

编 号	HR_SJ_QRSW4
密 级	内部公开
阶 段	发布
页 数	19

代 号

名 称

HR-RTLS4 上位机软件开发手册

基于 TDOA 的高精度 UWB 定位开发套件

会 签

大连浩如科技有限公司

文档控制

变更记录

版本号	日期	增加/修改/删除	简单描述	更改申请单号
V1.0	20240514	创建	根据模板创建，描述上位机软件开发方案和流程。	

目 录

1	系统概况	1
2	系统拓扑结构	1
3	上位机软件基本功能	2
4	上位机软件整体架构	3
5	数据解析	5
6	核心算法	6
6.1	Chan 算法.....	8
6.2	Taylor 算法.....	9
6.3	Chan_Taylor	10
6.4	卡尔曼滤波	11
6.5	坐标滤波	12
6.6	权重设计	12
7	UI 设计、更新、显示	13
8	开发环境说明	14
9	通讯协议	15
9.1	UDP 通讯协议	15
9.2	数据转发协议	15

1 系统概况

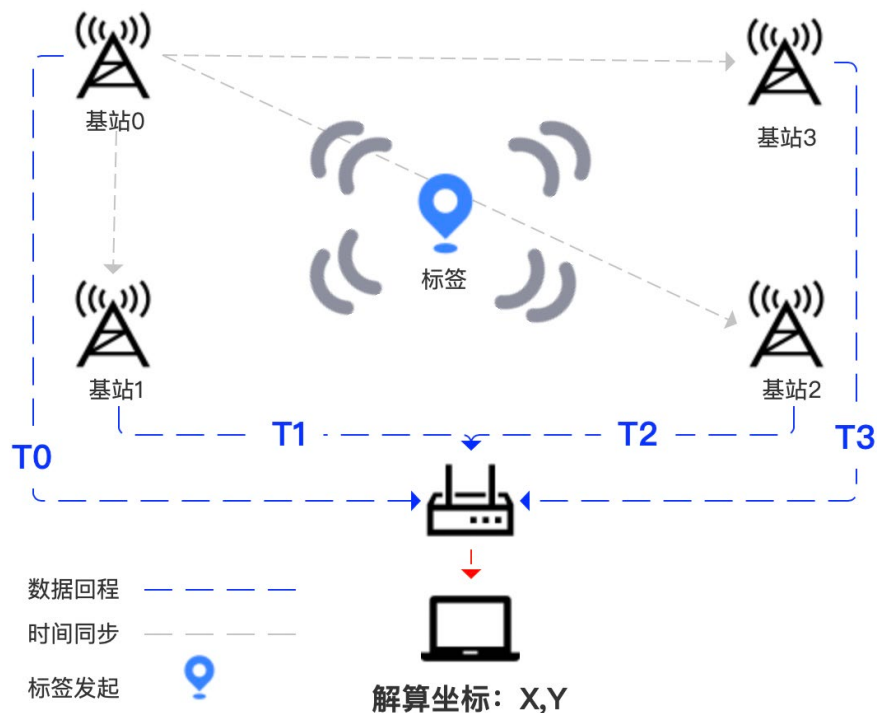
本上位机开发手册适用于浩如科技 HR-RTLS4 系列 UWB 开发套件，HR-RTLS4 采用 TDOA 定位方式，拓扑为全无线模式，基站采用通过 WIFI 接入网络，基站与 PC 接入统一局域网内，通过 UDP 数据传输。

TDOA 部分核心算法采用 Taylor+Chan 的方式，能够应用于多组 TDOA 测量值，对 TDOA 误差的统计特性没有特殊要求，能在适当的噪声水平上提供较为准确的定位估计。其鲁棒性相比单纯使用 Chan 等其他算法解析解算法有明显优势。贴合实际使用场景。

在阅读本手册之前，建议先观看开箱测试视频和阅读对应产品型号的用户手册，以便初步了解系统功能和使用方法。本手册也可以结合《课程 2：TDOA 嵌入式软件流程》视频教程进行同步学习。

2 系统拓扑结构

系统部署无需布线、无需同步节点模式。基站、上位机连入 WIFI，标定好基站坐标即可解算标签位置



3 上位机软件基本功能

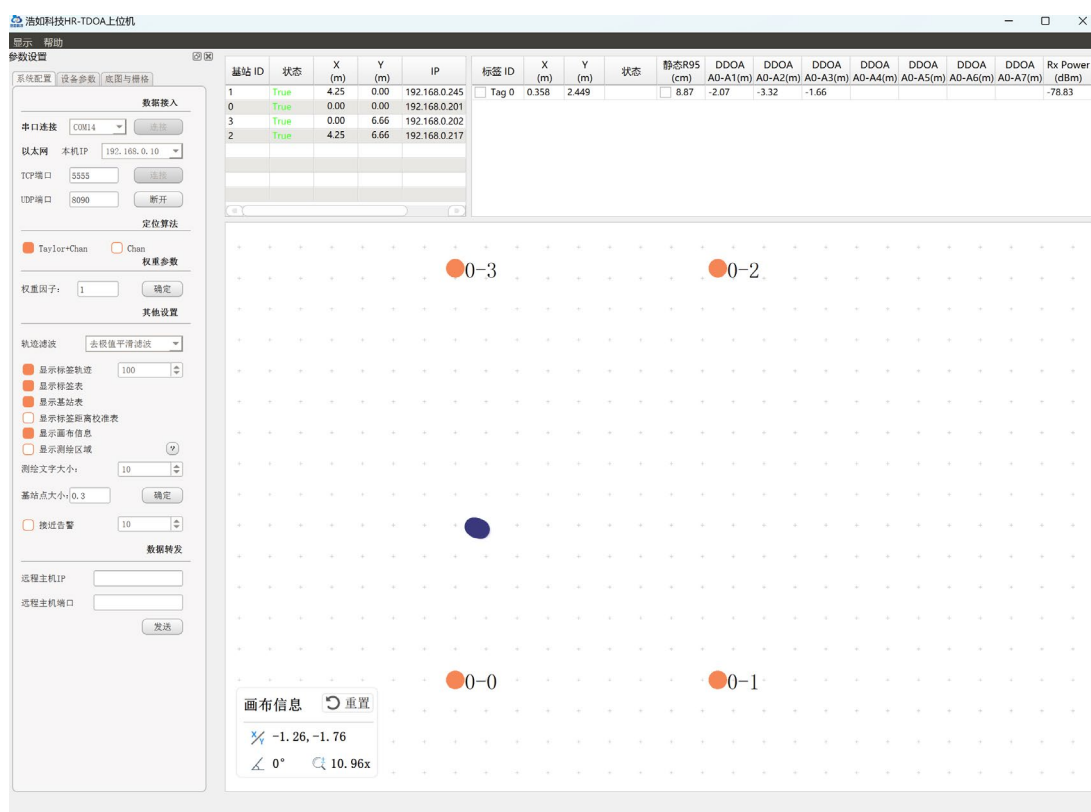


图 3-1 上位机软件截图

上位机软件基本功能包括：

1. **基站列表：**对基站进行标定、基站在线状态、IP、ID 获取展示。
2. **标签列表：**实时显示标签 DDOA 计算值、以及解算后的坐标。
3. **连接管理&参数设置：**各类参数设置，功能开关以及连接管理。
4. **定位展示：**二维视图形式显示基站与标签的位置点。
5. **轨迹显示：**显示标签移动轨迹路线。
6. **地图设置：**可上传地图，调整比例直观查看定位状态。
7. **数据转发：**可对坐标解算数据进行转发，方便二次开发、数据对接。

4 上位机软件整体架构

如图 4-1 所示，代码整体架构主要分为网络通讯、数据解析、算法解算、UI 展示四部分。

网络通讯

代码位于 `rtls_pc_tdoa/network/RTLSClient.cpp`，通过 UDP 进行网络连接，对数据进行协议解析以及存储，这部分除了要考虑协议对接以外，还要考虑 UDP 可靠性以及网络延迟对数据连续性的影响。

数据解析部分

代码位于 `rtls_pc_tdoa/network/RTLSClient.cpp` 中。通过验证有效包头、数据长度以及字段合法性等方法进行容错，对非法数据截获抛弃。原始数据为各时间戳数据，解析之后要对数据进行汇总计算得到对应的 TDOA(时间差)及 DDOA(距离差)数组

算法解算部分

代码位于 `rtls_pc_tdoa/tdoa/` 目录中。

核心算法部分采用 Taylor+Chan 算法进行坐标解算，通过解析的多组 DDOA 值进行双曲线方程进行加权求解，结合 Kalman、去极值滤波等手段进行滤波，文章后面会对算法部分进行展开讲解。

UI 部分

代码位于 `rtls_pc_tdoa/views` 目录中。主要包含基本 UI 创建，地图绘制等 UI 相关操作更新。

算法部分由浩如科技开发，是实现整体功能的核心代码，建议重点学习和熟练掌握。主要算法目录路径 `rtls_pc_tdoa/tdoa/`

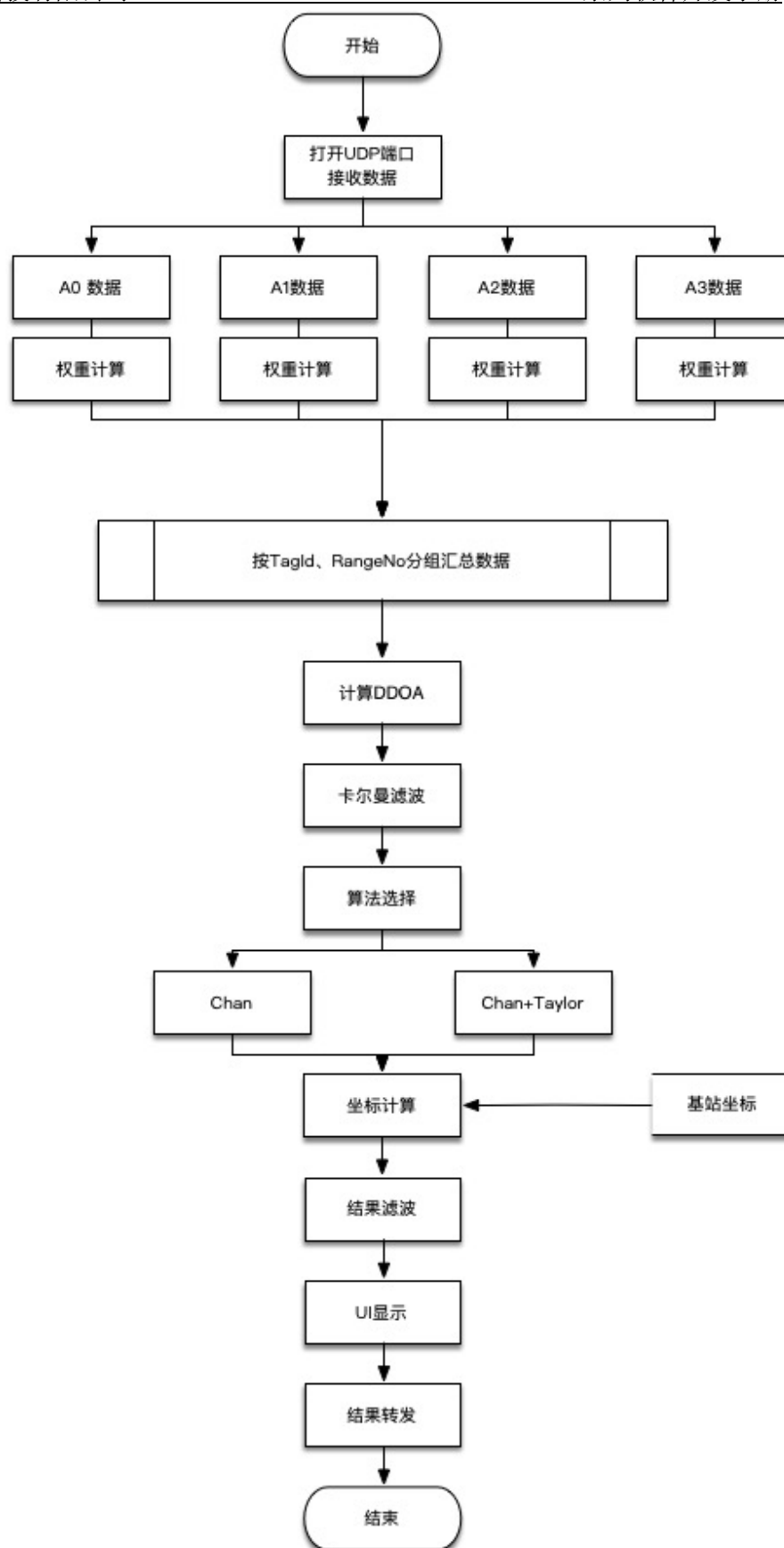


图 4-1 上位机软件整体流程

5 数据解析

代码位于 `rtls_pc_tdoa/network/RTLSCLinet.cpp`, `rtls_pc_tdoa/network/tdoa.cpp` 中。

RTLSCLinet 中负责连接、断开等状态处理, UDP 数据包获取、连包处理等。

而 `tdoa.cpp` 文件中对原始 UDP 数据进行汇总计算, 从而得到 TDOA、DDOA 等值, 为下一步计算做准备。流程如图 5-1 所示

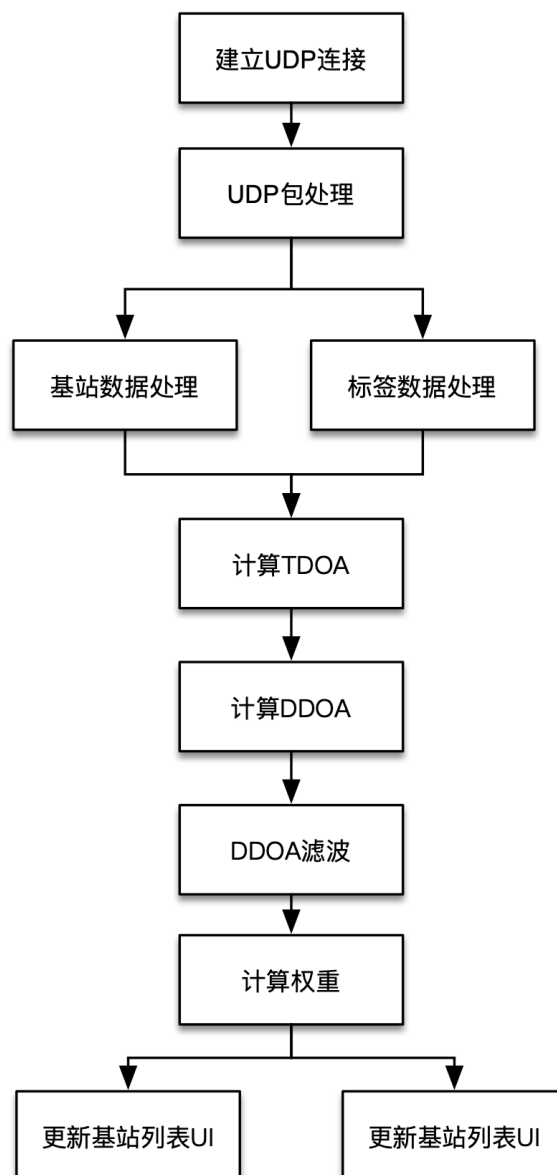


图 5-1 数据解析流程图

6 核心算法

核心算法为 Taylor+Chan 联合算法。使用 GSL 数学库进行矩阵计算。Taylor 算法文件位于 `rtls_pc_tdoa/tdoa/Taylor_tdoa.cpp`

Chan 算法和 Taylor 算法是 2 种经典的 TDOA 定位算法。其中，Chan 算法在时差值精确的情况下，可以实现较高精度的定位，但如果时差值精度不够，其定位精度会大幅降低。Taylor 算法则是在已有的定位坐标基础上，进行迭代递归，使定位出的坐标接近于目标的真实坐标。虽然 Taylor 算法定位精度较高，但需要提供初始估计坐标，否则就无法实现准确定位。基于 2 种算法的特点，我们将 Chan 算法解算出的目标坐标作为初始估计坐标值赋给 Taylor 算法进行迭代运算，即使获取的时差值存在一定误差，使初始估计坐标的精度不高，但可以通过迭代来提高定位坐标的精度。

通过算法对比和仿真与实验分析表明，Chan-Taylor 联合算法较 Chan 算法具有更高的定位精度和稳定性，较 Taylor 算法更具有实用性。

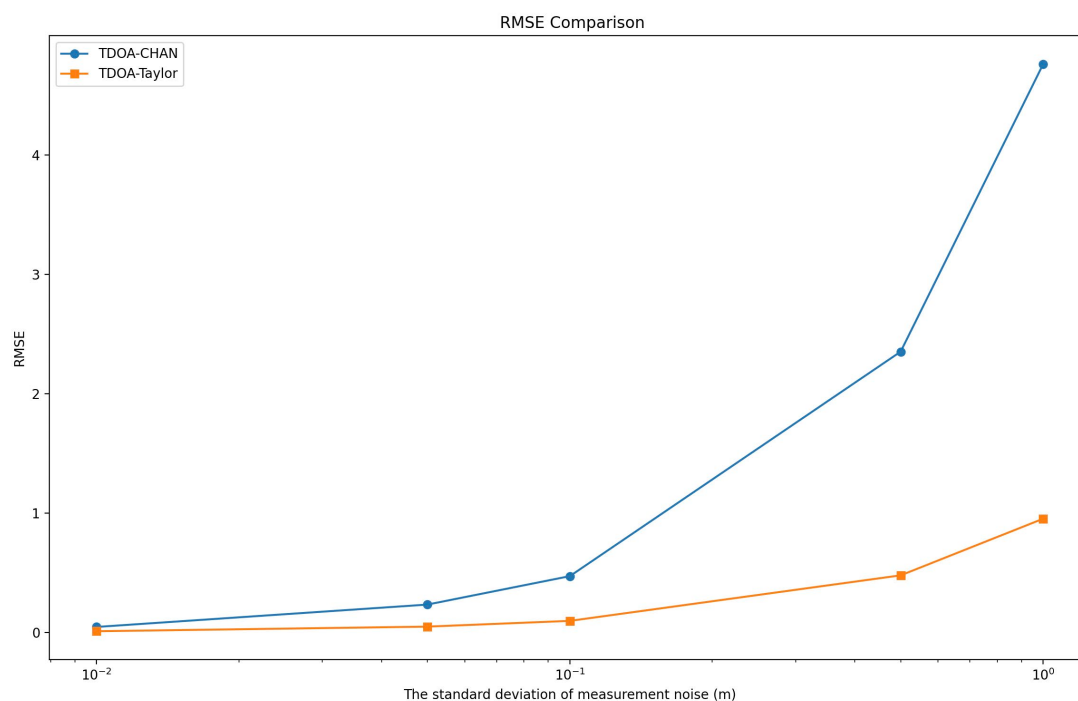


图 6-1 不同算法噪声 RMSE 对比

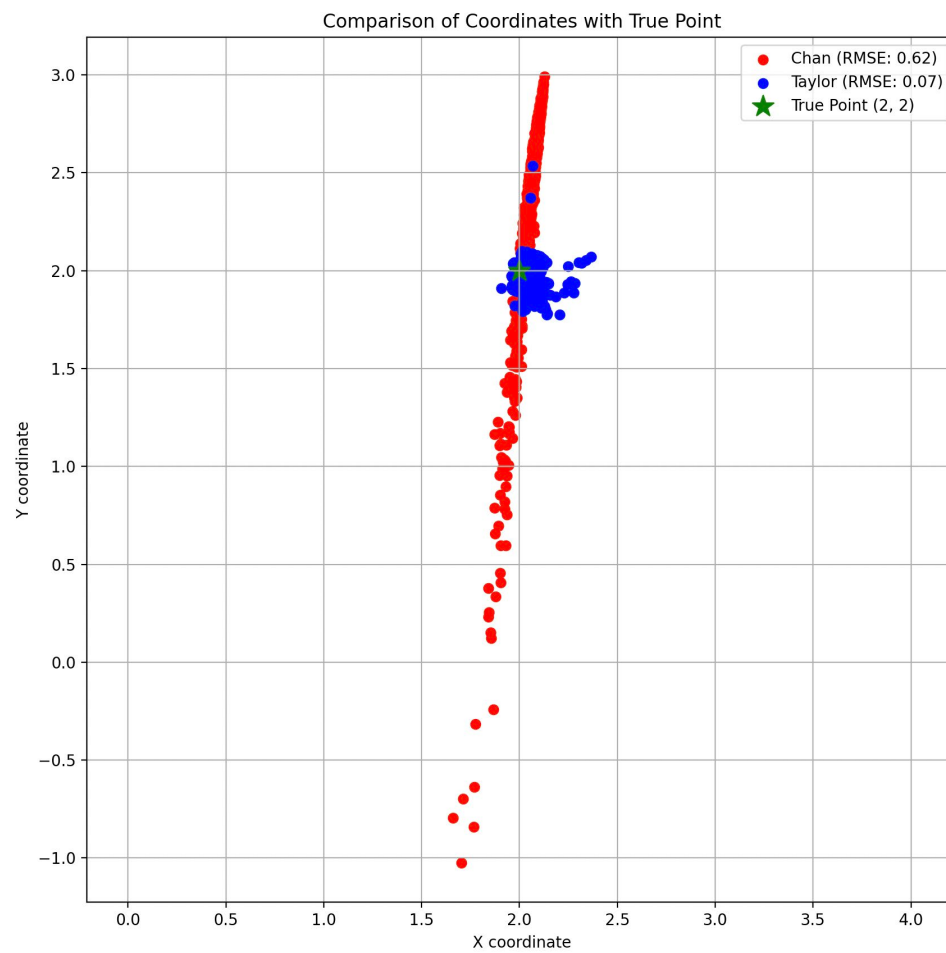


图 6-2 算法定位散点图对比

6.1 Chan 算法

Chan 算法文件位于 `rtls_pc_tdoa/tdoa/chan_tdoa_2D.cpp`

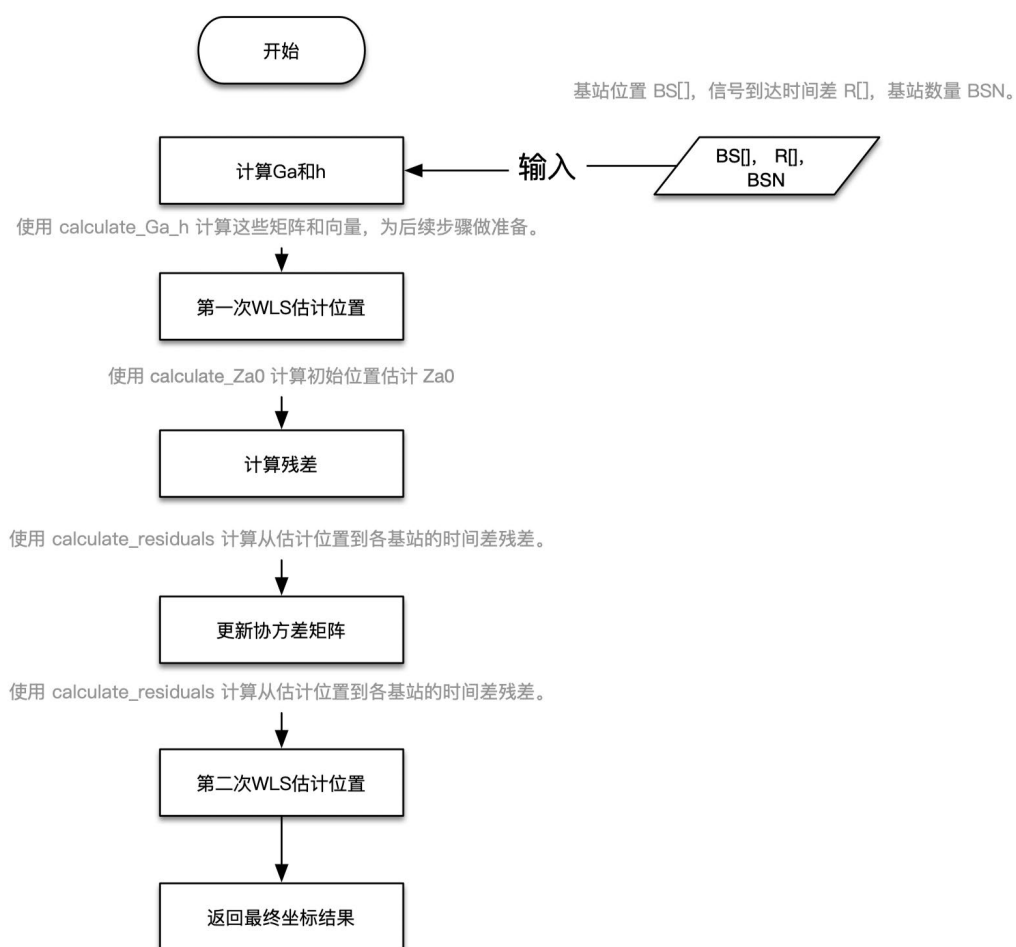


图 6-3 Chan 算法流程

6.2 Taylor 算法

Taylor 算法通过迭代方式逼近目标位置的最优估计。基本思想是，通过将目标函数（通常是到各个锚点的距离的平方和的误差）在当前估计点附近进行 Taylor 级数展开，然后求解展开式的最小值点，以此逐步逼近实际的目标位置。

这里使用 GSL 数学库进行辅助计算，简化代码提高效率。基本流程如图

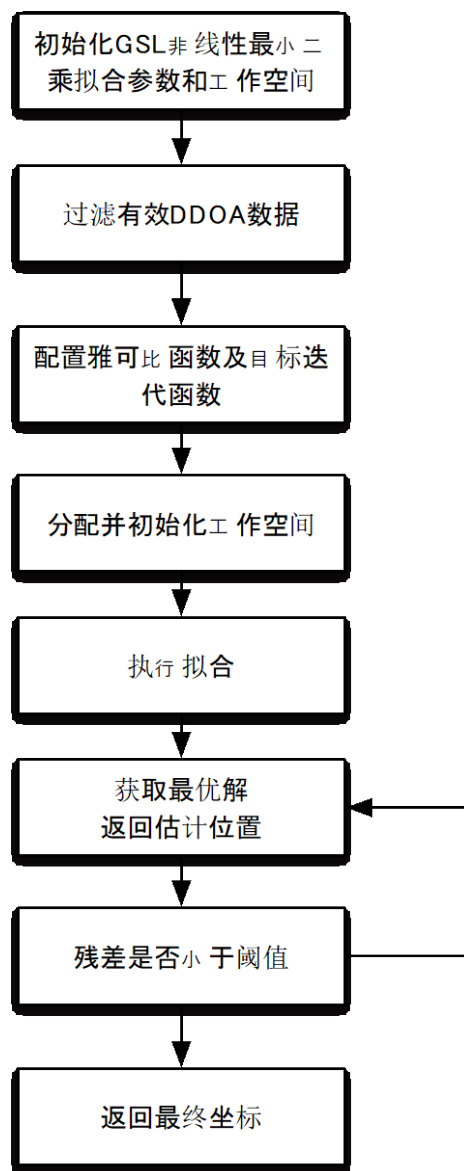


图 6-4 Taylor 算法流程

6.3 Chan_Taylor

Taylor 算法代码位于 tdoa/Taylor_tdoa.cpp

Chan-Taylor 联合算法. Chan-Taylor 联合算法是将 Chan 算法的解算结果作为初始估计坐标送入 Taylor 算法, 以达到对坐标进行误差计算和定位修正的作用. 算法在迭代时, 将误差与设定的阈值进行比较, 若误差值大于阈值, 则继续迭代;若误差值小于设定阈值, 则终止迭代并输出结果.

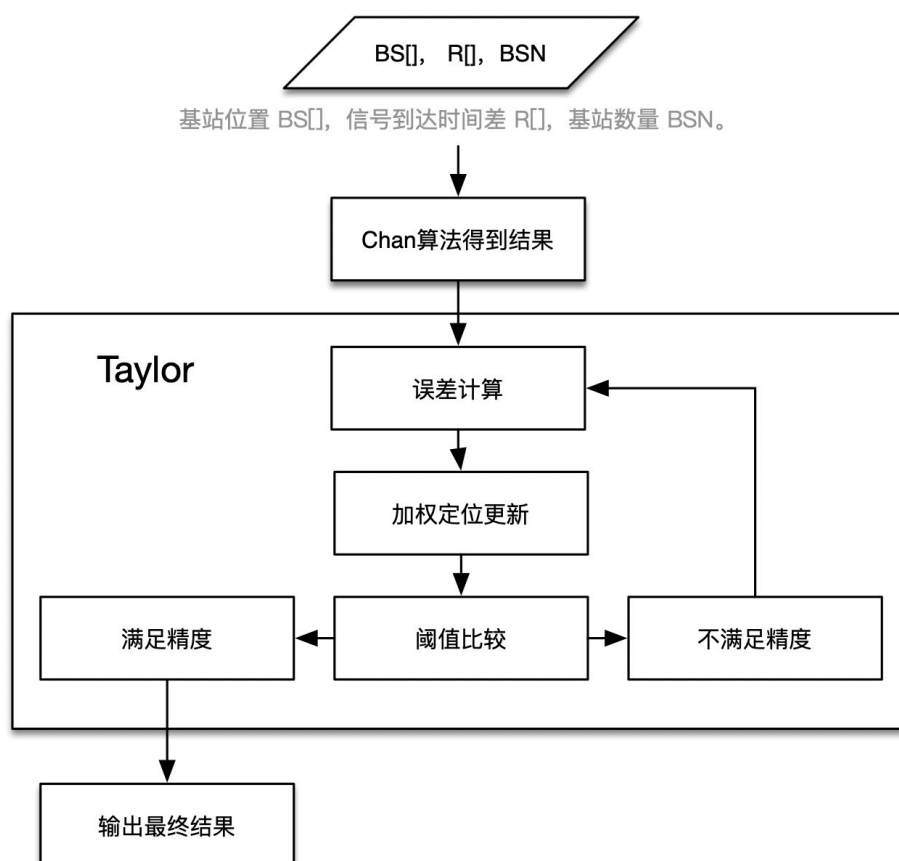


图 6-5 Chan-Taylor 联合算法流程

6.4 卡尔曼滤波

采用 Kalman 滤波器对 DDoA 值进行滤波，降低噪声对 ddoa 值的影响。代码位于 tools\kalman.cpp

- **常量定义** - 定义了 Q 和 R 两个常量，分别表示过程噪声和测量噪声的方差。
- **全局变量** - 使用一个三维数组 kalman 来存储不同标签和锚点的过滤状态和误差协方差。
- **卡尔曼滤波函数 (kalman_filter)** - 这个函数接受标签 ID (tag_id)，测量值 (kalman_val)，和锚点编号 (anc_num) 作为参数，并更新滤波状态，返回最新的估计值。

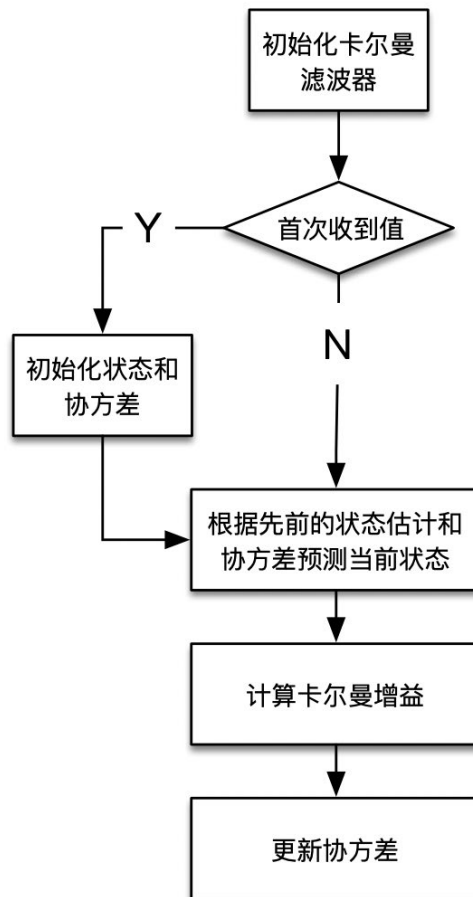


图 6-6 卡尔曼滤波流程

6.5 坐标滤波

采用平均滤波、去极值滤波 2 种算法对解算坐标结果进行滤波。两种算法类似，平均值滤波会选取最近 50 个坐标点做平均，而去极值滤波会先在 50 个坐标点种去掉最大值和最小值后再取平均值，使定位轨迹更平滑。

代码位于 network/RTLSCClient.cpp，函数名分别为：

Process_me () // 取极值滤波

Process_ma () //平均值滤波

6.6 权重设计

同时本引擎为了更好的优化 NLOS 定位精度，以及噪声干扰。对权重部分进行了着重处理，对信号功率、噪声等进行归一化，为每个 DDOA 进行单独权重计算。同时，根据环境对归一化参考值进行动态更新。

1. 基本参数获取

通过 DW1000 API，嵌入式获取相关底层射频参数：

RX_Power	估计接收信号功率，通过 CIR 中的 CIR_Power 经过公式计算
FP_Power	第一路径功率，UWB 信号第一路径传输的功率
STD_NOISE	噪声标准偏差

表 6-1 底层参数

2. 各参数归一化计算

将各参数进行统计，分别计算出最大值、最小值、平均值。并进行归一计算。得到 3 个权重 weight1, weight2, weight3，其中 weight3 为噪声值，越大代表可信度越差；根据其影响力分别添加系数。

```
double weight1 = (tagData->fp_power[anc_id] - cir.min_fp_power) / (cir.max_fp_power - cir.min_fp_power) * 0.05 ; // 第一路径功率归一 【0~1】
double weight2 = (tagData->rx_power[anc_id] - cir.min_rx_power) / (cir.max_rx_power - cir.min_rx_power) * 0.85 ; // 总功率归一 【0~1】
double weight3 = (1-((tagData->rx_noise[anc_id] - cir.min_noise) / (cir.max_noise - cir.min_noise))) * 0.1; //噪声归一
```

3. 参数动态更新

系统开始解算后会自动收集以上参数，累计 500 次之后重新根据当前环境动态更新最大值、最小值、平均值；让权重计算更适配当前环境

4. Cha 值计算

Cha 值为 rx_power 与 fp_power 的差，根据 DW1000 官方文档，评估 cha 值可以对 LOS 环境与否进行判断，当值小于 6db 时可估计为 LOS 环境，

大于 10db 可估计认为 NLOS 环境。

5. 权重最终计算

构建方程对 Cha 值进行权重计算，再与之前计算的 weight1,2,3 进行整合从而得到最终权重值 weight。

7 UI 设计、更新、显示

UI 显示部分代码位于 `rtls_pc_tdoa/views` 中，UI 设计位于负责各类数据的展示、更新以及各类 UI 点击、拖拽等事件。较为简单，这里就不在一一赘述。

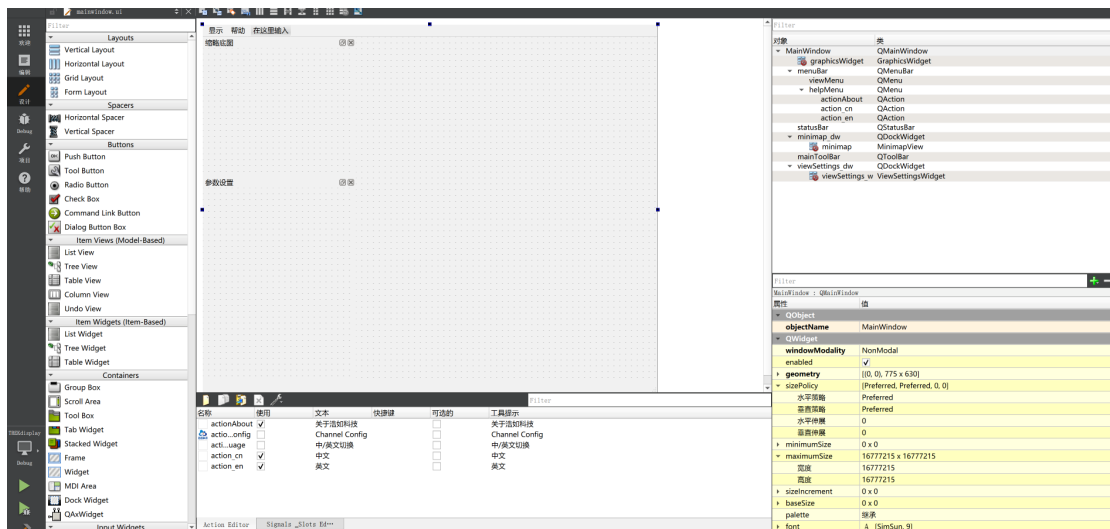


图 7-1 主窗口 UI 设计

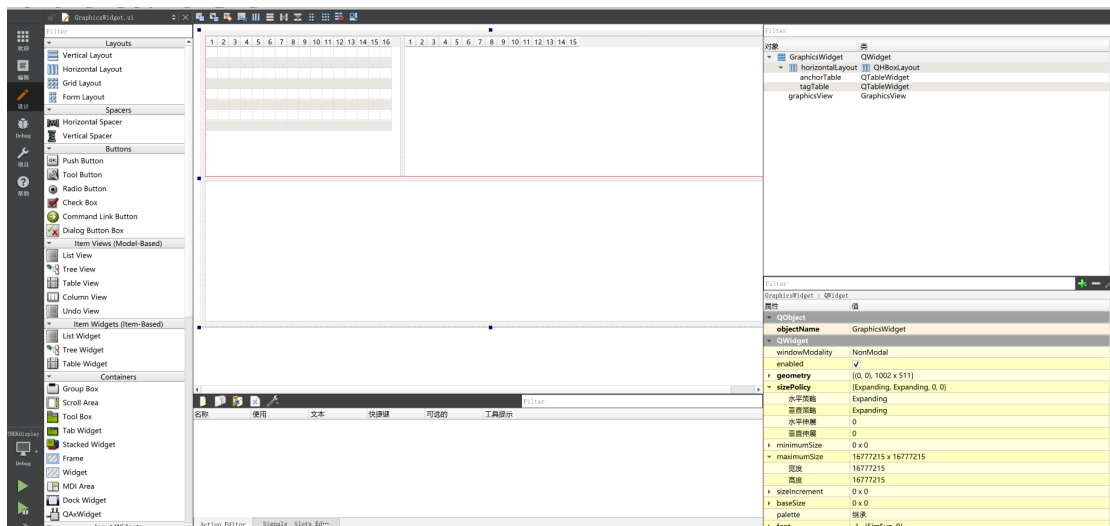


图 7-2 基站、标签列表 UI 设计

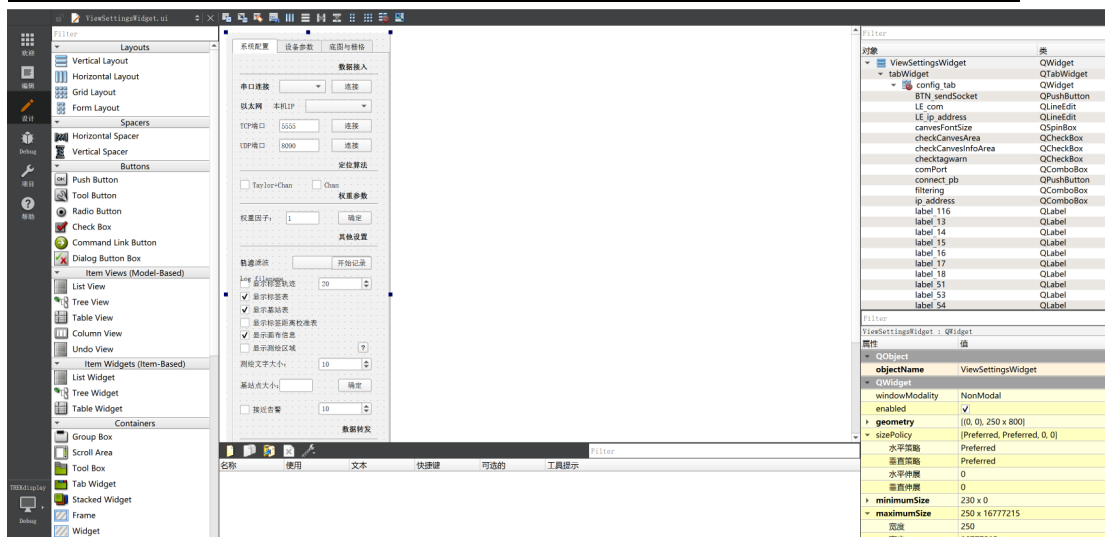


图 7-3 参数设计 UI 设计

8 开发环境说明

开发环境为 QT/C++语言

开发 IDE 使用 QT，我公司开发时使用的 QT 版本为 5.14.1，QT Creator 版本为 V4.11.1，理论上高版本也可向下兼容，如编译和运行出错，请降级至 V4.11.1 使用。



图 8-1 QT/C++版本

另需注意工程使用 GSL 库 2.7.1 版本，以编译并添加如 QT 工程中，若需
要自行编译参考这篇文章自行操作：

<https://blog.csdn.net/u013232740/article/details/127707289>

9 通讯协议

9.1 UDP 通讯协议

基站数据接收，使用 UDP 协议，QT 中用 Socket 实现接收并解析。

数据示例

示例 A0: mt,0,1,33,10872698843,10948064768,0,0,-78.16,-82.27,144

示例 A1: mt,1,2,33,610103618676,0,610179018053,-148,-78.27,-85.11,176

表 9-1 通信协议说明

内容	例子	功能
HEAD	mt	消息头，固定为 mt
ANCID	0	该条消息的基站 ID
TAGID	1	接收到定位请求的标签 ID
RN	33	Range number 不断累积增加范围 0~255
POLL_RX_TIME	10872698843	该基站收到标签发送的 POLL 消息时间戳
SYNC_TX_TIME	10948064768	主基站发送 SYNC 消息时间戳，只有主基站 A0 这个字段有效，其他从基站该字段为 0 无效
SYNC_RX_TIME	0	从基站接收主基站 SYNC 消息时间戳，只有从基站 A1,A2,A3...有效，主基站 A0 该字段为 0 无效
CLOCKOFF	0	从基站接收主基站 SYNC 消息频偏，只有从基站 A1,A2,A3...有效，主基站 A0 该字段为 0 无效
RX_PWR	-78.16	该基站接收标签消息的接收总功率，单位 dBm
FP_PWR	-82.27	该基站接收标签消息的第一路径功率，单位 dBm
STD_NOISE	144	该基站接收标签消息的噪声标准差 0~65535

9.2 数据转发协议

上位机软件可将标签定位结果数据通过 UDP 对外转发，在参数设置栏“数据转发”内设置远程主机 IP 和远程主机端口，即可将结果转发。

数据例：

```
{  
    "Command": "UpLink",  
    "TagID": 0,  
    "X": 1.089,  
    "Y": 1.056,  
}
```

表 9-2 数据转发协议说明

序号	KEY	Value	类型	说明
1	Command	UpLink	char	固定头
2	TagID	0	int	标签 ID
3	X	1.089	float	标签 X 坐标，单位米
4	Y	1.056	float	标签 Y 坐标，单位米