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

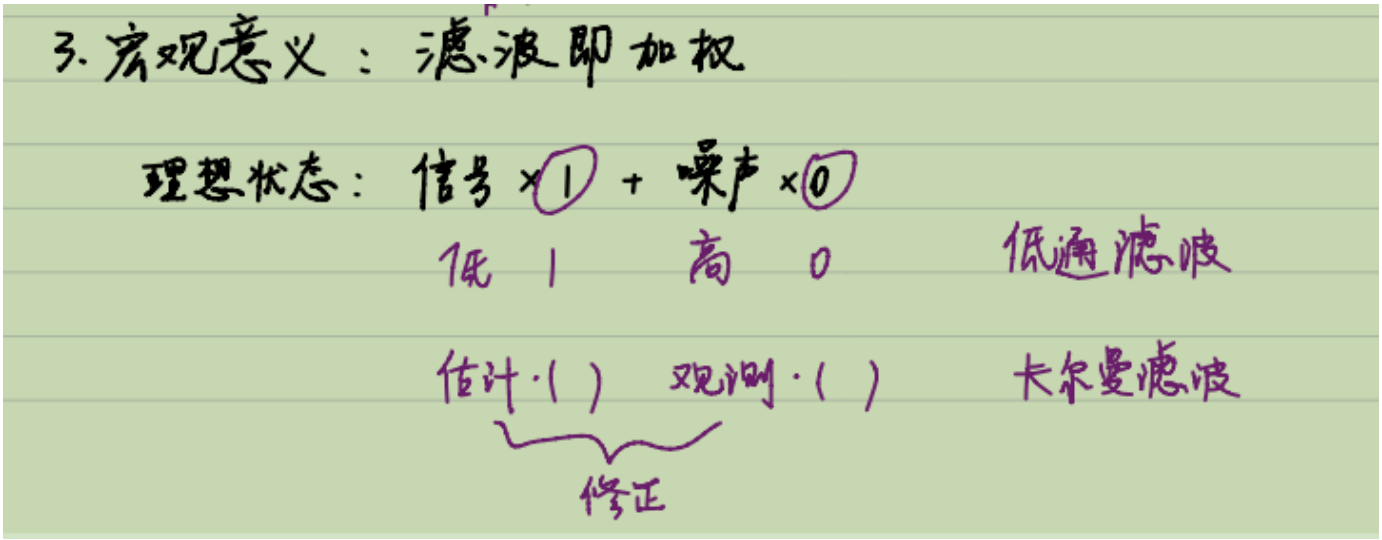
从放弃到精通！卡尔曼滤波从理论到实践~

匀加速直线运动的卡尔曼滤波推导及在matlab中的实现

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。



二、跟踪算法

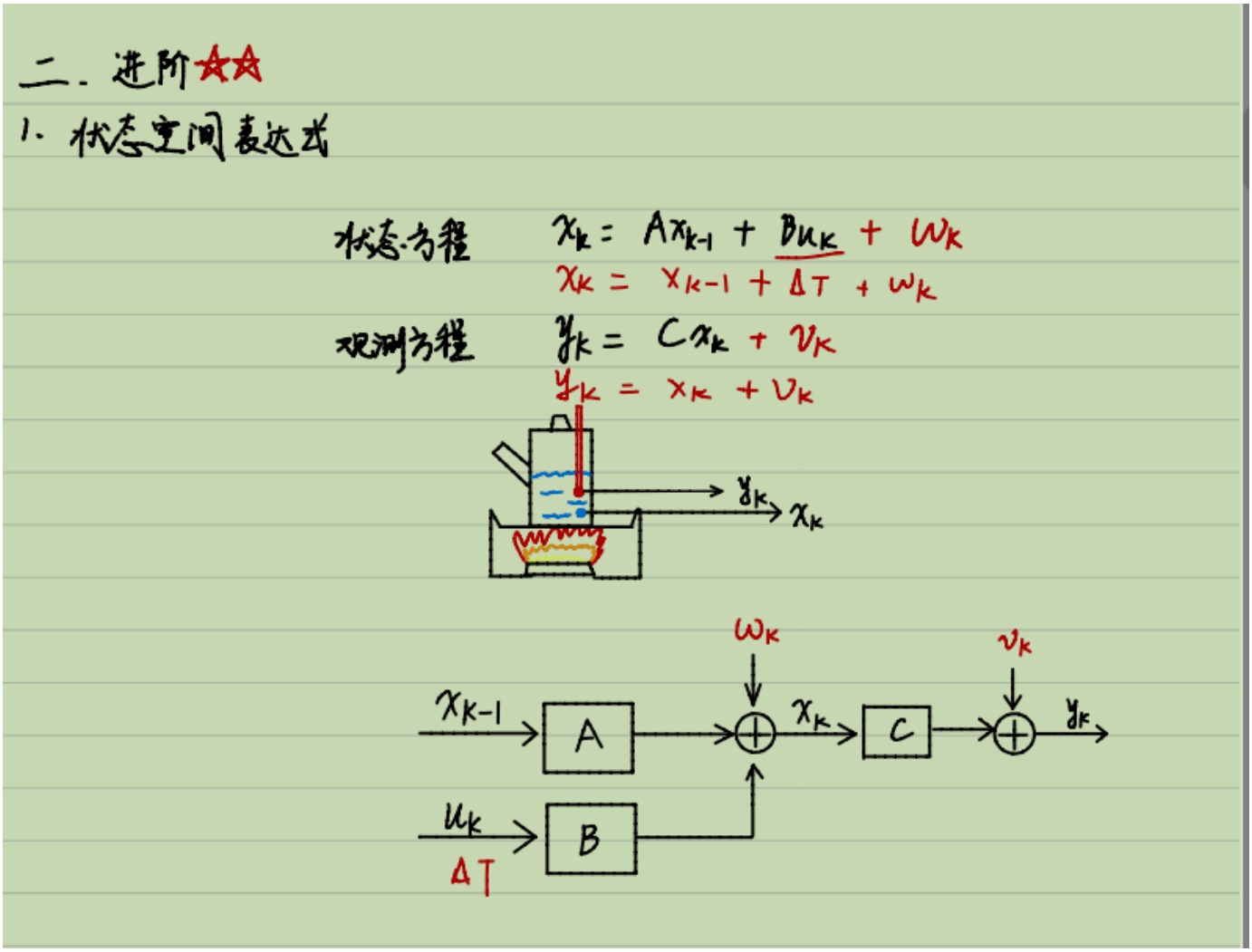
跟踪滤波的目的就是根据雷达录取设备提供的目标点迹参数来建立和更新轨迹，并外推下一次天线扫描周期目标出现的位置。为了解决机动目标的跟踪问题，许多学者对此进行了研究，并提出了许多跟踪算法，这些算法在特定条件下对目标机动问题都能进行比较准确的描述。但当条件发生变化时，这些算法的跟踪性能会有不同程度的降低。同时计算量偏大，实施比较困难。

Kalman滤波算法

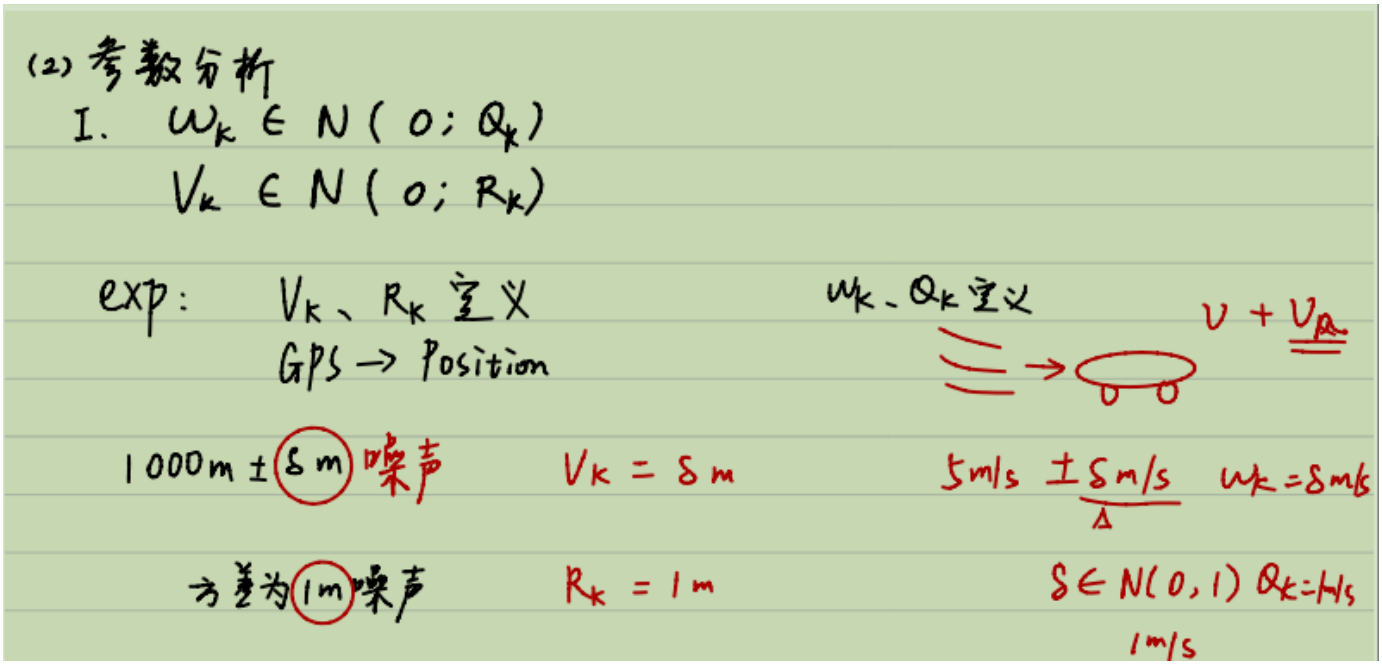
可以在任何含有不确定信息的动态系统中使用卡尔曼滤波，对系统下一步的走向做出有根据的预测，即使伴随着各种干扰，卡尔曼滤波总是能指出真实发生的情况。 在连续变化的系统中使用卡尔曼滤波是非常理想的，它具有占用内存小的优点（除了前一个状态量外，不需要保留其它历史数据），并且速度很快，很适合应用于实时问题和嵌入式系统。

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。



> w_k:过程噪声
>
> v_k:测量噪声



实例：匀速直线运动线性问题

（1）预测(或者叫传播)

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。

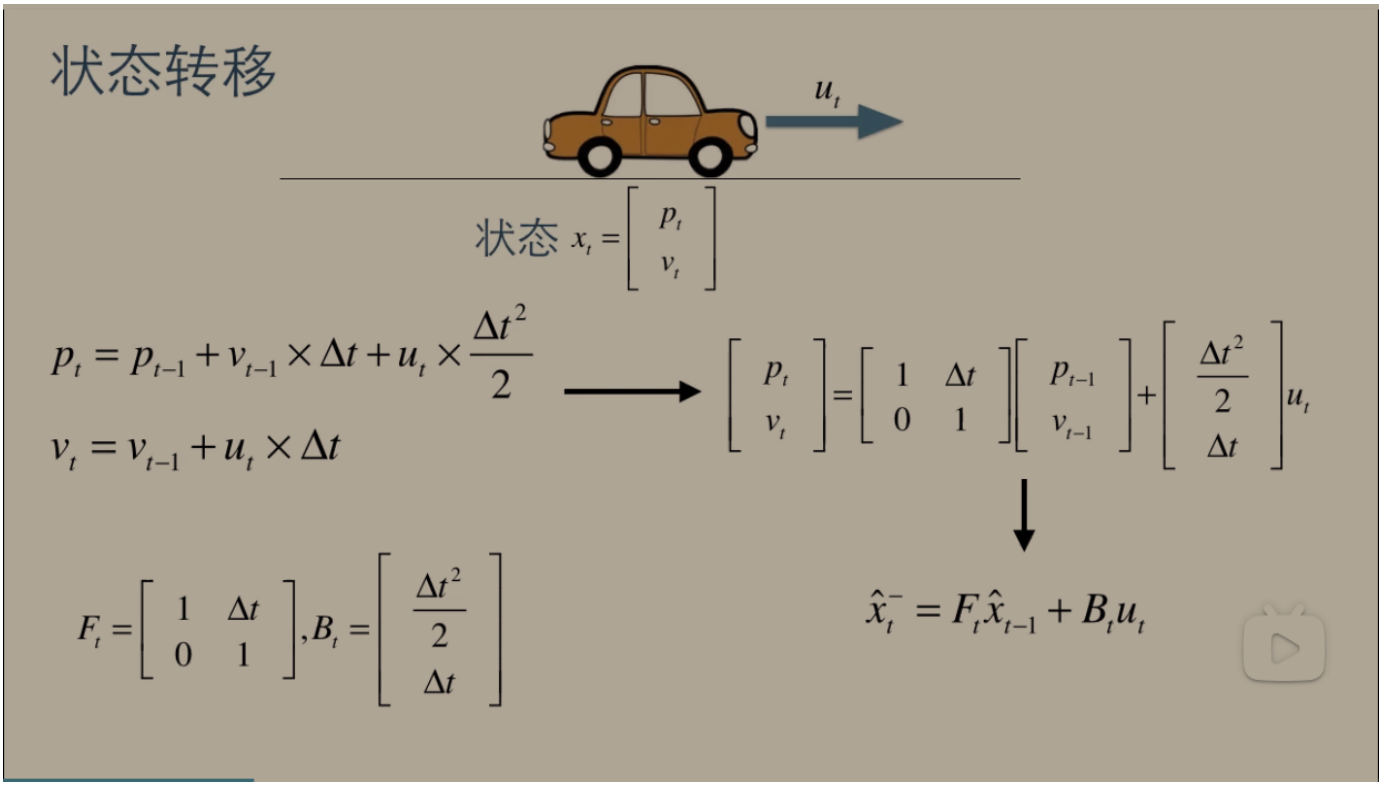
状态 $x_t = \begin{bmatrix} p_t \\ v_t \end{bmatrix}$

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):

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

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

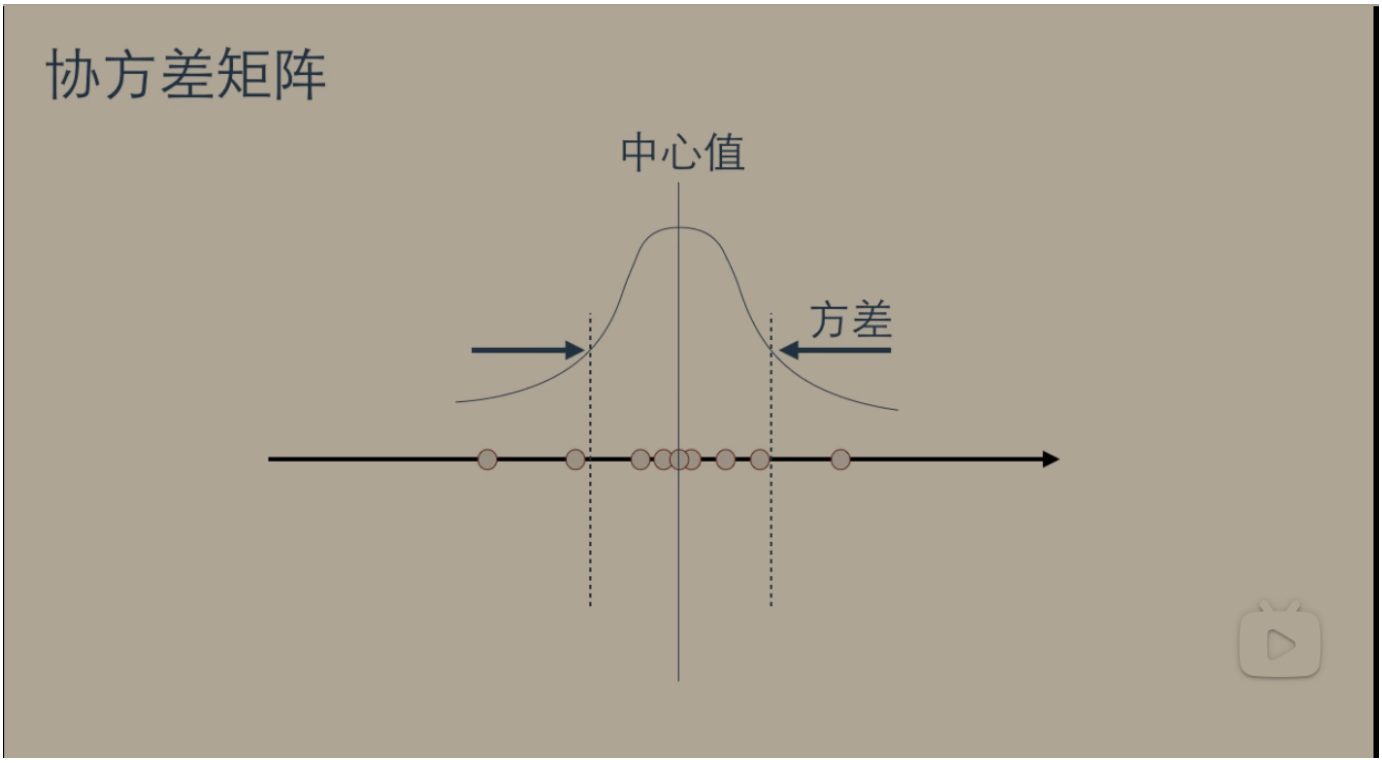


> 矩阵 F_t为状态转移矩阵，表示如何从上一状态来推测当前时刻的状态；
>
> B_t 为控制矩阵，表示控制量u_t 如何作用于当前矩阵；

> 带^帽子的表示估计值,并不是最优的。
>
> 带$\bar{x}$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)

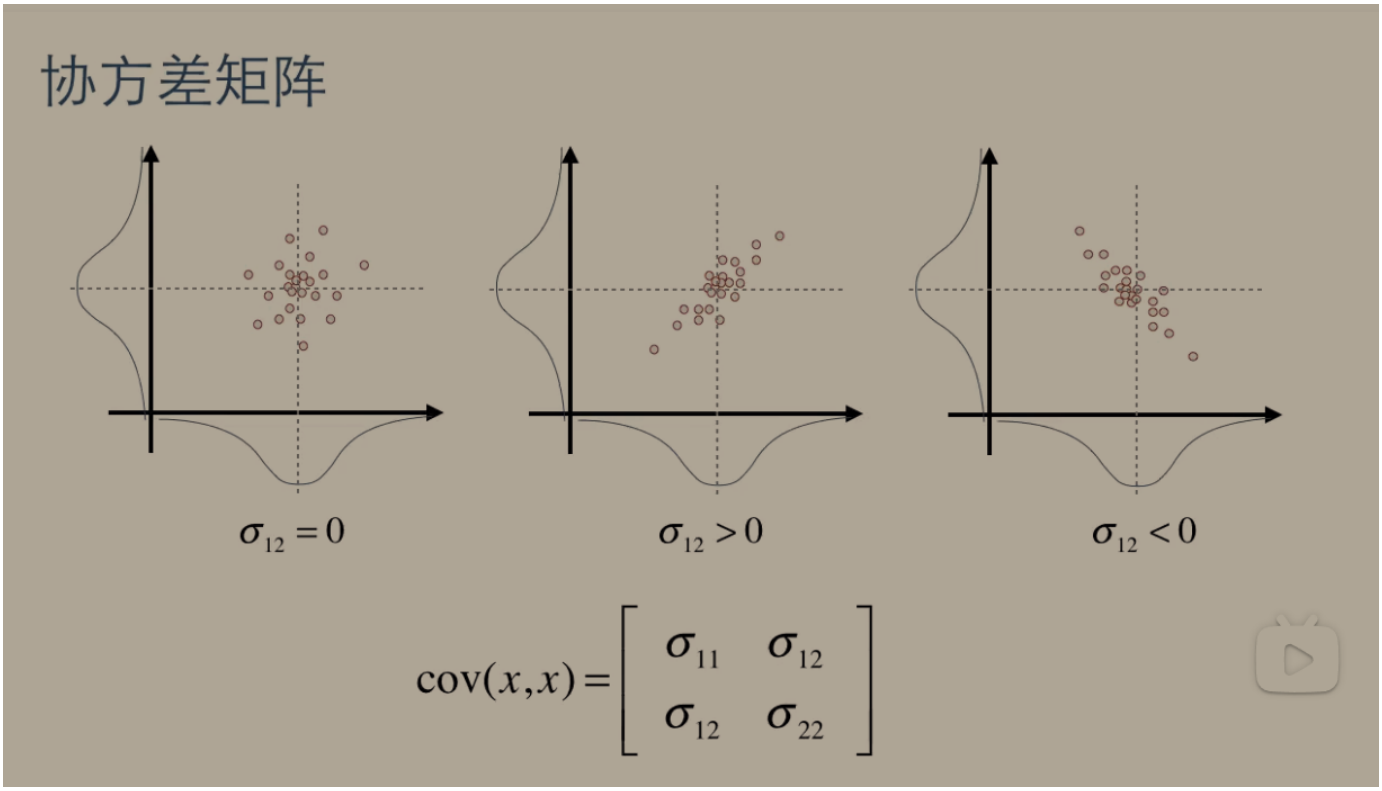
到这里，我们已经得到了t时刻的状态预测，但这里只有状态预测的均值（即$\bar{x}_t$）。我们是基于高斯分布来建立状态变量的，因此还需要协方差。（即一个多维高斯分布由均值【中心值】和协方差决定）

一维矩阵↓



假设P表示预测协方差 P=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



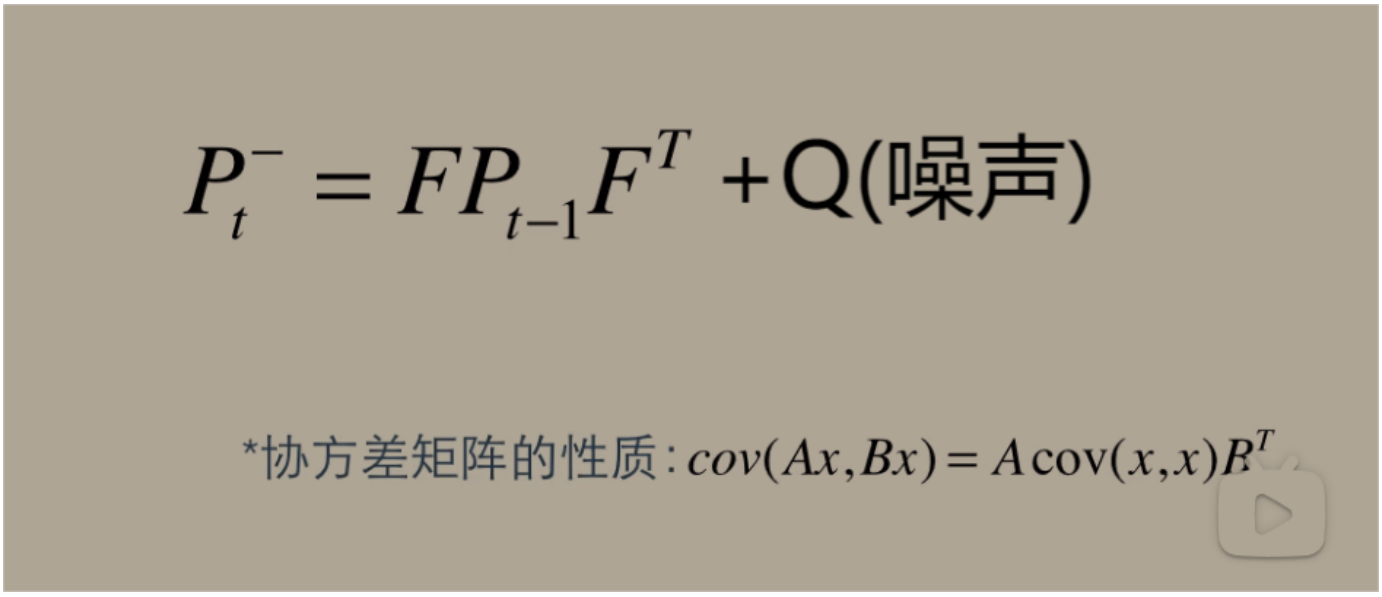
那么加入状态转换矩阵F_t后有

> 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?

$$\mathrm{P}_t = cov(Fx_{t-1}, Fx_{t-1}) = Fcov(x_{t-1}, x_{t-1})F^T = FP_{t-1}F^T$$

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

<u>②噪声协方差矩阵（先验估值协方差）</u>：

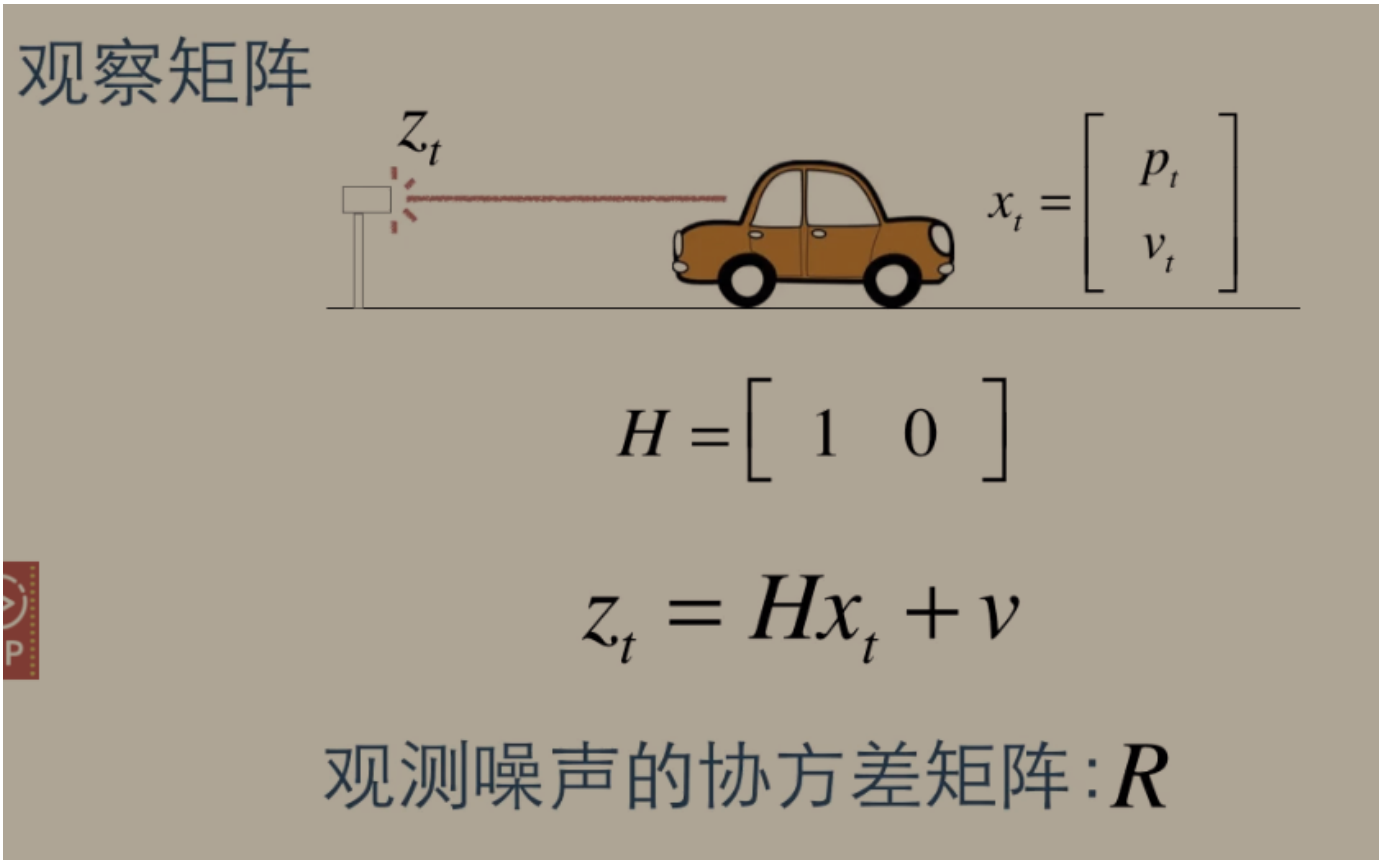


（2）更新(观测值)

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。

<u>③测量方程：</u>



> 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

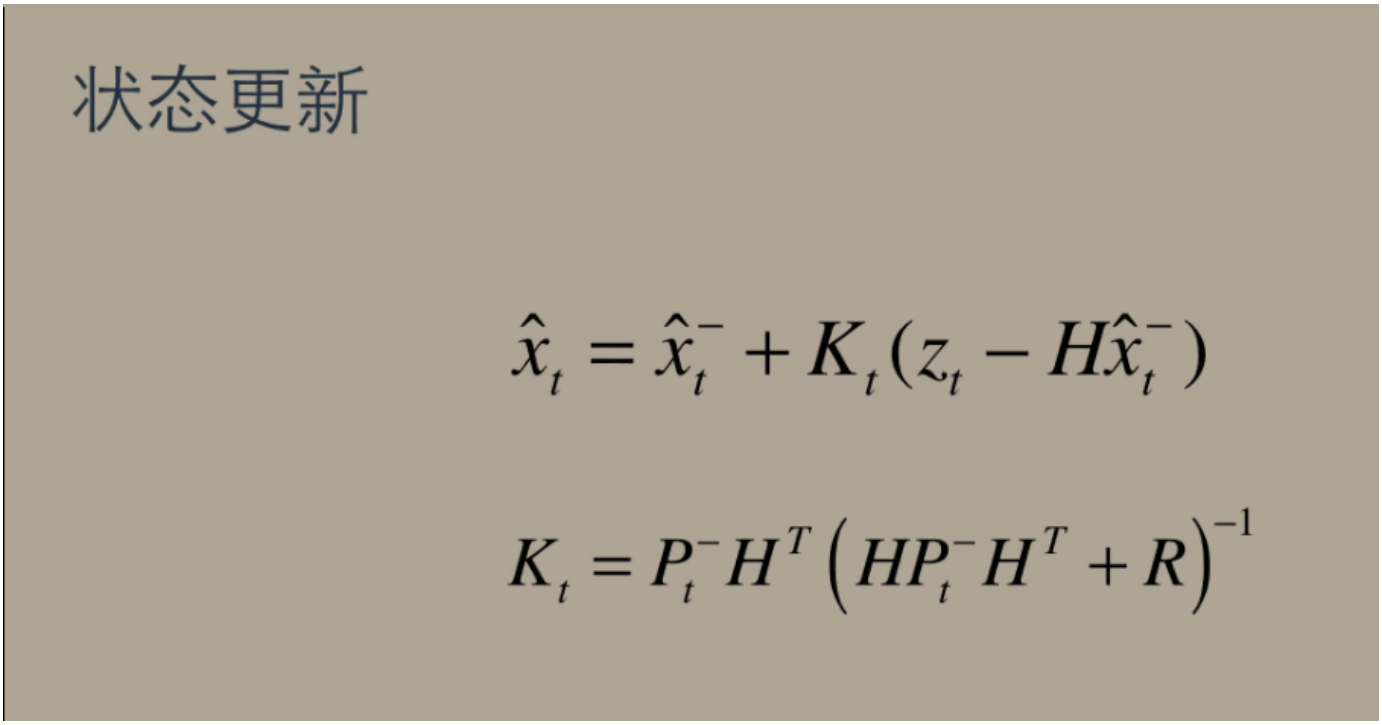
现在我们有了两个高斯分布，一个是在预测值附近，一个是在传感器读数附近。我们必须在'预测值'和'传感器测量值'之间找到最优解。我们只需将这两个高斯分布(前一状态的预测以及传感器的测量)相乘就可以了。

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



> 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

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

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

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

F和H矩阵取1

二. 调节超参数

(1) Q与R的取值

I、公式层面理解

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

$$K = \frac{P_t^- H^T}{HP_t^- H^T + R}$$

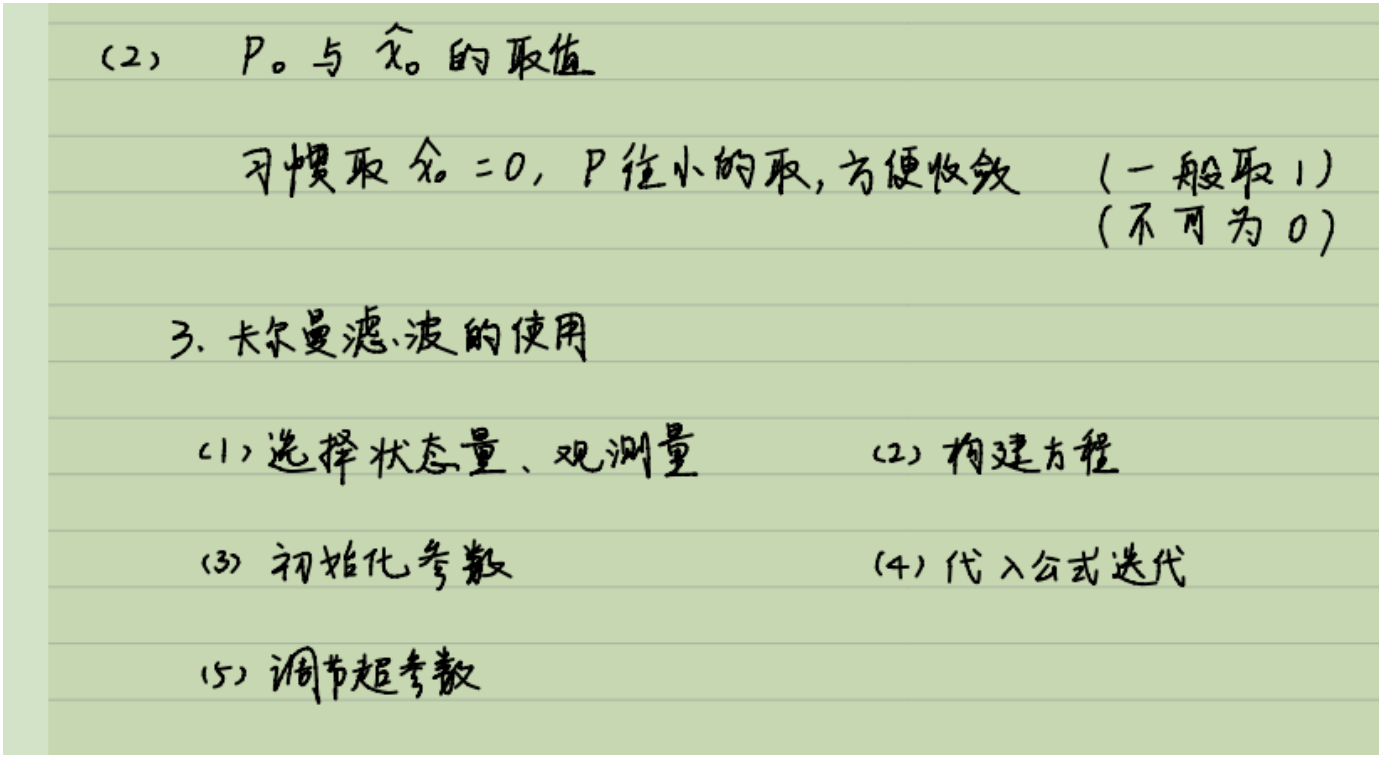
化简 $\longrightarrow$ $K = \frac{P_{t-1} + Q}{P_{t-1} + Q + R}$

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

① 分析Q ②分析R

II、其他层面理解

① 分析Q ②分析R



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。