

卡尔曼滤波器

Kalman Filter

learning note For reading translation

我真的不懂忧郁



卡尔曼滤波器

Kalman Filter
learning note For reading translation

by

我真的不懂忧郁

Student Name	Student Number
--------------	----------------

First Surname	1234567
---------------	---------

Instructor:	I. Surname
Teaching Assistant:	I. Surname
Project Duration:	Month, Year - Month, Year
Faculty:	Faculty of Aerospace Engineering, Delft

Cover: Canadarm 2 Robotic Arm Grapples SpaceX Dragon by NASA under
CC BY-NC 2.0 (Modified)

Style: TU Delft Report Style, with modifications by Daan Zwaneveld

Preface

A preface...

我真的不懂忧郁
Delft, June 2024

Summary

A summary...

目录

Preface	i
Summary	ii
Nomenclature	iv
1 Preview	1
1.1 协方差矩阵	1
1.2 矩阵微分	1
2 Kalman Filter	2
2.1 基本问题	2
2.2 数据融合	3
2.3 后验估计	3
2.4 算法	5
3 Generalize Kalman Filter	6
4 Kalman Filter 的实例	7
References	8
A Source Code Example	9
B Task Division Example	10

Nomenclature

If a nomenclature is required, a simple template can be found below for convenience. Feel free to use, adapt or completely remove.

Abbreviations

Abbreviation	Definition
ISA	International Standard Atmosphere
...	

Symbols

Symbol	Definition	Unit
V	Velocity	[m/s]
...		
ρ	Density	[kg/m ³]
...		

Chapter 1

Preview

1.1. 协方差矩阵

1.2. 矩阵微分

向量偏导算子为

$$D_x := \frac{\partial}{\partial x^T} = \left[\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \dots, \frac{\partial}{\partial x_m} \right] \quad (1.1)$$

Chapter 2

Kalman Filter

2.1. 基本问题

问题由两部分组成，第一式子是由物理算法构建的估计模型，第二条式子是测量仪器观测到的数据

$$\begin{cases} \hat{x}_k = Ax_{k-1} + Bu_{k-1} + \omega_{k-1} \\ z_k = Hx_k + \nu_k \end{cases} \quad (2.1)$$

其中 z_k 是仪器测量值， x_k 是估计的实际值， \hat{x}_k 是后验估计值， x_{k-1} 是当前已经测得的数据。方程中还有噪声模型

$$\begin{aligned} \omega_{k-1} &\sim N(0, Q) \\ \nu_k &\sim N(0, R) \end{aligned} \quad (2.2)$$

我们当然希望模型描述的是完全正确的，在完全理想情况下，问题的表达如下：

$$\begin{cases} x_k^- = Ax_{k-1} + Bu_{k-1} \\ z_k = Hx_k^m \end{cases} \quad (2.3)$$

如果完全没有噪声，那么我们完全可以从仪器测量值 z_k 去反推出理想实际值 x_k^m 并认为它就是实际值 x_k ，或者相信系统的数学模型算出来的先验值 x_k^- 认为它是 x_k ，但是由于实际的物理系统不论是测量系统还是算法的物理系统最终都会拟合进噪声，也就是 (2.1) 的形式。

(2.1) 前者事实上叠加了过程噪声，后者则叠加了测量噪声，于是我们已经得到了两把不准确的尺子。我们的目的就是根据两把不准确的尺子按照数据融合的方式得到一把更加精确的尺子，也就是后验估计。

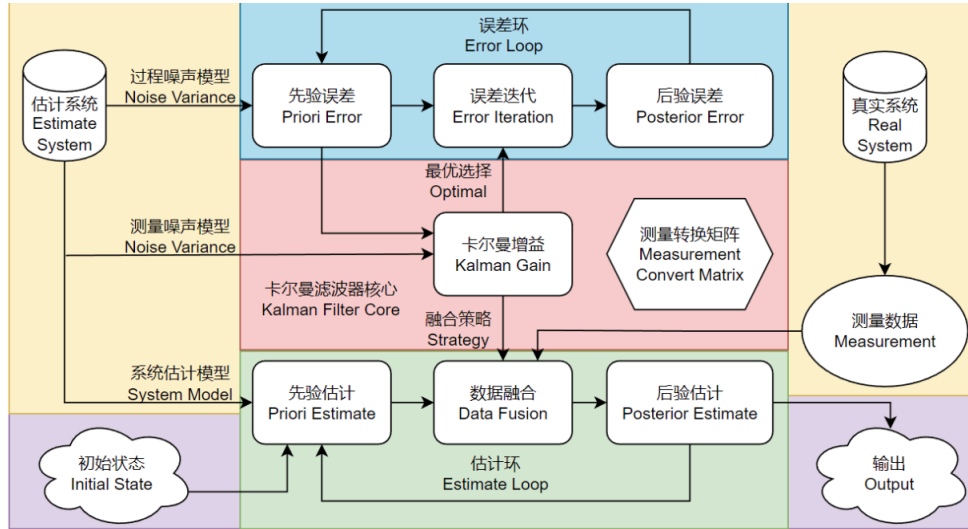


图 2.1: Kalman Filter

2.2. 数据融合

假设 \hat{x}_k^- 是模拟系统先验估计值, \hat{x}_k^m 为测量系统测量到的 z_k 利用 $z_k = H\hat{x}_k^m$ 反演得到理想实际值, 也就是说这是测量值, 假设系数矩阵 G 使得

$$\hat{x}_k = (1 - G)\hat{x}_k^- + G\hat{x}_k^m \quad (2.4)$$

如果 $G = 0$, 则后验估计偏向先验估计, $G = 1$ 则偏向测量值。令 $G = K_k H$, 又有 $z_k = H\hat{x}_k^m$, 整理得到

$$\begin{aligned} \hat{x}_k &= \hat{x}_k^- + G(\hat{x}_k^m - \hat{x}_k^-) = \hat{x}_k^- + K_k H(\hat{x}_k^m - \hat{x}_k^-) \\ &= \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \end{aligned} \quad (2.5)$$

我们的目标是估计卡尔曼系数使得后验估计的误差 e_k 最小。

2.3. 后验估计

后验估计误差为 $e_k = x_k - \hat{x}_k$, 其中 x_k 是实际值, 代入 (2.5)

$$\begin{aligned} e_k &= x_k - [\hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)] \\ &= x_k - [\hat{x}_k^- + K_k(H\hat{x}_k + \nu_k - H\hat{x}_k^-)] \\ &= [I - K_k H]e_k^- + K_k \nu_k \end{aligned} \quad (2.6)$$

其中 e_k^- 是先验误差, ν_k 是系统的测量误差, 先验误差的定义如下

$$\begin{aligned} e_k^- &= x_k - \hat{x}_k^- = (Ax_{k-1} + Bu_{k-1} + \omega_{k-1}) - (A\hat{x}_{k-1} + Bu_{k-1}) \\ &= A(x_{k-1} - \hat{x}_{k-1}) + \omega_{k-1} \\ &= Ae_{k-1} + \omega_{k-1} \end{aligned} \quad (2.7)$$

在此基础上考虑到误差值中仅仅耦合了过程噪声和测量噪声, 并且这两个噪声独立且服从各自的正态分布, 我们可以说滤波器的先验误差和后验误差一定也符合正态分布, 这个结论可以表

达为下式

$$\begin{aligned} e_k &\sim N(0, M_k), \quad M_k = E[e_k e_k^T] \\ e_k^- &\sim N(0, M_k^-), \quad M_k^- = E[e_k^- (e_k^-)^T] \end{aligned} \quad (2.8)$$

我们的目标是使的后验估计误差的方差最小，后验误差的协方差矩阵为

$$\begin{aligned} E[ee^T] &= E[(I - K_k H)e_k^- + K_k \nu_k] [(I - K_k H)e_k^- + K_k \nu_k]^T \\ &= E[(I - K_k H)e_k^- + K_k \nu_k] [(e_k^-)^T (I - K_k H)^T + \nu_k^T K_k^T] \\ &= E[(I - K_k H)e_k^- \cdot (e_k^-)^T (I - K_k H)^T + K_k \nu_k \cdot (e_k^-)^T (I - K_k H)^T \\ &\quad + (I - K_k H)e_k^- \cdot \nu_k^T K_k^T + K_k \nu_k \cdot \nu_k^T K_k^T] \\ &= E[(I - K_k H)e_k^- \cdot (e_k^-)^T (I - K_k H)^T] + E[K_k \nu_k \cdot (e_k^-)^T (I - K_k H)^T] \\ &\quad + E[(I - K_k H)e_k^- \cdot \nu_k^T K_k^T] + E[K_k \nu_k \cdot \nu_k^T K_k^T] \end{aligned} \quad (2.9)$$

由于测量误差 ν_k 和先验误差 e_k^- 服从零均值的相互独立正态分布，因此中间两项为零，因此上式最终写成

$$\begin{aligned} M_k &= E[ee^T] = E[(I - K_k H)e_k^- \cdot (e_k^-)^T (I - K_k H)^T] + E[K_k \nu_k \cdot \nu_k^T K_k^T] \\ &= (I - K_k H)E[e_k^- \cdot (e_k^-)^T] (I - K_k H)^T + K_k E[\nu_k \cdot \nu_k^T] K_k^T \\ &= (I - K_k H)M_k^- (I - K_k H)^T + K_k R K_k^T \end{aligned} \quad (2.10)$$

其中 R 是在本篇一开始就定义的 ν_k 的协方差矩阵。根据协方差矩阵的性质，对角线就是后验估计误差的方差，因此我们取 M_k 的迹

$$\begin{aligned} tr(M_k) &= tr[(I - K_k H)M_k^- (I - K_k H)^T] + tr[K_k R K_k^T] \\ &= tr[M_k^-] - tr[K_k H M_k^-] - tr[M_k^- H^T K_k^T] \\ &\quad + tr[K_k H M_k^- H^T K_k^T] + tr[K_k R K_k^T] \\ &== tr[M_k^-] - tr[K_k H M_k^-] - tr[K_k H M_k^-] \\ &\quad + tr[K_k H M_k^- H^T K_k^T] + tr[K_k R K_k^T] \end{aligned} \quad (2.11)$$

1

目标是选择 K_k 使的 $tr(M_k)$ 最小化，由于方差是凸函数，所以极小值必然存在，根据极值存在的必要条件

$$\frac{\partial tr(M_k)}{\partial K_k} = 0 \Rightarrow M_k^- H^T = K_k (H M_k^- H^T + R) \quad (2.12)$$

因此我们可以给出卡尔曼增益系数 K_k 在最优估计下的表达式

$$K_k = \frac{M_k^- H^T}{H M_k^- H^T + R} \quad (2.13)$$

将卡尔曼增益系数代回到数据融合方程里，就得到后验的表达

$$\hat{x}_k = \hat{x}_k^- + \frac{M_k^- H^T}{H M_k^- H^T + R} (z_k - H \hat{x}_k^-) \quad (2.14)$$

¹这里用到对称矩阵迹的性质： $tr(A^T) = tr(A)$ ， $tr[(AB)^T] = tr[B^T A^T]$ ， $tr(A + B) = tr(A) + tr(B)$

直观理解是：当测量误差 R 很大，估计趋于相信先验的估计，当 R 很小，则趋于相信仪器的测量。 R 源于仪器的固定属性，我们需要确定的是 M_k^-

$$\begin{aligned}
 M_k^- &= E[(e_k^-)(e_k^-)^T] = E[(Ae_{k-1} + \omega_{k-1})(Ae_{k-1} + \omega_{k-1})^T] \\
 &= E[Ae_{k-1}e_{k-1}^T A^T] + E[Ae_{k-1}\omega_{k-1}^T] + E[\omega_{k-1}A^T e_{k-1}^T] \\
 &\quad + E[\omega_{k-1}\omega_{k-1}^T] \\
 &= AE[e_{k-1}e_{k-1}^T]A^T + E[\omega_{k-1}\omega_{k-1}^T] \\
 &= AM_{k-1}A^T + Q
 \end{aligned} \tag{2.15}$$

这里 $Q = E[\omega_{k-1}\omega_{k-1}^T]$ 过程误差的协方差矩阵， M_{k-1} 为前面测得的后验误差协方差矩阵，这些都是前次计算得出来的数据，从这里也可以看出其实卡尔曼滤波是一种递归算法。

2.4. 算法

由于数学推导很乱，我们在这里做一个总结

算法：(Kalman Filter)

1. 预测：计算先验估计值 x_k^- 和先验误差的协方差矩阵 M_k^- ；
2. 校正：计算卡尔曼增益系数 K_k 和后验估计 \hat{x}_k ；
3. 额外更新：更新后验估计的误差协方差矩阵 M_k ，用来计算下一次的 M_{k+1}^- ；

$k-1$ 次计算需要准备好的数据为 M_{k-1}, Q, R 。

Chapter 3

Generalize Kalman Filter

扩展卡尔曼滤波器主要使用泰勒展开 (Taylor) 去处理系统将其线性化，而非更加彻底的使用微分同胚变换直接更改状态空间拓扑结构的反馈线性化 (Feedback Linearization) 去做这件事情。众所周知，泰勒展开是将非线性函数分解为若干个幂级数的叠加 (这里不讨论余项问题)，假设函数在 t_0 处展开。

Chapter 4

Kalman Filter 的实例

References

- [1] I. Surname, I. Surname, and I. Surname. “The Title of the Article”. In: *The Title of the Journal* 1.2 (2000), pp. 123–456.

Chapter A

Source Code Example

Adding source code to your report/thesis is supported with the package listings. An example can be found below. Files can be added using `\lstinputlisting[language=<language>]{<filename>}`.

```
1 """
2 ISA Calculator: import the function, specify the height and it will return a
3 list in the following format: [Temperature,Density,Pressure,Speed of Sound].
4 Note that there is no check to see if the maximum altitude is reached.
5 """
6
7 import math
8 g0 = 9.80665
9 R = 287.0
10 layer1 = [0, 288.15, 101325.0]
11 alt = [0,11000,20000,32000,47000,51000,71000,86000]
12 a = [-.0065,0,.0010,.0028,0,-.0028,-.0020]
13
14 def atmosphere(h):
15     for i in range(0,len(alt)-1):
16         if h >= alt[i]:
17             layer0 = layer1[:]
18             layer1[0] = min(h,alt[i+1])
19             if a[i] != 0:
20                 layer1[1] = layer0[1] + a[i]*(layer1[0]-layer0[0])
21                 layer1[2] = layer0[2] * (layer1[1]/layer0[1])**(-g0/(a[i]*R))
22             else:
23                 layer1[2] = layer0[2]*math.exp((-g0/(R*layer1[1]))*(layer1[0]-layer0[0]))
24     return [layer1[1],layer1[2]/(R*layer1[1]),layer1[2],math.sqrt(1.4*R*layer1[1])]
```

Chapter B

Task Division Example

If a task division is required, a simple template can be found below for convenience. Feel free to use, adapt or completely remove.

表 B.1: Distribution of the workload

Task	Student Name(s)
Summary	
Chapter 1 Introduction	
Chapter 2	
Chapter 3	
Chapter *	
Chapter * Conclusion	
Editors	
CAD and Figures	
Document Design and Layout	