

融合 IMU 与视觉的移动机器人位姿估计

作者姓名： 杨冠群
指导教师： 楚好 讲师
单位名称： 信息科学与工程学院
专业名称： 自动化

东 北 大 学

2017 年 6 月

Pose Estimation of Mobile Robots Based on the Integration of IMU and Vision

by Guan-Qun Yang

Supervisor: Lecturer Chu Hao

Northeastern University

June 2017

毕业设计（论文）任务书

毕业设计（论文）题目：

融合 IMU 与视觉的移动机器人位姿态估计

设计(论文)的基本内容：

移动机器人位置和姿态的精确估计对于人与机器人交互和机器人自主导航具有重要的意义。利用来自多个传感器的信息对移动机器人进行位姿估计可以互相提供冗余信息，提高解算精度。本课题拟融合 IMU 和 RGB-D 传感器信息，对移动机器人位姿估计问题进行研究。

1. 分别利用 IMU 和 RGB-D 传感器信息对移动机器人位姿进行估计
2. 利用卡尔曼滤波算法融合 IMU 和视觉信息进行位姿估计。
3. 利用 IMM-KF 算法优化 EKF 融合算法。
4. 利用 1-3 三种方法进行实验。
5. 总结研究成果，撰写毕业论文。

毕业设计（论文）专题部分：

题目： _____

设计或论文专题的基本内容：

学生接受毕业设计（论文）题目日期

第 周

指导教师签字：

年 月 日

融合 IMU 与视觉的移动机器人位姿估计

摘要

移动机器人位姿估计是解决自主移动机器人定位问题的核心。在室内环境中，由于 GPS 无法使用，通常采用惯性测量单元（IMU）和视觉等相对位姿估计方法。然而，单一传感器所获得的机器人位姿数据，容易受到环境干扰和移动机器人自身的影响，往往呈现出精度低、可靠性弱等问题。因此，多传感器融合引起了国内外研究机构的关注。本课题将视觉与 IMU 这两类传感器信息在滤波框架下进行融合，以提升移动机器人位姿估计的精度。

首先，本文针对单传感器位姿解算系统进行实验，通过实验证明由于漂移和两次积分引起的 IMU 位姿解算误差以及受到特征缺失和光照影响的 RGB-D 视觉里程计误差使得单传感器系统不能满足精度要求。

其次，本文提出了在卡尔曼滤波框架下的融合滤波算法，通过对移动机器人建立线性和非线性模型，选取观测变量并对系统运动过程中的噪声进行描述，融合 IMU 的偏航角信息和基于 Kinect 的 RGB-D 视觉里程计的位置信息，使得移动机器人在环境特征缺乏条件下仍然可以获得正确的位姿。

最后，针对在卡尔曼滤波框架下机器人位姿解算结果受到模型正确性影响的问题，本文提出使用包含多个运动模型的交互多模型滤波器，通过设计子运动模型并指定 Markov 矩阵和模型概率，利用机器人多个运动模型的交互在实现机器人在运动模式切换的同时描述其运动的模型的自适应切换，从而实现针对机器人运动的精确描述，实验表明交互多模型滤波器可以在每一个采样周期上对位姿的漂移进行校正，进一步提升移动机器人位姿解算的精度。

通过视觉与 IMU 信息的在滤波框架下进行融合可以克服单传感器位姿解算系统存在的问题，提升移动机器人位姿解算的精度。

关键词：位姿估计；RGB-D；IMU；卡尔曼滤波算法；交互多模型算法

Pose Estimation of Mobile Robots Based on the Integration of IMU and Vision

Abstract

Pose estimation of mobile robots is pivotal in solving the localization problem. Due to the unavailability of GPS in indoor environment, IMU- and vision-based methods are often utilized. However, the pose acquired with data coming from single sensor is prone to the effects caused by environmental perturbation and mobile robots themselves and are therefore lacking in accuracy and robustness. As a result, multi-sensor fusion problem has drawn attention globally. This project will fuse the vision and IMU information in the filtering framework in order to improve the accuracy of the pose estimation system.

First and foremost, experiments have been conducted for single-sensor estimator. It turns out that estimator may fail insofar as accuracy, which results from IMU measurement drift and double integration, feature loss and photometric error respectively.

Secondly, fusion algorithm under the Kalman filtering framework has been put forward. Through the linear and non-linear model of robot and the subsequent selection of observer variable and noise parameter, the orientation information from IMU and position information from RGB-D odometry have been fused and correct pose is still made possible where there exists feature loss.

Consequently, in order to tackle the single model influence towards tracking accuracy, an Interactive Multiple Model (IMM) consisting different kinematic models has been used. After sub-model design and Markov transition matrix assignment, models adaptively switch among each other when robot mode changes. Therefore, accurate description of robot movement is acquired. Empirical analyses of pose estimation based on IMM estimator suggest that pose drift could be corrected during each sampling period and pose accuracy has been further improved.

The fusion of both visual and inertial information under the filtering framework could overcome problems which single-sensor system have and improve the accuracy of pose estimation.

Keywords: Pose Estimation; RGB-D; IMU; Kalman Filter; Interactive Multiple Model (IMM)

目录

毕业设计（论文）任务书.....	I
摘要.....	II
ABSTRACT.....	III
第一章 绪论.....	1
1.1 课题背景，目的与意义.....	1
1.2 国内外研究现状与发展趋势.....	3
1.2.1 基于 RGB-D 的位姿估计.....	3
1.2.2 融合 IMU 和 RGB-D 的位姿估计.....	4
1.2.3 IMM 融合滤波算法.....	5
1.3 主要研究内容和章节安排.....	6
第二章 基于 IMU 和 KINECT 的移动机器人位姿估计.....	9
2.1 IMU 位姿解算.....	9
2.1.1 惯性导航系统.....	9
2.1.2 姿态表征.....	9
2.1.3 位姿解算.....	12
2.2 KINECT 位姿解算.....	14
2.2.1 相机模型.....	14
2.2.2 点云生成.....	14
2.2.3 特征提取和配准.....	15
2.2.4 点云拼接.....	17
2.2.5 视觉里程计.....	17
2.3 实验及结果分析.....	18
2.3.1 IMU 位姿解算实验.....	18
2.3.2 Kinect 位姿解算实验.....	20
2.4 本章小结.....	22
第三章 基于卡尔曼滤波的 IMU 和 KINECT 融合位姿估计.....	24
3.1 卡尔曼滤波算法.....	25
3.1.1 线性系统的卡尔曼滤波.....	25
3.1.2 非线性系统的扩展卡尔曼滤波.....	26
3.2 移动机器人建模.....	27
3.2.1 线性模型.....	28
3.2.2 非线性模型.....	28

3.3 融合 IMU 和 KINECT 的卡尔曼滤波器.....	31
3.3.1 状态向量和系统模型.....	31
3.3.2 观测向量和观测模型.....	34
3.3.3 协方差矩阵.....	34
3.4 实验及结果分析.....	37
3.4.1 匀速闭环实验.....	37
3.4.2 匀加速闭环实验.....	39
3.4.3 实验结果分析.....	41
3.5 本章小结.....	42
第四章 基于 IMM-KF 的 IMU 与 KINECT 融合位姿估计.....	42
4.1 交互多模型滤波算法.....	43
4.1.1 系统描述.....	43
4.1.2 多模型方法.....	44
4.1.3 交互多模型算法.....	45
4.2 融合 IMU 与 KINECT 的 IMM-KF 滤波框架.....	46
4.2.1 机动指数.....	46
4.2.2 KF 子模型数量选择.....	47
4.2.3 KF 子模型类型选择.....	48
4.2.4 KF 子模型建立.....	49
4.2.5 IMM 模型概率向量和状态转移矩阵设计.....	50
4.3 融合位姿估计实验与结果分析.....	51
4.3.1 位姿估计系统的精度评价.....	51
4.3.2 融合位姿估计实验.....	52
4.3.3 实验结果分析.....	54
4.4 本章小结.....	54
第五章 总结与展望.....	54
5.1 总结.....	55
5.2 展望.....	55
参考文献.....	56
致谢.....	61

第一章 绪论

1.1 课题背景，目的与意义

随着机器人技术在工业生产和人民生活中不同应用领域的不断拓展和深入，人们对机器人的需求量持续增加。根据国际机器人联盟（IFR, International Federation of Robots）的测算，在世界范围内未来三年将由超过 1500 万台服务机器人投入使用。作为一类重要的机器人，移动机器人由于其灵活性以及对环境的适应性而得到了广泛的应用，近年来随着机器人技术的发展，各类移动机器人在国民经济的诸多领域中起到日益重要的作用，在航空航天，土地管理，城市规划，资源勘查以及海洋研究等国民经济的各个重要部分都能够看到移动机器人技术的应用。由于移动机器人技术在国家的经济和发展中起到的作用，移动机器人技术的价值逐渐被世界各国的研究者所注意到，他们着手进行与移动机器人技术有关的研究工作。

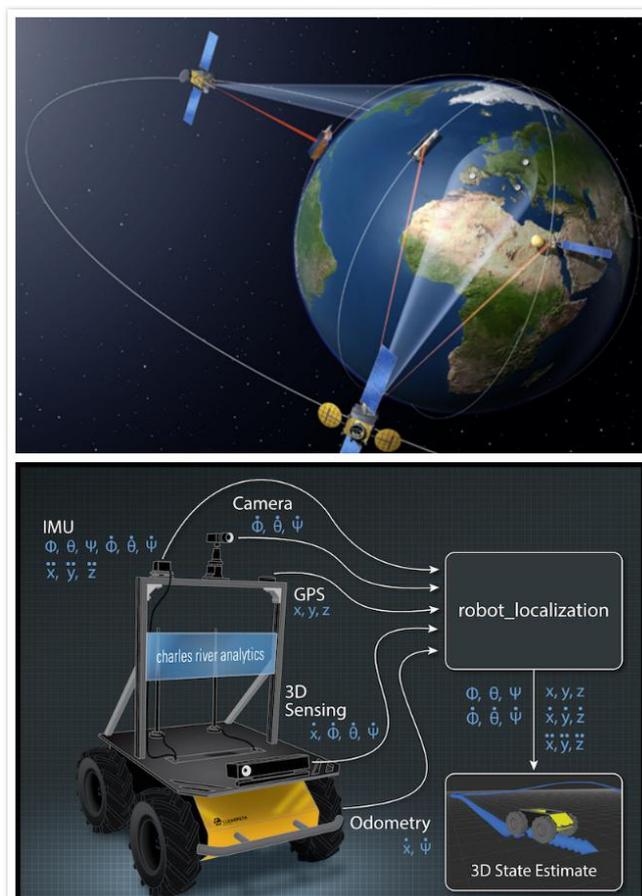


图 1.1 绝对位姿估计和相对位姿估计

如图 1.1 所示，位姿估计方法根据定位方式的不同分为绝对和相对位姿估计两类。绝对位姿估计需要获取预先设置的环境信息，虽然精度相对于相对位姿估计更高，但是

由于这种方法需要额外的信息，在很多应用场景下这种方法不能使用。相对位姿估计具有需要信息较少的特点，所以受到了研究者的注意^[1]。如图 1.2 所示，相对位姿估计可以通过惯性测量单元，RGB-D 传感器，激光雷达等多种方式实现。虽然取得位姿相对容易，但是这种方法没有利用其他的参考信息，属于一种通过开环获取数据的方法，且在两次积分过程当中也会造成误差的传播和累积，求得的位姿与实际值相比有较大的误差。通过利用由彩色照相机和深度照相机组成的 RGB-D 传感器同时得到移动机器人周围环境的 RGB 彩色图和深度图，利用相邻两帧图像，通过点云生成，点云配准等步骤即取得 RGB-D 传感器的平移和旋转信息。由于 RGB-D 传感器固定在机器人上，所以其平移和旋转也是机器人的平移和旋转，这样就获得了机器人的位置和姿态。虽然获得的位姿相对准确，但是由于获取平移和旋转的过程依赖于环境中特征的多少，对于特征缺乏的场景，这种方法很可能会失效。

考虑到 IMU 和 RGB-D 传感器进行移动机器人位姿解算的特点，本课题将这两种方法进行结合，从而达到在特征丰富和特征缺乏的场景下都能够实现移动机器人位姿解算的目的，以增强移动机器人进行位姿估计的鲁棒性。对于在复杂环境下运行的移动机器人，本课题的研究无疑具有积极的意义。



图 1.2 惯性测量元件，RGB-D 传感器和激光雷达

1.2 国内外研究现状与发展趋势

1.2.1 基于 RGB-D 的位姿估计

由于 IMU 在测量过程中的漂移问题，当前通常不单独使用 IMU 进行导航。利用 RGB-D 传感器进行导航的成果较多。

在 RGB-D 里程计的设计和测试方面。慕尼黑工业大学的 Jurgen Sturm 等人^[2]利用 Kinect 和高精度运动采集装置获取了在室内生活场景下以及在室内工业现场场景下具有 6 自由度相机位姿的基准 RGB-D 数据集，并基于此数据集提出了分别用于评价 RGB-D SLAM 系统以及视觉里程计精度的两个重要指标，为 RGB-D SLAM 以及视觉里程计的研究提供了依据。维拉诺瓦大学的 Joshua Fabian 等人^[3]通过量化 Kinect 在特征定位误差，分析 Kinect 的噪声以及噪声在实现整个视觉里程计过程中的传播规律，并利用误差对 RGB-D 里程计在 Kinect 出现特征定位误差条件下利用误差赋权方法来降低由于 Kinect 受到环境影响所带来的里程计误差。实验表明在没有离群点拒绝策略的辅助下，提出的误差赋权方法有效提升了基于 Kinect 的视觉里程计的精度。卡耐基梅隆大学的 Sebastian Scherer 等人^[4]设计实现了基于 Kinect 在特征缺失环境下进行无人机位姿估计的系统，分别利用基于距离变化率约束以及光度误差尺度所得到的帧间变换获取相对位姿估计，而利用粒子滤波获取绝对位姿估计。实验证明系统在普通板载 CPU 上可以使得无人机在多个特征缺失环境中获取精确的无人机位姿。东北大学的 Zheng Fang 等人^[5]通过比较不同的基于 RGB-D 的视觉里程计算法来研究对于计算资源有限，快速机动的无人机在选择在线 RGB-D 里程计的策略。通过选择光照情况不同，机动速度不同的不同场景来检验不同 RGB-D 里程计算法在精度，鲁棒性以及计算速度方面的有效性，实验表明在 RGB 图质量较好的条件下应该使用基于 RGB 的方法而不是基于深度图的方法。

在 RGB-D 里程计的应用和优化方面。斯坦福大学研究院的 Kurt Konolige 等人^[6]对在野外环境下的视觉里程计的应用展开研究，改进传统的运动估计算法，使系统适用于大尺度的野外环境并使得在该环境下物体的运动估计保持实时性和准确性。华盛顿大学 Seattle 分校 Peter Henry 等人^[7]提出结合节点优化算法以及视觉特征和形状匹配实现利用 RGB-D 传感器建立室内环境的三维地图，首先通过结合视觉信息和深度信息进行回环检测，之后利用姿态优化算法来实现全局一致的地图。宾夕法尼亚大学的 Giuseppe Loianno 等人^[8]利用深度图修正单目视觉里程计的定位结果，同时得到单目 SLAM 的稀疏图和高分辨率的稠密图，从而提高了系统的可靠性。德州农工大学的 Yan Lu 等人^[9]

针对 RGB-D 里程计中关键点特征容易受到光照影响到问题，利用对光照不敏感的线段，同时考虑深度图边界处由于深度信息不连续导致的缺乏线段特征的问题实现了基于线段特征的里程计，另外，对于这类里程计提出了衡量线段特征不确定性以及估计出的相机位姿的基于 Mahalanobis 距离的评价准则。实验表明这种基于线段特征的系统在精度方面优于基于关键点特征的系统。

1.2.2 融合 IMU 和 RGB-D 的位姿估计

视觉惯导融合应用可以分为视觉惯导融合同时定位与建图、视觉惯导融合里程计两类。根据系统中信息融合的程度又可以将视觉惯导融合系统分为两类，松耦合系统以及紧耦合系统。松耦合系统使用互相独立的视觉与惯导单元，这两种不同的传感器单元按照不同的频率进行运算，然后再将它们的运算结果进行时间同步以及数据的融合。紧耦合系统将原始的视觉数据和惯导数据进行时间同步后直接输入一个滤波器中，直接通过滤波算法对这两部分数据进行融合。

视觉与惯导融合里程计根据视觉信息来源的不同，分为单目惯导融合里程计，双目惯导融合里程计以及 RGB-D 惯导融合里程计三类。现有文献所涉及的视觉与惯导融合里程计系统大部分是松耦合系统。对于这类系统，出现了很多研究成果。

针对 RGB-D 与 IMU 融合里程计。卡尔加里大学的 Jacky C.K. Chow 等人^[10]考虑到现实中低速运动的机器人不断运动和停止的特点，设计实现了一种融合 IMU, LiDAR 和多个 Kinect 的导航系统，通过使用一种点到面的最近点迭代算法以及使用加入迭代步骤的扩展卡尔曼滤波（IEKF, Iterative Extended Kalman Filter）降低了重投影误差。在狭长走道中的实验表明该融合系统的误差低于 0.14%。圣埃斯皮里图联邦大学的 Milton C.P.Santos 等人^[11]利用 RGB-D 传感器信息来补偿 IMU 的数据漂移的问题，对无人飞行器的位置和姿态估计进行研究，通过故意引入会导致 IMU 数据漂移的飞行器迅速运动和转弯并将实际数据与预先标定的路点（waypoint）相比较验证了融合系统在位姿解算过程中的准确性。另外，他们^[12]比较了 DIF 可 DKF 两种不同的数据融合算法，并为不同实验场景下融合算法的选择提出依据。

苏黎世联邦理工学院的 Stefan Leutenegger 等人^[13]提出利用优化的方法来融合 RGB-D 和 IMU 信息，通过将 IMU 的误差项和和地标的重投影误差利用概率方式融合到一起，形成一个非线性代价函数，通过优化这个代价函数从而得出最优的融合信息。博洛尼亚大学的 Nicholas Brunetto 等人^[14]针对视觉 SLAM 中因为需要匹配特征而带来的

不准确性的问题，提出利用视觉融合 IMU 的方法进行姿态估计，并特别注意到滤波模型中状态的选择和 IMU 数据的预处理，拓展了视觉 SLAM 可以适用的场景范围。澳大利亚国立大学的 Usman Qayyum 等人^[15]通过分析 Kinect 在室外环境下所具有的信息遗漏和丢失问题（Drop-out Problem）设计实现了融合 IMU 和 Kinect 信息的融合导航系统，通过使用 IMU 在室内使用深度传感器融合 6 自由度位姿，而在室外则只使用 Kinect 的 RGB 相机，融合 5 自由度位姿，在室外通过将问题转化为单目 SLAM 问题将 Kinect 用于导航的环境要求扩展到室外。

针对单目或双目与 IMU 融合里程计。卡尔加里大学的 N. Mostofi 等人^[16]搭建基于智能手机和 IMU 的 SLAM 系统，并特别考虑了在利用 EKF 进行融合滤波的过程中存在的地标初始化延迟问题，利用磁传感器测量模型消除了航向误差引入 SLAM 算法中的不连续性。苏黎世联邦理工学院的 Sammy Omari 等人^[17]考虑到基于特征的系统的缺陷，提出利用来自于融合视觉系统的全部测量信息基于概率模型来建立里程计和建图系统，此系统可以鲁棒地估计自身的姿态并实时建立地图。明尼苏达大学的 Anastasios I. Mourikis 等人^[18]基于 EKF 滤波算法推导出考虑了由于照相机的不同姿态导致的几何约束的观测模型，该模型不需要状态向量中的三维特征位置且具有多项式时间的计算复杂度，因而适用于城市环境中融合系统导航系统。谷歌的 Joel A Hesch 等人^[19]研究了视觉惯导融合系统的可观性和观测不连续性，通过将非线性融合模型中的可观部分和不可观部分进行模型分解从而到达提高模型精度的目的。

1.2.3 IMM 融合滤波算法

相当大的一部分视觉惯导融合系统将扩展卡尔曼滤波（Extended Kalman Filter）算法作为融合工具。荷兰国家航天实验室的 H.A.P. Blom^[20]针对具有 Markov 系数的离散时间系统的滤波融合问题展开研究，提出了在保持模型输出准确性但有效降低模型计算复杂度的 IMM 算法。该算法可以被应用于目标跟踪，组合导航等问题中。考虑到融合系统中包括 RGB-D 和 IMU 两类传感器，可以建立多个运动学模型，因而可以考虑使用 IMM 算法对这两类数据进行融合。

在运动物体的建模方面。加州大学伯克利分校的 Jason Kong 等人^[21]通过分析两类静力学和动力学两种自行车模型（Bicycle Model）所具有的预测误差，说明了动力学模型实用性差以及计算复杂度高的缺点，通过实验指出利用静力学模型设计的控制器获取的控制效果更好，在实际问题中应该采用静力学自行车模型的结论。麦克马斯特大学的

T.Kirubarajan 等人^[22]通过引入同时考虑了目标运动的不确定性以及测量不确定性的目标机动指数（Target Maneuvering Index），通过分析交互多模型算法在跟踪单一问题和多个物体过程中的相对于卡尔曼滤波器的优秀表现，指出在一定条件下由于机动指数较大使得交互多模型滤波器在进行运动物体跟踪的过程中必不可少。

在 IMM 算法的应用方面。近年来针对利用 IMM 算法进行传感器数据融合有一些研究成果。东北大学付文艳等人^[23]提出利用 IMM 来实现室内移动传感器网络定位问题中视距（LOS）和非视距（NLOS）的自动切换，通过这两个模型的交互融合来提高移动目标的定位精度。香港科技大学的 Ying-Kin Yu 等人^[24]提出利用 IMM 提高增强现实（AR）中放置虚拟物体的精度，融合 3 个基于 EKF 进行滤波的运动学子模型（一般模型，平移模型和旋转模型）来描述待放置虚拟物体的运动，从而更加精确且高效的增强现实效果。中田纳西州立大学的 Lisa Bloomer 等人^[25]针对利用 IMM 算法进行建模的系统所采用的子模型数目进行研究，通过选择 3 个子模型对系统建模并特别注意到 Markov 状态转移矩阵中系数的位置对系统进行仿真分析，得出在大部分场景下基于 IMM 进行建模的系统只需要两个子模型。保加利亚科学院的 Iliyana Simeonova 等人^[26]综合前人研究，对在不同场景下的利用 IMM 对系统进行建模过程中的参数选择问题进行讨论，给出基于 IMM 系统模型设计的一般规则。

在 IMM 算法的优化方面。Mohamed Boudiaf 科学技术大学的 Dahmani Mohammed 等人^[27]提出利用 $\alpha - \beta$ 和 $\alpha - \beta - \gamma$ 模型代替 EKF 作为 IMM 的子模型，降低了系统演化过程中的计算复杂度，使得基于 IMM 进行建模的系统可以保证实时性。延世大学的 J.B Park 等人^[28]针对系统模型中的不确定参数，提出了基于模糊集理论的 IMM 算法，并利用遗传算法对模糊系统进行优化，克服了对移动物体进行位姿估计过程中需要先验知识的缺点。

虽然 RGB-D 与 IMU 组合导航已经有较多的研究，但是在文献中还没有看到将 IMM 算法应用于该领域。考虑 IMM 算法在处理滤波融合问题时所展现出的优点以及融合 RGB-D 和 IMU 传感器进行移动机器人位姿估计的实际需求，将 IMM 应用于视觉与 IMU 相融合的移动机器人位姿估计具有重要的现实意义。

1.3 主要研究内容和章节安排

本课题主要研究融合 IMU 和 Kinect 的移动机器人位姿估计问题。围绕融合在室内不同场景下，机器人不同的运动模式下解决 IMU 传感器的测量漂移以及 Kinect 传感器

对于环境特征的依赖问题，提高移动机器人位姿解算系统对于环境的适应性，在移动机器人快速机动，环境光照强烈和特征缺乏共同影响传感器性能的条件下保证位姿解算系统的精度。

本文各章节内容的安排如下：

第一章为绪论。对进行本课题研究的背景和意义以及国内外在该领域的研究情况进行了阐述。通过说明室内相对位姿的必要性，叙述使用 IMU 和 Kinect 传感器进行位姿解算的研究现状，以及通过卡尔曼滤波以及 IMM 算法进行传感器信息融合的可能性。

第二章是基于 IMU 与 Kinect 的移动机器人位姿估计。阐述具有 IMU 或者 Kinect 单传感器进行位姿解算的原理，通过对原理的分析指出单传感器位姿解算系统的不可靠性，通过实验证明分析过程中的推断。

第三章是基于卡尔曼滤波的 IMU 和 Kinect 融合位姿估计。给出卡尔曼滤波的原理和算法流程，移动机器人的两种运动学模型，分别建立移动机器人的线性和非线性模型，并基于这两个模型对线性的卡尔曼滤波器以及非线性的扩展卡尔曼滤波器进行设计，然后通过实验评价滤波器的表现以及融合的位姿估计与单传感器系统位姿估计的区别。

第四章是基于 IMM-KF 的融合 IMU 与 Kinect 的位姿估计。首先给出交互多模型滤波算法，通过系统的机动指数（Maneuvering Index）证明利用该算法对滤波结果进行改善的必要性，其次对 IMM-KF 滤波器进行设计，并针对滤波器中子滤波器的数量、类型和参数进行讨论，最后通过实验证明 IMM-KF 滤波器进行融合位姿估计的有效性。

第五章为总结和展望。通过总结本文的研究工作，指出本课题对于融合 IMU 和 Kinect 的研究贡献和主要不足，同时基于本课题中存在的问题对未来的研究进行展望。

第二章 基于 IMU 和 Kinect 的移动机器人位姿估计

为了实时获取移动机器人在室内运动过程中的位置和姿态，基于 IMU 与 Kinect 所进行的相对位姿估计分别通过测量角速度，线加速度等运动学参数或者获取环境的彩色和深度图的方式，利用直接积分或者视觉里程计的方法来间接得到移动机器人的位姿信息。本章将阐述基于 IMU 和 Kinect 的单传感器位姿解算系统的原理和特点，并对这两类位姿解算系统进行实验分析。

2.1 IMU 位姿解算

2.1.1 惯性导航系统

惯性导航系统是通过部署在运动物体（如飞行器，机器人和潜水艇等）的惯性测量元件 IMU 输出的线加速度和角速度，从而推算运动物体位置，航向等信息的系统。惯性导航系统的核心是测量加速度，角速度等关键参数的惯性测量元件 IMU。惯性测量元件 IMU 是随着惯性传感器发展起来的。由于其自主性，隐蔽性以及实时性的特点，IMU 不仅在军事领域的进步起到了关键的推动作用，对很多民用领域的发展也施加着日益重要的影响。

IMU 由线加速度计和陀螺仪这两类惯性测量元件组成。线加速度计在载体自身坐标系下测量其的线加速度，在获取线加速度的条件下即可通过对加速度两次积分得到载体的位置。陀螺仪则测量载体在旋转过程中的角速度和四元数，用于表征载体姿态的变化。

根据 IMU 安装条件的不同，可以将惯性导航系统分为捷联式惯性导航系统（Strapdown Inertial Navigation System）和平台式惯性导航系统（Gimbaled Inertial Navigation System）两类。平台式惯性导航系统中的 IMU 被部署在惯性平台上，由于惯性平台可以屏蔽外界的旋转，再利用万向节和扭矩电机等机电装置就能够保证 IMU 所在的惯导坐标系与世界坐标系一致。利用平台式惯导进行位姿解算不需要进行惯导坐标系到世界坐标系的坐标变换。捷联式惯导直接固定在载体上，没有惯性平台，所以易于受到载体抖动的影响，同时在进行位姿解算的过程中需要进行惯导坐标系与世界坐标系之间的坐标变换。

2.1.2 姿态表征

2.1.2.1 位置表征

只有在特定的坐标系下才能够对于移动机器人的位置进行定量的描述，根据机器人的运动范围和运动环境的不同，坐标轴的朝向也有不同，但是都保持右手直角坐标系惯例。

（1）世界坐标系

① 地球坐标系。当多地机器人的联合控制，具有多个局部坐标系时，使用地球坐标系作为世界坐标系。以地心为原点建立右手直角坐标系，即地球坐标系。在地球坐标系下，不同地方的机器人在地球坐标系下具有统一的坐标表示。

② 导航坐标系。当不要求多地的机器人具有统一的坐标表示时，采用导航坐标系。北-东-天坐标系是常用的一种导航坐标系，其中 x 轴指向东方， y 轴指向北方， z 轴指向上方。当进行小区域导航时，由于机器人移动的范围远小于地球的半径，可以采用另外一种导航坐标系，即将机器人初始状态下的前进方向定为 x 方向，并以此为基准建立右手直角坐标系，这就是前-左-上（Forward-Left-Up）坐标系。

（2）惯导坐标系

惯导坐标系是 IMU 测量值所对应的坐标系。惯导坐标系利用线加速度对应的三个轴建立右手直角坐标系，即 x 轴平行于载体水平面指向机器人运动的正前方， y 轴垂直于 x 轴并指向机器人运动方向的左侧，而 z 轴垂直于 x 和 y 轴组成的平面指向正上方。对于捷联式惯导，由于 IMU 与移动机器人固定在一起，惯导坐标系与机器人的载体坐标系重合，惯导坐标系测量的加速度等参数就是移动机器人的运动参数。

2.1.2.2 姿态表征

表征姿态的方法有欧拉角，四元数，旋转矩阵，旋转向量，轴角和方向余弦矩阵等。这些表示方法虽然各有自身的优缺点，但是可以互相转换。

（1）欧拉角

欧拉角是描述刚体相对于一个给定坐标系旋转的方法。坐标系固定时的旋转称为外部（extrinsic）旋转，而当坐标系随着刚体运动而运动时称此时的旋转为内部（intrinsic）旋转。在本课题中使用的欧拉角描述的旋转是外部旋转。对应于单一刚体的前-左-上（Forward-Left-Up）坐标系，如图 2.1 所示，将移动机器人绕 x, y 和 z 旋转角称为侧倾角，俯仰角和偏航角，分别用符号 ψ, θ, φ 表示。

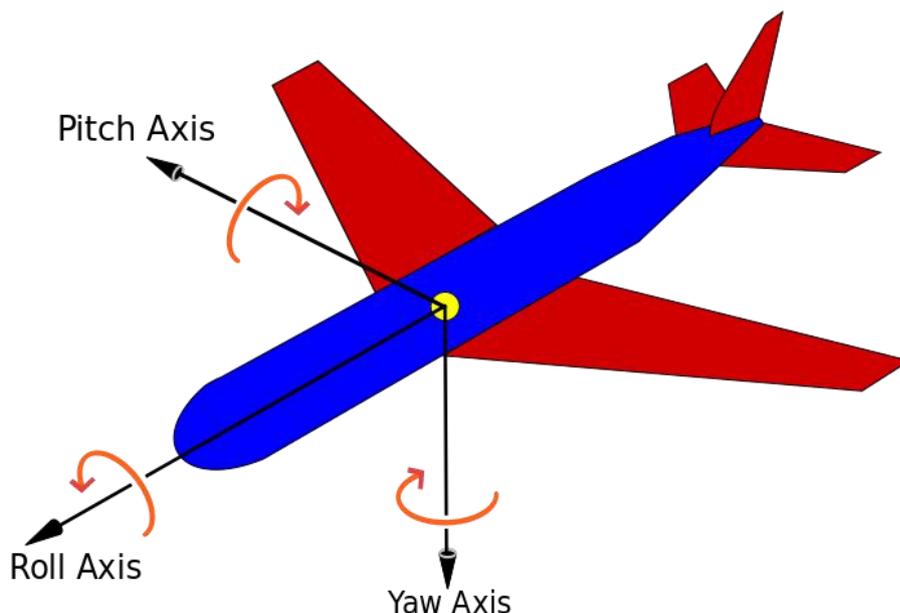


图 2.1 三维空间中旋转的欧拉角表示

(2) 四元数

四元数是复数的拓展，定义为：

$$\mathbf{q} = e_0 + e_1\mathbf{i} + e_2\mathbf{j} + e_3\mathbf{k} \quad (2.1)$$

由于四元数克服了欧拉角带来的旋转顺序不同导致的表示不唯一以及插值困难的问题，常用来表示三维空间中刚体的旋转。欧拉角和四元数之间的可以进行如下转换：

$$\begin{aligned} e_0 &= \cos \frac{\psi}{2} \cos \frac{\theta}{2} \cos \frac{\varphi}{2} + \sin \frac{\psi}{2} \sin \frac{\theta}{2} \sin \frac{\varphi}{2} \\ e_1 &= \sin \frac{\psi}{2} \cos \frac{\theta}{2} \cos \frac{\varphi}{2} - \cos \frac{\psi}{2} \sin \frac{\theta}{2} \sin \frac{\varphi}{2} \\ e_2 &= \cos \frac{\psi}{2} \sin \frac{\theta}{2} \cos \frac{\varphi}{2} + \sin \frac{\psi}{2} \cos \frac{\theta}{2} \sin \frac{\varphi}{2} \\ e_3 &= \cos \frac{\psi}{2} \cos \frac{\theta}{2} \sin \frac{\varphi}{2} - \sin \frac{\psi}{2} \sin \frac{\theta}{2} \cos \frac{\varphi}{2} \end{aligned} \quad (2.2)$$

$$\begin{aligned} \varphi &= \arctan\left(\frac{2(e_0e_1 + e_2e_3)}{e_0^2 - e_1^2 - e_2^2 + e_3^2}\right) \\ \theta &= \arctan(2(e_0e_2 - e_1e_3)) \\ \psi &= \arctan\left(\frac{2(e_0e_3 + e_1e_2)}{e_0^2 + e_1^2 - e_2^2 - e_3^2}\right) \end{aligned} \quad (2.3)$$

(3) 旋转矩阵

直观上欧拉角可以表示三维空间中的旋转，但是仅仅使用欧拉角不足以完成在导航

坐标系和惯导坐标系之间进行坐标变换。所以在欧拉角的基础上构造旋转矩阵，从而实现坐标变换。

当载体仅有偏航，侧倾和俯仰一个方向的姿态变化时，载体对应的旋转矩阵分别是：

$$R(\psi) = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}, R(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}, R(\varphi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & \sin\varphi \\ 0 & \sin\varphi & \cos\varphi \end{bmatrix} \quad (2.4)$$

当载体同时有俯仰，侧倾和偏航三个方向的姿态变化时，若载体按照 ψ, θ, φ 的顺序旋转，旋转矩阵为：

$$R = R(\psi)R(\theta)R(\varphi) = \begin{bmatrix} \cos\psi \cos\varphi - \cos\theta \sin\psi \sin\varphi & -\cos\theta \cos\varphi \sin\psi - \cos\psi \sin\varphi & \sin\psi \sin\theta \\ \cos\psi \sin\varphi + \cos\theta \sin\psi \cos\varphi & \cos\psi \cos\theta \cos\varphi - \sin\psi \sin\varphi & -\cos\psi \sin\theta \\ \sin\theta \sin\varphi & \cos\varphi \sin\theta & \cos\theta \end{bmatrix} \quad (2.5)$$

(4) 旋转向量

旋转矩阵是一个 3×3 的矩阵，在处理过程中需要耗费较多的计算资源，同时利用 9 个量表示 3 个自由度存在冗余信息。可以将旋转矩阵转换为一个三维向量，通过处理向量在节约计算资源的同时减少存储空间。

旋转向量 r 与旋转矩阵 R 之间的转换通过 Rodrigues 变换实现。

$$R = \cos\theta \mathbf{I} + (1 - \cos\theta)rr^T + \sin\theta \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix} \quad (2.6)$$

其中 $\theta = \|r\|$, $(r_x, r_y, r_z) = r/\theta$ 。

(5) 轴角表示法

旋转向量解决了计算过程中的问题，但是牺牲了表示旋转矩阵和向量的欧拉角所具有的直观性。可以将旋转向量转换为轴角来解决这一问题，单位化的旋转向量为轴角表示法中轴的指向，旋转向量的模为轴角表示法中角的大小，即：

$$\begin{aligned} x &= \|r\| \\ \theta &= \sqrt{r_1^2 + r_2^2 + r_3^2} \end{aligned} \quad (2.7)$$

2.1.3 位姿解算

随着移动机器人的运动，IMU 测量系统的角速度，姿态以及线加速度，并输出测量值。确定移动机器人位姿需要将测量值转换为对应的状态，即位置和航向。位姿解算就

是将 IMU 的测量值转换为移动机器人位置和航向的过程。

在移动机器人出发前，其质心在地面上的投影为坐标原点，以该点为基准，按照前-左-上（Forward-Left-Up）原则建立右手直角坐标系。由于本课题中移动机器人的运动范围在室内，利用坐标 (x, y) 即可描述机器人在室内相对于坐标原点的位置。另外，为了简便和直观起见，取欧拉角为姿态表征的方法。在室内场景下，移动机器人不可能发生俯仰和侧倾运动，只能发生偏航运动，因而仅使用偏航角 ψ 即可描述移动机器人在场景中的姿态。故移动机器人在室内场景下的位姿表示为 $x = [x, y, \psi]^T$ 。

以下分别考虑移动机器人的位置和姿态。

针对姿态，由于装配在移动机器人上的 IMU 可以提供角速度和四元数信息，可以直接从 IMU 中获取四元数，从四元数转换为偏航角：

$$\psi = \arctan\left(\frac{2(e_0e_3 + e_1e_2)}{e_0^2 + e_1^2 - e_2^2 - e_3^2}\right) \quad (2.8)$$

针对位置，由于 IMU 内部的加速度计只能提供移动机器人运动过程中在惯导坐标系中的加速度，不能直接获取速度和位置。所以基于已经获得的姿态，首先将 x 方向的线加速度分解到世界坐标系下：

$$\begin{aligned} a_x^W &= a_x^B \cdot \cos\psi \\ a_y^W &= a_x^B \cdot \sin\psi \end{aligned} \quad (2.9)$$

对加速度进行积分得到世界坐标系下的速度：

$$\begin{aligned} v_x^W &= v_x^W(0) + \int_t a_x^W dt \\ v_y^W &= v_y^W(0) + \int_t a_y^W dt \end{aligned} \quad (2.10)$$

对速度进行积分得到位置：

$$\begin{aligned} x^W &= x^W(0) + \int_t v_x^W dt \\ y^W &= y^W(0) + \int_t v_y^W dt \end{aligned} \quad (2.11)$$

在实际中由于被积函数表达式未知，利用求和代替积分，得到位置：

$$\begin{aligned} x &= \sum_i \Delta t_i \sum_j a_{xj}^W \Delta t_i \\ y &= \sum_i \Delta t_i \sum_j a_{yj}^W \Delta t_i \end{aligned} \quad (2.12)$$

根据公式可以利用 IMU 得到移动机器人的位置信息，但是由于运动过程中的抖动，传感器的测量误差以及针对加速度进行两次积分过程中的累积误差，随着时间的推移，

位置的误差将不断增大。

2.2 Kinect 位姿解算

2.2.1 相机模型

常规的 RGB 相机常使用针孔模型进行描述。如图 2.2 所示，在针孔模型中，成像的过程被简化为小孔成像的过程。在这一过程中，物点和像点之间的关系是透射变换，位于世界坐标系中的物点的坐标被变换到像素坐标系的像点上，即：

$$s \cdot m' = A \cdot [R \ t] \cdot M' \tag{2.13}$$

其中 A 是相机内参，描述相机在制造过程中就已经确定的焦距和光圈中心等参数， $[R, t]$ 是相机外参，描述相机的位姿。 s 是缩放因子，取 1000。

$$s \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{2.14}$$

Kinect 由普通的 RGB 相机，随机红外点云投射器和和红外 CMOS 相机组成，除了可以获得普通的彩色图像，还可以获得深度图像。利用结构光原理，Kinect 通过点云投射器将光源连续照射到物体上，当被照射场景的内不同物体距 Kinect 的距离不同时得到的图案也不同，之后通过测量图案即可得到 Kinect 的位置。

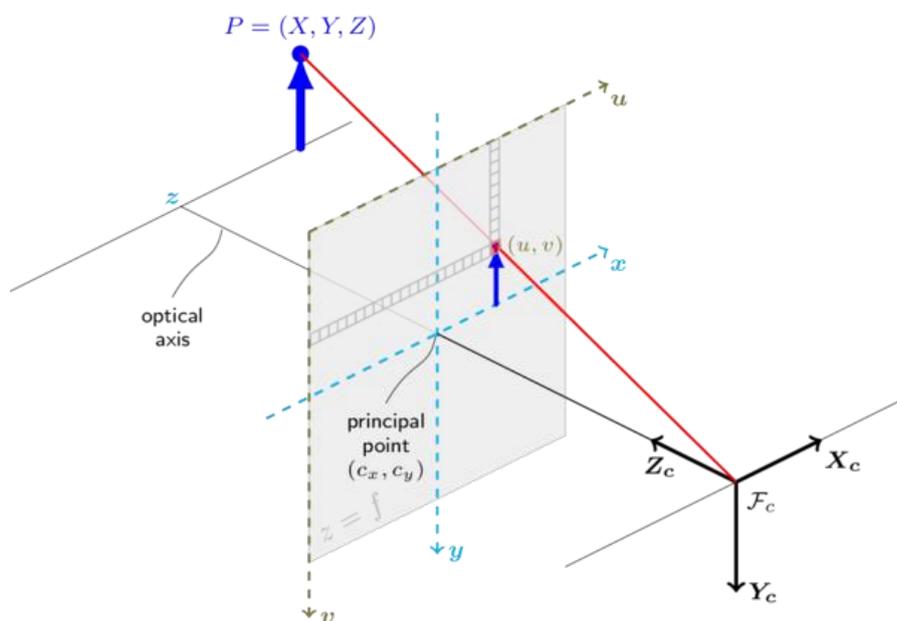


图 2.2 针孔相机模型

2.2.2 点云生成

现实场景可以利用若干个点的表示，其中每个点由位置和颜色两个向量组成，则对于场景 $X = \{x_1, x_2, \dots, x_n\}$ 。

$$x_i = [x, y, z, r, g, b]^T \quad (i = 1, 2, \dots, n) \quad (2.16)$$

组成 Kinect 相机的 RGB 相机和深度相机分别可以获得像点的二维坐标 (u, v) 和深度 d ，即 (u, v, d) 。利用相机的内参和外参矩阵可以求出该像点对应的三维点。生成单幅点云时，Kinect 相对于场景静止，则取旋转矩阵 $R = \mathbf{I}$ ，平移向量 $t = \mathbf{0}$ 。此时，物点到像点的透射变换为：

$$s \cdot \begin{bmatrix} u \\ v \\ d \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (2.17)$$

则可得到三维物点：

$$\begin{aligned} z &= \frac{d}{s} \\ x &= \frac{u - c_x}{f_x} \cdot z \\ y &= \frac{v - c_y}{f_y} \cdot z \end{aligned} \quad (2.18)$$

利用 Kinect 获取的图像对场景中的每一个点做以上运算，再加入 RGB 相机中的颜色信息，即可得到场景的点云 X 。

2.2.3 特征提取和配准

由于在获取点云的过程中进行了机器人与环境处于相对静止状态的假设，当机器人在整个场景中运动时，需要获取整个场景中的点云。这就需要估计不同帧图片之间的相对位置关系，而获取相对位置关系的前提就是从相邻帧的图像中提取可以认为相同的点，这就是特征提取的过程。

通常使用 SIFT/SURF 特征描述子和 ORB 特征描述子作为特征提取的依据。在本课题中使用 ORB 特征描述子用来提取移动机器人环境中的特征。

ORB 特征是一种基于 SIFT/SURF 并对其进行优化的二值特征描述子，与 Harris 角点等相比具有良好的旋转不变性以及缩放不变性，图 2.3 所示为经过特征提取后的图片。

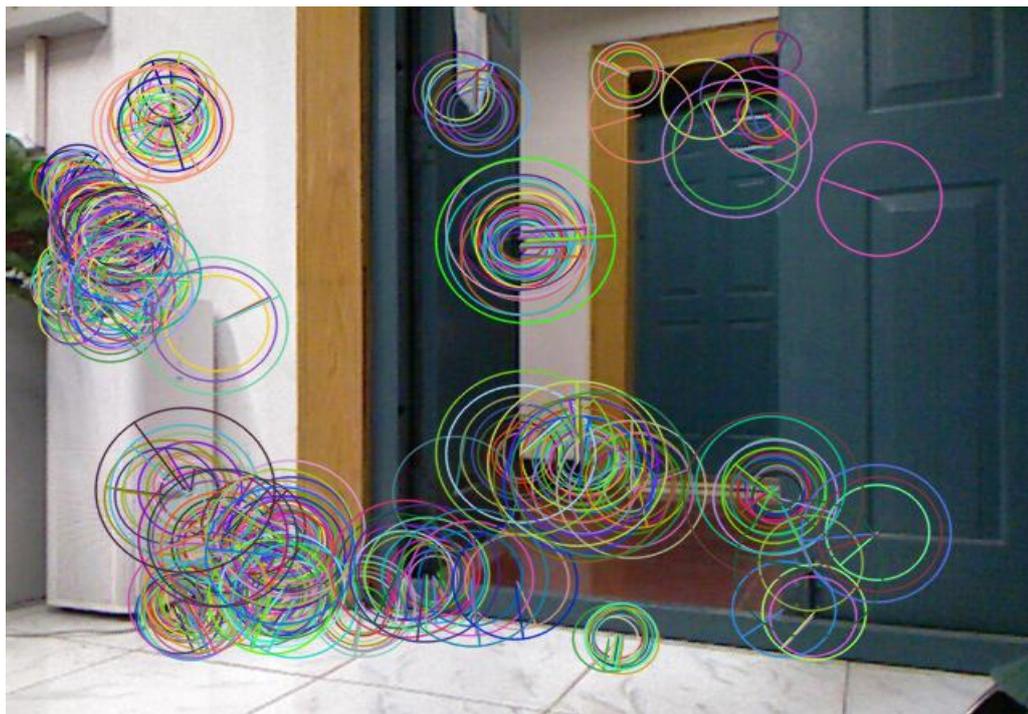


图 2.3 单帧图片 ORB 特征提取结果

在获得相邻两帧图像 F_1 和 F_2 的条件下，获得了一组对应的特征点：

$$\begin{aligned} P &= \{p_1, p_2, \dots, p_N\} \in F_1 \\ Q &= \{q_1, q_2, \dots, q_N\} \in F_2 \end{aligned} \quad (2.19)$$

在理想情况下，对于任意的 $i \in \{1, 2, \dots, N\}$ ，两两对应的特征点都 p_i 和 q_i 满足相同的平移和旋转关系，通过旋转矩阵 R 和旋转向量 t 可以进行坐标变换，即：

$$p_i = Rq_i + t \quad (2.20)$$

但是由于相机的误差和环境中的噪声，以上关系通常不能满足。所以考虑最小误差意义下的平移矩阵和旋转向量，即：

$$\min_{R, t} \sum_{i=1}^N \|p_i - (Rq_i + t)\| \quad (2.21)$$

实际中图像配准得到的通常是旋转向量 r ，通过 Rodrigues 变换可以变换为旋转矩阵 R 。这个优化问题可以用迭代最近点算法（ICP, Iterative Closet Point）求解。如图 2.4 所示，相邻帧的图片利用 ORB 描述子进行特征匹配，并计算出平移和旋转向量。

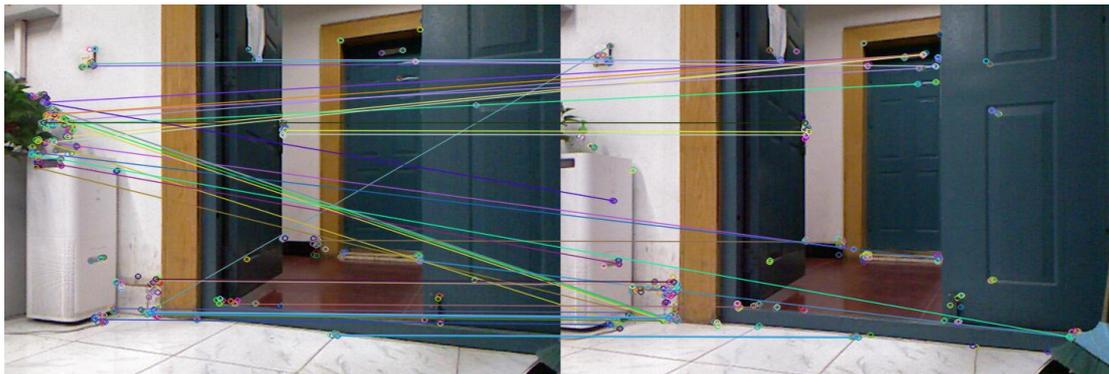


图 2.4 两帧图像特征匹配结果

2.2.4 点云拼接

当获取旋转矩阵 R 和旋转向量 t 后，即可进行点云拼接。点云拼接实际上是对点云中的每一个点进行现行变换的过程。已知变换矩阵 T ：

$$T = \begin{bmatrix} R_{3 \times 3} & t_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \in R^{4 \times 4} \quad (2.22)$$

则将由前一帧图像生成的点云变换到后一帧的点云所在坐标系中，可得到经过拼接的两帧图像的点云：

$$\begin{bmatrix} x^{F_2} \\ y^{F_2} \\ z^{F_2} \\ 1 \end{bmatrix} = T \cdot \begin{bmatrix} x^{F_1} \\ y^{F_1} \\ z^{F_1} \\ 1 \end{bmatrix} \quad (2.23)$$

如图 2.5 所示是相邻两帧进行点云拼接的结果，随着机器人运动范围的不拓展，可以获取整个场景的点云。



图 2.5 两帧图像的点云拼接结果

2.2.5 视觉里程计

在机器人的导航问题中，里程计是一种利用从传感器获取的数据来获得机器人位置变化的方法，可以通过旋转编码器和 IMU 等传感器获取机器人的里程计，但是对于地面状况不理想（比如不平整或者打滑）或者机器人在运动过程中抖动，这类里程计随着时间的推移会有较大的累积误差。视觉里程计使用相机作为传感器，是一种通过分析图像序列获取机器人的位姿的方法，由于图像仅依赖于环境中的特征点，基本不受地面状况的影响，所以是弥补 IMU 或者旋转编码器缺陷的有效方法。具体的，当获取拼接后的点云后，利用不断迭代的点云生成和配准过程，即可得到机器人相对于规定原点的位置，即实现视觉里程计。

2.3 实验及结果分析

2.3.1 IMU 位姿解算实验

利用 TurtleBot 移动机器人，在仅利用 IMU 作为位姿解算数据来源的条件下使机器人走矩形闭环，利用机器人所在惯导坐标系下/base_link 到机器人位置的世界坐标系 /odom 之间的坐标变换作为参考实际值，从而评价仅将 IMU 数据用于位姿解算时解算的精度。

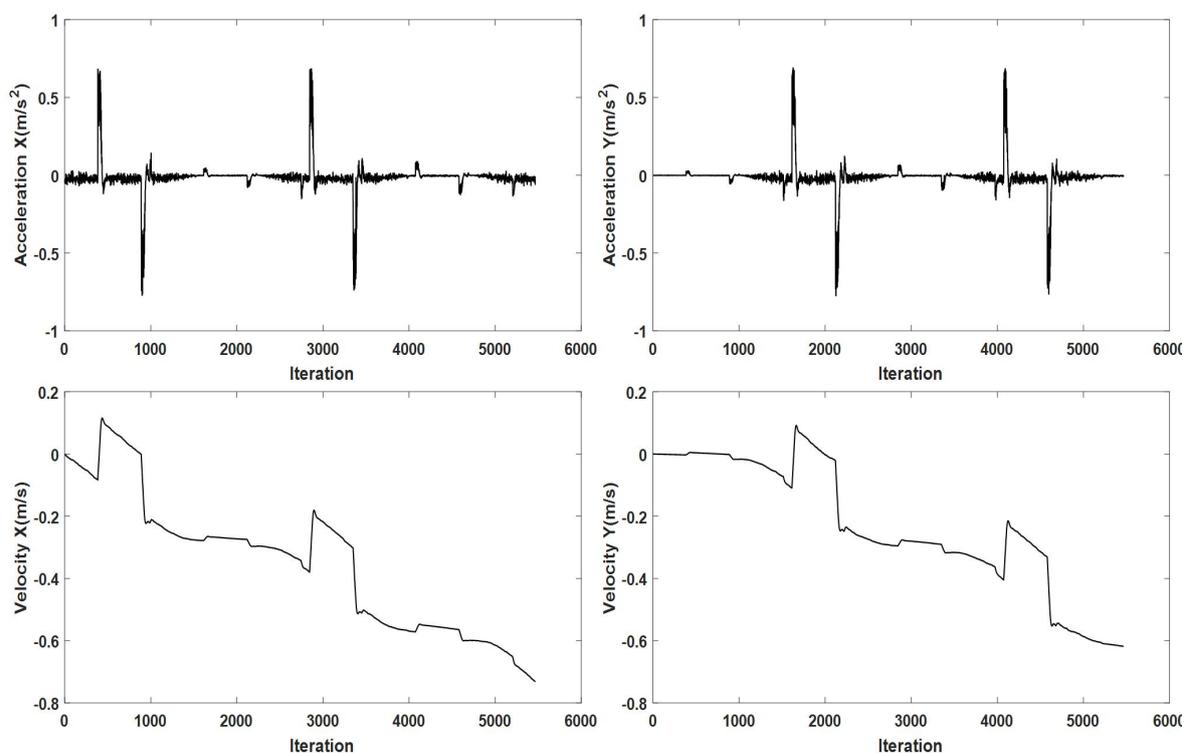


图 2.6 IMU 获取的加速度以及一次积分得到的线速度

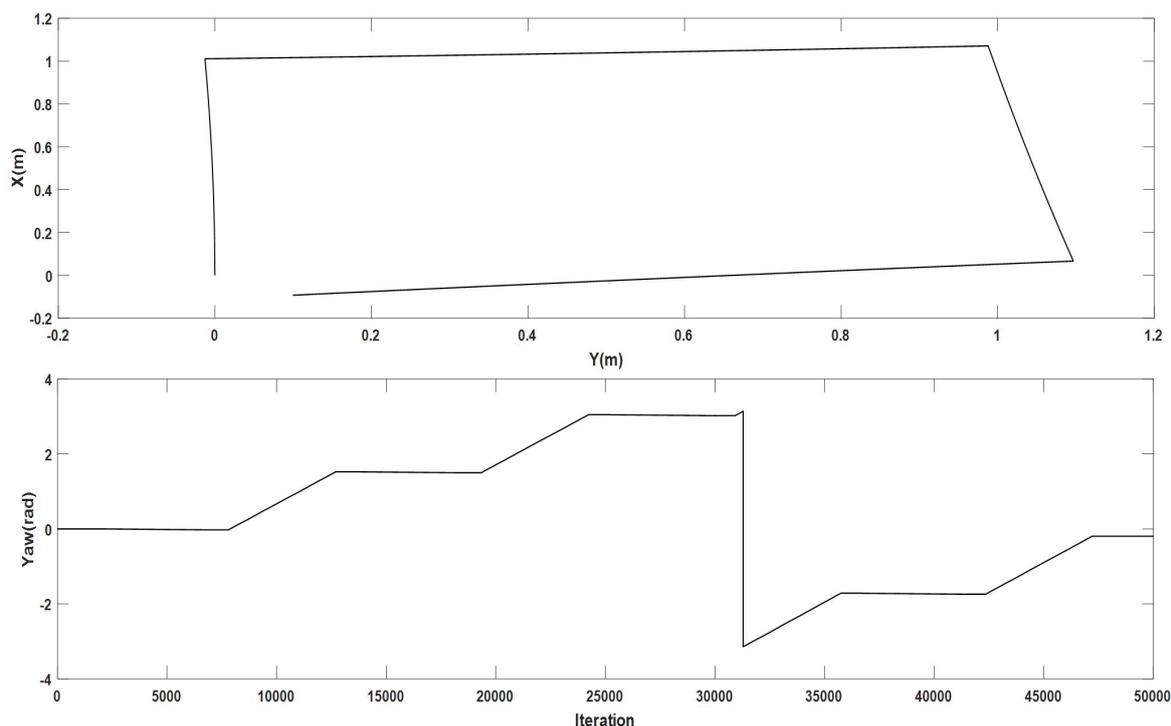


图 2.7 实际的线速度

如图 2.6 所示，由于 IMU 自身的噪声，从 IMU 中获取 x 和 y 方向的加速度在多数时候维持于一个非 0 值，且在多个节点处出现严重的测量错误，如 Iteration 约为 3400 时，如图 2.7 所示，速度实际维持不变，但是 IMU 测量得到的加速度出现一个负的尖峰。因为 IMU 存在的噪声，在多个节点处出现的错误以及在积分过程中这些误差的不断累积，导致利用 IMU 数据解算出的速度值向负方向不断漂移，达到了远超过机器人速度上限的 -0.6m/s 。

已经累积较大误差的速度经过再次积分得到的位置因为误差的再度累积出现无限的漂移，在机器人行走的矩形闭环边长不超过 1m 的情况下，在 x 和 y 方向的位移分别达到了 -19m 和 -13m 。

但是与位置不同，偏航角的求解利用四元数到欧拉角的对应关系可以直接得到，没有误差累积的积分过程，利用 IMU 测量得到的偏航角与实际情况基本吻合，可以作为位姿解算的数据来源使用。

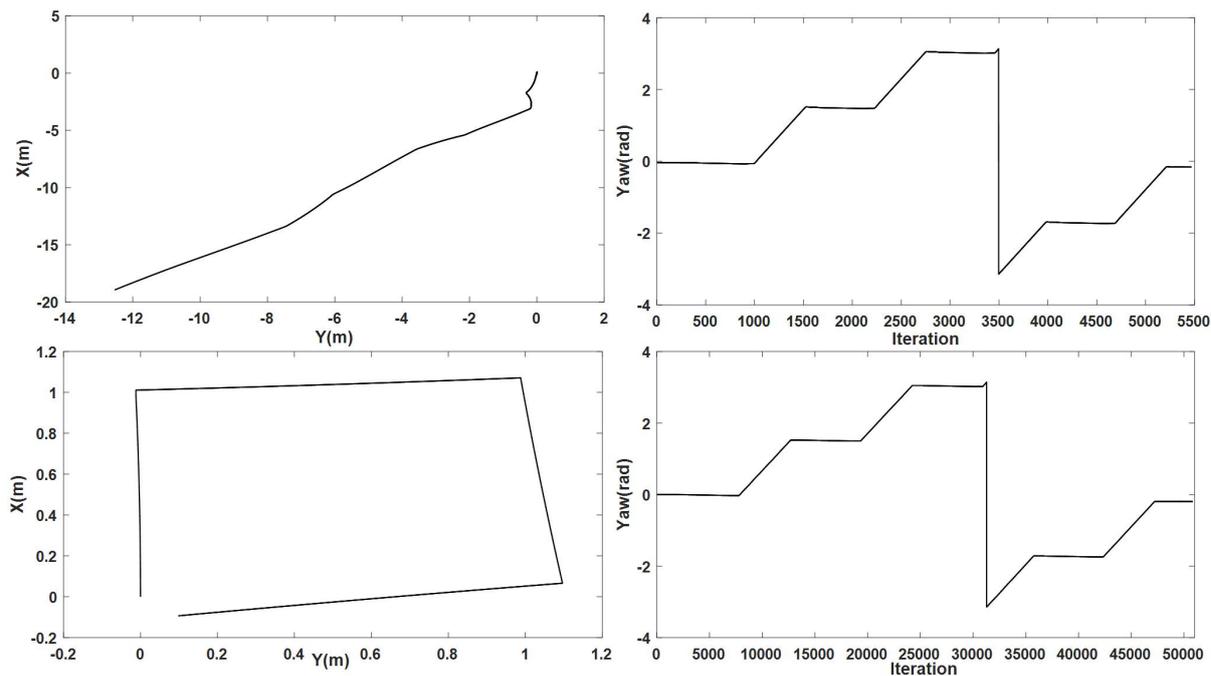


图 2.8 IMU 位姿解算结果和实际的机器人位姿

2.3.2 Kinect 位姿解算实验

与仅利用 IMU 数据进行位姿解算的实验条件类似，使用 TurtleBot 移动机器人搭载 Kinect V1 传感器进行实验。同时以惯导坐标系/base_link 到世界坐标系/odom 的变换作为参考值对 Kinect 位姿解算方法的表现进行评估。实验环境中部分部分场景缺乏光照，部分场景特征不足。

如图 2-8 所示，利用 Kinect 得到的位置和姿态出现很大误差。在光照不足和特征点缺乏的条件下，视觉里程计不断归零，导致通过 Kinect 获取的机器人位置不断呈现错误的(0, 0)坐标，而表征姿态的偏航角 ψ 则不断归零振荡。

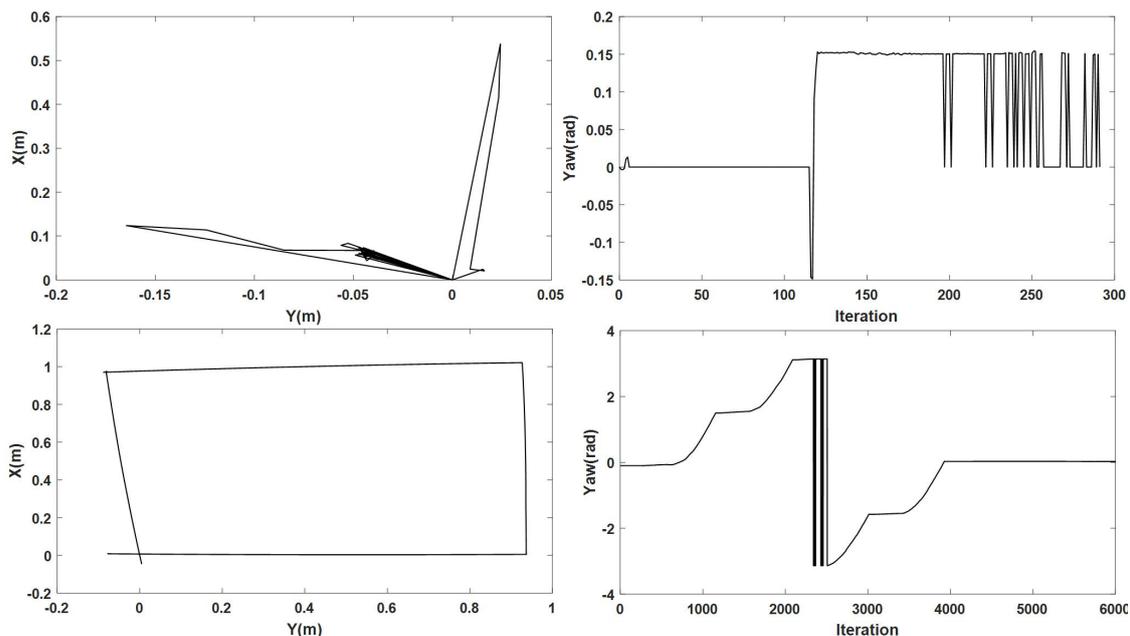


图 2.9 光照，特征缺乏条件下的 Kinect 位姿解算结果以及实际机器人位姿

分别考察在图 2.9 中的移动机器人运动轨迹中的 x 和 y 方向上的位移, 如图 2.10 所示, 在 x 方向, 利用 Kinect 获取的位置不能正确反映机器人向正前方移动并保持 $x = 1\text{m}$ 的事实, 在后半段则出现了实际中不存在的正向约 $x = 0.1\text{m}$ 位移。在 y 方向, 存在类似的问题, 即 y 坐标不能体现机器人在第二条边上的运动, 还在后半段出现了不存在的负向约 $y = -0.05\text{m}$ 的位移。

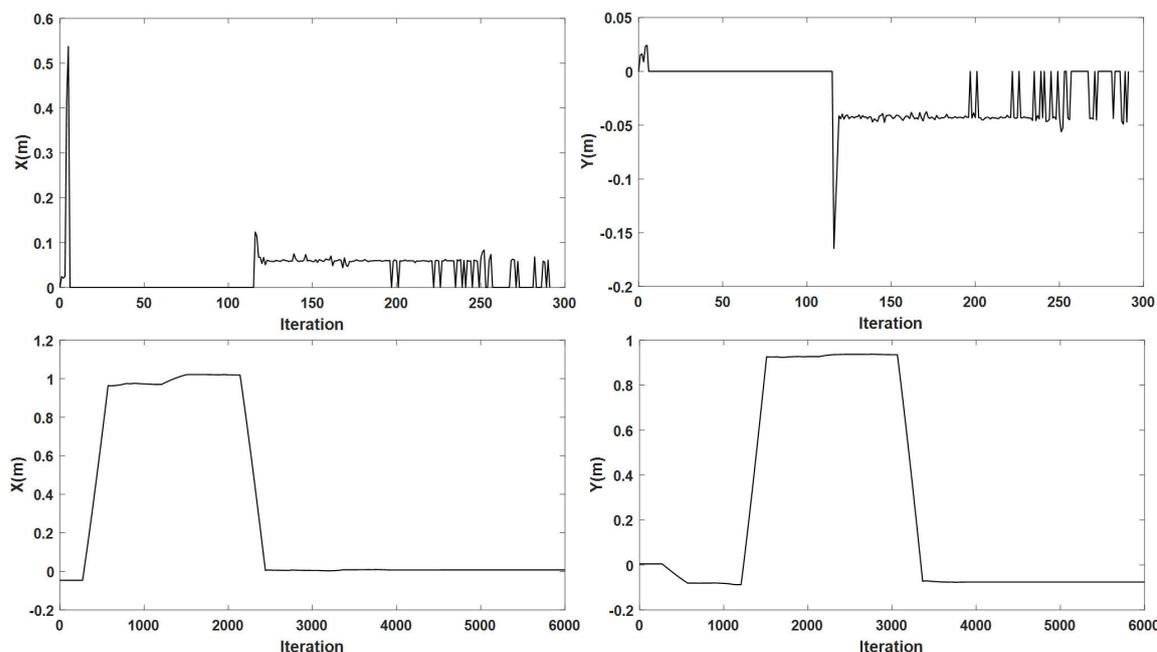


图 2.10 光照，特征缺乏条件下的利用 Kinect 获取位置以及实际的机器人位置

如图 2.11 所示，在特征充足，光照充分的条件下，可以获得正确的利用 Kinect 数据进行位姿解算的结果。单独考察 x 和 y 方向上的位移变化，如图 2.12 所示，可以发现 Kinect 获取的位置和实际位置基本保持一致，可以正确地反映机器人在走矩形闭环过程中的运动规律。

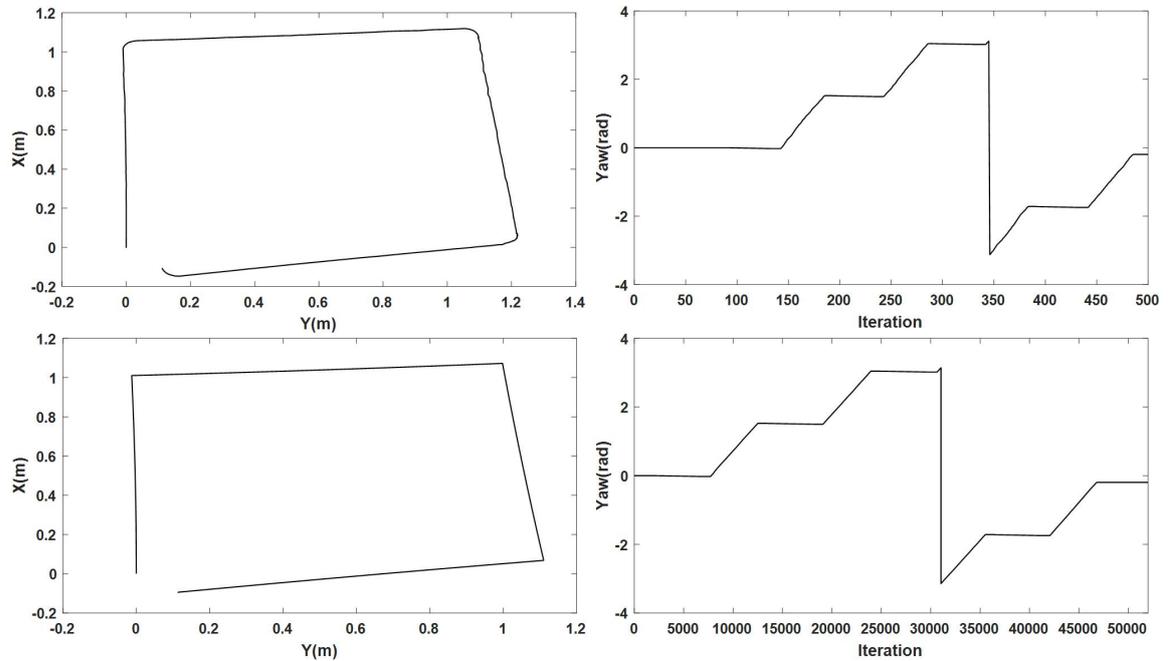


图 2.11 光照，特征充足条件下的 Kinect 位姿解算结果以及实际位姿

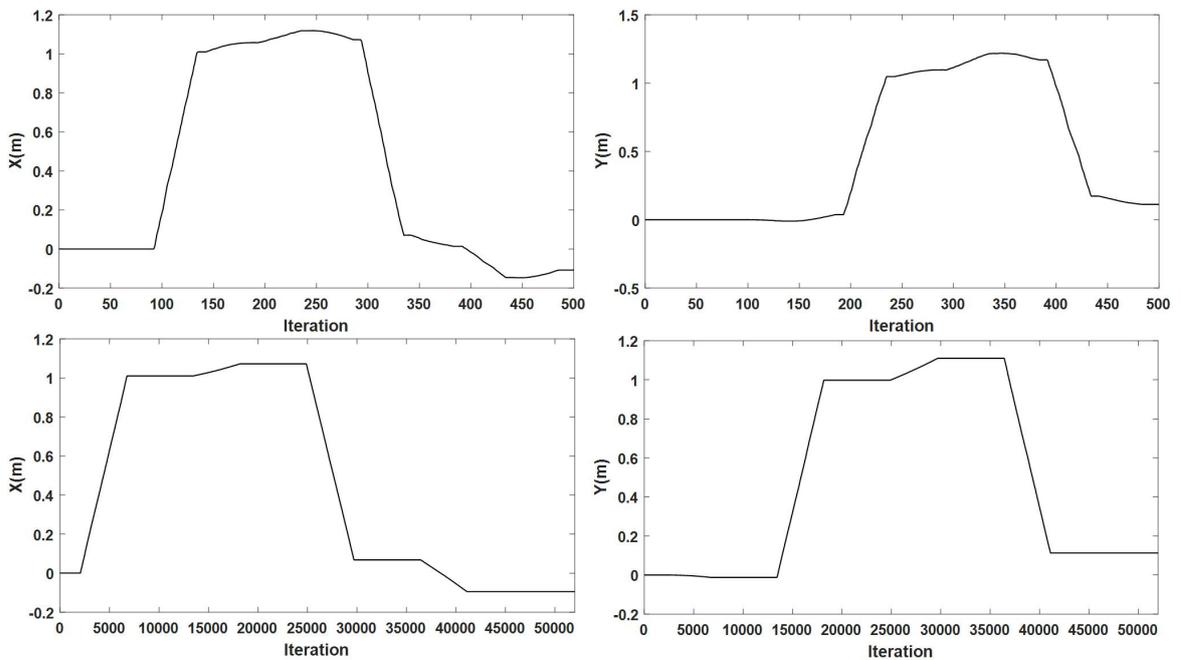


图 2.12 光照，特征充足条件下利用 Kinect 获取的位置以及实际的机器人位置

2.4 本章小结

本章主要介绍了 IMU 和 Kinect 位姿解算的原理和各自存在的问题，通过实现 IMU 和 Kinect 的单传感器位姿解算系统，利用实验证明单传感器系统在进行位姿解算过程中的不可靠性。陀螺仪及线加速度计引发的 IMU 加速度和偏航角测量漂移以及特征不足导致的 Kinect 工作失常将使得移动机器人的位姿与实际位姿具有较大偏差。最后，基于实验结论和相关研究，指出可以通过 IMU 和 Kinect 数据融合克服 IMU 漂移和 Kinect 工作环境限制，实现较高精度位姿结算的可能。

第三章 基于卡尔曼滤波的 IMU 和 Kinect 融合位姿估计

基于 IMU 或 Kinect 的单传感器位姿解算系统由于自身的缺陷或者环境条件的影响使得获取的位姿数据具有不可靠性。由于 IMU 和视觉数据分别是对机器人自身的观测和对环境的观测，具有互补特性，本章将在卡尔曼滤波框架下通过建立移动机器人的线性以及非线性模型，对这一对数据进行融合，优化单传感器位姿解算系统。

3.1 卡尔曼滤波算法

卡尔曼滤波器对线性系统是一种在最小均方差意义下的最优估计器，在非线性条件下则通过对预测方程进行一阶线性近似实现状态量的近优估计。由于卡尔曼滤波器所具有的对状态量“预测-更新”的估计模式，可以利用观测方程中的多个观测变量使得多个传感器的数据统一在一个框架下，实现多传感器数据融合的目的。

3.1.1 线性系统的卡尔曼滤波

在确定状态向量 x 的条件下，给出一个有差分方程描述的离散时间系统，该系统的状态向量受到加性白噪声的影响，则系统的预测方程为：

$$x(k+1) = Fx(k) + Gu(k) + v(k), k = 1, 2, \dots \quad (3.1)$$

观测方程为：

$$z(k) = Hx(k) + w(k), k = 1, 2, \dots \quad (3.2)$$

其中系统矩阵 F ，输入矩阵 G 和输出矩阵 H 分别描述状态向量 $x(k)$ 的变化规律，状态向量受控制输入 $u(k)$ 的影响以及将状态向量变换到观测空间（Measurement Space）的变换。根据系统变化的动态规律，矩阵 F, G 和 H 都可以是时变的。

矩阵 $Q(k)$ 和 $R(k)$ 用来描述滤波器在预测和观测过程中的不确定性，并表示不同状态变量和观测变量之间是否存在耦合关系，它们与系统在预测过程中的噪声 $v(k)$ 和观测过程中的噪声 $w(k)$ 有关：

$$\begin{aligned} E(v(k)v(k)^T) &= Q(k) \\ E(w(k)w(k)^T) &= R(k) \end{aligned} \quad (3.3)$$

针对卡尔曼滤波算法，只要给定初始的状态估计向量 $\hat{x}(0|0)$ 以及对应的协方差矩阵 $P(0|0)$ 就可以递推计算得到之后 $(k+1)$ 时刻的状态向量估计值 $\hat{x}(k+1|k+1)$ 及其协方差

矩阵 $P(k+1|k+1)$ 。卡尔曼滤波算法的算法流程如下：

- ① 预测状态向量

$$\hat{x}(k+1|k) = F\hat{x}(k|k) + Gu(k) \quad (3.4)$$

- ② 预测状态协方差矩阵

$$P(k+1|k) = HP(k+1|k)H^T + Q(k+1) \quad (3.5)$$

- ③ 更新观测向量

$$\hat{z}(k+1|k) = H(k+1|k)\hat{x}(k+1|k) \quad (3.6)$$

- ④ 求解观测向量的残差和卡尔曼增益

$$\begin{aligned} y(k+1) &= z(k+1) - \hat{z}(k+1|k) \\ S(k+1) &= HP(k+1|k)H^T + R \\ K(k) &= P(k+1|k)H^T S(k+1)^{-1} \end{aligned} \quad (3.7)$$

- ⑤ 更新状态向量，将状态向量输出

$$\begin{aligned} \hat{x}(k+1|k+1) &= \hat{x}(k+1|k) + K(k)(z(k+1) - \hat{z}(k+1|k)) \\ &= \hat{x}(k+1|k) + K(k)y(k+1) \end{aligned} \quad (3.8)$$

- ⑥ 更新状态协方差矩阵

$$P(k+1|k+1) = (I - K(k)H)P(k+1|k) \quad (3.9)$$

如图 3.1 所示，系统的状态向量 x 在在控制输入 $u(k-1)$ 和过程噪声 $v(k-1)$ 的作用下从 $(k-1)$ 时刻变化到 k 时刻。 k 时刻的系统状态在观测噪声 $w(k)$ 的影响下得到新的观测向量 $z(k)$ 。

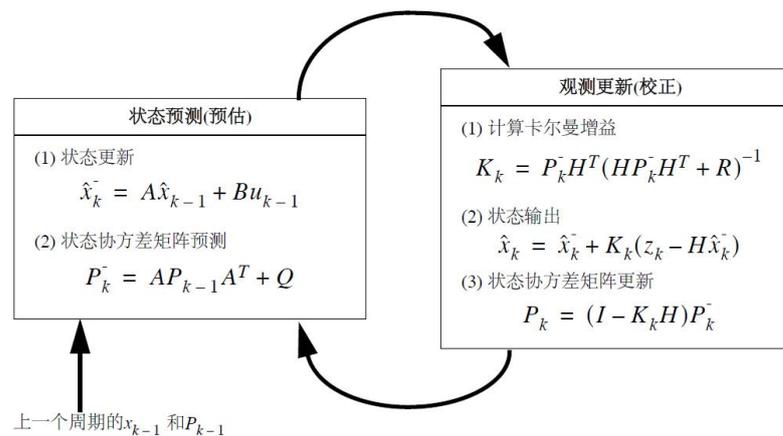


图 3.1 卡尔曼滤波算法

3.1.2 非线性系统的扩展卡尔曼滤波

当尝试使用卡尔曼滤波进行实际的状态估计问题时，常常遇到具有非线性的系统，当系统的状态向量变化较慢的时候，可以尝试近似认为系统在短时间内是线性的，即对系统进行分段线性处理，然后用处理线性系统的卡尔曼滤波器对状态进行估计。但是这种处理有时会导致模型近似度低，收敛缓慢甚至导致滤波输出发散。另外，当系统的状态向量变化率较高的时候，不能利用线性的卡尔曼滤波器求解状态估计问题。针对以上两个问题，扩展卡尔曼滤波器（EKF，Extended Kalman Filter）被提出并且在非线性系统的状态估计问题中被广泛使用。

假设系统的预测和观测方程中分别有非线性函数 $f(k, x(k))$ 和 $h(k, x(k))$ ，即：

$$\begin{aligned} x(k+1) &= f(k, x(k)) + v(k) \\ z(k) &= h(k, x(k)) + w(k) \end{aligned} \tag{3.10}$$

扩展卡尔曼滤波算法就是在每一步上将预测和观测方程线性化，得到近似线性卡尔曼滤波器中系统矩阵和输出矩阵的雅克比矩阵 $F(k)$ 和 $H(k+1)$ ，之后将雅克比矩阵代入线性的卡尔曼滤波流程中，完成正常的滤波流程。即：

$$\begin{aligned} F(k) &= \left. \frac{\partial f(x, x(k))}{\partial x(k)} \right|_{x(k) = \hat{x}(k|k)} \\ H(k+1) &= \left. \frac{\partial h(k+1, x(k+1))}{\partial x(k+1)} \right|_{x(k+1) = \hat{x}(k+1|k)} \end{aligned} \tag{3.11}$$

除了以上对非线性模型进行线性化的过程，两种卡尔曼滤波器的主要区别在于协方差矩阵的更新上，这是因为在扩展卡尔曼滤波器中，每一步的系统矩阵 $F(k)$ 和输出矩阵 $H(k+1)$ 与系统在进行一阶 Taylor 展开过程中的工作点有关。

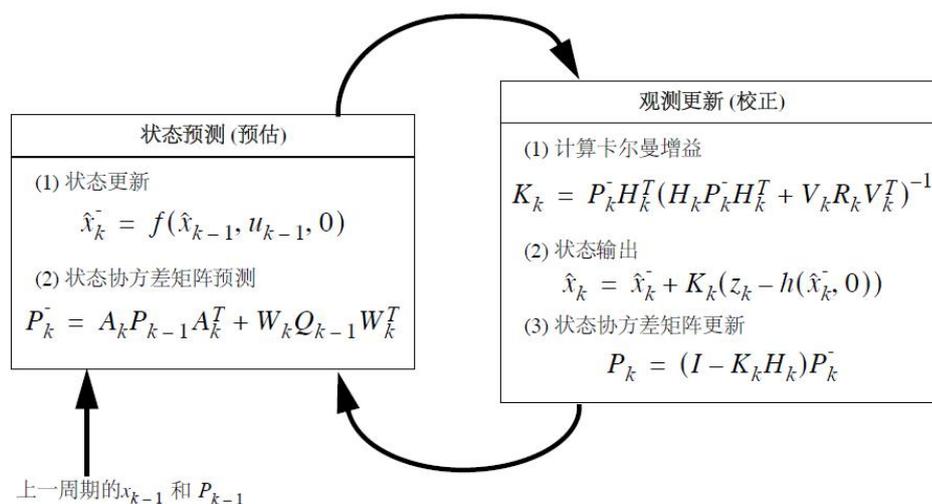


图 3.2 扩展卡尔曼滤波算法

3.2 移动机器人建模

位置和姿态作为移动机器人运动过程中的属性属于机器人的状态，对机器人状态进行估计的前提是对机器人的运动进行定量描述，根据将机器人几何外形抽象程度的不同可以将移动机器人的模型建立成线性或者非线性的。卡尔曼滤波器对系统的线性度有着特殊的要求，所以不同运动模型将直接决定滤波器的选择。

3.2.1 线性模型

在机器人运动过程中，忽略机器人的几何外形以及旋转过程中前后轮的旋转角等细节，将机器人看做一个椭圆柱体，其中长轴方向对应机器人的前进和后退方向，当机器人转弯时，长轴方向也相应变化。

假设机器人的加速度 a 保持恒定，即：

$$\begin{aligned} a_x(k+1) &= a_x(k) \\ a_y(k+1) &= a_y(k) \end{aligned} \quad (3.12)$$

则机器人在世界坐标系的 x 和 y 方向上的速度和位置分别是：

$$\begin{aligned} v_x(k+1) &= v_x(k) + a_x(k)\Delta t \\ v_y(k+1) &= v_y(k) + a_y(k)\Delta t \\ x(k+1) &= x(k) + v_x(k)\Delta t + \frac{1}{2}a_x(k)\Delta t^2 \\ y(k+1) &= y(k) + v_y(k)\Delta t + \frac{1}{2}a_y(k)\Delta t^2 \end{aligned} \quad (3.13)$$

机器人的偏航角 ψ 为：

$$\psi(k+1) = \psi(k) + \omega\Delta t \quad (3.14)$$

容易看出与机器人的平移和旋转有关的各个变量都是关于时段 Δt 的线性函数，即描述机器人运动的模型是线性模型。

3.2.2 非线性模型

在机器人运动过程中，考虑机器人底盘的体积，旋转过程中前后轮的旋转角度以及机器人的运动速度，这样得到的是机器人的非线性模型。对于低速运动的刚体，常采用自行车模型（Bicycle Model）对刚体的运动进行建模，以下即使用该模型来描述移动机器人的运动。

当机器人进行旋转运动时，前后轮具有相对于水平轴不同的夹角 δ_f 和 δ_r ，垂直于两轮的方向所得交点即机器人在旋转过程中的瞬时中心 O 。瞬时中心 O 距机器人质心的连

线长度为 R ，即旋转半径。线速度的方向与曲率半径垂直，则前后轮受到的合力为：

$$F = \frac{mV^2}{R} \quad (3.15)$$

其中 m 和 V 分别是机器人的质量和速度。

该力将有可能使得速度对前后轮的作用不仅是改变角速度，还有可能使得朝向发生变化，从而使得前后轮具有较大偏离角（Slip Angle），但是由于向心力的大小与刚体速度的二次方成正比，对于低速运动的刚体（通常认为速度低于 5m/s 为低速运动）可以认为偏离角很小，即速度对于前后轮的作用仅仅是在于改变前后轮的角速度。

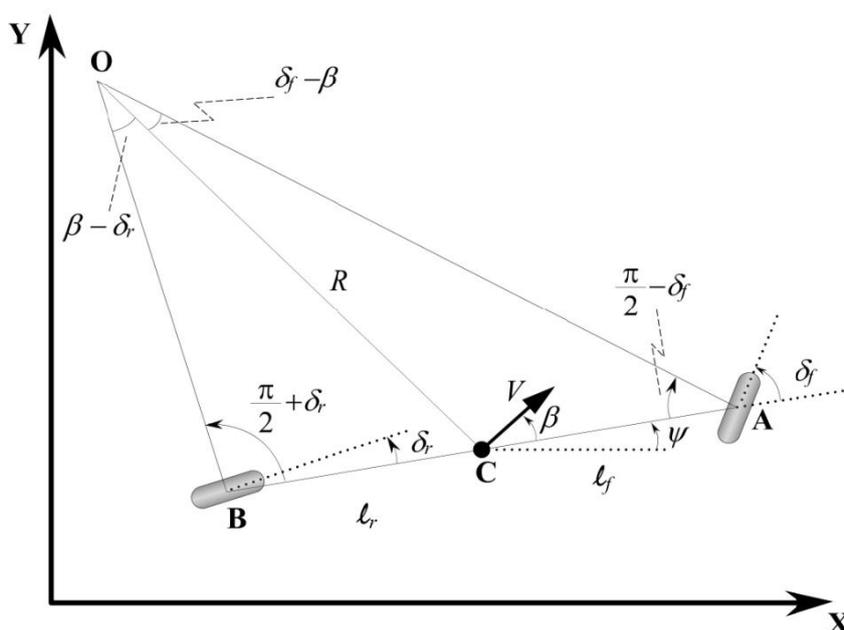


图 3.3 自行车模型

如图 3.3 所示， ψ 是机器人的偏航角，则机器人相对与行进路线的角度为：

$$\gamma = \psi + \beta \quad (3.16)$$

对 $\triangle OCA$ 和 $\triangle OCB$ ，有：

$$\frac{\sin(\delta_f - \beta)}{l_f} = \frac{\sin(\frac{\pi}{2} - \delta_f)}{R}$$

$$\frac{\sin(\beta - \delta_r)}{l_r} = \frac{\sin(\frac{\pi}{2} + \delta_r)}{R} \quad (3.17)$$

即：

$$\begin{aligned}\frac{\sin \delta_f \cos \beta - \sin \beta \cos \delta_f}{l_f} &= \frac{\cos \delta_f}{R} \\ \frac{\cos \delta_r \sin \beta - \cos \beta \sin \delta_r}{l_r} &= \frac{\cos \delta_r}{R}\end{aligned}\quad (3.18)$$

对上式两侧分别与 $l_f / \cos \delta_f$ 和 $l_r / \cos \delta_r$ 相乘，得到

$$\begin{aligned}\tan \delta_f \cos \beta - \sin \beta &= \frac{l_f}{R} \\ \sin \beta - \tan \delta_r \cos \beta &= \frac{l_r}{R}\end{aligned}\quad (3.19)$$

从而

$$(\tan \delta_f - \tan \delta_r) \cos \beta = \frac{l_f + l_r}{R}\quad (3.20)$$

由于机器人运动速率较低，可以认为曲率半径变化的速率也较低，则可以认为：

$$\omega = \dot{\psi} = \frac{V}{R}\quad (3.21)$$

可得

$$\omega = \dot{\psi} = \frac{V \cos \beta}{l_f + l_r} (\tan \delta_f - \tan \delta_r)\quad (3.22)$$

综合以上表达式可以得到描述机器人的运动模型：

$$\begin{aligned}\dot{x} &= V \cos(\psi + \beta) \\ \dot{y} &= V \sin(\psi + \beta) \\ \omega = \dot{\psi} &= \frac{V \cos \beta}{l_f + l_r} (\tan \delta_f - \tan \delta_r)\end{aligned}\quad (3.23)$$

对于后轮不能旋转的情况，取 $\delta_r = 0$ 得到机器人的运动模型为：

$$\begin{aligned}\dot{x} &= V \cos(\psi + \beta) \\ \dot{y} &= V \sin(\psi + \beta) \\ \omega = \dot{\psi} &= \frac{V \cos \beta}{l_f + l_r} \tan \delta_f\end{aligned}\quad (3.24)$$

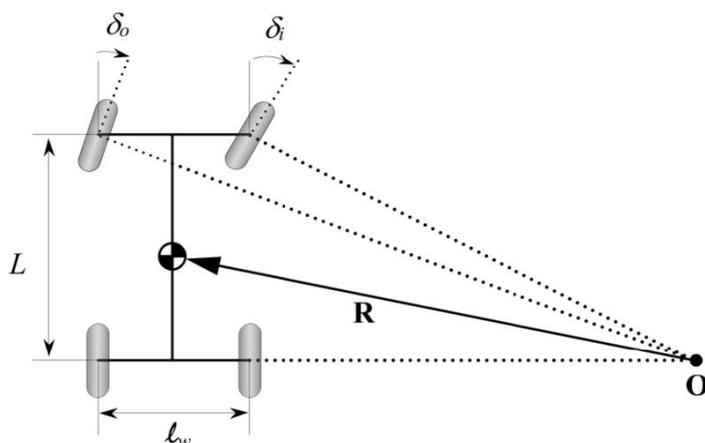


图 3.4 一般化的自行车模型

需要注意的是，自行车模型中只考虑了只有一个前轮的情况，对于两个前轮的情况，如图 3.4 所示，当两轮间的轴距 l_w 较大时，两轮间旋转角的关系为：

$$\delta_i - \delta_o = \frac{L}{R^2} l_w \quad (3.25)$$

对于底盘较小的移动机器人，由于 l_w 和 L 较小，该差值可以忽略，即认为两个前轮的旋转角度相等。

从基于自行车模型的移动机器人模型看，由于状态量之间具有非线性函数关系，该模型是非线性模型。

3.3 融合 IMU 和 Kinect 的卡尔曼滤波器

以下将针对 IMU 和 Kinect 两种传感器数据融合的情形在卡尔曼滤波器框架下，基于两类移动机器人模型对用于移动机器人位姿估计的滤波器进行设计。

3.3.1 状态向量和系统模型

在利用卡尔曼滤波器解决移动机器人状态估计问题的过程中，首先需要明确需要估计状态向量，只有在明确状态向量的前提下，才能够建立描述移动机器人运动的预测模型。以下根据两类机器人的建模方法，分别利用线性模型和非线性模型选择状态变量，并基于状态变量建立卡尔曼滤波器的预测模型。

3.3.1.1 线性模型

移动机器人在室内的运动主要分为两类：平移和旋转，以下是描述平移运动和旋转运动的数学模型。

针对平移运动，机器人可能有匀速运动，匀加速运动以及静止三类情况。由于需要

将这三类运动统一到同一个数学模型中，所以对于每一类情况，都具有状态变量：

$$X = [x, \dot{x}, \ddot{x}]^T \quad (3.26)$$

或者

$$Y = [y, \dot{y}, \ddot{y}]^T \quad (3.27)$$

其中 \ddot{x}, \ddot{y} 表示加速度， \dot{x}, \dot{y} 表示线速度，而 x, y 则表示位置。

当机器人处于匀加速运动，匀速运动和静止状态下，其数学模型分别为：

$$\begin{aligned} X_{k+1} &= \begin{bmatrix} 1 & \Delta t & \frac{1}{2}\Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} X_k, X_{k+1} = \begin{bmatrix} 1 & \Delta t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} X_k, X_{k+1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} X_k \\ Y_{k+1} &= \begin{bmatrix} 1 & \Delta t & \frac{1}{2}\Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} Y_k, Y_{k+1} = \begin{bmatrix} 1 & \Delta t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} Y_k, Y_{k+1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} Y_k \end{aligned} \quad (3.28)$$

针对旋转运动，机器人则具有匀速运动和静止两类情况，其中由于 IMU 以及本课题中的其他传感器都不能测量角加速度，不考虑机器人处于匀加速旋转的情况。对于机器人旋转的两类情况，状态向量为：

$$\Psi = [\psi, \dot{\psi}]^T \quad (3.29)$$

其中 $\dot{\psi}, \psi$ 分别表示角速度和角位移。

则当机器人处于匀速旋转以及静止状态下，其数学模型分别为：

$$\Psi_{k+1} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \Psi_k, \Psi_{k+1} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \Psi_k \quad (3.30)$$

以上仅仅按照对机器人运动情况分类的方法对机器人的各类运动选择对应的状态变量以及建立对应的预测方程。实际中，一方面机器人的直线运动分别发生在世界坐标系的 x 和 y 轴上，另一方面需要用统一的预测模型描述移动机器人在二维平面上可能发生的所有运动。

以下分别使用二阶和三阶模型描述机器人在二维平面上的运动。

当认为机器人在平面上只能匀速运动时，使用二阶匀速运动模型（CV, Constant Velocity）建立机器人运动的预测模型，即：

$$\begin{aligned} \mathbf{x}_{k+1} &= [X_{k+1}, Y_{k+1}, \Psi_{k+1}]^T \\ &= \begin{bmatrix} 1 & \Delta t & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta t & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_k + \mathbf{v}_k \end{aligned} \quad (3.31)$$

当认为机器人在平面上不仅能发生匀速运动，还可能进行匀加速运动时，使用三阶匀加速度运动模型（CA, Constant Acceleration）建立机器人运动的预测模型，即：

$$\begin{aligned} \mathbf{x}_{k+1} &= [X_{k+1}, Y_{k+1}, \Psi_{k+1}]^T \\ &= \begin{bmatrix} 1 & \Delta t & \frac{1}{2}\Delta t^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \Delta t & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta t & \frac{1}{2}\Delta t^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_k + \mathbf{v}_k \end{aligned} \quad (3.32)$$

在匀速运动时使用二阶匀速运动模型而不是三阶匀加速运动模型可以有效屏蔽 IMU 的加速度噪声在状态估计过程中对位置求解的影响，而在加速运动时使用三阶匀加速运动模型则可以精确地描述系统的运动。如果使用二阶模型描述加速度运动会导致系统在加速度输入下实际状态与估计状态具有较大偏差，且该偏差随着时间的推移不断增大。

3.3.1.2 非线性模型

使用自行车模型描述机器人运动，根据

$$\begin{aligned} \dot{x} &= V \cos(\psi + \beta) \\ \dot{y} &= V \sin(\psi + \beta) \\ \omega &= \dot{\psi} = \frac{V \cos \beta}{l_f + l_r} \tan \delta_f \end{aligned} \quad (3.33)$$

考虑在采样周期 Δt 内机器人速度的大小和方向保持恒定，可得机器人的位置：

$$\begin{aligned} x &= \int_{\Delta t} V \cos(\psi(t) + \beta) dt \\ y &= \int_{\Delta t} V \sin(\psi(t) + \beta) dt \end{aligned} \quad (3.34)$$

机器人的加速度:

$$\begin{aligned} \ddot{x} &= -V\omega \sin(\psi + \beta) \\ \ddot{y} &= V\omega \cos(\psi + \beta) \end{aligned} \quad (3.35)$$

机器人的角位移

$$\psi = \frac{V \cos \beta}{l_f + l_r} \tan \delta_f \Delta t \quad (3.36)$$

当系统中的每个状态量都受到系统内外其他变量的干扰时，即给每个状态变量施加加性噪声，即可得到机器人的非线性观测模型。

3.3.2 观测向量和观测模型

第二章的实验结果表明，在单一的 IMU 和 Kinect 位姿解算系统中，IMU 的加速度数据和 Kinect 的偏航角 ψ 具有不可靠性。在将 IMU 和 Kinect 传感器数据融合时，选择利用 IMU 的偏航角以及利用 Kinect 数据解算得到的位置数据作为观测向量，即

$$z = [x, y, \psi]^T \quad (3.37)$$

由于观测量与状态向量中的状态量相对应，不存在传感器数据的检测变送过程，则输出矩阵为

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (3.38)$$

则观测模型为

$$\begin{aligned} z_k &= Hx_k + w_k \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} x_k + w_k \end{aligned} \quad (3.39)$$

3.3.3 协方差矩阵

3.3.3.1 状态协方差

状态协方差矩阵 P 用来描述在假设滤波器所有参数选择适宜的条件下，滤波器输出的各个状态变量的不确定性。状态协方差在实际中是很难通过实验确定的，通常认为观测协方差中对于各个观测量的描述可以反映状态协方差对应的状态量的不确定程度。

由于在本课题中观测向量仅包括部分状态量，对于没有观测的状态量，考虑实际实验条件，取最差情况，即 3 倍的偏差量。

$$\sigma_v^2 = 3\delta_v, \sigma_a^2 = 3\delta_a, \sigma_\omega^2 = 3\delta_\omega \quad (3.40)$$

对于观测的状态量，根据观测协方差矩阵 R 进行取值。

3.3.3.2 过程协方差

过程协方差 Q 用来描述滤波器在预测状态向量的过程中受到未被包括到状态向量中的其他未知变量的影响，只有适宜的 Q 才能保证滤波器输出能够近似真实输出。但是由于难以通过实验确定所选状态向量在预测过程中受到其他变量的影响，所以只能通过间接的方法确定状态变量受到的影响。

由于机器人的位置和速度受到加速度的影响，姿态受到角速度的影响，考虑加速度和角速度受到未知变量的影响即可得到整个状态向量受到外界环境的影响。

对于三阶匀加速运动模型，将噪声向量改写为加速度受到扰动的形式，即将预测模型改写为：

$$X_{k+1} = \begin{bmatrix} 1 & \Delta t & \frac{1}{2}\Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} X_k + \Gamma w_k, Y_{k+1} = \begin{bmatrix} 1 & \Delta t & \frac{1}{2}\Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} Y_k + \Gamma w_k \quad (3.41)$$

其中 Γ 是噪声增益，是环境对加速度的影响。

由于外界扰动，外界对系统速度和位置的影响分别为：

$$\sigma_v^2 = \Delta t w, \sigma_x^2 = \frac{1}{2} \Delta t^2 w \quad (3.42)$$

则噪声增益为：

$$\Gamma = \begin{bmatrix} \frac{1}{2} \Delta t^2 \\ \Delta t \\ 1 \end{bmatrix} \quad (3.43)$$

则过程噪声协方差为：

$$Q = E[\Gamma w(t) w(t) \Gamma^T] = \begin{bmatrix} \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} & \frac{\Delta t^2}{2} \\ \frac{\Delta t^3}{2} & \Delta t^2 & \Delta t \\ \frac{\Delta t^2}{2} & \Delta t & 1 \end{bmatrix} \sigma_a^2 \quad (3.44)$$

对于二阶恒速度模型

$$X_{k+1} = \begin{bmatrix} 1 & \Delta t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} X_k + \Gamma w_k, Y_{k+1} = \begin{bmatrix} 1 & \Delta t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} Y_k + \Gamma w_k \quad (3.45)$$

类似可以得到过程噪声协方差为

$$Q = \begin{bmatrix} \Delta t^2 & \Delta t \\ \Delta t & 1 \end{bmatrix} \sigma_a^2 \quad (3.46)$$

对于旋转模型，利用相同的方法，将噪声向量改写为角速度受到扰动的形式，可以得到类似与二阶恒速度模型的过程协方差矩阵

$$Q = \begin{bmatrix} \Delta t^2 & \Delta t \\ \Delta t & 1 \end{bmatrix} \sigma_\omega^2 \quad (3.47)$$

对于 σ_a^2 和 σ_ω^2 的选取，通常将其设定为采样周期中线加速度和角加速度最大变化值的 $\frac{1}{2}$ 到最大变化值之间，即

$$\begin{aligned} \sigma_a^2 &= \left(\frac{1}{2} \sim 1\right) \Delta a_{max} \\ \sigma_\omega^2 &= \left(\frac{1}{2} \sim 1\right) \Delta \omega_{max} \end{aligned} \quad (3.48)$$

3.3.3.3 观测协方差

观测协方差用来描述各个传感器在测量相对应的变量的过程中，由于噪声的因素而引起的测量不确定性以及各个传感器测量值之间的相关性。

由于选择的观测变量包括利用 Kinect 得到的位置以及利用 IMU 得到的姿态，以下分别进行讨论。

针对位置。由于 Kinect 容易环境中光照以及特征缺乏的影响，不可能选择一个适合于所有场景下的协方差值，考虑实验中的最坏情况，同时假设 x 和 y 方向的位置服从正态分布，取 x 和 y 的方差值为横向和纵向最大位移的三倍，即

$$\begin{aligned} \sigma_x^2 &= 3x_{des} \\ \sigma_y^2 &= 3y_{des} \end{aligned} \quad (3.49)$$

针对姿态。本课题中使用的 IMU 包含的陀螺仪保持测量精度条件下载体的最大转速为 110deg/s 。考虑匀速旋转运动，则姿态 ψ 的偏移为：

$$\delta_\psi = 20^\circ (\approx 0.3491 \text{rad}) \quad (3.50)$$

假设姿态 ψ 服从正态分布，考虑最坏情况，则：

$$\sigma_\psi^2 = 3\delta_\psi^2 = 0.3655 \quad (3.51)$$

由于 x 和 y 之间的相关关系不能确定，同时显然对位置的计算与对姿态 ψ 的测量无关，设 x 和 y 与 ψ 之间没有耦合关系，则观测协方差矩阵为：

$$R = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\psi^2 \end{bmatrix} \quad (3.52)$$

对于移动机器人走边长为 1m 的正方形闭环的情况，观测协方差矩阵为：

$$R = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 0.3655 \end{bmatrix} \quad (3.53)$$

3.4 实验及结果分析

3.4.1 匀速闭环实验

控制 TurtleBot 移动机器人以 0.2m/s 的恒定速度走边长为 1m 的正方形闭环，在融合 IMU 和偏航角数据以及利用 Kinect 得到的位置数据条件下，利用卡尔曼滤波器获取融合位姿数据，利用机器人所在惯导坐标系下 /base_link 到机器人位置的世界坐标系 /odom 之间的坐标变换作为参考实际值，从而评价匀速闭环条件下融合 IMU 和 Kinect 进行位姿解算的精度。

如图 3.5 所示是机器人在走闭环过程中的实际位置和姿态。

如图 3.6 所示是利用 Kinect 数据进行位姿解算得到的位置和姿态以及融合 IMU 和 Kinect 数据的得到的融合后的位置和姿态。

从 Kinect 获取的位姿可以看出，对于位置，在机器人行走到最后一条边时出现了特征不足而导致的位姿解算失败的问题，从而使得位姿解算节点中的闭环检测部分引发强制闭环，出现了按照最短距离直接归零的错误结果。对于姿态，同样由于机器人行走到最后一条边时 Kinect 受到环境特征不足的影响，使得最后一部分（Iteration ≥ 325 ）的姿态完全错误。

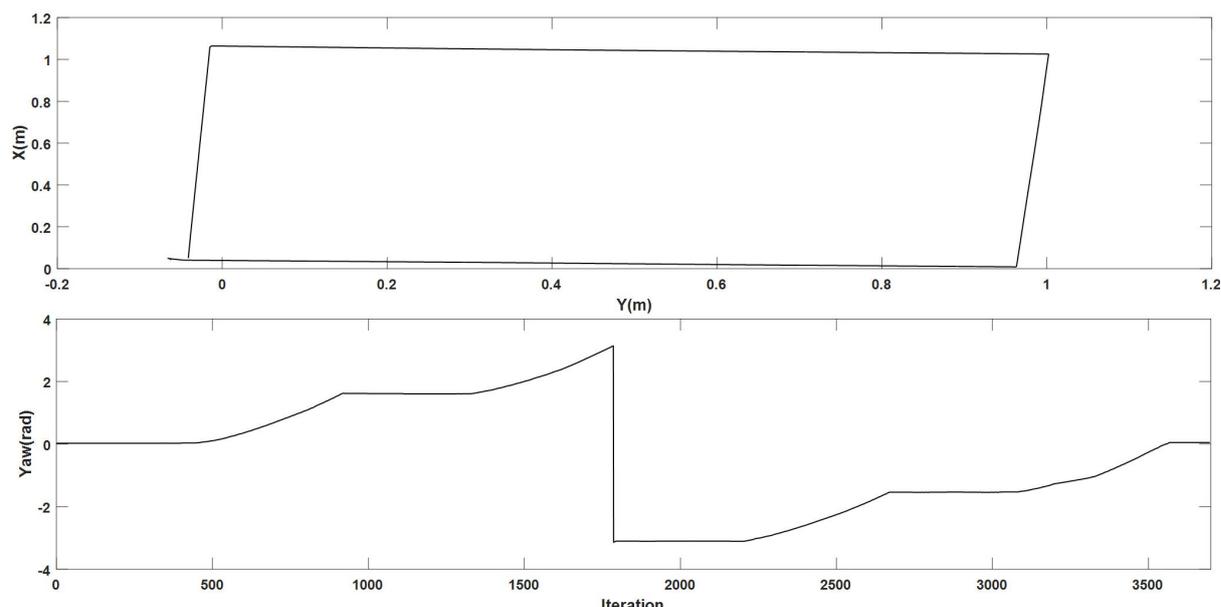


图 3.5 匀速闭环实际位姿

从融合 IMU 和 Kinect 的位姿可以看出，对于位置，通过引入 IMU 中的偏航角信息以及精确的预测模型，使得最后一部分的位置平稳过渡起始位置附近，而对于姿态，则获取了与实际姿态相近结果，在 Kinect 位姿解算的基础上纠正了错误的结果。

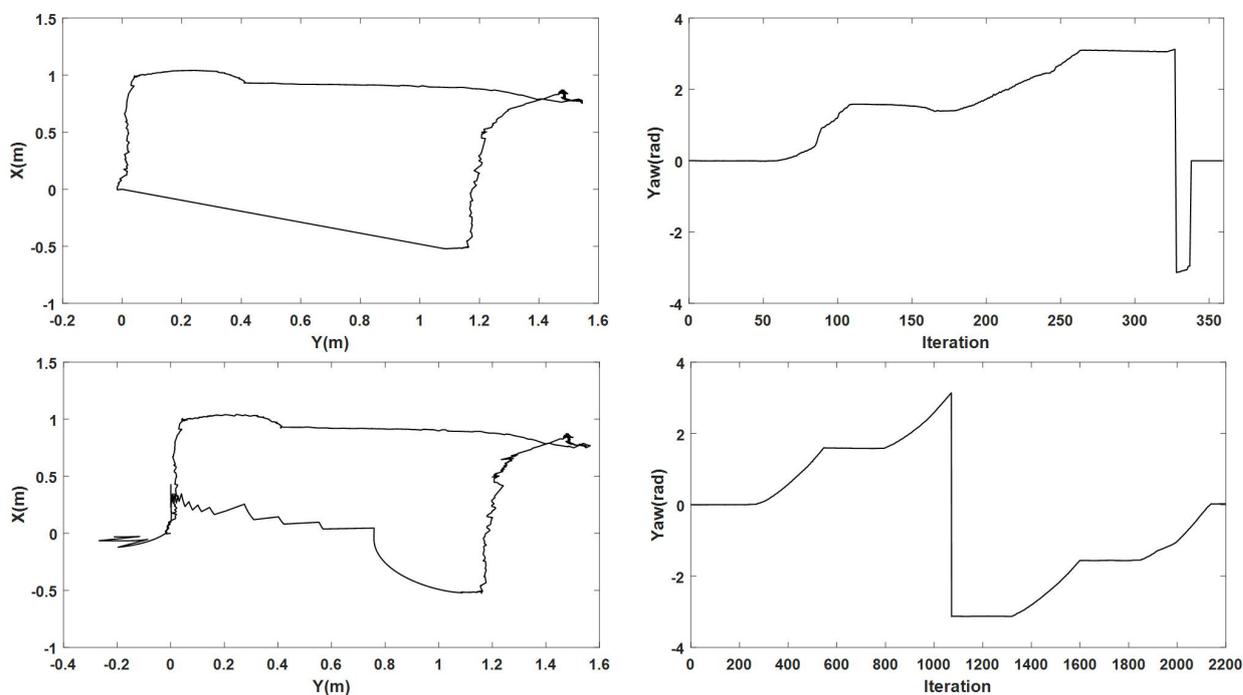


图 3.6 匀速闭环 Kinect 以及融合滤波位姿

如图 3.7 所示是机器人在匀速走闭环过程中的速度变化以及 IMU 和 Kinect 融合滤波之后的速度，虽然速度并没有用到后续的运算中，但是从速度的变化趋势可以看出机器人在最后一条边上运动时获取的速度数据不能体现机器人的匀速运动，同时，在其他

部分上也出现了不应该存在的尖峰，虽然利用从 IMU 获取的数据对这部分的速度估计进行了一定程度的纠正，但是受到环境限制的 Kinect 输出的位姿数据数据存在问题（如图 3.6 所示），整体效果与实际情况相差较多。

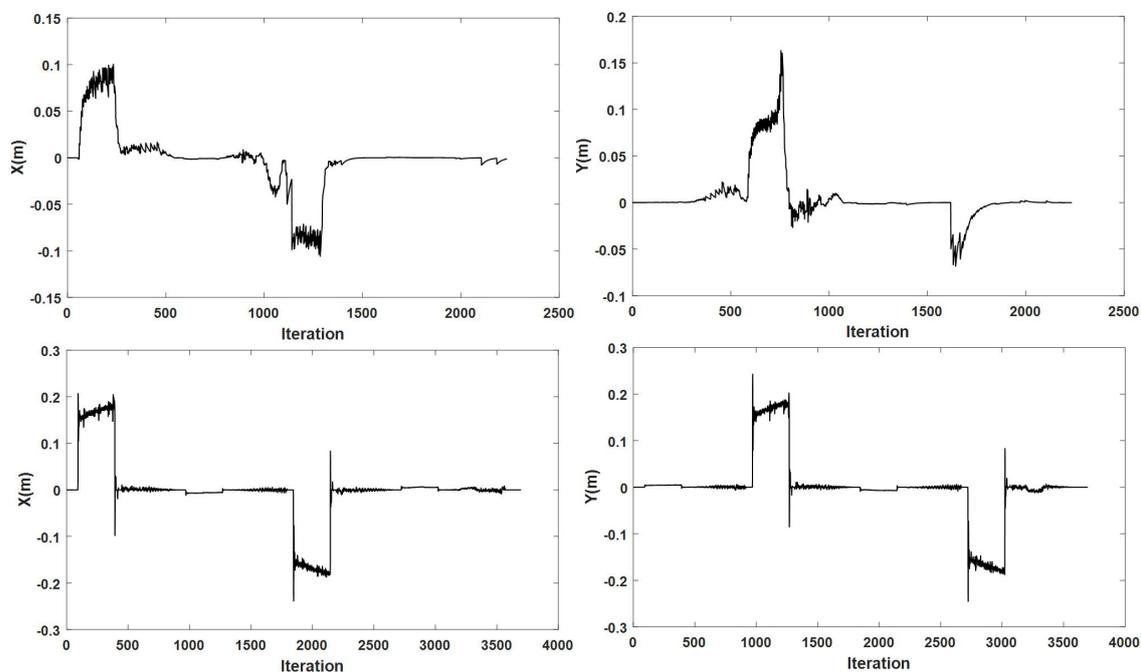


图 3.7 匀速闭环实际速度变化以及融合滤波速度变化

3.4.2 匀加速闭环实验

控制 TurtleBot 移动机器人以 0.1m/s^2 的加速度的恒定加速度走边长为 1m 的正方形闭环。对于正方形的每一条边，初始速度为 0m/s ，之后按照 0.1m/s^2 的加速度加速到 0.2m/s 之后按照 0.2m/s 的速度匀速运动。另外在经过每次旋转，在开始走新的边的时候，停留 5s。其他的实验设定与匀速闭环实验保持一致。

如图 3.8 所示是 TurtleBot 在按照设定的运动规律走闭环的过程中实际的位置和姿态。

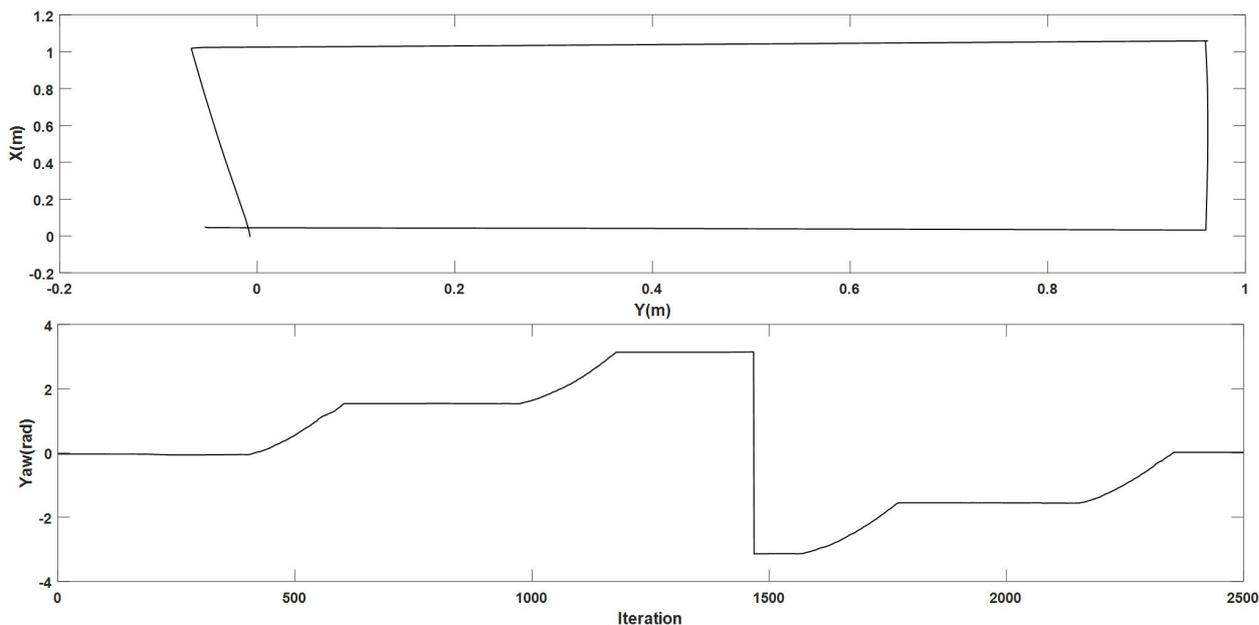


图 3.8 匀加速闭环实际位姿

如图 3-9 所示是利用 Kinect 数据进行位姿解算得到的位置和姿态以及引入 IMU 的偏航角数据进行融合滤波之后得到的位置和姿态。

从 Kinect 位姿解算结果可以看出，除了姿态在最后一部分以外，在特征充足的条件下位姿解算的结果基本正确，但是融合的姿态在最后不仅无法闭环，而且出现了位置“折返”的情况，这是由于 Kinect 得到的姿态对滤波过程中的位置解算造成了一定的影响，使得位置的解算出现问题。

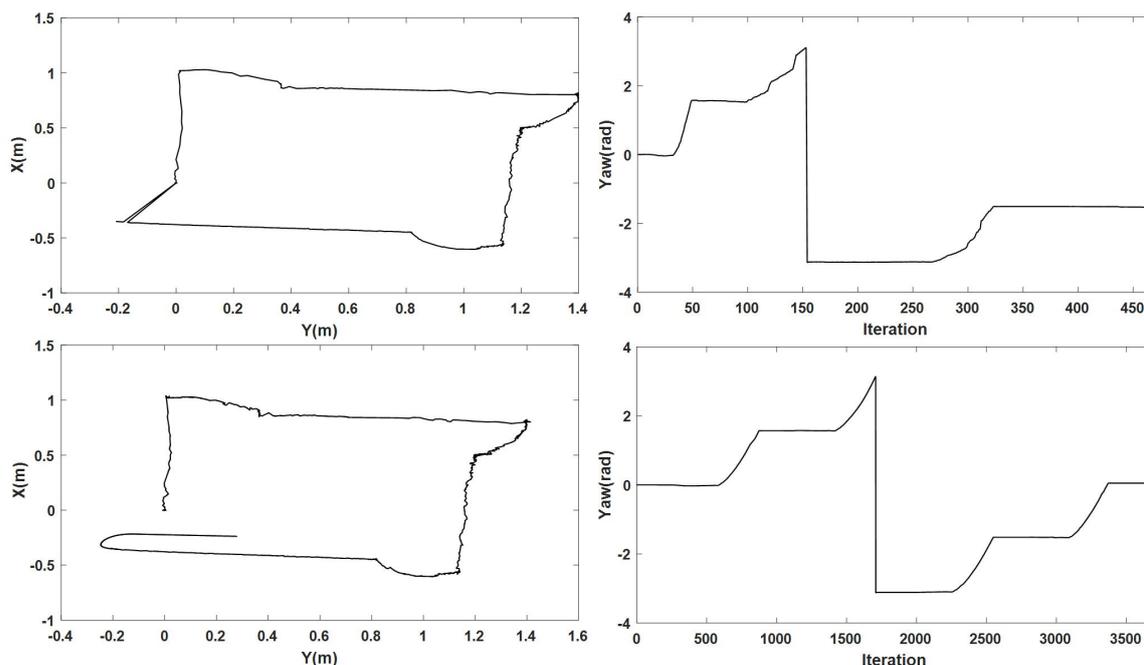


图 3.9 匀加速闭环 Kinect 以及融合滤波位姿

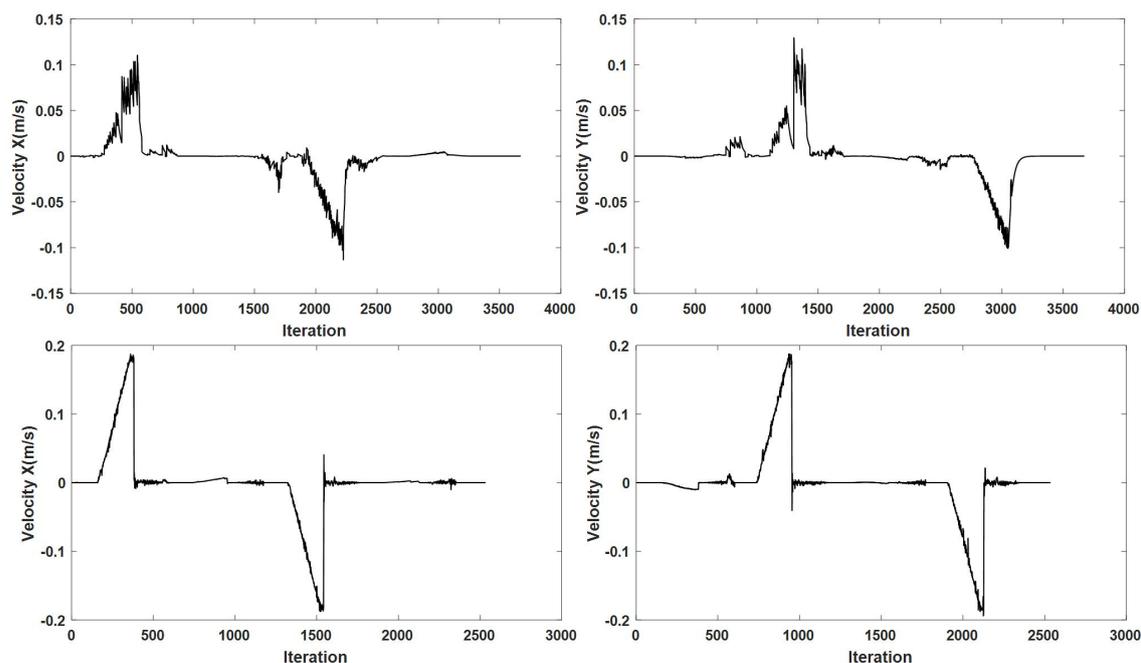


图 3.10 匀加速闭环实际速度变化以及融合滤波速度变化

如图 3-10 所示是机器人在按照设定走闭环的过程中实际的速度变化以及经过融合滤波估计出的速度变化。从滤波器的速度状态量可以看出，由于机器人在所有状态下的运动都由固定数学模型的一个滤波器进行处理，不论机器人处于匀速运动，匀加速运动还是静止状态，与实际速度相比，速度值都呈现出较大的振荡，比如 y 方向的速度在 Iteration 为 900 时，出现了不应存在的上升以及对错误速度值的保持，而在 Iteration 处于 1000~1500 时，机器人处于匀加速运动状态，但是滤波输出的速度出现了向 0 方向的振荡。

3.4.3 实验结果分析

分析融合 IMU 和 Kinect 数据进行移动机器人位姿估计结果可以发现融合滤波的位姿对单一传感器位姿解算结果进行了纠正，使得机器人的位置屏蔽了强制闭环的影响，实现了在走闭环过程中机器人轨迹向原点的平稳过渡，同时使得机器人的姿态克服了特征缺乏以及光照的影响，通过利用 IMU 提供的直接测量信息，在卡尔曼滤波框架下获取正确的姿态。

但是当机器人的运动模式不断在静止，匀速运动以及匀加速运动之间切换，由于使用单一的预测模型进行状态估计，位置出现了错误的“折返”现象，而速度则出现了振荡振荡的结果。考虑实验结果，由于单一预测模型不足以描述机器人在运动过程中的不同运动模式，更不能在这些不同的运动模式间通过自身的变化来反映机器人状态的变

化。考虑使用基于多个模型的自适应滤波估计算法，即交互多模型算法框架，来解决这一问题。交互多模型算法框架利用基于不同预测模型的多个并列的滤波器实现使用的滤波器随着机器人的运动状态实时切换，从而纠正运动模型不断切换过程中导致的误差，获取相对于单模型滤波融合算法更加精确的机器人位姿。

3.5 本章小结

本章主要阐述了融合 IMU 和 Kinect 的卡尔曼滤波器的原理和实现。首先通过介绍卡尔曼滤波算法的原理和移动机器人的线性和非线性模型说明利用卡尔曼滤波器提升位姿解算精度的可能性，之后在卡尔曼滤波器框架下设计用于融合 IMU 和 Kinect 传感器的位姿的线性卡尔曼滤波器和扩展卡尔曼滤波器，最后通过实验得出融合了 IMU 和 Kinect 传感器数据的两种滤波器都可以对估计结果进行总体的改善，但是在部分细节处仍具有优化空间的结论。

第四章 基于 IMM-KF 的 IMU 与 Kinect 融合位姿估计

基于 EKF 的融合 IMU 和 Kinect 位姿估计虽然可以提升单传感器系统的位姿解算精度，但是单一模型不能描述机器人运动模式不断切换过程中系统的动态。通过确定 Markov 状态转移矩阵，基于多个模型进行融合滤波的交互多模型（IMM）算法通过模型之间的交互获取每一个子模型对输出的权重。当机器人运动模式变化时，描述对应运动模式的权重将相应变化，这使得多模型的加权输出具有应对运动模式变化的能力，实现在单模型系统上位姿解算精度的提升。

4.1 交互多模型滤波算法

混合系统被认为是一种在专家框架（panel of experts）下进行状态估计的重要方向。由于不仅包含了系统的状态，还包含了系统运行模式的变化，混合系统在描述结构和参数变化的系统方面具有很大的优势，而处理混合系统问题的多模型估计被认为是一种可以有效处理系统结构以及参数变化问题的自适应估计方法。

4.1.1 系统描述

随机离散时间混合系统指的是定义在包括系统状态以及系统运行模式的混合空间 $R^n \times S$ 上的系统：

$$\begin{aligned} x_{k+1} &= f_k(x_k, s_{k+1}) + g_k(s_{k+1}, x_k, v_k(s_{k+1}, x_k)) \\ z_k &= h_k(x_k, s_k) + w(s_k, x_k) \end{aligned} \quad (4.1)$$

其中 $k \in N$ 是离散时间变量， $x_k \in R^n$ 是基础状态空间 R^n 上的状态向量， $s_k \in S$ 表示系统模式空间 S 上的模式变量， $z_k \in R^m$ 是对系统状态的观测向量， v_k, w_k 分别表示系统的过程噪声和观测噪声。

假设系统模式序列是一个依赖于系统状态的 Markov 链，且是 φ 的函数，则系统模式的变化具有以下转移概率：

$$P(s_{k+1} = s^{(j)} | s^{(i)}, x_k) = \varphi_k(s^{(i)}, s^{(j)}, x_k) \quad (4.2)$$

这表明状态向量依赖于模式，而且根据观测方程，对系统的观测也与系统的模式有关，即系统模式序列是一个隐藏的 Markov 模型。

当 Markov 链是时不变的时候，记从模式 $s^{(i)}$ 到 $s^{(j)}$ 模式的状态转移概率为 π_{ij} ，因而具有线性形式的混合系统描述为：

$$\begin{aligned} x_{k+1} &= F_k(s_k)x_k + v_k(s_k) \\ z_k &= H_k(s_k)x_k + w_k(s_k) \\ P(s_{k+1} = s^{(j)} | s^{(i)}) &= \pi_{ij}, \forall s^{(i)}, s^{(j)} \in S \end{aligned} \quad (4.3)$$

当系统的运行模式不确定的时候，以上系统仍然是非线性的，但是当给定系统的运动模式时，该系统转化为一个线性系统，即跳变线性系统。

4.1.2 多模型方法

由于使用了多个模型来更好地近似系统的动态，多模型方法（MM）是估计混合系统的主要方法。多模型估计由以下几部分组成。

4.1.2.1 模型设计和滤波器选择

设计一个由有限个模型构成的模型集：

$$M = \{m^{(j)}\}_{j=1,2,\dots,r} \quad (4.4)$$

其中每个模型 $m^{(j)}$ 是对模式空间中对应的模式 $s^{(i)}$ 的一种描述，这种描述可以是一一对应的，也可以不是。当存在模型集合对应于多个模式的情况时，模型集通常会远小于模式集。从而，模型集与模式集的对应关系可以描述为：

$$m_k^{(j)} = \{s_k = m^{(j)}\}, j = 1, 2, \dots, r \quad (4.5)$$

即在 k 时刻的混合系统的模式可以利用模型 $m^{(j)}$ 进行匹配。

当确定模型集后，通过选择如卡尔曼滤波等递推滤波器来完成混合估计。

4.1.2.2 估计融合

为了产生混合系统的融合估计，可以采用软决策，硬决策以及随机决策三种策略，常采用软决策方法产生融合估计。

软决策不硬性规定使用那些滤波器的输出，而是利用所有滤波器的输出来获得估计。把状态的条件均值作为估计，则融合估计由所有的滤波器估计利用概率加权并求和得到：

$$\hat{x}_{k|k} = E(x_k | z^k) = \sum_i \hat{x}_{k|k}^{(i)} P(m_k^{(i)} | z^k) \quad (4.6)$$

4.1.2.3 滤波器的重初始化

如何重新初始化滤波器是确定多模型滤波算法有效性的关键。对于混合系统，由于任意时刻下系统的模型可以是 r 个可能的模型之一，则到时刻 k ，混合系统不同模式的

变化就包括 r^k 个，即对该系统的最优状态滤波器的计算量随着时间的延长而呈指数增长，利用这种方法得到的最优滤波器在现实中由于计算资源的限制将是不可能实现的。在实际中常采用交互多模型算法来近似最优滤波器。

4.1.3 交互多模型算法

交互多模型算法在进行混合估计时由于效率和性能的优势，被认为是一种最有效的多模型算法。在时刻 k 利用交互多模型算法进行状态估计时，认为混合系统的当前状态可以利用模型集中的每一个滤波器进行描述。

交互多模型算法由以下四步组成：

(1) 模型条件重初始化

模型条件重初始化假定混合系统的当前状态可以利用模型集中第 j 个模型来描述，根据软决策策略，与该模型对应的滤波器输入由上一个时刻的各个滤波器的估计融合产生。

假定 $(k-1)$ 时刻的模型为 $m_{k-1}^{(i)}$ ，而在 k 时刻匹配的模型则是 $m_k^{(j)}$ ，则以 z^{k-1} 为条件的混合概率为：

$$\mu_{k-1|k-1}^{(i,j)} = P(m_{k-1}^{(i)} | m_k^{(j)}, z^{k-1}) = \frac{\pi_{ij} \mu_{k-1}^{(i)}}{\sum_{i=1}^r \pi_{ij} \mu_{k-1}^{(i)}}, \quad i, j = 1, 2, \dots, r \quad (4.7)$$

对于 $j=1, 2, \dots, r$ ，利用重新初始化的状态向量和协方差矩阵得到的估计为：

$$\begin{aligned} \hat{x}_{k-1|k-1}^{(j)} &= E(x_{k-1} | m_k^{(j)}, z_{k-1}) = \sum_{i=1}^r \hat{x}_{k-1|k-1}^{(i)} \mu_{k-1|k-1}^{(i,j)} \\ \hat{P}_{k-1|k-1}^{(j)} &= \sum_{i=1}^r [P_{k-1|k-1}^{(i)} + (\hat{x}_{k-1|k-1}^{(i)} - \hat{x}_{k-1|k-1}^{(j)})(\hat{x}_{k-1|k-1}^{(i)} - \hat{x}_{k-1|k-1}^{(j)})^T] \mu_{k-1|k-1}^{(i,j)} \end{aligned} \quad (4.8)$$

(2) 模型条件滤波

在确定重新初始化的状态和协方差矩阵并得到新的观测 z_k 之后所进行估计更新即模型条件滤波，模型条件滤波通过以下四步完成。

求解状态预测：

$$\begin{aligned} \hat{x}_{k|k-1}^{(i)} &= F_{k-1}^{(i)} \hat{x}_{k-1|k-1}^{(i)} + v_{k-1} \\ P_{k|k-1}^{(i)} &= F_{k-1}^{(i)} \hat{P}_{k-1|k-1}^{(i)} (F_{k-1}^{(i)})^T + Q_{k-1}^{(i)} \end{aligned} \quad (4.9)$$

求解残差以及对应的协方差矩阵：

$$\begin{aligned} y_k^{(i)} &= z_k - H_k^{(i)} \hat{x}_{k|k-1}^{(i)} - w_k^{(i)} \\ S_k^{(i)} &= H_k^{(i)} P_{k|k-1}^{(i)} (H_k^{(i)})^T + R_k^{(i)} \end{aligned} \quad (4.10)$$

则与 k 时刻的模型 $m_k^{(i)}$ 相匹配的似然函数为：

$$\Lambda_k^{(i)} = \frac{1}{\sqrt{2\pi |S_k^{(i)}|}} \exp\left(-\frac{1}{2} (y_k^{(i)})^T (S_k^{(i)})^{-1} y_k^{(i)}\right) \quad (4.11)$$

对于 $i = 1, 2, \dots, r$ ，分别计算滤波增益，状态估计更新以及状态估计更新的误差协方差，即进行滤波更新：

$$\begin{aligned} K_k^{(i)} &= P_{k|k-1}^{(i)} (H_k^{(i)})^T (S_k^{(i)})^{-1} \\ \hat{x}_{k|k}^{(i)} &= \hat{x}_{k|k-1}^{(i)} + K_k^{(i)} y_k^{(i)} \\ P_{k|k}^{(i)} &= P_{k|k-1}^{(i)} - K_k^{(i)} S_k^{(i)} (K_k^{(i)})^T \end{aligned} \quad (4.12)$$

(3) 模型概率更新

对于 $i = 1, 2, \dots, r$ ，计算模型概率：

$$\mu_k^{(i)} = \frac{\Lambda_k^{(i)} \sum_{j=1}^r \pi_{ji} \mu_{k-1}^{(j)}}{\sum_{j=1}^r \Lambda_k^{(j)} \sum_{i=1}^r \pi_{ij} \mu_{k-1}^{(i)}} \quad (4.13)$$

(4) 估计融合

融合各个滤波器的状态估计值和状态估计的误差协方差值，作为对混合系统进行估计的输出：

$$\begin{aligned} \hat{x}_{k|k} &= \sum_{i=1}^r \hat{x}_{k|k}^{(i)} \mu_k^{(i)} \\ P_{k|k} &= \sum_{i=1}^r [P_{k|k}^{(i)} + (\hat{x}_{k|k} - \hat{x}_{k|k}^{(i)}) (\hat{x}_{k|k} - \hat{x}_{k|k}^{(i)})^T] \mu_k^{(i)} \end{aligned} \quad (4.14)$$

4.2 融合 IMU 与 Kinect 的 IMM-KF 滤波框架

4.2.1 机动指数

机动指数（Maneuvering Index）是一种对动态系统的描述参数，机动指数的大小是衡量选择基于单一模型的滤波器还是多模型的滤波器的标准。当该指标越大时，选择多模型滤波器的必要性越大。

机动指数定义为运动不确定性和观测不确定性的比值：

$$\lambda = \frac{\sigma_a \Delta t^2}{\sigma_w} \quad (4.15)$$

其中 σ_v 和 σ_w 是系统的过程噪声和观测噪声， Δt 是传感器的采样周期。

如图 4.1 所示，对于本课题，由于 IMU 和 Kinect 传感器通过采集数据获取位姿的采样周期分别约为 100Hz 和 30Hz。由于 IMU 的采样周期和 Kinect 的采样周期不同，取二者数据同步的周期值，即 2s。

```

average rate: 1.916
           min: 0.043s max: 1.388s std dev: 0.30245s window: 69
average rate: 2.040
           min: 0.043s max: 1.388s std dev: 0.30911s window: 75
average rate: 2.031
           min: 0.043s max: 1.388s std dev: 0.31201s window: 77
average rate: 1.994
           min: 0.043s max: 1.388s std dev: 0.32011s window: 78
average rate: 2.078
           min: 0.043s max: 1.388s std dev: 0.32176s window: 83

```

图 4.1 IMU 和 Kinect 同步后数据的频率

本课题所用系统的运动不确定性为一周期内噪声对系统加速度的影响对位置的影响，即

$$\sigma_{des} = \frac{1}{2} \sigma_a \Delta t^2 \quad (4.16)$$

而观测不确定性则是对位置进行观测的观测噪声的均方根值，对于本课题，该值为：

$$\sigma_x = \sigma_y = \sqrt{3} \quad (4.17)$$

注意在机动指数表达式中省略了系数 1/2，但是这并不影响机动指数作为描述单模型和多模型算法选择标准时的作用。

根据系统的模型，本课题中机器人对应的机动指数为

$$\lambda = \frac{\sigma_a \Delta t^2}{\sigma_w} = \frac{0.3 \times (2)^2}{\sqrt{3}} = 0.6928 \quad (4.18)$$

对于机动指数较小的系统，选择多模型算法不仅会引起不必要的计算量，还会使得多模型滤波器的表现下降。选择单一模型和多模型算法的机动指数的分界值为 0.5，当机动指数低于 0.5 时，单模型滤波算法和多模型算法的表现相近，而当机动指数高于 0.5 时，使用多模型算法的必要性较大。由于本课题中机器人系统的机动指数达到了 0.6928，所以具有使用多模型算法的必要性。同时由于交互多模型算法与其他算法相比的有效性，使用交互多模型算法具有实际的意义。

4.2.2 KF 子模型数量选择

交互多模型算法对多个模型的输出进行加权，从而通过使得当前的系统状态对输出

贡献更多来实现滤波器精度的提升。从直观上看，选择更多数量的模型可以对更多种环境条件进行描述，通过交互多模型算法的模型交互和加权输出来获取更加精确的结果，但是更多的模型需要对系统的运行模式进行更加精确的分类以及针对每一个子类的建模，在实际中可能会导致由于模型不准确引发的滤波器精度下降。另外一方面，即使获得了针对每一种系统运行模型准确的模型，额外的模型还会使得需要初始化的参数数量增多，而不准确的滤波参数也会导致滤波器精度的降低。

通过实验可以证明，对于运动学模型，使用两个模型已经可以较好地利用交互多模型算法的优势，同时有效地控制利用多个模型所带来的额外的计算量。

4.2.3 KF 子模型类型选择

经常使用二阶匀速运动模型（CV），二阶匀速转动模型（CT）和三阶匀加速运动（CA）模型。之前已经讨论过 CV 和 CA 两类模型，以下讨论 CT 模型。

系统的观测方程为：

$$x(k+1) = F(k)x(k) + v(k) \quad (4.19)$$

其中， $x(k)$, $F(k)$ 和 $v(k)$ 分别是状态向量，状态转移矩阵以及扰动状态向量的噪声。

系统的状态为 $X(k) = [x, \dot{x}, y, \dot{y}, \omega]^T$ ，包括 x 和 y 方向的位置 x, y 和速度 \dot{x}, \dot{y} 以及进行转弯时的角速度 ω ，则系统的状态转移矩阵为：

$$F(k) = \begin{bmatrix} 1 & \frac{\sin(\omega\Delta t)}{\omega} & 0 & -\frac{1 - \cos(\omega\Delta t)}{\omega} & 0 \\ 0 & \cos(\omega\Delta t) & 0 & -\sin(\omega\Delta t) & 0 \\ 0 & \frac{1 - \cos(\omega\Delta t)}{\omega} & 1 & \frac{\sin(\omega\Delta t)}{\omega} & 0 \\ 0 & \sin(\omega\Delta t) & 0 & \cos(\omega\Delta t) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.20)$$

为了与 CA 和 CV 两个模型的状态变量进行匹配，希望可以在以上状态转移矩阵中加入系统在 x 和 y 方向上的加速度以及系统在进行转弯过程中的角位移 ψ 。但是容易看出系统在 x 和 y 上的速度和位置之间存在耦合关系，不能自由添加额外的状态变量。

另外，从状态转移矩阵可以看出，当角速度值是常量时，CT 模型才是线性的。由于角速度同时又出现在系统的状态向量中，通常 CT 模型是非线性模型，在使用该模型进行滤波的时候，需要使用扩展卡尔曼滤波算法。

由于与 CA 和 CV 模型相比，CT 模型的状态变量的个数不同而且模型具有非线性，考虑到两个子模型已经可以描述系统在进行模式切换过程中的动态，使用二阶匀速运动

模型和三阶匀加速运动模型作为交互多模型算法的子模型。

4.2.4 KF 子模型建立

使用 CA 和 CV 两个模型作为交互多模型滤波器的两个子模型，具有描述直线运动的位置，速度和加速度以及描述旋转运动的角位移和角速度一共 8 个状态变量的的二阶恒速度模型为：

$$\mathbf{x}_{k+1} = \begin{bmatrix} 1 & \Delta t & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta t & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_k + \mathbf{v}_k \quad (4.21)$$

而三阶恒加速度模型则是：

$$\mathbf{x}_{k+1} = \begin{bmatrix} 1 & \Delta t & \frac{1}{2}\Delta t^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \Delta t & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta t & \frac{1}{2}\Delta t^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_k + \mathbf{v}_k \quad (4.22)$$

对于不同模式的运动，使用二阶恒速度模型和三阶恒加速度模型的区别在于对系统变化跟踪的精度不一。

由于之前的实验结果已经证明了基于卡尔曼滤波的融合 IMU 和 Kinect 的位姿估计的有效性，可以用于观测的观测变量的数目不仅仅是来自于 Kinect 位姿解算系统的位置以及来自于 IMU 位姿解算系统的偏航角，还包括滤波器输出的线速度和角速度，这样一来获取的观测变量为：

$$\mathbf{z} = [x, \dot{x}, y, \dot{y}, \psi, \dot{\psi}]^T \quad (4.23)$$

从而系统的输出矩阵为：

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.24)$$

针对系统的过程协方差，预测协方差和观测协方差，由于系统的状态变量没有发生变化，可以沿用单模型滤波器的参数，但是由于系统的输出矩阵发生了变化，需要对预测协方差矩阵进行重新设计。

假设各个观测变量之间没有相关关系，则针对位置和角位移可以采用之前的测算值，而针对线速度，考虑之前的实验结果，即预测值和实际值之间相差约 0.1m/s，取最差情况，取在 x 和 y 方向上的速度协方差为：

$$\sigma_x^2 = \sigma_y^2 = 0.03 \quad (4.25)$$

而角速度的协方差的最差情况为

$$\sigma_\omega^2 = 0.27 \quad (4.26)$$

则交互多模型滤波器的观测协方差矩阵为：

$$R = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.03 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.03 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.3655 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.27 \end{bmatrix} \quad (4.27)$$

4.2.5 IMM 模型概率向量和状态转移矩阵设计

针对模型概率向量。由于在没有系统模式变化规律的先验知识的条件下难以确定系统是处于哪一种运行状态，对于二阶恒速度模型和三阶恒加速度模型，取二者初始化模型概率相等，即

$$\mu = [0.5, 0.5]^T \quad (4.28)$$

针对状态转移矩阵。状态转移矩阵中的元素值 π_{ij} 表示系统从模型 j 变化到模型 i 的概率，而 π_{ii} 则表示系统状态停留在 i 状态的概率。对于两个模型的情况，只需要确定模型停留在自身的概率，就可以根据概率的基本性质确定同一行的非主对角线元素。以下讨论模型停留在自身概率的确定方法。

考虑使用可以测量的量来间接计算模型停留在自身的概率，系统停留在状态 i 的时间为 τ_i ，停留时间（sojourn time）和状态转移概率具有一一对应关系：

$$\tau_i = \frac{1}{1 - \pi_{ii}} \quad (4.29)$$

则系统模型停留在状态 i 的概率为：

$$\pi_{ii} = 1 - \frac{1}{\tau_i} \quad (4.30)$$

虽然从信息量的角度看，停留时间越长（或者模型停留在自身的概率越大）则观测更新获取的信息量越少，但是考虑到卡尔曼滤波器与交互多模型滤波器的串联关系，即每次交互多模型滤波器获取的观测值具有相对更高的精度，滤波器输出收敛需要的数据量并不大，所以即使系统在某一个模式停留的时间较长，对于滤波输出精度的影响是有限的。

4.3 融合位姿估计实验与结果分析

4.3.1 位姿估计系统的精度评价

在评价位姿解算系统的对于同一数据集的精度以及表现的时候，需要采用定性或者定量的标准以供不同位姿解算算法的比较。

常用的位姿解算解决方案的评价标准包括以下三类。

① 不需要参考真实位姿值的定性方法。将利用位姿解算系统获取的路径投影到通过先验知识获取的地图上，如建筑物的平面图或者室外的地形地貌图，从而进行比较。这种方法依赖于人员的主观认识，对于表观差异不大的两类位姿解算系统不能适用。

② 不需要参考真实位姿值的定量方法。在获取经过优化的位姿数据之后计算内部误差（Intrinsic Errors），如重投影误差以及更一般的 χ^2 误差。但是使用 χ^2 误差不能保证实际中可以使用的路径，而且在不使用任何传感器的时候会导致错误的结果，即误差为 0。

③ 需要真实参考位姿值的定量方法。通过定量比较估计的路径以及参考真实路径，可以克服以上两类比较准则所具有的问题，从而获取可以通用的某种位姿解算系统的精度评价。常用的评价准则包括相对位姿误差（RPE, Relative Pose Error）和绝对路径误差（ATE, Absolute Trajectory Error）两类，其中相对位姿误差适用于评价位姿解算系统漂移的程度，而绝对路径误差则用于评价视觉 SLAM 系统。

本课题将采用相对位姿误差来评价位姿估计系统的精度。

假设位姿解算系统估计的路径与参考真实路径在时间上同步，且传感器在时间上均匀采样，即有以下两组位姿序列：

$$\begin{aligned} P_1, P_2, \dots, P_n &\in \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} \\ Q_1, Q_2, \dots, Q_n &\in \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} \end{aligned} \quad (4.31)$$

在采样周期 Δ 内，定义在第 i 步上的相对位姿误差为：

$$E_i = (Q_i^{-1} Q_{i+\Delta})^{-1} (P_i^{-1} P_{i+\Delta}) \quad (4.32)$$

对于 n 个位姿的情况，可以得到以上这样 $m = n - \Delta$ 个相对位姿误差，则整个路径上的相对位姿误差为：

$$RMSE(E_{1:n}, \Delta) = \sqrt{\frac{1}{m} \sum_{i=1}^m \|trans(E_i)\|^2} \quad (4.33)$$

在实际中不可能保证均匀采样，则 Δ 可能取遍所有值，则：

$$RMSE(E_{1:n}) = \frac{1}{n} \sum_{\Delta=1}^n RMSE(E_{1:n}, \Delta) \quad (4.34)$$

以上计算的计算复杂度随着位姿解算系统运行的时间成指数增大，达到了 $O(n^2)$ ，实际中 Δ 只取固定数目的值。

4.3.2 融合位姿估计实验

采用与第三章控制 TurtleBot 按照一定加速度走闭环相同的实验条件，即使得 TurtleBot 移动机器人按照 0.1m/s^2 加速度按照 1m 为边长的正方形闭环上行走。对于正方形闭环的每一条边，在加速到 0.2m/s 时按照该速度匀速运动，在即将开始走新的一条边的时候停留 5s 。通过设定匀加速运动，匀速运动和静止这三种运动模式，测试交互多模型算法框架下 IMU 和 Kinect 融合位姿估计的精度。

如表 4.1 所示，按照本章之前的设计选取二阶匀速直线运动以及三阶匀加速直线运动两个子模型，同时取 Markov 矩阵和初始化模型概率分别为

$$\pi = \begin{bmatrix} 0.88 & 0.12 \\ 0.25 & 0.75 \end{bmatrix}, \mu = [0.5, 0.5]^T$$

利用同一个包文件进行 8 次实验，实验结果如表 4.1 所示。

表 4.1 将匀速和匀加速模型作为子模型的 IMM-KF 位姿估计误差
(单位: m)

	实验 1	实验 2	实验 3	实验 4
EKF 融合滤波	0.411845	0.336182	0.385219	0.345106
IMM-KF 融合滤波	0.410847	0.335441	0.384838	0.344464
	实验 5	实验 6	实验 7	实验 8
EKF 融合滤波	0.357533	0.783102	0.824309	0.349850
IMM-KF 融合滤波	0.356819	0.782792	0.824080	0.349105

可以看出 IMM-KF 融合滤波算法可以在大多数情况下在每一个采样周期上对融合滤波精度提升约在 0.001m 级别上。

作为对比，修改 IMM-KF 滤波器的设定，进行两组对照实验。

不使用本章之前的设计，选择静止模型以及二阶匀速直线运动模型，作为 IMM-KF 滤波器的子模型，取 Markov 矩阵和初始化模型概率分别为

$$\pi = \begin{bmatrix} 0.80 & 0.20 \\ 0.12 & 0.88 \end{bmatrix}, \mu = [0.5, 0.5]^T$$

利用前一个实验中的包文件进行 8 次实验。

表 4.2 将静止和匀速作为子模型的 IMM-KF 位姿估计误差
(单位: m)

	实验 1	实验 2	实验 3	实验 4
EKF 融合滤波	0.353981	0.385219	0.354064	0.350601
IMM-KF 融合滤波	0.353412	0.384838	0.353902	0.350320
	实验 5	实验 6	实验 7	实验 8
EKF 融合滤波	0.793562	0.777758	0.344792	0.402299
IMM-KF 融合滤波	0.793317	0.777191	0.344639	0.402143

类似的，取静止模型，CV 模型以及 CA 模型作为子模型，取 Markov 矩阵以及初始化模型概率分别为：

$$\pi = \begin{bmatrix} 0.80 & 0.10 & 0.10 \\ 0.06 & 0.88 & 0.06 \\ 0.25 & 0.25 & 0.50 \end{bmatrix}, \mu = [0.25, 0.5, 0.25]^T$$

表 4.3 将静止，匀速以及匀加速作为子模型的 IMM-KF 位姿估计误差
(单位: m)

	实验 1	实验 2	实验 3	实验 4
EKF 融合滤波	0.780217	0.330618	0.351846	0.397647
IMM-KF 融合滤波	0.780076	0.330573	0.351511	0.397079

	实验 5	实验 6	实验 7	实验 8
EKF 融合滤波	0.789621	0.344815	0.762756	0.789727
IMM-KF 融合滤波	0.789436	0.344724	0.762078	0.789597

与匀速模型和匀加速模型两种子模型选择相比，这两种设定在单模型滤波器的基础上虽然也有提升，但是在每一个采样周期上精度的提升 0.0001m 级别上，精度提升较少。

4.3.3 实验结果分析

通过引入 IMM-KF 算法使得单传感器位姿解算在每个采样周期上都有提升，如图 4.2 所示，由于机器人在机器人走闭环的过程中采样周期大约为 1/30s，则机器人在运动的过程中，每秒的位姿漂移可以被纠正约 0.03m，如果假设机器人的位置在 x 和 y 方向上的误差相等，则在 x 和 y 方向上的定位漂移可以被纠正 0.021m。

```

average rate: 30.004
  min: 0.000s max: 0.044s std dev: 0.01030s window: 510
average rate: 29.972
  min: 0.000s max: 0.044s std dev: 0.01045s window: 540
average rate: 29.952
  min: 0.000s max: 0.044s std dev: 0.01059s window: 569
average rate: 30.004
  min: 0.000s max: 0.044s std dev: 0.01081s window: 600
average rate: 29.976
  min: 0.000s max: 0.044s std dev: 0.01091s window: 630
average rate: 29.959
  min: 0.000s max: 0.044s std dev: 0.01100s window: 659

```

图 4.2 IMM-KF 位姿估计结果输出周期

4.4 本章小结

本章主要介绍了交互多模型算法的原理和针对本课题实验平台的 IMM-KF 算法设计细节。利用 IMM-KF 算法，机器人在运动过程中，位姿解算系统将根据传感器数据自适应地选择最符合当前运动模式的滤波模型，从而优化由于运动模式切换而模型单一带来的误差而实现更高精度的位姿解算。通过实现基于 IMM-KF 的融合 IMU 与 Kinect 位姿解算系统，并设定机器人运动状态不断切换的实验场景，利用实验证明多模型算法在处理运动模式切换的系统中状态估计问题的有效性。

第五章 总结与展望

5.1 总结

本文针对基于 IMU 的位姿解算精度受到加速度漂移的影响严重以及基于 Kinect 的位姿解算精度受到环境中特征缺乏以及易受光照影响这两类问题开展研究, 提出将 IMU 的姿态数据以及利用 Kinect 获取的位置数据进行融合的方法, 从而克服了 IMU 和 Kinect 的缺陷, 达到这两类传感器缺点相抵, 优势互补的目的。本课题所进行的研究工作可以概括为以下三点:

(1) 阐述了 IMU 和 Kinect 位姿解算的原理和各自存在的问题, 通过实现 IMU 和 Kinect 的单传感器位姿解算系统, 利用实验证明仅使用 Kinect 或者 IMU 的单传感器系统在进行位姿解算过程中的不可靠性。

(2) 阐述了融合 IMU 和 Kinect 的卡尔曼滤波器的原理, 在卡尔曼滤波器框架下设计了用于融合 IMU 和 Kinect 位姿的线性卡尔曼滤波器和扩展卡尔曼滤波器, 并通过实验得出融合了 IMU 和 Kinect 的卡尔曼滤波器可以在总体上保证融合位姿估计的正确性, 可以单传感器位姿解算系统输出位姿进行纠正的结论。

(3) 阐述了交互多模型算法的原理和针对本课题实验平台的 IMM-KF 算法设计细节, 通过实现基于 IMM-KF 的融合 IMU 与 Kinect 位姿解算系统, 并设定机器人运动状态不断切换的实验场景, 利用实验证明多模型算法在处理运动模式切换的系统中状态估计问题的有效性。

5.2 展望

融合 IMU 和 Kinect 的位姿估计作为实现自主移动机器人主要问题之一, 属于当前对机器人科学研究的热点问题。本课题提出的方法从提升移动机器人位姿估计的精度出发, 对基于 IMU 与 Kinect 的单传感器系统在卡尔曼滤波的框架下进行融合, 之后又基于交互多模型算法使用多个模型描述移动机器人在运动过程中的运动模式切换, 在卡尔曼滤波框架下进一步提升了位姿估计的精度。虽然本课题的工作可以对位姿估计精度进行一定的提升, 但是仍然存在一些不足之处, 需要在未来的工作中对这些问题进行解决。

(1) 在单模型融合滤波算法的选择上, 由于模型潜在的非线性, 可以考虑使用无迹卡尔曼滤波 (UKF, Unscented Kalman Filter), 无迹卡尔曼滤波通过进行无迹变换并

选取 Sigma 点，将非线性模型变换为线性模型。与扩展卡尔曼滤波算法相比，由于无迹卡尔曼滤波采用了不同的策略，变换后的模型近似度更高。无迹卡尔曼滤波由于提出时间较短还没有达到扩展卡尔曼滤波的广泛的接受程度，在传感器数据融合问题中也使用频率较低。因此有必要将这一较新的方法应用到 IMU 与 Kinect 融合位姿估计的研究中来。

(2) IMM-KF 滤波器的 Markov 状态转移矩阵选择受到较大限制。由于本课题基于停留时间选择 Markov 矩阵中的主对角线元素，需要知道移动机器人在各个模式中运动的时间长短，当实验无法重复或者移动机器人处于未知环境的时候各个运动模式的停留时间无法得到，从而就不能使用本课题中的方法设计 Markov 矩阵。除此以外，Markov 矩阵中的元素一经确定就不能修改，当机器人所处的环境变化剧烈或者机器人实现自主运动，运动模式无规律切换的时候，Markov 矩阵中的元素由于无法修改，将导致滤波输出误差。因此，一方面应该考虑另外的方法确定 Markov 矩阵中的元素，另一方面应该实现 Markov 矩阵中元素的自适应变化。

参考文献

- [1] 王栋梁. 结构化环境中移动机器人位姿估计研究[D].武汉科技大学,2010.
- [2] Sturm J, Engelhard N, Endres F, et al. A benchmark for the evaluation of RGB-D SLAM systems[C]//2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2012: 573-580.
- [3] Fabian J, Clayton G M. Error analysis for visual odometry on indoor, wheeled mobile robots with 3-d sensors[J]. IEEE/ASME Transactions on Mechatronics, 2014, 19(6): 1896-1906.
- [4] Fang Z, Scherer S. Real-time onboard 6dof localization of an indoor mav in degraded visual environments using a rgb-d camera[C]//2015 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2015: 5253-5259.
- [5] Fang Z, Scherer S. Experimental study of odometry estimation methods using RGB-D cameras[C]//2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014). IEEE, 2014: 680-687.
- [6] Konolige K, Agrawal M. Large-scale visual odometry for rough terrain[M]//Robotics research. Springer Berlin Heidelberg, 2010: 201-212.
- [7] Henry P, Krainin M, Herbst E, et al. RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments[J]. The International Journal of Robotics Research, 2012, 31(5): 647-663.
- [8] Loianno G, Thomas J, Kumar V. Cooperative localization and mapping of MAVs using RGB-D sensors[C]//2015 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2015: 4021-4028.
- [9] Lu Y, Song D. Robustness to lighting variations: An RGB-D indoor visual odometry using line segments[C]//2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2015: 688-694.
- [10] Chow J C K, Lichti D D, Hol J D, et al. Imu and multiple RGB-D camera fusion for assisting indoor stop-and-go 3D terrestrial laser scanning[J]. Robotics, 2014, 3(3): 247-280.
- [11] Santos M C P, Sarcinelli-Filho M, Carelli R. Indoor waypoint UAV navigation using a RGB-D system[C]//2015 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS). IEEE, 2015: 84-91.
- [12] Santos M C P, Santana L V, Martins M M, et al. Estimating and controlling uav position using rgb-d/imu data fusion with decentralized information/kalman filter[C]//2015 IEEE International Conference on Industrial Technology (ICIT). IEEE, 2015: 232-239.
- [13] Leutenegger S, Lynen S, Bosse M, et al. Keyframe-based visual-inertial odometry using

- nonlinear optimization[J]. The International Journal of Robotics Research, 2015, 34(3): 314-334.
- [14] Brunetto N, Salti S, Fioraio N, et al. Fusion of inertial and visual measurements for rgb-d slam on mobile devices[C]//Proceedings of the IEEE International Conference on Computer Vision Workshops. 2015: 1-9.
- [15] Qayyum U, Kim J. Inertial-kinect fusion for outdoor 3d navigation[C]//Australasian Conference on Robotics and Automation (ACRA). 2013.
- [16] Mostofi N, Elhabiby M, El-Sheimy N. Indoor localization and mapping using camera and inertial measurement unit (IMU)[C]//2014 IEEE/ION Position, Location and Navigation Symposium-PLANS 2014. IEEE, 2014: 1329-1335.
- [17] Omari S, Bloesch M, Gohl P, et al. Dense visual-inertial navigation system for mobile robots[C]//IEEE International Conference on Robotics and Automation (ICRA), 2015. IEEE, 2015: 2634-2640.
- [18] Mourikis A I, Roumeliotis S I. A multi-state constraint Kalman filter for vision-aided inertial navigation[C]//IEEE international conference on Robotics and Automation (ICRA), 2007. IEEE, 2007: 3565-3572.
- [19] Hesch J A, Kottas D G, Bowman S L, et al. Camera-IMU-based localization: Observability analysis and consistency improvement[J]. The International Journal of Robotics Research, 2014, 33(1): 182-201.
- [20] Blom H A P. An efficient filter for abruptly changing systems[C]//The 23rd IEEE Conference on Decision and Control, 1984. IEEE, 1984, 23: 656-658.
- [21] Kong J, Pfeiffer M, Schildbach G, et al. Kinematic and dynamic vehicle models for autonomous driving control design[C]//IEEE Intelligent Vehicles Symposium (IV), 2015. IEEE, 2015: 1094-1099.
- [22] Kirubarajan T, Bar-Shalom Y. Kalman filter versus IMM estimator: when do we need the latter?[J]. IEEE Transactions on Aerospace and Electronic Systems, 2003, 39(4): 1452-1457.
- [23] 张云洲, 付文艳, 项姝, 等. 室内环境下基于 IMM-EKF 算法的移动目标定位[J]. 计算机研究与发展, 2014, 51(11): 2408-2415.
- [24] Yu Y K, Wong K H, Chang M M Y. A fast and robust simultaneous pose tracking and structure recovery algorithm for augmented reality applications[C]//2004 International Conference on Image Processing, 2004. ICIP'04. IEEE, 2004, 2: 1029-1032.
- [25] Bloomer L, Gray J E. Are more models better?: The effect of the model transition matrix on the IMM filter[C]//Proceedings of the Thirty-Fourth Southeastern Symposium on System Theory, 2002. . IEEE, 2002: 20-25.
- [26] Simeonova I, Semerdjiev T. About the specifics of the IMM algorithm design[C]//International Conference on Numerical Methods and Applications. Springer Berlin

Heidelberg, 2002: 333-341.

[27] Mohammed D, Mokhtar K, Abdelaziz O, et al. A new IMM algorithm using fixed coefficients filters (fastIMM)[J]. AEU-International Journal of Electronics and Communications, 2010, 64(12): 1123-1127.

[28] Lee B J, Park J B, Lee H J, et al. Fuzzy-logic-based IMM algorithm for tracking a manoeuvring target[J]. IEE Proceedings-Radar, Sonar and Navigation, 2005, 152(1): 16-22.

致谢`

下午 4 点 41 分，140 多个日夜的工作浮现于此时的眼际。经过超过二十周的持续工作，十余次方案的试错，论文的四次修改以及两次预答辩，终于完成了本科毕业设计的论文。这篇论文标志着的不仅仅是这 1400 多个小时知识，灵感甚至体力的碰撞交融，更标志着四年青春的强音。

与其高呼这样或者那样的宣言，还不如真实地踏出第一步。当刚从张云洲老师那里接收到这个毕设题目，我的内心是拒绝的。视觉和惯导都是那么遥远的词汇，机器人也似乎仅仅在展览会上看到过。但是，在张云洲老师的正确引导下，我从大量阅读国内外文献开始，终于在一个月的努力之后明确了自己的方向。

与其记挂着模糊的诗和远方，还不如静静地坐在计算机的身旁。当机器学习，深度学习各种看似高级的名词小虫般地在耳边振翅，似乎任何一个惊异于人类智能之伟大的个体似乎都不能抑制住内心的激动，想尽早地达到顶峰。但是，还是从最基本的做起吧。在张云洲老师的鞭策下，我从学习 ROS 框架开始，为之后算法的实现打下了坚实的基础，终于，我不仅实现了单模型滤波的 EKF 算法，还使得了多模型的 IMM 算法成功运行。

在完成毕业设计的过程中，我受到了实验室内外各种无私的建议和帮助。在其中首先要感谢的是张云洲老师，他对待工作的热情，对待学生的认真深深地感染并影响了我。其次，还需要感谢矿电楼 103 的所有师兄和师姐们，是他们让我了解到了安静，专注和持久的内涵。最后，要感谢提出线性系统最优估计器的 Rudolf Kalman，提出交互多模型算法的 Henk Blom 以及使得这些算法通俗易懂的 Roger R Labbe Jr，正是因为他们的工作，使得我的毕业设计成为可能。

在我的前 22 年生命中，我要感谢我的父母以及将我送到东北大学的母校——西北师范大学附属中学。我的父母给了我适宜的成长条件，而我的母校则使得我能够度过这 22 年生命中繁荣生长的四年。

坚信“卓犖全凭弱冠争”，让“勤慎诚勇”贯穿着“自强不息，知行合一”，未来的自己要 Let There Be Light。

杨冠群

2017 年 6 月 11 日