analyze calman filter output data

The "ExtendedKF" function has three options "fusion","onlyRadar","onlyLidar". So there is there type of log file. This ipynb(python)file is to readin these file, and output it in visulization.

In [2]:

```
#fusion data variables
fusion_x0,fusion_x1,fusion_x2,fusion_x3 = [],[],[],[]
fusion_px,fusion_py = [],[]
fusion_gt0,fusion_gt1,fusion_gt2,fusion_gt3 = [],[],[],[]
fusion_RMSE_x,fusion_RMSE_y,fusion_RMSE_vx,fusion_RMSE_vy = [],[],[],[]

#lidar data
lidar_x0,lidar_x1,lidar_x2,lidar_x3 = [],[],[],[]
lidar_px,lidar_py = [],[]
lidar_gt0,lidar_gt1,lidar_gt2,lidar_gt3 = [],[],[],[]
lidar_RMSE_x,lidar_RMSE_y,lidar_RMSE_vx,lidar_RMSE_vy = [],[],[],[]

#radar data
radar_x0,radar_x1,radar_x2,radar_x3 = [],[],[],[]
radar_px,radar_py = [],[]
radar_gt0,radar_gt1,radar_gt2,radar_gt3 = [],[],[],[]
radar_RMSE_x,radar_RMSE_y,radar_RMSE_vx,radar_RMSE_vy = [],[],[],[]
```

In [3]:

```
import re

# read EKF datalog, output the data to lists
def read_EKF_datalog(logname):
    biglist = []
    for line in open(logname):
        numberlist = []
        for myword in re.findall(r"[-]?\d+\.?\d*",line):
            numberlist.append(float(myword))
        biglist.append(numberlist)

    x0 = []
    x1 = []
    x2 = []
    x3 = []
    px = []
    py = []
    gt0 = []
    gt1 = []
    gt2 = []
    gt3 = []
    RMSE_x = []
    RMSE_y = []
    RMSE_vx = []
    RMSE_vy = []

    j = 0
    for mylist in biglist:
        j = j + 1
        #print(j)
        if j > 2 :
            x0.append(mylist[0])
            x1.append(mylist[1])
            x2.append(mylist[2])
            x3.append(mylist[3])
            px.append(mylist[4])
            py.append(mylist[5])
            gt0.append(mylist[6])
            gt1.append(mylist[7])
            gt2.append(mylist[8])
            gt3.append(mylist[9])
            RMSE_x.append(mylist[10])
            RMSE_y.append(mylist[11])
            RMSE_vx.append(mylist[12])
            RMSE_vy.append(mylist[13])

    print("data length is: ",len(x0))

    return x0,x1,x2,x3,px,py,gt0,gt1,gt2,gt3,RMSE_x,RMSE_y,RMSE_vx,RMSE_vy
```

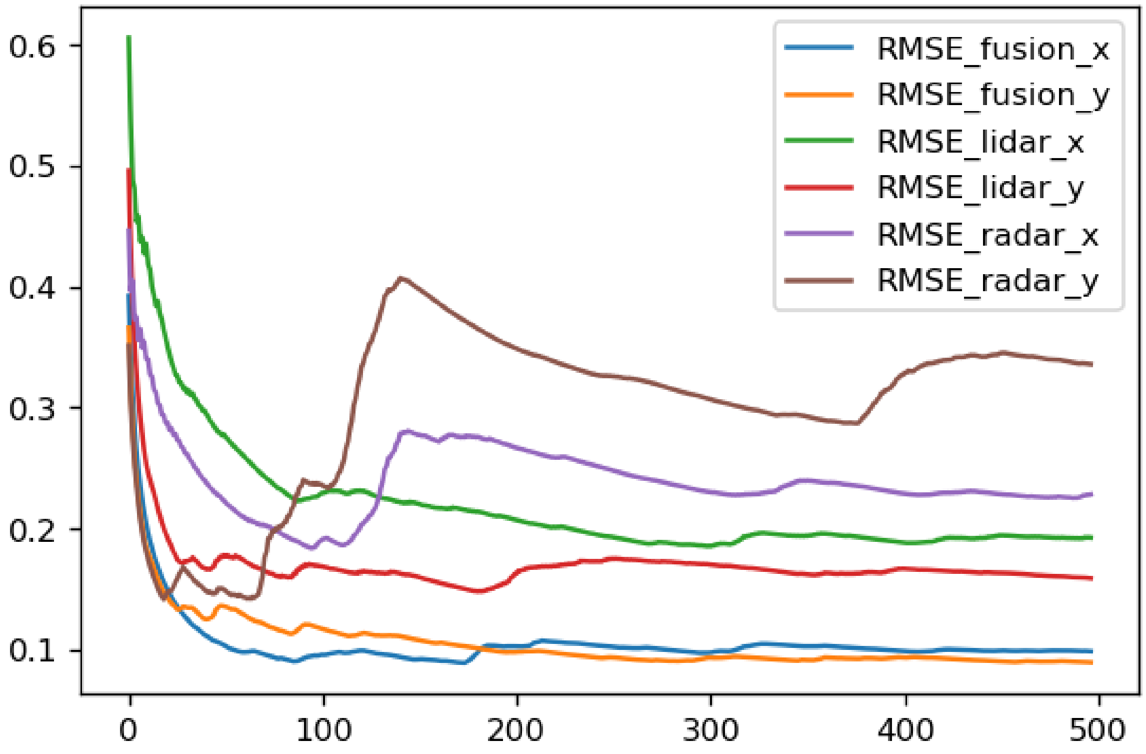
In [5]:

```
# read fusion data to list
fusion_x0,fusion_x1,fusion_x2,fusion_x3,fusion_px,fusion_py,\
fusion_gt0,fusion_gt1,fusion_gt2,fusion_gt3,\
fusion_RMSE_x,fusion_RMSE_y,fusion_RMSE_vx,fusion_RMSE_vy\
= read_EKF_datalog(logname = "log_fusion.txt")

# read lidar data to list
lidar_x0,lidar_x1,lidar_x2,lidar_x3,lidar_px,lidar_py,\
lidar_gt0,lidar_gt1,lidar_gt2,lidar_gt3,\
lidar_RMSE_x,lidar_RMSE_y,lidar_RMSE_vx,lidar_RMSE_vy\
= read_EKF_datalog(logname = "log_onlyLidar.txt")

# read radar data to list
radar_x0,radar_x1,radar_x2,radar_x3,radar_px,radar_py,\
radar_gt0,radar_gt1,radar_gt2,radar_gt3,\
radar_RMSE_x,radar_RMSE_y,radar_RMSE_vx,radar_RMSE_vy\
= read_EKF_datalog(logname = "log_onlyRadar.txt")

x = []
for i in range(0, len(fusion_x0)):
    x.append(i)
```

```
data length is:  497
data length is:  497
data length is:  497
```

In [17]:

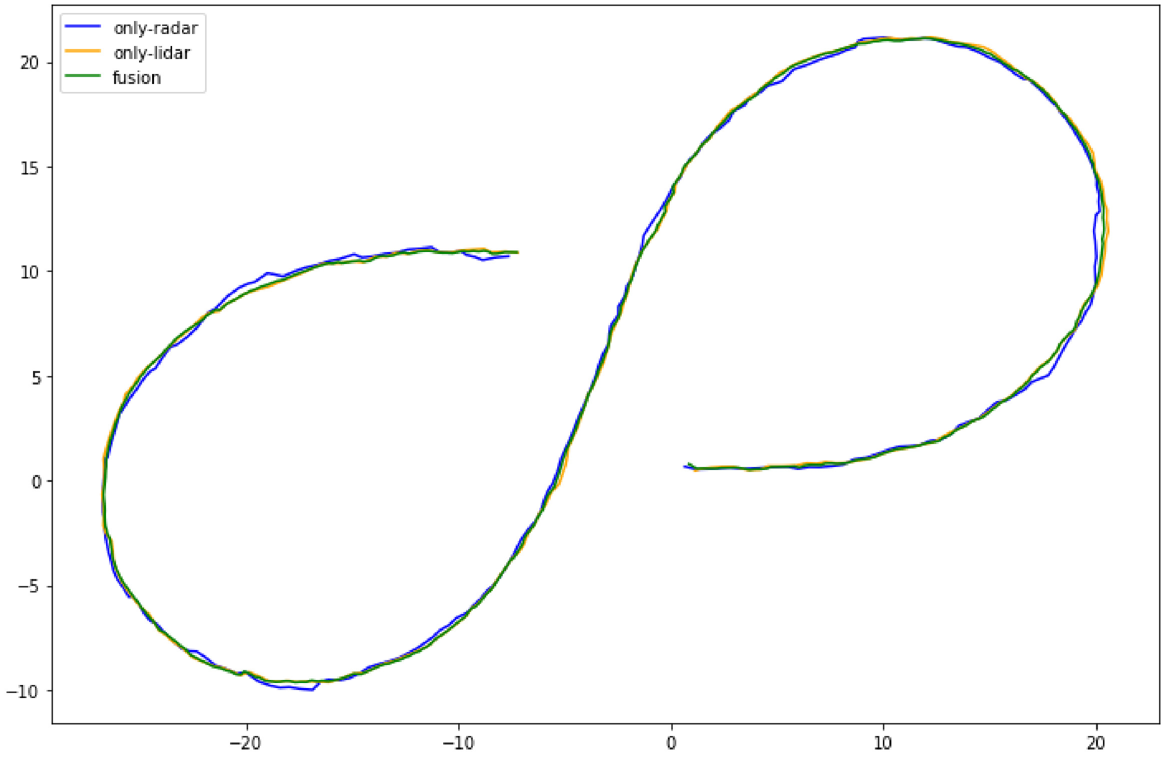
```
import numpy as np
import pylab as pl

pl.figure(figsize=(6,4), dpi=120)
pl.plot(x,fusion_RMSE_x,label='RMSE_fusion_x')# use pylab to plot x and y
pl.plot(x,fusion_RMSE_y,label='RMSE_fusion_y')# use pylab to plot x and y
pl.plot(x,lidar_RMSE_x,label='RMSE_lidar_x')# use pylab to plot x and y
pl.plot(x,lidar_RMSE_y,label='RMSE_lidar_y')# use pylab to plot x and y
pl.plot(x,radar_RMSE_x,label='RMSE_radar_x')# use pylab to plot x and y
pl.plot(x,radar_RMSE_y,label='RMSE_radar_y')# use pylab to plot x and y
pl.legend(loc='upper right')
pl.show()# show the plot on the screen
```



from the RMSE diagram, we can see that, the RMSE is almost fusion < lidar < radar

In [24]:

```
pl.figure(figsize=(12,8))
pl.plot(radar_x0, radar_x1,color="blue",label='only-radar')# use pylab to plot x an
pl.plot(lidar_x0, lidar_x1,color="orange",label='only-lidar')# use pylab to plot x
pl.plot(fusion_x0, fusion_x1,color="green",label='fusion')
pl.legend(loc='upper left')
pl.show()# show the plot on the screen
```



from the trace diagram, we also could see that , the fusion result is better than only-radar or only-lidar