一、参考资料

机器人算法：

（备份）（备份）机器人学 机器视觉与控制 MATLAB算法基础(带书签).pdf

机器人建模和控制.pdf

机器人学.蔡自兴.pdf 第三版

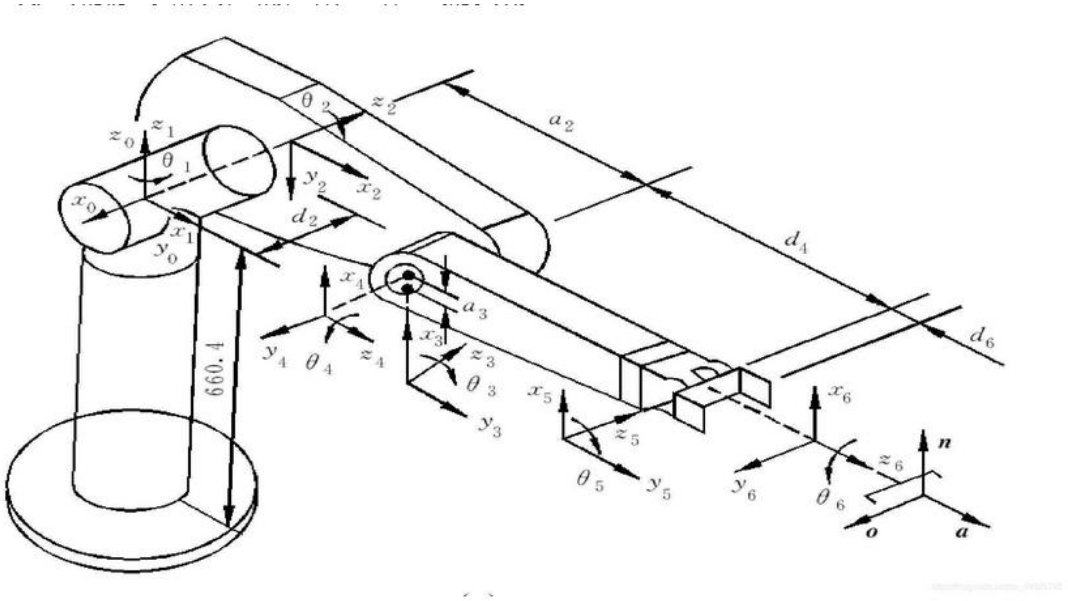
Robotics Toolbox for MATLAB 版本 10.4 函数 fkine() 及 ikine()源码

DSP 库：

安富莱_STM32-V5开发板_第2版DSP数字信号处理教程（V2.7）.pdf

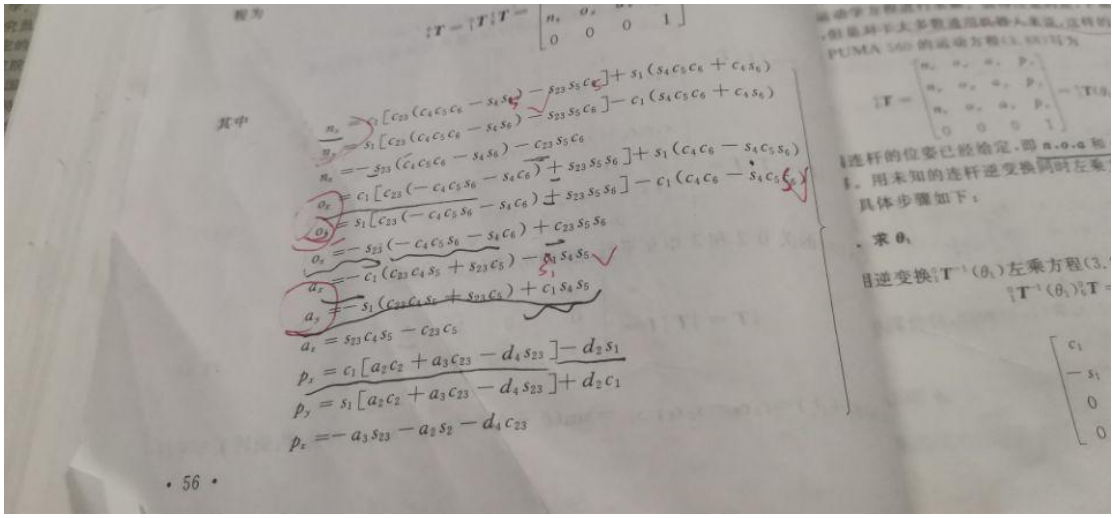
二、算法实现

1. 对机械臂进行参数辨识，DH 参数，机械臂结构等 （本例程基于 PUMA560 6 转动关节 无偏置量 ）；



2. 根据步骤 1 所得参数依次算出每个连杆变换矩阵 TN； C 语言中采用结构体，如下图所示。（书中公式有误，红色笔迹已更正）

```c
/**
 * @brief Instance structure for the floating-point matrix structure.
 */
typedef struct
{
  uint16_t numRows;     /**< number of rows of the matrix.     */
  uint16_t numCols;     /**< number of columns of the matrix.  */
  float32_t *pData;     /**< points to the data of the matrix. */
} arm_matrix_instance_f32;
```



$$
\begin{aligned}
n_x &= c_1[c_{23}(c_4c_5c_6 - s_4s_6) - s_{23}s_5c_6] + s_1(s_4c_5c_6 + c_4s_6)\\
n_y &= s_1[c_{23}(c_4c_5c_6 - s_4s_6) - s_{23}s_5c_6] - c_1(s_4c_5c_6 + c_4s_6)\\
n_z &= -s_{23}(c_4c_5c_6 - s_4s_6) - c_{23}s_5c_6\\
o_x &= c_1[c_{23}(-c_4c_5s_6 - s_4c_6) + s_{23}s_5s_6] + s_1(c_4c_6 - s_4c_5s_6)\\
o_y &= s_1[c_{23}(-c_4c_5s_6 - s_4c_6) + s_{23}s_5s_6] - c_1(c_4c_6 - s_4c_5s_6)\\
o_z &= -s_{23}(-c_4c_5s_6 - s_4c_6) + c_{23}s_5s_6\\
a_x &= -c_1(c_{23}c_4s_5 + s_{23}c_5) - s_1s_4s_5\\
a_y &= -s_1(c_{23}c_4s_5 + s_{23}c_5) + c_1s_4s_5\\
a_z &= s_{23}c_4s_5 - c_{23}c_5\\
p_x &= c_1[a_2c_2 + a_3c_{23} - d_4s_{23}] - d_2s_1\\
p_y &= s_1[a_2c_2 + a_3c_{23} - d_4s_{23}] + d_2c_1\\
p_z &= -a_3s_{23} - a_2s_2 - d_4c_{23}
\end{aligned}
\tag{3.6}
$$

3. 将连杆矩阵此次相乘得到最终各元素的表达式;三角函数采用 dsp 库 arm_cos_f32、arm_sin_f32.

```
 */
void fkine(mat *T0, float *q6 ,float32_t *inter_value)
{
    inter_value[0] = arm_cos_f32(q6[0]) * ( arm_cos_f32(q6[1] + q6[2]) * ( arm_cos_f32(q6[3]) * arm_cos_f32(q6[4]) * arm_cos_f32(q6[5]) - arm_sin_
    inter_value[1] = arm_cos_f32(q6[0]) * ( arm_cos_f32(q6[1] + q6[2]) * (-arm_cos_f32(q6[3]) * arm_cos_f32(q6[4]) * arm_sin_f32(q6[5]) - arm_sin_
    inter_value[2] = -arm_cos_f32(q6[0]) * ( arm_cos_f32(q6[1] + q6[2]) * arm_cos_f32(q6[3]) * arm_sin_f32(q6[4]) + arm_sin_f32(q6[1] + q6[2]) * a
    inter_value[3] = arm_cos_f32(q6[0]) * ( dh_[1] * arm_cos_f32(q6[1])  + dh_[2] * arm_cos_f32(q6[1] + q6[2]) - dh_[4] * arm_sin_f32(q6[1] + q6[2

    inter_value[4] = arm_sin_f32(q6[0]) * ( arm_cos_f32(q6[1] + q6[2]) * (arm_cos_f32(q6[3]) * arm_cos_f32(q6[4]) * arm_cos_f32(q6[5]) - arm_sin_f
    inter_value[5] = arm_sin_f32(q6[0]) * ( arm_cos_f32(q6[1] + q6[2]) * (-arm_cos_f32(q6[3]) * arm_cos_f32(q6[4]) * arm_sin_f32(q6[5]) - arm_sin_
    inter_value[6] = -arm_sin_f32(q6[0]) * ( arm_cos_f32(q6[1] + q6[2]) * arm_cos_f32(q6[3]) * arm_sin_f32(q6[4]) + arm_sin_f32(q6[1] + q6[2]) * a
    inter_value[7] = arm_sin_f32(q6[0]) * ( dh_[1] * arm_cos_f32(q6[1])  + dh_[2] * arm_cos_f32(q6[1] + q6[2]) - dh_[4] * arm_sin_f32(q6[1] + q6[2

    inter_value[8] = (-arm_sin_f32(q6[1] + q6[2]) * ( arm_cos_f32(q6[3]) * arm_cos_f32(q6[4]) * arm_cos_f32(q6[5]) - arm_sin_f32(q6[3]) * arm_sin_
    inter_value[9] = (-arm_sin_f32(q6[1] + q6[2]) * ( -arm_cos_f32(q6[3]) * arm_cos_f32(q6[4]) * arm_sin_f32(q6[5]) - arm_sin_f32(q6[3]) * arm_cos
    inter_value[10] = ( arm_sin_f32(q6[1] + q6[2]) * arm_cos_f32(q6[3]) * arm_sin_f32(q6[4]) - arm_cos_f32(q6[1] + q6[2]) * arm_cos_f32(q6[4]) );
    inter_value[11] = - dh_[2] * arm_sin_f32(q6[1] + q6[2]) - dh_[1] * arm_sin_f32(q6[1]) - dh_[4] * arm_cos_f32(q6[1] + q6[2]);
    arm_mat_init_f32(T0, matrow, matcol, inter_value);
}
```

4. 逆解部分理论基础见蔡自兴《机器人学》P57 -60 页;按照公式依次写出 4 组可能解得表达式，最后使机器人臂腕关节翻转得到另外四组解；

如果机械结构、DH 参数不同于 PUMA560，重新建立各连杆矩阵，再依次相乘得到各关节角未知数所对应的含未知数变换矩阵。利用矩阵各元素相等，建立关节角与已知量的函数关系，再采用双变量反正切函数求得关节角。（不采用反正弦或反余弦函数原因详见 蔡自兴《机器人学》P50 页）

```
/**
 * @description:  逆解 可以优化的地方：乘法 采用DSP 指令 而不是 “ * ”，反正切函数采用最新DSP库而不是C标准库或者使用查表法
 * @param {mat} T 传入连杆变换矩阵
 * @param {float} qn 得到 八组逆解
 * @return {*}
 */
```

```
    qn[0][5] = atan2f(-nx * (COS(qn[0][0]) * COS(qn[0][1] + qn[0][2]) * SIN(qn[0][3]) - SIN(qn[0][0]) * COS(qn[0][3]
    qn[1][5] = atan2f(-nx * (COS(qn[1][0]) * COS(qn[1][1] + qn[1][2]) * SIN(qn[1][3]) - SIN(qn[1][0]) * COS(qn[1][3]
    qn[2][5] = atan2f(-nx * (COS(qn[2][0]) * COS(qn[2][1] + qn[2][2]) * SIN(qn[2][3]) - SIN(qn[2][0]) * COS(qn[2][3]
    qn[3][5] = atan2f(-nx * (COS(qn[3][0]) * COS(qn[3][1] + qn[3][2]) * SIN(qn[3][3]) - SIN(qn[3][0]) * COS(qn[3][3]

/*
* 臂腕关节翻转得到另外四组解
*/
    qn[4][0] = qn[0][0];
    qn[4][1] = qn[0][1];
    qn[4][2] = qn[0][2];
    qn[4][3] = qn[0][3] + PI;
    qn[4][4] = - qn[0][4];
    qn[4][5] = qn[0][5] + PI;

    qn[5][0] = qn[1][0];
    qn[5][1] = qn[1][1];
    qn[5][2] = qn[1][2];
    qn[5][3] = qn[1][3] + PI;
    qn[5][4] = - qn[1][4];
    qn[5][5] = qn[1][5] + PI;

    qn[6][0] = qn[2][0];
    qn[6][1] = qn[2][1];
    qn[6][2] = qn[2][2];
    qn[6][3] = qn[2][3] + PI;
    qn[6][4] = - qn[2][4];
```

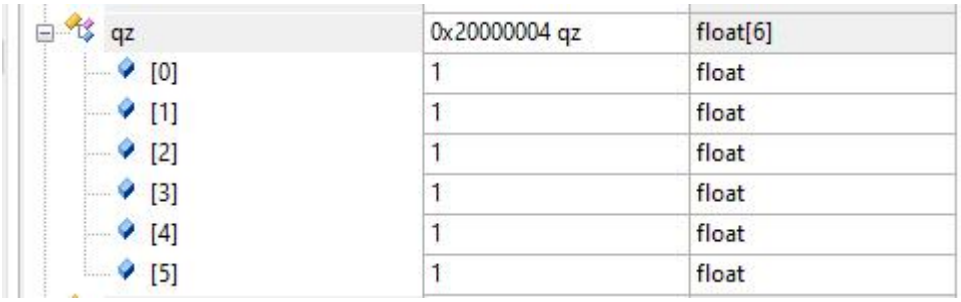
（dsp 求解 sin cos 函数使用 SIN COS 宏替代）

5.如果不需要所有解，可以函数体后面添加筛选解的内容。

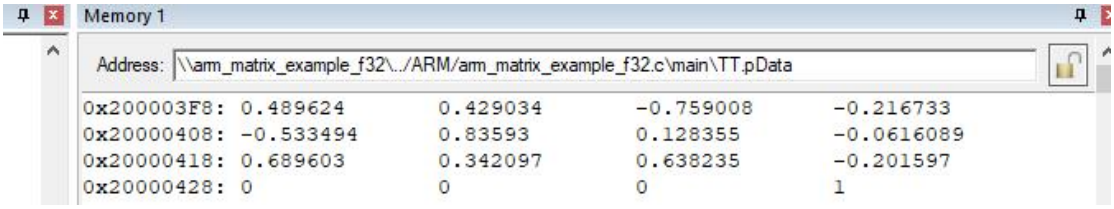
三、正确性验证

验证步骤：

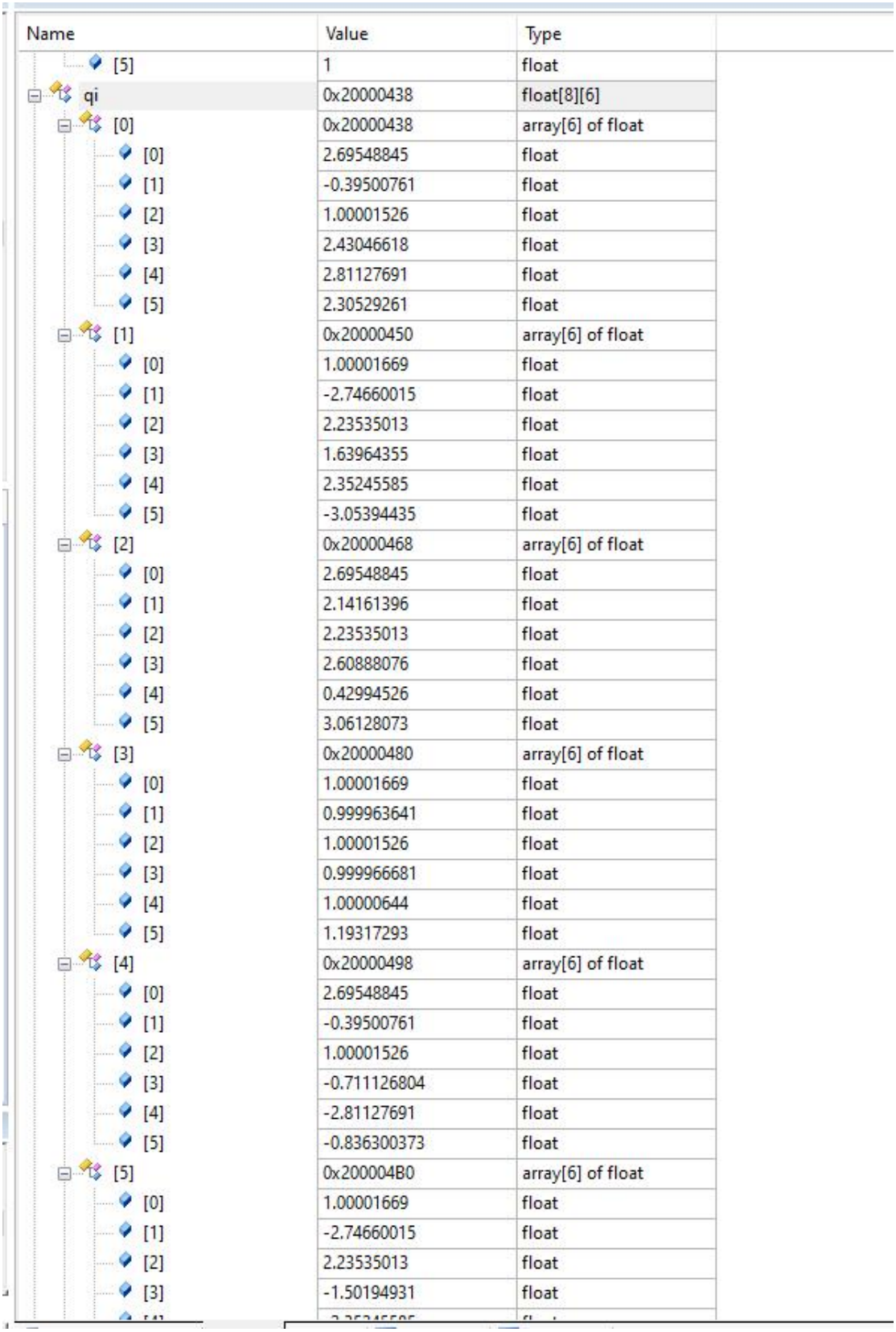
1. 输入任意关节角组合 qn【6】得到变换矩阵 TT ；



qz	0x20000004 qz	float[6]
[0]	1	float
[1]	1	float
[2]	1	float
[3]	1	float
[4]	1	float
[5]	1	float



2. 输入步骤 1 所得变换矩阵 TT 和 目标解所存储数组 qi[8][6],得到逆解。观察可以看出 8 组逆解中有一组即为我们在步骤 1 中输入的关节角组合；



Name	Value	Type
[5]	1	float
qi	0x20000438	float[8][6]
[0]	0x20000438	array[6] of float
[0]	2.69548845	float
[1]	-0.39500761	float
[2]	1.00001526	float
[3]	2.43046618	float
[4]	2.81127691	float
[5]	2.30529261	float
[1]	0x20000450	array[6] of float
[0]	1.00001669	float
[1]	-2.74660015	float
[2]	2.23535013	float
[3]	1.63964355	float
[4]	2.35245585	float
[5]	-3.05394435	float
[2]	0x20000468	array[6] of float
[0]	2.69548845	float
[1]	2.14161396	float
[2]	2.23535013	float
[3]	2.60888076	float
[4]	0.42994526	float
[5]	3.06128073	float
[3]	0x20000480	array[6] of float
[0]	1.00001669	float
[1]	0.999963641	float
[2]	1.00001526	float
[3]	0.999966681	float
[4]	1.00000644	float
[5]	1.19317293	float
[4]	0x20000498	array[6] of float
[0]	2.69548845	float
[1]	-0.39500761	float
[2]	1.00001526	float
[3]	-0.711126804	float
[4]	-2.81127691	float
[5]	-0.836300373	float
[5]	0x200004B0	array[6] of float
[0]	1.00001669	float
[1]	-2.74660015	float
[2]	2.23535013	float
[3]	-1.50194931	float

如图为第 4 组解 qi[3]

3. 将 8 组关节角依次代入 正解 fkine 求得每组 关节角对应变化矩阵 tt，观察看出 第四列前三个数据即笛卡尔空间坐标 xyz 基本相同，再将关节角组合输入到 matlab，通过

robot.plot(qi)方法观察姿态，可以观察到输入关节角组合 与 各逆解组合 xyz 轴和位置一致。注意使用打开两个 matlab 程序观察，用一个时使用 plot 方法所有图窗界面会保持一致。

```
0x200001F8: 0.922001        0.327066        -0.758999       -0.216727
0x20000208: -0.36312        0.922797        0.128337        -0.0616043
0x20000218: 0.742436        0.203373        0.638249        -0.201597
0x20000228: 0               0               0               1
0x20000238: 0.263302        0.551929        -0.759014       -0.216723
0x20000248: -0.751762       0.646737        0.128358        -0.0616109
0x20000258: 0.561741        0.526305        0.638247        -0.201597
0x20000268: 0               0               0               1
0x20000278: 0.369939        0.492573        -0.758996       -0.216729
0x20000288: -0.643891       0.754191        0.128351        -0.0616081
0x20000298: 0.635698        0.434107        0.638249        -0.201591
0x200002A8: 0               0               0               1
0x200002B8: 0.606803        0.363336        -0.759007       -0.216732
0x200002C8: -0.422955       0.896977        0.12835         -0.0616064
0x200002D8: 0.727476        0.251699        0.638257        -0.201589
0x200002E8: 0               0               0               1
0x200002F8: 0.475303        0.327066        -0.758998       -0.216727
0x20000308: -0.36312        0.922797        0.128337        -0.0616043
0x20000318: 0.742436        0.203373        0.638249        -0.201597
0x20000328: 0               0               0               1
0x20000338: -0.382188       0.551929        -0.759014       -0.216723
0x20000348: -0.751762       0.646736        0.128358        -0.0616109
0x20000358: 0.561741        0.526305        0.638247        -0.201597
0x20000368: 0               0               0               1
0x20000378: 0.105618        0.492573        -0.758996       -0.216729
0x20000388: -0.643891       0.754191        0.128351        -0.0616081
0x20000398: 0.635698        0.434107        0.638249        -0.201591
0x200003A8: 0               0               0               1
0x200003B8: 0.291967        0.363336        -0.759007       -0.216732
0x200003C8: -0.422955       0.896977        0.12835         -0.0616064
0x200003D8: 0.727475        0.251699        0.638257        -0.201589
0x200003E8: 0               0               0               1
0x200003F8: 0.489624        0.429034        -0.759008       -0.216733
0x20000408: -0.533494       0.83593         0.128355        -0.0616089
0x20000418: 0.689603        0.342097        0.638235        -0.201597
0x20000428: 0               0               0               1
```

图中依次为 qi[7]， qi[6], qi[5], qi[4], qi[3], qi[2] ,qi[1] ,qi[0] ,qz 对应的变换矩阵.

Matlab 仿真图：

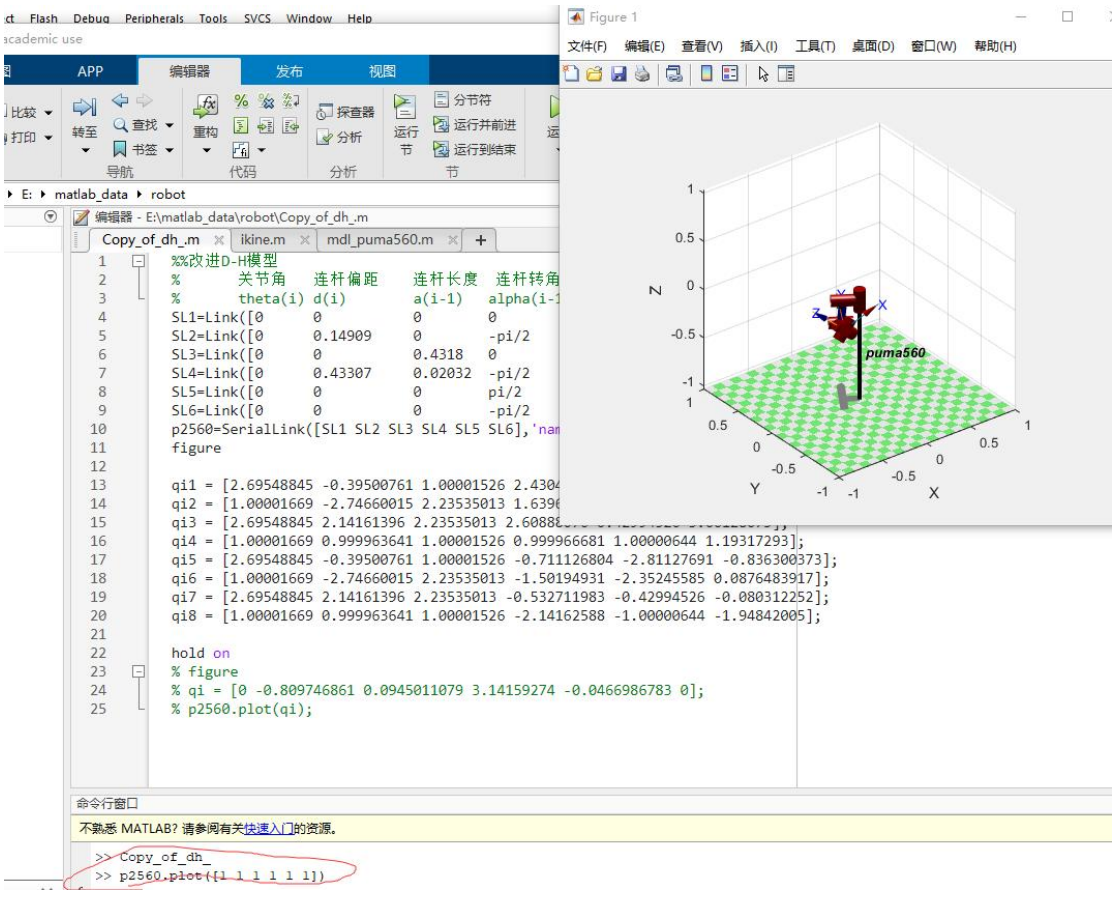


Figure 1
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
puma560
编辑器 - E:\matlab_data\robot\Copy_of_dh_.m
Copy_of_dh_.m
ikine.m
mdl_puma560.m
%%改进D-H模型
% 关节角 连杆偏距 连杆长度 连杆转角
% theta(i) d(i) a(i-1) alpha(i-1
SL1=Link([0 0 0 0
SL2=Link([0 0.14909 0 -pi/2
SL3=Link([0 0 0.4318 0
SL4=Link([0 0.43307 0.02032 -pi/2
SL5=Link([0 0 0 pi/2
SL6=Link([0 0 0 -pi/2
p2560=SerialLink([SL1 SL2 SL3 SL4 SL5 SL6],'nam
figure
qi1 = [2.69548845 -0.39500761 1.00001526 2.4304
qi2 = [1.00001669 -2.74660015 2.23535013 1.6396
qi3 = [2.69548845 2.14161396 2.23535013 2.60888
qi4 = [1.00001669 0.999963641 1.00001526 0.999966681 1.00000644 1.19317293];
qi5 = [2.69548845 -0.39500761 1.00001526 -0.711126804 -2.81127691 -0.836300373];
qi6 = [1.00001669 -2.74660015 2.23535013 -1.50194931 -2.35245585 0.0876483917];
qi7 = [2.69548845 2.14161396 2.23535013 -0.532711983 -0.42994526 -0.080312252];
qi8 = [1.00001669 0.999963641 1.00001526 -2.14162588 -1.00000644 -1.94842005];
hold on
% figure
% qi = [0 -0.809746861 0.0945011079 3.14159274 -0.0466986783 0];
% p2560.plot(qi);
命令行窗口
不熟悉 MATLAB? 请参阅有关快速入门的资源。
>> Copy_of_dh_
>> p2560.plot([1 1 1 1 1 1])

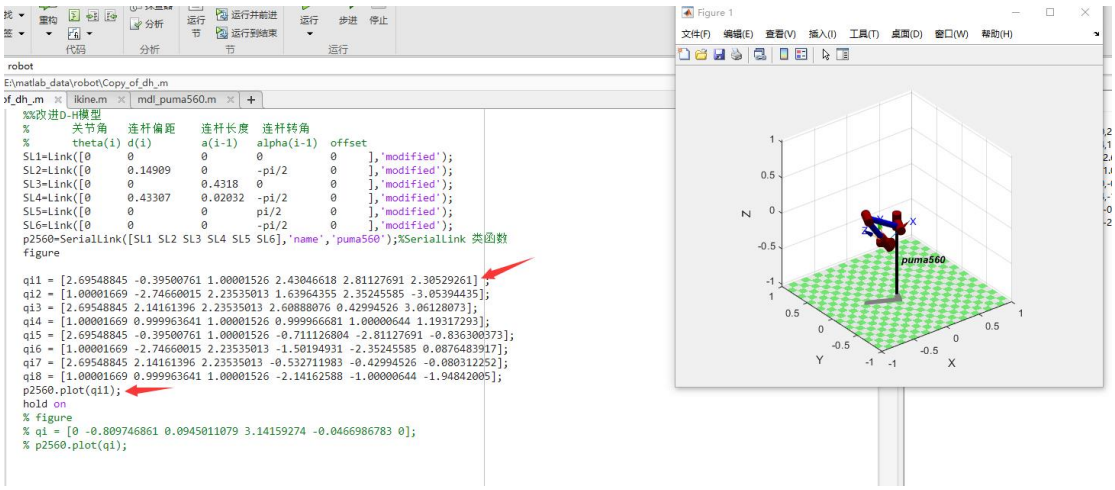


Figure 1
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
puma560
%%改进D-H模型
% 关节角 连杆偏距 连杆长度 连杆转角
% theta(i) d(i) a(i-1) alpha(i-1) offset
SL1=Link([0 0 0 0 0],'modified');
SL2=Link([0 0.14909 0 -pi/2 0],'modified');
SL3=Link([0 0 0.4318 0 0],'modified');
SL4=Link([0 0.43307 0.02032 -pi/2 0],'modified');
SL5=Link([0 0 0 pi/2 0],'modified');
SL6=Link([0 0 0 -pi/2 0],'modified');
p2560=SerialLink([SL1 SL2 SL3 SL4 SL5 SL6],'name','puma560');%SerialLink 类函数
figure
qi1 = [2.69548845 -0.39500761 1.00001526 2.43046618 2.81127691 2.30529261];
qi2 = [1.00001669 -2.74660015 2.23535013 1.63964355 2.35245585 -3.05394435];
qi3 = [2.69548845 2.14161396 2.23535013 2.60888076 0.42994526 3.06128073];
qi4 = [1.00001669 0.999963641 1.00001526 0.999966681 1.00000644 1.19317293];
qi5 = [2.69548845 -0.39500761 1.00001526 -0.711126804 -2.81127691 -0.836300373];
qi6 = [1.00001669 -2.74660015 2.23535013 -1.50194931 -2.35245585 0.0876483917];
qi7 = [2.69548845 2.14161396 2.23535013 -0.532711983 -0.42994526 -0.080312252];
qi8 = [1.00001669 0.999963641 1.00001526 -2.14162588 -1.00000644 -1.94842005];
p2560.plot(qi1);
hold on
% figure
% qi = [0 -0.809746861 0.0945011079 3.14159274 -0.0466986783 0];
% p2560.plot(qi);