智能机器人-动力学与控制

—动力学部分—

本章提纲

- **概述**
- 牛顿-欧拉方程（2学时）
- 拉格朗日方程（2学时）
- 投影牛顿欧拉法（PNE）（2学时）
- 案例分析（2学时）

概述

1. 机器人动力学方程：$H(q)\ddot{q}+C(q,\dot{q})\dot{q}+G(q)=\tau$

2. 机器人动力学研究内容：

① 正问题：已知机器人各个关节的驱动力和驱动力矩，求其各关节的位移、速度、加速度。

$$\tau \rightarrow q,\dot{q},\ddot{q}$$

② 反问题：已知机器人各关节的位移、速度和加速度，求其各关节的驱动力或驱动力矩。

$$q,\dot{q},\ddot{q} \rightarrow \tau$$

概述

1. **假设：**

   ① 机器人的各杆件都是刚体（不考虑杆件的变形）；

   ② 机器人的关节为理想运动副（不考虑摩擦和限位）；

   ③ 关节驱动为理想力/力矩（不考虑驱动系统的动力学）

2. **用到的力学原理：**

   多刚体系统适用的力学原理：虚功原理、动量矩定理、能量守恒定理、牛顿－欧拉方程、达朗贝尔原理、拉格朗日方程、哈密尔顿原理、凯恩方程等。

3. **用到的运动学知识：**质心的位置、速度和加速度。

概述

1. 牛顿-欧拉方程：分别列写每一个杆件的平动和转动动力学方程，并进行联立和消元；

2. 拉格朗日方程：利用拉格朗日函数列写系统的总能量，求解拉格朗日方程；

3. 投影牛顿欧拉法：矩阵形式的牛顿-欧拉方程；

本章提纲

- 概述
- **牛顿-欧拉方程（2学时）**
- 拉格朗日方程（2学时）
- 投影牛顿欧拉法（PNE）（2学时）
- 案例分析（2学时）

牛顿-欧拉方程

1. 质心的平动：

如图所示，假设刚体的质量为m，质心在C点，质心处的位置矢量用$\boldsymbol{C}$表示，则质心处的加速度为$\dot{\boldsymbol{C}}$，根据牛顿方程可得作用在刚体质心C处的力$\boldsymbol{F}$为：

$\boldsymbol{F} = m\ddot{\boldsymbol{C}}$—牛顿方程（平动）

2. 刚体的转动：

刚体绕质心转动的角速度用$\boldsymbol{\omega}$，绕质心的角加速度为$\boldsymbol{\varepsilon}$，根据三维空间欧拉方程，$\boldsymbol{I}_C$为刚体绕质心C的惯性矩阵（张量），作用在刚体上的力矩$\boldsymbol{M}$为：

$\boldsymbol{M} = \boldsymbol{I}_C\boldsymbol{\varepsilon} + \boldsymbol{\omega} \times \boldsymbol{I}_C\boldsymbol{\omega}$—欧拉方程(绕质心转动)

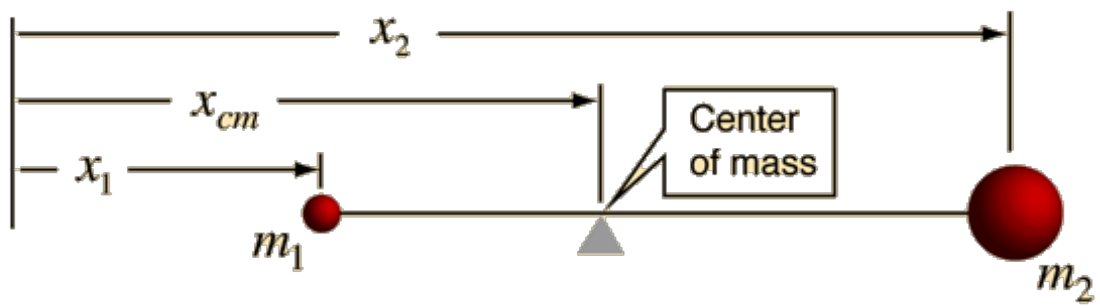
3. 牛顿-欧拉方程:

$$\boxed{\begin{cases} \boldsymbol{F} = m\ddot{\boldsymbol{C}} \\ \boldsymbol{M} = \boldsymbol{I}_C\boldsymbol{\varepsilon} + \boldsymbol{\omega} \times \boldsymbol{I}_C\boldsymbol{\omega} \end{cases}}$$

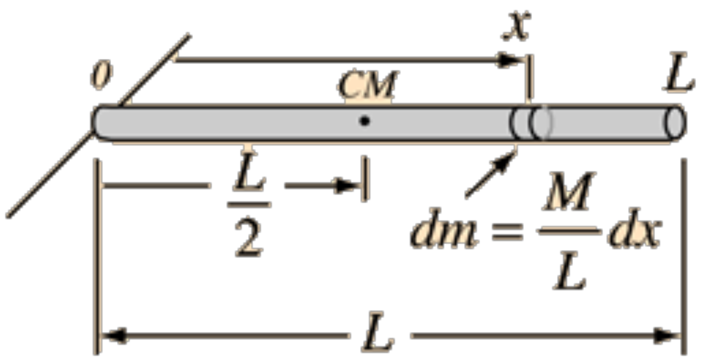
牛顿-欧拉方程

① 质心定义：$x_{com}=\lim_{\Delta m\to 0}\frac{\sum_{i=1}^{N}\Delta m_i x_i}{M}=\frac{\int_0^M x dm}{M}$

② 质点系：$x_{com}=\frac{m_1x_1+m_2x_2}{m_1+m_2}$



③ 连续刚体：$x_{com}=\frac{\int_0^L x\frac{M}{L}dx}{M}=\frac{1}{L}\frac{x^2}{2}\Big|_{x=0}^{x=L}=\frac{L}{2}$



牛顿-欧拉方程

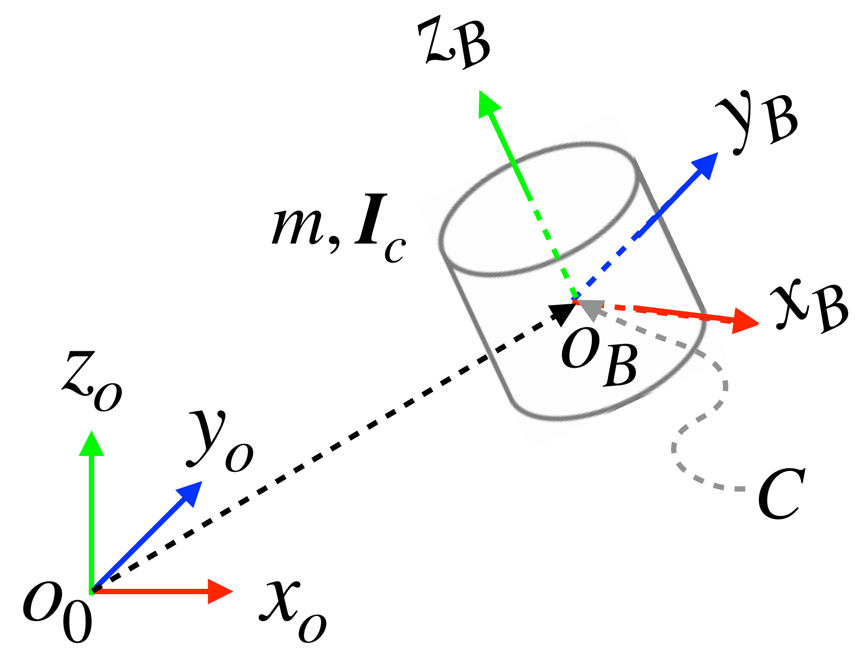
惯性矩阵(张量)：

牛顿-欧拉方程

惯性矩阵(张量):

如图所示，设刚体的质量为 m，相对于质心的惯量矩阵为$\boldsymbol{I}_c$（在刚体固联坐标系Σ_B）：

$$\boldsymbol{I}_c = \begin{bmatrix} \boldsymbol{I}_{xx} & -\boldsymbol{I}_{xy} & -\boldsymbol{I}_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix}$$



(应为${}^B\boldsymbol{I}_c$，为了清晰省略了左上标)

它的六个分量为:

$$\begin{cases} \boldsymbol{I}_{xy} = \boldsymbol{I}_{yx} = \sum m_i x_i y_i = \int x_i y_i dm \\ \boldsymbol{I}_{yz} = \boldsymbol{I}_{zy} = \sum m_i y_i z_i = \int y_i z_i dm \\ \boldsymbol{I}_{zx} = \boldsymbol{I}_{xz} = \sum m_i z_i x_i = \int z_i x_i dm \end{cases}, \quad \begin{cases} \boldsymbol{I}_{xx} = \sum m_i \left(y_i^2 + z_i^2\right) = \int \left(y^2 + z^2\right) dm \\ \boldsymbol{I}_{yy} = \sum m_i \left(z_i^2 + x_i^2\right) = \int \left(z^2 + x^2\right) dm \\ \boldsymbol{I}_{zz} = \sum m_i \left(x_i^2 + y_i^2\right) = \int \left(x^2 + y^2\right) dm \end{cases}$$

主转动惯量：

若存在惯量积为零的Σ_B，其坐标称为惯性主轴，相应的惯量积称为主转动惯量，惯性矩阵简化为：

$$\boldsymbol{I}_c = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}$$

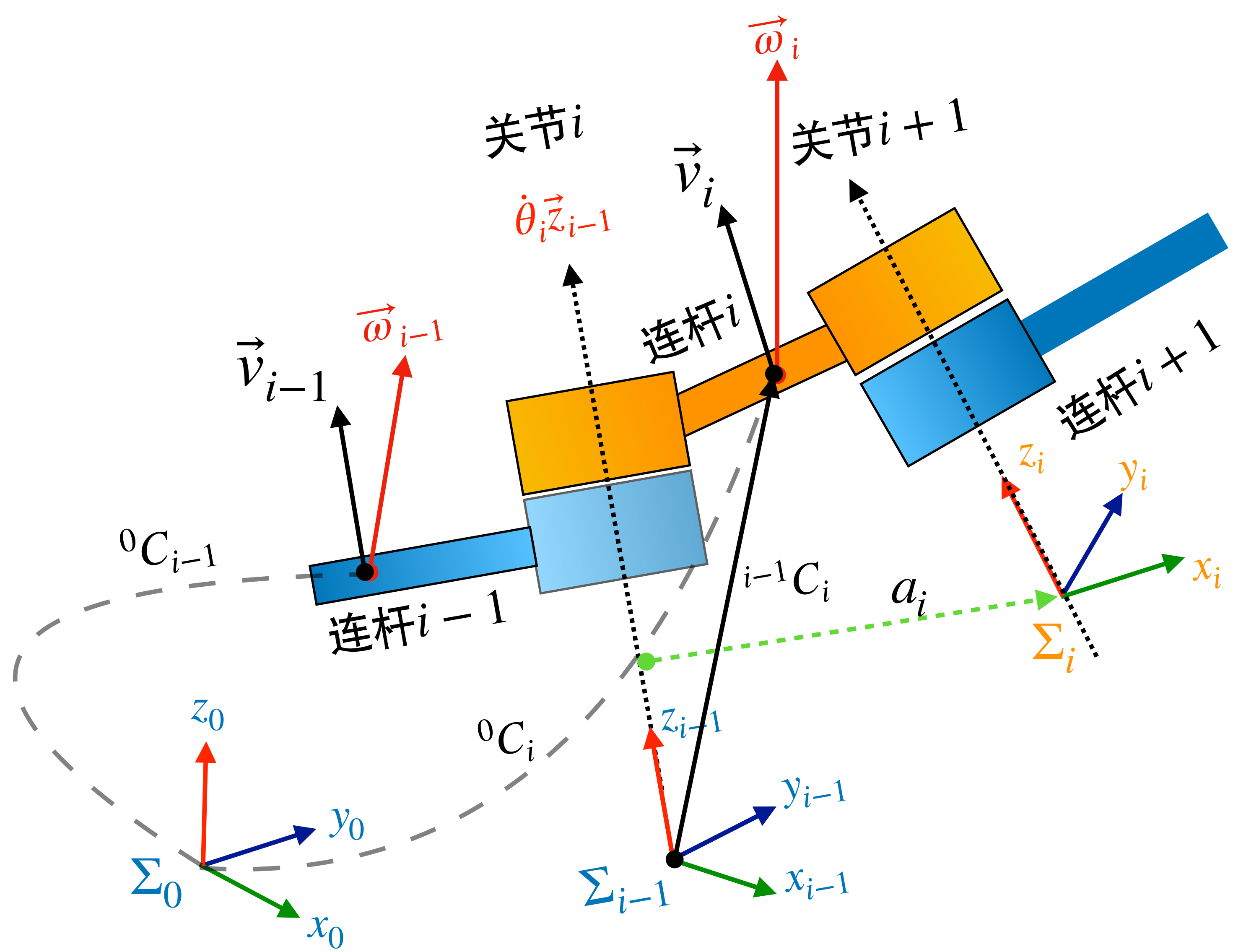
惯性矩阵的平行轴定理（Parallel-axis theorem）：

相对于物体质心的惯量张量${}^B\boldsymbol{I}_C$，则在与其坐标系Σ_B平行的坐标系Σ_A中的惯量张量为${}^A\boldsymbol{I}$，它的分量为：

$$\begin{cases} {}^AI_{xx} = {}^BI_{xx} + m\left(y_c^2 + z_c^2\right) \\ {}^AI_{yy} = {}^BI_{yy} + m\left(x_c^2 + z_c^2\right) \\ {}^AI_{zz} = {}^BI_{zz} + m\left(x_c^2 + y_c^2\right) \end{cases} \text{与} \begin{cases} {}^AI_{xy} = {}^BI_{xy} + m x_c y_c \\ {}^AI_{yz} = {}^BI_{yz} + m y_c z_c \\ {}^AI_{xz} = {}^BI_{xz} + m x_c z_c \end{cases}$$

其中，${}^A\boldsymbol{P}_C = [x_c \quad y_c \quad z_c]^T$为$\Sigma_B$的原点（物体质心$C$）在$\Sigma_A$中的坐标。

牛顿-欧拉方程



DH方法中，坐标系$\{i-1\}$与连杆$\{i-1\}$固联，其原点速度为$\boldsymbol{v}_{i-1}$、加速度为$\boldsymbol{a}_{i-1}$，连杆$\{i-1\}$的角速度为$\boldsymbol{\omega}_{i-1}$、角加速度$\dot{\boldsymbol{\omega}}_{i-1}$；杆件$i$相对坐标轴$z_{i-1}$旋转速率为$\dot{\theta}_i$，则有关于连杆$i$质心处的关系式如下：

牛顿-欧拉方程

角加速度:

i杆的角速度为：$\omega_i = \omega_{i-1} + \dot{\theta}_i z_{i-1}$(回转关节)

i杆的角加速度为：$\dot{\omega}_i = \dot{\omega}_{i-1} + \underbrace{\omega_{i-1} \times (\dot{\theta}_i z_{i-1}) + \ddot{\theta}_i z_{i-1}}$

由红色部分求导得到

(z_{i-1}轴随$i-1$杆以角速度ω_{i-1}运动)

质心的线加速度:

$$v_{ci} = v_{i-1} + \omega_i \times c_i$$

$$a_{ci} = a_{i-1} + \underbrace{\dot{\omega}_i \times c_i + \omega_i \times (\omega_i \times c_i)}$$

由红色部分求导得到，C_i轴随i杆以角速度ω_i运动

同样，可以确定坐标系$\{i\}$原点的加速度为：

$$a_{o,i+1} = a_{i-1} + \dot{\omega}_i \times a_i + \omega_i \times (\omega_i \times a_i)$$

⇐仅仅为了说明计算公式

注：以上变量都是在基坐标系下表示的，因此省略了左上角的坐标系上标

牛顿-欧拉方程

作用力和力矩：

将第i个杆件L_i作为隔离体进行分析，作用在其上的力和力矩有：

① 作用在杆件i上的外力和外力矩；

② $i-1$杆件作用在i杆件上的力和力矩；

③ $i+1$杆件作用在i杆件上的力和力矩；

其中：

① $\boldsymbol{f}_{i+1,i}$—构件L_{i+1}作用在构件L_i上的力。

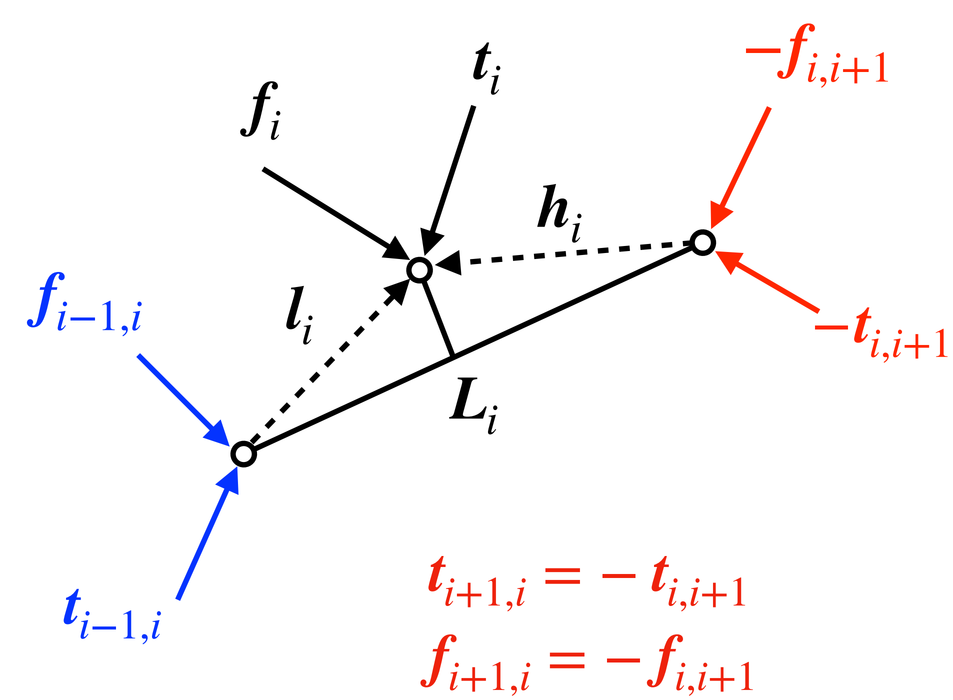
② $\boldsymbol{t}_{i+1,i}$—构件L_{i+1}作用在构件L_i上的力矩。

③ $\boldsymbol{f}_{i-1,i}$—构件L_{i-1}作用在构件L_i上的力。

④ $\boldsymbol{t}_{i-1,i}$—构件L_{i-1}作用在构件L_i上的力矩。

⑤ f_i—作用在第i个构件L_i上的外力简化到质心C处的合力，即外力的主矢。

⑥ t_i—作用在第i个构件L_i上的外力矩简化到质心C处的合力矩，即外力的主矩



另外：

① l_i表示杆件端点到质心的距离；

② $\boldsymbol{h}_i$表示杆件末端到质心的距离；

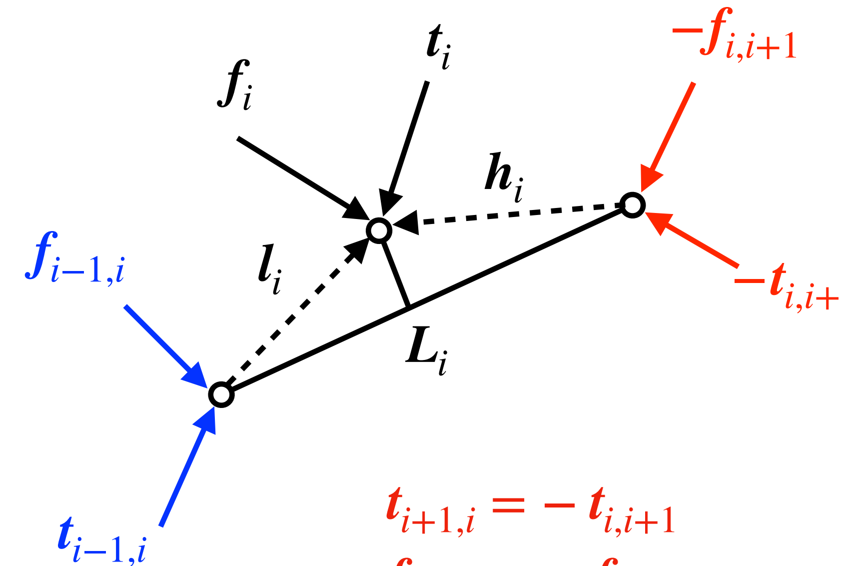
③ 杆件长度为$\boldsymbol{L}_i$；

牛顿-欧拉方程

上述力和力矩包括了运动副中的约束反力、驱动力、摩擦力等引起的作用力和作用力矩。基坐标系下，作用在第i个杆件上的所有力化简到质心的合力、合力矩为：

$$F_i = f_{i-1,i} - f_{i,i+1} + f_i$$

$$T_i = t_{i-1,i} - f_{i-1,i} \times l_i - t_{i,i+1} - f_{i,i+1} \times h_i + t_i$$



$t_{i+1,i} = -t_{i,i+1}$
$f_{i+1,i} = -f_{i,i+1}$

重新整理力和力矩计算公式为：

$$\begin{cases} f_{i-1,i} = f_{i,i+1} - F_i + f_i \\ t_{i-1,i} = \underbrace{f_{i,i+1} \times (l_i + h_i) + t_{i,i+1} + f_i \times l_i - F_i \times l_i - T_i + t_i}_{\text{其中带入了} f_{i-1,i}} \end{cases}$$

则i杆件需要的驱动力矩为$(i-1)$杆件作用于它的力矩在z_{i-1}轴上的分量，即：

$$\tau_i = t_{i-1,i} \cdot z_{i-1}$$

牛顿-欧拉方程的递推算法：

① 从1号杆到n号杆，向外（**Outward**）递推计算各杆的速度和加速度。

② 从n号杆到1号杆，向内（**Inward**）递推计算作用力和力矩，以及关节驱动力矩。

已知：基础杆件和各关节的角速度和角加速度

① 向外递推($i : 1 \to 6$)

$$\begin{cases} \boldsymbol{\omega}_i = \boldsymbol{\omega}_{i-1} + \dot{\theta}_i z_{i-1} \\ \dot{\boldsymbol{\omega}}_i = \dot{\boldsymbol{\omega}}_{i-1} + \boldsymbol{\omega}_{i-1} \times (\dot{\theta}_i z_{i-1}) + \ddot{\theta}_i z_{i-1} \\ \dot{\boldsymbol{v}}_{ci} = \dot{\boldsymbol{v}}_{i-1} + \dot{\boldsymbol{\omega}}_i \times \boldsymbol{c}_i + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \boldsymbol{c}_i) \\ \boldsymbol{F}_i = m_i \dot{\boldsymbol{v}}_{\mathrm{ci}} & \text{惯性力} \\ \boldsymbol{M}_i = \boldsymbol{I}_{ci} \dot{\boldsymbol{\omega}}_i + \boldsymbol{\omega}_i \times \boldsymbol{I}_{ci} \boldsymbol{\omega}_i & \text{惯性力矩} \end{cases}$$

② 向内递推($i : 6 \to 1$)

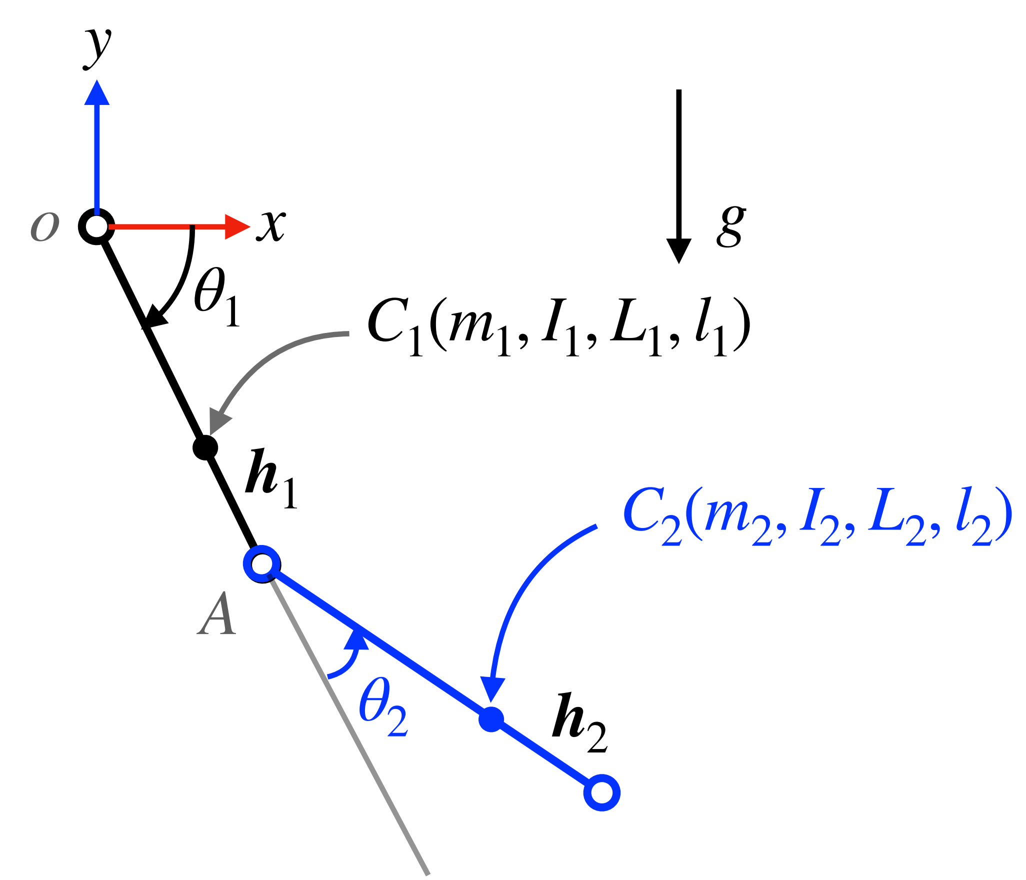
$$\begin{cases} \boldsymbol{f}_{i-1,i} = \boldsymbol{f}_{i,i+1} - \boldsymbol{F}_i + \boldsymbol{f}_i \\ \boldsymbol{t}_{i-1,i} = \boldsymbol{f}_{i,i+1} \times (\boldsymbol{l}_i + \boldsymbol{h}_i) + \boldsymbol{t}_{i,i+1} + \boldsymbol{f}_i \times \boldsymbol{l}_i - \boldsymbol{F}_i \times \boldsymbol{l}_i - \boldsymbol{T}_i + \boldsymbol{t}_i \\ \boldsymbol{\tau}_i = \boldsymbol{t}_{i-1,i} \cdot z_{i-1} \end{cases}$$

例题：

如图所示，竖直平面内的两自由度机器人，关节O和A处的转角为θ_1和θ_2，其中：

① **连杆1：杆长L_1，质心位于l_1，质心距离末端h_1；质量为m_1，绕质心转动惯量$\boldsymbol{I}_1$；驱动力矩为τ_1；与y轴角度为θ_1；**

② **连杆2：杆长L_2，质心位为l_2，质心距离末端h_2；质量为m_2，绕质心转动惯量$\boldsymbol{I}_2$；驱动力矩为τ_2；与连杆1的角度为θ_2；**



求： 该机器人的动力学方程表达式，即求取τ_1、τ_2与θ_1、θ_{12}的各阶导数的关系。

解：

① **运动学向外迭代：连杆1**

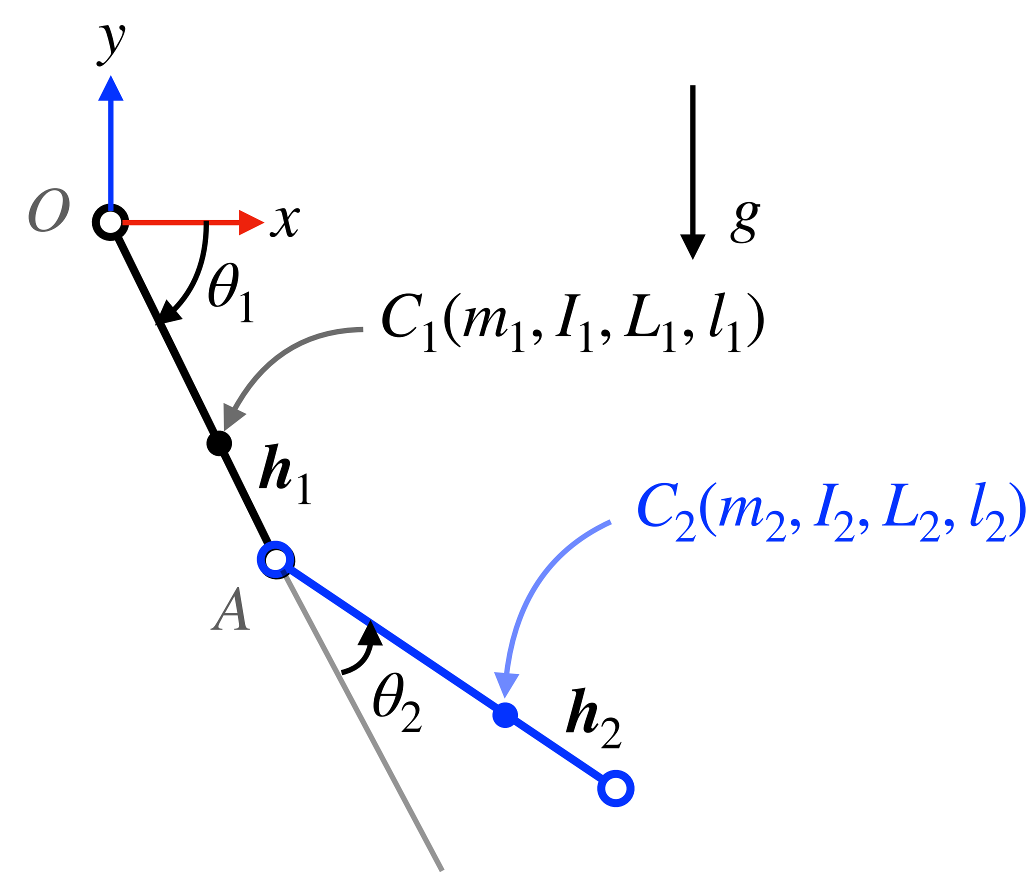
$$\boldsymbol{c}_1 = \begin{bmatrix} l_1 \cos\theta_1 \\ l_1 \sin\theta_1 \end{bmatrix}$$

$$\dot{\boldsymbol{c}}_1 = \begin{bmatrix} -l_1\dot{\theta}_1 \sin\theta_1 \\ l_1\dot{\theta}_1 \cos\theta_1 \end{bmatrix}$$

$$\ddot{\boldsymbol{c}}_1 = \begin{bmatrix} -l_1\left(\dot{\theta}_1^2 \cos\theta_1 + \ddot{\theta}_1 \sin\theta_1\right) \\ -l_1\left(\dot{\theta}_1^2 \sin\theta_1 - \ddot{\theta}_1 \cos\theta_1\right) \end{bmatrix}$$

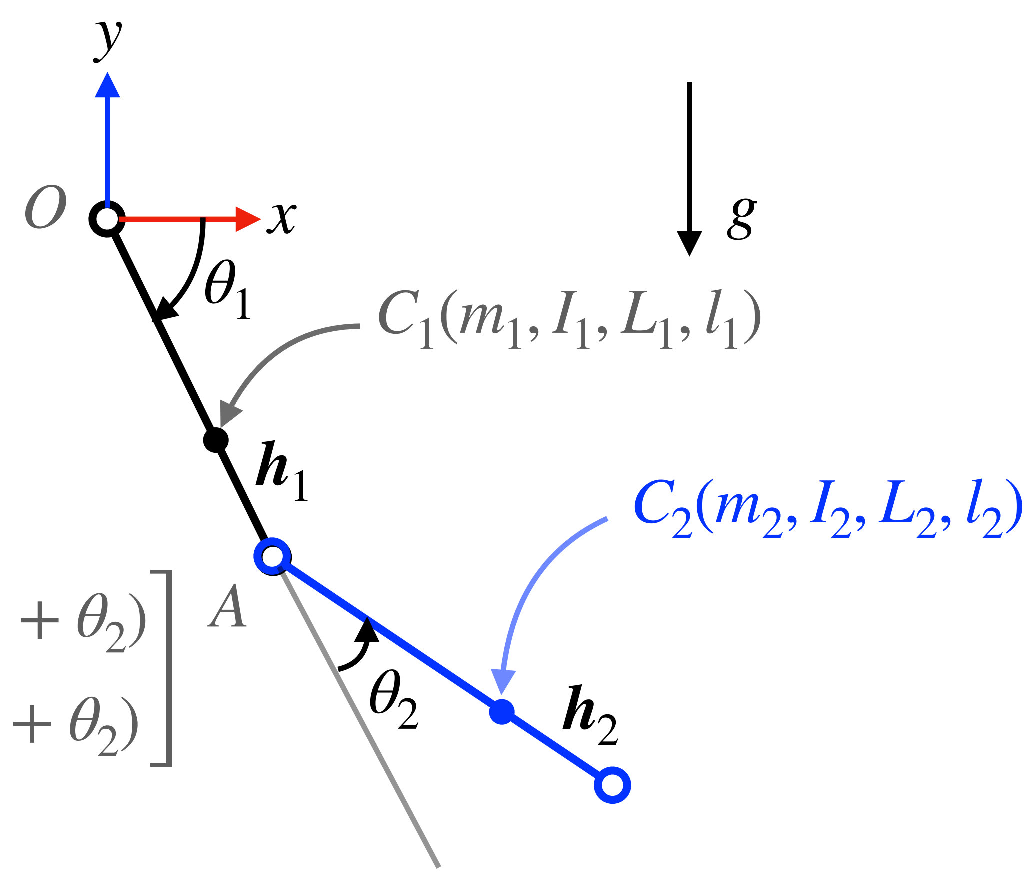
$$\boldsymbol{\omega}_1 = \dot{\theta}_1$$

$$\boldsymbol{\varepsilon}_1 = \dot{\boldsymbol{\omega}}_1 = \ddot{\boldsymbol{\theta}}_1$$



解：

① **运动学向外迭代：连杆2**



$$\boldsymbol{c}_2 = \begin{bmatrix} L_1 \cos\theta_1 + l_2 \cos\left(\theta_1 + \theta_{12}\right) \\ L_1 \sin\theta_1 + l_2 \sin\left(\theta_1 + \theta_{12}\right) \end{bmatrix}$$

$$\dot{\boldsymbol{c}}_2 = \begin{bmatrix} -\dot{\theta}_1(l_2 \sin(\theta_1 + \theta_2) + L_1 \sin(\theta_1)) - l_2\dot{\theta}_2 \sin(\theta_1 + \theta_2) \\ \dot{\theta}_1(l_2 \cos(\theta_1 + \theta_2) + L_1 \cos(\theta_1)) + l_2\dot{\theta}_2 \cos(\theta_1 + \theta_2) \end{bmatrix}$$

$$\ddot{\boldsymbol{c}}_2 = \begin{bmatrix} -l_2\dot{\theta}_1^2 \cos\left(\theta_1 + \theta_2\right) - l_2\dot{\theta}_2^2 \cos\left(\theta_1 + \theta_2\right) - L_1\dot{\theta}^2 \cos\left(\theta_1\right) - \ddot{\theta}_1 l_2 \sin\left(\theta_1 + \theta_2\right) - \ddot{\theta}_2 l_2 \sin\left(\theta_1 + \theta_2\right) - L_1\ddot{\theta}_1 \sin\left(\theta_1\right) - 2l_2\dot{\theta}_1\dot{\theta}_2 \cos\left(\theta_1 + \theta_2\right) \\ \ddot{\theta}_1 l_2 \cos\left(\theta_1 + \theta_2\right) - l_2\dot{\theta}_2^2 \sin\left(\theta_1 + \theta_2\right) - L_1\dot{\theta}_1^2 \sin\left(\theta_1\right) - l_2\dot{\theta}_1^2 \sin\left(\theta_1 + \theta_2\right) + \ddot{\theta}_2 l_2 \cos\left(\theta_1 + \theta_2\right) + L_1\ddot{\theta}_1 \cos\left(\theta_1\right) - 2l_2\dot{\theta}_1\dot{\theta}_2 \sin\left(\theta_1 + \theta_2\right) \end{bmatrix}$$

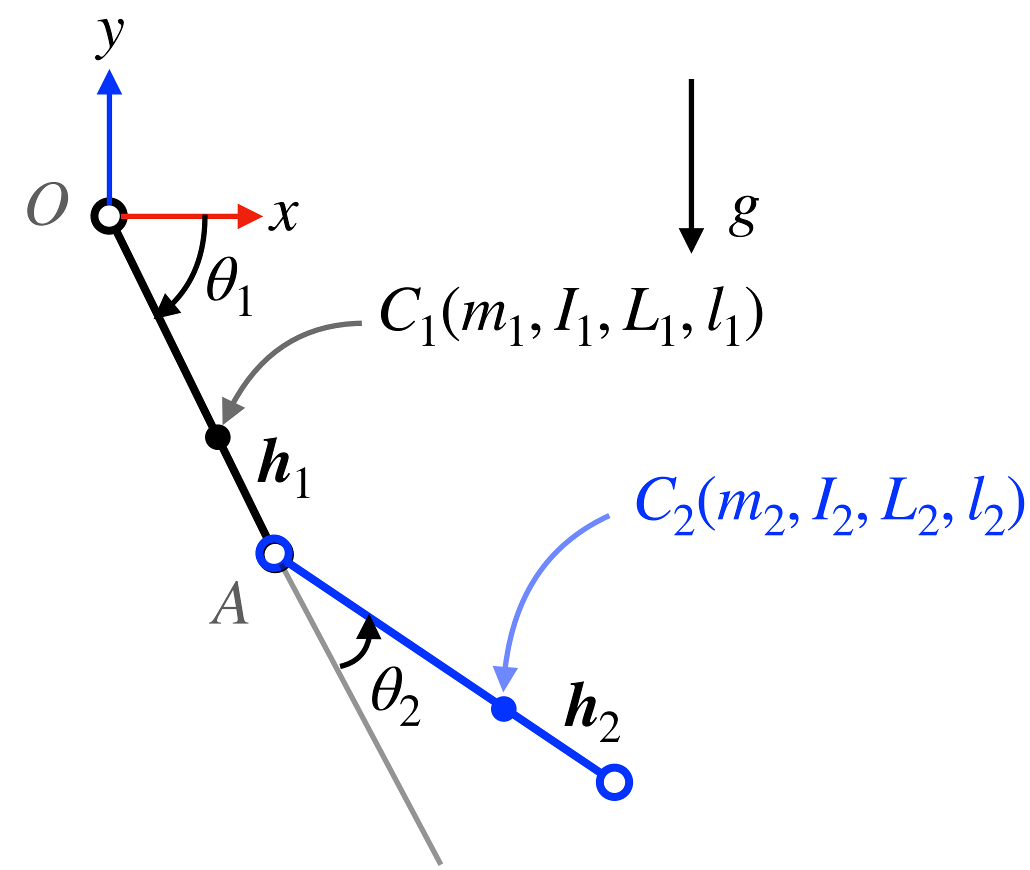
$$\boldsymbol{\omega}_{12} = \dot{\theta}_2 \;\Rightarrow \boldsymbol{\omega}_2 = \omega_1 + \omega_{12} = \dot{\theta}_1 + \dot{\theta}_2$$

$$\boldsymbol{\varepsilon}_2 = \dot{\boldsymbol{\omega}}_2 = (\ddot{\boldsymbol{\theta}}_1 + \ddot{\boldsymbol{\theta}}_2)$$

② 受力向内迭代：

连杆L_2的牛顿—欧拉方程为：

$$\begin{cases} \boldsymbol{f}_{1,2}+\boldsymbol{f}_2 = m_2\ddot{\boldsymbol{c}}_2 \\ \boldsymbol{t}_{1,2}+\boldsymbol{l}_2\times\boldsymbol{f}_{1,2} = \boldsymbol{I}_{C2}\boldsymbol{\varepsilon}_2 \end{cases}$$ （该例子中$\boldsymbol{t}_2=0$）



连杆L_1的牛顿—欧拉方程为：

$$\begin{cases} \boldsymbol{f}_{0,1}-\boldsymbol{f}_{1,2}+\boldsymbol{f}_1 = m_1\ddot{\boldsymbol{c}}_1 \\ \boldsymbol{t}_{0,1}-\boldsymbol{t}_{1,2}+\boldsymbol{l}_1\times\boldsymbol{f}_{0,1}-\boldsymbol{h}_1\times\boldsymbol{f}_{1,2} = \boldsymbol{I}_{C1}\boldsymbol{\varepsilon}_1 \end{cases}$$

其中：$\boldsymbol{f}_1=\begin{bmatrix}0 & -m_1 g\end{bmatrix}^T$，$\boldsymbol{f}_2=\begin{bmatrix}0 & -m_2 g\end{bmatrix}^T$，

由以上几式消去杆件间作用力，可解得：

$$\begin{aligned} \boldsymbol{t}_{1,2} &= \boldsymbol{I}_{C2}\cdot\boldsymbol{\varepsilon}_2 - m_2\boldsymbol{l}_2\times\left(\ddot{\boldsymbol{c}}_2-\boldsymbol{g}\right) \\ \boldsymbol{t}_{0,1} &= \boldsymbol{I}_{C1}\cdot\boldsymbol{\varepsilon}_1 - \boldsymbol{l}_1\times\left(m_1\ddot{\boldsymbol{c}}_1-m_1\boldsymbol{g}+m_2\ddot{\boldsymbol{c}}_2-m_2\boldsymbol{g}\right)-\boldsymbol{h}_1\times\left(m_2\ddot{\boldsymbol{c}}_2-m_2\boldsymbol{g}\right)+\boldsymbol{t}_{1,2} \end{aligned} \quad (*)$$

其中$\boldsymbol{g}=\begin{bmatrix}0 & -g\end{bmatrix}^T$

将$\boldsymbol{h}_1 = \begin{bmatrix} (L_1 - l_1) \sin\theta_1 \\ (L_1 - l_1) \cos\theta_1 \end{bmatrix}$, $\boldsymbol{l}_1 = \begin{bmatrix} l_1 \sin\theta_1 \\ l_1 \cos\theta_1 \end{bmatrix}$, $\boldsymbol{l}_2 = \begin{bmatrix} l_2 \sin(\theta_1 + \theta_2) \\ l_2 \cos(\theta_1 + \theta_2) \end{bmatrix}$

带入*式得:

$$\begin{aligned}\tau_2 &= (I_2 + m_2 l_2^2)(\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 l_2 L_1 \ddot{\theta}_1 \cos(\theta_2) \\ &+ m_2 l_2 L_1 \dot{\theta}_1^2 \sin(\theta_2) + m_2 l_2 g \sin(\theta_1 + \theta_2)\end{aligned}$$

$$\begin{aligned}\tau_1 &= (I_2 + m_2 l_2^2 + m_2 L_1 l_2 \cos(\theta_{12})(\ddot{\theta}_1 + \ddot{\theta}_{12}) \\ &+ (I_1 + m_2 L_1^2 + m_1 l_1^2)\ddot{\theta}_1 \\ &+ m_2 L_2 l_1 \cos(\theta_{12})\, \ddot{\theta}_1 + m_2 L_2 l_1 \sin(\theta_{12})\, \dot{\theta}_1^2 \\ &+ m_2 L_1 g \sin(\theta_1) + m_2 l_2 g \sin(\theta_1 + \theta_{12}) \\ &- m_2 L_1 l_2 \sin(\theta_{12})(\dot{\theta}_1 + \dot{\theta}_{12})^2\end{aligned}$$

整理后得:

$$\boldsymbol{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\tau}$$

$$\boldsymbol{H(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau}$$

$$H = \begin{bmatrix} m_2 L_1^2 + 2m_2 \cos\left(\theta_2\right) L_1 l_2 + m_1 l_1^2 + m_2 l_2^2 + I_1 + I_2 & m_2 l_2^2 + L_1 m_2 \cos\left(\theta_2\right) l_2 + I_2 \\ m_2 l_2^2 + L_1 m_2 \cos\left(\theta_{12}\right) l_2 + I_2 & m_2 l_2^2 + I_2 \end{bmatrix}$$

$$\boldsymbol{C} = \begin{bmatrix} -L_1 l_2 m_2 \dot{\theta}_2 \sin\left(\theta_2\right)\left(2\dot{\theta}_1 + \dot{\theta}_2\right) \\ L_1 l_2 m_2 \omega_1^2 \sin\left(\theta_2\right) \end{bmatrix}$$

$$\boldsymbol{G} = \begin{bmatrix} g\left(L_1 m_2 \sin\left(\theta_1\right) + l_1 m_1 \sin\left(\theta_1\right) + l_2 m_2 \sin\left(\theta_1 + \theta_2\right)\right) \\ g l_2 m_2 \sin\left(\theta_1 + \theta_2\right) \end{bmatrix}$$

本章提纲

- 概述
- 牛顿-欧拉方程（2学时）
- 拉格朗日方程（2学时）
  - 少自由度机器人的拉格朗日方程；
  - 多自由度机器人的拉格朗日方程；
- 投影牛顿欧拉法（PNE）（2学时）
- 案例分析（2学时）

机器人拉格朗日动力学方程

简单系统，采用拉格朗日方程法相较于采用牛顿—欧拉方程稍显复杂；随着系统复杂程度的增加，拉格朗日方程法就变得相对简单。

机器人系统的拉格朗日方程为：

$$Q_i = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} \quad i = 1,2,\ldots n$$

其中：

① n—系统的广义坐标数

② q_i—第i个广义坐标

③ $\dot{q}_i$—第i个广义速度

④ Q_i—作用在第i个广义坐标上的广义力或广义力矩

⑤ L—拉格朗日函数为系统的动能K和位能P之差，即：$L = K - P$

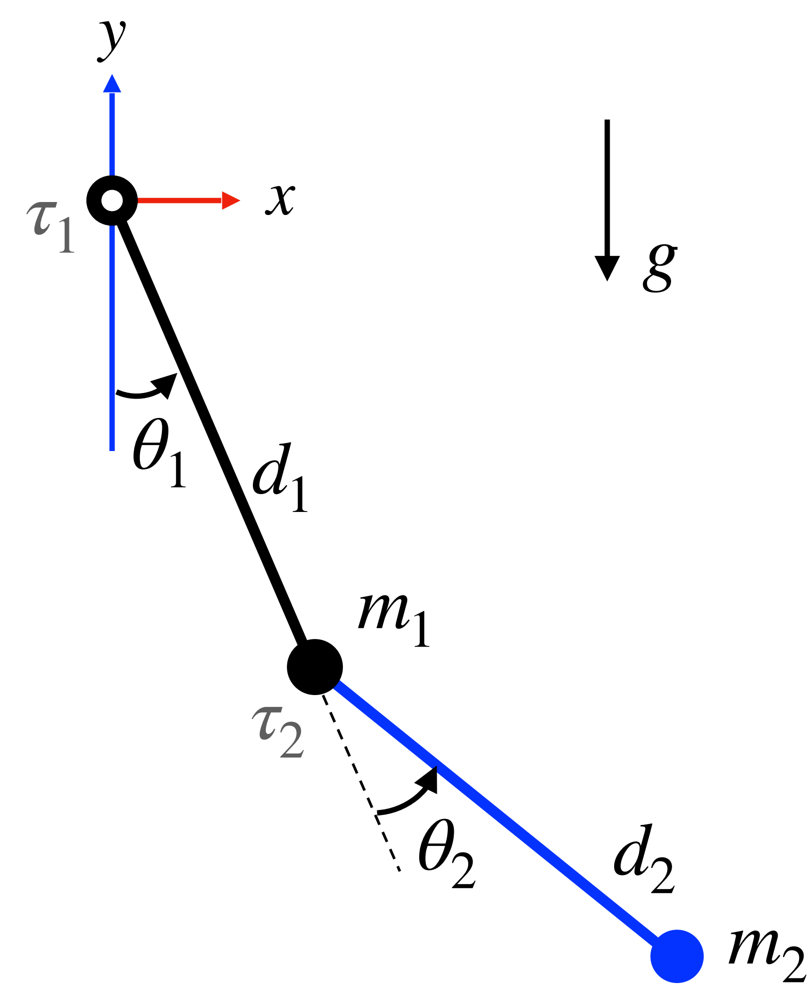
少自由度机器人例题：

如图所示的两连杆的机器人：

① 两个连杆的质量分别为m_1、m_2，且位于连杆的端部，两个连杆的长度分别为d_1、d_2；

② 机器人所在的竖直平面内存在加速度为g的重力场，连杆1与重力方向的夹角为θ_1，连杆2与连杆的夹角为θ_2；

③ 连杆1的驱动力矩为τ_1，连杆2的驱动力矩为τ_2；



动能和势能 (The Kinetic and Potential Energy)

质量m_1的动能可直接写出:

$$K_1 = \frac{1}{2} m_1 {v_1}^2 = \frac{1}{2} m_1 d_1^2 \dot{\theta}_1^2$$

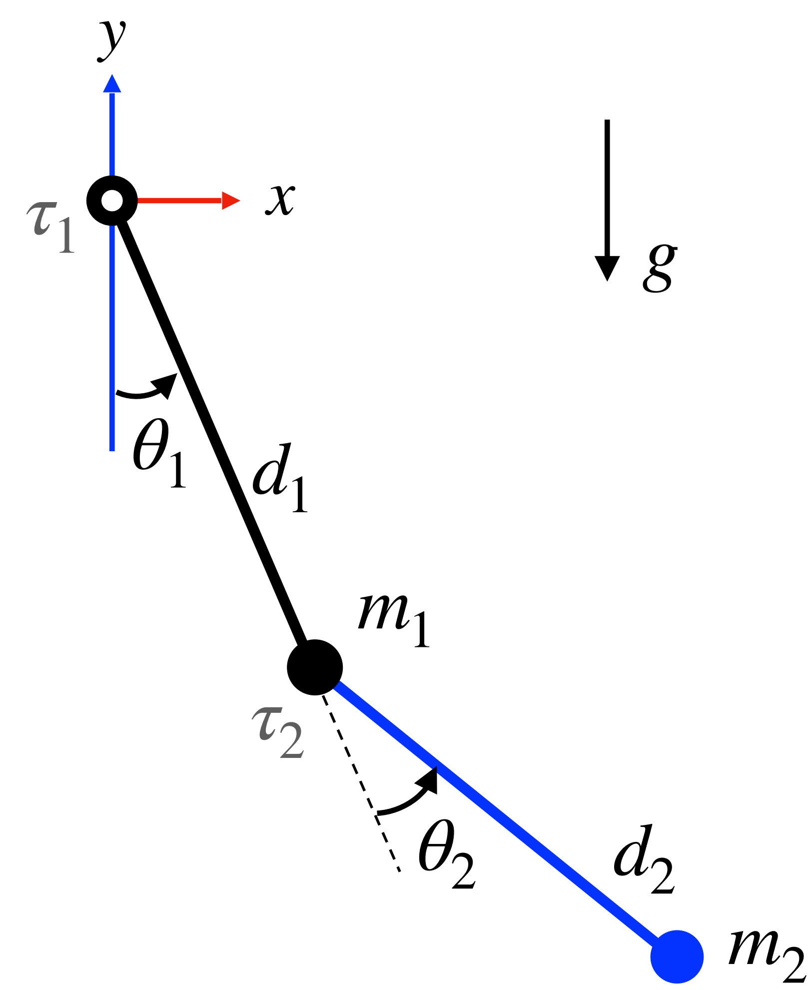
质量m_1势能与其坐标y有关，也可以直接写出：

$$P_1 = -m_1 g d_1 \cos\left(\theta_1\right)$$

质量m_2的直角坐标位置表达式为：

$$x_2 = d_1 \sin\left(\theta_1\right) + d_2 \sin\left(\theta_1 + \theta_2\right)$$

$$y_2 = -d_1 \cos\left(\theta_1\right) - d_2 \cos\left(\theta_1 + \theta_2\right)$$



然后求微分后得到m_2在基坐标系下的速度：

$$\dot{x}_2 = d_1 \cos\left(\theta_1\right)\dot{\theta}_1 + d_2 \cos\left(\theta_1 + \theta_2\right)\left(\dot{\theta}_1 + \dot{\theta}_2\right)$$

$$\dot{y}_2 = d_1 \sin\left(\theta_1\right)\dot{\theta}_1 + d_2 \sin\left(\theta_1 + \theta_2\right)\left(\dot{\theta}_1 + \dot{\theta}_2\right)$$

则其速度平方的值为：

$$
\begin{aligned}
V_2^2 = {} & d_1^2\dot{\theta}_1^2 + d_1^2\left(\ddot{\theta}_1^2 + 2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2\right) \\
& + 2d_1d_2\cos\left(\theta_1\right)\cos\left(\theta_1+\theta_2\right)\left(\dot{\theta}_1^2 + \dot{\theta}_1\dot{\theta}_2\right) \\
& + 2d_1d_2\sin\left(\theta_1\right)\sin\left(\theta_1+\theta_2\right)\left(\dot{\theta}_1^2 + \dot{\theta}_1\dot{\theta}_2\right) \\
= {} & d_1^2\dot{\theta}_1^2 + d_1^2\left(\dot{\theta}_1^2 + 2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2\right) + 2d_1d_2\cos\left(\theta_2\right)\left(\dot{\theta}_1^2 + \dot{\theta}_1\dot{\theta}_2\right)
\end{aligned}
$$

从而m_2的动能为：

$$
K_2 = \frac{1}{2}m_2d_1^2\dot{\theta}_1^2 + \frac{1}{2}m_2d_2^2\left(\dot{\theta}_1^2 + 2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2\right) + m_2d_1d_2\,\mathrm{Cos}\left(\theta_2\right)\left(\theta_1^2 + \dot{\theta}_1\dot{\theta}_2\right)
$$

m_2的势能为：

$$
P_2 = -m_2gd_1\cos\left(\theta_1\right) - m_2gd_2\cos\left(\theta_1+\theta_2\right)
$$

系统的拉格朗日算子 (The Lagrangian)：

$$L = \frac{1}{2}\left(m_1 + m_2\right) d_1^2 \dot{\theta}_1^2 + \frac{1}{2} m_2 d_2^2 \left(\dot{\theta}_1{}^2 + 2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2\right)$$

$$+ m_2 d_1 d_2 \cos\left(\theta_2\right) \left(\dot{\theta}_1{}^2 + \dot{\theta}_1 \dot{\theta}_2\right)$$

$$+ \left(m_1 + m_2\right) g d_1 \cos\left(\theta_1\right) + m_2 g d_2 \cos\left(\theta_1 + \theta_2\right)$$

对拉格朗日算子进行微分：

$$\frac{\partial L}{\partial \dot{\theta}_1} = \left(m_1 + m_2\right) d_1^2 \dot{\theta}_1 + m_2 d_2^2 \dot{\theta}_1^2 + m_2 d_2^2 \dot{\theta}_2^2$$

$$+ 2 m_2 d_1 d_2 \cos\left(\theta_2\right) \dot{\theta}_1 + m_2 d_1 d_2 \cos\left(\theta_2\right) \dot{\theta}_2$$

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_1} = \left[\left(m_1 + m_2\right) d_1^2 + m_2 d_2^2 + 2 m_2 d_1 d_2 \cos\left(\theta_2\right)\right] \ddot{\theta}_1$$

$$+ \left[m_2 d_2^2 + m_2 d_1 d_2 \cos\left(\theta_2\right)\right] \ddot{\theta}_2$$

$$- 2 m_2 d_1 d_2 \sin\left(\theta_2\right) \dot{\theta}_1 \dot{\theta}_2 - m_2 d_1 d_2 \sin\left(\theta_2\right) \dot{\theta}_2^2$$

$$\frac{\partial L}{\partial \theta_1} = - \left(m_1 + m_2\right) g d_1 \, \mathrm{Sin}\left(\theta_1\right) - m_2 g d_2 \, \mathrm{Sin}\left(\theta_1 + \theta_2\right)$$

根据拉格朗日方程关节1的驱动力矩应为：

$$
\begin{aligned}
\tau_1 = & \left[\left(m_1+m_2\right) d_1^2+m_2 d_2^2+2 m_2 d_1 d_2 \cos \left(\theta_2\right)\right] \ddot{\theta}_1 \\
& +\left[m_2 d_2^2+m_2 d_1 d_2 \cos \left(\theta_2\right)\right] \ddot{\theta}_2 \\
& -2 m_2 d_1 d_2 \sin \left(\theta_2\right) \dot{\theta}_1 \dot{\theta}_2-m_2 d_1 d_2 \sin \left(\theta_2\right) \dot{\theta}_2^2 \\
& +\left(m_1+m_2\right) g d_1 \sin \left(\theta_1\right)-m_2 g d_2 \sin \left(\theta_1+\theta_2\right)
\end{aligned}
$$

用拉格朗日算子对θ_2和$\dot{\theta}_2$求偏微分，进而得到连杆2的力矩方程

$$\frac{\partial L}{\partial \dot{\theta}_2} = m_2 d_2^2 \dot{\theta}_1 + m_2 d_2^2 \dot{\theta}_2 + m_2 d_1 d_2 \operatorname{Cos}\left(\theta_2\right) \dot{\theta}_1$$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}_2} = m_2 d_2^2 \ddot{\theta}_1 + m_2 d_2^2 \ddot{\theta}_2 + m_2 d_1 d_2 \cos(\theta_2) \ddot{\theta}_1$$
$$-m_2 d_1 d_2 \sin(\theta_2) \dot{\theta}_1 \dot{\theta}_2$$

$$\frac{\partial L}{\partial \theta_2} = -m_2 d_1 d_2 \sin\left(\theta_2\right)\left(\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2\right) - m_2 g d_2 \sin\left(\theta_1 + \theta_2\right)$$

根据拉格朗日方程关节2的驱动力矩应为：

$$\tau_2 = \left[m_2 d_2^2 + m_2 d_1 d_2 \cos\left(\theta_2\right)\right] \ddot{\theta}_1 + m_2 d_2^2 \dot{\theta}_2$$
$$-2 m_2 d_1 d_2 \sin\left(\theta_2\right) \dot{\theta}_1 \dot{\theta}_2 - m_2 d_1 d_2 \sin\left(\theta_2\right) \dot{\theta}_1^2$$
$$+m_2 g d_2 \sin\left(\theta_1 + \theta_2\right)$$

将结果按关节驱动力矩整理后得：

$$\tau_1 = D_{11}\ddot{\theta}_1 + D_{12}\ddot{\theta}_2 + D_{111}\dot{\theta}_1^2 + D_{122}\dot{\theta}_2^2 + D_{112}\dot{\theta}_1\dot{\theta}_2 + D_{121}\dot{\theta}_2\dot{\theta}_1 + D_1$$
$$\tau_2 = D_{21}\ddot{\theta}_1 + D_{22}\ddot{\theta}_2 + D_{211}\dot{\theta}_1^2 + D_{222}\dot{\theta}_2^2 + D_{212}\dot{\theta}_1\dot{\theta}_2 + D_{221}\dot{\theta}_2\dot{\theta}_1 + D_2$$

其中：

D_{ii}– 关节i的等效惯量，关节i的加速度所需要关节i提供的力矩为 $D_{ii}\ddot{\theta}_i$；

D_{ij}– 关节i与关节j之间的耦合惯量，关节i或关节j的加速度分别使关节j或i产生的力矩 $D_{ij}\ddot{\theta}_i$ 和 $D_{ij}\ddot{\theta}_j$；

D_{ijj}– 由关节j的速度产生的作用在关节i上的向心力 $D_{ijj}\dot{\theta}_j^2$ 系数；

D_{ijk}/D_{ikj}– 作用在关节i上的复合向心力（哥氏力）的组合项 $\left(D_{ijk}\dot{\theta}_j\dot{\theta}_k + D_{ikj}\dot{\theta}_k\dot{\theta}_j\right)$系数，这是关节$j$和关节$k$的速度产生的结果;

D_i– 作用在关节i上的重力；

具体各项系数为：

等效惯量: $$\begin{cases} D_{11} = \left[\left(m_1 + m_2 \right) d_1^2 + m_2 d_2^2 + 2 m_2 d_1 d_2 \cos\left(\theta_2 \right) \right] \\ D_{22} = m_2 d_2^2 \end{cases}$$

耦合惯量: $$D_{12} = m_2 d_2^2 + m_2 d_1 d_2 \cos\left(\theta_2 \right)$$

向心加速度系数: $$\begin{cases} D_{111} = 0 \\ D_{122} = - m_2 d_1 d_2 \sin\left(\theta_2 \right) \\ D_{211} = m_2 d_1 d_2 \sin\left(\theta_2 \right) \\ D_{222} = 0 \end{cases}$$

哥氏加速度系数: $$\begin{cases} D_{112} = D_{121} = - m_2 d_1 d_2 \sin\left(\theta_2 \right) \\ D_{212} = D_{221} = 0 \end{cases}$$

重力项为: $$\begin{cases} D_1 = (m_1 + m_2) g d_1 \sin(\theta 1) + m_2 g d_2 \sin(\theta 1 + \theta 2) \\ D_2 = m_2 g d_2 \sin(\theta 1 + \theta 2) \end{cases}$$

多自由度机器人的拉格朗日方程

机器人系统的拉格朗日方程为：

$$Q_i = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} \quad i = 1,2,\ldots n$$

其中：

① n—系统的广义坐标数

② q_i—第i个广义坐标

③ $\dot{q}_i$—第i个广义速度

④ Q_i—作用在第i个广义坐标上的广义力或广义力矩

⑤ L—拉格朗日函数为系统的动能K和位能P之差，即：$L = K - P$

多自由度机器人的拉格朗日方程

多自由度机器人的拉格朗日动力学方程的求取问题转化为：

已知机器人的运动学，求机器人系统的拉格朗日算子$L(\boldsymbol{q}, \dot{\boldsymbol{q}})$，进一步转化为：

① 连杆i的动能计算；

② 连杆i的位能计算；

这些计算应该依赖于位置运动学和速度(微分)运动学中计算手段。

连杆i上任意质点元dm_i在与连杆i固联的坐标系Σ_i中的齐次坐标为${}^i\boldsymbol{r}$，在基坐标系Σ_0中齐次坐标为${}^0\boldsymbol{r}$。因此有${}^0\boldsymbol{r} = {}^0_i\boldsymbol{T}\,{}^i\boldsymbol{r}$，为了简介起见记${}^0\boldsymbol{r}$为$\boldsymbol{r}$，${}^0_i\boldsymbol{T}$为${}_i\boldsymbol{T}$，故写成：

$$\boldsymbol{r} = {}_i\boldsymbol{T}\,{}^i\boldsymbol{r}$$

它表示dm_i在Σ_0中的位置，由此dm_i在Σ_0中的速度为：

$$\dot{\boldsymbol{r}} = {}_i\dot{\boldsymbol{T}}\,{}^i\boldsymbol{r} = \left(\sum_{j=1}^{i} \left(\frac{\partial\, {}_i\boldsymbol{T}}{\partial q_j}\dot{q}_j\right) \right) {}^i\boldsymbol{r}$$

建立拉格朗日方程的步骤1：系统动能

该质点元dm_i的动能为：

$$
\begin{aligned}
dK_i &= \frac{1}{2}\,\mathrm{Trac}\left(\dot{\boldsymbol{r}}\dot{\boldsymbol{r}}^T\right) dm_i \\
&= \frac{1}{2}\,\mathrm{Trac}\left[\left(\sum_{j=1}^{i}\frac{\partial_i \boldsymbol{T}}{\partial q_j}\dot{q}_j\right){}^i\boldsymbol{r}\cdot{}^i\boldsymbol{r}^T\left(\sum_{k=1}^{i}\frac{\partial_i \boldsymbol{T}}{\partial q_k}\dot{q}_k\right)^T\right] dm_i \\
&= \frac{1}{2}\,\mathrm{Trac}\left[\sum_{j=1}^{i}\sum_{k=1}^{i}\left(\frac{\partial_i \boldsymbol{T}}{\partial \boldsymbol{q}_j}{}^i\boldsymbol{r}\cdot{}^i\boldsymbol{r}^T\frac{\partial_i \boldsymbol{T}^T}{\partial \boldsymbol{q}_k}\,\dot{q}_j\dot{q}_k\right)\right] dm_i
\end{aligned}
$$

连杆i的动能为：

$$K_i = \int_{L_i} dK_i$$

$$= \frac{1}{2} \int_{Li} \mathrm{Trac} \left[\sum_{j=1}^{i} \sum_{k=1}^{i} \left(\frac{\partial_i \boldsymbol{T}}{\partial \boldsymbol{q}_j} {}^i\boldsymbol{r} \cdot {}^i\boldsymbol{r}^T \frac{\partial_i \boldsymbol{T}^T}{\partial \boldsymbol{q}_k} \dot{q}_j \dot{q}_k \right) \right] dm_i$$

$$= \frac{1}{2} \mathrm{Trac} \sum_{j=1}^{i} \sum_{k=1}^{i} \left(\frac{\partial_i \boldsymbol{T}}{\partial \boldsymbol{q}_j} \left[\int_{Li} \left({}^i\boldsymbol{r} \cdot {}^i\boldsymbol{r}^T \right) dm_i \right] \frac{\partial_i \boldsymbol{T}^T}{\partial \boldsymbol{q}_k} \right) \dot{q}_j \dot{q}_k$$

$$= \frac{1}{2} \mathrm{Trac} \sum_{j=1}^{i} \sum_{k=1}^{i} \left(\frac{\partial_i \boldsymbol{T}}{\partial \boldsymbol{q}_j} \boldsymbol{J}_i \frac{\partial_i \boldsymbol{T}^T}{\partial \boldsymbol{q}_k} \right) \dot{q}_j \dot{q}_k$$

其中： $\boldsymbol{J}_i = \int_{Li} \left({}^i\boldsymbol{r} \cdot {}^i\boldsymbol{r}^T \right) dm_i$**是一常数阵，称刚体i的伪惯性矩阵，具体计算为：**

$$
\boldsymbol{J}_i = \int_{L1} \left({}^{i}\boldsymbol{r} \cdot {}^{i}\boldsymbol{r}^{T}\right) dm_i = \int_{L1} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \begin{bmatrix} x\, y\, z\, 1 \end{bmatrix} dm_i
$$

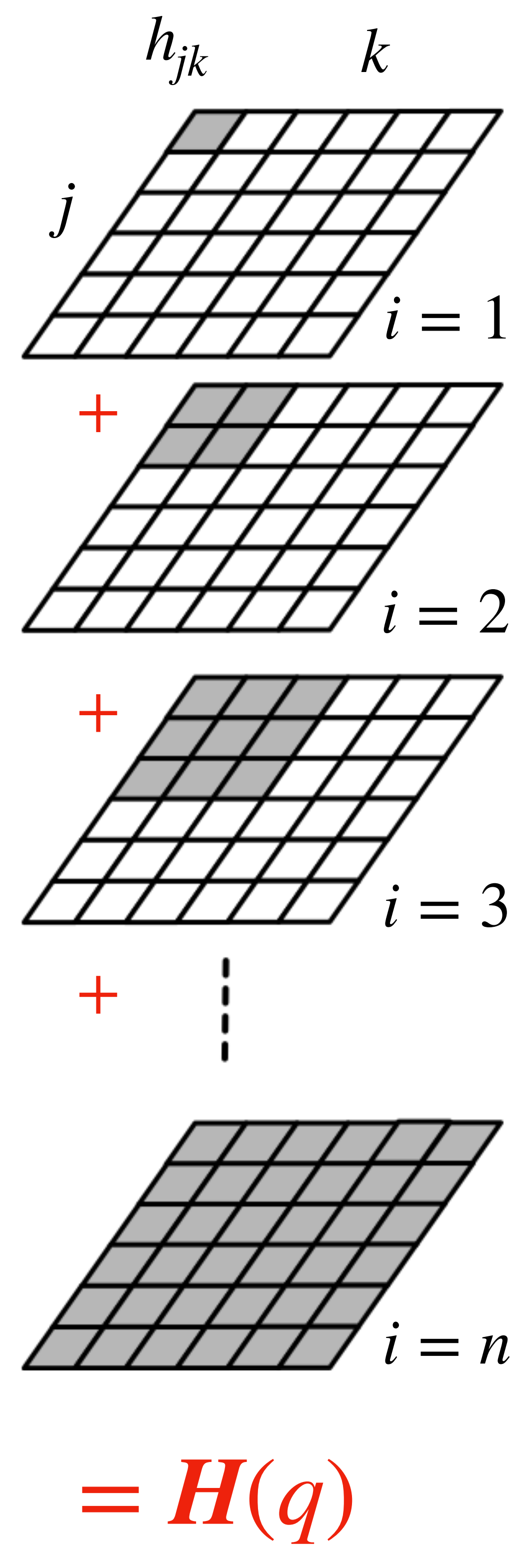
$$
= \begin{bmatrix} \int_{L1} x^2 dm_i & \int_{L1} xydm_i & \int_{L1} xzdm_i & \int_{L1} xdm_i \\ \int_{L1} xydm_i & \int_{L1} y^2 dm_i & \int_{L1} yzdm_i & \int_{L1} ydm_i \\ \int_{L1} zxdm_i & \int_{L1} zydm_i & \int_{L1} z^2 dm_i & \int_{L1} zdm_i \\ \int_{L1} xdm_i & \int_{L1} ydm_i & \int_{L1} zdm_i & \int_{L1} dm_i \end{bmatrix}
$$

$$
= \begin{bmatrix} \dfrac{-I_x + I_y + I_z}{2} & I_{xy} & I_{xz} & m_i x_{c_i} \\ I_{xy} & \dfrac{I_x - I_y + I_z}{2} & I_{yz} & m_i y_{c_i} \\ I_{xz} & I_{yz} & \dfrac{I_x + I_y - I_z}{2} & m_i z_{c_i} \\ m_i x_{c_i} & m_i y_{c_i} & m_i z_{c_i} & m_i \end{bmatrix}
$$

它完整地描述了杆件i的质量分布情况。

建立拉格朗日方程的步骤1：系统动能

$$
\begin{aligned}
K &= \sum_{i=1}^{n} K_i \\
&= \frac{1}{2}\sum_{i=1}^{n}\left[\mathrm{Trace}\sum_{j=1}^{i}\sum_{k=1}^{i}\left(\frac{\partial_i \boldsymbol{T}}{\partial q_j}\boldsymbol{J}_i\frac{\partial_i \boldsymbol{T}^T}{\partial q_k}\dot{q}_j\dot{q}_k\right)\right] \\
&= \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n}\left[\sum_{i=max(j,k)}^{n}\mathrm{Trace}\left(\frac{\partial_i \boldsymbol{T}}{\partial q_j}\boldsymbol{J}_i\frac{\partial_i \boldsymbol{T}^T}{\partial q_k}\right)\right]\dot{q}_j\dot{q}_k \\
&= \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n} h_{jk}\dot{q}_j\dot{q}_k \\
&= \frac{1}{2}\dot{q}^T\boldsymbol{H}(q)\dot{q}
\end{aligned}
$$



建立拉格朗日方程的步骤2：系统势能

$$P_i = \int\limits_{L_i} \boldsymbol{g}^T \left({}_i\boldsymbol{T}^i\boldsymbol{r} \right) dm_i = m_i \boldsymbol{g}^T \left({}_i\boldsymbol{T}^i\boldsymbol{r}_{ci} \right)$$

$$P = \sum_{i=1}^{n} P_i$$

$$= - \sum_{i=1}^{n} m_i \boldsymbol{g}^T {}_i\boldsymbol{T}^i\boldsymbol{r}_{ci}$$

建立拉格朗日方程的步骤3：计算拉格朗日算子

$$L = K - P$$

$$= \frac{1}{2}\dot{q}^T \boldsymbol{H}(q)\dot{q} + \sum_{i=1}^{n} m_i \boldsymbol{g}^T {}_i\boldsymbol{T}^i\boldsymbol{r}_{ci}$$

建立拉格朗日方程的步骤4：推导拉格朗日方程

$$\frac{\partial L}{\partial \dot{q}_j} = \frac{\partial K}{\partial \dot{q}_j} = \sum_{k=1}^{n} h_{jk}\dot{q}_k$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_j}\right) = \sum_{k=1}^{n} h_{jk}\ddot{q}_k + \sum_{k=1}^{n} \dot{h}_{jk}\dot{q}_k$$

$$\begin{aligned}\frac{\partial L}{\partial q_j} &= \frac{\partial K}{\partial q_j} - \frac{\partial P}{\partial q_j} = \frac{1}{2}\dot{\boldsymbol{q}}^T\frac{\partial H}{\partial q_j}\dot{\boldsymbol{q}} - \left(-\sum_{i=1}^{n} m_i \boldsymbol{g}^T \frac{\partial {}_i\boldsymbol{T}}{\partial q_j} {}^i\boldsymbol{r}_{ci}\right) \\ &= \frac{1}{2}\dot{\boldsymbol{q}}^T\frac{\partial H}{\partial q_j}\dot{\boldsymbol{q}} - g_j\end{aligned}$$

将上述三式带入到拉格朗日方程中得：

$$\sum_{k=1}^{n} h_{jk}\ddot{q}_k + \sum_{k=1}^{n} \dot{h}_{jk}\dot{q}_k - \frac{1}{2}\dot{\boldsymbol{q}}^T \frac{\partial H}{\partial q_j}\dot{\boldsymbol{q}} + g_j = \tau_j, (j = 1,2,\cdots,n)$$

其中:

$$\begin{aligned}
&\sum_{k=1}^{n} \dot{h}_{jk}\dot{q}_k - \frac{1}{2}\dot{\boldsymbol{q}}^T \frac{\partial H}{\partial q_j}\dot{\boldsymbol{q}} \\
&\quad = \sum_{k=1}^{n}\sum_{i=1}^{n} \frac{\partial h_{jk}}{\partial q_i}\dot{q}_k\dot{q}_i - \frac{1}{2}\sum_{k=1}^{n}\sum_{i=1}^{n} \frac{\partial h_{ki}}{\partial q_j}\dot{q}_k\dot{q}_i \\
&\quad = \sum_{k=1}^{n}\sum_{i=1}^{n} \left(\frac{\partial h_{jk}}{\partial q_j} - \frac{1}{2}\frac{\partial h_{ki}}{\partial q_j}\right)\dot{q}_k\dot{q}_i \\
&\quad = \dot{\boldsymbol{q}}^T \boldsymbol{C}_j \dot{\boldsymbol{q}}
\end{aligned}$$

$$\boldsymbol{C}_j = \left(\frac{\partial h_{jk}}{\partial q_j} - \frac{1}{2}\frac{\partial h_{ki}}{\partial q_j}\right)$$

拉格朗日方程可以写为：

$$\sum_{k=1}^{n} h_{jk}\ddot{q}_k + \dot{\boldsymbol{q}}^T \boldsymbol{C}_j \dot{\boldsymbol{q}} + g_j = \tau_j, \quad (j = 1,2,\cdots,n)$$

写成矩阵形式为：

$$\boldsymbol{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\tau}$$

式中：

$$\boldsymbol{H}(\boldsymbol{q}) = \left[h_{jk}\right],\ \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) = \begin{bmatrix} \dot{\boldsymbol{q}}^T\boldsymbol{C}_1 \\ \dot{\boldsymbol{q}}^T\boldsymbol{C}_2 \\ \vdots \\ \dot{\boldsymbol{q}}^T\boldsymbol{C}_n \end{bmatrix},\ \boldsymbol{G}(\boldsymbol{q}) = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{bmatrix},\ \ \boldsymbol{\tau} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \vdots \\ \tau_n \end{bmatrix}$$

本章提纲

- 概述
- 牛顿-欧拉方程（2学时）
- 拉格朗日方程（2学时）
- **投影牛顿欧拉法（PNE）（2学时）**
    - **投影牛顿欧拉法介绍**
    - **被动行走例子**
- 案例分析（2学时）

投影牛顿-欧拉法(PNE/JMJ)

无约束坐标：$\boldsymbol{x} = \left[x_1, x_2, \cdots, x_n\right]^T$

广义坐标：$\boldsymbol{q} = \left[q_1, q_2, \cdots, q_m\right]^T$

约束：

$$\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{q})$$

$$= \begin{bmatrix} x_1\left(q_1, q_2, \cdots, q_m\right) \\ x_2\left(q_1, q_2, \cdots, q_m\right) \\ \vdots \\ x_n\left(q_1, q_2, \cdots, q_m\right) \end{bmatrix}^T$$

投影牛顿-欧拉法(PNE/JMJ)

有约束速度：

$$\dot{\boldsymbol{x}} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}} = \boldsymbol{J}\dot{\boldsymbol{q}} = \begin{bmatrix} \frac{\partial x_1}{\partial q_1} & \frac{\partial x_1}{\partial q_2} & \cdots & \frac{\partial x_1}{\partial q_m} \\ \frac{\partial x_2}{\partial q_1} & \frac{\partial x_2}{\partial q_2} & \cdots & \frac{\partial x_2}{\partial q_m} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial x_n}{\partial q_1} & \frac{\partial x_n}{\partial q_2} & \cdots & \frac{\partial x_n}{\partial q_m} \end{bmatrix} \dot{\boldsymbol{q}}, \text{ 其中} \boldsymbol{J} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{q}}$$

有约束坐标：$\ddot{\boldsymbol{x}} = \boldsymbol{J}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}\dot{\boldsymbol{q}} = \boldsymbol{J}\ddot{\boldsymbol{q}} + \boldsymbol{D}$，其中$\boldsymbol{D} = \boldsymbol{D}(\boldsymbol{q}, \dot{\boldsymbol{q}})$

虚功虚位移原理：$\boldsymbol{J}^T \cdot \boldsymbol{F}^c = 0$

牛顿-欧拉方程：$\boldsymbol{M}\ddot{\boldsymbol{x}} = \boldsymbol{F}^c + \boldsymbol{F}^a$

投影牛顿-欧拉法(PNE/JMJ)

联立两个方程：

$$\begin{cases} MJ\ddot{q} + MD = F^c + F^a \\ J^T F^c = 0 \end{cases}$$

得：$J^T MJ\ddot{q} + J^T MD = J^T F^a$(逆动力学)

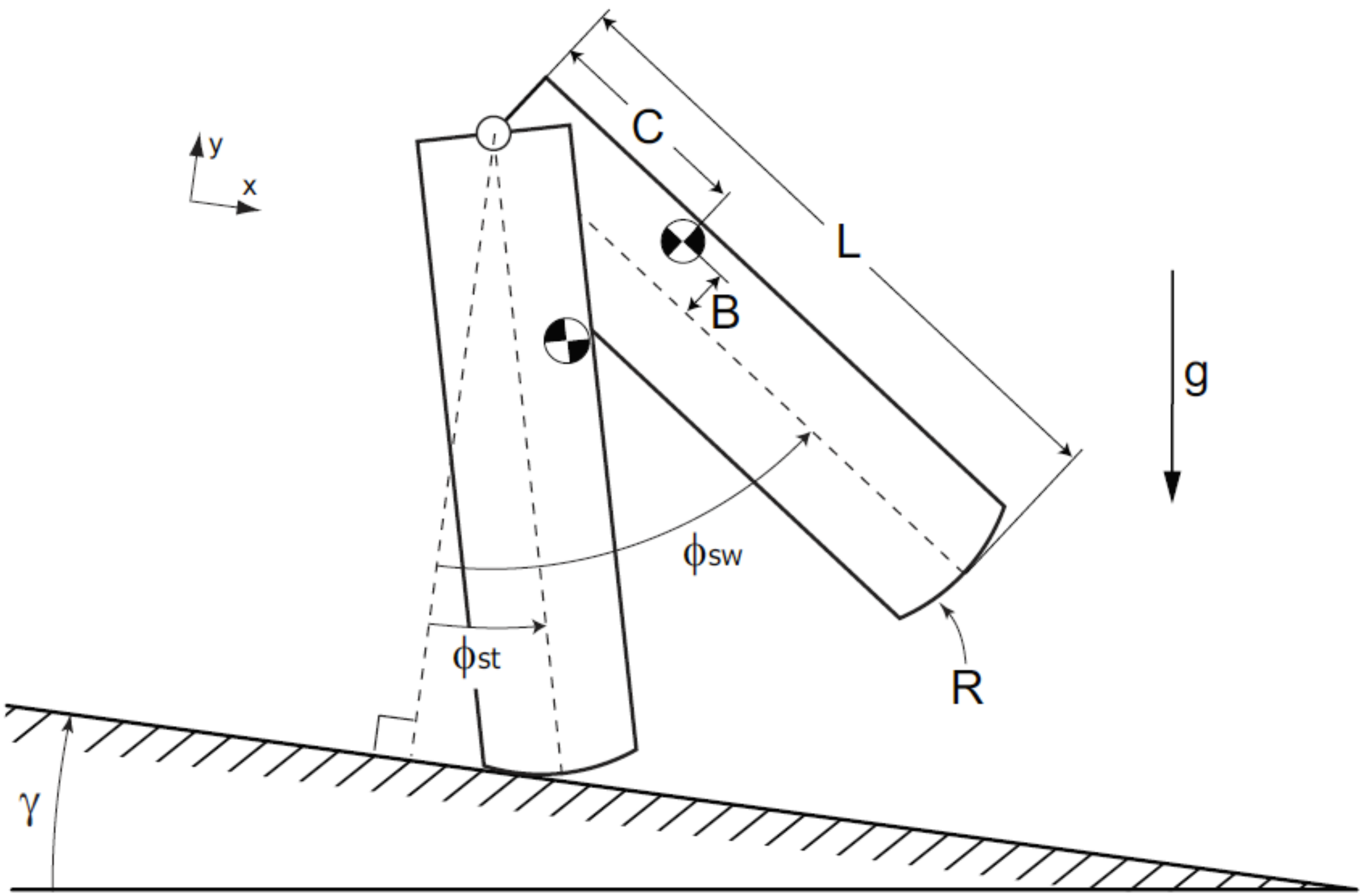
如果$J^T MJ$是非奇异的，则：

$$\ddot{q} = \left(J^T MJ\right)^{-1}\left(J^T F^a - J^T MD\right)$$(正动力学)

同时可以解得约束力：

$$F^c = MJ\ddot{q} - MD - F^a$$

投影牛顿-欧拉法(PNE/JMJ)



参数：

L －腿长

m－腿部质量

I_c－腿部质心转动惯量

B－质心水平偏移

C－质心竖直偏移

R－脚半径

γ－斜坡角度

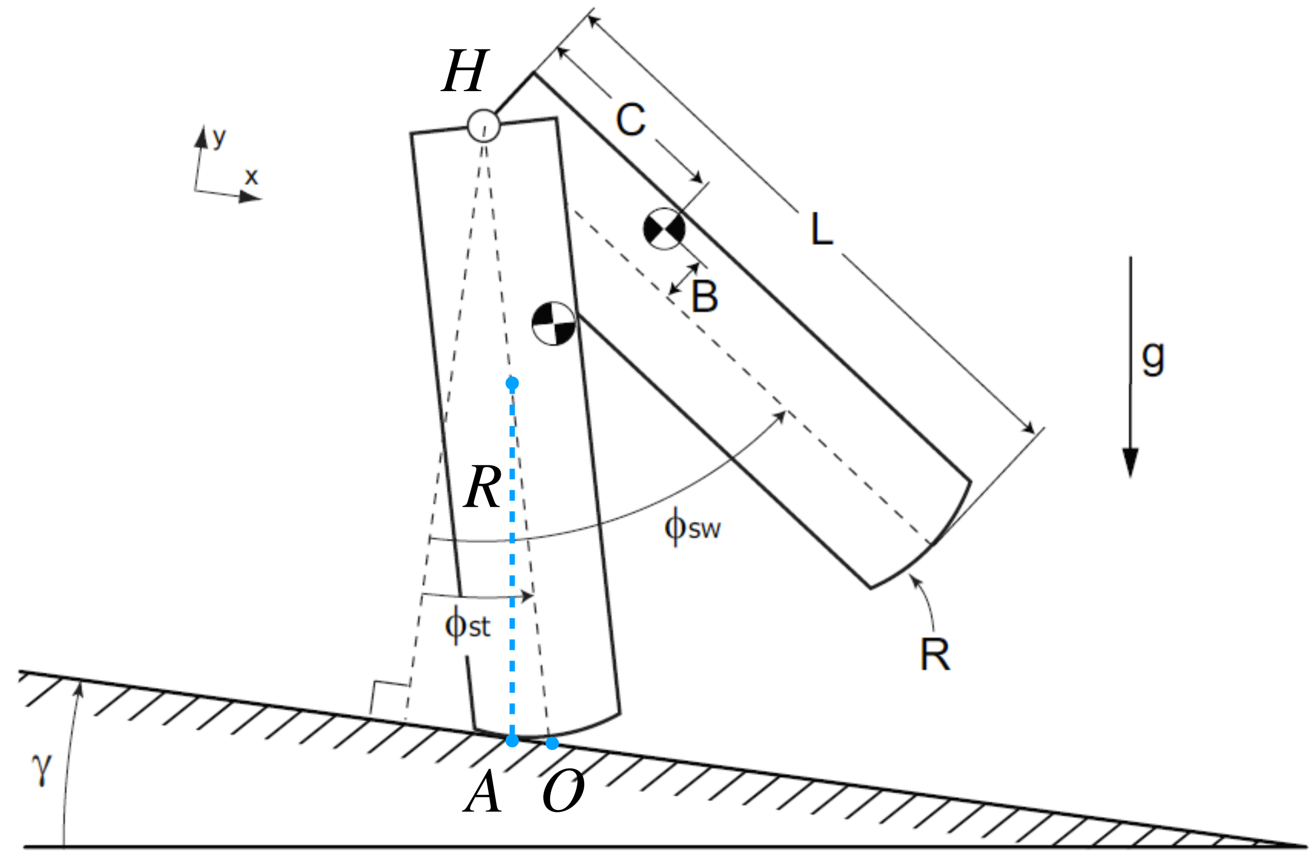
变量：

ϕ_{st}－支撑腿角度

ϕ_{sw}－摆动腿角度

投影牛顿-欧拉法(PNE/JMJ)

$$
\begin{cases}
\boldsymbol{P}_h = \begin{bmatrix} -R\phi_{st} - (L-R)\sin(\phi_{st}) \\ R + (L-R)\cos(\phi_{st}) \end{bmatrix} \\
\boldsymbol{R}_{st} = \begin{bmatrix} \cos(\phi_{st}) & -\sin(\phi_{st}) \\ \sin(\phi_{st}) & \cos(\phi_{st}) \end{bmatrix} \\
\boldsymbol{R}_{sw} = \begin{bmatrix} \cos(\phi_{sw}) & -\sin(\phi_{sw}) \\ \sin(\phi_{sw}) & \cos(\phi_{sw}) \end{bmatrix}
\end{cases}
$$



$$
\begin{cases}
\boldsymbol{P}_{st} = \boldsymbol{P}_h + \boldsymbol{R}_{st}\begin{bmatrix} B \\ C \end{bmatrix} \\
\boldsymbol{P}_{sw} = \boldsymbol{P}_h + \boldsymbol{R}_{sw}\begin{bmatrix} B \\ C \end{bmatrix}
\end{cases}
$$

$$
\begin{cases}
\boldsymbol{x} = [P_{st}(1), P_{st}(2), \phi_{st}, P_{sw}(1), P_{sw}(2), \phi_{sw}]^T \\
\boldsymbol{q} = [\phi_{st}, \phi_{sw}]^T
\end{cases}
\Rightarrow \boldsymbol{J} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{q}}
$$

$$
\begin{cases}
\boldsymbol{M} = \mathrm{diag}[m, m, I_c, m, m, I_c] \\
\boldsymbol{F}^a = mg[\sin(\gamma), -\cos(\gamma), 0, \sin(\gamma), -\cos(\gamma), 0]^T
\end{cases}
\Rightarrow \ddot{\boldsymbol{q}} = (\boldsymbol{J}^T\boldsymbol{M}\boldsymbol{J})^{-1}(\boldsymbol{J}^T\boldsymbol{F}^a - \boldsymbol{J}^T\boldsymbol{M}\boldsymbol{D})
$$

投影牛顿-欧拉法(PNE/JMJ)

Matlab运行结果及程序

```
J_st = [...
       - par.R - par.C*cos(phi_st) - par.B*sin(phi_st) - cos(phi_st)*(par.L - par.R),                          0;
                 par.B*cos(phi_st) - par.C*sin(phi_st) - sin(phi_st)*(par.L - par.R),                          0;
                                                                                   1,                          0;
                                          - par.R - cos(phi_st)*(par.L - par.R),     - par.C*cos(phi_sw) - par.B*sin(phi_sw);
                                                  -sin(phi_st)*(par.L - par.R),        par.B*cos(phi_sw) - par.C*sin(phi_sw);
                                                                                   0,                                    1  ];

D = [...
                    phi_st_d^2*(par.C*sin(phi_st) - par.B*cos(phi_st) + sin(phi_st)*(par.L - par.R));
                   -phi_st_d^2*(par.C*cos(phi_st) + par.B*sin(phi_st) + cos(phi_st)*(par.L - par.R));
                                                                                                   0;
         phi_st_d^2*sin(phi_st)*(par.L - par.R) - phi_sw_d^2*(par.B*cos(phi_sw) - par.C*sin(phi_sw));
       - phi_sw_d^2*(par.C*cos(phi_sw) + par.B*sin(phi_sw)) - phi_st_d^2*cos(phi_st)*(par.L - par.R);
                                                                                                   0  ];

J_sw = [...
       - par.C*cos(phi_st) - par.B*sin(phi_st),                                            - par.R - cos(phi_sw)*(par.L - par.R);
         par.B*cos(phi_st) - par.C*sin(phi_st),                                                    -sin(phi_sw)*(par.L - par.R);
                                             1,                                                                               0;
                                             0,      - par.R - par.C*cos(phi_sw) - par.B*sin(phi_sw) - cos(phi_sw)*(par.L - par.R);
                                             0,                par.B*cos(phi_sw) - par.C*sin(phi_sw) - sin(phi_sw)*(par.L - par.R);
                                             0,                                                                            1  ];
```

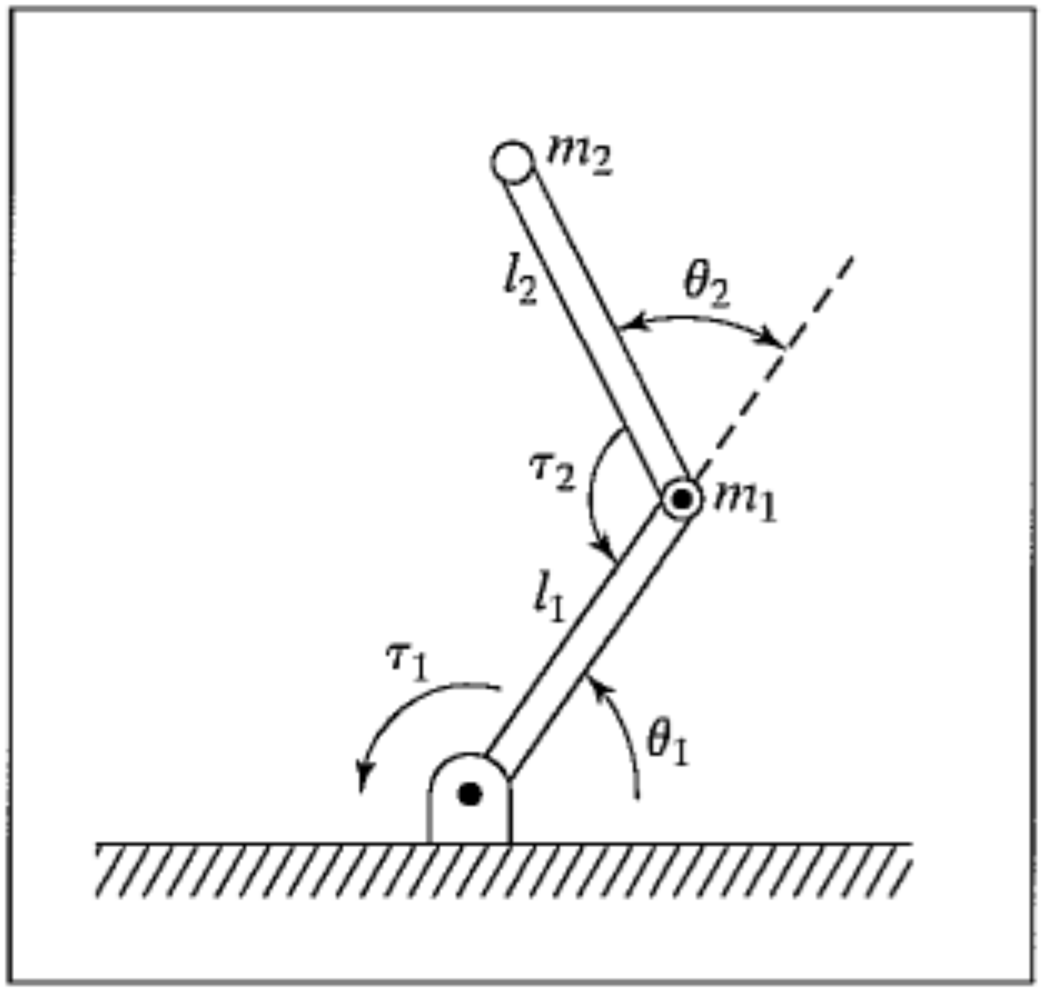
```
% clear memory
clear all;
clc;

% creation of symbolic variables
% leg parameters
syms  L     % leg length
syms  R     % foot radius
syms  B     % horizontal position of the CoM with respect to the hip
syms  C     % vertical position of the CoM with respect to the hip

% generalized coordinates and their derivatives
syms  phi_st  phi_st_d  % counter-clockwise rotation of the stance leg
syms  phi_sw  phi_sw_d  % counter-clockwise rotation of the swing leg

% creation of vector of generalized coordinates
q  = [phi_st;  phi_sw];      % generalized coordinates
qd = [phi_st_d; phi_sw_d];  % velocities of generalized coordinates

% auxiliary relationship: position of hip joint
pos_h = [-R*phi_st - (L-R)*sin(phi_st)
         R         + (L-R)*cos(phi_st) ];

% expression of all model coordinates in terms of generalized coordinates
x_st = [ pos_h + RotationMatrix(phi_st)*[B; C]    % x and y position of CoM of the stance leg
         phi_st                                   % angle of CoM of the stance leg
         pos_h + RotationMatrix(phi_sw)*[B; C]    % x and y position of CoM of the swing leg
         phi_sw  ];                               % angle of CoM of the swing leg

% derivation of partial derivatives of x to q
J_st =(jacobian(x_st,q));
```

```
% derivation of corriolis terms (second derivative of x)
D = (jacobian(J_st*qd,q)*qd);

% print matrices so that the user can copy-paste them to Step.m
PrintMatrix(J_st,'J_st')
PrintMatrix(D,'D')

% for the impact equation we need also to express the model coordinates in
% terms of generalized coordinates with the assumption that the swing leg is
% in contact with the ground

% position of hip joint with repect to the swing leg
pos_h_sw = [-R*phi_sw - (L-R)*sin(phi_sw)
            R         + (L-R)*cos(phi_sw) ];

% expression of all model coordinates in terms of generalized coordinates
x_sw = [ pos_h_sw + RotationMatrix(phi_st)*[B; C]
         phi_st
         pos_h_sw + RotationMatrix(phi_sw)*[B; C]
         phi_sw ];

% derivation of partial derivatives of x to q
J_sw = (jacobian(x_sw,q));

% print matrix so that the user can copy-paste it to Step.m
PrintMatrix(J_sw,'J_sw')
```

本章提纲

- 概述
- 牛顿-欧拉方程（2学时）
- 拉格朗日方程（2学时）
- 投影牛顿欧拉法（PNE）（2学时）
- 案例分析（2学时）
    - 两连杆机器人的牛顿欧拉方程
    - 三连杆机器人的拉格朗日方程

两连杆机器人的牛顿欧拉方程



Outward iterations: $i : 0 \to 5$

$$
\begin{aligned}
{}^{i+1}\omega_{i+1} &= {}^{i+1}_{i}R\,{}^{i}\omega_i + \dot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1},\\
{}^{i+1}\dot{\omega}_{i+1} &= {}^{i+1}_{i}R\,{}^{i}\dot{\omega}_i + {}^{i+1}_{i}R\,{}^{i}\omega_i \times \dot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1} + \ddot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1},\\
{}^{i+1}\dot{v}_{i+1} &= {}^{i+1}_{i}R({}^{i}\dot{\omega}_i \times {}^{i}P_{i+1} + {}^{i}\omega_i \times ({}^{i}\omega_i \times {}^{i}P_{i+1}) + {}^{i}\dot{v}_i),\\
{}^{i+1}\dot{v}_{C_{i+1}} &= {}^{i+1}\dot{\omega}_{i+1} \times {}^{i+1}P_{C_{i+1}}\\
&\quad + {}^{i+1}\omega_{i+1} \times ({}^{i+1}\omega_{i+1} \times {}^{i+1}P_{C_{i+1}}) + {}^{i+1}\dot{v}_{i+1},\\
{}^{i+1}F_{i+1} &= m_{i+1}\,{}^{i+1}\dot{v}_{C_{i+1}},\\
{}^{i+1}N_{i+1} &= {}^{C_{i+1}}I_{i+1}\,{}^{i+1}\dot{\omega}_{i+1} + {}^{i+1}\omega_{i+1} \times {}^{C_{i+1}}I_{i+1}\,{}^{i+1}\omega_{i+1}.
\end{aligned}
$$

Inward iterations: $i : 6 \to 1$

$$
\begin{aligned}
{}^{i}f_i &= {}^{i}_{i+1}R\,{}^{i+1}f_{i+1} + {}^{i}F_i,\\
{}^{i}n_i &= {}^{i}N_i + {}^{i}_{i+1}R\,{}^{i+1}n_{i+1} + {}^{i}P_{C_i} \times {}^{i}F_i\\
&\quad + {}^{i}P_{i+1} \times {}^{i}_{i+1}R\,{}^{i+1}f_{i+1},\\
\tau_i &= {}^{i}n_i^{T}\,{}^{i}\hat{Z}_i.
\end{aligned}
$$

两连杆机器人的牛顿欧拉方程

```
clc;
clear;
syms q1 q2 m1 m2 L1 L2 real

dh_params = [0, 0,  0, q1;
            0, L1, 0, q2;
            0, L2, 0, 0];
mass_center = [L1, 0,  0;
              L2, 0, 0;];
mass = [m1,m2];
inertia_1 = [0,0,0;
            0,0,0;
            0,0,0];
inertia_2 = [0,0,0;
            0,0,0;
            0,0,0];
inertia_tensor(:,:,1) = inertia_1;
inertia_tensor(:,:,2) = inertia_2;
f_ext = [0,0,0;
        0,0,0];
torque = NewtonEulerDynamics(dh_params, mass, mass_center,
inertia_tensor, f_ext)
```

```
function torque_list = NewtonEulerDynamics(dh_list, mass_list, mass_center_list, inertia_tensor_list, f_external)

[rows, columns] = size(dh_list);
number_of_links = rows - 1;
if columns ~= 4
    error('wrong DH parameters!')
end

T = sym([]);
R = sym([]);
a = sym([]);
d = sym([]);
alpha = sym([]);
theta = sym([]);

for i = 1:rows
    eval(['syms ','q',num2str(i),' real;']);
    eval(['syms ','dq',num2str(i),' real;']);
    eval(['syms ','ddq',num2str(i),' real;']);
    eval(['q(i)=','q',num2str(i),';']);
    eval(['dq(i)=','dq',num2str(i),';']);
    eval(['ddq(i)=','ddq',num2str(i),';']);
end

for i = 1:rows
    dh = dh_list(i,:);
    alpha(i) = dh(1);
    a(i) = dh(2);
    d(i) = dh(3);
    theta(i) = dh(4);
    if i == rows
        q(i) = 0;
    end
    T(:,:,i) = [cos(q(i)),           -sin(q(i)),           0,           a(i);
            sin(q(i))*cos(alpha(i)), cos(q(i))*cos(alpha(i)), -sin(alpha(i)), -sin(alpha(i))*d(i);
            sin(q(i))*sin(alpha(i)), cos(q(i))*sin(alpha(i)), cos(alpha(i)), cos(alpha(i))*d(i);
            0,                   0,                   0,          1];
    T = T(:,:,i);

    R(:,:,i) = simplify(inv(T(1:3,1:3)));
    P(:,:,i) = T(1:3,4:4);
end
```

```
z = [0,0,1]';

syms g real


for i = 0:number_of_links-1
    if i == 0
        wi = [0,0,0]';
        dwi = [0,0,0]';
        dvi = [0, g, 0]';
    else
        wi = w(:,i);
        dwi = dw(:,i);
        dvi = dv(:,i);
    end
    w(:,:,i+1) = R(:,:,i+1)*wi + dq(i+1)*z;
    dw(:,:,i+1) = R(:,:,i+1)*dwi + cross(R(:,:,i+1)*wi,dq(i+1)*z) + ddq(i+1)*z;
    dv(:,:,i+1) = R(:,:,i+1)*(cross(dwi,P(:,:,i+1)) + cross(wi,cross(wi,P(:,:,i+1))) + dvi);
    dvc(:,:,i+1) = cross(dw(:,:,i+1),mass_center_list(i+1,:)')...
                + cross(w(:,:,i+1),cross(w(:,:,i+1),mass_center_list(i+1,:)'))...
                + dv(:,:,i+1);
    F(:,:,i+1) = mass_list(i+1)*dvc(:,:,i+1);
    N(:,:,i+1) = inertia_tensor_list(:,:,i+1)*dw(:,:,i+1) + cross(w(:,:,i+1),inertia_tensor_list(:,:,i+1)*w(:,:
end

f = sym([]);
n = sym([]);

%
for i = number_of_links:-1:1
    if i == number_of_links
        f(:,:,i+1) = f_external(1,:)';
        n(:,:,i+1) = f_external(2,:)';
    end
    f(:,:,i) = R(:,:,i+1)\f(:,:,i+1) + F(:,:,i);
    f(:,:,i) = simplify(f(:,:,i));
    n(:,:,i) = N(:,:,i) + R(:,:,i+1)\n(:,:,i+1) + cross(mass_center_list(i,:)',F(:,:,i))...
            + cross(P(:,:,i+1),R(:,:,i+1)\f(:,:,i+1));
    n(:,:,i) = simplify
```

```
(n(:,:,i));
    torque_list(i) = dot(n(:,:,i),z);
end
torque_list = torque_list';
```

两连杆机器人的牛顿欧拉方程

torque =

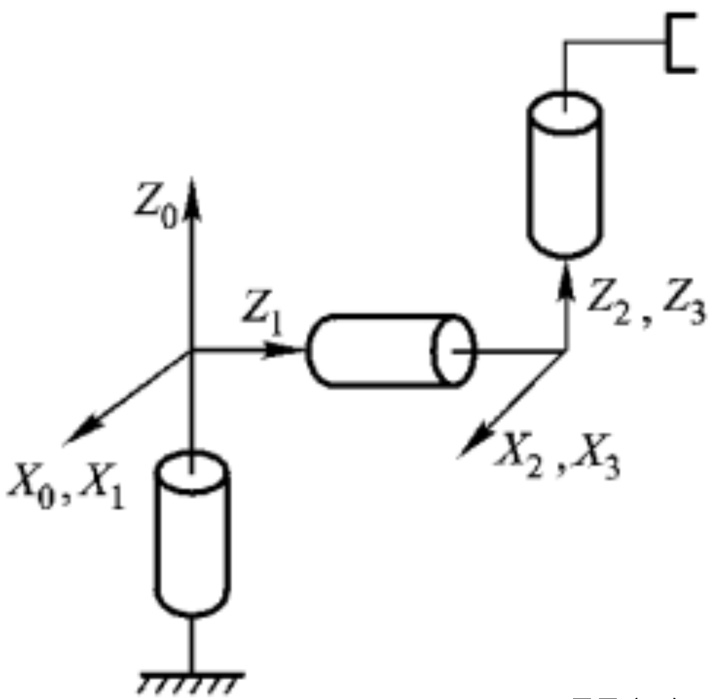
L1^2*ddq1*m1 + L1^2*ddq1*m2 + L2^2*ddq1*m2 + L2^2*ddq2*m2 + L2*g*m2*cos(q1 + q2) + L1*g*m1*cos(q1) + L1*g*m2*cos(q1) - L1*L2*dq2^2*m2*sin(q2) + 2*L1*L2*ddq1*m2*cos(q2) + L1*L2*ddq2*m2*cos(q2) - 2*L1*L2*dq1*dq2*m2*sin(q2)

L2*m2*(cos(q2)*(L1*ddq1 + g*cos(q1)) + sin(q2)*(L1*dq1^2 - g*sin(q1)) + L2*(ddq1 + ddq2))

$$\tau_1 = m_2 l_2^2 \left(\ddot{\theta}_1 + \ddot{\theta}_2\right) + m_2 l_1 l_2 c_2 \left(2\ddot{\theta}_1 + \ddot{\theta}_2\right) + \left(m_1 + m_2\right) l_1^2 \ddot{\theta}_1 - m_2 l_1 l_2 s_2 \dot{\theta}_2^2 - 2 m_2 l_1 l_2 s_2 \dot{\theta}_1 \dot{\theta}_2 + m_2 l_2 g c_{12} + \left(m_1 + m_2\right) l_1 g c_1$$

$$\tau_2 = m_2 l_1 l_2 c_2 \ddot{\theta}_1 + m_2 l_1 l_2 s_2 \dot{\theta}_1^2 + m_2 l_2 g c_{12} + m_2 l_2^2 \left(\ddot{\theta}_1 + \ddot{\theta}_2\right)$$

三连杆机器人的牛顿欧拉方程



$$H(q) = \left[h_{ij}\right]; h_{ij} = \sum_{k=\max(i,j)}^{n} \mathrm{tr}\left(\frac{\partial^0 A_k}{\partial q_i} J_k \frac{\partial \left(A_k\right)^T}{\partial q_j}\right); i, j = 1,2,3$$

$$C(q, q) = \left[c_{ij}\right]; c_{ij} = \sum_{k=1}^{n} \frac{1}{2}\left(\frac{\partial h_{ij}}{\partial q_k} + \frac{\partial h_{ik}}{\partial q_j} - \frac{\partial h_{ik}}{\partial q_i}\right) q_k; i, j = 1,2,3$$

$$G(q) = \left[g_1, g_2, g_3\right]^T; g_i = -\sum_{i=1}^{n} m_j \bar{g}^T \frac{\partial^o A_j}{\partial q_i} \tilde{r}_{Cj}; i = 1,2,3$$

三连杆机器人的牛顿欧拉方程

```
%%
% ++++++++++++++++++++
clear;
syms q1 q2 q3 m1 m2 m3 d2 real
syms Ix1 Iy1 Iz1 Ixy1 Iyz1 Ixz1 real
syms Ix2 Iy2 Iz2 Ixy2 Iyz2 Ixz2 real
syms Ix3 Iy3 Iz3 Ixy3 Iyz3 Ixz3 real
syms xc1 yc1 zc1 xc2 yc2 zc2 xc3 yc3

dh_params = [-pi/2, 0,  0, q1;
           pi/2, 0, d2, q2;
           0, 0, 0, q3];
mass_center = [xc1, yc1, zc1;
            xc2, yc2, zc2;
            xc3, yc3, zc3];
mass = [m1,m2,m3];
inertia_1 = [Ix1,  Ixy1, Ixz1;
           Ixy1, Iy1,  Iyz1;
           Ixz1, Iyz1, Iz1];
inertia_2 = [Ix2,  Ixy2, Ixz2;
           Ixy2, Iy2,  Iyz2;
           Ixz2, Iyz2, Iz2];
inertia_3 = [Ix3,  Ixy3, Ixz3;
           Ixy3, Iy3,  Iyz3;
           Ixz3, Iyz3, Iz3];

inertia_tensor(:,:,1) = inertia_1;
inertia_tensor(:,:,2) = inertia_2;
inertia_tensor(:,:,3) = inertia_3;

[h,c,g] = LagrangianDynamics(dh_p

%%++++++++++++++++++++
```

```
function [H,C,G] = LagrangianDynamics(dh_list, mass_list, mass_center_list, inertia_tensor_list)

[rows, columns] = size(dh_list);
number_of_links = rows;
if columns ~= 4
   error('wrong DH parameters!')
end

for i = 1:rows
   % ???????????????
   eval(['syms ','q',num2str(i),' real;']);
   eval(['syms ','dq',num2str(i),' real;']);
   eval(['syms ','ddq',num2str(i),' real;']);
   eval(['q(i)=','q',num2str(i),';']);
   eval(['dq(i)=','dq',num2str(i),';']);
   eval(['ddq(i)=','ddq',num2str(i),';']);
end

A = sym([]);
for i = 1:number_of_links
   dh = dh_list(i,:);
   alpha(i) = dh(1);
   a(i) = dh(2);
   d(i) = dh(3);
   q(i) = dh(4);
   A(:,:,i) = [cos(q(i)),   -sin(q(i))*cos(alpha(i)),  sir
         sin(q(i)), cos(q(i))*cos(alpha(i)), -sin(alp
         0,          sin(alpha(i)), cos(alpha(i)),  d(i
         0,                 0,                  0,
end
A = simplify(A);

% ??????????{0}?????
A0 = sym([]);
for i = 1:number_of_links
   A0(:,:,i) = eye(4,4);
   for j = 1:i
      A0(:,:,i) = A0(:,:,i)*A(:,:,j);
   end
```

```
J = sym([]);
for i = 1:number_of_links
   T = inertia_tensor_list(1,1,i)+ inertia_t

   J(1,1,i) = T/2-inertia_tensor_list(1,1,i);
   J(2,2,i) = T/2-inertia_tensor_list(2,2,i);
   J(3,3,i) = T/2-inertia_tensor_list(3,3,i);
   J(4,4,i) = mass_list(i);

   J(1,2,i) = inertia_tensor_list(1,2,i);
   J(1,3,i) = inertia_tensor_list(1,3,i);
   J(2,1,i) = inertia_tensor_list(2,1,i);
   J(3,1,i) = inertia_tensor_list(3,1,i);

   J(2,3,i) = inertia_tensor_list(2,3,i);
   J(3,2,i) = inertia_tensor_list(3,2,i);

   J(1,4,i) = mass_list(i)*mass_center_list(
   J(2,4,i) = mass_list(i)*mass_center_list(
   J(3,4,i) = mass_list(i)*mass_center_list(
   J(4,1,i) = J(1,4,i);
   J(4,2,i) = J(2,4,i);
   J(4,3,i) = J(3,4,i);
   %J(:,:,i) = JMatrix(mass_list(i),mass_cer
end
```

```
% ??H(q),?H(q)?????????????
syms tr
for i = 1:number_of_links
   for j = i:number_of_links
      tr = 0;
      for k = j:number_of_links
         tr = tr + trace(eval(['diff(A0(:,:,k),q',num2str(i),')'])*J(:,:,k)*...
            eval(['diff(transpose(A0(:,:,k)),q',num2str(j),')']));
      end
      H(i,j) = simplify(tr);
      H(j,i) = H(i,j);
   end
end

% ??C(q)
for i = 1:number_of_links
   for j = 1:number_of_links
      c = 0;
      for k = 1:number_of_links
         c = c + 1/2*(eval(['diff(H(i,j),q',num2str(k),')'])...
               + eval(['diff(H(i,k),q',num2str(j),')'])...
               - eval(['diff(H(j,k),q',num2str(i),')']))*eval(['dq',num2str(k)]);
      end
      C(i,j) = simplify(c);
   end
end

syms gc
g = [0,0,-gc,0]';

% ??G(q)
for i = 1:number_of_links
   gi = 0;
   for j = 1:number_of_links
      gi = gi - mass_list(j)*g'...
         *eval(['diff(A0(:,:,j),q',num2str(i),')'])...
```

三连杆机器人的牛顿欧拉方程

```
h =

[ Ix2 + Iy1 + Iy3 + d2^2*m2 + d2^2*m3 - Ix2*cos(q2)^2 + Ix3*cos(q3)^2 - Iy3*cos(q2)^2 - Iy3*cos(q3)^2 + Iz2*cos(q2)^2 + Iz3*cos(q2)^2 + Ixy3*sin(2*q3) + Ixz2*sin(2*q2) - Ix3*cos(q2)^2*cos(q3)^2 + Iy3*cos(q2)^2*cos(q3)^2 + 2*d2*m2*yc2 +
2*Ixz3*cos(q2)*cos(q3)*sin(q2) - 2*Iyz3*cos(q2)*sin(q2)*sin(q3) + 2*d2*m3*yc3*cos(q3) + 2*d2*m3*xc3*sin(q3) - 2*Ixy3*cos(q2)^2*cos(q3)*sin(q3), Ixy2*sin(q2) - Iyz2*cos(q2) - Ixy3*sin(q2) - Iyz3*cos(q2)*cos(q3) - Ixz3*cos(q2)*sin(q3) +
2*Ixy3*cos(q3)^2*sin(q2) - Ix3*cos(q3)*sin(q2)*sin(q3) + Iy3*cos(q3)*sin(q2)*sin(q3) - d2*m2*zc2*cos(q2) - d2*m3*zc3*cos(q2) + d2*m2*xc2*sin(q2) + d2*m3*xc3*cos(q3)*sin(q2) - d2*m3*yc3*sin(q2)*sin(q3), Iz3*cos(q2) + Ixz3*cos(q3)*sin(q2) -
Iyz3*sin(q2)*sin(q3) + d2*m3*yc3*cos(q2)*cos(q3) + d2*m3*xc3*cos(q2)*sin(q3)]
[                                                  Ixy2*sin(q2) - Iyz2*cos(q2) - Ixy3*sin(q2) - Iyz3*cos(q2)*cos(q3) - Ixz3*cos(q2)*sin(q3) + 2*Ixy3*cos(q3)^2*sin(q2) - Ix3*cos(q3)*sin(q2)*sin(q3) + Iy3*cos(q3)*sin(q2)*sin(q3) - d2*m2*zc2*cos(q2) -
d2*m3*zc3*cos(q2) + d2*m2*xc2*sin(q2) + d2*m3*xc3*cos(q3)*sin(q2) - d2*m3*yc3*sin(q2)*sin(q3),                                                                  Ix3/2 + Iy2 +
Iy3/2 - (Ix3*cos(2*q3))/2 + (Iy3*cos(2*q3))/2 - Ixy3*sin(2*q3),                                             - Iyz3*cos(q3) - Ixz3*sin(q3)]
[                                                                                                                   Iz3*cos(q2) + Ixz3*cos(q3)*sin(q2) - Iyz3*sin(q2)*sin(q3) +
d2*m3*yc3*cos(q2)*cos(q3) + d2*m3*xc3*cos(q2)*sin(q3),                                                                                                   - Iyz3*cos(q3) -
Ixz3*sin(q3),                                                     Iz3]


c =

[ dq2*(Iyz3*sin(q3) - Ixz3*cos(q3) - Ixz2 + 2*Ixz2*cos(q2)^2 + (Ix2*sin(2*q2))/2 + (Iy3*sin(2*q2))/2 - (Iz2*sin(2*q2))/2 - (Iz3*sin(2*q2))/2 + 2*Ixz3*cos(q2)^2*cos(q3) - 2*Iyz3*cos(q2)^2*sin(q3) + Ix3*cos(q2)*cos(q3)^2*sin(q2) -
Iy3*cos(q2)*cos(q3)^2*sin(q2) + 2*Ixy3*cos(q2)*cos(q3)*sin(q2)*sin(q3)) - dq3*(Ixy3 - Ixy3*cos(q2)^2 - 2*Ixy3*cos(q3)^2 + (Ix3*sin(2*q3))/2 - (Iy3*sin(2*q3))/2 + 2*Ixy3*cos(q2)^2*cos(q3)^2 + Iyz3*cos(q2)*cos(q3)*sin(q2) +
Ixz3*cos(q2)*sin(q2)*sin(q3) - d2*m3*xc3*cos(q3) + d2*m3*yc3*sin(q3) - Ix3*cos(q2)^2*cos(q3)*sin(q3) + Iy3*cos(q2)^2*cos(q3)*sin(q3)), dq1*(Iyz3*sin(q3) - Ixz3*cos(q3) - Ixz2 + 2*Ixz2*cos(q2)^2 + (Ix2*sin(2*q2))/2 + (Iy3*sin(2*q2))/2 -
(Iz2*sin(2*q2))/2 - (Iz3*sin(2*q2))/2 + 2*Ixz3*cos(q2)^2*cos(q3) - 2*Iyz3*cos(q2)^2*sin(q3) + Ix3*cos(q2)*cos(q3)^2*sin(q2) - Iy3*cos(q2)*cos(q3)^2*sin(q2) + 2*Ixy3*cos(q2)*cos(q3)*sin(q2)*sin(q3)) + dq2*(Ixy2*cos(q2) - Ixy3*cos(q2) + Iyz2*sin(q2)
+ Iyz3*cos(q3)*sin(q2) + Ixz3*sin(q2)*sin(q3) + 2*Ixy3*cos(q2)*cos(q3)^2 - Ix3*cos(q2)*cos(q3)*sin(q3) + Iy3*cos(q2)*cos(q3)*sin(q3) + d2*m2*xc2*cos(q2) + d2*m2*zc2*sin(q2) + d2*m3*zc3*sin(q2) + d2*m3*xc3*cos(q2)*cos(q3) -
d2*m3*yc3*cos(q2)*sin(q3)) - (dq3*sin(q2)*(Iz3 + Ix3*(2*cos(q3)^2 - 1) - Iy3*(2*cos(q3)^2 - 1) + 4*Ixy3*cos(q3)*sin(q3) + 2*d2*m3*yc3*cos(q3) + 2*d2*m3*xc3*sin(q3)))/2, - dq3*(Iyz3*cos(q3)*sin(q2) + Ixz3*sin(q2)*sin(q3) -
d2*m3*xc3*cos(q2)*cos(q3) + d2*m3*yc3*cos(q2)*sin(q3)) - dq1*(Ixy3 - Ixy3*cos(q2)^2 - 2*Ixy3*cos(q3)^2 + (Ix3*sin(2*q3))/2 - (Iy3*sin(2*q3))/2 + 2*Ixy3*cos(q2)^2*cos(q3)^2 + Iyz3*cos(q2)*cos(q3)*sin(q2) + Ixz3*cos(q2)*sin(q2)*sin(q3) -
d2*m3*xc3*cos(q3) + d2*m3*yc3*sin(q3) - Ix3*cos(q2)^2*cos(q3)*sin(q3) + Iy3*cos(q2)^2*cos(q3)*sin(q3)) - (dq2*sin(q2)*(Iz3 + Ix3*(2*cos(q3)^2 - 1) - Iy3*(2*cos(q3)^2 - 1) + 4*Ixy3*cos(q3)*sin(q3) + 2*d2*m3*yc3*cos(q3) + 2*d2*m3*xc3*sin(q3)))/
2]
[                                                        dq3*((Ix3*sin(q2))/2 - (Iy3*sin(q2))/2 + (Iz3*sin(q2))/2 - Ixz3*cos(q2)*cos(q3) + Iyz3*cos(q2)*sin(q3) - Ix3*cos(q3)^2*sin(q2) + Iy3*cos(q3)^2*sin(q2) - 2*Ixy3*cos(q3)*sin(q2)*sin(q3)) -
dq1*(Iyz3*sin(q3) - Ixz3*cos(q3) - Ixz2 + 2*Ixz2*cos(q2)^2 + (Ix2*sin(2*q2))/2 + (Iy3*sin(2*q2))/2 - (Iz2*sin(2*q2))/2 - (Iz3*sin(2*q2))/2 + 2*Ixz3*cos(q2)^2*cos(q3) - 2*Iyz3*cos(q2)^2*sin(q3) + Ix3*cos(q2)*cos(q3)^2*sin(q2) -
Iy3*cos(q2)*cos(q3)^2*sin(q2) + 2*Ixy3*cos(q2)*cos(q3)*sin(q2)*sin(q3)),
-dq3*(Ixy3*cos(2*q3) - (Ix3*sin(2*q3))/2 + (Iy3*sin(2*q3))/2),                                                                                  Ixy3*dq2 - 2*Ixy3*dq2*cos(q3)^2 + (Ix3*dq2*sin(2*q3))/2 -
(Iy3*dq2*sin(2*q3))/2 - Ixz3*dq3*cos(q3) + (Ix3*dq1*sin(q2))/2 - (Iy3*dq1*sin(q2))/2 + Iyz3*dq3*sin(q3) + (Iz3*dq1*sin(q2))/2 - Ix3*dq1*cos(q3)^2*sin(q2) + Iy3*dq1*cos(q3)^2*sin(q2) - Ixz3*dq1*cos(q2)*cos(q3) + Iyz3*dq1*cos(q2)*sin(q3) -
2*Ixy3*dq1*cos(q3)*sin(q2)*sin(q3)]
[                                                                  dq1*(Ixy3 - Ixy3*cos(q2)^2 - 2*Ixy3*cos(q3)^2 + (Ix3*sin(2*q3))/2 - (Iy3*sin(2*q3))/2 + 2*Ixy3*cos(q2)^2*cos(q3)^2 + Iyz3*cos(q2)*cos(q3)*sin(q2) +
Ixz3*cos(q2)*sin(q2)*sin(q3) - d2*m3*xc3*cos(q3) + d2*m3*yc3*sin(q3) - Ix3*cos(q2)^2*cos(q3)*sin(q3) + Iy3*cos(q2)^2*cos(q3)*sin(q3)) - dq2*((Ix3*sin(q2))/2 - (Iy3*sin(q2))/2 + (Iz3*sin(q2))/2 - Ixz3*cos(q2)*cos(q3) + Iyz3*cos(q2)*sin(q3) -
Ix3*cos(q3)^2*sin(q2) + Iy3*cos(q3)^2*sin(q2) - 2*Ixy3*cos(q3)*sin(q2)*sin(q3)),
dq2*(Ixy3*(2*cos(q3)^2 - 1) - Ix3*cos(q3)*sin(q3) + Iy3*cos(q3)*sin(q3)) - dq1*((Ix3*sin(q2))/2 - (Iy3*sin(q2))/2 + (Iz3*sin(q2))/2 - Ixz3*cos(q2)*cos(q3) + Iyz3*cos(q2)*sin(q3) - Ix3*cos(q3)^2*sin(q2) + Iy3*cos(q3)^2*sin(q2) -
2*Ixy3*cos(q3)*sin(q2)*sin(q3)),
0]


g =

                                                              0
 -conj(gc)*(m2*xc2*cos(q2) + m2*zc2*sin(q2) + m3*zc3*sin(q2) + m3*xc3*cos(q2)*cos(q3) - m3*yc3*cos(q2)*sin(q3))
                                   m3*conj(gc)*sin(q2)*(yc3*cos(q3) + xc3*sin(q3))
```

$$h_{11} = {}^{1}I_{y} + s_2^2 ({}^{2}I_{x} + c_3^2\,{}^{3}I_{x} + 2cs_3\,{}^{3}I_{xy} + s_3^2\,{}^{3}I_{y}) +$$
$$2cs_2 ({}^{2}I_{xz} + c_3\,{}^{3}I_{xz} - s_3\,{}^{3}I_{yz}) + c_2^2 ({}^{2}I_{z} + {}^{3}I_{z}) +$$
$$2d_2 (m_2\,{}^{2}y_{c2} + s_3 m_3\,{}^{3}x_{c3} + c_3 m_3\,{}^{3}y_{c3}) + d_2^2 (m_2 + m_3)$$
$$h_{12} = h_{21} = s_2 [{}^{2}I_{xy} + d_2 m_2\,{}^{2}x_{c2} - cs_3 ({}^{3}I_{x} - {}^{3}I_{y}) +$$
$$(c_3^2 - s_3^2)\,{}^{3}I_{xy} + d_2 (c_3 m_3\,{}^{3}x_{c3} - s_3 m_3\,{}^{3}y_{c3})] -$$
$$c_2 [{}^{2}I_{yz} + s_3\,{}^{3}I_{xz} + c_3\,{}^{3}I_{yz} + d_2 (m_2\,{}^{2}z_{c2} + m_3\,{}^{3}z_{c3})]$$
$$h_{13} = h_{31} = s_2 (c_3\,{}^{3}I_{xz} - s_3\,{}^{3}I_{yz}) + c_2 [{}^{3}I_{z} + d_2 (s_3 m_3\,{}^{3}x_{c3} +$$
$$c_3 m_3\,{}^{3}y_{c3})]$$
$$h_{22} = {}^{2}I_{y} + s_3^2\,{}^{3}I_{x} - 2cs_3\,{}^{3}I_{xy} + c_3^2\,{}^{3}I_{y}$$
$$h_{23} = h_{32} = -s_3\,{}^{3}I_{xz} - c_3\,{}^{3}I_{yz}$$
$$h_{33} = {}^{3}I_{z}$$
$$c_{11} = [cs_2 ({}^{2}I_{x} - {}^{2}I_{z} + c_3^2\,{}^{3}I_{x} + 2cs_3\,{}^{3}I_{xy} + s_3^2\,{}^{3}I_{y} - {}^{3}I_{z}) +$$
$$(c_2^2 - s_2^2) ({}^{2}I_{xz} + c_3\,{}^{3}I_{xz} - s_3\,{}^{3}I$$
$$\{s_2^2 [cs_3 (-{}^{3}I_{x} + {}^{3}I_{y}) + (c_3^2 -$$
$$cs_2 (s_3\,{}^{3}I_{xz} + c_3\,{}^{3}I_{yz}) + d_2 (c_3 m$$

$$c_{12} = [cs_2 ({}^{2}I_{x} - {}^{2}I_{z} + c_3^2\,{}^{3}I_{x} + 2cs_3\,{}^{3}I_{xy} + s_3^2\,{}^{3}I_{y} - {}^{3}I_{z}) +$$
$$(c_2^2 - s_2^2) ({}^{2}I_{xz} + c_3\,{}^{3}I_{xz} - s_3\,{}^{3}I_{yz})] q_1 +$$
$$\{c_2 [{}^{2}I_{xy} - cs_3 ({}^{3}I_{x} - {}^{3}I_{y}) + (c_3^2 - s_3^2)\,{}^{3}I_{xy} +$$
$$d_2 (m_2\,{}^{2}x_{c2} + c_3 m_3\,{}^{3}x_{c3} - s_3 m_3\,{}^{3}y_{c3})] +$$
$$s_2 [{}^{2}I_{yz} + s_3\,{}^{3}I_{xz} + c_3\,{}^{3}I_{yz} + d_2 (m_2\,{}^{2}z_{c2} + m_3\,{}^{3}z_{c3})]\} q_2 -$$
$$\frac{1}{2} s_2 [(c_3^2 - s_3^2) ({}^{3}I_{x} - {}^{3}I_{y}) + 4cs_3\,{}^{3}I_{xy} + {}^{3}I_{z} +$$
$$2d_2 (s_3 m_3\,{}^{3}x_{c3} + c_3 m_3\,{}^{3}y_{c3})] q_3$$
$$c_{13} = \{s_2^2 [cs_3 (-{}^{3}I_{x} + {}^{3}I_{y}) + (c_3^2 - s_3^2)\,{}^{3}I_{xy}] -$$
$$cs_2 (s_3\,{}^{3}I_{xz} + c_3\,{}^{3}I_{yz}) + d_2 (c_3 m_3\,{}^{3}x_{c3} - s_3 m_3\,{}^{3}y_{c3})\} q_1 -$$
$$\frac{1}{2} s_2 [(c_3^2 - s_3^2) ({}^{3}I_{x} - {}^{3}I_{y}) + 4cs_3\,{}^{3}I_{xz} + {}^{3}I_{z} +$$
$$2d_2 (s_3 m_3\,{}^{3}x_{c3} + c_3 m_3\,{}^{3}y_{c3})] q_2 -$$
$$[s_2 (s_3\,{}^{3}I_{xz} + c_3\,{}^{3}I_{yz}) - d_2 c_2 (c_3 m_3\,{}^{3}x_{c3} - s_3 m_3\,{}^{3}y_{c3})] q_3$$
$$c_{21} = -[cs_2 ({}^{2}I_{x} - {}^{2}I_{z} + c_3^2\,{}^{3}I_{x} + 2cs_3\,{}^{3}I_{xy} + s_3^2\,{}^{3}I_{y} - {}^{3}I_{z}) +$$
$$(c_2^2 - s_2^2) ({}^{2}I_{xz} + c_3\,{}^{3}I_{xz} - s_3\,{}^{3}I_{yz})] q_1 -$$
$$\frac{1}{2} \{s_2 [(c_3^2 - s_3^2) ({}^{3}I_{x} - {}^{3}I_{y}) + 4cs_3\,{}^{3}I_{xy} - {}^{3}I_{z}] +$$
$$2c_2 (c_3\,{}^{3}I_{xz} - s_3\,{}^{3}I_{yz})\} q_3$$
$$c_{22} = [cs_3 ({}^{3}I_{x} - {}^{3}I_{y}) - (c_3^2 - s_3^2)\,{}^{3}I_{xy}] q_3$$

$$c_{23} = -\frac{1}{2} \{s_2 [(c_3^2 - s_3^2) ({}^{3}I_{x} - {}^{3}I_{y}) + 4cs_3\,{}^{3}I_{xy} - {}^{3}I_{z}] +$$
$$2c_2 (c_3\,{}^{3}I_{xz} - s_3\,{}^{3}I_{yz})\} q_1 + [cs_3 ({}^{3}I_{x} - {}^{3}I_{y}) -$$
$$(c_3^2 - s_3^2)\,{}^{3}I_{xy}] q_2 - (c_3\,{}^{3}I_{xz} - s_3\,{}^{3}I_{yz}) q_3$$
$$c_{31} = -\{s_2^2 [cs_3 (-{}^{3}I_{x} + {}^{3}I_{y}) + (c_3^2 - s_3^2)\,{}^{3}I_{xy}] -$$
$$cs_2 (s_3\,{}^{3}I_{xz} + c_3\,{}^{3}I_{yz}) + d_2 (c_3 m_3\,{}^{3}x_{c3} - s_3 m_3\,{}^{3}y_{c3})\} q_1 \times$$
$$\frac{1}{2} \{s_2 [(c_3^2 - s_3^2) ({}^{3}I_{x} - {}^{3}I_{y}) + 4cs_3\,{}^{3}I_{xz} - {}^{3}I_{z}] +$$
$$2c_2 (c_3\,{}^{3}I_{xz} - s_3\,{}^{3}I_{yz})\} q_2$$
$$c_{32} = \frac{1}{2} \{s_2 [(c_3^2 - s_3^2) ({}^{3}I_{x} - {}^{3}I_{y}) + 4cs_3\,{}^{3}I_{xz} - {}^{3}I_{z}] +$$
$$2c_2 (c_3\,{}^{3}I_{xz} - s_3\,{}^{3}I_{yz})\} q_1 -$$
$$[cs_3 ({}^{3}I_{x} - {}^{3}I_{y}) - (c_3^2 - s_3^2)\,{}^{3}I_{xy}] q_2$$
$$c_{33} = 0$$
$$g_1 = 0$$
$$g_2 = -s_2 (m_2\,{}^{2}z_{c2} + m_3\,{}^{3}z_{c3}) g_C -$$
$$c_2 (m_2\,{}^{2}x_{c2} + c_3 m_3\,{}^{3}x_{c3} - s_3 m_3\,{}^{3}y_{c3}) g_C$$
$$g_3 = s_2 (s_3 m_3\,{}^{3}x_{c3} + c_3 m_3\,{}^{3}y_{c3}) g_C$$