



北京科技大学
University of Science and Technology Beijing

密级：公开

本科生毕业设计(论文)

题 目：异构多模型自适应卡尔曼滤波
和目标追踪仿真研究

作 者：李孟祺

学 号：41721158

学 院：自动化学院

专 业：自动化

成 绩：

2021 年 6 月

本科生毕业设计(论文)

题 目: 异构多模型自适应卡尔曼滤波
和目标追踪仿真研究

英文题目: The simulation research of heterogeneous
multi-model adaptive KF and target tracking

学 院: 自动化学院

班 级: 自 175

学 生: 李孟祺

学 号: 41721158

指导教师: 张维存 职称: 副教授/博导

指导教师: 职称:

声 明

本人郑重声明：所呈交的论文是本人在指导教师的指导下进行的研究工作及取得研究成果。论文在引用他人已经发表或撰写的研究成果时，已经作了明确的标识；除此之外，论文中不包括其他人已经发表或撰写的研究成果，均为独立完成。其它同志对本文所做的任何贡献均已在论文中做了明确的说明并表达了谢意。

学生签名：_____年__月__日

导师签名：_____年__月__日

毕 业 设 计 (论 文) 任 务 书

一、学生姓名：李孟祺

学号：41721158

二、题目：异构多模型自适应卡尔曼滤波及目标追踪仿真研究

三、题目来源：真实 ☐ 、 自拟 ☒

四、结业方式：设计 ☒ 、 论文 ☐

五、主要内容：

- 1、掌握卡尔曼滤波器与目标跟踪的工作原理；
- 2、掌握异构多模型卡尔曼滤波器与目标跟踪的工作原理；
- 3、掌握异构多模型卡尔曼滤波器与目标跟踪中的加权学习算法；
- 4、利用 MATLAB 语言实现异构多模型卡尔曼滤波器与目标跟踪的仿真验证。

六、主要（技术）要求：

- 1、熟悉 MATLAB 编程语言；
- 2、异构多模型卡尔曼滤波与目标跟踪算法；
- 3、系统不确定性的模型集合建模，包括状态方程模型阶数/参数的不确定性，各种噪声方差的不确定性；
- 4、将仿真效果图形曲线显示在人机界面上；
- 5、人机界面上可以任意修不确定性参数。

七、日程安排：

第七学期

第 15-16 周 查阅资料，学习异构多模型卡尔曼滤波与目标跟踪的相关理论知识；

第 17-18 周 完成文献综述，撰写开题报告；

第八学期

第 1-2 周 完成课题需求分析和任务分解，撰写并提交开题报告；

第 3-4 周 完成基于 MATLAB 的异构多模型卡尔曼滤波器的目标跟

第 5-6 周 完成基于 MATLAB 的异构多模型卡尔曼滤波器的目标跟踪系统仿真程序，并满足相应的技术要求；

第 9-10 周 设计针对各种不确定性情况的仿真算例并进行仿真验证;

第 14-16 周 撰写论文, 准备答辩。

[1] 付梦印, 邓志红, 闫莉萍. Kalman 滤波理论及其在导航系统中的应用
[M]. 北京: 科学出版社, 2003

[3] Magill D. Optimal adaptive estimation of sampled stochastic processes[J] . IEEE Transaction of Automatic Control, 1965, AC-10: 435-439

指导教师签字: _____ 年 月 日

学 生 签 字: _____ 年 月 日

系（所）负责人章： 年 月 日

摘 要

在信息时代,信号作为载体,是传播信息的重要介质。在本自动化专业,信号也是着重研究对象。但在现实中,信号通常在传输过程和测量时参杂了噪声,使得收到的信号波形失真,不能还原真实的初始形态。由此产生了滤波的理论和技巧,用来分离有用信号和无规则噪声。滤波按照种类,分别为硬件及软件滤波,本课题的研究对象为数据滤波,本质是在统计的理论基础上,对数据进行加权平均。

如果我们事先知道系统目标的初始条件,卡尔曼滤波算法可以通过自身的估计和观测到的,含有噪声的,观测值做出一定的预测,随着时间的迭代逼近真值。作为在工业界耳熟能详的滤波方法,Kalman 滤波在通信、导航、控制、跟踪等领域应用十分广泛。

十九世纪六十年代,通过发表了的一篇文章《A New Approach to Linear Filtering and Prediction Problems》^[1],综合之前自己的博士论文,卡尔曼提出了一种新的,区别于传统滤波器的滤波估计理论,即后来所谓的卡尔曼滤波。此后,人们根据现实的问题提出了更多的 Kalman 滤波拓展算法。本课题就是在 Kalman 滤波算法上进一步拓展,引入了自适应特性和异构多模型的要求,以最小均方误差为优化目标,将原来的线性单一目标拓宽到多维不同模型的目标。相应的,滤波算法也会优化过滤效果,尽可能地贴近原始信号。

为了验证此项理论技术的实际效果,本课题会使用 Kalman 滤波最常用的应用场景——目标跟踪,进行仿真验证,对比真实系统状态,估计状态和卡尔曼优化的结果。经过 MATLAB 的验证,可以得到在此套算法的优化下,能够很好地估计目标状态,使信号更平滑,与真实信号更加贴近,满足课题对目标的要求。

关键词: Kalman 滤波, 最小均方误差, 目标跟踪, 自适应, 多模型

The Simulation Research of Heterogeneous Multi-Model Adaptive KF and Target Tracking

Abstract

In the information age, signals, as carriers, are important media for spreading information. But in reality, the signal is usually mixed with noise during transmission and measurement, which makes the waveform of the received signal distorted and cannot restore the true initial form. This resulted in the theory and technology of filtering, used to separate useful signals and random noise. Filtering is divided into hardware filtering and software filtering. The research object of this subject is data filtering, which is essentially based on statistical theory and weighted average data.

Kalman filtering enables to predict the state of the system which is dynamic from a large quantity of data with measurement noise, only when this measurement variance is known. Because it is easy to implement computer programming, and can update and process the data, which are collected in real time. Kalman Filter is currently the most widely used filtering method, and it has been lots of applications of many fields, including telecom, navigation, guidance and trade predictions.

This topic is to further expand the Kalman filtering algorithm, introduce the requirements of adaptive characteristics and heterogeneous multiple models, take the minimum mean square error as the optimization goal, and expand the original linear single goal to the goal of multi-dimensional different models. Correspondingly, the filtering algorithm will also optimize the filtering effect and restore the original signal as much as possible.

In order to verify the actual effect of this theoretical technology, this project will use the most commonly used application scenario of Kalman filtering-target tracking, to perform simulation verification, compare the real system state, estimate the state and the result of Kalman optimization. After the verification of MATLAB, it can be obtained that under the optimization of this set of algorithms, after Kalman filtering, the signal is smoother, closer to the real signal, and meets the target's requirements of the subject.

Key Words: Kalman Filter, Minimum Mean Square Error, Target Tracking, Adaptive, Multi-Model

目 录

摘 要	I
Abstract	III
1 引 言	1
2 Kalman 滤波基础	3
2.1 卡尔曼滤波的背景	3
2.2 卡尔曼滤波的数学原理	4
2.2.1 算法的思想和来源	5
2.2.2 算法的具体数学推导	6
2.3 卡尔曼滤波的实际应用和仿真	10
3 Kalman 滤波拓展算法	14
3.1 拓展算法的背景	14
3.2 拓展算法的相关推导	15
3.2.1 拓展卡尔曼滤波(EKF)	15
3.2.2 无迹卡尔曼滤波(UKF)	17
3.3 拓展算法的应用	20
4 异构多模型自适应滤波算法及仿真	22
4.1 IMM 算法及背景	22
4.1.1 多模型自适应算法原理	22
4.1.2 CV 和 CA 模型	23
4.1.3 CT 模型	24
4.2 滤波算法的数学推导	26
4.3 目标跟踪仿真和比较	30
5 非技术因素的考虑与实现	37
6 结 论	39
参考文献	41
附录 A 程序清单	43
在学取得成果	59
致 谢	61

1 引 言

Kalman 滤波是 R. E. Kalman 于 1960 年代提出的一项用于过滤噪声的理论技术。在此之前，滤波的使用已经很常见，但常常受限于实际条件，过滤的效果并不能令人满意。维纳之前提出一种理论，能有效地优化过滤噪声，但对于信号和噪声在不平稳的情况下，此种方法就显得比较吃力，需要新的算法来解决这个问题，因此 Kalman 滤波诞生，成为了当时最有效的算法来解决维纳算法无法处理的滤波问题^[2]。

本科题的研究背景，正是从 Kalman 滤波出发，对这个算法的背景做阐述，并简要介绍其后续的发展和对工业界做出的贡献。对于传统的滤波方法，有效的使用前提是原生信号和噪声的频带不一样。1940 年，Wiener 和 Kolmogorov 在滤波理论中将目标和噪声的一系列统计特性代入，在信号和噪声均为相对平稳的前提下，使用数学统计中最优化的思路不断估计逼近真值，从而达到滤波目的，即为维纳滤波^[2]。

卡尔曼滤波是应用统计方法来设计权重的。其具体的数学推导将在第二章详细介绍。

基于初始的卡尔曼滤波理论，在 60 多年后的今天，现实工业及生活需要对噪声的过滤提出了更高的需求。现在各位学者都对卡尔曼滤波做了优化拓展，本课题也是同样扩大化研究对象，设计异构多模型的建模体系，并改进原有的算法，使新的算法能对不同模型下的对象，以最小均方误差为优化目的，实施新的过滤噪声的应用。

卡尔曼滤波的算法在各个领域都有着广泛的应用，这里我们简述一个导弹拦截的例子。在雷达探测到来袭的导弹时，会生成实时更新的观测数据，这里 Kalman 滤波算法就可以根据以前的轨迹来估计导弹下一时刻的落点方向，所以本课题的仿真验证也会基于目标跟踪这一方向，以 MATLAB 为仿真平台，设计相应算法，对机动目标的运动进行仿真，同时以图像形式来对比仿真前后的效果。

为了充分深入研究课题，完成毕设，所设想的研究方法如下：在研究课题的理论知识时，需要结合导师推荐的参考书目和自己通过其他资源查阅到的文献来理解课题内容的基础知识。在掌握基础知识后，就可以根据内容的概念来详细弄清楚这个课题的发展和来龙去脉，知道其在现实生活中具体应用。最后对其进行仿真查看相应的效果。

在获知毕设内容的具体概念后，就涉及本科的专业课数学基础《现代控

制理论》。这门课是现代控制的具体概念配合线性代数的基础学科。在现实的噪声会被提取成噪声矩阵用于计算在系统中的偏差。通过建模，方框架构优化，数学运算可得到具体的加权算法，从而获得最优的滤波效果。

实验设计方面也会逐步深入，先从最基础的一维卡尔曼滤波开始验证，以温度测量为例，比较滤波前后的温度曲线；在从二维的目标机动运动出发，其中涉及到方向，速度，位置的变化，同样会比较滤波后的效果。预期结果应达到实际需要的标准，可以切实应用到现实领域对原有信号的还原。

论文的主要内容安排如下：

第一章是引言部分，简要介绍一下课题的历史渊源和应用范围，说明了自己的研究方法和预期目标。

第二章是对 Kalman 滤波算法的详细介绍，叙说了当时存在的具体问题，构建的相关模型和具体算法，并对此进行了相关的数学推导，为后面两章的引入做了一定的铺垫。

第三章是对卡尔曼后续拓展做了行业内工作的具体介绍，应用了哪些新的算法，其中的一部分会引用到自己的模型架构上，同时开始做多模型自适应架构的准备。

第四章是全面论证异构多模型自适应算法的内容，先是说了一下本课题的背景，然后说了一些具体操作内容，包括如何搭建所需要的模型，针对模型提出的新算法，其中需要到具体在 MATLAB 上使用新算法编写代码，对一维，二维等逐渐复杂的场景进行验证，并给出图像对比来体现滤波后的效果。

最后，在第五章对本文内容进行了总结，并对未来的改进工作提出了自己的想法。

2 Kalman 滤波基础

本课题为异构多模型自适应卡尔曼滤波和目标跟踪仿真，其理论技术均是基于 R. E. Kalman 在 1960 年代提出的卡尔曼滤波算法。在后续众多的卡尔曼滤波技术拓展衍生中，都离不开这一最基础的理论支撑，所以在这一章将详细介绍线性卡尔曼滤波的背景，研究意义，数学推导，算法和应用场景，这部分内容既提供了基础的理论依据，也为后面章节的论述做了一定的铺垫，因此有必要详细论述。

2.1 卡尔曼滤波的背景

滤波这一词在日常生活中并不少见，尤其在理工科范围内，滤波是很多项目里不可避免的一环，它的效果好坏能直接影响整个结果的质量。简单地说，滤波就是在获得接收信号时，对现在的信号极其不满意，因为被传输过程中的噪声给干扰了，至于什么是噪声，会在下面一段细说，即想要还原信号原始的样子。这里我们介绍的就是通过一种软件算法来还原信号。

什么是噪声？与一般人认知不同的是，在他们眼中，噪声类似于马路上车流的嘈杂声，办公室里他人交流时模糊不清的说话声等。但在滤波，或称估计领域，噪声意味着对我们没有用的信息都是噪声，哪怕你的形状是优美的正弦波。因为在听取信息的过程中，有噪声的参杂，我们听到的和实际发生出信号有着或多或少的差别，这里就叫做误差，比如耳机播放的音乐即使是最高规格的无损形式，和录音棚里的声源都有着很大的差别。如果存在一种噪声，经过统计它服从高斯分布，而且是标准的，则称之为高斯白噪声。幸运的是，我们常见的采集噪声大多都是高斯白噪声，否则滤波的难度要比现在高至少两个档次。

在我们常见的工程场景下，比如数据采集类的一系列活动，包含位置，图像，声音等，在采集过程中都会受到噪声的干扰，让接收的数据不能和声源信号相匹配。这些在空气，电路传输层等发生的噪声干扰都极有可能让我们采集到的数据失效，所做的一切都白费了。这时候我们要通过滤波这种手段，排除噪声的干扰，但往往只是减弱，并不能消除。

根据以上对滤波的介绍，就可以开始介绍最优滤波理论。这个理论也被称为最优估计，只不过当时滤波为当时较为流行的研究方向，所以两个说法都会一块使用。在统计学的数学定义中，这个算法是基于最小方差标准上的滤波和估计，通俗的说如果在图像上画出来，估计值和真实值要能有多接近

就有多接近。本文研究的 Kalman 滤波理论和下面要简要介绍的 Wiener 滤波都隶属于最优滤波理论这个范畴中。两个使用的思路方法完全不同，后者的思路明显更适用于现代的使用环境^[2]，即现代控制理论。

经典 Wiener 滤波方法是由控制理论鼻祖 N. Wiener 在 1940 年代于二战期间出于研究对怎样控制火炮落位俯仰角的姿态的问题而提出的^[2]。这个滤波方法的基本思路是平稳随机过程谱分解，但适用的场景也必须是信号为平稳随机过程，更致命的是因为滤波算法不能递推迭代，所以要存储全部历史数据，对应的存储量和计算量都很难在当时的现实条件下实现，不便于实时计算，只能在单通道平稳随机场景下勉强够用。人们过去尝试将 Wiener 滤波看看能否应用到多维非平稳的场景，都因为无法突破计算上的困难而以失败告终。

因为使用频域法的思路进行计算，Wiener 滤波发展极其困难以至于人们不再对其进行深入研究，而是转头探索一种全新的架构方法来进行滤波，由此 Kalman 滤波应运而生^[3]，成为了新一代的最优滤波器的理论。采用了更符合人类直觉的时域算法，他往随即估计里加入了现代控制理论中的状态空间的概念，从而将信号从传统的传递函数变化为可随时观测的状态向量，整个系统的输入输出均是通过状态空间方程来表示。在方程中包含过渡，观测和噪声等因素，由此构成整个滤波算法。相比传统的 Wiener 滤波，Kalman 滤波具备巨大的优势以至于迅速的淘汰了传统算法，它不仅能够估计平稳的一维随机过程，还能估计非平稳多维过程，而且是迭代的关系，所以计算机的运行效率很高，这些都是前者所不具备的。

在现实生活中，Kalman 滤波理论在统计估计学术领域内占据着极其重要的位置。和“硅”在电子元器件的重要性一样，它的应用领域在提出以来得到广泛的发展，包括跟踪，导航，飞行器控制等各方面。因为其数学表示的先进性，它可以实时更新目标的各种状态变量。为了能够完成控制动态系统的功能，第一步需要获知被控对象的实时状态。对于复杂的动力系统，通常不可能或难以测量所有需要控制的变量。卡尔曼滤波使您可以使用测量信息（包括这种间接受限的噪声）来估计原始信息。

2.2 卡尔曼滤波的数学原理

卡尔曼滤波的本质是最小均方误差（Minimum Mean Squared Error, MMSE），即在模型已知（分布已知），参数未知的情况下，使误差平方和取均值再开方的结果最小，它反映了估计值与被估计值之间差异化程度的大小，

它的值越小，越能说明预测模型描述实验的效果越精确。所以在已知的状态空间方程的基础上，我们引入了估计值和预测值（修正值），使得每过一个时间，这些参数都会实时更新，从而实现迭代的效果。以下就是 Kalman 滤波算法的思想和具体数学推导。

2.2.1 算法的思想和来源

为了能够更加轻松的理解 Kalman 滤波器，我们用形象的例子来说明，然后我们根据其核心的 5 个公式进行进一步的说明和探索。在现代的计算机帮助下，Kalman 程序还是比较易读可修改的，前提是要充分理解并能熟练运用这五个核心方程。

在说明它的 5 个核心方程之前，我们先用下面的这个例子做一个直观的解释。

假设您要研究的主题是在室温下。根据直觉和生活经验，这个房间的温度是保持不变的。换言之，下一瞬间的温度与该瞬间的温度相同。假设您不完全相信自己的经验，它可能会波动很多次。这些偏差被认为是高斯白噪声。也就是说，这些偏差是相互独立的，与前后时间无关。此外，虽然内部安装了温度计，但该温度计并不完全准确，可能与实际值存在差异。这些偏差在这种场景下就可以直接当作高斯白噪声^[4]。

到目前为止，有两个特定时间的环境温度值。一个是经验估计值（估计系统值），另一个是来自温度计的值（测量值）。在统计学里面，我们可以找到相应的数学原理将两者混合起来估计真实的室温。

假设您要估计时间 k 的实际温度。首先，为了估计 k 时刻的温度，我们需要知道时间 $k-1$ 的温度值。很显然，我们根据之前的设定，前后时刻的温度的估计值都是保持一致的，为 23° ，因为我们认为温度保持不变，高斯噪声与该值的偏差为 5° 。在时间 $k-1$ 与估计温度值的偏差为 3，关于您的对象的估计不确定性为 4° ，然后将它们平方以提取平方。最后可以从温度计读取时间 k 的温度值。如果是 25° ，则与该值的偏差为 4° 。

确实，有两个温度值可以估算时间 k 的真实温度： 23° 和 25° 。但显然实际温度是唯一确定的，不可能出现两个实际温度，所以我们会判断哪个数据会更准确一点。这时候，数学知识告诉我们协方差可以作为判定的标准。通过计算可以得到 K_g 的值为 0.78，由此可以得出我们估计的温度值的计算式如下： $23+0.78*(25-23)=24.56^\circ$ 。根据这个结果我们得知，估计结果更偏向于 23° ，即验证了温度计自身的协方差较小。

我们现在有了 k 时刻的理想温度值。下一步是输入时间 $k+1$ 以执行新的最优估计。目前，我看不到任何事情可以再做。以前，在输入时间 $k+1$ 之前，需要计算时间 k 与最佳值的偏差。计算方式： $((1-K_g) * 5^2)^{0.5} = 2.35$ 。其中 5 为上述时间 k 温度估计值偏差 23° ，输入 $k+1$ 后 k 时刻温度估计值偏差为 2.35° 。

因此，卡尔曼滤波器可以在核心方程里不断迭代协方差来估计理想的温度值。它工作得非常快，并且只保留前一状态的数据。计算式的 K_g 是卡尔曼增益。

由以上以一维温度测量的例子可知，**Kalman** 滤波的本质在于迭代更新，从而不断估计。其中，卡尔曼增益是基于最小均方误差推导出来的，详细的数学解释将在下一节具体给出。

卡尔曼滤波是统计学的数学表达。要完全掌握 **Kalman** 滤波的算法使用，必须有一个良好的概率论和数理统计的数学底子。所以我们在学习它的基本思想后就需要严格推导所需要的 5 个公式，从而为后面的使用做铺垫。

卡尔曼滤波算法具有如下特点：

- (1) 核心迭代方程仅有 5 个，计算量较小
- (2) 采用时域量，不需要频域转换，导出的数据可以直接使用
- (3) 存储量小，对硬件的要求较少

2.2.2 算法的具体数学推导

在正式进行数学推导之前，我们需要对一些概念定义和相关量的说明。其中 X_k 是一个 n 维列向量， u_i 是 X_i 的期望，协方差矩阵为： $P_k = E[e_k e_k^T]$ ；显然，根据数学推理可得，协方差阵是正定的并且等于自身的转置，协方差矩阵的迹 $tr(P_k)$ 是 X_k 的均方误差。

一. 系统模型与变量说明

1. 系统离散型状态方程如下：

从 $k-1$ 到 k 时刻，系统状态估计方程

$$X_k = AX_{k-1} + Bu_{k-1} + w_k \quad (2-1)$$

系统状态观测方程

$$Z_k = HX_k + v_k \quad (2-2)$$

2.变量说明如下:

A : 状态转移矩阵

u_{k-1} : 系统输入向量

B : 输入增益矩阵

w_k : 形式为高斯白噪声的过程噪声

H : 测量矩阵

v_k : 形式为高斯白噪声的测量噪声

根据理论可知,我们认为初始状态和每一时刻的噪声 $X_0, w_1, w_2, \dots, w_k, v_1, v_2, \dots, v_k$ 都是没有直接关系的,即在统计特性上是相互独立的。

二. 卡尔曼滤波器

卡尔曼滤波估计的计算过程分成两个部分,估计与校正。在第一阶段的估计部分,我们会使用上一时刻的状态值。在修正阶段,滤波器在已经得知上一时刻的一步预测值的情况下加入观测值进行一个融合算法的修正,就可以得到一个新的估计值^[1]。

1.变量说明

x_k : 真实值

\hat{x}_k : 卡尔曼预测值(后验)

\hat{x}_k^- : 卡尔曼估计值(先验)

P_k : 卡尔曼预测误差协方差矩阵

P_k^- : 卡尔曼估计误差协方差矩阵

K_k : 卡尔曼增益

\hat{z}_k : 测量余量

2.卡尔曼滤波器计算过程

估计阶段:

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k \quad (2-3)$$

$$P_k^- = AP_{k-1}A^T + Q \quad (2-4)$$

修正阶段:

$$\hat{z}_k = z_k - H\hat{x}_k^- \quad (2-5)$$

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \quad (2-6)$$

$$\hat{x}_k = \hat{x}_k^- + K_k \hat{z}_k \quad (2-7)$$

更新协方差预测:

$$P_k = (I - K_k H) P_k^- \quad (2-8)$$

根据上面核心的六个迭代方程,我们在计算过程中要着重弄清楚 P_k^- , K_k 的算法原理,以及 P_k 的迭代算法。

3.卡尔曼滤波算法详细推导过程

从协方差矩阵开始说起,真实值与预测值之间的误差为

$$\begin{aligned} e_k &= x_k - \hat{x}_k = x_k - (\hat{x}_k^- + K_k(Hx_k + v_k - H\hat{x}_k^-)) \\ &= (I - K_k H)(x_k - \hat{x}_k^-) - K_k v_k \end{aligned} \quad (2-9)$$

卡尔曼预测误差协方差矩阵为

$$P_k = E[e_k e_k^T] \quad (2-10)$$

将真实值与预测值之间的误差代入得到

$$\begin{aligned} P_k &= E[(I - K_k H)(x_k - \hat{x}_k^-) - K_k v_k][(I - K_k H)(x_k - \hat{x}_k^-) - K_k v_k]^T \\ &= (I - K_k H)E[(x_k - \hat{x}_k^-)(x_k - \hat{x}_k^-)^T](I - K_k H)^T + K_k E[v_k v_k^T] K_k^T \end{aligned} \quad (2-11)$$

其中, $E[v_k v_k^T] = R$, 并将其带入到上式的协方差方程中, 得到

$$P_k = (I - K_k H) P_k^- (I - K_k H)^T + K_k R K_k^T \quad (2-12)$$

因为卡尔曼滤波的本质是基于最小均方差的估计算法, 而 P_k 的迹就是我们要求的均方差, 将上式展开并求迹可得

$$tr(P_k) = tr(P_k^-) - 2tr(K_k H P_k^-) + tr(K_k (H P_k^- H^T + R) K_k^T) \quad (2-13)$$

最优估计 K_k 使矩阵的迹最小, 所以上式两边对 K_k 求导

$$\frac{\partial \text{tr}(P_k)}{\partial K_k} = \frac{\partial \text{tr}(2K_k H P_k^-)}{\partial K_k} + \frac{\partial \text{tr}(K_k (H P_k^- H^T + R) K_k^T)}{\partial K_k} \quad (2-14)$$

根据矩阵微分学的相关公式，得到

$$\frac{\partial \text{tr}(P_k)}{\partial K_k} = -2(H P_k^-)^T + 2K_k (H P_k^- H^T + R) \quad (2-15)$$

令上式等于 0，可以得到

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \quad (2-16)$$

至此，我们就已经得知卡尔曼增益是怎么计算出来了，接下来是计算 P_k^- 的具体步骤

$$\begin{aligned} P_k^- &= E[(x_k - \hat{x}_k^-)(x_k - \hat{x}_k^-)^T] \\ &= E[(A(x_{k-1} - \hat{x}_{k-1}) + w_k)(A(x_{k-1} - \hat{x}_{k-1})^T)] \\ &= E[(A e_{k-1})(A e_{k-1})^T] + E[w_k w_k^T] \\ &= A P_{k-1} A^T + Q \end{aligned} \quad (2-17)$$

所以估计误差协方差矩阵 P_k^- 可以由上一次算出的预测误差协方差矩阵 P_{k-1} 及转移矩阵 A 和噪声协方差阵 Q 算出。

这里将 K_k 的值代入一开始的 P_k 表达式中，即可获得

$$\begin{aligned} P_k &= P_k^- - K_k H P_k^- - P_k^- H^T K_k^T + P_k^- H^T (H P_k^- H^T + R)^{-1} (H P_k^- H^T + R) K_k^T \\ &= P_k^- - K_k H P_k^- \\ &= (I - K_k H) P_k^- \end{aligned} \quad (2-18)$$

4.总结

卡尔曼滤波的更新过程如下：

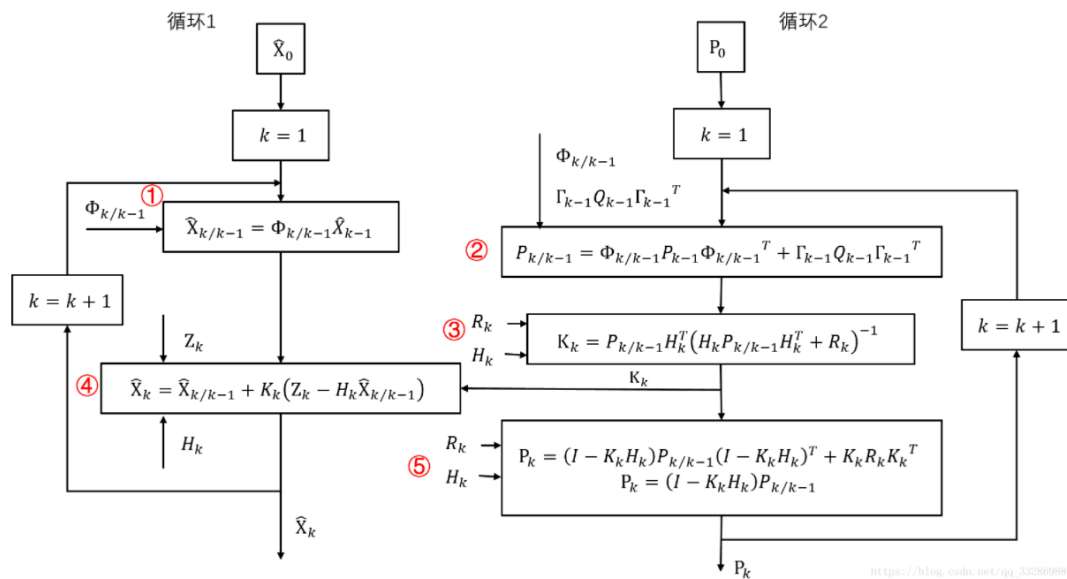
第一步，首先 P_0 ， X_0 已知，然后由 P_0 推出 P_1^- ，再由 P_1^- 算出 K_1 ，有了以上这些参数后，结合观测值就能预测出 X_1 ，再利用 K_1 更新 P_1 ；

第二步, 因为 P_1 , X_1 已知, 然后由 P_1 推出 P_2^- , 再由 P_2^- 算出 K_2 , 有了以上这些参数后, 结合观测值就能预测出 X_2 , 再利用 K_2 更新 P_2 ;

.....

第 n 步, 已知 P_{n-1} , X_{n-1} 已知, 然后由 P_{n-1} 推出 P_n^- , 再由 P_n^- 算出 K_n , 有了以上这些参数后, 结合观测值就能预测出 X_n , 再利用 K_n 更新 P_n ;

图 2-1 就是通过一组流程图来形象化描述滤波的整个迭代过程:



这就是卡尔曼滤波的递推过程。

2.3 卡尔曼滤波的实际应用和仿真

综合上述的两节内容, 我们已经清晰明了地知悉卡尔曼滤波诞生的相关背景, 它的数学指导思想和和具体的数学表达推导。以上皆是卡尔曼滤波在理论上所解释和展现的内容, 实际上还需要落实到现实应用之中。我们查阅相关资料, 可以得知此项理论技术应用于如下方面。

卡尔曼滤波器的首次应用是由 Stanley Schmidt 完成的。在访问美国宇航局艾姆斯研究中心期间, 卡尔曼恰巧发现阿波罗登月的轨道估计的难题可以通过施密特的想法完美解决, 之后阿波罗航天器的导航员使用了这种类型的过滤器。1860 年代, 卡尔曼发表了卡尔曼离散系统滤波器相关的文章。1961 年, 他与 R.S. Bushy 合作, 将这种滤波方法扩展到连续时间这个更广泛的领

域，完全塑造了卡尔曼滤波器设计理论^[5]。

可以由上文看出，**Kalman** 滤波相比其它滤波就像是一个新时代的产物，所以它一经提出就可以不经过长时间的工程验证而是直接在生产活动中使用^[6]。例如，使 **Kalman** 滤波一炮成名，驰名于全世界工业界的经典之作就是美国的阿波罗登月成功和飞机导航。

由上面的论述可知，**Kalman** 滤波的应用场景非常广泛，只要是一个系统模型能够离散化，同时里面存在随机干扰，我们就可以对所求的估计量建立方程并迭代出与真值极其接近的数据。简略的说，**Kalman** 的大范围应用场景有如下五点。

- 1) 移动物体定位，导弹的飞行和跟踪领域。
- 2) 天气预报，地震提前预警。
- 3) 地质勘探，矿物开采。
- 4) 故障诊断，检测。
- 5) 证券股票市场预测。

根据上面五个小点的简要介绍，下面将详细说明 **Kalman** 滤波在目标定位方面的具体应用场景。

全球定位系统（Global Positioning System, GPS）在日常开车导航，导弹的追踪定位方面有着非常重要的地位。船舶 GPS 导航定位原理如图 2-2 所示。在移动目标上可以安装一个 GPS 接收仪器，它能够具备对目标的进行一系列的运动计算。具体原理是接收仪器在收到在轨卫星的信号后，会通过一定的算法得到相应的位置和速度。GPS 传输的真实信号会添加人工的高频振荡随机信号，根据我们在大学物理所学习到的知识，这个噪声会致使目标脱离稳定状态，因此所有的相关卫星信号都会产生高频波动。为了提高定位精度，即在相同的条件下对目标的锁定范围的半径进一步缩小，需要根据船舶的位置（position）和速度（velocity）对 GPS 观测信号进行滤波。人为加入 GPS 系统的高频干扰信号可以认为是 GPS 位置观测噪声，观测噪声的强度（散射）可以通过 GPS 观测系统用系统识别方法得到。

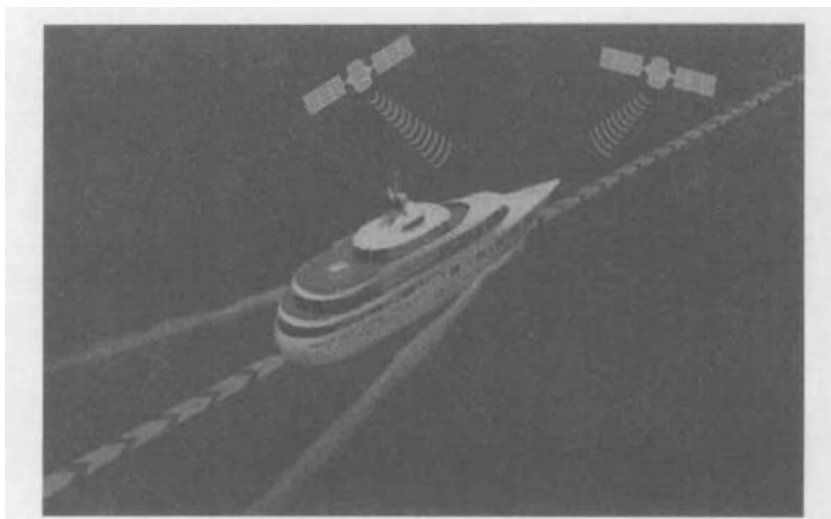


图 2-2 GPS 导航定位原理

下面是针对算法的一次温度检测滤波的仿真，使用了 MATLAB 仿真平台。先假设一般性的温度不变的条件模型，根据以下预设条件，我们可以建立起相关数学模型，从而实现代码编写和仿真。

现在要研究一个房间的温度。假设房间温度预设值为室温 25° （没有任何的干扰，分子热运动稳定，热传导稳定）。但是在实际状况下，空气流通、光变化的影响，真实的实际温度在波动，这个干扰就是过程噪声 W ，预设 $W \sim N(0, 0.01)$ 。现在用某温度计测量室温，假设温度计测量误差比较大，测量噪声服从 $V \sim N(0, 0.25)$ 。不失一般性，假设转移矩阵 $A=1$ ，过程噪声增益 $G=1$ ，观测矩阵 $H=1$ ，初始真实温度 25.1° ，初始测量值 24.9° ，因为没有控制输入量，所以 $u=0$ 。接下来就是具体的代码和注释，最后会给出仿真的结果和相关比较。

下面就是具体的仿真结果：

分析：由图 2-3 和图 2-4 可知，在线性卡尔曼滤波后，预测值可以很好的抑制观测值带来的波动，与实际值更加贴近，从偏差角度来看，其与真实值之间的偏差也远小于观测值和真实值之间的偏差，做到了对温度变化的很好的估计作用，从形式上看起到了优良的滤波效果。

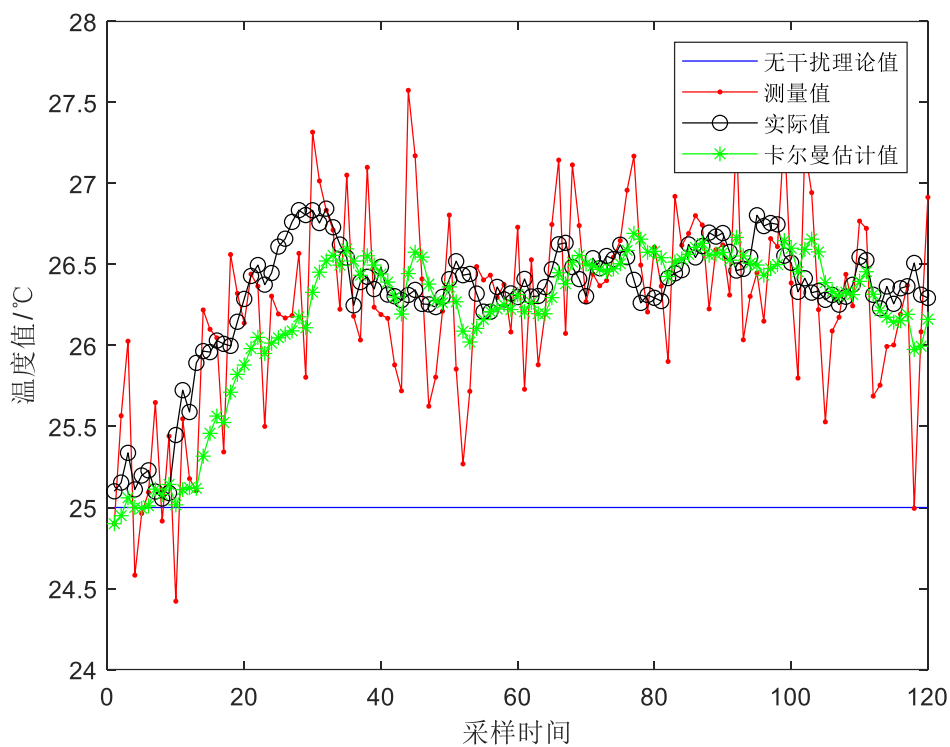


图 2-3 温度指标的相关比较

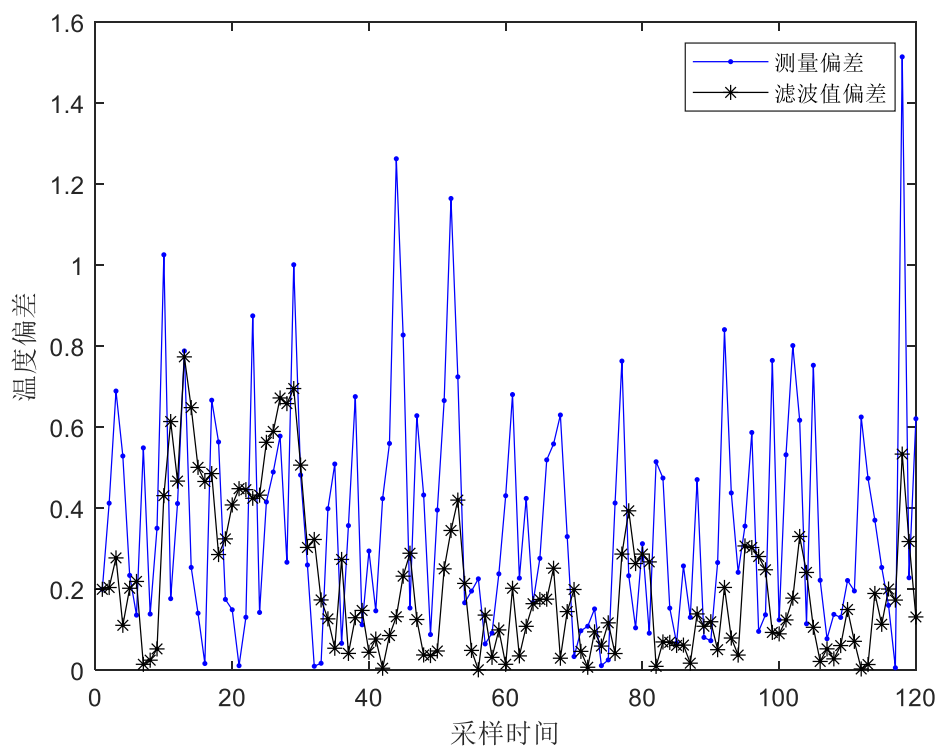


图 2-4 偏差之间的比较

3 Kalman 滤波拓展算法

在前一章我们详细了解了卡尔曼滤波的思想，具体数学方法和实际应用场景，但这种算法仅适用于线性系统的条件下，如果在非线性的条件下，滤波的实际效果和使用效率将大打折扣，甚至完全失效，所以在后续的工业使用场景中，需要引入新的，在线性卡尔曼滤波基础上拓展的算法，从而满足相应的要求^[7]。以下将对我们常见的拓展卡尔曼滤波（EKF）以及无迹卡尔曼滤波（UKF），做一个背景的说明和数学原理推导。

3.1 拓展算法的背景

卡尔曼滤波的拓展算法起源于现实工业的需要和解决传统卡尔曼滤波的应用缺陷，即无法在非线性系统中实现精准的跟踪和滤波功能。学过高数的同学都知道，面对非线性问题我们直觉上会通过泰勒展开转化成线性问题^[8]。

拓展卡尔曼滤波的优势在于非必须提前计算轨迹，即所谓的观测噪声 $\mathbf{V}(k)$ 和过程噪声 $\mathbf{W}(k)$ 均为 0 时非线性方程的解，但它的使用场景仅限于滤波误差 $e_k = x_k - \hat{x}_k$ 及一步预测误差 $\tilde{e}_{k,k-1} = x_k - x_{k-1}$ 较小时才能使用。

以上是对拓展卡尔曼滤波（EKF）的背景介绍，下面将针对 EKF 缺点新引进的一种算法，无迹卡尔曼滤波（Unscented Kalman Filter, UKF），具体介绍如下。

上面讨论的拓展卡尔曼滤波算法（EKF）是对函数进行泰勒展开并根据实际情况保留一定阶数的残余项，这样就会引入线性化误差。线性化误差在一定程度上会极大地影响滤波的性能，同时在计算理论上这种方法也是不易于实现的。

无迹卡尔曼滤波（UKF）和之前的拓展卡尔曼滤波算法完全不同，对于一步预测方程，采用无迹变换（Unscented Transform, UT）来解决协方差和均值相应的传递问题^[9]。在后面的数学推导和应用中举例中，可以清楚的知道 UKF 相比 EKF 的优势在于稳定性更高，但在一些特殊场景下效果会较前者差一些。

对于上述两种衍生的卡尔曼滤波，均是源于实际生活中遇到的更复杂的系统，原有的卡尔曼滤波算法不再满足需要的滤波性能表现，所以通过线性化和构建新架构的方法来开发新的滤波原理。在第二节将会详细推导两者的数学原理并给出相应的解释。

3.2 拓展算法的相关推导

下面将对两种线性卡尔曼滤波算法的拓展进行详细的数学推导。

3.2.1 拓展卡尔曼滤波(EKF)

拓展卡尔曼滤波原理：先是局部线性化，然后是线性卡尔曼滤波，总体是一个迭代的过程。

非线性系统动态方程具体形式如下

$$X_{(k+1)} = f[k, X_{(k)}] + G_{(k)}W_{(k)} \quad (3-1)$$

$$Z_{(k)} = h[k, X_{(k)}] + V_{(k)} \quad (3-2)$$

显而易见过程噪声 $W_{(k)}$ 和观测噪声 $V_{(k)}$ 保持为 0 时，系统模型（3-1），（3-2）的解为非线性模型的理论解，即所谓的假轨迹，而把非线性系统（3-1）和（3-2）的真实解称为真实轨迹。

EKF 因为自身带有可以将非线性进行线性化的特性，对非线性模型进行线性化处理。

由系统状态方程（3-1），将非线性函数 $f(*)$ 在对应需要的值附近进行高数中常见的 Taylor 展开，可以得到

$$X_{k+1} = f[k, \hat{X}_k] + \frac{\partial f}{\partial \hat{X}_k} [X_k - \hat{X}_k] + G[\hat{X}_k, k]W_k \quad (3-3)$$

令

$$\frac{\partial f}{\partial \hat{X}_k} = \frac{\partial f[\hat{X}_k, k]}{\partial \hat{X}_k} \Big|_{\hat{X}_k = X_k} = \phi_{k+1|k} \quad (3-4)$$

$$f[\hat{X}_k, k] - \frac{\partial f}{\partial \hat{X}_k} \Big|_{\hat{X}_k = \hat{X}_k} \hat{X}_k = \phi_k \quad (3-5)$$

则状态方程为

$$X_{k+1} = \phi_{k+1|k}X_k + G_kW_k + \phi_k \quad (3-6)$$

同卡尔曼滤波基本方程相比，在已经知道上一时刻的 \hat{X}_k 的条件下，状态方程增加了干扰 ϕ_k 。

由系统状态方程 (3-2)，对非线性函数 $h(*)$ 在滤波值附近 \hat{X}_k 做泰勒展开，可以得到

$$Z_k = h[\hat{X}_{k|k-1}, k] + \left. \frac{\partial h}{\partial \hat{X}_k} \right|_{\hat{X}_{k,k-1}} [X_k - \hat{X}_{k|k-1}] + V_k \quad (3-7)$$

令

$$\left. \frac{\partial h}{\partial \hat{X}_k} \right|_{X_k = \hat{X}_k} = H_k \quad (3-8)$$

$$y_k = h[\hat{X}_{k|k-1}, k] - \left. \frac{\partial h}{\partial \hat{X}_k} \right|_{X_k = \hat{X}_k} \hat{X}_{k|k-1} \quad (3-9)$$

则观测方程为

$$Z_k = H_k X_k + y_k + V_k \quad (3-10)$$

对线性化后的模型，用线性 Kalman 方程计算可得拓展卡尔曼滤波的递推方程，形式和传统 Kalman 迭代方程一样^[8]，具体可参考第二章的相关数学方程，在此不再赘述。

同 Kalman 滤波基本方程相比，在线性化后的系统方案中， $\phi_{k+1|k}$ 和 H_{k+1} 以后用 f 和 h 表示。假设状态变量有 n 维，即 $X = [x_1 \ x_2 \ x_3 \ \dots \ x_n]^T$ ，则相应雅可比矩阵的求法如下：

$$\phi_{k+1} = \frac{\partial f}{\partial X} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \quad (3-11)$$

$$H_{k+1} = \frac{\partial h}{\partial X} = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \cdots & \frac{\partial h_1}{\partial x_n} \\ \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} & \cdots & \frac{\partial h_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial h_n}{\partial x_1} & \frac{\partial h_n}{\partial x_2} & \cdots & \frac{\partial h_n}{\partial x_n} \end{bmatrix} \quad (3-12)$$

由以上数学公式的推导和说明，我们可知拓展卡尔曼滤波的思想和数学原理，从而丰富了相关的理论，为以后的实际运用打下了坚实的基础。

3.2.2 无迹卡尔曼滤波(UKF)

下面则是无迹卡尔曼滤波的数学推导具体过程：

无迹卡尔曼滤波（UKF）是一种应对非线性过程函数时无法采用传统线性滤波的滤波方法。和拓展卡尔曼算法不同的是，它不需要做线性化处理，而是用一种数学上的点阵投射，这些投射点表示对应的高斯密度函数^[9]。

无迹卡尔曼变换的实现方法为：在原来的数据中我们选取一些数据点，令这些数据点的统计特性和原有数据的统计特性相同，把这些数据点代入非线性函数中，然后可以得到相应的结果^[10]。

非线性变换比较如下图 3-1 所示：

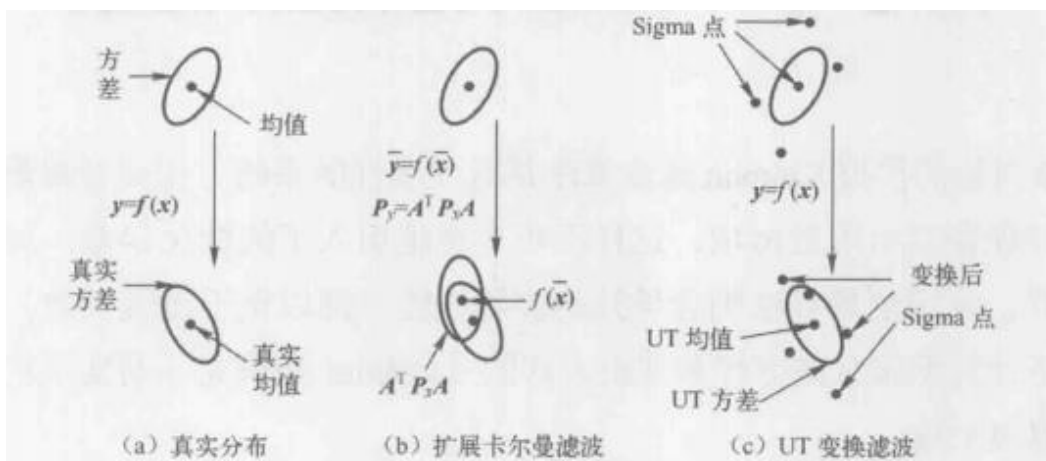


图 3-1 非线性变换比较

在这里我们为了更具科学性，使用了对称分布的采样方式，简单介绍以下 UT 的基本原理。

先计算 $2n+1$ 个 σ 点，即采样点，这里 n 指的是状态的维数；

$$\begin{cases} X^{(0)} = \bar{X}, i = 0 \\ X^{(i)} = \bar{X} + \left(\sqrt{(n+\lambda)P}\right)_i, i = 1 \rightarrow n \\ X^{(i)} = \bar{X} - \left(\sqrt{(n+\lambda)P}\right)_i, i = n+1 \rightarrow 2n \end{cases} \quad (3-13)$$

然后计算这些采样点对应的权值

$$\begin{cases} \omega_m^{(0)} = \frac{\lambda}{n+\lambda} \\ \omega_c^{(0)} = \frac{\lambda}{n+\lambda} + (1-a^2+\beta) \\ \omega_m^{(i)} = \omega_c^{(0)} = \frac{\lambda}{2(n+\lambda)}, i = 1 \rightarrow 2n \end{cases} \quad (3-14)$$

在方程 (3-14) 中, 对应的角标由上到下依次为均值和协方差。

无迹卡尔曼变换得到的 sigma 点集具有下述的性质:

样本均值和真实均值相同;

样本方差和真实方差相同;

正太分布的 sigma 点集由标准分布的 sigma 的 UT 变换得到。

由以上相关概念的建立, 就可以具体说明无迹 Kalman 滤波算法的具体实现步骤了。

对于不同时刻 k , 由具有高斯白噪声 $W(k)$ 的随机变量 X 和具有高斯白噪声的 $V(k)$ 的观测变量 Z 构成的非线性系统可以有以下式表达:

$$\begin{cases} X(k+1) = f(x(k), W(k)) \\ Z(k) = h(x(k), V(k)) \end{cases} \quad (3-15)$$

在方程 (3-15) 中, f, h 表示非线性函数, w, v 表示和线性卡尔曼一样的对应的协方差矩阵。

先利用方程 (3-13) 和 (3-14) 得到一组 sigma 点集及其权重;

$$X^i(k|k) = [\hat{X}(k+1|k) \quad \hat{X}(k|k) + \sqrt{(n+\lambda)P(k|k)} \quad \hat{X}(k|k) - \sqrt{(n+\lambda)P(k|k)}]$$

其中一步预测的结果如下, $i=1, 2, \dots$;

$$\hat{X}^i(k+1|k) = f[k, X^i(k|k)] \quad (3-16)$$

需要在计算中获得一步预测矩阵, 其中权值是通过式 (3-14) 得到。

$$\hat{X}(k+1|k) = \sum_{i=0}^{2n} \omega^i X^i(k+1|k) \quad (3-17)$$

$$P(k+1|k) = \sum_{i=0}^{2n} \omega^i [\hat{x}(k+1|k) - \hat{x}(k+1|k)][\hat{x}(k+1|k) - \hat{x}(k+1|k)]^T + Q \quad (3-18)$$

根据一步预测，我们使用 UT 变换，即能获得一组新的的 sigma 点集；

$$\begin{aligned} \hat{X}^i(k+1|k) &= [\hat{X}(k+1|k) \quad \hat{X}(k+1|k) + \sqrt{(n+\lambda)P(k+1|k)} \\ &\quad \hat{X}(k+1|k) - \sqrt{(n+\lambda)P(k+1|k)}] \end{aligned} \quad (3-19)$$

将上面得到的结果代入到观测方程中，可以得到估计的观测量， $i=1, 2, \dots, 2n+1$ ；

$$Z^i(k+1|k) = h[\hat{X}^i(k+1|k)] \quad (3-20)$$

由上式可以得到生成点集的观测估计值，最后经过一定的数学原理得到估计的均值和协方差；

$$\bar{Z}(k+1|k) = \sum_{i=0}^{2n} \omega^i Z^i(k+1|k) \quad (3-21)$$

$$P_{z_k z_k} = \sum_{i=0}^{2n} \omega^i [Z^i(k+1|k) - \bar{Z}(k+1|k)][Z^i(k+1|k) - \bar{Z}(k+1|k)]^T + R$$

$$P_{x_k z_k} = \sum_{i=0}^{2n} \omega^i [X^i(k+1|k) - \bar{Z}(k+1|k)][Z^i(k+1|k) - \bar{Z}(k+1|k)]^T$$

计算 Kalman 增益矩阵；

$$K(k+1) = P_{x_k z_k} P_{z_k z_k}^{-1} \quad (3-22)$$

最后，对系统的相应状态更新和对应的协方差进行更新；

$$\hat{X}(k+1|k+1) = \hat{X}(k+1|k) + K(k+1)[Z(k+1) - \hat{Z}(k+1|k)] \quad (3-23)$$

$$P(k+1|k+1) = P(k+1|k) - K(k+1)P_{z_k z_k} K^T(k+1) \quad (3-24)$$

由以上数学推导可以看出，通过非线性映射，这些 σ 点集可以近似地提取出状态概率密度^[2]。这个近似实质上是一种统计逼近而不是解。

3.3 拓展算法的应用

上面两章介绍了在拓展 Kalman 算法和无迹 Kalman 算法的背景和应用，这里将结合现当代的内容说明两者的一些实际工业应用。

在这里，我们先介绍拓展卡尔曼滤波在目标跟踪中的应用。这里先要介绍的是只观测距离的拓展卡尔曼算法，这里意思为通过距离传感器获得相应的观测值（如方位，距离等）对目标实施跟踪是目标运动分析领域的一个普遍问题，在许多的应用场景下可以获得精准的距离信息。一个例子如下：在水下射击弹道的测量过程中，得知监听器测量脉冲的到达时刻可以得到待测目标的距离信息，我们可以使用多传感器对目标实施仅距离跟踪定位；在海洋监测中，Ingara 雷达包含 EAR 成像和跟踪的两种模式。跟踪模式使用 ISAR 所生成高分辨率距离信息对海洋的目标进行探测和跟踪。所以，研究仅利用距离信息进行目标跟踪是十分有必要的。

接下来是基于 EKF 的纯方位目标跟踪算法。仅方位角的目标跟踪分析是一类隐藏攻击的有效办法，它是利用目标自身的主动辐射，如声波辐射，电磁波辐射，目标对辐射的散射，甚至目标实施的干扰辐射等，通过移动单站测量机动目标定位与跟踪^[11]。这种对运动目标的隐藏定位与跟踪是，反水声对抗，反侦察，对目标实施突然攻击的十分有效的手段。在实际环境的现当代战斗中，通常测量的敌舰（飞机）的特征信息非常有限，目标方位角基本上是唯一有用的参数。您可以使用目标的方位角来估计目标的运动。对敌舰（机器）进行直接攻击和电子干扰。以上是拓展卡尔曼滤波的一些衍生算法和实际的应用场景，接下来将具体介绍无迹卡尔曼滤波的一些相关应用。

第一种是将无迹卡尔曼滤波器应用于单个观测站的目标跟踪。换句话说，假设目标以恒定速度线性移动，并且通过以下假设已知目标的初始状态。目标由一个观察站观察，可以直接对目标的运动轨迹进行估计，这一方法克服了传统线性卡尔曼滤波的现实条件约束。同时 UKF 再匀加速直线运动方面也有目标跟踪的应用，这里需要构建观测角等新的变量，是同线性卡尔曼滤波完全不同的思想方法。

根据文献中的仿真结果^[2]，我们可得出以下结论：并不是每一次 UKF 估计的结果都比 EKF 估计的更准确，但是纵观全体，UKF 的估计误差会比 EKF 小。

在同等条件下，UKF 的滤波精度相比 EKF 有着巨大的优势。计算 RMSE 均值和方差，输出的结果则更有力地表明了 UKF 算法相对 EKF 算法的优秀之处。

UKF 算法基于 UT 变换和 Bayes 理论，在很大程度上解决了算法的线性化误差，具备广泛的应用场景。UKF 算法存在其参数选择上目前没有一个统一的解决方案的缺陷，而且其滤波效果和 EKF（拓展卡尔曼）算法一样，也受到滤波初值的影响。

4 异构多模型自适应滤波算法及仿真

在这一章节将详细介绍本课题的异构多模型自适应算法，包括它的实际背景来源，算法思想，数学推导和相应的跟踪仿真。其中在仿真部分会比较不同运动模式下滤波的效果如何，从而全面的展示这个这个滤波算法适用的真实条件和优劣。

4.1 IMM 算法及背景

在现实生活中，存在着对一运动目标进行跟踪的实际需求，尤其对于强机动目标而言（不仅仅只有匀速直线运动，还有曲线运动和加速运动），其运动方式十分复杂，往往采用一个模型已经很难表示其运动轨迹了，而选择多个模型对其跟踪且在不同状态下使用不同模型是一个很好的解决办法^[12]。多模型的基本思想就是对目标的同一时刻状态使用多个模型进行并行处理，再通过一定的加权算法将多个模型的跟踪状态进行结合，从而得到最适合的状态输出。这就是多模型自适应算法（Multiple Model Adaptive Estimation, MMAE）的思想基础。

什么是自适应算法？在目标跟踪中，这种算法无需提前判断目标的运动模型就可以实时地修正跟踪目标的轨迹参数，使其达到想要的贴合效果，具有优秀的适应性。但是当目标开始机动时，就是我们所说的突然转弯加速等，仅采用基础线性卡尔曼滤波，其只用一个运动模型，就可能出现轨迹偏移甚至发散，即跟踪失败的情况。这时自适应算法的优越性就显而易见了^[13]。我们之前介绍了 MMAE 算法，而在其基础上，衍生出来的交互式多模型（Interacting Multiple Model, IMM）算法则能够更好的获得想要的滤波结果。

本文所说的异构多模型自适应算法就是在上述算法的基础上，对跟踪强机动目标时有一个更好的估计效果和滤波性能提出了一个更高的要求。

4.1.1 多模型自适应算法原理

交互多模型（IMM）是我们之前提到的软件滤波技术。它的提出是出于对跟踪机动目标的需求应运而生的，与传统的线性 Kalman 滤波不同，这个算法使用了超过一个模型的模型集，结果输出也不是输出其中一个估计器的结果，而是每个结果的加权综合。

IMM 算法可以从后面的流程图看出，它的运算机制是多个基本滤波器进

行着多线并行运行，每个基本滤波器都是表示不同模式下的运动模型。不同的运动模型表示显示中不同的目标运动方式，所以基本滤波器在对同一个目标运动的估计结果都大有不同。因为 IMM 算法是假设时间为离散的，某个模型在现时刻估计结果是有效的条件下，因为已经预先获知了上一个时刻的输出，因此可以得到现阶段运算的初始参数，然后经过相应的滤波计算，即估计与修正，步骤。最终，系统的输出结果是由一系列的加权系数乘以每个基本估计器的输出结果得到的^[14]。因此，IMM 算法的输出是所有结果的融合输出，并不是我们所认为的依靠算法预先选择某一个模型并直接输出相应的结果。

根据 MMAE 和 IMM 算法的数学推导过程（在下一章会详细描述），我们可以看出两者的区别在于多模型交互自适应的算法将每个子模型的输入输出都通过马尔可夫状态转移矩阵进行了交互，不再只关注每个模型的权重，这样就可以避免 MMAE 中若模型权值降至零后无法继续进行迭代的问题，使其抗干扰能力得到了加强，但是因为计算过程加入了输入输出交互，导致计算量剧增，所以代码运行的速率会相应减慢^[15]。

通过阅读 Bitz Cecilia M^[16]，甘志梅^[17]的文章，对卡尔曼滤波的拓展算法在海冰模型预测和自动驾驶方面的应用有了进一步的了解；而在军事领域的多雷达组网扫描^[18]中，交互多模型算法的跟踪精度相比其他算法有着巨大的优势。由此可见，IMM 算法能够广泛应用于各种场景，因此，对这种算法深入研究并加以拓展是十分有必要的。

在本课题中，新引入了异构的要求，即在现实生活中，存在不同的模型结构，考虑异构这种条件，就可以将对象的不确定性范围扩充很大，实际上涵盖了被控对象出现各种机动的情况。通常我们使用匀速直线运动模型（Constant Velocity, CV），匀加速直线运动模型（Constant Acceleration, CA）和协调转弯模型（Coordinated Turn, CT）来描述目标的运动模式。例如车辆在行驶过程中，会出现匀速直线，变速直线，匀速率转弯，变速不规则曲线运动等场景^[19]，用不同模型来跟踪目标随机运动这种情况就被称为异构。

4.1.2 CV 和 CA 模型

目标做匀速或者匀变速运动的时候，常常使用 CV 模型和 CA 模型来描述其运动模式。

CV 模型的状态空间方程：

$$\begin{bmatrix} \dot{x} \\ \dot{\ddot{x}} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w(t) \quad (4-1)$$

CA 模型的状态空间方程:

$$\begin{bmatrix} \dot{x} \\ \dot{\ddot{x}} \\ \dot{\ddot{\ddot{x}}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w(t) \quad (4-2)$$

其中三个状态变量分别表示目标的位置, 速度, 加速度。 $w(t)$ 是均值为 0 的高斯白噪声。它们可以统一表示为标准形式的状态空间表达式。

假设扫描周期为 T , 则上式可以转化为离散形式的状态空间方程, 从而得到对应的相关参数。其中, CA 模型的相应参数为:

$$F = \begin{bmatrix} 1 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \quad (4-3)$$

$$Q = q \cdot \begin{bmatrix} \frac{T^5}{20} & \frac{T^4}{8} & \frac{T^3}{6} \\ \frac{T^4}{8} & \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^3}{6} & \frac{T^2}{2} & T \end{bmatrix} \quad (4-4)$$

这里 q 对应噪声的能量值, 如果是随机噪声较小, 则 q 相应为较小的值。

CV 模型是目标跟踪领域最基础的模型, 目标做近似匀速的运动都可以达到较好的跟踪效果^[19]。如果 CV 模型要描述目标机动状态, 则需要增大过程噪声协方差, 因为目标机动时自身的系统噪声会相应加大。

具体数学推导和表达将在下一节详细说明。

4.1.3 CT 模型

CT 模型即为联动转弯模型, 实际情况中存在两种两种不同的运动模式^[20]。第一种就是目标角速度及速度大小保持恒定, 速度方向始终垂直于运动轨迹的切线方向, 这种可以通过二维平面内圆周转弯模型来描述;

目标的状态向量由位置和速度组成，在 k 时刻的状态向量为： $X(k) = [x \ \dot{x} \ y \ \dot{y}]^T$ ，则状态转移矩阵，干扰矩阵和观测矩阵分别如下表示：

$$F(k) = \begin{bmatrix} 1 & \frac{\sin(\omega T)}{\omega} & 0 & -\frac{1 - \cos(\omega T)}{\omega} \\ 0 & \cos(\omega T) & 0 & -\sin(\omega T) \\ 0 & \frac{1 - \cos(\omega T)}{\omega} & 1 & \frac{\sin(\omega T)}{\omega} \\ 0 & \sin(\omega T) & 0 & \cos(\omega T) \end{bmatrix} \quad (4-5)$$

$$G(k) = \begin{bmatrix} \frac{T^2}{2} & 0 \\ T & 0 \\ 0 & \frac{T^2}{2} \\ 0 & T \end{bmatrix} \quad (4-6)$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (4-7)$$

当角速度大于 0 时，目标以角速度 ω 做匀速右转弯运动，小于 0 则是匀速左转弯运动，等于 0 时则是匀速直线运动。

上述模型为二阶圆周转弯模型，状态向量为四维。当加速度引进来之后，状态向量变成六维，如下： $X(k) = [x \ \dot{x} \ \ddot{x} \ y \ \dot{y} \ \ddot{y}]^T$ ，此时所用模型为三阶圆周转弯模型，其状态转移矩阵，噪声增益矩阵，观测矩阵分别为：

$$F(k) = \begin{bmatrix} A(k) & 0_{3 \times 3} \\ 0_{3 \times 3} & A(k) \end{bmatrix} \quad (4-8)$$

其中， $A(k)$ 为

$$A(k) = \begin{bmatrix} 1 & \frac{\sin(\omega T)}{\omega} & \frac{1 - \cos(\omega T)}{\omega^2} \\ 0 & \cos(\omega T) & \frac{\sin(\omega T)}{\omega} \\ 0 & -\omega \sin(\omega T) & \cos(\omega T) \end{bmatrix} \quad (4-9)$$

$$G(k) = \begin{bmatrix} \frac{T^6}{6} & 0 \\ \frac{T^2}{2} & 0 \\ T & 0 \\ 0 & \frac{T^6}{6} \\ 0 & \frac{T^2}{2} \\ 0 & T \end{bmatrix} \quad (4-10)$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (4-11)$$

此模型可以表示条件更宽松的不规则曲线运动。

4.2 滤波算法的数学推导

在这种算法下，在这里假定目标有 m 种状态，显然就是有对应的 m 个运动模型（即 m 个状态转移方程），设第 j 个模型表示的目标状态方程为

$$X_j(k+1) = \phi_j(k)X_j(k) + G_j(k)W_j(k) \quad (4-12)$$

观测方程为

$$Z(k) = H(k)X(k) + V(k) \quad (4-13)$$

方程中， $W_j(k)$ 为白噪声序列，它符合均值 0，协方差矩阵为 Q_j 的特性。各模型之间的转移的可能性取决于马尔科夫转移矩阵，目标从第 i 个运动模型转移到第 j 个运动模型的概率可以用元素 p_{ij} 表示，概率转移矩阵如下

$$P = \begin{bmatrix} p_{11} & \cdots & p_{1r} \\ \cdots & \cdots & \cdots \\ p_{r1} & \cdots & p_{rr} \end{bmatrix} \quad (4-14)$$

IMM 算法是以迭代方式进行的，每次迭代都要经过下面四个步骤。

1) 输入交互（第 j 个模型）

由目标的状态估计 $\hat{X}_i(k-1|k-1)$ 与上一步中每个滤波器的模型概率 $\mu_j(k-1)$ 得到混合估计 $\hat{X}_{0j}(k-1|k-1)$ 和混合协方差 $P_{0j}(k-1|k-1)$ ，把混合估计当作当前迭代的初始状态。具体的参数计算如下：

模型 j 的估计概率，就是所谓的归一化常数为

$$\bar{c}_j = \sum_{i=1}^r p_{ij} \mu_i(k-1) \quad (4-15)$$

模型 i 到模型 j 的混合概率为

$$\mu_{ij}(k-1|k-1) = \frac{\sum_{i=1}^r p_{ij} \mu_i(k-1)}{\bar{c}_j} \quad (4-16)$$

模型 j 的混合状态估计为

$$\hat{X}_{0j}(k-1|k-1) = \sum_{i=1}^r \hat{X}_i(k-1|k-1) \mu_{ij}(k-1|k-1) \quad (4-17)$$

模型 j 的混合协方差估计为

$$\begin{aligned} P_{0j}(k-1|k-1) = & \sum_{i=1}^r \mu_{ij}(k-1|k-1) P_i(k-1|k-1) \\ & + [\hat{X}_i(k-1|k-1) - \hat{X}_{0j}(k-1|k-1)] \\ & \cdot [\hat{X}_i(k-1|k-1) - \hat{X}_{0j}(k-1|k-1)]^T \end{aligned} \quad (4-18)$$

在上述方程式中， P_{ij} 模型为 i 到 j 的转移概率， $\mu_j(k-1)$ 是模型 j 在 k-1 时刻的概率。

2) 卡尔曼滤波

将上一步骤中产生的混合状态估计和混合协方差估计，作为输入量进行卡尔曼滤波，以此更新预测状态和滤波协方差。

预测状态为

$$\hat{X}_j(k|k-1) = \phi(k-1) \hat{X}_{0j}(k-1|k-1) \quad (4-19)$$

预测误差协方差为

$$P_j(k|k-1) = \phi_j P_{0j}(k|k-1) \phi_j^T + G_j Q_j G_j^T \quad (4-20)$$

卡尔曼增益为

$$K_j(k) = P_j(k|k-1) H^T [H P_j(k|k-1) H^T + R]^{-1} \quad (4-21)$$

更新滤波状态为

$$\hat{X}_j(k|k) = \hat{X}_j(k|k-1) + K_j(k) [Z(k) - H(k) \hat{X}_j(k|k-1)] \quad (4-22)$$

更新滤波协方差为

$$P_j(k|k) = [I - K_j(k) H(k)] P_j(k|k-1) \quad (4-23)$$

3) 模型概率更新

使用似然函数的方法来更新模型概率，模型 j 的似然函数为

$$\Lambda_j(k) = \frac{1}{(2\pi)^{n/2} |S_j(k)|^{1/2}} \exp \left\{ -\frac{1}{2} v_j^T S_j^{-1}(k) v_j \right\} \quad (4-24)$$

其中，

$$v_j(k) = Z(k) - H(k) \hat{X}_j(k|k-1) \quad (4-25)$$

$$S_j(k) = H(k) P_j(k|k-1) H(k)^T + R(k) \quad (4-26)$$

则模型 j 的概率为

$$\mu_j(k) = \frac{\Lambda_j(k) \bar{c}_j}{c} \quad (4-27)$$

其中，c 为归一化常数，且

$$c = \sum_{j=1}^r \Lambda_j(k) \bar{c}_j \quad (4-28)$$

4) 输出交互融合

由上面的数学公式推导我们很容易可以得出最终的系统结果是由一系列的基本估计结果加和而来，具体形式如下：

总状态估计为

$$\hat{X}(k|k) = \sum_{j=1}^r \hat{X}_j(k|k) \mu_j(k) \quad (4-29)$$

总协方差估计为

$$P(k|k) = \sum_{j=1}^r \mu_j(k) \left\{ P_j(k|k) + [\hat{X}_j(k|k) - \hat{X}(k|k)][\hat{X}_j(k|k) - \hat{X}(k|k)]^T \right\} \quad (4-30)$$

因此，由公式（4-29），（4-30）可以得知，系统的最终输出结果是由一系列加权系数乘以每个滤波结果所得到的。

下面将用一幅流程框图来简要说明 IMM 算法的迭代过程：

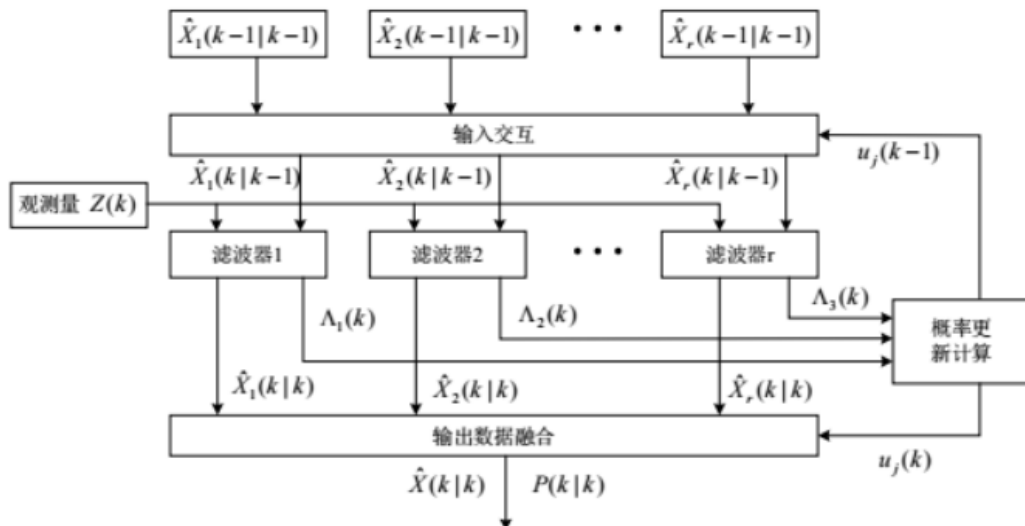


图 4-1 IMM 原理框图

选取滤波器的目标运动模型，需要考虑以下几个方面：

需要选取一定数量的 IMM 滤波器，包含较为精准的和较为粗劣的模型，IMM 滤波算法不仅可以描述目标的匀速运动状态，还可以描述目标的机动状态；

马尔科夫转移矩阵的具体形式也基本上决定了滤波器的滤波性能^[21]。通过查阅一定的资料可以得知,当转移概率超过一定阈值后,模型会倾向于保持原有的模式不发生改变,只有在小概率下会进行模型转换。在一般情况下,当马尔科夫状态转移概率符合现实中常用的状态转移规律的时候,IMM 才可以更稳健地跟踪目标运动

当对目标的机动规律把握很充分时,滤波器倾向于选择能比较精准描述对象运动的模型^[22],当无法预知目标的运动规律时,就应该选择更一般的模型,这意味着该模型应具有更强的鲁棒性。

4.3 目标跟踪仿真和比较

本课题对目标跟踪进行了三种仿真,分别为匀速运动的线性卡尔曼滤波仿真,蛇形曲线的线性卡尔曼/多模型自适应滤波仿真和 U 型机动多模型自适应滤波仿真。接下来将详细介绍各个仿真的具体场景和优化后的效果。

仿真一: 目标匀速运动的跟踪

我们给定场景为雷达的扫描周期为 1s,所有的采样次数为 100 次,这里设定的初始位置为 (-100m, 200m),然后初始速度为 (2m/s, 20m/s),即目标保持这个状态进行运动,假设过程噪声方差为 0.01,观测噪声方差为 1,现对此进行仿真。

分析:在已知初始位置和速度后,可以确定状态方程的有四个状态变量,分别为 X 轴位置, X 轴速度, Y 轴位置, Y 轴速度。因为观测条件受限,所以观测矩阵为 X 轴位置和 Y 轴位置的列向量。根据以前物理课程的知识可以轻松列出目标的运动方程,将其化成状态空间方程的标准矩阵形式,可以得到状态转移矩阵和观测矩阵。我们先画出目标在无任何干扰下的理论轨迹和只添加噪声的真实轨迹,然后我们给出预设的初始状态值和初始估计协方差。根据 Kalman 滤波的五个迭代方程迭代 100 次,我们可以得到滤波后的图像,我们将这些图像整合到一个图中,可以明确比较 Kalman 滤波对观测后的结果优化效果。

图 4-2 和图 4-3 就是相应的仿真结果:

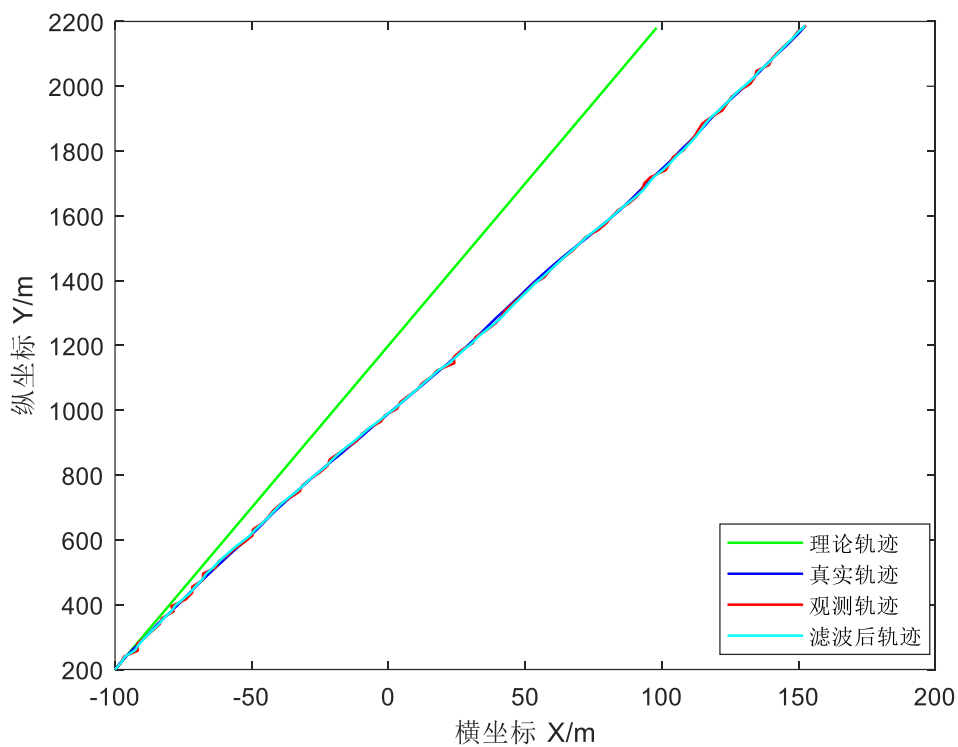


图 4-2 匀速运动跟踪仿真效果和比较

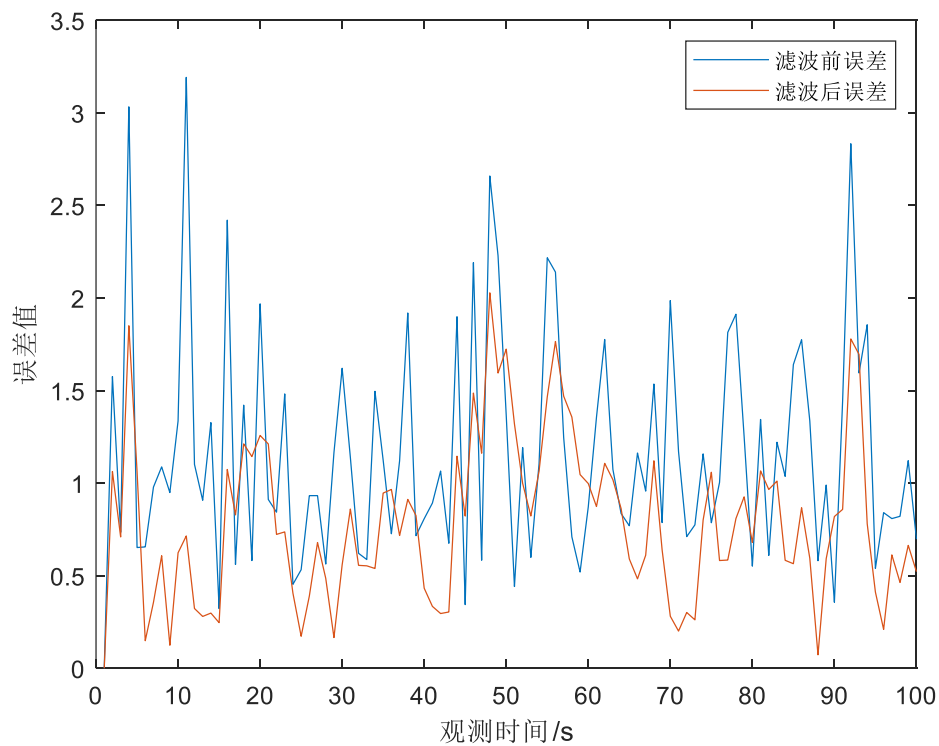


图 4-3 滤波前后的误差比较

比较：由基础的匀速运动的仿真可看出，线性卡尔曼滤波能够很好的整合估计值和观测值，从而得出预测值，和真实值极为接近，误差如上图也得到极大的改善。

仿真二：目标蛇形曲线的运动跟踪

在这次仿真实验中，我们模拟了实际汽车变道的问题，即预设的轨迹是一条蛇形曲线，具体场景如下：假设雷达的扫描周期为 2s，在 0-400s，目标做向下匀速直线运动；400-600s，目标产生向右的加速度 0.075m/s^2 ；600-610s；加速度消失，目标维持现有的速度保持匀速运动；610-660s，目标产生向左的加速度 3m/s^2 ；660-900s，加速度取消，目标此刻正好横向速度为 0，继续向下做匀速直线运动。

分析：根据题设所给条件，我们可以列出目标运动方程，然后在纸上画出目标的运动轨迹，即向下运动一段时间后向右换道随后继续向下匀速运动的场景。我们预设目标的初始位置为 (1000m, 8000m)，初始速度为 (0m/s, -12m/s)。过程噪声方差为 10，测量噪声方差为 900，根据相应的运动学方程，可以列出对应的状态空间方程矩阵标准形式，即可得到状态转移矩阵和观测矩阵。我们无法直接测量速度，所以只对位置进行观测。采样点总共 $900/2=450$ 个，我们分别设置两组变量记录真实轨迹和滤波后的轨迹，以便对比。在循环设置中，先设置初始估计协方差，然后对五个 Kalman 方程迭代 450 次，最后可以得到真实轨迹，观测轨迹和滤波后轨迹三者之间的比较。

图 4-4 和图 4-5 就是相应的仿真结果：

比较：将两个仿真结果进行对比，可以明确看出，仿真一，即只有线性卡尔曼滤波算法的仿真里，卡尔曼滤波后估计轨迹和观测轨迹贴近，没有很好的贴合实际轨迹，只是在幅度上有所衰减，抖动程度还是非常频繁的。在仿真二中，我们应用了异构多模型算法，将前面所说的模型构成一个模型集进行自适应估计，在图中很显然抑制了噪声的扰动，只在真实轨迹附近有微小的波动，起到了很好的跟踪效果。而在在实际场景中，仿真一这种现象不能满足我们对目标跟踪的要求，所以仿真二我们引入了异构多模型算法，能够很好的解决这个问题。

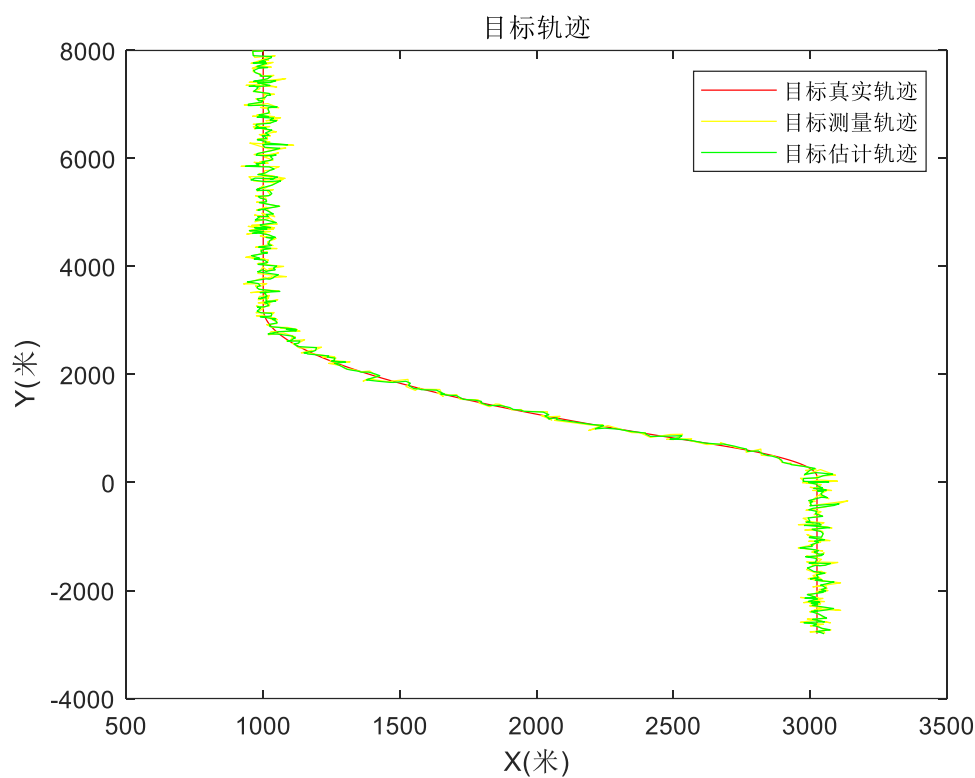


图 4-4 蛇形曲线仿真一

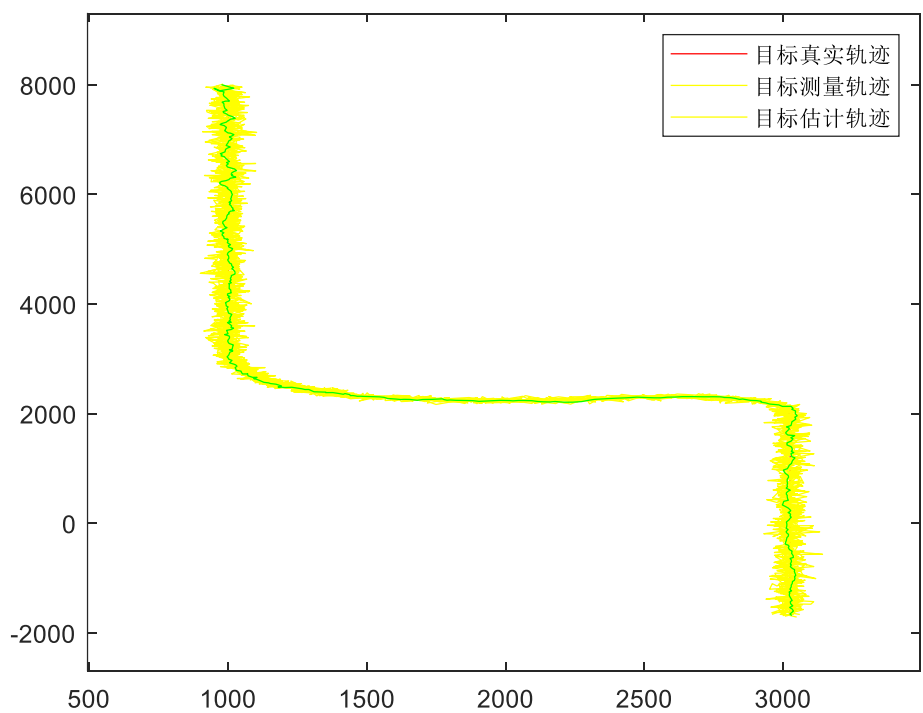


图 4-5 蛇形曲线仿真二

仿真三：目标做 U 型掉头运动的跟踪

这次的仿真实验是由上一个蛇形曲线（变道）的实际场景衍生而来的，模拟了机动车匀速前进然后向右转弯实现掉头的情况。为了进一步对比传统线性卡尔曼滤波和本次 IMM 算法的优劣，我们这次的雷达精度和扫描间隔都有所下降，扫描周期为 2s，总采样点数为 $N=900/T=450$ 次，测量噪声的方差为 900。现预设目标的初始位置为 (2000m, 0m)，从 2-400s，目标做向上的匀速运动，速度为 (0, 10m/s)，在 402-600s，产生加速度为 (0.075m/s², 0.075m/s²)，在 602-610s，加速度消失，保持匀速运动，在 612-660s，加速度为 (-0.3m/s², -0.3m/s²)，最后在 662-900s 时，加速度消失，保持匀速运动直到观测截止。

分析：在了解到仿真的场景条件后，我们可以在纸上先画出大致的轨迹，从而估计仿真出来的结果。通过以前学习的物理知识，可以建立目标做机动运动的方程，并化成标准状态空间矩阵形式，就可以得到状态转移矩阵和观测矩阵。经过计算，第一次转弯结束后 Y 轴速度降到 0，第二次转弯结束后，X 轴速度降到 0，符合实际场景 U 型转弯的条件。然后根据 IMM 算法中四个迭代步骤中设置初始参数，在这里设置三个模型，分别为非机动模型和两个机动模型，根据常规设置了马尔可夫状态转移矩阵。为了突出对比效果，我们在前 10 次迭代设置为线性卡尔曼滤波，后面转换为多模型自适应滤波，以便在仿真图像中观察两者的优劣。在这里预设初始模型概率为 [0.4, 0.3, 0.3]，在经过多次仿真后，我们将仿真次数定在 20 次，达到较好的对比结果。最后，我们绘制了 X, Y 轴的误差均值，误差方差和轨迹对比图像，可以看出 IMM 算法在跟踪强机动目标的场景下精度和速率远好于传统的线性卡尔曼滤波。

图 4-6，图 4-7 和图 4-8 就是具体的仿真结果：

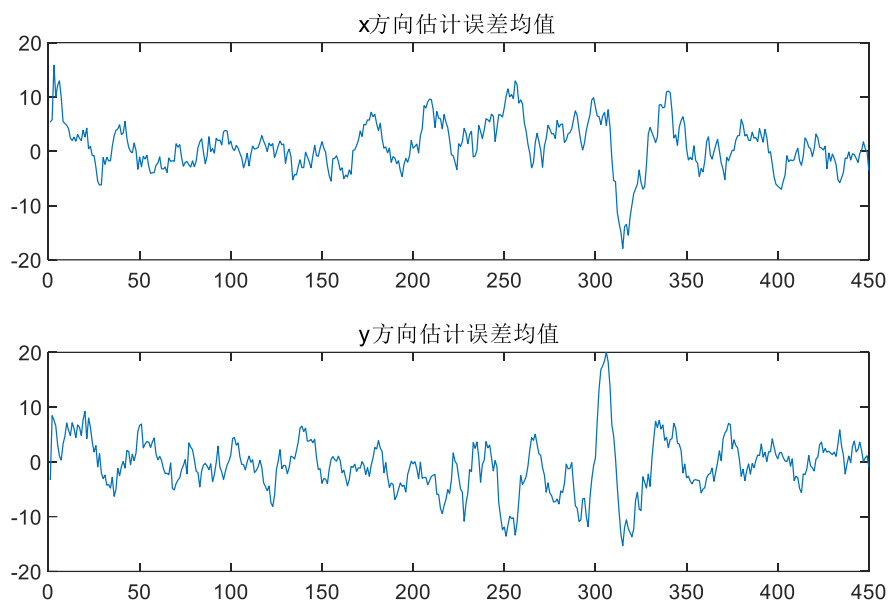


图 4-6 X、Y 轴方向的估计误差均值

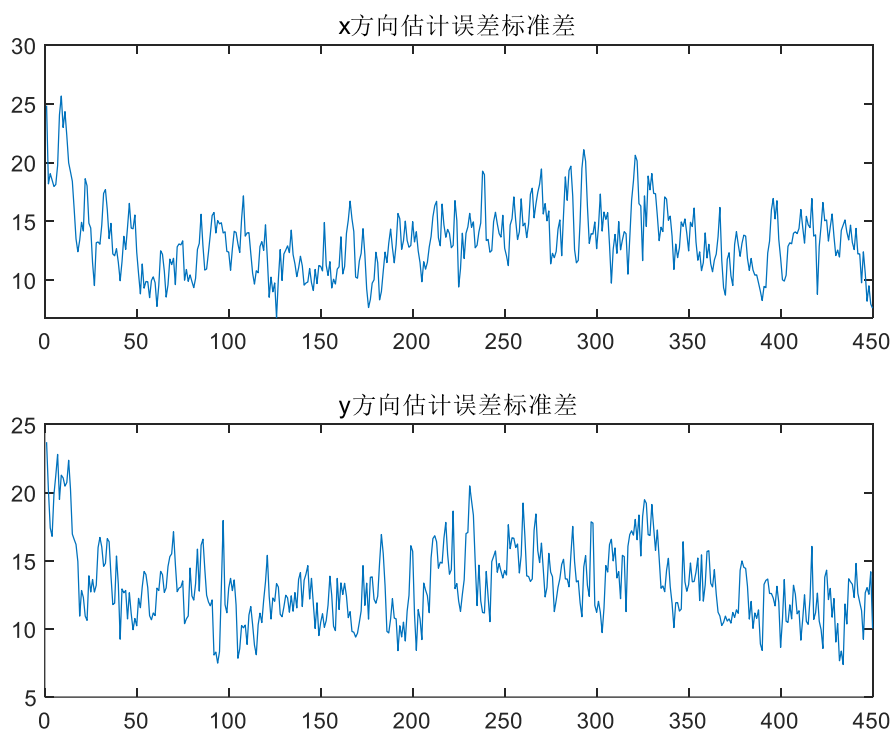


图 4-7 X、Y 轴方向估计误差标准差

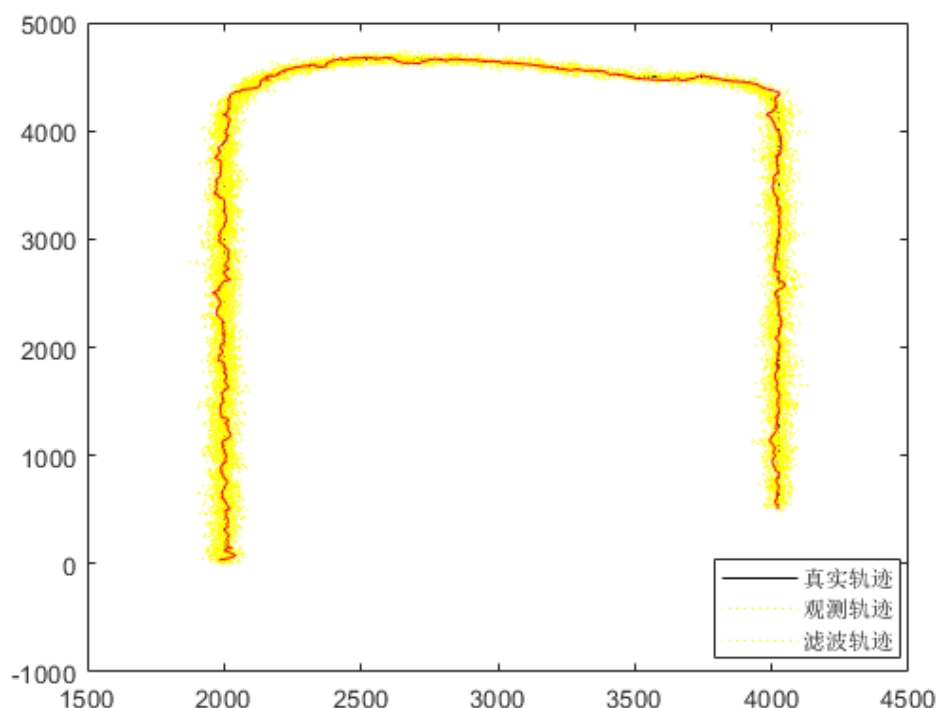


图 4-8 U 型曲线运动跟踪轨迹

比较：由仿真图的对比，可以看出，在前 10 次仿真内的结果，跟踪效果不如后期，抖动的幅度比较大。是因为我们故意在前 10 次仿真点仅使用线性卡尔曼滤波，等结束后才一直使用异构多模型的算法。在加速转弯的时候，异构自适应多模型算法的跟踪效果极佳，能很好地抑制噪声并贴近理论的运动轨迹。如果按照原有的线性 Kalman 滤波方法，在观测噪声和过程噪声稍大的情况下，在转弯加速的过程中就有可能出现滤波发散的现象，从而使跟踪失效，由此可见多模型自适应算法的优越之处。

综合以上的所有仿真，我们对比不难发现仅在复杂运动的情况下，原有的 Kalman 滤波器因为缺少足够多的模型会导致跟踪失败，而本文的异构多模型自适应方法，已经将常见的运动模型都放入建立好的模型集中，当目标的运动模式突然发生变化，自适应算法可以实时地调整每个模型的权重，保证跟踪的精准度。由此可见，在新的场景下需要对原有算法做出相应的改进以符合我们对应的需求。

5 非技术因素的考虑与实现

本文的性质为设计类论文，因此有必要在第五章单独叙述一下本课题所需要的非技术类因素的考虑和实现。

本文的主要内容为异构多模型自适应卡尔曼滤波的研究和仿真。从本科毕业论文的角度，我们只需要考虑理论知识的获得和仿真的实现；从实际工业角度，需要更多的硬件辅予以支持，同时需要配备备用的算法和软件，以备现场的突发情况。下面将从两方面详细说明在高校和实际工业场景下的非技术因素的一些考虑和实现过程。

在学校实现本课题研究内容需要的物质需求比较少，由于不需要一些上位机等专用性硬件，只需要一台电脑，配合 MATLAB 实现仿真的实现和论文的撰写。这学期在研究课题内容和撰写论文时，通常需要去图书馆参考相关的专业理论书，通过网络去寻找一些有关的资料，在仿真阶段，则会借助 MATLAB 进行仿真，最后通过 word 写出论文。从成本上考虑，一台满足性能要求的新电脑的价格在 10000 元以内，查阅资料的成本忽略不计，MATLAB 在个人非商业使用时会通过一定的手段免费获得，所以总体的成本会控制在一万人名币以内，在这方面会表现得比较好。

具体在算法方面，将对应用背景，意义，地位，发展趋势一一阐述。卡尔曼滤波算法在实际应用方面可以追溯到阿波罗登月的具体使用。当时为了估计飞船的绕月轨迹和返航轨迹，需要计算机进行大量的计算来进行模拟。相比过去的滤波估计器，卡尔曼滤波器以递推的方式，每次运算只需要上一次扫描时刻的数据，所以对硬件的存储量不高。与此同时，因为是人类直观感知的时域量，输出的数据不需要额外处理，就可以直接应用，相比过去的滤波器，就因为需要拉氏变换进行频域分析，就省去了很多计算步骤。从登月开始，卡尔曼滤波的使用场景在生活中就被大大拓宽，凡是带有随机误差和不确定性的实际问题，都可以运用卡尔曼的理论对需要的状态量进行精确的估计，经过很多年的工程验证，它的适用性已得到几乎所有人的认可。由此可见，对卡尔曼滤波深入研究和拓展都是十分有意义的，都可能对实际现实生活的某一具体问题的解决有着巨大的帮助。在本文中，异构多模型算法对跟踪强机动目标的问题就比传统线性滤波有着巨大的优势。

卡尔曼滤波理论在工业算法领域，就像“硅”在电子元器件在地位一样，成为了几乎不可能错过的理论基础。从 1960 年它被提出后，后人一直根据实际问题对其不断改进来克服原来所存在的缺陷。现在我们所熟悉的拓展卡尔曼滤波，无迹卡尔曼滤波和当前统计模型滤波等，都已以传统线性卡尔曼滤

波为基础的。现在，它已被金融界采用，进行股票分析与预测。相信在不久的未来，它不仅能在工业领域，甚至可能在农业，新兴产业方面有着更多的作为。

在现实工程中，比如在最近流行的无人驾驶的领域，就需要用汽车内的芯片进行实时的计算来保证汽车对周围事物保持最新的感知并做出决策。在现在我们自动驾驶的传感器有使用激光雷达，这一传感器并不少见，在上一代的 iPhone11 Pro 上就搭载了这一个先进的传感器，它能够采集到拍摄空间的景深信息，从而构建立体的三维模型。在自动驾驶上，激光雷达可以获得车子周围的物体远近和轮廓数据，但因为空气中有沙石，雾气等不可控的干扰物，接收到的信号会夹杂一定的随机噪声，这时我们就需要卡尔曼滤波来估计更精确的信息。在这里，成本就需要经过严谨的考虑，包括传感器的成本，电控系统的成本，集成芯片的成本和中央显示器的选取。因为没有参与过相关过程的实习，所以具体的成本控制策略在此无法给出。

最后，考虑到人为操作方面，个人在学习相关代码并调试的时候会出现很多的问题，经过不断的改进和咨询才最终实现想要的效果。在实际工业场景下，我们对于代码的要求是注释尽可能地写全，代码的规范必须遵守以备未来的功能需要改进时可以迅速地增加或删减，即代码的可读性要加强。然而，计算机上的模拟永远无法替代实际发生的状况，所以在面向消费者推出迭代版本之前，需要大量的实地检测来发现 bug 或潜在的隐患。

6 结 论

本文核心内容是对工程中常见的滤波领域进行了浅显的探究。目前，目标机动的场景愈发复杂，我们通过经典的 Kalman 滤波理论，得出一种跟踪目标运动的方法。对于匀速运动的系统，经典线性卡尔曼滤波能够很好地利用估计值和观测值给出较为精确的预测^[23]，但现实中，由于噪声的影响和目标直线加速或变速转弯的情况都对经典的滤波方法提出了挑战。根据我们仿真的结果，有的甚至出现了滤波发散的情况，所以这时候我们要引进新的滤波方法，使得结果更加准确迅速。

在这里我们采用了多模型自适应的方法，同时创新性地加入了异构的要求，就是同一场景下出现的不同状态向量，即不同维数下的模型情况。IMM 算法能够在每次迭代时自动计算各种模型的权重和概率更新，最后对结果输出融合，计算量比原来的线性 Kalman 滤波更大，但跟踪的效果也更好，更贴近目标运动的真实轨迹。因此，在实际生活中，如温度测量等一维情况我们会使用经典卡尔曼滤波；在跟踪目标运动，其中包含多个状态变量和系统输入时，往往使用多模型自适应算法可以在准确性和快速性之间做到极佳的平衡。

当然，在仿真的过程中，如果过程噪声和测量噪声过大，或者两者为有色噪声的情况下，滤波算法也会失效，需要新的算法来解决这种问题；同时，预设的马尔可夫状态转移矩阵也会对滤波的效果有较大影响，因此如果可以对其加入自适应算法^[24]，准确度会更高，计算量当然也会更大，很遗憾本文目前对这个问题暂时无法解决。抛开软件算法方面的缺陷，实际场景我们可能会面对单一高精度的观测器和多个低精度观测器的选取，此时又需要去开发新的算法对多观测器的数据进行融合输入，这些新的问题都需要不断去解决，所以在滤波领域可深度挖掘的内容还有很多。

在实际的现实应用中，我们所做的计算机仿真只能做一定的参考，因为仿真永远无法替代现实发生的状况，这些偶发性的事故在目前的计算机模拟中都无法建模，所以在推给面向消费者的产品之前都要进行大量的实地测试，来消除潜在的隐患和 bug。

自从传统线性 Kalman 滤波理论在 1960 年被提出后，许多现实场景都促使更多的衍生卡尔曼滤波算法的出现，用以解决不同具体条件下滤波或跟踪的问题，如非线性系统下的 EKF 和 UKF，还有在航天导航等领域都有相应的应用^[22]。本文做的工作仅仅是对比传统滤波方法和多模型自适应的方法在目标跟踪应用时的优劣，其中也使用了一些新的仿真场景，其结果也表明新

算法在这种情况下的优越之处。在代码实现中同样存在一些小瑕疵因为技术因素暂时无法解决，因此还需要在未来的工作中持续对其改进，同时要多多参与相关的实际工程项目，将理论和实际相结合才能使成果更完美。在完成毕设的过程中，尤其在仿真时，因为加入了随机噪声，每次的结果也会有所不同，这在现实中也是一个需要解决的难题。希望以后能考虑更多实际情况的不确定因素，将其加入到所建立的模型，使用更好的算法完成目标跟踪的工作。如果能够对具体环境进行设计，多模型自适应算法也可以应用于其他领域用于对某目标进行估计，这种方法的未来远不止于此。

参考文献

- [1] Kalman R. E.. A New Approach to Linear Filtering and Prediction Problems[J]. Journal of Basic Engineering,1960,82(1).
- [2] 付梦印, 邓志红, 闫莉萍. Kalman 滤波理论及其在导航系统中的应用 [M] . 北京: 科学出版社, 2003.
- [3] 张长春,黄英,杨刚.卡尔曼滤波在跟踪运动目标中的应用及仿真[J].现代电子技术,2009,32(20):54-56.
- [4] H.Feng. 什么是卡尔曼滤波 [J/OL]. 数学控制 Club, 2017.
- [5] 王雪瑞,何永强.基于卡尔曼滤波的目标跟踪系统的建模与仿真[J].微电子学与计算机,2014,31(08):123-126+131.
- [6] Magill D. Optimal adaptive estimation of sampled stochastic processes[J] . IEEE Transaction of Automatic Control, 1965, 10: 435-439.
- [7] 王向华,覃征,杨新宇,杨慧杰.基于多次卡尔曼滤波的目标自适应跟踪算法与仿真分析[J].系统仿真学报,2008,20(23):6458-6460+6465.
- [8] Lin Chen,Bowen Jiang,Yuqi Liu,Jiangyu Chen. Application of adaptive EKF in real-time orbit determination[J]. Journal of the Brazilian Society of Mechanical Sciences and Engineering,2021,43(4).
- [9] 孙君道.基于 Matlab 的 Kalman 滤波建模及仿真[J].科技信息(科学教研),2008(25):53-54.
- [10] Lainiotis D.G. Partitioning: A Unifying Framework for Adaptive System, I: Estimation. In: Proc IEEE:1976; 1976: 1126-1143.
- [11] 石章松,刘忠.单站纯方位目标多模型卡尔曼滤波跟踪算法的研究[J].系统仿真学报,2006(07):1802-1805+1814.
- [12] 姜楷娜.基于卡尔曼滤波的目标轨迹跟踪仿真研究[J].中国科技信息,2017(17):105-106+13.
- [13] Fraser Cory T.,Ulrich Steve. Adaptive extended Kalman filtering strategies for spacecraft formation relative navigation[J]. Acta Astronautica,2021,178.
- [14] 石章松,王树宗,刘忠.被动目标自适应跟踪算法仿真研究[J].计算机仿真,2004(06):78-82.
- [15] Antonio J. Rodriguez , Emilio Sanjurjo , Roland Pastorino , Miguel Ángel Naya. State, parameter and input observers based on multibody models and Kalman filters for vehicle dynamics [J]. Mechanical Systems and Signal

- Processing, 2021.155(3):1-17.
- [16] Zhang Yong Fei,Bitz Cecilia M.,Anderson Jeffrey L.,Collins Nancy S.,Hoar Timothy J.,Raeder Kevin D.,Blanchard Wrigglesworth Edward. Estimating parameters in a sea ice model using an ensemble Kalman filter[J]. The Cryosphere,2021,15(3).
- [17] 甘志梅,王春香,杨明.基于激光雷达的车辆跟踪与识别方法[J].上海交通大学学报,2009,43(06):923-926.
- [18] 赵温波,丁海龙.基于交互多模型虚拟观测卡尔曼滤波的多雷达组网机动目标跟踪算法[J].信息与控制,2015,44(01):15-20.
- [19] 王华松,李鹏,张家叶子,赵鑫.基于卡尔曼滤波的序贯融合对机动目标跟踪算法与仿真[J].工业控制计算机,2020,33(07):122-124+147.
- [20] 李春辉,马健,杨永建,肖冰松,邓有为,盛涛.基于修正的自适应平方根容积卡尔曼滤波算法[J/OL].系统工程与电子技术:1-9[2021-04-02].<http://kns.cnki.net/kcms/detail/11.2422.TN.20210207.1353.004.html>.
- [21] 岳崇伦,曾苑,郭云开.提高卫星导航精度的卡尔曼滤波算法应用研究[J].测绘工程,2021,30(02):60-64+71.
- [22] 杜世勇,李枢,唐川.一种基于自适应卡尔曼滤波的 ADS-B 点迹跟踪处理方法[J].电子世界,2021(05):110-112.
- [23] Pan Cheng,Gao Jingxiang,Li Zengke,Qian Nijia,Li Fangchao. Multiple fading factors-based strong tracking variational Bayesian adaptive Kalman filter[J]. Measurement,2021,176.
- [24] 刘国情,袁俊泉,马晓岩,陈阿磊,王力宝.一种马尔可夫矩阵自适应的 IMM-CKF 算法[J].雷达科学与技术,2017,15(03):241-246.

附录A 程序清单

2-1 程序源码

```
%卡尔曼滤波在一维温度测量中的应用

clear;
clc;
N=120; %采样点的个数
T=25; %房间温度的理论值, 假设不受干扰的话
X_expect=T*ones(1,N); %每个采样点没有干扰的话是 25 度,1 行 N 列的矩阵
X=zeros(1,N); %保存各个采样点的真实温度, 为 1xN 的零向量
X_est=zeros(1,N); %各个时刻的估计值/先验值
X_pre=0;%各个时刻的预测值, 也就是根据前面已经有的状态, 结合状态转移
    矩阵得到的预测值/后验值
Z=zeros(1,N); %各个时刻的测量值, 也就是传感器给出的
P_pre=0; %协方差矩阵有两类, 一个是预测协方差, 一个是估计协方差, 这
    里为预测协方差
P_est=zeros(1,N); %维数 1,估计协方差

%初始化
X(1)=25.1; %初始的真实温度
P_est(1)=0.01; %初始值的协方差
Z(1)=24.9; %初始的测量值
X_est(1)=Z(1); %初始的估计值, 直接用了初始测量值

%噪声
Q=0.01;%白噪声的方差,过程噪声, 加入到了状态转移的过程, 均值为 0, 方
    差为 Q
W=sqrt(Q)*randn(1,N);
R=0.25; %测量噪声的方差, 均值为 0
V=sqrt(R)*randn(1,N);
```

```

%系统矩阵
A=1; %状态转移矩阵, 理论方程应该是各个时刻温度恒定的
G=1; %噪声驱动矩阵
H=1; %观测矩阵, 即从状态空间到观测空间的映射
E=eye(1); %系统 1 维的单位阵

%模拟测量过程并滤波
for k=2:N
    X(k)=A*X(k-1)+G*W(k-1); %真实温度要考虑过程噪声
    Z(k)=H*X(k)+V(k); %测量

    %预测状态
    X_pre=A*X_est(k-1); %依靠上一次的最优估计值（估计值）推导这次的预测值
    P_pre=A*P_est(k-1)*A'+Q; %由上一时态的估计方差推导现在的预测方差, A'为 A 的转置

    K=P_pre/(H*P_pre*H'+R); %卡尔曼增益求解
    e=Z(k)-H*X_pre; %观测与预测的残差
    X_est(k)=X_pre+e*K; %由这次预测值得到最优估计值
    P_est(k)=(E-K*H)*P_pre; %更新估计方差
end

%误差计算, 第二张图的内容, 初始化:
er_meas=zeros(1,N);
er_kalm=zeros(1,N);

for k=1:N
    er_meas(k)=abs(Z(k)-X(k)); %测量值和实际值的误差
    er_kalm(k)=abs(X_est(k)-X(k)); %估计值和实际值的误差
end

```

```

%数据可视化
t=1:N;      %同时画出('无干扰理论值','直接测量值','实际值','kalman 估计值')
figure
plot(t,X_expect,'-b',t,Z,'-r',t,X,'-ko',t,X_est,'-g*');
legend('无干扰理论值','测量值','实际值','卡尔曼估计值');
xlabel('采样时间');
ylabel('温度值/°C');

figure
plot(t,er_meas,'-b.',t,er_kalm,'-k*');
xlabel('采样时间');
ylabel('温度偏差');
legend('测量偏差','滤波值偏差');

```

4-1 程序源码

```

function untitled4
%匀速运动
clc;clear;
T=1;                                %雷达扫描周期
N=100/T;                            %总的采样次数
X=zeros(4,N);                      %目标真实位置、速度
X(:,1)=[-100,2,200,20];           %目标初始位置、速度,X 轴位置, X 轴
速度, Y 轴位置, Y 轴速度
S(:,1)=[-100,2,200,20];           %目标理论轨迹初始值
Z=zeros(2,N);                      %传感器对位置的观测
Z(:,1)=[X(1,1),X(3,1)];           %观测初始化
delta_w=1e-2; %如果增大这个参数, 目标真实轨迹就是曲线了, 为 0.01
Q=delta_w*diag([0.5,1,0.5,1]);    %过程噪声均值矩阵
R=eye(2);                          %观测噪声均值矩阵
A=[1,T,0,0;
    0,1,0,0;

```

```

    0,0,1,T;
    0,0,0,1];           %状态转移矩阵
H=[1,0,0,0;
    0,0,1,0];           %观测矩阵
X_est=zeros(4,0);       %定义估计值
for i=2:N
    S(:,i)=A*S(:,i-1);%目标理论轨迹
    X(:,i)=A*X(:,i-1)+sqrtm(Q)*randn(4,1);%目标真实轨迹,sqrtm 为矩阵平方
    根
    Z(:,i)=H*X(:,i)+sqrtm(R)*randn(2,1);%对目标的观测
end

% Kalman 滤波
X_pre=zeros(4,N);
X_pre(:,1)=X(:,1);%卡尔曼滤波状态初始化
%M(1,:)=X_pre(:,1);
P_pre=100e-2*eye(4);% 协方差阵初始化

for i=2:N
    X_est=A*X_pre(:,i-1);           %估计协方差
    %M(i,:)=X_est;
    P_est=A*P_pre*A'+Q;             %预测误差协方差
    K=P_est*H'*inv(H*P_est*H'+R);   %卡尔曼增益, inv 为求逆
    X_pre(:,i)=X_est+K*(Z(:,i)-H*X_est); %状态更新
    P_pre=(eye(4)-K*H)*P_est;        %滤波预测误差协方差更新
end

% 误差分析
for i=1:N

    Err_Observation(i)=RMS(X(:,i),Z(:,i));%观测值和真实值的误差
    Err_KalmanFilter(i)=RMS(X(:,i),X_pre(:,i));%估计值和真实值的误差
end

```

```
figure
hold on;box on;
plot(S(1,:),S(3,:), 'g','LineWidth',1);%理论轨迹
plot(X(1,:),X(3,:), 'b','LineWidth',1);%真实轨迹
plot(Z(1,:),Z(2,:), 'r','LineWidth',1);%观测轨迹
plot(X_pre(1,:),X_pre(3,:), 'c','LineWidth',1);%卡尔曼滤波轨迹
legend('理论轨迹','真实轨迹','观测轨迹','滤波后轨迹');
xlabel('横坐标 X/m');
ylabel('纵坐标 Y/m');
```

```
figure
hold on;box on;
plot(Err_Observation);
plot(Err_KalmanFilter);
legend('滤波前误差','滤波后误差');
xlabel('观测时间/s');
ylabel('误差值');
```

```
% 计算欧氏距离函数
function dist=RMS(X1,X2)
if length(X2)<=2
    dist=sqrt((X1(1)-X2(1))^2+(X1(3)-X2(2))^2);
else
    dist=sqrt((X1(1)-X2(1))^2+(X1(3)-X2(3))^2);
end
```

4-2 程序源码

```
%二维蛇形运动跟踪
%时间间隔初始化
T1 = 400;
```

```

T2 = 600;
T3 = 610;
T4 = 660;
T5 = 900;
T = 2;%观测间隔
X_init = [1000 0 0 8000 -12]';%设置初始位置，位置 X，X 轴速度，X 轴加速度，位置 Y，Y 轴速度
b = 1000;
P = diag([b b b b b]); %初始协方差

meas_sigma = 900; %测量噪声方差
meas_noise = diag([meas_sigma meas_sigma]);

pro_sigma = 10;%过程噪声方差
pro_noise = diag([pro_sigma pro_sigma/T pro_sigma/T.^2 pro_sigma pro_sigma/T]);

%状态转移矩阵
A = [1 T 0.5*T*T 0 0;
     0 1 T 0 0;
     0 0 1 0 0;
     0 0 0 1 T;
     0 0 0 0 1];

H = [1 0 0 0 0;
     0 0 0 1 0];%观测矩阵

X_true_trac = X_init;%真实轨迹
Z_meas_trac = [1000;8000];%测量的轨迹
X_est_trac = [1000;8000];%估计的轨迹
X_init_copy = X_init;%copy 为中间变量，便于赋值

%卡尔曼滤波，构建运动模型
for i = 1:T5/2 %判断加速度并赋值
    if i>=T1/2 && i<=T2/2

```

```

        a      = 0.075;
        X_init(3,i) = a;
        X_init_copy(3,i) =a;
    end
    if i>=T2/2 &&i<=T3/2
        a      = 0;
        X_init(3,i) = a;
        X_init_copy(3,i) =a;

    end
    if i>=T3/2 &&i<T4/2
        a      = -0.3;
        X_init(3,i) = a;
        X_init_copy(3,i) =a;

    end
    if i>=T4/2
        a      = 0;
        X_init(3,i) = a;
        X_init_copy(3,i) =a;

    end

    %产生真实的轨迹
    X_true      = A * X_init(:,i);
    X_init(:,i+1)      = X_true;  %迭代
    X_true_trac(:,i+1) = X_true; %画图用的
    Z_meas_trac(:,i+1) = [X_true(1)+wgn(1,1,30);X_true(4)+wgn(1,1,30)];%在
    真实值上增加 1X1 的高斯噪声

    %卡尔曼滤波
    %一步预测
    X_est =  A * X_init_copy(:,i);
    P_est =  A * P * A' + pro_noise;

```

```

%滤波更新
K = P_est * H' * inv(H * P_est * H' + meas_noise);
X_pre = X_est + K * (Z_meas_trac(:,i+1) - H * X_est);
P_pre = (eye(5) - K * H) * P_est;

X_init_copy(:,i+1) = X_pre;
P = P_pre;

X_est_trac(:,i+1) = [X_pre(1); X_pre(4)];%卡尔曼滤波后估计值

end

plot(X_true_trac(1,:),X_true_trac(4:), 'g', Z_meas_trac(1,:), Z_meas_trac(2,:), 'r', X_
est_trac(1,:), X_est_trac(2,:), 'y')
title('目标轨迹');
xlabel('X(米)');
ylabel('Y(米)');
legend('目标真实轨迹','目标测量轨迹','目标估计轨迹');

```

4-3 程序源码

% 自适应多模型 Kalman 滤波的目标跟踪仿真

```

function Untitled6
clear;
T=2;%雷达扫描周期，即采样周期
M=20;%滤波次数
N=900/T;%总采样点数
N1=400/T;%第一转弯处采样起点
N2=600/T;%第一匀速处采样起点
N3=610/T;%第二转弯处采样起点
N4=660/T;%第二匀速处采样起点

```


Delta=30;%测量噪声标准差

%对真实的轨迹和观测轨迹数据的初始化，以向右向上为正

Rx=zeros(N,1);

Ry=zeros(N,1);

Zx=zeros(N,M);

Zy=zeros(N,M);

% 1-沿 y 轴匀速直线

t=2:T:400;

x0=2000+0*t';

y0=0+10*t';%初始速度 Vy=10，位置 (2000, 0)

% 2-慢转弯

t=402:T:600;

x1=x0(N1)+0.075*((t'-400).^2)/2;%X 轴正向加速度 0.075

y1=y0(N1)+10*(t'-400)-0.075*((t'-400).^2)/2;%Y 轴反向加速度 0.075

% 3-匀速

t=602:T:610;

vx=0.075*(600-400);

x2=x1(N2-N1)+vx*(t'-600);

y2=y1(N2-N1)+0*t';%此时 Y 轴速度正好为 0，因此 Y 位置不变

% 4-快转弯

t=612:T:660;

x3=x2(N3-N2)+vx*(t'-610)-0.3*((t'-610).^2)/2;%X 轴反向加速度 0.3

y3=y2(N3-N2)-0.3*((t'-610).^2)/2;%Y 轴反向加速度 0.3

% 5-匀速直线

t=662:T:900;

vy=-0.3*(660-610);

x4=x3(N4-N3)+0*t';%此时 X 轴速度正好为 0

y4=y3(N4-N3)+vy*(t'-660);

% 最终将所有轨迹整合成为一个列向量，即真实轨迹，Rx 为 Real-x,Ry 为 Real-y

Rx=[x0;x1;x2;x3;x4];

Ry=[y0;y1;y2;y3;y4];

```

% 对每次蒙特卡洛仿真的滤波估计位置的初始化
Mt_Est_Px=zeros(M,N);
Mt_Est_Py=zeros(M,N);
% 产生观测数据, 要仿真 M 次, 必须有 M 次
nx=randn(N,M)*Delta;%产生 X 轴和 Y 轴观测噪声
ny=randn(N,M)*Delta;
Zx=Rx*ones(1,M)+nx;%真实值的基础上叠加噪声, 即构成仿真的观测值
Zy=Ry*ones(1,M)+ny;

for m=1:M
    %滤波初始化
    Mt_Est_Px(m,1)=Zx(1,m);%初始数据
    Mt_Est_Py(m,1)=Zx(2,m);
    xn(1)=Zx(1,m);%滤波初值
    xn(2)=Zx(2,m);
    yn(1)=Zy(1,m);
    yn(2)=Zy(2,m);
    %非机动模型参数
    phi=[1,T,0,0;0,1,0,0;0,0,1,T;0,0,0,1];%状态转移矩阵, 没有加速度
    h=[1,0,0,0;0,0,1,0];%观测矩阵
    g=[T/2,0;1,0;0,T/2,0,1];%噪声增益矩阵
    q=[Delta^2,0;0,Delta^2];%过程噪声协方差
    vx=(Zx(2)-Zx(1,m))/2;%非机动模型速度不变, 所以每次滤波按位移比时
    间设定
    vy=(Zy(2)-Zy(1,m))/2;
    %初始状态估计
    x_est=[Zx(1,m);vx;Zy(1,m);vy];
    p_est=[Delta^2,Delta^2/T,0,0;Delta^2/T,2*Delta^2/(T^2),0,0;
           0,0,Delta^2,Delta^2/T;0,0,Delta^2/T,2*Delta^2/(T^2)];
    Mt_Est_Px(m,2)=x_est(1);%把初始估计值赋值给 M 仿真
    Mt_Est_Py(m,2)=x_est(3);
    %滤波开始
    for r=1:N
        z=[Zx(r,m);Zy(r,m)];

```

```

if r<10%前 10 次不做 IMM，只是单纯的线性卡尔曼滤波
    x_pre=phi*x_est;%预测
    p_pre=phi*p_est*phi';%预测误差协方差
    k=p_pre*h'*inv(h*p_pre*h'+q);%卡尔曼增益
    x_est=x_pre+k*(z-h*x_pre);%滤波
    p_est=(eye(4)-k*h)*p_pre;%滤波协方差
    xn(r)=x_est(1);%记录采样点滤波数据
    yn(r)=x_est(3);
    Mt_Est_Px(m,r)=x_est(1);%记录第 m 次仿真滤波估计数据
    Mt_Est_Py(m,r)=x_est(3);
else
    if r==10
        X_est=[x_est;0;0];%扩充维数，引入加速度
        P_est=p_est;
        P_est(6,6)=0;
        for i=1:3
            Xn_est{i,1}=X_est;
            Pn_est{i,1}=P_est;
        end
        u=[0.4,0.3,0.3];%模型概率初始化
    end
    %调用 IMM 算法
    [X_est,P_est,Xn_est,Pn_est,u]=IMM(Xn_est,Pn_est,T,z,Delta,u);
    xn(r)=X_est(1);
    yn(r)=X_est(3);
    Mt_Est_Px(m,r)=X_est(1);
    Mt_Est_Py(m,r)=X_est(3);
end
end
end
%结束滤波

%滤波结果的数据分析
err_x=zeros(N,1);
err_y=zeros(N,1);

```

```

delta_x=zeros(N,1);
delta_y=zeros(N,1);
%计算滤波的误差均值及标准差
for r=1:N
    %估计误差均值
    ex=sum(Rx(r)-Mt_Est_Px(:,r));
    ey=sum(Ry(r)-Mt_Est_Py(:,r));
    err_x(r)=ex/M;
    err_y(r)=ey/M;
    eqx=sum((Rx(r)-Mt_Est_Px(:,r)).^2);
    eqy=sum((Ry(r)-Mt_Est_Py(:,r)).^2);
    %估计误差标准差
    delta_x(r)=sqrt(abs(eqx/M-(err_x(r)^2)));
    delta_y(r)=sqrt(abs(eqy/M-(err_y(r)^2)));
end
%画图
figure(1);
plot(Rx,Ry,'k',Zx,Zy,'y:',xn,yn,'r');
legend('真实轨迹','观测轨迹','滤波轨迹');
figure(2);
subplot(2,1,1);
plot(err_x);
%axis([1,N,-300,300]);
title('x 方向估计误差均值');
subplot(2,1,2);
plot(err_y);
%axis([1,N,-300,300]);
title('y 方向估计误差均值');
figure(3);
subplot(2,1,1);
plot(delta_x);
%axis([1,N,0,1]);
title('x 方向估计误差标准差');
subplot(2,1,2);
plot(delta_y);

```

```

%axis([1,N,0,1]);
title('y 方向估计误差标准差');

% 子函数
% X_est,P_est 返回第 m 次仿真第 r 个采样点的滤波结果
% Xn_est,Pn_est 记录每个模型对应的第 m 次仿真第 r 次仿真第 r 个采样点的
滤波结果
% u 为模型概率
function [X_est,P_est,Xn_est,Pn_est,u]=IMM(Xn_est,Pn_est,T,Z,Delta,u)
% 控制模型转换的马尔可夫链的转移概率矩阵
P=[0.95,0.025,0.025;0.025,0.95,0.025;0.025,0.025,0.95];
%这里是第三个模型参数，模型一为非机动模型，模型二、三均为机动模型
%模型一
PHI{1,1}=[1,T,0,0;0,1,0,0;0,0,1,T;0,0,0,1];
PHI{1,1}(6,6)=0;
PHI{2,1}=[1,T,0,0,T^2/2,0;0,1,0,0,T,0;0,0,1,T,0,T^2/2;
          0,0,0,1,0,T;0,0,0,0,1,0;0,0,0,0,0,1];%模型二
PHI{3,1}=PHI{2,1};%模型三
G{1,1}=[T/2,0;1,0;0,T/2;0,1];%模型一
G{1,1}(6,2)=0;
G{2,1}=[T^2/4,0;T/2,0;0,T^2/4;0,T/2;1,0;0,1];%模型二
G{3,1}=G{2,1};%模型三
Q{1,1}=zeros(2);%模型一
Q{2,1}=0.005*eye(2);%模型二
Q{3,1}=0.015*eye(2);%模型三
H=[1,0,0,0,0,0;0,0,1,0,0,0];
R=eye(2)*Delta^2;%观测噪声协方差阵
mu=zeros(3,3);%混合概率矩阵
c_mean=zeros(1,3);%归一化常数
for i=1:3
    c_mean=c_mean+P(i,:)*u(i);
end
for i=1:3
    mu(i,:)=P(i,:)*u(i)./c_mean;

```

```

end
%输入交互
for j=1:3
    X0{j,1}=zeros(6,1);
    P0{j,1}=zeros(6);
    for i=1:3
        X0{j,1}=X0{j,1}+Xn_est{i,1}*mu(i,j);
    end
    for i=1:3
        P0{j,1}=P0{j,1}+mu(i,j)*( Pn_est{i,1}...
            +(Xn_est{i,1}-X0{j,1})*(Xn_est{i,1}-X0{j,1}));
    end
end
%模型条件滤波
a=zeros(1,3);
for j=1:3
    %观测预测
    X_pre{j,1}=PHI{j,1}*X0{j,1};
    %协方差预测
    P_pre{j,1}=PHI{j,1}*P0{j,1}*PHI{j,1}'+G{j,1}*Q{j,1}*G{j,1}';
    %计算卡尔曼增益
    K{j,1}=P_pre{j,1}*H'*inv(H*P_pre{j,1}*H'+R);
    %状态更新
    Xn_est{j,1}=X_pre{j,1}+K{j,1}*(Z-H*X_pre{j,1});
    %协方差更新
    Pn_est{j,1}=(eye(6)-K{j,1}*H)*P_pre{j,1};
end
%模型概率更新
for j=1:3
    v{j,1}=Z-H*X_pre{j,1};%新息
    s{j,1}=H*P_pre{j,1}*H'+R;%观测协方差矩阵
    n=length(s{j,1})/2;
    a(1,j)=1/((2*pi)^n*sqrt(det(s{j,1}))) *exp(-0.5*v{j,1}'...
        *inv(s{j,1})*v{j,1});%观测对于模型 j 的似然函数
end

```

```
c=sum(a.*c_mean);%归一化常数
u=a.*c_mean./c;%模型概率更新
%输出交互
Xn=zeros(6,1);
Pn=zeros(6);
for j=1:3
    Xn=Xn+Xn_est{j,1}.*u(j);
end
for j=1:3
    Pn=Pn+u(j).*(Pn_est{j,1}+(Xn_est{j,1}-Xn)*(Xn_est{j,1}-Xn)');
end
%返回滤波结果
X_est=Xn;
P_est=Pn;
```


在学取得成果

一、 在学期间所获的奖励

2017 年，北京科技大学，人民三等奖学金

2018 年，北京科技大学，人民三等奖学金

二、 在学期间发表的论文

三、 在学期间取得的科技成果

致 谢

时光匆匆，四年的本科生生活即将宣告结束，四年前踏入北京科技大学大门的场景仿佛还近在眼前。在这四年间，通过各位专业老师的教导，充实了自己的学术知识和专业素养。回过头才发现，大学最快乐的学生时光已经悄然而逝，即将开始新的人生旅途，继续在本校攻读控制科学与工程的研究生学位。在此向所有帮助过我的老师同学表示衷心的感谢。

首先感谢我的导师，张维存老师，从毕设的开题到结束，他都无微不至地对我的课题内容做了详细的学术指导，帮助我解决了很多学术的疑难问题。在每周例行的组会上，会指引我们小组成员接下来工作的方向，闲暇时间也会对我们为人处事方面做教导。在这里我表示衷心的感谢，能够非常荣幸的在老师的教导下完成毕业设计，获得了不少难能可贵的体验。

其次我想感谢郭金老师对我在毕设过程中的帮助。通过了解我的毕设内容，他对相关课程进行了推荐，同时引荐了学长学姐对我的疑问进行解答，也提供了一些优秀的文献让我参考学习，使我对自己的研究内容多了一些思路和方法。这些帮助都为我的毕设顺利进行提供了非常多的便利，再次表示衷心的感谢。

感谢同一毕设小组的张伯约，王乐宁，申佳晨，张呈同学，我们之间会在组会时讨论各自的研究工作，相互做一定的参考，涉及到一些毕设内容的排版问题也会相互帮助，共同解决。他们同时也是我专业班的同学，互相之间都非常熟悉，给与了我不少的帮助。

对专业班自 175 和行政班自 1705 的各位同学，感谢你们在本科阶段的陪伴，很高兴在这四年与你们相识，这必然是一段难忘的回忆与经历。

最后对各位参与评审和答辩的老师、专家表示真诚的感谢。