

# 运动控制方案

## 实例项目——六足机器人（Spider Robot）

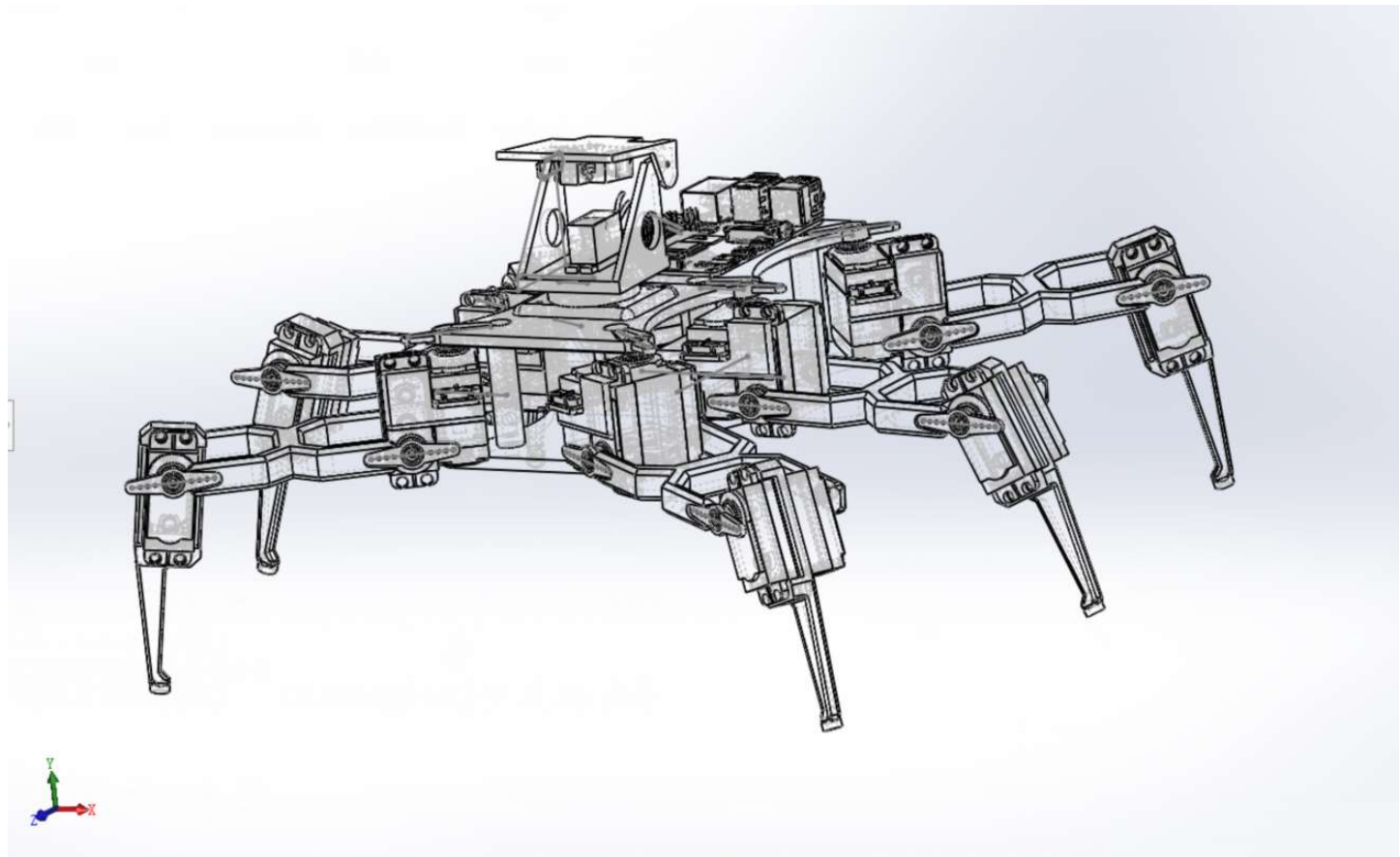
### 写在方案前的话

本方案根据**严密数学建模**和**工程快速实现**的方面不同准备有两套不同操作方案，希望了解严密数学建模的朋友请耐心看完全文，有迫切工程实现需求的朋友可在看完第1点简介后直接，从第五点开始查看，需要说明的是，两种方式仅在学习的时间成本上会有明显不同，在具体项目实现的时间上会因为个人的熟练程度的因素不一定会展现出差异

### 目录

- 1.本项目六足机器人简介
- 2.常用机械臂模型——DH模型的基本思想
- 3.DH模型提出的数学原理
- 4.本项目六足机器人的单足（单机械臂）建模方式
- 5.本项目六足机器人的多足（多机械臂）运动逻辑——步态规划
- 6.项目工程实现方式及相关软件
- 7.注意要点

### 1.本项目六足机器人简介



#### 项目SolidWorks预览图

在运动控制方面，本项目六足机器人使用树莓派作为主控板，使用24路舵机控制板作为18个舵机（6足：每足3个）的运动控制，并且使用降压芯片将7.4V电池电压降为6.2V电压给舵机控制板供电；在舵机选用上，使用最为常见的MG995舵机，并使用Lobot Servo Control软件对18个舵机同一管理。

## 2.常用机械臂模型——DH模型的基本思想

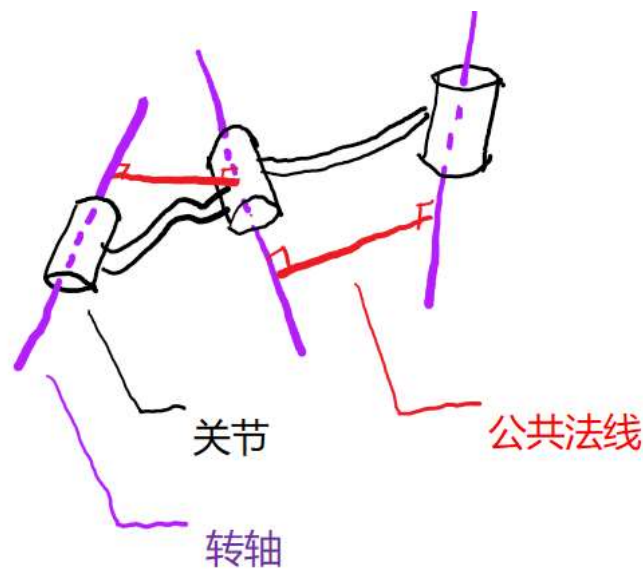
DH模型，即Denavit-Hartenberg模型，是连杆结构常用的建模方式，可以运用于任何机器人的运动构型，特别是在多自由度机械臂上运用广泛。

在DH模型中，一个多自由度的连杆结构（或机械臂）是由一个个**关节**和**连杆**组成的，其中每个关节只有一个自由度，根据运动形式的不同，关节可分为滑动关节（P关节）和转动关节（R关节）。

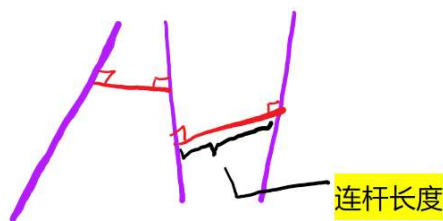
这里我们引入两个概念：**正运动学**和**逆运动学**，**正运动学**即是指已知各个关节的运动参数，求末端执行器相对于参考坐标的位姿，**逆运动学**是指根据已给定的满足工作要求的末端执行器相对于末端执行器的位置和姿态，求各个关节的运动参数。以正运动学为背景的话，DH模型通过限制**原点位置**和**X轴方向**，可以人为的减少2个参数，所以只需要4个参数就可以表达关节之间原本6参数才能表达的坐标变换。

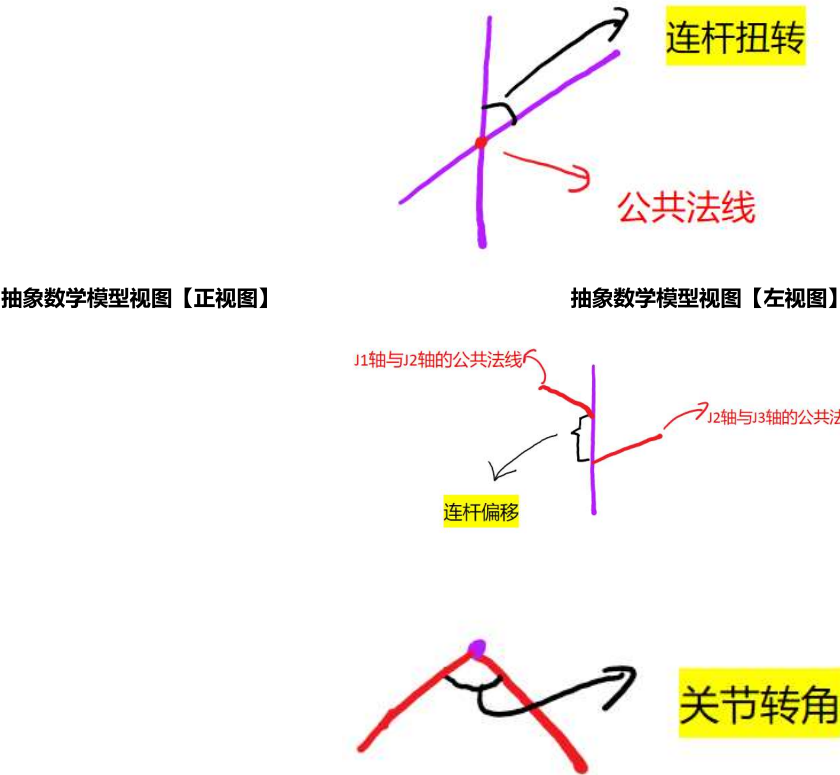
## 3.DH模型提出的数学原理

- DH模型的四个核心参数
  - 连杆长度（link length）：两个关节之间公共法线长度，以符号 $a_{i-1}$ 表示，  
【对于关节 $O_{i-1}$ 和 $O_i$ 】
  - 连杆扭转（link twist）：一个关节的轴相对于另一个关节的轴绕他们公共法线的旋转角度,以符号 $\alpha_{i-1}$ 表示  
【对于关节 $O_{i-1}$ 和 $O_i$ 】
  - 连杆偏移（link offset）：一个关节余下一个关节的公共法线和这个关节与它上一个关节公共法线沿这个关节轴的距离，以符号 $d_i$ 表示  
【对于轴 $X_{i-1}$ 和 $X_i$ 】
  - 关节转角（joint angel）：一个关节与下一个关节公共法线和它与上一个关节的公共法线绕这个关节的转角，以符号 $\theta_i$ 表示  
【对于轴 $X_{i-1}$ 和 $X_i$ 】
- 模型图像



任意一个拥有三个关节和两根连杆的机械结构【真实视图】





抽象数学模型视图【正视图】

抽象数学模型视图【正视图】

(去掉1, 3号关节转轴)

(去掉1, 3号关节转轴)

• 矩阵计算——涉及空间坐标变换

空间坐标变换

二维图形变换【旧坐标 $(x,y)$ ——>新坐标 $(X,Y)$ 】

三维图形变换【旧坐标 $(x,y,z)$ ——>新坐标 $(X,Y,Z)$ 】

平移

缩放（此处不考虑）

旋转

平移

缩放（此处不考虑）

旋转

为方便计算，在这里我们引入齐次坐标，齐次坐标是一种标记方式

- 对于二维坐标点来说，原来为  $(x,y)$  的坐标表示为  $(x,y,1)$
- 对于二维坐标点来说，原来为  $(x,y,z)$  的坐标表示为  $(x,y,z,1)$
- 齐次坐标可以在空间变换计算中区分向量和坐标，如二维的：

向量：

坐标：

$(x,y,0)$

$(x,y,1)$

二维平移变换【引入齐次坐标后，等待变换的点：  $(x,y,1)$  】

变换公式

变换矩阵：

$\begin{cases} X = x + \delta x \\ Y = y + \delta y \end{cases}$

$\begin{bmatrix} 1 & 0 & \delta x \\ 0 & 1 & \delta y \\ 0 & 0 & 1 \end{bmatrix} * (x,y,1) = [x + \delta x, y + \delta y, 1]$

二维旋转变换【引入齐次坐标后，等待变换的点：  $(x,y,1)$  】

变换公式

变换矩阵：

$\begin{cases} X = x * \cos\theta - y * \sin\theta \\ Y = y * \cos\theta + x * \sin\theta \end{cases}$

$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} * (x,y,1) = [x * \cos\theta - y * \sin\theta, y * \cos\theta + x * \sin\theta, 1]$

推得各个变换矩阵（左乘坐标变换）：  
(证明为三角变换内容，此处省略)  
二维平移变换矩阵：

$$\begin{bmatrix} 1 & 0 & \delta x \\ 0 & 1 & \delta y \\ 0 & 0 & 1 \end{bmatrix}$$

二维旋转变换矩阵：

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

三维同理：  
三维维平移变换矩阵：

$$\begin{bmatrix} 1 & 0 & 0 & \delta x \\ 0 & 1 & 0 & \delta y \\ 0 & 0 & 1 & \delta z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

三维旋转变换矩阵：  
若绕X轴：

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

若绕Y轴：

$$\begin{bmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

若绕Z轴：

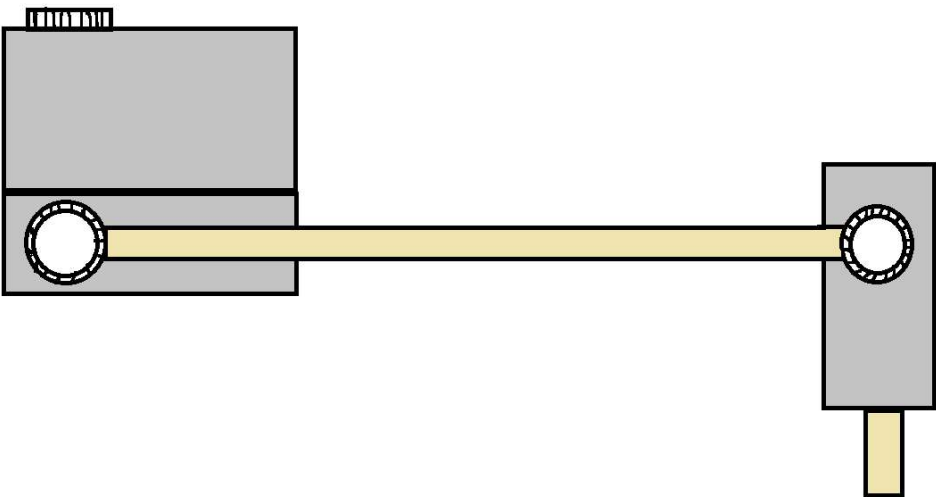
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

对于我们研究的六足机器人项目（三维结构），根据每两个关节之间的的旋转变换矩阵（ $R_x$ 或 $R_y$ 或 $R_z$ ）和平移变换矩阵（ $T$ ）相乘再与原坐标左乘，就可得到下一个点的坐标，递推直到获得末端足姿坐标。**这样一来，我们便实现了由已知各个关节的运动参数，求得末端执行器相对于参考坐标的位姿【由于推导比较麻烦，为回避计算量，下一板块会直接给出两关节间化简后的的变换矩阵，在填好DH表格后，可直接由数据计算坐标】**

## 4.本项目六足机器人的单足（单机械臂）建模方式——正运动学与逆运动学

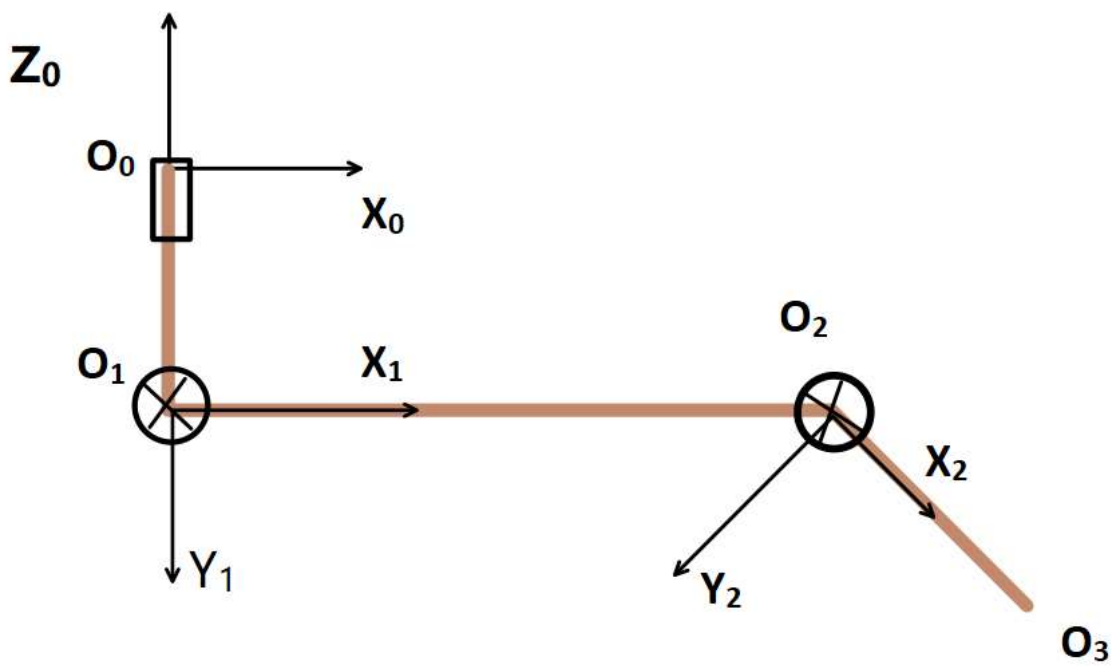
### 正运动学分析

#### ①物理结构抽象



项目单臂机械结构示例图

#### ②数学抽象



项目数学模型抽象示例图

- 对于一个机械臂的每一个关节 指定一个本地参考坐标系，即指定一个X轴和一个Z轴：其中Z轴以右手螺旋定则按照旋转方向规定正方向，X轴分两种情况：若两关节轴不平行或相交，用最短公垂线为X轴；平行，用无数条相等的公垂线中任选一条为X轴

③明确参数，填写DH表格

i	$a_{i-1}$	$\alpha_{i-1}$	$d_i$	$\theta_i$
1				
2				
3				

④将表格数据代入两关节间化简后转换矩阵公式

$$T_{i-1} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_{i-1} \\ \sin\theta_i\cos\alpha_{i-1} & \cos\theta_i\cos\alpha_{i-1} & -\sin\alpha_{i-1} & -\sin\alpha_{i-1}d_i \\ \sin\alpha_{i-1}\sin\theta_i & \cos\theta_i\sin\alpha_{i-1} & \cos\alpha_{i-1} & \cos\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

⑤将得到的变换矩阵T<sub>0</sub>、T<sub>1</sub>、T<sub>2</sub>与原始坐标（0,0,0,1）相乘得到足端位姿坐标：

$$(X,Y,Z,1) = T_2 * T_1 * T_0 * (0,0,0,1)$$

逆运动学分析

不同与正运动学清晰的矩阵计算，逆运动学的处理显得更为复杂，逆运动学的求解一般有**数值解计算**和**封闭形式解决方案**。由于本项目并非多自由度的工业机械臂一类的应用场景，可以通过之后描述的方案二使用**人为试错**的方式获得舵机转角值，在此不对逆运动学解做具体阐述，只是保证知识完整性做相关说明。

在实际运用中，我们并不是通过已知的各舵机转角来获得末端位姿。相反，我们需要在已知目标末端位姿（x,y,z,1）的前提下，获得每个舵机的转角以输入程序使得舵机带动机械臂末端到达目标位置。可以说，逆运动学在实际中的运用要多于正运动学。

具体的求解方法可参考CSDN博客：

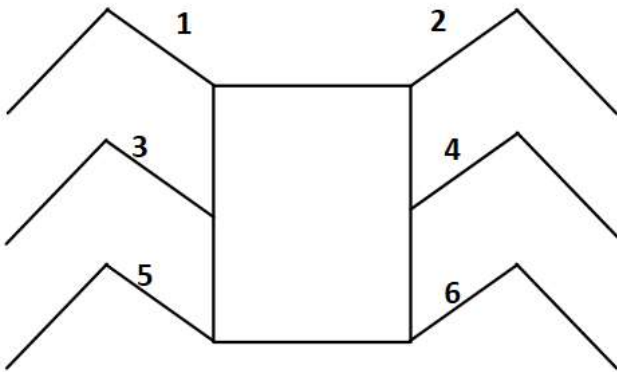
网址链接：[尘埃落定123——《逆运动学》](#)

5.本项目六足机器人的多足（多机械臂）运动逻辑——步态规划

为方便本栏的说明，我们需要先引入一些概念

- 步态：指机器人的每条腿按一定顺序和轨迹的运动过程
- 步态周期：多足机器人完成一个步态的时间，所有腿轮番完成“提起——摆动——放下”所花费的时间
- 占地系数：每条腿接触地面的时间和整个步态周期的比值
  - 【当占地系数小于0.5】六足机器人任何瞬间只有不到三条腿支撑地面，称为**跳跃步态**
  - 【当占地系数等于0.5】机器人用两组腿交替摆动，称为小跑**跳跃步态**
  - 【当占地系数大于0.5】机器人轮番有三条腿以上支撑地面，称为爬行**跳跃步态**
- 步幅：机器人重心在一个步态周期的平移

为方便后续步态说明，我们引入一般情况下的六足机器人简图如图所示，有前足1、2，中足3、4，后足5、6。



引入一般情况下的六足机器人简图

常见的步态

- 三角步态：即是将6条足分为两组，【1，4，5】为A组，【2，3，6】为B组，通过“在A组迈腿行走时，B组当作三角形稳定支架；B组迈腿行走时，A组当三角形稳定支架，来进行移动”，**其行进的方式是直线运动+原地自转**。三角步态是自然界“六足纲”昆虫最常用的方式。三角步态的优点是操控逻辑简单，运行平稳，故障率低；缺点是只适用于平整、路况良好地面，对于复杂山地或多石泥泞地形适应性不强。
- 跟导步态：也即是使中足（3，4）和后组（5，6）跟导两对前足（1，2），步态行走形式只由前两足**落点**决定。由于会根据每次前足落点的不同调节步态，而不是像三角步态一样固定行进，因此对于复杂地形的适应能力也更强；且由于只关注两个前足的落点作为输入就可以给出所有其余足的控制，因此其控制逻辑也不难。
- 交替步态：此步态以单独的一条足为基本单元，控制逻辑是当一条足的相邻两足均已经触地，该足才被允许抬起前进，同样在其落地时，会给相邻两组发出已触地的信号，不同的足顺次运动。可分为平整地形和复杂地形来讨论，在平整地形上，其表现效果与三角步态与三角步态相似；对



于复杂地形，步态的运动方式是不可预测的。

在本项目中我们选择我们选择最为常用的三角步态，即是六足分为两组轮流充当支撑组和移动组

## 6.项目工程实现方式及相关软件

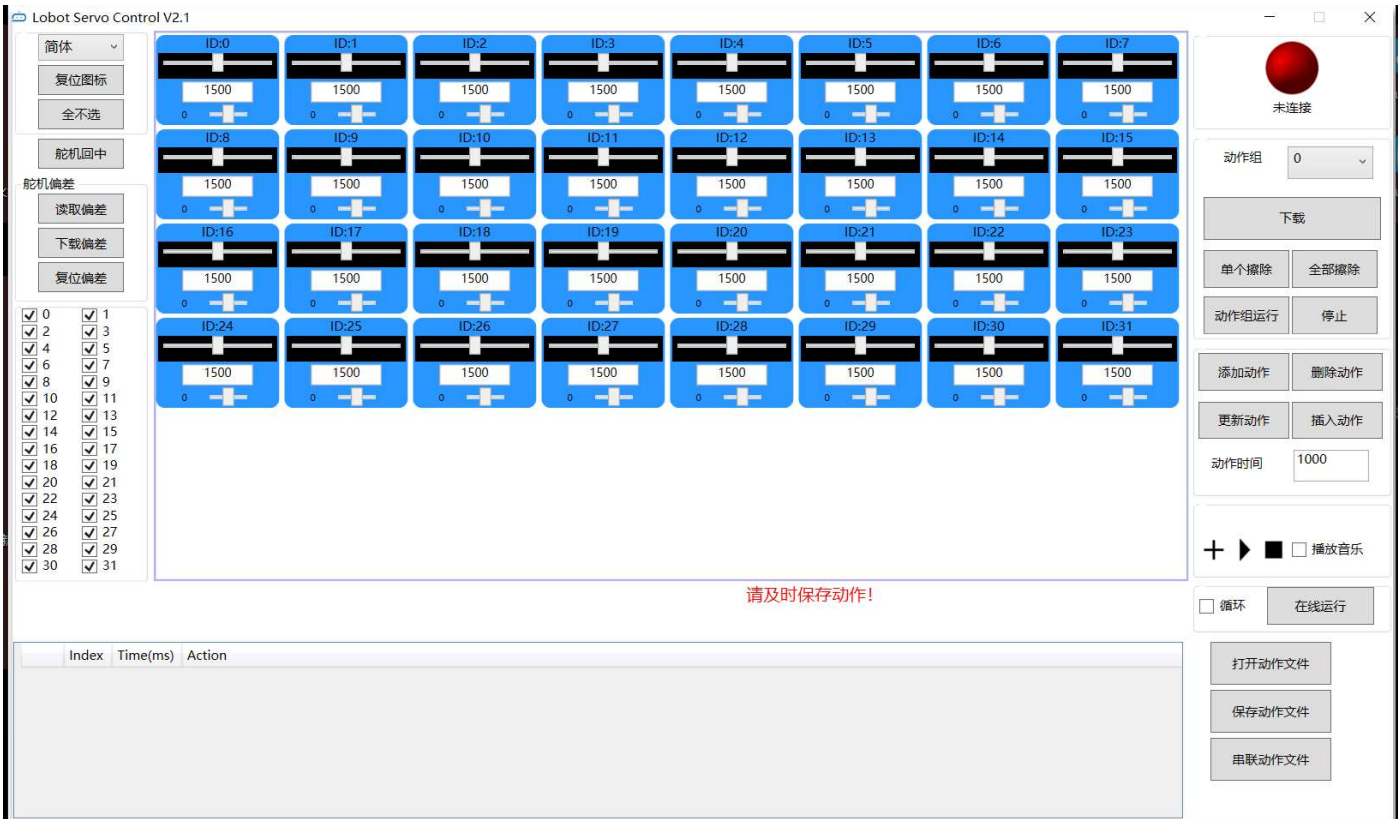
本项目使用Lobot Servo Control对18个舵机以**动作组**的形式控制。简单来说，我们把舵机的要完成的复杂操作称为动作组，而将动作组拆分则可以得到**单一动作**。所谓单一动作，指的是**当舵机控制板发出一个操作命令后，所有18个舵机所处的旋转稳定后的空间状态**。

举个例子，如我想先**使1，2足向前伸，3，4号足不动，然后5，6号足向后伸**。而这就称之为一个**运动组**，因为有时先后的关系，所以这个运动组包含，**使1，2足向前伸，3，4号足不动，5，6号足向后伸**这3个**单一动作**，分别受3条指令控制。我们要实现这个动作组需要先设置舵机位置使1，2号机械臂的6个舵机的目标值，使其伸直，这是第一条指令；在第二条指令中，我们完全保持现有的舵机状态，即所有舵机的目标值不变；第三条指令中，我们设置舵机位置使5，6号机械臂的6个舵机的目标值使5，6号机械臂后伸，这样就完成一个动作组。

而通过对于不同动作组的组合，我们便可以使六足机器人做出不同的行为。这就好比人走路有许多重复性的迈步，我们细分的话，将走路这一行为由不断重复地**迈左脚，然后迈右脚**的这一动作组组成，而在这个动作组当中包含具有先后顺序的**迈左脚**和**迈右脚**这两个动作。而本项目六足机器人的运动逻辑正是基于这一思路提出来的。

以下是两种不同的方案，其中方案一因为在前面的介绍中做了较为具体的阐述，在此就不过多说明。

- 方案一  
通过矩阵计算，可以达到的效果是,在获得需要到达的末端位姿坐标后，可以通过计算获得一个“最近”的接近末端目标点的3个多级旋转角度的数值解，原理上即使之前的逆运动学原理。
- 方案二  
上述的操作方式固然在数学上更具有严密性，但是我们知道，虽然我们能保证矩阵计算的准确性，但在实际过程中，测量误差、机械精度等问题是无法避免的，我们需要做细微调整，使运动效果更好，那么既然运动效果可以实时调整，且调节本身也是可视化的，我们可不可以直接“目测效果”调试呢？答案是可以的。



通过对六足机器人三角步态拆分，分为直线运动和原地自转，得到不同运动组，在分别编写（记录舵机位置）不同单一运动，我们同样可以使六足机器人顺利行走。

需要指出的时，上述的方案二是一种“力大砖飞”的方式，对于精度要求不高，且试错成本较低甚至没有试错成本的场合是适用的，如本项目当中，其本质是以人脑作为反馈调节机构，对错误进行修正，是“摸着石头过河”，不建议过于依赖，且之所以能够选择方案二是由于本项目的立项背景是应用于平整路面，也基于此选择了 **三角步态**，所以对于每一次步幅的增加，其控制代码没有差异；而对于精度要求高，试错成本大，需要其他步态模式以应对更加复杂地形的场景时，或难以使人直接对执行效果进行评估或修正的系统时，应对机械臂进行数学建模，通过正运动学和逆运动学的理论，为其建立准确的理论模型再进行工程实现。

## 7.注意要点

- 注意在实际代码编写中，各个舵机的中值可能会有不同，建议先进行统一调试

本笔记由Panzer-Crow(Github)整理

鸣谢本项目中的两位队友：stratosphericus(Github),balmung(Github)