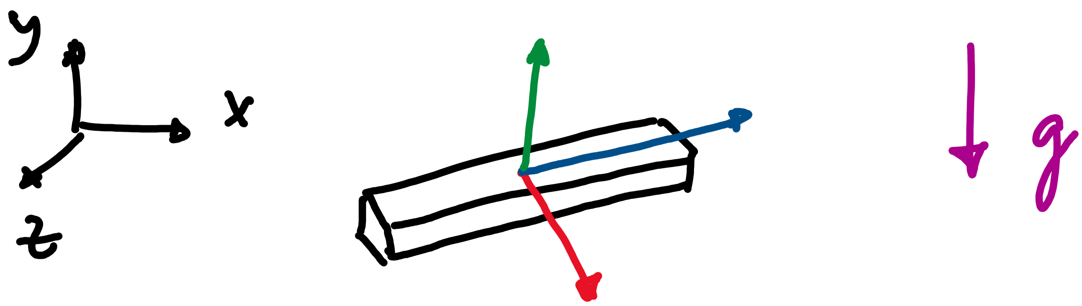
Dynamics

① Free Floating Body (free joint)



Given an initial position & orientation / initial linear & angular velocity, describe the movement of the object

Equations of motion:

(i) Translation :

linear acceleration 3 eqn

$$m\dot{v}_x = 0$$
$$m\dot{v}_y = 0$$
$$m\dot{v}_z = -mg$$

velocity 3 eqn

$$\dot{x} = v_x$$
$$\dot{y} = v_y$$
$$\dot{z} = v_z$$

Written in global frame

(ii) Rotation:

$$I_b \dot{\omega}_b + \omega_b \times (I_b \omega_b) = M_b$$

ω_b (Ω) - body from angular velocity

I_b - body frame inertia

M_b - Moment in the body frame

$$\dot{\omega}_b = (I_b)^{-1} \left[M_b - \omega_b \times (I \omega_b) \right]$$

angular acceleration 3 equations

2 methods to compute orientation

(a) Euler angles/rate [1-2-3)

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \frac{1}{\cos\theta} \begin{bmatrix} \cos\psi \cos\theta & \sin\psi & 0 \\ -\sin\psi \sin\theta & \cos\psi & 0 \\ \sin\theta & 0 & 1 \end{bmatrix} \omega_b$$

angular velocity

3 equations

(b) Quaternions

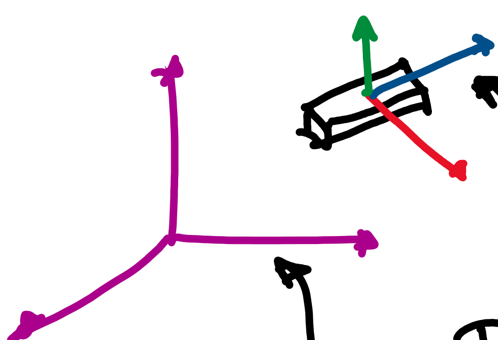
$\omega_b = 2\bar{q} \circ \dot{q} \Rightarrow \dot{q} = 0.5\, q \circ \omega_b$

angular velocity

3 equations

I - world frame inertia

I_b - body frame inertia



I_b is in this frame

I is in this frame

Relation between I & I_b

Rotational energy $= 0.5\,\omega^T(I\omega)$

$= 0.5\,\omega_b^T(I_b\omega_b)$

$$\omega^T(I\omega) = \underline{\omega_b}^T(I_b\,\omega_b)$$

But $\omega = R_1^0\,\omega_b = R\,\omega_b$

$\omega_b = R^T\omega$

$$\omega^T(I\omega) = (R^T\omega)^T\,(I_b R^T\omega)$$

$$\omega^T(I\omega) = \omega^T R(I_b R^T\omega)$$

$$\omega^T(I\omega) = \omega^T R I_b R^T\omega$$

$$\boxed{I = R I_b R^T}$$

"table wiki moment of inertia"

Dynamics

2 ways of deriving the equations of motion

① Euler-Lagrange method

- No Free Body Diagram
  - ~ does not give internal forces
- Require Symbolic computation
  - ~ long equations for long chains

② Newton-Euler method

- Requires Free Body Diagram
  - ~ gives interaction forces
- Symbolic OR Numeric computations
  - symbolic small chains
  - numeric for long chains

MuJoCo.

Recursive Newton Euler Algorithm (RNEA)

(I) Euler-lagrange method

(1) Write Formula for the position and velocity of the center of mass with respect to the world frame

$$\underset{\text{global frame position}}{p_c^0} = H_i^0 \underset{\text{local frame position}}{p_c^i}$$

(i) $\underline{v_c^0} = \frac{d}{dt}(p_c^0)$

$w_c^0 = \dot{R}_c^0 (R_c^0)^T$

(ii) $v_c^0 = J_{v_i} \dot{q}_i$

$w_c^0 = J_{w_i} \dot{q}_i$

(2) $\mathcal{L}$ - lagrangian

$$\mathcal{L} = T - V$$

$$T = \frac{1}{2} \Sigma \left[m_i \underset{1\times3}{v_i^T} \underset{3\times1}{v_i} + \underset{1\times3}{w_i^T} \underbrace{(\overset{3\times3}{I_i} \overset{3\times1}{w_i})}_{3\times1} \right]$$

T = kinetic energy

Using the Jacobian we can write

$$T = \frac{1}{2}\dot{q}^T \sum_{i=1}^{n}\left[m_i (J_{v_i})^T J_{v_i} + J_{w_i}^T \underbrace{(R_i I_b R_i^T)}_{I \text{ (world frame inertia)}} J_{w_i}\right]\dot{q}$$

V - potential energy

$$V = \sum_{i=1}^{n} m_i\, g\, z_G^0 + \frac{1}{2}\sum_{i=1}^{m} K_{p_i}(r_{p_i} - r_{p_0})^2$$

K_{p_i} - spring constant

r_{p_0}, r_{p_i} - spring length in relaxed state
spring length when stretched

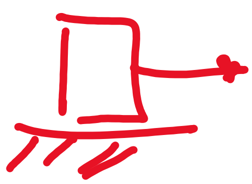
g - gravity (assumed to be along z direction)

③ Equations of motion

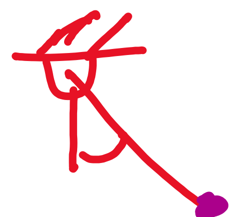
$$\boxed{\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_j}\right) - \frac{\partial \mathcal{L}}{\partial q_j} = Q_j}$$

q_j — degree of freedom (e.g. revolute, prismatic)

Q_j — external force / torque (damping friction)



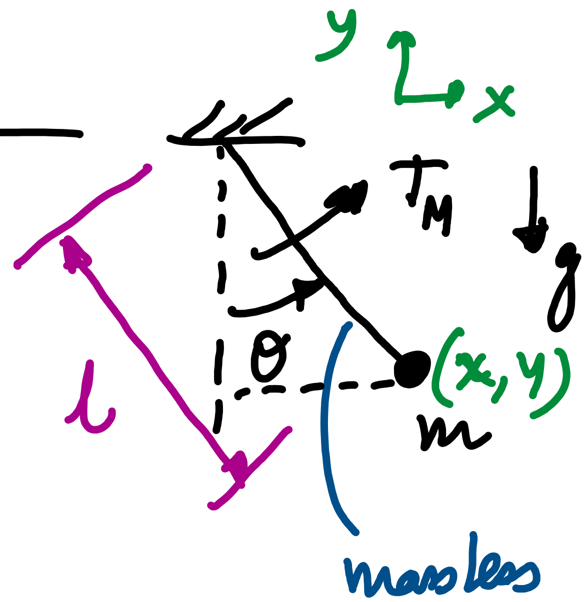
$q_j = x$



$q_j = \theta$

EXAMPLE

Derive the equation of motion for a simple pendulum subject to external torque T_M



①

$$x = l\sin\theta$$

$$y = -l\cos\theta$$

$$\dot{x} = l\cos\theta\,\dot{\theta}$$

$$\dot{y} = +l\sin\theta\,\dot{\theta}$$

$$v = \begin{bmatrix}\dot{x}\\ \dot{y}\end{bmatrix} = \begin{bmatrix} l\cos\theta\,\dot{\theta}\\ l\sin\theta\,\dot{\theta}\end{bmatrix}$$

② $\mathcal{L} = T - V$

$$= \frac{1}{2} m v^T v - mg\,y$$

$$= \frac{1}{2} m \left[l\cos\theta\,\dot{\theta},\ l\sin\theta\,\dot{\theta}\right]\begin{bmatrix} l\cos\theta\,\dot{\theta}\\ l\sin\theta\,\dot{\theta}\end{bmatrix} + mgl\cos\theta$$

$$\mathcal{L} = \frac{1}{2} m l^2 \dot{\theta}^2 + mgl\cos\theta$$

$$\mathcal{L} = \frac{1}{2} m l^2 \dot{\theta}^2 + mg\, l \cos\theta$$

③ $$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}}\right) - \frac{\partial \mathcal{L}}{\partial \theta} = T_m$$

$$\frac{d}{dt}\left[\frac{1}{\cancel{2}} m l^2 (\cancel{2}\dot{\theta})\right] - mgl(-\sin\theta) = T_m$$

$$\boxed{m l^2 \ddot{\theta} + mgl \sin\theta = T_m}$$

check: $T_m = 0$

$$m l^2 \ddot{\theta} + mgl \sin\theta = 0$$

$$\ddot{\theta} = -\frac{g}{l} \sin\theta$$

eq^n of a simple pendulum

General form of Equation of motion for a manipulator

$$\rightarrow \underline{\underline{\frac{d}{dt}}} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i} \right) - \frac{\partial \mathcal{L}}{\partial q_j} = Q_j$$

↓ Spong Chapter 6

torque

$$\rightarrow \boxed{\underbrace{\underline{M}(\underline{q})}_{n\times n} \underbrace{\underline{\ddot{q}}}_{n\times 1} + \underbrace{C(\underline{q}, \underline{\dot{q}})}_{n\times 1} + \underbrace{G(q, g)}_{n\times 1} = U}$$

g.

. $M(q)$ mass matrix,
depends only on q

$C(q, \dot{q})$ non-linear coriolis / centripetal acceleration term
depends on $q, \dot{q}$

$G(q, g)$ gravity term, it depends on gravity, q

$$x = \underbrace{l\sin\theta}$$

$$\dot{x} = l\cos\theta\,\dot{\theta}$$

another way

$$\dot{x} = \underbrace{\left(\frac{\partial x}{\partial \theta}\right)}_{\text{jacobian}}\dot{\theta} \rightsquigarrow \text{chain rule}$$

$$= (l\cos\theta)\,\dot{\theta}$$

$$EOM = \begin{bmatrix} EOM0 \\ EOM1 \\ EOM2 \end{bmatrix} \Big\} \quad M, C, G = ?$$

① $M = \dfrac{\partial EOM}{\partial \ddot{\theta}}$ $\qquad \ddot{\theta} = [\ddot{\theta}_1, \ddot{\theta}_2, \ddot{\theta}_3]$

② $C = -EOM(\ddot{q} = 0, \dot{q}, q, g = 0)$

③ $G = -EOM(\ddot{q} = 0, \dot{q} = 0, q, g)$

$$M\ddot{q} + C + G = 0$$

$$\ddot{q} = -M^{-1}(C+G)$$

Simulate : integration

✓ $\dot{q} = -\int \underbrace{M^{-1}(C+G)}_{\text{angular acceleration}} dt = \omega$

$q = \int \omega \, dt$

Once we obtain q we can do an animation $[\theta_1, \theta_2, \theta_3]$

odeint (runge-kutta adaptive)

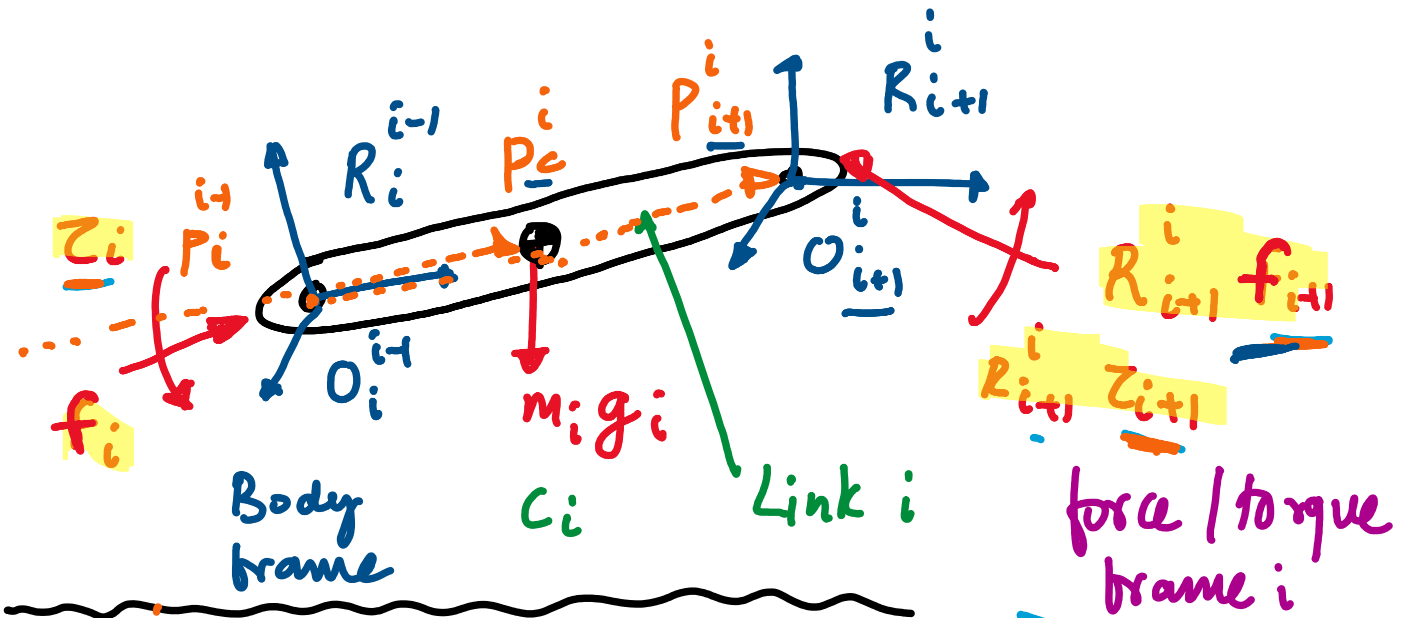
① Euler's method

② Runge kutta fixed, order 4

MuJoCo

more accurate

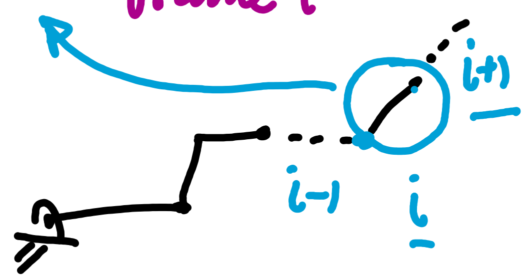
II Newton-Euler Method
(Recursive Newton-Euler Algorithm)



$$F_i = m_i \dot{v}_{c_i}$$

$$N_i = I_{b_i} \dot{\omega}_i + \omega_i \times (I_b \omega_i)$$

ω_i^i

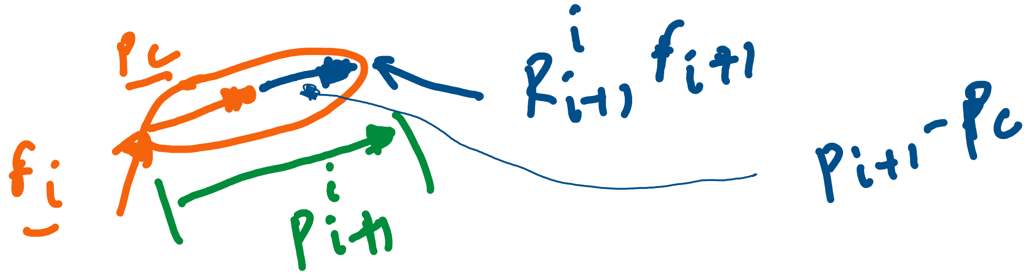


Force Balance

$$f_i = F_i + R_{i+1}^i f_{i+1} - m_i g_i$$

Torque Balance about point C_i

✓ $$z_i = N_i + R_{i+1}^i z_{i+1} + p_c^i \times f_i + (p_{i+1}^i - p_c^i) \times R_{i+1}^i f_{i+1}$$



Velocities and Accelerations

Joint i is revolute

$$\omega_i = (R_i^{i-1})^T \omega_{i-1} + \dot{q}_i \hat{n}_i$$ ← same as velocity.

$$\dot{\omega}_i = (R_i^{i-1})^T \dot{\omega}_{i-1} + \omega_i \times \dot{q}_i \hat{n}_i + \ddot{q}_i \hat{n}_i$$

$$\dot{v}_i = (R_i^{i-1})^T \Big[\underbrace{\dot{v}_{i-1} + \dot{\omega}_{i-1} \times p_i^{i-1} + \omega_{i-1} \times (\omega_{i-1} \times p_i^{i-1})}_{\ddot{o}_i^{i-1}} \Big]$$

$$\dot{v}_{c_i} = \dot{v}_i + \dot{\omega}_i \times p_c^i + \omega_i \times (\omega_i \times p_c^i)$$

Joint i is prismatic

$$v = v_0 + \omega \times (\omega \times r) + \dot{\omega} \times r$$

$$\omega_i = (R_i^{i-1})^T \omega_{i-1}$$

$$\dot{\omega}_i = (R_i^{i-1})^T \dot{\omega}_{i-1}$$

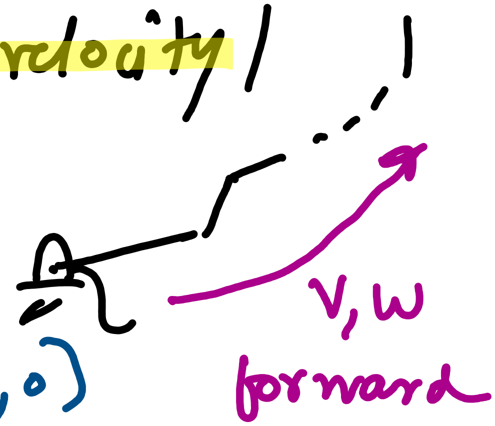
$\ddot{o}_i^{i-1}$

$$\dot{v}_i = (R_i^{i-1})^T \Big[\dot{v}_{i-1} + \dot{\omega}_{i-1} \times p_i^{i-1} + \omega_{i-1} \times (\omega_{i-1} \times p_i^{i-1}) + \ddot{q}_{i-1} \hat{n}_{i-1} + 2\omega_{i-1} \times \dot{q}_{i-1} \hat{n}_{i-1} \Big]$$

$$\dot{v}_{c_i} = \dot{v}_i + \dot{\omega}_i \times p_i^{i-1} + \omega_{i-1} \times (\omega_{i-1} \times p_i^{i-1}) + \ddot{q}_i \hat{n}_i + 2\omega_i \times \dot{q}_i \hat{n}_i$$

How to use these equations?

① Forward recursion for velocity / acceleration



$v_0 = [0,0,0)$ $\quad \dot{v}_0 = [0,0,0)$

$\omega_0 = (0,0,0)$ $\quad \dot{v}_0 = [0,0,0)$

Use recursion to compute

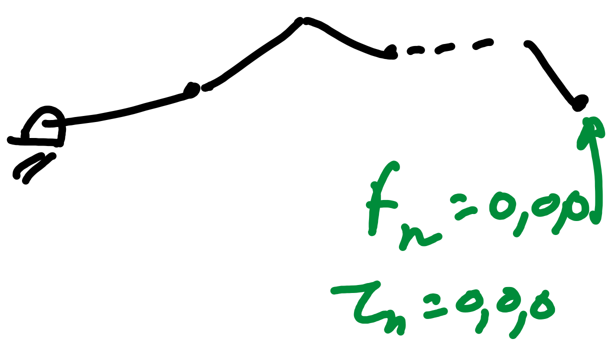
$v_1, \dot{v}_1, \omega_1, \dot{\omega}_1 \ldots$

$v_i, w_i, \dot{v}_i, \dot{\omega}_i = f(v_{i-1}, w_{i-1}, \dot{v}_{i-1}, \dot{w}_{i-1})$

$i = 1$

② Backward recursion for forces / torques

$f_n = [0,0,0)$
$\tau_n = [0,0,0)$

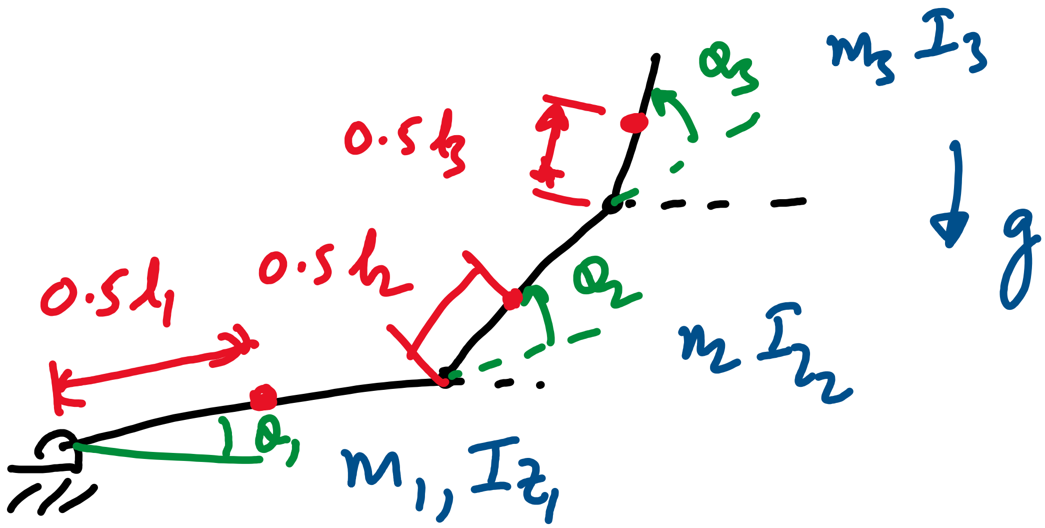


$f_i, \tau_i = \bar{F}(f_{i+1}, \tau_{i+1})$

$i+1 = n$

EOM: $\tau_1(\hat{n}_1), \tau_2(\hat{n}_2), \ldots$ $\quad \hat{n}_1$ = axis of motion for q_1

Derive the equations of motion of a 3-link planar manipulator



Compute M numerically

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

numerically

$$\begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} + C(q,\dot{q}) + G(q,g) = \underline{\tau}$$

$\rightarrow$ $\dot{q} = 0$, $g = 0$ $\Rightarrow$ $C = 0$, $G = 0$

Set $\ddot{q}_1 = 1$; $\ddot{q}_2 = \ddot{q}_3 = 0$

$$\tau = \begin{bmatrix} M_{11} \\ M_{21} \\ M_{31} \end{bmatrix}$$

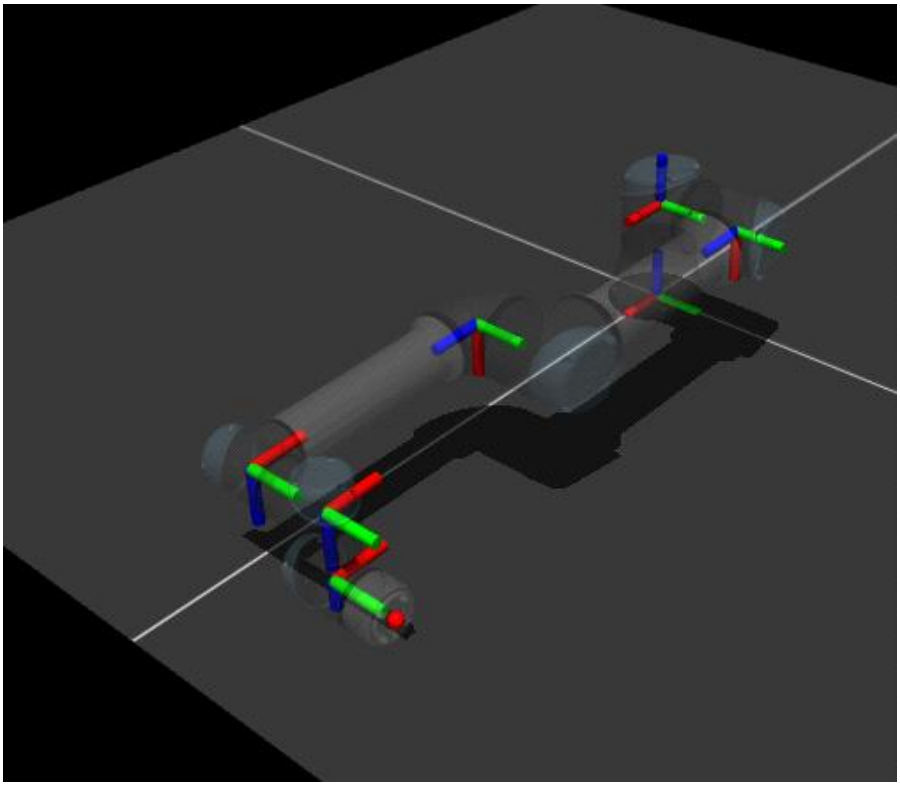
Set $\ddot{q}_2 = 1$; $\ddot{q}_1 = \ddot{q}_3 = 0$

$$\tau = \begin{bmatrix} M_{12} \\ M_{22} \\ M_{32} \end{bmatrix}$$

Set $\ddot{q}_3 = 1$; $\ddot{q}_1 = \ddot{q}_2 = 0$ $\quad \tau = \begin{bmatrix} M_{13} \\ M_{23} \\ M_{33} \end{bmatrix}$

3 calls to forward backward recursion

Recursive Newton Euler Algorithm for UR5



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

$q \in \mathbb{R}^6 \quad C, G \in \mathbb{R}^6 \quad M \in \mathbb{R}^{6\times 6}$

① Euler-Lagrange ~ symbolic (lengthy)

② Newton-Euler ~ " (")

③ Recursive Newton-Euler Algorithm (RNEA) ~ numerical

Use $J^T M J$ method to compute M

$$\underbrace{M = J^T \bar{M}}\, J$$

$$\underset{6\times 6}{M} = \underset{6\times 36}{J^T}\ \underset{36\times 36}{\bar{M}}\ \underset{36\times 6}{J}$$

J - jacobian of the center of mass of each link concatenated such that $J \in \mathbb{R}^{36\times 6}$

$$\underset{36\times 36}{\bar{M}} = \begin{bmatrix} M_{1\,(3\times 3)} & 0_{(3\times 3)} & 0 & 0 & & & 0 \\ 0 & I_{1\,(3\times 3)} & 0 & 0 & & & 0 \\ 0 & 0 & M_2 & 0 & & & \\ 0 & 0 & 0 & I_2 & & & \\ & & & & \ddots & & \\ & & & & & M_6 & 0 \\ & & & & & 0 & I_6 \end{bmatrix}$$

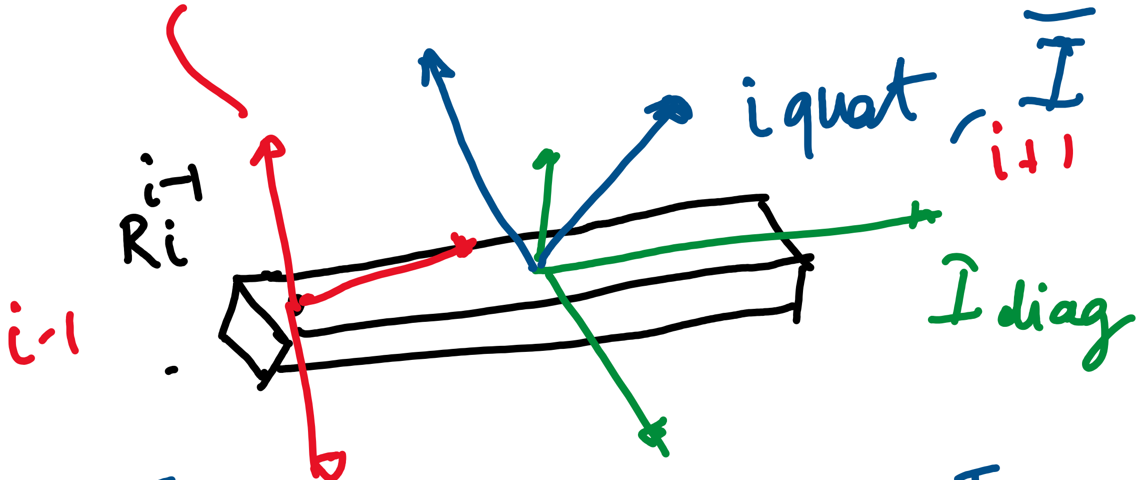
$$M_i = \begin{pmatrix} m_i & 0 & 0 \\ 0 & m_i & 0 \\ 0 & 0 & m_i \end{pmatrix}$$

$$I_i = R_i^{i-1} R_{inertia} I_{diag} R_{inertia}^T R_i^{i-1}$$

$R_{inertia}$ = iquat2Rotation

<inertial quat= diag: [_ _ _]]

iquat Idiag



$$\bar{I} = R_{inertia} \, I_{diag} \, R^T_{inertia}$$

$$I_i = R_i^{i-1} \, \bar{I} \, (R_{i-1}^{i})^T$$

$R_{inertia}$ = ram. quat2rotation(iquat)