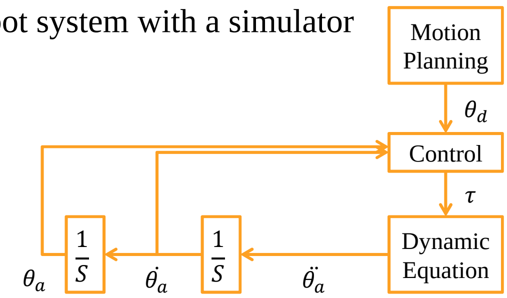
1

Chapter 6 – Dynamics

- 6.1 Lagragian Mechanics
- 6.2 Newton – Euler Dynamic Formulation
- 6.3 Recursive Newton-Euler Dynamic Algorithm
- 6.4 Dynamic Simulator

6.1 Lagrangian Mechanics

- The study considers the forces required to cause motion:
  1. Given $\theta, \dot{\theta}$ and $\ddot{\theta}$ find the required joint torques.
  2. Given joint torque τ, compute the resulting motion $\theta, \dot{\theta}$ and $\ddot{\theta}$.
- A robot system with a simulator



- Two famous methods for computing dynamics
  1. Lagrangian formulation
  2. Newton-Euler formulation

6.1 Lagrangian Mechanics

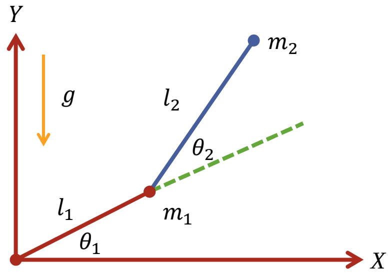
- A two-link manipulator example
- Lagrangian: defined as the difference between the kinematic and potential energy of a mechanical system.

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

Kinematic energy — Potential energy

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

$$\Rightarrow \frac{d}{dt}\frac{\partial K}{\partial \dot{\theta}} - \frac{\partial K}{\partial \theta} + \frac{\partial P}{\partial \theta} = \tau$$



Ref: H. Goldstein, “Classical Mechanics”, Addition-Wesley, Chapter 1& 2

6.1 Lagrangian Mechanics

For link 1: $\begin{cases} K_1 = \frac{1}{2} m_1 l_1^2 \dot{\theta}_1^2 \\ P_1 = m_1 g l_1 \sin\theta_1 \end{cases}$

For link 2: $\begin{cases} x_2 = l_1 \cos\theta_1 + l_2 \cos(\theta_1+\theta_2) \\ y_2 = l_1 \sin\theta_1 + l_2 \sin(\theta_1+\theta_2) \end{cases}$

$$\begin{cases} \dot{x}_2 = -l_1(\sin\theta_1)\dot{\theta}_1 - l_2[\sin(\theta_1+\theta_2)](\dot{\theta}_1+\dot{\theta}_2) \\ \dot{y}_2 = l_1(\cos\theta_1)\dot{\theta}_1 + l_2[\cos(\theta_1+\theta_2)](\dot{\theta}_1+\dot{\theta}_2) \end{cases}$$

$$\Rightarrow \dot{x}_2^2 + \dot{y}_2^2 = V_2^2$$

$$V_2^2 = l_1^2\dot{\theta}_1^2 + l_2^2(\dot{\theta}_1^2 + 2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2) + 2l_1l_2(\cos\theta_2)(\dot{\theta}_1^2 + \dot{\theta}_1\dot{\theta}_2)$$

$$\begin{cases} K_2 = \frac{1}{2} m_2 V_2^2 \\ P_2 = m_2 g l_1 \sin\theta_1 + m_2 g l_2 \sin(\theta_1+\theta_2) \end{cases}$$

6.1 Lagrangian Mechanics

5 NCTU - Human and Machine Lab

Lagrangian, L=K-P

$$=\frac{1}{2}(m_1+m_2)l_1^2\dot{\theta}_1^2+\frac{1}{2}m_2l_2^2(\dot{\theta}_1^2+2\dot{\theta}_1\dot{\theta}_2+\dot{\theta}_2^2)+m_2l_1l_2(\cos\theta_2)(\dot{\theta}_1^2+\dot{\theta}_1\dot{\theta}_2)$$
$$-(m_1+m_2)gl_1\sin\theta_1-m_2gl_2\sin(\theta_1+\theta_2)$$

Dynamic Equations

$$\frac{\partial L}{\partial\dot{\theta}_1}=(m_1+m_2)l_1^2\dot{\theta}_1+m_2l_2^2\dot{\theta}_1+m_2l_2^2\dot{\theta}_2$$
$$+2m_2l_1l_2(\cos\theta_2)\dot{\theta}_1+m_2l_1l_2(\cos\theta_2)\dot{\theta}_2$$

$$\frac{d}{dt}\frac{\partial L}{\partial\dot{\theta}_1}=[(m_1+m_2)l_1^2+m_2l_2^2+2m_2l_1l_2\cos\theta_2]\ddot{\theta}_1$$
$$+(m_2l_2^2+m_2l_1l_2\cos\theta_2)\ddot{\theta}_2$$
$$-2m_2l_1l_2\sin\theta_2\dot{\theta}_1\dot{\theta}_2-m_2l_1l_2\sin\theta_2\dot{\theta}_2^2$$

6.1 Lagrangian Mechanics

6 NCTU - Human and Machine Lab

$$\frac{\partial L}{\partial\theta_1}=-(m_1+m_2)gl_1\cos\theta_1-m_2gl_2\cos(\theta_1+\theta_2)$$

$$\tau_1=[(m_1+m_2)l_1^2+m_2l_2^2+2m_2l_1l_2\cos\theta_2]\ddot{\theta}_1$$
$$+(m_2l_2^2+m_2l_1l_2\cos\theta_2)\ddot{\theta}_2-2m_2l_1l_2\sin\theta_2\dot{\theta}_1\dot{\theta}_2$$
$$-m_2l_1l_2\sin\theta_2\dot{\theta}_2^2+(m_1+m_2)gl_1\cos\theta_1$$
$$+m_2gl_2\cos(\theta_1+\theta_2)$$

6.1 Lagrangian Mechanics

7 NCTU - Human and Machine lab

$$\frac{\partial L}{\partial \dot{\theta}_2} = m_2 l_2^2 \dot{\theta}_1 + m_2 l_2^2 \dot{\theta}_2 + m_2 l_1 l_2 (\cos\theta_2)\dot{\theta}_1$$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}_2} = m_2 l_2^2 \ddot{\theta}_1 + m_2 l_2^2 \ddot{\theta}_2 + m_2 l_1 l_2 (\cos\theta_2)\ddot{\theta}_1 - m_2 l_1 l_2 (\sin\theta_2)\dot{\theta}_1\dot{\theta}_2$$

$$\frac{\partial L}{\partial \theta_2} = -m_2 l_1 l_2 (\sin\theta_2)(\dot{\theta}_1^2 + \dot{\theta}_1\dot{\theta}_2) - m_2 g l_2 \cos(\theta_1 + \theta_2)$$

$$\tau_2 = (m_2 l_2^2 + m_2 l_1 l_2 \cos\theta_2)\ddot{\theta}_1 + m_2 l_2^2 \ddot{\theta}_2$$
$$+ m_2 l_1 l_2 (\sin\theta_2)\dot{\theta}_1^2 + m_2 l_2 g \cos(\theta_1 + \theta_2)$$

6.1 Lagrangian Mechanics

8 NCTU - Human and Machine lab

By rearrangement:

$$\tau_1 = D_{11}\ddot{\theta}_1 + D_{12}\ddot{\theta}_2 + D_{111}\dot{\theta}_1^2 + D_{122}\dot{\theta}_2^2 + D_{112}\dot{\theta}_1\dot{\theta}_2 + D_{121}\dot{\theta}_2\dot{\theta}_1 + D_1$$
$$\tau_2 = D_{21}\ddot{\theta}_1 + D_{22}\ddot{\theta}_2 + D_{211}\dot{\theta}_1^2 + D_{222}\dot{\theta}_2^2 + D_{212}\dot{\theta}_1\dot{\theta}_2 + D_{221}\dot{\theta}_2\dot{\theta}_1 + D_2$$

- D_{ii}: effective inertia at joint i ($D_{ii}\ddot{\theta}_i$)
- D_{ij}: coupling inertia between joints i and j $\left(D_{ij}\ddot{\theta}_i \text{ or } D_{ij}\ddot{\theta}_j\right)$
- $D_{ijj}\dot{\theta}^2{}_j$: centripetal force acting at joint i due to a velocity at joint j
- $D_{ijk}\dot{\theta}_j\dot{\theta}_k + D_{ijk}\dot{\theta}_k\dot{\theta}_j$: Coriolis force acting at joint i due to a velocity at joint j and k
- D_i: gravity force at joint i

6.1 Lagrangian Mechanics

Effective inertia

$$D_{11} = (m_1 + m_2)l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 \cos\theta_2$$

$$D_{22} = m_2 l_2^2$$

Coupling inertia

$$D_{21} = D_{12} = m_2 l_2^2 + m_2 l_1 l_2 \cos\theta_2$$

Centripetal acceleration coefficient

$$D_{111} = D_{222} = 0, D_{122} = -m_2 l_1 l_2 \sin\theta_2, D_{211} = m_2 l_1 l_2 \sin\theta_2$$

Coriolis acceleration coefficient

$$D_{112} = D_{121} = -m_2 l_1 l_2 \sin\theta_2$$

$$D_{212} = D_{221} = 0$$

Gravity

$$D_1 = (m_1 + m_2) g l_1 \cos\theta_1 + m_2 g l_2 \cos(\theta_1 + \theta_2)$$

$$D_2 = m_2 g l_2 \cos(\theta_1 + \theta_2)$$

6.1 Lagrangian Mechanics

- Example:
  - Evaluate the dynamic equations under different conditions (assume there is no gravity and it is at rest)

① Joint 2 locked ($\ddot{\theta}_2 = 0$)

$$\tau_1 = D_{11}\ddot{\theta}_1$$

$$\tau_2 = D_{21}\ddot{\theta}_1$$

② Joint 2 free ($\tau_2 = 0$)

$$\tau_2 = 0 = D_{21}\ddot{\theta}_1 + D_{22}\ddot{\theta}_2 \longrightarrow \ddot{\theta}_2 = -\frac{D_{12}}{D_{22}}\ddot{\theta}_1$$

$$\tau_1 = [D_{11} - \frac{D_{12}^2}{D_{22}}]\ddot{\theta}_1$$

  - Comparison
    - Let $m_1 = 1, m_2 = 1, l_1 = 1, l_2 = 1$

θ_2	D_{11}	D_{21}	D_{22}	① I_1	② I_1
0°	5	2	1	5	1
90°	3	1	1	3	2
180°	1	0	1	1	1
270°	3	1	1	3	2

6.2 Newton – Euler Dynamic Formulation

11 NCTU - Human and Machine Lab

Angular velocity

$$W_{i+1} = \begin{cases} W_i + \vec{Z}_i \cdot \dot{q}_{i+1} & \text{R joint i+1} \\ W_i & \text{P joint i+1} \end{cases}$$

$$\dot{W}_{i+1} = \begin{cases} \dot{W}_i + \vec{Z}_i \cdot \ddot{q}_{i+1} + W_i \times (\vec{Z}_i \dot{q}_{i+1}) & \text{R joint i+1} \\ \dot{W}_i & \text{P joint i+1} \end{cases}$$

Linear velocity

$$V_{i+1} = \begin{cases} W_{i+1} \times P^*_{i+1} + V_i & \text{R joint i+1} \\ \vec{Z}_i \cdot \dot{q}_{i+1} + W_{i+1} \times P^*_{i+1} + V_i & \text{P joint i+1 where } P^*_{i+1} = P_{i+1} - P_i \end{cases}$$

$$\dot{V}_{i+1} = \begin{cases} \dot{W}_{i+1} \times P^*_{i+1} + W_{i+1} \times (W_{i+1} \times P^*_{i+1}) + \dot{V}_i & \text{R joint i+1} \\ \vec{Z}_i \cdot \ddot{q}_{i+1} + \dot{W}_{i+1} \times P^*_{i+1} + 2W_{i+1} \times (\vec{Z}_i \cdot \dot{q}_{i+1}) + W_{i+1} \times (W_{i+1} \times P^*_{i+1}) + \dot{V}_i & \text{P joint i+1} \end{cases}$$

6.2 Newton – Euler Dynamic Formulation

12 NCTU - Human and Machine Lab

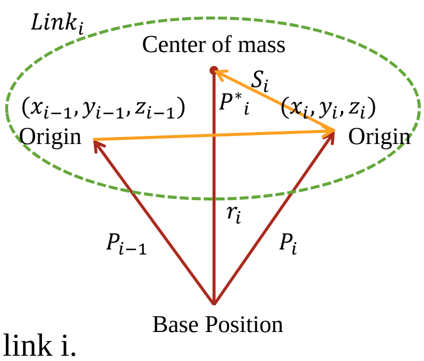
Newton's Equation

$$F_i = \frac{d}{dt}(m_i \cdot V_{Ci}) = m_i \cdot \dot{V}_{Ci}$$

Euler's Equation

$$N_i = \frac{d}{dt}(J_i \cdot W_i) = J_i \cdot \dot{W}_i + W_i \times (J_i \cdot W_i)$$

$$\left(\frac{dG}{dt}\right)_{space} = \left(\frac{dG}{dt}\right)_{body} + W \times G$$



- F_i: Total vector force exerted on link i.
- N_i: Total vector moment exerted on link i.
- J_i: The inertia matrix of link i about its c.o.m w.r.t base
- f_i: Force exerted on link i by link i-1
- n_i: Torque exerted on link i by link i-1

6.2 Newton – Euler Dynamic Formulation

13 NCTU - Human and Machine Lab

$$F_i = f_i - f_{i+1}$$

$$N_i = [n_i + (P_{i-1} - r_i) \times f_i] - [n_{i+1} + (P_i - r_i) \times f_{i+1}]$$

$$= (n_i - n_{i+1}) + (P_{i-1} - r_i) \times f_i - (P_i - r_i) \times f_{i+1}$$

$$= (n_i - n_{i+1}) + (P_{i-1} - r_i) \times F_i - P_i^* \times f_{i+1}$$

$$= (n_i - n_{i+1}) - (P_i^* + S_i) \times F_i - P_i^* \times f_{i+1}$$

$$\begin{cases} f_i = F_i + f_{i+1} \\ n_i = N_i + n_{i+1} + P_i^* \times f_{i+1} + (P_i^* + S_i) \times F_i \end{cases}$$

$$\begin{cases} \tau_i = n_i^t \vec{Z}_{i-1} \ , \ R \\ \tau_i = f_i^t \vec{Z}_{i-1} \ , \ P \end{cases}$$

6.2 Newton – Euler Dynamic Formulation

14 NCTU - Human and Machine Lab

- The two-link manipulator example

$$W_i = \dot{\theta}_1, \quad \dot{W}_i = \ddot{\theta}_1,$$

$$V_{Ci} = W_i \times S_i + V_i$$

$$\dot{V}_{Ci} = \dot{W}_i \times S_i + W_i \times (W_i \times S_i) + \dot{V}_i$$

$$\dot{V}_{i+1} = \dot{W}_{i+1} \times P_{i+1}^* + W_{i+1} \times (W_{i+1} \times P_{i+1}^*) + \dot{V}_i \ , \ R$$

$$\dot{V}_1 = \dot{W}_1 \times P_1^* + W_1 \times (W_1 \times P_1^*) + \dot{V}_0$$

$$\dot{V}_0 = \begin{pmatrix} 0 \\ g \\ 0 \end{pmatrix}$$

$$\dot{V}_1 = \dot{W}_1 \times \begin{pmatrix} l_1 C_1 \\ l_1 S_1 \\ 0 \end{pmatrix} + W_1 \times [W_1 \times \begin{pmatrix} l_1 C_1 \\ l_1 S_1 \\ 0 \end{pmatrix}] + \dot{V}_0 = \begin{pmatrix} -l_1 S_1 \ddot{\theta}_1 - l_1 C_1 \dot{\theta}_1^2 \\ l_1 C_1 \ddot{\theta}_1 - l_1 S_1 \dot{\theta}_1^2 + g \\ 0 \end{pmatrix}$$

6.2 Newton – Euler Dynamic Formulation

$$F_1 = m_1\dot{V}_1 \ , \ N_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$W_2 = \dot{\theta}_1 + \dot{\theta}_2$$

$$\dot{W}_2 = \ddot{\theta}_1 + \ddot{\theta}_2$$

$$\dot{V}_2 = \dot{W}_2 \times P_2^* + W_2 \times [W_2 \times P_2^*] + \dot{V}_1$$

$$= \begin{pmatrix} -l_2 S_{12}(\ddot{\theta}_1 + \ddot{\theta}_2) \\ l_2 C_{12}(\ddot{\theta}_1 + \ddot{\theta}_2) \\ 0 \end{pmatrix} + \begin{pmatrix} -l_2 C_{12}(\dot{\theta}_1 + \dot{\theta}_2)^2 \\ -l_2 S_{12}(\dot{\theta}_1 + \dot{\theta}_2)^2 \\ 0 \end{pmatrix} + \begin{pmatrix} -l_1 S_1 \ddot{\theta}_1 - l_1 C_1 \dot{\theta}_1^2 \\ l_1 C_1 \ddot{\theta}_1 - l_1 S_1 \dot{\theta}_1^2 + g \\ 0 \end{pmatrix}$$

6.2 Newton – Euler Dynamic Formulation

NCTU - Human and Machine Lab

$$F_2 = m_2\dot{V}_2, \quad N_2 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$f_2 = F_2 + f_3$$

$$n_2 = \begin{pmatrix} l_2 C_{12} \\ l_2 S_{12} \\ 0 \end{pmatrix} \times F_2 = \begin{pmatrix} 0 \\ 0 \\ m_2[l_1 l_2 C_2 \ddot{\theta}_1 + l_1 l_2 S_2 \dot{\theta}_1^2] \\ +m_2 l_2 g C_{12} + m_2 l_2^2 (\ddot{\theta}_1 + \ddot{\theta}_2) \end{pmatrix}$$

$$f_1 = F_1 + f_2$$

$$n_1 = n_2 + P_1^* \times f_2 + P_1^* \times F_1 = n_2 + \begin{pmatrix} l_1 C_1 \\ l_1 S_1 \\ 0 \end{pmatrix} \times f_2 + \begin{pmatrix} l_1 C_1 \\ l_1 S_1 \\ 0 \end{pmatrix} \times F_1$$

$$= \begin{pmatrix} 0 \\ 0 \\ m_2 l_1^2 \ddot{\theta}_1 - m_2 l_1 l_2 S_2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + m_2 l_1 l_2 C_2 (\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 l_1 g C_1 \end{pmatrix}$$

6.3 Recursive Newton-Euler Dynamic Algorithm

17 NCTU - Human and Machine Lab

- Recursive Newton-Euler Dynamic Algorithm
  - It will be more efficient to compute the joint input forces and torques referenced to its own coordinates due to the inertia matrix J_i being dependent on the orientation of link i, which is changing.
  - Reference:
    - ① "On-line Computational Scheme for mechanical Manipulators" J.Y.S. Luh, M.W. Walker and R.P.C. Paul J., Dynamic systems Measurement Control, pp 69-76, 1980.
    - ② "Efficient Dynamic Computer Simulation of Robot Mechanics" M.W. Walker and D.E. Orin J. Dynamic Systems, Measurement, Control, pp. 205-211, 1982.

6.3 Recursive Newton-Euler Dynamic Algorithm

18 NCTU - Human and Machine Lab

$$W_{i+1} = \begin{cases} W_i + \vec{Z}_i \dot{q}_{i+1} & \text{R joint i+1} \\ W_i & \text{P joint i+1} \end{cases}$$

$$A^o_{i+1} W_{i+1} = \begin{cases} A^i_{i+1}(A^o_i W_i + Z_0 \dot{q}_{i+1}) & \text{R joint i+1} \\ A^i_{i+1}(A^o_i W_i) & \text{P joint i+1} \end{cases}$$

$$A^o_{i+1} \dot{W}_{i+1} = \begin{cases} A^i_{i+1}[A^o_i \dot{W}_i + Z_0 \ddot{q}_{i+1} + (A^o_i W_i) \times (Z_0 \dot{q}_{i+1})] & \text{R joint i+1} \\ A^i_{i+1}(A^o_i \dot{W}_i) & \text{P joint i+1} \end{cases}$$

$$Z_0 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

6.3 Recursive Newton-Euler Dynamic Algorithm

$$A_{i+1}^{o}\dot{V}_{i+1} = (A_{i+1}^{o}\dot{W}_{i+1})\times(A_{i+1}^{o}P_{i+1}^{*})+(A_{i+1}^{o}W_{i+1})$$
$$\times[(A_{i+1}^{o}W_{i+1})\times(A_{i+1}^{o}P_{i+1}^{*})]+A_{i+1}^{i}(A_{i}^{o}\dot{V}_{i})$$
$$A_{i+1}^{i}(Z_{0}\ddot{q}_{i+1}+A_{i}^{o}\dot{V}_{i})+(A_{i+1}^{o}\dot{W}_{i+1})\times(A_{i+1}^{o}P_{i+1}^{*})$$
$$+2(A_{i+1}^{o}W_{i+1})\times(A_{i+1}^{i}Z_{0}\dot{q}_{i+1})+(A_{i+1}^{o}W_{i+1})$$
$$\times[A_{i+1}^{o}W_{i+1}\times(A_{i+1}^{o}P_{i+1}^{*})]$$

$$A_{i}^{o}\dot{\hat{V}}_{i}=(A_{i}^{o}\dot{W}_{i})\times(A_{i}^{o}\hat{S}_{i})+(A_{i}^{o}W_{i})\times[(A_{i}^{o}W_{i})\times(A_{i}^{o}\hat{S}_{i})]+A_{i}^{o}\dot{V}_{i}$$

$$A_{i}^{o}F_{i}=m_{i}A_{i}^{o}\dot{\hat{V}}_{i}$$

$$A_{i}^{o}N_{i}=(A_{i}^{o}J_{i}A_{o}^{i})(A_{i}^{o}\dot{W}_{i})+(A_{i}^{o}W_{i})\times[(A_{i}^{o}J_{i}A_{o}^{i})(A_{i}^{o}W_{i})]$$

- Where A_i^0, J_i, A_0^i is the inertia matrix of link i about its center of mass referred to its own coordinates (x_i, y_i, z_i)

6.3 Recursive Newton-Euler Dynamic Algorithm

- Parallel axis theorem

$$J_0 = J_G + M[(r\cdot r)I - rr^T]$$

  - For obtaining the moment of inertia matrix J_0 of a rigid body about an arbitrary origin of coordinate 0 in terms of the inertia matrix J_G relative to the center of mass

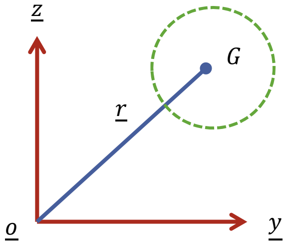
$$A_{i}^{o}f_{i}=A_{i}^{i+1}(A_{i+1}^{o}f_{i+1})+A_{i}^{o}F_{i}$$

$$f_{i}=F_{i}+f_{i+1}$$

$$n_{i}=N_{i}+n_{i+1}+P_{i}^{*}\times f_{i+1}+(P_{i}^{*}+S_{i})\times F_{i}$$

$$\tau_{i}=n_{i}^{t}\vec{Z}_{i-1}\ ,\ R$$

$$\tau_{i}=f_{i}^{t}\vec{Z}_{i-1}\ ,\ P$$



$$A_{i}^{o}n_{i}=A_{i}^{i+1}[A_{i+1}^{o}n_{i+1}+(A_{i+1}^{o}\underline{P}_{i}^{*})\times(A_{i+1}^{o}f_{i+1})]+(A_{i}^{o}\underline{P}_{i}^{*}+A_{i}^{o}\underline{\hat{S}}_{i})\times(A_{i}^{o}\underline{\hat{F}}_{i})+A_{i}^{o}N_{i}$$

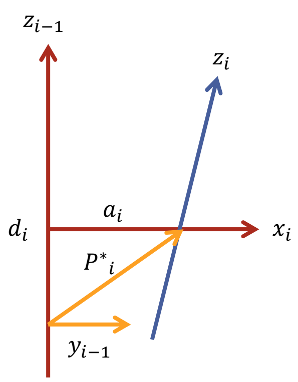
6.3 Recursive Newton-Euler Dynamic Algorithm

21 NCTU - Human and Machine Lab

$$\tau_i = \begin{cases} (A_i^o n_i)^T (A_i^{i-1} Z_0) & \text{R joint i+1} \\ (A_i^o f_i)^T (A_i^{i-1} Z_0) & \text{P joint i+1} \end{cases}$$

$$A_i^o P_i^* = A_i^{i-1} \cdot {}^{i-1}P_i^* = \begin{pmatrix} a_i \\ d_i \sin\alpha_i \\ d_i \cos\alpha_i \end{pmatrix}$$

$${}^{i-1}P_i^* = \begin{pmatrix} a_i \cos\theta_i \\ a_i \sin\theta_i \\ d_i \end{pmatrix}$$



6.3 Recursive Newton-Euler Dynamic Algorithm

22 NCTU - Human and Machine Lab

① Know $\underline{W}_0, \underline{\dot{W}}_0, \underline{\dot{V}}_0, \hat{S}_i, m_i, J_i, f_{n+1}$ *and* n_{n+1}

② For i=0,..., n-1

$$\underline{W}_{i+1} = \underline{W}_i + \ldots$$
$$\underline{\dot{W}}_{i+1} = \underline{\dot{W}}_i + \ldots$$
$$\underline{\dot{V}}_{i+1} = \underline{\dot{V}}_i + \ldots$$

③ For i = n,...,1

$$\underline{\dot{\hat{V}}}_i = \ldots$$
$$F_i = m_i \underline{\dot{\hat{V}}}_i$$
$$N_i = J_i \underline{\dot{W}}_i + \underline{W}_i \times (J_i \underline{\dot{W}}_i)$$
$$f_i = f_{i+1} + F_i$$
$$n_i = \ldots \quad ,compute \;\; \tau_i$$

6.4 Dynamic Simulator

23 NCTU - Human and Machine Lab

NxN inertia matrix

Nx1 vector specifying the gravity effect

$$\tau = H(q)\ddot{q} + C(q \cdot \dot{q})\dot{q} + G(q) + K^T(q) f_{ext}$$

NxN inertia specifying centrifugal and Coriolis effect

Transpose of the 6xN Jacobian matrix

$$F \cdot \delta x = \tau \cdot \delta\theta$$

$$F^t \cdot \delta x = \tau^t \cdot \delta\theta$$

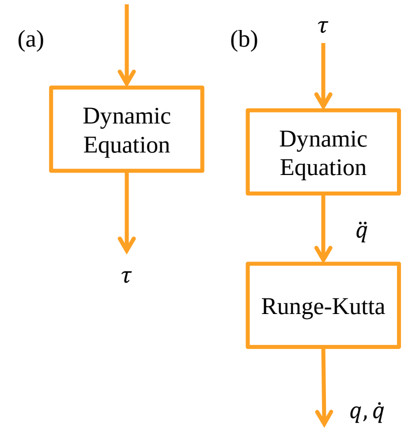
$$\delta x = J \cdot \delta\theta$$

$$F^t J \delta\theta = \tau^t \delta\theta \Rightarrow F^t J = \tau^t \Rightarrow J^t F = \tau$$

6.4 Dynamic Simulator

24 NCTU - Human and Machine Lab

$q, \dot{q}, \ddot{q}, f_{ext}$ → Stands for both force and movement



- Suppose that a subroutine SUB1 $(q, \dot{q}, \ddot{q}, f_{ext}, \tau)$ is written for computing τ in case (a)
- And another subroutine SUB2 $(q, \ddot{q}, \tau)$, it is identical to SUB1 except that all the codes relating to velocity, gravity, and external forces and moments are eliminated. That is, it stands for $\tau = H(q)\ddot{q}$

6.4 Dynamic Simulator

- Then the simulation in case (b) can be executed as follows:
  1. Utilize SUB1 and let $\ddot{q} = 0$ then a bias vector b can be computed:

     $$\underline{b} = C(q, \dot{q})\dot{q} + G(q) + K^{\tau}(q)\, f_{ext}$$

  2. Solve $H(q)$ by utilizing SUB2 and letting $\ddot{q} = \underline{l_j}$, where $\underline{l_j}$ is an Nx1 vector with the jth element equal to 1 and 0 everywhere else.
  3. Solve form $\mathrm{H(q)}\ddot{q} = \tau - b$ (numerical method, e.g. Gaussian Elimination)
  4. Integrating $\ddot{q}$ to find $\dot{q}$ and q. (Runge-Kutta)