$\bar\Theta, \dot{\bar\Theta}, \ddot{\bar\Theta} \longleftrightarrow \bar\tau \leftarrow$ Forces + Torques on each joint

For control + Simulation

Lagrange-Euler equations for robot dynamics $\leftarrow$ Energy balance

$$\frac{d}{dt}\left[\frac{\partial L}{\partial \dot q_i}\right] - \frac{\partial L}{\partial q_i} = \tau_i \qquad i = 1, 2, \dots n$$

Trace

L = Lagrangian function = Kinetic energy k − Potential energy P

K = Total kenetic energy of robot $= \sum_i \int dk = \frac{1}{2}\sum_{i=1}^{n}\sum_{p=1}^{i}\sum_{r=1}^{i}\left[Tr\left(\bar U_{ip}\bar J_i \bar U_{ir}^T\right)\dot q_p \dot q_r\right]$

P = Total Potential energy of robot $= \sum_i -m_i \bar g\, {}^0\bar r_i = \sum_i -m_i \bar g \left({}^0\bar A_i\, {}^i\bar r_i\right)$

q_i = generalized coordinates (joint or cartesian)

$\dot q_i$ = first time derivative

τ_i = Generalized force (or Torque) applied at joint i to drive link i

$U_{ij} = {}^0\bar A_{j-1} \cdot \overbrace{\frac{\partial}{\partial q_j}\, {}^{j-1}\bar A_j}^{Q_j \text{ (see below)}}\, {}^j\bar A_i \qquad j \le i \qquad \text{or } = 0,\ j > i \qquad = \frac{\partial\, {}^0\bar A_i}{\partial q_j}$

$\bar J_i$ = 4 x 4 matrix (computed once for the robot)

m_i = mass of link i

$\bar g_i$ = gravity row vector $= (g_x, g_y, g_z, 0)$ $= (0, 0, -|g|, 0)$ for a level system

${}^0\bar r_i$ = Position of link i with respect to frame 0

${}^0\bar A_i$ = position of frame i with respect to frame 0

${}^i\bar r_i$ = position of center of mass of link i with respect to frame i

$\boxed{\text{or}}$ $\bar\tau(t) = \bar D(\bar q(t))\,\ddot{\bar q}(t) + \bar h\left(\bar q(t), \dot{\bar q}(t) + \bar c(q(t))\right)$ = torque vector $(\tau_1 \tau_2 \cdots \tau_n)^T$

$\bar D(\bar q(t))$ = Inertial acceleration matrix $(n \times n)$ $\qquad D_{ik} = \sum_{j=\max(i,k)}^{n} Tr\left(\bar U_{jk}\bar J_j \bar U_{ji}\right)$

$\ddot{\bar q}(t)$ = second time derivative of $\bar q$

$\bar h(q, \dot q)$ = Coriolis and centrifugal force vector $\qquad h_i = \sum_{k=1}^{n}\sum_{m=1}^{n} h_{ikm}\,\dot q_k \dot q_m$

$$h_{ikm} = \sum_{j=\max(i,k,m)}^{n} Tr\left(\bar U_{jkm}\bar J_j \bar U_{ji}^T\right)$$

$$\bar U_{ijk} = \begin{cases} {}^0\bar A_{j-1}\,\bar Q_j\,{}^{j-1}\bar A_{k-1}\,\bar Q_k\,{}^{k-1}\bar A_i & i \ge k \ge j \\ {}^0\bar A_{k-1}\,\bar Q_k\,{}^{k-1}\bar A_{j-1}\,\bar Q_j\,{}^{j-1}\bar A_i & i \ge j \ge k \\ 0 & i < j \text{ or } i < k \end{cases}$$

$$\bar Q_i = \underset{\text{revolute joint}}{\begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}} \text{ or } \underset{\text{prismatic joint}}{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}}$$

$\bar c$ = gravity loading vector $\Rightarrow c_i = \sum_{j=i}^{n}\left(-m_j\,\bar g\,\bar U_{ji}\,{}^j\bar r_j\right)$

$$j \le i$$

$$\bar{U}_{ij} = \frac{\partial\, {}^0\bar{A}_i}{\partial q_j} = \frac{\partial}{\partial q_j}\, {}^0\bar{A}_1\, {}^1\bar{A}_2\, {}^2\bar{A}_3 \cdots {}^{i-1}\bar{A}_i$$

$$= \frac{\partial}{\partial q_j}\left({}^0\bar{A}_{j-1}\, {}^{j-1}\bar{A}_j\, {}^j\bar{A}_i \right)$$

$$= {}^0\bar{A}_{j-1}\, \underbrace{\frac{\partial}{\partial q_j}\, {}^{j-1}\bar{A}_j}\, {}^jA_i$$

$${}^{j-1}\bar{A}_j = \begin{pmatrix} \cos\theta_j & -\cos\alpha_j \sin\theta_j & \sin\alpha_j \sin\theta_j & a_j \cos\theta_j \\ \sin\theta_j & \cos\alpha_j \cos\theta_j & -\sin\alpha_j \cos\theta_j & a_j \sin\theta_j \\ 0 & \sin\alpha_j & \cos\alpha_j & d_j \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Rotating joint

$q_j = \theta_j$

$$\frac{\partial\, {}^{j-1}\bar{A}_j}{\partial q_j} = \begin{pmatrix} -\sin\theta_j & -\cos\alpha_j \cos\theta_j & \sin\alpha_j \cos\theta_j & -a_j \sin\theta_j \\ \cos\theta_j & -\cos\alpha_j \sin\theta_j & \sin\alpha_j \sin\theta_j & a_j \cos\theta_j \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$\bar{Q}_j$

$$= \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} {}^{j-1}\bar{A}_j = \bar{Q}_j\, {}^{j-1}A_j$$

$$\frac{\partial\, {}^0A_i}{\partial q_j} = {}^0\bar{A}_{j-1}\, Q_j\, {}^{j-1}\bar{A}_j\, {}^jA_i$$

$$= {}^0\bar{A}_{j-1}\, \bar{Q}_j\, {}^{j-1}\bar{A}_i$$

$$ {}^{j-1}\bar{A}_j = \begin{pmatrix} \cos\theta_j & -\cos\alpha_j \sin\theta_j & \sin\alpha_j \sin\theta_j & a_j \cos\theta_j \\ \sin\theta_j & \cos\alpha_j \cos\theta_j & -\sin\alpha_j \cos\theta_j & a_j \sin\theta_j \\ 0 & \sin\alpha_j & \cos\alpha_j & d_j \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

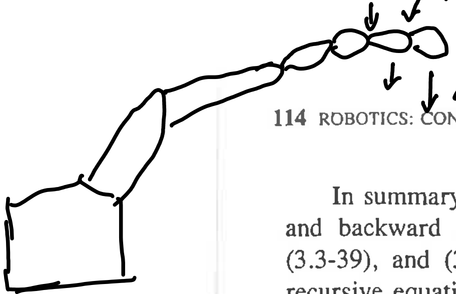
sliding joint

$$ \frac{\partial\, {}^{j-1}\bar{A}_j}{\partial q_j} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \leftarrow Q_j $$

$$ = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} {}^{j-1}A_j $$

Newton-Euler - force balance

many fewer Calculations



① work from base to ee & solve for position, velocity, acceleration

② work back from ee to base & solve for forces & Torques

In summary, the Newton-Euler equations of motion consist of a set of forward and backward recursive equations. They are Eqs. (3.3-28), (3.3-29), (3.3-35), (3 (3.3-39), and (3.3-43) to (3.3-45) and are listed in Table 3.2. For the forward recursive equations, linear velocity and acceleration, angular velocity and acceleration of each individual link, are propagated from the base reference system to the end-effector. For the backward recursive equations, the torques and forces exerted on each link are computed recursively from the end-effector to the base reference system. Hence, the forward equations propagate kinematics information of each link from the base reference frame to the hand, while the backward equations compute the necessary torques/forces for each joint from the hand to the base reference system.

Table 3.2 Recursive Newton-Euler equations of motion

Forward equations: $i = 1, 2, \ldots, n$

$v_i + \omega_i$ = linear & angular velocities of frame i wrt. base. centers of mass

a = linear acceleration

$$\bar{\omega}_i = \begin{cases} \bar{\omega}_{i-1} + \mathbf{z}_{i-1}\dot{q}_i & \text{if link } i \text{ is rotational} \\ \bar{\omega}_{i-1} & \text{if link } i \text{ is translational} \end{cases}$$

$$\dot{\omega}_i = \begin{cases} \dot{\omega}_{i-1} + \mathbf{z}_{i-1}\ddot{q}_i + \omega_{i-1} \times (\mathbf{z}_{i-1}\dot{q}_i) & \text{if link } i \text{ is rotational} \\ \dot{\omega}_{i-1} & \text{if link } i \text{ is translational} \end{cases}$$

$v_i =$

$$\dot{\mathbf{v}}_i = \begin{cases} \dot{\omega}_i \times \mathbf{p}_i^* + \omega_i \times (\omega_i \times \mathbf{p}_i^*) + \dot{\mathbf{v}}_{i-1} & \text{if link } i \text{ is rotational} \\ \mathbf{z}_{i-1}\ddot{q}_i + \dot{\omega}_i \times \mathbf{p}_i^* + 2\omega_i \times (\mathbf{z}_{i-1}\dot{q}_i) + \omega_i \times (\omega_i \times \mathbf{p}_i^*) + \dot{\mathbf{v}}_{i-1} & \text{if link } i \text{ is translational} \end{cases}$$

$$\bar{\mathbf{a}}_i = \dot{\omega}_i \times \bar{\mathbf{s}}_i + \omega_i \times (\omega_i \times \bar{\mathbf{s}}_i) + \dot{\mathbf{v}}_i$$

3.3-32

$$\bar{v}_i = \begin{cases} \bar{\omega}_i \times \bar{p}_i^* + \bar{v}_{i-1} & \text{rotational} \\ z_{i-1}\dot{q}_i + \bar{\omega}_i \times \bar{p}^* + \bar{v}_{i-1} & \text{translational} \end{cases}$$

Backward equations: $i = n, n-1, \ldots, 1$

$$\mathbf{F}_i = m_i \bar{\mathbf{a}}_i$$

$$\mathbf{N}_i = \mathbf{I}_i \dot{\omega}_i + \omega_i \times (\mathbf{I}_i \omega_i)$$

$$\mathbf{f}_i = \mathbf{F}_i + \mathbf{f}_{i+1}$$

$$\mathbf{n}_i = \mathbf{n}_{i+1} + \mathbf{p}_i^* \times \mathbf{f}_{i+1} + (\mathbf{p}_i^* + \bar{\mathbf{s}}_i) \times \mathbf{F}_i + \mathbf{N}_i$$

$$\tau_i = \begin{cases} \mathbf{n}_i^T \mathbf{z}_{i-1} + b_i \dot{q}_i & \text{if link } i \text{ is rotational} \\ \mathbf{f}_i^T \mathbf{z}_{i-1} + b_i \dot{q}_i & \text{if link } i \text{ is translational} \end{cases}$$

where b_i is the viscous damping coefficient for joint i.

The "usual" initial conditions are $\omega_0 = \dot{\omega}_0 = \mathbf{v}_0 = \mathbf{0}$ and $\dot{\mathbf{v}}_0 = (g_x, g_y, g_z)^T$ (to include gravity), where $|\mathbf{g}| = 9.8062 \text{ m/s}^2$.

# calculation			
Lagrange-Euler	101,348 *	77 405 +	
Newton-Euler	792 *	662 +	