Robot Dynamics Exercise Session 04

previous weeks: multi-body kinematics

- enough to describe the motion, but not the cause of motion
- motions may not be feasible in reality (especially when underactuated)

now: multi-body dynamics

- describe the cause of motion $\vec{F}, \vec{\tau} \quad \leftrightarrow \quad \ddot{\vec{q}}$
- motions are physically consistent

1) Dynamics Equations of Motion

- Newton-Euler Equations
- Projected Newton-Euler Equations

} conservation of linear and angular momentum

- Euler-Lagrange Equations (Lagrange II)
- Hamilton's Equations

} energy-based approaches

Kinetic Energy, Potential Energy

$$\mathcal{L} = \mathcal{T} - \mathcal{U} \quad \text{(Lagrangian)}$$

$$\mathcal{H} = \mathcal{T} + \mathcal{U} \quad \text{(Hamiltonian)}$$

"total energy"

Robot Dynamics Exercise Session 04

2) Generalized EoM

$$M(\vec{q})\,\ddot{\vec{q}} + \vec{b}(\vec{q}, \dot{\vec{q}}) + \vec{g}(\vec{q}) = S^\top \vec{\tau} + J^\top \vec{F} \quad \in \mathbb{R}^q$$

mass matrix	coriolis and centrifugal term	gravity term	selection matrix	joint torques	stacked jacobians	stacked forces acting on the robot
$\mathbb{R}^{q\times q}$	$\mathbb{R}^q$	$\mathbb{R}^q$	$\mathbb{R}^{\tau\times q}$	$\mathbb{R}^\tau$	$\mathbb{R}^{F\times q}$	$\mathbb{R}^F$

special cases:

- fully actuated: $S = \mathbb{I}_{q\times q}$
- freely moving: $J^\top \vec{F} = \vec{0}$

rotate

$\mathbb{R}^{3\times 3}$

Mass Matrix: $$M(\vec{q}) = \sum_{i=1}^{N_{\text{bodies}}} {}_{\mathcal{A}}J_{S_i}^\top m_i \, {}_{\mathcal{A}}J_{S_i} + \boxed{{}_{\mathcal{B}}J_{\mathcal{R}_i}^\top \, {}_{\mathcal{B}}\Theta_{S_i} \, {}_{\mathcal{B}}J_{\mathcal{R}_i}}$$

$${}_{\mathcal{C}}J_{\mathcal{R}_i}^\top \underbrace{R_{\mathcal{CB}} \, {}_{\mathcal{B}}\Theta_{S_i} \, R_{\mathcal{BC}}}_{{}_{\mathcal{C}}\Theta_{S_i}} \, {}_{\mathcal{C}}J_{\mathcal{R}_i}$$

CoM-Jacobian

≠ Jacobian at frame i, J_i

- symmetric positive definite (since kinetic Energy $\mathcal{T} = \frac{1}{2}\dot{\vec{q}}^\top M \dot{\vec{q}} > 0 \quad \forall \dot{\vec{q}}$) matrix ($\mathbb{R}^{q\times q}$)
  => invertible

Robotic Systems Lab

Robot Dynamics Exercise Session 04

Coriolis + Centrifugal Terms:

$$\vec{b}(\vec{q},\dot{\vec{q}}) = \sum_{i=1}^{N_{\text{bodies}}} {}_{\mathcal{A}}J_{S_i}^{\top} m_i \, {}_{\mathcal{A}}\dot{J}_{S_i}\dot{\vec{q}} + {}_{\mathcal{B}}J_{\mathcal{R}_i}^{\top} \, {}_{\mathcal{B}}\Theta_{S_i} \, {}_{\mathcal{B}}\dot{J}_{\mathcal{R}_i}\dot{\vec{q}} + {}_{\mathcal{C}}J_{\mathcal{R}_i}^{\top} \underbrace{\left({}_{\mathcal{C}}J_{\mathcal{R}_i}\dot{\vec{q}}\right)}_{{}_{\mathcal{C}}\vec{\omega}_i} \times {}_{\mathcal{C}}\Theta_{S_i} \underbrace{\left({}_{\mathcal{C}}J_{\mathcal{R}_i}\dot{\vec{q}}\right)}_{{}_{\mathcal{C}}\vec{\omega}_i}$$

- vector ($\mathbb{R}^q$) depends quadratically on velocities, i.e., $\dot{q}_i^2$ (centrifugal) and $\dot{q}_i \cdot \dot{q}_j$ (Coriolis)
- can be written as $C(\vec{q},\dot{\vec{q}}) \cdot \dot{\vec{q}}$ (non-unique choice of C)

Gravitational Term:

$$\vec{g}(\vec{q}) = \sum_{i=1}^{N_{\text{bodies}}} -{}_{\mathcal{A}}J_{S_i}^{\top} \, {}_{\mathcal{A}}\vec{F}_{g_i} \qquad \text{with} \qquad {}_{\mathcal{I}}\vec{F}_{g_i} = \begin{bmatrix} 0 \\ 0 \\ -m\underbrace{g}_{9.81\frac{m}{s^2}} \end{bmatrix} \quad \text{if } z\text{-axis points upwards}$$

Robot Dynamics Exercise Session 04

3) Practical Applications

physical simulation

- solve forward dynamics problem: $\ddot{\vec{q}} = \mathrm{FD}(\vec{\tau}, \vec{q}, \dot{\vec{q}}) \overset{\text{e.g.}}{=} M^{-1}(\vec{\tau} - \vec{b} - \vec{g})$
- <u>given</u>: $\vec{q}(0),\ \dot{\vec{q}}(0),\ \vec{\tau}(t) \quad \forall t \quad 0 \leq t \leq T$ solve for $\ddot{\vec{q}}$ and integrate to get $\dot{\vec{q}}(t)$ and $\vec{q}(t)$

model-based control

- solve inverse dynamics problem: $\vec{\tau} = \mathrm{ID}(\vec{q}, \dot{\vec{q}}, \ddot{\vec{q}}^*)$
- <u>given</u>: $\ddot{\vec{q}}^*(t) \rightarrow$ reference motion
  $\vec{q}(t),\ \dot{\vec{q}}(t) \rightarrow$ measurements at t

  solve for $\vec{\tau}(t)$ and apply it to the system

software implementation

- use pre-existing software tools (RBDL, RobCoGen, Pinocchio)
- some well-known algorithms:
  - Recursive Newton-Euler Algorithm (RNEA) => solve $\mathrm{ID}(\vec{q},\ \dot{\vec{q}},\ \ddot{\vec{q}})$
  - Articulated-Body Algorithm (ABA) => solve $\mathrm{FD}(\vec{\tau},\ \vec{q},\ \dot{\vec{q}})$
  - Composite Rigid-Body Algorithm (CRBA) => compute $M(\vec{q})$

Robotic Systems Lab