



西北工业大学

# 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

## 1. 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地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，特别适用于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。

## 2. 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  $f$  来表述：

$$f((\hat{p}_1, \hat{p}_2 \dots \hat{p}_n), (x, y)) = 0 \quad (2-1)$$

其中  $\hat{p}_1, \hat{p}_2 \dots \hat{p}_n$  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：

$$g((\hat{x}, \hat{y}), (p_1, p_2 \dots p_n)) = 0 \quad (2-2)$$

这就建立了一个图像空间到参数空间的映射，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  $p_1, p_2 \dots p_n$  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：

$$f((\hat{m}, \hat{c}), (x, y)) = y - \hat{m}x - \hat{c} = 0 \quad (2-3)$$

这是直线方程的斜率-截距式表达式，其中， $\hat{m}$  是该直线的斜率， $\hat{c}$  是截距。相应的，将  $(\hat{m}, \hat{c})$  与  $(x, y)$  交换角色，得到：

$$g((\hat{x}, \hat{y}), (m, c)) = \hat{y} - \hat{x}m - c = 0 \quad (2-4)$$

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，是急需解决的问题；针对不同的起始环境应该研究不同的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，而要通过大量的仿真实验得出。

## 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  $y - mx - c = 0$  不能表示形如  $x = a$  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

$$\rho = x \cdot \cos \theta + y \cdot \sin \theta \quad 0 \leq \theta < \pi, \quad (2-5)$$

称为标准 Hough 变换。其中  $\rho$  为原点到直线得法线距离， $\theta$  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  $(x_i, y_i)$  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中得多条正弦曲线，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:

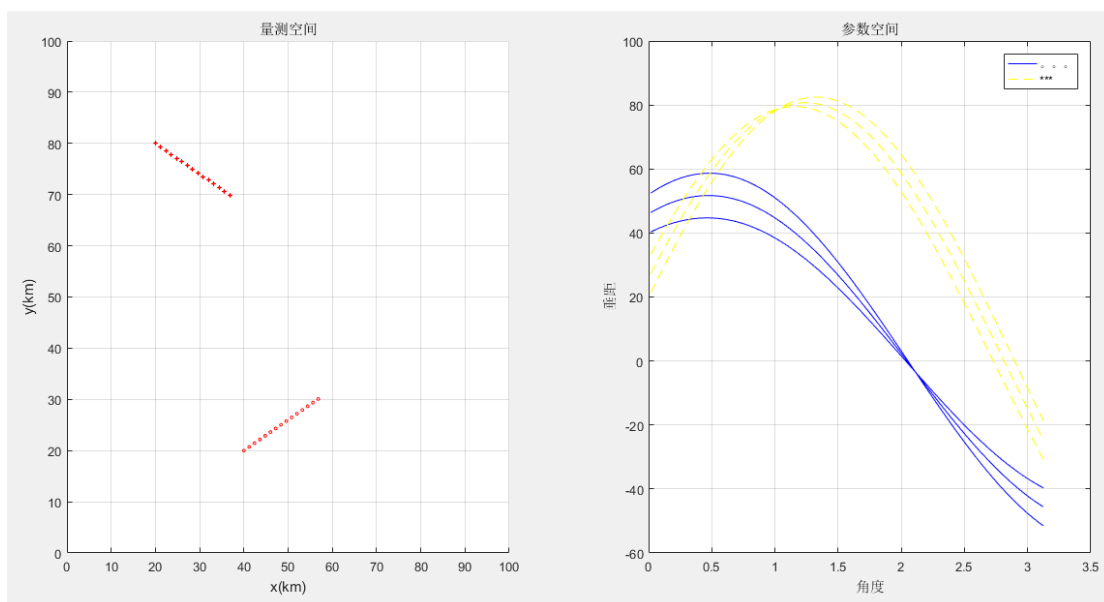


图 2-1 量测空间到参数空间的变换

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  $(x_i, y_i)$ ,  $x_i$  和  $y_i$  代表量测的位置信息,  $i=1, 2, 3, \dots, n$ 。

- (1) 按照一定的间隔  $\Delta\theta$  将自变量参数离散化, 离散取值为  $\theta_k, k = 1, 2, 3, \dots, h$ ;
- (2) 对于量测空间中每一个数据点  $(x_i, y_i)$  和每一个离散化的  $\theta_k$ , 计算  $\rho_{ik} = x_i \cos \theta_k + y_i \sin \theta_k$ , 将所有  $\rho_{ik}$  保存到矩阵  $\rho$  中;
- (3) 给定因变量量化间隔  $\Delta\rho$ , 对所有  $\rho_{ik}$  进行量化分区, 得到矩阵  $\rho'$ ;
- (4) 按照划分间隔  $\Delta\theta$  和  $\Delta\rho$  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  $\rho'$  中的所有元素  $\rho'(j, i)$ , 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  $A(\theta_{peak}, \rho_{peak})$ ;
- (7) 如果  $A(\theta_{peak}, \rho_{peak}) > \text{阈值 } T$ , 那么提取这些参数  $(\theta_i, \rho_i)$ 。

输出:  $(\theta_i, \rho_i)$  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 noisy
    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  $z_i(k)$ ,  $i=1, \dots, l_k$  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  $z_i(k)$ 、 $z_j(k+1)$ 、 $z_m(k+2)$ ,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  $\rho_k$ 、 $\rho_{k+1}$ 、 $\rho_{k+2}$ 。由此可得差分函数:

$$\Delta\rho_k = \rho_{k+1} - \rho_k \quad (2-6)$$

由  $\Delta\rho_k$  得过零交汇点可以得到两条信息。首先,它提供了交汇点  $\rho_{k+1}$  和  $\rho_k$  对应的  $\theta$  坐标  $\theta_k^0$ ;其次,过零处的斜率符号取决于  $z_j(k+1) - z_i(k)$  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  $\theta_k^0$  和  $\theta_{k+1}^0$  必须非常接近,即

$$|\theta_{k+1}^0 - \theta_k^0| \leq \sigma_0 \quad (2-7)$$

式中  $\Delta\theta \leq \sigma_0 \leq n\Delta\theta$  为允许误差,  $n$  为一较小的正整数。

判据二:  $\theta_k^0$  和  $\theta_{k+1}^0$  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  $(x_i, y_i)$ ， $x_i$  和  $y_i$  代表量测的位置信息， $i=1, 2, 3, \dots, n$ 。

(1) 设置最大速度  $V_{\max}$  和最小速度  $V_{\min}$ ，以下对连续三拍量测进行处理。

(2) 取第  $k$  拍的第  $i$  个量测  $z_i(k)$ 、第  $k+1$  拍的第  $j$  个量测  $z_j(k+1)$  与第  $k+2$  拍的第  $m$  个量测  $z_m(k+2)$  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  $V_{ij}$  和第  $k+1$  拍与第  $k+2$  拍量测间的速度  $V_{jm}$ 。将同时满足条件  $V_{\min} \leq V_{ij} \leq V_{\max}$  和  $V_{\min} \leq V_{jm} \leq V_{\max}$  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：

$$\begin{cases} x_k \cos \theta + y_k \sin \theta = \rho \\ x_{k+1} \cos \theta + y_{k+1} \sin \theta = \rho \end{cases} \quad (2-8)$$

解方程得到参数  $\theta_{ij}$ ， $\rho_{ij}$ ，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  $\theta_{jm}$ ， $\rho_{jm}$ ，分别与 (3) 中保存得参数进行比较，如果满足判据一： $|\theta_{ij} - \theta_{jm}| \leq \sigma$  和判据二： $\theta_{ij}$  和  $\theta_{jm}$  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);
```

布

```
X_z=unifrnd (0, 100, 1, R(1));%随机产生 (x,y) 坐标，服从 0-100 的均匀分
```

```
Y_z=unifrnd (0, 100, 1, R(1));
```

```
for i=2:n
```

```
    X_z(Rn+1:Rn+R(i))=unifrnd (0, 100, 1, R(i));
```

```
    Y_z(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 $(\rho_{01}, \theta_{01})$ ， $(\rho_{02}, \theta_{02})$ 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



$(\rho_0, \theta_0)$ , 同时将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  $(x_i, y_i)$ ,  $x_i$  和  $y_i$  代表量测的位置信息,  $i=1, 2, 3, \dots, n$ 。

- (1) 设定序列数  $S$ , 初始化  $s=1$ ;
  - (2) 按照一定的间隔  $\Delta\theta$  将自变量参数离散化, 离散取值为  $\theta_k, k = 1, 2, 3, \dots, h$ ;
  - (3) 对于量测空间中每一个数据点  $(x_i, y_i)$  和每一个离散化的  $\theta_k$ , 计算  $\rho_{ik} = x_i \cos \theta_k + y_i \sin \theta_k$ , 将所有  $\rho_{ik}$  保存到矩阵  $\rho$  中;
  - (4) 给定因变量量化间隔  $\Delta\rho$ , 对所有  $\rho_{ik}$  进行量化分区, 得到矩阵  $\rho'$ ;
  - (5) 按照划分间隔  $\Delta\theta$  和  $\Delta\rho$  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  $\rho'$  中的所有元素  $\rho'(j, i)$ , 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  $A(\theta_{max}, \rho_{max})$ , 保存  $(\theta_i, \rho_i)$  并将对  $A(\theta_{max}, \rho_{max})$  及其附近一小部分区域地投票置 0;
  - (8)  $s=s+1$ , 如果是  $s \leq S$ , 继续进行第 (7) 步, 否则提取这些参数  $(\theta_i, \rho_i)$ 。
- 输出:  $(\theta_i, \rho_i)$  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 noisy
    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
    noisy=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+noisy;Y2=Y2_init+noisy;

    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;
            end
        end
    end
end

```

```
        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：  $x_1=10\text{km}$ ,  $y_1=15\text{km}$ ,  $v_{x1}=0.3\text{km/s}$ ,  $v_{y1}=-0.18\text{km/s}$ ;

目标 2：  $x_2=20\text{km}$ ,  $y_2=80\text{km}$ ,  $v_{x2}=0.3\text{km/s}$ ,  $v_{y2}=-0.18\text{km/s}$ ;

采样周期  $T_s=4\text{s}$ ；量测噪声为均值等于 0、方差是  $\sigma_x = \sigma_y = 0.1\text{km}$  的高斯白噪声； $\Delta\rho = 6\sigma_x$ ,  $\Delta\theta = \pi/100$ ；每周期的杂波个数服从泊松分布，通过改变参数  $\lambda$ ，可以得到不同强度的杂波。目标速度范围是：最大速度  $V_{\max}=0.7\text{km/s}$ ，最小速度  $V_{\min}=0.05\text{km/s}$ 。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  $P_d=1$ ，杂波强度  $\lambda=50$ ，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。

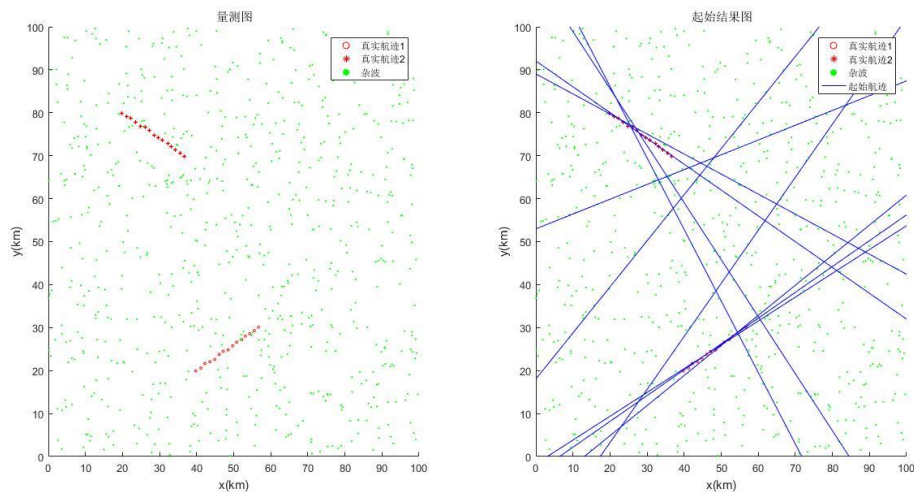


图 2-2 标准 Hough 变换航迹起始结果图

简易序列 Hough 变换仿真：检测概率  $P_d=1$ ，杂波强度  $\lambda=50$ ，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。

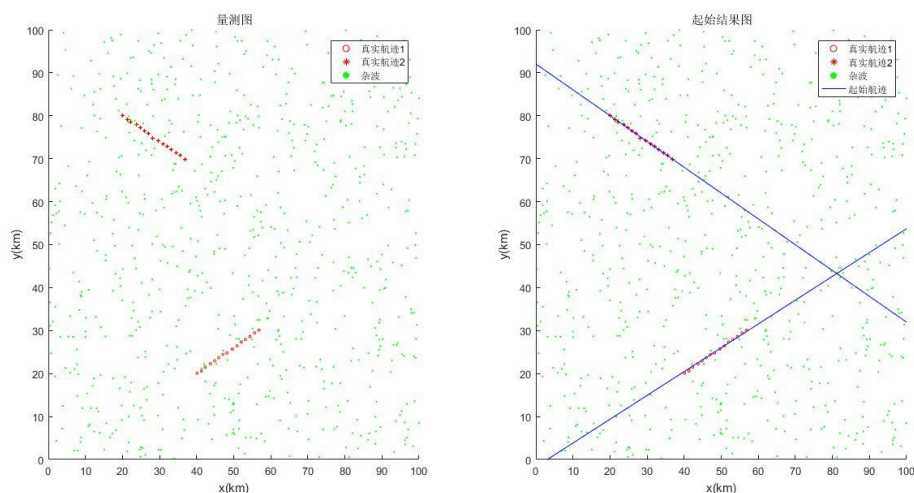


图 2-3 序列 Hough 变换航迹起始结果图

修正 Hough 变换仿真：检测概率  $P_d=1$ ，杂波强度  $\lambda=20$ ，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。

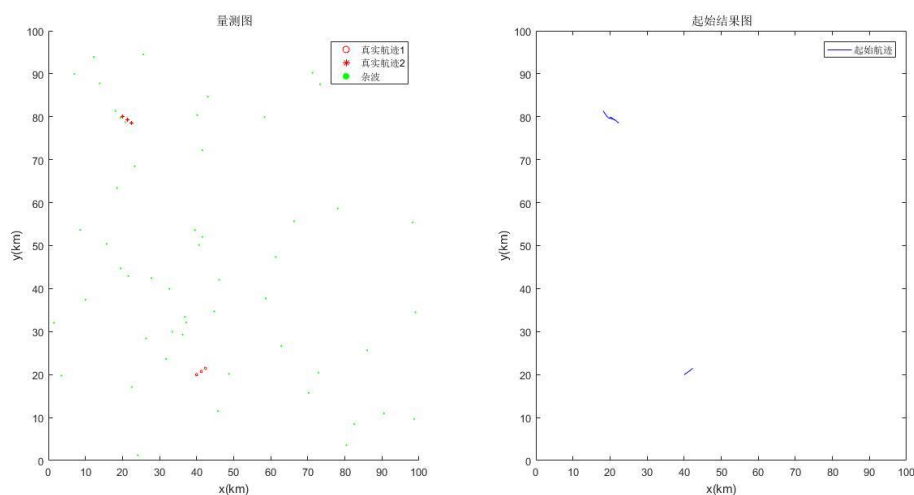


图 2-4 修正 Hough 变换航迹起始结果图

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  $S$ 。



### 3. 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。

### 4. 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

### 5. 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					