

多传感器信号融合理论及 应用结课论文

卡尔曼滤波器的基本原理及应用

姓名：连旭

学号：112081101147

专业：控制理论与控制工程

卡尔曼滤波器的基本原理及应用

一 引言

1960 年，卡尔曼发表了用递归方法解决离散数据线性滤波问题的论文，在这篇文章里，一种克服了维纳滤波缺点的新方法被提出来，这就是我们今天称之为卡尔曼滤波的方法。卡尔曼滤波应用广泛且功能强大，它可以估计信号的过去和当前状态，甚至能估计将来的状态，即使并不知道模型的确切性质。

本质上来讲，滤波就是一个信号处理与变换（去除或减弱不想要的成分，增强所需成分）的过程，这个过程既可以通过硬件来实现，也可以通过软件来实现。卡尔曼滤波属于一种软件滤波方法，其基本思想是：以最小均方误差为最佳估计准则。采用信号与噪声的状态空间模型，利用前一时刻的估计值和当前时刻的观测值来更新对状态变量的估计，求出当前时刻的估计值，算法根据建立的系统方程和观测方程对需要处理的信号做出满足最小均方误差的估计。

二 卡尔曼滤波器的介绍

为了可以更加容易的理解卡尔曼滤波器，这里会应用形象的描述方法来讲解，而不是像大多数参考书那样罗列一大堆的数学公式和数学符号。但是，他的 5 条公式是其核心内容。结合现代的计算机，其实卡尔曼的程序相当的简单，只要理解了那 5 条公式。

首先，我们先要引入一个离散控制过程的系统。该系统可用一个线性随机微分方程 (Linear Stochastic Difference equation) 来描述：

$$X(k) = AX(k-1) + BU(k) + W(k)$$

再加上系统的测量值：

$$Z(k) = HX(k) + V(k)$$

上两式子中， $X(k)$ 是 k 时刻的系统状态， $U(k)$ 是 k 时刻对系统的控制量。 A 和 B 是系统参数，对于多模型系统，他们为矩阵。 $Z(k)$ 是 k 时刻的测量值， H 是测量系统的参数，对于多测量系统， H 为矩阵。 $W(k)$ 和 $V(k)$ 分别表示过程和测量的噪声。他们被假设成高斯白噪声 (White Gaussian Noise)，他们的协方差分别是 Q ， R (这里我们假设他们不随系统状态变化而变化)。

对于满足上面的条件(线性随机微分系统, 过程和测量都是高斯白噪声), 卡尔曼滤波器是最优的信息处理器。下面我们来用他们结合他们的协方差来估算系统的最优化输出。

首先我们要利用系统的过程模型, 来预测下一状态的系统。假设现在的系统状态是 k , 根据系统的模型, 可以基于系统的上一状态而预测出现在状态:

$$X(k|k-1) = AX(k-1|k-1) + BU(k) \quad (1)$$

式(1)中 $X(k|k-1)$ 是利用上一状态预测的结果, $X(k-1|k-1)$ 是上一状态最优的结果, $U(k)$ 为现在状态的控制量, 如果没有控制量, 它可以为 0。

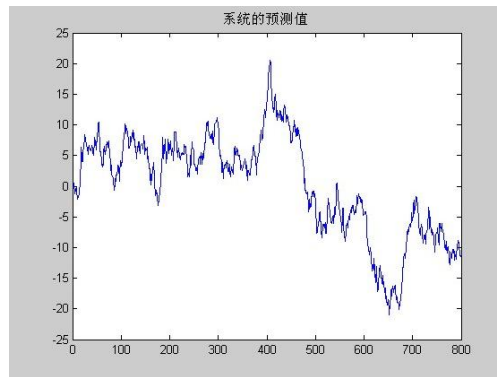
到现在为止, 我们的系统结果已经更新了, 可是, 对应于 $X(k|k-1)$ 的协方差还没更新。我们用 P 表示协方差:

$$P(k|k-1) = AP(k-1|k-1)A' + Q \quad (2)$$

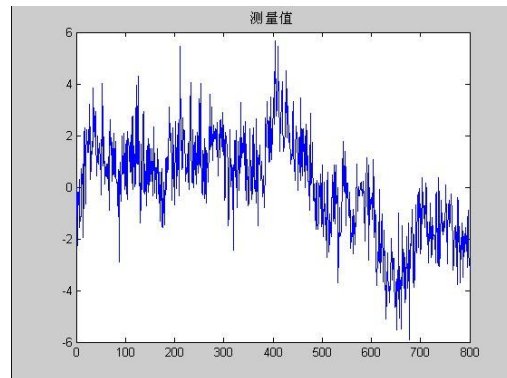
式(2)中, $P(k|k-1)$ 是 $X(k|k-1)$ 对应的协方差。

$P(k-1|k-1)$ 是 $X(k-1|k-1)$ 对应的协方差, A' 表示 A 的转置矩阵, Q 是系统过程的协方差。式子(1)(2)就是卡尔曼滤波器 5 个公式中的前两个, 也就是对系统的预测部分。

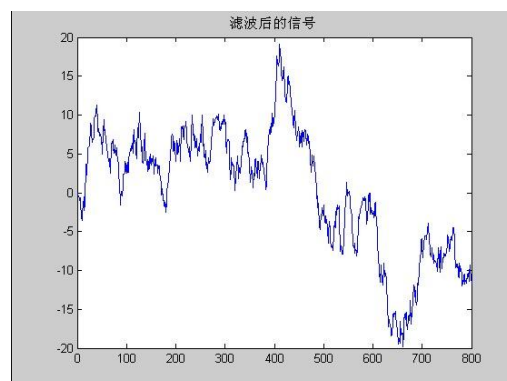
现在我们有现在状态的预测结果, 然后我们再收集现在状态的测量值。结合预测值和测量值, 我们可以得到现在状



图一 系统的预测值



图二 系统的观测值



图三 卡尔曼滤波后信号

态(k)的最优化估算值 $X(k|k)$:

$$X(k|k) = X(k|k-1) + Kg(k)(Z(k) - HX(k|k-1)) \quad (3)$$

其中 Kg 为卡尔曼增益:

$$Kg(k) = P(k|k-1)H' / (HP(k|k-1)H' + R) \quad (4)$$

到现在为止, 我们已经得到了 k 状态下最优的估算值 $X(k|k)$ 。但是为了要另卡尔曼滤波器不断的运行下去直到系统过程结束, 我们还要更新 k 状态下 $X(k|k)$ 的协方差:

$$P(k|k) = (I - Kg(k)H)P(k|k-1) \quad (5)$$

其中 I 为 I 的矩阵, 对于单模型单测量, $I=1$ 。当系统进入 $k+1$ 状态时, $P(k|k)$ 就是式子 (2) 的 $P(k-1|k-1)$ 。这样, 算法就可以自回归的运算下去。

卡尔曼滤波器的原理基本描述了, 式子 (1) (2) (3) (4) (5) 就是他的 5 个基本公式。根据这 5 个公式, 可以很容易的实现计算机的程序。

三 卡尔曼滤波器的简单举例

结合上边对于卡尔曼滤波器的算法分析, 我们举一个简单的例子进行比较直观的分析。并且采用 matlab 平台进行仿真参考。

假设我们要研究的对象是一个房间的温度。根据经验判断, 这个房间的温度是恒定的, 也就是下一分钟的温度等于现在这一分钟的温度 (假设采用一分钟来作为时间单位)。假设我们的经验不是 100% 的准确, 可能会有上下偏差几度。我们把这些偏差看成是高斯白噪声, 也就是这些偏差跟前后时间没有关系的而且符合高斯分配。另外, 我们在房间里放置了一个温度计, 但是这个温度计也不准确, 测量值会比实际值产生偏差。我们也把这些偏差看成高斯白噪声。

现在对于某一分钟我们有两个关于该房间的温度值: 根据经验的预测值 (系统地预测值) 和温度计的值 (测量值)。我们要用这两个测量值结合它们各自的噪声来估算出房间的实际温度值。

把房间看成一个系统, 然后对这个系统建模。当然, 我们见的模型不需要非常地精确。我们所知道的这个房间的温度是跟前一时刻的温度相同的, 所以 $A=1$ 。

没有控制量，所以 $U(k)=0$ 。因此可以得出：

$$X(k|k-1) = X(k-1|k-1) \quad (6)$$

式子 (2) 可以改成：

$$P(k|k-1) = P(k-1|k-1) + Q \quad (7)$$

因为测量的值是温度计的，跟温度直接对应，所以 $H=1$ 。式子 (3) (4) (5) 可以改写如下：

$$X(k|k) = X(k|k-1) + Kg(k)(Z(k) - X(k|k-1)) \quad (8)$$

$$Kg(k) = P(k|k-1)/(p(k|k-1) + R) \quad (9)$$

$$P(k|k) = (1 - Kg(k))P(k|k-1) \quad (10)$$

现在我们模拟一组测量值作为输入。假设房间的真实温度为 25 度，我们模拟了 200 个测量值，这些测量值的平均值为 25 度，但是加入了标准偏差为几度的高斯白噪声（图中用蓝线表示）。

为了令卡尔曼滤波器开始工作，我们首先给它赋予两个初始值 $X(0|0)$ 和 $P(0|0)$ ，对于 $X(0|0)$ 的值可以随意给，因为随着卡尔曼的工作， X 会收敛到正确值，但对于 $P(0|0)$ 的值最好不要给 0，因为这样可能会使卡尔曼完全相信你给定的 $X(0|0)$ 是系统最优的，从而使算法不能收敛。我们在此选定 $X(0|0)=1$ 度， $P(0|0)=1$ 。

该系统的真是温度是 25 度，图中用黑线表示。图中红线是卡尔曼滤波器输出的最优化结果（该结果在算法中设置了 $Q=1e-6$ ， $R=1e-1$ ）。

```
clear                                x(k)=a*x(k-1)+w(k-1);
N=200;                                end
w(1)=0;                                V=randn(1,N);
w=randn(1,N)                            q1=std(V);
x(1)=0;                                Rvv=q1.^2;
a=1;                                    q2=std(x);
for k=2:N;                              Rxx=q2.^2;
```

```

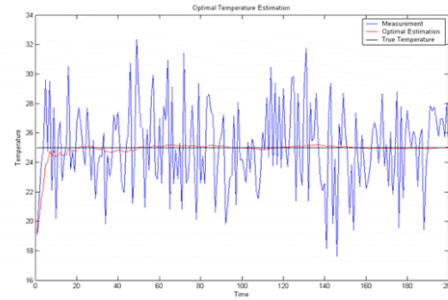
q3=std(w);
Rww=q3.^2;
c=0.2;
Y=c*x+V;
p(1)=0;
s(1)=0;
for t=2:N;
p1(t)=a.^2*p(t-1)+Rww;
b(t)=c*p1(t)/(c.^2*p1(t)+Rvv);
s(t)=a*s(t-1)+b(t)*(Y(t)-a*c*s(t-1));

```

```

p(t)=p1(t)-c*b(t)*p1(t);
end
t=1:N;
plot(t,s,'r',t,Y,'g',t,x,'b');

```



图四 卡尔曼滤波后的值的波形显示

四 卡尔曼滤波器的应用

目标跟踪问题的应用背景是雷达数据处理，即雷达在搜索到目标并记录目标的位置数据，对测量到的目标位置数据（称为点迹）进行处理，自动形成航迹，并对目标在下一时刻的位置进行预测。本节简要讨论了用 Kalman 滤波方法对单个目标航迹进行预测，并借助于 Matlab 仿真工具，对实验的效果进行评估。

4.1 情景假设

假定有一个二坐标雷达对一平面上运动的目标进行观测，目标在 $t = 0 \sim 400$ 秒沿 y 轴作恒速直线运动，运动速度为 -15 米/秒，目标的起始点为 (2000 米, 10000 米)，在 $t = 400 \sim 600$ 秒向 x 轴方向做 90° 的慢转弯，加速度均为 0.075 米/秒²，完成慢转弯后加速度将降为零，从 $t = 610$ 秒开始做 90° 的快转弯，加速度为 0.3 米/秒²，在 660 秒结束转弯，加速度降至零。雷达扫描周期 $T = 2$ 秒， x 和 y 独立地进行观测，观测噪声的标准差均为 100 米。

4.2 卡尔曼滤波算法分析

为了简单起见，仅对 x 轴方向进行考虑。

首先，目标运动沿 x 轴方向的运动可以用下面的状态方程描述：

$$\begin{aligned}x(k+1) &= x(k) + T\dot{x}(k) + (T^2/2)u_x(k) \\ \dot{x}(k+1) &= \dot{x}(k) + Tu_x(k)\end{aligned}\quad (11)$$

用矩阵的形式表述为,

$$X(k+1) = \Phi X(k) + \Gamma W(k) \quad (12)$$

在上式中, $X(k) = \begin{bmatrix} x(k) \\ \dot{x}(k) \end{bmatrix}$, $\Phi = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}$, $\Gamma = \begin{bmatrix} \frac{1}{2}T^2 \\ T \end{bmatrix}$, $W(k) = u_x$ 。

考虑雷达的观测, 得出观测方程为:

$$Z(k) = C(k)X(k) + V(k) \quad (13)$$

在 (13) 中, $C(k) = [1 \ 0]$, $V(k)$ 为零均值的噪声序列, 方差已知。

对目标进行预测, 由相关理论可得到下面的迭代式:

$$\hat{X}(k/k-1) = \Phi \hat{X}(k-1/k-1) \quad (14)$$

在 (14) 中, $\hat{X}(k/k-1) = E[X(k)|Z^{k-1}]$, 反映了由前 $k-1$ 各观测值对目前状态的估计。

而预测的误差协方差可由下式表出,

$$P_{\hat{X}}(k/k-1) = \Phi P_{\hat{X}}(k-1/k-1) \Phi^T + \Gamma Q(k-1) \Gamma^T \quad (15)$$

对于最佳滤波, 迭代表达式为:

$$\hat{X}(k/k) = \hat{X}(k/k-1) + K(k)[Z(k) - C(k)\hat{X}(k/k-1)] \quad (16)$$

在式 (16) 中, $K(k)$ 为 Kalman 增益。

而滤波误差的协方差为,

$$P_{\hat{X}}(k/k) = [I - K(k)C(k)]P_{\hat{X}}(k/k-1) \quad (17)$$

在应用上面的公式进行 Kalman 滤波时, 需要指定初值。由于实际中通常无法得到目标的初始状态, 我们可以利用前几个观测值建立状态的初始估计, 比如采用前两个观测值,

$$\hat{X}(2/2) = \begin{bmatrix} z_x(2) & [z_x(2) - z_x(1)]/T \end{bmatrix}^T \quad (18)$$

此时, 估计误差为

$$\tilde{X}(2/2) = \left[-v_x(2) \quad \frac{T}{2} \cdot u_x(1) + \frac{v_x(1) - v_x(2)}{T} \right]^T \quad (19)$$

而误差协方差矩阵为,

$$P_{\tilde{X}\tilde{X}}(2/2) = \begin{bmatrix} \sigma_x^2 & \sigma_x^2/T \\ \sigma_x^2/T & 2\sigma_x^2/T \end{bmatrix} \quad (20)$$

4.3 仿真计算与结果

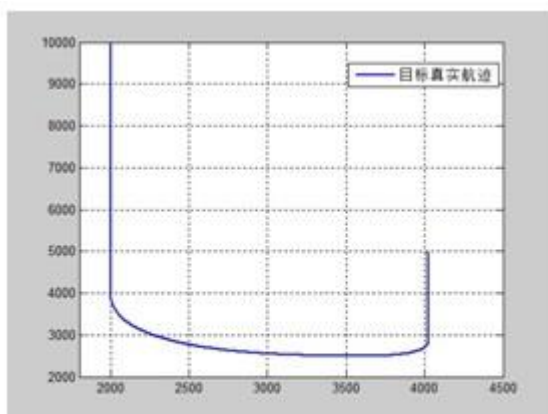
(1) 首先给出原始轨迹 (matlab 代码如下):

```
function [X,Y]=trajectory(Ts,offtime)
% 产生真实航迹[X,Y],并在直角坐标系下显示出
% Ts为雷达扫描周期,每隔Ts秒取一个观测数据
% 最初做匀速运动,接下来进行两个90度的机动转弯
if nargin>2
    error('输入的变量过多,请检查');
end
if offtime<600
    error('仿真时间必须大于600s,请重新输入');
end
x=zeros(offtime,1);
y=zeros(offtime,1);
X=zeros(ceil(offtime/Ts),1);
Y=zeros(ceil(offtime/Ts),1);
% t=0:400s,速度vx,vy为沿x和y轴的速度分量(m/s)
x0=2000;%起始点坐标
y0=10000;
vx=0;
vy=-15;%沿-y方向
for t=1:400
    x(t)=x0+vx*t;
    y(t)=y0+vy*t;
end
% t=400:600s,ax,ay为沿x和y轴的加速度分量(m/s/s)
ax=0.075;
ay=0.075;
```

```
for t=0:200
    x(t+401)=x(400)+vx*t+ax*t*t/2;
    y(t+401)=y(400)+vy*t+ay*t*t/2;
end
vx=vx+ax*200;%第一次机动转弯结束时的速度
vy=vy+ay*200;
% t=600:610s匀速运动
for t=0:10
    x(t+601)=x(601)+vx*t;
    y(t+601)=y(601)+vy*t;
end
% t=610:660s,第二次转弯
ax=-0.3;
ay=0.3;
for t=0:50
    x(t+611)=x(611)+vx*t+ax*t*t/2;
    y(t+611)=y(611)+vy*t+ay*t*t/2;
end
vx=vx+ax*(660-610);%第二次机动转弯结束时的速度
vy=vy+ay*(660-610);
% 660s以后匀速运动,一直到截止时间
for t=0:(offtime-660)
    x(t+661)=x(661)+vx*t;
    y(t+661)=y(661)+vy*t;
end
% 得到雷达的观测数据
for n=0:Ts:offtime
    X(n/Ts+1)=x(n+1);
    Y(n/Ts+1)=y(n+1);
end
%显示真实轨迹
```



```
plot(X, Y, 'LineWidth', 2), axis([1800 4500
2000 10000]), grid on;
legend('目标真实航迹');
```



图五 目标真实航迹

(2) 然后给出观测值与理论值的轨迹
(matlab 代码如下):

```
function
XE=Kalman_filter(Ts, offtime, d, Flag)
% Kalman_filter          采用Kalman
滤波方法, 从观测数值中得到航迹的估计
% XE                    输出x轴方向
上的误差
% Ts                    采样时间, 即
雷达工作周期
% offtime               仿真截止时
间
% d                    噪声的标准
差值
% Flag                 判断计算x轴
或y轴数据, '0' --x, '1' --y
if nargin>4
    error(' 输入的变量过多, 请检查');
end
if offtime<600
    error(' 仿真时间必须大于600s, 请重新
输入');
end
Pv=d*d; % 噪声的功率
N=ceil(offtime/Ts); % 采样点数
sigma=10;% 加速度方向的的扰动
switch Flag
    case 0
        a=[zeros(1, 400)
0.075*ones(1, 200) zeros(1, 10)
-0.3*ones(1, 50) zeros(1, offtime-660)]; %
对不同时期的加速度进行描述
        case 1
            a=[zeros(1, 400)
0.075*ones(1, 200) zeros(1, 10)
0.3*ones(1, 50) zeros(1, offtime-660)];
        otherwise
            error(' 输入仅能为0或1');
        end
% 定义系统的状态方程
Phi=[1, Ts; 0, 1];
Gamma=[Ts*Ts/2; Ts];
C=[1 0];
R=Pv;
Q=sigma^2; W=[];
randn('state', sum(100*clock)); % 设置随
机数发生器
for n=0:Ts:offtime-1
    W(n/Ts+1)=a(n+1)+sigma*randn(1, 1);
end
Xest=zeros(2, 1); % 用前k-1时刻的输出值
估计k时刻的预测值
Xfli=zeros(2, 1); % k时刻Kalman滤波器的输
出值
Xes=zeros(2, 1); % 预测输出误差
Xef=zeros(2, 1); % 滤波后输出的误差
Pxe=zeros(2, 1); % 预测输出误差均方差矩
Px=zeros(2, 1); % 滤波输出误差均方差矩
XE=zeros(1, N); % 得到最终的滤波输出值,
仅仅考虑距离分量
[x, y]=trajectory(Ts, offtime); % 产生理
论的航迹
for i=1:N
    vx(i)=d*randn(1); % 观测噪声, 两者独
立
    vy(i)=d*randn(1);
    zx(i)=x(i)+vx(i); % 实际观测值
    zy(i)=y(i)+vy(i);
end
switch Flag
    case 0
        Xfli=[zx(2) (zx(2)-zx(1))/Ts]'; %
```

利用前两个观测值来对初始条件进行估计

```
Xef=[-vx(2)
Ts*W(1)/2+(vx(1)-vx(2))/Ts]';
Px=[Pv,Pv/Ts;Pv/Ts,2*Pv/Ts+Ts*Ts*Q/4];
for k=3:N
Xest=Phi*Xfli; % 更新该时刻的预测值
Xes=Phi*Xef+Gamma*W(k-1); % 预测输出误差
Pxe=Phi*Px*Phi'+Gamma*Q*Gamma'; % 预测误差的协方差阵
K=Pxe*C'*inv(C*Pxe*C'+R); %
```

Kalman滤波增益

```
Xfli=Xest+K*(zx(k)-C*Xest);
Xef=(eye(2)-K*C)*Xes-K*vx(k);
Px=(eye(2)-K*C)*Pxe;
XE(k)=Xfli(1,1);
end
XE(1)=zx(1);XE(2)=zx(2);
```

case 1

利用前两个观测值来对初始条件进行估计

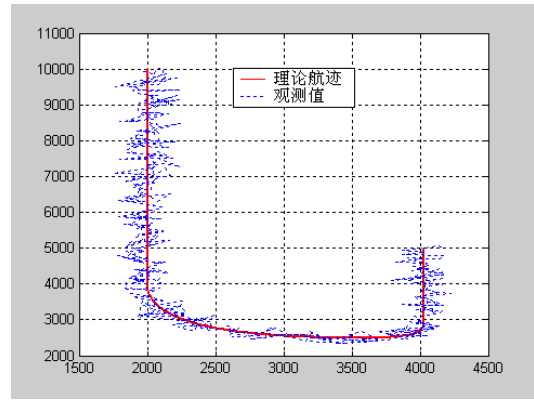
```
Xef=[-vy(2)
Ts*W(1)/2+(vy(1)-vy(2))/Ts]';
Px=[Pv,Pv/Ts;Pv/Ts,2*Pv/Ts+Ts*Ts*Q/4];
for k=3:N
Xest=Phi*Xfli; % 更新该时刻的预测值
Xes=Phi*Xef+Gamma*W(k-1); % 预测输出误差
Pxe=Phi*Px*Phi'+Gamma*Q*Gamma'; % 预测误差的协方差阵
K=Pxe*C'*inv(C*Pxe*C'+R); %
```

Kalman滤波增益

```
Xfli=Xest+K*(zy(k)-C*Xest);
Xef=(eye(2)-K*C)*Xes-K*vy(k);
Px=(eye(2)-K*C)*Pxe;
XE(k)=Xfli(1,1);
end
XE(1)=zy(1);XE(2)=zy(2);
```

otherwise

```
error('False input nargin');
end
```



图六 目标观测与真实航迹

(3) 最后给出卡尔曼滤波后与理论值的轨迹 (matlab 代码如下):

```
function
[XER, YER]=filter_result(Ts, mon, d)
% filter_result 对观测数据进行卡尔曼滤波, 得到预测的航迹以及估计误差的均值和标准差
% Ts 采样时间, 即雷达的工作周期
% mon 进行Monte-Carlo仿真的次数
% d 测量的误差, 单位
if nargin>3
error(' Too many input arguments. ');
end
offtime=800;
% 产生理论的航迹
[x, y]=trajectory(Ts, offtime);
Pv=d*d;
N=ceil(offtime/Ts);
randn('state', sum(100*clock)); % 设置随机数发生器
for i=1:N
vx(i)=d*randn(1); % 观测噪声, 两者独立
vy(i)=d*randn(1);
zx(i)=x(i)+vx(i); % 实际观测值
zy(i)=y(i)+vy(i);
end
% 产生观测数据
for n=1:mon
% 用卡尔曼滤波得到估计的航迹
```

```

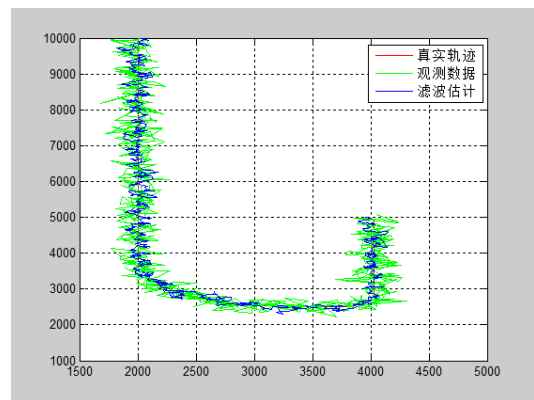
XE=Kalman_filter(Ts, offtime, d, 0);
YE=Kalman_filter(Ts, offtime, d, 1);
%误差矩阵
XER(1:N, n)=x(1:N)-(XE(1:N))';
YER(1:N, n)=y(1:N)-(YE(1:N))';
end
%滤波误差的均值
XERB=mean(XER, 2);
YERB=mean(YER, 2);
%滤波误差的标准差
XSTD=std(XER, 1, 2); % 计算有偏的估计值,
flag='1'
YSTD=std(YER, 1, 2);
%作图
figure
plot(x, y, 'r');hold on;
plot(zx, zy, 'g');hold on;
plot(XE, YE, 'b');hold off;
axis([1500 5000 1000 10000]),grid on;
legend('真实轨迹','观测数据','滤波估计');
figure
subplot(2, 2, 1)
plot(XERB)
axis([0 500 -50 50])
xlabel('观测次数')
ylabel('X方向滤波误差均值'),grid on;
subplot(2, 2, 2)
plot(YERB)
axis([0 500 -50 50])

```

```

xlabel('观测次数')
ylabel('Y方向滤波误差均值'),grid on;
subplot(2, 2, 3)
plot(XSTD)
axis([0 500 0 150])
xlabel('观测次数')
ylabel('X方向滤波误差标准值'),grid on;
subplot(2, 2, 4)
plot(YSTD)
axis([0 500 0 150])
xlabel('观测次数')
ylabel('Y方向滤波误差标准值'),grid on;
X=X ER;Y=YER;

```



图七 卡尔曼滤波与真实航迹

经过上面的仿真分析，可以看出 Kalman 滤波算法对于动态目标的跟踪有着比较好的效果，而且可以较好地抑止环境中的噪声影响。

参考文献:

-
- [1] 张开禾, 富立, 范耀祖. 基于卡尔曼滤波的信息融合算法优化研究[J]. 中国惯性技术学报. 2006 (02)
 - [2] 张怡, 周桃云, 杨晓亚. 一种基于信息融合的滤波算法研究[J]. 电子测量技术. 2007(02)
 - [3] 杜云峰. 卡尔曼滤波器在过程估计中的应用[J]. 机电产品开发与创新. 2007(04)
 - [4] 杨兆升, 冯金巧, 张林. 基于卡尔曼滤波的交通信息融合方法[J]. 吉林大学学报(工学版). 2007(05)