清華大學

Tsinghua University

智能机器人

课程报告

姓名：	高嘉伟
学号：	2020011073
日期：	2023年 6月 27日

题目分析与公式推导

首先对题目进行分析. 这是一个综合问题, 涉及冗余机器人的逆运动学求解, 机器人动力学, 以及面对实际问题建模优化目标的过程.

1 拉格朗日方程

我们的目标是求解机器人关节力矩τ_i, 而求解力矩可以通过机器人动力学(Dynamics)来进行.

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

而为求得$H(\boldsymbol{q})$, $C(\boldsymbol{q},\dot{\boldsymbol{q}})$与$G(\boldsymbol{q})$, 我们可以使用牛顿-欧拉法或者拉格朗日方程进行求解.

获得$H(\boldsymbol{q})$, $C(\boldsymbol{q},\dot{\boldsymbol{q}})$与$G(\boldsymbol{q})$后, 只需求得满足题设条件的$\ddot{\boldsymbol{q}}$与$\dot{\boldsymbol{q}}$, 而根据题意, 此时机器人处于静止状态, 因而有$\dot{\boldsymbol{q}}=0$, 故核心是求解$\ddot{\boldsymbol{q}}$.

由关系

$$\ddot{\boldsymbol{p}}=J(\boldsymbol{q})\ddot{\boldsymbol{q}}+\dot{J}(\boldsymbol{q})\dot{\boldsymbol{q}}$$

且机器人处于静止状态, 因而有

$$\ddot{\boldsymbol{p}}=J(\boldsymbol{q})\ddot{\boldsymbol{q}}$$

如果对于简单的如2自由度机器人, 则可以直接求解(雅可比矩阵可逆), 但是对于冗余度机器人, 给定末端的坐标与速度等条件后, 有无穷组可能的解. 而通过使某种性能指标最优, 可以获得一组确定的解. 此时涉及到冗余机器人逆运动学求解的问题.

2 冗余机器人逆运动学

对于冗余机器人逆运动学, 由于给定末端的坐标与速度等条件后, 有无穷组可能的解, 因而我们需要确定目标函数.

首先推导一般情况下的公式, 设给定关节加速度的二次型目标函数为:

$$G(\ddot{\boldsymbol{q}})=\ddot{\boldsymbol{q}}^T\boldsymbol{W}\ddot{\boldsymbol{q}}$$

其中, $\boldsymbol{W}\in R^{3\times3}$为加权矩阵.

在实际应用中, 我们一般希望令$G(\ddot{\boldsymbol{q}})$在满足末端期望加速度的情况下达到最小.

使用拉格朗日乘子法, 建立新的广义目标函数:

$$G(\boldsymbol{\theta},\lambda)=\ddot{\boldsymbol{q}}^T\boldsymbol{W}\ddot{\boldsymbol{q}}+\lambda^T(\ddot{\boldsymbol{p}}-\boldsymbol{J}\ddot{\boldsymbol{q}})$$

则该函数的极值应满足

$$\begin{cases}\frac{\partial G}{\partial\ddot{\boldsymbol{q}}}=0\\ \frac{\partial G}{\partial\lambda}=0\end{cases}$$

也即

$$\begin{cases} 2\boldsymbol{W}\ddot{\boldsymbol{q}} - \boldsymbol{J}^\top \lambda = 0 \\ \ddot{\boldsymbol{p}} - \boldsymbol{J}\ddot{\boldsymbol{q}} = 0 \end{cases}$$

则有

$$\begin{cases} \ddot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{W}^{-1}\boldsymbol{J}^\top \lambda \\ (\boldsymbol{J}\boldsymbol{W}^{-1}\boldsymbol{J}^\top)\lambda = 2\ddot{\boldsymbol{p}} \end{cases}$$

进一步得到

$$\lambda = 2\left(\boldsymbol{J}\boldsymbol{W}^{-1}\boldsymbol{J}^\top\right)^{-1}\ddot{\boldsymbol{p}}$$

进而可得

$$\ddot{\boldsymbol{q}} = \boldsymbol{W}^{-1}\boldsymbol{J}^\top\left(\boldsymbol{J}\boldsymbol{W}^{-1}\boldsymbol{J}^\top\right)^{-1}\ddot{\boldsymbol{p}}$$

3 动力学求解力矩

在求解获得$\ddot{\boldsymbol{q}}$后, 通过

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

即可求得关节力矩.

具体分析

1 拉格朗日方程求解

通过列写拉格朗日方程(代码见附录), 求解得本问题中初始条件下H, C, G如下:

$$H = \begin{pmatrix} 5l^2m & \frac{13l^2m}{6} & \frac{5l^2m}{6} \\ \frac{13l^2m}{6} & \frac{5l^2m}{3} & \frac{l^2m}{3} \\ \frac{5l^2m}{6} & \frac{l^2m}{3} & \frac{l^2m}{3} \end{pmatrix}$$

$$C = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$G = \begin{pmatrix} \frac{9\sqrt{2}glm}{4} \\ \sqrt{2}glm \\ \frac{\sqrt{2}glm}{4} \end{pmatrix}$$

2 最小化关节加速度

该优化目标的物理含义为: 关节空间加速度的二范数最小. 在日常应用中有这样的需求, 如关节空间中的控制器不希望关节空间中的状态变换太剧烈, 以及希望安全性的保证, 防止加速度较大而需要提供较大的力矩进而影响机器人的使用寿命.

由于目标为最小化关节加速度的平方:

$$\boldsymbol{H}_A = \frac{1}{2}|\ddot{\boldsymbol{q}}|^2$$

则对应

$$\boldsymbol{W} = \boldsymbol{I}$$

此时

$$\ddot{\boldsymbol{q}} = \boldsymbol{J}^{\top}\left(\boldsymbol{J}\boldsymbol{J}^{\top}\right)^{-1}\ddot{\boldsymbol{p}}$$

使用Matlab求解得到:

$$\ddot{\boldsymbol{q}} = \begin{pmatrix} -\frac{\sqrt{2}}{6l} \\ \frac{2\sqrt{2}}{3l} \\ -\frac{5\sqrt{2}}{6l} \end{pmatrix}$$

$$\tau = \begin{pmatrix} \frac{9\sqrt{2}glm}{4} - \frac{\sqrt{2}lm}{12} \\ \frac{17\sqrt{2}lm}{36} + \sqrt{2}glm \\ \frac{\sqrt{2}glm}{4} - \frac{7\sqrt{2}lm}{36} \end{pmatrix}$$

3 最小化关节绝对加速度

该优化目标的物理含义为: 在笛卡尔坐标系下关节加速度的二范数最小. 除去对机器人安全性的保证防止损害机器人寿命之外, 还希望对机器人工作环境中的安全性有所保证, 防止笛卡尔空间中变化过于剧烈而影响到机器人生产环境中的其他物体或人.

最小化关节绝对加速度的平方:

$$\boldsymbol{H}_B = \frac{1}{2}|\ddot{\boldsymbol{q}}_a|^2$$

由于

$$\ddot{\boldsymbol{q}}_{a,i} = \sum_{j=1}^{i} \ddot{\boldsymbol{q}}_j$$

也即

$$\ddot{\boldsymbol{q}}_a = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \ddot{\boldsymbol{q}}$$

记$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$, 则此时

$$\boldsymbol{W} = \boldsymbol{A}^{\top}\boldsymbol{A}$$

使用Matlab求解得到:

$$\ddot{\boldsymbol{q}} = \begin{pmatrix} -\frac{\sqrt{2}}{4l} \\ \frac{3\sqrt{2}}{4l} \\ -\frac{3\sqrt{2}}{4l} \end{pmatrix}$$

$$\tau = \begin{pmatrix} \frac{9\sqrt{2}glm}{4} - \frac{\sqrt{2}lm}{4} \\ \frac{11\sqrt{2}lm}{24} + \sqrt{2}glm \\ \frac{\sqrt{2}glm}{4} - \frac{5\sqrt{2}lm}{24} \end{pmatrix}$$

4 最小化惯量加权后的加速度

可将物理含义近似解释为“能量”与“负载力”. 由于在日常应用场景中, 不同机械臂的质量不同, 而驱动不同质量机械臂所需要的能量与力矩显然不同, 故在之前两种考虑的基础上, 我们希望惯量加权后的加速度二范数最小.

最小化惯量加权后的关节加速度的平方:

$$\boldsymbol{H}_C = \frac{1}{2}\ddot{\boldsymbol{q}}^T \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}$$

则通过拉格朗日方程求解出机器人的$\boldsymbol{M}(\boldsymbol{q})$之后, 取权重矩阵

$$\boldsymbol{W} = \boldsymbol{H}$$

则通过Matlab 程序求解得到:

$$\ddot{\boldsymbol{q}} = \begin{pmatrix} \frac{\sqrt{2}}{20l} \\ \frac{9\sqrt{2}}{20l} \\ -\frac{21\sqrt{2}}{20l} \end{pmatrix}$$

$$\tau = \begin{pmatrix} \frac{7\sqrt{2}lm}{20} + \frac{9\sqrt{2}glm}{4} \\ \frac{61\sqrt{2}lm}{120} + \sqrt{2}glm \\ \frac{\sqrt{2}glm}{4} - \frac{19\sqrt{2}\,\mathrm{lm}}{120} \end{pmatrix}$$

5 最小化关节驱动力矩

在日常机器人的使用中, 我们考虑到每个关节处驱动器可以提供力矩的范围, 以及不希望过分损耗驱动器的寿命, 故希望限制最小化关节的驱动力矩.

由动力学方程, 关节驱动力矩如下:

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

则可定义Cost Function为

$$\begin{aligned} \boldsymbol{H}_D &= \boldsymbol{\tau}^\top \boldsymbol{\tau} \\ &= \left(H(\boldsymbol{q})\ddot{\boldsymbol{q}} + G(\boldsymbol{q})\right)^\top \left(H(\boldsymbol{q})\ddot{\boldsymbol{q}} + G(\boldsymbol{q})\right) \\ &= \ddot{\boldsymbol{q}} H(\boldsymbol{q})^\top H(\boldsymbol{q})\ddot{\boldsymbol{q}}^\top + 2G(\boldsymbol{q})^\top H(\boldsymbol{q})\ddot{\boldsymbol{q}} + G(\boldsymbol{q})^\top G(\boldsymbol{q}) \end{aligned}$$

为方便起见, 将$H(\boldsymbol{q})$记为H, 将$G(\boldsymbol{q})$记为G. 由于$G^\top G$为常数, 故可省去, 则Cost Function为

$$\boldsymbol{H}_D = \ddot{\boldsymbol{q}} H^\top H \ddot{\boldsymbol{q}}^\top + 2G^\top H\ddot{\boldsymbol{q}}$$

则使用拉格朗日乘子法, 构建广义目标函数

$$G(\boldsymbol{\theta}, \lambda) = \ddot{\boldsymbol{q}} H^\top H \ddot{\boldsymbol{q}}^\top + 2G^\top H\ddot{\boldsymbol{q}} + \lambda^T(\ddot{\boldsymbol{p}} - \boldsymbol{J}\ddot{\boldsymbol{q}})$$

则求解该函数的极值, 可得:

$$\begin{cases} \frac{\partial G}{\partial \ddot{\boldsymbol{q}}} = 0 \\ \frac{\partial G}{\partial \lambda} = 0 \end{cases}$$

也即

$$\begin{cases} 2H^\top H\ddot{\boldsymbol{q}} + 2H^\top G - \boldsymbol{J}^\top \lambda = 0 \\ \ddot{\boldsymbol{p}} - \boldsymbol{J}\ddot{\boldsymbol{q}} = 0 \end{cases}$$

则有

$$\begin{cases} \ddot{\boldsymbol{q}} = \frac{1}{2}(H^\top H)^{-1}(\boldsymbol{J}^\top \lambda - 2H^\top G) \\ \boldsymbol{J}(H^\top H)^{-1}(\boldsymbol{J}^\top \lambda - 2H^\top G) = 2\ddot{\boldsymbol{p}} \end{cases}$$

进一步得到

$$\lambda = 2\left(\boldsymbol{J}(H^\top H)^{-1}\boldsymbol{J}^\top\right)^{-1}\left(\ddot{\boldsymbol{p}} + \boldsymbol{J}(H^\top H)^{-1}H^\top G\right)$$

代入可得

$$\ddot{\boldsymbol{q}} = (H^\top H)^{-1}\boldsymbol{J}^\top\left(\boldsymbol{J}(H^\top H)^{-1}\boldsymbol{J}^\top\right)^{-1}(\ddot{\boldsymbol{p}} + \boldsymbol{J}(H^\top H)^{-1}H^\top G) - (H^\top H)^{-1}H^\top G$$

通过 Matlab 求解可得.

$$\ddot{\boldsymbol{q}} = \begin{pmatrix} -\frac{\sqrt{2}(339g+40)}{292l} \\ \frac{\sqrt{2}(339g+186)}{292l} \\ \frac{\sqrt{2}(339g-252)}{292l} \end{pmatrix}$$

$$\tau = \begin{pmatrix} -\frac{7\sqrt{2}\,\mathrm{lm}(3g+1)}{292} \\ \frac{\sqrt{2}\,\mathrm{lm}(1413g+836)}{1752} \\ \frac{\sqrt{2}\,\mathrm{lm}(99g-332)}{1752} \end{pmatrix}$$

讨论: 使用深度强化学习方法确定最优方案

在以上尝试中, 我们试图用传统机器人学方法确定满足条件的关节力矩, 其中涉及到“启发式”的优化目标设计. 这样的好处是机器人行动的可解释性较强, 且有安全性的保证, 可以尽可能地满足设计者的目的.

然而, 在日常生活中, 我们有时并不能将优化目标一直固定下来, 因为机械臂需要在不同的环境之间进行切换, 我们需要有闭环反馈的思想, 让机器人意识到自己的环境变化了, 以及时切换优化目标; 此外, 我们定义的优化目标虽然有明确的物理含义, 但我们对于环境的理解也限制了机器人表现的上限.

深度强化学习近年来取得了显著的成功, 自最开始的DQN, 到之后的PPO等算法, 以及最近Offline RL, Decision Transformer等的迅猛发展, 使得机器人在一些复杂任务中的表现越来越好. 然而, 深度强化学习的缺点则是可解释性差, 以及样本采集的效率低, 安全性差, 机器人很容易在“探索”的过程中报废等.

考虑到结合深度强化学习与传统机器人方法的优点, 同时尽量避免它们的缺点, 我们考虑使用深度强化学习方法为机器人构建“决策网络”, 而这一小型“决策网络”的目标仅为确定上述冗余机器人逆运动学求解过程的优化目标. 具体方法可以将上述优化目标函数化, 用Deep Q Network 对其进行建模, 并且将机器人传感器的输入作为输入数据, 之后机器人可以在探索过程中更改Deep Q Network 中的Policy, 而我们给机器人设定目标, 如要求机械臂完成动作的质量尽可能地高, 或者机械臂控制的动作精度尽可能保证, 这样的话“决策网络”通过更改其 Policy 可以实现探索, 进而评估当前优化目标的合理性.

这一思想同时结合了深度强化学习与传统机器人的方法的优点. 比如, 使用深度强化学习函数建模, 可以使得机器人在不同的复杂环境及时切换, 并且通过对环境的“探索”确定此时最优的优化目标; 而结合传统机器人方法, 我们这里Deep Q Network控制的并不是多关节多自由度的高维空间中的控制问题, 而仅仅对“优化目标的选择”进行优化, 防止了深度强化学习在高维情况下表现不佳的问题, 同时, 我们也可以给优化目标实行限制, 借鉴CPO或TRPO等Safe RL

相关思想, 保证机器人的安全性, 避免机器人在探索过程中出现报废. 进一步, 我们也可以借鉴Offline RL相关思想, 采用Sim2Real的方法, 让机器人利用以往数据, 增加样本采样效率, 更好更快地学到更优策略.

而针对这一问题的分析与思考, 也是对老师课上对“控制 学习 优化”三者关系问题的一个回应. 在我看来, 目前仍需要以控制与优化为主, 学习为辅的架构, 用深度学习的方法控制一小部分的决策网络, 同时给它加上限制以保证可解释性与安全性, 而剩下部分采用传统控制方法或者人为启发式构建优化目标的方式以解决. 而随着时代的进步与科研的进展, 深度学习的可解释性越来越强, 威力越来越大, 学习方法在决策过程中的占比应越来越大, 最终实现自主决策, 自主智能, “Learning to Control”的理想. 这里对于一个简单机械臂控制问题的讨论即为佐证.

附录: Matlab代码

本次大作业求解中所需的所有Matlab代码如下:

```
clc
clear
close all

syms q1 q2 q3 l
%雅可比矩阵计算
T = trotz(q1) * transl([l;0;0])*trotz(q2) * transl([l;0;0])*trotz(q3) *
transl([l;0;0])
T = simplify(T(1:2,4))
q = [q1;q2;q3];
J = simplify(jacobian(T,q))
nu_J = subs(J, [q1,q2,q3],[pi/4,-pi/2,pi/2])

%拉格朗日方程求解
syms m I
syms q1 q2 q3 dq1 dq2 dq3 ddq1 ddq2 ddq3
syms g
Xc1 = rotz(q1) * [l/2; 0; 0]
Xc2 = rotz(q1) * [l; 0; 0] + rotz(q1+q2) * [l/2; 0; 0]
Xc3 = rotz(q1) * [l; 0; 0] + rotz(q1+q2) * [l; 0; 0] + rotz(q1+q2+q3) *
[l/2; 0; 0]
q = [q1;q2;q3];
dq = [dq1;dq2;dq3];
Vc1 = jacobian(Xc1,q)*dq;
Vc2 = jacobian(Xc2,q) * dq;
Vc3 = jacobian(Xc3,q) * dq;
K1 = (1/2)*Vc1.'*m*Vc1 + (1/2)*I*dq1^2;
K2 = (1/2)*Vc2.'*m*Vc2 + (1/2)*I*(dq1+dq2)^2;
%nu_k2 = simplify(subs(K2,[q1,q2,q3,I],[pi/4,-pi/2,pi/2,m*l*l/12]))
K3 = (1/2)*Vc3.'*m*Vc3 + (1/2)*I*(dq1+dq2+dq3)^2;
K = K1 + K2 + K3;
P1 = m*g*l/2*sin(q1);
```

```
P2 = m*g*(l*sin(q1)+l/2*sin(q1+q2));
P3 = m*g*(l*sin(q1)+l*sin(q1+q2)+l/2*sin(q1+q2+q3));
P = P1 + P2 + P3;

L = K - P

ddq = [ddq1;  ddq2; ddq3];

dLddq = [jacobian(L,dq1);jacobian(L,dq2);jacobian(L,dq3)];
ddLdqddt = jacobian(dLddq,q)*dq + jacobian(dLddq,dq)* ddq;
dLdq =  [jacobian(L,q1);jacobian(L,q2);jacobian(L,q3)];

% 计算H,C,G
tau = simplify(ddLdqddt - dLdq);
H = simplify(jacobian(tau, ddq))
C = simplify(jacobian(tau, dq))
G = simplify(jacobian(tau, g)*g)
numerical_H = subs(H,[q1,q2,q3,dq1,dq2,dq3],[pi/4,-pi/2,pi/2,0,0,0])
new_numerical_H = subs(numerical_H,[I],[1/12*m*l*l])
numerical_C = subs(C,[q1,q2,q3,dq1,dq2,dq3],[pi/4,-pi/2,pi/2,0,0,0])
numerical_G = subs(G,[q1,q2,q3,dq1,dq2,dq3],[pi/4,-pi/2,pi/2,0,0,0])

%计算关节空间加速度,可以更换权重矩阵W
syms ddp1 ddp2
%A = [1 0 0;1 1 0; 1 1 1]
%W = A.' * A
W = H
%W = eye(3)
ddp = [ddp1;ddp2];
robot_ddq = inv(W)*J.'*inv(J*inv(W)*J.')*ddp
numerical_robot_ddq = simplify(subs(robot_ddq,
[q1,q2,q3,dq1,dq2,dq3,ddp1,ddp2,I],[pi/4,-pi/2,pi/2,0,0,0,1,0,m*l*l/12]))
driven_tau = H * robot_ddq + C * dq + G
numerical_driven_tau = subs(driven_tau, [q1,q2,q3,dq1,dq2,dq3,ddp1,ddp2],
[pi/4,-pi/2,pi/2,0,0,0,1,0])
new_numerical_driven_tau = subs(numerical_driven_tau,[I],[1/12*m*l*l])

%关节驱动力矩最小化求解
new_W = H.' * H
new_ddq = inv(new_W)*J.'*inv(J*inv(new_W)*J.')*(ddp+J*inv(new_W)*H.'*G)-
inv(new_W)*H.'*G;
numerical_new_ddq = simplify(subs(new_ddq,
[q1,q2,q3,dq1,dq2,dq3,ddp1,ddp2,I],[pi/4,-pi/2,pi/2,0,0,0,1,0,m*l*l/12]))
new_new_tau = H * new_ddq + G;
numerical_new_new_tau = simplify(subs(new_new_tau,
[q1,q2,q3,dq1,dq2,dq3,ddp1,ddp2,I],[pi/4,-pi/2,pi/2,0,0,0,1,0,m*l*l/12]))
```