

参考如下：

[搜索及追踪雷达](#)

[B站卡尔曼滤波器的原理以及在matlab中的实现](#)

[卡尔曼滤波的原理、理解与仿真](#)

[从放弃到精通！卡尔曼滤波从理论到实践~](#)

[匀加速直线运动的卡尔曼滤波推导及在matlab中的实现](#)

[kalman滤波理解三：协方差矩阵的计算](#)

[图解卡尔曼滤波器，无需深厚的数学知识也易懂](#)

一、发展背景与目的

搜索及追踪雷达是一种在连续跟踪目标的同时，还继续对空间进行搜索的雷达。它在传统搜索雷达的基础上，借助计算机，能够实现多目标的快速跟踪。要跟踪目标，首先要找出目标的位置参量和运动参量，即要找出一种算法，利用目标的运动规律，对雷达测量数据进行滤波和外推处理，求出目标参量，实现多个目标的航迹(运动轨迹)跟踪。

就是根据雷达录取设备提供的目标点迹参数来建立和更新轨迹，并外推下一次天线扫描周期目标出现的位置。为了解决机动目标的跟踪问题，许多学者对此进行了研究，并提出了许多跟踪算法，这些算法在特定条件下对目标机动问题都能进行比较准确的描述。但当条件发生变化时，这些算法的跟踪性能会有不同程度的降低。同时计算量偏大，实施比较困难。

3. 宏观意义：滤波即加权

理想状态：信号 $\times 1$ + 噪声 $\times 0$

低 1

高 0

低通滤波

估计 $\cdot ()$

观测 $\cdot ()$

卡尔曼滤波

修正

二、跟踪算法

跟踪滤波的目的就是根据雷达录取设备提供的目标点迹参数来建立和更新轨迹，并外推下一次天线扫描周期目标出现的位置。为了解决机动目标的跟踪问题，许多学者对此进行了研究，并提出了许多跟踪算法，这些算法在特定条件下对目标机动问题都能进行比较准确的描述。但当条件发生变化时，这些算法的跟踪性能会有不同程度的降低。同时计算量偏大，实施比较困难。

Kalman滤波算法

假设你有两个传感器，测的是同一个信号。可是它们每次的读数都不太一样，怎么办？

取平均。

再假设你知道其中贵的那个传感器应该准一些，便宜的那个应该差一些。那有比取平均更好的办法吗？

加权平均。

怎么加权？假设两个传感器的误差都符合正态分布，假设你知道这两个正态分布的方差，用这两个方差值，（此处省略若干数学公式），你可以得到一个“最优”的权重。

接下来，重点来了：假设你只有一个传感器，但是你还有一个数学模型。模型可以帮你算出一个值，但也不是那么准。怎么办？

把模型算出来的值，和传感器测出的值，（就像两个传感器那样），取加权平均。

OK，最后一点说明：你的模型其实只是一个步长的，也就是说，知道 $x(k)$ ，我可以求 $x(k+1)$ 。问题是 $x(k)$ 是多少呢？答案： $x(k)$ 就是你上一步卡尔曼滤波得到的、所谓加权平均之后的那个、对 x 在 k 时刻的最佳估计值。

于是迭代也有了。

这就是卡尔曼滤波。

可以在任何含有不确定信息的动态系统中使用卡尔曼滤波，对系统下一步的走向做出有根据的预测，即使伴随着各种干扰，卡尔曼滤波总是能指出真实发生的情况。在连续变化的系统中使用卡尔曼滤波是非常理想的，它具有占用内存小的优点（除了前一个状态量外，不需要保留其它历史数据），并且速度很快，很适合应用于实时问题和嵌入式系统。

卡尔曼滤波是一种利用线性系统状态方程，通过系统输入输出观测数据，对系统状态进行最优估计的算法。

由于观测数据中包括系统中的噪声和干扰的影响，所以最优估计也可看作是滤波过程。最核心的意思就是卡尔曼滤波可以很好地从带有噪声的数据过程中估计状态。

二. 进阶☆☆

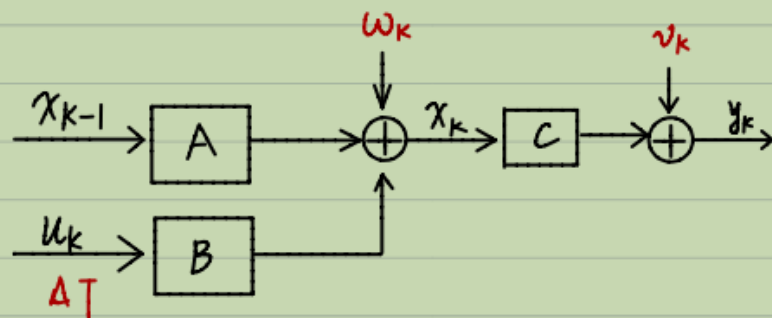
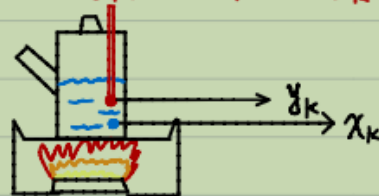
1. 状态空间表达式

状态方程 $x_k = Ax_{k-1} + Bu_k + w_k$

$$x_k = x_{k-1} + \Delta T + w_k$$

观测方程 $y_k = Cx_k + v_k$

$$y_k = x_k + v_k$$



w_k : 过程噪声

v_k : 测量噪声

(2) 参数分析

I. $w_k \in N(0; Q_k)$

$v_k \in N(0; R_k)$

exp: V_k, R_k 定义
GPS \rightarrow Position

w_k, Q_k 定义

$$\underline{\underline{v + v_a}}$$

1000m \pm 8m 噪声

$$V_k = 8m$$

5m/s \pm 5m/s $w_k = 8m/s$

方差为 1m 噪声

$$R_k = 1m$$

$S \in N(0, 1) Q_k = H S$
1m/s

实例：匀速直线运动线性问题

(1) 预测

假设有一辆小车，其在 t 时刻的位置为 p_t (假设其在一维直线上运动，则位置可以用数轴上的点表示)，速度为 v_t

因此在 t 时刻小车的状态可用向量表示为如下。


$$\text{状态 } x_t = \begin{bmatrix} p_t \\ v_t \end{bmatrix}$$

假设由于油门的设置或控制命令，我们知道了期望的加速度为 u_t (加速度理解为外部的控制量)，则可由运动学公式从 $t-1$ 时刻推出其在 t 时刻的速度与位置如下：

①状态预测公式(先验估值):

$$p_t = 1 \times p_{t-1} + \Delta t \times v_{t-1} + u_t \times \frac{\Delta t^2}{2}$$
$$v_t = 0 \times p_{t-1} + 1 \times v_{t-1} + u_t \times \Delta t$$

状态转移



状态 $x_t = \begin{bmatrix} p_t \\ v_t \end{bmatrix}$

$$p_t = p_{t-1} + v_{t-1} \times \Delta t + u_t \times \frac{\Delta t^2}{2}$$
$$v_t = v_{t-1} + u_t \times \Delta t$$
$$\begin{bmatrix} p_t \\ v_t \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_{t-1} \\ v_{t-1} \end{bmatrix} + \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix} u_t$$
$$F_t = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, B_t = \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix}$$
$$\hat{x}_t^- = F_t \hat{x}_{t-1} + B_t u_t$$

矩阵 F_t 为状态转移矩阵，表示如何从上一状态来推测当前时刻的状态；

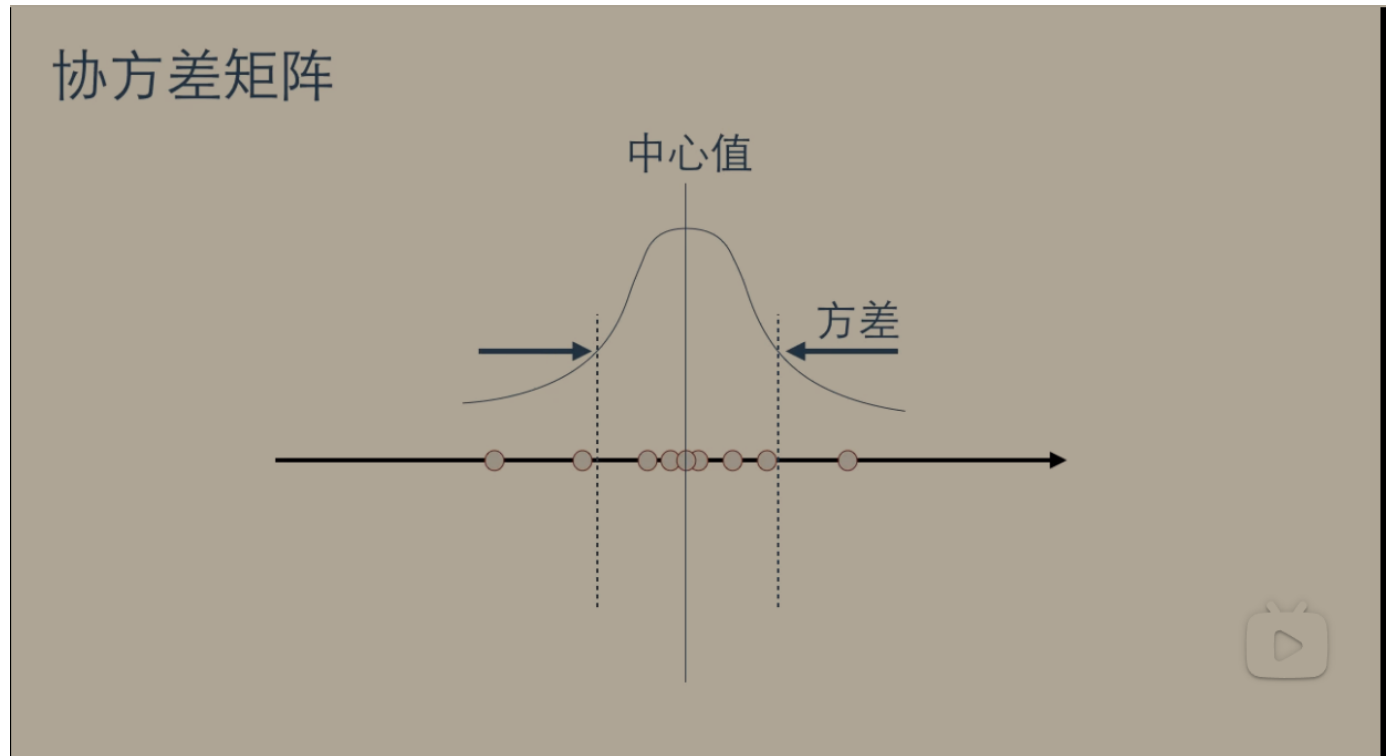
B_t 为控制矩阵，表示控制量 u_t 如何作用于当前矩阵；

带^帽子的表示估计值,并不是最优的。

带 \hat{x} 号的表示先验值 (根据当前卡尔曼滤波值通过理论公式推导下一时刻可能的值)

到这里，我们已经得到了t时刻的状态预测，但这里只有状态预测的均值（即 \hat{x}_t ）。我们是基于高斯分布来建立状态变量的，因此还需要协方差。（即一个多维高斯分布由均值【中心值】和协方差决定）

一维矩阵↓



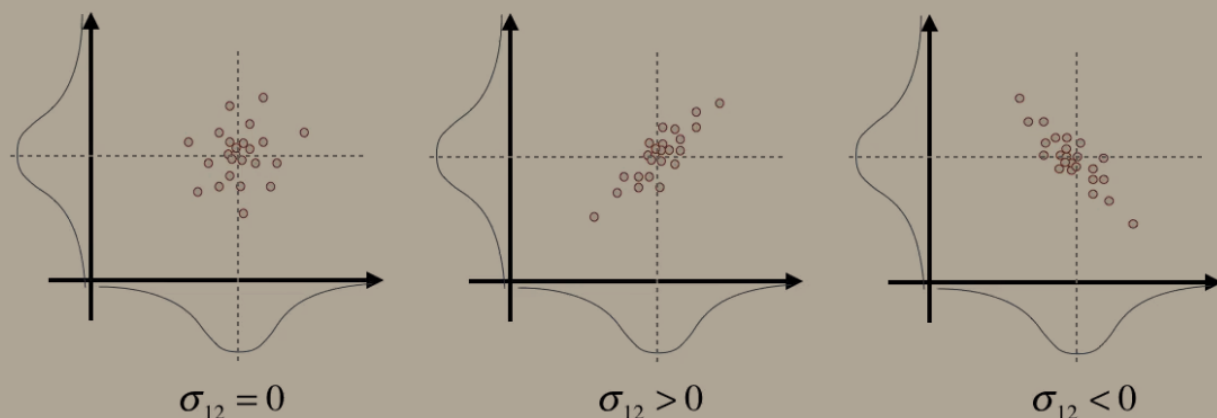
假设P表示预测协方差 $P = \text{cov}(x, x)$ ，二维矩阵↓

对角线上的值是两个维度上的方差

反对角线上的值是他们的协方差

就是 x ， x 是向量，有两个分量

协方差矩阵



$$\text{cov}(x, x) = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{bmatrix}$$



那么加入状态转换矩阵 F_t 后有

因为 P 是状态矩阵的标准差矩阵 $[p_t \ v_t]$ 乘以它的转置，所以方差也需要左右都乘以状态转移矩阵

滤波,平滑噪声曲线?

状态协方差矩阵 P ，对角线元素是各个状态的方差，其余元素是相应元素的协方差，由此也能看出 P 是一个多维方阵，维度和状态数一致，且 P 是对称方阵。

$$P_t = \text{cov}(Fx_{t-1}, Fx_{t-1}) = F \text{cov}(x_{t-1}, x_{t-1}) F^T = FP_{t-1}F^T$$

在使用时协方差矩阵 P 是一个迭代更新的量，每一轮预测和更新后， P 都会更新一个新的值，因此初始化时可以根据估计协定，不用太苛求初始化的精准度，因为随着几轮迭代会越来越趋紧真实值。

在每次预测之后，我们可以添加一些新的不确定性来建立这种与“外界”（即我们没有跟踪的干扰）之间的不确定性模型。我们将这些没有被跟踪的干扰当作协方差为 Q 的噪声来处理。最终

②过程噪声协方差矩阵（先验估值协方差）：

$$P_t^- = F P_{t-1} F^T + Q(\text{噪声})$$

*协方差矩阵的性质： $\text{cov}(Ax, Bx) = A \text{cov}(x, x) B^T$

Q过程噪声的协方差矩阵

该矩阵是由不确定的噪声引起的，确定Q的各元素大小是不容易的，使用时都是具体问题具体分析，比方说针对上文中的小车的状态，误差来源是移动中的打滑等底层的是由于力的变化导致的加速度的变化，最底层的是由于力的变化导致的加速度的变化，因此我们找到了加速度方差，就可以推导出Q矩阵，假设加速度方差是D(a)。

求以加速度为基准产生的高斯白噪声误差协方差矩阵，

$$\Delta v = v_0 + at$$

根据，
$$\Delta p = v_0 t + \frac{1}{2} at^2$$

$$D(v) = D(at) = t^2 D(a)$$

$$D(p) = D\left(\frac{1}{2} at^2\right) = \frac{1}{4} t^4 D(a)$$

得到，
$$\text{cov}(p, v) = \sqrt{D(v)} * \sqrt{D(p)} = \frac{1}{2} t^3 D(a)$$

$$Q = \begin{bmatrix} \frac{1}{4} t^4 & \frac{1}{2} t^3 \\ \frac{1}{2} t^3 & t^2 \end{bmatrix} D(a)$$

因此，

Q的初始值根据状态变化速度给，状态变化比较快的你就给大点，变化速度慢的就给小点。


(2) 更新(观测)

卡尔曼滤波的一大优点就是能处理传感器噪声，换句话说，我们的传感器或多或少都有点不可靠，并且原始估计中的每个状态可以和一定范围内的传感器读数对应起来。从测量到的传感器数据中，我们大致能猜到系统当前处于什么状态。但是由于存在不确定性，某些状态可能比我们得到的读数更接近真实状态。我们将这种不确定性（例如：传感器噪声）用更新协方差 R_t 表示，该分布的均值就是我们读取到的传感器数据，称之为 z_t 。

传感器读取的数据的单位和尺度有可能与我们要跟踪的状态的单位和尺度不一样，我们用H来表示传感器的数据。

③测量方程：

观察矩阵


$$x_t = \begin{bmatrix} p_t \\ v_t \end{bmatrix}$$
$$H = \begin{bmatrix} 1 & 0 \end{bmatrix}$$
$$z_t = Hx_t + v$$

观测噪声的协方差矩阵： R

H是用来将 x_t 与 z_t 统一维度的，这个例子里不需要 v_t ，所以H设为[1, 0]

其中H为观测矩阵，v为观测噪声。

因为观测的是距离值，只能去确定 P_t （小车移动的距离）；因为传感器只能测距，没法测速

测量噪声协方差矩阵R

测量噪声协方差的来源是传感器误差，即传感器的不准确性，在使用时一般传感器都会给出精度指标，该指标就可以直接转化到矩阵R中，还以上文小车为例，假设测量矩阵为下式所示：

$$H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

表示状态X中的位置p和速度v均可以测量，观测误差协方差矩阵R用下式表示：

$$\mathbf{R} = \begin{bmatrix} \sum p_z p_z & \sum p_z v_z \\ \sum p_z v_z & \sum v_z v_z \end{bmatrix}$$

此时对角线上就是各测量量的测量方差，从传感器说明书上均可以读到，另外针对状态之间的协方差吐过确定不了就可以设为0，即可以如下设置：

$$\mathbf{R} = \begin{bmatrix} D_z(p) & 0 \\ 0 & D_z(v) \end{bmatrix}$$

e.g.

```
//设置观测噪声方差R（超参数，可以根据点云效果修改，点云越准，R的值越小）
R[0] = 0.5F*0.5F;           //距离观测噪声 这里就是方差 0.5米平方
R[4] = (1.0f*3.1415926f/180.0f); //角度噪声 这里就是方差 1°
R[8] = 0.25F*0.25F;         //速度误差 这里就是方差 0.25米/s 平方
```

现在我们有了两个高斯分布，一个是在预测值附近，一个是在传感器读数附近。我们必须在'预测值'和'传感器测量值'之间找到最优解。我们只需将这两个高斯分布(前一状态的预测以及传感器的测量)相乘就可以了。

以两个一维高斯分布为例，其融合高斯分布为第一个式子（修正估计）；

带入状态预测和传感器测量协方差可得到第二个式子（卡尔曼增益）；

④ (修正估计)

⑤卡尔曼增益

状态更新

$$\hat{x}_t = \hat{x}_t^- + K_t(z_t - H\hat{x}_t^-)$$

$$K_t = P_t^- H^T (H P_t^- H^T + R)^{-1}$$

k表示卡尔曼增益

矩阵理论里用"I"代表单位矩阵

K的作用：

(1)K权衡预测协方差 P_t 和观察协方差矩阵 R_t 哪个更加重要。相信预测，则残差的权重小；相信观察，则残差权重大（由 K 的表达式可推出这个结论）。

(2)将残差的表现形式从观察域转换到状态域（残差与一个标量，通过K转换为向量），由状态 X的更新公式可得到该结论）

最优估计值=先验值+增益*(测量值-先验值)。

残差=测量值-先验值

⑥更新后的后验估计协方差 P_t

预测

$$\hat{x}_t^- = F\hat{x}_{t-1} + Bu_{t-1}$$

$$P_t^- = FP_{t-1}F^T + Q$$

更新

$$K_t = P_t^- H^T (HP_t^- H^T + R)^{-1}$$

$$\hat{x}_t = \hat{x}_t^- + K_t(z_t - H\hat{x}_t^-)$$

$$P_t = (I - K_t H)P_t^-$$

F和H矩阵取1

2. 调节超参数

(1) Q与R的取值

I、公式层面理解

$$\left. \begin{array}{l} P_t^- = FP_{t-1}F^T + Q \\ K = \frac{P_t^- H^T}{HP_t^- H^T + R} \end{array} \right\} \xrightarrow{\text{化简}} K = \frac{P_{t-1} + Q}{P_{t-1} + Q + R}$$

$$\hat{x}_t = \hat{x}_t^- + K(\hat{z}_t - H\hat{x}_t^-)$$

① 分析Q

② 分析R

II、其他层面理解

① 分析Q

② 分析R

(2) P_0 与 \hat{x}_0 的取值

习惯取 $\hat{x}_0 = 0$, P 往小的取, 方便收敛 (一般取 1)
(不可为 0)

3. 卡尔曼滤波的使用

(1) 选择状态量、观测量

(2) 构建方程

(3) 初始化参数

(4) 代入公式迭代

(5) 调节超参数

MATLAB代码:

```
%卡尔曼滤波(小车[速度,位置]例子)观测数据多
clc,clear,close all
%% 传感器观测
%-----
%Z = H * X + v
%X为t-1时刻实际状态
%Z为t-1时刻实际观测数据
%H为测量系统的参数,即观察矩阵
%v为观测噪声,其协方差矩阵为R
%-----
Z=(1:2:200); %理想观测值 (汽车的位置,也就是我们要修改的量),设定变化是1:2:200, 则速度就是2
noise=randn(1,100); %在理想观测值上叠加方差为1的高斯噪声
Z=Z+noise;%模拟的实际观测值
H=[1,0];%传感器提供的观测矩阵
R=1;%传感器的观测噪声协方差矩阵
%% 初始状态(均值)和状态协方差
%基于高斯分布建立状态变量需要均值和协方差(即X和P)
X=[0;0]; %初始状态 X=[位置;速度]
P=[1 0;0 1]; %状态协方差矩阵
%% 状态转移矩阵(表示如何由上一状态推测当前状态),它同时作用与X和P来预测下一时刻的X和P
F=[1 1;0 1]; %在速度例题中为[1 delta(t);0 1]
%% 外部干扰用状态转移协方差矩阵Q表示
Q=[0.0001,0;0 , 0.0001];
%% 卡尔曼滤波
figure;
hold on;
for i = 1:length(Z) %迭代次数
%% 预测
```

```

X_ = F*X;%基于上一状态预测当前状态  X_为t时刻状态预测(这里没有控制)
P_ = F*P*F'+Q;%更新协方差  Q系统过程的协方差
%% 计算卡尔曼增益
K = P_*H'/(H*P_*H'+R);
%% 更新
X = X_+K*(Z(i)-H*X_);% 得到当前状态的最优化估算值  增益乘以残差
P = (eye(2)-K*H)*P_;%更新K状态的协方差
%% 绘图
set(0,'defaultfigurecolor','w')
scatter(X(1),X(2));
xlabel('位置'),ylabel('速度')
grid on
%在代码中，我们设定x的变化是1:2:200，则速度就是2
%由上图看到，值经过几次迭代，速度就基本上在 2 附近摆动，摆动的原因是我们加入了噪声。
end

```

%在代码中，我们设定x的变化是1:2:200，则速度就是2 %由上图看到，值经过几次迭代，速度就基本上在 2 附近摆动，摆动的原因是我们加入了噪声。

