A mathematical Introduction to Robotic Manipulation

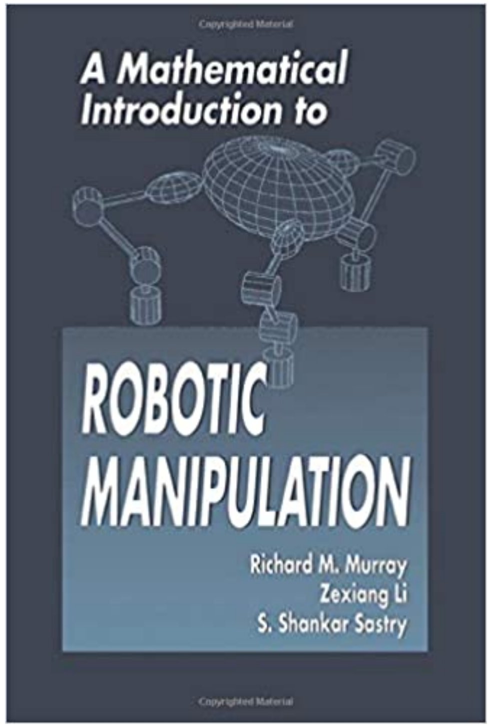
輪講第六章

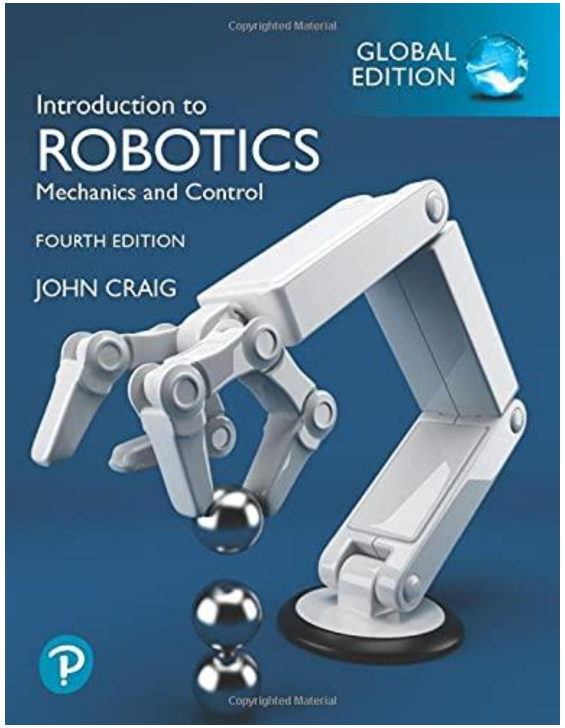
発表者: Zhang Xinyi (張 馨芸)

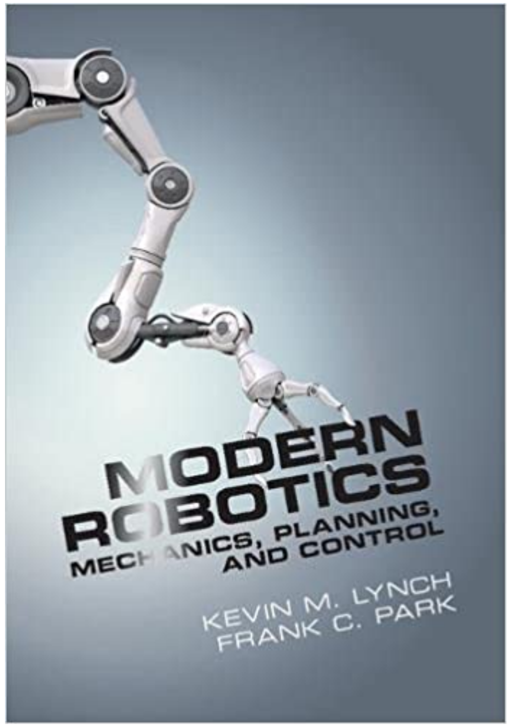
2021.10.8@マニピュレーション若手の会・勉強会

Some References

- Besides this book, I made this slides under the references of other two books:







A Mathematical Introduction to Robotic Manipulation

Introduction to Robotics Mechanics and Control

Modern Robotics

Chapter 6: Hand Dynamics and Control

Contents	Goal
1. Lagrange's Equations with Constraints	Calculate the dynamics of a mechanical system subject to Pfaffian constraints
2. Robot Hand Dynamics	Derive the equations of motion for a multifingered hand manipulating an object
3. Redundant and Nonmanipulable Robot Systems	Derive more complex equations of motion for redundant or nonmanipulable robot system
4. Kinematics and Statics of Tendon actuation	Describe the kinematics of tendon-driven systems
5. Control of Robot Hands	Introduce an extended control law for constraints-involved system and other control structures

Contents of This Talk

- Recall
- Lagrange’s Equations with Constraints
- Robot Hand Dynamics
- Redundant and Nonmanipulable Robot Systems
- Kinematics and Statics of Tendon Actuation
- Control of Robot Hand

Contents of This Talk

- **Recall**
  - Chapter 4 Robot Dynamics and Control
  - Chapter 5 Multifingered Hand Kinematics
- Lagrange's Equations with Constraints
- Robot Hand Dynamics
- Redundant and Nonmanipulable Robot Systems
- Kinematics and Statics of Tendon Actuation
- Control of Robot Hand

Recall

- We only need to recall Jacobian

The *manipulator Jacobian* relates the joint velocities $\dot{\theta}$ to the end-effector velocity V_{st} and the joint torques τ to the end-effector wrench F:

$$V^s_{st} = J^s_{st}(\theta)\dot{\theta} \qquad \tau = (J^s_{st})^T F_s \qquad \text{(spatial)}$$
$$V^b_{st} = J^b_{st}(\theta)\dot{\theta} \qquad \tau = (J^b_{st})^T F_t \qquad \text{(body)}.$$

If the manipulator kinematics is written using the product of exponentials formula, then the manipulator Jacobians have the form:

$$J^s_{st}(\theta) = \begin{bmatrix}\xi_1 & \xi'_2 & \cdots & \xi'_n\end{bmatrix} \qquad \xi'_i = \mathrm{Ad}_{\left(e^{\widehat{\xi}_1\theta_1} \cdots e^{\widehat{\xi}_{i-1}\theta_{i-1}}\right)} \xi_i$$
$$J^b_{st}(\theta) = \begin{bmatrix}\xi^\dagger_1 & \cdots & \xi^\dagger_{n-1} & \xi^\dagger_n\end{bmatrix} \qquad \xi^\dagger_i = \mathrm{Ad}^{-1}_{\left(e^{\widehat{\xi}_i\theta_i} \cdots e^{\widehat{\xi}_n\theta_n} g_{st}(0)\right)} \xi_i.$$

Recall

- Robotic dynamics: deriving the equation of motion including $q, \dot{q}, \ddot{q}$ and τ
- Forward dynamics: find joint accelerations
  - Given $q, \dot{q}$ and τ, find $\ddot{q}$
- Inverse dynamics: find joint forces and torques
  - Given $q, \dot{q}$ and $\ddot{q}$, find τ
- Two approaches for solving robot dynamics problem.

1. Lagrange's equations
   - Energy-based
   - Determine and exploit structural properties of the dynamics

2. Newton-Euler equations
   - Rely on $f = ma$
   - Often used for numerical solution of forward/inverse dynamics

- Lagrange's equation

1. The equations of motion for a mechanical system with Lagrangian $L = T(q, \dot{q}) - V(q)$ satisfies *Lagrange's equations*:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \Upsilon_i,$$

where $q \in \mathbb{R}^n$ is a set of generalized coordinates for the system and $\Upsilon \in \mathbb{R}^n$ represents the vector of generalized external forces.

- Newton-Euler equations

  - m: mass of the body, assume origin of {b} =CoM
  - F^b: total force and moment acting on the body
  - mv^b: linear momentum of the body
  - $\mathcal{I}\omega^b$: angular momentum of the body

2. The equations of motion for a rigid body with configuration $g(t) \in SE(3)$ are given by the *Newton-Euler equations*:

$$\begin{bmatrix} mI & 0 \\ 0 & \mathcal{I} \end{bmatrix} \begin{bmatrix} \dot{v}^b \\ \dot{\omega}^b \end{bmatrix} + \begin{bmatrix} \omega^b \times mv^b \\ \omega^b \times \mathcal{I}\omega^b \end{bmatrix} = F^b,$$

where m is the mass of the body, $\mathcal{I}$ is the inertia tensor, and $V^b = (v^b, \omega^b)$ and F^b represent the instantaneous body velocity and applied body wrench.

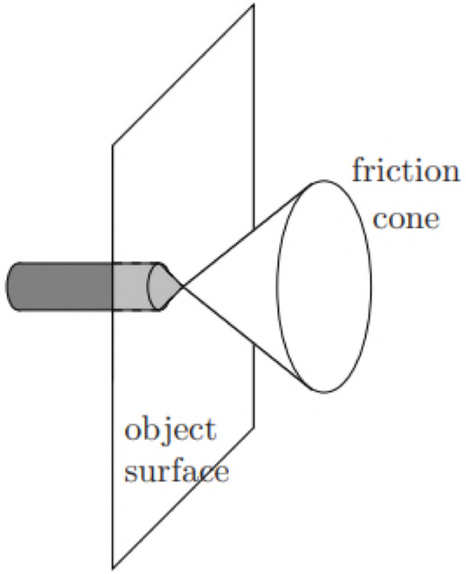
- Lagrange's equation for open-chain robot manipulator

慣性力＋遠心力・コリオリ力＋ポテンシャルエネルギーに伴う力
=関節に加えられるトルクとそれ以外の力

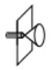
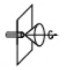
3. The equations of motion for an open-chain robot manipulator can be written as

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

Recall



Contact model

Contact type	Picture	Wrench basis	FC
Frictionless point contact		$\begin{bmatrix}0\\0\\1\\0\\0\\0\end{bmatrix}$	$f_1 \geq 0$
Point contact with friction		$\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\\0&0&0\\0&0&0\\0&0&0\end{bmatrix}$	$\sqrt{f_1^2+f_2^2} \leq \mu f_3$ $f_3 \geq 0$
Soft-finger		$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&0\\0&0&0&0\\0&0&0&1\end{bmatrix}$	$\sqrt{f_1^2+f_2^2} \leq \mu f_3$ $f_3 \geq 0$ $\lvert f_4\rvert \leq \gamma f_3$

Common contact types

$$F_o = G_1 f_{c_1} + \cdots + G_k f_{c_k} = \begin{bmatrix} G_1 & \cdots & G_k \end{bmatrix} \begin{bmatrix} f_{c_1} \\ \vdots \\ f_{c_k} \end{bmatrix}$$

Grasp map: map the contact forces to the total object force

Definition 5.2. Force-closure grasp
A grasp is a *force-closure* grasp if given any external wrench $F_e \in \mathbb{R}^p$ applied to the object, there exist contact forces $f_c \in FC$ such that

$$G f_c = -F_e.$$

Definition of force closure

Definition 5.3. Internal forces
If $f_N \in \mathcal{N}(G) \cap FC$, then f_N is an *internal force*. If $f_N \in \mathcal{N}(G)$ and $f_N \in \operatorname{int}(FC)$, then it is called a *strictly internal force*.

Definition of internal forces

$$J_h(\theta, x_o)\dot{\theta} = G^T(\theta, x_o)\dot{x}_o$$

Grasp constraints

$$\begin{tikzcd}
 & \text{finger} & \text{contact} & \text{object} \\
\text{velocity domain} & \dot{\theta} \arrow[r, "J_h"] & \dot{x}_c & V_{po}^b \arrow[l, "G^T"'] \\
\text{force domain} & \tau & f_c \arrow[l, "J_h^T"'] \arrow[r, "G"] & F_o
\end{tikzcd}$$

Relationship between forces and velocities

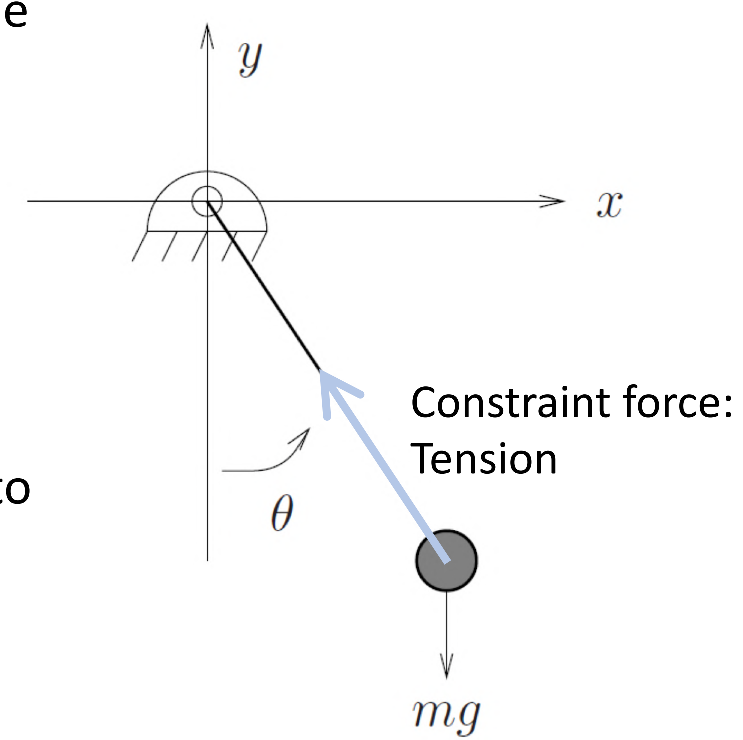
Contents of This Talk

Constraints

- A constraint restricts the motion of the mechanical system by limiting the set of paths which the system can follow.
- e.g. An idealized planar pendulum $q = (x, y) \in \mathbb{R}^2$
- All trajectories of the particles must satisfy the *algebraic constraint*:

$$x^2 + y^2 = l^2$$

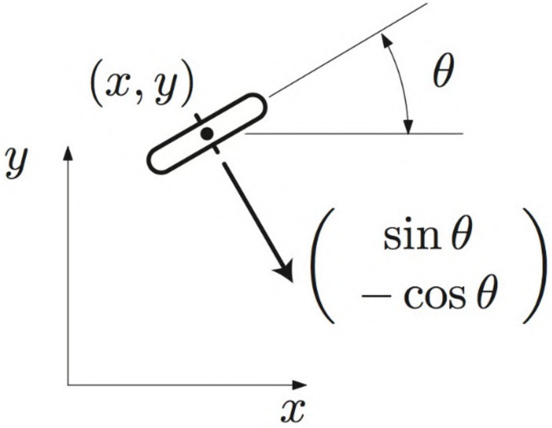
  - This constraint acts via *constraint forces*, which modify the motion to insure the constraint is always satisfied.
- Holonomic constraint vs. nonholonomic constraint

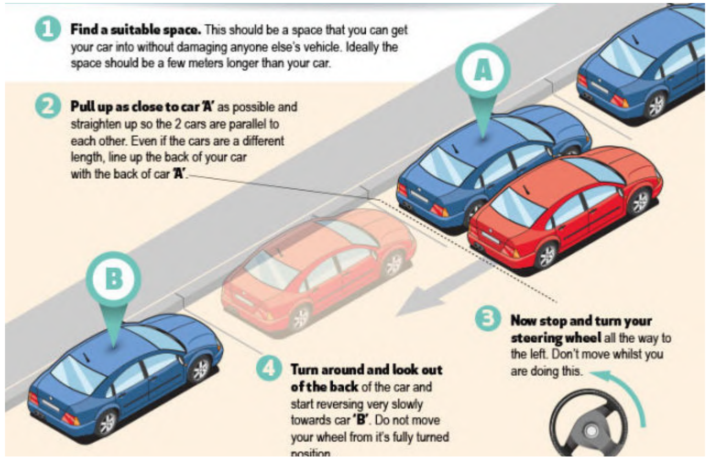


Constraints

- Holonomic constraint vs. nonholonomic constraint
- Let's explain simply using some mechanical system examples with constraints

Planar unicycle





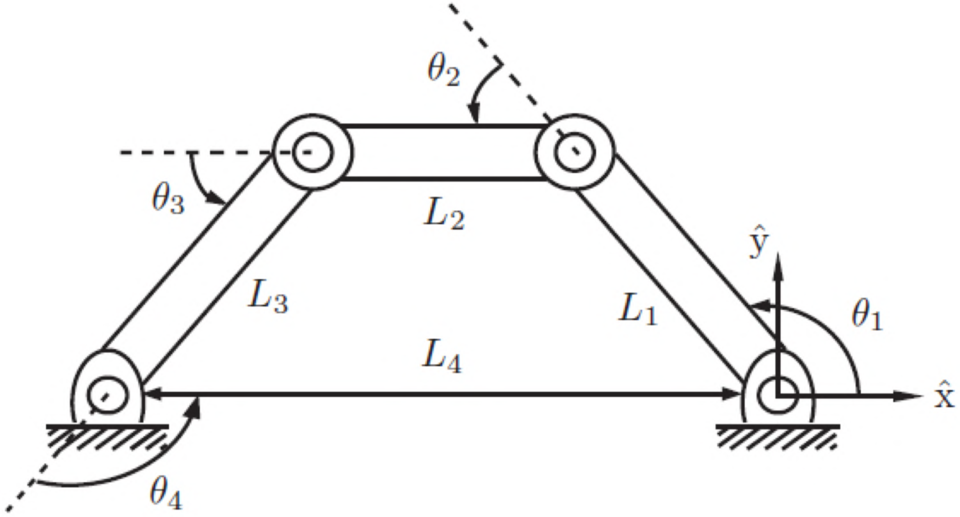
- Configuration space can be represented by vector:
  - $(x, y, \theta) \in \mathbb{R}^3$
- They always satisfy this equation:

$$\dot{y} - \dot{x} \cdot tan(\theta) = 0$$

- (Constraint involves velocity)
- It's a *nonholonomic* constraint this system could move between two arbitrary states with some constraint of velocity.

Constraints

- Holonomic constraint vs. nonholonomic constraint



Planar four-bar linkage

- Configuration space can be represented by vector:
  - $(\theta_1, \theta_2, \theta_3, \theta_4) \in \mathbb{R}^4$
- These four joints always satisfy these equations:

$$
\begin{aligned}
L_1 \cos\theta_1 + L_2 \cos(\theta_1 + \theta_2) + \cdots + L_4 \cos(\theta_1 + \cdots + \theta_4) &= 0, \\
L_1 \sin\theta_1 + L_2 \sin(\theta_1 + \theta_2) + \cdots + L_4 \sin(\theta_1 + \cdots + \theta_4) &= 0, \\
\theta_1 + \theta_2 + \theta_3 + \theta_4 - 2\pi &= 0.
\end{aligned}
$$

- Degree of Freedom: one
- It's a *holonomic* constraint because it reduces degrees of freedom in the system

Holonomic/Nonholonomic Constraint

- If we set
  - n: dimensions of configuration space $q = (q_1, \dots, q_n)$
  - k: number of independent constraints
  - A question: whether the system could be moved between two arbitrary states without violating the velocity constraint?
- *Holonomic constraints* can be represented locally as algebraic constraints:
  - $h(q) = 0, h : \mathbb{R}^n \rightarrow \mathbb{R}^k$
  - Answer: No
- *Nonholonomic constraints* can be represented as
  - $h(q, \dot{q}) = 0$
  - Answer: Yes

Holonomic constraint

- *Holonomic constraints* can be represented locally as algebraic constraints:
  - $h(q) = 0, h : \mathbb{R}^n \to \mathbb{R}^k$
- And the matrix $\dfrac{\partial h}{\partial q} = \begin{bmatrix} \frac{\partial h_1}{\partial q_1} & \cdots & \frac{\partial h_1}{\partial q_n} \\ & \ddots & \\ \frac{\partial h_k}{\partial q_1} & \cdots & \frac{\partial h_k}{\partial q_n} \end{bmatrix}$ is full row rank
- Constraint force $\Gamma = \dfrac{\partial h}{\partial q}^T \lambda,$
  - Constraint forces do no work (will be explained later)

Pfaffian constraint

- *Pfaffian constraint*: generally we write velocity constraints as:

$$A(q)\dot{q} = 0,$$ where $A(q) \in \mathbb{R}^{k \times n}$ represents a set of k velocity constraints.

- However, if there exist a vector-valued function $h : Q \to \mathbb{R}^k$ such that
  - $A(q)\dot{q} = 0 \quad \Longleftrightarrow \quad \frac{\partial h}{\partial q}\dot{q} = 0.$
  - Pfaffian constraint is integrable
  - Pfaffian constraint is equivalent to a holonomic constraint
- Otherwise, pfaffian constraint which is not integrable is an example of a non-holonomic constraint (not all).
- Constraint forces $\Gamma = A^T(q)\lambda,$

Dynamics with Constraints

- Goal: derive the equations of motion for a mechanical system with configuration $q \in \mathbb{R}^n$ subject to a set of *Pfaffian constraints*.
  - Mechanical system: constraints are everywhere smooth and linearly
  - Lagrangian: $L(q, \dot{q})$ kinetic energy minus potential energy
  - Constraint: $A(q)\dot{q} = 0 \qquad A(q) \in \mathbb{R}^{k \times n}$.
- Let's write the equations of motion considering the constraint can affects the motion additionally:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} + \underbrace{A^T(q)\lambda}_{\text{Constraint forces}} - \underbrace{\Upsilon}_{\text{Nonconservative and externally applied forces}} = 0,$$

  - $\lambda_i, \ldots, \lambda_k$: relative magnitudes of constraint forces, also called *Lagrange multipliers*

Dynamics with Constraints

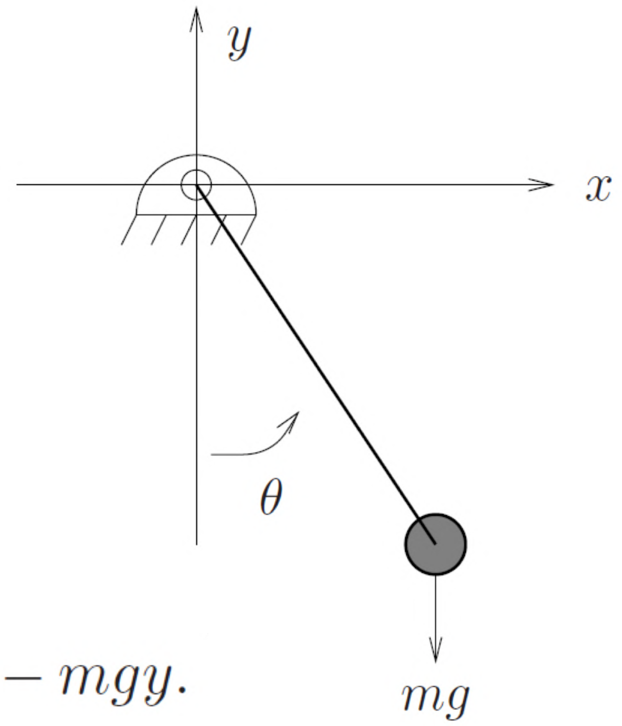
- 3 Steps for calculating the equation of motion with constraints
  ① Write the equations of motion (done, but Lagrange multipliers are unknown)
  ② **Solve these multipliers** because each λ_i will be a function with $q, \dot{q}, \Upsilon$
  ③ Substituting them back into the equations of motion
- We will show how to solve the multipliers λ in ②:
  - Differentiate the constraint equation $A(q)\dot{q} = 0$ (6.3) ⇒ $A(q)\ddot{q} + \dot{A}(q)\dot{q} = 0$ (6.3.1)
  - Write Lagrange's equations like this $M(q)\ddot{q} + C(q, \dot{q})\dot{q} + N(q, \dot{q}) + A^T(q)\lambda = F,$ (6.5)
  - Solve (6.5) for $\ddot{q}$ and substitute into (6.3.1), and we will get

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

If constraints are independent, this matrix is full rank

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

Dynamics with Constraints



① **Write the equations of motion**

② Solve these multipliers

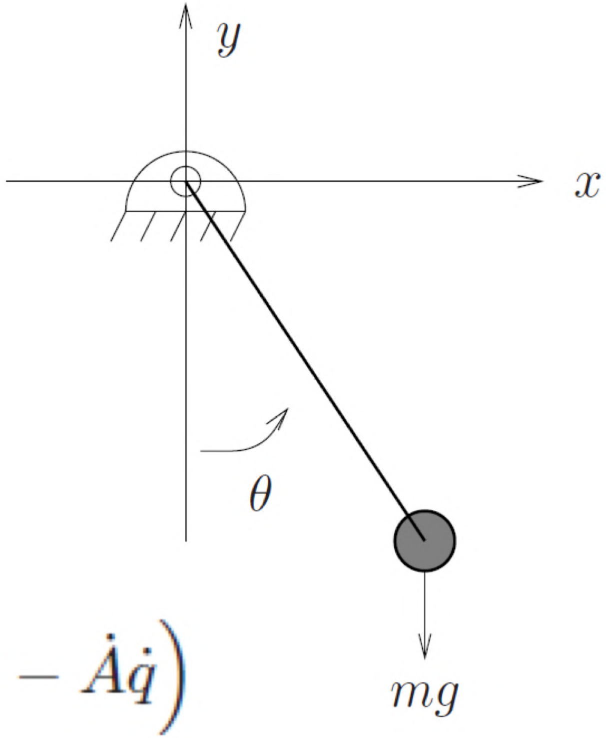
③ Substituting them back into the equations of motion

- Configuration $q = (x, y) \in \mathbb{R}^2$
- Constraint $x^2 + y^2 = l^2$
- Pfaffian constraint $\underbrace{\begin{bmatrix} x & y \end{bmatrix}}_{A(q)} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = 0$
- No constraint Lagrangian $L(q, \dot{q}) = \frac{1}{2}m(\dot{x}^2 + \dot{y}^2) - mgy.$
- Substitude these formulation into $\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} + A^T(q)\lambda = 0,$
- So Lagrangian with constraint will be:

$$\underbrace{\begin{bmatrix} m & 0 \\ 0 & m \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} + \begin{bmatrix} 0 \\ mg \end{bmatrix}}_{\textit{Forces that move the pendulum}} + \underbrace{\begin{bmatrix} x \\ y \end{bmatrix}}_{\textit{Forces against constraints}} \lambda = 0.$$

Unknown, let's move to step ②

Dynamics with Constraints



① Write the equations of motion

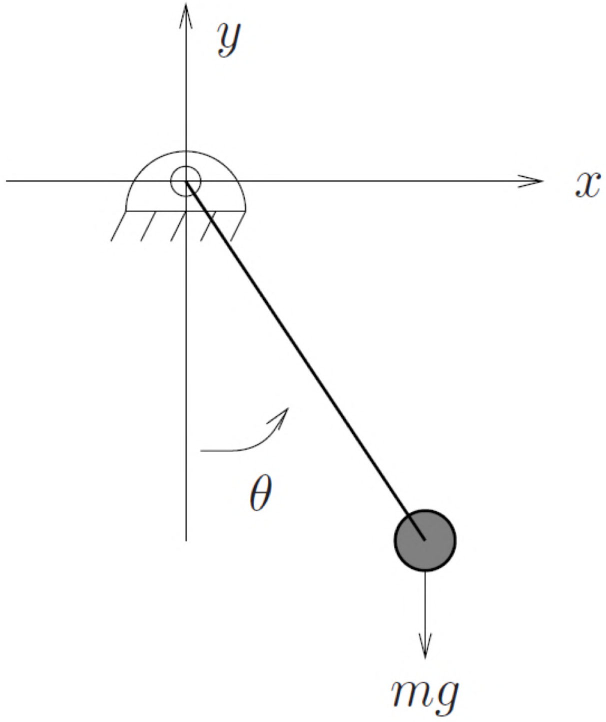
② **Solve these multipliers**

③ Substituting them back into the equations of motion

- Solve Lagrange Multipliers using this:

$$\lambda = (AM^{-1}A^T)^{-1}\left(AM^{-1}(Q - C\dot{q} - N) - \dot{A}\dot{q}\right)$$
$$= \frac{m}{x^2+y^2}(-gy - \dot{x}^2 - \dot{y}^2) = -\frac{m}{l^2}(gy + \dot{x}^2 + \dot{y}^2),$$

Dynamics with Constraints



① Write the equations of motion

② Solve these multipliers

③ **Substituting them back into the equations of motion**

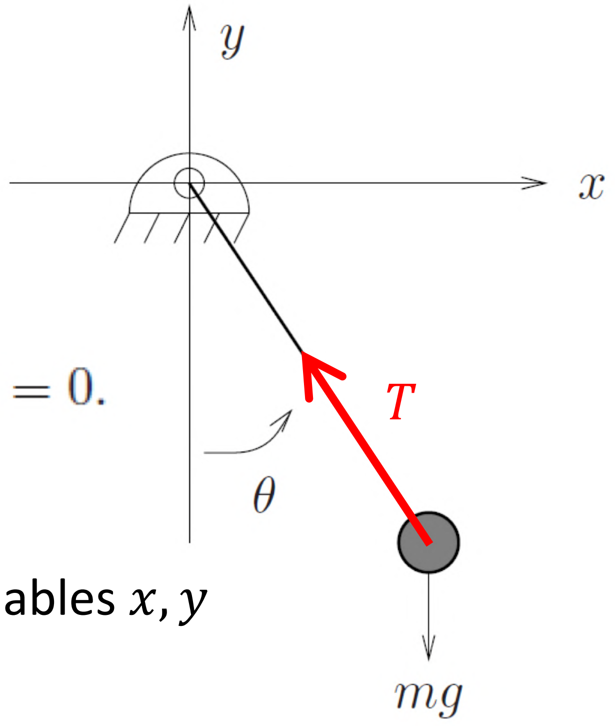
$$\begin{bmatrix} m & 0 \\ 0 & m \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} + \begin{bmatrix} 0 \\ mg \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix} \lambda = 0.$$

$$-\frac{m}{l^2}(gy + \dot{x}^2 + \dot{y}^2)$$

- Finally the equations of motion are:

$$\begin{bmatrix} m & 0 \\ 0 & m \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} + \begin{bmatrix} 0 \\ mg \end{bmatrix} - \frac{1}{l^2} \begin{bmatrix} x \\ y \end{bmatrix} (mgy + m(\dot{x}^2 + \dot{y}^2)) = 0.$$

Dynamics with Constraints



$$\begin{bmatrix} m & 0 \\ 0 & m \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} + \begin{bmatrix} 0 \\ mg \end{bmatrix} - \frac{1}{l^2} \begin{bmatrix} x \\ y \end{bmatrix} (mgy + m(\dot{x}^2 + \dot{y}^2)) = 0.$$

① Write the equations of motion

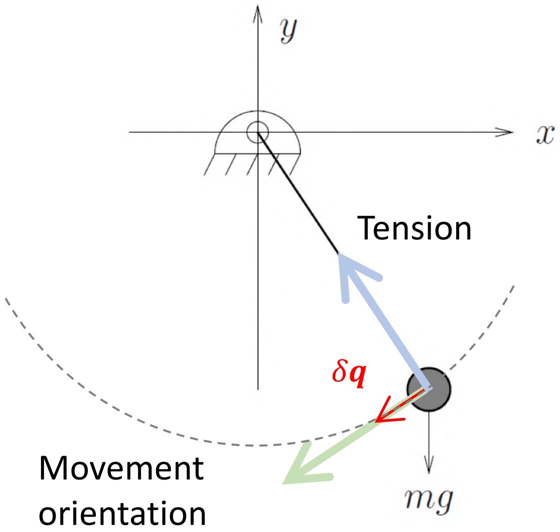
② Solve these multipliers

③ **Substituting them back into the equations of motion**

- This is a second-order differential equation in two variables x, y
- But system only has one degree of freedom
- Thus, we have increased the number of variables required to represent the motion of the system.
- Additionally, we can obtain constraint force: tension T in the rod:

$$\text{Tension} = \left\| \begin{bmatrix} x \\ y \end{bmatrix} \lambda \right\| = \frac{mg}{l} y + \frac{m}{l} (\dot{x}^2 + \dot{y}^2).$$

Lagrange-D'Alembert Equation



This example can show that constraint forces do no work

- D'Alembert's principle: constraint forces do no work for any instantaneous motion which satisfies the constraints.
  - Given configuration $\boldsymbol{q} \in \mathbb{R}^n$,
  - Virtual displacement $\delta \boldsymbol{q} \in \mathbb{R}^n$, an arbitrary infinitesimal displacement which satisfies the constraints $A(q)\delta q = 0.$

$$(A^T(q)\lambda) \cdot \delta q = 0$$

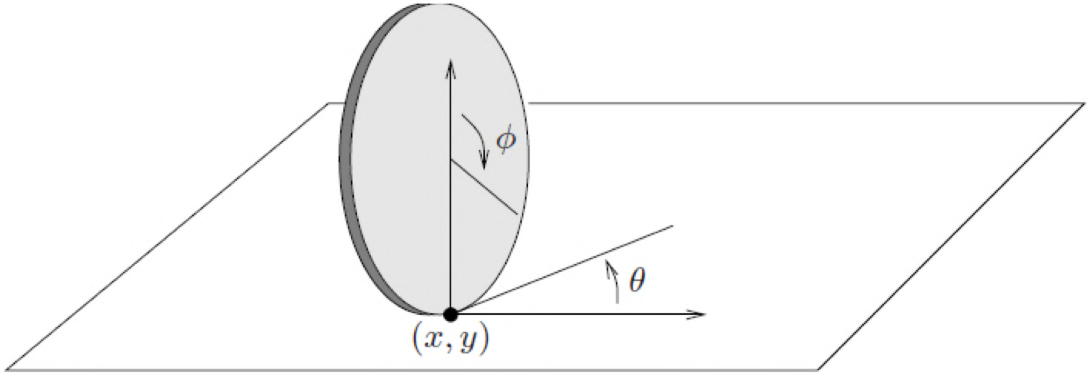
- The reason why we introduce D'Alembert's principle:
  - Project motion onto the feasible direction, ignore constraint forces
  - Obtain a more concise equation of the dynamics

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} + \underbrace{A^T(q)\lambda}_{\text{Constraint forces}} - \underbrace{\Upsilon}_{\text{Nonconservative and externally applied forces}} = 0, \quad \Rightarrow \quad \left(\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} - \Upsilon\right) \cdot \delta q = 0,$$

Lagrange equation can become this when Eliminating constraint force

Lagrange-D'Alembert Equation

- Let's use Lagrange-d'Alembert equation to solve the dynamics for a rolling disk



A rolling disk that rolls without slipping

- Configuration $q = (x, y, \theta, \phi)$
- Velocity constraints

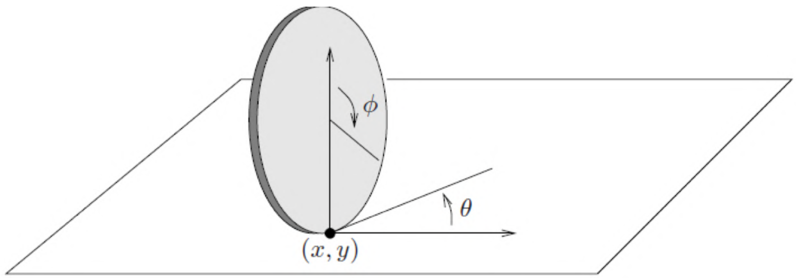
$$\begin{aligned} \dot{x} - \rho\cos\theta\dot{\phi} &= 0 \\ \dot{y} - \rho\sin\theta\dot{\phi} &= 0 \end{aligned} \quad \text{or} \quad A(q)\dot{q} = \begin{bmatrix} 1 & 0 & 0 & -\rho\cos\theta \\ 0 & 1 & 0 & -\rho\sin\theta \end{bmatrix}\dot{q} = 0.$$

- τ_θ: driving torque on the wheel
- τ_ϕ: steering torque (about the vertical axis)
- $\mathcal{I}_\infty$:inertia about the horizontal (rolling) axis
- $\mathcal{I}_\in$:inertia about the vertical axis

- Lagrangian will be:

$$L(q,\dot{q}) = \frac{1}{2}m(\dot{x}^2 + \dot{y}^2) + \frac{1}{2}\mathcal{I}_\infty\dot{\theta}^\in + \frac{\infty}{\in}\mathcal{I}_\in\dot{\phi}^\in. \iff L(q,\dot{q}) = \frac{1}{2}(\dot{x}^2+\dot{y}^2)+\frac{1}{2}\mathcal{I}_\infty\dot{\theta}^2+\frac{1}{2}\mathcal{I}_\in\dot{\phi}^2$$

Lagrange-D'Alembert Equation



① **Write the equations of motion**

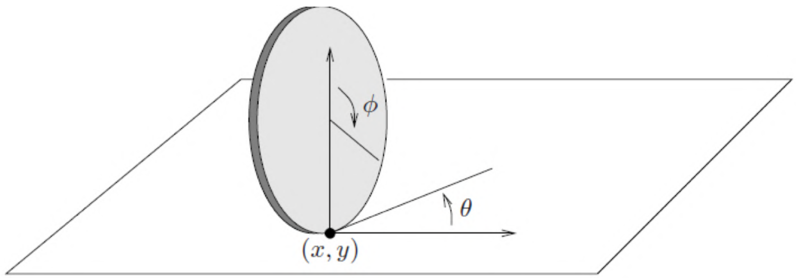
② Reduce the configuration

③ Further simplify the equation

- Virtual displacement $\delta q = (\delta x, \delta y, \delta \theta, \delta \phi)$
- Lagrange-d'Alembert equations

$$\left(\begin{bmatrix} m & & & 0 \\ & m & & \\ & & \mathcal{I}_\infty & \\ 0 & & & \mathcal{I}_\in \end{bmatrix} \ddot{q} - \begin{bmatrix} 0 \\ 0 \\ \tau_\theta \\ \tau_\phi \end{bmatrix} \right) \cdot \delta q = 0 \quad \text{where} \quad \begin{bmatrix} 1 & 0 & 0 & -\rho\cos\theta \\ 0 & 1 & 0 & -\rho\sin\theta \end{bmatrix} \delta q = 0.$$

Lagrange-D'Alembert Equation



① Write the equations of motion

② **Reduce the configuration**

③ Further simplify the equation

- Virtual displacement $\delta q = (\delta x, \delta y, \delta\theta, \delta\phi)$
- Lagrange-d'Alembert equations

$$\left(\begin{bmatrix} m & & & 0 \\ & m & & \\ & & \mathcal{I}_\infty & \\ 0 & & & \mathcal{I}_\in \end{bmatrix} \ddot{q} - \begin{bmatrix} 0 \\ 0 \\ \tau_\theta \\ \tau_\phi \end{bmatrix}\right) \cdot \delta q = 0 \quad \text{where} \quad \begin{bmatrix} 1 & 0 & 0 & -\rho\cos\theta \\ 0 & 1 & 0 & -\rho\sin\theta \end{bmatrix} \delta q = 0.$$

$\Downarrow$From constraint we can solve

$$\delta x = \rho\cos\theta\delta\phi$$
$$\delta y = \rho\sin\theta\delta\phi.$$

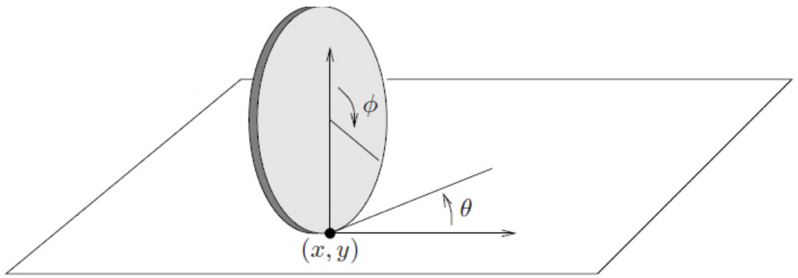
- Equation can be written without $\delta x, \delta y$

$$\left(\begin{bmatrix} 0 & 0 \\ m\rho\cos\theta & m\rho\sin\theta \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} + \begin{bmatrix} \mathcal{I}_\infty & 0 \\ 0 & \mathcal{I}_\in \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\phi} \end{bmatrix} - \begin{bmatrix} \tau_\theta \\ \tau_\phi \end{bmatrix}\right) \cdot \begin{bmatrix} \delta\theta \\ \delta\phi \end{bmatrix} = 0,$$

- Since $\delta\theta, \delta\phi$ are free, the dynamics become:

$$\begin{bmatrix} 0 & 0 \\ m\rho\cos\theta & m\rho\sin\theta \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} + \begin{bmatrix} \mathcal{I}_\infty & 0 \\ 0 & \mathcal{I}_\in \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} \tau_\theta \\ \tau_\phi \end{bmatrix}$$

Lagrange-D'Alembert Equation



① Write the equations of motion

② Reduce the configuration

③ **Further simplify the equation**

- We have dynamics equation:

$$\begin{bmatrix} 0 & 0 \\ m\rho\cos\theta & m\rho\sin\theta \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} + \begin{bmatrix} \mathcal{I}_\infty & 0 \\ 0 & \mathcal{I}_\in \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} \tau_\theta \\ \tau_\phi \end{bmatrix}$$

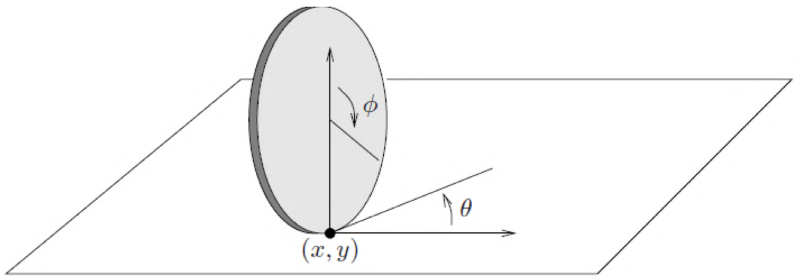
- We can eliminate $\dot{x},\ \dot{y}$ and $\ddot{x},\ \ddot{y}$ by differentiating the constraints

$$\begin{aligned} \dot{x} - \rho\cos\theta\dot{\phi} &= 0 \\ \dot{y} - \rho\sin\theta\dot{\phi} &= 0 \end{aligned} \quad \Rightarrow \quad \begin{aligned} \ddot{x} &= \rho\cos\theta\ddot{\phi} - \rho\sin\theta\dot{\theta}\dot{\phi} \\ \ddot{y} &= \rho\sin\theta\ddot{\phi} + \rho\cos\theta\dot{\theta}\dot{\phi}, \end{aligned}$$

- Finally, it's second-order differential equation in θ and ϕ

$$\begin{bmatrix} \mathcal{I}_\infty & 0 \\ 0 & \mathcal{I}_\in + \underbrace{\Updownarrow\rho^\in}_{\leftrightarrow m\rho^2} \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} \tau_\theta \\ \tau_\phi \end{bmatrix},$$

Lagrange-D'Alembert Equation



① Write the equations of motion

② Reduce the configuration

③ **Further simplify the equation**

- Let's summarize this rolling disk dynamics (a nonholonomic system).
- Given the trajectory of θ and ϕ, we can determine the trajectory of the disk as it rolls along the plane.
- The equation of motion is 1 + 2
  1. A second-order equations in a reduced set of variables plus

$$\begin{bmatrix} \mathcal{I}_\infty & 0 \\ 0 & \mathcal{I}_\in + \Updownarrow \rho^\in \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} \tau_\theta \\ \tau_\phi \end{bmatrix},$$

  2. A set of first-order equations

$$\dot{x} = \rho \cos\theta \dot{\phi}$$
$$\dot{y} = \rho \sin\theta \dot{\phi}.$$

Lagrange-D’Alembert Equation

- Let’s wrap it up with mathematical formulations
- Goal: get a more explicit description of the dynamics
  - Lagrange-d’Alembert equation $\left(\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} - \Upsilon\right) \cdot \delta q = 0, \text{ where } \delta q \in \mathbb{R}^n \text{ satisfies } A(q)\delta q = 0.$
  - Rewrite these:

$$A(q) = \begin{bmatrix} A_1(q) & A_2(q) \end{bmatrix}, \qquad q = (q_1, q_2) \in \mathbb{R}^{n-k} \times \mathbb{R}^k$$

  - So that we can use ∂q_1 to eliminate ∂q_2 . (∂q_1 is free or unconstrained)

$$A(q) \cdot \delta q = 0 \qquad \Longleftrightarrow \qquad \delta q_2 = -A_2^{-1}(q) A_1(q) \delta q_1,$$

$$\begin{aligned}
&\left(\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} - \Upsilon\right) \cdot \delta q \\
&= \left(\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_1} - \frac{\partial L}{\partial q_1} - \Upsilon_1\right) \cdot \delta q_1 + \left(\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_2} - \frac{\partial L}{\partial q_2} - \Upsilon_2\right) \cdot \delta q_2 \\
&= \left(\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_1} - \frac{\partial L}{\partial q_1} - \Upsilon_1\right) \cdot \delta q_1 + \left(\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_2} - \frac{\partial L}{\partial q_2} - \Upsilon_2\right) \cdot \left(-A_2^{-1} A_1\right) \delta q_1,
\end{aligned}$$

$$\Rightarrow \left(\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_1} - \frac{\partial L}{\partial q_1} - \Upsilon_1\right) - A_1^T A_2^{-T} \left(\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_2} - \frac{\partial L}{\partial q_2} - \Upsilon_2\right) = 0.$$

  - We can eliminate $\dot{q}_2, \ddot{q}_2$ using the constraint $\dot{q}_2 = -A_2^{-1} A_1 \dot{q}_1$

Nonholonomic System

- When we calculate the dynamics for a mechanical system with *nonholonomic* system
- Wrong: directly substitute the constraints to the equations of motion to eliminate the constraints
- For example:
  - Configuration $q = (r, s) \in \mathbb{R}^2 \times \mathbb{R}$
  - Constraints $\dot{s} + a^T(r)\dot{r} = 0 \qquad a(r) \in \mathbb{R}^2$, (nonholonomic)
  - Lagrangian $L_c(r, \dot{r}) = L(r, \dot{r}, -a^T(r)\dot{r})$. (for simplicity, assume it doesn't depend on s)
  - Substitute Lagrangian to the Lagrange-d'Alembert equation

$$\frac{d}{dt}\frac{\partial L_c}{\partial \dot{r}_i} - \frac{\partial L_c}{\partial r_i} = 0 \qquad i = 1, 2. \quad \Rightarrow \quad \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{r}_i} - a_i(r)\frac{\partial L}{\partial \dot{s}}\right) - \left(\frac{\partial L}{\partial r_i} - \frac{\partial L}{\partial \dot{s}}\sum_j \frac{\partial a_j}{\partial r_i}\dot{r}_j\right) = 0$$

  - Rearranging terms and we obtain:

$$\left(\frac{d}{dt}\frac{\partial L}{\partial \dot{r}_i} - \frac{\partial L}{\partial r_i}\right) - a_i(r)\left(\frac{d}{dt}\frac{\partial L}{\partial \dot{s}} - \frac{\partial L}{\partial s}\right) = \frac{\partial L}{\partial \dot{s}}\left(\dot{a}_i(r) - \sum_j \frac{\partial a_j}{\partial r_i}\dot{r}_i\right).$$

Nonholonomic System

- When we calculate the dynamics for a mechanical system with *nonholonomic* system
- Wrong: directly substitute the constraints to the equations of motion to eliminate the constraints
- For example:
  - Let’s look at the final equations

$$\underbrace{\left(\frac{d}{dt}\frac{\partial L}{\partial \dot{r}_i} - \frac{\partial L}{\partial r_i}\right) - a_i(r)\left(\frac{d}{dt}\frac{\partial L}{\partial \dot{s}} - \frac{\partial L}{\partial s}\right)}_{\text{Exactly Lagrange-d'Alembert equation}} = \underbrace{\frac{\partial L}{\partial \dot{s}}\left(\dot{a}_i(r) - \sum_j \frac{\partial a_j}{\partial r_i}\dot{r}_i\right)}_{\text{Spurious terms}}.$$

  - If we directly substitute the constraints to the equations of motion, we will get these *spurious terms*, the final dynamic equations are wrong

Holonomic System

- When we calculate the dynamics for a mechanical system with *nonholonomic* system
- Wrong: directly substitute the constraints to the equations of motion to eliminate the constraints
- Is it still wrong for a *holonomic* system?
  - We know the constraint is integrable, so that there exists $h(r)$ such that

$$\dot{s} + a^T(r)\dot{r} = 0 \qquad a(r) \in \mathbb{R}^2, \qquad \Rightarrow \quad a_i(r) = \frac{\partial h}{\partial r_i}.$$

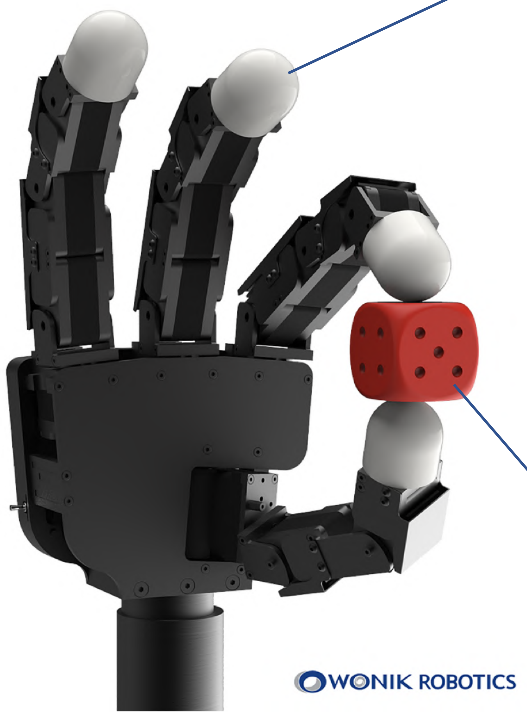
  - So that for the right side $\left(\frac{d}{dt}\frac{\partial L}{\partial \dot{r}_i} - \frac{\partial L}{\partial r_i}\right) - a_i(r)\left(\frac{d}{dt}\frac{\partial L}{\partial \dot{s}} - \frac{\partial L}{\partial s}\right) = \boxed{\frac{\partial L}{\partial \dot{s}}\left(\dot{a}_i(r) - \sum_j \frac{\partial a_j}{\partial r_i}\dot{r}_i\right).}$

$$\boxed{\frac{\partial L}{\partial \dot{s}}(\dot{a}_i(r) - \sum_j \frac{\partial a_j}{\partial r_i}\dot{r}_i)} = \frac{\partial L}{\partial \dot{s}}\left(\sum \frac{\partial^2 h}{\partial r_i \partial r_j}\dot{r}_j - \sum \frac{\partial^2 h}{\partial r_j \partial r_i}\dot{r}_i\right), \quad = 0$$

  - So for a *holonomic* system, if we substitute the constraints to the equations of motion, we can still get a correct equations of motion

Contents of This Talk

Equation of Motion



Dynamics of the fingers (using Lagrangian)

$$M_f(\theta)\ddot{\theta} + C_f(\theta,\dot{\theta})\dot{\theta} + N_f(\theta,\dot{\theta}) = \tau,$$

Joint angles for all fingers: $\theta = (\theta_{f_1}, \ldots, \theta_{f_k}) \in \mathbb{R}^n$

Joint torques for all fingers: $\tau \in \mathbb{R}^n$

$$M_f = \begin{bmatrix} M_{f_1} & & 0 \\ & \ddots & \\ 0 & & M_{f_k} \end{bmatrix} \quad C_f = \begin{bmatrix} C_{f_1} & & 0 \\ & \ddots & \\ 0 & & C_{f_k} \end{bmatrix} \quad N_f = \begin{bmatrix} N_{f_1} \\ \vdots \\ N_{f_k} \end{bmatrix}.$$

Dynamics of the object (Newton-Euler equation)

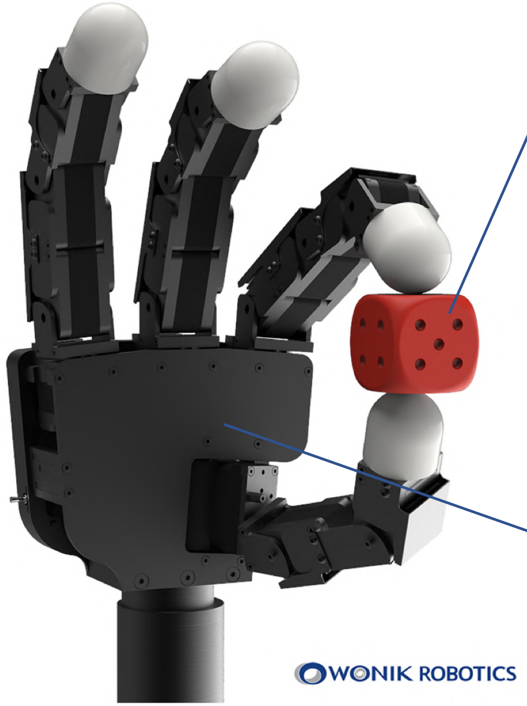
$$\begin{bmatrix} mI & 0 \\ 0 & \mathcal{I} \end{bmatrix} \begin{bmatrix} \dot{v}^b \\ \dot{\omega}^b \end{bmatrix} + \begin{bmatrix} \omega^b \times mv^b \\ \omega^b \times \mathcal{I}\omega^b \end{bmatrix} = F^b,$$

In Newton-Euler method: object $x_o = (p, R) \in SE(3)$

If object is subject to gravity alone:

$$\begin{bmatrix} mI & 0 \\ 0 & \mathcal{I} \end{bmatrix} \dot{V}^b + \begin{bmatrix} m\widehat{\omega}^b & 0 \\ 0 & \frac{1}{2}(\widehat{\omega}^b\mathcal{I} - \mathcal{I}\widehat{\omega}^b) \end{bmatrix} V^b + \begin{bmatrix} R^T(m\vec{g}) \\ 0 \end{bmatrix} = 0,$$

Equation of Motion



Dynamics of the **object**
(To apply Lagrangian-d'Alembert equation)

We have to convert object from $SE(3)$ to local coordinate, which is:
$x_o = (p, R) \in SE(3) \Rightarrow x \in \mathbb{R}^6$
So that the object dynamics can be written as:

$$M_o(x)\ddot{x} + C_o(x, \dot{x})\dot{x} + N_o(x, \dot{x}) = 0,$$

Grasp constraints

** Chapter 5*
5.5 Grasp Constraints

$$J_h(\theta, x)\dot{\theta} = G^T(\theta, x)\dot{x},$$

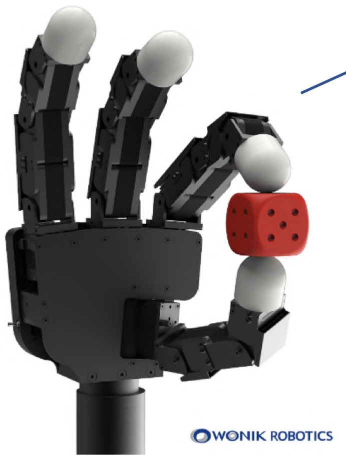
It's the relationship between the finger velocity and object velocity

Three assumptions of grasping
1) The grasp is force-closure and manipulable
2) The hand Jacobian is invertible
3) The contact forces remain in the friction cone at all times

Equation of Motion

Recall Lagrangian-d'Alembert equation

$$\left(\frac{d}{dt}\frac{\partial L}{\partial \dot q} - \frac{\partial L}{\partial q} - \Upsilon\right)\cdot \delta q = 0,$$
$$A(q)\delta q = 0.$$



Dynamics of the **system**
(Apply Lagrangian-d'Alembert equation)

- Configuration: $q = (\theta, x)$
- Lagrangian: $L = \frac{1}{2}\dot\theta^T M_f \dot\theta + \frac{1}{2}\dot x^T M_o \dot x - V_f(\theta) - V_o(x),$
- Virtual displacement: $\delta q = (\delta\theta, \delta x)$
- Constraints: $J_h(\theta, x)\dot\theta = G^T(\theta, x)\dot x,\quad \begin{bmatrix}-J_h & G^T\end{bmatrix}\begin{bmatrix}\dot\theta \\ \dot x\end{bmatrix} = 0$
- Lagrange-d'Alembert equations:

$$\begin{bmatrix}-J_h & G^T\end{bmatrix}\begin{bmatrix}\delta\theta \\ \delta x\end{bmatrix} = 0$$
$$\delta\theta = J_h^{-1} G^T \delta x$$

$$\left(\frac{d}{dt}\frac{\partial L}{\partial \dot q} - \frac{\partial L}{\partial q} - \begin{bmatrix}\tau \\ 0\end{bmatrix}\right)\cdot \delta q = \begin{bmatrix}\frac{d}{dt}\frac{\partial L}{\partial \dot\theta} - \frac{\partial L}{\partial \theta} - \tau \\ \frac{d}{dt}\frac{\partial L}{\partial \dot x} - \frac{\partial L}{\partial x}\end{bmatrix}\cdot\begin{bmatrix}\delta\theta \\ \delta x\end{bmatrix}$$
$$= \left(\frac{d}{dt}\frac{\partial L}{\partial \dot\theta} - \frac{\partial L}{\partial \theta} - \tau\right)\cdot \underline{(J_h^{-1}G^T\delta x)} + \left(\frac{d}{dt}\frac{\partial L}{\partial \dot x} - \frac{\partial L}{\partial x}\right)\cdot \delta x$$
$$= GJ_h^{-T}\left(\frac{d}{dt}\frac{\partial L}{\partial \dot\theta} - \frac{\partial L}{\partial \theta} - \tau\right)\cdot \delta x + \left(\frac{d}{dt}\frac{\partial L}{\partial \dot x} - \frac{\partial L}{\partial x}\right)\cdot \delta x,$$

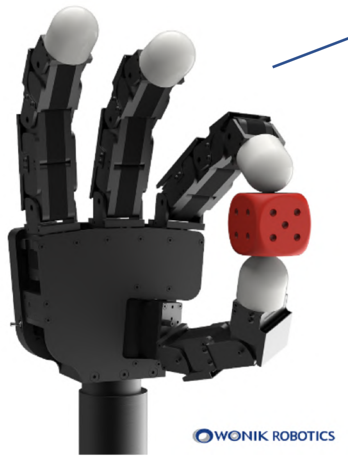
Apply steps from last section to using Lagrangian-d'Alembert equation

① **Write the equations of motion**

② Reduce the configuration

③ Further simplify the equation

Equation of Motion



Dynamics of the **system**
(Apply Lagrangian-d'Alembert equation)

- Configuration: $q = (\theta, x)$
- Lagrangian: $$L = \frac{1}{2}\dot{\theta}^T M_f \dot{\theta} + \frac{1}{2}\dot{x}^T M_o \dot{x} - V_f(\theta) - V_o(x),$$
- Virtual displacement: $\delta q = (\delta\theta, \delta x)$
- Lagrange-d'Alembert equations:

$$\left(\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} - \begin{bmatrix}\tau \\ 0\end{bmatrix}\right) \cdot \delta q = GJ_h^{-T}\left(\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta} - \tau\right) \cdot \delta x + \left(\frac{d}{dt}\frac{\partial L}{\partial \dot{x}} - \frac{\partial L}{\partial x}\right) \cdot \delta x, \quad = 0$$

- Since δx is free:

$$\left(\frac{d}{dt}\frac{\partial L}{\partial \dot{x}} - \frac{\partial L}{\partial x}\right) + GJ_h^{-T}\left(\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta}\right) = GJ_h^{-T}\tau.$$

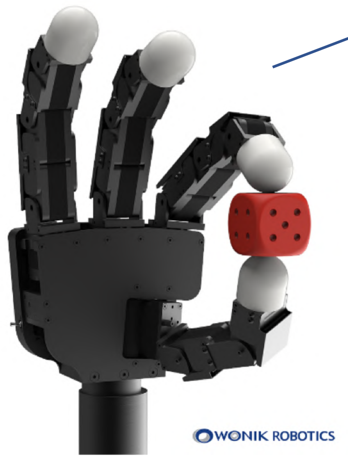
Apply steps from last section to using Lagrangian-d'Alembert equation

① Write the equations of motion

② **Reduce the configuration**

③ Further simplify the equation

Equation of Motion



Dynamics of the **system**
(Apply Lagrangian-d'Alembert equation)

- Furthermore, eliminate $\dot{\theta}, \ddot{\theta}$, and obtain the final equation of motion:

$$\left(\frac{d}{dt}\frac{\partial L}{\partial \dot{x}} - \frac{\partial L}{\partial x}\right) + GJ_h^{-T}\left(\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta}\right) = GJ_h^{-T}\tau.$$

Apply steps from last section to using Lagrangian-d'Alembert equation

① Write the equations of motion

② Reduce the configuration

③ **Further simplify the equation**

$$\tilde{M}(q)\ddot{x} + \tilde{C}(q,\dot{q})\dot{x} + \tilde{N}(q,\dot{q}) = F,$$

$$\tilde{M} = M_o + GJ_h^{-T}M_f J_h^{-1}G^T$$

$$\tilde{C} = C_o + GJ_h^{-T}\left(C_f J_h^{-1}G^T + M_f\frac{d}{dt}(J_h^{-1}G^T)\right)$$

$$\tilde{N} = N_o + GJ_h^{-T}N_f$$

$$F = GJ_h^{-T}\tau.$$

Equation of Motion (Conclusion)

- Equation of motion for robot hand

$$\tilde{M}(q)\ddot{x} + \tilde{C}(q,\dot{q})\dot{x} + \tilde{N}(q,\dot{q}) = F,$$

$$\tilde{M} = M_o + GJ_h^{-T}M_f J_h^{-1}G^T$$

$$\tilde{C} = C_o + GJ_h^{-T}\left(C_f J_h^{-1}G^T + M_f \frac{d}{dt}\left(J_h^{-1}G^T\right)\right)$$

$$\tilde{N} = N_o + GJ_h^{-T}N_f$$

$$F = GJ_h^{-T}\tau.$$ If a grasp is force-closure, this term is *internal forces*

- Properties of the derived equation of motion *(Temporally Proof omitted)*

1. $\tilde{M}(q)$ *is symmetric and positive definite.*

2. $\dot{\tilde{M}}(q) - 2\tilde{C}$ *is a skew-symmetric matrix.*

Finding Contact Force

- Goal: Find the instantaneous contact forces during motion.
- *Internal forces*: if a grasp is force-closure, then there exist contact forces which produce no net wrench on the object.
- In dynamics, internal forces $F = GJ_h^{-T}\tau$ maps joint torques into object forces.
  - If $J_h^{-T}\tau \in \mathcal{N}(\mathcal{G})$, no net wrench is generated
  - But even if $J_h^{-T}\tau \notin \mathcal{N}(\mathcal{G})$, internal forces still exists due to those *constraint forces* which the Lagrange-d'Alembert equations eliminated.
- Recall full equation of motion with pfaffian constraints:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} + A^T(q)\lambda - \Upsilon = 0, \qquad A(q) = \begin{bmatrix} -J_h(\theta, x) & G^T(\theta, x) \end{bmatrix}$$

$$\begin{bmatrix} M_f & 0 \\ 0 & M_o \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{x} \end{bmatrix} + \begin{bmatrix} C_f & 0 \\ 0 & C_o \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \dot{x} \end{bmatrix} + \begin{bmatrix} N_f \\ N_o \end{bmatrix} + \begin{bmatrix} -J_h^T \\ G \end{bmatrix} \lambda = \begin{bmatrix} \tau \\ 0 \end{bmatrix}$$

Lagrangian multiplier λ: contact forces

Finding Contact Force

- Solve for Lagrange multiplier using results in *Section 1.2. Lagrange Multipliers*

$$\underbrace{\begin{bmatrix} M_f & 0 \\ 0 & M_o \end{bmatrix}}_{\bar{M}} \begin{bmatrix} \ddot{\theta} \\ \ddot{x} \end{bmatrix} + \underbrace{\begin{bmatrix} C_f & 0 \\ 0 & C_o \end{bmatrix}}_{\bar{C}} \begin{bmatrix} \dot{\theta} \\ \dot{x} \end{bmatrix} + \underbrace{\begin{bmatrix} N_f \\ N_o \end{bmatrix}}_{\bar{N}} + \begin{bmatrix} -J_h^T \\ G \end{bmatrix} \lambda = \begin{bmatrix} \tau \\ 0 \end{bmatrix}$$

$$\lambda = (A\bar{M}^{-1}A^T)^{-1} \left(A\bar{M}^{-1} \left(\begin{bmatrix} \tau \\ 0 \end{bmatrix} - \bar{C}\dot{q} - \bar{N} \right) + \dot{A}\dot{q} \right).$$

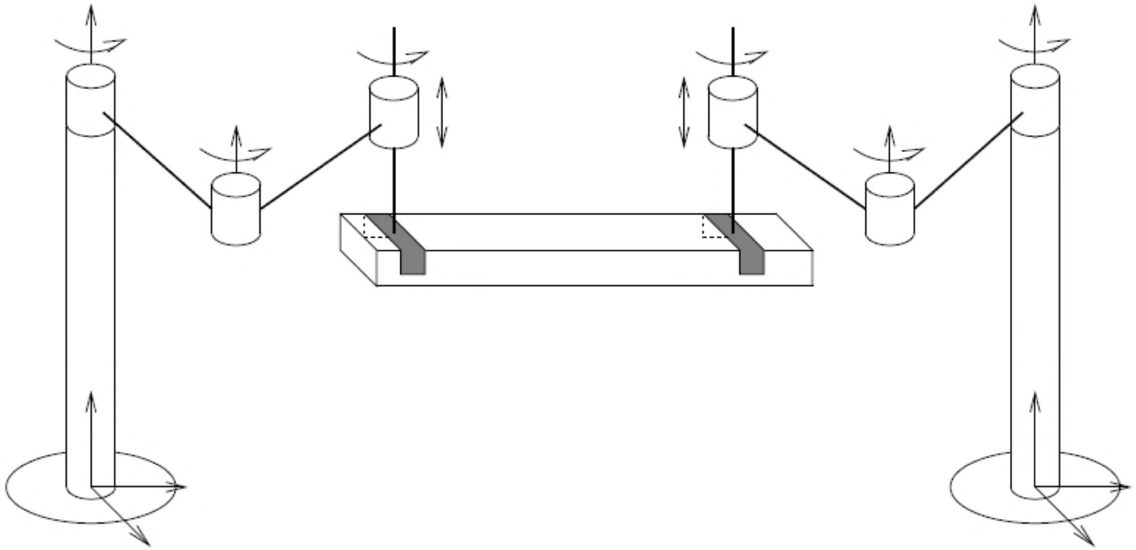
- Another method to solve for constraint forces
  - If J_h is invertible, directly using the joint acceleration.

$$\lambda = J_h^{-T} \left(\tau - M_f\ddot{\theta} - C_f\dot{\theta} - N_f \right).$$

Other Robot Systems

- Let's see some examples.
- Robot system subject to constrains of $J(q)\dot{\theta} = G^T(q)\dot{x}$ have dynamics with the same form and structure we introduced before.

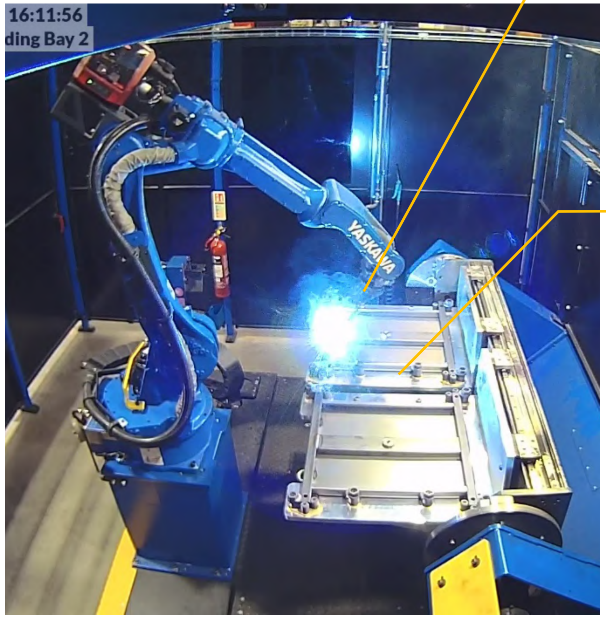
Coordinated lifting



$$\underbrace{\begin{bmatrix} \mathrm{Ad}^{-1}_{g_{s_1t_1}} J^s_{s_1t_1} & & 0 \\ & \ddots & \\ 0 & & \mathrm{Ad}^{-1}_{g_{s_kt_k}} J^s_{s_kt_k} \end{bmatrix}}_{J} \dot{\theta} = \underbrace{\begin{bmatrix} \mathrm{Ad}^{-1}_{g_{ot_1}} \\ \vdots \\ \mathrm{Ad}^{-1}_{g_{ot_k}} \end{bmatrix}}_{G^T} V^b_{po}.$$

Other Robot Systems

Workspace dynamics



Motoman robot performing a welding task
Robot grasping a welding tool

Dynamics of the **welding tool**

$$M_o(x)\ddot{x} + C_o(x,\dot{x})\dot{x} + N_o(x,\dot{x}) = 0,$$

Dynamics of the **system**

- $g : Q \rightarrow \mathbb{R}^p$, Jacobian: $J(\theta) = \frac{\partial g}{\partial \theta}$
- Kinematics: $J(\theta)\dot{\theta} = \dot{x},$
- Dynamics: $\tilde{M}(q)\ddot{x} + \tilde{C}(q,\dot{q})\dot{x} + \tilde{N}(q,\dot{q}) = F,$

$$\tilde{M} = M_o + J^{-T} M_f J^{-1}$$

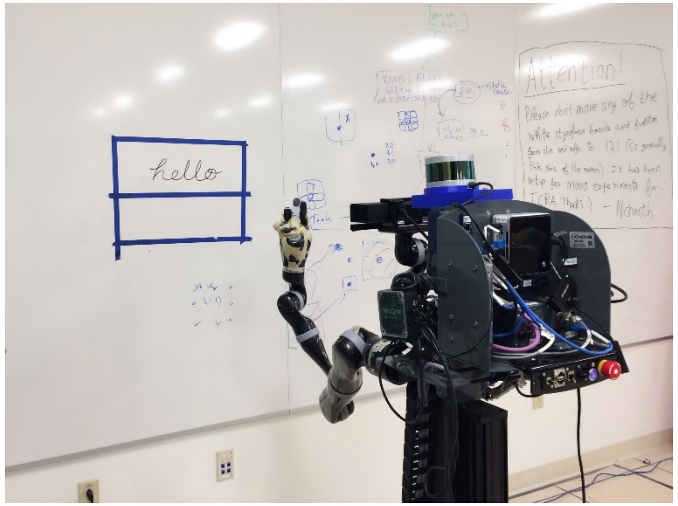
$$\tilde{C} = C_o + J^{-T}\left(C_f J^{-1} + M_f \frac{d}{dt}\left(J^{-1}\right)\right)$$

$$\tilde{N} = N_o + J^{-T} N_f$$

$$F = J^{-T}\tau$$

Other Robot Systems

Hybrid position/force dynamics



Robot writing on a planar

- This kind of tasks consist of both a desired motion and a desired force
- Constraint: $h(\theta, x) = 0$

$$\underbrace{\frac{\partial h}{\partial \theta}}_{J} \dot{\theta} = -\underbrace{\frac{\partial h}{\partial x}}_{G^T} \dot{x}$$

- Dynamics: $\tilde{M}(q)\ddot{x} + \tilde{C}(q, \dot{q})\dot{x} + \tilde{N}(q, \dot{q}) = F,$

$$\tilde{M} = GJ^{-T} M_f J^{-1} G^T$$

$$\tilde{C} = GJ^{-T} \left(C_f J^{-1} G^T + M_f \frac{d}{dt} \left(J^{-1} G^T \right) \right)$$

$$\tilde{N} = N_o + GJ^{-T} N_f$$

$$F = GJ^{-T} \tau.$$

Contents of This Talk

- Recall
- Lagrange's Equations with Constraints
- Robot Hand Dynamics
- **Redundant and Nonmanipulable Robot Systems**
    - Dynamics of redundant manipulator
    - Nonmanipulable grasps
    - Example: Two-fingered SCARA grasp
- Kinematics and Statics of Tendon Actuation
- Control of Robot Hand

Dynamics for These Robot Systems (Conclusion)

- How to analyze dynamics *redundant* and/or *nonmanipulable* robot systems subject to constraints?
- Constraints: $J_h(\theta, x)\dot{\theta} = G^T(\theta, x)\dot{x}$

	Redundant	Nonmanipulable
What it is	Constraints introduce *kinematic/actuator redundancy* into robot system. • *Kinematic redundancy* :finger **motions** which do not affect object motion. • *Actuator redundancy* : finger forces which do not affect object motion. i.e., Internal forces.	• *Manipulable*: when arbitrary motions can be generated by fingers • *Nonmanipulable*: when finger motion cannot achieve some motions of the individual contacts.
What J_h looks like	• J_h has a non-trivial null space, which describes those joint **motions**.	• J_h is not full row rank • J_h does not span the range of G^T
How to write equation of motion	Extend the constraints by brining K_h which spans the null space of J_h. $\underbrace{\begin{bmatrix} J_h \\ K_h \end{bmatrix}}_{\bar{J}_h} \dot{\theta} = \underbrace{\begin{bmatrix} G^T & 0 \\ 0 & I \end{bmatrix}}_{\bar{G}^T} \begin{bmatrix} \dot{x} \\ v_N \end{bmatrix}$	Rewrite the constraints by bringing H which spans the space of allowable object trajectories. $J_h\dot{\theta} = \underbrace{G^T H}_{\bar{G}^T} w$

Examples: Two-fingered SCARA grasp

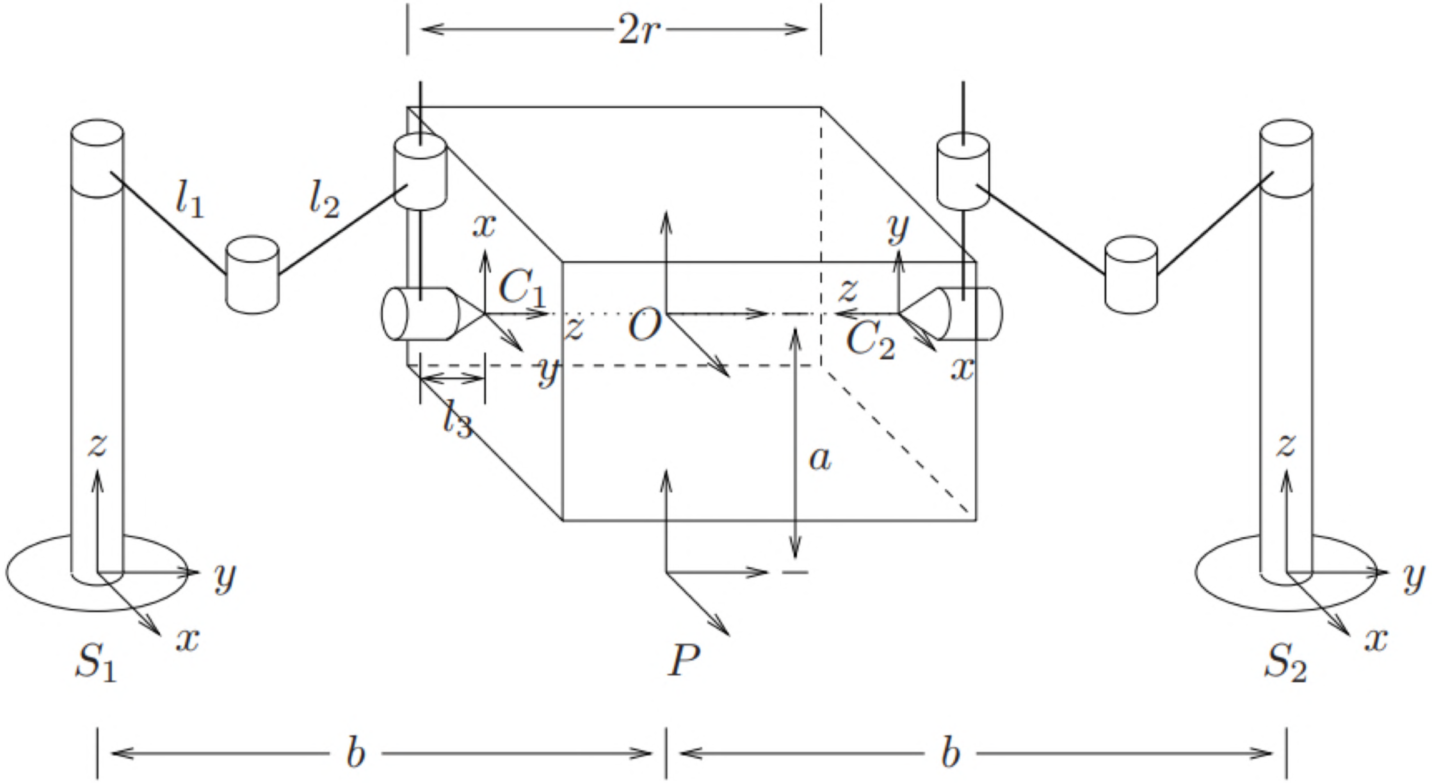
- Write the basic grasp constraints:

$$\begin{array}{c c} & \\ 8 & \begin{bmatrix} J_{h1} & 0 \\ 0 & J_{h2} \end{bmatrix} \\ & \underbrace{\hphantom{J_{h1} \quad J_{h2}}}_{8} \end{array} \dot{\theta} = \begin{array}{c} \begin{bmatrix} G_1^T \\ G_2^T \end{bmatrix} \\ 8 \end{array} V_{po}^b.$$

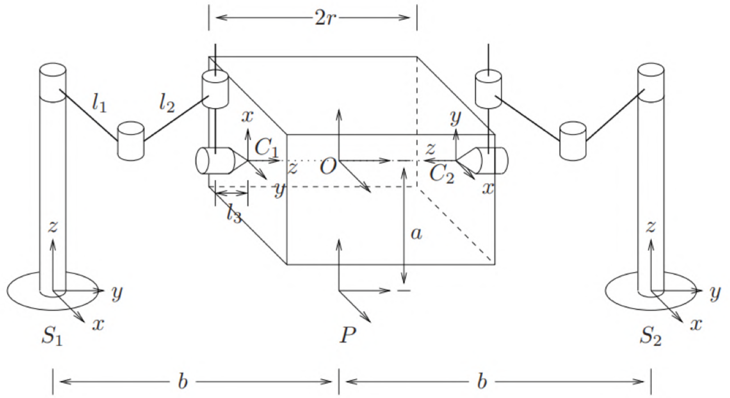
Notice this $J_h(\theta)$ is not invertible

i. **Solve for redundancy**

ii. **Solve for Nonmanipulable**



Examples: i. Solve for Redundancy

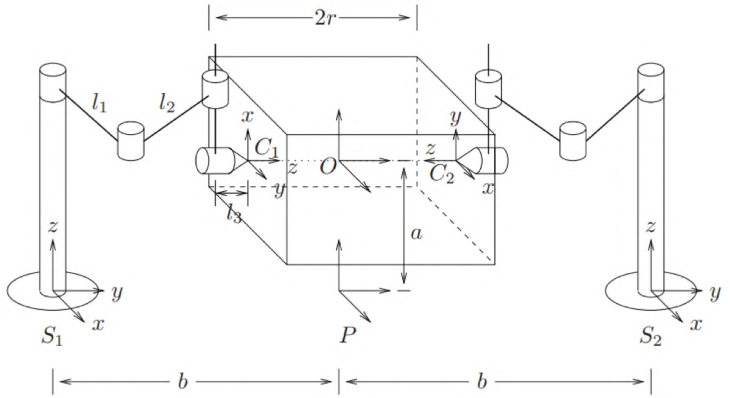


- Define K where $\frac{\partial y}{\partial \theta} = K(\theta)$.
  - We define $h(\theta) = (\theta_{11} + \theta_{12} + \theta_{13}, \theta_{21} + \theta_{22} + \theta_{23})$
  - So that $K_1 = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$, $K_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}$
- Expand the constraints:

$$10 \updownarrow \underbrace{\left[\begin{array}{c|c} J_{h1} & 0 \\ 0 & J_{h2} \\ \hline \begin{matrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{matrix} \end{array}\right]}_{8} \dot{\theta} = \underbrace{\left[\begin{array}{c|c} G_1^T & 0 \\ G_2^T & 0 \\ \hline 0 & I \end{array}\right]}_{8} \begin{bmatrix} V_{po}^b \\ \dot{y} \end{bmatrix}$$

- Notice we increased the internal variables to describe the internal motion. i.e. velocity $\dot{y}$.
- But it does not alter the nonmanipulable nature since J_h still does not span the range of G^T.

Examples: ii. Solve for Nonmanipulable



- Define the space of allowable object velocities
  - $W(\theta, x) = \{\dot{x} \in \mathbb{R}^p : \exists\, \dot{\theta} \in \mathbb{R}^m \text{ with } J_h\dot{\theta} = G^T\dot{x}\}.$

    ⇑ *It has l dimensions*
  - i.e. Object can move along $[0,1,0,0,0,0]^T$
  - i.e. But object cannot move along $[0,0,0,0,1,0]^T$ (Rotating around Y-axis)
- Next, we construct a matrix $H(\theta, x) \in \mathbb{R}^{p\times l}$ using $W(\theta, x)$
  - Every column of H is the allowing object velocity in W (basis)

$$H = \left[\begin{array}{ccccc|cc} 1&0&0&0&0&0&0\\ 0&1&0&0&0&0&0\\ 0&0&1&0&0&0&0\\ 0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0\\ \hline 0&0&0&0&0&1&0\\ 0&0&0&0&0&0&1 \end{array}\right] = \left[\begin{array}{c|c} H' & 0\\ \hline 0 & I \end{array}\right]$$

- Rewrite grasp constraints:

$$\underset{10\,\updownarrow}{} \underbrace{\left[\begin{array}{c|c} J_{h1} & 0\\ 0 & J_{h2}\\ \hline K_1 & K_2 \end{array}\right]}_{8}\dot{\theta} = \underbrace{\left[\begin{array}{c|c} G_1^T H' & 0\\ G_2^T H' & 0\\ \hline 0 & I \end{array}\right]}_{7}\begin{bmatrix} w'\\ \dot{y} \end{bmatrix}$$

Recall rewritten formulation

$$J_h\dot{\theta} = G^T H w$$
$$\dot{x} = Hw,$$

$\dot{x} \in \mathbb{R}^p$: object velocity
$w \in \mathbb{R}^l$: object velocity in terms of the basis of H

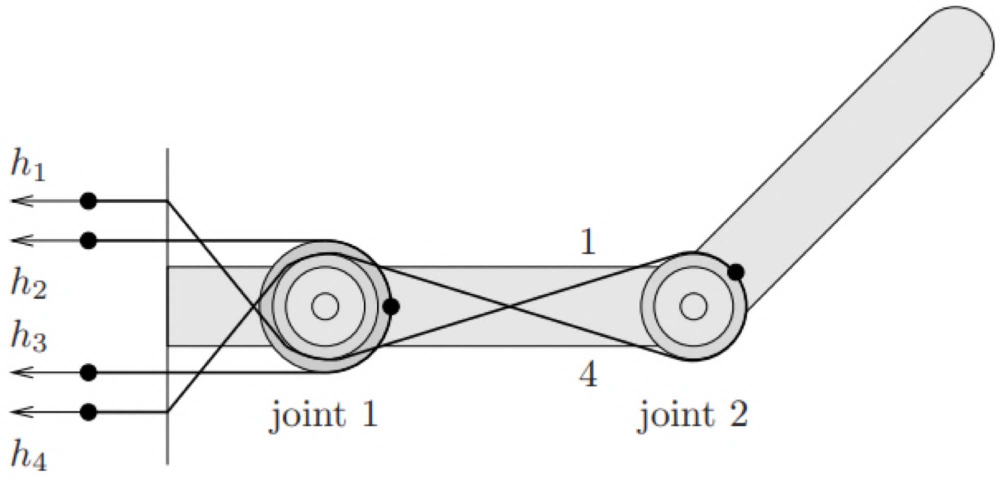
Contents of This Talk

- Recall
- Lagrange’s Equations with Constraints
- Robot Hand Dynamics
- Redundant and Nonmanipulable Robot Systems
- **Kinematics and Statics of Tendon Actuation**
  - Inelastic tendons
  - Elastic tendons
  - Analysis and control of tendon-driven fingers
- Control of Robot Hand

Tendon-Driven Finger

- Introduce a mechanism to carry forces from an actuator to the appropriate joint.
- Model the routing of each tendon by an *extension function*:
  - $h_i: \mathrm{Q} \to \mathbb{R}$
  - It measures the displacement of tendon end and the joint angles of the finger
  - i.e. $h_i(\theta) = \underbrace{l_i}\pm \underbrace{r_{i1}}\theta_1 \pm \cdots \pm r_{in}\theta_n$

l_i: Nominal extension (at $\theta = 0$)
r_{ij}: radius of the j-th joint pulley

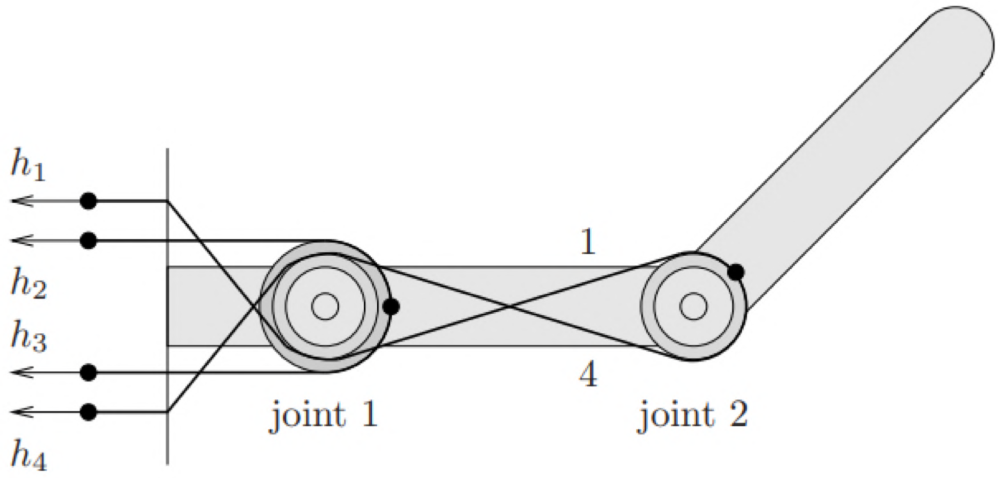


A simple tendon-driven finger
Consists of linkages, tendons, gears, and pulleys

Inelastic Tendons

- Introduce a mechanism to carry forces from an actuator to the appropriate joint.
- Model the routing of each tendon by an *extension function*:
  - $h_i : \mathrm{Q} \to \mathbb{R}$
  - It measures the displacement of tendon end and the joint angles of the finger
  - i.e. $h_i(\theta) = \underbrace{l_i}_{} \pm \underbrace{r_{i1}}_{}\theta_1 \pm \cdots \pm r_{in}\theta_n$

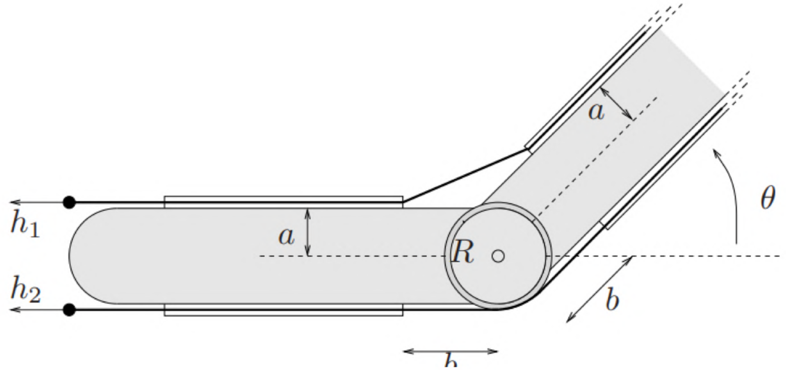
l_i: Nominal extension (at $\theta = 0$)
r_{ij}: radius of the j-th joint pulley



A finger which is actuated by a set of inelastic tendons

Inelastic Tendons

- Finger examples and their extension functions

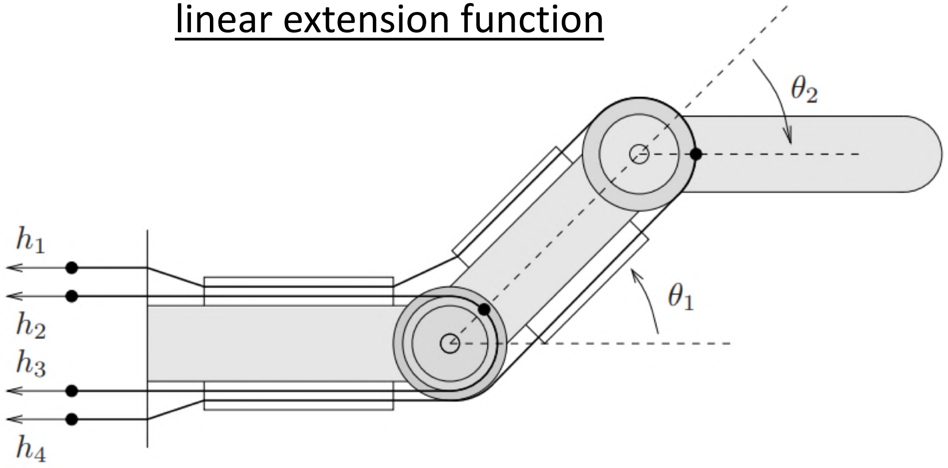


Example of tendon routing with non linear extension function

- Extension functions:

$$h_1(\theta) = l_1 + 2\sqrt{a^2 + b^2}\cos\left(\tan^{-1}\left(\frac{a}{b}\right) + \frac{\theta}{2}\right) - 2b \qquad \theta > 0$$

$$h_2(\theta) = l_2 + R\theta, \qquad \theta > 0.$$



Planar tendon-driven finger

- Extension functions:

$$h_2 = l_2 - R_1\theta_1$$
$$h_3 = l_3 + R_1\theta_1.$$

$$h_1 = l_1 + 2\sqrt{a^2 + b^2}\cos(\tan^{-1}\left(\frac{a}{b}\right) + \frac{\theta_1}{2}) - 2b - R_2\theta_2 \qquad \theta_1 > 0.$$
$$h_4 = l_4 + R_1\theta_1 + R_2\theta_2$$

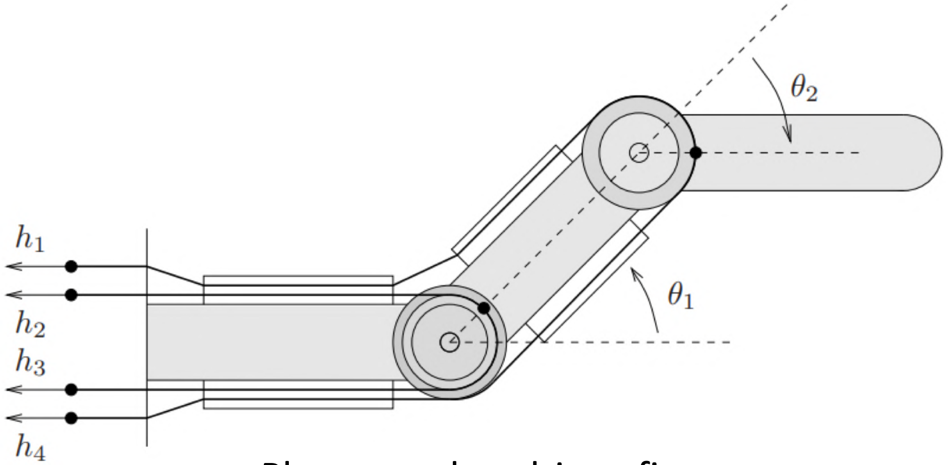
Inelastic Tendons

- Let's define the relationships between the tendon forces and the joint torques using tendon extension functions.
  - Tendon extensions vectors with p tendons: $e = h(\theta) \in \mathbb{R}^p$
  - Define *coupling matrix*: $P(\theta) = \frac{\partial h^T}{\partial \theta}(\theta)$ mapping tendon forces and the joint torques
  - So $\dot{e} = \frac{\partial h}{\partial \theta}(\theta)\dot{\theta} = P^T(\theta)\dot{\theta}.$
  - Since work done by the tendons must equal that done by the fingers (conservation of energy): $\tau = P(\theta) f$ where $f \in \mathbb{R}^p$ is the force applied to the tendons tends.
- Combined kinematics and dynamics:

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + N(\theta, \dot{\theta}) = P(\theta) f$$

Inelastic Tendons

- $\underbrace{\tau}_{\text{Joint torques}} = \underbrace{P(\theta)}_{\text{Coupling matrix}} \underbrace{f}_{\text{Tendon forces}}$
- $M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + N(\theta, \dot{\theta}) = P(\theta) f$
- An example



Planar tendon-driven finger

  - Extension function

$$h_2 = l_2 - R_1\theta_1 \qquad h_1 = l_1 + 2\sqrt{a^2+b^2}\cos\left(\tan^{-1}\left(\frac{a}{b}\right) + \frac{\theta_1}{2}\right) - 2b - R_2\theta_2 \qquad \theta_1 > 0.$$

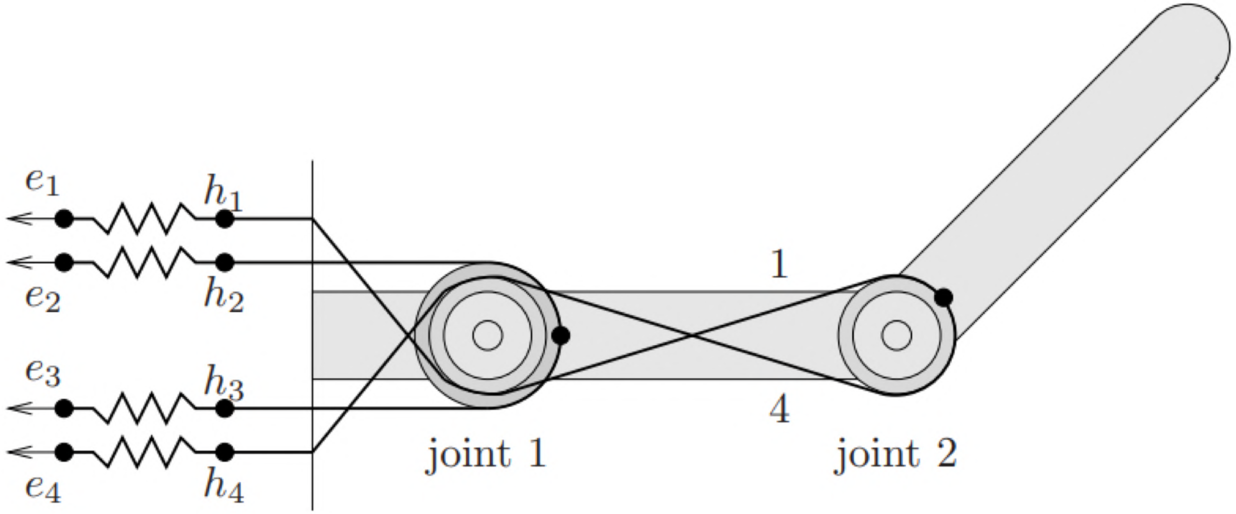
$$h_3 = l_3 + R_1\theta_1. \qquad h_4 = l_4 + R_1\theta_1 + R_2\theta_2$$

  - Coupling matrix

$$P(\theta) = \frac{\partial h}{\partial \theta}^T = \begin{bmatrix} -\sqrt{a^2+b^2}\sin\left(\tan^{-1}\left(\frac{a}{b}\right) + \frac{\theta_1}{2}\right) & -R_1 & R_1 & R_1 \\ -R_2 & 0 & 0 & R_2 \end{bmatrix}$$

Elastic Tendons

- Applying a single spring element at the base of the tendon:



Planar finger with position-controlled elastic tendons

- Extension functions

$$
\begin{aligned}
h_1 &= l_1 + r_{11}\theta_1 - r_{12}\theta_2 \\
h_2 &= l_2 - r_{21}\theta_1 \\
h_3 &= l_3 + r_{31}\theta_1 \\
h_4 &= l_4 - r_{41}\theta_1 + r_{42}\theta_2,
\end{aligned}
$$

- Coupling Matrix

$$
P(\theta) = \frac{\partial h}{\partial \theta}^T = \begin{bmatrix} r_{11} & -r_{21} & r_{31} & -r_{41} \\ -r_{12} & 0 & 0 & r_{42} \end{bmatrix}
$$

- We also want to establish the relationship between tendon extension and the joint torques using a *new coupling matrix*

Elastic Tendons

- Let's define the relationships between the tendon extension and the joint torques using a *new coupling matrix*.
- Extension of the tendon as commanded by the actuator: e_i
- Extension of the tendon due to the mechanism: $h_i(\theta)$
- Net force applied to tendons: $f_i = k_i(e_i + h_i(0) - h_i(\theta))$
- Define K: diagonal matrix of tendon stiffnesses, where k_i is the stiffness of i-th tendon

$$f = K(e + h(0) - h(\theta))$$

- Write dynamics:

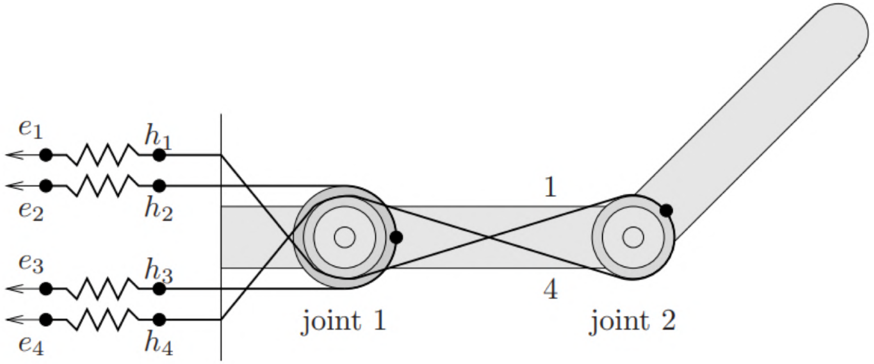
$$M(\theta) + C(\theta, \dot{\theta})\dot{\theta} + N(\theta, \dot{\theta}) + \underbrace{PK(h(\theta) - h(0))}_{\text{Models the stiffness of the tendon network}} = \underbrace{PK}_{\text{New coupling matrix}}e$$

$$S(\theta) := PK(h(\theta) - h(0)) \qquad Q := PK$$

Elastic Tendons

- $\underbrace{\tau}_{\text{Joint torques}} = \underbrace{Q}_{\text{New coupling matrix}}\underbrace{e}_{\text{Tendon extension}}, Q := PK$

- $M(\theta) + C(\theta, \dot{\theta})\dot{\theta} + N(\theta, \dot{\theta}) + PK(h(\theta) - h(0)) = PKe$



Planar finger with position-controlled elastic tendons

- An example (top-right finger):
  - We already wrote the extension function h_1, h_2, h_3, h_4 and coupling matrix $P(\theta)$
  - Stiffness matrix :

$$K = \begin{bmatrix} k_1 & 0 & 0 & 0 \\ 0 & k_2 & 0 & 0 \\ 0 & 0 & k_3 & 0 \\ 0 & 0 & 0 & k_4 \end{bmatrix}$$

  - Overall stiffness:

$$S(\theta) = PK(h(\theta) - h(0)) = \begin{bmatrix} k_1 r_{11}^2 + k_2 r_{21}^2 + k_3 r_{31}^2 + k_4 r_{41}^2 & -k_1 r_{11} r_{12} - k_4 r_{41} r_{42} \\ -k_1 r_{11} r_{12} - k_4 r_{41} r_{42} & k_1 r_{12}^2 + k_4 r_{42}^2 \end{bmatrix} \theta$$

  - *New coupling matrix* that mapping joint torques and tendon extension

$$Q = PK = \begin{bmatrix} k_1 r_{11} & -k_2 r_{21} & k_3 r_{31} & -k_4 r_{41} \\ -k_1 r_{12} & 0 & 0 & k_4 r_{42} \end{bmatrix}$$

Control of Tendon-Driven Fingers

- First, define a tendon network is *force-closure*:
  - For any $\tau \in \mathbb{R}^n$ there exists a set of forces $f \in \mathbb{R}^p$ such that

    $$P(\theta)f = \tau \qquad \text{and} \qquad f_i > 0,\ i = 1, \ldots, p.$$

  - So the necessary and sufficient condition is P be surjective and there exist a strictly positive vector of internal forces $f_N \in \mathbb{R}^p$, $f_{N,i} > 0$ such that $P(\theta) f_N = 0$
- Verify the necessary number of tendons to construct a *force-closure* tendon network:
  - *"N+1"* tendon configuration:
    - N tendons which generate torques in the opposite direction
    - *1* tendon which pulls on all of the joints in one direction
  - *"2N"* tendon configuration:
    - *2* tendons to each joint (total N joints), acting in opposite directions

Control of Tendon-Driven Fingers

- Next, write the tendon forces for **inelastic** tendons:

$$f = \underbrace{P^{+}}_{\text{pseudo-inversed coupling matrix}}(\theta)\tau + \underbrace{f_N}_{\text{Internal forces to ensure all } h > 0}$$

- Also, let's move on to elastic tendons:
  - We must solve the following equations:

$$P(\theta)Ke = \tau \quad \text{and} \quad e_i + h_i(0) - h_i(\theta) > 0,\ i = 1, \ldots, p.$$

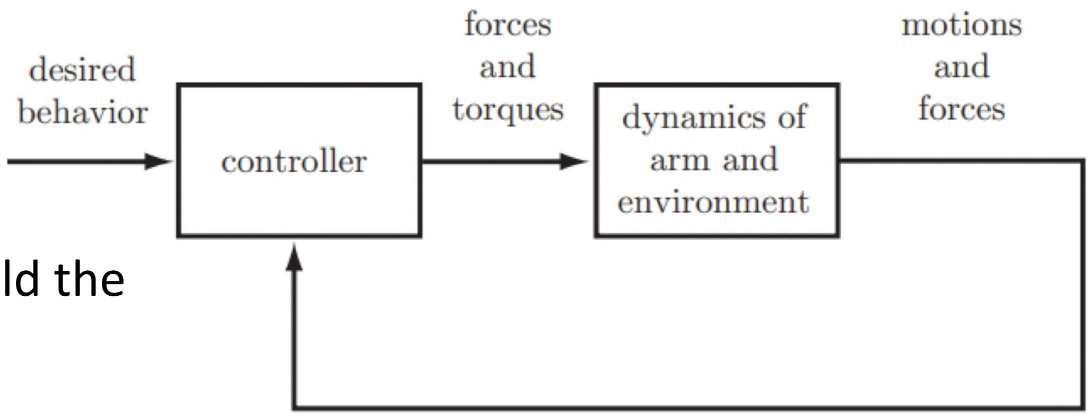
  - How to solve: assume tendon network is force-closure, there exists a vector of extensions $e_N \in \mathbb{R}^p$ such that $e_{N,i} > 0$ and $PKe_N = 0$, so we will choose very large e_N we can obtain:

$$e = (PK)^{+}\tau + e_N$$

Contents of This Talk

- Recall
- Lagrange’s Equations with Constraints
- Robot Hand Dynamics
- Redundant and Nonmanipulable Robot Systems
- Kinematics and Statics of Tendon Actuation
- **Control of Robot Hand**
  - Extending controllers
  - Hierarchical control structures

Control



A simple model of robot closed-loop control system

- Recall some definition in Chapter 4:
- *Position control*: given a designed trajectory, how should the joint torques be chosen to follow that trajectory?
    - Desired motion: θ_d
    - Actual motion: θ
    - Error: $e = \theta_d - \theta$
    - Constant gain matrices: K_v, K_p
    - Dynamics (without constraints): $M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + N(\theta,\dot{\theta}) = \tau$
    - Computed torque control law: $\tau = M(\theta)\left(\ddot{\theta}_d - K_v\dot{e} - K_p e\right) + C(\theta,\dot{\theta})\dot{\theta} + N(\theta,\dot{\theta})$
    - Computing torque $\tau = \underbrace{M(\theta)\ddot{\theta}_d + C\dot{\theta} + N}_{\tau_{\mathrm{ff}}} + \underbrace{M(\theta)\left(-K_v\dot{e} - K_p e\right)}_{\tau_{\mathrm{fb}}}$

Control

- Here, we consider robot hand control as control problems with constraints

Goal	How to achieve?
i. Tracking a given object/workspace trajectory	Find joint torques which satisfy the tracking requirement
ii. Maintaining a desired internal force	Add sufficient internal forces to keep the contact forces inside the appropriate friction cones

- We derived dynamics of this kind of constrained system

$$M(q)\ddot{x} + C(q,\dot{q})\dot{x} + N(q,\dot{q}) = F = GJ^{-T}\tau,$$

  - Error: $e \coloneqq x - x_d$

- Let’s achieve these two goal one by one

i. Tracking Trajectory

$$M(q)\ddot{x} + C(q,\dot{q})\dot{x} + N(q,\dot{q}) = F = GJ^{-T}\tau$$

- Given a desired workspace trajectory $x_d(\cdot)$
  - Computed torque controller:

    $$F = M(q)\left(\ddot{x}_d - K_v\dot{e} - K_p e\right) + C(q,\dot{q})\dot{x} + N(q,\dot{q})$$

  - From $F = GJ^{-T}\tau$ we can find τ than satisfying F (actually we could find extra τ that corresponds to internal forces)
  - Solve for τ:

    $$\tau = J^T G^+ F + J^T f_N$$

ii. Maintaining Internal Forces

$$M(q)\ddot{x} + C(q,\dot{q})\dot{x} + N(q,\dot{q}) = F = GJ^{-T}\tau,$$

- f_N must be chosen such that the net contact force lies in the friction cone FC
- Two ways to solve for internal forces
  - Method 1: compute final control law

$$\tau = J_h^T G^+ F + J_h^T f_{N,d}$$

  - Method 2: measure the applied internal forces and adjust f_N using a second feedback control law.

$$f = f_d + \alpha \int (f - f_d)\, dt$$

Hierarchical Control Structures

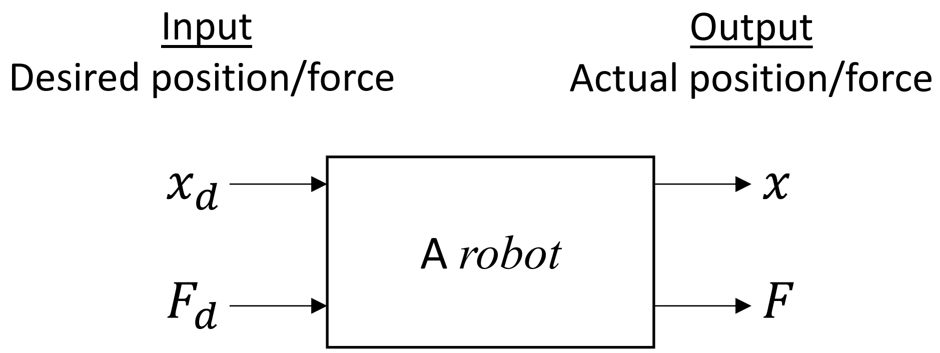
- A multifingered robot hand can be modeled as a set of robots which are coupled to each other and an object by a set of velocity constraints
- Let’s establish the control system following these steps:
    1. Defining robots
    2. Attaching robots
    3. Controlling robots
    4. Building hierarchical controllers

Hierarchical Control Structures

1. **Defining robots**
2. Attaching robots
3. Controlling robots
4. Building hierarchical controllers

- A *robot*: model all mechanisms as a generalized object
- This is system dynamics:

$$M(q)\ddot{x} + C(q,\dot{q})\dot{x} + N(q,\dot{q}) = F$$



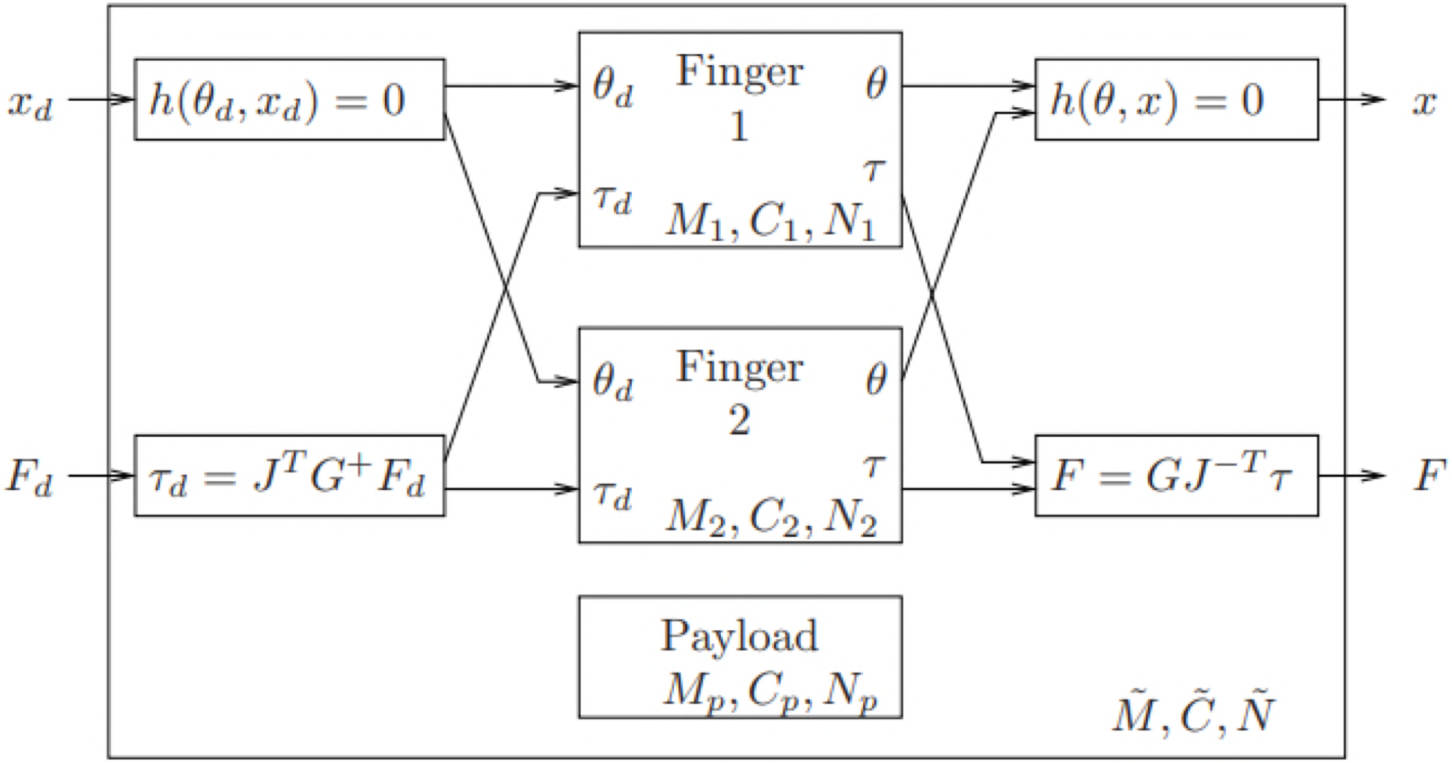
- System dynamic parameter: M, C, N

Hierarchical Control Structures

1. Defining robots
2. **Attaching robots**
3. Controlling robots
4. Building hierarchical controllers

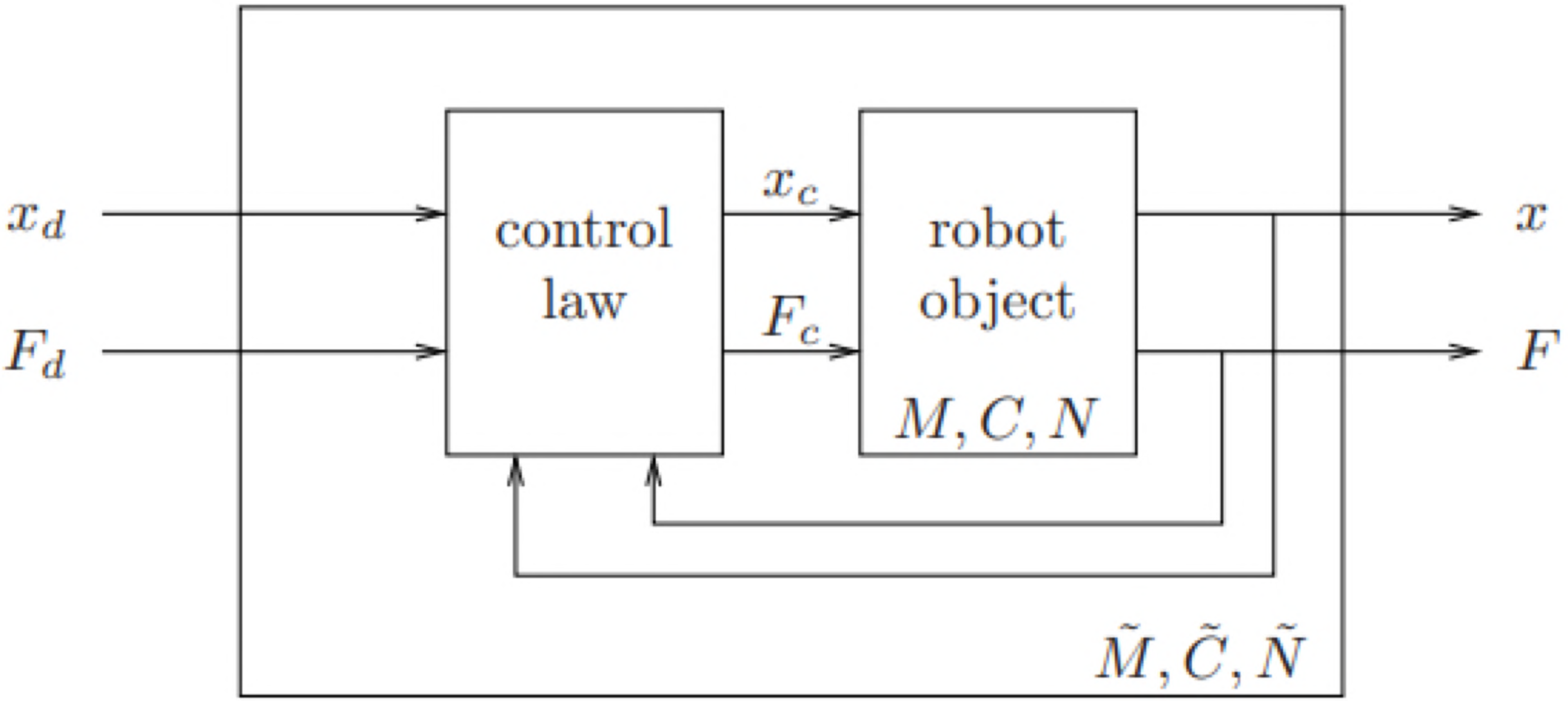
- *Attach* operation: create a new robot object from the attachment (two or more robots)
- System dynamic parameters: $\tilde{M}, \tilde{C}, \tilde{N}$ (on the left)

$$\tilde{M} := M_p + GJ^{-T} M_f J^{-1} G$$
$$\tilde{C} := C_p + GJ^{-T} C_f J^{-1} G + GJ^{-T} M_f \frac{d}{dt}(J^{-1}G)$$
$$\tilde{N} := N_p + GJ^{-T} N_f,$$



Hierarchical Control Structures

1. Defining robots
2. Attaching robots
3. **Controlling robots**
4. Building hierarchical controllers



Hierarchical Control Structures

1. Defining robots
2. Attaching robots
3. Controlling robots
4. **Building hierarchical controllers**

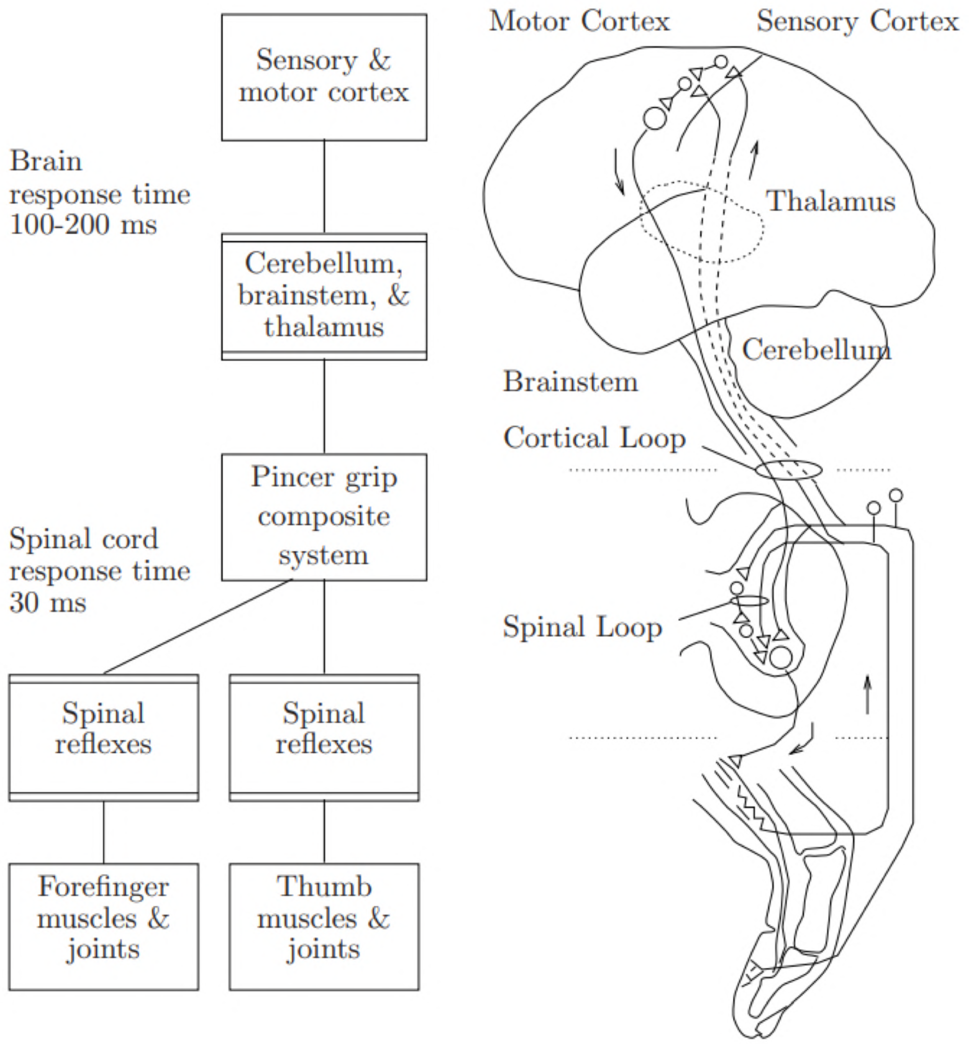
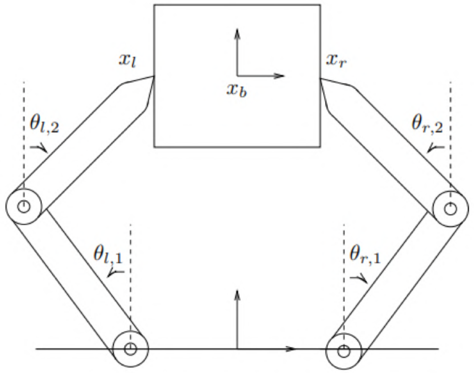


Figure 6.13: Hierarchical control scheme for a human finger. (Figure courtesy of D. Curtis Deno)

Hierarchical Control Structures

1. Defining robots
2. Attaching robots
3. Controlling robots
4. **Building hierarchical controllers**



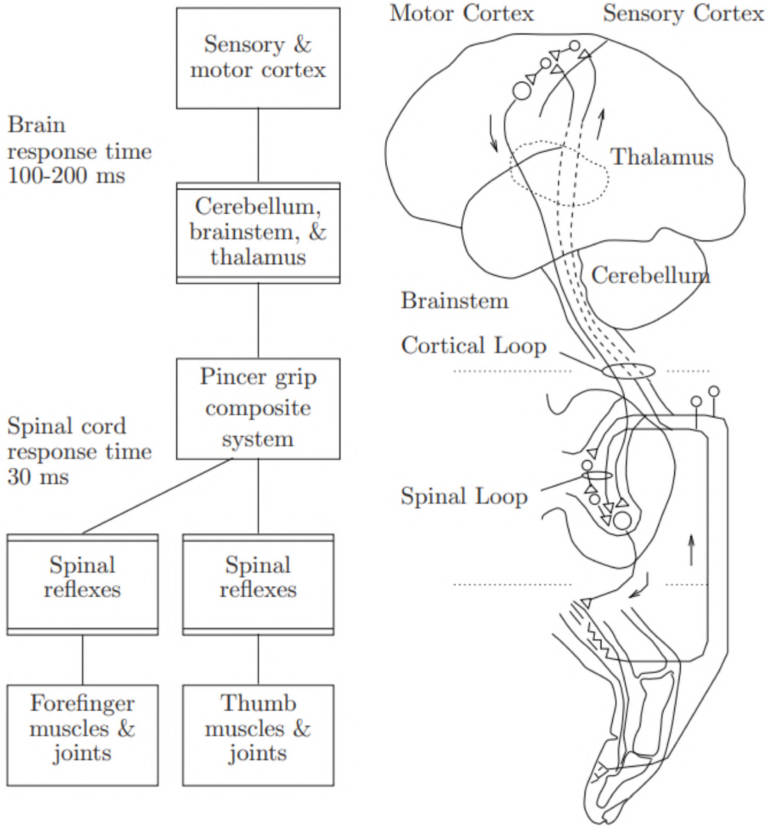
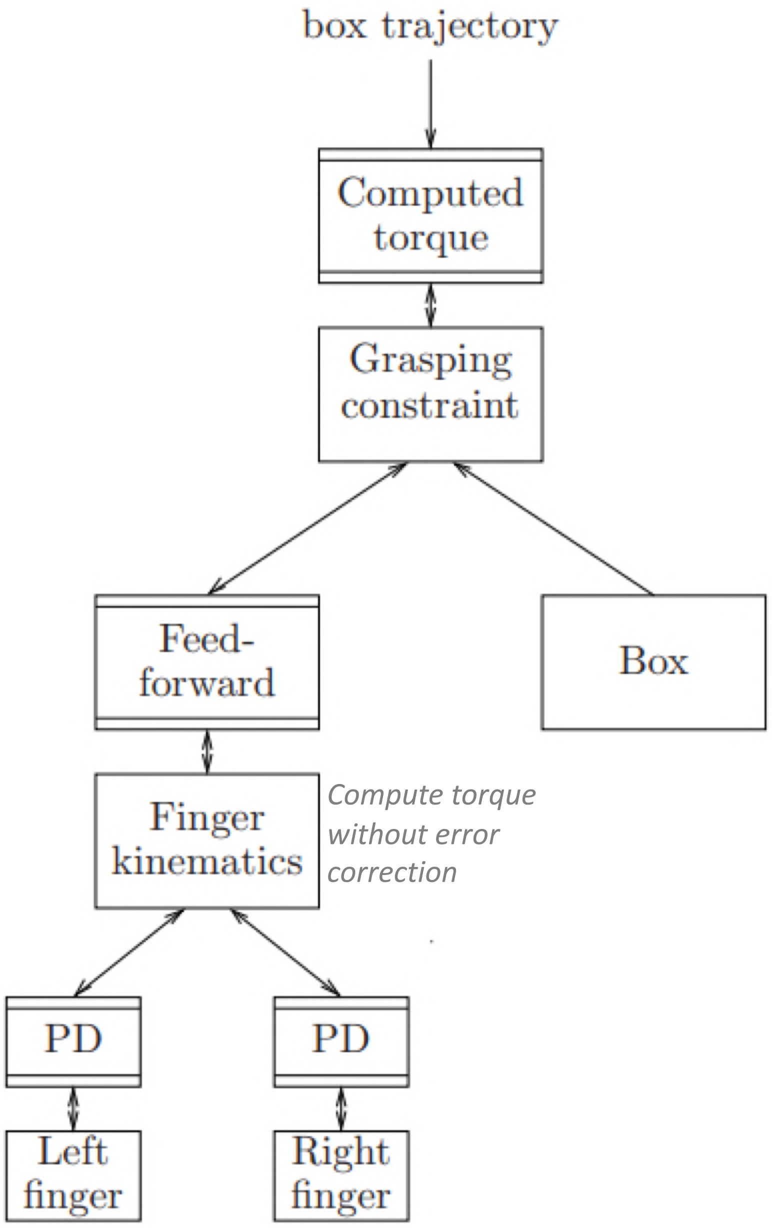


Figure 6.13: Hierarchical control scheme for a human finger. (Figure courtesy of D. Curtis Deno)

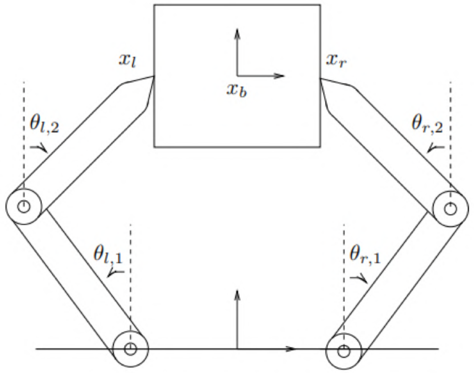


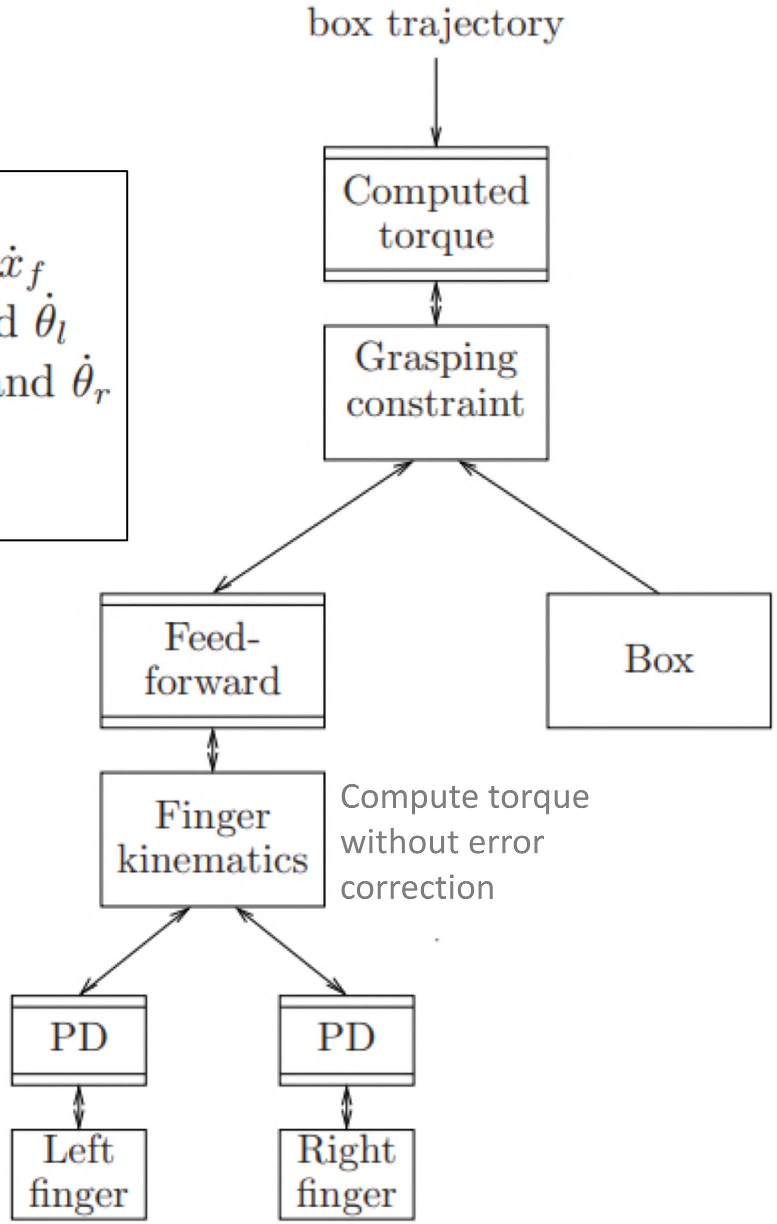
A hierarchical controller for multifingered grasping

Hierarchical Control Structures

1. Defining robots
2. Attaching robots
3. Controlling robots
4. **Building hierarchical controllers**

Hand: asks for current state, x_b and $\dot{x}_b$
Finger: ask for current state, x_f and $\dot{x}_f$
Left: read current state, θ_l and $\dot{\theta}_l$
Right: read current state, θ_r and $\dot{\theta}_r$
Finger: $x_f, \dot{x}_f \leftarrow f(\theta_l, \theta_r), J(\dot{\theta}_l, \dot{\theta}_r)$
Hand: $x_b, \dot{x}_b \leftarrow g(x_f), G^{+T}\dot{x}_f$.





Compute torque without error correction