

单位代码: 10293 密 级:

南京邮电大学

专 业 学 位 硕 士 论 文



论文题目: 基于 WiFi 的运动定位和轨迹追踪系统

学 号 1318044908

姓 名 马大禹

导 师 张载龙

专业学位类别 专业硕士

类 型 非全日制

专业（领域） 计算机技术

论文提交日期 2021.4

WiFi Based Motion Positioning and Tracking System

Thesis Submitted to Nanjing University of Posts and
Telecommunications for the Degree of
Master of Master of Engineering



By

Dayu Ma

Supervisor: Prof. Zailong Zhang

April 2021

摘要

由于物联网技术和各种无线通信技术的飞速发展，现如今人们对于定位服务的需求也越来越高，这使得位置服务尤其是室内位置服务的需求在不断增加且要求也在不断提升。由于 WiFi 技术的发展和普及，使用 WiFi 实现定位技术一直是定位领域的研究热点。相比于传统的信号强度，WiFi 信道状态信息(Channel State Information, CSI)由于具有更多的细粒度信息、更好的稳定性和更多的数据，因而成为 WiFi 定位领域的一个研究热点。另一方面，传统的 WiFi 定位为静止场景下的定位，实际应用限制较大。因此，本文主要研究的目标是一个对运动目标进行定位的方案。本文具体的工作内容如下：

(1) 本文设计并实现了一款基于 CSI 中的相位信息的定位算法。该算法会对 CSI 中的相位数据进行误差值处理、离群值剔除、偏移值补偿来降低外部误差对于定位结果的干扰。之后，该算法会使用差分方式来计算 CSI 子载波的相位值，从而计算出无线信号发送设备相对于信号接收设备的距离信息和到达角信息，进而得到电磁波信号接收端和发送端的相对位置。

(2) 使用卡尔曼滤波器，设计和实现一个轨迹追踪算法来对定位目标进行移动轨迹追踪。卡尔曼滤波器不仅能够对运动目标进行移动轨迹的追踪，还能够对直接定位结果的定位误差进行抑制。由于传统卡尔曼滤波器难以处理测试案例中人物运动状态突变的问题，所以本次设计提出了一种改进型的卡尔曼滤波器算法。改进后的卡尔曼滤波器算法能够在人物运动状态突变时依旧能正常工作。由于该算法会导致轨迹追踪的准确度降低，因此又对该算法进行了再次改进。再次改进后的卡尔曼滤波器算法的准确度得到了改善。

(3) 对上述定位算法、轨迹追踪算法进行了程序实现和实验分析。本文在 Linux 平台上搭建了系统原型，并将相关程序内容在 OpenWrt 智能路由器系统中实现。之后根据仿真结果，提出相应测试案例，进行算法功能的测试。测试结果表明：人物处于静止状态下时，定位算法能提供平均误差值为 0.5m 的定位结果；人物处于线性运动状态下时，卡尔曼滤波算法能够提供误差值为 0.05m 的定位结果。

关键词：信道状态信息，WiFi 定位，卡尔曼滤波器，OpenWrt，智能路由器

Abstract

Due to the rapid development of Internet of Things technology and various wireless communication technologies, nowadays people's demand for location-based services is also higher and higher, which makes the demand for location-based services, especially indoor location-based services is increasing and the requirements are also rising. Due to the development and popularization of WiFi technology, using WiFi to achieve positioning technology has always been a hot research in the field of positioning. Compared with the traditional RSSI, WiFi Channel State Information (CSI) has become a hot research in the field of WiFi positioning because it has more fine-grained information, better stability and more data. On the other hand, the traditional WiFi location is usually applied to the static scene, which limits the practical application of this technology. Therefore, the main goal of this paper is to design a scheme to locate the moving target. The specific work of this paper is as follows:

(1) This paper designs and implements a location algorithm based on phase information in CSI. The algorithm can process the phase data in CSI, eliminate outliers and compensate offset to reduce the interference of external errors. Then, the algorithm calculates the phase value of the CSI subcarrier by using the differential method, so as to calculate the distance information and the angle of arrival information of the wireless signal transmitting device relative to the signal receiving device, and then obtain the relative position of the electromagnetic wave signal receiving end and the transmitting end.

(2) In this paper, a trajectory tracking algorithm is designed and implemented by using Kalman Filter. Kalman Filter can not only track the moving target trajectory, but also suppress the positioning error of direct positioning results. Because the traditional Kalman Filter can hardly to deal with the problem of the mutation of human motion state in the test case, this paper proposes an improved Kalman Filter algorithm. The improved Kalman Filter algorithm can still work normally when the human motion state changes suddenly. But the algorithm will lead to the accuracy of trajectory tracking is reduced, so the algorithm is improved again. The accuracy of the twice improved Kalman Filter algorithm is improved.

(3) In this paper, the program implementation and experimental analysis of the above positioning algorithm and trajectory tracking algorithm are carried out. In this paper, the system prototype is built on Linux platform, and the related program content is implemented in OpenWrt

intelligent router system. Then, according to the simulation results, the corresponding test cases are proposed to test the algorithm function. The test results show that the algorithm can provide a positioning result with an average error of 0.5m when the person is in a static state. When the person is in linear motion, the Kalman filter algorithm can provide a positioning result with an error of 0.05m.

Key words: CSI, WiFi Localization, Kalman Filter, OpenWrt, Intelligent Router

目录

专用术语注释表	VI
第一章 绪论	1
1.1 研究背景和意义	1
1.2 国内外发展状况	2
1.2.1 CSI 定位技术的发展状况	2
1.2.2 卡尔曼滤波器的发展状况	4
1.3 研究内容	5
1.4 章节安排	5
第二章 相关背景知识介绍	7
2.1 无线定位技术理论	7
2.1.1 基于角度的无线定位技术	7
2.1.2 基于距离的无线定位技术	7
2.1.3 基于距离和角度的无线定位技术	9
2.2 OFDM 系统和 CSI 技术原理	9
2.3 Atheros 网卡中的 CSI 的获取	11
2.3.1 CSI 的硬件处理流程	11
2.3.2 CSI 的数据的收集流程	12
2.3.3 CSI 的数据结构	13
2.4 本章小结	14
第三章 使用 CSI 中相位信息的定位算法	15
3.1 CSI 相位数据的处理	15
3.1.1 2π 缠绕处理	15
3.1.2 基于方差分布的误差过滤算法	16
3.2 基于差分计算 CSI 相位的定位算法	21
3.2.1 距离计算算法	21
3.2.2 到达角计算算法	23
3.3 仿真结果	24
3.4 本章小结	26
第四章 基于卡尔曼滤波器的轨迹追踪算法	27
4.1 传统卡尔曼滤波器	27
4.1.1 卡尔曼滤波器的原理	27
4.1.2 卡尔曼滤波器的使用	28
4.2 卡尔曼滤波器的改进	29
4.2.1 惯性状态算法	29
4.2.2 状态量化算法	31
4.3 仿真结果	32
4.4 本章小结	33
第五章 框架设计与实验分析	35
5.1 框架设计	35
5.1.1 定位信息获取系统的框架设计	35
5.1.2 轨迹追踪系统的框架设计	35
5.1.3 总体流程框架	36
5.2 测试准备	36

5.2.1 OpenWrt 版本选择..... 36

5.2.2 OpenWrt 编译..... 37

5.3 程序效果测试 38

5.3.1 测试场景模型 38

5.3.2 测试操作步骤 39

5.3.3 测试结果评估 40

5.4 本章小结 42

第六章 总结与展望 44

6.1 本文总结 44

6.2 后续展望 45

参考文献 46

附录 1 程序清单 49

附录 2 攻读硕士学位期间申请的专利 50

致谢 51

专用术语注释表

符号说明:

$h(t)$	无线信道的信道冲激响应
$H(f)$	无线信道的信道频域响应
f_i	编号为 i 的子载波的频率值
f_c	信道的基波的频率值
Δf	相邻子载波的频率差值
P_i	编号为 i 的子载波在发射端的相位值
$P_{i,j}$	编号为 i 和 j 的子载波的相位差值
x_k	状态系统在时刻 k 时的系统状态
u_k	状态系统在时刻 k 时的外部控制向量
m_k	状态系统在时刻 k 时的输入噪声
y_k	时刻 k 时的系统状态的观测值
n_k	时刻 k 时的系统状态的观测值的观测噪声
A_k	线性高斯系统在 k 时刻的状态转移矩阵
B_k	线性高斯系统在 k 时刻的控制矩阵
C_k	线性高斯系统在 k 时刻的观测矩阵
\hat{x}_t^-	卡尔曼滤波器在 t 时刻对系统状态进行估计的估计值
\hat{c}_t^-	卡尔曼滤波器在 t 时刻对系统中的噪声进行估计的估计值
K_t	卡尔曼滤波器在 t 时刻的卡尔曼增益
\hat{x}_t	卡尔曼滤波器在 t 时刻对系统状态的进行预测的预测值
\hat{c}_t	卡尔曼滤波器在 t 时刻对系统的噪声状态进行预测的预测值
$p_{x,k}$	被测目标在时刻 k 、 x 方向上的坐标状态
$p_{y,k}$	被测目标在时刻 k 、 y 方向上的坐标状态
$v_{x,k}$	被测目标在时刻 k 、 x 方向上的速度状态
$v_{y,k}$	被测目标在时刻 k 、 y 方向上的速度状态
$f_p(\cdot)$	对系统状态中的坐标状态进行计算的映射
$f_v(\cdot)$	对系统状态中的速度状态进行计算的映射
$A_{p,k}$	线性高斯系统在 k 时刻、位置状态向量上的状态转移矩阵
$m_{p,k}$	线性高斯系统在 k 时刻、位置状态向量上的输入噪声
$A_{v,k}$	线性高斯系统在 k 时刻、速度状态向量上的转移矩阵
$n_{p,k}$	线性高斯系统在 k 时刻、速度状态向量上的输入噪声

缩略词说明:

IEEE	Institute of Electrical and Electronic Engineers	电气与电子工程师学会
RSSI	Receive Signal Strength Indicator	接收信号强度指示器
AoA	Angel of Arrive	到达角
ToA	Time of Arrive	到达时间
TDoA	Time Difference of Arrive	到达时间差
ToF	Time of Flight	飞行时间
CSI	Channel State Information	信道状态信息
OFDM	Orthogonal Frequency Division Multiplex	正交频分复用
LTE	Long Term Evolution	通用移动通信技术的长期演进
AP	Access Point	访问接入点
IFFT	Inverse Fast Fourier Transform	快速傅里叶逆变换
FFT	Fast Fourier Transform	快速傅里叶变换
DLNA	Digital Living Network Alliance	数字生活网络联盟
NAS	Network Attached Storage	网络附属存储

第一章 绪论

1.1 研究背景和意义

由于移动互联网的迅速发展,越来越多的传统行业使用移动互联网来进行产业升级。因此各行各业对于定位服务的需求也在不断增加^[1]。不论是出门旅游还是工厂运货,定位服务都是不可或缺的一部分。近年来,各种各样的定位方式层数不穷,不同定位方式的使用场景、定位效果一直是相关研究中的热门话题。

定位系统可以笼统的分为两种:室外场景下的定位和室内场景下的定位。这两种定位场景下定位系统有着显著不同。全球定位系统是目前使用最广泛的室外定位系统之一,已经成为了很多应用的基础服务^[2]。此外,一些其他的卫星定位系统,如北斗卫星导航系统、伽利略卫星导航系统和格洛纳斯卫星导航系统等,也都在室外定位的场景中有着广泛的应用^[3]。

相比成熟的室外定位技术,室内场景下的定位技术发展速度较为缓慢。由于室内场景具有复杂的无线传播环境,导致室内场景下的定位结果会受到多方干扰。即便如此,室内定位技术依旧具备重大应用需求与前景^[4]。

室内定位系统有着很大的应用空间和应用需求。常见的应用场景包括大型公共场馆内的人员定位,医院等建筑内的人员位置跟踪,工厂中的物流状态检查等等。目前已经有了基于各种信号、各种原理的定位系统并有着广泛的应用^[5]。

目前常见的室内定位技术包括超宽带定位技术、射频识别定位技术、WiFi 定位技术、红外定位技术、蓝牙定位技术、超声波定位技术、可见光定位技术、视觉定位技术、ZigBee 定位技术、地磁定位技术等^[6-15]。这些定位方式由于其各自的特点不同,而被用于不同的定位场景中。如射频识别定位技术多用于工厂管理、仓储物流等场景中。由于射频本身无法被用于通信,且该项技术在工程实现上有一定的难度,限制了射频识别定位技术的应用范围。超宽带定位技术由于有着很高的时间分辨率,因而有着超高的定位精度。但是由于超宽带信号传输距离短、穿透性差,并且部署设备成本高,因而难以进行大规模的覆盖和使用。本次设计将会研究 WiFi 定位系统。相较于其他定位系统而言,WiFi 定位技术有着广泛的设施基础,并且相关设备部署成本低廉。此外因为广泛的应用基础和大量的相关研究内容,使得 WiFi 信号的精细度、分辨率、带宽等方面不断提升^[16]。这为 WiFi 定位系统提供了广阔的前景与庞大的市场。

目前,WiFi 定位技术领域已经有着不少的优秀定位方案。按照定位实现方式划分,这

些方案可以在大体上分为 AoA、ToA、TDoA、ToF、指纹匹配等方式^[17]。其中, 基于接收信号强度(Receive Signal Strength Indicator, RSSI)的指纹匹配式定位方案由于适用范围广, 是目前使用最广泛的定位方案。而在 802.11n 协议被提出后, 研究人员开始研究使用信道状态信息(Channel State Information, CSI)实现定位方案。相比于 RSSI, CSI 能够提供更加稳定、更加准确、更加细粒度的信息^[18]。

如今针对 CSI 构建的室内定位方案正在不断增多, 得到的定位结果的精度也说明了其将会是下一代 WiFi 定位方案的最好选择。但是目前, 由于 CSI 定位方案有着计算流程复杂、定位周期长的问题, 而多被用于进行静止场景下的定位研究。但是实际生产生活中很少会有这样的使用场景, 针对此问题, 本次设计会将 CSI 定位方案与卡尔曼滤波器相结合, 设计一个用于对运动物体进行定位和轨迹追踪的方案。

1.2 国内外发展状况

1.2.1 CSI 定位技术的发展状况

目前国内外的学者提出的 CSI 定位技术定位方案, 可分为以下几种:

1. 基于 CSI 的测距定位技术

2012 年, Kaishun Wu 的团队实现了 FILA 系统^[19]。该系统对 CSI 进行傅里叶逆变换和平方处理后得到功率时延谱, 然后使用修正后的功率传播模型实现距离的计算。相较于传统的 RSSI 的距离计算方式, 该算法计算得出的距离精度值有着显著的提高。与之类似的方案还有 CUPID 等。

2016 年, Deepak Vasisht 的团队设计出了 chronos 系统^[20]。该系统会收集不同频带下的 CSI 相位并进行相位矫正, 之后根据不同频带下电磁波频率值不一致的特点, 使用余数定理来计算整体的频谱图。之后通过非均匀的离散傅里叶变换来计算 CSI 信号的时域图, 从而获得信号的 ToF 值。最后使用接收端多天线互相验证的方式, 计算出接收端到发送的距离信息。该系统的定位精度达到了分米级别。

2017 年, Kotaru 等人提出了一个 WiCapture 系统^[21]。该系统通过主动进行数据传输的方式来获取连续的 CSI 相位变动信息, 从而获得不同的位置下物体移动对 CSI 相位的影响。该算法通过精确的相位测量和误差补偿, 能够实现亚厘米级别的位置定位并且识别出物体的移动状态。

2019 年, Shuai Han 等人实现了一种室内定位方案 S-Phaser^[22]。该方案能够结合两个不同

WiFi 信道内信道的时域功率值来进行信号时延的计算。并且能够通过几何定位的方式对目标的位置进行立即定位。该方案定位误差为 1.5m。

2. 基于 CSI 的测角定位技术

2013 年, Xiong 等人使用改装的商用无线网卡实现了一个名为 ArrayTrack 的硬件测量平台^[23]。该平台使用 8 根天线来进行室内场景下的 CSI 定位。其作者在该平台上, 使用空间平滑、阵列几何加权和 AoA 谱合成的技术成功获得了精度为 50cm 的定位结果。

同年, Gjengset 等人通过网卡拼接的方式实现了 Phaser 系统^[24]。该系统通过使多个常规商用网卡共用天线的方式来实现对无线网卡的的天线数进行拓展。此外, 此系统中还使用了一种相位自动校准算法, 用于处理不同无线电振荡器之间的相位误差。当采用 4 个无线网卡拼接时, 该系统能够获得平均精度小于 1m 的定位结果。

2015 年 Kotartu 等人实现了 SpotFi 系统^[25]。该系统仅使用具有三根天线的商用网卡且不需要对其进行额外的硬件更改就可以实现精确定位。SpotFi 利用了不同 CSI 子载波的相位测量值, 虚拟的扩展了天线的数量来实现超分辨率的 AoA 估计。

2017 年, Wei 等人的团队提出了一种稀疏恢复的算法来进行室内环境下的 AoA 估计^[26]。2019 年, Zhang 等人对稀疏恢复算法进行了改进, 并设计了一个 L 型天线阵列实现对三维空间下的 AoA 进行估计^[27]。该算法能够在三维空间中进行分米级别精度的定位。

3. 基于 CSI 的混合定位技术

基于 CSI 的直接定位方案各自都有着各自的缺点。基于 CSI 的测角的定位方案实现相对简单, 但是在处理接收端设备之间的角度关系时, 操作较为复杂; 基于 CSI 的测距的定位方案能够提供较高精准度, 但是需要对收发双方的时间精度要求较高。此外, 这两种方案都需要多台设备才能对被测目标的位置进行定位。因此有人结合了两方案的优点, 提出了基于 CSI 的混合定位技术^[28]。

混合定位技术是指将两种或多种定位方法相互混合实现定位的估计方法。目前最常用的混合定位方式是采用 ToF 和 AoA 相结合的方式。这种方式的优点是将两种定位方案的优点互补, 在定位误差较小时, 能够得到更加精准的定位结果。并且该项技术可以和多天线技术相配合, 增加定位的精度。

Zhu 等人借助混合定位技术, 提升了传统定位算法的准确率^[29]。Zhu 的团队对 MUSIC 算法的分辨率进行改进, 使用跳频的方式扩大了 WiFi 信道的带宽。他们将获得到的结果进行二维估计, 并使用测量功率延迟和估计聚类的方式来识别直达路径。最后, 他们将 ToF 结果和 AoA 结果混合, 使用一种几何算法来得到目标的最终位置。

Afaz 等人提出了一种改进的 MMP 算法^[30]。Afaz 等人在商用网卡上收集不同天线上的 CSI 子载波数据。他们使用 MMP 算法将这些数据进行分组划分, 然后根据分组结果来计算受到多径影响的 ToF 和 AoA 的值。MMP 算法在结果精度与传统算法相同的条件下, 定位计算速度提升了 200 倍。

4. 基于 CSI 的指纹定位技术

指纹定位方法是一种信号特征和与具体位置相关联的定位方法。相比于 AoA 等定位技术, 指纹定位有着易实现、精度高等优点。WiFi 信号由于容易获得而常被用于指纹识别的特征信号。CSI 由于其稳定性强, 空间分辨率高而被用于指纹定位技术。CSI 信号中包含幅值和相位两方面的信息, 指纹定位技术会使用 CSI 幅值信息作为特征信息。不过近年来, 也有团队使用 CSI 中的相位信息作为指纹定位的识别信息。

2014 年, Chen 等人采用 RSSI 和 CSI 相结合的方式来进行指纹定位^[31]。在室内布置杂乱的条件下, 该方案中 90% 的测试点的定位误差均在 2m 以内。

2018 年, Wang X 等人设计的 DeepFi 系统使用深度学习的方式来进行 CSI 特征的提取^[32]。DeepFi 系统会进行两部分的工作: 离线阶段下的学习工作和在线阶段下的识别工作。DeepFi 系统在离线阶段会记录下参考位置的指纹特征, 构建指纹数据库; 而在在线阶段, 系统将接收到的信号特征去匹配一个数据库中的信号特征。DeepFi 系统在两种典型的室内定位场景下的表现均优于其他指纹定位系统。

2019 年, Dang 的团队提出了一种名为 FapFi 的室内人体定位方法, 该方法将 CSI 幅度和相位信息融合, 使用进行融合处理后的幅度和相位信息作为指纹识别特征^[33]。实验结果表明, 该方法能很好抑制室内场景下的多径效应, 在实验室和会议室中分别能够获得 0.5m 和 1.2m 的平均定位误差。

1.2.2 卡尔曼滤波器的发展状况

卡尔曼滤波器是由 Rudolf Emil Kalman 于 1960 年在整理阿波罗计划中相关技术时所整理归纳出的理论。由于卡尔曼滤波器在计算量、结果精确度、计算参数需求上都优于当时的其他的系统状态预测算法。因而其一出现就受到了大量地关注并且不断的有人对其进行优化, 形成了现在多种多样的改进型卡尔曼滤波器。为了解决卡尔曼滤波器的在非线性系统中的表现不佳的问题, 扩展卡尔曼滤波器、无迹卡尔曼滤波器等算法先后被提出。

何康辉、董朝阳等人提出了一种基于正交变换的五阶容积卡尔曼滤波器算法^[34]。该算法利用了三角函数的正交性特点, 对原有的容积点进行适当的容积变换, 将变换后的容积点代

入容积卡尔曼滤波器模型中。仿真验算表明,该算法的结果精确度明显优于传统的五阶卡尔曼滤波器。

李开龙等人为了解决四元数无迹卡尔曼滤波器的计算量较大的问题,将四元数与修正罗德里格斯参数进行线性变换^[35]。该算法避免了四元数与误差罗德里格斯参数频繁切换所带来的计算量增大的问题。作者在捷联惯性导航系统中对该算法进行了仿真实验,确认了算法的效果和精确度。

Munkhzul Enkhtur 等人提出了一种多速率的导航系统,通过减小测量结果更新时系统状态变量的概率分布的变化速率来降低无迹卡尔曼滤波器的计算量^[36]。该算法使得基于无迹卡尔曼滤波器的导航算法能够在低阶或中级微控制器中实现。最后作者通过模拟计算的方式,确认了该算法的优点。

Benzerrouk Hamza 等人为了优化导航系统的适应性,提出了一种将线性卡尔曼滤波器和非线性卡尔曼滤波器相结合的算法^[37]。该算法同时还结合扩展卡尔曼滤波器、西格玛点卡尔曼滤波器和分差卡尔曼滤波器的适应性和鲁棒性,能够实现对 GNSS 导航结果的优化。

1.3 研究内容

本文就 WiFi 网络下使用智能路由器实现的无线定位技术进行研究。具体内容为研究一种无线定位算法和一种运动轨迹追踪算法。要求研究的算法分别能够对运动目标进行高精度的定位和处理常规的线性运动状态突变的问题。本文主要的研究内容如下:

(1) 研究一种使用 CSI 进行混合定位的算法。该算法会首先对接收到的 CSI 数据进行误差值抑制、离群值剔除、缺失值补偿。然后此算法会使用这些 CSI 数据分别进行被定位目标的 ToF 和 AoA 的计算。最后此算法使用几何计算的方式,计算被测目标的位置。

(2) 研究一种适用于 WiFi 定位场景下的运动目标移动轨迹追踪算法。此算法会使用卡尔曼滤波器对运动中的目标进行移动轨迹的追踪,并能够根据追踪到的运动状态结果降低位置定位的结果的误差。此算法同时还对卡尔曼滤波器进行了改进,解决了标准卡尔曼滤波器在运动状态突变的场景下表现不佳的问题。改进后的卡尔曼滤波能够在系统状态发生变换后依旧保证较好的目标位置追踪和定位误差降低效果。

1.4 章节安排

本文的主要研究内容阐述如下:

第一章为绪论。这章介绍了本次设计的产生背景和本次研究的价值意义。这章主要介绍室内定位技术的出现背景、CSI 技术的发展和现状。并对当前国内外的 CSI 定位技术进行分类,介绍了几种常见的 CSI 定位技术的国内外研究状况。同时,还对卡尔曼滤波器的的一些国内外研究结果做了相应介绍。最后介绍了本次设计的研究内容和本论文的章节安排。

第二章为相关背景知识介绍。这章会分别介绍几种常见的定位技术的原理,重点介绍 CSI 直接定位技术的定位原理以及混合式 CSI 定位技术的定位原理。之后本章会对 CSI 技术的技术原理进行相关介绍。最后本章会以一款具体的硬件为例介绍硬件设备中的 CSI 的相关内容:介绍 CSI 是怎样从硬件中产生的;介绍 CSI 是如何在软件中进行传递的;介绍 CSI 被获取后需要以什么样的方式对其进行读取。

第三章中提出了一种使用 CSI 中相位信息进行定位算法。这章提出了一种基于相位 CSI 定位算法。该算法会对 CSI 相位数据进行差值计算,绕过了原始 CSI 相位值数据难以获得的问题。该算法具体又包括基于 CSI 相位误差过滤算法、基于 CSI 中相位的距离测量算法、基于 CSI 中相位的到达角测量算法。之后本章对这些算法依次进行仿真场景下的测试,并给出相应的分析结论。

第四章中提出了一种改进型的卡尔曼滤波器算法。这章首先会介绍传统卡尔曼滤波器的原理,然后提出传统卡尔曼滤波器会在系统状态突变时产生较大的误差。之后这章会介绍传统卡尔曼滤波器的改进算法:惯性状态算法。由于惯性状态算法的平均误差值较高,所以本章又对惯性状态算法进行了改进,提出了状态量化算法。最后这章会给出这三种算法在测试场景下的表现,并分别给出相应的分析。

第五章对前两章提出的算法进行编程语言实现并给出实验测试结果。这章首先会展示实验程序的算法流程图,之后会给出实验测试中具体测试场景模型、测试操作步骤。最后本章会给出实验测试的测试表现、测试结果、测试结论。

第六章为总结与展望。总结全文工作,阐述了本文的主要特色和创新点,并就目前实验过程中还需改进的方面做出了说明。

第二章 相关背景知识介绍

2.1 无线定位技术理论

2.1.1 基于角度的无线定位技术

早期无线定位技术由于测距技术相对不成熟，通常会采用基于 AoA 的定位模型。现在由于相应技术的发展，基于 AoA 的定位技术由于分辨率高，常被用来与其他定位技术进行联合使用。基于 AoA 的定位技术会通过待定位点与多个 AP 之间的到达角之间的几何关系进行位置估计。原理如图 2.1 所示：

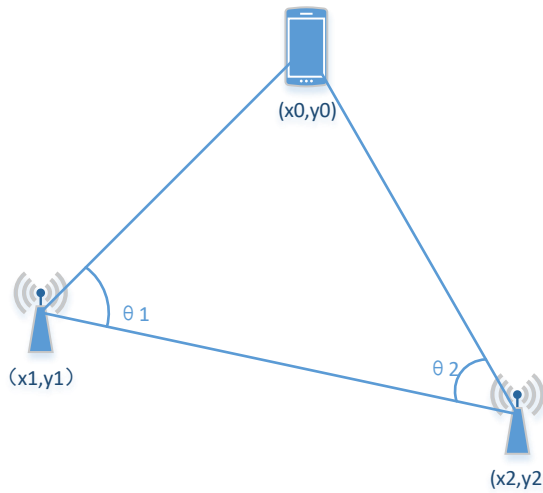


图 2.1 基于角度的无线定位模型

如图 2.1 所示，使用 AoA 进行定位，需要至少两个 AP 分别测量待定位点的 AoA 信息，之后通过计算两个 AP 点的射线交点来计算获得待定位点的位置信息^[38]。AoA 定位技术的定位算法如下：已知两个 AP 点的坐标分别为 (x_1, y_1) 和 (x_2, y_2) ，待定位点相对于两个 AP 的 AoA 值分别为 θ_1 和 θ_2 ，则待定位节点的坐标值 (x_0, y_0) 可以分别用以下公式进行计算：

$$x_0 = \frac{x_1 \tan \theta_2 + x_2 \tan \theta_1 + |y_2 - y_1| \tan \theta_1 \tan \theta_2}{\tan \theta_1 + \tan \theta_2} \quad (2.1)$$

$$y_0 = \frac{y_1 \tan \theta_2 + y_2 \tan \theta_1 + |x_2 - x_1| \tan \theta_1 \tan \theta_2}{\tan \theta_1 + \tan \theta_2} \quad (2.2)$$

2.1.2 基于距离的无线定位技术

三边定位法是常见的测距式定位算法。三边定位法的使用场景是已知三个及以上参考位

置坐标和待定位点到各个参考位置坐标的欧式距离^[39]。使用 ToA 和 RSSI 传播模型获得测距结果后，将结果带入到三点定位算法中来进行待定位点的位置计算。该算法的理想模型如图 2.2 所示：

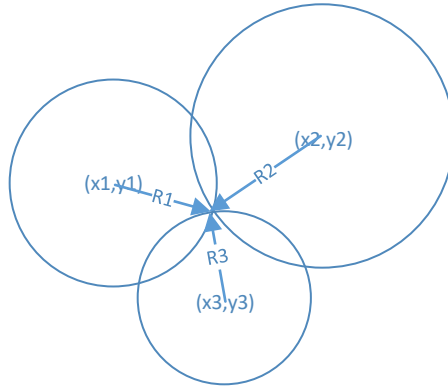


图 2.2 基于距离的无线定位(理想状况)

设待定位点到三个参考点的距离分别为 R_1 、 R_2 、 R_3 ，设三个参考点的的坐标值分别为 (x_1, y_1) 、 (x_2, y_2) 、 (x_3, y_3) ，设待定位点的坐标为 (x_0, y_0) ，则可以得到如下的联立方程组：

$$(x_1 - x_0)^2 + (y_1 - y_0)^2 = R_1^2 \quad (2.3)$$

$$(x_2 - x_0)^2 + (y_2 - y_0)^2 = R_2^2 \quad (2.4)$$

$$(x_3 - x_0)^2 + (y_3 - y_0)^2 = R_3^2 \quad (2.5)$$

但是在实际场景中由于干扰误差的存在，导致距离测量结果并非准确值，此时的场景模型如图 2.3 所示：

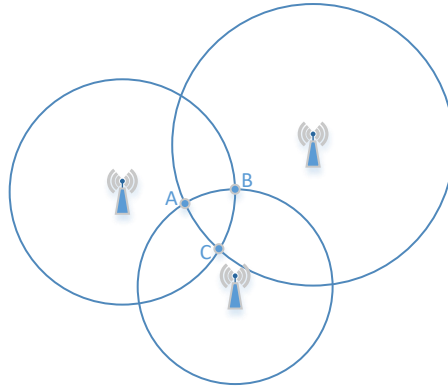


图 2.3 基于距离的无线定位(实际状况)

针对实际情况下误差问题，国内外一些人员推出了各种各样的优化算法，这些算法会根据各种误差与先验信息条件来逐步求解出待定位点的坐标。这类算法的常见的方法是通过泰勒级数展开的方法进行近似求解，同时也有使用牛顿迭代法、子空间法、加权最小二乘法等进行求解的算法^[40]。

2.1.3 基于距离和角度的无线定位技术

基于距离和角度的混合式定位技术的优势是可以构建单 AP 的定位方案。本次设计中使用的定位信息获取方案的最终一步将会采用这种技术。图 2.4 展示了此种算法的场景模型：

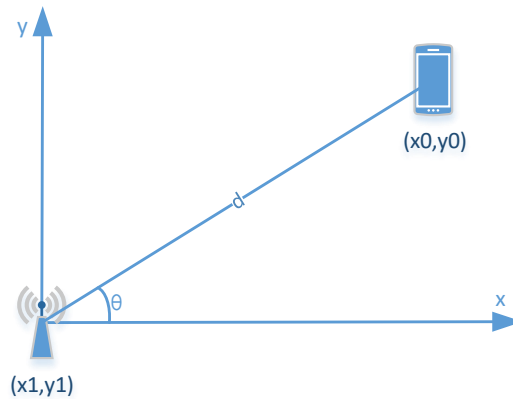


图 2.4 混合定位模型

基于距离和角度的无线定位技术的算法如下：已知 AP 点的坐标为 (x_1, y_1) ，设待定位点的坐标为 (x_0, y_0) 。在通过相应的算法计算并同时获取待定位点到 AP 点的直射径到达角和直射径距离分别为 θ 和 d 时，待定位点坐标可以通过如下公式计算：

$$x_0 = x_1 + d \times \cos\theta \quad (2.6)$$

$$y_0 = y_1 + d \times \sin\theta \quad (2.7)$$

2.2 OFDM 系统和 CSI 技术原理

由于 CSI 技术是本文研究的核心内容之一，所以本文需要对 CSI 的技术原理进行一定的介绍。

目前，正交频分复用技术被广泛用于无线局域网、LTE 网络的来提高信道利用率、增加网络的数据传输速率。此项技术的核心内容主要是在频域上将整个信道分隔成若干个子载波^[41]。这些子载波之间相互重叠但又互不影响，因此可以通过正交的方式方便的进行调制和解调。相比于传统的频分复用技术，正交频分复用技术能够在有限的带宽内大幅度的增加用于携带信号的子载波个数，从而大大提升了信道的传输容量。

OFDM 技术的另一个优点是其可以通过快速傅里叶变换来实现对信号的快速调制和解调。在 OFDM 系统中，发射端会首先将所有待传输的数据调制在各个子载波上，之后 OFDM 系统会对这些正交的子载波进行快速傅里叶逆变换(IFFT)以将其转换为时域上的信号。在 IFFT 执行过程中，复杂的时间域样本的数据进行进一步的正交混合，之后还会在其中插入时钟脉

冲以防止干扰^[42]。在完成 IFFT 后，数据将会通过 D/A 转换变成电磁波信号向四周发送。当接收端接收到相应的电磁波信号后，则会进行相反的操作后使用数字调制的方式来获得其中携带的信息。图 2.5 分别表示了 OFDM 系统中接收端和发射端的各自处理流程。

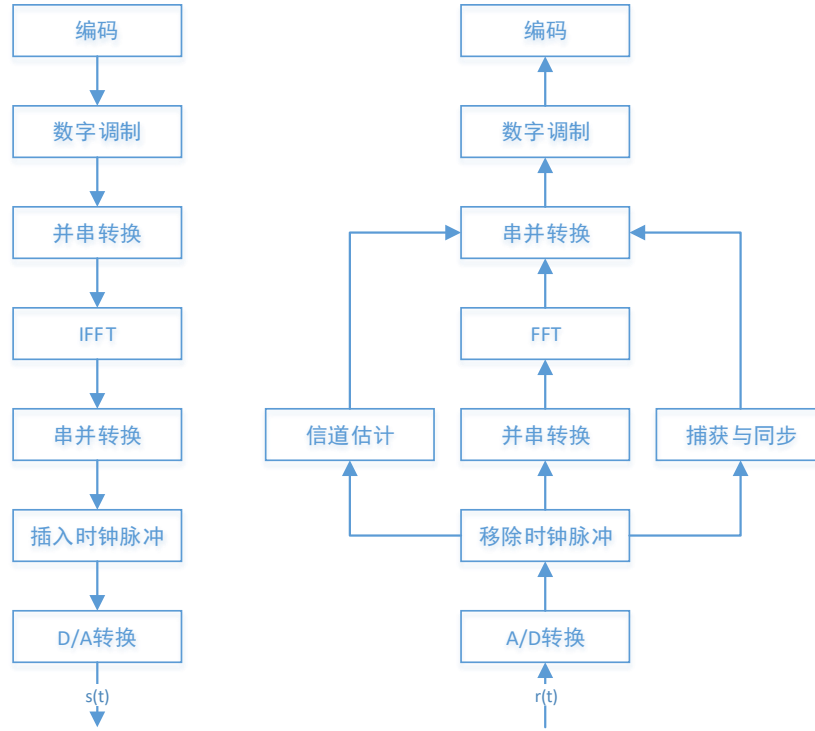


图 2.5 OFDM 系统(左: 发射端, 右: 接收端)

由于 OFDM 系统中用于传输数据的若干子载波会分别进行解调，所以 OFDM 系统中信号的信道状态信息会形成一个矩阵，该矩阵中包含了一条射频链路上每个子载波经历的信道响应。这种信道响应矩阵同时描述了电磁波信号在时域和频域上的特征，相当于将 WiFi 信息从一维拓展到了二维。

从数学的角度分析，假设 OFDM 将信道划分为 N (N 为偶数) 个正交的子载波，且该信道的带宽为 B ，则在一个符号周期 $(t_0, t_0 + T]$ 中，经过 OFDM 调制后的发射信号 $x(t)$ 可以表示为：

$$x(t) = \sum_{i=-\frac{N}{2}}^{\frac{N}{2}-1} A_i e^{j \cdot 2\pi f_i (t-t_0)} \quad (2.8)$$

其中 A_i 表示数据经过符号映射之后的数据， f_i 表示对应子载波的频率。接收端收到的接收信号 $y(t)$ 由发射信号 $x(t)$ 与信道冲激响应 $h(t)$ 进行卷积产生。 $y(t)$ 表达式如下：

$$y(t) = h(t) \otimes x(t) = \int_{-\infty}^{+\infty} h(\tau) \cdot x(t - \tau) d\tau \quad (2.9)$$

对公式 2.9 进行傅里叶变换后，可以得到接收信号的频域表达式：

$$Y(f) = H(f) \cdot X(f), f \in \left[f_{-\frac{N}{2}}, f_{\frac{N}{2}-1} \right] \quad (2.10)$$

其中 $Y(f)$ 为 $y(t)$ 的频域表示， $X(f)$ 为 $x(t)$ 的频域表示。 $H(f)$ 为信道频率响应，是信道冲激

响应 $h(t)$ 经过傅里叶变换后得到的结果。假设电磁波信号从发射端天线经过 L 条不同的传播路径到达接收端天线, 则 OFDM 信号下单个子载波的信道频率响应可以用以下方式表示:

$$H(f) = \sum_{n=1}^L a_n e^{-j \cdot 2\pi \cdot f \cdot \tau_n} \quad (2.11)$$

其中, a_n 表示信号经过第 n 条路径后衰减的幅度值, τ_n 表示第 n 条路径的信号时延值。在 OFDM 系统中, CSI 矩阵 \mathbf{H} 表示的是单频带内各个子载波信号在频域内的采样结果。设 OFDM 系统中发射端天线数为 1, 接收天线数为 M , 则 \mathbf{H} 可以表示为如下形式:

$$\mathbf{H} = \begin{bmatrix} H_{1,1} & H_{1,2} & \cdots & H_{1,N} \\ H_{2,1} & H_{2,2} & \cdots & H_{2,N} \\ \vdots & \vdots & H_{m,n} & \vdots \\ H_{M,1} & H_{M,2} & \cdots & H_{M,N} \end{bmatrix} \quad (2.12)$$

其中, $H_{m,n}$ 表示发射天线与第 m 根接收天线在第 n 号子载波上的通信链路的信道状态, 其数值的格式为复数形式, 数值的大小为该号子载波相对于网卡内部的参考电平的采样测量值。 $H_{m,n}$ 可以具体表示为如下形式, 其中 $A_{m,n}$ 和 $\varphi_{m,n}$ 分别表示接收端第 m 根天线上的第 n 号子载波的信道幅度增益和信道相位偏移。

$$H_{m,n} = |H_{m,n}| \cdot e^{j \cdot \angle \varphi_{m,n}} = A_{m,n} \cdot e^{j \cdot \angle \varphi_{m,n}} \quad (2.13)$$

目前常用的商用设备一般使用三根接收天线来接收电磁波信息, 因此商用设备接收到的 CSI 信息矩阵 \mathbf{H} 通常只有三行。另外, 由于 CSI 信息矩阵中的 $H_{m,n}$ 并不是实际的电平值, 而是采样值基于接收机电平值的比较值, 因此在进行信道频率响应的理论值计算时, 还需要考虑噪声功率和天线增益的影响:

$$\text{CFR} = \mathbf{H} \cdot \sqrt{\text{SNR}} = \mathbf{H} \cdot \sqrt{\frac{P_{\text{RSSI}} \cdot \overline{P_{\text{CSI}}}}{P_{\text{noise}}}} \quad (2.14)$$

其中, CFR 表示信道状态响应的理想值, P_{RSSI} 表示从 CSI 数据包中获得 RSSI 功率值与自动增益控制进行计算后得到接收信号的功率。 $\overline{P_{\text{CSI}}}$ 表示 CSI 的平均功率, P_{noise} 表示噪声功率。

2.3 Atheros 网卡中的 CSI 的获取

2.3.1 CSI 的硬件处理流程

当射频信号被天线接收到后, 会经历以下步骤

- (1) 使用低噪声放大器对原始信号进行放大。
- (2) 将放大后的电磁波信号与其自身所处信道的基波进行乘法计算, 将所接收信号的中

心频率变为 0。之后该信号会经过一个可编程放大器，进行选择性的信号幅值放大。

(3) 将信道的基波进行与 $\pi/2$ 的相移后，再次进行将其与步骤 1 所得到的信号进行相乘，得到另一组中心频率为 0 的信号。之后该信号会通过另一个可编程放大器，进行选择性的信号幅值放大。

(4) 将步骤 2 与步骤 3 所得到信号分别进行模数转换后相加。在相加步骤中，以步骤 2 中获取的信号作为实数域的信号，以步骤 3 中所获取到的信号作为虚数域的信号。实数域信号与虚数域的信号相组合后，就能够得到以时域形式表示的射频信号^[43]。

(5) 对获取到的时域信号进行能量检测，以确认所接收到的信号是根据 802.11n 协议传输的信号。

(6) 对信号中心频率偏移的问题进行处理，以减少信号的码间干扰。

(7) 对时域形式的信号进行傅里叶变换，将信号转换为频域的形式

(8) 对信号进行子载波分离，得到该信号中所含的数据和用于携带该数据子载波^[44]。

在上述步骤的步骤(8)中，将会得到一个用于携带信息的子载波。测量这些子载波的相位、幅度和频率值，将测量结果进行记录所得到的数据就是 CSI。原始的 CSI 数据不会被无线网卡设备记录和保存，无线网卡设备读取其中的一些信道状态信息后就会将其丢弃。如果要获取 CSI 数据并对其进行处理，则还需要在驱动程序中将此数据取出后再进行操作。

2.3.2 CSI 的数据的收集流程

CSI 数据在硬件上产生后，还需要借助软件工具 CSI-tool 将其从驱动程序中读取出来。Atheros 网卡下的 CSI-tool 会使用通过以下步骤来获取 CSI:

(1) 首先，在 OpenWrt 系统编译开始前，CSI-tool 工具会在 Atheros 的官方驱动程序中源代码中的函数 `ath9k_tasklet` 中，添加一个由自身所实现的函数 `csi_record_status`^[45]。

(2) 当系统启动时，CSI-tool 的相关程序会在内核中注册一个中断类型事件，并在内核空间的内存中创建一个循环队列用于存储数据。

(3) Atheros 网络设备会使用软中断机制来对硬件解析出的数据进行处理。当硬件设备接收到一个外部信号后，会上报一个软中断。`ath9k_tasklet` 负责进行软中断的下半部分的处理。而被添加到 `ath9k_tasklet` 中的 `csi_record_status` 函数，则会将 `ath9k_tasklet` 所接收到的数据进行复制。

(4) `csi_record_status` 函数会对接收到的数据进行过滤和整理操作。对于含有无效内容的数据或者是由于 Atheros 网络设备接收到的 Frame Body 长度为 0 的帧而产生的数据，

csi_record_status 函数会将其滤除^[46]。之后 csi_record_status 函数会通过访问全局变量、调用状态函数、读取结构体数据等方式，来获取诸如信道编号、调制方式、噪声强度、系统时间等后续的 CSI 计算中可能会使用到的数据的状态信息。之后会将这些数据和这些数据的状态信息写入到步骤 2 中的循环队列中^[47]。之后 csi_record_status 会将开启一个用于获取 CSI 数据的中断以激活另一边的对应处理函数。

(5) 另一边，csi_read 会被用于处理 csi_record_status 函数存放于循环缓冲区的数据。csi_read 函数会调用 Linux 内核中的事件等待函数 wait_event_interruptible_timeout 来等待中断产生。在接收到步骤 4 中的中断信号后，csi_read 函数会将这些位于循环队列中的数据取出并将其存放于字符设备的 CSI_dev 中。

(6) 之后，位于用户空间的数据处理程序会使用 open 函数打开文件，从中读取原始的数据流。之后这些数据会被按照 CSI 数据的存放结构进行分割。当完成一次数据分割后，这些带有相应格式的数据内容就能够被各自对应的处理算法处理。

2.3.3 CSI 的数据结构

由于 CSI 数据是多根天线通过互相通信所获得的，所以从混乱地原始数据中解析出每根天线所对应的 CSI 数据也是一道难题。在 Atheros 系列无线网卡中，CSI 数据会以发射端天线编号——接收端天线编号——子载波编号的方式进行存储^[48]。因此在 Atheros 相关设备中的 CSI 读取时，需要按照子载波编号进行 CSI 读取，之后按照接收端天线编号对读取到的一组 CSI 进行排列，最后再按照发射端天线编号对读取到的接收端天线的 CSI 进行排列。算法 2.1 展示了 CSI 数据读取算法的算法流程：

算法 2.1: CSI 数据读取算法	
Input:	CSI 数据队列 data, 发射端天线数 Nc, 接收端天线数 Nr, 子信道个数 Ns
Output:	三维 CSI 数据数组 csi_data
1.	Begin
2.	order<-0
3.	for nc_index=1:Nc do
4.	for nr_index =1:Nr do
5.	for ns_index =1:Ns do
6.	csi[nc_index][nr_index][ns_index]<-data[order]
7.	order<-order+1
8.	end for
9.	end for
10.	end for
11.	End

在进行 CSI 结构解析时，另一个需要解决的问题就是子载波的数目和索引位置的问题。802.11n 协议指出无线信道理论上最多有 64 个子载波，但是实际上由于边界位置、基波位置以及设备自身因素的影响，实际可用的 CSI 信道数会少于 64 个^[49]。并且对于不同类型的无线网卡，其所使用的 CSI 载波的数量、索引位置也不一样。此外，超宽信道的 CSI 载波的数量、索引位置也和常规信道下的 CSI 子载波数量、索引位置不一致。表 2.1 展示了 802.11n 协议所允许的几种 CSI 子载波分布状态。表中的载波编号值表示该载波与信道基波相差的间隔数，正值表示子载波频率高于基波频率，负值表示子载波频率低于基波频率。每个子载波间隔为 312.5KHz^[50]。

表 2.1 CSI 的子信道分布

带宽	编号	子载波个数	子载波编号
20MHZ	1	56	-28, -27, -26 -2, -1, 1 27, 28
	2	30	-28, -26, -24 -2, -1, 1, 3, 5 25, 27, 28
	4	16	-28, -24, -20, -16, -12, -8, -4, -1, 1, 5, 9, 13, 17, 21, 25, 28
40MHz	1	114	-58, -57, -56 -2, -1, 1 57, 58
	2	58	-58, -56, -54 -2, 2, 4 56, 58
	4	30	-58, -54, -50 -2, 2, 6 54, 58

本次设计中所使用 Atheros 无线网卡的 CSI 分布状态为表 2.1 中编号 1 所对应的 CSI 分布状态。

2.4 本章小结

本章介绍了无线定位技术和 CSI 技术的相关背景知识。这章首先介绍了基于角度的无线定位技术和基于距离的无线定位技术。之后，本章引出了本次设计中所采用的混合式无线定位技术。随后介绍了 CSI 的技术原理，介绍了 802.11n 协议下 OFDM 系统中，CSI 信号的生成、处理、发送、接收、解析过程的数学表示，并对 CSI 矩阵的构成，进行了简单的分析。之后对 CSI 信号的在接收端的硬件处理流程进行了介绍，介绍了 CSI 数据是如何从射频信号中被解析出来，以及操作系统是如何驱动硬件收集 CSI 数据的。最后介绍了 CSI 的数据结构，介绍了原始 CSI 数据需要通过何种处理方式才能变成能够被各种算法使用的幅度、相位、频率信息。

第三章 使用 CSI 中相位信息的定位算法

支持 802.11n 协议的无线网络设备,在进行通信时,会记录下用于传输数据的子载波的幅度和相位信息。这些子载波在频域上等间距排列,在时域位置重叠。其在传播过程中受到传播信道的影响,幅度值和相位值会发生变化。因此通过对这些子载波的状态信息进行研究,就能分析出这些传输信道的状态信息。由于 CSI 中相位信息有着与具体设备无关、携带信息量大、程序实现较容易的优点,所以本次设计中将会使用 CSI 中的相位信息设计一款定位算法。

3.1 CSI 相位数据的处理

3.1.1 2π 缠绕处理

在使用 CSI 中的相位信息之前,首先要解决 CSI 中相位数据值不连续的问题,之后才能进行后续的处理操作。从图 3.1 中可以看到,每隔若干个子信道,CSI 的相位值就会发生一个跳变。这是因为在 A/D 转换的过程中,由于相位的 2π 周期性使得 CSI 的相位只能显示在 $0\sim 2\pi$ 之间的投影值。当 CSI 的相位值从小于 2π 的值增加到大于 2π 的值时,新的相位值会被减去 2π 从而到 0 附近。当 CSI 的相位值从大于 0 的值降低到小于 0 的值时,新的相位值会被加上 2π 从而到 2π 附近。因此 CSI 的相位会在 0 和 2π 处形成断点。这种现象被称为 2π 缠绕。

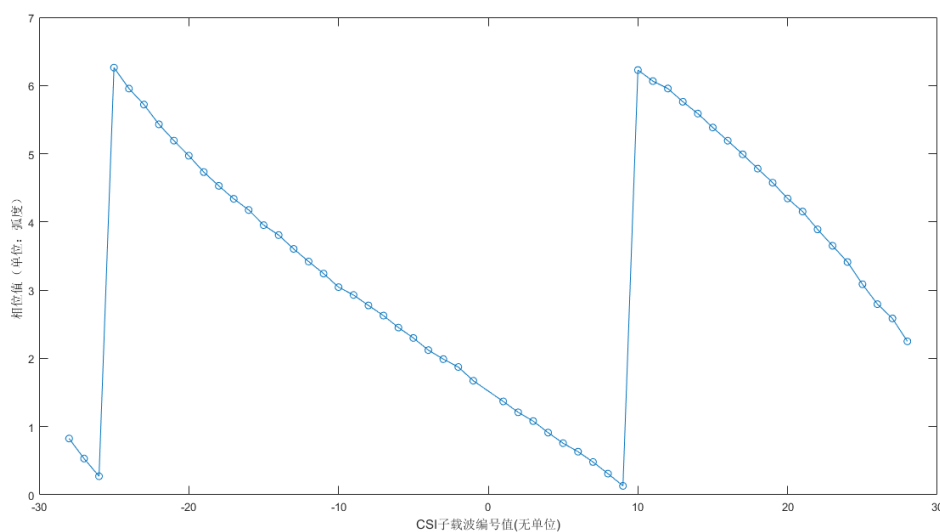


图 3.1 CSI 数据的 2π 缠绕

为了获得连续的相位变化,需要对发生跳变的相位进行 2π 解缠,使得相位值前后保持一致。 2π 解缠的具体操作步骤是:选择一批时域上重叠、频域上互相关联的子载波的 CSI,将

从第二个子载波开始的之后每个子载波的相位与其频域上的前一个子载波的相位值进行做差比较。如果后一个子载波的相位值减去前一个子载波的相位值的结果大于 π ，则需要将后一个子载波的相位减去 2π 。如果后一个子载波的相位值减去前一个子载波的相位值小于 $-\pi$ ，则需要将后一个子载波的相位加上 2π 。如果前后相位差值小于 π ，则认为数据没有发生相位跳变，无需进行相位补偿。

除了 2π 缠绕的问题外，CSI 的 0 号载波问题也需要进行处理。CSI 矩阵在子载波在编号值为 0 的子载波上，并没有 CSI 数据的相位值采样点。这是因为 0 号子载波对应的频率，就是该信道的中心频率值。在硬件中，该频率值会被作为基波用于进行正交信号分离，因而不能用于数据通信和进行 CSI 数据测量。所以此频率下的载波不携带数据，因而也不会被显示出来。虽然硬件上不存在该采样点，但是为了保证 CSI 子载波在频域上的连续性，需要将缺失的载波相位值填补上。在本次设计中，使用三次样条插值法来计算基波的对应相位值。

由于噪声干扰、多径效应、频率偏移的影响，实际情况中的 CSI 信号的子载波的相位值远不如图 3.1 所示的标准。因此需要尽可能地减少这些误差带来的影响。由于这些误差的具有高斯分布的特性，所以使用统计计算的方式和去除极值的方式能有效地降低随机误差的干扰。

3.1.2 基于方差分布的误差过滤算法

在 CSI 相位的实际测量中，由于硬件干扰、噪声误差和多径效应的影响，使得直接获取得到的 CSI 相位信息中包含了大量的误差信息。这些误差会在相位-频率关系图中表现出来：硬件误差会导致信道边界处的相位值发生扭曲；噪声误差会导致所有采样点的相位值有一定程度的上下波动；多径效应会导致特定采样点的相位值发生一个巨大的跳变。为了解决这些干扰因素的影响。本次设计提出了一种基于方差分布的误差过滤算法。

设 CSI 相位的观测矩阵为 H^φ ，则有：

$$H^\varphi = [H_1^\varphi \quad H_2^\varphi \quad \cdots \quad H_{n_p}^\varphi] \quad (3.1)$$

其中， n_p 表示接收到的数据包的个数。在本小节中，假设所有数据均是由收发端的各自编号为 1 的天线进行通信所产生，则每一个数据包的相位观测矩阵 H_i^φ 又可以表示为：

$$H_i^\varphi = [H_{1,i}^\varphi \quad H_{2,i}^\varphi \quad \cdots \quad H_{n_s,i}^\varphi]^T \quad (3.2)$$

其中， n_s 表示单个 WiFi 信道下的 CSI 子载波的个数。根据实验的参数设定，该数值固定

为 56。下面开始介绍基于方差分布的误差处理算法的实现流程。

首先，对相位观测矩阵 H_i^φ 进行线性变换：以子载波索引号为 x 轴坐标，以相位值为 y 轴坐标，对相位-频率关系图中的点进行线性拟合。得到线性变换的结果 (w_i, b_i) 。

$$w_i = \frac{\sum_{j=1}^{n_s} (j * H_{j,i}^\varphi - \frac{1}{2} * n_s^2 * \overline{H_i^\varphi})}{\sum_{j=1}^{n_s} j^2 - \frac{1}{2} * n_s^3} \quad (3.3)$$

$$b_i = \overline{H_i^\varphi} - \frac{1}{2} * n_s * \widehat{w_i} \quad (3.4)$$

其中 $\overline{H_i^\varphi}$ 表示数据包的相位观测矩阵 H_i^φ 的平均值，其计算方式如下：

$$\overline{H_i^\varphi} = \frac{1}{n_s} * \sum_{j=1}^{n_s} H_{j,i}^\varphi \quad (3.5)$$

之后，计算线性变换的优度 R ：

$$R = \frac{\sum_{j=1}^{n_s} (j * w_i + b_i - \overline{H_i^\varphi})^2}{\sum_{j=1}^{n_s} (H_{j,i}^\varphi - \overline{H_i^\varphi})^2} \quad (3.6)$$

优度 R 用于表示该线性变换后的 (w_i, b_i) 的数据图与原本的相位观测矩阵 H_i^φ 的数据图的关联的紧密程度。优度 R 越高，表示接收到的 CSI 数据中的误差值越小。为了判断某条 CSI 数据是否因为误差干扰而需要进行误差消除操作，本文在此处会给出一个判断阈值 t ，如果 R 未达到阈值 t ，则还需要进行接下来的公式 3.7-公式 3.8 的操作后，再次进行公式 3.3-公式 3.6 的计算操作，循环进行这些处理操作直至 R 值满足阈值 t 。

然后分别计算出相位观测矩阵 H_i^φ 中每一个具体的相位值 $H_{j,i}^\varphi$ 在相频图中所对应的采样点与线性变换的结果 $(\widehat{w_i}, \widehat{b_i})$ 的欧式距离。计算方式如下：

$$d_j = \frac{|j * \widehat{w_i} - H_{j,i}^\varphi + \widehat{b_i}|}{(\widehat{w_i}^2 + \widehat{b_i}^2)^{\frac{1}{2}}} \quad (3.7)$$

接下来对有着最远欧式距离的 $H_{j,i}^\varphi$ ，将其从相位观测矩阵 H_i^φ 中暂时移除。此时，将 H_i^φ 更新为：

$$H_i^\varphi = [H_{1,i}^\varphi \quad \cdots \quad H_{j-1,i}^\varphi \quad H_{j+1,i}^\varphi \quad \cdots \quad H_{n_s,i}^\varphi]^T \quad (j = \max_{j=1, n_s} d_j) \quad (3.8)$$

图 3.2 对比展示了进行误差过滤前后的相位值分布图以及其线性变换的结果。可以看到在相频图中，靠近信道边界的硬件偏移误差、信道内部噪声干扰误差均被较好的移除了。移除的这些子载波中，大部分为靠近信道边界的子载波。这些子载波因为硬件误差而产生了规律性的相位偏移。这些规律性的相位偏移会对定位结果产生一个规律性的位置偏移，从而影

响后续的运动计算。

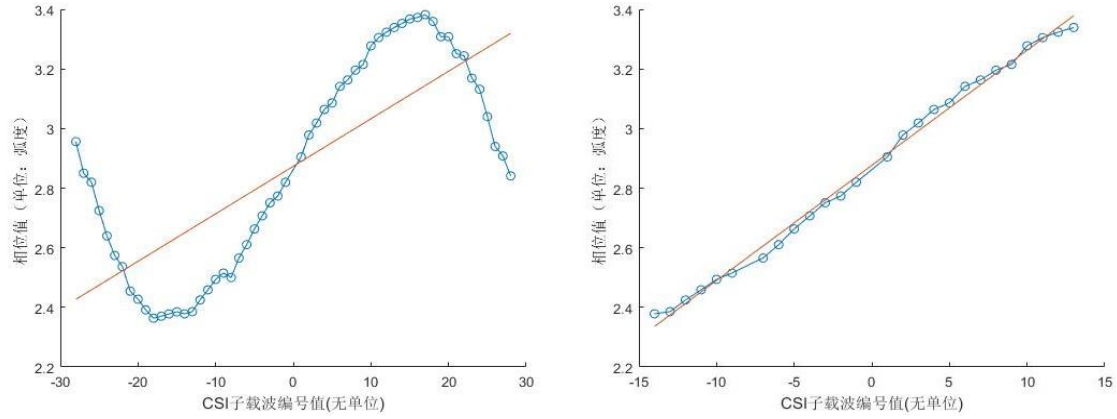


图 3.2 误差过滤对比（左图：未进行过滤处理；右图：进行了过滤处理）

下一步需要对多个数据包进行处理。在计算出全部 n_p 个数据包所对应的 (w_i, b_i) 后，对所有数据包的线性变换结果进行分布的方差分析，以得出合适的线性变换阈值来进行误差过滤。线性变换阈值 t 设置的过高或过低都会对后续的计算产生不良影响。如果线性变换阈值 t 设置的过低，会导致一部分干扰误差没有被消除，从而对后续计算的结果缺乏可信度。如果线性变换阈值 t 设置的过高，会导致一部分正常的 CSI 数据也会被移除，从而导致最后的计算结果与实际值有一定的偏差。因此，本次设计将会使用一种计算方差分布的方式来选择最佳的线性变换阈值。

首先，构建矩阵 **A**、矩阵 **B**、矩阵 **Y**：

$$A = \begin{bmatrix} 1 & 1 & 1 \\ \vdots & \vdots & \vdots \\ 1 & i & i^2 \\ \vdots & \vdots & \vdots \\ 1 & n_p & n_p^2 \end{bmatrix}, i = 1 \cdots n_p \quad (3.9)$$

$$B = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad (3.10)$$

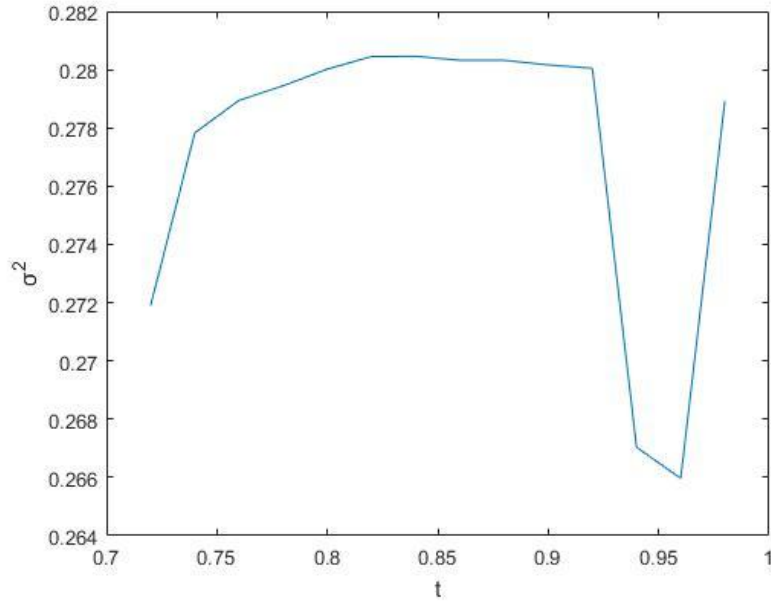
$$Y = \begin{bmatrix} \ln(w_1) \\ \vdots \\ \ln(w_i) \\ \vdots \\ \ln(w_{n_p}) \end{bmatrix}, i = 1 \cdots n_p \quad (3.11)$$

其中， b_1 、 b_2 、 b_3 为未知量。

之后进行如下计算：

$$B = (A^T A)^{-1} A^T Z \quad (3.12)$$

$$\sigma^2 = -\frac{1}{b_3} \quad (3.13)$$

图 3.3 t - σ^2 分布关系图

此时，将会得优度阈值为 t 时的方差值 σ^2 。对于不同的优度阈值， σ^2 也会有不同的数值。因此，计算不同 t 值对应的方差值 σ^2 就能够获得一个 t - σ^2 的分布关系图。图 3.3 展示了本次设计中的 t - σ^2 的分布关系图。从图 3.3 中可以看到，当 t 值在 0.96 附近时，方差值将会达到最小值 0.266。即当优度阈值设置为 0.96 时，过滤效果最好。

最后，根据阈值-方差分布关系图，得出本组数据包的最优过滤阈值参数。将得到的最优参数回代至原本的误差过滤算法中，重新进行公式 3.3-公式 3.5 的操作，得出每一组相位序列的线性变换的结果 (w_i, b_i) 。本次设计中将会使用线性变换后各个子载波的相位值代替原本观测到的子载波相位值作为相位观测矩阵 \mathbf{H}' 。替换公式如下：

$$\mathbf{H}'_i = w_i * j + b_i \quad (3.14)$$

在本次设计的接下来的操作中，将会使用 \mathbf{H}' ，进行定位值的计算。

在本节中，我们假设电磁波的发射端和接收端为均只有一根天线的场景来简化计算。但是在实际应用场景中，则还需要考虑天线数量的问题的。假设发送端有 n_t 根天线，接受端有 n_r 根天线，那么就会产生 $n_t \times n_r$ 种天线组合。由于每一个天线组合所生成的观测矩阵的大小都与 \mathbf{H} 大小相同，所以在实际场景中需要将公式 3.14 重复执行 $n_t \times n_r$ 次。因此，对公式 3.14 进行如下更新：

$$\mathbf{H}'_{i,n_t,n_r} = w_{i,n_t,n_r} * j + b_{i,n_t,n_r} \quad (3.15)$$

为了确保结果的精确度，每一批数据都需要使用单独的参数进行误差过滤。但是由于该算法计算耗时久，并且在实际场景中无线信道的状态基本固定，所以本次设计中所有数据的处理参数都会沿用首批数据的处理参数。算法 3.1 展示了本节的算法的全部流程：

算法 3.1: 基于方差分布的误差过滤算法

Input: 大小为 $1 * n_p * n_s$ 的 CSI 相位观测矩阵 \mathbf{H}

Output: 经过误差处理的相位观测矩阵 \mathbf{H}'

```

1. Begin
2. initial  $\mathbf{H}' < -0$ 
3. for t from 0 to 0.98 step 0.02 do
4.     for i in  $n_p$  do
5.         for j in  $n_s$  do
6.             计算线性变换的结果  $w_{i,j}$  和  $b_{i,j}$ 
7.             计算本次线性变换的优度 R
8.         end for
9.         while ( $R \leq t$ ) do
10.            计算全部  $n_s$  个点的  $d_j$ 
11.            找到  $d_j$  最大的点的对应的 j 值
12.            从观测矩阵  $\mathbf{H}_i$  中暂时移除 j 所对应的采样点
13.            使用新的观测矩阵计算线性变换的结果  $w_{i,j}$  和  $b_{i,j}$ 
14.            计算本次线性变换的优度 R
15.        end for
16.    end for
17.    构建矩阵 A, 向量 Z
18.    进行矩阵计算  $\mathbf{B} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{Z}$ 
19.    根据向量 B, 计算出方差值  $\sigma^2$ 
20.    记录下当前 t 值和对应的方差值  $\sigma^2$ 
21. end for
22. 根据记录下的值, 绘制  $t-\sigma^2$  关系图
23. 找到关系图中  $\sigma^2$  到达峰值时对应的 t 值
24. for i in  $n_p$  do
25.     for j in  $n_s$  do
26.         计算线性变换的结果  $w_{i,j}$  和  $b_{i,j}$ 
27.         计算本次线性变换的优度 R
28.     end for
29.     while ( $R \leq t$ ) do
30.        计算全部  $n_s$  个点的  $d_j$ 
31.        找到  $d_j$  最大的点的对应的 j 值
32.        从观测矩阵  $\mathbf{H}_i$  中暂时移除 j 所对应的采样点
33.        使用新的观测矩阵计算线性变换的结果  $w_{i,j}$  和  $b_{i,j}$ 
34.        计算本次线性变换的优度 R
35.    end while
36.    for j in  $n_s$  do
37.        对  $\mathbf{H}'$  中的值  $\mathbf{H}'_{i,j}$  进行更新:  $\mathbf{H}'_{i,j} < -w_{i,j} * j + b_{i,j}$ 
38.    end for
39. end for
40. End

```

3.2 基于差分计算 CSI 相位的定位算法

传统的 CSI 相位处理算法尽管可以很好计算出 CSI 中相位值的发射端原始值，但是由于这些算法有着硬件要求高、处理速度慢等缺点导致无法被应用于实际的场景中。并且考虑到使用 CSI 中的相位数据来实现定位不需要具体相位值，因此本次设计将会采用一种另一种算法来代替传统的 CSI 相位处理算法进行位置计算。

3.2.1 距离计算算法

CSI 中的各个子载波在从发射端天线发出时的起始相位均是相同的，之所以会出现接收端接收各个子载波的相位不一致情形，是因为其在传播过程中由于各自频率不一致而导致与信道频率响应进行计算后得到了不同的结果，从而产生了不同的相位的变化。因此在拥有距离因素对不同频率的子载波的相位值的产生影响的具体计算表达式之后，就可以根据 CSI 中相位信息分析环境中一些距离数值，其中就包括被测目标与无线信号接收端的距离信息。由于电磁波信号的原始相位值会在 2π 解缠中丢失，因此本次设计中会使用一种差分计算的方式来进行相位的相关计算从而测量收发双方的距离。

假设发射端天线与接收端天线的距离为 d ，所使用的信道的中心频率为 f_c ，此电磁波信号从发射端发射出时的初始相位为 P ，CSI 子载波之间的频率间隔为 Δf ，电磁波在空气中的传播速度为 c ，则电磁波从发射端传输到接收端的耗时 t 为：

$$t = \frac{d}{c} \quad (3.16)$$

对于编号为 i 的子载波而言，其频率 f_i 分别为：

$$f_i = f_c + i * \Delta f \quad (3.17)$$

经过时间 t 后，编号为 i 的子载波的相位值 P_i 为：

$$P_i = 2 * \pi * f_i * t + P \quad (3.18)$$

在 3.1 节的数据预处理中，所有子载波的相位值都会根据编号值最小的子载波的相位值进行调整。所以相位值比较的参考对象只能是编号值最小的子载波。设编号值最小的子载波编号值为 \min ，对应的频率值为 f_{\min} 、相位值为 P_{\min} 则：

$$f_{\min} = f_c + \min * \Delta f \quad (3.19)$$

$$P_{\min} = 2 * \pi * f_{\min} * t + P \quad (3.20)$$

由于在 2π 解缠过程中，子载波 i 的相位值会根据编号为 \min 的子载波的相位值进行 $\pm 2k\pi$ 的调整。所以需要计算编号为 i 的子载波的相位值与编号为 \min 的子载波的相位值的差值：

$$P_{i,min} = P_i - P_{min} \quad (3.21)$$

将公式 3.18、公式 3.19、公式 3.20 依次带入公式 3.21，得到

$$P_{i,min} = 2 * \pi * \frac{d}{c} * \Delta f * (i - min) \quad (3.22)$$

上述公式中，编号 i 的子载波和编号 min 的子载波的相位值已知。 Δf 为子载波之间的频率间隔值，在 802.11n 协议中该值被固定为 312.5KHz。因此只需要知道编号值最小的子载波相位值和任意一个其他子载波的相位值，就可以通过计算得出电磁波接收端和发送端之间的距离。将上面公式进行推广到子载波编号分别为 i 和 j 的场景，再令 $j=i+1$ 可以得到：

$$P_{i,j} = 2 * \pi * \frac{d}{c} * \Delta f * (i - j) \quad (3.23)$$

$$P_{i,i+1} = 2 * \pi * \frac{d}{c} * \Delta f \quad (3.24)$$

因而，只需要知道任意两个相邻的子载波的相位差就可以计算出电磁波接收端和发送端之间的距离。考虑到基波附近的子载波受到中心频偏引发的误差最小，我们选择计算编号为 -1 和编号为 1 的两个子载波的相位差，来带入公式 3.24 并获得距离距离。但是由于噪声以及其他干扰的影响，只选用一组数据的计算结果明显较差。

对于这些相位点的计算，除了可以从统计学角度进行累计计算后统计结果外，还可以从解析几何角度进行数据计算。对于两个相邻的载波，其相位差值还可以被视为经过这两个点的直线的斜率值。对于多个相位点平均相位差的计算，则可以视为对这些相位点进行一个一维多项式拟合，得出的线性函数的 x 方向的系数值就是这些相位点的平均相位差。

算法 3.2: 基于差分计算 CSI 相位的距离计算算法

Input: n_p 个大小为 $n_r * n_c * n_s$ 的 CSI 相位观测矩阵

Output: 距离值 d

1. **Begin**
 2. **for** $i=1:n_p$ **do**
 3. **for** $j=1:n_r$ **do**
 4. **for** $k=1:n_c$ **do**
 5. 对相位-频率图中的采样点进行一维多项式拟合，得出拟合函数斜率 $w_{i,j,k}$
 6. **end for**
 7. **end for**
 8. **end for**
 9. 找出所有的 $w_{i,j,k}$ 中的最小的数值 w_{min}
 10. 找出所有的 $w_{i,j,k}$ 中的最大的数值 w_{max}
 11. 将 w_{min} 到 w_{max} 之间的值，等分为 20 个区间，统计 $w_{i,j,k}$ 落在各个区间上的数据个数
 12. 根据统计出的数据个数，绘制柱状图
 13. 对柱状图做一维高斯分布的拟合，得出拟合函数 $y = ae^{-(x-b)^2/2c^2}$ 中的参数 b 的值。
 14. 计算距离值: $d = bc/2\pi\Delta f$
 15. **End**
-

为了抑制距离计算中误差干扰,本次设计还会使用高斯拟合的方式来进行计算。算法 3.2 展示了基于差分计算 CSI 相位的距离计算算法的具体操作流程。

3.2.2 到达角计算算法

借助无线路由器的天线阵列能够计算所接收电磁波信号的 AoA。不同的算法中,使用天线阵列测量 AoA 的方式也不一样。

由于在大部分实验场景中,电磁波的发射端和接收端的距离是接收端的天线阵列间距的 10 倍以上。因此,发射端向外发送的电磁波对于接收端而言可以近似认为是平行的。图 3.4 表示了其基本的工作场景:

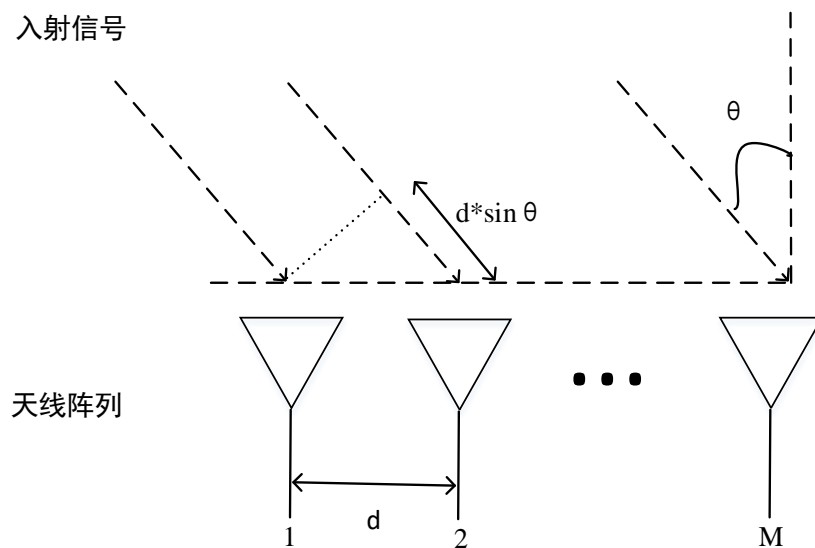


图 3.4 到达角测量原理

当接收机的天线阵列接收到了电磁波后,由于存在入射角的原因,天线 2 接收到的电磁波入射信号的传播距离比天线 1 接收到的电磁波入射信号的传播距离多一个 $d * \sin \theta$ 的距离。因此,分别计算出两根天线所接收到的电磁波的传输距离,进行差值计算得到 $d * \sin \theta$ 的数值,是这种算法的基本思想。

但是由于测距算法的测量误差值,要远大于接收端天线的距离值,因此无法使用直接测量电磁波传播距离的方式测量 $d * \sin \theta$ 的值。在 3.2.1 节中,我们使用了一种相位差的计算方法,来计算电磁波,从发射端到接收端所经过的距离。由于 AoA 算法实质上是距离 $d * \sin \theta$ 进行计算,因此在对相位进行同步后,也可以使用距离计算的方法来进行角度计算。

设电磁波从发射端到接收端天线 1 到的传播距离为 l_1 , 到天线 2 的传播距离为 l_2 。则其对应的相位差 P_1 和 P_2 分别为:

$$P_1 = 2 * \pi * \frac{l_1}{c} * f_c \quad (3.25)$$

$$P_2 = 2 * \pi * \frac{l_2}{c} * f_c \quad (3.26)$$

因而可得:

$$P_2 - P_1 = \frac{2 * \pi * f_c}{c} * d * \sin \theta \quad (3.27)$$

在此公式中, 相位值 P_1 和 P_2 就是不同天线的相位观测值。 d 为天线阵列的距离值, 为固定值。因此根据公式 3.27 将能够计算出 AoA 的弧度值 θ 。为了减少误差值的干扰, 在获取到相位数据后不立刻进行计算而是重复进行数据采集, 之后使用高斯拟合算法计算出最接近的距离值。算法 3.3 展示了基于差分计算 CSI 相位的到达角计算算法的流程:

算法 3.3 差分计算 CSI 相位的到达角计算算法

Input: n_p 个大小为 $n_r * n_c * n_s$ 的 CSI 相位观测矩阵 $\mathbf{H}(n_r \geq 2)$, 接收端天线阵列间距 d , 信道中心频率 f_c

Output: 到达角 θ

1. **Begin**
 2. **for** $i=1:n_p$ **do**
 3. **for** $j=1:n_c$ **do**
 4. **for** $k=1:n_s$ **do**
 5. 计算不同接收端天线的相位差值: $\Delta P_{i,j,k} = H_{i,2,j,k} - H_{i,1,j,k}$
 6. **end for**
 7. **end for**
 8. **end for**
 9. 对所有的 $\Delta P_{i,j,k}$ 进行加减 $k\pi$ 的操作, 使得所有的 $\Delta P_{i,j,k}$ 数值均落在 $[-\pi, \pi]$ 区间中
 10. 找出所有的 $\Delta P_{i,j,k}$ 的最小的数值 ΔP_{min}
 11. 找出所有的 $\Delta P_{i,j,k}$ 中的最大的数值 ΔP_{max}
 12. 将 ΔP_{min} 到 ΔP_{max} 之间的值, 等分为 20 个区间, 统计 $\Delta P_{i,j,k}$ 落在各个区间上的数据个数
 13. 根据统计出的数据个数, 绘制柱状图
 14. 对柱状图做一维高斯分布的拟合, 得出拟合函数 $y = ae^{-(x-b)^2/2c^2}$ 中的参数 b 的值。
 15. 计算到达角: $\theta = \arcsin(bc/2\pi f_c d)$
 16. **End**
-

3.3 仿真结果

为了确认算法的有效性, 本次设计收集了 500 条 CSI 数据并使用 Matlab 进行算法测试。首先测试的是基于方差分布的误差过滤算法。图 3.5 展示了数据的误差过滤前后, 数据分布情况的对比。从图 3.5 中可以看到, 经过误差处理算法进行处理后, 这些数据均方差值从 0.374 降低到了 0.266, 均方差值降低了 28%。

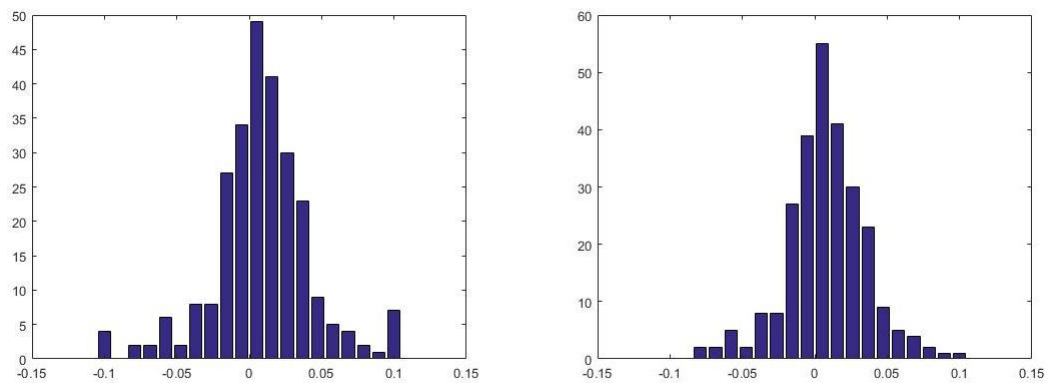


图 3.5 误差过滤算法表现（左图：原始数据；右图:过滤之后的数据）

之后进行定位算法的测试。用于仿真计算的 CSI 数据在采集时被测目标与接收端之间的距离为 3.73m，被测目标相对与到接收端的到达角为 0.269rad。图 3.6 展示了距离计算算法的高斯拟合结果，图 3.7 展示了到达角计算算法的高斯拟合结果。

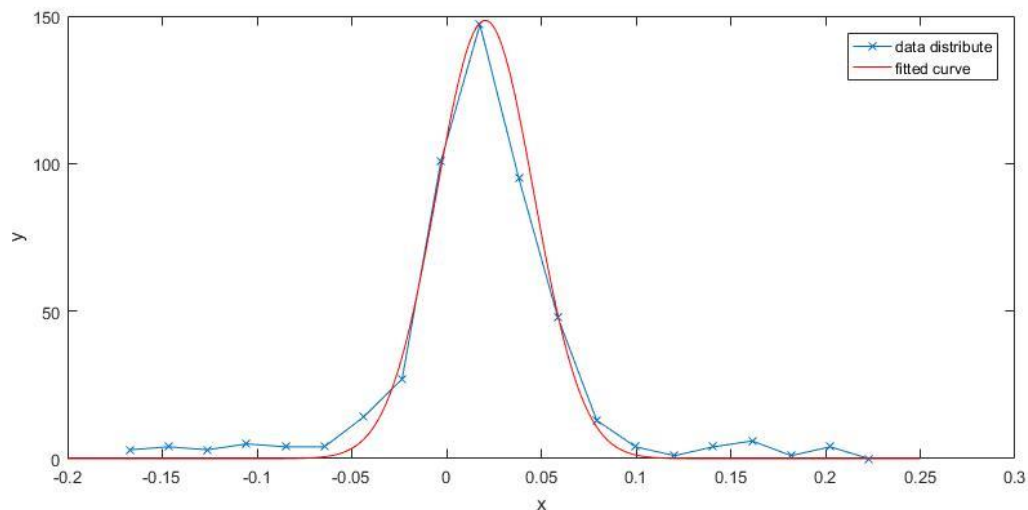


图 3.6 距离计算结果

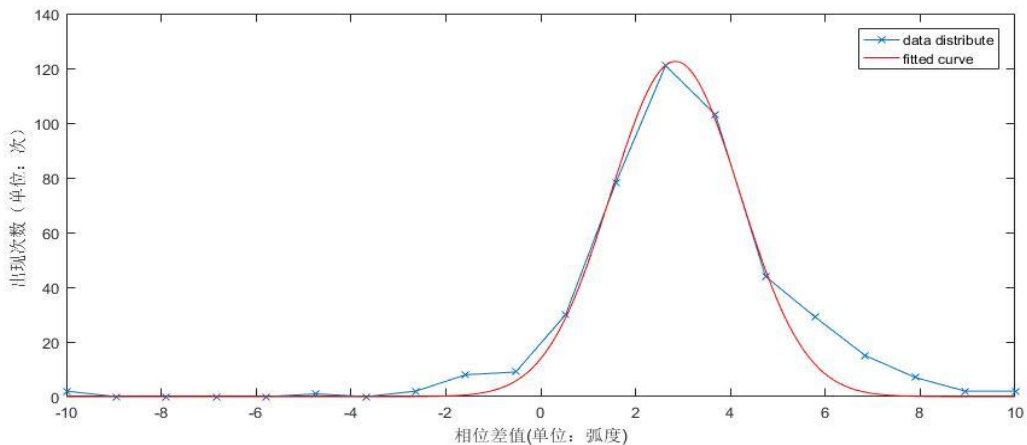


图 3.7 到达角高斯拟合结果图

图 3.6 所示的距离计算算法的高斯拟合中心值为 0.0226，按照相应公式转换，可以算出

距离值为 3.45m，误差为 8%。图 3.7 所示的到达角计算算法的高斯拟合中心值为 2.83。依据数据采集场景下 $d=12\text{cm}$ 、 $f_c=5.18\text{GHz}$ 的系统状态参数，可以算出到达角为 0.232rad，误差值为 13%。

3.4 本章小结

在这一章中，首先介绍了 CSI 相位在使用前的预处理。CSI 相位的预处理，要解决两个问题。第一个问题时 CSI 相位会因为 2π 缠绕而产生 2π 跳变。该问题的解决方式是对所有相位与前一相位依次进行相位比较。如果差值大于 π 则认为发生了跳变，需要进行相应处理。之后介绍了一种基于方差分布的误差过滤算法。该算法的核心思想是对所有 CSI 数据进行一个线性变换操作，然后判断线性变换的优度值是否满足阈值，如果不满足则会对该组数据进行一次最差值过滤，则再次进行线性变换并进行优度值的检查，重复循环此操作直到优度值满足阈值。之后对线性变换的结果进行一个评估：计算线性变换结果的均方差值，记录下此线性变换结果的均方差值以及和其对应的优度阈值。计算不同阈值下的均方差值，根据优度阈值-均方差值分布关系图，选择最小的均方差值对应的优度阈值作为最佳优度阈值。最后使用此最佳阈值，更新 CSI 观测相位矩阵。

本章还介绍了本次设计中使用的基于差分计算 CSI 相位的定位算法。该算法会在两个维度上分别使用差分计算的方式，来计算出被测目标的距离信息和方向角信息。通过计算同一时刻、同一收发端天线、不同频率下的子载波的相位差来获得接收信号的 ToF 测量结果。之后根据此测量结果来计算无线信号接收端和发送端的距离。通过计算同一时刻、同一频率，同一发送端天线、不同接收端天线下的子载波的相位差来获得接收信号的 AoA 测量结果。该测量结果将被用来计算无线信号发送端相对于接收端的到达角度。

最后，本章给出了 CSI 定位算法的中测角算法和测距算法的仿真结果。仿真结果表明，经过误差过滤算法处理的仿真数据的分布方差值，相较于未经处理的数据的分布方差值，降低了 28%。ToF 测量算法的仿真数据计算误差为 8%，AoA 测量算法的仿真数据计算误差为 13%。

第四章 基于卡尔曼滤波器的轨迹追踪算法

卡尔曼滤波是一种优秀的对线性高斯系统状态进行最优估计的算法，因而被广泛地用于交通、控制、航海、图像处理等领域。它能够很好的利用线性系统状态方程，通过输入观测数据来实现对线性高斯系统的系统状态进行最优值预测。借助卡尔曼滤波器可以对线性高斯系统的过去、现在、将来的状态进行很好的估计。传统卡尔曼滤波器虽然表现优秀，但是在本次设计中的应用场景下会出现一些问题，因此需要对其进行一定的改良。

4.1 传统卡尔曼滤波器

4.1.1 卡尔曼滤波器的原理

在状态系统中记 x_k 为系统在时刻 k 的所处状态，记 u_k 为系统在时刻 k 的外部控制向量，记 m_k 为系统在时刻 k 的输入噪声。则有：

$$x_k = f(x_{k-1}, u_k, m_k) \quad (4.1)$$

公式 4.1 中, f 为运动方程,用于描述系统如何从 $k-1$ 时刻的状态 x_{k-1} 变成 k 时刻的状态 x_k 。这个公式表达的是系统的真实状态的转移。但是系统的真实状态是无法获知的，所以实际中使用系统的观测状态来代替实际状态^[51]。

记 y_k 为系统在时刻 k 的时被观测获得的状态，记 n_k 为在时刻 k 进行状态观测时的观测噪声，则有：

$$y_k = g(x_k, n_k) \quad (4.2)$$

公式 4.2 中的 g 表示系统的观测方程。

在线性高斯系统中，运动方程 f 和观测方程 g 都是线性的，并且两个噪声值 m_k 和 n_k 也均是服从零均值的高斯分布。因此在线性高斯系统下，公式 4.1 和公式 4.2 可以分别转化为如下的形式^[52]。

$$x_k = A_k x_{k-1} + B_k u_k + m_k \quad (4.3)$$

$$y_k = C_k x_k + n_k \quad (4.4)$$

其中：

$$m_k \sim N(0, Q_k) \quad (4.5)$$

$$n_k \sim N(0, R_k) \quad (4.6)$$

其中, A_k 为线性高斯系统在 k 时刻的状态转移矩阵, B_k 为系统在 k 时刻的控制矩阵, C_k 为系统的观测矩阵, Q_k 和 R_k 为系统的噪声方差。在线性高斯系统中, 函数 $f(\cdot)$ 变成了 A_k 和 B_k , 函数 $g(\cdot)$ 变成了 C_k 。

在线性高斯系统下使用贝叶斯定理来计算 x_k 的后验概率就是卡尔曼滤波器的核心思想。卡尔曼滤波器通过线性系统的递推形式来得到线性系统的无偏最优估计。卡尔曼滤波器的过程推导这里不做描述, 详细内容见参考文献[53], 这里直接给出卡尔曼滤波器的公式。

当 $k=t$ 时, 卡尔曼滤波的五个公式如下:

$$x_t^- = A_t x_{t-1} + B_t u_t \quad (4.7)$$

$$c_t^- = A_t c_{t-1} A_t^T + Q_t \quad (4.8)$$

$$K_t = c_t^- A_t^T / (A_t c_t^- A_t^T + R_t) \quad (4.9)$$

$$x_t = x_t^- + K_t (y_t - A_t x_t^-) \quad (4.10)$$

$$c_t = (I - K_t A_t) c_t^- \quad (4.11)$$

在这些公式中, 公式 4.7 和公式 4.8 称为估计公式, 用于根据现有的状态, 对下一时刻的状态进行估计。公式 4.9、公式 4.10、公式 4.11 称为更新公式, 用于根据系统协方差状态对估计值进行更新^[54]。

4.1.2 卡尔曼滤波器的使用

卡尔曼滤波器的工作步骤如下:

1. 根据线性高斯系统的 $t-1$ 时刻的状态的观测值 x_{t-1} 和控制向量 u_t , 计算出在 t 时刻的系统状态估计值 x_t^- 。 x_t^- 只是系统基于上一状态计算而来的估计值, 并不是最优估计值。公式 4.12 是对公式 4.1 在线性高斯系统下的特殊表现形式。

$$x_t^- = A_t x_{t-1} + B_t u_t \quad (4.12)$$

2. 根据系统的 $t-1$ 时刻的噪声协方差矩阵的最优估计值对 t 时刻的噪声协方差矩阵进行估计。 c_t^- 表示系统在 t 时刻的噪声的协方差矩阵的估计值。 c_{t-1} 表示系统在 $t-1$ 时刻的噪声的协方差矩阵的实际值。 Q_t 为 t 时刻的系统的噪声协方差矩阵。

$$c_t^- = A_t c_{t-1} A_t^T + Q_t \quad (4.13)$$

3. 计算系统的卡尔曼增益。 K_t 表示系统在 t 时刻的卡尔曼增益值。 A_t^T 为系统在 t 时刻的转移矩阵的转置矩阵。 R_t 为系统在 t 时刻的观测噪声。

$$K_t = c_t^- A_t^T / (A_t c_t^- A_t^T + R_t) \quad (4.14)$$

4. 根据系统的观测值和估计值，计算此时系统的最优状态估计值。方程的计算结果 x_t 就是系统在 t 时刻的状态值的估计值。

$$x_t = x_t^- + K_t(y_t - A_t x_t^-) \quad (4.15)$$

5. 更新系统的噪声协方差矩阵。 \hat{p}_t 为 t 时刻的噪声协方差矩阵的最优估计值。 I 是单位矩阵。

$$c_t = (I - K_t A_t) c_t^- \quad (4.16)$$

上述步骤为卡尔曼滤波器的标准用法。通过不断迭代和重复，系统的最优估计值 x_t 会逐渐向实际值靠近^[55]。通过最优估计值和转移矩阵所得出的估计值 x_t^- 也会逐渐的向最优估计值 x_t 靠近。同时，卡尔曼增益 K_t 也会逐渐的向零矩阵靠近^[56]。

4.2 卡尔曼滤波器的改进

由于 WiFi 定位场景下被测物体的加速度未知，所以传统卡尔曼滤波器中控制变量 u_t 是无法得出。因此本次设计中提出了一种借助线性运动物体的“惯性”进行状态预测的算法。但是使用这种算法会导致常规场景下的计算结果误差增加。因而本次设计又提出了一种状态值量化的算法来进行常规场景下的计算误差抑制。

4.2.1 惯性状态算法

由于传统卡尔曼滤波器中的控制变量 u_t 本次实验中为不可知参数，因此本次实验中将会根据线性运动下的物体的运动状态不会发生较大改变的特点，对 k 时刻的系统观测值和 $k-1$ 时刻的观测值进行线性处理，并将处理后的结果作为系统状态受外部变量影响的结果。

设 WiFi 定位的目标在 k 时刻下的系统状态如下：

$$x_k = \begin{bmatrix} p_{x,k} \\ p_{y,k} \\ v_{x,k} \\ v_{y,k} \end{bmatrix} \quad (4.17)$$

其中， $p_{x,k}$ 表示系统在 k 时刻、 x 方向上的坐标状态。 $p_{y,k}$ 表示系统在 k 时刻、 y 方向上的坐标状态。 $v_{x,k}$ 表示系统在 k 时刻、 x 方向上的速度状态。 $v_{y,k}$ 表示系统在 k 时刻、 y 方向上的速度状态。

由于外部控制变量 u_k 不可知，因此无法将其带入计算。不妨假设：

$$u_k = \vec{0} \quad (4.18)$$

则公式 4.1 可以简化为如下形式:

$$x_k = f(x_{k-1}, m_k) \quad (4.19)$$

考虑到在实际的 WiFi 定位场景中, 系统状态的突变, 只是系统的速度状态的突变, 而位置状态依旧连续。因此, 对系统状态 x_k 进行如下操作:

将系统状态 x_k 分隔为位置状态 p_k 和速度状态 v_k :

$$x_k = \begin{bmatrix} p_k \\ v_k \end{bmatrix} \quad (4.20)$$

$$p_k = \begin{bmatrix} p_{x,k} \\ p_{y,k} \end{bmatrix} \quad (4.21)$$

$$v_k = \begin{bmatrix} v_{x,k} \\ v_{y,k} \end{bmatrix} \quad (4.22)$$

对于位置状态 p_k , 则有:

$$p_k = f_p(p_{k-1}, m_{p,k}) \quad (4.23)$$

其中 $m_{p,k}$ 表示位置状态的输入噪声。在惯性状态算法下, 设被定位目标的位置状态仍然与前一时刻的系统状态呈线性关系, 因此公式 4.3 可以表示为:

$$p_k = A_{p,k} x_{k-1} + m_{p,k} \quad (4.24)$$

其中, $A_{p,k}$ 为位置状态的转移方程。由于 WiFi 定位的局限性, 被测目标的速度状态 v_k 的外部控制变量 $u_{v,k}$ 无法被测量出来。在惯性状态算法下, 设被定位目标在 k 时刻的速度状态 v_k 为最近两个时刻观测值 y_k 和 y_{k-1} 的线性组合:

$$v_k = f_v(y_k, y_{k-1}) \quad (4.25)$$

此时 f_v 则可以视为一种线性的方程, 则有:

$$v_k = A_{v,y_k} y_k + A_{v,y_{k-1}} y_{k-1} + m_{p,k} \quad (4.26)$$

虽然此方程解决了 v_k 的迭代问题, 但是同时也将观测误差 n_k 和 n_{k-1} 引入到了系统的转移状态中。因此需要对公式 4.8 进行更新。由于在本次设计的场景中, 速度状态和位置状态的噪声互不影响。因此可以将速度状态的协方差和位置状态的协方差矩阵进行拆分:

$$c_k = \begin{bmatrix} c_{p,k} & 0 \\ 0 & c_{v,k} \end{bmatrix} \quad (4.27)$$

其中, $c_{p,k}$ 表示位置状态的协方差矩阵, $c_{v,k}$ 表示速度状态的协方差矩阵。经过此拆分, 则可以将式 4.26 中引入的观测误差进行计算。计算表达式如下:

$$c_{pk}^- = A_{p,k} c_{t-1} A_k^T + Q_{p,k} \quad (4.28)$$

$$c_{vk}^- = A_{v,y_k} n_k A_{v,y_k}^T - A_{v,y_{k-1}} n_{k-1} A_{v,y_{k-1}}^T + Q_{v,k} \quad (4.29)$$

其中, $Q_{p,k}$ 和 $Q_{v,k}$ 分别为系统在 k 时刻的位置状态均方差和速度状态的均方差。在计算出 $c_{p,k}^-$ 和 $c_{v,k}^-$ 之后, 将其按照公式 4.27 进行组合成为 c_k 。之后的按照传统卡尔曼算法的计算步骤进行计算, 得出系统状态预测值。

4.2.2 状态量化算法

惯性状态算法使用 k 到 $k-1$ 的系统状态观测值的线性组合来对系统的速度状态 v_k 进行更新。此方法虽然能实现对系统状态突变的快速反应, 但是也会因为破坏了整体的迭代连续性而导致此算法对标准线性系统的计算结果的误差值增加。为了解决此问题, 本次设计中会对惯性状态算法进行量化处理: 从多个预先给定好的状态值中, 选择最接近实际值的结果, 来代替直接的计算结果。

在 WiFi 定位场景下的物体移动系统中, 系统从 $k-1$ 到 k 时刻的状态转移结果可以划分为四个状态, 分别代表被测物体前进、后退和左右转。用 $x_{k|k-1}^i, i = 1, 2, 3, 4$ 表示系统从 $k-1$ 时刻转移到 k 时刻的可能状态位置。则有:

$$x_k \in \{x_k^i | i = 1, 2, 3, 4\} \quad (4.30)$$

对惯性状态算法下的卡尔曼滤波器的处理公式进行如下更新:

$$x_k^i = \begin{bmatrix} p_k^i \\ v_k^i \end{bmatrix} \quad (4.31)$$

$$p_k^i = A_{p,k} p_{k-1} + B_{p,k} u_{p,k}^i \quad (4.32)$$

$$v_k^i = A_{v,y_k} y_k + A_{v,y_{k-1}} y_{k-1} + B_{v,k} u_{v,k}^i + m_{p,k} \quad (4.33)$$

$u_{p,k}^i$ 和 $u_{v,k}^i$ 为自定义的控制矩阵, 用于计算系统状态转移的多个可能量化结果。不妨设 $B_{p,k}$ 和 $B_{v,k}$ 均为单位矩阵, 则有

$$p_k^i = A_{p,k} p_{k-1} + u_{p,k}^i \quad (4.34)$$

$$v_k^i = A_{v,y_k} y_k + A_{v,y_{k-1}} y_{k-1} + u_{v,k}^i \quad (4.35)$$

对于这些量化的状态结果, 将会通过计算马氏距离的方式, 来计算具体使用哪一个结果作为系统状态的最优估计值。马氏距离的计算方式如下:

$$M_k^i = \sqrt{(x_k - x_k^i)^T \Sigma_k^{-1} (x_k - x_k^i)} \quad (4.36)$$

其中, Σ_k 表示向量 x_k^- 中各个维度的协方差矩阵, 计算方式如下:

$$\Sigma_k = c_k(c_k + R_k)^{-1}c_k \quad (4.37)$$

当全部四个状态的马氏距离被计算出后, 选择距离最小的状态, 作为本次预测的结果。

惯性状态算法和状态量化算法的综合步骤流程如算法 4.1 所示。

算法 4.1 改进型卡尔曼滤波器算法

Input: 系统前一时刻的状态向量 x_{k-1} , 前一时刻的噪声协方差 c_{k-1} , 状态转移矩阵 A ,
当前时刻的系统观测向量 y_k , 前一时刻的状态观测向量 x_{k-1} , 状态转移的噪声方差 Q_k ,
观测的噪声方差 R_k , 速度状态转移矩阵 A_{v,y_k} 和 $A_{v,y_{k-1}}$, 量化状态控制矩阵 $u_{p,k}^i$ 和 $u_{v,k}^i (i=1,2,3,4)$

Output: 系统在当前时刻的状态向量 x_k , 系统在当前时刻的噪声协方差 c_k

1. **Begin**
 2. initial $m < -4$
 3. 将状态向量 x_{k-1} 拆分为位置状态向量 p_{k-1} 和速度状态向量 v_{k-1}
 4. 从状态转移矩阵 A 中拆分出位置状态转移矩阵 $A_{p,k}$
 5. 计算位置状态向量预测值: $p_k^- = A_{p,k}p_{k-1}$
 6. 计算速度状态向量预测值: $v_k^- = A_{v,y_k}y_k + A_{v,y_{k-1}}y_{k-1}$
 7. 将速度状态预测值 p_k^- 和位置状态预测值 v_k^- , 组合为状态向量预测值 x_k^-
 8. 将状态转移的噪声方差 Q_k 拆分为位置状态转移的噪声方差 $Q_{p,k}$ 和速度状态转移的噪声方差 $Q_{v,k}$
 9. 计算位置状态的协方差矩阵: $\hat{c}_{p,k}^- = A_{p,k}\hat{c}_{t-1}A_k^T + Q_{p,k}$
 10. 计算速度状态的协方差矩阵: $\hat{c}_{v,k}^- = A_{v,y_k}n_kA_{v,y_k}^T - A_{v,y_{k-1}}n_{k-1}A_{v,y_{k-1}}^T + Q_{v,k}$
 11. 将位置状态的协方差 $\hat{c}_{p,k}^-$ 和速度状态的协方差 $\hat{c}_{v,k}^-$ 组合为系统状态协方差预测值 \hat{c}_k^-
 12. 计算系统的卡尔曼增益: $K_t = \hat{c}_t^- A_t^T / (A_t \hat{c}_t^- A_t^T + R_t)$
 13. 计算系统状态值: $x_k' = x_k^- + K_k(y_k - A_k x_k^-)$
 14. 计算系统的协方差: $c_t = (I - K_t A_t)c_t^-$
 15. 计算各个维度的协方差矩阵: $\Sigma_k = c_k(c_k + R_k)^{-1}c_k$
 16. **for** $I=1:m$ **do**
 17. 计算位置状态的量化状态: $p_k^i = p_{k-1} + u_{p,k}^i$
 18. 计算速度状态的量化状态: $v_k^i = v_{k-1} + u_{v,k}^i$
 19. 将位置状态的量化状态 p_k^i 和速度状态的量化状态组合为量化状态 x_k^i
 20. 计算系统状态到量化状态 i 的马氏距离: $M_k^i = ((x_k^- - x_k^i)^T \Sigma_k^{-1} (x_k^- - x_k^i))^{1/2}$
 21. **end for**
 22. 选择马氏距离最小的量化状态, 作为系统状态
 23. **End**
-

4.3 仿真结果

使用 matlab 对此算法的有效性进行仿真测试。仿真场景中, 定位目标首先向东北方向移动, 之后转为向东南移动, 整个移动轨迹呈现一个“Λ”的形状。系统噪声 $Q = \text{diag}(0.2, 4)$, 观测噪声 $R = \text{diag}(0.2, 2)$ 。仿真结果如图 4.1 所示:

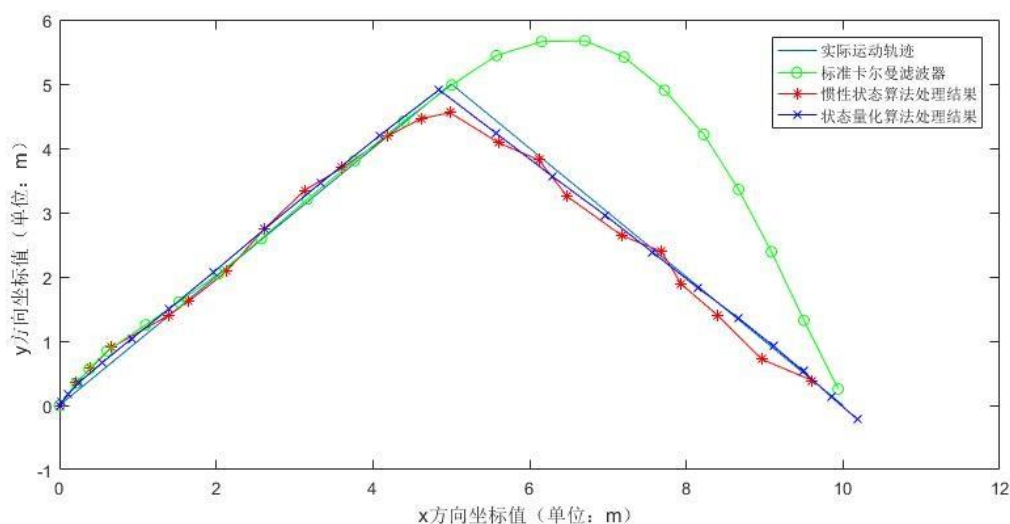


图 4.1 不同算法的效果

如图 4.1 所示, 标准卡尔曼算法在线性状态系统中有着优异的表现, 但是当系统状态突变时会产生巨大误差。惯性状态算法虽然受系统状态突变的影响较小, 但是其在常规线性情况下误差比标准卡尔曼算法高。状态量化算法同时拥有两者的特点, 在仿真场景中表现优异。误差分析结果见图 4.2。

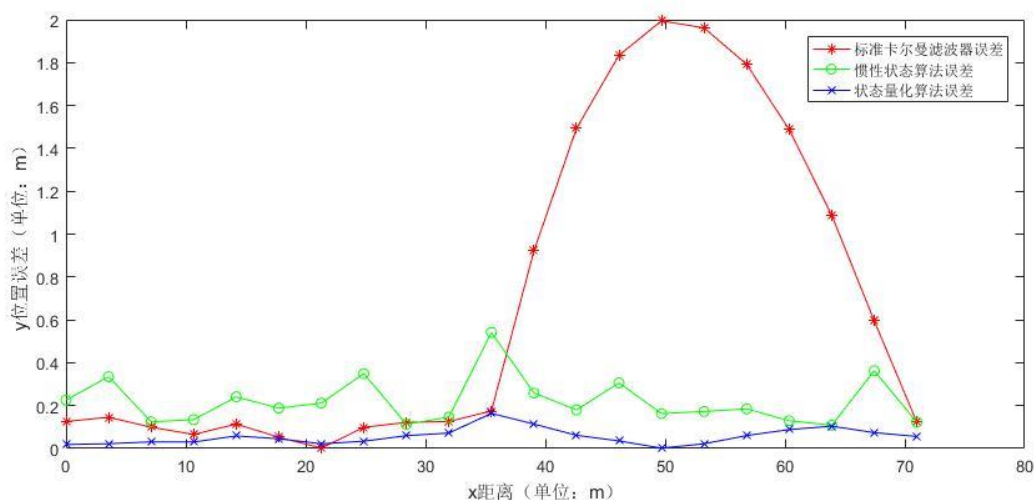


图 4.2 算法误差分析

如图 4.2 所示, 标准卡尔曼算法的平均误差值为 0.657m, 惯性状态算法的平均误差为 0.172m, 状态量化算法的平均误差为 0.081m。这表明状态量化算法的结果更加稳定和优异。

4.4 本章小结

本章首先对传统的卡尔曼滤波器的原理进行分析, 同时指出了传统卡尔曼滤波器无法处理在 WiFi 定位场景下人物移动状态突变的问题。为了解决此问题, 本次章依次提出两种改进

算法：惯性状态算法和状态量化算法。惯性状态算法会将系统状态拆分为位移状态和速度状态。其中位移状态会使用传统卡尔曼滤波器算法来进行计算。位置状态的会通过计算自身在前两个时刻的系统状态值的线性组合来获得，即使用人物在之前时刻的“运动惯性”来计算当前时刻的状态。惯性状态算法虽然能较好的解决状态突变的问题，但是其在常规场景下表现欠佳。因此本次设计又对惯性状态算法进行二次改进，提出了状态量化算法。在状态量化算法中，人物运动的预测位置不再是直接计算得出，而是从四个固定的位置中，选择一个作为下一时刻的预测位置。具体的位置选择的评判标准是各个量化状态点到惯性状态算法的计算结果的最小马氏距离。最后，本章给出几种算法在仿真环境下的表现，证实了改进后的算法的优越性。

第五章 框架设计与实验分析

5.1 框架设计

5.1.1 定位信息获取系统的框架设计

WiFi 室内定位技术的功能模块可以分为以下几个部分：

1. 数据采集模块

数据采集功能负责从无线路由器设备内部的硬件中读取 CSI 的具体数据和描述信息，包括 CSI 相位、幅度信息和无线信道状态信息以及天线状态信息。

2. 数据整理模块

数据整理模块将根据硬件工作参数和无线信道工作参数，将 CSI 信息按照天线编号、数据包编号、子载波编号进行整理和打包，使得之后的应用程序能够正常使用这些 CSI 数据。

3. 算法功能模块

算法功能模块将会进行算法功能的实现。具体内容包括：

(1) 对相关数据进行预处理，解除 2π 缠绕，过滤干扰误差，获得线性变换结果、计算工作参数

(2) 使用预处理后的数据，分别计算 ToF 和 AoA 的相关参数的数值。

(3) 使用统计计算方法，对 ToF 和 AoA 的数值进行计算

5.1.2 轨迹追踪系统的框架设计

轨迹追踪系统的功能模块可以分为以下几个部分：

1. 定位信息获取模块

定位信息获取模块将通过进程间通信的方式，从定位进程中读取被定位目标的位置信息。同时，该模块还会负责控制定位进程的启动和关闭。

2. 算法实现模块

算法实现模块将会负责使用获取得到的位置信息，进行位置轨迹的计算。此模块会对所有的定位结果进行记录，然后使用其计算运动轨迹并根据运动轨迹计算定位结果的估计值。算法估计值的误差小于直接定位结果的误差。

5.1.3 总体流程框架

本文所设计的动态目标定位系统的总体内容框架如图 5.1 所示。当应用程序被启动后会首先启动定位进程。之后会不断的尝试从定位进程中读取位置数据，并将获取到的位置数据交由数据处理算法进行处理。另一方面，定位进程被启动后，将会开始进行相关数据的获取、整理、计算并且同时还会开辟一个新的线程，用于对来自其他进程的请求进行处理。

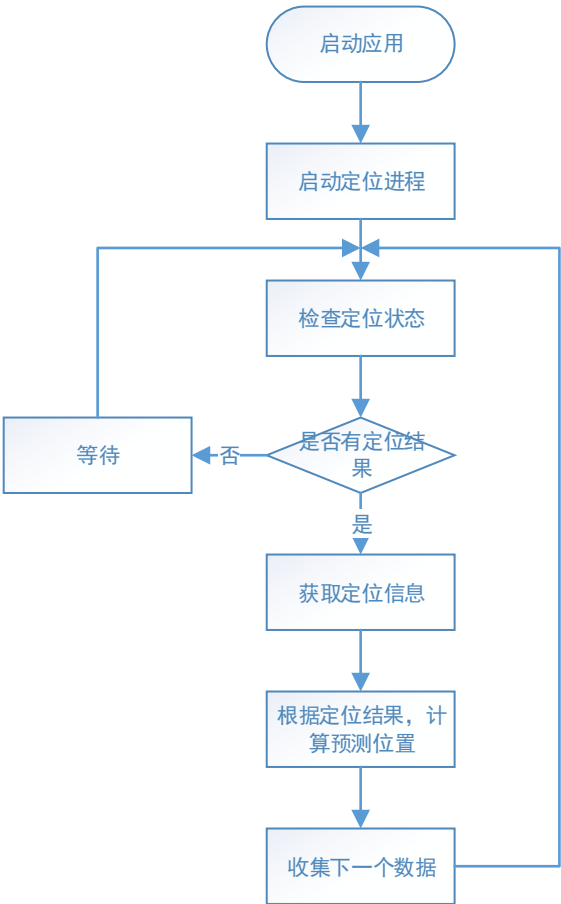


图 5.1 实验总体流程

5.2 测试准备

由于程序的实现代码是在 OpenWrt 的系统源码中实现的。所以在完成代码相关工作后，需要进行 OpenWrt 的程序编译，从而将用于实验分析的程序代码加入路由器的系统中。

5.2.1 OpenWrt 版本选择

OpenWrt 的原始代码，由于不包含 CSI-tool，所以需要对其进行安装。由于最新版本的 OpenWrt 系统因为代码改动而需要对 CSI-tool 中的内容重新进行适配，所以本次设计中将会

使用 17.01 版本的 OpenWrt 作为实验使用的操作系统版本。

5.2.2 OpenWrt 编译

代码的编译工作需要在 Linux 操作系统下完成。本次设计中，使用的电脑是 Ubuntu 14.04 的系统。表 5.1 展示了在编译开始前，所需要安装的各种工具插件。

表 5.1 OpenWrt 编译准备工具

subversion	gcc	g++	binutils	patch
bzip2	flex	bison	make	autoconf
gettext	texinfo	unzip	sharutils	subversion
libncurses5-dev	ncurses-term	zlib1g-dev	git-core	gawk
asciidoc	libz-dev	libssl-dev		

其中的部分工具会在 ubuntu 系统安装时，作为预装程序而被安装。在进行代码编译时，编译程序会自动检测所需工具是否已安装。如果没有安装或者使用的版本较低，会导致编译程序无法识别到该指令工具从而产生报错。因而在代码编译完成前，需要先对上表中的工具进行更新。

当准备工作完成后，将进行编译操作。官方在 OpenWrt 代码的主目录中的 README.md 文件中，给出了编译操作流程。表 5.2 出了需要依次执行的指令：

表 5.2 OpenWrt 编译指令

./scripts/feeds update -a
./scripts/feeds install -a
make menuconfig
make V=s -j1

./scripts/feeds update -a 指令执行之后，会从 OpenWrt 官网上下载 packages、luci、routing、telephony 这四个额外代码包。

./scripts/feeds install -a 指令，会将所下载的包添加到本地的代码路径中。

make menuconfig 指令执行后，会进入如图 5.2 所示的编译配置界面。本次设计的编译必选配置有三项。一项是第一行的“Target System”选择“AR7XXX/AR9XXX”。另一项配置项是第三行的“Target Profile”选择配置参数值为“TP-Link TL-WDR 4310 v1”。另外由于本次设计中需要收集 CSI，因此还需要启用 CSI-tool:在第十四行的“CSI”选项中，选择“recvCSI”，表示启用 CSI 工具。

make V=s -j1 指令表示正式开始编译。因为 OpenWrt 首次编译的用时很久并且会占用大量的系统资源，所以在编译开始前最好检查一遍代码状态、网络状态、系统资源占用状态以降低出现编译失败的可能性。另外由于编译期间 PC 的各项系统资源都会被大幅度占用而导致难以进行其他工作。所有编译工作多在夜晚或其他休息时间进行。

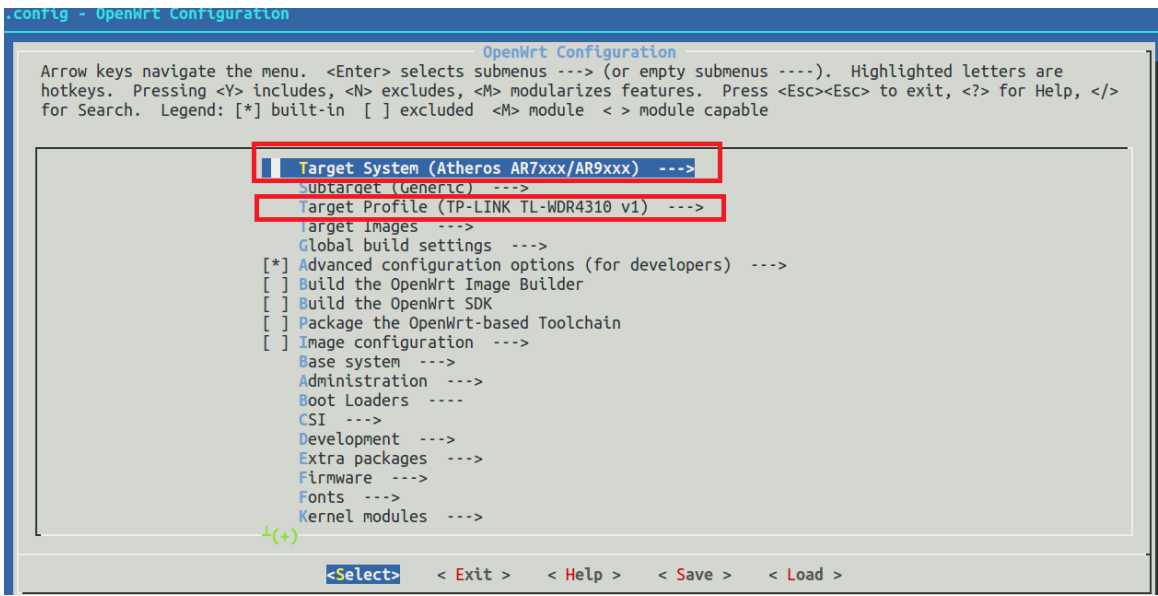


图 5.2 系统编译的配置选项

编译完成后，将生成的系统镜像文件，刷入到对本次实验设计所使用的智能路由器设备中。之后将会进行具体的实验测试。

5.3 程序效果测试

5.3.1 测试场景模型

在进行定位之前，首先需要对待测试的场景建立坐标系，以便于进行定位的计算。图 5.3 所示的为实验室的测试场景。以其俯视图对其进行建立坐标系，以路由器的中间天线为中心原点，以三天线共线的方向为 y 轴正方向，建立坐标系。



图 5.3 测试场景

建立后的坐标系模型如图 5.4 所示：

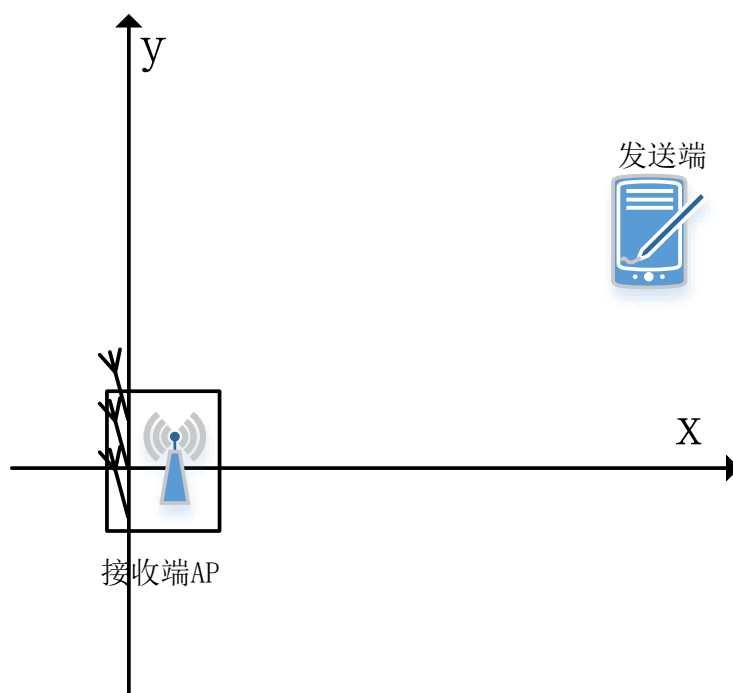


图 5.4 测试场景的坐标模型

5.3.2 测试操作步骤

1. 定位功能测试的操作步骤如下

(1) 使用无线路由器一创建无线局域网。该无线路由器中需要使用 Atheros 系列无线网卡并且装有本次设计中所编译的 OpenWrt 操作系统。之后，将无线路由器一放置再坐标场景中的原点位置处。

(2) 启动无线路由器二，连接至无线路由器一所创建的无线局域网中

(3) 使用控制终端登录无线路由器一，启动 CSI 定位功能的测试程序，能够看到测试程序在控制终端的输出如表 5.3 所示的日志内容。

(4) 将无线路由器二放置于测试场景中标示的测试点位中。

(5) 记录下当前测试点位下的输出日志中的 ToF 和 AoA 的相关信息，并根据记录下的数值使用混合式定位算法计算出被定位点的坐标值。

(6) 记录下多条日志提供的 ToF 和 AoA 值并进行坐标计算。对这些坐标进行平均值的计算。

(7) 将无线路由器二移动至下一个测试点位，待数据稳定后重复进行步骤 5、6、7 的操作。

表 5.3 定位功能测试的输出日志

283th:receive 961 byte, include 840 byte csi data
r2 is 0.013161, can't meet requirement
tof_slope=0.037214, TOF=0.000019 ms, distance=5.688702 m
offset=4.077941 cm, angle = 0.420037 (rad) , 24.066342 (degree)

2. 轨迹追踪功能和动态目标定位效果的测试步骤如下:

- (1) 使用无线路由器一创建无线局域网并使用无线路由器二连接至此网络。具体操作与定位功能测试中操作 1、2、3 相同。
- (2) 使用控制登录无线路由器, 启动轨迹追踪算法的程序, 将能够看到如图 5.5 所示的程序输出日志。
- (3) 让人员手持无线路由器二, 按照测试场景模型下预先设定的移动轨迹进行移动。
- (4) 在人员进行移动期间, 记录下轨迹追踪算法的程序的输出日志中的追踪点位的位置。
- (5) 重复进行步骤 3、步骤 4 的操作, 记录下多次测试的追踪点位的结果。

```
observed:(x:1.152676, y:3.748623) calculated:(x:0.000000, y:0.000000)
observed:(x:1.152676, y:3.748623) calculated:(x:1.152676, y:3.748623)
observed:(x:1.690151, y:3.879964) calculated:(x:1.672301, y:3.852754)
observed:(x:2.200125, y:3.909891) calculated:(x:2.180956, y:3.902435)
observed:(x:2.979989, y:4.012570) calculated:(x:2.768512, y:4.022434)
observed:(x:3.346728, y:4.129785) calculated:(x:3.192857, y:4.076892)
observed:(x:3.420264, y:3.721550) calculated:(x:3.203411, y:3.757682)
observed:(x:3.199957, y:3.230012) calculated:(x:3.155709, y:3.350769)
observed:(x:3.180062, y:3.000113) calculated:(x:3.137623, y:3.072345)
```

图 5.5 轨迹追踪算法输出日志

5.3.3 测试结果评估

在定位程序测试中, 将会使用四个不同的位置点分别进行定位测试。测试点的参考位置见图 5.6:

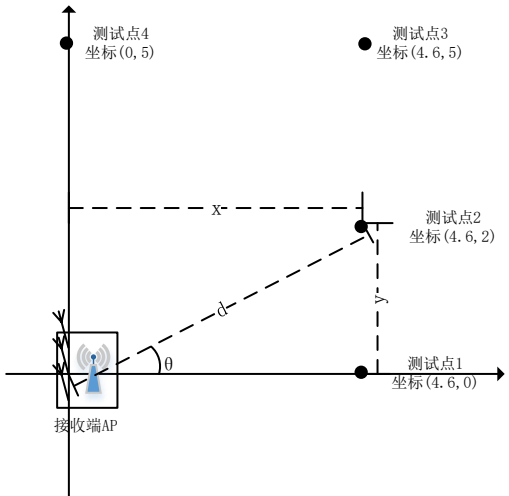


图 5.6 定位测试的测试案例

在运动定位测试中，受限与实际场景的限制，将会使用一个固定的运动路径来进行轨迹追踪算法的测试。场景模型下的运动模型如图 5.7 所示：

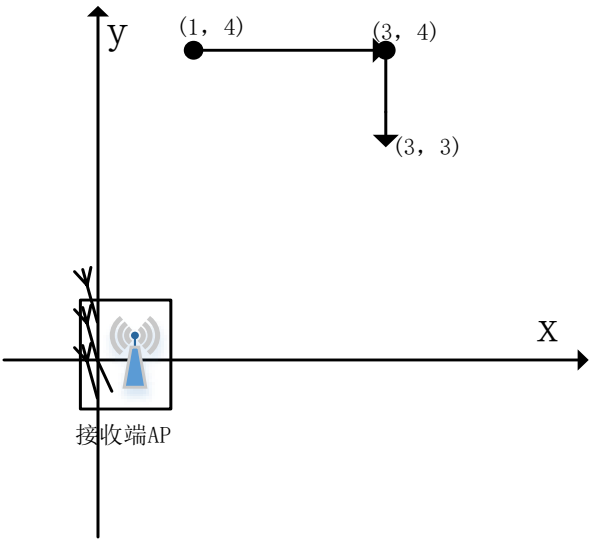


图 5.7 轨迹追踪算法的测试案例

测量值和参考值的对比结果在表 5.4 示：

表 5.4 定位结果的测试结果对比

测量结果	测量结果对应位置	参考坐标位置	误差值
$d=4.35\text{m}, \theta=-0.01\text{rad}$	$x=4.33\text{m}, y=-0.43\text{m}$	$x=4.6\text{m}, y=0\text{m}$	$\Delta x=0.27\text{m}, \Delta y=0.43\text{m}$
$d=4.63\text{m}, \theta=0.38\text{rad}$	$x=4.27\text{m}, y=1.79\text{m}$	$x=4.6\text{m}, y=2\text{m}$	$\Delta x=0.32\text{m}, \Delta y=0.21\text{m}$
$d=6.48\text{m}, \theta=0.77\text{rad}$	$x=4.41\text{m}, y=4.75\text{m}$	$x=4.6\text{m}, y=5\text{m}$	$\Delta x=0.19\text{m}, \Delta y=0.25\text{m}$
$d=4.49\text{m}, \theta=1.55\text{rad}$	$x=0.23\text{m}, y=4.49\text{m}$	$x=0\text{m}, y=5\text{m}$	$\Delta x=0.23\text{m}, \Delta y=0.51\text{m}$

可以看到这种基于 CSI 相位的定位算法，能够提供精度为 $\pm 0.7\text{m}$ 定位结果。并且在大部分情况下的定位精度能达到 $\pm 0.5\text{m}$ 。测试中的最大误差百分比为 11.7%。

表 5.5 展示了卡尔曼滤波器的轨迹追踪算法的表现结果。可以看到，正常情况下，轨迹追踪的追踪点位置与实际位置点之间的误差在 $\pm 0.05\text{m}$ 之内。即使在位置运动状态发生突变的时刻附近，追踪点位置值与实际值的误差也和正常情况下相同，为 0.03m。

表 5.5 基于卡尔曼滤波器的轨迹追踪结果

计算出的位置	参考位置	差值
$x=1.00\text{m}, y=4.00\text{m}$	$x=1.00\text{m}, y=4.00\text{m}$	$\Delta x=0.00\text{m}, \Delta y=0.00\text{m}$
$x=1.48\text{m}, y=3.96\text{m}$	$x=1.50\text{m}, y=4.00\text{m}$	$\Delta x=0.02\text{m}, \Delta y=0.04\text{m}$
$x=1.99\text{m}, y=4.01\text{m}$	$x=2.00\text{m}, y=4.00\text{m}$	$\Delta x=0.01\text{m}, \Delta y=0.01\text{m}$
$x=2.49\text{m}, y=4.02\text{m}$	$x=2.50\text{m}, y=4.00\text{m}$	$\Delta x=0.01\text{m}, \Delta y=0.02\text{m}$
$x=2.98\text{m}, y=4.02\text{m}$	$x=3.00\text{m}, y=4.00\text{m}$	$\Delta x=0.02\text{m}, \Delta y=0.02\text{m}$
$x=2.98\text{m}, y=3.76\text{m}$	$x=3.00\text{m}, y=3.75\text{m}$	$\Delta x=0.02\text{m}, \Delta y=0.01\text{m}$
$x=2.98\text{m}, y=3.51\text{m}$	$x=3.00\text{m}, y=3.50\text{m}$	$\Delta x=0.02\text{m}, \Delta y=0.01\text{m}$
$x=2.99\text{m}, y=3.25\text{m}$	$x=3.00\text{m}, y=3.25\text{m}$	$\Delta x=0.01\text{m}, \Delta y=0.00\text{m}$
$x=3.01\text{m}, y=2.99\text{m}$	$x=3.00\text{m}, y=3.00\text{m}$	$\Delta x=0.01\text{m}, \Delta y=0.01\text{m}$

图 5.8 表示使用改进算法后的卡尔曼滤波器的轨迹追踪结果。可以看到，本次设计中所提出的惯性状态算法和状态量化算法，能够很好的对人员的运动情况进行追踪。图 5.8 中坐标 (3,4) 的位置，为本次设计中的运动方向变化点，也是线性系统的状态突变点。可以看到本次设计所提出的算法在该点处的表现较好，与仿真结果基本一致。

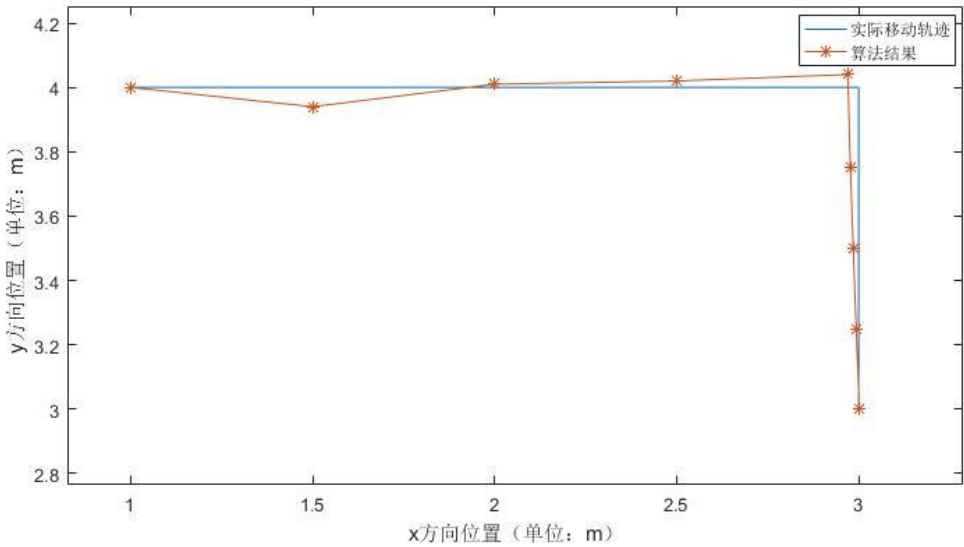


图 5.8 轨迹追踪结果

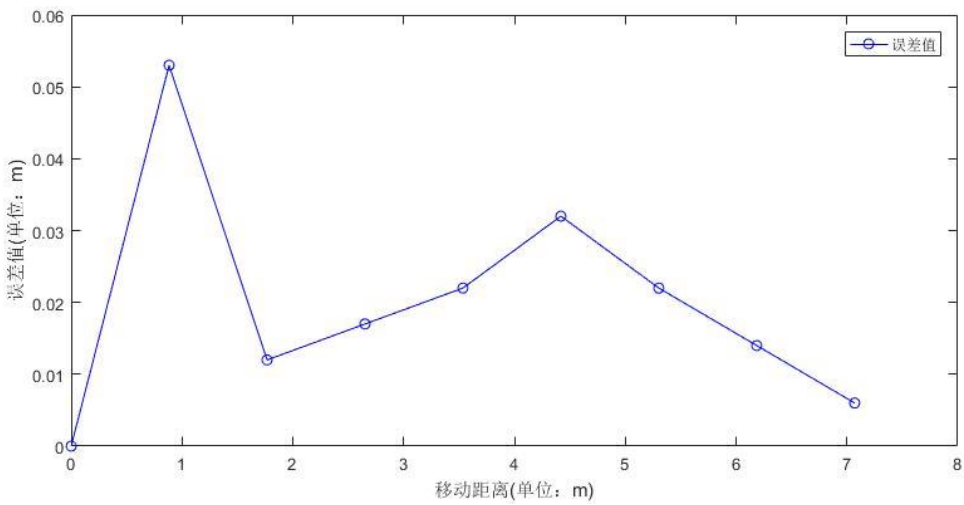


图 5.9 实验误差分析

在图 5.9 中可以看到测试场景下的算法误差值。在测试场景下，改进后的卡尔曼滤波器的最大误差值为 0.054m，平均误差值为 0.021m，轨迹追踪中计算出的追踪点位置精度达到了厘米级别。表现结果与仿真结果基本相同。

5.4 本章小结

在本章中，首先介绍了本次设计的程序总体框架。之后，介绍进行实验测试所需要准备

工作：选择合适的 OpenWrt 的版本，以及进行相关代码编译，使得本次设计的内容能够被加入到路由器操作系统中。最后，对本次设计程序在实际场景下进行测试和分析。在本次设计的场景中，定位算法的最大误差值为 0.7m，最大误差百分比为 11.7%；轨迹追踪算法的平均误差为 0.021m，最大误差值为 0.054m。

第六章 总结与展望

6.1 本文总结

本文主要的研究内容是研究使用信道状态信息的室内定位技术。文章首先介绍了目前互联网产业与传统行业相结合的发展模式,并指出该种模式下位置服务有着非常广阔的市场空间和发展未来。之后本文介绍了各种室内定位技术,并着重介绍了 WiFi 定位技术和 CSI 混合式定位方案。针对目前 WiFi 定位系统中广泛应用的指纹定位系统移动不便、研究领域热点的 MUSIC 定位算法计算复杂的问题,本文提出了一种适用于运动物体定位的综合算法。首先,本文分别提出了一种 CSI 误差过滤、电磁波 ToF 测量、电磁波 AoA 测量的算法,并通过将这些算法的结果进行组合的方式来实现对物体进行快速定位。同时,本文还实现了一种基于卡尔曼滤波器的改进算法,用于实现对运动目标的运动轨迹追踪和定位误差优化。最后,本文对这些算法进行仿真与实验。总结全文,主要工作如下:

1. 本文在查阅大量文献的基础上,深入地研究了 CSI 定位技术的原理,分析了现有的 CSI 定位算法和 CSI 相位处理算法的缺点。本文提出了一种基于方差分布的误差过滤算法。该算法使用一种线性变换的方法,将对 CSI 数据值的分析转变为对线性变换结果的分析。同时,该算法会计算不同的误差过滤参数对线性变换结果的均方差的影响,从而能够判断使用具体使用什么参数能够获得最优误差过滤效果。之后本文还提出了一种基于 CSI 相位的距离计算算法和一种基于 CSI 相位的到达角计算算法。

2. 本文通过查阅资料和编程测试的方式,对卡尔曼滤波器进行了研究。本文研究发现传统卡尔曼滤波器算法在 WiFi 定位的场景下进行轨迹追踪时,表现效果欠佳。针对此问题,本文提出惯性状态算法。在惯性状态算法中,系统状态中的速度状态不再是通过卡尔曼增益进行状态更新计算获得,而是通过对系统在之前时刻的“惯性状态”进行线性处理所获得。惯性状态算法能够很好的处理传统卡尔曼滤波器无法处理的系统状态突变的问题。虽然惯性状态算法能较好的处理系统状态突变的问题,但是该算法在常规条件下的误差较大,所以本次设计中又对该算法进行了改进,提出了状态量化算法。状态量化算法将系统状态量化固定,通过计算理论值与量化值的马氏距离来选择使用哪一个状态作为系统下一时刻的状态值。将惯性状态算法和状态量化算法结合所得到的改进型卡尔曼滤波器算法能够很好的处理系统状态突变的问题,并且在常规场景下也有着不俗的表现。

此外,本次设计中由于设计需要,还对 OpenWrt 系统的一些代码结构进行了梳理。尽管

相关内容未在设计中体现出来,但是这些工作无疑是能极大的方便后续的其他工作。

6.2 后续展望

本文提出的方法,尽管在功能上满足了要求,但是由于研究时长以及实验的条件等原因,本系统还处在初步设计阶段,还存在着许多的不足之处。基于本文的研究程度以及系统中存在的不足之处给出以下几个方面进行改进:

1. 本次设计中使用的基于方差分布的误差过滤算法由于自身包含三重循环,导致算法运行效率低下。因此,在实验中的测试程序中,该算法只会在程序启动时运行一次,后续的处理步骤都会是沿用之前的计算结果,而无法针对环境的变化而改变相关参数。这无疑降低了程序的实用性。如果要解决此问题,则需要对算法进行进一步优化,如增加能够跳过计算的判断条件、使用优化的拟合模型、使用梯度计算的方式进行局部计算而非整体计算等。此外,随着 WiFi 系统的发展,新一代的 WiFi 协议为 CSI 数据提供了更加强大的基础,包括更大的带宽,支持更多天线的 MIMO 系统等,最新的 802.11 协议中还增加了精准测时机制来支持 WiFi 测距技术,让利用 CSI 进行测距的定位方案具有了更多的潜力与方向^[57]。

2. 本次设计中提出的状态量化算法,只会在状态误差小于一定数值时才能发挥出较好的效果。在仿真模拟测试中,当观测误差值大于一个阈值时,状态量化算法就会产生状态结果的误判。对该问题的一种较好的处理方法是增加量化状态点的数目,从而能够避免对一些难以界定的中间值进行误判。

参考文献

- [1] Chen L, Thombre S, Järvinen K, et al. Robustness, security and privacy in location-based services for future IoT: A survey[J]. IEEE Access, 2017, 5: 8956-8977.
- [2] Chung J, Sargent L, Brown R, et al. GPS Tracking Technologies to Measure Mobility-Related Behaviors in Community-Dwelling Older Adults: A Systematic Review[J]. Journal of Applied Gerontology, 2021, 40(5): 547-557.
- [3] 张晨, 闫芳君, 王阳, 胡阳. 基于伪卫星技术的导航定位与应用探讨[J]. 中国设备工程, 2020(24): 212-213.
- [4] Rai A, Chintalapudi K K, Padmanabhan V N, et al. Zee: Zero-effort crowdsourcing for indoor localization[C]//Proceedings of the 18th annual international conference on Mobile computing and networking. 2012: 293-304.
- [5] 刘挺, 杜小甫, 黄昆达, 等. 高精度室内定位算法与技术综述[J]. 电子测试, 2021(5): 73-75
- [6] Guo K, Qiu Z, Miao C, et al. Ultra-wideband-based localization for quadcopter navigation[J]. Unmanned Systems, 2016, 4(01): 23-34.
- [7] Kotaru M, Zhang P, Katti S. Localizing low-power backscatter tags using commodity wifi[C]//Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies. 2017: 251-262.
- [8] Guo X, Li L, Ansari N, et al. Accurate WiFi localization by fusing a group of fingerprints via a global fusion profile[J]. IEEE Transactions on Vehicular Technology, 2018, 67(8): 7314-7325.
- [9] 孙全, 黄楷, 陈宇威, 等. 红外跟踪系统单站被动定位方法研究[J]. 光学与光电技术, 2018, 16(5): 55-59.
- [10] Topak F, Peker i M K, Tanyer A M. Technological viability assessment of Bluetooth low energy technology for indoor localization[J]. Journal of computing in civil engineering, 2018, 32(5): 04018034.
- [11] Khyam M O, Alam M J, Lambert A J, et al. High-precision OFDM-based multiple ultrasonic transducer positioning using a robust optimization approach[J]. IEEE Sensors Journal, 2016, 16(13): 5325-5336.
- [12] Armstrong J, Sekercioglu Y A, Neild A. Visible light positioning: a roadmap for international standardization[J]. IEEE Communications Magazine, 2013, 51(12): 68-73.
- [13] Khattar F, Luthon F, Larroque B, et al. Visual localization and servoing for drone use in indoor remote laboratory environment[J]. Machine Vision and Applications, 2021, 32(1): 1-13.
- [14] Shimosaka M, Saisho O, Sunakawa T, et al. ZigBee based wireless indoor localization with sensor placement optimization towards practical home sensing[J]. Advanced Robotics, 2016, 30(5): 315-325.
- [15] 何瑞. 基于地磁的室内定位方法研究[D]. 西安电子科技大学, 2020.
- [16] 闫大禹, 宋伟, 王旭丹, 等. 国内室内定位技术发展现状综述[J]. 收藏, 2019, 7(4): 5-12.
- [17] Sneha V, Nagarajan M. Localization in Wireless Sensor Networks: A Review[J]. Cybernetics and Information Technologies, 2020, 20(4): 3-26.
- [18] Yang Z, Zhou Z, Liu Y. From RSSI to CSI: Indoor localization via channel response[J]. ACM Computing Surveys (CSUR), 2013, 46(2): 1-32.
- [19] Wu K, Xiao J, Yi Y, et al. FILA: Fine-grained indoor localization[C]//2012 Proceedings IEEE INFOCOM. IEEE, 2012: 2210-2218.
- [20] Vasisht D, Kumar S, Katabi D. Decimeter-level localization with a single WiFi access point[C]//13th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 16). 2016: 165-178.
- [21] Kotaru M, Katti S. Position tracking for virtual reality using commodity WiFi[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017: 68-78.

- [22] Han S, Li Y, Meng W, et al. Indoor localization with a single Wi-Fi access point based on OFDM-MIMO[J]. IEEE Systems Journal, 2018, 13(1): 964-972.
- [23] Tong X, Liu K, Tian X, et al. Fineloc: A fine-grained self-calibrating wireless indoor localization system[J]. IEEE Transactions on Mobile Computing, 2018, 18(9): 2077-2090.
- [24] Gjengset J, Xiong J, McPhillips G, et al. Phaser: Enabling phased array signal processing on commodity Wi-Fi access points[J]. GetMobile: Mobile Computing and Communications, 2015, 19(2): 6-9.
- [25] Kotaru M, Joshi K, Bharadia D, et al. Spotfi: Decimeter level localization using wifi[C]//Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication. 2015: 269-282.
- [26] Gong W, Liu J. Robust indoor wireless localization using sparse recovery[C]//2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS). IEEE, 2017: 847-856.
- [27] Zhang R, Xia W, Yan F, et al. A single-site positioning method based on TOA and DOA estimation using virtual stations in NLOS environment[J]. China Communications, 2019, 16(2): 146-159.
- [28] Demeechai T, Kukieattikool P, Ngo T, et al. Localization based on standard wireless LAN infrastructure using MIMO-OFDM channel state information[J]. EURASIP Journal on Wireless Communications and Networking, 2016, 2016(1): 1-16.
- [29] Shao Z C, Wang D, Zhao R, et al. WILS: Wireless Indoor Localization System using Commercial WiFi Infrastructures with Decimeter Accuracy[C]//3rd International Conference on Wireless Communication and Sensor Networks (WCSN 2016). Atlantis Press, 2016: 270-275.
- [30] Ahmed A U, Arablouei R, De Hoog F, et al. Estimating angle-of-arrival and time-of-flight for multipath components using wifi channel state information[J]. Sensors, 2018, 18(6): 1753.
- [31] Chen Z, Jiang Y, Han W, et al. A Modified RSSI-Based Indoor Localization Method in Wireless Sensor Network[C]//2014 International Conference on Mechatronics, Control and Electronic Engineering (MCE-14). Atlantis Press, 2014.
- [32] Wang Y, Xiu C, Zhang X, et al. WiFi indoor localization with CSI fingerprinting-based random forest[J]. Sensors, 2018, 18(9): 2869.
- [33] Dang X, Si X, Hao Z, et al. A novel passive indoor localization method by fusion CSI amplitude and phase information[J]. Sensors, 2019, 19(4): 875.
- [34] 何康辉, 董朝阳. 基于正交变换的五阶容积卡尔曼滤波导航算法[J]. 系统工程与电子技术, 2020, 42(3): 680-685.
- [35] 李开龙, 胡柏青, 常路宾. 改进四元数无味卡尔曼滤波算法[J]. 系统工程与电子技术, 2016, 38(6): 1399-1404.
- [36] Enkhtur M, Cho S Y, Kim K H. Modified unscented Kalman filter for a multirate INS/GPS integrated navigation system[J]. Etri Journal, 2013, 35(5): 943-946.
- [37] Benzerrouk H, Nebylov A V. Integrated navigation system ins/gnss based on joint application of linear and nonlinear filtering[J]. IFAC Proceedings Volumes, 2012, 45(1): 208-213.
- [38] 王珊婷. 高精度室内定位方法研究与实现[D]. 北方工业大学, 2017.
- [39] 朱明清, 吴冈. 室内定位方法的分析和对比[J]. 自动化技术与应用, 2017, 36(8):61-63,80.
- [40] 卢会芳. WIFI 室内定位关键技术研究[D]. 电子科技大学, 2016.
- [41] 周成. 面向 5G 通信索引调制技术研究及实现[D]. 电子科技大学, 2017.
- [42] Elango K, Muniandi K. Hardware implementation of FFT/IFFT algorithms incorporating efficient computational elements[J]. Journal of Electrical Engineering & Technology, 2019, 14(4): 1717-1721.
- [43] 陈丽琴. Study of WiFi Signal Propagation and the Location of the Wireless AP[J]. 无线通信, 2021, 11(02): 26-35.
- [44] Kaczmarek M, Ruminski J, Bujnowski A. Accuracy analysis of the RSSI BLE SensorTag signal for indoor localization purposes[C]//2016 Federated Conference on Computer Science and Information Systems

- (FedCSIS). IEEE, 2016: 1413-1416.
- [45] Breslin D, Akay E. Systems and methods for compensating antenna gain imbalance: U.S. Patent 8,971,948[P]. 2015-3-3.
- [46] 陈慕涵, 郭佳佳, 李潇, 等. 基于深度学习的大规模 MIMO 信道状态信息反馈[J]. 物联网学报, 2020, 4(1): 33-44.
- [47] Munanar H, Arif T Y. AARF-HT: Adaptive auto rate fallback for high-throughput IEEE 802.11 n WLANs[J]. International Journal of Communication Networks and Information Security, 2018, 10(1): 170-179.
- [48] 杨轶. IEEE 802.11 n 信号解调的设计与实现[D]. 重庆邮电大学, 2017.
- [49] Atbaei Z, Tadaion A. Interference alignment in MIMO interference broadcast channels with imperfect CSI[J]. IET communications, 2019, 13(5): 469-480.
- [50] Sanam T F, Godrich H. A multi-view discriminant learning approach for indoor localization using amplitude and phase features of CSI[J]. IEEE Access, 2020, 8: 59947-59959.
- [51] Gan D, Xie S, Liu Z X. Stability of the distributed Kalman filter using general random coefficients[J]. Science China Information Sciences, 2021, 64(7): 1-14.
- [52] 彭丁聪. 卡尔曼滤波的基本原理及应用[J]. 软件导刊, 2009, 8(11): 32-34.
- [53] Sun T, Wu R, Cui Y, et al. Sequent extended Kalman filter capacity estimation method for lithium-ion batteries based on discrete battery aging model and support vector machine[J]. Journal of Energy Storage, 2021, 39: 102594.
- [54] Böhler L, Ritzberger D, Hametner C, et al. Constrained extended Kalman filter design and application for on-line state estimation of high-order polymer electrolyte membrane fuel cell systems[J]. international journal of hydrogen energy, 2021, 46(35): 18604-18614.
- [55] 赵天东. 基于卡尔曼滤波器残差的矢量跟踪完好性研究[J]. 遥测遥控, 2021, 42(03): 21-25.
- [56] Conjard M, Grana D. Ensemble-Based Seismic and Production Data Assimilation Using Selection Kalman Model[J]. Mathematical Geosciences, 2021: 1-24.
- [57] Rea M, Abrudan T E, Giustiniano D, et al. Smartphone positioning with radio measurements from a single wifi access point[C]//Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies. 2019: 200-206.

附录 1 程序清单

第四章 差分计算 CSI 相位的定位程序

CSI_all: 单天线全部 CSI 数据计算 Matlab 仿真代码

CSI_phase_all: CSI 相位状态分析 Matlab 仿真代码

t_σ: 最优误差阈值计算的 Matlab 仿真代码

CSI_aoa_gausse: 高斯拟合计算 AoA 的 Matlab 仿真代码

CSI_tof_gausse: 高斯拟合计算 ToF 的 Matlab 仿真代码

第五章 改进型卡尔曼算法程序。

Kalman_old: 标准卡尔曼算法 Matlab 仿真代码

Kalman_changed: 改进型卡尔曼算法 Matlab 仿真代码

第六章 OpenWrt 系统程序

CSI_location: CSI 定位算法 C 程序

WiFi_Kalman: 改进型卡尔曼算法 C 程序

附录 2 攻读硕士学位期间申请的专利

(1)张戟龙、马大禹，“一种基于信道状态信息的相位的距离计算方法”，202110377614.1, 2021.4.

致谢

不知不觉中，我已经来到南京邮电大学三年了。三年时光过得飞快，让我有一种仿佛昨天还是开学典礼的感觉。当时，刚刚就读研究生的我，还不知道接下来的三年会有怎样的人生旅程，对一切感到茫然和未知。但是我在这所学校里遇到的老师、同学和家人都给了我帮助和鼓励。在此，我真诚地向我的家人、老师、同学表达我衷心的感谢！

首先我要感谢我的导师张载龙教授。张教授有在智能无线设备领域有着多年的经验和丰富的理论。同时张教授还对我们学生非常认真负责。即使工作再忙，也依然会抽出时间来对我的科研进度和学习成果进行仔细把关。每当我遇到难题时，张老师总能指点迷津，给出关键的建议，让我极为崇拜。在这个过程中，我深切感受到张老师学识渊博、高尚的自我修养和严谨的治学精神，这让我受益匪浅，正是在张老师的指导下，我才能顺利完成毕业论文以及相关文档的撰写改进工作。感谢张老师在学术研究上给予我深厚的指导。

同时，我要感谢研究生期间的师兄周健伟同学。周师兄在我进行课题研究时，曾经多次的给予我非常大地帮助。在我的研究生最后一年，周师兄已经在苏州工作了。但是对于我在研究课题中遇到的问题，周师兄依然不辞辛劳，亲自过来指点我，让我很受感动。

另外，我还要感谢网络交流论坛、QQ 群中的不知名的前辈。你们的指点让我在课题研究的过程中少走了很多弯路。让我能够很快地脱离连基础知识都不是很懂的小白身份。感谢你们对我的帮助。

最后，我感谢我的父母，你们为我倾注了无数的心血，也身体力行的影响到我的性格、价值观等所有方面，我想，成长的路虽然不平，但最终不会让你们失望。

至此，再一次感谢所有关心和帮助过我的人们，感谢论文评审的各位老师和专家们，感谢南京邮电大学给我提供了这样一个学习的平台，谢谢！