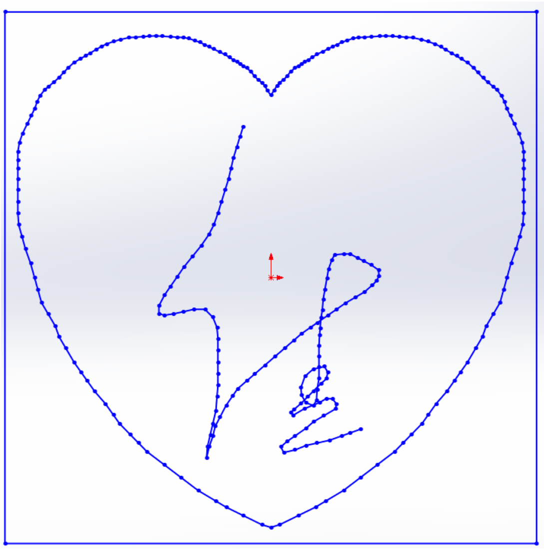
M.E. 530.646 Final Project Report

Bo Lei

1. Goal

The goal is to move the UR5 robot through the trajectory I set, so that it is able to draw a picture of a heart with a Chinese word inside if it. Here is the pattern, which is composed of a lot of points and lines connecting the consecutive points:



2. Basic idea

1. Inverse Kinematics
Move the robot to a very close end-effector position. Move the end-effector through the trajectory step by step.

2. Resolved-Rate Control
Move the robot with a small step each time. The position change in spatial frame is

$$x_{t+1} - x_t = J\dot{\theta} = J(\theta_{t+1} - \theta_t)$$

$$\Rightarrow \theta_{t+1} = J^{-1}(x_{t+1} - x_t) + \theta_t$$

In this way, we can move the end-effector through the trajectory by moving the joint angles each time.

2. Calculations relating to the code

1. homogenous transformation from body frame to spatial frame
Given the plane normal $\mathbf{n_0}$, and a point $\mathbf{P_0}$, compute the transformation between the body frame and the spatial frame:

$$\text{Body Frame normal: } n_b = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Ratation Axis: $\mathbf{n_r} = -\mathbf{n_0} \times \mathbf{n_b}$

Rotation Angle: $\theta = \arccos(\frac{-\mathbf{n0} \cdot \mathbf{nb}}{||\mathbf{n0}||\ ||\mathbf{nb}||})$

Then we can get the homogenous transformation from body frame to spatial frame:

$$g_{sb} = \begin{bmatrix} \exp(\widehat{\boldsymbol{n_r}}\theta) & p_0 \\ 0 & 1 \end{bmatrix}$$

2. When finishing drawing a pattern, move away from the board with height h:

$$p_h = g_{sb}\,(p_{current} + \begin{bmatrix} 0 \\ 0 \\ h \end{bmatrix})$$

3. Inverse and forward kinematics have been reported in previous lab.

4. Spatial Jacobian:

File：Ad.m, getJacobian.m

$$J = [\xi_1\ \xi_2^{'}\ \xi_3^{'}\ \xi_4^{'}\ \xi_5^{'}\ \xi_6^{'}\ ]$$

$$\xi_i^{'} = Ad_{e^{\hat{\xi}_1\theta_1}\cdots e^{\hat{\xi}_{i-1}\theta_{i-1}}}\ \xi_i$$

$e^{\hat{\xi}_i\theta_i}$ is computed in forward kinwmatics

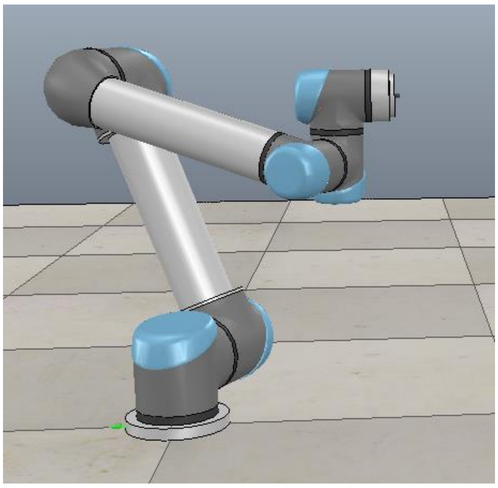
$$Ad_g = \begin{bmatrix} R & \hat{\mathbf{p}}R \\ \mathbf{1} & R \end{bmatrix}$$

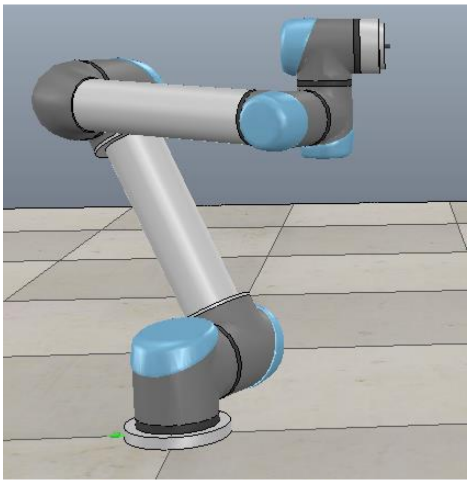
3. Screen shots showing the results

Inverse Kinematics(Using the 4th inverse kinematics solution):

File: DummyMain_IK.m

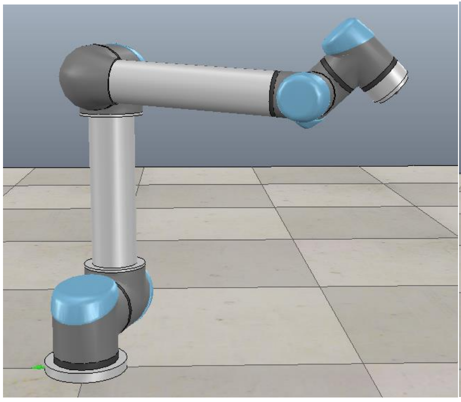
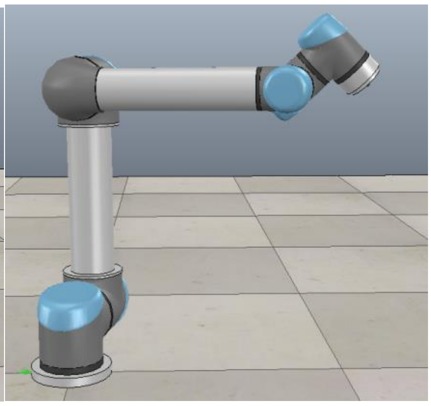
```
p0 = [0;-0.3;0.5];n = [0;1;0];
```





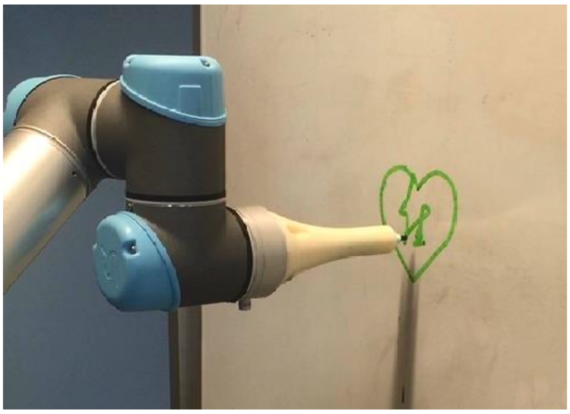
Resolved-Rate Control:
File: DummyMain_jacobian.m

```
p0 = [0;-0.5;0.5];n = [0;1;1];
```

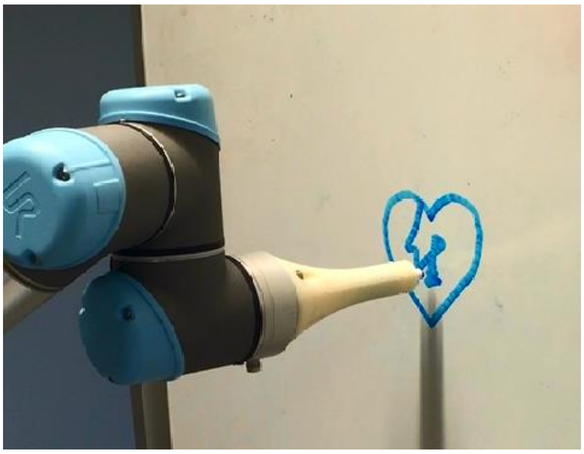



4. Drawing result

Inverse Kinematics：



Resolved-Rate Control:



5. Lab video

Inverse Kinematics: https://www.youtube.com/watch?v=9RyKTatm_xo
Resolved-Rate Control: https://www.youtube.com/watch?v=xy0HFCJL9QM