

分类号 _____

U D C _____

密 级 _____

编 号 10486 _____

武汉大学

硕 士 学 位 论 文

基于 DSP 的 GNSS/INS 实时
组合导航系统的研究

研 究 生 姓 名 : 常 乐

学 号 : 2014206180014

指导教师姓名、职称 : 章红平 教授

专 业 名 称 : 大地测量学与测量工程

研 究 方 向 : GNSS/INS 组合导航

二〇一七年五月

Research on Real-time GNSS/INS Integrated Navigation
System Based on Digital Signal Processor

By
Chang Le

Supervised by
Prof. Zhang Hongping

Wuhan University

May, 2017

论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的研究成果。除文中已经标明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本声明的法律结果由本人承担。

学位论文作者（签名）：

年 月 日

摘 要

在经济与社会快速发展的今天，人们对位置服务的准确性、可靠性的要求越来越高，单一传感器的导航设备已经无法应对。目前的组合导航系统中，多以 INS 与 GNSS 组合为主，通过 Kalman 滤波有效地将两种信息进行融合，发挥出两种手段的互补特性。

本文研究了 GNSS/INS 组合导航系统的算法，并针对实时导航系统的要求，进行了算法优化，完成了 DSP 嵌入式平台上的 GNSS/INS 实时组合导航系统以及 Windows 平台的配套客户端程序的开发，并在开阔环境和城市环境中进行了系统导航性能的验证。本文研究的主要内容如下：

1、详细阐述了 GNSS/INS 组合导航的基本理论，包含组合导航中的坐标系以及相互之间的转化、捷联惯性导航的机械编排原理、GNSS/INS 组合导航系统的 Kalman 滤波器构成和车辆约束。

2、描述了实时组合导航系统的详细实现方式，重点探讨了组合导航的实时性改进。针对 GNSS/INS 嵌入式实时组合导航系统中存在的 GNSS 数据延迟以及组合更新解算耗时过长的的问题，本文基于 Kalman 滤波理论，提出了一种在完成了滞后的观测更新后将误差状态向量一步转移到当前时刻的 Kalman 滤波观测更新的数学模型。该模型能够有效降低 GNSS 数据延迟和组合解算耗时对实时导航结果输出的影响，因此能够容忍观测信息有较大的时间延迟，并允许观测更新解算在比较充足的时间里完成，对组合导航算法在运算能力较低的处理器上的实现和应用于实时性要求严格的场合有重要的意义。此外，本文对组合导航硬件平台设计方案进行了详细说明，介绍了嵌入式平台的 GNSS/INS 实时组合导航程序以及 Windows 平台的客户端应用程序的具体实施方法。

3、为了验证本文完成的 GNSS/INS 实时组合导航系统，在武汉市进行了多次车载测试，并且在测试中通过拔掉 GNSS 天线馈线的方式模拟 GNSS 卫星信号中断。导航误差统计表明，本文完成的 GNSS/INS 实时组合导航系统的位置标准差为 0.05m，速度标准差为 0.005m/s，横滚、俯仰标准差为 0.004deg，航向标准差为 0.03deg，达到了预期导航精度，并且 GNSS 卫星信号中断期间的漂移水

平符合器件水平，各项车辆辅助工作正常。测试实验中的耗时统计表明，相对于标准 Kalman 模型，本文提出的实时性改进模型能够在保障组合导航系统精度的前提下，将输出导航结果的时间延迟减小到 1ms 左右，不受 GNSS 观测数据以及组合更新计算延迟的影响，确保了导航系统的实时性能。

关键词：GNSS/INS 组合导航；实时性；DSP；Kalman 滤波；车辆导航

ABSTRACT

With the development of today's society, the accuracy and reliability of the navigation system are required higher and higher. The single navigation system has been unable to meet the needs of practical applications. At present, the combination of the inertial navigation system and the satellite navigation system is the main component in the various kinds of integrated navigation systems, and the Kalman filtering method can effectively integrate the information of the two system to realize the complementary characteristics of the integrated navigation system.

In this thesis, the GNSS/INS integrated navigation system is implemented and the algorithm is optimized for the real-time system. The real-time GNSS/INS integrated navigation system of the embedded platform and the client of the Windows platform are developed. And the validation of the system developed in this thesis was conducted in an open and urban environment. The main research contents are as follows:

1、The basic knowledge of GNSS / INS integrated navigation is introduced in detail, including the common coordinate system and mutual conversion relationship, the mechanization of strapdown inertial navigation, the Kalman filter composition of GNSS / INS integrated navigation system and the additional constraints in the vehicle mode.

2、This thesis studies the concrete realization method of real-time integrated navigation system, and discusses the real-time improvement of the system. The problems of the GNSS data delay and too long time consuming for Kalman updating exist in the real-time GNSS/INS integrated navigation system. In this thesis, an improved mathematical model which can transfer error state to the current time in one step after finishing the delayed observation update is proposed based on the Kalman filter theory. The improved model can reduce the effect of GNSS data delay and time-consuming observation update on the real time navigation output. So the model allows to finish the observation update in more sufficient time. It has the important significance

for the integrated navigation algorithm's implementation in a low computing power processor and the application with highly real time requirement. Then, the design scheme of the integrated navigation hardware platform is described in detail, and the implementation method of the real-time GNSS / INS integrated navigation program in the embedded platform and the client program in the Windows platform are introduced.

3、 In order to verify the real-time GNSS / INS integrated navigation system in this thesis, many vehicle tests has been carried out in Wuhan, and in these tests we unplug the GNSS antenna to simulate the GNSS satellite signal interrupt. The navigation error statistics of these tests show that the standard deviation of position and velocity is 0.05m, 0.005m/s respectively, the standard deviation of roll and pitch is 0.004deg, and the standard deviation of heading is 0.03deg. The above results achieved the expected navigation accuracy. The drift level during the interruption of the GNSS satellite signal is in accordance with the IMU level and all the additional constraints in the vehicle mode work well. The time-consuming statistics in these tests show that, compared with the standard Kalman model, the improved model proposed in this thesis can reduce the time delay to about 1ms and is not affected by the GNSS observation data and the Kalman update calculation delay under the premise of ensuring the accuracy of the navigation system. So the improved model can make sure of the system's real time performance.

Keyword: GNSS/INS integrated navigation; real-time performance; DSP; Kalman filter; vehicle navigation

目 录

摘 要.....	I
ABSTRACT.....	III
1 绪论.....	1
1.1 研究背景和意义.....	1
1.2 国内外研究现状.....	2
1.3 导航系统嵌入式平台（DSP）概述.....	4
1.4 论文章节安排.....	4
2 GNSS/INS 组合导航系统原理.....	6
2.1 常用坐标系统及转换.....	6
2.1.1 组合导航坐标系统.....	6
2.1.2 组合导航坐标系统间的变换矩阵.....	8
2.2 惯性导航算法.....	10
2.2.1 观测误差补偿.....	10
2.2.2 速度更新.....	11
2.2.3 位置更新.....	14
2.2.4 姿态更新.....	14
2.3 GNSS/INS 组合 Kalman 滤波.....	16
2.3.1 Kalman 滤波.....	17
2.3.2 状态方程.....	19
2.3.3 观测方程.....	23
2.3.4 观测噪声和状态噪声方差阵的设置.....	23
2.3.5 状态估计的反馈.....	24
2.4 GNSS/INS 组合导航抗差处理.....	25
2.5 车辆约束.....	26
2.5.1 IMU 安装角估计.....	26
2.5.2 ZUPT 辅助.....	27
2.5.3 里程计辅助.....	27
2.5.4 NHC 辅助.....	29
2.5.5 里程计杆臂与 IMU 安装误差角的影响.....	29
3 GNSS/INS 实时组合导航系统设计.....	31
3.1 时间延迟处理及效果验证.....	31
3.1.1 状态转移.....	32
3.1.2 效果验证.....	34
3.2 组合导航硬件平台设计.....	35
3.3 时间同步处理.....	37
3.4 DSP 端程序设计.....	40

3.4.1 引导程序设计.....	40
3.4.2 组合导航程序设计.....	41
3.5 PC 端程序设计	44
3.5.1 程序关键模块介绍.....	45
3.5.2 程序操作说明.....	48
4 GNSS/INS 实时组合导航性能测试与验证.....	52
4.1 导航性能评估方法	53
4.2 测试实验	54
4.2.1 开阔环境中无车辆辅助的测试.....	54
4.2.2 开阔环境中使用 NHC 辅助的测试	55
4.2.3 开阔环境中使用 NHC 及里程计辅助的测试	57
4.2.4 城市环境中的测试.....	59
4.3 实时性能分析	60
4.4 测试结论	61
5 总结与展望	63
6 附录.....	64
6.1 导航状态数据格式	64
6.2 导航参数配置格式	65
6.3 交互命令说明	66
6.4 PC 端程序辅助工具	66
6.4.1 解码.....	66
6.4.2 估计 IMU 安装误差角.....	67
6.4.3 固件升级.....	67
参考文献.....	68
致谢.....	71

1 绪论

1.1 研究背景和意义

导航就是获取载体相对于某一参考坐标系的位置、速度、姿态等信息，并将载体引导到目标位置的技术或方法^[1]。随着当今社会的发展，对导航系统的准确性和可靠性都提出了更苛刻的指标，要求导航系统为用户提供全天候、全球、全面的实时导航信息，单一的导航系统已经很难满足实际应用的需求，将两种或两种以上的导航系统组合在一起成为了目前导航系统的发展方向。

当前各种不同的组合导航系统中，多以惯性导航系统(Inertial Navigation System, INS)为主要导航系统，全球卫星导航系统(Global Navigation Satellite System, GNSS)为辅助导航系统的 GNSS/INS 组合导航系统^[2]为主。作为经典的主动式导航方法，惯性导航是通过陀螺仪、加速度计观测载体在惯性空间的角速度、加速度，由运动方程准确地得到载体详细的运动状态^[1]，包括载体位置、速度、姿态、角速度、加速度，是最重要、最主要的现代导航方式之一。惯性导航技术最早可追溯到上世纪 40 年代火箭技术发展的初期，第二次世界大战期间，经过德国人冯布劳恩改造后的陀螺成功地应用在了 V-2 火箭的制导中^[3]。在此之后，越来越多的高校、研究机构深入研究了惯性导航，随着时代发展，最初的平台式惯性导航也有了更多的种类，比如微机电、光纤、激光惯导^[3]等。惯性导航系统拥有动态反应灵敏、短时间内导航误差小、观测数据采样率高、主动式工作等特点，但由于采用时间积分的方式，其导航误差会随时间的累积逐步变大，不适合长期工作。而全球卫星导航系统利用载体被动接收来自导航卫星的伪距、载波观测值，能够为载体提供全天候、连续、实时的位置、速度导航信息，在载波差分模式下的定位精度可达厘米级，具有导航精度不受工作时间的影响的特点。目前全球卫星导航系统已经成熟地应用于导航、定位、授时等领域^[4]。GPS 在全球导航领域占据垄断地位，我国于 2004 年开始了北斗卫星导航二代系统的建设，已具备区域导航定位能力，今年开始了第三代北斗卫星导航系统的建设，计划 2020 年实现全球导航定位能力，使我国自主研发的北斗系统成为继美国的 GPS、

俄罗斯的 GLONASS 之后全球第三个完备的卫星导航系统^[5]。卫星导航作为被动式导航的一种,很容易受到干扰,依赖于连续稳定的卫星信号,另外,卫星导航还有数据低(典型数据率为 1Hz)、短期定位噪声大的缺点。可以看出,卫星导航与惯性导航系统的特点形成鲜明互补,通过 Kalman 滤波的方式有效地将 GNSS 信息与 INS 信息融合起来形成的 GNSS/INS 组合导航可以发挥两种系统的互补特性,惯性导航系统在很大程度上弥补了卫星导航易受到干扰的缺点,卫星导航则为惯性导航提供周期性的修正信息,产生协同作用提高导航性能,在载体经历高动态或者 GNSS 信号载噪比较小时这种优势更加明显,比单一系统单独使用时有大幅改善^[6]。GNSS/INS 组合导航系统已经历了几十年的发展,其理论基础研究已经日趋完善,已经被广泛应用于武器制导、航空航天、高精度移动测绘、道路检查等领域。实时 GNSS/INS 组合导航系统在完成大量数据采集的同时,还要完成惯性导航计算、GNSS 定位解算、数据融合以及导航输出等工作,对系统性能有很高的要求。

目前,自动驾驶、智能机器人等领域对载体实时导航的需求越来越大,尤其是在城市复杂环境下的高精度导航定位^[7]。实时 GNSS/INS 组合导航系统作为比较成熟的导航定位手段得到了越来越多的应用,具有很大的实际应用需求,开展该系统的研制工作是十分有意义的。

1.2 国内外研究现状

从惯性导航理论提出到具体导航系统的研制已经发展了许多年,惯性导航系统已经相当成熟。伴随卫星导航系统的发展而出现的 GNSS/INS 组合导航系统也已经历几十年的发展,其理论的逐步完善,被广泛应用。目前比较成熟的组合导航数据处理商业软件有 NovAtel 公司的 Inertial Explorer、Applanix 公司的 POSPac、加拿大卡尔加里大学基于 Matlab 开发的 AINS 软件以及武汉迈普时空公司的 GINS 软件。国内外实时 GNSS/INS 组合产品也十分丰富,国外的 NovAtel、Trimble 公司以及国内的星网宇达、耐威科技公司等都有不同等级的实时组合导航系统。

GNSS/INS 组合导航系统的研究对于已经相当成熟,更多的国内外院校、科研单位以及企业侧重于实时导航软硬研究^[8-11]。目前实时导航系统的处理器组成

方案主要有：单独使用 DSP 芯片，与传感器进行信息交换的同时进行导航数据解算；DSP 联合 ARM(Acorn RISC Machine)芯片的组合方式，DSP 进行导航解算，ARM 主要进行传感器数据交互；DSP 结合 FPGA(Field Programmable Gate Array)芯片的处理方式，利用 FPGA 的并行性，实现外部数据高效采集，DSP 负责数据融合并输出导航结果。DSP 作为专用数字信号处理器，不同方案都将其作为核心算法处理器。考虑到 FPGA 的成本高，且需要特定的编程语言进行开发，本文采用多个 ARM 和 DSP 结合的方式：多个 ARM 协同工作，利用其接口丰富的特点负责在线降采样、补偿 IMU 原始数据，打包发送 IMU 原始数据至各个外设；DSP 利用其数字计算能力强的特点，专门进行从 ARM 接收 IMU 原始数据以及从 GNSS 板卡接收 GNSS 原始数据，并进行组合导航解算的任务。

在实时应用场景中，时间延迟问题是一个不得不考虑的问题，它会直接影响导航系统的实时性能：观测数据时间延迟会导致当前时刻观测数据无法实时获取，组合解算时间延迟会导致观测数据无法及时处理。目前，时间延迟的解决策略主要采用增广状态方法^[12, 13]和基于新息重组理论^[14, 15]建立最优滤波器：增广状态方法会增加状态维度，增大计算负担；而新息重组的方法，需要建立多个观测方程，如果有多步延迟，则不利于对 Kalman 滤波器的灵活拓展与应用。并且这些方法只能解决观测数据传输延迟，不能解决更新计算延迟的问题。针对该问题，本文基于 Kalman 滤波理论，提出了一种在完成了滞后的观测更新后将误差状态向量一步转移到当前时刻的 Kalman 滤波数学模型。该模型能够降低 GNSS 数据延迟和组合解算耗时对实时导航结果输出的影响，因此允许观测更新解算在比较充足的时间里从容完成，对组合导航算法在运算能力较低的处理器的实现和对实时性要求严格的应用场合有重要的意义。

伴随着传感器技术和处理器的发展，未来 GNSS/INS 组合导航系统将拥有更高的导航精度、更低的成本、更小的体积、更低的功耗以及更高的稳健性^[16]，数据处理不局限于某一特定平台，各种终端的组合导航处理软件会得到发展，以满足多种导航场景下的导航需求。

1.3 导航系统嵌入式平台（DSP）概述

实时导航系统的实现中，选择一款体积小、功耗低、效率高的导航计算机是十分有必要的，目前的嵌入式处理器主要包括 DSP、ARM、FPGA^[17]。随着电子科技的发展，DSP 在数字信号处理方面得到了越来越广泛的应用，其高运行速度和强大的数据处理能力正好满足了 GNSS/INS 组合导航算法运算量大、采样率高的要求，成为了导航计算机的首要选择。DSP 的特点^[18, 19]主要有：

- 1) 拥有独立的累加器、加法器，一次乘法和一次加法可以同时在一个指令周期内可完成；
- 2) 采样了程序和数据空间分开的哈佛结构和改进的哈佛结构，执行指令速度更快，并且可以同时访问指令和数据；
- 3) 内部具有快速 RAM(Random Access Memory)，访问速度优于普通内存，但大小有限；
- 4) 在硬件层面支持低开销或无开销循环、跳转；
- 5) 丰富的软硬件中断处理和硬件 I/O；
- 6) 内部总线丰富，不同数据存取操作可以同步进行；
- 7) 可以利用流水线操作并行运行取指、译码和执行等操作，有效地缩短了指令执行的时间。

DSP 产品的类型丰富，按照性能的差异可以分为低、中、高端 3 类，本文使用的为高端 DSP，型号为 TI 的 TMS320C6747，主频高达 300MHz，可超频至 480MHz^[20]。

1.4 论文章节安排

实时 GNSS/INS 组合导航系统是组合导航中很有实际应用需求的系统之一，本文在分析和研究现有 GNSS/INS 组合系统实现方法的基础上，开发了 DSP 平台上的实时 GNSS/INS 组合导航系统以及 Windows 平台上的配套程序，全文共五章，具体安排如下：

第一章首先说明了 GNSS/INS 实时组合导航的背景和意义，然后简述了组合

导航的发展历史、国内外研究现状以及未来的发展趋势，最后介绍了全文的章节安排。

第二章说明了 GNSS/INS 组合导航的基础理论，首先对导航解算中使用的坐标系统和相互变换公式做了说明；然后对惯导机械编排算法公式进行了推导；接着分析了 GNSS/INS 组合的 Kalman 滤波器的具体组成，包括状态方程、观测方程的具体表达式以及初始化参数的设置方法、状态参数的反馈；最后介绍了车辆约束的使用方法，包含零速修正、里程计辅助以及非完整性约束。

第三章介绍了实时组合导航系统具体实现方法，首先介绍了实时系统中时间延迟的处理方案；接着介绍了组合导航系统硬件平台设计以及时间同步处理方法，详细说明了组合导航系统的硬件组成与集成方案；然后介绍了嵌入端程序设计，包括引导程序的设计以及组合导航程序的具体实施方法；最后介绍了 PC 端的交互程序设计，说明了程序中关键模块的开发方法以及程序操作。

第四章为测试实验，为验证本文开发的 GNSS/INS 实时组合导航系统的性能，本文进行了多次车载测试，包含开阔天空以及复杂城市环境，使用更高精度的组合导航系统的后处理结果作为参考基准，详细分析了本文开发的实时导航系统的性能。

第五章为总结与展望，对本文的研究内容做了综合阐述，总结了本文开发的 GNSS/INS 实时组合导航系统的不足，提出今后深入展开的方向。

2 GNSS/INS 组合导航系统原理

作为目前最具有应用价值的组合导航模式之一，GNSS/INS 组合导航系统的基本理论已十分成熟。本章首先对导航解算中使用的坐标系统和相互变换公式做了说明；接着推导了惯性导航的机械编排算法；最后描述了 GNSS/INS 组合导航 Kalman 滤波器的设计。

2.1 常用坐标系统及转换

组合导航系统中往往会涉及到多个坐标系统，需要对不同坐标系下的信息进行融合，下面给出 GNSS/INS 组合导航解算中使用的坐标系统以及相互变换公式。

2.1.1 组合导航坐标系统

1) i 系——惯性坐标系

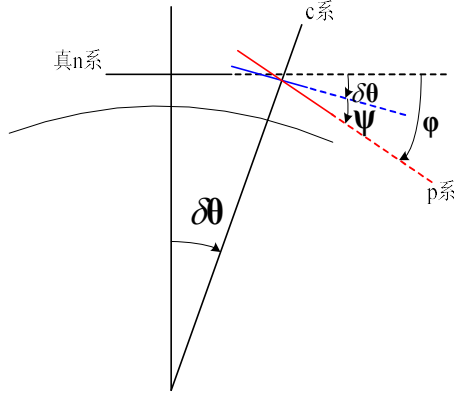
惯性坐标系为在空间中不变的坐标系，不同的应用场景有不同的定义，日心、地心、地球卫星轨道惯性坐标系等都是比较常用的惯性坐标系，通常使用地心惯性坐标系来研究地球表面附近运动的载体。地心惯性坐标系的定义为：忽略地球公转以及太阳的运动，取地球质心为坐标原点，将地球自转轴作为 Z 轴，北极方向为正，X 轴与 Z 轴垂直，X 轴指向春分点，Y 轴与 XOZ 平面垂直，形成右手坐标系。

2) e 系——地心地固坐标系

地心地固坐标系跟随地球旋转，以地球质心为坐标原点，X 轴指向赤道与本初子午线的交点，将地球自转轴作为 Z 轴，北极方向为正，Y 轴与 XOZ 平面垂直，形成右手坐标系。

3) n 系——导航坐标系

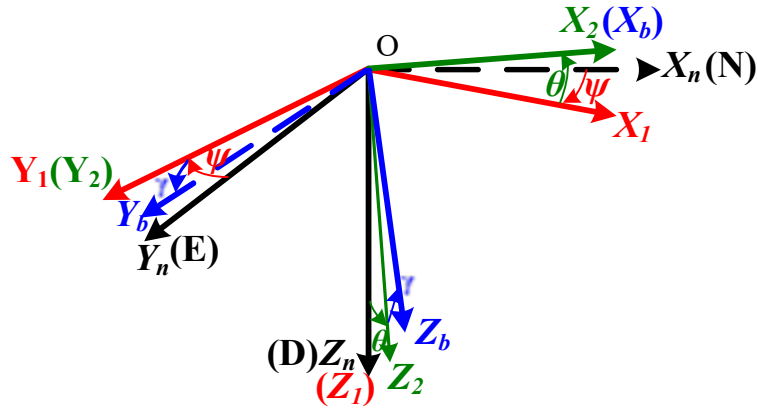
导航坐标系为描述导航结果时使用的坐标系，是一种当地水平坐标系。原点为载体重心，XOY 平面为当地水平面，X 轴为指向北的经线切线，Y 轴为指向


 图 2.3 真 n 系与 c 系和 p 系的关系图^[21]

2.1.2 组合导航坐标系统间的变换矩阵

1) n 系与 b 系的变换

n 系到 b 系的旋转可以由载体的姿态进行描述, 首先把 n 系围绕 Z_n 轴逆时针旋转航向角(ψ), 然后围绕 Y_1 逆时针旋转俯仰角(θ), 最后围绕 X_2 逆时针旋转横滚角(γ)^[16]。


 图 2.4 n 系转到 b 系^[21]

上述旋转过程可以由方向余弦矩阵 C_n^b 表示:

$$\begin{aligned}
 C_n^b &= R_x(\gamma)R_y(\theta)R_z(\psi) \\
 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & \sin \gamma \\ 0 & -\sin \gamma & \cos \gamma \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ -\cos \gamma \sin \psi + \sin \gamma \sin \theta \cos \psi & \cos \gamma \cos \psi + \sin \gamma \sin \theta \sin \psi & \sin \gamma \cos \theta \\ \sin \gamma \sin \psi + \cos \gamma \sin \theta \cos \psi & -\sin \gamma \cos \psi + \cos \gamma \sin \theta \sin \psi & \cos \theta \cos \gamma \end{bmatrix}
 \end{aligned} \tag{2.1}$$

2) e 系与 n 系的变换

先将 e 系绕 z_e 轴旋转 λ 角, 接着再绕 Y_e 轴旋转 $(-\varphi - \pi/2)$ 角, 便可转换到 n 系, 如图 2.1, 其中 λ 、 φ 分别表示经度、纬度, 旋转可由方向余弦矩阵 \mathbf{C}_e^n 表示:

$$\begin{aligned}\mathbf{C}_e^n &= \mathbf{R}_Y(-\varphi - \frac{\pi}{2})\mathbf{R}_Z(\lambda) \\ &= \begin{bmatrix} -\sin\varphi & 0 & \cos\varphi \\ 0 & 1 & 0 \\ -\cos\varphi & 0 & -\sin\varphi \end{bmatrix} \begin{bmatrix} \cos\lambda & \sin\lambda & 0 \\ -\sin\lambda & \cos\lambda & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} -\sin\varphi\cos\lambda & -\sin\varphi\sin\lambda & \cos\varphi \\ -\sin\lambda & \cos\lambda & 0 \\ -\cos\varphi\cos\lambda & -\cos\varphi\sin\lambda & -\sin\varphi \end{bmatrix}\end{aligned}\quad (2.2)$$

3) n 系与 c 系的变换

n 系到 c 系的方向余弦可以表示为:

$$\mathbf{C}_n^c = \mathbf{I} - (\delta\boldsymbol{\theta} \times) \quad (2.3)$$

$$\delta\boldsymbol{\theta} = \begin{bmatrix} \delta r_E / (N+h) \\ -\delta r_N / (M+h) \\ -\delta r_E \tan\varphi / (N+h) \end{bmatrix} \quad (2.4)$$

式中, δr_E 、 δr_N 分别表示东向、北向位置误差, M 、 N 分别表示子午圈曲率、卯酉圈曲率半径, h 为载体高程。

4) c 系与 p 系的变换

c 系到 p 系的方向余弦可以表示为:

$$\mathbf{C}_c^p = \mathbf{I} - (\boldsymbol{\psi} \times) \quad (2.5)$$

式中, $\boldsymbol{\psi}$ 为 c 系到 p 系的旋转矢量。

5) n 系与 p 系的变换

n 系到 p 系的方向余弦可以表示为:

$$\mathbf{C}_n^p = \mathbf{I} - (\boldsymbol{\varphi} \times) \quad (2.6)$$

式中, $\boldsymbol{\varphi}$ 为 n 系到 p 系的旋转矢量, 由图 2.3 可知 $\boldsymbol{\varphi} = \boldsymbol{\psi} + \delta\boldsymbol{\theta}$ 。

2.2 惯性导航算法

捷联惯性导航系统使用“数学平台”来替代平台式惯性导航系统中的实体平台的功能，实现机械编排的过程，节省了设备空间以及复杂的机械控制。主要过程如下：在给定了初始位置、姿态的情况下，利用陀螺仪观测的 b 系相对于 i 系的角速度，扣除 n 系相对于 i 系的角速度，得到 b 系相对于 n 系的角速度，使用该角速度计算姿态增量，即可进行载体的姿态更新，加速度计观测的 b 系中的比力通过已知的载体姿态转化到 n 系中，从而进行速度、位置更新。具体流程如图 2.5 所示。

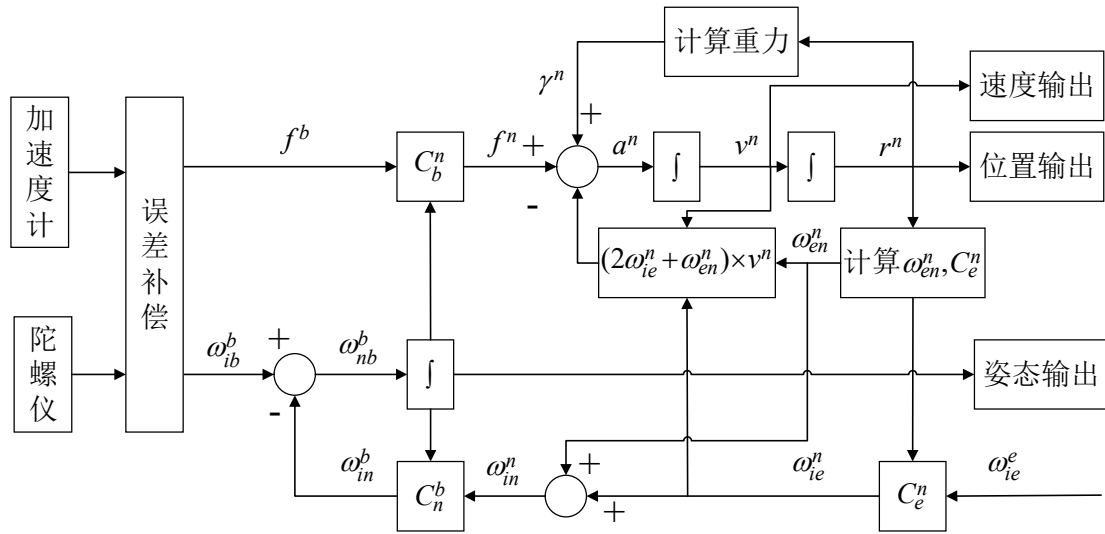


图 2.5 机械编排流程图

2.2.1 观测误差补偿

惯性传感器的观测数据会受到零偏、比例因子以及交轴耦合的影响，导致观测数据存在误差，因此，在使用这些观测数据时，必须进行补偿，本文忽略交轴耦合的影响，陀螺的角增量输出和加速度计的速度增量输出可以通过式(2.7)进行误差补偿^[22]。

$$\begin{cases} \Delta \theta_k = [\mathbf{I} + \text{diag}(\mathbf{s}_g)]^{-1} [\Delta \tilde{\theta}_k - \mathbf{b}_g \Delta t_k] \\ \Delta \mathbf{v}_{f,k}^b = [\mathbf{I} + \text{diag}(\mathbf{s}_a)]^{-1} [\Delta \tilde{\mathbf{v}}_{f,k}^b - \mathbf{b}_a \Delta t_k] \end{cases} \quad (2.7)$$

式中， \mathbf{s}_g 、 \mathbf{b}_g 分别为陀螺的比例因子、零偏， \mathbf{s}_a 、 \mathbf{b}_a 分别为加速度计的比例因子、零偏， $\Delta \tilde{\theta}_k$ 、 $\Delta \tilde{\mathbf{v}}_{f,k}^b$ 分别为陀螺、加速度计的原始观测量， $\Delta \theta_k$ 、 $\Delta \mathbf{v}_{f,k}^b$ 分别

为陀螺、加速度计经过误差补偿的观测量， Δt_k 为增量形式观测量的观测时间长度。

里程计的输出值会受到刻度因子以及车辆轮胎形变的影响，在使用里程计测量速度时需要引入里程计比例因子估计量，以补偿里程计的测量速度，具体补偿公式如下：

$$v_{odom} = \tilde{v}_{odom} / (1 + s_{odom}) \quad (2.8)$$

式中， \tilde{v}_{odom} 为里程计观测速率， v_{odom} 为补偿后的里程计观测速率， s_{odom} 为里程计比例因子。

2.2.2 速度更新

通过哥氏定理可以得到捷联惯性导航的速度微分方程^[23]：

$$\dot{\mathbf{v}}^n = \mathbf{C}_b^n \mathbf{f}^b + \mathbf{g}^n - 2\boldsymbol{\omega}_{ie}^n \times \mathbf{v}^n - \boldsymbol{\omega}_{en}^n \times \mathbf{v}^n \quad (2.9)$$

式中， \mathbf{v}^n 表示载体速度， \mathbf{f}^b 表示加表观测的比力， \mathbf{g}^n 表示重力加速度， $2\boldsymbol{\omega}_{ie}^n \times \mathbf{v}^n$ 表示哥氏加速度， $\boldsymbol{\omega}_{en}^n \times \mathbf{v}^n$ 表示向心加速度。其中， $\boldsymbol{\omega}_{ie}^n$ 和 $\boldsymbol{\omega}_{en}^n$ 由下式计算：

$$\begin{cases} \boldsymbol{\omega}_{ie}^n = \mathbf{C}_e^n \boldsymbol{\omega}_e^e = [\omega_e \cos \varphi & 0 & -\omega_e \sin \varphi]^T \\ \boldsymbol{\omega}_{en}^n = \begin{bmatrix} \frac{v_E}{N+h} & -\frac{v_N}{M+h} & -\frac{v_E \tan \varphi}{N+h} \end{bmatrix}^T \end{cases} \quad (2.10)$$

式中， ω_e 为地球自转角速度， φ 为载体纬度， v_N 为北向速度， v_E 为东向速度， N 为卯酉圈曲率半径， M 为子午圈曲率半径， h 为载体高程。卯酉圈曲率半径以及子午圈曲率半径如下进行计算：

$$\begin{cases} M = \frac{a(1-e^2)}{(1-e^2 \sin^2 \varphi)^{3/2}} \\ N = \frac{a}{(1-e^2 \sin^2 \varphi)^{1/2}} \end{cases} \quad (2.11)$$

式中， a 为地球椭圆的长半径， e 为地球椭圆的偏心率。

速度更新方程可以通过对微分方程(2.9)离散后得到^[23]：

$$\begin{aligned} \mathbf{v}_{t_k}^n &= \mathbf{v}_{t_{k-1}}^n + \int_{t_{k-1}}^{t_k} \mathbf{C}_b^n(t) \mathbf{f}^b(t) dt + \int_{t_{k-1}}^{t_k} (\mathbf{g}^n - (2\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n) \times \mathbf{v}^n) dt \\ &= \mathbf{v}_{t_{k-1}}^n + \Delta \mathbf{v}_{f,t_k}^n + \Delta \mathbf{v}_{g/cor,t_k}^n \end{aligned} \quad (2.12)$$

式中， $\Delta \mathbf{v}_{g/cor, t_k}^n$ 代表向心加速度、哥氏加速度和地球重力带来的速度变化， $\Delta \mathbf{v}_{f, t_k}^n$ 代表比力引起的速度变化，可如下表示：

$$\begin{aligned}
 \Delta \mathbf{v}_{f, t_k}^n &= \int_{t_{k-1}}^{t_k} \mathbf{C}_b^n(t) \mathbf{f}^b(t) dt \\
 &= \int_{t_{k-1}}^{t_k} \mathbf{C}_{n(t_{k-1})}^n(t) \mathbf{C}_{b(t_{k-1})}^{b(t)} \mathbf{f}^b(t) dt \\
 &= \mathbf{C}_{n(t_{k-1})}^{\bar{n}} \mathbf{C}_{b(t_{k-1})}^{n(t_{k-1})} \int_{t_{k-1}}^{t_k} \mathbf{C}_{b(t)}^{b(t_{k-1})} \mathbf{f}^b(t) dt \\
 &= \frac{1}{2} \left[\mathbf{C}_{n(t_{k-1})}^{n(t_k)} + \mathbf{I} \right] \mathbf{C}_{b(t_{k-1})}^{n(t_{k-1})} \Delta \mathbf{v}_{f, t_k}^{b(t_{k-1})}
 \end{aligned} \tag{2.13}$$

式中，由于 n 系在采样间隔内变化不大，使用 t_{k-1} 时刻到 t_k 时刻的平均 n 系 \bar{n} 表示。

$$\begin{aligned}
 \Delta \mathbf{v}_{f, t_k}^{b(t_{k-1})} &= \int_{t_{k-1}}^{t_k} \mathbf{C}_{b(t)}^{b(t_{k-1})} \mathbf{f}^b(t) dt \\
 &= \int_{t_{k-1}}^{t_k} [\mathbf{I} + (\Delta \boldsymbol{\theta}(t) \times)] \mathbf{f}^b(t) dt \\
 &= \Delta \mathbf{v}_{f, t_k}^b + \int_{t_{k-1}}^{t_k} \Delta \boldsymbol{\theta}(t) \times \mathbf{f}^b(t) dt
 \end{aligned} \tag{2.14}$$

式中， $\Delta \boldsymbol{\theta}(t)$ 为角增量，可表示为：

$$\Delta \boldsymbol{\theta}(t) = \int_{t_{k-1}}^t \boldsymbol{\omega}_{ib}^b(\tau) d\tau \tag{2.15}$$

对式(2.14)的积分部分进行处理：

$$\begin{aligned}
 \frac{d(\Delta \boldsymbol{\theta}(t) \times \Delta \mathbf{v}(t))}{dt} &= \Delta \boldsymbol{\theta}(t) \times \dot{\mathbf{f}}^b(t) + \boldsymbol{\omega}_{ib}^b(t) \times \Delta \mathbf{v}(t) \\
 \int_{t_{k-1}}^{t_k} \Delta \boldsymbol{\theta}(t) \times \mathbf{f}^b(t) dt &= \Delta \boldsymbol{\theta}(t_k) \times \Delta \mathbf{v}(t_k) + \int_{t_{k-1}}^{t_k} \Delta \mathbf{v}(t) \times \boldsymbol{\omega}_{ib}^b(t) dt
 \end{aligned} \tag{2.16}$$

等式左右同时加上左边部分，可得：

$$\int_{t_{k-1}}^{t_k} \Delta \boldsymbol{\theta}(t) \times \mathbf{f}^b(t) dt = \frac{1}{2} \Delta \boldsymbol{\theta}(t_k) \times \Delta \mathbf{v}(t_k) + \frac{1}{2} \int_{t_{k-1}}^{t_k} (\Delta \boldsymbol{\theta}(t) \times \mathbf{f}^b(t) + \Delta \mathbf{v}(t) \times \boldsymbol{\omega}_{ib}^b(t)) dt \tag{2.17}$$

假设角速度和比力是线性变换的，即双子样推导法，可得：

$$\frac{1}{2} \int_{t_{k-1}}^{t_k} (\Delta \boldsymbol{\theta}(t) \times \mathbf{f}^b(t) + \Delta \mathbf{v}(t) \times \boldsymbol{\omega}_{ib}^b(t)) dt = \frac{1}{12} (\Delta \boldsymbol{\theta}_{t_{k-1}} \times \Delta \mathbf{v}_{f, t_k}^b + \Delta \mathbf{v}_{f, t_{k-1}}^b \times \Delta \boldsymbol{\theta}_{t_k}) \tag{2.18}$$

比力引起的速度增量最终表示为：

$$\Delta \mathbf{v}_{f, t_k}^{b(t_{k-1})} = \Delta \mathbf{v}_{f, t_k}^b + \frac{1}{2} \Delta \boldsymbol{\theta}(t_k) \times \Delta \mathbf{v}(t_k) + \frac{1}{12} (\Delta \boldsymbol{\theta}_{t_{k-1}} \times \Delta \mathbf{v}_{f, t_k}^b + \Delta \mathbf{v}_{f, t_{k-1}}^b \times \Delta \boldsymbol{\theta}_{t_k}) \tag{2.19}$$

式 $\mathbf{C}_{n(t_{k-1})}^{n(t_k)} = \mathbf{I} - (\boldsymbol{\zeta}_k \times)$ 代入式(2.13)中:

$$\begin{cases} \Delta \mathbf{v}_{f,t_k}^n = [\mathbf{I} - (0.5\boldsymbol{\zeta}_k \times)] \mathbf{C}_{b(t_{k-1})}^{n(t_{k-1})} \Delta \mathbf{v}_{f,t_k}^{b(t_{k-1})} \\ \boldsymbol{\zeta}_k = [\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n]_{t_{k-1/2}} \Delta t_k \end{cases} \quad (2.20)$$

式中, $\boldsymbol{\zeta}_k$ 表示 n 系 t_{k-1} 时刻到 t_k 时刻的旋转矢量, $t_{k-1/2}$ 表示 t_{k-1} 与 t_k 时刻的中间时刻。为了得到中间时刻的角速度, 本文采用外推的方式。

根据经纬度可以用四元数 \mathbf{q}_n^e 来表示 n 系到 e 系的旋转, 如下:

$$\mathbf{q}_n^e = \begin{bmatrix} \cos(-\pi/4 - \varphi/2) \cos(\lambda/2) \\ -\sin(-\pi/4 - \varphi/2) \sin(\lambda/2) \\ \sin(-\pi/4 - \varphi/2) \cos(\lambda/2) \\ \cos(-\pi/4 - \varphi/2) \sin(\lambda/2) \end{bmatrix} \quad (2.21)$$

纬度和经度的外推可以通过旋转四元数 \mathbf{q}_n^e 得到, 公式如下:

$$\mathbf{q}_{n(t_{k-1/2})}^{e(t_{k-1/2})} = \mathbf{q}_{e(t_{k-1})}^{e(t_{k-1/2})} \otimes \mathbf{q}_{n(t_{k-1})}^{e(t_{k-1})} \otimes \mathbf{q}_{n(t_{k-1/2})}^{n(t_{k-1})} \quad (2.22)$$

$$\begin{cases} \mathbf{q}_{n(t_{k-1/2})}^{n(t_{k-1})} = \begin{bmatrix} \cos \|0.5\boldsymbol{\zeta}_{t_{k-1/2}}\| \\ \frac{\sin \|0.5\boldsymbol{\zeta}_{t_{k-1/2}}\|}{\|\boldsymbol{\zeta}_{t_{k-1/2}}\|} \boldsymbol{\zeta}_{t_{k-1/2}} \end{bmatrix} \\ \mathbf{q}_{e(t_{k-1})}^{e(t_{k-1/2})} = \begin{bmatrix} \cos \|0.5\boldsymbol{\xi}_{t_{k-1/2}}\| \\ -\frac{\sin \|0.5\boldsymbol{\xi}_{t_{k-1/2}}\|}{\|\boldsymbol{\xi}_{t_{k-1/2}}\|} \boldsymbol{\xi}_{t_{k-1/2}} \end{bmatrix} \end{cases} \quad (2.23)$$

式中, $\boldsymbol{\xi}_{t_{k-1/2}}$ 、 $\boldsymbol{\zeta}_{t_{k-1/2}}$ 分别为 e 系、n 系从 t_{k-1} 时刻旋转到 $t_{k-1/2}$ 时刻的旋转矢量:

$$\begin{cases} \boldsymbol{\xi}_{t_{k-1/2}} = \boldsymbol{\omega}_{ie}^e(t_{k-1}) \Delta t_k / 2 \\ \boldsymbol{\zeta}_{t_{k-1/2}} = \boldsymbol{\omega}_{in}^n(t_{k-1}) \Delta t_k / 2 \end{cases} \quad (2.24)$$

式中, $\boldsymbol{\omega}_{ie}^e$ 、 $\boldsymbol{\omega}_{in}^n$ 分别为 e 系、n 系的旋转角速度:

$$\begin{cases} \boldsymbol{\omega}_{ie}^e = [0 \quad 0 \quad \omega_e]^T \\ \boldsymbol{\omega}_{in}^n = \boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n \end{cases} \quad (2.25)$$

高程的外推公式:

$$h_{t_{k-1/2}} = h_{t_{k-1}} - \frac{v_{D,t_{k-1}} \Delta t_k}{2} \quad (2.26)$$

速度的外推公式:

$$\mathbf{v}_{t_{k-1/2}}^n = \mathbf{v}_{t_{k-1}}^n + \frac{1}{2} \Delta \mathbf{v}_{t_{k-1}}^n = \mathbf{v}_{t_{k-1}}^n + \frac{1}{2} (\Delta \mathbf{v}_{f,t_{k-1}}^n + \Delta \mathbf{v}_{g/cor,t_{k-1}}^n) \quad (2.27)$$

$\Delta \mathbf{v}_{g/cor,t_k}^n$ 的计算公式:

$$\Delta \mathbf{v}_{g/cor,t_k}^n = [\mathbf{g}^n - (2\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n) \times \mathbf{v}^n]_{t_{k-1/2}} \Delta t_k \quad (2.28)$$

2.2.3 位置更新

本文对于位置更新仍可以通过旋转四元数 \mathbf{q}_n^e 得到, 公式如下:

$$\mathbf{q}_{n(t_k)}^{e(t_k)} = \mathbf{q}_{e(t_{k-1})}^{e(t_k)} \otimes \mathbf{q}_{n(t_{k-1})}^{e(t_{k-1})} \otimes \mathbf{q}_{n(t_k)}^{n(t_{k-1})} \quad (2.29)$$

$$\begin{cases} \mathbf{q}_{n(t_k)}^{n(t_{k-1})} = \begin{bmatrix} \cos \|0.5\boldsymbol{\zeta}_{t_k}\| \\ \frac{\sin \|0.5\boldsymbol{\zeta}_{t_k}\|}{\|\boldsymbol{\zeta}_{t_k}\|} \boldsymbol{\zeta}_{t_k} \end{bmatrix} \\ \mathbf{q}_{e(t_k)}^{e(t_{k-1})} = \begin{bmatrix} \cos \|0.5\boldsymbol{\xi}_{t_k}\| \\ -\frac{\sin \|0.5\boldsymbol{\xi}_{t_k}\|}{\|\boldsymbol{\xi}_{t_k}\|} \boldsymbol{\xi}_{t_k} \end{bmatrix} \end{cases} \quad (2.30)$$

式中, $\boldsymbol{\xi}_{t_k}$ 、 $\boldsymbol{\zeta}_{t_k}$ 分别代表 e 系、n 系从 t_{k-1} 到 t_k 时刻的旋转矢量。

将 \mathbf{q}_b^n 更新结束后, 根据式(2.21)可以得到更新后的载体的经纬度, 高程的更新可以用中间时刻的速度计算: $h_k = h_{k-1} - \mathbf{v}_{D,k-1/2} \Delta t_k$ 。

2.2.4 姿态更新

本文的姿态更新采用双子样姿态四元数更新算法, 形式如下:

$$\mathbf{q}_{b(k)}^{n(k)} = \mathbf{q}_{n(k-1)}^{n(k)} \otimes \mathbf{q}_{b(k-1)}^{n(k-1)} \otimes \mathbf{q}_{b(k)}^{b(k-1)} \quad (2.31)$$

$$\mathbf{q}_{n(k-1)}^{n(k)} = \begin{bmatrix} \cos \|0.5\boldsymbol{\zeta}_k\| \\ -\frac{\sin \|0.5\boldsymbol{\zeta}_k\|}{\|\boldsymbol{\zeta}_k\|} \boldsymbol{\zeta}_k \end{bmatrix} \quad (2.32)$$

刚体力学中, 有限转动具有不可交换的特点。所谓转动的不可交换性是指刚体的旋转顺序的不同会导致不同的结果, 这种特性说明了刚体转动不是矢量, 对转动的叠加是没有物理含义的。

在四元数算法中，用到了角速度矢量的积分，对于 \mathbf{b} 系，它可能发生剧烈的变化，使得旋转轴在积分过程中发生很大变化，使得角速度的方向可能在积分区间内是剧烈变化的，此时再对角速度进行积分是无意义的。如果继续和 \mathbf{n} 系、 \mathbf{e} 系类似的进行处理，就会有较大的不可交换误差。为了补偿不可交换误差，John E. Bortz 在 1971 年提出了等效旋转矢量^[24]概念，采用曲线拟合的方式对角速度矢量进行了补偿，符合实际转动，等效旋转矢量能够很好地代表角速度矢量，准确地反应其变化量，所以可以对等效旋转矢量进行积分，而不会产生不可交换误差。

等效旋转矢量的微分方程^[24]如下：

$$\dot{\Phi} = \omega + \frac{1}{2}\Phi \times \omega + \frac{1}{\Phi^2} \left[1 - \frac{\Phi \sin \Phi}{2(1 - \cos \Phi)} \right] \Phi \times (\Phi \times \omega) \quad (2.33)$$

简化式(2.33)便可得到近似的微分方程：

$$\dot{\Phi} = \omega + \frac{1}{2}\Phi \times \omega + \frac{1}{12}\Phi \times (\Phi \times \omega) \quad (2.34)$$

式中， ω 为转动角速度， Φ 是其对应的等效旋转矢量，使用 ω_{nb}^b 代替 ω ，就可以得到 \mathbf{b} 系的等效旋转矢量，其微分方程为：

$$\dot{\Phi} = \omega_{ib}^b + \frac{1}{2}\Phi \times \omega_{ib}^b + \frac{1}{12}\Phi \times (\Phi \times \omega_{ib}^b) \quad (2.35)$$

式(2.35)经过工程实践可以做以下近似：

$$\dot{\Phi} = \omega_{ib}^b + \frac{1}{2}\Delta\theta(t) \times \omega_{ib}^b \quad (2.36)$$

对式(2.36)积分可得：

$$\begin{aligned} \Phi_k &= \int_{t_{k-1}}^{t_k} \left[\omega_{ib}^b + \frac{1}{2}\Delta\theta(t) \times \omega_{ib}^b \right] dt \\ &\approx \Delta\theta_k + \frac{1}{12}\Delta\theta_{k-1} \times \Delta\theta_k \end{aligned} \quad (2.37)$$

其中， $\frac{1}{12}\Delta\theta_{k-1} \times \Delta\theta_k$ 是二阶圆锥误差项也可称为二子样圆锥误差项。此时，

\mathbf{b} 系的旋转四元素可以表示为：

$$\mathbf{q}_{b(k)}^{b(k-1)} = \begin{bmatrix} \cos\|0.5\mathbf{\Phi}_k\| \\ \frac{\sin\|0.5\mathbf{\Phi}_k\|}{\|\mathbf{\Phi}_k\|} \mathbf{\Phi}_k \end{bmatrix} \quad (2.38)$$

随着姿态更新计算次数的增加， \mathbf{q}_b^n 的模将逐渐不等于1，因此每次进行姿态更新或者进行了一定次数后，需要对 \mathbf{q}_b^n 进行归一化操作，计算方式为：

$$\begin{aligned} \mathbf{q}_b^n &= (1 - e_q) \mathbf{q}_b^n \\ e_q &= \frac{1}{2} [(\mathbf{q}_b^n)^T \mathbf{q}_b^n - 1] \end{aligned} \quad (2.39)$$

式中， e_q 代表归一化误差。

将 \mathbf{q}_b^n 更新结束后，根据 \mathbf{C}_b^n 与 \mathbf{q}_b^n 的如下关系^[22]，可以得到更新后的 \mathbf{C}_b^n ，进而根据式(2.1)得到载体更新后的姿态。

$$\mathbf{C}_b^n = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (2.40)$$

2.3 GNSS/INS 组合 Kalman 滤波

几乎所有的 GNSS/INS 组合导航都会估计位置、速度、姿态误差。加速度和角速度误差均可通过其带来的速度和姿态误差的增长来观测，两种误差都导致线性增加的速度误差和二次增加的位置误差，陀螺零偏短期内则会带来三次方的位置误差增长^[25]。

载体经历大的机动并在 GNSS 信息的辅助下可对 INS 航向误差进行观测和估计。当载体在水平面上匀速行驶或静止时，垂向加速度是带来垂向速度的唯一原因，此时垂向加表零偏是三个加表轴向中可观测性最强的一项，因此当 GNSS 信号失锁后垂向的位置误差漂移应该比水平位置误差漂移小很多。如果某项 IMU 误差对导航精度影响显著，它就会可观；相反，如果某项误差带来的影响远小于随机误差带来的影响，则这项误差不可观。可观测性其次的是陀螺零偏，因为它带来速度误差的平方增长和位置误差的三次方增长。在大部分的组合导航算法中均估计陀螺零偏，一般来说相比于其他两轴，垂向陀螺（航向陀螺）的可观测性比较弱。标度因子和交叉耦合误差只有在载体经历高动态的情况下才能被估计的较准确^[25]。

对于 GNSS 状态量的选取取决于组合的架构,在松组合中,GNSS 相关误差已经通过 GNSS 滤波器进行了处理,在组合导航处理中就不需要估计;在紧组合和深组合中,INS 和 GNSS 状态量在同一个滤波器中进行估计,接收机的钟差和钟漂需要被估计,系统结构复杂,但相对于松组合,优势在于只有一个滤波器,当卫星数目小于 4 个时,也可以利用所有的测量信息辅助 INS,限制导航误差的积累^[26]。本文组合导航系统设计导航精度为 cm 级,GNSS 需要使用 RTK(Real Time Kinematic)技术,鉴于 RTK 技术的复杂性,本文采用商业 GNSS 接收机直接提供 RTK 结果,组合导航采用松组合方案。

里程计辅助是一种车载导航辅助方法,可有效地抑制 GNSS 信号中断期间,INS 的误差发散。在实际使用中,由于轮胎在行驶过程中周长会有轻微变化,并且里程计的刻度也存在误差,使得里程计测量的速度需要一个比例因子来修正实际观测量,因此,里程计比例因子也应该被估计^[27]。

综合以上讨论,本文采用的 Kalman 滤波器状态量由 INS 的误差状态及其他辅助传感器的误差状态组成,状态参数 \mathbf{x} 有 21 个:位置误差 $\delta \mathbf{r}^n$ 、速度误差 $\delta \mathbf{v}^n$ 、姿态角误差 $\boldsymbol{\psi}$ 、陀螺零偏误差 $\delta \mathbf{b}_g$ 、加速度计零偏误差 $\delta \mathbf{b}_a$ 、陀螺比例因子误差 $\delta \mathbf{s}_g$ 、加速度计比例因子误差 $\delta \mathbf{s}_a$ 。对于使用里程计的模式,还需要增加 1 个里程计比例因子误差 δs_{odom} :

$$\mathbf{x} = [\delta \mathbf{r}^n \ \delta \mathbf{v}^n \ \boldsymbol{\psi} \ \delta \mathbf{b}_g \ \delta \mathbf{b}_a \ \delta \mathbf{s}_g \ \delta \mathbf{s}_a \ (\delta s_{odom})]^T \quad (2.41)$$

Kalman 滤波器的观测量由 INS 导航结果推算的位置与 GNSS 接收机解算的位置作差得到,利用 Kalman 滤波可以计算出上述状态量,将计算得到的惯导相关状态量反馈到机械编排中,从而抑制惯导的发散,如图 2.6 所示。

2.3.1 Kalman 滤波

Kalman 滤波是大部分导航算法的理论基础,广泛应用于卫星导航、卫星定轨、GNSS/INS 等多传感器组合导航领域,是利用多种传感器信息获得最优导航解的关键技术^[6]。利用 Kalman 滤波可以估计系统中的时变参数,在提供初始值后, Kalman 滤波递归地进行计算,将预测的先验信息和新的测量信息进行最优加权进行状态估计。

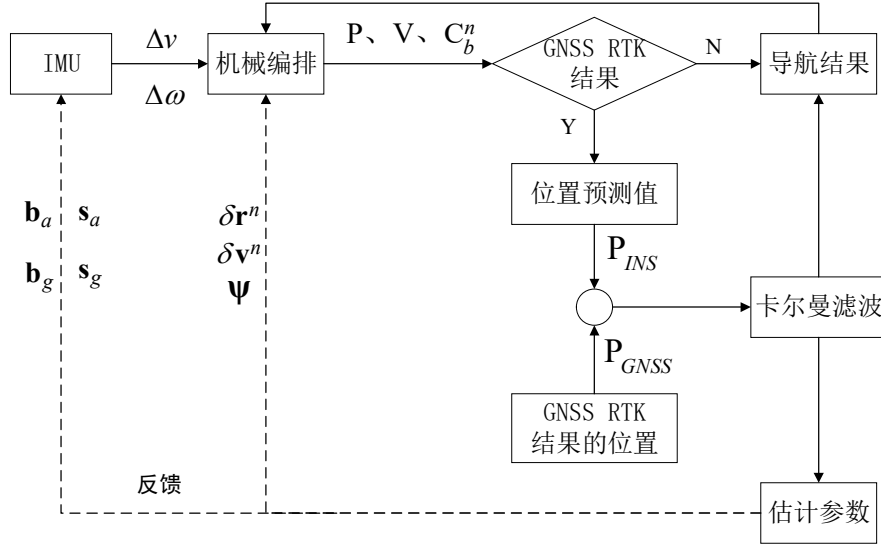


图 2.6 组合流程图

设随机线性的状态方程为：

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{G}(t)\mathbf{w}(t) \quad (2.42)$$

式中， \mathbf{F} 为动态矩阵， \mathbf{x} 为状态向量， \mathbf{G} 为系统噪声驱动矩阵， \mathbf{w} 为系统状态噪声。

式(2.42)的离散形式为：

$$\mathbf{x}_k = \Phi_{k,k-1}\mathbf{x}_{k-1} + \int_{t_{k-1}}^{t_k} \Phi_{k,\tau} \mathbf{G}(\tau)\mathbf{w}(\tau)d\tau \quad (2.43)$$

观测方程为：

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \quad (2.44)$$

式中， $\Phi_{k,k-1}$ 是状态转移矩阵， \mathbf{z}_k 为观测向量， \mathbf{H}_k 为设计矩阵， \mathbf{v}_k 是观测噪声。

假设系统状态噪声和观测噪声均为白噪声，设系统噪声方差阵为 \mathbf{Q}_{k-1} ，观测噪声方差阵为 \mathbf{R}_k 。

$$\begin{aligned} \mathbf{Q}_{k-1} &= E\left(\int_{t_{k-1}}^{t_k} \Phi_{k,\tau} \mathbf{G}(\tau)\mathbf{w}(\tau)d\tau \int_{t_{k-1}}^{t_k} \Phi_{k,t} \mathbf{G}(t)\mathbf{w}(t)dt\right) \\ &= \frac{1}{2} \left[\Phi_{k,k-1} \mathbf{G}_{k,k-1} \mathbf{q}_{k-1} \mathbf{G}_{k,k-1}^T + \mathbf{G}_{k,k-1} \mathbf{q}_{k-1} \mathbf{G}_{k,k-1}^T \Phi_{k,k-1}^T \right] (t_k - t_{k-1}) \end{aligned} \quad (2.45)$$

式中， \mathbf{q} 为 \mathbf{w} 的方差阵。

由于 GNSS/INS 组合的观测向量维数比较高，常规 Kalman 滤波需要的运算内存将会比较多，而嵌入式系统中的内存资源比较有限，并且观测向量维数越高，

计算增益矩阵时取逆的计算量就越大，因此，本文采用序贯 Kalman 滤波。序贯 Kalman 滤波对各分量组分散开来顺序处理，避免了对观测向量的集中处理，降低了内存的消耗量，同时也降低了取逆的计算量^[28]。

观测方程可以改写为：

$$\begin{bmatrix} \mathbf{z}_k^1 \\ \mathbf{z}_k^2 \\ \vdots \\ \mathbf{z}_k^r \end{bmatrix} = \begin{bmatrix} \mathbf{H}_k^1 \\ \mathbf{H}_k^2 \\ \vdots \\ \mathbf{H}_k^r \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} \mathbf{v}_k^1 \\ \mathbf{v}_k^2 \\ \vdots \\ \mathbf{v}_k^r \end{bmatrix} \quad (2.46)$$

则线性离散系统的序贯 Kalman 滤波的求解过程^[6]如下：

1) 预测

$$\hat{\mathbf{x}}_{k,k-1} = \Phi_{k,k-1} \hat{\mathbf{x}}_{k-1} \quad (2.47)$$

$$\mathbf{P}_{k,k-1} = \Phi_{k,k-1} \mathbf{P}_{k-1} \Phi_{k,k-1}^T + \mathbf{G}_{k,k-1} \mathbf{Q}_{k-1} \mathbf{G}_{k,k-1}^T \quad (2.48)$$

式中， $\hat{\mathbf{x}}_{k-1}$ 代表 \mathbf{x}_{k-1} 估计值， $\hat{\mathbf{x}}_{k,k-1}$ 表示 \mathbf{x}_k 的预测值。 \mathbf{P}_{k-1} 为 $\hat{\mathbf{x}}_{k-1}$ 的方差阵， $\mathbf{P}_{k,k-1}$ 为在 \mathbf{P}_{k-1} 的基础上考虑到系统噪声阵是影响都得到的预测方差阵，即 $\hat{\mathbf{x}}_{k,k-1}$ 的方差阵。

2) 更新

$$\mathbf{K}_k^i = \mathbf{P}_k^{i-1} \mathbf{H}_k^{iT} (\mathbf{H}_k^i \mathbf{P}_k^{i-1} \mathbf{H}_k^{iT} + \mathbf{R}_k^i)^{-1} \quad (2.49)$$

$$\hat{\mathbf{x}}_k^i = \hat{\mathbf{x}}_k^{i-1} + \mathbf{K}_k^i (\mathbf{z}_k^i - \mathbf{H}_k^i \hat{\mathbf{x}}_k^{i-1}) \quad (2.50)$$

$$\mathbf{P}_k^i = (\mathbf{I} - \mathbf{K}_k^i \mathbf{H}_k^i) \mathbf{P}_k^{i-1} (\mathbf{I} - \mathbf{K}_k^i \mathbf{H}_k^i)^T + \mathbf{K}_k^i \mathbf{R}_k^i \mathbf{K}_k^{iT} \quad (2.51)$$

其中， $i = 1, 2, 3, \dots, r$ ， $\hat{\mathbf{x}}_k^0 = \hat{\mathbf{x}}_{k,k-1}$ ， $\mathbf{P}_k^0 = \mathbf{P}_{k,k-1}$ 。

式中， \mathbf{K}_k^i 为增益矩阵， \mathbf{R}_k^i 为观测噪声方差协方差阵， $\hat{\mathbf{x}}_k^r$ 即为 k 时刻序贯更新后的状态估计量。式(2.51)为 Joseph 形式的 \mathbf{P} 阵更新方程，可以保证 \mathbf{P} 阵在计算机有舍入误差的情况下仍然保证为对称阵^[29]。

2.3.2 状态方程

Kalman 滤波中的状态方程描述了状态量随着时间的变化关系，面对各种各样的使用场景，可以选择多种不同误差模型来描述。本文的应用场景为中低端惯

性导航，倾向使用 Psi 角误差模型。Psi 角误差模型的导航计算是在 c 系中进行的，因此，对 c 系进行误差扰动分析后，保留一阶小量，忽略高阶量便可得到误差方程，进一步得到 Kalman 滤波的状态方程。

对导航参数进行如下扰动：

$$\begin{aligned}\hat{\mathbf{v}}^n &= \mathbf{v}^n + \delta \mathbf{v}^n = \mathbf{v}^c + \delta \mathbf{v}^c \\ \hat{\mathbf{g}}^n &= \mathbf{g}^n + \delta \mathbf{g}^n = \mathbf{g}^c + \delta \mathbf{g}^c \\ \hat{\boldsymbol{\omega}}_{ie}^n &= \boldsymbol{\omega}_{ie}^n + \delta \boldsymbol{\omega}_{ie}^n = \boldsymbol{\omega}_{ie}^c \\ \hat{\boldsymbol{\omega}}_{in}^n &= \boldsymbol{\omega}_{in}^n + \delta \boldsymbol{\omega}_{in}^n = \boldsymbol{\omega}_{ic}^c\end{aligned}\quad (2.52)$$

将式(2.3)代入式(2.52)中可得：

$$\begin{aligned}\delta \mathbf{v}^n &= \delta \mathbf{v}^c - \delta \boldsymbol{\theta} \times \mathbf{v}^c \\ \delta \mathbf{g}^n &= \delta \mathbf{g}^c - \delta \boldsymbol{\theta} \times \mathbf{g}^c \\ \delta \boldsymbol{\omega}_{ie}^n &= -\delta \boldsymbol{\theta} \times \boldsymbol{\omega}_{ie}^c\end{aligned}\quad (2.53)$$

n 系中的重力误差模型可以简单的表示为：

$$\delta \mathbf{g}^n = [0 \quad 0 \quad 2g\delta r_D / (R+h)]^T \quad (2.54)$$

由式(2.53)、式(2.54)和式(2.4)可以得到 c 系中的重力误差模型：

$$\delta \mathbf{g}^c = \begin{bmatrix} -g\delta r_N & -g\delta r_E & 2g\delta r_D \\ R_M + h & R_N + h & R + h \end{bmatrix}^T \quad (2.55)$$

式中， $R = \sqrt{R_M R_N}$ ， g 为重力值。

1) 速度误差方程

根据式(2.9)可以得到 c 系的速度微分方程：

$$\dot{\mathbf{v}}^c + (2\boldsymbol{\omega}_{ie}^c + \boldsymbol{\omega}_{ec}^c) \times \mathbf{v}^c - \mathbf{g}^c = \mathbf{f}^c \quad (2.56)$$

其中， \mathbf{v}^c 代表 c 系中的地速。由于 $\boldsymbol{\omega}_{ie}^c$ 和 $\boldsymbol{\omega}_{ec}^c$ 是没有误差的，所以：

$$\hat{\dot{\mathbf{v}}}^c + (2\boldsymbol{\omega}_{ie}^c + \boldsymbol{\omega}_{ec}^c) \times \hat{\mathbf{v}}^c - \hat{\mathbf{g}}^c = \hat{\mathbf{f}}^c \quad (2.57)$$

使用 $\mathbf{f}^p + \delta \mathbf{f}^p$ 替换 $\hat{\mathbf{f}}^c$ ，得：

$$\dot{\mathbf{v}}^c + \delta \dot{\mathbf{v}}^c + (2\boldsymbol{\omega}_{ie}^c + \boldsymbol{\omega}_{ec}^c) \times (\mathbf{v}^c + \delta \mathbf{v}^c) - (\mathbf{g}^c + \delta \mathbf{g}^c) = \mathbf{f}^p + \delta \mathbf{f}^p \quad (2.58)$$

由式(2.6)可得：

$$\mathbf{f}^p = \mathbf{C}_c^p \mathbf{f}^c = \mathbf{f}^c - \boldsymbol{\psi} \times \mathbf{f}^c \quad (2.59)$$

将式(2.56)、式(2.59)代入(2.58)得到最终的速度误差方程：

$$\delta \dot{\mathbf{v}}^c = \mathbf{f}^c \times \boldsymbol{\Psi} - (2\boldsymbol{\omega}_{ie}^c + \boldsymbol{\omega}_{ec}^c) \times \delta \mathbf{v}^c + \delta \mathbf{g}^c + \mathbf{C}_b^p \delta \mathbf{f}^b \quad (2.60)$$

2) 姿态误差方程

\mathbf{C}_b^n 的微分方程如下^[30]：

$$\dot{\mathbf{C}}_b^n = \mathbf{C}_b^n (\boldsymbol{\omega}_{nb}^b \times) \quad (2.61)$$

对式(2.61)进行扰动可以得到：

$$\dot{\hat{\mathbf{C}}}_b^n = \hat{\mathbf{C}}_b^n (\hat{\boldsymbol{\omega}}_{nb}^b \times) \quad (2.62)$$

由式(2.5)可得：

$$\hat{\mathbf{C}}_b^n = \mathbf{C}_b^p = [\mathbf{I} - (\boldsymbol{\Psi} \times)] \mathbf{C}_b^c \quad (2.63)$$

令 $(\boldsymbol{\omega}_{ab}^c \times) = \boldsymbol{\Omega}_{ab}^c$ ，将式(2.63)代入式(2.62)可得：

$$-(\dot{\boldsymbol{\Psi}} \times) \mathbf{C}_b^c + [\mathbf{I} - (\boldsymbol{\Psi} \times)] \dot{\mathbf{C}}_b^c = [\mathbf{I} - (\boldsymbol{\Psi} \times)] \mathbf{C}_b^c (\boldsymbol{\Omega}_{ib}^b - \boldsymbol{\Omega}_{in}^b + \delta \boldsymbol{\Omega}_{ib}^b - \delta \boldsymbol{\Omega}_{in}^b) \quad (2.64)$$

忽略 c 系、n 系的相互转动，式(2.64)可转化为：

$$\begin{aligned} -(\dot{\boldsymbol{\Psi}} \times) \mathbf{C}_b^c + [\mathbf{I} - (\boldsymbol{\Psi} \times)] \dot{\mathbf{C}}_b^c &= [\mathbf{I} - (\boldsymbol{\Psi} \times)] \mathbf{C}_b^c (\boldsymbol{\Omega}_{ib}^b - \boldsymbol{\Omega}_{ic}^b + \delta \boldsymbol{\Omega}_{ib}^b - \delta \boldsymbol{\Omega}_{in}^b) \\ &= [\mathbf{I} - (\boldsymbol{\Psi} \times)] \mathbf{C}_b^c (\boldsymbol{\Omega}_{cb}^b + \delta \boldsymbol{\Omega}_{ib}^b - \delta \boldsymbol{\Omega}_{in}^b) \end{aligned} \quad (2.65)$$

\mathbf{C}_b^c 的微分方程可表示为：

$$\dot{\mathbf{C}}_b^c = \mathbf{C}_b^c (\boldsymbol{\omega}_{cb}^b \times) = \mathbf{C}_b^c (\boldsymbol{\Omega}_{cb}^b) \quad (2.66)$$

将式(2.66)代入式(2.65)，忽略二阶小量，可得：

$$(\dot{\boldsymbol{\Psi}} \times) = -\mathbf{C}_b^c (\delta \boldsymbol{\Omega}_{ib}^b - \delta \boldsymbol{\Omega}_{in}^b) \mathbf{C}_c^b \quad (2.67)$$

将式(2.67)写成向量形式：

$$\dot{\boldsymbol{\Psi}} = -\mathbf{C}_b^c (\delta \boldsymbol{\omega}_{ib}^b - \delta \boldsymbol{\omega}_{in}^b) \quad (2.68)$$

为了得到 $\delta \boldsymbol{\omega}_{in}^b$ ，需要根据 $\hat{\boldsymbol{\omega}}_{in}^b = \hat{\mathbf{C}}_n^b \hat{\boldsymbol{\omega}}_{in}^n$ 展开，可得：

$$\boldsymbol{\omega}_{in}^b + \delta \boldsymbol{\omega}_{in}^b = \mathbf{C}_c^b [\mathbf{I} + (\boldsymbol{\Psi} \times)] (\boldsymbol{\omega}_{in}^n + \delta \boldsymbol{\omega}_{in}^n) \quad (2.69)$$

对上式忽略二阶小量化简可得：

$$\delta \boldsymbol{\omega}_{in}^b = \mathbf{C}_c^b [\delta \boldsymbol{\omega}_{in}^n + (\boldsymbol{\Psi} \times) \boldsymbol{\omega}_{in}^n] \quad (2.70)$$

将式(2.70)代入式(2.68)可得：

$$\dot{\boldsymbol{\psi}} = \delta \boldsymbol{\omega}_{in}^n - (\boldsymbol{\omega}_{in}^n \times) \boldsymbol{\psi} - \mathbf{C}_b^c \delta \boldsymbol{\omega}_{ib}^b \quad (2.71)$$

对式(2.71)同样地忽略 c 系、n 系的相互转动，考虑到 $\delta \boldsymbol{\omega}_{ib}^p = \delta \boldsymbol{\omega}_{ib}^c = \delta \boldsymbol{\omega}_{ib}^n$ ，姿态误差方程最终表示为：

$$\begin{aligned} \dot{\boldsymbol{\psi}} &= -\boldsymbol{\omega}_{ic}^c \times \boldsymbol{\psi} - \mathbf{C}_b^c \delta \boldsymbol{\omega}_{ib}^b \\ &= -\boldsymbol{\omega}_{ic}^c \times \boldsymbol{\psi} - [\mathbf{I} + (\boldsymbol{\psi} \times)] \mathbf{C}_b^p \delta \boldsymbol{\omega}_{ib}^b \\ &= -\boldsymbol{\omega}_{ic}^c \times \boldsymbol{\psi} - \mathbf{C}_b^p \delta \boldsymbol{\omega}_{ib}^b \end{aligned} \quad (2.72)$$

3) 位置误差方程

载体的位置可以由载体的速度积分得到， t_k 时刻载体的位置可以表示为：

$$\hat{\mathbf{r}}^e = \int_0^{t_k} \mathbf{C}_c^e \hat{\mathbf{v}}_t^c dt \quad (2.73)$$

对上式做扰动分析， $\hat{\mathbf{r}}^e = \mathbf{r}^e + \delta \mathbf{r}^e$ ， $\hat{\mathbf{v}}^c = \mathbf{v}^c + \delta \mathbf{v}^c$ ，可得：

$$\delta \mathbf{r}^e = \int_0^{t_k} \mathbf{C}_c^e \delta \mathbf{v}^c dt \quad (2.74)$$

将上式中的位置误差投影到 c 系中，得：

$$\delta \mathbf{r}^c = \mathbf{C}_e^c \delta \mathbf{r}^e = \mathbf{C}_e^c \int_0^{t_k} \mathbf{C}_c^e \delta \mathbf{v}^c dt \quad (2.75)$$

对式(2.75)进行求导，同时考虑到 $\dot{\mathbf{C}}_e^c = \mathbf{C}_e^c (\boldsymbol{\omega}_{ce}^e \times)$ ，可得：

$$\begin{aligned} \delta \dot{\mathbf{r}}^c &= \left(\mathbf{C}_e^c \int_0^{t_k} \mathbf{C}_c^e \delta \mathbf{v}^c dt \right)' \\ &= \dot{\mathbf{C}}_e^c \int_0^{t_k} \mathbf{C}_c^e \delta \mathbf{v}^c dt + \delta \mathbf{v}^c \\ &= \mathbf{C}_e^c (\boldsymbol{\omega}_{ce}^e \times) \mathbf{C}_e^c \delta \mathbf{r}^e + \delta \mathbf{v}^c \\ &= \left[(\mathbf{C}_e^c \boldsymbol{\omega}_{ce}^e) \times \right] \delta \mathbf{r}^c + \delta \mathbf{v}^c \\ &= -\boldsymbol{\omega}_{ec}^c \times \delta \mathbf{r}^c + \delta \mathbf{v}^c \end{aligned} \quad (2.76)$$

4) 传感器误差方程

陀螺仪、加速度计的观测误差可如下表示：

$$\begin{aligned} \delta \boldsymbol{\omega}_{ib}^b &= \delta \mathbf{b}_g + \text{diag}(\boldsymbol{\omega}_{ib}^b) \delta \mathbf{s}_g + \mathbf{w}_g \\ \delta \mathbf{f}^b &= \delta \mathbf{b}_a + \text{diag}(\mathbf{f}^b) \delta \mathbf{s}_a + \mathbf{w}_a \end{aligned} \quad (2.77)$$

$\mathbf{w}_g, \mathbf{w}_a$ 分别表示陀螺仪、加速度计观测噪声。

本文采用一阶高斯马尔科夫过程对陀螺仪、加速度计的零偏以及比例因子进

行建模^[22]:

$$\dot{x} = -\frac{1}{\tau}x + w \quad (2.78)$$

式中, τ 是相关时间, w 是驱动白噪声。

5) 里程计误差方程

里程计的比例因子可按随机常数或者具有很小的驱动白噪声的随机游走建模^[22], 本文使用随机游走建模:

$$\dot{\delta s}_{odom} = \eta_{odom} \quad (2.79)$$

式中, η_{odom} 为里程计比例因子驱动白噪声。

2.3.3 观测方程

由于 GNSS 板卡输出的位置是天线相位中心, 而惯导计算的位置是 IMU 中心, 所以在建立观测方程时, 需要将二者计算的位置归算到同一个点, 也就是要进行杆臂补偿。天线相位中心与 IMU 中心在 n 系的位置 \mathbf{r}_{GNSS}^n 、 \mathbf{r}_{INS}^n 存在如下关系:

$$\mathbf{r}_{GNSS}^n = \mathbf{r}_{INS}^n + \mathbf{D}_R^{-1} \mathbf{C}_b^n \mathbf{I}_{GNSS}^b \quad (2.80)$$

其中, $\mathbf{D}_R^{-1} = \text{diag}(1/(M+h), 1/[(N+h)\cos\varphi], -1)$, \mathbf{I}_{GNSS}^b 为天线杆臂, 即 b 系中 IMU 中心到天线相位中心矢量。

根据惯导计算的位置可以得到的估算的天线相位中心位置:

$$\begin{aligned} \hat{\mathbf{r}}_{GNSS}^n &= \hat{\mathbf{r}}_{INS}^n + \mathbf{D}_R^{-1} \hat{\mathbf{C}}_b^n \mathbf{I}_{GNSS}^b \\ &= \mathbf{r}_{INS}^n + \mathbf{D}_R^{-1} \delta \mathbf{r}^n + \mathbf{D}_R^{-1} [\mathbf{I} - (\boldsymbol{\varphi} \times)] \mathbf{C}_b^n \mathbf{I}_{GNSS}^b \\ &= \mathbf{r}_{GNSS}^n + \mathbf{D}_R^{-1} \delta \mathbf{r}^n + \mathbf{D}_R^{-1} [(\mathbf{C}_b^n \mathbf{I}_{GNSS}^b) \times] \boldsymbol{\varphi} \\ &= \mathbf{r}_{GNSS}^n + \mathbf{D}_R^{-1} \delta \mathbf{r}_{GNSS}^n \end{aligned} \quad (2.81)$$

由上式可以得位置辅助的观测方程为:

$$\delta \mathbf{r}_{GNSS}^n = \delta \mathbf{r}^n + [(\mathbf{C}_b^n \mathbf{I}_{GNSS}^b) \times] (\boldsymbol{\psi} + \delta \boldsymbol{\theta}) \quad (2.82)$$

2.3.4 观测噪声和状态噪声方差阵的设置

1) 初始状态量协方差阵

初始状态量方差阵是根据 IMU 的性能指标以及初始对准完成后的位置、速度、姿态误差确定的。

2) 状态噪声方差阵

状态噪声方差阵描述了 Kalman 滤波系统在系统噪声的影响下，状态预测估计的不确定性随时间增长的规律，其与 IMU 设备的性能指标密切相关，可表示为^[16]：

$$Q = \text{diag}\{ARW^2, ARW^2, ARW^2, VRW^2, VRW^2, VRW^2, Q_{bg}, Q_{bg}, Q_{bg}, Q_{ba}, Q_{ba}, Q_{ba}, Q_{sg}, Q_{sg}, Q_{sg}, Q_{sa}, Q_{sa}, Q_{sa}\} \quad (2.83)$$

其中， $Q_{bg} = 2\sigma_{bg}^2 / \tau_{bg}$, $Q_{ba} = 2\sigma_{ba}^2 / \tau_{ba}$, $Q_{sg} = 2\sigma_{sg}^2 / \tau_{sg}$, $Q_{sa} = 2\sigma_{sa}^2 / \tau_{sa}$ 。

式中，ARW 是角度随机随机游走，VRW 是速度随机游走， σ_{bg} 是陀螺零偏稳定性， σ_{ba} 是加速度计零偏稳定性， σ_{sg} 是陀螺比例因子标准差， σ_{sa} 是加速度计比例因子标准差，以上 IMU 性能参数都由 IMU 厂商提供， τ 是相关时间，通常低精度 IMU 的相关时间一般小于 1h，精度较高的 IMU 的相关时间通常在 2~4h。

3) 观测噪声方差阵

根据 GNSS 解算的位置精度进行确定：

$$\mathbf{R} = \begin{bmatrix} Bstd^2 & 0 & 0 \\ 0 & Lstd^2 & 0 \\ 0 & 0 & hstd^2 \end{bmatrix} \quad (2.84)$$

式中，Bstd、Lstd、hstd 分别为 GNSS 解算的载体纬度、经度、大地高的标准差。

2.3.5 状态估计的反馈

本文采用闭环 Kalman 滤波器，经过观测更新后的状态参数估计值，需反馈到当前导航状态中，以修正导航状态，如图 2.6 所示。反馈修正包括位置、速度、姿态、陀螺仪和加速度计的零偏、比例因子以及里程计比例因子的反馈。

1) 位置反馈

由式(2.21)可知，载体的经度、纬度可以由 \mathbf{q}_n^e 得到，因此水平位置反馈如下进行：

$$\mathbf{q}_n^e = \mathbf{q}_c^e \otimes \mathbf{q}_n^c \quad (2.85)$$

$$\mathbf{q}_n^c = \begin{bmatrix} \cos\|0.5\delta\boldsymbol{\theta}\| \\ -\frac{\sin\|0.5\delta\boldsymbol{\theta}\|}{\|0.5\delta\boldsymbol{\theta}\|} 0.5\delta\boldsymbol{\theta} \end{bmatrix} \quad (2.86)$$

高程的反馈如下：

$$h = \hat{h} + \delta r_D \quad (2.87)$$

2) 速度反馈

$$\mathbf{v}^n = \hat{\mathbf{v}}^n - \delta \mathbf{v}^n \quad (2.88)$$

使用 Psi 角误差模型时，估计的速度误差为 c 系，c 系与 n 的速度误差存在以下关系^[22]：

$$\delta \mathbf{v}^n = \delta \mathbf{v}^c - \delta \boldsymbol{\theta} \times \hat{\mathbf{v}}^n \quad (2.89)$$

3) 姿态反馈

由于 $\hat{\mathbf{q}}_b^n = \mathbf{q}_b^p$ ，载体姿态可由 \mathbf{q}_b^n 得到，因此姿态反馈如下进行：

$$\mathbf{q}_b^n = \mathbf{q}_p^n \otimes \mathbf{q}_b^p \quad (2.90)$$

$$\mathbf{q}_p^n = \begin{bmatrix} \cos\|0.5\hat{\boldsymbol{\phi}}\| \\ \frac{\sin\|0.5\hat{\boldsymbol{\phi}}\|}{\|0.5\hat{\boldsymbol{\phi}}\|} 0.5\hat{\boldsymbol{\phi}} \end{bmatrix} \quad (2.91)$$

4) 里程计比例因子反馈

$$s_{odom} = \hat{s}_{odom} + \delta s_{odom} \quad (2.92)$$

5) 陀螺仪和加速度计的零偏、比例因子反馈

$$\begin{aligned} \mathbf{b} &= \hat{\mathbf{b}} + \delta \mathbf{b} \\ \mathbf{s} &= \hat{\mathbf{s}} + \delta \mathbf{s} \end{aligned} \quad (2.93)$$

2.4 GNSS/INS 组合导航抗差处理

由于很多不确定的因素，导航系统偶尔会出现远超过限定误差的情况，如硬件、软件故障，观测条件恶劣等。在组合导航系统中，利用惯导短期内精度较高的特点，使用惯导计算的导航结果探测观测数据中粗差，对粗差观测数据进行剔除或降权处理来起到抗差的目的。

本文采样新息滤波的方式进行抗差处理^[25]，Kalman 滤波器的新息给出了观测与状态估计是否一致的指示，新息滤波可以迅速检测大的异常，并且新息及其方差作为 Kalman 滤波正常计算的一部分，不会带来额外的计算量^[31]。

Kalman 滤波器的新息为：

$$\delta \mathbf{z}_k^- = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^- \quad (2.94)$$

新息的方差阵包含了观测噪声以及状态估计的的方差阵，也是增益矩阵的分母，可表示为：

$$\mathbf{C}_k = \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k \quad (2.95)$$

对计算得到的新息与其对应的方差进行对比，并剔除超过阈值的观测量。

$$\left| (\delta \mathbf{z}_k^-)_i \right| \leq r \sqrt{(\mathbf{C}_k)_{i,i}} \quad (2.96)$$

R 取 2 或 3 对应的置信度分别为 95%、99.73%。

2.5 车辆约束

在车载模式下，使用零速修正（Zero Velocity Update, ZUPT）、非完整性约束（Non-Holonomic, NHC）以及里程计辅助可以增加量测观测，得到更优的导航结果。特别是在城市环境中，卫星信息质量无法保障，GNSS 定位时常无法用于惯导的修正，而上述车辆约束不受卫星信号的影响，能连续提供辅助信息以修正惯导误差，保障导航精度。

2.5.1 IMU 安装角估计

惯性导航的结果是以 **b** 系为基准的，由于 IMU 和车辆之间存在安装差异，轴线并不完全重合，而车载约束是在车辆坐标系 **v** 系下进行的。定义 **v** 系的坐标原点在车辆的重心处，X 轴沿车体纵轴指向前，Y 轴沿车体横轴指向右，Z 轴与 X 轴和 Y 轴构成右手坐标系。安装角描述的就是 **v** 系到 **b** 系的旋转姿态角。

通常安装误差角可以由测量或计算估计进行标定，认为在设备安装关系不变的情况下，安装误差角固定不变。测量标定的主要方法是使用全站仪测量 IMU 以及车辆的轴线上若干点的坐标，得到相同点在 **b** 系和 **v** 系的两套坐标值，进而通过最小二乘方法估计出 **b** 系与 **v** 系的相对角度关系，即安装误差角。但测量

标定的方式操作比较繁琐，车辆以及 IMU 的轴线也无法精确得到，因此，本文采用计算估计的方式进行安装误差角标定，主要方法是采集 GNSS 信号良好环境中的一段组合导航数据，通过不使用车辆辅助的后处理组合导航算法，可以得到精度较高的 b 系速度，v 系速度与 b 系速度可以由安装误差角建立联系，类似 n 系与 b 系的关系式(2.1)，并且考虑到安装误差角一般较小，进行小角度近似，可以得到如下关系式：

$$\begin{bmatrix} v_x^v \\ v_y^v \\ v_z^v \end{bmatrix} = \begin{bmatrix} 1 & -\psi & \theta \\ \psi & 1 & -\gamma \\ -\theta & \gamma & 1 \end{bmatrix} \begin{bmatrix} v_x^b \\ v_y^b \\ v_z^b \end{bmatrix} \quad (2.97)$$

根据上式，利用 v 系速度在 y、z 方向的速度为零以及 b 系速度，得到求解安装误差角的方程式(2.98)，可以看出该方程秩亏，可由多个历元的数据通过最小二次估计解算安装误差角^[32]。

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -v_z^b & 0 & v_x^b \\ v_y^b & -v_x^b & 0 \end{bmatrix} \begin{bmatrix} \gamma \\ \theta \\ \psi \end{bmatrix} + \begin{bmatrix} v_y^b \\ v_z^b \end{bmatrix} \quad (2.98)$$

2.5.2 ZUPT 辅助

载体处于静止状态时，载体速度始终为零，此时使用零速为条件形成观测方程，就相当于有了精度极高的速度观测值来抑制惯导误差。具体观测方程如下：

$$\mathbf{z} = \mathbf{v}_{IMU}^n - \mathbf{0}_{3 \times 1} = \delta \mathbf{v}^c \quad (2.99)$$

观测噪声设为常数，本文中设定为 0.1m/s。

对于载体静止状态的判断，本文在里程计辅助模式下，使用里程计的输出来进行判断，认为载体静止时，里程计输出脉冲增量为零；没有里程计辅助时，使用惯性器件的加速度、角速度输出的 RMS 进行判断，认为载体静止时，加速度、角速度的输出很稳定且接近零，RMS 应小于阈值^[33]。

2.5.3 里程计辅助

在车载导航中，把里程计与车辆非动力轮的轮轴固连，便可利用里程计来获取车辆的前进方向的速率，用于辅助惯性导航。车轮处的速度与惯性导航的速度存在如下关系：

$$\mathbf{v}_{wheel}^v = \mathbf{C}_b^v \mathbf{C}_n^b \mathbf{v}_{IMU}^n + \mathbf{C}_b^v (\boldsymbol{\omega}_{nb}^b \times) \mathbf{I}_{wheel}^b \quad (2.100)$$

式中, \mathbf{v}_{wheel}^v 为车轮处的速度, \mathbf{I}_{wheel}^b 为里程计杆臂, 即 b 系中 IMU 中心到车轮与地面切点的矢量。

根据上式, 由 IMU 推算的速度 $\hat{\mathbf{v}}_{IMU}^n$ 可以得到车轮的估计速度 $\hat{\mathbf{v}}_{wheel}^v$:

$$\begin{aligned} \hat{\mathbf{v}}_{wheel}^v &= \mathbf{C}_b^v \hat{\mathbf{C}}_n^b \hat{\mathbf{v}}_{IMU}^n + \mathbf{C}_b^v (\hat{\boldsymbol{\omega}}_{nb}^b \times) \mathbf{I}_{wheel}^b \\ &\approx \mathbf{C}_b^v \mathbf{C}_n^b [\mathbf{I} + (\boldsymbol{\phi} \times)] (\mathbf{v}_{IMU}^n + \delta \mathbf{v}^n) + \mathbf{C}_b^v (\boldsymbol{\omega}_{nb}^b \times) \mathbf{I}_{wheel}^b + \\ &\quad \mathbf{C}_b^v (\delta \boldsymbol{\omega}_{ib}^b \times) \mathbf{I}_{wheel}^b \\ &\approx \mathbf{v}_{wheel}^v + \mathbf{C}_b^v \mathbf{C}_n^b \delta \mathbf{v}^n - \mathbf{C}_b^v \mathbf{C}_n^b (\mathbf{v}_{IMU}^n \times) \boldsymbol{\phi} - \mathbf{C}_b^v (\mathbf{I}_{wheel}^b \times) \delta \boldsymbol{\omega}_{ib}^b \end{aligned} \quad (2.101)$$

车轮处的速度还可以表示为:

$$\tilde{\mathbf{v}}_{wheel}^v = \mathbf{v}_{wheel}^v + \mathbf{e}_v \quad (2.102)$$

式(2.101)减去式(2.102)得到如下观测方程:

$$\begin{aligned} \mathbf{z} &= \hat{\mathbf{v}}_{wheel}^v - \tilde{\mathbf{v}}_{wheel}^v \\ &= \mathbf{C}_b^v \mathbf{C}_n^b \delta \mathbf{v}^n - \mathbf{C}_b^v \mathbf{C}_n^b (\mathbf{v}_{IMU}^n \times) \boldsymbol{\phi} - \mathbf{C}_b^v (\mathbf{I}_{wheel}^b \times) \delta \boldsymbol{\omega}_{ib}^b - \mathbf{e}_v \end{aligned} \quad (2.103)$$

\mathbf{v}_{IMU}^n 近似为 \mathbf{v}_{IMU}^c 可以得到:

$$\begin{aligned} \mathbf{z} &= \mathbf{C}_b^v \mathbf{C}_n^b \delta \mathbf{v}^n - \mathbf{C}_b^v \mathbf{C}_n^b (\mathbf{v}_{IMU}^c \times) (\boldsymbol{\psi} + \delta \boldsymbol{\theta}) - \mathbf{C}_b^v (\mathbf{I}_{wheel}^b \times) \delta \boldsymbol{\omega}_{ib}^b - \mathbf{e}_v \\ &= \mathbf{C}_b^v \mathbf{C}_n^b (\delta \mathbf{v}^n - \mathbf{v}_{IMU}^c \times \delta \boldsymbol{\theta}) - \mathbf{C}_b^v \mathbf{C}_n^b (\mathbf{v}_{IMU}^c \times) \boldsymbol{\psi} - \mathbf{C}_b^v (\mathbf{I}_{wheel}^b \times) \delta \boldsymbol{\omega}_{ib}^b - \mathbf{e}_v \\ &= \mathbf{C}_b^v \mathbf{C}_n^b (\delta \mathbf{v}^c) - \mathbf{C}_b^v \mathbf{C}_n^b (\mathbf{v}_{IMU}^c \times) \boldsymbol{\psi} - \mathbf{C}_b^v (\mathbf{I}_{wheel}^b \times) \delta \boldsymbol{\omega}_{ib}^b - \mathbf{e}_v \end{aligned} \quad (2.104)$$

里程计测量的速率为 $\tilde{\mathbf{v}}_{wheel}^v$ 的 x 轴分量, 考虑到里程计测量的比例因子误差,

式(2.102)的 x 分量部分可以改写为:

$$\tilde{v}_{odom}^v = (1 + \delta s_{odom}) v_{odom}^v + e_v \quad (2.105)$$

式中, v_{odom}^v 为车轮速度的 x 分量, δs_{odom} 为里程计比例因子误差。

式(2.101)左成 $\mathbf{A} = [1 \ 0 \ 0]$ 得 x 分量, 然后减去式(2.105)可以得到里程计速度观测方程:

$$\begin{aligned} z &= \hat{v}_{odom}^v - \tilde{v}_{odom}^v \\ &= \mathbf{A} \mathbf{C}_b^v \mathbf{C}_n^b \delta \mathbf{v}^n - \mathbf{A} \mathbf{C}_b^v \mathbf{C}_n^b (\mathbf{v}_{IMU}^n \times) \boldsymbol{\phi} - \mathbf{A} \mathbf{C}_b^v (\mathbf{I}_{wheel}^b \times) \delta \boldsymbol{\omega}_{ib}^b \\ &\quad - v_{odom}^v \delta s_{odom} - e_v \end{aligned} \quad (2.106)$$

观测噪声阵的设置需要考虑里程计分辨率和车轮周长，本文中使用 0.1m/s。

2.5.4 NHC 辅助

车辆在行驶过程中，认为非转向非动力轮不会发生侧向移动和垂直运动^[34]，即 $\tilde{\mathbf{v}}_{wheel}^v$ 的 y、z 轴分量为零。NHC 辅助就是使用这 2 个条件形成的观测方程，具体方程可以由式(2.104)得到，观测噪声阵为常数，本文使用 0.1m/s。

2.5.5 里程计杆臂与 IMU 安装误差角的影响

由上述车辆约束的观测方程可以看出，在使用车辆约束的情况下，应当正确确定里程计杆臂和 IMU 安装误差角。本小节使用 2015.3.27 采集的一组数据进行后处理分析里程计杆臂和 IMU 安装误差角的影响。数据处理中，参考真值与两种对比方案均使用里程计和非完整性约束辅助，参考真值为该组数据的反向平滑处理结果，两种对比方案均采用正向滤波处理结果，并在车辆转弯处进行仿真卫星信号中断处理，每次中断时间为 1 分钟，区别为一个方案正确使用里程计杆臂、IMU 安装误差角（方案 1），另一个方案将里程计杆臂、IMU 安装误差角都设置为零（方案 2），导航误差曲线分别如图 2.7、图 2.8 所示。从表 2.1 可以看出，方案 2 的导航误差明显比方案 1 大，北、东、地方向的位置误差 RMS 分别从 1.020、1.474、0.346m 增大到了 2.749、1.980、5.484m，北、东、地方向速度误差 RMS 分别从 0.052、0.070、0.013m/s 增大到了 0.102、0.102、0.142m/s，航向的误差 RMS 从 0.073deg 增大到了 0.234deg，而且造成了较大的常值偏差，这是由于 NHC 错误地约束航向导致的，横滚和俯仰没有太大影响。

表 2.1 仿真中断期间导航误差统计表

误差项		方案 1			方案 2		
		最大值	均值	RMS	最大值	均值	RMS
位置 误差 (m)	北向	2.076	0.823	1.020	3.690	2.670	2.749
	东向	1.994	1.369	1.474	2.919	1.874	1.980
	地尚	0.540	0.318	0.346	6.262	5.412	5.484
速度 误差 (m/s)	北向	0.100	0.045	0.052	0.136	0.101	0.102
	东向	0.096	0.067	0.070	0.138	0.097	0.102
	地尚	0.020	0.012	0.013	0.160	0.141	0.142
姿态 误差 (deg)	横滚	0.012	0.010	0.010	0.013	0.009	0.009
	俯仰	0.020	0.015	0.016	0.026	0.018	0.019
	航向	0.108	0.068	0.073	0.278	0.231	0.234

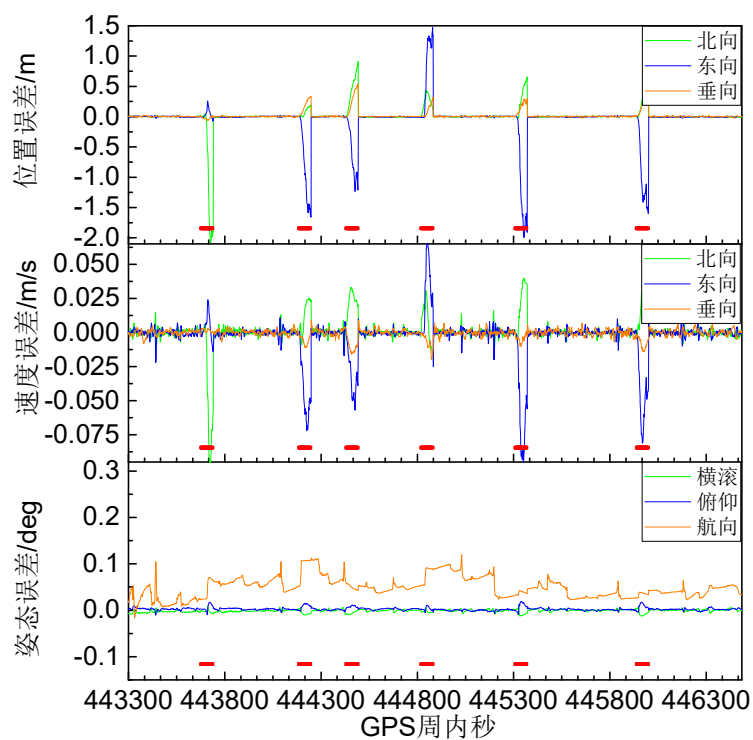


图 2.7 方案 1 导航误差

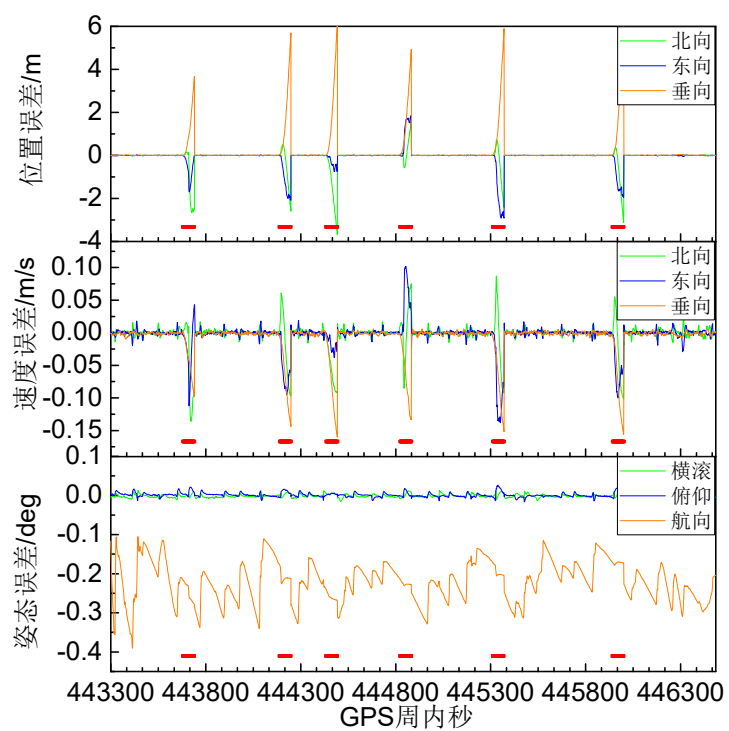


图 2.8 方案 2 导航误差

3 GNSS/INS 实时组合导航系统设计

本文使用 CCS(Code Composer Studio) 6.1.0 基于 DSP 平台开发了 GNSS/INS 实时组合导航系统,并在 Windows 平台中使用 VS(Visual Studio)2015+Qt5.6 开发了配套的交互应用程序。该系统具有零速修正、里程计辅助及非完整性约束的车辆约束功能,惯性导航解算率为 200Hz,组合更新率为 1Hz,最高导航结果输出率为 100Hz,可输出指定目标点的位置、速度、姿态、加速度以及角速度。

3.1 时间延迟处理及效果验证

GNSS/INS 组合实时应用中,算法可以主要分为机械编排-Kalman 预测及组合更新 2 部分,前者运算量较小,一般可以在 IMU 采样间隔内完成计算;后者则比较耗时,普通嵌入式处理器中无法在 IMU 采样间隔内及时完成,并且 GNSS 观测数据的处理和传输也会有一定的延迟,GNSS 观测数据获取不及时,最终导致组合更新无法同步完成。该问题直接影响 GNSS/INS 实时组合系统的性能,如果将传输结束时刻的 GNSS 观测数据与该时刻的 INS 推算结果直接形成观测方程送入 Kalman 滤波器进行数据融合,就会由于时间异步引入较大误差,并通过反馈校正将系统带入恶性循环,最终可能导致滤波器发散^[35];如果在接收到 GNSS 观测数据之后,再从观测数据采样时刻到当前时刻重新进行计算,虽然可以得到无量测延迟情况下相同的结果,但会大大增加导航系统的计算压力和存储数据的内存压力,最终导致系统的实时性能大打折扣^[36]。

目前,时间延迟的解决策略主要有外推观测数据^[37]、增广状态方法^[12]、新息重组方法^[14,15]。其中,外推观测数据依赖外推模型,外推精度无法保障;增广状态方法会增加状态维度,使得增大计算量;新息重组的方法,需要建立多个观测方程,如果有多步延迟,则不利于对 Kalman 滤波器的灵活拓展与应用。而且这些方法只能解决观测数据传输延迟,不能解决组合更新计算延迟的问题。

针对时间延迟问题,本文提出了适用于 GNSS 观测数据延迟及组合更新计算延迟的 Kalman 滤波数学模型。该模型将机械编排-预测以及组合更新分解成具有优先级别的 2 个独立任务进行,处理时序示意如图 3.1 所示。在 GNSS 采样时刻

保存预测误差协方差矩阵和机械编排解算的导航状态，在该采样时刻 GNSS 观测数据实际接收到的时刻，开始进行组合更新计算，在组合更新计算完成后，还需要将滞后计算出来的反映 GNSS 采样时刻的状态参数估计量以及状态参数协方差估计量正确转移到计算完成时刻（即当前时刻），以修正当前时刻的惯导误差，输出最优的组合结果，本文利用状态转移达到这一目的。在进行这些处理的过程中，一直保持优先进行 INS 机械编排和 Kalman 预测计算。这样，只要保证机械编排-Kalman 预测在 IMU 采样间隔（如 5ms）内完成，以及 Kalman 观测更新计算在 GNSS 采样间隔（如 1s）内完成，就能够确保组合导航解算的实时性。

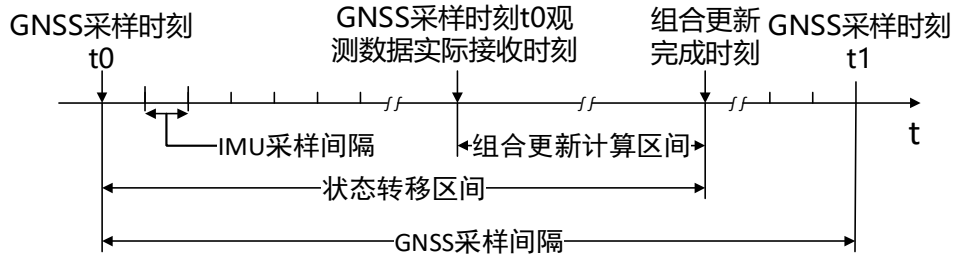


图 3.1 处理时序示意图

3.1.1 状态转移

状态转移是根据 k 时刻的状态估计 j 时刻（ $j > k$ ）的状态，即根据 k 个测量 $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_k$ 对 \mathbf{x}_j 作线性最小方差估计 $\hat{\mathbf{x}}_{j,k}^-$ [6]:

$$\hat{\mathbf{x}}_{j,k}^- = \hat{E}[\mathbf{x}_j / \mathbf{z}_1 \mathbf{z}_2 \dots \mathbf{z}_k] \quad (3.1)$$

根据(2.43)，并且考虑到 $\Phi_{k+1/k-1} = \Phi_{k+1/k} \Phi_{k/k-1}$ ，从 k 时刻迭代到 j 时刻可以得到：

$$\mathbf{x}_j = \Phi_{j/k} \mathbf{x}_k + \sum_{i=k+1}^j (\Phi_{j/i} \mathbf{G}_{i/i-1} \mathbf{w}_{i-1}) \quad (3.2)$$

将式(3.2)代入式(3.1)得到：

$$\begin{aligned} \hat{\mathbf{x}}_{j,k}^- &= \hat{E}[\mathbf{x}_j / \mathbf{z}_1 \mathbf{z}_2 \dots \mathbf{z}_k] \\ &= \hat{E}[(\Phi_{j/k} \mathbf{x}_k + \sum_{i=k+1}^j (\Phi_{j/i} \mathbf{G}_{i/i-1} \mathbf{w}_{i-1})) / \mathbf{z}_1 \mathbf{z}_2 \dots \mathbf{z}_k] \\ &= \hat{E}[\Phi_{j/k} \mathbf{x}_k / \mathbf{z}_1 \mathbf{z}_2 \dots \mathbf{z}_k] + \sum_{i=k+1}^j \Phi_{j/i} \mathbf{G}_{i/i-1} \hat{E}[\mathbf{w}_{i-1} / \mathbf{z}_1 \mathbf{z}_2 \dots \mathbf{z}_k] \end{aligned} \quad (3.3)$$

由式(2.43)可知， \mathbf{w}_{k-1} 只影响到 \mathbf{x}_k ，所以 \mathbf{w}_{i-1} ($i > k+1$ 且 $i \leq j$) 与 $\mathbf{z}_1 \mathbf{z}_2 \dots \mathbf{z}_k$ 不相

关，且 $E(\mathbf{w}_{i-1})=0$ ，因此

$$\hat{\mathbf{x}}_{j,k}^- = \Phi_{j/k} \hat{E}[\mathbf{x}_k / \mathbf{z}_1 \mathbf{z}_2 \cdots \mathbf{z}_k] = \Phi_{j/k} \hat{\mathbf{x}}_k^+ \quad (3.4)$$

式中， $\hat{\mathbf{x}}_k^+$ 为 \mathbf{x}_k 更新估计量。

对于预测误差的协方差矩阵 $\mathbf{P}_{j,k}^- = E[\tilde{\mathbf{x}}_{j,k} \tilde{\mathbf{x}}_{j,k}^T]$ ， $\tilde{\mathbf{x}}_{j,k}$ 为 $\hat{\mathbf{x}}_{j,k}^-$ 的误差，即

$$\begin{aligned} \tilde{\mathbf{x}}_{j,k} &= \mathbf{x}_j - \hat{\mathbf{x}}_{j,k}^- \\ &= \Phi_{j/k} \mathbf{x}_k + \sum_{i=k+1}^j (\Phi_{j/i} \mathbf{G}_{i/i-1} \mathbf{w}_{i-1}) - \Phi_{j/k} \hat{\mathbf{x}}_k^+ \\ &= \Phi_{j/k} \tilde{\mathbf{x}}_k + \sum_{i=k+1}^j (\Phi_{j/i} \mathbf{G}_{i/i-1} \mathbf{w}_{i-1}) \end{aligned} \quad (3.5)$$

$$\begin{aligned} \mathbf{P}_{j,k}^- &= E[\tilde{\mathbf{x}}_{j,k} \tilde{\mathbf{x}}_{j,k}^T] \\ &= \Phi_{j/k} E[\tilde{\mathbf{x}}_k \tilde{\mathbf{x}}_k^T] \Phi_{j/k}^T + \sum_{i=k+1}^j (\Phi_{j/i} \mathbf{G}_{i/i-1} E[\mathbf{w}_{i-1} \mathbf{w}_{i-1}^T] \mathbf{G}_{i/i-1}^T \Phi_{j/i}^T) \\ &= \Phi_{j/k} \mathbf{P}_k^+ \Phi_{j/k}^T + \sum_{i=k+1}^j (\Phi_{j/i} \mathbf{G}_{i/i-1} \mathbf{Q}_{i-1} \mathbf{G}_{i/i-1}^T \Phi_{j/i}^T) \end{aligned} \quad (3.6)$$

式中， $\mathbf{P}_{j,k}^-$ 为 \mathbf{x} 在 j 时刻的预测协方差矩阵； \mathbf{P}_k^+ 为 \mathbf{x} 在 k 时刻的协方差矩阵；

\mathbf{Q} 为 \mathbf{w} 的协方差矩阵。在 GNSS/INS 紧组合中 \mathbf{G} 为正定矩阵，因此， $\mathbf{G} \mathbf{Q} \mathbf{G}^T = \mathbf{Q}$ 。

令 $\mathbf{M}_{j,k+1} = \sum_{i=k+1}^j \Phi_{j/i} \mathbf{Q}_{i-1} \Phi_{j/i}^T$ ，可以得到

$$\begin{aligned} \mathbf{M}_{j+1,k+1} &= \mathbf{Q}_j + \Phi_{j+1/j} \mathbf{M}_{j,k+1} \Phi_{j+1/j}^T \\ \mathbf{M}_{k+1,k+1} &= \mathbf{Q}_k \end{aligned} \quad (3.7)$$

综上，状态转移公式为

$$\begin{aligned} \hat{\mathbf{x}}_{j,k}^- &= \Phi_{j/k} \hat{\mathbf{x}}_k^+ \\ \mathbf{P}_{j,k}^- &= \Phi_{j/k} \mathbf{P}_k^+ \Phi_{j/k}^T + \mathbf{M}_{j,k+1} \end{aligned} \quad (3.8)$$

由式(3.8)可以看出，状态转移可以由累积的状态转移矩阵及状态噪声阵得到，把 GNSS 采样时刻看作是 k 时刻，更新计算完成时刻看作是 j 时刻，利用式(3.8)可以把 k 时刻的状态参数更新估计量以及状态参数协方差更新估计量转移到 j 时刻，进而在更新计算完成时刻进行组合更新。

状态转移矩阵 Φ 中存在大量零元素，在实时程序中，为了减少计算量，可以对与 Φ 相关的计算进行专门处理，避免零元素参与乘法运算。

3.1.2 效果验证

为了验证本文对延迟处理的正确性,利用 2015.3.27 采集的原始数据,采用后处理前向滤波的方式来验证。对测试时段从 GPS 周内秒 443400 开始到 GPS 周内秒 446280 结束,共模拟了 13 次卫星信号中断,每次中断时长 1min,中断结束后信号恢复 3min,用来恢复状态估计。通过对比在没有延迟 GNSS 观测数据的情况下,标准 Kalman 滤波处理方法(方案 1)与 GNSS 观测数据被人为延迟 0.1s 的情况下,采用本文提出的改进 Kalman 滤波(方案 2)处理方法在信号中断时段的导航误差,来评估本文对时间延迟处理的精度,其中参考真值使用该组数据的后处理反向平滑结果。

两种方案处理的误差曲线如图 3.2、图 3.3 所示,从图中可以看出导航误差曲线基本一致,从表 3.1 的统计结果来看,方案 1 和方案 2 在卫星信号中断期间的导航误差统计值具有很好的一致性,可以认为方案 1 与方案 2 处理精度一样,证实了本文提出的改进 Kalman 滤波算法对延迟处理的正确性。

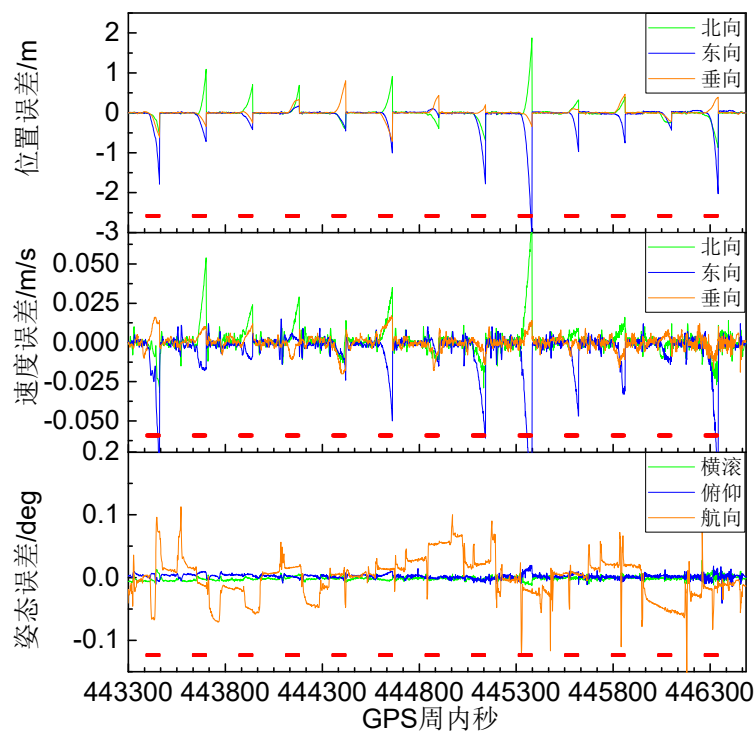


图 3.2 方案 1 导航误差

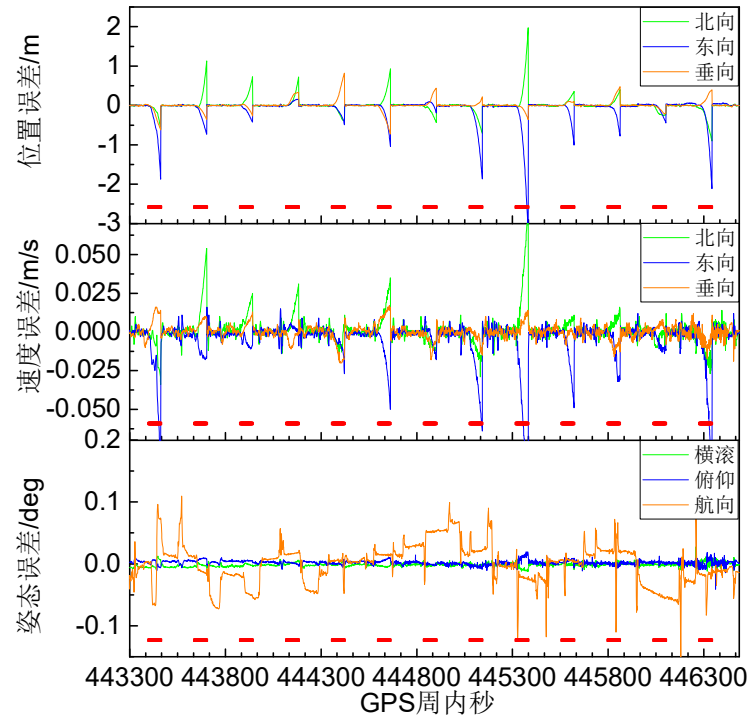


图 3.3 方案 2 导航误差

表 3.1 仿真中断期间导航误差统计表

误差项		方案 1			方案 2		
		最大值	均值	RMS	最大值	均值	RMS
位置 误差 (m)	北向	1.793	0.680	0.784	1.798	0.680	0.787
	东向	3.104	1.024	1.314	3.104	1.024	1.320
	地(垂)向	0.790	0.392	0.437	0.791	0.392	0.438
速度 误差 (m/s)	北向	0.075	0.028	0.031	0.074	0.028	0.033
	东向	0.115	0.043	0.055	0.115	0.043	0.054
	地(垂)向	0.020	0.014	0.013	0.020	0.014	0.014
姿态 误差 (deg)	横滚	0.012	0.007	0.007	0.014	0.007	0.008
	俯仰	0.020	0.010	0.010	0.020	0.010	0.011
	航向	0.119	0.060	0.065	0.119	0.060	0.068

3.2 组合导航硬件平台设计

本文开发的 GNSS/INS 实时组合导航系统主要由 4 个功能模块设备组成, 分别为: IMU 单元、GNSS 单元、数据存储单元、组合解算单元, 各单元之间通过内部串口通信, 如图 3.4 所示。IMU 单元主要包括 3 只石英加速度计、一只三轴光纤陀螺仪、2 个单片机 (MCU1_Sample、MCU2_IMU)。MCU1_Sample 负责传感器的原始数据采集, 加速度计输出的信号为模拟信号, 信号经连接器进入到

采样电路板，首先通过运算放大器把信号放大，之后输入 A/D 芯片，经模数转换后输入到 MCU1_Sample；陀螺仪输出的信号为脉冲信号，无需转换直接进入 MCU1_Sample。MCU2_IMU 负责原始数据的补偿，主要包括温度补偿、刻度因子补偿、零偏修正、安装误差补偿，非线性补偿等，最终得到载体在运动过程中沿载体坐标系的角速率和加速度信息，并通过 GNSS 单元的 1PPS 信号与 GPS 时间消息进行 IMU 数据与 GNSS 数据的时间同步，当系统接入里程计时，时间同步的同时也进行里程计数据的采集，最后按照协议 IMU 观测数据。GNSS 单元为 Trimble BD982 板卡，该板卡为多系统多频点的高精度测量型 GNSS 板卡，支持双天线测姿^[38]，负责为本系统提供统一的时间基准以及 GNSS 定位、定姿结果。存储单元主要为 1 个单片机 (MCU3_SDCARD)，负责本系统所有数据的在线存储，存储媒介为 microSD 卡，通过 USB 口进行数据拷贝。组合解算单元主要为 1 个 DSP，负责从 GNSS 单元以及 IMU 单元获取 GNSS 观测数据以及 IMU 观测数据，进行组合导航解算，最后以协议导航结果格式向存储单元以及用户实时输出导航结果。

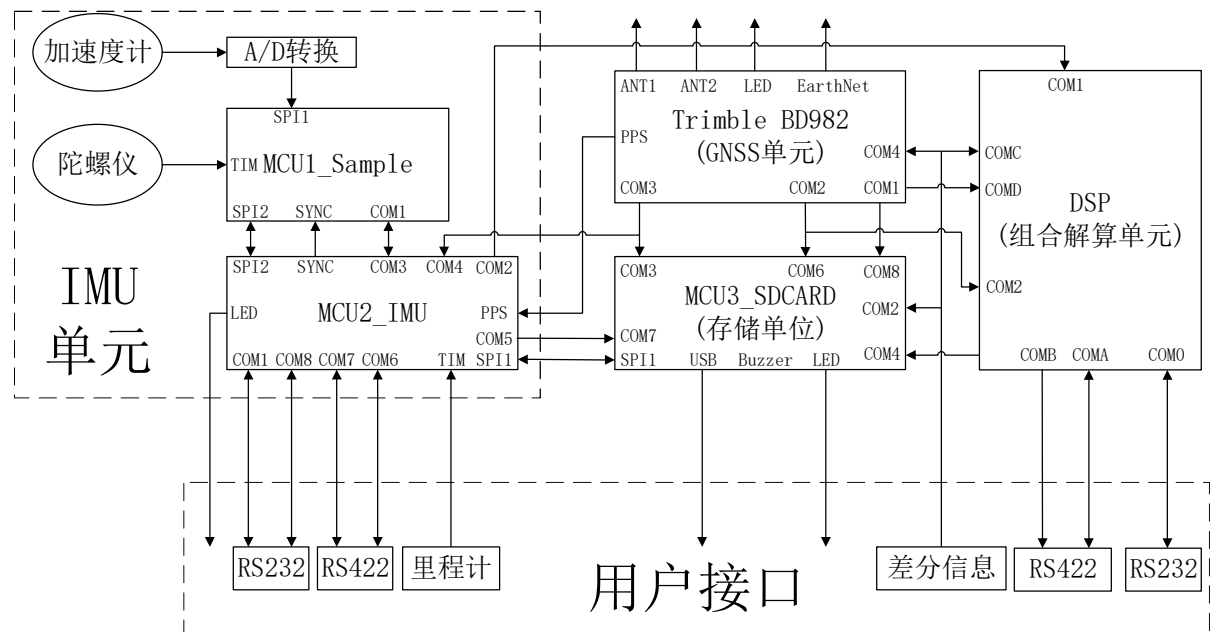


图 3.4 GNSS/INS 实时组合导航系统设计图

本系统中采用的单片机的具体型号为 ST 公司的 STM32F429，该处理器为 32 位处理器采用 Cortex™-M4 内核，主频高、解算速度快，可靠稳定^[39]。A/D 转换选用 TI 公司的 ADS1258，该芯片为 16 路 24 位 A/D 转换进行数据采集，具有精度高，采样速率快等特点^[40]。DSP 采用 TI 公司的 TMS320C6747，为低功耗设

计的 32 位高端浮点数字信号处理器，主频高达 300MHz，可超频至 480MHz^[41]。

本系统是在武汉迈普时空导航科技有限公司的组合导航系统 POS320 的基础进行开发的，其外观如图 3.5 所示，主要性能参数见表 3.2。



图 3.5 POS320

表 3.2 POS320 的 IMU 主要性能参数

指标	数值
陀螺零偏/ $(^{\circ})h^{-1}$	0.5
角度随机游走/ $(^{\circ})h^{-1/2}$	0.05
加速度零偏/mGal	25
速度随机游走/ $(m)s^{-1}h^{-1/2}$	0.1

3.3 时间同步处理

多源导航传感器进行组合时，导航信息的空间和时间上的统一是重要前提。空间统一是指坐标框架的一致，在 GNSS/INS 组合导航中一般使用 GNSS 系统中的坐标框架，如 WGS84(World Geodetic System 1984)、ITRF(International Terrestrial Reference Frame)、CGCS2000(China Geodetic Coordinate System 2000)等，在组合导航解算时加以转化改正，比较容易实现。而时间统一相对复杂，如果将时间异步的导航信息进行融合，组合导航滤波器将会出现严重误差，甚至引起整个导航系统的异常^[42]。因此，GNSS/INS 实时组合导航系统必须解决时间同步的问题。

GNSS/INS 组合导航系统中时间异步的主要原因有：

1) 时间起点和基准不一致

作为独立的两个导航系统，GNSS 和 INS 各自具有时间系统。GNSS 的时间系统通常为 GPS 时间，其为原子时，由 GPS 系统的原子钟维护，可以看作是一个精度极高的时间系统。INS 的时间系统为本地时间，依靠惯性传感器内部晶振维护，通常每次上电开机后从零开始计时，属于相对时间系统，而且相对于 GNSS 时间系统，其频率不稳定，容易发生漂移。

2) 采样点不一致

GNSS 系统的采样率一般为 1Hz，并且能够严格在 GPS 整秒时刻进行采样，而 INS 系统的采样率一般高达 200Hz 或者更高，不能保证在 GPS 整秒时刻进行采样。而 GNSS/INS 组合系统的 Kalman 滤波中，会选择 GNSS 采样时刻进行观

测更新，此时由于采样时刻的不一致，会导致不一定存在 INS 导航结果。

3) 硬件延迟

硬件延迟主要为 INS 和 GNSS 系统观测数据的输出时延，主要表现为内部通信延迟、数据处理延迟等。

为了解决时间同步的问题，本文借助 GNSS 接收机的 1PPS(Pulse Per Second) 信号设计了同步方案。GNSS 接收机的 1PPS 的上升沿精度通常小于 $1\mu s$ ，对于授时型接收机则优于 100ns。因此，1PPS 可以作为一个将 INS 本地时间与 GPS 时间进行同步的信号^[43]。但 1PPS 仅是一个发生在 GPS 整秒时刻的脉冲信号，并不能代表具体时间，因此，进行 INS 除了接受来自 GNSS 接收机的 1PPS 信号外，还需要接收一个有带有时间信息的信息，如图 3.4 中的 MCU2_IMU 与 GNSS 单元之间的 1PPS 和 COM4-COM3 连接。

表 3.3 IMU 数据格式

地址偏移	数据类型	说明
0	unsigned char	字头 0x55
1	unsigned char	字头 0xAA
2~5	unsigned int	GPS 周内秒(s)
6~9	unsigned int	1PPS 计数
10~13	unsigned int	IMU 计数
14~17	float	Gyro X(deg)
18~21	float	Gyro Y(deg)
22~25	float	Gyro Z(deg)
26~29	float	ACC X(m/s)
30~33	float	ACC Y(m/s)
34~37	float	ACC Z(m/s)
38~41	unsigned char	预留
42	unsigned char	检校位,2~41 字节的和检校
115200-8-N-1, 小端编码, 数据率: 200Hz		

本文使用的 IMU 数据格式如表 3.3 所示，时间同步方案为：在 MCU2_IMU 模块中，系统自主采样，在 IMU 数据采样时刻读取本地晶振计数，更新 IMU Count；在收到 1PPS 信号时刻，读取本地晶振计数，更新 PPS Count；在收到时间消息时，更新 GPS Sow(Second Of Week)。在组合导航解算模块中，利用 IMU Count、PPS Count 以及 GPS Sow 可以得到每个 IMU 数据对应的 GPS 时间，整个过程中，除了需要当前时刻的 IMU 数据外，还要保存上一时刻的 IMU Count、PPS Count 以及上一次判断为 1PPS 触发时的整秒时间，具体操作如下：在初始

化过程中，利用 PPS 计数以及 GPS Sow 得到第一次判断为 1PPS 触发的 GPS 整秒时间，通常 GPS Sow 消息更新会滞后于 1PPS 触发，这里需要特殊处理一下，之后，每次 IMU 数据解码成功后，与上一时刻的 PPS Count 比较，判断当前 PPS Count 是否改变，如果没有改变，就使用当前 IMU Count 与上一时刻的 IMU Count 之差除以本地晶振频率得到时间间隔，累加到上一时刻的 IMU 数据的时间上作为当前 IMU 数据的时间；如果改变，判断变化差异是否与本地晶振频率差不多，以确保是正确的 1PPS 信号触发时刻，如果差异较大，认为 1PPS 异常，给出警告，并和判断为 PPS Count 没有改变一样，使用内部晶振计数计算时间差，在上一时刻时间上累加作为当前 IMU 数据的时间，如果判断为 1PPS 触发时刻，将当前的 IMU Count 与 PPS Count 做差除以本地晶振频率得到当前时间和整秒时刻的时间差，再由 GPS Sow 得到当前整秒时间，加上前一步得到的时间差，作为 1PPS 修正后的当前 IMU 数据的时间。

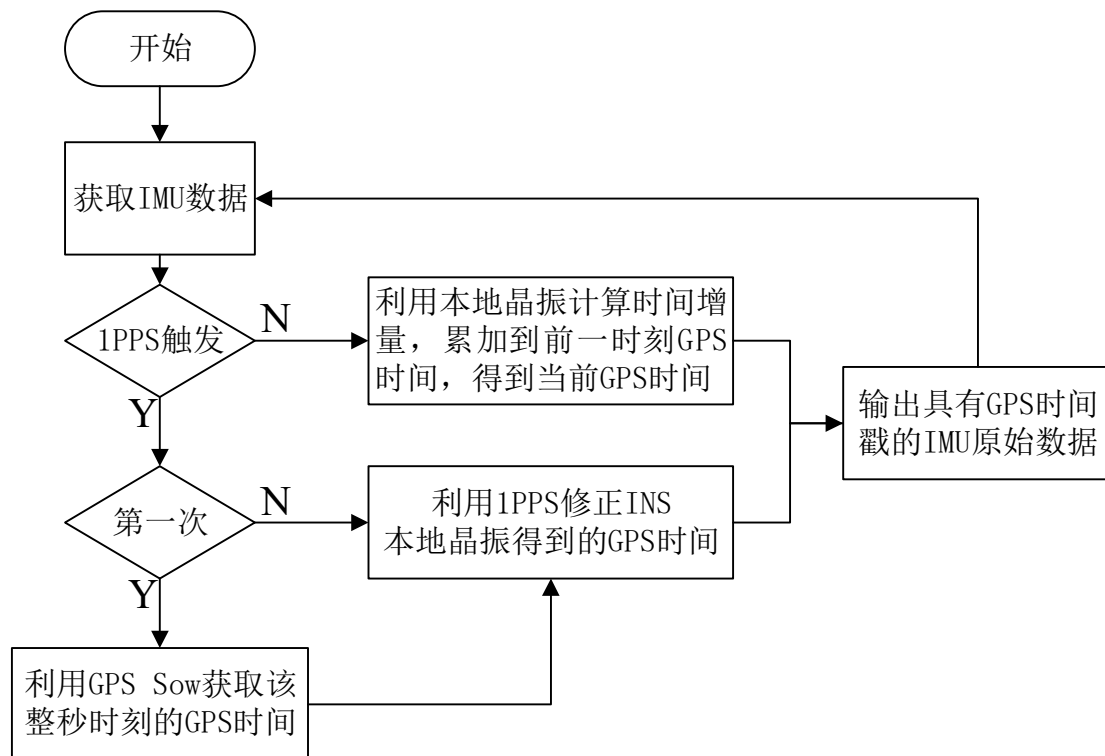


图 3.6 时间同步流程图

对于 IMU 数据的硬件延迟，系统采用 SPI(Serial Peripheral Interface)口进行 IMU 采样部分的传输，时间延迟可以忽略不计，GNSS 数据延迟采用 3.1 中补偿方法进行了补偿。IMU 数据和 GNSS 数据采样时间点不一致的问题，由于 GNSS 数据本身就存在时间延迟，在利用 1PPS 对组合系统进行时间基准统一后，本系

统使用横跨整秒时刻的 IMU 导航结果，线性内插出整秒时刻的预测导航状态，用于和整秒的 GNSS 观测进行 Kalman 组合更新。

3.4 DSP 端程序设计

3.4.1 引导程序设计

DSP 程序在调试阶段时，可以使用仿真器让程序在线运行，方便调试程序。当 DSP 程序正式投入应用时，就要求其能够自主启动，因此需要为 DSP 程序添加引导（Bootload）程序。TMS320C6000 系列 DSP 自带了 Bootload 程序，上电之后，该程序会从外部空间中复制不大于 1kB 的程序到地址 0 处开始运行。然而，一个应用程序往往远大于 1kB，因此无法使用自带的引导程序直接引导应用程序，需要设计二次引导（Second Bootload）程序，Second Bootload 程序代码量小于 1kB，负责从外部存储空间搬移应用程序到 DSP 中，并启动应用程序。可以通过多种方式启动片内固化的 Bootload 程序，例如 *NOR*、*NAND*、*SPI0~1*、*UART0~2* 启动或者仿真模式，通过 DSP 上电时 Boot 引脚的状态来选择具体使用的启动模式^[44]。

本文采用了 *SPI1* 启动模式，在 DSP 的 *SPI1* 处连接了 FLASH，利用 CCS 在仿真模式下向 FLASH 地址 0 处烧写了 Second Bootload 程序，该程序流程图如图 3.7 所示，主要负责：

- 1) 配置 PLL(Phase Locked Loop，锁相环)，确保 DSP 工作在正确的频率。
- 2) 根据 FLASH 存储器的时序要求配置 EMIF(External Memory Interface，外部存储器接口)，使得代码能够访问外置存储 FLASH。
- 3) 超时等待上位机是否有固件升级指令，如果没有就根据存储格式定义，开始搬移应用程序到 DSP 中；如果有升级请求就根据外部输入的数据进行应用程序更新。DSP 与上位机之间通过 DSP 的 COM0 进行通讯，固件传输协议为 xmodem^[45]。
- 4) 调用 C 语言入口函数，开始 C 语言运行环境的初始化以及用户代码的执行。

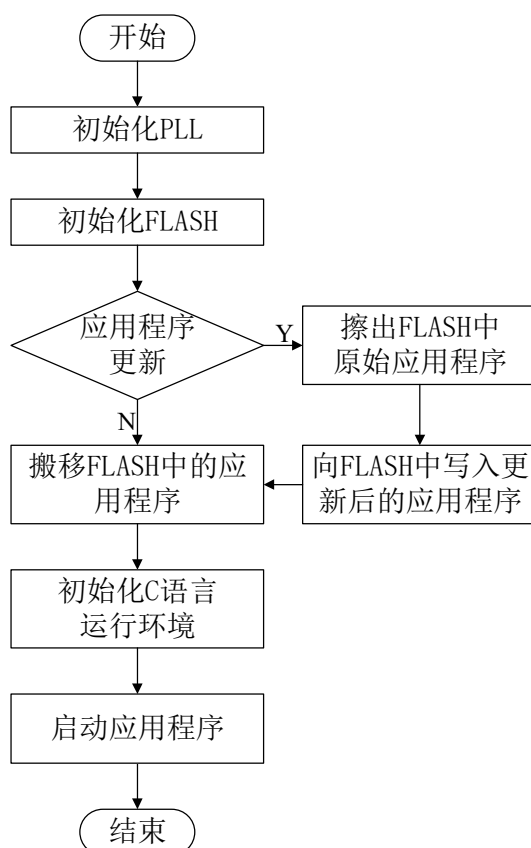


图 3.7 Second Bootload 流程图

3.4.2 组合导航程序设计

组合解算单元与 GNSS 单元、IMU 单元、存储单元以及用户都是通过串口进行通讯，TMS320C6000 系列 DSP 提供了 FIFO(First Input First Output)模式的串口，发送及接收串口均有 16 字节的缓存，FIFO 模式包含中断模式以及轮询模式^[46]。考虑到本系统串口比较多并且 IMU 数据量大、传输速率快，如果使用中断模式，中断服务函数会比较多，程序逻辑复杂，因此，本系统采用了轮询模式，利用一个定时器中断，在中断服务函数中依次查询本系统使用的所有串口：对于接收串口，如果缓存区中有数据，就接收数据存入相应的缓存中；对于发送串口，如果处于可发送状态，就从导航缓存中读取 16 字节数据进行发送，以减少程序占用时间。实际测试表明 4kHz 的中断频率可以保证数据完整接收。

1) 环形缓存区

对于各个串口接收以及本系统发送的数据，程序中采用环形缓存区进行管理。环形缓存区是一种固定尺寸、头尾相连的数据结构^[47]，具有删除数据元素后，剩余数据存储位置保存不变的特点。也就是说，圆形缓冲区是一个先进先出的队列，

本系统的数据正需要先进先出的处理，因此选用环形缓存区。

考虑到内存是线性地址空间，要想从逻辑上实现圆形缓冲区，便需要经过特别的处理，从图 3.8 可以看出要实现圆形缓冲区，一共需要 4 个指针：

- (1) 缓存物理开始地址（首地址）；
- (2) 缓存物理结束地址（尾地址），或者使用缓冲区长度；
- (3) 缓存中有效数据的开始地址（读指针）；
- (4) 缓冲中有效数据的结束地址（写指针）。

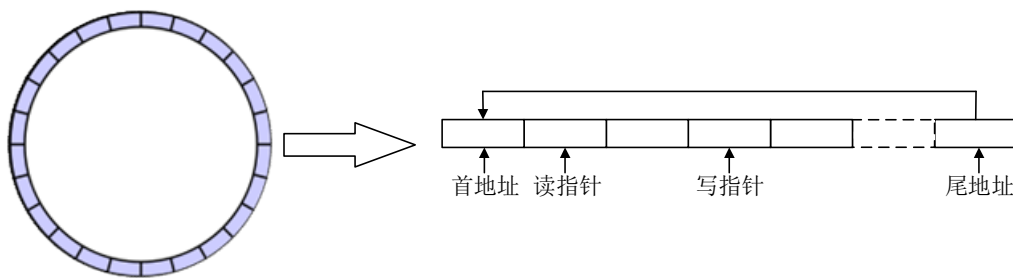


图 3.8 环形缓存区示意图

为了避免将读写指针转化为缓冲区的数组索引时进行运算量大的“求余”操作，本文采用的长度为 2 的幂的缓冲区，读、写指针均为无符号长整型变量，仅需要把读写指针值和缓冲区长度减一的值进行“按位与”操作，便可到达目的。并且读写指针总是满足以下关系：

$$\text{写指针} = \text{缓冲区存储长度} + \text{读指针}$$

由于无符号长整型的加法特点，上式在读写指针达到无符号长整型最大值后仍然成立，根据上式可以方便的获得缓存区已经存储的数据长度。

获得了读写指针对应的数组索引后，便可进行读取、写入操作，如图 3.9 所示。读写操作类似，以写操作为例进行说明。对缓存进行写操作时，首先，获得当前缓存区还可以写入的长度，与待写入数据做对比，取较小的值为写入数据的长度；然后，分两步向缓存写入数据，先从写指针位置向缓冲区尾地址方向写入，如果还有剩余未写完的数据，再从缓冲区的首地址开始写入余下数据，并增加写指针写入的长度。

环形缓冲区在进行读写操作时，读用户只会修改读指针，而写用户只会修改写指针。当环形缓存区只有一个读用户和一个写用户而且读写操作不能再同一时刻进行，那么就没有必要使用互斥保护机制，否则必须使用互斥保护机制才能够保证数据的正确性。

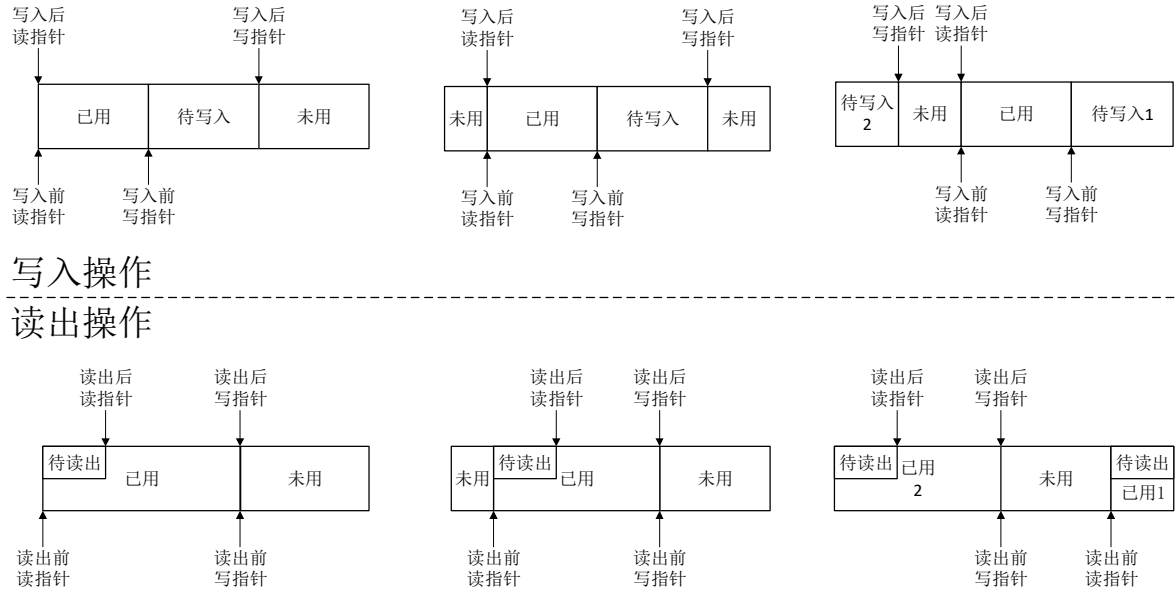


图 3.9 环形缓存区操作示意图

2) 解码与编码

本系统采用串口与外设通讯，还涉及到数据解码与编码问题，包括 IMU 数据、Trimble GSOE 格式解码以及解算结果编码。虽然数据格式比较多，但都基于数据块的协议来输出数据的，每个数据块基本都包含该数据块的帧头、该数据块的字节数、数据正文、校验等内容，其中数据块的帧头是判断数据块类型的依据。在解码时首先就要判断数据块的类型，然后将不同类型交给不同的解码函数进行解码并进行相应的数据处理。解码时要注意的是 IMU 数据和 GNSS 数据的时间同步问题，IMU 数据本身是没有明确的时间信息，仅有通过 GNSS 板卡 PPS 信号触发得到的同步脉冲计数，可使用该计数以及 GNSS 数据中获取的 GPS 时间得到 IMU 观测数据的具体 GPS 时间戳，详见 3.3。

为了与上位机进行交互，本系统设计了通讯协议，详见 6。对数据进行编码时，本系统使用的数据长度均是某一基本数据类型（比如 int，double 类型等）的长度，可以直接使用 memcpy() 函数将待编码的数据直接赋值给无符号数组，以供串口设备进行发送。

3) 实时组合处理

根据第 3.1 节的时间延迟处理的要求，组合解算程序应分为机械编排-Kalman 预测以及组合更新两部分同步进行，为了实现多任务模式的解算，在组合解算程序中进行了中断优先级和中断嵌套设计^[41,44,46]，使用 DSP 片内 2 个 32 位定时器将整个程序分为 4kHz 中断、250Hz 中断以及主函数 3 部分：4kHz 中断优先级高

于 200Hz 中断，负责把 GNSS、IMU 原始数据分别保存到 GNSS、IMU 缓存区以及把导航缓存区中的数据通过串口发送到用户端；250Hz 中断从 IMU 缓存区中获取 IMU 观测数据进行机械编排、Kalman 预测以及从主函数获取组合更新量进行反馈，再把导航结果存入导航结果环形缓存区中；主函数负责从 GNSS 缓存区中获取 GNSS 观测数据进行组合更新计算，并将组合更新量传入 250Hz 中断。GNSS/INS 组合导航程序具体流程如下：

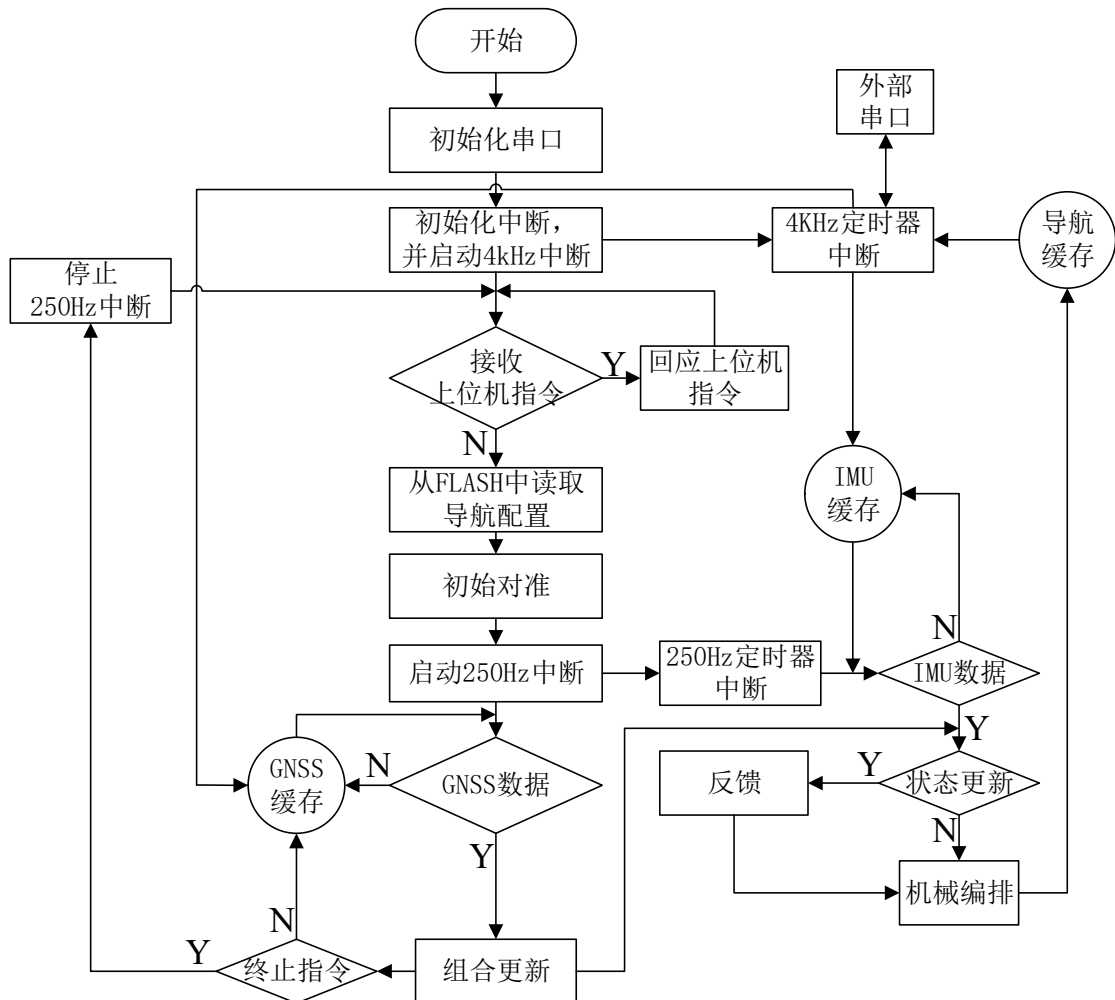


图 3.10 组合导航流程图

3.5 PC 端程序设计

为了方便用户与实时 GNSS/INS 组合导航系统进行交互，本文在 Windows 平台上使用 Qt^[48, 49]开发了配套的应用程序，具有升级 GNSS/INS 组合导航系统固件、管理配置组合导航系统设置以及实时显示载体的位置、速度、姿态、卫星星图 and 信噪比的功能，主界面如图 3.11 所示。该程序的主要功能由 2 个线程

组成，一个为 GUI（Graphical User Interface）线程，负责响应用户界面上的操作以及导航结果的可视化；另一个线程为串口数据接收线程，负责从串口接收导航结果，保存到存储设备，并最高以 20Hz 的频率对接收到的导航结果进行解码，通知 GUI 线程进行相关信息显示。

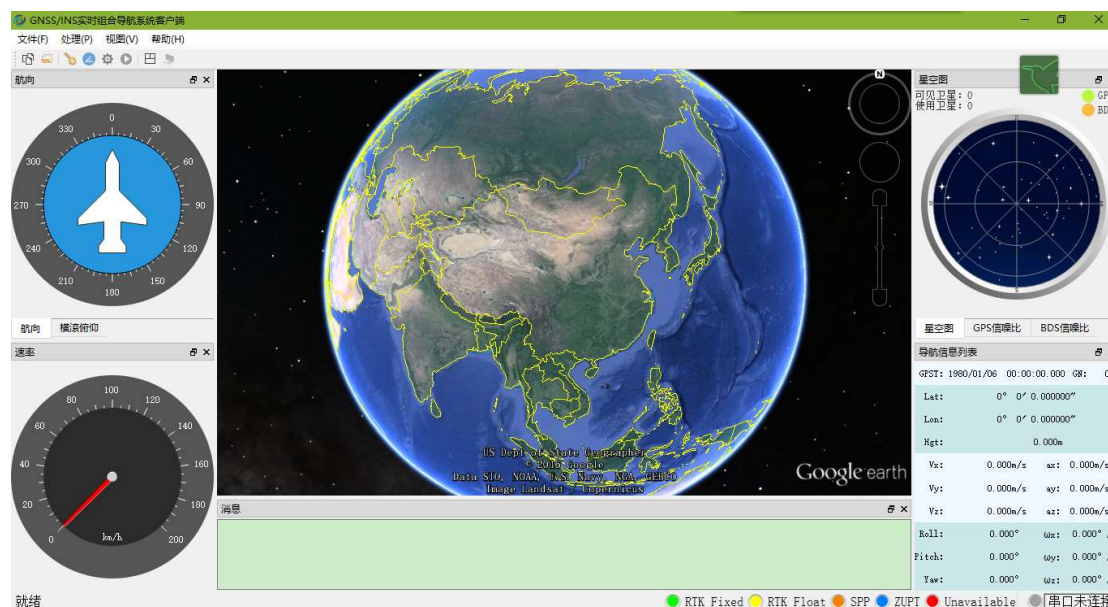


图 3.11 GNSS/INS 实时组合导航系统客户端

程序运行时，默认情况下，将会有载体方向角、横滚角、俯仰角、速率的可视化显示综合导航信息列表、参与定位解算的卫星星空图以及信噪比；当前位置将在嵌入的 Google Earth 中显示，Google Earth 视图不断移动，保证当前位置基本在界面的中部。在程序的状态栏的右侧会显示当前接收的串口数据的数据量，左侧会显示当前的项目名称。

状态栏指示灯的颜色说明：灰色-串口未连接、绿色-RTK 固定解、黄色-RTK 浮点解、橙色-单点定位解、蓝色-零速修正状态、红色-无卫星信号。

3.5.1 程序关键模块介绍

由于实时组合导航系统与 PC 之间通过串口进行通讯，本程序中采用基于 Qt 的第三方开源串口类库(QextSerialPort^[50])获取组合导航系统的实时导航数据流以及进行实时组合导航系统的配置管理和固件升级；为了将实时组合导航系统的导航结果直观地显示在地图中，本程序采用了 Google Earth COM API 结合 KML 脚本语言实现导航结果在 Google Earth 中的实时显示；本程序中使用了基于 Qt 的第三方图像库 Qwt^[51]实现了导航信息的可视化显示。

1) 串口通讯

串口通信操作流程一般是设置串口参数、打开串口、使用串口进行数据传输、关闭串口。QextSerialPort 串口库具体使用方式：

定义：Win_QextSerialPort *_serialPort;

a) 设置串口

```
_serialPort->setPortName("COM1");//设置串口编号
_serialPort->setQueryMode(QextSerialBase::EventDriven);//设置串口使用模式
_serialPort->setBaudRate(BAUD115200);//设置串口的波特率
_serialPort->setDataBits(DATA_8);//设置数据位
_serialPort->setParity(PAR_NONE);//设置奇偶校验位
_serialPort->setStopBits(STOP_1);//设置停止位
_serialPort->setFlowControl(FLOW_OFF); //设置数据流控制
将_serialPort 的 readyRead 信号与数据接收类的接收槽函数连接。
```

b) 打开串口

```
_serialPort ->open(QIODevice::ReadWrite)//返回值为 true 时打开串口成功
```

c) 串口通讯

在数据接收类的数据接收槽函数里：

```
int bytesAvailable=_serialPort->bytesAvailable();//获取串口可读字节数目
QByteArray byteArray=_serialPort->read(bytesAvailable);//读取指定数目数据
接收的串口数据存储在 byteArray 变量里。
```

发送串口数据时，直接使用 write 函数：

```
unsigned char *data;//发送数据指针
int size;//数据长度
_serialPort ->write((char*)data,size);
```

d) 关闭串口

```
_serialPort->close();
```

2) Google Earth COM API

2005 年，Google 公司向全球发布了 3D 地图软件——Google Earth，在一个三维地球模型上综合显示了全球卫星、航空照片以及照片对应的地理信息。作为一个优秀的地图平台，Google Earth 不仅为用户提供丰富准确的地图信息，而且

还为用户提供了可再编程的应用程序接口,用户使用这些接口可以方便地对地图显示进行控制,按需求自己编辑、修改、添加地理要素。基于 Google Earth 的应用程序,可以充分使 Google Google 的卫星图片和地图,而不需要自己去维护^[52]。

COM API 主要用于控制 Google Earth 的视角、移动当前视图、获得窗体句柄等^[53]。COM API 的类有 CAnimationControllerGE、CSearchControllerGE、CApplicationGE、CCameraInfoGE、CFeatureCollectionGE、CFeatureGE、CTimeGE、CPointOnTerrainGE、CKHFeature、CKHInterface、CKHViewExtents、CKHViewInfo、COleDispatchDriver、CViewExtentsGE、CTimeIntervalGE、CTourControllerGE。这些类在与类名相同的头文件中,使用时添加相应的头文件引用即可。在 Visual Studio 的非 MFC 工程中使用 COM API 时,需要注意将工程的“常规设置”中的“MFC 的使用”设置为“在共享 DLL 中使用 MFC”,本文仅使用了 CApplicationGE 和 CFeatureGE,下面具体介绍使用方法:

a) 获得窗体句柄,以便将 Google Earth 嵌入到应用程序中

GetMainHwnd 获得主窗体句柄,GetRenderHwnd 获得客户端图层句柄。获取 Google Earth 的窗体句柄后将其传给应用程序的相应显示窗体,即可把 Google Earth 嵌入到应用程序中。

b) 打开 KML 文件

OpenKmlFile (LPCTSTR fileName, long suppressMessages), 参数 fileName 为文件的绝对路径,参数 suppressMessages 一般使用常数 1,表示允许消息反馈。

c) 清理打开的 KML 文件

使用 GetTemporaryPlaces()获得一个 CFeatureGE 对象,使用 CFeatureGE 对象的 put_Visibility(int)函数设置为 0,使其不可见。

d) 调整视角、移动视图

SetCameraParams 设置当前观察位置、视高等参数。

3) KML 脚本编程

Keyhole 公司开发的 KML (Keyhole Markup Language) 是一种广泛应用于表达地理信息的 XML 格式编码规范,Google Earth、Google Maps 均能够读取和显示 KML 文件^[54]。KML 与 HTML 类似,采用包含名称、属性的标签来确定显示方式。

a) KML 文件的常用标签

KML 文档结构主要是由标签组成，这些中有些是抽象的概念，如<Feature>标签，有的是实际要素或者要素属性，如<Point>标签，但绝大部分是实体标签。KML 提供了与地理元素有关的标签用来对地理要素进行操作，常用的有 Coordinates、Point、LineString、Polygon 等。Coordinates 标签由经度、纬度高度构成的序列组成，用于表示序列中点的连线。在 Feature 标签子类中有一个非常重要的标签——Placemark，Placemark 标签可以包含点、线、多边形和地图视角等标签，这些标签用来描述地标的位置以及 Google Earth/Maps 显示地标的视场。同时，为了显示点的详细信息，我们可以对每个地标加上一段状态信息，比如时间信息（GPS 周秒）和点的详细位置参数，这时主要使用 description 标签来完成。description 标签的使用比较灵活，可以加入 HTML 格式的文本描述，也可以加入表格一类的数据。

b) KML 数据动态更新

实时定位时需要 Google Earth 中的位置进行实时更新，为了达到这个目的需要创建 2 个 KML 文件：一个是数据文件，存放当前点的位置信息；另一个添加对坐标 KML 文件的监视。在监视 KML 文件中使用<NetworkLink>和<refreshMode>标签，设置 refreshMode 为 onInterval 模式，设置刷新闻隔，Google Earth 便会按照指定间隔来刷新<href>标签中的链接地址，该链接地址既可以是本地磁盘的文件，即数据文件，也可以是网络链接，文本中采用前者。

最后使用时，将监视文件在 Google Earth 中打开即可，应用程序不断更新数据文件便可实现位姿的定时更新。

3.5.2 程序操作说明

1) 启动

程序可由安装后的快捷方式或直接双击已经创建的项目文件(*.rtnav) 启动。

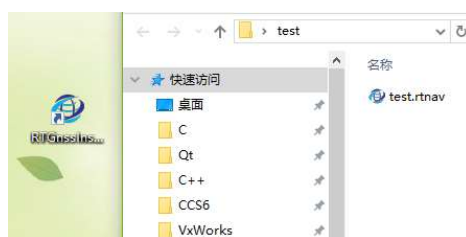


图 3.12 程序启动方式

2) 创建项目

在菜单中选择“文件(F)”-“新建(N)”，弹出图 3.13 所示的设置对话框，设置项目名、项目路径完成项目创建。

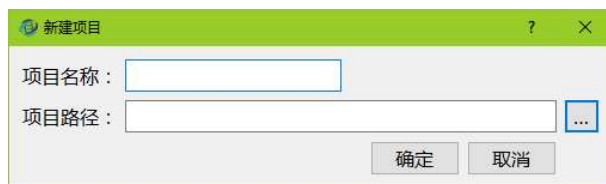


图 3.13 新建项目

3) 打开项目

在菜单中选择“文件(F)”-“打开(O)”，在资源管理其中选择要打开的项目的项目文件或者由程序的历史记录直接打开最近打开过的项目，最多记录 5 个历史项目。



图 3.14 打开项目

4) 开始导航

实时组合导航系统会优先使用内部保存的配置进行导航解算，在确保实时组合导航系统连接正确、当前参数满足需求以及项目参数设置正确的情况下，在菜单中选择“处理(P)”-“开始(B)”开始导航数据接收与显示，在命令提示行将会有实时组合导航系统运行状态提示。

5) 结束导航

选择“处理(P)”-“停止(S)”停止导航数据接收。

6) 参数管理

对实时组合导航系统的导航参数进行管理时，在菜单中选择“处理(P)”-“设置(C)”，弹出图 3.15 所示的对话框。此界面下，可以对系统现有配置进行查询以及参数配置。

The interface is titled "设置" (Settings) and contains the following sections:

- 串口号:** COM6 (dropdown)
- 对准方式:** 动态 (dropdown)
- 数据率/Hz:** 20 (dropdown)
- ☒ 卫星信息
- 获取配置:** [刷新图标]
- 组合模式:** 松组合 (dropdown)
- 导航场景:** 车载 (dropdown)
- ☐ 里程计
- ☐ 航向辅助
- ☒ 至目标点
- ☒ 至载体系
- 天线杆臂/m:** X: 0, Y: 0, Z: 0
- 安装角/deg:** 横滚: 0, 俯仰: 0, 航向: 0.01
- 目标杆臂/m:** X: 0, Y: 0, Z: 0
- 初始姿态/deg:** 横滚: 2.704, 俯仰: 0.123, 航向: 0
- 里程计:**
 - 车轮直径/m: 1.2
 - 分辨率: 2048
 - 杆臂/m: X: 0, Y: 0.1, Z: 0
- GNSS:**
 - 系统: GPS (dropdown)
 - 观测类型: PC (dropdown)
 - 截止高度角/deg: 2
 - 姿态辅助安装角/deg: 横滚: 1.2, 俯仰: 24, 航向: 12
- 保存文件名:** C:\Users\chang\Desktop\new\182109.bin
- Buttons:** [发送] [关闭]

图 3.15 参数管理

现从左到右、由上而下依次对设置参数进行具体说明：

- 1) 串口号为导航输出串口的端口号；
- 2) 对准方式有 3 种：动态、静态、给定，请根据实际情况选择，其中动态对准要求载体速度持续达到 5m/s，静态要求至少为战术级组合导航系统；
- 3) 数据率是指实时导航结果的输出速率，有 5 种：100Hz、50Hz、20Hz、10Hz、5Hz；
- 4) 在紧组合模式下，如果勾选，将输出卫星高度角、方位角以及信噪比信息；
- 5) “刷新”图标按键在连接实时组合导航系统的情况下，用于获取系统当前配置；
- 6) 组合模式分为紧组合、松组合，请与实时系统选择匹配；
- 7) 导航场景分为车载、机载、船载，目前机载与船载没有区别；
- 8) 在车载情况下，可以启用里程计；
- 9) 启用航向辅助；
- 10) 如果勾选“至目标点”，导航结果将归算到目标点，否则，导航结果将归算到组合导航的几何中心；
- 11) 如果勾选“至载体系”，导航结果将归算到载体坐标系(前右下)，否则，导航

结果将归算到当地水平坐标系(北东地);

- 12) 天线杆臂输入;
- 13) 组合导航系统与载体的安装角输入;
- 14) 目标点杆臂输入, 在勾选“输出至目标点”的情况下起作用;
- 15) 组合导航系统初始姿态角, 在对准模式选择“给定”的情况下起作用;
- 16) 里程计的杆臂、分辨率以及车轮直径输入, 在勾选“启用里程计”的情况下起作用;
- 17) GNSS 参数设置, 设置解算使用 GNSS 系统、观测值类型以及截止高度角, 在组合模式选择“紧组合”的情况下起作用, 航向安装角在使用航向辅助时使用, 表示姿态辅助的安装偏差;
- 18) 选择导航结果存入的文件名, 当文件名为空时, 程序将不会将接收的数据存储在本地;
- 19) 在连接实时组合导航系统的情况下, 点击“发送”按钮, 可以将当前配置界面的配置参数写入系统。

程序的辅助工具介绍详见附录 6.4。

4 GNSS/INS 实时组合导航性能测试与验证

为了全面验证 GNSS/INS 实时组合导航的性能,本章于 2017 年 1 月 13 日进行了多组车载测试,在开阔天空(武汉市文化大道附近)以及城市环境中评估了该系统的导航性能。测试中除了使用本文开发完成的 GNSS/INS 实时组合组合导航系统 POS320,其主要性能指标如表 3.2 所示,还搭载了更高精度的光纤惯性组合导航系统 POS620,其主要性能参数如表 4.1 所示,将其后处理的结果作为参考基准。

表 4.1 POS620 的 IMU 主要性能参数

指标	数值
陀螺零偏/ $(^{\circ})h^{-1}$	0.1
角度随机游走/ $(^{\circ})h^{-1/2}$	0.03
加速度零偏/mGal	25
速度随机游走/ $(m)s^{-1}h^{-1/2}$	0.05

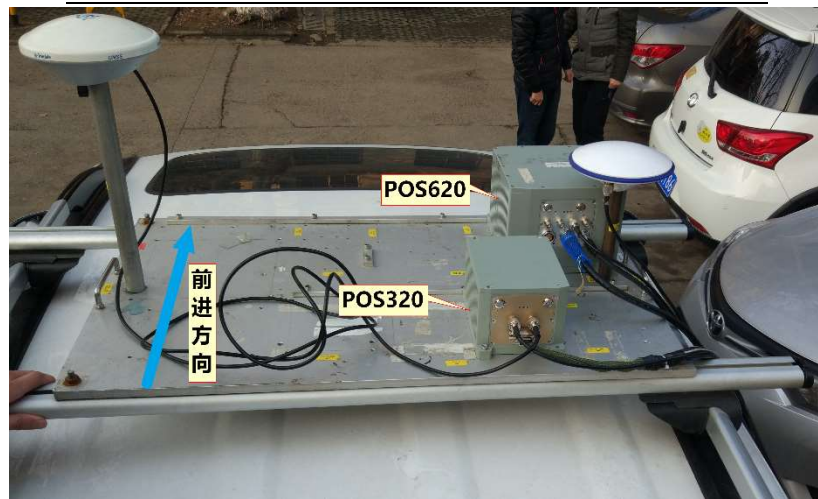


图 4.1 设备连接图



图 4.2 测试环境

4.1 导航性能评估方法

在有连续 GNSS 位置修正的情况下，组合导航滤波器一直利用 GNSS 位置进行观测更新，GNSS/INS 组合导航的精度，尤其是定位精度，主要由 GNSS 的位置精度决定^[55]。因此，GNSS/INS 组合导航系统的精度评估手段通常为模拟几段 GNSS 信号中断，即考察在 GNSS 充分修正后短时间纯惯性导航情况下的导航误差漂移是否和预期的 IMU 导航性能一致。

测试中，POS320 与 POS620 通过功分器接收同一个天线的信号。在开阔环境时，通过人为拔掉 POS320 的天线馈线模拟 GNSS 信号中断，每次中断时长约 1 分钟，中断结束后信号恢复大约 5 分钟，用来恢复状态估计，每组测试仿真了 5 次 GNSS 信号中断。

实时和后处理均采用松组合模式，把 GNSS 差分解算结果作为组合导航的位置修正信息。对于基站数据部分，实时处理中，POS320 通过 3G 上网卡使用 Ntrip 客户端获取武汉大学 GNSS 中心的基站数据，进行差分处理；后处理中，使用该基站的原始观测数据进行差分解算。基站的数据采样率为 1Hz，接收机型号为 Trimble NetR9。对于流动站部分，POS320 与 POS620 具有内置存储单元，可同时存储 IMU、GNSS 原始观测数据，POS320 的实时组合导航结果使用上位机程序存储在测试笔记本电脑中。2 款组合导航系统的 IMU 原始数据率均为 200Hz，GNSS 数据率均为 1Hz，POS320 的实时导航数据率最高为 100Hz，测试中使用 20Hz。

后处理采用武汉迈普时空导航科技有限公司开发的高精度 GNSS/INS 组合导航数据处理软件 GINS v2.0，利用基站的原始观测数据、POS620 的原始 GNSS 观测数据以及 IMU 的原始观测数据，得到 PPK(Post Processed Kinematic)/INS 松组合反向平滑的结果，并转化到 POS320 的中心作为本次测试的位置、速度和姿态的参考真值。

为了验证系统的实时性能，使用 DSP 片内计时器统计 250Hz 中断内从 IMU 原始数据获取到输出导航结果以及主函数中组合更新的运行耗时，并在导航结果中记录输出。统计本次测试中全部数据所记录的耗时，分析实时系统中的计算延迟。

4.2 测试实验

本章共进行了四次车载测试，开阔天空下 3 次，城市环境中 1 次。在开阔天空的测试中特意设计载体运动为田字格轨迹，有利于惯导的误差参数的估计，更好地评定惯性传感器性能。测试中将 GNSS 的差分处理的中定位状态划分为 4 种，利用 0、1、2、3 表示，分别对应 GNSS 定位处于单点定位、浮点差分解、固定差分解以及无效。

4.2.1 开阔环境中无车辆辅助的测试

本次测试为开阔环境中不使用车辆辅助的导航性能评估，测试轨迹如图 4.3 所示。从图 4.4 中可以看出，本次测试中 GNSS 定位状态基本都为固定差分解，参与定位的卫星保持在 15 颗左右，可见 GNSS 辅助信息良好。



图 4.3 测试轨迹

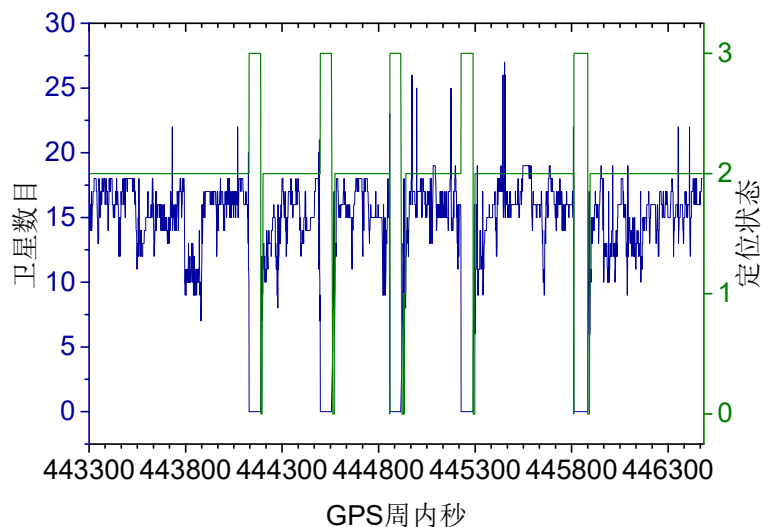


图 4.4 参与解算卫星数目以及定位状态

实时导航结果与参考真值的对比如图 4.5 所示,可以看出在 5 个卫星中断期间,位置漂移到达 1.5m 左右,速度误差达到 0.05m/s 左右,姿态误差与中断无明显关联,符合 POS320 的性能指标,与 POS320 后处理仿真 1min 中断的位置、速度漂移水平相当。从实时导航误差误差统计表 4.2 中可以看出,北向、东向、垂向的位置误差的标准差分别为 0.116、0.118、0.139m,北向、东向、垂向的速度误差的标准差分别为 0.007、0.007、0.006m/s,横滚、俯仰、航向的姿态误差的标准差分别为 0.011、0.007、0.067deg。

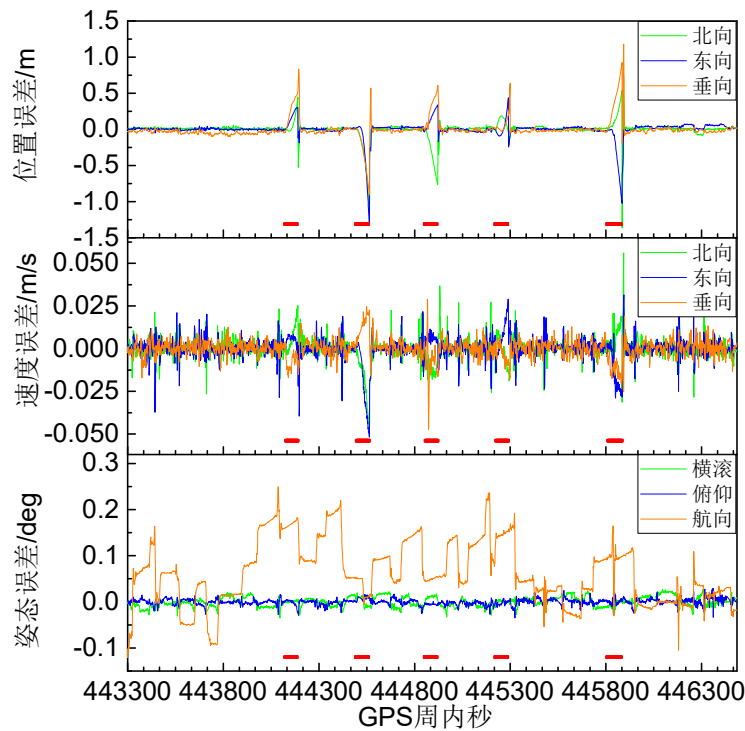


图 4.5 实时导航误差曲线
表 4.2 实时导航误差统计表

	位置误差 (m)			速度误差(m/s)			姿态误差(deg)		
	北向	东向	垂向	北向	东向	垂向	横滚	俯仰	航向
Max	1.364	1.299	1.183	0.056	0.052	0.047	0.035	0.035	0.249
Mean	0.000	0.000	0.002	0.000	0.000	0.000	0.000	0.000	0.063
Std	0.116	0.118	0.139	0.007	0.007	0.006	0.011	0.007	0.067

4.2.2 开阔环境中使用 NHC 辅助的测试

本次测试仍在开阔天空下,测试轨迹如图 4.6 所示,根据 4.2.1 测试中的导航结果,估计出 POS320 的安装误差角后,启用 NHC 辅助。由图 4.7 可以看出,本次测试 GNSS 状态与 4.2.1 测试中基本一致,可用卫星为 15 颗左右,定位状态

基本为固定差分分解。



图 4.6 测试轨迹

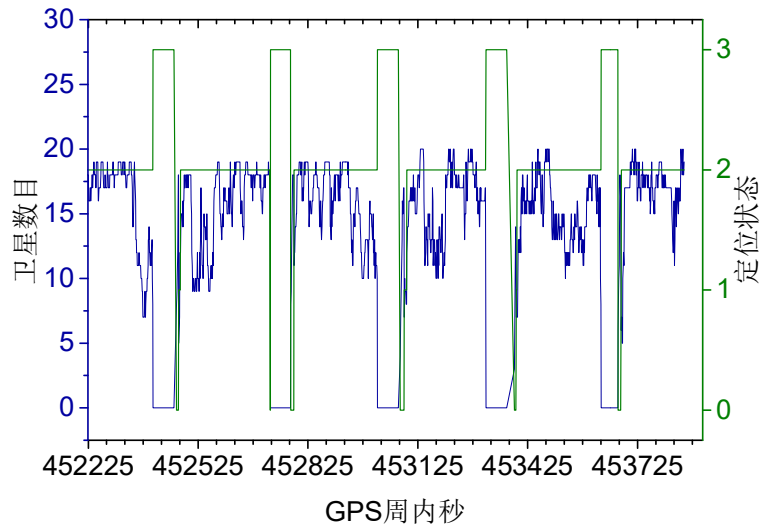


图 4.7 参与解算卫星数目以及定位状态

实时导航结果与参考真值的对比如图 4.8 所示,可以看出在 5 个卫星中断期间,位置漂移到达 0.8m 左右,速度误差达到 0.04m/s 左右,优于 4.2.1 测试中仅使用 GNSS 位置修正的情况,证实了实时 NHC 工作正常。从实时导航误差误差统计表 4.2 中可以看出,北向、东向、垂向的位置误差的标准差分别为 0.136、0.110、0.092m,北向、东向、垂向的速度误差的标准差分别为 0.008、0.008、0.005m/s,横滚、俯仰、航向的姿态误差的标准差分别为 0.011、0.007、0.056deg。

表 4.3 实时导航误差统计表

	位置误差(m)			速度误差(m/s)			姿态误差(deg)		
	北向	东向	垂向	北向	东向	垂向	横滚	俯仰	航向
Max	0.890	0.692	1.165	0.043	0.040	0.034	0.048	0.035	0.197
Mean	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Std	0.136	0.110	0.092	0.008	0.008	0.005	0.011	0.007	0.056

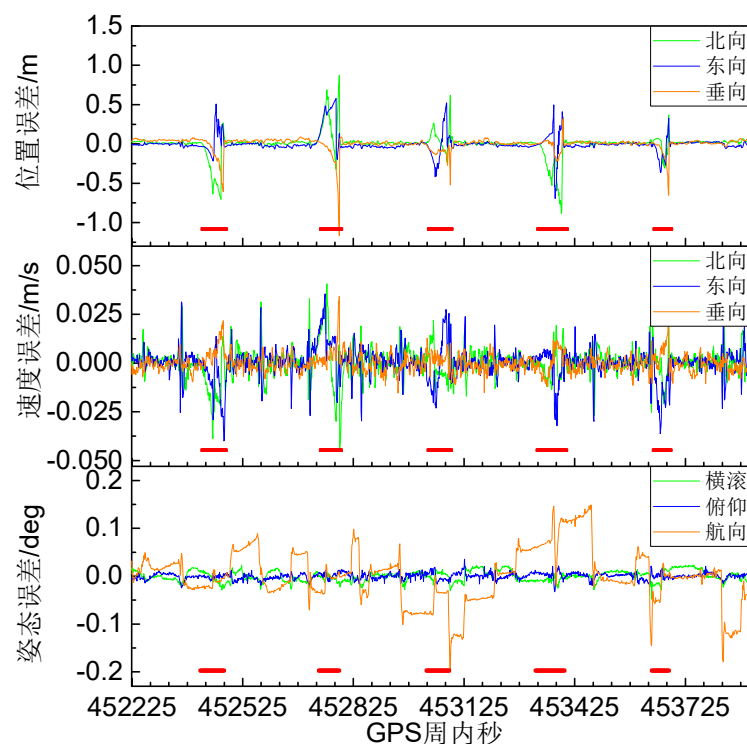


图 4.8 实时导航误差曲线

4.2.3 开阔环境中使用 NHC 及里程计辅助的测试

本次测试，在使用 NHC 的基础上，增加里程计辅助，测试轨迹如图 4.9 所示。由图 4.10 可以看出，GNSS 状态和前两次测试基本一致。



图 4.9 测试轨迹

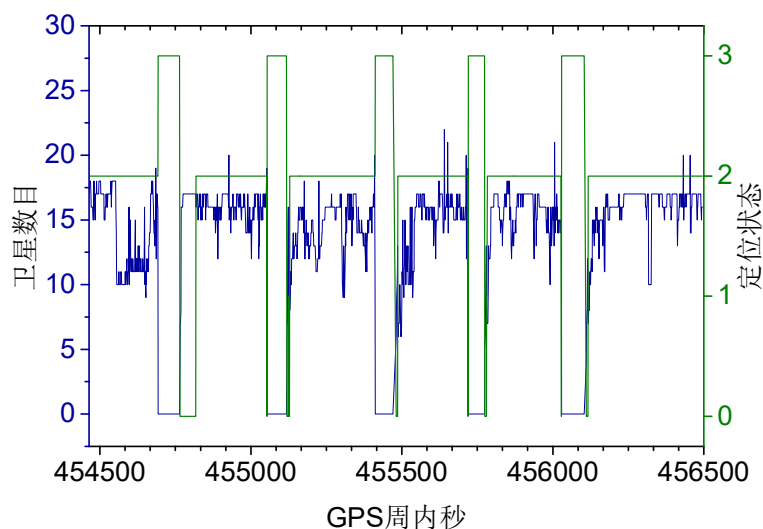


图 4.10 参与解算卫星数目以及定位状态

实时导航结果与参考真值的对比如图 4.11 所示，可以看出在 5 个卫星中断期间，位置漂移到达 0.5m 左右，速度误差达到 0.04m/s 左右，位置漂移水平略好于 4.2.2 测试中只使用 NHC 的情况，说明了里程计辅助在实时组合导航系统中的正常应用。从实时导航误差统计表 4.2 中可以看出，北向、东向、垂向的位置误差的标准差分别为 0.136、0.110、0.092m，北向、东向、垂向的速度误差的标准差分别为 0.008、0.008、0.005m/s，横滚、俯仰、航向的姿态误差的标准差分别为 0.011、0.007、0.103deg。

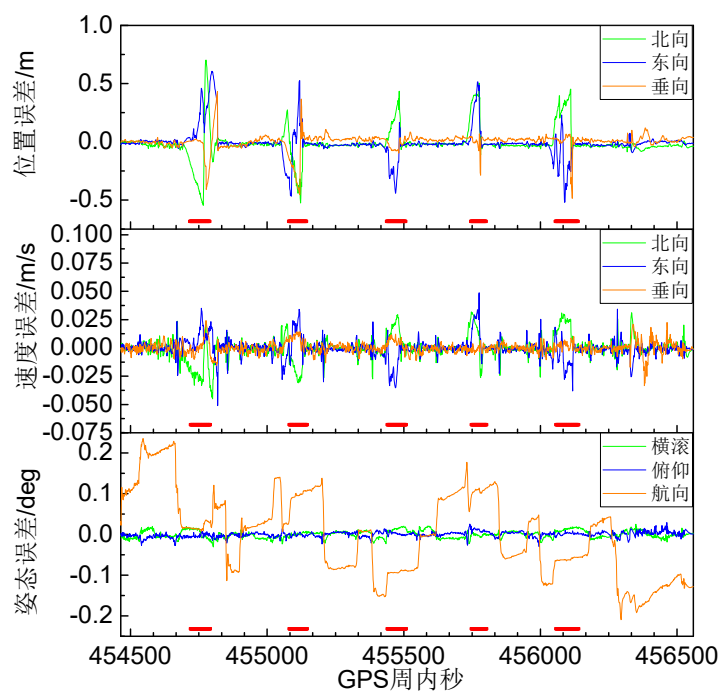


图 4.11 实时导航误差曲线

表 4.4 实时导航误差统计表

	位置误差(m)			速度误差(m/s)			姿态误差 deg		
	北向	东向	垂向	北向	东向	垂向	横滚	俯仰	航向
Max	0.702	0.606	0.485	0.045	0.051	0.033	0.031	0.030	0.235
Mean	-0.012	-0.009	0.003	0.000	0.000	0.000	0.000	0.000	0.000
Std	0.126	0.118	0.070	0.010	0.009	0.005	0.009	0.007	0.103

4.2.4 城市环境中的测试

最后一组测试为城市环境，从武汉市文化大道途径珞狮南路、珞狮路，然后由武汉大学茶港门进入学校，再从武汉大学工学部返回信息学部的卫星导航定位技术研究中心，测试轨迹如图 4.12 所示，途中多次经过高架桥、城市峡谷以及校园中的高大树木。由图 4.10 可以看出，可用卫星明显少于开阔天空的测试环境，大部分在 10 颗卫星左右，少部分仅有 5、6 颗可用卫星，GNSS 定位状态较多部分可以维持固定差分解状态，少部分处于浮点解和单点定位解状态，甚至有不可用状态。本次测试中启用 NHC、ZUPT 以及里程计三种车辆辅助。

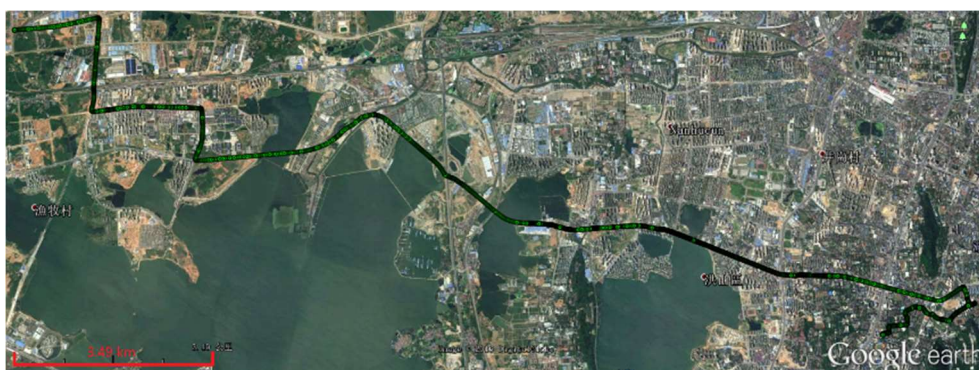


图 4.12 测试轨迹

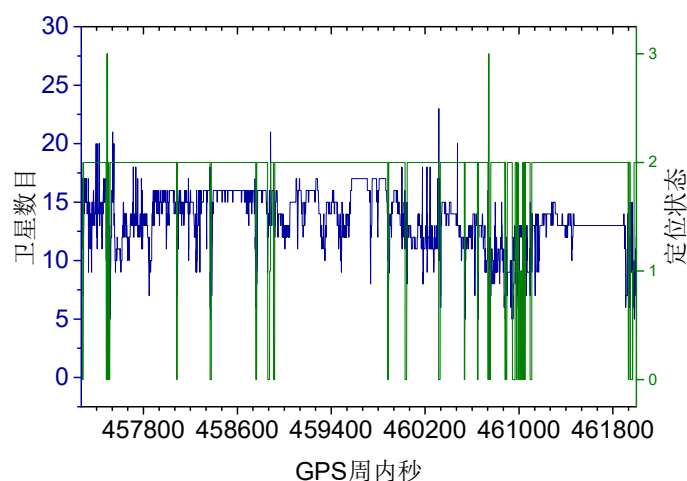


图 4.13 参与解算卫星数目以及定位状态

实时导航结果与参考真值的对比如图 4.14 所示，可以看出整个测试阶段，位置、速度误差与 GNSS 定位状态有密切关系，误差曲线表现为凸起。从实时导航误差统计表 4.1 中可以看出，北向、东向、垂向的位置误差的标准差分别为 0.046, 0.048, 0.047m，北向、东向、垂向的速度误差的标准差分别为 0.005, 0.006, 0.005m/s，横滚、俯仰、航向的姿态误差的标准差分别为 0.003, 0.004, 0.030deg。

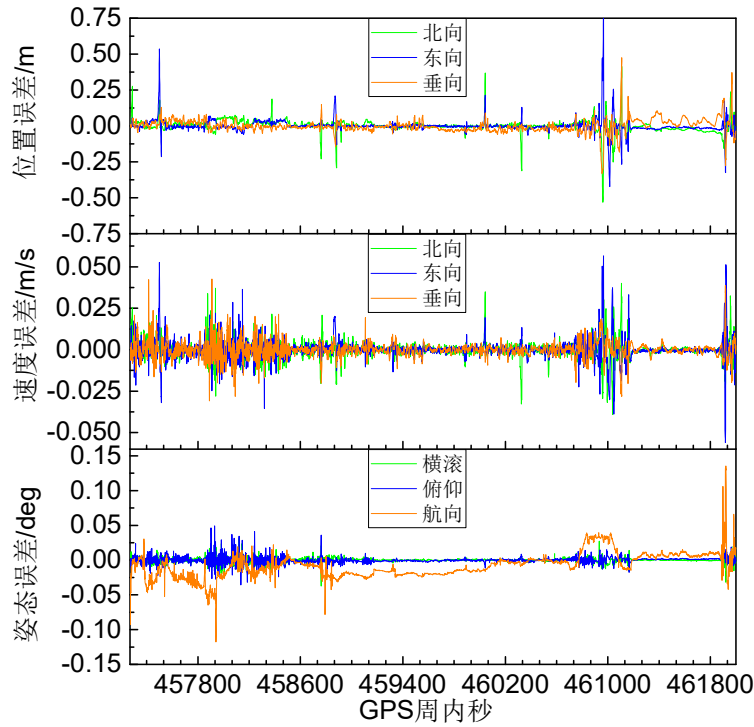


图 4.14 实时导航误差曲线

表 4.5 实时导航误差统计表

	位置误差(m)			速度误差(m/s)			姿态误差(deg)		
	北向	东向	垂向	北向	东向	垂向	横滚	俯仰	航向
Max	-0.532	0.826	0.475	0.040	0.057	0.043	-0.037	0.049	0.136
Mean	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Std	0.046	0.048	0.047	0.005	0.006	0.005	0.003	0.004	0.030

4.3 实时性能分析

将 4.2 中四次测试的时间记录结果一起进行统计，统计结果如图 4.15 所示，从图 a 中可以看出，250Hz 中断函数的执行时间大致可以分为 2 部分，95.7%左右的都大约是 1ms，剩余的大约为 3ms，延迟较小的部分完成的是基本机械编排

-Kalman 预测，延迟较大的部分是要同时处理组合更新量的状态转移以及反馈，可以认为 250Hz 中断函数可以及时处理 200Hz 采样的 IMU 原始数据；从图 b 中可以看出，组合更新耗时基本保持在 13ms 左右完成；从图 c 中可以看出，获取 GNSS 数据有明显的延迟，大约在 40ms 左右，时间延迟的差异可能是观测卫星数目不同以及基站数据网络传输延迟差异导致的；综合图 b 和图 c，可以看出 GNSS 延迟更新时间大约为 53ms，不会超过 100ms，完全可以在一个 GNSS 采样时间间隔内完成组合更新计算。由此看来，本文提出的改进 Kalman 滤波实时处理方案在实际测试中能够按照设计要求运行，可以满足 GNSS/INS 实时组合系统的实时性要求。

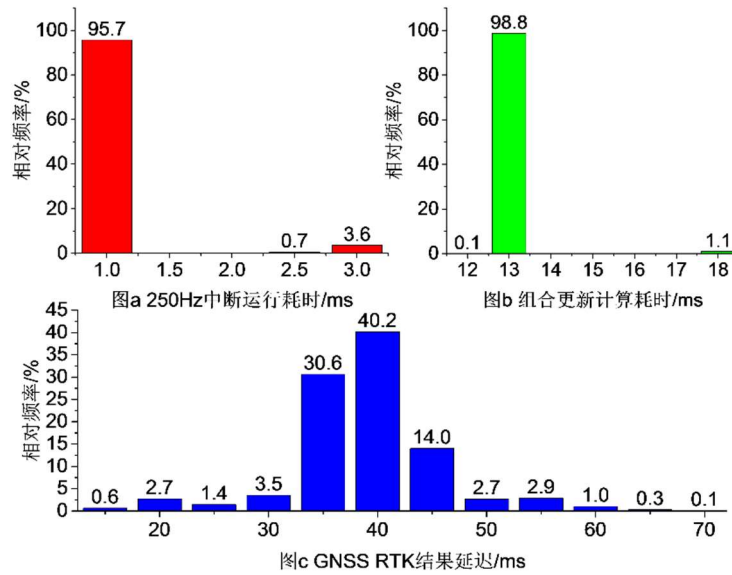


图 4.15 时间统计图

4.4 测试结论

本章通过四次车载测试，验证了本文开发完成的 GNSS/INS 实时组合导航系列的导航性能与精度。前三次测试环境为开阔天空，通过拔掉天线馈线，仿真中断 GNSS 信号，测试表明：实时导航系统在 1min 中断期间的漂移水平（位置漂移 1.5m 左右，速度漂移 0.05m/s 左右）符合 POS320 的器件水平，说明了实时组合导航系统工作正常；NHC 以及里程计辅助的加入，有效地改善了 GNSS 信号中断期间，位置、速度漂移的水平，1min 中断的位置漂移减少到 0.5m 左右，速度漂移减少到 0.04m/s 左右，说明了实时系统中车辆辅助功能的正确性。最后一组测试环境为普通城市环境，使用了 NHC、ZUPT 以及里程计辅助，测试结果表

明,位置标准差为 0.05m 左右,速度标准差为 0.005m/s 左右,横滚、俯仰标准差为 0.004deg,航向标准差为 0.03deg。GNSS 信号短期中断期间的惯导漂移误差及普通城市环境中的导航误差都表明,本文开发的 GNSS/INS 实时组合导航系统的导航精度能够满足自动驾驶、实时导航、智能机器人导航等实时应用场景的需求。通过对四次测试中的组合导航输出时间延迟的统计,验证了本文针对 GNSS/INS 实时组合中存在的 GNSS 数据获取延迟以及更新计算耗时较长的问题,从 Kalman 滤波基本原理出发,所提出的适用于 GNSS 观测及更新延迟的 Kalman 滤波数学模型对延迟更新处理的正确性;导航输出延迟可以保持为 1ms 左右,不会因为组合更新解算导致输出延迟增大,说明了本文开发的 GNSS/INS 实时组合导航系统的实时性能满足使用要求。综合以上测试,可以认为本文开发的 GNSS/INS 实时组合导航系统的导航精度及实时性达到了设计要求,能够满足应用场景的需求。

5 总结与展望

本文在介绍了 GNSS/INS 组合导航原理的基础之上,详细说明了基于 DSP 平台的 GNSS/INS 实时组合导航系统设计与实施方案。针对实时应用场景,本文对 GNSS/INS 组合导航算法进行了优化,尤其是对 GNSS/INS 实时组合导航系统中存在的 GNSS 数据延迟以及组合更新解算耗时过长的处理,基于 Kalman 滤波理论,提出了一种在完成了滞后的观测更新后将误差状态向量一步转移到当前时刻的 Kalman 滤波数学模型。该模型能够降低 GNSS 数据延迟和组合解算耗时对实时导航结果输出的影响,因此允许观测更新解算在比较充足的时间里完成,对组合导航算法在运算能力较低的处理器的实现和对实时性要求严格的应用场合有重要的意义。实际车载测试表明,本文开发完成的 GNSS/INS 实时组合导航系统的位置误差标准差为 0.05m,速度误差标准差为 0.005m/s,横滚、俯仰误差标准差为 0.004deg,航向误差标准差为 0.03deg,导航结果的输出延迟为 1ms 左右,车辆辅助功能正常,能够保证实时应用场景的实时性与导航精度的要求。此外,为了方便用户与实时组合导航系统的交互,本文开发了 Windows 平台上的客户端应用程序。本导航系统已在郑州宇通集团有限公司、武汉光庭信息技术股份有限公司及武汉环宇智行科技有限公司的部分自动驾驶项目组得到应用,能够满足车辆自动驾驶的实时导航需求。

需要进一步研究和完善的工作有:

- 1) 本导航系统是针对车载环境下的中高精度惯性导航设备。有必要加入机载、船载导航场景的支持,并针对 MEMS 惯性导航设备,改进相关车辆辅助,比如零速判断准则、静止条件下的航向锁定等。
- 2) 即使使用车辆辅助,本导航系统仍无法长时间在室内或城市峡谷等苛刻信号条件下进行精密定位,对 GNSS 有较多依赖,当 GNSS 信号严重受到限制时,本导航系统将无法保障导航精度。因此,有必要进行多传感器组合方面的研究,在 GNSS/INS 组合导航系统的基础上,增加具有自主定位能力的传感器,比如激光雷达、深度相机、双目相机、全景相机^[56-59]等,开发出一套在卫星信号缺失的情况下,仍能提供厘米级定位的室内外无缝实时组合导航系统。

6 附录

6.1 导航状态数据格式

开始字节	长度	类型	定义
0	1	Unsigned Char	字头, 0xAA
1	1	Unsigned Char	字头, 0x55
2	1	Unsigned Char	数据长度
3	1	Unsigned Char	组合模式: 0-紧组合(TC), 1-松组合(LC)
4	2	Short	GPS Time: GPS week number
6	4	Long	GPS Time: GPS millisecond of week
10	1	Unsigned Char	观测卫星个数 N
11	8	Double	纬度/deg
19	8	Double	经度/deg
27	4	Float	高程/m
31	4	Float	北向速度/m/s
35	4	Float	东向速度/m/s
39	4	Float	地向速度/m/s
43	4	Float	横滚/deg
47	4	Float	俯仰/deg
51	4	Float	航向/deg
55	2	Short	北向加速度/m/s ² , 10 ⁻³
57	2	Short	东向加速度/m/s ² , 10 ⁻³
59	2	Short	地向加速度/m/s ² , 10 ⁻³
61	2	Short	横滚角速度/deg/s, 10 ⁻³
63	2	Short	俯仰角速度/deg/s, 10 ⁻³
65	2	Short	航向角速度/deg/s, 10 ⁻³
67	1	Unsigned Char	高 2 位: 表示解算状态, 0-机械编排, 非 0-组合解算 低 6 位: LC 时, 表示 GNSS 定位模式, 0-SPP, 1-RTK float, 2-RTK fixed, 3-No GNSS; TC 时, 表示使用的卫星个数 n
每个卫星信息 (i 从 0 开始, 共 n 个, 只在 TC 时使用)			
68+6*i	1	Unsigned Char	PRN(GPS:0~32;BDS:35~70)
69+6*i	1	Unsigned Char	高度角/deg
70+6*i	2	Unsigned Short	方位角/deg
71+6*i	1	Unsigned Char	信噪比 S1/dB.Hz
72+6*i	1	Unsigned Char	信噪比 S1/dB.Hz
68+6*n	1	Unsigned Char	检校和, 除字头以外的和

6.2 导航参数配置格式

开始字节	长度	类型	定义	
0	1	Unsigned Char	字头:	0x55
1	1	Unsigned Char	字头:	0xAA
2	4	Unsigned Long	版本:	高 20 位
			预留:	低 12 位, 设置为 0
6	1	Unsigned Char	是否输出为车辆系:	高 1 位, 0-否,1-是
			是否输出至目标点:	中 1 位, 0-否,1-是
			是否使用航向辅助:	中 1 位, 0-否,1-是
			是否输出卫星信息*:	中 1 位, 0-不输出,1-输出
			观测值类型*:	中 2 位, 0-C/A,1-P1,2-P2,3-PC
			系统*:	低 2 位, 0-GPS,1-BDS,2-GPS+BDS
7	1	Unsigned Char	模式:	高 2 位, 0-车载,1-机载,2-船载
			是否使用里程计:	中 1 位, 0-否,1-是
			对准模式:	中 2 位, 0-动态对准,1-静态对准,2-给定
			输出率/Hz:	低 3 位, 0-100,1-50,2-20,3-10,4-5
8	1	Unsigned Char	截止高度角*/deg	
9	4	Float	天线杆臂 X/m	
13	4	Float	天线杆臂 Y/m	
17	4	Float	天线杆臂 Z/m	
21	4	Float	安装角, 横滚/deg	
25	4	Float	安装角, 俯仰/deg	
29	4	Float	安装角, 航向/deg	
33	4	Float	初始姿态角, 横滚/deg	
37	4	Float	初始姿态角, 俯仰/deg	
41	4	Float	初始姿态角, 航向/deg	
45	4	Float	目标杆臂 X/m	
49	4	Float	目标杆臂 Y/m	
53	4	Float	目标杆臂 Z/m	
57	4	Float	里程计杆臂 X/m	
61	4	Float	里程计杆臂 Y/m	
65	4	Float	里程计杆臂 Z/m	
69	2	Unsigned Short	里程计比例因子	
71	4	Float	车轮直径/m	
75	4	Float	姿态辅助安装角, 横滚/deg	
79	4	Float	姿态辅助安装角, 俯仰/deg	
83	4	Float	姿态辅助安装角, 航向/deg	
87	1	Unsigned Char	检校和, 除字头以外的和	
*: 仅在紧组合模式下有效				

6.3 交互命令说明

- 1) RTTC、RTLTC
DSP 程序初始化完成，等待设置参数输入。
- 2) V
上位机与固件版本不一致，重置 FLASH 配置失败。
- 3) F
重置 FLASH 配置成功。
- 4) U
读取 FLASH 参数失败。
- 5) B
读取 FLASH 参数成功，导航解算开始。
- 6) W
等待。
- 7) P
定位解算中。
- 8) S
静态对准中，完成 1min 的数据获取，共需要 5min 数据。
- 9) D
仅在静态粗对准模式中，载体不满足静止条件，开始重新对准。
- 10) E
初始对准完成，进入组合导航解算中。
- 11) STOP
外部输入命令，进行 DSP 程序重启。
- 12) 0x55 0xBB
外部输入命令，用以获取 DSP 程序现有配置。

6.4 PC 端程序辅助工具

6.4.1 解码

在菜单中选择“处理(P)”-“解码(D)”，弹出图 6.1 所示的对话框，将保存的实时导航结果文件转化为文本文件以及 Google Earth 文件，以便方便查看结果，对于转化时可以设置轨迹的颜色(选择无时，不显示轨迹)、点的颜色、是否添加时间轴、是否以 KMZ 的压缩形式保存以及控制输出率(方便控制结果点的密集度，避免点过密导致 Google Earth 无法加载)。



图 6.1 解码工具

6.4.2 估计 IMU 安装误差角

在菜单中选择“处理(P)”-“安装角估计(A)”，弹出图 6.2 安装角估计工具所示的对话框，将保存的实时导航结果文件输入，设置估计安装角的时间段，点击解算即可得到 IMU 的安装误差角以及估计误差曲线。



图 6.2 安装角估计工具

6.4.3 固件升级

在菜单中选择“帮助(H)”-“升级(U)”，弹出图 6.3 所示的对话框，选择实时组合导航系统配置串口以及正确的固件文件，点击“发送”按钮进行实时组合导航系统固件升级。

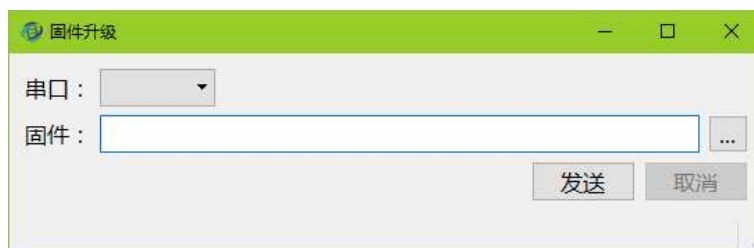


图 6.3 固件升级

参考文献

- [1] 高钟毓.2012.惯性导航系统技术[M].清华大学出版社.
- [2] Cox D B.1978.Integration of GPS with inertial navigation systems[J].Navigation,25(2): 236-245.
- [3] 张炎华, 王立端, 战兴群, 等.2008.惯性导航技术的新进展及发展趋势[J].中国造船:49(B10): 134-144.
- [4] 李征航, 黄劲松.2010.GPS 测量与数据处理[M].武汉: 武汉大学出版社.
- [5] 高法钦, 谈展中.2007.北斗/惯导组合导航算法性能分析[J].系统工程与电子技术:(07): 1149-1154.
- [6] 秦永元.2000.卡尔曼滤波与组合导航原理[M].西北工业大学出版社, 33-56.
- [7] 杨明.2006.无人自动驾驶车辆研究综述与展望[J].哈尔滨工业大学学报:38(8).
- [8] 李玉寰.2010.基于 DSP 和 FPGA 的机载导航计算机设计[D]:南京: 南京理工大学
- [9] 李海洋.2005.基于 DSP 和 FPGA 的多功能嵌入式导航计算机系统设计[D]:南京航空航天大学
- [10] 张复宜.2012.SINS/GPS 组合导航双 DSP 硬件平台的设计研究[D]:哈尔滨工程大学
- [11] 杨侃.2009.基于 DSP 和 FPGA 导航计算机硬件电路研究与设计[D]:哈尔滨工程大学
- [12] 王光辉.2013.基于时滞多传感器数据的信息融合滤波[D]:[硕士].黑龙江:黑龙江大学控制理论与控制工程, 90.
- [13] Yao Y, Xu D, Yan W. Cooperative localization with communication delays for MAUVs, 2009[C]. IEEE.
- [14] 卢晓.2008.观测时滞系统的最优滤波[D]:大连:大连理工大学
- [15] 李林林.2012.测量数据延迟下的不完全量测滤波研究[D]:[硕士].南京:南京理工大学, 76.
- [16] 刘川川.2012.基于伪距/多普勒的GPS/INS紧组合算法实现与抗差性能研究[D]:武汉:武汉大学
- [17] 吕京建, 肖海桥. 嵌入式处理器分类与现状[Z]. 2000.
- [18] 张辉, 胡广书.2004.DSP 的特点, 发展趋势与应用[J].电子产品世界:(05A): 35-37.
- [19] 廖娜.2008.DSP 应用技术综述[J].科技信息:(32): 78.
- [20] TMS C X Optimizing C.1998.Compiler User's Guide, SPRU187, Texas Instruments[J].Inc., Dallas, Tx.
- [21] 程亚豪.2013.伪距差分北斗/惯导紧组合算法实现与GNSS数据质量控制研究[D]:武汉:武汉大学
- [22] Shin E.2005.Estimation techniques for low-cost inertial navigation[D]:[20219].Calgary:The University of Calgary.
- [23] Savage P G.1998.Strapdown inertial navigation integration algorithm design part 2: Velocity and position algorithms[J].Journal of Guidance, Control, and Dynamics,21(2): 208-221.
- [24] Bortz J E.1971.A new mathematical formulation for strapdown inertial navigation[J].IEEE transactions on aerospace and electronic systems(1): 61-66.
- [25] Groves P D.2013.Principles of GNSS, inertial, and multisensor integrated navigation systems[M].Artech house.
- [26] Knight D T. Achieving modularity with tightly-coupled GPS/INS, 1992[C]. IEEE.
- [27] Zhao H S, Miao L J, Shen J, et al.2014.High accuracy algorithm for SINS/Odometer integrated navigation system[J].Acta Armamentarii,35(4): 433-440.
- [28] 孙德波, 杨功流, 张玉堂.2005.SINS/GPS 组合导航序贯滤波算法[J].中国惯性技术学

- 报:13(1): 59-63.
- [29] Grewal M S, Andrews A P, Filtering A K.2001.Theory and practice using Matlab[J].Wiley-Interscience, Canada.
- [30] Shin E.2003.Accuracy improvement of low cost INS/GPS for land applications[D]:[20219].Calgary: The University of Calgary.
- [31] Groves P D.2013.Principles of GNSS, inertial, and multisensor integrated navigation systems[M]. Artech house.
- [32] 朱春云, 庄广琛, 宋雅兰, 等.2011.DVL安装角估计方法在水下组合导航系统中的应用[J].中国惯性技术学报:(01): 75-78.
- [33] 高钟毓, 王进, 董景新, 等.1995.惯性测量系统零速修正的几种估计方法[J].中国惯性技术学报:03(02): 24-29.
- [34] 方靖, 顾启泰, 丁天怀.2008.车载惯性导航的动态零速修正技术[J].中国惯性技术学报:16(03): 265-268.
- [35] 刘帅.2012.GPS/INS组合导航算法研究与实现[D]:解放军信息工程大学
- [36] Smith S C, Seiler P.2003.Estimation with lossy measurements: jump estimators for jump systems[J].IEEE Transactions on Automatic Control,48(12): 2163-2171.
- [37] 杜瑾, 李杰, 罗丹瑶, 等.2016.高动态GPS/INS组合导航中时间延迟软硬件 补偿算法研究[J].传感技术学报:29(12): 1858-1863.
- [38] Trimble. 2006.Trimble BD9xx GNSS Receivers Help[EB/OL].Trimble Navigation Limited,http://www.trimble.com/OEM_ReceiverHelp/v5.11/en/default.html.
- [39] POR P. ARM Cortex-M4 32b MCU+ FPU, 225DMIPS, up to 2MB Flash/256+ 4KB RAM, USB OTG HS/FS, Ethernet, 17 TIMs, 3 ADCs, 20 comm. interfaces, camera & LCD-TFT[EB/OL].<http://www.st.com/content/ccc/resource/technical/document/datasheet/03/b4/b2/36/4c/72/49/29/DM00071990.pdf/files/DM00071990.pdf/jcr:content/translations/en.DM00071990.pdf>.
- [40] 16-Channel, 24-Bit Analog-to-Digital Converter[EB/OL].<http://www.ti.com/lit/ds/symlink/ads1258.pdf>.
- [41] TMS320C674x DSP CPU and Instruction Set[EB/OL].www.ti.com/lit/ug/sprufe8b/sprufe8b.pdf.
- [42] 刘帅, 孙付平, 陈坡, 等.2012.GPS/INS组合导航系统时间同步方法综述[J].全球定位系统:(01): 53-56.
- [43] 李彩容.2013.基于FPGA的GNSS/INS组合系统时间同步设计[J].现代科学仪器:(02): 92-94.
- [44] TMS320C6745/C6747 DSP Technical Reference Manual[EB/OL].www.ti.com/lit/ug/spruh91b/spruh91b.pdf.
- [45] Forsberg C.1988.Xmodem/Ymodem Protocol Reference[J].online]. http://www.commonsoftinc.com/Babylon_Cpp/Documentation/Res/KYModem.htm.
- [46] TMS320C674x DSP Megamodule Reference Guide[EB/OL].www.ti.com/lit/ug/sprufk5a/Sprufk5a.pdf.
- [47] 姚章俊, 陈蜀宇, 卢尧.2012.一种高性能环形缓冲区的研究与实现[J].计算机工程:(08): 228-231.
- [48] Blanchette J, Summerfield M.2006.C++ GUI programming with Qt 4[M].Prentice Hall Professional.
- [49] Summerfield M.2010.Advanced Qt Programming: Creating Great Software with C++ and Qt 4[M].Pearson Education.
- [50] QextSerialPort[EB/OL].<https://github.com/qextserialport/qextserialport>.

- [51] Rathmann U, Wilgen J.2016.Qwt-Qt Widgets for technical applications[J].Qwt-Qt Widgets for Technical Applications.
- [52] Boulos M N K.2005.Web GIS in practice III: creating a simple interactive map of England's strategic Health Authorities using Google Maps API, Google Earth KML, and MSN Virtual Earth Map Control[J].International journal of health geographics,4(1): 22.
- [53] MA L, CAO Y.2010.Application of Google Earth COM API and KML in Tourism Information System Development [J][J].Journal of Geo-Information Science,6: 90-96.
- [54] Du Y, Yu C, Liu J. A study of GIS development based on KML and Google Earth, 2009[C]. IEEE.
- [55] Niu X, Nassar S, EL-SHEIMY N.2007.An Accurate Land-Vehicle MEMS IMU/GPS Navigation System Using 3D Auxiliary Velocity Updates[J].Navigation,54(3): 177-188.
- [56] Tong G, Wu Z, Tan J. Omni-vision mobile robot vSLAM based on spherical camera model, 2011[C]. IEEE.
- [57] 詹文强, 陈绪兵, 陈凯, 等.2016.基于深度相机的机器人室内定位技术研究[J].信息通信:(1): 25-26.
- [58] 焦富龙.2015.基于双目视觉的移动机器人 SLAM 系统设计与实现[D]:北京工业大学
- [59] Kim J, Sukkarieh S. SLAM aided GPS/INS navigation in GPS denied and unknown environments, 2004[C].

致谢