Articulated Body Animation
- Dynamics
Topics

- Model
- Forward Kinematics
- Inverse Kinematics
- **Forward Dynamics**
- **Inverse Dynamics**
- Low-level path planning (not in this course)
- High-level scripting and action control (not in this course)
Taxonomy

Articulated Simulation

Kinematics
  - Position Kinematics
    - Forward
    - Inverse
  - Velocity Kinematics
    - Forward
    - Backward

Dynamics
Dynamics vs. Kinematics

- Kinematics deal with position and velocity
- Dynamics deal with force and torque
- You can go a long way with kinematics simulation
- You can do more with dynamics simulation
  - Robotics: maintain constant pressure of the end effector
  - Graphics: collision of two objects (transfer of force and torque)
Need to Know: Theory

- Newton’s formulation
  - Govern translation motion
- Euler’s formulation
  - Govern rotational motion
- Newton-Euler formulation
  - Both translation and rotation
- Lagrange formulation
  - Using generalized forces/coordinates
- Transferring force and torque representation over multiple linkages (both Newton-Euler and Lagrange)
Need to Know: Practice

- Static
- Dynamic
Need to Know: Practice

- Given the torque or force at the end effector & Robot’s current configuration
  - Static: to maintain the position of a load
  - Dynamic: to move a welding gun or to exert constant pressure on a surface
- What are the force and torque needed to apply to each joint?
- An inverse dynamics problem
Progression Roadmap

- From particle
  - Newton’s law, position only
- To rigid body
  - Newton + Euler, position + velocity
- To generalized representation
  - Lagrange
- To multiple linkages
  - Newton-Euler, or
  - Lagrange
- Btw, another possibility is particle to rigid body to deformable body (later)
## Particle vs. Rigid Body

<table>
<thead>
<tr>
<th>Particle(s)</th>
<th>Rigid body(-ies)</th>
</tr>
</thead>
<tbody>
<tr>
<td>do not occupy space</td>
<td>occupy space</td>
</tr>
<tr>
<td>position only</td>
<td>position + orientation</td>
</tr>
<tr>
<td>state vector: position + linear momentum</td>
<td>state vector: position + linear momentum, orientation + angular momentum</td>
</tr>
<tr>
<td>velocity is related to position by derivative</td>
<td>angular velocity is related to orientation by derivative</td>
</tr>
<tr>
<td>ODE: force induces a change of linear momentum</td>
<td>ODE: force induces a change in linear momentum, torque induces a change in angular momentum</td>
</tr>
<tr>
<td>Collision: instantaneous change of linear momentum</td>
<td>Collision: instantaneous change of both linear and angular momentum</td>
</tr>
</tbody>
</table>

*Computer Graphics*
Translation

- **Particles**
- **Mass**
- **Position**
- **Velocity**
- **Force**

\[ M = \sum_i m_i \]
\[ \mathbf{X} = \frac{\sum_i m_i \mathbf{x}_i}{\sum_i m_i} \]
\[ \mathbf{V} = \frac{\sum_i m_i \mathbf{v}_i}{\sum_i m_i} = \frac{\sum_i m_i \mathbf{\dot{x}}_i}{\sum_i m_i} \]
\[ \mathbf{F} = \sum_i \mathbf{F}_i \]

- **Rigid Body**
- **Mass**
- **Position**
- **Velocity**
- **Force**

- Total mass of the object
- The centroid position
- The centroid velocity
- The total force
Rotation

- Particles
- Nothing like it for particles

- Rigid bodies
- Moment of inertia
- Orientation
- Angular velocity
- Angular momentum
- Torque
Rotation (cont.)

- Rigid bodies
- Mass
- Position
- Velocity
- Linear momentum
- Force
- Rigid bodies
- Moment of inertia
- Orientation
- Angular velocity
- Angular momentum
- Torque
Equations of Motion

- Translational motion: $F = ma = \frac{d(mv)}{dt}$ (Newton)
  - If you don’t know this one, then cannot help you 😊
- Rotational motion: (Euler) $\tau = \frac{dIw}{dt} = I\dot{w} + \omega \times Iw$
  - Torque = moment of inertia * $\dot{\text{angular vel}}$ is not it!
  - Still in Cartesian coordinate system
- More general: (Lagrange) $f_i = \frac{d}{dt} \left( \frac{\partial(K-V)}{\partial\dot{q_i}} \right) - \frac{\partial(K-V)}{\partial q_i}$
  - Generalized coordinates + generalized forces
  - Using variational principle
Derivation of Euler Eq.

\[ I_{\text{body}} = R^T \dot{I} R \]

\[ \dot{I}_{\text{body}} = \dot{R}^T I R + R^T \dot{I} R + R^T \dot{I} \dot{R} = 0 \]

\[ R^T \dot{I} R = -\dot{R}^T I R - R^T I \dot{R} \]

\[ R(R^T \dot{I} R)R^T = R(-\dot{R}^T I R - R^T I \dot{R})R^T \]

\[ \dot{I} = -R \dot{R}^T I - I \dot{R} R^T \]

\[ \dot{I} = -R(\Omega R)^T I - R(\Omega R)R^T \]

\[ \dot{I} = -\Omega^T I - I \Omega \]

\[ \dot{I} = \Omega I - I \Omega \]

\[ \tau = \frac{d(Iw)}{dt} = \dot{I} w + I \dot{w} = (\Omega I - I \Omega)w + I \dot{w} = I \dot{w} + \Omega I w = I \dot{w} + \omega \times I w \]
**Lagrange Formulation**

- Usually derived from variational principle
- Functional $\int (K-V)$ (generalized kinetic energy and generalized potential energy) to be minimized
- We will derive it using virtual work principle
**Generalized Coords/Forces**

- So far, all derivations are in Cartesian coordinates
- Generalized coordinates (usually) refer to the system’s DOFs
- E.g.,
  - Revolute joint – rotation angle
  - Prismatic join – translation motion
- Generalized force is the force (or torque) applied to a generalized coordinate
**Example:**

**Generalized Coordinate**

\[
\begin{align*}
\begin{bmatrix}
    x_1 \\
y_1
\end{bmatrix} &= \begin{bmatrix}
x_o \\
y_o
\end{bmatrix} + \begin{bmatrix}
l_1 \cos \theta_1 \\
l_1 \sin \theta_1
\end{bmatrix}
\]

\[
\begin{align*}
\begin{bmatrix}
x_2 \\
y_2
\end{bmatrix} &= \begin{bmatrix}
x_1 \\
y_1
\end{bmatrix} + \begin{bmatrix}
l_2 \cos(\theta_1 + \theta_2) \\
l_2 \sin(\theta_1 + \theta_2)
\end{bmatrix} = \begin{bmatrix}
l_1 \cos \theta_1 + l_2 \cos(\theta_1 - \theta_2) \\
l_1 \sin \theta_1 + l_2 \sin(\theta_1 - \theta_2)
\end{bmatrix} \Rightarrow x_i = f_i(\theta_1, \theta_2)
\end{align*}
\]

\[
\begin{align*}
\begin{bmatrix}
    \dot{x}_2 \\
    \dot{y}_2
\end{bmatrix} &= \begin{bmatrix}
    -l_1 \sin \theta_1 \dot{\theta}_1 - l_2 \sin(\theta_1 - \theta_2)(\dot{\theta}_1 - \dot{\theta}_2) \\
l_1 \cos \theta_1 \dot{\theta}_1 + l_2 \cos(\theta_1 - \theta_2)(\dot{\theta}_1 - \dot{\theta}_2)
\end{bmatrix} \Rightarrow \dot{x}_i = f_i(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2)
\end{align*}
\]
Generalized Coordinates

- End effector position as a function of DOFs
- Choice is not unique
- Freedom in choice is good

In robotics

- Use link variables (for position and velocity)
  - Prismatic joint – displacement
  - Revolute joint – angle
- their derivatives (for velocity)
Why Generalized Coors/Forces?

- A link-centered representation
  - Center of mass
  - Coordinate system
  - Force
  - Torque
- A DOF-based representation
  - One DOF
  - One variable (angle for revolute or displacement for prismatic)
  - One generalized force (torque for revolute and force for prismatic, all other components do not matter)
A Generalized Force

- The force that applies to a generalized coordinate
- Work relation in Cartesian coordinate = force * displacement
- Work relation in DOF coordinate = generalized force * generalized coordinate
- Work done is the same in both representations!!!
- The system behaves the same in either representation
- Note: generalized force can be either force or torque, depending on the DOF
\[ \tau_1 \delta \theta_1 + \tau_2 \delta \theta_2 = \mathbf{f}_{c_1} \delta \mathbf{r}_{c_1} + \mathbf{f}_{c_2} \delta \mathbf{r}_{c_2} \]

\[ \tau_1 = \mathbf{f}_{c_1} \frac{\partial \mathbf{r}_{c_1}}{\partial \theta_1} + \mathbf{f}_{c_2} \frac{\partial \mathbf{r}_{c_2}}{\partial \theta_1} \]

\[ \tau_2 = \mathbf{f}_{c_1} \frac{\partial \mathbf{r}_{c_1}}{\partial \theta_2} + \mathbf{f}_{c_2} \frac{\partial \mathbf{r}_{c_2}}{\partial \theta_2} \]

\[ \tau_i = \sum_{j=1}^{n} \mathbf{f}_{c_j} \frac{\partial \mathbf{r}_{c_j}}{\partial \theta_{ij}} \]

\[ \tau_i \delta \theta_i = \sum_{j=1}^{n} \mathbf{f}_{c_j} \delta \mathbf{r}_{c_j} \]

- Work expressions are \textit{identical} in either Cartesian or generalized coordinates
- Collect all forces and all displacements (or all work) done by change of generalized coordinate

\textit{Intuition}
Virtual Work Principle

- Force does not (necessarily) cause motion
- Energy (work) cause motion
- Energy
  - Internal (for rigid object internal=0)
  - External (applied + potential)
  - Inertia (think of \( ma \))
- For any compatible virtual displacement (or deformation) must satisfy
  - External work = Internal work (resting condition)
  - External work = Inertia work (moving condition)
Derivation of Lagrange Eq.

- Virtual Displacement
- External
  - Work done by potential energy
  - Work done by external forces
- Internal
  - Work done by inertia force
- External work = Inertia work
- Assuming a system of $n$ particles and $n$ DOFs (generalized coordinates)
Displacement and Velocity

\[ \mathbf{r}_i = \mathbf{r}_i(q_1, q_2, \cdots, q_n) \]

\[ \delta \mathbf{r}_i = \sum_{j=1}^{n} \frac{\partial \mathbf{r}_i}{\partial q_j} \delta q_j \quad \text{or} \quad \delta \mathbf{r}_i = \sum_{j=1}^{n} \frac{\partial \mathbf{r}_i}{\partial \dot{q}_j} \delta \dot{q}_j \]

\[ \mathbf{v}_i = \mathbf{r}_i = \sum_{j=1}^{n} \frac{\partial \mathbf{r}_i}{\partial q_j} \dot{q}_j \quad \text{or} \quad \mathbf{v}_i = \mathbf{r}_i = \sum_{j=1}^{n} \frac{\partial \mathbf{r}_i}{\partial \dot{q}_j} \dot{q}_j \]

\[ \Rightarrow \frac{\partial \mathbf{r}_i}{\partial \dot{q}_j} = \frac{\partial}{\partial \dot{q}_j} \sum_{k=1}^{n} \frac{\partial \mathbf{r}_i}{\partial q_k} \dot{q}_k = \frac{\partial \mathbf{r}_i}{\partial q_j} \]
Work by External Force

- Potential

\[ V(q_1, q_2, \ldots, q_n) = \sum_{i=1}^{n} m_i g^t \dot{r}_i(q_1, q_2, \ldots, q_n) \]

\[ \delta W_v = -\delta V = -\sum_{i=1}^{n} \frac{\partial V}{\partial q_i} \delta q_i \]

\[ \frac{\partial V}{\partial \dot{q}_i} = 0 \]

\[ \delta W_f = \delta W_v + \delta W_\tau = \sum_{i=1}^{n} \left( -\frac{\partial V}{\partial q_i} + \tau_i \right) \delta q_i \]

- Other

\[ \delta W_\tau = \sum_{i=1}^{n} \tau_i \delta q_i \]

Generalized force
Work by Inertia Force

\[ \delta W_k = \sum_{i=1}^{n} F_i \delta r_i = \sum_{i=1}^{n} m_i \dot{v}_i \delta r_i = \sum_{i=1}^{n} m_i \frac{d(v_i \delta r_i)}{dt} - \sum_{i=1}^{n} m_i v_i \delta r_i \]

\[ = \sum_{i=1}^{n} \left( \frac{d}{dt} \left( \frac{\partial K}{\partial \dot{q}_i} \right) - \frac{\partial K}{\partial q_i} \right) \delta q_i \quad \text{where} \quad K = \sum_{i} \frac{m_i}{2} v_i v_i \]

- Detail can be found in most robotics and mechanics books
- If you are really interested, check out the following slides
Really Ugly Details – term A

\[
\delta W_k = \sum_{i=1}^{n} m_i \frac{d (v_i \delta r_i)}{dt} - \sum_{i=1}^{n} m_i v_i \delta \dot{r}_i
\]

\[
A = \sum_{i=1}^{n} \left( m_i \cdot \frac{d}{dt} \left( \sum_{j=1}^{n} \frac{\partial \dot{r}_i}{\partial \dot{q}_j} \delta q_j \right) \right) \cdot \delta q_j.
\]

\[
A = \sum_{j=1}^{n} \frac{d}{dt} \left( \sum_{i=1}^{n} m_i \cdot v_i \cdot \frac{\partial \dot{r}_i}{\partial \dot{q}_j} \right) \cdot \delta q_j.
\]

\[
A = \sum_{j=1}^{n} \frac{d}{dt} \left( \frac{\partial K}{\partial \dot{q}_j} \right) \cdot \delta q_j.
\]

\[
\delta \dot{r}_i = \sum_{j=1}^{n} \frac{\partial \dot{r}_i}{\partial \dot{q}_j} \delta q_j,
\]

\[
\frac{\partial K}{\partial q_j} = \sum_{i=1}^{n} m_i \cdot v_i^t \cdot \frac{\partial v_i}{\partial q_j}, \quad \forall j \in [1, n]
\]

\[
\frac{\partial K}{\partial q_j} = \sum_{i=1}^{n} m_i \cdot v_i^t \cdot \frac{\partial v_i}{\partial q_j}, \quad \forall j \in [1, n].
\]
Really Ugly Details – term B

\[ \delta W_k = \sum_{i=1}^{n} m_i \frac{d (v_i \delta r_i)}{dt} - \sum_{i=1}^{n} m_i v_i \delta \dot{r}_i \]

\[ B = \sum_{i=1}^{n} \left\{ m_i \cdot v_i \cdot \sum_{j=1}^{n} \left( \frac{\partial v_i}{\partial q_j} \cdot \delta q_j \right) \right\} \]

\[ B = \sum_{j=1}^{n} \left\{ \sum_{i=1}^{n} \left( m_i \cdot v_i \cdot \frac{\partial v_i}{\partial q_j} \right) \cdot \delta q_j \right\} \]

\[ B = \sum_{j=1}^{n} \left( \frac{\partial K}{\partial q_j} \cdot \delta q_j \right) \]

\[ B = \sum_{i=1}^{n} \left( \frac{\partial K}{\partial q_i} \cdot \delta q_i \right) \]

\[ \delta v_i = \delta \dot{r}_i = \sum_{j=1}^{n} \left( \frac{\partial v_i}{\partial q_j} \cdot \delta q_j \right) \]

\[ \frac{\partial K}{\partial q_j} = \sum_{i=1}^{n} m_i \cdot v_i^t \cdot \frac{\partial v_i}{\partial q_j}, \quad \forall j \in [1, n] \]

\[ \frac{\partial K}{\partial q_j} = \sum_{i=1}^{n} m_i \cdot v_i^t \cdot \frac{\partial v_i}{\partial q_j}, \quad \forall j \in [1, n]. \]
Really Ugly Detail - combined

\[ \delta W_k = |A - B| = \sum_{i=1}^{n} \left\{ \left[ \frac{d}{dt} \left( \frac{\partial K}{\partial \dot{q}_i} \right) - \frac{\partial K}{\partial q_i} \right] \cdot \delta q_i \right\} \]
Putting it all together

\[
\sum_{i=1}^{n} \left( \frac{d}{dt} \frac{\partial K}{\partial \dot{q}_i} - \frac{\partial K}{\partial q_i} \right) \delta q_i = \sum_{i=1}^{n} \left( -\frac{\partial V}{\partial q_i} + f_i \right) \delta q_i
\]

\[
\frac{\partial V}{\partial q_i} + f_i = \frac{d}{dt} \frac{\partial K}{\partial \dot{q}_i} - \frac{\partial K}{\partial q_i}
\]

\[
\Rightarrow f_i = \frac{d}{dt} \frac{\partial K}{\partial \dot{q}_i} - \frac{d}{dt} \frac{\partial V}{\partial \dot{q}_i} - \frac{\partial K}{\partial q_i} + \frac{\partial V}{\partial q_i}
\]

\[
\Rightarrow f_i = \frac{d}{dt} \frac{\partial (K-V)}{\partial \dot{q}_i} - \frac{\partial (K-V)}{\partial q_i}
\]
Mother of All Motion Eqs.

\[ f_i = \frac{d}{dt} \frac{\partial (K - V)}{\partial \dot{q}_i} - \frac{\partial (K - V)}{\partial q_i} \]

- Whether you follow the derivation or not
- Lagrange can be used to derive motion eqs for
  - Particles (Newton)
  - Rigid body (Euler)
- Advantage
  - It is much easier to figure out energy (scalar) than figure out force (vector)
  - Valid for all cases and all parameterizations
Lagrange to Newton

\[ K = \frac{1}{2} m(\dot{x}, \dot{y}) \cdot (\dot{x}, \dot{y}) \]

\[ V = mgy \]

\[ L = K - V = \frac{1}{2} m(\dot{x}, \dot{y}) \cdot (\dot{x}, \dot{y}) - mgy \]

\[ f_i = \frac{d}{dt} \frac{\partial (K - V)}{\partial \dot{q_i}} - \frac{\partial (K - V)}{\partial q_i} \]

\[ f_x = \frac{d}{dt} \frac{\partial L}{\partial \dot{x}_i} - \frac{\partial L}{\partial x_i} = \frac{d}{dt} m\ddot{x} = m\dddot{x} \]

\[ f_y = \frac{d}{dt} \frac{\partial L}{\partial \dot{y}_i} - \frac{\partial L}{\partial y_i} = \frac{d}{dt} m\ddot{y} - mg = m\dddot{y} - mg \]
Lagrange to Euler

\[ f_i = \frac{\partial (K - V)}{\partial q_i} - \frac{\partial (K - V)}{\partial \dot{q}_i} \]

\[ K = \frac{1}{2} mv^T v_c + \frac{1}{2} w^T IW = \frac{1}{2} mv^T v_c + \frac{1}{2} w^T RI_{body} R^T w \]

\[ V = mg^T r_c \]

\[ f_x = m\ddot{x} \]

\[ f_y = m\ddot{y} - mg \]

\[ \tau = \frac{d}{dt} \left( \frac{\partial w^T IW}{\partial \theta} \right) - \frac{\partial}{\partial \theta} \left( \frac{1}{2} w^T IW \right) = \frac{d}{dt} (IW) - \frac{\partial}{\partial \theta} \left( \frac{1}{2} w^T IW \right) = \frac{d}{dt} (IW) \]

\[ \frac{\partial w^T IW}{\partial \theta} = w^T \frac{\partial R}{\partial \theta} I_{body} R^T w + w^T RI_{body} \frac{\partial R^T}{\partial \theta} w \]

\[ = w^T (w \times R) I_{body} R w + w^T R I_{body} (w \times R)^T w = 0 \]

Just like translational energy is not a function of position, rotational energy is not a function of orientation.
Self Serving

- That is, if you buy the energy expressions
- But where do the energy expressions come from?
Multiple Links

- Solutions can be derived either using Lagrange or Newton-Euler formulation.
- Newton-Euler results in an iterative formulation that can be solved efficiently using computer.
- Lagrange results in a big matrix problem (usually diagonal and sparse).
Newton-Euler Analysis Methodology

- Robot arm as rigid bodies, satisfying
  - Single link force and torque constraints
  - Multiple link force and torque constraints
**Newton-Euler**

- $O_i$: origin of link $i$'s frame
- $R_i$: rotation matrix of orientation of link $i$'s frame
- $r_{ci}$: center of gravity of link $i$
- $v_i$: linear velocity of link $i$'s frame
- $a_i$: linear acceleration of link $i$'s frame
- $w_i$: angular velocity of link $i$'s frame
- $a_i$: angular acceleration of link $i$'s frame

[Diagram showing links, joints, and coordinate systems]
**Newton-Euler**

- $m_i$: mass of link $i$
- $I_{i}^{body}$: inertia matrix in body frame
- $I_{i}^{global}$: inertia matrix in global frame
- $f_i$: force from joint $i$ to link $i$
- $-f_{i+1}$: reaction force from joint $i+1$ to link $i$
- $\tau_i$: torque from joint $i$ to link $i$
- $-\tau_{i+1}$: reaction torque from joint $i+1$ to link $i$
Graphic Interpretation for link i

Joint i
Link i
Coor i
Coor i-1
Joint i+1

Joint i
Link i
Coor i
Coor i-1
Joint i+1

Joint i
Link i
Coor i
Coor i-1
Joint i+1

Joint i
Link i
Coor i
Coor i-1
Joint i+1

f_i
r_{ci}
-f_{i+1}

\tau_i
\tau_{i+1}
Newton-Euler Steps

- Sum of forces
- Sum of torques
- Equation of linear motion
- Equation of angular motion
**Forces and Torques**

- **Sum of forces**
  - From joint i
  - From joint i+1
  - Gravity

\[
F_i = f_i - f_{i+1} + m_i g
\]

- **Sum of torques**
  - From joint i
  - From joint i+1
  - From force of joint i
  - From force of joint i+1

\[
T_i = \tau_i - \tau_{i+1} + r_{ci}^{i-1} \times f_i - r_{ci}^i \times f_{i+1}
\]

\[
r_{ci}^{i-1} = O_{i-1} - r_{ci}^i
\]

\[
r_{ci}^i = O_i - r_{ci}
\]

- $r_{ci}$: center of mass of link i
- $O_i$: origin of reference frame i
Forces and Torques

\[ F_i = f_i - f_{i+1} + m_i g \]
Forces and Torques (cont)

\[ T_i = \tau_i - \tau_{i+1} + r_{ci}^{i-1} \times f_i - r_{ci}^i \times f_{i+1} \]

\[ r_{ci}^{i-1} = O_{i-1} - r_{ci} \]

\[ r_{ci}^i = O_i - r_{ci} \]
Linear Motion

\[ F_i = f_i - f_{i+1} + m_i g = m_i a_{ci} \]

\[ = m_i \left( a_i - \alpha_i \times r_{ci}^i - w_i \times w_i \times r_{ci}^i \right) \]

\[ v_{ci} = \frac{dr_{ci}}{dt} = \frac{dO_i}{dt} - \frac{dr_{ci}^i}{dt} = v_i - w_i \times r_{ci}^i \]

\[ a_{ci} = \frac{dv_{ci}}{dt} = \frac{dv_i}{dt} - \frac{dw_i}{dt} \times r_{ci}^i - w_i \times \frac{dr_{ci}^i}{dt} \]

\[ = a_i - \alpha_i \times r_{ci}^i - w_i \times w_i \times r_{ci}^i \]
Angular Motion

\[ T_i = \tau_i - \tau_{i+1} + r_{i-1}^0 \times f_i - r_i^0 \times f_{i+1} = I_i \alpha_i + \frac{dI_i}{dt} w_i \]

\[ = I_i \alpha_i + w_i \times (I_i w_i) \]

\[ \therefore \tau = \frac{d(Iw)}{dt} = \dot{I}w + I\dot{w} = I\ddot{w} + \omega \times Iw \]
Recursive Algorithm

- Backward recursion from end effector to base

\[ \mathbf{f}_i = \mathbf{f}_{i+1} + m_i \left( \mathbf{a}_i - \mathbf{a}_i \times \mathbf{r}_{ci}^i - \mathbf{w}_i \times \mathbf{w}_i \times \mathbf{r}_{ci}^i \right) - m_i \mathbf{g} \]

\[ \mathbf{\tau}_i = \mathbf{\tau}_{i+1} - \mathbf{r}_{i-1}^0 \times \mathbf{f}_i + \mathbf{r}_i^0 \times \mathbf{f}_{i+1} + \mathbf{I}_i \mathbf{a}_i + \mathbf{w}_i \times (\mathbf{I}_i \mathbf{w}_i) \]
Recursive Algorithms

- Given
  - Robot configuration
    - Positions of joints
    - Velocities of joints
    - Accelerations of joints
  - Desired force and torque at end effector

- Compute
  - Forces and torques to be applied at each joint

- A backward recursive algorithms
Recursive Algorithms (cont.)

- Previous: forces and torques as functions of (linear, angular) (velocity, acceleration)
  - A backward iteration

- Need: (linear, angular) (velocity, acceleration) as functions of generalized coordinates
  - A forward iteration
Angular Velocity and Acceleration

\[ \mathbf{w}^{i-2} = \mathbf{w}^{i-2} + \mathbf{R}^{i-2} \dot{\theta}^i \mathbf{z}^{i-1} \]

\[ \Rightarrow \mathbf{w}^0 = \mathbf{w}^{i-1} + \mathbf{R}^{0} \dot{\theta}^i \mathbf{z}^{i-1} \]

\[ \Rightarrow \mathbf{w}^0 = \mathbf{w}^{i-1} + \mathbf{q}^i \mathbf{z}^{0} \]

\[ \Rightarrow \alpha^0 = \alpha^{i-1} + \mathbf{q}^i \mathbf{z}^{0} + \mathbf{q}^i (\mathbf{w}^{0} \times \mathbf{z}^{0}) \]
Linear Velocity and Acceleration

\[ \mathbf{O}_i^0 = \mathbf{O}_{i-1}^0 + \mathbf{R}_{i-1}^0 \mathbf{O}_{i-1,i}^{i-1} \]

\[ \Rightarrow \mathbf{v}_i^0 = \mathbf{v}_{i-1}^0 + \frac{d\mathbf{R}_{i-1}^0}{dt} \mathbf{O}_{i-1,i}^{i-1} + \mathbf{R}_{i-1}^0 \frac{d\mathbf{O}_{i-1,i}^{i-1}}{dt} \]

\[ \Rightarrow \mathbf{v}_i^0 = \mathbf{v}_{i-1}^0 + \mathbf{w}_{i-1}^0 \times \mathbf{R}_{i-1}^0 \mathbf{O}_{i-1,i}^{i-1} + \mathbf{R}_{i-1}^0 \dot{\theta}_i z_{i-1}^0 \]

\[ \Rightarrow \mathbf{v}_i^0 = \mathbf{v}_{i-1}^0 + \mathbf{w}_{i-1}^0 \times \mathbf{O}_{i-1,i}^{i-1} + \dot{\theta}_i z_{i-1}^0 \]

\[ \Rightarrow \mathbf{a}_i^0 = \mathbf{a}_{i-1}^0 + \mathbf{w}_{i-1}^0 \times \frac{d\mathbf{O}_{i-1,i}^{i-1}}{dt} + \dot{\theta}_i z_{i-1}^0 + \ddot{\theta}_i \frac{dz_{i-1}^0}{dt} \]

\[ \Rightarrow \mathbf{a}_i^0 = \mathbf{a}_{i-1}^0 + \mathbf{w}_{i-1}^0 \times \frac{d\mathbf{O}_{i-1,i}^{i-1}}{dt} + \dot{\theta}_i z_{i-1}^0 + \ddot{\theta}_i \mathbf{w}_{i-1}^0 \times z_{i-1}^0 \]

- Either a revolute or prismatic joint
- This term is zero for revolute joint

\[ \mathbf{O}_i^0 = \mathbf{O}_{i-1}^0 + \mathbf{R}_{i-1}^0 \mathbf{O}_{i-1,i}^{i-1} \]
Basic Procedures

- Specify
  - End effector torque and force
  - Joint displacement, velocity, and acceleration
- Use the equations in previous slides to
  - Recursively compute linear and angular velocities and acceleration of links forward (from base to tool), then
  - Recursively compute the torques and forces of links backward (from tool to base)
In More Details

- Static
- Dynamic

Computer Graphics
Inverse position kinematics

End effector’s position and orientation

Inverse Dynamics

(f, τ)

End effector’s force and torque

Inverse position kinematics

(q, q, ̈q)

Forward velocity kinematics

(v, a, w, ß)
Why is Dynamics Important?

- Scenario:
  - Robot arm holds and moves the tool following a pre-planned trajectory (right)
  - If the goal is to visualize (graphics), $q(t)$ is sufficient

Inverse position kinematics ($q, \dot{q}, \ddot{q}$)

End effector’s position and orientation

Computer Graphics
CG Does not Need Dynamics?

- The answer is no
- Scenario
  - Collision of rigid bodies and robots
    - Transfer of forces and torques
  - Does the pre-planned path make sense?
    - Will the simulation look realistic?
    - Can robot move that fast or support that much load?
Why Robotics Need Dynamics?

- If we have \((q, \dot{q}, \ddot{q})\) from position kinematics, do we really need dynamics?
- The answer is definitely yes
- Why?
- Many reasons
**Reason #1**

- Knowing $\ddot{q}$, calculate needed robot control signals
  - Revolute joint: apply torque
  - Prismatic joint: apply force
- You need to know the load you carry or move at the end effector
- You also need to know weight and inertia tensor of the particular link that you want to move
- You also need to know the force and torque needed to counter reaction from neighboring links (Newton’s third law)
- Otherwise, you are in for a surprise
Reason #2

- Prismatic joint
  - Can only apply force
  - Too much torque applied to it, it is going to break
- Revolute joint
  - Can only apply torque
  - Too much force applied to it, it is going to break
- Physically, is it at all possible to realized the desired motion?
Inverse position kinematics

End effector’s position and orientation

Inverse Dynamics

(\mathbf{f}, \tau)

Feedback link
From dynamics analysis

Inverse position kinematics

(q, \dot{q}, \ddot{q})

Forward velocity kinematics

(v, a, w, \alpha)

End effector’s force and torque

Computer Graphics
Based on Lagrange equation

Need to figure out the expression of $K$ and $V$ in terms of $q_i$

\[
f_i = \frac{d}{dt} \left( \frac{\partial (K - V)}{\partial \dot{q}_i} - \frac{\partial (K - V)}{\partial q_i} \right)
\]

\[
K = \frac{1}{2} m_i v_i^T v_i + \frac{1}{2} w_i^T I_i w_i
\]

\[
V = m_i g^T r_{ci}
\]
Example: two link system
the distance between joint \( i \) and the center of mass of link \( i \) (the center of mass is assumed to be on the straight line connecting the two joints).

We assume that the first joint driving torque \( \tau_1 \) acts between the base and link 1, and the second joint driving torque \( \tau_2 \) acts between links 1 and 2. We will also assume that the gravitational force acts in the negative \( Y \) direction.

Choosing \( q_1 = \theta_1 \) and \( q_2 = \theta_2 \) as generalized coordinates, we will find the Lagrangian function. Let the kinetic energy and the potential energy for link \( i \) be \( K_i \) and \( P_i \), respectively. For link 1, we have

\[
K_1 = \frac{1}{2} m_1 l_{g1}^2 \dot{\theta}_1^2 + \frac{1}{2} I_1 \dot{\theta}_1^2, \tag{3.32}
\]

\[
P_1 = m_1 g s_1, \tag{3.33}
\]

where \( g \) is the magnitude of gravitational acceleration. For link 2, since the position of its center of mass \( s_2 = [s_{2x}, s_{2y}]^T \) is given by

\[
s_{2x} = l_1 C_1 + l_2 C_{12}, \tag{3.34a}
\]

\[
s_{2y} = l_1 S_1 + l_2 S_{12}, \tag{3.34b}
\]

we have the relation

\[
\dot{s}_2^T \ddot{s}_2 = l_1^2 \dot{\theta}_1^2 + l_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + 2l_1 l_2 C_2 (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2). \tag{3.35}
\]

Hence,

\[
K_2 = \frac{1}{2} m_2 s_2^T \ddot{s}_2 + \frac{1}{2} \ddot{I}_2 (\dot{\theta}_1 + \dot{\theta}_2)^2 \tag{3.36}
\]

and

\[
P_2 = m_2 g l_1 S_1 + l_2 S_{12}). \tag{3.37}
\]

Calculating \( L = K_1 + K_2 - P_1 - P_2 \) and substituting this into equation 3.29 yields the equations of motion for the two-link manipulator:

\[
\tau_1 = [m_1 l_{g1}^2 + I_1 + m_2 (l_1^2 + l_2^2 + 2l_1 l_2 C_2) + I_2] \ddot{\theta}_1

+ [m_2 (l_2^2 + I_2) + I_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}). \tag{3.38a}
\]

\[
\tau_2 = [m_2 l_2 C_2 + I_2 + l_2 C_{12}] \ddot{\theta}_1 + (m_2 l_2^2 + I_2) \ddot{\theta}_2

+ m_2 l_1 l_2 S_2 \dot{\theta}_1^2 + m_2 g l_2 C_{12}. \tag{3.38b}
\]
This can also be rewritten as
\begin{align}
\tau_1 &= M_{11} \ddot{\theta}_1 + M_{12} \ddot{\theta}_2 + h_{122} \dot{\theta}_2^2 + 2 h_{112} \dot{\theta}_1 \dot{\theta}_2 + g_1, \quad (3.39a) \\
\tau_2 &= M_{21} \ddot{\theta}_1 + M_{22} \ddot{\theta}_2 + h_{211} \dot{\theta}_1^2 + g_2, \quad (3.39b)
\end{align}

where
\begin{align}
M_{11} &= m_1 l_g^2 + \bar{T}_1 + m_2 (l_4^2 + l_5^2 + 2 l_1 l_2 C_2) + \bar{I}_2, \quad (3.40a) \\
M_{12} &= M_{21} = m_2 (l_2^2 + l_4 l_2 C_2) + \bar{I}_2, \quad (3.40b) \\
M_{22} &= m_2 l_2^2 + \bar{I}_2, \quad (3.40c) \\
h_{122} &= h_{112} = - h_{211} = - m_2 l_1 l_2 S_2, \quad (3.41) \\
g_1 &= m_1 g l_g C_1 + m_2 g (l_1 C_1 + l_2 C_1), \quad (3.42a) \\
g_2 &= m_2 g l_2 C_1. \quad (3.42b)
\end{align}

We call $M_{ij}$ the effective inertia, $M_{ij}$ the coupling inertia, $h_{ij}$ the centrifugal acceleration coefficient, and $h_{ik}$ ($j \neq k$) the Coriolis acceleration coefficient. The term $g_1$ represents the gravity force.

We can further rewrite equation 3.39 in a more compact form by noting that the kinetic energy $K = K_1 + K_2$ may be expressed in a quadratic form as
\begin{equation}
K = \frac{1}{2} \dot{\theta}^T M(\theta) \dot{\theta}, \quad (3.43)
\end{equation}

where $\dot{\theta} = [\dot{\theta}_1, \dot{\theta}_2]^T$ and $M(\theta)$ is the following positive-definite matrix:
\begin{equation}
M(\theta) = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}. \quad (3.44)
\end{equation}

Using equations 3.43, 3.44, and 3.29, the equations of motion 3.39 are also given by
\begin{equation}
\tau = M(\theta) \ddot{\theta} + h(\theta, \dot{\theta}) + g(\theta), \quad (3.45)
\end{equation}

where, with $\text{col}[\cdot]$ denoting a column vector,
\begin{align}
h(\theta, \dot{\theta}) &= \text{col} \left[ \sum_{i=1}^{2} \sum_{k=1}^{2} \frac{1}{2} \left( \frac{\partial M_{ij}}{\partial \theta_k} - \frac{1}{2} \frac{\partial M_{ik}}{\partial \theta_l} \right) \dot{\theta}_i \dot{\theta}_k \right], \quad (3.46) \\
g(\theta) &= \text{col}[g_i]. \quad (3.47)
\end{align}

and where $M(\theta) \ddot{\theta}$ is the inertial force term, $h(\theta, \dot{\theta})$ is the centrifugal and
Lagrange Steps

- Kinetic energy
- Potential energy
- Substitution into Lagrange equation
**Kinetic Energy**

\[
\begin{bmatrix}
\mathbf{v}_{ci} \\
\mathbf{w}_i
\end{bmatrix} = \begin{bmatrix}
\mathbf{J}_{ci}^V \\
\mathbf{J}_{ci}^w
\end{bmatrix} \dot{\mathbf{q}}_i
\]

\[
K_i = \frac{1}{2} m_i \mathbf{v}_i^T \mathbf{v}_i + \frac{1}{2} \mathbf{w}_i^T \mathbf{I}_i \mathbf{w}_i
\]

\[
= \frac{1}{2} m_i (\mathbf{J}_{ci}^V \dot{\mathbf{q}}_i)^T (\mathbf{J}_{ci}^V \dot{\mathbf{q}}_i) + \frac{1}{2} (\mathbf{J}_{ci}^w \dot{\mathbf{q}}_i)^T \mathbf{R}_i^{0body} \mathbf{R}_i^{0T} (\mathbf{J}_{ci}^w \dot{\mathbf{q}}_i)
\]

\[
K = \sum_{i=1}^{n} \frac{1}{2} m_i \mathbf{v}_i^T \mathbf{v}_i + \frac{1}{2} \mathbf{w}_i^T \mathbf{I}_i \mathbf{w}_i
\]

\[
= \frac{1}{2} \dot{\mathbf{q}}_i^T \left\{ \sum_{i=1}^{n} m_i \mathbf{J}_{ci}^V^T \mathbf{J}_{ci}^V + \mathbf{J}_{ci}^w^T \mathbf{R}_i^{0body} \mathbf{R}_i^{0T} \mathbf{J}_{ci}^w \right\} \dot{\mathbf{q}}_i
\]

\[
= \frac{1}{2} \dot{\mathbf{q}}_i^T \mathbf{B}(\mathbf{q}) \dot{\mathbf{q}}_i = \sum_{j=1}^{n} \sum_{i=1}^{n} b_{ij} \ddot{q}_i \ddot{q}_j
\]
\[
\begin{align*}
\begin{cases}
    v_{ci} &= J^o_{ci} \cdot \dot{q} \\
    \omega_i &= J^\omega_{ci} \cdot \dot{q}.
\end{cases}
\end{align*}
\]

\[
K_i = \frac{1}{2} m_i \cdot v_{ci}^t \cdot v_{ci} + \frac{1}{2} \omega_i^t \cdot R_{ci} \cdot \{^ci I_{ci}\} \cdot R_{ci}^t \cdot \omega_i.
\]

\[
K_i = \frac{1}{2} m_i \cdot (J^o_{ci} \cdot \dot{q})^t \cdot (J^o_{ci} \cdot \dot{q}) + \frac{1}{2} (J^\omega_{ci} \cdot \dot{q})^t \cdot R_{ci} \cdot \{^ci I_{ci}\} \cdot R_{ci}^t \cdot (J^\omega_{ci} \cdot \dot{q})
\]

or

\[
K_i = \frac{1}{2} \dot{q}^t \cdot \left\{ m_i \cdot (J^o_{ci})^t \cdot J^o_{ci} + (J^\omega_{ci})^t \cdot R_{ci} \cdot \{^ci I_{ci}\} \cdot R_{ci}^t \cdot J^\omega_{ci} \right\} \cdot \dot{q}. \quad (4.123)
\]

\[
K = \sum_{i=1}^n K_i
\]

\[
K = \frac{1}{2} \dot{q}^t \cdot \left\{ \sum_{i=1}^n \left[ m_i \cdot (J^o_{ci})^t \cdot J^o_{ci} + (J^\omega_{ci})^t \cdot R_{ci} \cdot \{^ci I_{ci}\} \cdot R_{ci}^t \cdot J^\omega_{ci} \right] \right\} \cdot \dot{q}.
\]

If we define

\[
B(q) = \sum_{i=1}^n \left[ m_i \cdot (J^o_{ci})^t \cdot J^o_{ci} + (J^\omega_{ci})^t \cdot R_{ci} \cdot \{^ci I_{ci}\} \cdot R_{ci}^t \cdot J^\omega_{ci} \right], \quad (4.124)
\]

\[
K = \frac{1}{2} \dot{q}^t \cdot B(q) \cdot \dot{q}
\]

\[
K = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n b_{ij}(q) \cdot \dot{q}_i \cdot \dot{q}_j.
\]
\[ f_i = \frac{d}{dt} \frac{\partial(K - V)}{\partial \dot{q}_i} - \frac{\partial(K - V)}{\partial q_i} \]

\[ K = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} b_{ij}(q) \cdot \dot{q}_i \cdot \dot{q}_j. \]

\[ \frac{\partial K}{\partial \dot{q}_i} = \sum_{j=1}^{n} b_{ij}(q) \cdot \dot{q}_j. \]

\[ \frac{d}{dt} \left( \frac{\partial K}{\partial \dot{q}_i} \right) = \sum_{j=1}^{n} \left[ b_{ij}(q) \cdot \dot{q}_j + b_{ij}(q) \cdot \ddot{q}_j \right]. \]

\[ \dot{b}_{ij}(q) = \frac{\partial b_{ij}}{\partial q_1} \cdot \dot{q}_1 + \frac{\partial b_{ij}}{\partial q_2} \cdot \dot{q}_2 + \ldots + \frac{\partial b_{ij}}{\partial q_n} \cdot \dot{q}_n \]

\[ \dot{b}_{ij}(q) = \sum_{k=1}^{n} \frac{\partial b_{ij}}{\partial q_k} \cdot \dot{q}_k. \]

\[ \frac{d}{dt} \left( \frac{\partial K}{\partial \dot{q}_i} \right) = \sum_{j=1}^{n} b_{ij}(q) \cdot \ddot{q}_j + \sum_{j=1}^{n} \sum_{k=1}^{n} \frac{\partial b_{ij}}{\partial q_k} \cdot \dot{q}_k \cdot \dot{q}_j. \]
\[ f_i = \frac{d}{dt} \frac{\partial (K - V)}{\partial \dot{q}_i} - \frac{\partial (K - V)}{\partial q_i} \]

\[ K = \frac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{n} b_{jk}(q) \cdot \dot{q}_k \cdot \dot{q}_j. \]

\[ \frac{\partial K}{\partial q_i} = \frac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{n} \frac{\partial b_{jk}}{\partial q_i} \cdot \dot{q}_k \cdot \dot{q}_j. \]
\[ V_i = -m_i \cdot g_i \cdot r_{ci}. \]

\[ V = \sum_{i=1}^{n} V_i = \sum_{i=1}^{n} m_i \cdot g_i \cdot r_{ci}. \]

\[ \frac{\partial V}{\partial q_i} = -\sum_{j=1}^{n} m_j \cdot g_j \cdot \frac{\partial r_{cj}}{\partial q_i}. \]

\[ \frac{\partial r_{cj}}{\partial q_s} = \frac{\partial r_{cj}}{\partial q_i} = \frac{\partial v_{cj}}{\partial q_i}. \]

\[ \frac{\partial V}{\partial q_i} = -\sum_{j=1}^{n} m_j \cdot g_j \cdot J_{cj}^o(i). \]
\[
\sum_{j=1}^{n} b_{ij}(q) \cdot \ddot{q}_j + \sum_{j=1}^{n} c_{ij}(q, \dot{q}) \cdot \dot{q}_j + g_i(q) = \tau_i, \quad \forall i \in [1, n]
\]

compact matrix form as:

\[
\begin{bmatrix}
B(q) & \dddot{q} + C(q, \dot{q}) \cdot \dot{q} + G(q) - \tau
\end{bmatrix}
\]
Potential Energy and Lagrange

\[ V = \sum_{i=1}^{n} m_i g^T r_{ci}(q_1, q_2, \ldots, q_n) \]

\[ L = K + V = \sum_{j=1}^{n} \sum_{i=1}^{n} b_{ij} \dot{q}_i \dot{q}_j + \sum_{i} m_i g^T r_{ci}(q_1, \ldots, q_n) \]

\[ f_i = \frac{d}{dt} \frac{\partial(K-V)}{\partial \dot{q}_i} - \frac{\partial(K-V)}{\partial q_i} \]

\[ B(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = f \]
Put It All Together

\[ B(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = f \]

- A very complicated expression 😊
- How do you use the equations?
- Force calculation
  - Given \( q, \dot{q}, \ddot{q}, \) and force and torque required at the end effector
  - Calculate generalized force at all others
  - A forward evaluation
- Coordinate calculation
  - Given force and position and force
  - Calculate velocity and acceleration
  - A inverse solve

Computer Graphics
Angular Velocity and Acceleration

\[ \omega_i = \omega_{i-1} + R_{i-1} \cdot \{ \dot{q}_i \cdot i^{-1} \bar{k}_{i-1} \} \]

\[ \omega_i = \omega_{i-1} + \dot{q}_i \cdot \bar{k}_{i-1}. \]

\[ \alpha_i = \alpha_{i-1} + \ddot{q}_i \cdot \bar{k}_{i-1} + \ddot{q}_i \cdot (\omega_{i-1} \times \bar{k}_{i-1}). \]

\[ \frac{d}{dt}(\bar{k}_{i-1}) = \omega_{i-1} \times \bar{k}_{i-1}. \]
Linear Velocity and Acceleration

\[ O_i = O_{i-1} + O_{i-1,i} \]

\[ O_{i-1,i} = R_i \cdot \dot{O}_{i-1,i} \]

\[ O_i = O_{i-1} + R_i \cdot \dot{O}_{i-1,i} \]

\[ v_i = v_{i-1} + \frac{dR_i}{dt} \cdot \dot{O}_{i-1,i} + R_i \cdot \frac{d}{dt} \{ \dot{O}_{i-1,i} \} \]

\[ v_i = v_{i-1} + S(\omega_i) \cdot R_i \cdot \dot{O}_{i-1,i} + R_i \cdot \dot{q_i} \cdot \dot{k}_{i-1} \]

\[ v_i = \dot{v}_{i-1} + \omega_i \times O_{i-1,i} + \dot{q_i} \cdot \dot{k}_{i-1} \]

\[ = a_{i-1} + \alpha_i \times O_{i-1,i} + \omega_i \times \frac{d}{dt} \{ O_{i-1,i} \} + \dot{q_i} \cdot \dot{k}_{i-1} + \dot{q_i} \cdot (\omega_{i-1} \times \dot{k}_{i-1}) \]

\[ \text{NOTE:} \quad \frac{d}{dt}(\dot{k}_{i-1}) = \omega_{i-1} \times \dot{k}_{i-1} \]