ETH zürich

Lecture «Robot Dynamics»: Floating-base Dynamics

151-0851-00 V

lecture:	HG F3	Tuesday 10:15 – 12:00, every week
exercise:	HG D7.1	Wednesday 8:15 – 10:00, according to schedule

Marco Hutter, Roland Siegwart, and Thomas Stastny

ETH zürich

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			

Recapitulation of Introduction to Dynamics

- Description of "cause of motion"
  - Input $\boldsymbol{\tau}$ Force/Torque acting on system
  - Output $\ddot{\mathbf{q}}$ Motion of the system
- 3 methods to get the EoM
  - Newton-Euler: Free cut and conservation of impulse & angular mome
  - Projected Newton-Euler (generalized coordinates)
  - Lagrange II (energy)
- External forces
- Task Space Dynamics
- Inverse-dynamics-based control methods
  - Inverse Dynamics
  - OSC: Inverse Task-Space Dynamics
  - Quadratic Optimization

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

$\mathbf{q}$	Generalized coordinates
$\mathbf{M}(\mathbf{q})$	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

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{b}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} \longleftrightarrow \boldsymbol{\Lambda}\dot{\mathbf{w}}_e + \boldsymbol{\mu} + \mathbf{p} = \mathbf{F}_e$$

Recap: Inverse Dynamics, OSC, QP optimization

3 variants

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{b}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} \longleftrightarrow \boldsymbol{\Lambda}\dot{\mathbf{w}}_e + \boldsymbol{\mu} + \mathbf{p} = \mathbf{F}_e$$

- Different «inverse dynamics»-based methods

1. Classic ID: $\boldsymbol{\tau}^* = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}}^* + \mathbf{b}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{g}$ $\quad \dot{\mathbf{w}} = \mathbf{J}\ddot{\mathbf{q}} + \dot{\mathbf{J}}\dot{\mathbf{q}} \longrightarrow \ddot{\mathbf{q}}^* = \mathbf{J}^+\left(\dot{\mathbf{w}} - \dot{\mathbf{J}}\dot{\mathbf{q}}\right) \quad \min\|\ddot{\mathbf{q}}\|$

2. OSC $\boldsymbol{\tau}^* = \mathbf{J}_t^T\left(\boldsymbol{\Lambda}\dot{\mathbf{w}}_t^* + \boldsymbol{\mu} + \mathbf{p}\right) + N\left(\mathbf{J}_t^T\right)\boldsymbol{\tau}_0$ $\quad N\left(\mathbf{J}_t^T\right) = \left(\mathbf{I} - \mathbf{J}^T\left(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T\right)^{-1}\mathbf{J}\mathbf{M}^{-1}\right)$

$$\boldsymbol{\tau}^* = \mathbf{J}_t^T\left(\left(\mathbf{J}_t\mathbf{M}^{-1}\mathbf{J}_t^T\right)^{-1}\dot{\mathbf{w}} + \boldsymbol{\mu} + \mathbf{p}\right)$$

$$\ddot{\mathbf{q}} = \mathbf{M}^{-1}\left(\boldsymbol{\tau}^* - \mathbf{b} - \mathbf{g}\right) = \boxed{\mathbf{M}^{-1}\left(\mathbf{J}_t^T\left(\left(\mathbf{J}_t\mathbf{M}^{-1}\mathbf{J}_t^T\right)^{-1}\right.\right.}\left.\left.\dot{\mathbf{w}} + \boldsymbol{\mu} + \mathbf{p}\right) - \mathbf{b} - \mathbf{g}\right)$$

"some sort of mass-matrix weighted pseudo-inverse"

If the torque is applied in this null-space, there is no acceleration at the end-effector

3. Quadratic optimization

$$\boldsymbol{\tau} = \mathbf{M}\ddot{\mathbf{q}} + \mathbf{b} + \mathbf{g}$$
$$\mathbf{J}\ddot{\mathbf{q}} + \dot{\mathbf{J}}\dot{\mathbf{q}} = \dot{\mathbf{w}}$$

$\min\|\ddot{\mathbf{q}}\|$ or $\min\|\boldsymbol{\tau}\|$

$$\left.\begin{aligned} &\begin{bmatrix}\mathbf{M} & -\mathbf{I}\end{bmatrix}\begin{pmatrix}\ddot{\mathbf{q}}\\ \boldsymbol{\tau}\end{pmatrix} + \mathbf{b} + \mathbf{g} = \mathbf{0} \\ &\begin{bmatrix}\mathbf{J}_e & \mathbf{0}\end{bmatrix}\begin{pmatrix}\ddot{\mathbf{q}}\\ \boldsymbol{\tau}\end{pmatrix} + \dot{\mathbf{J}}\dot{\mathbf{q}} = \dot{\mathbf{w}}_e^*\end{aligned}\right\}$$

Single task $\min\limits_{\ddot{\mathbf{q}},\boldsymbol{\tau}}\left\|\begin{bmatrix}\mathbf{M} & -\mathbf{I}\\ \mathbf{J}_e & \mathbf{0}\end{bmatrix}\begin{pmatrix}\ddot{\mathbf{q}}\\ \boldsymbol{\tau}\end{pmatrix} - \begin{pmatrix}-\mathbf{b}-\mathbf{g}\\ \dot{\mathbf{w}}_e^* - \dot{\mathbf{J}}\dot{\mathbf{q}}\end{pmatrix}\right\|_2$

Priority $\begin{cases}\min\limits_{\ddot{\mathbf{q}},\boldsymbol{\tau}}\left\|\begin{bmatrix}\mathbf{J}_e & \mathbf{0}\end{bmatrix}\begin{pmatrix}\ddot{\mathbf{q}}\\ \boldsymbol{\tau}\end{pmatrix} - \left(\dot{\mathbf{w}}_e^* - \dot{\mathbf{J}}\dot{\mathbf{q}}\right)\right\|_2 \\ s.t. \begin{bmatrix}\mathbf{M} & -\mathbf{I}\end{bmatrix}\begin{pmatrix}\ddot{\mathbf{q}}\\ \boldsymbol{\tau}\end{pmatrix} - \left(-\mathbf{b}-\mathbf{g}\right) = \mathbf{0}\end{cases}$

Dynamics of Floating Base Systems

- Today: Floating base systems

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

$\mathbf{q}$	Generalized coordinates
$\mathbf{u}$	Generalized velocities
$\dot{\mathbf{u}}$	Generalized accelerations
$\mathbf{M}(\mathbf{q})$	Mass matrix
$\mathbf{b}(\mathbf{q},\mathbf{u})$	Centrifugal and Coriolis forces
$\mathbf{g}(\mathbf{q})$	Gravity forces
$\boldsymbol{\tau}$	Generalized forces
$\mathbf{S}_\tau$	Selection matrix/Jacobian
$\mathbf{F}_c$	External forces
$\mathbf{J}_c$	Contact Jacobian

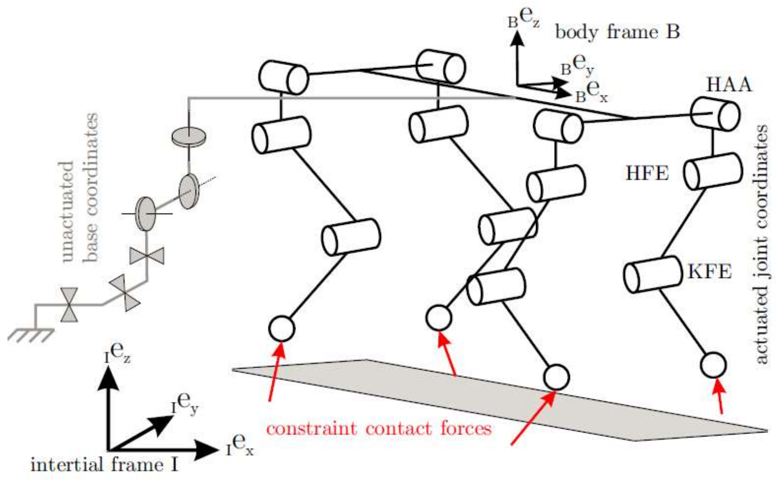
$$\mathbf{u}_j=\mathbf{S}\mathbf{u}=\mathbf{S}\begin{pmatrix}\mathbf{u}_b\\ \mathbf{u}_j\end{pmatrix}=\begin{bmatrix}0_{6\times6} & \mathbb{I}_{6\times n_j}\end{bmatrix}\begin{pmatrix}\mathbf{u}_b\\ \mathbf{u}_j\end{pmatrix}$$

Floating Base Systems
Kinematics

- Generalized coordinates

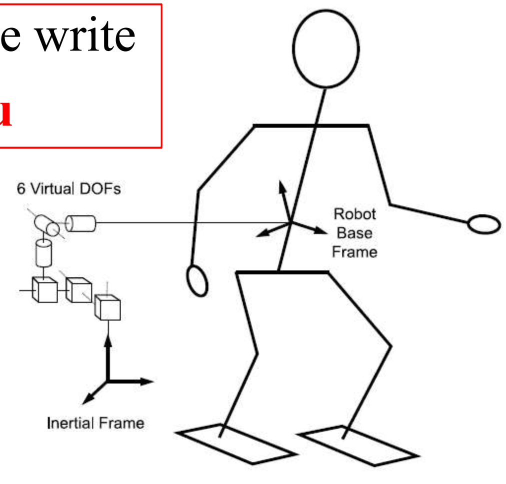
$$\mathbf{q} = \begin{pmatrix} \mathbf{q}_b \\ \mathbf{q}_j \end{pmatrix} \quad \text{with} \quad \mathbf{q}_b = \begin{pmatrix} \mathbf{q}_{b_P} \\ \mathbf{q}_{b_R} \end{pmatrix} \in \mathbb{R}^3 \times SO(3)$$

- Generalized velocities and accelerations?
  - Time derivatives $\dot{\mathbf{q}}, \ddot{\mathbf{q}}$ depend on parameterization
  - Often $\mathbf{u} = \begin{pmatrix} {}_I\mathbf{v}_B \\ {}_{\boxed{B}}\boldsymbol{\omega}_{IB} \\ \dot{\varphi}_1 \\ \vdots \\ \dot{\varphi}_{n_j} \end{pmatrix} \in \mathbb{R}^{6+n_j} = \mathbb{R}^{n_u}$ $\quad \dot{\mathbf{u}} = \begin{pmatrix} {}_I\mathbf{a}_B \\ {}_{\boxed{B}}\boldsymbol{\psi}_{IB} \\ \ddot{\varphi}_1 \\ \vdots \\ \ddot{\varphi}_{n_j} \end{pmatrix} \in \mathbb{R}^{6+n_j}$
  - Linear mapping $\mathbf{u} = \mathbf{E}_{fb} \cdot \dot{\mathbf{q}}$, with $\mathbf{E}_{fb} = \begin{bmatrix} \mathbb{I}_{3\times 3} & 0 & 0 \\ 0 & \mathbf{E}_{\chi_R} & 0 \\ 0 & 0 & \mathbb{I}_{n_j \times n_j} \end{bmatrix}$



(a) Quadruped

Very often, people write $\dot{\mathbf{q}}$ but they mean $\mathbf{u}$



(b) Humanoid

Floating Base Systems
Differential kinematics

- Position of an arbitrary point on the robot

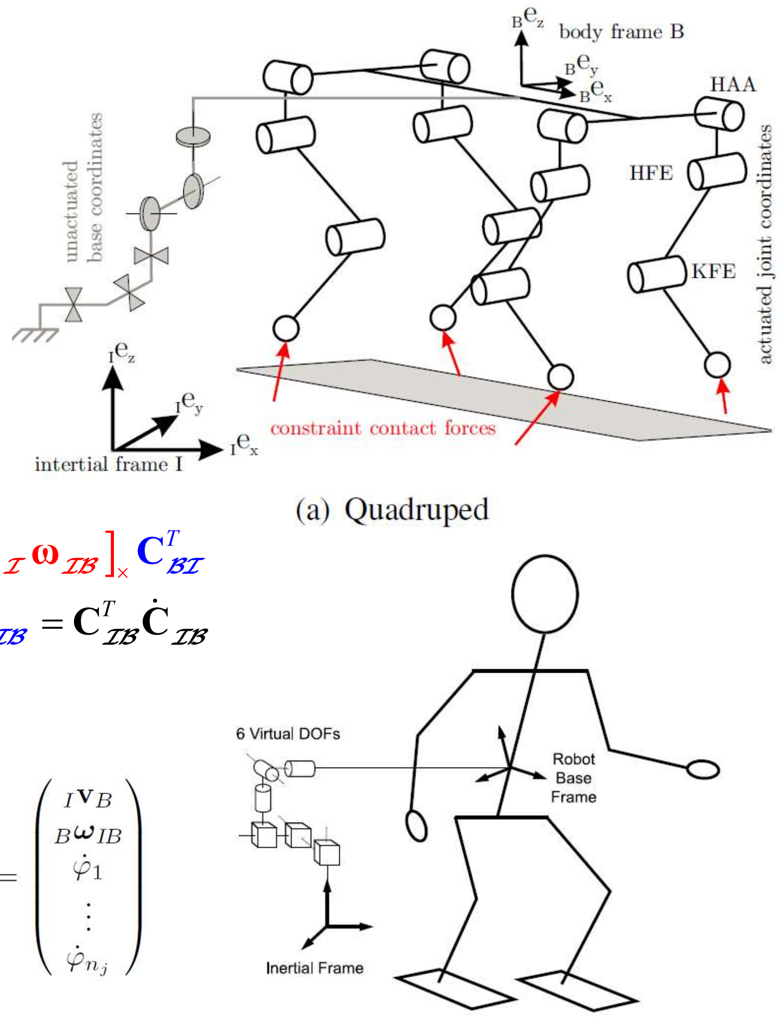
$$
{}_{\mathcal{I}}\mathbf{r}_{IQ}(\mathbf{q}) = \underbrace{\boxed{{}_{\mathcal{I}}\mathbf{r}_{IB}(\mathbf{q})}}_{{}_{\mathcal{I}}\mathbf{r}_{IB}(\mathbf{q}_b)} + \underbrace{\boxed{\mathbf{C}_{\mathcal{IB}}(\mathbf{q})}}_{\mathbf{C}_{\mathcal{IB}}(\mathbf{q}_b)} \cdot \underbrace{\boxed{{}_{\mathcal{B}}\mathbf{r}_{BQ}(\mathbf{q})}}_{{}_{\mathcal{B}}\mathbf{r}_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

- Velocity of this point

$$
\begin{aligned}
{}_{\mathcal{I}}\mathbf{v}_Q &= {}_{\mathcal{I}}\mathbf{v}_B + \dot{\mathbf{C}}_{\mathcal{IB}} \cdot {}_{\mathcal{B}}\mathbf{r}_{BQ} + \mathbf{C}_{\mathcal{IB}} \cdot {}_{\mathcal{B}}\dot{\mathbf{r}}_{BQ} \\
&= {}_{\mathcal{I}}\mathbf{v}_B + \mathbf{C}_{\mathcal{IB}} \cdot [{}_{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{IB}}]_\times \cdot {}_{\mathcal{B}}\mathbf{r}_{BQ} + \mathbf{C}_{\mathcal{IB}} \cdot {}_{\mathcal{B}}\dot{\mathbf{r}}_{BQ} \\
&= {}_{\mathcal{I}}\mathbf{v}_B - \mathbf{C}_{\mathcal{IB}} \cdot [{}_{\mathcal{B}}\mathbf{r}_{BQ}]_\times \cdot {}_{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{IB}} + \mathbf{C}_{\mathcal{IB}} \cdot {}_{\mathcal{B}}\dot{\mathbf{r}}_{BQ} \\
&= {}_{\mathcal{I}}\mathbf{v}_B - \mathbf{C}_{\mathcal{IB}} \cdot [{}_{\mathcal{B}}\mathbf{r}_{BQ}]_\times \cdot {}_{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{IB}} + \mathbf{C}_{\mathcal{IB}} \cdot {}_{\mathcal{B}}\mathbf{J}_{P_{q_j}}(\mathbf{q}_j) \cdot \dot{\mathbf{q}}_j \\
&= \underbrace{\boxed{\begin{bmatrix} \mathbb{I}_{3\times 3} & -\mathbf{C}_{\mathcal{IB}} \cdot [{}_{\mathcal{B}}\mathbf{r}_{BQ}]_\times & \mathbf{C}_{\mathcal{IB}} \cdot {}_{\mathcal{B}}\mathbf{J}_{P_{q_j}}(\mathbf{q}_j) \end{bmatrix}}}_{{}_{\mathcal{I}}\mathbf{J}_Q(\mathbf{q})} \cdot \mathbf{u}
\end{aligned}
$$

with $\mathbf{u} = \begin{pmatrix} {}_{I}\mathbf{v}_B \\ {}_{B}\boldsymbol{\omega}_{IB} \\ \dot{\varphi}_1 \\ \vdots \\ \dot{\varphi}_{n_j} \end{pmatrix}$

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



(a) Quadruped

(b) Humanoid

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

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

Last time: Null-space motion

- Remember:

$$\mathbf{0} = \dot{\mathbf{r}}_c = \mathbf{J}_c \dot{\mathbf{q}} \quad \longrightarrow \quad \dot{\mathbf{q}} = \mathbf{J}_c^+ \mathbf{0} + \mathbf{N}_c \dot{\mathbf{q}}_0 = \mathbf{N}_c \dot{\mathbf{q}}_0$$

$$\mathbf{0} = \ddot{\mathbf{r}}_c = \mathbf{J}_c \ddot{\mathbf{q}} + \dot{\mathbf{J}}_c \dot{\mathbf{q}} \quad \longrightarrow \quad \ddot{\mathbf{q}} = \mathbf{J}_c^+ \left(-\dot{\mathbf{J}}_c \dot{\mathbf{q}}\right) + \mathbf{N}_c \ddot{\mathbf{q}}_0$$

- The system can be moved without violating the contact constraints!

- !! However, the base is unactuated !!
  - Which ones can be ACTIVELY controlled?

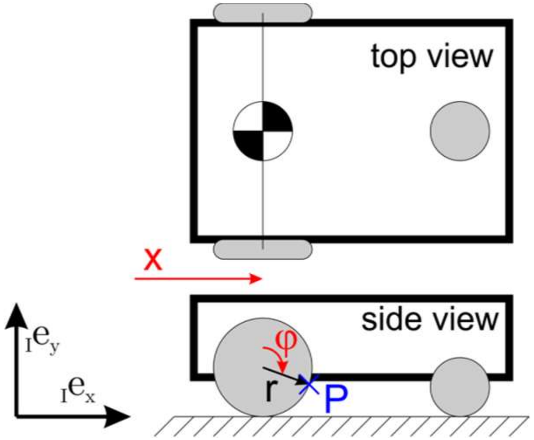
Contact Constraint
Wheeled vehicle simple example

5

5

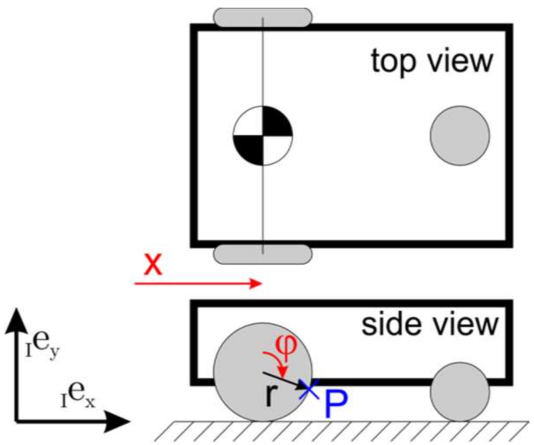
Quiz
2min
0min

- Contact constraints
  - Point on wheel
  - Jacobian
  - Contact constraints
  - Possible base motion (Nullspace)



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 $\dot{x} - r\dot{\varphi} = 0$

  $$\mathbf{q} = \begin{pmatrix} x \\ \varphi \end{pmatrix}$$ Un-actuated base / Actuated joints
- Possible base motion $$\dot{\mathbf{q}} = \mathbf{J}_c^{+}\mathbf{0} + \mathbf{N}_c\dot{\mathbf{q}}_0 = \mathbf{N}_c\dot{\mathbf{q}}_0 = \begin{bmatrix} r \\ 1 \end{bmatrix}\dot{q}_0$$

Properties of Contact Jacobian

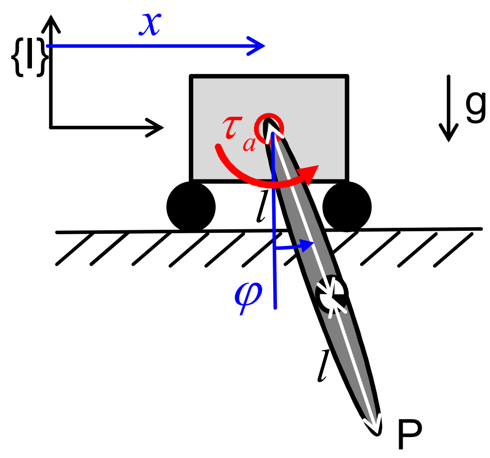
- Contact Jacobian tells us, how a system can move.
  - Separate stacked Jacobian $\mathbf{J}_c = \begin{bmatrix} \boxed{\mathbf{J}_{c,b}} & \mathbf{J}_{c,j} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{r}_c}{\partial \mathbf{q}_b} & \frac{\partial \mathbf{r}_c}{\partial \mathbf{q}_j} \end{bmatrix}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

    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)

Stupid, simple example
Cart pendulum

$$r_P = \begin{pmatrix} x + 2l\sin(\varphi) \\ -2l\cos(\varphi) \end{pmatrix}$$

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



- Analyse the kinematic constraints of this example
  - 1) P cannot move at all

$$\dot{\mathbf{r}}_P = \begin{pmatrix} \dot{x} + \dot{\varphi}2l\cos(\varphi) \\ \dot{\varphi}2l\sin(\varphi) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \Rightarrow \mathbf{J}_c = \begin{bmatrix} \mathbf{J}_{c,b} & \mathbf{J}_{c,j} \end{bmatrix} = \begin{bmatrix} 1 & 2l\cos(\varphi) \\ 0 & 2l\sin(\varphi) \end{bmatrix}$$

$$rank(\mathbf{J}_c) = 2$$
$$rank(\mathbf{J}_{c,b}) = 1$$

  - 2) P can only move horizontally

$$\dot{y}_P = \dot{\varphi}2l\sin(\varphi) = 0 \Rightarrow \mathbf{J}_c = \begin{bmatrix} \mathbf{J}_{c,b} & \mathbf{J}_{c,j} \end{bmatrix} = \begin{bmatrix} 0 & 2l\sin(\varphi) \end{bmatrix}$$

$$rank(\mathbf{J}_c) = 1$$
$$rank(\mathbf{J}_{c,b}) = 0$$

  - 3) P can only move vertically

$$\dot{x}_P = \dot{x} + \dot{\varphi}2l\cos(\varphi) = 0 \Rightarrow \mathbf{J}_c = \begin{bmatrix} \mathbf{J}_{c,b} & \mathbf{J}_{c,j} \end{bmatrix} = \begin{bmatrix} 1 & 2l\cos(\varphi) \end{bmatrix}$$

$$rank(\mathbf{J}_c) = 1$$
$$rank(\mathbf{J}_{c,b}) = 1$$

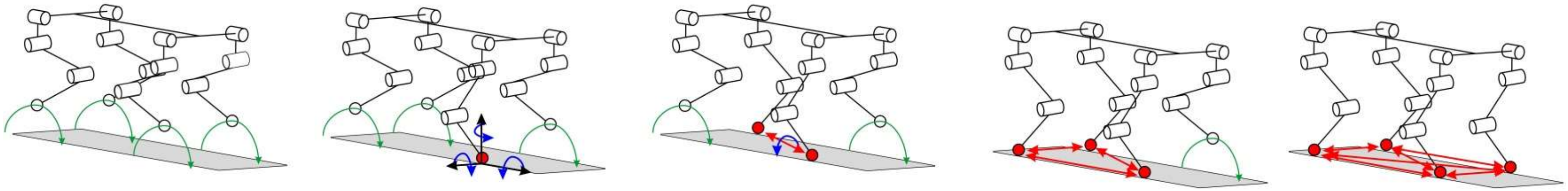
Quadrupedal Robot with Point Feet

5

5

Quiz
2min
1min

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



Total constraints
$rank\left(\mathbf{J}_c\right)$

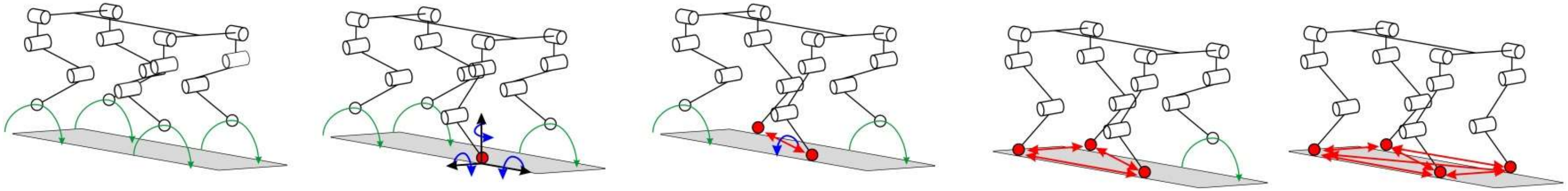
Base constraints
$rank\left(\mathbf{J}_{c,b}\right)$

Internal constraints
$rank\left(\mathbf{J}_c\right) - rank\left(\mathbf{J}_{c,b}\right)$

Uncontrollable DoFs
$6 - rank\left(\mathbf{J}_{c,b}\right)$

Quadrupedal Robot with Point Feet

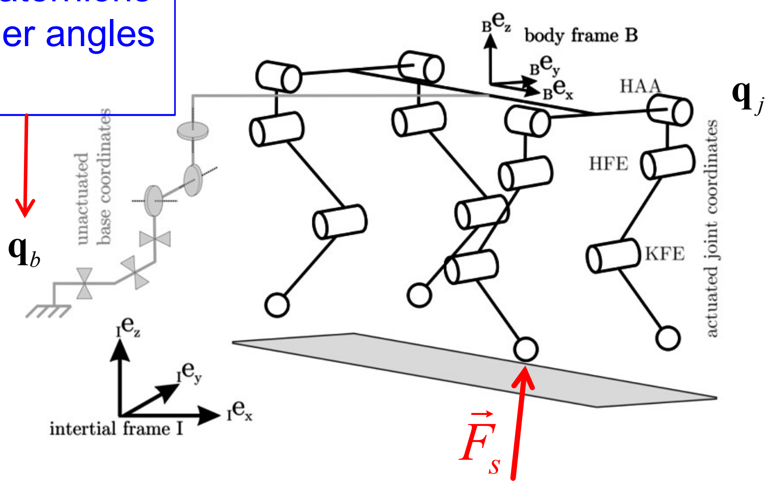
- Floating base system with 12 actuated joint and 6 base coordinates (18DoF)



Total constraints $rank(\mathbf{J}_c)$	0	3	6	9	12
Base constraints $rank(\mathbf{J}_{c,b})$	0	3	5	6	6
Internal constraints $rank(\mathbf{J}_c) - rank(\mathbf{J}_{c,b})$	0	0	1	3	6
Uncontrollable DoFs $6 - rank(\mathbf{J}_{c,b})$	6	3	1	0	0

Dynamics of Floating Base Systems

- Quaternions
- Euler angles
- …



$$\mathbf{q} = \begin{pmatrix} \boxed{\mathbf{q}_b} \\ \mathbf{q}_j \end{pmatrix}$$ Un-actuated base / Actuated joints

- 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

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

- Contact force acting on system

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

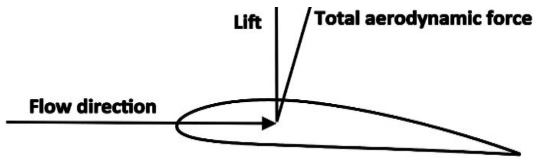
$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{b} + \mathbf{g} + \mathbf{J}_s^T \mathbf{F}_{s,\text{ exerted by robot}} = \mathbf{S}^T \boldsymbol{\tau}$$

Manipulator:	interaction forces at end-effector
Legged robot:	ground contact forces
UAV:	lift force

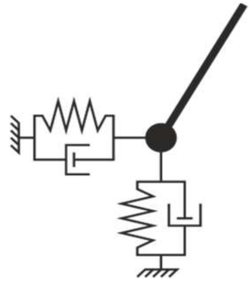
*Note: for simplicity we don't use here **u** but only time derivatives of **q***

External Forces
Some notes

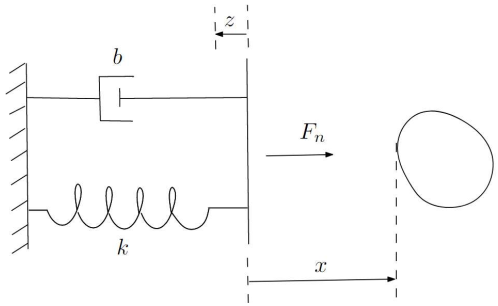
- External forces from force elements or actuator



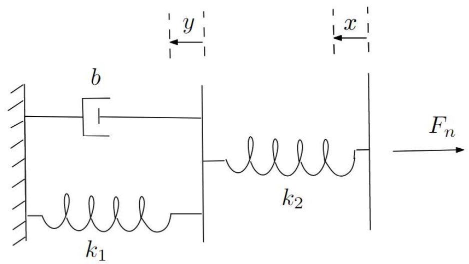
  - Aerodynamics $F_s = \frac{1}{2}\rho c_v A c_L$



  - Contact
    - Simple solution: soft contact model $\mathbf{F}_c = k_p(\mathbf{r}_c - \mathbf{r}_{c0}) + k_d\dot{\mathbf{r}}_c$
    - no negative force $F_n = \begin{cases} 0 & \text{if } x > z \\ max(0, kz + b\dot{z}) & \text{if } x = z \end{cases}$



Linear S-D 1

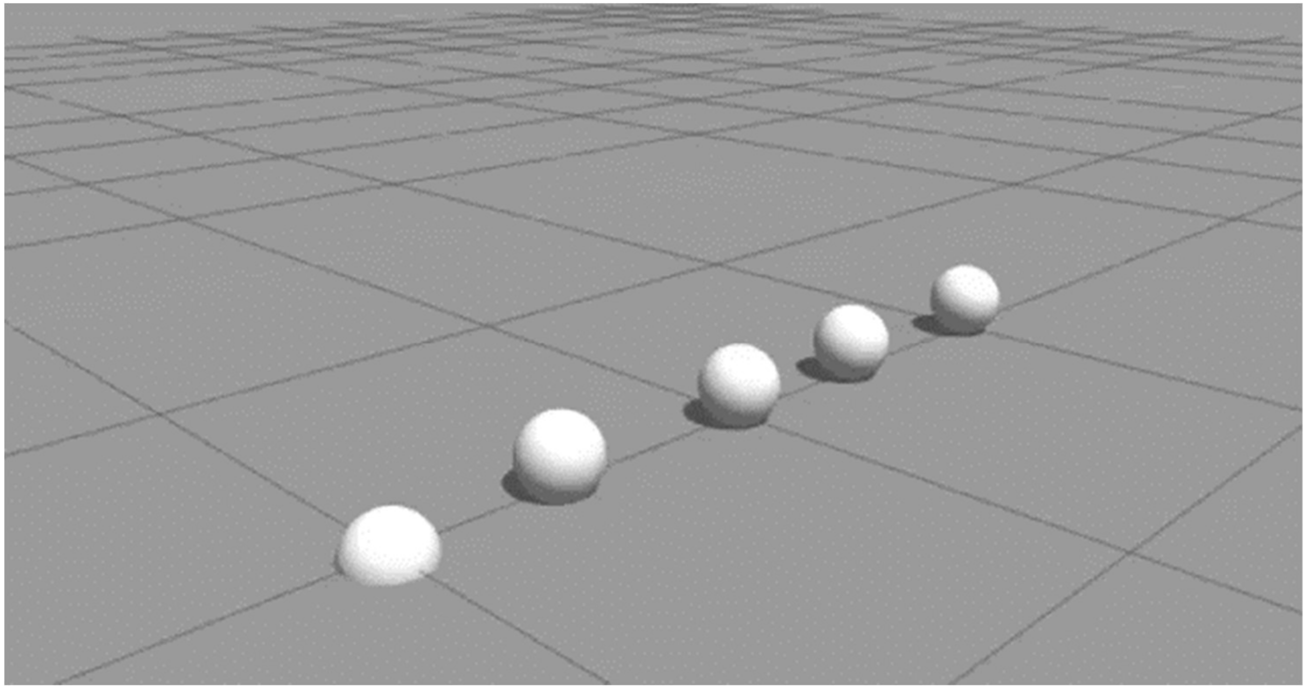


Linear S-D 2

$F_n = bx^n\dot{x} + kx^n$

Nonlinear

Soft Contact
Physical accuracy vs. numerical stability?

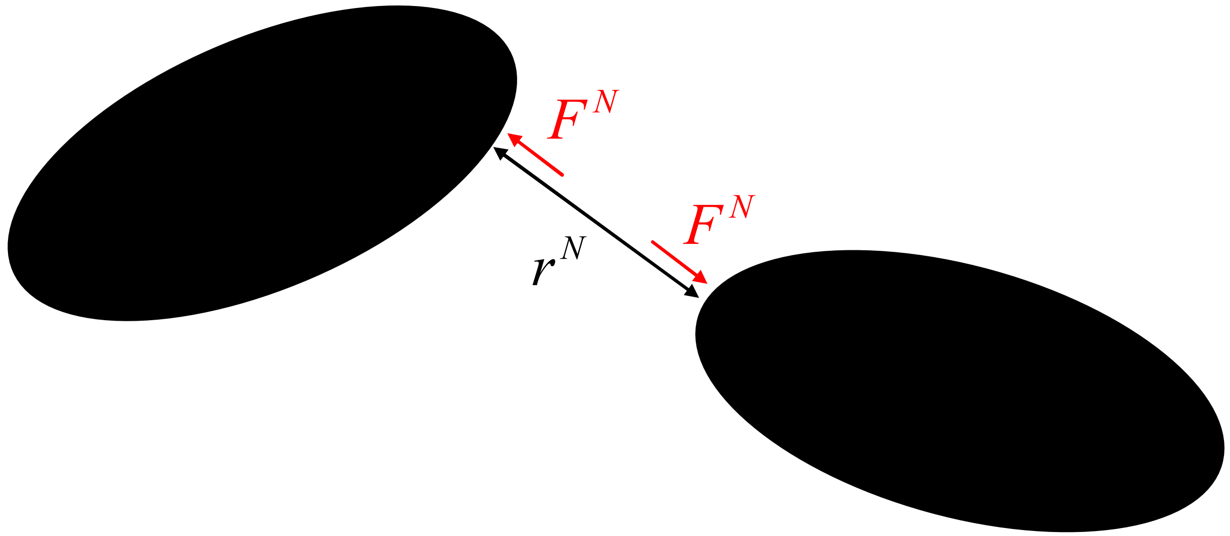


Soft Contact??

Soft Contact
Physical accuracy vs. numerical stability?

- Stiff equation of motion
  - Small time steps
  - Can lead to instability
- Contact behavior strongly depends on the robot parameters/configuration
- Contact parameters are not selected as physical parameters but as numerical
  - Trade-off stability ⇔ accuracy

Hard Contact



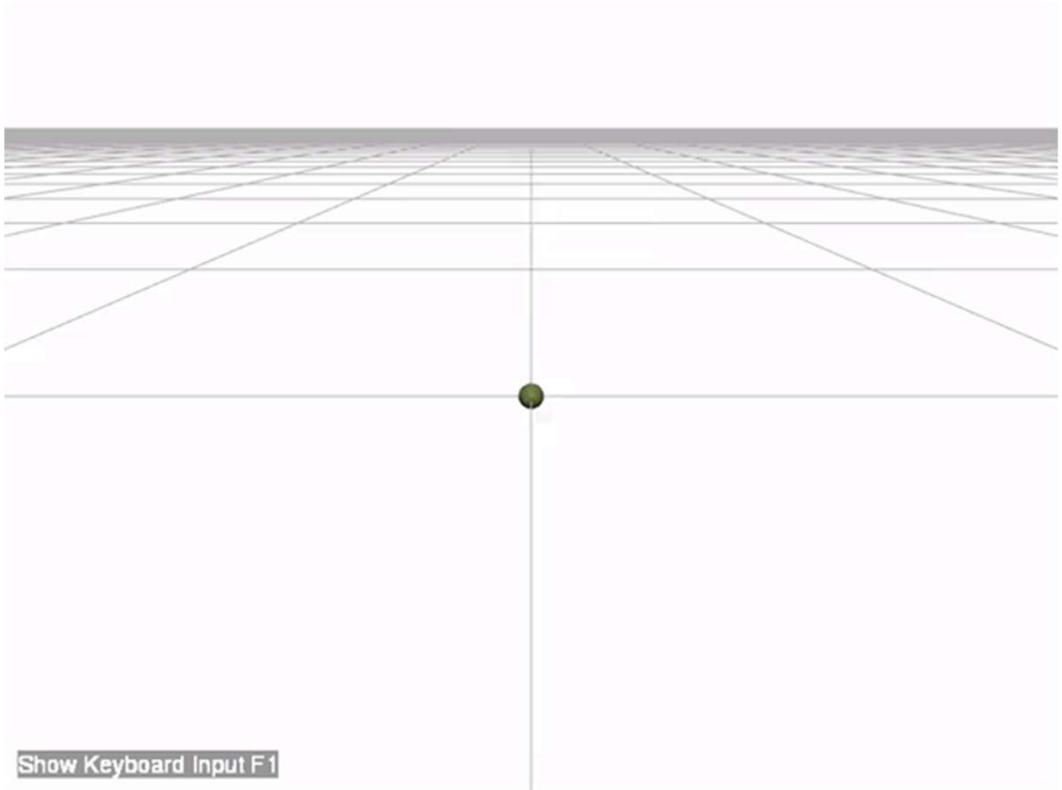
For closed contacts

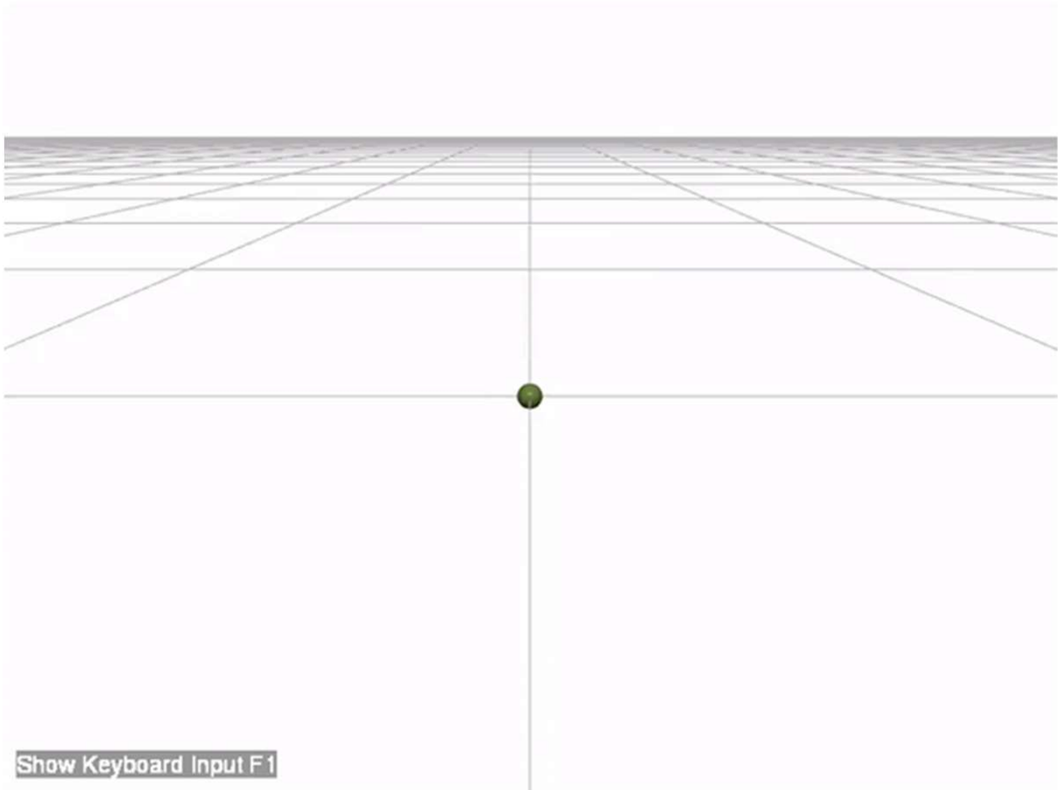
$$r^N = 0, \dot{r}^N = 0$$

Linear complemantary constraint (LCP)

$$\ddot{r}^N \geq 0 \quad F^N \geq 0 \quad \ddot{r}^N F^N = 0$$

Simulation of Hard Contacts





Hard Contact of an MBS

- External forces from constraints
  - Equation of motion $$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{b} + \mathbf{g} + \mathbf{J}_c^T \mathbf{F}_c = \mathbf{S}^T \boldsymbol{\tau} \quad (1)$$
  - Contact constraint $$\dot{\mathbf{r}}_c = \mathbf{J}_c \dot{\mathbf{q}} = \mathbf{0} \Longrightarrow \ddot{\mathbf{r}}_c = \mathbf{J}_c \ddot{\mathbf{q}} + \dot{\mathbf{J}}_c \dot{\mathbf{q}} = \mathbf{0} \quad (2)$$
  - Substitute $\ddot{q}$ in (2) from (1) $$\ddot{\mathbf{r}}_c = \mathbf{J}_c \mathbf{M}^{-1}\left(\mathbf{S}^T \boldsymbol{\tau} - \left(\mathbf{b} + \mathbf{g} + \mathbf{J}_c^T \mathbf{F}_c\right)\right) + \dot{\mathbf{J}}_c \dot{\mathbf{q}} = \mathbf{0} \quad (3)$$
  - Solve (3) for contact force $$\mathbf{F}_c = \left(\mathbf{J}_c \mathbf{M}^{-1} \mathbf{J}_c^T\right)^{-1} \left(\mathbf{J}_c \mathbf{M}^{-1}\left(\mathbf{S}^T \boldsymbol{\tau} - (\mathbf{b} + \mathbf{g})\right) + \dot{\mathbf{J}}_c \dot{\mathbf{q}}\right)$$
  - Back-substitute in (1), replace $\dot{\mathbf{J}}_s \dot{\mathbf{q}} = -\mathbf{J}_s \ddot{\mathbf{q}}$ and use support null-space projection $$\mathbf{N}_c = \mathbf{I} - \mathbf{M}^{-1} \mathbf{J}_c^T \left(\mathbf{J}_c \mathbf{M}^{-1} \mathbf{J}_c^T\right)^{-1} \mathbf{J}_c$$ $$\boxed{\mathbf{J}_c \mathbf{N}_c = \mathbf{0}}$$
- Support consistent dynamics $$\boxed{\mathbf{N}_c^T \mathbf{M}\ddot{\mathbf{q}} + \mathbf{N}_c^T (\mathbf{b} + \mathbf{g}) = \mathbf{N}_c^T \mathbf{S}^T \boldsymbol{\tau}}$$
  -

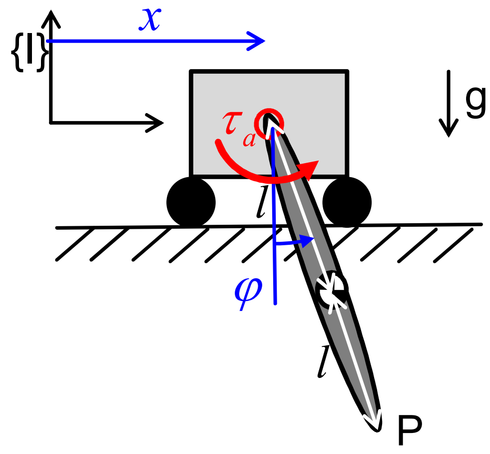
This is only a projector, … we can use other ones

Simple example
Cart Pendulum

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

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

$$\underbrace{\begin{bmatrix} m_c + m_p & l m_p \cos(\varphi) \\ l m_p \cos(\varphi) & m_p l^2 + \theta_p \end{bmatrix}}_{\mathbf{M}} \ddot{\mathbf{q}} + \underbrace{\begin{pmatrix} -\dot{\varphi}^2 l m_p \sin(\varphi) \\ 0 \end{pmatrix}}_{\mathbf{b}} + \underbrace{\begin{pmatrix} 0 \\ m_p g l \sin(\varphi) \end{pmatrix}}_{\mathbf{g}} = \underbrace{\begin{pmatrix} 0 \\ 1 \end{pmatrix}}_{\mathbf{S}^T} \tau_a$$



- Contact Jacobian: $\mathbf{J}_c = \begin{bmatrix} \mathbf{J}_{c,b} & \mathbf{J}_{c,j} \end{bmatrix} = \begin{bmatrix} 1 & 2l\cos(\varphi) \\ 0 & 2l\sin(\varphi) \end{bmatrix}$

1) both constraints active:

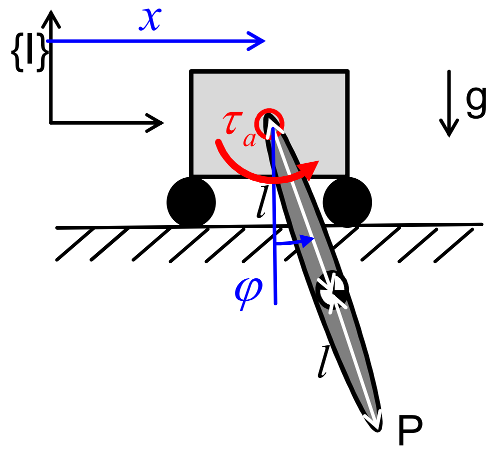
- No controllable subspace $\mathcal{N}(\mathbf{J}_c) = 0$
- => constraint define the «motion», τ_a can be freely choosen and only changes $\mathbf{F}_c$

Simple example
Cart Pendulum

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

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

$$\underbrace{\begin{bmatrix} m_c + m_p & l m_p \cos(\varphi) \\ l m_p \cos(\varphi) & m_p l^2 + \theta_p \end{bmatrix}}_{\mathbf{M}}\ddot{\mathbf{q}} + \underbrace{\begin{pmatrix} -\dot{\varphi}^2 l m_p \sin(\varphi) \\ 0 \end{pmatrix}}_{\mathbf{b}} + \underbrace{\begin{pmatrix} 0 \\ m_p g l \sin(\varphi) \end{pmatrix}}_{\mathbf{g}} = \underbrace{\begin{pmatrix} 0 \\ 1 \end{pmatrix}}_{\mathbf{S}^T}\tau_a$$



- Contact Jacobian: $\mathbf{J}_c = \begin{bmatrix} \mathbf{J}_{c,b} & \mathbf{J}_{c,j} \end{bmatrix} = \begin{bmatrix} \cancel{1} & \cancel{2l\cos(\varphi)} \\ 0 & 2l\sin(\varphi) \end{bmatrix}$
- 2) vertical motion locked: $\mathbf{N}_c = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ $\quad \mathbf{N}_c^T\mathbf{S}^T = \begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0$
  - first line corresponds to support consistent dynamics
  - Torque has no influence on motion
  - Torque can be used to modify the constraint force $\mathcal{N}\left(\mathbf{N}_c^T\mathbf{S}^T\right) = \mathcal{N}(0) = 1$

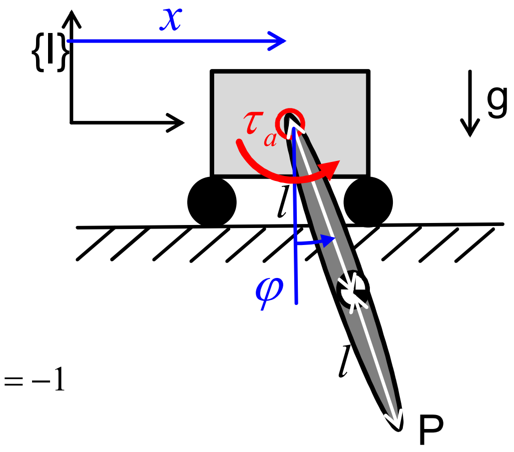
ETH zürich

Simple example
Cart Pendulum

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

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

$$\underbrace{\begin{bmatrix} m_c + m_p & l m_p \cos(\varphi) \\ l m_p \cos(\varphi) & m_p l^2 + \theta_p \end{bmatrix}}_{\mathbf{M}} \ddot{\mathbf{q}} + \underbrace{\begin{pmatrix} -\dot{\varphi}^2 l m_p \sin(\varphi) \\ 0 \end{pmatrix}}_{\mathbf{b}} + \underbrace{\begin{pmatrix} 0 \\ m_p g l \sin(\varphi) \end{pmatrix}}_{\mathbf{g}} = \underbrace{\begin{pmatrix} 0 \\ 1 \end{pmatrix}}_{\mathbf{S}^T} \tau_a$$



- Contact Jacobian: $\mathbf{J}_c = \begin{bmatrix} \mathbf{J}_{c,b} & \mathbf{J}_{c,j} \end{bmatrix} = \begin{bmatrix} 1 & 2l\cos(\varphi) \\ \cancel{0} & \cancel{2l\sin(\varphi)} \end{bmatrix}$
- 2) horizontal motion locked: $\mathbf{N}_c = \begin{bmatrix} 2l\cos(\varphi) \\ -1 \end{bmatrix}$ $\quad \mathbf{N}_c^T \mathbf{S}^T = \begin{bmatrix} 2l\cos(\varphi) & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = -1$
  -
  - first line corresponds to support consistent dynamics
  - There is no null-space that can change the constraint force without changing the motion

Small Excursion: Contact Dynamics

- Impulse transfer at contact
  - Integration over a single point in time

$$\int_{\{t_0\}} \left(\mathbf{M}\dot{\mathbf{u}} + \mathbf{b} + \mathbf{g} + \mathbf{J}_c^T \mathbf{F}_c - \mathbf{S}^T \boldsymbol{\tau}\right) dt = \mathbf{M}\left(\mathbf{u}^+ - \mathbf{u}^-\right) + \mathbf{J}_c^T \mathcal{F}_c = \mathbf{0}$$

  - Post impact condition

$$\dot{\mathbf{r}}_c^+ = \mathbf{J}_c \mathbf{u}^+ = \mathbf{0}$$

  - Impulsive force

$$\mathcal{F}_c = \left(\mathbf{J}_c \mathbf{M}^{-1} \mathbf{J}_c^T\right)^{-1} \mathbf{J}_c \dot{\mathbf{q}}^- = \boldsymbol{\Lambda}_c \dot{\mathbf{r}}_c^-$$

    - End-effector inertia

$$\boldsymbol{\Lambda}_c = \left(\mathbf{J}_c \mathbf{M}^{-1} \mathbf{J}_c^T\right)^{-1}$$

    - Change in generalized velocity

$$\Delta \mathbf{u} = \mathbf{u}^+ - \mathbf{u}^- = -\mathbf{M}^{-1} \mathbf{J}_c^T \left(\mathbf{J}_c \mathbf{M}^{-1} \mathbf{J}_c^T\right)^{-1} \mathbf{J}_c \mathbf{u}^-$$

    - Post-impact velocity

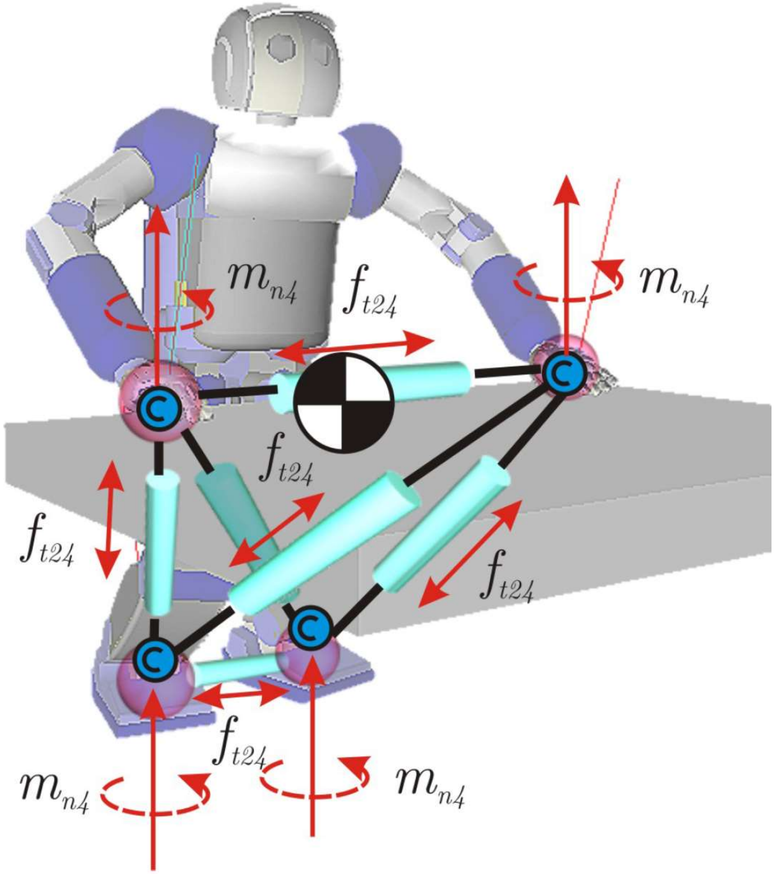
$$\mathbf{u}^+ = \left(\mathbb{I} - \mathbf{M}^{-1} \mathbf{J}_c^T \left(\mathbf{J}_c \mathbf{M}^{-1} \mathbf{J}_c^T\right)^{-1} \mathbf{J}_c\right) \mathbf{u}^- = \mathbf{N}_c \mathbf{u}^-$$

    - Energy loss

$$\begin{aligned} E_{loss} = \Delta E_{kin} &= -\tfrac{1}{2} \Delta \mathbf{u}^T \mathbf{M} \Delta \mathbf{u} \\ &= -\tfrac{1}{2} \Delta \dot{\mathbf{r}}_c^T \boldsymbol{\Lambda}_c \Delta \dot{\mathbf{r}}_c = -\tfrac{1}{2} \dot{\mathbf{r}}_c^{-T} \boldsymbol{\Lambda}_c \dot{\mathbf{r}}_c^- \end{aligned}$$

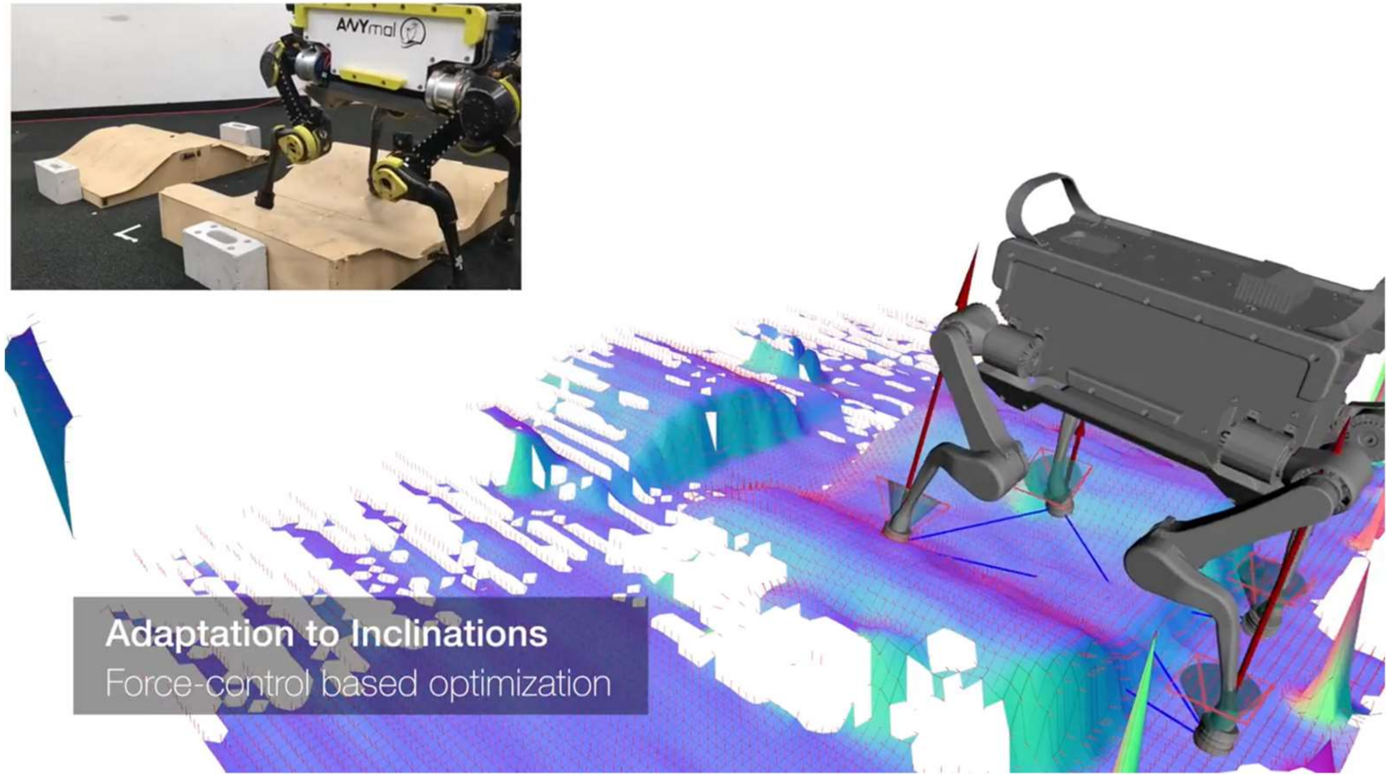
Application to Floating base Robots

Some Examples of Using Internal Forces

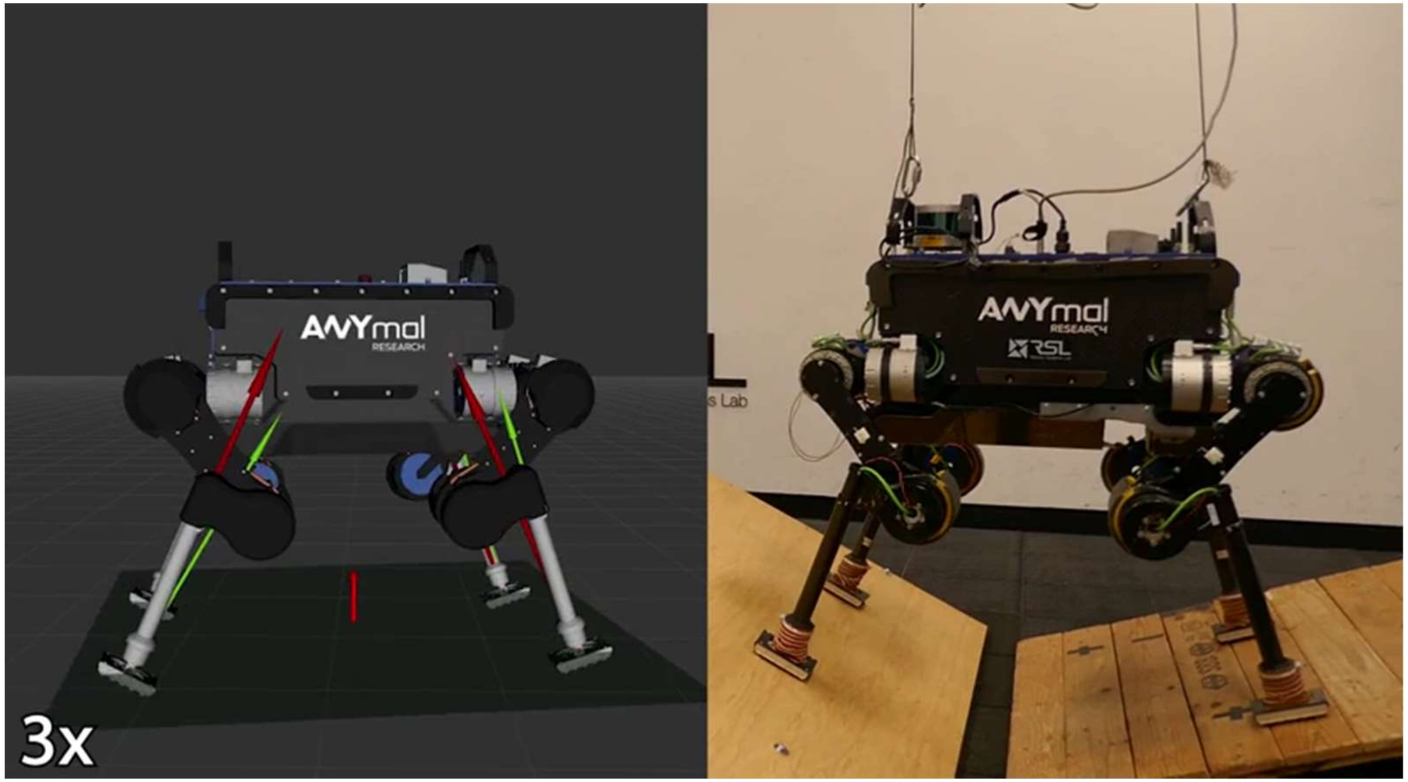


Internal Forces
extreme example

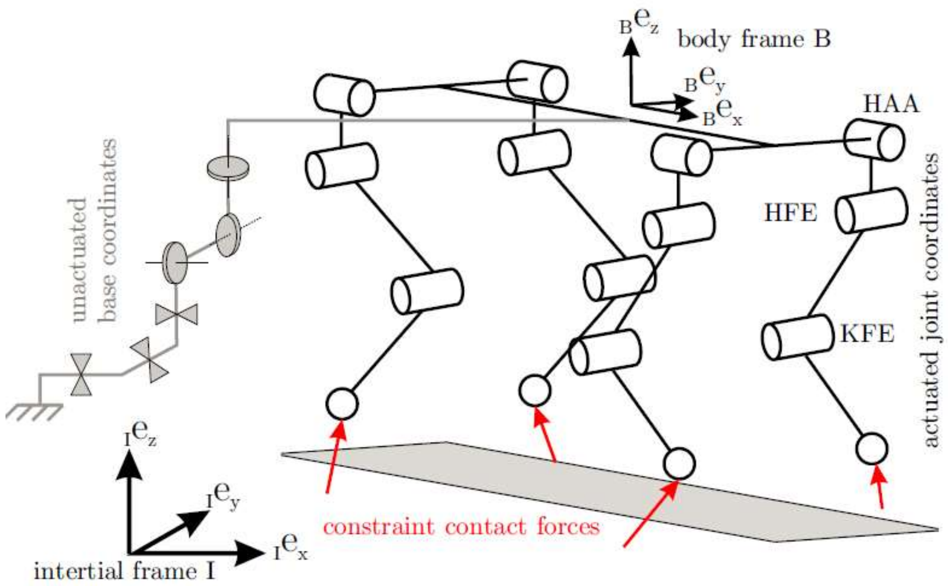
Internal Forces
a little less extreme



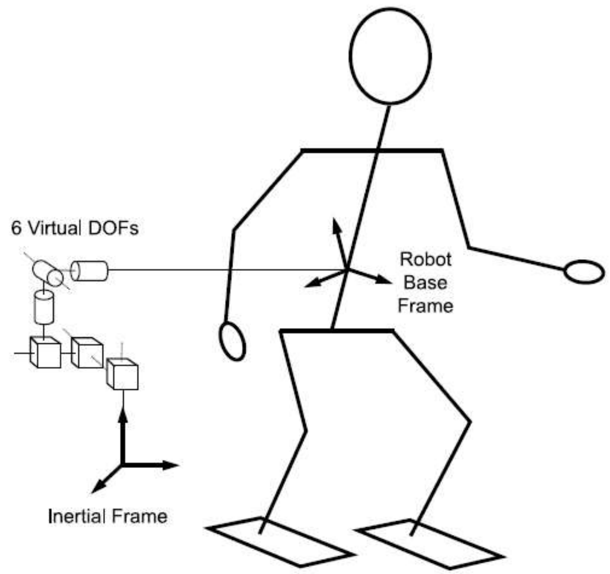
Internal Forces
local force adaptation



Inverse Dynamics of Floating Base Systems



(a) Quadruped



(b) Humanoid

Recapitulation: Support Consistent Dynamics

- Equation of motion $$\mathbf{M}(\mathbf{q})\dot{\mathbf{u}} + \mathbf{b}(\mathbf{q},\mathbf{u}) + \mathbf{g}(\mathbf{q}) + \mathbf{J}_c^T \mathbf{F}_c = \mathbf{S}^T \boldsymbol{\tau}$$
  - Cannot directly be used for control due to the occurrence of contact forces
- Contact constraint $$\ddot{\mathbf{r}}_c = \mathbf{J}_c \dot{\mathbf{u}} + \dot{\mathbf{J}}_c \mathbf{u} = 0$$
- Contact force $$\mathbf{F}_c = \left(\mathbf{J}_c \mathbf{M}^{-1} \mathbf{J}_c^T\right)^{-1} \left(\mathbf{J}_c \mathbf{M}^{-1}\left(\mathbf{S}^T \boldsymbol{\tau} - \mathbf{b} - \mathbf{g}\right) + \dot{\mathbf{J}}_c \mathbf{u}\right)$$
  - Back-substitute in (1), replace $\dot{\mathbf{J}}_s \dot{\mathbf{q}} = -\mathbf{J}_s \ddot{\mathbf{q}}$ and use support null-space projection $$\mathbf{N}_c = \mathbb{I} - \mathbf{M}^{-1} \mathbf{J}_c^T \left(\mathbf{J}_c \mathbf{M}^{-1} \mathbf{J}_c^T\right)^{-1} \mathbf{J}_c$$
- Support consistent dynamics $$\mathbf{N}_c^T \left(\mathbf{M}\ddot{\mathbf{q}} + \mathbf{b} + \mathbf{g}\right) = \mathbf{N}_c^T \mathbf{S}^T \boldsymbol{\tau}$$
- Inverse-dynamics $$\boldsymbol{\tau}^* = \left(\mathbf{N}_c^T \mathbf{S}^T\right)^+ \mathbf{N}_c^T \left(\mathbf{M}\ddot{\mathbf{q}} + \mathbf{b} + \mathbf{g}\right)$$
- Multiple solutions $$\boldsymbol{\tau}^* = \left(\mathbf{N}_c^T \mathbf{S}^T\right)^+ \mathbf{N}_c^T \left(\mathbf{M}\ddot{\mathbf{q}}^* + \mathbf{b} + \mathbf{g}\right) + \mathcal{N}\left(\mathbf{N}_c^T \mathbf{S}^T\right) \boldsymbol{\tau}_0^*$$

5

5

Quiz
2min
1min

Task Space Control as Quadratic Program

A general problem

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

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

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

- Motion tasks: $\mathbf{J}\dot{\mathbf{u}} + \dot{\mathbf{J}}\mathbf{u} = \dot{\mathbf{w}}^* \quad \Rightarrow$
- Force tasks: $\mathbf{F}_i = \mathbf{F}_i^* \quad \Rightarrow$
- Torque min: $\min \left\|\boldsymbol{\tau}\right\|_2 \quad \Rightarrow$

Tasl 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

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

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

Behavior as Multiple Tasks