

MEAM 520

Lecture 21: Joint Space Dynamics

Cynthia Sung, Ph.D.

Mechanical Engineering & Applied Mechanics

University of Pennsylvania

Lab 5: Potential Field Planning

MEAM 520, University of Pennsylvania

November 6, 2020

This lab consists of two portions, with a pre-lab due on **Friday, November 13, by midnight (11:59 p.m.)** and a lab (code+report) due on **Friday, November 20, by midnight (11:59 p.m.)**. Late submissions will be accepted until midnight on Saturday following the deadline, but they will be penalized by 25% for each partial or full day late. After the late deadline, no further assignments may be submitted; post a private message on Piazza to request an extension if you need one due to a special situation. This assignment is worth 50 points.

You may talk with other students about this assignment, ask the teaching team questions, use a calculator and other tools, and consult outside sources such as the Internet. To help you actually learn the material, what you submit must be your own work, not copied from any other individual or team. Any submissions suspected of violating Penn's Code of Academic Integrity will be reported to the Office of Student Conduct. When you get stuck, post a question on Piazza or go to office hours!

Individual vs. Pair Programming

Work closely with your partner throughout the lab, following these guidelines, which were adapted from "All I really needed to know about pair programming I learned in kindergarten," by Williams and Kessler, *Communications of the ACM*, May 2000. This article is available on Canvas under Files / Resources.

- Start with a good attitude, setting aside any skepticism, and expect to jell with your partner.
- Don't start alone. Arrange a meeting with your partner as soon as you can.
- Use just one setup, and sit side by side. For a programming component, a desktop computer with a large monitor is better than a laptop. Make sure both partners can see the screen.
- At each instant, one partner should be driving (writing, using the mouse/keyboard, moving the robot) while the other is continuously reviewing the work (thinking and making suggestions).
- Change driving/reviewing roles at least every 30 minutes, *even if one partner is much more experienced than the other*. You may want to set a timer to help you remember to switch.
- If you notice an error in the equation or code that your partner is writing, wait until they finish the line to correct them.
- Stay focused and on-task the whole time you are working together.
- Take a break periodically to refresh your perspective.
- Share responsibility for your project; avoid blaming either partner for challenges you run into.
- Recognize that working in pairs usually takes more time than working alone, but it produces better work, deeper learning, and a more positive experience for the participants.

Lab 5 due 11/20

- Try out a potential field planner.
- Explore the effect of different parameters.
- Choose collision geometries, potential functions, etc.
- Compare to your Lab 3 planner.



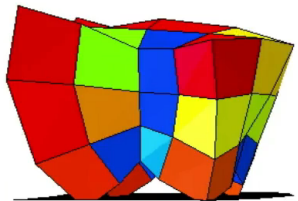
Thanks to Jake for covering



SCF 2020

ACM SYMPOSIUM ON COMPUTATIONAL FABRICATION

NOVEMBER 5-6, 2020



S Kriegman, N Cheney, J Bongard (2018). How morphological development can guide evolution. *Nature Scientific Reports*, 8(1): 13934.



J Bongard



Thanks to Jake for covering



SCF 2020

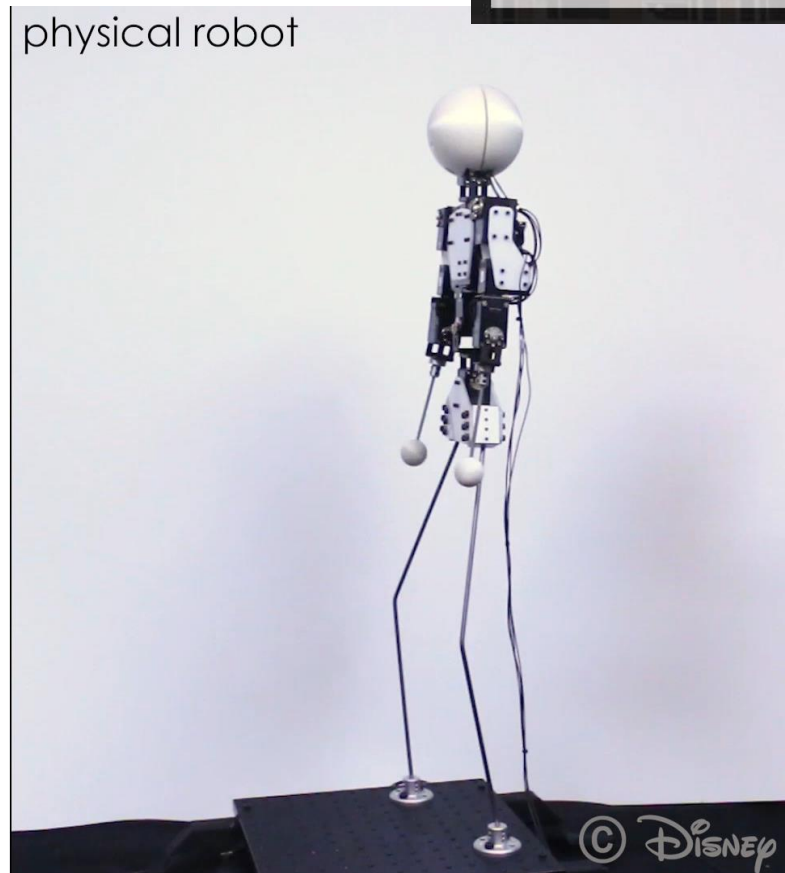
ACM SYMPOSIUM ON COMPUTATIONAL FABRICATION

NOVEMBER 5-6, 2020

input animation



physical robot



Hoshyari, S., Xu, H., Knoop, E., Coros, S. and Bächer, M., 2019.
Vibration-minimizing motion retargeting for robotic characters.
ACM Transactions on Graphics (TOG), 38(4), pp.1-14.

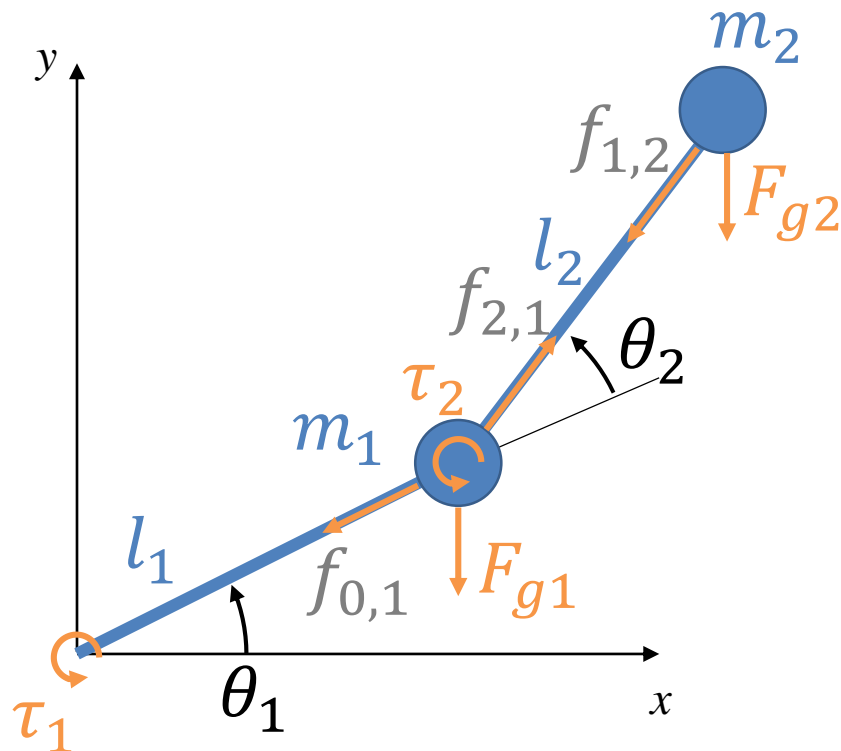


M Bächer

Previously: RR Manipulator w/ mass concentrated at ends of links

AKA the double pendulum

EOM:



coefficients of \ddot{q}_i depend only on q

$$\begin{aligned}
 & [m_1 l_1^2 + m_2 (l_1^2 + l_2^2 + 2l_1 l_2 c_2)] \ddot{\theta}_1 \\
 & + [m_2 (l_2^2 + l_1 l_2 c_2)] \ddot{\theta}_2 - m_2 l_1 l_2 s_2 (2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2) \\
 & + m_1 g l_1 c_1 + m_2 g (l_1 c_1 + l_2 c_{12}) = \tau_1
 \end{aligned}$$

$$\begin{aligned}
 & [m_2 (l_2^2 + l_1 l_2 c_2)] \ddot{\theta}_1 + m_2 l_2 \ddot{\theta}_2 + m_2 l_1 l_2 s_2 \dot{\theta}_1^2 \\
 & + m_2 g l_2 c_{12} = \tau_2
 \end{aligned}$$

centrifugal and Coriolis terms
depend on q and \dot{q}

gravitational terms depend only on q

Previously: Manipulator Equation

We can write this as a matrix equation

$$\tau = \underline{D(q)}\ddot{q} + C(q, \dot{q})\dot{q} + g(q)$$

SHV uses a bit of strange notation.

Most people call this matrix H or M .

where

$D(q)$ is the $n \times n$ mass matrix (inertia terms)

$C(q, \dot{q})$ is the $n \times n$ matrix of centrifugal (square of joint velocities) and Coriolis (product of two different joint velocities) terms

$g(q)$ is a $n \times 1$ vector of gravitational terms

Previously: Manipulator Equation for an N(R) robot

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q)$$

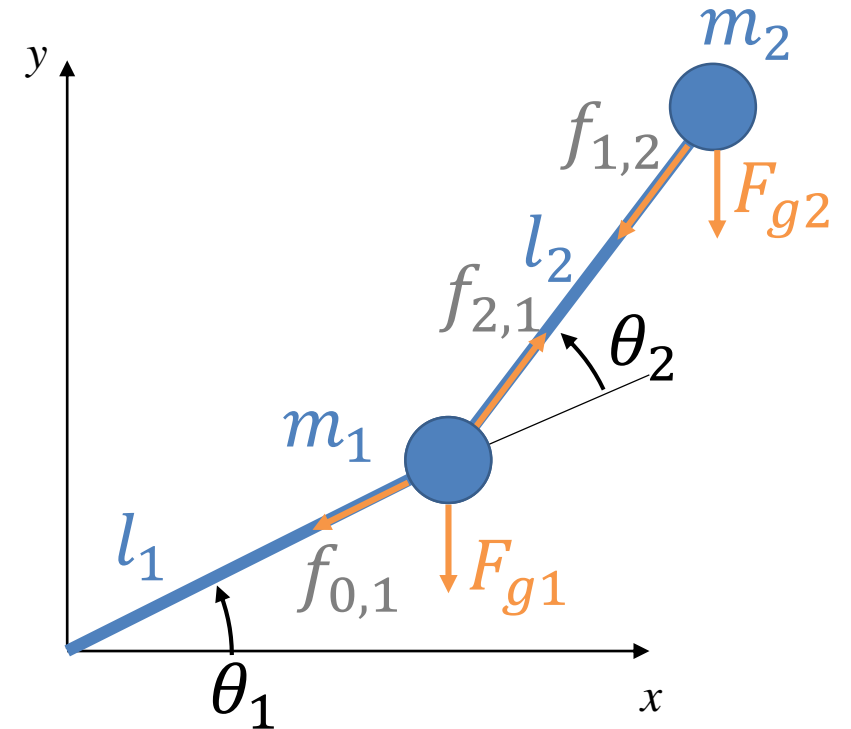
$$D = \sum_{i=1}^N m_i J_{vi}^\top J_{vi}$$

$$g = \frac{\partial}{\partial q} \sum_{i=1}^N m_i \vec{g} \cdot \vec{r}_i$$

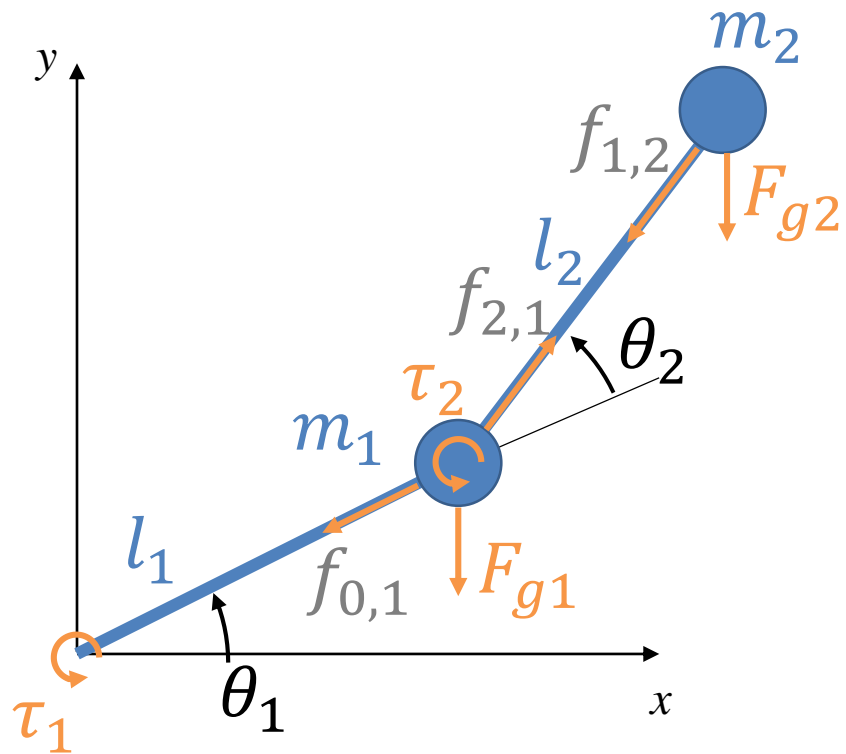
$$(C\dot{q})_k = \sum_{i,j} \frac{1}{2} \left(\frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k} \right) \dot{q}_i \dot{q}_j$$

or

$$c_{kj} = \sum_i \frac{1}{2} \left(\frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k} \right) \dot{q}_i$$

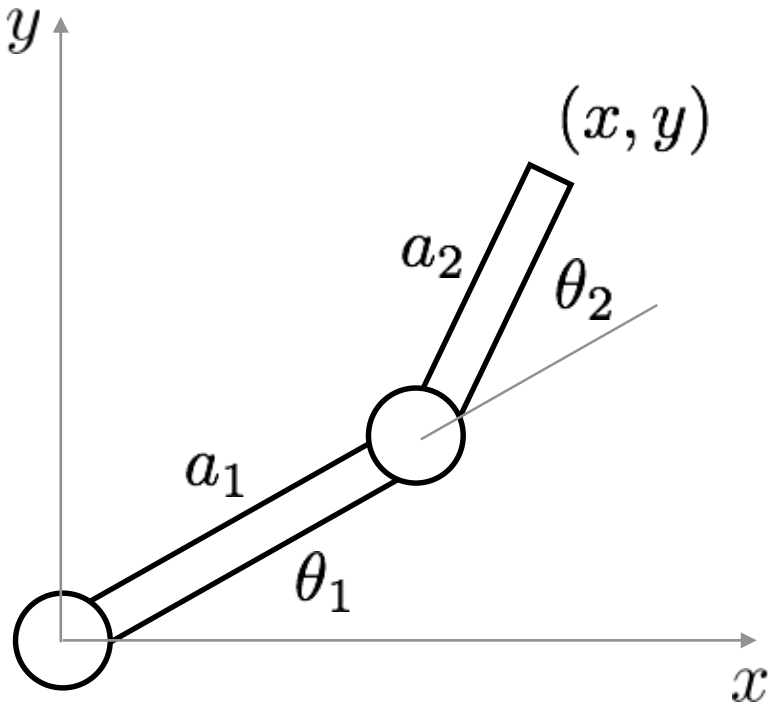


Inertia Tensor



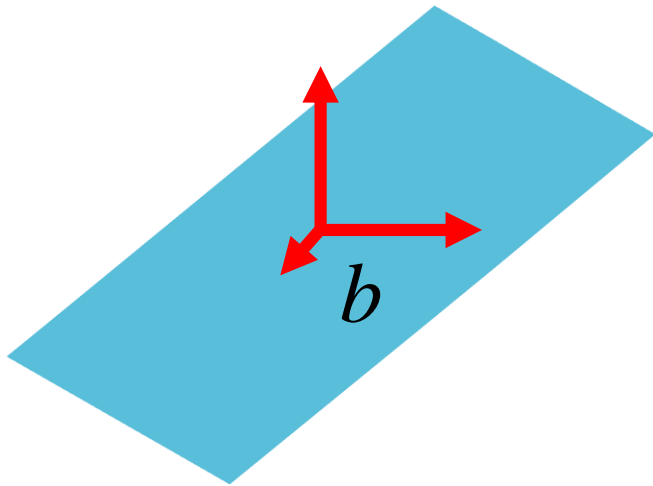
This manipulator has all the mass concentrated at the joints

Inertia Tensor



What happens when you have distributed mass?

Rigid Body Rotating in Space

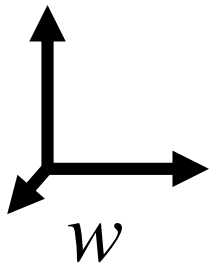


$$\text{Kinetic Energy: } K = \frac{1}{2} m \vec{v}_{COM}^T \vec{v}_{COM} + \frac{1}{2} \vec{\omega}^T I_{COM} \vec{\omega}$$

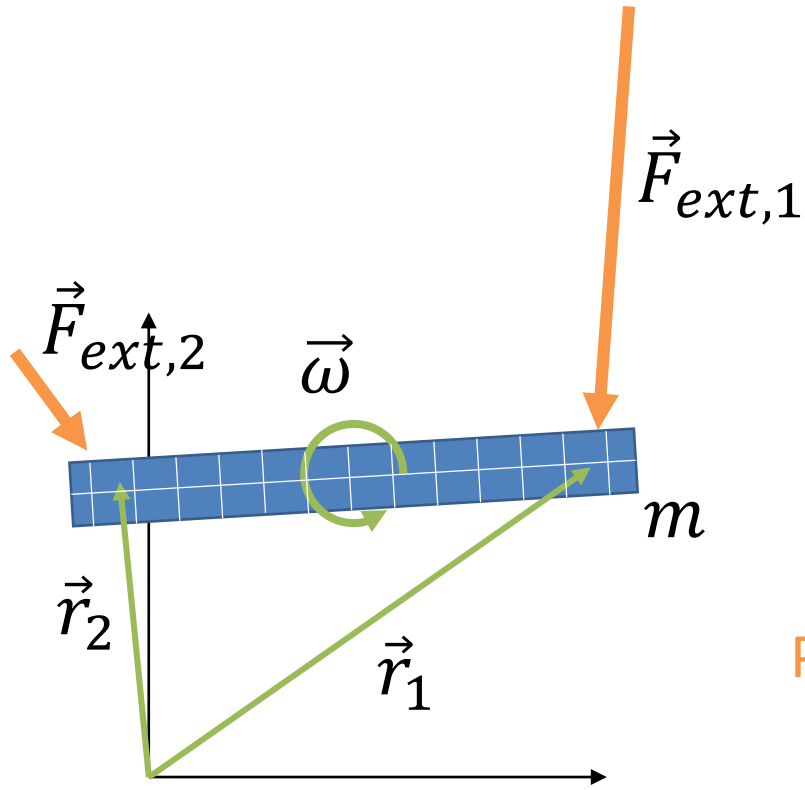
All of these quantities are expressed in the inertial frame (i.e., frame w)

I is constant in frame b but not in frame w

What is I in frame w ?



Rigid Body Dynamics



$$\vec{r}_{COM} = \frac{1}{m} \int \vec{r} dm$$

$$I = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix}$$

Principal Moments of Inertia

$$I_{xx} = \iiint (y^2 + z^2) dm$$

$$I_{yy} = \iiint (x^2 + z^2) dm$$

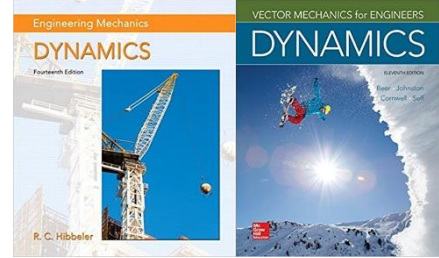
$$I_{zz} = \iiint (x^2 + y^2) dm$$

Products of Inertia

$$I_{xy} = I_{yx} = - \iiint xy dm$$

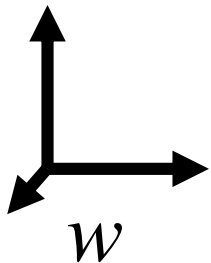
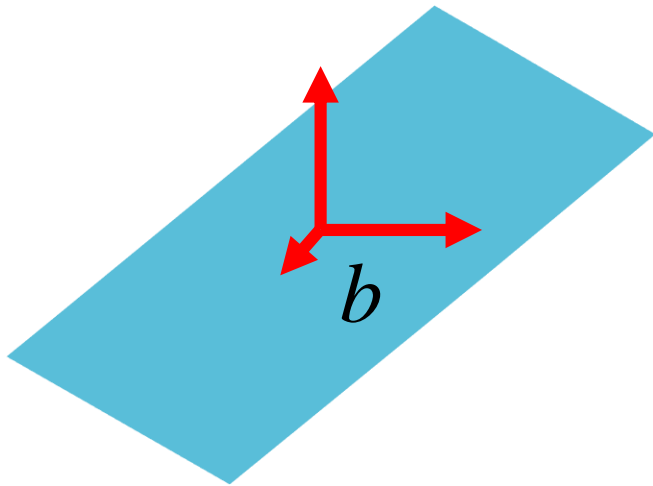
$$I_{xz} = I_{zx} = - \iiint xz dm$$

$$I_{yz} = I_{zy} = - \iiint yz dm$$



Hibbeler Ch. 21.1
Beer Ch. 18.2

Rigid Body Rotating in Space



$$(\vec{\omega}^w)^\top I_{COM}^w \vec{\omega}^w$$

scalar

=

$$(\vec{\omega}^b)^\top \boxed{I_{COM}^b} \vec{\omega}^b$$

known

$$(R_w^b \vec{\omega}^w)^\top I_{COM}^b (R_w^b \vec{\omega}^w)$$

$$(\vec{\omega}^w)^\top (R_w^b)^\top I_{COM}^b R_w^b \vec{\omega}^w$$

$$(\vec{\omega}^w)^\top \boxed{R_b^w I_{COM}^b (R_b^w)^\top} \vec{\omega}^w$$

I_{COM}^w

Manipulator Kinetic Energy

$$K_i = \frac{1}{2} m_i \vec{v}_i^\top \vec{v}_i + \frac{1}{2} \vec{\omega}_i^\top I_i \vec{\omega}_i$$

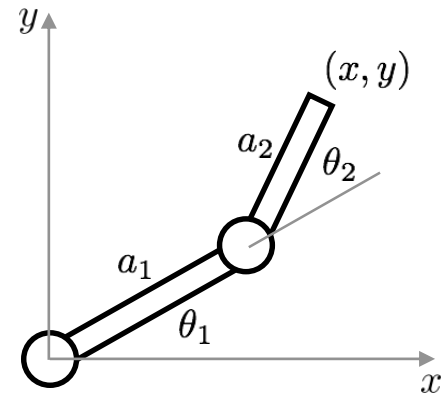
$$K_i = \frac{1}{2} \dot{q}^\top \underbrace{(m_i J_{vi}^\top J_{vi})}_{\text{last time}} \dot{q} + \frac{1}{2} \vec{\omega}_i^\top I_i \vec{\omega}_i$$

$$K_i = \frac{1}{2} \dot{q}^\top (m_i J_{vi}^\top J_{vi}) \dot{q} + \frac{1}{2} \vec{\omega}_i^\top R_i^0 I_i (R_i^0)^\top \vec{\omega}_i$$

$$K_i = \frac{1}{2} \dot{q}^\top (m_i J_{vi}^\top J_{vi}) \dot{q} + \frac{1}{2} (J_{\omega i} \dot{q})^\top R_i^0 I_i (R_i^0)^\top (J_{\omega i} \dot{q})$$

$$K_i = \frac{1}{2} \dot{q}^\top (m_i J_{vi}^\top J_{vi}) \dot{q} + \frac{1}{2} \dot{q}^\top J_{\omega i}^\top R_i^0 I_i (R_i^0)^\top J_{\omega i} \dot{q}$$

$$K_i = \frac{1}{2} \dot{q}^\top \left[m_i J_{vi}^\top J_{vi} + J_{\omega i}^\top R_i^0 I_i (R_i^0)^\top J_{\omega i} \right] \dot{q}$$



Angular velocity Jacobian: $\omega_i = J_{\omega i} \dot{q}$

$$(AB)^\top = B^\top A^\top$$

Inertia Matrix D

Manipulator Equation for an N(R) robot

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q)$$

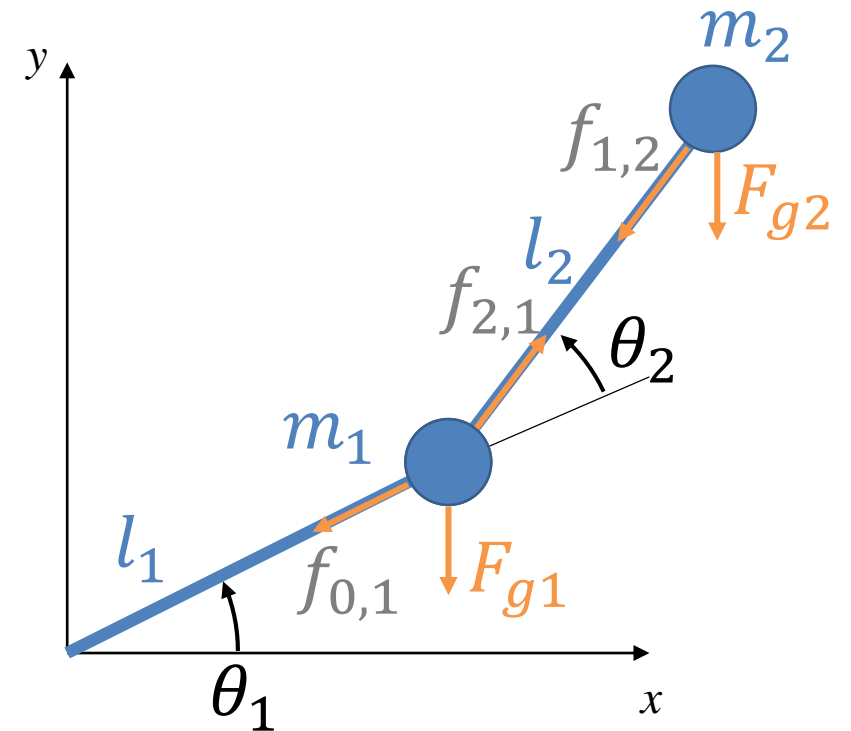
$$D = \sum_{i=1}^N m_i J_{vi}^T J_{vi}$$

$$g = \frac{\partial}{\partial q} \sum_{i=1}^N m_i \vec{g} \cdot \vec{r}_i$$

$$(C\dot{q})_k = \sum_{i,j} \frac{1}{2} \left(\frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k} \right) \dot{q}_i \dot{q}_j$$

or

$$c_{kj} = \sum_i \frac{1}{2} \left(\frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k} \right) \dot{q}_i$$



Manipulator Equation for an N(R) robot

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q)$$

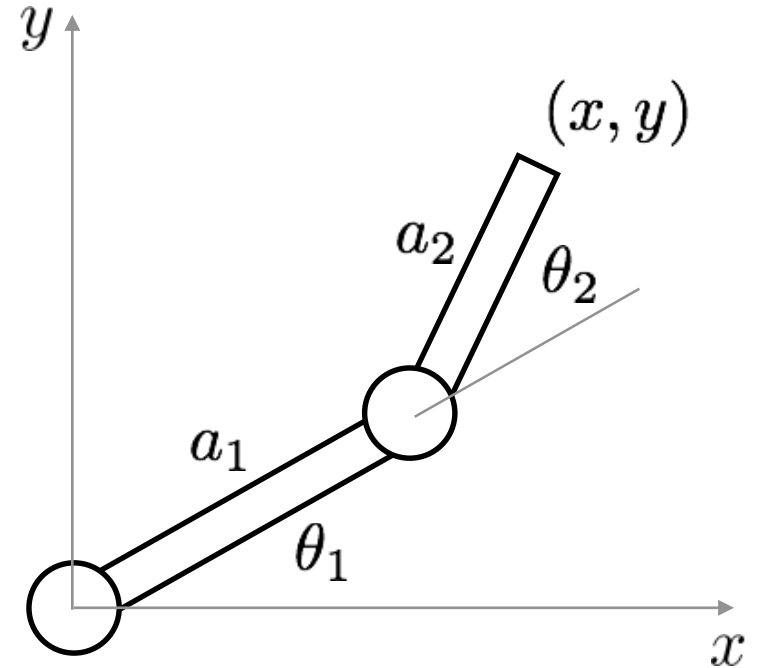
$$D = \sum_{i=1}^N (m_i J_{vci}^\top J_{vci} + J_{\omega i}^\top R_i I_i R_i^\top J_{\omega i})$$

$$g = \frac{\partial}{\partial q} \sum_{i=1}^N m_i \vec{g} \cdot \vec{r}_i$$

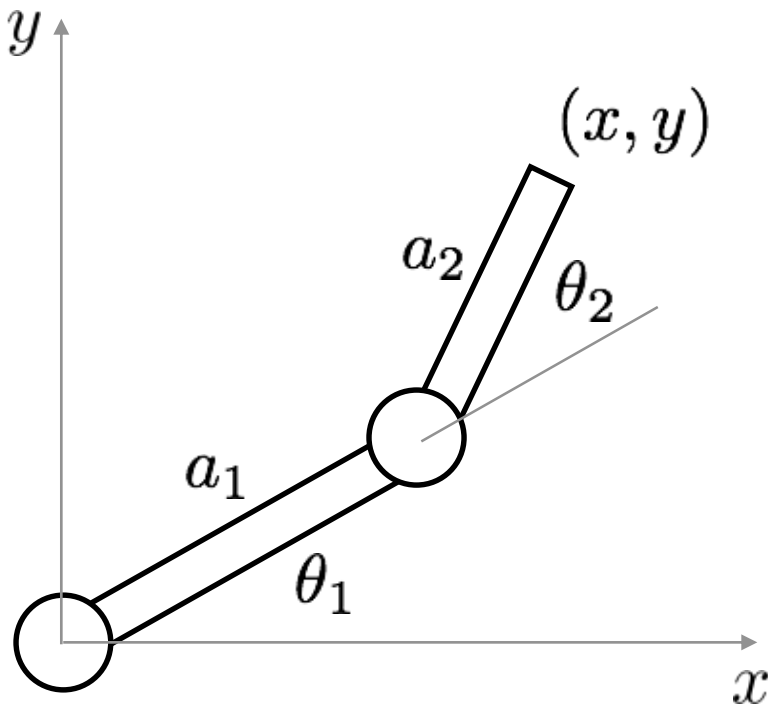
$$(C\dot{q})_k = \sum_{i,j} \frac{1}{2} \left(\frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k} \right) \dot{q}_i \dot{q}_j$$

or

$$c_{kj} = \sum_i \frac{1}{2} \left(\frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k} \right) \dot{q}_i$$



Example: Planar RR Manipulator



Angular velocity

$$\omega_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}$$

$J_{\omega 1}$

$$\omega_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}$$

$J_{\omega 2}$

Inertia

$$I_1 = \begin{bmatrix} 0 & I_{x_1 y_1} & 0 \\ I_{y_1 x_1} & 0 & 0 \\ 0 & 0 & I_{z_1 z_1} \end{bmatrix}$$

$$I_2 = \begin{bmatrix} 0 & I_{x_2 y_2} & 0 \\ I_{y_2 x_2} & 0 & 0 \\ 0 & 0 & I_{z_2 z_2} \end{bmatrix}$$

Rotation Matrix

$$R_1 = \begin{bmatrix} c_1 & s_1 & 0 \\ -s_1 & c_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_2 = \begin{bmatrix} c_{12} & s_{12} & 0 \\ -s_{12} & c_{12} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

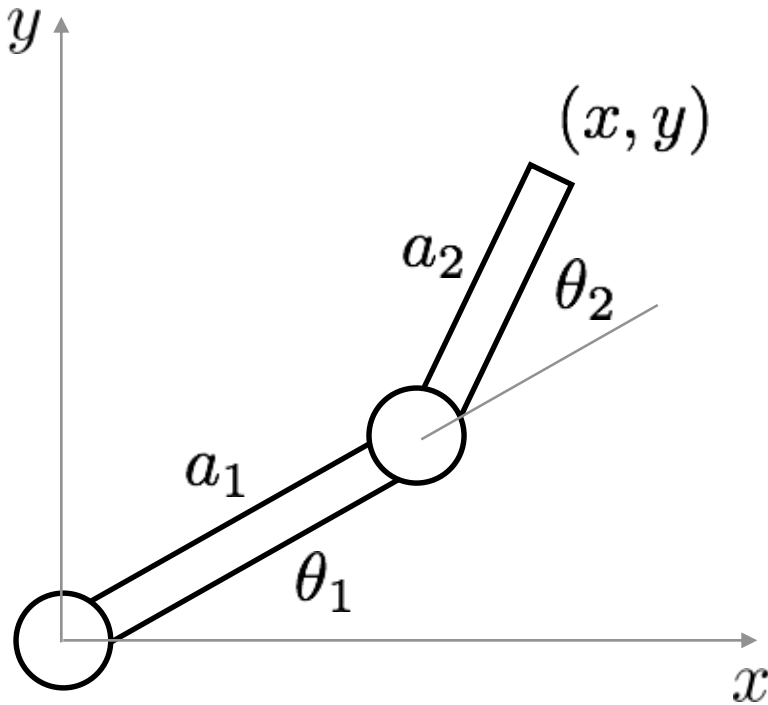
$$\text{Inertia: } D_{rot} = \sum_i J_{\omega i}^T R_i^0 I_i (R_i^0)^T J_{\omega i} = I_{z_1 z_1} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + I_{z_2 z_2} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

Example: Planar RR Manipulator

EOM

$$d_{11}\ddot{q}_1 + d_{12}\ddot{q}_2 + c_{121}\dot{q}_1\dot{q}_2 + c_{211}\dot{q}_2\dot{q}_1 + c_{221}\dot{q}_2^2 + g_1 = \tau_1$$

$$d_{21}\ddot{q}_2 + d_{22}\ddot{q}_2 + c_{112}\dot{q}_1^2 + g_2 = \tau_2$$



Inertia

$$d_{11} = m_1 a_{c,1}^2 + m_2 (a_1^2 + a_{c,2}^2 + 2a_1 a_{c,2} c_2) + I_{z_1 z_1} + I_{z_2 z_2}$$

$$d_{12} = d_{21} = m_2 (a_{c,2}^2 + a_1 a_2 c_2 + I_{z_2 z_2})$$

$$d_{22} = m_2 a_{c,2}^2 + I_{z_2 z_2}$$

Christoffel Symbols

$$c_{111} = c_{122} = c_{222} = 0$$

$$c_{121} = c_{221} = -c_{112} = -m_2 a_1 a_{c,2} s_2$$

Gravity

$$g_1 = (m_1 a_{c,1} + m_2 a_1) g c_1 + m_2 a_{c,2} g c_{12}$$

$$g_2 = m_2 a_{c,2} g c_{12}$$

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q)$$

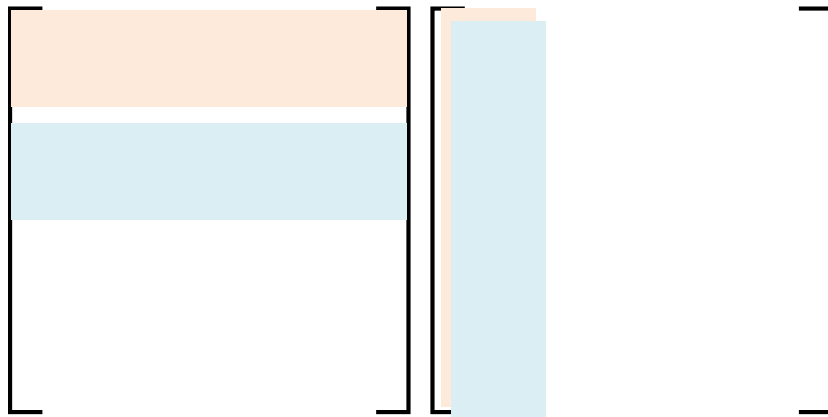
Intuition behind D

$$D = \sum_{i=1}^N \left(m_i \underbrace{J_{vci}^\top J_{vci}}_{\text{sum over links}} + \underbrace{J_{\omega i}^\top R_i I_i R_i^\top J_{\omega i}}_{\text{rotational inertia of the link in the correct frame}} \right)$$

Recall: the Jacobian is made of cross-products

The i th column of J_{vci} is $z_{i-1} \times (p_{ci} - o_{i-1})$

The product $J_{vci}^\top J_{vci}$ is a pairwise dot product between columns



Along the diagonal, we get $\|z_{i-1} \times (p_{ci} - o_{i-1})\|^2$
 i.e., (perpendicular distance to rotation axis)²
 i.e., parallel axis theorem

Off the diagonal, you can think of these terms as the resistance to motion due to another joint's acceleration

Properties: Passivity

We can show that

$$\int_0^T \dot{q}^\top \tau \, dt > -\beta$$

Total energy
dissipated over
time $[0, T]$ is lower-bounded!

Newton-Euler vs. Euler-Lagrange

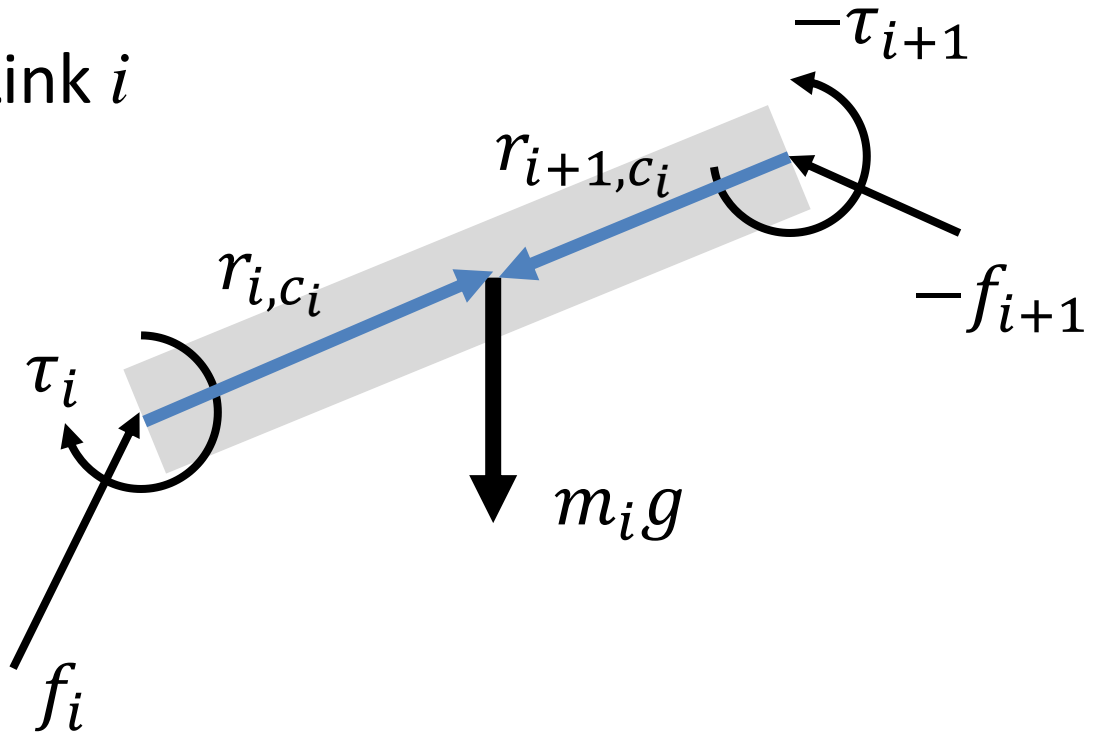
Euler-Lagrange produces a **closed-form** differential equation that describes the time evolution of the generalized coordinates

Computing these equations requires you to take partial derivatives

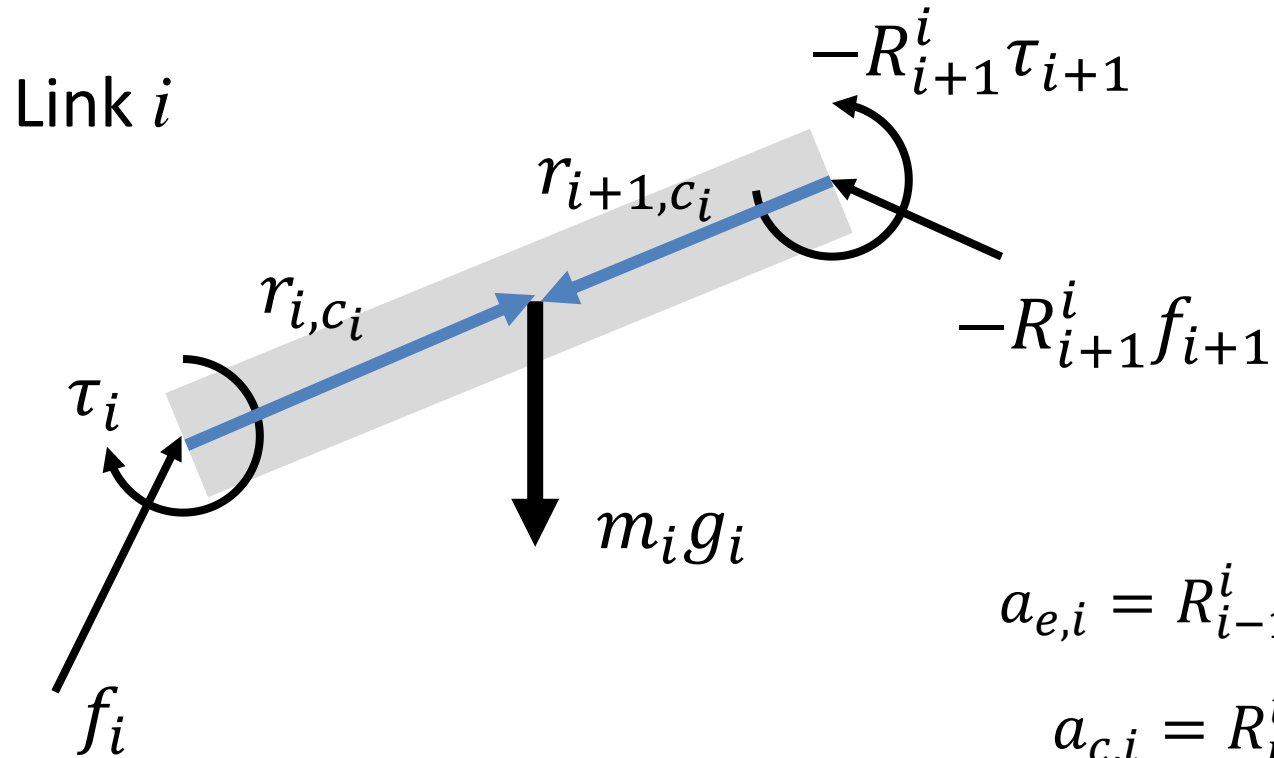
Newton-Euler allows you to do **recursive computation** to figure out the generalized forces for a particular time evolution

Newton-Euler

Link i



Newton-Euler (for revolute joints)



Kinematic constraints

$$\omega_i = R_{i-1}^i \omega_{i-1} + z_{i-1}^i \dot{q}_i$$

$$\alpha_i = R_{i-1}^i \alpha_{i-1} + z_{i-1}^i \ddot{q}_i + \omega_i \times z_{i-1}^i \dot{q}_i$$

$$a_{e,i} = R_{i-1}^i a_{e,i-1} + \alpha_i \times r_{i,i+1} + \omega_i \times (\omega_i \times r_{i,i+1})$$

$$a_{c,i} = R_{i-1}^i a_{e,i-1} + \alpha_i \times r_{i,c_i} + \omega_i \times (\omega_i \times r_{i,c_i})$$

Forces/Moments

$$f_i - R_{i+1}^i f_{i+1} + m_i g_i = m_i a_{c,i}$$

$$\tau_i - R_{i+1}^i \tau_{i+1} + f_i \times r_{i,c_i} - (R_{i+1}^i f_{i+1}) \times r_{i+1,c_i} = I_i \alpha_i + \omega_i \times (I_i \omega_i)$$

Newton-Euler (for revolute joints)

Start with $\omega_0 = 0, \alpha_0 = 0, a_{c,0} = 0, a_{e,0} = 0$

Solve kinematic constraints for i from 1 to n

Start with $f_{n+1} = 0, \tau_{n+1} = 0$

Solve force/moments for i from n to 1

Kinematic constraints

No forces/moments!

$$\left\{ \begin{array}{l} \omega_i = R_{i-1}^i \omega_{i-1} + z_{i-1}^i \dot{q}_i \\ \alpha_i = R_{i-1}^i \alpha_{i-1} + z_{i-1}^i \ddot{q}_i + \omega_i \times z_{i-1}^i \dot{q}_i \\ a_{e,i} = R_{i-1}^i a_{e,i-1} + \alpha_i \times r_{i,i+1} + \omega_i \times (\omega_i \times r_{i,i+1}) \\ a_{c,i} = R_{i-1}^i a_{e,i-1} + \alpha_i \times r_{i,c_i} + \omega_i \times (\omega_i \times r_{i,c_i}) \end{array} \right.$$

i terms on the left

$i-1$ terms on the right

Careful! All terms
written in frame i

Forces/Moments

$$f_i - R_{i+1}^i f_{i+1} + m_i g_i = m_i a_{c,i}$$

$$\tau_i - R_{i+1}^i \tau_{i+1} + f_i \times r_{i,c_i} - (R_{i+1}^i f_{i+1}) \times r_{i+1,c_i} = I_i \alpha_i + \omega_i \times (I_i \omega_i)$$

Example: Planar RR Manipulator

Forward Recursion

$$\omega_0 = 0, \alpha_0 = 0, a_{e,0} = 0, a_{c,0} = 0$$

$$\omega_1 = \dot{q}_1 \mathbf{z}$$

$$\alpha_1 = \ddot{q}_1 \mathbf{z}$$

$$a_{e,1} = \ddot{q}_1 \mathbf{z} \times l_1 \mathbf{x} + \dot{q}_1 \mathbf{z} \times (\dot{q}_1 \mathbf{z} \times l_1 \mathbf{x}) = [-l_1 \dot{q}_1^2 \quad l_1 \ddot{q}_1 \quad 0]^\top$$

$$a_{c,1} = [-l_{c,1} \dot{q}_1^2 \quad l_{c,1} \ddot{q}_1 \quad 0]^\top$$

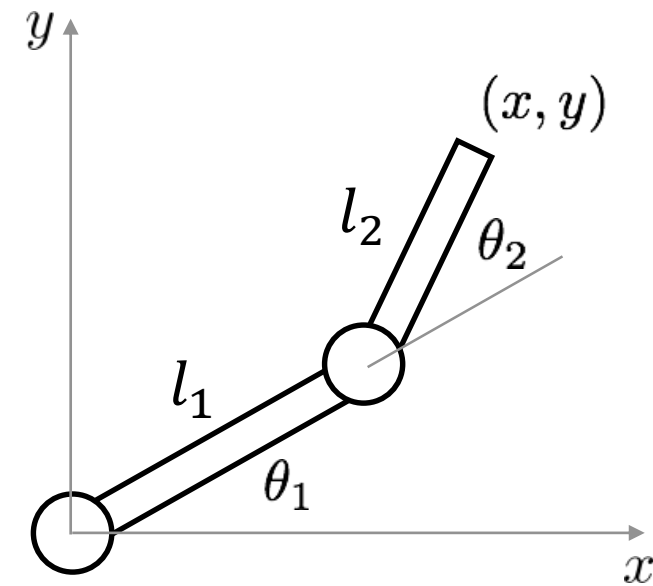
$$\omega_2 = (\dot{q}_1 + \dot{q}_2) \mathbf{z}$$

$$\alpha_2 = (\ddot{q}_1 + \ddot{q}_2) \mathbf{z}$$

$$a_{e,2} = R_1^2 a_{e,1} + (\ddot{q}_1 + \ddot{q}_2) \mathbf{z} \times l_2 \mathbf{x} + (\dot{q}_1 + \dot{q}_2) \mathbf{z} \times ((\dot{q}_1 + \dot{q}_2) \mathbf{z} \times l_2 \mathbf{x})$$

$$= [-l_1 \dot{q}_1^2 c_2 + l_1 \ddot{q}_1 + s_2 - l_{c2} (\dot{q}_1 + \dot{q}_2)^2 \quad l_1 \dot{q}_1^2 s_2 + l_1 \ddot{q}_1 c_2 - l_2 (\ddot{q}_1 + \ddot{q}_2) \quad 0]^\top$$

$$a_{c,2} = [-l_1 \dot{q}_1^2 c_2 + l_1 \ddot{q}_1 + s_2 - l_{c,2} (\dot{q}_1 + \dot{q}_2)^2 \quad l_1 \dot{q}_1^2 s_2 + l_1 \ddot{q}_1 c_2 - l_{c,2} (\ddot{q}_1 + \ddot{q}_2) \quad 0]^\top$$



Example: Planar RR Manipulator

Backward Recursion

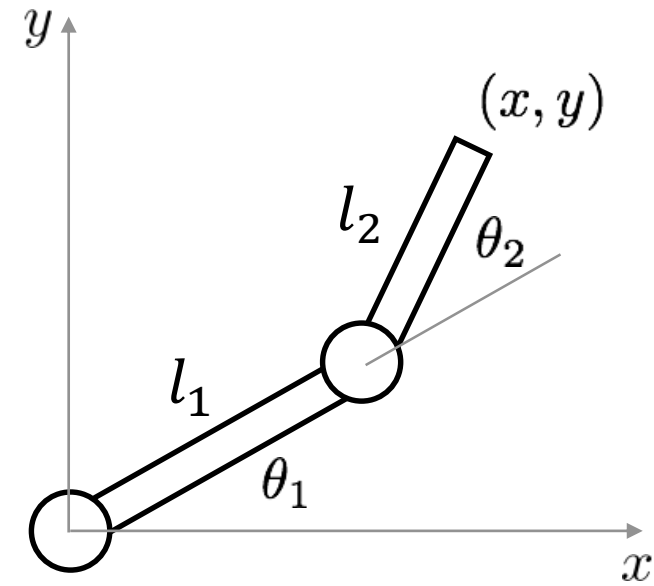
$$f_3 = 0, \tau_3 = 0$$

$$f_2 = m_2 a_{c,2} \boxed{- m_2 g_2}$$

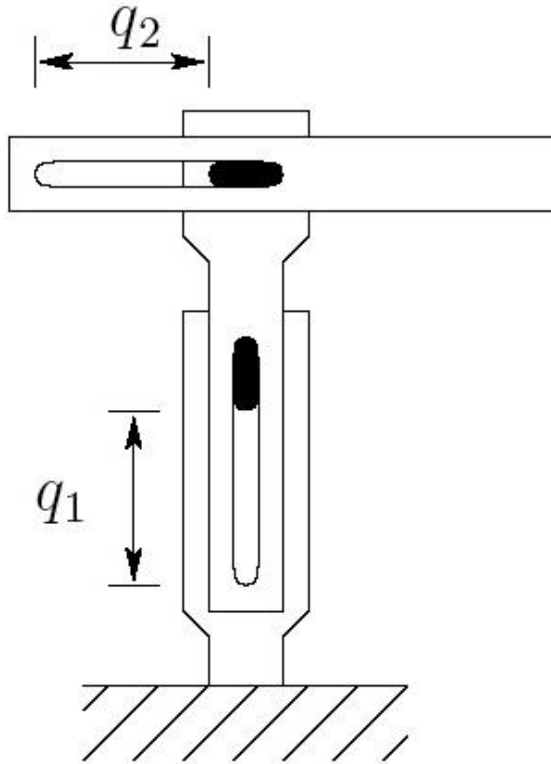
$$\tau_2 = -f_2 \times l_{c,2} \mathbf{x} + I_2 \alpha_2 + \omega_2 \times (I_2 \omega_2)$$

$$f_1 = m_1 a_{c,1} + R_2^1 f_2 \boxed{- m_1 g_1}$$

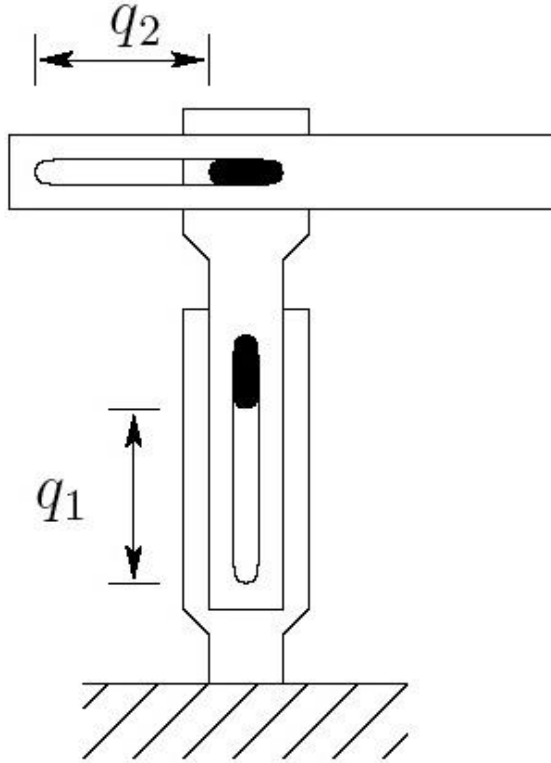
$$\tau_1 = R_2^1 \tau_2 - f_1 \times l_{c,1} \mathbf{x} - R_2^1 f_2 \times (l_1 - l_{c,1}) \mathbf{x} + I_1 \alpha_1 + \omega_1 \times (I_1 \omega_1)$$



Example: 2-Link Cartesian Manipulator



Example: 2-Link Cartesian Manipulator



$$J_{vc_1} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$$

$$J_{vc_2} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$D = m_1 J_{vc_1}^T J_{vc_1} + m_2 J_{vc_2}^T J_{vc_2}$$

No rotational components!

$$D = \begin{bmatrix} m_1 + m_2 & 0 \\ 0 & m_2 \end{bmatrix}$$

$$C = 0$$

$$P = g(m_1 + m_2)q_1 \quad \Rightarrow \quad \frac{\partial P}{\partial q} = \begin{bmatrix} g(m_1 + m_2) \\ 0 \end{bmatrix}$$

$$(m_1 + m_2)\ddot{q}_1 + g(m_1 + m_2) = \tau_1$$

$$m_2\ddot{q}_2 = \tau_2$$

Method Comparisons

Newton-Euler

- Complete solution for all forces and kinematic variables
- Inefficient when only a few of the system's forces need to be solved for

Euler-Lagrange

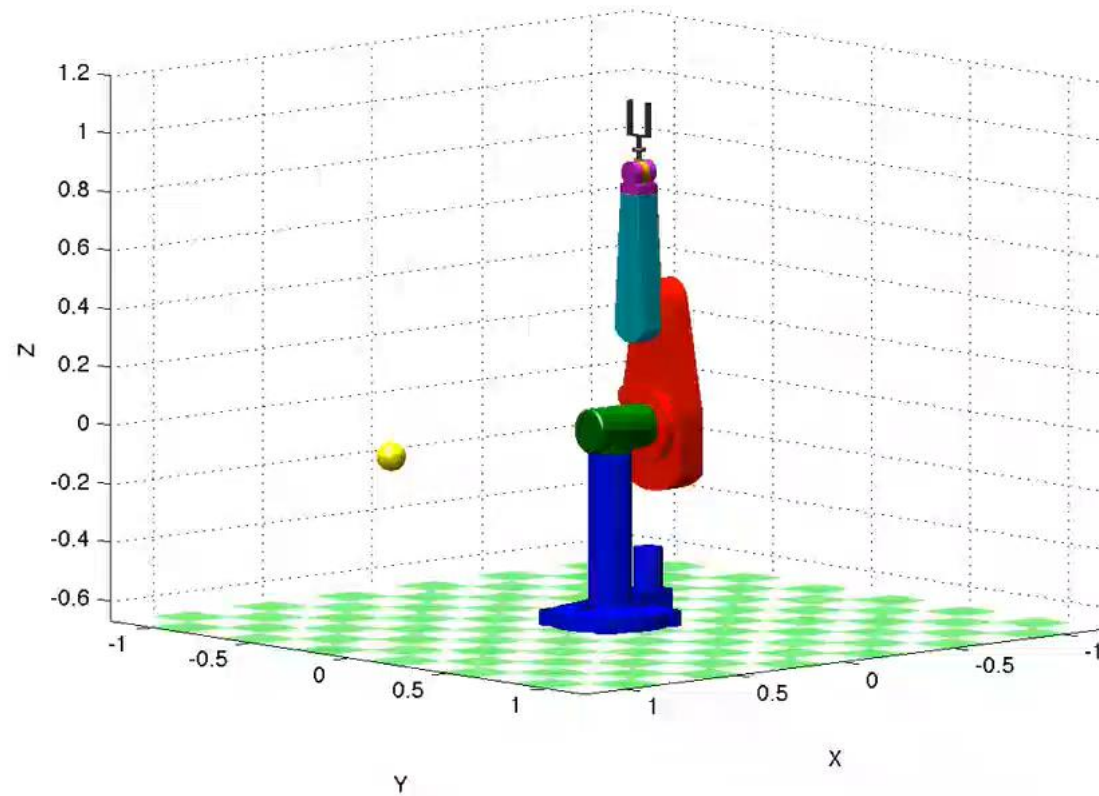
- Disregard all interactive and constraint forces that do not perform work
- Need to differentiate scalar energy functions
- Inefficient for large multi-body systems

Kane's Method

- Generalized forces so eliminate interactive and constraint forces
- Does not employ energy functions, so no derivatives
- Lends itself to automated numerical computation

Check out the Robotics Toolbox

<http://petercorke.com/wordpress/toolboxes/robotics-toolbox>



Peter Corke
Professor
Queensland University of Technology

Next time: Robotic Operating System



Lab 5: Potential Field Planning

MEAM 520, University of Pennsylvania

November 6, 2020

This lab consists of two portions, with a pre-lab due on Friday, November 13, by midnight (11:59 p.m.) and a lab (code+report) due on Friday, November 20, by midnight (11:59 p.m.). Late submissions will be accepted until midnight on Saturday following the deadline, but they will be penalized by 25% for each partial or full day late. After the late deadline, no further assignments may be submitted; post a private message on Piazza to request an extension if you need one due to a special situation. This assignment is worth 50 points.

You may talk with other students about this assignment, ask the teaching team questions, use a calculator and other tools, and consult outside sources such as the Internet. To help you actually learn the material, what you submit must be your own work, not copied from any other individual or team. Any submissions suspected of violating Penn's Code of Academic Integrity will be reported to the Office of Student Conduct. When you get stuck, post a question on Piazza or go to office hours!

Individual vs. Pair Programming

Work closely with your partner throughout the lab, following these guidelines, which were adapted from "All I really needed to know about pair programming I learned in kindergarten," by Williams and Kessler, *Communications of the ACM*, May 2000. This article is available on Canvas under Files / Resources.

- Start with a good attitude, setting aside any skepticism, and expect to jell with your partner.
- Don't start alone. Arrange a meeting with your partner as soon as you can.
- Use just one setup, and sit side by side. For a programming component, a desktop computer with a large monitor is better than a laptop. Make sure both partners can see the screen.
- At each instant, one partner should be driving (writing, using the mouse/keyboard, moving the robot) while the other is continuously reviewing the work (thinking and making suggestions).
- Change driving/reviewing roles at least every 30 minutes, even if one partner is much more experienced than the other. You may want to set a timer to help you remember to switch.
- If you notice an error in the equation or code that your partner is writing, wait until they finish the line to correct them.
- Stay focused and on-task the whole time you are working together.
- Take a break periodically to refresh your perspective.
- Share responsibility for your project; avoid blaming either partner for challenges you run into.
- Recognize that working in pairs usually takes more time than working alone, but it produces better work, deeper learning, and a more positive experience for the participants.

1

Lab 5 due 11/20