Group number: 2

Group 2 Report

	Student number	Student name	contribution	signature
1	18222007	Fan Junming	45%	Fan Junming
2	18222003	Chen Xuan’ang	35%	Chen Xuan’ang
3	18222006	Du Tianhao	10%	Du Tianhao
4	18222009	Hou Jun	10%	Hou Jun
5	18222005	Ding Haochen	0	Ding Haochen

Catlog

1.Robot system design

We chose cooperative robot GCR20-1100 from SIASUN robotic company to do build our robotic system.

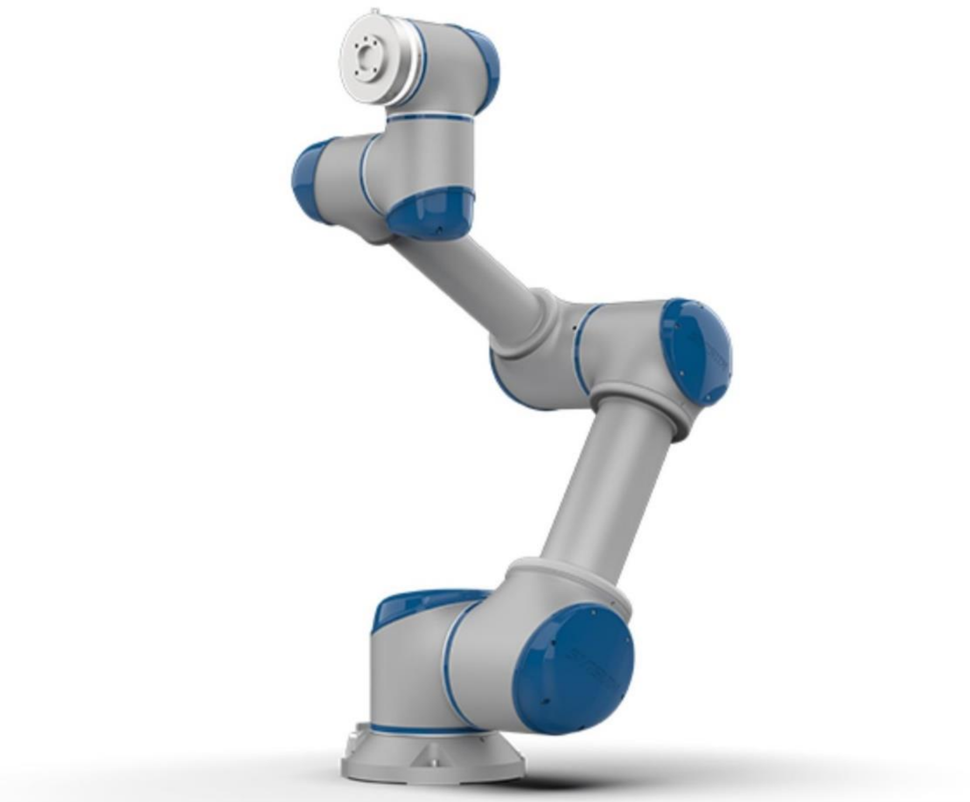


Fig 1-1 GCR20-1100 cooperative robot

And the technical parameters are shown in the following figure .

Basic parameters

Max Payload	20KG(22KG)
Degree of Freedom	6
Rotation	±360°
Repeatability	±0.05mm
Reach	1100mm
Max Straight-line Speed	1.0m/s
Max TCP Speed	1.2m/s
Tool Interface	GB/T 14468.1-50-4-M6 (EQV ISO 9409-1)
Power Supply	230VAC (-15%~+10%), 50-60HZ
Power Consumption	Typical power consumption600W
Installation	Installed in any direction
Robot Dimensions	1320mm X 420mm X 290 mm
Control Box Size	540mm X 520mm X 220 mm
Ambient Temperature	-10℃ - 45℃
Storage Temperature	-40℃-55℃
IPClass	IP54
Net Weight	55KG

Fig 1-2 Basic parameters of GCR20-1100

a) The DOF of this robot: six;

b) The number of joints: six. Type of each joint: all of the are rotating vice.

c) The base position of the robot is [0.4,0,0] (in meters);

d) The length of each link is shown on the following figure,

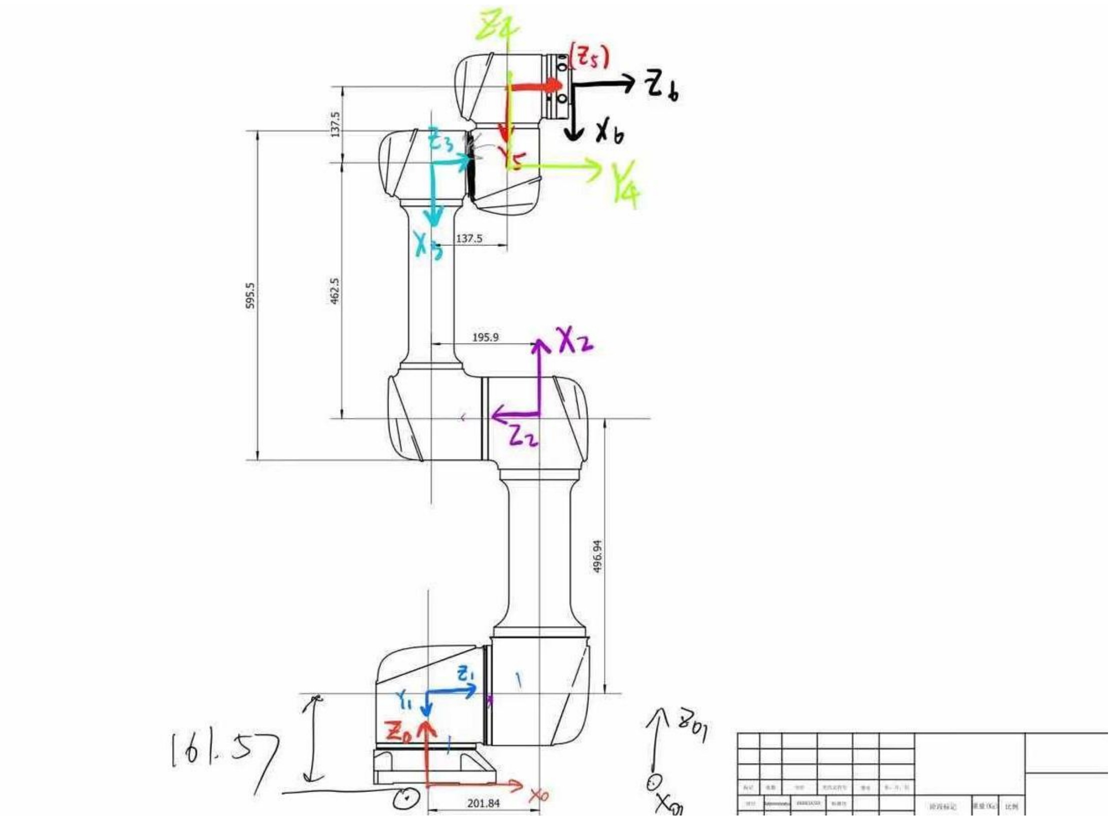


Fig 1-3 length of each link

e) Horizontal reach: from z=0 to z=1258.51mm.

Vertical reach: from y=-1258.51mm to y=1258.51mm.

f) Joint space work envelope:-2pi~+2pi

2.Tool location

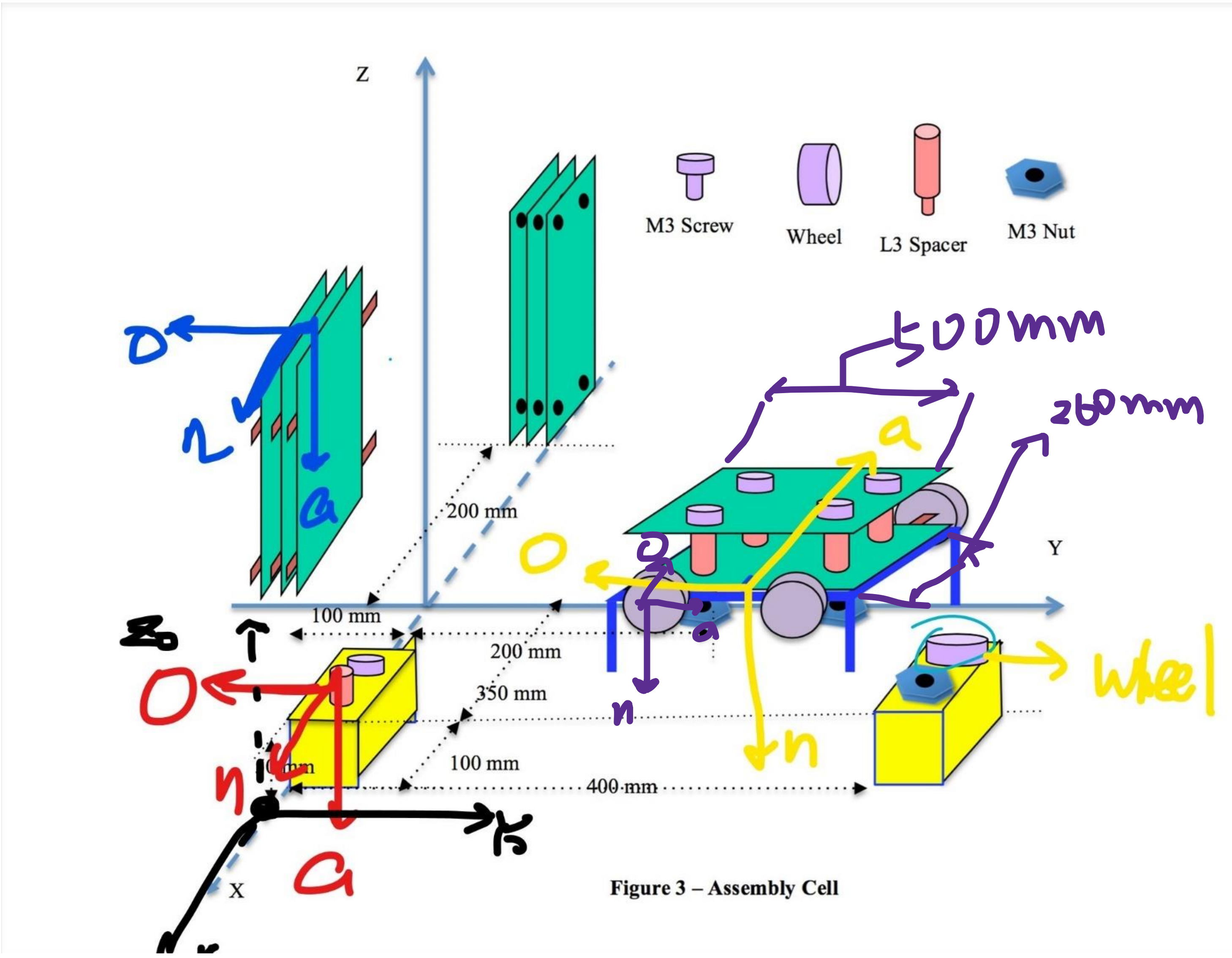


Fig 2-1 n-o-a coordinates

The n-o-a coordinate vector is shown on the figure, from which we could obtain:

1)The initial and the final configuration frame of the lower panel are given as follows,

$$T_{initial}=\begin{bmatrix}1&0&0&0.45\\0&-1&0&-0.1\\0&0&-1&0.4\\0&0&0&1\end{bmatrix},T_{final}=\begin{bmatrix}0&0&1&-0.37\\0&-1&0&0.45\\-1&0&0&0.05\\0&0&0&1\end{bmatrix}$$

2)The initial and the final configuration frame of the wheel are given as follows,

$$T_{initial}=\begin{bmatrix}1&0&0&0.45\\0&-1&0&0.4\\0&0&-1&0\\0&0&0&1\end{bmatrix},T_{final}=\begin{bmatrix}0&0&0&-0.37\\0&-1&0&0.2\\-1&0&-1&0.05\\0&0&0&1\end{bmatrix}$$

3.Kinematic model

a) The coordinate frames attached to the links of the robotic system. We labeled these frames with 0 to 6 in fig3-1.

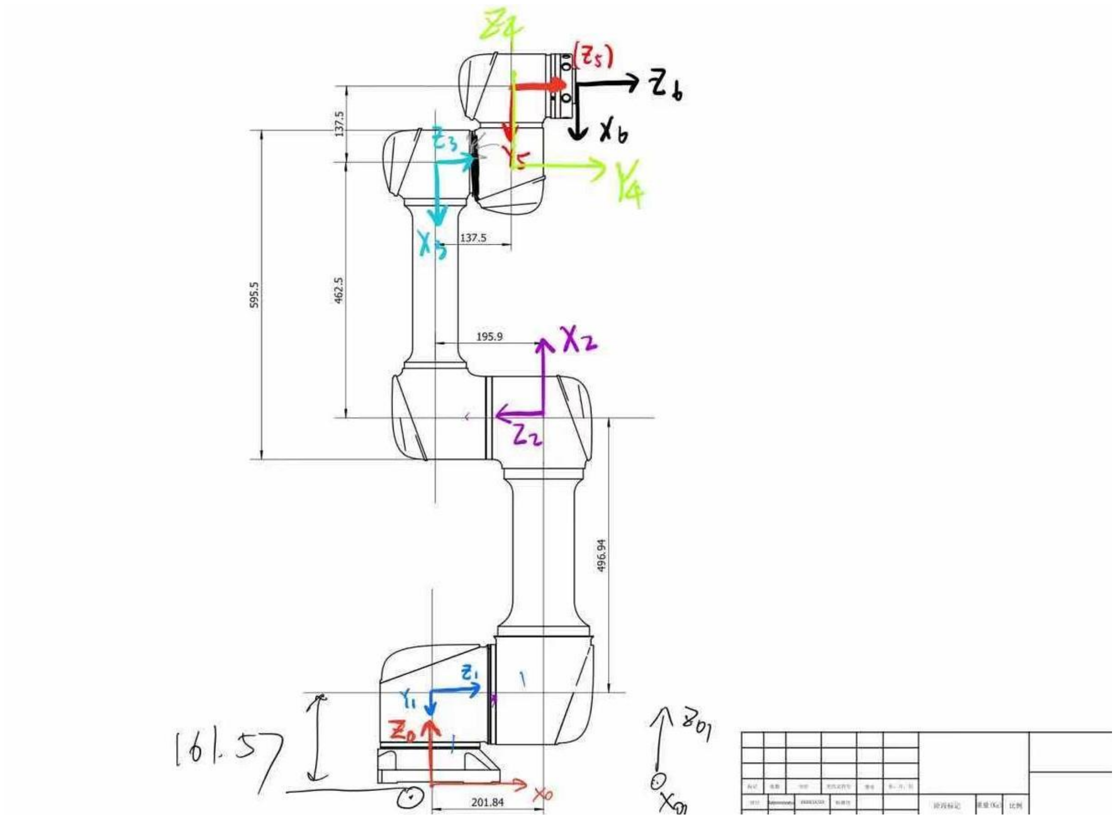


Fig 3-1 coordinate frames attached to the links

b) The D-H table is shown below:

J_i	α/rad	a/m	d/m	θ/rad	Jv
1	π/2	0	0.1616	θ_1	θ_1
2	π	0.4969	0.2018	θ_2	θ_2
3	π	-0.4625	0.1959	θ_3	θ_3
4	π/2	0	0.1375	θ_4	θ_4
5	−π/2	0	0.1375	θ_5	θ_5
6	0	0	0.1157	θ_6	θ_6

c) The arm equations relating the coordinate frames of the robot tool to the coordinate frame of its base:

By the transform matrix:

$$A_i^{i-1} = \begin{bmatrix} cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & \alpha_i cos\theta_i \\ sin\theta_i & \cos\alpha_i \cos\theta_i & -sin\alpha_i cos\theta_i & \alpha_i \sin\theta_i \\ 0 & sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

We could obtain:

$$A_1^0 = \begin{bmatrix} cos\theta_1 & 0 & -\sin\theta_1 & 0 \\ sin\theta_1 & 0 & cos\theta_1 & 0 \\ 0 & -1 & 0 & 0.16157 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2^1 = \begin{bmatrix} cos\theta_2 & sin\theta_2 & 0 & 0.4969cos\theta_2 \\ sin\theta_2 & -cos\theta_2 & 0 & 0.4969sin\theta_2 \\ 0 & 0 & -1 & 0.20184 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3^2 = \begin{bmatrix} cos\theta_3 & sin\theta_3 & 0 & -0.4625cos\theta_3 \\ sin\theta_3 & -cos\theta_3 & 0 & -0.4625sin\theta_3 \\ 0 & 0 & -1 & 0.1959 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_4^3 = \begin{bmatrix} cos\theta_4 & 0 & sin\theta_4 & 0 \\ sin\theta_4 & 0 & -cos\theta_4 & 0 \\ 0 & 1 & 0 & 0.1375 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_5^4 = \begin{bmatrix} cos\theta_5 & 0 & -sin\theta_5 & 0 \\ sin\theta_5 & 0 & cos\theta_5 & 0 \\ 0 & -1 & 0 & 0.1375 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

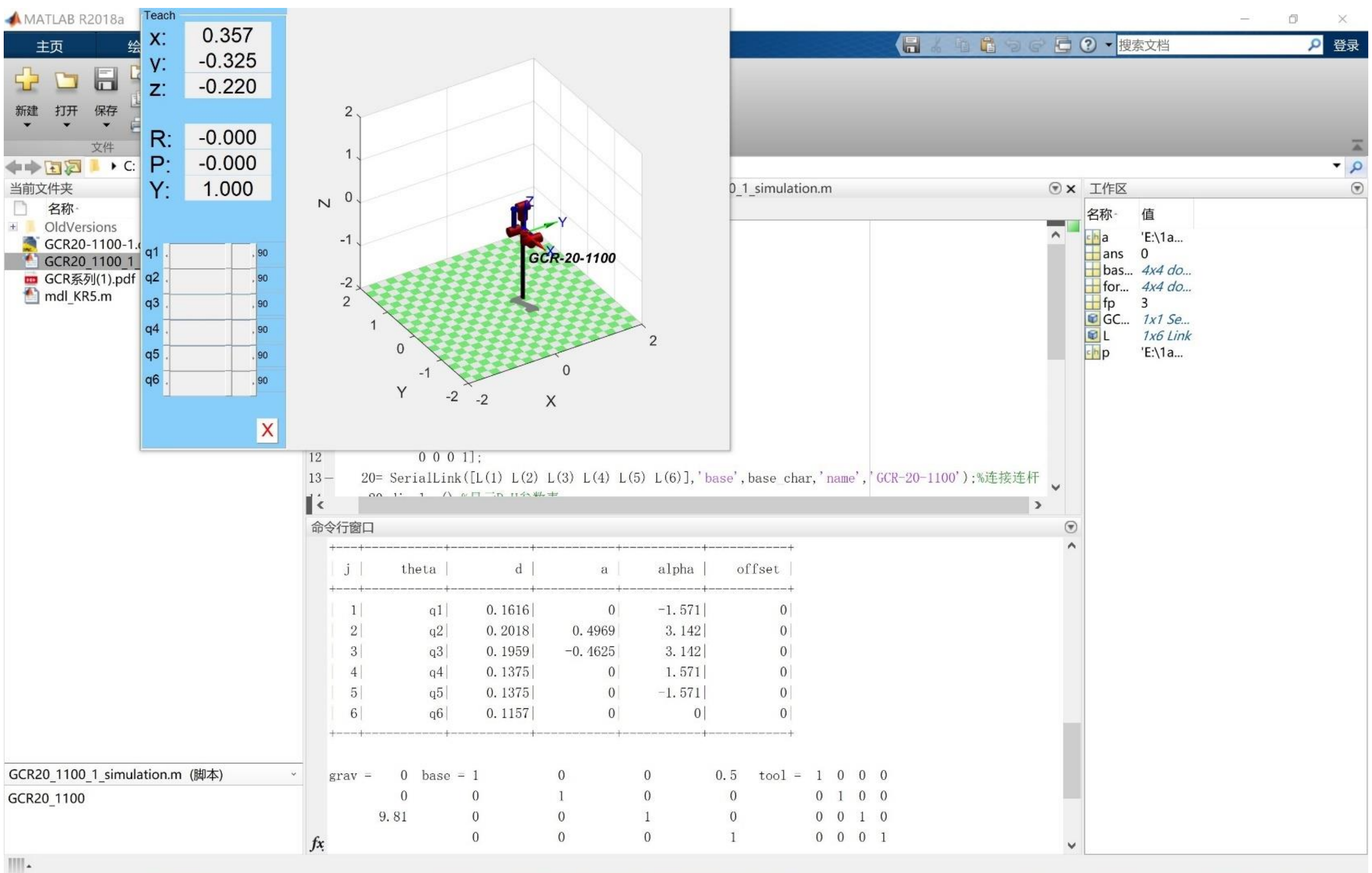
$$A_6^5 = \begin{bmatrix} cos\theta_6 & -sin\theta_6 & 0 & 0 \\ sin\theta_6 & cos\theta_6 & 0 & 0 \\ 0 & 0 & 1 & 0.1157 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

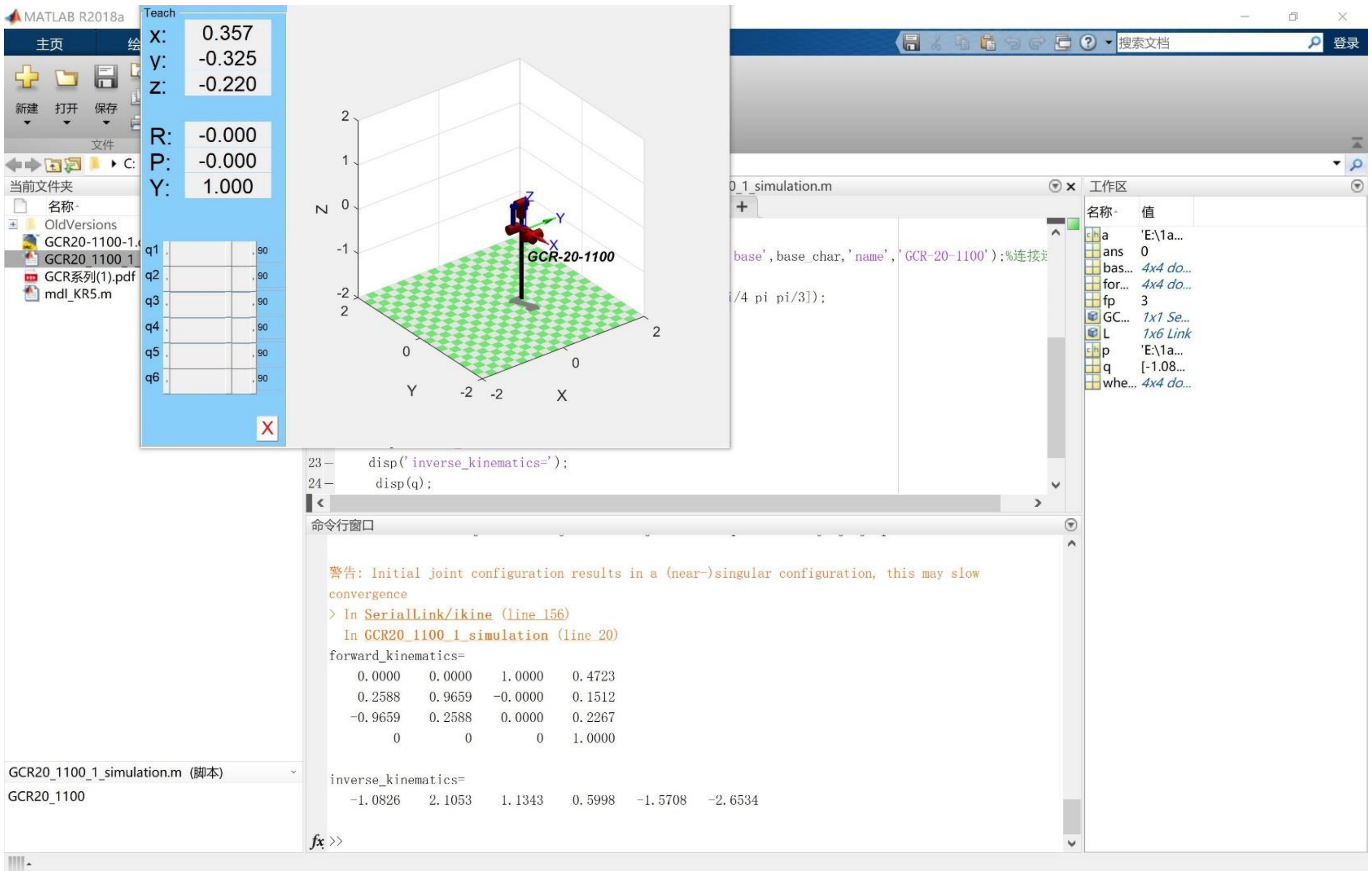
Multiple them together we find

$$T_4^0 = \begin{bmatrix} c_1(c_{234}c_5c_6 - s_{234}s_6) - s_1s_5c_6 & c_1(-c_{234}c_5c_6 - s_{234}s_6) + s_1s_5c_6 & c_1c_{234}s_5 + s_1c_5 & c_1(-0.4625c_{23} + 0.4969c_2) \\ s_1(c_{234}c_5c_6 - s_{234}s_6) + c_1s_5c_6 & s_1(-c_{234}c_5c_6 - s_{234}c_6) - c_1s_5c_6 & s_1c_{234}s_5 - c_1c_5 & s_1(-0.4625c_{23} + 0.4969c_2 \\ s_{234}c_5c_6 + c_{234}s_6 & -s_{234}c_5c_6 + c_{234}s_6 & s_{234}s_5 & 0.4625s_{23} + 0.4969s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

4.Matlab Results

a) Screenshot of the Matlab results:





b) Matlab codes:

```
% GCR20_1100
clear L
L(1)=Link('d',0.16157,'a',0,'alpha',-pi/2);
L(2)=Link('d',0.20184,'a',0.49694,'alpha',pi);
L(3)=Link('d',0.1959,'a',-0.4625,'alpha',pi);
L(4)=Link('d',0.1375,'a',0,'alpha',pi/2);
L(5)=Link('d',0.1375,'a',0,'alpha',-pi/2);
L(6)=Link('d',0.1157,'a',0,'alpha',0);
base_char=[ 1 0 0 0.5;
            0 1 0 0;
            0 0 1 0;
            0 0 0 1];
GCR_20= SerialLink([L(1) L(2) L(3) L(4) L(5)
L(6)],'base',base_char,'name','GCR-20-1100');%joints and links
 GCR_20.display();%show the DH table
 forward_kinematics=GCR_20.fkine([pi/2 -pi/3 pi/6 pi/4 pi pi/3]);
wheel_initial=[ 1 0 0 0.45;
                0 -1 0 0.4;
                0 0 -1 0;
                0 0 0 1];
q=GCR_20.ikine(wheel_initial);
disp('forward_kinematics=');
 disp(forward_kinematics);
disp('inverse_kinematics=');
disp(q);
GCR_20.plot([pi/2 pi/2 pi/2 pi/2 pi/2 pi/2]);
GCR_20.teach();
```

5.Jacobian matrix

```
clear L
%           theta    d          a       alpha
L(1) = Link([0       0.4        0.18    pi/2]);
L(2) = Link([0       0.135      0.60    pi]);
L(3) = Link([0       0.135      0.12   -pi/2]);
L(4) = Link([0       0.62       0       pi/2]);
L(5) = Link([0       0          0      -pi/2]);
L(6) = Link([0       0          0       0]);
KR5=SerialLink(L, 'name', 'Kuka KR5');
KR5.tool=transl(0,0,0.05);%The fingers of the robot gripper is 50mm
KR5.ikineType = 'kr5';
KR5.model3d = 'KUKA/KR5_arc';


T0=[1 0 0 0.5;
   0 1 0 0.5;
   0 0 1 0.5;
   0 0 0 1;];
IK0=KR5.ikine6s(T0);
disp(IK0);
FK0=KR5.fkine(IK0);
disp(FK0);
%Suppose the base point is at(0,0,0)
%Suppose the robot is at this position due to T0 at first.Each of the
coodinate of gripper and base is on the same direction.
%The position of the gripper is (0.5 0.5 0.5)
```

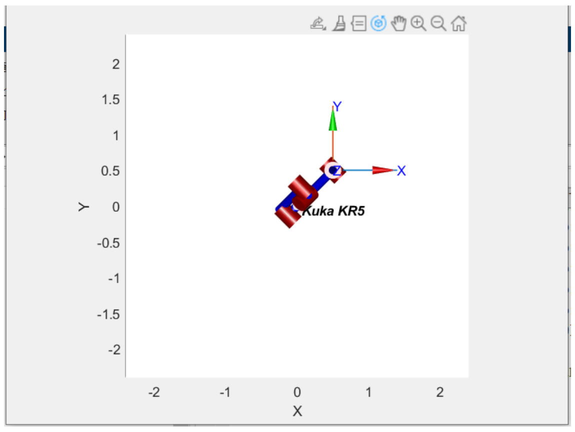
The code above gives the forward and inverse kinematics of KR5,the initial position of the gripper is at (0.0,0.5,0.5),call it A.

```
T1=transl(-0.125,-0.5,-0.45)*troty(pi) ;%First the screw is at (0.375
0 0.05)
disp(T1);
IK1=KR5.ikine6s(T1);%KR5 inverse kinematics
disp(IK1);
FK1=KR5.fkine(IK1);
disp(FK1);
T2=transl(-0.375,0.2,-0.025);%The screw is moved to (0 0.2 0.025) from
(0.375 0 0.05) and the orientation is not changed.
IK2=KR5.ikine6s(T2);
disp(IK2);
FK2=KR5.fkine(IK2);
disp(FK2);
J1=KR5.jacob0(IK1);
J2=KR5.jacob0(IK2);
disp(J1);
disp(J2);
```

```
KR5.plot(IK0);
KR5.plot(IK1);
KR5.plot(IK2);
```

There are two significant point, one is (0.375,0,0.05) where the screw put, the other is(0,0.2,0.025)where the screw needed to be put, B and C respectively.

From the orientation and position of the TCP, we find the homogenous transformation equation from A to B and B to C, do the inverse kinematics and we find the pose in joint space. We use the function "jacob0()" in robotics toolbox to find the two jacobian matrixes.



This picture shows the initial pose of the robot.

Here is the two Jacobian matrixes from A to B and from B to C.

J1=	0.5000	0.2062	-0.1623	0.0183	0.0121	0
	-0.1250	0.8246	-0.6494	-0.0046	0.0485	0
	0.0000	-0.6954	0.1232	0.0000	0.0000	0
	-0.0000	0.9701	-0.9701	-0.0917	0.9701	0.0000
	-0.0000	-0.2425	0.2425	-0.3667	-0.2425	0.0000
	1.0000	0.0000	0.0000	-0.9258	0.0000	-1.0000

J2=	-0.2000	0.3750	-0.5126	0.0054	0.0441	0
	-0.3750	-0.2000	0.2734	0.0101	-0.0235	0
	0.0000	-0.6050	0.0256	-0.0000	-0.0000	0
	-0.0000	-0.4706	0.4706	-0.2027	0.4706	0.0000
	0.0000	-0.8824	0.8824	0.1081	0.8824	0.0000
	1.0000	0.0000	0.0000	-0.9733	-0.0000	1.0000

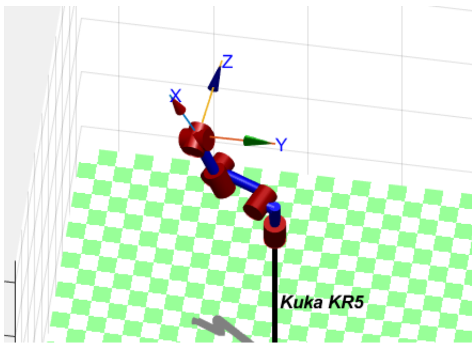
6.Smooth transition for the tooltip of the robot

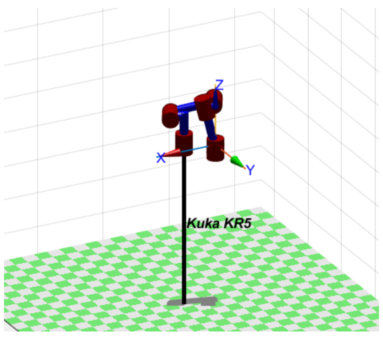
```
T0=[1  0  0  0.5;
    0  1  0  0.5;
    0  0  1  0.5;
    0  0  0  1;];
IK0=KR5.ikine6s(T0,'lun');
disp(IK0);
FK0=KR5.fkine(IK0);
```

```
disp(FK0);
%Suppose the base point is at(0,0,0)
%Suppose the robot is at this position due to T0 at first.Each of the
coodinate of gripper and base is on the same direction.
%The position of the gripper is (0.5 0.5 0.5)
T1=transl(-0.125,-0.5,-0.45)*troty(pi) ;%First the screw is at (0.375
0 0.05)
disp(T1);
IK1=KR5.ikine6s(T1,'lun');%KR5 inverse kinematics
disp(IK1);
FK1=KR5.fkine(IK1);
disp(FK1);
T2=transl(-0.375,0.2,-0.025);%The screw is moved to (0 0.2 0.025) from
(0.375 0 0.05) and the orientation is not changed.
IK2=KR5.ikine6s(T2,'lun');
disp(IK2);
FK2=KR5.fkine(IK2);
disp(FK2);

t=[0:0.05:2]';
q1=jtraj(IK0,IK1,t);
KR5.plot(q1);
q2=jtraj(IK1,IK2,t);
KR5.plot(q2);
```

Use the function plot and jtraj to build the animations of trajectory planning. Here are some of the screen shot of the animations. Run the code to find the full animations.





We made the smooth transition in joint space with inverse kinematics of the point in Cartesian space. Both q1 and q2 are 41X6 matrixes, means we choose 41 via points.

7.The velocity of the robot joint

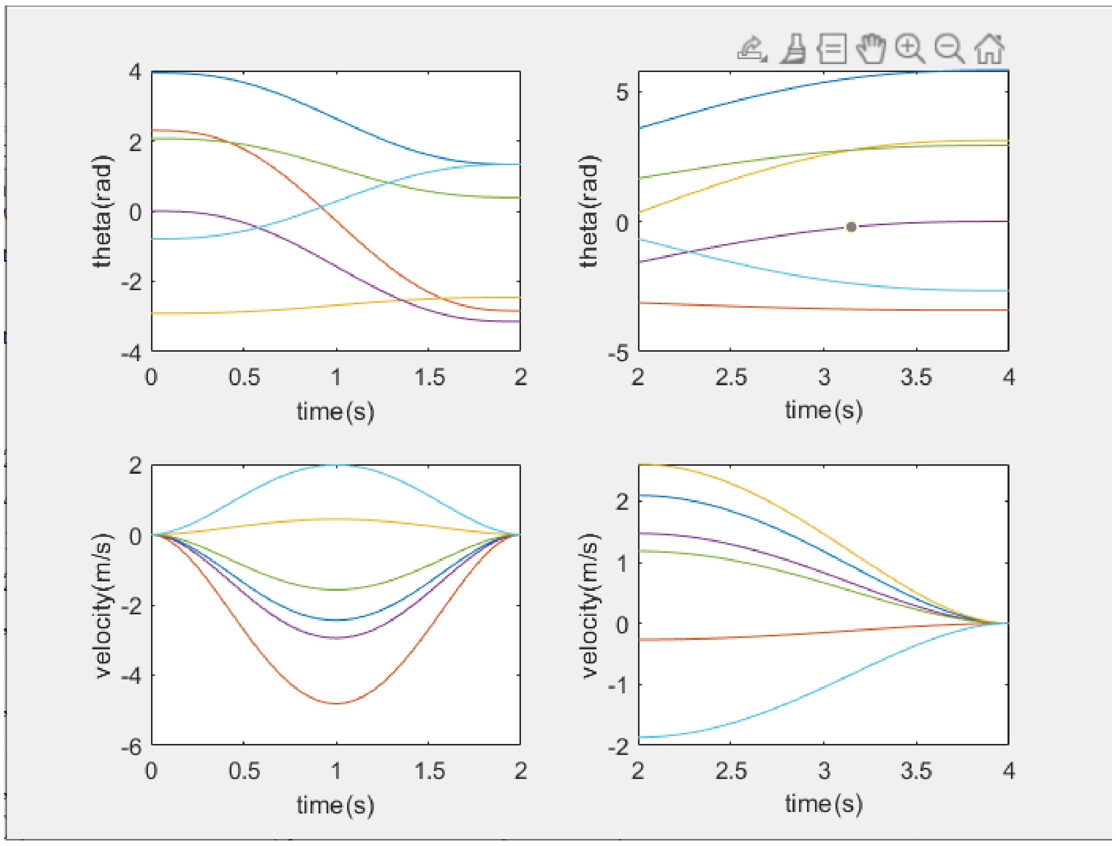
```
clear L
%           theta    d          a       alpha
L(1) = Link([0       0.4        0.18    pi/2]);
L(2) = Link([0       0.135       0.60    pi]);
L(3) = Link([0       0.135       0.12   -pi/2]);
L(4) = Link([0       0.62        0       pi/2]);
L(5) = Link([0       0          0      -pi/2]);
```

```
L(6) = Link([0       0          0      0]);
KR5=SerialLink(L, 'name', 'Kuka KR5');
KR5.tool=transl(0,0,0.05);%The fingers of the robot gripper is 50mm
KR5.ikineType = 'kr5';
KR5.model3d = 'KUKA/KR5_arc';


T0=[1 0 0 0.5;
   0 1 0 0.5;
   0 0 1 0.5;
   0 0 0 1;];
IK0=KR5.ikine6s(T0,'lun');
disp(IK0);
FK0=KR5.fkine(IK0);
disp(FK0);
%Suppose the base point is at(0,0,0)
%Suppose the robot is at this position due to T0 at first.Each of the
coodinate of gripper and base is on the same direction.
%The position of the gripper is (0.5 0.5 0.5)
T1=transl(-0.125,-0.5,-0.45)*troty(pi) ;%First the screw is at (0.375
0 0.05)
disp(T1);
IK1=KR5.ikine6s(T1,'lun');%KR5 inverse kinematics
disp(IK1);
FK1=KR5.fkine(IK1);
disp(FK1);
T2=transl(-0.375,0.2,-0.025);%The screw is moved to (0 0.2 0.025) from
(0.375 0 0.05) and the orientation is not changed.
IK2=KR5.ikine6s(T2,'lun');
disp(IK2);
FK2=KR5.fkine(IK2);
disp(FK2);

t1=[0:0.05:2]';
t2=[2:0.05:4]';
[q1,qd1,qdd1]=jtraj(IK0,IK1,t1);
[q2,qd2,qdd2]=jtraj(IK1,IK2,t2);
subplot(2,2,1)
plot(t1,q1);xlabel('time(s)'),ylabel('theta(rad)');
subplot(2,2,2)
plot(t2,q2);xlabel('time(s)'),ylabel('theta(rad)');
subplot(2,2,3)
plot(t1,qd1);xlabel('time(s)'),ylabel('velocity(m/s)');
subplot(2,2,4)
plot(t2,qd2);xlabel('time(s)'),ylabel('velocity(m/s)');
```

q means the trajectory plan, qd means the velocity, qdd means the acceleration. Here are four pictures, first two pictures show the trajectory planning using smooth interpolation. Other two pictures show the velocity.



8.Different configurations

'l' ***arm to the left (default)***

'r' ***arm to the right***

'u' ***elbow up (default)***

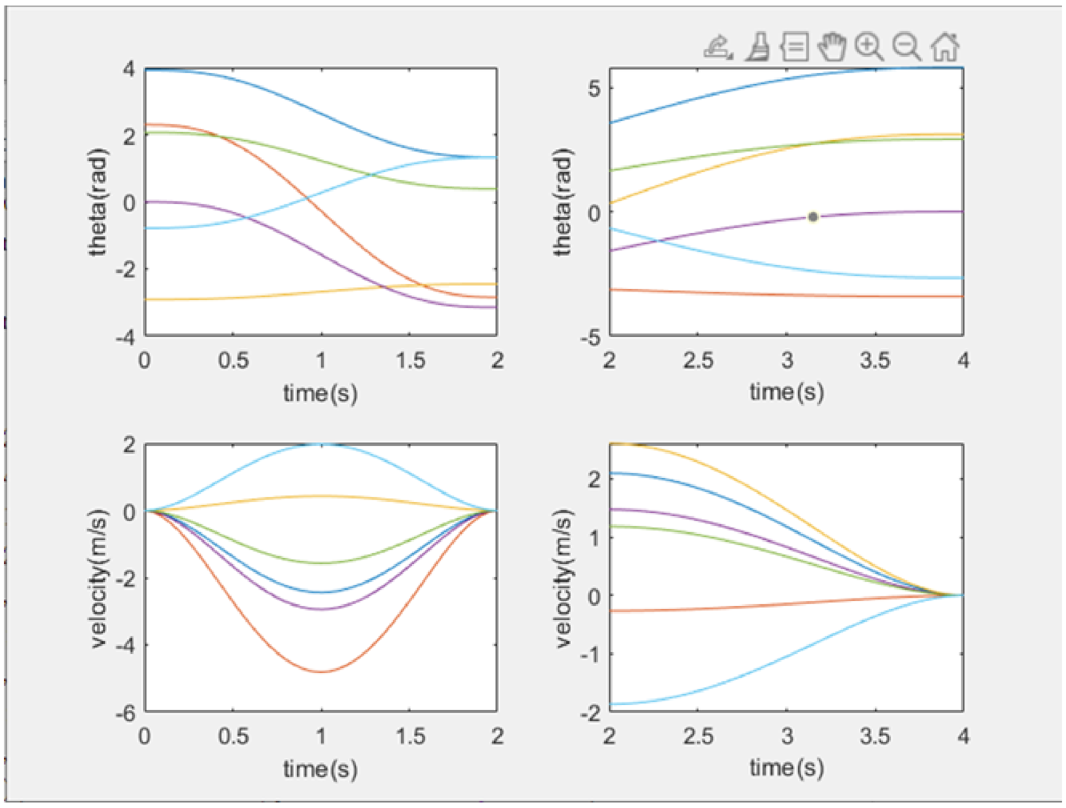
'd' ***elbow down***

'n' ***wrist not flipped (default)***

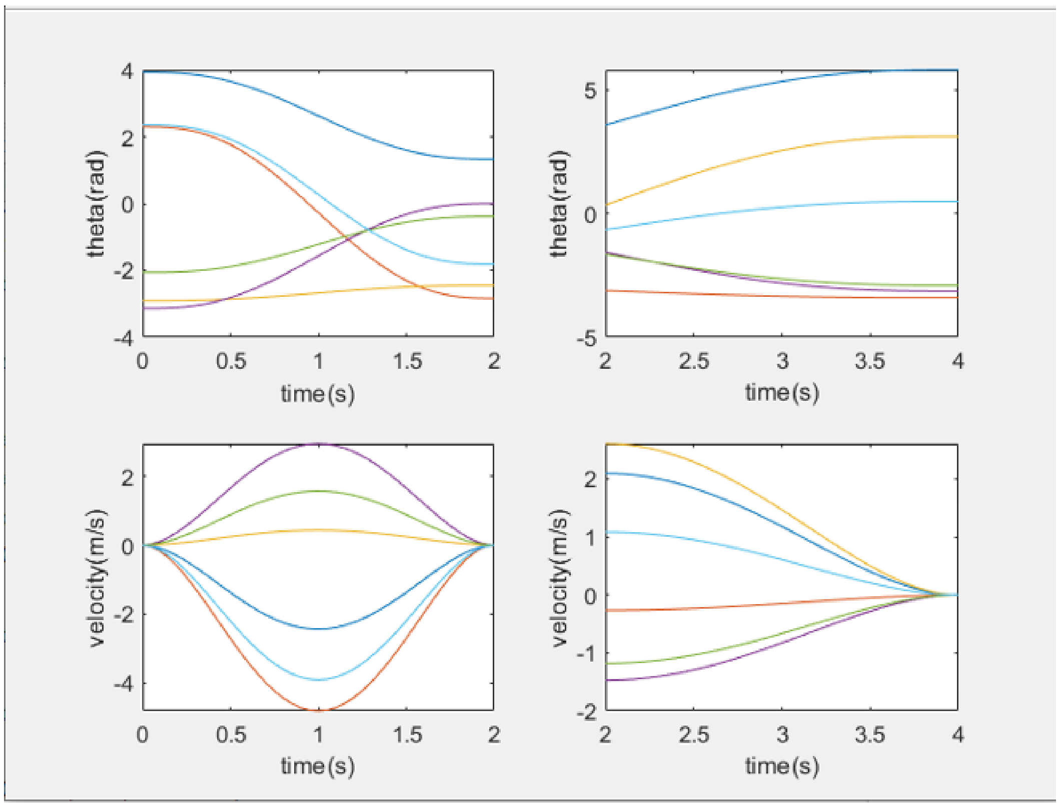
'f' ***wrist flipped (rotated by 180 deg)***

There are 8 different configurations in total.

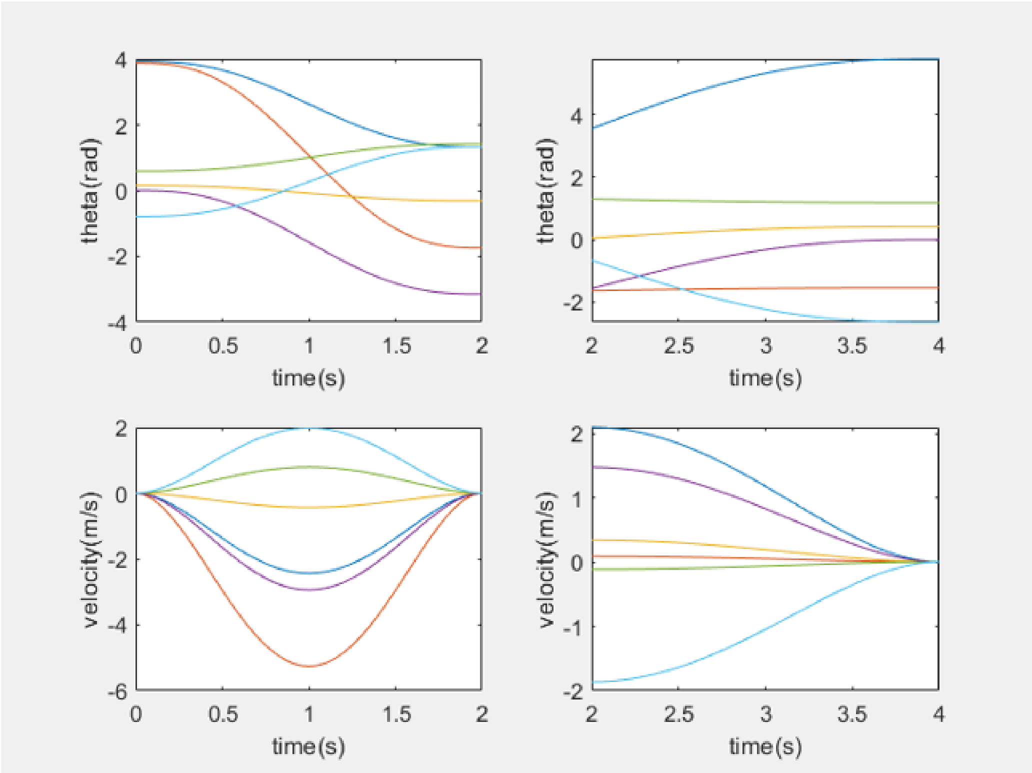
(1) lun



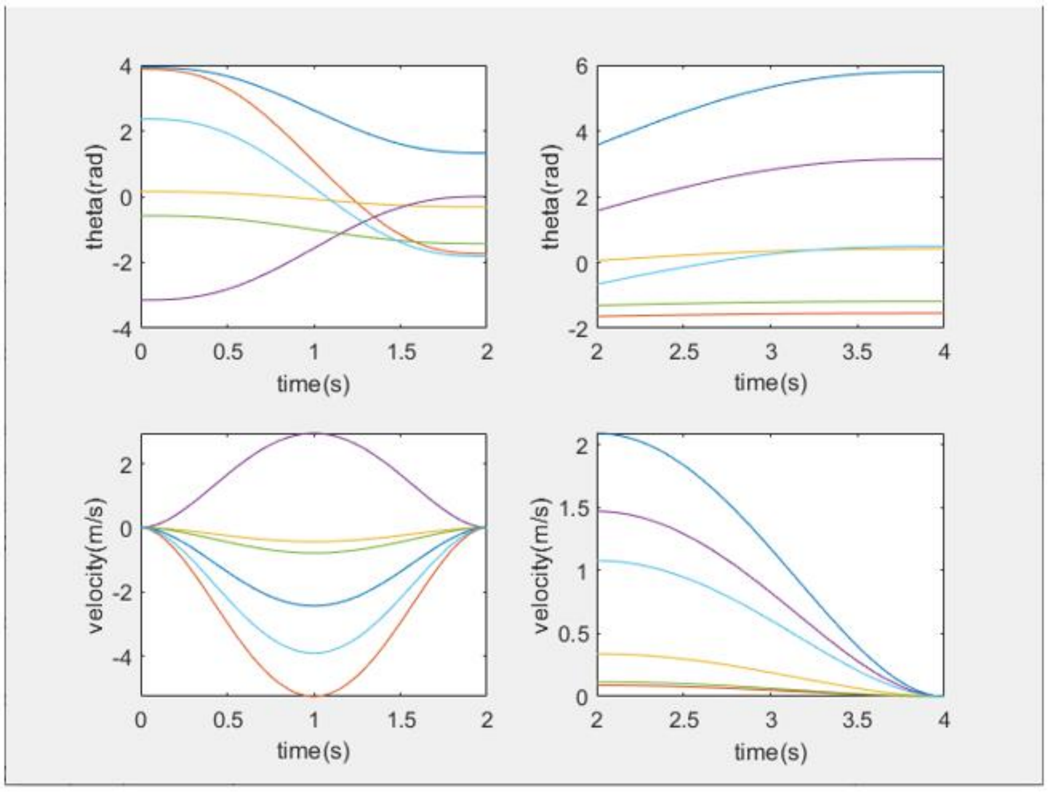
(2) luf



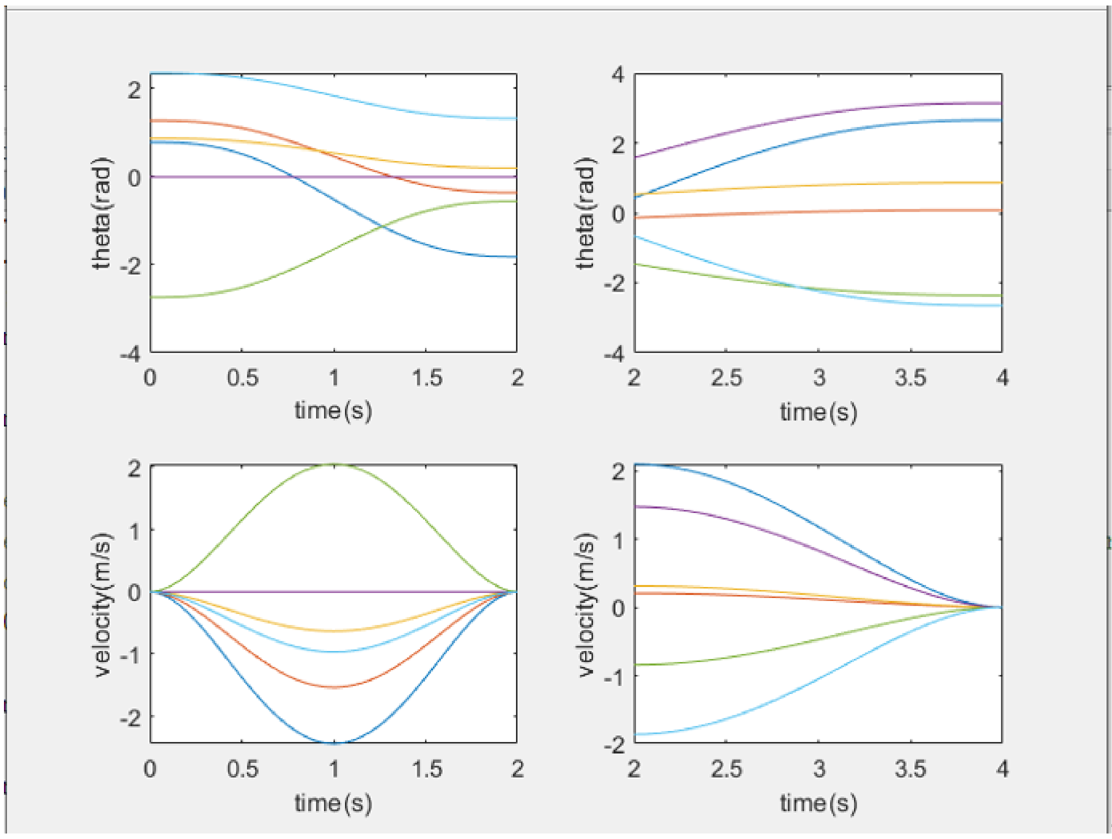
(3) ldn



(4) ldf

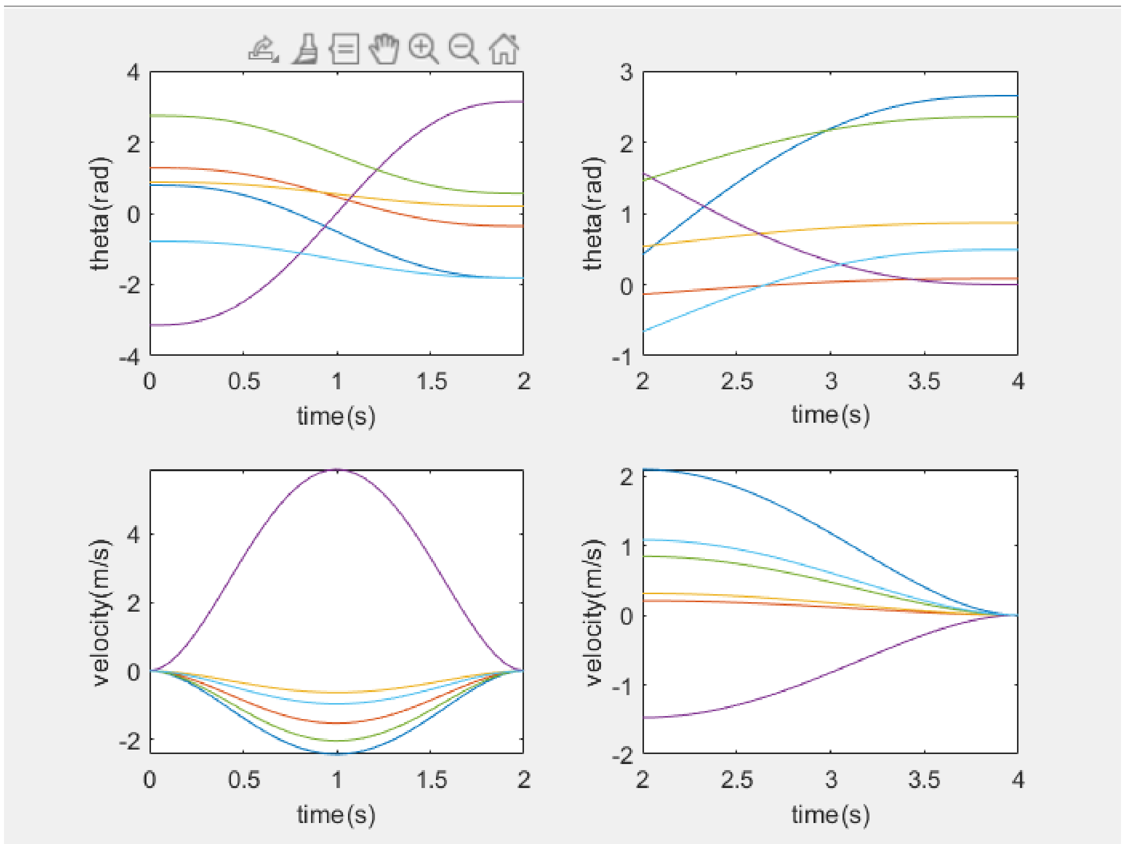


(5) run



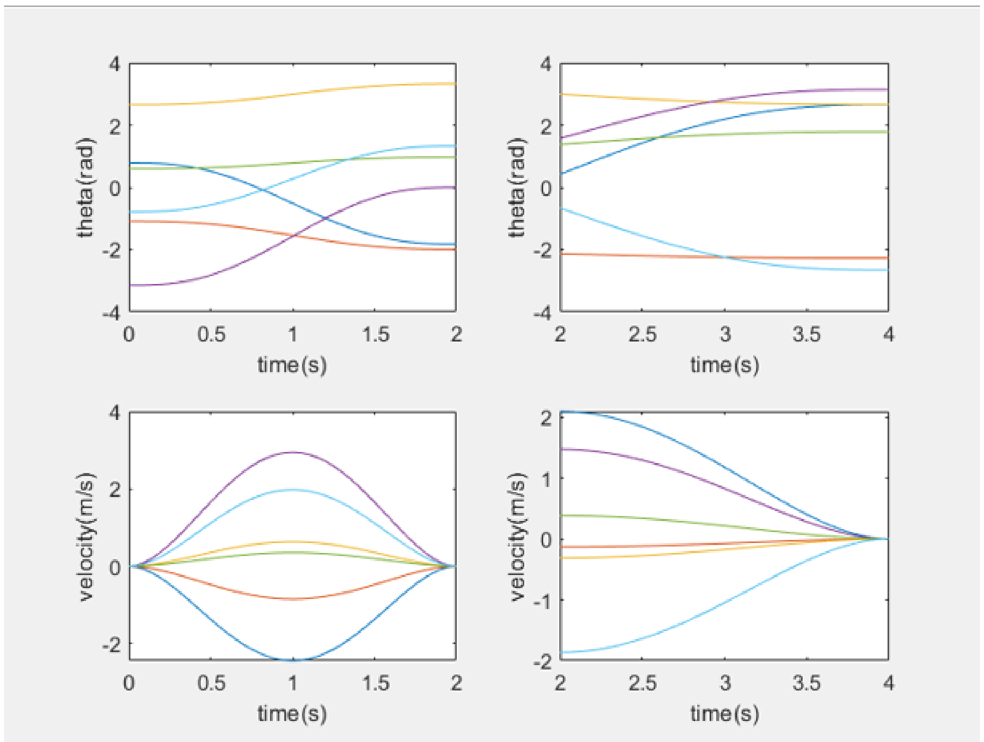
theta(rad)
time(s)
theta(rad)
time(s)
velocity(m/s)
time(s)
velocity(m/s)
time(s)

(6) ruf



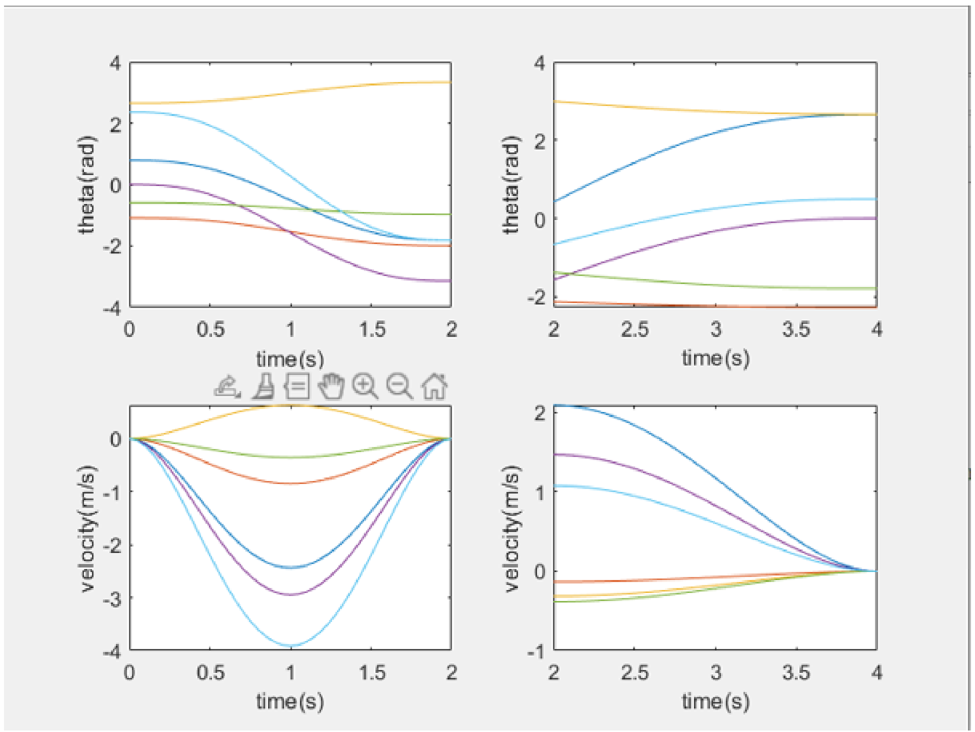
theta(rad)
time(s)
theta(rad)
time(s)
velocity(m/s)
time(s)
velocity(m/s)
time(s)

(7) rdn



theta(rad)
time(s)
theta(rad)
time(s)
velocity(m/s)
time(s)
velocity(m/s)
time(s)

(8) rdf



theta(rad)
time(s)
theta(rad)
time(s)
velocity(m/s)
time(s)
velocity(m/s)
time(s)