



普通高校“十三五”规划教材

# Kalman滤波基础 **及** MATLAB仿真

王可东 编著



北京航空航天大学出版社  
BEIHANG UNIVERSITY PRESS

## 内 容 简 介

本书以随机过程为基础,从最优估计基本概念入手,系统讲解了 Kalman 滤波理论、应用方法和性能分析等,并通过 MATLAB 编程示范,促进对滤波算法的掌握和应用。

全书共 9 章,内容包括:最优估计和 Kalman 滤波的历史与发展趋势、向量矩阵运算基础、随机过程基础、线性系统基础、典型最优估计算法、Kalman 滤波算法、Kalman 滤波应用方法、Kalman 滤波性能分析、非线性滤波基础、Kalman 滤波算法在卫星/惯性组合导航中的应用。书中各章配备有相应的练习题,主要算法和例题均提供了 MATLAB 参考程序。

本书可作为高等院校控制类、仪器类和应用数学类专业最优估计和信息融合方法的教学用书,也可供其他相关专业的师生和科技人员参考。

### 图书在版编目(CIP)数据

Kalman 滤波基础及 MATLAB 仿真 / 王可东编著. -- 北京:北京航空航天大学出版社,2019.1

ISBN 978-7-5124-2843-0

I. ①K… II. ①王… III. ①卡尔曼滤波器—高等学校—教材②Matlab 软件—高等学校—教材 IV. ①TN713  
②TP317

中国版本图书馆 CIP 数据核字(2018)第 230461 号

版权所有,侵权必究。

### Kalman 滤波基础及 MATLAB 仿真

王可东 编著

责任编辑 刘晓明

\*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(邮编 100191) <http://www.buaapress.com>

发行部电话:(010)82317024 传真:(010)82328028

读者信箱:goodtextbook@126.com 邮购电话:(010)82316836

艺堂印刷(天津)有限公司印装 各地书店经销

\*

开本:787×1092 1/16 印张:18.5 字数:474 千字

2019 年 1 月第 1 版 2019 年 1 月第 1 次印刷 印数:3 000 册

ISBN 978-7-5124-2843-0 定价:58.00 元



# 前 言

由于现实世界中随机噪声无处不在,使得基于确定性信号处理方法获得的结果缺乏理论基础,因而基于统计量的信号处理方法获得了越来越多的研究和应用。从统计角度出发,对混有随机噪声的随机过程进行处理的过程,通常称为估计;而从某种意义上,使得估计结果最接近被估计量的真值,则称为最优估计。显然,这里的“最优”是有特定意义的,即从不同的角度出发,都可以称为“最优”,但各个最优结果之间可能会有一定的差异。Kalman 滤波就属于一种最优估计算法,即在线性、无偏和最小方差意义下,其估计结果是最优的。由于 Kalman 滤波算法适合于多输入、多输出线性系统的状态估计,自 1960 年提出后,迅速成功地应用于 Apollo 登月飞行器的组合导航系统中,随后在导航、天文、模式识别、金融、气象和统计等涉及随机信号处理的领域得到了广泛应用。

因此,在很多课程中都涉及到 Kalman 滤波的内容,例如在图像处理中;而在组合导航领域中,由于 Kalman 滤波内容极为重要,研究人员将 Kalman 滤波内容作为主要内容,撰写了很多教材,为读者提供了扎实掌握 Kalman 滤波原理和应用方法的基础。但是,目前的相关教材存在比较突出的问题,包括:① 理论性过强,使得初学者容易陷入复杂推导的困境中,不利于对基本原理的快速掌握和对重要内容的循序渐进的学习;② 部分内容陈旧,未将 Kalman 滤波相关最新成果包括在内,学习的时效性不强;③ 缺乏编程指导,算法实现过程是理解和掌握算法的关键,但是,目前的大部分教材均未提供相关例程,不利于初学者理解和掌握相关算法,以及后续的应用。

针对这些问题,笔者以初学者为对象,编写了这本从入门到精通的 Kalman 滤波算法理论和应用方法的教材,在加强基本原理讲解的同时,配备了适当的例题,解释和示范有关原理和方法;针对大部分滤波算法,基于 MATLAB 语言,配备了示范编程代码,为初学者提供可以执行的算法程序,以加深对有关算法的理解,后续也可以基于这些例程进行更复杂的应用编程。

本教材主要包括如下几个部分:

① 绪论和相关基础部分。这部分由第 1~3 章组成,其中:第 1 章主要介绍估计的定义和 Kalman 滤波的发展历史,建立最优估计的基本概念;第 2 章主要介绍本教材用到的向量、矩阵和随机过程等相关的数学知识,为后续的算法学习奠定数学基础;第 3 章主要介绍线性系统的相关内容,主要包括建模方法、离散化方法、可观性、可控性和误差传播方程等,是 Kalman 滤波建模的基础。

② Kalman 滤波算法部分。这部分由第 4~8 章组成,其中:第 4 章主要介绍

# 目 录

第 1 章 绪 论	1
1.1 最优估计	1
1.1.1 估计的定义	1
1.1.2 最优估计的定义	3
1.1.3 最优估计的一般架构	4
1.2 Kalman 滤波的发展历史	5
1.2.1 最小二乘算法	5
1.2.2 Wiener 滤波算法	5
1.2.3 Kalman 滤波算法	6
1.2.4 非线性滤波算法	7
1.3 本教材所包括的内容	9
1.4 MATLAB 软件简介	9
1.4.1 软件界面简介	9
1.4.2 常用的操作命令	10
1.4.3 m 文件	11
1.4.4 绘 图	12
习 题	15
第 2 章 数学基础	16
2.1 向 量	16
2.1.1 表示法	16
2.1.2 基本运算方法	16
2.2 矩 阵	18
2.2.1 表示法	18
2.2.2 基本运算方法	18
2.3 向量-矩阵运算	21
2.3.1 二次型	21
2.3.2 定	21
2.3.3 范 数	22
2.3.4 梯度运算	23
2.4 最小二乘算法	24
2.5 概 率	26
2.6 随机变量	28
2.6.1 概率分布函数	28
2.6.2 概率分布密度函数	28

3.5.5 指数型自相关函数的随机过程 .....	92
习 题 .....	92
<b>第 4 章 最优估计算法 .....</b>	<b>95</b>
4.1 最小二乘估计算法 .....	95
4.1.1 LS 估计算法 .....	95
4.1.2 WLS 估计算法 .....	97
4.1.3 RLS 估计算法 .....	98
4.2 最小方差估计 .....	101
4.2.1 最小方差估计算法 .....	101
4.2.2 估计偏差特性 .....	102
4.2.3 Gauss 分布时的最小方差估计 .....	102
4.3 线性最小方差估计 .....	109
4.3.1 估计算法 .....	110
4.3.2 线性特性 .....	111
4.3.3 正交投影定理 .....	111
4.4 极大验后估计 .....	113
4.5 极大似然估计 .....	113
4.6 Wiener 滤波 .....	115
习 题 .....	117
<b>第 5 章 Kalman 滤波算法 .....</b>	<b>120</b>
5.1 递推滤波器 .....	121
5.2 离散 Kalman 滤波算法 .....	121
5.2.1 系统建模 .....	121
5.2.2 算法推导 .....	122
5.2.3 算法总结 .....	123
5.2.4 正交投影法推导 .....	126
5.3 离散 Kalman 滤波使用方法 .....	132
5.3.1 初值的确定 .....	132
5.3.2 $P_k(+)$ 计算公式 .....	133
5.3.3 离散化 .....	133
5.3.4 系统模型中有确定性项 .....	136
5.3.5 状态噪声与量测噪声相关 .....	136
5.4 连续 Kalman 滤波算法 .....	139
5.4.1 系统模型 .....	139
5.4.2 算法推导 .....	140
5.4.3 Riccati 方程求解 .....	142
习 题 .....	149
<b>第 6 章 Kalman 滤波应用技术 .....</b>	<b>152</b>
6.1 有色噪声 .....	152

8.2	UKF 算法	235
8.2.1	Unscented 变换	235
8.2.2	SUT 变换	236
8.2.3	滤波算法	237
8.3	PF 算法	242
8.3.1	递推 Bayes 估计	242
8.3.2	Monte Carlo 模拟	243
8.3.3	重要性采样	243
8.3.4	序贯重要性采样	244
8.3.5	重采样	246
8.4	改进的 PF 算法	247
8.4.1	通用算法	247
8.4.2	优化重采样算法	248
8.4.3	改进的粒子滤波算法	248
	习 题	253
<b>第 9 章</b>	<b>GPS/INS 组合导航</b>	<b>256</b>
9.1	INS 导航解算	256
9.1.1	陀螺仪简介	256
9.1.2	加速度计简介	257
9.1.3	常用坐标系	258
9.1.4	导航解算	258
9.2	GPS 定位测速原理	266
9.2.1	定位原理	266
9.2.2	测速原理	266
9.2.3	精度因子	267
9.3	GPS/INS 组合导航方法	269
9.3.1	组合模式	269
9.3.2	系统建模	270
9.3.3	量测建模	274
9.3.4	滤波算法	275
	习 题	280
	参考文献	282

本章主要介绍最优估计的基本概念、Kalman 滤波的发展历史和当前还存在的主要问题，给出最优估计的基本框架。

## 1.1 最优估计

### 1.1.1 估计的定义

先看如下问题：

**【例 1-1】** 如图 1-1 所示。设一静止的车辆以  $2.5 \text{ m/s}^2$  的加速度沿一公路直线行驶，问 10 s 后该车辆的速度和位移分别是多少？



图 1-1 车辆直线行驶

**【解】** 由于车辆进行匀加速直线运动，由运动学可知，在已知车辆加速度  $a$  的情况下，车辆的速度  $v(t)$  和位移  $s(t)$  随运动时间  $t$  的关系分别为

$$\left. \begin{aligned} v(t) &= v_0 + a(t - t_0) \\ s(t) &= s_0 + v_0(t - t_0) + \frac{1}{2}a(t - t_0)^2 \end{aligned} \right\} \quad (1.1)$$

如果设初始时刻为 0，位移起始点也为 0，由于车辆初始时静止，因此，上式可简化为

$$\left. \begin{aligned} v(t) &= at \\ s(t) &= \frac{1}{2}at^2 \end{aligned} \right\} \quad (1.2)$$

当  $t=10 \text{ s}$  和  $a=2.5 \text{ m/s}^2$  时，可得速度和位移分别为  $25 \text{ m/s}$  和  $125 \text{ m}$ ，即 10 s 时车速达到  $90 \text{ km/h}$ ，位移为  $125 \text{ m}$ 。

在例 1-1 中，只要给定时间点，即可准确地推算出该时刻车辆的速度和位移，这是由式(1.1)所反映的客观规律所决定的，而且推算结果不存在任何误差。因此，由事物的客观规律可以对我们的感兴趣的某些量进行准确推算。

但是，在例 1-1 中，进行速度和位移的准确推算是有条件的，即车辆的准确加速度是知道的，但实际中加速度只能由传感器测量获得，而任何传感器都存在确定性和随机测量误差，那么即使给定运行时间，也不能获得车辆的准确运行速度和位移。设加速度的测量误差为  $\Delta a$ ，由式(1.2)可知，推算的速度和位移误差将随时间  $t$  持续发散。因此，基于这种积分式工作原

理的推算误差是随工作时间发散的。

为了解决上述误差发散问题,可以采用直接测量物理量的方法,仍然以图 1-1 所示的问题为例,提出例 1-2。

**【例 1-2】** 如图 1-1 所示,设车辆从  $O$  处由静止状态启动,行驶 10 s 后,车载北斗卫星导航接收机(Beidou navigation satellite system, BDS)进行了一次速度和位置测量,其中该接收机的测速误差为 0.1 m/s,定位误差为 5 m。问如何由接收机的输出得到车辆的速度和位移?

**【解】** 设接收机的测速和定位误差都是随机误差,且都是零期望正态分布。如果车辆的行驶状态和例 1-1 相同,那么,车辆 10 s 时的速度为  $25 \text{ m/s} \pm 0.3 \text{ m/s}$  的概率为 99.7%。

在计算位移时,需要确定坐标系,这里是一维的,设  $O$  为起始零点,那么,将接收机输出的坐标与起始点做差即可得到车辆的位移,车辆行驶 10 s 时的位移为  $125 \text{ m} \pm 15 \text{ m}$  的概率为 99.7%。

在例 1-2 中,车辆的速度和位移是由传感器直接测量的,如果没有测量误差,那么,可以得到车辆的准确速度和位移。但是,实际上传感器也存在误差,只是这里不是基于积分推算,测量误差随工作时间发散。

实际上,例 1-1 和例 1-2 分别代表两种获取物体状态的方法,即积分推算法和直接测量法,如果没有测量误差,那么这两种方法获得的状态量都是准确的。但是,由于实际的测量都是有误差的,导致这两种方法获得的结果都不准确,因而,应用中只能给出物体状态在一定误差范围内的大概结果,即估计值。

误差分为确定性的和随机的,从理论上讲,确定性的误差是可以完全补偿的,但随机误差是无法完全补偿的。随机误差只能从统计意义上认识,因此,在对包含随机误差的状态进行确定时,也只能得到其统计值,而统计值是基于大样本试验获得的,具体到某一个样本结果,可能与统计值相差很大。所以,在对随机状态进行确定时,不能追求对其某个样本值的确定,而是对其统计值的确定,对随机状态统计值的确定就是“估计”。

根据上面的描述,可以定义“估计”如下:基于测量结果,按照状态与其测量值之间的内在关系,确定状态统计量的过程。

因此,可以说,如果没有测量误差,就不存在“估计”问题。但是,实际上任何测量都存在误差,在确定状态时其实都是“估计”,即只能得到状态的统计值,而不可能得到其所谓的“真值”,实际上也没有必要得到“真值”。

按照状态估计的时刻与利用测量值的时刻之间的先后关系,估计分为如下三类:

#### (1) 预 测

利用从初始时刻到当前时刻的所有测量结果,对未来某一时刻的状态进行估计的过程。

#### (2) 滤 波

利用从初始时刻到当前时刻的所有测量结果,对当前时刻的状态进行估计的过程。

#### (3) 平 滑

利用从初始时刻到当前时刻的所有测量结果,对过往某一时刻的状态进行估计的过程。

如图 1-2 所示为上述三类估计的原理示意图。显然,从常识角度看,由于预测过程中利用的是旧的测量结果对还未发生的状态进行估计,因而预测的精度在三者中是最差的;相反,



由于平滑过程中利用的是最新的测量结果对过往的状态进行估计,属于事后处理,因而平滑的精度在三者中是最高的。滤波的精度介于预测的和平滑的之间。

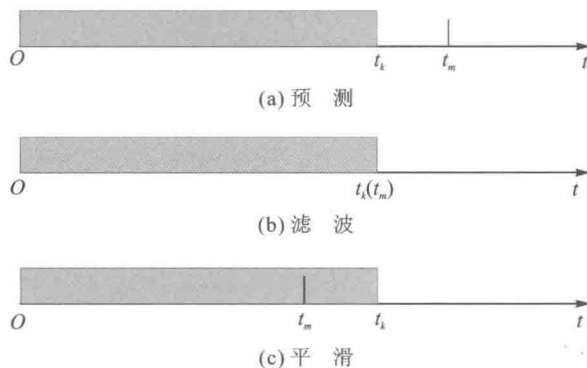


图 1-2 三类估计的示意图

### 1.1.2 最优估计的定义

基于上面的描述可知,由于测量误差的存在,使得积分推算法和直接测量法得到的状态量都存在误差,都只能得到状态的统计值,再考虑到这两种方法的其他特性,比如前者状态输出率较后者高,因此,在实际应用中,往往是将二者结合起来,共同应用,以获得状态的估计值。显然,结合的方法将决定状态估计的精度。另外,在通过测量获得状态的方法中,有时使用多个传感器同时测量物体的某一个状态量,例如,通过惯性导航系统(Inertial Navigation System, INS)和 BDS 接收机同时测量车辆的速度,每个传感器都有测量误差,且误差量级可能是不一样的,此时,采用不同的方法处理测量值,得到该状态量的精度可能也有差异。因此,如何基于多个测量结果,得到状态的某种最好估计,就非常值得研究。

综上,可以定义“最优估计”为:基于对状态的多个测量结果,按照某种最优准则,实现对状态的估计。

下面举一个例子来说明如何基于多个测量结果进行状态的最优估计。

**【例 1-3】** 设对某一常量  $x$  进行两次独立无偏测量,试基于这两次测量结果给出对常量  $x$  的线性、无偏和最小方差的估计结果。

**【解】** 由题意,设两次测量值  $z_1$  和  $z_2$  分别为

$$\left. \begin{aligned} z_1 &= x + v_1 \\ z_2 &= x + v_2 \end{aligned} \right\} \quad (1.3)$$

式中,  $v_1$  和  $v_2$  分别为两次测量的误差。由于两次测量都是无偏的,即  $E(z_1) = E(z_2) = E(x)$ , 又  $x$  为常量,因此,  $E(x) = x$ , 可得  $E(v_1) = E(v_2) = 0$ 。同时,设

$$\left. \begin{aligned} E(v_1^2) &= \sigma_1^2 \\ E(v_2^2) &= \sigma_2^2 \end{aligned} \right\} \quad (1.4)$$

又因为两次测量是独立的,即两次测量误差之间是不相关的,故有  $E(v_1 v_2) = 0$ 。设  $x$  的估计值为  $\hat{x}$ , 按照线性假设,有

$$\hat{x} = k_1 z_1 + k_2 z_2 \quad (1.5)$$

式中,  $k_1$  和  $k_2$  为待定的线性加权系数。两个未知数,需要两个独立方程来确定。估计偏差为

$\tilde{x} = \hat{x} - x$ , 基于无偏估计要求, 有

$$\begin{aligned} E[\tilde{x}] &= E[k_1(x + v_1) + k_2(x + v_2) - x] \\ &= (k_1 + k_2 - 1)E(x) \\ &= (k_1 + k_2 - 1)x \\ &= 0 \end{aligned} \quad (1.6)$$

可得

$$k_1 + k_2 - 1 = 0 \quad (1.7)$$

再考虑到估计偏差的方差最小, 有

$$E[\tilde{x}^2] = k_1^2 \sigma_1^2 + (1 - k_1)^2 \sigma_2^2 \quad (1.8)$$

对上式求关于  $k_1$  的导数, 并令其为 0, 得

$$k_1 = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} \quad (1.9)$$

所以, 最小均方估计误差为

$$E(\tilde{x}^2)_{\min} = \left( \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} \right)^{-1} \quad (1.10)$$

$$\hat{x} = \left( \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} \right) z_1 + \left( \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} \right) z_2 \quad (1.11)$$

由式(1.11)可知, 当  $z_1$  的测量误差较大时,  $\sigma_1^2$  较大, 那么  $z_2$  在最后的估计值中的权重就增大; 反之亦然, 即式(1.11)的估计分配给测量精度高的值更大的权重, 显然, 这是符合常理的。在获得式(1.11)的估计结果时, 利用了两次测量结果, 并设定了线性、无偏和估计偏差方差最小等三个条件, 因此, 式(1.11)在这个意义上是对常量  $x$  的最优估计。可见, 所谓的“最优”是在特定条件下的某种最优, 若这些条件变化了, 则最优的结果可能也会相应变化, 具体在后面章节中将进行详细讲述。

### 1.1.3 最优估计的一般架构

由例 1-1~例 1-3 可以归纳如下最优估计的一般过程:

① 基于事物自身的某种规律, 建立状态随时间的变化关系, 称为“系统建模”, 如位移、速度和加速度之间的积分推算关系。基于建立的系统模型, 可以推算出状态的估计值, 即预测; 但是, 由于测量误差的存在, 通常只是基于系统模型的预测误差是随时间发散的, 即推算的时间越长, 误差越大。因此, 只是基于系统模型进行预测是不能长时间工作的。

② 建立测量值与状态之间的变化关系, 称为“量测建模”, 如 BDS 接收机的位置和速度测量。为了避免系统模型中的积分推算所导致的误差发散, 在建立的量测模型中, 应尽量避免测量值与状态之间有微积分的关系, 即最好是测量量与状态之间是直接对应的, 如 BDS 接收机输出的位置就是对物体位置的直接测量, 二者之间不是微积分的关系。相反, 如果测量量与状态之间存在微积分关系, 将导致与系统模型相似的后果, 即基于测量值对状态进行估计时也会导致误差发散。

③ 对误差进行建模, 其中包括系统误差和测量误差。在例 1-3 中, 测量误差的无偏、独立和方差都是进行状态精确估计的必要条件, 而这些条件的获取过程就是误差建模。因此, 系统误差和测量误差的建模是系统和量测建模不可缺少的一部分, 误差的精确建模是取得精确

估计的必要基础。

④ 基于某种最优准则构建最优估计算法。在例 1-3 中,就是基于线性、无偏和估计偏差方差最小的最优准则,建立了式(1.11)对常量  $x$  的最优估计算法。在一般的状态估计中,由于建立的是关于时间的递推算法,因此,还需要进行状态的初始化。

图 1-3 显示了上述最优估计的一般架构和流程。

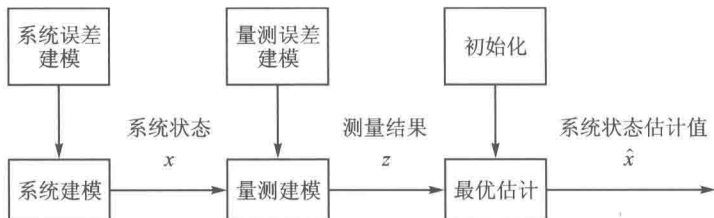


图 1-3 最优估计的一般架构

## 1.2 Kalman 滤波的发展历史

Kalman 滤波算法是一种线性、无偏和估计偏差方差最小的最优估计算法,在 Kalman 滤波算法之前,与之有密切渊源的是最小二乘算法(Least Squares, LS)和 Wiener 滤波算法,而目前在发展中的是非线性滤波算法。下面分别简单总结一下这些最优估计算法的发展过程。

### 1.2.1 最小二乘算法

1801 年, Karl Gauss 在处理行星轨道观测数据时,提出了 LS 算法,其最优准则是估计值与观测值之间的偏差平方和最小,成功地预测到了谷神星(Ceres)的位置。目前,LS 算法在几乎所有涉及数据处理的领域都得到了广泛应用。LS 算法的最大特点包括:

① 批处理算法。基于到当前时刻为止所有的测量结果,对当前时刻的状态进行估计,因而通常适用于事后处理。

② 应用简单。只需要量测模型即可完成估计,且无需对测量误差进行建模,实际上 LS 算法在有测量误差时均可用,应用条件非常宽泛。

③ 稳定可靠。其收敛性好,是到目前为止应用最广泛的估计算法。

上述特点有时也成为阻碍其应用的缺点,例如,对实时应用来说,批处理是不可接受的。因此,针对这些可能存在的问题,研究者也提出了相应的改进算法,包括递推 LS(Recursive Least Squares, RLS)、加权 LS(Weighted Least Squares, WLS)和加权递推 LS(Weighted Recursive Least Squares, WRLS)等,其中:RLS 主要是解决批处理所带来的实时性差的问题;WLS 主要是解决不同测量精度数据一起处理时的估计精度提升问题,因为在忽略误差模型时虽然方便应用,但不利于估计精度提升;而 WRLS 则可以实现高精度实时估计。

### 1.2.2 Wiener 滤波算法

虽然 WLS 算法较 LS 算法的精度有一定的提高,但是,通常都是假设测量误差为高斯白噪声,局限性较大。为了解决在任意噪声情况下的状态估计问题,1942 年 Robert Wiener 提出

了一种有随机噪声干扰下的信号最优估计算法,即维纳滤波算法(Wiener Filter, WF),同时期,苏联的 Andrey Kolmogorov 也独立提出了相似的算法,因此,有时也称为 Wiener-Kolmogorov 滤波算法。WF 算法是一种线性、无偏和估计偏差方差最小的最优估计算法,是第一个明确从随机过程角度提出的状态最优估计算法,这是其区别于 LS 算法最明显之处,标志着状态估计正式从确定性过程处理转为随机过程处理,也使得在 LS 算法中不需要进行的误差建模,变为算法构建必不可少的一部分,实际上误差建模是 WF 算法这类随机过程处理算法精度提升的基础。

### 1.2.3 Kalman 滤波算法

但是,WF 算法是在频域设计的,过程复杂,一般只适用于一维状态估计,导致其应用范围很小。20 世纪 50 年代,电子计算机的发明和快速发展,使得信号处理从一维向多维实时处理成为可能,也需要有取代 WF 算法的适用于多维状态最优估计的算法出现。

1960 年,Rudolf Kalman 和 Richard Bucy 提出了一种适用于多维状态的线性、无偏和估计偏差方差最小的估计算法,即 Kalman 滤波算法(Kalman Filter, KF)。从其最优准则上看, KF 算法与 WF 算法是一样的,都是线性、无偏和估计偏差方差最小,但是,二者有明显的差异,主要包括:

① KF 算法既可用于一维状态估计,也可用于多维状态估计;而 WF 算法通常很难用于多维状态估计。

② KF 算法是基于现代控制理论提出的,是基于状态空间方程设计的,是一种时域滤波算法;而 WF 算法是基于经典控制理论提出的,是基于传递函数设计的,是一种频域滤波算法。

KF 算法提出后,迅速受到了美国航空航天局(National Aeronautics and Space Administration, NASA)的关注,并成功应用于 Apollo 登月。自此, KF 算法在航空航天领域得到了广泛研究和应用。不过, KF 算法在应用过程中先后遇到了一系列的问题:

第一个问题就是计算量过大。当时的计算能力非常有限,例如阿波罗登月的导航计算机的硬盘只有区区几十 KB,在处理多维矩阵运算时困难是可想而知的。为了提高计算的实时性,研究者提出了序贯处理算法,将多维的量测量分解为标量处理,在提高计算速度的同时,也保证了计算的稳定性。考虑到滤波算法中协方差矩阵更新消耗了大量的计算资源问题,针对二维和三维滤波分别提出了固定增益的  $\alpha-\beta$  滤波算法和  $\alpha-\beta-\gamma$  滤波算法,由于不需要更新协方差矩阵,计算量大幅度降低,有效地提高了实时性;不过,此时的滤波算法不是最优的。为了降低计算量,还提出了状态删减算法和状态解耦等次优算法。随着计算机计算能力的提升,目前这些解决方法是否有必要再使用,特别是那些次优算法,可以根据具体情况具体甄别。

第二个问题是滤波发散。按照预期,滤波结果应该是估计偏差方差最小意义上的最优估计,但是在实际应用中,往往不仅不能得到最优估计结果,反而出现的是估计值与真值偏差越来越大,即滤波结果发散。引起滤波发散的原因主要有两个:一是计算误差;二是模型误差。

由于计算机中处理的都是有限位数值,存在截断误差,同时还存在着算法近似误差(例如正弦函数在计算机中只能按照 Taylor 级数展开,取有限阶),这样就导致计算误差的出现,而且随着滤波周期的迭代,计算误差可能越来越大,最终导致滤波发散。计算误差引起滤波发散主要是通过协方差矩阵传递实现的,即在计算协方差矩阵时,由于计算误差的累积,导致协方差

矩阵失去正定性,即不可逆;而在更新增益矩阵时,涉及协方差矩阵的求逆,从而导致滤波发散。为了解决由于计算误差引起的滤波发散问题,研究者先后提出了平方根滤波算法和 UD 分解算法,其中平方根滤波算法又分为 Potter 算法和 Calson 算法,这些算法都是从协方差矩阵在更新过程中保持其对称性,进而保持其正定性入手的。

进行 KF 滤波算法设计之前,需要进行系统建模和量测建模,其中还包括系统误差和量测误差的建模,当这些建立的模型与真实模型存在偏差时,就可以引起滤波发散。需要注意的是, KF 算法只适用于线性系统,而实际中几乎所有的系统都是非线性的,因此,从严格意义上讲,模型误差是客观存在的,只是严重程度不同。当系统和/或量测模型与真实模型相差很大时,就会导致滤波发散,其表现就是在滤波结果中分配给离当前时刻久远的状态过大的权重,而分配给当前的测量值的权重过小。因此,为了解决由模型误差所导致的滤波发散问题,研究者先后提出了限定记忆法和衰减记忆法,都是提高离当前时刻近的测量值在滤波结果中的权重,显然,这些滤波算法也是次优的。

第三个问题是多传感器测量时的冗余容错。在航空航天这类高风险应用中,往往需要进行冗余配置,而且采用多源测量,例如在卫星上同时采用陀螺仪、星敏感器、太阳敏感器和磁强计等进行姿态确定;在之前的 KF 滤波中,采用的是集中式滤波方案,即将这些测量值都输入到一个 KF 滤波器中。其存在的风险是,如果有一个传感器出现故障而导致测量精度下降,则很难通过滤波器自行监测和排除,最终导致整个滤波性能下降,即集中式滤波的可靠性差,冗余容错能力弱。为了提高滤波算法的可靠性和冗余容错能力,研究者提出了分散式滤波方案,其中最典型的就是 1988 年 Calson 提出的联邦滤波算法,其具有故障监测和自行诊断隔离的功能,在正常情况下,还可以实现全局最优。

## 1.2.4 非线性滤波算法

其实 KF 算法在应用时面临的最大困难是非线性问题。模型非线性给滤波算法带来了两方面的问题:

① 由于 KF 算法只适用于线性系统,因此为了应用 KF 算法,只能将非线性系统通过 Taylor 级数展开进行线性化,这样不可避免地就引入了模型误差,当非线性比较强的时候,模型误差也很大,从而导致滤波发散。在阿波罗登月任务中,针对实际模型的非线性问题,采用的就是线性化方法,即扩展 Kalman 滤波算法(Extended Kalman Filter, EKF)。不过, EKF 算法只是一阶近似,当非线性很强时,仍然会导致滤波发散,因此,之后又提出了二阶近似和高阶近似算法;但是,当展开阶次增高时,计算量将大幅度提升,算法的实时性又成为问题。

② KF 算法中,只是对状态的期望和协方差进行了估计,由随机过程理论可知,当随机过程符合高斯分布时,其所有的统计特性由其期望和协方差确定,因此,在高斯分布情况下,利用 KF 算法进行状态估计,获得状态的期望和协方差,实现了对状态所有统计特性的估计。所以,在 KF 算法中,通常还将高斯分布作为必要条件之一。但是,对非线性系统来说,其不具有高斯分布保持性,即输入是高斯分布,非线性系统的输出不能保持也是高斯分布,因而,对非线性系统再进行高斯分布假设通常是不符合实际情况的。为了解决这个问题,一方面利用 Taylor 级数展开,进行模型线性化;另一方面利用任意分布都可以等效为无穷多个高斯分布的线性加权,即进行高斯和展开近似,并与模型线性化相结合,构建高斯和展开滤波算法。不过,高斯和展开算法非常复杂,计算量大,而且随着迭代的进行,展开的维数越来越高,即出现所谓的

“维数灾难”，因此，这种算法的应用非常有限。

EKF 算法是基于状态进行的线性化近似，受计算量的限制，通常仅限于一阶近似，当模型非线性较强时，将失效。为了进一步提高近似精度，1995 年，Julier 和 Uhlmann 认为对概率分布进行近似要比对非线性函数进行近似容易，提出了基于状态的期望和协方差的近似方法，利用 UT(Unscented Transform)变换，构建了 UKF(Unscented Kalman Filter)算法，在高斯分布时，可以获得三阶近似精度，大大提高了对模型非线性程度的适应性。随后，Ito 和 Zhang 提出了基于数值积分的中心差分滤波算法(Central Difference Filter, CDF)，Norqaard 等提出了基于多项式插值的分散差分滤波算法(Divided Difference Filter, DDF)，这两种算法本质上都是基于多项式拟合构建的非线性滤波算法，因此，又被统称为中心差分 Kalman 滤波算法(Central Difference Kalman Filter, CDKF)。2009 年，Arasaratnam 和 Haykin 针对 UKF 在解决高维系统滤波时精度下降的问题，利用球面径向规则逼近非线性状态验后统计特性，构建了容积 Kalman 滤波算法(Cubature Kalman Filter, CKF)，在处理低维和高维系统非线性滤波时，精度一致性好。实际上，UKF、CDKF 和 CKF 都是通过对状态的统计特性进行有限采样构建的非线性滤波算法，可统称为“确定性采样型”非线性滤波算法。相较于 EKF 算法，确定性采样型算法中不需要计算 EKF 算法中的 Jacobian 矩阵，计算量增加适中，应用方便，因而得到了广泛应用。但是，这类算法均基于高斯分布假设，所以并不是彻底的非线性非高斯算法，在处理强非线性系统滤波时，仍然存在发散的风险。

到目前为止，真正的非线性非高斯滤波算法只有基于大样本进行 Monte Carlo 仿真的粒子滤波算法(Particle Filter, PF)。PF 算法的思想早在 20 世纪 50 年代就提出了，但受计算能力的限制，在很长时间内并未受到关注，到 20 世纪 90 年代，随着计算机计算能力的快速提升，基于 Monte Carlo 仿真的 PF 算法被 Gordon 等提出，并迅速受到广泛关注和深入研究。由于当采样样本数趋于无穷大时，可以以足够高的精度去模拟任意分布的随机过程，而且适用于任意非线性系统，因此，PF 算法是真正意义上的非线性非高斯滤波算法。但是，PF 算法在应用过程中遇到了两大难题：

① 粒子退化。模拟的大样本粒子在经过迭代后，大部分粒子的权重都趋于 0，但这些粒子的迭代却消耗了大量的计算量，这就是粒子退化现象。为此，研究者提出了重采样方法，即对权重大的粒子进行多次采样，以保持有效粒子数，但这样又导致粒子趋于一致，多样性不足，统计代表性下降。因此，研究者又提出了一系列改善粒子多样性的重采样方法，如正则化粒子滤波算法、粒子群粒子滤波算法和混合退火粒子滤波算法等。还有将 EKF 和 UKF 引入到 PF 算法中，利用当前最新的测量值，采用 EKF 或 UKF 设计采样重要性密度函数，使重要性密度函数更接近实际的验后概率密度函数，从而避免了粒子退化，改善了粒子滤波性能。

② 计算量大。PF 算法是基于大样本模拟采样设计实现的，一般要求有效的样本数在 20 000~100 000 之间，当进行多维状态估计时，计算量大是可以想象的。为了降低 PF 算法的计算量，研究者提出了无需重采样的高斯 PF 算法(Gaussian Particle Filter, GPF)，在高斯分布假设下，其计算量有一定降低，但又破坏了非高斯的假设条件；另外，研究者还提出了诸如 Rao-Blackwellization 粒子滤波算法等降低计算量的改进算法，但这些改进算法的计算量降低得都有限。因此，PF 算法的计算量大的问题到目前为止仍然没有很好的解决方法。

## 1.3 本教材所包括的内容

本教材以线性 KF 算法为主要内容,以组合导航为主要应用对象,设计如下教学内容:

① 相关的数学基础和线性系统基础。KF 算法从推导到应用,都需要有一定的数学基础作为保障,其中最相关的就是向量与矩阵运算方法和随机过程理论。为了更便于学习和掌握,本教材对与 KF 算法相关的向量矩阵运算方法和随机过程理论知识进行了总结,如需更全面的学习,建议参考相关线性代数和随机过程的教材。KF 算法是一种线性滤波算法,是基于线性系统提出的,因而,需要有线性系统的相关知识,本教材对系统建模、误差传播方程、离散化等内容进行了讲解,更系统的内容,建议参考现代控制理论或线性系统等相关教材。

② KF 算法基础。这里分为两部分,在第一部分对典型最优估计算法进行了总结和对比分析,为全面掌握最优估计算法打下基础,也是更好地学习 KF 算法的条件;在第二部分将从递推滤波概念入手,推导 KF 算法,并给出 KF 算法的一般应用方法。

③ 改进的 KF 算法。针对 KF 算法在应用过程中会遇到的计算量大和滤波发散等问题,分别提出相应的改进算法,为 KF 算法走向应用奠定基础。

④ 非线性滤波基础。以 EKF 算法、UKF 算法和 PF 算法为代表,对非线性滤波算法进行简单介绍。

⑤ KF 算法在组合导航中的应用示范。以捷联惯性导航系统(Strapdown Inertial Navigation System, SINS)为基础,进行卫星/惯性组合导航滤波算法设计和仿真示范。

本教材的所有算法实现和仿真均基于 MATLAB 编程实现,因此,掌握 MATLAB 编程知识是进行本教材学习的必要基础。

## 1.4 MATLAB 软件简介

本教材基于 MATLAB 软件进行算法编制和仿真计算, MATLAB 软件是美国 Math-Works 公司的商用产品,具有强大的计算、仿真和图形显示能力,编程语言简单,画图功能强大,特别适用于算法仿真。在本教材中,只使用 MATLAB 软件进行算法的编程、算法仿真和图形显示等,因此,这里只对该软件的相关内容进行简单介绍,在后续章节中,当涉及到不一样的功能应用时,在具体的程序中会予以介绍。

### 1.4.1 软件界面简介

不同的版本, MATLAB 软件的界面有一定的差异,但总体相差不大,本教材以 R2014a 版本为例进行介绍。

当软件安装成功后,在开始菜单栏找到“MATLAB R2014a”图标,如图 1-4 所示,即可进入软件操作界面。如图 1-5 所示,其中最上面一行是“主页”、“绘图”和“应用程序”三个选项,在本教材中,通常在“主页”中操作即可。在“主页”的下面是一些常用的操作按钮,包括“新建脚本”、“新建”和“打开”等。再下面一行为当前目录,通常是我们自己编辑的程序和数据等文件所存放的目录,通过点击下拉菜单,可以在不同的目录之间转换,调整所需要的当前目录。



图 1-4 软件启动图标



图 1-5 MATLAB R2014a 软件操作界面

在当前目录栏下面，界面被分为三个活动区域，其中：

① 左边的一整块区域为命令窗口，所有操作命令和部分结果显示都是在这个区域里进行的，在“ $f_x \gg$ ”提示符之后输入命令，回车后，命令就得到了执行；如果在执行命令中出现了错误，错误提示信息也是在该窗口显示的。因此，该窗口是接触最多的区域。

② 右上区域为当前目录下的所有文件列表，其中文件的后缀也会显示，点击某个文件时，如果该文件 MATLAB 能显示，则会弹出一个窗口予以显示。

③ 右下区域为内存中的变量列表，这些变量可以直接在命令窗口中处理，比如对某个变量进行绘图，在进行程序调试时，往往会在该区域查看有关变量的结果。

## 1.4.2 常用的操作命令

在命令窗口中，本教材常用的命令包括：

### (1) help

当对某个函数不是很熟悉的时候，可以在命令行中输入“help XXX”（“XXX”为函数名称）即可得到官方的解释信息和用法等，这是获取帮助的最直接途径。例如，输入“help svd”，即



可得到如下内容:

```
>> help svd
svd - Singular value decomposition
```

This MATLAB function returns a vector of singular values.

```
s = svd(X)
[U,S,V] = svd(X)
[U,S,V] = svd(X,0)
[U,S,V] = svd(X,'econ')
```

[svd 的参考页](#)

名为 svd 的其他函数  
[symbolic/svd](#)

## (2) addpath

MATLAB 中所有的函数都放在安装目录下,在使用时,软件会自动找到相关目录。但是,在使用我们自己编辑的函数时,如果也放到安装目录下,则容易与软件自带程序混淆,特别是当计算机重新安装时,容易丢失文件,因此,不建议将自编程序放在软件目录下。在使用时,一种方法是通过改变当前目录,使得程序所在目录为当前目录;另一种方法是通过在命令行中运行“addpath 程序目录”将软件所在目录加入到目录索引中,这样在使用有关程序时,软件也会到相应的目录中去寻找。如图 1-6 所示为在命令行输入“addpath”之后的效果,此时,就可以使用该目录下的所有程序和数据文件。



```
命令行窗口
>> addpath E:\Teaching\Kalman
fx >>
```

图 1-6 addpath 应用举例

## (3) clc

在命令行输入该命令后,命令窗口内的所有内容都被清除。

## (4) close all

在命令行输入该命令后,MATLAB 打开的所有图形窗口均被关闭。

## (5) clear all

在命令行输入该命令后,MATLAB 导入内存中的所有变量均被释放,即图 1-5 左下区域中的内容均被清空。如果只是想释放某个变量,改用“clear 变量名”即可。

clc、close all 和 clear all 这三个命令在编程中也经常使用。

## 1.4.3 m 文件

在命令行中通常都是单个命令的执行,因此,只适用于较为简单的任务。当任务比较复杂时,则需要编辑 m 文件,形成程序,并进行命名,然后在命令行中输入该程序的名称,即可执行所设计的任务。当然,在执行 m 文件时,需要先把该文件所在的目录调整为当前目录,或者通