Xiangyu Ding

ECE2595 Radar Signal Processing

Murat Akcakaya

3/22/2018

Project2 in Radar Signal Processing

In the second project of radar signal processing, the main goal is to use MATLAB to simulate the target motion in a plane with constant course and speed, and then it completes a coordinated turn. Besides, implementing a Kalman filter and the IMM estimator to track the target motion.

I solve them step-by-step, each problem may share the same workspace. Please do not skip the former problem when running and analyzing the next problem.

The project runs in the following environment, different working environment may have different customized results. Besides, the scenario is a random process, so the result probably will be different from my report when running my function again. For checking my result, I attached my workspace in my files.

System: Windows 10 Home basic edition 64 bit, Intel Core i7-6700HQ 2.60Ghz CPU, NVidia GTX980M GPU, 16.0 GB RAM

Software: MATLAB R2016a(64bit)

1. Target motion simulation

For problem 1, I use my MATLAB script scenario.m to simulate the target motion with T=1s, k=40 and the initial condition $x(0)$=[2000m 0m/s 1000m−15m/s]T. The command code is

>>[Zt,Zm] = scenario(1,40,[2000,0,1000,-15]')

Then I plot the target motion. The total course looks like an oval but the noise is so trivial that it is hard to recognize the difference between measured location and true location, but I still plot a total motion course in my project2 and attach it in image files. Therefore, I decide to split them into 4 sample parts. The command code of the first part is

>> plot(Zm(1,1:40),Zm(2,1:40),'b*--',Zt(1,1:40),Zt(2,1:40),'ko-') % changing the column number when plot other motion parts

xlabel('X-Range[m]')

ylabel('Y-Range[m]')

legend('Measured','True','Location','Northwest')

The remain parts share the same command code to plot them except changing the column number of Zm and Zt(e.g. >>plot(Zm(1,41:80),Zm(2,41:80),'b*--',Zt(1,41:80),Zt(2,41:80),'ko-') to plot the second parts)

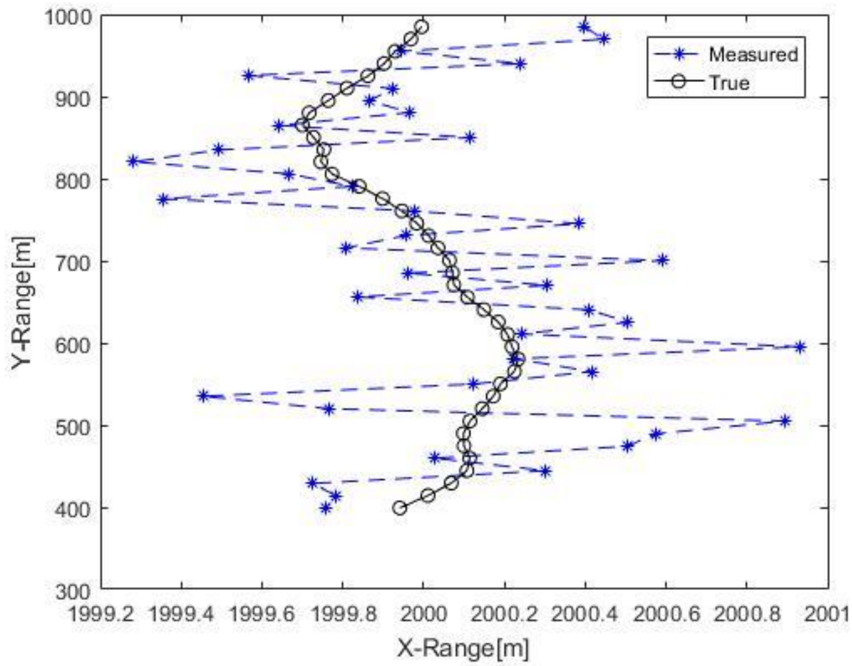

Figure 1.1 Samples 1-40

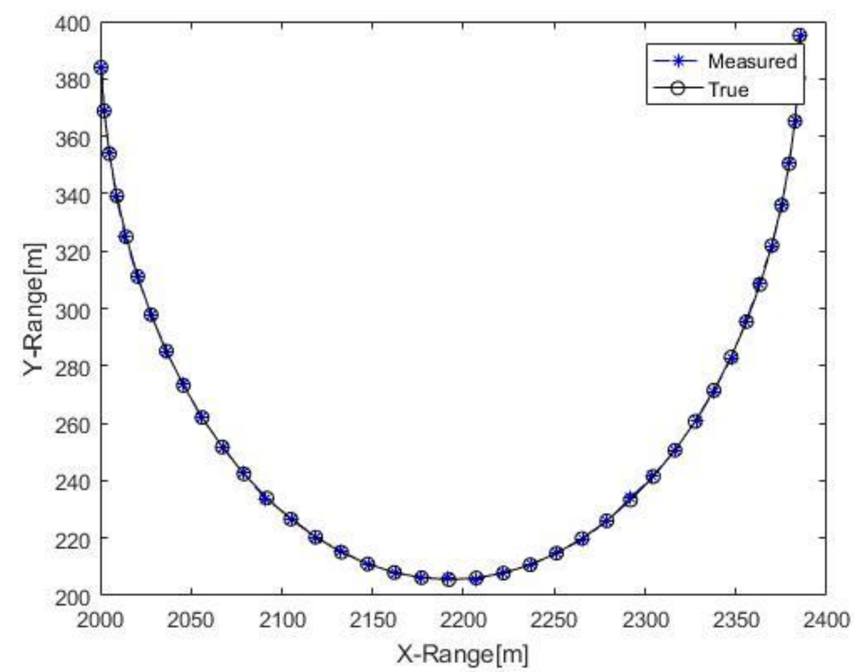

Figure 1.2 Samples 41-80

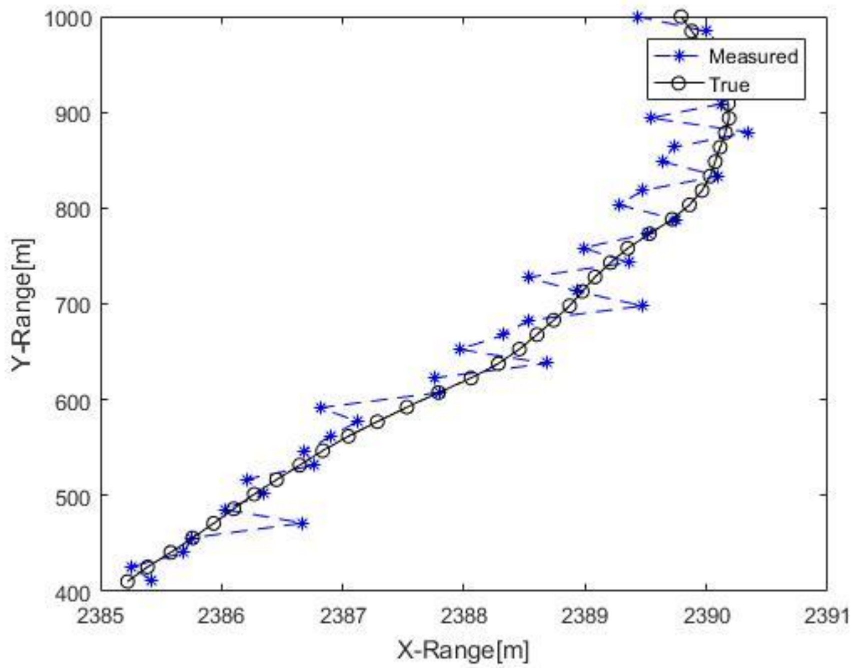

Figure 1.3 Samples 81-120

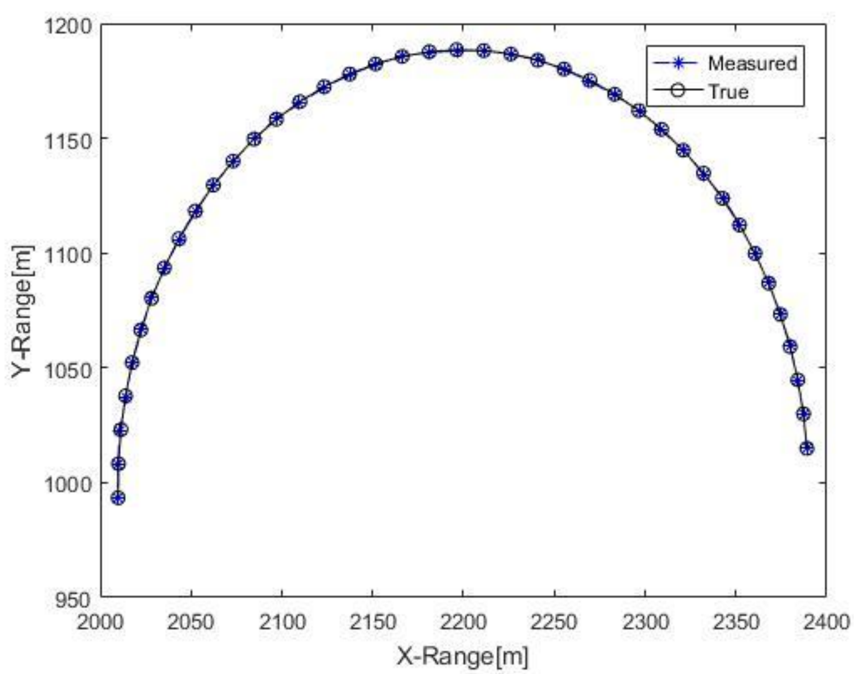

Figure 1.4 Samples121-160

2. Kalman filter implementation

(a) Reference model is constant velocity model

I implement the Kalman filter track the measured location and try to restore the condition like the true condition. My function kalmanfilter.m is attached in my project2 files, it can apply different reference model such as constant velocity model or coordinate turn model, more details and explanations can be found in the m-file.

Based on the target motion in Problem1, I use my MATLAB script to track the measured location Zm with T=1s, k=40, the initial condition $x(0)=[2000m\ 0m/s\ 1000m-15m/s]T$ and reference model in equation(1). The command code is

```
>>Zk1 = kalmanfilter(1,40,[2000,0,1000,-15]',Zm,1)
```

Then I plot the result of tracking, the command code is

```
>> plot(Zm(1,:),Zm(2,:),'b*--',Zt(1,:),Zt(2,:),'k.-',Zk1(1,:),Zk1(3,:),'r+-.')
xlabel('X-Range[m]')
ylabel('Y-Range[m]')
legend('Measured','True','Track','Location','Northeast')
```

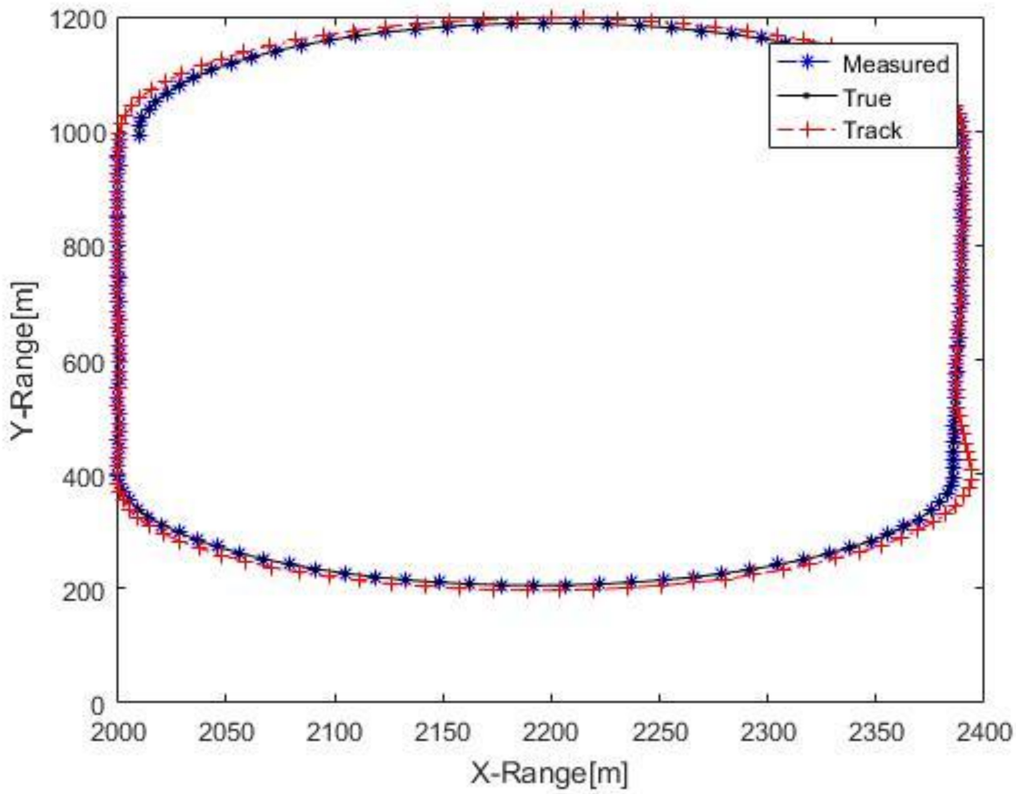

Figure 2.1 Track in 1st model

Then I calculate the estimation error Ze1 and plot it by using this command code

```
>> Ze1 = abs([Zk1(1,:)-Zt(1,:);Zk1(3,:)-Zt(2,:)])

k=(1:1:160)

plot(k,Ze1)

xlabel('Time[t]')

ylabel('Error[m]')

legend('X-error','Y-error','Location','Northeast')
```

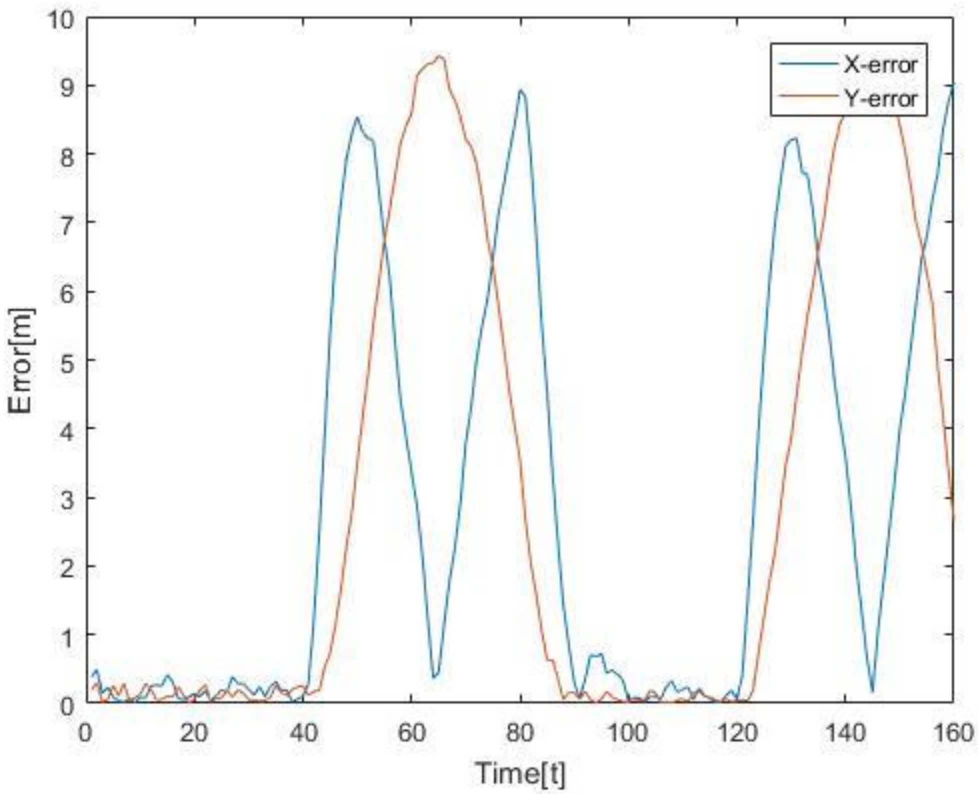

Figure 2.2 Estimation error in 1st model

The result is attached into my project2 image files.

When Kalman filter tracks the constant velocity course, the estimated error is trivial, but when it tracks the coordinate turn course, the estimated error becomes significant.

(b) Reference model is coordinate turn model

It is easy to change the reference model from equation(1) to equation(2) when using my Kalman filter function. The command code is

```
>>Zk2 = kalmanfilter(1,40,[2000,0,1000,-15]',Zm,2)
```

Then I plot them using command code

```
>> plot(Zm(1,:),Zm(2,:),'b*--',Zt(1,:),Zt(2,:),'k.-',Zk2(1,:),Zk2(3,:),'m+-.')
xlabel('X-Range[m]')
```

ylabel('Y-Range[m]')

legend('Measured','True','Track','Location','Northeast')

I attach the result into my project2 image files.

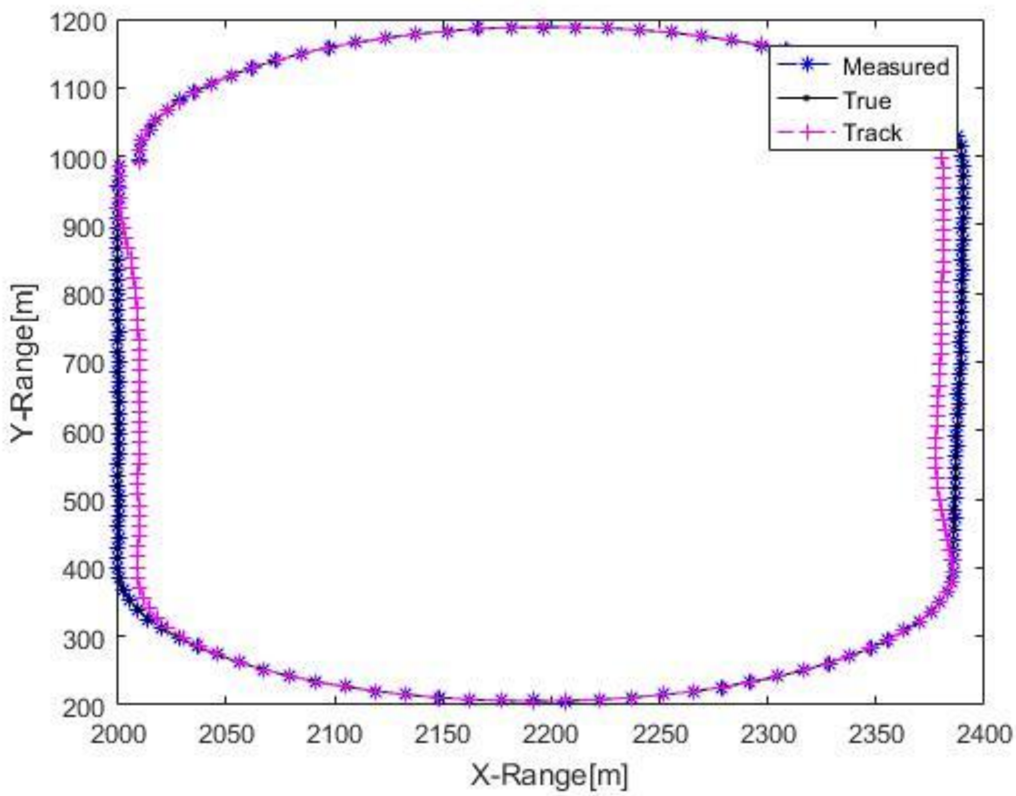

Figure 2.3 Track in 2nd model

As same as before, I calculate the error and plot it, the command code is

```
>> Ze2 = abs([Zk2(1,:)-Zt(1,:);Zk2(3,:)-Zt(2,:)])
k=(1:1:160)
plot(k,Ze2)
xlabel('Time[t]')
ylabel('Error[m]')
legend('X-error','Y-error','Location','Northeast')
```

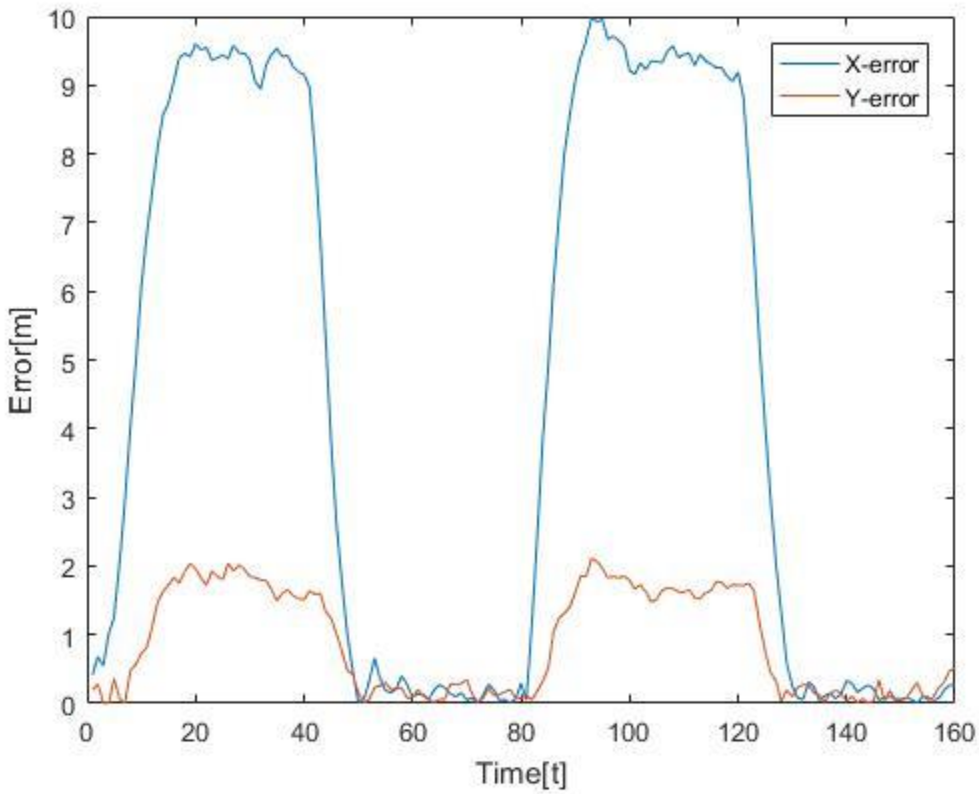

Figure 2.4 Estimation error in 2^{nd} model

In opposite, the 2^{nd} Kalman filter can track the coordinate turn course accurate but the estimated error of constant velocity course is bigger.

3. Interacting Multiple Model Estimation

I implement the IMM estimator in MATLAB script. The function name is IMMestimator.m, this function is able to output a more accurate estimated condition. The more details are attached in my project2 files. The command code of this processing is

```
>>[X,U] = IMMestimator(1,40,[2000,0,1000,-15]',Zm)
```

Then I plot them using command code

```
>> plot(Zm(1,:),Zm(2,:),'b*--',Zt(1,:),Zt(2,:),'k.-',X(1,:),X(3,:),'g+-.')
```

xlabel('X-Range[m]')

ylabel('Y-Range[m]')

legend('Measured','True','Track','Location','Northeast')

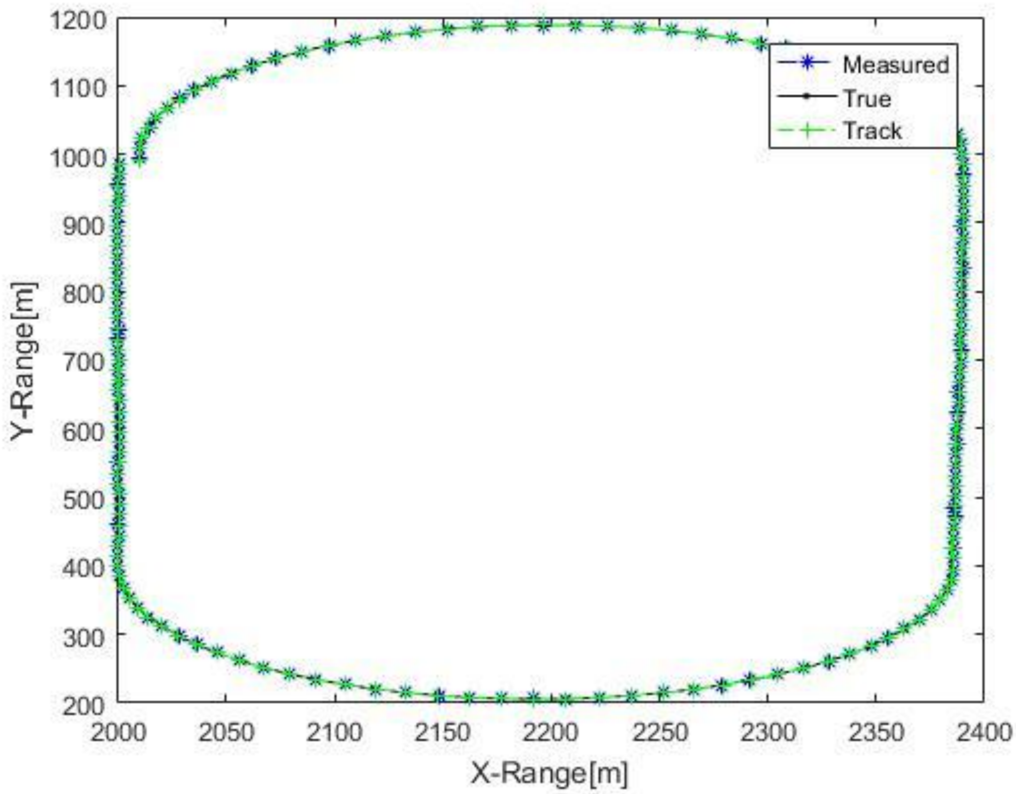

Figure 3.1 Track in IMM estimator

In conclusion, the tracking result of IMM estimator is very accurate to the true state of target motion, with combination of two models we assumed.

Then calculating the error and plot it by using this command code

>>Xe = abs([X(1,:)-Zt(1,:);X(3,:)-Zt(2,:)])

k=(1:1:160)

plot(k,Xe)

xlabel('Time[t]')

ylabel('Error[m]')

legend('X-error','Y-error','Location','Northeast')

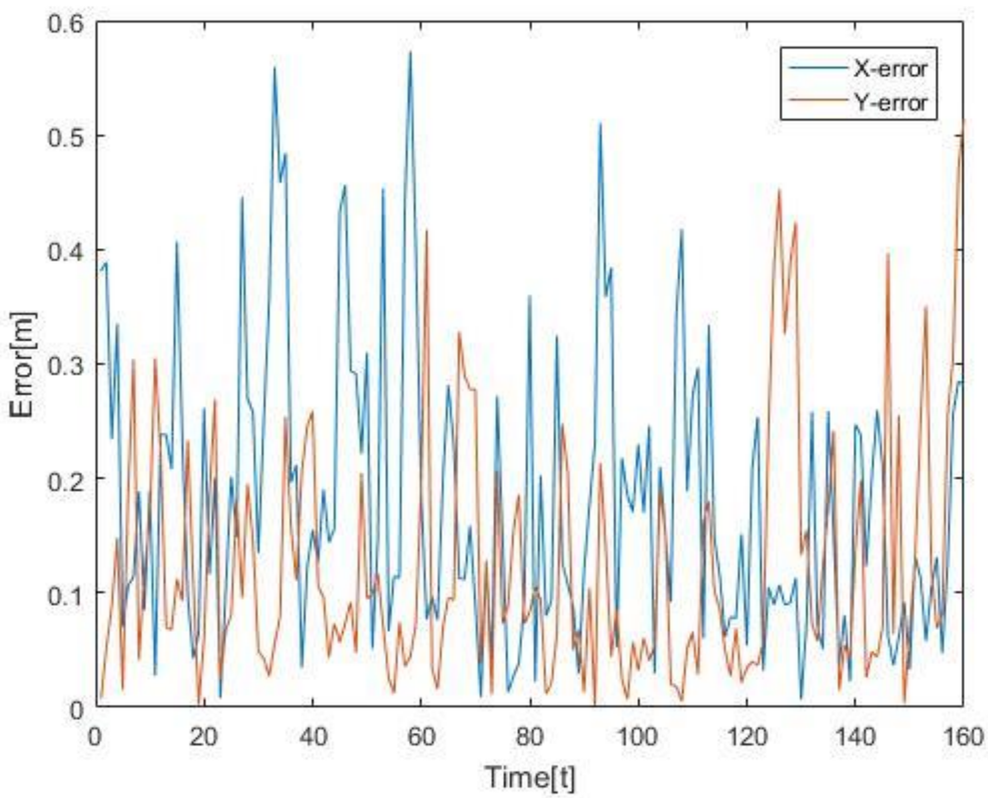

Figure 3.2 Estimation error in IMM

The error of IMM is lower than 0.6m during the total tracking processing. Therefore, it is very useful in this specific scenario.

For the special requirement in project2, I plot the mode probability, the command code is

>>plot(k,U)

xlabel('Time[t]')

ylabel('Probability')

legend('Pcv','Pct','Location','Northeast')

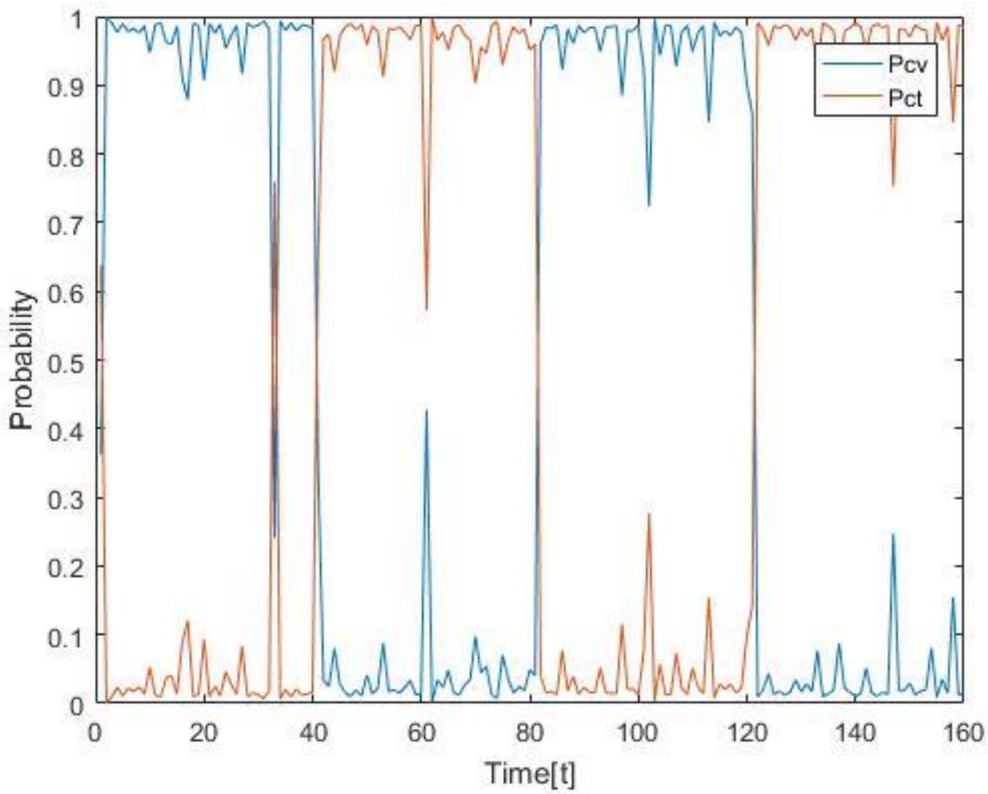

Figure 3.3 Mode probability in IMM

The result is suit to the scenario that I implement in problem1. The IMM estimator works well in this project, it's a useful anticipating model in real target tracking.

References

[1] Radar Tracking with an Interacting Multiple Model and Probabilistic Data Association Filter for Civil Aviation Applications, Shau-Shiun Jan * and Yu-Chun Kao

[2] Linear estimation in dynamic systems – The Kalman filter

[3] Webpage: https://www.youtube.com/watch?v=2-lu3GNbXM8&t=847s