This is the title: it contains a colon

Author One

Author Two

Abstract

This is the abstract.

It consists of two paragraphs.

Table of Contents

# 卡尔曼滤波

## 参考教程

1. [【从放弃到精通！卡尔曼滤波从理论到实践~】](https://www.bilibili.com/video/BV1Rh41117MT/?p=3&share_source=copy_web&vd_source=6b55cb6788b1952e04c06b095d772810)：从该教程中提取公式并理解。这个讲得很好，但是公式不够严谨，还有小错误，而且完全跳过了重要的更新过程。
2. [【【不要再看那些过时的卡尔曼滤波老教程了】2022巨献，卡尔曼滤波-目标追踪从理论到实战最新版全套教程！建议收藏】](https://www.bilibili.com/video/BV1iv4y1N7As/?share_source=copy_web&vd_source=6b55cb6788b1952e04c06b095d772810)：只看了p1而且讲得很不好，但是帮助理解更细那块有点点小用吧。
3. [一文看懂卡尔曼滤波（附全网最详细公式推导） - 付聪的文章 - 知乎](https://zhuanlan.zhihu.com/p/433560568)：公式推导挺全面的，但是流畅度还是不如我自己的这篇。

## 文件说明

1. [KalmanFiltering](./卡尔曼滤波Clang.pdf)：上文中提到的卡尔曼滤波视频教程的pdf文件。
2. ：。

## P3 进阶（基本滤波知识储备）

### 状态空间表达式

#### 状态方程

解释：当前时刻的状态等于上一时刻的状态乘以系数，加上输入值乘以系数，最后加上过程噪声。

  又称为状态转移矩阵，因为的大小决定了状态变化的快慢。又称为控制矩阵，因为的大小决定了输入对系统影响的程度。

  如何理解过程噪声？状态方程的含义是我们对系统的数学建模，构建了一个数学表达式，但是实际的系统往往是复杂的，在随时间变化的时候，会受到各种各样的因素的干扰，可以认为是一个总干扰，这种干扰是随机的，符合正太分布（高斯分布）。

#### 观测方程

是观测量，是状态量，是状态量的系数，又称观测矩阵因为我们假定观测量和状态量之间的关系是线性的，可以由这个线性矩阵表示。是观测噪声。

  可以这样理解：测量值和状态量存在一种线性关系，但是测量总是有误差的，所以要加上观测噪声，观测噪声同样符合正太分布。

  我们可以用一个方框图表示这个系统，但是没有找到一个合适的绘制方框图的绘图软件。

### 参数分析

  噪音服从均值为零的正态分布，可以记为：

他们的均值均为零，方差分别为和，方差为常数，与时间无关。和就是其中的一个取值，只是取得这个结果的概率服从正态分布：

### 超参数

  噪音的方差，具体是多少，我们是不知道的，是需要不断调节的参数，从而使系统达到稳定。和调节PID一样。

### 卡尔曼滤波直观图解

  通过卡尔曼滤波也不会得到真实值，而是每个时刻状态的最优估计值，是修正值，也叫后验估计值。总结来说是，上一时刻的最优估计值，根据物理理论再加上过程噪声（状态方程）得到的先验估计值，然后利用有噪音的观测值（观测方程），进行加权得到最优估计值。这样从先验到后验，也就实现了预测到更新，而最优估计值的不断变化，也就是迭代。因为观测值也是有噪声的，通过得到最优估计值，也就是实现了滤波，去除了部分噪音。

## P4 放弃（通俗公式理解）

### 预测模型

  考虑一个情景，一辆匀加速（加速度为）直线运动的小车行驶在公路上，我们研究小车的状态有位置和速度，根据物理规律构建状态方程：

其中 表示当前时刻， 表示上一时刻。这是高中物理学习的标准的位置和速度公式。写成矩阵形式：

写成矩阵的形式为：

其中下划线 表示没用噪音的理论值。上式加上噪音后：

满足状态方程[式 (1.1)](#eq:状态方程)。其中，是位置噪音，是速度噪音，均服从正太分布，噪音是随机的，与时间有关，每个时刻产生随机的噪音。

### 预测方程

#### 状态预测

  依据去除噪音的状态方程[式 (1.1)](#eq:状态方程)，就能直接写出先验估计的表达式：

符号和的含义在前文[卡尔曼滤波直观图解](#卡尔曼滤波直观图解)中已经提到，引入符号 和是为了表示估计和先验。

#### 状态误差预测

  上[式 (1.10)](#eq:先验估计)对比状态方程[式 (1.1)](#eq:状态方程)，可以求出先验误差：

先验误差表示真实值和预测值的差值。其中为后验误差，即：

后验误差表示真实值和估计值的差值。所以上[式 (1.11)](#eq:先验误差)先验误差可以进一步表示为：

  接下来求先验误差的协方差矩阵：

记先验误差的协方差矩阵，则上[#eq:先验误差的协方差矩阵1]可以写作：

其中是上一时刻的后验误差的协方差矩阵。不同状态之间的噪音很可能不是独立的，他们具有一定的相关性。

### 更新模型

  对于这个小车，我们有卫星测量，卫星只能测量小车的位置，卫星无法测量速度，所以理想的观测方程为：

写成矩阵的形式：

一定不能写成下面的形式，因为 没有被观测，而不是 ！

错误的矩阵的形式：

其中引入了速度观测噪音，但是速度是没有测量的，可以去掉这个测量量。同样的有其他传感器的话可以增加测量量。所以观测方程的维度可以和状态方程的维度不同。写成矩阵形式：

其中下划线 表示没用噪音的理论值，上式加上以下噪音：

其中，测量值和实际值之间存在位置观测噪音，速度观测噪音。可以看到该[式 (1.17)](#eq:位置和速度观测方程的矩阵形式)加上噪音后，满足观测方程[式 (1.2)](#eq:观测方程)。

### 更新方程

#### 修正估计

  先验估计值是理想状态下，没有噪音，是根据物理理论规律推导出来的当前时刻的预测的状态值。如果我们把观测方程也去除噪音值，当作一个理想的公式，并把这个先验估计值带入去除噪音的观测方程中，那么实际的观测值和理论上理想的观测值之间存在一个差值，这个值就叫测量残差，也叫残差，记为：

  我们永远无法得到状态的真实值，只能尽量得到一个最优的估计值，但是这个值无法直接用先验估计值和残差直接表示，因为残差是理想状态计算的观测值和实际的观测值之间的差值，但是这个值并不能直接表示后验估计值和先验估计值之间的差值，但是可以用一个系数使他们相等：

其中又称为卡尔曼滤波系数，上[式 (1.23)](#eq:最优估计偏差和残差的关系)我们习惯写成：

这样，最优估计值等于当前时刻的预测值，加上权重乘以观测误差。

#### 更新卡尔曼增益

  那么这个权重给多少呢？怎么给呢？是随便给吗？有什么依据吗？

  答：当然是有依据的。依据当然是真实值和最优估计值之间的差值最小，即后验误差最小，后验误差见上[式 (1.12)](#eq:后验误差)。将上[式 (1.24)](#eq:最优估计值计算公式)最优估计值带入上[式 (1.12)](#eq:后验误差)中：

继续化简，将观测方程[式 (1.2)](#eq:观测方程)带入上[式 (1.25)](#eq:后验误差变换1)中：

继续化简，将上[式 (1.11)](#eq:先验误差)的先验误差带入上[式 (1.26)](#eq:后验误差变换2)中：

  接下来求后验误差的协方差矩阵：

记后验误差的协方差矩阵，则上[#eq:后验误差的协方差矩阵1]可以写作：

其中之所以能化简，是由于误差的协方差矩阵和均是自相关矩阵，即以对角线为分界线，矩阵是对称的，这是协方差矩阵的性质。我们想要后验误差的协方差矩阵最小，只要让该矩阵对角线上的和最小就行了，即矩阵的迹最小。为什么呢？因为对角线上是每个误差本身的协方差，也就是方差，他们的和最小，协方差矩阵就最小；矩阵的其他位置是两两误差的相关性，与整体误差无关，对整体误差大小没有影响。因此：

其中代表矩阵的迹。把上[式 (1.30)](#eq:后验误差的协方差矩阵的迹)看作是关于的函数：

显然，这个函数是一个开口向上的一元二次函数，有极小值。它是一个凸函数，导数为0的点就是最小值。上[式 (1.30)](#eq:后验误差的协方差矩阵的迹)求导为：

这里是对矩阵的迹的求导，求导公式从函数求导理解，想弄清楚，那么具体矩阵迹的求导需要单独学习。使得上式为零，则可计算出卡尔曼滤波系数：

#### 更新后验误差的协方差

  将计算得到的卡尔曼滤波系数带入上[式 (1.29)](#eq:后验误差的协方差矩阵2)，计算得到后验误差协方差矩阵：

## 总结

  至此，卡尔曼滤波的五个重要的公式已经全部推导完成。分别是：[式 (1.10)](#eq:先验估计)、[式 (1.15)](#eq:先验误差的协方差矩阵2)、[式 (1.33)](#eq:卡尔曼滤波系数)、[式 (1.24)](#eq:最优估计值计算公式)、[式 (1.34)](#eq:后验误差的协方差矩阵3)。综上，我们把这五个重要的公式写在一起：

预测公式：

更新公式：

### matlab仿真

  所有的代码算法，严格按照公式推导进行设计。所有的代码变量全部取自于latex公式符号。有两点需要额外设计：一是初始值，这是在公式推导中没有提及的。二是遇到了数值分析问题，需要对公式进行移向等基本变化进行改造；但也是要详细列写公式的，并严格依据公式写代码；数值问题是在跑代码时发现的，发现数值问题，再回头改公式。matlab仿真公式如下：

%% 清理  
close all; clear; clc;  
% 估计小车在每一时刻的位置和速度  
  
%% 无噪音的理想值  
T = (1 : 100); % 离散的时间序列，单位是s  
p = zeros(1, length(T)); v = zeros(1, length(T));   
p(1) = 0 % 小车的初始位置  
v(1) = 3 % 小车的初始速度  
a = 0.1; % 加速度  
  
for t = 2 : length(T)  
 p(t) = p(t - 1) + v(t - 1) + 1/2 \* a; % 理想的位置  
 v(t) = v(t - 1) + a; % 理想的速度值  
end  
  
underline\_x = [p; v];  
   
%% 有噪音的实际值，模拟噪音  
% 状态噪音  
sigma\_r = 0.01  
r\_p = sigma\_r \* randn(1, length(T));  
r\_v = sigma\_r \* randn(1, length(T));  
x = underline\_x;  
for t = 2 : length(T)  
 x(1, t) = x(1, t - 1) + x(2, t - 1) + 1/2 \* a + r\_p(t); % 真实的位置  
 x(2, t) = x(2, t - 1) + a + r\_v(t); % 真实的速度值  
end  
  
% 观测噪音  
sigma\_Delta\_p = 3  
Delta\_p = sigma\_Delta\_p \* randn(1, length(T)); % GPS测量误差，标准差为3m  
z = x(1, :) + Delta\_p; % GPS的观测值，带有测量误差  
  
%% 卡尔曼滤波  
% 已知量  
% 已知的线性变换矩阵  
F = [1, 1; 0, 1]; % 状态转移矩阵  
B = [1/2; 1]; % 控制矩阵  
u = a;  
H = [1, 0];  
sigma\_Q = 0.01  
Q = [sigma\_Q^2, 0; 0, sigma\_Q^2]; % 过程噪声的协方差矩阵，这是一个超参数  
sigma\_R = 3  
R = sigma\_R^2; % 观测噪声的协方差矩阵，也是一个超参数。因为是一维的，就是一个数  
  
  
% 初始化  
hat\_x\_ = zeros(2, length(T)); hat\_x = zeros(2, length(T));  
hat\_x(:, 1) = [0; 0]; % 初始状态，[位置, 速度]，就是我们要估计的内容，开始时都未知，设为0  
P = [0.1, 0; 0, 0.1]; % 先验误差协方差矩阵的初始值，根据经验给出  
  
% 卡尔曼滤波5个关键公式  
for t = 2:length(T)  
 hat\_x\_(:, t) = F \* hat\_x(:, t - 1) + B \* u;  
 P\_ = F \* P \* F' + Q;  
 K = (P\_ \* H') ./ (H \* P\_ \* H' + R);  
 hat\_x(:, t) = hat\_x\_(:, t) + K \* (z(:, t) - H \* hat\_x\_(:, t));  
 P = (eye(2) - K \* H) \* P\_;  
end  
  
%% 绘图  
figure(1);  
plot(T, underline\_x(1, :), 'b'); % 位置的理想值  
hold on;  
plot(T, x(1, :), 'g'); % 位置的实际值  
plot(T, z(1, :), 'm'); % 位置的观测值  
plot(T, hat\_x(1, :), 'r.'); % 位置的最优估计值  
title('对位置的估计');  
xlabel('时间'); ylabel('位置');  
  
figure(2);  
plot(T, underline\_x(2, :)); % 速度的理想值  
hold on;  
plot(T, x(2, :), 'g'); % 速度的实际值  
plot(T, hat\_x(2, :), 'r.'); % 速度的最优估计值  
title('对速度的估计');  
xlabel('时间'); ylabel('速度');

其中，用matlab做实时脚本时，这些地方可以做数字滑块：1. 初始位置p(1)和速度v(1)，2. 过程噪声的标准差sigma\_r和观测噪声的标准差sigma\_Delta\_p，3. 超参数 和 的标准差sigma\_Q和sigma\_R。

  不管初始位置和速度时多少，卡尔曼滤波很快就能收敛，跟上脚步，和实际值接近。过程噪声的标准差不能很大，很大说明状态方程的物理模型建立的不合理，且在没有观测速度的条件下，卡尔曼滤波效果特别是对速度的估计值就很差；除非观测噪声很小，观测精度很高，卡尔曼滤波在状态方程的物理模型建立的不合理的条件下，依赖观测值也能得到较好的结果。超参数 和 可能需要不断地调试，以达到最佳的曲线效果；可以发现把超参数的标准差和模拟噪音的标准差设置一样时效果最好；实际中由于对过程噪声的未知，需要通过不断调试更改超参数，以使得曲线最佳，此时的超参数基本上就和系统的噪音相差无几。

#### 时间复杂度

  需要考虑matlab仿真的时间复杂度。就是考虑算力问题。

### GNSS定位领域

### 变换

  以上推导的公式形式常用于自动控制领域。在卫星导航领域，状态空间表达式如下：

上式和状态方程[式 (1.1)](#eq:状态方程)的以及观测方程[式 (1.2)](#eq:观测方程)还有三个区别：

一个是下角标出现在一些系数矩阵上，这是为什么呢？是因为系数矩阵所乘以的矩阵变量是上一时刻的，从上一时刻到这一个时刻的矩阵变换关系表示为。第二个变化是观测方程的单噪音项变成了两项相乘，其中为上一时刻的动态噪声，为噪音变化矩阵。第三个变化是在观测方程中，该时刻的控制项，该控制项将影响观测值，在部分传感器中会出现这种观测方程，这种传感器我目前还没见过。

  同理，可以得到卡尔曼滤波公式：

时间更新：

上式与上[式 (1.35)](#eq:预测)预测方程除了符号以外完全相同，但是确实不够优雅，可读性也不如[式 (1.35)](#eq:预测)，比如先验误差符号有点怪怪的。之前叫预测，但是这里叫一步预测公式的时间更新，因为从公式上来说，是从上一个时刻推导到这时刻。但是预测的叫法和时间更新的叫法值得是一个东西。

测量更新：

其中：

其中：

上式和[式 (1.36)](#eq:更新)更新方程除了符号以及状态方程那儿说的不同之处以外没有任何不同。这里除了之前叫卡尔曼滤波系数，这里还叫滤波增益矩阵。除了叫验前残差以外还叫新息矩阵、OMC向量。

### 简化

  在GNSS定位数据处理中，通常不涉及控制向量。非随机控制项为零时，即方程和观测方程中，也即卡尔曼滤波公式中和，去掉包含这两项的内容即可。