L10: Constraint Force Computation, Control

Hao Su

Spring, 2021

Agenda

- Constrained Lagrangian Method
- Variational Method
- Control
- PID Control

click to jump to the section.

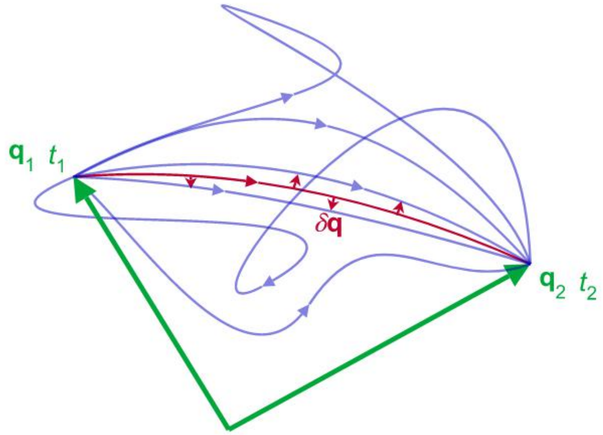
Review: Lagrangian Function

- Let $q \in \mathbb{R}^n$ be the generalized coordinates.
- **Lagrangian function:** $L(q, \dot{q}) = T(q, \dot{q}) - V(q)$
  - $T(q, \dot{q})$: kinetic energy of system
  - $V(q)$: potential energy (given by some conservative force, e.g., gravity, electric field force)

Review: The Principle of Stationary Action

- Given a pair of time instants, t_1 and t_2
- What is the curve $\boldsymbol{q} : [t_1, t_2] \to \mathcal{C}$ in the configuration space $\mathcal{C}$?
- **Action** is defined to be a functional of $\boldsymbol{q}(t)$:

$$S[\boldsymbol{q}] = \int_{t_1}^{t_2} L(q, \dot{q}) \mathrm{d}t = \int_{t_1}^{t_2} [T(\boldsymbol{q}, \dot{\boldsymbol{q}}) - V(\boldsymbol{q})] \mathrm{d}t$$



Wikipedia: Stationary Action Principle

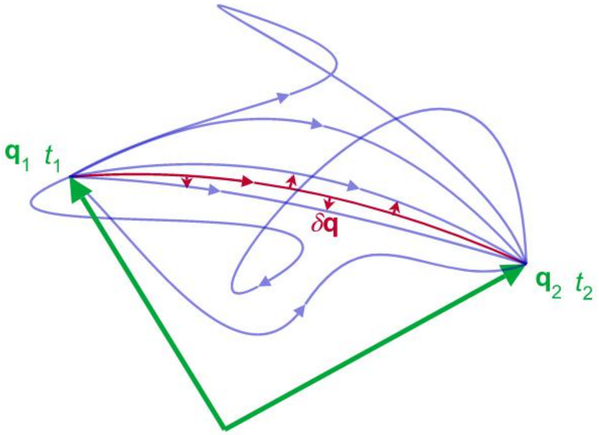
Review: The Principle of Stationary Action

- The actual curve $\boldsymbol{q}(t)$ is a stationary point of the $S[\boldsymbol{q}]$:

$$\forall \boldsymbol{\delta} : [t_1, t_2] \to \mathcal{C}, \quad \lim_{\epsilon \to 0} \frac{1}{\epsilon} (\mathcal{S}[\boldsymbol{q} + \epsilon \boldsymbol{\delta}] - \mathcal{S}[\boldsymbol{q}]) = 0 \tag{1}$$

- Note: Treating $\boldsymbol{q}$ as a variable, and (1) is an extension of the first-order optimality condition that we use in calculus:

$$\nabla_q S[q] = 0$$



Wikipedia: Stationary Action Principle

Review: The Principle of Stationary Action

- The actual curve $\boldsymbol{q}(t)$ is a stationary point of the $S[\boldsymbol{q}]$:

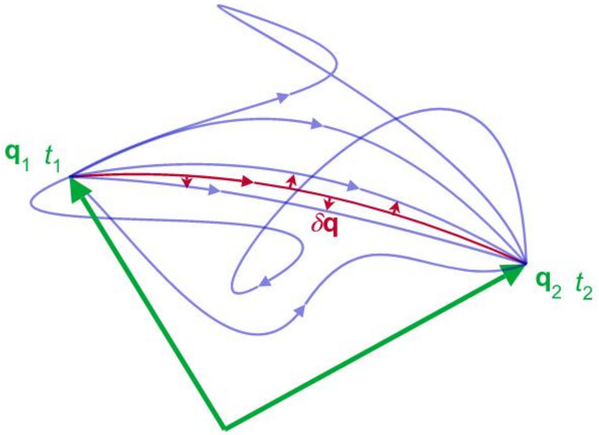
$$\forall \boldsymbol{\delta} : [t_1, t_2] \to \mathcal{C}, \quad \lim_{\epsilon \to 0} \frac{1}{\epsilon} (\mathcal{S}[\boldsymbol{q} + \epsilon \boldsymbol{\delta}] - \mathcal{S}[\boldsymbol{q}]) = 0 \qquad (1)$$

- Note: Treating $\boldsymbol{q}$ as a variable, and (1) is an extension of the first-order optimality condition that we use in calculus:

$$\nabla_q S[q] = 0$$

- Using *variational method*, condition (1) becomes

$$\frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = 0$$



Wikipedia: Stationary Action Principle

Review: Euler-Lagrange Equation

- When there are external non-conservative generalized force $\boldsymbol{F} \in \mathbb{R}^n$ added to the system (e.g., torque at robot arm joints), we have the following Euler-Lagrange equation:

$$\boldsymbol{F} = \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} \qquad \text{(Euler-Lagrange Equation)}$$

Example: Robot Arm

Robot Arm

- For kinematic chains with n joints, it is convenient and always possible to choose the joint angles $\theta = (\theta_1, \ldots, \theta_n)$ and the joint torques $\tau = (\tau_1, \ldots, \tau_n)$ as the generalized coordinates and generalized forces, respectively.
  - If joint i is revolute: θ_i joint angle and τ_i is joint torque
  - If joint i is prismatic: θ_i joint position and τ_i is joint force
  - Then, the joint force/torque τ and joint angular/linear velocity $\dot{\theta}$ form dual pairs.
- Lagrangian Equations:

$$\tau_i = \frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial L}{\partial \dot{\theta}_i} - \frac{\partial L}{\partial \theta_i}$$

Review of Notations for Robot Arms

For each link $i \in \{1, \dots, n\}$, the corresponding body-frame $\mathcal{F}_i$ is attached to the center of mass of link i. All the following quantities are expressed in $\mathcal{F}_i$

- $\boldsymbol{\xi}_i^b$: body twist of the link frame $\mathcal{F}_i$
- $\mathfrak{M}_i^b = \begin{bmatrix} m_i \mathrm{Id}_{3\times 3} & 0 \\ 0 & \boldsymbol{I}_i^b \end{bmatrix}$: body inertia matrix
- Kinetic energy of link i: $T_i = \frac{1}{2}(\boldsymbol{\xi}_i^b)^T \mathfrak{M}_i^b \boldsymbol{\xi}_i^b$
- $J_i^b \in \mathbb{R}^{6\times n}$: body Jacobian of link i (so that we have the forward dynamics $\boldsymbol{\xi}_i^b = J_i^b \dot{\theta}$)

Kinetic and Potential Energies

- Total kinetic energy:

$$T(\theta,\dot{\theta}) = \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{\xi}_i^b)^T\mathfrak{M}_i^b\boldsymbol{\xi}_i^b = \frac{1}{2}\dot{\theta}^T\underbrace{(\sum_{i=1}^{n}(J_i^b(\theta)\mathfrak{M}_i^bJ_i^b(\theta)))}_{\boldsymbol{M}^b(\theta)}\dot{\theta} := \frac{1}{2}\dot{\theta}^T\boldsymbol{M}^b(\theta)\dot{\theta}$$

- Potential energy:

$$V(\theta) = \sum_{i=1}^{n} m_i g h_i(\theta)$$

  - $h_i(\theta)$: height of center of mass of link i

Lagrangian Equation

- Plug $L = T - V$ into $F = \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q}$, and we have
- $\tau_i = \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta_i}$

$$\tau_i = \sum_{j=1}^{n} M_{ij}^b(\theta)\ddot{\theta}_j + \sum_{j=1}^{n}\sum_{k=1}^{n} \Gamma_{ijk}^b(\theta)\dot{\theta}_j\dot{\theta}_k + \frac{\partial V}{\partial \theta_i}$$

  M_{ij}^b is the (i,j)-th entry of matrix $\boldsymbol{M}^b$

- $\Gamma_{ijk}^b(\theta)$ is called the **Christoffel symbols of the first kind**

$$\Gamma_{ijk}^b(\theta) = \frac{1}{2}\left(\frac{\partial M_{ij}^b}{\partial \theta_k} + \frac{\partial M_{ik}^b}{\partial \theta_j} - \frac{\partial M_{jk}^b}{\partial \theta_i}\right)$$

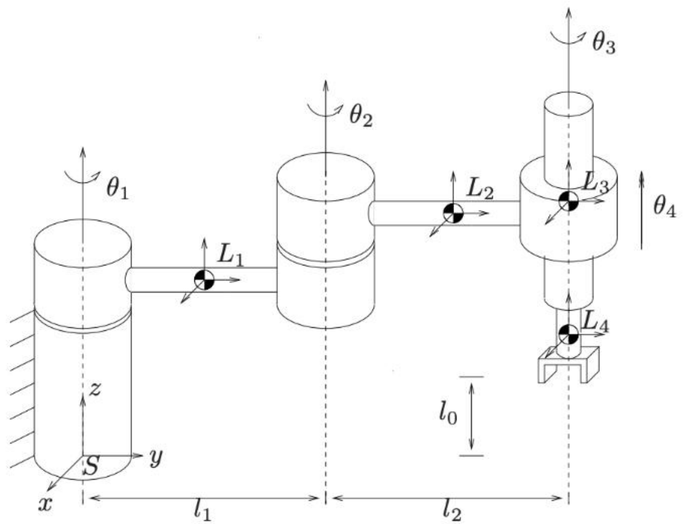
Lagrangian Equation

- Lagrangian equation in vector form:

$$\tau = \boldsymbol{M}^b(\theta)\ddot{\theta} + C^b(\theta, \dot{\theta})\dot{\theta} + g^b(\theta)$$

  - $C^b_{ij}(\theta, \dot{\theta}) := \sum_{k=1}^n \Gamma^b_{ijk}\dot{\theta}_k$ is called the **Coriolis matrix**
    - Recall that in the body-frame Newton Euler equation, we also have a Coriolis term that comes from the derivative of the rotational inertia ($\boldsymbol{\tau}^b = \boldsymbol{I}^b\dot{\boldsymbol{\omega}}^b + \boldsymbol{\omega}^b \times \boldsymbol{I}^b\boldsymbol{\omega}^b$). It was used to compensate for the rotational acceleration of the body frame.
    - This $C^b_{ij}(\theta, \dot{\theta})$ also comes from taking the derivative of $\boldsymbol{M}^b$ w.r.t. θ. Because $\boldsymbol{M}^b$ and $\boldsymbol{\xi}^b$ are described in the body frame in our derivation, we also need this Coriolis term to compensate for the movement of the body frame.
  - $g^b(\theta)$ is due to gravity in our derivation. If there are other external forces (e.g., friction), it would also show up here.

- Equations for a simple arm



$$
\begin{aligned}
M_{11} &= I_{y2}s_2^2 + I_{y3}s_{23}^2 + I_{z1} + I_{z2}c_2^2 + I_{z3}c_{23}^2 \\
&\quad + m_2r_1^2c_2^2 + m_3(l_1c_2 + r_2c_{23})^2 \\
M_{12} &= 0 \\
M_{13} &= 0 \\
M_{21} &= 0 \\
M_{22} &= I_{x2} + I_{x3} + m_3l_1^2 + m_2r_1^2 + m_3r_2^2 + 2m_3l_1r_2c_3 \\
M_{23} &= I_{x3} + m_3r_2^2 + m_3l_1r_2c_3 \\
M_{31} &= 0 \\
M_{32} &= I_{x3} + m_3r_2^2 + m_3l_1r_2c_3 \\
M_{33} &= I_{x3} + m_3r_2^2.
\end{aligned}
$$

$$
\begin{bmatrix}
0 \\
-(m_2gr_1 + m_3gl_1)\cos\theta_2 - m_3r_2\cos(\theta_2 + \theta_3)) \\
-m_3gr_2\cos(\theta_2 + \theta_3))
\end{bmatrix}
$$

$$
\begin{aligned}
\Gamma_{112} &= (I_{y2} - I_{z2} - m_2r_1^2)c_2s_2 + (I_{y3} - I_{z3})c_{23}s_{23} \\
&\quad - m_3(l_1c_2 + r_2c_{23})(l_1s_2 + r_2s_{23}) \\
\Gamma_{113} &= (I_{y3} - I_{z3})c_{23}s_{23} - m_3r_2s_{23}(l_1c_2 + r_2c_{23}) \\
\Gamma_{121} &= (I_{y2} - I_{z2} - m_2r_1^2)c_2s_2 + (I_{y3} - I_{z3})c_{23}s_{23} \\
&\quad - m_3(l_1c_2 + r_2c_{23})(l_1s_2 + r_2s_{23}) \\
\Gamma_{131} &= (I_{y3} - I_{z3})c_{23}s_{23} - m_3r_2s_{23}(l_1c_2 + r_2c_{23}) \\
\Gamma_{211} &= (I_{z2} - I_{y2} + m_2r_1^2)c_2s_2 + (I_{z3} - I_{y3})c_{23}s_{23} \\
&\quad + m_3(l_1c_2 + r_2c_{23})(l_1s_2 + r_2s_{23}) \\
\Gamma_{223} &= -l_1m_3r_2s_3 \\
\Gamma_{232} &= -l_1m_3r_2s_3 \\
\Gamma_{233} &= -l_1m_3r_2s_3 \\
\Gamma_{311} &= (I_{z3} - I_{y3})c_{23}s_{23} + m_3r_2s_{23}(l_1c_2 + r_2c_{23}) \\
\Gamma_{322} &= l_1m_3r_2s_3
\end{aligned}
$$

- Equations for PUMA 560 Arm

$$
\begin{aligned}
I_2 &= I_{zz2} + m_2 * (r_{x2}^2 + r_{y2}^2) + (m_5 + m_4 + m_5 + m_6) * a_2^2 ; \\
I_3 &= -I_{xx2} + I_{yy2} + (m_3 + m_4 + m_5 + m_6) * a_2^2 \\
&\quad m_2 * r_{x2}^2 - m_2 * r_{y2}^2 ; \\
I_4 &= m_3 * r_{x2} * (d_2 + r_{z2}) + m_5 * a_2 * r_{z3} \\
&\quad + (m_5 + m_4 + m_5 + m_6) * a_2 * (d_2 + d_3) ; \\
I_5 &= -m_5 * a_2 * r_{y3} + (m_4 + m_5 + m_6) * a_2 * d_4 + m_4 * a_2 * r_{z4} ; \\
I_6 &= I_{zz5} + m_5 * r_{z3}^2 + m_4 * a_3^2 + m_4 * (d_4 + r_{z4})^2 + I_{yy4} \\
&\quad + m_5 * a_3^2 + m_5 * d_4^2 + I_{zz5} + m_6 * a_3^2 + m_6 * d_4^2 \\
&\quad + m_6 * r_{z6}^2 + I_{zz6} ; \\
I_7 &= m_3 * r_{y3}^2 + I_{xx5} - I_{yy3} + m_4 * r_{x4}^2 + 2 * m_4 * d_4 * r_{z4} \\
&\quad + (m_4 + m_5 + m_6) * (d_4^2 - a_3^2) + I_{yy4} - I_{zz4} + I_{zz5} \\
&\quad - I_{yy5} + m_6 * r_{z6}^2 - I_{zz6} + I_{zz6} ; \\
I_8 &= -m_4 * (d_2 + d_3) * (d_4 + r_{z4}) - (m_5 + m_6) * (d_2 + d_3) * d_4 \\
&\quad m_3 * r_{y3} * r_{z3} + m_5 * (d_2 + d_3) * r_{y3} ; \\
I_9 &= m_2 * r_{y2} * (d_2 + r_{z2}) ; \\
I_{10} &= 2 * m_4 * a_3 * r_{z4} + 2 * (m_4 + m_5 + m_6) * a_3 * d_4 ; \\
I_{11} &= -2 * m_2 * r_{x2} * r_{y2} ; \\
I_{12} &= (m_4 + m_5 + m_6) * a_2 * a_3 ; \\
I_{13} &= (m_4 + m_5 + m_6) * a_3 * (d_2 + d_3) ; \\
I_{14} &= I_{zz4} + I_{yy5} + I_{zz6} ; \\
I_{15} &= m_6 * d_4 * r_{z6} ; \\
I_{16} &= m_6 * a_2 * r_{z6} ; \\
I_{17} &= I_{zz5} + I_{zz6} + m_6 * r_{z6}^2 ; \\
I_{18} &= m_6 * (d_2 + d_3) * r_{z6} ; \\
I_{19} &= I_{yy4} - I_{zz4} + I_{zz5} - I_{yy5} + m_6 * r_{z6}^2 + I_{zz6} - I_{zz6} ; \\
I_{20} &= I_{yy5} - I_{zz5} - m_6 * r_{z6}^2 + I_{zz6} - I_{zz6} ; \\
I_{21} &= I_{zz4} - I_{yy4} + I_{zz5} - I_{zz5} ; \\
I_{22} &= m_6 * a_3 * r_{z6} ; \\
I_{23} &= I_{zz6} ;
\end{aligned}
$$

Part II. Gravitional Constants

$$
\begin{aligned}
g_1 &= -g * ((m_3 + m_4 + m_5 + m_6) * a_2 + m_2 * r_{x2}) ; \\
g_2 &= g * (m_3 * r_{y3} - (m_4 + m_5 + m_6) * d_4 - m_4 * r_{z4}) ; \\
g_3 &= g * m_2 * r_{y2} ; \\
g_4 &= -g * (m_4 + m_5 + m_6) * a_3 ; \\
g_5 &= -g * m_6 * r_{z6} ;
\end{aligned}
$$

Table A3. Computed Values for the Constants Appearing in the Equations of Forces of Motion.

(Inertial constants have units of kilogram meters-squared)

I_1	1.43	± 0.05	I_2	1.75	± 0.07
I_3	1.38	± 0.05	I_4	6.90×10^{-1}	$\pm 0.20\times10^{-1}$
I_5	3.72×10^{-1}	$\pm 0.31\times10^{-1}$	I_6	3.33×10^{-1}	$\pm 0.16\times10^{-1}$
I_7	2.98×10^{-1}	$\pm 0.29\times10^{-1}$	I_8	-1.34×10^{-1}	$\pm 0.14\times10^{-1}$
I_9	2.38×10^{-2}	$\pm 1.20\times10^{-2}$	I_{10}	-2.13×10^{-2}	$\pm 0.22\times10^{-2}$
I_{11}	-1.42×10^{-2}	$\pm 0.70\times10^{-2}$	I_{12}	-1.10×10^{-2}	$\pm 0.11\times10^{-2}$
I_{13}	-3.70×10^{-3}	$\pm 0.90\times10^{-3}$	I_{14}	1.64×10^{-3}	$\pm 0.07\times10^{-3}$
I_{15}	1.25×10^{-3}	$\pm 0.30\times10^{-3}$	I_{16}	1.24×10^{-3}	$\pm 0.30\times10^{-3}$
I_{17}	6.42×10^{-4}	$\pm 3.00\times10^{-4}$	I_{18}	4.31×10^{-4}	$\pm 1.30\times10^{-4}$
I_{19}	3.00×10^{-4}	$\pm 14.0\times10^{-4}$	I_{20}	-2.02×10^{-4}	$\pm 8.00\times10^{-4}$
I_{21}	-1.00×10^{-4}	$\pm 6.00\times10^{-4}$	I_{22}	-5.80×10^{-5}	$\pm 1.50\times10^{-5}$
I_{23}	4.00×10^{-5}	$\pm 2.00\times10^{-5}$			
I_{m1}	1.14	± 0.27	I_{m2}	4.71	± 0.54
I_{m3}	8.27×10^{-1}	$\pm 0.93\times10^{-1}$	I_{m4}	2.00×10^{-1}	$\pm 0.16\times10^{-1}$
I_{m5}	1.79×10^{-1}	$\pm 0.14\times10^{-1}$	I_{m6}	1.93×10^{-1}	$\pm 0.16\times10^{-1}$

(Gravitational constants have units of newton meters)

g_1	-37.2	± 0.5	g_2	-8.44	± 0.20
g_3	1.02	± 0.50	g_4	2.49×10^{-1}	$\pm 0.25\times10^{-1}$
g_5	-2.82×10^{-2}	$\pm 0.56\times10^{-2}$			

Table A4. The expressions giving the elements of the kinetic energy matrix.

(The Abbreviated Expressions have units of kg-m^2.)

$$
\begin{aligned}
a_{11} &= I_{m1} + I_1 + I_3 * CC2 + I_7 * SS23 + I_{10} * SC23 + I_{11} * SC2 \\
&\quad + I_{20} * (S55 * (SS23 * (1 + CC4) - 1) - 2 * SC23 * C4 * SC5) \\
&\quad + I_{21} * SS23 * CC4 + 2 * (I_5 * C2 * S23 + I_{12} * C2 * C23 \\
&\quad + I_{13} * (S23 * C5 + SC23 * C4 * S5) \\
&\quad + I_{16} * C2 * (S23 * C5 + C23 * C4 * S5) \\
&\quad + I_{18} * S4 * S5 + I_{22} * (SC23 * C5 + CC23 * C4 * S5)) ; \\
&\approx 2.57 + 1.38 * CC2 + 0.30 * SS23 + 7.44\times10^{-1} * C2 * S23 . \\
a_{12} &= I_4 * S2 + I_8 * C23 + I_9 * C2 + I_{13} * S23 - I_{15} * C23 * S4 * S5 \\
&\quad + I_{16} * S2 * S4 * S5 + I_{18} * (S23 * C4 * S5 - C23 * C5) \\
&\quad + I_{19} * S23 * SC4 + I_{20} * S4 * (S23 * C4 * CC5 + C23 * SC5) \\
&\quad + I_{22} * S23 * S4 * S5 ; \\
&\approx 6.90\times10^{-1} * S2 - 1.34\times10^{-1} * C23 + 2.38\times10^{-2} * C2 . \\
a_{13} &= I_8 * C23 + I_{13} * S23 - I_{15} * C23 * S4 * S5 + I_{19} * S23 * SC4 \\
&\quad + I_{18} * (S23 * C4 * S5 - C23 * C5) + I_{22} * S23 * S4 * S5 \\
&\quad + I_{20} * S4 * (S23 * C4 * CC5 + C23 * SC5) ; \\
&\approx -1.34\times10^{-1} * C23 + -3.97\times10^{-3} * S23 . \\
a_{14} &= I_{14} * C23 + I_{15} * S23 * C4 * S5 + I_{16} * C2 * C4 * S5 \\
&\quad + I_{18} * C23 * S4 * S5 - I_{20} * (S23 * C4 * SC5 + C23 * SS5) \\
&\quad + I_{22} * C23 * C4 * S5 ; \qquad \approx 0 . \\
a_{15} &= I_{15} * S23 * S4 * C5 + I_{16} * C2 * S4 * C5 + I_{17} * S23 * S4 \\
&\quad + I_{18} * (S23 * S5 - C23 * C4 * C5) + I_{22} * C23 * S4 * C5 ; \\
&\quad \approx 0 . \\
a_{16} &= I_{23} * (C23 * C5 - S23 * C4 * S5) ; \qquad \approx 0 . \\
a_{22} &= I_{m2} + I_2 + I_6 + I_{20} * SS4 * SS5 + I_{21} * SS4 \\
&\quad + 2 * (I_5 * S3 + I_{12} * C3 + I_{18} * C5 \\
&\quad + I_{16} * (S3 * C5 + C3 * C4 * S5) + I_{22} * C4 * S5) ; \\
&\approx 6.79 + 7.44\times10^{-1} * S3 . \\
a_{23} &= I_5 * S3 + I_6 + I_{12} * C3 + I_{16} * (S3 * C5 + C3 * C4 * S5) \\
&\quad + I_{20} * SS4 * SS5 + I_{21} * SS4 + 2 * (I_{15} * C5 + I_{22} * C4 * S5) ; \\
&\approx .333 + 3.72\times10^{-1} * S3 - 1.10\times10^{-2} * C3 . \\
a_{24} &= -I_{15} * S4 * S5 - I_{16} * S3 * S4 * S5 + I_{20} * S4 * SC5 ; \\
&\quad \approx 0 . \\
a_{25} &= I_{15} * C4 * C5 + I_{16} * (C3 * S5 + S3 * C4 * C5) \\
&\quad + I_{17} * C4 + I_{22} * S5 ; \qquad \approx 0 . \\
a_{26} &= I_{23} * S4 * S5 ; \qquad \approx 0 . \\
a_{33} &= I_{m3} + I_6 + I_{20} * SS4 * SS5 + I_{21} * SS4 \\
&\quad + 2 * (I_{15} * C5 + I_{22} * C4 * S5) ; \qquad \approx 1.16 . \\
a_{34} &= -I_{15} * S4 * S5 + I_{20} * S4 * SC5 ; \\
&\quad \approx -1.25\times10^{-3} * S4 * S5 . \\
a_{35} &= I_{15} * C4 * C5 + I_{17} * C4 + I_{22} * S5 ; \\
&\quad \approx 1.25\times10^{-3} * C4 * C5 . \\
a_{36} &= I_{23} * S4 * S5 ; \qquad \approx 0 . \\
a_{44} &= I_{m4} + I_{14} - I_{20} * SS5 ; \qquad \approx 0.20 . \\
a_{45} &= 0. \\
a_{46} &= I_{23} * C5 ; \qquad \approx 0 . \\
a_{55} &= I_{m5} + I_{17} ; \qquad \approx 0.18 . \\
a_{56} &= 0. \\
a_{66} &= I_{m6} + I_{23} ; \qquad \approx 0.19 .
\end{aligned}
$$

Table A5. The expressions giving the elements of the Coriolis matrix.

(The Abbreviated Expressions have units of kg-m^2.)

$$
\begin{aligned}
b_{112} &= 2 * \{-I_5 * SC2 + I_5 * C223 + I_7 * SC23 - I_{12} * S223 \\
&\quad + I_{13} * (2 * SC23 * C5 + (1 - 2 * SS23) * C4 * S5) \\
&\quad + I_{16} * (C223 * C5 - S223 * C4 * S5) + I_{21} * SC23 * CC4 \\
&\quad + I_{20} * ((1 + CC4) * SC23 * SS5 - (1 - 2 * SS23) * C4 * SC5) \\
&\quad + I_{22} * ((1 - 2 * SS23) * C5 - 2 * SC23 * C4 * S5)\} \\
&\quad + I_{10} * (1 - 2 * SS23) + I_{11} * (1 - 2 * SS2) ; \\
&\approx -2.76 * SC2 + 7.44\times10^{-1} * C223 + 0.60 * SC23 \\
&\quad - 2.13\times10^{-2} * (1 - 2 * SS23) . \\
b_{113} &= 2 * \{I_5 * C2 * C23 + I_7 * SC23 - I_{12} * C2 * S23 \\
&\quad + I_{13} * (2 * SC23 * C5 + (1 - 2 * SS23) * C4 * S5) \\
&\quad + I_{16} * C2 * (C23 * C5 - S23 * C4 * S5) + I_{21} * SC23 * CC4 \\
&\quad + I_{20} * ((1 + CC4) * SC23 * SS5 - (1 - 2 * SS23) * C4 * SC5) \\
&\quad + I_{22} * ((1 - 2 * SS23) * C5 - 2 * SC23 * C4 * S5)\} \\
&\quad + I_{10} * (1 - 2 * SS23)\} ; \\
&\approx 7.44\times10^{-1} * C2 * C23 + 0.60 * SC23 \\
&\quad + 2.20\times10^{-2} * C2 * S23 - 2.13\times10^{-2} * (1 - 2 * SS23) . \\
b_{114} &= 2 * \{-I_{15} * SC23 * S4 * S5 - I_{16} * C2 * C23 * S4 * S5 \\
&\quad + I_{18} * C4 * S5 - I_{20} * (SS23 * SS5 * SC4 - SC23 * S4 * SC5) \\
&\quad - I_{22} * CC23 * S4 * S5 - I_{21} * SS23 * SC4\} ; \\
&\approx -2.50\times10^{-3} * SC23 * S4 * S5 + 8.60\times10^{-4} * C4 * S5 \\
&\quad - 2.48\times10^{-3} * C2 * C23 * S4 * S5 . \\
b_{115} &= 2 * \{I_{20} * (SC5 * (CC4 * (1 - CC23) - CC23) \\
&\quad - SC23 * C4 * (1 - 2 * SS5)) - I_{15} * (SS23 * S5 - SC23 * C4 * C5) \\
&\quad - I_{16} * C2 * (S23 * S5 - C23 * C4 * C5) + I_{18} * S4 * C5 \\
&\quad + I_{22} * (CC23 * C4 * C5 - SC23 * S5)\} ; \\
&\approx -2.50\times10^{-3} * (SS23 * S5 - SC23 * C4 * C5) \\
&\quad - 2.48\times10^{-3} * C2 * (SS3 * S5 - C23 * C4 * C5) \\
&\quad + 8.60\times10^{-4} * S4 * C5. \\
b_{116} &= 0 . \\
b_{125} &= 2 * \{-I_5 * S23 + I_{13} * C23 + I_{15} * S23 * S4 * S5 \\
&\quad + I_{18} * (C23 * C4 * S5 + S23 * C5) + I_{19} * C23 * SC4 \\
&\quad + I_{20} * S4 * (C23 * C4 * CC5 - S23 * SC5) \\
&\quad + I_{22} * C23 * S4 * S5)\} ; \\
&\approx 2.67\times10^{-1} * S23 - 7.58\times10^{-5} * C23 . \\
b_{124} &= -I_{14} * 2 * S23 * S4 * S5 + I_{19} * S23 * (1 - (2 * SS4)) \\
&\quad + I_{20} * S23 * (1 - 2 * SS4 * CC5) - I_{14} * S23 ; \qquad \approx 0 . \\
b_{125} &= I_{17} * C23 * S4 + I_{18} * 2 * (S23 * C4 * C5 + C23 * S5) \\
&\quad + I_{20} * S4 * (C23 * (1 - 2 * SS5) - S23 * C4 * 2 * SC5) ; \\
&\quad \approx 0 . \\
b_{126} &= -I_{23} * (S23 * C5 + C23 * C4 * S5) ; \qquad \approx 0 . \\
b_{134} &= b_{124} . \qquad b_{135} = b_{125} . \qquad b_{136} = b_{126} . \\
b_{145} &= 2 * \{I_{15} * S23 * C4 * C5 + I_{16} * C2 * C4 * C5 \\
&\quad + I_{18} * C23 * S4 * C5 + I_{22} * C23 * C4 * C5\} + I_{17} * S23 * C4 \\
&\quad - I_{20} * (S23 * C4 * (1 - 2 * SS5) + 2 * C23 * SC5) ; \\
&\quad \approx 0 . \\
b_{146} &= I_{23} * S23 * S4 * S5 ; \qquad \approx 0 . \\
b_{156} &= -I_{23} * (C23 * S5 + S23 * C4 * C5) ; \qquad \approx 0 . \\
b_{212} &= 0 . \qquad b_{213} = 0 . \\
b_{214} &= I_{14} * S23 + I_{19} * S23 * (1 - (2 * SS4)) \\
&\quad + 2 * \{-I_{18} * C23 * C4 * S5 + I_{16} * S2 * C4 * S5 \\
&\quad + I_{20} * (S23 * (CC5 * CC4 - 0.5) + C23 * C4 * SC5) \\
&\quad + I_{22} * S23 * C4 * S5)\} ; \\
&\approx 1.64\times10^{-3} * S23 - 2.50\times10^{-3} * C23 * C4 * S5 + \\
&\quad 2.48\times10^{-3} * S2 * C4 * S5 + 0.30\times10^{-3} * S23 * (1 - (2 * SS4)) . \\
b_{215} &= 2 * \{-I_{15} * C23 * S4 * C5 + I_{22} * S23 * S4 * C5 \\
&\quad + I_{16} * S2 * S4 * C5\} - I_{17} * C23 * S4 \\
&\quad + I_{20} * (C23 * S4 * (1 - 2 * SS5) - 2 * S23 * SC4 * SC5)\} ; \\
&\approx -2.50\times10^{-3} * C23 * S4 * C5 + 2.48\times10^{-3} * S2 * S4 * C5 \\
&\quad - 6.42\times10^{-4} * C23 * S4 . \\
b_{216} &= -b_{126} . \\
b_{223} &= 2 * \{-I_{12} * S3 + I_5 * C3 + I_{16} * (C3 * C5 - S3 * C4 * S5)\} ; \\
&\approx 2.20\times10^{-2} * S3 + 7.44\times10^{-1} * C3 . \\
b_{224} &= 2 * \{-I_{16} * C3 * S4 * S5 + I_{20} * SC4 * SS5 \\
&\quad + I_{21} * SC4 - I_{22} * S4 * S5\} ; \\
&\approx -2.48\times10^{-3} * C3 * S4 * S5 . \\
b_{225} &= 2 * \{-I_{15} * S5 + I_{16} * (C3 * C4 * C5 - S3 * S5) \\
&\quad + I_{20} * S54 * SC5 + I_{22} * C4 * C5\} ; \\
&\approx -2.50\times10^{-3} * S5 + 2.48\times10^{-3} * (C3 * C4 * C5 - S3 * S5) . \\
b_{226} &= 0 . \qquad b_{234} = b_{224} . \\
b_{235} &= b_{225} . \qquad b_{236} = 0 . \\
b_{245} &= 2 * \{-I_{15} * S4 * C5 - I_{16} * S3 * S4 * C5\} \\
&\quad - I_{17} * S4 + I_{20} * S4 * (1 - 2 * SS5) ; \qquad \approx 0 . \\
b_{246} &= I_{23} * C4 * S5 ; \qquad \approx 0 . \\
b_{256} &= I_{23} * S4 * C5 ; \qquad \approx 0 . \\
b_{312} &= 0 . \qquad b_{313} = 0 . \\
b_{314} &= 2 * \{-I_{15} * C23 * C4 * S5 + I_{22} * S23 * C4 * S5 \\
&\quad + I_{20} * (S23 * (CC5 * CC4 - 0.5) + C23 * C4 * SC5)\} \\
&\quad + I_{14} * S23 + I_{19} * S23 * (1 - (2 * SS4)) ; \\
&\approx -2.50\times10^{-3} * C23 * C4 * S5 + 1.64\times10^{-3} * S23 \\
&\quad + 0.30\times10^{-3} * S23 * (1 - 2 * SS4) . \\
b_{315} &= 2 * \{-I_{15} * C23 * S4 * C5 + I_{22} * S23 * S4 * C5) \\
&\quad - I_{17} * C23 * S4 \\
&\quad + I_{20} * S4 * (C23 * (1 - 2 * SS5) - 2 * S23 * C4 * SC5)\} ; \\
&\approx -2.50\times10^{-3} * C23 * S4 * C5 - 6.42\times10^{-4} * C23 * S4 . \\
b_{316} &= -b_{126} . \qquad b_{323} = 0 . \\
b_{324} &= 2 * \{I_{20} * SC4 * SS5 + I_{21} * SC4 - I_{22} * S4 * S5\} ; \\
&\quad \approx 0 . \\
b_{325} &= 2 * \{-I_{15} * S5 + I_{20} * SS4 * SC5 + I_{22} * C4 * C5\} ; \\
&\quad \approx -2.50\times10^{-3} * S5 . \\
b_{326} &= 0 . \qquad b_{334} = b_{224} . \\
b_{335} &= b_{325} . \qquad b_{336} \approx 0 . \\
b_{345} &= -I_{15} * 2 * S4 * C5 - I_{17} * S4 + I_{20} * S4 * (1 - 2 * SS5) ; \\
&\quad \approx -2.50\times10^{-3} * S4 * C5 . \\
b_{346} &= b_{246} . \qquad b_{356} = b_{256} . \qquad b_{412} = -b_{214} . \\
b_{413} &= -b_{314} . \qquad b_{414} = 0 . \\
b_{415} &= -I_{20} * (S23 * C4 * (1 - 2 * SS5) + 2 * C23 * SC5) \\
&\quad - I_{17} * S23 * C4 ; \\
&\quad \approx -6.42\times10^{-4} * S23 * C4 . \\
b_{416} &= -b_{146} . \qquad b_{423} = -b_{324} . \qquad b_{424} = 0 . \\
b_{425} &= I_{17} * S4 + I_{20} * S4 * (1 - 2 * SS5) ; \\
&\quad \approx 6.42\times10^{-4} * S4 . \\
b_{426} &= -b_{246} . \qquad b_{434} = 0 . \\
b_{435} &= b_{425} . \qquad b_{436} = -b_{346} . \\
b_{445} &= -I_{20} * 2 * SC5 ; \qquad \approx 0 . \\
b_{446} &= 0 ; \\
b_{456} &= -I_{23} * S5 ; \qquad \approx 0 . \\
b_{512} &= -b_{215} . \qquad b_{513} = -b_{315} . \qquad b_{514} = -b_{415} . \\
b_{515} &= 0 . \qquad b_{516} = -b_{156} . \qquad b_{523} = -b_{325} . \\
b_{524} &= -b_{425} . \qquad b_{525} = 0 . \qquad b_{526} = -b_{256} . \\
b_{534} &= b_{524} . \qquad b_{535} = 0 . \qquad b_{536} = -b_{356} . \\
b_{545} &= 0 . \qquad b_{546} = -b_{456} . \qquad b_{556} = 0 . \\
b_{612} &= b_{126} . \qquad b_{613} = b_{136} . \qquad b_{614} = b_{146} . \\
b_{615} &= b_{156} . \qquad b_{616} = 0 . \qquad b_{623} = 0 . \\
b_{624} &= b_{246} . \qquad b_{625} = b_{256} . \qquad b_{626} = 0 . \\
b_{634} &= b_{624} . \qquad b_{635} = b_{625} . \qquad b_{636} = 0 . \\
b_{645} &= b_{456} . \qquad b_{646} = 0 . \qquad b_{656} = 0 .
\end{aligned}
$$

Table A6. The expressions for the terms of the centrifugal matrix.

(The Abbreviated Expressions have units of kg-m^2.)

$$
\begin{aligned}
c_{11} &= 0 . \\
c_{12} &= +I4 * C2 - I8 * S23 - I9 * S2 + I13 * C23 \\
&\quad + I15 * S23 * S4 * S5 + I16 * C2 * S4 * S5 \\
&\quad + I18 * (C23 * C4 * S5 + S23 * C5) + I19 * C23 * SC4 \\
&\quad + I20 * S4 * (C23 * C4 * CC5 - S23 * SC5) \\
&\quad + I22 * C23 * S4 * S5 ; \\
&\approx 6.90\times10^{-1} * C2 + 1.34\times10^{-1} * S23 - 2.38\times10^{-2} * S2 . \\
c_{13} &= 0.5 * b_{123} . \\
c_{14} &= -I15 * S23 * S4 * S5 - I16 * C2 * S4 * S5 \\
&\quad + I18 * C23 * C4 * S5 + I20 * S23 * S4 * SC5 \\
&\quad - I22 * C23 * S4 * S5 ; \qquad \approx 0 . \\
c_{15} &= -I15 * S23 * S4 * S5 - I16 * C2 * S4 * S5 ; \\
&\quad + I18 * (S23 * C5 + C23 * C4 * S5) - I22 * C23 * S4 * S5 \\
&\quad \approx 0 . \\
c_{16} &= 0 . \qquad c_{21} = -0.5 * b_{112} . \\
c_{22} &= 0 . \qquad c_{23} = 0.5 * b_{223} . \\
c_{24} &= -I15 * C4 * S5 - I16 * S3 * C4 * S5 + I20 * C4 * SC5 ; \\
&\quad \approx 0 . \\
c_{25} &= -I15 * C4 * S5 + I16 * (C3 * C5 - S3 * C4 * S5) \\
&\quad + I22 * C5 ; \qquad \approx 0 . \\
c_{26} &= 0 . \qquad c_{31} \geq -0.5 * b_{113} . \\
c_{32} &= -c_{23} . \qquad c_{33} = 0 . \\
c_{34} &= -I15 * C4 * S5 + I20 * C4 * SC5 ; \\
&\quad \approx -1.25\times10^{-3} * C4 * S5 . \\
c_{35} &= -I15 * C4 * S5 + I22 * C5 ; \qquad \approx c_{34} . \\
c_{36} &= 0 . \qquad c_{41} = -0.5 * b_{114} . \qquad c_{42} = -0.5 * b_{224} \\
c_{43} &= 0.5 * b_{423} . \qquad c_{44} = 0 . \qquad c_{45} = 0 . \\
c_{46} &= 0 . \qquad c_{51} = -0.5 * b_{115} . \qquad c_{52} = -0.5 * b_{225} \\
c_{53} &= 0.5 * b_{323} . \qquad c_{54} = -0.5 * b_{445} . \qquad c_{55} = 0 . \\
c_{56} &= 0 . \qquad c_{61} = 0 . \qquad c_{62} = 0 . \\
c_{63} &= 0 . \qquad c_{64} = 0 . \qquad c_{65} = 0 . \\
c_{66} &= 0 .
\end{aligned}
$$

Table A7. Gravity Terms.

(The Abbreviated Expressions have units of newton-meters.)

$$
\begin{aligned}
g_1 &= 0 . \\
g_2 &= g1 * C2 + g2 * S23 + g3 * S2 + g4 * C23 ; \\
&\quad + g5 * (S23 * C5 + C23 * C4 * S5) \\
&\quad \approx -37.2 * C2 - 8.4 * S23 + 1.02 * S2 . \\
g_3 &= g2 * S23 + g4 * C23 + g5 * (S23 * C5 + C23 * C4 * S5) ; \\
&\quad \approx -8.4 * S23 + 0.25 * C23 . \\
g_4 &= -g5 * S23 * S4 * S5 ; \\
&\quad \approx 2.8\times10^{-2} * S23 * S4 * S5 . \\
g_5 &= g5 * (C23 * S5 + S23 * C4 * C5) ; \\
&\quad \approx -2.8\times10^{-2} * (C23 * S5 + S23 * C4 * C5) . \\
g_6 &= 0 .
\end{aligned}
$$

Are We Done Yet?

- Let us use the tools to solve our grasping example

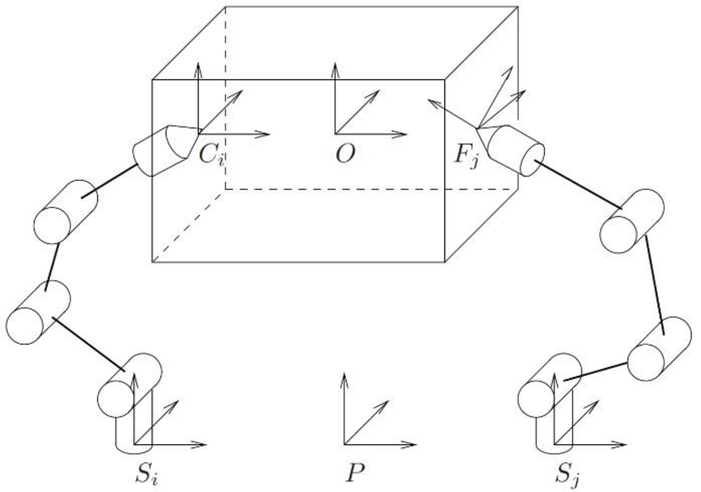


Figure 5.14: Grasp coordinate frames.

Are We Done Yet?

- Let us use the tools to solve our grasping example
  - Suppose that each arm has 3 joints with motors installed.

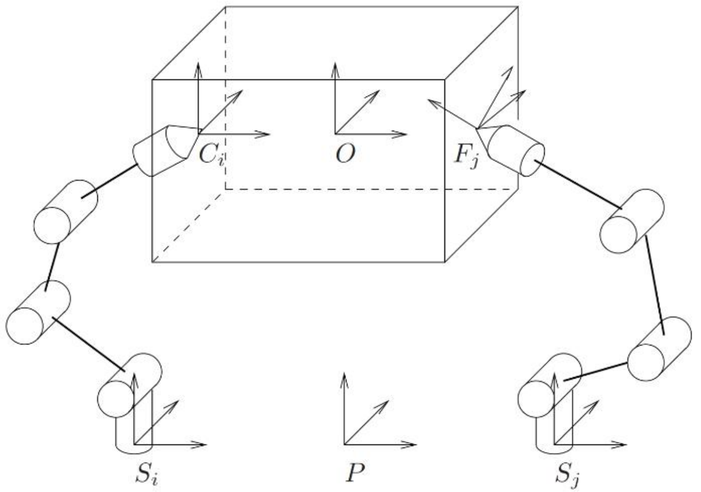


Figure 5.14: Grasp coordinate frames.

Are We Done Yet?

- Let us use the tools to solve our grasping example
  - Suppose that each arm has 3 joints with motors installed.
  - The **degree-of-freedom** (independent coordinates) of the system is: $2 \times 3 + 6 = 12$ including both two arms and the rigid object.

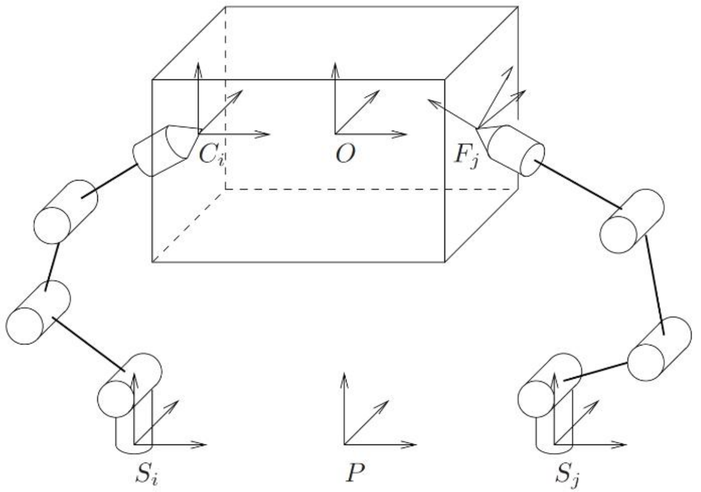


Figure 5.14: Grasp coordinate frames.

Are We Done Yet?

- Let us use the tools to solve our grasping example
  - Suppose that each arm has 3 joints with motors installed.
  - The **degree-of-freedom** (independent coordinates) of the system is: $2 \times 3 + 6 = 12$ including both two arms and the rigid object.
  - If we would like control the robot to hold the box firmly, the degree-of-freedom will reduce due to constraints.

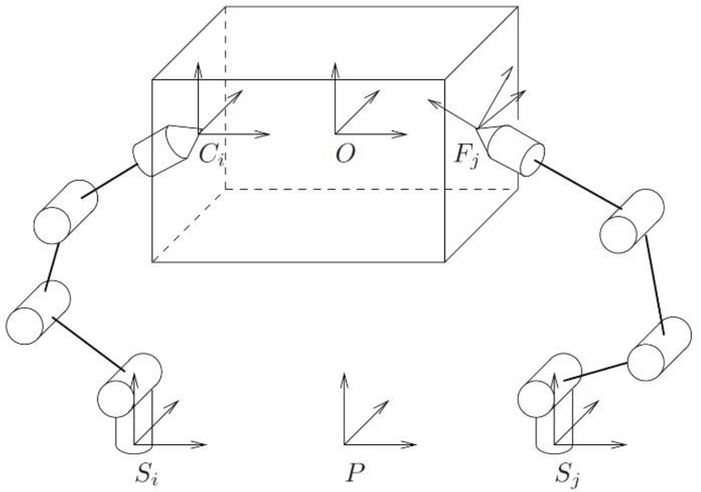


Figure 5.14: Grasp coordinate frames.

Are We Done Yet?

- Let us use the tools to solve our grasping example
  - Suppose that each arm has 3 joints with motors installed.
  - The **degree-of-freedom** (independent coordinates) of the system is: $2 \times 3 + 6 = 12$ including both two arms and the rigid object.
  - If we would like control the robot to hold the box firmly, the degree-of-freedom will reduce due to constraints.
  - How to solve FD/ID for this system?

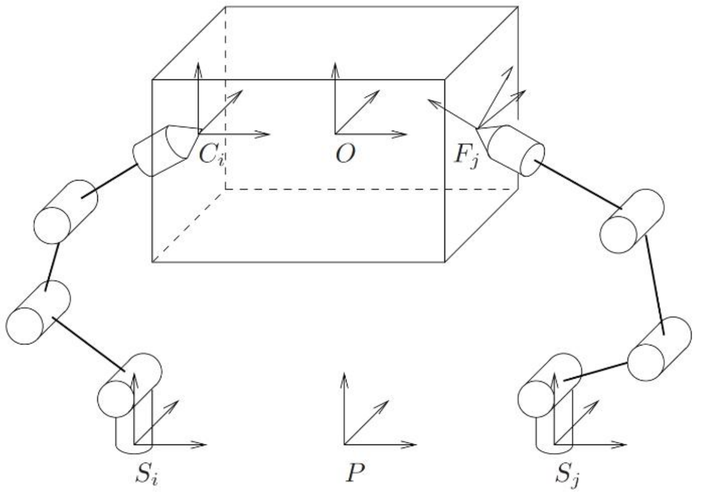


Figure 5.14: Grasp coordinate frames.

Are We Done Yet?

- Let us use the tools to solve our grasping example
  - Suppose that each arm has 3 joints with motors installed.
  - The **degree-of-freedom** (independent coordinates) of the system is: $2 \times 3 + 6 = 12$ including both two arms and the rigid object.
  - If we would like control the robot to hold the box firmly, the degree-of-freedom will reduce due to constraints.
  - How to solve FD/ID for this system?
  - Additionally, if we are to keep the box static, among many τ candidates, we want to choose one by the contact force and torque. However, our previously developed tool does not tell us about these quantities!

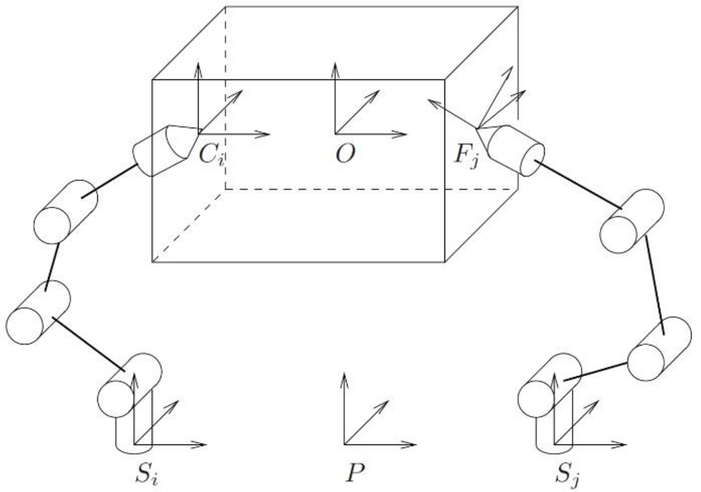


Figure 5.14: Grasp coordinate frames.

Constrained Lagrangian Method

Constrained Lagrangian Method

- Our previous Lagrangian method uses independent coordinates to describe the state of the system

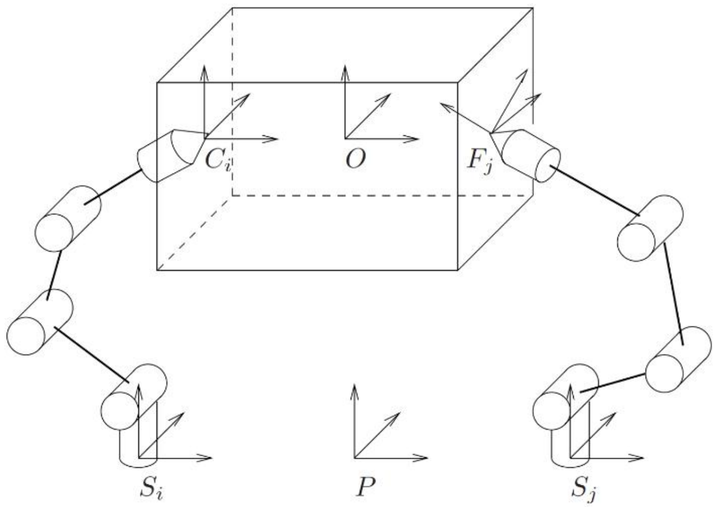


Figure 5.14: Grasp coordinate frames.

Constrained Lagrangian Method

- Our previous Lagrangian method uses independent coordinates to describe the state of the system
- Sometimes, we are interested in the situation happening not at these independent coordinates (e.g., in the grasping example, we are interested in the internal forces at the contact points).

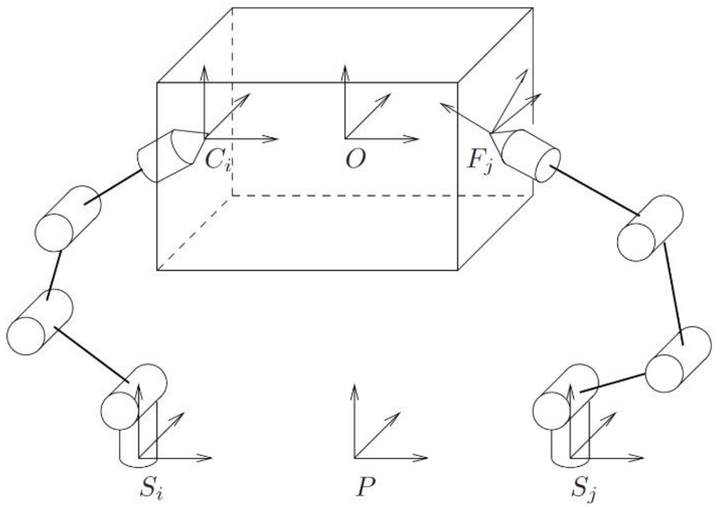


Figure 5.14: Grasp coordinate frames.

Constrained Lagrangian Method

- Our previous Lagrangian method uses independent coordinates to describe the state of the system
- Sometimes, we are interested in the situation happening not at these independent coordinates (e.g., in the grasping example, we are interested in the internal forces at the contact points).
- We augment our previous Lagrangian methods to answer such query.

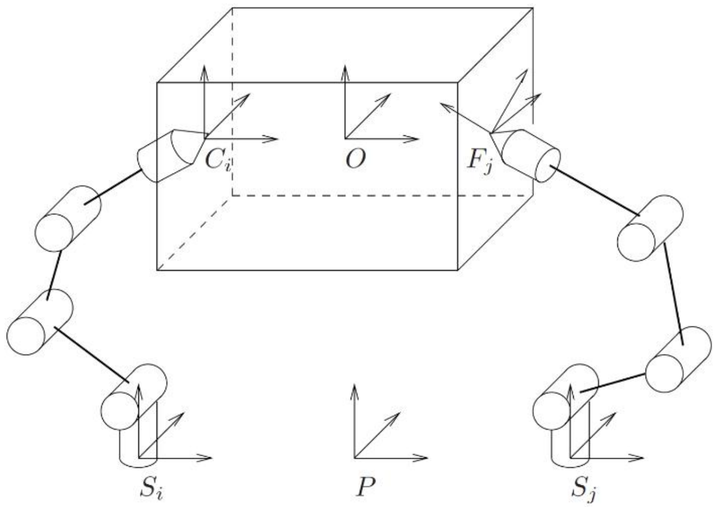


Figure 5.14: Grasp coordinate frames.

Review: Contact Coordinate Frame

- We build a **contact frame** C_i at each contact point
- The z-axis of the frame points inward along *surface normal*
- When recording force and torque at the contact point, it is natural to set C_i as the *observer's frame*, i.e.,

$$\boldsymbol{F}^{C_i} = \begin{bmatrix} \boldsymbol{f}^{C_i} \\ \boldsymbol{\tau}^{C_i} \end{bmatrix}$$

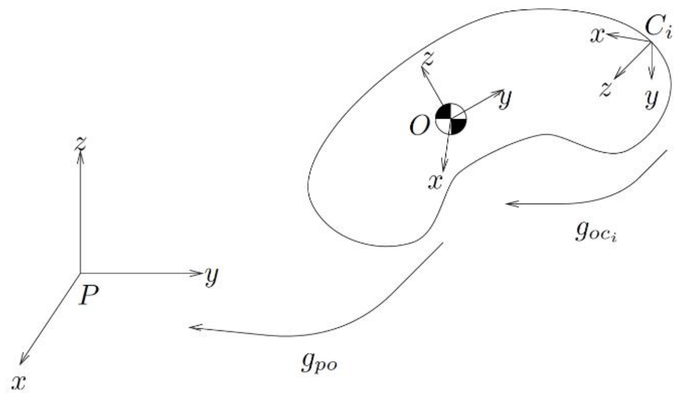


Figure 5.2: Coordinate frames for contact and object forces.

Step 1: Adding Generalized Coordinates for Constraint Force

- We assign additional generalized coordinates at the constraints.
  - E.g., we add the 6D pose of the contact frames at the end of the two finger tips as the generalized coordinates.
  - So the generalized velocities are the twists at the finger tips $\boldsymbol{\xi}^{C_i}$.

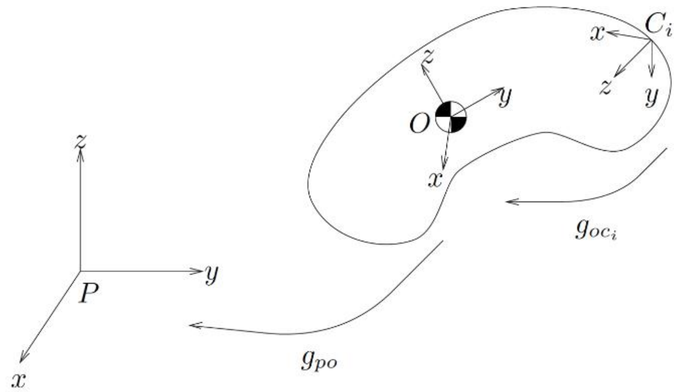


Figure 5.2: Coordinate frames for contact and object forces.

Step 1: Adding Generalized Coordinates for Constraint Force

- We assign additional generalized coordinates at the constraints.
  - E.g., we add the 6D pose of the contact frames at the end of the two finger tips as the generalized coordinates.
  - So the generalized velocities are the twists at the finger tips $\boldsymbol{\xi}^{C_i}$.
- These additional generalized coordinates have their corresponding generalized forces (our query).
  - In our example, the additional generalized forces are the internal forces ($\boldsymbol{f}^{C_i}$ and $\boldsymbol{\tau}^{C_i}$) at the contact points $\boldsymbol{F}^{C_i}$.

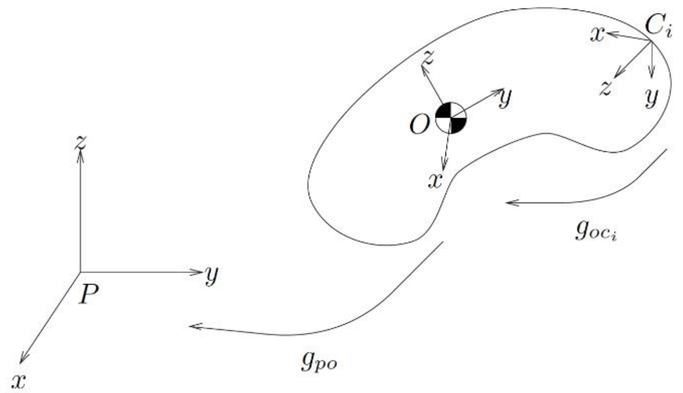


Figure 5.2: Coordinate frames for contact and object forces.

Step 2: Adding Constraints

- While assigning the additional generalized coordinates, we also need to add constraints.
  - In our example, $\boldsymbol{\xi}^{C_i}$ must conform to two sets of constraints:
    - All contact frames must move in accordance with the movement of the object;
    - All contact frames must move in accordance with the movement of the arms.

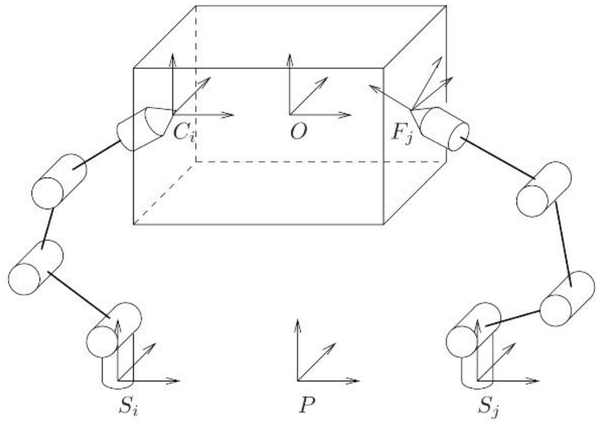


Figure 5.14: Grasp coordinate frames.

Step 2: Adding Constraints

"All contact frames must move in accordance with the movement of the object."

- Assume the body frame twist of the box is $\boldsymbol{\xi}^b$.
- Then, $\boldsymbol{\xi}^{C_i} = [\mathrm{Ad}_{\mathrm{C_i}\to\mathrm{b}}]\boldsymbol{\xi}^{\mathrm{b}} \quad \forall \mathrm{i}.$

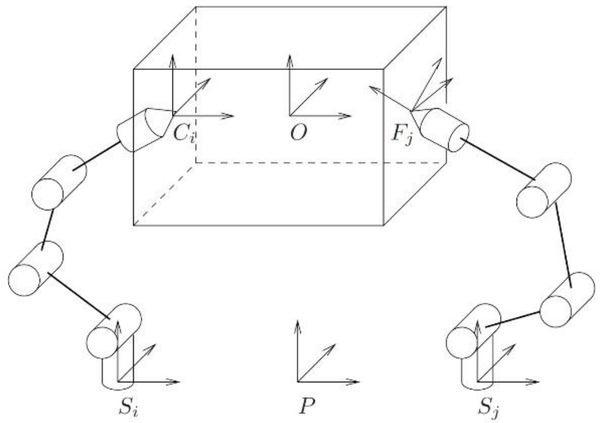


Figure 5.14: Grasp coordinate frames.

Step 2: Adding Constraints

"All contact frames must move in accordance with the movement of the object."

- Assume the body frame twist of the box is $\boldsymbol{\xi}^b$.
- Then, $\boldsymbol{\xi}^{C_i} = [\mathrm{Ad}_{\mathrm{C_i}\to\mathrm{b}}]\boldsymbol{\xi}^{\mathrm{b}} \quad \forall \mathrm{i}.$

"All contact frames must move in accordance with the movement of the arms."

- The body frame twist of the contact frame is $\boldsymbol{\xi}^{C_i}$.
- We can develop the hand Jacobian as $\boldsymbol{\xi}^{C_i} = J^{C_i}(\theta)\dot{\theta} \quad \forall i.$

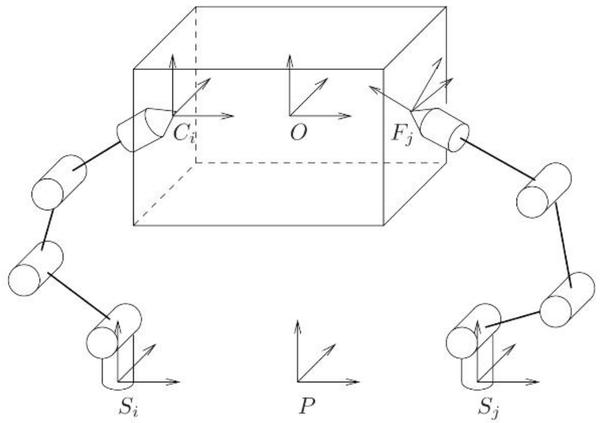


Figure 5.14: Grasp coordinate frames.

Constraints for Augmented Generalized Velocities

- We put all constraints over generalized velocities in a linear equation:

$$A(q)\dot{q} = 0, \quad \text{where } A(q) \in \mathbb{R}^{m \times n}$$

  - m: number of constraints
  - n: dimension of generalized coordinates
- After some infinitesimal time internal δt, the displacement of the generalized coordinates is $\delta q = \dot{q}\,\delta t$. We rewrite the constraints as

$$A(q)\delta q = 0$$

A Few More Words on Constraints

- A constraint on a mechanical system restricts the motion of the system by limiting the set of paths that the system can follow.
- **Holonomic constraints**:
  - Roughly speaking, the constraint $A(q)\delta q = 0$ does not depend on $\dot{q}$ (e.g., our grasping example)
  - Strictly speaking, a constraint is said to be **holonomic** if it restricts the motion of the system to a smooth hyper-surface in the (unconstrained) space $\mathcal{C}$.
- Example of **Nonholonomic constraints**:
  - No-side-slip constraint of wheeled robot: Due to *rolling friction*, a car can only move along the velocity direction, thus the constraint is velocity-based ($A(q, \dot{q})\delta q = 0$), and cannot be written as purely position-based.

Variational Method

Some Tricky (but Sloppy) Math is Coming!

source: Plants vs Zombies V3

Background: Unconstrained Function Optimization

- For some $f : \mathbb{R}^n \to \mathbb{R}$, let us solve the optimization problem

$$\underset{x}{\text{minimize}} \quad f(x)$$

- The first-order Taylor's expansion of $f(x)$ is

$$f(x + \delta x) - f(x) = \nabla f(x)^T \delta x + o(\delta x)$$

- The optimal point must be a *stationary point*, i.e., *the first-order term diminishes*:

$$\nabla f(x^*)^T \delta x = 0$$

- Because the problem is unconstrained, δx can vary along any direction.
- Therefore, $\nabla f(x^*) = 0$ is a necessary condition.

Background: Constrained Function Optimization

- For some $f : \mathbb{R}^n \to \mathbb{R}$, let us solve the optimization problem

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & Ax = 0, \qquad A \in \mathbb{R}^{m \times n} \end{aligned}$$

- The first-order Taylor's expansion of $f(x)$ is

$$f(x + \delta x) - f(x) = \nabla f(x)^T \delta x + o(\delta x)$$

- The optimal point must be a *stationary point*, i.e., *the first-order term diminishes for variations in the constraint set*:

$$\forall \delta x : A\delta x = 0, \nabla f(x^*)^T \delta x = 0$$

- $\delta x \in Null(A)$ and $\nabla f(x^*)^T \perp \delta x$ implies that $\nabla f(x^*) = -A^T \lambda$ for some $\lambda \in \mathbb{R}^m$

Background: Constrained Function Optimization

- Optimality Condition:

$$\begin{cases} \nabla f(x) + A^T \lambda = 0 & \text{(stationarity)} \\ Ax = 0 & \text{(feasibility)} \end{cases}$$

Principle of Stationary Action (Unconstrained)

- Given some $q : [t_1, t_2] \to \mathcal{C}$, assume a small perturbation function $\delta q : [t_1, t_2] \to \mathcal{C}$ such that

$$\delta q(t_1) = \delta q(t_2) = 0$$

- For the functional $S[q] = \int_{t_1}^{t_2} L(q, \dot{q}) \mathrm{d}t$, the first-order variation is

$$\begin{aligned}
\delta S = S[q + \delta q] - S[q] &= \int_{t_1}^{t_2} L(q + \delta q, \dot{q} + \delta \dot{q}) - L(q, \dot{q}) \\
&= \int_{t_1}^{t_2} L(q, \dot{q}) + \left(\frac{\partial L}{\partial q}\right)^T \delta q + \left(\frac{\partial L}{\partial \dot{q}}\right)^T \delta \dot{q} - L(q, \dot{q}) + o(q) \\
&= \int_{t_1}^{t_2} \left(\frac{\partial L}{\partial q}\right)^T \delta q + \left(\frac{\partial L}{\partial \dot{q}}\right)^T \delta \frac{\mathrm{d}}{\mathrm{d}t} q + o(q) = \int_{t_1}^{t_2} \left(\frac{\partial L}{\partial q}\right)^T \delta q + \left(\frac{\partial L}{\partial \dot{q}}\right)^T \frac{\mathrm{d}}{\mathrm{d}t} \delta q + o(q) \\
&= \int_{t_1}^{t_2} \left(\frac{\partial L}{\partial q}\right)^T \delta q + \frac{\partial L}{\partial \dot{q}} \delta q \Big|_{t_1}^{t_2} - \left(\frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial L}{\partial \dot{q}}\right)^T \delta q + o(t) = \int_{t_1}^{t_2} \left(\frac{\partial L}{\partial q} - \frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial L}{\partial \dot{q}}\right)^T \delta q + o(t)
\end{aligned}$$

- Because the problem is unconstrained, δq can vary along any direction.
- Therefore, $\frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = 0$ is the condition for stationary solution.

Principle of Stationary Action (Holonomically Constrained)

- Given some $q : [t_1, t_2] \to \mathcal{C}$, assume a small perturbation function $\delta q : [t_1, t_2] \to \mathcal{C}$ such that

$$\delta q(t_1) = \delta q(t_2) = 0, \qquad A(q)\delta q = 0$$

- For the functional $S[q] = \int_{t_1}^{t_2} L(q, \dot{q})\mathrm{d}t$, the first-order variation is

$$\delta S = S[q + \delta q] - S[q] = \int_{t_1}^{t_2} \left(\frac{\partial L}{\partial q} - \frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial L}{\partial \dot{q}} \right)^T \delta q + o(t)$$

- The optimal point must be a *stationary point*, i.e., *the first-order term diminishes for variations in the constraint set*:

$$\forall \delta q : A(q)\delta q = 0, \int_{t_1}^{t_2} \left(\frac{\partial L}{\partial q} - \frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial L}{\partial \dot{q}} \right)^T \delta q = 0$$

- $\forall \delta q \in Null(A(q))$, $\frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} \perp \delta q$, which implies that $\frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = -A(q)^T \lambda$ for some $\lambda \in \mathbb{R}^m$

Principle of Stationary Action (Holonomically Constrained)

- Stationary point condition (no external force):

$$\begin{cases} \dfrac{\mathrm{d}}{\mathrm{d}t}\dfrac{\partial L}{\partial \dot{q}} - \dfrac{\partial L}{\partial q} + A(q)^T\lambda = 0 & \text{(stationarity)} \\ A(q)\dot{q} = 0 & \text{(feasibility)} \end{cases}$$

- When external force exists,

$$\begin{cases} \dfrac{\mathrm{d}}{\mathrm{d}t}\dfrac{\partial L}{\partial \dot{q}} - \dfrac{\partial L}{\partial q} + A(q)^T\lambda = \boldsymbol{F} & \text{(stationarity)} \\ A(q)\dot{q} = 0 & \text{(feasibility)} \end{cases}$$

- **Constraint Force**: $\boldsymbol{F}_{cons} = A(q)^T\lambda$
- Note that $\boldsymbol{F}_{cons}^T\dot{q} = 0$, thus internal force does no work.

Principle of Stationary Action (Holonomically Constrained)

- The feasibility constraint can be rewritten as:

$$\dot{A(q)}\dot{q} + A(q)\ddot{q} = 0$$

- The system of equations is thus:

$$\begin{cases} \dfrac{\mathrm{d}}{\mathrm{d}t}\dfrac{\partial L}{\partial \dot{q}} - \dfrac{\partial L}{\partial q} + A(q)^T\lambda = \boldsymbol{F} & \text{(stationarity)} \\ \dot{A(q)}\dot{q} + A(q)\ddot{q} = 0 & \text{(feasibility)} \end{cases}$$

Principle of Stationary Action (Holonomically Constrained)

- Our object manipulation example:

$$\begin{cases} M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q,\dot{q}) + A(q)^T\lambda = \boldsymbol{F} & \text{(stationarity)} \\ \dot{A(q)}\dot{q} + A(q)\ddot{q} = 0 & \text{(feasibility)} \end{cases}$$

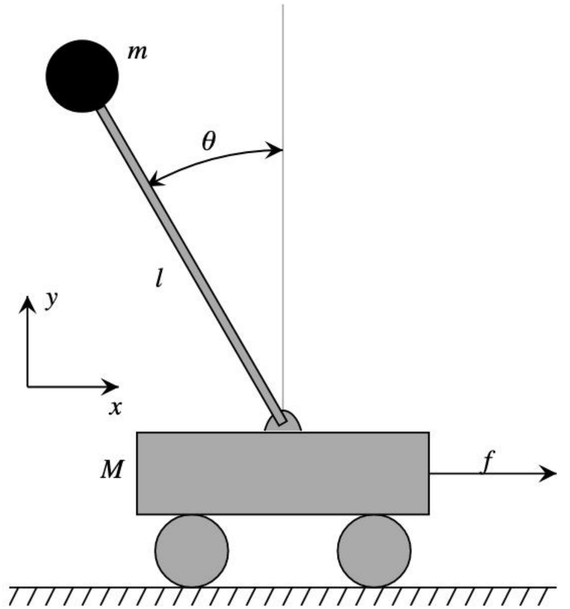
  therefore,

$$\lambda = (AM^{-1}A^T)^{-1}(AM^{-1}(F - C\dot{q} - N) + \dot{A}\dot{q})$$

  where $AM^{-1}A^T$ is full rank if the constraints are independent.
- Note: $M(q)$, $C(q,\dot{q})$, and $g(q,\dot{q})$ are computed from the two arms and the object.

Example: Cart Pole

- Kinetic energy: $T = \frac{1}{2}Mv_1^2 + \frac{1}{2}mv_2^2$
- Assume the joint position is $[x_1(t), 0]^T$ and the point mass position is $[x_2(t), y_2(t)]$, then
  - $v_1^2 = \dot{x}_1^2$
  - $v_2^2 = \dot{x}_2^2 + \dot{y}_2^2$
- Potential energy: $V = mgy_2$
- Constraint: $C(x_1, x_2, y_2) = (x_1 - x_2)^2 + y_2^2 - l^2$
  - $A(q)\dot{q} = \nabla_q C(q)\dot{q}$

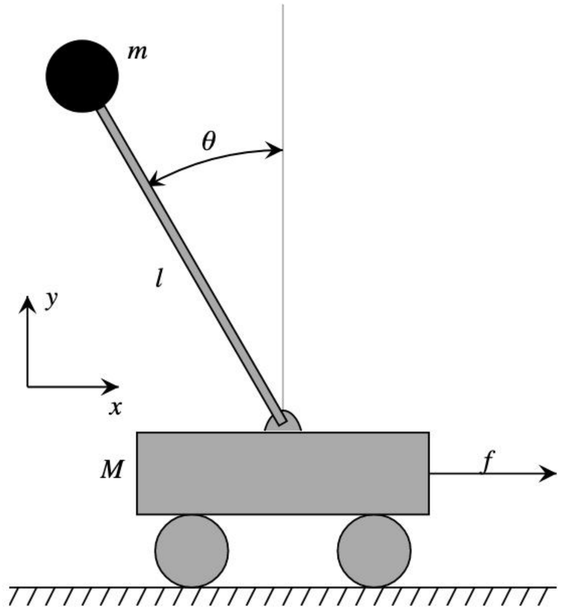


A schematic drawing of the inverted pendulum on a cart. The rod is considered massless.

https://en.wikipedia.org/wiki/Inverted_pendulum

Example: Cart Pole

```
from sympy import *
from sympy.physics.vector import dynamicsymbols
from sympy import diff, Symbol

M, m, t, g, l, lmd, f = symbols('M m t g l lmd f')

x1 = dynamicsymbols('x1')
x2 = dynamicsymbols('x2')
th = dynamicsymbols('th')
q = Matrix([x1, x2, th])

dqdt = diff(q, t)
dx1dt = dqdt[0]
dx2dt = dqdt[1]
dthdt = dqdt[2]
y2 = l * cos(th)
dy2dt = diff(y2, t)

T = 0.5 * M * dx1dt * dx1dt + 0.5 * m * (dx2dt * dx2dt + dy2dt * dy2dt)
V = m * g * y2
C = (x1-x2)**2+y2**2-l**2
L = T - V + C*lmd

Lagrangian = Lag1 = diff(diff(L, dqdt), t)- diff(L, q)


F = Matrix([f, 0, 0])
stationarity = Lagrangian - F

A = diff(C, q).T

feasibility = diff(A, t)* dqdt + A * diff(dqdt, t)

ddqdt = diff(dqdt, t)
sol = simplify(solve((stationarity[0], stationarity[1], stationarity[2], feasibility), lmd,
ddqdt[0], ddqdt[1], ddqdt[2]))
```



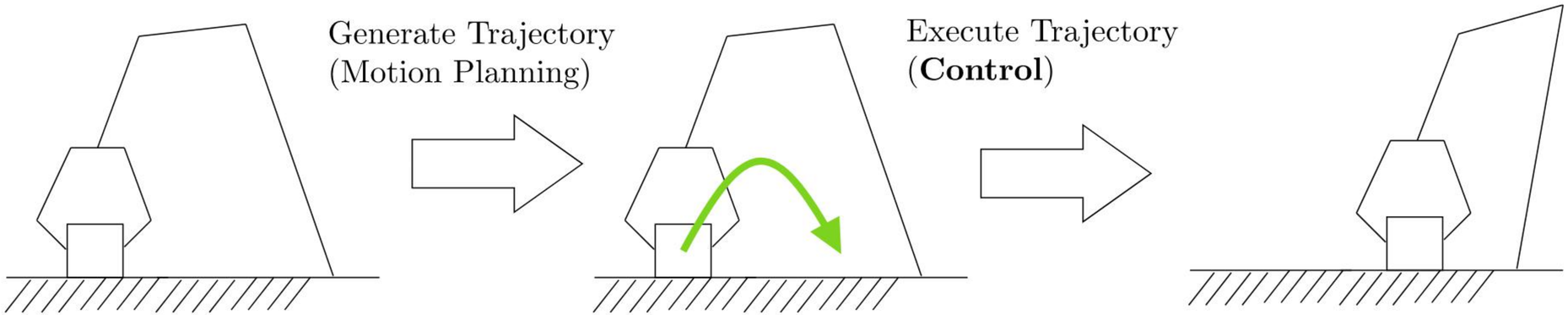
A schematic drawing of the inverted pendulum on a cart. The rod is considered massless.

https://en.wikipedia.org/wiki/Inverted_pendulum

Control

Plan versus Control

- Suppose we have a robot, how do we use it to move objects?



- Motion planning algorithms can generate a trajectory (position, velocity, and acceleration) of the robot.
- Now we need to know how to **control** the robot to follow such a trajectory.

Control

- Let us look at what we have
  - A *desired* trajectory to follow: $(q_d, \dot{q}_d, \ddot{q}_d)$
  - Forward dynamics $\ddot{q} = \mathrm{FD}(\boldsymbol{F}; \mathrm{q}, \dot{\mathrm{q}})$
  - Inverse dynamics $\boldsymbol{F} = \mathrm{ID}(\ddot{\mathrm{q}}; \mathrm{q}, \dot{\mathrm{q}})$
- Ideally, we just use inverse dynamics to compute $\ddot{q}$ at every moment to match $\ddot{q}_d$, and we are done!

Control

- Let us look at what we have
  - A *desired* trajectory to follow: $(q_d, \dot{q}_d, \ddot{q}_d)$
  - Forward dynamics $\ddot{q} = \mathrm{FD}(\boldsymbol{F}; \mathrm{q}, \dot{\mathrm{q}})$
  - Inverse dynamics $\boldsymbol{F} = \mathrm{ID}(\ddot{\mathrm{q}}; \mathrm{q}, \dot{\mathrm{q}})$
- Ideally, we just use inverse dynamics to compute $\ddot{q}$ at every moment to match $\ddot{q}_d$, and we are done!
- However, the real world is not perfect. There will be many sources of error, and error accumulates if only acceleration is matched.
- We use **control** to deal with delay, overshoot, or steady-state error, and ensure stability.

$$e = q - q_d \qquad \text{(steady-state error)}$$

PID Controller

Feedforward and Feedback Control

- We need some force to match $\ddot{q}_d$. This component is called the **feed-forward component**, which comes from $\mathrm{ID}(\cdot)$:

$$\boldsymbol{F}_{ff} = \mathrm{ID}(\ddot{\mathrm{q}}_\mathrm{d}; \mathrm{q}, \dot{\mathrm{q}})$$

- We also need some additional force to correct the steady-state error, which is called the **feedback component**:

$$\boldsymbol{F}_{fb} = M(q)(-K_v\dot{e} - K_p e)$$

  where $M(q)$ is the inertia of the system.

Computed Torque Control Law

- Let us apply the previous control law to our single-arm robot derived last lecture.
- Inverse dynamics equation:

$$\tau = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + g(\theta)$$

Computed Torque Control Law

- Let us apply the previous control law to our single-arm robot derived last lecture.
- Inverse dynamics equation:

$$\tau = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + g(\theta)$$

- Therefore,

$$\tau_{ff} = M(\theta)\ddot{\theta}_d + C(\theta, \dot{\theta})\dot{\theta} + g(\theta)$$
$$\tau_{fb} = M(\theta)(-K_v\dot{e} - K_p e)$$

where $K_v, K_p \in \mathbb{S}^+$ are constant matrices ($\mathbb{S}^+$: positive-semidefinite matrices cone).

Computed Torque Control Law

- Let us apply the previous control law to our single-arm robot derived last lecture.
- Inverse dynamics equation:

$$\tau = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + g(\theta)$$

- Therefore,

$$\tau_{ff} = M(\theta)\ddot{\theta}_d + C(\theta, \dot{\theta})\dot{\theta} + g(\theta)$$
$$\tau_{fb} = M(\theta)(-K_v\dot{e} - K_p e)$$

  where $K_v, K_p \in \mathbb{S}^+$ are constant matrices ($\mathbb{S}^+$: positive-semidefinite matrices cone).
- Combine them together, and the *computed torque control law* is:

$$\tau = M(\theta)(\ddot{\theta}_d - K_v\dot{e} - K_p e) + C(\theta, \dot{\theta})\dot{\theta} + g(\theta) \quad \text{(computed torque control)}$$

Convergence Analysis

- We will control the system by the torque

$$\tau = M(\theta)(\ddot{\theta}_d - K_v\dot{e} - K_p e) + C(\theta, \dot{\theta})\dot{\theta} + g(\theta) \tag{1}$$

- However, inverse dynamics equation tells us that the acceleration from τ is

$$\tau = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + g(\theta) \tag{2}$$

- Subtracting (2) from (1) and cancel $M(\theta)$, we get the error equation:

$$\ddot{e} + K_v\dot{e} + K_p e = 0$$

Convergence Analysis

$$\ddot{e} + K_v \dot{e} + K_p e = 0$$

- Because $K_v, K_p \in \mathbb{S}^+$, by the theory of ODE, $e(t) = \mathcal{O}(e^{\alpha t})$, $\alpha \leq 0$.
- We say that the computed torque control law has **exponential convergence rate**.

- Sometimes, we do not have the inverse dynamics equation of the system, but we still hope the controller to work

  (Isn't that the advertisement of model-free reinforcement learning?)

PD Control

- The PD control law has the form:

$$\tau = -K_v \dot{e} - K_p e \qquad \text{(PD control)}$$

  where $K_v, K_p \in \mathbb{S}^+$ and $e = \theta - \theta_d$.
- Does not compute inverse dynamics at all.
- No theoretical guarantee in general.
- Not practical in general
- e may converge; however, it usually does not converge to 0.

PID Control

- PID Control law has the form

$$\tau = -K_v \dot{e} - K_p e - K_i \int_0^t e(t) \mathrm{d}t \qquad \text{(PID control)}$$

  where $K_v, K_p, K_i \in \mathbb{S}^+$ and $e = \theta - \theta_d$.
- K_i term: accumulate errors over all past time steps.
- Inherits all the issues of PD, except
- When e converges, it usually converges to 0.
- Widely used in practice.

Augmented PID Control

- Augmented PID control law has the form:

$$\tau = \mathrm{ID}(\ddot{\theta}_\mathrm{d}) - \mathrm{K_v}\dot{\mathrm{e}} - \mathrm{K_p}\mathrm{e}$$

- Recall the computed torque control law:

$$\tau = \mathrm{ID}(\ddot{\theta}_\mathrm{d}) - \mathrm{M}(\theta)\mathrm{K_v}\dot{\mathrm{e}} - \mathrm{M}(\theta)\mathrm{K_p}\mathrm{e}$$

- Compared with the computed torque control law, K's may be harder to tune.
- Does result in exponential convergence for $K_v, K_p \in \mathbb{S}^+$

Tuning PID

Effects of *increasing* a parameter independently

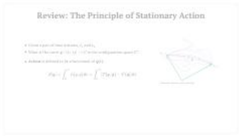
Parameter	Rise time	Overshoot	Settling time	Steady-state error	Stability
K_p	Decrease	Increase	Small change	Decrease	Degrade
K_i	Decrease	Increase	Increase	Eliminate	Degrade
K_d	Minor change	Decrease	Decrease	No effect in theory	Improve if K_d small

Kiam Heong Ang; Chong, G.; Yun Li (2005). "PID control system analysis, design, and technology". IEEE Transactions on Control Systems Technology. 13 (4): 559–576.
Jinghua Zhong (Spring 2006). "PID Controller Tuning: A Short Tutorial" (PDF). Archived from the original on 2015-04-21. Retrieved 2011-04-04.
https://en.wikipedia.org/wiki/PID_controller#cite_note-24

L10: Constraint Force Computation, Control
Han Su
Spring, 2021


Agenda
Constrained Lagrangian Method
Variational Method
Control
PID Control

Review: Lagrangian Function


Review: The Principle of Stationary Action

Review: The Principle of Stationary Action

Review: Euler-Lagrange Equation

Example: Robot Arm

Robot Arm