卡尔曼滤波学习讨论

由于工程实践中经常会遇到关于卡尔曼滤波的问题，经过一段时间的学习，我对卡尔曼滤波的基本思想有了一定的掌握，同时工作中也在用卡尔曼滤波，所以也有很多体会。为了更好的方便大家学习研究卡尔曼滤波，我将自己的一些心得分享一下，不足之处，欢迎批评指正。

一、卡尔曼滤波的基本思想

第一个问题就是我为什么要用卡尔曼滤波，它是如何产生的？在工程实践中，我们经常需要对一个物体的状态进行确认，比如室内的温度、物体的姿态角度、飞行器的空间位置等等。我先从室内温度入手吧，这个例子相信接触过卡尔曼滤波的人都会比较熟悉。

我们手中有一个温度计，可以测得任何时刻的温度值，但我们也知道这个测量出来的温度值（也就是观测值），它是不准确的，有误差的，它不是我们想要的真实值，那么我们如何得到温度的真值，或者尽可能的得到接近真值的结果呢？

我们可以研究温度的变化规律，任何事物的变化都服从一定的内在规律，也就是物理学定律。通过研究物理学定律，我们可以发现温度的变化是满足某种关系的，这里我假定它满足一个线性关系，，如果k时刻的温度值为5，那么通过这个函数关系，我就可以知道k+1时刻的温度值6，这就是一个预测的过程，这个过程的关键就是对温度的变化建立了一个数学模型，通过模型可以进行预测，它只需要知道上一时刻的值就可以了。

现在假定我用温度计测出来一个温度值为7，又通过模型预测得到了一个预测值为6，那么当前时刻的温度值到底应该是多少呢？现在我们手中已经得到了两个值，一个预测值，一个实测值，但他们都是不准确的，都是含有误差的，这个时候我们该如何判断呢？其实很简单，哪个值可信度高，我就想相信它多一点，可信度低，就相信它少一点。假设预测值的

可信度为2，实测值的可信度为3，那么我们就可以得到温度的最终结果为



这就是我们的一个最终结果。此时，可能有人要问，这个可信度是如何得到的呢？在后面的过程中，我会慢慢讲到。

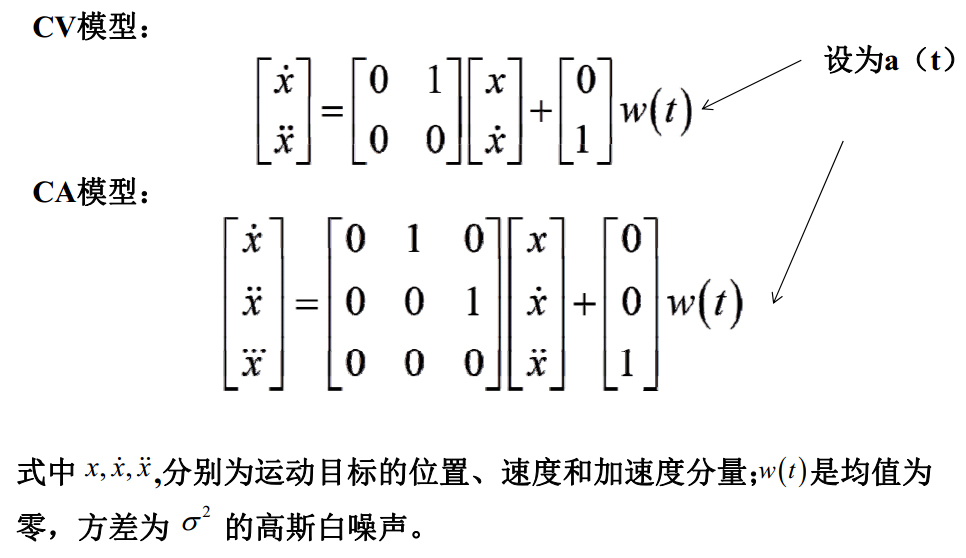
通过上面的分析，我们可以看到卡尔曼滤波其实就是一个加权的过程，预测值和实测值进行综合加权得到了一个最优估计。说起来很简单，可问题的关键就在于权值的确定，我该如何给它们分配权重呢？卡尔曼在通过模型预测的同时，也会计算相应的可信度，也就是我们经常看到的那个协方差矩阵P，同时它也要得到测量值的可信度，也就是书上的那个R，有了这两个可信度就可以进行综合加权了。

在这个时候，我们只要知道卡尔曼滤波的基本思想是一个加权的过程就可以了，它通过模型进行预测，通过仪器等进行测量，预测值和实测值进行综合加权得到了最优估计。在后面的学习中，我会再慢慢讨论卡尔曼的理论。

二、模型预测

很多同学手上都有一些测量值，想用卡尔曼滤波进行平滑处理，不想去进行数学建模，其关键还是没有理解卡尔曼滤波。如果你只是想平滑数据，那么可以有多种方法，中值滤波、最小二乘、傅里叶变换等都可以，为什么非要用卡尔曼滤波？既然要用卡尔曼滤波，就要按照它的规则去做，其实卡尔曼滤波比最小二乘的思想还是要好一些，不然卡尔曼也不会研究，这个问题后续再探讨。

关于数学建模，要根据自己研究的问题而定，不同的问题模型肯定不一样，同时模型建立的准确与否，直接关系到预测的准确度。所以一个好的模型，对于预测来说，是很重要的，模型最好能够符合物体的变化规律。我所知道的一些模型，包括CA模型、CV模型，还有“当前”模型。



这就是两个基本的状态方程。“当前”模型为周宏仁提出的机动目标跟踪模型，可以百度搜索。

三、实际观测

回到第一节所讲的温度那个例子，其中实际观测值就是温度计输出的数值，在这里我们把温度作为一个状态量进行考虑，观测值与状态量恰好是一个东西，都是温度，这是比较特殊的情况。通常来说，比如我想要得到一个姿态角，它是一个角度，但在实际中我测出来的东西可能是陀螺仪输出的角速度、或者加速度计输出的角加速度，我不能直接观测到角度值。 又比如说，我要得到一个飞行器的空间坐标（X，Y，Z），但我并不能直接观测到它的值，我能得到的数据是雷达测量的距离值，或者多普勒频率值，在这里观测值和状态量就不是一个东西，但是二者之间是有联系，是满足一定函数关系的。这里，我们再来看卡尔曼滤波中的量测方程：



在温度那个例子中，H就是单位矩阵，因为观测值就是温度，而不是其他东西。但如果观测值不是状态量，那H就不再是单位矩阵。在上式中，我们称为状态量，为观测量，二者是有函数关系的。比如我知道了k时刻的状态量（X，Y，Z），我就可以计算出雷达的测距值r1，两点之间的距离公式嘛，于此同时在k时刻还有一个雷达实际的测距值r2，r1与r2一般是不相等的，用卡尔曼增益K\*（r1- r2）就得到了在状态量（X，Y，Z）上的偏差，简单来说就是观测值上的偏差反映在状态量上偏差了多少。比如距离变化了10m，可能在X方向只变化了0.1m，打个比方而已。

所以说我们一定要理解观测值和状态量之间的关系，通常来说都是非线性的。其中还有一项重要的内容，就是我们前面提到的观测值的可信度，我只是拿可信度打个比方，为了方便大家理解而已。能够反映观测值可信度的，其实就是R，随机误差V(k)的协方差矩阵。这个R越小，说明观测值的随机误差越小，既然这样，那就说明观测值准确可靠啊，反映在状态量上，也就是对应的权值要增大。我们前面提到的权值分配问题，怎么分配权重，就看这个，看预测值和观测值对应的随机误差协方差矩阵P和R。一般来说，测量误差包含系统误差和随机误差，为什么只看随机误差就可以了，因为卡尔曼滤波假设数据的期望都是0，即测量数据中系统误差为0，只含随机误差，所以说只要知道随机误差的大小就能得出相应的权值。

四、最优估计

为了进一步说明，我举一个例子，考虑卡尔曼滤波方程的原始问题，有两个方程，状态方程和量测方程：

 （1）

 （2）

上两式子中，是k时刻的系统状态，是k时刻对系统的控制量。A和B是系统参数。是k时刻的测量值，H是测量系统的参数。和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(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）。

按照卡尔曼滤波的5个公式，得到的最优估计为

 （3）

其中

 （4）

 （5）

下面我们举一个特例，也就是温度的例子。

假定是线性关系，没有控制量可以为0，因为观测值就是状态量，所以H=I为单位矩阵，你把H=I带入到式(3)和(4) 中去，会得到下面的公式：





在这里面就是预测值，用代替；是观测值，用代替，可以进一步得到



到这里，是不是很熟悉了，预测和实测综合加权的结果。

五、卡尔曼5个公式

对于卡尔曼的5个公式，如下所示：

 (4-1)

 (4-2)

 (4-3)

 (4-4)

 (4-5)

相信现在大家应该基本明白卡尔曼滤波的过程了，再简单熟悉一下：

第一个公式（4-1）：这个就是根据模型进行预测，知道k-1时刻的状态量X，就可以预测出k时刻的状态量了，直接运算就OK了。

第二个公式（4-2）：这个就是计算预测值的误差协方差阵了，看看预测的准确度如何，为后面分配权重做准备。

第三个公式（4-3）：这个就是进行最优估计。利用预测状态信息反算观测量，然后计算实测值与反算值之差，再进行综合加权，当然应该先计算（4-4）式。

第四个公式（4-4）：这个就是计算卡尔曼增益，也就是给实测和预测分配权重。

第五个公式（4-5）：计算出了最优估计结果，当然应该对结果进行一下评价，计算一下最优估计的准确度如何，也是为了k+1时刻做准备，保证后续可以一直循环下去。

可以回答一下前面的问题了，关于最小二乘的问题。

当状态量和观测值是同一个东西时，可以不用卡尔曼滤波，直接对观测量采用最小二乘

进行拟合或者其他方法的数据平滑，都是可以的，最小二乘也是一种最优估计，使得总误差

达到最小。

但是当状态量和观测量不是同一个东西时，我想知道飞机的位置坐标，但我的观测量只

有距离，这个时候总不能再用最小二乘或数据平滑了吧，采用常规的几何求解是一种方法，

另外一种方法就是卡尔曼滤波，给定一个初值，就可以一直预测下去，结合实际的观测量再

进行修正，不断预测修正再预测的过程，关键是模型要准确，要能够收敛。

六、其他内容

卡尔曼滤波过程中初始时刻的状态量、初始的随机误差矩阵、还有实测的随机误差矩阵

都是需要人为给定的，这些值的大小直接关系到卡尔曼滤波的效果。通常来说，实测的随机

误差，可以通过仪器的指标获得，或者直接统计测量数据的方差。

由于状态模型经常是非线性的，不满足卡尔曼滤波的条件，所以就进行了各种扩展研究，

主要包括EKF、UKF、PF等

EKF（扩展卡尔曼滤波）：主要就是对非线性模型进行泰勒展开，保留一次项，略去高阶项，这样模型就是线性方程了，就可以按照卡尔曼滤波的套路去做了，不好的地方就是总要计算雅克比矩阵。

UKF（无迹卡尔曼滤波）：主要就是做了一个UT变换，用2N+1个sigma点都进行状态预测，在得到2N+1个预测值后进行加权平均，这个平均值就作为最终的预测结果。在进行反算观测量时，再做一次类似的变换。

PF（粒子滤波）：这个主要就是用大量的粒子去做蒙特卡洛实验，以逼近想要得到的概率密

度。不是太懂，它的主要思想是基于大数定理，足够多次试验以后就接近

了真实的概率，好处就是不用进行积分等复杂的运算，不足就是需要试验

的次数要太多，实时性不强。

联邦滤波：二级或者多级滤波器，子滤波器将信息传递给主滤波器，主滤波器获得了最多信息，从而再反馈信息给子滤波器进行控制，具体没用过，不再细说，有兴趣的自己再学习吧。

以上仅为个人的一点心得体会，不足之处还望批评指正，欢迎大家讨论交流，共同进步。

trick

2016-1-10