曼滤波原理

4:15 engineerlixl 阅读数：38546 标签： 卡尔曼滤波 更多

详解卡尔曼滤波原理

›与卡尔曼滤波相关的博客、论文，要么是只谈理论、缺乏感性，或者有感性认识，缺乏理论推导。能兼顾二者的少之又少，直到我看到了国 一篇博文 ﹔者这种细致入微的精神，翻译过来跟大家分享一下，原文链接：http://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/

:尔曼滤波，因为它能做到的事情简直让人惊叹！意外的是很少有软件工程师和科学家对对它有所了解，这让我感到沮丧，因为卡尔曼滤波是一个如此强大 ﹔息，与此同时，它提取精确信息的能力看起来不可思议。

波?

;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。 **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

训的大多数关于实现卡尔曼滤波的数学公式看起来有点晦涩难懂，这个状况有点糟糕。实际上，如果以正确的方式看待它，卡尔曼滤波是非常简单和容易]色彩清晰的阐述它，你只需要懂一些基本的概率和矩阵的知识就可以了。

滤波做什么?

;开发了一个可以在树林里到处跑的小机器人，这个机器人需要知道它所在的确切位置才能导航。

﹔人有一个状态 $\vec{x_k}$ ，表示位置和速度：

$$\vec{x_k} = (\vec{p}, \vec{v})$$

]是关于这个系统基本属性的一堆数字，它可以是任何其它的东西。在这个例子中是位置和速度，它也可以是一个容器中液体的总量，汽车发动机的温度， 或者任何你需要跟踪的信号。

﹔GPS，精度大约为10米，还算不错，但是，它需要将自己的位置精确到10米以内。树林里有很多沟壑和悬崖，如果机器人走错了一步，就有可能掉下悬崖

一些机器人如何运动的信息：例如，机器人知道发送给电机的指令，知道自己是否在朝一个方向移动并且没有人干预，在下一个状态，机器人很可能朝着
]自己的运动是一无所知的：它可能受到风吹的影响，轮子方向偏了一点，或者遇到不平的地面而翻倒。所以，轮子转过的长度并不能精确表示机器人实际

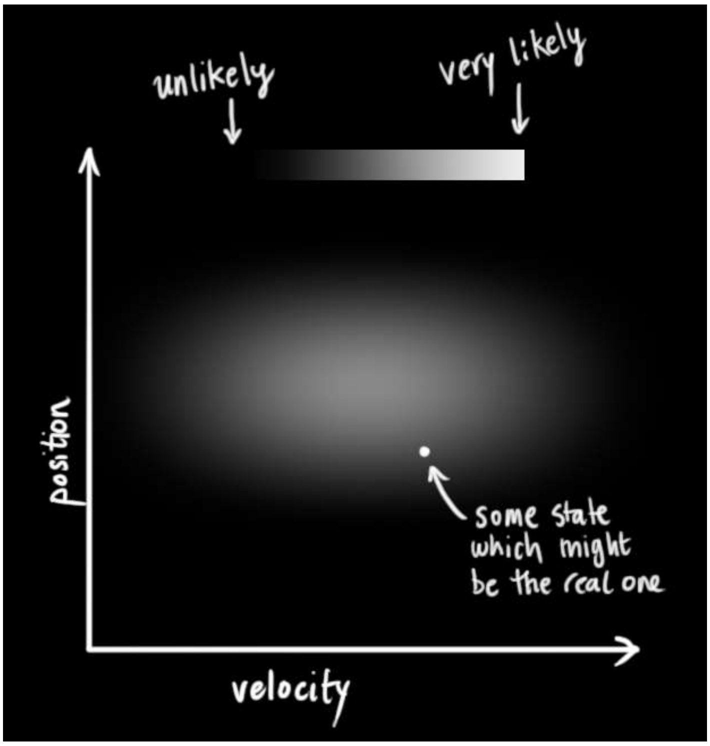
斥了我们一些状态信息，我们的**预测**告诉了我们机器人会怎样运动，但都只是间接的，并且伴随着一些不确定和不准确性。但是，如果使用……时我们可
]依据自身估计更好的结果吗？回答当然是YES，这就是卡尔曼滤波的用处。

何看到你的问题的

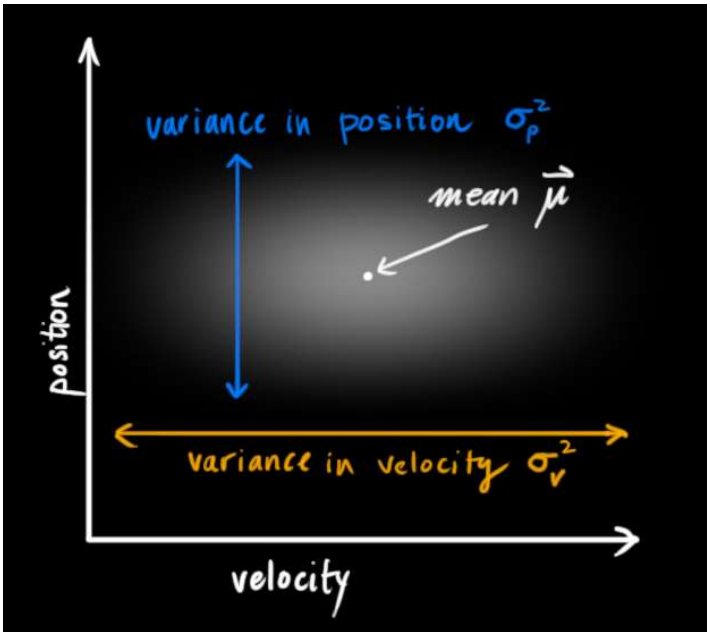
l只有位置和速度这两个状态的简单例子做解释。

$$\vec{x} = \begin{bmatrix} p \\ v \end{bmatrix}$$

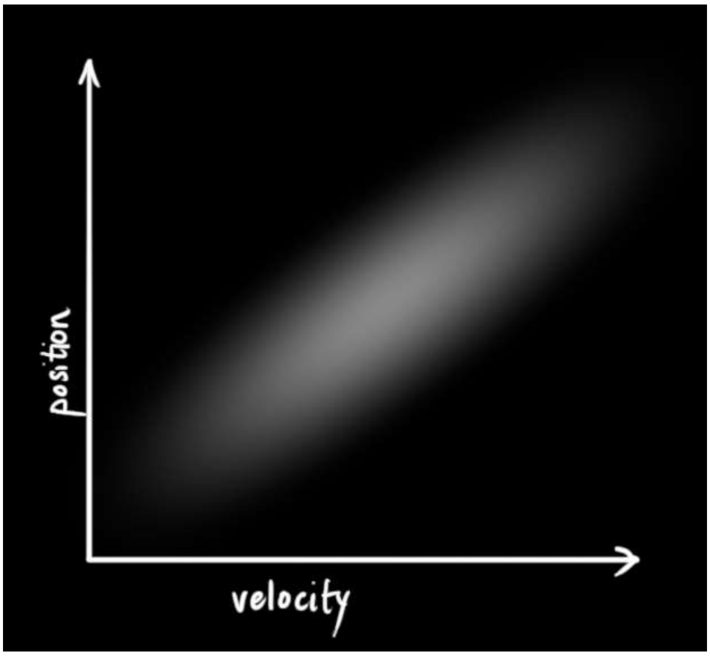
:际的位置和速度，它们之间有很多种可能正确的组合，但其中一些的可能性要大于其它部分：



:两个变量（位置和速度，在这个例子中）都是随机的，并且***服从高斯分布***。每个变量都有一个**均值** μ，表示随机分布的中心（最可能的状态），以及**方差**



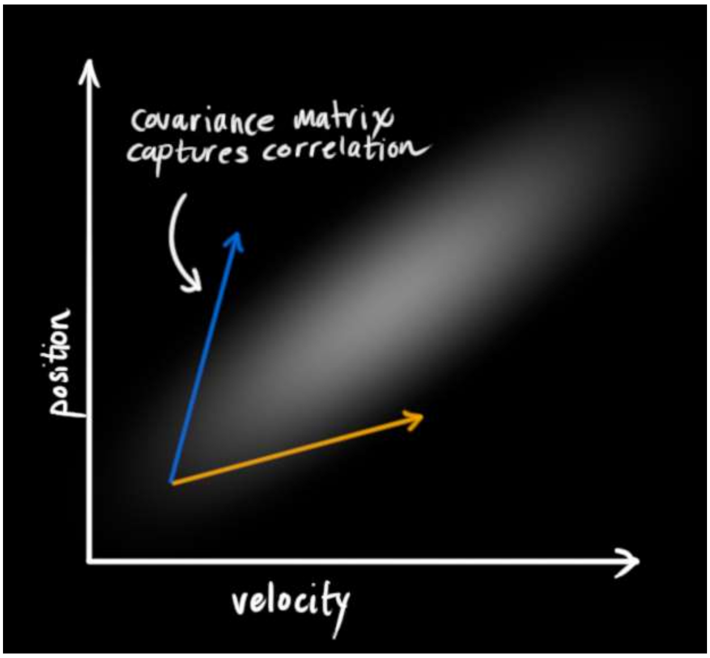
置和速度是**不相关**的，这意味着由其中一个变量的状态无法推测出另一个变量可能的值。下面的例子更有趣：位置和速度是**相关**的，观测特定位置的可能性



J能发生的，例如，我们基于旧的位置来估计新位置。如果速度过高，我们可能已经移动很远了。如果缓慢移动，则距离不会很远。跟踪这种关系是非常重
‌‌‌‌:：其中一个测量值告诉了我们其它变量可能的值，这就是卡尔曼滤波的目的，尽可能地在包含不确定性的测量数据中提取更多信息！

)方差矩阵来表示，简而言之，矩阵中的每个元素 Σ_{ij} 表示第 i 个和第 j 个状态变量之间的相关度。（你可能已经猜到协方差矩阵是一个**对称矩阵**，这意味

巨阵通常用“ Σ ”来表示，其中的元素则表示为“ Σ_{ij} ”。

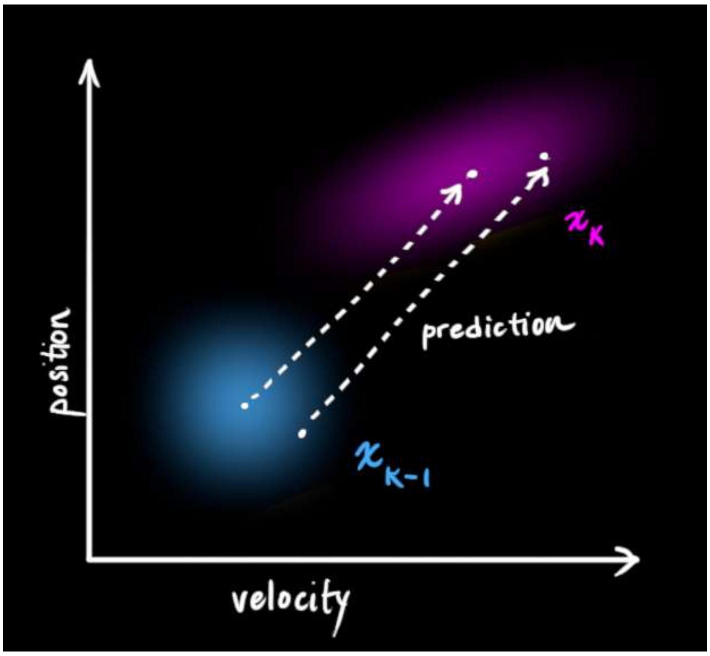


问题

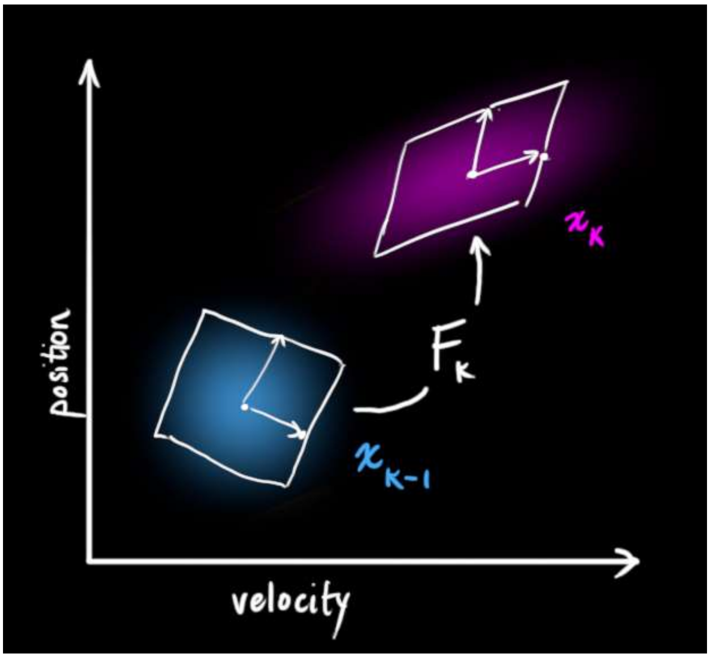
布来建立状态变量，所以在时刻 k 需要两个信息：最佳估计 $\hat{x}_k$（即均值，其它地方常用 μ 表示），以及协方差矩阵 P_k。

$$\hat{x}_k = \begin{bmatrix} position \\ velocity \end{bmatrix}, P_k = \begin{bmatrix} \Sigma_{pp} & \Sigma_{pv} \\ \Sigma_{vp} & \Sigma_{vv} \end{bmatrix} \quad (1)$$

我们只用到了位置和速度，实际上这个状态可以包含多个变量，代表任何你想表示的信息）。接下来，我们需要根据**当前状态**（**k-1** 时刻）来**预测下**
不知道对下一状态的所有预测中哪个是“真实”的，但我们的预测函数并不在乎。它对所有的可能性进行预测，并给出新的高斯分布。



F_k 来表示这个预测过程：



计中的每个点都移动到了一个新的预测位置，如果原始估计是正确的话，这个新的预测位置就是系统下一步会移动到的位置。那我们又如何用矩阵来预测
用一个基本的运动学公式来表示：

$$
\begin{aligned}
p_k &= p_{k-1} + \Delta t \, v_{k-1} \\
v_k &= \qquad\quad\;\; v_{k-1}
\end{aligned}
$$

In other words:

$$
\begin{aligned}
\hat{\mathbf{x}}_k &= \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \hat{\mathbf{x}}_{k-1} && (2) \\
&= \mathbf{F}_k \hat{\mathbf{x}}_{k-1} && (3)
\end{aligned}
$$

一个预测矩阵来表示下一时刻的状态，但是，我们仍然不知道怎么更新协方差矩阵。此时，我们需要引入另一个公式，如果我们将分布中的每个点都乘以

会怎样变化呢？很简单，下面给出公式：

$$
\begin{aligned}
Cov(x) &= \Sigma \\
Cov(\mathbf{A}x) &= \mathbf{A}\Sigma\mathbf{A}^T
\end{aligned} \qquad (4)
$$

和（3）得到：

$$
\begin{aligned}
\hat{\mathbf{x}}_k &= \mathbf{F}_k \hat{\mathbf{x}}_{k-1} \\
\mathbf{P}_k &= \mathbf{F}_k \mathbf{P}_{k-1} \mathbf{F}_k^T
\end{aligned} \qquad (5)
$$

到一切信息，可能存在外部因素会对系统进行控制，带来一些与系统自身状态没有相关性的改变。
态模型为例，火车司机可能会操纵油门，让火车加速。相同地，在我们机器人这个例子中，导航软件可能会发出一个指令让轮子转向或者停止。如果知道

丶向量 $\vec{\mathbf{u}_k}$ 来表示，将它加到我们的预测方程中做修正。

]设置或控制命令，我们知道了期望的加速度 a ，根据基本的运动学方程可以得到：

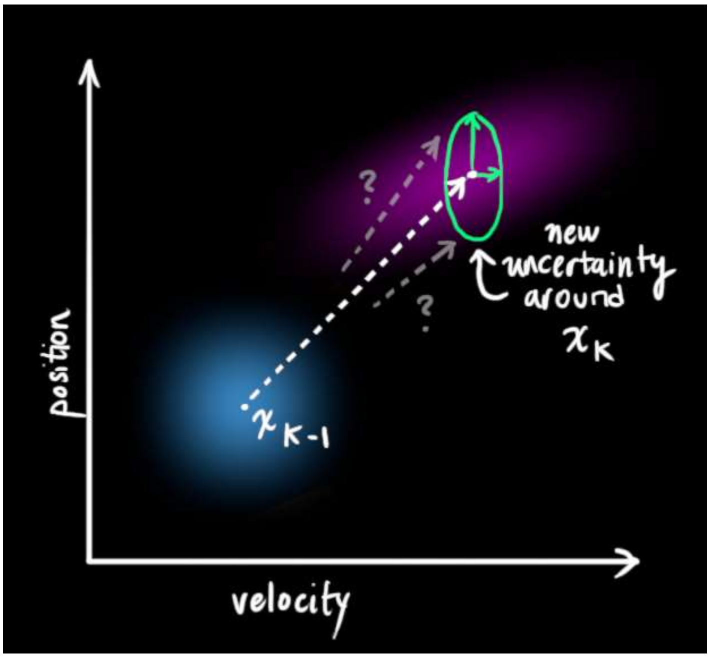
$$\begin{aligned} p_k &= p_{k-1} + \Delta t\, v_{k-1} + \frac{1}{2} a \Delta t^2 \\ v_k &= \qquad\qquad\quad v_{k-1} + a\Delta t \end{aligned}$$

表示就是：

$$\begin{aligned} \hat{\mathbf{x}}_k &= \mathbf{F}_k \hat{\mathbf{x}}_{k-1} + \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix} a \\ &= \mathbf{F}_k \hat{\mathbf{x}}_{k-1} + \mathbf{B}_k \vec{\mathbf{u}_k} \end{aligned} \tag{6}$$

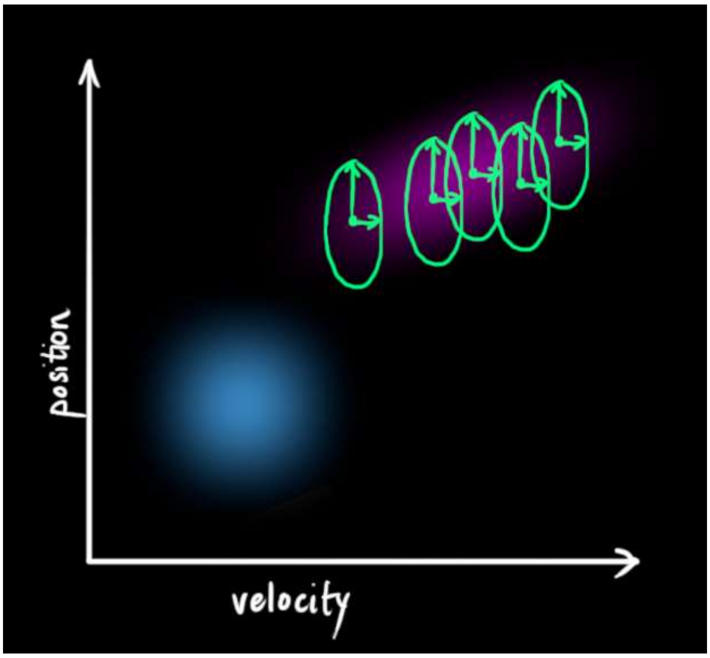
阵， $\vec{\mathbf{u}_k}$ 称为控制向量（对于没有外部控制的简单系统来说，这部分可以忽略）。让我们再思考一下，如果我们的预测并不是100%准确的，该怎么办呢？

是基于系统自身的属性或者已知的外部控制作用来变化的，则不会出现什么问题。
三未知的干扰呢？例如，假设我们跟踪一个四旋翼飞行器，它可能会受到风的干扰，如果我们跟踪一个轮式机器人，轮子可能会打滑，或者路面上的小坡会
续对这些状态进行跟踪，如果没有把这些外部干扰考虑在内，我们的预测就会出现偏差。
5，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：

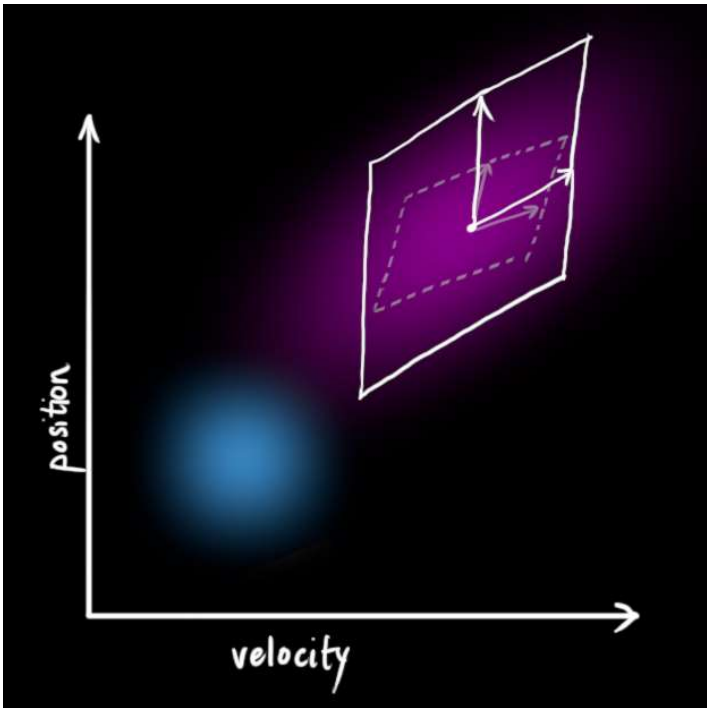


寻个状态变量更新到新的状态后，仍然服从高斯分布。我们可以说 $\hat{\mathbf{x}}_{k-1}$ 的每个状态变量移动到了一个新的服从高斯分布的区域，协方差为 $\mathbf{Q}_k$ 。换句话说

干扰当作协方差为 $\mathbf{Q}_k$ 的**噪声**来处理。



同协方差（但是具有相同的均值）的新的高斯分布。



添加 $\mathbf{Q}_k$ 得到扩展的协方差，下面给出预测步骤的完整表达式：

$$
\begin{aligned}
\hat{\mathbf{x}}_k &= \mathbf{F}_k \hat{\mathbf{x}}_{k-1} + \mathbf{B}_k \overrightarrow{\mathbf{u}_k} \\
\mathbf{P}_k &= \mathbf{F}_k \mathbf{P}_{k-1} \mathbf{F}_k^T + \mathbf{Q}_k
\end{aligned}
\tag{7}
$$

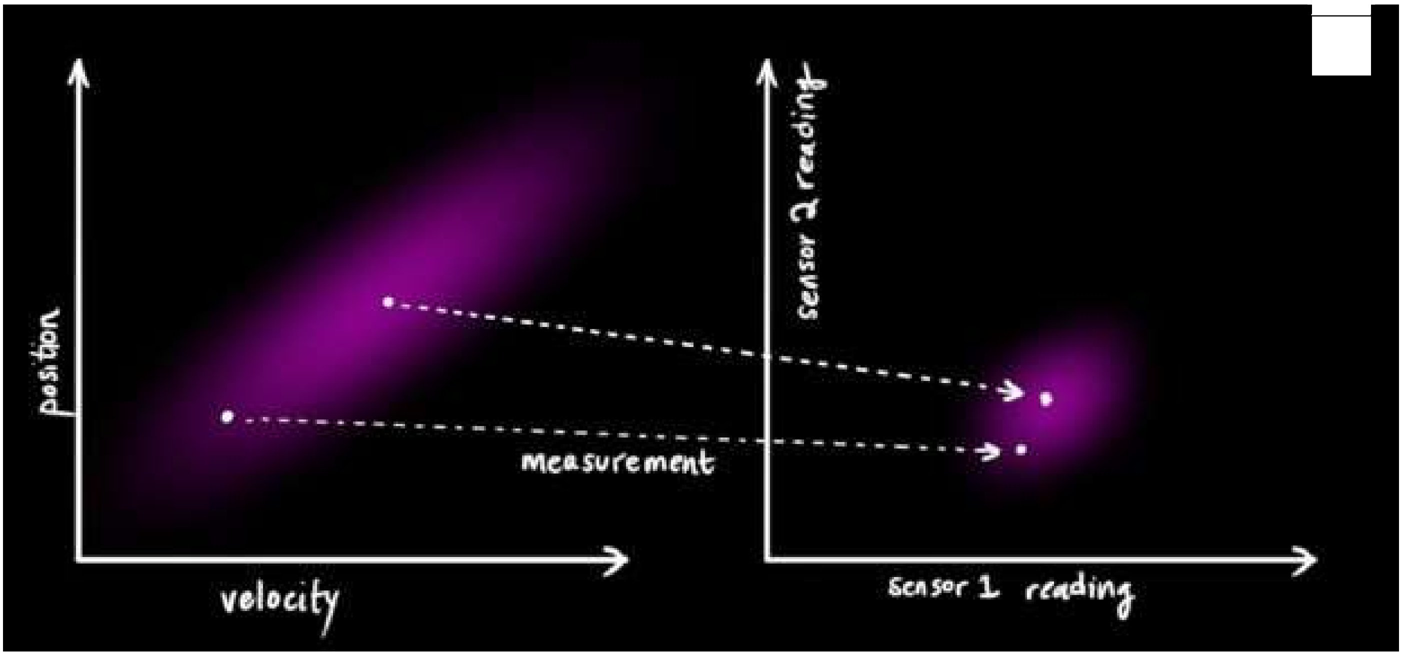
的最优估计是根据**上一最优估计预测**得到的，并加上**已知外部控制量的修正。**

由**上一不确定性预测**得到，并加上**外部环境的干扰。**

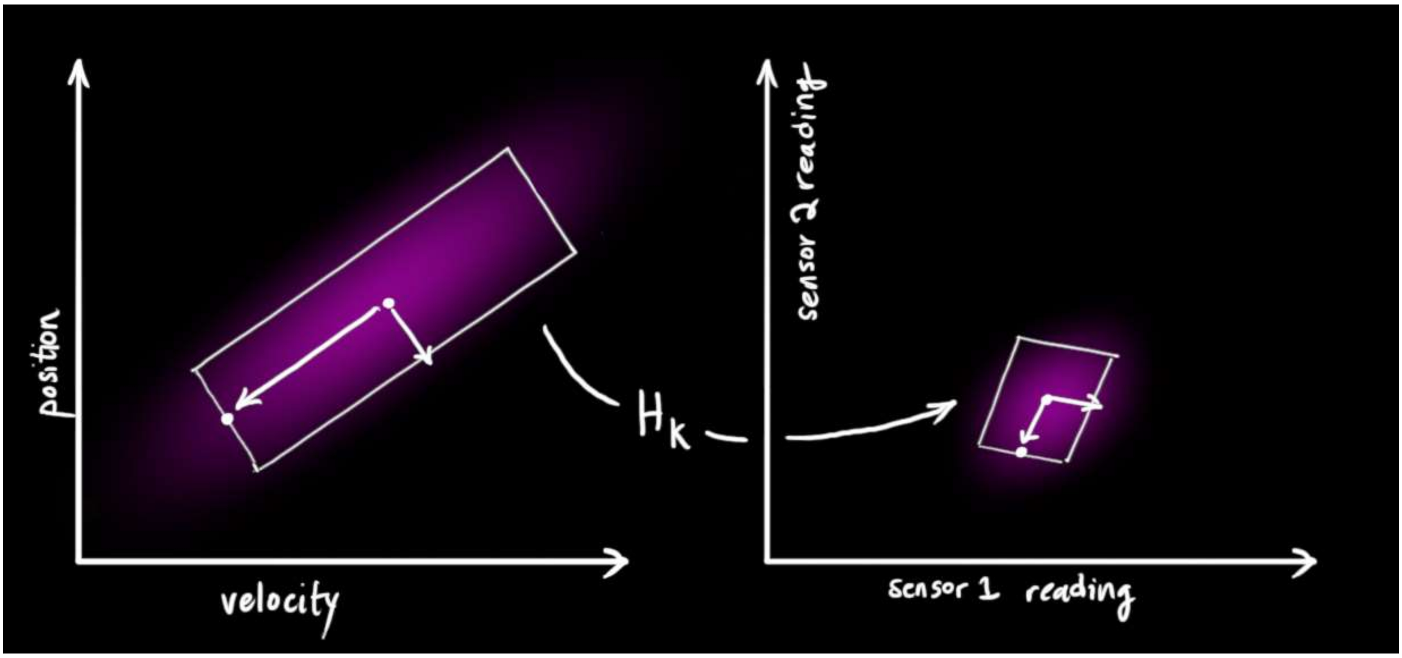
:统可能的动向有了一个模糊的估计，用 $\hat{\mathbf{x}}_k$ 和 $\mathbf{P}_k$ 来表示。如果再结合传感器的数据会怎样呢？

估计值

;个传感器来测量系统当前的状态，哪个传感器具体测量的是哪个状态变量并不重要，也许一个是测量位置，一个是测量速度，每个传感器间……告诉了我



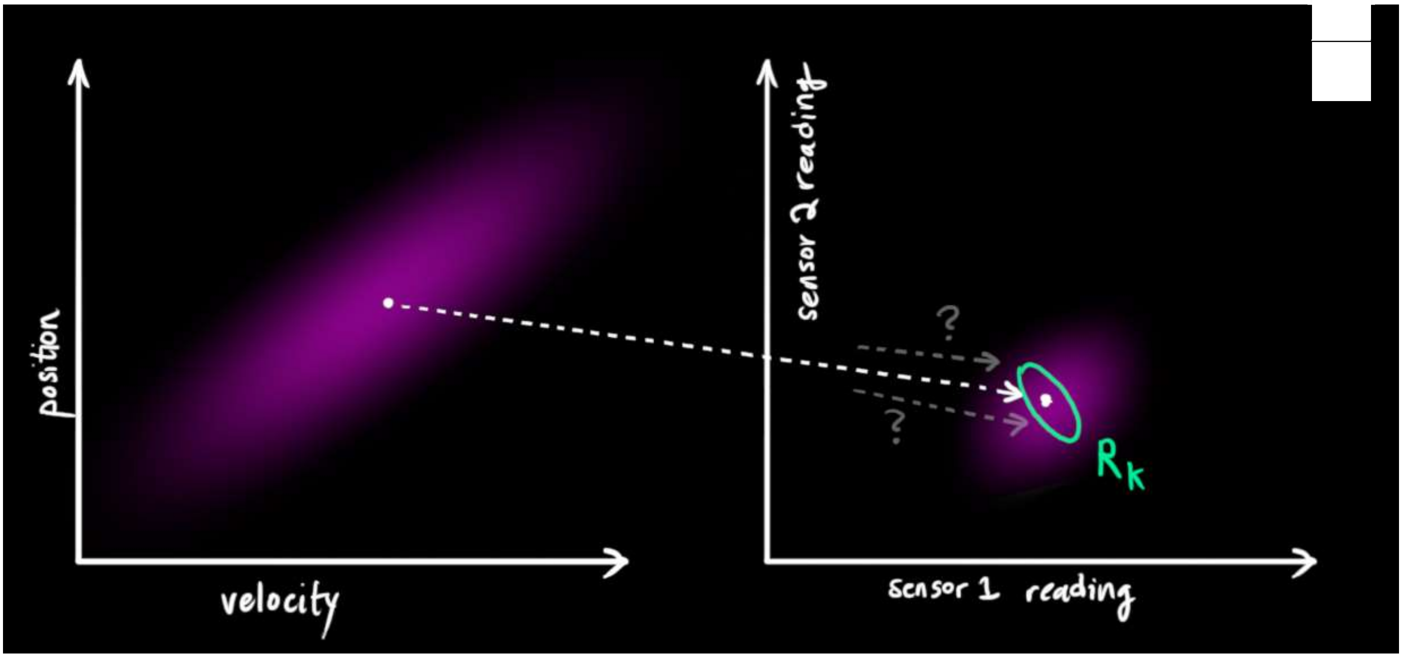
:取的数据的单位和尺度有可能与我们要跟踪的状态的单位和尺度不一样，我们用矩阵 H_k 来表示传感器的数据。



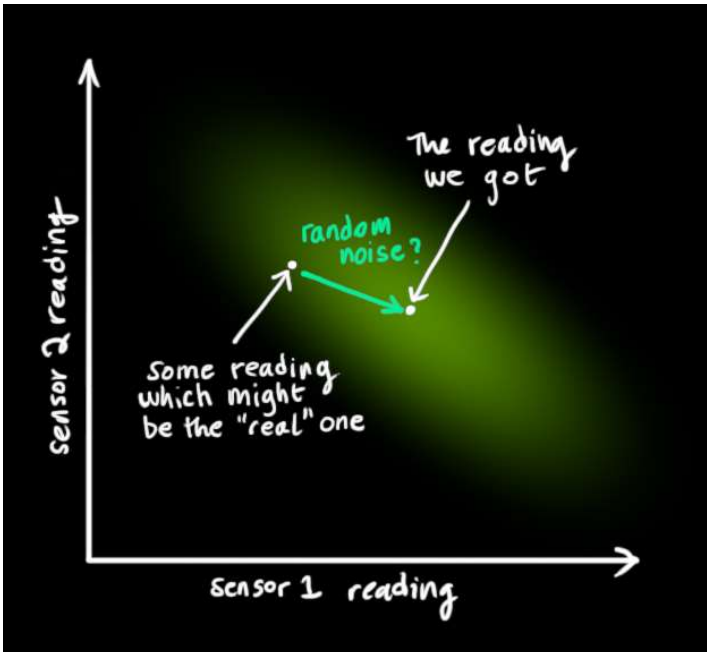
:传感器读数的分布，用之前的表示方法如下式所示：

$$
\begin{aligned}
\vec{\mu}_{\text{expected}} &= \mathbf{H}_k \hat{\mathbf{x}}_k \\
\mathbf{\Sigma}_{\text{expected}} &= \mathbf{H}_k \mathbf{P}_k \mathbf{H}_k^T
\end{aligned} \tag{8}
$$

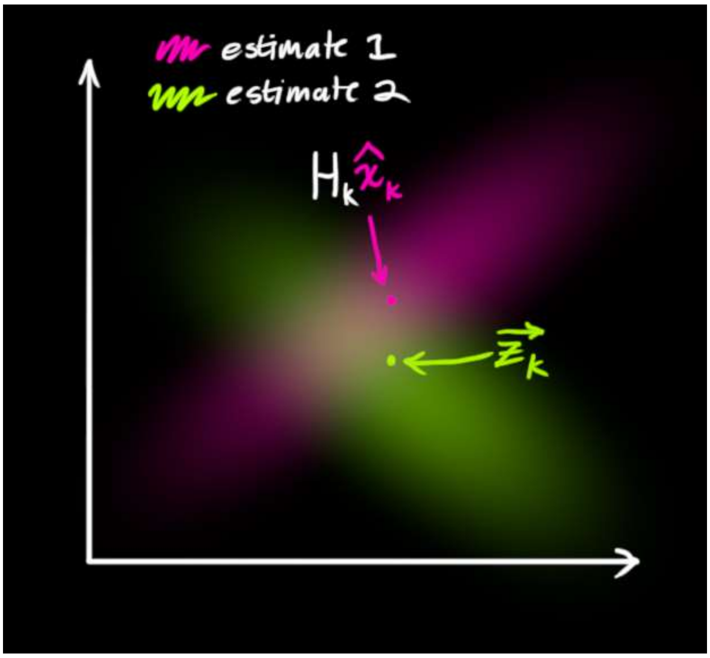
-大优点就是能处理传感器噪声，换句话说，我们的传感器或多或少都有点不可靠，并且原始估计中的每个状态可以和一定范围内的传感器读 起来。



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。

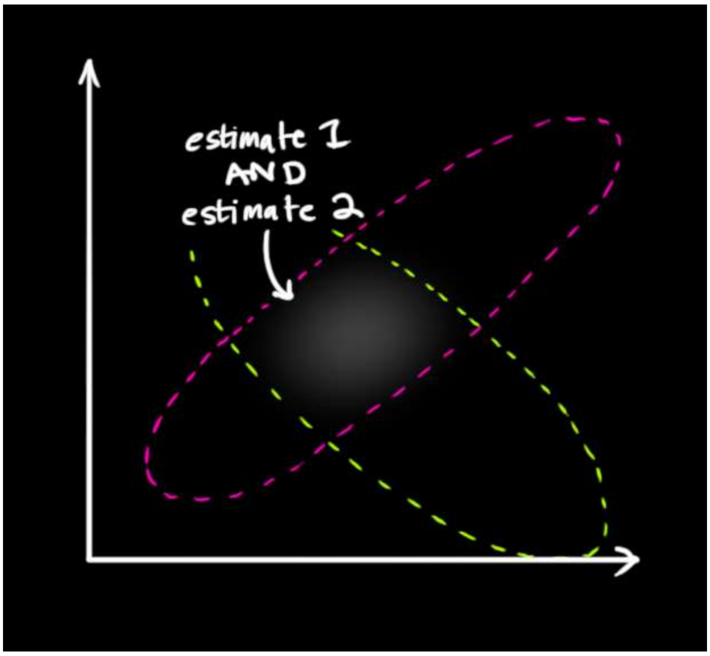


定性（例如：传感器噪声）用**协方差** $\mathbf{R}_k$ 表示，该分布的**均值**就是我们读取到的传感器数据，称之为 $\vec{\mathbf{z}_k}$。
斯分布，一个是在预测值附近，一个是在传感器读数附近。

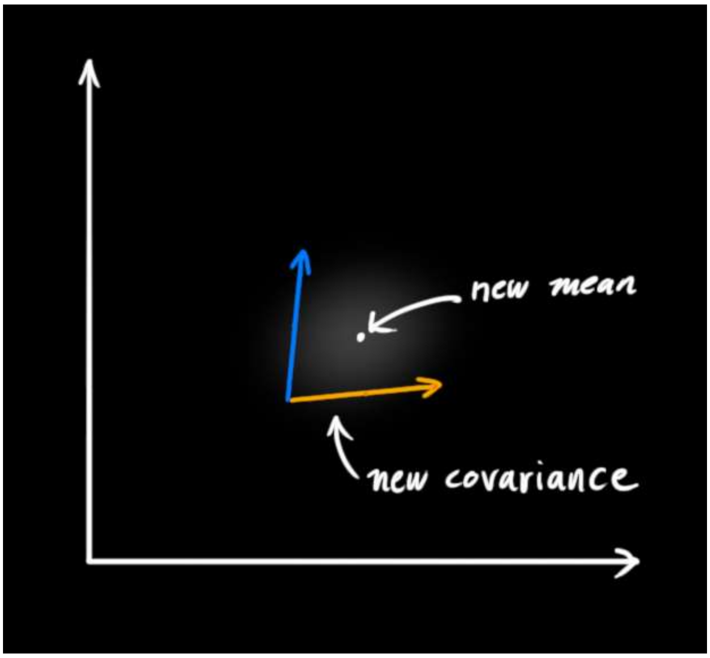


值（**粉红色**）和**传感器测量值**（**绿色**）之间找到最优解。

可能的状态是什么呢？对于任何可能的读数 (z_1, z_2)，有两种情况：（1）传感器的测量值；（2）由前一状态得到的预测值。如果我们想知道这两种情况分布相乘就可以了。



部分了，这个重叠部分的均值就是两个估计最可能的值，也就是给定的所有信息中的**最优估计**。
区域看起来像另一个高斯分布。

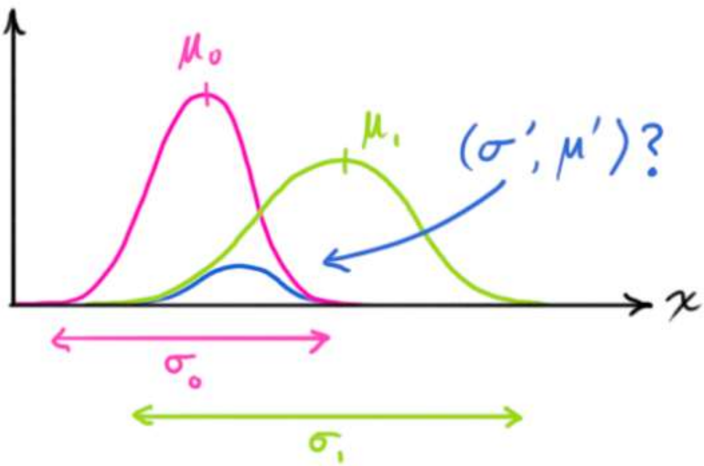


i个具有不同均值和方差的高斯分布相乘，你会得到一个新的具有独立均值和方差的高斯分布！下面用公式讲解。

)布来分析比较简单点，具有方差 σ^2 和 μ 的高斯曲线可以用下式表示：

$$\mathcal{N}(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{9}$$

、高斯分布的函数相乘会得到什么呢？



$$\mathcal{N}(x, \mu_0, \sigma_0) \cdot \mathcal{N}(x, \mu_1, \sigma_1) \stackrel{?}{=} \mathcal{N}(x, \mu', \sigma') \tag{10}$$

到式（10）中（注意重新归一化，使总概率为1）可以得到：

$$
\begin{aligned}
\mu' &= \mu_0 + \frac{\sigma_0^2(\mu_1 - \mu_0)}{\sigma_0^2 + \sigma_1^2} \\
\sigma'^2 &= \sigma_0^2 - \frac{\sigma_0^4}{\sigma_0^2 + \sigma_1^2}
\end{aligned} \tag{11}
$$

]两个式子相同的部分用 **k** 表示：

$$
\mathbf{k} = \frac{\sigma_0^2}{\sigma_0^2 + \sigma_1^2} \tag{12}
$$

$$
\begin{aligned}
\mu' &= \mu_0 + \mathbf{k}(\mu_1 - \mu_0) \\
\sigma'^2 &= \sigma_0^2 - \mathbf{k}\sigma_0^2
\end{aligned} \tag{13}
$$

弌（12）和（13）写成矩阵的形式，如果 **Σ** 表示高斯分布的协方差，$\vec{\mu}$ 表示每个维度的均值，则：

$$
\mathbf{K} = \Sigma_0(\Sigma_0 + \Sigma_1)^{-1} \tag{14}
$$

$$
\begin{aligned}
\vec{\mu}' &= \vec{\mu_0} + \mathbf{K}(\vec{\mu_1} - \vec{\mu_0}) \\
\Sigma' &= \Sigma_0 - \mathbf{K}\Sigma_0
\end{aligned} \tag{15}
$$

ℛ曼增益，下面将会用到。放松！我们快要完成了！

起来

彳分布，预测部分 $(\mu_0, \Sigma_0) = (\mathbf{H}_k\hat{\mathbf{x}}_k, \mathbf{H}_k\mathbf{P}_k\mathbf{H}_k^T)$ ，和测量部分 $(\mu_1, \Sigma_1) = (\vec{\mathbf{z}_k}, \mathbf{R}_k)$ ，将它们放到式（15）中算出它们之间的重叠部分：

$$
\begin{aligned}
\mathbf{H}_k\hat{\mathbf{x}}_k' &= \mathbf{H}_k\hat{\mathbf{x}}_k + \mathbf{K}(\vec{\mathbf{z}_k} - \mathbf{H}_k\hat{\mathbf{x}}_k) \\
\mathbf{H}_k\mathbf{P}_k'\mathbf{H}_k^T &= \mathbf{H}_k\mathbf{P}_k\mathbf{H}_k^T - \mathbf{K}\mathbf{H}_k\mathbf{P}_k\mathbf{H}_k^T
\end{aligned} \tag{16}
$$

寻卡尔曼增益为：

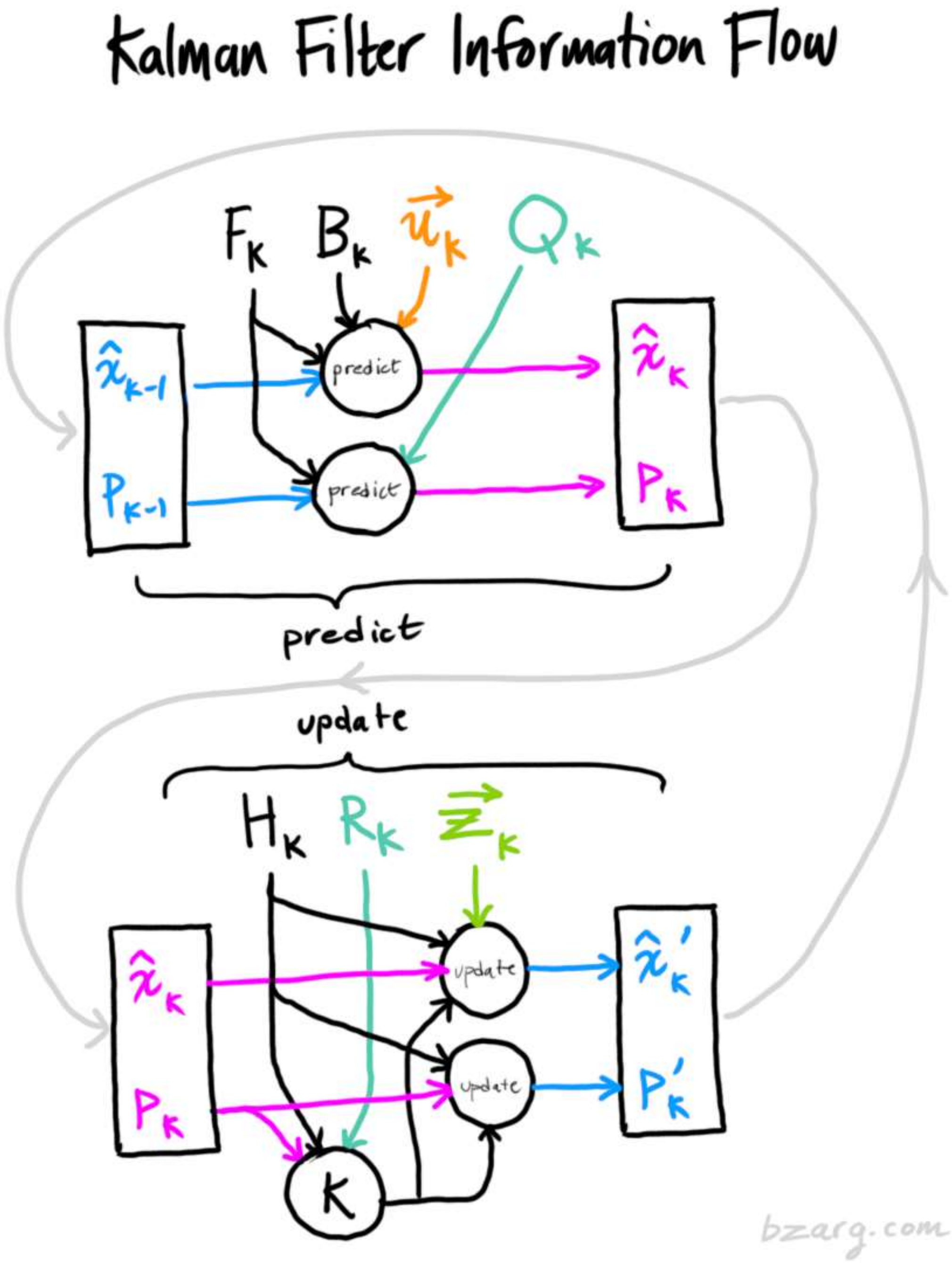
$$
\mathbf{K} = \mathbf{H}_k\mathbf{P}_k\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_k\mathbf{H}_k^T + \mathbf{R}_k)^{-1} \tag{17}
$$

弋（17）的两边同时左乘矩阵的逆（注意 $\mathbf{K}$ 里面包含了 H_k ）将其约掉，再将式（16）的第二个等式两边同时右乘矩阵 H_k^T 的逆得到以下等式：

$$
\begin{aligned}
\hat{\mathbf{x}}_k' &= \hat{\mathbf{x}}_k + \mathbf{K}'(\vec{\mathbf{z}_k} - \mathbf{H}_k\hat{\mathbf{x}}_k) \\
\mathbf{P}_k' &= \mathbf{P}_k - \mathbf{K}'\mathbf{H}_k\mathbf{P}_k
\end{aligned} \tag{18}
$$

$$
\mathbf{K}' = \mathbf{P}_k\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_k\mathbf{H}_k^T + \mathbf{R}_k)^{-1} \tag{19}
$$

的**更新步骤**方程。 $\hat{\mathbf{x}}'_k$ 就是新的最优估计，我们可以将它和 $\mathbf{P}'_k$ 放到下一个**预测**和**更新**方程中不断迭代。



，你只需要用到式（7）、（18）、（19）。（如果忘了的话，你可以根据式（4）和（15）重新推导一下）

公式对任何线性系统建立精确的模型，对于非线性系统来说，我们使用**扩展卡尔曼滤波**，区别在于EKF多了一个把预测和测量部分进行线性化的过程。

down，添加图片和公式心累啊，什么时候能直接拖拽就好了~~)

巧:

号与颜色。参考链接：(http://blog.csdn.net/testcs_dn/article/details/45719357/)

编辑好了右键“复制图片地址”，当作图片来添加。

.codecogs.com/latex/eqneditor.php)

Shift+Space将输入法切换到全角状态，然后敲空格即可，一个空格代表一个汉字的间隔。

中显示。参考链接：(http://blog.csdn.net/soindy/article/details/50427079)

57

37

么

(09-21 16:35 #26楼)

通俗易懂，谢谢博主的无私分享 (09-07 09:15 #25楼)

键观测矩阵能求逆吗？这个文章更多的是从便于理解的角度出发的吧？ (09-04 15:09 #24楼)

麻烦作者将这种好文章放在知乎或者其他分享网站上吧,CSDN广告太多,体验真是垃圾 (09-03 09:58 #23楼)

查看 37 条热评

波及其算法实现（实例解析） 3.3万

tion)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...

推导到应用(一) 9.8万

E估计线性系统状态的过程中，以最小均方差为目的而推导出的几个递推数学等式,也可以...

曼滤波算法』 3.1万

自： 1.http://blog.csdn.net/karen99/article/details/7771743 2.http://blog.csdn.net/tudou...

解卡尔曼滤波的三重境界 7168

设我养了一只猪： 一周前，这只猪的体重是46±0.5kg。注意，在这里我用了±0.5，表示...

解释与实现 993

持：http://www.cnblogs.com/ycwang16/p/5999034.html 认知计算，还要从贝叶斯滤波...

理解 868

波器？？？ 假设我们要跟踪一个穿过摄像机视场的人。每一帧都要确定这个人的位置...

论.pdf 04-24

特卡洛(Monte Carlo)模拟方法来实现递推贝叶斯滤波，适用 于任何能用状态空间模型描述的非线性系统，精度可以逼近最优估计。粒子滤波器具有简单、...

（通俗易懂）

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

准差，与其意义

式 协方差的意义和计算公式 学过概率统计的孩子都知道，统计里最基本的概念就是样本的均值，方差，或者再加个标准差。首先我们给你一个含有n个样本的集合，依次给出...

解及C语言代码

位，需要对RSSI值进滤波，采用卡尔曼滤波。网上看了很多卡尔曼滤波的文章，为了加深自己的印象。所以打算结合网上的资料加上自己理解方式写一篇文章（如果不对还请您

if详解 详解》 this详解 on详解

engineerlixl 关注

最新文章

基于光流传感器定位和导航的自主飞行无人机

如果地球上挖了一个洞，出口在地球的另一端，一个人跳进洞中，请问他会怎样运动？

光流定点程序梳理

光流定点若干问题分析

为什么人不能永生？

个人分类

飞控	4篇
笔记	1篇
哲思	1篇
脑洞大开	1篇
翻译	2篇

归档

2017年3月	1篇
2017年1月	8篇
2016年7月	1篇

热门文章

详解卡尔曼滤波原理
阅读量：38510

基于光流传感器定位和导航的自主飞行无人机
阅读量：14551

光流定点程序梳理
阅读量：3964

APM添加超声模块及定高程序分析
阅读量：2562

光流定点若干问题分析
阅读量：2349

最新评论

详解卡尔曼滤波原理
weixin_43230212：mark

详解卡尔曼滤波原理

m0_37362454：[reply]weixin_41692946[/reply] 你说的对，确实很多情况，实际处理情况...

详解卡尔曼滤波原理

hkf136400287：通俗易懂，谢谢博主的无私分享

详解卡尔曼滤波原理

zephyr_john：关键观测矩阵能求逆吗？ 这个文章更多的是从便于理解的角度出发的吧?

详解卡尔曼滤波原理

chenwisdom1：麻烦作者将这种好文章放在知乎或者其他分享网站上吧,CSDN广告太多,体验真是垃圾