****

**信息融合课程报告**

——多传感器-多目标跟踪算法

|  |  |
| --- | --- |
| **学 院** | 自动化学院 |
| **专 业** | 控制科学与工程 |
| **学 号** | 191060027 |
| **学生姓名** | 杨豪杰 |
| **指导教师** | 刘伟峰 |

2020年6月20日

### 信息融合课程报告

——多传感器-多目标跟踪算法

**一、实验场景与参数设置**

### 1. 多目标CV运动

监测区域为，检测概率为0.98，杂波密度为，既平均40个杂波点，状态协方差阵，，目标的初始分布为其中，,目标运动和量测方程分别为：



其中, ，

2. 最近邻算法：

### 二、报告要求

1. 仿真杂波条件下3个目标的CV运动轨迹（仿真总步数K=50s）；

2. 采用Kalman滤波算法，分别完成传感器1，2对多目标的分别跟踪过程。

1. 采用最紧邻算法进行数据关联。
2. 通过100次蒙特卡罗仿真实验，对比分析传感器1，2的RMSE跟踪误差。
3. 采用集中式多传感器，进行多目标跟踪估计，给出跟踪结果图，并进行RMSE误差分析

(-1000m,1000m) (1000m,1000m)

目标1

目标2

目标3

(-1000m,-1000m) (1000m,-1000m)

1. 计算多传感器跟踪系统的CR下界。

### 结果分析

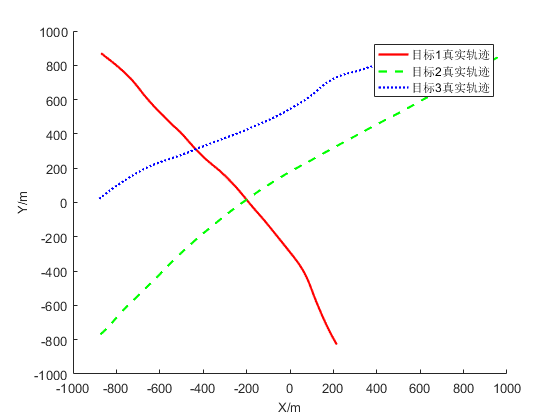
**要求1**：设置三个目标的初始位置为，；

目标运动方程为





其中,设置单位采样周期T为1，运动时间为50，滤波条件下3个目标的CV图像如图一所示：



图一 多目标真实轨迹

**要求2(1)**:使用传感器一对多目标进行跟踪，观测方程为：



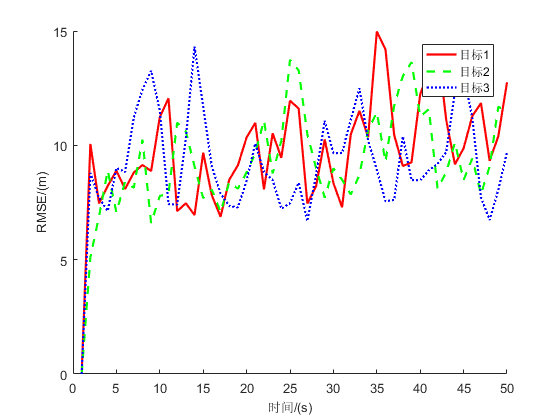
其中，，且检测概率为0.98，杂波密度为，即平均40个杂波点。数据关联采用最近邻算法：



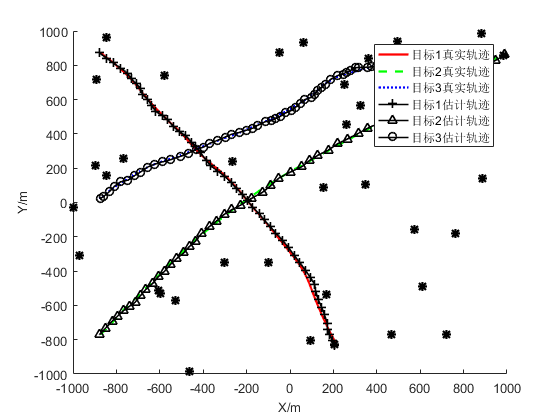
采用Kalman滤波算法进行多目标跟踪：



进行100次蒙特卡洛实验，得到传感一的多目标跟踪结果，以及RMSE跟踪误差。



图二 传感器一的RMSE误差



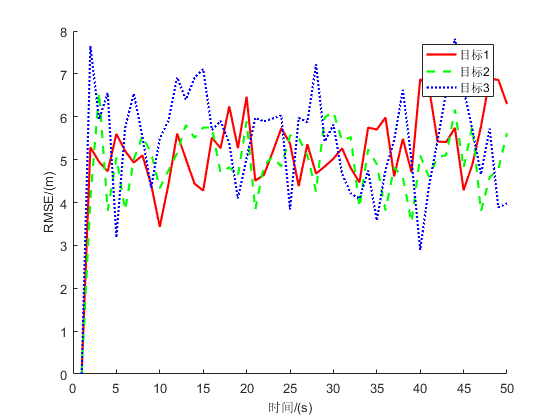
图三 传感一观测下的某次跟踪结果

**要求2(2)**:使用传感器二对多目标进行跟踪，观测方程为：

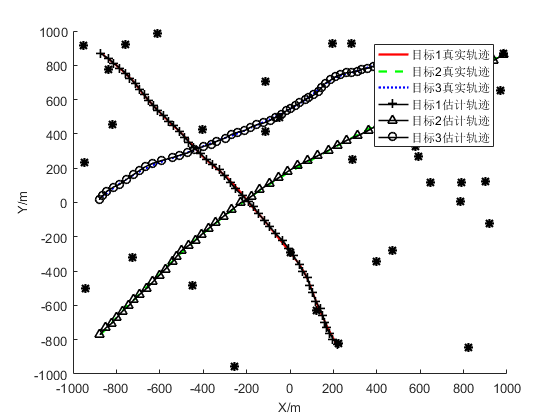


其中，其余同要求2(1),进行100次蒙特卡洛实验，得到传感器二的多目标跟踪结果，以及RMSE跟踪误差。

通过图一和图三的对比，当多目标的状态模型参数一致时，同一传感器的跟踪精度差别不大，对于不同传感器，发现观测噪声小的单一传感器轨迹跟踪的RMSE误差更小。



图三 传感器二的RMSE误差



图四 传感二观测下的某次跟踪结果

**要求3**：采用集中式多传感器，进行多目标跟踪。多目标的运动方程和多传感器的观测方程为：



其中

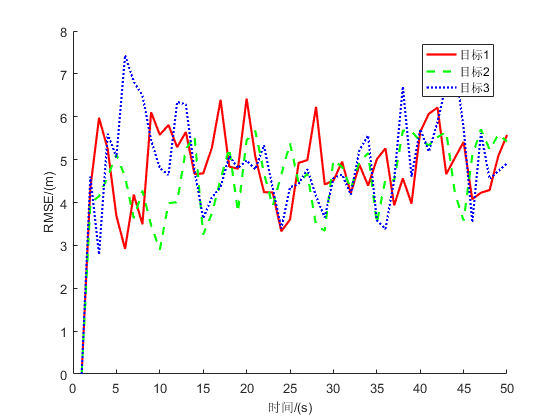
。



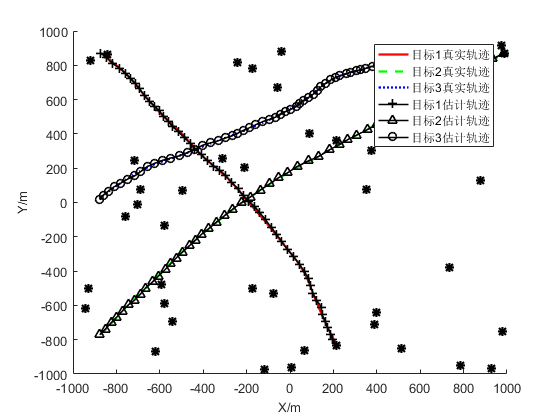
在两个传感器相互独立的情况下



利用两个传感器的量测，每个时刻都会得到两组多目标的观测值。并对两组值赋予不同的权重，传感器一权重为0.2，传感器二权重为0.8，进行观测信息融合。通过最近邻算法进行数据关联，从每时刻观测融合后的观测值中提取出当前时刻相对合理的观测值进行Kalman滤波，对多目标进行 踪得到多传感器跟踪结果和RMSE跟踪误差。



图五 多传感器的RMSE误差



图六 多传感器观测下的某次跟踪结果

对比图一、图三、图五的RMSE跟踪误差，我们可以发现多传感器跟踪算法的RMSE误差相对最小。从算法特点上我们可以发现，多传感器系统相对单传感器系统获取到更多的观测信息，并且使观测噪声得到有效减小。我们通过合理数据关联算法和融合决策，可以实现估计误差的有效减小。

**要求4**：计算多传感器跟踪系统的CR下界。多目标的运动方程和多传感器的观测方程为：



其中

。



在两个传感器相互独立的情况下



多传感器模型为线性高斯模型，其似然函数为





基于上述表述，该系统模型的CRLB为



对于X轴和Y轴的位置的CRLB为，当时，多传感器观测噪声协方差矩阵最小为，按照要求3中取50，

多传感器系统的CRLB为0.4。

### **附录**

要求1代码：

clc;clear ;close all;

i = 0;

while(i == 0)

n=50;%仿真时间

range=[-1000 1000;-1000 1000];

A=[1 1 0 0;0 1 0 0;0 0 1 1;0 0 0 1];

B=[0.5 0;1 0;0 0.5;0 1];

C=[1 0 0 0;0 0 1 0];

B0=diag([100 25 100 25]);

Q=[4 0;0 4];%噪声过程的协方差阵

q\_std=sqrt(Q);%噪声过程的标准差阵

R=[100 0;0 100];%量测噪声协方差阵

r\_std=sqrt(R);%量测噪声标准差阵

%-----------定义矩阵含义-------------

X1=zeros(4,n);%目标1的真实状态

X2=zeros(4,n);%目标2的真实状态

X3=zeros(4,n);%目标3的真实状态

Xp=zeros(4,3\*n);%目标状态的提前一步预测

Xe=zeros(4,3\*n);%目标某时刻的估计状态

Z1=zeros(2,n);%目标1真实量测

Z2=zeros(2,n);%目标2真实量测

Z3=zeros(2,n);%目标3真实量测

Zp=zeros(2,3\*n);%某时刻目标量测的提前一步预测

Zk=zeros(2,3\*n);%认定为目标量测的集合

Pe=zeros(4,4,3\*n);%滤波误差的协方差阵

Pp=zeros(4,4,3\*n);%预测误差的协方差阵

K=zeros(4,2,3\*n);%滤波增益

%------------初始化3个目标的状态----

X1(:,1)=[-900;30;900;-30];

X2(:,1)=[-900;25;-800;30];

X3(:,1)=[-900;20;0;20];

for k=1:n-1

X1(:,k+1)=A\*X1(:,k)+B\*(q\_std\*randn(2,1));

X2(:,k+1)=A\*X2(:,k)+B\*(q\_std\*randn(2,1));

X3(:,k+1)=A\*X3(:,k)+B\*(q\_std\*randn(2,1));

end

if X1(1,n) < 1000 && X1(3,n) < 1000 && X2(1,n) < 1000 && X2(3,n) < 1000 && X3(1,n) < 1000 && X3(3,n) < 1000 ...

&& X1(1,n) >-1000 && X1(3,n) >-1000 && X2(1,n) >-1000 && X2(3,n) >-1000 && X3(1,n) >-1000 && X3(3,n) >-1000

i = 1;

else

i = 0;

end

end

save('real\_6','X1','X2','X3')

%% 单独真实轨迹

figure;hold on;

plot(X1(1,2:k+1),X1(3,2:k+1),'-r','LineWidth',1.6);

plot(X2(1,2:k+1),X2(3,2:k+1),'--g','LineWidth',1.6);

plot(X3(1,2:k+1),X3(3,2:k+1),':b','LineWidth',1.6);

xlabel('X/m','fontsize',10);ylabel('Y/m','fontsize',10);

legend('目标1真实轨迹','目标2真实轨迹','目标3真实轨迹');

hold off;

要求2、3、4代码（不同要求只需改变观测噪声协方差阵即可）：

clc;clear ;close all;

n=50;%仿真时间

L=10;%蒙特卡洛仿真次数

range=[-1000 1000;-1000 1000];

lambda=40;

PD=1;

num\_target=3;%一共3个目标

A=[1 1 0 0;0 1 0 0;0 0 1 1;0 0 0 1];

B=[0.5 0;1 0;0 0.5;0 1];

C=[1 0 0 0;0 0 1 0];

B0=diag([100 25 100 25]);

Q=[4 0;0 4];%噪声过程的协方差阵

q\_std=sqrt(Q);%噪声过程的标准差阵

R=[100 0;0 100];%量测噪声协方差阵

r\_std=sqrt(R);%量测噪声标准差阵

Sz1=inv(R);

Sz2=inv(R);

Sz3=inv(R);

%-----------定义矩阵含义-------------

X1=zeros(4,n);%目标1的真实状态

X2=zeros(4,n);%目标2的真实状态

X3=zeros(4,n);%目标3的真实状态

Xp=zeros(4,3\*n);%目标状态的提前一步预测

Xe=zeros(4,3\*n);%目标某时刻的估计状态

Z1=zeros(2,n);%目标1真实量测

Z2=zeros(2,n);%目标2真实量测

Z3=zeros(2,n);%目标3真实量测

Zp=zeros(2,3\*n);%某时刻目标量测的提前一步预测

Zk=zeros(2,3\*n);%认定为目标量测的集合

Pe=zeros(4,4,3\*n);%滤波误差的协方差阵

Pp=zeros(4,4,3\*n);%预测误差的协方差阵

K=zeros(4,2,3\*n);%滤波增益

RMSE=zeros(1,3\*n);

error=zeros(1,3\*n);

%------------初始化3个目标的状态----

X1(:,1)=[-900;30;900;-30]+B0\*randn(4,1);

X2(:,1)=[-900;25;-800;30]+B0\*randn(4,1);

X3(:,1)=[-900;20;0;20]+B0\*randn(4,1);

Xe(:,1)=[-900;30;900;-30];

Xe(:,51)=[-900;25;-800;30];

Xe(:,101)=[-900;20;0;20];

Pe(:,:,1)=[100 0 0 0; 0 25 0 0;0 0 100 0;0 0 0 25];

Pe(:,:,51)=[100 0 0 0; 0 25 0 0;0 0 100 0;0 0 0 25];

Pe(:,:,101)=[100 0 0 0; 0 25 0 0;0 0 100 0;0 0 0 25];

load('real\_6.mat');%加载目标轨迹数据

for mont=1:L %蒙特卡洛仿真

%------------Kalman滤波跟踪过程-----

mont

for k=1:n-1

%% %--三个目标的真实航迹和量测------已经保存真实航迹数据数据，

Z1(:,k+1)=C\*X1(:,k+1)+r\_std\*randn(2,1);

Z2(:,k+1)=C\*X2(:,k+1)+r\_std\*randn(2,1);

Z3(:,k+1)=C\*X3(:,k+1)+r\_std\*randn(2,1);

%% ---kalman滤波预测过程

Xp(:,k+1)=A\*Xe(:,k);%求k时刻目标1状态的提前一步预测

Zp(:,k+1)=C\*Xp(:,k+1);%求k时刻目标1量测的提前一步预测

Pp(:,:,k+1)=A\*Pe(:,:,k)\*A'+B\*Q\*B';%求k时刻目标1预测误差的协方差阵

Xp(:,k+51)=A\*Xe(:,k+50);%求k时刻目标2状态的提前一步预测

Zp(:,k+51)=C\*Xp(:,k+51);%求k时刻目标2量测的提前一步预测

Pp(:,:,k+51)=A\*Pe(:,:,k+50)\*A'+B\*Q\*B';%求k时刻目标2预测误差的协方差阵

Xp(:,k+101)=A\*Xe(:,k+100);%求k时刻目标3状态的提前一步预测

Zp(:,k+101)=C\*Xp(:,k+101);%求k时刻目标3量测的提前一步预测

Pp(:,:,k+101)=A\*Pe(:,:,k+100)\*A'+B\*Q\*B';%求k时刻目标3预测误差的协方差阵

%----模拟杂波并获得所有可能的量测点

Nc=poissrnd(lambda); Zc=repmat(range(:,1),[1,Nc])+(range(1,2)-range(1,1))\*rand(2,Nc);

idx=find(rand(num\_target,1)<=PD);

Z\_real=[];

if find(idx==1)

Z\_real=[Z\_real,Z1(:,k+1)];

end

if find(idx==2)

Z\_real=[Z\_real,Z2(:,k+1)];

end

if find(idx==3)

Z\_real=[Z\_real,Z3(:,k+1)];

end

Z\_real=[Z\_real,Zc];

num\_meas=size(Z\_real,2);

D=zeros(1,num\_meas,3\*n);%目标可能的量测与提前一步预测的距离

%------获取最近邻量测---------------

for i=1:num\_meas

D(1,i,k+1)=(Z\_real(:,i)-Zp(:,k+1))'\*Sz1\*(Z\_real(:,i)-Zp(:,k+1));

D(1,i,k+51)=(Z\_real(:,i)-Zp(:,k+51))'\*Sz2\*(Z\_real(:,i)-Zp(:,k+51));

D(1,i,k+101)=(Z\_real(:,i)-Zp(:,k+101))'\*Sz3\*(Z\_real(:,i)-Zp(:,k+101));

end

d1=min(D(1,:,k+1));%求距离中的最小值

x=find(D(1,:,k+1)==d1);%找到最小值的位置 目标1

d2=min(D(1,:,k+51));%目标2

y=find(D(1,:,k+51)==d2);

d3=min(D(1,:,k+101));%目标3

z=find(D(1,:,k+101)==d3);

%------kalman滤波更新过程

Zk(:,k+1)=Z\_real(:,x(1));%获取目标1的新的量测

Sk1=(C\*Pp(:,:,k+1)\*C'+R);

K(:,:,k+1)=Pp(:,:,k+1)\*C'\*inv(Sk1);%k时刻系统的滤波增益

a=Zk(:,k+1)-Zp(:,k+1);%新息

Zk(:,k+51)=Z\_real(:,y(1));%获取目标2的新量测

Sk2=(C\*Pp(:,:,k+51)\*C'+R);

K(:,:,k+51)=Pp(:,:,k+51)\*C'\*inv(Sk2);

b=Zk(:,k+51)-Zp(:,k+51);%新息

Zk(:,k+101)=Z\_real(:,z(1));%获取目标3的新量测

Sk3=(C\*Pp(:,:,k+101)\*C'+R);

K(:,:,k+101)=Pp(:,:,k+101)\*C'\*inv(Sk3);

c=Zk(:,k+101)-Zp(:,k+101);%新息

Xe(:,k+1)=Xp(:,k+1)+K(:,:,k+1)\*a;%目标1的滤波更新值

Pe(:,:,k+1)=Pp(:,:,k+1)-K(:,:,k+1)\*C\*Pp(:,:,k+1);%目标1的滤波误差的协方差

Xe(:,k+51)=Xp(:,k+51)+K(:,:,k+51)\*b;%目标2的滤波更新值

Pe(:,:,k+51)=Pp(:,:,k+51)-K(:,:,k+51)\*C\*Pp(:,:,k+51);%目标2的滤波误差的协方差阵

Xe(:,k+101)=Xp(:,k+101)+K(:,:,k+101)\*c;%目标3的滤波更新值

Pe(:,:,k+101)=Pp(:,:,k+101)-K(:,:,k+101)\*C\*Pp(:,:,k+101);%目标3的滤波误差的协方差阵

%% 跟踪误差计算

error(1,k+1)=sum((Xe([1 3],k+1)-X1([1 3],k+1)).^2);

error(1,k+51)=sum((Xe([1 3],k+51)-X2([1 3],k+1)).^2);

error(1,k+101)=sum((Xe([1 3],k+101)-X3([1 3],k+1)).^2);

%% 绘制实时跟踪图

clf; hold on;%Zk量测 ，Zp提前一步量测的预测，Xe目标状态估计，X1目标真实状态

% plot 真实轨迹

plot(X1(1,2:k+1),X1(3,2:k+1),'-r','LineWidth',1.6);

plot(X2(1,2:k+1),X2(3,2:k+1),'--g','LineWidth',1.6);

plot(X3(1,2:k+1),X3(3,2:k+1),':b','LineWidth',1.6);

%plot 估计轨迹

plot(Xe(1,2:k+1),Xe(3,2:k+1),'k-+','LineWidth',1.2);

plot(Xe(1,52:k+51),Xe(3,52:k+51),'k-^','LineWidth',1.2);

plot(Xe(1,102:k+101),Xe(3,102:k+101),'k-o','LineWidth',1.2);

plot(Z\_real(1,:),Z\_real(2,:),'k\*','LineWidth',1.6);

% axis(equal);axis(limit);

xlabel('X/m','fontsize',10);ylabel('Y/m','fontsize',10);

legend('目标1真实轨迹','目标2真实轨迹','目标3真实轨迹','目标1估计轨迹','目标2估计轨迹','目标3估计轨迹');

hold off;

pause(0.01);

end

%% 蒙特卡洛仿真误差累加

RMSE=RMSE+error/L;

end

%% 更新RMSE

RMSE=sqrt(RMSE);

%% 分别绘制RMSE

figure;hold on;

plot(RMSE(1:n),'-r','LineWidth',1.6);

plot(RMSE(n+1:2\*n),'--g','LineWidth',1.6);

plot(RMSE(2\*n+1:3\*n),':b','LineWidth',1.6);

xlabel('时间/(s)','fontsize',10);ylabel('RMSE/(m)','fontsize',10);

legend('目标1','目标2','目标3');

hold off;

%% 单独真实轨迹

figure;hold on;

plot(X1(1,2:k+1),X1(3,2:k+1),'-r','LineWidth',1.6);

plot(X2(1,2:k+1),X2(3,2:k+1),'--g','LineWidth',1.6);

plot(X3(1,2:k+1),X3(3,2:k+1),':b','LineWidth',1.6);

xlabel('X/m','fontsize',10);ylabel('Y/m','fontsize',10);

legend('目标1真实轨迹','目标2真实轨迹','目标3真实轨迹');

hold off;