Week 9 – Manipulator Dynamics

ELEC0129 Introduction to Robotics

Dr. Chow Yin Lai

Email: uceecyl@ucl.ac.uk

Schedule

Legend:
A: MPEB 6th Floor Lab
B: PC Lab

Legend:
C: 11am-1pm
D: 11.30am-1pm

Week	Recorded, uploaded by Friday of previous week	F2F Workshop, Mondays 11am-1pm and/or Wednesdays 9am-11am	Virtual Workshop, Mondays (C/D) and Wednesdays 9am-11am
1	(Scenario Week)		
2	Lec: Intro; Spatial description	A: Workshop: Offline programming	C: Workshop: Offline programming
3	Lec & Tut: Spatial description	A: Workshop: Build robot	D: Workshop: Build robot
4	Lec & Tut: Forward kinematics	A: Workshop: Forward kinematics	D: Workshop: Forward kinematics
5	Lec & Tut: Inverse kinematics	B: Workshop: Offline programming	C: Workshop: Offline programming
RW	(Reading Week)		
6	(Scenario Week)		
7	Lec & Tut: Jacobians	A: Workshop: Inverse kinematics	D: Workshop: Inverse kinematics
8	Lec: Trajectory Planning	A: Workshop: Trajectory planning	D: Workshop: Trajectory planning
9	Lec & Tut: Dynamics	A: Workshop: Trajectory planning	C: Workshop: Trajectory planning
10	Lec & Tut: Control	A: Workshop: Pick-and-place demo	C: Workshop: Pick-and-place demo

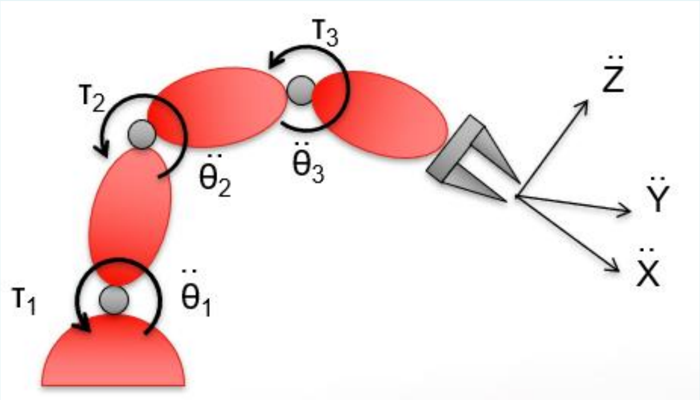
Content

Content

- Introduction
- Newton-Euler Formulation
- Inclusion of Friction Force
- Manipulator Dynamics in Cartesian Space
- Derivation of Inertia Tensor
- Matlab Simulation
- Tutorial Questions

Introduction

- Manipulator Dynamics:
  - How much torque is needed to accelerate the manipulator from rest to constant velocity, and then back to stop?
  - Dynamics also provide us a model (equations of motions) for simulation and control design purpose.



Manipulator's Dynamic Equations (1)

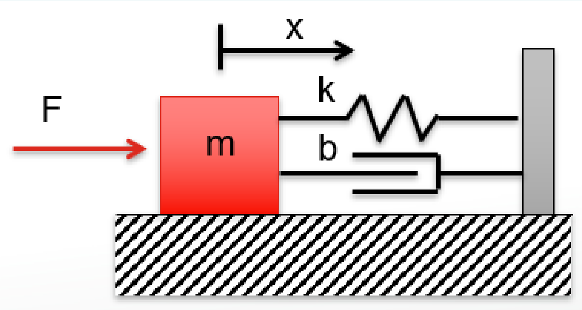
- Before we go into details of how to derive the manipulator's joint space dynamic equations, let's first have a glimpse of how the equations look like:

$$\boxed{M(q)\ddot{q} + V(q,\dot{q}) + G(q) = \tau}$$

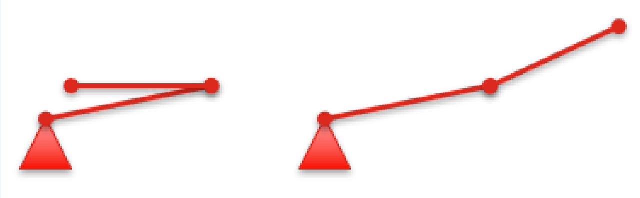
- A comparison with the well-known mass-spring-damper system:

$$\boxed{m\ddot{x} + b\dot{x} + kx = F}$$

  - There are some similarities.

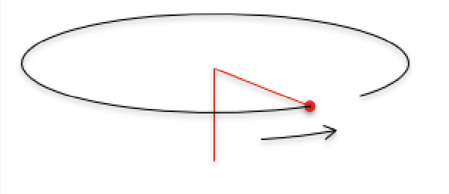


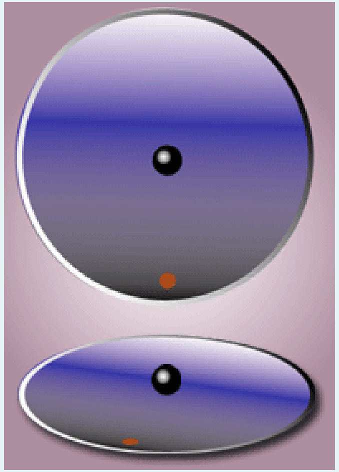
Manipulator's Dynamic Equations (2)

- $M(q)$ is the $n \times n$ mass matrix of the manipulator, which depends on the generalized joint coordinates q (angles / displacement).
  - For e.g. two link robot: 
  - The "perceived inertia" at joint 1 of the right configuration is larger than that of the left configuration.
  - The "perceived inertia" also depends on the mass contribution and length of the links.

Manipulator’s Dynamic Equations (3)

- $V(q, \dot{q})$ is an $n \times 1$ vector consisting of:
  - Centrifugal force: A ‘fictitious’ force acting away from axis of rotation. E.g. whirling a stone on a string.
  - Coriolis force: A ‘fictitious’ force that acts on objects that are in motion within a frame of reference that rotates with respect to an inertial frame.
    - (In inertial frame: a body with zero net force acting upon it does not accelerate).





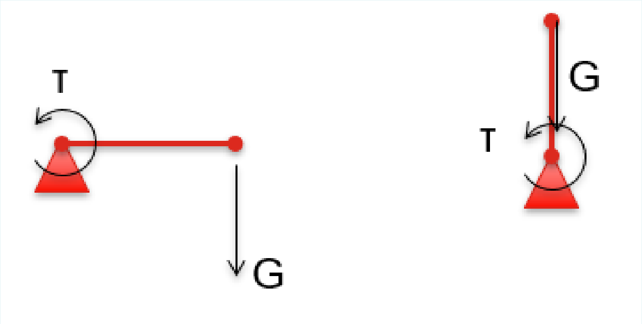
Manipulator's Dynamic Equations (4)

- $V(q,\dot{q})$ depends on the generalized joint coordinates q as well as the joint velocities $\dot{q}$.
  - It is zero if velocities = 0.

- Also, $V(q,\dot{q})$ can be derived from $M(q)$.
  - It is also zero if $M(q)$ is a constant matrix

Manipulator's Dynamic Equations (5)

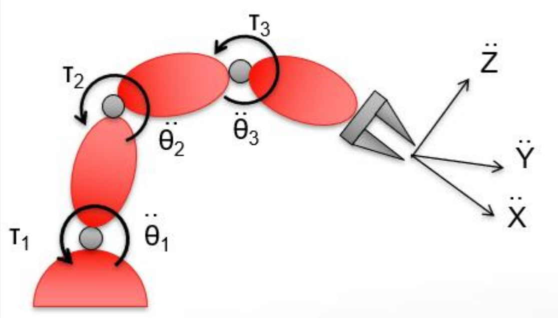
- $G(q)$ is the $n \times 1$ vector of gravity terms.
  - It is dependent on the joint coordinates / configuration of the robot.

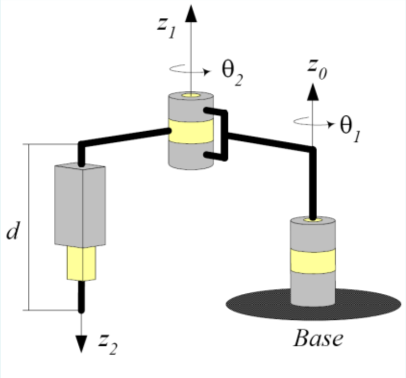


  - In the left figure, the joint torque is nonzero, and in the right figure, the joint torque is zero.

Manipulator's Dynamic Equations (6)

- Finally, τ is the generalized forces (force or torque) at each joints.
  - E.g. for 3R robots, τ means torque – torque – torque.
  - For RRP robots, τ means torque – torque– force.





Content

Basic Idea – Newton's Second Law

- In an inertial frame of reference,
- The vector sum (F) of the forces (f_i) on an object…
- is equal to the mass (m) of that object multiplied by the acceleration (a) of the object:

$$\boxed{F = \sum_i f_i = ma}$$

Basic Idea – Euler's Equation

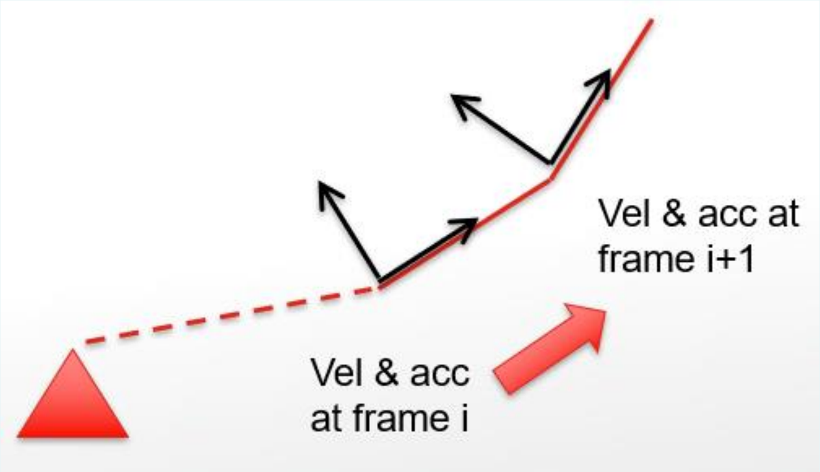
- There is also a similar equation for rotary motions:
- A rigid body is rotating with angular velocity ω and with angular acceleration $\dot{\omega}$.
- The moment N which must be acting on the body to cause this motion is given by:

$$\boxed{N = {}^{c}I\dot{\omega} + \omega \times {}^{c}I\omega}$$

  - where ${}^{c}I$ is the inertia tensor of the body written in frame $\{C\}$ whose origin is located at the centre of mass.
  - The first two terms are very similar to $F = ma$. You may think of ${}^{c}I$ as the "mass" but for rotary motion.

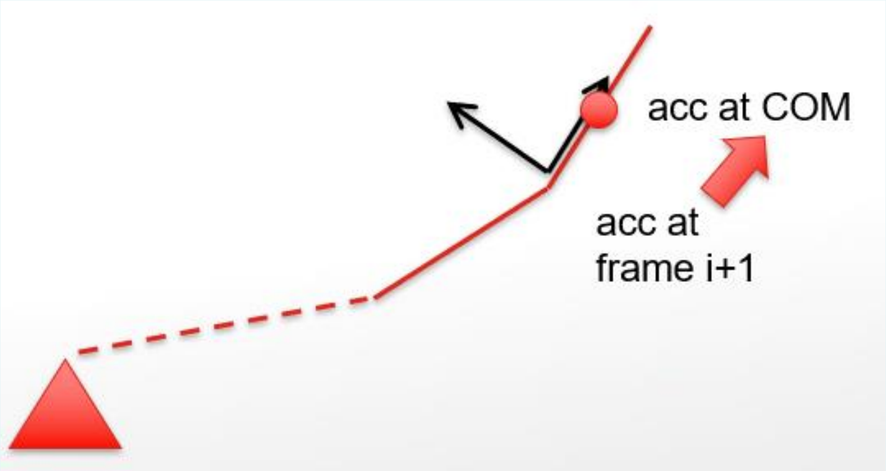
Basic Idea – Newton Euler Formula (1)

- Firstly, similar to velocity propagation which you learnt last week, acceleration can also be propagated from lower frame to upper frame.



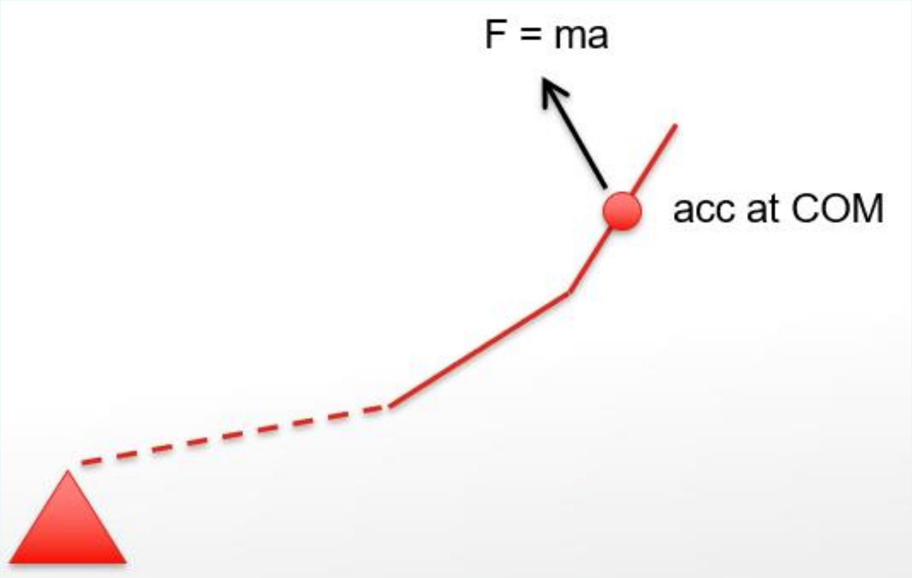
Basic Idea – Newton Euler Formula (2)

- Next, the acceleration at frame $i + 1$ can be propagated to the centre of mass.



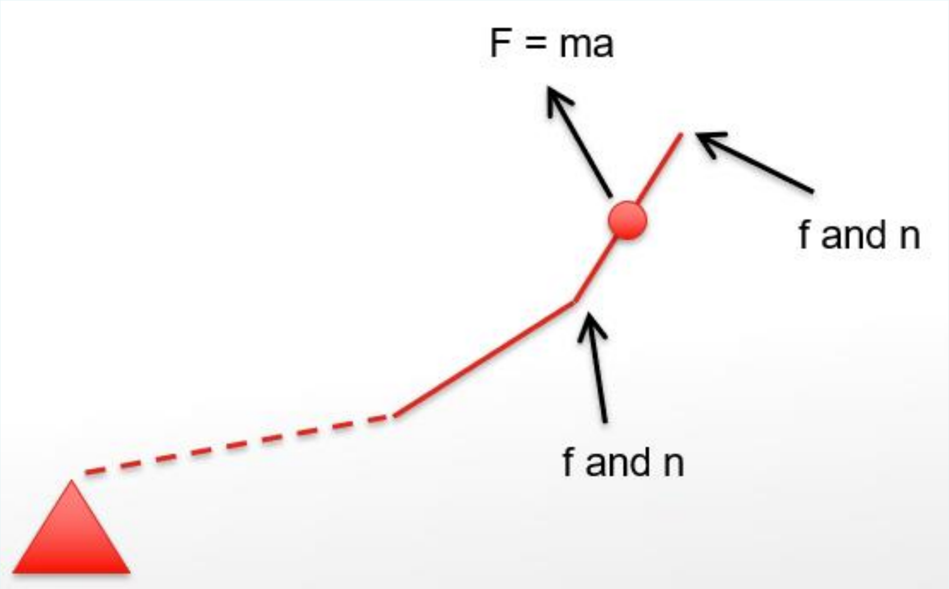
Basic Idea – Newton Euler Formula (3)

- Once the acceleration at centre of mass is known, then we also know the force acting on the centre of mass since F = ma.



Basic Idea – Newton Euler Formula (4)

- But what “creates” F? The forces / torques caused by the motors at both ends of the link, and contact force at the end-effector.



Summary – Newton Euler Formula (1)

- The Newton-Euler iterative formula is:
- Outward iteration:
  - Start with $\boxed{{}^{0}\omega_0 = 0,\ {}^{0}\dot{\omega}_0 = 0,\ {}^{0}\dot{v}_0 = \text{depends}}$
  - Calculate velocities and accelerations of frames:

$$
\boxed{
\begin{aligned}
&{}^{i+1}\omega_{i+1} = \left({}^{i+1}_{i}R \cdot {}^{i}\omega_i\right) + \left(\dot{\theta}_{i+1} \cdot {}^{i+1}\hat{Z}_{i+1}\right) \\
&{}^{i+1}\dot{\omega}_{i+1} = \left({}^{i+1}_{i}R \cdot {}^{i}\dot{\omega}_i\right) + \left(\left({}^{i+1}_{i}R \cdot {}^{i}\omega_i\right) \times \left(\dot{\theta}_{i+1} \cdot {}^{i+1}\hat{Z}_{i+1}\right)\right) + \left(\ddot{\theta}_{i+1} \cdot {}^{i+1}\hat{Z}_{i+1}\right) \\
&{}^{i+1}\dot{v}_{i+1} = \left({}^{i+1}_{i}R \cdot {}^{i}\dot{v}_i\right) + \left(2\,{}^{i+1}\omega_{i+1} \times \left(\dot{d}_{i+1} \cdot {}^{i+1}\hat{Z}_{i+1}\right)\right) + \left(\ddot{d}_{i+1} \cdot {}^{i+1}\hat{Z}_{i+1}\right) \\
&\qquad\qquad + {}^{i+1}_{i}R\left(\left({}^{i}\dot{\omega}_i \times {}^{i}P_{i+1}\right) + \left({}^{i}\omega_i \times \left({}^{i}\omega_i \times {}^{i}P_{i+1}\right)\right)\right)
\end{aligned}
}
$$

Summary – Newton Euler Formula (2)

- Propagate accelerations from frames to centre of mass:

$$\boxed{{}^{i+1}\dot{v}_{ci+1} = \left({}^{i+1}\dot{v}_{i+1}\right) + \left({}^{i+1}\dot{\omega}_{i+1} \times {}^{i+1}P_{ci+1}\right) + \left({}^{i+1}\omega_{i+1} \times \left({}^{i+1}\omega_{i+1} \times {}^{i+1}P_{ci+1}\right)\right)}$$

- Calculate the force and moment at the centre of mass:

$$\boxed{\begin{aligned} &{}^{i+1}F_{i+1} = m_{i+1}\,{}^{i+1}\dot{v}_{ci+1} \\ &{}^{i+1}N_{i+1} = \left({}^{ci+1}I_{i+1} \cdot {}^{i+1}\dot{\omega}_{i+1}\right) + \left({}^{i+1}\omega_{i+1} \times \left({}^{ci+1}I_{i+1} \cdot {}^{i+1}\omega_{i+1}\right)\right) \end{aligned}}$$

- Do until the nth link.

Summary – Newton Euler Formula (3)

- Inward iteration:
  - Start with force and torque at robot tip ${}^{n}f_{n}$, ${}^{n}n_{n}$.
  - Calculate force and torque at the starting end of each link:

$$\boxed{\begin{gathered} {}^{i}f_{i} = \left({}^{i}_{i+1}R \cdot {}^{i+1}f_{i+1}\right) + {}^{i}F_{i} \\ {}^{i}n_{i} = \left({}^{i}_{i+1}R \cdot {}^{i+1}n_{i+1}\right) + \left({}^{i}P_{ci} \times {}^{i}F_{i}\right) + \left({}^{i}P_{i+1} \times \left({}^{i}_{i+1}R \cdot {}^{i+1}f_{i+1}\right)\right) + {}^{i}N_{i} \end{gathered}}$$

  - Do until $i = 1$.
  - Finally, extract the joint (motor) torques or forces:

$$\boxed{\begin{aligned} \tau_{i} &= {}^{i}n_{i}^{T} \cdot {}^{i}\hat{Z}_{i} \ \text{ if revolute} \\ \tau_{i} &= {}^{i}f_{i}^{T} \cdot {}^{i}\hat{Z}_{i} \ \text{ if prismatic} \end{aligned}}$$

Inclusion of Gravity Force

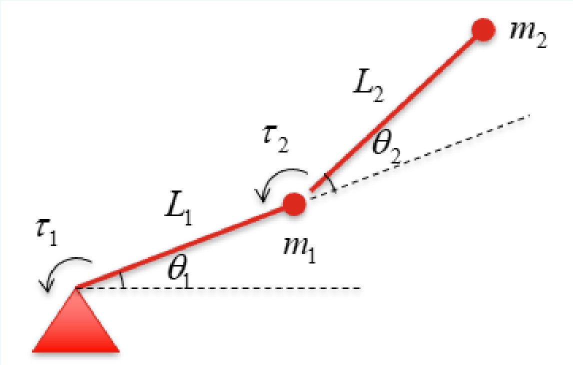
- The effect of gravity forces can be included by setting:

$$\boxed{{}^{0}\dot{v}_0 = G}$$

  - where G has the magnitude of gravity vector but points in the opposite direction.
- This can be interpreted as the base moving upwards with 1g acceleration.

Example – Outward Iteration (1)

- Two link robot, where the mass of each link is a point mass at the end of the link:



- The vectors that locate the center of mass for each link are:

$$\boxed{{}^{1}P_{c1} = L_1\hat{X}_1,\ {}^{2}P_{c2} = L_2\hat{X}_2}$$

- Because the mass of each link is point mass, the inertia tensor at the center of mass is zero:

$$\boxed{{}^{c1}I_1 = 0, {}^{c2}I_2 = 0}$$

Example – Outward Iteration (2)

- Furthermore, the rotation matrices between successive links are:

$$\boxed{{}^{i}_{i+1}R = \begin{bmatrix} c_{i+1} & -s_{i+1} & 0 \\ s_{i+1} & c_{i+1} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad {}^{i+1}_{i}R = \begin{bmatrix} c_{i+1} & s_{i+1} & 0 \\ -s_{i+1} & c_{i+1} & 0 \\ 0 & 0 & 1 \end{bmatrix}}$$

Example – Outward Iteration (3)

- Now we use the Iterative Newton-Euler algorithm.
- First, we start with:

$$\boxed{{}^{0}\omega_0 = 0,\ {}^{0}\dot{\omega}_0 = 0,\ {}^{0}\dot{v}_0 = g\hat{Y}_0}$$

- Then, the outward iterations for link 1 give the following frame velocities and accelerations:

$$\boxed{{}^{1}\omega_1 = \left({}^{1}_{0}R \cdot \underbrace{{}^{0}\omega_0}_{0}\right) + \left(\dot{\theta}_1 \cdot\ {}^{1}\hat{Z}_1\right) = \left(\dot{\theta}_1 \cdot\ {}^{1}\hat{Z}_1\right) = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix}}$$

Example – Outward Iteration (4)

$$ {}^{1}\dot{\omega}_1 = \left({}^{1}_{0}R \cdot \underbrace{{}^{0}\dot{\omega}_0}_{0}\right) + \left(\left({}^{1}_{0}R \cdot \underbrace{{}^{0}\omega_0}_{0}\right) \times \left(\dot{\theta}_1 \cdot {}^{1}\hat{Z}_1\right)\right) + \left(\ddot{\theta}_1 \cdot {}^{1}\hat{Z}_1\right) = \begin{bmatrix} 0 \\ 0 \\ \ddot{\theta}_1 \end{bmatrix} $$

$$ {}^{1}\dot{v}_1 = \left({}^{1}_{0}R \cdot {}^{0}\dot{v}_0\right) + \left(2\ {}^{1}\omega_1 \times \left(\underbrace{\dot{d}_1}_{0} \cdot {}^{1}\hat{Z}_1\right)\right) + \left(\underbrace{\ddot{d}_1}_{0} \cdot {}^{1}\hat{Z}_1\right) $$

$$ + {}^{1}_{0}R\left(\left(\underbrace{{}^{0}\dot{\omega}_0}_{0} \times {}^{0}P_1\right) + \left(\underbrace{{}^{0}\omega_0}_{0} \times \left({}^{0}\omega_0 \times {}^{0}P_1\right)\right)\right) $$

$$ = \left({}^{1}_{0}R \cdot {}^{0}\dot{v}_0\right) = \begin{bmatrix} c_1 & s_1 & 0 \\ -s_1 & c_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ g \\ 0 \end{bmatrix} = \begin{bmatrix} gs_1 \\ gc_1 \\ 0 \end{bmatrix} $$

Example – Outward Iteration (5)

- We then propagate the frame acceleration to the centre of mass:

$$\boxed{\begin{aligned} {}^{1}\dot{v}_{c1} &= \left({}^{1}\dot{v}_{1}\right) + \left({}^{1}\dot{\omega}_{1} \times {}^{1}P_{c1}\right) + \left({}^{1}\omega_{1} \times \left({}^{1}\omega_{1} \times {}^{1}P_{c1}\right)\right) \\ &= \begin{bmatrix} gs_1 \\ gc_1 \\ 0 \end{bmatrix} + \left(\begin{bmatrix} 0 \\ 0 \\ \ddot{\theta}_1 \end{bmatrix} \times \begin{bmatrix} L_1 \\ 0 \\ 0 \end{bmatrix}\right) + \left(\begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} \times \left(\begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} \times \begin{bmatrix} L_1 \\ 0 \\ 0 \end{bmatrix}\right)\right) \\ &= \begin{bmatrix} gs_1 \\ gc_1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ L_1\ddot{\theta}_1 \\ 0 \end{bmatrix} + \begin{bmatrix} -L_1\dot{\theta}_1^2 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} gs_1 - L_1\dot{\theta}_1^2 \\ gc_1 + L_1\ddot{\theta}_1 \\ 0 \end{bmatrix} \end{aligned}}$$

Example – Outward Iteration (6)

- With velocity and acceleration of centre of mass, total force and moment of the link can be calculated:

$$ {}^{1}F_{1} = m_{1}\ {}^{1}\dot{v}_{c1} = m_{1}\begin{bmatrix} gs_{1} - L_{1}\dot{\theta}_{1}^{2} \\ gc_{1} + L_{1}\ddot{\theta}_{1} \\ 0 \end{bmatrix} = \begin{bmatrix} m_{1}gs_{1} - m_{1}L_{1}\dot{\theta}_{1}^{2} \\ m_{1}gc_{1} + m_{1}L_{1}\ddot{\theta}_{1} \\ 0 \end{bmatrix} $$

$$ {}^{1}N_{1} = \left(\underbrace{{}^{c1}I_{1}}_{0} \cdot\ {}^{1}\dot{\omega}_{1}\right) + \left({}^{1}\omega_{1} \times \left(\underbrace{{}^{c1}I_{1}}_{0} \cdot\ {}^{1}\omega_{1}\right)\right) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} $$

Example – Outward Iteration (7)

- We now continue with the outward iteration for the second link. First we calculate the velocities and accelerations of frame {2}.

$$
{}^{2}\omega_2 = \left({}^{2}_{1}R \cdot {}^{1}\omega_1\right) + \left(\dot{\theta}_2 \cdot {}^{2}\hat{Z}_2\right) = \begin{bmatrix} c_2 & s_2 & 0 \\ -s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 + \dot{\theta}_2 \end{bmatrix}
$$

$$
{}^{2}\dot{\omega}_2 = \left({}^{2}_{1}R \cdot {}^{1}\dot{\omega}_1\right) + \left(\left({}^{2}_{1}R \cdot {}^{1}\omega_1\right) \times \left(\dot{\theta}_2 \cdot {}^{2}\hat{Z}_2\right)\right) + \left(\ddot{\theta}_2 \cdot {}^{2}\hat{Z}_2\right)
$$

$$
= \begin{bmatrix} c_2 & s_2 & 0 \\ -s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \ddot{\theta}_1 \end{bmatrix} + \underbrace{\left(\begin{bmatrix} c_2 & s_2 & 0 \\ -s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix}\right) \times \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_2 \end{bmatrix}}_{0} + \begin{bmatrix} 0 \\ 0 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \ddot{\theta}_1 + \ddot{\theta}_2 \end{bmatrix}
$$

Example – Outward Iteration (8)

$$
\begin{aligned}
{}^{2}\dot{v}_2 &= \left({}^{2}_{1}R \cdot {}^{1}\dot{v}_1\right) + \left(2\,{}^{2}\omega_2 \times \left(\underbrace{\dot{d}_2}_{0} \cdot {}^{2}\hat{Z}_2\right)\right) + \left(\underbrace{\ddot{d}_2}_{0} \cdot {}^{2}\hat{Z}_2\right) \\
&\quad + {}^{2}_{1}R\left(\left({}^{1}\dot{\omega}_1 \times {}^{1}P_2\right) + \left({}^{1}\omega_1 \times \left({}^{1}\omega_1 \times {}^{1}P_2\right)\right)\right) \\
&= \begin{bmatrix} c_2 & s_2 & 0 \\ -s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} g s_1 \\ g c_1 \\ 0 \end{bmatrix} + 0 + 0 \\
&\quad + \begin{bmatrix} c_2 & s_2 & 0 \\ -s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}\left(\begin{bmatrix} 0 \\ 0 \\ \ddot{\theta}_1 \end{bmatrix} \times \begin{bmatrix} L_1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} \times \left(\begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} \times \begin{bmatrix} L_1 \\ 0 \\ 0 \end{bmatrix}\right)\right) \\
&= \begin{bmatrix} -L_1 c_2 \dot{\theta}_1^2 + L_1 s_2 \ddot{\theta}_1 + g s_{12} \\ L_1 s_2 \dot{\theta}_1^2 + L_1 c_2 \ddot{\theta}_1 + g c_{12} \\ 0 \end{bmatrix}
\end{aligned}
$$

Example – Outward Iteration (9)

- We then propagate the acceleration from frame {2} to centre of mass of link 2.

$$
\begin{aligned}
{}^{2}\dot{v}_{c2} &= \left({}^{2}\dot{v}_{2}\right) + \left({}^{2}\dot{\omega}_{2} \times {}^{2}P_{c2}\right) + \left({}^{2}\omega_{2} \times \left({}^{2}\omega_{2} \times {}^{2}P_{c2}\right)\right) \\
&= \begin{bmatrix} -L_1 c_2 \dot{\theta}_1^2 + L_1 s_2 \ddot{\theta}_1 + g s_{12} \\ L_1 s_2 \dot{\theta}_1^2 + L_1 c_2 \ddot{\theta}_1 + g c_{12} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \ddot{\theta}_1 + \ddot{\theta}_2 \end{bmatrix} \times \begin{bmatrix} L_2 \\ 0 \\ 0 \end{bmatrix} \\
&\quad + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 + \dot{\theta}_2 \end{bmatrix} \times \left(\begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 + \dot{\theta}_2 \end{bmatrix} \times \begin{bmatrix} L_2 \\ 0 \\ 0 \end{bmatrix} \right) \\
&= \begin{bmatrix} -L_1 c_2 \dot{\theta}_1^2 + L_1 s_2 \ddot{\theta}_1 + g s_{12} - L_2 \left(\dot{\theta}_1 + \dot{\theta}_2\right)^2 \\ L_1 s_2 \dot{\theta}_1^2 + L_1 c_2 \ddot{\theta}_1 + g c_{12} + L_2 \left(\ddot{\theta}_1 + \ddot{\theta}_2\right) \\ 0 \end{bmatrix}
\end{aligned}
$$

Example – Outward Iteration (10)

- Finally, we can calculate the total force and moment acting on link 2:

$$\boxed{{}^2F_2 = m_2\ {}^2\dot{v}_{c2} = \begin{bmatrix} -m_2L_1c_2\dot{\theta}_1^2 + m_2L_1s_2\ddot{\theta}_1 + m_2gs_{12} - m_2L_2\left(\dot{\theta}_1 + \dot{\theta}_2\right)^2 \\ m_2L_1s_2\dot{\theta}_1^2 + m_2L_1c_2\ddot{\theta}_1 + m_2gc_{12} + m_2L_2\left(\ddot{\theta}_1 + \ddot{\theta}_2\right) \\ 0 \end{bmatrix}}$$

$$\boxed{{}^2N_2 = \left(\underbrace{{}^{c2}I_2}_{0} \cdot\ {}^2\dot{\omega}_2\right) + \left({}^2\omega_2 \times \left(\underbrace{{}^{c2}I_2}_{0} \cdot\ {}^2\omega_2\right)\right) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}}$$

- The outward iteration is complete now.

Example – Inward Iteration (1)

- Now let's do the inward iteration to calculate forces and torques at the joints.
- Because the end-effector is not in contact with the environment, we start with:

$$\boxed{{}^3f_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \ {}^3n_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}}$$

Example – Inward Iteration (2)

- Using the inward iteration formula,

$$\boxed{{}^{2}f_2 = \left({}^{2}_{3}R \cdot \underbrace{{}^{3}f_3}_{0}\right) + {}^{2}F_2 = \begin{bmatrix} {}^{2}F_{2x} \\ {}^{2}F_{2y} \\ {}^{2}F_{2z} \end{bmatrix} = \begin{bmatrix} {}^{2}F_{2x} \\ {}^{2}F_{2y} \\ 0 \end{bmatrix}}$$

  - Hint: Rather than writing the long expression for ${}^{2}F_2$, we just use the symbols ${}^{2}F_{2x}$ etc. and substitute back at the end.
  - This makes the subsequent calculations simpler.

Example – Inward Iteration (3)

$$ {}^{2}n_2 = \left({}^{2}_{3}R \cdot \underbrace{{}^{3}n_3}_{0}\right) + \left({}^{2}P_{c2} \times {}^{2}F_2\right) + \left({}^{2}P_3 \times \left({}^{2}_{3}R \cdot \underbrace{{}^{3}f_3}_{0}\right)\right) + \underbrace{{}^{2}N_2}_{0} $$

$$ = \begin{bmatrix} L_2 \\ 0 \\ 0 \end{bmatrix} \times \begin{bmatrix} {}^{2}F_{2x} \\ {}^{2}F_{2y} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ L_2\,{}^{2}F_{2y} \end{bmatrix} $$

$$ {}^{1}f_1 = \left({}^{1}_{2}R \cdot {}^{2}f_2\right) + {}^{1}F_1 = \begin{bmatrix} c_2 & -s_2 & 0 \\ s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{2}F_{2x} \\ {}^{2}F_{2y} \\ 0 \end{bmatrix} + \begin{bmatrix} {}^{1}F_{1x} \\ {}^{1}F_{1y} \\ 0 \end{bmatrix} $$

$$ = \begin{bmatrix} c_2\,{}^{2}F_{2x} - s_2\,{}^{2}F_{2y} + {}^{1}F_{1x} \\ s_2\,{}^{2}F_{2x} + c_2\,{}^{2}F_{2y} + {}^{1}F_{1y} \\ 0 \end{bmatrix} $$

Example – Inward Iteration (4)

$$
{}^{1}n_1 = \left({}^{1}_{2}R \cdot {}^{2}n_2\right) + \left({}^{1}P_{c1} \times {}^{1}F_1\right) + \left({}^{1}P_2 \times \left({}^{1}_{2}R \cdot {}^{2}f_2\right)\right) + \underbrace{{}^{1}N_1}_{0}
$$

$$
= \begin{bmatrix} c_2 & -s_2 & 0 \\ s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ L_2\ {}^{2}F_{2y} \end{bmatrix} + \begin{bmatrix} L_1 \\ 0 \\ 0 \end{bmatrix} \times \begin{bmatrix} {}^{1}F_{1x} \\ {}^{1}F_{1y} \\ 0 \end{bmatrix} + \begin{bmatrix} L_1 \\ 0 \\ 0 \end{bmatrix} \times \left(\begin{bmatrix} c_2 & -s_2 & 0 \\ s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{2}F_{2x} \\ {}^{2}F_{2y} \\ 0 \end{bmatrix} \right)
$$

$$
= \begin{bmatrix} 0 \\ 0 \\ L_2\ {}^{2}F_{2y} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ L_1\ {}^{1}F_{1y} \end{bmatrix} + \begin{bmatrix} L_1 \\ 0 \\ 0 \end{bmatrix} \times \begin{bmatrix} c_2\ {}^{2}F_{2x} - s_2\ {}^{2}F_{2y} \\ s_2\ {}^{2}F_{2x} + c_2\ {}^{2}F_{2y} \\ 0 \end{bmatrix}
$$

$$
= \begin{bmatrix} 0 \\ 0 \\ L_2\ {}^{2}F_{2y} + L_1\ {}^{1}F_{1y} + L_1\left(s_2\ {}^{2}F_{2x} + c_2\ {}^{2}F_{2y}\right) \end{bmatrix}
$$

Example – Inward Iteration (5)

- The inward iteration is now complete.
- We can finally extract the joint torque / forces.
- Since both joints are revolute, we use $\boxed{\tau_i = {}^{i}n_i^T \cdot {}^{i}\hat{Z}_i}$ which means the third row of ${}^{i}n_i$.

$$\boxed{\begin{aligned}\tau_2 &= \text{3rd row of } {}^{2}n_2 \\ &= L_2 \, {}^{2}F_{2y} \\ &= L_2\left(m_2 L_1 s_2 \dot{\theta}_1^2 + m_2 L_1 c_2 \ddot{\theta}_1 + m_2 g c_{12} + m_2 L_2\left(\ddot{\theta}_1 + \ddot{\theta}_2\right)\right) \\ &= m_2 L_1 L_2 s_2 \dot{\theta}_1^2 + m_2 L_1 L_2 c_2 \ddot{\theta}_1 + m_2 g L_2 c_{12} + m_2 L_2^2\left(\ddot{\theta}_1 + \ddot{\theta}_2\right)\end{aligned}}$$

Example – Inward Iteration (6)

$$\begin{aligned}
\tau_1 &= \text{3rd row of } {}^1n_1 \\
&= L_2\ {}^2F_{2y} + L_1\ {}^1F_{1y} + L_1\left(s_2\ {}^2F_{2x} + c_2\ {}^2F_{2y}\right) \\
&= \cdots \\
&= m_2L_2^2\left(\ddot{\theta}_1 + \ddot{\theta}_2\right) + m_2L_1L_2c_2\left(2\ddot{\theta}_1 + \ddot{\theta}_2\right) + (m_1 + m_2)L_1^2\ddot{\theta}_1 - m_2L_1L_2s_2\dot{\theta}_2^2 \\
&\quad -2m_2L_1L_2s_2\dot{\theta}_1\dot{\theta}_2 + m_2gL_2c_{12} + (m_1 + m_2)gL_1c_1
\end{aligned}$$

Structure of Dynamics Equations (1)

- As a summary:

$$\boxed{\begin{aligned}\tau_1 = {} & m_2 L_2^2(\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 L_1 L_2 c_2(2\ddot{\theta}_1 + \ddot{\theta}_2) + (m_1 + m_2)L_1^2\ddot{\theta}_1 - m_2 L_1 L_2 s_2 \dot{\theta}_2^2 \\ & -2m_2 L_1 L_2 s_2 \dot{\theta}_1 \dot{\theta}_2 + m_2 g L_2 c_{12} + (m_1 + m_2) g L_1 c_1\end{aligned}}$$

$$\boxed{\tau_2 = m_2 L_1 L_2 s_2 \dot{\theta}_1^2 + m_2 L_1 L_2 c_2 \ddot{\theta}_1 + m_2 g L_2 c_{12} + m_2 L_2^2(\ddot{\theta}_1 + \ddot{\theta}_2)}$$

- Recall that manipulator’s dynamic equations have the following structure:

$$\boxed{M(q)\ddot{q} + V(q,\dot{q}) + G(q) = \tau}$$

Structure of Dynamics Equations (2)

- The τ_1 and τ_2 equations can thus be rewritten as:

$$\underbrace{\begin{bmatrix} m_2L_2^2 + 2m_2L_1L_2c_2 + (m_1+m_2)L_1^2 & m_2L_2^2 + m_2L_1L_2c_2 \\ m_2L_2^2 + m_2L_1L_2c_2 & m_2L_2^2 \end{bmatrix}}_{M(q)} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix}$$

$$+ \underbrace{\underbrace{\begin{bmatrix} -m_2L_1L_2s_2\dot{\theta}_2^2 \\ m_2L_1L_2s_2\dot{\theta}_1^2 \end{bmatrix}}_{\text{Centrifugal}} + \underbrace{\begin{bmatrix} -2m_2L_1L_2s_2\dot{\theta}_1\dot{\theta}_2 \\ 0 \end{bmatrix}}_{\text{Coriolis}}}_{V(q,\dot{q})} + \underbrace{\begin{bmatrix} m_2gL_2c_{12} + (m_1+m_2)gL_1c_1 \\ m_2gL_2c_{12} \end{bmatrix}}_{G(q)} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$$

- This form will be useful for simulation later.

Content

- Introduction
- Newton-Euler Formulation
- Inclusion of Friction Force
- Manipulator Dynamics in Cartesian Space
- Derivation of Inertia Tensor
- Matlab Simulation
- Tutorial Questions

Friction (1)

- All mechanisms are affected by friction.
- The effect of friction to the manipulator's dynamic can be included in the dynamic equation:

$$\boxed{M(q)\ddot{q} + V(q,\dot{q}) + G(q) = \tau - \tau_{\text{friction}}}$$

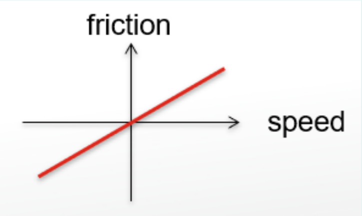
  - It appears on the right hand side with a minus sign, because intuitively it slows down the robot.

- There are several models for friction, which will be discussed in this section.

Friction (2)

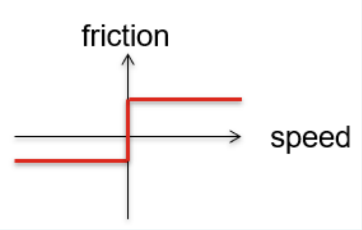
- The viscous friction assumes that the friction is proportional to the velocity:

$$\boxed{\tau_{\text{friction}} = k\dot{q}}$$



- The Coulomb friction assumes that the friction is constant, but the sign depends on the sign of velocity.

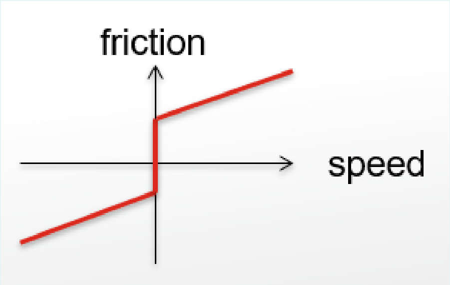
$$\boxed{\tau_{\text{friction}} = c \cdot \text{sgn}(\dot{q})}$$



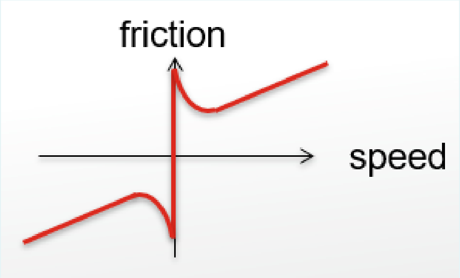
Friction (3)

- The viscous and Coulomb friction can be combined to give a more accurate representation of friction:

$$\boxed{\tau_{\text{friction}} = c \cdot \text{sgn}(\dot{q}) + k\dot{q}}$$



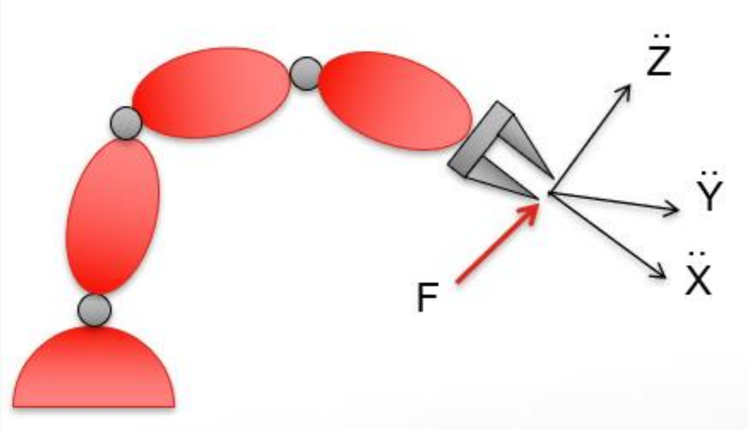
- There are even more accurate models, such as those which include the Stribeck friction.
  - Will not be discussed here.



Content

Why Cartesian Space?

- Rather than looking at how the joint torques provide acceleration to the links in joint space,
- we can also look at how a force acting on the end-effector affects the acceleration of the robot in Cartesian space.
  - This is useful for controlling the robot in force-control operations such as polishing.



Dynamics in Cartesian Space (1)

- The dynamics in Cartesian space is:

$$\boxed{M_x(q)\ddot{x} + V_x(q,\dot{q}) + G_x(q) = F}$$

- This can be derived from our joint-space dynamic equation as follows:
  - The joint-space dynamic equation is $\boxed{M(q)\ddot{q} + V(q,\dot{q}) + G(q) = \tau}$
  - We know that the relationship between forces on the end-effector (F) and the joint torques (τ) is: $\boxed{\tau = J^T(q)F}$ or $\boxed{J^{-T}(q)\tau = F}$
  - We also know that the joint velocities and the end-effector velocity are related as: $\boxed{\dot{x} = J(q)\dot{q}}$

Dynamics in Cartesian Space (2)

- Premultiply the joint-space dynamic equation with $J^{-T}(q)$ gives:

$$\boxed{J^{-T}(q)M(q)\ddot{q} + J^{-T}(q)V(q,\dot{q}) + J^{-T}(q)G(q) = J^{-T}(q)\tau = F}$$

- From $\dot{x} = J(q)\dot{q}$, we can obtain through differentiation:

$$\ddot{x} = \dot{J}(q)\dot{q} + J(q)\ddot{q} \rightarrow \boxed{\ddot{q} = J^{-1}(q)\ddot{x} - J^{-1}(q)\dot{J}(q)\dot{q}}$$

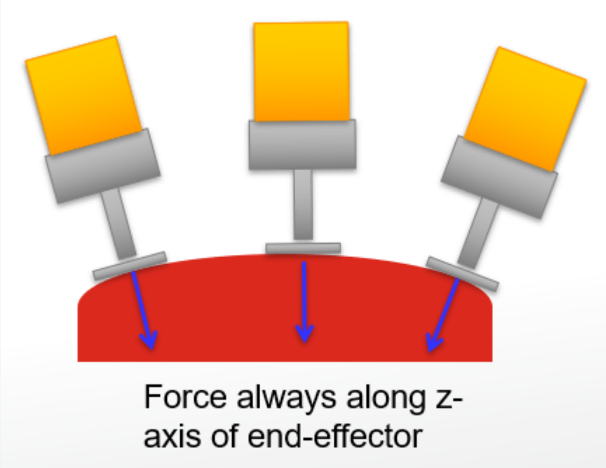
- Substitute the 2nd equation into the first gives:

$$J^{-T}(q)M(q)\left(J^{-1}(q)\ddot{x} - J^{-1}(q)\dot{J}(q)\dot{q}\right) + J^{-T}(q)V(q,\dot{q}) + J^{-T}(q)G(q)$$
$$= \boxed{\underbrace{J^{-T}M(q)J^{-1}}_{M_x}\ddot{x} + \underbrace{J^{-T}\left(V(q,\dot{q}) - M(q)J^{-1}\dot{J}\dot{q}\right)}_{V_x} + \underbrace{J^{-T}G(q)}_{G_x} = F}$$

Dynamics in Cartesian Space (3)

- Note: Rather than using ${}^{0}J_v$, it is advantageous to use the Jacobian wrt. end-effector frame ${}^{e}J_v$ instead.
- This is because we often want the force to be along certain end-effector axis (e.g. z-axis), regardless of position and orientation of the end-effector.
- ${}^{e}J_v$ can be calculated from ${}^{0}J_v$ as:

$$\boxed{{}^{e}J_v = {}^{e}_{0}R \cdot {}^{0}J_v}$$



Example: Cartesian Dynamics (1)

- For the two link robot example just now, the Jacobian matrix wrt. base frame was:

$$\boxed{{}^{0}J_v = \begin{bmatrix} -L_1 s_1 - L_2 s_{12} & -L_2 s_{12} \\ L_1 c_1 + L_2 c_{12} & L_2 c_{12} \end{bmatrix}}$$

- The Jacobian wrt. end-effector frame is therefore:

$$\boxed{\begin{aligned} {}^{e}J_v &= {}^{e}_{0}R \cdot {}^{0}J_v \\ &= \begin{bmatrix} c_{12} & s_{12} \\ -s_{12} & c_{12} \end{bmatrix} \begin{bmatrix} -L_1 s_1 - L_2 s_{12} & -L_2 s_{12} \\ L_1 c_1 + L_2 c_{12} & L_2 c_{12} \end{bmatrix} \\ &= \begin{bmatrix} L_1 s_2 & 0 \\ L_1 c_2 + L_2 & L_2 \end{bmatrix} \end{aligned}}$$

Example: Cartesian Dynamics (2)

- This gives:

$$\boxed{{}^{e}J_v^{-1} = \frac{1}{L_1 L_2 s_2}\begin{bmatrix} L_2 & 0 \\ -L_1 c_2 - L_2 & L_1 s_2 \end{bmatrix}}$$

$$\boxed{{}^{e}\dot{J}_v = \begin{bmatrix} L_1 c_2 \dot{\theta}_2 & 0 \\ -L_1 s_2 \dot{\theta}_2 & 0 \end{bmatrix}}$$

Example: Cartesian Dynamics (3)

- Substituting these into:

$$\boxed{\underbrace{J^{-T}M(q)J^{-1}}_{M_x}\ddot{x} + \underbrace{J^{-T}\left(V(q,\dot{q}) - M(q)J^{-1}\dot{J}\dot{q}\right)}_{V_x} + \underbrace{J^{-T}G(q)}_{G_x} = F}$$

- gives:

$$\boxed{M_x(q) = \begin{bmatrix} m_2 + \frac{m_1}{s_2^2} & 0 \\ 0 & m_2 \end{bmatrix}}$$

Example: Cartesian Dynamics (4)

$$V_x(q,\dot{q}) = \begin{bmatrix} -(m_2L_1c_2 + m_2L_2)\dot{\theta}_1^2 - m_2L_2\dot{\theta}_2^2 - \left(2m_2L_2 + m_2L_1c_2 + m_1L_1\frac{c_2}{s_2^2}\right)\dot{\theta}_1\dot{\theta}_2 \\ m_2L_1s_2\dot{\theta}_1^2 + m_2L_1s_2\dot{\theta}_1\dot{\theta}_2 \end{bmatrix}$$

$$G_x(q) = \begin{bmatrix} m_1g\frac{c_1}{s_2} + m_2gs_{12} \\ m_2gc_{12} \end{bmatrix}$$

Content

- Introduction
- Newton-Euler Formulation
- Inclusion of Friction Force
- Manipulator Dynamics in Cartesian Space
- Derivation of Inertia Tensor
- Matlab Simulation
- Tutorial Questions

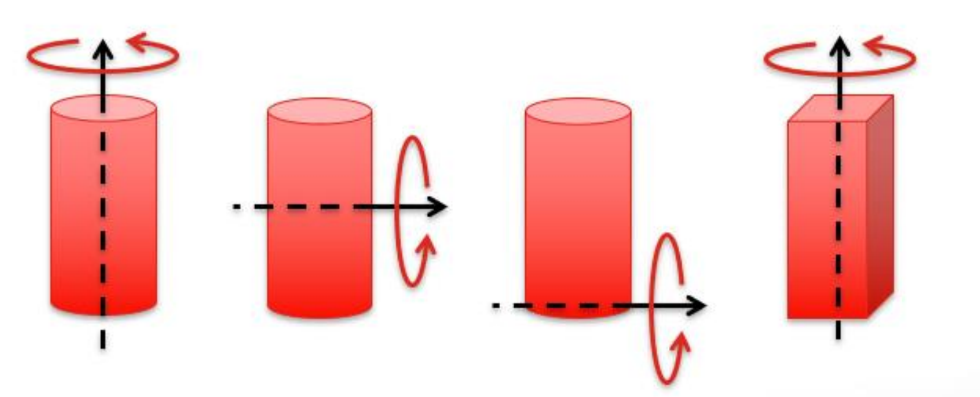
Inertia Tensor (1)

- We have so far calculated the manipulator's dynamics assuming that the inertia tensor ${}^{ci}I_i$ is known or given.
- What is it and how can we calculate it if not given?

- Analogy to linear motion: $F = ma$
  - If mass is small, then the acceleration is huge.
  - And if the mass is large, then the acceleration is small.
  - The mass presents a "resistance" to the linear motion.

Inertia Tensor (2)

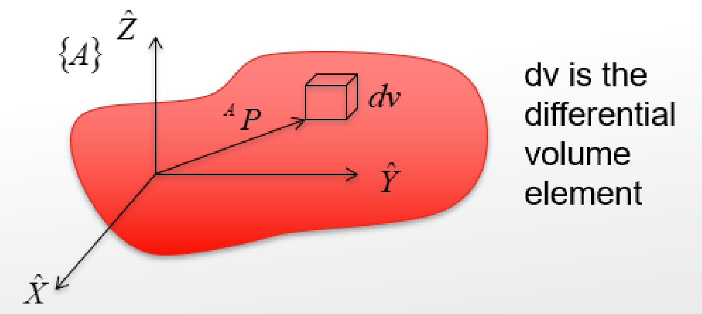
- For the case of rotational motion about a single axis, the moment of inertia also represents a resistance to motion.
- The resistance is different depending on the shape and mass distribution of the object, as well as the axis of rotation.



Inertia Tensor (3)

- For the case of a rigid body which is free to move in 3D space, there are infinitely many possible rotation axis.
- The inertia tensor is the generalization of the moment of inertia.
- For one object, if we place a frame {A} at a particular location, the inertia tensor is:

$$\boxed{{}^{A}I = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix}}$$



Inertia Tensor (4)

- The elements of the inertia tensor are:
- Mass moment of inertia:

$$\boxed{I_{xx} = \iiint\limits_V (y^2 + z^2)\rho dV \,, I_{yy} = \iiint\limits_V (x^2 + z^2)\rho dV \,, I_{zz} = \iiint\limits_V (x^2 + y^2)\rho dV}$$

- Mass product of inertia:

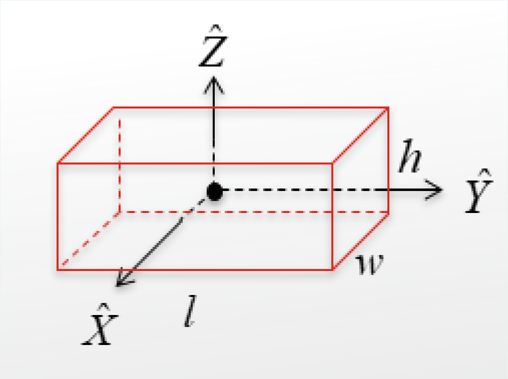
$$\boxed{I_{xy} = \iiint\limits_V xy\rho dV \,, I_{xz} = \iiint\limits_V xz\rho dV \,, I_{yz} = \iiint\limits_V yz\rho dV}$$

Inertia Tensor (5)

- The elements depend on the position and orientation of the frame.
  - If the frame is at a ‘special’ orientation, the products of inertia can be zero.
  - In this case, the axes of the frame are called “principal axes”, and the moments of inertia are called “principal moments of inertia”.

- For manipulator dynamics, we put the frame at the centre of mass of each link, hence ${}^{ci}I_i$.

Example – Inertia Tensor (1)



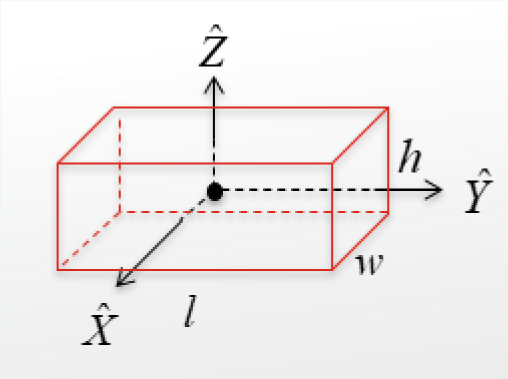
- A rectangular block with length l, width w, and height h.
- The frame is in the centre of the block.

$$I_{xx} = \iiint_V (y^2 + z^2)\rho dV = \int_{-h/2}^{h/2}\int_{-l/2}^{l/2}\int_{-w/2}^{w/2}(y^2 + z^2)\rho dx \cdot dy \cdot dz$$

$$= \int_{-h/2}^{h/2}\int_{-l/2}^{l/2}(y^2 + z^2)w\,\rho \cdot dy \cdot dz = \int_{-\frac{h}{2}}^{\frac{h}{2}}\left(\frac{l^3}{12} + z^2 l\right)w\rho \cdot dz$$

$$= \left(\frac{l^3 h}{12} + \frac{h^3 l}{12}\right)w\rho = \left(\frac{l^2}{12} + \frac{h^2}{12}\right)\underbrace{\underbrace{hlw}_{V}\rho}_{m} = \frac{m}{12}(l^2 + h^2)$$

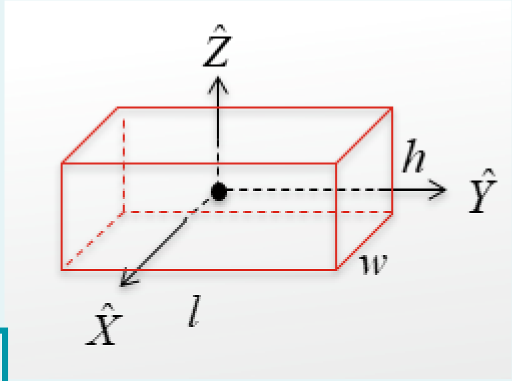
Example – Inertia Tensor (2)



- By symmetry,

$$\boxed{\begin{aligned} I_{yy} &= \frac{m}{12}(w^2 + h^2) \\ I_{zz} &= \frac{m}{12}(w^2 + l^2) \end{aligned}}$$

Example – Inertia Tensor (3)



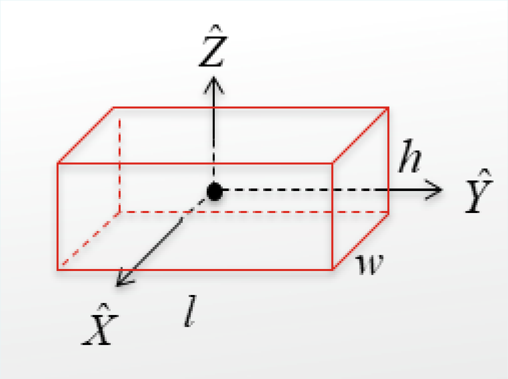
- Next,

$$I_{xy} = \iiint_V xy\rho dV = \int_{-h/2}^{h/2} \int_{-l/2}^{l/2} \int_{-w/2}^{w/2} xy\rho dx \cdot dy \cdot dz$$

$$= \int_{-h/2}^{h/2} \int_{-l/2}^{l/2} \left. \frac{x^2}{2} \right|_{x=-w/2}^{w/2} y\rho \cdot dy \cdot dz$$

$$= \int_{-h/2}^{h/2} \int_{-l/2}^{l/2} 0 \cdot y\rho \cdot dy \cdot dz = 0$$

Example – Inertia Tensor (4)



- By symmetry,

$$\boxed{\begin{aligned} I_{xz} &= 0 \\ I_{yz} &= 0 \end{aligned}}$$

- Therefore, the inertia tensor for the rectangular block with respect to a frame in its centre is:

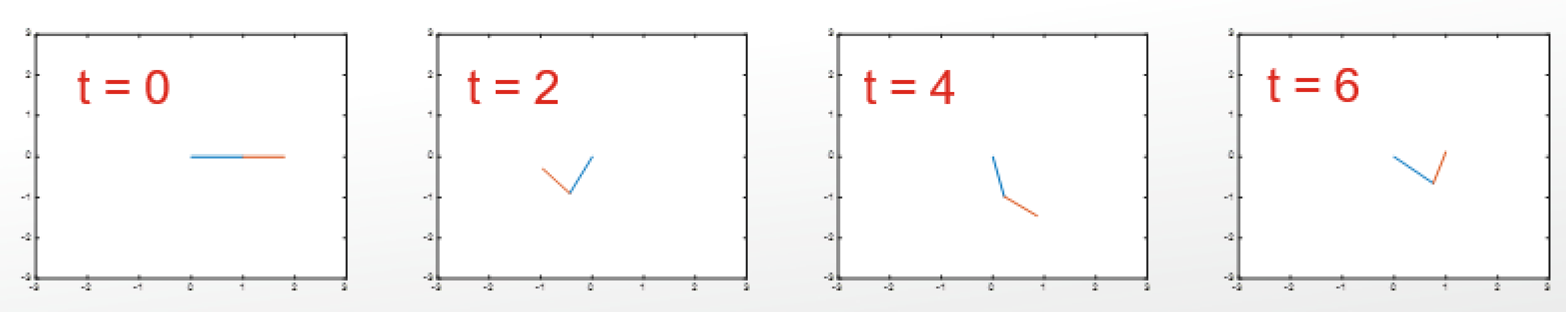
$$\boxed{{}^{C}I = \begin{bmatrix} \frac{m}{12}(l^2+h^2) & 0 & 0 \\ 0 & \frac{m}{12}(w^2+h^2) & 0 \\ 0 & 0 & \frac{m}{12}(w^2+l^2) \end{bmatrix}}$$

Content

- Introduction
- Newton-Euler Formulation
- Inclusion of Friction Force
- Manipulator Dynamics in Cartesian Space
- Derivation of Inertia Tensor
- Matlab Simulation
- Tutorial Questions

Simulation (1)

- As mentioned in the introduction, the dynamics provide us a model (equations of motions) for simulation and control design purpose.
  - The control design will be taught in later weeks.
  - Here, we will see how to simulate a robot – Given joint torques, how will the robot move.



Simulation (2)

- The idea of the simulation is as follows:
- From the manipulator dynamics:

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

- We leave the highest derivative on the left hand side.

$$\boxed{\ddot{q} = M^{-1}(q)\big(\tau - V(q, \dot{q}) - G(q)\big)}$$

  - This is an important equation for our simulation.

Simulation (3)

- Given the initial joint parameters $q_{T0}, \dot{q}_{T0}$, we first calculate $\boxed{M(q_{T0}), V(q_{T0}, \dot{q}_{T0}), G(q_{T0})}$.

- Then, given the value of τ_{T0} at the initial instant, we can calculate the joint acceleration.

$$\boxed{\ddot{q}_{T0} = M^{-1}(q_{T0})\big(\tau_{T0} - V(q_{T0}, \dot{q}_{T0}) - G(q_{T0})\big)}$$

Simulation (4)

- The acceleration will give rise to change in velocity and position.
- If we assume that the acceleration is constant for a duration of T, the new velocity and position at the end of T will be:

$$\boxed{\begin{aligned}\dot{q}_{T1} &= \dot{q}_{T0} + \ddot{q}_{T0}T \\ q_{T1} &= q_{T0} + \dot{q}_{T0}T + \frac{1}{2}\ddot{q}_{T0}T^2\end{aligned}}$$

Simulation (5)

- Next, using the newly calculated $q_{T1}, \dot{q}_{T1}$, we first calculate $\boxed{M(q_{T1}), V(q_{T1}, \dot{q}_{T1}), G(q_{T1}).}$

- Then, given the value of τ_{T1} at the next instant, we can calculate the joint acceleration.

$$\boxed{\ddot{q}_{T1} = M^{-1}(q_{T1})\big(\tau_{T1} - V(q_{T1}, \dot{q}_{T1}) - G(q_{T1})\big)}$$

Simulation (6)

- The new velocity and position at the end of next T will be:

$$\boxed{\begin{aligned}\dot{q}_{T2} &= \dot{q}_{T1} + \ddot{q}_{T1}T \\ q_{T2} &= q_{T1} + \dot{q}_{T1}T + \frac{1}{2}\ddot{q}_{T1}T^2\end{aligned}}$$

- This process can be implemented in programming language using a FOR loop.

Example – Simulation (1)

- We will use the same 2 link robot in our example:

$$\underbrace{\begin{bmatrix} m_2L_2^2 + 2m_2L_1L_2c_2 + (m_1+m_2)L_1^2 & m_2L_2^2 + m_2L_1L_2c_2 \\ m_2L_2^2 + m_2L_1L_2c_2 & m_2L_2^2 \end{bmatrix}}_{M(q)} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix}$$

$$+ \underbrace{\underbrace{\begin{bmatrix} -m_2L_1L_2s_2\dot{\theta}_2^2 \\ m_2L_1L_2s_2\dot{\theta}_1^2 \end{bmatrix}}_{\text{Centrifugal}} + \underbrace{\begin{bmatrix} -2m_2L_1L_2s_2\dot{\theta}_1\dot{\theta}_2 \\ 0 \end{bmatrix}}_{\text{Coriolis}}}_{V(q,\dot{q})} + \underbrace{\begin{bmatrix} m_2gL_2c_{12} + (m_1+m_2)gL_1c_1 \\ m_2gL_2c_{12} \end{bmatrix}}_{G(q)} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$$

- And we will do the simulation in Matlab.

Example – Simulation (2)

- First we provide parameter values and torque values.

```
%%%%%%%%%%%%%%%%%%%
% Robot Parameters %
%%%%%%%%%%%%%%%%%%%

m1 = 3;    % kg
m2 = 2;    % m
L1 = 3;    % kg
L2 = 2;    % m
g = 9.8;   % m/s^2

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Torque Arrays and Time Interval %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Tau1 = ones(10000,1)*0.1;   % constant torque with value = multiplier
Tau2 = ones(10000,1)*0.1;   % constant torque with value = multiplier
T = 1e-3;                   % time interval for integration
```

Example – Simulation (3)

- We also provide initial conditions, and declare variables to record results.

```
%%%%%%%%%%%%%%%%%%%%%%
% Initial Conditions %
%%%%%%%%%%%%%%%%%%%%%%

q1 = 0;      % joint 1 angle
q1Dot = 0;   % joint 1 angular velocity
q2 = 0;      % joint 2 angle
q2Dot = 0;   % joint 2 angular velocity

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculate Acceleration and Update Positions %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

q1DoubleDotRecord = [];
q2DoubleDotRecord = [];
q1DotRecord = [];
q2DotRecord = [];
q1Record = [];
q2Record = [];
```

Example – Simulation (4)

- For each time step, calculate $M(q), V(q, \dot{q}), G(q)$.

```
for i=1:length(Tau1)

    %%%%%%%%%%%%%%%%%%%%%%
    % Calculate M, V, G %
    %%%%%%%%%%%%%%%%%%%%%%

    m11 = m2*L2^2 + 2*m2*L1*L2*cos(q2) + (m1+m2)*L1^2;
    m12 = m2*L2^2 + m2*L1*L2*cos(q2);
    m21 = m12;
    m22 = m2*L2^2;
    M = [m11,m12;m21,m22];

    v1 = -m2*L1*L2*sin(q2)*q2Dot^2 - 2*m2*L1*L2*sin(q2)*q1Dot*q2Dot;
    v2 = m2*L1*L2*sin(q2)*q1Dot^2;
    V = [v1;v2];

    g1 = m2*g*L2*cos(q1+q2) + (m1+m2)*g*L1*cos(q1);
    g2 = m2*g*L2*cos(q1+q2);
    G = [g1;g2];
```

Example – Simulation (5)

- Calculate acceleration.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculate Acceleration %
%%%%%%%%%%%%%%%%%%%%%%%%%%

qDoubleDot = inv(M)*([Tau1(i);Tau2(i)]-V-G);
q1DoubleDot = qDoubleDot(1);
q2DoubleDot = qDoubleDot(2);

q1DoubleDotRecord = [q1DoubleDotRecord;q1DoubleDot];
q2DoubleDotRecord = [q2DoubleDotRecord;q2DoubleDot];
```

Example – Simulation (6)

- Use the acceleration to update velocity and position.
- End of FOR loop.

```
    %%%%%%%%%%%%%%%%%%%%%%
    % Calculate Velocity %
    %%%%%%%%%%%%%%%%%%%%%%

    q1Dot = q1Dot + q1DoubleDot*T;
    q2Dot = q2Dot + q2DoubleDot*T;

    q1DotRecord = [q1DotRecord;q1Dot];
    q2DotRecord = [q2DotRecord;q2Dot];

    %%%%%%%%%%%%%%%%%%%%%%
    % Calculate Position %
    %%%%%%%%%%%%%%%%%%%%%%

    q1 = q1 + q1Dot*T + 1/2*q1DoubleDot*T^2;
    q2 = q2 + q2Dot*T + 1/2*q2DoubleDot*T^2;

    q1Record = [q1Record;q1];
    q2Record = [q2Record;q2];

end
```

Example – Simulation (7)

- Animate the robot motion using forward kinematics.

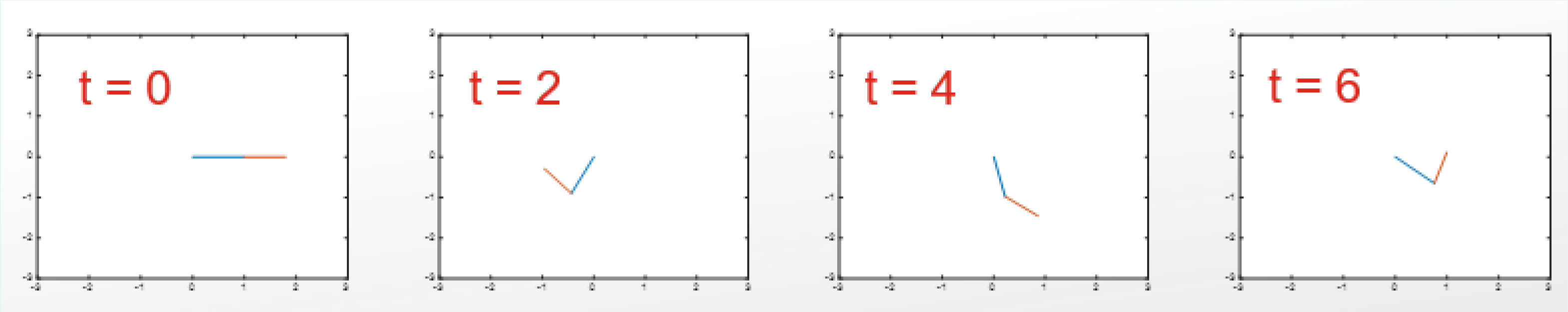
```
%%%%%%%%%%%%%%%%%%%%%
% Show Robot Motion %
%%%%%%%%%%%%%%%%%%%%%

x1 = L1*cos(q1Record);                       % These are kinematic
y1 = L1*sin(q1Record);                       % equation of the
x2 = x1 + L2*cos(q1Record+q2Record);         % 2 link robot
y2 = y1 + L2*sin(q1Record+q2Record);

for i = 1:100:length(x1)     % Plot every 100th data
    clf;             % Clear figure before new plot
    plot([0 x1(i)],[0 y1(i)]);
    axis([-(L1+L2+0.2) (L1+L2+0.2) -(L1+L2+0.2) (L1+L2+0.2)]);
    hold on,plot([x1(i) x2(i)],[y1(i) y2(i)]);
    pause(0.1);
end
```

Example – Simulation (8)

- Run the code and an animation will run.



- You may try changing the robot parameters and the torque array (e.g. change to sinusoid) and see the robot response.

Content

- Introduction
- Newton-Euler Formulation
- Inclusion of Friction Force
- Manipulator Dynamics in Cartesian Space
- Derivation of Inertia Tensor
- Matlab Simulation
- Tutorial Questions

Question 1

- Find the inertia tensor of a right cylinder of homogenous density, with respect to a frame with origin at the center of mass of the body.

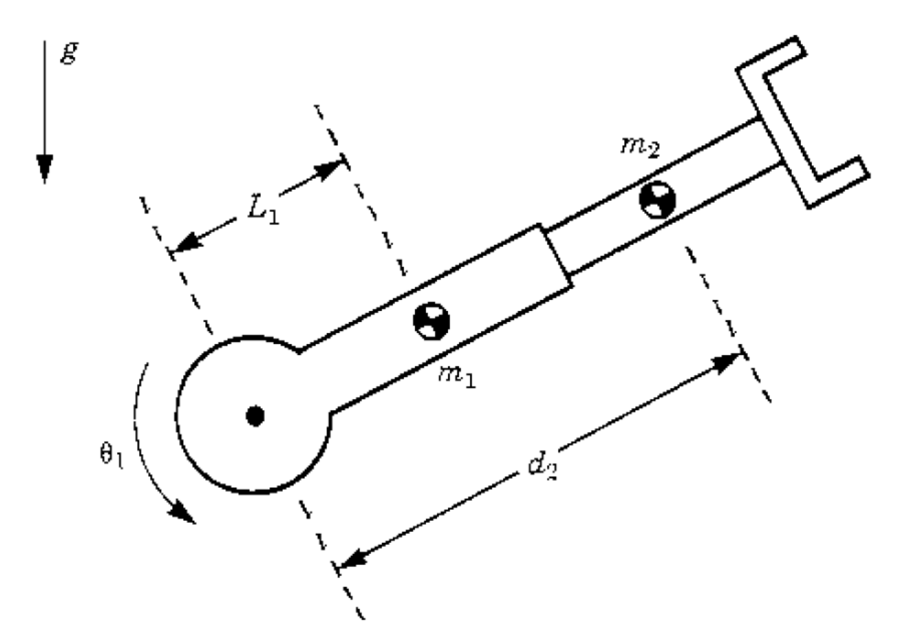
Question 2

- Consider the following robot, with:

$$ {}^{c1}I_1 = \begin{bmatrix} I_{xx1} & 0 & 0 \\ 0 & I_{yy1} & 0 \\ 0 & 0 & I_{zz1} \end{bmatrix} $$

$$ {}^{c2}I_2 = \begin{bmatrix} I_{xx2} & 0 & 0 \\ 0 & I_{yy2} & 0 \\ 0 & 0 & I_{zz2} \end{bmatrix} $$

- Derive its dynamic equations.



Thank you for your attention!

Any questions?