{J}_{e0R}} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{pmatrix}$$



Analytical and Kinematic Jacobian

- Analytical Jacobian

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

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

- 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

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

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

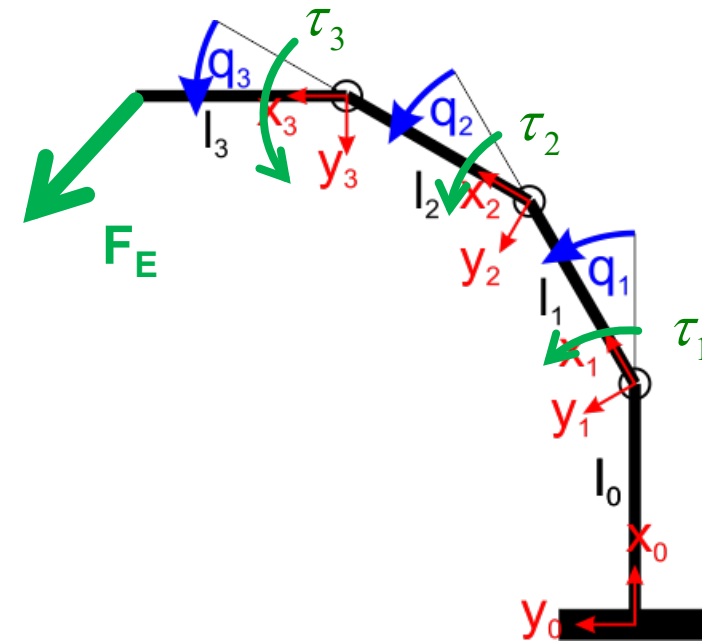
Importance of Jacobian

- 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

$$\begin{aligned}\delta W &= \sum_i \mathbf{f}_i \mathbf{x}_i = \boldsymbol{\tau}^T \delta \mathbf{q} + (-\mathbf{F}_E)^T \delta \mathbf{x}_E \\ &= \boldsymbol{\tau}^T \delta \mathbf{q} + (-\mathbf{F}_E)^T \mathbf{J} \delta \mathbf{q} = 0 \quad \forall \delta \mathbf{q}\end{aligned}$$

➤ Dual problem from principle of virtual work

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{J} \dot{\mathbf{q}} \\ \boldsymbol{\tau} &= \mathbf{J}^T \mathbf{F}\end{aligned}$$



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 $\|\mathbf{w}_e^* - \mathbf{J}_{e0} \dot{\mathbf{q}}\|^2$
 - Redundancy: $\dot{\mathbf{q}} = \mathbf{J}_{e0}^+ \mathbf{w}_e^* + \mathbf{N} \dot{\mathbf{q}}_0$ null-space projection matrix $\mathbf{N} = \mathcal{N}(\mathbf{J}_{e0})$
- 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

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

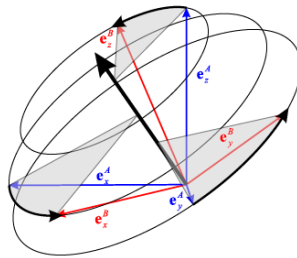
Position/Rotation Errors and Trajectory Control

- Position error $\Delta \mathbf{r}_e^t = \mathbf{r}_e^*(t) - \mathbf{r}_e(\mathbf{q}^t)$
- Trajectory control with position-error feedback $\dot{\mathbf{q}} = \mathbf{J}_{e0P}^+ (\dot{\mathbf{r}}^* + k_{PP} \Delta \mathbf{r}_e^t)$

goal and real

- Rotation error $\Delta \varphi$ is not $\varphi^* - \varphi^t \longrightarrow \mathbf{C}_{gS}(\Delta \varphi) = \mathbf{C}_{gI}(\varphi^*) \mathbf{C}_{SI}^T(\varphi^t)$

- Trajectory control with rotation-error feedback $\dot{\mathbf{q}} = \mathbf{J}_{e0R}^+ (\omega(t)_e^* + k_{PR} \Delta \varphi)$



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

Multi-body Dynamics

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

- Started from the principle for virtual work

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

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

$\ddot{\mathbf{q}}$	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

$d\mathbf{F}_{ext}$	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

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

Newton
Euler
 External forces and moments
 Change in impulse and angular momentum

A free body can move
In all directions

$$\begin{aligned} \dot{\mathbf{p}}_S &= \mathbf{F}_{ext} \\ \dot{\mathbf{N}}_S &= \mathbf{T}_{ext} \end{aligned}$$

Projected Newton Euler

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

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

- Resulting in

$$\begin{aligned} \mathbf{M} &= \sum_{i=1}^{n_b} \left(\mathcal{A} \mathbf{J}_{S_i}^T \cdot m \cdot \mathcal{A} \mathbf{J}_{S_i} + \mathcal{B} \mathbf{J}_{R_i}^T \cdot \mathcal{B} \boldsymbol{\Theta}_{S_i} \cdot \mathcal{B} \mathbf{J}_{R_i} \right) \\ \mathbf{b} &= \sum_{i=1}^{n_b} \left(\mathcal{A} \mathbf{J}_{S_i}^T \cdot m \cdot \mathcal{A} \dot{\mathbf{J}}_{S_i} \cdot \dot{\mathbf{q}} + \mathcal{B} \mathbf{J}_{R_i}^T \cdot \left(\mathcal{B} \boldsymbol{\Theta}_{S_i} \cdot \mathcal{B} \dot{\mathbf{J}}_{R_i} \cdot \dot{\mathbf{q}} + \mathcal{B} \boldsymbol{\Omega}_{S_i} \times \mathcal{B} \boldsymbol{\Theta}_{S_i} \cdot \mathcal{B} \boldsymbol{\Omega}_{S_i} \right) \right) \\ \mathbf{g} &= \sum_{i=1}^{n_b} \left(-\mathcal{A} \mathbf{J}_{S_i}^T \mathcal{A} \mathbf{F}_{g,i} \right) \end{aligned}$$

Lagrange II

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

- 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
$$\mathbf{F}_{g_i} = m_i g \mathbf{e}_g$$

$$\mathcal{U}_g = - \sum_{i=1}^{n_b} \mathbf{r}_{S_i}^T \mathbf{F}_{g_i}$$

$$\mathcal{U}_{E_j} = \frac{1}{2} k_j (d(\mathbf{q}) - d_0)^2$$

External Forces

Cart pendulum example

- Equation of motion without actuation

$$\underbrace{\begin{bmatrix} m_c + m_p & lm_p \cos(\varphi) \\ lm_p \cos(\varphi) & m_p l^2 + \theta_p \end{bmatrix}}_{\mathbf{M}} \ddot{\mathbf{q}} + \underbrace{\begin{pmatrix} -\dot{\varphi}^2 lm_p \sin(\varphi) \\ 0 \end{pmatrix}}_{\mathbf{b}} + \underbrace{\begin{pmatrix} 0 \\ m_p gl \sin(\varphi) \end{pmatrix}}_{\mathbf{g}} = \mathbf{0}$$

- Add actuator for the pendulum

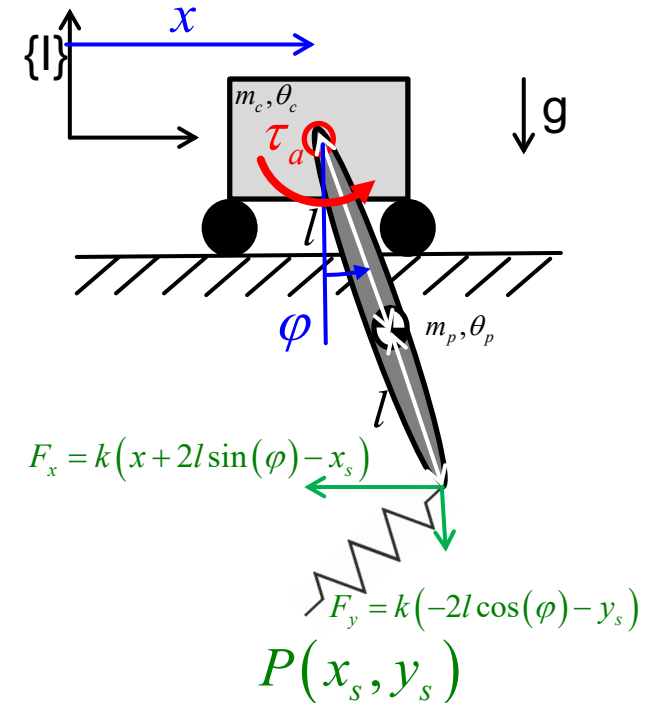
- Action on pendulum $T_p = \tau_a$
- Reaction on cart $T_c = -\tau_a$

$$\left. \begin{array}{l} \mathbf{J}_{Rp} = \begin{bmatrix} 0 & 1 \end{bmatrix} \\ \mathbf{J}_{Rc} = \begin{bmatrix} 0 & 0 \end{bmatrix} \end{array} \right\} \boldsymbol{\tau} = \sum \mathbf{J}_{R,i}^T \mathbf{T}_i = \mathbf{J}_{Rc}^T \mathbf{T}_c + \mathbf{J}_{Rp}^T \mathbf{T}_p = \begin{pmatrix} 0 \\ \tau_a \end{pmatrix}$$

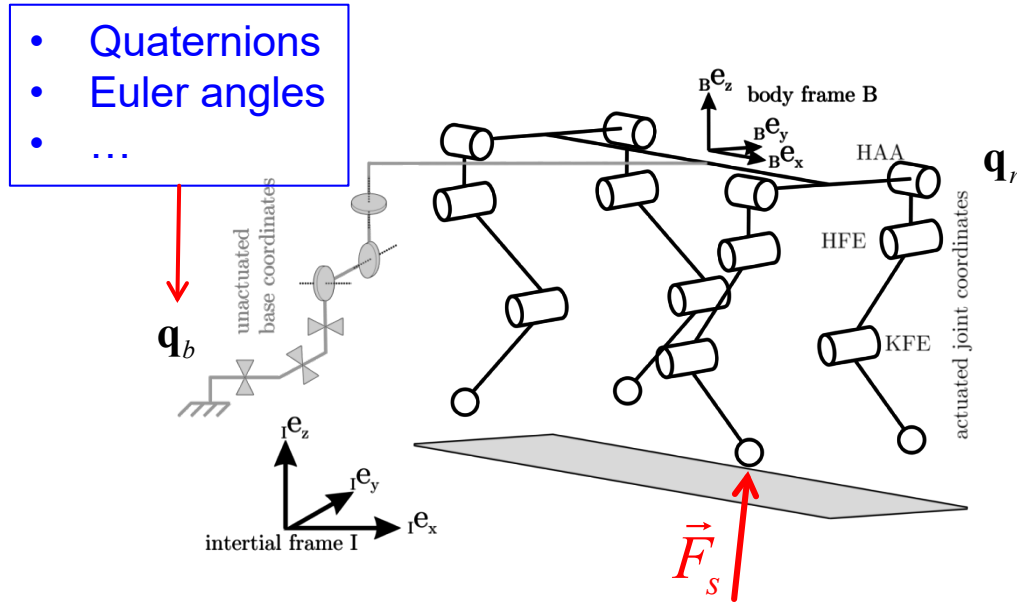
- Add spring to the pendulum

- (world attachment point P, zero length 0, stiffness k)
- Action on pendulum

$$\mathbf{F}_s = \begin{pmatrix} -F_x \\ -F_y \end{pmatrix} \quad \left. \begin{array}{l} \mathbf{r} = \begin{pmatrix} x + 2l \sin(\varphi) \\ -2l \cos(\varphi) \end{pmatrix} \\ \mathbf{J}_s = \frac{\partial \mathbf{r}_s}{\partial \mathbf{q}} = \begin{bmatrix} 1 & 2l \cos(\varphi) \\ 0 & 2l \sin(\varphi) \end{bmatrix} \end{array} \right\} \boldsymbol{\tau} = \mathbf{J}^T \mathbf{F}_s = \begin{pmatrix} -F_x \\ -2l(F_x \cos(\varphi) + F_y \sin(\varphi)) \end{pmatrix}$$



Dynamics of Floating Base Systems



$$\mathbf{q} = \begin{pmatrix} \mathbf{q}_b \\ \mathbf{q}_j \end{pmatrix} \quad \begin{array}{l} \text{Un-actuated base} \\ \text{Actuated joints} \end{array}$$

- EoM from last time

$$\mathbf{M}\ddot{\mathbf{q}}_j + \mathbf{b} + \mathbf{g} = \boldsymbol{\tau}$$

- Not all joint are actuated

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{b} + \mathbf{g} = \mathbf{S}^T \boldsymbol{\tau}$$

- Selection matrix of actuated joints

$$\mathbf{S} = \begin{bmatrix} \mathbf{0}_{n \times 6} & \mathbf{I}_{n \times n} \end{bmatrix} \quad \mathbf{q}_j = \mathbf{S}\mathbf{q}$$

- Contact force acting on system

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{b} + \mathbf{g} = \mathbf{S}^T \boldsymbol{\tau} + \mathbf{J}_s^T \mathbf{F}_s, \text{ acting on system}$$

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{b} + \mathbf{g} + \mathbf{J}_s^T \mathbf{F}_s, \text{ exerted by robot} = \mathbf{S}^T \boldsymbol{\tau}$$

Manipulator:

Legged robot:

UAV:

interaction forces at end-effector

ground contact forces

lift force

Note: for simplicity we don't use here \mathbf{u} but only time derivatives of \mathbf{q}

External Forces

Some notes

- External forces from force elements or actuator

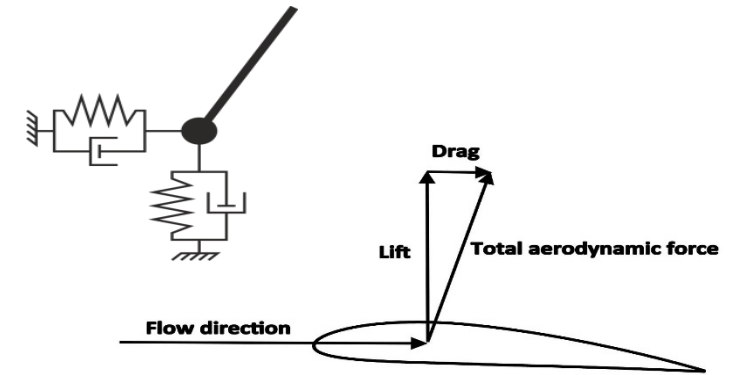
- E.g. soft contact

$$\mathbf{F}_s = k_p (\mathbf{r}_{des} - \mathbf{r}) + k_d (-\dot{\mathbf{r}})$$

- Aerodynamics

$$F_s = \frac{1}{2} \rho c_v A c_L$$

- External forces from constraints...



Support Consistent Dynamics

- Equation of motion $\mathbf{M}\ddot{\mathbf{q}} + \mathbf{b} + \mathbf{g} + \mathbf{J}_s^T \mathbf{F}_s = \mathbf{S}^T \boldsymbol{\tau} \quad (1)$
 - Cannot directly be used for control due to the occurrence of contact forces

- Contact constraint

$$\mathbf{r}_s = \mathbf{J}_s \ddot{\mathbf{q}} + \dot{\mathbf{J}}_s \dot{\mathbf{q}} = \mathbf{0}$$

- Contact force

- Back-substitute in (1),
replace $\dot{\mathbf{J}}_s \dot{\mathbf{q}} = -\mathbf{J}_s \ddot{\mathbf{q}}$ and use
support null-space projection

$$\mathbf{F}_s = \left(\mathbf{J}_s \mathbf{M}^{-1} \mathbf{J}_s^T \right)^{-1} \left(\mathbf{J}_s \mathbf{M}^{-1} \left(\mathbf{S}^T \boldsymbol{\tau} - (\mathbf{b} + \mathbf{g}) \right) + \dot{\mathbf{J}}_s \dot{\mathbf{q}} \right)$$

$$\mathbf{N}_s = \mathbf{I} - \mathbf{M}^{-1} \mathbf{J}_s^T \left(\mathbf{J}_s \mathbf{M}^{-1} \mathbf{J}_s^T \right)^{-1} \mathbf{J}_s$$

$$\mathbf{J}_s \mathbf{N}_s = \mathbf{0}$$

- Support consistent dynamics

$$\mathbf{N}_s^T \mathbf{M} \ddot{\mathbf{q}} + \mathbf{N}_s^T (\mathbf{b} + \mathbf{g}) = \mathbf{N}_s^T \mathbf{S}^T \boldsymbol{\tau}$$

Dynamic Control Methods

- Joint impedance control (w/o and w/ gravity compensation)
- 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

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

star: desired

- 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

- 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

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

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

$$\Lambda = (\mathbf{J}_e \mathbf{M}^{-1} \mathbf{J}_e^T)^{-1}$$

$$\boldsymbol{\mu} = \Lambda \mathbf{J}_e \mathbf{M}^{-1} \mathbf{b} - \Lambda \dot{\mathbf{J}}_e \dot{\mathbf{q}}$$

$$\mathbf{p} = \Lambda \mathbf{J}_e \mathbf{M}^{-1} \mathbf{g}$$

- Determine the corresponding joint torque

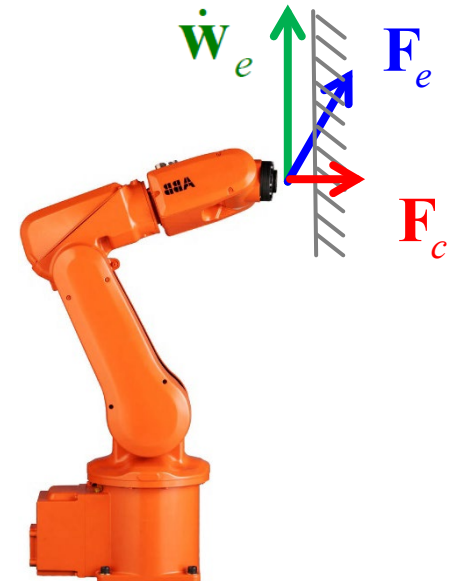
$$\boldsymbol{\tau}^* = \hat{\mathbf{J}}^T \left(\hat{\Lambda}_e \dot{\mathbf{w}}_e^* + \hat{\boldsymbol{\mu}} + \hat{\mathbf{p}} \right)$$

- Extend end-effector dynamics in contact with contact force

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

- Introduce selection matrices to separate motion force directions

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



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

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

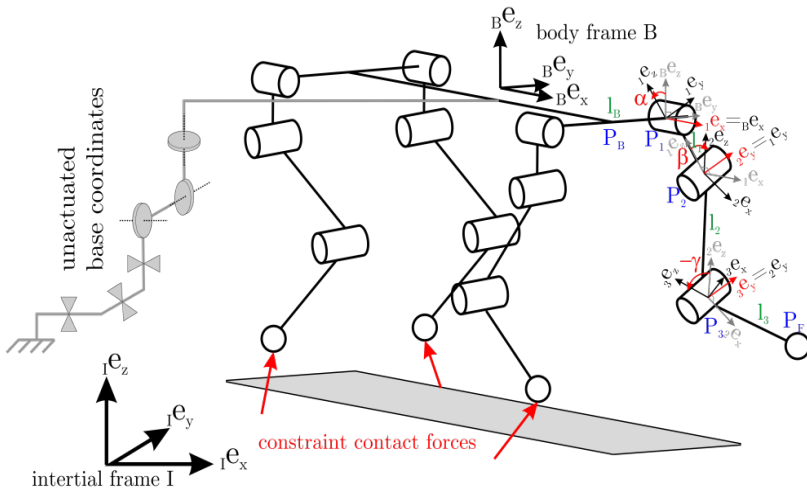
- We search for a solution that fulfills the equation of motion

$$\mathbf{M}(\mathbf{q}) \dot{\mathbf{u}} + \mathbf{b}(\mathbf{q}, \mathbf{u}) + \mathbf{g}(\mathbf{q}) + \mathbf{J}_c^T \mathbf{F}_c = \mathbf{S}^T \boldsymbol{\tau} \quad \Rightarrow \quad \mathbf{A} = [\hat{\mathbf{M}} \quad \hat{\mathbf{J}}_c^T \quad -\mathbf{S}^T] \quad \mathbf{b} = -\hat{\mathbf{b}} - \hat{\mathbf{g}}$$

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

Kinematics of Floating Base / Mobile Systems

example



- Quadrupedal robot
- Static walking
- 3 legs in stance [NR 1,2,3]
- 1 in swing [NR 4]

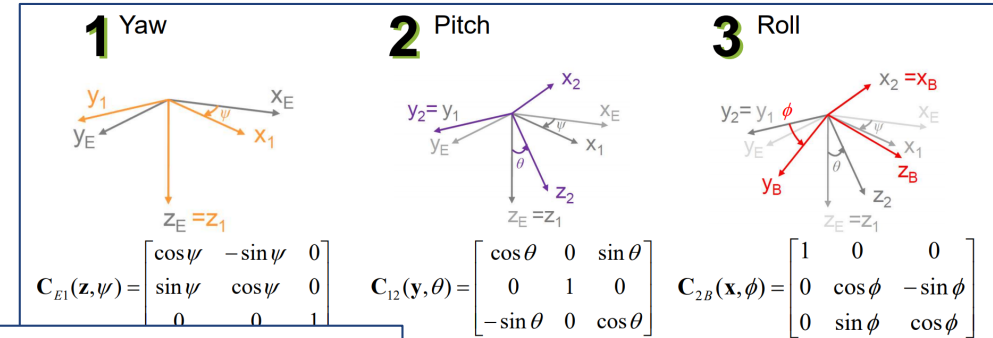
1. How many generalized coordinates?
2. How many base coordinates?
3. How many actuated joint coordinates?
4. How many contact constraints?
5. Write down the contact constraint
6. How many DoFs remain adjustable?
7. Which DoFs remain adjustable?
8. Given a desired swing velocity, what is the generalized velocity?
9. Is it unique?
10. Is it possible to follow the desired swing trajectory without moving the joints of leg 4? How?

Modeling of Rotorcrafts

Representing Altitude

Rotational Velocity

Gimbal lock



- Split angular velocity into the three basic rotations

$${}_B \boldsymbol{\omega} = {}_B \boldsymbol{\omega}_{roll} + {}_B \boldsymbol{\omega}_{pitch} + {}_B \boldsymbol{\omega}_{yaw}$$

- Angular velocity due to change in roll angle

$${}_B \boldsymbol{\omega}_{roll} = (\dot{\phi}, 0, 0)^T$$

$${}_B \boldsymbol{\omega}_{roll} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix}$$

- Angular velocity due to change in pitch angle

$${}_B \boldsymbol{\omega}_{pitch} = C_{2B}^T(x, \phi) (0, \dot{\theta}, 0)^T$$

$${}_B \boldsymbol{\omega}_{pitch} = C_{2B}^T(x, \phi) \cdot \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \dot{\theta} \cos \phi \\ \dot{\theta} \sin \phi \end{bmatrix}$$

- Angular velocity due to change in yaw angle

$${}_B \boldsymbol{\omega}_{yaw} = [C_{12}(y, \theta) C_{2B}(x, \phi)]^T (0, 0, \dot{\psi})^T$$

$${}_B \boldsymbol{\omega}_{yaw} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \cdot \begin{bmatrix} \dot{\psi} \cos \phi \\ 0 \\ \dot{\psi} \sin \phi \end{bmatrix}$$

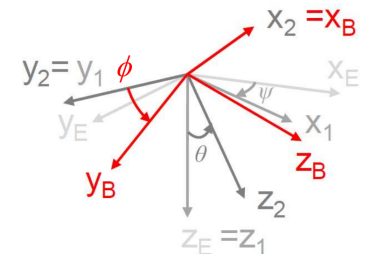
- Relation between Tait-Bryan angles χ and angular velocities

$$E_r \dot{\chi}_r = {}_B \boldsymbol{\omega} \quad E_r = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \sin \phi \cos \theta \\ 0 & -\sin \phi & \cos \phi \cos \theta \end{bmatrix}; \quad \dot{\chi}_r = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

\Rightarrow Singularity for $\theta = \pm 90^\circ$

- Linearized relation at hover

$$E_r|_{\phi=0, \theta=0} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = {}_B \boldsymbol{\omega}$$



Modeling of Rotorcrafts

Body dynamics

$$\mathbf{M}(\vec{\varphi})\ddot{\vec{\varphi}} + \vec{b}(\vec{\varphi}, \dot{\vec{\varphi}}) + \vec{g}(\vec{\varphi}) + \mathbf{J}_{ex}^T \vec{F}_{ex} = \mathbf{S}^T \vec{\tau}_{act}$$

- Change of momentum and spin in the body frame

always in B frame

$$\begin{bmatrix} mE_{3 \times 3} & 0 \\ 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} {}_B \dot{\mathbf{v}} \\ {}_B \dot{\boldsymbol{\omega}} \end{bmatrix} + \begin{bmatrix} {}_B \boldsymbol{\omega} \times m {}_B \mathbf{v} \\ {}_B \boldsymbol{\omega} \times \mathbf{I} {}_B \boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} {}_B \mathbf{F} \\ {}_B \mathbf{M} \end{bmatrix}$$

$E_{3 \times 3}$: Identity matrix

Torque!!

- Position in the inertial frame and the attitude

$${}_E \dot{\mathbf{x}} = \mathbf{C}_{EB} {}_B \mathbf{v} \quad {}_E \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = {}_B \boldsymbol{\omega}$$

- Forces and moments
 - Aerodynamics and gravity

$$\begin{aligned} {}_B \mathbf{F} &= {}_B \mathbf{F}_G + {}_B \mathbf{F}_{Aero} \\ {}_B \mathbf{M} &= {}_B \mathbf{M}_{Aero} \end{aligned} \quad {}_B \mathbf{F}_G = \mathbf{C}_{EB}^T \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}$$

- Hover forces
 - Thrust forces in the shaft direction

$${}_B \mathbf{F}_{Aero} = \sum_{i=1}^4 \begin{bmatrix} 0 \\ 0 \\ -T_i \end{bmatrix}$$

$$T_i = b_i \omega_{p,i}^2$$

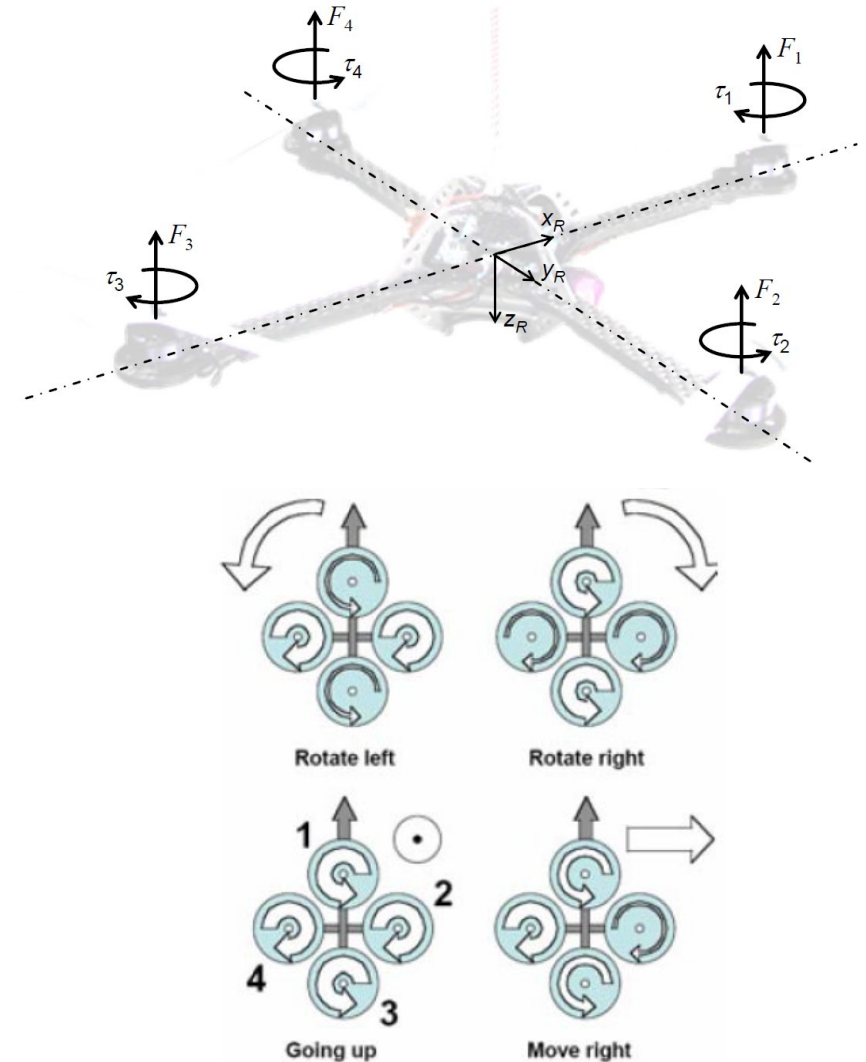
- Hover moments
 - Thrust induced moment
 - Drag torques

$${}_B \mathbf{M}_{Aero} = {}_B \mathbf{M}_T + {}_B \mathbf{Q} = \begin{bmatrix} l(T_4 - T_2) \\ l(T_1 - T_3) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^4 Q_i (-1)^{(i-1)} \end{bmatrix}$$

$$T_i = b_i \omega_{p,i}^2 \quad Q_i = d_i \omega_{p,i}^2$$

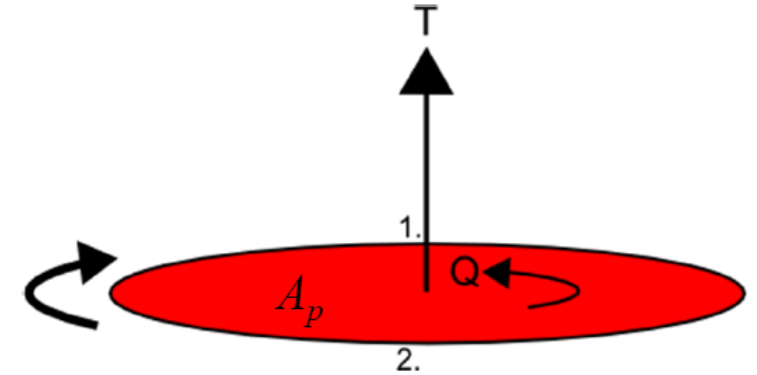
Modeling of a Rotorcraft

- Singularity issues (Tait-bryan angles)
- Analysis of Dynamics
 - Linearization of dynamics at hover
- Control
 - Virtual control input for decoupled attitude control



Introduction to propeller aerodynamics

- Thrust force
- Drag torque
- Momentum theory
 - Conservation of fluid mass, fluid momentum and energy
- BEMT (blade element momentum theory)



Fixed wing aerodynamics

- Lift force and drag force
 - Polars of airfoils
- Kinematics and dynamics of an aircraft
 - Forces and moments acting on airplane
 - Lift $L = \frac{1}{2}\rho V^2 S c_L$
 - Drag $D = \frac{1}{2}\rho V^2 S c_D$
 - Thrust
 - Gravity
- Cascaded control of FWs
 - Attitude controller, body rate controller

