

# 目 录

符号和缩略语说明	2
第 1 章 机械狗的运动模型	1
1.1 机械狗运动模型描述	1
1.2 质心运动 (CoM Motion) 问题表述	2
第 2 章 质心运动优化	4
2.1 质心运动优化	4
2.1.1 成本函数：最小化运动规划结果的加速度	4
2.1.2 等式约束：运动曲线段节点连续性	5
2.1.3 不等约束：基于 ZMP	6
2.2 落脚点优化：倒立摆模型	7
2.2.1 成本函数	8
2.2.2 不等约束：不可达落脚点	8
第 3 章 分层次优化计算	10
3.1 分层次优化计算	10
3.1.1 运动方程	10
3.1.2 接触部分运动约束	10
3.1.3 接触力和扭矩限制	10
3.1.4 运动跟随	11
3.1.5 接触力最小化	11
3.1.6 计算电机扭矩	11
3.2 支撑多边形序列生成	12
第 4 章 XXXxx	13
第 5 章 参考文献	14
结 论	15
参考文献	16
附录 A 补充内容	17

## 符号和缩略语说明

$\kappa$	传输系数 (Transmission Coefficient)
$\nu_i$	虚频 (Imaginary Frequency)

## 第 1 章 机械狗的运动模型

### 1.1 机械狗运动模型描述

机器人与环境接触的机械系统的运动模型描述方程可以描述如下<sup>[1]</sup>:

$$\mathbf{M}(\mathbf{q})\dot{\mathbf{u}} + \mathbf{h}(\mathbf{q}, \mathbf{u}) = \mathbf{S}^T \boldsymbol{\tau} + \mathbf{J}_S^T \boldsymbol{\lambda} \quad (1-1)$$

这其中  $\mathbf{q}$  是一个描述机器人主体及各个节点的广义位置矢量:

$$\mathbf{q} = \begin{bmatrix} {}_I \mathbf{r}_{IB} \\ \mathbf{q}_{IB} \\ \mathbf{q}_j \end{bmatrix} \in SE(3) \times \mathbb{R}^{n_j} \quad (1-2)$$

它里面  ${}_I \mathbf{r}_{IB} \in \mathbb{R}^3$  是机器人主体相对于惯性系的三维位置矢量;  $\mathbf{q}_{IB} \in SO(3)$  是机器人主体相对于惯性系的转动描述, 用哈密顿单位四元数表示的;  $\mathbf{q}_j \in \mathbb{R}^{n_j}$  是一个储存机器人所有节点角度的  $n_j$  维矢量。

**Notes 1.1.1.** 不过我还是不太清楚  $\mathbf{q}$  也就是  $SE(3) \times \mathbb{R}^{n_j}$  到底是一个什么形状的矩阵? 单纯看表达的话他应该是一个  $3 + 4 + n_j$  维的矢量, 包含了整个机器人的所有位置有关的状态信息。它似乎不用直接参与运算, 因此可以先放一放, 重要的是  $SE(3) \times \mathbb{R}^{n_j}$  的含义到底是什么需要好好理解一下。

这其中  $\mathbf{u}$  是一个描述机器人主体及各个节点的广义速度矢量:

$$\mathbf{u} = \begin{bmatrix} {}_I \mathbf{v}_B \\ {}_B \boldsymbol{\omega}_{IB} \\ \dot{\mathbf{q}}_j \end{bmatrix} \in \mathbb{R}^{n_u} \quad (1-3)$$

它里面的  ${}_I \mathbf{v}_B$  描述了机器人主体相对于惯性系的速度;  ${}_B \boldsymbol{\omega}_{IB}$  描述了机器人主体相对于自身的角速度;  $\dot{\mathbf{q}}_j$  描述了机器人的各个关节转动的速度。这其中  $\mathbf{M}$  是一个关于机器人整体的质量矩阵, 它是一个  $n_u \times n_u$  的矩阵:

$$\mathbf{M} \in \mathbb{R}^{n_u \times n_u} \quad (1-4)$$

这个质量矩阵的具体数值跟机器人的机械系统的状态(各个节点的位置  $\mathbf{q}$ ) 相关, 可以通过通用的方式计算出它的表达式。实际计算的时候只需要带入  $\mathbf{q}$  的值, 就可以计算出  $\mathbf{M}$  矩阵的各个元素具体数值。这其中  $\mathbf{h}$  是一个跟机器人的机械位置

和速度都有关的量，包含了机械系统产生的克里奥利力、离心力和重力的作用，它是一个  $n_u$  维的矢量：

$$\mathbf{h} \in \mathbb{R}^{n_u} \quad (1-5)$$

**Notes 1.1.2.** 这个  $\mathbf{h}$  的具体计算方式我现在还不是很清楚。

这其中  $\mathbf{S}$  是一个选择矩阵，可以用来选择整个公式中哪些自由度被激活，它是一个  $n_\tau \times n_u$  的矩阵：

$$\mathbf{S} = \begin{bmatrix} 0_{n_\tau \times (n_u - n_\tau)} & \mathbb{I}_{n_\tau \times n_\tau} \end{bmatrix} \in \mathbb{R}^{n_\tau \times n_u} \quad (1-6)$$

它其中包含的参数  $n_\tau$  表示被激活的自由度数量，如果机器人的所有自由度都被激活，则  $n_\tau = n_j$ 。这其中  $\boldsymbol{\tau}$  是机器人各个关节的电机提供的扭矩，它是一个  $n_j$  维的向量：

$$\boldsymbol{\tau} \in \mathbb{R}^{n_j} \quad (1-7)$$

它的节点成员是否产生作用受到  $\mathbf{S}^T$  矩阵的选择。这其中  $\mathbf{J}_S$  是一些列雅可比矩阵的集合：

$$\mathbf{J}_S = \begin{bmatrix} J_{C_1}^T & \dots & J_{C_{n_c}}^T \end{bmatrix}^T \in \mathbb{R}^{3n_c \times n_u} \quad (1-8)$$

它是接触点的支撑力  $\boldsymbol{\lambda}$  向节点力转换的矩雅可比矩阵，包含了  $n_c$  个雅可比矩阵， $n_c$  为接触地面的肢体个数。

## 1.2 质心运动 (CoM Motion) 问题表述

每一个坐标方向的质心运动规划都被描述成一个系列的五次样条。比如沿着  $x$  方向的其中第  $i$ -th 曲线可以描述为：

$$\begin{aligned} x(t) &= a_{i5}^x t^5 + a_{i4}^x t^4 + a_{i3}^x t^3 + a_{i2}^x t^2 + a_{i1}^x t + a_{i0}^x \\ &= \begin{bmatrix} t^5 & t^4 & t^3 & t^2 & t & 1 \end{bmatrix} \\ &\quad \cdot \begin{bmatrix} a_{i5} & a_{i4} & a_{i3} & a_{i2} & a_{i1} & a_{i0} \end{bmatrix} \\ &= \boldsymbol{\eta}^T(t) \boldsymbol{\alpha}_i^x \end{aligned} \quad (1-9)$$

这其中  $t \in [t, t + \Delta t_i]$  是第  $i$  个曲线段前所有  $(i-1)$  个曲线持续时间长度的总和， $\Delta t_i$  是第  $i$  个曲线持续的时长。基于 (2) 的关于位置的表述，可以很容易地将关于速度和加速度的表述写出来：

$$\dot{x}(t) = \dot{\boldsymbol{\eta}}^T(t) \boldsymbol{\alpha}_i^x \quad (1-10)$$

$$\ddot{x}(t) = \dot{\boldsymbol{\eta}}^T(t) \boldsymbol{\alpha}_i^x \quad (1-11)$$

这其中：

$$\dot{\boldsymbol{\eta}}^T(t) = \begin{bmatrix} 5t^4 & 4t^3 & 3t^2 & 2t^1 & 1 & 0 \end{bmatrix}^T \quad (1-12)$$

$$\ddot{\boldsymbol{\eta}}^T(t) = \begin{bmatrix} 20t^3 & 12t^2 & 6t^1 & 2 & 0 & 0 \end{bmatrix}^T \quad (1-13)$$

对于质心在  $y, z$  方向上的描述是一样的。每一条质心曲线都由  $x, y, z$  三部分分量  $\boldsymbol{\alpha}_i = [\boldsymbol{\alpha}_i^x \quad \boldsymbol{\alpha}_i^y \quad \boldsymbol{\alpha}_i^z]^T$  组成，可以将  $n_s$  条质心曲线的  $3n_s$  条曲线参数写到一起，优化的参数矢量可以写成： $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_0^T \quad \dots \quad \boldsymbol{\alpha}_i^T \quad \dots \quad \boldsymbol{\alpha}_{n_s}^T]^T$ 。这样一来，质心的位置可以表示为：

$$\boldsymbol{p}_{CoM}(t) = \boldsymbol{T}(t) \boldsymbol{\alpha}_i \in \mathbb{R}^3, \quad \boldsymbol{T}(t) = \begin{bmatrix} \boldsymbol{\eta}^T(t) & 0 & 0 \\ 0 & \boldsymbol{\eta}^T(t) & 0 \\ 0 & 0 & \boldsymbol{\eta}^T(t) \end{bmatrix} \quad (1-14)$$

同样质心的速度和位置也可以得到了：

$$\dot{\boldsymbol{p}}_{CoM}(t) = \dot{\boldsymbol{T}}(t) \boldsymbol{\alpha}_i \in \mathbb{R}^3 \quad (1-15)$$

$$\ddot{\boldsymbol{p}}_{CoM}(t) = \ddot{\boldsymbol{T}}(t) \boldsymbol{\alpha}_i \in \mathbb{R}^3 \quad (1-16)$$

## 第 2 章 质心运动优化

### 2.1 质心运动优化

机械狗的质心运动天然受到多种条件约束。因此，我们将质心运动规划问题描述为一个非线性优化问题，它的目标是在相等约束  $\mathbf{c}(\xi)$  和不等约束  $\mathbf{h}(\xi)$  的条件下最小化一个通用的成本函数  $f(\xi)$ 。这个问题的数值方法被称作为顺序二次规划算法 ((SQP) algorithm)，这个算法要求计算约束条件和成本函数的雅可比和海森矩阵。这个优化算法会以局部运动的踱步周期  $\tau$  为时间间隔，不断地重新计算新的结果。下面的内容将介绍一些用到的成本函数和约束条件。

#### 2.1.1 成本函数：最小化运动规划结果的加速度

通过计算样条段的二次成本可以得到：

$$\mathbf{Q}_k^{acc} = \begin{bmatrix} (400/7)\Delta t_k^7 & 40\Delta t_k^6 & 24\Delta t_k^5 & 10\Delta t_k^4 & & \\ & 40\Delta t_k^6 & 28.8\Delta t_k^5 & 18\Delta t_k^4 & 8\Delta t_k^3 & \\ & 24\Delta t_k^5 & 18\Delta t_k^4 & 12\Delta t_k^3 & 6\Delta t_k^2 & 0_{4 \times 2} \\ & 10\Delta t_k^4 & 8\Delta t_k^3 & 6\Delta t_k^2 & 4\Delta t_k & \\ & & 0_{2 \times 4} & & & 0_{2 \times 2} \end{bmatrix} \quad (2-1)$$

相应的  $\mathbf{c}_k^{acc} = 0$ 。  $\Delta t_k$  是第  $k$  个曲线段以秒为单位的持续时间。  $\mathbf{Q}_k^{acc}$  是通过平方和积分 CoM 在第  $k$  样条的持续时间上的加速度得到。  $\mathbf{Q}_k^{acc}$  被添加为整个成本函数的 Hessian 的子矩阵。

**Notes 2.1.1.** ( $\mathbf{Q}_k^{acc}$  到底是成本函数还是 Hessian 矩阵？它们之间有什么关系？有必要了解一下 Hessian 矩阵去。)

最终期望的位置计算为参考高级线性速度  $\mathbf{v}_{ref}$  和角速度  $\omega_{ref}$  的  $z$  分量和优化时间间隔  $\tau$  的函数。在以上参数相对于优化时间间隔  $\tau$  变化很缓慢的情况下，最后的位置可以计算为：

$$\mathbf{p}_{final} = \mathbf{p}_{init} + \mathbf{R}(\tau\hat{\omega}_z)(\tau\mathbf{v}_{ref}) \quad (2-2)$$

这里面第一项  $\mathbf{p}_{init}$  是前一时刻的位置，第二项是当前时刻的位移  $\tau\mathbf{v}_{ref}$  乘上一个旋转矩阵  $\mathbf{R}(\tau\hat{\omega}_z) = \begin{bmatrix} 0 & 0 & \omega_{ref_z} \end{bmatrix}^T$ 。路径正则化  $\pi$ 。  $\pi$  的初始位置计算为优化视界  $\tau$  上平均的足迹中心，假设脚要么接触，要么以恒定的速度移动。最终位置

是通过参考高级速度计算的，如参考文献（7）所示。

我们将初始速度设置为参考速度  $v_{ref}$ ，最后一个速度与运动计划结束时预测的躯干方向对齐。 $\pi$  的初始和最终加速度设置为零。 $\pi$  的初始和最终高度设置为用户指定的参考值。（这个  $\pi$  到底是个啥？运动规划的高级计算近似结果，比如用传感器得到的轨迹推算出来的结果？）我们通过添加以下形式的成本  $\lambda_{lin}\epsilon_z + \lambda_{quad}\epsilon_z^2$  来限制沿  $z$  轴超调。同时对原有描述扩充以下几个约束：

$$p_{CoM}^z(t) - \pi_z(t) \leq \epsilon_z \quad (2-3)$$

$$p_{CoM}^z(t) - \pi_z(t) \geq -\epsilon_z \quad (2-4)$$

$$\epsilon_z \geq 0 \quad (2-5)$$

这里面  $\epsilon$  是一个松弛变量，它会被添加入优化参数  $\xi$  中。由于时间依赖性，需要对轨迹进行采样，其中每个采样点引入两个额外的不等式约束。（为什么？）

### 2.1.2 等式约束：运动曲线段节点连续性

对于  $x$  方向上的第  $k$  段和第  $k+1$  段曲线的连续性要求可以写作：

$$\begin{bmatrix} \eta(t_{fk})^T & -\eta(0)^T \\ \dot{\eta}(t_{fk})^T & -\dot{\eta}(0)^T \end{bmatrix} \begin{bmatrix} \alpha_k^x \\ \alpha_{k+1}^x \end{bmatrix} = 0 \quad (2-6)$$

这里面  $t_{fk}$  代表曲线  $s_k$  以秒为单位的持续时间长度。同样的方法，可以将  $y, z$  方向上的连续性要求写出来。节点连续性在涉及到悬空自由落体过程时需要被另外对待：起飞时的质心的动力学可以描述为： $\ddot{p}_{CoM} = g$ ，其中  $g$  是重力加速度矢量。将上式积分可以得到：

$$\dot{p}(t) = gt + \dot{p}_{CoM}(0) \quad (2-7)$$

$$p(t) = \frac{1}{2}gt^2 + \dot{p}_{CoM}(t) + p_{CoM}(0)(0) \quad (2-8)$$

着陆时的质心动力学可以描述为：

$$\ddot{T}(0)\alpha_{i+1} = g \quad (2-9)$$

$$\dot{T}(0)\alpha_{i+1} = gt_f + \dot{T}(t_i)\alpha_i \quad (2-10)$$

$$T(0)\alpha_{i+1} = \frac{1}{2}gt_f^2 + [\dot{T}(t_i)t_f + T(t_i)]\alpha_i \quad (2-11)$$

这里面  $t_i$  表示第  $i$  条曲线段的持续时间。如果第一个或最后一个支持多边形对应于一个完整的飞行阶段，则可以在初始和最终条件下找到类似的替换。

**Notes 2.1.2.** 这句话的含义是什么？实践上应该怎样应用？

### 2.1.3 不等约束：基于 ZMP

**Notes 2.1.3.** 这部分我还是看不太懂，这家伙已经不用购物车模型来近似计算 ZMP 了，还要再去看看 ZMP 的知识。

#### 2.1.3.1 ZMP 的计算

$$\mathbf{p}_{ZMP} = \frac{\mathbf{n} \times \mathbf{M}_O^{gi}}{\mathbf{n}^T \mathbf{F}^{gi}} \quad (2-12)$$

这里面  $\mathbf{M}_O^{gi}$  和  $\mathbf{F}^{gi}$  是重力-惯性力螺旋的组成部分，它们的计算方式如下：

$$\mathbf{M}_O^{gi} = m \cdot \mathbf{p}_{CoM} \times (\mathbf{g} - \ddot{\mathbf{p}}_{CoM}) - \dot{\mathbf{L}} \quad (2-13)$$

$$\mathbf{F}^{gi} = m \cdot \mathbf{g} - \dot{\mathbf{P}} \quad (2-14)$$

这里面  $m, \mathbf{P}, \mathbf{L}$  分别是质心的质量、线性动量和角动量。因为我们将不会针对转动及其延伸进行优化，因此下面的计算中角动量近似为零  $\dot{\mathbf{L}} = 0$ 。

#### 2.1.3.2 引入的不等约束

基本要求是 ZMP 点要被约束在机械狗的支撑多边形内部，这给出以下形式的不等约束：

$$\mathbf{h}_{ZMP} = \mathbf{d}^T \mathbf{p}_{ZMP} + r \geq 0 \quad (2-15)$$

这其中  $\mathbf{d}^T = [p \ q \ 0]$  和  $r$  是描述支撑多边形的边的系数。接下来将 (12) 代入 (14) 后可以得到：

$$\begin{aligned} & \mathbf{d}^T \mathbf{S}(\mathbf{n}) \mathbf{M}_O^{gi} + r \mathbf{n}^T \mathbf{F}^{gi} \\ &= \mathbf{d}^T \mathbf{S}(\mathbf{n}) [m \cdot \mathbf{p}_{CoM} \times (\mathbf{g} - \ddot{\mathbf{p}}_{CoM}) - \dot{\mathbf{L}}] + r \mathbf{n}^T (m \cdot \mathbf{g} - \dot{\mathbf{P}}) \\ & \stackrel{\dot{\mathbf{L}} \approx 0, \dot{\mathbf{P}} = m \ddot{\mathbf{p}}_{CoM}}{=} m \mathbf{d}^T \mathbf{S}(\mathbf{n}) \mathbf{p}_{CoM} \times (\mathbf{g} - \ddot{\mathbf{p}}_{CoM}) + m r \mathbf{n}^T (\mathbf{g} - \ddot{\mathbf{p}}_{CoM}) \geq 0 \\ & \rightarrow \mathbf{d}^T \mathbf{S}(\mathbf{n}) \mathbf{S}(\mathbf{p}_{CoM}) (\mathbf{g} - \ddot{\mathbf{p}}_{CoM}) + r \mathbf{n}^T (\mathbf{g} - \ddot{\mathbf{p}}_{CoM}) \geq 0 \end{aligned} \quad (2-16)$$

这里面定义了一个斜对称矩阵  $\mathbf{S}(\mathbf{a})$  用来实现将矩阵叉乘转化为点乘的效果（这个方法优惠什么标准依据？）。它通过计算一个可以实现以下效果的方程得到： $\mathbf{S}(\mathbf{a})\mathbf{b} = \mathbf{a} \times \mathbf{b}$ 。用 (15) 也可以计算出 (14) 相对于质心位置  $\mathbf{p}_{CoM}$  和质心加速度



$\ddot{\mathbf{p}}_{CoM}$  的梯度（为什么要求这个梯度？怎么求得？）：

$$\nabla \mathbf{h}_{ZMP} = \begin{bmatrix} \mathbf{\Gamma} \cdot (\ddot{\mathbf{p}}_{CoM} - \mathbf{g}) \\ -\mathbf{\Gamma} \cdot \mathbf{p}_{CoM} - r\mathbf{n} \end{bmatrix} \quad (2-17)$$

这里面定义  $\mathbf{\Gamma} = \mathbf{S}(\mathbf{S}^T(\mathbf{n})\mathbf{d})$ 。同样，它的 Hessian 矩阵计算为：

$$\nabla^2 \mathbf{h}_{ZMP} = \begin{bmatrix} 0 & \mathbf{\Gamma}^T \\ -\mathbf{\Gamma}^T & 0 \end{bmatrix} \quad (2-18)$$

这个 Hessian 矩阵是一个反对称矩阵，因此可以从优化问题中去除。（为什么呢？Hessian 矩阵怎么计算的也得去了解一下。）

**Notes 2.1.4.** 这部分不是很明白。为什么对问题描述添加那样的项就可以软化不等约束？具体的表现项是与软化参数之间的对应什么？

我们软化了初始  $n_{ineq}$  个样本的不等式约束，其中  $n_{ineq}$  是用户设置的调整参数。为了实现这一点，我们在优化参数  $\xi$  中添加松弛变量  $\xi_{ineq}$ 。这样一来，问题的表述会增加  $\lambda_{lin}\xi_{ineq} + \lambda_{quad}\xi_{ineq}^2$ ，同时添加下面两个不等约束  $\mathbf{c}_{ineq} \geq -\xi_{ineq}$ ,  $\xi_{ineq} \geq 0$ ，其中  $\mathbf{c}_{ineq}$  是如 (15) 中描述的前  $n_{ineq}$  个约束条件。从物理的角度来看，松弛相当于一个可变大小的支撑多边形，不能小于标称多边形。

### 2.1.3.3 分配一个新的规划：搜索算法

下面讨论一下如何将一个新得到的运动规划添加到一个之前已经在执行的运动规划上，避免两者的冲突。为此，我们首先存储求解器优化所需的计算时间  $t_c$ 。我们使用它作为初始猜测来搜索最接近当前测量值的运动规划中的位置，以便过渡到新计划是一个平滑的。为此，我们通过写作来解决线性搜索问题：

$$t = \arg \min_{\mathbf{w}} \|\mathbf{p}(t) - \mathbf{p}_{meas}\|_2^2 \quad (2-19)$$

这里面  $\mathbf{W}$  是一个正定权重矩阵  $\mathbf{p}_{meas}$  是完成优化动作后的测量得到的质心位置。

## 2.2 落脚点优化：倒立摆模型

建立如下二次规划问题：

$$\min_{\xi} \quad \frac{1}{2} \xi^T \mathbf{Q} \xi + \mathbf{c}^T \xi \quad \text{s.t.} \quad \mathbf{D} \xi \leq \mathbf{f} \quad (2-20)$$

这里面  $\xi \in \mathbb{R}^{2n_{feet}}$  是裸足点  $\mathbf{p}_{fi}$  的  $x, y$  方向分量，其中  $i = 1, \dots, n_{feet}$ ,  $n_{feet} = 4$  是机器所有脚的总数。与 CoM 运动规划器所做的类似，我们并行优化主控制回路。

因此，每当新的优化准备好时，我们都会更新立足点计划。

### 2.2.1 成本函数

相对于默认的战力姿势配置，用户可以自定义一个落脚点位置。这个可以解释为落脚点优化问题的一个正则化项。

**Notes 2.2.1.**（正则化的概念需要熟悉一下？）

为了跟踪默认落脚点的位置  $\mathbf{p}_{ref_i}$  我们为成本函数 (19) 添加下述表述：

$$\mathbf{p}_{ref_i} = \mathbf{W}_{def}, \quad \mathbf{c}_{def_i} = -\mathbf{W}_{def}^T \mathbf{p}_{ref_i} \quad (2-21)$$

默认立足点的选择将影响当所有腿与环境接触时足迹的程度。虽然这可能会使小跑步态对干扰更加健壮，但它会在较慢的步行步态中产生更广泛的横向运动。（不知所云？）在实践中，我们在 ANYmal 的实验中可靠地工作的默认立足点位置计算为臀部到地形的垂直投影。

为了跟踪驱动整个运动框架的平均高级速度，我们惩罚与立足点位置的偏差。这些是在优化视界的持续时间内实现恒定速度来计算的。为了避免参考立足点中的跳跃，我们还为当前解决方案和先前计算的解决方案之间的距离设置了成本。

最后，我们为成本函数添加一个稳定项，它是一个倒立摆模型的函数。如参考文献 10 中所述，这个函数用  $\omega(\mathbf{v}_{ref} - \mathbf{v}_{hips} \sqrt{h/g})$  计算第  $k$  个落脚点。这其中， $\omega$  是一个正的权重标量， $\mathbf{v}_{ref}$  是高级速度参考， $\mathbf{b}_{hips}$  是第  $k$  个臀部的速度， $h$  是第  $k$  个臀部到地面的高度， $g$  是加速度常量。

**Notes 2.2.2.** 这引导了去关注倒立摆模型的建模？

### 2.2.2 不等约束：不可达落脚点

为了避免计算违反腿运动学扩展的立足点，我们通过在每个立足点的可行位置上添加不等式约束来利用 QP 设置。我们通过考虑腿的最大可伸展长度和测量的臀部与地面的高度关系来完成这个约束。将臀部的位置投影到地形平面上记作  $\mathbf{h}_0$ ，我们设置了描述一个多边形的不等式，该多边形具有均匀分布在  $\mathbf{h}_0$  附近的  $n_p$  个顶点。多边形的各个顶点到臀部投影在地形的点  $\mathbf{h}_0$  的距离计算为：

$$\sqrt{l_{max}^2 - h_i^2} \quad (2-22)$$

**Notes 2.2.3.** 这就是一个直角三角形，斜边长度为腿的最大伸展长度  $l_{max}$ ，高为臀部到地形的投影高度  $h_i$ ，计算落脚点的可行范围，也就是底边长度。

## 第 3 章 分层次优化计算

### 3.1 分层次优化计算

$\xi_d$  下面的控制方法采用接触力控制

#### 3.1.1 运动方程

$$\begin{bmatrix} \mathbf{M}_{fb} & -\mathbf{J}_{sfb}^T \end{bmatrix} \xi_d = -\mathbf{h}_{fb} \quad (3-1)$$

**Notes 3.1.1.**  $XXX_{fb}$ : 下标的意思是浮动的主体-‘floating base’

- (1)  $\xi_d$  是一个一个  $n_u + n_c$  行 1 列的向量:  $\xi_d = \begin{bmatrix} \mathbf{u}_d^T & \boldsymbol{\lambda}_d^T \end{bmatrix} \in \mathbb{R}^{n_u+n_c}$ , 其中:
  - ①  $\mathbf{u}_d^T$  是目标节点的加速度;
  - ②  $\boldsymbol{\lambda}_d^T$  是目标接触力;
- (2)  $\mathbf{M}_{fb}$  是复合型惯性矩阵的前六行; 啥是复合惯性矩阵?
- (3)  $\mathbf{J}_{sfb}^T$  是雅克比阵的前六行, 它将接触力转换到节点的扭矩;
- (4)  $\mathbf{h}_{fb}$  是非线性项的前六行, 包括克里奥力、离心力和重力项;

#### 3.1.2 接触部分运动约束

控制器找到的解决方案不应违反 (2) 中定义的接触约束。因此, 我们通过设置在接触点施加空加速度:

$$\begin{bmatrix} \mathbf{J}_s & \mathbf{0}_{3n_c \times 3n_c} \end{bmatrix} \xi_d = -\dot{\mathbf{j}}_s \mathbf{u} \quad (3-2)$$

哦 它将  $\xi_d$  带进去计算后就得到了:  $\mathbf{J}_s \xi_d = -\dot{\mathbf{j}}_s \mathbf{u}$  其实就是公式 (2) 的第二个式子  $\mathbf{J}_s \xi_d + \dot{\mathbf{j}}_s \mathbf{u} = 0$ 。

#### 3.1.3 接触力和扭矩限制

$$({}_I \mathbf{h} - {}_I \mathbf{n}_\mu)^T {}_I \boldsymbol{\lambda}_k \leq 0 \quad (3-3)$$

$$-({}_I \mathbf{h} + {}_I \mathbf{n}_\mu)^T {}_I \boldsymbol{\lambda}_k \leq 0 \quad (3-4)$$

$$({}_I \mathbf{l} - {}_I \mathbf{n}_\mu)^T {}_I \boldsymbol{\lambda}_k \leq 0 \quad (3-5)$$

$$-(l + l_{\mu})^T l_{\lambda_k} \leq 0 \quad (3-6)$$

$l_n$  是接触面的法向量； $\mu$  是摩擦系数；有了这两个再乘上受力  $l_{\lambda}$ ，就可以得出最大静摩擦力  $l_{\mu}^T \lambda$ 。实际在接触点平行于地面方向的两个分力  $l_h$ ， $l$  在正反方向上都不应该比这个值大，否则就会发生滑动。这就是公式 5 约束的由来。

$$\tau_{min} - h_j \leq [M_j \quad -J_{s_j}^T] \leq \tau_{max} - h_j \quad (3-7)$$

这里的  $h_j$  应该就是科里奥利力那一堆东西。而  $[M_j \quad -J_{s_j}^T]$  就是加速度力  $M_j \dot{u}_d^T$  和传递力  $-J_{s_j}^T \lambda_d^T$  两项的和。它们计算的结果就是电机提供的扭矩力  $\tau$  克服完  $h(q, u)$  剩下的力。

### 3.1.4 运动跟随

为了能跟随浮动主体和摆动腿的目标运动。我们通过实现具有前馈参考加速度和运动相关状态反馈状态的操作空间控制器来约束关节加速度。对于主体的线性运动：

$$[cJ_P \quad 0] \xi_d = {}_c \ddot{r}_{IB}^d + k_D^{pos} ({}_c \dot{r}_{IB}^d - {}_c v_B) + k_P^{pos} ({}_c r_{IB}^d - {}_c r_B) \quad (3-8)$$

对于主体的角度运动：

$$[cJ_R \quad 0] \xi_d = -k_D^{ang} {}_c \omega_B + k_P^{ang} (q_{CB}^d \Xi q_{CB}) \quad (3-9)$$

\* 雅可比矩阵  $cJ_P$  和  $cJ_R$  是与‘控制坐标系 C（这是一个与地形局部估计和机器人航向方向对齐的帧）’中表达的基相关的平移和旋转雅可比矩阵。

\*  $\Xi$  这个算子产生欧拉向量，表示期望姿态  $q_{CB}^d$  和估计姿态  $q_{CB}$  之间的相对方向。

\* 这里面  $k_P^{pos}, k_D^{pos}, k_P^{ang}, k_D^{ang}$  是用来控制增益的对角正定矩阵。

\* 参考的运动  ${}_c r_{IB}$  和它的导数是运动规划的结果。

### 3.1.5 接触力最小化

可以通过下面的方法将接触力设置为最小值：

$$\begin{bmatrix} 0_{3n_c \times n_u} & \mathbb{I}_{3n_c \times 3n_c} \end{bmatrix} \xi_d = 0 \quad (3-10)$$

### 3.1.6 计算电机扭矩

如果给定了一个优化的节点运动和接触力， $\xi_d = [\dot{u}_d^T \quad \lambda_d^T]^T$  我们可以用以下公式计算各个电机的扭矩：

$$\tau_d = [M_j \quad -J_{s_j}^T] \xi_d + h_j$$

其中  $\mathbf{M}_j, -\mathbf{J}_{sj}^T, \mathbf{h}_j$  在公式 (6) 中定义。

**Notes 3.1.2.** 因此，所有规划的目的就是给出节点的运动加速度  $\dot{\mathbf{u}}_d^T$  和接触力  $\lambda_d^T$ 。

## 3.2 支撑多边形序列生成

> 这部分的详细描述要参考文献 2。

## 第 4 章 XXxxx

## 第 5 章 参考文献



## 结 论

学位论文的结论作为论文正文的最后一章单独排写，但不加章标题序号。

结论应是作者在学位论文研究过程中所取得的创新性成果的概要总结，不能与摘要混为一谈。博士学位论文结论应包括论文的主要结果、创新点、展望三部分，在结论中应概括论文的核心观点，明确、客观地指出本研究内容的创新性成果（含新见解、新观点、方法创新、技术创新、理论创新），并指出今后进一步在本研究方向进行研究工作的展望与设想。对所取得的创新性成果应注意从定性和定量两方面给出科学、准确的评价，分（1）、（2）、（3）…条列出，宜用“提出了”、“建立了”等词叙述。

## 参考文献

- [1] 张昆, 冯立群, 余昌钰, 等. 机器人柔性手腕的球面齿轮设计研究[J]. 清华大学学报: 自然科学版, 1994, 34(2): 1-7.

## 附录 A 补充内容

在附录里面我会写一些控制相关的一些数学概念的理解。

### A.1 拉格朗日乘子法

在机械狗的运动规划中，机械狗质心的运动是通过求解空间中  $x, y, z$  方向上的五次曲线得到的。这个五次曲线是在一些列约束条件下的最优运动结果。这个结果是通过求解 QP(二次优化) 问题得到的。实际上所谓的 QP 问题的一种形式就是之前高数上很熟悉的‘拉格朗日乘子法 (Lagrange Multiplier)’。下面就对这部分知识进行总结，这部分内容主要参考[这个博客](#)。

#### A.1.1 优化问题

通常我们需要求解的优化问题有如下几类：

- (1) 无约束优化问题，可以写为：

$$\min f(x);$$

- (2) 有等式约束的优化问题，可以写为：

$$\begin{aligned} \min f(x); \\ \text{s.t. } h_i(x) = 0, i = 1, \dots, n; \end{aligned}$$

- (3) 有等式和不等式约束的优化问题，可以写为：

$$\begin{aligned} \min f(x); \\ \text{s.t. } g_i \leq 0, i = 1, \dots, n; \\ \text{and } h_i(x) = 0, i = 1, \dots, n; \end{aligned}$$

对于第一类优化问题，常使用的求解方式是 *Fermat* 定理，即使用求取  $f(x)$  的导数，然后令其为零，可以求得候选的最优值，再在这些候选值中验证；如果是凸函数，可以保证是最优解。

对于第二类优化问题，常使用的求解方式是拉格朗日乘子法，即把等式约束  $h_i(x)$  用一个系数与  $f(x)$  写为一个式子，称为‘拉格朗日函数’，而系数称为拉格朗日乘子。通过拉格朗日函数对各个变量求导，令其为零，可以求得候选值集合，然后验证求得最优值。

对于第三类优化问题，常使用的求解方式是 *KKT* 条件。同样地，把所有等式、不等式约束条件与  $f(x)$  写成一个式子，也叫拉格朗日函数，系数也称为拉格朗日乘子。通过一些条件可以求出最优值的必要条件，这个条件称为 *KKT* 条件。

### A.1.2 拉格朗日乘子法

对于等式约束，我们可以通过一个拉格朗日系数  $a$  把等式约束和目标函数组合成一个拉格朗日函数：

$$L(a, x) = f(x) + ah(x) \quad (A-1)$$

其中  $a, h(x)$  分别为一行向量和列向量。最优值可以通过对拉格朗日函数  $L(a, x)$  的各个参数求导取零，联立等式进行求取。它要求：

- (1) 拉格朗日函数对  $a$  的偏导为零，即  $\frac{\partial L(a, x)}{\partial a} = 0$ ;
- (2) 拉格朗日函数对  $x$  的偏导为零，即  $\frac{\partial L(a, x)}{\partial x} = 0$ ;

通过联立求解上面两个条件等式可以求得候选的最优值结果  $a, x$  的候选值，然后验证  $x$  求得最优值（如果是如函数可以保证是最优解）。

### A.1.3 KKT 条件

对于含有不等式约束的优化问题，常用的方法是 *KKT* 条件同样像等式约束一样将所有的不等约束也添加到拉格朗日函数中谢伟一个式子：

$$L(a, b, x) = f(x) + ag(x) + bh(x) \quad (A-2)$$

*KKT* 条件指出最优值必须满足一下三个条件：

- (1) 拉格朗日函数对  $x$  求导为零，即  $\frac{\partial L(a, b, x)}{\partial x} = 0$ ;
- (2) 等式约束自身为零，即  $h(x) = 0$ ;
- (3) 不等约束及其系数乘积为零，即  $ag(x) = 0$ ;

求取这三个等式之后就能得到候选最优值。其中第三个式子非常有趣，因为  $g(x) \leq 0$ ，如果要满足这个等式，必须  $a = 0$  或者  $g(x) = 0$ 。这是 *SVM* 的很多重要性质的来源，如支持向量的概念。

## A.2 二次规划问题

二次规划 (QP, Quadratic Programming) 定义：目标函数为二次函数，约束条件为线性约束，属于最简单的一种非线性规划。这部分内容主要参考[这个博客](#)。

### A.2.1 等式约束的 QP 问题

一个标准的等式约束 QP 问题模型如下：

$$\begin{aligned} \min \quad & \frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x} + \mathbf{g}^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{a}_i^T \mathbf{x} = \mathbf{b}_i, \quad i = 1, 2, \dots, m \end{aligned} \quad (\text{A-3})$$

其中  $\mathbf{H}$  是由二阶导构成的 Hessian 矩阵,  $\mathbf{g}^T \mathbf{x}$  是由梯度构成的 Jacobi 矩阵, 这里的向量都是列向量,  $\mathbf{g}^T \mathbf{x}$  表示转置成行向量。

其对应的拉格朗日函数为:

$$L(\mathbf{x}, \boldsymbol{\lambda}) = \frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x} + \mathbf{g}^T \mathbf{x} + \sum_{i=1}^m \lambda_i (\mathbf{a}_i^T \mathbf{x} - \mathbf{b}_i) \quad (\text{A-4})$$

满足 KKT 条件, 即满足:

$$\begin{cases} \frac{\partial L}{\partial \mathbf{x}} = \mathbf{H} \mathbf{x} + \mathbf{g} + \sum_{i=1}^m \lambda_i \mathbf{a}_i = 0 \\ \frac{\partial L}{\partial \lambda_i} = \mathbf{a}_i^T \mathbf{x} - \mathbf{b}_i = 0, \quad i = 1, 2, \dots, m \end{cases} \quad (\text{A-5})$$

将其写成矩阵形式:

$$\mathbf{F}(\mathbf{x}, \boldsymbol{\lambda}) = \begin{bmatrix} \mathbf{H} \mathbf{x} + \mathbf{g} + \boldsymbol{\lambda} \mathbf{a} \\ \mathbf{a}^T \mathbf{x} - \mathbf{b} \end{bmatrix} = 0 \quad (\text{A-6})$$

其中

$$\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_m \end{bmatrix}, \quad \mathbf{a}^T = \begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \vdots \\ \mathbf{a}_m^T \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_m \end{bmatrix} \quad (\text{A-7})$$

这是一个线性方程组, 易于求解, KKT 方程组的解  $\mathbf{x}^*, \boldsymbol{\lambda}^*$ , 即为优化模型的解。

### A.2.2 不等式约束的 QP 问题

一个标准的等式不等式联合约束 QP 问题模型如下:

$$\begin{aligned} \min \quad & \frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x} + \mathbf{g}^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{a}_i^T \mathbf{x} = \mathbf{b}_i, \quad i \in \mathbb{E} \end{aligned} \quad (\text{A-8})$$

$$\mathbf{h}_j^T \mathbf{x} \leq \mathbf{t}_j, \quad j \in \mathbb{I} \quad (\text{A-9})$$

其中  $\mathbf{H}$  是由二阶导构成的 Hessian 矩阵,  $\mathbf{g}^T \mathbf{x}$  是由梯度构成的 Jacobi 矩阵, 这里的向量都是列向量,  $\mathbf{g}^T \mathbf{x}$  表示转置成行向量。  $i \in \mathbb{E}$  表示  $m$  个等式约束集合,  $i \in \mathbb{I}$  表示  $n$  个不等式约束集合。下面介绍两种不等式约束条件下的 QP 问题求解方案。

## A.2.2.1 内点法

模型的拉格朗日函数为：

$$L(x, \lambda, \mu) = \frac{1}{2}x^T Hx + g^T x + \sum_{i=1}^m \lambda_i (a_i^T x - b_i) + \sum_{j=1}^n \mu_j (h_j^T x - t_j) \quad (\text{A-10})$$

**Notes A.2.1.** 内点法详细可以在接下来的内容中了解后补充一下。

以原始对偶内点法为例，加入少量  $\tau$  扰动后 KKT 条件为：

$$\begin{cases} \frac{\partial L}{\partial x} = Hx + g + \sum_{i=1}^m \lambda_i a_i + \sum_{j=1}^n \mu_j h_j = 0 \\ \frac{\partial L}{\partial \lambda_i} = a_i^T x - b_i = 0, \quad i = 1, 2, \dots, m \\ h_j x \leq t_j \\ \mu_j (h_j x - t_j) + \tau_j = 0 \\ \mu_j \geq 0, \quad j = 1, 2, \dots, n \end{cases} \quad (\text{A-11})$$

为了更快地检查解是否在约束空间内，我们在不等式方程组引入了松弛变量  $s_j = t_j - h_j x$ ，因为判断  $s_j \leq 0$  要比判断  $h_j x \leq t_j$  方便的多，这样一来上式变为：

$$\begin{cases} \frac{\partial L}{\partial x} = Hx + g + \sum_{i=1}^m \lambda_i a_i + \sum_{j=1}^n \mu_j h_j = 0 \\ \frac{\partial L}{\partial \lambda_i} = a_i^T x - b_i = 0, \quad i = 1, 2, \dots, m \\ h_j x + s_j = t_j \\ \mu_j (h_j x - t_j) + \tau_j = 0 \rightarrow \mu_j s_j - \tau_j = 0 \\ \mu_j, s_j \geq 0, \quad j = 1, 2, \dots, n \end{cases} \quad (\text{A-12})$$

将其写成矩阵形式：

$$F(x, s, \lambda, \mu) = \begin{bmatrix} Hx + g + \lambda a + \mu h \\ h^T x + s - t \\ a^T x - b \\ \mu s - \tau 1 \end{bmatrix} = 0 \quad (\text{A-13})$$

**Notes A.2.2.** 不知道这里  $\tau$  后面的那个 1 是什么意思？难道是笔误？

其中：

$$\begin{aligned} \boldsymbol{\lambda} &= \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_m \end{bmatrix}, \quad \mathbf{a}^T = \begin{bmatrix} a_1^T \\ a_2^T \\ \vdots \\ a_m^T \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}, \\ \boldsymbol{\mu} &= \begin{bmatrix} \mu & 0 & \cdots & 0 \\ 0 & \mu_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mu_m \end{bmatrix}, \quad \mathbf{h}^T = \begin{bmatrix} h_1^T \\ h_2^T \\ \vdots \\ h_n^T \end{bmatrix}, \quad \mathbf{t} = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_n \end{bmatrix}, \quad \mathbf{s} = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix} \end{aligned} \quad (\text{A-14})$$

牛顿解法方程组：

$$\mathbf{F}(\mathbf{x}_k, \mathbf{s}_k, \boldsymbol{\lambda}_k, \boldsymbol{\mu}_k) + \mathbf{F}'(\Delta \mathbf{x}_k, \Delta \mathbf{s}_k, \Delta \boldsymbol{\lambda}_k, \Delta \boldsymbol{\mu}_k) = 0 \quad (\text{A-15})$$

也即：

$$\begin{bmatrix} \mathbf{H} & 0 & \mathbf{a}^T & \mathbf{h}^T \\ \mathbf{h}^T & \mathbf{I} & 0 & 0 \\ \mathbf{a}^T & 0 & 0 & 0 \\ 0 & \boldsymbol{\mu}_k & 0 & \mathbf{s}^T \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x}_k \\ \Delta \mathbf{s}_k \\ \Delta \boldsymbol{\lambda}_k \\ \Delta \boldsymbol{\mu}_k \end{bmatrix} = - \begin{bmatrix} \mathbf{H}\mathbf{x}_k + \mathbf{g} + \boldsymbol{\lambda}_k \mathbf{a} + \boldsymbol{\mu}_k \mathbf{h} \\ \mathbf{h}^T \mathbf{x}_k + \mathbf{s}_k - \mathbf{t} \\ \mathbf{a}^T \mathbf{x}_k - \mathbf{b} \\ \boldsymbol{\mu}_k \mathbf{s}_k - \boldsymbol{\tau}_k \mathbf{1} \end{bmatrix} \quad (\text{A-16})$$

得到一组  $\Delta \mathbf{x}_k, \Delta \mathbf{s}_k, \Delta \boldsymbol{\lambda}_k, \Delta \boldsymbol{\mu}_k$  然后更新变量  $\mathbf{x}_k, \mathbf{s}_k, \boldsymbol{\lambda}_k, \boldsymbol{\mu}_{k+1} = (\mathbf{x}_k, \mathbf{s}_k, \boldsymbol{\lambda}_k, \boldsymbol{\mu}_k) + \alpha \Delta \mathbf{x}_k, \Delta \mathbf{s}_k, \Delta \boldsymbol{\lambda}_k, \Delta \boldsymbol{\mu}_k$ 。同时更新  $\boldsymbol{\tau}_{k+1} = -\sigma \sum_{j=1}^n \boldsymbol{\mu}_{j,k} \mathbf{s}_{j,k}$ ,  $\sigma \in [0, 1]$  进入第  $k+1$  次迭代直到方程组的解，并且满足  $\boldsymbol{\tau}_k \leq \epsilon$ 。

**Notes A.2.3.** 这下终于清楚论文里面所谓的二次规划问题具体在说什么了。巨额这个过程放在编程上，想要实现也不是个容易的事情。除此之外，实现了的接口用起来对里面的各个参数选择也是个有挑战大问题，可能要花费较长的时间来调整。2023-10-05 23:13:53，竟然这么快又 11 点多了！好在我下午还是回来了，不然这花功夫的笔记进度要延后许久呀。

#### A.2.2.2 积极集法