



南京大學

本科畢業論文

院 系 計算機科學與技術系

專 業 計算機科學與技術

題 目 基於慣性感知的機器人運動軌迹追蹤機制

年 級 2013 學 號 131220058

學生姓名 葛澤凡

指導教師 謝磊 職 稱 副教授

提交日期

基于惯性感知的机器人运动轨迹追踪机制

Robot Trajectory Tracking Based on Inertial Sensing

南京大学本科生毕业论文（设计、作品）中文摘要

题目：基于惯性感知的机器人运动轨迹追踪机制

院系：计算机科学与技术系

专业：计算机科学与技术

本科生姓名：葛泽凡

指导教师（姓名、职称）：谢磊 副教授

摘要：

随着软硬件技术的发展，智能手机内置的传感器越来越丰富，随之衍生的技术在各领域都有应用。同时，针对智能设备的惯性轨迹追踪技术正在蓬勃发展。通过对设备运动轨迹的追踪，以达到惯性导航、空间运动监测和室内定位等目的。那么，基于智能手机的惯性传感器，亦可实现对智能手机的轨迹追踪，这在室内定位、行为感知和人机交互等领域有着重要意义。

本文结合智能手机的惯性感知功能与作为运动载体的拥有良好平面运动特征的智能扫地机器人，提出了一种基于惯性感知的机器人运动轨迹追踪机制。首先分析了机器人平面运动的特征，构建了轨迹追踪的数学计算模型。接着，给出了完整的基于惯性感知的机器人运动轨迹追踪机制，内容包括：实时获取智能手机加速度传感器、陀螺仪、重力传感器和磁力计的数据，对它们进行处理与优化，并将它们应用于运动的转角计算和位移计算；对机器人运动进行状态的分类与判别，并基于分类对运动数据进一步处理；结合上述内容，实时计算出机器人运动轨迹。最后，基于安卓系统的智能手机进行了系统实现，通过实验验证了本文提出的机器人运动轨迹追踪机制的有效性。

本文主要贡献如下：

1. 构建了针对平面运动机器人的运动轨迹追踪的计算模型，并提出了应用四元数优化转角计算的方法，优化效果显著。
2. 对机器人运动状态进行分类，提出对应判别算法。判别算法将为整套轨迹追踪机制提供数据处理依据，以进行特定区间数据的放缩和速度较准。
3. 提出了对加速度信号异常区间的削弱填补方案。这是一种针对传感器在

锯齿波输入下产生异常输出的处理办法,解决了由于加速度数据异常导致的速度计算偏差。

4. 给出了完整的基于惯性感知的机器人轨迹追踪机制的系统实现。在一分钟左右的机器人自由运动实验中,计算轨迹和实际轨迹的绝对路程平均相对偏差为 16.2%, 计算转角与实际转角的平均累计偏差为 2.1%。

关键词：惯性感知；轨迹追踪；智能手机；机器人运动

南京大学本科生毕业论文（设计、作品）英文摘要

THESIS: Robot Trajectory Tracking Based on Inertial Sensing

DEPARTMENT: Department of Computer Science and Technology

SPECIALIZATION: Computer Science and Technology

UNDERGRADUATE: Zefan Ge

MENTOR: Associate Professor. Lei Xie

ABSTRACT:

With the development of software and hardware technology, more and more sensors are embedded in the smart phones, and the technology has been applied in many fields. Meanwhile, the inertial trajectory tracking technology for intelligent devices is developing rapidly. Tracking the trajectory of the device can achieve the purpose of inertial navigation, space motion monitoring and indoor positioning. Based on the inertial sensors, trajectory tracking for smart phones can also be realized, which is important in the field of indoor location, behavior perception and human-computer interaction.

With the inertial sensing function of smart phones and an intelligent sweeping robot as a moving carrier, which has plane motion characteristics, this paper proposes a mechanism of robot trajectory tracking based on inertial sensing. Firstly, this paper analyses the features of the robot's plane motion and constructs a mathematical model of trajectory tracking. Then, the mechanism of robot trajectory tracking based on inertial sensing is proposed. The contents include obtaining the data of the accelerometer, gyroscope, gravity sensor and magnetometer, processing and optimizing them, and applying them to the calculation of the displacement and the angle of rotation and the classification and discrimination of the states of the robot movement and the further processing of the motion data. In combination with the above contents, the system calculates the trajectory in real time. At last, we develop an application of the robot trajectory tracking and verify the efficiency of the mechanism by the experiment.

The main contribution in this paper are as follows:

1. We construct a computational model of trajectory tracking for planar motion robot and propose a method to optimize angle calculation by using quaternion, which has remarkable effect.

2. We classify the states of robot movement and propose the discrimination algorithm. The discrimination algorithm provides the data processing basis for the trajectory tracking mechanism, so we can reduce the acceleration data in special interval and calibrate the speed.

3. We propose a method to weaken the acceleration signal in abnormal intervals. The method deals with the sensor's abnormal output with the sawtooth wave as input, and can solve the problem of the bias in speed calculation caused by abnormal acceleration data.

4. We have implemented a complete system of the robot trajectory tracking mechanism. In the experiment of robot free movement in a minute or so, the average relative deviation of the absolute distance between the calculated trajectory and the actual trajectory is 16.2%, and the average accumulative deviation between the rotation angle and the actual rotation angle is 2.1%.

KEY WORDS: Inertial Sensing; Trace tracking; Smart Phone; Robot Movement

目录

1	绪论	1
1.1	研究背景	1
1.2	研究目的与意义	1
1.3	研究内容与本文工作	2
1.4	本文结构	3
2	相关工作	4
2.1	惯性轨迹追踪	4
2.2	智能手机传感器	5
2.2.1	加速度传感器	5
2.2.2	陀螺仪	6
2.2.3	重力传感器	6
2.2.4	磁力计	7
2.3	智能手机的感知与识别	7
2.4	本章小结	8
3	机器人运动轨迹计算模型	9
3.1	机器人运动分析	9
3.2	计算模型设计	10
3.2.1	轨迹追踪计算模型	10
3.2.2	应用四元数优化转角计算	13
3.3	实际问题与误差分析	15
3.4	本章小结	16
4	基于惯性感知的机器人运动轨迹追踪机制	17
4.1	惯性感知数据的处理与应用	17
4.1.1	加速度信号初步处理	17
4.1.2	陀螺仪信号的处理与使用	19
4.1.3	应用磁力计参与转角计算	21
4.2	运动状态判别	22
4.2.1	状态判别的准备	23

4.2.2	状态分类与判别方法.....	24
4.3	加速度的异常区间填补.....	27
4.4	速度校零.....	31
4.5	轨迹追踪机制综述.....	32
4.6	本章小结.....	33
5	系统实现与实验分析	34
5.1	开发环境与平台.....	34
5.2	系统设计.....	35
5.2.1	多模块设计.....	35
5.2.2	界面与运行.....	37
5.3	实验与分析.....	38
5.4	本章小结.....	43
6	总结与展望	44
6.1	总结.....	44
6.2	展望.....	44
	参考文献	45
	致谢	47

1 绪论

1.1 研究背景

如今，以智能手机为代表的智能设备已成为人们日常生活必不可少的工具，而衍生的应用更是五花八门，涉足了人类生活的各个领域。在这些应用中，使用智能设备所携带的各类传感器进行智能感知与识别的应用是新兴的翘楚。

在丰富的传感器中，有部分可将它们归类为惯性感知传感器，包括加速度传感器，陀螺仪，重力传感器等，他们在惯性导航、行为识别、交互游戏等方面有着广泛应用。

与此同时，针对智能设备的轨迹追踪（轨迹跟踪）的研究正在蓬勃发展。在室外，可以利用 GPS（Global Positioning System，全球定位系统）实现相当精准的设备运动轨迹跟踪；而在室内环境，由于卫星信号被建筑物遮蔽或削弱等问题，无法使用卫星定位来确定当前位置，故欲实现室内轨迹追踪，将依赖于其他技术，如无线传感技术、射频识别技术及红外传感技术等。

基于智能设备的惯性感知传感器进行室内轨迹追踪的技术研究目前还处在初步发展阶段，基于智能手机的惯性传感器直接进行运动轨迹追踪的工作也是相当之少，但我们可以通过对其的研究，探索惯性感知传感器数据的特点与处理方法，发掘轨迹追踪技术的创新点。实现对智能手机的轨迹追踪，这在室内定位、行为感知和人机交互等领域有着重要意义。

1.2 研究目的与意义

本文工作的研究的根本目的是给出一种基于惯性感知的智能设备运动轨迹追踪机制，但由于单纯的运动概念过于庞大且空泛，我们需为研究对象加上一个运动模式的约束。本文工作选择使用平面运动的机器人搭载智能手机运动，机器人的运动有客观规律可循，这有利于惯性感知和轨迹追踪的研究的循序渐进。于是本文工作对象确立为机器人与手机共同的运动，研究目的即为给出基于惯性感知的机器人运动轨迹追踪机制。

在本文工作中,首先将分析机器人运动特征,为其构建用于计算的数学模型,为整个研究工作奠定基础。基于计算模型,使用惯性感知的数据对模型进行实现,并在智能设备上系统进行实现与实验。

本文工作将主要面临以下两个具有挑战性的问题:

1. 对传感器信号的积分计算存在较大偏差。在各类应用中,基于惯性测量相关传感器信号进行积分运算结果与实际结果都有很大偏差,这已成为业内普遍承认的客观存在。本文将涉及对传感器信号的二次积分运算,其理论上存在的误差便已不容小觑。对此,本文将给出减小误差的解决方案,提出较为精准的转角计算方法,并进行一些工程性的处理。

2. 运动状态的识别。与行为识别类似,在本文研究工作中,需要对机器人的运动状态做识别与针对性处理。对此,本文将给出状态分类与具有容错能力的判别方法。

关于上述问题的解决之道和应对之策会在正文部分详尽阐述,解决轨迹追踪机制中存在的问题也是本文工作的一大目标,这具有开拓性的意义。

1.3 研究内容与本文工作

首先,本文工作对机器人运动原理进行探究,展开了运动学层面的分析,并给出了机器人运动轨迹追踪的计算模型。其次,通过加速度计、陀螺仪、重力传感器和磁力计等惯性感知传感器获取实时数据,并对数据进行处理,继而进行转角计算和位移计算。同时,将给出运动状态的判别,基于运动特征的工程性处理等操作。通过系列处理,最终实时计算出机器人平面运动轨迹,形成完整的运动轨迹追踪机制。最后,研究工作中还为提出的机器人运动轨迹追踪机制完成了系统实现,并通过实验验证了机制的有效性。

本文工作主要做出了如下贡献:

1. 构建了针对平面运动机器人的运动轨迹追踪的计算模型,该计算模型可扩展性较强。提出了应用四元数优化转角计算的方法,优化效果显著。

2. 对机器人运动状态进行分类,并提出对应判别算法。判别算法将为整套轨迹追踪机制提供数据处理依据,以进行特定区间数据的放缩和速度较准。

3. 提出了对加速度信号异常区间的削弱填补方案。这是一种针对压电传感

器在锯齿波输入下产生异常输出的处理办法,解决了由于加速度数据异常导致的速度计算偏差。

4. 给出了完整的基于惯性感知的机器人轨迹追踪机制的系统实现。最终实验中,对于一分钟左右的机器人自由运动,计算轨迹和实际轨迹的绝对路程相对偏差在 10%到 25%之间,平均为 16.2%;计算转角与实际转角的累计偏差在 1.5%到 3.5%之间,平均为 2.1%。

1.4 本文结构

本文的研究内容是基于惯性感知的机器人运动轨迹追踪机制,全文共分为六章,内容与组织结构如下:

第一章绪论。本章从课题研究背景出发,对相关背景进行介绍,并对研究目的和意义进行了简要说明,给出了面临的挑战性问题的应对思路。并对本文工作做了概括性介绍,给出了几个贡献点。

第二章相关工作。首先阐述当前惯性轨迹追踪及相关追踪技术,简单介绍研究现状。然后对智能手机的多模态传感器进行介绍,重点介绍了本文工作涉及的四种传感器。最后对智能手机的感知识别相关研究工作做了相关阐述。

第三章机器人运动轨迹计算模型。基于对机器人运动模式的分析,给出了轨迹追踪的计算模型。在计算模型中又提出了应用四元数计算旋转角度的方法,对通用方案进行了优化。最后讨论了一些运动实际问题并进行了理论上的误差分析。

第四章基于惯性感知的机器人运动轨迹追踪机制。本章首先介绍了相关传感器的基本处理方式,提出了使用磁力计数据参与辅助计算转角的方案。其次对运动状态的分类与判别算法进行了详细描述。最后,给出对加速度异常区间的工程性处理办法和速度校零方案,并给出机制综述。

第五章系统实现与实验分析。本章首先介绍了开发环境和硬件条件,接着给出了系统设计的介绍,阐述了系统各模块的作用和协同方式,介绍界面与运行流程。最后,给出了实验数据和结果分析。

第六章总结与展望。对本文的基于惯性感知的机器人运动轨迹追踪机制进行总结,并对将智能手机惯性感知应用于轨迹追踪的未来工作进行展望。

2 相关工作

2.1 惯性轨迹追踪

惯性轨迹追踪，是基于惯性测量单元的轨迹追踪技术，一般应用于对目标对象在三维空间中的运动轨迹进行跟踪，同时可参与惯性导航、空间动作监测与识别等相关技术工作。

在轨迹追踪的各项技术中，广为人知的由有 GPS 全球定位系统，这项技术已深入全人类的工作生活。其他比较成熟的追踪技术有磁空间追踪技术[1, 2]，此项技术原理如下，设立磁场发射源向空间中发射磁场，再使用配有三维正交线圈的接收器在磁场的有效作用范围内运动产生电磁感应，基于三轴磁场强度计算相对于发射源的空间位置从而感知在三维空间内的运动。另一种较为精准的追踪技术为激光追踪[3]，此项技术在探测源与目标对象之间发射激光，以此进行距离测量，通过捕捉目标对象的实时的距离，来计算位移数据，从而得到空间运动轨迹。

这些技术也存在着相关问题，如磁空间追踪技术易受外界磁场干扰或是被环境阻隔屏蔽，激光追踪技术则受光照范围限制。而基于惯性感知传感器的惯性轨迹追踪技术虽然在测算精度上无法向上述技术看齐，但却有着可测量空间广阔，不受外界参考系控制的优点。

目前惯性追踪技术绝大部分是基于 MEMS（Micro-electromechanical Systems，微型机电系统）的惯性传感器来研究开发[4, 5, 6, 7]，使用 MEMS 加速度传感器、MEMS 陀螺仪是惯性轨迹追踪技术的基础。因为测量精度受制，惯性追踪技术在精度要求极高的领域还鲜有单独的应用，不过经过近年来的发展，其在空间动作检测和模式识别领域已经挑起大梁。如在虚拟现实和人机交互方面，惯性追踪技术被用来实现三维空间鼠标及三维空间画笔[8, 9]；在人体动作识别方面，又可被用来实现可穿戴键盘、虚拟手套等；而其在工程测量领域则已经大放异彩，空间运动监测、滑动监测、惯性导航等无不有突出贡献。

利用智能手机内置的惯性测量传感器进行轨迹追踪的工作目前还相对较少，其面临的问题有内置传感器精度不高，难以在硬件层面做出改良优化，难以在微电路层面做出系统优化等。故目前智能手机内置的惯性测量传感器大部分还是应

用在感知识别领域,基于测量的惯性数据,可以做出动作识别、状态识别等应用。下面对智能手机的传感器及其相关应用进行介绍。

2.2 智能手机传感器

现如今,智能手机与硬件技术的协同发展有目共睹,无论是 iOS 系统还是 Android 系统的智能手机,其内置传感器已愈加丰富。这些传感器的集成亦为智能手机的发展带来更多可能。

智能手机内置的传感器有硬件实现和软件实现之分[10],硬件传感器是指物理层实际存在的传感器,可以将物理信息转换为可用信号,如加速度计磁力计等;软件传感器则是综合了其他传感器的数据,经过了软件层的处理所得,如方向传感器。按照所测量内容分类,则有:运动学量、力学量、光学量、磁学量、热学量等。在本文工作中,主要使用了测量运动学量的加速度传感器、陀螺仪、重力传感器以及测量磁学量的磁力计。前两者又可以合称为惯性测量单元。下面将简单介绍此四种传感器。

2.2.1 加速度传感器

加速度传感器是用于测量设备在运动中的加速度的传感器,其理论基础为牛顿第二定律。智能手机的惯性测量传感器坐标系如图 2-1 所示,与常识中的地球坐标系相异:x 轴与手机的宽边平行,y 轴与长边平行,z 轴则与手机平面垂直。当手机沿着各轴正方向加速运动时,传感器测得数值为正。

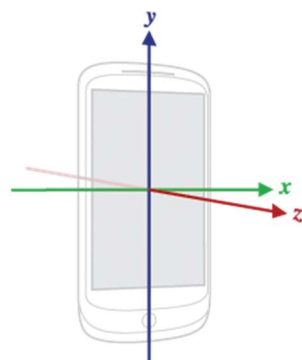


图 2-1 惯性感知相关的传感器坐标系

值得注意的是，加速度传感器所测得数据包含了地球引力对设备的作用，也即在理想状态下，当我们将手机正面朝上，水平置于地平面时， z 轴数据即为当地重力加速度。目前大部分的智能手机还提供了线性加速度传感器，此传感器经系统处理（一般会使用低通滤波），滤除重力影响，得到设备线性运动的三轴加速度数值。智能手机对加速度传感器应用很广，如运动软件的计步器，重力感应游戏，微信的“摇一摇”功能等。

2.2.2 陀螺仪

陀螺仪（Gyroscope）又称角速度传感器，其坐标系与图 2-1 一致，它可以测出设备当前旋转的角速度。根据陀螺仪获取的实时数据，我们可以计算出设备在一段时间内旋转过的角度，即转角。本文工作主要关注二维平面内的旋转，也即 x - y 平面内（绕 z 轴）的旋转。

陀螺仪的应用极为广泛，其应用主要有以下几类：

1) 配合 GPS 进行导航。在用户遇到建筑物阻挡或者进入隧道等情况时，陀螺仪可用以测量设备的偏航与直线位移，加上地图本身的信息，即可实现继续导航。

2) 实现快捷控制。利用手机轻微的转动进行对用户动作的感应，可以晃动手机实现快捷控制，例如撤销输入。

3) 摄影抗抖动。陀螺仪与手机摄像头配合，防止由于手部的抖动干扰拍照而影响成像质量。用户在按下快门时，陀螺仪记录抖动，并将手机的抖动反馈给图像处理器，再进行分析处理，使得手机捕捉到的画面更稳定清晰。

4) 感知动作，可应用于体感游戏。陀螺仪对三维空间内的手机转动动作的感知弥补了加速度传感器的缺憾，两者配合可以重构出完整动作，这在体感互动类游戏开发中被使用得淋漓尽致。

2.2.3 重力传感器

重力传感器又称重力感应器。智能设备的重力传感器既有硬件实现也有软件实现。硬件实现一般时利用压电效应与敏感元件直接感知地球引力，而软件实现则是通过对加速度传感器数据的滤波等处理（如卡尔曼滤波器），滤出重力数据。

重力传感器的坐标系和前两者相同，即如图 2-1 所示。

目前绝大多数智能手机和平板电脑都内置了重力传感器，其应用也很广。手机的横竖状态切换利用重力传感器便能实现屏幕的变化，手机左右前后摇晃也能代替方向控制应用于赛车类游戏，迷宫类游戏等。

2.2.4 磁力计

磁力计，又称高斯计、磁强计。磁力计用于侦测磁场，虽然测量的时磁学量，但它的坐标系与惯性感知坐标系一致，即图 2-1，它测得数据对应于手机在三个坐标轴上受到的磁场强度。

智能手机已普遍配有磁力计，其应用也极为普遍，最基本的手机自带指南针便是其中之一。磁力计感知地磁，则顺水推舟的亦可应用于设备的方向定位，事实上，结合磁力计与加速度传感器的数据，便能计算出设备的方位角。本文工作也需要这样的功能。Android 系统向开发者提供了相关功能接口，这在第四章中将有涉及。

2.3 智能手机的感知与识别

智能手机内置了丰富的传感器，基于这些传感器衍生出各类应用技术，其应用种类之广，应用范围之大，应用功能之强无不令人咋舌称赞。这些应用之中，跟感知与识别相关的尤其突出。智能手机能感知周遭环境，提供所处环境的各项物理数据，也可以对用户的行为动作进行感知，对这些行为进行数学建模和即时监测以做出识别[11, 12]。

在行为识别中，加速度传感器的使用至关重要，如 Kwapisz 等人[13]的一项研究工作中，对标记过的加速度数据识别一个时间片内的用户动作，基于机器学习方法，可以对站立、坐下、行走、奔跑、上楼、下楼进行分类识别，再将几个时间片内的识别结果综合以推算的用户此段时间是在进行什么活动。同时，在长时间的识别后，根据用户在一天甚至一周内的活动整理分析，用以判断用户日常生活习惯是否健康，是否经常锻炼等。相似的，在 Sun 等人[14]的论文中，通过加速度传感器识别了用户平日里的活动，判断是否有久坐习惯，适时地给出运动建议。

综合各类传感器，智能手机的感知识别应用将更加强大，如在使用惯性感知传感器记录手机运动状态的基础上结合光线传感器和麦克风来记录手机周围的光线和声音环境，可以对用户的睡眠进行监测[15]。又如根据各类传感器感知用户行为如走路模式、屏幕触碰等的生物性特征，来进行手机当前使用者的身份验证[16]。

2.4 本章小结

本章首先对轨迹追踪相关技术进行了介绍，比较了惯性追踪技术和其他追踪技术。接着对智能手机内置传感器进行介绍，着重介绍了本文工作中使用到的四项传感器。最后对智能手机基于各类传感器在感知识别领域的技术应用与研究工作进行了介绍。

3 机器人运动轨迹计算模型

本文工作所使用的运动载体为 iRobot Create2 可编程机器人[17]（下简称 Create2），由美国 iRobot 公司设计生产，基本运作模式同该公司生产的 Roomba 扫地机器人一致，一般在水平地面运动，其外形如图 3-1 所示。



图 3-1 iRobot Create2 可编程机器人俯视图

下面阐述基于 Create2 机器人的运动模式建立计算模型，并进行相关分析。

3.1 机器人运动分析

Create2 机器人由底面中部的左右两轮驱动，两轮可独立向前或向后运转，前部有一个导向平衡轮，无动力作用。其运动可抽象为二维平面的运动，本质上来说，它仅有两种基础运动方式：

1. 左右两轮相同转速向前运动。理想状态下，机器人启动之后，即进行匀速直线运动。
2. 左右两轮不同转速进行运动。若两者转向相反，则会达到绕两者间某点自旋转的效果，如两者转速一致，左轮向前，右轮向后，则相当于绕机器人中心做顺时针自旋转；当两者转向相同，则绕两者连线延长线上某点进行圆周运动，若只完成一部分圆弧运动，则相当于转弯。

基于上述两种基础运动方式，Create2 机器人组合出了两种运动模式：

1. 绕某点半径不断扩大的螺旋运动。在机器人自主运转时，此种模式会出现于断电开启后的第一次运动。
2. 探路型运动。一般会进行匀速直线运动，在发生碰撞或一定时间不遇到阻碍物时，会进行绕中心的自旋转，来改变前进方向，继而继续探路运动。在自旋转阶段，也会因地制宜或者随机产生转弯，来改变下一段直线运动的方向。

根据上述两种运动模式，我们将构建运动轨迹的数学计算模型。

3.2 计算模型设计

3.2.1 轨迹追踪计算模型

理论上，机器人从启动到停止的运动轨迹计算模型是十分直观的，若我们得到机器人质点相对于全局坐标系的实时的加速度连续信号，便能通过运动学公式（如式 3.1）积分计算得到位移。

$$\vec{S}(t) = \int \left(\int \vec{a}(t) dt \right) dt \quad (3.1)$$

我们基于智能手机的惯性感知单元，仅能获得离散信号，例如从智能手机的加速度传感器获得的是相对于自身坐标系的实时离散信号，参见 2.2.1 节。当惯性感知数据采样频率足够高，根据微积分思想，即可以模仿连续信号。将智能手机搭载固定于机器人中心位置，将两者看作整体，基于智能手机的加速度信号和陀螺仪信号，我们可以得到 Create2 机器人运动（也即手机运动）的实时加速度 a_i 和角速度 ω_i ，继而可以得到在平面运动 T 时刻的路程 S （轨迹长度）和转角 θ ，如式 3.2 与式 3.3。

$$S_T = \sum_{t=0}^T \left(\sum_{i=0}^t a_i \Delta t \right) \Delta t \quad (3.2)$$

$$\theta_T = \sum_{t=0}^T \omega_t \Delta t \quad (3.3)$$

下面我们将重点放在对位移本身的计算上，也即根据可获得的离散惯性感知信号，给出轨迹追踪计算模型。我们以轨迹追踪开始时，机器人的启动位置为原点，构建一个全局坐标系，也即其起始前进方向为 Y 轴正方向， X 轴则为自身中

心横向轴线，正方向向右。坐标系见图 3-2。

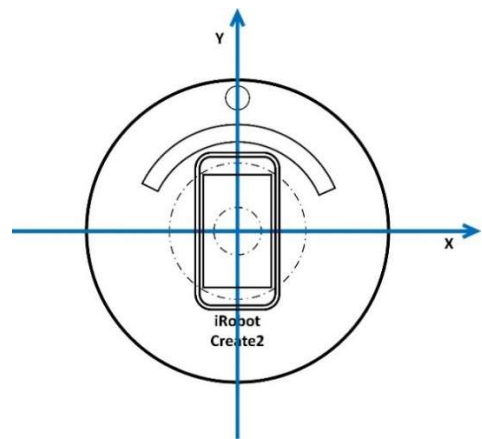


图 3-2 轨迹追踪计算模型坐标系

实际轨迹追踪实验时，手机固定位置与机器人中心重合，手机正面向上，自身 Y 轴与机器人 Y 轴重合，也即机器人运动速度方向永远与手机的 Y 轴重合。这样，我们在实际模型中着重关注 Y 轴加速度数据。下图 3-3 为机器人运动轨迹的几何模型图，结合它我们给出计算模型的工作流程。

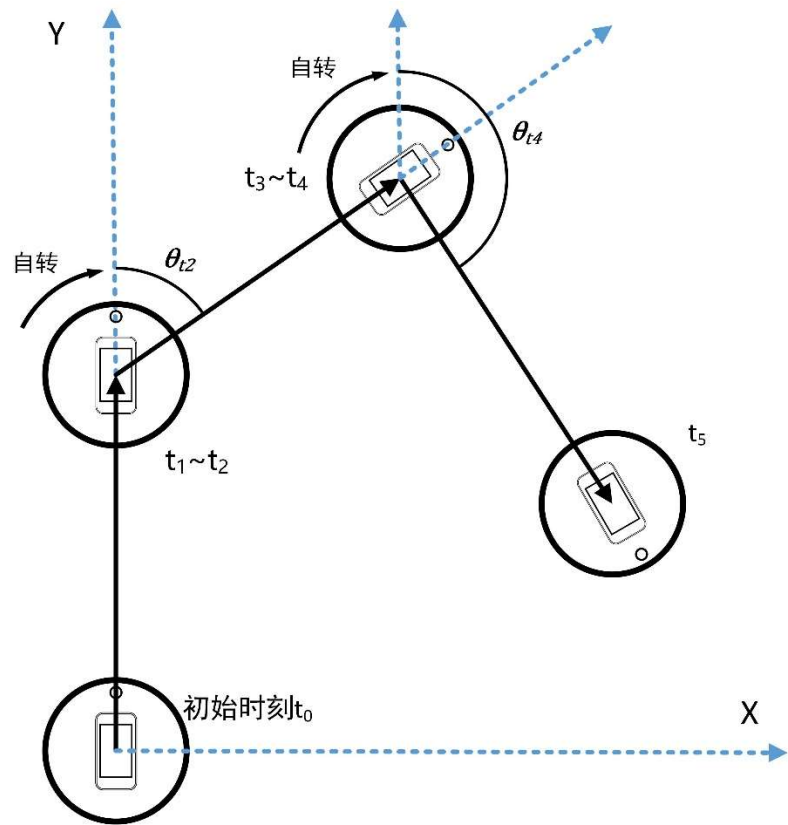


图 3-3 运动轨迹几何模型示意图

本文工作使用的轨迹计算模型是一个迭代过程，采用梯形积分方法，目标是不断得到当前时刻的位移量 (x_t, y_t) 以进行追踪轨迹，记当前时刻为 t ，前一时刻为 $t-1$ ，其算法流程表述如下：

重复：

步骤1： 计算当前时刻的转角 θ 。关于转角计算，传统方法是对陀螺仪的绕 Z 轴角速度信号积分获得，但我们将使用一种更精准稳定的方法计算转角，在此不展开，详见 3.2.2 节。

步骤2： 将当前时刻 t 的前进方向加速度信号作为输入 a_{input} ，根据当前转角 θ_t （如图 3-3 中的 θ_{t2} ， θ_{t4} ）分解到坐标系 X 轴与 Y 轴，得到当前时刻全局坐标系的加速度：

$$\begin{aligned} a_x[t] &= a_{input} \sin \theta_t \\ a_y[t] &= a_{input} \cos \theta_t \end{aligned} \quad (3.4)$$

步骤3： 对当前时刻 t 的速度分量进行梯形积分计算，公式如下：

$$\begin{aligned} v_x[t] &= v_x[t-1] + \frac{1}{2}(a_x[t-1] + a_x[t])\Delta t \\ v_y[t] &= v_y[t-1] + \frac{1}{2}(a_y[t-1] + a_y[t])\Delta t \end{aligned} \quad (3.5)$$

步骤4： 若非第一种运动模式螺旋运动，则直接进入步骤 5，否则，由于机器人的旋转在时间切片内可视为匀速圆周运动，线速度方向会发生改变，我们需要将速度再投射到新的转角上，参见示意图 3-4 如下，图中将运动轨迹上的两个时刻的机器人位置重合到一处：

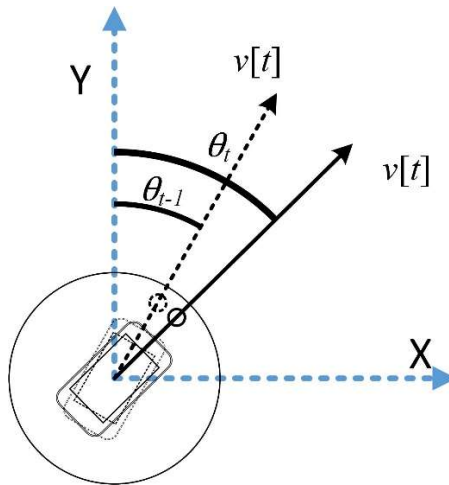


图 3-4 圆周运动致使速度方向改变

速度方向调整使用的公式如下：

$$v_x[t] = v_x[t] \frac{\sin \theta_t}{\sin \theta_{t-1}} \quad v_y[t] = v_y[t] \frac{\cos \theta_t}{\cos \theta_{t-1}} \quad (3.6)$$

步骤5： 对当前时刻 t 的 X 轴与 Y 轴位移分量进行梯形积分计算，得到当前位移量。公式如下：

$$\begin{aligned} x_t &= x_{t-1} + \frac{1}{2}(v_x[t-1] + v_x[t])\Delta t \\ y_t &= y_{t-1} + \frac{1}{2}(v_y[t-1] + v_y[t])\Delta t \end{aligned} \quad (3.7)$$

直到 轨迹追踪停止。

以上便是轨迹追踪计算模型的算法描述，而基于智能手机的惯性感知相关优化与实际运动中的特殊处理将在下一章阐述。

3.2.2 应用四元数优化转角计算

用对陀螺仪数据进行梯形积分的方法来计算转角将会产生很大误差，于是这里引入应用四元数，得到更精准的方位角计算方法，这是本文工作基础部分的一大重点。

首先对四元数进行简单介绍，它是一种简单的超复数，可用于表示三维空间中的旋转。四元数由一个实数加上三个虚数单位 i, j, k 组成，满足下列关系：

$$\begin{aligned} i^2 + j^2 + k^2 &= -1 \\ ij &= k \quad jk = i \quad ki = j \end{aligned}$$

四元数 q 表示为：

$$q = (x, y, z, w) = xi + yj + zk + w$$

式中 x, y, z, w 为实数。根据坐标系三轴的叉乘性质，四元数可以被表示为向量与实数相组合的形式，如下：

$$q = ((x, y, z), w)$$

那么，绕单位向量表示的轴 (x, y, z) 旋转角度 θ ，四元数如下表示：

$$q = \left((x, y, z) \sin \frac{\theta}{2}, \cos \frac{\theta}{2} \right) \quad (3.8)$$

刚体坐标系中的一个点 p 经过旋转后得到：

$$p' = qpq^{-1}$$

以上便是四元数的简单介绍。对比其他表示三维空间旋转的方法，如旋转矩阵和欧拉角，四元数有下列几个优点：

1. 可快速连接，四元数的叉乘能将角位移序列转换为单个角位移，这对本文工作即为重要，较之使用旋转矩阵做相同操作，则效率明显降低。
2. 四元数共轭可用于计算角位移，虽然通过转置矩阵亦能殊途同归，但不如四元数来得简单自然。
3. 能和矩阵形式快速转换，相比欧拉角与矩阵之间的转换效率更高。
4. 表示数据量小，仅使用 4 个参数，而旋转矩阵需要 9 个。

旋转矩阵在实际运算方面要付出更大代价，而表示更简单的欧拉角则要面对万向节死锁等可能发生的问题。当然，四元数也有它的弊端，它可能不合法，不合理的数据或对浮点数的舍入累积都可能致使四元数不合法。同时，四元数也不是很直观，这导致了它难于使用。

下面介绍应用四元数对累积转角的计算方法。基于智能手机的惯性感知数据获取当前初始转角的四元数 q_0 ，其方法将在下一章中介绍。上一节中，图 3-4 可作为原地旋转的几何示意图，下面给出前进中发生微小偏转的计算模型几何示意图，如图 3-5。

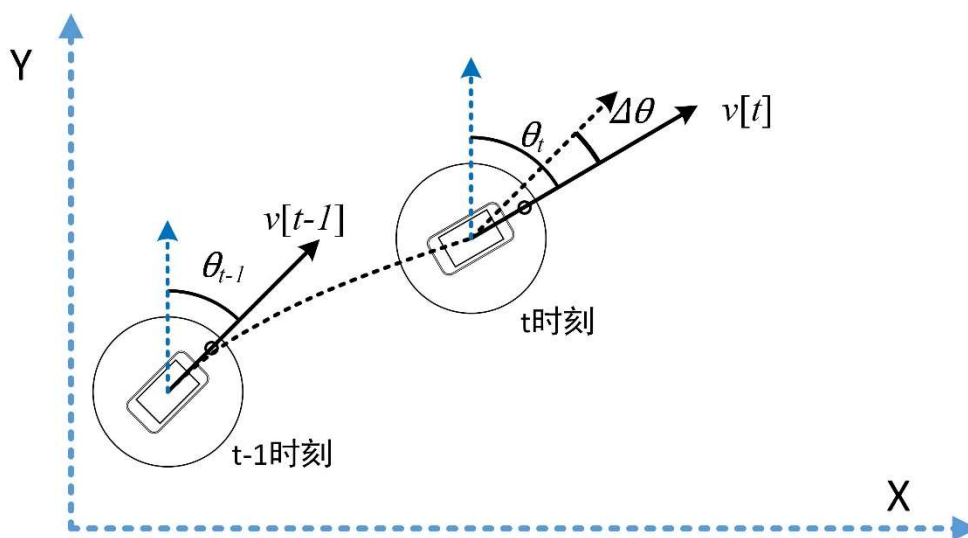


图 3-5 前进中发生微小偏转的情况

在初始化后，进入迭代过程，算法如下：

重复：

步骤1： 计算当前时刻 t 的陀螺仪三轴角速度 ω 的模，并进行单位化：

$$|\omega| = \sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2} \quad (3.9)$$

$$\omega_x = \frac{\omega_x}{|\omega|} \quad \omega_y = \frac{\omega_y}{|\omega|} \quad \omega_z = \frac{\omega_z}{|\omega|}$$

步骤2： 计算当前时刻与上一时刻内的转角 $\Delta\theta$, 即为图 4-5 中 t 时刻机器人运动对应的 $\Delta\theta$ 公式如下：

$$\Delta\theta = |\omega|\Delta t \quad (3.10)$$

步骤3： 根据式 3.8 我们得到 $\Delta\theta$ 的四元数：

$$q_{\Delta\theta} = \left((\omega_x, \omega_y, \omega_z) \sin \frac{\Delta\theta}{2}, \cos \frac{\Delta\theta}{2} \right) \quad (3.11)$$

步骤4： 计算当前时刻 t 的累积转角，这将使用上一时刻的四元数：

$$q_t = q_{t-1} q_{\Delta\theta} \quad (3.12)$$

步骤5： 基于式 3.8 求出当前的转角四元数对应转角值 θ 。

直到 轨迹追踪停止。

以上便是应用四元数求转角的算法过程，结合 3.2.1 节内容，我们便得到了完整的轨迹追踪计算模型，这个计算模型是本文工作的理论基础。

3.3 实际问题与误差分析

在进入基于惯性感知的机器人运动轨迹追踪的具体介绍前，我们需要对机器人自身的一些问题做些阐述，并进行一定的运动学层面的误差分析。

当我们控制 Create2 机器人运动时，仅有运动模块启用，没有多余的干扰；而当其自由运动时，会有其他模块启动（如扫地引擎），这会对智能手机的惯性感知及磁场感知产生极大干扰。此外，机器人运行时，有较强内部磁场，我们可以通过架高手机来避免对地磁场感知的干扰，不过架高手机后有可能产生简谐振动等震动干扰。引擎震动的干扰也会造成真实的运动加速度信号淹没在噪声中。以上这些问题皆为机器人运动相伴相随的，我们将在下一章给出一些解决或是削弱干扰的办法。下面给出计算模型的运动学层面的误差分析。

我们通过智能手机的加速度传感器获取机器人实时加速度，这里真实加速度作为传感器的输入，输出加速度便会存在一定偏差，越精准的传感器硬件，输入输出就会越一致。我们设传感器系统的噪声与直流分量造成的误差项为 δ ，理想的加速度输入输出满足 $f(t)$ 关系，那么我们就有实际输出加速度信号如下：

$$a(t) = f(t) + \delta \quad (3.13)$$

通过一次积分，我们可以得到速度表达：

$$v(t) = \int a(t)dt = \int (f(t) + \delta)dt = \int f(t)dt + \delta t + \varepsilon \quad (3.14)$$

可以看到，误差项 δ 已经被放大， ε 作为不定积分产物，在实际计算模型中将被消去，视其为数值积分本身的误差。继而进行二次积分，位移的表达如下：

$$\begin{aligned} s(t) &= \int v(t)dt = \int \left(\int f(t)dt + \delta t + \varepsilon \right) dt \\ &= \int \left(\int f(t)dt \right) dt + \frac{1}{2}\delta t^2 + \varepsilon t + \lambda \end{aligned} \quad (3.15)$$

上式表明，通过二次积分得到的位移结果已经存在二次项的累计误差，位移信号极可能失真。这也证明了二次积分在信号处理方面确实有极大的困难。关于处理上述误差，我们做以下几个层面的讨论：

1) 理论层面，提高数值积分本身精准度，削弱一次误差项。除了梯形积分法，数值积分方法还有很多种，如辛普森法、牛顿科茨公式法等，理论上他们具有更精准的积分结果。但它们未从根本上解决噪声误差造成的问题，为了方便对计算模型进行优化修正，避免模型进一步的复杂，本文工作仍然使用梯形积分。

2) 硬件层面，提高惯性感知单元的精准程度。这是本文工作无法触及改变的，这将依靠于传感器厂商和智能手机制造商。

3) 软件层面，使用滤波处理，互补处理等方法。这是惯性感知信号处理的常见手段，本文工作亦将涉及，但这些方法不能完全解决问题，它们也具有造成新问题的风险。在下一章中我们将讨论滤波的使用。

3.4 本章小结

本章首先对所研究使用的 iRobot Create2 机器人的运动进行了分析，阐述了机器人的运动原理和两种运动模式。然后设计了对 Create2 机器人的轨迹追踪计算模型，重点阐述了轨迹追踪计算的算法流程，同时简单介绍了四元数，强调了使用四元数进行转角计算的过程，分析了使用四元数相较于欧拉角和旋转矩阵的优势。最后阐述了机器人实际运动的相关问题，并结合加速度传感器的输入输出对轨迹追踪做了运动学层面的误差分析。

4 基于惯性感知的机器人运动轨迹追踪机制

在绪论中已提到，本文工作使用智能手机作为惯性感知体与计算单元，实时追踪并计算被搭载的机器人运动的轨迹，这对整套机制还是有相当严苛的要求的。在第三章的介绍中，我们已经拥有了轨迹追踪的计算模型，那么结合惯性感知的工作，即可给出具体完整的机器人运动轨迹追踪机制。

4.1 惯性感知数据的处理与应用

在第二章中我们介绍了智能手机所携带的四种传感器：加速度传感器、陀螺仪、重力传感器和磁力计。在第三章的计算模型的介绍中已经涉及了加速度传感器和陀螺仪的数据使用。在 Android 系统中，我们可以很方便的获取各类传感器的实时数据，这些数据是惯性感知和轨迹追踪的基础。下面阐述对智能手机传感器信号的处理与应用。

4.1.1 加速度信号初步处理

信号初步处理是类似的，我们首先介绍加速度传感器的信号处理。下面给出机器人使用手动控制和自由运动的前进方向原始加速度信号，如图 4-1，图 4-2 所示。

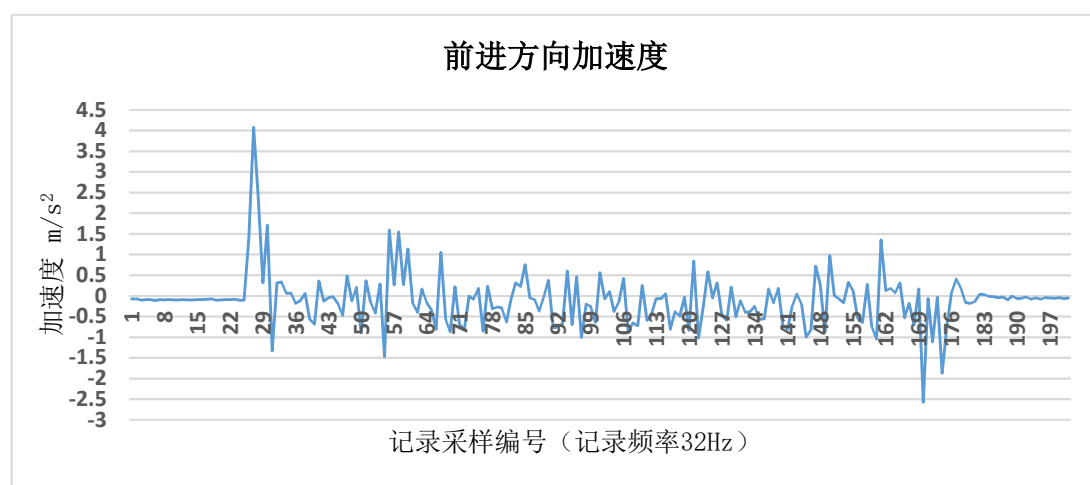


图 4-1 手动控制 Create2 机器人运动一段直线的加速度情况

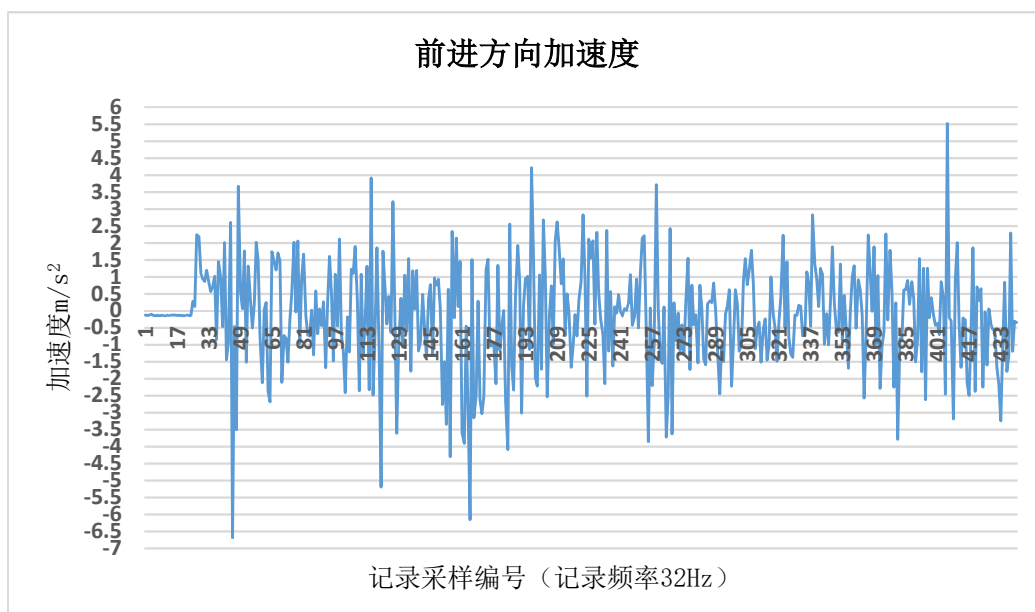


图 4-2 Create2 机器人自主运动一段直线的加速度情况

注意，记录频率非加速度采样实际频率，而是程序做数据的本地记录的频率，关于具体加速度采样频率的阐述将在系统实现章节给出。

手动控制 Create2 机器人的运动情况为进行一段直线运动，可以在对应图表中看到，启动和停止对应的加速和减速的波峰还是比较明显的，且与行进期间的波形有着可以区分的特征。

Create2 机器人自由运动的加速度原始波形则很不理想。上一章中，我们已经提及机器人运动的问题，在自由运动模式下，前进方向的加速度信号将伴随着很大的噪声。在图 4-2 中可以看到，启动之后的波形是极不规律的，但我们仍然可以观察到这里面有高频噪声的存在。

为了消除高频噪声，一般可以使用低通滤波，这在信号处理领域已经成熟，然而应用到本文工作时，虽然它有着很好的消除噪声的效果，但它同时削弱了频率与噪声相近的正常加速减速波形，并会把波形展宽。对比下来，我们使用更为简洁且效果不俗的均值滤波来削弱高频噪声，均值滤波的算法流程如下：

初始化： 对加速度信号设置一个宽度为 W 的滑动窗口，并维护为一个队列。

重复：

步骤1： 获得新的加速度值作为输入，加入窗口队列并更新。

步骤2： 重新计算窗口队列内数据的均值 M 。

步骤3: 将均值 M 返回, 作为均值滤波的结果, 也即处理后的加速度值。

直到 轨迹追踪停止。

在进行轨迹追踪时, 手机与 Create2 机器人所处的平面并不严格水平, 实验时手机自身与机器人平面也不能严格水平, 这时候我们需要滤除重力对加速度值的干扰。注意, 本文工作并不直接使用线性加速度传感器, 线性加速度传感器是官方提供的基于软件的传感器, 本质上也是对加速度的传感器的滤波相关处理, 但不是每个智能设备都会配有, 且不利于实验探究。

我们把经过均值滤波的加速度传感器数据减去重力传感器的数据(重力数据同样可以使用均值滤波处理), 滤去重力影响以得到前件方向的线性加速度。图 4-2 的数据经过处理后, 呈现如下图 4-3:

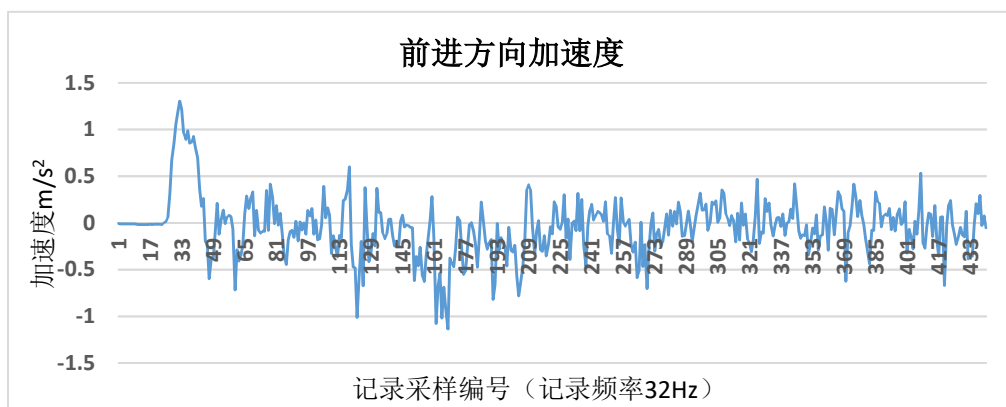


图 4-3 经过初步处理的 Create2 机器人自主运动一段直线的加速度情况

观察上图表, 高频噪声已经得到极大削弱, 启动时的正向加速度也突出明显。理论上来说, 此时的加速度数据便可用于积分计算, 然而我们可以看到, 数据的异常波动依然占主导地位, 所以我们需要根据运动特征, 进一步处理。从 4.2 节开始的相关处理将有部分会涉及工程性的优化办法。

4.1.2 陀螺仪信号的处理与使用

在 3.2.2 节中, 本文给出了应用四元数计算转角的计算模型, 这是本文工作中计算转角的主体部分。根据该节提出的算法, 系统需要不断获取陀螺仪实时数据, 用以构建四元数。下面给出机器人原地旋转的手机陀螺仪信号, 如图 4-4:

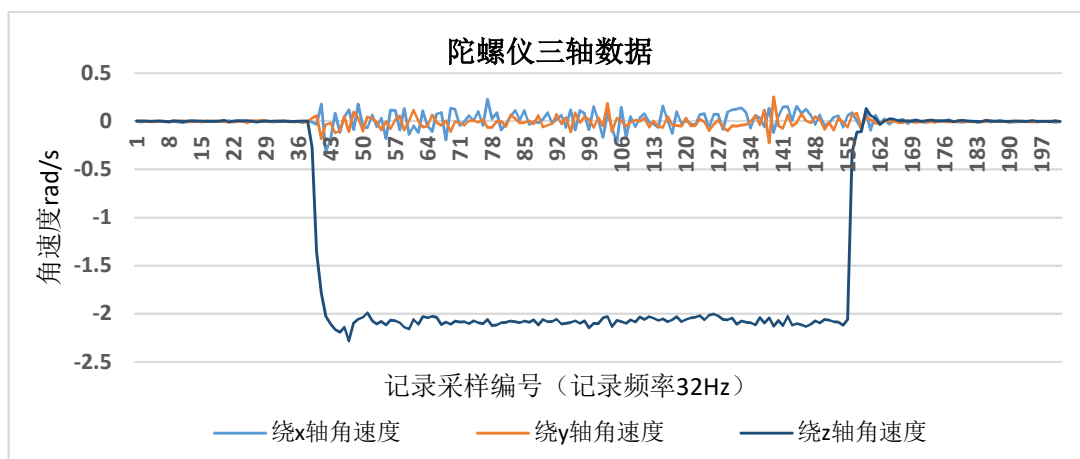


图 4-4 Create2 机器人原地旋转 345° 的陀螺仪三轴数据情况

从上图可以看到在发生平面旋转（绕 z 轴）的同时，其余两个轴也会存在数据波动，但这类波动在单位化后会相对削弱，在实际的追踪过程中，只要波动始终在 0 值附近，那么不会造成大的偏差。

陀螺仪在不发生转向时，会产生数据的基线偏移或者噪声抖动，为此可以直接对其设置阈值，小于阈值的数据将被做缩放处理，弱化其对转角的影响。

基于陀螺仪的数据，我们经过四元数计算转角的算法后，得到转角实时数据如下图 4-5：

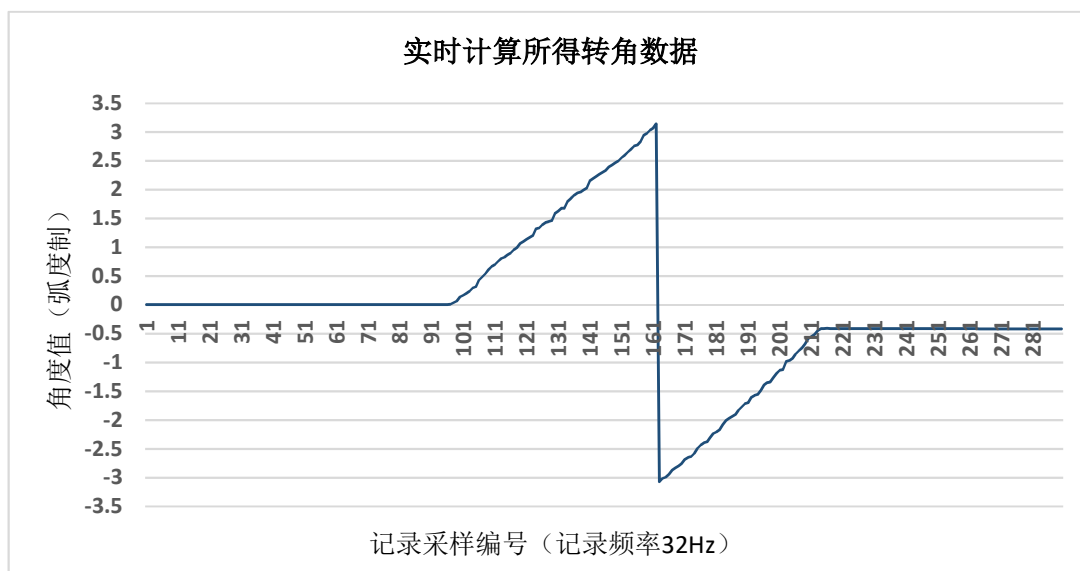


图 4-5 对应于图 4-4 原地旋转的实时计算所得转角情况

转角数据最终停留在-0.414 左右，转换为弧度制约为 -23.73° ，也即计算出最终转角为 336.27° ，误差为 -3.73° ，在可接受范围内。转角的计算情况在较

短时间内都比较出色，不过当轨迹追踪系统长时间运行，会导致转角误差的累积，或者是机器人发生连续的碰撞，会导致出现不可接受的累积偏差。

相较于处理加速度数据，陀螺仪数据是较为简单明了的。基于 3.2.2 节算法计算所得的转角数据将支撑整个轨迹追踪的进行。

4.1.3 应用磁力计参与转角计算

除了上一小节所提及的基于陀螺仪数据计算转角之外，本文给出的轨迹追踪机制还涉及了应用磁力计参与转角计算的方法。根据 Android 开发官方介绍，结合磁力计和加速度/重力传感器可以得到设备方位角，官方提供的 API(Application Programming Interface, 应用程序编程接口)中提供了相关方法[10]。

在手机静止状态下，重力传感器与加速度传感器测得数据理论上一致，而本质上需要的是重力向量，故我们分析重力矢量与设备所受地磁场矢量如何结合计算出方位角。在使用磁力计前，我们需要对设备进行八字校准操作，来纠正磁力计可能存在的系统偏差。

我们令设备重力矢量为 \mathbf{g} ，方向竖直向下，所受地磁场矢量为 \mathbf{m} ，不考虑磁偏角的情况下，方位为由南朝北。两者做叉乘可得方向由东指向西的矢量 \mathbf{h} 。经过这次叉乘，获得可构建三维空间的三个矢量。我们再将 \mathbf{g} 与 \mathbf{h} 叉乘，这次运算重新得出一个水平面由南指北方向的向量 \mathbf{l} ，最后由 $\mathbf{h}, \mathbf{l}, \mathbf{g}$ 三个矢量单位化后构成旋转矩阵用以得到方位角。矢量关系与旋转矩阵如下：

$$\mathbf{h} = \mathbf{m} \times \mathbf{g} \quad \mathbf{l} = \mathbf{g} \times \mathbf{h} \quad \mathbf{R} = [\mathbf{h}, \mathbf{l}, \mathbf{g}]^T$$

设备相对于世界坐标系的三轴方位角依照如下原理得到，绕 Z 轴的方位角 θ ，绕 X 轴的方位角 φ 和绕 Y 轴的方位角 σ 分别如下：

$$\theta = \tan^{-1} \frac{h_y}{l_y} \quad \varphi = \sin^{-1}(-g_y) \quad \sigma = \tan^{-1} \left(-\frac{g_x}{g_z} \right)$$

如此，我们便可以通过实时监测设备的方位角，计算出水平面的转角，也即记录轨迹追踪启动时 θ 的初始值记为 θ_0 ，那么之后每次计算得的 θ 与 θ_0 的差值即为当前转角。下面给出磁力计计算转角与上一小节介绍的陀螺仪计算转角的对比，见图 4-6 如下：

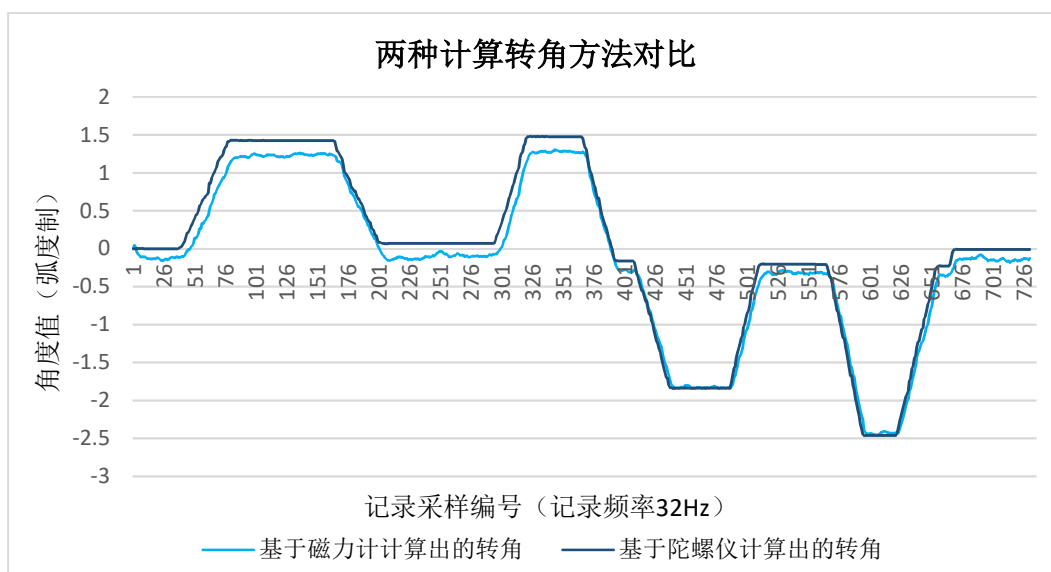


图 4-6 Create2 机器人原地来回旋转的情况

可以看到，两者的计算结果还是比较相近的，当然这是在比较空旷的环境下才能达到的理想效果。在机器人进行直线运动时，基于磁力计计算出的转角很有可能因为重力传感器受到的影响以及周围环境变化的影响而产生漂移，这是很难解决的问题。所以在直线运动之时，我们还是主要依靠陀螺仪的计算结果。

回到应用场景中，由于我们需要磁力计测得地磁场强度，但是机器人在室内运动时，环境是复杂而不可测的，手机周围磁场极易受到干扰，如承重墙内的钢筋或电线影响。所以，单纯使用磁力计对转角的计算是极不安全不稳定的。此种方法亦只能作为辅助，在一定条件下纠正已经存在偏差的基于陀螺仪计算出的转角值。

在较为空旷的室内，则可以将两种方法结合使用，形成一个互补关系，共同计算转角值，最简单的，可以对两者进行加权平均计算出转角值。而在复杂环境下，此种方法目前还不是很适合。

4.2 运动状态判别

对机器人运动状态的判别是本文工作的一个重点，不同的运动状态作为轨迹追踪机制的内部反馈，决定了如何对加速度数据进一步处理，并且将作为依据对计算中的速度进行校零。

4.2.1 状态判别的准备

宏观来看，机器人的运动是无序的且无规律的，惯性感知获得的数据充斥着大量的噪声，这两个特征对实时判别运动状态带来了挑战。运动状态判别本质上是进行数据分类，然而，由于研究对象的这两个特征，我们无法获得大量并且精准的样本，这意味着使用数据挖掘相关方法的可能性极低。

在轨迹追踪的正是追踪阶段进行之前，也即机器人运动之前，我们需要手机在静止状态下进行各类惯性感知传感器的数据采集和进行相关计算，这是进行运动状态判别的准备，也是整个轨迹追踪的准备。

在准备阶段，轨迹追踪机制将同时进行消除惯性感知的基线偏移和计算下列五种数据特征值的操作：静止状态下前进方向（Y 轴）加速度的**平均值**、**方差**和相邻数据的**绝对差值**平均值以及实时计算出的转角的**方差**和相邻数据的**绝对差值**平均值。

每种特征值都依靠基于传感器数据的滑动窗口计算获得，可以设置不同的窗口长度，分别求出对应的值，相当于一种特征值可以拥有多项具体表示。

所以，不妨设它们的窗口长度为只有一种，也即每种特征值只有一项数据。我们对静止状态下的 5 项运动特征值做如下简记：

- 前进方向加速度：
1. 平均值：SAM
 2. 方差：SAV
 3. 相邻数据绝对差值的平均值：SADM
- 实时计算的转角：
5. 方差：SOV
 6. 相邻数据绝对差值的平均值：SODM

S 代表静止状态 (Static)，A 代表加速度 (Acceleration)，O 代表方位角 (Orientation)，D 为 Delta 的简记，M 和 V 为平均值 (Mean) 和方差 (Variance) 的数学记号。这 5 项数据在轨迹追踪开始后将保持不变。

不失一般性，给出计算前三项的算法流程如下：

初始化： 对静止状态下的加速度数据 a 和相邻两项加速度数据的绝对差值 Δa 分别设置一个宽度为 W 的滑动窗口，并维护为两个队列。为了体现即时性又防止噪声数据点干扰， W 一般设在 10 左右。之后开始对准备阶段的计

数 c ，计数上限为 C 。

重复：

步骤1： 获得新的加速度值 a ，计算新的 Δa ，将两者加入各自的窗口队列并更新队列。

步骤2： 基于 a 的滑动窗口更新计算 SAM。

步骤3： 基于 a 的滑动窗口与已计算所得的 SAM 更新计算 SAV。

步骤4： 基于 Δa 的滑动窗口更新计算 SADM。

直到 c 等于 C ，计数停止。

转角的相关特征值计算与上述流程相似，不再赘述。在轨迹追踪的准备阶段完成后，我们将在机器人启动（实时追踪阶段开始）后，充分利用这 5 个运动特征值。

4.2.2 状态分类与判别方法

基于上一章的运动分析，本文工作将 Create2 机器人的运动状态分为如下五类：静止态、加速态、减速态、前进态、旋转态。

静止态为机器人运动前与运动完成后的状态。这个状态下手机惯性感知获得的数据相对稳定，加速度传感器信号和陀螺仪信号在一个范围内小幅波动，可能存在初始的基线偏移，在这个阶段我们也很容易计算出这个平均偏移值，在机器人进入其他状态后，进行消除。机器人启动前的静止态即为 4.2.1 节的准备阶段，基于惯性感知的轨迹追踪机制将完成该节涉及的所有准备操作。

加速态为机器人加速运动状态。理论上持续时间极短，一般发生在启动时，或是碰撞转向之后的重新加速阶段。加速态下，加速度信号将出现明显的波峰，

减速态为机器人减速运动状态。理论上持续时间极短，一般发生在碰撞或者停止前进之时。减速态下，加速度信号将出现明显波谷。

前进态为机器人正常前进状态。理论上这个状态进行的是匀速直线运动，机器人的引擎给左右两个轮子的输出功率是一致的。但在实际运动中，由于环境干扰以及系统偏差，完美的匀速直线运动是不存在的。在这一状态，惯性感知的信号波动皆比静止状态下剧烈。

旋转态为机器人进行旋转运动状态。一般发上在碰撞后的自旋转或是螺旋运

动之时，角速度理论上为匀速。实际情况则会产生一定偏差，螺旋运动则半径不断扩大。这个状态下，陀螺仪的绕 Z 轴角速度明显增大，计算出的转角值成线性增长。下面给出机器人进行探路型运动的状态转换模拟示意图与五种状态的转换图，如图 4-7：

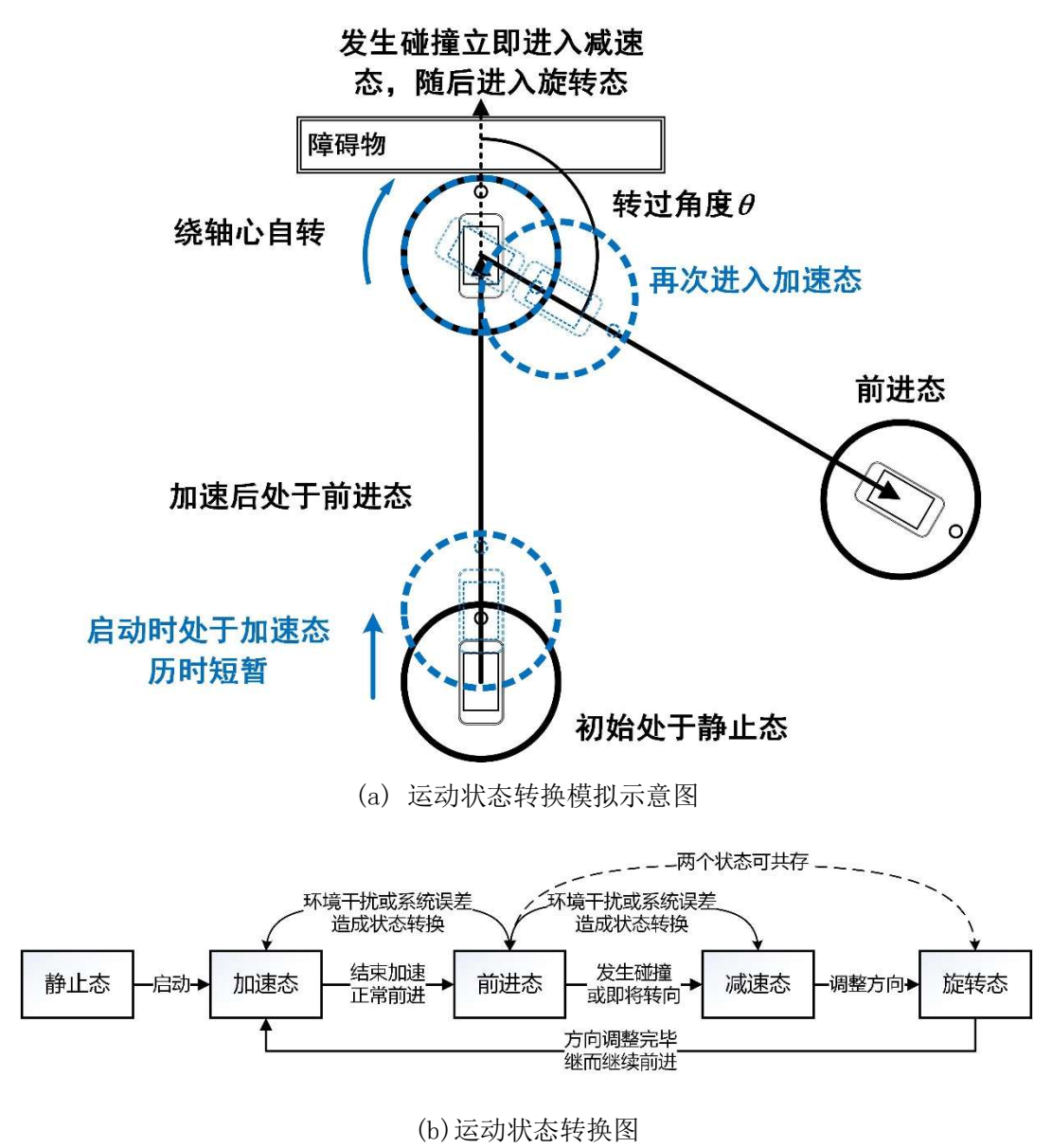


图 4-7 基于惯性感知的机器人运动轨迹追踪机制的运动状态示意图与转换图

图中没有给出运动停止的转换，因为实际情况一般是人为干预结束机器人的运动并结束轨迹追踪的，故在此说明。下面给出五种状态判别的方法。

机器人启动后，实时追踪阶段开始，系统将新建 4.2.1 小节中阐述的 5 个运动特征值，这不再是静止态的特征，而是实时运动的特征，仿照该节，将它们如

下简记：

- 前进方向加速度： 1. 平均值：AM
2. 方差：AV
3. 相邻数据绝对差值的平均值：ADM
实时计算的转角： 4. 方差：OV
5. 相邻数据绝对差值的平均值：ODM

此 5 项特征数据将如该节介绍的算法一致进行实时更新与监测，与准备阶段获得的 5 项数据将进行对比分析。本文工作给出的五种运动状态的判别依据如下：

静止态：数据方差和相邻数据绝对差值的均值可以反应数据波动程度，一旦处于非静止态，惯性感知信号都将产生较大波动，故当加速度波动处在围绕零点的合理范围内，即可判为静止态（合理范围将依赖经验参数）。判别条件如下：

$$ADM < r \times SADM \wedge SAM - r \times SADM < AM < SAM + r \times SADM$$

其中 r 为经验参数，一般设在 2~3 左右。

加速态：加速态和减速态的判别将为下一节的异常区间填补服务，加速和减速态的判别很直观，加速度当前的绝对差值必须大到一定程度，这可以和静止态的绝对差值作比，同时方差也将比静止态高。而我们根据加速度数据当前值 a 与经验阈值相比来判定加速还是减速。判别条件同样将涉及经验参数，条件如下：

$$ADM > t \times SADM \wedge AV > u \times SAV \wedge a > threshold$$

其中 t, u 为经验参数，一般设在 (50, 100) 范围内， $threshold$ 为加速减速态阈值，本文给出的轨迹追踪机制将其设定为 0.5。

减速态：与加速态同理，不多赘述，给出判别条件如下：

$$ADM > t \times SADM \wedge AV > u \times SAV \wedge a < -threshold$$

旋转态：当实时计算的转角的方差波动变大以及相邻差值明显增大时，就说明机器人正在旋转，这是很容易判别的，因为在正常探路前进时，并不会出现这种情况。判别条件如下：

$$OV > c \times SOV \wedge ODM > d \times SODM$$

其中 c, d 为经验参数，一般设在 (500, 600) 范围内。在实际的系统实现中，还要考虑防止直线运动时，转角从 180° 到 -180° 的变化带来误判，可以添加一

个相邻转角值的差值小于 π 这个条件。

前进态：原则上，经过上述所有条件还不能判别状态的，皆为前进态。因为判别顺序为：旋转态，加速态，减速态，静止态，前进态，所以一般来说前进态和静止态大致互补。也即这个状态起码满足：

$$AM < SAM - r \times SADM \vee AM > SAM + r \times SAD$$

前进态的加速度抖动将通过缩小数据来进一步平滑，可见下一节图表。

运动状态的判别与轨迹计算时同时进行的，判别结果将实时反馈给轨迹追踪的计算模块，以便进行进一步处理。

下面将阐述轨迹追踪中的加速度数据特殊处理，对加速度信号异常区间的处理。

4.3 加速度的异常区间填补

在本文工作研究初期，时常会发现在开始运动之后，计算出的轨迹方向与机器人运动方向相反的现象。除了可能的积分误差的外，主要原因来源于智能手机配备的压电加速度传感器会有低频失真现象出现。这在 WU 等人的论文[7]中有着理论分析与实验论证，阐明了此现象的产生原因。下面选取一段机器人自由运动的加速度数据，来反映该问题，见下图 4-8：

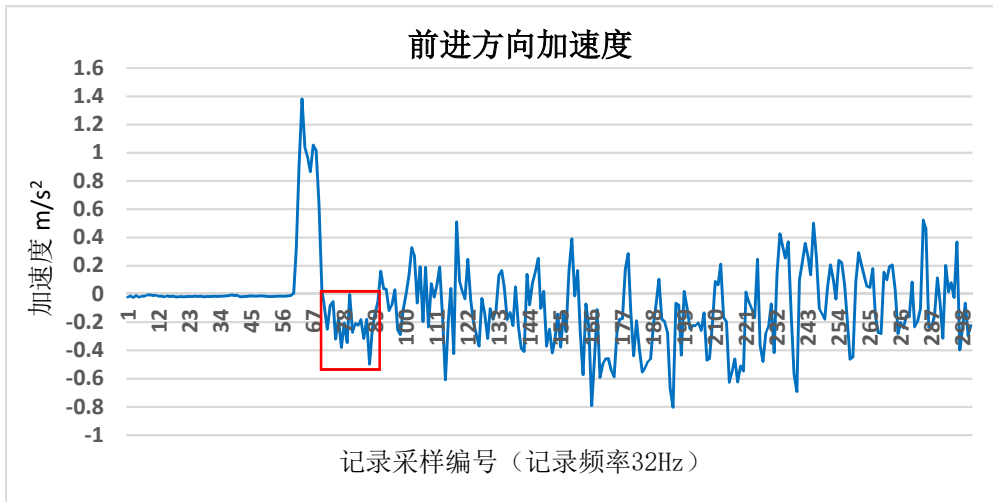


图 4-8 Create2 机器人自由运动直线前进的加速度情况（已经过初步处理）

可以看到，在红色矩形框范围内，加速度呈现异常的负向，这在加速状态下

是不合理的。根据[18]的理论分析，压电传感器会对输入的锯齿波做如下输出，如下图 4-9（引自[18]）所示：

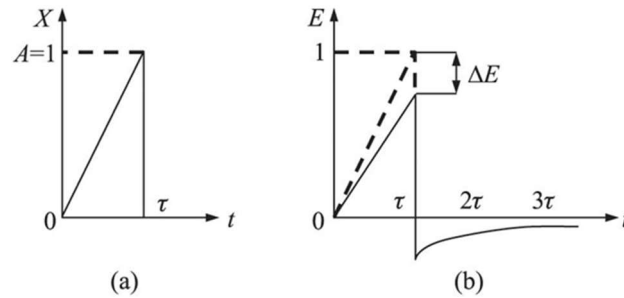


图 4-9 锯齿形输入波与压电传感器输出波形

其具体原理超出本文工作范畴，但是智能手机内的压电传感器对实际加速度波形的反应偏差是存在的，如缩小了最高峰值，增添了反向的波峰等。图 4-8 中矩形框内的反向加速度区间在经过计算模型积分后，很可能抵消之前波峰积分结果，甚至出现方向速度的结果。

对此问题，本文提出的轨迹追踪机制将采取比较工程性的办法，对异常的反向区间进行填补。系统实现中的具体做法是缩小红色矩形框内的波形，削弱其对积分计算的影响，避免反向速度的出现。

出现在加速态的反向加速度与出现在减速态的正向加速度都是需要处理的对象，本质上这是对加速态和减速态的一种保护，在一段时间内保证加速或是减速的纯洁性，使计算模型生效，速度能到达一定可观值。下面介绍对异常区间的判定方法，算法流程如下：

初始化： 设置加速态和减速态的判别标志 F ，初始为 0；设置状态持续计数 Count，初始为 0；设置平衡计数 BalanceCount，初始为 0；设置 BalanceCount 的上限值 B （系统实现中 B 为 20）；设置填补触发标志位 Flag，初始为 false；在实时最终开始后，进行下列步骤。

重复：

步骤 1： 获取当前运动状态。

若为加速态并且此时 F 为 0，则 $F = 1$ ，进入步骤 2-a；

若为减速态并且此时 F 为 0，则 $F = -1$ ，进入步骤 2-b；

否则直接跳入步骤 3。

步骤 2-a: 如果 F 为 1 且处于加速或减速态, 则进行如下分步骤:

- 1) 如果当前 $a \geq 0$, Count 加 1; 否则 Count 减 1, Flag 置为 true;
 - 2) 如果 $F \leq 0$, $F = 0$;
 - 3) BalanceCount 加 1; 如果此时 BalanceCount 大于上限值 B ,
则 $F = 0$, BalanceCount = 0, Count = 0;
 - 4) 直接跳入步骤 4;
- 否则顺序进入步骤 3。

步骤 2-b: 如果 F 为 -1 且处于加速或减速态, 则进行如下分步骤:

- 1) 如果当前 $a \leq 0$, Count 加 1; 否则 Count 减 1, Flag 置为 true;
 - 2) 如果 $F \leq 0$, $F = 0$;
 - 3) BalanceCount 加 1; 如果此时 BalanceCount 大于上限值 B ,
则 $F = 0$, BalanceCount = 0, Count = 0;
 - 4) 直接跳入步骤 4;
- 否则顺序进入步骤 3。

步骤 3:

- 1) 若 Count > 0, Count 减 1;
- 2) 若此时 Count 为 0, $F=0$;
- 3) 若此时 F 为 0, BalanceCount = 0; 否则 BalanceCount 加 1;

步骤 4: 如果 Flag 为 true, 则需要对处于异常区间的当前数据进行填补操作, 即缩小数值; 之后重置 Flag 为 false。

直到 轨迹追踪停止。

上述算法叙述略显冗长, 其实它的根本目的是确保一个合理的加速态或减速态能专注于自己方向的加速度, 而削弱反向加速度。

削弱反向加速度的操作, 是有一个合理时间跨度的, 它由正向加速度的持续计数 Count 决定, 例如在加速态正向加速度连续出现 10 次, Count 也就达到了 10, 这之后不再出现达标的加速度或是出现反向加速度, 那么在接下来的 10 次计数内, 就是削弱阶段, 直到 Count 归为 0。削弱加速度阶段同时受制于平衡计数 BalanceCount, 它保证了加速度不会受异常数据段影响而一直被削弱。下面给出图 4-10 对应运动的两个计数情况, 它们反映了运动状态和进一步处理的依据。

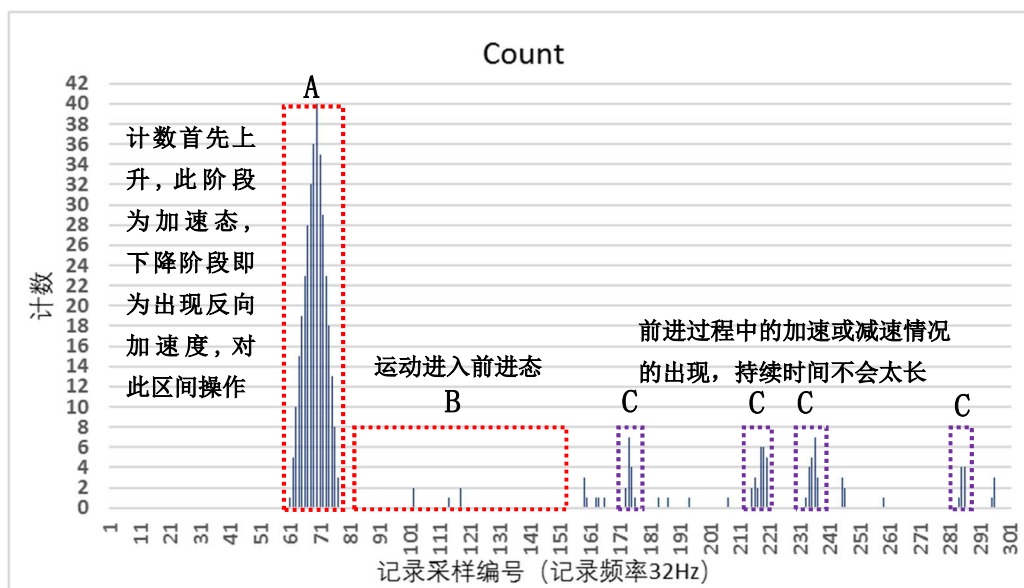


图 4-10 对图 4-8 运动的异常区间填补算法 Count 计数情况

上图中对计数情况进行了区域分割，A 区域为异常区间识别和处理，首先会看到计数的上升，这代表了处于正常的启动后的加速态，当数据下降则表示此时出现了反向加速度，需要进行处理；B 区域则也已进入正常的前进态；C 区域则表示了在前进过程中，加速度数据仍存在一定程度的抖动，所以出现了运动状态向加速或减速态的跳动，但持续时间都较短，不影响总体的状态判别。理论上，由于地面不平整或者内部功率不恒定等问题，也并不存在严格的匀速直线运动。

异常区间填补处理与上一小节的运动状态判别中提到的前进态也需要削弱波形结合，将使得最终计算出的前进速度没有大幅度抖动或偏移，维持在一个合理波动内。同时，基于上述内容提及的削弱的参数依赖于实验而取得，对它们得做工程性的参数调整，目前设置削弱的缩放比例为 0.1。

加速度数据最后的处理为放大启动加速度，也即由其他状态转为加速态之时的加速度，由图 4-9 可知，锯齿波峰信号会被传感器内部削弱，所以我们需要将他们重新放大一些。即在这些转换状态的时段内：

$$a_{\text{最终值}} = A a_{\text{经过异常区间处理}}$$

经过上述步骤，所有针对加速度数据的处理都已结束，对应图 4-8，我们给出处理完毕之后的加速度波形，如下图 4-11：

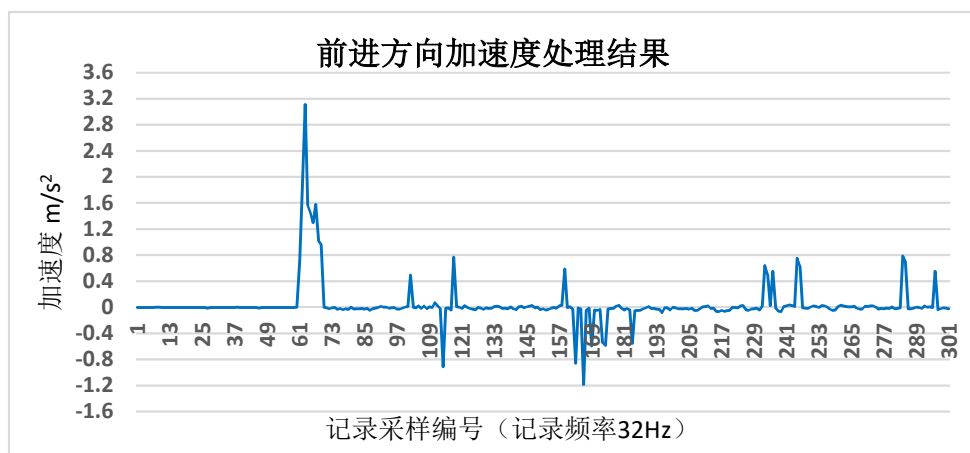


图 4-11 Create2 机器人自由运动直线前进的加速度情况（最终处理结果）

可以看到，此时的加速度数据经过异常区间的填补缩放处理，已经具有很强的启动加速特征，同时也保留了在前进过程中的合理抖动。此数据将用于计算模型的速度运算之中。

4.4 速度校零

针对实时计算出的速度的处理，机制中只有一项，便是速度的过零点校准。轨迹追踪计算模型的误差以及数据本身误差会导致积分所得速度存在偏差。这导致很多速度本应该为 0 时间点，计算出的速度依旧大于 0，甚至远大于 0。这是轨迹追踪机制要及时纠正的。

针对机器人的运动特点，正在发生碰撞或者停止前进时，都会进入旋转态，此时便是进行速度的校准的时机，示意图见下图 4-12。由于旋转态可能会伴随前进态，所以我们只在旋转态的开始阶段进行校准。

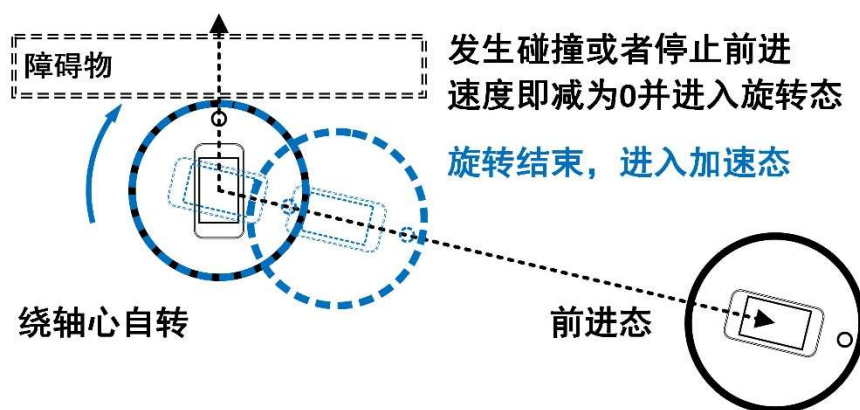


图 4-12 速度校零的依据示意图

当系统监测到旋转态时，对前进方向的速度进行连续 m 次的折半操作，那么速度将立即趋近于 0，达到校零效果：

$$v_{\text{校零结果}} = \left(\frac{1}{2}\right)^m v_{\text{原始}}$$

式中的 m 在系统实现中设为 5，效果已经较为理想。同时，如果当前运动状态转为静止态，则表示机器人此时已结束运动，即可对当前速度立刻置为 0，以防轨迹计算继续，形成偏差。

4.5 轨迹追踪机制综述

至此，基于惯性感知的机器人运动轨迹追踪机制基本介绍完毕，将前四小节的所有操作应用于第三章的计算模型中，便可以完成即时的轨迹追踪。我们对整个机制给出模块化的系统图，如下图 4-13：

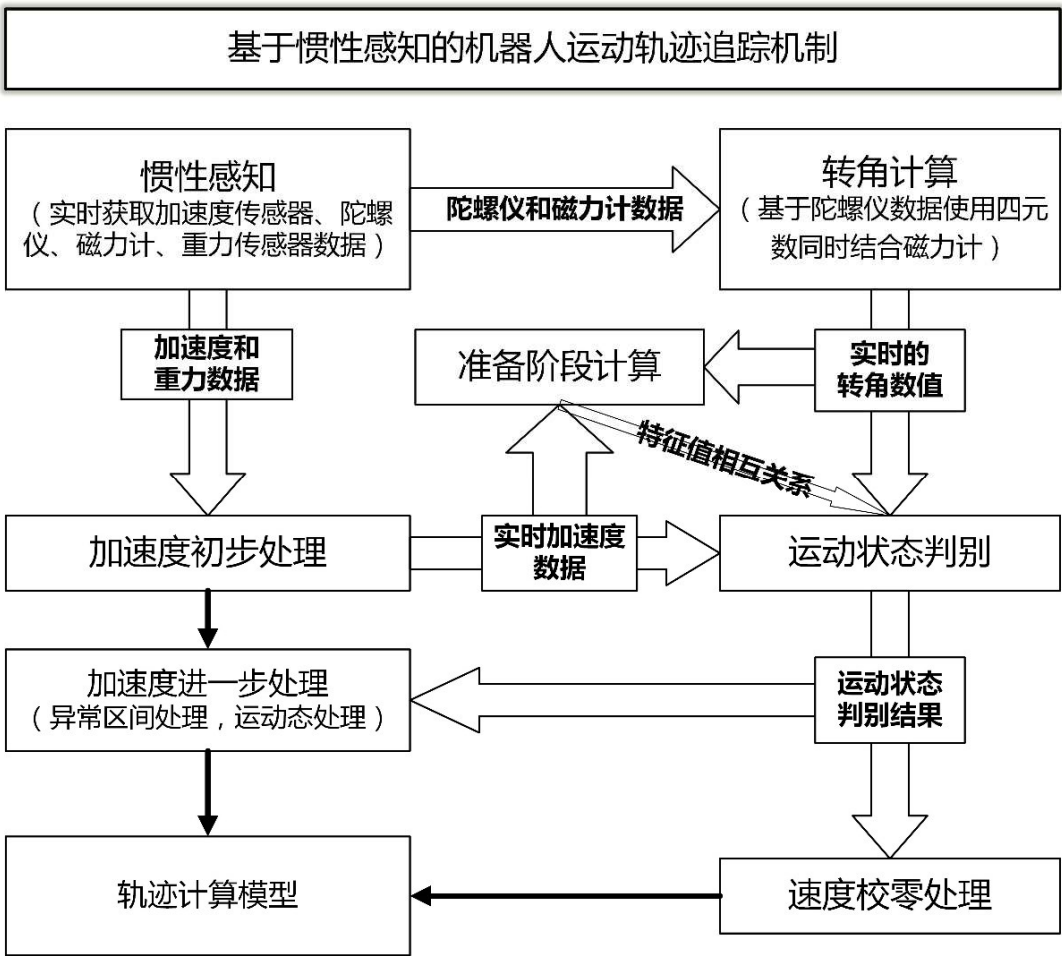


图 4-13 基于惯性感知的机器人运动轨迹追踪机制系统图

上图描述了各模块的逻辑关系，而非严格的时序关系，在追踪进行时，它们都是同时运行工作互相协调的。我们可以依据实时的坐标位置在系统实现中给出轨迹显示，或是记录到本地文件，利用别的软件做出轨迹图。

4.6 本章小结

本章完整的给出了基于惯性感知的机器人运动轨迹追踪机制。首先介绍了惯性感知所得的数据基本处理，给出了应用磁力计参与转角计算的方法。接着重点阐述了运动状态判别相关内容，基于各类状态，将对加速度进行包括异常区间填补的深入处理，并可以进行速度校零。最后我们给出了整个机制的系统图并对整个机制做了简单综述。

5 系统实现与实验分析

本系统是对前述的基于惯性感知的机器人运动轨迹追踪机制的实现，系统包括惯性感知、轨迹追踪和轨迹即时显示三个模块。系统的呈现形式为 Android 系统的智能手机应用。系统应用对象为搭载了 Android 系统智能手机的 iRobot Create2 机器人，使用此系统进行实际测试与实验记录的设备如下图 5-1：

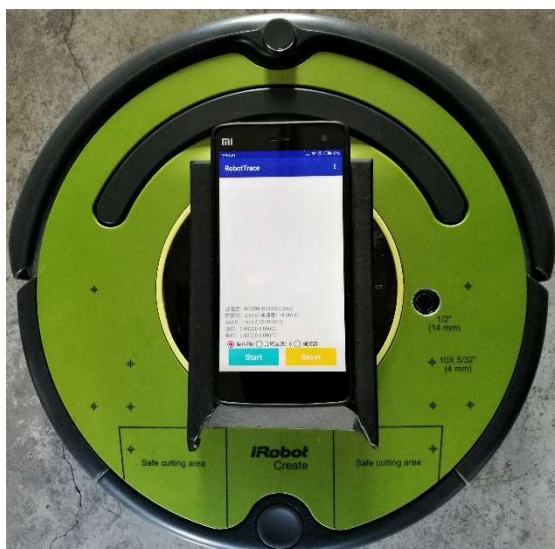


图 5-1 搭载了 Android 智能手机的 Create2 机器人

图中实时轨迹追踪的手机应用正处于开启模式。下面就整个系统实现进行要点介绍。

5.1 开发环境与平台

硬件条件：

本文工作使用的实验手机是小米 4 TD-LTE 型智能手机。

本文工作基于的机器人是 iRobot Create2 型号智能扫地机器人。

系统平台：

软件应用运行平台为 Android 系统，目前最低兼容 4.0 版本。

开发工具：

谷歌公司提供的 Android Studio 版本号：2.2。

5.2 系统设计

本文工作涉及的系统实现目的是为了轨迹追踪机制的完善和优化，所以最终的实现并不能说是一款完善的鲁棒的手机应用，但它能完成机器人运动的实时轨迹追踪，真实反应本文工作。

实际上，系统实现有两个版本。一个版本为实时计算机器人坐标位置，并将轨迹追踪相关数据记录到本地，最终我们可以根据这些数据进行线下的分析，继而继续优化追踪机制，优化实验。另一个版本则为在手机屏幕上实时绘制轨迹，即为提供给用户使用的一款手机应用，此版本更适合于展示。两个版本的系统实现核心设计是一样的，只是实时显示轨迹的版本需要增加一个视图绘制模块，故系统设计的相关介绍不将两者分开。

5.2.1 多模块设计

本系统多模块并行化特点突出，故从模块介绍开始，剖析整个系统。系统涵盖的模块可分为如下 7 个：传感器监听模块，数据处理模块，运动状态别模块，转角计算模块，轨迹计算模块，数据记录模块和实时轨迹绘制模块。

传感器监听模块：

Android 官方提供的 API 很好的方便了开发人员获取各类传感器的数据，系统通过 SensorManager 对象对各类传感器进行监听并获取实时数据。这一模块根植于系统的主线程，实现了系统的传感器监听接口，对加速度传感器、陀螺仪、磁力计和重力传感器四者注册监听。在系统运行时，程序可以通过监听机制，即时从 SensorEvent 对象中获取四种传感器的数据，并将这些数据拷贝到系统内部，用于后续处理。

数据处理模块：

此模块工作在数据监听模块之上，对其得到的数据进行处理封装。在准备阶段，此模块计算并处理各传感器初始基线偏移。在用户开始轨迹追踪后，对需进行均初始处理的传感器数据进行均值滤波处理，均值滤波被实现为即时输出处理并得到返回的方法。初步处理结束后，将陀螺仪数据和磁力计数据传递给转角计算模块，将加速度数据传递给运动状态判定模块。

运动状态判别模块：

根据 4.2.3 节方法，实现对运动状态的判定同时对加速度是否进一步处理进行判断，也即判定异常区间。此模块针对加速度部分，更新一个全局状态量 status,。系统内对运动状态判定使用的 5 个特征值各维护了三个不同长短的滑动窗口，以产生三项判定结果，最终采用投票机制得出 status 值。当 status 值为 0 时，表示静止态；值为 11 代表加速态，值为 10 则代表处于加速异常区间，需要进行反向加速度填补处理；值为-11 代表减速态，值为-10 则代表处于减速异常区间，需要进行正向加速度填补处理；值在 (0, 10) 间，则为前进态。针对转角部分，则根据 4.2.2 节内容，直接判定是否处于旋转态。

转角计算模块：

根据实时得到的陀螺仪数据，计算转角。并将当前转角值传递与运动状态判定模块和轨迹计算模块。

轨迹计算模块：

作为并行的计算线程而存在。根据从运动状态判定模块得到的状态 status 的值以及是否处于旋转态，进行一系列的深入处理，包括加速度异常区间填补和速度校零。上述处理过后，依据第三章给出的计算模型进行速度和位移的积分计算，得到实时的轨迹情况。实验给出的结果显示，计算线程的运行间隔在 10^{-5} s 这个量级，这意味着数值积分切片已经非常小，理论上已经能达到拟合真实积分的要求。

数据记录模块：

这个模块独立于前五个模块，工作在单独的线程，记录所有所需的数据，包括原始传感器数据，各个加工处理过的中间变量，以及实时计算出的转角、速度与位移等。系统将所有数据以 csv 文件的格式记录在本地。

实时轨迹绘制模块：

同样作为一个独立工作线程，从主线程中获取机器人当前的位置坐标。轨迹绘制通过 Path 类实现。关于 Android 的 UI 绘制研究并不在本文工作范畴内，轨迹绘制的工程实现基本参考官方手册。更新屏幕画面的基本逻辑为：根据新传入的坐标值，构建轨迹 Path 需要到达的新的坐标点，再将这条 Path 延伸到该点，并将 Path 在画布上画出。

各模块协同工作，其流程和数据传递情况如下图：

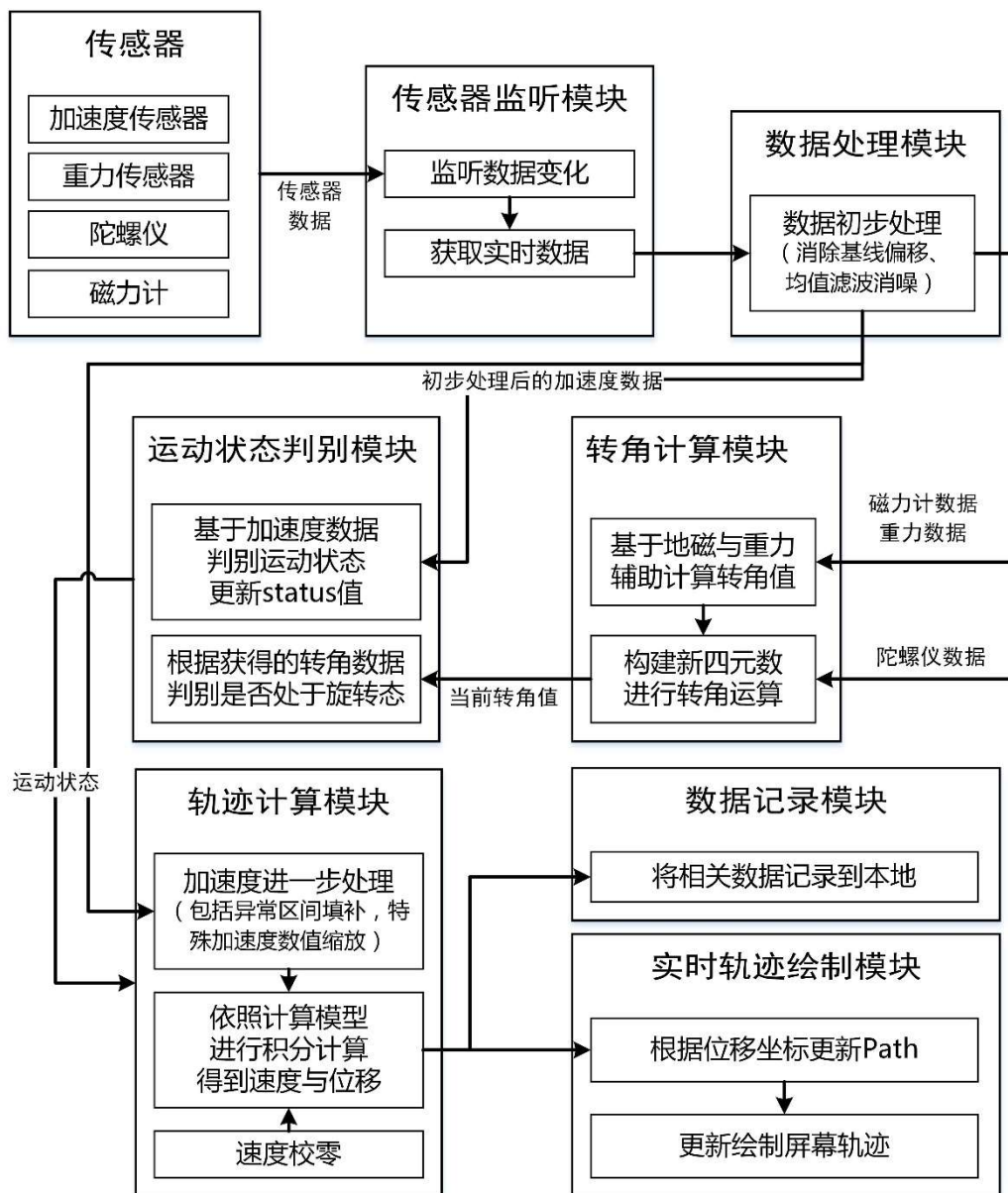
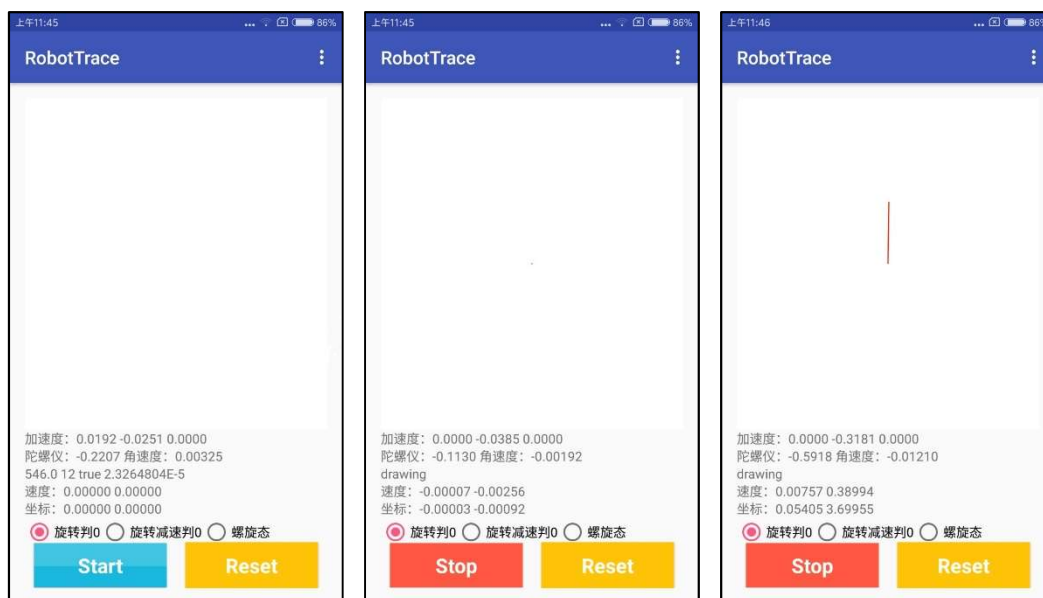


图 5-2 系统流程和数据流简图

以上为系统设计的核心内容介绍，下面给出系统界面以及运行相关互介绍。

5.2.2 界面与运行

下面给出系统实现的实时轨迹绘制版本的应用界面，如下图 5-3：



(a) 准备阶段的界面

(b) 按下 Start 开始追踪

(c) 追踪过程中

图 5-3 系统实现界面展示

简单介绍运行流程：上图 5-3(a) 展示的是系统处于轨迹追踪的准备阶段，在准备阶段完成后，第三行的数据显示会停止变动，此时开始实时追踪，按下 Start，即进入图 5-3(b) 的界面，开始机器人运动的追踪，图 5-3(c) 为已经前进一段时间的机器人实时轨迹。

同时可以看到，界面下方的单选按钮组件提供了运动模式和速度校零方式的选择。用户一般不用做出选择，系统自动运行并进行轨迹追踪与轨迹绘制。第一项选择为“旋转判 0”，为系统默认选项，也即完全根据前文所述机制进行轨迹追踪；第二项选择基本不使用，作为实验对比用途，对速度校准的条件更加苛刻，需要判定机器人处于减速态；第三项为机器人进行螺旋运动而准备。选择“螺旋态”后，追踪过程不进行旋转态的速度校零。

以上便是系统运行的简单介绍，下面介绍本文工作的实验情况，并进行分析。

5.3 实验与分析

本文工作伴随着长时间的大量实验而完成。下面对基于完整的轨迹追踪机制所做实验给出数据展示和分析。

Create2 机器人运动环境：宿舍楼室内较大空地。

Create2 机器人运动模式：自由随机探路型运动。

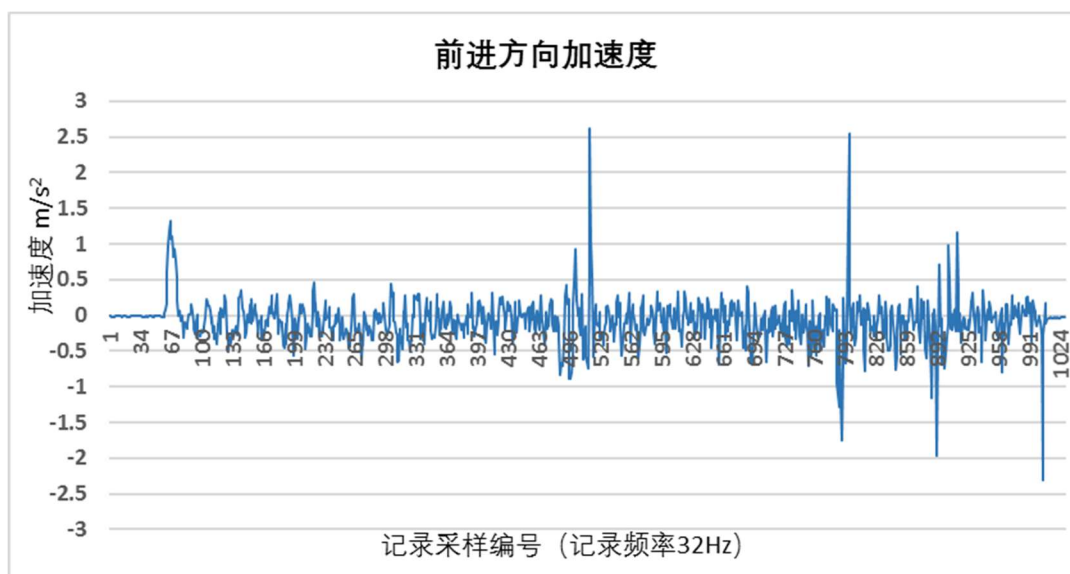


图 5-4 经过初步处理的前进方向加速度

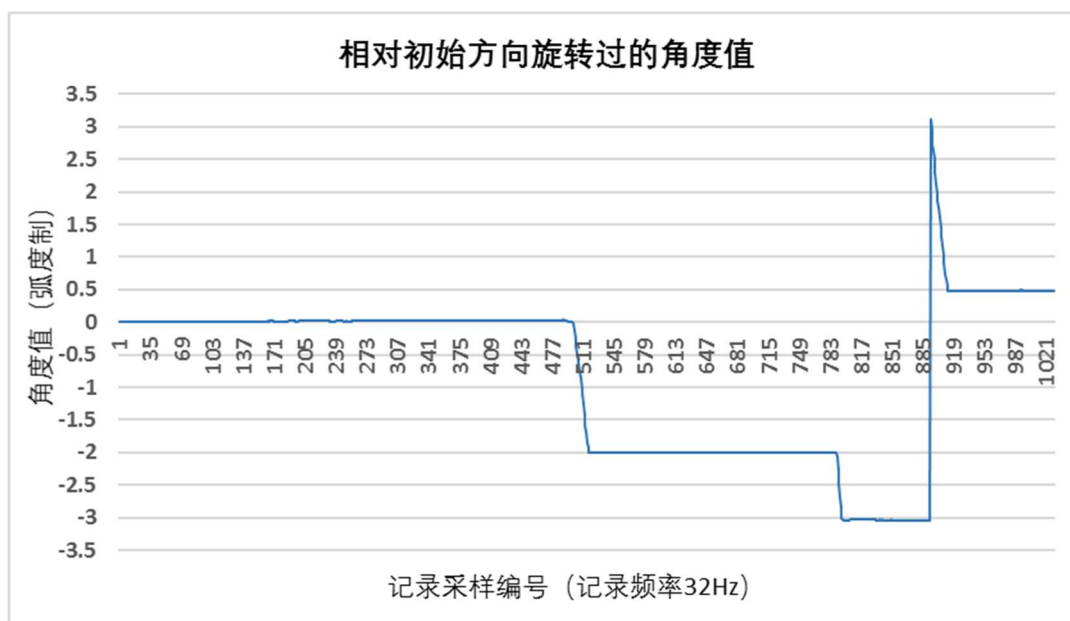
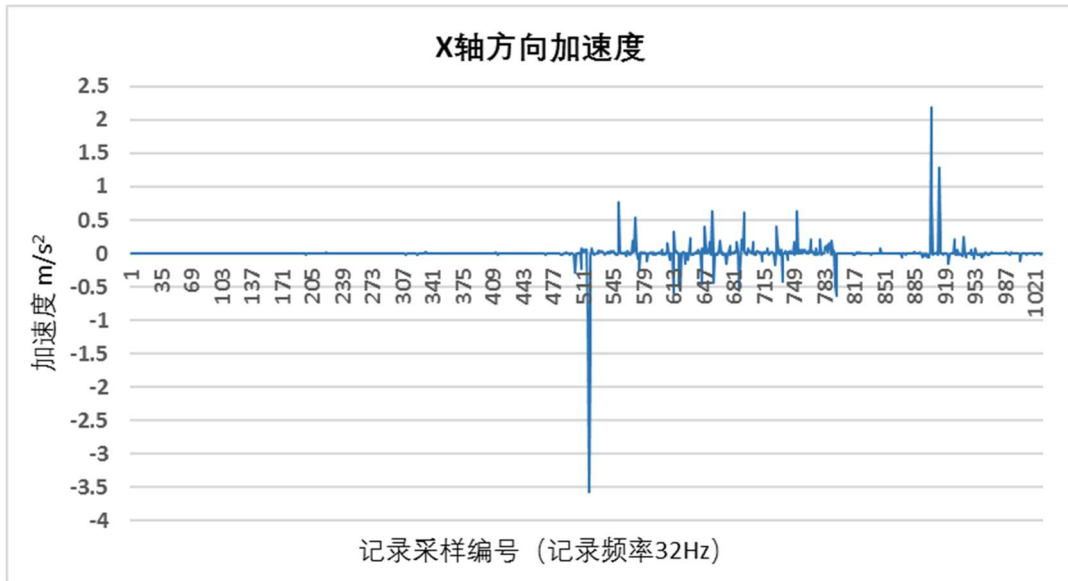
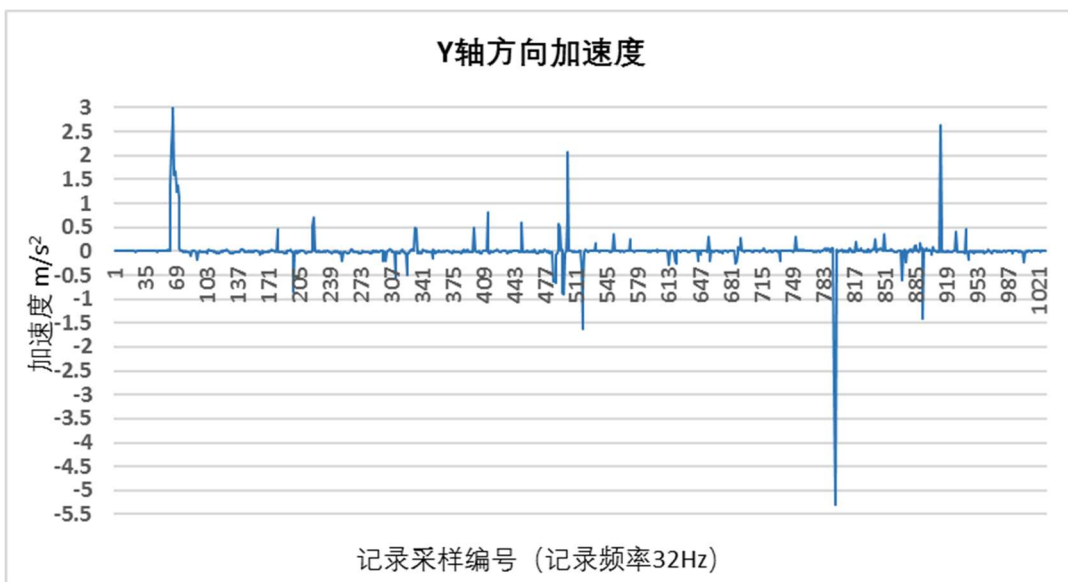


图 5-5 经过转角计算所得的实时转角变化情况

根据第四章机制，结合图 5-4，5-5 两者，系统对加速度基于全局坐标系做正交分解，并根据运动状态进一步处理得到的加速度情况如下：



(a) X 轴方向加速度



(b) Y 轴方向加速度

图 5-6 用于轨迹计算模型的全局坐标系加速度情况

下面给出轨迹计算模块中实时的速度矢量数据，它们同样被分解到 X 轴和 Y 轴两个方向上。由下图 5-7 中显示的速度情况中可以看到，在经过加速态后，计算所得速度有一定范围的波动，这是不可避免的。图中虚线矩形框分别选中了几个区域，我们将他们从左到右依次标为 A，B，C，D 四项，它们代表了对运动过程依据不同状态的切割。

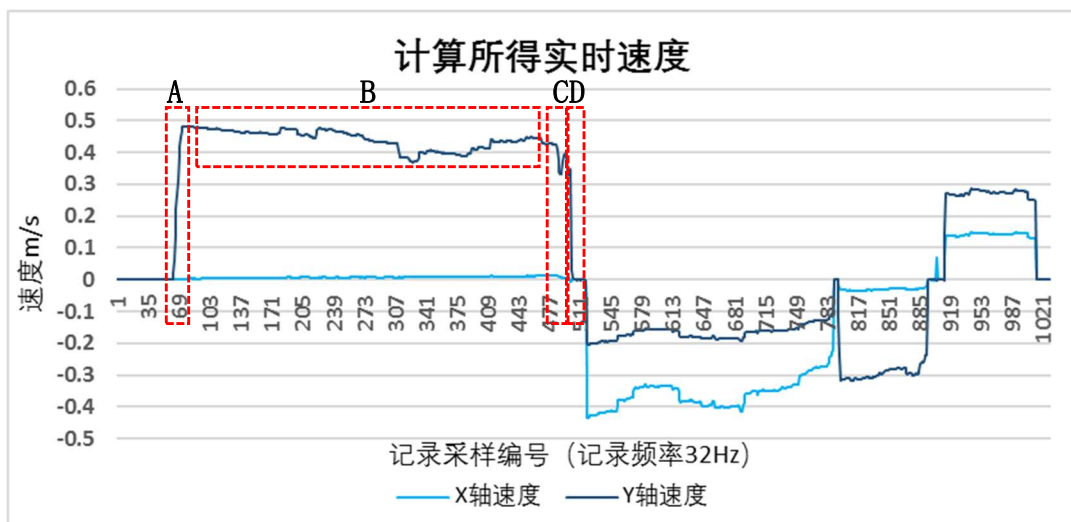


图 5-7 经轨迹计算模型计算所得速度情况

上图中，A 区代表运动由静止态进入加速态，此阶段进行加速；B 区则为机器人运动处于前进态，不过其中可能出现加速或减速态的段时间噪声；C 区为运动进入减速态，但计算模型并没能把速度降到实际的零值，故需要校准；D 区代表旋转态，这时候进行速度校零，两个方向的速度皆被判 0。

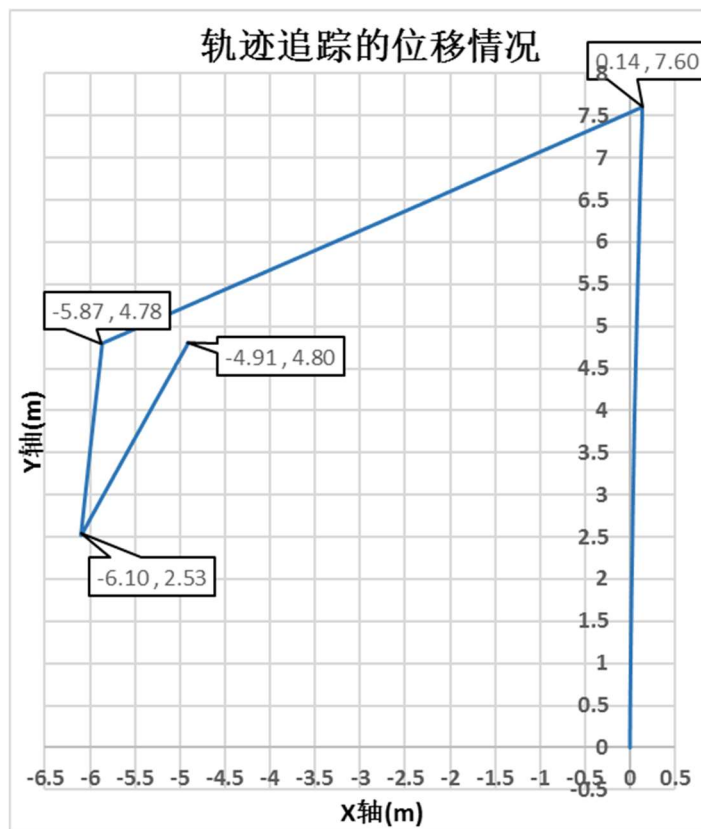


图 5-8 轨迹追踪的位移情况对应连续轨迹

在图 5-8 中给出了四个旋转转向点，这四个点将作为与实际位移情况的对比点，继而可以做简单的误差分析。实验对应的 Create2 机器人真实自旋转点的坐标为：(0.25, 7.20)，(-5.33, 5.02)，(-5.90, 2.36)，(-4.65, 4.14)。

对应的实测运动轨迹应如下图 5-9 所示：

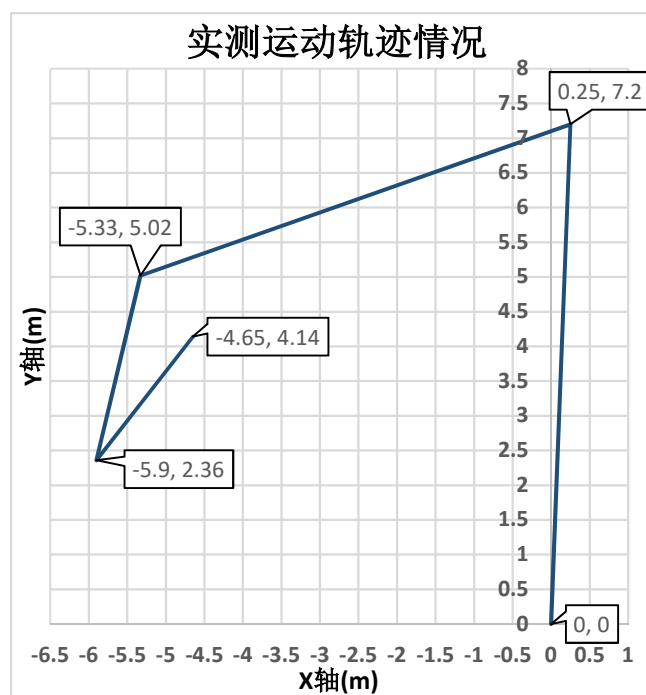


图 5-9 根据实际对比点坐标的运动轨迹

这次实验是众多实验中效果较好的一组，可以看到两张图中，不仅运动轨迹形状基本一致，在绝对长度上也能相近。运动时长约 60s。我们把运动分成四段直线运动来看待，分析直线运动的绝对长度也即前进路程，和相对前一点的转角，我们有：

第一段直线运动，计算所得前进路程为 7.60m，实测前进路程 7.20m，计算转角为 1.05° 而实测转角为 1.98° ；第二段直线运动，计算所得前进路程为 6.64m，实测前进路程 5.99m，计算转角为 116.19° 而实测转角为 113.33° ；第三段直线运动，计算所得前进路程为 2.26m，实测前进路程 2.72m，计算转角为 59.03° 而实测转角为 56.57° ；第四段直线运动，计算所得前进路程为 2.56m，实测前进路程 2.17m，计算转角为 158.17° 而实测转角为 157.02° 。

四段直线运动，前进路程绝对误差值分别为：+0.40m，+0.65m，-0.46m，+0.39m。相对于实测路程的平均误差为 12.82%。转角绝对误差值分别为： -0.93° ，

+2.86° , +2.46° , +1.15° 。在累计转角 328.90° 的情况下累计误差为 5.54° , 偏差 1.68%, 这相对来说已经很理想。综合其他实验, 对于一分钟左右的机器人自由运动, 系统在绝对前进路程方面的偏差在 10%到 25%之间, 而转角的累计偏差在 1.5%到 3.5%之间。

5.4 本章小结

本章是对基于惯性感知的机器人运动轨迹追踪机制的系统实现介绍, 内容涵盖运行开发环境和硬件条件介绍, 系统设计和实验与分析。在系统设计中, 详细阐述了多模块设计, 并展示了系统界面与运行流程。实验与分析中, 基于一次完整实验, 给出了数据与分析, 在最后进行了误差计算, 交代了系统进行轨迹追踪的平均偏差情况。

6 总结与展望

6.1 总结

本文完整的给出了一套基于惯性感知的机器人运动轨迹追踪机制。首先对有客观规律的机器人运动进行研究，提出了对应的轨迹计算模型。又将智能手机绑定于机器人，基于手机的惯性感知传感器数据，提出了对应的操作处理办法，对运动状态进行实时判别，并最终实现了轨迹计算模型，以达成追踪目的。同时，本文工作涵盖了对机制的系统实现，开发了对应的 Android 移动应用程序，以便于实验探究和实际使用。

6.2 展望

惯性轨迹追踪领域的研究会越来越深，越来越广，随着智能手机融入大众生活，与之结合的轨迹追踪工作也将不断涌现。出色的基于惯性感知的轨迹追踪机制将被广泛应用于虚拟现实人机交互等领域，同时将会为三维空间内的自由追踪与动作识别打下理论与实践基础。

本文工作是一个基础性的探究工作，为智能手机相关的惯性轨迹追踪工作做了探索性的贡献，但它仍存在若干不足，如运动状态判别方面时有误判，机制有较多对经验参数的依赖，追踪效果不稳定且偏差较大，系统实现的轨迹绘制部分有待提高等。

相信未来对智能手机相关的惯性轨迹追踪的研究工作会取得丰硕成果！

参考文献

- [1] Clymer M, Graves G. New approaches in magnetic sensing for tracking devices[C]//IS&T/SPIE 1994 International Symposium on Electronic Imaging: Science and Technology. International Society for Optics and Photonics, 1994: 288-296.
- [2] Hu C, Meng M Q H, Mandal M, et al. 3-axis magnetic sensor array system for tracking magnet's position and orientation[C]//Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on. IEEE, 2006, 2: 5304-5308.
- [3] Gebhart S C, Galloway Jr R L, Jansen E D. Optical tracking of the three-dimensional position of an ablative focused laser beam [C]//Biomedical Optics 2003. International Society for Optics and Photonics, 2003: 225-234.
- [4] 胡三庆. 基于MEMS加速度传感器的空间运动轨迹追踪系统设计与实现[D]. 华中科技大学, 2009.
- [5] 杨波. 基于 MEMS 惯性传感器动作捕捉系统与轨迹追踪的研究设计[D]. 电子科技大学, 2014.
- [6] 常磊. 基于复合传感器的运动轨迹追踪技术研究[D]. 沈阳理工大学, 2015.
- [7] 张家田, 王望, 严正国. 基于 MEMS 传感器的运动轨迹的追踪系统[J]. 现代电子技术, 2016, (12): 117-119+122.
- [8] Yun S, Chen Y C, Qiu L. Turning a mobile device into a mouse in the air[C]//Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services. ACM, 2015: 15-29.
- [9] Hu S, Liu B. An Intelligent Device for Non-Restricted 3D Tracking and Drawing[C]//Computational Intelligence and Industrial Application, 2008. PACIIA'08. Pacific-Asia Workshop on. IEEE, 2008, 1: 202-206.

- [10] Google Inc. Android Developers Sensors Overview. https://developer.android.com/guide/topics/sensors/sensors_overview.html, 2017.
- [11] Yang Z, Shangguan L, Gu W, et al. Sherlock: Micro-environment sensing for smartphones[J]. IEEE Transactions on Parallel and Distributed Systems, 2014, 25(12): 3295–3305.
- [12] Li Q, Han Q, Sun L. Collaborative recognition of queuing behavior on mobile phones[J]. IEEE Transactions on Mobile Computing, 2016, 15(1): 60–73.
- [13] Kwapisz J R, Weiss G M, Moore S A. Activity recognition using cell phone accelerometers[J]. ACM SigKDD Explorations Newsletter, 2011, 12(2): 74–82.
- [14] Sun L, Zhang D, Li B, et al. Activity recognition on an accelerometer embedded mobile phone with varying positions and orientations[C]//International Conference on Ubiquitous Intelligence and Computing. Springer Berlin Heidelberg, 2010: 548–562.
- [15] Chen Z, Lin M, Chen F, et al. Unobtrusive sleep monitoring using smartphones[C]//Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2013 7th International Conference on. IEEE, 2013: 145–152.
- [16] Bo C, Zhang L, Li X Y, et al. Silentsense: silent user identification via touch and movement behavioral biometrics[C]//Proceedings of the 19th annual international conference on Mobile computing & networking. ACM, 2013: 187–190.
- [17] iRobot Inc. Create2 Programmable Robot. <http://www.irobot.com/About-iRobot/STEM/Create-2.aspx>, 2015.
- [18] 吴祖堂, 杨德猛, 邹虹. 压电加速度传感器冲击测量中低频失真的理论分析与实验验证[J]. 传感技术学报, 2010, (11): 1586–1589.

致谢

经过十多天的专心撰写，我的毕业论文终于写下了最后一个句号。我的毕设工作历时业已大半年了，期间不免遇到困难，不免走过弯路，不过最终还是取得较为理想的结果，历经风雨，终见彩虹。

在此，首先要感谢我的导师谢磊副教授，从毕设的选题到研究工作的展开再到最后论文的撰写，谢老师都给予了我细心的指导与点拨，不断排忧解难，为我的研究工作指引了方向。在此向谢老师致以最真诚的谢意。

同时也要感谢分布式计算实验室内同一课题组内的师兄师姐，他们对我的研究工作提出了宝贵建议。

感谢我的母亲，经常关心我的科研工作与论文进展，并始终鼓励和支持我，她给了我乘风破浪的勇气和力量。

感谢大学四年里始终相伴的同学与朋友，因为你们，我的大学才会精彩。感谢每一位予我谆谆教诲的老师，普我真理，塑我灵魂。

特别的，感谢张国荣先生的音乐，在我撰写论文的过程中一路相随，春天该很好，你若尚在场。感谢迈克尔·杰克逊，我一生的标杆。