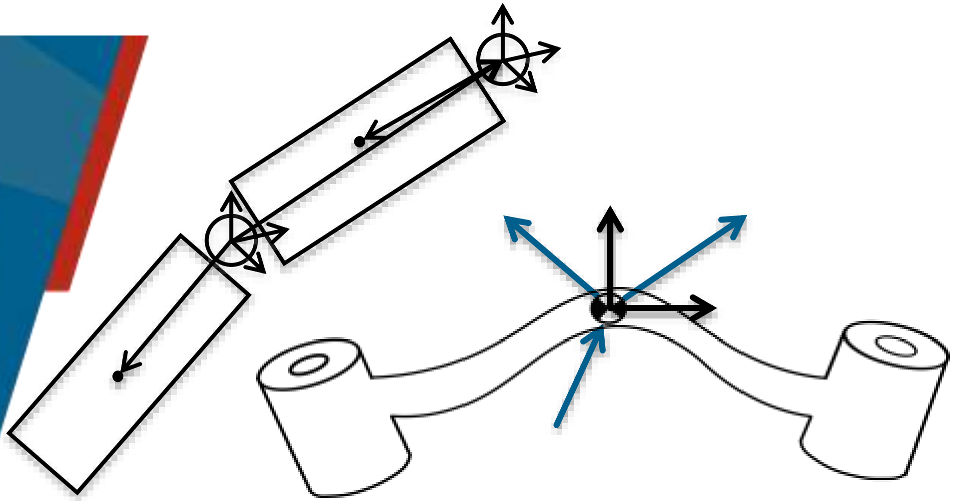
Robot Modelling IV



Prof. Karsten Berns
Robotics Research Lab
Department of Computer Science
University of Kaiserslautern, Germany

Content

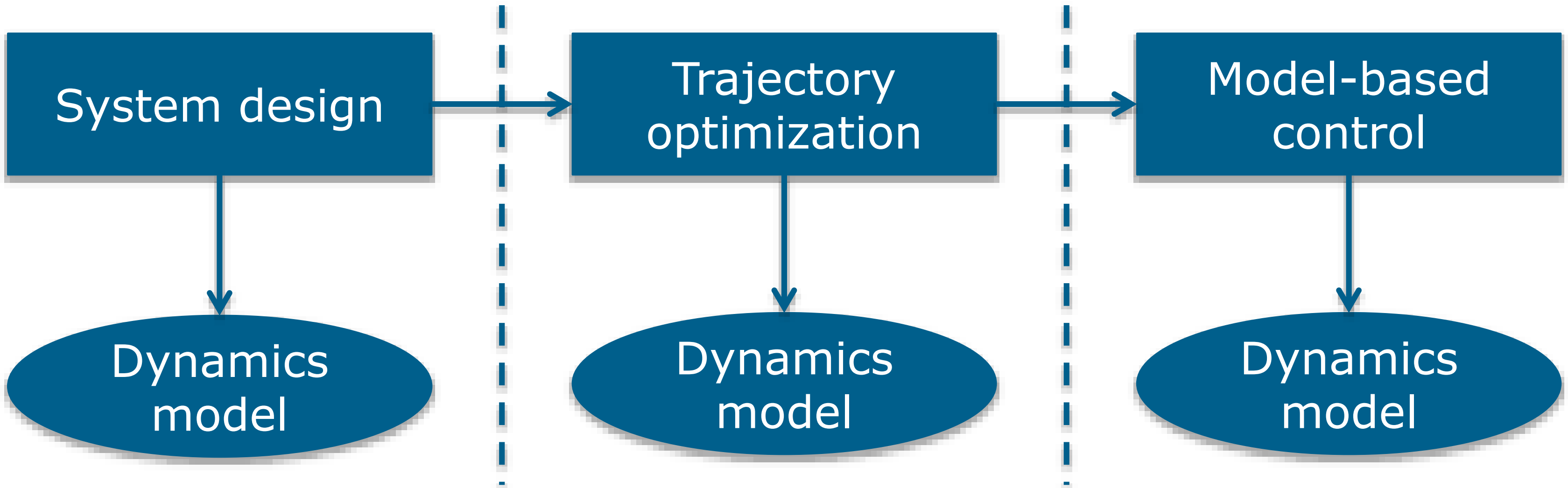
- Utilization of the dynamic model
- Linear and angular accelerations
- Moment of inertia
- Calculation of dynamics via Newton-Euler
- Calculation of dynamics via Lagrange
- Comparison of approaches

Dynamics Modelling

- Calculates the relations between forces, torques and motions which occur in a mechanical multi-body system
- Applications
  - Analysis of dynamics
  - Synthesis of mechanical structures
  - Modelling of elastic structures
  - Controller design

Dynamics Modelling: Application

- Phases of robot development and operation



- Modelling in different phases
- High effort; errors and inconsistencies probable
- Reusability of model's code difficult if structure changes (kinematic structure, joint types, actuators)

Dynamics Modelling: Equations of Movement

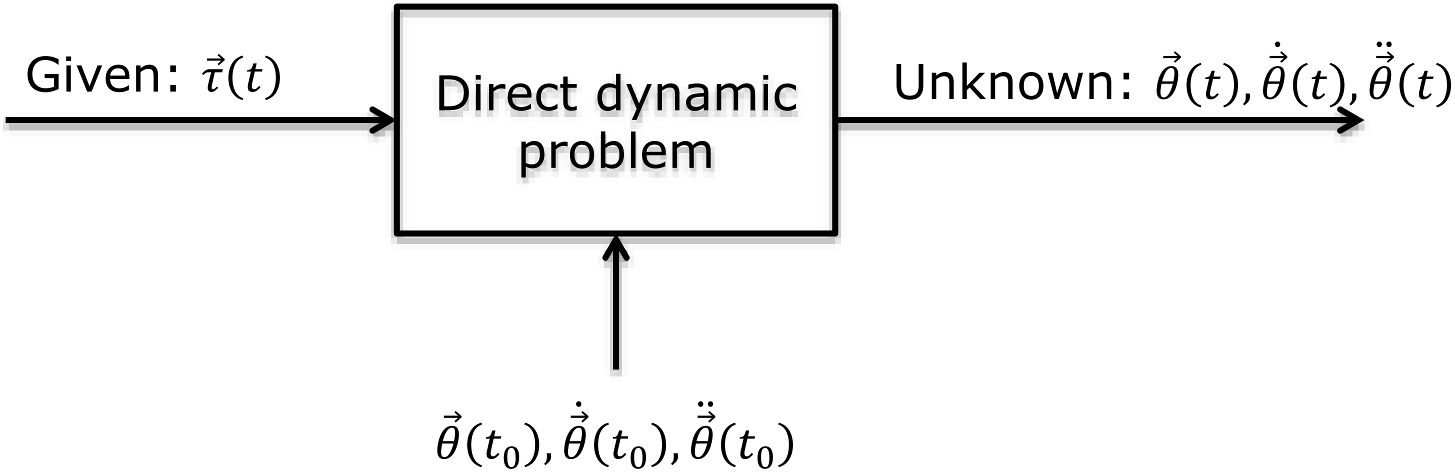
- Relation between forces/torques, poses, velocities and accelerations of the n links

$$\vec{\tau} = M(\vec{\theta}) \cdot \ddot{\vec{\theta}} + n\left(\dot{\vec{\theta}}, \vec{\theta}\right) + g(\vec{\theta}) + R \cdot \dot{\vec{\theta}} \tag{8.1}$$

- $\vec{\tau}$: $n \times 1$ vector of general actuating forces and torques
- $M(\vec{\theta})$: $n \times n$ moment of inertia matrix
- $n\left(\dot{\vec{\theta}}, \vec{\theta}\right)$: $n \times 1$ vector with centrifugal and Coriolis components
- $g(\vec{\theta})$: $n \times 1$ vector with gravitational components
- R: $n \times n$ diagonal matrix describing friction forces
- $\vec{\theta}$: $n \times 1$ manipulator variables

Direct Dynamic Problem

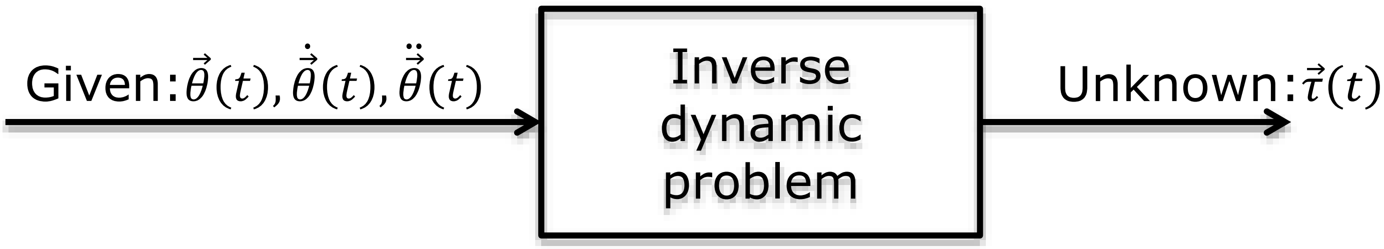
- Given the mass, external forces and torques, as well as pose, initial velocity and accelerations the resulting difference of motion is calculated



- Solve Eq. (8.1) for: $\vec{\theta}(t), \dot{\vec{\theta}}(t), \ddot{\vec{\theta}}(t)$

Inverse Dynamic Problem

- From desired parameters of motion and kinematics, determine the required actuation forces and torques



- Calculate Eq. (8.1)

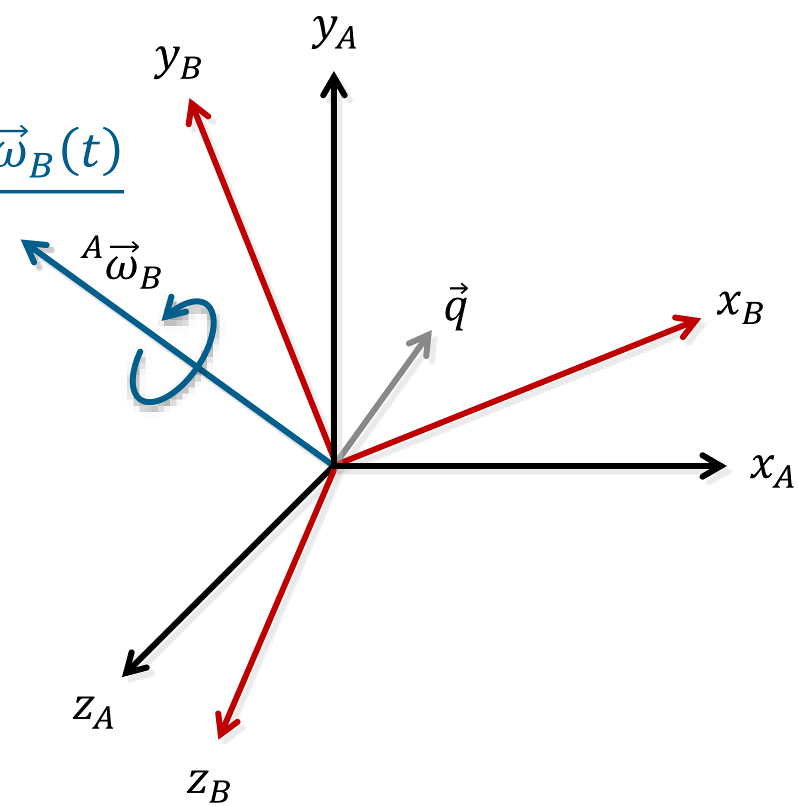
Acceleration of Rigid Bodies

- Linear acceleration

$$ {}^{B}\dot{\vec{v}}_q = \frac{d}{dt}\, {}^{B}\vec{v}_q = \lim_{\Delta t \to 0} \frac{ {}^{B}\vec{v}_q(t+\Delta t) - {}^{B}\vec{v}_q(t)}{\Delta t} $$

- Angular acceleration

$$ {}^{A}\dot{\vec{\omega}}_B = \frac{d}{dt}\, {}^{A}\vec{\omega}_B = \lim_{\Delta t \to 0} \frac{ {}^{A}\vec{\omega}_B(t+\Delta t) - {}^{A}\vec{\omega}_B(t)}{\Delta t} $$



Linear Acceleration

- Linear acceleration based on velocity

$$ {}^{A}\vec{v}_q = {}^{A}_{B}R \cdot {}^{B}\vec{v}_q + {}^{A}\vec{\omega}_B \times {}^{A}_{B}R \cdot {}^{B}\vec{q} \qquad \text{Compare } (7.1) $$

- Because the origins of frames A and B coincide, it follows:

$$ \frac{d}{dt}\left({}^{A}_{B}R \cdot {}^{B}\vec{q}\right) = {}^{A}_{B}R \cdot {}^{B}\vec{v}_q + {}^{A}\vec{\omega}_B \times {}^{A}_{B}R \cdot {}^{B}\vec{q} \qquad (8.2) $$

- Derivative of velocity: Linear acceleration

$$ {}^{A}\dot{\vec{v}}_q = \frac{d}{dt}\left({}^{A}_{B}R \cdot {}^{B}\vec{v}_q\right) + {}^{A}\dot{\vec{\omega}}_B \times {}^{A}_{B}R \cdot {}^{B}\vec{q} + {}^{A}\vec{\omega}_B \times \frac{d}{dt}\left({}^{A}_{B}R \cdot {}^{B}\vec{q}\right) \qquad (8.3) $$

- Substituting (8.2) into (8.3)

$$ \begin{aligned} {}^{A}\dot{\vec{v}}_q = {}& {}^{A}_{B}R \cdot {}^{B}\dot{\vec{v}}_q + {}^{A}\vec{\omega}_B \times {}^{A}_{B}R \cdot {}^{B}\vec{v}_q + {}^{A}\dot{\vec{\omega}}_B \times {}^{A}_{B}R \cdot {}^{B}\vec{q} \\ & + {}^{A}\vec{\omega}_B \times \left({}^{A}_{B}R \cdot {}^{B}\vec{v}_q + {}^{A}\vec{\omega}_B \times {}^{A}_{B}R \cdot {}^{B}\vec{q}\right) \end{aligned} $$

Linear Acceleration

- Simplification

$$
\begin{aligned}
{}^{A}\dot{\vec{v}}_q = {}^{A}_{B}R \cdot {}^{B}\dot{\vec{v}}_q + 2 \cdot \left({}^{A}\vec{\omega}_B \times {}^{A}_{B}R \cdot {}^{B}\vec{v}_q \right) \\
+ {}^{A}\dot{\vec{\omega}}_B \times {}^{A}_{B}R \cdot {}^{B}\vec{q} + {}^{A}\vec{\omega}_B \times \left({}^{A}\vec{\omega}_B \times {}^{A}_{B}R \cdot {}^{B}\vec{q} \right)
\end{aligned}
$$

- General case (frames A, B without common origin)

$$
\begin{aligned}
{}^{A}\dot{\vec{v}}_q = {}^{A}\dot{\vec{v}}_{OB} + {}^{A}_{B}R \cdot {}^{B}\dot{\vec{v}}_q + 2 \cdot \left({}^{A}\vec{\omega}_B \times {}^{A}_{B}R \cdot {}^{B}\vec{v}_q \right) \\
+ {}^{A}\dot{\vec{\omega}}_B \times {}^{A}_{B}R \cdot {}^{B}\vec{q} + {}^{A}\vec{\omega}_B \times \left({}^{A}\vec{\omega}_B \times {}^{A}_{B}R \cdot {}^{B}\vec{q} \right)
\end{aligned}
$$

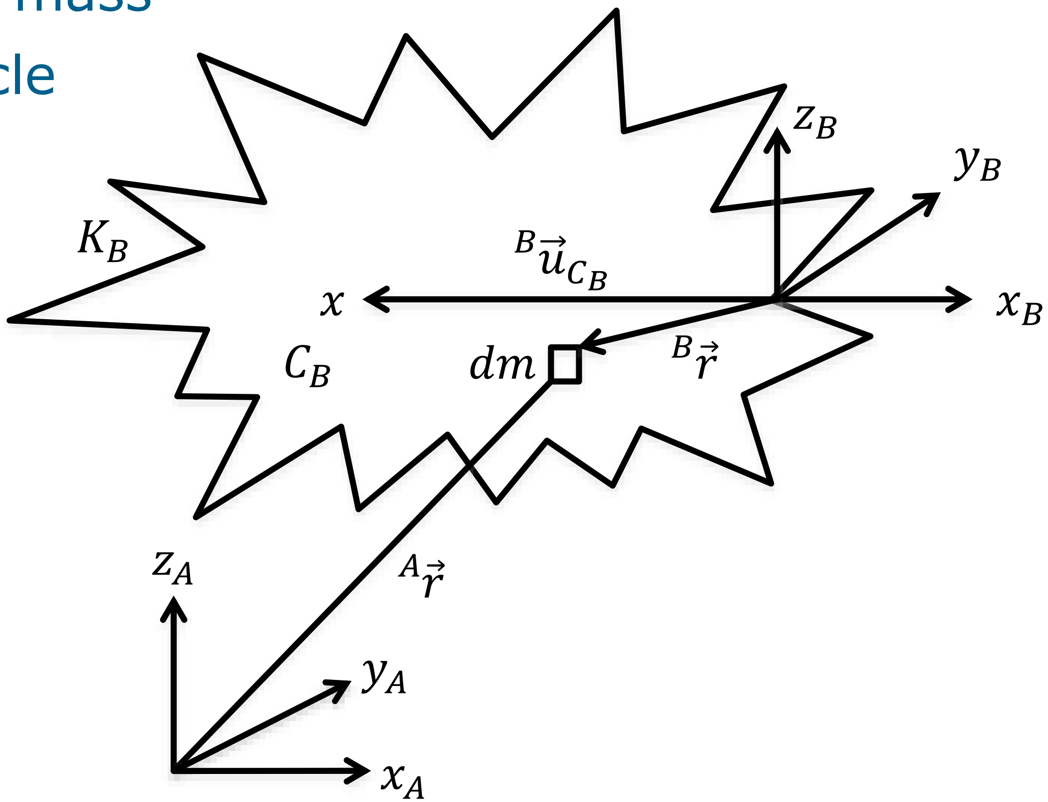
- Considering ${}^{B}\vec{q}$ does not move

$$
{}^{B}\vec{v}_q = {}^{B}\dot{\vec{v}}_q = \vec{0}
$$

$$
\Rightarrow {}^{A}\dot{\vec{v}}_q = {}^{A}\dot{\vec{v}}_{OB} + {}^{A}\dot{\vec{\omega}}_B \times {}^{A}_{B}R \cdot {}^{B}\vec{q} + {}^{A}\vec{\omega}_B \times \left({}^{A}\vec{\omega}_B \times {}^{A}_{B}R \cdot {}^{B}\vec{q} \right)
$$

Distribution of Mass: Geometric Pre-Examination

- dm: Mass particle
- C_B: Center of mass of body K_B
- $\vec{u}_{C_B}$: Vector to center of mass
- $\vec{r}$: Vector to mass particle



Distribution of Mass : Inertia Tensor

- Inertia tensor in reference to frame A, specifying the body‘s inertia regarding rotation

$$ {}^{A}I = \begin{bmatrix} {}^{A}i_{xx} & -{}^{A}i_{xy} & -{}^{A}i_{xz} \\ -{}^{A}i_{xy} & {}^{A}i_{yy} & -{}^{A}i_{yz} \\ -{}^{A}i_{xz} & -{}^{A}i_{yz} & {}^{A}i_{zz} \end{bmatrix} $$

- Scalar elements of inertia tensor (calculation through integration of mass distribution M)
  - Axial moments of inertia

$$ {}^{A}i_{xx} = \iiint_M (y_A^2 + z_A^2)\,dm \qquad {}^{A}i_{yy} = \iiint_M \left(x_A^2 + z_A^2\right)dm \qquad {}^{A}i_{zz} = \iiint_M \left(x_A^2 + y_A^2\right)dm $$

  - Inertia products

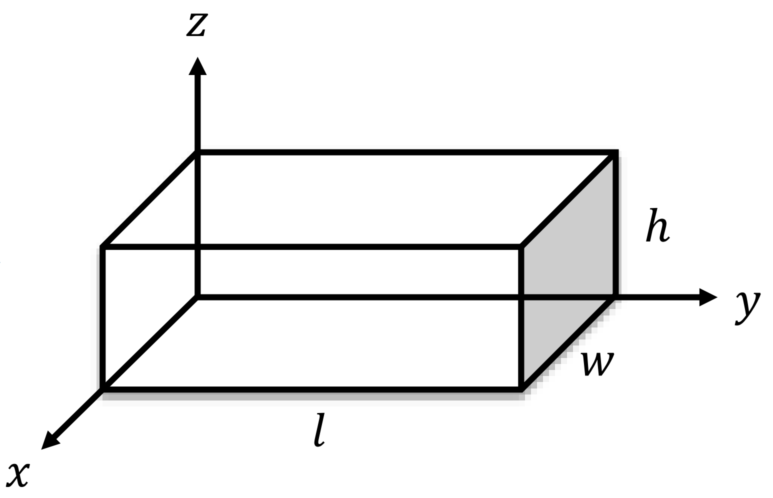
$$ {}^{A}i_{xy} = \iiint_M x_A y_A\,dm \qquad {}^{A}i_{xz} = \iiint_M x_A z_A\,dm \qquad {}^{A}i_{yz} = \iiint_M y_A z_A\,dm $$

- For a point mass the tensor becomes a zero matrix

Distribution of Mass : Example Cuboid

- Calculation of inertia tensor for cuboid with uniform density ρ
- With $dm = \rho dx dy dz$ it follows:

$$
\begin{aligned}
{}^{A}i_{xx} &= \int_0^h \int_0^l \int_0^w (y_A^2 + z_A^2)\rho dx_A dy_A dz_A \\
&= \int_0^h \int_0^l (y_A^2 + z_A^2) w\rho dy_A dz_A \\
&= \int_0^h \left(\frac{l^3}{3} + z_A^2 l\right) w\rho dz_A \\
&= \left(\frac{hl^3w}{3} + \frac{h^3lw}{3}\right)\rho \\
&= \frac{m}{3}(l^2 + h^2) \quad \text{(with total mass } m\text{)}
\end{aligned}
$$



- For ${}^{A}i_{yy}$ and ${}^{A}i_{zz}$ it follows analogously:

$$ {}^{A}i_{yy} = \frac{m}{3}(w^2 + h^2) $$

$$ {}^{A}i_{zz} = \frac{m}{3}(l^2 + w^2) $$

Distribution of Mass : Example Cuboid

- Calculation of

$$
\begin{aligned}
{}^{A}i_{xy} &= \int_0^h \int_0^l \int_0^w x_A y_A \rho dx_A dy_A dz_A \\
&= \int_0^h \int_0^l \frac{w^2}{2} y_A \rho dy_A dz_A = \int_0^h \frac{w^2 l^2}{4} \rho dz_A = \frac{m}{4} wl
\end{aligned}
$$

- Analogous computation of ${}^{A}i_{xz} = \frac{m}{4} hw$, ${}^{A}i_{yz} = \frac{m}{4} hl$
- Inertia tensor

$$
{}^{A}I = \begin{bmatrix} \frac{m}{3}(l^2 + h^2) & -\frac{m}{4} wl & -\frac{m}{4} hw \\ -\frac{m}{4} wl & \frac{m}{3}(w^2 + h^2) & -\frac{m}{4} hl \\ -\frac{m}{4} hw & -\frac{m}{4} hl & \frac{m}{3}(l^2 + w^2) \end{bmatrix}
$$

Steiner's Theorem

- For parallel axes through the center of mass
- For arbitrary frame A and frame C with origin in center of mass and axes parallel to frame A, the following holds:
  - ${}^{A}i_{zz} = {}^{C}i_{zz} + m \cdot \left({}^{A}u_{C_x}^{2} + {}^{A}u_{C_y}^{2} \right)$
  - ${}^{A}i_{xy} = {}^{C}i_{xy} - m \cdot {}^{A}u_{C_x} \cdot {}^{A}u_{C_y}$
- With position vector ${}^{A}\vec{u}_{C} = \left({}^{A}u_{C_x}, {}^{A}u_{C_y}, {}^{A}u_{C_z} \right)^{T}$
- Remaining scalars follow analogously

Distribution of Mass : Example Cuboid

- Steiner's theorem in matrix notation:

$$^{A}I = {}^{C}I + m \cdot \left[{}^{A}\vec{u}_C^T \cdot {}^{A}\vec{u}_C \cdot E_3 - {}^{A}\vec{u}_C^T \cdot {}^{A}\vec{u}_C \right]$$

  - With $E_3 = 3 \times 3$ identity matrix
- Applied to cuboid example

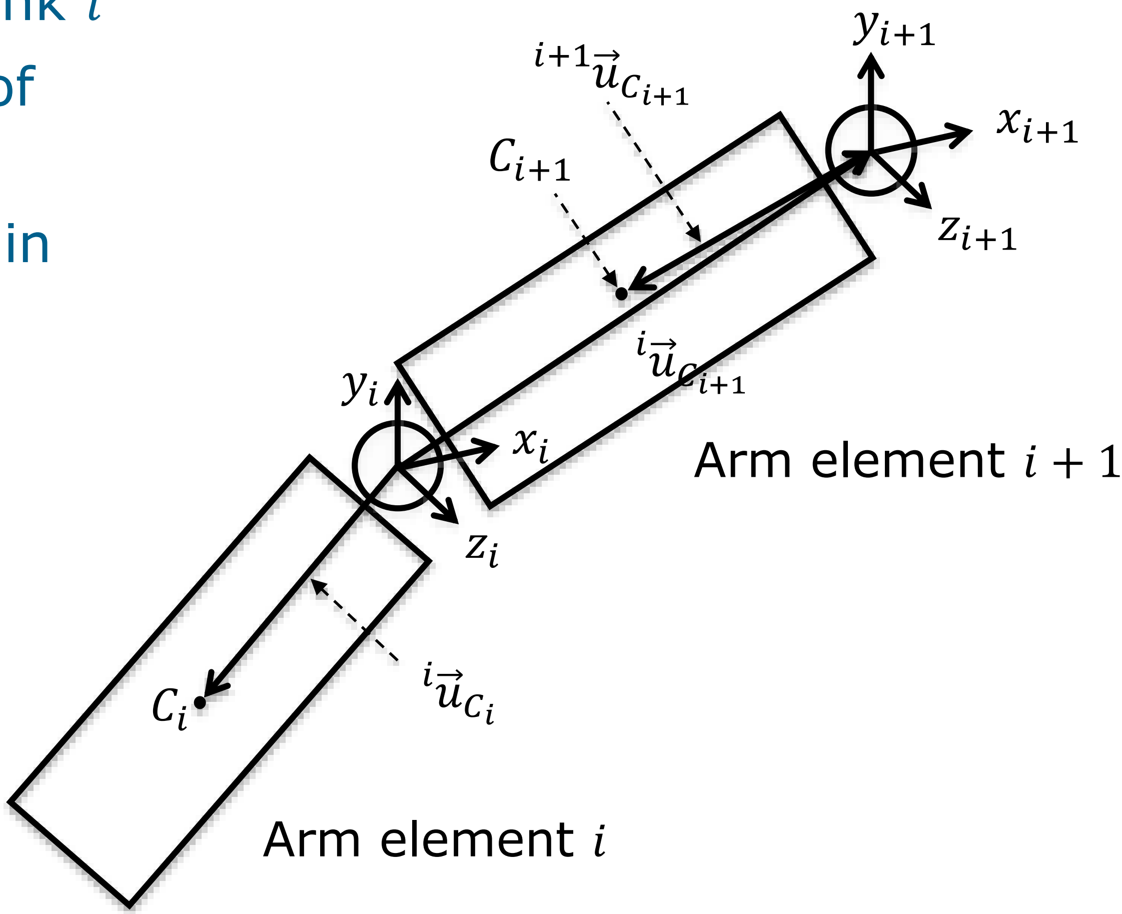
$$^{A}\vec{u}_C = \begin{pmatrix} {}^{A}u_{C_x} \\ {}^{A}u_{C_y} \\ {}^{A}u_{C_z} \end{pmatrix} = \frac{1}{2}\begin{pmatrix} w \\ l \\ h \end{pmatrix} \qquad {}^{C}i_{zz} = \frac{m}{12} \cdot (w^2 + l^2) \qquad {}^{C}i_{xy} = 0$$

- The remaining elements follow from symmetry considerations. Resulting inertia tensor:

$$^{C}I = \begin{bmatrix} \frac{m}{12} \cdot (h^2 + l^2) & 0 & 0 \\ 0 & \frac{m}{12} \cdot (w^2 + h^2) & 0 \\ 0 & 0 & \frac{m}{12} \cdot (l^2 + w^2) \end{bmatrix}$$

Geometric Description of Neighboring Arm Elements

- C_i: Center of mass of link i
- ${}^{i}\vec{u}_{C_i}$: Vector to center of mass of link i in CS i
- ${}^{i}\vec{u}_{i+1}$: Vector from origin i to $i+1$ in CS i

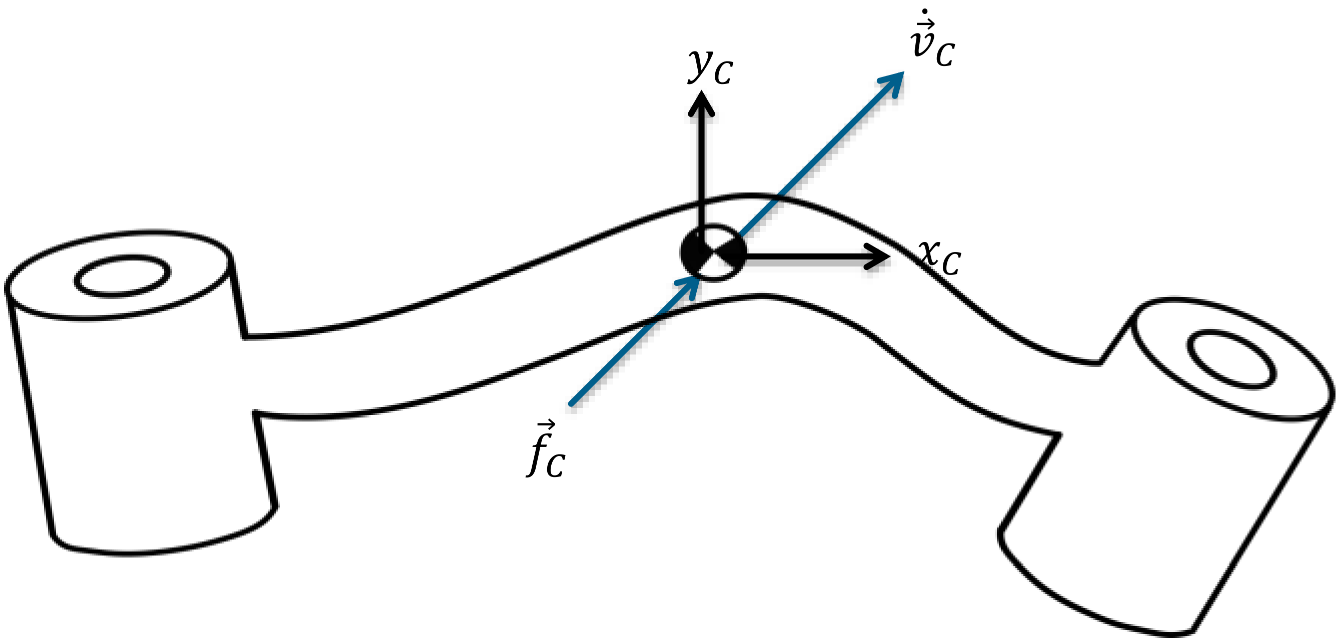


Derivation of Equations of Motion

- Synthetic method (Newton-Euler):
  Free body diagram
  - Conservation of (angular) momentum
  - Elimination of constraining forces results in equations of motion
- Analytic methods (Lagrange):
  Application of extremal principles
  - Work and energy considerations
  - Formal derivation yields equations of movement

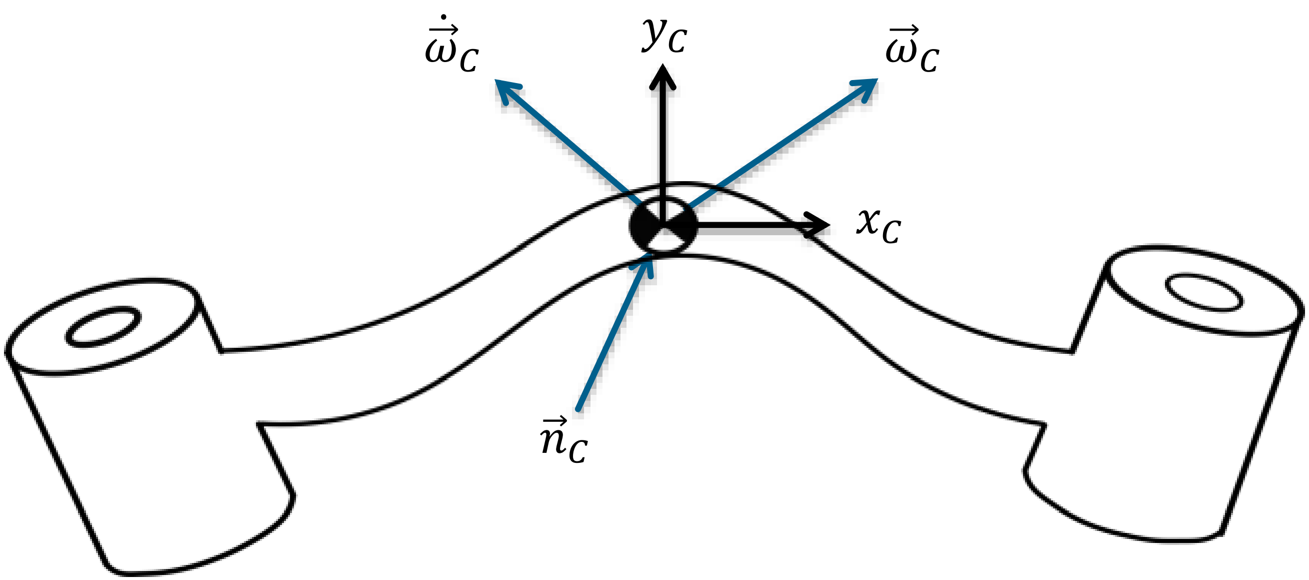
Newton-Euler Method: Fundamental Equations

- Newton equation: $\vec{f}_C = m \cdot \dot{\vec{v}}_C$
  - m: Total mass of body
  - $\dot{\vec{v}}_C$: Acceleration in center of mass C
  - $\vec{f}_C$: Force acting on the center



Newton-Euler Method: Fundamental Equations

- Euler equation: $\vec{n}_C = {}^{C}I \cdot \dot{\vec{\omega}}_C + \vec{\omega}_C \times {}^{C}I \cdot \vec{\omega}_C$
  - $\vec{\omega}_C$: Body‘s angular velocity
  - ${}^{C}I$: Inertia tensor in frame C (center of mass)
  - $\vec{n}_C$: Torque in center, causing the rotation



Newton-Euler Method

- Iterative determination of velocities and accelerations in order to calculate the segments‵ mass forces
- Rotational velocity of element $i + 1$

$$ {}^{i+1}\vec{\omega}_{i+1} = {}^{i+1}_{i}R \cdot \left({}^{i}\vec{\omega}_{i} + \dot{\theta}_{i+1} \cdot {}^{i}\vec{e}_{z_i} \right) $$

$$ {}^{i}_{i+1}R \cdot {}^{i+1}\vec{\omega}_{i+1} = {}^{i}\vec{\omega}_{i} + \dot{\theta}_{i+1} \cdot {}^{i}\vec{e}_{z_i} $$

Newton-Euler Method

- For rotational acceleration the following applies

Rotation matrix ${}_{i+1}^{i}R$ dependent on $\vec{\theta}$ and thus time dependent

$$\frac{d}{dt}\left({}_{i+1}^{i}R \cdot {}^{i+1}\vec{\omega}_{i+1}\right) = \frac{d}{dt}\left({}^{i}\vec{\omega}_i + \dot{\theta}_{i+1} \cdot {}^{i}\vec{e}_{z_i}\right)$$

$$\left(\frac{d}{dt}\,{}_{i+1}^{i}R\right) \cdot {}^{i+1}\vec{\omega}_{i+1} + {}_{i+1}^{i}R \cdot {}^{i+1}\dot{\vec{\omega}}_{i+1} = {}^{i}\dot{\vec{\omega}}_i + \ddot{\theta}_{i+1} \cdot {}^{i}\vec{e}_{z_i}$$ (8.2) follows

$$\dot{\theta}_{i+1} \cdot {}^{i}\vec{e}_{z_i} \times {}_{i+1}^{i}R \cdot {}^{i+1}\vec{\omega}_{i+1} + {}_{i+1}^{i}R \cdot {}^{i+1}\dot{\vec{\omega}}_{i+1} = {}^{i}\dot{\vec{\omega}}_i + \ddot{\theta}_{i+1} \cdot {}^{i}\vec{e}_{z_i}$$

$${}_{i+1}^{i}R \cdot {}^{i+1}\dot{\vec{\omega}}_{i+1} = {}^{i}\dot{\vec{\omega}}_i + \ddot{\theta}_{i+1} \cdot {}^{i}\vec{e}_{z_i} - \dot{\theta}_{i+1} \cdot {}^{i}\vec{e}_{z_i} \times {}_{i+1}^{i}R \cdot {}^{i+1}\vec{\omega}_{i+1}$$

$${}_{i+1}^{i}R \cdot {}^{i+1}\dot{\vec{\omega}}_{i+1} = {}^{i}\dot{\vec{\omega}}_i + \ddot{\theta}_{i+1} \cdot {}^{i}\vec{e}_{z_i} + {}^{i}\vec{\omega}_{i+1} \times \dot{\theta}_{i+1} \cdot {}^{i}\vec{e}_{z_i}$$

$${}^{i+1}\dot{\vec{\omega}}_{i+1} = {}_{i}^{i+1}R \cdot \left({}^{i}\dot{\vec{\omega}}_i + \ddot{\theta}_{i+1} \cdot {}^{i}\vec{e}_{z_i} + \left({}^{i}\vec{\omega}_i + \dot{\theta}_{i+1} \cdot {}^{i}\vec{e}_{z_i}\right) \times \dot{\theta}_{i+1} \cdot {}^{i}\vec{e}_{z_i}\right)$$

$$\boxed{{}^{i+1}\dot{\vec{\omega}}_{i+1} = {}_{i}^{i+1}R \cdot \left({}^{i}\dot{\vec{\omega}}_i + \ddot{\theta}_{i+1} \cdot {}^{i}\vec{e}_{z_i} + {}^{i}\vec{\omega}_i \times \dot{\theta}_{i+1} \cdot {}^{i}\vec{e}_{z_i}\right)}$$

- Simplification for linear joints: ${}^{i+1}\dot{\vec{\omega}}_{i+1} = {}_{i}^{i+1}R \cdot {}^{i}\dot{\vec{\omega}}_i$

Newton-Euler Method

- Linear velocity of element $i+1$

$$ {}^{i+1}\vec{v}_{i+1} = {}^{i+1}_{i}R \cdot \left({}^{i}\vec{v}_i + {}^{i}\vec{\omega}_{i+1} \times {}^{i}\vec{u}_{i+1} + \dot{d}_{i+1}\ {}^{i}\vec{e}_{z_i} \right) $$

- Linear acceleration in link origin

$$ {}^{i+1}\dot{\vec{v}}_{i+1} = {}^{i+1}_{i}R \cdot \Big({}^{i}\dot{\vec{v}}_i + \ddot{d}_{i+1} \cdot {}^{i}\vec{e}_{z_i} + {}^{i}\dot{\vec{\omega}}_{i+1} \times {}^{i}\vec{u}_{i+1} $$
$$ + {}^{i}\vec{\omega}_{i+1} \times \left({}^{i}\vec{\omega}_{i+1} \times {}^{i}\vec{u}_{i+1} \right) + 2\ {}^{i}\vec{\omega}_{i+1} \times \left({}^{i}\vec{e}_{z_i} \dot{d}_{i+1} \right) \Big) $$

- Simplification for revolute joint

$$ {}^{i+1}\dot{\vec{v}}_{i+1} = {}^{i+1}_{i}R \cdot \Big({}^{i}\dot{\vec{v}}_i + {}^{i}\dot{\vec{\omega}}_{i+1} \times {}^{i}\vec{u}_{i+1} + {}^{i}\vec{\omega}_{i+1} \times \left({}^{i}\vec{\omega}_{i+1} \times {}^{i}\vec{u}_{i+1} \right) \Big) $$

- Linear acceleration in center of mass

$$ {}^{i}\dot{\vec{v}}_{C_i} = {}^{i}\dot{\vec{v}}_i + {}^{i}\dot{\vec{\omega}}_i \times {}^{i}\vec{u}_{C_i} + {}^{i}\vec{\omega}_i \times \left({}^{i}\vec{\omega}_i \times {}^{i}\vec{u}_{C_i} \right) $$

Newton-Euler Method

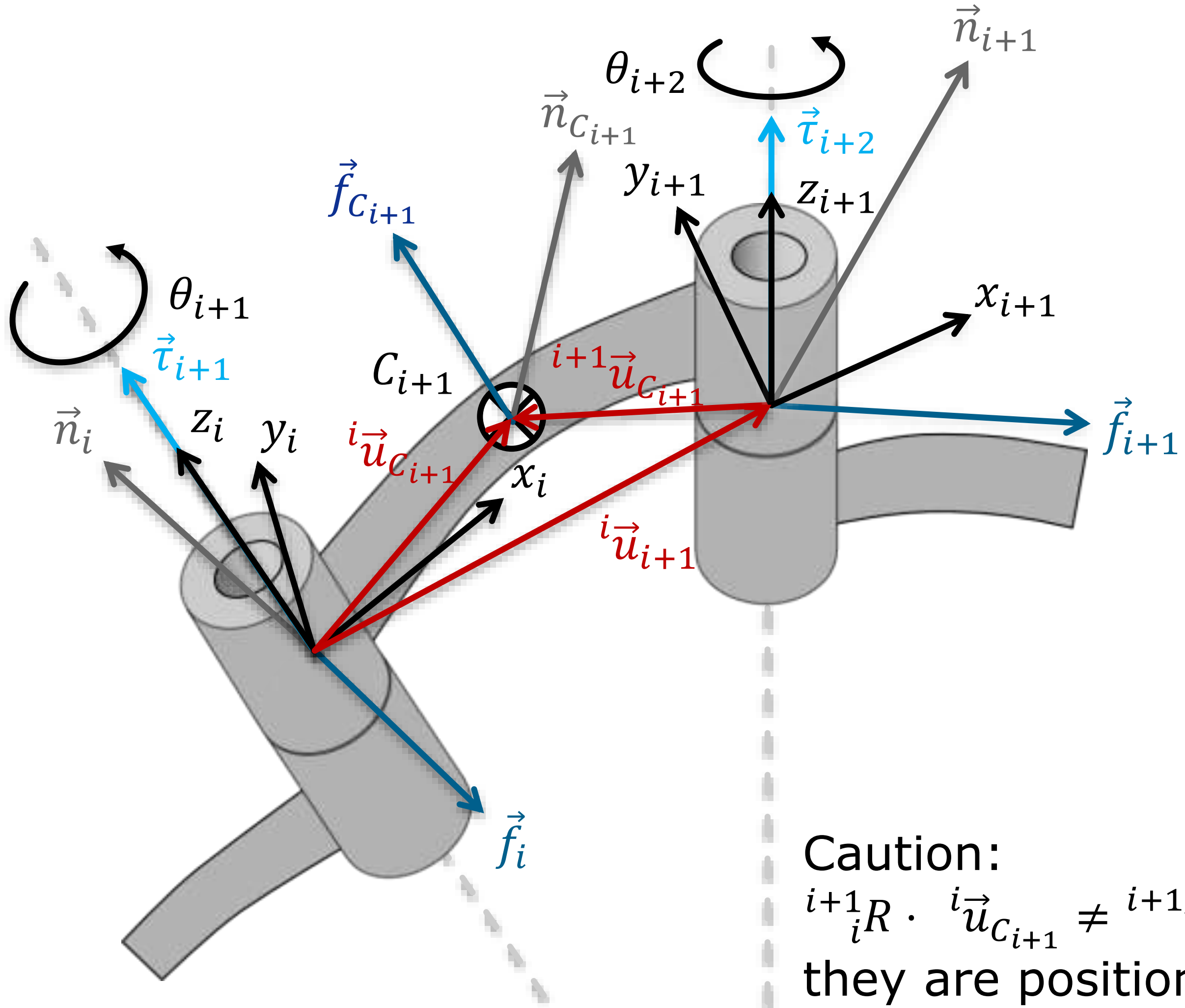
- Calculation of first link: ${}^{0}\vec{\omega}_0 = {}^{0}\dot{\vec{\omega}}_0 = \vec{0}$
- With linear and angular accelerations in the centers of mass the following forces and torques result:

$$\vec{f}_{C_i} = m_i \cdot \dot{\vec{v}}_{C_i}$$

$$\vec{n}_{C_i} = {}^{C_i}I \cdot \dot{\vec{\omega}}_i + \vec{\omega}_i \times {}^{C_i}I \cdot \vec{\omega}_i$$

- Forces and torques equilibrium for each link
  - Consideration of own mass force and inertia
  - Consideration of forces and torques enacted by neighboring links
- $\vec{f}_i$: Force enacted upon link i by link $i+1$
- $\vec{n}_i$: Torque enacted upon link i by link $i+1$

Coordinate Systems and Designators



Caution:
${}^{i+1}_{\ \ i}R \cdot {}^{i}\vec{u}_{C_{i+1}} \neq {}^{i+1}\vec{u}_{C_{i+1}}$ because they are position vectors

Newton-Euler Method

- Force equilibrium in joint i

$$ {}^{i}\vec{f}_{i} = {}^{i}\vec{f}_{C_{i}+1} + {}_{i+1}^{\;\;i}R \cdot {}^{i+1}\vec{f}_{i+1} $$

- Torque equilibrium

$$ {}^{i}\vec{n}_{i} = {}^{i}\vec{n}_{C_{i+1}} + {}_{i+1}^{\;\;i}R \cdot {}^{i+1}\vec{n}_{i+1} + {}^{i}\vec{u}_{C_{i+1}} \times {}^{i}\vec{f}_{C_{i+1}} + {}^{i}\vec{u}_{i+1} \times {}_{i+1}^{\;\;i}R $$

- Calculation proceeds from last joint to base („backwards“)

Newton-Euler Method

- For calculation of the forces required in joint i, only the z component is used

$$\tau_{i+1} = {}^{i}\vec{n}_i^T \cdot {}^{i}\vec{e}_{z_i}$$

- Linear force for linear joints

$$\tau_{i+1} = {}^{i}\vec{f}_i^T \cdot {}^{i}\vec{e}_{z_i}$$

- In free space the initial forces and torques are set to 0:

$$\vec{f}_N = \vec{n}_N = \vec{0}$$

- (If contact with environment or existing load -> ≠ 0)

Newton-Euler Method: Algorithm for Calculation of Torques

1. Iterative calculation of velocities and accelerations starting from first link (outer iteration)

$$^{i+1}\vec{\omega}_{i+1} = {}^{i+1}_{\ i}R \cdot \left({}^{i}\vec{\omega}_i + \dot{\theta}_{i+1} \cdot {}^{i}\vec{e}_{z_i} \right)$$

$$^{i+1}\dot{\vec{\omega}}_{i+1} = {}^{i+1}_{\ i}R \cdot \left({}^{i}\dot{\vec{\omega}}_i + \ddot{\theta}_{i+1} \cdot {}^{i}\vec{e}_{z_i} + {}^{i}\vec{\omega}_i \times \dot{\theta}_{i+1} \cdot {}^{i}\vec{e}_{z_i} \right)$$

$$^{i+1}\dot{\vec{v}}_{i+1} = {}^{i+1}_{\ i}R \cdot \Big({}^{i}\dot{\vec{v}}_i + \ddot{d}_{i+1} \cdot {}^{i}\vec{e}_{z_i} + {}^{i}\dot{\vec{\omega}}_{i+1} \times {}^{i}\vec{u}_{i+1}$$
$$+ {}^{i}\vec{\omega}_{i+1} \times \left({}^{i}\vec{\omega}_{i+1} \times {}^{i}\vec{u}_{i+1} \right) + 2\, {}^{i}\vec{\omega}_{i+1} \times \left({}^{i}\vec{e}_{z_i} \dot{d}_{i+1} \right) \Big)$$

$$^{i}\dot{\vec{v}}_{C_i} = {}^{i}\dot{\vec{v}}_i + {}^{i}\dot{\vec{\omega}}_i \times {}^{i}\vec{u}_{C_i} + {}^{i}\vec{\omega}_i \times \left({}^{i}\vec{\omega}_i \times {}^{i}\vec{u}_{C_i} \right)$$

$$^{i}\vec{f}_{C_i} = m_i \cdot {}^{i}\dot{\vec{v}}_{C_i}$$

$$^{i}\vec{n}_{C_i} = {}^{C_i}I \cdot {}^{i}\dot{\vec{\omega}}_i + {}^{i}\vec{\omega}_i \times {}^{C_i}I \cdot {}^{i}\vec{\omega}_i$$

Newton-Euler Method: Algorithm for Calculation of Torques

- If gravity is considered, then:

$$ {}^{0}\dot{\vec{v}}_0 = \vec{g}' $$

  - g' in opposite direction of gravitation vector
  - Corresponds to acceleration of robot base by $1g$ upward

2. Backward calculation of forces and torques starting from last link and ending in robot base (inner iteration)

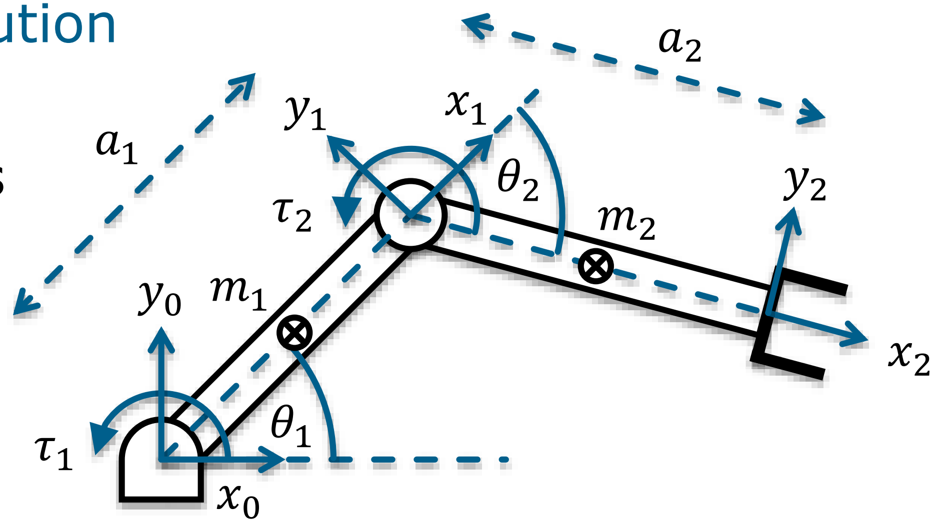
$$ {}^{i}\vec{f}_i = {}^{i}\vec{f}_{C_{i+1}} + {}^{i}_{i+1}R \cdot {}^{i+1}\vec{f}_{i+1} $$

$$ {}^{i}\vec{n}_i = {}^{i}\vec{n}_{C_{i+1}} + {}^{i}_{i+1}R \cdot {}^{i+1}\vec{n}_{i+1} + {}^{i}\vec{u}_{C_{i+1}} \times {}^{i}\vec{f}_{C_{i+1}} + {}^{i}\vec{u}_{i+1} \times {}^{i}_{i+1}R \cdot {}^{i+1}\vec{f}_{i+1} $$

$$ \tau_{i+1} = {}^{i}\vec{n}_i^T \cdot {}^{i}\vec{e}_{z_i} \text{ bzw. } \tau_{i+1} = {}^{i}\vec{f}_i^T \cdot {}^{i}\vec{e}_{z_i} $$

Newton-Euler Method: Example

- Example of a closed-form solution
  - Two-joint robot
  - Simplification: Point masses m_1, m_2 in link centers



Procedure

- Determining known values
- Determining rotation matrices between links
- Outer iteration (velocity, acceleration)
  - For joint 1, 2
- Inner iteration (forces, torques)
  - For joint 2, 1

Newton-Euler Method: Example

- Determining known values
  - Vectors to centers of mass

$$ {}^{1}\vec{u}_{C_1} = -\frac{a_1}{2} \cdot {}^{1}\vec{e}_{x_1}, \; {}^{2}\vec{u}_{C_2} = -\frac{a_2}{2} \cdot {}^{2}\vec{e}_{x_2} $$

  - Inertia tensor (because of point mass)

$$ {}^{C_1}I_1 = {}^{C_2}I_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} $$

  - No forces acting on TCP: $\vec{f}_2 = \vec{0}$, $\vec{n}_2 = \vec{0}$
  - No movement of robot base: $\vec{\omega}_0 = \vec{0}$, $\dot{\vec{\omega}}_0 = \vec{0}$
  - Consideration of gravity: ${}^{0}\dot{\vec{v}}_0 = g \cdot {}^{0}\vec{e}_{y_0}$

Newton-Euler Method: Example

- Vector to next coordinate system

$$ {}^{0}\vec{u}_1 = \begin{pmatrix} c_1 a_1 \\ s_1 a_1 \\ 0 \end{pmatrix}, \; {}^{1}\vec{u}_2 = \begin{pmatrix} c_2 a_2 \\ s_2 a_2 \\ 0 \end{pmatrix} $$

2. Rotation matrices between joint-frames (see chapter VII)

$$ {}^{i}_{i+1}R = \begin{bmatrix} \cos(\theta_{i+1}) & -\sin(\theta_{i+1}) & 0 \\ \sin(\theta_{i+1}) & \cos(\theta_{i+1}) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_{i+1} & -s_{i+1} & 0 \\ s_{i+1} & c_{i+1} & 0 \\ 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{i+1}_{i}R = \begin{bmatrix} \cos(\theta_{i+1}) & \sin(\theta_{i+1}) & 0 \\ -\sin(\theta_{i+1}) & \cos(\theta_{i+1}) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_{i+1} & s_{i+1} & 0 \\ -s_{i+1} & c_{i+1} & 0 \\ 0 & 0 & 1 \end{bmatrix} $$

Newton-Euler Method: Example

- Outer iteration (1st step)

$$ {}^{1}\vec{\omega}_1 = {}^{1}_{0}R \cdot \left({}^{0}\vec{\omega}_0 + \dot{\theta}_1 \cdot {}^{0}\vec{e}_{z_0} \right) = \vec{0} + \begin{pmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{pmatrix} $$

$$ {}^{1}\dot{\vec{\omega}}_1 = {}^{1}_{0}R \cdot \left({}^{0}\dot{\vec{\omega}}_0 + \ddot{\theta}_1 \cdot {}^{0}\vec{e}_{z_0} + {}^{0}\vec{\omega}_0 \times \dot{\theta}_1 \cdot {}^{0}\vec{e}_{z_0} \right) = \vec{0} + \begin{pmatrix} 0 \\ 0 \\ \ddot{\theta}_1 \end{pmatrix} + \vec{0} $$

$$ {}^{1}\dot{\vec{v}}_1 = {}^{1}_{0}R \cdot \left({}^{0}\dot{\vec{v}}_0 + {}^{0}\dot{\vec{\omega}}_1 \times {}^{0}\vec{u}_1 + {}^{0}\vec{\omega}_1 \times \left({}^{0}\vec{\omega}_1 \times {}^{0}\vec{u}_1 \right) \right) $$

$$ = \begin{pmatrix} s_1 g \\ c_1 g \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ \ddot{\theta}_1 \cdot a_1 \\ 0 \end{pmatrix} + \begin{pmatrix} -\dot{\theta}_1^2 \cdot a_1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} s_1 g - \dot{\theta}_1^2 \cdot a_1 \\ c_1 g + \ddot{\theta}_1 \cdot a_1 \\ 0 \end{pmatrix} $$

Newton-Euler Method: Example

- Outer iteration (1st step)

$$ {}^{1}\dot{\vec{v}}_{C_1} = {}^{1}\dot{\vec{v}}_{1} + {}^{1}\dot{\vec{\omega}}_{1} \times {}^{1}\vec{u}_{C_1} + {}^{1}\vec{\omega}_{1} \times \left({}^{1}\vec{\omega}_{1} \times {}^{1}\vec{u}_{C_1} \right) $$

$$ = \begin{pmatrix} s_1 g - \dot{\theta}_1^2 \cdot a_1 \\ c_1 g + \ddot{\theta}_1 \cdot a_1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ -\ddot{\theta}_1 \cdot \frac{a_1}{2} \\ 0 \end{pmatrix} + \begin{pmatrix} \dot{\theta}_1^2 \cdot \frac{a_1}{2} \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} s_1 g - \dot{\theta}_1^2 \cdot \frac{a_1}{2} \\ c_1 g + \ddot{\theta}_1 \cdot \frac{a_1}{2} \\ 0 \end{pmatrix} $$

$$ {}^{1}\vec{f}_{C_1} = m_1 \cdot {}^{1}\dot{\vec{v}}_{C_1} $$

$$ \vec{n}_{C_1} = {}^{C_1}I \cdot \dot{\vec{\omega}}_{C_1} + \vec{\omega}_{C_1} \times {}^{C_1}I \cdot \vec{\omega}_{C_1} = \vec{0} + \vec{0} $$

Newton-Euler Method

- ☺ Arbitrary number of joints
- ☺ Loads on links are calculated
- ☺ Small computational effort $O(n)$
  (n = number of joints)
- ☹ Recursion

Physical Background: Energy

- Potential energy $P = m \cdot g \cdot h$ with mass m, height h
- Kinetic energy $K = \frac{1}{2} \cdot m \cdot v^2$
- Kinetic energy for a rotating body

$$K_{\mathrm{rot}} = \frac{1}{2} \cdot m \cdot v^2 = \frac{1}{2} \cdot m \cdot r^2 \cdot \omega^2 = \frac{1}{2} \cdot J \cdot \omega^2$$

- Kinetic energy after a free fall from height h

$$K = m \cdot g \cdot h = m \cdot \frac{v^2}{2 \cdot g} \cdot g = \frac{1}{2} m \cdot v^2$$

Lagrange Method

- Equation of movement according to Lagrange

$$\tau_i = \frac{d}{dt}\frac{\delta l}{\delta \dot{\theta}_i} - \frac{\delta l}{\delta \theta_i}$$

  - θ_i: Rotation angle or translation distance
  - $\dot{\theta}_i$: joint velocities
  - τ_i: force/torque vector in joints
- Lagrange function: $l = e_{kin} - e_{pot}$ (in reference to base)
  - Describes the difference between kinetic and potential energy of a mechanical system

Lagrange Method: Kinetic Energy

- Kinetic energy $e_{kin,i}$ of joint i

$$e_{kin,i} = \underbrace{\frac{1}{2} m_i \cdot \vec{v}_{C_i}^T \cdot \vec{v}_{C_i}}_{\text{Linear portion}} + \underbrace{\frac{1}{2}\ {}^{i}\vec{\omega}_i^T \cdot {}^{C_i}I_i \cdot {}^{i}\vec{\omega}_i}_{\text{Rotational portion}}$$

- $\vec{v}_{C_i}$ and ${}^{i}\vec{\omega}_i$ dependent on position and velocity of joints
- Total kinetic energy

$$e_{kin} = \sum_{i=1}^{n} e_{kin,i}$$

Lagrange Method: Kinetic Energy

- Kinetic energy can be described dependent on position and velocity

$$e_{kin}(\vec{\theta}, \dot{\vec{\theta}}) = \frac{1}{2} \dot{\vec{\theta}}^T \cdot M(\vec{\theta}) \cdot \dot{\vec{\theta}}$$

  - $M(\vec{\theta})$: Here $n \times n$ mass matrix, in which every element is a complex function depending on $\vec{\theta}$
  - $M(\vec{\theta})$: Positive-definite matrix, thus $\dot{\vec{\theta}}^T \cdot M(\vec{\theta}) \cdot \dot{\vec{\theta}}$ always yields a positive scalar

- This equation corresponds to the common formulation of kinetic energy of a point mass

$$e_{kin} = \frac{1}{2} m \cdot v^2$$

Lagrange Method: Potential Energy

- Potential energy u_i of link i

$$e_{pot,i} = -m_i \cdot {}^0\vec{g}^T \cdot {}^0\vec{u}_{C_i} + e_{pot,ref_i}$$

  - ${}^0\vec{g}$: 3×1 Gravitation vector, in reference to frame 0
  - ${}^0\vec{u}_{C_i}$: 3×1 Vector, describing the center of mass of i (dependent on joint position)
  - e_{pot,ref_i}: Constant, so that $e_{pot,i} \geq 0$ holds
- The total potential energy e_{pot} is given by

$$e_{pot} = \sum_{i=1}^{n} e_{pot,i}$$

- The potential energy can also be formulated as a function $e_{pot}(\vec{\theta})$ in dependence of the joint values

Lagrange Method

- Thus, for the Lagrange function follows:

$$l\left(\vec{\theta},\dot{\vec{\theta}}\right) = e_{kin}\left(\vec{\theta},\dot{\vec{\theta}}\right) - e_{pot}(\vec{\theta})$$

- For the equation of movement with torque vector $\vec{\tau}$:

$$\vec{\tau} = \frac{d}{dt}\frac{\delta l}{\delta\dot{\vec{\theta}}} - \frac{\delta l}{\delta\vec{\theta}}$$

- For a manipulator:

$$\vec{\tau} = \frac{d}{dt}\frac{\delta e_{kin}\left(\vec{\theta},\dot{\vec{\theta}}\right)}{\delta\dot{\vec{\theta}}} - \frac{\delta e_{kin}\left(\vec{\theta},\dot{\vec{\theta}}\right)}{\delta\vec{\theta}} + \frac{\delta e_{pot}(\vec{\theta})}{\delta\vec{\theta}}$$

Lagrange Method

- ☺ Formulating the equations is simple
- ☺ Closed model
- ☺ Analytical evaluation possible
- ☹ Computationally very expensive $O(n^4)$ (n = number of joints)
- ☹ Only actuating torques are calculated

Efficiency of the Approaches

- Newton-Euler method
  - Multiplications: $126n - 99$
  - Additions: $106n - 92$
- Lagrange method
  - Multiplications: $32n^4 + 86n^3 + 171n^2 + 53n - 128$
  - Additions: $25n^4 + 66n^3 + 129n^2 + 42n - 96$
- For typical robots ($n = 6$ joints) the Newton-Euler method is $100 \times$ more efficient
- Optimizations possible for both methods

Requirements for Manipulators

- <u>Reliable</u> positioning : Accuracy (repeatability)
- Collision avoidance
- Execution of movement: Fluid with appropriate velocities and accelerations
- Adaptation to changing conditions

Fundamental Questions

- Direct kinematics
  - Given all joint values. Where is the TCP?
- Inverse kinematics
  - Given TCP-pose. Which joint values are required to achieve pose?
- Dynamics
  - Which forces/torques do the actuators have to enact to accelerate TCP by a certain magnitude?
- Trajectory planning
  - How does a „good“ trajectory that avoids collisions look like?

Next Lecture …

Trajectory control

- Fundamentals
- Methods of interpolation
- Spline-interpolation