Dynamics deals with the forces required to cause motion. Mainly to problems/queries are solved:

1) Given a target pose, which forces do we need at the joints to achieve it?

→ important for CONTROL

$$(\theta, \dot\theta, \ddot\theta)_i \xrightarrow[\text{INVERSE dynamics}]{1} (\tau)_i$$

$$(\theta, \dot\theta, \ddot\theta)_i \xleftarrow[2]{\text{FORWARD dynamics}} (\tau)_i$$

Joint positions, velocities, accelerations

Forces/Torques at joints

2) If we apply some joint forces/torques, which motion will be caused?

important for SIMULATION

1 Acceleration of a Rigid Body

Following the same definitions as in chapter 5...

$$^A\vec{V}_Q = {}^A\vec{V}_{BORG} + {}^A_B\underset{\sim}{R}\,{}^B\vec{V}_Q + {}^A\vec{\Omega}_B \times {}^A_B R\,{}^B\vec{Q}$$

$\downarrow d/dt$

this term is $\emptyset$ if ${}^BQ$ is constant, ie.: ${}^B\vec{V}_Q = {}^B\dot{\vec{V}}_Q = 0$

$$\boxed{\begin{aligned} {}^A\dot{\vec{V}}_Q = {}^A\dot{\vec{V}}_{BORG} &+ {}^A_B\underset{\sim}{R}\,{}^B\dot{\vec{V}}_Q + 2\,{}^A\vec{\Omega}_B \times {}^A_B\underset{\sim}{R}\,{}^B\vec{V}_Q \\ &+ {}^A\dot{\vec{\Omega}}_B \times {}^A_B\underset{\sim}{R}\,{}^B\vec{Q} + {}^A\vec{\Omega}_B \times \left({}^A\vec{\Omega}_B \times {}^A_B R\,{}^B\vec{Q}\right) \end{aligned}}$$

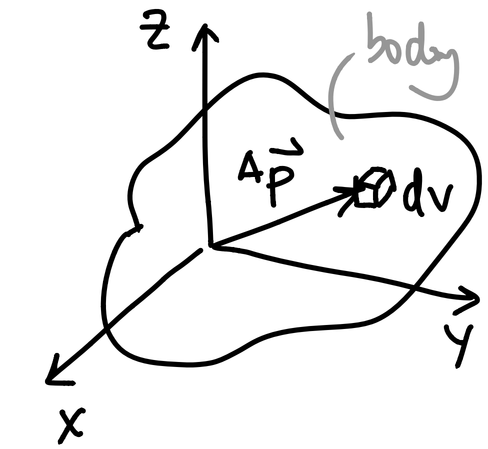
Linear acceleration of a rigid body: acceleration field of a point Q embedded in {B}, as seen from {A}.

Now, we consider the case in which $\{B\}$ is rotating wrt $\{A\}$ and $\{C\}$ is rotating relative to $\{B\}$; then:

$$\boxed{{}^{B}\dot{\vec{\Omega}}_C = {}^{A}\dot{\vec{\Omega}}_B + {}^{A}\underset{\sim}{R}_B \, {}^{B}\dot{\vec{\Omega}}_C + {}^{A}\vec{\Omega}_B \times {}^{A}\underset{\sim}{R}_B \, {}^{B}\vec{\Omega}_C}$$

② Mass Distribution

Inertia tensor: generalization of the scalar moment of inertia



frame $\{A\}$, usually attached to the body

$${}^{A}\underset{\sim}{I} = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix}$$

Inertia tensor of a body expressed in frame $\{A\}$

mass products of inertia

$$I_{xy} = \iiint_V xy\rho \, dv$$

mass moments of inertia

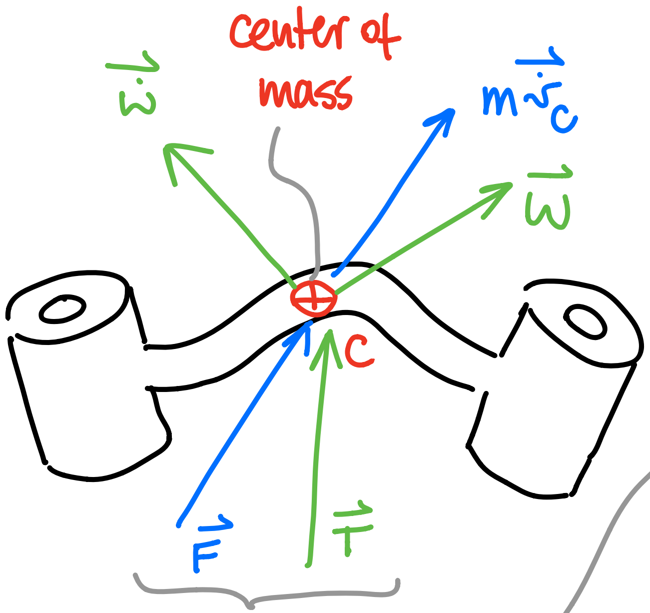
$$I_{xx} = \iiint_V (y^2+z^2)\rho \, dv$$

the eigen-values and vectors are the Principal Moments & Axes

Parallel-axis theorem: relationship of $\underset{\sim}{I}$ in two parallel (translated) systems $\{A\}$ and $\{C\}$, where C is the center of mass

$$\boxed{{}^{A}\underset{\sim}{I} = {}^{C}\underset{\sim}{I} + m\left[\vec{P}_c \cdot \vec{P}_c \underbrace{\mathbb{I}_3}_{\text{eye}(3)} - \vec{P}_c \cdot \vec{P}_c\right]}, \quad \vec{P}_c = (x_c, y_c, z_c)^T : \text{center of mass rel. to } \{A\}$$

③ Newton-Euler Equations of Motion



sum of all forces and torques expressed in the center of mass c

$$\boxed{\vec{F} = m \cdot \dot{\vec{v}}_c}$$

mass scales the linear acceleration

$$\boxed{\vec{T} = \underbrace{{}^{C}\underset{\sim}{I}\,\dot{\vec{\omega}}}_{\text{angular acceleration is projected with } {}^{C}\underset{\sim}{I}} + \underbrace{\vec{\omega} \times {}^{C}\underset{\sim}{I} \cdot \vec{\omega}}_{\text{component } \perp \text{ to } \vec{\omega}}}$$

④ Iterative Newton-Euler Dynamic Formulation: Obtain Joint Torques Necessary for Join Motion

- We have the target motion : $\theta, \dot{\theta}, \ddot{\theta}$ INVERSE dynamics
- We want to compute the joint forces & torques for that
- The iterative algorithm below computes that numerically for a robot arm with R joints using the formulas defined so far. 2 steps:

  ① Link velocities, accelerations and Newton-Euler equations are defined from link $0 \longrightarrow 5$

  ② Forces and torques necessary in the links computed: $6 \rightarrow 1$

1 Outward iterations: $i : 0 \to 5$

$$ {}^{i+1}\omega_{i+1} = {}^{i+1}_{i}R\,{}^{i}\omega_i + \dot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1}, \tag{6.45} $$

$$ {}^{i+1}\dot{\omega}_{i+1} = {}^{i+1}_{i}R\,{}^{i}\dot{\omega}_i + {}^{i+1}_{i}R\,{}^{i}\omega_i \times \dot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1} + \ddot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1}, \tag{6.46} $$

$$ {}^{i+1}\dot{v}_{i+1} = {}^{i+1}_{i}R({}^{i}\dot{\omega}_i \times {}^{i}P_{i+1} + {}^{i}\omega_i \times ({}^{i}\omega_i \times {}^{i}P_{i+1}) + {}^{i}\dot{v}_i), \tag{6.47} $$

$$ \begin{aligned} {}^{i+1}\dot{v}_{C_{i+1}} &= {}^{i+1}\dot{\omega}_{i+1} \times {}^{i+1}P_{C_{i+1}} \\ &\quad + {}^{i+1}\omega_{i+1} \times ({}^{i+1}\omega_{i+1} \times {}^{i+1}P_{C_{i+1}}) + {}^{i+1}\dot{v}_{i+1}, \end{aligned} \tag{6.48} $$

Newton-Euler on link

$$ {}^{i+1}F_{i+1} = m_{i+1}\,{}^{i+1}\dot{v}_{C_{i+1}}, \tag{6.49} $$

$$ {}^{i+1}N_{i+1} = {}^{C_{i+1}}I_{i+1}\,{}^{i+1}\dot{\omega}_{i+1} + {}^{i+1}\omega_{i+1} \times {}^{C_{i+1}}I_{i+1}\,{}^{i+1}\omega_{i+1}. \tag{6.50} $$

2 Inward iterations: $i : 6 \to 1$

joint forces and torques

$$ {}^{i}f_i = {}^{i}_{i+1}R\,{}^{i+1}f_{i+1} + {}^{i}F_i, \tag{6.51} $$

$$ \begin{aligned} {}^{i}n_i &= {}^{i}N_i + {}^{i}_{i+1}R\,{}^{i+1}n_{i+1} + {}^{i}P_{C_i} \times {}^{i}F_i \\ &\quad + {}^{i}P_{i+1} \times {}^{i}_{i+1}R\,{}^{i+1}f_{i+1}, \end{aligned} \tag{6.52} $$

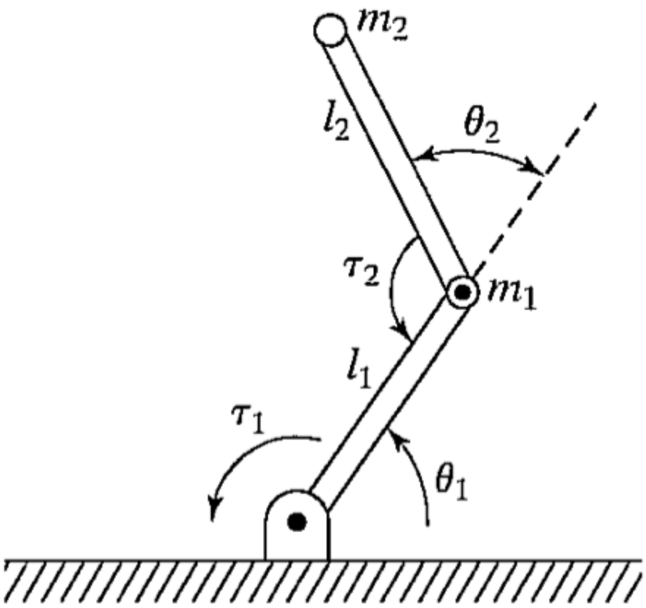
$$ \tau_i = {}^{i}n_i^T\,{}^{i}\hat{Z}_i. \tag{6.53} $$

→ projection on joint axis

Craig, 3rd Ed. p. 176

- To take into account the self-weight of all the links due to gravity, set $\boxed{{}^{0}\dot{\vec{v}}_0 = -\vec{g}}$
- This algorithm is very attractive, because it lets us compute numerically the forces necessary for any robot arm with R joints; we only need the geometric and mass params
- However, it is sometimes interesting to obtain a close-form solution to analyze the system behavior.

5 An Example of Closed-Form Dynamic Equations



Craig, 3rd Ed., p. 177

- Closed-form dynamic equations can be obtained following the previous section and substituting the geometric & mass components by params
- However, the complexity of the equations increases even for easy examples

... SOLUTION:

$$\tau_1 = m_2 l_2^2(\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 l_1 l_2 c_2 (2\ddot{\theta}_1 + \ddot{\theta}_2) + (m_1 + m_2) l_1^2 \ddot{\theta}_1 - m_2 l_1 l_2 s_2 \dot{\theta}_2^2$$
$$-2 m_2 l_1 l_2 s_2 \dot{\theta}_1 \dot{\theta}_2 + m_2 l_2 g c_{12} + (m_1 + m_2) l_1 g c_1,$$
$$\tau_2 = m_2 l_1 l_2 c_2 \ddot{\theta}_1 + m_2 l_1 l_2 s_2 \dot{\theta}_1^2 + m_2 l_2 g c_{12} + m_2 l_2^2 (\ddot{\theta}_1 + \ddot{\theta}_2). \tag{6.58}$$

Craig, 3rd Ed., p. 180

These equations can be written in matricial form and different relevant terms can be factored. Next section deals with that.

⑥ The Structure of a Manipulator's Dynamic Equations

The closed-form equations of a manipulator can be factored into matricial expressions that highlight details on the structure of the equations

6.1. The State-Space Equation

State = position + velocity: both appear in V

$$\boxed{\underbrace{\vec{\tau}}_{\text{joint torques}} = \underbrace{\underset{\sim}{M}(\vec{\Theta})}_{n\times n \text{ mass matrix}} \cdot \ddot{\vec{\Theta}} + \underbrace{\underset{\sim}{V}(\vec{\Theta}, \dot{\vec{\Theta}})}_{n\times 1 \text{ vector with 2 terms}} + \underbrace{\underset{\sim}{G}(\vec{\Theta})}_{n\times 1 \text{ vector for gravity terms}}}$$

n×n mass matrix

joint torques

n×1 vector with 2 terms:
- Centrifugal forces: square of joint velocities
- Coriolis forces: always product of 2 different joint velocities

n×1 vector for gravity terms: all with constant g

6.2. The Configuration-Space Equation

In this form, the $\underset{\sim}{V}(\vec{\Theta}, \dot{\vec{\Theta}})$ is factored in two terms and all matrices end up depending only on Θ, thus the configuration

$$\boxed{\vec{\tau} = \underset{\sim}{M}(\vec{\theta})\ddot{\vec{\theta}} + \overbrace{\underset{\sim}{B}(\vec{\theta})[\dot{\vec{\theta}}\dot{\vec{\theta}}] + \underset{\sim}{C}(\vec{\theta})[\dot{\vec{\theta}}^{2}]}^{\text{previous } \underset{\sim}{V}(\vec{\theta},\dot{\vec{\theta}})} + \underset{\sim}{G}(\vec{\theta})}$$

$\underset{\sim}{B}(\vec{\theta}) : n \times \frac{n(n-1)}{2}$

Coriolis coefficients

$[\dot{\vec{\theta}}\dot{\vec{\theta}}] = (\dot{\theta}_1\dot{\theta}_2, \dot{\theta}_1\dot{\theta}_3, \dots, \dot{\theta}_{n-1}\dot{\theta}_n)^T$

$\frac{n(n-1)}{2} \times 1$

vector of joint velocity pairs

$\underset{\sim}{C}(\vec{\theta}) : n \times n$

centrifugal coefficients

$[\dot{\vec{\theta}}^2] = (\dot{\theta}_1^2, \dot{\theta}_2^2, \dots, \dot{\theta}_n^2)^T$

$n \times 1$

This form of the equation of motion is very practical, because the equations are transformed into matrices with coefficients that only depend on the position! $\underset{\sim}{M}(\vec{\theta})$, $\underset{\sim}{B}(\vec{\theta})$, $\underset{\sim}{C}(\vec{\theta})$, $\underset{\sim}{G}(\vec{\theta})$

7 Lagrangian Formulation of Manipulator Dynamics

- We can derive the equations of motion with Lagrange
- While Newton-Euler focuses on the force balance, Lagrangian mechanics is energy-based

$$\boxed{\mathcal{L}(\vec{\theta},\dot{\vec{\theta}}) = \underbrace{\text{Kinetic energy}}_{\frac{1}{2}mv^2} - \underbrace{\text{Potential energy}}_{mgh} = K(\vec{\theta},\dot{\vec{\theta}}) - U(\vec{\theta})}$$

LAGRANGIAN

Particle equivalent formulas

$$\boxed{K(\vec{\theta},\dot{\vec{\theta}}) = \sum_{i=1}^{n} K_i} = \sum_{i=1}^{n} \underbrace{\frac{1}{2} m_i \vec{v}_{c_i}^{\,T} \vec{v}_{c_i} + \frac{1}{2}\, {}^{i}\vec{\omega}_i^{\,T}\, {}^{c_i}\underset{\sim}{I}\, {}^{i}\vec{\omega}_i}_{\text{kinetic energy of link } i}$$

KINETIC ENERGY

$$\boxed{= \frac{1}{2} \dot{\vec{\theta}}^{T} \underbrace{\underset{\sim}{M}(\vec{\theta})}_{\text{mass matrix}} \dot{\vec{\theta}}}$$

vector pointing to the center of mass of link i

$$\boxed{U(\vec{\theta}) = \sum_{i=1}^{n} u_i} = \sum_{i=1}^{n} \underbrace{-m_i\, {}^{0}\vec{g}^{\,T}\, {}^{0}\vec{p}_{c_i}(\vec{\theta})}_{mgh} + \underbrace{u_{ref_i}}$$

reference potential constant, it disappears, because we derivate $\mathcal{L}$

According to the Lagrangian formulation, the equations of motion are:

$$\boxed{\vec{\tau} = \frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}}\right) - \frac{\partial \mathcal{L}}{\partial \theta} = \frac{d}{dt}\left(\frac{\partial K(\vec{\theta},\dot{\vec{\theta}})}{\partial \dot{\theta}}\right) - \frac{\partial K(\vec{\theta},\dot{\vec{\theta}})}{\partial \theta} + \frac{\partial U(\vec{\theta})}{\partial \theta}}$$

We can apply this method and obtain the equivalent equations of motion.

8 Manipulator Dynamics in Cartesian Space

The state-space and configuration-space equations of motion are in joint-space (θ). We can transform these to Cartesian space using the Jacobian:

$$\vec{\tau} = \underset{\sim}{J}^T(\theta) \cdot \vec{F}$$

$$\underbrace{\dot{\vec{X}}}_{\text{velocity in Cartesian space}} = \underset{\sim}{J}(\theta) \cdot \dot{\vec{\theta}} \rightarrow \ddot{\vec{X}} = \dot{\underset{\sim}{J}}\dot{\vec{\theta}} + \underset{\sim}{J}\ddot{\vec{\theta}} \rightarrow \ddot{\vec{\theta}} = \underset{\sim}{J}^{-1}\ddot{\vec{X}} - \underset{\sim}{J}^{-1}\dot{\underset{\sim}{J}}\dot{\theta}$$

We substitute these and obtain $\underset{\sim}{M}_x(\vec{\theta})$, $V_x(\vec{\theta}, \dot{\vec{\theta}})$, $G_x(\vec{\theta})$ so that:

$$\boxed{\vec{F} = \underset{\sim}{M}_x(\theta)\ddot{\vec{X}} + \underset{\sim}{V}_x(\vec{\theta}, \dot{\vec{\theta}}) + \underset{\sim}{G}_x(\theta)}$$

9 Inclusion of Other Effects

The equations shown so far do not take into account many factors that might require >20% larger joint torques to produce the motion, such as:

- Friction in the joints — PROBABLY THE MOST IMPORTANT
- Inertia of the motor rotor

- The operating region of the motor: there is a V & I limit which restricts motor ω & τ; similarly there is a range of τ values achievable for a given ω, and vice-versa
- Joint and link flexibility: we have considered them infinitely stiff — but that's only theory...

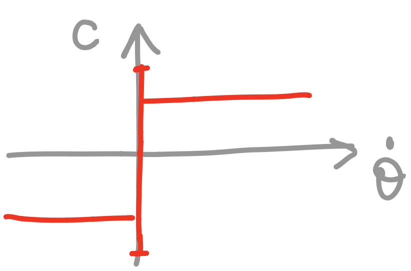
Friction Model

$$\boxed{\tau_{friction} = \underbrace{c \cdot \mathrm{sign}(\dot{\theta})}_{\text{Coulomb friction constant } c} + \underbrace{\nu \dot{\theta}}_{\text{Viscous friction}}}$$

BUT: in many cases we can observe:

$$\tau_{friction} = f(\theta, \dot{\theta})$$

dependent on joint position...



After characterizing the friction component, the equation of motion ends up with a new term:

$$\boxed{\vec{\tau} = \underset{\sim}{M}(\vec{\theta})\ddot{\vec{\theta}} + \underset{\sim}{V}(\vec{\theta}, \dot{\vec{\theta}}) + \underset{\sim}{G}(\vec{\theta}) + \underset{\sim}{F}(\vec{\theta}, \dot{\vec{\theta}})}$$

(10) Dynamic Simulation: Forward Dynamics

Forward dynamics consists in simulating the motion of a mechanism given the torques in its joints $\vec{\tau}$:

$$\boxed{\ddot{\vec{\theta}} = \underset{\sim}{M}(\vec{\theta})^{-1} \cdot \left(\vec{\tau} - \underset{\sim}{V}(\vec{\theta}, \dot{\vec{\theta}}) - \underset{\sim}{G}(\vec{\theta}) - \underset{\sim}{F}(\vec{\theta}, \dot{\vec{\theta}}) \right)}$$

With initial conditions:

$$\vec{\theta}(t=0) = \vec{\theta}_0 = (\theta_{0,0}, \theta_{1,0}, \theta_{2,0}, \ldots)^T$$

$$\dot{\vec{\theta}}(t=0) = \vec{\emptyset}$$

We can perform a Euler integration:

$$\left.\begin{aligned} \dot{\vec{\theta}}(t+\Delta t) &= \dot{\vec{\theta}}(t) + \ddot{\vec{\theta}}(t) \cdot \Delta t \\ \vec{\theta}(t+\Delta t) &= \vec{\theta}(t) + \dot{\vec{\theta}}(t)\, \Delta t + \frac{1}{2} \ddot{\vec{\theta}}(t) \cdot \Delta t^2 \end{aligned}\right\}$$

Note that $\Delta t \to 0$, and more sophisticated schemes than the Euler integration exist.