

# 中国科学技术大学

# 本科毕业论文



## 高带宽轮式激光惯性里程计

作者姓名： 张昊鹏

学 号： PB21050991

专 业： 信息安全

导 师： 张世武教授

完成时间： 2025 年 6 月 10 日



## 摘要

本文介绍了 Point-LIWO (Batch based Direct Point Lidar-IMU-Wheeled Speed Odometry): 一种高效、鲁棒、通用的基于轮式移动机器人平台的增强型雷达-惯性-轮速里程计框架。Point-LIWO 基于高效的紧耦合扩展卡尔曼滤波器实现，具有三个关键创新点：引入点云批处理思想、引入轮速及其数据源协方差度量方式、实现了实时的嵌入式端轮速观测器。其能够在激光雷达退化、IMU 退化或轮速退化等极端场景下实现快速、稳健、精确的机器人状态估计。

第一项创新是基于 Batch-Wise 的批内运动传播和基于 Batch Points 的残差构建。这种方法充分利用了非重复扫描雷达的大视场角特性，通过在一帧内利用分布于空间的多个平面显著降低平均测量误差，并提高处理速度。第二项创新是将轮速作为观测源纳入紧耦合扩展卡尔曼滤波器，并提出了一种有效的协方差度量方法。这显著增强了在雷达和 IMU 退化场景下的里程计鲁棒性，并确保在打滑、滞空等轮速退化环境中的系统稳定性。第三项创新是为轮式移动机器人平台设计了可在嵌入式平台实时运行的观测器。

本文在 2WIS-4WID (二轮独立转向、四轮独立驱动) 底盘上基于 Livox MID360 激光雷达进行了测试，并与最先进的激光惯性里程计进行了比较。结果表明，Point-LIWO 以更低的计算负荷，实现了更高的姿态估计带宽和建图精度。总体而言，Point-LIWO 具有高计算效率（在 20m×40m 大场景下，intel 1240P CPU，单线程点云数为 80,000 pts/s，占用 50% 线程时间）、强鲁棒性（在 15 rad/s、4.5 倍重力加速度下的稳定性，以及在极端退化场景下的可靠性）、高有效输出带宽（可选输出频率在 10 Hz 至 1000 Hz 之间）以及低耦合性（将轮速作为可选观测源纳入融合框架）等优点。

**关键词：**激光惯性里程计；紧耦合里程计；鲁棒姿态估计；移动机器人；退化场  
景建图

## ABSTRACT

This paper presents Point-LIWO, an efficient, robust, and versatile enhanced lidar-inertial-wheel speed odometry framework based on a wheeled mobile robot platform. Point-LIWO is built on an efficient tightly-coupled extended Kalman filter and features three key innovations that enable rapid, robust, and accurate robot state estimation in extreme scenarios such as lidar degradation, IMU degradation, and wheel speed degradation.

The first innovation is the Batch-Wise motion propagation and residual construction based on batch points within the batch. This method fully exploits the large field of view characteristics of non-repetitive scanning lidar, significantly reducing average error and improving processing speed by utilizing multiple planes distributed throughout space within a single frame. The second innovation incorporates wheel speed as an observation source into the tightly-coupled extended Kalman filter, providing an effective covariance measurement method. This significantly enhances the odometry robustness in scenarios of lidar and IMU degradation while ensuring system stability in conditions of wheel speed degradation, such as slipping and off ground. The third innovation is the design of an observer capable of real-time operation on an embedded platform for wheeled mobile robots.

The framework was tested on a 2WIS-4WID (two-wheel independent steering, four-wheel independent drive) chassis using the Livox MID360 lidar and compared with state-of-the-art lidar-inertial odometry systems. Results demonstrate that Point-LIWO consistently achieves higher pose estimation bandwidth and mapping accuracy with lower computational load. Overall, Point-LIWO boasts high computational efficiency (processing up to 80,000 pts/s in a large scene of 20m×40m with 50% time utilization), strong robustness (stability under 15 rad/s rotation, 45  $m/s^2$  inertial acceleration, and in extremely degraded scenarios), high effective output bandwidth (output frequency selectable between 10 Hz and 1000 Hz), and low coupling (with wheel speed incorporated as an optional observation source in the fusion framework).

---

**Key Words:** Lidar-Inertial Odometry; Tightly-Coupled Odometry; Robust Pose Estimation; Mobile Robot; Degraded Scenario Mapping



## 目 录

第一章 绪论 .....	2
第一节 简介 .....	2
第二节 相关工作 .....	4
一、3D LiDAR 里程计与地图构建 .....	4
二、运动畸变补偿 .....	5
三、IMU 数据的融合 .....	6
第二章 系统概述 .....	7
第三章 方法 .....	9
一、符号约定 .....	9
二、运动学模型 .....	10
三、扩展卡尔曼滤波器 .....	13
四、嵌入式底盘观测器 .....	18
第四章 测试与实验 .....	24
一、实验平台介绍 .....	24
二、实时建图性能展示 .....	25
第五章 研究总结 .....	33
参考文献 .....	34
致谢 .....	37

# 第一章 绪论

## 第一节 简介

对于自主移动机器人而言，一个非常重要的课题是如何实时构建未知环境的地图并基于该地图进行定位（SLAM）与实时机器人运动状态估计。稳定鲁棒的运动状态估计为机器人控制器提供实时可靠的状态反馈，实时构建的高精度去畸变稠密地图为后端的轨迹规划器提供必要的环境信息。

视觉 SLAM 算法在常规室内环境中通常能够取得较好的定位效果，但受限于环境的特征丰富度和图像质量，视觉 SLAM 难以应用于运动剧烈或空旷的任务场景。同时，实时处理高分辨率图像对边缘设备的算力来说是巨大的挑战。与相机二维成像的特点相比，LIDAR 能够提供直接，密集，准确，鲁棒的深度数据。这些优势让 LIDAR 在过去的十几年内在移动机器人场景中发挥着越来越重要的作用。几个很好的例子便是自动驾驶汽车和无人机。LIDAR 在 SLAM 场景中日益广泛的应用吸引了很多厂商生产面向不同场景的激光雷达。其中一个重点便是研发应用在移动机器人上的成本，尺寸，重量，功率都显著降低并且有着大 FOV 的激光雷达产品，比如 Livox MID360 和 RoboSense Airy。它们每秒能分别输出 20w 和 80w 的激光测量点，且都具有 360 度的水平 FOV，这使得他们非常适合用于开发以移动机器人为中心的感知和规划算法。

以 MID360 激光雷达为例，它是非重复扫描激光雷达，传感器可以等效为 4 个点光源在半球面上高速旋转并采样周围的深度信息。它的深度采样在时间上顺序发生，连续，并且短时间内可以采样大范围环境。但是目前绝大多数的基于 LIDAR 的 LIO 方法忽略了这些信息。它们采用类似视觉方法的帧结构组织点云，点云被打包并以特定的帧率（比如 10Hz）定期处理，通过 IMU 进行帧内的去畸变和位姿传递。这样的处理引入了难以去除的帧内畸变，对里程计的精度和测绘地图的精度造成了不利影响。同时，低帧率还会增大里程计的延迟并根据内奎斯特采样定律从原理上限制了里程计的带宽上限。

针对这些问题，HKU MARS 在 23 年的工作 Point LIO<sup>[1]</sup> 中给出了很好的解决思路。Point LIO 基于紧耦合的流形上的扩展卡尔曼滤波器实现，基于测量点进行 Point Wise（逐测量点）的观测器更新。基于雷达点的采样时间顺序参与融合，从原理上最小化了帧内运动失真对 LIO 系统的影响。从易于理解的角度来

说, Point LIO 方法将激光雷达等效为了每秒产生 20w 个深度采样的传感器。每个雷达点都是这个传感器的一个采样。同时为了适配这样的思路, Point-LIO 将 IMU 的角速度和加速度都建模到了观测器的状态量之中, 通过 LIDAR Point 和 IMU 两个数据来源对观测器维护的角速度和加速度状态量进行更新。这使其在 IMU 饱和时也可以平滑地估计系统状态, 有效提高了系统整体带宽。

然而, Point Wise 的更新策略存在局限性。首先: 它无法将系统并行化处理; 其次: 因为每个测量点都需要对状态量进行时间对齐和更新, 系统计算量显著增大。这两个问题导致系统必须要对输入的点云进行大规模降采样处理, 以保证系统保持实时运行。接着: 因为基于 ICP 的局部残差构建方式, 用于更新的激光点的选取对系统的表现和稳定性非常重要。这使得 Point LIO 系统对带噪声甚至有偏的雷达测量非常敏感。最后: 因为逐点产生残差, Point LIO 系统在 LIDAR 退化场景下的稳定性不佳。

本文在 Point LIO 工作的启发下提出了 Point LIWO 框架。首先, 本文基于 Batch Wise (逐雷达点批) 的思想将一段时间内的点云 (通常 1ms 为基准值) 集合到一起组成一个雷达点 Batch, 然后根据系统状态量中的角速度和速度进行位姿传播并对 Batch 内的点云进行去畸变。去畸变后的点云在实时构建的点云地图中匹配临近点构造残差, 将 Batch 中所有点的残差汇总并更新卡尔曼滤波器。首先, 这样的系统架构在保证系统带宽的同时减少了状态量更新次数, 降低了系统的计算量。其次, 因为 Batch 中的点分布于空间中的多个平面上, 所以测量的平均噪声大幅下降, 奇异测量点对系统的影响减小。接着, 基于 Batch 的残差计算方案可以实现 Batch 内的并行最近点查找和残差生成, 这将有效提高系统在多线程嵌入式设备上的运行表现。然后, 本文添加了轮速作为 LIDAR 和 IMU 之外的新测量来源, 基于流型扩展卡尔曼滤波框架进行融合并且给出了基于 RMS (残差平方和) 的轮速协方差描述方案。这么做的优点主要在于以下几个方面: 1. 轮速作为一阶速度来源很好弥补了加速度计二阶积分容易漂移的问题; 2. 轮速的融合将有效提高里程计在通过狭窄隧道、剧烈撞击等极端场景下的稳定性; 3. 本文对于协方差的描述使得系统能够在面对打滑等轮速发生严重偏移的退化场景时自动降低轮速在更新过程中的置信度, 确保系统的稳定性。最后, 本文设计了移动机器人的底盘观测器框架, 目的是在嵌入式平台上仅轮速的数据输入下实现鲁棒的底盘速度观测、残差估计、主动轮力矩分配和主动功率控制。最大限度避免轮打滑, 同时提供度量轮打滑程度的方法。这些改进使得 Point-LIWO 系统能够在大规模场景下长时间稳定工作并且为后端规划算法提供稳定的运行平台。

---

## 第二节 相关工作

### 一、3D LiDAR 里程计与地图构建

近年来，许多关于 3D LiDAR 里程计和地图构建的研究都基于 LiDAR-里程计与地图构建（LOAM LIDAR Odometry and Mapping in Real-time）框架<sup>[2]</sup>。在这个框架中，原始 LiDAR 点云被积累成一个帧（也称为扫描），算法从中提取特征点（例如边缘和平面）。这些提取的特征点随后与前一帧进行匹配，以帧扫描频率（例如 10Hz，与激光雷达本身实现有关）的里程计输出，并将几帧扫描累积成一个小的子地图，最终以较低的频率（例如 1 Hz）与全局地图进行匹配和合并，以精确定位 LiDAR 的位置。LOAM 中扫描与扫描、扫描与地图之间的分离结构被许多后续工作所采用，例如 Lego-LOAM<sup>[3]</sup>，它通过考虑地面约束来提高扫描匹配的精度。LINS<sup>[4]</sup>，它将 IMU 数据与扫描匹配结合。其他如参考文献<sup>[5-6]</sup>，则专注于提高计算效率或精度。虽然 LOAM 中扫描与扫描、扫描与地图的分离结构能够显著减轻里程计所需的计算负担，但扫描到扫描的注册往往很容易导致快速的漂移累积。此外，扫描到扫描的注册需要连续扫描之间存在大幅视角重叠，但在小视场（FoV）的固态 LiDAR 中，这种重叠无法实现。为此，LOAM-Livox<sup>[7]</sup>将 LOAM 应用于新兴的固态激光雷达，通过直接将新扫描注册到全局地图来获得里程计。

为了解决这些问题，直接的扫描到地图（或扫描到局部地图）方法被广泛采用。例如基于点地图<sup>[8-9]</sup>、基于 ICP||NDT<sup>[10]</sup>、Surfel 地图<sup>[11]</sup>或体素地图<sup>[12]</sup>的方法。特别是参考文献<sup>[7]</sup>提出了一种并行的扫描到地图方法，来处理固态 LiDAR 小视场的问题。参考文献<sup>[13]</sup>在一个高效的迭代卡尔曼滤波框架中融合了 IMU 数据以优化扫描到地图的匹配。基于扫描到地图方法的框架中的一个关键问题是如何保持地图结构，以便能够增量地添加新的扫描点并且实现高效查询。为了解决这个问题，参考文献<sup>[14]</sup>提出了一种增量式 k-d 树（ikd-Tree）作为地图结构。得益于这一高效的增量式地图结构，FAST-LIO2 系统能够在低功耗的 ARM 计算机上，实时执行机械式旋转 LiDAR 的 10 Hz 里程计和地图构建，以及固态 LiDAR 的 100 Hz 里程计和地图构建。

扫描到地图注册的一个主要缺点是里程计的估计是在扫描频率下进行的，这限制了里程计的输出频率。这个有限的输出频率会导致与扫描时长相等的里程计延迟。此外，有限的状态估计频率会因为奈奎斯特采样定理给里程计造成一个带宽上限。Lola-SLAM<sup>[15]</sup>部分解决了这个问题，它提出将扫描划分为多个子扫

描，并在接收到每个子扫描后立即注册到地图，从而实现了最高 160 Hz 的里程计。Point LIO<sup>[1]</sup>给出了一个点到地图框架，这个框架在接收到每个单独的点时立即将其注册到地图，所以这个框架理论上可以实现以点采样频率输出的里程计，测试结果来看大概 4-8kHz，它显著提高了里程计的带宽但是对点云进行了过度的降采样，导致了很大的高斯噪声和不稳定性。

本文的方法提出了一个 Batch Points 到地图的框架，这个框架在收到一个 Batch 的点后根据自身运动估计对 Batch 内的点进行去畸变，然后一起注册到地图之中。本文在实现时选取 1ms 作为 Batch 的标准时间长度，可以实现 1000hz 的稳定里程计输出，同时显著提高了点云吞吐量（8w pts/s 约 100 倍于 Point LIO），降低了平均误差，同时因为多平面共同构造残差，有效降低了异常点和平面法向拟合不准确对系统的影响。显著提高了系统的鲁棒性和精度。

## 二、运动畸变补偿

如上所述，现有的 LIO 工作几乎都基于扫描（即基于帧的），扫描中由于 LiDAR 本身在帧内持续运动而产生的运动畸变是无法避免的。为了修正这种畸变，通常需要使用补偿方法。在大多数方法中，LIDAR 本体假设在扫描过程中是恒速的，例如<sup>[3-4]</sup>。恒速运动假设在扫描时长较短或运动较为平缓时是有效的。然而，对于非常剧烈的运动过程（例如无人机特技飞行，移动机器人发生碰撞，地面机器人快速旋转等极端场景），帧内速度可能会发生很大变化，这时恒速方法往往会导致大的里程计和建图漂移甚至里程计失效。另一种常见的运动补偿方法是基于连续时间进行轨迹优化，例如基于 B 样条<sup>[16-18]</sup>和高斯过程模型<sup>[19-21]</sup>的方法。连续时间轨迹允许在任何时刻评估位姿，因此能够补偿每个单独点的畸变。然而，连续时间轨迹优化非常耗时，通常是离线进行的。也有一些在线计算的实现，但里程计的速率通常较低（例如 10 Hz），难以保证里程计的实时性。此外，这些方法通常需要累积足够的点数来进行可靠的轨迹参数优化，这也显著增大了的里程计延迟。连续时间轨迹如 BspLine（B 样条曲线）固有的平滑性也很大程度阻止了其描述机器人经历的高动态运动、大加速度运动。

利用 IMU 的测量数据进行运动补偿是另一种有效的运动补偿方法<sup>[14,22]</sup>。这些方法通过在帧内融合 IMU 数据，结合 LiDAR 位姿来消除点云的畸变。由于 IMU 测量频率较高（例如 200 Hz），基于 IMU 的运动补偿对于常见的机器人运动、翻滚的无人机机动非常有效。然而，这种方法仍然受到 IMU 频率的限制，并且容易受到 IMU 测量噪声和偏差估计误差的影响。上述的运动补偿方法最终都

基于现有方法中的基于帧的里程计框架实现。在 Point-LIO<sup>[1]</sup> 中，雷达点在真实采样时间点进行融合，消除了运动畸变。本文的工作延伸了 Point LIO 的内容，结合小 Batch 中运动畸变不明显的特点根据角速度和速度进行 Batch 内的去畸变。并且和 IMU 数据同步时间并进行融合，同时因为更小的估计误差和更好的姿态估计，总体上而言达到了更好的去畸变效果并且输出了更大量的去畸变点云。

### 三、IMU 数据的融合

在 LIO 框架中融合 IMU 测量数据通常采用两种主流方法：松耦合方法和紧耦合方法。松耦合方法将 IMU 测量数据整合，以获得位姿先验估计，并将该估计作为后续点云注册的初始位姿<sup>[2]</sup>。紧耦合方法将 IMU 测量数据和 LiDAR 点云融合到一个联合优化中。紧耦合方法目前已经提出两种可行的实现方式：基于扩展卡尔曼滤波（EKF）的方法<sup>[4,14,23-24]</sup>和基于优化的方法<sup>[5]</sup>。基于 EKF 的方法通过扩展卡尔曼滤波的传播过程融合 IMU 测量数据，获得位姿估计，随后与 LiDAR 测量数据在 EKF 更新步骤中融合<sup>[14]</sup>。与之不同，基于优化的方法则预先积分 IMU 测量数据，获得相对位姿约束，然后将这些预积分的相对位姿约束与点云注册误差融合<sup>[5]</sup>。通常，紧耦合方法比松耦合方法具有更高的鲁棒性和精度。然而，在所有这些紧耦合方法中，IMU 数据通常作为模型的输入，在 EKF 传播或预积分过程中只能为一个帧的时长提供更新<sup>[25]</sup>。基于这样的 EKF 传播或预积分方法，如果机器人运动超出了 IMU 测量范围（IMU 饱和），将会发生严重问题。Point LIO 系统基于 EKF 方式实现了紧耦合的 IMU 数据融合，同时将 IMU 角速度写进 EKF 状态量之中。在 LiDAR 点更新时将对 IMU 数据也进行更新。这样的系统可以在 IMU 饱和测量情况下保证里程计的稳定性。但是其无法长时间应对饱和问题，因为根据雷达传播出的角速度的难以满足稳定性的要求，这在 Point-LIO 工作的饱和测试场景中可以看出。所以本文将轮速作为可选项加入了基于 EKF 的紧耦合框架之中，通过 EKF 的传播将状态量对齐到 IMU/轮速/雷达点的时间戳并进行更新。以很高的计算效率实现了稳定且高带宽的位姿估计效果。

## 第二章 系统概述

系统框架如图2.1所示。本框架把 Lidar、IMU、轮速都认为是带噪声的，在特定时刻发生的测量。在拿到一段时间的传感器数据后，本框架将 Lidar 点按照时间进行打包形成 Lidar Batch，时间戳为 Batch 中最后一个采样点的时间戳。然后将三种传感器数据（Lidar Batch、IMU、轮速）按照时间升序排列形成按照时间升序的测量量队列，通过状态量传播来同步状态量和测量量之间的时间差。即在一次更新过程中，对于测量队列的头元素，首先将状态量预积分到该元素对应的时间戳。同步时间之后，基于状态量和测量量根据模型计算残差，并基于残差扩展卡尔曼滤波框架进行状态更新。

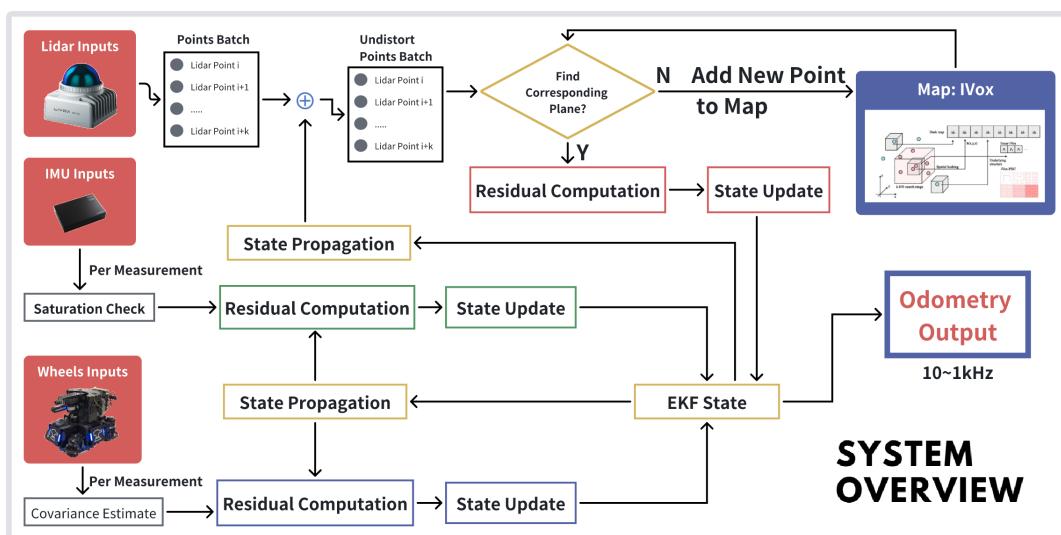


图 2.1 LIWO 系统框架展示

三种观测源有各自的更新方程。对于每个 IMU 测量，IMU 有 6 个通道数据 (ACC 3D + GYRO 3D)。框架会对每个通道单独进行饱和性检查，如果检测到饱和则通过调整更新方程关闭对应数据通道的数据融合。对于每个轮速测量，它有三个有效通道 ( $V_x + V_y + \omega_z$ )。每个轮速测量先被投影到 IMU 坐标系下，然后基于轮速的残差平方和 (三维速度投影到四个轮子上后计算平方和) 度量轮速的协方差。同时框架可选在 IMU 坐标系下关闭来自轮速的角速度信息融合来保证角速度的时间一致性和 IMU 时间同步。在这之后，系统会根据 IMU 坐标系下的残差和协方差构造更新方程和测量噪声矩阵。对于 Lidar Batch 来说，Batch 中的

---

每个雷达点都有各自的时间戳。对于每一个雷达点，框架首先计算其和 Batch 时间戳（最后一个雷达点时间戳）之间的时间差，然后根据状态量的速度和角速度基于匀速假设在时间差尺度下进行雷达点的位置传播。然后对 Batch 中的每一个点，将点变换到世界坐标系下，然后匹配临近地图点并拟合平面，使用流形卡尔曼滤波器计算残差并汇总残差以更新系统状态。如果在搜索半径内没有找到足够的临近点，则直接将点注册到地图之中。

快速的点云注册和快速的范围内最临近查找对系统实时性非常重要，为了提高系统的实时性，在经过与 ikd-tree 的对比之后，本文最终使用了 iVOX<sup>[26]</sup>(incremental voxels) 做为地图表示方式。在不考虑构建体素内稠密地图的情况下，iVOX 近似于哈希表，在控制临近搜索范围之后能在  $O(N)$  复杂度之内近似得到欧几里得距离阈值之内的 N 近邻点（SLAM 算法其实并不关心找到真正的 N 近邻，只关心能不能在一定距离阈值之内找到近邻并排序得到最近几个）。

## 第三章 方法

Point-LIWO 的主干是一个基于流型的紧耦合扩展卡尔曼滤波器。这部分实现主要基于工作 (IKFOM: 流形上迭代误差状态扩展卡尔曼滤波器的符号表示和工具包)<sup>[27]</sup> 和它所提供的开源工具包实现。

### 一、符号约定

表 3.1 状态估计的符号定义

符号	含义
$\mathbf{x}_k$	在第 $k$ 次测量采样时间的状态量 $\mathbf{x}$
$\mathbf{x}$	状态量的真值 $\mathbf{x}$ .
$\hat{\mathbf{x}}, \bar{\mathbf{x}}$	传播后和更新后的状态量 $\mathbf{x}$ .
$\delta\mathbf{x}$	真值 $\mathbf{x}$ 和估计值 $\hat{\mathbf{x}}$ 的误差

在这些定义之外，工作<sup>[27]</sup>引入了  $\boxplus$  和  $\boxminus$  两个封装操作，这些操作可以更紧凑地描述离散时间的系统状态空间模型。因为使用范围的问题，本文只关注于流型  $\text{SO}(3)$  和  $\mathbb{R}^n$ 。

首先给出  $\boxplus$  和  $\boxminus$  操作的含义和作用空间。

$$\boxplus : \mathcal{M} \times \mathbb{R}^n \rightarrow \mathcal{M}, \quad \boxminus : \mathcal{M} \times \mathcal{M} \rightarrow \mathbb{R}^n \quad (3.1)$$

$$\text{SO}(3) : \quad \mathbf{R} \boxplus \mathbf{r} = \mathbf{R} \cdot \text{Exp}(\mathbf{r}), \quad \mathbf{R}_1 \boxminus \mathbf{R}_2 = \text{Log}(\mathbf{R}_2^\top \mathbf{R}_1) \quad (3.2)$$

$$\mathbb{R}^n : \quad \mathbf{a} \boxplus \mathbf{b} = \mathbf{a} + \mathbf{b}, \quad \mathbf{a} \boxminus \mathbf{b} = \mathbf{a} - \mathbf{b} \quad (3.3)$$

$\text{Exp}(\mathbf{r})$  是  $\text{SO}(3)$  上的指数映射， $\text{Log}$  为其逆映射。 $\text{Exp}(\mathbf{r})$  在  $\text{SO}(3)$  上可以如下得到：

$$\text{Exp}(\mathbf{r}) = \mathbf{I} + \sin(\|\mathbf{r}\|) \frac{[\mathbf{r}]}{\|\mathbf{r}\|} + (1 - \cos(\|\mathbf{r}\|)) \frac{[\mathbf{r}]^2}{\|\mathbf{r}\|^2} \quad (3.4)$$

对于复合流形  $\mathcal{M} = \text{SO}(3) \times \mathbb{R}^n$ ，它是两个子流形  $\text{SO}(3)$  和  $\mathbb{R}^n$  的笛卡尔积。对此，我们有：

$$\begin{bmatrix} \mathbf{R} \\ \mathbf{a} \end{bmatrix} \boxplus \begin{bmatrix} \mathbf{r} \\ \mathbf{b} \end{bmatrix} = \begin{bmatrix} \mathbf{R} \boxplus \mathbf{r} \\ \mathbf{a} + \mathbf{b} \end{bmatrix} \quad (3.5)$$

$$\begin{bmatrix} \mathbf{R}_1 \\ \mathbf{a} \end{bmatrix} \boxminus \begin{bmatrix} \mathbf{R}_2 \\ \mathbf{b} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_1 \boxminus \mathbf{R}_2 \\ \mathbf{a} - \mathbf{b} \end{bmatrix} \quad (3.6)$$

## 二、运动学模型

### 1. 状态传播模型

取 IMU 坐标系写作 ( $I$ ) 为 Body 系并且取 IMU 初始化时的坐标系为世界坐标系 (写作  $G$ )，连续时间下的运动学模型是：

$${}^G\dot{\mathbf{R}}_I = {}^G\mathbf{R}_I[\mathbf{I}\boldsymbol{\omega}]^\wedge \quad (3.7)$$

$${}^G\dot{\mathbf{P}}_I = {}^G\mathbf{v}_I \quad (3.8)$$

$${}^G\dot{\mathbf{v}}_I = {}^G\mathbf{R}_I{}^I\mathbf{a} + {}^G\mathbf{g} \quad (3.9)$$

$$\dot{\mathbf{g}}_G = \mathbf{0} \quad (3.10)$$

$$\dot{\mathbf{b}}_g = \mathbf{n}_{b_g}, \quad \dot{\mathbf{b}}_a = \mathbf{n}_{b_a}, \quad \mathbf{I}\dot{\boldsymbol{\omega}} = \mathbf{w}_g, \quad \mathbf{I}\dot{\mathbf{a}} = \mathbf{w}_a \quad (3.11)$$

其中， ${}^G\mathbf{R}_I$ ,  ${}^G\mathbf{P}_I$ ,  ${}^G\mathbf{v}_I$  分别表示 IMU 在全局坐标系下的姿态、位置和速度； ${}^G\mathbf{g}$  为重力向量； $\mathbf{b}_g$ ,  $\mathbf{b}_a$  是由高斯噪声描述的陀螺仪与加速度计的偏置，满足：

$$\mathbf{n}_{bg} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{bg}), \quad \mathbf{n}_{ba} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{ba}) \quad (3.12)$$

符号  $[\cdot]^\wedge$  表示  $\mathbb{R}^3$  向量对应的反对称矩阵 (skew-symmetric matrix)，定义如下：

$$[\mathbf{r}]^\wedge = \begin{bmatrix} 0 & -r_3 & r_2 \\ r_3 & 0 & -r_1 \\ -r_2 & r_1 & 0 \end{bmatrix}, \quad \text{for } \mathbf{r} = [r_1, r_2, r_3]^\top \quad (3.13)$$

向量  ${}^I\boldsymbol{\omega}, {}^I\mathbf{a}$  分别表示在 IMU 坐标系下的角速度与线加速度。根据文献<sup>[27]</sup>，这两者可以建模为随机过程的样本，因此本文采用一阶积分器模型并引入高斯过程噪声：

$$\mathbf{w}_g \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_g), \quad \mathbf{w}_a \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_a) \quad (3.14)$$

为了进行数值计算，本文将上述连续模型离散化。设当前时刻为  $k$ ，采样间隔为  $\Delta t_k$ ，即前后两次测量之间的时间差（如两帧 IMU 数据），得到离散时间下的状态转移模型为：

$$\mathbf{x}_{k+1} = \mathbf{x}_k + (\Delta t_k \cdot \mathbf{f}(\mathbf{x}_k, \mathbf{w}_k)) \quad (3.15)$$

其中状态所在流形  $\mathcal{M}$ 、状态变量  $\mathbf{x}$ 、过程噪声  $\mathbf{w}$  以及状态导函数  $\mathbf{f}$  定义如下：

$$\mathcal{M} \triangleq \text{SO}(3) \times \mathbb{R}^{21} \text{ where } \dim(\mathcal{M}) = 24 \quad (3.16)$$

$$\mathbf{x} \triangleq [{}^G\mathbf{R}_I, {}^G\mathbf{P}_I, {}^G\mathbf{v}_I, \mathbf{b}_g, \mathbf{b}_a, \mathbf{g}_G, {}^I\boldsymbol{\omega}, {}^I\mathbf{a}] \quad (3.17)$$

$$\mathbf{w} \triangleq [\mathbf{n}_{bg}, \mathbf{n}_{ba}, \mathbf{w}_g, \mathbf{w}_a] \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}) \quad (3.18)$$

$$\mathbf{f}(\mathbf{x}, \mathbf{w}) \triangleq [{}^I\boldsymbol{\omega}, {}^G\mathbf{v}_I, {}^G\mathbf{R}_I {}^I\mathbf{a} + {}^G\mathbf{g}, \mathbf{n}_{b_g}, \mathbf{n}_{b_a}, \mathbf{0}_{3 \times 1}, \mathbf{w}_g, \mathbf{w}_a] \in \mathbb{R}^{24} \quad (3.19)$$

其中，协方差矩阵  $\mathbf{Q}$  定义为：

$$\mathbf{Q} = \text{diag}(\mathbf{Q}_{bg}, \mathbf{Q}_{ba}, \mathbf{Q}_g, \mathbf{Q}_a) \quad (3.20)$$

## 2. 测量模型

系统的每次测量来源于三种可能的方式中的一个，一个 LiDAR Batch，一次 IMU 采样或者一次轮速采样。这些数据源之间的采样互不相关，同时在不同时刻被系统采样，所以本文对三个测量来源分开进行建模。

假设 LiDAR 坐标系与机体坐标系（即 IMU 坐标系）之间的外参已知，一个在 IMU 坐标系下的 LiDAR 点  ${}^I\mathbf{p}_{mk}$  被认为是该点在 IMU 坐标系下的真实位置  ${}^I\mathbf{p}_k^{\text{gt}}$ （gt 代表 ground truth，是未知的）加上一个加性高斯噪声项  $\mathbf{n}_{\mathcal{L}_k} \sim \mathcal{N}(\mathbf{0}, \mathcal{R}_{\mathcal{L}_k})$ ：

$${}^I\mathbf{p}_{mk} = {}^I\mathbf{p}_k^{\text{gt}} + \mathbf{n}_{\mathcal{L}_k} \quad (3.21)$$

将该点投影到全局坐标系下，其应该位于地图中的一个局部小平面（假设存在），即满足如下几何约束：

$$\mathbf{0} = \mathbf{G}\mathbf{u}_k^T \left( {}^G\mathbf{T}_{I_k}({}^I\mathbf{p}_{m_k} - \mathbf{n}_{\mathcal{L}_k}) - \mathbf{G}\mathbf{q}_k \right) \quad (3.22)$$

其中， $\mathbf{G}\mathbf{u}_k$  是对应平面的法向量， $\mathbf{G}\mathbf{q}_k$  是该平面上任意一点。值得一提的是  ${}^G\mathbf{T}_{I_k}$  包含在状态量  $\mathbf{x}_k$  之中，所以式3.22中隐含了状态向量  $\mathbf{x}_k$  的测量方程。

IMU 观测包括 IMU 坐标系下的角速度  ${}^I\boldsymbol{\omega}_m$  与线加速度  ${}^I\mathbf{a}_m$ ，它们都受到偏置和噪声影响，可建模为：

$$\begin{bmatrix} {}^I\boldsymbol{\omega}_{m_k} \\ {}^I\mathbf{a}_{m_k} \end{bmatrix} = \begin{bmatrix} {}^I\boldsymbol{\omega}_k + \mathbf{b}_{g_k} + \mathbf{n}_{g_k} \\ {}^I\mathbf{a}_k + \mathbf{b}_{a_k} + \mathbf{n}_{a_k} \end{bmatrix} \quad (3.23)$$

其中  $\mathbf{n}_g \sim \mathcal{N}(\mathbf{0}, \mathcal{R}_g)$ ,  $\mathbf{n}_a \sim \mathcal{N}(\mathbf{0}, \mathcal{R}_a)$  为高斯噪声，合并写作  $\mathbf{n}_I = [\mathbf{n}_g^T, \mathbf{n}_a^T]^T \sim \mathcal{N}(\mathbf{0}, \mathcal{R}_I)$ ，其中协方差矩阵  $\mathcal{R}_I = \text{diag}(\mathcal{R}_g, \mathcal{R}_a)$ 。

轮速观测包括 Odometry (轮速里程计) 坐标系下的线速度  ${}^O\mathbf{v}_m$  和角速度  ${}^O\boldsymbol{\omega}_m$  (其中上标 O 表示在轮速里程计-Odometry 坐标系下)。首先需要将他们换算到 IMU 坐标系下。在本工作中，本文采用预标定轮速坐标系到 IMU 坐标系变换的方式解决高动态、有偏环境下的偏差观测不稳定问题和过度收敛问题。本文定义  ${}^I\mathbf{T}_O$ ,  ${}^I\mathbf{R}_O$ , 他们是 IMU 坐标系下的轮速坐标系的位姿表示。

$$\begin{aligned} {}^I\boldsymbol{\omega} &= {}^I\mathbf{R}_O \cdot {}^O\boldsymbol{\omega}_m \\ {}^I\mathbf{v} &= {}^I\mathbf{R}_O \cdot {}^O\mathbf{v}_m + [{}^I\mathbf{T}_O]^\wedge \cdot {}^I\boldsymbol{\omega} \\ {}^O\mathbf{v} &= {}^W\mathbf{R}_I \cdot {}^I\mathbf{v} \end{aligned} \quad (3.24)$$

因为在打滑场景下轮速计会带有严重偏差，所以不能直接假设轮速计采样为噪声不变的测量来源。为了解决这个问题，本文通过调整协方差使得分布覆盖有偏的数据点。通过四轮轮速的残差平方和 (RMS, 它将在后续的嵌入式底盘观测器中进行详细讨论) 构造线性协方差映射。轮速测量可以建模为：

$$RMS = \sum_i ({}^{wheel}v_i - {}^{wheel}\hat{v}_i)^2 \quad (3.25)$$

其中  ${}^{wheel}v_i$  为第 i 轮坐标系下的第 i 个轮的速度， ${}^{wheel}\hat{v}_i$  为第 i 个轮坐标系下的第 i 个轮的速度先验。这个先验可以来自于匀速底盘假设的卡尔曼滤波器投影到

每个轮坐标系下或者当前每个独立轮速以最小二乘解直接融合后投影到每个轮坐标系下。在理想无噪声四轮环境下，如果只有单个轮发生打滑，平均总残差为 1/4 的打滑轮偏差，所以打滑轮偏差的平方的期望为 RMS/2，每个非打滑轮为 RMS/6。它们都与 RMS 成线性关系，故可以构造与 RMS 成一次函数关系的协方差  $\mathcal{R}_v = k \times RMS + staticloss$  来覆盖轮速的最大偏差情况。其中常数项代表了轮速测量的固有噪声，正比例项用于在打滑情况下使高斯分布覆盖测量的大小。

$$\begin{bmatrix} {}^0\mathbf{v}_{m_k} \\ {}^I\boldsymbol{\omega}_{m_k} \end{bmatrix} = \begin{bmatrix} {}^0\mathbf{v}_k + \mathbf{n}_{v_k} \\ {}^I\boldsymbol{\omega}_k + \mathbf{n}_{g_k} \end{bmatrix} \quad (3.26)$$

其中  $\mathbf{n}_g \sim \mathcal{N}(\mathbf{0}, \mathcal{R}_g)$ ,  $\mathbf{n}_v \sim \mathcal{N}(\mathbf{0}, \mathcal{R}_v)$  为高斯噪声，合并写作  $\mathbf{n}_W = [\mathbf{n}_v^\top, \mathbf{n}_g^\top]^\top \sim \mathcal{N}(\mathbf{0}, \mathcal{R}_W)$ ，其中协方差矩阵  $\mathcal{R}_W = \text{diag}(\mathcal{R}_v, \mathcal{R}_g)$ 。构造  $\mathcal{R}_v = k \times RMS + staticloss$ 。因此，整个观测模型可以简洁写为：

$$\mathbf{0} = \mathbf{h}_L(\mathbf{x}_k, {}^I\mathbf{p}_{m_k}, \mathbf{n}_{\mathcal{L}_k}) \quad (3.27)$$

$$\begin{bmatrix} {}^I\boldsymbol{\omega}_m \\ {}^I\mathbf{a}_m \end{bmatrix} = \mathbf{h}_I(\mathbf{x}_k, \mathbf{n}_{\mathcal{I}_k}) \quad (3.28)$$

$$\begin{bmatrix} {}^0\mathbf{v}_m \\ {}^I\boldsymbol{\omega}_m \end{bmatrix} = \mathbf{h}_{\mathcal{W}}(\mathbf{x}_k, \mathbf{n}_{\mathcal{W}_k}) \quad (3.29)$$

### 三、扩展卡尔曼滤波器

本文采用了一个紧耦合的 EKF（扩展卡尔曼滤波器）来进行 Point-LIWO 的传感器融合，它的工作流程将在这个部分中进行展示。

#### 1. 状态传播

假设本文已经接收到  $k$  时刻的观测，并且此时的更新状态为  $\bar{\mathbf{x}}_k$ ，协方差为  $\bar{\mathbf{P}}_k$ 。从时刻  $k$  到  $k+1$  的状态传播可由式 3.15 通过设置  $\mathbf{w}_k = 0$  推导而得：

$$\hat{\mathbf{x}}_{k+1} = \bar{\mathbf{x}}_k \boxplus (\Delta t_k f(\bar{\mathbf{x}}_k, 0)) \quad (3.30)$$

协方差的传播形式为：

$$\hat{\mathbf{P}}_{k+1} = \mathbf{F}_{\mathbf{x}_k} \bar{\mathbf{P}}_k \mathbf{F}_{\mathbf{x}_k}^T + \mathbf{F}_{\mathbf{w}_k} \mathbf{Q}_k \mathbf{F}_{\mathbf{w}_k}^T \quad (3.31)$$

其中,  $\mathbf{Q}_k$  为过程噪声  $\mathbf{w}_k$  的协方差矩阵, 矩阵  $\mathbf{F}_{\mathbf{x}_k}$  与  $\mathbf{F}_{\mathbf{w}_k}$  的计算如下:

$$\begin{aligned} \mathbf{F}_{\mathbf{x}_k} &= \frac{\partial(\mathbf{x}_{k+1} \boxminus \hat{\mathbf{x}}_{k+1})}{\partial \delta \mathbf{x}_k} \Big|_{\delta \mathbf{x}_k=0, \mathbf{w}_k=0} \\ &= \frac{\partial((\bar{\mathbf{x}}_k \boxplus \delta \mathbf{x}_k) \boxplus (\Delta t_k f(\bar{\mathbf{x}}_k \boxplus \delta \mathbf{x}_k, 0)) \boxplus (\bar{\mathbf{x}}_k \boxplus \Delta t_k f(\bar{\mathbf{x}}_k, 0)))}{\partial \delta \mathbf{x}_k} \\ &= \begin{bmatrix} \mathbf{F}_{11} & 0 & 0 & 0 & 0 & 0 & \mathbf{I} \Delta t_k & 0 \\ 0 & \mathbf{I} & \mathbf{I} \Delta t_k & 0 & 0 & 0 & 0 & 0 \\ \mathbf{F}_{31} & 0 & \mathbf{I} & 0 & 0 & \mathbf{I} \Delta t_k & 0 & \mathbf{F}_{38} \\ 0 & 0 & 0 & \mathbf{I} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{I} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{I} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{I} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{I} \end{bmatrix} \end{aligned} \quad (3.32)$$

噪声对应的雅可比矩阵为:

$$\mathbf{F}_{\mathbf{w}_k} = \frac{\partial(\mathbf{x}_{k+1} \boxminus \hat{\mathbf{x}}_{k+1})}{\partial \mathbf{w}_k} \Big|_{\delta \mathbf{x}_k=0, \mathbf{w}_k=0} \quad (3.33)$$

$$= \frac{\partial(\bar{\mathbf{x}}_k \boxplus (\Delta t_k f(\bar{\mathbf{x}}_k, \mathbf{w}_k)) \boxplus (\bar{\mathbf{x}}_k \boxplus \Delta t_k f(\bar{\mathbf{x}}_k, 0)))}{\partial \mathbf{w}_k} \quad (3.34)$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \mathbf{I} & 0 & 0 & 0 \\ 0 & \mathbf{I} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{I} & 0 \\ 0 & 0 & 0 & \mathbf{I} \end{bmatrix} \quad (3.35)$$

其中  $\mathbf{x}_{k+1}$  表示时刻  $k+1$  的真实状态向量,  $\mathbf{F}_{11} = \text{Exp}(-{}^I \bar{\omega}_k \Delta t_k)$ ,  $\mathbf{F}_{38} = {}^G \bar{\mathbf{R}}_{I_k} \Delta t_k$ 。

## 2. 残差计算

(1) 激光雷达测量:

根据卡尔曼传播 (式3.30) 得到的预测位姿

$${}^G\hat{\mathbf{T}}_{I_{k+1}} = \left( {}^G\hat{\mathbf{R}}_{I_{k+1}}, {}^G\hat{\mathbf{p}}_{I_{k+1}} \right) \quad (3.36)$$

本文将激光雷达测量点  ${}^I\mathbf{p}_{m_{k+1}}$  投影至全局坐标系:

$${}^G\hat{\mathbf{p}}_{k+1} = {}^G\hat{\mathbf{R}}_{I_{k+1}} {}^I\mathbf{p}_{m_{k+1}} + {}^G\hat{\mathbf{p}}_{I_{k+1}} \quad (3.37)$$

然后在地图中搜索其 5 个最近邻点 (对于稠密地图而言通常控制在半径为 0.5m 的球中), 使用 IVox 数据结构进行组织 (为了效率本文把它退化成了一个均匀体素的哈希映射地图)。

若 5 个最近邻点不能拟合出局部平面 (即任意点到平面的距离大于设置的匹配阈值), 当前激光雷达点将直接并入地图而不进行残差计算或状态更新; 否则, 将计算该点的残差:

$$\mathbf{r}_{L_{k+1}} = 0 - \mathbf{h}_t(\hat{\mathbf{x}}_{k+1}, {}^I\mathbf{p}_{m_{k+1}}, 0) \quad (3.38)$$

$$= \mathbf{h}_L(\mathbf{x}_{k+1}, {}^I\mathbf{p}_{m_{k+1}}, \mathbf{n}_{L_{k+1}}) - \mathbf{h}_L(\hat{\mathbf{x}}_{k+1}, {}^I\mathbf{p}_{m_{k+1}}, 0) \quad (3.39)$$

$$\approx \mathbf{H}_{L_{k+1}} \delta \mathbf{x}_{k+1} + \mathbf{D}_{L_{k+1}} \mathbf{n}_{L_{k+1}} \quad (3.40)$$

其中  $\delta \mathbf{x}_{k+1} = \mathbf{x}_{k+1} \ominus \hat{\mathbf{x}}_{k+1}$ ,  $\mathbf{x}_{k+1}$  为  $k+1$  时刻的状态向量。

残差雅可比矩阵  $\mathbf{H}_{L_{k+1}}$  和噪声雅可比矩阵  $\mathbf{D}_{L_{k+1}}$  为:

$$\mathbf{H}_{L_{k+1}} = \left. \frac{\partial \mathbf{h}_L(\hat{\mathbf{x}}_{k+1} \boxplus \delta \mathbf{x}, {}^I\mathbf{p}_{m_{k+1}}, 0)}{\partial \delta \mathbf{x}} \right|_{\delta \mathbf{x}=0} \quad (3.41)$$

$$= \begin{bmatrix} -{}^G\mathbf{u}_{k+1}^\top {}^G\hat{\mathbf{R}}_{I_{k+1}} [{}^I\mathbf{p}_{m_{k+1}}] & {}^G\mathbf{u}_{k+1}^\top & 0_{1 \times 18} \end{bmatrix} \quad (3.42)$$

$$\mathbf{D}_{L_{k+1}} = \left. \frac{\partial \mathbf{h}_L(\hat{\mathbf{x}}_{k+1}, {}^I\mathbf{p}_{m_{k+1}}, \mathbf{n})}{\partial \mathbf{n}} \right|_{\mathbf{n}=0} = -{}^G\mathbf{u}_{k+1}^\top {}^G\hat{\mathbf{R}}_{I_{k+1}} \quad (3.43)$$

基于 Batch Lidar 的思想, 本文将参考 Fast-LIO<sup>[14]</sup>思想的单点残差构造推广到基于 Batch 的残差构建。首先, 通过自身速度和角速度传播运动状态, 将 Batch 中每一个点 (以序号为  $j$  的点为例) 投影到 Batch 中最后一个点对应时刻的坐标系下。

$$\Delta t_j = t_j - t_{Last} \quad (3.44)$$

$$\theta_j = \|\boldsymbol{\omega}\| \cdot \Delta t_j, \quad \mathbf{a} = \frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|} \quad (3.45)$$

$$\mathbf{R}_j = \exp(\theta_j [\mathbf{a}]^\wedge), \quad \mathbf{T}_j = \mathbf{v} \cdot \Delta t_j \quad (3.46)$$

$$\mathbf{p}'_j = \mathbf{R}_j \mathbf{p}_j + \mathbf{T}_j \quad (3.47)$$

其中  $t_j$  和  $t_{Last}$  分别代表序号为  $j$  的点的采样时间戳和 Batch 中最后一个点的采样时间戳,  $\theta_j$ 、 $\mathbf{a}$ 、 $\mathbf{T}_j$  分别对应时间差下 IMU 在世界坐标系中运动的转角、转轴和平移向量,  $\mathbf{R}_j$  为转动对应的旋转矩阵,  $\mathbf{p}'_j$  为投影到 Batch 中最后一个点坐标系下的序号为  $j$  的点。

然后将成功和局部拟合平面匹配的点的  $\mathbf{H}$  矩阵和  $\mathbf{D}$  矩阵按行合并为 Batch 的残差雅可比矩阵和噪声雅可比矩阵。

$$\mathbf{H}_L = \begin{bmatrix} \mathbf{H}_{L_0} \\ \mathbf{H}_{L_1} \\ \dots \\ \mathbf{H}_{L_{Last}} \end{bmatrix} \quad \mathbf{D}_L = \begin{bmatrix} \mathbf{D}_{L_0} \\ \mathbf{D}_{L_1} \\ \dots \\ \mathbf{D}_{L_{Last}} \end{bmatrix} \quad (3.48)$$

## (2) IMU 测量:

对于 IMU 测量, 它由 6 个测量维度组成 (3 GYRO + 3 ACC), 若某一维度测量饱和 (根据当测量值和传感器量程的差值判断, 本文设置为 95%), 通过剩余的加速度和角速度测量维度构建 IMU 残差:

$$\mathbf{r}_{I_{k+1}} = \begin{bmatrix} I \boldsymbol{\omega}_{m_{k+1}} \\ I \mathbf{a}_{m_{k+1}} \end{bmatrix}^\top - \mathbf{h}_I(\hat{\mathbf{x}}_{k+1}, 0) \quad (3.49)$$

$$= \mathbf{h}_I(\mathbf{x}_{k+1}, \mathbf{n}_{I_{k+1}}) - \mathbf{h}_I(\hat{\mathbf{x}}_{k+1}, 0) \quad (3.50)$$

$$= \mathbf{H}_{I_{k+1}} \delta \mathbf{x}_{k+1} + \mathbf{D}_{I_{k+1}} \mathbf{n}_{I_{k+1}} \quad (3.51)$$

其中残差雅可比矩阵和噪声雅可比矩阵为:

$$\mathbf{H}_{I_{k+1}} = \left. \frac{\partial \mathbf{h}_I(\hat{\mathbf{x}}_{k+1} \boxplus \delta \mathbf{x}, 0)}{\partial \delta \mathbf{x}} \right|_{\delta \mathbf{x}=0} = \begin{bmatrix} 0_{6 \times 9} & \mathbf{I}_{6 \times 6} & 0_{6 \times 3} & \mathbf{I}_{6 \times 6} \end{bmatrix} \quad (15)$$

$$\mathbf{D}_{I_{k+1}} = \left. \frac{\partial \mathbf{h}_I(\hat{\mathbf{x}}_{k+1}, \mathbf{n})}{\partial \mathbf{n}} \right|_{\mathbf{n}=0} = \mathbf{I}_{6 \times 6} \quad (3.52)$$

### (3) 轮速测量:

对于原始轮速测量，它由 3 个维度测量组成 ( $Vx$   $Vy$   $\omega_z$ )，与 IMU 不同，轮速并不会存在明显的饱和问题。但是它存在偏置和比较大的噪声。

比较好的处理方法是首先将轮速扩展为 6 个测量维度，缺少的维度通过 0 速度填充 (3 VEL + 3 OMEGA)。这个处理是合理的，相当于给系统提供了机器人贴服于地面的假设。然后通过式3.24将轮速坐标系下的六个轮速测量维度变换到世界系下的速度和 IMU 坐标系下的角速度，和状态量坐标系对齐。

通过变换后的六个通道数据与状态量计算残差，然后根据预先设置的六个标志位控制通道是否丢弃。一般来说框架会屏蔽轮速对状态量中角速度的影响以增强里程计角速度的平滑程度。以感知机器人的振动与起伏。

$$\mathbf{r}_{W_{k+1}} = \begin{bmatrix} I \\ W \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega}_{m_{k+1}} \\ \mathbf{v}_{m_{k+1}} \end{bmatrix}^\top - \mathbf{h}_w(\hat{\mathbf{x}}_{k+1}, 0) \quad (3.53)$$

$$= \mathbf{h}_w(\mathbf{x}_{k+1}, \mathbf{n}_{w_{k+1}}) - \mathbf{h}_w(\hat{\mathbf{x}}_{k+1}, 0) \quad (3.54)$$

$$= \mathbf{H}_{w_{k+1}} \delta \mathbf{x}_{k+1} + \mathbf{D}_{w_{k+1}} \mathbf{n}_{w_{k+1}} \quad (3.55)$$

其中残差雅可比矩阵和噪声雅可比矩阵为：

$$\mathbf{H}_{w_{k+1}} = \left. \frac{\partial \mathbf{h}_w(\hat{\mathbf{x}}_{k+1} \boxplus \delta \mathbf{x}, 0)}{\partial \delta \mathbf{x}} \right|_{\delta \mathbf{x}=0} = \begin{bmatrix} 0_{6 \times 9} & \mathbf{I}_{6 \times 6} & 0_{6 \times 3} & \mathbf{I}_{6 \times 3} & 0_{6 \times 3} \end{bmatrix} \quad (3.56)$$

$$\mathbf{D}_{w_{k+1}} = \left. \frac{\partial \mathbf{h}_w(\hat{\mathbf{x}}_{k+1}, \mathbf{n})}{\partial \mathbf{n}} \right|_{\mathbf{n}=0} = \mathbf{I}_{6 \times 6} \quad (3.57)$$

### (4) 总结:

综上，对于任一 LiDAR 测量 (如式3.40) 或 IMU 测量 (如式3.51)，或轮速测量 (如式3.55)，其残差  $\mathbf{r}_{k+1}$  可统一表示为：

$$\mathbf{r}_{k+1} \approx \mathbf{H}_{k+1} \delta \mathbf{x}_{k+1} + \mathbf{D}_{k+1} \mathbf{n}_{k+1}, \quad \mathbf{n}_{k+1} \sim \mathcal{N}(0, \mathbf{R}_{k+1}) \quad (3.58)$$

## 3. 状态更新

状态量  $\mathbf{x}_{k+1}$  和协方差  $\hat{\mathbf{P}}_{k+1}$  给出误差状态的先验分布：

$$\delta \mathbf{x}_{k+1} = \mathbf{x}_{k+1} \boxminus \hat{\mathbf{x}}_{k+1} \sim \mathcal{N}(\mathbf{0}, \hat{\mathbf{P}}_{k+1}) \quad (3.59)$$

由测量模型（式3.58）可得观测分布：

$$\mathbf{D}_{k+1}n_{k+1} = r_{k+1} - \mathbf{H}_{k+1}\delta\mathbf{x}_{k+1} \sim \mathcal{N}(0, \bar{\mathcal{R}}_{k+1}) \quad (3.60)$$

$$\bar{\mathcal{R}}_{k+1} = \mathbf{D}_{k+1}\mathcal{R}_{k+1}\mathbf{D}_{k+1}^T \quad (3.61)$$

将先验与测量融合，可得最小化代价函数：

$$\arg \min_{\delta\mathbf{x}_{k+1}} \|r_{k+1} - \mathbf{H}_{k+1}\delta\mathbf{x}_{k+1}\|_{\bar{\mathcal{R}}_{k+1}}^2 + \|\delta\mathbf{x}_{k+1}\|_{\hat{\mathbf{P}}_{k+1}}^2 \quad (3.62)$$

由此得最优解，即卡尔曼滤波器的更新形式：

$$\delta\mathbf{x}_{k+1}^O = \mathbf{K}_{k+1}r_{k+1} \quad (3.63)$$

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1|k}\mathbf{H}_{k+1}^T (\mathbf{H}_{k+1}\mathbf{P}_{k+1}\mathbf{H}_{k+1}^T + \bar{\mathcal{R}}_{k+1})^{-1} \quad (3.64)$$

$$\mathbf{P}_{k+1} = (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1})\hat{\mathbf{P}}_{k+1} \quad (3.65)$$

状态更新：

$$\bar{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_{k+1} \boxplus \delta\mathbf{x}_{k+1}^O \quad (3.66)$$

为进行下一步传播，还需估计  $\mathbf{P}_{k+1}^-$ ，即与真实状态误差的协方差：

$$\mathbf{x}_{k+1} \boxminus \bar{\mathbf{x}}_{k+1} = \mathbf{J}_{k+1}^{-1}\delta\mathbf{x}_{k+1}^* \quad (3.67)$$

$$\mathbf{P}_{k+1}^t = \mathbf{J}_{k+1}^{-1}\mathbf{P}_{k+1}(\mathbf{J}_{k+1}^{-1})^\top \quad (3.68)$$

其中  $\mathbf{J}_{k+1}$  为投影矩阵，定义为：

$$\mathbf{J}_{k+1} = \frac{\partial(\hat{\mathbf{x}}_{k+1} \boxplus \delta\mathbf{x} \boxminus \bar{\mathbf{x}}_{k+1})}{\partial \delta\mathbf{x}} \Big|_{\delta\mathbf{x}=\delta\mathbf{x}_{k+1}^O} = \begin{bmatrix} \mathbf{A}(\boldsymbol{\theta}_{k+1})^{-T} & \mathbf{0}_{3 \times 21} \\ \mathbf{0}_{21 \times 3} & \mathbf{I}_{21} \end{bmatrix} \quad (3.69)$$

其中  $\mathbf{A}(\cdot)^-$  为旋转在李代数下的映射矩阵。定义如下：

$$\mathbf{A}(\mathbf{u})^{-1} = \mathbf{I} - \frac{[\mathbf{u}]}{2} + \left(1 - \frac{\|\mathbf{u}\|}{2}\cot\left(\frac{\|\mathbf{u}\|}{2}\right)\right) \frac{[\mathbf{u}]}{\|\mathbf{u}\|} \quad (3.70)$$

残差  $x_{k+1} \boxminus \bar{x}_{k+1}$  的协方差矩阵是：

$$\bar{\mathbf{P}}_{k+1} = \mathbf{J}_{k+1}\mathbf{P}_{k+1}\mathbf{J}_{k+1}^T \quad (3.71)$$

## 四、嵌入式底盘观测器

### 1. 运动学建模

运动学建模基于两舵轮两全向轮 (2WIS4WID) 底盘构型展开。建模的假设是舵轮没有沿转轴方向的滑动，全向轮小滚子有不可观测的自由转动（表现为平行于转轴方向的不可测速度）。所以，我们把两种轮子分开进行考虑。我们从工

作<sup>[28]</sup>中引用了适用于 4WIS4WID 的图3.1, 我们将基于图3.1的符号定义将两个全向轮角度描述为固定的  $\delta_1\delta_4$  展开后续介绍:

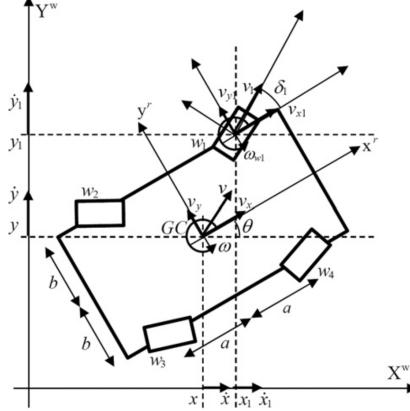


图 3.1 底盘符号定义

### (1) 舵轮建模

在无滑动假设下, 根据机器人刚体约束, 可得到舵轮与机器人本体之间的速度关系:

$$v_{xi} = v_i \cos(\delta_i) = v_x - y_{wi}^r \omega \quad (3.72)$$

$$v_{yi} = v_i \sin(\delta_i) = v_y + x_{wi}^r \omega \quad (3.73)$$

由此可得轮子 2 和 3 与本体速度的关系:

$$\mathbf{P} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} = \mathbf{X} \begin{bmatrix} v_2 \\ v_3 \end{bmatrix} \quad (3.74)$$

其中,

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & -b \\ 0 & 1 & -a \\ 1 & 0 & b \\ 0 & 1 & -a \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} \cos(\delta_2) & 0 \\ \sin(\delta_2) & 0 \\ 0 & \cos(\delta_3) \\ 0 & \sin(\delta_3) \end{bmatrix} \quad (3.75)$$

### (2) 全向轮建模

全向轮存在一个无法测量的滑动速度分量 (记作  $v'_i$ )。根据轮子方向与坐标轴的旋转关系, 有:

$$\begin{bmatrix} \cos(\delta_i) & -\sin(\delta_i) \\ \sin(\delta_i) & \cos(\delta_i) \end{bmatrix} \begin{bmatrix} v_i \\ v'_i \end{bmatrix} = \begin{bmatrix} v_{xi} \\ v_{yi} \end{bmatrix} \quad (3.76)$$

结合式 (3.72) 和 (3.73) , 可得全向轮速度  $v_i$  与本体速度的关系:

$$v_i = v_x \cos(\delta_i) + v_y \sin(\delta_i) - \omega y_{wi}^r \cos(\delta_i) + \omega x_{wi}^r \sin(\delta_i) \quad (3.77)$$

其矩阵形式为:

$$\begin{bmatrix} \cos(\delta_i) & \sin(\delta_i) & -y_{wi}^r \cos(\delta_i) + x_{wi}^r \sin(\delta_i) \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} = v_i \quad (3.78)$$

**注意:** 全向轮的  $\delta_i$  为固定常数, 不随时间变化。

### (3) 舵轮与全向轮联合建模

综合以上建模结果, 得到整体轮速观测模型:

$$\mathbf{P} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} = \mathbf{X} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} \quad (3.79)$$

其中:

$$\mathbf{P} = \begin{bmatrix} \cos(\delta_1) & \sin(\delta_1) & -b \cos(\delta_1) + a \sin(\delta_1) \\ 1 & 0 & -b \\ 0 & 1 & -a \\ 1 & 0 & b \\ 0 & 1 & -a \\ \cos(\delta_4) & \sin(\delta_4) & b \cos(\delta_4) + a \sin(\delta_4) \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\delta_2) & 0 & 0 \\ 0 & \sin(\delta_2) & 0 & 0 \\ 0 & 0 & \cos(\delta_3) & 0 \\ 0 & 0 & \sin(\delta_3) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.80)$$

**注意:** 我们通过伪逆来求得一个能量最小解, 即每个轮子都满足约束下的最小残差平方和解。

由此得到速度估计模型:

$$\begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} = \mathbf{P}^\dagger \mathbf{X} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} \quad (3.81)$$

### (4) 正向运动学

系统的正向运动学可由以下矩阵变换描述。

$$\mathbf{X}^\dagger = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & c_2 & s_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_3 & s_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.82)$$

$$\mathbf{X}^\dagger \mathbf{P} = \begin{bmatrix} c_1 & s_1 & as_1 - bc_1 \\ c_2 & s_2 & -as_2 - bc_2 \\ c_3 & s_3 & -as_3 + bc_3 \\ c_4 & s_4 & as_4 + bc_4 \end{bmatrix} \quad (3.83)$$

最终，可以通过以下线性组合表达各轮的速度  $v_i$  与底盘中心速度  $(v_x, v_y)$  和角速度  $w$  的关系：

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} c_1 v_x + s_1 v_y + w(as_1 - bc_1) \\ c_2 v_x + s_2 v_y + w(-as_2 - bc_2) \\ c_3 v_x + s_3 v_y + w(-as_3 + bc_3) \\ c_4 v_x + s_4 v_y + w(as_4 + bc_4) \end{bmatrix} \quad (3.84)$$

## 2. 观测器设计

### (1) 线性卡尔曼滤波器

线性 Kalman 滤波器的预测与更新方程如下：

$$\mathbf{x}_k^- = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{B}_k \mathbf{u}_k \quad (3.85)$$

$$\mathbf{P}_k^- = \mathbf{F}_k \mathbf{P}_{k-1} \mathbf{F}_k^T + \mathbf{Q} \quad (3.86)$$

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R})^{-1} \quad (3.87)$$

$$\mathbf{x}_k = \mathbf{x}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \mathbf{x}_k^-) \quad (3.88)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- \quad (3.89)$$

## (2) 本场景下的观测器设计

本系统的目标是观测底盘中心的平动速度速度和角速度，输入为四轮轮速。结合式3.83以及匀速假设，可以得到如下状态空间表达式：

$$\mathbf{u}_k = 0 \quad (3.90)$$

$$\mathbf{x} = \begin{bmatrix} v_x & v_y & \omega \end{bmatrix}^T \quad (3.91)$$

$$\mathbf{z} = \begin{bmatrix} v_1 & v_2 & v_3 & v_4 \end{bmatrix}^T \quad (3.92)$$

$$F = \begin{bmatrix} \cos(w \cdot dt) & \sin(w \cdot dt) & 0 \\ -\sin(w \cdot dt) & \cos(w \cdot dt) & 0 \\ 0 & 0 & 1 \end{bmatrix}_{3 \times 3} \quad (3.93)$$

$$H = \begin{bmatrix} c_1 & s_1 & as_1 - bc_1 \\ c_2 & s_2 & -as_2 - bc_2 \\ c_3 & s_3 & -as_3 + bc_3 \\ c_4 & s_4 & as_4 + bc_4 \end{bmatrix}_{4 \times 3} \quad (3.94)$$

需要说明的是式3.93的来源：速度应在惯性坐标系下考虑。假设前一时刻的真实状态已知，时间间隔也已知，那么可以计算出相对于上一帧的角度变化，从而通过二维旋转变换推算当前时刻底盘速度。

## (3) 协方差设计

接下来讨论关键问题：如何设置  $Q$  与  $R$ 。

$Q$  可以通过机器人本身的可能最大加速度根据一阶模型进行设计：

$$Q = \begin{bmatrix} \left(\frac{a_{x_{\max}} \cdot dt}{2}\right)^2 & 0 & 0 \\ 0 & \left(\frac{a_{y_{\max}} \cdot dt}{2}\right)^2 & 0 \\ 0 & 0 & \left(\frac{a_{\omega_{\max}} \cdot dt}{2}\right)^2 \end{bmatrix}_{3 \times 3} \quad (3.95)$$

因为四个轮子的测量之间是无关的， $R$  可以写为对角线形式：

$$R = \begin{bmatrix} \sigma_1^2 & 0 & 0 & 0 \\ 0 & \sigma_2^2 & 0 & 0 \\ 0 & 0 & \sigma_3^2 & 0 \\ 0 & 0 & 0 & \sigma_4^2 \end{bmatrix}_{4 \times 4} \quad (3.96)$$

其中  $\sigma_i^2$  表示不同轮在滑动和非滑动条件下的测量方差。

为了保证卡尔曼滤波器有一定对抗打滑的能力，我们需要根据输入调节  $\sigma_i^2$ ，保证高斯分布覆盖打滑速度。这在理论上是可行的，因为输入的信息包含四个轮子而只需要映射到三个速度，我们可以通过剩下的一个维度的信息对测量噪声进行评估。

一个简单高效的评估方案是残差和残差平方和 (RMS)。操作流程是先根据式3.81计算无打滑假设下的  $\hat{v}_x \hat{v}_y \hat{\omega}$ ，然后根据式3.84计算每个轮子在无打滑假设下的速度  $\hat{v}_i$ 。定义每个轮子的残差  $loss_i = (\hat{v}_i - v_i)^2$ 。这表示了每个轮子可能的偏移范围。我们将它作为协方差的一部分构造协方差  $\sigma_i^2 = k * loss_i$ ，k 为一个可调整的参数。

同时我们将残差平方和  $RMS = \sum loss_i$  传递给 Point-LIWO 系统，让其拥有了分辨轮速数据源打滑和动态整定协方差的能力。

## 第四章 测试与实验

本文将 Point-LIWO 在各种有代表性的室内场景进行了测试，其中包括狭窄退化场景，大面积开阔场景和常见的楼道场景。本文的测试表明，Point-LIWO 的 Batch based 思想取得了很好的效果。

本文给融合了轮速的 Point-LIWO 设置了一个挑战性任务：在地形复杂，障碍物密集，点云遮挡多，多减速带的大规模室内场馆进行建图和定位。Point-LIWO 系统在其中表现出了很高的定位性能与实时性，成功完成了这项在雷达退化且轮速有部分时刻打滑场景下的挑战。体现了轮速融合对系统稳定性的提升，验证了所提出的融合方式的可靠。

### 一、实验平台介绍

本文自主研发了一款轮式移动机器人，它有着体积小，重量轻和运行灵活的特点。本文的实验与测试将基于它展开。

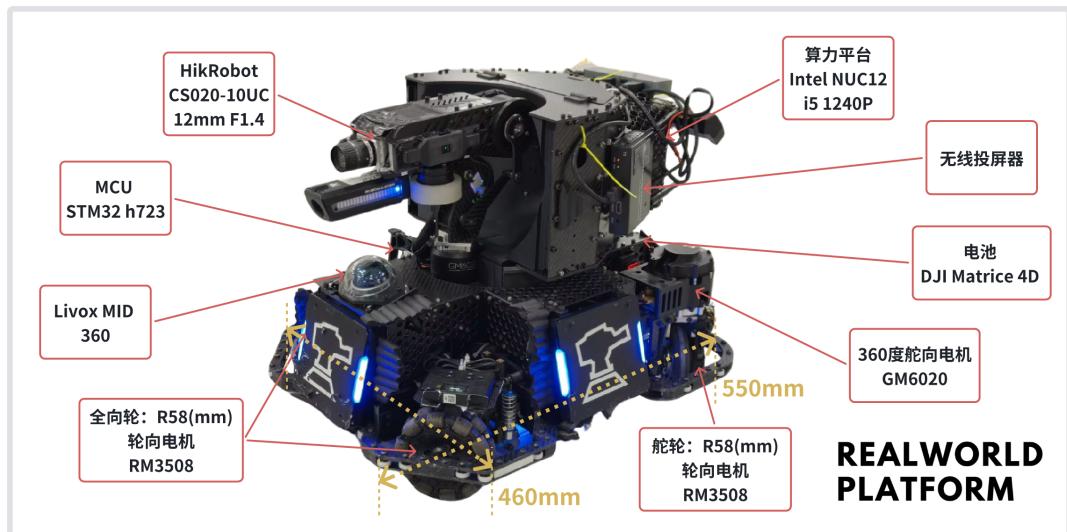


图 4.1 两舵轮两全向轮移动机器人平台展示

**计算设备：**其搭载了 Intel 12th NUC 1240p 作为计算平台，同时搭载了两块 stm32 h723 单片机用于底盘和云台的实时控制和传感器数据汇总。

**传感器：**平台搭载了一个 MID360 激光雷达用于定位，感知，规划，控制任务与识别任务。Livox MID360 激光雷达具有 360° 水平视野和 59° 垂直视野，可

以支持 20 万点每秒的测量速率。同时 MID360 内置了 IMU，可以以 200Hz 的频率反馈陀螺仪和加速度计数据。

**供电：**平台安装了一块大疆 Matrice 4D 电池，可以支持机器人至多达两小时的持续运行。

**底盘设计：**平台采用两舵轮两全向轮设计，保证基本在和四舵轮性能无差异的基础上的尽可能轻量化设计。同时底盘尺寸控制在了 460mm(W)\*550mm(L)，这样的小尺寸使其拥有了穿过狭窄地形的能力。

## 二、实时建图性能展示

为了证明本文提出的逐 Batch 更新和融合轮速方法对基于 EKF 框架的紧耦合 LIO 系统的提升，本文在两个不同但具有代表性的场景进行了测试并与以往工作进行比较。它们分别为：武汉光谷机器人创新中心搭建的 RoboMaster2025 比赛场地（非结构化大规模场景），中国科学技术大学 RoboWalker 战队基地大厅（非结构化小规模场景）。在这两个场景中，激光雷达固定在机器人底盘上，这带来了实验的第一个挑战：激光雷达随着机器人在地面的运动剧烈振动，这样的振动将直接传递到加速度计导致加速度饱和从而使得 IMU 测量失效。同时，雷达偏心于底盘约 25cm 进行安装，这会导致底盘旋转时 IMU 坐标系下有非常非线性的运动速度，对 EKF 这样基于一阶展开的框架造成很大影响。此外，场景中有许多隧道和遮挡，这带来了实验的另一个挑战：为了避免误识别自身，雷达设置了大约 0.5m 的自身点云过滤范围，这将导致激光雷达点云数量在这样的场景中下降到接近于 0，导致激光雷达测量失效。最后，机器人需要通过起伏路段，这带来了第三个挑战：较小的悬行程导致四个轮子并不都与地面接触，未触地的轮子会空转，导致轮速计的误差增大甚至失效。

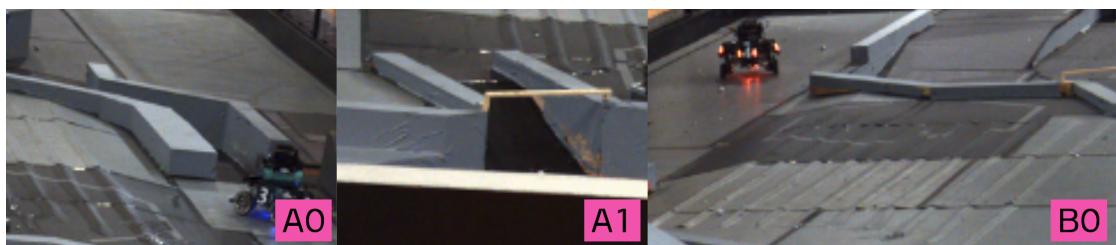


图 4.2 挑战地形展示：A0 与 A1 为隧道地形，宽 500mm；B0 为大面积起伏路段

本文采用了现场录制 ROS Bag，然后运行 ROS Bag 给里程计算法提供传感器数据的测试形式保证测试的可重复性。在运行前本文通过调整参数确保里程计系统以实时速度运行，同时尽可能优化系统表现。

## 1. 大场景建图

场地主体为 RoboMaster2025 比赛规则定义下的标准比赛场地，周边为场馆墙壁，场馆顶棚为斜面，离地高度范围从 20m 到 30m。场地边有人和机器人等动态障碍物无规律运动。

### (1) 场地展示



图 4.3 大规模非结构化场景

### (2) 无轮速的 Point-LIWO

在这里，本文先不比较 Point Lio<sup>[1]</sup> 和 Point LIWO 在无轮速下的差别。因为两者的主要区别体现在里程计的精度上。这部分内容将在本文后续部分讨论。

首先，本文使用不带轮速的 Point-LIWO 进行建图，降采样为 Voxel Size 为 0.1m 的体素降采样。得到了如图4.5的效果。其中数字标志 0-2 表示建图过程，A/B 表示两个显著特征。

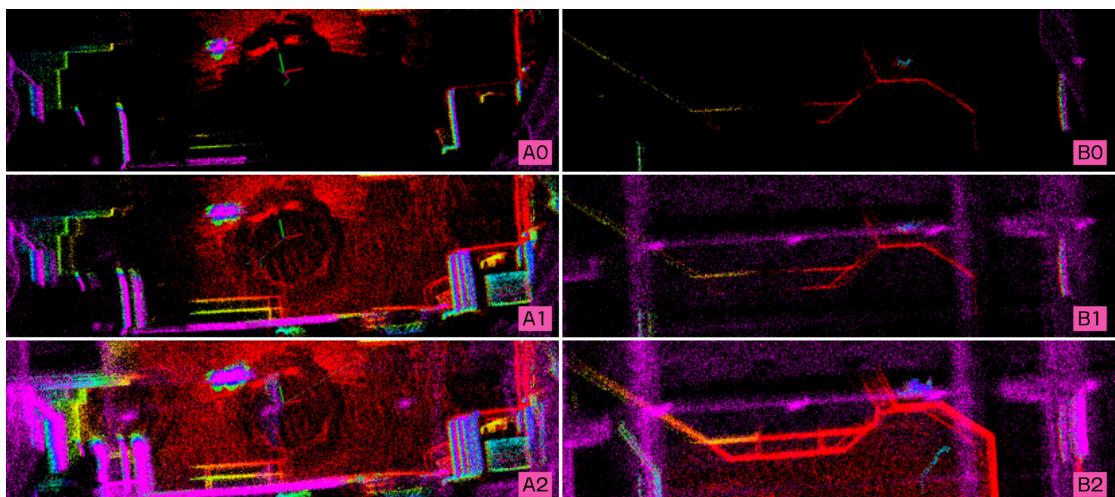


图 4.4 无轮速 Point-LIWO 局部建图效果

运行过程如下：

从 A0/B0 开始，在缓慢平移的情况下点云累积准确。

随后，机器人在原地进行了约  $6\text{rad/s}$  的旋转。旋转后特征如 A1/B1 所示，可以看出 A1/B1 中的墙壁出现重影，这代表自身位置估计出现偏差。但是因为增量建图，里程计依然可以稳定运行。

接着，本文将机器人通过隧道和起伏路段运行到场地的另一端后返回原地。可以看出 A2/B2 中的墙壁特征变得很“厚”，这代表里程计发生了漂移但是依然在运行，姿态估计基本没有很大偏差，足以满足一般的实时机器人控制任务的要求。

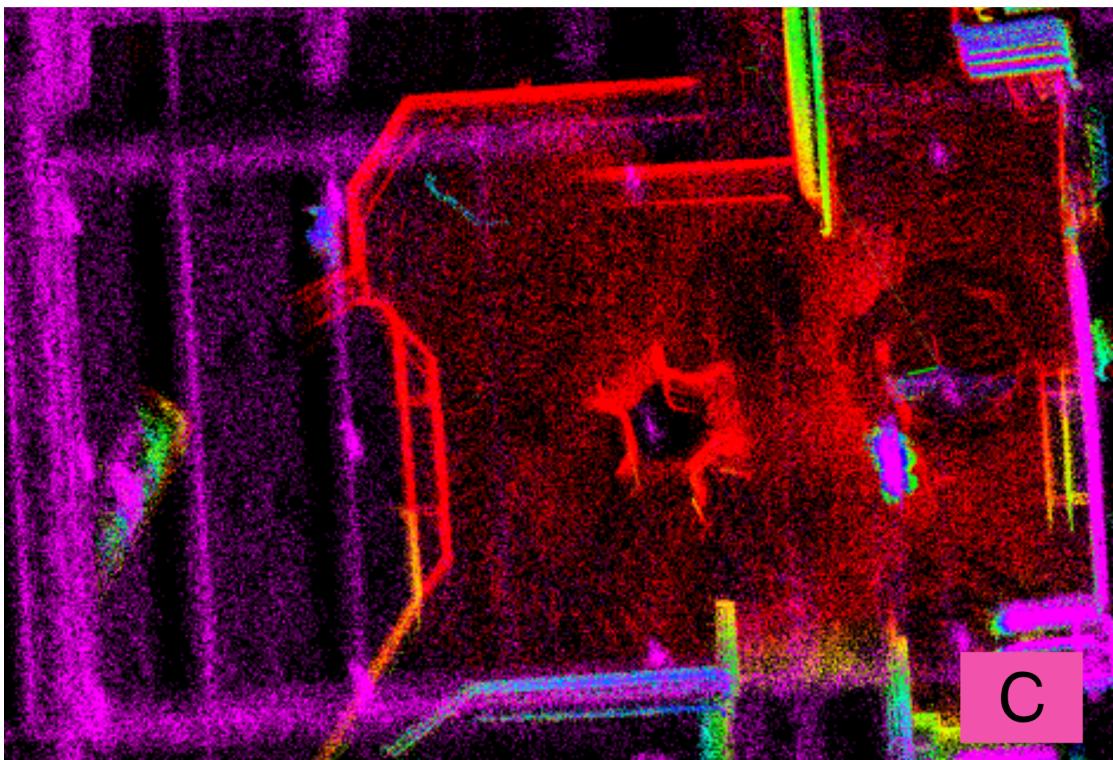


图 4.5 无轮速 Point-LIWO 的全局建图效果（图片逆时针旋转  $90^\circ$ ）

### (3) 有轮速的 Point-LIWO

重复和前面一样的操作，有轮速的 Point LIWO 在效果上获得了很大的提升。定位建图精度显著提高。

同样，本文采用降采样为 Voxel Size 为  $0.1\text{m}$  的体素降采样。图片分为 D0 和 D1，其中 D0 为  $6\text{rad/s}$  旋转以后的累积点云图，D1 为通过所有路段后累积的全场地点云图。

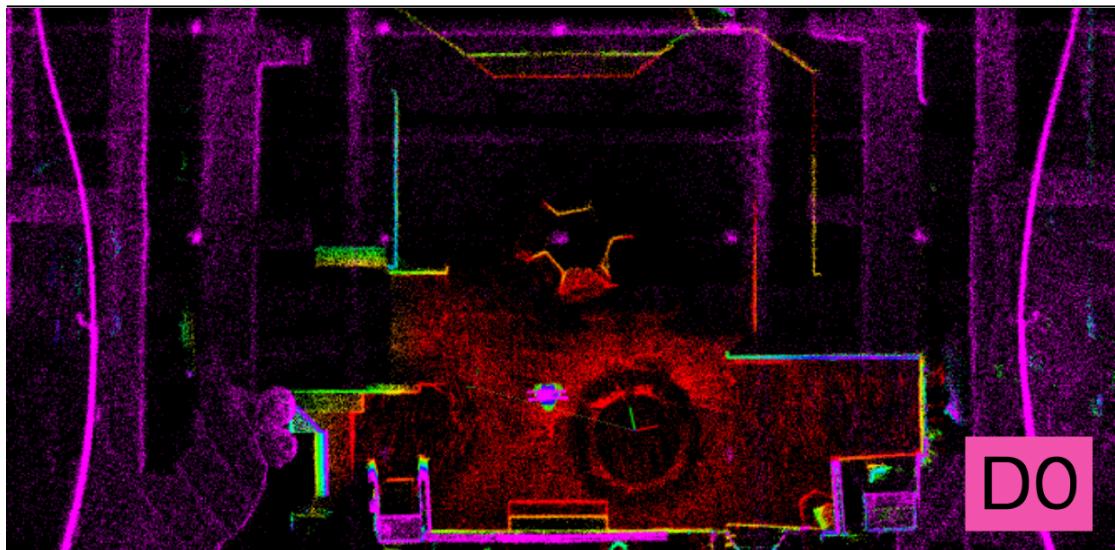


图 4.6 有轮速 Point-LIWO 的原地旋转后效果

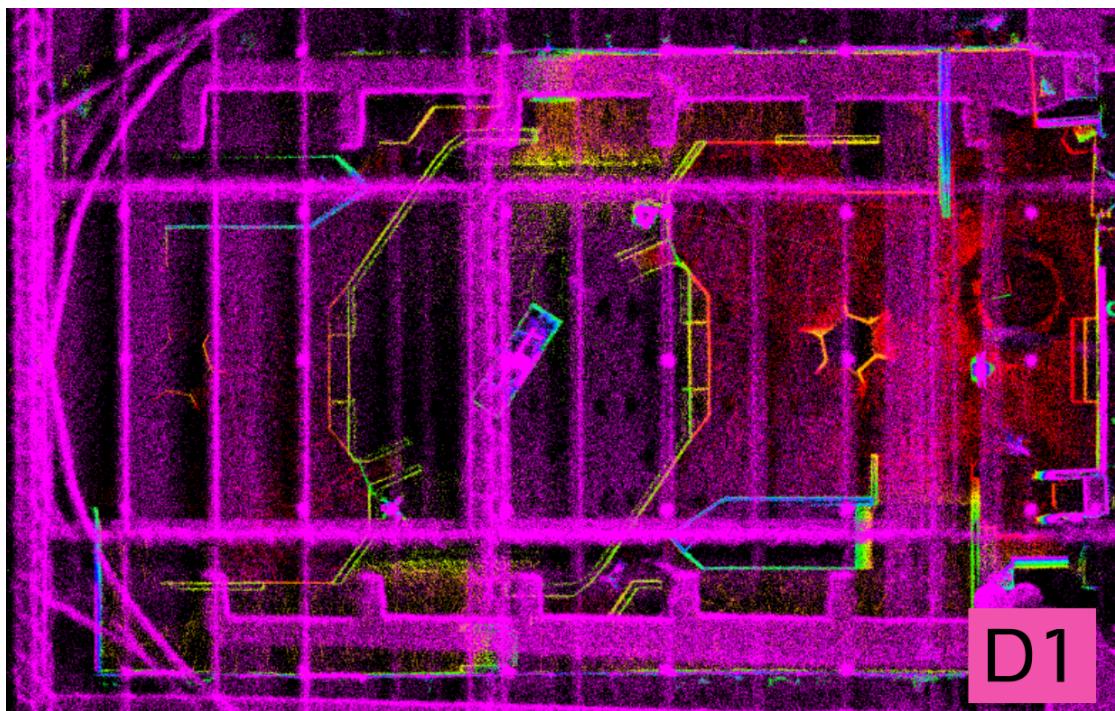


图 4.7 有轮速 Point-LIWO 的全局建图效果（图片逆时针旋转 90°）

可以看出，对于墙壁特征，有轮速 Point LIWO 的建图效果有了很大的改进。从结果上来说，这也使 Point Liwo 的点云残差构造更为准确，进而进入建图与定位的循环正反馈，提高整个系统的运行性能。

结束时，在回到启动位置之后，可以看出 D1 右侧的墙壁点云“厚度”基本没有发生改变，仍然可以清晰看出墙壁（厚度为 10cm）所具有的厚度（扫描得到两个表面）。这表明在大面积室内场景中，Point-Liwo 在机器人经过 100m 的长途跋涉之后将定位误差控制在了 2cm 以内。体现了 Point-Liwo 框架的优越性。

## 2. 小场景建图

### (1) 场地展示

场地主体为一般室内实验室场景，场景顶部有离地约 8m 高的通风管道，房顶离地高约 10m，有动态行人。

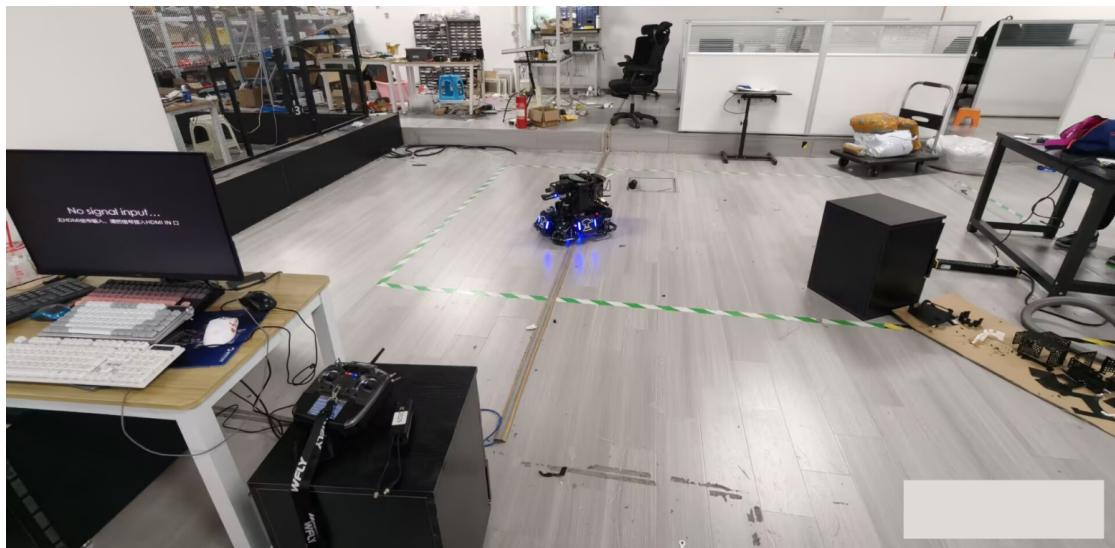


图 4.8 小规模室内非结构化场景

### (2) 静止下逐点和逐 Batch 的对比

本文运行了 Point-Lio<sup>[1]</sup>和没有轮速加入的 Point-Liwo，保证主要降采样参数相同。但是需要注意，Point-LIO 需要添加额外的随机降采样来保证系统运行在实时状态，而 Point-LIWO 则不需要。这是因为 Point-Lio 需要更多次数的 Update，导致运行时间的更长，这显示了 Point-LIWO 在实时运行上更具优势。

本文采用使机器人完全静止，调整两个模块到完全相同的 CPU 资源占用，对比里程计位移和速度抖动的方式对比 Point-Lio 和无轮速 Point-Liwo 的效果。

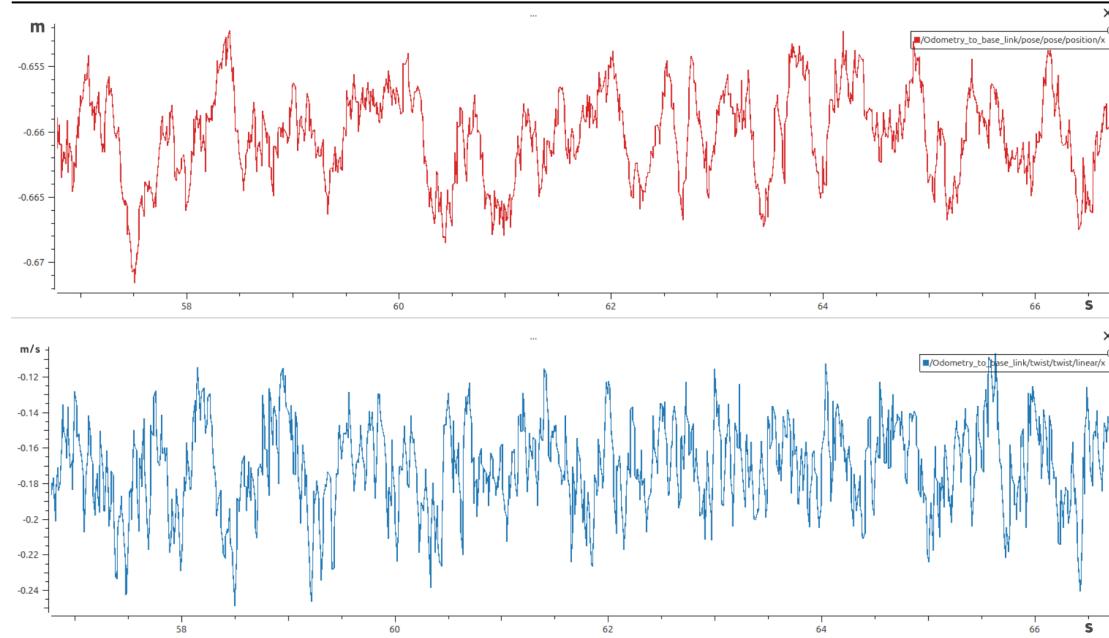


图 4.9 静止下的 Point-LIO 里程计：上曲线为位移曲线，下曲线为速度曲线

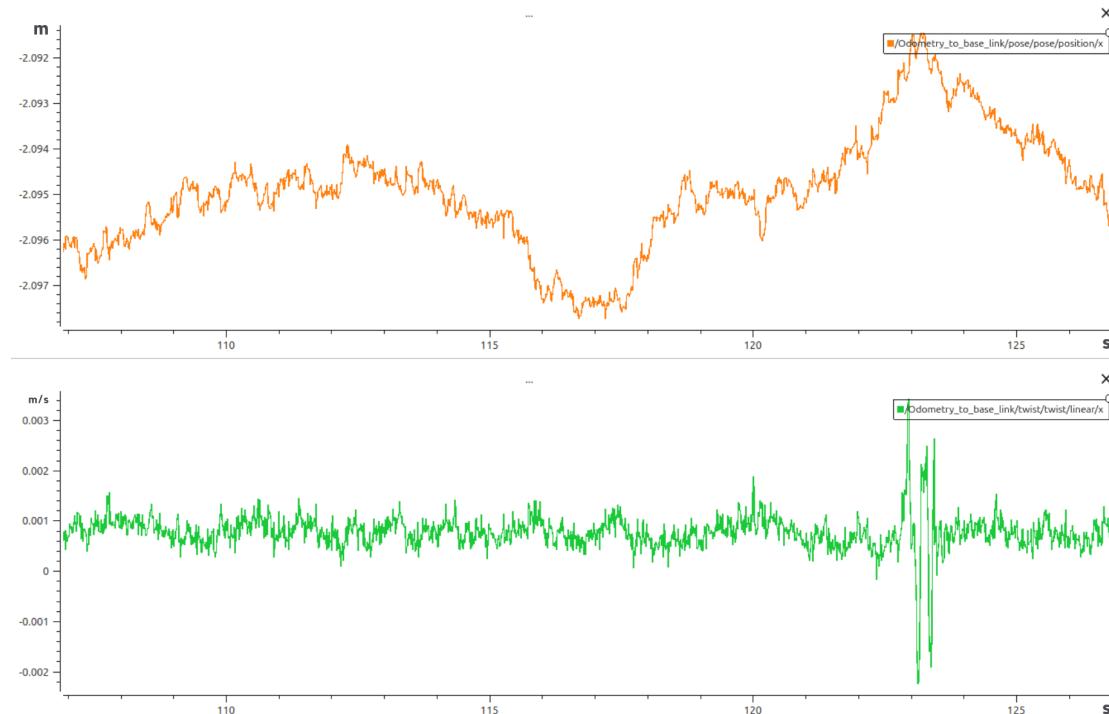


图 4.10 静止下的 Point-LIWO 里程计：上曲线为位移曲线，下曲线为速度曲线

可以看出，Point-LIWO 在更换 Point-LIO 的逐点更新到逐 Batch 更新之后，定位精度从 2cm 抖动范围提高到了 5mm 抖动范围。速度精度从 0.1m/s 抖动且有偏提高到了 0.004m/s 以内无显著偏移。提升与理论分析相同：基于 Batch 更新的 Point-LIWO 有更好的对抗点云噪声的能力，并且因为同时通过多平面构造残差有效降低了建图过程对平面的描述不准确对里程计的影响。

### (3) 高速旋转

本文让机器人以  $10\text{rad/s}$  围绕中心进行旋转。系统为加入了轮速计进行融合 Point-LIWO。激光雷达安装在距离旋转中心  $25\text{cm}$  处，这使得机器人绕旋转中心旋转以  $10\text{rad/s}$  旋转时将在 IMU 上引入  $2.5\text{g}$  的向心加速度（加速度计量程大小为  $4\text{g}$ ）。这样的加速度将给观测器引入正弦的速度变化，对观测器提出了很大挑战。测试实验为在旋转过程中将系统冷启动，通过建图效果、位移、速度估计效果对 Point-LIWO 系统的运行效果进行评定。

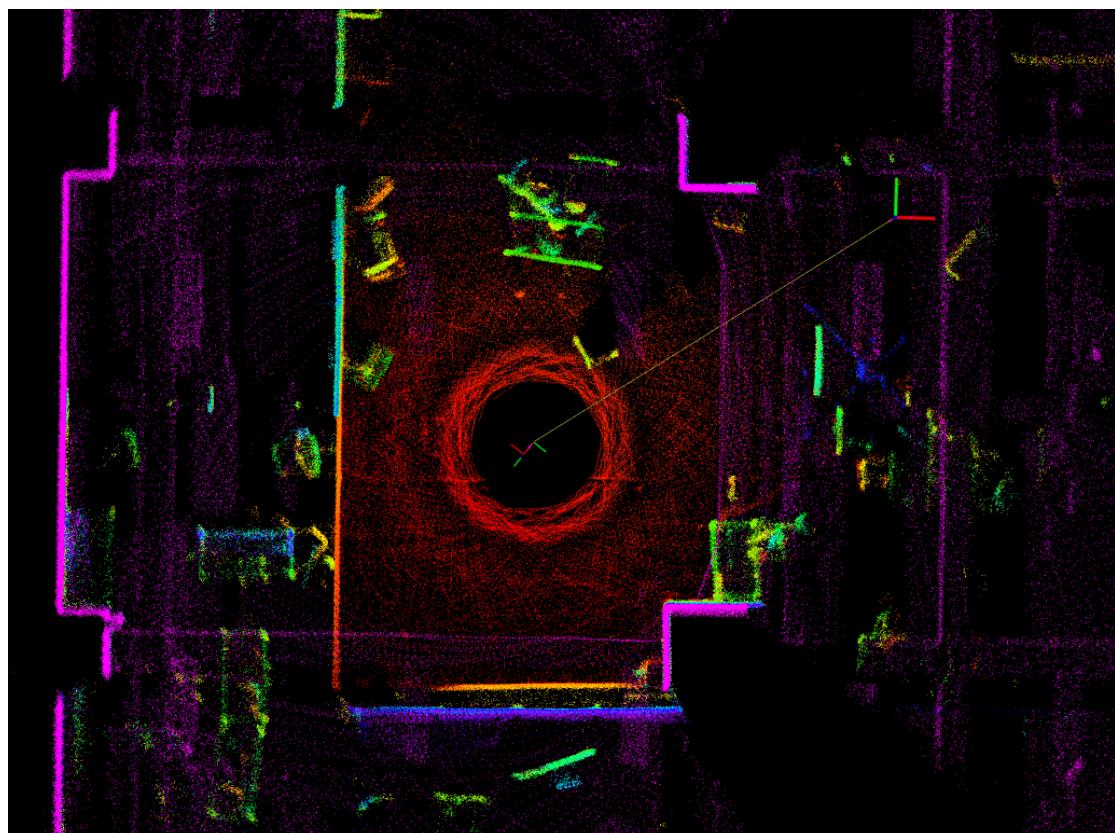


图 4.11  $10\text{rad/s}$  旋转  $20\text{s}$  累计点云

如图4.11所示，旋转建图的地面和墙面描述准确。Point-LIWO 在这样的场景下的实时建图稳定。

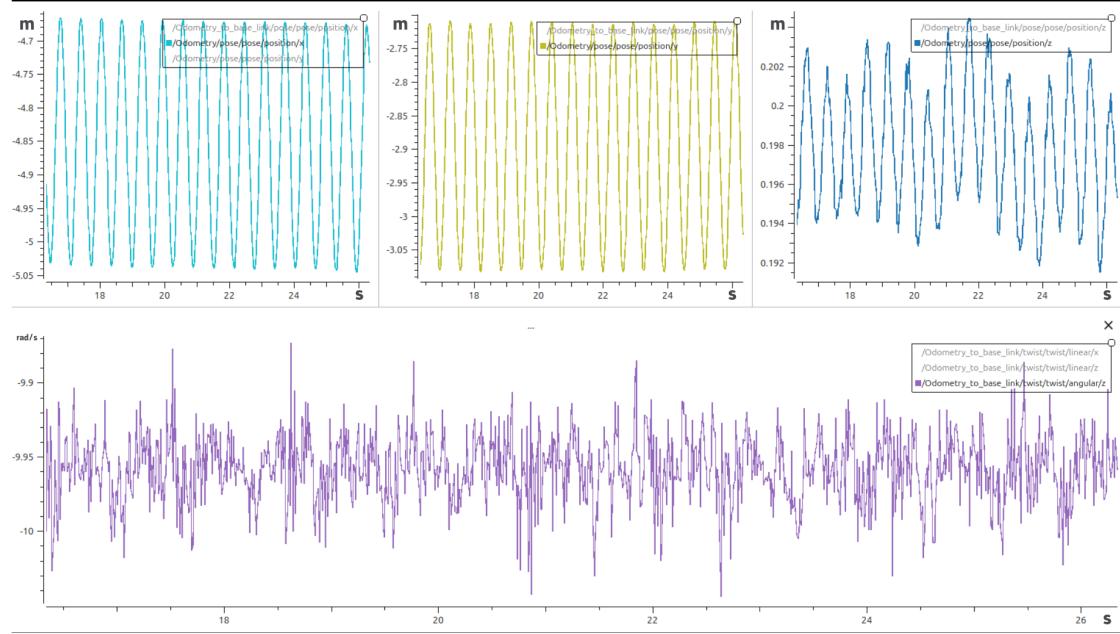


图 4.12 10rad/s 旋转 10s 曲线图：上为 x/y/z 位移，下为绕 z 轴旋转角速度

如图4.12所示，旋转建图时的 x, y 坐标呈正弦变化，符合围绕中心运动的运动模式。同时因为在平面运行，z 高度应该基本不变，如图所示在运行途中的 z 高度变化在 1cm 以内，与之符合。旋转角速度在下方图例中表示，大约为 10rad/s，角速度估计噪声在 0.2rad/s 以内，系统运行稳定准确。

## 第五章 研究总结

本文提出了一种稳健融合轮速的基于 Batch 的紧耦合雷达-惯性-轮速计里程计框架。通过构建在流型上的多传感器紧耦合扩展卡尔曼滤波器，本文创新性地将轮式编码器数据以紧耦合的形式融入激光雷达惯性里程计系统，采用批优化策略在有效提高计算速率的同时降低了雷达观测源噪声带来的影响。本文进行了大量测试并且将其部署于高速旋转，频繁碰撞、打滑的机器人平台，证明了 Point-LIWO 系统在可容忍运动带宽和建图精度方面相比于以往系统的优越性，并且通过对比实验证明了 Point-LIWO 相比于以往方法的性能和精度提升。

## 参 考 文 献

- [1] HE D, XU W, CHEN N, et al. Point-lio: Robust high-bandwidth light detection and ranging inertial odometry[J]. Advanced Intelligent Systems, 2023, 5(7): 2200459.
- [2] ZHANG J, SINGH S. Loam: Lidar odometry and mapping in real-time[C/OL]// Robotics: Science and Systems X. 2015. <http://dx.doi.org/10.15607/rss.2014.x.007>.
- [3] SHAN T, ENGLOT B. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain[C/OL]//2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2018. <http://dx.doi.org/10.1109/iros.2018.8594299>.
- [4] QIN C, YE H, PRANATA C E, et al. Lins: A lidar-inertial state estimator for robust and efficient navigation[C]//2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020: 8899-8906.
- [5] YE H, CHEN Y, LIU M. Tightly coupled 3d lidar inertial odometry and mapping[C/OL]//2019 International Conference on Robotics and Automation (ICRA). 2019: 3144-3150. DOI: 10.1109/ICRA.2019.8793511.
- [6] LI X, RONG H, ZHANG J, et al. Modulating the local coordination environment of single-atom catalysts for enhanced catalytic performance[J]. Nano Research, 2020, 13: 1842-1855.
- [7] LIN J, ZHANG F. Loam livox: A fast, robust, high-precision lidar odometry and mapping package for lidars of small fov[C/OL]//2020 IEEE International Conference on Robotics and Automation (ICRA). 2020. <http://dx.doi.org/10.1109/icra40945.2020.9197440>.
- [8] LI K, LI M, HANEBECK U D. Towards high-performance solid-state-lidar-inertial odometry and mapping[J/OL]. IEEE Robotics and Automation Letters, 2021, 6(3): 5167-5174. DOI: 10.1109/LRA.2021.3070251.
- [9] SHAN T, ENGLOT B, MEYERS D, et al. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping[C/OL]//2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2020: 5135-5142. DOI:

---

10.1109/IROS45743.2020.9341176.

- [10] TAGLIABUE A, TORDESILLAS J, CAI X, et al. Lion: Lidar-inertial observability-aware navigator for vision-denied environments[C]//Experimental robotics: The 17th international symposium. Springer, 2021: 380-390.
- [11] PARK C, MOGHADAM P, KIM S, et al. Elastic lidar fusion: Dense map-centric continuous-time slam[C]//2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018: 1206-1213.
- [12] YUAN C, XU W, LIU X, et al. Efficient and probabilistic adaptive voxel mapping for accurate online lidar odometry[J]. IEEE Robotics and Automation Letters, 2022, 7(3): 8518-8525.
- [13] XU W, ZHANG F. Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter[J/OL]. IEEE Robotics and Automation Letters, 2021, 6(2): 3317-3324. DOI: 10.1109/LRA.2021.3064227.
- [14] XU W, CAI Y, HE D, et al. Fast-lio2: Fast direct lidar-inertial odometry[J/OL]. IEEE Transactions on Robotics, 2022, 38(4): 2053-2073. DOI: 10.1109/TRO.2022.3141876.
- [15] KARIMI M, OELSCH M, STENGEL O, et al. Lola-slam: Low-latency lidar slam using continuous scan slicing[J/OL]. IEEE Robotics and Automation Letters, 2021, 6(2): 2248-2255. DOI: 10.1109/LRA.2021.3060721.
- [16] ZLOT R, BOSSE M. Efficient large-scale three-dimensional mobile mapping for underground mines[J]. Journal of Field Robotics, 2014, 31(5): 758-779.
- [17] KAUL L, ZLOT R, BOSSE M. Continuous-time three-dimensional mapping for micro aerial vehicles with a passively actuated rotating laser scanner[J]. Journal of Field Robotics, 2016, 33(1): 103-132.
- [18] DROESCHEL D, BEHNKE S. Efficient continuous-time slam for 3d lidar-based online mapping[C]//2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018: 5000-5007.
- [19] ANDERSON S, BARFOOT T D. Full steam ahead: Exactly sparse gaussian process regression for batch continuous-time trajectory estimation on  $se(3)$ [C]//2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2015: 157-164.
- [20] LE GENTIL C, VIDAL-CALLEJA T, HUANG S. In2lama: Inertial lidar localisa-

- 
- tion and mapping[C]//2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019: 6388-6394.
- [21] LE GENTIL C, VIDAL-CALLEJA T, HUANG S. In2laama: Inertial lidar localization autocalibration and mapping[J]. IEEE Transactions on Robotics, 2020, 37(1): 275-290.
- [22] ZHANG Y. Lilo: A novel lidar–imu slam system with loop optimization[J]. IEEE Transactions on Aerospace and Electronic Systems, 2021, 58(4): 2649-2659.
- [23] HEMANN G, SINGH S, KAESZ M. Long-range gps-denied aerial inertial navigation with lidar localization[C]//2016 IEEE/RSJ International conference on intelligent robots and systems (IROS). IEEE, 2016: 1659-1666.
- [24] HESCH J A, MIRZAEI F M, MARIOTTINI G L, et al. A laser-aided inertial navigation system (l-ins) for human localization in unknown indoor environments [C]//2010 IEEE International Conference on Robotics and Automation. IEEE, 2010: 5376-5382.
- [25] FORSTER C, CARLONE L, DELLAERT F, et al. On-manifold preintegration for real-time visual–inertial odometry[J]. IEEE Transactions on Robotics, 2016, 33(1): 1-21.
- [26] BAI C, XIAO T, CHEN Y, et al. Faster-lio: Lightweight tightly coupled lidar-inertial odometry using parallel sparse incremental voxels[J]. IEEE Robotics and Automation Letters, 2022, 7(2): 4861-4868.
- [27] HE D, XU W, ZHANG F. Symbolic representation and toolkit development of iterated error-state extended kalman filters on manifolds[J/OL]. IEEE Transactions on Industrial Electronics, 2023, 70(12): 12533-12544. DOI: 10.1109/TIE.2023.3237872.
- [28] LEE M H, LI T H S. Kinematics, dynamics and control design of 4wis4wid mobile robots[J]. The Journal of Engineering, 2015, 2015(1): 6-16.

## 致 谢

在我的本科四年中，收获颇丰，感慨良多。在此，我想向所有支持与帮助我的人们表达衷心的感谢。

首先，我要特别感谢张世武教授和高纬老师的悉心指导。您们的渊博知识和严谨治学的态度，深深影响了我的学术成长。在每一次的交流中，我不仅获得了宝贵的专业知识，也感受到您们对学生的关爱与鼓励。

感谢邵宜阳同学的交流与讨论，你的独到见解和热情分享，让我在研究中受益匪浅。我们一起探讨问题、互相启发，使我对课题的理解更加深入，也增添了许多乐趣。

感谢卢星宇同学在项目部署方面的测试和使用反馈，你的建议和细致的工作让我在项目的推进中少走了许多弯路。

特别感谢薛佳龙学长和其他实验室的师兄师姐们对我论文的指导。你们的建议和反馈让我在写作过程中不断进步，提升了我的研究能力和学术思维。

我也要感谢 RoboWalker 战队为我提供了机器人平台，让我在实践中得以锻炼队伍的团队合作精神和专业能力。此外，感谢战队的刘彦宏学弟实现了机器人控制器，你的数理基础和创造力为我的研究提供了坚实的技术支持。

最后，我要特别感谢我的父母和家人，您们的理解与支持是我前行的最大动力。每一份关心和鼓励都让我在困难时刻坚定信念，也让我在学业上不断追求卓越。

感谢每一位师长、同学和家人，正是你们的陪伴与支持，成就了我今天的成长。我将铭记这段美好的时光，继续努力，不负众望。

2025 年 6 月