MEE5114 Advanced Control for Robotics

Lecture 9: Dynamics of Open Chains

Prof. Wei Zhang

CLEAR Lab

Department of Mechanical and Energy Engineering

Southern University of Science and Technology, Shenzhen, China

`https://www.wzhanglab.site/`

- Spatial Accelleration: $A \in \mathbb{R}^6$, $A_{body} \triangleq \dot{V}_{body}$ (coordinate)

working with inertia/stationary frame: ${}^{o}A_{body} = \underbrace{\frac{d}{dt}({}^{o}V_{body})}_{{}^{o}\overset{\circ}{V}_{body}}$ apparent derivative

working with moving frame:

Outline

$$ {}^{B}A_{body} = \frac{d}{dt}\left({}^{B}\mathcal{V}_{body}\right) + {}^{B}\mathcal{V}_{B} \times {}^{B}\mathcal{V}_{body} $$

$\rightarrow \left[{}^{B}\mathcal{V}_{B}\times\right]$ 6x6 matrix

$$ {}^{B}A_{body} = {}^{B}X_{0}\, {}^{0}A_{body} $$

- Introduction

$- \ \dot{R}_A = \omega_A \times R_A, \quad [R\omega] = R[\omega]R^T$

$- \ {}^{0}\dot{X}_A = \mathcal{V}_A \times {}^{0}X_A = [\mathcal{V}_A \times]_{6\times 6}\, {}^{0}X_A, \quad [X\mathcal{V}\times] = X[\mathcal{V}\times]X^T$

- Inverse Dynamics: Recursive Newton-Euler Algorithm (RNEA)

\- spatial force/wrench:

- Analytical Form of the Dynamics Model

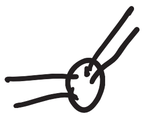
$$ {}^{B}\mathcal{F} = \begin{bmatrix} {}^{B}n_{O_B} \\ {}^{B}f \end{bmatrix}, \quad {}^{A}\mathcal{F} = {}^{A}X_B^{*}\, {}^{B}\mathcal{F} $$

$$ {}^{A}X_B^{*} = \left({}^{B}X_A\right)^T $$

- Forward Dynamics Algorithms

$$ \left({}^{0}\dot{X}_A^{*}\right) = \left[\mathcal{V}_A \times^{*}\right] {}^{0}X_A^{*} $$

\- Joint torque:



$$ \tau\dot{\theta} = \mathcal{V}^T\mathcal{F} = S^T\dot{\theta}\mathcal{F} $$

$$ \tau = S^T\mathcal{F} = \mathcal{F}^T S $$

spatial momentum:

$$ {}^{A}h = \begin{bmatrix} {}^{A}\phi_{O_A} \\ {}^{A}L \end{bmatrix}, \quad {}^{A}h = {}^{A}X_B^{*}\, {}^{B}h $$

spatial inertia

From Single Rigid Body to Open Chains

$${}^C\mathcal{I} = \begin{bmatrix} {}^C\bar{I} & 0 \\ 0 & m I_{3\times 3} \end{bmatrix}$$

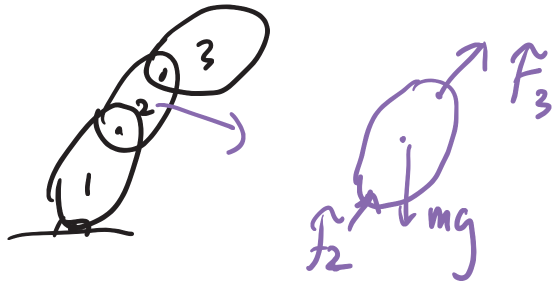
$${}^A\mathcal{I} = ({}^A X_C^*)\, {}^C\mathcal{I}\, ({}^C X_A)$$

- Recall Newton-Euler Equation for a single rigid body:

$$\underbrace{\mathcal{F} = \frac{d}{dt}h = \mathcal{I}\mathcal{A}}_{\text{coordinate-free}} + \mathcal{V} \times^* \mathcal{I}\mathcal{V}$$

- Open chains consist of multiple rigid links connected through joints
  bodies

- Dynamics of adjacent links are coupled.



- This lecture: model multi-body dynamics subject to joint constraints.

Preview of Open-Chain Dynamics

- Equations of Motion are a set of 2nd-order differential equations:

$$\tau = M(\theta)\ddot{\theta} + \tilde{c}(\theta, \dot{\theta})$$

$+ c(\theta, \dot{\theta}) + \tau_g(\theta) + J^T F_{ext} + \cdots$

  - $\theta \in \mathbb{R}^n$: vector of joint variables; $\tau \in \mathbb{R}^n$: vector of joint forces/torques
  - $M(\theta) \in \mathbb{R}^{n \times n}$: mass matrix
  - $\tilde{c}(\theta, \dot{\theta}) \in \mathbb{R}^n$: forces that lump together centripetal, Coriolis, gravity, friction terms, and torques induced by external forces. These terms depend on θ and/or $\dot{\theta}$

like simulation

- **Forward dynamics:** Determine acceleration $\ddot{\theta}$ given the state $(\theta, \dot{\theta})$ and the joint forces/torques:

$$\ddot{\theta} \leftarrow \text{FD}(\tau, \theta, \dot{\theta}, \mathcal{F}_{ext})$$

- **Inverse dynamics:** Finding torques/forces given state $(\theta, \dot{\theta})$ and desired acceleration $\ddot{\theta}$

Given desired motion $(\theta, \dot{\theta}, \ddot{\theta})$, $\tau \leftarrow \text{ID}(\theta, \dot{\theta}, \ddot{\theta}, \mathcal{F}_{ext})$

find the required torque to generate the desired motion

Lagrangian vs. Newton-Euler Methods

- There are typically two ways to derive the equation of motion for an open-chain robot: Lagrangian method and Newton-Euler method

Lagrangian Formulation

- Energy-based method
- Dynamic equations in closed form
- Often used for study of dynamic properties and analysis of control methods

Newton-Euler Formulation

- Balance of forces/torques
- Dynamic equations in numeric/recursive form
- Often used for numerical solution of forward/inverse dynamics

- We focus on Newton-Euler Formulation

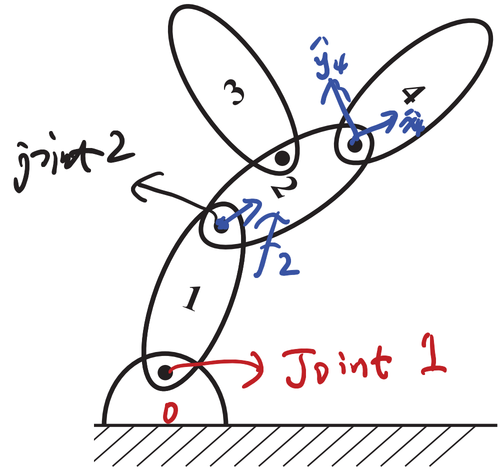
Featherstone's book

wensing's note

Outline

- Introduction
- Inverse Dynamics: Recursive Newton-Euler Algorithm (RNEA)
- Analytical Form of the Dynamics Model
- Forward Dynamics Algorithms

RNEA: Notations



- Number bodies: 1 to N
  - Parent: $p(i)$: e.g. $p(3) = 2$, $p(4) = 2$
  - Children: $c(i)$ e.g. $c(2) = \{3, 4\}$, $c(1) = \{2\}$
- Joint i connects $p(i)$ to i
- Frame {i} attached to body i at the joint; frame {4} moves with body {4}
- $\mathcal{S}_i$: Spatial velocity (screw axis) of joint i : e.g. ${}^4\mathcal{S}_4 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ constant
- $\mathcal{V}_i$ and $\mathcal{A}_i$: spatial velocity and acceleration of body i
- $\mathcal{F}_i$ $\in \mathbb{R}^6$: force (wrench) onto body i from body $p(i)$
- Note: By default, all vectors $(\mathcal{S}_i, \mathcal{V}_i, \mathcal{F}_i)$ are expressed in local frame {i}

RNEA: Velocity and Accel. Propagation (Forward Pass)

Goal: Given joint velocity $\dot{\theta}$ and acceleration $\ddot{\theta}$, compute the body spatial velocity $\mathcal{V}_i$ and spatial acceleration $\mathcal{A}_i$

$$\begin{cases} \text{Velocity Propagation:} & {}^i\mathcal{V}_i = \left({}^iX_{p(i)}\right)\ {}^{p(i)}\mathcal{V}_{p(i)} + {}^i\mathcal{S}_i\,\dot{\theta}_i \\ \text{Accel Propagation:} & {}^i\mathcal{A}_i = \left({}^iX_{p(i)}\right)\ {}^{p(i)}\mathcal{A}_{p(i)} + {}^i\mathcal{V}_i \times {}^i\mathcal{S}_i\dot{\theta}_i + {}^i\mathcal{S}_i\ddot{\theta}_i \end{cases}$$

Recall

$\tau = ID(\theta, \dot{\theta}, \ddot{\theta}, F_{ext})$

motion of joint variables

Velocity: $\mathcal{V}_1 = S_1\dot{\theta}_1$, $\mathcal{V}_2 = \mathcal{V}_1 + \mathcal{V}_{2/1}$

$= S_1\dot{\theta}_1 + S_2\dot{\theta}_2$

work with local coordinates: ${}^1\mathcal{V}_1 = {}^1S_1\dot{\theta}_1$

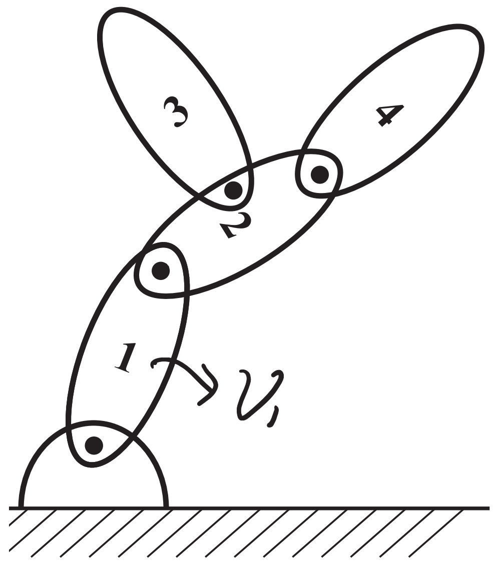
${}^2\mathcal{V}_2 = {}^2X_1\,{}^1S_1\dot{\theta}_1 + {}^2S_2\dot{\theta}_2$

In general: $\boxed{{}^i\mathcal{V}_i = {}^iX_{p(i)}\,{}^{p(i)}\mathcal{V}_{p(i)} + {}^iS_i\dot{\theta}_i}$

Accel: $\mathcal{A}_2 = \dot{\mathcal{V}}_2 = \dot{\mathcal{V}}_1 + \dot{\mathcal{V}}_{2/1} = \mathcal{A}_1 + \mathcal{A}_{2/1}$

In coordinate: ${}^2\mathcal{A}_2 = {}^2X_1\,{}^1\mathcal{A}_1 + {}^2\left[\frac{d}{dt}(S_2\dot{\theta}_2)\right]$

coordinate free notation



$${}^{2}\left[\frac{d}{dt}\left(\underbrace{S_2\dot{\theta}_2}_{V_2}\right)\right] = \frac{d}{dt}\left(\underbrace{{}^{2}S_2\dot{\theta}_2}_{{}^{2}V_2}\right) + {}^{2}V_2 \times {}^{2}S_2\dot{\theta}_2 = {}^{2}S_2\ddot{\theta}_2 + \underline{{}^{2}V_2} \times {}^{2}S_2\dot{\theta}_2$$

$${}^{2}A_2 = \underline{{}^{2}X_1\,{}^{1}A_1 + {}^{2}V_2 \times {}^{2}S_2\dot{\theta}_2 + {}^{2}S_2\ddot{\theta}_2}$$

RNEA: Force Propagation (Backward Pass)

Goal: Given body spatial velocity $\mathcal{V}_i$ and spatial acceleration $\mathcal{A}_i$, compute the joint wrench $\mathcal{F}_i$ and the corresponding torque $\tau_i = \mathcal{S}_i^T \mathcal{F}_i$

$$\begin{cases} \mathcal{F}_i & = \mathcal{I}_i \mathcal{A}_i + \mathcal{V}_i \times^* \mathcal{I}_i \mathcal{V}_i + \sum_{j \in c(i)} \mathcal{F}_j \\ \tau_i & = \mathcal{S}_i^T \mathcal{F}_i \end{cases}$$

$f = mg$

$\mathcal{F} = \mathcal{I} A_g$

Body 4: $\mathcal{F}_4 + \mathcal{F}_{g4} = \mathcal{I}_4 \mathcal{A}_4 + \mathcal{V}_4 \times^* \mathcal{I}_4 \mathcal{V}_4$

$$\mathcal{F}_4 = \mathcal{I}_4 \mathcal{A}_4 + \mathcal{V}_4 \times^* \mathcal{I}_4 \mathcal{V}_4 - \mathcal{F}_{g4}$$

Note: $\mathcal{F}_{g4} = \mathcal{I}_4 \, {}^4\mathcal{A}_g = \mathcal{I}_4 \, {}^4X_0 \, {}^0\mathcal{A}_g$

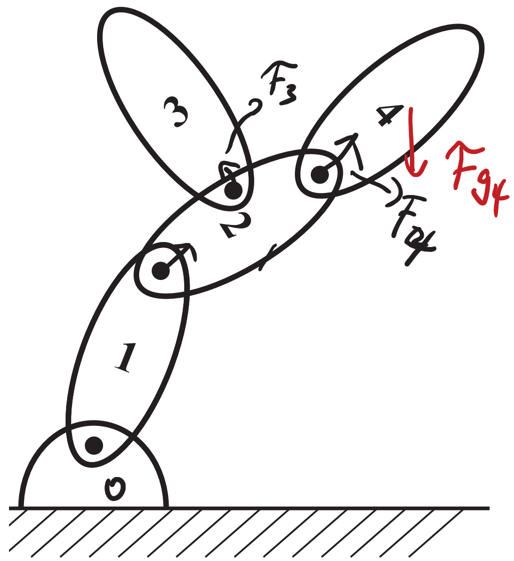
$\tau_4 = S_4^T \mathcal{F}_4$ ✓

Body 2: $\mathcal{F}_2 = \underbrace{\mathcal{I}_2 \mathcal{A}_2 + \mathcal{V}_2 \times^* \mathcal{I}_2 \mathcal{V}_2}_{} + \underbrace{\mathcal{F}_3 + \mathcal{F}_4}_{} - \mathcal{F}_{g2}$

$\Downarrow$

$\mathcal{I}_2 \, {}^2X_0 \, {}^0\mathcal{A}_g$

$\tau_2 = S_2^T \mathcal{F}_2$



Recursive Newton-Euler Algorithm

without gravity trick:

modify ① to

$\mathcal{F}_i = \mathcal{I}_i \mathcal{A}_i + \mathcal{V}_i \times^* \mathcal{I}_i \mathcal{V}_i - \mathcal{I}_i {}^iX_0 {}^0\mathcal{A}_g$

$$\tau \leftarrow \text{RNEA}(\theta, \dot\theta, \ddot\theta, \mathcal{F}_{ext}; \text{Model})$$

Initialize: $\mathcal{V}_0 = 0$, $\mathcal{A}_0 = -\mathcal{A}_g$

- Forward pass:

For $i=1$ to N

$\mathcal{V}_i = {}^iX_{p(i)} \mathcal{V}_{p(i)} + S_i \dot\theta_i$

$\mathcal{A}_i = {}^iX_{p(i)} \mathcal{A}_{p(i)} + S_i \ddot\theta_i + \mathcal{V}_i \times S_i \dot\theta_i$

$\mathcal{F}_i = \mathcal{I}_i \mathcal{A}_i + \mathcal{V}_i \times^* \mathcal{I}_i \mathcal{V}_i$ ... ①

wrench due to i^{th}-body motion only

- Backward pass:

For $i = N:-1:1$

$\tau_i = S_i^T \mathcal{F}_i$

$\mathcal{F}_{p(i)} = \mathcal{F}_{p(i)} + {}^{p(i)}X_i^* \mathcal{F}_i$

End

Outline

- Introduction
- Inverse Dynamics: Recursive Newton-Euler Algorithm (RNEA)
- Analytical Form of the Dynamics Model
- Forward Dynamics Algorithms

Structures in Dynamic Equation (1/3)

- Jacobian of each link (body): $J_1, \ldots, J_4$

J_i : denote the Jacobian of body i (Link), i.e. $\mathcal{V}_i = J_i \dot{\theta} = [J_{i1}\ J_{i2} \cdots J_{i4}] \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \end{bmatrix}$

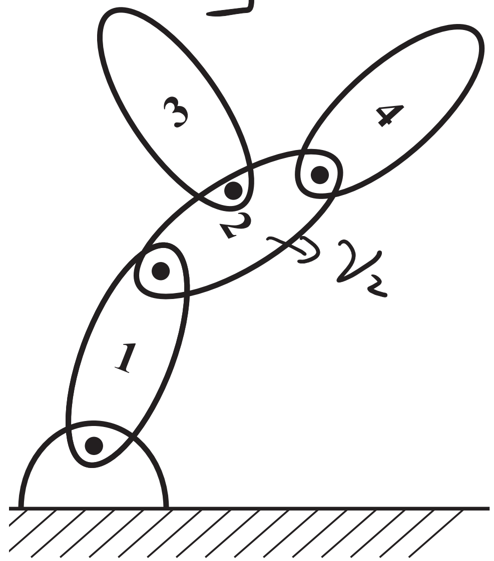
e.g. $\mathcal{V}_1 = J_1 \dot{\theta} = [\delta_{11} S_1\ \ \delta_{12} S_2\ \ \delta_{13} S_3\ \ \delta_{14} S_4] \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \end{bmatrix} = [S_1, 0, 0, 0] \begin{bmatrix} \dot{\theta}_1 \\ \vdots \\ \dot{\theta}_4 \end{bmatrix}$

$$\delta_{ij} = \begin{cases} 1, & \text{if joint } j \text{ supports body } i \\ 0, & \text{otherwise} \end{cases}$$

$\mathcal{V}_2 = J_2 \dot{\theta} = [S_1\ \ S_2\ \ 0\ \ 0]\, \dot{\theta}$

In $\{2\}$: ${}^2\mathcal{V}_2 = \underbrace{[{}^2X_1\, {}^1S_1 \ \vdots\ {}^2S_2 \ \vdots\ 0, 0]}_{{}^2J_2}\, \dot{\theta}$

${}^4J_4 = [{}^4X_1\, {}^1S_1 \ \vdots\ {}^4X_2\, {}^2S_2 \ \vdots\ 0\ \ {}^4S_4]$



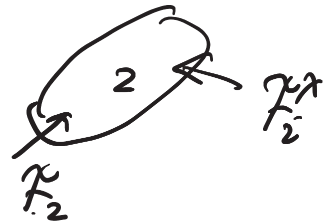
Structures in Dynamic Equation (2/3)

- ~~Torque required to generate a "force" $\mathcal{F}_4$ to body~~ 4

see the two-body example:

① Forward pass: $\mathcal{V}_1 = S_1\dot{\theta}_1$, $\mathcal{V}_2 = [{}^2X_1 S_1 \;\vdots\; S_2]\begin{bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\end{bmatrix}$

$\mathcal{A}_1$, $\mathcal{A}_2 = \dots$

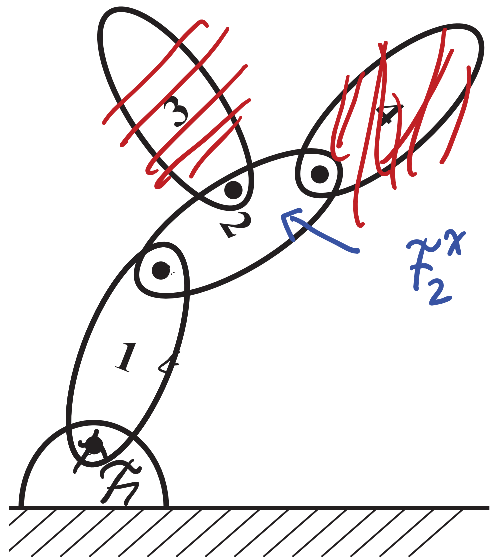


② Backward pass: $\mathcal{F}_2 = \mathcal{I}_2\mathcal{A}_2 + \mathcal{V}_2\times^*\mathcal{I}_2\mathcal{V}_2 - \mathcal{F}_2^x$

$\mathcal{F}_1 = \mathcal{I}_1\mathcal{A}_1 + \mathcal{V}_1\times^*\mathcal{I}_1\mathcal{V}_1 + {}^1X_2^*\mathcal{F}_2$

$= \mathcal{I}_1\mathcal{A}_1 + \mathcal{V}_1\times^*\mathcal{I}_1\mathcal{V}_1 + {}^2X_1^T(\mathcal{I}_2\mathcal{A}_2 + \mathcal{V}_2\times^*\mathcal{I}_2\mathcal{V}_2) - {}^2X_1^T\mathcal{F}_2^x$

$\tau_2 = S_2^T\mathcal{F}_2 = S_2^T(\mathcal{I}_2\mathcal{A}_2+\cdots) - S_2^T\mathcal{F}_2^x$



$\Rightarrow \tau_1 = S_1^T\mathcal{F}_1 = \underbrace{\boxed{S_1^T(\mathcal{I}_1\mathcal{A}_1+\cdots)}}_{①} + \underbrace{\boxed{({}^2X_1S_1)^T(\mathcal{I}_2\mathcal{A}_2+\cdots)}}_{②} \underbrace{\boxed{- ({}^2X_1S_1)^T\mathcal{F}_2^x}}_{③}$

Structures in Dynamic Equation (3/3)

- Overall torque expression: ①: $\underbrace{S_1^T(\mathcal{I}_1 A_1 + \mathcal{V}_1 \times^* \mathcal{I}_1 \mathcal{V}_1)}$

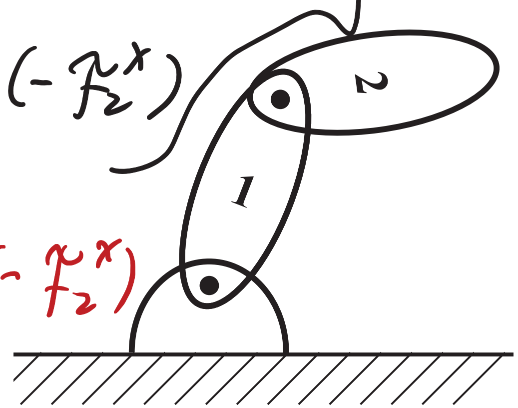
torque @ joint 1 due to motion of body 2.

② torque @ joint 1 due to motion of body 2

③ torque . . . . external force of body 2 $\mathcal{F}_2^x$

$$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} S_1^T(\mathcal{I}_1 A_1 + \cdots) + ({}^2X_1 S_1)^T(\mathcal{I}_2 A_2 + \cdots) + ({}^2X_1 S_1)^T(-\mathcal{F}_2^x) \\ 0\,(\mathcal{I}_1 A_1 + \cdots) + S_2^T(\mathcal{I}_2 A_2 + \cdots) + S_2^T(-\mathcal{F}_2^x) \end{bmatrix}$$

$$= \underbrace{\begin{bmatrix} S_1^T \\ 0 \end{bmatrix}}_{= J_1^T}(\mathcal{I}_1 A_1 + \cdots) + \underbrace{\begin{bmatrix} ({}^2X_1 S_1)^T \\ S_2^T \end{bmatrix}}_{= J_2^T}(\mathcal{I}_2 A_2 + \cdots) + \underbrace{\begin{bmatrix} ({}^2X_1 S_1)^T \\ S_2^T \end{bmatrix}}_{J_2^T}(-\mathcal{F}_2^x)$$



$\underbrace{[S_1 \;\vdots\; 0]}_{{}^1J_1}$

$[{}^2X_1 S_1 \;\vdots\; S_2] \to J_2$

Derivation of Overall Dynamics Equation

- overall: in general with N-Links/Joints

$$\tau = \sum_{i=1}^{N} \left\{ J_i^T \left(\mathcal{I}_i \mathcal{A}_i + \mathcal{V}_i \times^* \mathcal{I}_i \mathcal{V}_i \right) + J_i^T \left(\underbrace{\text{external force terms}}_{} \right) \right\}$$

e.g. gravity or other external forces.

$\mathcal{V}_i = J_i \dot{\theta}$ → body i Jacobian

$$\mathcal{A}_i = \dot{\mathcal{V}}_i = \left(J_i \ddot{\theta} + \dot{J}_i \dot{\theta} + \mathcal{V}_i \times J_i \dot{\theta} \right)$$

$$\Rightarrow \tau = \sum_{i=1}^{N} J_i^T \mathcal{I}_i J_i \ddot{\theta} + J_i^T \mathcal{I}_i \dot{J}_i \dot{\theta} + J_i^T \mathcal{I}_i \mathcal{V}_i \times J_i \dot{\theta} + J_i^T \mathcal{V}_i \times^* \mathcal{I}_i \mathcal{V}_i$$

$J_i \dot{\theta}$

$$= \underbrace{\left(\sum_{i=1}^{N} J_i^T \mathcal{I}_i J_i \right)}_{M(\theta)} \ddot{\theta} + \underbrace{\sum_{i=1}^{N} J_i^T \left(\mathcal{I}_i \dot{J}_i + \mathcal{I}_i \mathcal{V}_i \times J_i + \mathcal{V}_i \times^* \mathcal{I}_i J_i \right)}_{\triangleq\, C(\theta, \dot{\theta})} \dot{\theta}$$

$$\boxed{\tau = M(\theta)\ddot{\theta} + c(\theta, \dot{\theta})\dot{\theta} + \tau_g + J^T(\theta)\mathcal{F}_{ext}} \tag{1}$$

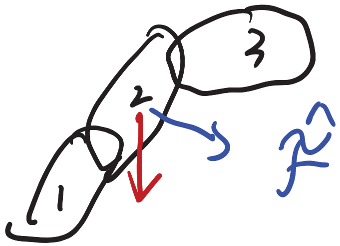
If consider gravity we need to add: $\sum_{i=1}^{N} J_i^T \mathcal{I}_i \, {}^iX_0 (-{}^0\mathcal{A}_g)$ τ_g

- J_i : body / link i Jacobian, $\underset{6\times 1}{V_i} = \underset{6\times n}{J_i}\, \dot{\theta} \to \begin{bmatrix} \dot\theta_1 \\ \vdots \\ \dot\theta_n \end{bmatrix}$

- $\tau = \begin{bmatrix} \tau_1 \\ \vdots \\ \tau_n \end{bmatrix} \in \mathbb{R}^n$, τ plays two major roles.

  ① generate motion

  ② generate force / torque



$\hat{\mathcal{F}}$ (wrench from body 2 to the environment)

- ~~Co~~ Only consider body 2's effect

$$\tau = J_2^T(\underbrace{\mathcal{I}_2 A_2 + V_2 \times^* \mathcal{I}_2 V_2}) + J_2^T \hat{\mathcal{F}}$$

If consider gravity, we also add $J_2^T(-\mathcal{I}_2\, {}^2X_0\, {}^0A_g)$

Properties of Dynamics Model of Multi-body Systems

- - If consider all the bodies.

$$\tau = \text{all motions} + \text{all forces}$$

$$= \boxed{\sum_{i=1}^{n} J_i^T \big(\underbrace{\mathcal{I}_i A_i + V_i \times^* \mathcal{I}_i V_i}_{} \big) + J_i^T \big(-\mathcal{I}_i \, {}^i X_0 \, {}^0 A_g \big)}$$

Outline

- Introduction
- Inverse Dynamics: Recursive Newton-Euler Algorithm (RNEA)
- Analytical Form of the Dynamics Model
- Forward Dynamics Algorithms

Forward Dynamics Problem

$$\tau = M(\theta)\ddot{\theta} + \underbrace{c(\theta,\dot{\theta})\dot{\theta} + \tau_g + J^T(\theta)\mathcal{F}_{ext}}_{\tilde{c}(\theta,\dot{\theta})} \tag{2}$$

given

applied from the body to environment

- Inverse dynamics: $\tau \leftarrow \text{RNEA}(\theta, \dot{\theta}, \ddot{\theta}, \mathcal{F}_{ext})$ $O(N)$ complexity
  - RNEA can work directly with a given URDF model (kinematic tree + joint model + dynamic parameters). It does not require explicit formula for $M(\theta), \tilde{c}(\theta, \dot{\theta})$

- **Forward dynamics:** Given $(\theta, \dot{\theta})$, τ, $\mathcal{F}_{ext}$, find $\ddot{\theta}$
  1. Calculate $\tilde{c}(\theta, \dot{\theta}) = c(\theta,\dot{\theta})\dot{\theta} + \tau_g + J^T \mathcal{F}_{ext}$
  2. Calculate mass matrix $M(\theta)$
  3. Solve $M\ddot{\theta} = \tau - \tilde{c}$ $\Rightarrow \ddot{\theta} = M^{-1}(\tau - \tilde{c})$

this is not the most efficient way to find $\ddot{\theta}$

Calculations of $\tilde{c}$ and M

- Denote our inverse dynamics algorithm: $\tau = \text{RNEA}(\theta, \dot{\theta}, \ddot{\theta}, \mathcal{F}_{ext}) = M\ddot{\theta} + \tilde{c}$

1. **Calculation of $\tilde{c}$**: obviously, $\tau = \tilde{c}(\theta, \dot{\theta})$ if $\ddot{\theta} = 0$. Therefore, $\tilde{c}$ can be computed via:

$$\tilde{c}(\theta, \dot{\theta}) = \text{RNEA}(\theta, \dot{\theta}, 0, \mathcal{F}_{ext}) = c(\theta,\dot{\theta})\dot{\theta} + \tau_g + J^T f_{ext}$$

- **Calculation of M:** Note that $\tilde{c}(\theta, \dot{\theta}) = c(\theta, \dot{\theta})\dot{\theta} - \tau_g - J^T(\theta)\mathcal{F}_{ext}$.
  - Set $g = 0$, $\mathcal{F}_{ext} = 0$, and $\dot{\theta} = 0$, then $\tilde{c}(\theta, \dot{\theta}) = 0 \Rightarrow \tau = M(\theta)\ddot{\theta}$ if $\ddot{\theta} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$

    $\tau_g = 0$ $\quad \Rightarrow c(\theta,\dot{\theta})\dot{\theta} = 0$, $\quad \tau = [M_1(\theta) \; M_2(\theta) \; \cdots]\ddot{\theta} = M_1(\theta)$
  - We can compute the jth column of $M(\theta)$ by calling the inverse algorithm

$$M_{:,j}(\theta) = \text{RNEA}(\theta, 0, \ddot{\theta}_j^0, 0) \qquad \ddot{\theta}_j^0 = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \end{bmatrix} \leftarrow j^{th} \text{ element}$$

    where $\ddot{\theta}_j^0$ is a vector with all zeros except for a 1 at the jth entry.

$$\ddot{\theta}_1^0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \ddot{\theta}_2^0 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \end{bmatrix}$$

- A more efficient algorithm for computing M is the *Composite-Rigid-Body Algorithm (CRBA)*. Details can be found in Featherstone's book.

Forward Dynamics Algorithm

- Now assume we have $\theta, \dot{\theta}, \tau, M(\theta), \tilde{c}(\theta, \dot{\theta})$, then we can immediately compute $\ddot{\theta}$ as $\ddot{\theta} = \underline{M^{-1}(\theta)\left[\tau - \tilde{c}(\theta, \dot{\theta})\right]}$ $\quad \ddot{\theta} = FD(\tau, \theta, \dot{\theta}, \mathcal{F}_{ext})$

- This provides a 2nd-order differential equation in $\mathbb{R}^n$, we can easily simulate the joint trajectory over any time period (under given ICs θ^o and $\dot{\theta}^o$)

$x_1 = \theta \in \mathbb{R}^n, \quad x_2 = \dot{\theta} \in \mathbb{R}^n$

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ M^{-1}(x_1)\left(\tau - \tilde{c}(x_1, x_2)\right) \end{bmatrix} = f(x)$$

- Computational Complexity:
  - RNEA: $O(N)$
  - $\tilde{c} = RNEA(\theta, \dot{\theta}, 0, \mathcal{F}_{ext})$: $O(N)$
  - $M(\theta)$: $O(N^2)$
  - $M^{-1}(\theta)$: $O(N^3)$
  - Most efficient forward dynamics algorithm:
    Articulated-Body Algorithm (ABA): $\underline{O(N)}$

More Discussions

Inertia matrix symmetric / positive semidefinite

- $$\tau = \underbrace{\left(\sum_{i=1}^{N}\left(J_i^T \mathcal{G}_i J_i\right)\right)}_{\triangleq M(\theta)} \ddot{\theta} + \sum\left(\quad \right)\dot{\theta} \quad \cdots$$

- $M(\theta)$: Mass matrix, $\underline{M(\theta)^T} = \underline{M(\theta)}$, $M(\theta)$ is also positive semi-definite.

- There are many equivalent ways to define $C(\theta, \dot{\theta})$, they all lead to the same product $C(\theta,\dot{\theta})\dot{\theta}$

e.g.
$$\boxed{C(\theta,\dot{\theta})\dot{\theta}} = \begin{bmatrix} -2\dot{\theta}_2\dot{\theta}_1 \\ \dot{\theta}_1^2 \end{bmatrix} = \overbrace{\begin{bmatrix} -2\dot{\theta}_2 & 0 \\ \dot{\theta}_1 & 0 \end{bmatrix}}^{C(\theta,\dot{\theta})} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$
$$= \begin{bmatrix} 0 & -2\dot{\theta}_1 \\ \dot{\theta}_1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

More Discussions

- Typical expression for C: $[C]_{ij} = \sum_{k=1} \frac{1}{2}\left(\frac{\partial M_{ij}}{\partial \theta_k} + \frac{\partial M_{ik}}{\partial \theta_j} - \frac{\partial M_{jk}}{\partial \theta_i}\right)\dot{\theta}_k$

$\triangleq \Gamma_{ijk}$ christoffel

- $C(\theta,\dot\theta)$ defined using Γ_{ikj}

satisfies: $\underbrace{\dot{M} - 2C}_{n\times n}$ skew symmetric

$f(x)$

$f(\alpha x+\beta y) = \alpha f(x) + \beta f(y)$

- $\boxed{M(\theta),\ C(\theta,\dot\theta),\ \tau_j \text{ all depend on } \mathcal{I}_i \text{ linearly.}}$

$\Downarrow$

$M(\theta) \triangleq \sum_i J_i^T \mathcal{I}_i J_i$

~~Fix θ:~~

$M(\mathcal{I}_i)$:

$M(\alpha \mathcal{I}_i^{(1)} + \beta \mathcal{I}_i^{(2)})$

$= \alpha M(\mathcal{I}_i^{(1)}) + \beta M(\mathcal{I}_i^{(2)})$

$\Rightarrow$ Identification of $\{\mathcal{I}_i\}$, can be done using linear least squares.