# 目 录

# 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[1]p2：

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

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

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

它里面 ${}_I\boldsymbol{r}_{IB} \in \mathbb{R}^3$ 是机器人主体相对于惯性系的三维位置矢量；$\boldsymbol{q}_{IB} \in SO(3)$ 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；$\boldsymbol{q}_j \in \mathbb{R}^{n_j}$ 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 $\boldsymbol{q}$ 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 $\boldsymbol{u}$ 是一个描述机器人主体及各个节点的广义速度矢量：

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

它里面的 ${}_I\boldsymbol{v}_B$ 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；$\dot{\boldsymbol{q}}_j$ 描述了机器人的各个关节转动的速度。这其中 $\boldsymbol{M}$ 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：

$$\boldsymbol{M} \in \mathbb{R}^{n_u \times n_u} \tag{1-4}$$

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 $\boldsymbol{q}$）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 $\boldsymbol{q}$ 的值，就可以计算出 $\boldsymbol{M}$ 矩阵的各个元素具体数值。这其中 $\boldsymbol{h}$ 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：

$$\boldsymbol{h} \in \mathbb{R}^{n_u} \tag{1-5}$$

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

这其中 $\boldsymbol{S}$ 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：

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

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：

$$\boldsymbol{\tau} \in \mathbb{R}^{n_j} \tag{1-7}$$

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

$$J_S = \begin{bmatrix} J_{C_1}^T & \dots & J_{C_{n_c}}^T \end{bmatrix}^T \in \mathbb{R}^{3n_c \times n_u} \tag{1-8}$$

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[2]p7：

$$\begin{aligned} x(t) =& a_{i5}^x t^5 + a_{i4}^x t^4 + a_{i3}^x t^3 + a_{i2}^x t^5 2 + a_{i1}^x t^1 + a_{i0}^x t^0 \\ =& \begin{bmatrix} t^5 & t^4 & t^3 & t^2 & t^1 & t^0 \end{bmatrix} \\ & \cdot \begin{bmatrix} a_{i5} & a_{i4} & a_{i3} & a_{i2} & a_{i1} & a_{i0} \end{bmatrix} \\ =& \boldsymbol{\eta}^T(t) \boldsymbol{\alpha}_i^x \end{aligned} \tag{1-9}$$

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：

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

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

这其中：

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

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

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 $\boldsymbol{\alpha}_i = \begin{bmatrix} \boldsymbol{\alpha}_i^{x^T} & \boldsymbol{\alpha}_i^{y^T} & \boldsymbol{\alpha}_i^{z^T} \end{bmatrix}^T$ 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：$\boldsymbol{\alpha} = \begin{bmatrix} \boldsymbol{\alpha}_0^T & \dots & \boldsymbol{\alpha}_i^T & \dots & \boldsymbol{\alpha}_{n_s}^T \end{bmatrix}^T$。这样一来，质心的位置可以表示为：

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

同样质心的速度和位置也可以得到了：

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

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

# 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 $\boldsymbol{c}(\boldsymbol{\xi})$ 和不等约束 $\boldsymbol{h}(\boldsymbol{\xi})$ 的条件下最小化一个通用的成本函数 $\boldsymbol{f}(\boldsymbol{\xi})$。解决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：

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

相应的 $\boldsymbol{c}_k^{acc} = 0$。$\Delta t_k$ 是第 $k$ 个曲线段以秒为单位的持续时间。$\boldsymbol{Q}_k^{acc}$ 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。$\boldsymbol{Q}_k^{acc}$ 被添加为整个成本函数的 Hessian 的子矩阵。

> **Notes 2.1.1.**（$\boldsymbol{Q}_k^{acc}$ 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 $v_{ref}$ 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：

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

这里面第一项 $\boldsymbol{p}_{init}$ 是前一时刻的位置，第二项是当前时刻的位移 $\tau\boldsymbol{v}_{ref}$ 乘上一个旋转矩阵 $\boldsymbol{R}(\tau\hat{\boldsymbol{\omega}}_z) = \begin{bmatrix} 0 & 0 & \boldsymbol{\omega}_{ref_z} \end{bmatrix}^T$。路径正则化 $\boldsymbol{\pi}$。$\boldsymbol{\pi}$ 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。$\boldsymbol{\pi}$ 的初始和最终加速度设置为零。$\boldsymbol{\pi}$ 的初始和最终高度设置为用户指定的参考值。（这个 $\boldsymbol{\pi}$ 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：

$$\boldsymbol{p}_{CoM}^{z}(t)-\boldsymbol{\pi}_z(t)\leqslant\epsilon_z \tag{2-3}$$

$$\boldsymbol{p}_{CoM}^{z}(t)-\boldsymbol{\pi}_z(t)\geqslant-\epsilon_z \tag{2-4}$$

$$\epsilon_z\geqslant 0 \tag{2-5}$$

这里面 $\epsilon$ 是一个松弛变量，它会被添加入优化参数 $\boldsymbol{\xi}$ 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：

$$\begin{bmatrix}\boldsymbol{\eta}(t_{fk})^T & -\boldsymbol{\eta}(0)^T\\ \dot{\boldsymbol{\eta}}(t_{fk})^T & -\dot{\boldsymbol{\eta}}(0)^T\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}_k^x\\ \boldsymbol{\alpha}_{k+1}^x\end{bmatrix}=0 \tag{2-6}$$

这里面 $t_{fk}$ 代表曲线 $\boldsymbol{s}_k$ 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：$\ddot{\boldsymbol{p}}_{CoM}=\boldsymbol{g}$，其中 $\boldsymbol{g}$ 是重力加速度矢量。将上式积分可以得到：

$$\dot{\boldsymbol{p}}(t)=\boldsymbol{g}t+\dot{\boldsymbol{p}}_{CoM}(0) \tag{2-7}$$

$$\boldsymbol{p}(t)=\frac{1}{2}\boldsymbol{g}t^2+\dot{\boldsymbol{p}}_{CoM}(t)+\boldsymbol{p}_{CoM}(0)(0) \tag{2-8}$$

着陆时的质心动力学可以描述为：

$$\ddot{\boldsymbol{T}}(0)\boldsymbol{\alpha}_{i+1}=\boldsymbol{g} \tag{2-9}$$

$$\dot{\boldsymbol{T}}(0)\boldsymbol{\alpha}_{i+1}=\boldsymbol{g}t_f+\dot{\boldsymbol{T}}(t_i)\boldsymbol{\alpha}_i \tag{2-10}$$

$$\boldsymbol{T}(0)\boldsymbol{\alpha}_{i+1}=\frac{1}{2}\boldsymbol{g}t_f^2+[\dot{\boldsymbol{T}}(t_i)t_f+\boldsymbol{T}(t_i)]\boldsymbol{\alpha}_i \tag{2-11}$$

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

$$\boldsymbol{p}_{ZMP} = \frac{\boldsymbol{n} \times \boldsymbol{M}_O^{gi}}{\boldsymbol{n}^T \boldsymbol{F}^{gi}} \tag{2-12}$$

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

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

$$\boldsymbol{F}^{gi} = m \cdot \boldsymbol{g} - \dot{\boldsymbol{P}} \tag{2-14}$$

这里面 $m, \boldsymbol{P}, \boldsymbol{L}$ 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 $\dot{\boldsymbol{L}} = 0$。

#### 2.1.3.2　引入的不等约束

基本要求是 ZMP 点药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：

$$\boldsymbol{h}_{ZMP} = \boldsymbol{d}^T \boldsymbol{p}_{ZMP} + r \geqslant 0 \tag{2-15}$$

这其中 $\boldsymbol{d}^T = \begin{bmatrix} p & q & 0 \end{bmatrix}$ 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：

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

这里面定义了一个斜对称矩阵 $\boldsymbol{S}(a)$ 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：$\boldsymbol{S}(\boldsymbol{a})\boldsymbol{b} = \boldsymbol{a} \times \boldsymbol{b}$。用 (15) 也可以计算出 (14) 相对于质心位置 $\boldsymbol{p}_{CoM}$ 和质心加速度

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

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

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

$$\nabla^2 \boldsymbol{h}_{ZMP} = \begin{bmatrix} 0 & \boldsymbol{\Gamma}^T \\ -\boldsymbol{\Gamma}^T & 0 \end{bmatrix} \tag{2-18}$$

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 $\boldsymbol{\xi}$ 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 $\boldsymbol{c}_{ineq} \geqslant -\xi_{ineq}$, $\xi_{ineq} \geqslant 0$，其中 $\boldsymbol{c}_{ineq}$ 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：

$$t = arg\,min_{\boldsymbol{W}} ||\boldsymbol{p}(t) - \boldsymbol{p}_{meas}||_2^2 \tag{2-19}$$

这里面 $\boldsymbol{W}$ 是一个正定权重矩阵 $\boldsymbol{p}_{meas}$ 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：

$$\min_{\boldsymbol{\xi}} \quad \frac{1}{2}\boldsymbol{\xi}^T\boldsymbol{Q}\boldsymbol{\xi} + \boldsymbol{c}^T\boldsymbol{\xi} \quad s.t. \quad \boldsymbol{D}\boldsymbol{\xi} \leqslant \boldsymbol{f} \tag{2-20}$$

这里面 $\boldsymbol{\xi} \in \mathbb{R}^{2n_{feet}}$ 是裸足点 $\boldsymbol{p}_{f_i}$ 的 $x, y$ 方向分量，其中 $i = 1, \ldots, n_{feet}$，$n_{feet} = 4$ 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 $\boldsymbol{p}_{ref_i}$ 我们为成本函数 (19) 添加下述表述：

$$\boldsymbol{p}_{ref_i} = \boldsymbol{W}_{def}, \quad \boldsymbol{c}_{def_i} = -\boldsymbol{W}_{def}^{T}\boldsymbol{p}_{ref_i} \tag{2-21}$$

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 $\omega(\boldsymbol{v}_ref - \boldsymbol{v}_{hip_k}\sqrt{h/g})$ 计算第 k 个落脚点。这其中，$\omega$ 是一个正的权重标量，$\boldsymbol{v}_ref$ 是高级速度参考，$\boldsymbol{b}_{hips}$ 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 $\boldsymbol{h}_0$，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 $\boldsymbol{h}_0$ 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 $\boldsymbol{h}_0$ 的距离计算为：

$$\sqrt{l_{max}^2 - h_i^2} \tag{2-22}$$

**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

$\boldsymbol{\xi}_d$ 下面的控制方法采用接触力控制

### 3.1.1　运动方程

通过利用选择矩阵 $\boldsymbol{S}^T$ 诱导的分解，我们可以将运动和接触力限制在浮动基系统动力学描述的流形上，有如下式[1]p2：

$$\begin{bmatrix}\boldsymbol{M}_{fb} & -\boldsymbol{J}_{s_{fb}}^T\end{bmatrix}\boldsymbol{\xi}_d = -\boldsymbol{h}_{fb} \tag{3-1}$$

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

（1）$\boldsymbol{\xi}_d$ 是一个一个 $n_u + n_c$ 行 1 列的向量：$\boldsymbol{\xi}_d = \begin{bmatrix}\boldsymbol{u}_d^T & \boldsymbol{\lambda}_d^T\end{bmatrix} \in \mathbb{R}^{n_u+n_c}$，其中:

① $\boldsymbol{u}_d^T$ 是目标节点的加速度；

② $\boldsymbol{\lambda}_d^T$ 是目标接触力；

（2）$\boldsymbol{M}_{fb}$ 是复合型惯性矩阵的前六行；啥是复合惯性矩阵？

（3）$\boldsymbol{J}_{s_{fb}}^T$ 是雅克比阵的前六行，它将接触力转换到节点的扭矩；

（4）$\boldsymbol{h}_{fb}$ 是非线性项的前六行，包括克里奥利力、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:

$$\begin{bmatrix}\boldsymbol{J}_s & 0_{3n_c\times 3n_c}\end{bmatrix}\boldsymbol{\xi}_d = -\dot{\boldsymbol{J}}_s\boldsymbol{u} \tag{3-2}$$

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

### 3.1.3　接触力和扭矩限制

$$({}_I\boldsymbol{h} - {}_I\boldsymbol{n}_\mu)^T {}_I\boldsymbol{\lambda}_k \leqslant 0 \tag{3-3}$$

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

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

$$-({}_I\boldsymbol{l} + {}_I\boldsymbol{n}_\mu)^T {}_I\boldsymbol{\lambda}_k \leqslant 0 \tag{3-6}$$

${}_In$ 是接触面的法向量；$\mu$ 是摩擦系数；有了这两个再乘上受力 ${}_I\lambda$，就可以得出最大静摩擦力 ${}_I\boldsymbol{n}_\mu^T\lambda$。实际在接触点平行于地面方向的两个分力 ${}_I\boldsymbol{h}$，${}_I\boldsymbol{l}$ 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。

$$\boldsymbol{\tau}_{min} - \boldsymbol{h}_j \leqslant \begin{bmatrix}\boldsymbol{M}_j & -\boldsymbol{J}_{s_j}^T\end{bmatrix} \leqslant \boldsymbol{\tau}_{max} - \boldsymbol{h}_j \tag{3-7}$$

这里的 $\boldsymbol{h}_j$ 应该就是科里奥利力那一堆东西。而 $\begin{bmatrix}\boldsymbol{M}_j & -\boldsymbol{J}_{s_j}^T\end{bmatrix}$ 就是加速度力 $\boldsymbol{M}_j\dot{\boldsymbol{u}}_d^T$ 和传递力 $-\boldsymbol{J}_{s_j}^T\boldsymbol{\lambda}_d^T$ 两项的和。它们计算的结果就是电机提供的扭矩力 $\boldsymbol{\tau}$ 克服完 $\boldsymbol{h}(\boldsymbol{q},\boldsymbol{u})$ 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：

$$\begin{bmatrix}{}_C\boldsymbol{J}_P & 0\end{bmatrix}\boldsymbol{\xi}_d = {}_C\,\ddot{\boldsymbol{r}}_{IB}^d + \boldsymbol{k}_D^{pos}({}_C\dot{\boldsymbol{r}}_{IB}^d - {}_C\,\boldsymbol{v}_B) + \boldsymbol{k}_P^{pos}({}_C\boldsymbol{r}_{IB}^d - {}_C\,\boldsymbol{r}_B) \tag{3-8}$$

对于主体的角度运动：

$$\begin{bmatrix}{}_C\boldsymbol{J}_R & 0\end{bmatrix}\boldsymbol{\xi}_d = -\boldsymbol{k}_D^{ang}{}_C\boldsymbol{\omega}_B + \boldsymbol{k}_P^{ang}(\boldsymbol{q}_{CB}^d \Xi \boldsymbol{q}_{CB}) \tag{3-9}$$

* 雅可比矩阵 ${}_C\boldsymbol{J}_P$ 和 ${}_C\boldsymbol{J}_R$ 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。

* $\boldsymbol{\Xi}$ 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 $\boldsymbol{k}_P^{pos},\boldsymbol{k}_D^{pos},\boldsymbol{k}_P^{ang},\boldsymbol{k}_D^{ang}$ 是用来控制增益的对角正定矩阵。

* 参考的运动 ${}_C\boldsymbol{r}_{IB}$ 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：

$$\begin{bmatrix}0_{3n_c\times n_u} & \mathbb{I}_{3n_c\times 3n_c}\end{bmatrix}\boldsymbol{\xi}_d = 0 \tag{3-10}$$

### 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，$\boldsymbol{\xi}_d = \begin{bmatrix} \dot{\boldsymbol{u}}_d^T & \boldsymbol{\lambda}_d^T \end{bmatrix}^T$ 我们可以用以下公式计算各个电机的扭矩：

$$\boldsymbol{\tau}_d = \begin{bmatrix} \boldsymbol{M}_j & -\boldsymbol{J}_{s_j}^T \end{bmatrix} \boldsymbol{\xi}_d + \boldsymbol{h}_j$$

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

**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 $\dot{\boldsymbol{u}}_d^T$ 和接触力 $\boldsymbol{\lambda}_d^T$。

# 第 4 章　运动优化

这部分主要参考文献[1]p3-6。

## 4.1　足态保持

## 4.2　支撑多边形序列生成

## 4.3　问题描述

## 4.4　规划初始化

## 4.5　成本函数

## 4.6　等式约束

## 4.7　不等式约束

## 4.8　约束松弛

> 这部分的详细描述要参考文献 2。

# 第 5 章　如何构建 RL 控制训练模型

在这部分将阐述关于如何使用强化学习来进行运动控制的内容。相比于传统的基于模型的控制方法，该方法通过在模拟环境中训练控制策略，可以更好地适应复杂的非线性系统动力学和环境不确定性，并且能够实时响应用户命令和环境变化。在这里我们主要关注如何使用强化学习来训练机械狗类型的足式和轮式机器人控制器。

## 5.1　强化学习控制

本部分内容的探究起点是这篇文献的内容：*Control of Wheeled-Legged Quadrupeds Using Deep Reinforcement Learning*[3]

### 5.1.1　强化学习的基本概念

在强化学习中，控制问题被建模为一个马尔可夫决策过程（Markov Decision Process）。MDP 是一个 RL 中常用的用于随机控制过程建模的数学框架，它定义了一个包含状态空间、动作空间、奖励函数和状态转移函数的元组，描述了一个决策过程的基本组成部分。在 MDP 中，每个时间步骤代理从周围环境中观察到某个状态 $s_t \in \mathcal{S}$，并输出一个动作 $a_t \in \mathcal{A}$，接着环境通过状态转移函数 $p(s_{t+1}|s_t, a_t)$ 演化到新的状态 $s_{t+1}$，并且根据奖励函数给出相应的奖励 $r_t \in \mathcal{R}: \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ 和对更新后的环境状态的观察结果作为下一周期的 $s_t$。代理可以根据 $\theta$ 参数化的随机策略 $\pi_\theta(a_t|s_t)$ 来采取行动。RL 通过与环境交互来更新参数 $\theta$ 以最大化累计折扣奖励（cumulative discounted rewards）$\mathbb{E}[\sum_{t=k}^{\infty} \gamma^t r_t]$，其中 $k$ 是当前时间步长（timestep），$\gamma$ 是折扣因子（discount factor）。

### 5.1.2　如何实现基于深度强化学习的控制器在时间环境中的部署和优化

（1）在仿真环境中训练控制器：使用深度强化学习算法在仿真环境中训练控制器，使其能够完成所需的任务。在训练过程中，可以使用特权训练方法来提高训练效率和性能。

（2）部署控制器到实际环境中：将训练好的控制器部署到实际环境中，例如机器人或移动设备。在实际环境中，控制器将接收传感器数据并输出动作命令。

（3）优化控制器：在实际环境中，可以使用在线学习方法来进一步优化控制

器的性能。例如，可以使用模型预测控制方法来对控制器进行在线微调。

需要注意的是，在实际环境中，传感器数据可能会受到噪声和不确定性的影响，因此需要设计鲁棒性强的控制器来应对这些问题。此外，还需要考虑实际环境中的安全性和可靠性问题。

## 5.2　RL 的构建、训练和部署

### 5.2.1　MDP 的制定

一个 MDP 是由一个元组定义，它包含状态空间 $\mathcal{S}$、动作空间 $\mathcal{A}$、奖励函数 $r_t(s_t|s_{t+1})$、转移函数 $p(s_{t+1}|s_t, a_t)$。这个策略的环境观察部分包括周围地面的高度扫描结果、来自 IMU 的本体姿态信息、来自各个关节点的编码器信息。运动被定义为一个 21 维的向量，包括步态参数偏移、关节点位置偏移和轮子速度控制指令。奖励函数由表5-1定义。

表 5-1　Reward functions. (The velocities are defined in the base frame)

| **Reward terms** | |
|---|---|
| Linear velocity | $1.5\exp(-3(v^{xy}-\hat{v}^{xy})^2)$ |
| Angular velocity | $1.0\exp(-3(\omega^{z}-\hat{\omega}^{z})^2)$ |
| Base motion penalty | $0.8\exp(-(v^{z})^2)+\exp(-(\omega^{x,y})^2)$ |
| Torque penalty | $-10^{-6}\|\tau\|^2$ |
| Joint speed penalty | $-0.05\sum_{i=0}^{12}\max(\|\dot{\phi}_i\|-\dot{\phi}_{jslim},0)^2$ |
| Action smoothness penalty | $-0.05\|a_{t-2}-2a_{t-1}+a_t\|$ |
| **Symbols** | |
| $\dot{\phi}_{jslim}$ | Maximum joint speed |
| $\tau$ | Vector of joint torques |
| $f_c$ | Foot contact state |
| $\hat{(\cdot)}$ | Traget quantity |
| $(\cdot)_g$ | Sub-goal quantity |
| $\mathbb{1}_{(condition)}(\cdot)$ | Indicator function |

状态转换遵循刚性的动力学模拟，当机器人主体接触地面或到达关节扭矩和速度限制时，该训练集终止。

### 5.2.2　RL 模型的训练方式

我们遵循 Lee 等人[4]的特权训练方法。我们首先使用 on-policy RL 算法[5]训练具有无噪声模拟状态 (特权信息) 的教师策略。换句话说，我们用额外的模拟状态附加给 $s_t$，包括地面摩擦系数、地面反作用力以及每个连杆的接触状态。特权信息使教师策略能够学习地形自适应行为。由于无噪声和准确的模拟状态，教师策略快速收敛到高性能策略。在桌面 PC 上使用 PPO 算法[5]，教师策略训练大约需要 8 小时。然后我们为实际部署训练一个学生策略。学生策略在没有特权信息和观察噪声的情况下模仿老师。通过模仿学习[6]，它学习从嘈杂的真实世界观察序列构建世界的内部表示。以这种方式训练的策略已被证明在具有高干扰和噪声观测的真实世界环境中更具适应性和鲁棒性[4]。

### 5.2.3　训练平台和部署

训练采用的平台有 Raisim[3,7]仿真器、英伟达的 Isaac[8]仿真器。

**Notes 5.2.1.** 2023-10-06 18:19:57，感觉强化学习者部署起来也不是件容易得事情呀。后面关于运动控制建模的方式就先不看了。重点关注用强化学习做控制要关注的内容。

# 第 6 章　Xxx

# 参考文献


[1] BELLICOSO C, JENELTEN F, FANKHAUSER P, et al. Dynamic locomotion and whole-body control for quadrupedal robots[C/OL]//2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2017. http://dx.doi.org/10.1109/iros.2017.8206174.

[2] BELLICOSO C D, JENELTEN F, GEHRING C, et al. Dynamic Locomotion Through Online Nonlinear Motion Optimization for Quadrupedal Robots[J/OL]. IEEE Robotics and Automation Letters, 2018: 2261–2268. http://dx.doi.org/10.1109/lra.2018.2794620.

[3] LEE J, BJELONIC M, HUTTER M. Control of Wheeled-Legged Quadrupeds Using Deep Reinforcement Learning[M/OL]. 2023: 119–127. http://dx.doi.org/10.1007/978-3-031-15226-9_14.

[4] LEE J, HWANGBO J, WELLHAUSEN L, et al. Learning Quadrupedal Locomotion over Challenging Terrain[J/OL]. Science Robotics, 2020. http://dx.doi.org/10.1126/scirobotics.abc5986.

[5] SCHULMAN J, WOLSKI F, DHARIWAL P, et al. Proximal Policy Optimization Algorithms [A]. 2017.

[6] ROSS S, GORDON G, BAGNELL J. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning[A]. 2010.

[7] HWANGBO J, LEE J, HUTTER M. Per-Contact Iteration Method for Solving Contact Dynamics[J/OL]. IEEE Robotics and Automation Letters, 2018: 895–902. http://dx.doi.org/10.1109/lra.2018.2792536.

[8] RUDIN N, HOELLER D, REIST P, et al. Learning to Walk in Minutes Using Massively Parallel Deep Reinforcement Learning[A]. 2021.

# 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：

$$minf(x);$$

（2）有等式约束的优化问题，可以写为：

$$minf(x);$$
$$s.t. \quad h_i(x) = 0, i = 1, \cdots, n;$$

（3）有等式和不等式约束的优化问题，可以写为：

$$minf(x);$$
$$s.t. \quad g_i \leqslant 0, i = 1, \cdots, n;$$
$$and \quad h_i(x) = 0, i = 1, \cdots, n;$$

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：

$$\boldsymbol{L(a,x)=f(x)+ah(x)} \tag{A-1}$$

其中 $\boldsymbol{a,h(x)}$ 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 $\boldsymbol{L(a,x)}$ 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 $\boldsymbol{a}$ 的偏导为零，即 $\frac{\partial \boldsymbol{L(a,x)}}{\partial \boldsymbol{a}}=0$；

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

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 $\boldsymbol{a,x}$ 的候选值，然后验证 $\boldsymbol{x}$ 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：

$$\boldsymbol{L(a,b,x)=f(x)+ag(x)+bh(x)} \tag{A-2}$$

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 $\frac{\boldsymbol{L(a,b,x)}}{\partial \boldsymbol{x}}$；

（2）等式约束自身为零，即 $\boldsymbol{h(x)}=0$；

（3）不等约束及其系数乘积为零，即 $\boldsymbol{ag(x)}=0$；

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 $\boldsymbol{g(x)}\leqslant 0$，如果要满足这个等式，必须 $\boldsymbol{a}=0$ 或者 $\boldsymbol{g(x)}=0$. 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：

$$\begin{aligned} min \quad & \frac{1}{2}\boldsymbol{x}^T\boldsymbol{H}\boldsymbol{x}+\boldsymbol{g}^T\boldsymbol{x} \\ s.t. \quad & \boldsymbol{a}_i^T\boldsymbol{x}=b_i, \quad i=1,2,\cdots,m \end{aligned} \tag{A-3}$$

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

其对应的拉格朗日函数为:

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

满足 KKT 条件，即满足：

$$\begin{cases} \frac{\partial \boldsymbol{L}}{\partial \boldsymbol{x}}=\boldsymbol{H}\boldsymbol{x}+\boldsymbol{g}+\sum_{i=1}^{m}\lambda_i\boldsymbol{a}_i=0 \\ \frac{\partial \boldsymbol{L}}{\partial \lambda_i}=\boldsymbol{a}_i^T\boldsymbol{x}-b_i=0, \quad i=1,2,\cdots,m \end{cases} \tag{A-5}$$

将其写成矩阵形式：

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

其中

$$\boldsymbol{\lambda}=\begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_m \end{bmatrix}, \quad \boldsymbol{a}^T=\begin{bmatrix} \boldsymbol{a}_1^T \\ \boldsymbol{a}_2^T \\ \vdots \\ \boldsymbol{a}_m^T \end{bmatrix}, \quad \boldsymbol{b}=\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \tag{A-7}$$

这是一个线性方程组，易于求解，KKT 方程组的解 $\boldsymbol{x}^*,\boldsymbol{\lambda}^*$，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：

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

$$\boldsymbol{h}_j^T\boldsymbol{x}\leqslant t_j, \quad j\in\mathbb{I} \tag{A-9}$$

其中 $\boldsymbol{H}$ 是由二阶导构成的 Hessian 矩阵，$\boldsymbol{g}^T\boldsymbol{x}$ 是由梯度构成的 Jacobi 矩阵，这里的向量都是列向量，$\boldsymbol{g}^T\boldsymbol{x}$ 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：

$$\boldsymbol{L}(\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{\mu})=\frac{1}{2}\boldsymbol{x}^T\boldsymbol{H}\boldsymbol{x}+\boldsymbol{g}^T\boldsymbol{x}+\sum_{i=1}^{m}\lambda_i(\boldsymbol{a}_i^T\boldsymbol{x}-b_i)+\sum_{j=1}^{n}\mu_j(\boldsymbol{h}_j^T\boldsymbol{x}-t_j) \tag{A-10}$$

**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 $\boldsymbol{\tau}$ 扰动后 KKT 条件为：

$$\begin{cases}\frac{\partial \boldsymbol{L}}{\partial \boldsymbol{x}}=\boldsymbol{H}\boldsymbol{x}+\boldsymbol{g}+\sum_{i=1}^{m}\lambda_i\boldsymbol{a}_i+\sum_{j=1}^{n}\mu_j\boldsymbol{h}_j=0\\ \frac{\partial \boldsymbol{L}}{\partial \lambda_i}=\boldsymbol{a}_i^T\boldsymbol{x}-b_i=0,\quad i=1,2,\cdots,m\\ \boldsymbol{h}_j^T\boldsymbol{x}\leqslant t_j\\ \mu_j(\boldsymbol{h}_j^T\boldsymbol{x}-t_j)+\tau_j=0\\ \mu_j\geqslant 0,\quad j=1,2,\cdots,n\end{cases} \tag{A-11}$$

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 $s_j=t_j-\boldsymbol{h}_j^T\boldsymbol{x}$，因为判断 $s_j\leqslant 0$ 要比判断 $\boldsymbol{h}_j^T\boldsymbol{x}\leqslant t_j$ 方便的多，这样一来上式变为：

$$\begin{cases}\frac{\partial \boldsymbol{L}}{\partial \boldsymbol{x}}=\boldsymbol{H}\boldsymbol{x}+\boldsymbol{g}+\sum_{i=1}^{m}\lambda_i\boldsymbol{a}_i+\sum_{j=1}^{n}\mu_j\boldsymbol{h}_j=0\\ \frac{\partial \boldsymbol{L}}{\partial \lambda_i}=\boldsymbol{a}_i^T\boldsymbol{x}-b_i=0,\quad i=1,2,\cdots,m\\ \boldsymbol{h}_j^T\boldsymbol{x}+s_j=t_j\\ \mu_j(\boldsymbol{h}_j^T\boldsymbol{x}-t_j)+\tau_j=0\rightarrow\mu_j s_j-\tau_j=0\\ \mu_j,s_j\geqslant 0,\quad j=1,2,\cdots,n\end{cases} \tag{A-12}$$

将其写成矩阵形式：

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

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

理论上第一行是关于 $\boldsymbol{x}$ 的偏导项，第二项和第三项分别是关于等式约束和不等式约束引入的拉格朗日乘子的偏导项。最后一项 $\boldsymbol{\mu s}-\boldsymbol{\tau}1=0$ 的含义不是很清楚是什么意思？这里面 $\boldsymbol{\mu}$ 是引入的拉格朗日乘子，$\boldsymbol{s}$ 是引入的松弛变量，$\boldsymbol{\tau}$ 是少量扰动。

其中：

$$\boldsymbol{\lambda}=\begin{bmatrix}\lambda_1 & 0 & \cdots & 0\\ 0 & \lambda_2 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & \lambda_m\end{bmatrix},\quad \boldsymbol{a}^T=\begin{bmatrix}\boldsymbol{a}_1^T\\ \boldsymbol{a}_2^T\\ \vdots\\ \boldsymbol{a}_m^T\end{bmatrix},\quad \boldsymbol{b}=\begin{bmatrix}b_1\\ b_2\\ \vdots\\ b_m\end{bmatrix},$$

$$\boldsymbol{\mu}=\begin{bmatrix}\mu & 0 & \cdots & 0\\ 0 & \mu_2 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & \mu_m\end{bmatrix},\quad \boldsymbol{h}^T=\begin{bmatrix}\boldsymbol{h}_1^T\\ \boldsymbol{h}_2^T\\ \vdots\\ \boldsymbol{h}_n^T\end{bmatrix},\quad \boldsymbol{t}=\begin{bmatrix}t_1\\ t_2\\ \vdots\\ t_n\end{bmatrix},\quad \boldsymbol{s}=\begin{bmatrix}s_1\\ s_2\\ \vdots\\ s_n\end{bmatrix} \tag{A-14}$$

牛顿解法方程组：

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

也即：

$$\begin{bmatrix}\boldsymbol{H} & 0 & \boldsymbol{a}^T & \boldsymbol{h}^T\\ \boldsymbol{h}^T & I & 0 & 0\\ \boldsymbol{a}^T & 0 & 0 & 0\\ 0 & \boldsymbol{\mu}_k & 0 & \boldsymbol{s}^T\end{bmatrix}\begin{bmatrix}\Delta\boldsymbol{x}_k\\ \Delta\boldsymbol{s}_k\\ \Delta\boldsymbol{\lambda}_k\\ \Delta\boldsymbol{\mu}_k\end{bmatrix}=-\begin{bmatrix}\boldsymbol{H}\boldsymbol{x}_k+\boldsymbol{g}+\boldsymbol{\lambda}_k\boldsymbol{a}+\boldsymbol{\mu}_k\boldsymbol{h}\\ \boldsymbol{h}^T\boldsymbol{x}_k+\boldsymbol{s}_k-\boldsymbol{t}\\ \boldsymbol{a}^T\boldsymbol{x}_k-\boldsymbol{b}\\ \boldsymbol{\mu}_k\boldsymbol{s}_k-\boldsymbol{\tau}_k1\end{bmatrix} \tag{A-16}$$

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

**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。刚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

积极集法属于图形法在 QP 问题上的扩展，其通过求解有限个等式约束 QP 问题来解决一般约束下的 QP 模型。当不等式约束条件不多时，也是一种高效的求解 QP 算法。积极集法首先将 QP 中所有的不等式约束视为等式约束。把不等式约束直接转成等式约束当然是存在问题的，不等式约束存在有效和无效两种情况，而有效无效很容易通过该不等式对应的拉格朗日乘子进行判断。不等式约束的互补松弛条件告诉我们，不等式对应的拉格朗日乘子应当满足 $\lambda_i \geqslant 0$。

**Notes A.2.4.** $\lambda_i \geqslant 0$ 这个松弛条件哪里提到的？得查一查。

在第 $k$ 次迭代开始时，我们首先检查当前迭代点 $\boldsymbol{x}_k$ 是否为当前工作集 $\mathbb{W}_{\mathbb{k}}$（有效不等式约束集合）下的最优点。如果不是，我们就通过求解一个等式约束的 QP 命题来得到一个前进方向 $p$。在计算 $p$ 的时候，只关注 $\mathbb{W}_{\mathbb{k}}$ 中的不等式约束并将其转化为等式约束，而忽略其他不等式约束。令 $\boldsymbol{d} = \boldsymbol{x} - \boldsymbol{x}_k$，带入原命题得：

$$\begin{aligned} min \quad & \frac{1}{2}(\boldsymbol{d}+\boldsymbol{x}_k)^T \boldsymbol{H}(\boldsymbol{d}+\boldsymbol{x}_k) + \boldsymbol{g}^T(\boldsymbol{d}+\boldsymbol{x}_k) \\ s.t. \quad & \boldsymbol{a}_i^T(\boldsymbol{d}+\boldsymbol{x}_k) = b_i, \quad i \in \mathbb{W}_{\mathbb{k}} \end{aligned} \tag{A-17}$$

**Notes A.2.5.** 这块暂时不继续花时间了，理解问题就可以了。另外也感觉这部分讲的不是很清楚。