



**课 程 设 计 报 告**

|  |  |
| --- | --- |
| 学 院： | 自动化学院 |
| 班 级： | 09011505 |
| 学 号： | 2015302112 |
| 学生姓名： | 沈宝印 |
| 指导教师： | 杨峰 |
| 时 间： | 2018年 7 月 |

课程设计任务书

一、设计内容

利用Hough变换处理量测得到的含杂波的二维坐标，解决多目标航迹起始问题。使用Matlab进行仿真，用离散点绘制原始数据，用连线绘制处理结果。

二、主要技术指标

1、掌握标准Hough变换检测直线的基本原理，以及一些改进的Hough变换算法。

2、用Matlab实现Hough变换算法，并绘制处理结果。

4、阅读Hough变换航迹起始的文献3篇以上。

三、进度要求

2周完成设计任务，撰写设计报告3000字以上，应包含设计过程、 计算结果、 图表等内容。

|  |  |  |  |
| --- | --- | --- | --- |
| 学 生 | 沈宝印 | 指导教师 | 杨峰 |

## 设计内容

航迹起始是航迹处理中的首要问题，在各种航迹处理的情况下都存在，对于多目标航迹处理来说，航迹起始是第一步，是进行航迹跟踪的基础。 由于被探测目标一般都是由远到近的出现在搜索雷达的有效探测范围内的，在航迹起始时，目标一般距离雷达较远，此时雷达分辨力低、测量精度差，加之真假目标的出现无真正的统计规律，因此在搜索雷达数据处理技术中，航迹起始问题是一个难题。 现有的航迹起始算法可分为顺序处理技术和批处理技术两大类。通常，顺序数据处理技术适用于弱杂波环境中的航迹起始，主要包括启发式规则方法和基于逻辑方法；批处理技术则更适用于强杂波环境，主要包括Hough变换等方法。基于逻辑的方法在虚警概率比较低的情况下，起始航迹的效果比较好，但在复杂环境下虚假航迹比较多；Hough变换法则适用于强杂波背景下航迹成直线的环境，但是Hough变换法通常需要多次的扫描才能较好地起始航迹，且计算量大不符合工程应用的需要。

低信噪比、低信杂比下的航迹起始是多目标航迹起始的关键问题。Hough变换具有对局部缺损的不敏感性、对随机噪声的鲁棒性以及适于并行处理、实时应用等特点，特别史和解决多目标航迹起始问题。本文对Hough变换航迹起始算法进行了研究，主要工作如下：

1.概述了主要的航迹起始方法，介绍了Hough变换基本原理、Hough变换的特点，指出了Hough变换在航迹起始中存在的问题。

2.研究与分析了标准Hough变换、修正Hough变换和序列Hough变换三种典型航迹起始算法。通过仿真分析，总结出每种算法的适用环境。

## 设计过程

### 2.1 Hough变换的基本原理

1962年，Paul Hough提出了Hough变换法,它是实现图像边缘检测的一种有效方法。其基本思想是将图像空间中的检测问题转换到参数空间，通过在参数空间里进行简单的累加统计完成检测任务，用大多数边界点满足的某种参数形式来描述图像的区域边界曲线。同时Hough变换也可以被描述为证据积累过程：图像空间中的任意数据点，通过变换函数的作用，在参数空间中，对所有可能经过这一数据点的图形对应的参数进行投票；所有数据点的投票在积累矩阵中进行积累，投票结束后，各积累单元的积累值表示所检测图形的参数为相应积累单元对应参数的概率的大小。因而对于被噪声干扰或间断区域边界的图像，Hough变换具有很好的容错性和鲁棒性。

Hough变换可以识别和检测图像空间中的任意解析曲线，如直线、圆、椭圆、双曲线、抛物线等。对于图像空间中位于同一条曲线上点(x,y)的集合，我们可以用一个参数空间到图像空间的映射来表述：

其中是曲线的特征参数。式(2-1)中的自变量与因变量交换角色，得下式：

这就建立了一个图像空间到参数空间得映射，图像空间中属于同一解析曲线得点经过式(2-1)后，都被映射到参数空间中，且相交于由参数确定的点。

对于图像空间中位于同一条直线上的点(x,y)的集合，(2-1)式可以写成：

这是直线方程的斜率-截距式表达式，其中，是该直线的斜率，是截距。相应的，将与(x,y)交换角色，得到：

可以看到，图像空间中的每一个点(x,y),都映射为参数空间中的一条直线。参数空间中有公共交点的那些直线则代表了图像空间中共线的点的集合。

### 2.2 Hough变换的特点、优点和缺点

Hough变换用于航迹起始具有以下特点：

（1）将量测空间中的检测问题转换到参数空间进行，具有很强的抗干扰能力，对随机噪声具有一定的鲁棒性；

（2）量测中的每一个点都参加“投票”，所以它特别适合并行处理；

（3）一种变换方程只对某一种特定的曲线进行检测，针对性强；

（4）不受空间和曲线形状的影响：广义Hough变换可以检测任意形状的曲线，三维空间Hough变换可以检测空间曲线。

Hough变换用于航迹起始具有以下优点：

（1）Hough变换可以检测任意已知形状的曲线，从而能够起始某类特定航迹，作为先验信息，特定航迹的选定提高了信号相干累积的效率，避免了大量杂波引起的虚假航迹问题；

（2）Hough变换不要求曲线连续或可导，并且对局部缺损和随机噪声鲁棒，适于低检测率和低量测精度下的起始；

（3）Hough变换用于航迹起始不需要目标状态初值，可实现全自动起始。

本文通过大量仿真实验体会到Hough变换航迹起始算法存在若干问题，严重地影响这航迹起始的性能，具体体现在：

（1）标准Hough变换的计算量庞大，并且需要很大的存储空间，在低信噪比环境下表现得特别明显，延长了航迹起始时间，达不到快速性起始的要求。

（2）Hough变换航迹起始算法在提取航迹参数时一般使用阈值法，这就不可避免的出现航迹簇拥现象，即一个目标产生了参数近似的多条轨迹。如何选择峰值提取方法既能准确地提取峰值，同时又能有效地解决航迹簇拥现象，是亟待解决的问题。

（3）Hough变换航迹起始算法应用于工程实际带来了诸多问题，如：实际系统中涉及参数众多，如何对这些参数进行全面有效地利用，是急需解决地问题；针对不同地起始环境应该研究不同地专用算法与之相匹配；实际系统中，很多参数地选取都不能由理论公式推导得出，而要通过大量得仿真实验得出。

### 2.3 典型Hough变换航迹起始算法

#### 2.3.1 标准Hough变换航迹起始算法

在实际应用中，直线方程得斜率-截距式不能表示形如这种直线，因为这时直线得斜率为无穷大。因此将极坐标引入Hough变换，采用变换函数

称为标准Hough变换。其中为原点到直线得法线距离，为该法线与x轴正向所成夹角。这样量测空间中得任意一点将对应参数空间中得一条正弦曲线，量测空间中位于同一条直线上得点确定了参数空间得多条正弦曲线，且这些正弦曲线交与同一点，此交点得坐标确定了原图像中直线得参数。如图2-1所示，量测空间到参数空间的变换：

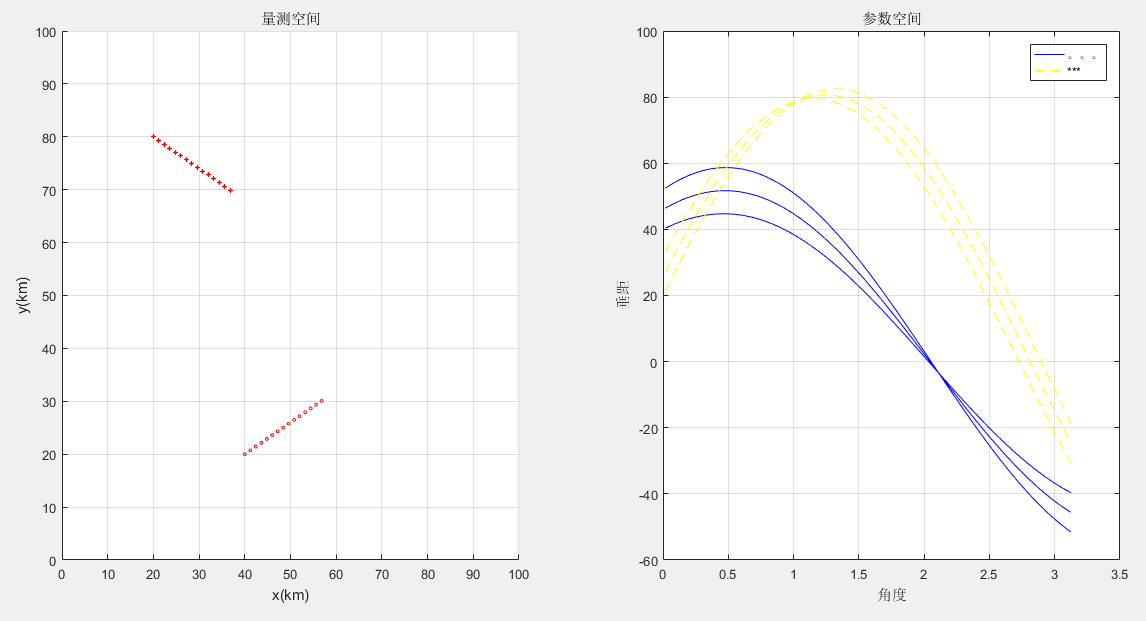


图2-1 量测空间到参数空间的变换

对量测空间中所有点都进行Hough变换后，再对参数空间进行峰值提取，就得到了直线的参数。

标准Hough变换航迹起始算法如下：

输入：所有量测空间中数据点，和代表量测的位置信息，i=1,2,3…,n。

（1）按照一定的间隔将自变量参数离散化，离散取值为;

（2）对于量测空间中每一个数据点和每一个离散化的，计算，将所有保存到矩阵中；

（3）给定因变量量化间隔，对所有进行量化分区，得到矩阵；

（4）按照划分间隔和建立参数空间积累矩阵A，并置每一个元素为0；

（5）对于矩阵中的所有元素，考察是否对积累单元A(j,m)投票，若投票则A(j,m)=A(j,m)+1；

（6）在积累矩阵中寻找局部峰值；

（7）如果>阈值T,那么提取这些参数。

输出：代表了提取出来的航迹参数，航迹起始完毕。

标准Hough变换Matlab程序如下：

close all

clear all

target=2;%目标数

n=15;%起始拍数

k=90;%sig分的个数

m=500;%p分的个数

Monte\_Carlo=100;%Monte\_Carlo仿真次数

L=150;%雷达量测距离

Pd=1;%检测概率

%目标起始坐标及速度

x1=40;y1=20;vx1=0.3;vy1=0.18;%单位km,km/s

x2=20;y2=80;vx2=0.3;vy2=-0.18;

Ts=4;%采样周期，单位s

success=zeros(Monte\_Carlo,target);%目标航迹成功起始矩阵

fake(1:Monte\_Carlo)=0;%目标航迹虚假起始矩阵

track\_number(1:Monte\_Carlo)=0;%总航迹起始数

N=0:n-1;

X1\_init=x1+Ts\*N\*vx1;%真实航迹1

Y1\_init=y1+Ts\*N\*vy1;

Y1\_0=y1-vy1\*x1/vx1;

offset(1)=Y1\_0\*cos(atan(abs(vy1)/abs(vx1)));%航迹1真实垂距

X2\_init=x2+Ts\*N\*vx2;%真实航迹2

Y2\_init=y2+Ts\*N\*vy2;

Y2\_0=y2-vy2\*x2/vx2;

offset(2)=Y2\_0\*cos(atan(abs(vy2)/abs(vx2)));%航迹2真实垂距

Np=1:k;

dNp=pi/k;%参数空间角度间隔

angle=(Np-1/2)\*dNp;

dMp=6\*0.1;%%参数空间垂距间隔

for monte=1:Monte\_Carlo

clear R Rn A0 P0 R X\_za Y\_za noisex noisey

R = poissrnd(50,1,n);%每拍杂波个数，服从泊松分布

Rn=R(1);

X\_za=unifrnd (0, 100, 1, R(1));%随机产生（x,y）坐标，服从0-100的均匀分布

Y\_za=unifrnd (0, 100, 1, R(1));

for i=2:n

X\_za(Rn+1:Rn+R(i))=unifrnd (0, 100, 1, R(i));

Y\_za(Rn+1:Rn+R(i))=unifrnd (0, 100, 1, R(i));

Rn=Rn+R(i);

end

noisex=normrnd(0,0.1,1,n);%x量测噪声

noisey=normrnd(0,0.1,1,n);

X1=X1\_init+noisex;X2=X2\_init+noisex;%实际量测

Y1=Y1\_init+noisey;Y2=Y2\_init+noisey;

A=zeros(k,2\*m);%积累矩阵

%航迹1 Hough变换

for i=1:n

for j=1:k

P(i,j)=X1(i)\*cos(angle(j))+Y1(i)\*sin(angle(j));

end

end

%航迹2 Hough变换

for i=(n+1):(2\*n)

for j=1:k

P(i,j)=X2(i-n)\*cos(angle(j))+Y2(i-n)\*sin(angle(j));

end

end

%杂波的Hough变换

for i=2\*n+1:(2\*n+Rn)

for j=1:k

P(i,j)=X\_za(i-2\*n)\*cos(angle(j))+Y\_za(i-2\*n)\*sin(angle(j));

end

end

%对积累矩阵投票

for i=1:k

for j=1:2\*m

a=-L+(j-1)\*dMp;

b=-L+j\*dMp;

for h=1:2\*n+Rn

if (P(h,i)>=a && P(h,i)<b)

A(i,j)=A(i,j)+1;

end

end

end

end

%寻找投票数大于等于阈值的参数

count=0;

for i=1:k

for j=1:2\*m

if A(i,j)>=Pd\*n\*0.95%阈值=检测概率\*起始拍数\*（0到1之间的值）

count=count+1;

P0(count)=-L+(j-1/2)\*dMp;%存储符合要求的参数

A0(count)=angle(i);%存储符合要求的参数

end

end

end

track\_number(monte)=count;%记录每次Monte\_Carlo仿真总航迹数

%输出符合要求的参数

for h=1:count

flag=0;

for din=1:target

if abs(P0(h)-offset(din))<=3

success(monte,din)=1;

flag=1;

end

end

if flag==0

fake(monte)=fake(monte)+1;

end

fprintf('the value of P0 is %f;the value of A0 is %f\n',P0(h),A0(h));

end

fprintf('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n');

%绘图

figure

subplot(1,2,1)

scatter(X1,Y1,5,'r');

hold on

scatter(X2,Y2,6,'\*','r');

hold on

scatter(X\_za,Y\_za,3,'filled','g')

hold on

xlabel('x(km)');

ylabel('y(km)');

legend('真实航迹1','真实航迹2','杂波');

title('量测图')

axis([0 100 0 100])

axis on

subplot(1,2,2)

scatter(X1,Y1,5,'r');

hold on

scatter(X2,Y2,6,'\*','r');

hold on

scatter(X\_za,Y\_za,3,'filled','g')

hold on

for h=1:count

X=0:1:100;

YS=(P0(h)-X\*cos(A0(h)))/(sin(A0(h)));

plot(X,YS,'b');

hold on

xlabel('x(km)');

ylabel('y(km)');

legend('真实航迹1','真实航迹2','杂波','起始航迹');

title('起始结果图')

axis([0 100 0 100])

axis on

end

end

%计算航迹起始成功率

success\_number=0;

for i=1:Monte\_Carlo

for j=1:target

success\_number=success\_number+success(i,j);

end

end

success\_rate=success\_number/(Monte\_Carlo\*target);

fprintf('the rate of the successful Track initialization is %f%%\n',success\_rate\*100);

%计算航迹虚假率

fake\_rate=fake./track\_number;

fake\_mean=0;

for i=1:Monte\_Carlo

fprintf('the rate of the fake Track initialization is %f%%\n',fake\_rate(i)\*100);

fake\_mean=fake\_mean+fake\_rate(i);

end

fprintf('the mean of the fake Track initialization is %f%%\n',fake\_mean/Monte\_Carlo\*100);

figure

plot(fake\_rate.\*100);

xlabel('Monte\_Carlo');

ylabel('fake\_rate(%)');

title('虚假航迹率');

#### 2.3.2 修正Hough变换航迹起始算法

标准Hough变换航迹起始算法需要经过多拍扫描后才能得到满意的效果，但是起始航迹建立在多次扫描的基础上是不符合实际工程需要的，最明显的缺陷表现为计算量大和起始时间过长。修正Hough变换充分考虑了目标的运动学信息，仅用较少拍数的量测就可以完成起始，在检测概率较高的环境下具有良好的起始性能。

设zi(k),i=1,…,lk是k时刻的第i个量测点迹，假定雷达在第k、k+1、k+2扫描时刻分别接受到三个量测zi(k)、zj(k+1)、zm(k+2),通过式(2-5)可以将这三个量测转换为参数空间中的三条曲线、、。由此可得差分函数：

由得过零交汇点可以得到两条信息。首先，它提供了交汇点和对应的坐标；其次，过零处的斜率符号取决于的指向。根据这两条信息可以建立两个航迹起始判据。

判据一：过零点的和必须非常接近，即

式中为允许误差，n为一较小的正整数。

判据二：和过零率符号必须相同。

修正Hough变换航迹起始算法如下：

输入：所有量测空间中数据点，和代表量测的位置信息，i=1,2,3…,n。

（1）设置最大速度Vmax和最小速度Vmin，以下对连续三拍量测进行处理。

（2）取第k拍的第i个量测、第k+1拍的第j个量测与第k+2拍的第m个量测zm(k+2)进行遍历式关联，分别求出第k拍与第k+1拍量测间的速度Vij和第k+1拍与第k+2拍量测间的速度Vjm。将同时满足条件和的三拍量测保存下来。

（3）将第k拍量测与第k+1拍量测代入方程：

解方程得到参数，，保存为候选航迹。

（4）同（3）一样，解出第k+1拍量测与第k+2拍量测得参数，，分别与(3)中保存得参数进行比较，如果满足判据一：和判据二：和的斜率符号必须相同，那么将这三拍量测同时保存到航迹矩阵Para中，这就是三拍量测的起始结果；

输出：Para中保存的就是起始的航迹，航迹起始完毕。

修正Hough变换Matlab程序如下：

close all

clear all

target=2;%目标数

n=3;%起始拍数

k=90;%sig分的个数

m=500;%p分的个数

Monte\_Carlo=100;%Monte\_Carlo仿真次数

L=150;%雷达量测距离 单位（km）

Vmax=0.7;%目标最大速度 单位（km/s）

Vmin=0.05;%目标最小速度

%目标起始坐标及速度

x1=40;y1=20;vx1=0.3;vy1=0.18;%单位km,km/s

x2=20;y2=80;vx2=0.3;vy2=-0.18;

Ts=4;%采样周期，单位s

success=zeros(Monte\_Carlo,target);%目标航迹成功起始矩阵

fake(1:Monte\_Carlo)=0;%目标航迹虚假起始矩阵

track\_number(1:Monte\_Carlo)=0;%总航迹起始数

N=0:n-1;

X1\_init=x1+Ts\*N\*vx1;%真实航迹1

Y1\_init=y1+Ts\*N\*vy1;

Y1\_0=y1-vy1\*x1/vx1;

offset(1)=Y1\_0\*cos(atan(abs(vy1)/abs(vx1)));%航迹1真实垂距

X2\_init=x2+Ts\*N\*vx2;%真实航迹2

Y2\_init=y2+Ts\*N\*vy2;

Y2\_0=y2-vy2\*x2/vx2;

offset(2)=Y2\_0\*cos(atan(abs(vy2)/abs(vx2)));%航迹2真实垂距

Np=1:k;

dNp=pi/k;%参数空间角度间隔

angle=(Np-1/2)\*dNp;

dMp=6\*0.1;%参数空间垂距间隔

R = poissrnd(20,1,n);%每拍杂波个数，服从泊松分布

Rn=R(1);

X\_za=unifrnd (0, 100, 1, R(1));%随机产生（x,y）坐标，服从0-100的均匀分布

Y\_za=unifrnd (0, 100, 1, R(1));

for i=2:n

X\_za(Rn+1:Rn+R(i))=unifrnd (0, 100, 1, R(i));

Y\_za(Rn+1:Rn+R(i))=unifrnd (0, 100, 1, R(i));

Rn=Rn+R(i);

end

noisex=normrnd(0,0.05,1,n);%x量测噪声

noisey=normrnd(0,0.05,1,n);%y量测噪声

X1=X1\_init+noisex;X2=X2\_init+noisex;%实际量测

Y1=Y1\_init+noisey;Y2=Y2\_init+noisey;

Z1(1,1:2)=[X1(1),Y1(1)];

Z1(2,1:2)=[X2(1),Y2(1)];

Z1(3:2+R(1),1:2)=[(X\_za(1:R(1)))',(Y\_za(1:R(1)))'];

Z2(1,1:2)=[X1(2),Y1(2)];

Z2(2,1:2)=[X2(2),Y2(2)];

Z2(3:2+R(2),1:2)=[(X\_za(1:R(2)))',(Y\_za(1:R(2)))'];

Z3(1,1:2)=[X1(3),Y1(3)];

Z3(2,1:2)=[X2(3),Y2(3)];

Z3(3:2+R(3),1:2)=[(X\_za(1:R(3)))',(Y\_za(1:R(3)))'];

%遍历式关联

number=1;

for i=1:2+R(1)

for j=1:2+R(2)

for h=1:2+R(3)

x11=Z1(i,1);y11=Z1(i,2);

x21=Z2(j,1);y21=Z2(j,2);

x31=Z3(h,1);y31=Z3(h,2);

V12=sqrt((x11-x21)\*(x11-x21)+(y11-y21)\*(y11-y21))/Ts;

V23=sqrt((x31-x21)\*(x31-x21)+(y31-y21)\*(y31-y21))/Ts;

if Vmin<=V12 && V12<=Vmax && Vmin<=V23 && V23<=Vmax

Zc1(number,1:2)=[x11,y11];

Zc2(number,1:2)=[x21,y21];

Zc3(number,1:2)=[x31,y31];

number=number+1;

end

end

end

end

%计算垂距

for i=1:number-1

x11=Zc1(i,1);y11=Zc1(i,2);

x21=Zc2(i,1);y21=Zc2(i,2);

x31=Zc3(i,1);y31=Zc3(i,2);

for j=1:k

P1(i,j)=x11\*cos(angle(j))+y11\*sin(angle(j));

P2(i,j)=x21\*cos(angle(j))+y21\*sin(angle(j));

P3(i,j)=x31\*cos(angle(j))+y31\*sin(angle(j));

end

end

%解过零处角度

for i=1:number-1

x11=Zc1(i,1);y11=Zc1(i,2);

x21=Zc2(i,1);y21=Zc2(i,2);

x31=Zc3(i,1);y31=Zc3(i,2);

sign12(i)=atan((x21-x11)/(y11-y21));

sign23(i)=atan((x31-x21)/(y21-y31));

end

%寻找符合要求的量测点

count=1;

for i=1:number-1

x11=Zc1(i,1);y11=Zc1(i,2);

x21=Zc2(i,1);y21=Zc2(i,2);

x31=Zc3(i,1);y31=Zc3(i,2);

slope12=(y21-y11)/(x21-x11);

slope23=(y31-y21)/(x31-x21);

if abs(sign12(i)-sign23(i))<=3\*dNp && (slope23/slope12)>0

para(count,:)=[x11,y11,x21,y21,x31,y31];

count=count+1;

end

end

%输出符合要求的参数

for h=1:count-1

for j=1:6

fprintf('%f ',para(h,j));

end

fprintf('\n');

end

fprintf('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n');

%绘图

figure

subplot(1,2,1)

scatter(X1,Y1,5,'r');

hold on

scatter(X2,Y2,6,'\*','r');

hold on

scatter(X\_za,Y\_za,3,'filled','g')

hold on

xlabel('x(km)');

ylabel('y(km)');

legend('真实航迹1','真实航迹2','杂波');

title('量测图')

axis([0 100 0 100])

axis on

%figure

subplot(1,2,2)

for h=1:count-1

X=[para(h,1),para(h,3),para(h,5)];

YS=[para(h,2),para(h,4),para(h,6)];

plot(X,YS,'b');

xlabel('x(km)');

ylabel('y(km)');

legend('起始航迹');

title('起始结果图')

hold on

axis([0 100 0 100])

axis on

end

#### 2.3.3 简易序列Hough变换

标准Hough变换投票结束后，参数空间中投票值的分布具有一定的规律性。假设检测区域中有三个目标做匀速直线运动，经过Hough变换后，参数空间记录了对这两条航迹的投票情况。假设最大峰值出现在量化值,处，那么它附近也会出现一些较高的峰值。如果简单用阈值法来提取峰值，那么在最大峰值被提取出来的同时，周围一些超过阈值的峰值也将被提取出来，并且这些值代表同一条航迹的参数。这就是标准Hough变换航迹起始算法中的航迹簇拥现象。

航迹簇拥现象与设定的阈值密切相关。对于单目标情况，在较低的阈值下航迹簇拥现象非常严重，选择较高的阈值可以减少峰值簇拥现象。对于多目标情况，较高的阈值则会导致航迹的检漏。由于不知到目标的具体情况，如目标量测拍数，噪声及杂波情况，检测概率信息等等，使用阈值法提取峰值时，峰值簇拥现象不可避免。本节的简易Hough变换采用了新的峰值提取方法，在一定程度上解决了航迹簇拥问题。

简易Hough变换的积累过程和标准Hough变换完全一样。所谓序列是指在峰值提取时，每次都检测一个全局最大值，记录并保存最大峰值在参数空间的位置,同时将对该单元及该单元附近一小部分的区域的投票置0，之后再次寻找最大峰值并记录保存，直到峰值个数达到设定值。所以序列Hough变换较为适用于多目标航迹起始，由于每次只提取最大峰值，所以Hough变换可以有效地解决峰值簇拥现象，尤其在已知目标个数时。

简易序列Hough变换航迹起始算法如下：

输入：所有量测空间中数据点，和代表量测的位置信息，i=1,2,3…,n。

（1）设定序列数S，初始化s=1；

（2）按照一定的间隔将自变量参数离散化，离散取值为;

（3）对于量测空间中每一个数据点和每一个离散化的，计算，将所有保存到矩阵中；

（4）给定因变量量化间隔，对所有进行量化分区，得到矩阵；

（5）按照划分间隔和建立参数空间积累矩阵A，并置每一个元素为0；

（6）对于矩阵中的所有元素，考察是否对积累单元A(j,m)投票，若投票则A(j,m)=A(j,m)+1；

（7）在积累矩阵中寻找全局最大值,保存并将对及其附近一小部分区域地投票置0；

（8）s=s+1, 如果是s<=S,继续进行第(7)步，否则提取这些参数。

输出：代表了提取出来的航迹参数，航迹起始完毕。

序列Hough变换Matlab程序如下：

close all

clear all

target=2;%目标数

targets=2;%序列数

n=15;%起始拍数

k=90;%sig分的个数

m=200;%p分的个数

Monte\_Carlo=10;%Monte\_Carlo仿真次数

L=150;%雷达量测距离

Pd=1;%检测概率

%目标起始坐标及速度

x1=40;y1=20;vx1=0.3;vy1=0.18;%单位km,km/s

x2=20;y2=80;vx2=0.3;vy2=-0.18;

Ts=4;%采样周期，单位s

success=zeros(Monte\_Carlo,target);%目标航迹成功起始矩阵

fake(1:Monte\_Carlo)=0;%目标航迹虚假起始矩阵

track\_number(1:Monte\_Carlo)=0;%总航迹起始数

N=0:n-1;

X1\_init=x1+Ts\*N\*vx1;%目标航迹

Y1\_init=y1+Ts\*N\*vy1;

Y1\_0=y1-vy1\*x1/vx1;

offset(1)=Y1\_0\*cos(atan(abs(vy1)/abs(vx1)));%航迹1真实垂距

X2\_init=x2+Ts\*N\*vx2;

Y2\_init=y2+Ts\*N\*vy2;

Y2\_0=y2-vy2\*x2/vx2;

offset(2)=Y2\_0\*cos(atan(abs(vy2)/abs(vx2)));%航迹2真实垂距

Np=1:k;

dNp=pi/k;%参数空间角度间隔

angle=(Np-1/2)\*dNp;

dMp=6\*0.1;%参数空间垂距间隔

for monte=1:Monte\_Carlo

clear A0 P0 R X\_za Y\_za noisex noisey

R = poissrnd(50,1,n);%每拍杂波个数

Rn=R(1);

X\_za=unifrnd (0, 100, 1, R(1));

Y\_za=unifrnd (0, 100, 1, R(1));

for i=2:n

X\_za(Rn+1:Rn+R(i))=unifrnd (0, 100, 1, R(i));

Y\_za(Rn+1:Rn+R(i))=unifrnd (0, 100, 1, R(i));

Rn=Rn+R(i);

end

noisex=normrnd(0,0.1,1,n);%x量测噪声

noisey=normrnd(0,0.1,1,n);%y量测噪声

X1=X1\_init+noisex;X2=X2\_init+noisex;%真实量测

Y1=Y1\_init+noisey;Y2=Y2\_init+noisey;

A=zeros(k,2\*m);%积累矩阵

%航迹1 Hough变换

for i=1:n

for j=1:k

P(i,j)=X1(i)\*cos(angle(j))+Y1(i)\*sin(angle(j));

end

end

%航迹2 Hough变换

for i=(n+1):(2\*n)

for j=1:k

P(i,j)=X2(i-n)\*cos(angle(j))+Y2(i-n)\*sin(angle(j));

end

end

%杂波的Hough变换

for i=2\*n+1:(2\*n+Rn)

for j=1:k

P(i,j)=X\_za(i-2\*n)\*cos(angle(j))+Y\_za(i-2\*n)\*sin(angle(j));

end

end

%对积累矩阵投票

for i=1:k

for j=1:2\*m

a=-L+(j-1)\*dMp;

b=-L+j\*dMp;

for h=1:2\*n+Rn

if (P(h,i)>=a && P(h,i)<b)

A(i,j)=A(i,j)+1;

end

end

end

end

%寻找投票数最大的两个参数

for s=1:targets

max=A(1,1);

maxi=1;maxj=1;

for i=1:k

for j=1:2\*m

if A(i,j)>=max

max=A(i,j);

maxi=i;

maxj=j;

end

end

end

for i=maxi-5:maxi+5

for j=maxj-5:maxj+5

if i<=0

i=1;

end

if j<=0

j=1;

end

A(i,j)=0;

end

end

P0(s)=-L+(maxj-1/2)\*dMp;

A0(s)=angle(maxi);

%输出符合要求的参数

flag=0;

for din=1:target

if abs(P0(s)-offset(din))<=3

success(monte,din)=1;

flag=1;

end

end

if flag==0

fake(monte)=fake(monte)+1;

end

fprintf('the value of P0 is %f;the value of A0 is %f\n',P0(s),A0(s));

end

fprintf('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n');

%绘图

figure

subplot(1,2,1)

scatter(X1,Y1,5,'r');

hold on

scatter(X2,Y2,6,'\*','r');

hold on

scatter(X\_za,Y\_za,3,'filled','g')

hold on

xlabel('x(km)');

ylabel('y(km)');

legend('真实航迹1','真实航迹2','杂波');

title('量测图')

axis([0 100 0 100])

axis on

subplot(1,2,2)

scatter(X1,Y1,5,'r');

hold on

scatter(X2,Y2,6,'\*','r');

hold on

scatter(X\_za,Y\_za,3,'filled','g')

hold on

for s=1:targets

X=0:1:100;

YS=(P0(s)-X\*cos(A0(s)))/(sin(A0(s)));

plot(X,YS,'b');

hold on

xlabel('x(km)');

ylabel('y(km)');

legend('真实航迹1','真实航迹2','杂波','起始航迹');

title('起始结果图')

hold on

axis([0 100 0 100])

axis on

end

end

%计算航迹起始成功率

success\_number=0;

for i=1:Monte\_Carlo

for j=1:target

success\_number=success\_number+success(i,j);

end

end

success\_rate=success\_number/(Monte\_Carlo\*target);

fprintf('the rate of the successful Track initialization is %f%%\n',success\_rate\*100);

%计算航迹虚假率

fake\_rate=fake./targets;

fake\_mean=0;

for i=1:Monte\_Carlo

fprintf('the rate of the fake Track initialization is %f%%\n',fake\_rate(i)\*100);

fake\_mean=fake\_mean+fake\_rate(i);

end

fprintf('the mean of the fake Track initialization is %f%%\n',fake\_mean/Monte\_Carlo\*100);

figure

plot(fake\_rate.\*100);

xlabel('Monte\_Carlo');

ylabel('fake\_rate(%)');

title('虚假航迹率');

### 2.4 仿真分析

以多目标航迹起始为例，假定监视区域内有两个目标做匀速直线运动，初始值分别为：

目标1：x1=10km,y1=15km,vx1=0.3km/s,vy1=-0.18km/s;

目标2：x2=20km,y1=80km,vx1=0.3km/s,vy1=-0.18km/s;

采样周期Ts=4s;量测噪声为均值等于0、方差是的高斯白噪声；，；每周期的杂波个数服从泊松分布，通过改变参数，可以得到不同强度的杂波。目标速度范围是：最大速度Vmax=0.7km/s, 最小速度Vmin=0.05km/s。以下仿真在Matlab2016b中进行。

查阅文献资料，对于标准Hough变换，起始拍数在10拍以上为宜，此时的航迹起始成功率和航迹虚假率都比较好。

下面给出在100次Monte-Carlo仿真下，5拍、10拍、15拍时的航迹起始成功率和航迹虚假率：

（1）标准Hough变换：

表2-1 标准Hough变换起始拍数对航迹起始性能的影响

|  |  |  |
| --- | --- | --- |
| 拍数 | 成功率（%） | 平均虚假率（%） |
| 5 | 100.0 | 88.3 |
| 10 | 99.5 | 70.9 |
| 15 | 100.0 | 32.9 |

（2）简易序列Hough变换

表2-2 序列Hough变换起始拍数对航迹起始性能的影响

|  |  |  |
| --- | --- | --- |
| 拍数 | 成功率（%） | 平均虚假率（%） |
| 5 | 28.5 | 71.0 |
| 10 | 69.5 | 29.0 |
| 15 | 92.0 | 7.0 |

标准Hough变换仿真：检测概率Pd=1，杂波强度=50，起始结果如图2-2所示。从图中看出，虽然2条真实目标航迹被准确起始，但是同时也起始了多条虚假航迹，且存在航迹簇拥现象。

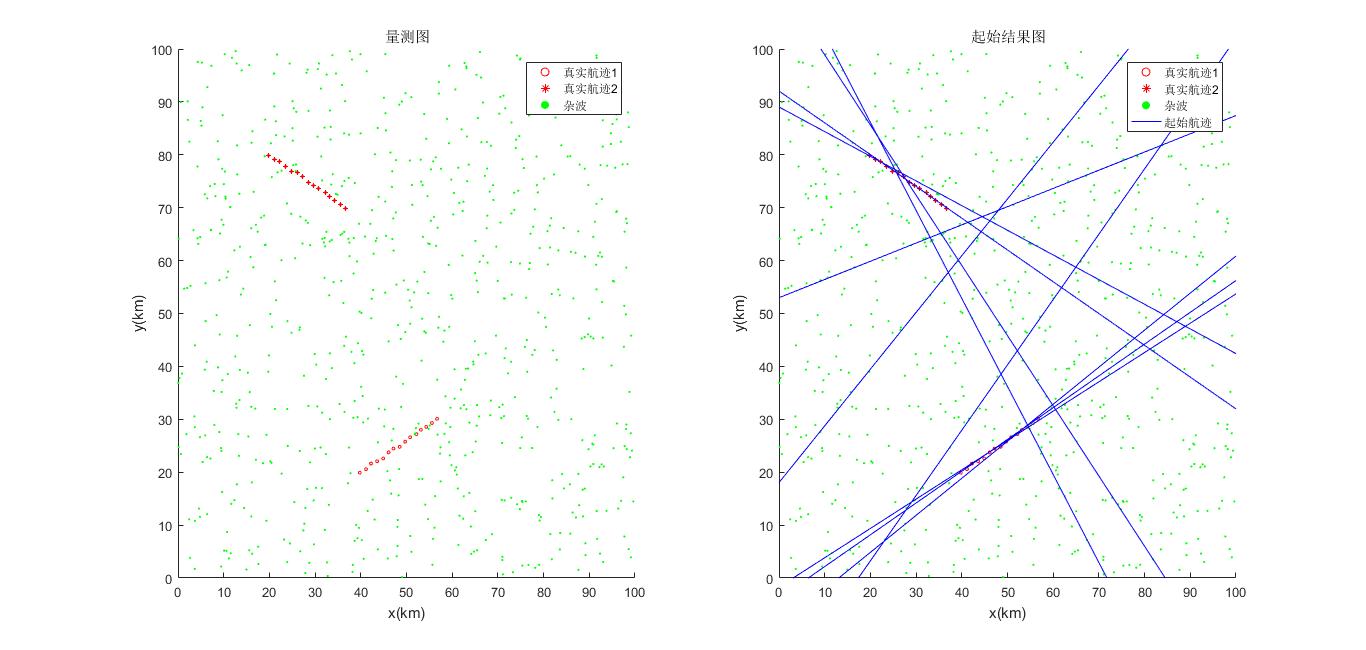


图2-2 标准Hough变换航迹起始结果图

简易序列Hough变换仿真：检测概率Pd=1，杂波强度=50，序列数S=2，起始结果如图2-3所示。从图中看出，2条真实目标航迹被准确起始，同时也抑制虚假航迹，且避免了航迹簇拥现象。

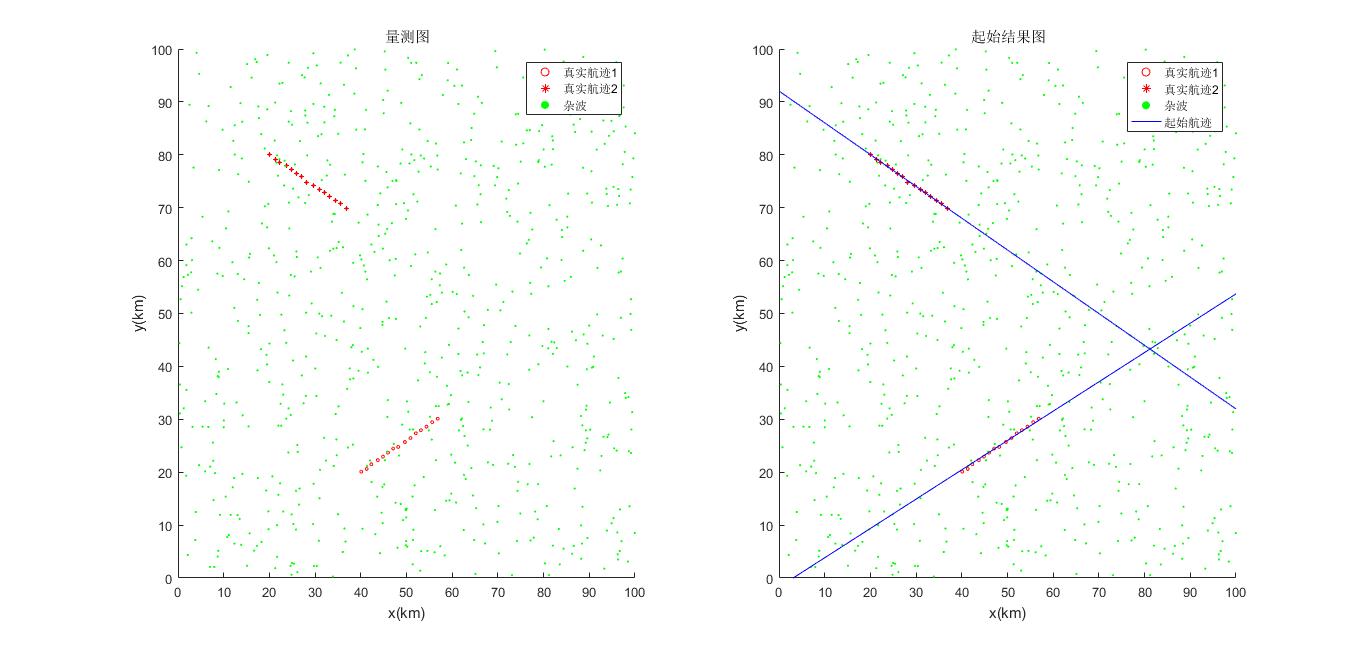


图2-3 序列Hough变换航迹起始结果图

修正Hough变换仿真：检测概率Pd=1，杂波强度=20，起始结果如图2-4所示。从图中看出，该算法使用了3拍量测将2条真实目标航迹准确起始。

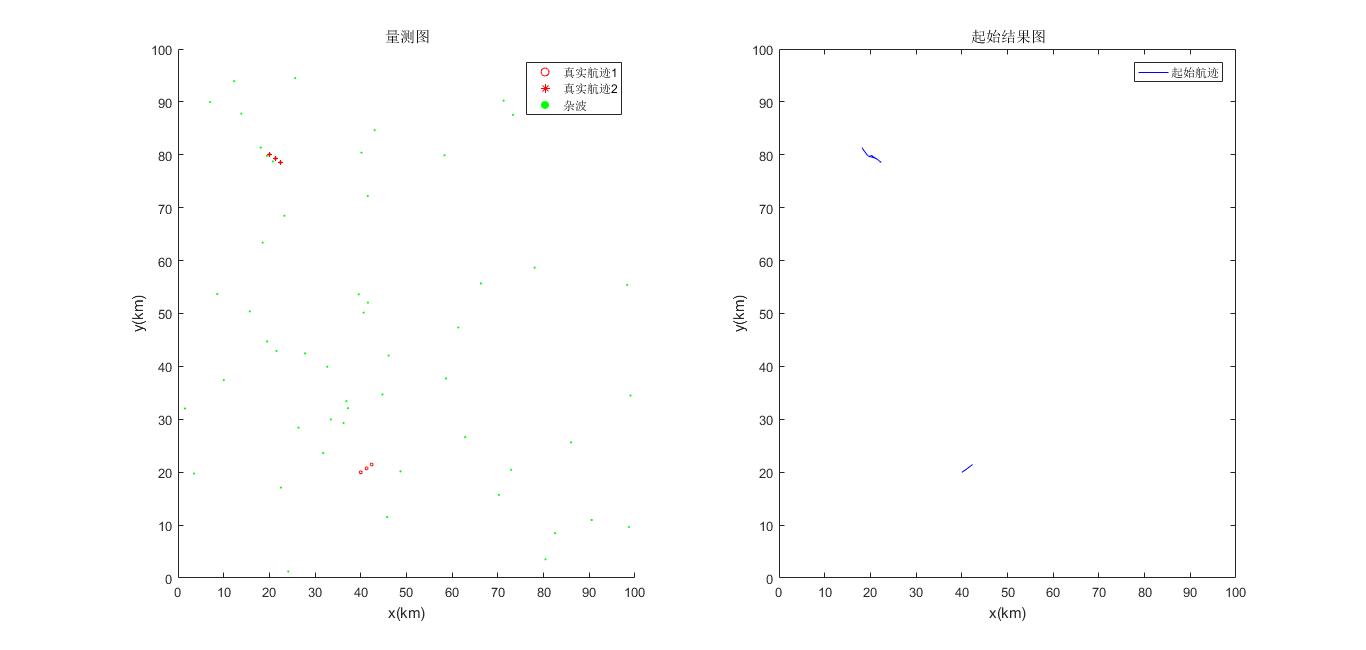


图2-4 修正Hough变换航迹起始结果图

综合分析可得，标准Hough变换航迹起始成功率最高，但是其虚假航迹占有率过高，且航迹簇拥现象十分严重，该算法不适于直接起始，可以作为一种预处理算法。修正Hough变换没有标准Hough变换的虚假航迹占有率过高和航迹簇拥问题，但收起始拍数限制，只适用于高检测率下的起始问题，有很大局限性。序列Hough变换有效解决了航迹簇拥现象，但起始时间依旧较长，起始性能受限于序列数S。

## 小结

针对需要解决的不同问题，本次课程设计给出了几种典型的Hough变换航迹起始算法。首先讲述了各种算法的基本原理，然后给出了算法步骤和相应的Matlab仿真程序。最后通过仿真给出了每种算法的有缺点以及适用环境。

本次课程设计仍有许多不足以及可以改进的空间，比如缺乏曲线航迹以及三维空间的仿真分析，除此之外仍有许多有关Hough变换航迹起始的问题以及资料未能涉猎，许多参考文献中的优秀算法也未能详尽研究。

## 参考资料

[1] 王峰. 基于Hough变换的航迹起始算法[J]. 杭州电子科技大学学报, 2008, 28(6):87-90.

[2] 汤金平, 时银水, 朱岩,等. 一种基于Hough变换的航迹起始新方法[J]. 舰船电子对抗, 2009, 32(2):87-91.

[3] 贺鹏. Hough变换航迹起始算法研究[D]. 西北工业大学, 2007.

[4] 杨峰. 《信息融合（目标跟踪）》讲义

## 致谢

首先向我的指导老师杨峰老师致以诚挚的谢意！

本次课程设计是在杨峰副教授的直接指导下完成的。在本次的课程设计选题、确定设计方向以及实际研究等方面都得到了杨老师悉心的指导和启发。再次向杨老师致以最衷心的感谢！

感谢同级一起生活的同学，在课程设计过程中，提供给我一个宽松、和谐的学习环境。

课程设计成绩评定表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 质量评价指标（在相应栏目打√） | | | | | | |
| 评 价 项 目 | 评 价 质 量 | | | | | |
| 优秀 | | 良好 | 一般 | 及格 | 不及格 |
| 工作量和态度 |  | |  |  |  |  |
| 实验、计算可靠性 |  | |  |  |  |  |
| 文字和图表质量 |  | |  |  |  |  |
| 总体评价 |  | |  |  |  |  |
| 评定成绩（百分制） | |  | | | | |
| 评定小组成员签名 | | 2018年 月 日 | | | | |
| 备 注 | |  | | | | |