

1) 关于传感器坐标轴的问题

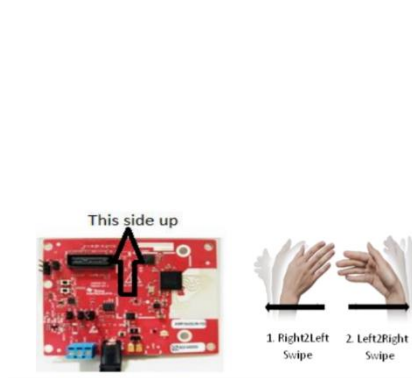


图 1

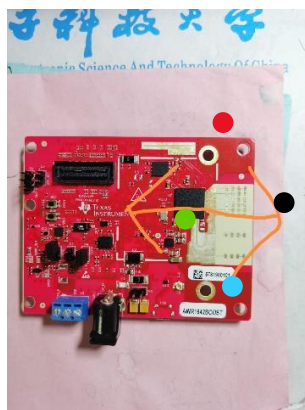


图 2



图 3

受到官方示意图图 1 的影响，我将板子如图 2 所示放置进行实验，其中双向箭头是运动目标的运动路径（左右移动的手），然而，图 3 所示的坐标轴才是正确的，图 2 的运动路径垂直于 xy 平面，因此经常不能捕捉到运动目标。

多次实验后，我才注意到角度图中角度总小于 0，而不是预料中的大于 0 和小于 0 的点都有，这说明每次捕捉到的运动目标的 x 坐标小于 0。实验时我一般将手置于发射天线上端，于是我猜测传感器的坐标轴应当如图 3 所示。

之后，我依次将手指放在图 2 所示的四个小圆点处，距离传感器平面约 8cm 处，发现绿点和黑点处均未捕捉到手指，而红点处捕捉到的手指角度为正值，蓝点处捕捉到的手指角度为负值。

传感器的坐标轴应当如图 3 所示。

2) sensorStop 执行出错问题

```
Opening configuration file awr16xx.cfg ...
Sending configuration from awr16xx.cfg file to AWR16xx ...
sensorStop
操作在以下过程中被用户终止 serial/fgetl (第 58 行)

位置 RadarRead (第 25 行)
echo = fgetl(hControlSerialPort); % Get an echo of a command
```

图 4

```
prompt_1=[];
for k=1:freadnums
    Data_dec= fread(hdataSerialPort); % Ge
    prompt_1=[prompt_1;Data_dec];
end
save(filename,'prompt_1');
fprintf(hControlSerialPort, 'sensorStop');
fclose(hControlSerialPort);
```

图 5

问题描述：第一次运行代码成功后，第二次再运行总会卡如图 4 所示位置。若每次运行代码成功后重启板子再运行代码则无此问题。

详情：未知原因使 sensorStop 指令失效，正常情况下执行此指令 DS3 指示灯将熄灭（GPIO_1 置为 0），等到执行 sensorStart 指令 GPIO_1 置 1，DS3 指示灯亮起。失效情况下经上次成功运行代码后，GPIO_1 为 1，指示灯亮起，执行此指令却无法复位 GPIO_1，没有响应，程序得不到'Done'的回复，只能等待串口读取超时。重启板子则可以对 GPIO_1 复位。

解决方法：如图 5 所示，每次运行代码获取完数据后执行 sensorStop 指令，复位 GPIO_1 后再关闭串口。

影响：原本发送的指令以 sensorStop 指令开头，以 sensorStart 指令结尾，修改后相当于延时一段时间后执行 sensorStop 指令。修改前每次成功运行代码 sensorStart 指令的回复是'Debug: Init Calibration Status = 0x17fe\n\nDone'，修改后除了启动板子第一次运行代码 sensorStart 指令的回复与修改前相同，此后每次（不重启板子）运行代码回复均为'Done'。

3) 处理静态物体

沿图 3 所示双向箭头方向移动手，某次实验采集到的数据经处理后如下图 6、7、8、9 所示。

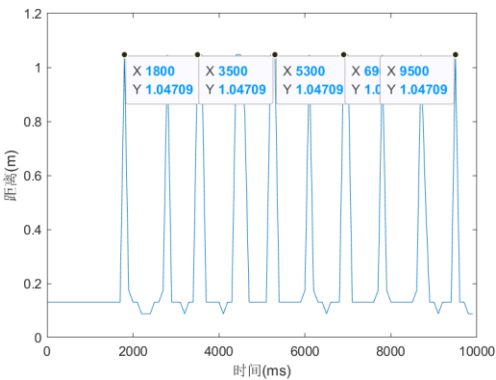


图 6

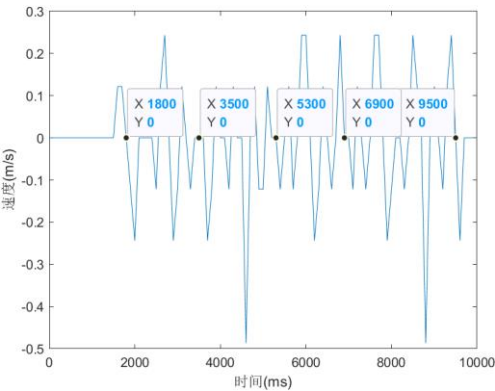


图 7

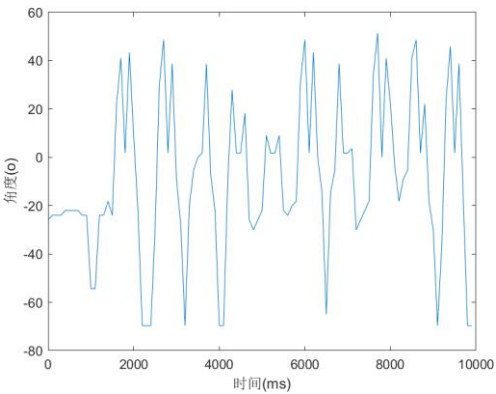


图 8

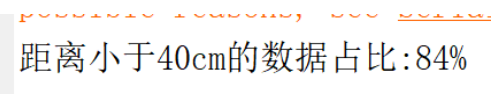


图 9

经统计得到目标为手的数据占比约 84%。观察图 6 和图 7 可以发现，图 6 中所有出现毛刺的点在图 7 中纵坐标均为 0，说明毛刺代表目标为静态物体。

处理这些毛刺目前主要有两条思路。一是不必每一帧数据都用于绘图，可以采用每隔几帧的数据用于绘图一次；另一思路是采用插值的方法替换毛刺处的数据。

第一种思路代码如下图 10 所示。绘图代码如下图 11 所示。

```
[range, doppler, angle]=UartProcess(udata);
len=size(doppler, 2);
t=(0:len-1)*100;
index=find(range<0.4);
disp(['距离小于40cm的数据占比:', num2str(1
n=2;
n_range=range(1:n:end);
n_t=t(1:n:end);
```

图 10

```
figure();
plot(t, range);
xlabel('时间(ms)');ylabel('距离(m)');
figure();
plot(n_t, n_range);
xlabel('时间(ms)');ylabel('距离(m)');
```

图 11

通过修改 n 的值，对帧数进行“采样”，结果如下图所示。

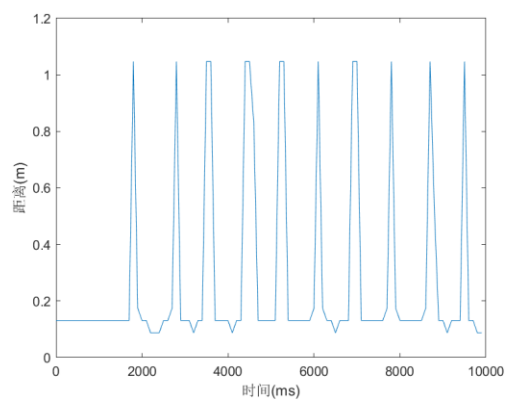


图 12 $n=1$

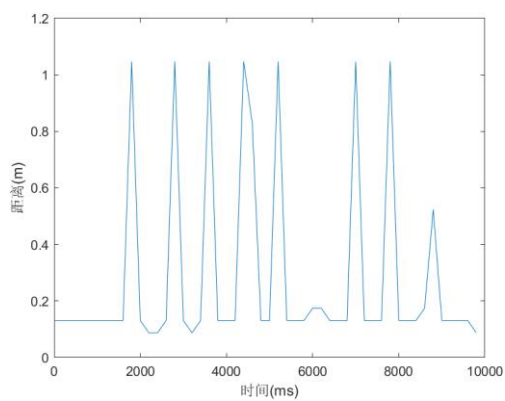


图 13 $n=2$

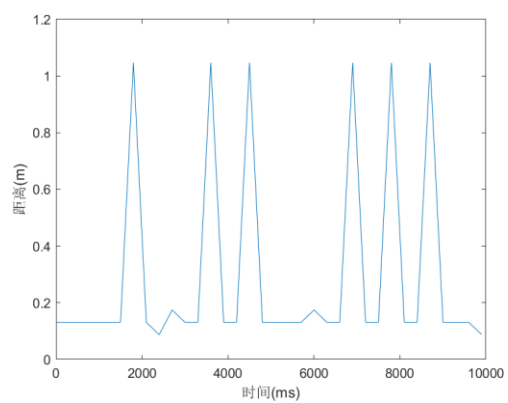


图 14 $n=3$

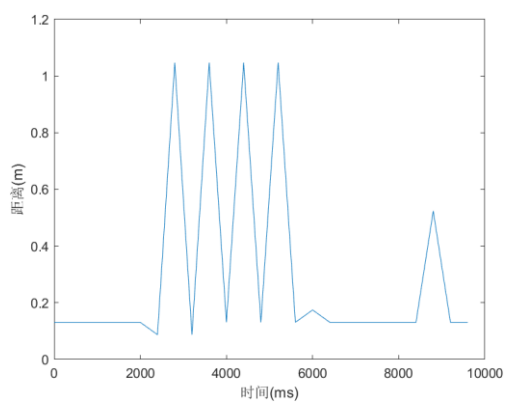


图 15 $n=4$

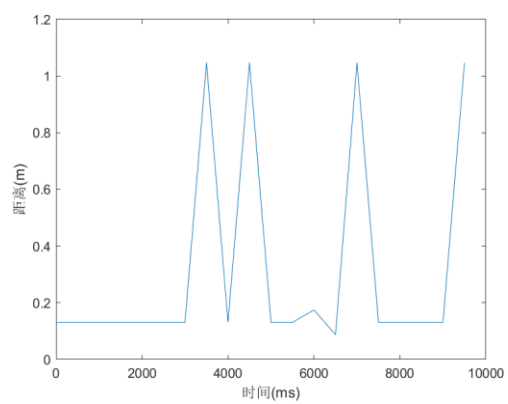


图 16 $n=5$

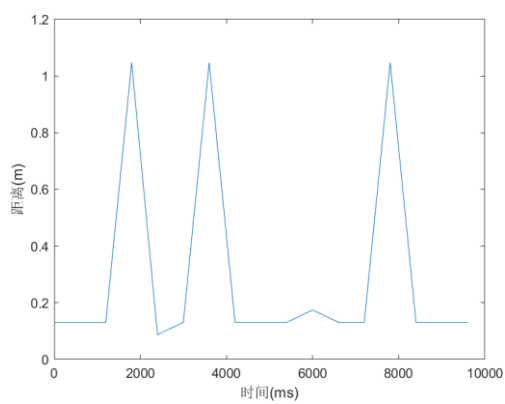


图 17 $n=6$

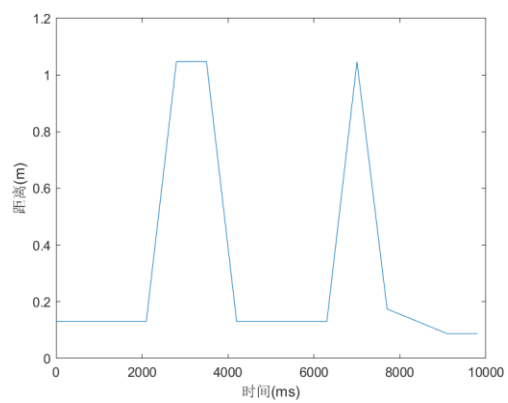


图 18 $n=7$

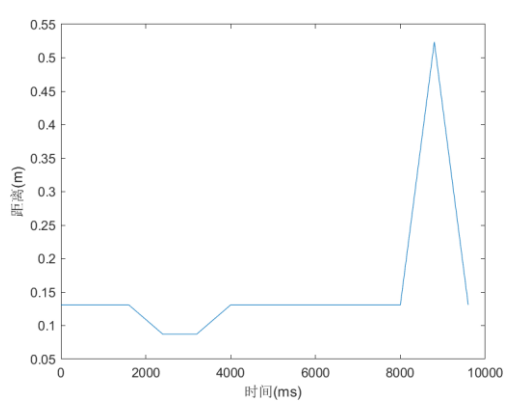


图 19 $n=8$

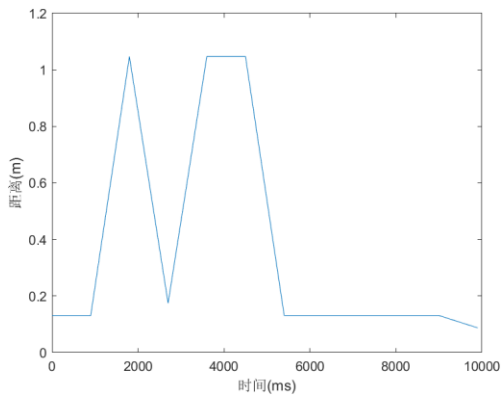


图 20 n=9

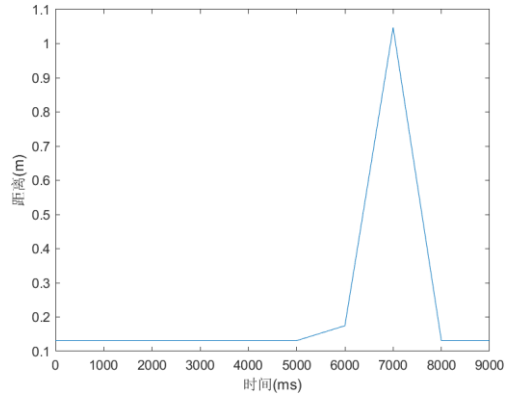


图 21 n=10

可以看到， n 的值并非越大越好，原因在于无法预测每隔 n 帧取的数据中是否会包含静止目标。但一个较大的 n 依然可以消除较多的毛刺。此外，速度和角度数据也应当取相同的 n 值进行“采样”。

第二种思路代码如下图 22 所示。绘图代码如图 23 所示。

```
[range, doppler, angle]=UartProcess(udata);
len=size(doppler, 2);
t=(0:len-1)*100;
index=find(range<0.5);
n_range=interp1(index, range(index), 1:len, 'linear');
```

图 22

```
figure();|
plot(t, range);
xlabel('时间(ms)');ylabel('距离(m)');
figure();
plot(t, n_range);
xlabel('时间(ms)');ylabel('距离(m)');
```

图 23

采用的插值方法是双线性插值算法。结果如下图所示。

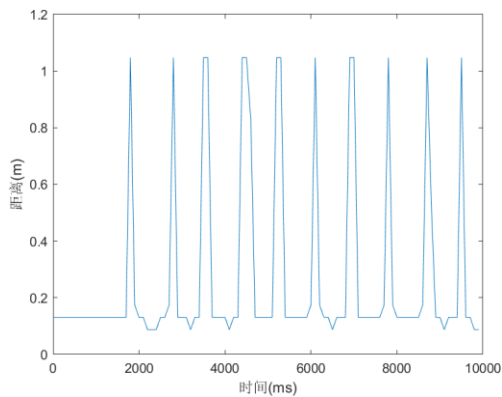


图 24 插值前

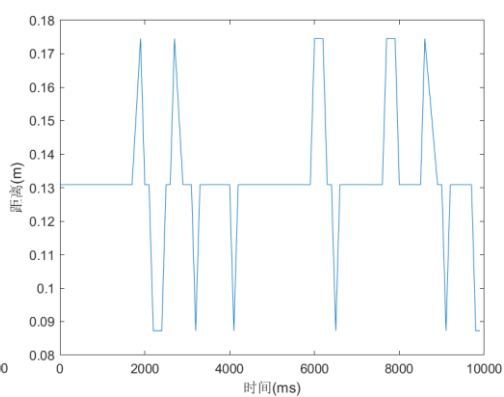


图 25 插值后

同样，速度和角度数据也应用索引为 $index$ 的数据进行插值，将目标为静止物体的数据替换掉。

4) 帧数据未完整读取

位置 1 处的索引超出数组边界(不能超出 73320)。
出错 RadarRead>UartProcess (第 83 行)
fdata=udata((cur+1):(cur+framelen),:);

图 26

```
framelen=hex2dec(framelenhex);
next=cur+framelen;
if next>len
    break
end
fdata=udata((cur+1):(cur+framelen),:);
```

图 27

问题描述：从串口读取的数据偶尔会出现最后一帧的数据不完整的情况。

详情：最后一帧虽然不完整，但仍能从帧头部分解析其帧长 framelen，代码试图将该帧的 framelen 个字节用变量记录，交由函数处理，自然会导致索引越界而报错，如图 26 所示。

解决方法：如图 27 所示，在取得帧的全部数据前先判断其索引是否越界，若越界，则丢弃该帧。

影响：不完整的帧中可能包含有 detected objects TLV，丢弃该帧则失去目标的该数据。

5) 手势数据记录

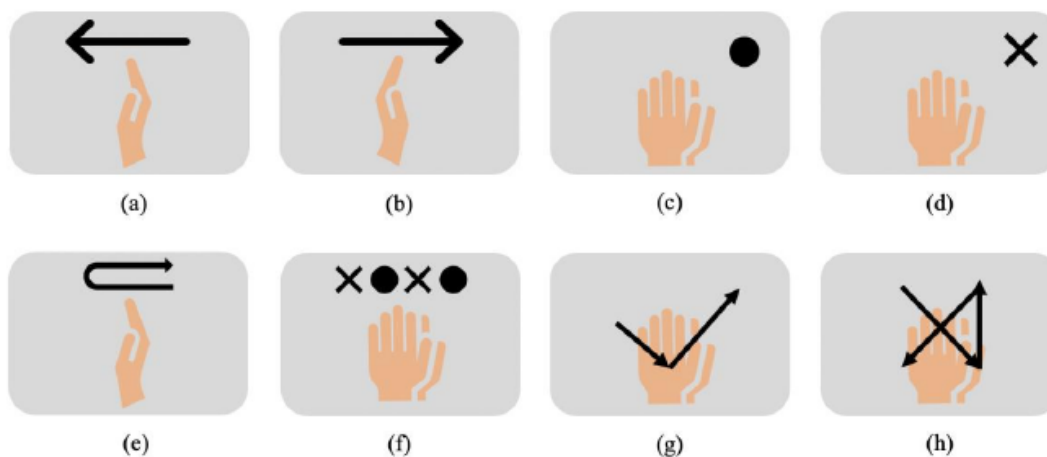


图 2-6 八类手势示意图。(a)左挥；(b)右挥；(c)上拉；(d)下推；(e)左右小挥；(f)上下轻拍；(g)画勾；(h)画叉

八类手势具体设计如下：

- (1) 左挥：在雷达上方 10cm-20cm 高度，手掌及小臂快速从左到右挥，动作持续 0.5s 到 1s
- (2) 右挥：在雷达上方 10cm-20cm 高度，手掌及小臂快速从右到左挥，动作持续 0.5s 到 1s
- (3) 上拉：手掌于雷达上方 5cm-10cm 处，快速上拉至 40cm 处以上，动作持续 0.5s 到 1s
- (4) 下推：手掌于雷达上方 40cm 处，快速下推至 5cm-10cm 处，动作持续 0.5s 到 1s
- (5) 左右小挥：在雷达上方 10cm-20cm 高度，手掌快速先向左后向右小挥两次，动作持续 0.5s 到 1s
- (6) 上下轻拍：手掌于雷达上方 20cm 处，小幅度轻拍两次，动作持续 0.5s 到 1s
- (7) 画勾：手掌于雷达斜上方约 20cm 处，先左上挥至中间下方，再挥至右上，动作持续约 1s 到 1.5s
- (8) 画叉：手掌于雷达斜上方约 20cm 处，先左上挥至右下，后右上挥至左下，动作持续约 1s 到 1.5s

图 28

如图 28 所示, 8 类手势位于雷达上方高度不超过 50cm, 使用插值法处理数据时 interp1 函数的输入为距离小于 50cm 的数据及其对应的索引。手势的持续时间不超过 2s, 可以设置数据串口 timeout 为 2, 即每 2s 读一次串口。

```

index=find(range<0.5);
n_range=interp1(index, range(index), 1:len, 'linear');
n_doppler=interp1(index, doppler(index), 1:len, 'linear');
n_angle=interp1(index, angle(index), 1:len, 'linear');

```

```

sphandle = serial(comPortString, 'Ba
set(sphandle, 'Timeout', 15);
set(sphandle, 'Terminator', '');
set(sphandle, 'InputBufferSize', buf
set(sphandle, 'Timeout', 2);

```

图 29

图 30

另外, 在做手势时, 存在不明原因导致某些 detected objects TLV 中出现距离约为 8.73cm、角度约-70°的目标, 将其屏蔽后图像效果较好, 相关代码如图 31、32 所示。

```

p=typecast(uint8(hex2dec(['BF';'88';'A7';'BD'])), 'single');
q=typecast(uint8(hex2dec(['85';'BE';'F8';'3C'])), 'single');
r=sqrt(p^2+q^2);

```

```

dst=sqrt(x^2+y^2+z^2);
if dst==r
    continue
end

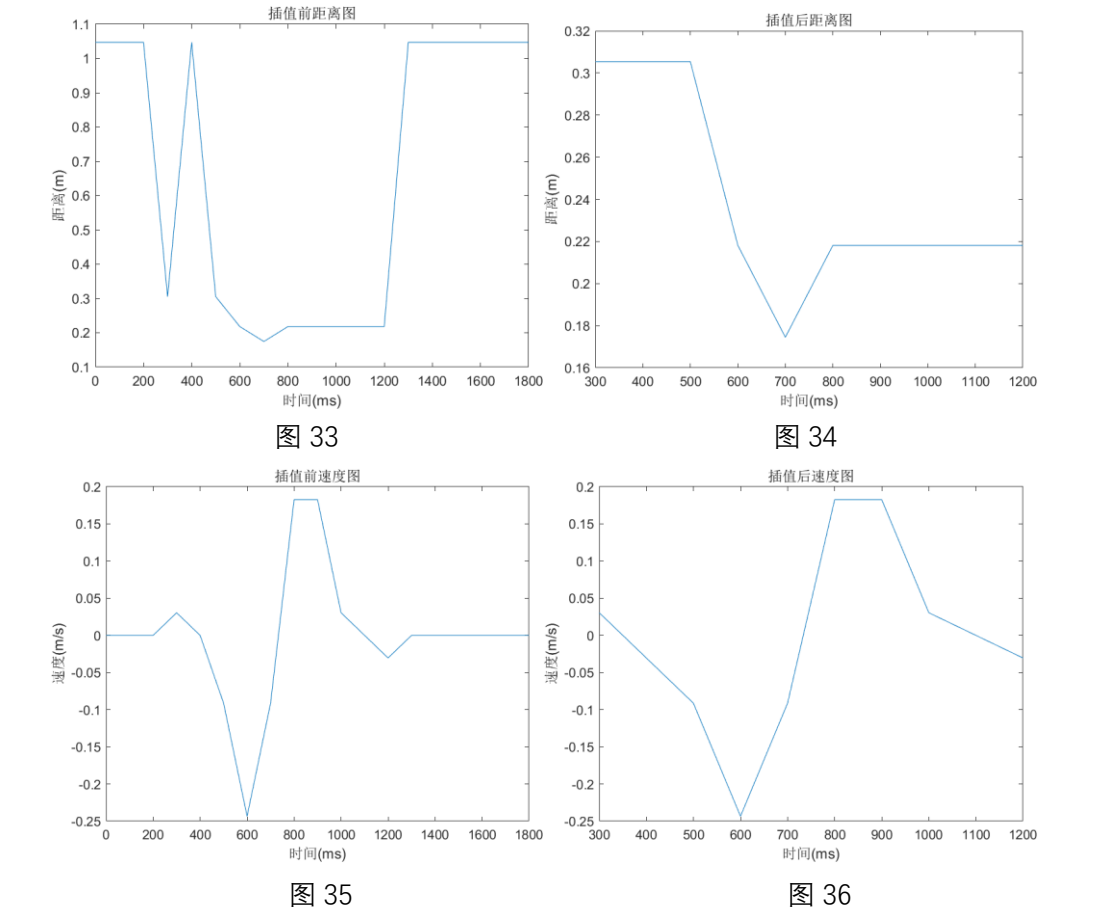
```

图 31

图 32

手势数据部分结果如下图所示。

(a) 左挥



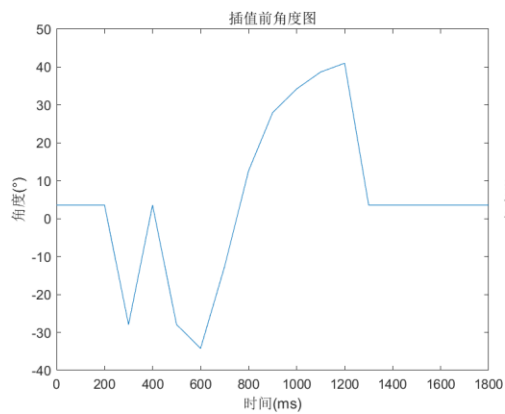


图 37

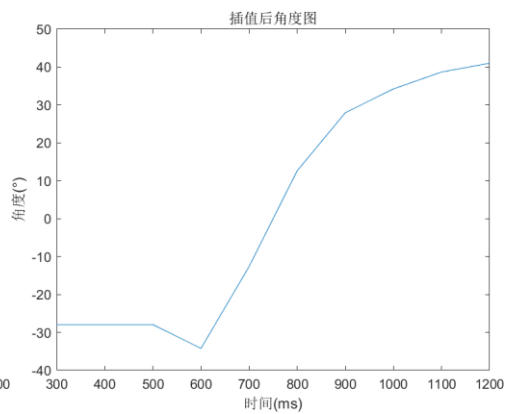


图 38

(b) 右挥

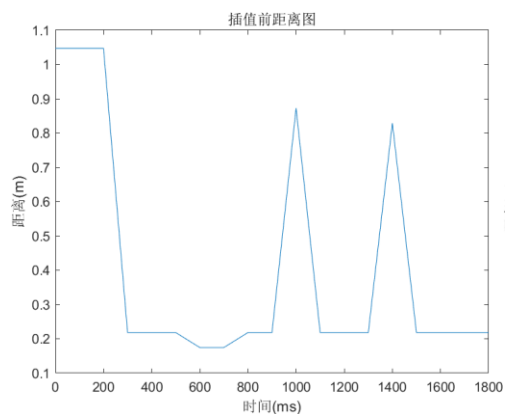


图 39

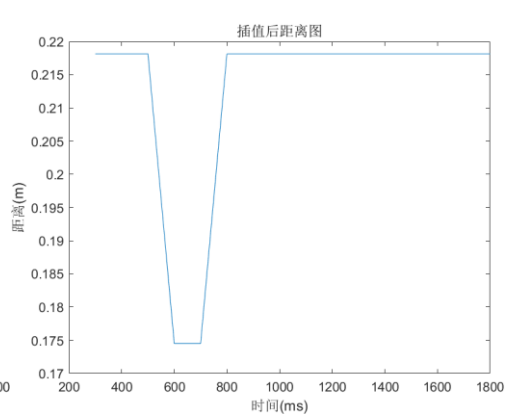


图 40

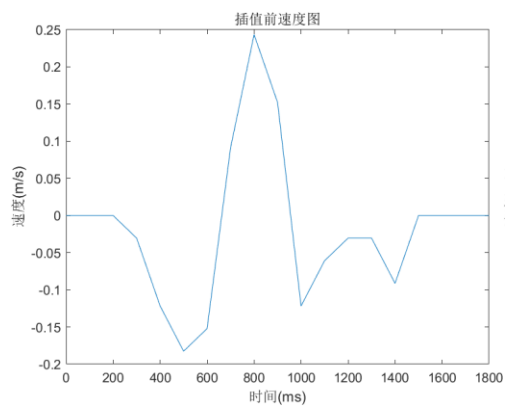


图 41

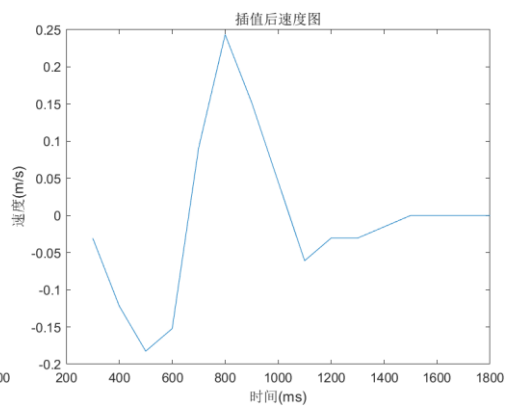


图 42

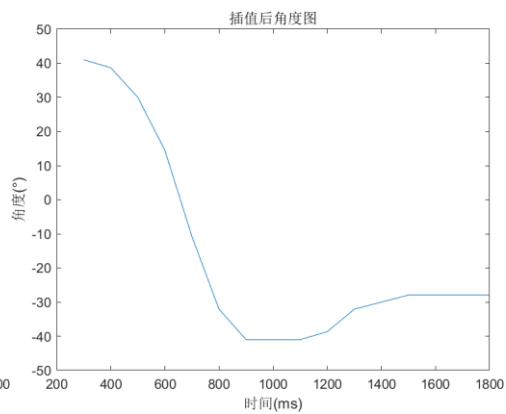
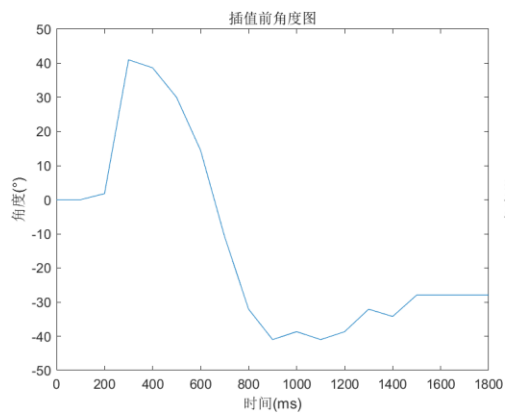


图 43

(c) 上拉

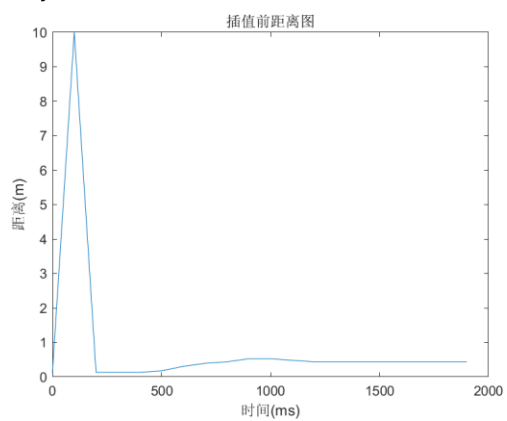


图 44

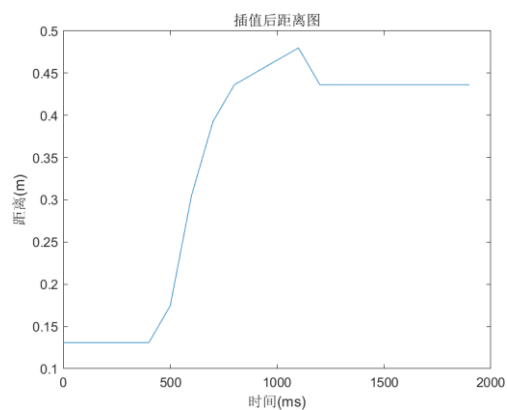


图 45

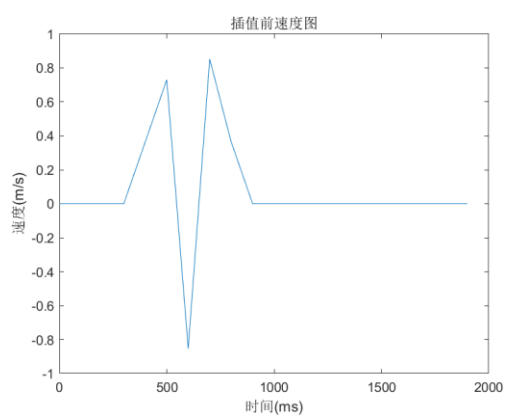


图 46

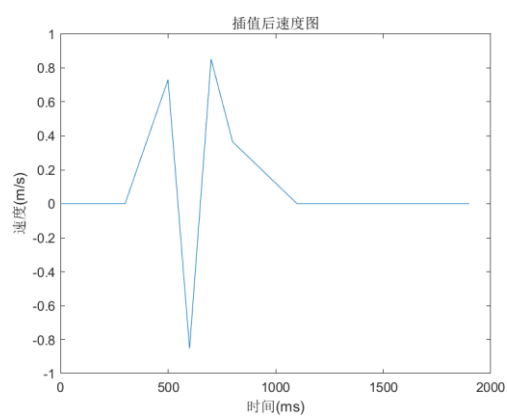


图 47

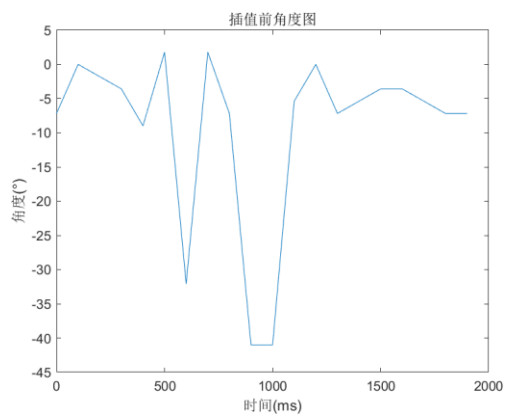


图 48

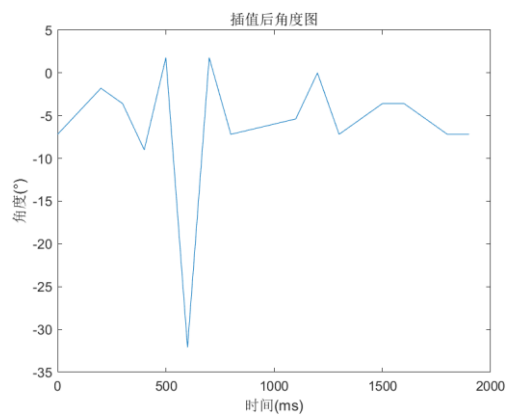


图 49

(d) 下推

图 50

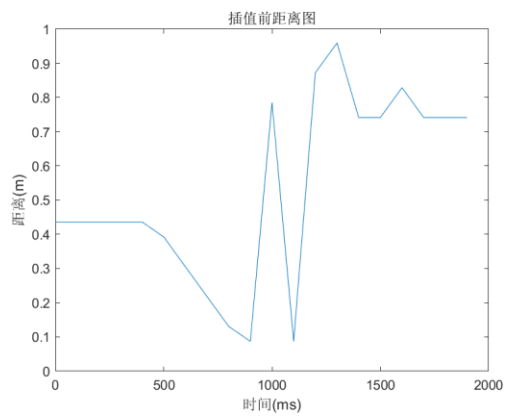


图 51

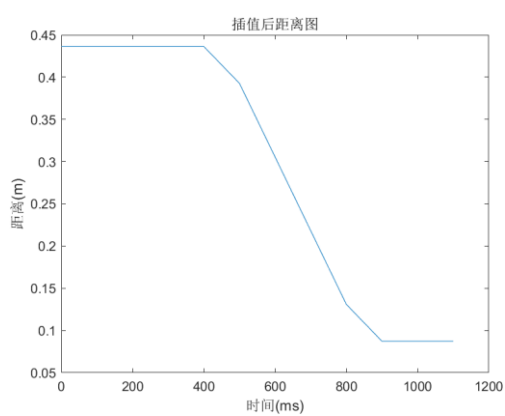


图 52

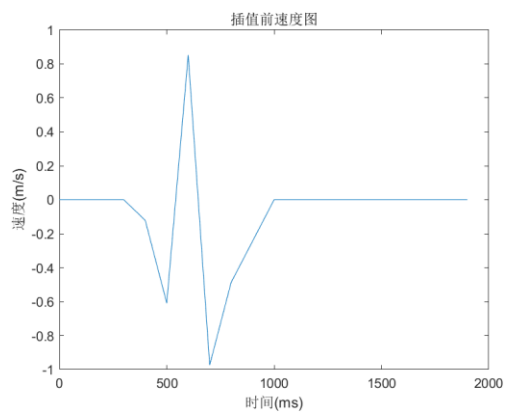


图 53

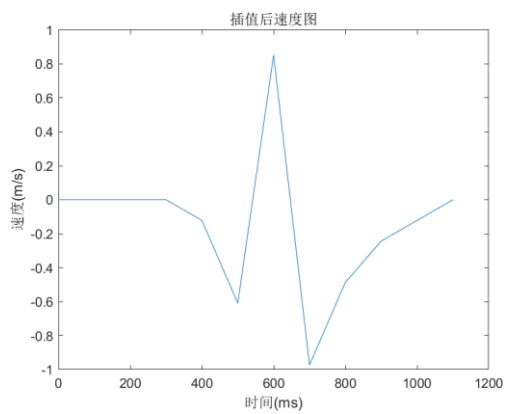


图 54

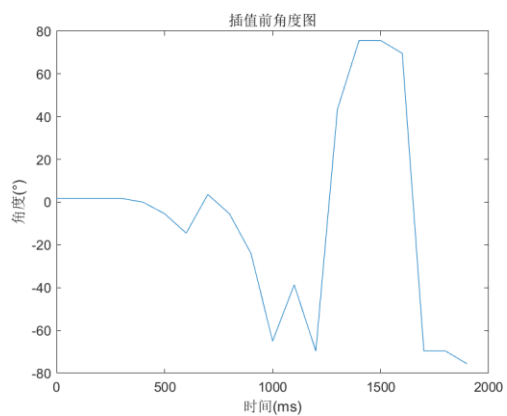


图 55

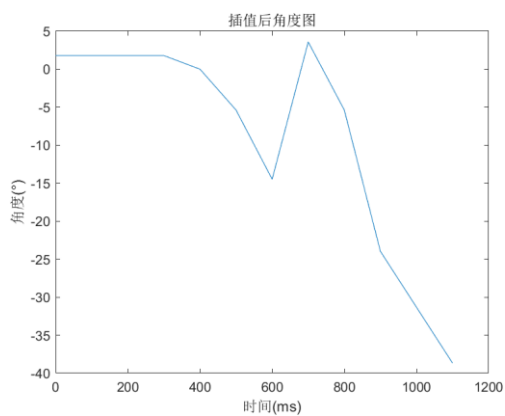


图 56

(e) 左右小挥

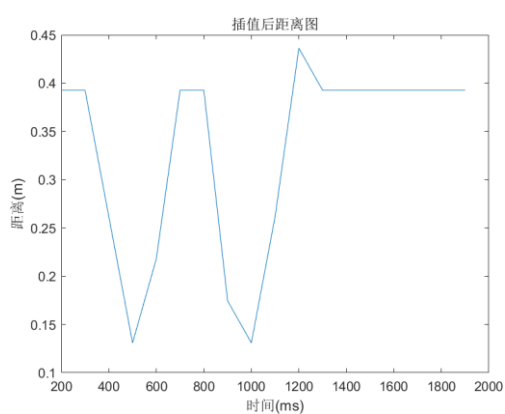
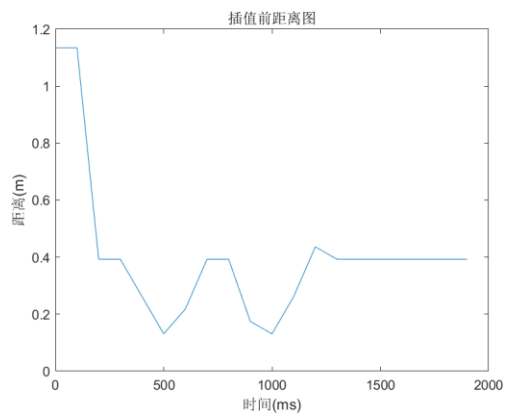


图 57

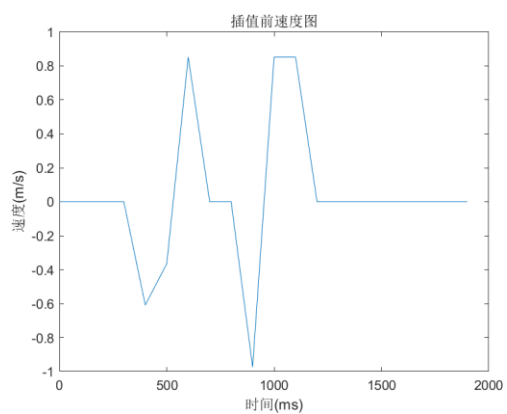


图 58

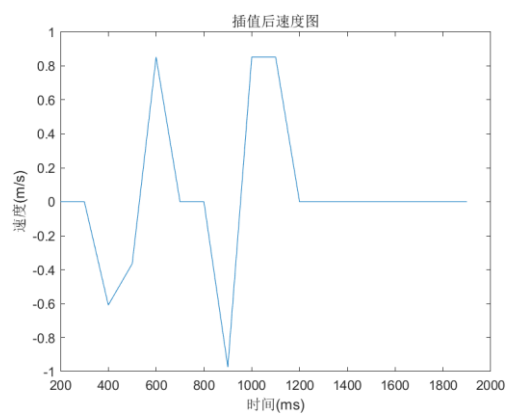


图 59

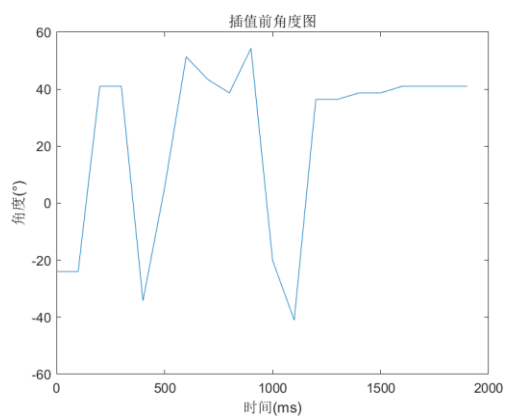


图 60

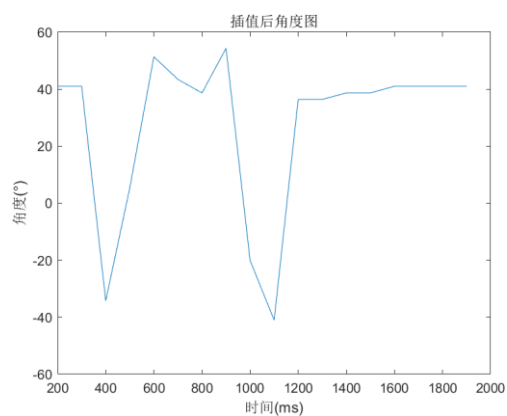


图 61

图 62

(f) 上下轻拍

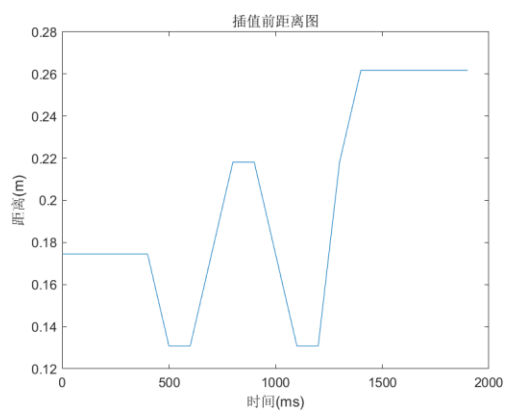


图 63

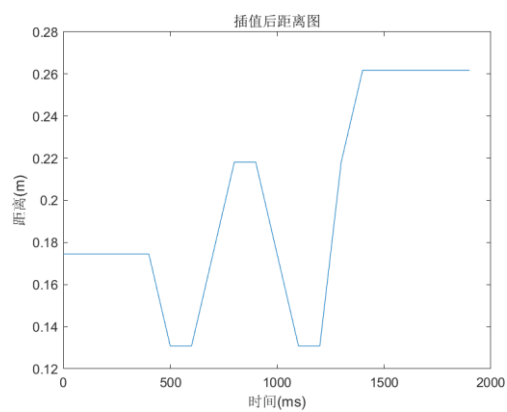


图 64

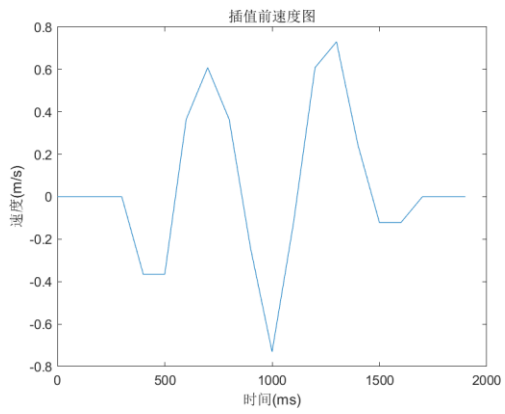


图 65

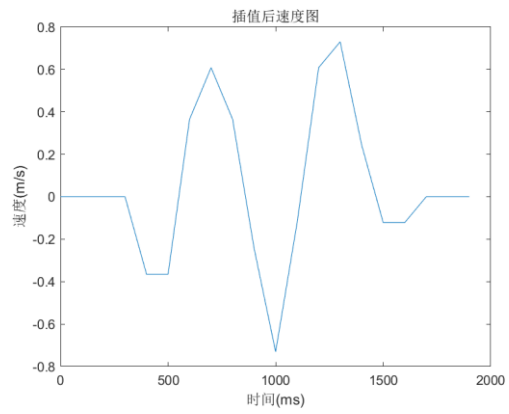


图 66

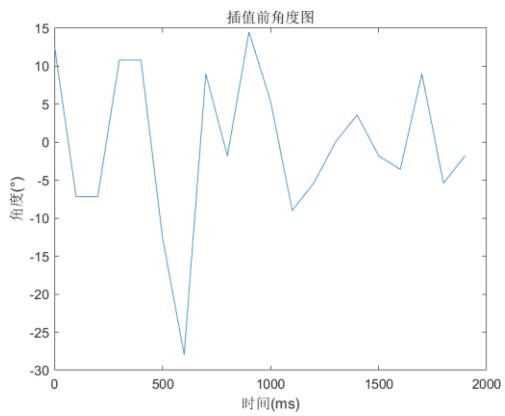


图 67

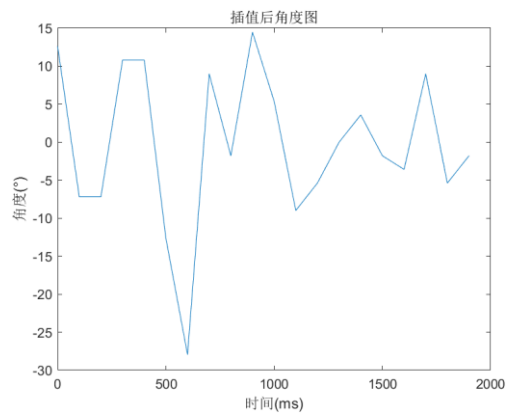


图 68

(g) 画勾

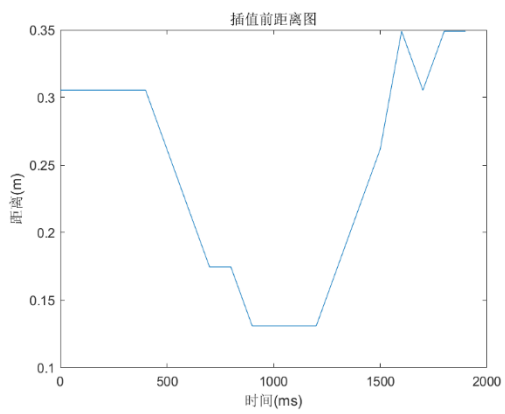


图 69

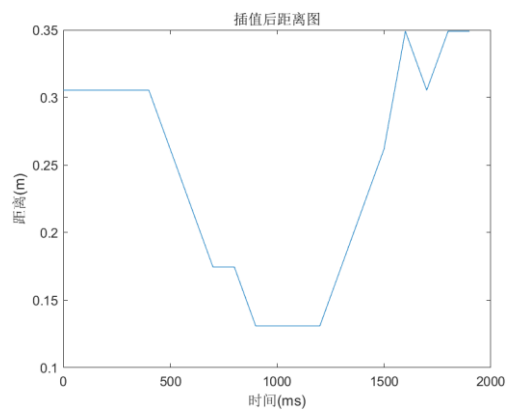


图 70

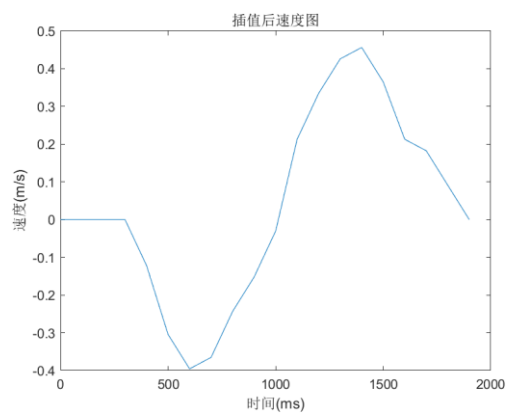
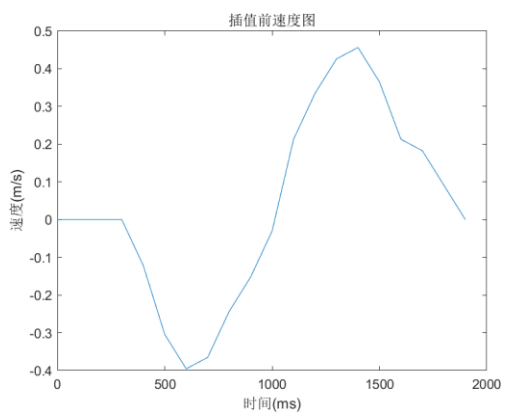


图 71

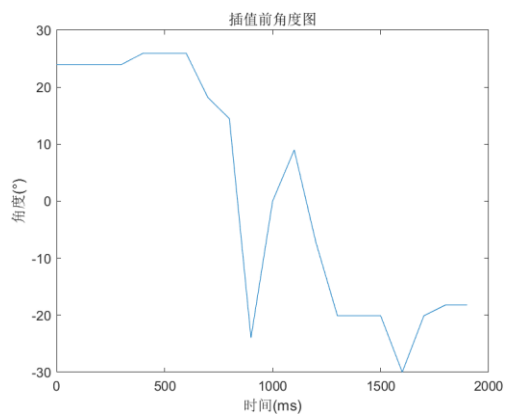


图 72

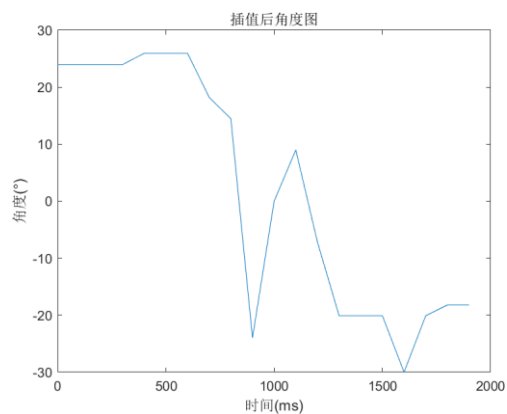


图 73

图 74

(h) 画叉

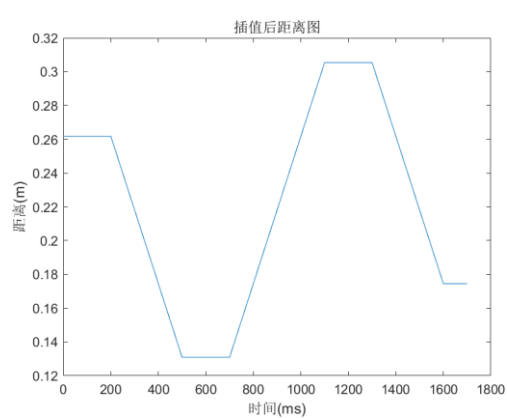
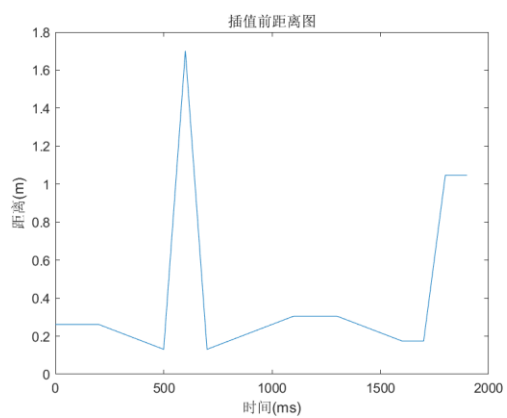


图 75

图 76

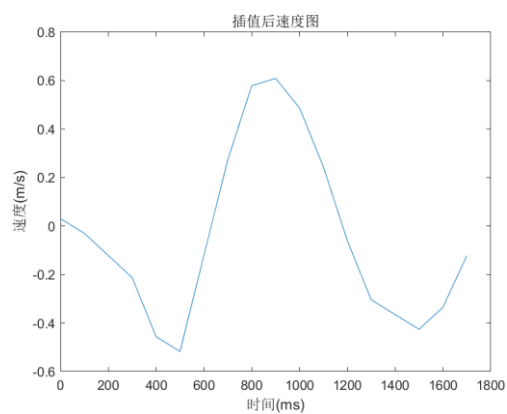
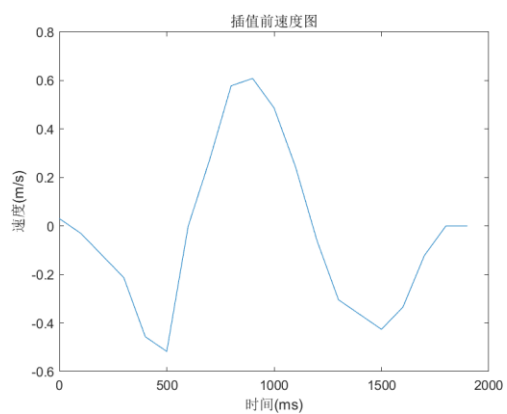


图 77

图 78

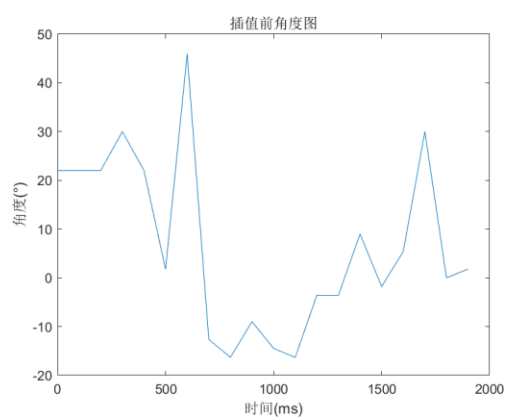


图 79

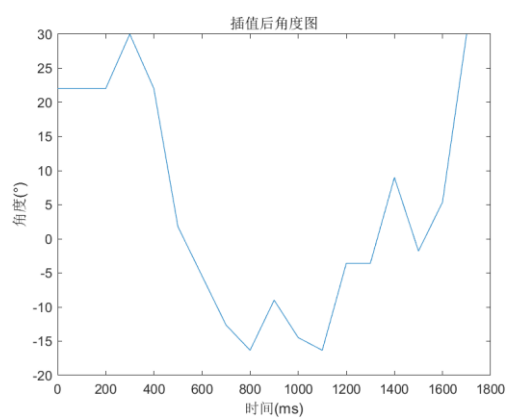


图 80