Karlsruhe Institute of Technology

High Performance Humanoid Technologies

Robotics I: Introduction to Robotics
Kapitel 4 – Dynamics

Tamim Asfour

https://h2t.iar.kit.edu/

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

$$\tau_i = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i}$$

Models in Robotics – Outline

- **Kinematic Models**
  Kinematics studies of motion of bodies and systems based **only on geometry**, i.e. without considering the physical properties and the forces acting on them. The essential concept is a **pose** (position and orientation).

- **Dynamic Models**
  Dynamics studies the relationship between the **forces and moments** acting on a robot and accelerations they produce

- **Geometric Models**
  Geometry: Mathematical description of the **shape of bodies**

Contents

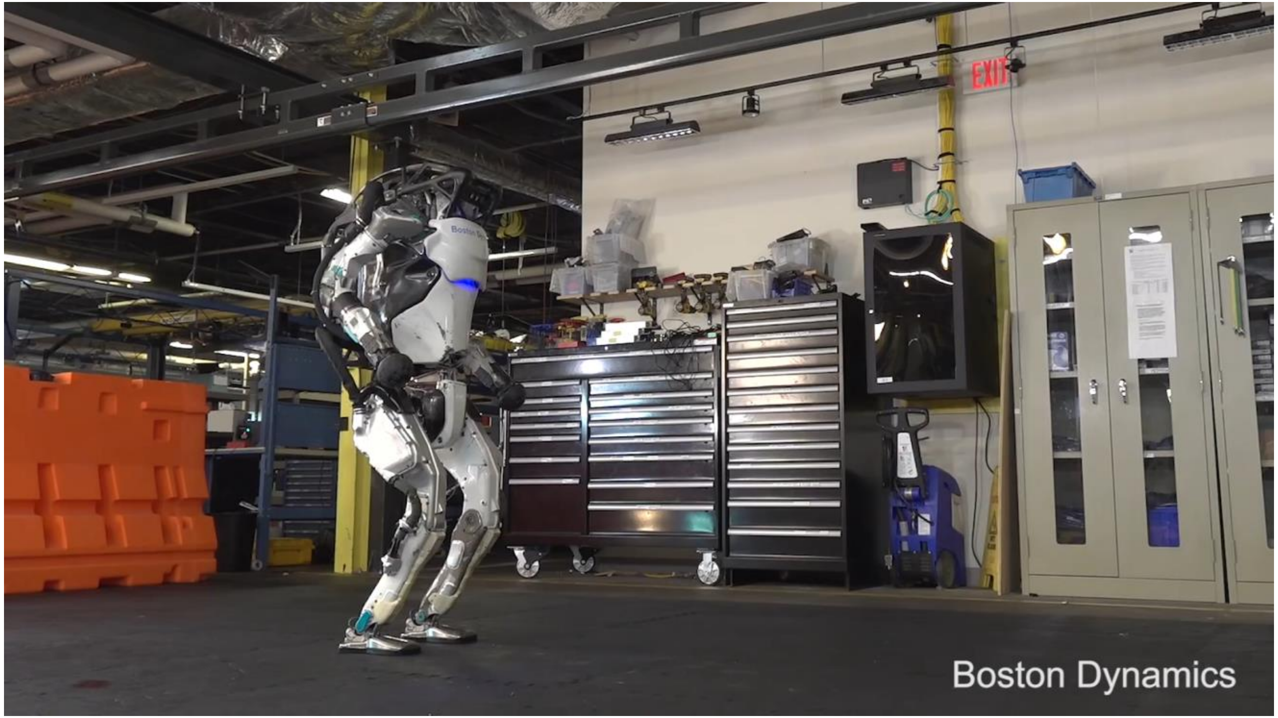
Dynamic Model: Definition & Goal

- Definition:

> The **dynamic model** describes the relationship between the **actuator and contact forces and moments** acting on a robot and accelerations and motions they produce

- Goal:
  - Analysis of the dynamics
  - Design und synthesis of mechanical structures
  - Controller design and control (→ Inverse Dynamics)
  - Modeling and simulation (→ Forward Dynamics)

Motivation



Boston Dynamics: https://www.youtube.com/watch?v=_sBBaNYex3E

Dynamic Model: Equation of Motion

- General equation of motion

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

$\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}$:	$n \times 1$	vector of generalized coordinates (position, velocity and acceleration)
$\boldsymbol{\tau}$:	$n \times 1$	vector of generalized forces
$M(\boldsymbol{q})$:	$n \times n$	matrix of mass inertia (symmetric, positive-definite)
$C(\dot{\boldsymbol{q}},\boldsymbol{q})\dot{\boldsymbol{q}}$:	$n \times 1$	vector with centripetal and Coriolis components
$g(\boldsymbol{q})$:	$n \times 1$	vector of gravitational components
$\epsilon(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})$:	$n \times 1$	non-linear effects, e.g. friction (often neglected)

n: degrees of freedom of the robot

Dynamic Model: Equation of Motion

- General equation of motion

$$\boxed{\boldsymbol{\tau} = M(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + g(\boldsymbol{q}) + \epsilon(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})}$$

$\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}$:	$n \times 1$	vector of **generalized coordinates** (**position, velocity and acceleration**)
$\boldsymbol{\tau}$:	$n \times 1$	vector of **generalized forces**
$M(\boldsymbol{q})$:	$n \times n$	matrix of mass inertia (symmetric, positive-definite)
$C(\dot{\boldsymbol{q}}, \boldsymbol{q})\dot{\boldsymbol{q}}$:	$n \times 1$	vector with centripetal and Coriolis components
$g(\boldsymbol{q})$:	$n \times 1$	vector of gravitational components
$\epsilon(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$:	$n \times 1$	non-linear effects, e.g. friction (often neglected)

What are generalized coordinates?

n: degrees of freedom of the robot

Contents

Generalized Coordinates (1)

- Definition

> Minimum set of independent coordinates that completely describe the system state.

- General Model
  - A robot consists of N particles with mass m_i and coordinate $\boldsymbol{x_i}$
  - For each position vector of a particle 3 spatial coordinates are needed, in total $\boldsymbol{3N}$ **coordinates**, to describe the system
  - Newton's second law: $\boldsymbol{F}_i = m_i \cdot \ddot{\boldsymbol{x}}_i$ with $i = 1, \dots, N$
  - Particles cannot move independently of each other due to connections and joints

→ Introduction of **constraints**

Generalized Coordinates (2)

- **Holonomic constraints** can be formulated as equations of the coordinates $\boldsymbol{x_i}$ (k: number of constraints):

$$f_j(x_1, \dots, x_{3N}) = 0 \qquad j = 1, \dots, k$$

- The $3N$ coordinates can be reduced to $\boldsymbol{n = 3N - k}$ independent **generalized coordinates** $\boldsymbol{q_i}$ using k independent constraints which must automatically satisfy the constraints:

$$x_i = x_i(q_1, \dots, q_n) \qquad i = 1, \dots, 3N \quad \text{and} \quad n = 3N - k$$

$$f_j(q_1, \dots, q_n) = 0 \qquad j = 1, \dots, k \quad \text{and} \quad n = 3N - k$$

Generalized Coordinates : 2D pendulum

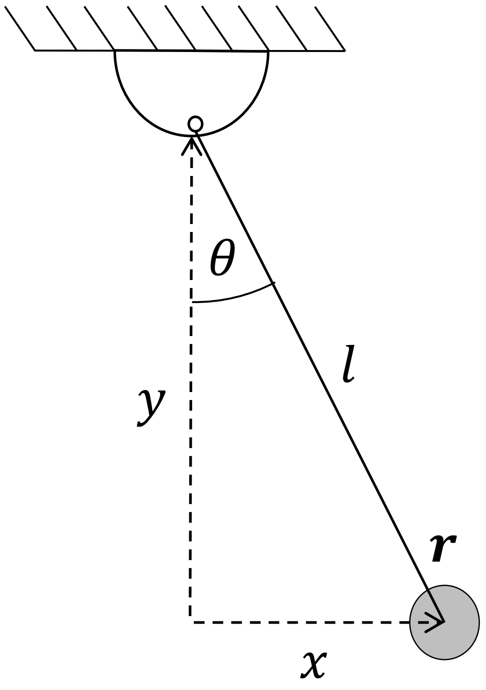
- The rod of a **plane pendulum (2D)** should always have the **length l** and must therefore fulfill the following constraint ($k = 1$) according to Pythagoras:

$$f_1(x_1, x_2) = 0 \qquad x_1 = x,\ x_2 = y$$

$$\Leftrightarrow x^2 + y^2 - l^2 = 0$$

- There is only one **generalized coordinate q**, since $n = 2N - k = 1$. The coordinates x, y of the center of mass $\boldsymbol{r}$ depend on θ:

$$x = l \cdot \sin\theta \qquad y = l \cdot \cos\theta \quad \rightarrow \quad \boldsymbol{r} = f(q) = l \cdot \begin{pmatrix} \sin\theta \\ \cos\theta \end{pmatrix}$$



Generalized Coordinates : 2D pendulum

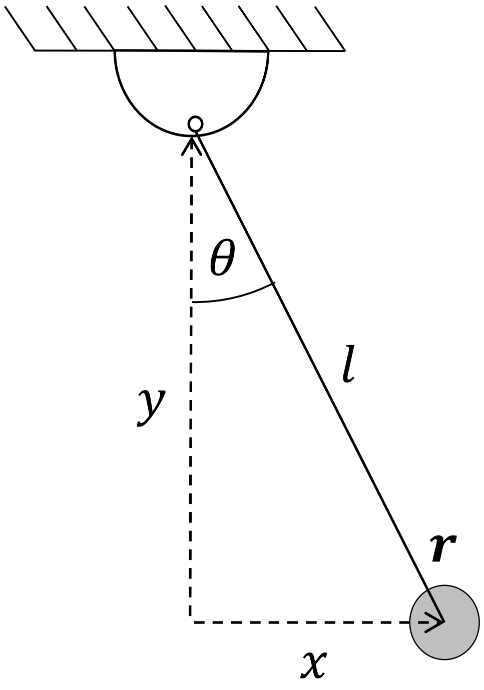
- The generalized coordinate automatically satisfies the constraint:

$$(l \cdot \sin\theta)^2 + (l \cdot \cos\theta)^2 - l^2 = 0$$

$$\Leftrightarrow l^2 \cdot (\sin^2\theta + \cos^2\theta - 1) = 0$$

- As the following generally applies:

$$\sin^2\theta + \cos^2\theta = 1$$



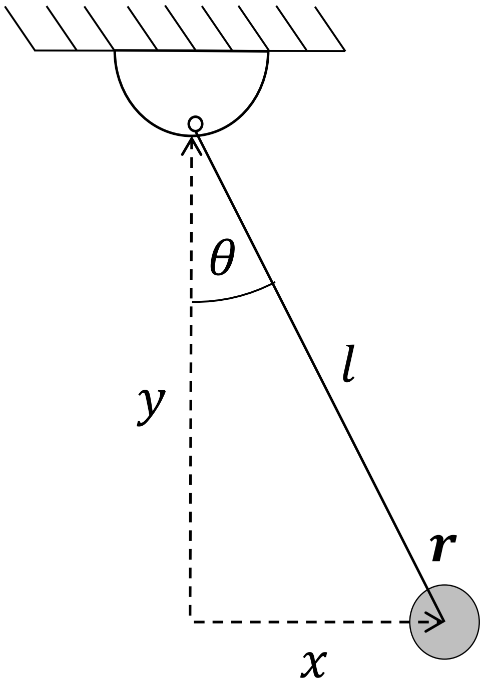
Generalized Coordinates: 3D pendulum

- Position of the mass: $\boldsymbol{r} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$
- Constraint ($k = 1$) on a sphere surface

$$|\boldsymbol{r}| = l \Leftrightarrow |\boldsymbol{r}| - l = 0$$
$$f_1(\boldsymbol{r}) = |\boldsymbol{r}| - l = 0$$

- Generalisierte Koordinaten ($n = 3N - k = 2$): $\boldsymbol{q} = \begin{pmatrix} \theta \\ \phi \end{pmatrix}$

$$\boldsymbol{r} = f(\boldsymbol{q}) = l \cdot \begin{pmatrix} \sin\theta\cos\phi \\ \sin\theta\sin\phi \\ -\cos\theta \end{pmatrix}$$



Generalized Coordinates: Example

$$|\boldsymbol{r}| - l = 0 \Rightarrow |\boldsymbol{r}|^{\mathbf{2}} - l^2 = 0$$

$$\boldsymbol{r} = f(\boldsymbol{q}) = l \cdot \begin{pmatrix} \sin\theta\cos\phi \\ \sin\theta\sin\phi \\ -\cos\theta \end{pmatrix}$$

$$|\boldsymbol{r}|^{\mathbf{2}} =$$

Generalized Coordinates: Example

$$|\boldsymbol{r}| - l = 0 \Rightarrow |\boldsymbol{r}|^{\mathbf{2}} - l^2 = 0$$

$$\boldsymbol{r} = f(\boldsymbol{q}) = l \cdot \begin{pmatrix} \sin\theta\cos\phi \\ \sin\theta\sin\phi \\ -\cos\theta \end{pmatrix}$$

$$\begin{aligned} |\boldsymbol{r}|^{\mathbf{2}} = l^2 \cdot \begin{pmatrix} \sin\theta\cos\phi \\ \sin\theta\sin\phi \\ -\cos\theta \end{pmatrix}^2 \\ &= l^2 \cdot (\sin^2\theta\cos^2\phi + \sin^2\theta\sin^2\phi + \cos^2\theta) \\ &= l^2 \cdot (\sin^2\theta \cdot (\cos^2\phi + \sin^2\phi) + \cos^2\theta) \\ &= l^2 \cdot (\sin^2\theta + \cos^2\theta) \\ &= l^2 \end{aligned}$$

Dynamic Model: Equation of Motion

- General equation of motion

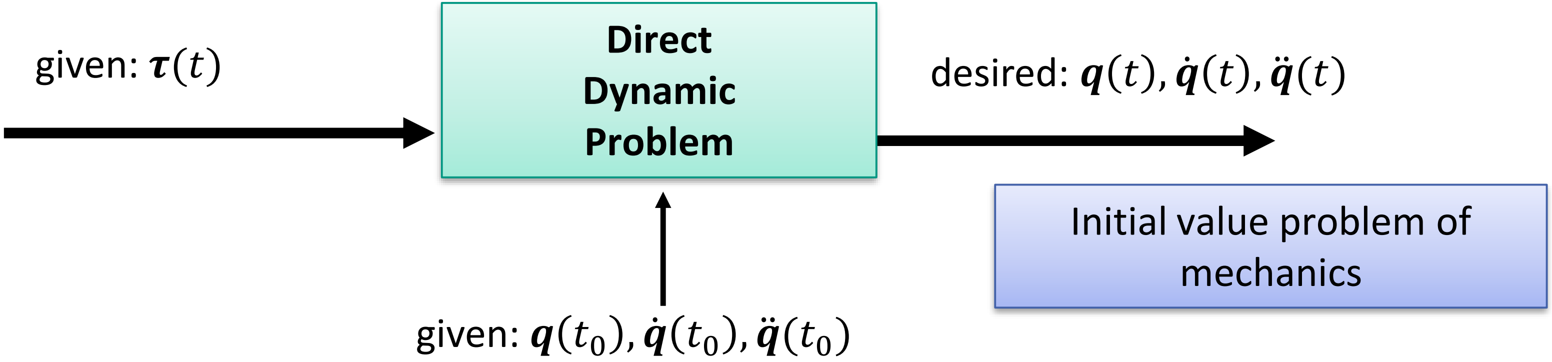
$$\boldsymbol{\tau} = M(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + g(\boldsymbol{q}) + \epsilon(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})$$

$\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}$:	$n \times 1$	vector of **generalized coordinates** (**position, velocity and acceleration**)
$\boldsymbol{\tau}$:	$n \times 1$	vector of **generalized forces**
$M(\boldsymbol{q})$:	$n \times n$	matrix of mass inertia (symmetric, positive-definite)
$C(\dot{\boldsymbol{q}},\boldsymbol{q})\dot{\boldsymbol{q}}$:	$n \times 1$	vector with centripetal and Coriolis components
$g(\boldsymbol{q})$:	$n \times 1$	vector of gravitational components
$\epsilon(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})$:	$n \times 1$	non-linear effects, e.g. friction (often neglected)

n: degrees of freedom of the robot

Direct Dynamic Problem

- Calculate the resulting changes in movement based on external forces and moments as well as the initial state and the dynamic properties of the robot



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

(non-linear effects neglected)

→ solve differential equation for $q(t), \dot{q}(t), \ddot{q}(t)$

Inverse Dynamic Problem

Calculate required driving forces and torques based on the desired motion parameters and the dynamic properties of the robot

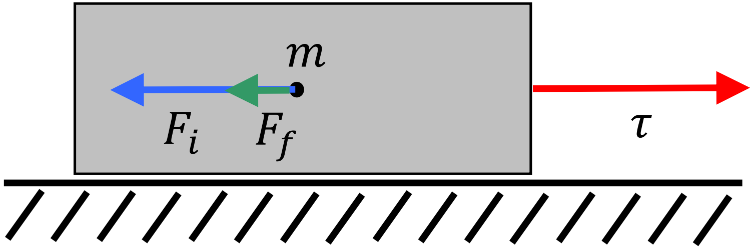
given: $\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t), \ddot{\boldsymbol{q}}(t)$ → **Inverse Dynamic Problem** → desired: $\boldsymbol{\tau}(t)$

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

(non-linear effects neglected)

Calculate the right part of the equation by inserting $\boldsymbol{q}(t)$, $\dot{\boldsymbol{q}}(t)$, $\ddot{\boldsymbol{q}}(t)$

Dynamic Model: Example



$F_i = -m\ddot{x}$	inertia
$F_f = -K_{sf}\dot{x}$	sliding friction
τ	external force

- Balance of forces: $\tau = -(F_i + F_f)$
- Equation of motion: $\tau = m\ddot{x} + K_{sf}\dot{x}$
- **Inverse problem:** Given the state of motion, what external force τ acts on the system or is required to maintain the state of motion?
- **Direct problem:** Given the external force and current state of motion, what is the new motion (or acceleration) state of the system?

Contents

Modeling of Dynamics

- There are various methods for deriving the terms of the general equation of motion:

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

- **Lagrange**
  - Work or energy considerations of the overall system
  - Equations of motion by formal derivation

- **Newton-Euler**
  - Based on the Newton and Euler equations for rigid bodies
  - Isolated consideration of the arm elements
  - Efficient method due to recursive algorithm

Contents

Method of Lagrange

- Lagrange function:

$$L(\boldsymbol{q}, \dot{\boldsymbol{q}}) = E_{kin}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - E_{pot}(\boldsymbol{q})$$

- The equation of motion can be derived using the Lagrange function for each generalized coordinate:

$$\tau_i = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i}$$

q_i: i-th component of the generalized coordinates
τ_i: i-th component of the generalized forces

Method of Lagrange

- The resulting equation can be written in scalar form:

$$\tau_i = \sum_{j=1}^{n} M_{ij}\,\ddot{q}_j + \sum_{j=1}^{n}\sum_{k=1}^{n} C_{ijk}\,\dot{q}_j\dot{q}_k + g(q)$$

C_{ijk}: first order Christoffel symbols

$$C_{ijk} = \frac{1}{2}\left(\frac{\partial M_{ij}}{\partial q_k} + \frac{\partial M_{ik}}{\partial q_j} - \frac{\partial M_{jk}}{\partial q_i}\right)$$

Method of Lagrange: Procedure

- **Goal:** Determine the equation of motion for each joint i of a robot

$$\tau_i = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i}$$

- **Procedure**:
    1. Calculate E_{kin} and E_{pot}
    2. Express E_{kin} and E_{pot} in generalized coordinates

$$L(\boldsymbol{q}, \dot{\boldsymbol{q}}) = E_{kin}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - E_{pot}(\boldsymbol{q})$$

    3. Calculate the derivations

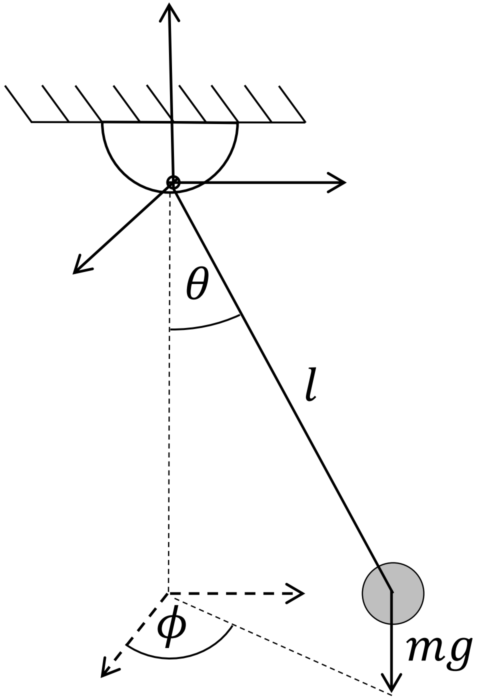
Method of Lagrange: 3D-Pendulum (1)

- 3D-pendulum with gravity
  (see example of generalized coordinates)

- $\boldsymbol{r} = f(\boldsymbol{q}) = l \cdot \begin{pmatrix} \sin\theta\cos\phi \\ \sin\theta\sin\phi \\ -\cos\theta \end{pmatrix}$

- $E_{kin} = \frac{1}{2} m |\dot{\boldsymbol{r}}|^2 = \frac{1}{2} m l^2 \left(\dot{\theta}^2 + \dot{\phi}^2 \cdot \sin^2\theta\right)$

- $E_{pot} = m \cdot g \cdot h = mg \cdot (-l \cdot \cos\theta)$



$f(x) = u(x) \cdot v(x) \Rightarrow \dot{f}(x) = u(x) \cdot \dot{v}(x) + \dot{u}(x) \cdot v(x)$

Method of Lagrange: 3D-Pendulum (2)

- Lagrange function with $\boldsymbol{q} = (\theta, \phi)$

$$L(\boldsymbol{q}, \dot{\boldsymbol{q}}) = E_{kin}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - E_{pot}(\boldsymbol{q})$$

$$= \frac{1}{2} m l^2 \left(\dot{\theta}^2 + \dot{\phi}^2 \cdot \sin^2\theta\right) + mgl \cdot \cos\theta$$

- Derive:

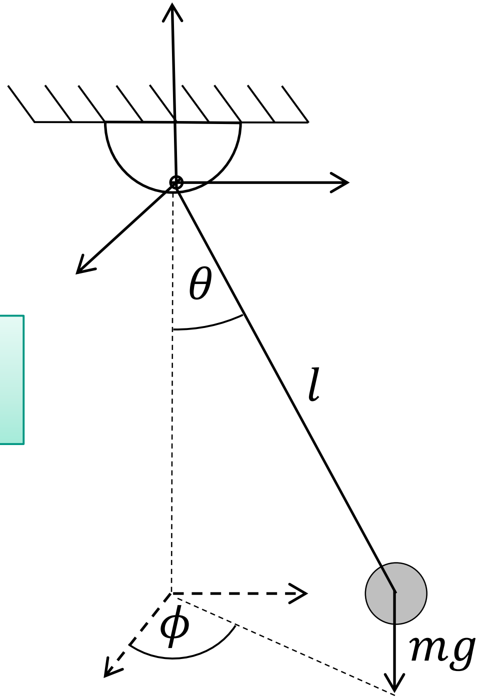
$$\boldsymbol{\tau_i = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i}}$$

$$\frac{\partial L}{\partial \theta} =$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) =$$

$$\frac{\partial L}{\partial \phi} =$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\phi}}\right) =$$



Method of Lagrange: 3D-Pendulum (2)

- Lagrange function with $\boldsymbol{q} = (\theta, \phi)$

$$L(\boldsymbol{q}, \dot{\boldsymbol{q}}) = E_{kin}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - E_{pot}(\boldsymbol{q})$$

$$= \frac{1}{2} m l^2 \left(\dot{\theta}^2 + \dot{\phi}^2 \cdot \sin^2 \theta\right) + mgl \cdot \cos\theta$$

- Derive:

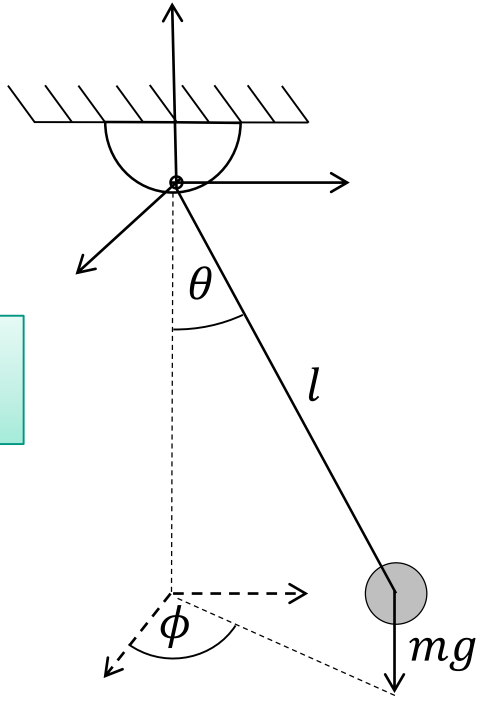
$$\boldsymbol{\tau_i = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i}}$$

$$\frac{\partial L}{\partial \theta} = ml^2 \sin\theta \cos\theta \cdot \dot{\phi}^2 - mgl \cdot \sin\theta$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) = \frac{d}{dt}\left(ml^2 \dot{\theta}\right) = ml^2 \ddot{\theta}$$

$$\frac{\partial L}{\partial \phi} = 0$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\phi}}\right) = \frac{d}{dt}\left(ml^2 \dot{\phi} \cdot \sin^2 \theta\right) = ml^2 \sin^2 \theta \cdot \ddot{\phi} + 2ml^2 \sin\theta \cos\theta \cdot \dot{\theta} \cdot \dot{\phi}$$



Method of Lagrange: 3D-Pendulum (3)

$$\boldsymbol{\tau_i = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i}}$$

$$\frac{\partial L}{\partial \theta} = ml^2 \sin\theta \cos\theta \cdot \dot{\phi}^2 - mgl \cdot \sin\theta$$
$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) = \frac{d}{dt}\left(ml^2\dot{\theta}\right) = ml^2\ddot{\theta}$$
$$\frac{\partial L}{\partial \phi} = 0$$
$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\phi}}\right) = \frac{d}{dt}\left(ml^2\,\dot{\phi} \cdot \sin^2\theta\right) = ml^2 \sin^2\theta \cdot \ddot{\phi} + 2ml^2 \sin\theta\cos\theta \cdot \dot{\theta}\dot{\phi}$$

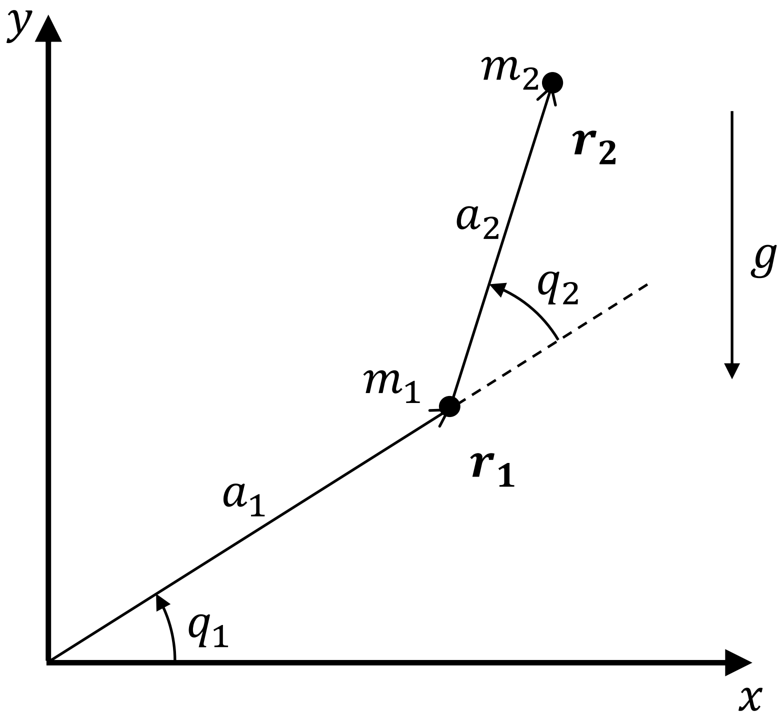
- Structure of general equations of motion

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

- Equation of motion of the 3D-pendulum (no external forces -> $\boldsymbol{\tau} = \mathbf{0}$)

$$\mathbf{0} = \begin{bmatrix} ml^2 & 0 \\ 0 & ml^2\sin^2\theta \end{bmatrix}\begin{bmatrix} \ddot{\theta} \\ \ddot{\phi} \end{bmatrix} + \begin{bmatrix} -ml^2 \sin\theta\cos\theta \cdot \dot{\phi}^2 \\ 2ml^2 \sin\theta\cos\theta \cdot \dot{\theta}\dot{\phi} \end{bmatrix} + \begin{bmatrix} mgl\sin\theta \\ 0 \end{bmatrix}$$

Lagrange: Example – Two pivot joints (1)



Idealization:

- masses of arm elements as point masses in m_1 and m_2
- no friction

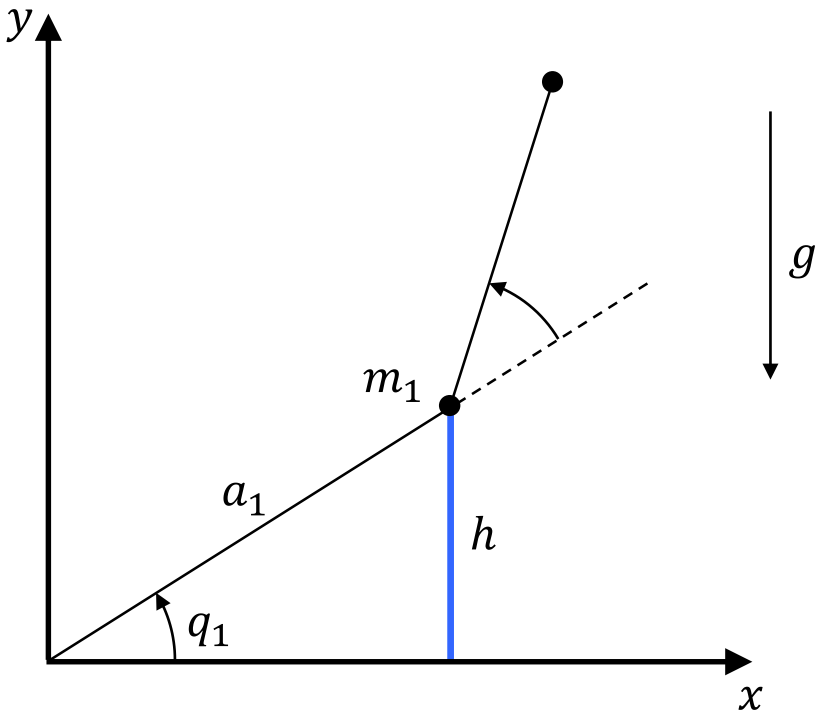
Constraints of the system ($k = 2$):

$$f_1(\boldsymbol{r}_1, \boldsymbol{r}_2) = |\boldsymbol{r}_1|^2 - a_1^2 = 0$$
$$f_2(\boldsymbol{r}_1, \boldsymbol{r}_2) = |\boldsymbol{r}_2 - \boldsymbol{r}_1|^2 - a_2^2 = 0$$

→$n = 2N - k = 2$

→generalized coordinates q_1, q_2

Lagrange: Example – Two pivot joints (2)

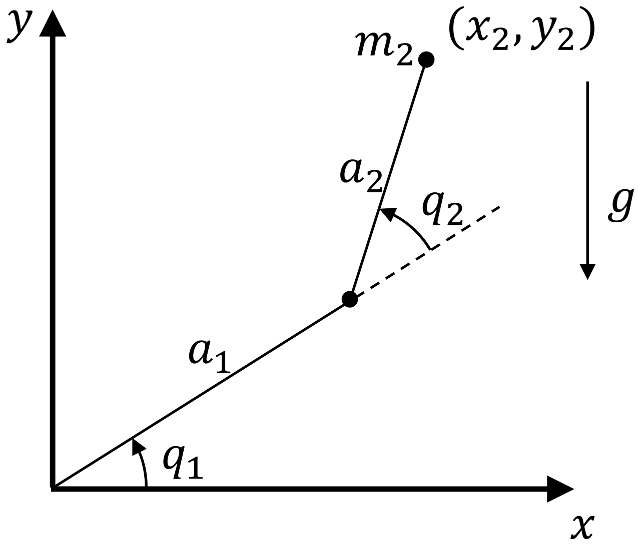


- Joint 1:

$$E_{kin,1} = \frac{1}{2} m_1 v^2 = \frac{1}{2} m_1 a_1^2 \dot{q_1}^2$$

$$E_{pot,1} = m_1 g h = m_1 g a_1 \sin(q_1)$$

Lagrange: Example – Two pivot joints (3)



Joint 2:

Position:

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} a_1 \cos(q_1) + a_2 \cos(q_1 + q_2) \\ a_1 \sin(q_1) + a_2 \sin(q_1 + q_2) \end{bmatrix}$$

Velocity:

$$\begin{bmatrix} \dot{x}_2 \\ \dot{y}_2 \end{bmatrix} = \begin{bmatrix} -a_1 \dot{q}_1 \sin(q_1) - a_2(\dot{q}_1 + \dot{q}_2)\sin(q_1 + q_2) \\ a_1 \dot{q}_1 \cos(q_1) + a_2(\dot{q}_1 + \dot{q}_2)\cos(q_1 + q_2) \end{bmatrix}$$

Lagrange: Example – Two pivot joints (4)

$$\begin{bmatrix}\dot{x}_2\\ \dot{y}_2\end{bmatrix} = \begin{bmatrix}-a_1\dot{q}_1\sin(q_1) - a_2(\dot{q}_1 + \dot{q}_2)\sin(q_1+q_2)\\ a_1\dot{q}_1\cos(q_1) + a_2(\dot{q}_1 + \dot{q}_2)\cos(q_1+q_2)\end{bmatrix}$$

- Joint 2:
  - Kinetic energy:

    $$v_2{}^2 = \dot{x}_2{}^2 + \dot{y}_2{}^2 = a_1{}^2\dot{q}_1{}^2 + a_2{}^2(\dot{q}_1 + \dot{q}_2)^2 + 2a_1a_2\left(\dot{q}_1{}^2 + \dot{q}_1\dot{q}_2\right)\cos(q_2)$$

    $$E_{kin,2} = \frac{1}{2}m_2v_2{}^2 = \frac{1}{2}m_2a_1{}^2\dot{q}_1{}^2 + \frac{1}{2}m_2a_2{}^2(\dot{q}_1 + \dot{q}_2)^2 + m_2a_1a_2\left(\dot{q}_1{}^2 + \dot{q}_1\dot{q}_2\right)\cos(q_2)$$

  - Potential energy:

    $$E_{pot,2} = m_2gy_2 = m_2g[a_1\sin(q_1) + a_2\sin(q_1+q_2)]$$

  - Lagrange function:

    $$L = E_{kin} - E_{pot} = E_{kin,1} + E_{kin,2} - E_{pot,1} - E_{pot,2}$$

    $$= \frac{1}{2}(m_1+m_2)a_1{}^2\dot{q}_1{}^2 + \frac{1}{2}m_2a_2{}^2(\dot{q}_1 + \dot{q}_2)^2 + m_2a_1a_2\left(\dot{q}_1{}^2 + \dot{q}_1\dot{q}_2\right)\cos(q_2)$$

    $$-(m_1+m_2)ga_1\sin(q_1) - m_2ga_2\sin(q_1+q_2)$$

Lagrange: Example – Two pivot joints (5)

- Equation of motion
  - Joint 1:

$$\frac{\partial L}{\partial \dot{q}_1} = (m_1 + m_2)a_1{}^2\dot{q}_1 + m_2 a_2{}^2(\dot{q}_1 + \dot{q}_2) + m_2 a_1 a_2(2\dot{q}_1 + \dot{q}_2)\cos(q_2)$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_1}\right) = (m_1 + m_2)a_1{}^2\ddot{q}_1 + m_2 a_2{}^2(\ddot{q}_1 + \ddot{q}_2) + m_2 a_1 a_2(2\ddot{q}_1 + \ddot{q}_2)\cos(q_2)$$
$$-m_2 a_1 a_2\left(2\dot{q}_1\dot{q}_2 + \dot{q}_2{}^2\right)\sin(q_2)$$

$$\frac{\partial L}{\partial q_1} = -(m_1 + \mathrm{m}_2)g a_1 \cos(q_1) - m_2 g a_2 \cos(q_1 + q_2)$$

Lagrange: Example – Two pivot joints (5)

- Equation of motion
  - Joint 2:

$$\frac{\partial L}{\partial \dot{q}_2} = m_2 {a_2}^2(\dot{q}_1 + \dot{q}_2) + m_2 a_1 a_2 \dot{q}_1 \cos(q_2)$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_2}\right) = m_2 {a_2}^2(\ddot{q}_1 + \ddot{q}_2) + m_2 a_1 a_2 \ddot{q}_1 \cos(q_2) - m_2 a_1 a_2 \dot{q}_1 \dot{q}_2 \sin(q_2)$$

$$\frac{\partial L}{\partial q_2} = -m_2 a_1 a_2\left({\dot{q}_1}^2 + \dot{q}_1 \dot{q}_2\right)\sin(q_2) - m_2 g a_2 \cos(q_1 + q_2)$$

Lagrange: Example – Two pivot joints (6)

- Equation of motion:

$$\begin{bmatrix}\tau_1\\ \tau_2\end{bmatrix} = \begin{bmatrix}(m_1+m_2){a_1}^2 + m_2{a_2}^2 + 2m_2a_1a_2\cos(q_2) & m_2{a_2}^2 + m_2a_1a_2\cos(q_2)\\ m_2{a_2}^2 + m_2a_1a_2\cos(q_2) & m_2{a_2}^2\end{bmatrix}\begin{bmatrix}\ddot{q}_1\\ \ddot{q}_2\end{bmatrix}$$

$$+\begin{bmatrix}-m_2a_1a_2\left(2\dot{q}_1\dot{q}_2 + {\dot{q}_2}^2\right)\sin(q_2)\\ m_2a_1a_2{\dot{q}_1}^2\sin(q_2)\end{bmatrix}$$

$$+\begin{bmatrix}(m_1+m_2)ga_1\cos(q_1) + m_2ga_2\cos(q_1+q_2)\\ m_2ga_2\cos(q_1+q_2)\end{bmatrix}$$

- Summarized:

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

Method of Lagrange: Summary

To determine the equations of motion, the kinetic and potential energy must be determined. From this, the Lagrange function can be calculated.

$$L(\boldsymbol{q}, \dot{\boldsymbol{q}}) = E_{kin}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - E_{pot}(\boldsymbol{q})$$

The equations of motion then follow formally by differentiation:

$$\boldsymbol{\tau_i} = \frac{\boldsymbol{d}}{\boldsymbol{dt}}\left(\frac{\boldsymbol{\partial L}}{\boldsymbol{\partial \dot{q}_i}}\right) - \frac{\boldsymbol{\partial L}}{\boldsymbol{\partial q_i}}$$

Method of Lagrange: Properties

Karlsruhe Institute of Technology

Properties

- Simple formulation of the equations
- Closed model, analytically evaluable
- Very extensive calculations $O(n^3)$
  (n : number of joints)
- Only driving torques are calculated

H2T

Contents

Method of Newton-Euler

- **Idea:** Forces and moments acting on an arm element can be calculated from the joint angle positions, velocities and accelerations using the **recursive Newton-Euler algorithm (RNEA)**

Input: $\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t), \ddot{\boldsymbol{q}}(t)$ → Recursive Newton-Euler Algorithm → Output: $\boldsymbol{\tau}(t)$

- **Properties**
  - Isolated considertaion of each arm element
  - Efficient calculation in real-time with **complexity $\boldsymbol{O(n)}$** possible through recursive algorithm

Newton-Euler: Mathematical Basics

- The **moment of inertia** of a rigid body in a **rotation motion** is comparable to the **mass** in a linear motion:
  - linear motion: **force** = *mass* · *acceleration* (Newton‘s second law)

$$\boldsymbol{f} = m \cdot \boldsymbol{a} = m \cdot \dot{\boldsymbol{v}}_c = m \cdot \ddot{\boldsymbol{c}}$$

  - rotation motion: **torque** = *moment of inertia* · *angular acceleration* (Angular momentum theorem)

$$\boldsymbol{M} = \bar{\boldsymbol{I}}^{CoM} \boldsymbol{\alpha} = \bar{\boldsymbol{I}}^{CoM} \dot{\boldsymbol{\omega}}$$

CoM: Center of Mass

Newton-Euler: Euler‘s Equation of Motion

Karlsruhe Institute of Technology

If a body is subjected to a torque, **gyroscopic effects** develop (Euler forces and centrifugal forces at all mass points)

The torques can be added up and described by **Euler's equation of motion** for rigid bodies:

$$\boldsymbol{n}_{CoM} = \bar{\boldsymbol{I}}^{CoM}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \bar{\boldsymbol{I}}^{CoM}\boldsymbol{\omega}$$

$\boldsymbol{n}_{CoM}$:	torques around the center of mass CoM
$\bar{\boldsymbol{I}}^{CoM}$:	moments of inertia around the center of mass
$\boldsymbol{\omega}$:	angular velocities of the rigid body
$\dot{\boldsymbol{\omega}}$:	angular accelerations (time derivative of $\boldsymbol{\omega}$)

gyroscopic effects: Kreiselwirkung

Newton-Euler: Equation of Motion

- The **Newton-Euler equations**, which describe the complete motion of a **rigid body**, can be expressed in the form of a single equation:

$$\boxed{\begin{pmatrix} \boldsymbol{n}_{COM} \\ \boldsymbol{f} \end{pmatrix} = \begin{pmatrix} \bar{\boldsymbol{I}}^{CoM}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \bar{\boldsymbol{I}}^{CoM}\boldsymbol{\omega} \\ m\ddot{\boldsymbol{c}} \end{pmatrix}}$$

- in simple terms:

$$\boxed{\mathbf{f} = \boldsymbol{I}\mathbf{a} + \mathbf{v} \times \boldsymbol{I}\mathbf{v}}$$

where

$$\mathbf{f} = \begin{pmatrix} \boldsymbol{n}_{COM} \\ \boldsymbol{f} \end{pmatrix} \qquad \mathbf{v} = \begin{pmatrix} \boldsymbol{\omega} \\ \boldsymbol{v}_C \end{pmatrix} \qquad \mathbf{a} = \begin{pmatrix} \dot{\boldsymbol{\omega}} \\ \dot{\boldsymbol{v}}_C \end{pmatrix}$$

$\boldsymbol{v}_C$: linear velocity of the body in relation to CoM

$\dot{\boldsymbol{v}}_C$: linear acceleration of the body in relation to CoM

$\mathbf{f}, \mathbf{v}, \mathbf{a}$: 6D force or motion vectors, which describe all forces and motions (velocity, acceleration) acting on the body

Newton-Euler: Basic Principle

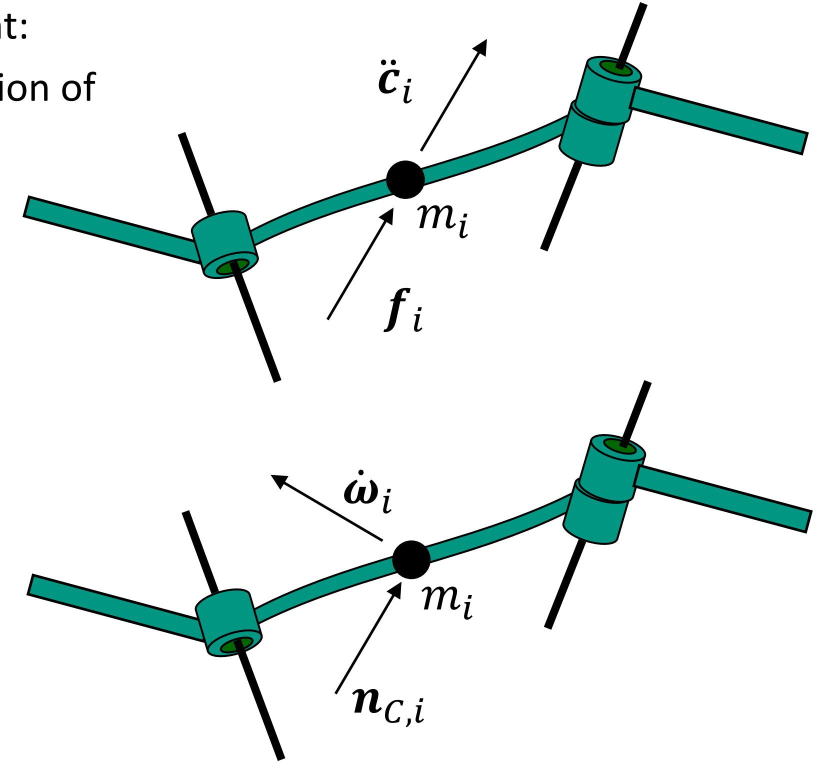
- Considering the center of mass of a single arm element:
  - **force = change of momentum** → temporal derivation of the momentum (Newton's second law)

$$\boldsymbol{f}_i = \frac{d}{dt}(m_i\, \boldsymbol{v}_{C,i}) = m_i \ddot{\boldsymbol{c}}_i$$

  - **torque = change of angular momentum** → time derivative of the angular momentum + torque of gyroscopic effects (Euler's equation of motion)

$$\begin{aligned}\boldsymbol{n}_{C,i} &= \frac{d}{dt}(I_i\, \boldsymbol{\omega}_i) + \boldsymbol{\omega}_i \times I_i\, \boldsymbol{\omega}_i \\ &= I_i \dot{\boldsymbol{\omega}}_i + \boldsymbol{\omega}_i \times I_i\, \boldsymbol{\omega}_i\end{aligned}$$

- Forces and torques acting on an arm element can be calculated from velocity and joint angular velocity.



Newton-Euler: Concatenation

The accelerations $\ddot{\boldsymbol{c}}_i$ and $\dot{\boldsymbol{\omega}}_{\boldsymbol{i}}$ of an arm element i depend on the accelerations of the **preceding** arm elements.

Accelerations can be calculated recursively via the kinematic model
from the base to the gripper → **forward equations**

The force $\boldsymbol{f}_{\boldsymbol{i}}$ and the torque $\boldsymbol{n}_{C,i}$ which act on an arm element i depend on the **subsequent** arm elements.

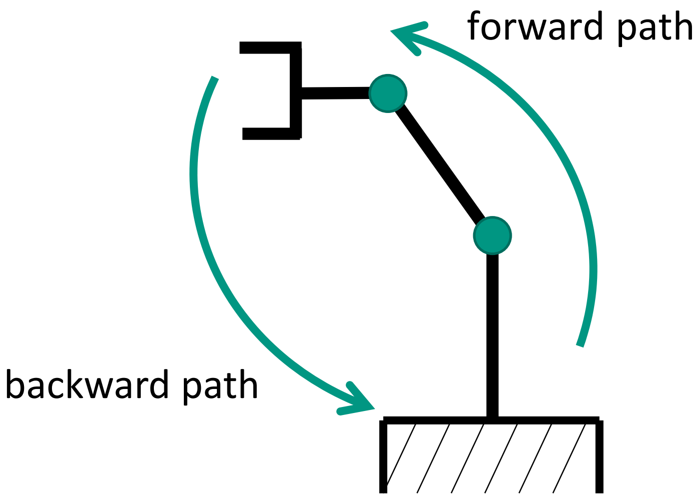
Forces and moments can be calculated recursively
from the gripper to the base → **backward equations**

→ **Recursive Newton-Euler Algorithm (RNEA)**

concatenation: Verkettung

Recursive Newton-Euler Algorithm (RNEA)

- General procedure:
  1. Recursive calculation of **velocity** and **acceleration** for each arm element from the base to the end effector (**forward path**)
  2. Calculation of the **forces/moments** which act on each arm element, or which are required for the accelerations using **Newton-Euler**
  3. Recursive calculation of the forces **over all arm elements** and the **joint force variables** for the respective joint type (**backward path**)



RNEA: Step 1

- Recursive calculation of the **velocity** and **acceleration** of each individual arm element i from the base to the end effector (**forward path**)

- **Velocity**

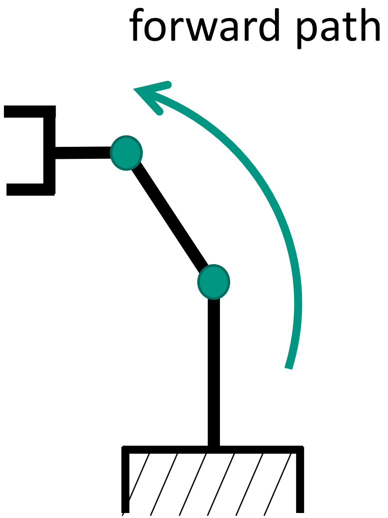
$$\mathbf{v}_i = \mathbf{v}_{p(i)} + \boldsymbol{\phi}_i \dot{\boldsymbol{q}}_i$$

$\dot{\boldsymbol{q}}_i$: generalized velocity of the arm element i
$\boldsymbol{\phi}_i$: $6 \times n$ motion matrix (depends on joint type)
$\mathbf{v}_{p(i)}$: velocity of the preceding element $p(i)$

- **Acceleration**

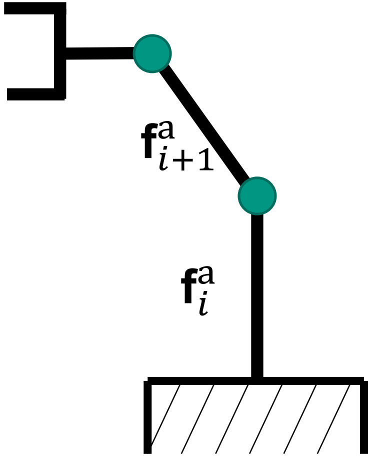
$$\mathbf{a}_i = \mathbf{a}_{p(i)} + \boldsymbol{\phi}_i \ddot{\boldsymbol{q}}_i + \dot{\boldsymbol{\phi}}_i \dot{\boldsymbol{q}}_i$$

$\ddot{\boldsymbol{q}}_i$: generalized acceleration of the arm element i
$\dot{\boldsymbol{\phi}}_i$: derivation of $\boldsymbol{\phi}_i$



RNEA: Step 2

- Calculation of the **forces/moments** using the **Newton-Euler equation**, which act on each arm element i due to the acceleration (from step 1)



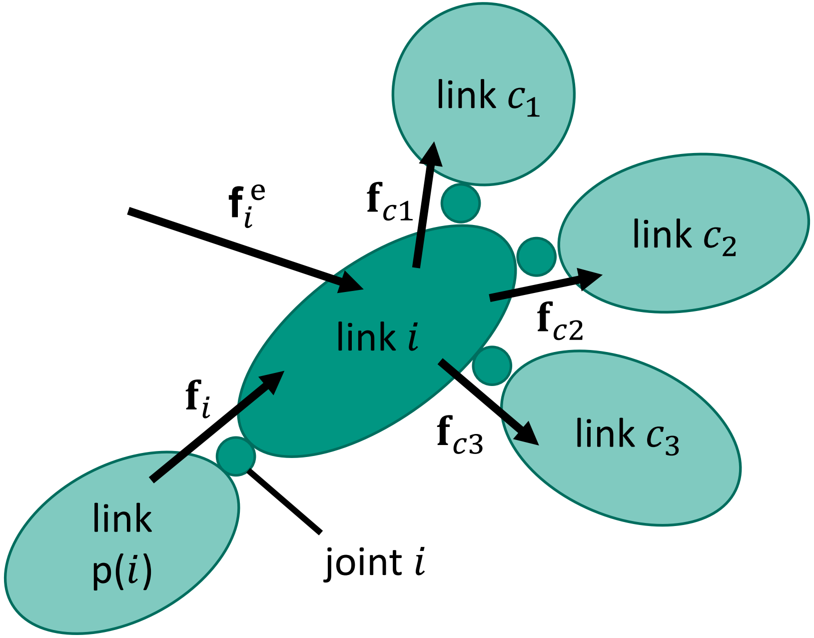
$$\mathbf{f}_i^{\mathrm{a}} = \boldsymbol{I}_i \mathbf{a}_i + \mathbf{v}_i \times \boldsymbol{I}_i \mathbf{v}_i$$

$\mathbf{f}_i^{\mathrm{a}}$: forces acting on arm element i **due to $\mathbf{a}_i$**
$\boldsymbol{I}_i$: moment of inertia of arm element i
$\mathbf{v}_i$: velocity of arm element i (calculated in step 1)
$\mathbf{a}_i$: acceleration of arm element i (calculated in step 1)

RNEA: Step 3

- Recursive calculation of the forces **between the arm elements (left)** and the **joint force variables** for the respective joint type (**backward path**)

$$\mathbf{f}_i = \mathbf{f}_i^{\mathrm{a}} - \mathbf{f}_i^{\mathbf{e}} + \sum_{j \,\epsilon\, c(i)} \mathbf{f}_j$$



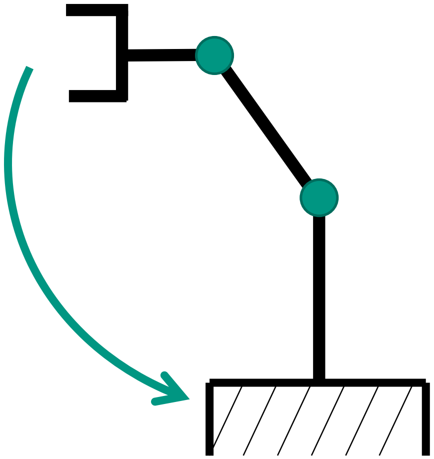
RNEA: Step 3

- Recursive calculation of the forces **between the arm elements (left)** and the **joint force variables** for the respective joint type (**backward path**)

$$\mathbf{f}_i = \mathbf{f}_i^{\mathrm{a}} - \mathbf{f}_i^{\mathbf{e}} + \sum_{j \,\epsilon\, c(i)} \mathbf{f}_j$$

$$\boldsymbol{\tau}_i = \boldsymbol{\phi}_i^T \mathbf{f}_i$$

backward path



$\mathbf{f}_i$: resulting force on arm element i
$\mathbf{f}_i^{\mathbf{e}}$: sum of all external forces acting on i
$\mathbf{f}_j$: force of an adjacent arm element j
$c(i)$: set of arm elements in the kinematic chain subsequent to i
$\boldsymbol{\phi}_i$: $6 \times n$ motion matrix (depends on joint type)
$\boldsymbol{\tau}_i$: generalized forces/torques acting on i

RNEA: Summary

1. Recursive calculation of the **velocity** and **acceleration** of each individual arm element i from the base to the end effector:

$$\mathbf{v}_i = \mathbf{v}_{p(i)} + \boldsymbol{\phi}_i \dot{\boldsymbol{q}}_i \qquad \text{with} \quad \mathbf{v}_0 = 0$$

$$\mathbf{a}_i = \mathbf{a}_{p(i)} + \boldsymbol{\phi}_i \ddot{\boldsymbol{q}}_i + \dot{\boldsymbol{\phi}}_i \dot{\boldsymbol{q}}_i \qquad \text{with} \quad \mathbf{a}_0 = -\mathbf{a}_g$$

2. Calculation of the **forces/moments** on each individual arm element i using **Newton-Euler**:

$$\mathbf{f}_i^{\mathrm{a}} = \boldsymbol{I}_i \mathbf{a}_i + \mathbf{v}_i \times \boldsymbol{I}_i \mathbf{v}_i$$

3. Recursive calculation of the forces **between the arm elements** and the **generalized forces** for the respective joint type

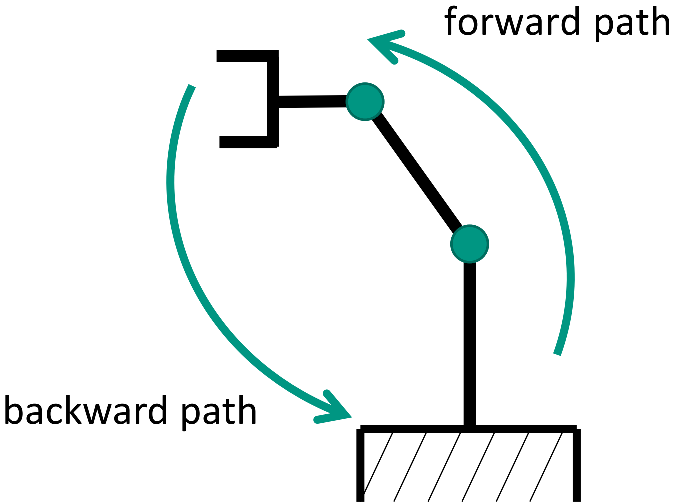
$$\boldsymbol{\tau}_i = \boldsymbol{\phi}_i^T \mathbf{f}_i \qquad \text{with} \quad \mathbf{f}_i = \mathbf{f}_i^{\mathrm{a}} - \mathbf{f}_i^{\mathbf{e}} + \sum_{j \,\epsilon c(i)} \mathbf{f}_j$$

Recursive Newton-Euler Algorithm (RNEA)

Karlsruhe Institute of Technology

- **Complete Algorithm**

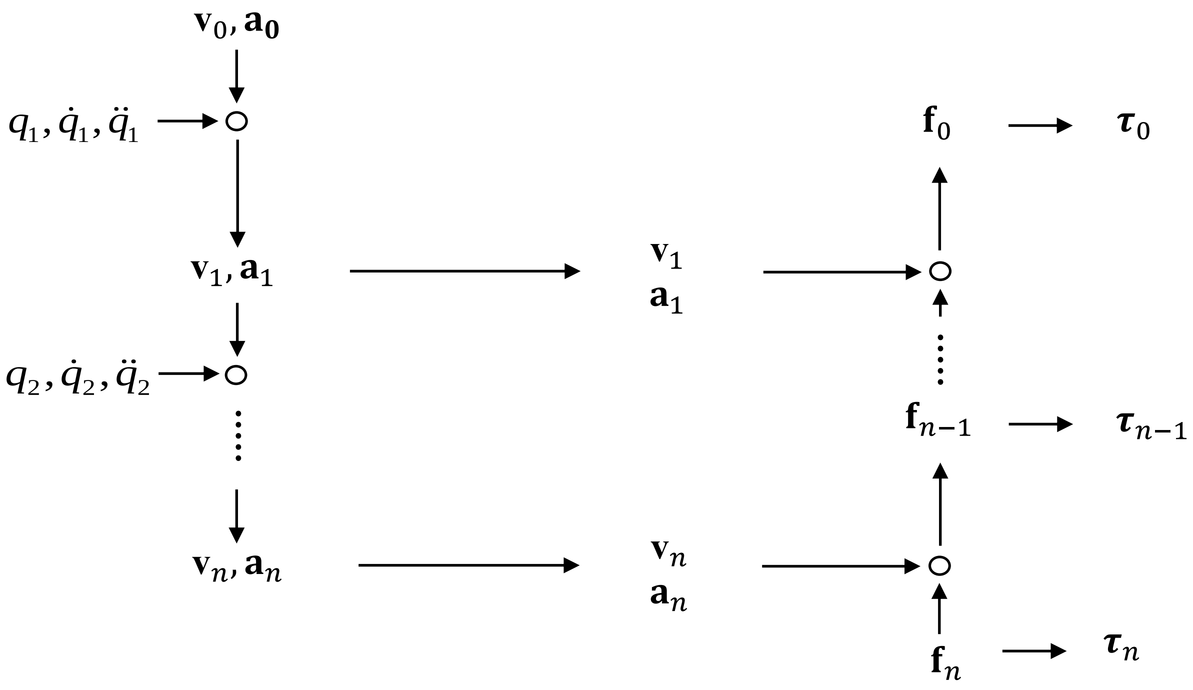
$$
\begin{aligned}
&\mathbf{v}_0 = 0 \\
&\mathbf{a}_0 = -\mathbf{a}_g \\
&\textbf{for } i = 1 \textbf{ to } n \textbf{ do} \\
&\quad \mathbf{v}_i = \mathbf{v}_{p(i)} + \boldsymbol{\phi}_i \dot{\boldsymbol{q}}_i \\
&\quad \mathbf{a}_i = \mathbf{a}_{p(i)} + \boldsymbol{\phi}_i \ddot{\boldsymbol{q}}_i + \dot{\boldsymbol{\phi}}_i \dot{\boldsymbol{q}}_i \\
&\quad \mathbf{f}_i = \boldsymbol{I}_i \mathbf{a}_i + \mathbf{v}_i \times \boldsymbol{I}_i \mathbf{v}_i - \mathbf{f}_i^{\mathbf{e}} \\
&\textbf{end for} \\
&\textbf{for } i = \mathrm{n} \textbf{ to } 1 \textbf{ do} \\
&\quad \boldsymbol{\tau}_i = \boldsymbol{\phi}_i^T \mathbf{f}_i \\
&\quad \textbf{if } p(i) \neq 0 \textbf{ then} \\
&\quad\quad \mathbf{f}_{p(i)} = \mathbf{f}_{p(i)} + \mathbf{f}_i \\
&\quad \textbf{end if} \\
&\textbf{end for}
\end{aligned}
$$



H2T

Method of Newton-Euler: Summary

(motion of the base)



(forces and torques on the end effector)

$$\boxed{\mathbf{f} = \boldsymbol{I}\mathbf{a} + \mathbf{v} \times \boldsymbol{I}\mathbf{v}}$$

where

$$\mathbf{v} = \begin{pmatrix} \boldsymbol{\omega} \\ \boldsymbol{v}_C \end{pmatrix} \quad \mathbf{a} = \begin{pmatrix} \dot{\boldsymbol{\omega}} \\ \dot{\boldsymbol{v}}_C \end{pmatrix}$$

$$\mathbf{f} = \begin{pmatrix} \boldsymbol{n}_C \\ \boldsymbol{f} \end{pmatrix}$$

Method of Newton-Euler: Properties

Karlsruhe Institute of Technology

Properties

- Arbitrary number of joints
- Loads on arm elements are calculated
- Effort $O(n)$ (n: number of joints)
- Recursive

Contents

Challenges of Dynamics

- The methods presented for modeling dynamics (Lagrange and Newton-Euler) are only approximations of the dynamics
- Non-linear forces (e.g. friction) cannot be modeled directly, but have a major influence:

$$\boxed{\boldsymbol{\tau} = M(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + g(\boldsymbol{q}) + \epsilon(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})}$$

$\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}}$:	$n \times 1$	vector of generalized coordinates (position, velocity and acceleration)
$\boldsymbol{\tau}$:	$n \times 1$	vector of generalized forces
$M(\boldsymbol{q})$:	$n \times n$	matrix of mass inertia (symmetric, positive-definite)
$C(\dot{\boldsymbol{q}},\boldsymbol{q})\dot{\boldsymbol{q}}$:	$n \times 1$	vector with centripetal and Coriolis components
$g(\boldsymbol{q})$:	$n \times 1$	vector of gravitational components
$\epsilon(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})$:	$n \times 1$	non-linear effects, e.g. friction

Challenges of Dynamics

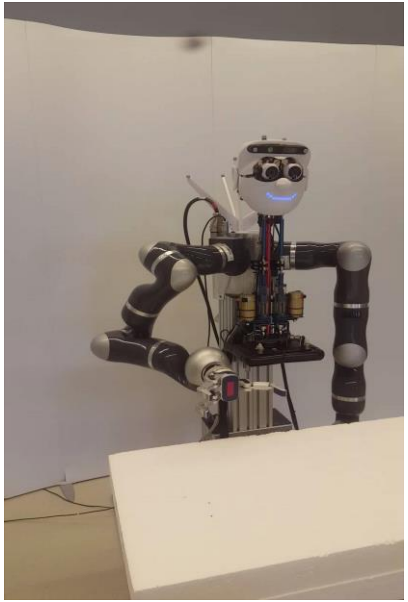
- The dynamics of a robot can change considerably over time, e.g. due to
  - Wear and tear
  - Material changes (elongation, etc.)

- The dynamics vary greatly depending on the task to be performed Examples:
  - Interaction with the environment
  - Grasping and manipulating objects
  - Use of tools

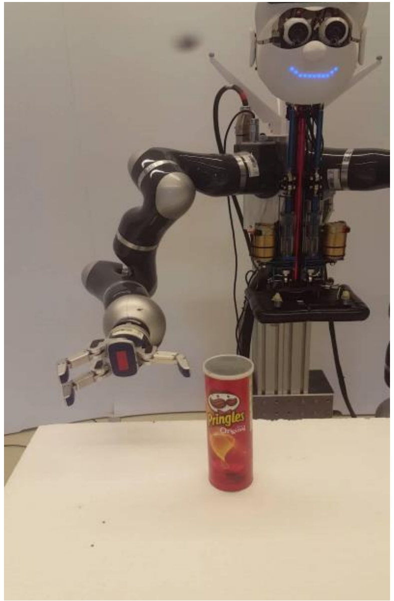
Learning of Dynamics

- Dynamics depend on the task to be performed (here: ‘pick and place’)

without object

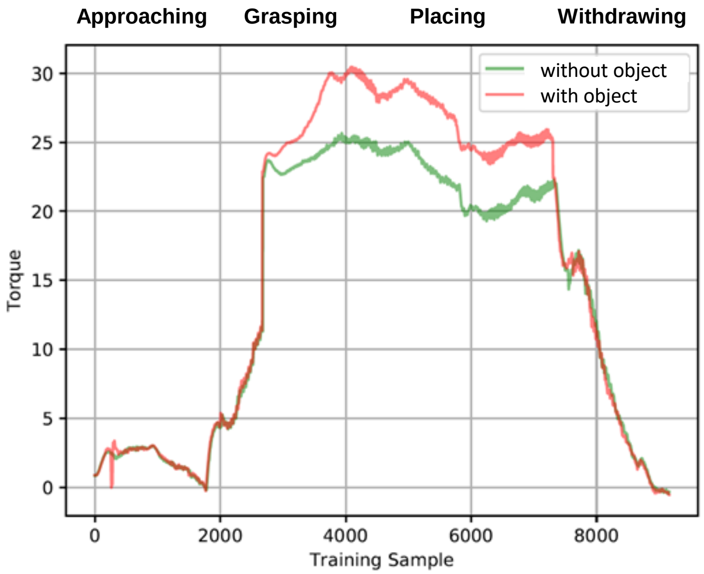


with object (851g)



Learning of Dynamics

- The ‘pick and place’ task can be divided into several phases:
  1. **Approaching** the object
  2. **Grasping** the object
  3. **Placing** the object
  4. **Withdrawing** from the object



- The diagram shows that the torques with and without the object differ greatly from each other

  → Dynamics must be adapted or learned during the task

Hitzler, K., Meier, F., Schaal, S. and Asfour, T., *Learning and Adaptation of Inverse Dynamics Models: A Comparison*, IEEE/RAS International Conference on Humanoid Robots (Humanoids), October, 2019

Learning of Kinematics and Dynamics