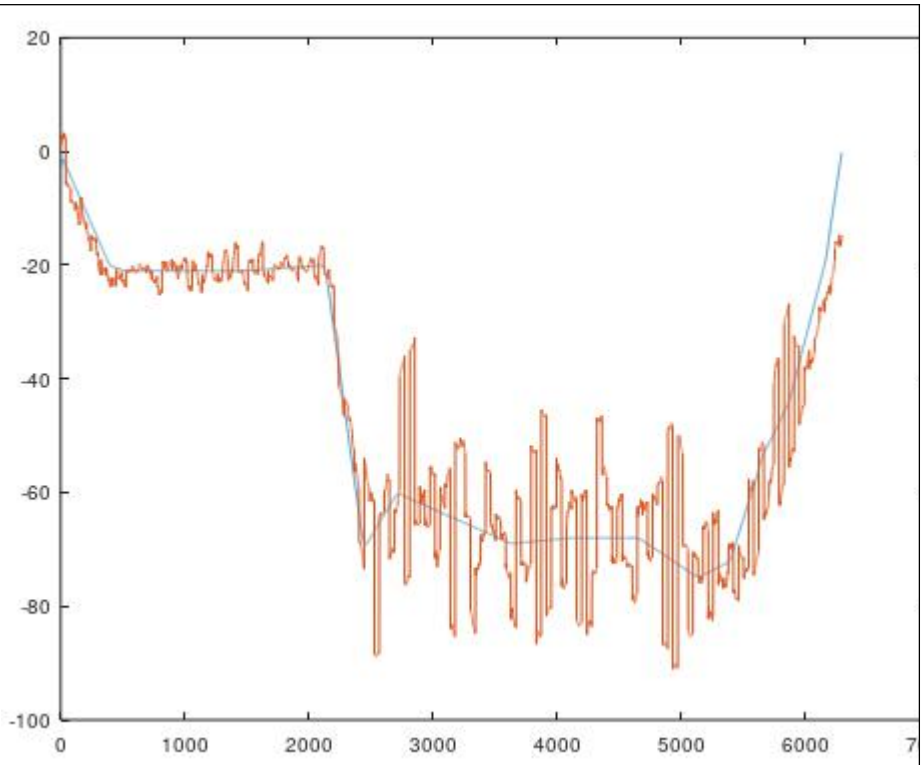
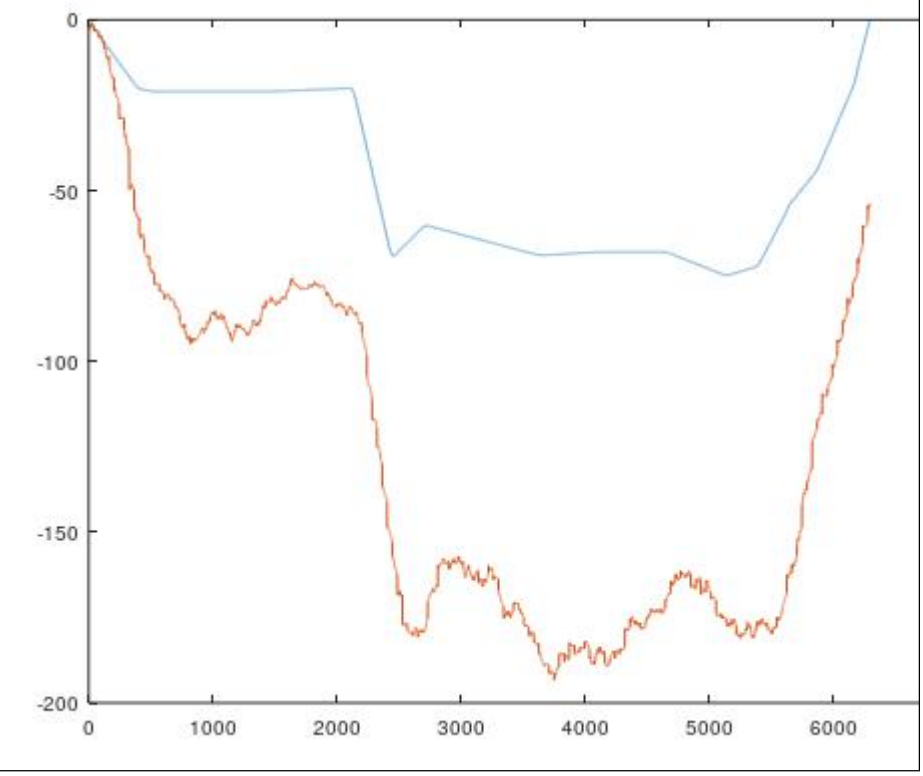
iter=40;
#setting Input for inertial sensor and positioning
AIMU=0.1;
sigmaIMU=20;
ADVL=0.1;
sigmaDVL=2;
AUSBL=40;
sigmaUSBL=0.1;

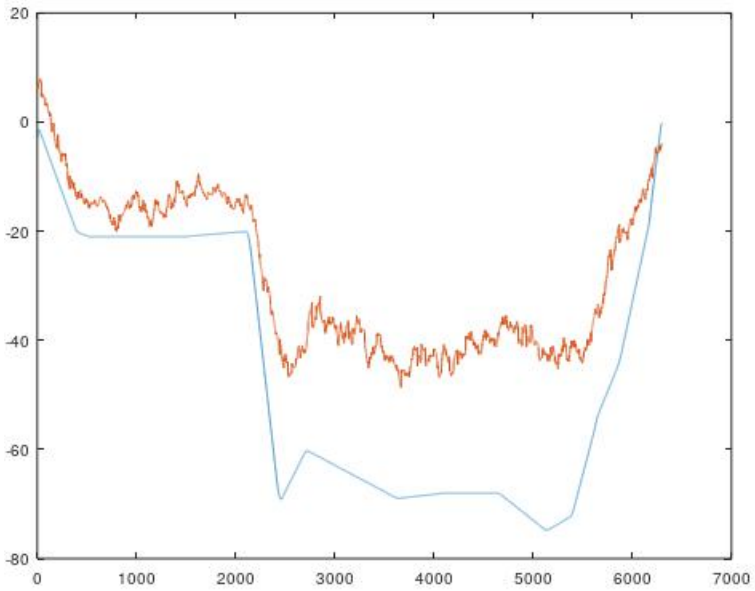
Z axis all



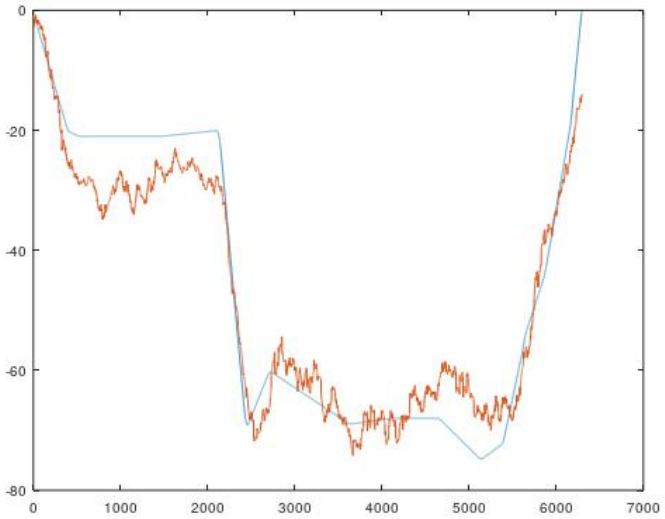

```
 xUSBL = G1D(range,xCusbl(iUsbl,4)-xCusbl(iUsbl-1,4),sigmaUSBL,AUSBL);
   xDVL = G1D(range,localTrajvar,35,20);
   xIMU = G1D(range,localTrajvar,35,20);
   tUsbl=xCusbl(iUsbl+1,1);
```



```
xUSBL = G1D(range,xCusbl(iUsbl,4)-xCusbl(iUsbl-1,4),sigmaUSBL,AUSBL);
   xDVL = G1D(range,localTrajvar,20,20);
   xIMU = G1D(range,localTrajvar,20,20);
   tUsbl=xCusbl(iUsbl+1,1);
```

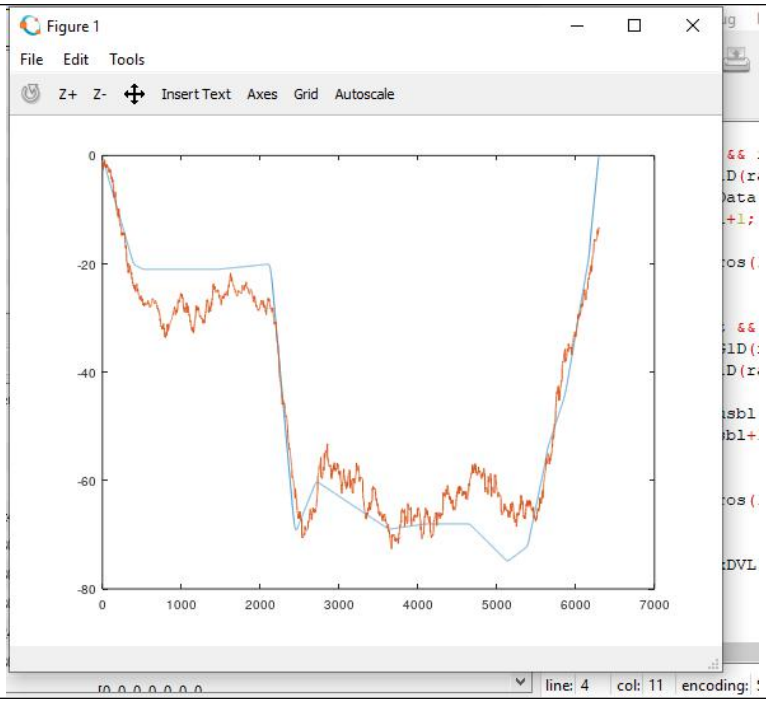


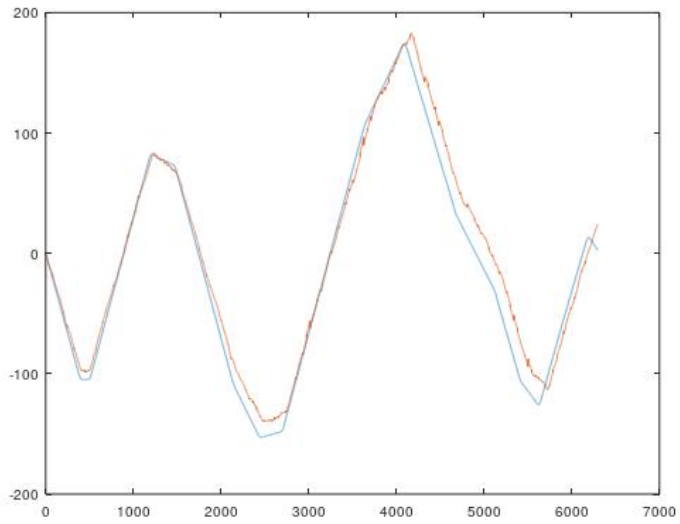

```
if tUsbl<=t && iUsbl<158
   xUSBL = G1D(range,xCusbl(iUsbl,4)-xCusbl(iUsbl-1,4),sigmaUSBL,AUSBL);
   xDVL = G1D(range,localTrajvar,30,20);
   xIMU = G1D(range,localTrajvar,30,20);
   tUsbl=xCusbl(iUsbl+1,1);
   iUsbl=iUsbl+1;
```

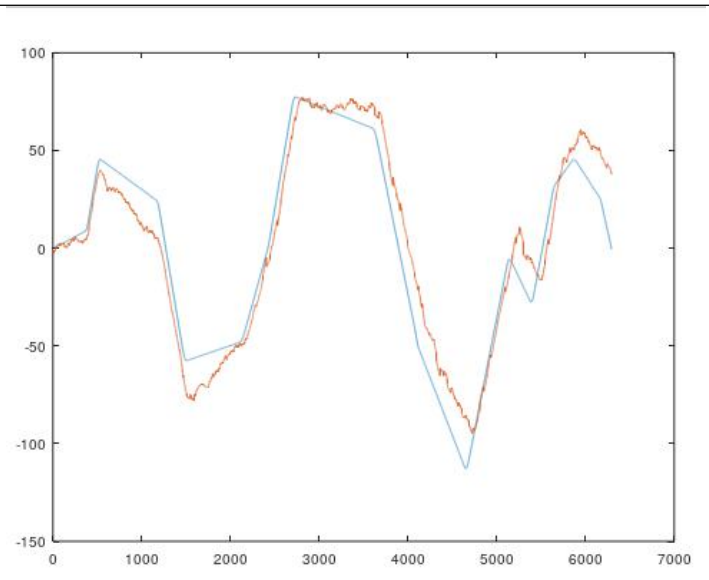


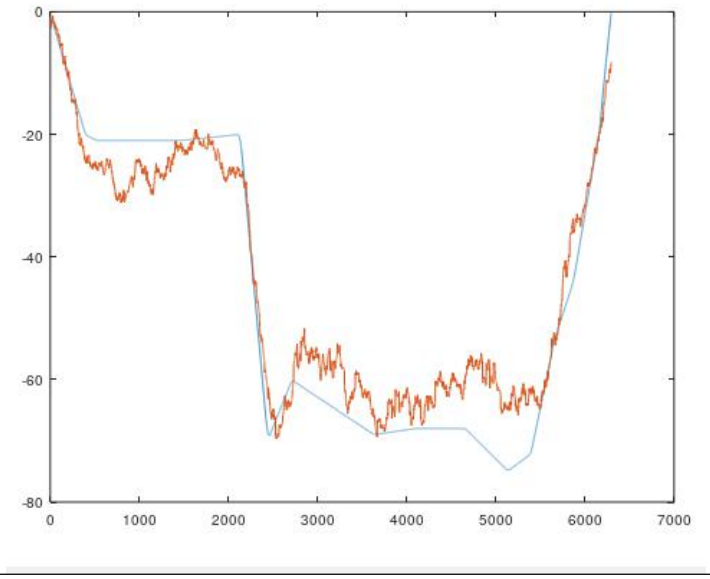

```
iter=40;
#setting Input for inertial sensor and positioning
AIMU=0.1;
sigmaIMU=20;
ADVL=0.1;
sigmaDVL=2;
AUSBL=40;
sigmaUSBL=0.1;
```

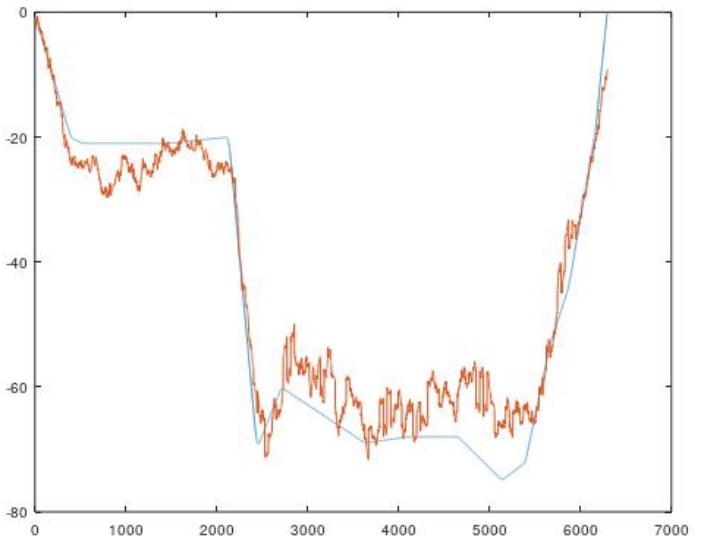
Z axis

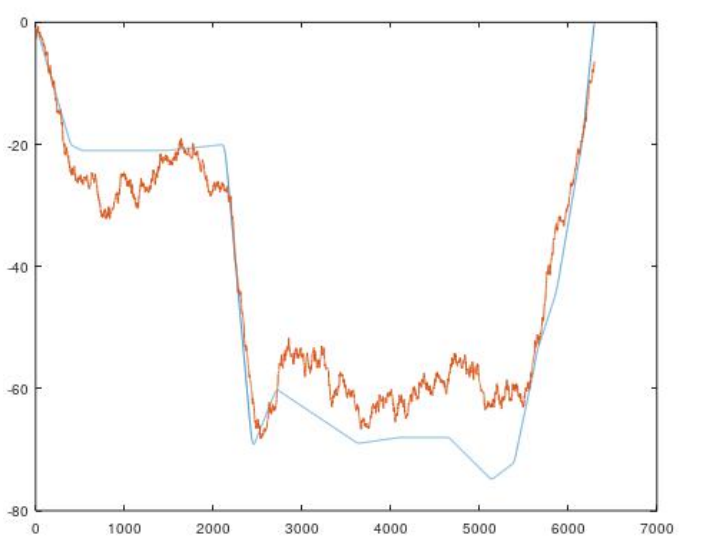
xUSBL = G1D(range,xCusbl(iUsbl,4)-xCusbl(iUsbl-1,4),sigmaUSBL,AUSBL);
xDVL = G1D(range,localTrajvar,30,20);
 	
tUsbl=xCusbl(iUsbl+1,1);
iUsbl=iUsbl+1;
iter = 25;
 	
Zaxis
	

xUSBL = G1D(range,xCusbl(iUsbl,2)-xCusbl(iUsbl-1,2),sigmaUSBL,AUSBL);
xDVL = G1D(range,localTrajvar,30,20);
 	
tUsbl=xCusbl(iUsbl+1,1);
 	
Abrupt error overcome but residual error came
	

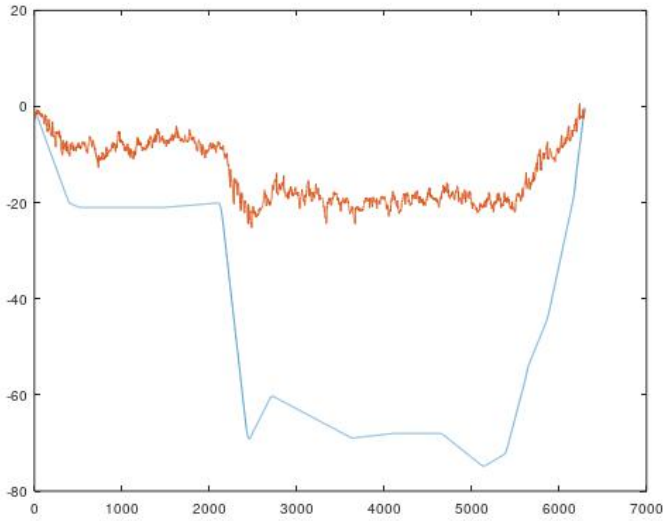
Y axis With above configurations	

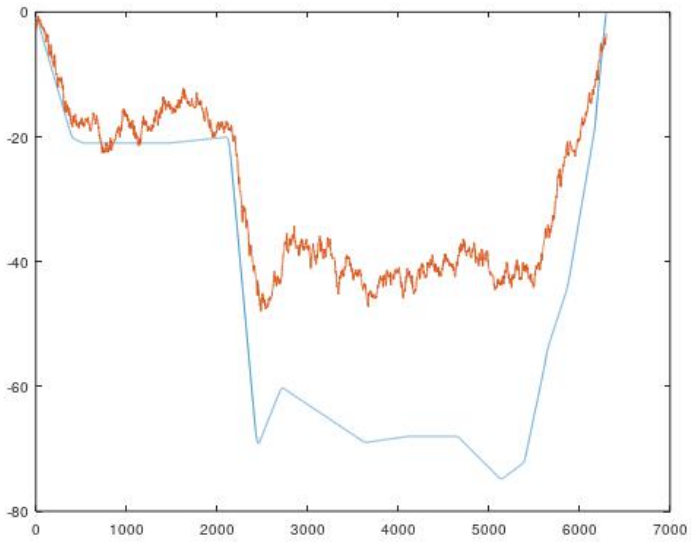
Same conf z axis 25 iterations	

if tUsbl<=t && iUsbl<158
xUSBL = G1D(range,xCusbl(iUsbl,4)-xCusbl(iUsbl-1,4),sigmaUSBL,AUSBL);
xDVL = G1D(range,localTrajvar,30,10);
 	
tUsbl=xCusbl(iUsbl+1,1);
iUsbl=iUsbl+1;
 	
Oscillations by smaller feedback value
	

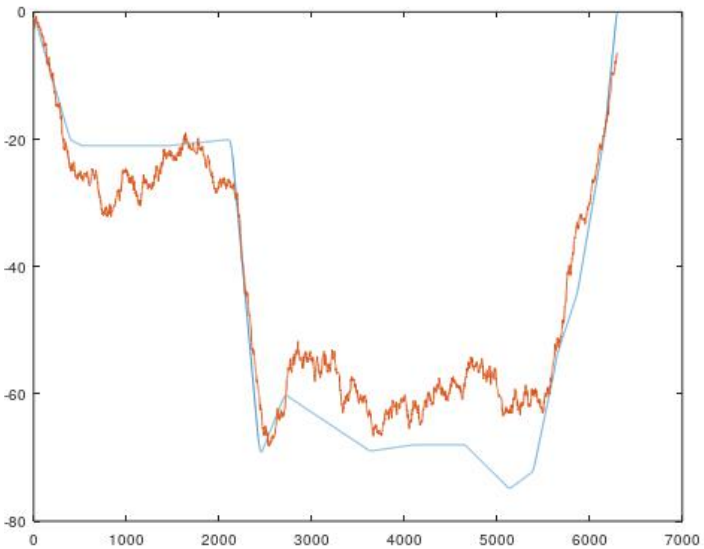
if tUsbl<=t && iUsbl<158
xUSBL = G1D(range,xCusbl(iUsbl,4)-xCusbl(iUsbl-1,4),sigmaUSBL,AUSBL);
xDVL = G1D(range,localTrajvar,30,40);
 	
tUsbl=xCusbl(iUsbl+1,1);
iUsbl=iUsbl+1;
 	
Now a bit smooth
	

```
if tUsbl<=t && iUsbl<158
    xUSBL = G1D(range,xCusbl(iUsbl,4)-xCusbl(iUsbl-1,4),sigmaUSBL,AUSBL);
    xDVL = G1D(range,localTrajvar,50,40);

    tUsbl=xCusbl(iUsbl+1,1);
    iUsbl=iUsbl+1;
```

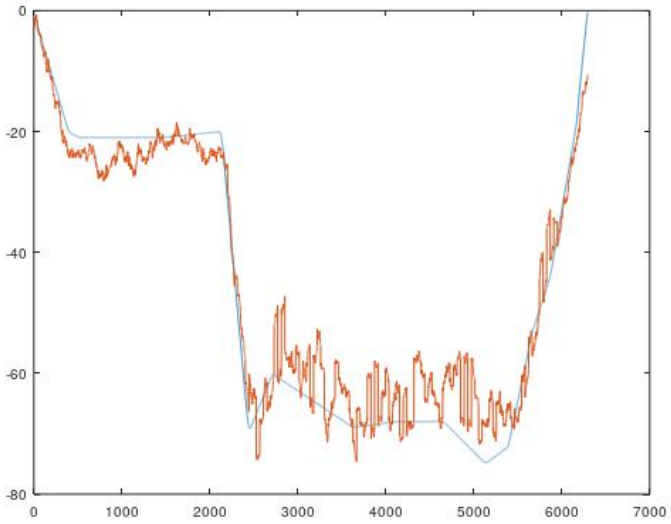


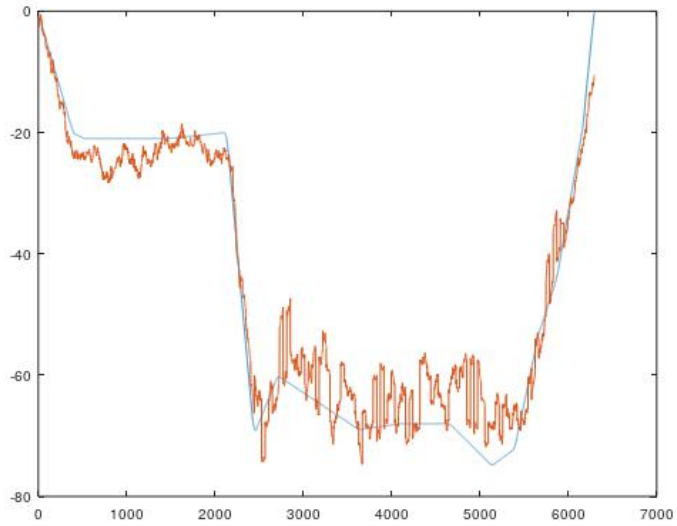



```
    xUSBL = G1D(range,xCusbl(iUsbl,4)-xCusbl(iUsbl-1,4),sigmaUSBL,AUSBL);
    xDVL = G1D(range,localTrajvar,30,40);

    tUsbl=xCusbl(iUsbl+1,1);
    iUsbl=iUsbl+1;


iter=25;
#setting Input for inertial sensor and positioning
AIMU=0.1;
sigmaIMU=20;
ADVL=0.1;
sigmaDVL=2;
AUSBL=40;
```

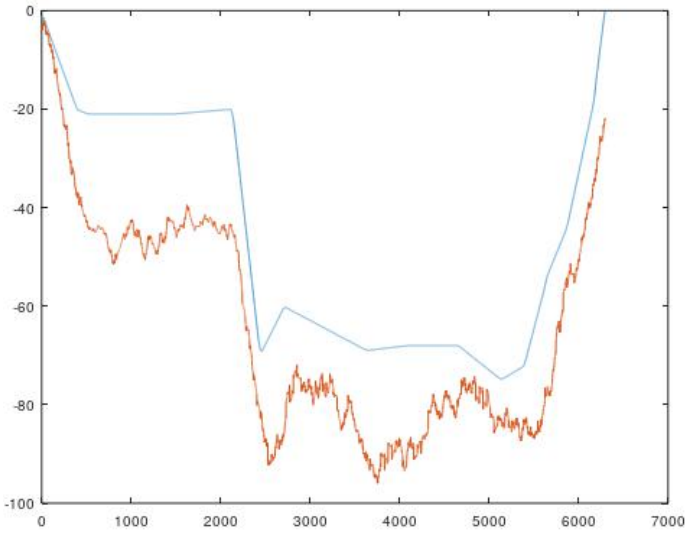


sigmaUSBL=0.1; zaxis	

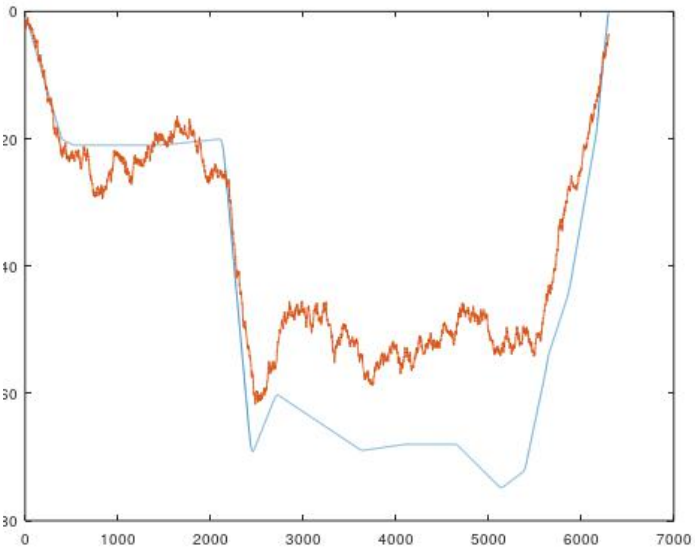
xUSBL = G1D(range,xCusbl(iUsbl,4)-xCusbl(iUsbl-1,4),sigmaUSBL,AUSBL);
xDVL = G1D(range,localTrajvar,30,5);
 	
tUsbl=xCusbl(iUsbl+1,1);
iUsbl=iUsbl+1;
 	
iter=25;
#setting Input for inertial sensor and positioning
AIMU=0.1;
sigmaIMU=20;
ADVL=0.1;
sigmaDVL=2;
AUSBL=40;
sigmaUSBL=0.1;
	

xUSBL = G1D(range,xCusbl(iUsbl,4)-xCusbl(iUsbl-1,4),sigmaUSBL,AUSBL);
xDVL = G1D(range,localTrajvar,30,5);
 	
tUsbl=xCusbl(iUsbl+1,1);
iUsbl=iUsbl+1;
 	
iter=25;
#setting Input for inertial sensor and positioning
AIMU=0.1;
sigmaIMU=30;
ADVL=0.1;
sigmaDVL=2;
AUSBL=40;
sigmaUSBL=0.1;
	

```
  xUSBL = G1D(range,xCusbl(iUsbl,4)-xCusbl(iUsbl-1,4),sigmaUSBL,AUSBL);
   xDVL = G1D(range,localTrajvar,30,40);

   tUsbl=xCusbl(iUsbl+1,1);
   iUsbl=iUsbl+1;

iter=25;
#setting Input for inertial sensor and positioning
AIMU=0.1;
sigmaIMU=30;
ADVL=0.1;
sigmaDVL=5;
AUSBL=40;
sigmaUSBL=0.1;
```




```
   xUSBL = G1D(range,xCusbl(iUsbl,4)-xCusbl(iUsbl-1,4),sigmaUSBL,AUSBL);
   xDVL = G1D(range,localTrajvar,30,40);

   tUsbl=xCusbl(iUsbl+1,1);
   iUsbl=iUsbl+1;


iter=25;
#setting Input for inertial sensor and positioning
AIMU=0.1;
sigmaIMU=30;
ADVL=0.1;
sigmaDVL=0.5;
AUSBL=40;
sigmaUSBL=0.1;
```



```
  xUSBL = G1D(range,xCusbl(iUsbl,4)-xCusbl(iUsbl-1,4),sigmaUSBL,AUSBL);
    xDVL = G1D(range,localTrajvar,30,40);

    tUsbl=xCusbl(iUsbl+1,1);
    iUsbl=iUsbl+1;


iter=25;
#setting Input for inertial sensor and positioning
AIMU=0.1;
sigmaIMU=30;
ADVL=0.1;
sigmaDVL=3;
AUSBL=40;
sigmaUSBL=0.1;
```