

机械手臂的逆运动学解

正运动学分析是已知每个关节的姿态的前提下，解算出末端执行器的姿态。而逆运动学研究的问题是，要求控制末端执行器到达某一位置时，各关节应处于什么姿态。

1. 逆运动学概述

逆运动学最基本的思路，是从正运动学反过来，对于我们的机械臂而言，也就是已知末端执行器的位置和朝向，求每个关节的角度。我们的机械臂的三维运动是比较复杂的，这里为了简化模型更加便于大家的理解，我们对模型进行精简，先去掉下方云台的旋转关节，这样就可以在二维的平面上进行运动学分析了。

其中 $\theta_0, \theta_1, \theta_2$ 是各个关节的角度，未知量。 $P(x, y, \alpha)$ 是末端执行器的位姿表示， x 和 y 是在 OXY 平面的坐标， α 是末端执行器的朝向。一般我们看的文章，计算运动学分析那都是一大堆的矩阵变换，实际上那些算法在单片机上执行是比较费时的，甚至可以说是无法满足实时控制的。适合在 `matlab` 上面运算。所以，我们需要寻找一种高效的方案，以满足我们实时控制的需要。下面我们将使用几何法进行分析。

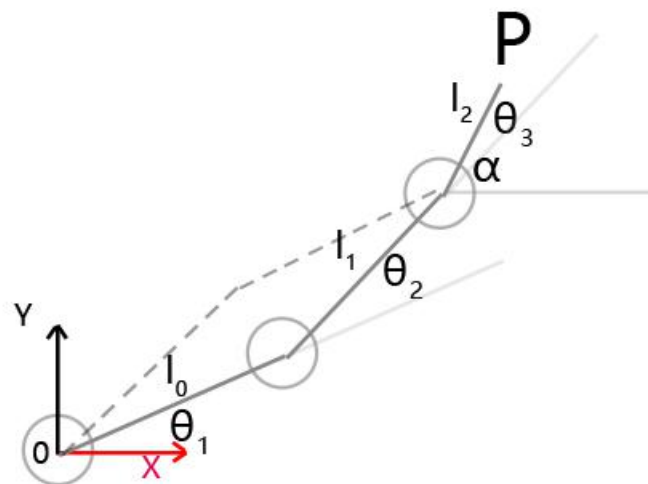


图 1. 机械臂的几何表示

根据上面的图示，很容易列出如下方程：

$$x = l_0 \cos \theta_1 + l_1 \cos(\theta_1 + \theta_2) + l_2 \cos(\theta_1 + \theta_2 + \theta_3) \quad (1)$$

$$y = l_0 \sin \theta_1 + l_1 \sin(\theta_1 + \theta_2) + l_2 \sin(\theta_1 + \theta_2 + \theta_3) \quad (2)$$

$$\alpha = \theta_1 + \theta_2 + \theta_3 \quad (3)$$

实际上，这也是运动学正解的一种表达方式，既然可以如此简单的表达，为什么要之前要学习 DH 模型呢？主要是我们需要通过这种机构掌握一种通用的方法，后面解决更复杂的模型分析的时候能够游刃有余。毕竟几何法不具备通用性，每个模型都需要单独的分析，一些超出我们感性认识的复杂模型更是无能为力。

下面对上述方程组进行化简，把式(3)代入式(2)和(1)中，得：

$$x = l_0 \cos \theta_1 + l_1 \cos(\theta_1 + \theta_2) + l_2 \cos(\alpha) \quad (4)$$

$$y = l_0 \sin \theta_1 + l_1 \sin(\theta_1 + \theta_2) + l_2 \sin(\alpha) \quad (5)$$

为了方便计算，令

$$m = l_2 \cos(\alpha) - x$$

$$n = l_2 \sin(\alpha) - y$$

化简(4)、(5)，得：

$$l_1 = (l_0 \cos \theta_1 + m)^2 + (l_0 \sin \theta_1 + n)^2 \quad (6)$$

通过简单的运算，可解：

$$\sin \theta_1 = (-b \pm \sqrt{b^2 - 4ac}) / 2a$$

其中：

$$k = (l_0^2 - l_1^2 - m^2 - n^2) / 2l_1$$

$$a = m^2 + n^2$$

$$b = -2nk$$

$$c = k^2 - m^2$$

通过同样的方法，可求得 θ_2 ，这样就完成了逆运动学的计算。显然，最终有 2 个正确的解，这个根据图 1 的虚线部分可以看出。一般我们选取的是虚线部分的解，这样每个关节的受力可以小一点。下面是 matlab 的实现代码：

```
clear;
clc;
syms theta1 theta2 theta3 alpha;
syms l0 l1 l2 ;
syms x y m n k a b c;
l0=0.105
```

```
l1=0.078
l2=0.055
%目标值
x=l0
y=l1+l2
alpha=pi/2;
m=l2*cos(alpha)-x
n=l2*sin(alpha)-y
k=(l1^2-l0^2-m^2-n^2)/2/l0
a=m^2+n^2
b=-2*n*k
c=k^2-m^2
(-b-sqrt(b^2-4*a*c))/2/a
k=(l0^2-l1^2-m^2-n^2)/2/l1;
a=m^2+n^2;
b=-2*n*k;
c=k^2-m^2;
(-b+sqrt(b^2-4*a*c))/2/a
```

显然这么简单的算法，使用 matlab 已经是杀鸡用牛刀，我们还编写了 C 语言的代码，可以在单片机上面运行的，如下：

```
/******
函数功能：数学模型
入口参数：末端执行器位姿态
返回 值：无
******/
void Kinematic_Analysis(float x,float y,float Beta,float Alpha)
{
float m,n,k,a,b,c,theta1,theta2,theta3,s1ps2;
m=l2*cos(Alpha)-x;
n=l2*sin(Alpha)-y;
k=(l1*l1-l0*l0-m*m-n*n)/2/l0;
a=m*m+n*n;
b=-2*n*k;
c=k*k-m*m;
theta1=(-b+sqrt(b*b-4*a*c))/2/a;
theta1=asin(theta1)*180/PI;
if(theta1>90)theta1=90;
if(theta1<-90)theta1=-90;
k=(l0*l0-l1*l1-m*m-n*n)/2/l1;
a=m*m+n*n;
b=-2*n*k;
c=k*k-m*m;
s1ps2=(-b-sqrt(b*b-4*a*c))/2/a;
```

```
slps2=asin(slps2)*180/PI;

if(slps2>90)theta2=90;
if(slps2<-90)theta2=-90;
theta2=slps2-theta1;
if(theta2>90)theta2=90;
if(theta2<-90)theta2=-90;
theta3=Alpha*180/PI-theta1-theta2;
if(theta3>90)theta3=90;
if(theta3<-90)theta3=-90;
Target1  = 750-(Beta)*Ratio;
Target2  = 735+(theta1-90)*Ratio;
Target3  = 717-(theta2)*Ratio;
Target4  = 702-(theta3)*Ratio;;
Target5  = 750;
}
```