



Manipulator Dynamics 1



Introduction

Forward / Inverse Dynamics



Forward Dynamics

Problem

Given: Joint torques and links geometry, mass, inertia, friction, joint torques

Compute: Angular acceleration of the links (solve differential equations)

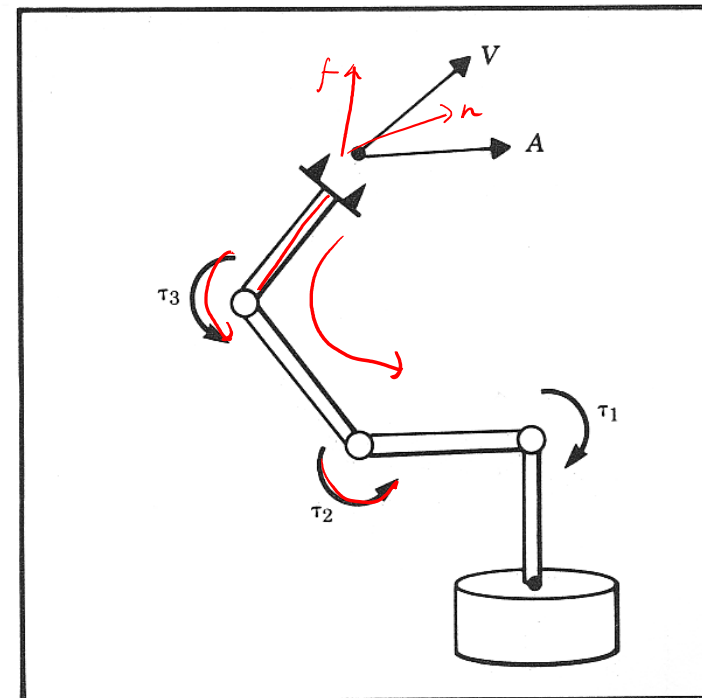
Solution

solve a set of differential equations

Dynamic Equations - Newton-Euler method or Lagrangian Dynamics

$$\tau = M(\Theta)\ddot{\Theta} + V(\Theta, \dot{\Theta}) + G(\Theta) + F(\Theta, \dot{\Theta})$$

$$\left\{ \begin{array}{l} \tau_i \\ \text{Link}_i(x, y, z) \\ m_i \\ I_i \\ P_{Ci} \\ f_i \\ n_i \end{array} \right\} \left\{ \begin{array}{l} \Theta \\ \dot{\Theta} \\ \ddot{\Theta} \end{array} \right\}$$





Inverse Dynamics

Problem

Given: Angular acceleration, velocity and angles of the links in addition to the links geometry, mass, inertia, friction

Compute: Joint torques

Solution

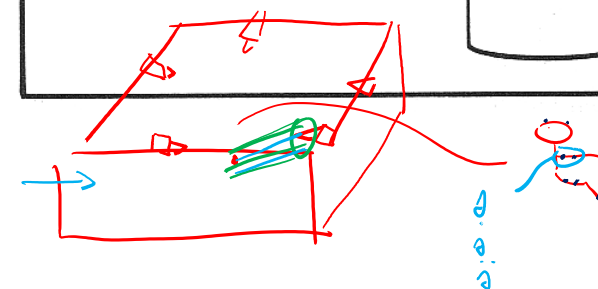
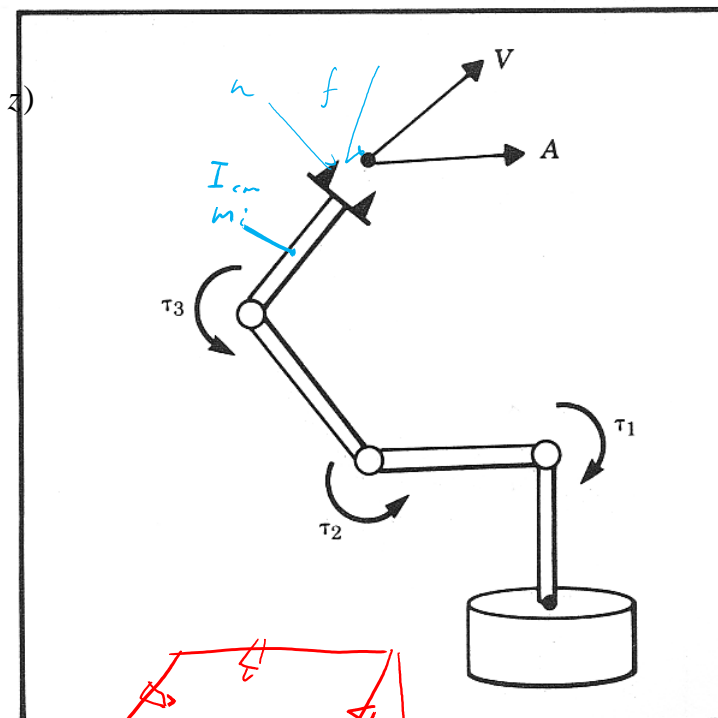
Solve a set of algebraic equations

Dynamic Equations - Newton-Euler method or
 ➔ Lagrangian Dynamics

$$\tau = \underbrace{M(\Theta)\ddot{\Theta}}_{\text{inertia}} + \underbrace{V(\Theta, \dot{\Theta})}_{\text{Coriolis/centrifugal}} + \underbrace{G(\Theta)}_{\text{gravity}} + \underbrace{F(\Theta, \dot{\Theta})}_{\text{friction}}$$

Handwritten notes: Blue arrows point from the terms in the equation to the list of parameters. A blue circle highlights the G(Θ) term. A blue arrow points from the τ term to the list of parameters.

- Θ
- $\dot{\Theta}$
- $\ddot{\Theta}$
- $Link_i(x, y, z)$
- m_i
- I_i
- P_{Ci}
- f_i
- n_i
- $\{\tau$



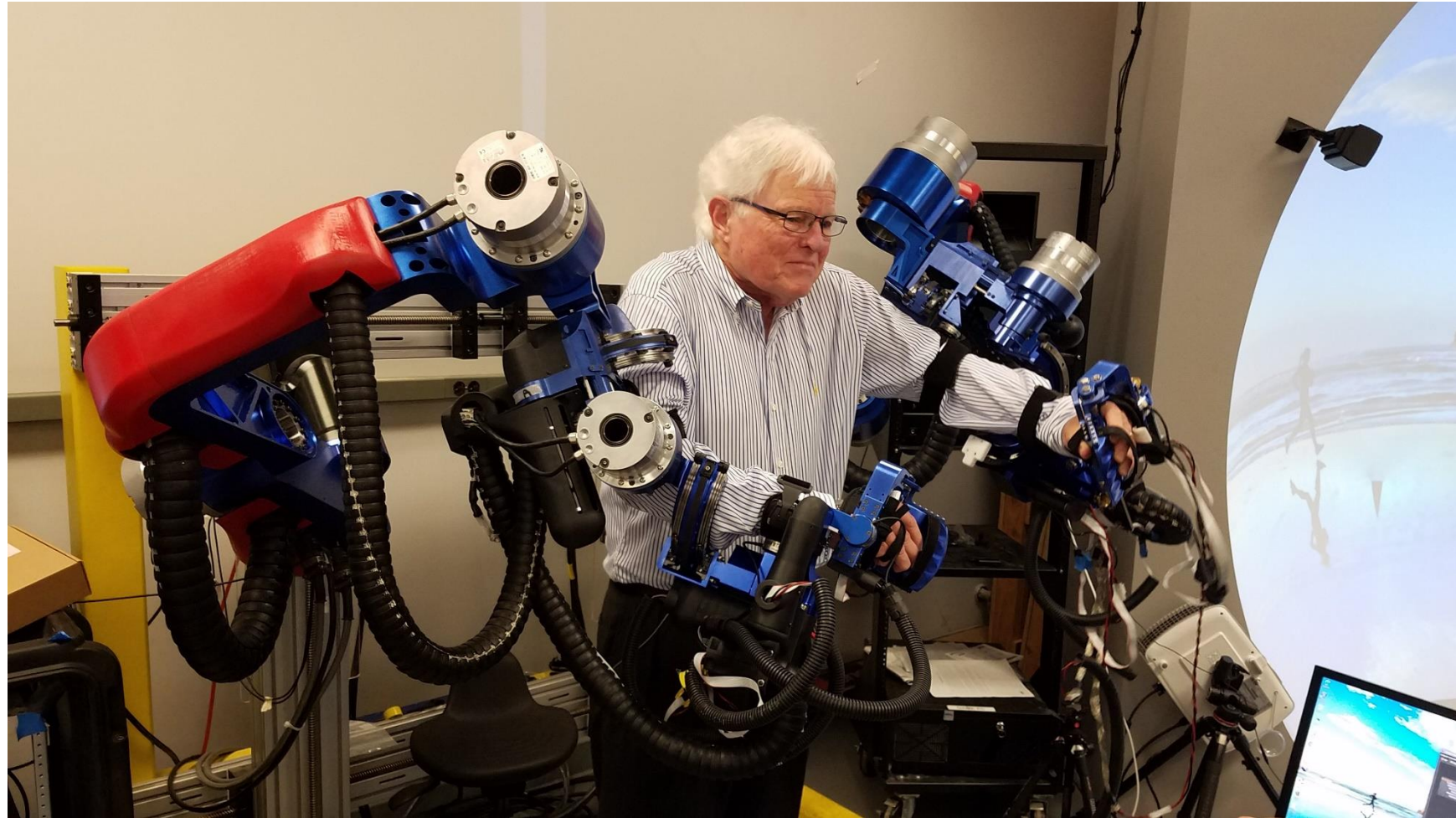


Applications

Human Arm Dynamics – Exoskeleton Design
Control



Dynamics Modeling Application – Motivation Exoskeleton Design





Dynamics Modeling Application – Motivation Exoskeleton Design

- **Activities of Daily Living**

1. Arm in lap
2. Arm reach to head level
3. Arm reach to right, head level
4. Arm reach to left, head level
5. Arm reach right, move object to left side
6. Open door
7. Open Drawer/Close Drawer
8. Move object at waist level
9. Pick up phone on table/hang up
10. Pick up phone on wall/hang up
11. Eat with fork
12. Eat with spoon
13. Eat with hands
14. Drink with cup
15. Eat with spoon
16. Pour from bottle
17. Brush teeth
18. Comb hair
19. Wash face
20. Wash neck
21. Shave
22. Eat with fork (power – disabled grasp)
23. Eat with spoon (power – disabled grasp)
24. Full workspace motion

[[CP19](#)] Rosen Jacob, Joel C. Perry, Nathan Manning, Stephen Burns, Blake Hannaford,
The Human Arm Kinematics and Dynamics During Daily Activities – Toward a 7 DOF Upper Limb Powered Exoskeleton, -
ICAR 2005 – Seattle WA, July 2005



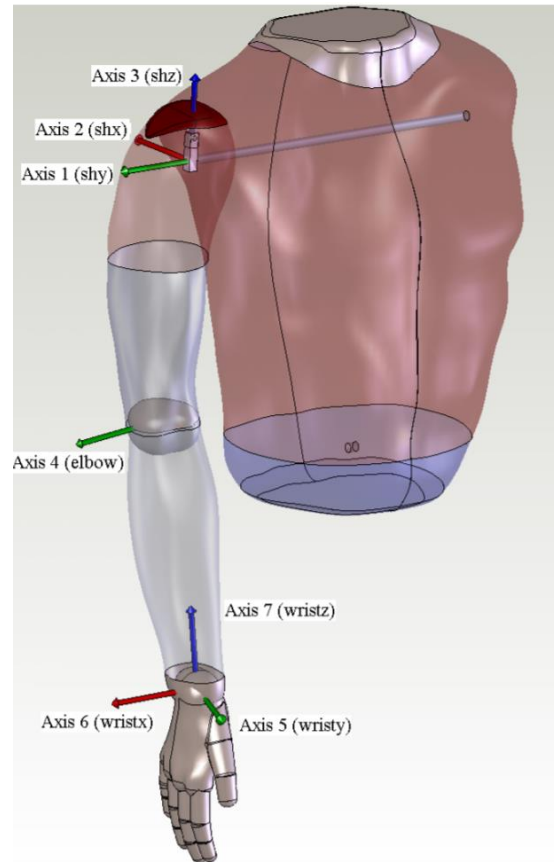
Dynamics Modeling Application – Motivation Exoskeleton Design



[[CP19](#)] Rosen Jacob, Joel C. Perry, Nathan Manning, Stephen Burns, Blake Hannaford,
The Human Arm Kinematics and Dynamics During Daily Activities – Toward a 7 DOF Upper Limb Powered Exoskeleton, -
ICAR 2005 – Seattle WA, July 2005



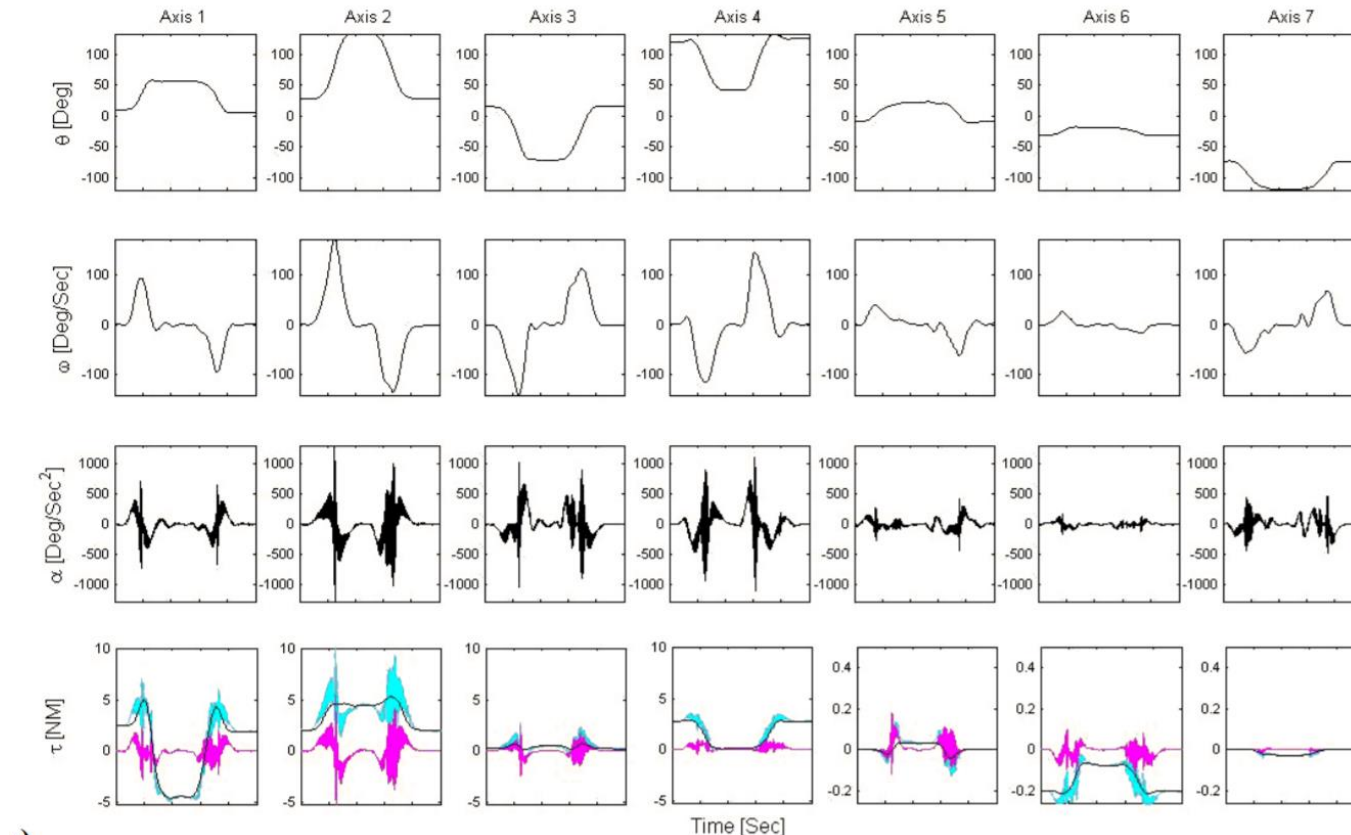
Dynamics Modeling Application – Motivation Exoskeleton Design



[[CP19](#)] Rosen Jacob, Joel C. Perry, Nathan Manning, Stephen Burns, Blake Hannaford,
The Human Arm Kinematics and Dynamics During Daily Activities – Toward a 7 DOF Upper Limb Powered Exoskeleton, -
ICAR 2005 – Seattle WA, July 2005



Dynamics Modeling Application – Motivation Exoskeleton Design

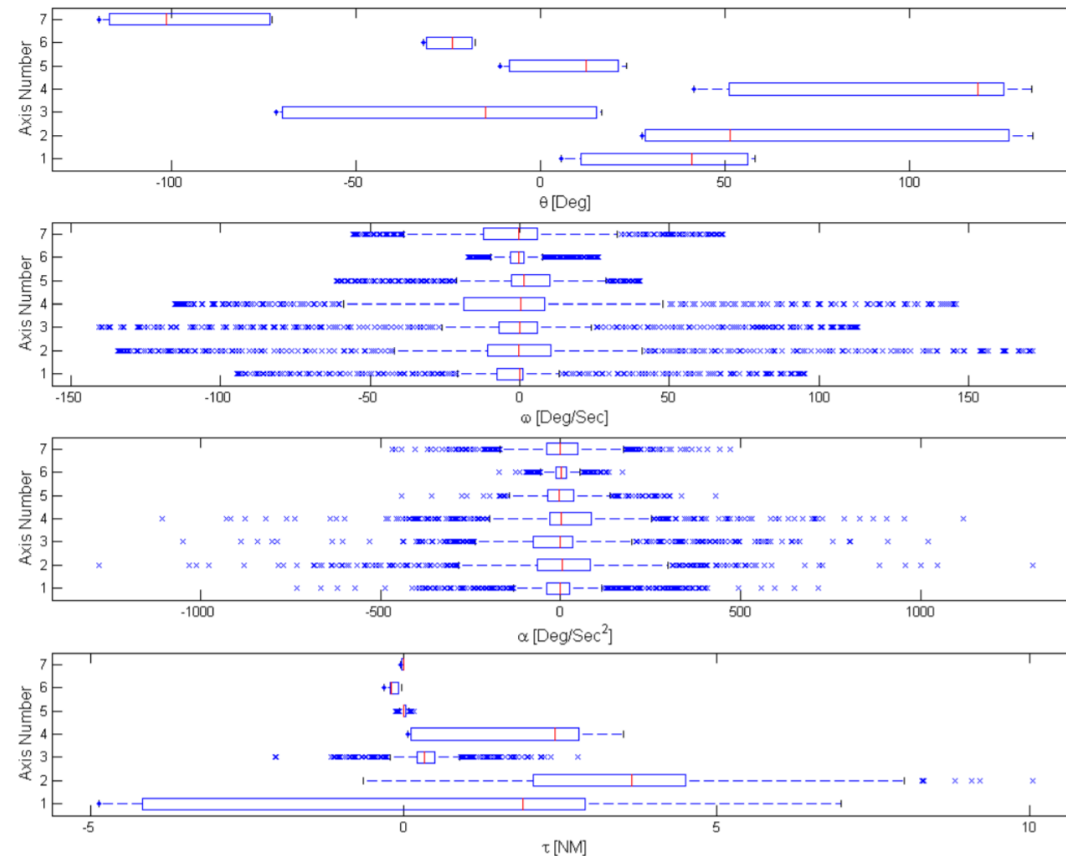


Time histories and statistics of the kinematics and dynamics of the human arm during an arm reach to head level (action 2): (a) Time histories of the joint kinematics and dynamics (b) Statistical distribution of the joint kinematics and dynamics. The three torque curves in (a) illustrate the total joint axis torque (cyan), in comparison to the gravitational torque (black) and the combined torque due to inertial, centrifugal, and coriolis terms (magenta). The line box plots of (b) indicate the lower quartile, median, and upper quartile values. The dashed lines extend beyond the upper and lower quartiles by one and a half times the interquartile range. Data that lies outside of this range is displayed with the symbol 'x'.

[CP19] Rosen Jacob, Joel C. Perry, Nathan Manning, Stephen Burns, Blake Hannaford,
The Human Arm Kinematics and Dynamics During Daily Activities – Toward a 7 DOF Upper Limb Powered Exoskeleton, -
ICAR 2005 – Seattle WA, July 2005



Dynamics Modeling Application – Motivation Exoskeleton Design

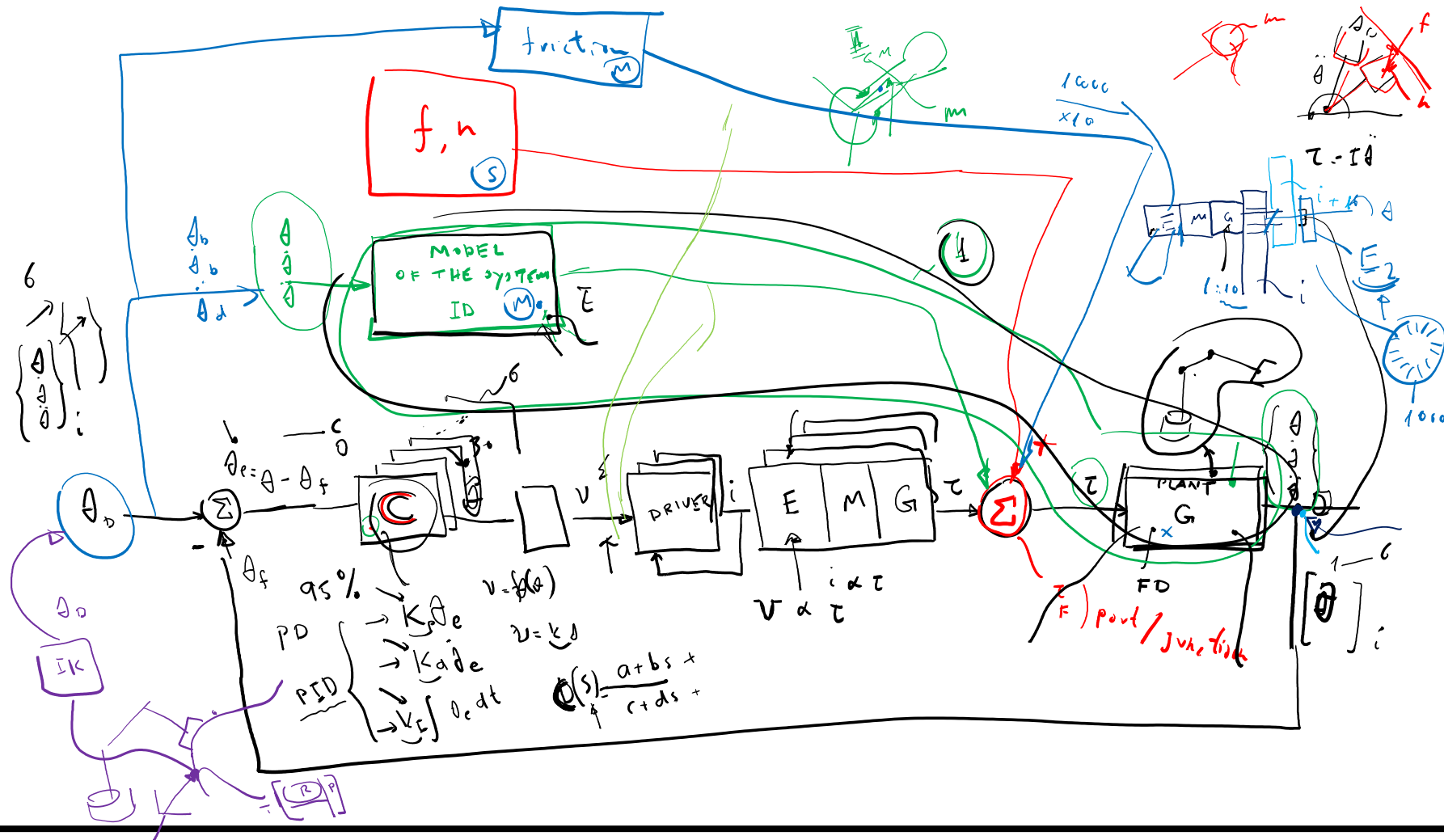


Time histories and statistics of the kinematics and dynamics of the human arm during an arm reach to head level (action 2): (a) Time histories of the joint kinematics and dynamics (b) Statistical distribution of the joint kinematics and dynamics. The three torque curves in (a) illustrate the total joint axis torque (cyan), in comparison to the gravitational torque (black) and the combined torque due to inertial, centrifugal, and coriolis terms (magenta). The line box plots of (b) indicate the lower quartile, median, and upper quartile values. The dashed lines extend beyond the upper and lower quartiles by one and a half times the interquartile range. Data that lies outside of this range is displayed with the symbol 'x'.

[CP19] Rosen Jacob, Joel C. Perry, Nathan Manning, Stephen Burns, Blake Hannaford,
The Human Arm Kinematics and Dynamics During Daily Activities – Toward a 7 DOF Upper Limb Powered Exoskeleton, -
ICAR 2005 – Seattle WA, July 2005



Dynamics Model Application – Motivation Position Control





Manipulator Dynamics – Newton Euler Equations



Iterative Newton Euler Equations

Steps of the Algorithm

- (1) Outward Iterations

- Starting With velocities and accelerations of the base

$${}^0\omega_0 = 0, {}^0\dot{\omega}_0 = 0, {}^0v_0 = 0, {}^0\dot{v}_0 = +g\hat{z}$$

- Calculate velocities accelerations, along with forces and torques (at the CM)

$$\omega, \dot{\omega}, \dot{v}, \dot{v}_{CM}, F, N$$

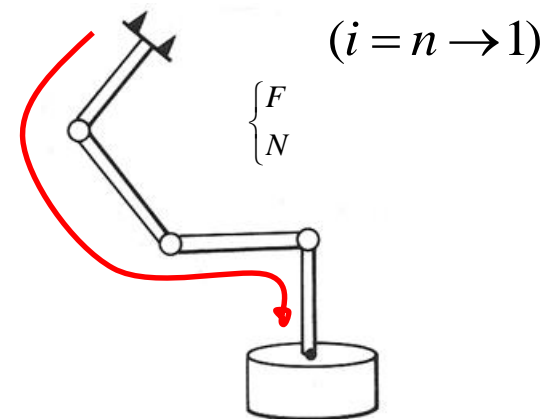
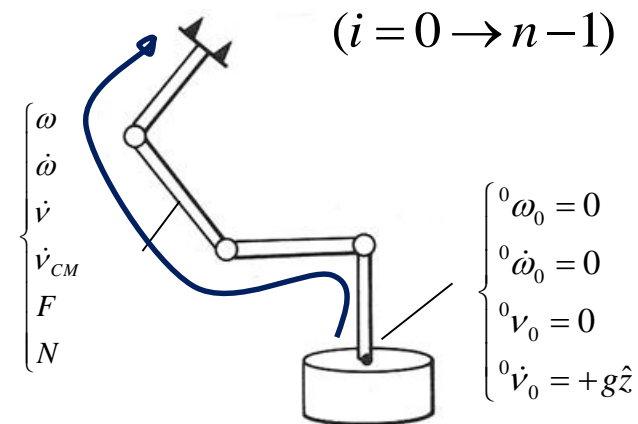
- (2) Inward Iteration ($i = n \rightarrow 1$)

- Starting with forces and torques (at the CM)

$$F, N$$

- Calculate forces and torques at the joints

$$f, n$$





Iterative Newton-Euler Equations - Solution Procedure

Phase 1: Outward Iteration

Outward Iteration: $i : 0 \rightarrow 5$

- Calculate the link velocities and accelerations iteratively from the robot's base to the end effector

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

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

$${}^{i+1}\dot{v}_{i+1} = {}^{i+1}R(i\dot{\omega}_i \times {}^iP_{i+1} + {}^i\omega_i \times ({}^i\omega_i \times {}^iP_{i+1}) + {}^i\dot{v}_i)$$

$${}^{i+1}\dot{v}_{Ci+1} = {}^{i+1}\dot{\omega}_{i+1} \times {}^{i+1}P_{Ci+1} + {}^{i+1}\omega_{i+1} \times ({}^{i+1}\omega_{i+1} \times {}^{i+1}P_{Ci+1}) + {}^{i+1}\dot{v}_{i+1}$$

- Calculate the force and torques applied on the CM of each link using the Newton and Euler equations

$${}^{i+1}F_{i+1} = m_{i+1} {}^{i+1}\dot{v}_{Ci+1}$$

$${}^{i+1}N_{i+1} = {}^C {}^{i+1}I_{i+1} {}^{i+1}\dot{\omega}_{i+1} + {}^{i+1}\omega_{i+1} \times {}^C {}^{i+1}I_{i+1} {}^{i+1}\omega_{i+1}$$



Iterative Newton-Euler Equations - Solution Procedure

Phase 2: Inward Iteration

Inward Iteration: $i : 6 \rightarrow 1$

- Use the forces and torques generated at the joints starting with forces and torques generating by interacting with the environment (that is, tools, work stations, parts etc.) at the end effector all the way to the robot's base.

$${}^i f_i = {}^i R^{i+1} f_{i+1} + {}^i F_i$$

$${}^i n_i = {}^i N_i + {}^i R^{i+1} n_{i+1} + {}^i P_{Ci} \times {}^i F_i + {}^i P_{i+1} \times {}^i R^{i+1} f_{i+1}$$

$$\tau_i = {}^{i+1} n_{i+1}^T {}^i \hat{Z}_i$$



Dynamics - Newton-Euler Equations

- To solve the Newton and Euler equations, we'll need to develop mathematical terms for:

→ \dot{v}_c – The linear acceleration of the center of mass

→ $\dot{\omega}$ – The angular acceleration

→ cI – The Inertia tensor (moment of inertia)

→ F – The sum of all the forces applied on the center of mass

→ N – The sum of all the moments applied on the center of mass

$$\begin{cases} F = m\dot{v}_c \\ N = {}^cI\dot{\omega} + \omega \times {}^cI\omega \end{cases}$$



Newton Euler Equations Derivation Based on Momentum



Dynamics - Newton-Euler Equations

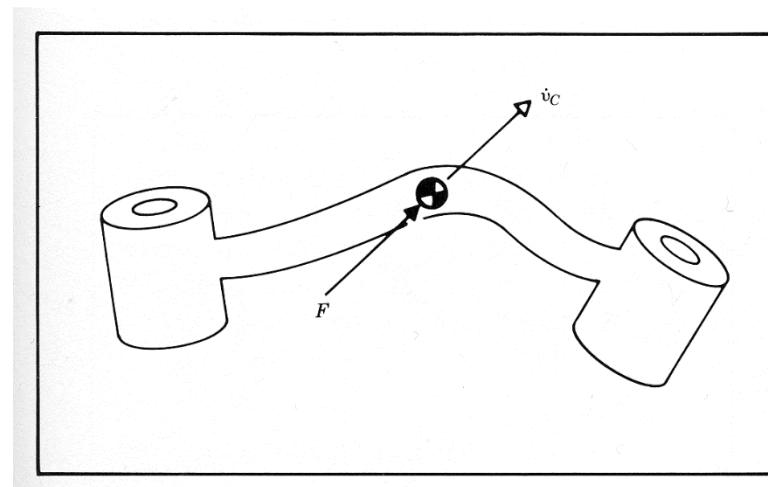
Newton Equation

- For a rigid body (like a robot link) whose center of mass is experiencing an acceleration, the force acting at the center of mass that caused the acceleration is given by Newton's equation

$$F = \frac{d(mv_c)}{dt}$$

- For our robot manipulators, whose link masses are constant, this equation simplifies to

$$F = m\dot{v}_c = ma_c$$





Dynamics - Newton-Euler Equations

Eular Equation

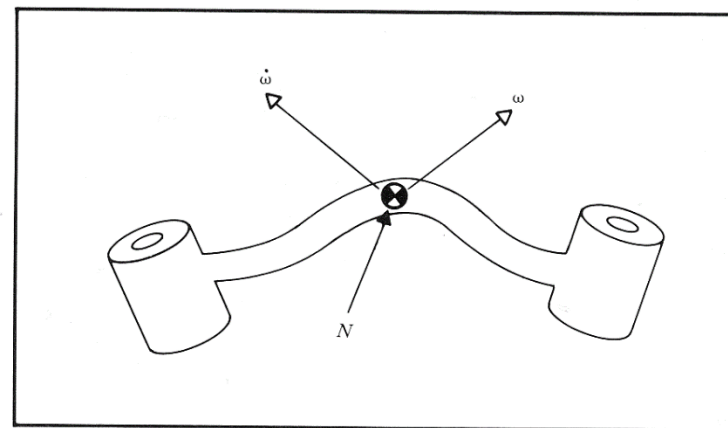
- For a rotating rigid body, the moment that causes an angular acceleration is given by Euler's equation

$$\underline{N} = \frac{d(\underline{I}\underline{\omega})}{dt}$$

- For our robot manipulators, whose **link moment of inertia is constant**, this equation simplifies to

$$\underline{N} = \underline{I}\dot{\underline{\omega}} + \underline{\omega} \times \underline{I}\underline{\omega}$$

- The second term on the right will be non-zero when the link's angular velocity vector is not coincident with the link's principle axis of inertia.



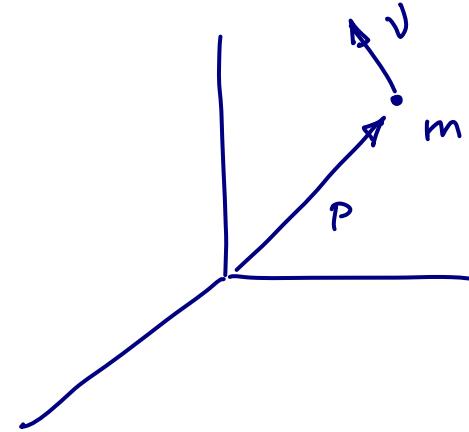


Linear Momentum – Particle

$$L = mv = m \frac{dP}{dt}$$

$$\frac{dL}{dt} = \frac{d}{dt}(mv) = m \frac{d^2 P}{dt^2} = \sum f_i \text{ external}$$

$$\sum f_i \text{ external} = ma$$





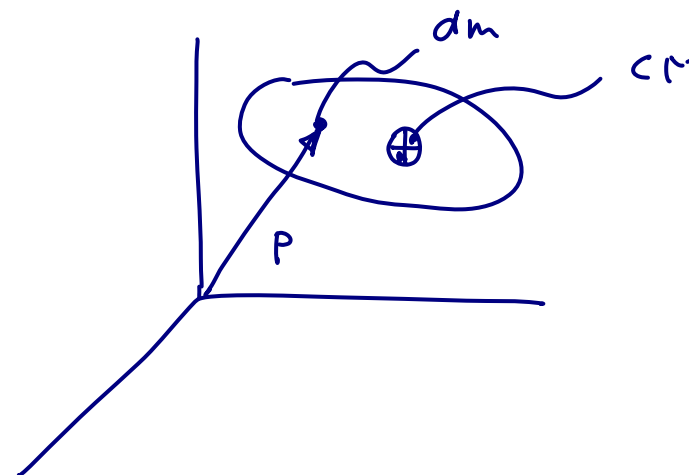
Linear Momentum – Rigid Body

$$\sum f_{i \text{ external}} = dm_i \frac{d^2 P}{dt^2}$$

- Integrate over the entire volume of the rigid body

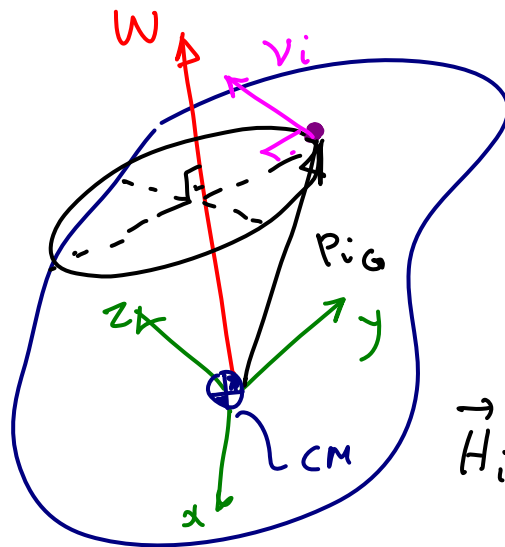
$$F = M \frac{d^2}{dt^2} \underbrace{\int_V \frac{P dm}{M}}_{P_{CM}}$$

$$F = M \frac{d^2}{dt^2} P_{CM} = ma_{CM}$$





Angular Momentum – Particle



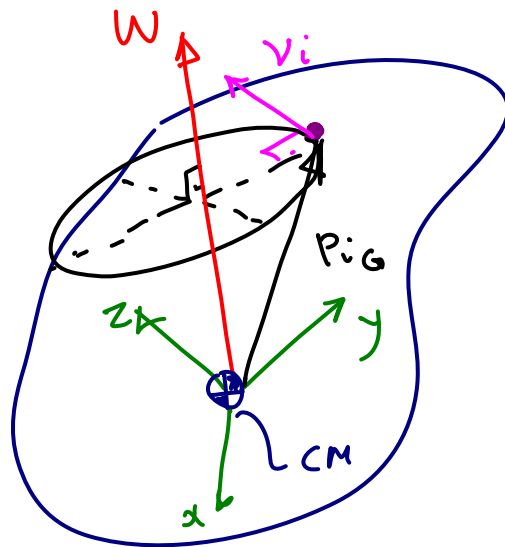
$$\vec{H}_{iG} = \vec{P}_{iG} \times m_i \vec{v}_{iG}$$

$$\vec{v}_{iG} = \vec{\omega} \times \vec{P}_{iG}$$

$$\vec{H}_{iG} = \vec{P}_{iG} \times m_i (\vec{\omega} \times \vec{P}_{iG}) = m_i \vec{P}_{iG} \times (\vec{\omega} \times \vec{P}_{iG})$$



Angular Momentum – Rigid Body



$$H_G = \sum H_{iG} = \sum_i m_i P_{iG} \times (\omega \times P_{iG}) \quad | \quad m_i = \int dv$$

$$H_G = \int_V \underbrace{P \times \omega \times P}_{P(-P)\omega} \underbrace{\int dv}_{m_i} = \omega \int_V \underbrace{P(-P)}_I \int dv = \omega I$$



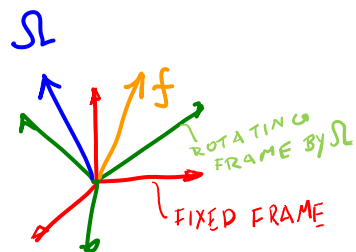
Angular Momentum – Rigid Body

$$\sum \tau_i = \frac{d}{dt} H_G = \frac{d}{dt} (I\omega)$$

$$\sum \tau_i = I\dot{\omega} + \omega \times I\omega$$



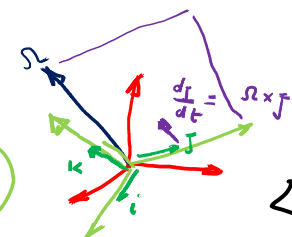
Transport Theory



$$\frac{d}{dt} f = \left(\left(\frac{d}{dt} \right)_r + \Omega \times \right) f \quad \text{if } f = w \Rightarrow \dot{w} + w \times w$$

$$f(t) = f_x(t)\hat{i} + f_y(t)\hat{j} + f_z(t)\hat{k}$$

$$\begin{aligned} \frac{d}{dt} f &= \underbrace{\frac{df_x}{dt} \hat{i}}_{\text{fixed}} + \underbrace{\frac{d\hat{i}}{dt} f_x}_{\text{rotating}} + \underbrace{\frac{df_y}{dt} \hat{j}}_{\text{fixed}} + \underbrace{\frac{d\hat{j}}{dt} f_y}_{\text{rotating}} + \underbrace{\frac{df_z}{dt} \hat{k}}_{\text{fixed}} + \underbrace{\frac{d\hat{k}}{dt} f_z}_{\text{rotating}} \\ &= \frac{df_x}{dt} \hat{i} + \frac{df_y}{dt} \hat{j} + \frac{df_z}{dt} \hat{k} + \left[\Omega \times (f_x \hat{i} + f_y \hat{j} + f_z \hat{k}) \right] = \left(\frac{d}{dt} \right)_r + \Omega \times f(t) \end{aligned}$$



$$\frac{d}{dt} f = \frac{df_x}{dt} \hat{i} + \frac{d\hat{i}}{dt} f_x + \frac{df_y}{dt} \hat{j} + \frac{d\hat{j}}{dt} f_y + \frac{df_z}{dt} \hat{k} + \frac{d\hat{k}}{dt} f_z$$



Manipulator Dynamics – Newton Euler Equations

The Angular Acceleration

$$\dot{\omega}$$



Dynamics - Newton-Euler Equations

- To solve the Newton and Euler equations, we'll need to develop mathematical terms for:

\dot{v}_c – The linear acceleration of the center of mass

$\dot{\omega}$ – The angular acceleration

cI – The Inertia tensor (moment of inertia)

F – The sum of all the forces applied on the center of mass

N – The sum of all the moments applied on the center of mass

$$F = m\dot{v}_c$$

$$N = {}^cI\dot{\omega} + \omega \times {}^cI\omega$$



Deriving Angular Acceleration – Vector Approach

$$\dot{\omega}$$



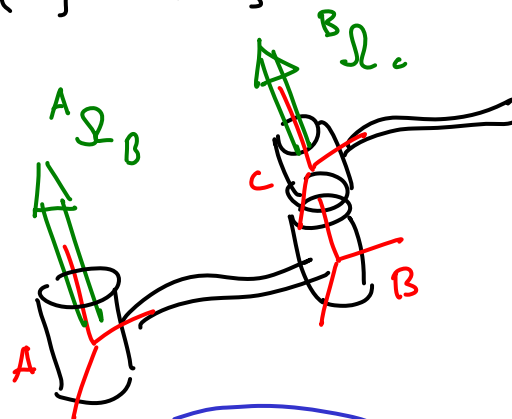
Propagation of Acceleration – Angular Vector Approach

3 Frames

{B} rotates relative to {A} with ${}^A\Omega_B$

{C} rotates relative to {B} with ${}^B\Omega_C$

Calculate ${}^A\dot{\Omega}_C$



$${}^A\Omega_C = {}^A\Omega_B + {}^A_B R {}^B\Omega_C \quad (*)$$

$${}^A\dot{\Omega}_C = {}^A\dot{\Omega}_B + \frac{d}{dt} ({}^A_B R {}^B\Omega_C)$$

$$\frac{d}{dt} ({}^A_B R {}^B\Omega_C) = {}^A_B R \dot{{}^B\Omega}_C + {}^A\Omega_B \times {}^A_B R {}^B\Omega_C$$

$${}^A\dot{\Omega}_C = {}^A\dot{\Omega}_B + {}^A_B R \dot{{}^B\Omega}_C + {}^A\Omega_B \times {}^A_B R {}^B\Omega_C$$

$A \rightarrow 0$
 $B \rightarrow i$
 $C \rightarrow i+1$



Propagation of Acceleration – Angular Vector Approach

$$\dot{\Omega}_{i+1} = \dot{\Omega}_i + {}^i R^i \dot{\Omega}_{i+1} + \underbrace{{}^o \Omega_i \times {}^i R^i \Omega_{i+1}}$$

Multiply by ${}^{i+1} R^i$ *

$$\underbrace{{}^{i+1} R^i \dot{\Omega}_{i+1}} = \underbrace{{}^{i+1} R^o \dot{\Omega}_i} + \underbrace{{}^{i+1} R^i \dot{\Omega}_{i+1}} + \underbrace{{}^{i+1} R^o \left(\Omega_i \times {}^i R^i \Omega_{i+1} \right)}$$

$${}^{i+1} \dot{W}_{i+1} = \underbrace{{}^{i+1} \dot{W}_i} + \underbrace{{}^{i+1} R^i \dot{\Omega}_{i+1}} + \underbrace{{}^{i+1} R^o \Omega_i \times {}^{i+1} R^i \Omega_{i+1}}$$

$${}^{i+1} \dot{W}_{i+1} = {}^{i+1} R^i \dot{W}_i + \begin{bmatrix} 0 \\ 0 \\ \ddot{A}_{i+1} \end{bmatrix} + {}^i R^{i+1} W_i \times \begin{bmatrix} 0 \\ 0 \\ \dot{A}_{i+1} \end{bmatrix}$$



Deriving Angular Acceleration – Matrix Approach

$$\dot{\omega}$$



Propagation of Acceleration – Angular Matrix Approach

- To derive a general formula for the angular acceleration, we will differentiate the angular velocity

$${}^{i+1}\omega_{i+1} = {}^{i+1}R^i \omega_i + \begin{bmatrix} 0 & 0 & \dot{\theta}_{i+1} \end{bmatrix}^T$$

- Applying the chain rule, we find:

$${}^{i+1}\dot{\omega}_{i+1} = {}^{i+1}R^i \dot{\omega}_i + \boxed{{}^{i+1}\dot{R}^i} \omega_i + \begin{bmatrix} 0 & 0 & \ddot{\theta}_{i+1} \end{bmatrix}^T$$

- Recall that

$${}^A_B \dot{R} = \frac{d}{dt} \begin{bmatrix} {}^A_B R \end{bmatrix} = \begin{bmatrix} 0 & -\Omega_z & \Omega_y \\ \Omega_z & 0 & -\Omega_x \\ -\Omega_y & \Omega_x & 0 \end{bmatrix} {}^A_B R = \boxed{{}^A_B \dot{R}_\Omega {}^A_B R}$$

- Substitution of this result yields



Propagation of Acceleration - Angular

Matrix form (Revolute Joint)

$${}^{i+1}\dot{\omega}_{i+1} = {}^{i+1}R^i \dot{\omega}_i + {}^{i+1}\dot{R}_\Omega^i {}^{i+1}R^i \omega_i + \begin{bmatrix} 0 & 0 & \ddot{\theta}_{i+1} \end{bmatrix}^T$$

- Converting from matrix to vector form gives the angular acceleration vector

Vector form (Revolute Joint)

$${}^{i+1}\dot{\omega}_{i+1} = {}^{i+1}R^i \dot{\omega}_i + {}^{i+1}R^i \omega_i \times \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_{i+1} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \ddot{\theta}_{i+1} \end{bmatrix}$$

- If joint $i+1$ is prismatic, the angular terms are zero ($\dot{\theta}_{i+1} = \ddot{\theta}_{i+1} = 0$) and the above equation simplifies to:

Matrix form (Prismatic Joint)

$${}^{i+1}\dot{\omega}_{i+1} = {}^{i+1}R^i \dot{\omega}_i$$



Manipulator Dynamics – Newton Euler Equations

The Linear Acceleration of the Center of Mass

$$\dot{v}_c$$



Dynamics - Newton-Euler Equations

- To solve the Newton and Euler equations, we'll need to develop mathematical terms for:

\dot{v}_c – The linear acceleration of the center of mass

$\dot{\omega}$ – The angular acceleration

cI – The Inertia tensor (moment of inertia)

F - The sum of all the forces applied on the center of mass

N - The sum of all the moments applied on the center of mass

$$F = m\dot{v}_c$$

$$N = {}^cI\dot{\omega} + \omega \times {}^cI\omega$$



Propagation of Acceleration - Linear Simultaneous Linear and Rotational Velocity

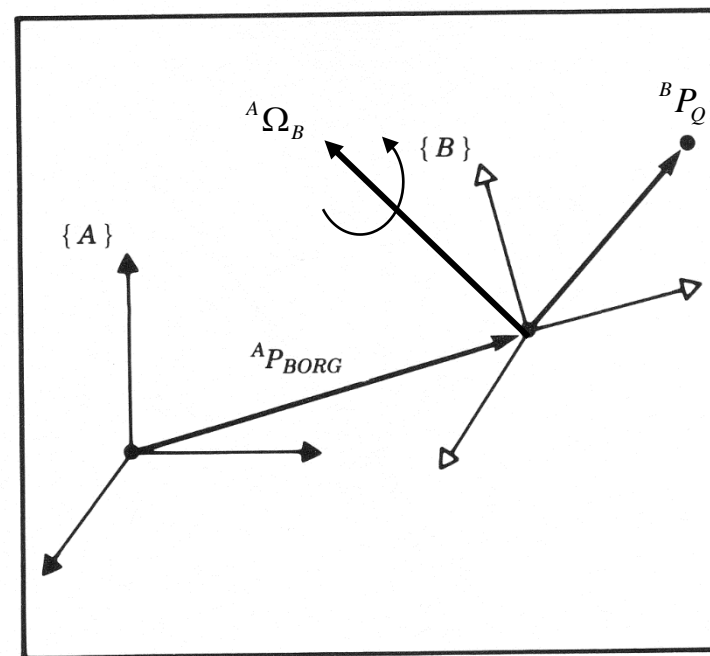
$${}^A V_Q = f({}^B P_Q, {}^B V_Q, {}^A V_{BORG}, {}^A \Omega_B, {}^A_B R)$$

- Vector Form

$${}^A V_Q = {}^A V_{BORG} + {}^A_B R {}^B V_Q + {}^A \Omega_B \times {}^A_B R {}^B P_Q$$

- Matrix Form

$${}^A V_Q = {}^A V_{BORG} + {}^A_B R {}^B V_Q + {}^A_B \dot{R}_\Omega \left({}^A_B R {}^B P_Q \right)$$





Deriving Linear Acceleration – Matrix Approach

$$\dot{v}_c$$



Propagation of Acceleration – Linear Matrix Approach

- To derive a general formula for the linear acceleration, we will differentiate the linear velocity. However, instead of differentiating the **recursive equation** like we did for the angular acceleration derivation, we'll begin at a slightly earlier step. Recall the three-part expression:

$${}^A V_Q = {}^A \dot{R}_\Omega \left({}^B R^B P_{Q \text{ org}} \right) + {}^A V_{B \text{ org}} + {}^A R^B V_Q$$

- Re-assigning the link frames ($A \rightarrow 0 \quad B \rightarrow i \quad Q \rightarrow i+1$), we find

$${}^0 V_{i+1} = {}^0 \dot{R}_\Omega \left({}^i R^i P_{i+1} \right) + {}^0 V_i + {}^0 R^i V_{i+1}$$

- Differentiating using the chain rule gives:

$${}^0 \dot{V}_{i+1} = {}^0 \ddot{R}_\Omega {}^i R^i P_{i+1} + {}^0 \dot{R}_\Omega {}^0 \dot{R}_\Omega {}^i R^i P_{i+1} + {}^0 \dot{R}_\Omega {}^0 R^i \dot{P}_{i+1} + {}^0 \dot{V}_i + {}^0 \dot{R}_\Omega {}^0 R^i V_{i+1} + {}^0 R^i \dot{V}_{i+1}$$



Propagation of Acceleration – Linear Matrix Approach

$${}^0\dot{V}_{i+1} = {}^0\ddot{R}_{\Omega i} {}^0R^i P_{i+1} + {}^0\dot{R}_{\Omega i} {}^0\dot{R}_{\Omega i} {}^0R^i P_{i+1} + \boxed{{}^0\dot{R}_{\Omega i} {}^0R^i \dot{P}_{i+1}} + {}^0\dot{V}_i + \boxed{{}^0\dot{R}_{\Omega i} {}^0R^i V_{i+1}} + {}^0R^i \dot{V}_{i+1}$$

- Combining the two like terms, we find:

$${}^0\dot{V}_{i+1} = {}^0\ddot{R}_{\Omega i} {}^0R^i P_{i+1} + {}^0\dot{R}_{\Omega i} {}^0\dot{R}_{\Omega i} {}^0R^i P_{i+1} + {}^0\dot{V}_i + \boxed{2{}^0\dot{R}_{\Omega i} {}^0R^i V_{i+1}} + {}^0R^i \dot{V}_{i+1}$$

- Pre-multiplying both sides of the equation by gives: $\boxed{{}^{i+1}_0 R}$

$$\boxed{{}^{i+1}_0 R} {}^0\dot{V}_{i+1} = \boxed{{}^{i+1}_0 R} {}^0\ddot{R}_{\Omega i} {}^0R^i P_{i+1} + \boxed{{}^{i+1}_0 R} {}^0\dot{R}_{\Omega i} {}^0\dot{R}_{\Omega i} {}^0R^i P_{i+1} + \boxed{{}^{i+1}_0 R} {}^0\dot{V}_i + \boxed{2{}^{i+1}_0 R} {}^0\dot{R}_{\Omega i} {}^0R^i V_{i+1} + \boxed{{}^{i+1}_0 R} {}^0R^i \dot{V}_{i+1}$$

- Expanding terms gives

$${}^{i+1}_0 R {}^0\dot{V}_{i+1} = \boxed{{}^{i+1}_i R} {}^i {}^0\ddot{R}_{\Omega i} {}^0R^i P_{i+1} + \boxed{{}^{i+1}_i R} {}^i {}^0\dot{R}_{\Omega i} {}^0\dot{R}_{\Omega i} {}^0R^i P_{i+1} + {}^{i+1}_0 R {}^0\dot{V}_i + \boxed{2{}^{i+1}_i R} {}^i {}^0\dot{R}_{\Omega i} {}^0R^i V_{i+1} + {}^{i+1}_0 R {}^0R^i \dot{V}_{i+1}$$



Propagation of Acceleration – Linear Matrix Approach

$${}^{i+1}_0 R^0 \dot{V}_{i+1} = {}^{i+1}_i R {}^i_0 R^0 \ddot{R}_{\Omega i} {}^i R^i P_{i+1} + {}^{i+1}_i R {}^i_0 R^0 \dot{R}_{\Omega i} {}^i \dot{R}_{\Omega i} {}^i R^i P_{i+1} + {}^{i+1}_0 R^0 \dot{V}_i + 2 {}^{i+1}_i R {}^i_0 R^0 \dot{R}_{\Omega i} {}^i R^i V_{i+1} + {}^{i+1}_0 R^0 R^i \dot{V}_{i+1}$$

- Simplifying the previous equation using (Note:

$${}^s_t R^A \dot{R}_{\Omega t} {}^s R^T = {}^s_t R^0 A \Omega_B$$

$${}^i_0 R^0 \ddot{R}_{\Omega i} {}^i R^i = {}^i_0 R^0 \ddot{R}_{\Omega 0} {}^i R^T = {}^i_0 R^0 \dot{\Omega}_i = {}^i_0 R \dot{\omega}_i = {}^i \dot{\omega}_i$$

$${}^i_0 R^0 \dot{R}_{\Omega i} {}^i \dot{R}_{\Omega i} {}^i R^i = {}^i \omega_i \times {}^i \omega_i$$

$${}^{i+1}_0 R^0 \dot{V}_i = {}^{i+1}_i R {}^i_0 R^0 \dot{V}_i = {}^{i+1}_i R^i \dot{v}_i$$

$${}^i_0 R^0 \dot{R}_{\Omega i} {}^i R^i = {}^i_0 R^0 \omega_i = {}^i \omega_i$$

- we have

$${}^{i+1}_0 \dot{v}_{i+1} = {}^{i+1}_i R \left[{}^i \dot{\omega}_i \times {}^i P_{i+1} + {}^i \omega_i \times {}^i \omega_i \times {}^i P_{i+1} + {}^i \dot{v}_i \right] + 2 {}^i \omega_i \times {}^{i+1}_i v_{i+1} + {}^{i+1}_0 \dot{v}_{i+1}$$



Propagation of Acceleration – Linear Matrix Approach

$${}^{i+1}\dot{v}_{i+1} = {}^{i+1}R \left[{}^i\dot{\omega}_i \times {}^iP_{i+1} + {}^i\omega_i \times {}^i\omega_i \times {}^iP_{i+1} + {}^i\dot{v}_i \right] + 2{}^{i+1}\omega_{i+1} \times {}^{i+1}v_{i+1} + {}^{i+1}\dot{v}_{i+1}$$

- This equation can be written equivalently as:

General form

$${}^{i+1}\dot{v}_{i+1} = {}^{i+1}R \left[{}^i\dot{\omega}_i \times {}^iP_{i+1} + {}^i\omega_i \times {}^i\omega_i \times {}^iP_{i+1} + {}^i\dot{v}_i \right] + 2{}^{i+1}\omega_{i+1} \times \begin{bmatrix} 0 \\ 0 \\ \dot{d}_{i+1} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \ddot{d}_{i+1} \end{bmatrix}$$

Handwritten notes:

Frame $i \parallel$ Frame $cm \Rightarrow [I]$

$R = {}^i_{cm}R$

${}^i_{cm}v_{i+1} = v_{i+1}$

- If joint $i+1$ is revolute joint, the linear terms are zero and the above equation simplifies to:

Revolute Joint

$$\rightarrow {}^{i+1}\dot{v}_{i+1} = {}^{i+1}R \left[{}^i\dot{\omega}_i \times {}^iP_{i+1} + {}^i\omega_i \times {}^i\omega_i \times {}^iP_{i+1} + {}^i\dot{v}_i \right] + 0 + 0$$

Handwritten notes:

For Revolute joints

${}^i_{cm}v_{i+1} = {}^i_{cm}R \left[{}^i\dot{\omega}_i \times {}^iP_{i+1} + {}^i\omega_i \times {}^i\omega_i \times {}^iP_{i+1} + {}^i\dot{v}_i \right]$

$i+1 \rightarrow i_{cm}$



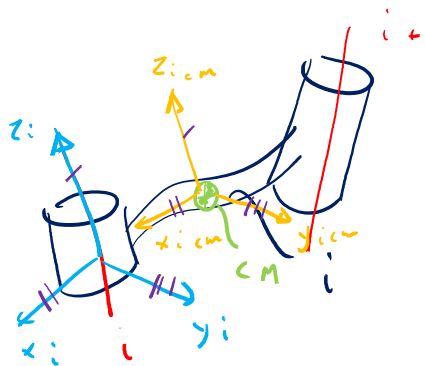
Propagation of Acceleration - Linear

- From the general equation, we can also get the solution for the **acceleration of the center of mass** for link i . Appropriate frame substitution and elimination of prismatic terms gives we find:

$$\rightarrow \boxed{{}^i \dot{v}_{ci} = \left({}^i \dot{\omega}_i \times {}^i P_{ci} + {}^i \omega_i \times ({}^i \omega_i \times {}^i P_{ci}) + {}^i \dot{v}_i \right)}$$

\uparrow
 ${}^i_{cm} R = [I] \quad - \quad \begin{matrix} \text{ } \end{matrix} \parallel \begin{matrix} \text{ } \end{matrix}$

- Frame $\{C_i\}$ attached to each link with its origin located at the Center of mass of the link, and with the same orientation as the link frame $\{i\}$





Deriving Linear Acceleration – Vector Approach

$$\dot{\mathbf{v}}_c$$



Propagation of Acceleration – Linear Vector Approach

ORG {B} ACC WRT {A}

$$\rightarrow {}^A \dot{V}_Q = {}^A R^B \dot{V}_Q + 2 {}^A \Omega_B \times {}^A R^B V_Q + {}^A \dot{\Omega}_B \times {}^A R^B P_Q + {}^A \Omega_B \times ({}^A \Omega_B \times {}^A R^B P_Q)$$

$$\boxed{{}^A a_Q = {}^A \dot{V}_Q} = \boxed{{}^A \dot{V}_{BORG}} + \boxed{{}^A R^B \dot{V}_Q} + \boxed{2 {}^A \Omega_B \times {}^A R^B V_Q} + \boxed{{}^A \dot{\Omega}_B \times {}^A R^B P_Q} + \boxed{{}^A \Omega_B \times ({}^A \Omega_B \times {}^A R^B P_Q)}$$

${}^A a_Q = {}^A \dot{V}_Q =$ Absolute acc. of point Q expressed in frame A

${}^A \dot{V}_{BORG}$ - Absolute acc. of the origine of frame B expressed in frame A

${}^A R^B \dot{V}_Q$ - Acc. of point Q with respect to frame B and expressed in frame A

$2 {}^A \Omega_B \times {}^A R^B V_Q$ (Coriolis Acc) - Combined effect of point Q moving with a velocity ${}^B V_Q$ relative to frame B and the rotation of frame B wrt frame A

${}^A \dot{\Omega}_B \times {}^A R^B P_Q$ - Angular Acc. effect caused by the rotation of frame (B) wrt frame (A)

${}^A \Omega_B \times ({}^A \Omega_B \times {}^A R^B P_Q)$ - Centrifugal Acc - Angular velocity effect caused by the rotation of frame {B} wrt frame {A}

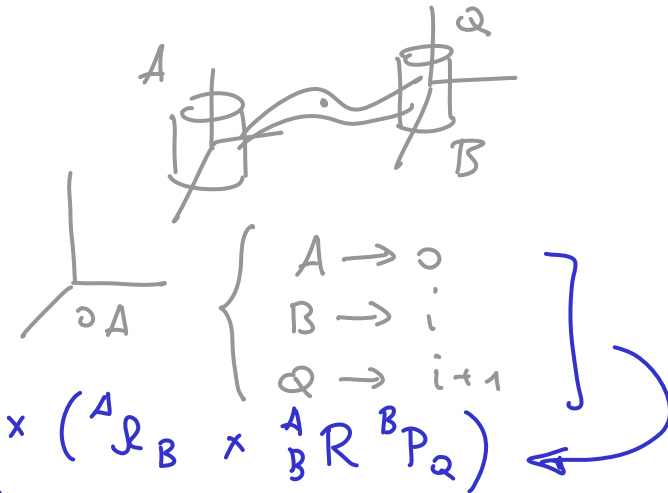


Propagation of Acceleration – Linear Vector Approach

Special case ${}^B P_Q$ is constant

$${}^B \dot{P}_Q = {}^B V_Q = 0$$

$${}^B \ddot{P}_Q = {}^B \dot{V}_Q = {}^B a_Q = 0$$



$${}^A a_Q = {}^A \dot{V}_Q = {}^A \dot{V}_{B \text{ or } G} + {}^A \dot{\Omega}_B \times {}^A {}_B R {}^B P_Q + {}^A \Omega_B \times ({}^A \Omega_B \times {}^A {}_B R {}^B P_Q)$$

$${}^o \dot{V}_{i+1} = {}^o \dot{V}_i + {}^o \dot{\Omega}_i \times {}^o {}_i R {}^i P_{i+1} + {}^o \Omega_i \times ({}^o \Omega_i \times {}^o {}_i R {}^i P_{i+1})$$

Multiply by ${}^{i+1} {}_o R$

$$\underbrace{{}^{i+1} {}_o R {}^o \dot{V}_{i+1}}_{{}^{i+1} \dot{V}_{i+1}} = \underbrace{{}^{i+1} {}_o R {}^o \dot{V}_i}_{{}^{i+1} V_i} + \underbrace{{}^{i+1} {}_o R ({}^o \dot{\Omega}_i \times {}^o {}_i R {}^i P_{i+1})}_{{}^{i+1} {}_i R ({}^i {}_o \dot{\Omega}_i \times {}^i P_{i+1})} + \underbrace{{}^{i+1} {}_o R ({}^o \Omega_i \times ({}^o \Omega_i \times {}^o {}_i R {}^i P_{i+1}))}_{{}^{i+1} {}_i R [{}^i {}_o \Omega_i \times ({}^i {}_o \Omega_i \times {}^i P_{i+1})]}$$

$${}^{i+1} \dot{V}_{i+1} = {}^{i+1} {}_i R [{}^i \dot{V}_i + {}^i \omega_i \times {}^i P_{i+1} + {}^i \omega_i \times ({}^i \omega_i \times {}^i P_{i+1})]$$



Manipulator Dynamics – Newton Euler Equations

Forces & Torques

Applied at the Center of Mass F N

Applied at the Joints ${}^i f_i$ ${}^i n_i$



Dynamics - Newton-Euler Equations

- To solve the Newton and Euler equations, we'll need to develop mathematical terms for:

\dot{v}_c – The linear acceleration of the center of mass

$\dot{\omega}$ – The angular acceleration

cI – The Inertia tensor (moment of inertia)

F – The sum of all the forces applied on the center of mass

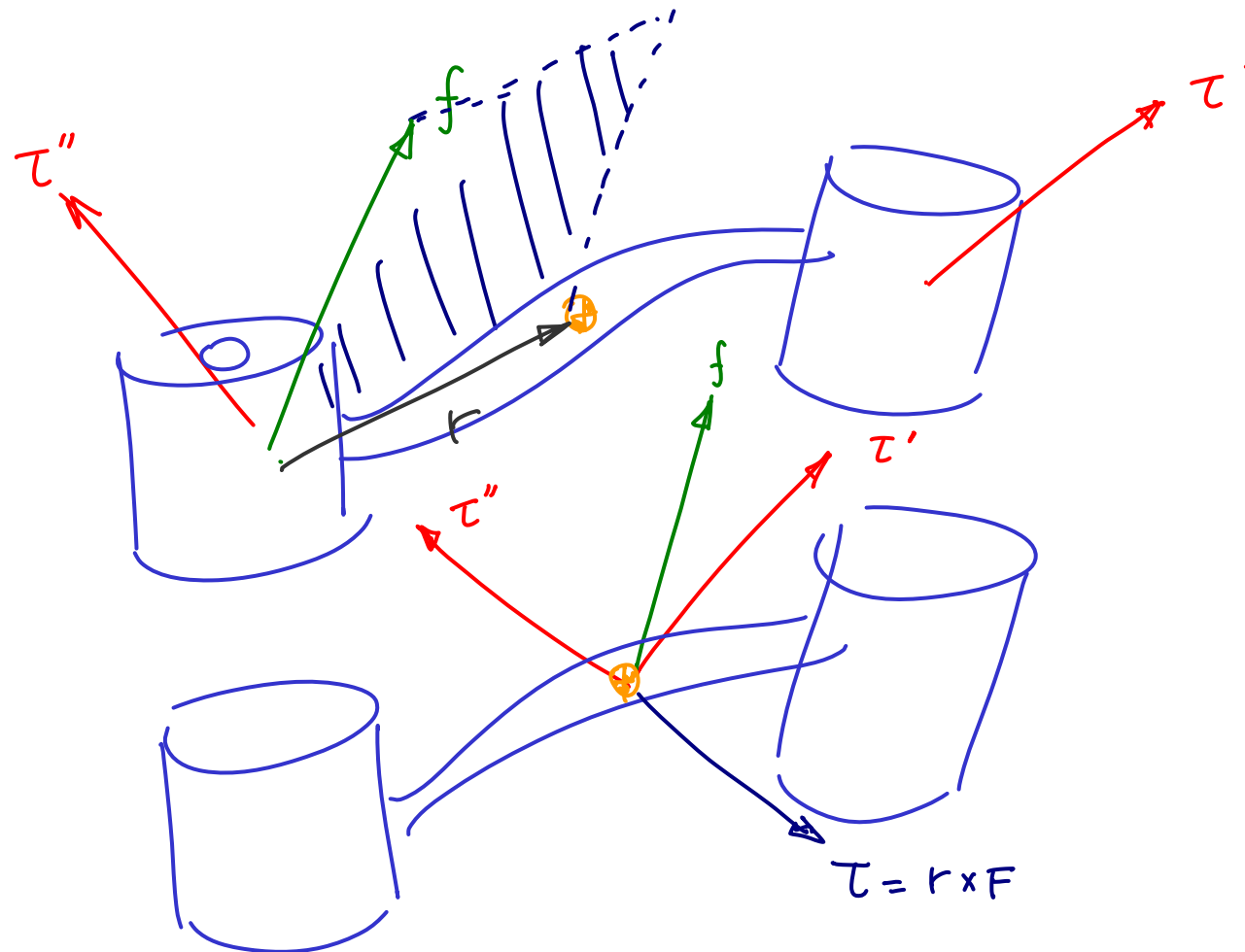
N – The sum of all the moments applied on the center of mass

$$F = m\dot{v}_c$$

$$N = {}^cI\dot{\omega} + \omega \times {}^cI\omega$$



Expression of Force & Torque With Respect to the CM





Iterative Newton-Euler Equations

- The Newton and Euler equations are re-written for the forces and moments at each link:

$${}^i F_i = m_i {}^i \dot{v}_{ci}$$

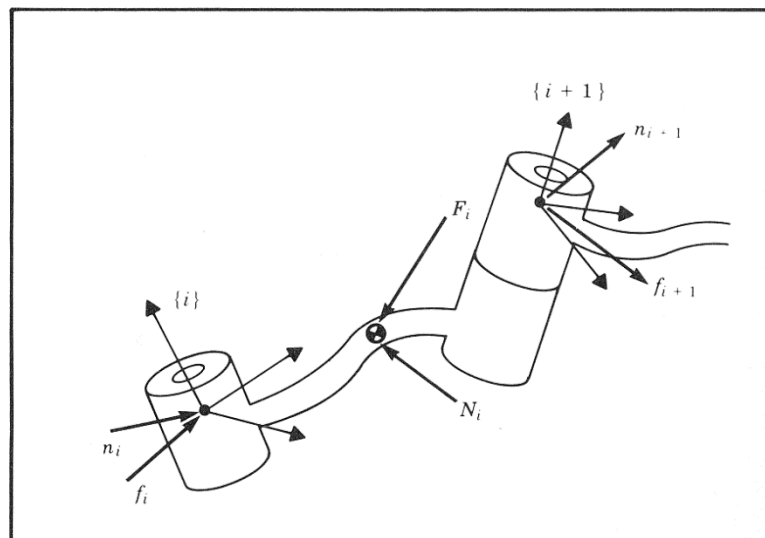
$${}^i N_i = {}^{ci} I_i {}^i \dot{\omega}_i + {}^i \omega_i \times {}^{ci} I_i {}^i \omega_i$$

- Where $\{C_i\}$ is a frame who has its origin at the link's center of mass and has the same orientation as the link frame $\{i\}$.



Sum of Forces and Moment on a Link

- In addition to calculating the **forces and torques arising from link accelerations**, we also need to account for how they **affect the neighboring links** as well as the end effectors interactions with the environment.



- Balancing the forces shown in the above figure, we can find the total force and torque on each link.



Sum of Forces and Moment on a Link

$${}^iF_i = m\dot{v}_{ci}$$

$${}^iN_i = {}^{ci}I\dot{\omega}_i + \omega_i \times {}^cI\omega_i$$

$${}^iF_i = {}^if_i - {}^if_{i+1} = {}^if_i - {}^iR^{i+1}f_{i+1}$$

$${}^iN_i = {}^in_i - {}^in_{i+1} + (-{}^iP_{ci}) \times {}^if_i - (-{}^iP_{ci} + {}^iP_{i+1}) \times {}^if_{i+1}$$

$${}^iN_i = {}^in_i - {}^iR^{i+1}n_{i+1} - \boxed{{}^iP_{ci} \times {}^if_i + {}^iP_{ci} \times {}^iR^{i+1}f_{i+1}} - {}^iP_{i+1} \times {}^iR^{i+1}f_{i+1}$$

$$\boxed{{}^iP_{ci} \times ({}^if_i - {}^iR^{i+1}f_{i+1})}$$

$$\boxed{{}^iP_{ci} \times {}^iF_i}$$

$${}^iN_i = {}^in_i - {}^iR^{i+1}n_{i+1} - \boxed{{}^iP_{ci} \times {}^iF_i} - {}^iP_{i+1} \times {}^iR^{i+1}f_{i+1}$$



Sum of Forces and Moment on a Link

- Rearranging the force / torque equations so that they appear as iterative relationship from higher number neighbor to lower number neighbor. The total force and torque on each link.

$$\begin{aligned} {}^i f_i &= {}^i R^{i+1} f_{i+1} + {}^i F_i \\ {}^i n_i &= {}^i N_i + {}^i R^{i+1} n_{i+1} + {}^i P_{ci} \times {}^i F_i + {}^i P_{i+1} \times {}^i R^{i+1} f_{i+1} \end{aligned}$$

- Compare with the same equation for the static conditions

$$\begin{aligned} {}^i f_i &= {}^i R^{i+1} f_{i+1} \\ {}^i n_i &= {}^i R^{i+1} n_{i+1} + {}^i P_{i+1} \times {}^i f_{i+1} \end{aligned}$$



Sum of Forces and Moment on a Link

- The joint torque is simply the component of torque that projects onto the joint axis (Z axis by definition)

$$\tau_i = {}^i n_i \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$



Sum of Forces and Moment on a Link

- For a robot moving in free space

$${}^{N+1}f_{N+1} = 0$$

$${}^{N+1}n_{N+1} = 0$$

- If the robot is contacting the environment, the forces/ torques due to this contact may be included in the equations

$${}^{N+1}f_{N+1} \neq 0$$

$${}^{N+1}n_{N+1} \neq 0$$



Manipulator Dynamics – Newton Euler Equations

Equation Formulation Procedure



Iterative Newton Euler Equations

Steps of the Algorithm

- (1) Outward Iterations

- Starting With velocities and accelerations of the base

$${}^0\omega_0 = 0, {}^0\dot{\omega}_0 = 0, {}^0v_0 = 0, {}^0\dot{v}_0 = +g\hat{z}$$

- Calculate velocities accelerations, along with forces and torques (at the CM)

$$\omega, \dot{\omega}, \dot{v}, \dot{v}_{CM}, F, N$$

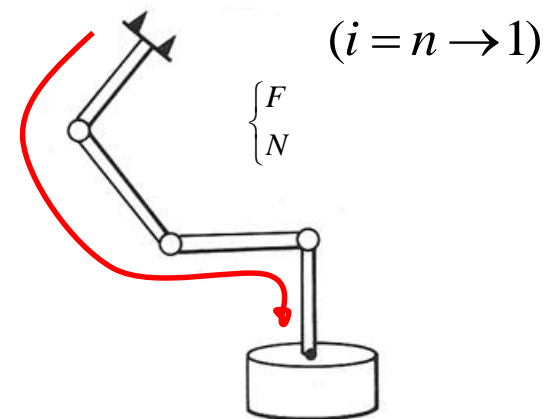
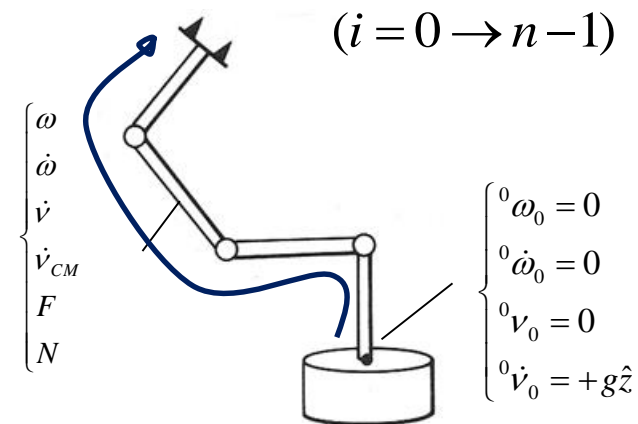
- (2) Inward Iteration ($i = n \rightarrow 1$)

- Starting with forces and torques (at the CM)

$$F, N$$

- Calculate forces and torques at the joints

$$f, n$$





Iterative Newton-Euler Equations - Solution Procedure

Phase 1: Outward Iteration

Outward Iteration: $i : 0 \rightarrow 5$

- Calculate the link velocities and accelerations iteratively from the robot's base to the end effector

$${}^{i+1}\omega_{i+1} = {}^iR^{i+1} {}^i\omega_i + \dot{\theta}_{i+1} {}^{i+1}\hat{Z}_{i+1}$$

$${}^{i+1}\dot{\omega}_{i+1} = {}^iR^{i+1} {}^i\dot{\omega}_i + {}^iR^{i+1} {}^i\omega_i \times \dot{\theta}_{i+1} {}^{i+1}\hat{Z}_{i+1} + \ddot{\theta}_{i+1} {}^{i+1}\hat{Z}_{i+1}$$

$${}^{i+1}\dot{v}_{i+1} = {}^iR^{i+1} ({}^i\dot{\omega}_i \times {}^iP_{i+1} + {}^i\omega_i \times ({}^i\omega_i \times {}^iP_{i+1}) + {}^i\dot{v}_i)$$

$${}^{i+1}\dot{v}_{Ci+1} = {}^{i+1}\dot{\omega}_{i+1} \times {}^{i+1}P_{Ci+1} + {}^{i+1}\omega_{i+1} \times ({}^{i+1}\omega_{i+1} \times {}^{i+1}P_{Ci+1}) + {}^{i+1}\dot{v}_{i+1}$$

- Calculate the force and torques applied on the CM of each link using the Newton and Euler equations

$${}^{i+1}F_{i+1} = m_{i+1} {}^{i+1}\dot{v}_{Ci+1}$$

$${}^{i+1}N_{i+1} = {}^C {}^{i+1}I_{i+1} {}^{i+1}\dot{\omega}_{i+1} + {}^{i+1}\omega_{i+1} \times {}^C {}^{i+1}I_{i+1} {}^{i+1}\omega_{i+1}$$



Iterative Newton-Euler Equations - Solution Procedure

Phase 2: Inward Iteration

Inward Iteration: $i : 6 \rightarrow 1$

- Use the forces and torques generated at the joints starting with forces and torques generating by interacting with the environment (that is, tools, work stations, parts etc.) at the end effector all the way the robot's base.

$${}^i f_i = {}^i R^{i+1} {}^{i+1} f_{i+1} + {}^i F_i$$

$${}^i n_i = {}^i N_i + {}_{i+1} {}^i R^{i+1} n_{i+1} + {}^i P_{Ci} \times {}^i F_i + {}^i P_{i+1} \times {}_{i+1} {}^i R^{i+1} f_{i+1}$$

$$\tau_i = {}^{i+1} n_{i+1}^T {}^i \hat{Z}_i$$



Iterative Newton-Euler Equations - Solution Procedure

- **Error Checking** - Check the units of each term in the resulting equations
- **Gravity Effect** - The effect of gravity can be included by setting ${}^0\dot{v}_0 = g$. This is the equivalent to saying that the base of the robot is accelerating upward at 1 g. The result of this accelerating is the same as accelerating all the links individually as gravity does.



Manipulator Dynamics – Newton Euler Equations

The Inertia Tensor (Moment of Inertia)

$${}^c I$$



Dynamics - Newton-Euler Equations

- To solve the Newton and Euler equations, we'll need to develop mathematical terms for:

\dot{v}_c – The linear acceleration of the center of mass

$\dot{\omega}$ – The angular acceleration

cI – The Inertia tensor (moment of inertia)

F – The sum of all the forces applied on the center of mass

N – The sum of all the moments applied on the center of mass

$$F = m\dot{v}_c$$

$$N = {}^cI\dot{\omega} + \omega \times {}^cI\omega$$