24.3.23

Chapter 8: Dynamics of open chains

Dynamics: forces and torques that cause motion.

- Forward dynamics: Given $\left\{ \begin{array}{l} \cdot\ \theta \\ \cdot\ \dot{\theta} \\ \cdot\ \tau \end{array} \right.$ (θ, θ̇: state) find $\ddot{\theta}$
  - calculate joint acceleration
  - useful for simulation
  - $\tau \rightarrow \ddot{\theta}$
- Inverse dynamics: Given $\left\{ \begin{array}{l} \cdot\ \theta \\ \cdot\ \dot{\theta} \\ \cdot\ \ddot{\theta} \end{array} \right.$ (θ, θ̇: state) find τ
  - find joint forces and torques
  - useful for control
  - $\ddot{\theta} \rightarrow \tau$

↳ second-order differential equation (equations of motion) in the form $\tau = M(\theta)\ddot{\theta} + h(\theta, \dot{\theta})$

2 approaches discussed:

- Lagrangian formulation: elegant, effective for simple robots
- Newton-Euler formulation: efficient recursive algorithms for forward and inverse dynamics

where
- θ: joint positions
- $\ddot{\theta}$: joint accelerations
- τ: forces and torques applied at each joint

$\tau \in \mathbb{R}^n$: vector of joint forces and torques

$\theta \in \mathbb{R}^n$ vector of joint variables

$M(\theta) \in \mathbb{R}^{n \times n}$: mass matrix

$h(\theta, \dot{\theta}) \in \mathbb{R}^n$:
- centripetal
- coriolis
- gravity
- friction

Lagrangian Formulation

$$\underbrace{L(\theta, \dot{\theta})}_{\text{Lagrangian}} = \underbrace{K(\theta, \dot{\theta})}_{\text{Kinetic Energy}} - \underbrace{P(\theta)}_{\text{potential Energy}}$$

$$\Rightarrow \tau = \frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta}$$

for the i-th element of τ:

$$\tau_i = \frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}_i} - \frac{\partial L}{\partial \theta_i}$$

8.1 Lagrangian Formulation

1.4.23

First: choose a set of <u>independent</u> coordinates $q \in \mathbb{R}^n$

↳ generalized coordinates

These generalized coordinates define <u>generalized forces</u> $f \in \mathbb{R}^n$

$\Rightarrow \quad f^T \dot{q} \mathrel{\hat{=}} \text{power}$

<u>Lagrangian function</u>:

$$\mathcal{L}(q,\dot{q}) = \underbrace{\mathcal{K}(q,\dot{q})}_{\text{kinetic energy}} - \underbrace{\mathcal{P}(q)}_{\text{potential energy}}$$

Equations of motions expressed in terms of the Lagrangian

$$f = \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}} - \frac{\partial \mathcal{L}}{\partial q}$$

Euler-Lagrange equations (with external forces)

For a n-link serial robot:

$$\mathcal{L}(\theta,\dot{\theta}) = \sum_{i=1}^{n} (\mathcal{K}_i - \mathcal{P}_i)$$

generalized coordinates: $\theta = (\theta_1, \dots \theta_n)$

generalized forces: $\tau = (\tau_1, \dots, \tau_n)$

$$\hookrightarrow \tau_i = \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\theta}_i} - \frac{\partial \mathcal{L}}{\partial \theta_i} \qquad i = 1,\dots,n$$

The resulting terms can be grouped:

$$\tau = M(\theta)\ddot{\theta} + c(\theta,\dot{\theta}) + g(\theta) + \left(+ J^T(\theta)\,\mathcal{F}_{tip}\right)$$

wrench that the end-effector applies to the environment

$M(\theta) \in \mathbb{R}^{n\times n}$ for n joints

where: • $M(\theta)$: positive-definite, symmetric <u>Mass</u> Matrix

• $c(\theta,\dot{\theta})$: vector containing <u>coriolis</u> and <u>centripetal</u> torques

• $g(\theta)$: vector containing <u>gravitational</u> torques

Equations of motion are:
- linear in $\ddot{\theta}$ (joint acceleration)
- quadratic in $\dot{\theta}$ (joint velocity)
- trigonometric in θ

Coriolis and Centripetal Torques:

$C(\theta, \dot{\theta})$: quadratic terms

- Centripetal terms have a square of a single joint velocity $\dot{\theta}_i^2$
- Coriolis terms have a product of two velocities $\dot{\theta}_i \cdot \dot{\theta}_j$ $i \neq j$
  different joint

$\ddot{\theta} = 0$ does not mean zero acceleration due to centripetal and Coriolis terms

Mass Matrix

Point mass: $\mathcal{K} = \frac{1}{2} m v^T v$

Generalization: $\mathcal{K} = \frac{1}{2} \dot{\theta}^T M(\theta) \dot{\theta}$

$M(\theta)$: positive definite (mass is always > 0)

- Point mass in Cartesian coordinates:

  $f = m\ddot{x}$

  - mass is independent of direction of acceleration
  - acceleration $\ddot{x}$ is always "parallel" to force f
- Mass matrix $M(\theta)$:
  - different effective mass in different directions

Effective mass at end-effector $\Lambda(\theta)$

$$\frac{1}{2} \dot{\theta}^T M(\theta) \dot{\theta} = \frac{1}{2} V^T \Lambda(\theta) V$$

$$\to \Lambda(\theta) = J^{-T}(\theta) M(\theta) J^{-1}(\theta)$$

General Formulation of the Lagrangian

For n-link open chains

$$\mathcal{L}(\theta,\dot\theta) = \mathcal{K}(\theta,\dot\theta) - P(\theta)$$

Kinetic energy (for rigid link robots):

$$\mathcal{K}(\theta,\dot\theta) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} m_{ij}(\theta)\dot\theta_i\dot\theta_j$$

$$= \frac{1}{2}\dot\theta^T M(\theta)\dot\theta$$

where $m_{ij}(\theta)$: Element of mass matrix $M \in \mathbb{R}^{n\times n}$

Dynamic equations:

$$\tau_i = \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot\theta_i} - \frac{\partial \mathcal{L}}{\partial \theta_i}, \quad i = 1,\dots,n$$

this can be expressed as:

$$\tau_i = \sum_{j=1}^{n} m_{ij}(\theta)\ddot\theta_j + \sum_{j=1}^{n}\sum_{k=1}^{n}\Gamma_{ijk}(\theta)\dot\theta_j\dot\theta_k + \frac{\partial P}{\partial\theta_i}, \quad i=1,\dots,n$$

where $\Gamma_{ijk}(\theta) = \frac{1}{2}\left(\frac{\partial m_{ij}}{\partial\theta_k} + \frac{\partial m_{ik}}{\partial\theta_j} - \frac{\partial m_{jk}}{\partial\theta_i}\right)$

$\Gamma_{ijk}(\theta)$: Christoffel symbols of the first kind

→ Coriolis and centripetal terms $c(\theta,\dot\theta)$, derived from $M(\theta)$

This can also be written as:

$$\tau = M(\theta)\ddot\theta + c(\theta,\dot\theta) + g(\theta)$$

where $g(\theta) = \frac{\partial P}{\partial\theta}$

or $\tau = M(\theta)\ddot\theta + h(\theta,\dot\theta)$

Modern Robotics

Lagrangian: General formulation

n-link open chains

1. select generalized coordinates $\theta \in \mathbb{R}^n$
   for configuration space
   - for open chains: choose θ to be vector of joint values

   ↳ generalized forces: $\tau \in \mathbb{R}^n$
   - if θ_i revolute $\Rightarrow \tau_i$: torque
   - if θ_i prismatic $\Rightarrow \tau_i$: force

2. Formulate Lagrangian $\mathcal{L}(\theta, \dot{\theta})$:

$$\mathcal{L}(\theta, \dot{\theta}) = \underbrace{\mathcal{K}(\theta, \dot{\theta})}_{\text{kinetic Energy}} - \underbrace{P(\theta)}_{\text{potential Energy}}$$

The dynamics equations can also be written with the Christoffel Symbols:

christoffel symbol

$$\tau = M(\theta)\ddot{\theta} + \underbrace{\dot{\theta}^T \Gamma(\theta)\dot{\theta}}_{} + g(\theta)$$

Coriolis and centripetal terms are quadratic in the velocity

where $\Gamma(\theta)$ is a $n \times n \times n$ matrix
and $\dot{\theta}^T\Gamma(\theta)\dot{\theta}$ is:

$$\dot{\theta}^T \Gamma(\theta)\dot{\theta} = \begin{bmatrix} \dot{\theta}^T \Gamma_1(\theta)\dot{\theta} \\ \dot{\theta}^T \Gamma_2(\theta)\dot{\theta} \\ \vdots \\ \dot{\theta}^T \Gamma_n(\theta)\dot{\theta} \end{bmatrix}$$

where $\Gamma_i(\theta)$ is a $n \times n$ matrix with (j,k)th entry of Γ_{ijk}.

It's also common to write:

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

where $C(\theta,\dot{\theta}) \in \mathbb{R}^{n \times n}$: Coriolis matrix
with entries

$$c_{ij} = \sum_{k=1}^{n} \Gamma_{ijk}(\theta)\dot{\theta}_k$$

The Coriolis matrix can be used to prove passivity property (used to prove stability of certain robot control laws).

Christoffel symbols (summary):

$\Gamma(\theta)$: $n \times n \times n$ matrix

$\Gamma_i(\theta)$: $n \times n$ matrix

$\Gamma_{ijk}(\theta)$: (j,k)th element of $\Gamma_i(\theta)$

$$\Gamma_{ijk}(\theta) = \frac{1}{2}\left(\frac{\partial m_{ij}}{\partial \theta_k} + \frac{\partial m_{ik}}{\partial \theta_j} - \frac{\partial m_{jk}}{\partial \theta_i}\right)$$

$$\dot{\theta}^T \Gamma(\theta)\dot{\theta} = \begin{bmatrix} \dot{\theta}^T \Gamma_1(\theta)\dot{\theta} \\ \dot{\theta}^T \Gamma_2(\theta)\dot{\theta} \\ \vdots \\ \dot{\theta}^T \Gamma_n(\theta)\dot{\theta} \end{bmatrix}$$

Analogy:

	constant mass (scalar)	mass changing over time
momentum	$p = m\dot{x}$	$p = m(x(t)) \cdot \dot{x}$
force	$f = \frac{dp}{dt} = m\ddot{x}$	$f = \frac{dp}{dt} = m\ddot{x} + \underbrace{\frac{\partial m}{\partial x}}_{\text{same role as Christoffel symbols for mass matrix}} \dot{x}\dot{x}$

Dynamic Equations (summary)

$$\tau = M(\theta)\ddot{\theta} + c(\theta,\dot{\theta}) + g(\theta) + J^T(\theta)F_{tip}$$

$$= M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + g(\theta) + J^T(\theta)F_{tip}$$

$$= M(\theta)\ddot{\theta} + h(\theta,\dot{\theta}) + J^T(\theta)F_{tip}$$

$$= M(\theta)\ddot{\theta} + \dot{\theta}^T\Gamma(\theta)\dot{\theta} + g(\theta) + J^T(\theta)F_{tip}$$

with $M(\theta)$: $n \times n$ mass matrix (positive definitive)

$g(\theta)$: gravity (potential) terms

$\Gamma(\theta)$: $n \times n \times n$ tensor of Christoffelsymbols

$C(\theta,\dot{\theta})$: Coriolis matrix

c: velocity-product terms

$h(\theta,\dot{\theta}) = c(\theta,\dot{\theta}) + g(\theta)$

$$\tau = \underbrace{M(\theta)\ddot{\theta} + c(\theta,\dot{\theta})}_{f\,=\,m\cdot a} + g(\theta) + J^T(\theta)F_{tip}$$

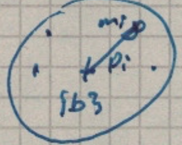
Newton-Euler Dynamics

18.4.23

Newton-Euler method is derived from $f = m \cdot a$ for rigid bodies of robot.
↳ recursive algorithm

Single Rigid Body

Centre of mass of a rigid body:
sum of mass-weighted vectors to the point masses in the $\{b\}$ frame is zero



$$\sum_i m_i p_i = 0$$

origin of $\{b\}$ frame is at centre of mass

Twist of rigid body:

$$\mathcal{V}_b = \begin{bmatrix} \omega_b \\ v_b \end{bmatrix}$$

linear velocity of point mass i:

$$\dot{p}_i = v_b + \omega_b \times p_i$$

acceleration: $$\ddot{p}_i = \dot{v}_b + \frac{d\omega_b}{dt} \times p_i + \omega_b \times \underbrace{\frac{d p_i}{dt}}_{\dot{p}_i}$$

$$= \dot{v}_b + \dot{\omega}_b \times p_i + \underbrace{\omega_b \times (v_b + \omega_b \times p_i)}_{\text{velocity-product terms}}$$

$f = m \cdot a$ mass
↳ $f_i = m_i \ddot{p}_i$
moment $m_i = [p_i] f_i$
moment

Wrench

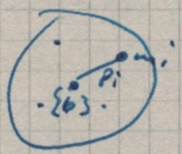
$$\mathcal{F}_b = \begin{bmatrix} m_b \\ f_b \end{bmatrix} = \begin{bmatrix} \sum_i m_i \\ \sum_i f_i \end{bmatrix} = \begin{bmatrix} I_b \dot{\omega}_b + [\omega_b] I_b \omega_b \\ m(\dot{v}_b + [\omega_b] v_b) \end{bmatrix}$$

I_b : inertia matrix (rotational)

$$I_b = -\sum_i m_i [p_i]^2 \in \mathbb{R}^{3 \times 3}$$

Inertia Matrix:

$$I_b = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{xy} & I_{yy} & I_{yz} \\ I_{xz} & I_{yz} & I_{zz} \end{bmatrix}$$



with: $I_{xx} = \sum m_i (y_i^2 + z_i^2)$

$I_{yy} = \sum m_i (x_i^2 + z_i^2)$

$I_{zz} = \sum m_i (x_i^2 + y_i^2)$

$I_{xy} = -\sum m_i \, x_i \cdot y_i$

$I_{xz} = -\sum m_i \, x_i \cdot z_i$

$I_{yz} \quad -\sum m_i \, y_i \cdot z_i$

same with mass density $\rho(x,y,z)$

$$I_{xx} = \int_B (y^2+z^2)\rho(x,y,z)\,dV$$

$$I_{yy} = \int_B (x^2+z^2)\rho(x,y,z)\,dV$$

$$I_{zz} \quad \int_B (x^2+y^2)\rho(x,y,z)\,dV$$

$$I_{xy} = -\int_B xy\,\rho(x,y,z)\,dV$$

$$I_{xz} = -\int_B xz\,\rho(x,y,z)\,dV$$

$$I_{yz} \quad -\int_B yz\,\rho(x,y,z)\,dV$$

See Book for examples of some common bodies

Inertia Matrix I_b is:

- Symmetric
- positive definite

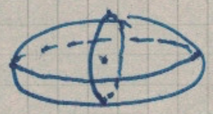
Rotational Kinetic Energy

$$KE = \frac{1}{2}\omega_b^T I_b \omega_b$$

Euler's equation for rotating rigid body

$$m_b = I_b \dot{\omega}_b + [\omega_b] I_b \omega_b$$

Ellipsoid (rigid body): 18.4.22



$$I_b = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{xy} & I_{yy} & I_{yz} \\ I_{xz} & I_{yz} & I_{zz} \end{bmatrix}$$

if frame $\{p\}$ is alligned with the principal axes of the ellipsoid the inertia matrix has a simple form:

$$I_p = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}$$

$\{p\}$ frame coordinate axes: <u>principal axes of inertia</u>
scalar inertias about those axes are <u>principal moments of inertia</u>

Find principal axes of inertia for any frame $\{b\}$:
eigenvalues and eigenvectors of I_b:

- v_1, v_2, v_3 = eigen-vectors(I_b)
- $\lambda_1, \lambda_2, \lambda_3$ = eigen-values(I_b)

- <u>principal axes</u> of inertia are aligned with the <u>eigen-vectors</u> expessed in $\{b\}$ frame.
- <u>principal moments</u> are the <u>eigenvalues</u>
- Rotation matrix expressing $\{p\}$ frame in $\{b\}$ frame has eigenvectors as columns

$$R_{bp} = [v_1 \; v_2 \; v_3] \text{ (right handed)}$$

kinetic energy: $\underbrace{\frac{1}{2}\omega_p^T I_p \omega_p}_{\text{in } \{p\} \text{ frame}} = \underbrace{\frac{1}{2}\omega_b^T I_b \omega_b}_{\text{in } \{b\} \text{ frame}}$

$\hookrightarrow I_p = R_{bp}^T I_b R_{bp}$

when possible: chose body frame aligned with pricipal axes of inertia to simplify the inertia matrix.

General $\{b\}$:

$$m_b = I_b \dot{\omega}_b + [\omega_b] I_b \omega_b$$

$\{b\}$ aligned with principal axes of inertia :

$$m_b = \begin{bmatrix} I_{xx} \dot{\omega}_x + (I_{zz} - I_{yy}) \omega_y \omega_z \\ I_{yy} \dot{\omega}_y + (I_{xx} - I_{zz}) \omega_x \omega_z \\ I_{zz} \dot{\omega}_z + (I_{yy} - I_{xx}) \omega_x \omega_y \end{bmatrix}$$

this form is simpler and involves less calculation

Summary:

Equations of motions for a single rigid body

$$\mathcal{F}_b = \begin{bmatrix} m_b \\ f_b \end{bmatrix} = \begin{bmatrix} I_b \dot{\omega}_b + [\omega_b] I_b \omega_b \\ m (\dot{v}_b + [\omega_b] v_b) \end{bmatrix}$$

$$= \begin{bmatrix} I_b & 0 \\ 0 & mI \end{bmatrix} \begin{bmatrix} \dot{\omega}_b \\ \dot{v}_b \end{bmatrix} + \begin{bmatrix} [\omega_b] & 0 \\ 0 & [\omega_b] \end{bmatrix} \begin{bmatrix} I_b & 0 \\ 0 & mI \end{bmatrix} \begin{bmatrix} \omega_b \\ v_b \end{bmatrix}$$

Changing Reference Frame

Given: frame $\{c\}$ described by R_{bc} (relative to $\{b\}$)

$$I_c = R_{bc}^T I_b R_{bc}$$

19.4.23

$$\begin{bmatrix} m_b \\ f_b \end{bmatrix} = \underbrace{\begin{bmatrix} I_b & 0 \\ 0 & mI \end{bmatrix}}_{G_b \in \mathbb{R}^{6\times 6}\text{ : spacial inertia matrix}} \begin{bmatrix} \dot\omega_b \\ \dot v_b \end{bmatrix} + \begin{bmatrix} [\omega_b] & 0 \\ 0 & [\omega_b] \end{bmatrix} \begin{bmatrix} I_b & 0 \\ 0 & mI \end{bmatrix} \begin{bmatrix} \omega_b \\ v_b \end{bmatrix}$$

$$\text{kinetic energy} = \underbrace{\frac{1}{2}\omega_b^T I_b \omega_b}_{\text{rotational kinetic energy}} + \underbrace{\frac{1}{2} m v_b^T v_b}_{\text{linear kinetic energy}}$$

$$= \frac{1}{2} V_b^T G_b V_b$$

$G_b \in \mathbb{R}^{6\times 6}$:

- symmetric
- positive definite

Lie bracket

Analogous operation as crossproduct for $\mathbb{R}^3$ vectors for $\mathbb{R}^6$ twists:

vector form

$\omega_1 \times \omega_2 = [\omega_1]\omega_2 \in \mathbb{R}^3$

$\underbrace{[ad_{V_1}] V_2 \in \mathbb{R}^6}_{\text{Lie bracket of } V_1 \text{ and } V_2}$

matrix form

$[\omega_1 \times \omega_2] = [\omega_1][\omega_2] - [\omega_2][\omega_1] \in so(3)$

$[V_1][V_2] - [V_2][V_1] \in se(3)$

$$[ad_V] = \begin{bmatrix} [\omega] & 0 \\ [v] & [\omega] \end{bmatrix} \in \mathbb{R}^{6\times 6}$$

Lie bracket of V_1 and V_2 ($[ad_{V_1}]V_2$) is an acceleration, measuring how motion along the twist V_2 would change if the body follows twist V_1.

$$\mathcal{F}_b = G_b \dot{\mathcal{V}}_b \underbrace{- [\mathrm{ad}_{\mathcal{V}_b}]^T G_b \mathcal{V}_b}^{\text{velocity-product term}}$$: inverse dynamics for a rigid body

The equation of motion in a different ({a}) frame

$$\frac{1}{2} \mathcal{V}_a^T G_a \mathcal{V}_a = \frac{1}{2} \mathcal{V}_b^T G_b \mathcal{V}_b$$

↳ $$\mathcal{F}_b = G_b \dot{\mathcal{V}}_b - [\mathrm{ad}_{\mathcal{V}_b}]^T G_b \mathcal{V}_b$$

$$\mathcal{F}_a = G_a \dot{\mathcal{V}}_a - [\mathrm{ad}_{\mathcal{V}_a}]^T G_a \mathcal{V}_a$$

Forward dynamics

$$\dot{\mathcal{V}}_b = G_b^{-1} \left(\mathcal{F}_b + [\mathrm{ad}_{\mathcal{V}_b}]^T G_b \mathcal{V}_b \right)$$

Analogy <u>rotating</u> rigid body and <u>linear motion</u> of rigid body:

$$\mathcal{F}_b = G_b \dot{\mathcal{V}}_b - [\mathrm{ad}_{\mathcal{V}_b}]^T G_b \mathcal{V}_b$$

is analogous to the equation for a rotating rigid body, replacing:

- Rotational Inertia Matrix I_b with Spacial Inertia Matrix G_b
- Angular velocity ω_b with Twist $\mathcal{V}_b$
- Cross-product with ω_b ($[\omega_b]$) with Lie bracket with $\mathcal{V}_b$ ($[\mathrm{ad}_{\mathcal{V}_b}]$)

↳ $$m_b = I_b \dot{\omega}_b + [\omega_b] I_b \omega_b$$

(Euler equation for rotating rigid body)

Twist, Wrench, Spacial Inertia and Spacial Momentum

a) Twist: $\mathcal{V}_b = \begin{bmatrix} \omega_b \\ v_b \end{bmatrix} \in \mathbb{R}^6$

Wrench: $\mathcal{F}_b = \begin{bmatrix} m_b \\ f_b \end{bmatrix} \in \mathbb{R}^6$

b) Spacial Inertia Matrix

$$G_b = \begin{bmatrix} I_b & 0 \\ 0 & mI \end{bmatrix} \in \mathbb{R}^{6\times 6}$$

side Note: kinetic Energy =

$$\frac{1}{2}\omega_b^T I_b \omega_b + \frac{1}{2} m v_b^T v_b = \frac{1}{2}\mathcal{V}_b^T G_b \mathcal{V}_b$$

c) Spacial Momentum

$$P_b = \begin{bmatrix} I_b\omega_b \\ m v_b \end{bmatrix} = \begin{bmatrix} I_b & 0 \\ 0 & mI \end{bmatrix} \begin{bmatrix} \omega_b \\ v_b \end{bmatrix} = G_b \mathcal{V}_b \in \mathbb{R}^6$$

↳ Dynamics of a single rigid body (Twist-Wrench Formulation)

$$\mathcal{F}_b = G_b \dot{\mathcal{V}}_b - ad^T_{\mathcal{V}_b}(P_b)$$
$$= G_b \dot{\mathcal{V}}_b - [ad_{\mathcal{V}_b}]^T G_b \mathcal{V}_b$$

Dynamics in other Frames (than body frame)

$$G_a = [Ad_{T_{ba}}]^T G_b [Ad_{T_{ba}}]$$

$$\mathcal{F}_a = G_a \dot{V}_a - [ad_{V_a}]^T G_a V_a$$

$$\Leftrightarrow \quad \mathcal{F}_b = G_b \dot{V}_b - [ad_{V_b}]^T G_b V_b$$

Summary

Inverse dynamics for a rigid body:

$$\mathcal{F}_b = G_b \dot{V}_b - [ad_{V_b}]^T G_b V_b$$

Forward dynamics for a rigid body:

$$\dot{V}_b = G_b^{-1}(\mathcal{F}_b + [ad_{V_b}]^T G_b V_b)$$

20.4.23

Newton-Euler inverse dynamics algorithm

For open-chain robot
inverse dynamics: useful for robot control

Calculate $\tau = M(\theta)\ddot{\theta} + c(\theta, \dot{\theta}) + g(\theta) + J^T(\theta) \mathcal{F}_{tip}$

given
- θ: joint positions
- $\dot{\theta}$: joint velocities
- $\ddot{\theta}$: joint accelerations
- $\mathcal{F}_{tip}$: wrench that end-effector applies to environment

- velocities and accelerations depend on the previous links (forward iterations)
- Forces and moments depend on following links (backward iterations)

Define:

- $M_{i,i-1}$: $\{i-1\}$ in $\{i\}$ when $\theta_i = 0$
  - ↳ transform frame $\{i-1\}$ relative to frame $\{i\}$ when joint is at zero position
- A_i: screw axis of joint i in $\{i\}$
- $\mathcal{F}_{n+1}$: $\mathcal{F}_{tip}$ applied by end-effector
- $\dot{\mathcal{V}}_0 = (\dot{\omega}_0, \dot{v}_0) = (0, -g)$
  - ↳ linear acceleration opposite the gravity vector (gravity is indistinguishable from upwards acceleration)

1. Forward iterations (from base to end-effector)

Given $\theta, \dot{\theta}, \ddot{\theta}$, for $i = 1$ to n do:

- $T_{i,i-1} = e^{-[A_i]\theta_i} M_{i,i-1}$: configuration of frame $\{i-1\}$ relative to frame $\{i\}$
- $\mathcal{V}_i = [Ad_{T_{i,i-1}}] \mathcal{V}_{i-1} + A_i \dot{\theta}_i$: twist of link i is the sum:
  - twist of link $i-1$ (expressed in frame $\{i\}$)
  - \+ twist due to joint velocity
- $\dot{\mathcal{V}}_i = [Ad_{T_{i,i-1}}] \dot{\mathcal{V}}_{i-1} + [ad_{\mathcal{V}_i}] A_i \dot{\theta}_i + A_i \ddot{\theta}_i$: acceleration of link $\{i\}$ is the sum:
  - acceleration of link $i-1$ expressed in $\{i\}$ frame
  - \+ acceleration due to velocity product term

At the end of forward iteration we have:
- configurations
- twists
- accelerations

of all links } expressed in center-of-mass frames $\{i\}$

2. Backward iterations (from end-effector to base)

For $i = n$ to 1 do:

- $\mathcal{F}_i = [Ad_{T_{i+1,i}}]^T \mathcal{F}_{i+1} + G_i \dot{\mathcal{V}}_i - [ad_{\mathcal{V}_i}]^T G_i \mathcal{V}_i$

  ↳ wrench $\mathcal{F}_i$ required by link {i}

  sum of:
  - wrench required by link {i+1} expressed in {i}
  - \+ wrench required by link {i} due to dynamics of a single rigid body

- $\tau_i = \mathcal{F}_i^T A_i$: joint torque for joint i:
  - projecting wrench $\mathcal{F}_i$ on screw axis A_i

We now have vector $\underline{\tau}$ of all joint forces and torques needed for a given
- θ
- $\dot{\theta}$
- $\ddot{\theta}$
- $\mathcal{F}_{tip}$

Forward Dynamics (Newton-Euler) 20.4.23

Usefulfor simulation

Solve: can be $\emptyset$

$$M(\theta)\ddot{\theta} = \tau - c(\theta,\dot{\theta}) - g(\theta) - J^T(\theta)\mathcal{F}_{tip}$$

for $\ddot{\theta}$. Given $\theta, \dot{\theta}, \tau$ and wrench $\mathcal{F}_{tip}$

Using inverse dynamics algorithm (Newton-Euler):

$$\tau = M(\theta)\ddot{\theta} + \overbrace{c(\theta,\dot{\theta}) + g(\theta)}^{h(\theta,\dot{\theta})} + J^T(\theta)\mathcal{F}_{tip}$$

1. Solve for (using NE inverse dynamics algorithm) $\underbrace{c(\theta,\dot{\theta}) + g(\theta)}_{h(\theta,\dot{\theta})} + J^T(\theta)\mathcal{F}_{tip}$ by setting $\ddot{\theta} = 0$ and $\mathcal{F}_{tip} = 0$
2. Solve for (using NE inv. dyn. algo.)

$$M(\theta) = [M_1(\theta) \dots M_n(\theta)]$$

where $\tau = M_i(\theta)$ if $\ddot{\theta}_i = 1$, $\ddot{\theta}_j = 0$ for all $j \neq i$, $\dot{\theta} = 0$, $g = 0$, $\mathcal{F}_{tip} = 0$

↳ call inverse dynamics algorithm n times (for n joints)
- each time, exept for one joint reset!
  - joint acceleration $\ddot{\theta}_j = 0$ $j \neq i$
  - ith joint acceleration $\ddot{\theta}_i = 1$
- construct $M(\theta)$ from $M_i(\theta)$ as columns

> See book for more detailed description.

⇒ with $M(\theta)$, $h(\theta,\dot{\theta})$ and $\mathcal{F}_{tip}$ solve

$$M(\theta)\ddot{\theta} = \tau(t) - h(\theta,\dot{\theta}) - J^T(\theta)\mathcal{F}_{tip}$$

for $\ddot{\theta}$ (it has the form $M\ddot{\theta} = b$ solve for $\ddot{\theta}$)

- this can be solved by numerical integration with Euler iteration, Runge Kutta . . .

Simulation of Robot motion with forward dynamics

The motion of a robot can be simulated with the forward dynamics:

Given

- Robots initial state
- joint forces-torques $\tau(t)$
- external wrench $\mathcal{F}_{tip}$ (optional)

for time $t \in [0, t_f]$

- Define function 'ForwardDynamics' that returns solution to $M(\theta)\ddot{\theta} = \tau(t) - h(\theta, \dot{\theta}) - J^T(\theta)\mathcal{F}_{tip}$

i.e. $\ddot{\theta} = \text{ForwardDynamics}(\theta, \dot{\theta}, \tau, \mathcal{F}_{tip})$

with $q_1 = \theta$

$q_2 = \dot{\theta}$

and

to

$$\dot{q}_1 = q_2$$

$$\dot{q}_2 = \text{ForwardDynamics}(q_1, q_2, \tau, \mathcal{F}_{tip})$$

integrate (numerically) system of first order differential equation of the form $\dot{q} = f(q, t), q \in \mathbb{R}^n$.

e.g. first-order Euler iteration:

$$q(t + \delta t) = q(t) + \delta t f(q(t), t)$$

δt: timestep

Euler iteration of robot dynamics:

$$q_1(t + \delta t) = q_1(t) + q_2(t)\delta t$$

$$q_2(t + \delta t) = q_2(t) + \underbrace{\text{ForwardDynamics}(q_1, q_2, \tau, \mathcal{F}_{tip})}_{\dot{q}_2} \cdot \delta t$$

given set of initial values:

- $q_1(0) = \theta(0)$
- $q_2(0) = \dot{\theta}(0)$

iterate above equations to obtain the motion $\theta(t) = q_1(t)$

Dynamics in the Task Space

S.S. 23

Robot Dynamics can also be expressed in End-Effector motions and wrenches (task space) instead of joint motions and joint forces and torques. (joint-space)

joint-space dynamics (for 6 dof open chain) *

$$\tau = M(\theta)\ddot{\theta} + h(\theta, \ddot{\theta}) \qquad \theta \in \mathbb{R}^6,\ \tau \in \mathbb{R}^6$$

assume: $\mathcal{V} = J(\theta)\dot{\theta}$, J invertible, $\mathcal{V} = (\omega, v)$

$\mathcal{V}$ and J can be either in
- Space frame

or
- end-effector frame

then: $\dot{\mathcal{V}} = J(\theta)\ddot{\theta} - \dot{J}(\theta)\dot{\theta}$

↳ solve

$$\dot{\theta} = J^{-1}\mathcal{V}$$

$$\ddot{\theta} = J^{-1}\dot{\mathcal{V}} - J^{-1}\dot{J}J^{-1}\mathcal{V}$$

↳ insert into joint-space dynamics

$$\mathcal{F} = \Lambda(\theta)\dot{\mathcal{V}} + \eta(\theta, \mathcal{V})$$

with $\Lambda(\theta) = J^{-T}M(\theta)J^{-1}$

$$\eta(\theta, \mathcal{V}) = J^{-T}h(\theta, J^{-1}\mathcal{V}) - \Lambda(\theta)\dot{J}J^{-1}\mathcal{V}$$

expressed in terms of joint positions θ, not in end-effector configuration, since there could be several robot config. for a end-effector config.

$\Lambda(\theta)$: Mass matrix in task-space

$\eta(\theta, \mathcal{V})$: Sum of velocity-product and gravity terms expressed as end-effector wrench

If end-effector applies a wrench $\mathcal{F}_{tip}$ it is added to the total wrench.

Constrained Dynamics

The Robots motion can be subject constraints.

- n-joint robot
- k pfaffian (holonomic or nonholonomic) constraints of the form

$$A(\theta)\dot{\theta} = 0, \quad A(\theta) \in \mathbb{R}^{k \times n}$$

Constraints can come from loop-closure constraints.

Assume the constraints do no work on the robot:

↳ forces that enforce the constraints do no work on robot (workless)

i.e. generalized forces:

$$\tau_{con}^T \dot{\theta} = 0$$

this means τ_{con} must be linear combination of the columns of $A^T(\theta)$ i.e.

↳ $\tau_{con} = A^T(\theta)\lambda$ for some $\lambda \in \mathbb{R}^k$

$$\Rightarrow (A^T(\theta)\lambda)^T\lambda)^T \dot{\theta} = \lambda^T A(\theta)\dot{\theta} = 0 \quad \text{for all } \lambda \in \mathbb{R}^k$$

Adding constrained forces $A^T(\theta)\lambda$ to the equations of motion (n+k constrained equation of motion):

Can be used to solve constrained inverse dynamics for a given λ and a $\ddot{\theta}$ chosen from the (n-k)-dimensional space of accelerations satisfying $\dot{A}(\theta)\dot{\theta} + A(\theta)\ddot{\theta} = 0$

$$\tau = \overbrace{M(\theta)\ddot{\theta} + h(\theta, \dot{\theta})}^{\text{joint forces/torques to move robot}} + \overbrace{\underbrace{A^T(\theta)\lambda}_{\tau_{con}}}^{\text{joint forces/torques against constraints}}$$

$$\underbrace{A(\theta)\dot{\theta}}_{\tau_{con}^T\dot{\theta} = 0} = 0 \Rightarrow A(\theta)\ddot{\theta} + \dot{A}(\theta)\dot{\theta} = 0$$

n+k equations in n+k variables, λ and either $\ddot{\theta}$ or τ — depending if forward or inverse dynamics

- λ is a set of <u>Lagrange multipliers</u>
- $A^T(\theta)\lambda$: forces applied against the constraints as expressed in joint forces and torques.

The robot has • n-k velocity freedoms
• k "force freedoms"

The constraints allow the robot to create any generalized force in the form of $A^T(\theta)\lambda$ independent of the robots motion.

Since constraints $A(\theta)\dot{\theta} = 0$ are satisfied all time the time rate of change of constraints satisfy:

$$\dot{A}(\theta)\dot{\theta} + A(\theta)\ddot{\theta} = 0$$

Assuming $M(\theta)$ and $A(\theta)$ are full rank.
Solve constrained equations of motion

$$\tau = M(\theta)\ddot{\theta} + h(\theta, \dot{\theta}) + A^T(\theta)\lambda$$

for $\ddot{\theta}$.

↳ $\ddot{\theta} = M^{-1}(\theta)(\tau - h(\theta, \dot{\theta}) - A^T(\theta)\lambda)$

$$\dot{A}\dot{\theta} + AM^{-1}(\tau - h - A^T\lambda) = 0$$

(dependences on $\dot{\theta}$ and θ are omitted)

↳ Solve for Lagrange Multipliers:

$$\lambda = (AM^{-1}A^T)^{-1}(AM^{-1}(\tau - h) + \dot{A}\dot{\theta})$$

Constraint force depends on τ and the state
If constraint acts at end-effector λ is related to end-effector wrench applied to the constraint by

$$J^T(\theta)\mathcal{F}_{tip} = A^T(\theta)\lambda$$

- $J(\theta)$: Jacobian satisfying $\mathcal{V} = J(\theta)\dot{\theta}$
  if $J(\theta)$ is invertible: $\mathcal{F}_{tip} = J^{-T}(\theta)A^T(\theta)\lambda$

Projection Matrix (to eliminate the Lagrange multipliers):

$$P(\theta) = I - A^T(AM^{-1}A^T)^{-1}AM^{-1} \quad \in \mathbb{R}^{n\times n}, \text{ rank } n-k$$

↳ constrained inverse dynamics:

$$P\tau = P(M\ddot{\theta} + h)$$

P projects joint torques τ to the joint torques that moves the robot eliminating the joint torques against the constraints that cause no motion.