

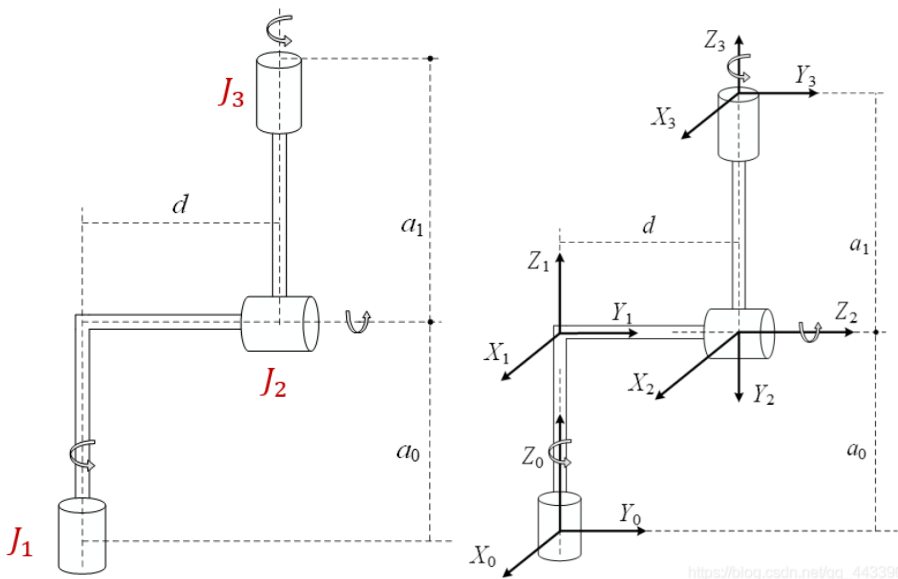
# 实验一：机器人运动学模型建立

## 1. 实验目的

本实验介绍如何使用 matlab 工具箱（RTB 工具箱 10.3）建立机器人模型，并对其进行绘制。

## 2. 实验内容

以如下三轴机械臂模型为例（三轴的明白了，其他的也会了），如下图所示该模型具有三个转动关节：



要求：利用机器人学中的相关知识，使用 DH 法建立连杆坐标系，使用机器人工具箱建立如上的机械臂模型。

D-H参数表

$i$	$a_{(i-1)}$	$\alpha_{(i-1)}$	$d_i$	$\theta_i$
1	0	0	$a_0$	$\theta_1(0)$
2	0	-90	$d$	$\theta_2(0)$
3	0	90	$a_1$	$\theta_3(0)$

[https://blog.csdn.net/qq\\_44339029](https://blog.csdn.net/qq_44339029)

### 3. 重要函数学习

#### 1、Link 类函数

Link 类函数，基于 DH 法建模，建立其相关关系，DH 法建模分改进型和标准型，Link 类函数的一种用法是  $R = \text{Link}([\theta, d, a, \alpha])$ ，其中参数  $\theta$  代表 DH 建模的关节角、参数  $d$  代表 DH 建模的连杆偏距、参数  $a$  代表 DH 建模的连杆长度、参数  $\alpha$  代表 DH 建模的连杆转角。例如：

$L(1) = \text{Link}([1, 2, 3, 4], 'modified')$ ，其中 `modified` 表示用改进型 DH 法建模：

(1) 我们在命令行窗口输入上面的例子

```
1 | L(1)=Link([1,2,3,4], 'modified')
```

(2) Link 类函数中包含一些属性，我们可以用像调用 c/c++ 语言里面的结构体成员类似的方法去调用它，如下所示：

R.RP: 可以用来获取连杆关节类型, 如下所示我们可以知道我们建立的 L(1) 是转动关节 (若为 P 则为移动关节)

```
1 | L(1).RP
```

R.theta: 可以用来获取连杆关节角, 如下所示我们可以知道我们建立的 L(1) 的关节角为 1

```
1 | L(1).theta
```

R.d: 可以用来获取连杆偏距, 如下所示我们可以知道我们建立的 L(1) 的连杆偏距为 2

```
1 | L(1).d
```

R.a: 可以用来获取连杆长度, 如下所示我们可以知道我们建立的 L(1) 的连杆长度为 3

```
1 | L(1).a
```

R.alpha: 可以用来获取连杆扭转角, 如下所示我们可以知道我们建立的 L(1) 的连杆扭转角为 4

```
1 | L(1).alpha
```

R.sigma: 也可以用来查询我们建立的是转动关节还是移动关节, 默认 0 是转动关节; 1 是移动关节, 如下所示我们可以知道我们建立的 L(1) 为转动关节

```
1 | L(1).sigma
```

R.mdh: 可以用来查询我们是按照标准 DH 法建立的还是按改进型 DH 法建

立的模型，默认 0 为标准 D.H 法，1 为改进 D-H 法，如下所示我们可以知道我们是按照改进型 DH 法建立的 L(1)

```
1 | L(1).mdh
```

## 2、Seriallink 类函数

Seriallink 类函数可以把我们使用 Link 函数建立的连杆连成一个整体，生成一个串联机械臂，比如下面这个例子，我们已经使用 Link 函数建立好了 L(1)~L(6)这六个关节模型，然后我们只需要使用：

```
Six_Link=SerialLink([L(1),L(2),L(3),L(4),L(5),L(6)]);
```

就可以将其连成一个整体生成一个六周的串联机械臂，并取名为 Six\_Link（名字可以任意取）

```
1 | L(1)=Link([0,0,0,0],'modified');
2 | L(2)=Link([1.25,pi/2,0,pi/2],'modified');
3 | L(3)=Link([10.5,0,0,pi/2],'modified');
4 | L(4)=Link([0,pi/2,13,-pi],'modified');
5 | L(5)=Link([0,pi/2,0,-pi],'modified');
6 | L(6)=Link([0,pi/2,10,-pi],'modified');
7 | Six_Link=SerialLink ([L(1),L(2),L(3),L(4),L(5),L(6)]);
```

在命令行窗口输入 help Seriallink 会发现 Seriallink 类函数有很多可以设置的参数，在这里只针对一小部分进行介绍

①只读参数：关节自由度 n、机械臂配置字符串 config、DH 约定形式布尔值 mdh、DH 参数 theta、d、a、alpha

```
1 | Properties (read only)::
2 |
3 |     n          number of joints
4 |     config     joint configuration string, eg. 'RRRRRR'
5 |     mdh        kinematic convention boolean (0=DH, 1=MDH)
6 |     theta      kinematic: joint angles (1xN)
7 |     d          kinematic: link offsets (1xN)
8 |     a          kinematic: link lengths (1xN)
9 |     alpha      kinematic: link twists (1xN)
```

他们的使用方法很简单，比如我们已经按上面建立好了一个机械臂模型 Six\_Link，要查看当前关节的自由度，只需要使用 Six\_Link.n 就可以了，如下所示：

```
>> Six_Link.n
```

```
ans =
```

```
6
```

要查看当前各个关节的类型，只需要使用 `Six_Link.config` 就可以了，如下所示，可以知道我们建立的六轴机械臂各个关节均为转动关节：

```
>> Six_Link.config
```

```
ans =
```

```
'RRRRRR'
```

②可读可写参数：连杆对象的矢量 `links` 、重力方向 `gravity` 、机器人基座的姿态 `base` 、 机器人工具坐标系转换 `tool` 、关节极限 `qlim` 、运动关节坐标偏移 `offset`、机器人的名称（用于图形显示）`name` 、注释 `manuf` 和 `comment`、`plot()`方法的选项(单元格数组)`plotopt` 、使用 MEX 版本的 RNE（只有当 `mex` 文件存在时才能设置为 `true`。默认为 `true`）`fast`

```
1 Properties (read/write)::
2
3   links      vector of Link objects (1xN)
4   gravity    direction of gravity [gx gy gz]
5   base       pose of robot's base (4x4 homog xform)
6   tool       robot's tool transform, T6 to tool tip (4x4 homog xform)
7   qlim       joint limits, [qmin qmax] (Nx2)
8   offset     kinematic joint coordinate offsets (Nx1)
9   name       name of robot, used for graphical display
10  manuf      annotation, manufacturer's name
11  comment    annotation, general comment
12  plotopt    options for plot() method (cell array)
13  fast       use MEX version of RNE. Can only be set true if the mex
14            file exists. Default is true.
```

比如要查看当前各个关节连杆情况，只需要使用 `Six_Link.links` 就可以了，如下所示：

```
>> Six_Link.links

ans =
theta=q1, d=      0, a=      0, alpha=      0, offset=      0 (R, modDH)
theta=q2, d=    1.571, a=      0, alpha=    1.571, offset=      0 (R, modDH)
theta=q3, d=      0, a=      0, alpha=    1.571, offset=      0 (R, modDH)
theta=q4, d=    1.571, a=    13, alpha=   -3.142, offset=      0 (R, modDH)
theta=q5, d=    1.571, a=      0, alpha=   -3.142, offset=      0 (R, modDH)
theta=q6, d=    1.571, a=    10, alpha=   -3.142, offset=      0 (R, modDH)
```

比如要查看当前机器人基座的姿态，只需要使用 `Six_Link.base` 就可以了，如下所示：

```
>> Six_Link.base

ans =

     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1
```

③Seriallink 类函数中的一些方法或者函数，本部分不做详细介绍，有兴趣的可自行研究，比如显示机器人的图形表示的 `plot`、显示机器人的三维图形模型 `plot3d`（只能用于标准 DH 建模）、`teach`、关节空间轨迹 `jtraj` 等

```

1 SerialLink Serial-link robot class
2
3 A concrete class that represents a serial-link arm-type robot. The
4 mechanism is described using Denavit-Hartenberg parameters, one set
5 per joint.
6
7 Methods::
8
9 plot          display graphical representation of robot
10 plot3d       display 3D graphical model of robot
11 teach        drive the graphical robot
12 getpos       get position of graphical robot
13 -
14 jtraj        a joint space trajectory
15 -
16 edit         display and edit kinematic and dynamic parameters
17 -
18 isspherical  test if robot has spherical wrist
19 islimit      test if robot at joint limit
20 isconfig     test robot joint configuration
21 -
22 fkine        forward kinematics
23 A           link transforms
24 trchain      forward kinematics as a chain of elementary transforms
25 -
26 ikine6s      inverse kinematics for 6-axis spherical wrist revolute robot
27 ikine        inverse kinematics using iterative numerical method
28 ikunc        inverse kinematics using optimisation
29 ikcon        inverse kinematics using optimisation with joint limits
30 ikine_sym    analytic inverse kinematics obtained symbolically
31 -
32 jacob0       Jacobian matrix in world frame
33 jacobn       Jacobian matrix in tool frame
34 jacob_dot    Jacobian derivative
35 maniplty     manipulability
36 vellipse     display velocity ellipsoid
37 fellipse     display force ellipsoid
38 qmincon      null space motion to centre joints between limits
39 -
40 accel        joint acceleration
41 coriolis     Coriolis joint force
42 dyn          show dynamic properties of links
43 friction     friction force
44 gravload     gravity joint force
45 inertia      joint inertia matrix
46 cinertia     Cartesian inertia matrix
47 nofriction    set friction parameters to zero
48 rne          inverse dynamics
49 fdyn         forward dynamics
50 -
51 payload      add a payload in end-effector frame
52 perturb      randomly perturb link dynamic parameters
53 gravjac      gravity load and Jacobian
54 paycap       payload capacity
55 pay          payload effect
56 -
57 sym          a symbolic version of the object
58 gencoords    symbolic generalized coordinates

```