

梦，要做就做到最巅峰！

衡量一个人活了多少年，要看他有多少的思考时间！【百家争鸣】【独立思考】【记录分享】

目录视图

摘要视图

RSS 订阅

个人资料



Fighting_MX

访问: 105687次

积分: 1749分

排名: 第5984名

原创: 47篇

转载: 135篇

译文: 0篇

评论: 57条

文章搜索

文章分类

C&C++ (58)

Matlab (6)

计算机视觉 (30)

摘录与随笔 (20)

数学 (5)

算法 (5)

Python (18)

Latex (3)

Linux (9)

C# (4)

Something about computer (4)

大杂烩 (21)

文章存档

2013年12月 (1)

2013年09月 (6)

2013年05月 (3)

2013年04月 (14)

2013年03月 (9)

展开

阅读排行

DOS批处理高级教程(还 (5431)

波利亚-怎样解题的思考 (3358)

文思海辉第一届在线编程大赛 消灭0回答，赢下载分 订阅CSDN社区周刊，及时了解社区精华内容 办公大师系列经典丛书 诚聘英才

对Kalman(卡尔曼)滤波器的理解

分类: 计算机视觉 2013-04-20 15:07 2150人阅读 评论(1) 收藏 举报

1. 简介(Brief Introduction)

在学习卡尔曼滤波器之前，首先看看为什么叫“卡尔曼”。跟其他著名的理论（例如傅立叶变换，泰勒级数等等）一样，卡尔曼也是一个人的名字，而跟他们不同的是，他是个现代人！

卡尔曼全名Rudolf Emil Kalman，匈牙利数学家，1930年出生于匈牙利首都布达佩斯。1953，1954年于麻省理工学院分别获得电机工程学士及硕士学位。1957年于哥伦比亚大学获得博士学位。我们现在要学习的卡尔曼滤波器，正是源于他的博士论文和1960年发表的论文《A New Approach to Linear Filtering and Prediction Problems》（线性滤波与预测问题的新方法）。如果对这篇论文有兴趣，可以到这里的地址下载：
http://www.cs.unc.edu/~welch/kalman/media/pdf/Kalman1960.pdf

卡尔曼滤波器到底是干嘛的？我们来看下wiki上的解释：

卡尔曼滤波的一个典型实例是从一组有限的，包含噪声的，对物体位置的观察序列（可能有偏差）预测出物体的位置的坐标及速度。在很多工程应用(如雷达、计算机视觉)中都可以找到它的身影。同时，卡尔曼滤波也是控制理论以及控制系统工程中的一个重要课题。例如, 对于雷达来说，人们感兴趣的是其能够跟踪目标。但目标的位置、速度、加速度的测量值往往在任何时候都有噪声。卡尔曼滤波利用目标的动态信息，设法去掉噪声的影响，得到一个关于目标位置的好的估计。这个估计可以是对当前目标位置的估计(滤波)，也可以是对将来位置的估计(预测)，也可以是对过去位置的估计(插值或平滑)。

斯坦利.施密特(Stanley Schmidt)首次实现了卡尔曼滤波器。卡尔曼在NASA埃姆斯研究中心访问时，发现他的方法对于解决阿波罗计划的轨道预测很有用，后来阿波罗飞船的导航电脑便使用了这种滤波器。关于这种滤波器的论文由Swering (1958)、Kalman (1960)与 Kalman and Bucy (1961)发表。

目前，卡尔曼滤波已经有很多不同的实现. 卡尔曼最初提出的形式现在一般称为简单卡尔曼滤波器。除此以外，还有施密特扩展滤波器、信息滤波器以及很多Bierman, Thornton 开发的平方根滤波器的变种。也许最常见的卡尔曼滤波器是锁相环，它在收音机、计算机和几乎任何视频或通讯设备中广泛存在。

简单来说，卡尔曼滤波器是一个“optimal recursive data processing algorithm（最优化自回归数据处理算法）”。对于解决很大部分的问题，他是最优，效率最高甚至是最有用的。他的广泛应用已经超过30年，包括机器人导航，控制，传感器数据融合甚至在军事方面的雷达系统以及导弹追踪等等。近年来更被应用于计算机图像处理，例如头脸识别，图像分割，图像边缘检测等等。

2. 卡尔曼滤波器的介绍(Introduction to the Kalman Filter)

matlab中如何把m文件默	(3025)
multisim 元件库介绍	(2717)
VC编译错误: msvcrt.lib	(2430)
VS2010调用Matlab 2010C	(2352)
给中国学生的第三封信:	(2312)
python重定向标准输入、	(2281)
对Kalman(卡尔曼)滤波器	(2149)
很不错的演讲大全	(2006)

评论排行	
Texstudio中文乱码问题	(9)
几何矩的物理意义, 由C	(7)
matlab中如何把m文件默	(5)
很不错的演讲大全	(5)
VS2010+Opencv+SIFT	(4)
BMP文件格式	(4)
VS2010调用Matlab 2010C	(4)
DOS批处理高级教程(辽	(2)
Cin详解	(2)
this指针的总结	(2)

推荐文章	
* android跨进程通信 (IPC) : 使用AIDL	
* 经典排序算法分析和代码-下篇	
* 69. XPages里的Java日志器	
* 界面编程模仿篇(QQ登录界面逼真篇)	
* OpenCV4Android开发之旅(二)——Eclipse调用OpenCV2. 4. 4的Java接口配置及示例	
* Android开发学习之Gallery	

最新评论	
考研期间写给自己的短信 Marconi1901: 认同你	
对Kalman(卡尔曼)滤波器的理解 Suehoa1117: 非常感谢博主, 讲的非常详细! 总结的深入浅出, 很适合初学者! 最后的代码有点小问题, 直接运行的话得出的不...	
Matlab实现类 usdxjt: 有没有个讲matlab高级编程的书给推荐下阿, 包括用matlab制作G U I , 面向对象编程等等。多谢阿...	
VS2010调用Matlab 2010 引擎的桡 ALASHJ: 确实, 最后用build里面那个configuration manager将32位改成64位平台之后编...	
随机过程之一——关于条件数学期 loveindemon: 你好, 我想请教一下, 如果X, Y是两个随机过程, 不是随机变量, 那么E{XY}表示的是什么, 一直不太懂...	
VS2010+Opencv+SIFT wenshixian1206: 现在主页没有那个链接了? 你可以将Windows VC++的版本 sift-latest_win.zi...	
matlab中如何把m文件默认打开方 hbxiangji: 按照lz说的做了, 不会打开两个MATLAB, 但是双击m文件还是会启动MATLAB, 求单独打开medi...	
matlab中如何把m文件默认打开方 hbxiangji: lz阿, 我是xp系统, 按你的方法设置后, 双击M文件, 还是会开启MATLAB。	
VS2010+Opencv+SIFT	

为了可以更加容易的理解卡尔曼滤波器, 首先应用形象的描述方法来讲解, 然后我们结合其核心的5条公式进行进一步的说明和探索。结合现代的计算机, 其实卡尔曼的程序相当的简单, 只要你理解了他的那5条公式。

在介绍他的5条公式之前, 先让我们来根据下面的例子做个直观的解释。

假设我们要研究的对象是一个房间的温度。根据你的经验判断, 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(假设我们用一分钟来做时间单位)。假设你对你的经验不是100%的相信, 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(White Gaussian Noise), 也就是这些偏差跟前后时间是没有关系的而且符合高斯分配(Gaussian Distribution)。另外, 我们在房间里放一个温度计, 但是这个温度计也不准确的, 测量值会比实际值偏差。我们也把这些偏差看成是高斯白噪声。

好了, 现在对于某一分钟我们有两个有关于该房间的温度值: 你根据经验的预测值(系统的预测值)和温度计的值(测量值)。下面我们要用这两个值结合他们各自的噪声来估算出房间的实际温度值。

假如我们要估算k时刻的是实际温度值。首先你要根据k-1时刻的温度值, 来预测k时刻的温度。因为你相信温度是恒定的, 所以你会得到k时刻的温度预测值是跟k-1时刻一样的, 假设是23度, 同时该值的高斯噪声的偏差是5度(5是这样得到的: 如果k-1时刻估算出的最优温度值的偏差是3, 你對自己预测的不确定度是4度, 他们平方相加再开方, 就是5)。然后, 你从温度计那里得到了k时刻的温度值, 假设是25度, 同时该值的偏差是4度。

由于我们用于估算k时刻的实际温度有两个温度值, 分别是23度和25度。究竟实际温度是多少呢? 相信自己还是相信温度计呢? 究竟相信谁多一点, 我们可以用他们的covariance来判断。因为 $Kg^2 = 5^2 / (5^2 + 4^2)$, 所以 $Kg = 0.78$, 我们可以估算出k时刻的实际温度值是: $23 + 0.78 * (25 - 23) = 24.56$ 度。可以看出, 因为温度计的covariance比较小(比较相信温度计), 所以估算出的最优温度值偏向温度计的值。

现在我们已经得到k时刻的最优温度值了, 下一步就是要进入k+1时刻, 进行新的最优估算。到现在为止, 好像还没看到什么自回归的东西出现。对了, 在进入k+1时刻之前, 我们还要算出k时刻那个最优值(24.56度)的偏差。算法如下: $((1 - Kg) * 5^2)^{0.5} = 2.35$ 。这里的5就是上面的k时刻你预测的那个23度温度值的偏差, 得出的2.35就是进入k+1时刻以后k时刻估算出的最优温度值的偏差(对应于上面的3)。

就是这样, 卡尔曼滤波器就不断的把covariance递归, 从而估算出最优的温度值。他运行的很快, 而且它只保留了上一时刻的covariance。上面的Kg, 就是卡尔曼增益(Kalman Gain)。他可以随不同的时刻而改变他自己的值, 是不是很神奇!

下面就要言归正传, 讨论真正工程系统上的卡尔曼。

3. 卡尔曼滤波器算法(The Kalman Filter Algorithm)

在这一部分, 我们就来描述源于Dr Kalman 的卡尔曼滤波器。下面的描述, 会涉及一些基本的概念知识, 包括概率(Probability), 随即变量(Random Variable), 高斯或正态分配(Gaussian Distribution)还有State-space Model等等。但对于卡尔曼滤波器的详细证明, 这里不能一一描述。

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, 我们结合下面PPT截图进行说明:

Wat_Mir: 没有那个dirent.h 怎么破
 matlab中如何把m文件默认打开方:
 nngunngu: 终于弄好了, 太谢谢楼
 主了

卡尔曼滤波器 (Kalman Filter)

基本假设:

□ 后验概率分布 $p(x_{k-1} | y_{1:k-1})$ 为高斯分布

□ 动态系统是线性的

$$x_k = Ax_{k-1} + Bu_{k-1} + q_{k-1}$$

$$y_k = Hx_k + r_k$$

□ 系统噪声和测量噪声都是高斯分布的, 协方差为 Q_{k-1} 和 R_k 。

National Laboratory of Pattern Recognition
 Institute of Automation, Chinese Academy of Sciences



模式识别
 中国

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。y(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。q(k) 和 r(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 covariance 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。先给出 KF 算法的流程和五个核心更新方程如下:
 KF 算法

X

$$x_i = A x_{i-1} + q$$

$$y_i = H x_i + r$$

KF

minimize the difference $x -$

- Prediction: $\hat{x}_i^- = A \hat{x}_{i-1}^- + q$
- Correction: $\hat{x}_i = \hat{x}_i^- + K (y_i - H \hat{x}_i^-)$

Kalman gain n

National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences

五个更新方程为:

卡尔曼滤波器—时间更新和状态更新

- 时间更新
- 状态更新

$$\hat{x}_k^- = A \hat{x}_{k-1} + B u_{k-1}$$

$$P_k^- = A P_{k-1} A^T + Q$$

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1}$$

$$\hat{x}_k = \hat{x}_k^- + K_k (y_k - H \hat{x}_k^-)$$

$$P_k = (I - K_k H) P_k^-$$

National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences

编写公式不方便，所以写成了PDF然后做了截图粘在了下面，下面就上面的例子和五个核心的公式对Kalman算法进行下说明：

- 一、 我们由k-1时刻的最优温度值 \hat{x}_{k-1} 去预测k时刻系统的状态(度) \hat{x}_k^- , $\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}$, 对应于上例中 $\hat{x}_k^- = \hat{x}_{k-1} = 23^\circ$
- 二、 由上一次的误差协方差 P_{k-1} 和过程噪声 Q 预测新的误差 $P_k^- = AP_{k-1}A^T + Q$, 对应于上例中 $\sqrt{3^2 + 4^2} = 5$ 。
- 三、 计算卡尔曼增益, $K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$, 对应于上例中 $5^2 / (5^2 + 4^2)$, $K_k = 0.78$ 。
- 四、 进行校正更新, $\hat{x}_k = \hat{x}_k^- + K_k(y_k - H\hat{x}_k^-)$, 对应于上例中 $\hat{x}_k = 23 + 0.78(25 - 23) = 24.56$, 此 \hat{x}_k 即为k时刻的最优温度值。
- 五、 为下一步估计k+1时刻的最优温度值的迭代进行更新操作新 P_k 值, $P_k = (I - K_k H)P_k^-$, 对应于上例中的 $((1 - K_k) * 5^2)^{0.5} = 3$ 其值和上一次迭代中的3代表的意义一样。

就这样, 算法就可以自回归的运算下去。

看到这聪明的同学可能已经看出来了, 问道卡尔曼增益为什么会是第三步中那样求, 现在只大致说一下原理, 具体推到比较复杂, 有兴趣的同学可以参考这文献去推一推。

还记得前面我们说的误差协方差矩阵 P_k 么, 即求第k次最优温度的误差协方差矩阵, 对应于上例中的3和2.35...这些值。看下面PPT, 我们最小化P即可得到卡尔曼增益K, 对应上例求解K只最小化最优温度值的偏差, 即最小化 $P(K)$:

• Minimize the expected error

$$\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}}; \quad \mathbf{P} = E(\mathbf{e}\mathbf{e}^T) = \begin{pmatrix} E(\mathbf{e}_1\mathbf{e}_1) & E(\mathbf{e}_1\mathbf{e}_2) & \cdots \\ E(\mathbf{e}_2\mathbf{e}_1) & E(\mathbf{e}_2\mathbf{e}_2) & \\ \vdots & & \ddots \end{pmatrix}$$

• It is easy to show

$$\mathbf{K} = \mathbf{P}^- \mathbf{H}^T (\mathbf{H} \mathbf{P}^- \mathbf{H}^T + \mathbf{R})^{-1},$$

where $\mathbf{P}_i^- = \mathbf{A} \mathbf{P}_{i-1} \mathbf{A}^T + \mathbf{Q}$

• Limits:

- system noise \ll measurement noise $\Rightarrow \hat{\mathbf{x}}_i = \hat{\mathbf{x}}_i^-$
- system noise \gg measurement noise $\Rightarrow \hat{\mathbf{x}}_i = \mathbf{H}^{-1} \mathbf{y}_i$

National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences



模式识别
中国科学院

我们由第四步可以看出, k时刻系统的最优温度值=k-1时刻状态估计值(由上一状态的最优温度值加上过程误差)+带卡尔曼增益权值项的偏差。如果观测误差远远大于估计误差, 那么K就很小, k时刻的预测值约等于k时刻的状态估计值, 如果对i时刻的状态估计值误差远远大于观测误差, 此时相应的q较大, K较大, i时刻的状态估计值更倾向于观察的数据。

卡尔曼滤波器的原理基本描述就完成了, 希望能帮助大家理解这5个公式, 其算法可以很容易的用计算机的程序实现。下面, 我会用程序举一个实际运行的例子。

4. 简单例子 (A Simple Example)

这里我们结合第二第三节，举一个非常简单的例子来说明卡尔曼滤波器的工作过程。所举的例子是进一步描述第二节的例子，而且还会配以程序模拟结果。

根第二节的描述，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 $u(k)=0$ 。因此得出：

$$x(k|k-1)=x(k-1|k-1) \dots\dots\dots (6)$$

式子（2）可以改成：

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

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可以改成以下：

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

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

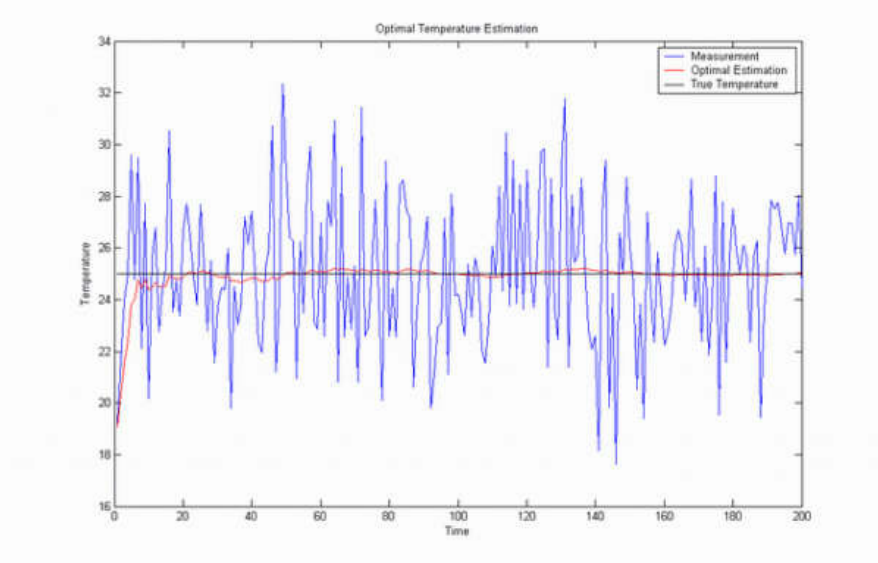
$$P(k|k)= (1-Kg(k)) P(k|k-1) \dots\dots\dots (10)$$

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（在图中为蓝线）。



为了令卡尔曼滤波器开始工作，我们需要告诉卡尔曼两个零时刻的初始值，是 $X(0|0)$ 和 $P(0|0)$ 。他们的值不用太在意，随便给一个就可以了，因为随着卡尔曼的工作， X 会逐渐的收敛。但是对于 P ，一般不要取0，因为这样可能会令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。我选了 $X(0|0)=1$ 度， $P(0|0)=10$ 。

该系统的真实温度为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 $Q=1e-6$, $R=1e-1$ ）。



附matlab下面的kalman滤波程序：

```
clear
N=200;
w(1)=0; %w为过程噪声
w=randn(1,N)
x(1)=25;
a=1; %a为方程中A(k)
for k=2:N;
x(k)=a*x(k-1)+w(k-1);
end

V=randn(1,N); %V为观察噪声
q1=std(V);
Rvv=q1.^2;
q2=std(x);
Rxx=q2.^2;
q3=std(w);
Rww=q3.^2;
c=0.2; %c为方程中H(k)
Y=c*x+V; %Y为观察值

p(1)=0;
s(1)=0;
for t=2:N;
p1(t)=a.^2*p(t-1)+Rww; %p1为方程中p'
```

```
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');
```


更为详细的过程可参考有关的资料。
文章参考了：
1 博文<http://hi.baidu.com/irvkqscjezbrtwq/item/4ad3bb018b8c7e37a3332a07>
2 自动化所董秋雷上课课件
3 《学习Opencv》于仕琪 P384 kalman滤波器部分
4 如果做视频跟踪具体参数选择可参考《数字视频处理》黎洪松 P102-106
5 如果想探索其具体推导过程可参考《现代信号处理》张贤达 P177-188

更多 2

上一篇：[让基于Opencv的程序在别的机子上跑起来-实例](#)
下一篇：[Latex显示罗马数字](#)

顶 7 踩 0

查看评论

1楼 [Suehoa1117](#) 2013-11-05 23:12发表
 非常感谢博主，讲的非常详细！总结的深入浅出，很适合初学者！
最后的代码有点小问题，直接运行的话得出的不是上面那个图

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

专区推荐内容

从零开始编写网络游戏
#HTML (HTML5)：多媒体...
[C/C++]类别互相引用
超极本 和平板电脑 Window...
面向新型一体式电脑开发浸入式应用
以用户为中心实现绝佳的用户体验

<< >>

更多招聘1

【职威
【北京
【天翼
【北京
【新浪
【格林

核心技术类目											
全部主题	数据挖掘	SOA	UML	开放平台	HTML5	开源	移动开发	iOS	Android	移动游戏	
Windows Phone	JavaScript	CSS	游戏引擎	云计算	大数据	Hadoop	OpenStack	云平台			
PHP	MongoDB	JSON	Xcode	Node.js	前端开发	神经网络	安全	Java	.NET	MySQL	
textView	BigTable	web框架	SQL	Redis	CouchDB	Linux	可穿戴计算	NoSQL	Ruby	API	
GPL	XAML	ASP.NET	前端开发	虚拟化	框架	机器学习	数据中心	IE10	敏捷	集群	